

gclib 2.0.8  
C API for Galil controllers and PLCs

Galil Motion Control

Thu Dec 7 2023



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Example Projects</b>	<b>3</b>
2.1	Commands Example	4
2.2	Message Example	5
2.3	Position Tracking Example	6
2.4	Jog Example	6
2.5	Vector Mode Example	7
2.6	IP Assigner Example	8
2.7	Motion Complete Example	9
2.8	Record Position Example	10
2.9	Contour Example	10
2.10	Remote Server Example	11
2.11	Remote Client Example	12
2.12	C/C++	13
2.12.1	Microsoft Windows	13
2.12.2	Linux	14
2.13	C#.NET	14
2.14	VB.NET	15
<b>3</b>	<b>Language Support</b>	<b>17</b>
3.1	C/C++	17
3.1.1	Microsoft Visual Studio 2019 (16.0)	17
3.1.2	Microsoft Visual Studio 2017 (15.0)	19
3.1.3	Microsoft Visual Studio 2015 (14.0)	22
3.1.4	Microsoft Visual Studio 2013 (12.0)	24
3.1.5	MinGW	26
3.1.6	Borland C++	29
3.1.7	gcc (Linux)	32
3.1.8	clang (OS X)	35
3.2	Python	38
3.3	.Net	41
3.3.1	VB.NET	41
3.3.2	C#.NET	43
3.4	Java	45
3.5	LabVIEW	47
<b>4</b>	<b>Using gclib</b>	<b>49</b>
4.1	gcaps	49
4.2	Program Preprocessor	50
4.3	Thread Safety	54
4.4	Galil Widgets	55
4.5	Rebuilding gclibo	55
4.6	Legacy Compatibility	59
4.6.1	GalilTools	59
4.6.2	DMC32 OSU	60

<b>5</b>	<b>Module Index</b>	<b>63</b>
5.1	Modules	63
<b>6</b>	<b>Namespace Index</b>	<b>65</b>
6.1	Namespace List	65
<b>7</b>	<b>Hierarchical Index</b>	<b>67</b>
7.1	Class Hierarchy	67
<b>8</b>	<b>Data Structure Index</b>	<b>69</b>
8.1	Data Structures	69
<b>9</b>	<b>File Index</b>	<b>73</b>
9.1	File List	73
<b>10</b>	<b>Module Documentation</b>	<b>77</b>
10.1	API	77
10.1.1	Description	77
10.1.2	Function Documentation	84
10.1.2.1	GAddresses()	84
10.1.2.2	GArrayDownload()	85
10.1.2.3	GArrayDownloadFile()	85
10.1.2.4	GArrayUpload()	85
10.1.2.5	GArrayUploadFile()	86
10.1.2.6	GAssign()	86
10.1.2.7	GClose()	87
10.1.2.8	GCmd()	87
10.1.2.9	GCmdD()	88
10.1.2.10	GCmdI()	88
10.1.2.11	GCmdT()	89
10.1.2.12	GCommand()	89
10.1.2.13	GError()	90
10.1.2.14	GFirmwareDownload()	90
10.1.2.15	GInfo()	90
10.1.2.16	GInterrupt()	91
10.1.2.17	GIpRequests()	91
10.1.2.18	GListServers()	92
10.1.2.19	GMessage()	92
10.1.2.20	GMotionComplete()	93
10.1.2.21	GOpen()	94
10.1.2.22	GProgramDownload()	95
10.1.2.23	GProgramDownloadFile()	95
10.1.2.24	GProgramUpload()	96
10.1.2.25	GProgramUploadFile()	96
10.1.2.26	GPublishServer()	97
10.1.2.27	GRead()	97
10.1.2.28	GRecord()	98
10.1.2.29	GRecordRate()	98
10.1.2.30	GRemoteConnections()	99
10.1.2.31	GServerStatus()	99
10.1.2.32	GSetServer()	100
10.1.2.33	GSetupDownloadFile()	100
10.1.2.34	GSleep()	101
10.1.2.35	GTimeout()	102
10.1.2.36	GUtility()	102
10.1.2.37	GVersion()	105
10.1.2.38	GWaitForBool()	105
10.1.2.39	GWrite()	106
10.2	C++ examples	106

10.2.1	Description	106
10.2.2	Function Documentation	108
10.2.2.1	commands()	108
10.2.2.2	contour()	108
10.2.2.3	e()	109
10.2.2.4	ip_assigner()	109
10.2.2.5	jog()	109
10.2.2.6	load_buffer()	111
10.2.2.7	main()	111
10.2.2.8	message()	112
10.2.2.9	motion_complete()	112
10.2.2.10	position_tracking()	112
10.2.2.11	record_position()	113
10.2.2.12	remote_client()	113
10.2.2.13	remote_server()	113
10.2.2.14	vector()	114
10.3	C#/VB examples	114
10.3.1	Description	114
10.3.2	Function Documentation	116
10.3.2.1	Commands()	116
10.3.2.2	Contour()	116
10.3.2.3	IP_Assigner()	117
10.3.2.4	Jog()	117
10.3.2.5	Message()	118
10.3.2.6	Motion_Complete()	118
10.3.2.7	Position_Tracking()	119
10.3.2.8	Record_Position()	119
10.3.2.9	Remote_Client()	120
10.3.2.10	Remote_Server()	120
10.3.2.11	Vector_Mode()	120
10.4	C#/VB API	121
10.4.1	Description	121
<b>11</b>	<b>Namespace Documentation</b>	<b>123</b>
11.1	examples Namespace Reference	123
11.1.1	Function Documentation	124
11.1.1.1	Commands()	124
11.1.1.2	Main()	124
11.1.1.3	PrintError()	125
11.1.1.4	Remote_Client()	125
11.1.1.5	Remote_Server()	125
11.1.2	Variable Documentation	125
11.1.2.1	Examples	126
11.2	gclib_example Namespace Reference	126
11.3	Package gclibjava	126
11.3.1	Detailed Description	126
11.4	vb Namespace Reference	126
<b>12</b>	<b>Data Structure Documentation</b>	<b>127</b>
12.1	Commands_Example Class Reference	127
12.1.1	Detailed Description	127
12.1.2	Member Function Documentation	127
12.1.2.1	Main()	127
12.2	Contour_Example Class Reference	128
12.2.1	Detailed Description	128
12.2.2	Member Function Documentation	128
12.2.2.1	Main()	128
12.3	Examples Class Reference	128
12.3.1	Detailed Description	129

12.3.2	Member Function Documentation	129
12.3.2.1	Commands()	129
12.3.2.2	Contour()	130
12.3.2.3	IP_Assigner()	130
12.3.2.4	Jog()	131
12.3.2.5	Message()	131
12.3.2.6	Motion_Complete()	131
12.3.2.7	Position_Tracking()	132
12.3.2.8	PrintError()	132
12.3.2.9	Record_Position()	133
12.3.2.10	Remote_Client()	133
12.3.2.11	Remote_Server()	133
12.3.2.12	Vector_Mode()	134
12.4	Galil Class Reference	135
12.4.1	Detailed Description	135
12.5	GalilPrivate Class Reference	136
12.5.1	Detailed Description	136
12.6	gclib Class Reference	136
12.6.1	Detailed Description	138
12.6.2	Constructor & Destructor Documentation	138
12.6.2.1	gclib()	138
12.6.3	Member Function Documentation	138
12.6.3.1	GAddresses()	139
12.6.3.2	GArrayDownload()	139
12.6.3.3	GArrayDownloadFile()	139
12.6.3.4	GArrayUpload()	140
12.6.3.5	GArrayUploadFile()	140
12.6.3.6	GAssign()	140
12.6.3.7	GClose()	141
12.6.3.8	GCmdD()	141
12.6.3.9	GCmdI()	141
12.6.3.10	GCommand()	142
12.6.3.11	GFirmwareDownload()	142
12.6.3.12	GInfo()	143
12.6.3.13	GInterrupt()	143
12.6.3.14	GIpRequests()	143
12.6.3.15	GListServers()	143
12.6.3.16	GMessage()	144
12.6.3.17	GMotionComplete()	144
12.6.3.18	GOpen()	144
12.6.3.19	GProgramDownload()	145
12.6.3.20	GProgramDownloadFile()	145
12.6.3.21	GProgramUpload()	146
12.6.3.22	GProgramUploadFile()	146
12.6.3.23	GPublishServer()	146
12.6.3.24	GRead()	147
12.6.3.25	GRecord< T >()	147
12.6.3.26	GRecordRate()	147
12.6.3.27	GRemoteConnections()	148
12.6.3.28	GServerStatus()	148
12.6.3.29	GSetServer()	148
12.6.3.30	GSetupDownloadFile()	149
12.6.3.31	GTimeout()	149
12.6.3.32	GVersion()	149
12.6.3.33	GWrite()	150
12.7	Gclib Class Reference	150
12.7.1	Detailed Description	152
12.7.2	Member Function Documentation	152

12.7.2.1	GAddresses()	152
12.7.2.2	GArrayDownload()	152
12.7.2.3	GArrayDownloadFile()	153
12.7.2.4	GArrayUpload()	153
12.7.2.5	GArrayUploadFile()	153
12.7.2.6	GAssign()	154
12.7.2.7	GClose()	154
12.7.2.8	GCmdD()	154
12.7.2.9	GCmdI()	155
12.7.2.10	GCommand()	155
12.7.2.11	GFirmwareDownload()	155
12.7.2.12	GInfo()	156
12.7.2.13	GInterrupt()	156
12.7.2.14	GIpRequests()	156
12.7.2.15	GListServers()	156
12.7.2.16	GMessage()	157
12.7.2.17	GMotionComplete()	157
12.7.2.18	GOpen()	157
12.7.2.19	GProgramDownload()	157
12.7.2.20	GProgramDownloadFile()	158
12.7.2.21	GProgramUpload()	158
12.7.2.22	GProgramUploadFile()	158
12.7.2.23	GPublishServer()	159
12.7.2.24	GRead()	159
12.7.2.25	GRecord()	159
12.7.2.26	GRecordRate()	160
12.7.2.27	GRemoteConnections()	160
12.7.2.28	GServerStatus()	160
12.7.2.29	GSetServer()	161
12.7.2.30	GSetupDownloadFile()	161
12.7.2.31	GTimeout()	161
12.7.2.32	GVersion()	162
12.7.2.33	GWrite()	162
12.7.2.34	New()	162
12.8	gclib.GclibError Class Reference	162
12.8.1	Detailed Description	163
12.9	GclibJava Class Reference	163
12.9.1	Detailed Description	163
12.9.2	Constructor & Destructor Documentation	164
12.9.2.1	GclibJava()	164
12.9.3	Member Function Documentation	164
12.9.3.1	finalize()	164
12.9.3.2	GAddresses()	164
12.9.3.3	GArrayDownload()	164
12.9.3.4	GArrayDownloadFile()	165
12.9.3.5	GArrayUpload()	165
12.9.3.6	GArrayUploadFile() [1/2]	165
12.9.3.7	GArrayUploadFile() [2/2]	166
12.9.3.8	GAssign()	166
12.9.3.9	GClose()	166
12.9.3.10	GCommand()	167
12.9.3.11	GFirmwareDownload()	167
12.9.3.12	GInfo()	167
12.9.3.13	GInterrupt()	168
12.9.3.14	GIpRequests()	168
12.9.3.15	GListServers()	168
12.9.3.16	GMessage()	169
12.9.3.17	GOpen()	169

12.9.3.18 GProgramDownload() [1/2]	169
12.9.3.19 GProgramDownload() [2/2]	170
12.9.3.20 GProgramDownloadFile() [1/2]	170
12.9.3.21 GProgramDownloadFile() [2/2]	170
12.9.3.22 GProgramUpload()	171
12.9.3.23 GProgramUploadFile()	171
12.9.3.24 GPublishServer()	171
12.9.3.25 GRemoteConnections()	172
12.9.3.26 GServerStatus()	172
12.9.3.27 GSetServer()	172
12.9.3.28 GSleep()	173
12.9.3.29 GTimeout()	173
12.9.3.30 GVersion()	173
12.10 GcLibJavaException Class Reference	173
12.10.1 Detailed Description	174
12.11 GcLibTest Class Reference	174
12.11.1 Detailed Description	174
12.11.2 Member Function Documentation	174
12.11.2.1 main()	174
12.12 GcLib.GDataRecord Interface Reference	174
12.12.1 Detailed Description	175
12.13 GDataRecord Union Reference	175
12.13.1 Detailed Description	176
12.14 GcLib.GDataRecord Interface Reference	176
12.14.1 Detailed Description	177
12.15 GcLib.GDataRecord1802 Struct Reference	177
12.15.1 Detailed Description	181
12.16 GDataRecord1802 Struct Reference	181
12.16.1 Detailed Description	184
12.17 GcLib.GDataRecord1802 Struct Reference	184
12.17.1 Detailed Description	187
12.17.2 Member Function Documentation	187
12.17.2.1 byte_array()	187
12.18 GcLib.GDataRecord1806 Struct Reference	188
12.18.1 Detailed Description	194
12.19 GDataRecord1806 Struct Reference	194
12.19.1 Detailed Description	201
12.20 GcLib.GDataRecord1806 Struct Reference	201
12.20.1 Detailed Description	207
12.20.2 Member Function Documentation	207
12.20.2.1 byte_array()	207
12.21 GcLib.GDataRecord2103 Struct Reference	208
12.21.1 Detailed Description	212
12.22 GDataRecord2103 Struct Reference	212
12.22.1 Detailed Description	217
12.23 GcLib.GDataRecord2103 Struct Reference	217
12.23.1 Detailed Description	222
12.23.2 Member Function Documentation	222
12.23.2.1 byte_array()	222
12.24 GcLib.GDataRecord30000 Struct Reference	222
12.24.1 Detailed Description	224
12.25 GDataRecord30000 Struct Reference	224
12.25.1 Detailed Description	226
12.26 GcLib.GDataRecord30000 Struct Reference	226
12.26.1 Detailed Description	227
12.26.2 Member Function Documentation	227
12.26.2.1 byte_array()	228
12.27 GcLib.GDataRecord4000 Struct Reference	228



12.27.1 Detailed Description	234
12.28GDataRecord4000 Struct Reference	234
12.28.1 Detailed Description	241
12.29Gclib.GDataRecord4000 Struct Reference	241
12.29.1 Detailed Description	247
12.29.2 Member Function Documentation	248
12.29.2.1 byte_array()	248
12.30gclib.GDataRecord47000_ENC Struct Reference	248
12.30.1 Detailed Description	249
12.31GDataRecord47000_ENC Struct Reference	250
12.31.1 Detailed Description	251
12.32Gclib.GDataRecord47000_ENC Struct Reference	251
12.32.1 Detailed Description	253
12.32.2 Member Function Documentation	253
12.32.2.1 byte_array()	253
12.33gclib.GDataRecord47162 Struct Reference	253
12.33.1 Detailed Description	255
12.34GDataRecord47162 Struct Reference	255
12.34.1 Detailed Description	257
12.35Gclib.GDataRecord47162 Struct Reference	257
12.35.1 Detailed Description	259
12.35.2 Member Function Documentation	259
12.35.2.1 byte_array()	259
12.36gclib.GDataRecord47300_24EX Struct Reference	259
12.36.1 Detailed Description	261
12.37GDataRecord47300_24EX Struct Reference	261
12.37.1 Detailed Description	262
12.38Gclib.GDataRecord47300_24EX Struct Reference	263
12.38.1 Detailed Description	264
12.38.2 Member Function Documentation	264
12.38.2.1 byte_array()	264
12.39gclib.GDataRecord47300_ENC Struct Reference	265
12.39.1 Detailed Description	266
12.40GDataRecord47300_ENC Struct Reference	266
12.40.1 Detailed Description	268
12.41Gclib.GDataRecord47300_ENC Struct Reference	268
12.41.1 Detailed Description	270
12.41.2 Member Function Documentation	270
12.41.2.1 byte_array()	270
12.42gclib.GDataRecord52000 Struct Reference	270
12.42.1 Detailed Description	277
12.43GDataRecord52000 Struct Reference	277
12.43.1 Detailed Description	283
12.44Gclib.GDataRecord52000 Struct Reference	284
12.44.1 Detailed Description	290
12.44.2 Member Function Documentation	290
12.44.2.1 byte_array()	290
12.45H_ArrayData Struct Reference	290
12.45.1 Detailed Description	291
12.46IP_Assigner_Example Class Reference	291
12.46.1 Detailed Description	291
12.46.2 Member Function Documentation	291
12.46.2.1 Main()	291
12.47Jog_Example Class Reference	292
12.47.1 Detailed Description	292
12.47.2 Member Function Documentation	292
12.47.2.1 Main()	292
12.48MainForm Class Reference	292

12.48.1 Detailed Description	293
12.49 Message_Example Class Reference	293
12.49.1 Detailed Description	293
12.49.2 Member Function Documentation	293
12.49.2.1 Main()	293
12.50 Motion_Complete_Example Class Reference	293
12.50.1 Detailed Description	294
12.50.2 Member Function Documentation	294
12.50.2.1 Main()	294
12.51 Position_Tracking_Example Class Reference	294
12.51.1 Detailed Description	294
12.51.2 Member Function Documentation	294
12.51.2.1 Main()	295
12.52 Program Class Reference	295
12.52.1 Detailed Description	295
12.53 gclib.py Class Reference	295
12.53.1 Detailed Description	296
12.53.2 Constructor & Destructor Documentation	296
12.53.2.1 __init__()	296
12.53.2.2 __del__()	296
12.53.3 Member Function Documentation	296
12.53.3.1 GAddresses()	296
12.53.3.2 GArrayDownload()	297
12.53.3.3 GArrayDownloadFile()	297
12.53.3.4 GArrayUpload()	297
12.53.3.5 GArrayUploadFile()	297
12.53.3.6 GAssign()	297
12.53.3.7 GClose()	298
12.53.3.8 GCommand()	298
12.53.3.9 GFirmwareDownload()	298
12.53.3.10 GInfo()	298
12.53.3.11 GInterrupt()	298
12.53.3.12 GIpRequests()	299
12.53.3.13 GMessage()	299
12.53.3.14 GMotionComplete()	299
12.53.3.15 GOpen()	299
12.53.3.16 GProgramDownload()	299
12.53.3.17 GProgramDownloadFile()	300
12.53.3.18 GProgramUpload()	300
12.53.3.19 GProgramUploadFile()	300
12.53.3.20 GSetupDownloadFile()	300
12.53.3.21 GSleep()	300
12.53.3.22 GTimeout()	301
12.53.3.23 GVersion()	301
12.53.3.24 timeout() [1/2]	301
12.53.3.25 timeout() [2/2]	301
12.54 Record_Position_Example Class Reference	301
12.54.1 Detailed Description	302
12.54.2 Member Function Documentation	302
12.54.2.1 Main()	302
12.55 Remote_Client_Example Class Reference	302
12.55.1 Detailed Description	302
12.55.2 Member Function Documentation	302
12.55.2.1 Main()	303
12.56 Remote_Server_Example Class Reference	303
12.56.1 Detailed Description	303
12.56.2 Member Function Documentation	303
12.56.2.1 Main()	303

12.57	Source Struct Reference	304
12.57.1	Detailed Description	304
12.58	Vector_Mode_Example Class Reference	304
12.58.1	Detailed Description	304
12.58.2	Member Function Documentation	304
12.58.2.1	Main()	304
<b>13</b>	<b>File Documentation</b>	<b>307</b>
13.1	arrays.c File Reference	307
13.1.1	Detailed Description	308
13.1.2	Function Documentation	308
13.1.2.1	GArrayDownloadFile()	308
13.1.2.2	GArrayUploadFile()	308
13.1.2.3	GSetupDownloadFile()	309
13.1.2.4	H_DownloadArraysFromList()	310
13.2	commands.cpp File Reference	310
13.2.1	Detailed Description	310
13.2.2	Function Documentation	310
13.2.2.1	commands()	310
13.3	commands.cs File Reference	311
13.3.1	Detailed Description	311
13.4	Commands.vb File Reference	311
13.5	commands_example.cpp File Reference	311
13.5.1	Detailed Description	311
13.5.2	Function Documentation	311
13.5.2.1	main()	311
13.6	commands_example.cs File Reference	312
13.6.1	Detailed Description	312
13.7	Commands_Example.vb File Reference	312
13.8	contour.cpp File Reference	312
13.8.1	Detailed Description	313
13.8.2	Function Documentation	313
13.8.2.1	contour()	313
13.9	contour.cs File Reference	313
13.9.1	Detailed Description	313
13.10	Contour.vb File Reference	313
13.11	contour_example.cpp File Reference	314
13.11.1	Detailed Description	314
13.11.2	Function Documentation	314
13.11.2.1	main()	314
13.12	contour_example.cs File Reference	315
13.12.1	Detailed Description	315
13.13	Contour_Example.vb File Reference	315
13.14	example.py File Reference	315
13.15	examples.cs File Reference	315
13.15.1	Detailed Description	315
13.16	examples.h File Reference	315
13.16.1	Detailed Description	316
13.16.2	Function Documentation	316
13.16.2.1	commands()	316
13.16.2.2	contour()	317
13.16.2.3	e()	317
13.16.2.4	ip_assigner()	317
13.16.2.5	jog()	318
13.16.2.6	message()	318
13.16.2.7	motion_complete()	318
13.16.2.8	position_tracking()	319
13.16.2.9	record_position()	319

13.16.2.10remote_client()	319
13.16.2.11remote_server()	320
13.16.2.12vector()	320
13.17Examples.vb File Reference	321
13.18Form1.cs File Reference	321
13.18.1 Detailed Description	321
13.19Form1.vb File Reference	321
13.20Galil.h File Reference	321
13.21gcl_datarecord.cpp File Reference	321
13.22gcl_galil.cpp File Reference	322
13.23gcl_galil.h File Reference	322
13.24gcl_simple.cpp File Reference	322
13.25gclib.cs File Reference	322
13.26gclib.h File Reference	323
13.26.1 Detailed Description	326
13.26.2 Function Documentation	326
13.26.2.1 GArrayDownload()	326
13.26.2.2 GArrayUpload()	327
13.26.2.3 GClose()	327
13.26.2.4 GCommand()	328
13.26.2.5 GFirmwareDownload()	328
13.26.2.6 GInterrupt()	329
13.26.2.7 GMessage()	329
13.26.2.8 GOpen()	330
13.26.2.9 GProgramDownload()	331
13.26.2.10GProgramUpload()	332
13.26.2.11GRead()	332
13.26.2.12GRecord()	332
13.26.2.13GUtility()	333
13.26.2.14GWrite()	336
13.27gclib.py File Reference	336
13.28gclib.vb File Reference	337
13.29gclib_errors.h File Reference	337
13.29.1 Detailed Description	339
13.30gclib_record.h File Reference	339
13.30.1 Detailed Description	340
13.31GclibJava.java File Reference	340
13.32GclibJavaException.java File Reference	340
13.33gclibo.c File Reference	341
13.33.1 Detailed Description	342
13.33.2 Function Documentation	342
13.33.2.1 GAddresses()	342
13.33.2.2 GAssign()	343
13.33.2.3 GCmd()	343
13.33.2.4 GCmdD()	344
13.33.2.5 GCmdI()	344
13.33.2.6 GCmdT()	344
13.33.2.7 GError()	345
13.33.2.8 GInfo()	345
13.33.2.9 GIpRequests()	346
13.33.2.10GListServers()	346
13.33.2.11GMotionComplete()	347
13.33.2.12GProgramDownloadFile()	347
13.33.2.13GProgramUploadFile()	348
13.33.2.14GPublishServer()	348
13.33.2.15GRecordRate()	349
13.33.2.16GRemoteConnections()	349
13.33.2.17GServerStatus()	350

13.33.2.18GSetServer()	350
13.33.2.19GSleep()	351
13.33.2.20GTimeout()	351
13.33.2.21GVersion()	352
13.33.2.22GWaitForBool()	352
13.34gclibo.h File Reference	353
13.34.1 Detailed Description	354
13.34.2 Function Documentation	354
13.34.2.1 GAddresses()	354
13.34.2.2 GArrayDownloadFile()	355
13.34.2.3 GArrayUploadFile()	355
13.34.2.4 GAssign()	356
13.34.2.5 GCmd()	356
13.34.2.6 GCmdD()	357
13.34.2.7 GCmdI()	357
13.34.2.8 GCmdT()	358
13.34.2.9 GError()	358
13.34.2.10GInfo()	359
13.34.2.11GIpRequests()	359
13.34.2.12GListServers()	360
13.34.2.13GMotionComplete()	360
13.34.2.14GProgramDownloadFile()	361
13.34.2.15GProgramUploadFile()	361
13.34.2.16GPublishServer()	361
13.34.2.17GRecordRate()	362
13.34.2.18GRemoteConnections()	362
13.34.2.19GServerStatus()	363
13.34.2.20GSetServer()	364
13.34.2.21GSetupDownloadFile()	364
13.34.2.22GSleep()	365
13.34.2.23GTimeout()	365
13.34.2.24GVersion()	366
13.34.2.25GWaitForBool()	366
13.35GclibTest.java File Reference	367
13.36ip_assigner.cpp File Reference	367
13.36.1 Detailed Description	367
13.36.2 Function Documentation	367
13.36.2.1 ip_assigner()	367
13.37ip_assigner.cs File Reference	368
13.37.1 Detailed Description	368
13.38IP_Assigner.vb File Reference	368
13.39ip_assigner_example.cpp File Reference	368
13.39.1 Detailed Description	368
13.39.2 Function Documentation	368
13.39.2.1 main()	368
13.40ip_assigner_example.cs File Reference	369
13.40.1 Detailed Description	369
13.41IP_Assigner_Example.vb File Reference	369
13.42jog.cpp File Reference	369
13.42.1 Detailed Description	370
13.42.2 Function Documentation	370
13.42.2.1 jog()	370
13.43jog.cs File Reference	370
13.43.1 Detailed Description	370
13.44Jog.vb File Reference	370
13.45jog_example.cpp File Reference	370
13.45.1 Detailed Description	371
13.45.2 Function Documentation	371

13.45.2.1 main()	371
13.46jog_example.cs File Reference	371
13.46.1 Detailed Description	372
13.47Jog_Example.vb File Reference	372
13.48main.cpp File Reference	372
13.49message.cpp File Reference	372
13.49.1 Detailed Description	372
13.49.2 Function Documentation	372
13.49.2.1 message()	372
13.50message.cs File Reference	373
13.50.1 Detailed Description	373
13.51Message.vb File Reference	373
13.52message_example.cpp File Reference	373
13.52.1 Detailed Description	373
13.52.2 Function Documentation	373
13.52.2.1 main()	373
13.53message_example.cs File Reference	374
13.53.1 Detailed Description	374
13.54Message_Example.vb File Reference	374
13.55motion_complete.cpp File Reference	374
13.55.1 Detailed Description	375
13.55.2 Function Documentation	375
13.55.2.1 motion_complete()	375
13.56motion_complete.cs File Reference	375
13.56.1 Detailed Description	375
13.57Motion_Complete.vb File Reference	375
13.58motion_complete_example.cpp File Reference	375
13.58.1 Detailed Description	376
13.58.2 Function Documentation	376
13.58.2.1 main()	376
13.59motion_complete_example.cs File Reference	376
13.59.1 Detailed Description	376
13.60Motion_Complete_Example.vb File Reference	377
13.61position_tracking.cpp File Reference	377
13.61.1 Detailed Description	377
13.61.2 Function Documentation	377
13.61.2.1 position_tracking()	377
13.62position_tracking.cs File Reference	377
13.62.1 Detailed Description	377
13.63Position_Tracking.vb File Reference	378
13.64position_tracking_example.cpp File Reference	378
13.64.1 Detailed Description	378
13.64.2 Function Documentation	378
13.64.2.1 main()	378
13.65position_tracking_example.cs File Reference	379
13.65.1 Detailed Description	379
13.66Position_Tracking_Example.vb File Reference	379
13.67Program.cs File Reference	379
13.68record_position.cpp File Reference	379
13.68.1 Detailed Description	379
13.68.2 Function Documentation	379
13.68.2.1 record_position()	380
13.69record_position.cs File Reference	380
13.69.1 Detailed Description	380
13.70Record_Position.vb File Reference	380
13.71record_position_example.cpp File Reference	380
13.71.1 Detailed Description	380
13.71.2 Function Documentation	381

13.71.2.1 main()	381
13.72record_position_example.cs File Reference	381
13.72.1 Detailed Description	381
13.73Record_Position_Example.vb File Reference	382
13.74remote_client.cpp File Reference	382
13.74.1 Detailed Description	382
13.74.2 Function Documentation	382
13.74.2.1 remote_client()	382
13.75Remote_Client.cs File Reference	382
13.75.1 Detailed Description	383
13.76Remote_Client.vb File Reference	383
13.77remote_client_example.cpp File Reference	383
13.77.1 Detailed Description	383
13.77.2 Function Documentation	383
13.77.2.1 main()	383
13.78remote_client_example.cs File Reference	384
13.78.1 Detailed Description	384
13.79Remote_Client_Example.vb File Reference	384
13.80remote_server.cpp File Reference	384
13.80.1 Detailed Description	384
13.80.2 Function Documentation	384
13.80.2.1 remote_server()	385
13.81Remote_Server.cs File Reference	385
13.81.1 Detailed Description	385
13.82Remote_Server.vb File Reference	385
13.83remote_server_example.cpp File Reference	385
13.83.1 Detailed Description	386
13.83.2 Function Documentation	386
13.83.2.1 main()	386
13.84remote_server_example.cs File Reference	386
13.84.1 Detailed Description	386
13.85Remote_Server_Example.vb File Reference	387
13.86setup.py File Reference	387
13.87vector.cpp File Reference	387
13.87.1 Detailed Description	387
13.87.2 Function Documentation	387
13.87.2.1 load_buffer()	387
13.87.2.2 vector()	387
13.88vector_example.cpp File Reference	388
13.88.1 Detailed Description	388
13.88.2 Function Documentation	388
13.88.2.1 main()	388
13.89vector_mode.cs File Reference	389
13.89.1 Detailed Description	389
13.90Vector_Mode.vb File Reference	389
13.91vector_mode_example.cs File Reference	389
13.91.1 Detailed Description	389
13.92Vector_Mode_Example.vb File Reference	390
13.93x_arrays.cpp File Reference	390
13.93.1 Detailed Description	390
13.93.2 Function Documentation	390
13.93.2.1 x_arrays()	390
13.94x_examples.cpp File Reference	390
13.94.1 Detailed Description	390
13.95x_examples.h File Reference	390
13.95.1 Detailed Description	391
13.95.2 Function Documentation	391
13.95.2.1 x_arrays()	391

13.95.2.2	x_dr_motioncomplete()	392
13.95.2.3	x_e()	392
13.95.2.4	x_ei_motioncomplete()	392
13.95.2.5	x_gcommand()	392
13.95.2.6	x_ginterrupt()	392
13.95.2.7	x_gmessage()	393
13.95.2.8	x_gmotioncomplete()	393
13.95.2.9	x_gread_gwrite()	393
13.95.2.10	x_grecord()	393
13.95.2.11	x_nonblocking()	393
13.95.2.12	x_programs()	394
13.96	x_gcommand.cpp File Reference	394
13.96.1	Detailed Description	394
13.96.2	Function Documentation	394
13.96.2.1	x_gcommand()	394
13.97	x_ginterrupt.cpp File Reference	394
13.97.1	Detailed Description	394
13.97.2	Function Documentation	394
13.97.2.1	x_ei_motioncomplete()	395
13.97.2.2	x_ginterrupt()	395
13.98	x_gmessage.cpp File Reference	395
13.98.1	Detailed Description	395
13.98.2	Function Documentation	395
13.98.2.1	x_gmessage()	395
13.99	x_gmotioncomplete.cpp File Reference	396
13.99.1	Detailed Description	396
13.99.2	Function Documentation	396
13.99.2.1	x_gmotioncomplete()	396
13.100	x_gread_gwrite.cpp File Reference	396
13.100.1	Detailed Description	396
13.100.2	Function Documentation	396
13.100.2.1	x_gread_gwrite()	396
13.101	x_grecord.cpp File Reference	397
13.101.1	Detailed Description	397
13.101.2	Function Documentation	397
13.101.2.1	x_dr_motioncomplete()	397
13.101.2.2	x_grecord()	397
13.102	x_nonblocking.cpp File Reference	397
13.102.1	Detailed Description	398
13.102.2	Function Documentation	398
13.102.2.1	x_nonblocking()	398
13.103	x_programs.cpp File Reference	398
13.103.1	Detailed Description	398
13.103.2	Function Documentation	398
13.103.2.1	x_programs()	398
13.104	x_simple.c File Reference	399
13.104.1	Detailed Description	399



# Chapter 1

## Introduction

The Galil Communication Library (gclib) is a communication library for Galil motion controllers and PLCs. A number of programming languages, operating systems, and hardware platforms are supported.

The library consists of a basic set of function calls ([gclib.h](#)), and an open-source extension library ([gclibo.h](#)). A number of examples are provided to demonstrate how to use the library with various [languages](#).

gclib will import virtually anywhere a dll/so/dylib can be imported. Please contact [support@galil.com](mailto:support@galil.com) if the language or platform required is not listed.

## Installation

- [Windows](#)
- [Linux](#)

## License

gclib binaries are covered under the [Galil Closed Source License](#).

The open source portion (gclibo), examples, and wrappers are covered under the [Galil Open Source License](#).

gclib and gcaps use [OpenSSL](#), which is licensed under the [Apache 2.0 license](#).

## Getting Started

- [Language Support](#)
- [Using gclib](#)
- [Example Projects](#)
- [List of all functions](#)

## Release Notes

See the update history of gclib in the [release notes](#).

Galil maintains an [RSS](#) page to notify users of updates.

See the update history of [gcaps](#) in the [release notes](#).

## Technical Support

For help please email [support@galil.com](mailto:support@galil.com), or call [Galil Applications](#).

## Chapter 2

# Example Projects

### Description

Welcome to gclib Example Projects. The Galil Communication Library (gclib) is a communication library for Galil motion controllers and PLCs. A number of programming languages, operating systems, and hardware platforms are supported. These in-depth examples will demonstrate how to use the basics of gclib such as connecting to the controller and issuing commands, as well as more advanced topics such as assigning a controller an IP Address and monitoring interrupts.

### Projects

Example	Description
<a href="#">Commands Example</a>	Demonstrates various uses of <a href="#">GCommand()</a> and <a href="#">GUtility()</a> .
<a href="#">Message Example</a>	Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.
<a href="#">Position Tracking Example</a>	Puts controller into Position Tracking Mode and accepts user-entered positions.
<a href="#">Jog Example</a>	Puts controller into Jog Mode and accepts user input to adjust the speed.
<a href="#">Vector Mode Example</a>	Puts controller into Vector Mode and accepts a file defining vector points.
<a href="#">IP Assigner Example</a>	Assigns controller an IP Address given a serial number and a 1 byte address.
<a href="#">Motion Complete Example</a>	Uses interrupts to track when the motion of controller is completed.
<a href="#">Record Position Example</a>	Record user's training and saves to a text file.
<a href="#">Contour Example</a>	Record user's training and plays back training through contour mode.
<a href="#">Remote Server Example</a>	Advertise local gcaps server on the network.
<a href="#">Remote Client Example</a>	Discover and connect to other gcaps servers on the network.

### Instructions

For build instructions, please select a supported language:

- [C/C++](#)
- [C#.NET](#)
- [VB.NET](#)

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

## 2.1 Commands Example

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

### Concepts

This example demonstrates:

- How to connect to a controller via IP Address.
- How to issue basic commands.
- How to read data back from the controller.
- How to get context on errors that occur in the program.

### Prerequisites

A Galil controller is required for this example.

### Command Line Arguments

This example requires 1 argument:

- The IP Address of a Galil controller.

```
commands_example.exe 192.168.42.200
```

### Example Output

```
`*****`
`*****      GCmdT() example      *****`
`*****`
GCmdT() will return a trimmed response of GCommand()
The command 'PR ?,?' will return the relative position of the A and B axes
«PR ?,? with GCommand(): 0, 10000
:»
«PR ?,? with GCmdT(): 0, 10000»
`*****`
`*****      GCmdI() example      *****`
`*****`
GCmdI() will return the value of GCommand() parsed as an int
The command 'MG _LMS' will return the available space in the vector buffer of the S plane.
MG _LMS with GCmdT(): 511.0000
MG _LMS with GCmdI(): 511
`*****`
`*****      GCmd() example      *****`
`*****`
GCmd() will execute the given command but does not return a value.
GCmd is useful for basic operations such as beginning motion or setting speed
GCmd(g, "BG A");
GCmd(g, "SP 5000");
`*****`
`*****      GCmdD() example      *****`
`*****`
GCmdD() will return the value of GCommand parsed as a double
The command 'MG @AN[1]' will return the value of Analog Input 1
MG @AN[1] with GCmdD(): 9.7726
```

```

`*****`
`***** Galil Double Format *****`
`*****`
Galil Controllers expect double values to be formatted to 4 decimal places
Unformatted double value: 0.00235
Formatted double value rounded to 4 decimal places: 0.0024
`*****`
`***** G_UTIL_ERROR_CONTEXT example *****`
`*****`
GUtility() with G_UTIL_ERROR_CONTEXT: Broken Pipe

```

## 2.2 Message Example

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

### Concepts

This example demonstrates:

- How to connect to a controller via IP Address.
- How to issue basic commands.
- How to reconstitute a full message from `GMessage()`.
- How to detect differences in crashed DMC code and Trace.
- How to print messages.
- How to use Keep Alive to maintain connection to gcaps.

### Prerequisites

A Galil controller with a motor connected at the A axis is needed for this example.

### Command Line Arguments

This example requires 1 argument:

- The IP Address of a Galil controller.

```
message_example.exe 192.168.42.96
```

### Example Output

```

`*****`
Example GMessage() usage
`*****`
<HELLO WORLD
>
Trace Line: 0 i=0
Trace Line: 1 #A
Trace Line: 2 MGi
Standard Line: 0.0000

```

```
Trace Line: 3 i=i+1
Trace Line: 4 WT100
Trace Line: 5 JP#A,i<1
Trace Line: 6 i=i/0
Crashed Code: ?6 i=i/0
```

## 2.3 Position Tracking Example

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

### Concepts

This example demonstrates:

- How to connect to a controller via IP Address.
- How to issue basic commands.
- A controller in position tracking mode.

### Prerequisites

A Galil controller with a motor connected at the A axis is needed for this example.

### Command Line Arguments

This example has 1 required argument and 1 optional argument:

- Required: The IP Address of a Galil controller.
- Optional: The speed of the controller in Position Tracking mode (Default 5000).

```
position_tracking_example.exe 192.168.42.96 4000
```

### Example Output

```
Begin Position Tracking with speed 5000. Enter a non-number to exit.
Enter a new position:
4000
Enter a new position:
-8000
Enter a new position:
10000
Enter a new position:
```

## 2.4 Jog Example

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>

C++	C#	Visual Basic
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

## Concepts

This example demonstrates:

- How to connect to a controller via IP Address.
- How to issue basic commands.
- A controller in jogging mode.
- How to utilize keyboard input at the console.

## Prerequisites

A Galil controller with a motor connected at the A axis is needed for this example.

### Note

Linux users will need to install the ncurses library.

## Command Line Arguments

This example requires 1 argument:

- The IP Address of a Galil controller.

```
jog_example.exe 192.168.42.96
```

## Example Output

```
Enter a character on the keyboard to change the motor's speed:
<q> Quit
<a> -2000 counts/s
<s> -500 counts/s
<d> +500 counts/s
<f> +2000 counts/s
<r> Direction Reversal
Jog Speed: 0
Jog Speed: 2000
Jog Speed: 4000
Jog Speed: 6000
Jog Speed: -6000
Jog Speed: -8000
Jog Speed: -6000
```

## 2.5 Vector Mode Example

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

## Concepts

This example demonstrates:

- How to connect to a controller via IP Address.
- How to issue basic commands.
- A controller in vector mode.
- How to read and maintain the length of the vector buffer.
- How to read in a file of vector points and apply them to the controller.

## Prerequisites

A Galil controller with two motors: one connected at the A axis and the other connected at the B axis.

## Command Line Arguments

This example requires 2 arguments:

- The IP Address of a Galil controller.
- The path to a file containing vector commands.

```
vector_example.exe 192.168.42.92 vector_points.txt
```

## 2.6 IP Assigner Example

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

## Concepts

This example demonstrates:

- How to issue basic commands.
- How to listen on the network for Galil Controllers requesting an IP Address.
- How to assign a Galil Controller an IP Address.
- How to connect to a controller via IP Address.
- How to get information on a connected controller such as MAC Address and Serial Number.



## Prerequisites

A Galil controller connected to the same network as the host computer.

## Command Line Arguments

This example requires 2 arguments:

- The serial number of your controller. The value to use is the number after the prefix on the controller's serial number marking. For example, if the serial number is marked as *BV-1234*, the value to use for this argument is *1234*.
- A value between 1-254 that defines the last byte of the newly assigned IP Address. This example will assign an IP address that matches your computer's IP address, with the last byte changed. For example, if your IP address is *192.168.42.92* and *96* is specified, the controller will be assigned *192.168.42.96*. The example will ping the IP address to ensure that the IP address is not already taken.

```
ip_assigner_example.exe 1234 96
```

## 2.7 Motion Complete Example

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

## Concepts

This example demonstrates:

- How to connect to a controller via IP Address.
- How to issue basic commands.
- How to move the controller to a precise position.
- How to monitor the interrupts of the controller.

## Prerequisites

A Galil controller with two motors: one connected at the A axis and the other connected at the B axis.

## Command Line Arguments

This example requires 1 argument:

- The IP Address of a Galil controller.

```
motion_complete_example.exe 192.168.42.96
```

## Example Output

```
\*****\
Example GInterrupt() usage
\*****\
Position: 0, 0, 0, 0, 0, 0, 0, 0
Beginning independent motion...
Motion Complete on A and B
Position: 8000, 10000, 0, 0, 0, 0, 0, 0
```

## 2.8 Record Position Example

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

### Concepts

This example demonstrates:

- How to connect to a controller via IP Address.
- How to issue basic commands.
- How to manage a record array (RC/RD/RA) ring buffer.
- How to record position data and save to a text file.

### Prerequisites

A Galil controller with a motor connected at the A axis and B axis is needed for this example.

### Command Line Arguments

This example requires 3 arguments:

- The IP Address of a Galil controller.
- The path to a file to save Axis A positional data.
- The path to a file to save Axis B positional data.

```
record_position_example.exe 192.168.42.96 axis_a.csv axis_b.csv
```

## 2.9 Contour Example

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

## Concepts

This example demonstrates:

- How to connect to a controller via IP Address.
- How to issue basic commands.
- How to record position data and save to a text file.
- How to play back recorded data using contour mode.

## Prerequisites

A Galil controller with a motor connected at the A axis and the B axis is needed for this example.

## Command Line Arguments

This example requires 3 arguments:

- The IP Address of a Galil controller.
- The path to a csv file to store positional data for the A axis.
- The path to a csv file to store positional data for the B axis.

```
contour_example.exe 192.168.42.200 axis_a.csv axis_b.csv
```

## 2.10 Remote Server Example

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

## Concepts

This example demonstrates:

- How to advertise your gcaps server on the network for others to discover

## Prerequisites

This example works best in conjunction with the [Remote Client Example](#) running on a separate machine on the same network.

**Note**

Linux users will need to install the ncurses library.

**2.11 Remote Client Example**

C++	C#	Visual Basic
<a href="#">Example</a>	<a href="#">Example</a>	<a href="#">Example</a>
<a href="#">Logic</a>	<a href="#">Logic</a>	<a href="#">Logic</a>
<a href="#">Instructions</a>	<a href="#">Instructions</a>	<a href="#">Instructions</a>

We are always interested in what our customers would like to see! To request any new examples or supported languages, please email [support@galil.com](mailto:support@galil.com).

**Concepts**

This example demonstrates:

- How to discover other gcaps servers on your local network
- How to connect to other gcaps servers
- How to list available hardware on your connected server

**Prerequisites**

This example works best in conjunction with the [Remote Server Example](#) running on a separate machine on the same network.

**Note**

Linux users will need to install the ncurses library.

**Example Output**

Text colored green represents user input.

```
<s> List available servers on the network
<h> List available hardware on currently connected server
<0-9> Enter numbers 0-9 to connect to a server by index
<l> Set active server back to local server
<q> Quit
```

```
h
192.168.42.28, DMC4080 Rev 1.3c, Controller, 192.168.42.1
COM1
COM3
COM4
```

```
s
Available Servers:
<0> Example Server
```

```
0
Server set to: Example Server
```

```
h
/dev/ttyS0
/dev/ttyUSB0
```

```
l
Server set to: Local
```

## 2.12 C/C++

Please choose an operating system to get detailed instructions on how to build the gclib example projects.

- [Microsoft Windows](#)
- [Linux](#)

### 2.12.1 Microsoft Windows

#### Copy files

- Navigate to a convenient, empty, writable location, e.g. `C:\Users\{username}\Documents\Galil\cpp_←examples`.
- Copy the contents of `C:\Program Files (x86)\Galil\gclib\examples\cpp\examples` to this location.

#### Open Visual Studio Project

The following instructions were performed on *Visual Studio Professional 2017* and *Visual Studio Professional 2019* and can be extended to other Visual Studio versions. For brevity, the instructions assume the default installation location of `C:\Program Files (x86)\Galil\gclib` and a build type of `x64`.

- Launch *Visual Studio 2017* or *Visual Studio 2019*.
- Choose *File->Open->Project/Solution....*
- Navigate to the `examples.vcxproj` file in the `C:\Users\{username}\Documents\Galil\cpp_examples` directory.
- Click *Open*.
- In the *Solution Explorer* right-click on the `examples` project file, choose *Properties*.
  - Click the *Configuration Manager...* button.
    - \* Under *Active solution platform*: choose `x64`.
    - \* Click *Close*.
  - Highlight *Configuration Properties* in the side bar, and set the following project properties.
    - \* *Configuration Properties -> Debugging -> Environment* add **PATH=C:\Program Files (x86)\Galil\gclib\dll\x64;%PATH%**
    - \* Click *OK*.
- Many of the examples require command line arguments to execute. To enter command line arguments in Visual Studio:
  - In the *Solution Explorer* right-click on the `examples` project file, choose *Properties*.
    - \* Under *Configuration Properties*, highlight *Debugging* in the side bar. Enter the appropriate arguments in the *Command Arguments* box. Refer to each [example's landing page](#) for required command line arguments.
- Ensure the *Solution Configurations* and *Solution Platforms* are set to *Debug* and *x64* respectively.
- Hit *F5* to build and run the example.

## Run a Different Example

To run a different example, remove the current example from the solution and add the next example.

- In the *Solution Explorer* right-click on `commands_example.cpp`, choose *Remove*.
  - Click the *Remove* button.
- In the *Solution Explorer* right-click on the *examples* project file, choose *Add->Existing Item*.
  - Navigate to the desired example file and click *Add*.
- Hit *F5* to build and run the example.

### 2.12.2 Linux

## Copy examples to a temporary directory

```
cp -r /usr/share/gclib/doc/examples/cpp .
cd cpp
```

## Run Make

To build all examples:  
make

To build a single example:  
make commands.o

To run an example:  
./commands\_example.out

## 2.13 C#.NET

### Open Visual Studio Project

The following instructions were performed on *Visual Studio Professional 2017* and can be extended to other Visual Studio versions. For brevity, the instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib** and a build type of **x64**.

#### Copy files

- Navigate to a convenient, empty, writable location, e.g. **C:\Users\{username}\Documents\Galil\cs\_↔examples**.
- Copy the contents of **C:\Program Files (x86)\Galil\gclib\examples\cs\examples** to this location.

#### Configure Project

- Launch *Visual Studio 2017*.
- Choose *File->Open->Project/Solution....*
- Navigate to the `examples.sln` file in the **C:\Users\{username}\Documents\Galil\cs\_examples** directory.
- Click *Open*.

- In the *Solution Explorer* right-click on the *examples* project file, choose *Properties*.
  - Click the *Build* tab.
    - \* At the top of the window next to *Platform* choose *x64*.
    - \* Click *Save*.
    - \* Close the properties window.
  - In the *Solution Explorer* right-click on the *examples* project file, choose *Add->Existing Item*.
    - \* Navigate to the gclib C# wrapper at location **C:\Program Files (x86)\Galil\gclib\source\wrappers\cs** and select [gclib.cs](#).
    - \* Click *OK*.
- Many of the examples require command line arguments to execute. To enter command line arguments in Visual Studio:
  - In the *Solution Explorer* right-click on the *examples* project file, choose *Properties*.
    - \* Under *Debug*, enter the appropriate arguments in the *Command line arguments* box. Refer to each [example's landing page](#) for required command line arguments.
- Ensure the *Solution Configurations* and *Solution Platforms* are set to *Debug* and *x64* respectively.
- Hit *F5* to build and run the example.

## Run a Different Example

To run a different example, change the *Startup object* to the new example.

- In the *Solution Explorer* right-click on the *examples* project file, choose *Properties*.
  - Click the *Application* tab.
    - \* Under the *Startup object* dropdown, select a different example.
  - Click the *Debug* tab.
    - \* Enter the appropriate arguments in the *Command line arguments* box. Refer to each [example's landing page](#) for required command line arguments.
- Hit *F5* to build and run the example.

## 2.14 VB.NET

### Open Visual Studio Project

The following instructions were performed on *Visual Studio Professional 2017* and can be extended to other Visual Studio versions. For brevity, the instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib** and a build type of **x64**.

### Copy files

- Navigate to a convenient, empty, writable location, e.g. **C:\Users\{username}\Documents\Galil\vb\_↔  
examples**.
- Copy the contents of **C:\Program Files (x86)\Galil\gclib\examples\vb\examples** to this location.

## Configure Project

- Launch *Visual Studio 2017*.
- Choose *File->Open->Project/Solution....*
- Navigate to the `examples.sln` file in the `C:\Users\{username}\Documents\Galil\vb_examples` directory.
- Click *Open*.
- In the *Solution Explorer* right-click on the `examples` project file, choose *Properties*.
  - Click the *Compile* tab.
    - \* At the top of the window next to *Platform* choose *x64*.
    - \* Close the properties window.
  - In the *Solution Explorer* right-click on the `examples` project file, choose *Add->Existing Item*.
    - \* Navigate to the `gclib` VB wrapper at location `C:\Program Files (x86)\Galil\gclib\source\wrappers\vb` and select `gclib.vb`.
    - \* Click *OK*.
- Many of the examples require command line arguments to execute. To enter command line arguments in Visual Studio:
  - In the *Solution Explorer* right-click on the `examples` project file, choose *Properties*.
    - \* Under *Debug*, enter the appropriate arguments in the *Command line Arguments* box. Refer to each [example's landing page](#) for required command line arguments.
- Ensure the *Solution Configurations* and *Solution Platforms* are set to *Debug* and *x64* respectively.
- Hit *F5* to build and run the example.

## Run a Different Example

To run a different example, change the *Startup object* to the new example.

- In the *Solution Explorer* right-click on the `examples` project file, choose *Properties*.
  - Click the *Application* tab.
    - \* Under the *Startup object* dropdown, select a different example.
  - Click the *Debug* tab.
    - \* Enter the appropriate arguments in the *Command line arguments* box. Refer to each [example's landing page](#) for required command line arguments.
- Hit *F5* to build and run the example.



## Chapter 3

# Language Support

Below are a number of examples demonstrating how to use the library with various languages and on various platforms.

- [C/C++](#)
- [Python](#)
- [.Net](#)
- [Java](#)
- [LabVIEW](#)

Can't find what you need? Please email [support@galil.com](mailto:support@galil.com), or call [Galil Applications](#).

### 3.1 C/C++

- [Microsoft Visual Studio 2019 \(16.0\)](#)
- [Microsoft Visual Studio 2017 \(15.0\)](#)
- [Microsoft Visual Studio 2015 \(14.0\)](#)
- [Microsoft Visual Studio 2013 \(12.0\)](#)
- [MinGW](#)
- [Borland C++](#)
- [gcc \(Linux\)](#)
- [clang \(OS X\)](#)

#### 3.1.1 Microsoft Visual Studio 2019 (16.0)

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

#### **[x\\_simple.c](#) from *VS2019 x64 Native Tools Command Prompt***

Open *x64 Native Tools Command Prompt for VS 2019*.

## Copy files

Navigate to a convenient, empty, writable location.

### Set an environment variable for the base path.

```
>set base=C:\Program Files (x86)\Galil\gclib
```

### Copy simple example

```
>copy "%base%\examples\cpp\x_simple.c" .
```

### Edit [GOpen\(\)](#) call as necessary

In a text editor, open [x\\_simple.c](#). Find the [GOpen\(\)](#) call and update the address to match the desired hardware. See the documentation for [GOpen\(\)](#) for address formatting options.

## Compile

```
>cl x_simple.c "%base%\lib\dynamic\x64\*.lib" -I "%base%\include"
```

### Set Path to DLL

```
>set PATH=%base%\dll\x64\;%PATH%
```

## Execute

```
>x_simple.exe
version: 211.211.211 1.0.0.128
info: 192.168.0.42, DMCC640 Rev 1.0g, 9999
response: 3757802.0000
:
```

## Using the pre-configured MSVC project ([x\\_examples.cpp](#))

The directory `gclib\examples\msvc` has fully functional MSVC examples. These instructions detail how to use the 2019 version.

- Copy `gclib\examples\msvc\2019_16.0\gclib_example` to a convenient, writable location.
- Run `gclib_example\gclib_example\copy_source.bat` to copy the files.
- Open `gclib_example\gclib_example.sln` in Visual Studio 2019.
- In the *Solution Explorer*, expand the `gclib_example` and expand *Source Files* to show a listing of source.
- Open [x\\_examples.cpp](#).
- Find the [GOpen\(\)](#) call and update the address to match the desired hardware. See the documentation for [GOpen\(\)](#) for address formatting options.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit *F5* to build and run the example.

## Create Project with MSVC 2019 (x\_examples.cpp)

The instructions below allow building a project from scratch.

The following instructions were performed on *Visual Studio Professional 2019* and can be extended to other Visual Studio versions. For brevity, the instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib** and a build type of **x64**.

- Launch *Visual Studio 2019*.
- At the initial start window, Choose *Create a new project*.
- In the *Create a new project* window, choose *Empty Project* and click *Next*.
- Choose a Name, e.g. **gclib\_example**.
- Choose a Location, e.g. *C:\Users\user\Desktop*.
- Uncheck *Place solution and project in the same directory*.
- Click *Create*.
- In the *Solution Explorer*, right-click on *Source Files* and choose *Add->Existing Item*.
  - Navigate to the gclib installation directory, then to *examples\cpp* in the installation directory.
  - In *File Name* type **x\_\*.cpp** and click *Add*, this will filter out the files needed
  - Select all files in the file chooser and click *Add*.
- In the *Solution Explorer* right-click on *gclib\_example*, choose *Properties*.
  - Click the *Configuration Manager...* button.
    - \* Under *Active solution platform*: choose *x64*.
    - \* Click *Close*.
  - Highlight *Configuration Properties* in the side bar, and set the following project properties.
    - \* At the top of the window, change *Configuration*: to *All Configurations* and ensure *Platform* lists *Active(x64)*.
    - \* *Configuration Properties -> C/C++ -> General -> Additional Include Directories* add **C:\Program Files (x86)\Galil\gclib\include**
    - \* *Configuration Properties -> Linker -> General -> Additional Library Directories* add **C:\Program Files (x86)\Galil\gclib\lib\dynamic\x64**
    - \* *Configuration Properties -> Linker -> Input -> Additional Dependencies* add **gclib.lib;gclibo.lib;{rest of text}** where {rest of text} is the original string that was in the cell. Note the semicolons between library files.
    - \* *Configuration Properties -> Debugging -> Environment* add **PATH=C:\Program Files (x86)\Galil\gclib\dll\x64;%PATH%**
    - \* Click *OK*.
- In the *Solution Explorer* open *x\_examples.cpp*. Find the *GOpen()* call and update the address to match the desired hardware. See the documentation for *GOpen()* for address formatting options.
- Find the `#if 0` preprocessor blocks enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit *F5* to build and run the example.

### 3.1.2 Microsoft Visual Studio 2017 (15.0)

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

## [x\\_simple.c](#) from *VS2017 x64 Native Tools Command Prompt*

Open *x64 Native Tools Command Prompt for VS 2017*.

### Copy files

Navigate to a convenient, empty, writable location.

### Set an environment variable for the base path.

```
>set base=C:\Program Files (x86)\Galil\gclib
```

### Copy simple example

```
>copy "%base%\examples\cpp\x_simple.c" .
```

### Edit [GOpen\(\)](#) call as necessary

In a text editor, open [x\\_simple.c](#). Find the [GOpen\(\)](#) call and update the address to match the desired hardware. See the documentation for [GOpen\(\)](#) for address formatting options.

### Compile

```
>cl x_simple.c "%base%\lib\dynamic\x64\*.lib" -I "%base%\include"
```

### Set Path to DLL

```
>set PATH=%base%\dll\x64\;%PATH%
```

### Execute

```
>x_simple.exe
version: 211.211.211 1.0.0.128
info: 192.168.0.42, DMCC640 Rev 1.0g, 9999
response: 3757802.0000
:
```

## Using the pre-configured MSVC project ([x\\_examples.cpp](#))

The directory *gclib\examples\msvc* has fully functional MSVC examples. These instructions detail how to use the 2017 version.

- Copy *gclib\examples\msvc\2017\_15.0\gclib\_example* to a convenient, writable location.
- Run *gclib\_example\gclib\_example\copy\_source.bat* to copy the files.
- Open *gclib\_example\gclib\_example.sln* in Visual Studio 2017.
- In the *Solution Explorer*, expand the *gclib\_example* and expand *Source Files* to show a listing of source.
- Open [x\\_examples.cpp](#).
- Find the [GOpen\(\)](#) call and update the address to match the desired hardware. See the documentation for [GOpen\(\)](#) for address formatting options.

- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit *F5* to build and run the example.

## Create Project with MSVC 2017 ([x\\_examples.cpp](#))

The instructions below allow building a project from scratch.

The following instructions were performed on *Visual Studio Professional 2017* and can be extended to other Visual Studio versions. For brevity, the instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib** and a build type of **x64**.

- Launch *Visual Studio 2017*.
- Choose *File->New->Project*.
- In the *New Project* dialog, choose *Visual C++->Empty Project*.
- Choose a Name, e.g. **gclib\_example**.
- Choose a Location, e.g. *C:\Users\user\Desktop*.
- Check *Create directory for solution*.
- Click *OK*.
- In the *Solution Explorer*, right-click on *Source Files* and choose *Add->Existing Item*.
  - Navigate to the gclib installation directory, then to *examples\cpp* in the installation directory.
  - In *File Name* type **x\_\*.cpp** and click *Add*, this will filter out the files needed
  - Select all files in the file chooser and click *Add*.
- In the *Solution Explorer* right-click on *gclib\_example*, choose *Properties*.
  - Click the *Configuration Manager...* button.
    - \* Under *Active solution platform*: choose **x64**.
    - \* Click *Close*.
  - Highlight *Configuration Properties* in the side bar, and set the following project properties.
    - \* At the top of the window, change *Configuration:* to *All Configurations* and ensure *Platform* lists *Active(x64)*.
    - \* *Configuration Properties -> C/C++ -> General -> Additional Include Directories* add **C:\Program Files (x86)\Galil\gclib\include**
    - \* *Configuration Properties -> Linker -> General -> Additional Library Directories* add **C:\Program Files (x86)\Galil\gclib\lib\dynamic\x64**
    - \* *Configuration Properties -> C/C++ -> Code Generation -> Spectre Mitigation* set to **Disabled**. If your application will cross trust boundaries, consider Spectre and Meltdown vulnerabilities before deploying.
    - \* *Configuration Properties -> Linker -> Input -> Additional Dependencies* add **gclib.lib;gclibo.lib**;{rest of text} where {rest of text} is the original string that was in the cell. Note the semicolons between library files.
    - \* *Configuration Properties -> Debugging -> Environment* add **PATH=C:\Program Files (x86)\Galil\gclib\dll\x64;%PATH%**
    - \* Click *OK*.
- In the *Solution Explorer* open [x\\_examples.cpp](#). Find the [GOpen\(\)](#) call and update the address to match the desired hardware. See the documentation for [GOpen\(\)](#) for address formatting options.

- Find the `#if 0` preprocessor blocks enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit `F5` to build and run the example.

### 3.1.3 Microsoft Visual Studio 2015 (14.0)

For brevity, these instructions assume the default installation location of `C:\Program Files (x86)\Galil\gclib`.

#### [x\\_simple.c](#) from *VS2015 x64 Native Tools Command Prompt*

Open *VS2015 x64 Native Tools Command Prompt*.

#### Copy files

Navigate to a convenient, empty, writable location.

#### Set an environment variable for the base path.

```
>set base=C:\Program Files (x86)\Galil\gclib
```

#### Copy simple example

```
>copy "%base%\examples\cpp\x_simple.c" .
```

#### Edit [GOpen\(\)](#) call as necessary

In a text editor, open [x\\_simple.c](#). Find the [GOpen\(\)](#) call and update the address to match the desired hardware. See the documentation for [GOpen\(\)](#) for address formatting options.

#### Compile

```
>cl x_simple.c "%base%\lib\dynamic\x64\*.lib" -I "%base%\include"
```

#### Set Path to DLL

```
>set PATH=%base%\dll\x64;%PATH%
```

#### Execute

```
>x_simple.exe
version: 211.211.211 1.0.0.128
info: 192.168.0.42, DMCC640 Rev 1.0g, 9999
response: 3757802.0000
:
```

## Using the pre-configured MSVC project ([x\\_examples.cpp](#))

The directory `gclib\examples\msvc` has fully functional MSVC examples. These instructions detail how to use the 2015 version.

- Copy `gclib\examples\msvc\2015_14.0\gclib_example` to a convenient, writable location.
- Run `gclib_example\gclib_example\copy_source.bat` to copy the files.
- Open `gclib_example\gclib_example.sln` in Visual Studio 2015.
- In the *Solution Explorer*, expand the `gclib_example` and expand *Source Files* to show a listing of source.
- Open [x\\_examples.cpp](#).
- Find the `GOpen()` call and update the address to match the desired hardware. See the documentation for [GOpen\(\)](#) for address formatting options.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit *F5* to build and run the example.

## Create Project with MSVC 2015 ([x\\_examples.cpp](#))

The instructions below allow building a project from scratch.

The following instructions were performed on *Visual Studio Professional 2015* and can be extended to other Visual Studio versions. For brevity, the instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib** and a build type of **x64**.

- Launch *Visual Studio 2015*.
- Choose *File->New->Project*.
- In the *New Project* dialog, choose *Visual C++->Empty Project*.
- Choose a Name, e.g. **gclib\_example**.
- Choose a Location, e.g. `C:\Users\user\Desktop`.
- Check *Create directory for solution*.
- Click *OK*.
- In the *Solution Explorer*, right-click on *Source Files* and choose *Add->Existing Item*.
  - Navigate to the gclib installation directory, then to `examples\cpp` in the installation directory.
  - In *File Name* type **x\_\*.cpp** and click *Add*, this will filter out the files needed
  - Select all files in the file chooser and click *Add*.
- In the *Solution Explorer* right-click on `gclib_example`, choose *Properties*.
  - Click the *Configuration Manager...* button.
    - \* Under *Active solution platform*: choose **x64**.
    - \* Click *Close*.
  - Highlight *Configuration Properties* in the side bar, and set the following project properties.
    - \* At the top of the window, change *Configuration:* to *All Configurations* and ensure *Platform* lists **Active(x64)**.

- \* *Configuration Properties* -> *C/C++* -> *Additional Include Directories* add **C:\Program Files (x86)\Galil\gclib\include**
  - \* *Configuration Properties* -> *Linker* -> *General* -> *Additional Library Directories* add **C:\Program Files (x86)\Galil\gclib\lib\dynamic\x64**
  - \* *Configuration Properties* -> *Linker* -> *Input* -> *Additional Dependencies* add **gclib.lib;gclibo.lib;{rest of text}** where {rest of text} is the original string that was in the cell. Note the semicolons between library files.
  - \* *Configuration Properties* -> *Debugging* -> *Environment* add **PATH=C:\Program Files (x86)\Galil\gclib\dll\x64;%PATH%**
  - \* Click *OK*.
- In the *Solution Explorer* open [x\\_examples.cpp](#). Find the [GOpen\(\)](#) call and update the address to match the desired hardware. See the documentation for [GOpen\(\)](#) for address formatting options.
  - Find the `#if 0` preprocessor blocks enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
  - Hit *F5* to build and run the example.

### 3.1.4 Microsoft Visual Studio 2013 (12.0)

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

#### [x\\_simple.c](#) from *VS2013 x64 Native Tools Command Prompt*

Open *VS2013 x64 Native Tools Command Prompt*.

#### Copy files

Navigate to a convenient, empty, writable location, e.g. *C:\temp*.

#### Set an environment variable for the base path.

```
C:\temp>set base=C:\Program Files (x86)\Galil\gclib
```

#### Copy simple example

```
C:\temp>copy "%base%\examples\cpp\x_simple.c" .
```

#### Edit [GOpen\(\)](#) call as necessary

In a text editor, open [x\\_simple.c](#). Find the [GOpen\(\)](#) call and update the address to match the desired hardware. See the documentation for [GOpen\(\)](#) for address formatting options.

#### Compile

```
C:\temp>cl x_simple.c "%base%\lib\dynamic\x64\*.lib" -I "%base%\include"
```

#### Set Path to DLL

```
C:\temp>set PATH=%base%\dll\x64;%PATH%
```



## Execute

```
C:\temp>x_simple.exe
rc: 0
version: 85.60.138
rc: 0
rc: 0
info: 10.1.3.17, DMC4020 Rev 1.2b, 291
rc: 0
response: 357247808.0000
:
```

## Using the pre-configured MSVC project ([x\\_examples.cpp](#))

The directory `gclib\examples\msvc` has fully functional MSVC examples. These instructions detail how to use the 2013 version.

- Copy `gclib\examples\msvc\2013_12.0\gclib_example` to a convenient, writable location, e.g. `C:\temp`.
- Run `gclib_example\gclib_example\copy_source.bat` to copy the files.
- Open `gclib_example\gclib_example.sln` in Visual Studio 2013.
- In the *Solution Explorer*, expand the `gclib_example` and expand *Source Files* to show a listing of source.
- Open [x\\_examples.cpp](#)
- Find the `GOpen()` call and update the address to match the desired hardware. See the documentation for `GOpen()` for address formatting options.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit `F5` to build and run the example.

## Create Project with MSVC 2013 ([x\\_examples.cpp](#))

The instructions below allow building a project from scratch.

The following instructions were performed on *Visual Studio Professional 2013* and can be extended to other Visual Studio versions. For brevity, the instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib** and a build type of **x86 (win32)**.

- Launch *Visual Studio 2013*
- Choose *File->New->Project*
- In the *New Project* dialog, choose *Visual C++->Empty Project*
- Choose a Name, e.g. **gclib\_example**
- Choose a Location, e.g. `C:\Users\user\Desktop`
- Check *Create directory for solution*
- Click *OK*
- In the *Solution Explorer*, right-click on *Source Files* and choose *Add->Existing Item*
  - Navigate to the `gclib` installation directory, then to `examples\cpp` in the installation directory
  - In *File Name* type `x_*.cpp` and click *Add*, this will filter out the files needed

- Select all files in the file chooser and click *Add*
- In the *Solution Explorer* right-click on *gclib\_example*, choose *Properties*, highlight *Configuration Properties*, and set the following project properties
  - At the top of the window, change *Configuration:* to *All Configurations* and ensure *Platform* lists *Active(← Win32)*
  - *Configuration Properties* -> *C/C++* -> *Additional Include Directories* add **C:\Program Files (x86)\Galil\gclib\include**
  - *Configuration Properties* -> *Linker* -> *General* -> *Additional Library Directories* add **C:\Program Files (x86)\Galil\gclib\lib\dynamic\x86**
  - *Configuration Properties* -> *Linker* -> *Input* -> *Additional Dependencies* add **gclib.lib;gclibo.← lib;{rest of text}** where {rest of text} is the original string that was in the cell. Note the semicolons between library files.
  - *Configuration Properties* -> *Debugging* -> *Environment* add **PATH=C:\Program Files (x86)\Galil\gclib\dll\x86;%PATH%**
- In the *Solution Explorer* open *x\_examples.cpp*. Find the **GOpen()** call and update the address to match the desired hardware. See the documentation for **GOpen()** for address formatting options.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit *F5* to build and run the example.

### 3.1.5 MinGW

The following instructions were performed with x86 Minimalist GNU for Windows (MinGW) installed from <http://mingw-w64.sourceforge.net/download.php#mingw-builds>

For brevity, these instructions assume the default installation location of "C:\Program Files (x86)\Galil\gclib".

## Copy Files

Copy "gclib\examples\mingw" to a convenient, writable location, e.g. "C:\temp". Run `C:\temp\mingw\copy←_source.bat` to copy all files.

## x\_simple.c

### Edit **GOpen()** call as necessary

In a text editor, open *x\_simple.c*. Find the **GOpen()** call and update the address to match the desired hardware. See the documentation for **GOpen()** for address formatting options.

## Compile

- Launch the MinGW terminal, e.g. *Start* -> *All Programs* -> *MinGW-W64 project* -> *i686-4.9.1-posix-dwarf-rt\_v3-rev3* -> *Run Terminal*.
- Navigate to the directory with the files above.
- Compile the code.

```
C:\temp\mingw>gcc x_simple.c -L. -lgclibo -lgclib -o simple.exe
```

## Execute

```
C:\temp\mingw>simple.exe
rc: 0
version: 85.60.138
rc: 0
rc: 0
info: 10.1.3.17, DMC4020 Rev 1.2b, 291
rc: 0
response: 1584328.0000
:
```

## x\_examples.cpp

### Review and Modify source

- In a text editor, open [x\\_examples.cpp](#). Find the `GOpen()` call and update the address to match the desired hardware. See the documentation for `GOpen()` for address formatting options.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.

### Compile

- Launch the MinGW terminal, e.g. *Start -> All Programs -> MinGW-W64 project -> i686-4.9.1-posix-dwarf-rt\_v3-rev3 -> Run Terminal*.
- Navigate to the directory with the files above.
- Compile the code.

```
C:\temp\mingw>g++ *.cpp -L. -lgclibo -lgclib -o examples.exe
```

## Execute

```
C:\temp\mingw>examples.exe
Library version: 41.35.34

192.168.0.43, DMC4020 Rev 1.2b, 291

*****
Example GRead() and GWrite() usage
*****

Read 155 QR bytes.

*****
Example GCommand() usage
*****
Revision report, ^R^V
DMC4020 Rev 1.2b
:

Command Values
val is 10
val is 11
val is 3.1415
val is 9.869

Command Trimming
> 95653016.0000
```



```

*****
Example GInterrupt() usage
*****
"UI 8" executed.

*****
Example GMotionComplete() usage
*****

Position: 0, 0
Beginning independent motion... Motion Complete on A
Position: 8000, 0

Position: 0, 0
Beginning vector motion... Motion Complete on vector plane S
Position: 6000, 0

examples.cpp executed OK
main() is finished. Press Enter to exit:

```

### 3.1.6 Borland C++

The following instructions were performed on:

Embarcadero C++ 7.10 for Win32 Copyright (c) 1993-2015 Embarcadero Technologies, Inc.

For brevity, these instructions assume the default installation location of "C:\Program Files (x86)\Galil\gclib".

## Copy Files

Copy "gclib\examples\borland" to a convenient, writable location, e.g. "C:\temp". Run C:\temp\borland\copy←\_source.bat to copy all files.

```

C:\temp>cd borland

C:\temp\borland>copy_source.bat
\Program Files (x86)\Galil\gclib\examples\cpp\x_arrays.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_examples.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_examples.h
\Program Files (x86)\Galil\gclib\examples\cpp\x_gcommand.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_ginterrupt.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_gmessage.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_gmotioncomplete.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_gread_gwrite.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_grecord.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_nonblocking.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_programs.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_simple.c
    12 file(s) copied.
\Program Files (x86)\Galil\gclib\include\gclib.h
\Program Files (x86)\Galil\gclib\include\gclibo.h
\Program Files (x86)\Galil\gclib\include\gclib_errors.h
\Program Files (x86)\Galil\gclib\include\gclib_record.h
    4 file(s) copied.
\Program Files (x86)\Galil\gclib\lib\dynamic\x86\gclib.lib
\Program Files (x86)\Galil\gclib\lib\dynamic\x86\gclibo.lib
    2 file(s) copied.
\Program Files (x86)\Galil\gclib\dll\x86\gclib.dll
\Program Files (x86)\Galil\gclib\dll\x86\gclibo.dll
    2 file(s) copied.

C:\temp\borland>

```

## Modify Path

- Add Borland's compiler to the PATH variable.

```
C:\temp\borland>set PATH=c:\Program Files (x86)\Embarcadero\Studio\17.0\bin;%PATH%
```

## Convert lib files

```
C:\temp\borland>move gclib.lib _gclib.lib
1 file(s) moved.
```

```
C:\temp\borland>move gclibo.lib _gclibo.lib
1 file(s) moved.
```

```
C:\temp\borland>coff2omf.exe _gclib.lib gclib.lib
COFF to OMF Converter Version 1.2.0 Copyright (c) 1999-2009 Embarcadero Technologies, Inc.
All rights reserved.
```

```
C:\temp\borland>coff2omf.exe _gclibo.lib gclibo.lib
COFF to OMF Converter Version 1.2.0 Copyright (c) 1999-2009 Embarcadero Technologies, Inc.
All rights reserved.
```

## x\_simple.c

### Edit `GOpen()` call as necessary

In a text editor, open `x_simple.c`. Find the `GOpen()` call and update the address to match the desired hardware. See the documentation for `GOpen()` for address formatting options.

### Compile

```
C:\temp\borland>bcc32 gclib.lib gclibo.lib x_simple.c
Embarcadero C++ 7.10 for Win32 Copyright (c) 1993-2015 Embarcadero Technologies, Inc.
x_simple.c:
Turbo Incremental Link 6.72 Copyright (c) 1997-2015 Embarcadero Technologies, Inc.
```

### Execute

```
C:\temp\borland>x_simple.exe
version: 130.115.279
info: 192.168.0.43, DMC4143 Rev 1.2b, 9998
response: 61016.0000
:
```

## x\_examples.cpp

### Review and Modify source

- In a text editor, open `x_examples.cpp`. Find the `GOpen()` call and update the address to match the desired hardware. See the documentation for `GOpen()` for address formatting options.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.

## Compile

```
C:\temp\borland>bcc32 -c *.cpp
```

## Link

```
C:\temp\borland>bcc32 -o examples.exe *.obj gclib.lib gclibo.lib
```

## Execute

```
C:\temp\borland>examples.exe
Library version: 130.115.279
```

```
192.168.0.43, DMC4020 Rev 1.2b, 291
```

```
*****
Example GRead() and GWrite() usage
*****
```

```
Read 155 QR bytes.
```

```
*****
Example GCommand() usage
*****
```

```
Revision report, ^R^V
```

```
DMC4020 Rev 1.2b
```

```
:
```

```
Command Values
```

```
val is 10
```

```
val is 11
```

```
val is 3.1415
```

```
val is 9.869
```

```
Command Trimming
```

```
> 95653016.0000
```

```
:<
```

```
> 95653016.0000<
```

```
>95653016.0000<
```

```
Receiving Binary Data
```

```
QR read 155 bytes
```

```
Error handling
```

```
QD correctly trapped, not allowed, try GArrayDownload()
```

```
DL correctly trapped, not allowed, try GProgramDownload()
```

```
Modifying timeout
```

```
Burning program...OK
```

```
*****
Example GProgramDownload() and GProgramUpload() usage
*****
```

```
GProgramDownload() correctly errored. Can't fit with level 3 compression
Program Downloaded with compression level 4
```

```
Uploading program:
```

```
#A;i=0;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i
i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;EN
```

```
Program executed as expected
```

```
*****
Example GArrayDownload() and GArrayUpload() usage
*****
```

```
2.0000, 4.0000, 6.0000, 8.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.000
0000
```

```
2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.000
0000
```

```
3.0000, 5.0000, 10.0000
```

```
*****
Example GRecord() usage
*****
```

```
QR-based data record
38564
393216000
```

```
DR-based data record
38670
38772
38874
38976
39078
39180
39282
39384
39486
39588
39690
```

```
QR-based data record with offsets
39692
39692
```

```
*****
Example GMessage() usage
*****
```

```
0.0000
1.0000
2.0000
3.0000
4.0000
5.0000
6.0000
7.0000
8.0000
9.0000
```

```
*****
Example GInterrupt() usage
*****
```

```
"UI 8" executed.
```

```
*****
Example GMotionComplete() usage
*****
```

```
Position: 0, 0
Beginning independent motion... Motion Complete on A
Position: 8000, 0
```

```
Position: 0, 0
Beginning vector motion... Motion Complete on vector plane S
Position: 6000, 0
```

```
examples.cpp executed OK
main() is finished. Press Enter to exit:
```

### 3.1.7 gcc (Linux)

The following instructions were performed on

```
$ uname -a
Linux localhost.localdomain 3.17.4-301.fc21.x86_64 #1 SMP Thu Nov 27 19:09:10 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
$ g++ --version
g++ (GCC) 4.9.2 20150212 (Red Hat 4.9.2-6)
```



## Copy Files

```
$ cp -r /usr/share/gclib/doc/examples/cpp/x_examples .
$ cd x_examples
$ ls
x_arrays.cpp      x_gcommand.cpp    x_gmotioncomplete.cpp  x_programs.cpp
x_examples.cpp   x_ginterrupt.cpp  x_gread_gwrite.cpp     x_simple.c
x_examples.h     x_gmessage.cpp    x_grecord.cpp
```

## x\_simple.c

- In a text editor, open [x\\_simple.c](#). Find the `GOpen()` call and update the address to match the desired hardware. See the documentation for `GOpen()` for address formatting options.

## Compile

```
$ gcc -Wall -Werror x_simple.c -lgclib -lgclibo -o simple
```

## Run

```
$ ./simple
rc: 0
version: 85.60.131
rc: 0
rc: 0
info: 10.1.3.17, DMC4020 Rev 1.2b, 291
rc: 0
response: 179340166.0000
:
```

## x\_examples.cpp

- In a text editor, open [x\\_examples.cpp](#). Find the `GOpen()` call and update the address to match the desired hardware. See the documentation for `GOpen()` for address formatting options. Don't forget `-s ALL` if data records, interrupts, and messages are to be tested.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.

## Compile

```
$ g++ x_*.cpp -lgclib -lgclibo -o example
```

## Run

```
$/example Library version: 85.60.131
```

```
10.1.3.17, DMC4020 Rev 1.2b, 291
```

```
*****
Example GRead() and GWrite() usage
*****
```

```
Read 155 QR bytes.
```

```

*****
Example GCommand() usage
*****
Revision report, ^R^V
DMC4020 Rev 1.2b
:

Command Values
val is 10
val is 11
val is 3.1415
val is 9.869

Command Trimming
> 179798738.0000
:<
> 179798738.0000<
>179798738.0000<

Receiving Binary Data
QR read 155 bytes

Error handling
QD correctly trapped, not allowed, try GArrayDownload()
DL correctly trapped, not allowed, try GProgramDownload()

Modifying timeout
Burning program...OK

*****
Example GProgramDownload() and GProgramUpload() usage
*****
GProgramDownload() correctly errored. Can't fit with level 3 compression
Program Downloaded with compression level 4
Uploading program:
#A;i=0;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1
i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;EN

Program executed as expected
*****
Example GArrayDownload(), GArrayUploadFile()
GArrayDownloadFile(), and GArrayUpload usage
*****
2.0000, 4.0000, 6.0000, 8.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000

2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000

3.0000, 5.0000, 10.0000
2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000

*****
Example GRecord() usage
*****

QR-based data record
36100
6000

DR-based data record
36204
36306
36408
36510
36612
36714
36816
36918
37020
37122
37224

QR-based data record with offsets

```

```
37224
37224
```

```
*****
Example GMessage() usage
*****
0.0000
1.0000
2.0000
3.0000
4.0000
5.0000
6.0000
7.0000
8.0000
9.0000

*****
Example GInterrupt() usage
*****
"UI 8" executed.

*****
Example GMotionComplete() usage
*****

Position: 0, 0
Beginning independent motion... Motion Complete on A
Position: 8000, 0

Position: 0, 0
Beginning vector motion... Motion Complete on vector plane S
Position: 6000, 0

examples.cpp executed OK
main() is finished. Press Enter to exit:
```

### 3.1.8 clang (OS X)

The following instructions were performed on

```
$ sw_vers
ProductName: Mac OS X
ProductVersion: 10.10.5
BuildVersion: 14F27
$ gcc --version
Configured with: --prefix=/Library/Developer/CommandLineTools/usr --with-gxx-include-dir=/usr/include/c++/4.2.
Apple LLVM version 6.1.0 (clang-602.0.53) (based on LLVM 3.6.0svn)
Target: x86_64-apple-darwin14.5.0
Thread model: posix
```

## Copy Files

```
$ cd ~
$ mkdir test
$ cd test
$ tar -xzf /Applications/gclib/examples/gclib_examples.tar.gz
$ cp /Applications/gclib/include/* .
$ cp /Applications/gclib/dylib/* .
$ ls
gclib.0.dylib  x_arrays.cpp      x_gmotioncomplete.cpp
gclib.h        x_examples.cpp    x_gread_gwrite.cpp
gclib_errors.h x_examples.h      x_grecord.cpp
gclib_record.h x_gcommand.cpp    x_nonblocking.cpp
gclibo.0.dylib x_ginterrupt.cpp  x_programs.cpp
gclibo.h       x_gmessage.cpp    x_simple.c
```

## x\_simple.c

- In a text editor, open [x\\_simple.c](#). Find the `GOpen()` call and update the address to match the desired hardware. See the documentation for `GOpen()` for address formatting options.

### Compile

```
$ gcc -Wall -Werror x_simple.c gclib.0.dylib gclibo.0.dylib -o simple
```

### Run

```
$ ./simple
rc: 0
version: 126.108.229
rc: 0
rc: 0
info: 10.1.3.142, DMC4020 Rev 1.2a-BH, 291
rc: 0
response: 206676.0000
:
```

## x\_examples.cpp

- In a text editor, open [x\\_examples.cpp](#). Find the `GOpen()` call and update the address to match the desired hardware. See the documentation for `GOpen()` for address formatting options. Don't forget `-s ALL` if data records, interrupts, and messages are to be tested.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.

### Compile

```
$ g++ x_*.cpp gclib.0.dylib gclibo.0.dylib -o example
```

### Run

```
$ ./example
Library version: 126.108.229

10.1.3.142, DMC4020 Rev 1.2a-BH, 291

*****
Example GRead() and GWrite() usage
*****

Read 1 byte(s)
:
Program test OK.

*****
Example GCommand() usage
*****
Revision report, ^R^V
DMC4020 Rev 1.2a-BH
:
```

## Command Values

```
val is 10
val is 11
val is 3.1415
val is 9.869
```

## Command Trimming

```
> 408978.0000
:<
> 408978.0000<
>408978.0000<
```

## Receiving Binary Data

```
QR read 155 bytes
```

## Error handling

```
QD correctly trapped, not allowed, try GArrayDownload()
DL correctly trapped, not allowed, try GProgramDownload()
```

## Modifying timeout

```
Burning program...OK
```

```
*****
Example GProgramDownload() and GProgramUpload() usage
```

```
*****
GProgramDownload() correctly errored. Can't fit with level 3 compression
Program Downloaded with compression level 4
```

```
Uploading program:
```

```
#A;i=0;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1
i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;EN
```

```
Program executed as expected
```

```
*****
Example GArrayDownload(), GArrayUploadFile()
GArrayDownloadFile(), and GArrayUpload usage
```

```
*****
2.0000, 4.0000, 6.0000, 8.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000
```

```
2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000
```

```
3.0000, 5.0000, 10.0000
```

```
2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000
```

```
*****
Example GRecord() usage
```

```
*****
```

## QR-based data record

```
18358
0
```

## DR-based data record

```
18462
18564
18666
18768
18870
18972
19074
19176
19278
19380
19482
```

## QR-based data record with offsets

```
19482
19482
```

```
*****
Example GMessage() usage
```

```
*****
0.0000
```

```

1.0000
2.0000
3.0000
4.0000
5.0000
6.0000
7.0000
8.0000
9.0000

*****
Example GInterrupt() usage
*****
"UI 8" executed.

*****
Example GMotionComplete() usage
*****

Position: 0, 0
Beginning independent motion... Motion Complete on A
Position: 8000, 0

Position: 0, 0
Beginning vector motion... Motion Complete on vector plane S
Position: 6000, 0

*****
Example GMessage non-blocking usage
*****
422902.0000

*****
Example GInterrupt non-blocking usage
*****
F1

*****
Example GRecord non-blocking usage
*****
33786

examples.cpp executed OK
main() is finished. Press Enter to exit:

```

## 3.2 Python

The gclib Python wrapper assumes the default gclib installation location.

### Install Python

- See <https://www.python.org/> if Python is not already installed on the system. The gclib Python wrapper supports Python versions 2 and 3.
- On Windows, choose to add Python to the environment variable during installation. This allows Python to be invoked from the command line.

## Install the gclib Python module

### Windows

- Type the following commands into a command prompt.

```
>cd %temp%
>mkdir py
>cd py
>copy "c:\Program Files (x86)\Galil\gclib\source\wrappers\python\*" .
c:\Program Files (x86)\Galil\gclib\source\wrappers\python\gclib.py
c:\Program Files (x86)\Galil\gclib\source\wrappers\python\setup.py
2 file(s) copied.
>copy "c:\Program Files (x86)\Galil\gclib\examples\python\*" .
c:\Program Files (x86)\Galil\gclib\examples\python\example.py
1 file(s) copied.
>python setup.py install
running install
running build
running build_py
creating build
creating build\lib
copying gclib.py -> build\lib
running install_lib
running install_egg_info
Removing C:\Users\user\AppData\Local\Programs\Python\Python37-32\Lib\site-packages\gclib-1.0-py3.7.egg-info
Writing C:\Users\user\AppData\Local\Programs\Python\Python37-32\Lib\site-packages\gclib-1.0-py3.7.egg-info
```

- The gclib Python wrapper is now installed. Go to the next section, **Using gclib from the Python Interpreter**.

### Linux

- Type the following commands into a terminal prompt.

```
$ mkdir py
$ cd py
$ cp /usr/share/gclib/src/wrappers/python/* .
$ cp /usr/share/gclib/doc/examples/python/* .
$ sudo python setup.py install
[sudo] password for user:
running install
running build
running build_py
creating build
creating build/lib
copying gclib.py -> build/lib
running install_lib
copying build/lib/gclib.py -> /usr/lib/python2.7/site-packages
byte-compiling /usr/lib/python2.7/site-packages/gclib.py to gclib.pyc
running install_egg_info
Writing /usr/lib/python2.7/site-packages/gclib-1.0-py2.7.egg-info
```

- The gclib Python wrapper is now installed. Go to the next section, **Using gclib from the Python Interpreter**.

### OS X

- Be sure that the *Create Environment Variable* step has been followed in the [OS X installation instructions](#).
- Type the following commands into a Terminal prompt.

```
$ mkdir ~/python_temp
$ cd ~/python_temp/
$ tar -xvf /Applications/gclib/source/gclib_python.tar.gz
```

```
x gclib.py
x setup.py
$ tar -xvf /Applications/gclib/examples/gclib_python_examples.tar.gz
x example.py
$ sudo python setup.py install
running install
running build
running build_py
creating build
creating build/lib
copying gclib.py -> build/lib
running install_lib
copying build/lib/gclib.py -> /Library/Python/2.7/site-packages
byte-compiling /Library/Python/2.7/site-packages/gclib.py to gclib.pyc
running install_egg_info
Writing /Library/Python/2.7/site-packages/gclib-1.0-py2.7.egg-info
```

- The gclib Python wrapper is now installed. Go to the next section, **Using gclib from the Python Interpreter**.

## Using gclib from the Python Interpreter

- Invoke the [Python Interpreter](#).
- Type the following into the Python prompt.

```
>>> import gclib
>>> g = gclib.py()
>>> g.GOpen('192.168.0.42')
>>> print(g.GInfo())
192.168.0.42, DMC4080 Rev 1.2c, 783
```

## Running Python scripts

- Navigate the terminal to the location from **Install the gclib Python module** where [example.py](#) was copied.
- Open [example.py](#) in a text editor.
- Set the address in the `g.GOpen()` call to match an available connection.
- Execute the following command at the Terminal.

```
$ python example.py
gclib version: py.127.110.250
192.168.0.42, DMC4080 Rev 1.2c, 783
```

- Experiment with the example by uncommenting sections, between the triple quotes, `'''`.

```
$ python example.py
gclib version: py.127.110.250
192.168.0.42, DMC4080 Rev 1.2c, 783
GProgramDownload() correctly errored. Can't fit with level 3 compression
Uploaded program:
#A;i=0;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1
i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;EN
Downloaded program verified
Array element verified
 187942.0000

Starting move...
done.
```



## Getting help

```
>>> help(g.GOpen)
Help on method GOpen in module gclib:

GOpen(address) method of gclib.py instance
  Opens a connection a galil controller.
  See the gclib docs for address string formatting.

>>> help(g.GCommand)
Help on method GCommand in module gclib:

GCommand(command) method of gclib.py instance
  Performs a command-and-response transaction on the connection.
  Trims the response.

>>> 'for a full listing, try help(g)'
```

## 3.3 .Net

- [VB.NET](#)
- [C#.NET](#)

### 3.3.1 VB.NET

gclib ships with [gclib.vb](#), a Visual Basic class which exposes the functionality of the gclib. In addition, a VB forms example is included which demonstrates how to use [gclib.vb](#). The following instructions were performed on Visual Studio Professional 2013 and can be extended to other Visual Studio versions.

## Running the included Visual Basic Example

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

### Copy files

- Navigate to a convenient, empty, writable location, e.g. *C:\temp*.
- Copy the contents of *C:\Program Files (x86)\Galil\gclib\examples\vb\2013\_12.0\gclib\_example* to this location.

### Open in Microsoft Visual Studio 2013

- Open *gclib\_example.sln* in Visual Studio. This demo was tested on MSVS 2013.

### Add existing item, [gclib.vb](#)

- In the *Solution Explorer*, right-click on *gclib\_example* and choose *Add->Existing Item...*
- Choose *C:\Program Files (x86)\Galil\gclib\source\wrappers\vb\gclib.vb*

### Run Demo

- Type *F5* to run the program.
- Type a valid [GOpen\(\)](#) address in the text box and click Go.

## Create Project from scratch with MSVC 2013

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

### Configure Project

- Launch Visual Studio 2013
- Choose File->New->Project
- In the *New Project* dialog, choose Visual Basic -> Windows Forms Application
- Type *gclib\_example* for the Name
- Choose a Location, e.g. C:\Users\user\Desktop
- Check *Create directory for solution*
- Click OK, the project will configure itself
- In the *Solution Explorer*, right click on *Solution 'gclib\_example' (1 project)* and choose *Configuration Manager...*
  - In the *gclib\_example* project row, click in the *Platform* column and choose <New...>
    - \* Choose *x86* from *Type or select the new platform:*
    - \* Choose *Any CPU* from *Copy settings from:*
    - \* Check *Create new solutions platform*
    - \* Click OK.
  - If x64 support is also desired, repeat the <New...> procedure for *x64*
  - In the *Active solution platform* combobox at the top of the *Configuration Manager* dialog, choose <Edit...>
    - \* Select *Any CPU* and click the *Remove* button
    - \* Click *Close*
  - Close the *Configuration Manager* dialog
- In the *Solution Explorer*, right-click on *gclib\_example* and choose Add->Existing Item
  - Navigate to the installation location C:\Program Files (x86)\Galil\gclib\source\wrappers\vb
  - Choose *gclib.vb*
- In the *Solution Explorer* double-click on *gclib.vb*
  - Note that there is a preprocessor definition starting with `#if PLATFORM = "x86" Then` and `#ElseIf PLATFORM = "x64" Then`
  - Note that these sections of code enable/disable with the choice of the *Solution Platform* x86/x64, usually found in the Visual Studio toolbar
  - If a non-default gclib installation location is used, the paths in these sections of code must be updated to reflect the dll locations

### Add some simple code

- In the *Solution Explorer* right-click on *Form1.vb* and choose *View Code*
- Replace the text in *Form1.vb* with the following code

```

Public Class Form1
    Dim gclib As New Gclib()
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "gclib simple example"
        Dim tb As New TextBox
        With tb
            .Multiline = True
            .Dock = DockStyle.Fill
            .Parent = Me
        End With
        Try
            'calls to gclib should be in a try-catch
            .AppendText("GVersion: " & gclib.GVersion() & vbCrLf)
            gclib.GOpen("192.168.0.42") 'Set an appropriate IP address here
            .AppendText("GInfo: " & gclib.GInfo() & vbCrLf)
            .AppendText("GCommand: " & gclib.GCommand("MG TIME") & vbCrLf)
        Catch ex As Exception
            .AppendText("ERROR: " & ex.Message)
        End Try
        Finally
            gclib.GClose() ' Don't forget to close!
        End Try
    End Sub
End Class

```

- In the `gclib.GOpen()` call, indicate a correct IP address for the controller that is used for this project
- Hit *F5* to run the project

### 3.3.2 C#.NET

`gclib` ships with [gclib.cs](#), a C# class which exposes the functionality of the `gclib`. In addition, a C# forms example is included which demonstrates how to use [gclib.cs](#).

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

## Running the C# Example

### Copy files

- Navigate to a convenient, empty, writable location, e.g. *C:\temp*.
- Copy the contents of *C:\Program Files (x86)\Galil\gclib\examples\cs\2013\_12.0\gclib\_example* to this location.

### Open in Microsoft Visual Studio 2013

- Open *gclib\_example.sln* in Visual Studio. This demo was tested on MSVS 2013.

### Add existing item, [gclib.cs](#)

- In the *Solution Explorer*, right-click on *gclib\_example* and choose *Add->Existing Item...*
- Choose *C:\Program Files (x86)\Galil\gclib\source\wrappers\cs\gclib.cs*

### Run Demo

- Type *F5* to run the program.
- Type a valid `GOpen()` address in the text box and click Go.

## Create Project from scratch with MSVC 2013

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

### Configure Project

- Launch Visual Studio 2013
- Choose File->New->Project
- In the *New Project* dialog, choose Visual C# -> Windows Forms Application
- Type *gclib\_example* for the Name
- Choose a Location, e.g. C:\Users\user\Desktop
- Check *Create directory for solution*
- Click OK, the project will configure itself
- In the *Solution Explorer*, right click on *Solution 'gclib\_example' (1 project)* and choose *Configuration Manager...*
  - In the *gclib\_example* project row, click in the *Platform* column and choose <New...>
    - \* Choose *x86* from *Type or select the new platform:*
    - \* Choose *Any CPU* from *Copy settings from:*
    - \* Check *Create new solutions platform*
    - \* Click OK.
  - If x64 support is also desired, repeat the <New...> procedure for *x64*
  - In the *Active solution platform* combobox at the top of the *Configuration Manager* dialog, choose <Edit...>
    - \* Select *Any CPU* and click the *Remove* button
    - \* Click *Close*
  - Close the *Configuration Manager* dialog
- In the *Solution Explorer*, right-click on *gclib\_example* and choose *Properties*
  - Choose the *Build* item on the left
    - \* In the *Configuration:* combobox, choose *All Configurations*
    - \* Choose *x86* from the *Platform* combobox
    - \* In *Conditional compilation symbols* type *x86*
  - If x64 is to be used also, add an *x64* token as well to the *x64 Platform*
  - Save and close the *Properties* window
- In the *Solution Explorer*, right-click on *gclib\_example* and choose Add->Existing Item
  - Navigate to the installation location C:\Program Files (x86)\Galil\gclib\source\wrappers\cs
  - Choose *gclib.cs*
- In the *Solution Explorer* double-click on *gclib.cs*
  - Note that there is a preprocessor definition starting with `#if x86` and `#elif x64`
  - Note that these sections of code enable/disable with the choice of the *Solution Platform* x86/x64, usually found in the Visual Studio toolbar
  - If a non-default gclib installation location is used, the paths in these sections of code must be updated to reflect the dll locations

## Add some simple code

- In the *Solution Explorer* right-click on [Form1.cs](#) and choose *View Code*
- Replace the text in [Form1.vb](#) with the following code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace gclib_example
{
    public partial class Form1 : Form
    {
        gclib gclib = new gclib();
        public Form1()
        {
            InitializeComponent();
            this.Text = "gclib simple example";
            TextBox tb = new TextBox();
            tb.Multiline = true;
            tb.Dock = DockStyle.Fill;
            tb.Parent = this;
            try
            {
                //calls to gclib should be in a try-catch
                tb.AppendText("GVersion: " + gclib.GVersion() + "\n");
                gclib.GOpen("192.168.0.42"); //Set an appropriate IP address here
                tb.AppendText("GInfo: " + gclib.GInfo() + "\n");
                tb.AppendText("GCommand: " + gclib.GCommand("MG TIME") + "\n");
            }
            catch(Exception ex)
            {
                tb.AppendText("ERROR: " + ex.Message);
            }
            finally
            {
                gclib.GClose(); //Don't forget to close!
            }
        }
    }
}
```

- In the [gclib.GOpen\(\)](#) call, indicate a correct IP address for the controller that is used for this project
- Hit *F5* to run the project

## 3.4 Java

gclib uses the venerable [Java Native Access \(JNA\)](#) library to simplify integration into the Java Native Interface (JNI).

### Attention

This is the initial version of the the gclib Java wrapper. As such, GclibJava ships as source files, not the compiled jar files. All functions are subject to change in future releases of gclib. Java hackers with recommendations on how to make this library better are encouraged to email [support@galil.com](mailto:support@galil.com). Somebody has to teach those Galil Java noobs what's what.

## Windows

The following instructions were performed with 64 bit Windows 7 on [Oracle NetBeans IDE 8.2](#) and [Java 1.8.0\\_131](#).

For brevity, these instructions assume the default gclib installation location of "C:\Program Files (x86)\Galil\gclib".

### Step-by-Step

1. Install [gclib](#) with 64 bit binaries (default install).
2. Install 64 bit NetBeans and Java, [jdk-8u131-nb-8\\_2-windows-x64.exe](#).
3. Launch NetBeans.
4. Create a new application.
  - (a) File | New Project...
  - (b) Under *Categories*, select *Java*.
  - (c) Under *Projects*, select *Java Application*.
  - (d) Click *Next*.
  - (e) Type `GclibTest` for the *Project Name*.
  - (f) Note the location of the *Project Folder*.
  - (g) Uncheck *Create Main Class*
  - (h) Click *Finish*
5. Open the *Project Folder* as noted above.
6. Open the *src* directory in the *Project Folder* location.
7. Copy the whole directory `C:\Program Files (x86)\Galil\gclib\examples\java\gclibtest` to this directory.
8. Copy the whole directory `C:\Program Files (x86)\Galil\gclib\source\wrappers\java\gclibjava` to this directory.
9. Create a directory at `c:\jna\`.
  - Another directory may be chosen. The purpose of this directory is to hold jna's *jar* binary for the Java classpath.
10. Download a copy of *jna.jar* to the new directory.
  - <https://github.com/java-native-access/jna#download>
  - This example uses *jna-4.4.0.jar*.
11. In the NetBeans *Projects* tab, expand *GclibTest*.
12. Right-click on *Libraries* and choose *Add JAR/Folder...*
13. Navigate to the *jna.jar* saved above. Click *Open* to add *jna.jar* to the classpath.
14. In the NetBeans *Projects* tab, right-click on *GclibTest* and choose *Properties*.
15. Choose the *Run* item out of the *Categories* options tree.
16. In the *Main Class* text box, type `gclibtest.GclibTest`. Click *OK*.
17. In the NetBeans *Projects* tab, expand *GclibTest* | *Source Packages* | *gclibtest*.
18. Double click [GclibTest.java](#), and find the line containing `gclib.GOpen`.
19. Update the address for the desired hardware.
20. Choose *Run* | *Run Project (GclibTest)* or hit the `F6` key to run the application.
21. The application output will print in the NetBeans *Output* window.

## Documentation

The GclibJava class has helpful documentation for developing a Java application. Use the following instructions to create the Javadoc.

1. In the NetBeans *Projects* tab, right-click *GclibTest*.
2. Choose *Generate Javadoc* to create the documentation and open it in the system's default browser.

## 3.5 LabVIEW

LabVIEW instructions are currently not available in PDF format. Please visit the web version of this document for LabVIEW instructions.





## Chapter 4

# Using gclib

- [gcaps](#)
- [Program Preprocessor](#)
- [Thread Safety](#)
- [Galil Widgets](#)
- [Rebuilding gclibo](#)
- [Legacy Compatibility](#)

### 4.1 gcaps

gcaps is a communication server natively supported by gclib to multiplex Galil hardware communication features. It runs in the background on the host computer, as a service or daemon.

Incidentally, the name *gcaps* is an acronym for the improbable name *Galil Controller Asynchronous Proxy Server*. Yet another tidbit to impress friends at parties.

### gclib & gcaps

gclib will attempt to use gcaps whenever [GOpen\(\)](#) is called without the `--direct` or `-d` switch. Other than this small difference, gclib function calls through gcaps operate as if the connection was direct. The first version of gclib supporting gcaps is 299.

### Other gcaps Usage

The following functions will attempt to use gcaps first to gather data. If gcaps is not found, the functions will fall back to user space calls to populate information.

gclib Function	Usage	If gcaps unavailable
<a href="#">GVersion()</a>	Provide the version of gclib and gcaps (if available).	No gcaps version.
<a href="#">GIpRequests()</a>	Provide a list of all Galil controllers requesting IP addresses via BOOT-P or DHCP.	Must be root.
<a href="#">GAssign()</a>	Assigns an IP address over the Ethernet to a controller at a given MAC address.	Must be root.
<a href="#">GAddresses()</a>	Provides a listing of all available connection addresses.	Must be root, or user must be in device group.

Because gcaps runs as a service on Windows, and as a system daemon on Linux, gcaps runs with root privileges. See *If gcaps unavailable* column in the above table when running without gcaps.

If gcaps is unavailable when these functions are run, a ~1 second delay will be incurred while gclib searches for the absent server. In order to prevent gcaps usage in these functions, comment out the symbol `G_USE_GCAPS` in `gclibo.h` and rebuild gclibo. See [Rebuilding gclibo](#).

## 4.2 Program Preprocessor

gclib's program downloader provides a preprocessor for DMC code. The preprocessor modifies the program prior to download providing a number of language features not present in native DMC code.

The preprocessor is invoked in the following two ways.

1. With both `GProgramDownload()` and `GProgramDownloadFile()` via the `preprocessor` argument. Downloading code with null for the preprocessor argument uses defaults.
2. From within DMC code via in-band preprocessor directives.

---

### The preprocessor argument

`GProgramDownload()` and `GProgramDownloadFile()` can be called with a string passed to the `preprocessor` argument. The program will be modified based on this string prior to download. See *Preprocessor Options* below for syntax.

---

### In-band Operation

DMC code can be written with special markup to signal the preprocessor to take actions prior to download. For example, the following program will invoke the in-band preprocessor. The specifics are described below.

```
## Author: Zaphod Beeblebrox
## Project: Total Perspective Vortex
//the above 4 hashmarks enable the preprocessor
##option "--min 4" //use a minimum of level four compression
REM REM-style comments are supported at all times
PRA=1000
BGA
AMA
EN
```

### The REM Comment

Lines beginning with the string `REM` are removed prior to download. `REM` comments are always removed regardless of whether the other preprocessor options are enabled or not.

### Double Hash

Most preprocessor statements begin with a double hash, `##`. When preceded by a space, the double hash acts like a `REM` comment.

When preceded by a character other than space, `##` is interpreted as a preprocessor directive. For example, see `##option` below.

#### Note

Double hash lines are removed from the program only when the preprocessor is enabled with a quad hash.

### Quad Hash to enable

In order to enable the in-band preprocessor, the first two lines of the DMC program must start with a double hash. This syntax of using two lines with double hashmarks is called a *quad hash*.

Content may follow the hash marks. For example, a good code writing style is to use double hash comments as a comment header showing author, project name, etc.

## C-style comments

With the preprocessor enabled, C-style comments may be used with the `//` prefix. These comments are very similar to `REM` comments. The primary advantage of using this comment over `REM` is that `//` comments may occur anywhere in a line. This is helpful for line comments such as the following.

```
SIA= 1,25,25,0<4>1 //SSI 25 bits total, all single turn, no status
```

Strings containing `//` are not interpreted as comments.

### Note

`//` comments are removed from the program only when the preprocessor is enabled with a quad hash.

## Preprocessor Directives

### Note

Directives are only followed when the preprocessor is enabled with a quad hash.

### ##option

The `option` directive allows passing switches directly to the preprocessor with the same syntax as the `preprocessor` argument in `GProgramDownload()` and `GProgramDownloadFile()`. The syntax of the `option` directive is the following.

```
##option "{preprocessor switches}"
```

For example, the following line will disable compression in the program.

```
##option "--max 0"
```

See *Preprocessor Options* below for other switches.

### ##include

The `include` directive provides a way to include the contents of another DMC file in the current program. This is useful for reusing code such as automatic subroutines, homing operations, or controller initialization routines.

The contents of the file will be inserted in place of the `include` line. The insertion occurs prior to code compression.

The syntax of the `include` directive is the following.

```
##include "{filename}"
```

For example,

```
##include "c:\galil\initialize.dmc"
##include "homing.dmc"
```

To write more portable code, use the `include` directive with just the file name, no absolute path. The path to find the file on the system is set depending on usage.

1. In the *Galil Design Kit*, specify the include path in GDK's *settings* with the `--search` or `-I` switch as defined below.
2. When downloading code via `GProgramDownload()` or `GProgramDownloadFile()`, use the `--search` or `-I` switch in the `preprocessor` argument.
3. Finally, if the file is in the executable search path, the file will be found. However, one of the previous two options is more reliable.

### ##gclib

*Galil Design Kit* uses the `##gclib` directive in *GDK Macros*. `gclib` ignores this directive.

## In-band Support

In addition to `gclib`, *Galil Design Kit* supports the preprocessor. Proper preprocessor usage will be colored in the Editor's syntax highlighter. If the quad hash is not present, preprocessor syntax will be colored differently to indicate improper usage.

The preprocessor is not supported in software prior to GDK/`gclib`. DMC code downloads using the in-band preprocessor in prior generation software (e.g. *GalilTools* or *SmartTerm*) will fail with a TC code of 61, *Duplicate or bad label*.

## Preprocessor Options

### Compression, `--min`, `--max`

- Default uses minimum compression needed to fit the program.
- `--max n` provides compression up to and including level *n*. Only the necessary compression will be performed up to level *n*.
- `--min n` will compress at least up to and including *n*. *n* defined as with `--max`.

### Compression Levels, *n*

- Level 0 (mandatory)
  1. Remove lines beginning with `REM`.
  2. Remove trailing semicolons.
  3. Comment blank lines with `'`.
  4. Remove white space (space/tab) in front of `#` (label declarations).
  5. Remove white space after commands.
  6. Line ends changed to carriage return.
  7. Replace leading tabs with double space.
  8. Replace non-leading tabs with single space.
  9. A backslash (`\`) character on a line other than a preprocessor line will result in an error.
- Level 1
  1. Remove unnecessary spaces. Strings, comments (`'`), and no-ops (`NO`) are not changed.
- Level 2
  1. Remove comments (`'`) but not no-ops (`NO`).
- Level 3
  1. Remove no-ops (`NO`).
- Level 4
  1. Break apart compound lines that are too long.
  2. Compact lines of code to maximize line usage.
  3. Use backtick to support long lines where applicable.

### Code insertion, `--insert`

- Default begins at line zero and overwrites anything present.
- `--insert arg` invokes the insert option of the firmware's `DL` command. *arg* can be one of the following.
  1. Line number, e.g. `100`. Program insertion will occur on the line after the line specified.
  2. Variable name, e.g. `myvar`. Program insertion will occur on the line after the line equal to the value of the variable.
  3. Label callout, e.g. `#mylabel`. Program insertion will occur on the line after the label.
  4. A lone `#` symbol. Program insertion will occur on the line after the last line in the program buffer.
- Compression directives `--max` and `--min` are followed.
- All original code following the point of insertion is cleared.
- Not all products support the `--insert` operation, e.g. DMC-30010. See the `DL` command for support.

#### Warning

It is the user's responsibility to ensure that the code will fit in the inserted location. The preprocessor will not check line numbers when executing the `--insert` option.

### Include Search Paths, `--search`, `-I`

- The `##include` directive will attempt to open its string argument directly. The open will succeed if the argument is the absolute path, or if the argument is in the executable's path, e.g. in the same directory.
- `--search path` allows the user to specify a directory or directories to be searched for the `include` file in case the first open fails.
  - For historical reasons, `-I` is shorthand for `--search`.
- Multiple directories may be specified with multiple `-I` directives.
- For in-band code, `-I` must be specified prior to the include.
- A common use for `-I` is to specify only the filename in the DMC source code and use the `preprocessor` argument during download to specify the path to the files. This allows the files to be moved without a change to source code.
- Search order
  1. The `##include` argument is checked first as-is.
  2. Then each `-I` argument in the `preprocessor` argument, in the order specified.
  3. Then `##option` directives in the DMC file, in the order specified.
- If the search path contains spaces, enclose the path in double quotes, escaped with a backslash. See example below.

#### In-band Example

```
##option "-I /code/dmc/homing"
##option "-I \"/code/dmc/other code\"
##include "auto.dmc"
//executable's directory will be checked
//then c:\code\dmc\homing
//then c:\code\dmc\other code
```

### Macro Definition, `--define`, `-D`

- `--define` provides a way to substitute one token for another. This is useful for writing code that is generic until program download. Wherever the token is found in code, it is substituted by the replacement. The replacement occurs right before code compression.
- `-D` is shorthand for `--define`.
- The token should consist of a starting backslash character, followed by upper or lower case alphanumeric characters, underscores, and an ending backslash.
- The common usage for this feature is to write code with a token, and then call the program download with the `-D` switch.

In this example, an axis is defined at download time. Specifying the following for the preprocessor argument

```
--define \ax\:A
```

would cause the following code

```
SH\ax\
JG\ax|=1000
BG\ax\
```

to be downloaded as

```
SHA
JGA=1000
BGA
```

This causes the *A* axis to be addressed.

#### Note

The macro `\pid\` is reserved for exclusive use by GDK.

## Conditional Directives, `--ifdef`, `--ifndef`

To specify a preprocessor directive should be executed only if a macro is defined, use `--ifdef`.  
`##option "--ifdef \minify\ --min 4" //maximally compress code if minify macro set`

To specify a preprocessor directive should be executed only if a macro is NOT defined, use `--ifndef`.  
`##option "--ifndef \axis\ -D \axis:A" //Default to axis A`

## GDK Support

- See the `preprocessor` text box in the *Editor* settings page to set the desired preprocessor setting for developing in GDK's editor.

## 4.3 Thread Safety

### The Basics

- The easiest way to multithread, and/or to use multiple applications to access the same hardware, is to communicate through [gcaps](#).
- Just leave out `-d` and `--direct` in your `GOpen()` address and `gcaps` will be used.
- Each thread, and each application, should use their own `GCon` handle. In the higher-level [Language Support](#), each thread or application should manage their own `gclib` object. Don't pass the connection handle between threads.

### The Formalism

`gclib` supports multi-threaded operation with the following operational definitions.

#### `gclib` is "reentrant"

Reentrant means that a given `gclib` function call may be invoked in multiple threads when passed distinct arguments. For example, `GCommand()` may be called simultaneously in different threads so long as the following arguments have unique values, indicating they point to unique memory.

- `GCon g`, the connection must be unique.
- `GBufOut buffer`, the writable buffer must be unique.
- `GSize *bytes_returned`, the writable value must be unique.

#### `gclib` is not "thread-safe"

Thread safety would imply that a given `gclib` function call could be invoked in multiple threads when passed *the same* arguments. This mode of operation **is not** supported by `gclib`. In other words, it is not safe to call `GCommand()` simultaneously in different threads if any mutable arguments point to the same memory.

In short, it is **not** safe to call `GCommand()` in multiple threads to the same physical connection.

If such operation is required, it is the user's responsibility to use a mutual exclusion (mutex) or other mechanism to protect memory.

### Multi-threaded access to the same connection with `gcaps`

`gcaps` provides a multiplexing capability to Galil hardware. When using `gcaps`, it is therefore safe to call `GCommand()` in multiple threads to the *same physical connection* (though not the same `GCon` value). `gclib` can connect multiple times to the same Galil connection through `gcaps`. Because the `GCon` variable is unique, the reentrant capability of `gclib` can be used to communicate to the same physical connection through `gcaps`.

## 4.4 Galil Widgets

### Note

gclib provides the communications foundation for the Galil Widgets project. Galil Widgets are a collection of .Net WinForms User Controls that provide quick development of custom graphical user interfaces (GUIs) that communicate with [Galil](#) Motion Controllers and PLCs.

## Galil Widgets has been designed to support three general user needs

### The software novice, or the hurried prototyper

Within minutes, a full UI can be laid out. All controls can be configured with menus and mouse clicks for an absolute minimum requirement for writing code. The quick start guide, and Microsoft Visual Studio Express is all that is needed to make a free application GUI with minimal effort.

### The .Net developer, adding to pre-existing code.

In addition to the point-and-click configuration of the tools, each tool has a set of public function calls and properties which allows the C# or VB.Net user the ability to integrate the Galil Widgets into a .Net application with ease.

### The power user

The entire Galil Widgets source code is available in the installation package. This allows users to tweak, extend, and add Widgets to the library with ease. The "GalilWidget" interface defines a number of function calls that new Widgets should implement to function correctly.

## The following widgets are currently available

- **GWComs**: Communications to Galil hardware including event-driven handling of asynchronous traffic.
- **GWTerm**: A terminal for direct user interaction with the hardware.
- **GWPoll**: A polling tool to display important data on screen.
- **GWSettings**: A tool for displaying, editing, backing up, and restoring controller parameters and mission-critical variables. Program backup and loading, and firmware upgrades are also supported.
- **GWDatRec**: A data record visualization tool. Used to display controller status through user-configurable labels, "soft LEDs", and analog sliders.

---

## For more information, get the free [Galil Widgets package](#)

See the [Galil Widgets release notes](#) for changes.

**Screen shots of an example motion controller configuration (left), and a similar RIO configuration (right)**

---

## 4.5 Rebuilding gclibo

gclib ships with a compiled version of the open source portion, *gclibo*. However, if a source modification is desired, the following instructions will help with recompiling this portion of the library.

---

## Windows

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib** and a build type of **x86 (win32)**. The following instructions were performed on *Visual Studio Professional 2015* and can be extended to other Visual Studio versions.

---

## Preparation

Create a working directory. A convenient, empty, writable location, e.g.

```
C:\>mkdir %homepath%\Desktop\temp
```

### Note

In this documentation, a single *greater-than* character (>) will indicate a command prompt at this working directory.

Recompiling gclibo requires the source code for the open source compression library **zlib**. This can be downloaded from the zlib website: <http://zlib.net/zlib1211.zip>.

Extract the downloaded zlib source files to the working directory.

Open *VS2015 x86 Native Tools Command Prompt* and navigate to the working directory.

```
C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC>cd %homepath%\Desktop\temp
C:\Users\user\Desktop\temp>dir /b
zlib-1.2.11
```

## Copy files

### Set an environment variable for the base path.

```
>set base="C:\Program Files (x86)\Galil\gclib"
```

### Set an environment variable for the zlib base path.

```
>set zlib="%CD%\zlib-1.2.11"
```

### Copy the gclibo source files.

```
>copy %base%\source\gclibo\*.c .
C:\Program Files (x86)\Galil\gclib\source\gclibo\arrays.c
C:\Program Files (x86)\Galil\gclib\source\gclibo\gclibo.c
      2 file(s) copied.
```

## Modify source

Make any necessary changes. For this example, the `GInfo()` function was changed from

```
GReturn GCALL GInfo(GCon g, GCStringOut info, GSize info_len)
{
    return GUtility(g, G_UTIL_INFO, info, &info_len);
}
```

to

```
GReturn GCALL GInfo(GCon g, GCStringOut info, GSize info_len)
{
    strncpy(info, "My controller", info_len);
    return G_NO_ERROR;
    //return GUtility(g, G_UTIL_INFO, info, &info_len);
}
```

## Compile and copy

### Compile the source code.

```
>cl -c *.c %zlib%\*.c -I %base%\include -I %zlib% -DBUILDING_GCLIB
```

### Link the binaries.

```
>link /DLL *.obj %base%\lib\dynamic\x86\gclib.lib /OUT:gclibo.dll
```

### Copy

Copy back to the installation location from the file explorer. This will require administrator privileges.

- Copy `gclibo.lib` to "C:\Program Files (x86)\Galil\gclib\lib\dynamic\x86"
- Copy `gclibo.dll` to "C:\Program Files (x86)\Galil\gclib\dl\x86"



## Test

### Copy simple example

```
>copy %base%\examples\cpp\x_simple.c .
```

### Edit [GOpen\(\)](#) call as necessary.

### Compile

```
>cl x_simple.c %base%\lib\dynamic\x86\*.lib -I %base%\include
```

### Set Path to DLL

```
>set PATH=%base%\dll\x86\;%PATH%
```

### Execute

```
>x_simple.exe
rc: 0
version: 85.60.138
rc: 0
rc: 0
info: My controller
rc: 0
response: 355000958.0000
:
```

---

## Linux

### Copy files

```
$ cp -r /usr/share/gclib/src/gclibo .
$ cd gclibo
$ cp /usr/include/gclib*.h .
```

### Modify source

Make any necessary changes. For this example, the [GInfo\(\)](#) function was changed from

```
GReturn GCALL GInfo(GCon g, GCStringOut info, GSize info_len)
{
    return GUtility(g, G_UTIL_INFO, info, &info_len);
}
to
GReturn GCALL GInfo(GCon g, GCStringOut info, GSize info_len)
{
    strncpy(info, "My controller", info_len);
    return G_NO_ERROR;
    //return GUtility(g, G_UTIL_INFO, info, &info_len);
}
```

### Make and install

```
# make install -f makefile_gclibo
# make clean -f makefile_gclibo
```

## Test

### Copy simple example

```
$ cp /usr/share/gclib/doc/examples/cpp/x_examples/x_simple.c .
```

### Edit [GOpen\(\)](#) call as necessary.

### Compile

```
$ gcc x_simple.c -Wall -Werror -lgclib -lgclibo -o simple
```

---

## Execute

```
$ ./simple
rc: 0
version: 85.60.131
rc: 0
rc: 0
info: My controller
rc: 0
response: 182879322.0000
:
```

---

## OS X

### Copy files

```
$ mkdir test
$ cd test
$ tar -xvf /Applications/gclib/source/gclibo_src.tar.gz x gclibo.h
x gclibo.c
x arrays.c
x makefile_gclibo
$ cp /Applications/gclib/include/* .
$ cp /Applications/gclib/dylib/gclib.0.dylib .
$ ls
arrays.c gclib.h gclib_record.h gclibo.h
gclib.0.dylib gclib_errors.h gclibo.c makefile_gclibo
```

### Modify source

Make any necessary changes. For this example, the `GInfo()` function was changed from

```
GReturn GCALL GInfo(GCon g, GCStringOut info, GSize info_len)
{
    return GUtility(g, G_UTIL_INFO, info, &info_len);
}
to
GReturn GCALL GInfo(GCon g, GCStringOut info, GSize info_len)
{
    strncpy(info, "My controller", info_len);
    return G_NO_ERROR;
    //return GUtility(g, G_UTIL_INFO, info, &info_len);
}
```

### Make and install

```
$ make -f makefile_gclibo
Open source component, gclibo.0.dylib
Compiling open source component.
gcc -c -Wall -Werror -fPIC -fvisibility=hidden -DBUILDING_GCLIB -DHAVE_VISIBILITY *.c
Linking open source component into shared library.
gcc -dynamiclib -o gclibo.0.dylib *.o gclib.0.dylib
strip -u -r gclibo.0.dylib
Cleaning up.
$ make install -f makefile_gclibo
Installing gclibo.0.dylib
cp gclibo.0.dylib /Applications/gclib/dylib
$ make clean -f makefile_gclibo
Cleaning project...
```

## Test

### Extract simple example

```
$ tar -xzf /Applications/gclib/examples/gclib_examples.tar.gz x_simple.c
```

Edit `GOpen()` call as necessary.

### Compile

```
$ gcc x_simple.c -Wall -Werror gclib.0.dylib gclibo.0.dylib -o simple
```

---

**Execute**

```
$ ./simple
rc: 0
version: 127.110.253
rc: 0
rc: 0
info: My controller
rc: 0
response: 182879322.0000
:
```

## 4.6 Legacy Compatibility

- [GalilTools](#) included the GCL (GalilTools Communication Library). gclib ships with an open source wrapper implementation of the GCL.
- [DMC32 OSU](#) is intended for existing applications that used software based on the legacy DMCWIN32 library for Windows XP and earlier.

### 4.6.1 GalilTools

To provide maximum compatibility, gclib ships with an open source wrapper implementation of the GCL (GalilTools Communication Library). Users wanting to upgrade to gclib that have source built on [Galil.h](#) can use this wrapper to minimize source changes. This wrapper is also indicated for users that want the same function calls as [Galil.h](#), but don't want the usage of `QT` as in `galil1.dll`.

**This wrapper is intended for existing applications already using the library distributed with GalilTools (galil1.dll) or the previous STL library (galil2.dll). New applications should be written with gclib.**

## Windows

### Compile galil2.dll with MSVC 2013

The following instructions were performed on *Visual Studio Professional 2013* and can be extended to other Visual Studio versions. For brevity, the instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib** and a build type of **x86 (win32)**.

#### Launch the compiler command prompt

- Open *VS2013 x86 Native Tools Command Prompt*.
- Navigate to a convenient, writable location, e.g. *C:\temp*.

#### Set an environment variable for the base path

```
C:\temp>set base=C:\Program Files (x86)\Galil\gclib
```

#### Compile the source code

**Note the quotes.**

```
C:\temp>cl -c "%base%\source\wrappers\gcl\*.cpp" -I "%base%\include" -EHsc -MD
```

#### Link the source code

**Note the quotes.**

```
C:\temp>link /DLL gcl_datarecord.obj gcl_galil.obj "%base%\lib\dynamic\x86\gclib.lib" "%base%\lib\dynamic\x86\
```

The output files *galil2.dll* and *galil2.lib* can now be used in a project using the GCL.

## Test

Help the loader find the right dlls.

```
C:\temp>set PATH=%PATH%;%BASE%\dll\x86
```

Link the simple example.

```
C:\temp>link gcl_simple.obj "%base%\lib\dynamic\x86\gclib.lib" "%base%\lib\dynamic\x86\gclibo.lib" galil2.lib
```

Run the example.

```
C:\temp>simple.exe
Galil2.dll wrapper, gclib 106.75.180
10.1.3.169, DMC4020 Rev 1.2c, 291
```

## Linux

### Copy files

```
$ cp -r /usr/share/gclib/src/wrappers/gcl .
$ cd gcl
$ ls
Galil.h          gcl_galil.cpp  gcl_simple.cpp
gcl_datarecord.cpp gcl_galil.h   makefile
```

### Make and install

```
$ make
gcl open source wrapper for gclib
  Compiling wrapper, libgalil.so.2.0
g++ -c -fPIC -std=c++11 gcl_datarecord.cpp gcl_galil.cpp
  Linking wrapper into shared library.
g++ -shared -o libgalil.so.2.0 *.o -Wl,-soname=libgalil.so.2
strip --strip-unneeded libgalil.so.2.0
  Cleaning up.
$ sudo make install
Installing libgalil.so.2.0
install -m 755 libgalil.so.2.0 /usr/lib
install -m 644 Galil.h /usr/lib
ldconfig
ln -s /usr/lib/libgalil.so.2 /usr/lib/libgalil.so
$ make clean
Cleaning project...
```

## Test

```
$ g++ gcl_simple.cpp -lgalil -lgclib -lgclibo -o simple
$ ./simple
Galil2.dll wrapper, gclib 95.71.164
10.1.3.169, DMC4020 Rev 1.2c, 291
```

### 4.6.2 DMC32 OSU

#### Note

gclib provides the communications foundation for the *DMC32 Operating System Upgrade (OSU)* project.

DMC32 OSU is intended for existing applications that used software based on the legacy DMCWIN32 library for Windows XP and earlier. If such an application must be upgraded to Windows 7 ‡, 8, 8.1, or 10 DMC32 OSU may be used on these O.S. upgrades.

‡ Galil's support for Windows 7 has ended. Please click [here](#) for more information.

### Galil's Windows XP support statement,

<http://www.galil.com/about/xp-support>

- For more information refer to the documentation, <http://www.galil.com/sw/pub/all/doc/dmc32osu/html/index.html>

- See the release notes for changes, <http://www.galil.com/sw/pub/all/rn/dmc32osu.html>
- The installer is available for download from Galil's website, [http://www.galil.com/sw/pub/win/dmc32osu/galil\\_dmc32\\_osu\\_exe.html](http://www.galil.com/sw/pub/win/dmc32osu/galil_dmc32_osu_exe.html)



# Chapter 5

## Module Index

### 5.1 Modules

Here is a list of all modules:

API . . . . .	77
C++ examples . . . . .	106
C#/VB examples . . . . .	114
C#/VB API . . . . .	121





# Chapter 6

## Namespace Index

### 6.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">examples</a>	123
<a href="#">gclib_example</a>	126
<a href="#">gclibjava</a>	126
<a href="#">vb</a>	126



# Chapter 7

## Hierarchical Index

### 7.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Commands_Example	127
Contour_Example	128
Examples	128
Exception	
gclib.GclibError	162
GclibJavaException	173
Form	
MainForm	292
Galil	135
GalilPrivate	136
gclib	136
Gclib	150
GclibJava	163
GclibTest	174
gclib.GDataRecord	174
gclib.GDataRecord1802	177
gclib.GDataRecord1806	188
gclib.GDataRecord2103	208
gclib.GDataRecord30000	222
gclib.GDataRecord4000	228
gclib.GDataRecord47000_ENC	248
gclib.GDataRecord47162	253
gclib.GDataRecord47300_24EX	259
gclib.GDataRecord47300_ENC	265
gclib.GDataRecord52000	270
GDataRecord	175
Gclib.GDataRecord	176
Gclib.GDataRecord1802	184
Gclib.GDataRecord1806	201
Gclib.GDataRecord2103	217
Gclib.GDataRecord30000	226
Gclib.GDataRecord4000	241
Gclib.GDataRecord47000_ENC	251
Gclib.GDataRecord47162	257
Gclib.GDataRecord47300_24EX	263
Gclib.GDataRecord47300_ENC	268
Gclib.GDataRecord52000	284
GDataRecord1802	181
GDataRecord1806	194
GDataRecord2103	212

---

GDataRecord30000 . . . . .	224
GDataRecord4000 . . . . .	234
GDataRecord47000_ENC . . . . .	250
GDataRecord47162 . . . . .	255
GDataRecord47300_24EX . . . . .	261
GDataRecord47300_ENC . . . . .	266
GDataRecord52000 . . . . .	277
H_ArrayData . . . . .	290
IP_Assigner_Example . . . . .	291
Jog_Example . . . . .	292
Message_Example . . . . .	293
Motion_Complete_Example . . . . .	293
Position_Tracking_Example . . . . .	294
Program . . . . .	295
gclib.py . . . . .	295
Record_Position_Example . . . . .	301
Remote_Client_Example . . . . .	302
Remote_Server_Example . . . . .	303
Source . . . . .	304
Vector_Mode_Example . . . . .	304

# Chapter 8

## Data Structure Index

### 8.1 Data Structures

Here are the data structures with brief descriptions:

- [Commands\\_Example](#)
  - Demonstrates various uses of `GCommand()` and basic controller queries . . . . . 127
- [Contour\\_Example](#)
  - Record user's training and plays back training through contour mode . . . . . 128
- [Examples](#)
  - Provides a class of shared constants and methods for gclib's example projects . . . . . 128
- [Galil](#) . . . . . 135
- [GalilPrivate](#) . . . . . 136
- [gclib](#)
  - Provides a class that binds to gclib's unmanaged dll. Wraps each call and provides a more user-friendly interface for use in C# . . . . . 136
- [Gclib](#)
  - Provides a class that binds to gclib's unmanaged dll. Wraps each call and provides a more user-friendly interface for use in Visual Basic . . . . . 150
- [gclib.GclibError](#) . . . . . 162
- [GclibJava](#) . . . . . 163
- [GclibJavaException](#) . . . . . 173
- [GclibTest](#) . . . . . 174
- [gclib.GDataRecord](#) . . . . . 174
- [GDataRecord](#)
  - Data record union, containing all structs and a generic byte array accessor . . . . . 175
- [Gclib.GDataRecord](#) . . . . . 176
- [gclib.GDataRecord1802](#)
  - Data record struct for DMC-1802 controllers . . . . . 177
- [GDataRecord1802](#) . . . . . 181
- [Gclib.GDataRecord1802](#) . . . . . 184
- [gclib.GDataRecord1806](#)
  - Data record struct for DMC-1806 controller . . . . . 188
- [GDataRecord1806](#)
  - Data record struct for DMC-1806 controller . . . . . 194
- [Gclib.GDataRecord1806](#) . . . . . 201
- [gclib.GDataRecord2103](#)
  - Data record struct for DMC-2103 controllers . . . . . 208
- [GDataRecord2103](#)
  - Data record struct for DMC-2103 controllers . . . . . 212
- [Gclib.GDataRecord2103](#) . . . . . 217
- [gclib.GDataRecord30000](#)
  - Data record struct for DMC-30010 controllers . . . . . 222
- [GDataRecord30000](#)
  - Data record struct for DMC-30010 controllers . . . . . 224

Gclib.GDataRecord30000	226
gclib.GDataRecord4000	
Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0	228
GDataRecord4000	
Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0	234
Gclib.GDataRecord4000	241
gclib.GDataRecord47000_ENC	
Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields	248
GDataRecord47000_ENC	
Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields	250
Gclib.GDataRecord47000_ENC	251
gclib.GDataRecord47162	
Data record struct for RIO-47162	253
GDataRecord47162	
Data record struct for RIO-47162	255
Gclib.GDataRecord47162	257
gclib.GDataRecord47300_24EX	
Data record struct for RIO-47300 with 24EX I/O daughter board	259
GDataRecord47300_24EX	
Data record struct for RIO-47300 with 24EX I/O daughter board	261
Gclib.GDataRecord47300_24EX	263
gclib.GDataRecord47300_ENC	
Data record struct for RIO-47300. Includes encoder fields	265
GDataRecord47300_ENC	
Data record struct for RIO-47300. Includes encoder fields	266
Gclib.GDataRecord47300_ENC	268
gclib.GDataRecord52000	
Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40	270
GDataRecord52000	
Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40	277
Gclib.GDataRecord52000	284
H_ArrayData	
Structure to create a linked list for array data	290
IP_Assigner_Example	
Assigns controller an IP Address given a serial number and a 1 byte address	291
Jog_Example	
Accepts user-input at the command line to control the speed of the controller in Jog mode	292
MainForm	
Demonstrates using gclib in a Windows Form, including using a second thread to free the GUI	292
Message_Example	
Demonstrates how to handle and interpret messages from the controller	293
Motion_Complete_Example	
Uses controller interrupts to detect when motion is complete	293
Position_Tracking_Example	
Places controller into position tracking mode. Accepts user-defined positional values at the command line	294
Program	295
gclib.py	295
Record_Position_Example	
Takes two file paths at the command line to hold positional data for Axis A and Axis B. Positional data is saved to the two files until an analog input value changes	301
Remote_Client_Example	
Demonstrates various uses of GListServers() and GSetServer()	302
Remote_Server_Example	
Demonstrates various uses of GPublishServer()	303
Source	304

[Vector\\_Mode\\_Example](#)

Takes a path to a file at the command line holding vector commands for the controller. The controller is placed into vector mode and commands are read from the file and sent to the controller [304](#)





# Chapter 9

## File Index

### 9.1 File List

Here is a list of all documented files with brief descriptions:

arrays.c	307
commands.cpp	310
commands.cs	311
Commands.vb	311
commands_example.cpp	311
commands_example.cs	312
Commands_Example.vb	312
contour.cpp	312
contour.cs	313
Contour.vb	313
contour_example.cpp	314
contour_example.cs	315
Contour_Example.vb	315
example.py	315
examples.cs	315
examples.h	315
Examples.vb	321
Form1.cs	321
Form1.vb	321
Galil.h	321
gcl_datarecord.cpp	321
gcl_galil.cpp	322
gcl_galil.h	322
gcl_simple.cpp	322
gclib.cs	322
gclib.h	323
gclib.py	336
gclib.vb	337
gclib_errors.h	337
gclib_record.h	339
GclibJava.java	340
GclibJavaException.java	340
gclibo.c	341
gclibo.h	353
GclibTest.java	367
ip_assigner.cpp	367
ip_assigner.cs	368
IP_Assigner.vb	368
ip_assigner_example.cpp	368
ip_assigner_example.cs	369

IP_Assigner_Example.vb	369
jog.cpp	369
jog.cs	370
Jog.vb	370
jog_example.cpp	370
jog_example.cs	371
Jog_Example.vb	372
main.cpp	372
message.cpp	372
message.cs	373
Message.vb	373
message_example.cpp	373
message_example.cs	374
Message_Example.vb	374
motion_complete.cpp	374
motion_complete.cs	375
Motion_Complete.vb	375
motion_complete_example.cpp	375
motion_complete_example.cs	376
Motion_Complete_Example.vb	377
position_tracking.cpp	377
position_tracking.cs	377
Position_Tracking.vb	378
position_tracking_example.cpp	378
position_tracking_example.cs	379
Position_Tracking_Example.vb	379
Program.cs	379
record_position.cpp	379
record_position.cs	380
Record_Position.vb	380
record_position_example.cpp	380
record_position_example.cs	381
Record_Position_Example.vb	382
remote_client.cpp	382
Remote_Client.cs	382
Remote_Client.vb	383
remote_client_example.cpp	383
remote_client_example.cs	384
Remote_Client_Example.vb	384
remote_server.cpp	384
Remote_Server.cs	385
Remote_Server.vb	385
remote_server_example.cpp	385
remote_server_example.cs	386
Remote_Server_Example.vb	387
setup.py	387
vector.cpp	387
vector_example.cpp	388
vector_mode.cs	389
Vector_Mode.vb	389
vector_mode_example.cs	389
Vector_Mode_Example.vb	390
x_arrays.cpp	390
x_examples.cpp	390
x_examples.h	390
x_gcommand.cpp	394
x_ginterrupt.cpp	394
x_gmessage.cpp	395

---

<a href="#">x_gmotioncomplete.cpp</a>	396
<a href="#">x_gread_gwrite.cpp</a>	396
<a href="#">x_grecord.cpp</a>	397
<a href="#">x_nonblocking.cpp</a>	397
<a href="#">x_programs.cpp</a>	398
<a href="#">x_simple.c</a>	399



# Chapter 10

## Module Documentation

### 10.1 API

#### 10.1.1 Description

Language Support	C++	C#	VB.NET	Java	Python
gclib Functions	Yes	Yes	Yes	Yes	Yes
Data Records and Data Structures	Yes	Yes	Yes	No	No
gclib Macros	Yes	No	No	No	No
gclib Typedefs	Yes	No	No	No	No

#### Files

- file [gclib.h](#)
- file [gclibo.h](#)
- file [gclibo.c](#)
- file [gclib\\_record.h](#)
- file [gclib\\_errors.h](#)

#### Data Structures

- struct [GDataRecord4000](#)  
*Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0.*
- struct [GDataRecord52000](#)  
*Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40.*
- struct [GDataRecord1806](#)  
*Data record struct for DMC-1806 controller.*
- struct [GDataRecord2103](#)  
*Data record struct for DMC-2103 controllers.*
- struct [GDataRecord1802](#)
- struct [GDataRecord30000](#)  
*Data record struct for DMC-30010 controllers.*
- struct [GDataRecord47000\\_ENC](#)  
*Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields.*
- struct [GDataRecord47300\\_ENC](#)  
*Data record struct for RIO-47300. Includes encoder fields.*
- struct [GDataRecord47300\\_24EX](#)  
*Data record struct for RIO-47300 with 24EX I/O daughter board.*
- struct [GDataRecord47162](#)  
*Data record struct for RIO-47162.*

- union [GDataRecord](#)  
*Data record union, containing all structs and a generic byte array accessor.*

## Macros

- #define **GCLIB\_DLL\_EXPORTED**
- #define **GCALL** `__stdcall`  
*Specify calling convention for Windows.*
- #define **G\_DR** 1  
*Value for [GRecord\(\)](#) method variable for acquiring a data record via DR mode.*
- #define **G\_QR** 0  
*Value for [GRecord\(\)](#) method variable for acquiring a data record via QR mode.*
- #define **G\_BOUNDS** -1  
*For functions that take range options, e.g. [GArrayUpload\(\)](#), use this value for full range.*
- #define **G\_CR** 0  
*For [GArrayUpload\(\)](#), use this value in the delim field to delimit with carriage returns.*
- #define **G\_COMMA** 1  
*For [GArrayUpload\(\)](#), use this value in the delim field to delimit with commas.*
- #define **G\_PUBLISH\_SERVER** 1  
*For [GPublishServer\(\)](#), use this value to publish server to local network.*
- #define **G\_REMOVE\_SERVER** 0  
*For [GPublishServer\(\)](#), use this value to remove server from local network.*
- #define **G\_UTIL\_TIMEOUT** 1  
*[GUtility\(\)](#), Access to timeout.*
- #define **G\_UTIL\_TIMEOUT\_OVERRIDE** 2  
*[GUtility\(\)](#), read/write access to timeout override.*
- #define **G\_USE\_INITIAL\_TIMEOUT** -1  
*[GUtility\(\)](#), for timeout override. Set `G_UTIL_TIMEOUT_OVERRIDE` to this value to use initial [GOpen\(\)](#) timeout (`--timeout`).*
- #define **G\_UTIL\_VERSION** 128  
*[GUtility\(\)](#), get a library version string.*
- #define **G\_UTIL\_INFO** 129  
*[GUtility\(\)](#), get a connection info string.*
- #define **G\_UTIL\_SLEEP** 130  
*[GUtility\(\)](#), specify an interval to sleep.*
- #define **G\_UTIL\_ADDRESSES** 131  
*[GUtility\(\)](#), get a list of available connections.*
- #define **G\_UTIL\_IPREQUEST** 132  
*[GUtility\(\)](#), get a list of hardware requesting IPs.*
- #define **G\_UTIL\_ASSIGN** 133  
*[GUtility\(\)](#), assign IP addresses over the network.*
- #define **G\_UTIL\_DEVICE\_INITIALIZE** 134  
*[GUtility\(\)](#), sends CF, CW, EO etc. to initialize the connection. Useful after RS or other reset.*
- #define **G\_UTIL\_PING** 135  
*[GUtility\(\)](#), uses ICMP ping to determine if an IP address is reachable and assigned.*
- #define **G\_UTIL\_ERROR\_CONTEXT** 136  
*[GUtility\(\)](#), provides additional error context, where available.*
- #define **G\_UTIL\_GCAPS\_HOST** 256
- #define **G\_UTIL\_GCAPS\_VERSION** 257  
*[GUtility\(\)](#), get the version of the `gcaps` server.*
- #define **G\_UTIL\_GCAPS\_KEEPALIVE** 258

- GUtility()*, *Deprecated 20210119. No longer functional.*
- #define **G\_UTIL\_GCAPS\_ADDRESSES** 259
  - GUtility()*, *get a list of available connections from the gcaps server.*
- #define **G\_UTIL\_GCAPS\_IPREQUEST** 260
  - GUtility()*, *get a list of hardware requesting IPs from the gcaps server.*
- #define **G\_UTIL\_GCAPS\_ASSIGN** 261
  - GUtility()*, *assign IP addresses over the network from the gcaps server.*
- #define **G\_UTIL\_GCAPS\_PING** 262
  - GUtility()*, *uses ICMP ping to determine if an IP address is reachable and assigned. Ping sent from the gcaps server.*
- #define **G\_UTIL\_GCAPS\_LIST\_SERVERS** 263
  - GUtility()*, *get a list of all available gcaps servers on the local network.*
- #define **G\_UTIL\_GCAPS\_PUBLISH\_SERVER** 264
  - GUtility()*, *make local gcaps server discoverable by other gcaps servers on the local network.*
- #define **G\_UTIL\_GCAPS\_SET\_SERVER** 265
  - GUtility()*, *set the new active gcaps server.*
- #define **G\_UTIL\_GCAPS\_SERVER\_STATUS** 266
  - GUtility()*, *get information on the local server's name and if it is published to the local network.*
- #define **G\_UTIL\_GCAPS\_REMOTE\_CONNECTIONS** 267
  - GUtility()*, *get a list of remote addresses connected to local server.*
- #define **G\_UTIL\_GCAPS\_SERVER\_INFO** 268
- #define **G\_UTIL\_GCAPS\_ADDRESSES\_GET\_REMEMBERED** 269
  - GUtility()*, *returns true if gcaps is remembering ip assignments.*
- #define **G\_UTIL\_GCAPS\_ADDRESSES\_SET\_REMEMBERED** 270
  - GUtility()*, *sets if gcaps should remember ip assignments.*
- #define **G\_SMALL\_BUFFER** 1024
  - Most reads from Galil are small. This value will easily hold most, e.g. TH, TZ, etc.*
- #define **G\_HUGE\_BUFFER** 524288
  - Most reads from Galil hardware are small. This value will hold the largest array or program upload/download possible.*
- #define **G\_LINE\_BUFFER** 80
  - For writes, via command interpreter, to the Galil.*
- #define **GCLIB\_DLL\_EXPORTED**
- #define **GCALL** \_\_stdcall
- #define **MALLOCBUF** G\_HUGE\_BUFFER
  - Malloc used for large program and array uploads.*
- #define **MAXPROG** MALLOCBUF
  - Maximum size for a program.*
- #define **MAXARRAY** MALLOCBUF
  - Maximum size for an array table upload.*
- #define **POLLINGINTERVAL** 100
  - Interval, in milliseconds, for polling commands, e.g. GWaitForBool().*
- #define **G\_USE\_GCAPS**
  - Use the GCAPS server in GAddresses(), GAssign(), GIpRequests(), and GVersion(). To avoid GCAPS, comment out this line and recompile, <http://galil.com/sw/pub/all/doc/gclib/html/gclibo.html>.*
- #define **\_CRT\_SECURE\_NO\_WARNINGS**
- #define **GALILDATARECORDMAXLENGTH** 512
  - Max size for any Galil data record, equal to dual port ram size of PCI.*
- #define **G\_NO\_ERROR** 0
  - Return value if function succeeded.*
- #define **G\_NO\_ERROR\_S** "no error"
- #define **G\_GCLIB\_ERROR** -1

General library error. Indicates internal API caught an unexpected error. Contact [Galil support](mailto:softwaresupport@galil.com) if this error is returned, [softwaresupport@galil.com](mailto:softwaresupport@galil.com).

- #define **G\_GCLIB\_ERROR\_S** "gclib unexpected error"
- #define **G\_GCLIB\_UTILITY\_ERROR** -2
 

*An invalid request value was specified to GUtility.*
- #define **G\_GCLIB\_UTILITY\_ERROR\_S** "invalid request value or bad arguments were specified to [GUtility\(\)](#)"
- #define **G\_GCLIB\_UTILITY\_IP\_TAKEN** -3
 

*The IP cannot be assigned because ping returned a reply.*
- #define **G\_GCLIB\_UTILITY\_IP\_TAKEN\_S** "ip address is already taken by a device on the network"
- #define **G\_GCLIB\_NON\_BLOCKING\_READ\_EMPTY** -4
 

*GMessage, GInterrupt, and GRecord can be called with a zero timeout. If there wasn't data waiting in memory, this error is returned.*
- #define **G\_GCLIB\_NON\_BLOCKING\_READ\_EMPTY\_S** "data was not waiting for a zero-timeout read"
- #define **G\_GCLIB\_POLLING\_FAILED** -5
 

*GWaitForBool out of polling trials.*
- #define **G\_GCLIB\_POLLING\_FAILED\_S** "exit condition not met in specified polling period"
- #define **G\_TIMEOUT** -1100
 

*Operation timed out. Timeout is set by the `-timeout` option in [GOpen\(\)](#) and can be overridden by [GSetting\(\)](#).*
- #define **G\_TIMEOUT\_S** "device timed out"
- #define **G\_OPEN\_ERROR** -1101
 

*Device could not be opened. E.G. Serial port or PCI device already open.*
- #define **G\_OPEN\_ERROR\_S** "device failed to open"
- #define **G\_ALREADY\_OPEN** -1111
 

*Serial or PCI file has a flock placed on it, presumably by another gclib connection.*
- #define **G\_ALREADY\_OPEN\_S** "Serial or PCI port already open"
- #define **G\_READ\_ERROR** -1103
 

*Device read failed. E.G. Socket was closed by remote host. See [G\\_UTIL\\_GCAPS\\_KEEPALIVE](#).*
- #define **G\_READ\_ERROR\_S** "device read error"
- #define **G\_WRITE\_ERROR** -1104
 

*Device write failed. E.G. Socket was closed by remote host. See [G\\_UTIL\\_GCAPS\\_KEEPALIVE](#).*
- #define **G\_WRITE\_ERROR\_S** "device write error"
- #define **G\_INVALID\_PREPROCESSOR\_OPTIONS** -1204
 

*GProgramDownload was called with a bad preprocessor directive.*
- #define **G\_INVALID\_PREPROCESSOR\_OPTIONS\_S** "preprocessor did not recognize options"
- #define **G\_COMMAND\_CALLED\_WITH\_ILLEGAL\_COMMAND** -1106
 

*GCommand() was called with an illegal command, e.g. ED, DL or QD.*
- #define **G\_COMMAND\_CALLED\_WITH\_ILLEGAL\_COMMAND\_S** "illegal command passed to command call"
- #define **G\_DATA\_RECORD\_ERROR** -1107
 

*Data record error, e.g. DR attempted on serial connection.*
- #define **G\_DATA\_RECORD\_ERROR\_S** "data record error"
- #define **G\_UNSUPPORTED\_FUNCTION** -1109
 

*Function cannot be called on this bus. E.G. [GInterrupt\(\)](#) on serial.*
- #define **G\_UNSUPPORTED\_FUNCTION\_S** "function not supported on this communication bus"
- #define **G\_FIRMWARE\_LOAD\_NOT\_SUPPORTED** -1110
 

*Firmware is not supported on this bus, e.g. Ethernet for the DMC-21x3 series.*
- #define **G\_FIRMWARE\_LOAD\_NOT\_SUPPORTED\_S** "firmware cannot be loaded on this communication bus to this hardware"
- #define **G\_ARRAY\_NOT\_DIMENSIONED** -1200
 

*Array operation was called on an array that was not in the controller's array table, see LA command.*
- #define **G\_ARRAY\_NOT\_DIMENSIONED\_S** "array not dimensioned on controller or wrong size"
- #define **G\_CONNECTION\_NOT\_ESTABLISHED** -1201



- Function was called with no connection.*

  - #define **G\_CONNECTION\_NOT\_ESTABLISHED\_S** "connection to hardware not established"
  - #define **G\_ILLEGAL\_DATA\_IN\_PROGRAM** -1202

*Data to download not valid, e.g. \ in data.*

  - #define **G\_ILLEGAL\_DATA\_IN\_PROGRAM\_S** "illegal ASCII character in program"
  - #define **G\_UNABLE\_TO\_COMPRESS\_PROGRAM\_TO\_FIT** -1203

*Program preprocessor could not compress the program within the user's constraints.*

  - #define **G\_UNABLE\_TO\_COMPRESS\_PROGRAM\_TO\_FIT\_S** "program cannot be compressed to fit on the controller"
  - #define **G\_BAD\_RESPONSE\_QUESTION\_MARK** -10000

*Operation received a ?, indicating controller has a TC error.*

  - #define **G\_BAD\_RESPONSE\_QUESTION\_MARK\_S** "question mark returned by controller"
  - #define **G\_BAD\_VALUE\_RANGE** -10002

*Bad value or range, e.g. GCon g variable passed to function was bad.*

  - #define **G\_BAD\_VALUE\_RANGE\_S** "value passed to function was bad or out of range"
  - #define **G\_BAD\_FULL\_MEMORY** -10003

*Not enough memory for an operation, e.g. all connections allowed for a process already taken.*

  - #define **G\_BAD\_FULL\_MEMORY\_S** "operation could not complete because of a memory error"
  - #define **G\_BAD\_LOST\_DATA** -10004

*Lost data, e.g. GCommand() response buffer was too small for the controller's response.*

  - #define **G\_BAD\_LOST\_DATA\_S** "data was lost due to buffer or fifo limitations"
  - #define **G\_BAD\_FILE** -10005

*Bad file path, bad file contents, or bad write.*

  - #define **G\_BAD\_FILE\_S** "file was not found, contents are invalid, or write failed"
  - #define **G\_BAD\_ADDRESS** -10006

*Bad address.*

  - #define **G\_BAD\_ADDRESS\_S** "a bad address was specified in open"
  - #define **G\_BAD\_FIRMWARE\_LOAD** -10008

*Bad firmware upgrade.*

  - #define **G\_BAD\_FIRMWARE\_LOAD\_S** "Firmware upgrade failed"
  - #define **G\_GCAPS\_OPEN\_ERROR** -20000

*gcaps connection couldn't open. Server is not running or is not reachable.*

  - #define **G\_GCAPS\_OPEN\_ERROR\_S** "gcaps connection could not be opened"
  - #define **G\_GCAPS\_SUBSCRIPTION\_ERROR** -20002

*GMessage(), GRecord(), GInterrupt() called on a connection without –subscribe switch.*

  - #define **G\_GCAPS\_SUBSCRIPTION\_ERROR\_S** "function requires subscription not specified in GOpen()"

## Typedefs

- typedef int **GReturn**
  - Every function returns a value of type GReturn. See [gclib\\_errors.h](#) for possible values.*
- typedef void \* **GCon**
  - Connection handle. Unique for each connection in process. Assigned a non-zero value in [GOpen\(\)](#).*
- typedef unsigned int **GSize**
  - Size of buffers, etc.*
- typedef int **GOption**
  - Option integer for various formatting, etc.*
- typedef char \* **GCStringOut**
  - C-string output from the library. Implies null-termination.*
- typedef const char \* **GCStringIn**
  - C-string input to the library. Implies null-termination.*
- typedef char \* **GBufOut**

Data output from the library. No null-termination implied. Returned values may be null-terminated, see function documentation for details.

- typedef const char \* **GBufln**  
Data input to the library. No null-termination, function will have a *GSize* to indicate bytes to write .
- typedef unsigned char **GStatus**  
Interrupt status byte.
- typedef void \* **GMemory**  
Pointer to untyped memory for use in *GUtility()*.
- typedef uint8\_t **UB**
- typedef uint16\_t **UW**
- typedef int16\_t **SW**
- typedef int32\_t **SL**
- typedef uint32\_t **UL**

## Functions

- GCLIB\_DLL\_EXPORTED **GReturn GCALL GOpen** (*GCStringIn* address, *GCon* \*g)  
Open a connection to a *Galil* Controller.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GClose** (*GCon* g)  
Closes a connection to a *Galil* Controller.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GRead** (*GCon* g, *GBufOut* buffer, *GSize* buffer\_len, *GSize* \*bytes\_read)  
Performs a read on the connection.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GWrite** (*GCon* g, *GBufln* buffer, *GSize* buffer\_len)  
Performs a write on the connection.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GCommand** (*GCon* g, *GCStringIn* command, *GBufOut* buffer, *GSize* buffer\_len, *GSize* \*bytes\_returned)  
Performs a command-and-response transaction on the connection.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GProgramDownload** (*GCon* g, *GCStringIn* program, *GCStringIn* preprocessor)  
Downloads a program to the controller's program buffer.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GProgramUpload** (*GCon* g, *GBufOut* buffer, *GSize* buffer\_len)  
Uploads a program from the controller's program buffer.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GArrayDownload** (*GCon* g, const *GCStringIn* array\_name, *GOption* first, *GOption* last, *GCStringIn* buffer)  
Downloads array data to a pre-dimensioned array in the controller's array table.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GArrayUpload** (*GCon* g, const *GCStringIn* array\_name, *GOption* first, *GOption* last, *GOption* delim, *GBufOut* buffer, *GSize* buffer\_len)  
Uploads array data from the controller's array table.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GRecord** (*GCon* g, union *GDataRecord* \*record, *GOption* method)  
Provides a fresh copy of the controller's data record. Data is cast into a union, *GDataRecord*.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GMessage** (*GCon* g, *GCStringOut* buffer, *GSize* buffer\_len)  
Provides access to unsolicited messages from the controller.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GInterrupt** (*GCon* g, *GStatus* \*status\_byte)  
Provides access to PCI and UDP interrupts from the controller.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GFirmwareDownload** (*GCon* g, *GCStringIn* filepath)  
Upgrade firmware.
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GUtility** (*GCon* g, *GOption* request, *GMemory* memory1, *GMemory* memory2)  
Provides read/write access to driver settings and convenience features based on the request variable.
- GCLIB\_DLL\_EXPORTED void **GCALL GSleep** (unsigned int timeout\_ms)  
Uses *GUtility()* and *G\_UTIL\_SLEEP* to provide a blocking sleep call which can be useful for timing-based chores.

- GCLIB\_DLL\_EXPORTED GReturn GCALL GVersion (GCStringOut ver, GSize ver\_len)  
*Uses GUtility(), G\_UTIL\_VERSION and G\_UTIL\_GCAPS\_VERSION to provide the library and gcaps version numbers.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GAddresses (GCStringOut addresses, GSize addresses\_len)  
*Uses GUtility(), G\_UTIL\_GCAPS\_ADDRESSES or G\_UTIL\_ADDRESSES to provide a listing of all available connection addresses.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GInfo (GCon g, GCStringOut info, GSize info\_len)  
*Uses GUtility() and G\_UTIL\_INFO to provide a useful connection string.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GTimeout (GCon g, short timeout\_ms)  
*Uses GUtility() and G\_UTIL\_TIMEOUT\_OVERRIDE to set the library timeout.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GCmd (GCon g, GCStringIn command)  
*Wrapper around GCommand for use when the return value is not desired.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GCmdT (GCon g, GCStringIn command, GCStringOut trimmed\_response, GSize response\_len, GCStringOut \*front)  
*Wrapper around GCommand that trims the response.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GCmdI (GCon g, GCStringIn command, int \*value)  
*Wrapper around GCommand that provides the return value of a command parsed into an int.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GCmdD (GCon g, GCStringIn command, double \*value)  
*Wrapper around GCommand that provides the return value of a command parsed into a double.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GWaitForBool (GCon g, GCStringIn predicate, int trials)  
*Blocking call that returns when the controller evaluates the predicate as true.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GMotionComplete (GCon g, GCStringIn axes)  
*Blocking call that returns once all axes specified have completed their motion.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GRecordRate (GCon g, double period\_ms)  
*Sets the asynchronous data record to a user-specified period via DR.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GProgramDownloadFile (GCon g, GCStringIn file\_path, GCStringIn preprocessor)  
*Program download from file.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GProgramUploadFile (GCon g, GCStringIn file\_path)  
*Program upload to file.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GArrayDownloadFile (GCon g, GCStringIn file\_path)  
*Array download from file.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GArrayUploadFile (GCon g, GCStringIn file\_path, GCStringIn names)  
*Array upload to file.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GIpRequests (GCStringOut requests, GSize requests\_len)  
*Uses GUtility(), G\_UTIL\_GCAPS\_IPREQUEST or G\_UTIL\_IPREQUEST to provide a list of all Galil controllers requesting IP addresses via BOOT-P or DHCP.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GSetServer (GCStringIn server\_name)  
*Uses GUtility(), G\_UTIL\_GCAPS\_SET\_SERVER to set the new active server.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GListServers (GCStringOut servers, GSize servers\_len)  
*Uses GUtility(), G\_UTIL\_GCAPS\_LIST\_SERVERS to provide a list of all available gcaps services on the local network.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GPublishServer (GCStringIn name, GOption publish, GOption save)  
*Uses GUtility(), G\_UTIL\_GCAPS\_PUBLISH\_SERVER to publish local gcaps server to the local network.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GServerStatus (GCStringOut status, GSize status\_len)  
*Uses GUtility(), G\_UTIL\_GCAPS\_SERVER\_STATUS to get information on the local server name and if it is published to the local network.*
- GCLIB\_DLL\_EXPORTED GReturn GCALL GRemoteConnections (GCStringOut connections, GSize connections\_length)  
*Uses GUtility(), G\_UTIL\_GCAPS\_REMOTE\_CONNECTIONS to get a list of remote addresses connected to the local server.*

- GCLIB\_DLL\_EXPORTED [GReturn](#) [GCALL](#) [GAssign](#) ([GCStringIn](#) ip, [GCStringIn](#) mac)  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_ASSIGN](#) or [G\\_UTIL\\_ASSIGN](#) to assign an IP address over the Ethernet to a controller at a given MAC address.*
- GCLIB\_DLL\_EXPORTED void [GCALL](#) [GError](#) ([GReturn](#) rc, [GCStringOut](#) error, [GSize](#) error\_len)  
*Provides a human-readable description string for return codes.*
- GCLIB\_DLL\_EXPORTED [GReturn](#) [GCALL](#) [GSetupDownloadFile](#) ([GCon](#) g, [GCStringIn](#) file\_path, [GOption](#) options, [GCStringOut](#) info, [GSize](#) info\_len)  
*Download a saved controller configuration from a file.*

## 10.1.2 Function Documentation

### 10.1.2.1 GAddresses()

```
GReturn GCALL GAddresses (
    GCStringOut addresses,
    GSize addresses_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_ADDRESSES](#) or [G\\_UTIL\\_ADDRESSES](#) to provide a listing of all available connection addresses.

#### Note

Serial ports are listed, e.g. COM1. Upon open, it may be necessary to specify a baud rate for the controller, e.g. `--baud 19200`. Default baud is 115200. See [GOpen\(\)](#).

#### Parameters

<i>addresses</i>	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so. See below for more information.
<i>addresses_len</i>	Length of buffer.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

If [gcaps](#) is available, the listing will come from the server via [G\\_UTIL\\_GCAPS\\_ADDRESSES](#). In the absence of the server, [gclib](#) will use [G\\_UTIL\\_ADDRESSES](#) to generate the list.

- Ethernet controllers will be listed as *ip\_address, revision\_report, network\_adapter\_name, network\_adapter←\_ip\_address*. If an IP address is unreachable via ping, the address will be in parentheses.
- PCI controllers will be listed by their identifier, e.g. GALILPCI1.
- Serial ports will be listed by their identifier, e.g. COM1.

```
10.1.3.91, DMC4020 Rev 1.2e, LAN, 10.1.3.10
192.168.0.63, DMC4040 Rev 1.2f, Static, 192.168.0.41
(192.0.0.42), RIO47102 Rev 1.1j, Static, 192.168.0.41
GALILPCI1
COM1
COM2
```

#### Note

[GAddresses\(\)](#) will take up to 1 second to look for [gcaps](#).

See [x\\_examples.cpp](#) for an example.

Definition at line 54 of file [gclibo.c](#).

References [G\\_NO\\_ERROR](#), [G\\_UTIL\\_ADDRESSES](#), [G\\_UTIL\\_GCAPS\\_ADDRESSES](#), and [GUtility\(\)](#).

### 10.1.2.2 GArrayDownload()

```
GCLIB_DLL_EXPORTED GReturn GCALL GArrayDownload (
    GCon g,
    const GCStringIn array_name,
    GOption first,
    GOption last,
    GCStringIn buffer )
```

Downloads array data to a pre-dimensioned array in the controller's array table.

#### Warning

The array must already exist on the controller and be sufficient dimension to hold the desired array data, e.g. via DM.

#### Parameters

<i>g</i>	Connection's handle.
<i>array_name</i>	Null-terminated string containing the name of the array to download. Must match the array name used in DM.
<i>first</i>	The first element of the array for sub-array downloads. <code>G_BOUNDS</code> to omit.
<i>last</i>	The last element of the array for sub-array downloads. <code>G_BOUNDS</code> to omit.
<i>buffer</i>	Buffer containing the null-terminated data to be sent to the controller. The array data may be separated with <i>carriage return</i> , <i>carriage return + line feed</i> , or a <i>comma</i> . No spaces.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_arrays.cpp](#) for an example.

### 10.1.2.3 GArrayDownloadFile()

```
GCLIB_DLL_EXPORTED GReturn GCALL GArrayDownloadFile (
    GCon g,
    GCStringIn file_path )
```

Array download from file.

Downloads a csv file containing array data at `file_path`. If the arrays don't exist, they will be dimensioned.

#### Parameters

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the array file.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_arrays.cpp](#) for an example.

Definition at line 380 of file `arrays.c`.

### 10.1.2.4 GArrayUpload()

```
GCLIB_DLL_EXPORTED GReturn GCALL GArrayUpload (
    GCon g,
    const GCStringIn array_name,
    GOption first,
```

```

    GOption last,
    GOption delim,
    GBufOut buffer,
    GSize buffer_len )

```

Uploads array data from the controller's array table.

#### Parameters

<i>g</i>	Connection's handle.
<i>array_name</i>	Null-terminated string containing the name of the array to upload.
<i>first</i>	The first element of the array for sub-array uploads. <code>G_BOUNDS</code> to omit.
<i>last</i>	The last element of the array for sub-array uploads. <code>G_BOUNDS</code> to omit.
<i>delim</i>	Sets the delimiter between array elements in the returned data, <code>G_CR</code> specifies carriage return, <code>G_COMMA</code> specifies comma.
<i>buffer</i>	Buffer to receive the uploaded data. The data will be null terminated unless function returns <code>G_BAD_LOST_DATA</code> due to the buffer being too small to hold the data.
<i>buffer_len</i>	The length of the receive buffer.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_arrays.cpp](#) for an example.

#### 10.1.2.5 GArrayUploadFile()

```

GCLIB_DLL_EXPORTED GReturn GCALL GArrayUploadFile (
    GCon g,
    GCStringIn file_path,
    GCStringIn names )

```

Array upload to file.

Uploads the entire controller array table or a subset and saves the data as a csv file specified by *file\_path*.

#### Parameters

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the array file, file will be overwritten if it exists.
<i>names</i>	Null-terminated string containing the arrays to upload, delimited with space. "" or null uploads all arrays listed in LA.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_arrays.cpp](#) for an example.

Definition at line 408 of file arrays.c.

#### 10.1.2.6 GAssign()

```

GReturn GCALL GAssign (
    GCStringIn ip,
    GCStringIn mac )

```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_ASSIGN](#) or [G\\_UTIL\\_ASSIGN](#) to assign an IP address over the Ethernet to a controller at a given MAC address.

## Parameters

<i>ip</i>	The null-terminated ip address to assign. The hardware should not yet have an IP address.
<i>mac</i>	The null-terminated MAC address of the hardware.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

On Linux and Mac, the desired IP address will be pinged prior to the assignment. If the ping is returned, [GAssign\(\)](#) will return [G\\_GCLIB\\_UTILITY\\_IP\\_TAKEN](#).

If [gcaps](#) is available, the assign will be performed from the server via [G\\_UTIL\\_GCAPS\\_ASSIGN](#). [gcaps](#) will remember the assignment and will automatically assign the desired IP address should the controller ever request one again, e.g. after a controller master reset. To clear the remembered IP address from [gcaps](#), call [GAssign\(\)](#) with a blank string in place of the ip address. To remove all remembered ip addresses, specify a blank string for the mac address.

In the absence of the server, [gclib](#) will use [G\\_UTIL\\_ASSIGN](#) to assign. [GAssign\(\)](#) will take up to 1 second to look for [gcaps](#). When not using [gcaps](#), Linux/OS X users must be root to use [GAssign\(\)](#) and have UDP access to send on port 68.

See [x\\_examples.cpp](#) for an example.

Definition at line 70 of file [gclibo.c](#).

References [G\\_GCLIB\\_UTILITY\\_IP\\_TAKEN](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_ASSIGN](#), [G\\_UTIL\\_GCAPS\\_ASSIGN](#), [G\\_UTIL\\_GCAPS\\_PING](#), [G\\_UTIL\\_PING](#), and [GUtility\(\)](#).

## 10.1.2.7 GClose()

```
GCLIB_DLL_EXPORTED GReturn GCALL GClose (
    GCon g )
```

Closes a connection to a [Galil](#) Controller.

## Attention

[gclib](#) requires that [GClose\(\)](#) be called whenever a program is finished with a controller. This includes when a program closes. A rule of thumb is that for every [GOpen\(\)](#) call on a given connection, a [GClose\(\)](#) call should be found on every code path. Failing to call [GClose\(\)](#) may cause controller resources to not be released or can hang the process if there are outstanding asynchronous operations. The latter can occur, for example, if a call to [GRead\(\)](#) times out and the process exits without calling [GClose\(\)](#). In this case, [GRead\(\)](#) still has an outstanding asynchronous read pending. [GClose\(\)](#) will terminate this operation allowing the process to exit correctly.

## Parameters

<i>g</i>	Connection's handle.
----------	----------------------

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_examples.cpp](#) for an example.

## 10.1.2.8 GCmd()

```
GReturn GCALL GCmd (
    GCon g,
    GCStringIn command )
```

Wrapper around [GCommand](#) for use when the return value is not desired.

The returned data is still checked for error, e.g. ? or timeout, but is not brought out through the prototype.

**Parameters**

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 237 of file `gclibo.c`.

**10.1.2.9 GCmdD()**

```
GReturn GCALL GCmdD (
    GCon g,
    GCStringIn command,
    double * value )
```

Wrapper around GCommand that provides the return value of a command parsed into a double.

Use this function to retrieve the full [Galil 4.2](#) range, e.g. for a variable value with fractional data, or the value of an Analog input or Output.

**Parameters**

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.
<i>value</i>	Pointer to a double that will be filled with the return value.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 289 of file `gclibo.c`.

**10.1.2.10 GCmdI()**

```
GReturn GCALL GCmdI (
    GCon g,
    GCStringIn command,
    int * value )
```

Wrapper around GCommand that provides the return value of a command parsed into an int.

Use this function to get most values including TP, RP, TE, Digital I/O states, etc.

**Parameters**

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.
<i>value</i>	Pointer to an int that will be filled with the return value.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 278 of file `gclibo.c`.



### 10.1.2.11 GCmdT()

```
GReturn GCALL GCmdT (
    GCon g,
    GCStringIn command,
    GCStringOut trimmed_response,
    GSize response_len,
    GCStringOut * front )
```

Wrapper around GCommand that trims the response.

For use when the return value is desired, is ASCII (not binary), and the response should be trimmed of trailing colon, whitespace, and optionally leading space.

#### Parameters

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.
<i>trimmed_response</i>	The trimmed response from the controller. Trailing space is trimmed by null terminating any trailing spaces, carriage returns, or line feeds.
<i>response_len</i>	The length of the trimmed_response buffer.
<i>front</i>	If non-null, upon return *front will point to the first non-space character in trimmed_response. This allows trimming the front of the string without modifying the user's buffer pointer, which may be allocated on the heap.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 243 of file gclibo.c.

### 10.1.2.12 GCommand()

```
GCLIB_DLL_EXPORTED GReturn GCALL GCommand (
    GCon g,
    GCStringIn command,
    GBufOut buffer,
    GSize buffer_len,
    GSize * bytes_returned )
```

Performs a *command-and-response* transaction on the connection.

#### Parameters

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller. The library will append a carriage return to the command string.
<i>buffer</i>	Buffer for the response. Will be filled with the response from the controller. The data will be null terminated unless the function returns G_BAD_LOST_DATA due to the buffer being too small to hold the data.
<i>buffer_len</i>	The size of the response buffer.
<i>bytes_returned</i>	The size of the data returned from the controller. This does not include null termination. This argument may be null if the value is not desired.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

**10.1.2.13 GError()**

```
void GCALL GError (
    GReturn rc,
    GCStringOut error,
    GSize error_len )
```

Provides a human-readable description string for return codes.

**Parameters**

<i>rc</i>	The return code to lookup.
<i>error</i>	The buffer to fill with the error text. Buffer will be null terminated, even if the data must be truncated to do so.
<i>error_len</i>	The length of the error buffer.

See [x\\_examples.cpp](#) for an example.

Definition at line 459 of file `gclibo.c`.

References `G_NO_ERROR`.

**10.1.2.14 GFirmwareDownload()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GFirmwareDownload (
    GCon g,
    GCStringIn filepath )
```

Upgrade firmware.

**Parameters**

<i>g</i>	Connection's handle.
<i>filepath</i>	The full file path to the Galil-supplied firmware hex file. See <a href="http://www.galil.com/downloads/firmware">http://www.galil.com/downloads/firmware</a>

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

```
ec(GInfo(g, buf, sizeof(buf))); //get controller info
cout << buf << '\n'; //print the info
ec(GFirmwareDownload(g, "F:/1806.dmc/dmc-1806-r11a.hex"));
ec(GInfo(g, buf, sizeof(buf))); //get the info again
cout << buf << '\n';
// example output:
// GALILPCI1, DMC1846 Rev 1.1a-CM, 4232
// GALILPCI1, DMC1846 Rev 1.1a, 4232
```

**10.1.2.15 GInfo()**

```
GReturn GCALL GInfo (
    GCon g,
    GCStringOut info,
    GSize info_len )
```

Uses [GUtility\(\)](#) and [G\\_UTIL\\_INFO](#) to provide a useful connection string.

## Parameters

<i>g</i>	Connection's handle.
<i>info</i>	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so.
<i>info_len</i>	Length of buffer.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

The response is *address*, *revision\_report*, *serial\_number*. For example:

```
COM2, RIO47102 Rev 1.1j, 37290
```

See [x\\_examples.cpp](#) for an example.

Definition at line 49 of file `gclibo.c`.

## 10.1.2.16 GInterrupt()

```
GCLIB_DLL_EXPORTED GReturn GCALL GInterrupt (
    GCon g,
    GStatus * status_byte )
```

Provides access to PCI and UDP interrupts from the controller.

Interrupts can be generated automatically by the firmware on important events via `EI` (Enable Interrupt) or by the user in embedded DMC code via `UI` (User Interrupt). To use this function, `-s EI` must be used in the `GOpen()` address string to subscribe to interrupts.

## Parameters

<i>g</i>	Connection's handle.
<i>status_byte</i>	A pointer to a <code>GStatus</code> to receive the status byte.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

`GInterrupt()` will block until an interrupt is received, or the function times out.

## Note

If this function is called with a timeout of zero, a non-blocking read is performed. If interrupt data is waiting in the interrupt queue, the oldest byte will be popped off the queue. If there is no interrupt data queued, but there is data waiting in the socket or PCI FIFO, one read will be performed to process the waiting data. If new data is still not found after these two attempts, `G_GCLIB_NON_BLOCKING_READ_EMPTY` will be returned.

See [x\\_ginterrupt.cpp](#) for an example. See [x\\_nonblocking.cpp](#) for an example of non-blocking usage.

## 10.1.2.17 GIpRequests()

```
GReturn GCALL GIpRequests (
    GCStringOut requests,
    GSize requests_len )
```

Uses `GUtility()`, `G_UTIL_GCAPS_IPREQUEST` or `G_UTIL_IPREQUEST` to provide a list of all `Galil` controllers requesting IP addresses via BOOT-P or DHCP.

## Parameters

<i>requests</i>	The buffer to hold the list of requesting controllers. Data will be null terminated, even if the data must be truncated to do so. See below for more information.
<i>requests_len</i>	The length of the requests buffer.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

[GlpRequests\(\)](#) will block up to 5 seconds while listening for requests.

If [gcaps](#) is available, the listing will come from the server via [G\\_UTIL\\_GCAPS\\_IPREQUEST](#). In the absence of the server, [gclib](#) will use [G\\_UTIL\\_IPREQUEST](#) to generate the list. [GlpRequests\(\)](#) will take up to 1 second to look for [gcaps](#). When not using [gcaps](#), Linux/OS X users must be root to use [GlpRequests\(\)](#) and have UDP access to bind and listen on port 67.

Each line of the returned data will be of the form *model, serial\_number, MAC\_address, network\_adapter\_name, network\_adapter\_ip\_address, remembered\_ip\_assignment*. See [GAssign\(\)](#) for more information about remembered IP assignments. The following is an example output.

```
DMC2000, 34023, 00:50:4C:00:84:E7, enp5s0, 192.168.42.92, 192.168.42.200
DMC2105, 7, 00:50:4C:58:00:07, enp5s0, 192.168.42.92, 0.0.0.0
DMC2105, 13, 00:50:4C:58:00:0D, enp5s0, 192.168.42.92, 0.0.0.0
```

See [x\\_examples.cpp](#) for an example.

Definition at line 106 of file [gclibo.c](#).

References [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_IPREQUEST](#), [G\\_UTIL\\_IPREQUEST](#), [GSleep\(\)](#), and [GUtility\(\)](#).

Referenced by [ip\\_assigner\(\)](#).

**10.1.2.18 GListServers()**

```
GReturn GCALL GListServers (
    GCStringOut servers,
    GSize servers_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_LIST\\_SERVERS](#) to provide a list of all available gcaps services on the local network.

**Note**

This function is only available on Windows 10 and Linux.

**Parameters**

<i>servers</i>	The buffer to hold the list of available gcaps servers
<i>servers_len</i>	The length of the servers buffer

This function is used to find a list of available gcaps servers that have made themselves "Discoverable".

The list of available servers are separated by a newline '\n' character.

**Attention**

This function will always use your local gcaps server, regardless of which server you have set as your active server.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 169 of file [gclibo.c](#).

References [G\\_GCAPS\\_OPEN\\_ERROR](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_LIST\\_SERVERS](#), and [GUtility\(\)](#).

**10.1.2.19 GMessage()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GMessage (
    GCon g,
    GCStringOut buffer,
    GSize buffer_len )
```

Provides access to unsolicited messages from the controller.

To use this function, `-s MG` must be used in the `GOpen()` `address` string to subscribe to messages. Unsolicited bytes must be flagged by the high-bit setting, `CW 1`. The driver will automatically set this when subscribing to messages. The user should not overwrite this setting.

Unsolicited messages are data generated by the controller that are not in response to a command, a data record, or an interrupt. Examples follow.

1. Data generated by the `MG` command from embedded code. `MG` sent from the host is solicited.
2. Any command in an embedded program that returns data, e.g. `TP, RP, var=?`
3. A run time error in an embedded program, e.g. `?55 i=var`

#### Note

Messages are unframed byte streams. There is no guarantee that the user will get complete messages or single messages in a call to `GMessage()`. If multiple messages have been sent from the controller since the last call to `GMessage()`, they will all be placed in the buffer, separated by newline characters.

#### Parameters

<code>g</code>	Connection's handle.
<code>buffer</code>	The buffer to write the message data. The buffer will be null terminated.
<code>buffer_len</code>	The length of the user's buffer.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

`GMessage()` will block until a message is received, or the function times out.

#### Note

If this function is called with a timeout of zero, a non-blocking read is performed. If message data has been processed since the last time the function was called, this data will be returned. If there is no processed message data, but there is data waiting in the socket or PCI FIFO, one read will be performed to process the waiting data. If new data is still not found after these two attempts, `G_GCLIB_NON_BLOCKING_READ_↔EMPTY` will be returned.

#### Warning

When sending message streams through `gcaps`, the following non-printable bytes are illegal, `$00-$07` and `$10-$17`. These bytes may be routed to a third party device such as an HMI or display panel. See `MG` and `CF`.

See [x\\_gmessage.cpp](#) for an example. See [x\\_nonblocking.cpp](#) for an example of non-blocking usage.

#### 10.1.2.20 GMotionComplete()

```
GReturn GCALL GMotionComplete (
    GCon g,
    GCStringIn axes )
```

Blocking call that returns once all axes specified have completed their motion.

#### Note

This function uses a profiled motion indicator, not the position of the encoder. E.G. see the difference between `AM` (profiled) and `MC` (encoder-based).

Although using the `_BGm` operand is the most generally compatible method, there are higher-performance ways to check for motion complete by using the data record, or interrupts. See examples [x\\_dr\\_motioncomplete\(\)](#) and [x\\_ei\\_motioncomplete\(\)](#).

## Parameters

<i>g</i>	Connection's handle.
<i>axes</i>	A null-terminated string containing a multiple-axes mask. Every character in the string should be a valid argument to MG_BGm, i.e. XYZWABCDEFHGST.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gmotioncomplete.cpp](#) for an example.

Definition at line 300 of file [gclibo.c](#).

## 10.1.2.21 GOpen()

```
GCLIB_DLL_EXPORTED GReturn GCALL GOpen (
    GCStringIn address,
    GCon * g )
```

Open a connection to a [Galil](#) Controller.

## Parameters

<i>address</i>	Null-terminated address string. See table below.
<i>g</i>	Pointer to user's <i>GCon</i> variable. On success, the library will fill the user's variable with the handle to use for the rest of the connection. A valid <i>g</i> value is nonzero.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

address switch	Meaning	Arguments (default), other options	Examples
--address	<b>Simple address to hardware</b>	<i>IP address, PCI, COM port</i>	--address COM1
-a	shorthand for --address	See <i>Address Ranges</i> below	-a GALILPCI1
{no switch}	--address is implicit for any lone token		192.168.0.42
--baud	<b>Baud rate</b>	(115200), <i>valid baud...</i>	COM2 --baud 19200
-b	shorthand for --baud		COM3 -b 38400
--command	<b>Command-and-response socket protocol</b>	(TCP), UDP	192.168.0.42 --command TCP
-c	shorthand for --command		192.168.0.42 -c UDP
--direct	<b>Connect directly to hardware instead of via <a href="#">gcaps</a></b>		-a GALILPCI2 --direct
-d	shorthand for --direct		GALILPCI2 -d
--subscribe	<b>Subscribe to messages, data records, and/or interrupts</b>	(NONE), MG, DR, EI, ALL	192.168.0.42 --subscribe MG
-s	shorthand for --subscribe		192.168.0.42 -s DR -s EI
--timeout	<b>timeout in ms</b>	(5000), <i>0-65535</i>	192.168.0.42 --timeout 5000

address switch	Meaning	Arguments (default), other options	Examples
-t	shorthand for --timeout		GALILPCI2 -t 500
--unsolicited	<b>Unsolicited socket protocol</b>	(UDP), NONE	192.168.0.42 --unsolicited NONE
-u	shorthand for --unsolicited		192.168.1.42 -u UDP
<b>The following address switches are deprecated and will be unavailable starting July 1st, 2020.</b>			
--p1	<b>Primary port for command-and-response traffic</b>	(23), valid port number	192.168.0.42 --p1 5000
--p2	<b>Secondary port for unsolicited traffic</b>	(60007), valid port number	192.168.0.42 --p2 5000

Operating System	Address Range	Notes
Windows	COM1 - COM256	RS232 and USB-to-serial
Linux	/dev/ttyS0 - /dev/ttyS255	RS232
Linux	/dev/ttyUSB0 - /dev/ttyUSB255	USB-to-serial, e.g. DMC-4103
Windows	GALILPCI1 - GALILPCI8	PCI
Linux	/dev/galilpci0 - /dev/galilpci7	PCI

See [x\\_examples.cpp](#) for an example.

When connecting to a network device, if the command-and-response socket is opened successfully but the unsolicited socket fails, `GOpen()` will still complete successfully. This allows connection to a [Galil](#) controller when only one Ethernet handle is available. Unsolicited traffic will not be accessible in this case.

### 10.1.2.22 GProgramDownload()

```
GCLIB_DLL_EXPORTED GReturn GCALL GProgramDownload (
    GCon g,
    GCStringIn program,
    GCStringIn preprocessor )
```

Downloads a program to the controller's program buffer.

#### Parameters

<i>g</i>	Connection's handle.
<i>program</i>	Null-terminated program for download.
<i>preprocessor</i>	Options string for preprocessing the program before sending it to the controller. Null allows the library to use defaults for the download. See the <a href="#">Program Preprocessor</a> documentation for options.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_programs.cpp](#) for an example.

### 10.1.2.23 GProgramDownloadFile()

```
GReturn GCALL GProgramDownloadFile (
    GCon g,
```

```

    GCStringIn file_path,
    GCStringIn preprocessor )

```

Program download from file.

#### Parameters

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the program file.
<i>preprocessor</i>	Options string for preprocessing the program before sending it to the controller. See <a href="#">GProgramDownload()</a> .

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_programs.cpp](#) for an example.

Definition at line 387 of file `gclibo.c`.

#### 10.1.2.24 GProgramUpload()

```

GCLIB_DLL_EXPORTED GReturn GCALL GProgramUpload (
    GCon g,
    GBufOut buffer,
    GSize buffer_len )

```

Uploads a program from the controller's program buffer.

#### Parameters

<i>g</i>	Connection's handle.
<i>buffer</i>	Buffer to receive the controller's program. The data will be null terminated unless function returns <code>G_BAD_LOST_DATA</code> due to the buffer being too small to hold the data.
<i>buffer_len</i>	The length of the receive buffer.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_programs.cpp](#) for an example.

#### 10.1.2.25 GProgramUploadFile()

```

GReturn GCALL GProgramUploadFile (
    GCon g,
    GCStringIn file_path )

```

Program upload to file.

#### Parameters

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the program file, file will be overwritten if it exists.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_programs.cpp](#) for an example.



Definition at line 430 of file gclibo.c.

### 10.1.2.26 GPublishServer()

```
GReturn GCALL GPublishServer (
    GCStringIn name,
    GOption publish,
    GOption save )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_PUBLISH\\_SERVER](#) to publish local gcaps server to the local network.

#### Note

This function is only available on Windows 10 and Linux.

#### Parameters

<i>name</i>	The name of the server to publish or remove
<i>publish</i>	Option to publish or remove server from network
<i>save</i>	Option to save this configuration for future reboots

This function is used to make your local gcaps server "Discoverable" or "Invisible"

publish Option:

Set to 1 to publish server to the network and make "Discoverable"

Set to 0 to remove server from the network and make "Invisible"

save Option:

Set to 1 to save the configuration for future reboots of the server

Set to 0 to use this configuration once, and not overwrite previous server settings

#### Attention

This function will always use your local gcaps server, regardless of which server you have set as your active server.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 189 of file gclibo.c.

Referenced by [remote\\_server\(\)](#).

### 10.1.2.27 GRead()

```
GCLIB_DLL_EXPORTED GReturn GCALL GRead (
    GCon g,
    GBufOut buffer,
    GSize buffer_len,
    GSize * bytes_read )
```

Performs a read on the connection.

#### Parameters

<i>g</i>	Connection's handle.
<i>buffer</i>	The user's read buffer.
<i>buffer_len</i>	The length of the user's read buffer.
<i>bytes_read</i>	Pointer to a GSize which will be filled with the number of bytes read upon return.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

**Warning**

This function is deprecated and will be removed in a future gclib version. Please contact [Galil](#) for needs not covered by the other gclib functions.

Unsolicited messages may be returned in the read data. The high bit of each message byte will be set unless the user changes the CW setting. Interrupts and Data Records are always filtered from a read. See [x\\_gread\\_gwrite.cpp](#) for an example.

**10.1.2.28 GRecord()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GRecord (
    GCon g,
    union GDataRecord * record,
    GOption method )
```

Provides a fresh copy of the controller's data record. Data is cast into a union, [GDataRecord](#).

**Parameters**

<i>g</i>	Connection's handle.
<i>record</i>	A pointer to the user's DataRecord union to hold the copy.
<i>method</i>	Determines the method for acquiring the data. <ul style="list-style-type: none"> <li>• G_QR: QR is used via command-and-response.</li> <li>• G_DR: DR is used for asynchronous acquisition.</li> </ul>

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

When using G\_DR, the asynchronous data record must already be set up.

- `-s DR` must be used in the [GOpen\(\)](#) `address` string to subscribe to records. The driver will automatically set the second argument of `DR`, where applicable.
- [GRecordRate\(\)](#) should be issued to set `DR` to an appropriate interval, `n`. The interval must be no faster than the rate at which [GRecord\(\)](#) is called.
- If [GRecord\(\)](#) is called more slowly than the data record rate, stale data will be returned until [GRecord\(\)](#) has been called once for each record sent by the controller.

[GRecord\(\)](#) will block until the data record is received, or the transaction times out.

**Note**

If this function is called with a timeout of zero and the G\_DR method, a non-blocking read is performed. If a data record has been processed since the last time the function was called, this data will be returned. If there is not a processed data record, but there is data waiting in the socket or PCI FIFO, one read will be performed to process the waiting data. If new data is still not found after these two attempts, G\_GCLIB\_↔ NON\_BLOCKING\_READ\_EMPTY will be returned.

See [x\\_grecord.cpp](#) for an example. See [x\\_nonblocking.cpp](#) for an example of non-blocking usage.

**10.1.2.29 GRecordRate()**

```
GReturn GCALL GRecordRate (
    GCon g,
    double period_ms )
```

Sets the asynchronous data record to a user-specified period via `DR`.

Takes `TM` and product type into account and sets the `DR` period to the period requested by the user, if possible.

#### Parameters

<i>g</i>	Connection's handle.
<i>period_ms</i>	Period, in milliseconds, to set up for the asynchronous data record.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_grecord.cpp](#) for an example.

Definition at line 342 of file `gclibo.c`.

#### 10.1.2.30 GRemoteConnections()

```
GReturn GCALL GRemoteConnections (
    GCStringOut connections,
    GSize connections_length )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_REMOTE\\_CONNECTIONS](#) to get a list of remote addresses connected to the local server.

#### Note

This function is only available on Windows 10 and Linux.

#### Parameters

<i>connections</i>	The buffer to hold the list of remote IP addresses currently connected to your hardware
<i>connections_len</i>	The length of the connections buffer

This function is used to find a list of IP Addresses of machines that currently have open connections to your local hardware. If another user sets your local server as their active server, and then opens a connection to your hardware, their IP Address will appear in this list.

The list of IP addresses are separated by a newline '\n' character.

#### Attention

This function will always use your local `gcaps` server, regardless of which server you have set as your active server.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 217 of file `gclibo.c`.

References [G\\_GCAPS\\_OPEN\\_ERROR](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_REMOTE\\_CONNECTIONS](#), and [GUtility\(\)](#).

#### 10.1.2.31 GServerStatus()

```
GReturn GCALL GServerStatus (
    GCStringOut status,
    GSize status_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_SERVER\\_STATUS](#) to get information on the local server name and if it is published to the local network.

**Note**

This function is only available on Windows 10 and Linux.

**Parameters**

<i>status</i>	The buffer to hold the status of the local gcaps server
<i>status_len</i>	The length of the status buffer

This function is used to find the status of your local gcaps server. Use this function to determine the name your server is currently using, and whether or not your gcaps server is currently set to "Discoverable" or "Invisible" The status buffer will be filled in the form of "[Server Name], [Discoverable]" For example, for a server with the name "Example Server" that is set to "Discoverable", the status buffer would contain "Example Server, true".

**Attention**

This function will always use your local gcaps server, regardless of which server you have set as your active server.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 149 of file gclibo.c.

References `G_GCAPS_OPEN_ERROR`, `G_NO_ERROR`, `G_UTIL_GCAPS_SERVER_STATUS`, and `GUtility()`.

**10.1.2.32 GSetServer()**

```
GReturn GCALL GSetServer (
    GCStringIn server_name )
```

Uses [GUtility\(\)](#), `G_UTIL_GCAPS_SET_SERVER` to set the new active server.

**Note**

This function is only available on Windows 10 and Linux.

**Parameters**

<i>server_name</i>	The name of the server to set as your new active server.
--------------------	--

Use this function in conjunction with [GListServers\(\)](#). Choose a name received from [GListServers\(\)](#) to set as your new active server.

After setting a new active server, all gclib calls will route through that new active server, unless explicitly noted otherwise.

To set your active server back to your local server, simply pass "Local" to [GSetServer\(\)](#):

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 128 of file gclibo.c.

References `G_GCAPS_OPEN_ERROR`, `G_NO_ERROR`, `G_UTIL_GCAPS_SET_SERVER`, and `GUtility()`.

**10.1.2.33 GSetupDownloadFile()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GSetupDownloadFile (
    GCon g,
```

```

GCStringIn file_path,
GOption options,
GCStringOut info,
GSize info_len )

```

Download a saved controller configuration from a file.

#### Parameters

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the gcb file.
<i>options</i>	Bit mask to determine what configuration data to download. See below for all options.
<i>info</i>	Optional pointer to a buffer to store the controller info. If no info is needed, specify as NULL.
<i>info_len</i>	Length of optional info buffer. If no info is needed, specify as NULL.

#### Returns

The success status or error code of the function. If the options parameter is set to 0, the return value will be a bit mask indicating which sectors in the specified GCB are not empty. Otherwise, see [gclib\\_errors.h](#) for possible error values.

#### Note

By default, [GSetupDownloadFile\(\)](#) will stop immediately if an error is encountered downloading data. This can be overridden in the options parameter. For example, you may want to override the error if you have a backup from an 8-axis controller and want to restore the parameters for the first 4 axes to a 4-axis controller.

If both *info* and *info\_len* are not NULL, the controller information will be provided regardless of the options parameter. The options parameter is a bit mask. If options is set to 0, [GSetupDownloadFile\(\)](#) will return a bit mask indicating which sectors in the specified GCB are not empty. The following contains a list of all currently available options:

Bit	Value	Function	Description
1	0x0002	Restore parameters	<b>KPA, KIA, KDA</b> , etc...
3	0x0008	Restore variables	Variables are listed by the <b>LV</b> command
4	0x0010	Restore arrays	Arrays are listed by the <b>LA</b> command
5	0x0020	Restore program	The program is listed by the <b>LS</b> command
31	0x8000	Ignore errors	Ignore invalid parameter errors and continue restoring data. <a href="#">GSetupDownloadFile()</a> will still stop immediately if a connection issue or other fatal error is encountered

#### Usage example:

```

GCon g;
GOption opt = 0;
GCStringOut info;
GSize info_len = 4096;
GReturn rc = GOpen("192.168.0.50", &g);
if (rc) return rc;
// Call GSetupDownloadFile() with options set to 0 so we can get the non-empty sector bit mask
opt = GSetupDownloadFile(g, "C:\\path\\to\\gcb\\file.gcb", 0, NULL, NULL);
info = (GCStringOut)malloc(sizeof(GCStringOut) * info_len);
// Call GSetupDownloadFile() with the bit mask returned in the previous function call
rc = GSetupDownloadFile(g, "C:\\path\\to\\gcb\\file.gcb", opt, info, info_len);
printf("Info:\n\n%s", info);
GClose(g);
free(info);
return rc;

```

Definition at line 476 of file arrays.c.

#### 10.1.2.34 GSleep()

```

void GCALL GSleep (
    unsigned int timeout_ms )

```

Uses [GUtility\(\)](#) and [G\\_UTIL\\_SLEEP](#) to provide a blocking sleep call which can be useful for timing-based chores.

#### Parameters

<i>timeout_ms</i>	The timeout, in milliseconds, to block before returning.
-------------------	--

See [GWaitForBool\(\)](#) for an example.

Definition at line 24 of file `gclibo.c`.

References [G\\_UTIL\\_SLEEP](#), and [GUtility\(\)](#).

Referenced by [GlpRequests\(\)](#).

#### 10.1.2.35 GTimeout()

```
GReturn GCALL GTimeout (
    GCon g,
    short timeout_ms )
```

Uses [GUtility\(\)](#) and [G\\_UTIL\\_TIMEOUT\\_OVERRIDE](#) to set the library timeout.

#### Parameters

<i>g</i>	Connection's handle.
<i>timeout_ms</i>	The value to be used for the timeout. Use <a href="#">G_USE_INITIAL_TIMEOUT</a> to set the timeout back to the initial <a href="#">GOpen()</a> value, <code>--timeout</code> .

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) and [x\\_gread\\_gwrite.cpp](#) for examples.

Definition at line 65 of file `gclibo.c`.

#### 10.1.2.36 GUtility()

```
GCLIB_DLL_EXPORTED GReturn GCALL GUtility (
    GCon g,
    GOption request,
    GMemory memory1,
    GMemory memory2 )
```

Provides read/write access to driver settings and convenience features based on the request variable.

#### Note

The open source library, [gclibo.h](#), has wrappers for most of these utilities.

#### Parameters

<i>g</i>	Connection's handle.
<i>request</i>	Defines the request. Input/Output and type of memory are implicit in the value of request. The following lists the supported request values.

- [G\\_UTIL\\_TIMEOUT](#) Read initial timeout value, as specified in [GOpen\(\)](#) via `--timeout` switch.
  - `memory1` is output and must be an unsigned `short*`.
  - `memory2` is ignored, use null.
- [G\\_UTIL\\_TIMEOUT\\_OVERRIDE](#) See [GTimeout\(\)](#). Write/Read override timeout value.

- `memory1` is input. If nonnull, value must be a `short*` holding the override, in milliseconds, for the timeout. Write `G_USE_INITIAL_TIMEOUT` to use initial timeout. If null, no write occurs.
  - `memory2` is output. If nonnull, value must be a `short*` which will be filled with the current override. `G_USE_INITIAL_TIMEOUT` indicates initial timeout used. If null, no read occurs. `memory2` is processed before `'memory1'`.
- [G\\_UTIL\\_VERSION](#) See [GVersion\(\)](#). Returns the library version. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is output, and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_INFO](#) See [GInfo\(\)](#). Returns information about the connection.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_SLEEP](#) See [GSleep\(\)](#). Platform-independent, non-busy, sleep. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is input and must be an `unsigned int*`, units are milliseconds.
  - `memory2` is ignored, use null.
- [G\\_UTIL\\_ADDRESSES](#) see [GAddresses\(\)](#). Provides a `\n` delimited listing of all available IP addresses, PCI addresses, and COM ports. A valid connection (`g`) is not necessary, i.e. `g` may be null. The suffix `-d` will be appended to each address to indicate these addresses are available via direct connection. See [G\\_UTIL\\_↔GCAPS\\_ADDRESSES](#) for addresses through [gcaps](#).
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_IPREQUEST](#) see [GIpRequests\(\)](#). Listens and returns a `\n` delimited listing of Galil MAC addresses sending BOOT-P or DHCP requests. The function will listen, and block, for roughly 5 seconds. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_ASSIGN](#) see [GAssign\(\)](#). Provides a method to assign an IP address given a Galil MAC address. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is input and must be a `char*` containing the null terminated address that is to be assigned. e.g. `"192.168.0.43"`.
  - `memory2` is input and must be a `char*` containing the null terminated controller MAC address. e.g. `"00:50:4C:20:01:23"`.
- [G\\_UTIL\\_DEVICE\\_INITIALIZE](#) Provides a method to reinitialize a connection after a reset, e.g. an RS command. Depending on the device type, the appropriate commands will be sent to configure the communication bus for optimal performance.
  - `memory1` is ignored, use null.
  - `memory2` is ignored, use null.
- [G\\_UTIL\\_PING](#) Uses ICMP ping to determine if an IP address is reachable and assigned. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is input and must be a `char*` containing the null terminated address that is to be pinged. e.g. `"192.168.0.43"`.

- `memory2` is output and must be an `int*`. The value will be set to zero if the ping times out, and nonzero if a ping reply is returned.
- [G\\_UTIL\\_ERROR\\_CONTEXT](#) More error detail for the last error on GCon, where available. The internal error message is cleared upon read.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.

The following request values are for use with a [@ref gcaps server](#).

- [G\\_UTIL\\_GCAPS\\_VERSION](#) see [GVersion\(\)](#). Returns the `gcaps` server version. A valid connection (`g`) is not necessary, i.e. `g` may be null. This operation will connect to the server to determine the version.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_GCAPS\\_ADDRESSES](#) see [GAddresses\(\)](#). Provides a `\n` delimited listing of all available IP addresses, PCI addresses, and COM ports as available from the `gcaps` server. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_GCAPS\\_IPREQUEST](#) see [GIpRequests\(\)](#). Connects to `gcaps` and returns a `\n` delimited listing of Galil MAC addresses sending BOOT-P or DHCP requests. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_GCAPS\\_ASSIGN](#) see [GAssign\(\)](#). Provides a method to assign an IP address through `gcaps` given a Galil MAC address. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is input and must be a `char*` containing the null terminated address that is to be assigned. e.g. "192.168.0.43".
  - `memory2` is input and must be a `char*` containing the null terminated controller MAC address. e.g. "00:50:4C:20:01:23".
- [G\\_UTIL\\_GCAPS\\_PING](#) Uses ICMP ping to determine if an IP address is reachable and assigned. Ping sent from the `gcaps` server. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is input and must be a `char*` containing the null terminated address that is to be pinged. e.g. "192.168.0.43".
  - `memory2` is output and must be an `int*`. The value will be set to zero if the ping times out, and nonzero if a ping reply is returned.

#### Parameters

<i>memory1</i>	An untyped pointer to data required for request. The data type is defined by the request variable.
<i>memory2</i>	An untyped pointer to data required for request. The data type is defined by the request variable.



**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See the following functions from gclibo, the open source portion, for implementation of several [GUtility\(\)](#) requests.:

- [GAddresses\(\)](#)
- [GAssign\(\)](#)
- [GInfo\(\)](#)
- [GIpRequests\(\)](#)
- [GSleep\(\)](#)
- [GTimeout\(\)](#)
- [GVersion\(\)](#)

Referenced by [GAddresses\(\)](#), [GAssign\(\)](#), [GIpRequests\(\)](#), [GListServers\(\)](#), [GRemoteConnections\(\)](#), [GServer↔Status\(\)](#), [GSetServer\(\)](#), [GSleep\(\)](#), and [GVersion\(\)](#).

**10.1.2.37 GVersion()**

```
GReturn GCALL GVersion (
    GCStringOut ver,
    GSize ver_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_VERSION](#) and [G\\_UTIL\\_GCAPS\\_VERSION](#) to provide the library and [gcaps](#) version numbers.

**Parameters**

<i>ver</i>	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so.
<i>ver_len</i>	Length of buffer.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

The version number of gclib is provided first. If the [gcaps](#) server can be found, its version will be provided after a space.

Example with [gcaps](#) version.

```
154.190.329 1.0.0.82
```

Example with gclib version only.

```
154.190.329
```

**Note**

[GVersion\(\)](#) will take up to 1 second to look for [gcaps](#).

See [x\\_examples.cpp](#) for an example.

Definition at line 29 of file [gclibo.c](#).

References [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_VERSION](#), [G\\_UTIL\\_VERSION](#), and [GUtility\(\)](#).

**10.1.2.38 GWaitForBool()**

```
GReturn GCALL GWaitForBool (
    GCon g,
```

```
GCStringIn predicate,
int trials )
```

Blocking call that returns when the controller evaluates the predicate as true.

Polls the message command (MG) to check the value of predicate. Polling will continue until the controller responds with a nonzero value or the number of polling trials is reached.

The amount of time until the function fails with `G_GCLIB_POLLING_FAILED` is roughly (trials \* `POLLINGINTERVAL`) milliseconds.

#### Parameters

<i>g</i>	Connection's handle.
<i>predicate</i>	A null-terminated string containing the predicate to be polled. The predicate will be enclosed in parentheses and used in the command <code>MG (predicate)</code> to return the value.
<i>trials</i>	The number of polling cycles to perform looking for a nonzero value. Use -1 to poll indefinitely.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [GMotionComplete\(\)](#) for an example.

Definition at line 318 of file `gclibo.c`.

### 10.1.2.39 GWrite()

```
GCLIB_DLL_EXPORTED GReturn GCALL GWrite (
    GCon g,
    GBufIn buffer,
    GSize buffer_len )
```

Performs a write on the connection.

#### Parameters

<i>g</i>	Connection's handle.
<i>buffer</i>	The user's write buffer. To send a <a href="#">Galil</a> command, a terminating carriage return is usually required.
<i>buffer_len</i>	The length of the data in the buffer.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values. If `G_NO_ERROR` is returned, all bytes were written.

#### Warning

This function is deprecated and will be removed in a future `gclib` version. Please contact [Galil](#) for needs not covered by the other `gclib` functions.

See [x\\_gread\\_gwrite.cpp](#) for an example.

## 10.2 C++ examples

### 10.2.1 Description

Files included in the C++ examples.

#### Files

- file [commands.cpp](#)

- file [commands\\_example.cpp](#)
- file [contour.cpp](#)
- file [contour\\_example.cpp](#)
- file [examples.h](#)
- file [ip\\_assigner.cpp](#)
- file [ip\\_assigner\\_example.cpp](#)
- file [jog.cpp](#)
- file [jog\\_example.cpp](#)
- file [message.cpp](#)
- file [message\\_example.cpp](#)
- file [motion\\_complete.cpp](#)
- file [motion\\_complete\\_example.cpp](#)
- file [position\\_tracking.cpp](#)
- file [position\\_tracking\\_example.cpp](#)
- file [record\\_position.cpp](#)
- file [record\\_position\\_example.cpp](#)
- file [remote\\_client.cpp](#)
- file [remote\\_client\\_example.cpp](#)
- file [remote\\_server.cpp](#)
- file [remote\\_server\\_example.cpp](#)
- file [vector.cpp](#)
- file [vector\\_example.cpp](#)

## Macros

- `#define _CRT_SECURE_NO_WARNINGS`
- `#define GALIL_EXAMPLE_OK 0`
- `#define GALIL_EXAMPLE_ERROR -100`
- `#define G_LASTINDEX 999`

## Typedefs

- typedef `std::vector< string > tokens`

## Functions

- [GReturn commands](#) (`GCon g`)  
*Demonstrates various uses of [GCommand\(\)](#) and [GUtility\(\)](#).*
- `int main` (`int argc, char *argv[]`)  
*Main function for Commands Example.*
- `bool load_buf` (`GCon g, const std::vector< int > &positions_A, const std::vector< int > &positions_B, int capacity, int &cmd`)  
*Loads contour buffer with commands from the given text file.*
- `std::vector< int > csv_to_vector` (`ifstream &is`)  
*Converts a file of comma separated values to a vector.*
- [GReturn contour](#) (`GCon g, char *fileA, char *fileB`)  
*Record user's training and plays back training through contour mode.*
- `void e` (`GReturn rc`)  
*A trivial, C++ style return code check used in [Galil's](#) examples and demos.*
- `void error` (`GCon g, GReturn rc`)  
*An example of error handling and debugging information.*
- `int pause` (`()`)  
*Pauses console apps for a user key stroke.*
- [GReturn position\\_tracking](#) (`GCon g, int speed=5000`)

- Puts controller into Position Tracking Mode and accepts user-entered positions.*

  - [GReturn jog](#) ([GCon g](#))

*Puts controller into Jog Mode and accepts user input to adjust the speed.*
- [GReturn vector](#) ([GCon g](#), char \*file)

*Puts controller into Vector Mode and accepts a file defining vector points.*
- [GReturn ip\\_assigner](#) (char \*serial\_num, int address)

*Assigns controller an IP Adress given a serial number and a 1 byte address.*
- [GReturn motion\\_complete](#) ([GCon g](#))

*Uses interrupts to track when the motion of controller is completed.*
- [GReturn message](#) ([GCon g](#))

*Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.*
- [GReturn record\\_position](#) ([GCon g](#), char \*fileA, char \*fileB)

*Record user's training and saves to a text file.*
- [GReturn remote\\_server](#) (const char \*server\_name)

*Publishes local gcaps server to the network.*
- [GReturn remote\\_client](#) ()

*Lists available remote servers and allows connection to remote server.*
- tokens [string\\_split](#) (const string &str, const string &token)

*Splits a string into a vector based on a token.*
- int [check\\_interrupts](#) ([GCon g](#), [GCStringIn axes](#))

*Monitors interrupt status on the given axes and returns when interrupts are fired.*
- void [write\\_array\\_to\\_file](#) ([GCon g](#), ofstream &os, const char \*array\_name, int previous\_rd, int rd)

*Grabs data from array on controller and writes it to the given text file.*
- void [print\\_client\\_message](#) (const char \*message)
- void [print\\_servers\\_list](#) (const [std::vector](#)< [std::string](#) > &server\_list)
- void [servers\\_to\\_list](#) ([std::vector](#)< [std::string](#) > &server\_list, [std::string](#) servers)
- void [print\\_server\\_message](#) (const char \*message)
- bool [load\\_buffer](#) ([GCon g](#), ifstream &fs, int capacity)

## 10.2.2 Function Documentation

### 10.2.2.1 commands()

[GReturn](#) `commands` (  
[GCon g](#) )

Demonstrates various uses of [GCommand\(\)](#) and [GUtility\(\)](#).

#### Parameters

<i>g</i>	Connection's handle.
----------	----------------------

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [commands\\_example.cpp](#) for an example.  
 Definition at line 16 of file `commands.cpp`.

### 10.2.2.2 contour()

[GReturn](#) `contour` (  
[GCon g](#),

```
char * fileA,
char * fileB )
```

Record user's training and plays back training through contour mode.

#### Parameters

<i>g</i>	Connection's handle.
<i>fileA</i>	A Path to a text file where training for Axis A will be recorded.
<i>fileB</i>	A Path to a text file where training for Axis B will be recorded.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [contour\\_example.cpp](#) for an example.

Definition at line 20 of file `contour.cpp`.

#### 10.2.2.3 e()

```
void e (
    GReturn rc ) [inline]
```

A trivial, C++ style return code check used in [Galil's](#) examples and demos.

Throws GReturn if return value is not G\_NO\_ERROR. See [Commands\\_Example.cpp](#) for example usage and catch() handler.

Definition at line 33 of file `examples.h`.

References G\_NO\_ERROR.

Referenced by `GclibJava::GArrayUpload()`, `ip_assigner()`, and `remote_server()`.

#### 10.2.2.4 ip\_assigner()

```
GReturn ip_assigner (
    char * serial_num,
    int address )
```

Assigns controller an IP Address given a serial number and a 1 byte address.

#### Parameters

<i>serial_num</i>	Serial Number of the controller.
<i>address</i>	A 1 byte address that defines the last byte of the IP Address.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [ip\\_assigner\\_example.cpp](#) for an example.

This function will listen on the network for controllers requesting an IP Address. If a detected controller matches the serial number provided by the user, a new IP Address will be assigned based on the first 3 bytes of the detected IP Address combined with the user defined 1 byte address.

Definition at line 26 of file `ip_assigner.cpp`.

References `e()`, `G_SMALL_BUFFER`, `GlpRequests()`, and `string_split()`.

#### 10.2.2.5 jog()

```
GReturn jog (
    GCon g )
```

Puts controller into Jog Mode and accepts user input to adjust the speed.

## Parameters

<i>g</i>	Connection's handle.
----------	----------------------

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [jog\\_example.cpp](#) for an example.

Key	Usage
q	Quit Jogging
a	-2000 counts / second
s	-500 counts / second
d	+500 counts / second
f	+2000 counts / second
r	Direction Reversal

Definition at line 29 of file jog.cpp.

## 10.2.2.6 load\_buffer()

```
bool load_buffer (
    GCon g,
    ifstream & fs,
    int capacity )
```

Loads vector buffer with commands from the given text file.

Returns false when there are no more lines in the text file

Definition at line 88 of file vector.cpp.

## 10.2.2.7 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file `commands_example.cpp`.

References `G_SMALL_BUFFER`.

### 10.2.2.8 `message()`

```
GReturn message (
    GCon g )
```

Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.

#### Parameters

<i>g</i>	Connection's handle.
----------	----------------------

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [message\\_example.cpp](#) for an example.

Definition at line 14 of file `message.cpp`.

Referenced by `Examples::Message()`.

### 10.2.2.9 `motion_complete()`

```
GReturn motion_complete (
    GCon g )
```

Uses interrupts to track when the motion of controller is completed.

#### Parameters

<i>g</i>	Connection's handle.
----------	----------------------

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [motion\\_complete\\_example.cpp](#) for an example.

Definition at line 18 of file `motion_complete.cpp`.

### 10.2.2.10 `position_tracking()`

```
GReturn position_tracking (
    GCon g,
    int speed = 5000 )
```

Puts controller into Position Tracking Mode and accepts user-entered positions.

#### Parameters

<i>g</i>	Connection's handle.
<i>speed</i>	Optional speed of the controller in Position Tracking Mode. Default value of 5000.



**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [position\\_tracking\\_example.cpp](#) for an example.

Definition at line 15 of file `position_tracking.cpp`.

**10.2.2.11 record\_position()**

```
GReturn record_position (
    GCon g,
    char * fileA,
    char * fileB )
```

Record user's training and saves to a text file.

**Parameters**

<i>g</i>	Connection's handle.
<i>fileA</i>	A Path to a text file where training for Axis A will be recorded.
<i>fileB</i>	A Path to a text file where training for Axis B will be recorded.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [record\\_position\\_example.cpp](#) for an example.

Definition at line 20 of file `record_position.cpp`.

**10.2.2.12 remote\_client()**

```
GReturn remote_client ( )
```

Lists available remote servers and allows connection to remote server.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [remote\\_client\\_example](#) for an example.

Key	Usage
q	Quit
s	List available servers on then network
h	List available hardware on the current server
0-9	Connect to server instance by number
l	Connect back to local server

Definition at line 89 of file `remote_client.cpp`.

References `G_SMALL_BUFFER`.

**10.2.2.13 remote\_server()**

```
GReturn remote_server (
    const char * server_name )
```

Publishes local gcaps server to the network.

**Parameters**

<i>Name</i>	to publish server under.
-------------	--------------------------

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See `remote_server_example` for an example.

Key	Usage
q	Quit
p	Publish this server to the network
r	Remove this server from the network

Definition at line 39 of file `remote_server.cpp`.

References `e()`, `G_SMALL_BUFFER`, and `GPublishServer()`.

**10.2.2.14 vector()**

```
GReturn vector (
    GCon g,
    char * file )
```

Puts controller into Vector Mode and accepts a file defining vector points.

**Parameters**

<i>g</i>	Connection's handle.
<i>file</i>	A Path to a file that defines vector commands.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See `vector_example.cpp` for an example.

**Example text file:**

```
VP -2219,-2667
VP -2523,-2832
VP 2844,-1425
VP 728,1971
VP 2127,183
VP -997,688
VP 725,-1893
VP 527,2899
VP -37,2523
VP 1277,1425
VP 857,2388
VP 1096,-1694
CR 1000,0,90
```

Definition at line 36 of file `vector.cpp`.

**10.3 C#/VB examples****10.3.1 Description**

Files included in the C# [Examples](#).

**Files**

- file [commands.cs](#)

- file [commands\\_example.cs](#)
- file [contour.cs](#)
- file [contour\\_example.cs](#)
- file [examples.cs](#)
- file [ip\\_assigner.cs](#)
- file [ip\\_assigner\\_example.cs](#)
- file [jog.cs](#)
- file [jog\\_example.cs](#)
- file [message.cs](#)
- file [message\\_example.cs](#)
- file [motion\\_complete.cs](#)
- file [motion\\_complete\\_example.cs](#)
- file [position\\_tracking.cs](#)
- file [position\\_tracking\\_example.cs](#)
- file [record\\_position.cs](#)
- file [record\\_position\\_example.cs](#)
- file [Remote\\_Client.cs](#)
- file [remote\\_client\\_example.cs](#)
- file [Remote\\_Server.cs](#)
- file [remote\\_server\\_example.cs](#)
- file [vector\\_mode.cs](#)
- file [vector\\_mode\\_example.cs](#)

## Data Structures

- class [Commands\\_Example](#)  
*Demonstrates various uses of `GCommand()` and basic controller queries.*
- class [Contour\\_Example](#)  
*Record user's training and plays back training through contour mode.*
- class [Examples](#)  
*Provides a class of shared constants and methods for gclib's example projects.*
- class [IP\\_Assigner\\_Example](#)  
*Assigns controller an IP Adress given a serial number and a 1 byte address.*
- class [Jog\\_Example](#)  
*Accepts user-input at the command line to control the speed of the controller in Jog mode.*
- class [Message\\_Example](#)  
*Demonstrates how to handle and interpret messages from the controller.*
- class [Motion\\_Complete\\_Example](#)  
*Uses controller interrupts to detect when motion is complete.*
- class [Position\\_Tracking\\_Example](#)  
*Places controller into position tracking mode. Accepts user-defined positional values at the command line.*
- class [Record\\_Position\\_Example](#)  
*Takes two file paths at the command line to hold positional data for Axis A and Axis B. Positional data is saved to the two files until an analog input value changes.*
- class [Remote\\_Client\\_Example](#)  
*Demonstrates various uses of `GListServers()` and `GSetServer()`*
- class [Remote\\_Server\\_Example](#)  
*Demonstrates various uses of `GPublishServer()`*
- class [Vector\\_Mode\\_Example](#)  
*Takes a path to a file at the command line holding vector commands for the controller. The controller is placed into vector mode and commands are read from the file and sent to the controller.*

## Functions

- static int [Commands](#) ([gclib gclib](#))  
*Demonstrates various uses of [GCommand\(\)](#) and basic controller queries.*
- static int [Contour](#) ([gclib gclib](#), string fileA, string fileB)  
*Record user's training and plays back training through contour mode.*
- static int [IP\\_Assigner](#) ([gclib gclib](#), string serial\_num, byte address)  
*Assigns controller an IP Adress given a serial number and a 1 byte address.*
- static int [Jog](#) ([gclib gclib](#))  
*Puts controller into Jog Mode and accepts user input to adjust the speed.*
- static int [Message](#) ([gclib gclib](#))  
*Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.*
- static int [Motion\\_Complete](#) ([gclib gclib](#))  
*Uses interrupts to track when the motion of controller is completed.*
- static int [Position\\_Tracking](#) ([gclib gclib](#), int speed)  
*Puts controller into Position Tracking Mode and accepts user-entered positions.*
- static int [Record\\_Position](#) ([gclib gclib](#), string fileA, string fileB)  
*Record user's training and saves to a text file.*
- static int [Remote\\_Client](#) ()  
*Accepts user input to publish to list and connect to available servers.*
- static int [Remote\\_Server](#) (string server\_name)  
*Accepts user input to publish or remove local gcaps server from the network.*
- static int [Vector\\_Mode](#) ([gclib gclib](#), string file)  
*Puts controller into Vector Mode and accepts a file defining vector points.*

### 10.3.2 Function Documentation

#### 10.3.2.1 Commands()

```
static int Commands (
    gclib gclib ) [inline], [static]
```

Demonstrates various uses of [GCommand\(\)](#) and basic controller queries.

##### Parameters

<a href="#">gclib</a>	A gclib object with a valid connection.
-----------------------	---

##### Returns

The success status or error code of the function.

See [commands\\_example.cs](#) for an example.

For VB.NET, see definition in file [commands.vb](#)

Definition at line 28 of file [commands.cs](#).

References Examples.GALIL\_EXAMPLE\_OK, [gclib.GCmdD\(\)](#), [gclib.GCmdI\(\)](#), and [gclib.GCommand\(\)](#).

Referenced by [Commands\\_Example.Main\(\)](#).

#### 10.3.2.2 Contour()

```
static int Contour (
    gclib gclib,
    string fileA,
    string fileB ) [inline], [static]
```

Record user's training and plays back training through contour mode.

#### Parameters

<i>gclib</i>	A gclib object with a valid connection.
<i>fileA</i>	A Path to a text file where training for Axis A will be recorded.
<i>fileB</i>	A Path to a text file where training for Axis B will be recorded.

#### Returns

The success status or error code of the function.

See [contour\\_example.cs](#) for an example.

For VB.NET, see definition in file [contour.vb](#)

Definition at line 32 of file contour.cs.

References `Examples.GALIL_EXAMPLE_ERROR`, `Examples.GALIL_EXAMPLE_OK`, `gclib.GCmdl()`, `gclib.GCommand()`, `gclib.GMotionComplete()`, and `Examples.Record_Position()`.

Referenced by `Contour_Example.Main()`.

#### 10.3.2.3 IP\_Assigner()

```
static int IP_Assigner (
    gclib gclib,
    string serial_num,
    byte address ) [inline], [static]
```

Assigns controller an IP Address given a serial number and a 1 byte address.

#### Parameters

<i>gclib</i>	A gclib object.
<i>serial_num</i>	The serial number of a <a href="#">Galil</a> controller.
<i>address</i>	A 1 byte value to be added to the new IP Address.

#### Returns

The success status or error code of the function.

This function will listen on the network for controllers requesting an IP Address.

If a detected controller matches the serial number provided by the user, a new IP Address will be assigned based on the first 3 bytes of the detected IP Address combined with the user defined 1 byte address.

See [ip\\_assigner\\_example.cs](#) for an example.

For VB.NET, see definition in file [ip\\_assigner.vb](#)

Definition at line 36 of file ip\_assigner.cs.

References `Examples.GALIL_EXAMPLE_ERROR`, `Examples.GALIL_EXAMPLE_OK`, `gclib.GAssign()`, `gclib.GCommand()`, `gclib.GInfo()`, `gclib.GIpRequests()`, and `gclib.GOpen()`.

Referenced by `IP_Assigner_Example.Main()`.

#### 10.3.2.4 Jog()

```
static int Jog (
    gclib gclib ) [inline], [static]
```

Puts controller into Jog Mode and accepts user input to adjust the speed.

**Parameters**

<i>gclib</i>	A gclib object with a valid connection.
--------------	---

**Returns**

The success status or error code of the function.

Key	Usage
q	Quit Jogging
a	-2000 counts / second
s	-500 counts / second
d	+500 counts / second
f	+2000 counts / second
r	Direction Reversal

See [jog\\_example.cs](#) for an example.

For VB.NET, see definition in file [jog.vb](#)

Definition at line 35 of file jog.cs.

References Examples.GALIL\_EXAMPLE\_OK, gclib.GCommand(), and gclib.GMotionComplete().

Referenced by Jog\_Example.Main().

**10.3.2.5 Message()**

```
static int Message (
    gclib gclib ) [inline], [static]
```

Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.

**Parameters**

<i>gclib</i>	A gclib object with a valid connection.
--------------	---

**Returns**

The success status or error code of the function.

See [message\\_example.cs](#) for an example.

For VB.NET, see definition in file [message.vb](#)

Definition at line 27 of file message.cs.

References Examples.GALIL\_EXAMPLE\_OK, gclib.GCommand(), gclib.GMessage(), gclib.GProgramDownload(), and message().

Referenced by Message\_Example.Main().

**10.3.2.6 Motion\_Complete()**

```
static int Motion_Complete (
    gclib gclib ) [inline], [static]
```

Uses interrupts to track when the motion of controller is completed.

**Parameters**

<i>gclib</i>	A gclib object with a valid connection.
--------------	---

**Returns**

The success status or error code of the function.

See [motion\\_complete\\_example.cs](#) for an example.

For VB.NET, see definition in file [motion\\_complete.vb](#)

Definition at line 26 of file [motion\\_complete.cs](#).

References [Examples.GALIL\\_EXAMPLE\\_ERROR](#), [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GCommand\(\)](#), [gclib.GInterrupt\(\)](#), and [gclib.GTimeout\(\)](#).

Referenced by [Motion\\_Complete\\_Example.Main\(\)](#).

**10.3.2.7 Position\_Tracking()**

```
static int Position_Tracking (
    gclib gclib,
    int speed ) [inline], [static]
```

Puts controller into Position Tracking Mode and accepts user-entered positions.

**Parameters**

<i>gclib</i>	A gclib object with a valid connection.
<i>speed</i>	Optional speed of the controller in Position Tracking Mode. Default value of 5000

**Returns**

The success status or error code of the function.

See [position\\_tracking\\_example.cs](#) for an example.

For VB.NET, see definition in file [position\\_tracking.vb](#)

Definition at line 28 of file [position\\_tracking.cs](#).

References [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GCommand\(\)](#), and [gclib.GMotionComplete\(\)](#).

Referenced by [Position\\_Tracking\\_Example.Main\(\)](#).

**10.3.2.8 Record\_Position()**

```
static int Record_Position (
    gclib gclib,
    string fileA,
    string fileB ) [inline], [static]
```

Record user's training and saves to a text file.

**Parameters**

<i>gclib</i>	A gclib object with a valid connection.
<i>fileA</i>	A Path to a text file where training for Axis A will be recorded.
<i>fileB</i>	A Path to a text file where training for Axis B will be recorded.

**Returns**

The success status or error code of the function.

See [record\\_position\\_example.cs](#) for an example.

For VB.NET, see definition in file [record\\_position.vb](#)

Definition at line 32 of file [record\\_position.cs](#).

References [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GCmndI\(\)](#), [gclib.GCommand\(\)](#), and [gclib.GProgramDownload\(\)](#).

Referenced by `Examples.Contour()`, and `Record_Position_Example.Main()`.

### 10.3.2.9 Remote\_Client()

```
static int Remote_Client ( ) [inline], [static]
```

Accepts user input to publish to list and connect to available servers.

#### Returns

The success status or error code of the function.

Key	Usage
q	Quit
s	List available servers on then network
h	List available hardware on the current server
0-9	Connect to server instance by number
l	Connect back to local server

See [remote\\_client\\_example.cs](#) for an example.

For VB.NET, see definition in file [remote\\_client.vb](#)

Definition at line 33 of file `Remote_Client.cs`.

References `gclib.GAddresses()`, `Examples.GALIL_EXAMPLE_OK`, `gclib.GListServers()`, and `gclib.GSetServer()`.

Referenced by `Remote_Client_Example.Main()`.

### 10.3.2.10 Remote\_Server()

```
static int Remote_Server (
    string server_name ) [inline], [static]
```

Accepts user input to publish or remove local gcaps server from the network.

#### Parameters

<code>server_name</code>	The name to publish local gcaps server under.
--------------------------	---

#### Returns

The success status or error code of the function.

Key	Usage
q	Quit
p	Publish this server to the network
r	Remove this server from the network

See [remote\\_server\\_example.cs](#) for an example.

For VB.NET, see definition in file [remote\\_server.vb](#)

Definition at line 32 of file `Remote_Server.cs`.

References `Examples.GALIL_EXAMPLE_OK`, and `gclib.GPublishServer()`.

Referenced by `Remote_Server_Example.Main()`.

### 10.3.2.11 Vector\_Mode()

```
static int Vector_Mode (
    gclib gclib,
```



```
string file ) [inline], [static]
```

Puts controller into Vector Mode and accepts a file defining vector points.

#### Parameters

<i>gclib</i>	A gclib object with a valid connection.
<i>file</i>	A path to a file with stored vector commands.

#### Returns

The success status or error code of the function.

#### Example text file:

```
VP -2219,-2667
VP -2523,-2832
VP 2844,-1425
VP 728,1971
VP 2127,183
VP -997,688
VP 725,-1893
VP 527,2899
VP -37,2523
VP 1277,1425
VP 857,2388
VP 1096,-1694
CR 1000,0,90
```

See [vector\\_mode\\_example.cs](#) for an example.

For VB.NET, see definition in file [vector\\_mode.vb](#)

Definition at line 45 of file [vector\\_mode.cs](#).

References [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GCmdI\(\)](#), [gclib.GCommand\(\)](#), and [gclib.GMotionComplete\(\)](#).

Referenced by [Vector\\_Mode\\_Example.Main\(\)](#).

## 10.4 C#/VB API

### 10.4.1 Description

Files included in the C#/VB API.

#### Data Structures

- class [gclib](#)

*Provides a class that binds to gclib's unmanaged dll. Wraps each call and provides a more user-friendly interface for use in C#.*



# Chapter 11

## Namespace Documentation

### 11.1 examples Namespace Reference

#### Data Structures

- class [Examples](#)  
*Provides a class of shared constants and methods for gclib's example projects.*
- class [Commands\\_Example](#)  
*Demonstrates various uses of `GCommand()` and basic controller queries.*
- class [Contour\\_Example](#)  
*Record user's training and plays back training through contour mode.*
- class [IP\\_Assigner\\_Example](#)  
*Assigns controller an IP Address given a serial number and a 1 byte address.*
- class [Jog\\_Example](#)  
*Accepts user-input at the command line to control the speed of the controller in Jog mode.*
- class [Message\\_Example](#)  
*Demonstrates how to handle and interpret messages from the controller.*
- class [Motion\\_Complete\\_Example](#)  
*Uses controller interrupts to detect when motion is complete.*
- class [Position\\_Tracking\\_Example](#)  
*Places controller into position tracking mode. Accepts user-defined positional values at the command line.*
- class [Record\\_Position\\_Example](#)  
*Takes two file paths at the command line to hold positional data for Axis A and Axis B. Positional data is saved to the two files until an analog input value changes.*
- class [Remote\\_Client\\_Example](#)  
*Demonstrates various uses of `GListServers()` and `GSetServer()`*
- class [Remote\\_Server\\_Example](#)  
*Demonstrates various uses of `GPublishServer()`*
- class [Vector\\_Mode\\_Example](#)  
*Takes a path to a file at the command line holding vector commands for the controller. The controller is placed into vector mode and commands are read from the file and sent to the controller.*

#### Functions

- int [Commands](#) (Gclib [gclib](#))  
*Demonstrates various uses of `GCommand()` And basic controller queries.*
- int [Main](#) ()
- int [Contour](#) (Gclib [gclib](#), string fileA, string fileB)
- void [PrintError](#) (Gclib [gclib](#), Exception ex)  
*Prints the exception to the console And queries the controller for the most recent error message.*

- int **IP\_Assigner** (Gclib [gclib](#), string serial\_num, byte address)
- int **Jog** (Gclib [gclib](#))
- **Message** (Gclib [gclib](#))
- **Motion\_Complete** (Gclib [gclib](#))
- **Position\_Tracking** (Gclib [gclib](#), int speed)
- **Record\_Position** (Gclib [gclib](#), string fileA, string fileB)
- int **Remote\_Client** ()
  - Demonstrates various uses of [GListServers\(\)](#) and [GSetServer\(\)](#)*
- int **Remote\_Server** (string server\_name)
  - Demonstrates various uses of [GPublishServer\(\)](#)*
- int **Vector\_Mode** (Gclib [gclib](#), string file)

## Variables

- partial Module [Examples](#)
- positions\_A As **List**< **string** >
- const int **GALIL\_EXAMPLE\_OK** = 0
- const **GALIL\_EXAMPLE\_ERROR** = -100

## 11.1.1 Function Documentation

### 11.1.1.1 Commands()

```
int examples.Commands (
    Gclib gclib )
```

Demonstrates various uses of [GCommand\(\)](#) And basic controller queries.

#### Parameters

<i>gclib</i>	A gclib object with a valid connection.
--------------	---

#### Returns

The success status Or error code of the function.

See [commands\\_example.cs](#) for an example.

### 11.1.1.2 Main()

```
int Main ( )
```

Demonstrates various uses of [GCommand\(\)](#) And basic controller queries.

The first argument should be the IP Address of a [Galil](#) controller.

Main function for the commands example.

#### Returns

The success status Or error code of the function.

The first argument should be the IP Address of a [Galil](#) controller.

Demonstrates various uses of [GListServers\(\)](#) and [GSetServer\(\)](#).

This example requires no command line arguments.

Main function for the Remote Client example.

**Returns**

The success status Or error code of the function.

This example requires no command line arguments.

Demonstrates various uses of [GPublishServer\(\)](#).

The first argument is optional and defines the name to publish the server under.

Main function for the Remote Server example.

**Returns**

The success status Or error code of the function.

The first argument is optional and defines the name to publish the server under.

**11.1.1.3 PrintError()**

```
void examples.PrintError (
    Gclib gcLib,
    Exception ex )
```

Prints the exception to the console And queries the controller for the most recent error message.

**Parameters**

<i>gcLib</i>	The gcLib object from where the exception originated.
<i>ex</i>	The exception object caught by the example.

See [commands\\_example.cs](#) for an example.

**11.1.1.4 Remote\_Client()**

```
int examples.Remote_Client ( )
```

Demonstrates various uses of [GListServers\(\)](#) and [GSetServer\(\)](#)

**Returns**

The success status Or error code of the function.

See [remote\\_client\\_example.cs](#) for an example.

**11.1.1.5 Remote\_Server()**

```
int examples.Remote_Server (
    string server_name )
```

Demonstrates various uses of [GPublishServer\(\)](#)

**Parameters**

<i>server_name</i>	The name to publish the server under.
--------------------	---------------------------------------

**Returns**

The success status Or error code of the function.

See [remote\\_server\\_example.cs](#) for an example.

**11.1.2 Variable Documentation**

### 11.1.2.1 Examples

partial Module [Examples](#)

Provides a class of shared constants And methods for gclib's example projects.

Definition at line 4 of file Commands.vb.

## 11.2 gclib\_example Namespace Reference

### Data Structures

- class [MainForm](#)  
*Demonstrates using gclib in a Windows Form, including using a second thread to free the GUI.*
- class [Program](#)

## 11.3 Package gclibjava

### Data Structures

- class [GclibJava](#)
- class [GclibJavaException](#)

### 11.3.1 Detailed Description

This is the initial version of the the gclib Java wrapper. All functions are subject to change in future releases of gclib.

Java hackers with recommendations on how to make this library better are encouraged to email [softwaresupport@galil.com](mailto:softwaresupport@galil.com). Somebody has to teach those [Galil](#) Java noobs what's what.

Some identified "To Do" tasks:

1. Synchronize access to Gclib and Gclibo interfaces.
2. Choose a data structure to return GAddresses and GIpRequests.
- 3.

## 11.4 vb Namespace Reference

### Data Structures

- class [Gclib](#)  
*Provides a class that binds to gclib's unmanaged dll. Wraps each call and provides a more user-friendly interface for use in Visual Basic.*

### Typedefs

- using **GReturn** = System.Int32
- using **GCon** = System.IntPtr
- using **GSize** = System.UInt32
- using **GCStringOut** = System.Text.StringBuilder
- using **GCStringIn** = System.String
- using **GBufOut** = System.Text.StringBuilder
- using **GBufIn** = System.String
- using **GStatus** = System.Byte

### Variables

- Module **LibraryPath**
- const string **GclibDllPath\_** = "C:\Program Files[\Galil\gclib\dll\x86\gclib.dll"
- const string **GcliboDllPath\_** = "C:\Program Files[\Galil\gclib\dll\x86\gclibo.dll"

# Chapter 12

## Data Structure Documentation

### 12.1 Commands\_Example Class Reference

Demonstrates various uses of [GCommand\(\)](#) and basic controller queries.

#### Static Public Member Functions

- static int [Main](#) (string[] args)  
*Main function for the commands example.*

#### 12.1.1 Detailed Description

Demonstrates various uses of [GCommand\(\)](#) and basic controller queries. The first argument should be the IP Address of a [Galil](#) controller. For VB.NET, see definition in file [commands\\_example.vb](#) Definition at line 23 of file [commands\\_example.cs](#).

#### 12.1.2 Member Function Documentation

##### 12.1.2.1 Main()

```
static int Main (
    string[] args ) [inline], [static]
```

Main function for the commands example.

#### Parameters

<i>args</i>	An array of command line arguments.
-------------	-------------------------------------

#### Returns

The success status or error code of the function.

The first argument should be the IP Address of a [Galil](#) controller. Definition at line 31 of file [commands\\_example.cs](#).

References [Examples.Commands\(\)](#), [Examples.GALIL\\_EXAMPLE\\_ERROR](#), [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GClose\(\)](#), [gclib.GOpen\(\)](#), and [Examples.PrintError\(\)](#).

The documentation for this class was generated from the following file:

- [commands\\_example.cs](#)

## 12.2 Contour\_Example Class Reference

Record user's training and plays back training through contour mode.

### Static Public Member Functions

- static int [Main](#) (string[] args)  
*Main function for the contour example.*

#### 12.2.1 Detailed Description

Record user's training and plays back training through contour mode.

The first argument should be the IP Address of a [Galil](#) controller.

The second argument should be a path to a csv file holding positional data for the A axis.

The third argument should be a path to a csv file holding positional data for the B axis.

For VB.NET, see definition in file [contour\\_example.vb](#)

Definition at line 27 of file contour\_example.cs.

#### 12.2.2 Member Function Documentation

##### 12.2.2.1 Main()

```
static int Main (
    string[] args ) [inline], [static]
```

Main function for the contour example.

##### Parameters

<i>args</i>	An array of command line arguments.
-------------	-------------------------------------

##### Returns

The success status or error code of the function.

The first argument should be the IP Address of a [Galil](#) controller.

The second argument should be a path to a text file where training for Axis A will be recorded.

The third argument should be a path to a text file where training for Axis B will be recorded..

Definition at line 39 of file contour\_example.cs.

References [Examples.Contour\(\)](#), [Examples.GALIL\\_EXAMPLE\\_ERROR](#), [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GCclose\(\)](#), [gclib.GOpen\(\)](#), and [Examples.PrintError\(\)](#).

The documentation for this class was generated from the following file:

- [contour\\_example.cs](#)

## 12.3 Examples Class Reference

Provides a class of shared constants and methods for gclib's example projects.

### Static Public Member Functions

- static int [Commands](#) (gclib gclib)  
*Demonstrates various uses of GCommand() and basic controller queries.*
- static int [Contour](#) (gclib gclib, string fileA, string fileB)  
*Record user's training and plays back training through contour mode.*
- static void [PrintError](#) (gclib gclib, Exception ex)



*Prints the exception to the console and queries the controller for the most recent error message.*

- static int [IP\\_Assigner](#) ([gclib gclib](#), string serial\_num, byte address)  
*Assigns controller an IP Adress given a serial number and a 1 byte address.*
- static int [Jog](#) ([gclib gclib](#))  
*Puts controller into Jog Mode and accepts user input to adjust the speed.*
- static int [Message](#) ([gclib gclib](#))  
*Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.*
- static int [Motion\\_Complete](#) ([gclib gclib](#))  
*Uses interrupts to track when the motion of controller is completed.*
- static int [Position\\_Tracking](#) ([gclib gclib](#), int speed)  
*Puts controller into Position Tracking Mode and accepts user-entered positions.*
- static int [Record\\_Position](#) ([gclib gclib](#), string fileA, string fileB)  
*Record user's training and saves to a text file.*
- static int [Remote\\_Client](#) ()  
*Accepts user input to publish to list and connect to available servers.*
- static int [Remote\\_Server](#) (string server\_name)  
*Accepts user input to publish or remove local gcaps server from the network.*
- static int [Vector\\_Mode](#) ([gclib gclib](#), string file)  
*Puts controller into Vector Mode and accepts a file defining vector points.*

### Static Public Attributes

- const int [GALIL\\_EXAMPLE\\_OK](#) = 0  
*Examples success code.*
- const int [GALIL\\_EXAMPLE\\_ERROR](#) = -100  
*Examples error code.*

### 12.3.1 Detailed Description

Provides a class of shared constants and methods for gclib's example projects.  
For VB.NET, see definition in file [examples.vb](#)  
Definition at line 15 of file [commands.cs](#).

### 12.3.2 Member Function Documentation

#### 12.3.2.1 Commands()

```
static int Commands (
    gclib gclib ) [inline], [static]
```

Demonstrates various uses of [GCommand\(\)](#) and basic controller queries.

#### Parameters

<i>gclib</i>	A gclib object with a valid connection.
--------------	---

#### Returns

The success status or error code of the function.

See [commands\\_example.cs](#) for an example.  
For VB.NET, see definition in file [commands.vb](#)  
Definition at line 28 of file [commands.cs](#).

References Examples.GALIL\_EXAMPLE\_OK, gclib.GCmdD(), gclib.GCmdI(), and gclib.GCommand().  
 Referenced by Commands\_Example.Main().

### 12.3.2.2 Contour()

```
static int Contour (
    gclib gclib,
    string fileA,
    string fileB ) [inline], [static]
```

Record user's training and plays back training through contour mode.

#### Parameters

<i>gclib</i>	A gclib object with a valid connection.
<i>fileA</i>	A Path to a text file where training for Axis A will be recorded.
<i>fileB</i>	A Path to a text file where training for Axis B will be recorded.

#### Returns

The success status or error code of the function.

See [contour\\_example.cs](#) for an example.

For VB.NET, see definition in file [contour.vb](#)

Definition at line 32 of file contour.cs.

References Examples.GALIL\_EXAMPLE\_ERROR, Examples.GALIL\_EXAMPLE\_OK, gclib.GCmdI(), gclib.GCommand(), gclib.GMotionComplete(), and Examples.Record\_Position().

Referenced by Contour\_Example.Main().

### 12.3.2.3 IP\_Assigner()

```
static int IP_Assigner (
    gclib gclib,
    string serial_num,
    byte address ) [inline], [static]
```

Assigns controller an IP Address given a serial number and a 1 byte address.

#### Parameters

<i>gclib</i>	A gclib object.
<i>serial_num</i>	The serial number of a <a href="#">Galil</a> controller.
<i>address</i>	A 1 byte value to be added to the new IP Address.

#### Returns

The success status or error code of the function.

This function will listen on the network for controllers requesting an IP Address.

If a detected controller matches the serial number provided by the user, a new IP Address will be assigned based on the first 3 bytes of the detected IP Address combined with the user defined 1 byte address.

See [ip\\_assigner\\_example.cs](#) for an example.

For VB.NET, see definition in file [ip\\_assigner.vb](#)

Definition at line 36 of file ip\_assigner.cs.

References Examples.GALIL\_EXAMPLE\_ERROR, Examples.GALIL\_EXAMPLE\_OK, gclib.GAssign(), gclib.GCommand(), gclib.GInfo(), gclib.GIpRequests(), and gclib.GOpen().

Referenced by IP\_Assigner\_Example.Main().

**12.3.2.4 Jog()**

```
static int Jog (
    gclib gclib ) [inline], [static]
```

Puts controller into Jog Mode and accepts user input to adjust the speed.

**Parameters**

<i>gclib</i>	A gclib object with a valid connection.
--------------	---

**Returns**

The success status or error code of the function.

Key	Usage
q	Quit Jogging
a	-2000 counts / second
s	-500 counts / second
d	+500 counts / second
f	+2000 counts / second
r	Direction Reversal

See [jog\\_example.cs](#) for an example.

For VB.NET, see definition in file [jog.vb](#)

Definition at line 35 of file jog.cs.

References Examples.GALIL\_EXAMPLE\_OK, [gclib.GCommand\(\)](#), and [gclib.GMotionComplete\(\)](#).

Referenced by [Jog\\_Example.Main\(\)](#).

**12.3.2.5 Message()**

```
static int Message (
    gclib gclib ) [inline], [static]
```

Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.

**Parameters**

<i>gclib</i>	A gclib object with a valid connection.
--------------	---

**Returns**

The success status or error code of the function.

See [message\\_example.cs](#) for an example.

For VB.NET, see definition in file [message.vb](#)

Definition at line 27 of file message.cs.

References Examples.GALIL\_EXAMPLE\_OK, [gclib.GCommand\(\)](#), [gclib.GMessage\(\)](#), [gclib.GProgramDownload\(\)](#), and [message\(\)](#).

Referenced by [Message\\_Example.Main\(\)](#).

**12.3.2.6 Motion\_Complete()**

```
static int Motion_Complete (
    gclib gclib ) [inline], [static]
```

Uses interrupts to track when the motion of controller is completed.

**Parameters**

<i>gclib</i>	A gclib object with a valid connection.
--------------	---

**Returns**

The success status or error code of the function.

See [motion\\_complete\\_example.cs](#) for an example.

For VB.NET, see definition in file [motion\\_complete.vb](#)

Definition at line 26 of file [motion\\_complete.cs](#).

References [Examples.GALIL\\_EXAMPLE\\_ERROR](#), [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GCommand\(\)](#), [gclib.GInterrupt\(\)](#), and [gclib.GTimeout\(\)](#).

Referenced by [Motion\\_Complete\\_Example.Main\(\)](#).

**12.3.2.7 Position\_Tracking()**

```
static int Position_Tracking (
    gclib gclib,
    int speed ) [inline], [static]
```

Puts controller into Position Tracking Mode and accepts user-entered positions.

**Parameters**

<i>gclib</i>	A gclib object with a valid connection.
<i>speed</i>	Optional speed of the controller in Position Tracking Mode. Default value of 5000

**Returns**

The success status or error code of the function.

See [position\\_tracking\\_example.cs](#) for an example.

For VB.NET, see definition in file [position\\_tracking.vb](#)

Definition at line 28 of file [position\\_tracking.cs](#).

References [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GCommand\(\)](#), and [gclib.GMotionComplete\(\)](#).

Referenced by [Position\\_Tracking\\_Example.Main\(\)](#).

**12.3.2.8 PrintError()**

```
static void PrintError (
    gclib gclib,
    Exception ex ) [inline], [static]
```

Prints the exception to the console and queries the controller for the most recent error message.

**Parameters**

<i>gclib</i>	The gclib object from where the exception originated.
<i>ex</i>	The exception object caught by the example.

See [commands\\_example.cs](#) for an example.

Definition at line 39 of file [examples.cs](#).

References [gclib.GCommand\(\)](#).

Referenced by [Commands\\_Example.Main\(\)](#), [Contour\\_Example.Main\(\)](#), [IP\\_Assigner\\_Example.Main\(\)](#), [Jog\\_](#)

Example.Main(), Message\_Example.Main(), Motion\_Complete\_Example.Main(), Position\_Tracking\_Example.Main(), Record\_Position\_Example.Main(), and Vector\_Mode\_Example.Main().

### 12.3.2.9 Record\_Position()

```
static int Record_Position (
    gclib gclib,
    string fileA,
    string fileB ) [inline], [static]
```

Record user's training and saves to a text file.

#### Parameters

<i>gclib</i>	A gclib object with a valid connection.
<i>fileA</i>	A Path to a text file where training for Axis A will be recorded.
<i>fileB</i>	A Path to a text file where training for Axis B will be recorded.

#### Returns

The success status or error code of the function.

See [record\\_position\\_example.cs](#) for an example.

For VB.NET, see definition in file [record\\_position.vb](#)

Definition at line 32 of file [record\\_position.cs](#).

References [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GCmdI\(\)](#), [gclib.GCommand\(\)](#), and [gclib.GProgramDownload\(\)](#).

Referenced by [Examples.Contour\(\)](#), and [Record\\_Position\\_Example.Main\(\)](#).

### 12.3.2.10 Remote\_Client()

```
static int Remote_Client ( ) [inline], [static]
```

Accepts user input to publish to list and connect to available servers.

#### Returns

The success status or error code of the function.

Key	Usage
q	Quit
s	List available servers on then network
h	List available hardware on the current server
0-9	Connect to server instance by number
l	Connect back to local server

See [remote\\_client\\_example.cs](#) for an example.

For VB.NET, see definition in file [remote\\_client.vb](#)

Definition at line 33 of file [Remote\\_Client.cs](#).

References [gclib.GAddresses\(\)](#), [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GListServers\(\)](#), and [gclib.GSetServer\(\)](#).

Referenced by [Remote\\_Client\\_Example.Main\(\)](#).

### 12.3.2.11 Remote\_Server()

```
static int Remote_Server (
    string server_name ) [inline], [static]
```

Accepts user input to publish or remove local gcaps server from the network.

**Parameters**

<i>server_name</i>	The name to publish local gcaps server under.
--------------------	---

**Returns**

The success status or error code of the function.

Key	Usage
q	Quit
p	Publish this server to the network
r	Remove this server from the network

See [remote\\_server\\_example.cs](#) for an example.

For VB.NET, see definition in file [remote\\_server.vb](#)

Definition at line 32 of file Remote\_Server.cs.

References Examples.GALIL\_EXAMPLE\_OK, and gclib.GPublishServer().

Referenced by Remote\_Server\_Example.Main().

**12.3.2.12 Vector\_Mode()**

```
static int Vector_Mode (
    gclib gclib,
    string file ) [inline], [static]
```

Puts controller into Vector Mode and accepts a file defining vector points.

**Parameters**

<i>gclib</i>	A gclib object with a valid connection.
<i>file</i>	A path to a file with stored vector commands.

**Returns**

The success status or error code of the function.

**Example text file:**

```
VP -2219,-2667
VP -2523,-2832
VP 2844,-1425
VP 728,1971
VP 2127,183
VP -997,688
VP 725,-1893
VP 527,2899
VP -37,2523
VP 1277,1425
VP 857,2388
VP 1096,-1694
CR 1000,0,90
```

See [vector\\_mode\\_example.cs](#) for an example.

For VB.NET, see definition in file [vector\\_mode.vb](#)

Definition at line 45 of file vector\_mode.cs.

References Examples.GALIL\_EXAMPLE\_OK, gclib.GCmdI(), gclib.GCommand(), and gclib.GMotionComplete().

Referenced by Vector\_Mode\_Example.Main().

The documentation for this class was generated from the following files:

- [commands.cs](#)
- [examples.cs](#)
- [contour.cs](#)
- [ip\\_assigner.cs](#)

- [jog.cs](#)
- [message.cs](#)
- [motion\\_complete.cs](#)
- [position\\_tracking.cs](#)
- [record\\_position.cs](#)
- [Remote\\_Client.cs](#)
- [Remote\\_Server.cs](#)
- [vector\\_mode.cs](#)

## 12.4 Galil Class Reference

### Public Member Functions

- **Galil** (std::string address="")
- std::string **connection** ()
- std::string **command** (const std::string &command="MG TIME", const std::string &terminator="\r", const std::string &ack=":", bool trim=true)
- double **commandValue** (const std::string &command="MG TIME")
- std::string **message** (int timeout\_ms=500)
- int **interrupt** (int timeout\_ms=500)
- std::string **programUpload** ()
- void **programDownload** (const std::string &program="MG TIME\rEN")
- void **programUploadFile** (const std::string &file="program.dmc")
- void **programDownloadFile** (const std::string &file="program.dmc")
- [std::vector](#)< double > **arrayUpload** (const std::string &name="array")
- void **arrayDownload** (const [std::vector](#)< double > &array, const std::string &name="array")
- void **arrayUploadFile** (const std::string &file="arrays.csv", const std::string &names="")
- void **arrayDownloadFile** (const std::string &file="arrays.csv")
- void **firmwareDownloadFile** (const std::string &file="firmware.hex")
- int **write** (const std::string &bytes="\r")
- std::string **read** ()
- [std::vector](#)< std::string > **sources** ()
- void **recordsStart** (double period\_ms=-1)
- [std::vector](#)< char > **record** (const std::string &method="QR")
- double **sourceValue** (const [std::vector](#)< char > &record, const std::string &source="TIME")
- std::string **source** (const std::string &field="Description", const std::string &source="TIME")
- void **setSource** (const std::string &field="Description", const std::string &source="TIME", const std::string &to="Sample counter")

### Static Public Member Functions

- static std::string **libraryVersion** ()
- static [std::vector](#)< std::string > **addresses** ()

### Data Fields

- int **timeout\_ms**

#### 12.4.1 Detailed Description

Definition at line 25 of file Galil.h.

The documentation for this class was generated from the following files:

- [Galil.h](#)
- [gcl\\_datarecord.cpp](#)
- [gcl\\_galil.cpp](#)

## 12.5 GalilPrivate Class Reference

### Public Member Functions

- **GalilPrivate** ([Galil](#) \*galil\_ptr, [GCon](#) gclib\_handle)
- void **InitializeDataRecord** ()

### Data Fields

- [GCon](#) g
- char **tbuf** [TRAFFICBUF]
- std::unordered\_map< std::string, [Source](#) > **map**

### 12.5.1 Detailed Description

Definition at line 32 of file gcl\_galil.h.

The documentation for this class was generated from the following files:

- [gcl\\_galil.h](#)
- [gcl\\_datarecord.cpp](#)

## 12.6 gclib Class Reference

Provides a class that binds to gclib's unmanaged dll. Wraps each call and provides a more user-friendly interface for use in C#.

### Data Structures

- class [GclibError](#)
- interface [GDataRecord](#)
- struct [GDataRecord1802](#)  
*Data record struct for DMC-1802 controllers.*
- struct [GDataRecord1806](#)  
*Data record struct for DMC-1806 controller.*
- struct [GDataRecord2103](#)  
*Data record struct for DMC-2103 controllers.*
- struct [GDataRecord30000](#)  
*Data record struct for DMC-30010 controllers.*
- struct [GDataRecord4000](#)  
*Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0.*
- struct [GDataRecord47000\\_ENC](#)  
*Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields.*
- struct [GDataRecord47162](#)  
*Data record struct for RIO-47162.*
- struct [GDataRecord47300\\_24EX](#)  
*Data record struct for RIO-47300 with 24EX I/O daughter board.*
- struct [GDataRecord47300\\_ENC](#)  
*Data record struct for RIO-47300. Includes encoder fields.*
- struct [GDataRecord52000](#)  
*Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40.*
- class [py](#)



## Public Member Functions

- [gclib](#) ()  
*Constructor of the gclib wrapper class.*
- [string\[\] GAddresses](#) ()  
*Return a string array of available connection addresses.*
- [void GArrayDownload](#) (string array\_name, ref List< double > data, Int16 first=-1, Int16 last=-1)  
*Downloads array data to a pre-dimensioned array in the controller's array table.*
- [void GArrayDownloadFile](#) (string Path)  
*Allows downloading of a program array file to the controller.*
- [List< double > GArrayUpload](#) (string array\_name, Int16 first=-1, Int16 last=-1)  
*Uploads array data from the controller's array table.*
- [void GArrayUploadFile](#) (string Path, string Names)  
*Allows uploading of a program array file from the controller to an array CSV file.*
- [void GAssign](#) (string ip, string mac)  
*Assigns IP address over the Ethernet to a controller at a given MAC address.*
- [void GClose](#) ()  
*Used to close a connection to Galil hardware.*
- [string GCommand](#) (string Command, bool Trim=true)  
*Used for command-and-response transactions.*
- [Int16 GCmdl](#) (string Command)  
*Used for command-and-response transactions.*
- [double GCmdD](#) (string Command)  
*Used for command-and-response transactions.*
- [void GFirmwareDownload](#) (string filepath)  
*Upgrade firmware.*
- [string GInfo](#) ()  
*Provides a useful connection string.*
- [byte GInterrupt](#) ()  
*Provides access to PCI and UDP interrupts from the controller.*
- [string\[\] GIpRequests](#) ()  
*Provides a list of all Galil controllers requesting IP addresses via BOOT-P or DHCP.*
- [string GMessage](#) ()  
*Provides access to unsolicited messages.*
- [void GMotionComplete](#) (string axes)  
*Blocking call that returns once all axes specified have completed their motion.*
- [void GOpen](#) (string address)  
*Used to open a connection to Galil hardware.*
- [void GProgramDownload](#) (string program, string preprocessor="")  
*Allows downloading of a DMC program from a string buffer.*
- [void GProgramDownloadFile](#) (string file\_path, string preprocessor="")  
*Allows downloading of a DMC program from file.*
- [string GProgramUpload](#) ()  
*Allows uploading of a DMC program to a string.*
- [void GProgramUploadFile](#) (string file\_path)  
*Allows uploading of a DMC program to a file.*
- [byte\[\] GRead](#) ()  
*Performs a read on the connection.*
- [T GRecord< T >](#) (bool async)  
*Used for retrieving data records from the controller.*
- [void GRecordRate](#) (double period\_ms)

- Sets the asynchronous data record to a user-specified period via DR.*

  - void [GTimeout](#) (Int16 timeout\_ms)

*Set the timeout of communication transactions. Use -1 to set the original timeout from [GOpen\(\)](#).*
- string [GVersion](#) ()
- Used to get the gclib version.*

  - void [GWrite](#) (string buffer)

*Performs a write on the connection.*
- string[] [GSetupDownloadFile](#) (string path, Int32 options)
- Allows downloading of a [Galil](#) compressed backup (gcb) file to the controller.*

  - void [GSetServer](#) (string server\_name)

*Connects gclib to a new gcaps server*
- string [GServerStatus](#) ()
- Retrieves the name of your local gcaps server and whether or not it is currently published*

  - string[] [GListServers](#) ()

*Retrieves a list of gcaps servers that are advertising themselves on the local network*
- void [GPublishServer](#) (string server\_name, bool publish, bool save)
- Publishes or removes local gcaps server from the network*

  - string[] [GRemoteConnections](#) ()

*Returns a list of IP Addresses that currently have an open connection to your hardware.*

## Data Fields

- **argtypes**
- **restype**

### 12.6.1 Detailed Description

Provides a class that binds to gclib's unmanaged dll. Wraps each call and provides a more user-friendly interface for use in C#.

The Gclib class assumes the default installation of gclib, "C:\Program Files (x86)\Galil\gclib". If the dlls are elsewhere, change the path strings GclibDllPath\_, and GcliboDllPath\_.

Definition at line 67 of file gclib.cs.

### 12.6.2 Constructor & Destructor Documentation

#### 12.6.2.1 gclib()

`gclib ( ) [inline]`

Constructor of the gclib wrapper class.

Checks to ensure gclib dlls are in the correct location.

#### Exceptions

<i>System.Exception</i>	Will throw an exception if either dll isn't found.
-------------------------	--

Definition at line 85 of file gclib.cs.

### 12.6.3 Member Function Documentation

### 12.6.3.1 GAddresses()

```
string [] GAddresses ( ) [inline]
```

Return a string array of available connection addresses.

#### Returns

String array containing all available [Galil](#) Ethernet controllers, PCI controllers, and COM ports.

Wrapper around gclib [GAddresses\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo\\_8h.html#a6a6114683ed5749519b64f19512c24d6](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#a6a6114683ed5749519b64f19512c24d6) An empty array is returned on error.

Definition at line 102 of file gclib.cs.

Referenced by [Examples.Remote\\_Client\(\)](#).

### 12.6.3.2 GArrayDownload()

```
void GArrayDownload (
    string array_name,
    ref List< double > data,
    Int16 first = -1,
    Int16 last = -1 ) [inline]
```

Downloads array data to a pre-dimensioned array in the controller's array table.

#### Parameters

<i>array_name</i>	String containing the name of the array to download. Must match the array name used in DM.
<i>data</i>	A list of doubles, to be downloaded.
<i>first</i>	The first element of the array for sub-array downloads.
<i>last</i>	The last element of the array for sub-array downloads.

Wrapper around gclib [GArrayDownload\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo\\_8h.html#a6ea5ae6d167675e4c27ccfaf2f240f8a](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#a6ea5ae6d167675e4c27ccfaf2f240f8a) The array must already exist on the controller, see DM and LA.

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 126 of file gclib.cs.

### 12.6.3.3 GArrayDownloadFile()

```
void GArrayDownloadFile (
    string Path ) [inline]
```

Allows downloading of a program array file to the controller.

#### Parameters

<i>Path</i>	The full filepath of the array csv file.
-------------	--

Wrapper around gclib [GArrayDownload\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo\\_8h.html#a14b448ab8c7e6cf495865af301be398e](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#a14b448ab8c7e6cf495865af301be398e)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 153 of file gclib.cs.

### 12.6.3.4 GArrayUpload()

```
List<double> GArrayUpload (
    string array_name,
    Int16 first = -1,
    Int16 last = -1 ) [inline]
```

Uploads array data from the controller's array table.

#### Parameters

<i>array_name</i>	String containing the name of the array to upload.
<i>first</i>	The first element of the array for sub-array uploads.
<i>last</i>	The last element of the array for sub-array uploads.

#### Returns

The desired array as a list of doubles.

Wrapper around gclib [GArrayUpload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#af215806ec26ba06ed3f174ebeeafa7a7), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#af215806ec26ba06ed3f174ebeeafa7a7](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#af215806ec26ba06ed3f174ebeeafa7a7)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 173 of file gclib.cs.

### 12.6.3.5 GArrayUploadFile()

```
void GArrayUploadFile (
    string Path,
    string Names ) [inline]
```

Allows uploading of a program array file from the controller to an array CSV file.

#### Parameters

<i>Path</i>	The full filepath of the array csv file to save.
<i>Names</i>	A space separated list of the array names to upload. A null string uploads all arrays in the array table (LA).

Wrapper around gclib [GArrayUpload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#af215806ec26ba06ed3f174ebeeafa7a7), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#af215806ec26ba06ed3f174ebeeafa7a7](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#af215806ec26ba06ed3f174ebeeafa7a7)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 206 of file gclib.cs.

### 12.6.3.6 GAssign()

```
void GAssign (
```

```
string ip,
string mac ) [inline]
```

Assigns IP address over the Ethernet to a controller at a given MAC address.

#### Parameters

<i>ip</i>	The ip address to assign. The hardware should not yet have an IP address.
<i>mac</i>	The MAC address of the hardware.

Wrapper around gclib [GAssign\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#acc996b7c22cfed8e5573d096ef1ab759), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#acc996b7c22cfed8e5573d096ef1ab759](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#acc996b7c22cfed8e5573d096ef1ab759)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 224 of file gclib.cs.

Referenced by Examples.IP\_Assigner().

#### 12.6.3.7 GClose()

```
void GClose ( ) [inline]
```

Used to close a connection to Galil hardware.

Wrapper around gclib [GClose\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a24a437bcde9637b0db4b94176563a052), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a24a437bcde9637b0db4b94176563a052](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a24a437bcde9637b0db4b94176563a052) Be sure to call [GClose\(\)](#) whenever a connection is finished.

Definition at line 239 of file gclib.cs.

Referenced by gclib.py.\_\_del\_\_(), Commands\_Example.Main(), Contour\_Example.Main(), IP\_Assigner\_Example.Main(), Jog\_Example.Main(), Message\_Example.Main(), Motion\_Complete\_Example.Main(), Position\_Tracking\_Example.Main(), Record\_Position\_Example.Main(), and Vector\_Mode\_Example.Main().

#### 12.6.3.8 GCmdD()

```
double GCmdD (
string Command ) [inline]
```

Used for command-and-response transactions.

#### Parameters

<i>Command</i>	The command to send to the controller. Do not append a carriage return. Use only ASCII-based commands.
----------------	--

#### Returns

The command's response parsed as a double.

Wrapper around gclib [GCmdD\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a5ac031e76efc965affdd73a1bec084a8](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8)

Definition at line 301 of file gclib.cs.

References GCommand().

Referenced by Examples.Commands().

#### 12.6.3.9 GCmdI()

```
Int16 GCmdI (
string Command ) [inline]
```

Used for command-and-response transactions.

#### Parameters

<i>Command</i>	The command to send to the controller. Do not append a carriage return. Use only ASCII-based commands.
----------------	--

#### Returns

The command's response parsed as an integer.

Wrapper around gclib [GCmdI\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a5ac031e76efc965affdd73a1bec084a8](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8)

Definition at line 288 of file gclib.cs.

References GCommand().

Referenced by Examples.Commands(), Examples.Contour(), Examples.Record\_Position(), and Examples.Vector\_Mode().

### 12.6.3.10 GCommand()

```
string GCommand (
    string Command,
    bool Trim = true ) [inline]
```

Used for command-and-response transactions.

#### Parameters

<i>Command</i>	The command to send to the controller. Do not append a carriage return. Use only ASCII-based commands.
<i>Trim</i>	If true, the response will be trimmed of the trailing colon and any leading or trailing whitespace.

#### Returns

The command's response.

Wrapper around gclib [GCommand\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a5ac031e76efc965affdd73a1bec084a8](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 257 of file gclib.cs.

Referenced by Examples.Commands(), Examples.Contour(), GCmdD(), GCmdI(), Examples.IP\_Assigner(), Examples.Jog(), Examples.Message(), Examples.Motion\_Complete(), Examples.Position\_Tracking(), Examples.PrintError(), Examples.Record\_Position(), and Examples.Vector\_Mode().

### 12.6.3.11 GFirmwareDownload()

```
void GFirmwareDownload (
    string filepath ) [inline]
```

Upgrade firmware.

#### Parameters

<i>filepath</i>	The full filepath of the firmware hex file.
-----------------	---

Wrapper around gclib [GFirmwareDownload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a1878a2285ff17897fa4fb20182ba6fdf), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a1878a2285ff17897fa4fb20182ba6fdf](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a1878a2285ff17897fa4fb20182ba6fdf)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 330 of file gclib.cs.

#### 12.6.3.12 GInfo()

```
string GInfo ( ) [inline]
```

Provides a useful connection string.

Wrapper around gclib [GInfo\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a08abfcff8a1a85a01987859473167518), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a08abfcff8a1a85a01987859473167518](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a08abfcff8a1a85a01987859473167518)

#### Returns

String containing connection information, e.g. "192.168.0.43, DMC4020 Rev 1.2c, 291". A null string indicates an error was returned from the library.

Definition at line 344 of file gclib.cs.

Referenced by Examples.IP\_Assigner().

#### 12.6.3.13 GInterrupt()

```
byte GInterrupt ( ) [inline]
```

Provides access to PCI and UDP interrupts from the controller.

#### Returns

The status byte from the controller. Zero will be returned if a status byte is not read.

Wrapper around gclib [GInterrupt\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5bcf802404a96343e7593d247b67f132), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a5bcf802404a96343e7593d247b67f132](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5bcf802404a96343e7593d247b67f132) -s ALL or -s EI must be specified in the address argument of [GOpen\(\)](#) to receive interrupts.

Definition at line 364 of file gclib.cs.

Referenced by Examples.Motion\_Complete().

#### 12.6.3.14 GIpRequests()

```
string [] GIpRequests ( ) [inline]
```

Provides a list of all [Galil](#) controllers requesting IP addresses via BOOT-P or DHCP.

#### Returns

Each line of the returned data will be of the form "model, serial\_number, mac".

Wrapper around gclib [GIpRequests\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a0afb4c82642a4ef86f997c39a5518952), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a0afb4c82642a4ef86f997c39a5518952](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a0afb4c82642a4ef86f997c39a5518952) An empty array is returned on error. Call will take roughly 5 seconds to return.

Definition at line 386 of file gclib.cs.

Referenced by Examples.IP\_Assigner().

#### 12.6.3.15 GListServers()

```
string [] GListServers ( ) [inline]
```

Retrieves a list of gcaps servers that are advertising themselves on the local network

**Returns**

A list of available gcaps server names

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 716 of file gclib.cs.

Referenced by Examples.Remote\_Client().

**12.6.3.16 GMessage()**

```
string GMessage ( ) [inline]
```

Provides access to unsolicited messages.

**Returns**

String containing all messages received by controller.

Wrapper around gclib [GMessage\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#aabc5eaa09ddec55ab8ee048b916cbcd](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#aabc5eaa09ddec55ab8ee048b916cbcd) An empty string is returned on error. -s ALL or -s MG must be specified in the address argument of [GOpen\(\)](#) to receive messages.

Definition at line 407 of file gclib.cs.

Referenced by Examples.Message().

**12.6.3.17 GMotionComplete()**

```
void GMotionComplete (
    string axes ) [inline]
```

Blocking call that returns once all axes specified have completed their motion.

**Parameters**

<i>axes</i>	A string containing a multiple-axes mask. Every character in the string should be a valid argument to MG_BGm, i.e. XYZWABCEFGHST.
-------------	---

Wrapper around gclib [GMotionComplete\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a19c220879442987970706444197f397a](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a19c220879442987970706444197f397a)

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 428 of file gclib.cs.

Referenced by Examples.Contour(), Examples.Jog(), Examples.Position\_Tracking(), and Examples.Vector\_Mode().

**12.6.3.18 GOpen()**

```
void GOpen (
    string address ) [inline]
```

Used to open a connection to [Galil](#) hardware.

**Parameters**

<i>address</i>	Address string including any connection switches. See gclib documentation for <a href="#">GOpen()</a> .
----------------	---



Wrapper around gclib [GOpen\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#aef4aec8a85630eed029b7a46aea7db54), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#aef4aec8a85630eed029b7a46aea7db54](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#aef4aec8a85630eed029b7a46aea7db54)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 445 of file gclib.cs.

Referenced by Examples.IP\_Assigner(), Commands\_Example.Main(), Contour\_Example.Main(), Jog\_Example.Main(), Message\_Example.Main(), Motion\_Complete\_Example.Main(), Position\_Tracking\_Example.Main(), Record\_Position\_Example.Main(), and Vector\_Mode\_Example.Main().

#### 12.6.3.19 GProgramDownload()

```
void GProgramDownload (
    string program,
    string preprocessor = "" ) [inline]
```

Allows downloading of a DMC program from a string buffer.

#### Parameters

<i>program</i>	The program to download.
<i>preprocessor</i>	Preprocessor directives. Use nullstring for none.

Wrapper around gclib [GProgramDownload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#acafe19b2dd0537ff458e3c8afe3acfeb), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#acafe19b2dd0537ff458e3c8afe3acfeb](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#acafe19b2dd0537ff458e3c8afe3acfeb)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 465 of file gclib.cs.

Referenced by Examples.Message(), and Examples.Record\_Position().

#### 12.6.3.20 GProgramDownloadFile()

```
void GProgramDownloadFile (
    string file_path,
    string preprocessor = "" ) [inline]
```

Allows downloading of a DMC program from file.

#### Parameters

<i>file_path</i>	The full filepath of the DMC file.
<i>preprocessor</i>	Preprocessor directives. Use nullstring for none.

Wrapper around gclib [GProgramDownloadFile\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a8e44e2e321df9e7b8c538bf2d640633f), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a8e44e2e321df9e7b8c538bf2d640633f](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a8e44e2e321df9e7b8c538bf2d640633f)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 483 of file gclib.cs.

### 12.6.3.21 GProgramUpload()

```
string GProgramUpload ( ) [inline]
```

Allows uploading of a DMC program to a string.

Wrapper around gclib [GProgramUpload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a80a653ce387a2bd16bde2793c6de77e9), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a80a653ce387a2bd16bde2793c6de77e9](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a80a653ce387a2bd16bde2793c6de77e9)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 499 of file gclib.cs.

### 12.6.3.22 GProgramUploadFile()

```
void GProgramUploadFile (
    string file_path ) [inline]
```

Allows uploading of a DMC program to a file.

#### Parameters

<i>file_path</i>	The full filepath of the DMC file to save.
------------------	--

Wrapper around gclib [GProgramUploadFile\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a38c5565afc11762fa19d37fbaa3c9aa3), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a38c5565afc11762fa19d37fbaa3c9aa3](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a38c5565afc11762fa19d37fbaa3c9aa3)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 520 of file gclib.cs.

### 12.6.3.23 GPublishServer()

```
void GPublishServer (
    string server_name,
    bool publish,
    bool save ) [inline]
```

Publishes or removes local gcaps server from the network

#### Parameters

<i>server_name</i>	Name to publish server under.
<i>publish</i>	True=publish server, False=remove server.
<i>save</i>	Save this configuration for future server reboots.

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 738 of file gclib.cs.

Referenced by Examples.Remote\_Server().

### 12.6.3.24 GRead()

```
byte [] GRead ( ) [inline]
```

Performs a read on the connection.

#### Returns

String containing the read data, or a nullstring if nothing was read or an error occurred.

Wrapper around gclib [GRead\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#adab6ec79b7e1bc7f0266684dd3434923), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#adab6ec79b7e1bc7f0266684dd3434923](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#adab6ec79b7e1bc7f0266684dd3434923)  
Definition at line 536 of file gclib.cs.

### 12.6.3.25 GRecord< T >()

```
T GRecord< T > (
    bool async ) [inline]
```

Used for retrieving data records from the controller.

#### Returns

A struct containing the information from the retrieved data record.

#### Parameters

<i>async</i>	False to user QR, True to use DR.
--------------	-----------------------------------

Wrapper around gclib [GRecord\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a1f39cd57dcfa55d065c972a020b1f8ee), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a1f39cd57dcfa55d065c972a020b1f8ee](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a1f39cd57dcfa55d065c972a020b1f8ee) To use async, -s ALL or -s DR must be specified in the address argument of [GOpen\(\)](#), and the records must be started via DR or RecordRate().

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

#### Type Constraints

**T**: *struct*

**T**: [GDataRecord](#)

Definition at line 566 of file gclib.cs.

### 12.6.3.26 GRecordRate()

```
void GRecordRate (
    double period_ms ) [inline]
```

Sets the asynchronous data record to a user-specified period via DR.

#### Parameters

<i>period_ms</i>	Period, in milliseconds, to set up for the asynchronous data record.
------------------	--

Wrapper around gclib [GRecordRate\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#ada86dc9d33ac961412583881963a1b8a), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#ada86dc9d33ac961412583881963a1b8a](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#ada86dc9d33ac961412583881963a1b8a) Takes TM and product type into account and sets the DR period to the period requested by the user, if possible.

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 588 of file gclib.cs.

**12.6.3.27 GRemoteConnections()**

```
string [] GRemoteConnections ( ) [inline]
```

Returns a list of IP Addresses that currently have an open connection to your hardware.

**Returns**

Returns a list of IP Addresses that currently have an open connection to your hardware.

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 751 of file gclib.cs.

**12.6.3.28 GServerStatus()**

```
string GServerStatus ( ) [inline]
```

Retrieves the name of your local gcaps server and whether or not it is currently published

**Returns**

A string in the form "<server\_name>, <isPublished>"

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 701 of file gclib.cs.

**12.6.3.29 GSetServer()**

```
void GSetServer (
    string server_name ) [inline]
```

Connects gclib to a new gcaps server

**Parameters**

<i>server_name</i>	Name of the server to connect.
--------------------	--------------------------------

Wrapper around gclib [GSetServer\(\)](#), Call GSetServer("Local") to connect gclib back to local gcaps server

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 686 of file gclib.cs.

Referenced by Examples.Remote\_Client().

**12.6.3.30 GSetupDownloadFile()**

```
string [] GSetupDownloadFile (
    string path,
    Int32 options ) [inline]
```

Allows downloading of a [Galil](#) compressed backup (gcb) file to the controller.

**Parameters**

<i>path</i>	The full filepath of the gcb file.
<i>options</i>	A bit mask indicating which sectors of the gcb file to restore to the controller.

**Returns**

The controller information stored in the gcb file.

Wrapper around gclib [GSetupDownloadFile\(\)](#),

If options is specified as 0, the return string will have a number appended corresponding to a bit mask of the available gcb sectors

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 655 of file gclib.cs.

**12.6.3.31 GTimeout()**

```
void GTimeout (
    Int16 timeout_ms ) [inline]
```

Set the timeout of communication transactions. Use -1 to set the original timeout from [GOpen\(\)](#).

**Parameters**

<i>timeout_ms</i>	New timeout in milliseconds.
-------------------	------------------------------

Wrapper around gclib [GTimeout\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←\\_8h.html#a179aa2d1b8e2227944cc06a7ceaf5640](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#a179aa2d1b8e2227944cc06a7ceaf5640)

Definition at line 604 of file gclib.cs.

Referenced by [Examples.Motion\\_Complete\(\)](#), and [gclib.py.timeout\(\)](#).

**12.6.3.32 GVersion()**

```
string GVersion ( ) [inline]
```

Used to get the gclib version.

**Returns**

The library version, e.g. "104.73.179". A null string indicates an error was returned from the library.

Wrapper around gclib [GVersion\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←\\_8h.html#a1784b39416b77af20efc98a05f8ce475](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#a1784b39416b77af20efc98a05f8ce475)

Definition at line 614 of file gclib.cs.

### 12.6.3.33 GWrite()

```
void GWrite (
    string buffer ) [inline]
```

Performs a write on the connection.

#### Parameters

<i>buffer</i>	The user's write buffer. To send a <a href="#">Galil</a> command, a terminating carriage return is usually required.
---------------	--

Wrapper around gclib [GWrite\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#abe28ebaecd5b3940adf4e145d40e5456](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#abe28ebaecd5b3940adf4e145d40e5456)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

Definition at line 635 of file gclib.cs.

The documentation for this class was generated from the following files:

- [gclib.cs](#)
- [gclib.py](#)

## 12.7 Gclib Class Reference

Provides a class that binds to gclib's unmanaged dll. Wraps each call and provides a more user-friendly interface for use in Visual Basic.

### Data Structures

- interface [GDataRecord](#)
- struct [GDataRecord1802](#)
- struct [GDataRecord1806](#)
- struct [GDataRecord2103](#)
- struct [GDataRecord30000](#)
- struct [GDataRecord4000](#)
- struct [GDataRecord47000\\_ENC](#)
- struct [GDataRecord47162](#)
- struct [GDataRecord47300\\_24EX](#)
- struct [GDataRecord47300\\_ENC](#)
- struct [GDataRecord52000](#)

### Public Member Functions

- void [New](#) ()  
*Constructor of the gclib wrapper class.*
- [GAddresses](#) ()  
*Return a string array of available connection addresses.*
- void [GArrayDownload](#) (string array\_name, ref List< double > data, Int16 first=-1, Int16 last=-1)  
*Downloads array data to a pre-dimensioned array in the controller's array table.*
- void [GArrayDownloadFile](#) (string Path)  
*Allows downloading of a program array file to the controller.*
- List< double > [GArrayUpload](#) (string array\_name, Int16 first=-1, Int16 last=-1)  
*Uploads array data from the controller's array table.*
- void [GArrayUploadFile](#) (string Path, string Names)

- Allows uploading of a program array file from the controller to an array CSV file.*

  - void [GAssign](#) (string ip, string mac)

*Assigns IP address over the Ethernet to a controller at a given MAC address.*
- void [GClose](#) ()

*Used to close a connection to [Galil](#) hardware.*
- string [GCommand](#) (string Command, bool Trim=True)

*Used for command-and-response transactions.*
- Int16 [GCmdI](#) (string Command)

*Used for command-And-response transactions.*
- double [GCmdD](#) (string Command)

*Used for command-And-response transactions.*
- void [GFirmwareDownload](#) (string filepath)

*Upgrade firmware.*
- string [GInfo](#) ()

*Provides a useful connection string.*
- byte [GInterrupt](#) ()

*Provides access to PCI and UDP interrupts from the controller.*
- [GIpRequests](#) ()

*Provides a list of all [Galil](#) controllers requesting IP addresses via BOOT-P or DHCP.*
- string [GMessage](#) ()

*Provides access to unsolicited messages.*
- void [GMotionComplete](#) (string axes)

*Blocking call that returns once all axes specified have completed their motion.*
- void [GOpen](#) (string address)

*Used to open a connection to [Galil](#) hardware.*
- void [GProgramDownload](#) (ref string program, string preprocessor="")

*Allows downloading of a DMC program from a string buffer.*
- void [GProgramDownloadFile](#) (string file\_path, string preprocessor="")

*Allows downloading of a DMC program from file.*
- string [GProgramUpload](#) ()

*Allows uploading of a DMC program to a string.*
- void [GProgramUploadFile](#) (string file\_path)

*Allows uploading of a DMC program to a file.*
- [GRead](#) ()

*Performs a read on the connection.*
- T [GRecord](#) (Of [GDataRecord](#))(async T bool)

*Used for retrieving data records from the controller.*
- void [GRecordRate](#) (double period\_ms)

*Sets the asynchronous data record to a user-specified period via DR.*
- void [GTimeout](#) (Int16 timeout\_ms)

*Set the timeout of communication transactions. Use -1 to set the original timeout from [GOpen\(\)](#).*
- string [GVersion](#) ()

*Used to get the gclib version.*
- void [GWrite](#) (ref string buffer)

*Performs a write on the connection.*
- string[] [GSetupDownloadFile](#) (string Path, Int32 Options)

*Allows downloading of a [Galil](#) compressed backup (gcb) file to the controller.*
- void [GSetServer](#) (string server\_name)

*Connects gclib to a New gcaps server*
- string [GServerStatus](#) ()

*Retrieves the name of your local gcaps server And whether Or Not it Is currently published*

- [GListServers](#) ()  
*Retrieves a list of gcaps servers that are advertising themselves on the local network*
- void [GPublishServer](#) (string server\_name, bool publish, bool save)  
*Publishes Or removes local gcaps server from the network*
- [GRemoteConnections](#) ()  
*Returns a list of IP Addresses that currently have an open connection to your hardware.*

### 12.7.1 Detailed Description

Provides a class that binds to gclib's unmanaged dll. Wraps each call and provides a more user-friendly interface for use in Visual Basic.

The [Gclib](#) class assumes the default installation of gclib, "C:\Program Files (x86)\Galil\gclib". If the dlls are elsewhere, change the path strings `GclibDllPath_`, and `GcliboDllPath_`.

Definition at line 44 of file `gclib.vb`.

### 12.7.2 Member Function Documentation

#### 12.7.2.1 GAddresses()

`GAddresses` ( )

Return a string array of available connection addresses.

##### Returns

String array containing all available [Galil](#) Ethernet controllers, PCI controllers, and COM ports.

Wrapper around gclib [GAddresses\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo\\_8h.html#a6a6114683ed5749519b64f19512c24d6](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#a6a6114683ed5749519b64f19512c24d6) An empty array is returned on error.

#### 12.7.2.2 GArrayDownload()

```
void GArrayDownload (
    string array_name,
    ref List< double > data,
    Int16 first = -1,
    Int16 last = -1 )
```

Downloads array data to a pre-dimensioned array in the controller's array table.

##### Parameters

<i>array_name</i>	String containing the name of the array to download. Must match the array name used in DM.
<i>data</i>	A list of doubles, to be downloaded.
<i>first</i>	The first element of the array for sub-array downloads.
<i>last</i>	The last element of the array for sub-array downloads.

Wrapper around gclib [GArrayDownload\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo\\_8h.html#a6ea5ae6d167675e4c27ccfaf2f240f8a](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#a6ea5ae6d167675e4c27ccfaf2f240f8a) The array must already exist on the controller, see DM and LA.

##### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than <code>G_NO_ERROR</code> is received from gclib.
-------------------------	--



### 12.7.2.3 GArrayDownloadFile()

```
void GArrayDownloadFile (
    string Path )
```

Allows downloading of a program array file to the controller.

#### Parameters

<i>Path</i>	The full filepath of the array csv file.
-------------	--

Wrapper around gclib [GArrayDownload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#a14b448ab8c7e6cf495865af301be398e), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←\\_8h.html#a14b448ab8c7e6cf495865af301be398e](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#a14b448ab8c7e6cf495865af301be398e)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

### 12.7.2.4 GArrayUpload()

```
List<double> GArrayUpload (
    string array_name,
    Int16 first = -1,
    Int16 last = -1 )
```

Uploads array data from the controller's array table.

#### Parameters

<i>array_name</i>	String containing the name of the array to upload.
<i>first</i>	The first element of the array for sub-array uploads.
<i>last</i>	The last element of the array for sub-array uploads.

#### Returns

The desired array as a list of doubles.

Wrapper around gclib [GArrayUpload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#af215806ec26ba06ed3f174ebeeafa7a7), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←\\_8h.html#af215806ec26ba06ed3f174ebeeafa7a7](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#af215806ec26ba06ed3f174ebeeafa7a7)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

### 12.7.2.5 GArrayUploadFile()

```
void GArrayUploadFile (
    string Path,
    string Names )
```

Allows uploading of a program array file from the controller to an array CSV file.

#### Parameters

<i>Path</i>	The full filepath of the array csv file to save.
<i>Names</i>	A space separated list of the array names to upload. A null string uploads all arrays in the array table (LA).

Wrapper around gclib [GArrayUpload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#af215806ec26ba06ed3f174ebeeafa7a7). [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#af215806ec26ba06ed3f174ebeeafa7a7](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#af215806ec26ba06ed3f174ebeeafa7a7)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

#### 12.7.2.6 GAssign()

```
void GAssign (
    string ip,
    string mac )
```

Assigns IP address over the Ethernet to a controller at a given MAC address.

#### Parameters

<i>ip</i>	The ip address to assign. The hardware should not yet have an IP address.
<i>mac</i>	The MAC address of the hardware.

Wrapper around gclib [GAssign\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#acc996b7c22cfed8e5573d096ef1ab759), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#acc996b7c22cfed8e5573d096ef1ab759](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#acc996b7c22cfed8e5573d096ef1ab759)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

#### 12.7.2.7 GClose()

```
void GClose ( )
```

Used to close a connection to [Galil](#) hardware.

Wrapper around gclib [GClose\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a24a437bcde9637b0db4b94176563a052), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a24a437bcde9637b0db4b94176563a052](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a24a437bcde9637b0db4b94176563a052) Be sure to call [GClose\(\)](#) whenever a connection is finished.

Referenced by gclib.py.\_\_del\_\_().

#### 12.7.2.8 GCmdD()

```
double GCmdD (
    string Command )
```

Used for command-And-response transactions.

#### Parameters

<i>Command</i>	The command to send to the controller. Do Not append a carriage return. Use only ASCII-based commmands.
----------------	---

#### Returns

The command's response parsed as a double.

Wrapper around gclib [GCmdD\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a5ac031e76efc965affdd73a1bec084a8](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8)

### 12.7.2.9 GCmdI()

```
Int16 GCmdI (
    string Command )
```

Used for command-And-response transactions.

#### Parameters

<i>Command</i>	The command to send to the controller. Do Not append a carriage return. Use only ASCII-based commmands.
----------------	---

#### Returns

The command's response parsed as an integer.

Wrapper around gclib [GCmdI\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a5ac031e76efc965affdd73a1bec084a8](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8)

### 12.7.2.10 GCommand()

```
string GCommand (
    string Command,
    bool Trim = True )
```

Used for command-and-response transactions.

#### Parameters

<i>Command</i>	The command to send to the controller. Do not append a carriage return. Use only ASCII-based commmands.
<i>Trim</i>	If true, the response will be trimmed of the trailing colon and any leading or trailing whitespace.

#### Returns

The command's response.

Wrapper around gclib [GCommand\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a5ac031e76efc965affdd73a1bec084a8](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a5ac031e76efc965affdd73a1bec084a8)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

### 12.7.2.11 GFirmwareDownload()

```
void GFirmwareDownload (
    string filepath )
```

Upgrade firmware.

#### Parameters

<i>filepath</i>	The full filepath of the firmware hex file.
-----------------	---

Wrapper around gclib [GFirmwareDownload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a1878a2285ff17897fa4fb20182ba6fdf), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a1878a2285ff17897fa4fb20182ba6fdf](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a1878a2285ff17897fa4fb20182ba6fdf)

## Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

**12.7.2.12 GInfo()**

```
string GInfo ( )
```

Provides a useful connection string.

Wrapper around gclib [GInfo\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#a08abfcff8a1a85a01987859473167518), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←\\_8h.html#a08abfcff8a1a85a01987859473167518](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#a08abfcff8a1a85a01987859473167518)

**Returns**

String containing connection information, e.g. "192.168.0.43, DMC4020 Rev 1.2c, 291". A null string indicates an error was returned from the library.

**12.7.2.13 GInterrupt()**

```
byte GInterrupt ( )
```

Provides access to PCI and UDP interrupts from the controller.

**Returns**

The status byte from the controller. Zero will be returned if a status byte is not read.

Wrapper around gclib [GInterrupt\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#a5bcf802404a96343e7593d247b67f132), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←\\_8h.html#a5bcf802404a96343e7593d247b67f132](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#a5bcf802404a96343e7593d247b67f132) -s ALL or -s EI must be specified in the address argument of [GOpen\(\)](#) to receive interrupts.

**12.7.2.14 GIpRequests()**

```
GIpRequests ( )
```

Provides a list of all [Galil](#) controllers requesting IP addresses via BOOT-P or DHCP.

**Returns**

Each line of the returned data will be of the form "model, serial\_number, mac".

Wrapper around gclib [GIpRequests\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#a0afb4c82642a4ef86f997c39a5518952), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←\\_8h.html#a0afb4c82642a4ef86f997c39a5518952](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo←_8h.html#a0afb4c82642a4ef86f997c39a5518952) An empty array is returned on error. Call will take roughly 5 seconds to return.

**12.7.2.15 GListServers()**

```
GListServers ( )
```

Retrieves a list of gcaps servers that are advertising themselves on the local network

**Returns**

A list of available gcaps server names

## Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

**12.7.2.16 GMessage()**

```
string GMessage ( )
```

Provides access to unsolicited messages.

**Returns**

String containing all messages received by controller.

Wrapper around gclib [GMessage\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#aabc5eaa09ddec55ab8ee048b916cbcd](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#aabc5eaa09ddec55ab8ee048b916cbcd) An empty string is returned on error. -s ALL or -s MG must be specified in the address argument of [GOpen\(\)](#) to receive messages.

**12.7.2.17 GMotionComplete()**

```
void GMotionComplete (
    string axes )
```

Blocking call that returns once all axes specified have completed their motion.

**Parameters**

<i>axes</i>	A string containing a multiple-axes mask. Every character in the string should be a valid argument to MG_BGm, i.e. XYZWABCEFGHST.
-------------	---

Wrapper around gclib [GMotionComplete\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a19c220879442987970706444197f397a](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a19c220879442987970706444197f397a)

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

**12.7.2.18 GOpen()**

```
void GOpen (
    string address )
```

Used to open a connection to [Galil](#) hardware.

**Parameters**

<i>address</i>	Address string including any connection switches. See gclib documentation for <a href="#">GOpen()</a> .
----------------	---

Wrapper around gclib [GOpen\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#aef4aec8a85630eed029b7a46aea7db54](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#aef4aec8a85630eed029b7a46aea7db54)

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

**12.7.2.19 GProgramDownload()**

```
void GProgramDownload (
    ref string program,
    string preprocessor = "" )
```

Allows downloading of a DMC program from a string buffer.

## Parameters

<i>program</i>	The program to download.
<i>preprocessor</i>	Preprocessor directives. Use nullstring for none.

Wrapper around gclib [GProgramDownload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#aca19b2dd0537ff458e3c8afe3acfeb), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#aca19b2dd0537ff458e3c8afe3acfeb](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#aca19b2dd0537ff458e3c8afe3acfeb)

## Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

**12.7.2.20 GProgramDownloadFile()**

```
void GProgramDownloadFile (
    string file_path,
    string preprocessor = "" )
```

Allows downloading of a DMC program from file.

## Parameters

<i>file_path</i>	The full filepath of the DMC file.
<i>preprocessor</i>	Preprocessor directives. Use nullstring for none.

Wrapper around gclib [GProgramDownloadFile\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a8e44e2e321df9e7b8c538bf2d640633f), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a8e44e2e321df9e7b8c538bf2d640633f](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a8e44e2e321df9e7b8c538bf2d640633f)

## Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

**12.7.2.21 GProgramUpload()**

```
string GProgramUpload ( )
```

Allows uploading of a DMC program to a string.

Wrapper around gclib [GProgramUpload\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a80a653ce387a2bd16bde2793c6de77e9), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a80a653ce387a2bd16bde2793c6de77e9](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a80a653ce387a2bd16bde2793c6de77e9)

## Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

**12.7.2.22 GProgramUploadFile()**

```
void GProgramUploadFile (
    string file_path )
```

Allows uploading of a DMC program to a file.

## Parameters

<i>file_path</i>	The full filepath of the DMC file to save.
------------------	--

Wrapper around gclib [GProgramUploadFile\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a38c5565afc11762fa19d37fbaa3c9aa3), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a38c5565afc11762fa19d37fbaa3c9aa3](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a38c5565afc11762fa19d37fbaa3c9aa3)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

#### 12.7.2.23 GPublishServer()

```
void GPublishServer (
    string server_name,
    bool publish,
    bool save )
```

Publishes Or removes local gcaps server from the network

#### Parameters

<i>server_name</i>	Name to publish server under.
<i>publish</i>	True=publish server, False=remove server.
<i>save</i>	Save this configuration for future server reboots.

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR Is received from gclib.
-------------------------	---

#### 12.7.2.24 GRead()

```
GRead ( )
```

Performs a read on the connection.

#### Returns

String containing the read data, or a nullstring if nothing was read or an error occurred.

Wrapper around gclib [GRead\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#adab6ec79b7e1bc7f0266684dd3434923), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#adab6ec79b7e1bc7f0266684dd3434923](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#adab6ec79b7e1bc7f0266684dd3434923)

#### 12.7.2.25 GRecord()

```
T GRecord (
    Of GDataRecord )
```

Used for retrieving data records from the controller.

#### Returns

A struct containing the information of the retrieved data record.

#### Parameters

<i>async</i>	False to user QR, True to use DR.
--------------	-----------------------------------

Wrapper around gclib [GRecord\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a1f39cd57dcfa55d065c972a020b1f8ee), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclib\\_8h.html#a1f39cd57dcfa55d065c972a020b1f8ee](http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h.html#a1f39cd57dcfa55d065c972a020b1f8ee) To use async, -s ALL or -s DR must be specified in

the address argument of [GOpen\(\)](#), and the records must be started via DR or RecordRate().

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

#### 12.7.2.26 GRecordRate()

```
void GRecordRate (
    double period_ms )
```

Sets the asynchronous data record to a user-specified period via DR.

#### Parameters

<i>period_ms</i>	Period, in milliseconds, to set up for the asynchronous data record.
------------------	--

Wrapper around gclib [GRecordRate\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo\\_8h.html#ada86dc9d33ac961412583881963a1b8a](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#ada86dc9d33ac961412583881963a1b8a) Takes TM and product type into account and sets the DR period to the period requested by the user, if possible.

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

#### 12.7.2.27 GRemoteConnections()

```
GRemoteConnections ( )
```

Returns a list of IP Addresses that currently have an open connection to your hardware.

#### Returns

Returns a list of IP Addresses that currently have an open connection to your hardware.

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR Is received from gclib.
-------------------------	---

#### 12.7.2.28 GServerStatus()

```
string GServerStatus ( )
```

Retrieves the name of your local gcaps server And whether Or Not it Is currently published

#### Returns

A string in the form "<server\_name>, <isPublished>"

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR Is received from gclib.
-------------------------	---



**12.7.2.29 GSetServer()**

```
void GSetServer (
    string server_name )
```

Connects gclib to a New gcaps server

**Parameters**

<i>server_name</i>	Name of the server to connect.
--------------------	--------------------------------

Wrapper around gclib [GSetServer\(\)](#), Call GSetServer("Local") to connect gclib back to local gcaps server

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR Is received from gclib.
-------------------------	---

**12.7.2.30 GSetupDownloadFile()**

```
string [] GSetupDownloadFile (
    string Path,
    Int32 Options )
```

Allows downloading of a [Galil](#) compressed backup (gcb) file to the controller.

**Parameters**

<i>Path</i>	The full filepath of the gcb file.
<i>Options</i>	A bit mask indicating which sectors of the gcb file to restore to the controller.

**Returns**

The controller information stored in the gcb file.

Wrapper around gclib [GSetupDownloadFile\(\)](#),

If options is specified as 0, the return string will have a number appended corresponding to a bit mask of the available gcb sectors

**Exceptions**

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

**12.7.2.31 GTimeout()**

```
void GTimeout (
    Int16 timeout_ms )
```

Set the timeout of communication transactions. Use -1 to set the original timeout from [GOpen\(\)](#).

**Parameters**

<i>timeout_ms</i>	New timeout in milliseconds.
-------------------	------------------------------

Wrapper around gclib [GTimeout\(\)](#), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo\\_8h.html#a179aa2d1b8e2227944cc06a7ceaf5640](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#a179aa2d1b8e2227944cc06a7ceaf5640)

Referenced by gclib.py.timeout().

### 12.7.2.32 GVersion()

```
string GVersion ( )
```

Used to get the gclib version.

#### Returns

The library version, e.g. "104.73.179". A null string indicates an error was returned from the library.

Wrapper around gclib [GVersion\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#a1784b39416b77af20efc98a05f8ce475), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo\\_8h.html#a1784b39416b77af20efc98a05f8ce475](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#a1784b39416b77af20efc98a05f8ce475)

### 12.7.2.33 GWrite()

```
void GWrite (
    ref string buffer )
```

Performs a write on the connection.

#### Parameters

<i>buffer</i>	The user's write buffer. To send a <a href="#">Galil</a> command, a terminating carriage return is usually required.
---------------	--

Wrapper around gclib [GWrite\(\)](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#abe28ebaecd5b3940adf4e145d40e5456), [http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo\\_8h.html#abe28ebaecd5b3940adf4e145d40e5456](http://www.galil.com/sw/pub/all/doc/gclib/html/gclibo_8h.html#abe28ebaecd5b3940adf4e145d40e5456)

#### Exceptions

<i>System.Exception</i>	Will throw an exception if anything other than G_NO_ERROR is received from gclib.
-------------------------	---

### 12.7.2.34 New()

```
void New ( )
```

Constructor of the gclib wrapper class.

Checks to ensure gclib dlls are in the correct location.

#### Exceptions

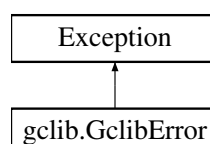
<i>System.Exception</i>	Will throw an exception if either dll isn't found.
-------------------------	--

The documentation for this class was generated from the following file:

- [gclib.vb](#)

## 12.8 gclib.GclibError Class Reference

Inheritance diagram for gclib.GclibError:



### 12.8.1 Detailed Description

Error class for non-zero gclib return codes.

Definition at line 137 of file gclib.py.

The documentation for this class was generated from the following file:

- [gclib.py](#)

## 12.9 GclibJava Class Reference

### Data Structures

- interface **Gclib**
- interface **Gclibo**

### Public Member Functions

- [GclibJava](#) ()
- void [GArrayDownload](#) (String arrayName, List< Double > data) throws GclibJavaException
- List< Double > [GArrayUpload](#) (String arrayName) throws GclibJavaException
- void [GClose](#) ()
- String [GCommand](#) (String command) throws GclibJavaException
- void [GFirmwareDownload](#) (String filePath) throws GclibJavaException
- byte [GInterrupt](#) () throws GclibJavaException
- String [GMessage](#) () throws GclibJavaException
- void [GOpen](#) (String address) throws GclibJavaException
- void [GProgramDownload](#) (String program, String preprocessor) throws GclibJavaException
- void [GProgramDownload](#) (String program) throws GclibJavaException
- String [GProgramUpload](#) () throws GclibJavaException
- String [GAddresses](#) () throws GclibJavaException
- void [GArrayDownloadFile](#) (String filePath) throws GclibJavaException
- void [GArrayUploadFile](#) (String filePath, String names) throws GclibJavaException
- void [GArrayUploadFile](#) (String filePath) throws GclibJavaException
- void [GAssign](#) (String ipAddress, String macAddress) throws GclibJavaException
- String [GInfo](#) () throws GclibJavaException
- String [GIpRequests](#) () throws GclibJavaException
- void [GProgramDownloadFile](#) (String filePath, String preprocessor) throws GclibJavaException
- void [GProgramDownloadFile](#) (String filePath) throws GclibJavaException
- void [GProgramUploadFile](#) (String filePath) throws GclibJavaException
- void [GSleep](#) (int timeout\_ms)
- void [GTimeout](#) (short timeout\_ms) throws GclibJavaException
- String [GVersion](#) () throws GclibJavaException
- void [GSetServer](#) (String server\_name) throws GclibJavaException
- String [GServerStatus](#) () throws GclibJavaException
- String [GListServers](#) () throws GclibJavaException
- void [GPublishServer](#) (String server\_name, int publish, int save) throws GclibJavaException
- String [GRemoteConnections](#) () throws GclibJavaException

### Protected Member Functions

- void [finalize](#) () throws Throwable

### 12.9.1 Detailed Description

[GclibJava](#) uses Java Native Access (JNA) internally to wrap the gclib functions in a Java-callable class.

Definition at line 36 of file GclibJava.java.

## 12.9.2 Constructor & Destructor Documentation

### 12.9.2.1 GclibJava()

`GclibJava ( )` [inline]

Constructor adds gclib to JNA's path.

Definition at line 45 of file GclibJava.java.

## 12.9.3 Member Function Documentation

### 12.9.3.1 finalize()

`void finalize ( )` throws `Throwable` [inline], [protected]

The last line of defense to close connection. Do NOT rely on `finalize()`, call `GCclose()` explicitly.

#### Exceptions

<code>Throwable</code>	super can throw.
------------------------	------------------

Definition at line 57 of file GclibJava.java.

References `GclibJava.GCclose()`.

### 12.9.3.2 GAddresses()

`String GAddresses ( )` throws `GclibJavaException` [inline]

Uses `GUtility()`, `G_UTIL_GCAPS_ADDRESSES` or `G_UTIL_ADDRESSES` to provide a listing of all available connection addresses.

#### Returns

String containing the available addresses.

10.1.3.91, DMC4020 Rev 1.2e, LAN, 10.1.3.10 192.168.0.63, DMC4040 Rev 1.2f, Static, 192.168.0.41 (192.0.0.↵42), RIO47102 Rev 1.1j, Static, 192.168.0.41 10.1., RIO47102 Rev 1.1j, Static, 192.168.0.41 GALILPCI1 COM1 COM2

#### Exceptions

<code>GclibJavaException</code>	If an error is generated by gclib.
---------------------------------	------------------------------------

Definition at line 359 of file GclibJava.java.

### 12.9.3.3 GArrayDownload()

`void GArrayDownload (`  
     `String arrayName,`  
     `List< Double > data )` throws `GclibJavaException` [inline]

Downloads array data to a pre-dimensioned array in the controller's array table.

#### Parameters

<code>arrayName</code>	String containing the name of the array to download. Must match the array name used in DM.
<code>data</code>	List containing the array data. The length of data may not be larger than the array dimensioned.

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 111 of file GclibJava.java.

**12.9.3.4 GArrayDownloadFile()**

```
void GArrayDownloadFile (  
    String filePath ) throws GclibJavaException [inline]
```

Array download from file. Downloads a csv file containing array data at file\_path. If the arrays don't exist, they will be dimensioned.

**Parameters**

<i>filePath</i>	String containing the path to the array file.
-----------------	---

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 373 of file GclibJava.java.

**12.9.3.5 GArrayUpload()**

```
List<Double> GArrayUpload (  
    String arrayName ) throws GclibJavaException [inline]
```

Uploads array data from the controller's array table.

**Parameters**

<i>arrayName</i>	String containing the name of the array to upload.
------------------	--

**Returns**

A List of Doubles, containing the array data.

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 129 of file GclibJava.java.

References e().

**12.9.3.6 GArrayUploadFile() [1/2]**

```
void GArrayUploadFile (  
    String filePath ) throws GclibJavaException [inline]
```

Overload of GArrayUploadFile to upload all arrays.

**Parameters**

<i>filePath</i>	String containing the path to the array file. File will be overwritten if it exists.
-----------------	--

## Exceptions

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 404 of file GclibJava.java.  
References GclibJava.GArrayUploadFile().

**12.9.3.7 GArrayUploadFile() [2/2]**

```
void GArrayUploadFile (
    String filePath,
    String names ) throws GclibJavaException [inline]
```

Array upload to file. Uploads the entire controller array table or a subset and saves the data as a csv file specified by file\_path.

## Parameters

<i>filePath</i>	String containing the path to the array file. File will be overwritten if it exists.
<i>names</i>	String containing the arrays to upload, delimited with space. "" uploads all arrays listed in LA.

## Exceptions

<a href="#">gclibjava.GclibJavaException</a>	If an error is generated by gclib.
--	------------------------------------

Definition at line 391 of file GclibJava.java.  
Referenced by GclibJava.GArrayUploadFile().

**12.9.3.8 GAssign()**

```
void GAssign (
    String ipAddress,
    String macAddress ) throws GclibJavaException [inline]
```

Uses [GUtility\(\)](#), G\_UTIL\_GCAPS\_ASSIGN or G\_UTIL\_ASSIGN to assign an IP address over the Ethernet to a controller at a given MAC address.

## Parameters

<i>ipAddress</i>	The IP address to assign.
<i>macAddress</i>	The MAC address of the hardware.

## Exceptions

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 419 of file GclibJava.java.

**12.9.3.9 GClose()**

```
void GClose ( ) [inline]
```

Closes a connection to a [Galil](#) Controller.

Definition at line 151 of file GclibJava.java.

Referenced by gclib.py.\_\_del\_\_(), GclibJava.finalize(), and GclibJava.GOpen().

**12.9.3.10 GCommand()**

String GCommand (   
                   String *command* ) throws [GclibJavaException](#) [inline]

Performs a command-and-response transaction on the connection.

**Parameters**

<i>command</i>	command string to send to the controller. The library will append a carriage return to the command string.
----------------	--

**Returns**

The response from the controller.

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 167 of file GclibJava.java.

**12.9.3.11 GFirmwareDownload()**

void GFirmwareDownload (   
                   String *filePath* ) throws [GclibJavaException](#) [inline]

Upgrade firmware.

**Parameters**

<i>filePath</i>	The full file path to the Galil-supplied firmware hex file. See <a href="http://www.galil.com/downloads/firmware">http://www.galil.com/downloads/firmware</a>
-----------------	---

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 190 of file GclibJava.java.

**12.9.3.12 GInfo()**

String GInfo ( ) throws [GclibJavaException](#) [inline]  
 Uses [GUtility\(\)](#) and G\_UTIL\_INFO to provide a useful connection string.

**Returns**

A String containing the info, e.g. 192.168.0.42, DMC30010 Rev 1.2i, 6969

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 432 of file GclibJava.java.

### 12.9.3.13 GInterrupt()

byte GInterrupt ( ) throws [GclibJavaException](#) [inline]

Provides access to PCI and UDP interrupts from the controller.

Interrupts can be generated automatically by the firmware on important events via EI (Enable Interrupt) or by the user in embedded DMC code via UI (User Interrupt). To use this function, -s EI must be used in the [GOpen\(\)](#) address string to subscribe to interrupts.

#### Returns

The status byte of the interrupt.

#### Exceptions

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 207 of file GclibJava.java.

### 12.9.3.14 GIpRequests()

String GIpRequests ( ) throws [GclibJavaException](#) [inline]

Uses [GUtility\(\)](#), G\_UTIL\_GCAPS\_IPREQUEST or G\_UTIL\_IPREQUEST to provide a list of all [Galil](#) controllers requesting IP addresses via BOOT-P or DHCP.

#### Returns

String containing hardware requesting IP addresses.

DMC4000, 291, 00:50:4C:20:01:23, LAN, 10.1.3.10 RIO47000, 37290, 00:50:4C:28:91:AA, Static, 192.168.0.41

#### Exceptions

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 449 of file GclibJava.java.

### 12.9.3.15 GListServers()

String GListServers ( ) throws [GclibJavaException](#) [inline]

Retrieves a list of gcaps servers that are advertising themselves on the local network.

#### Returns

A list of available gcaps server names.

#### Exceptions

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 568 of file GclibJava.java.



**12.9.3.16 GMessage()**

String GMessage ( ) throws [GclibJavaException](#) [inline]

Provides access to unsolicited messages from the controller.

To use this function, -s MG must be used in the [GOpen\(\)](#) address string to subscribe to messages. Unsolicited bytes must be flagged by the high-bit setting, CW 1. The driver will automatically set this when subscribing to messages. The user should not overwrite this setting.

Unsolicited messages are data generated by the controller that are not in response to a command, a data record, or an interrupt.

[GMessage\(\)](#) will block until a message is received, or the function times out.

Messages are unframed byte streams. There is no guarantee that the user will get complete messages or single messages in a call to [GMessage\(\)](#).

**Returns**

the message received.

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 235 of file GclibJava.java.

**12.9.3.17 GOpen()**

void GOpen (

String address ) throws [GclibJavaException](#) [inline]

Open a connection to a [Galil](#) Controller.

**Parameters**

<i>address</i>	address string. See <a href="#">gclib GOpen()</a>
----------------	---

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 250 of file GclibJava.java.

References [GclibJava.GClose\(\)](#).

**12.9.3.18 GProgramDownload() [1/2]**

void GProgramDownload (

String program ) throws [GclibJavaException](#) [inline]

Overload of GProgramDownload to use default preprocessor options.

**Parameters**

<i>program</i>	Program for download.
----------------	-----------------------

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 282 of file GclibJava.java.  
References GclibJava.GProgramDownload().

### 12.9.3.19 GProgramDownload() [2/2]

```
void GProgramDownload (
    String program,
    String preprocessor ) throws GclibJavaException [inline]
```

Downloads a program to the controller's program buffer.

#### Parameters

<i>program</i>	Program for download.
<i>preprocessor</i>	Options string for preprocessing the program before sending it to the controller.

#### Exceptions

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 270 of file GclibJava.java.  
Referenced by GclibJava.GProgramDownload().

### 12.9.3.20 GProgramDownloadFile() [1/2]

```
void GProgramDownloadFile (
    String filePath ) throws GclibJavaException [inline]
```

Overload of GProgramDownloadFile to use default preprocessor options.

#### Parameters

<i>filePath</i>	String containing the path to the program file.
-----------------	---

#### Exceptions

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 476 of file GclibJava.java.  
References GclibJava.GProgramDownloadFile().

### 12.9.3.21 GProgramDownloadFile() [2/2]

```
void GProgramDownloadFile (
    String filePath,
    String preprocessor ) throws GclibJavaException [inline]
```

Program download from file.

#### Parameters

<i>filePath</i>	String containing the path to the program file.
<i>preprocessor</i>	Options string for preprocessing the program before sending it to the controller.

## Exceptions

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 464 of file GclibJava.java.  
Referenced by GclibJava.GProgramDownloadFile().

**12.9.3.22 GProgramUpload()**

String GProgramUpload ( ) throws [GclibJavaException](#) [inline]  
Uploads a program from the controller's program buffer.

## Returns

The uploaded program.

## Exceptions

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 294 of file GclibJava.java.

**12.9.3.23 GProgramUploadFile()**

void GProgramUploadFile (   
    String *filePath* ) throws [GclibJavaException](#) [inline]  
Program upload to file.

## Parameters

<i>filePath</i>	String containing the path to the program file, file will be overwritten if it exists.
-----------------	--

## Exceptions

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 489 of file GclibJava.java.

**12.9.3.24 GPublishServer()**

void GPublishServer (   
    String *server\_name*,   
    int *publish*,   
    int *save* ) throws [GclibJavaException](#) [inline]  
Publishes or removes local gcaps server from the network

## Parameters

<i>server_name</i>	Name to publish server under.
<i>publish</i>	True=publish server, False=remove server.
<i>save</i>	Save this configuration for future server reboots.

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 583 of file GclibJava.java.

**12.9.3.25 GRemoteConnections()**

String GRemoteConnections ( ) throws [GclibJavaException](#) [inline]

Returns a list of IP Addresses that currently have an open connection to your hardware.

**Returns**

a list of IP Addresses that currently have an open connection to your hardware.

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 595 of file GclibJava.java.

**12.9.3.26 GServerStatus()**

String GServerStatus ( ) throws [GclibJavaException](#) [inline]

Retrieves the name of your local gcaps server and whether or not it is currently published Retrieves a list of gcaps servers that are advertising themselves on the local network.

**Returns**

A string in the form "<server\_name>, <isPublished>"

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 555 of file GclibJava.java.

**12.9.3.27 GSetServer()**

void GSetServer (   
 String server\_name ) throws [GclibJavaException](#) [inline]

Connects gclib to a new gcaps server

**Parameters**

<i>server_name</i>	Name to publish server under.
--------------------	-------------------------------

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 542 of file GclibJava.java.

**12.9.3.28 GSleep()**

```
void GSleep (
    int timeout_ms ) [inline]
```

Uses [GUtility\(\)](#) and G\_UTIL\_SLEEP to provide a blocking sleep call which can be useful for timing-based chores. In [GclibJava](#), this is primarily a debugging call.

**Parameters**

<i>timeout_ms</i>	Sleep time in milliseconds.
-------------------	-----------------------------

Definition at line 502 of file GclibJava.java.

**12.9.3.29 GTimeout()**

```
void GTimeout (
    short timeout_ms ) throws GclibJavaException [inline]
```

Uses [GUtility\(\)](#) and G\_UTIL\_TIMEOUT\_OVERRIDE to set the library timeout.

**Parameters**

<i>timeout_ms</i>	The value to be used for the timeout. Use -1 to set the timeout back to the initial <a href="#">GOpen()</a> value, -timeout.
-------------------	--

**Exceptions**

<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 515 of file GclibJava.java.  
Referenced by gclib.py.timeout().

**12.9.3.30 GVersion()**

```
String GVersion ( ) throws GclibJavaException [inline]
```

Uses [GUtility\(\)](#), G\_UTIL\_VERSION and G\_UTIL\_GCAPS\_VERSION to provide the library and gcaps version numbers.

**Returns**

A String containing the version, e.g. 189.224.370 1.0.0.125

**Exceptions**

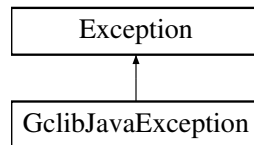
<a href="#">GclibJavaException</a>	If an error is generated by gclib.
------------------------------------	------------------------------------

Definition at line 529 of file GclibJava.java.  
The documentation for this class was generated from the following file:

- [GclibJava.java](#)

**12.10 GclibJavaException Class Reference**

Inheritance diagram for GclibJavaException:



## Public Member Functions

- **GclibJavaException** (int errorCode, String [message](#))
- int **getErrorCode** ()

### 12.10.1 Detailed Description

Definition at line 6 of file `GclibJavaException.java`.

The documentation for this class was generated from the following file:

- [GclibJavaException.java](#)

## 12.11 GclibTest Class Reference

### Static Public Member Functions

- static void **main** (String[] args)

#### 12.11.1 Detailed Description

A test of the Java gclib wrapper.

Definition at line 13 of file `GclibTest.java`.

#### 12.11.2 Member Function Documentation

##### 12.11.2.1 main()

```
static void main (
    String[] args ) [inline], [static]
```

##### Parameters

<i>args</i>	the command line arguments
-------------	----------------------------

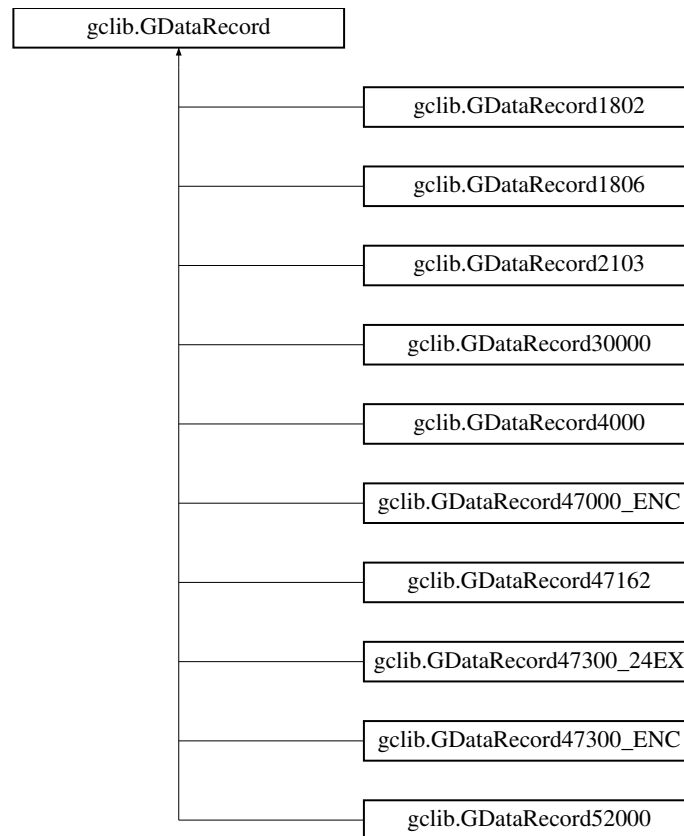
Definition at line 18 of file `GclibTest.java`.

The documentation for this class was generated from the following file:

- [GclibTest.java](#)

## 12.12 gclib.GDataRecord Interface Reference

Inheritance diagram for `gclib.GDataRecord`:



## Public Member Functions

- `byte[] byte_array ()`

*Returns the data record as a byte array and allows for access to individual bytes.*

### 12.12.1 Detailed Description

Definition at line 897 of file `gclib.cs`.

The documentation for this interface was generated from the following file:

- [gclib.cs](#)

## 12.13 GDataRecord Union Reference

Data record union, containing all structs and a generic byte array accessor.

```
#include <gclib_record.h>
```

### Data Fields

- struct [GDataRecord4000](#) `dmc4000`  
*The DMC-4000 data record.*
- struct [GDataRecord4000](#) `dmc4103`  
*The DMC-4103 data record.*
- struct [GDataRecord4000](#) `dmc50000`  
*The DMC-50000 data record.*
- struct [GDataRecord52000](#) `dmc52000`  
*The DMC-52000 data record.*
- struct [GDataRecord30000](#) `dmc30000`

*The DMC-30000 data record.*

- struct [GDataRecord2103 dmc2103](#)

*The DMC-21x3 data record.*

- struct [GDataRecord1806 dmc1806](#)

*The DMC-1806 data record.*

- struct [GDataRecord1802 dmc1802](#)

*The DMC-1802 data record.*

- struct [GDataRecord47000\\_ENC rio47000](#)

*The RIO-471xx & 472xx data record, including encoder support.*

- struct [GDataRecord47300\\_ENC rio47300](#)

*The RIO 473xx data record, including encoder support.*

- struct [GDataRecord47300\\_24EX rio47300\\_24ex](#)

*The RIO 473xx data record, with 24EXOUT/24EXIN support.*

- struct [GDataRecord47162 rio47162](#)

*The RIO 47162 data record.*

- unsigned char [byte\\_array \[GALILDATARECORDMAXLENGTH\]](#)

*Generic byte array for offsets.*

### 12.13.1 Detailed Description

Data record union, containing all structs and a generic byte array accessor.

Named structs can be used to access typed data by name. Offsets into the data record can also be used by referencing the member `byte_array`.

```
//Getting the sample counter for the DMC-4000.
cout << data_record->dmc4000.sample_number << '\n'; //access by 4000 product
cout << * ((unsigned short *) (data_record->byte_array + 4)) << '\n'; //access by pointer arithmetic
```

Definition at line 1082 of file `gclib_record.h`.

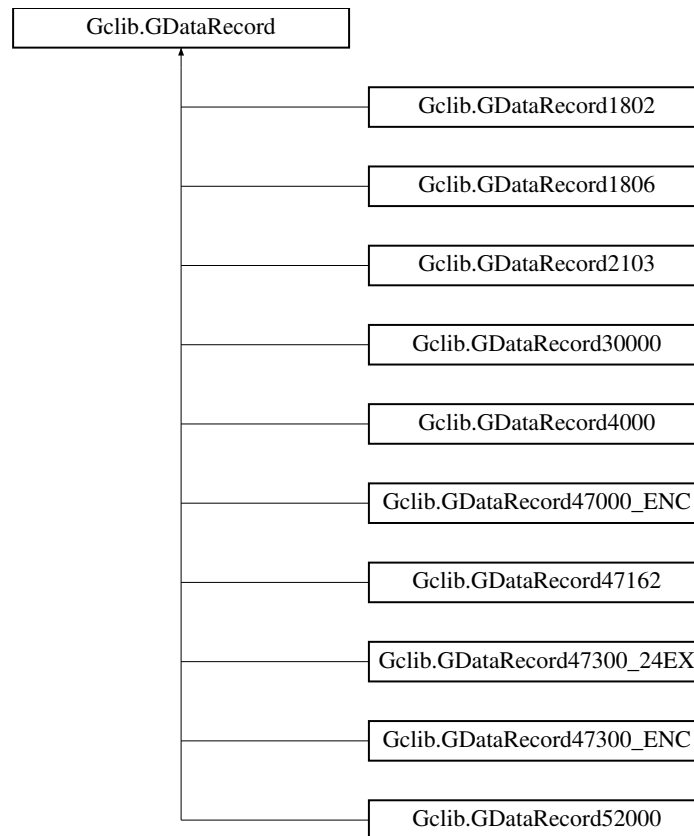
The documentation for this union was generated from the following file:

- [gclib\\_record.h](#)

## 12.14 Gclib.GDataRecord Interface Reference

Inheritance diagram for `Gclib.GDataRecord`:





## Public Member Functions

- `byte_array ()`

### 12.14.1 Detailed Description

Definition at line 467 of file `gclib.vb`.

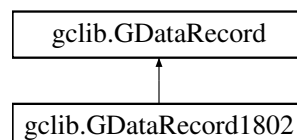
The documentation for this interface was generated from the following file:

- [gclib.vb](#)

## 12.15 gclib.GDataRecord1802 Struct Reference

Data record struct for DMC-1802 controllers.

Inheritance diagram for `gclib.GDataRecord1802`:



## Public Member Functions

- `byte[] byte_array ()`

*Returns the data record as a byte array and allows for access to individual bytes.*

## Data Fields

- UW [sample\\_number](#)  
*sample number.*
- UB [input\\_bank\\_0](#)  
*general input bank 0 (inputs 1-8).*
- UB [input\\_bank\\_1](#)  
*general input bank 1 (inputs 9-16).*
- UB [input\\_bank\\_2](#)  
*general input bank 2 (inputs 17-24).*
- UB [input\\_bank\\_3](#)  
*general input bank 3 (inputs 25-32).*
- UB [input\\_bank\\_4](#)  
*general input bank 4 (inputs 33-40).*
- UB [input\\_bank\\_5](#)  
*general input bank 5 (inputs 41-48).*
- UB [input\\_bank\\_6](#)  
*general input bank 6 (inputs 49-56).*
- UB [input\\_bank\\_7](#)  
*general input bank 7 (inputs 57-64).*
- UB [input\\_bank\\_8](#)  
*general input bank 8 (inputs 65-72).*
- UB [input\\_bank\\_9](#)  
*general input bank 9 (inputs 73-80).*
- UB [output\\_bank\\_0](#)  
*general output bank 0 (outputs 1-8).*
- UB [output\\_bank\\_1](#)  
*general output bank 1 (outputs 9-16).*
- UB [output\\_bank\\_2](#)  
*general output bank 2 (outputs 17-24).*
- UB [output\\_bank\\_3](#)  
*general output bank 3 (outputs 25-32).*
- UB [output\\_bank\\_4](#)  
*general output bank 4 (outputs 33-40).*
- UB [output\\_bank\\_5](#)  
*general output bank 5 (outputs 41-48).*
- UB [output\\_bank\\_6](#)  
*general output bank 6 (outputs 49-56).*
- UB [output\\_bank\\_7](#)  
*general output bank 7 (outputs 57-64).*
- UB [output\\_bank\\_8](#)  
*general output bank 8 (outputs 65-72).*
- UB [output\\_bank\\_9](#)  
*general output bank 9 (outputs 73-80).*
- UB [error\\_code](#)  
*error code.*
- UB [general\\_status](#)  
*general status*
- UW [s\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)

- coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*
- UW [t\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for T plane.*
- UW [t\\_plane\\_move\\_status](#)  
*Coordinated move status for T plane.*
- SL [t\\_distance](#)  
*distance traveled in coordinated move for T plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)  
*A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SW [axis\\_a\\_torque](#)  
*A axis torque.*
- UB [axis\\_a\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_a\\_reserved\\_1](#)  
*Reserved.*
- UW [axis\\_b\\_status](#)  
*B axis status.*
- UB [axis\\_b\\_switches](#)  
*B axis switches.*
- UB [axis\\_b\\_stop\\_code](#)  
*B axis stop code.*
- SL [axis\\_b\\_reference\\_position](#)  
*B axis reference position.*
- SL [axis\\_b\\_motor\\_position](#)  
*B axis motor position.*
- SL [axis\\_b\\_position\\_error](#)  
*B axis position error.*
- SL [axis\\_b\\_aux\\_position](#)  
*B axis auxiliary position.*
- SL [axis\\_b\\_velocity](#)  
*B axis velocity.*
- SW [axis\\_b\\_torque](#)  
*B axis torque.*
- UB [axis\\_b\\_reserved\\_0](#)  
*Reserved.*

- UB [axis\\_b\\_reserved\\_1](#)  
*Reserved.*
- UW [axis\\_c\\_status](#)  
*C axis status.*
- UB [axis\\_c\\_switches](#)  
*C axis switches.*
- UB [axis\\_c\\_stop\\_code](#)  
*C axis stop code.*
- SL [axis\\_c\\_reference\\_position](#)  
*C axis reference position.*
- SL [axis\\_c\\_motor\\_position](#)  
*C axis motor position.*
- SL [axis\\_c\\_position\\_error](#)  
*C axis position error.*
- SL [axis\\_c\\_aux\\_position](#)  
*C axis auxiliary position.*
- SL [axis\\_c\\_velocity](#)  
*C axis velocity.*
- SW [axis\\_c\\_torque](#)  
*C axis torque.*
- UB [axis\\_c\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_c\\_reserved\\_1](#)  
*Reserved.*
- UW [axis\\_d\\_status](#)  
*D axis status.*
- UB [axis\\_d\\_switches](#)  
*D axis switches.*
- UB [axis\\_d\\_stop\\_code](#)  
*D axis stop code.*
- SL [axis\\_d\\_reference\\_position](#)  
*D axis reference position.*
- SL [axis\\_d\\_motor\\_position](#)  
*D axis motor position.*
- SL [axis\\_d\\_position\\_error](#)  
*D axis position error.*
- SL [axis\\_d\\_aux\\_position](#)  
*D axis auxiliary position.*
- SL [axis\\_d\\_velocity](#)  
*D axis velocity.*
- SW [axis\\_d\\_torque](#)  
*D axis torque.*
- UB [axis\\_d\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_d\\_reserved\\_1](#)  
*Reserved.*

### 12.15.1 Detailed Description

Data record struct for DMC-1802 controllers.

The 18x2 Data record is the Same as 2103 except the following.

1. No header bytes. Software removes it from QR.
2. No analog in axis data.

Definition at line 1615 of file gclib.cs.

The documentation for this struct was generated from the following file:

- [gclib.cs](#)

## 12.16 GDataRecord1802 Struct Reference

```
#include <gclib_record.h>
```

### Data Fields

- UW [sample\\_number](#)  
*sample number.*
- UB [input\\_bank\\_0](#)  
*general input bank 0 (inputs 1-8).*
- UB [input\\_bank\\_1](#)  
*general input bank 1 (inputs 9-16).*
- UB [input\\_bank\\_2](#)  
*general input bank 2 (inputs 17-24).*
- UB [input\\_bank\\_3](#)  
*general input bank 3 (inputs 25-32).*
- UB [input\\_bank\\_4](#)  
*general input bank 4 (inputs 33-40).*
- UB [input\\_bank\\_5](#)  
*general input bank 5 (inputs 41-48).*
- UB [input\\_bank\\_6](#)  
*general input bank 6 (inputs 49-56).*
- UB [input\\_bank\\_7](#)  
*general input bank 7 (inputs 57-64).*
- UB [input\\_bank\\_8](#)  
*general input bank 8 (inputs 65-72).*
- UB [input\\_bank\\_9](#)  
*general input bank 9 (inputs 73-80).*
- UB [output\\_bank\\_0](#)  
*general output bank 0 (outputs 1-8).*
- UB [output\\_bank\\_1](#)  
*general output bank 1 (outputs 9-16).*
- UB [output\\_bank\\_2](#)  
*general output bank 2 (outputs 17-24).*
- UB [output\\_bank\\_3](#)  
*general output bank 3 (outputs 25-32).*
- UB [output\\_bank\\_4](#)  
*general output bank 4 (outputs 33-40).*
- UB [output\\_bank\\_5](#)  
*general output bank 5 (outputs 41-48).*

- UB [output\\_bank\\_6](#)  
*general output bank 6 (outputs 49-56).*
- UB [output\\_bank\\_7](#)  
*general output bank 7 (outputs 57-64).*
- UB [output\\_bank\\_8](#)  
*general output bank 8 (outputs 65-72).*
- UB [output\\_bank\\_9](#)  
*general output bank 9 (outputs 73-80).*
- UB [error\\_code](#)  
*error code.*
- UB [general\\_status](#)  
*general status*
- UW [s\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)  
*coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*
- UW [t\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for T plane.*
- UW [t\\_plane\\_move\\_status](#)  
*Coordinated move status for T plane.*
- SL [t\\_distance](#)  
*distance traveled in coordinated move for T plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)  
*A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SW [axis\\_a\\_torque](#)  
*A axis torque.*
- UB [axis\\_a\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_a\\_reserved\\_1](#)  
*Reserved.*
- UW [axis\\_b\\_status](#)  
*B axis status.*
- UB [axis\\_b\\_switches](#)  
*B axis switches.*
- UB [axis\\_b\\_stop\\_code](#)

- B axis stop code.*
- SL [axis\\_b\\_reference\\_position](#)  
*B axis reference position.*
- SL [axis\\_b\\_motor\\_position](#)  
*B axis motor position.*
- SL [axis\\_b\\_position\\_error](#)  
*B axis position error.*
- SL [axis\\_b\\_aux\\_position](#)  
*B axis auxiliary position.*
- SL [axis\\_b\\_velocity](#)  
*B axis velocity.*
- SW [axis\\_b\\_torque](#)  
*B axis torque.*
- UB [axis\\_b\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_b\\_reserved\\_1](#)  
*Reserved.*
- UW [axis\\_c\\_status](#)  
*C axis status.*
- UB [axis\\_c\\_switches](#)  
*C axis switches.*
- UB [axis\\_c\\_stop\\_code](#)  
*C axis stop code.*
- SL [axis\\_c\\_reference\\_position](#)  
*C axis reference position.*
- SL [axis\\_c\\_motor\\_position](#)  
*C axis motor position.*
- SL [axis\\_c\\_position\\_error](#)  
*C axis position error.*
- SL [axis\\_c\\_aux\\_position](#)  
*C axis auxiliary position.*
- SL [axis\\_c\\_velocity](#)  
*C axis velocity.*
- SW [axis\\_c\\_torque](#)  
*C axis torque.*
- UB [axis\\_c\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_c\\_reserved\\_1](#)  
*Reserved.*
- UW [axis\\_d\\_status](#)  
*D axis status.*
- UB [axis\\_d\\_switches](#)  
*D axis switches.*
- UB [axis\\_d\\_stop\\_code](#)  
*D axis stop code.*
- SL [axis\\_d\\_reference\\_position](#)  
*D axis reference position.*
- SL [axis\\_d\\_motor\\_position](#)  
*D axis motor position.*
- SL [axis\\_d\\_position\\_error](#)  
*D axis position error.*

- SL [axis\\_d\\_aux\\_position](#)  
*D axis auxiliary position.*
- SL [axis\\_d\\_velocity](#)  
*D axis velocity.*
- SW [axis\\_d\\_torque](#)  
*D axis torque.*
- UB [axis\\_d\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_d\\_reserved\\_1](#)  
*Reserved.*

### 12.16.1 Detailed Description

Data record struct for DMC-1802 controllers.

The 18x2 Data record is the Same as 2103 except the following.

1. No header bytes. Software removes it from QR.
2. No analog in axis data.

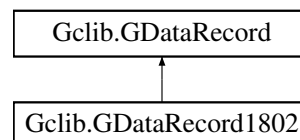
Definition at line 727 of file `gclib_record.h`.

The documentation for this struct was generated from the following file:

- [gclib\\_record.h](#)

## 12.17 Gclib.GDataRecord1802 Struct Reference

Inheritance diagram for `Gclib.GDataRecord1802`:



### Public Member Functions

- general input [bank](#) (inputs 1-8). *\*/public UB input\_bank\_1*  
*/\*03*
- general input [bank](#) (inputs 9-16). *\*/public UB input\_bank\_2*  
*/\*04*
- general input [bank](#) (inputs 17-24). *\*/public UB input\_bank\_3*  
*/\*05*
- general input [bank](#) (inputs 25-32). *\*/public UB input\_bank\_4*  
*/\*06*
- general input [bank](#) (inputs 33-40). *\*/public UB input\_bank\_5*  
*/\*07*
- general input [bank](#) (inputs 41-48). *\*/public UB input\_bank\_6*  
*/\*08*
- general input [bank](#) (inputs 49-56). *\*/public UB input\_bank\_7*  
*/\*09*
- general input [bank](#) (inputs 57-64). *\*/public UB input\_bank\_8*  
*/\*10*
- general input [bank](#) (inputs 65-72). *\*/public UB input\_bank\_9*



- */\* 11*  
general input [bank](#) (inputs 73-80). *\*/public UB output\_bank\_0*
- */\* 12*  
general output [bank](#) (outputs 1-8). *\*/public UB output\_bank\_1*
- */\* 13*  
general output [bank](#) (outputs 9-16). *\*/public UB output\_bank\_2*
- */\* 14*  
general output [bank](#) (outputs 17-24). *\*/public UB output\_bank\_3*
- */\* 15*  
general output [bank](#) (outputs 25-32). *\*/public UB output\_bank\_4*
- */\* 16*  
general output [bank](#) (outputs 33-40). *\*/public UB output\_bank\_5*
- */\* 17*  
general output [bank](#) (outputs 41-48). *\*/public UB output\_bank\_6*
- */\* 18*  
general output [bank](#) (outputs 49-56). *\*/public UB output\_bank\_7*
- */\* 19*  
general output [bank](#) (outputs 57-64). *\*/public UB output\_bank\_8*
- */\* 20*  
general output [bank](#) (outputs 65-72). *\*/public UB output\_bank\_9*
- */\* 21*  
general output [bank](#) (outputs 73-80). *\*/public UB error\_code*
- */\* 22*  
Reserved.\* [byte\\_array](#) ()

## Data Fields

- UW [sample\\_number](#)  
*/\*00-01*
- sample number.\*UB [input\\_bank\\_0](#)  
*/\*02*
- error code.\*UB [general\\_status](#)  
*/\*23*
- general status \*UW [s\\_plane\\_segment\\_count](#)  
*/\*24-25*
- segment count of coordinated move for S plane.\*UW [s\\_plane\\_move\\_status](#)  
*/\*26-27*
- coordinated move status for S plane.\*SL [s\\_distance](#)  
*/\*28-31*
- distance traveled in coordinated move for S plane.\*UW [t\\_plane\\_segment\\_count](#)  
*/\*32-33*
- segment count of coordinated move for T plane.\*UW [t\\_plane\\_move\\_status](#)  
*/\*34-35*
- Coordinated move status for T plane.\*SL [t\\_distance](#)  
*/\*36-39*
- distance traveled in coordinated move for T plane.\*UW [axis\\_a\\_status](#)  
*/\*40-41*
- A axis status.\*UB [axis\\_a\\_switches](#)  
*/\*42*
- A axis switches.\*UB [axis\\_a\\_stop\\_code](#)  
*/\*43*

- A axis stop code.\*SL [axis\\_a\\_reference\\_position](#)  
*/\*44-47*
- A axis reference position.\*SL [axis\\_a\\_motor\\_position](#)  
*/\*48-51*
- A axis motor position.\*SL [axis\\_a\\_position\\_error](#)  
*/\*52-55*
- A axis position error.\*SL [axis\\_a\\_aux\\_position](#)  
*/\*56-59*
- A axis auxiliary position.\*SL [axis\\_a\\_velocity](#)  
*/\*60-63*
- A axis velocity.\*SW [axis\\_a\\_torque](#)  
*/\*64-65*
- A axis torque.\*UB [axis\\_a\\_reserved\\_0](#)  
*/\*66*
- Reserved.\*UB [axis\\_a\\_reserved\\_1](#)  
*/\*67*
- Reserved.\*UW [axis\\_b\\_status](#)  
*/\*68-69*
- B axis status.\*UB [axis\\_b\\_switches](#)  
*/\*70*
- B axis switches.\*UB [axis\\_b\\_stop\\_code](#)  
*/\*71*
- B axis stop code.\*SL [axis\\_b\\_reference\\_position](#)  
*/\*72-75*
- B axis reference position.\*SL [axis\\_b\\_motor\\_position](#)  
*/\*76-79*
- B axis motor position.\*SL [axis\\_b\\_position\\_error](#)  
*/\*80-83*
- B axis position error.\*SL [axis\\_b\\_aux\\_position](#)  
*/\*84-87*
- B axis auxiliary position.\*SL [axis\\_b\\_velocity](#)  
*/\*88-91*
- B axis velocity.\*SW [axis\\_b\\_torque](#)  
*/\*92-93*
- B axis torque.\*UB [axis\\_b\\_reserved\\_0](#)  
*/\*94*
- Reserved.\*UB [axis\\_b\\_reserved\\_1](#)  
*/\*95*
- Reserved.\*UW [axis\\_c\\_status](#)  
*/\*96-97*
- C axis status.\*UB [axis\\_c\\_switches](#)  
*/\*98*
- C axis switches.\*UB [axis\\_c\\_stop\\_code](#)  
*/\*99*
- C axis stop code.\*SL [axis\\_c\\_reference\\_position](#)  
*/\*100-103*
- C axis reference position.\*SL [axis\\_c\\_motor\\_position](#)  
*/\*104-107*
- C axis motor position.\*SL [axis\\_c\\_position\\_error](#)  
*/\*108-111*
- C axis position error.\*SL [axis\\_c\\_aux\\_position](#)

- /\* 112-115*
- C axis auxiliary position.\*SL [axis\\_c\\_velocity](#)
- /\* 116-119*
- C axis velocity.\*SW [axis\\_c\\_torque](#)
- /\* 120-121*
- C axis torque.\*UB [axis\\_c\\_reserved\\_0](#)
- /\* 122*
- Reserved.\*UB [axis\\_c\\_reserved\\_1](#)
- /\* 123*
- Reserved.\*UW [axis\\_d\\_status](#)
- /\* 124-125*
- D axis status.\*UB [axis\\_d\\_switches](#)
- /\* 126*
- D axis switches.\*UB [axis\\_d\\_stop\\_code](#)
- /\* 127*
- D axis stop code.\*SL [axis\\_d\\_reference\\_position](#)
- /\* 128-131*
- D axis reference position.\*SL [axis\\_d\\_motor\\_position](#)
- /\* 132-135*
- D axis motor position.\*SL [axis\\_d\\_position\\_error](#)
- /\* 136-139*
- D axis position error.\*SL [axis\\_d\\_aux\\_position](#)
- /\* 140-143*
- D axis auxiliary position.\*SL [axis\\_d\\_velocity](#)
- /\* 144-147*
- D axis velocity.\*SW [axis\\_d\\_torque](#)
- /\* 148-149*
- D axis torque.\*UB [axis\\_d\\_reserved\\_0](#)
- /\* 150*
- Reserved.\*UB [axis\\_d\\_reserved\\_1](#)
- /\* 151*

### 12.17.1 Detailed Description

Definition at line 1076 of file gclib.vb.

### 12.17.2 Member Function Documentation

#### 12.17.2.1 `byte_array()`

Reserved.\* `byte_array ( )`

Implements [GDataRecord.byte\\_array](#)

Implements [Gclib.GDataRecord](#).

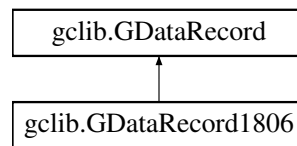
The documentation for this struct was generated from the following file:

- [gclib.vb](#)

## 12.18 gclib.GDataRecord1806 Struct Reference

Data record struct for DMC-1806 controller.

Inheritance diagram for gclib.GDataRecord1806:



### Public Member Functions

- `byte[] byte_array ()`  
*Returns the data record as a byte array and allows for access to individual bytes.*

### Data Fields

- UW `sample_number`  
*sample number.*
- UB `input_bank_0`  
*general input bank 0 (inputs 1-8).*
- UB `input_bank_1`  
*general input bank 1 (inputs 9-16).*
- UB `input_bank_2`  
*general input bank 2 (inputs 17-24).*
- UB `input_bank_3`  
*general input bank 3 (inputs 25-32).*
- UB `input_bank_4`  
*general input bank 4 (inputs 33-40).*
- UB `input_bank_5`  
*general input bank 5 (inputs 41-48).*
- UB `input_bank_6`  
*general input bank 6 (inputs 49-56).*
- UB `input_bank_7`  
*general input bank 7 (inputs 57-64).*
- UB `input_bank_8`  
*general input bank 8 (inputs 65-72).*
- UB `input_bank_9`  
*general input bank 9 (inputs 73-80).*
- UB `output_bank_0`  
*general output bank 0 (outputs 1-8).*
- UB `output_bank_1`  
*general output bank 1 (outputs 9-16).*
- UB `output_bank_2`  
*general output bank 2 (outputs 17-24).*
- UB `output_bank_3`  
*general output bank 3 (outputs 25-32).*
- UB `output_bank_4`  
*general output bank 4 (outputs 33-40).*
- UB `output_bank_5`  
*general output bank 5 (outputs 41-48).*

- UB [output\\_bank\\_6](#)  
*general output bank 6 (outputs 49-56).*
- UB [output\\_bank\\_7](#)  
*general output bank 7 (outputs 57-64).*
- UB [output\\_bank\\_8](#)  
*general output bank 8 (outputs 65-72).*
- UB [output\\_bank\\_9](#)  
*general output bank 9 (outputs 73-80).*
- SW [reserved\\_0](#)  
*Reserved.*
- SW [reserved\\_2](#)  
*Reserved.*
- SW [reserved\\_4](#)  
*Reserved.*
- SW [reserved\\_6](#)  
*Reserved.*
- SW [reserved\\_8](#)  
*Reserved.*
- SW [reserved\\_10](#)  
*Reserved.*
- SW [reserved\\_12](#)  
*Reserved.*
- SW [reserved\\_14](#)  
*Reserved.*
- UB [reserved\\_16](#)  
*Reserved.*
- UB [reserved\\_17](#)  
*Reserved.*
- UB [reserved\\_18](#)  
*Reserved.*
- UB [reserved\\_19](#)  
*Reserved.*
- UB [reserved\\_20](#)  
*Reserved.*
- UB [reserved\\_21](#)  
*Reserved.*
- UB [reserved\\_22](#)  
*Reserved.*
- UB [reserved\\_23](#)  
*Reserved.*
- UB [error\\_code](#)  
*error code.*
- UB [thread\\_status](#)  
*thread status.*
- UL [reserved\\_24](#)  
*Reserved.*
- UL [contour\\_segment\\_count](#)  
*Segment Count for Contour Mode.*
- UW [contour\\_buffer\\_available](#)  
*Buffer space remaining, Contour Mode.*
- UW [s\\_plane\\_segment\\_count](#)

- segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)  
*coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*
- UW [s\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, S Plane.*
- UW [t\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for T plane.*
- UW [t\\_plane\\_move\\_status](#)  
*Coordinated move status for T plane.*
- SL [t\\_distance](#)  
*distance traveled in coordinated move for T plane.*
- UW [t\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, T Plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)  
*A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SL [axis\\_a\\_torque](#)  
*A axis torque.*
- UW [axis\\_a\\_analog\\_in](#)  
*A axis analog input.*
- UB [axis\\_a\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_a\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_a\\_variable](#)  
*A User-defined variable (ZA).*
- UW [axis\\_b\\_status](#)  
*B axis status.*
- UB [axis\\_b\\_switches](#)  
*B axis switches.*
- UB [axis\\_b\\_stop\\_code](#)  
*B axis stop code.*
- SL [axis\\_b\\_reference\\_position](#)  
*B axis reference position.*
- SL [axis\\_b\\_motor\\_position](#)  
*B axis motor position.*

- SL [axis\\_b\\_position\\_error](#)  
*B axis position error.*
- SL [axis\\_b\\_aux\\_position](#)  
*B axis auxiliary position.*
- SL [axis\\_b\\_velocity](#)  
*B axis velocity.*
- SL [axis\\_b\\_torque](#)  
*B axis torque.*
- UW [axis\\_b\\_analog\\_in](#)  
*B axis analog input.*
- UB [axis\\_b\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_b\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_b\\_variable](#)  
*B User-defined variable (ZA).*
- UW [axis\\_c\\_status](#)  
*C axis status.*
- UB [axis\\_c\\_switches](#)  
*C axis switches.*
- UB [axis\\_c\\_stop\\_code](#)  
*C axis stop code.*
- SL [axis\\_c\\_reference\\_position](#)  
*C axis reference position.*
- SL [axis\\_c\\_motor\\_position](#)  
*C axis motor position.*
- SL [axis\\_c\\_position\\_error](#)  
*C axis position error.*
- SL [axis\\_c\\_aux\\_position](#)  
*C axis auxiliary position.*
- SL [axis\\_c\\_velocity](#)  
*C axis velocity.*
- SL [axis\\_c\\_torque](#)  
*C axis torque.*
- UW [axis\\_c\\_analog\\_in](#)  
*C axis analog input.*
- UB [axis\\_c\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_c\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_c\\_variable](#)  
*C User-defined variable (ZA).*
- UW [axis\\_d\\_status](#)  
*D axis status.*
- UB [axis\\_d\\_switches](#)  
*D axis switches.*
- UB [axis\\_d\\_stop\\_code](#)  
*D axis stop code.*
- SL [axis\\_d\\_reference\\_position](#)  
*D axis reference position.*
- SL [axis\\_d\\_motor\\_position](#)

- D axis motor position.*
- SL [axis\\_d\\_position\\_error](#)  
*D axis position error.*
- SL [axis\\_d\\_aux\\_position](#)  
*D axis auxiliary position.*
- SL [axis\\_d\\_velocity](#)  
*D axis velocity.*
- SL [axis\\_d\\_torque](#)  
*D axis torque.*
- UW [axis\\_d\\_analog\\_in](#)  
*D axis analog input.*
- UB [axis\\_d\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_d\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_d\\_variable](#)  
*D User-defined variable (ZA).*
- UW [axis\\_e\\_status](#)  
*E axis status.*
- UB [axis\\_e\\_switches](#)  
*E axis switches.*
- UB [axis\\_e\\_stop\\_code](#)  
*E axis stop code.*
- SL [axis\\_e\\_reference\\_position](#)  
*E axis reference position.*
- SL [axis\\_e\\_motor\\_position](#)  
*E axis motor position.*
- SL [axis\\_e\\_position\\_error](#)  
*E axis position error.*
- SL [axis\\_e\\_aux\\_position](#)  
*E axis auxiliary position.*
- SL [axis\\_e\\_velocity](#)  
*E axis velocity.*
- SL [axis\\_e\\_torque](#)  
*E axis torque.*
- UW [axis\\_e\\_analog\\_in](#)  
*E axis analog input.*
- UB [axis\\_e\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_e\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_e\\_variable](#)  
*E User-defined variable (ZA).*
- UW [axis\\_f\\_status](#)  
*F axis status.*
- UB [axis\\_f\\_switches](#)  
*F axis switches.*
- UB [axis\\_f\\_stop\\_code](#)  
*F axis stop code.*
- SL [axis\\_f\\_reference\\_position](#)  
*F axis reference position.*



- SL [axis\\_f\\_motor\\_position](#)  
*F axis motor position.*
- SL [axis\\_f\\_position\\_error](#)  
*F axis position error.*
- SL [axis\\_f\\_aux\\_position](#)  
*F axis auxiliary position.*
- SL [axis\\_f\\_velocity](#)  
*F axis velocity.*
- SL [axis\\_f\\_torque](#)  
*F axis torque.*
- UW [axis\\_f\\_analog\\_in](#)  
*F axis analog input.*
- UB [axis\\_f\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_f\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_f\\_variable](#)  
*F User-defined variable (ZA).*
- UW [axis\\_g\\_status](#)  
*G axis status.*
- UB [axis\\_g\\_switches](#)  
*G axis switches.*
- UB [axis\\_g\\_stop\\_code](#)  
*G axis stop code.*
- SL [axis\\_g\\_reference\\_position](#)  
*G axis reference position.*
- SL [axis\\_g\\_motor\\_position](#)  
*G axis motor position.*
- SL [axis\\_g\\_position\\_error](#)  
*G axis position error.*
- SL [axis\\_g\\_aux\\_position](#)  
*G axis auxiliary position.*
- SL [axis\\_g\\_velocity](#)  
*G axis velocity.*
- SL [axis\\_g\\_torque](#)  
*G axis torque.*
- UW [axis\\_g\\_analog\\_in](#)  
*G axis analog input.*
- UB [axis\\_g\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_g\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_g\\_variable](#)  
*G User-defined variable (ZA).*
- UW [axis\\_h\\_status](#)  
*H axis status.*
- UB [axis\\_h\\_switches](#)  
*H axis switches.*
- UB [axis\\_h\\_stop\\_code](#)  
*H axis stop code.*
- SL [axis\\_h\\_reference\\_position](#)

- *H axis reference position.*
- SL [axis\\_h\\_motor\\_position](#)  
*H axis motor position.*
- SL [axis\\_h\\_position\\_error](#)  
*H axis position error.*
- SL [axis\\_h\\_aux\\_position](#)  
*H axis auxiliary position.*
- SL [axis\\_h\\_velocity](#)  
*H axis velocity.*
- SL [axis\\_h\\_torque](#)  
*H axis torque.*
- UW [axis\\_h\\_analog\\_in](#)  
*H axis analog input.*
- UB [axis\\_h\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_h\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_h\\_variable](#)  
*H User-defined variable (ZA).*

### 12.18.1 Detailed Description

Data record struct for DMC-1806 controller.

The 18x6 Data record is the same as 4000 except the following.

1. No header bytes. Firmware strips it in DR. Software removes it from QR.
2. No Ethernet status (bytes 42-49).
3. No amplifier status (bytes 52-55).
4. No axis-specific hall input status.

Definition at line 1294 of file gclib.cs.

The documentation for this struct was generated from the following file:

- [gclib.cs](#)

## 12.19 GDataRecord1806 Struct Reference

Data record struct for DMC-1806 controller.

```
#include <gclib_record.h>
```

### Data Fields

- UW [sample\\_number](#)  
*sample number.*
- UB [input\\_bank\\_0](#)  
*general input bank 0 (inputs 1-8).*
- UB [input\\_bank\\_1](#)  
*general input bank 1 (inputs 9-16).*
- UB [input\\_bank\\_2](#)  
*general input bank 2 (inputs 17-24).*
- UB [input\\_bank\\_3](#)  
*general input bank 3 (inputs 25-32).*

- UB [input\\_bank\\_4](#)  
*general input bank 4 (inputs 33-40).*
- UB [input\\_bank\\_5](#)  
*general input bank 5 (inputs 41-48).*
- UB [input\\_bank\\_6](#)  
*general input bank 6 (inputs 49-56).*
- UB [input\\_bank\\_7](#)  
*general input bank 7 (inputs 57-64).*
- UB [input\\_bank\\_8](#)  
*general input bank 8 (inputs 65-72).*
- UB [input\\_bank\\_9](#)  
*general input bank 9 (inputs 73-80).*
- UB [output\\_bank\\_0](#)  
*general output bank 0 (outputs 1-8).*
- UB [output\\_bank\\_1](#)  
*general output bank 1 (outputs 9-16).*
- UB [output\\_bank\\_2](#)  
*general output bank 2 (outputs 17-24).*
- UB [output\\_bank\\_3](#)  
*general output bank 3 (outputs 25-32).*
- UB [output\\_bank\\_4](#)  
*general output bank 4 (outputs 33-40).*
- UB [output\\_bank\\_5](#)  
*general output bank 5 (outputs 41-48).*
- UB [output\\_bank\\_6](#)  
*general output bank 6 (outputs 49-56).*
- UB [output\\_bank\\_7](#)  
*general output bank 7 (outputs 57-64).*
- UB [output\\_bank\\_8](#)  
*general output bank 8 (outputs 65-72).*
- UB [output\\_bank\\_9](#)  
*general output bank 9 (outputs 73-80).*
- SW [reserved\\_0](#)  
*Reserved.*
- SW [reserved\\_2](#)  
*Reserved.*
- SW [reserved\\_4](#)  
*Reserved.*
- SW [reserved\\_6](#)  
*Reserved.*
- SW [reserved\\_8](#)  
*Reserved.*
- SW [reserved\\_10](#)  
*Reserved.*
- SW [reserved\\_12](#)  
*Reserved.*
- SW [reserved\\_14](#)  
*Reserved.*
- UB [reserved\\_16](#)  
*Reserved.*
- UB [reserved\\_17](#)

- *Reserved.*
- UB [reserved\\_18](#)  
*Reserved.*
- UB [reserved\\_19](#)  
*Reserved.*
- UB [reserved\\_20](#)  
*Reserved.*
- UB [reserved\\_21](#)  
*Reserved.*
- UB [reserved\\_22](#)  
*Reserved.*
- UB [reserved\\_23](#)  
*Reserved.*
- UB [error\\_code](#)  
*error code.*
- UB [thread\\_status](#)  
*thread status.*
- UL [reserved\\_24](#)  
*Reserved.*
- UL [contour\\_segment\\_count](#)  
*Segment Count for Contour Mode.*
- UW [contour\\_buffer\\_available](#)  
*Buffer space remaining, Contour Mode.*
- UW [s\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)  
*coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*
- UW [s\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, S Plane.*
- UW [t\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for T plane.*
- UW [t\\_plane\\_move\\_status](#)  
*Coordinated move status for T plane.*
- SL [t\\_distance](#)  
*distance traveled in coordinated move for T plane.*
- UW [t\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, T Plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)  
*A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*

- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SL [axis\\_a\\_torque](#)  
*A axis torque.*
- UW [axis\\_a\\_analog\\_in](#)  
*A axis analog input.*
- UB [axis\\_a\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_a\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_a\\_variable](#)  
*A User-defined variable (ZA).*
- UW [axis\\_b\\_status](#)  
*B axis status.*
- UB [axis\\_b\\_switches](#)  
*B axis switches.*
- UB [axis\\_b\\_stop\\_code](#)  
*B axis stop code.*
- SL [axis\\_b\\_reference\\_position](#)  
*B axis reference position.*
- SL [axis\\_b\\_motor\\_position](#)  
*B axis motor position.*
- SL [axis\\_b\\_position\\_error](#)  
*B axis position error.*
- SL [axis\\_b\\_aux\\_position](#)  
*B axis auxiliary position.*
- SL [axis\\_b\\_velocity](#)  
*B axis velocity.*
- SL [axis\\_b\\_torque](#)  
*B axis torque.*
- UW [axis\\_b\\_analog\\_in](#)  
*B axis analog input.*
- UB [axis\\_b\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_b\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_b\\_variable](#)  
*B User-defined variable (ZA).*
- UW [axis\\_c\\_status](#)  
*C axis status.*
- UB [axis\\_c\\_switches](#)  
*C axis switches.*
- UB [axis\\_c\\_stop\\_code](#)  
*C axis stop code.*
- SL [axis\\_c\\_reference\\_position](#)  
*C axis reference position.*
- SL [axis\\_c\\_motor\\_position](#)  
*C axis motor position.*
- SL [axis\\_c\\_position\\_error](#)

- C axis position error.*
- SL [axis\\_c\\_aux\\_position](#)  
*C axis auxiliary position.*
- SL [axis\\_c\\_velocity](#)  
*C axis velocity.*
- SL [axis\\_c\\_torque](#)  
*C axis torque.*
- UW [axis\\_c\\_analog\\_in](#)  
*C axis analog input.*
- UB [axis\\_c\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_c\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_c\\_variable](#)  
*C User-defined variable (ZA).*
- UW [axis\\_d\\_status](#)  
*D axis status.*
- UB [axis\\_d\\_switches](#)  
*D axis switches.*
- UB [axis\\_d\\_stop\\_code](#)  
*D axis stop code.*
- SL [axis\\_d\\_reference\\_position](#)  
*D axis reference position.*
- SL [axis\\_d\\_motor\\_position](#)  
*D axis motor position.*
- SL [axis\\_d\\_position\\_error](#)  
*D axis position error.*
- SL [axis\\_d\\_aux\\_position](#)  
*D axis auxiliary position.*
- SL [axis\\_d\\_velocity](#)  
*D axis velocity.*
- SL [axis\\_d\\_torque](#)  
*D axis torque.*
- UW [axis\\_d\\_analog\\_in](#)  
*D axis analog input.*
- UB [axis\\_d\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_d\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_d\\_variable](#)  
*D User-defined variable (ZA).*
- UW [axis\\_e\\_status](#)  
*E axis status.*
- UB [axis\\_e\\_switches](#)  
*E axis switches.*
- UB [axis\\_e\\_stop\\_code](#)  
*E axis stop code.*
- SL [axis\\_e\\_reference\\_position](#)  
*E axis reference position.*
- SL [axis\\_e\\_motor\\_position](#)  
*E axis motor position.*

- SL [axis\\_e\\_position\\_error](#)  
*E axis position error.*
- SL [axis\\_e\\_aux\\_position](#)  
*E axis auxiliary position.*
- SL [axis\\_e\\_velocity](#)  
*E axis velocity.*
- SL [axis\\_e\\_torque](#)  
*E axis torque.*
- UW [axis\\_e\\_analog\\_in](#)  
*E axis analog input.*
- UB [axis\\_e\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_e\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_e\\_variable](#)  
*E User-defined variable (ZA).*
- UW [axis\\_f\\_status](#)  
*F axis status.*
- UB [axis\\_f\\_switches](#)  
*F axis switches.*
- UB [axis\\_f\\_stop\\_code](#)  
*F axis stop code.*
- SL [axis\\_f\\_reference\\_position](#)  
*F axis reference position.*
- SL [axis\\_f\\_motor\\_position](#)  
*F axis motor position.*
- SL [axis\\_f\\_position\\_error](#)  
*F axis position error.*
- SL [axis\\_f\\_aux\\_position](#)  
*F axis auxiliary position.*
- SL [axis\\_f\\_velocity](#)  
*F axis velocity.*
- SL [axis\\_f\\_torque](#)  
*F axis torque.*
- UW [axis\\_f\\_analog\\_in](#)  
*F axis analog input.*
- UB [axis\\_f\\_reserved\\_0](#)  
*Reserved.*
- UB [axis\\_f\\_reserved\\_1](#)  
*Reserved.*
- SL [axis\\_f\\_variable](#)  
*F User-defined variable (ZA).*
- UW [axis\\_g\\_status](#)  
*G axis status.*
- UB [axis\\_g\\_switches](#)  
*G axis switches.*
- UB [axis\\_g\\_stop\\_code](#)  
*G axis stop code.*
- SL [axis\\_g\\_reference\\_position](#)  
*G axis reference position.*
- SL [axis\\_g\\_motor\\_position](#)

- G axis motor position.*

  - SL [axis\\_g\\_position\\_error](#)

*G axis position error.*
  - SL [axis\\_g\\_aux\\_position](#)

*G axis auxiliary position.*
  - SL [axis\\_g\\_velocity](#)

*G axis velocity.*
  - SL [axis\\_g\\_torque](#)

*G axis torque.*
  - UW [axis\\_g\\_analog\\_in](#)

*G axis analog input.*
  - UB [axis\\_g\\_reserved\\_0](#)

*Reserved.*
  - UB [axis\\_g\\_reserved\\_1](#)

*Reserved.*
  - SL [axis\\_g\\_variable](#)

*G User-defined variable (ZA).*
- UW [axis\\_h\\_status](#)

*H axis status.*
- UB [axis\\_h\\_switches](#)

*H axis switches.*
- UB [axis\\_h\\_stop\\_code](#)

*H axis stop code.*
- SL [axis\\_h\\_reference\\_position](#)

*H axis reference position.*
- SL [axis\\_h\\_motor\\_position](#)

*H axis motor position.*
- SL [axis\\_h\\_position\\_error](#)

*H axis position error.*
- SL [axis\\_h\\_aux\\_position](#)

*H axis auxiliary position.*
- SL [axis\\_h\\_velocity](#)

*H axis velocity.*
- SL [axis\\_h\\_torque](#)

*H axis torque.*
- UW [axis\\_h\\_analog\\_in](#)

*H axis analog input.*
- UB [axis\\_h\\_reserved\\_0](#)

*Reserved.*
- UB [axis\\_h\\_reserved\\_1](#)

*Reserved.*
- SL [axis\\_h\\_variable](#)

*H User-defined variable (ZA).*



### 12.19.1 Detailed Description

Data record struct for DMC-1806 controller.

The 18x6 Data record is the same as 4000 except the following.

1. No header bytes. Firmware strips it in DR. Software removes it from QR.
2. No Ethernet status (bytes 42-49).
3. No amplifier status (bytes 52-55).
4. No axis-specific hall input status.

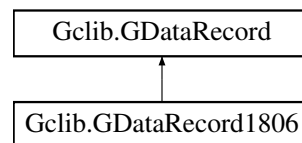
Definition at line 409 of file `gclib_record.h`.

The documentation for this struct was generated from the following file:

- [gclib\\_record.h](#)

## 12.20 Gclib.GDataRecord1806 Struct Reference

Inheritance diagram for Gclib.GDataRecord1806:



### Public Member Functions

- general input **bank** (inputs 1-8). `*/public UB input_bank_1`  
`/*03`
- general input **bank** (inputs 9-16). `*/public UB input_bank_2`  
`/*04`
- general input **bank** (inputs 17-24). `*/public UB input_bank_3`  
`/*05`
- general input **bank** (inputs 25-32). `*/public UB input_bank_4`  
`/*06`
- general input **bank** (inputs 33-40). `*/public UB input_bank_5`  
`/*07`
- general input **bank** (inputs 41-48). `*/public UB input_bank_6`  
`/*08`
- general input **bank** (inputs 49-56). `*/public UB input_bank_7`  
`/*09`
- general input **bank** (inputs 57-64). `*/public UB input_bank_8`  
`/*10`
- general input **bank** (inputs 65-72). `*/public UB input_bank_9`  
`/*11`
- general input **bank** (inputs 73-80). `*/public UB output_bank_0`  
`/*12`
- general output **bank** (outputs 1-8). `*/public UB output_bank_1`  
`/*13`
- general output **bank** (outputs 9-16). `*/public UB output_bank_2`  
`/*14`
- general output **bank** (outputs 17-24). `*/public UB output_bank_3`

- /\* 15*
  - general output [bank](#) (outputs 25-32). *\*/public UB output\_bank\_4*
  - /\* 16*
  - general output [bank](#) (outputs 33-40). *\*/public UB output\_bank\_5*
  - /\* 17*
  - general output [bank](#) (outputs 41-48). *\*/public UB output\_bank\_6*
  - /\* 18*
  - general output [bank](#) (outputs 49-56). *\*/public UB output\_bank\_7*
  - /\* 19*
  - general output [bank](#) (outputs 57-64). *\*/public UB output\_bank\_8*
  - /\* 20*
  - general output [bank](#) (outputs 65-72). *\*/public UB output\_bank\_9*
  - /\* 21*
  - general output [bank](#) (outputs 73-80). *\*/public SW reserved\_0*
  - /\* 22-23*
- H User defined variable.\*[] [byte\\_array](#) ()

## Data Fields

- UW [sample\\_number](#)
  - /\* 00-01*
- sample number.\*UB [input\\_bank\\_0](#)
  - /\* 02*
- Reserved.\*SW [reserved\\_2](#)
  - /\* 24-25*
- Reserved.\*SW [reserved\\_4](#)
  - /\* 26-27*
- Reserved.\*SW [reserved\\_6](#)
  - /\* 28-29*
- Reserved.\*SW [reserved\\_8](#)
  - /\* 30-31*
- Reserved.\*SW [reserved\\_10](#)
  - /\* 32-33*
- Reserved.\*SW [reserved\\_12](#)
  - /\* 34-35*
- Reserved.\*SW [reserved\\_14](#)
  - /\* 36-37*
- Reserved.\*UB [reserved\\_16](#)
  - /\* 38*
- Reserved.\*UB [reserved\\_17](#)
  - /\* 39*
- Reserved.\*UB [reserved\\_18](#)
  - /\* 40*
- Reserved.\*UB [reserved\\_19](#)
  - /\* 41*
- Reserved.\*UB [reserved\\_20](#)
  - /\* 42*
- Reserved.\*UB [reserved\\_21](#)
  - /\* 43*
- Reserved.\*UB [reserved\\_22](#)
  - /\* 44*

- Reserved.\*UB [reserved\\_23](#)  
/\*45
- Reserved.\*UB [error\\_code](#)  
/\*46
- [error code](#).\*UB [thread\\_status](#)  
/\*47
- thread status.\*UL [reserved\\_24](#)  
/\*48-51
- Reserved.\*UL [contour\\_segment\\_count](#)  
/\*52-55
- Segment Count for Contour Mode.\*UW [contour\\_buffer\\_available](#)  
/\*56-57
- Buffer space **remaining**
- Buffer space Contour Mode.\*UW [s\\_plane\\_segment\\_count](#)  
/\*58-59
- segment count of coordinated move for S plane.\*UW [s\\_plane\\_move\\_status](#)  
/\*60-61
- coordinated move status for S plane.\*SL [s\\_distance](#)  
/\*62-65
- distance traveled in coordinated move for S plane.\*UW [s\\_plane\\_buffer\\_available](#)  
/\*66-67
- Buffer space S Plane.\*UW [t\\_plane\\_segment\\_count](#)  
/\*68-69
- segment count of coordinated move for T plane.\*UW [t\\_plane\\_move\\_status](#)  
/\*70-71
- Coordinated move status for T plane.\*SL [t\\_distance](#)  
/\*72-75
- distance traveled in coordinated move for T plane.\*UW [t\\_plane\\_buffer\\_available](#)  
/\*76-77
- Buffer space T Plane.\*UW [axis\\_a\\_status](#)  
/\*78-79
- A axis status.\*UB [axis\\_a\\_switches](#)  
/\*80
- A axis switches.\*UB [axis\\_a\\_stop\\_code](#)  
/\*81
- A axis stop code.\*SL [axis\\_a\\_reference\\_position](#)  
/\*82-85
- A axis reference position.\*SL [axis\\_a\\_motor\\_position](#)  
/\*86-89
- A axis motor position.\*SL [axis\\_a\\_position\\_error](#)  
/\*90-93
- A axis position error.\*SL [axis\\_a\\_aux\\_position](#)  
/\*94-97
- A axis auxiliary position.\*SL [axis\\_a\\_velocity](#)  
/\*98-101
- A axis velocity.\*SL [axis\\_a\\_torque](#)  
/\*102-105
- A axis torque.\*UW [axis\\_a\\_analog\\_in](#)  
/\*106-107
- A axis analog input.\*UB [axis\\_a\\_reserved\\_0](#)  
/\*108

- Reserved.\*UB [axis\\_a\\_reserved\\_1](#)  
/\* 109
- Reserved.\*SL [axis\\_a\\_variable](#)  
/\* 110-113
- A User defined variable.\*[] UW [axis\\_b\\_status](#)  
/\* 114-115
- B axis status.\*UB [axis\\_b\\_switches](#)  
/\* 116
- B axis switches.\*UB [axis\\_b\\_stop\\_code](#)  
/\* 117
- B axis stop code.\*SL [axis\\_b\\_reference\\_position](#)  
/\* 118-121
- B axis reference position.\*SL [axis\\_b\\_motor\\_position](#)  
/\* 122-125
- B axis motor position.\*SL [axis\\_b\\_position\\_error](#)  
/\* 126-129
- B axis position error.\*SL [axis\\_b\\_aux\\_position](#)  
/\* 130-133
- B axis auxiliary position.\*SL [axis\\_b\\_velocity](#)  
/\* 134-137
- B axis velocity.\*SL [axis\\_b\\_torque](#)  
/\* 138-141
- B axis torque.\*UW [axis\\_b\\_analog\\_in](#)  
/\* 142-143
- B axis analog input.\*UB [axis\\_b\\_reserved\\_0](#)  
/\* 144
- Reserved.\*UB [axis\\_b\\_reserved\\_1](#)  
/\* 145
- Reserved.\*SL [axis\\_b\\_variable](#)  
/\* 146-149
- B User defined variable.\*[] UW [axis\\_c\\_status](#)  
/\* 150-151
- C axis status.\*UB [axis\\_c\\_switches](#)  
/\* 152
- C axis switches.\*UB [axis\\_c\\_stop\\_code](#)  
/\* 153
- C axis stop code.\*SL [axis\\_c\\_reference\\_position](#)  
/\* 154-157
- C axis reference position.\*SL [axis\\_c\\_motor\\_position](#)  
/\* 158-161
- C axis motor position.\*SL [axis\\_c\\_position\\_error](#)  
/\* 162-165
- C axis position error.\*SL [axis\\_c\\_aux\\_position](#)  
/\* 166-169
- C axis auxiliary position.\*SL [axis\\_c\\_velocity](#)  
/\* 170-173
- C axis velocity.\*SL [axis\\_c\\_torque](#)  
/\* 174-177
- C axis torque.\*UW [axis\\_c\\_analog\\_in](#)  
/\* 178-179
- C axis analog input.\*UB [axis\\_c\\_reserved\\_0](#)

- /\* 180*
  - Reserved.\*UB [axis\\_c\\_reserved\\_1](#)
- /\* 181*
  - Reserved.\*SL [axis\\_c\\_variable](#)
- /\* 182-185*
  - C User defined variable.\*[] UW [axis\\_d\\_status](#)
- /\* 186-187*
  - D axis status.\*UB [axis\\_d\\_switches](#)
- /\* 188*
  - D axis switches.\*UB [axis\\_d\\_stop\\_code](#)
- /\* 189*
  - D axis stop code.\*SL [axis\\_d\\_reference\\_position](#)
- /\* 190-193*
  - D axis reference position.\*SL [axis\\_d\\_motor\\_position](#)
- /\* 194-197*
  - D axis motor position.\*SL [axis\\_d\\_position\\_error](#)
- /\* 198-201*
  - D axis position error.\*SL [axis\\_d\\_aux\\_position](#)
- /\* 202-205*
  - D axis auxiliary position.\*SL [axis\\_d\\_velocity](#)
- /\* 206-209*
  - D axis velocity.\*SL [axis\\_d\\_torque](#)
- /\* 210-213*
  - D axis torque.\*UW [axis\\_d\\_analog\\_in](#)
- /\* 214-215*
  - D axis analog input.\*UB [axis\\_d\\_reserved\\_0](#)
- /\* 216*
  - Reserved.\*UB [axis\\_d\\_reserved\\_1](#)
- /\* 217*
  - Reserved.\*SL [axis\\_d\\_variable](#)
- /\* 218-221*
  - D User defined variable.\*[] UW [axis\\_e\\_status](#)
- /\* 222-223*
  - E axis status.\*UB [axis\\_e\\_switches](#)
- /\* 224*
  - E axis switches.\*UB [axis\\_e\\_stop\\_code](#)
- /\* 225*
  - E axis stop code.\*SL [axis\\_e\\_reference\\_position](#)
- /\* 226-229*
  - E axis reference position.\*SL [axis\\_e\\_motor\\_position](#)
- /\* 230-233*
  - E axis motor position.\*SL [axis\\_e\\_position\\_error](#)
- /\* 234-237*
  - E axis position error.\*SL [axis\\_e\\_aux\\_position](#)
- /\* 238-241*
  - E axis auxiliary position.\*SL [axis\\_e\\_velocity](#)
- /\* 242-245*
  - E axis velocity.\*SL [axis\\_e\\_torque](#)
- /\* 256-249*
  - E axis torque.\*UW [axis\\_e\\_analog\\_in](#)
- /\* 250-251*

- E axis analog input.\*UB [axis\\_e\\_reserved\\_0](#)  
*/\*252*
- Reserved.\*UB [axis\\_e\\_reserved\\_1](#)  
*/\*253*
- Reserved.\*SL [axis\\_e\\_variable](#)  
*/\*254-257*
- E User defined variable.\*[] UW [axis\\_f\\_status](#)  
*/\*258-259*
- F axis status.\*UB [axis\\_f\\_switches](#)  
*/\*260*
- F axis switches.\*UB [axis\\_f\\_stop\\_code](#)  
*/\*261*
- F axis stop code.\*SL [axis\\_f\\_reference\\_position](#)  
*/\*262-265*
- F axis reference position.\*SL [axis\\_f\\_motor\\_position](#)  
*/\*266-269*
- F axis motor position.\*SL [axis\\_f\\_position\\_error](#)  
*/\*270-273*
- F axis position error.\*SL [axis\\_f\\_aux\\_position](#)  
*/\*274-277*
- F axis auxiliary position.\*SL [axis\\_f\\_velocity](#)  
*/\*278-281*
- F axis velocity.\*SL [axis\\_f\\_torque](#)  
*/\*282-285*
- F axis torque.\*UW [axis\\_f\\_analog\\_in](#)  
*/\*286-287*
- F axis analog input.\*UB [axis\\_f\\_reserved\\_0](#)  
*/\*288*
- Reserved.\*UB [axis\\_f\\_reserved\\_1](#)  
*/\*289*
- Reserved.\*SL [axis\\_f\\_variable](#)  
*/\*290-293*
- F User defined variable.\*[] UW [axis\\_g\\_status](#)  
*/\*294-295*
- G axis status.\*UB [axis\\_g\\_switches](#)  
*/\*296*
- G axis switches.\*UB [axis\\_g\\_stop\\_code](#)  
*/\*297*
- G axis stop code.\*SL [axis\\_g\\_reference\\_position](#)  
*/\*298-301*
- G axis reference position.\*SL [axis\\_g\\_motor\\_position](#)  
*/\*302-305*
- G axis motor position.\*SL [axis\\_g\\_position\\_error](#)  
*/\*306-309*
- G axis position error.\*SL [axis\\_g\\_aux\\_position](#)  
*/\*310-313*
- G axis auxiliary position.\*SL [axis\\_g\\_velocity](#)  
*/\*314-317*
- G axis velocity.\*SL [axis\\_g\\_torque](#)  
*/\*318-321*
- G axis torque.\*UW [axis\\_g\\_analog\\_in](#)

- /\*322-323*

  - G axis analog input.\*UB [axis\\_g\\_reserved\\_0](#)
- /\*324*

  - Reserved.\*UB [axis\\_g\\_reserved\\_1](#)
- /\*325*

  - Reserved.\*SL [axis\\_g\\_variable](#)
- /\*326-329*

  - G User defined variable.\*[] UW [axis\\_h\\_status](#)
- /\*330-331*

  - H axis status.\*UB [axis\\_h\\_switches](#)
- /\*332*

  - H axis switches.\*UB [axis\\_h\\_stop\\_code](#)
- /\*333*

  - H axis stop code.\*SL [axis\\_h\\_reference\\_position](#)
- /\*334-337*

  - H axis reference position.\*SL [axis\\_h\\_motor\\_position](#)
- /\*338-341*

  - H axis motor position.\*SL [axis\\_h\\_position\\_error](#)
- /\*342-345*

  - H axis position error.\*SL [axis\\_h\\_aux\\_position](#)
- /\*346-349*

  - H axis auxiliary position.\*SL [axis\\_h\\_velocity](#)
- /\*350-353*

  - H axis velocity.\*SL [axis\\_h\\_torque](#)
- /\*354-357*

  - H axis torque.\*UW [axis\\_h\\_analog\\_in](#)
- /\*358-359*

  - H axis analog input.\*UB [axis\\_h\\_reserved\\_0](#)
- /\*360*

  - Reserved.\*UB [axis\\_h\\_reserved\\_1](#)
- /\*361*

  - Reserved.\*SL [axis\\_h\\_variable](#)
- /\*362-365*

### 12.20.1 Detailed Description

Definition at line 799 of file gclib.vb.

### 12.20.2 Member Function Documentation

#### 12.20.2.1 `byte_array()`

H User defined variable.\* [] `byte_array ( )`

Implements [GDataRecord.byte\\_array](#)

Implements [Gclib.GDataRecord](#).

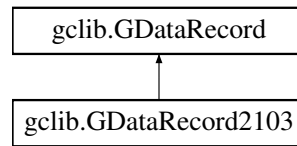
The documentation for this struct was generated from the following file:

- [gclib.vb](#)

## 12.21 gclib.GDataRecord2103 Struct Reference

Data record struct for DMC-2103 controllers.

Inheritance diagram for gclib.GDataRecord2103:



### Public Member Functions

- `byte[] byte_array ()`  
*Returns the data record as a byte array and allows for access to individual bytes.*

### Data Fields

- UB `header_0`  
*1st Byte of Header.*
- UB `header_1`  
*2nd Byte of Header.*
- UB `header_2`  
*3rd Byte of Header.*
- UB `header_3`  
*4th Byte of Header.*
- UW `sample_number`  
*sample number.*
- UB `input_bank_0`  
*general input bank 0 (inputs 1-8).*
- UB `input_bank_1`  
*general input bank 1 (inputs 9-16).*
- UB `input_bank_2`  
*general input bank 2 (inputs 17-24).*
- UB `input_bank_3`  
*general input bank 3 (inputs 25-32).*
- UB `input_bank_4`  
*general input bank 4 (inputs 33-40).*
- UB `input_bank_5`  
*general input bank 5 (inputs 41-48).*
- UB `input_bank_6`  
*general input bank 6 (inputs 49-56).*
- UB `input_bank_7`  
*general input bank 7 (inputs 57-64).*
- UB `input_bank_8`  
*general input bank 8 (inputs 65-72).*
- UB `input_bank_9`  
*general input bank 9 (inputs 73-80).*
- UB `output_bank_0`  
*general output bank 0 (outputs 1-8).*
- UB `output_bank_1`  
*general output bank 1 (outputs 9-16).*



- UB [output\\_bank\\_2](#)  
*general output bank 2 (outputs 17-24).*
- UB [output\\_bank\\_3](#)  
*general output bank 3 (outputs 25-32).*
- UB [output\\_bank\\_4](#)  
*general output bank 4 (outputs 33-40).*
- UB [output\\_bank\\_5](#)  
*general output bank 5 (outputs 41-48).*
- UB [output\\_bank\\_6](#)  
*general output bank 6 (outputs 49-56).*
- UB [output\\_bank\\_7](#)  
*general output bank 7 (outputs 57-64).*
- UB [output\\_bank\\_8](#)  
*general output bank 8 (outputs 65-72).*
- UB [output\\_bank\\_9](#)  
*general output bank 9 (outputs 73-80).*
- UB [error\\_code](#)  
*error code.*
- UB [general\\_status](#)  
*general status*
- UW [s\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)  
*coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*
- UW [t\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for T plane.*
- UW [t\\_plane\\_move\\_status](#)  
*Coordinated move status for T plane.*
- SL [t\\_distance](#)  
*distance traveled in coordinated move for T plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)  
*A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SW [axis\\_a\\_torque](#)  
*A axis torque.*
- UW [axis\\_a\\_analog\\_in](#)

- A axis analog input.*
- UW [axis\\_b\\_status](#)  
*B axis status.*
- UB [axis\\_b\\_switches](#)  
*B axis switches.*
- UB [axis\\_b\\_stop\\_code](#)  
*B axis stop code.*
- SL [axis\\_b\\_reference\\_position](#)  
*B axis reference position.*
- SL [axis\\_b\\_motor\\_position](#)  
*B axis motor position.*
- SL [axis\\_b\\_position\\_error](#)  
*B axis position error.*
- SL [axis\\_b\\_aux\\_position](#)  
*B axis auxiliary position.*
- SL [axis\\_b\\_velocity](#)  
*B axis velocity.*
- SW [axis\\_b\\_torque](#)  
*B axis torque.*
- UW [axis\\_b\\_analog\\_in](#)  
*B axis analog input.*
- UW [axis\\_c\\_status](#)  
*C axis status.*
- UB [axis\\_c\\_switches](#)  
*C axis switches.*
- UB [axis\\_c\\_stop\\_code](#)  
*C axis stop code.*
- SL [axis\\_c\\_reference\\_position](#)  
*C axis reference position.*
- SL [axis\\_c\\_motor\\_position](#)  
*C axis motor position.*
- SL [axis\\_c\\_position\\_error](#)  
*C axis position error.*
- SL [axis\\_c\\_aux\\_position](#)  
*C axis auxiliary position.*
- SL [axis\\_c\\_velocity](#)  
*C axis velocity.*
- SW [axis\\_c\\_torque](#)  
*C axis torque.*
- UW [axis\\_c\\_analog\\_in](#)  
*C axis analog input.*
- UW [axis\\_d\\_status](#)  
*D axis status.*
- UB [axis\\_d\\_switches](#)  
*D axis switches.*
- UB [axis\\_d\\_stop\\_code](#)  
*D axis stop code.*
- SL [axis\\_d\\_reference\\_position](#)  
*D axis reference position.*
- SL [axis\\_d\\_motor\\_position](#)  
*D axis motor position.*

- SL [axis\\_d\\_position\\_error](#)  
*D axis position error.*
- SL [axis\\_d\\_aux\\_position](#)  
*D axis auxiliary position.*
- SL [axis\\_d\\_velocity](#)  
*D axis velocity.*
- SW [axis\\_d\\_torque](#)  
*D axis torque.*
- UW [axis\\_d\\_analog\\_in](#)  
*D axis analog input.*
- UW [axis\\_e\\_status](#)  
*E axis status.*
- UB [axis\\_e\\_switches](#)  
*E axis switches.*
- UB [axis\\_e\\_stop\\_code](#)  
*E axis stop code.*
- SL [axis\\_e\\_reference\\_position](#)  
*E axis reference position.*
- SL [axis\\_e\\_motor\\_position](#)  
*E axis motor position.*
- SL [axis\\_e\\_position\\_error](#)  
*E axis position error.*
- SL [axis\\_e\\_aux\\_position](#)  
*E axis auxiliary position.*
- SL [axis\\_e\\_velocity](#)  
*E axis velocity.*
- SW [axis\\_e\\_torque](#)  
*E axis torque.*
- UW [axis\\_e\\_analog\\_in](#)  
*E axis analog input.*
- UW [axis\\_f\\_status](#)  
*F axis status.*
- UB [axis\\_f\\_switches](#)  
*F axis switches.*
- UB [axis\\_f\\_stop\\_code](#)  
*F axis stop code.*
- SL [axis\\_f\\_reference\\_position](#)  
*F axis reference position.*
- SL [axis\\_f\\_motor\\_position](#)  
*F axis motor position.*
- SL [axis\\_f\\_position\\_error](#)  
*F axis position error.*
- SL [axis\\_f\\_aux\\_position](#)  
*F axis auxiliary position.*
- SL [axis\\_f\\_velocity](#)  
*F axis velocity.*
- SW [axis\\_f\\_torque](#)  
*F axis torque.*
- UW [axis\\_f\\_analog\\_in](#)  
*F axis analog input.*
- UW [axis\\_g\\_status](#)

- *G axis status.*
- UB [axis\\_g\\_switches](#)  
*G axis switches.*
- UB [axis\\_g\\_stop\\_code](#)  
*G axis stop code.*
- SL [axis\\_g\\_reference\\_position](#)  
*G axis reference position.*
- SL [axis\\_g\\_motor\\_position](#)  
*G axis motor position.*
- SL [axis\\_g\\_position\\_error](#)  
*G axis position error.*
- SL [axis\\_g\\_aux\\_position](#)  
*G axis auxiliary position.*
- SL [axis\\_g\\_velocity](#)  
*G axis velocity.*
- SW [axis\\_g\\_torque](#)  
*G axis torque.*
- UW [axis\\_g\\_analog\\_in](#)  
*G axis analog input.*
- UW [axis\\_h\\_status](#)  
*H axis status.*
- UB [axis\\_h\\_switches](#)  
*H axis switches.*
- UB [axis\\_h\\_stop\\_code](#)  
*H axis stop code.*
- SL [axis\\_h\\_reference\\_position](#)  
*H axis reference position.*
- SL [axis\\_h\\_motor\\_position](#)  
*H axis motor position.*
- SL [axis\\_h\\_position\\_error](#)  
*H axis position error.*
- SL [axis\\_h\\_aux\\_position](#)  
*H axis auxiliary position.*
- SL [axis\\_h\\_velocity](#)  
*H axis velocity.*
- SW [axis\\_h\\_torque](#)  
*H axis torque.*
- UW [axis\\_h\\_analog\\_in](#)  
*H axis analog input.*

### 12.21.1 Detailed Description

Data record struct for DMC-2103 controllers.

Definition at line 1473 of file gclib.cs.

The documentation for this struct was generated from the following file:

- [gclib.cs](#)

## 12.22 GDataRecord2103 Struct Reference

Data record struct for DMC-2103 controllers.

```
#include <gclib_record.h>
```

## Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*sample number.*
- UB [input\\_bank\\_0](#)  
*general input bank 0 (inputs 1-8).*
- UB [input\\_bank\\_1](#)  
*general input bank 1 (inputs 9-16).*
- UB [input\\_bank\\_2](#)  
*general input bank 2 (inputs 17-24).*
- UB [input\\_bank\\_3](#)  
*general input bank 3 (inputs 25-32).*
- UB [input\\_bank\\_4](#)  
*general input bank 4 (inputs 33-40).*
- UB [input\\_bank\\_5](#)  
*general input bank 5 (inputs 41-48).*
- UB [input\\_bank\\_6](#)  
*general input bank 6 (inputs 49-56).*
- UB [input\\_bank\\_7](#)  
*general input bank 7 (inputs 57-64).*
- UB [input\\_bank\\_8](#)  
*general input bank 8 (inputs 65-72).*
- UB [input\\_bank\\_9](#)  
*general input bank 9 (inputs 73-80).*
- UB [output\\_bank\\_0](#)  
*general output bank 0 (outputs 1-8).*
- UB [output\\_bank\\_1](#)  
*general output bank 1 (outputs 9-16).*
- UB [output\\_bank\\_2](#)  
*general output bank 2 (outputs 17-24).*
- UB [output\\_bank\\_3](#)  
*general output bank 3 (outputs 25-32).*
- UB [output\\_bank\\_4](#)  
*general output bank 4 (outputs 33-40).*
- UB [output\\_bank\\_5](#)  
*general output bank 5 (outputs 41-48).*
- UB [output\\_bank\\_6](#)  
*general output bank 6 (outputs 49-56).*
- UB [output\\_bank\\_7](#)  
*general output bank 7 (outputs 57-64).*
- UB [output\\_bank\\_8](#)  
*general output bank 8 (outputs 65-72).*
- UB [output\\_bank\\_9](#)

- general output bank 9 (outputs 73-80).*
- UB [error\\_code](#)  
*error code.*
- UB [general\\_status](#)  
*general status*
- UW [s\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)  
*coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*
- UW [t\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for T plane.*
- UW [t\\_plane\\_move\\_status](#)  
*Coordinated move status for T plane.*
- SL [t\\_distance](#)  
*distance traveled in coordinated move for T plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)  
*A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SW [axis\\_a\\_torque](#)  
*A axis torque.*
- UW [axis\\_a\\_analog\\_in](#)  
*A axis analog input.*
- UW [axis\\_b\\_status](#)  
*B axis status.*
- UB [axis\\_b\\_switches](#)  
*B axis switches.*
- UB [axis\\_b\\_stop\\_code](#)  
*B axis stop code.*
- SL [axis\\_b\\_reference\\_position](#)  
*B axis reference position.*
- SL [axis\\_b\\_motor\\_position](#)  
*B axis motor position.*
- SL [axis\\_b\\_position\\_error](#)  
*B axis position error.*
- SL [axis\\_b\\_aux\\_position](#)  
*B axis auxiliary position.*

- SL [axis\\_b\\_velocity](#)  
*B axis velocity.*
- SW [axis\\_b\\_torque](#)  
*B axis torque.*
- UW [axis\\_b\\_analog\\_in](#)  
*B axis analog input.*
- UW [axis\\_c\\_status](#)  
*C axis status.*
- UB [axis\\_c\\_switches](#)  
*C axis switches.*
- UB [axis\\_c\\_stop\\_code](#)  
*C axis stop code.*
- SL [axis\\_c\\_reference\\_position](#)  
*C axis reference position.*
- SL [axis\\_c\\_motor\\_position](#)  
*C axis motor position.*
- SL [axis\\_c\\_position\\_error](#)  
*C axis position error.*
- SL [axis\\_c\\_aux\\_position](#)  
*C axis auxiliary position.*
- SL [axis\\_c\\_velocity](#)  
*C axis velocity.*
- SW [axis\\_c\\_torque](#)  
*C axis torque.*
- UW [axis\\_c\\_analog\\_in](#)  
*C axis analog input.*
- UW [axis\\_d\\_status](#)  
*D axis status.*
- UB [axis\\_d\\_switches](#)  
*D axis switches.*
- UB [axis\\_d\\_stop\\_code](#)  
*D axis stop code.*
- SL [axis\\_d\\_reference\\_position](#)  
*D axis reference position.*
- SL [axis\\_d\\_motor\\_position](#)  
*D axis motor position.*
- SL [axis\\_d\\_position\\_error](#)  
*D axis position error.*
- SL [axis\\_d\\_aux\\_position](#)  
*D axis auxiliary position.*
- SL [axis\\_d\\_velocity](#)  
*D axis velocity.*
- SW [axis\\_d\\_torque](#)  
*D axis torque.*
- UW [axis\\_d\\_analog\\_in](#)  
*D axis analog input.*
- UW [axis\\_e\\_status](#)  
*E axis status.*
- UB [axis\\_e\\_switches](#)  
*E axis switches.*
- UB [axis\\_e\\_stop\\_code](#)

- E axis stop code.*
- SL [axis\\_e\\_reference\\_position](#)  
*E axis reference position.*
- SL [axis\\_e\\_motor\\_position](#)  
*E axis motor position.*
- SL [axis\\_e\\_position\\_error](#)  
*E axis position error.*
- SL [axis\\_e\\_aux\\_position](#)  
*E axis auxiliary position.*
- SL [axis\\_e\\_velocity](#)  
*E axis velocity.*
- SW [axis\\_e\\_torque](#)  
*E axis torque.*
- UW [axis\\_e\\_analog\\_in](#)  
*E axis analog input.*
- UW [axis\\_f\\_status](#)  
*F axis status.*
- UB [axis\\_f\\_switches](#)  
*F axis switches.*
- UB [axis\\_f\\_stop\\_code](#)  
*F axis stop code.*
- SL [axis\\_f\\_reference\\_position](#)  
*F axis reference position.*
- SL [axis\\_f\\_motor\\_position](#)  
*F axis motor position.*
- SL [axis\\_f\\_position\\_error](#)  
*F axis position error.*
- SL [axis\\_f\\_aux\\_position](#)  
*F axis auxiliary position.*
- SL [axis\\_f\\_velocity](#)  
*F axis velocity.*
- SW [axis\\_f\\_torque](#)  
*F axis torque.*
- UW [axis\\_f\\_analog\\_in](#)  
*F axis analog input.*
- UW [axis\\_g\\_status](#)  
*G axis status.*
- UB [axis\\_g\\_switches](#)  
*G axis switches.*
- UB [axis\\_g\\_stop\\_code](#)  
*G axis stop code.*
- SL [axis\\_g\\_reference\\_position](#)  
*G axis reference position.*
- SL [axis\\_g\\_motor\\_position](#)  
*G axis motor position.*
- SL [axis\\_g\\_position\\_error](#)  
*G axis position error.*
- SL [axis\\_g\\_aux\\_position](#)  
*G axis auxiliary position.*
- SL [axis\\_g\\_velocity](#)  
*G axis velocity.*



- SW [axis\\_g\\_torque](#)  
*G axis torque.*
- UW [axis\\_g\\_analog\\_in](#)  
*G axis analog input.*
- UW [axis\\_h\\_status](#)  
*H axis status.*
- UB [axis\\_h\\_switches](#)  
*H axis switches.*
- UB [axis\\_h\\_stop\\_code](#)  
*H axis stop code.*
- SL [axis\\_h\\_reference\\_position](#)  
*H axis reference position.*
- SL [axis\\_h\\_motor\\_position](#)  
*H axis motor position.*
- SL [axis\\_h\\_position\\_error](#)  
*H axis position error.*
- SL [axis\\_h\\_aux\\_position](#)  
*H axis auxiliary position.*
- SL [axis\\_h\\_velocity](#)  
*H axis velocity.*
- SW [axis\\_h\\_torque](#)  
*H axis torque.*
- UW [axis\\_h\\_analog\\_in](#)  
*H axis analog input.*

### 12.22.1 Detailed Description

Data record struct for DMC-2103 controllers.

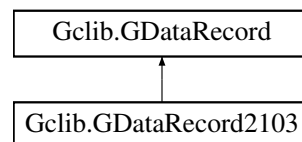
Definition at line 586 of file `gclib_record.h`.

The documentation for this struct was generated from the following file:

- [gclib\\_record.h](#)

## 12.23 Gclib.GDataRecord2103 Struct Reference

Inheritance diagram for Gclib.GDataRecord2103:



### Public Member Functions

- general input [bank](#) (inputs 1-8). `*/public UB input_bank_1`  
`/*07`
- general input [bank](#) (inputs 9-16). `*/public UB input_bank_2`  
`/*08`
- general input [bank](#) (inputs 17-24). `*/public UB input_bank_3`  
`/*09`
- general input [bank](#) (inputs 25-32). `*/public UB input_bank_4`

- */\* 10*  
• general input [bank](#) (inputs 33-40). *\*/public UB input\_bank\_5*
- */\* 11*  
• general input [bank](#) (inputs 41-48). *\*/public UB input\_bank\_6*
- */\* 12*  
• general input [bank](#) (inputs 49-56). *\*/public UB input\_bank\_7*
- */\* 13*  
• general input [bank](#) (inputs 57-64). *\*/public UB input\_bank\_8*
- */\* 14*  
• general input [bank](#) (inputs 65-72). *\*/public UB input\_bank\_9*
- */\* 15*  
• general input [bank](#) (inputs 73-80). *\*/public UB output\_bank\_0*
- */\* 16*  
• general output [bank](#) (outputs 1-8). *\*/public UB output\_bank\_1*
- */\* 17*  
• general output [bank](#) (outputs 9-16). *\*/public UB output\_bank\_2*
- */\* 18*  
• general output [bank](#) (outputs 17-24). *\*/public UB output\_bank\_3*
- */\* 19*  
• general output [bank](#) (outputs 25-32). *\*/public UB output\_bank\_4*
- */\* 20*  
• general output [bank](#) (outputs 33-40). *\*/public UB output\_bank\_5*
- */\* 21*  
• general output [bank](#) (outputs 41-48). *\*/public UB output\_bank\_6*
- */\* 22*  
• general output [bank](#) (outputs 49-56). *\*/public UB output\_bank\_7*
- */\* 23*  
• general output [bank](#) (outputs 57-64). *\*/public UB output\_bank\_8*
- */\* 24*  
• general output [bank](#) (outputs 65-72). *\*/public UB output\_bank\_9*
- */\* 25*  
• general output [bank](#) (outputs 73-80). *\*/public UB error\_code*
- */\* 26*  
• H axis analog input.\* [byte\\_array](#) ()

## Data Fields

- UB [header\\_0](#)  
*/\* 00*
- byte of Header.\*UB [header\\_1](#)  
*/\* 01*
- byte of Header.\*UB [header\\_2](#)  
*/\* 02*
- byte of Header.\*UB [header\\_3](#)  
*/\* 03*
- byte of Header.\*UW [sample\\_number](#)  
*/\* 04-05*
- sample number.\*UB [input\\_bank\\_0](#)  
*/\* 06*
- [error](#) code.\*UB [general\\_status](#)  
*/\* 27*

- general status.\*UW [s\\_plane\\_segment\\_count](#)  
*/\*28-29*
- segment count of coordinated move for S plane.\*UW [s\\_plane\\_move\\_status](#)  
*/\*30-31*
- coordinated move status for S plane.\*SL [s\\_distance](#)  
*/\*32-35*
- distance traveled in coordinated move for S plane.\*UW [t\\_plane\\_segment\\_count](#)  
*/\*36-37*
- segment count of coordinated move for T plane.\*UW [t\\_plane\\_move\\_status](#)  
*/\*38-39*
- Coordinated move status for T plane.\*SL [t\\_distance](#)  
*/\*40-43*
- distance traveled in coordinated move for T plane.\*UW [axis\\_a\\_status](#)  
*/\*44-45*
- A axis status.\*UB [axis\\_a\\_switches](#)  
*/\*46*
- A axis switches.\*UB [axis\\_a\\_stop\\_code](#)  
*/\*47*
- A axis stop code.\*SL [axis\\_a\\_reference\\_position](#)  
*/\*48-51*
- A axis reference position.\*SL [axis\\_a\\_motor\\_position](#)  
*/\*52-55*
- A axis motor position.\*SL [axis\\_a\\_position\\_error](#)  
*/\*56-59*
- A axis position error.\*SL [axis\\_a\\_aux\\_position](#)  
*/\*60-63*
- A axis auxiliary position.\*SL [axis\\_a\\_velocity](#)  
*/\*64-67*
- A axis velocity.\*SW [axis\\_a\\_torque](#)  
*/\*68-69*
- A axis torque.\*UW [axis\\_a\\_analog\\_in](#)  
*/\*70-71*
- A axis analog input.\*UW [axis\\_b\\_status](#)  
*/\*72-73*
- B axis status.\*UB [axis\\_b\\_switches](#)  
*/\*74*
- B axis switches.\*UB [axis\\_b\\_stop\\_code](#)  
*/\*75*
- B axis stop code.\*SL [axis\\_b\\_reference\\_position](#)  
*/\*76-79*
- B axis reference position.\*SL [axis\\_b\\_motor\\_position](#)  
*/\*80-83*
- B axis motor position.\*SL [axis\\_b\\_position\\_error](#)  
*/\*84-87*
- B axis position error.\*SL [axis\\_b\\_aux\\_position](#)  
*/\*88-91*
- B axis auxiliary position.\*SL [axis\\_b\\_velocity](#)  
*/\*92-95*
- B axis velocity.\*SW [axis\\_b\\_torque](#)  
*/\*96-97*
- B axis torque.\*UW [axis\\_b\\_analog\\_in](#)

- /\*98-99*
- B axis analog input.\*UW [axis\\_c\\_status](#)  
*/\*100-101*
- C axis status.\*UB [axis\\_c\\_switches](#)  
*/\*102*
- C axis switches.\*UB [axis\\_c\\_stop\\_code](#)  
*/\*103*
- C axis stop code.\*SL [axis\\_c\\_reference\\_position](#)  
*/\*104-107*
- C axis reference position.\*SL [axis\\_c\\_motor\\_position](#)  
*/\*108-111*
- C axis motor position.\*SL [axis\\_c\\_position\\_error](#)  
*/\*112-115*
- C axis position error.\*SL [axis\\_c\\_aux\\_position](#)  
*/\*116-119*
- C axis auxiliary position.\*SL [axis\\_c\\_velocity](#)  
*/\*120-123*
- C axis velocity.\*SW [axis\\_c\\_torque](#)  
*/\*124-125*
- C axis torque.\*UW [axis\\_c\\_analog\\_in](#)  
*/\*126-127*
- C axis analog input.\*UW [axis\\_d\\_status](#)  
*/\*128-129*
- D axis status.\*UB [axis\\_d\\_switches](#)  
*/\*130*
- D axis switches.\*UB [axis\\_d\\_stop\\_code](#)  
*/\*131*
- D axis stop code.\*SL [axis\\_d\\_reference\\_position](#)  
*/\*132-135*
- D axis reference position.\*SL [axis\\_d\\_motor\\_position](#)  
*/\*136-139*
- D axis motor position.\*SL [axis\\_d\\_position\\_error](#)  
*/\*140-143*
- D axis position error.\*SL [axis\\_d\\_aux\\_position](#)  
*/\*144-147*
- D axis auxiliary position.\*SL [axis\\_d\\_velocity](#)  
*/\*148-151*
- D axis velocity.\*SW [axis\\_d\\_torque](#)  
*/\*152-153*
- D axis torque.\*UW [axis\\_d\\_analog\\_in](#)  
*/\*154-155*
- D axis analog input.\*UW [axis\\_e\\_status](#)  
*/\*156-157*
- E axis status.\*UB [axis\\_e\\_switches](#)  
*/\*158*
- E axis switches.\*UB [axis\\_e\\_stop\\_code](#)  
*/\*159*
- E axis stop code.\*SL [axis\\_e\\_reference\\_position](#)  
*/\*160-163*
- E axis reference position.\*SL [axis\\_e\\_motor\\_position](#)  
*/\*164-167*

- E axis motor position.\*SL [axis\\_e\\_position\\_error](#)  
*/\*168-171*
- E axis position error.\*SL [axis\\_e\\_aux\\_position](#)  
*/\*172-175*
- E axis auxiliary position.\*SL [axis\\_e\\_velocity](#)  
*/\*176-179*
- E axis velocity.\*SW [axis\\_e\\_torque](#)  
*/\*180-181*
- E axis torque.\*UW [axis\\_e\\_analog\\_in](#)  
*/\*182-183*
- E axis analog input.\*UW [axis\\_f\\_status](#)  
*/\*184-185*
- F axis status.\*UB [axis\\_f\\_switches](#)  
*/\*186*
- F axis switches.\*UB [axis\\_f\\_stop\\_code](#)  
*/\*187*
- F axis stop code.\*SL [axis\\_f\\_reference\\_position](#)  
*/\*188-191*
- F axis reference position.\*SL [axis\\_f\\_motor\\_position](#)  
*/\*192-195*
- F axis motor position.\*SL [axis\\_f\\_position\\_error](#)  
*/\*196-199*
- F axis position error.\*SL [axis\\_f\\_aux\\_position](#)  
*/\*200-203*
- F axis auxiliary position.\*SL [axis\\_f\\_velocity](#)  
*/\*204-207*
- F axis velocity.\*SW [axis\\_f\\_torque](#)  
*/\*208-209*
- F axis torque.\*UW [axis\\_f\\_analog\\_in](#)  
*/\*210-211*
- F axis analog input.\*UW [axis\\_g\\_status](#)  
*/\*212-213*
- G axis status.\*UB [axis\\_g\\_switches](#)  
*/\*214*
- G axis switches.\*UB [axis\\_g\\_stop\\_code](#)  
*/\*215*
- G axis stop code.\*SL [axis\\_g\\_reference\\_position](#)  
*/\*216-219*
- G axis reference position.\*SL [axis\\_g\\_motor\\_position](#)  
*/\*220-223*
- G axis motor position.\*SL [axis\\_g\\_position\\_error](#)  
*/\*224-227*
- G axis position error.\*SL [axis\\_g\\_aux\\_position](#)  
*/\*228-231*
- G axis auxiliary position.\*SL [axis\\_g\\_velocity](#)  
*/\*232-235*
- G axis velocity.\*SW [axis\\_g\\_torque](#)  
*/\*236-237*
- G axis torque.\*UW [axis\\_g\\_analog\\_in](#)  
*/\*238-239*
- G axis analog input.\*UW [axis\\_h\\_status](#)

- /\*240-241
- H axis status.\*UB [axis\\_h\\_switches](#)
- /\*242
- H axis switches.\*UB [axis\\_h\\_stop\\_code](#)
- /\*243
- H axis stop code.\*SL [axis\\_h\\_reference\\_position](#)
- /\*244-247
- H axis reference position.\*SL [axis\\_h\\_motor\\_position](#)
- /\*248-251
- H axis motor position.\*SL [axis\\_h\\_position\\_error](#)
- /\*252-255
- H axis position error.\*SL [axis\\_h\\_aux\\_position](#)
- /\*256-259
- H axis auxiliary position.\*SL [axis\\_h\\_velocity](#)
- /\*260-263
- H axis velocity.\*SW [axis\\_h\\_torque](#)
- /\*264-265
- H axis torque.\*UW [axis\\_h\\_analog\\_in](#)
- /\*266-267

### 12.23.1 Detailed Description

Definition at line 958 of file gclib.vb.

### 12.23.2 Member Function Documentation

#### 12.23.2.1 `byte_array()`

H axis analog input.\* `byte_array ( )`

Implements [GDataRecord.byte\\_array](#)

Implements [Gclib.GDataRecord](#).

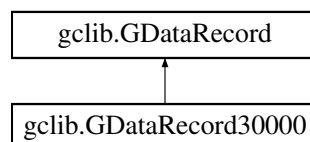
The documentation for this struct was generated from the following file:

- [gclib.vb](#)

## 12.24 `gclib.GDataRecord30000` Struct Reference

Data record struct for DMC-30010 controllers.

Inheritance diagram for `gclib.GDataRecord30000`:



### Public Member Functions

- `byte[]` [byte\\_array \( \)](#)

*Returns the data record as a byte array and allows for access to individual bytes.*

## Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*sample number.*
- UB [input\\_bank\\_0](#)  
*general input bank 0 (inputs 1-8).*
- UB [input\\_bank\\_1](#)  
*general input bank 1 (inputs 9-16).*
- UB [output\\_bank\\_0](#)  
*general output bank 0 (outputs 1-8).*
- UB [output\\_bank\\_1](#)  
*general output bank 1 (outputs 9-16).*
- UB [error\\_code](#)  
*error code.*
- UB [thread\\_status](#)  
*thread status.*
- UW [input\\_analog\\_2](#)  
*Analog input 2. 1 is in axis data, see [axis\\_a\\_analog\\_in](#).*
- UW [output\\_analog\\_1](#)  
*Analog output 1.*
- UW [output\\_analog\\_2](#)  
*Analog output 2.*
- UL [amplifier\\_status](#)  
*Amplifier Status.*
- UL [contour\\_segment\\_count](#)  
*Segment Count for Contour Mode.*
- UW [contour\\_buffer\\_available](#)  
*Buffer space remaining, Contour Mode.*
- UW [s\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)  
*coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*
- UW [s\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, S Plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)

- *A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SL [axis\\_a\\_torque](#)  
*A axis torque.*
- UW [axis\\_a\\_analog\\_in](#)  
*A axis analog input.*
- UB [axis\\_a\\_halls](#)  
*A Hall Input Status.*
- UB [axis\\_a\\_reserved](#)  
*Reserved.*
- SL [axis\\_a\\_variable](#)  
*A User-defined variable (ZA).*

### 12.24.1 Detailed Description

Data record struct for DMC-30010 controllers.

Definition at line 1708 of file `gclib.cs`.

The documentation for this struct was generated from the following file:

- [gclib.cs](#)

## 12.25 GDataRecord30000 Struct Reference

Data record struct for DMC-30010 controllers.

```
#include <gclib_record.h>
```

### Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*sample number.*
- UB [input\\_bank\\_0](#)  
*general input bank 0 (inputs 1-8).*
- UB [input\\_bank\\_1](#)  
*general input bank 1 (inputs 9-16).*
- UB [output\\_bank\\_0](#)  
*general output bank 0 (outputs 1-8).*
- UB [output\\_bank\\_1](#)



- general output bank 1 (outputs 9-16).*
- UB [error\\_code](#)  
*error code.*
- UB [thread\\_status](#)  
*thread status.*
- UW [input\\_analog\\_2](#)  
*Analog input 2. 1 is in axis data, see [axis\\_a\\_analog\\_in](#).*
- UW [output\\_analog\\_1](#)  
*Analog output 1.*
- UW [output\\_analog\\_2](#)  
*Analog output 2.*
- UL [amplifier\\_status](#)  
*Amplifier Status.*
- UL [contour\\_segment\\_count](#)  
*Segment Count for Contour Mode.*
- UW [contour\\_buffer\\_available](#)  
*Buffer space remaining, Contour Mode.*
- UW [s\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)  
*coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*
- UW [s\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, S Plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)  
*A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SL [axis\\_a\\_torque](#)  
*A axis torque.*
- UW [axis\\_a\\_analog\\_in](#)  
*A axis analog input.*
- UB [axis\\_a\\_halls](#)  
*A Hall Input Status.*
- UB [axis\\_a\\_reserved](#)  
*Reserved.*
- SL [axis\\_a\\_variable](#)  
*A User-defined variable (ZA).*

### 12.25.1 Detailed Description

Data record struct for DMC-30010 controllers.

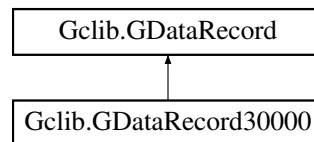
Definition at line 818 of file `gclib_record.h`.

The documentation for this struct was generated from the following file:

- [gclib\\_record.h](#)

## 12.26 Gclib.GDataRecord30000 Struct Reference

Inheritance diagram for `Gclib.GDataRecord30000`:



### Public Member Functions

- general input `bank` (inputs 1-8). `*/public UB input_bank_1`  
`/*07`
- general input `bank` (inputs 9-16). `*/public UB output_bank_0`  
`/*08`
- general output `bank` (outputs 1-8). `*/public UB output_bank_1`  
`/*09`
- general output `bank` (outputs 9-16). `*/public UB error_code`  
`/*10`
- A User defined variable. `*[] byte_array ()`

### Data Fields

- UB `header_0`  
`/*00`
- byte of Header. `*UB header_1`  
`/*01`
- byte of Header. `*UB header_2`  
`/*02`
- byte of Header. `*UB header_3`  
`/*03`
- byte of Header. `*UW sample_number`  
`/*04-05`
- sample number. `*UB input_bank_0`  
`/*06`
- error code. `*UB thread_status`  
`/*11`
- thread status. `*UW input_analog_2`  
`/*12-13`
- Analog input. is in axis `data`
- Analog input. is in axis see axis\_a\_analog\_in. `*UW output_analog_1`  
`/*14-15`
- Analog output. `*UW output_analog_2`  
`/*16-17`

- Analog output.\*UL [amplifier\\_status](#)  
/\*18-21
- Amplifier Status.\*UL [contour\\_segment\\_count](#)  
/\*22-25
- Segment Count for Contour Mode.\*UW [contour\\_buffer\\_available](#)  
/\*26-27
- Buffer space **remaining**
- Buffer space Contour Mode.\*UW [s\\_plane\\_segment\\_count](#)  
/\*28-29
- segment count of coordinated move for S plane.\*UW [s\\_plane\\_move\\_status](#)  
/\*30-31
- coordinated move status for S plane.\*SL [s\\_distance](#)  
/\*32-35
- distance traveled in coordinated move for S plane.\*UW [s\\_plane\\_buffer\\_available](#)  
/\*36-37
- Buffer space S Plane.\*UW [axis\\_a\\_status](#)  
/\*38-39
- A axis status.\*UB [axis\\_a\\_switches](#)  
/\*40
- A axis switches.\*UB [axis\\_a\\_stop\\_code](#)  
/\*41
- A axis stop code.\*SL [axis\\_a\\_reference\\_position](#)  
/\*42-45
- A axis reference position.\*SL [axis\\_a\\_motor\\_position](#)  
/\*46-49
- A axis motor position.\*SL [axis\\_a\\_position\\_error](#)  
/\*50-53
- A axis position error.\*SL [axis\\_a\\_aux\\_position](#)  
/\*54-57
- A axis auxiliary position.\*SL [axis\\_a\\_velocity](#)  
/\*58-61
- A axis velocity.\*SL [axis\\_a\\_torque](#)  
/\*62-65
- A axis torque.\*UW [axis\\_a\\_analog\\_in](#)  
/\*66-67
- A axis analog input.\*UB [axis\\_a\\_halls](#)  
/\*68
- A Hall Input Status.\*UB [axis\\_a\\_reserved](#)  
/\*69
- Reserved.\*SL [axis\\_a\\_variable](#)  
/\*70-73

### 12.26.1 Detailed Description

Definition at line 1154 of file gcLib.vb.

### 12.26.2 Member Function Documentation

### 12.26.2.1 `byte_array()`

A User defined variable.\* `[] byte_array ( )`

Implements [GDataRecord.byte\\_array](#)

Implements [Gclib.GDataRecord](#).

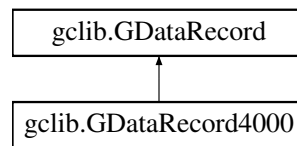
The documentation for this struct was generated from the following file:

- [gclib.vb](#)

## 12.27 `gclib.GDataRecord4000` Struct Reference

Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0.

Inheritance diagram for `gclib.GDataRecord4000`:



### Public Member Functions

- `byte[] byte_array ( )`  
*Returns the data record as a byte array and allows for access to individual bytes.*

### Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*sample number.*
- UB [input\\_bank\\_0](#)  
*general input bank 0 (inputs 1-8).*
- UB [input\\_bank\\_1](#)  
*general input bank 1 (inputs 9-16).*
- UB [input\\_bank\\_2](#)  
*general input bank 2 (inputs 17-24).*
- UB [input\\_bank\\_3](#)  
*general input bank 3 (inputs 25-32).*
- UB [input\\_bank\\_4](#)  
*general input bank 4 (inputs 33-40).*
- UB [input\\_bank\\_5](#)  
*general input bank 5 (inputs 41-48).*
- UB [input\\_bank\\_6](#)  
*general input bank 6 (inputs 49-56).*
- UB [input\\_bank\\_7](#)  
*general input bank 7 (inputs 57-64).*

- UB [input\\_bank\\_8](#)  
*general input bank 8 (inputs 65-72).*
- UB [input\\_bank\\_9](#)  
*general input bank 9 (inputs 73-80).*
- UB [output\\_bank\\_0](#)  
*general output bank 0 (outputs 1-8).*
- UB [output\\_bank\\_1](#)  
*general output bank 1 (outputs 9-16).*
- UB [output\\_bank\\_2](#)  
*general output bank 2 (outputs 17-24).*
- UB [output\\_bank\\_3](#)  
*general output bank 3 (outputs 25-32).*
- UB [output\\_bank\\_4](#)  
*general output bank 4 (outputs 33-40).*
- UB [output\\_bank\\_5](#)  
*general output bank 5 (outputs 41-48).*
- UB [output\\_bank\\_6](#)  
*general output bank 6 (outputs 49-56).*
- UB [output\\_bank\\_7](#)  
*general output bank 7 (outputs 57-64).*
- UB [output\\_bank\\_8](#)  
*general output bank 8 (outputs 65-72).*
- UB [output\\_bank\\_9](#)  
*general output bank 9 (outputs 73-80).*
- SW [reserved\\_0](#)  
*Reserved.*
- SW [reserved\\_2](#)  
*Reserved.*
- SW [reserved\\_4](#)  
*Reserved.*
- SW [reserved\\_6](#)  
*Reserved.*
- SW [reserved\\_8](#)  
*Reserved.*
- SW [reserved\\_10](#)  
*Reserved.*
- SW [reserved\\_12](#)  
*Reserved.*
- SW [reserved\\_14](#)  
*Reserved.*
- UB [ethernet\\_status\\_a](#)  
*Ethernet Handle A Status.*
- UB [ethernet\\_status\\_b](#)  
*Ethernet Handle B Status.*
- UB [ethernet\\_status\\_c](#)  
*Ethernet Handle C Status.*
- UB [ethernet\\_status\\_d](#)  
*Ethernet Handle D Status.*
- UB [ethernet\\_status\\_e](#)  
*Ethernet Handle E Status.*
- UB [ethernet\\_status\\_f](#)

- Ethernet Handle F Status.*
- UB [ethernet\\_status\\_g](#)
  - Ethernet Handle G Status.*
- UB [ethernet\\_status\\_h](#)
  - Ethernet Handle H Status.*
- UB [error\\_code](#)
  - error code.*
- UB [thread\\_status](#)
  - thread status*
- UL [amplifier\\_status](#)
  - Amplifier Status.*
- UL [contour\\_segment\\_count](#)
  - Segment Count for Contour Mode.*
- UW [contour\\_buffer\\_available](#)
  - Buffer space remaining, Contour Mode.*
- UW [s\\_plane\\_segment\\_count](#)
  - segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)
  - coordinated move status for S plane.*
- SL [s\\_distance](#)
  - distance traveled in coordinated move for S plane.*
- UW [s\\_plane\\_buffer\\_available](#)
  - Buffer space remaining, S Plane.*
- UW [t\\_plane\\_segment\\_count](#)
  - segment count of coordinated move for T plane.*
- UW [t\\_plane\\_move\\_status](#)
  - Coordinated move status for T plane.*
- SL [t\\_distance](#)
  - distance traveled in coordinated move for T plane.*
- UW [t\\_plane\\_buffer\\_available](#)
  - Buffer space remaining, T Plane.*
- UW [axis\\_a\\_status](#)
  - A axis status.*
- UB [axis\\_a\\_switches](#)
  - A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)
  - A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)
  - A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)
  - A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)
  - A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)
  - A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)
  - A axis velocity.*
- SL [axis\\_a\\_torque](#)
  - A axis torque.*
- UW [axis\\_a\\_analog\\_in](#)
  - A axis analog input.*

- UB [axis\\_a\\_halls](#)  
*A Hall Input Status.*
- UB [axis\\_a\\_reserved](#)  
*Reserved.*
- SL [axis\\_a\\_variable](#)  
*A User-defined variable (ZA).*
- UW [axis\\_b\\_status](#)  
*B axis status.*
- UB [axis\\_b\\_switches](#)  
*B axis switches.*
- UB [axis\\_b\\_stop\\_code](#)  
*B axis stop code.*
- SL [axis\\_b\\_reference\\_position](#)  
*B axis reference position.*
- SL [axis\\_b\\_motor\\_position](#)  
*B axis motor position.*
- SL [axis\\_b\\_position\\_error](#)  
*B axis position error.*
- SL [axis\\_b\\_aux\\_position](#)  
*B axis auxiliary position.*
- SL [axis\\_b\\_velocity](#)  
*B axis velocity.*
- SL [axis\\_b\\_torque](#)  
*B axis torque.*
- UW [axis\\_b\\_analog\\_in](#)  
*B axis analog input.*
- UB [axis\\_b\\_halls](#)  
*B Hall Input Status.*
- UB [axis\\_b\\_reserved](#)  
*Reserved.*
- SL [axis\\_b\\_variable](#)  
*B User-defined variable (ZA).*
- UW [axis\\_c\\_status](#)  
*C axis status.*
- UB [axis\\_c\\_switches](#)  
*C axis switches.*
- UB [axis\\_c\\_stop\\_code](#)  
*C axis stop code.*
- SL [axis\\_c\\_reference\\_position](#)  
*C axis reference position.*
- SL [axis\\_c\\_motor\\_position](#)  
*C axis motor position.*
- SL [axis\\_c\\_position\\_error](#)  
*C axis position error.*
- SL [axis\\_c\\_aux\\_position](#)  
*C axis auxiliary position.*
- SL [axis\\_c\\_velocity](#)  
*C axis velocity.*
- SL [axis\\_c\\_torque](#)  
*C axis torque.*
- UW [axis\\_c\\_analog\\_in](#)

- C axis analog input.*
- UB [axis\\_c\\_halls](#)
  - C Hall Input Status.*
- UB [axis\\_c\\_reserved](#)
  - Reserved.*
- SL [axis\\_c\\_variable](#)
  - C User-defined variable (ZA).*
- UW [axis\\_d\\_status](#)
  - D axis status.*
- UB [axis\\_d\\_switches](#)
  - D axis switches.*
- UB [axis\\_d\\_stop\\_code](#)
  - D axis stop code.*
- SL [axis\\_d\\_reference\\_position](#)
  - D axis reference position.*
- SL [axis\\_d\\_motor\\_position](#)
  - D axis motor position.*
- SL [axis\\_d\\_position\\_error](#)
  - D axis position error.*
- SL [axis\\_d\\_aux\\_position](#)
  - D axis auxiliary position.*
- SL [axis\\_d\\_velocity](#)
  - D axis velocity.*
- SL [axis\\_d\\_torque](#)
  - D axis torque.*
- UW [axis\\_d\\_analog\\_in](#)
  - D axis analog input.*
- UB [axis\\_d\\_halls](#)
  - D Hall Input Status.*
- UB [axis\\_d\\_reserved](#)
  - Reserved.*
- SL [axis\\_d\\_variable](#)
  - D User-defined variable (ZA).*
- UW [axis\\_e\\_status](#)
  - E axis status.*
- UB [axis\\_e\\_switches](#)
  - E axis switches.*
- UB [axis\\_e\\_stop\\_code](#)
  - E axis stop code.*
- SL [axis\\_e\\_reference\\_position](#)
  - E axis reference position.*
- SL [axis\\_e\\_motor\\_position](#)
  - E axis motor position.*
- SL [axis\\_e\\_position\\_error](#)
  - E axis position error.*
- SL [axis\\_e\\_aux\\_position](#)
  - E axis auxiliary position.*
- SL [axis\\_e\\_velocity](#)
  - E axis velocity.*
- SL [axis\\_e\\_torque](#)
  - E axis torque.*



- UW [axis\\_e\\_analog\\_in](#)  
*E axis analog input.*
- UB [axis\\_e\\_halls](#)  
*E Hall Input Status.*
- UB [axis\\_e\\_reserved](#)  
*Reserved.*
- SL [axis\\_e\\_variable](#)  
*E User-defined variable (ZA).*
- UW [axis\\_f\\_status](#)  
*F axis status.*
- UB [axis\\_f\\_switches](#)  
*F axis switches.*
- UB [axis\\_f\\_stop\\_code](#)  
*F axis stop code.*
- SL [axis\\_f\\_reference\\_position](#)  
*F axis reference position.*
- SL [axis\\_f\\_motor\\_position](#)  
*F axis motor position.*
- SL [axis\\_f\\_position\\_error](#)  
*F axis position error.*
- SL [axis\\_f\\_aux\\_position](#)  
*F axis auxiliary position.*
- SL [axis\\_f\\_velocity](#)  
*F axis velocity.*
- SL [axis\\_f\\_torque](#)  
*F axis torque.*
- UW [axis\\_f\\_analog\\_in](#)  
*F axis analog input.*
- UB [axis\\_f\\_halls](#)  
*F Hall Input Status.*
- UB [axis\\_f\\_reserved](#)  
*Reserved.*
- SL [axis\\_f\\_variable](#)  
*F User-defined variable (ZA).*
- UW [axis\\_g\\_status](#)  
*G axis status.*
- UB [axis\\_g\\_switches](#)  
*G axis switches.*
- UB [axis\\_g\\_stop\\_code](#)  
*G axis stop code.*
- SL [axis\\_g\\_reference\\_position](#)  
*G axis reference position.*
- SL [axis\\_g\\_motor\\_position](#)  
*G axis motor position.*
- SL [axis\\_g\\_position\\_error](#)  
*G axis position error.*
- SL [axis\\_g\\_aux\\_position](#)  
*G axis auxiliary position.*
- SL [axis\\_g\\_velocity](#)  
*G axis velocity.*
- SL [axis\\_g\\_torque](#)

- G axis torque.*
- UW [axis\\_g\\_analog\\_in](#)
  - G axis analog input.*
- UB [axis\\_g\\_halls](#)
  - G Hall Input Status.*
- UB [axis\\_g\\_reserved](#)
  - Reserved.*
- SL [axis\\_g\\_variable](#)
  - G User-defined variable (ZA).*
- UW [axis\\_h\\_status](#)
  - H axis status.*
- UB [axis\\_h\\_switches](#)
  - H axis switches.*
- UB [axis\\_h\\_stop\\_code](#)
  - H axis stop code.*
- SL [axis\\_h\\_reference\\_position](#)
  - H axis reference position.*
- SL [axis\\_h\\_motor\\_position](#)
  - H axis motor position.*
- SL [axis\\_h\\_position\\_error](#)
  - H axis position error.*
- SL [axis\\_h\\_aux\\_position](#)
  - H axis auxiliary position.*
- SL [axis\\_h\\_velocity](#)
  - H axis velocity.*
- SL [axis\\_h\\_torque](#)
  - H axis torque.*
- UW [axis\\_h\\_analog\\_in](#)
  - H axis analog input.*
- UB [axis\\_h\\_halls](#)
  - H Hall Input Status.*
- UB [axis\\_h\\_reserved](#)
  - Reserved.*
- SL [axis\\_h\\_variable](#)
  - H User-defined variable (ZA).*

### 12.27.1 Detailed Description

Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0.

Definition at line 918 of file `gclib.cs`.

The documentation for this struct was generated from the following file:

- [gclib.cs](#)

## 12.28 GDataRecord4000 Struct Reference

Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0.

```
#include <gclib_record.h>
```

## Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*sample number.*
- UB [input\\_bank\\_0](#)  
*general input bank 0 (inputs 1-8).*
- UB [input\\_bank\\_1](#)  
*general input bank 1 (inputs 9-16).*
- UB [input\\_bank\\_2](#)  
*general input bank 2 (inputs 17-24).*
- UB [input\\_bank\\_3](#)  
*general input bank 3 (inputs 25-32).*
- UB [input\\_bank\\_4](#)  
*general input bank 4 (inputs 33-40).*
- UB [input\\_bank\\_5](#)  
*general input bank 5 (inputs 41-48).*
- UB [input\\_bank\\_6](#)  
*general input bank 6 (inputs 49-56).*
- UB [input\\_bank\\_7](#)  
*general input bank 7 (inputs 57-64).*
- UB [input\\_bank\\_8](#)  
*general input bank 8 (inputs 65-72).*
- UB [input\\_bank\\_9](#)  
*general input bank 9 (inputs 73-80).*
- UB [output\\_bank\\_0](#)  
*general output bank 0 (outputs 1-8).*
- UB [output\\_bank\\_1](#)  
*general output bank 1 (outputs 9-16).*
- UB [output\\_bank\\_2](#)  
*general output bank 2 (outputs 17-24).*
- UB [output\\_bank\\_3](#)  
*general output bank 3 (outputs 25-32).*
- UB [output\\_bank\\_4](#)  
*general output bank 4 (outputs 33-40).*
- UB [output\\_bank\\_5](#)  
*general output bank 5 (outputs 41-48).*
- UB [output\\_bank\\_6](#)  
*general output bank 6 (outputs 49-56).*
- UB [output\\_bank\\_7](#)  
*general output bank 7 (outputs 57-64).*
- UB [output\\_bank\\_8](#)  
*general output bank 8 (outputs 65-72).*
- UB [output\\_bank\\_9](#)

- general output bank 9 (outputs 73-80).*
- SW [reserved\\_0](#)  
*Reserved.*
- SW [reserved\\_2](#)  
*Reserved.*
- SW [reserved\\_4](#)  
*Reserved.*
- SW [reserved\\_6](#)  
*Reserved.*
- SW [reserved\\_8](#)  
*Reserved.*
- SW [reserved\\_10](#)  
*Reserved.*
- SW [reserved\\_12](#)  
*Reserved.*
- SW [reserved\\_14](#)  
*Reserved.*
- UB [ethernet\\_status\\_a](#)  
*Ethernet Handle A Status.*
- UB [ethernet\\_status\\_b](#)  
*Ethernet Handle B Status.*
- UB [ethernet\\_status\\_c](#)  
*Ethernet Handle C Status.*
- UB [ethernet\\_status\\_d](#)  
*Ethernet Handle D Status.*
- UB [ethernet\\_status\\_e](#)  
*Ethernet Handle E Status.*
- UB [ethernet\\_status\\_f](#)  
*Ethernet Handle F Status.*
- UB [ethernet\\_status\\_g](#)  
*Ethernet Handle G Status.*
- UB [ethernet\\_status\\_h](#)  
*Ethernet Handle H Status.*
- UB [error\\_code](#)  
*error code.*
- UB [thread\\_status](#)  
*thread status*
- UL [amplifier\\_status](#)  
*Amplifier Status.*
- UL [contour\\_segment\\_count](#)  
*Segment Count for Contour Mode.*
- UW [contour\\_buffer\\_available](#)  
*Buffer space remaining, Contour Mode.*
- UW [s\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)  
*coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*
- UW [s\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, S Plane.*

- UW [t\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for T plane.*
- UW [t\\_plane\\_move\\_status](#)  
*Coordinated move status for T plane.*
- SL [t\\_distance](#)  
*distance traveled in coordinated move for T plane.*
- UW [t\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, T Plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)  
*A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SL [axis\\_a\\_torque](#)  
*A axis torque.*
- UW [axis\\_a\\_analog\\_in](#)  
*A axis analog input.*
- UB [axis\\_a\\_halls](#)  
*A Hall Input Status.*
- UB [axis\\_a\\_reserved](#)  
*Reserved.*
- SL [axis\\_a\\_variable](#)  
*A User-defined variable (ZA).*
- UW [axis\\_b\\_status](#)  
*B axis status.*
- UB [axis\\_b\\_switches](#)  
*B axis switches.*
- UB [axis\\_b\\_stop\\_code](#)  
*B axis stop code.*
- SL [axis\\_b\\_reference\\_position](#)  
*B axis reference position.*
- SL [axis\\_b\\_motor\\_position](#)  
*B axis motor position.*
- SL [axis\\_b\\_position\\_error](#)  
*B axis position error.*
- SL [axis\\_b\\_aux\\_position](#)  
*B axis auxiliary position.*
- SL [axis\\_b\\_velocity](#)  
*B axis velocity.*
- SL [axis\\_b\\_torque](#)

- B axis torque.*
- UW [axis\\_b\\_analog\\_in](#)  
*B axis analog input.*
- UB [axis\\_b\\_halls](#)  
*B Hall Input Status.*
- UB [axis\\_b\\_reserved](#)  
*Reserved.*
- SL [axis\\_b\\_variable](#)  
*B User-defined variable (ZA).*
- UW [axis\\_c\\_status](#)  
*C axis status.*
- UB [axis\\_c\\_switches](#)  
*C axis switches.*
- UB [axis\\_c\\_stop\\_code](#)  
*C axis stop code.*
- SL [axis\\_c\\_reference\\_position](#)  
*C axis reference position.*
- SL [axis\\_c\\_motor\\_position](#)  
*C axis motor position.*
- SL [axis\\_c\\_position\\_error](#)  
*C axis position error.*
- SL [axis\\_c\\_aux\\_position](#)  
*C axis auxiliary position.*
- SL [axis\\_c\\_velocity](#)  
*C axis velocity.*
- SL [axis\\_c\\_torque](#)  
*C axis torque.*
- UW [axis\\_c\\_analog\\_in](#)  
*C axis analog input.*
- UB [axis\\_c\\_halls](#)  
*C Hall Input Status.*
- UB [axis\\_c\\_reserved](#)  
*Reserved.*
- SL [axis\\_c\\_variable](#)  
*C User-defined variable (ZA).*
- UW [axis\\_d\\_status](#)  
*D axis status.*
- UB [axis\\_d\\_switches](#)  
*D axis switches.*
- UB [axis\\_d\\_stop\\_code](#)  
*D axis stop code.*
- SL [axis\\_d\\_reference\\_position](#)  
*D axis reference position.*
- SL [axis\\_d\\_motor\\_position](#)  
*D axis motor position.*
- SL [axis\\_d\\_position\\_error](#)  
*D axis position error.*
- SL [axis\\_d\\_aux\\_position](#)  
*D axis auxiliary position.*
- SL [axis\\_d\\_velocity](#)  
*D axis velocity.*

- SL [axis\\_d\\_torque](#)  
*D axis torque.*
- UW [axis\\_d\\_analog\\_in](#)  
*D axis analog input.*
- UB [axis\\_d\\_halls](#)  
*D Hall Input Status.*
- UB [axis\\_d\\_reserved](#)  
*Reserved.*
- SL [axis\\_d\\_variable](#)  
*D User-defined variable (ZA).*
- UW [axis\\_e\\_status](#)  
*E axis status.*
- UB [axis\\_e\\_switches](#)  
*E axis switches.*
- UB [axis\\_e\\_stop\\_code](#)  
*E axis stop code.*
- SL [axis\\_e\\_reference\\_position](#)  
*E axis reference position.*
- SL [axis\\_e\\_motor\\_position](#)  
*E axis motor position.*
- SL [axis\\_e\\_position\\_error](#)  
*E axis position error.*
- SL [axis\\_e\\_aux\\_position](#)  
*E axis auxiliary position.*
- SL [axis\\_e\\_velocity](#)  
*E axis velocity.*
- SL [axis\\_e\\_torque](#)  
*E axis torque.*
- UW [axis\\_e\\_analog\\_in](#)  
*E axis analog input.*
- UB [axis\\_e\\_halls](#)  
*E Hall Input Status.*
- UB [axis\\_e\\_reserved](#)  
*Reserved.*
- SL [axis\\_e\\_variable](#)  
*E User-defined variable (ZA).*
- UW [axis\\_f\\_status](#)  
*F axis status.*
- UB [axis\\_f\\_switches](#)  
*F axis switches.*
- UB [axis\\_f\\_stop\\_code](#)  
*F axis stop code.*
- SL [axis\\_f\\_reference\\_position](#)  
*F axis reference position.*
- SL [axis\\_f\\_motor\\_position](#)  
*F axis motor position.*
- SL [axis\\_f\\_position\\_error](#)  
*F axis position error.*
- SL [axis\\_f\\_aux\\_position](#)  
*F axis auxiliary position.*
- SL [axis\\_f\\_velocity](#)

- F axis velocity.*
- SL [axis\\_f\\_torque](#)  
*F axis torque.*
- UW [axis\\_f\\_analog\\_in](#)  
*F axis analog input.*
- UB [axis\\_f\\_halls](#)  
*F Hall Input Status.*
- UB [axis\\_f\\_reserved](#)  
*Reserved.*
- SL [axis\\_f\\_variable](#)  
*F User-defined variable (ZA).*
- UW [axis\\_g\\_status](#)  
*G axis status.*
- UB [axis\\_g\\_switches](#)  
*G axis switches.*
- UB [axis\\_g\\_stop\\_code](#)  
*G axis stop code.*
- SL [axis\\_g\\_reference\\_position](#)  
*G axis reference position.*
- SL [axis\\_g\\_motor\\_position](#)  
*G axis motor position.*
- SL [axis\\_g\\_position\\_error](#)  
*G axis position error.*
- SL [axis\\_g\\_aux\\_position](#)  
*G axis auxiliary position.*
- SL [axis\\_g\\_velocity](#)  
*G axis velocity.*
- SL [axis\\_g\\_torque](#)  
*G axis torque.*
- UW [axis\\_g\\_analog\\_in](#)  
*G axis analog input.*
- UB [axis\\_g\\_halls](#)  
*G Hall Input Status.*
- UB [axis\\_g\\_reserved](#)  
*Reserved.*
- SL [axis\\_g\\_variable](#)  
*G User-defined variable (ZA).*
- UW [axis\\_h\\_status](#)  
*H axis status.*
- UB [axis\\_h\\_switches](#)  
*H axis switches.*
- UB [axis\\_h\\_stop\\_code](#)  
*H axis stop code.*
- SL [axis\\_h\\_reference\\_position](#)  
*H axis reference position.*
- SL [axis\\_h\\_motor\\_position](#)  
*H axis motor position.*
- SL [axis\\_h\\_position\\_error](#)  
*H axis position error.*
- SL [axis\\_h\\_aux\\_position](#)  
*H axis auxiliary position.*



- SL [axis\\_h\\_velocity](#)  
*H axis velocity.*
- SL [axis\\_h\\_torque](#)  
*H axis torque.*
- UW [axis\\_h\\_analog\\_in](#)  
*H axis analog input.*
- UB [axis\\_h\\_halls](#)  
*H Hall Input Status.*
- UB [axis\\_h\\_reserved](#)  
*Reserved.*
- SL [axis\\_h\\_variable](#)  
*H User-defined variable (ZA).*

### 12.28.1 Detailed Description

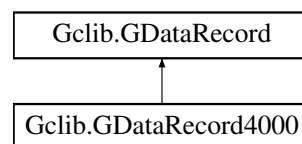
Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0. Definition at line 35 of file `gclib_record.h`.

The documentation for this struct was generated from the following file:

- [gclib\\_record.h](#)

## 12.29 Gclib.GDataRecord4000 Struct Reference

Inheritance diagram for Gclib.GDataRecord4000:



### Public Member Functions

- general input [bank](#) (inputs 1-8). `*/public UB input_bank_1`  
`/*07`
- general input [bank](#) (inputs 9-16). `*/public UB input_bank_2`  
`/*08`
- general input [bank](#) (inputs 17-24). `*/public UB input_bank_3`  
`/*09`
- general input [bank](#) (inputs 25-32). `*/public UB input_bank_4`  
`/*10`
- general input [bank](#) (inputs 33-40). `*/public UB input_bank_5`  
`/*11`
- general input [bank](#) (inputs 41-48). `*/public UB input_bank_6`  
`/*12`
- general input [bank](#) (inputs 49-56). `*/public UB input_bank_7`  
`/*13`
- general input [bank](#) (inputs 57-64). `*/public UB input_bank_8`  
`/*14`
- general input [bank](#) (inputs 65-72). `*/public UB input_bank_9`  
`/*15`
- general input [bank](#) (inputs 73-80). `*/public UB output_bank_0`

- /\* 16*
- general output [bank](#) (outputs 1-8). *\*/public UB output\_bank\_1*
- /\* 17*
- general output [bank](#) (outputs 9-16). *\*/public UB output\_bank\_2*
- /\* 18*
- general output [bank](#) (outputs 17-24). *\*/public UB output\_bank\_3*
- /\* 19*
- general output [bank](#) (outputs 25-32). *\*/public UB output\_bank\_4*
- /\* 20*
- general output [bank](#) (outputs 33-40). *\*/public UB output\_bank\_5*
- /\* 21*
- general output [bank](#) (outputs 41-48). *\*/public UB output\_bank\_6*
- /\* 22*
- general output [bank](#) (outputs 49-56). *\*/public UB output\_bank\_7*
- /\* 23*
- general output [bank](#) (outputs 57-64). *\*/public UB output\_bank\_8*
- /\* 24*
- general output [bank](#) (outputs 65-72). *\*/public UB output\_bank\_9*
- /\* 25*
- general output [bank](#) (outputs 73-80). *\*/public SW reserved\_0*
- /\* 26-27*
- H User defined variable.\*[] [byte\\_array](#) ()

## Data Fields

- UB [header\\_0](#)
- /\* 00*
- byte of Header.\*UB [header\\_1](#)
- /\* 01*
- byte of Header.\*UB [header\\_2](#)
- /\* 02*
- byte of Header.\*UB [header\\_3](#)
- /\* 03*
- byte of Header.\*UW [sample\\_number](#)
- /\* 04-05*
- sample number.\*UB [input\\_bank\\_0](#)
- /\* 06*
- Reserved.\*SW [reserved\\_2](#)
- /\* 28-29*
- Reserved.\*SW [reserved\\_4](#)
- /\* 30-31*
- Reserved.\*SW [reserved\\_6](#)
- /\* 32-33*
- Reserved.\*SW [reserved\\_8](#)
- /\* 34-35*
- Reserved.\*SW [reserved\\_10](#)
- /\* 36-37*
- Reserved.\*SW [reserved\\_12](#)
- /\* 38-39*
- Reserved.\*SW [reserved\\_14](#)
- /\* 40-41*

- Reserved.\*UB [ethernet\\_status\\_a](#)  
/\*42
- Ethernet Handle A Status.\*UB [ethernet\\_status\\_b](#)  
/\*43
- Ethernet Handle B Status.\*UB [ethernet\\_status\\_c](#)  
/\*44
- Ethernet Handle C Status.\*UB [ethernet\\_status\\_d](#)  
/\*45
- Ethernet Handle D Status.\*UB [ethernet\\_status\\_e](#)  
/\*46
- Ethernet Handle E Status.\*UB [ethernet\\_status\\_f](#)  
/\*47
- Ethernet Handle F Status.\*UB [ethernet\\_status\\_g](#)  
/\*48
- Ethernet Handle G Status.\*UB [ethernet\\_status\\_h](#)  
/\*49
- Ethernet Handle H Status.\*UB [error\\_code](#)  
/\*50
- [error code](#).\*UB [thread\\_status](#)  
/\*51
- [thread status](#) \*UL [amplifier\\_status](#)  
/\*52-55
- [Amplifier Status](#).\*UL [contour\\_segment\\_count](#)  
/\*56-59
- [Segment Count for Contour Mode](#).\*UW [contour\\_buffer\\_available](#)  
/\*60-61
- Buffer space **remaining**
- [Buffer space Contour Mode](#).\*UW [s\\_plane\\_segment\\_count](#)  
/\*62-63
- [segment count of coordinated move for S plane](#).\*UW [s\\_plane\\_move\\_status](#)  
/\*64-65
- [coordinated move status for S plane](#).\*SL [s\\_distance](#)  
/\*66-69
- [distance traveled in coordinated move for S plane](#).\*UW [s\\_plane\\_buffer\\_available](#)  
/\*70-71
- [Buffer space S Plane](#).\*UW [t\\_plane\\_segment\\_count](#)  
/\*72-73
- [segment count of coordinated move for T plane](#).\*UW [t\\_plane\\_move\\_status](#)  
/\*74-75
- [Coordinated move status for T plane](#).\*SL [t\\_distance](#)  
/\*76-79
- [distance traveled in coordinated move for T plane](#).\*UW [t\\_plane\\_buffer\\_available](#)  
/\*80-81
- [Buffer space T Plane](#).\*UW [axis\\_a\\_status](#)  
/\*82-83
- [A axis status](#).\*UB [axis\\_a\\_switches](#)  
/\*84
- [A axis switches](#).\*UB [axis\\_a\\_stop\\_code](#)  
/\*85
- [A axis stop code](#).\*SL [axis\\_a\\_reference\\_position](#)  
/\*86-89

- A axis reference position.\*SL [axis\\_a\\_motor\\_position](#)  
/\*90-93
- A axis motor position.\*SL [axis\\_a\\_position\\_error](#)  
/\*94-97
- A axis position error.\*SL [axis\\_a\\_aux\\_position](#)  
/\*98-101
- A axis auxiliary position.\*SL [axis\\_a\\_velocity](#)  
/\* 102-105
- A axis velocity.\*SL [axis\\_a\\_torque](#)  
/\* 106-109
- A axis torque.\*UW [axis\\_a\\_analog\\_in](#)  
/\* 110-111
- A axis analog input.\*UB [axis\\_a\\_halls](#)  
/\* 112
- A Hall Input Status.\*UB [axis\\_a\\_reserved](#)  
/\* 113
- Reserved.\*SL [axis\\_a\\_variable](#)  
/\* 114-117
- A User defined variable.\*[] UW [axis\\_b\\_status](#)  
/\* 118-119
- B axis status.\*UB [axis\\_b\\_switches](#)  
/\* 120
- B axis switches.\*UB [axis\\_b\\_stop\\_code](#)  
/\* 121
- B axis stop code.\*SL [axis\\_b\\_reference\\_position](#)  
/\* 122-125
- B axis reference position.\*SL [axis\\_b\\_motor\\_position](#)  
/\* 126-129
- B axis motor position.\*SL [axis\\_b\\_position\\_error](#)  
/\* 130-133
- B axis position error.\*SL [axis\\_b\\_aux\\_position](#)  
/\* 134-137
- B axis auxiliary position.\*SL [axis\\_b\\_velocity](#)  
/\* 138-141
- B axis velocity.\*SL [axis\\_b\\_torque](#)  
/\* 142-145
- B axis torque.\*UW [axis\\_b\\_analog\\_in](#)  
/\* 146-147
- B axis analog input.\*UB [axis\\_b\\_halls](#)  
/\* 148
- B Hall Input Status.\*UB [axis\\_b\\_reserved](#)  
/\* 149
- Reserved.\*SL [axis\\_b\\_variable](#)  
/\* 150-153
- B User defined variable.\*[] UW [axis\\_c\\_status](#)  
/\* 154-155
- C axis status.\*UB [axis\\_c\\_switches](#)  
/\* 156
- C axis switches.\*UB [axis\\_c\\_stop\\_code](#)  
/\* 157
- C axis stop code.\*SL [axis\\_c\\_reference\\_position](#)

- /\* 158-161*
- C axis reference position.\*SL [axis\\_c\\_motor\\_position](#)
- /\* 162-165*
- C axis motor position.\*SL [axis\\_c\\_position\\_error](#)
- /\* 166-169*
- C axis position error.\*SL [axis\\_c\\_aux\\_position](#)
- /\* 170-173*
- C axis auxiliary position.\*SL [axis\\_c\\_velocity](#)
- /\* 174-177*
- C axis velocity.\*SL [axis\\_c\\_torque](#)
- /\* 178-181*
- C axis torque.\*UW [axis\\_c\\_analog\\_in](#)
- /\* 182-183*
- C axis analog input.\*UB [axis\\_c\\_halls](#)
- /\* 184*
- C Hall Input Status.\*UB [axis\\_c\\_reserved](#)
- /\* 185*
- Reserved.\*SL [axis\\_c\\_variable](#)
- /\* 186-189*
- C User defined variable.\*[] UW [axis\\_d\\_status](#)
- /\* 190-191*
- D axis status.\*UB [axis\\_d\\_switches](#)
- /\* 192*
- D axis switches.\*UB [axis\\_d\\_stop\\_code](#)
- /\* 193*
- D axis stop code.\*SL [axis\\_d\\_reference\\_position](#)
- /\* 194-197*
- D axis reference position.\*SL [axis\\_d\\_motor\\_position](#)
- /\* 198-201*
- D axis motor position.\*SL [axis\\_d\\_position\\_error](#)
- /\* 202-205*
- D axis position error.\*SL [axis\\_d\\_aux\\_position](#)
- /\* 206-209*
- D axis auxiliary position.\*SL [axis\\_d\\_velocity](#)
- /\* 210-213*
- D axis velocity.\*SL [axis\\_d\\_torque](#)
- /\* 214-217*
- D axis torque.\*UW [axis\\_d\\_analog\\_in](#)
- /\* 218-219*
- D axis analog input.\*UB [axis\\_d\\_halls](#)
- /\* 220*
- D Hall Input Status.\*UB [axis\\_d\\_reserved](#)
- /\* 221*
- Reserved.\*SL [axis\\_d\\_variable](#)
- /\* 222-225*
- D User defined variable.\*[] UW [axis\\_e\\_status](#)
- /\* 226-227*
- E axis status.\*UB [axis\\_e\\_switches](#)
- /\* 228*
- E axis switches.\*UB [axis\\_e\\_stop\\_code](#)
- /\* 229*

- E axis stop code.\*SL [axis\\_e\\_reference\\_position](#)  
/\*230-233
- E axis reference position.\*SL [axis\\_e\\_motor\\_position](#)  
/\*234-237
- E axis motor position.\*SL [axis\\_e\\_position\\_error](#)  
/\*238-241
- E axis position error.\*SL [axis\\_e\\_aux\\_position](#)  
/\*242-245
- E axis auxiliary position.\*SL [axis\\_e\\_velocity](#)  
/\*246-249
- E axis velocity.\*SL [axis\\_e\\_torque](#)  
/\*250-253
- E axis torque.\*UW [axis\\_e\\_analog\\_in](#)  
/\*254-255
- E axis analog input.\*UB [axis\\_e\\_halls](#)  
/\*256
- E Hall Input Status.\*UB [axis\\_e\\_reserved](#)  
/\*257
- Reserved.\*SL [axis\\_e\\_variable](#)  
/\*258-261
- E User defined variable.\*[] UW [axis\\_f\\_status](#)  
/\*262-263
- F axis status.\*UB [axis\\_f\\_switches](#)  
/\*264
- F axis switches.\*UB [axis\\_f\\_stop\\_code](#)  
/\*265
- F axis stop code.\*SL [axis\\_f\\_reference\\_position](#)  
/\*266-269
- F axis reference position.\*SL [axis\\_f\\_motor\\_position](#)  
/\*270-273
- F axis motor position.\*SL [axis\\_f\\_position\\_error](#)  
/\*274-277
- F axis position error.\*SL [axis\\_f\\_aux\\_position](#)  
/\*278-281
- F axis auxiliary position.\*SL [axis\\_f\\_velocity](#)  
/\*282-285
- F axis velocity.\*SL [axis\\_f\\_torque](#)  
/\*286-289
- F axis torque.\*UW [axis\\_f\\_analog\\_in](#)  
/\*290-291
- F axis analog input.\*UB [axis\\_f\\_halls](#)  
/\*292
- F Hall Input Status.\*UB [axis\\_f\\_reserved](#)  
/\*293
- Reserved.\*SL [axis\\_f\\_variable](#)  
/\*294-297
- F User defined variable.\*[] UW [axis\\_g\\_status](#)  
/\*298-299
- G axis status.\*UB [axis\\_g\\_switches](#)  
/\*300
- G axis switches.\*UB [axis\\_g\\_stop\\_code](#)

- /\*301*
- G axis stop code.\*SL [axis\\_g\\_reference\\_position](#)
  - /\*302-305*
- G axis reference position.\*SL [axis\\_g\\_motor\\_position](#)
  - /\*306-309*
- G axis motor position.\*SL [axis\\_g\\_position\\_error](#)
  - /\*310-313*
- G axis position error.\*SL [axis\\_g\\_aux\\_position](#)
  - /\*314-317*
- G axis auxiliary position.\*SL [axis\\_g\\_velocity](#)
  - /\*318-321*
- G axis velocity.\*SL [axis\\_g\\_torque](#)
  - /\*322-325*
- G axis torque.\*UW [axis\\_g\\_analog\\_in](#)
  - /\*326-327*
- G axis analog input.\*UB [axis\\_g\\_halls](#)
  - /\*328*
- G Hall Input Status.\*UB [axis\\_g\\_reserved](#)
  - /\*329*
- Reserved.\*SL [axis\\_g\\_variable](#)
  - /\*330-333*
- G User defined variable.\*[] UW [axis\\_h\\_status](#)
  - /\*334-335*
- H axis status.\*UB [axis\\_h\\_switches](#)
  - /\*336*
- H axis switches.\*UB [axis\\_h\\_stop\\_code](#)
  - /\*337*
- H axis stop code.\*SL [axis\\_h\\_reference\\_position](#)
  - /\*338-341*
- H axis reference position.\*SL [axis\\_h\\_motor\\_position](#)
  - /\*342-345*
- H axis motor position.\*SL [axis\\_h\\_position\\_error](#)
  - /\*346-349*
- H axis position error.\*SL [axis\\_h\\_aux\\_position](#)
  - /\*350-353*
- H axis auxiliary position.\*SL [axis\\_h\\_velocity](#)
  - /\*354-357*
- H axis velocity.\*SL [axis\\_h\\_torque](#)
  - /\*358-361*
- H axis torque.\*UW [axis\\_h\\_analog\\_in](#)
  - /\*362-363*
- H axis analog input.\*UB [axis\\_h\\_halls](#)
  - /\*364*
- H Hall Input Status.\*UB [axis\\_h\\_reserved](#)
  - /\*365*
- Reserved.\*SL [axis\\_h\\_variable](#)
  - /\*366-369*

### 12.29.1 Detailed Description

Definition at line 472 of file gclib.vb.

## 12.29.2 Member Function Documentation

### 12.29.2.1 byte\_array()

H User defined variable.\* [] byte\_array ( )

Implements [GDataRecord.byte\\_array](#)

Implements [Gclib.GDataRecord](#).

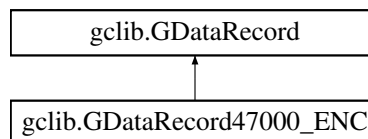
The documentation for this struct was generated from the following file:

- [gclib.vb](#)

## 12.30 gclib.GDataRecord47000\_ENC Struct Reference

Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields.

Inheritance diagram for [gclib.GDataRecord47000\\_ENC](#):



### Public Member Functions

- `byte[] byte_array ( )`  
Returns the data record as a byte array and allows for access to individual bytes.

### Data Fields

- UB [header\\_0](#)  
1st Byte of Header.
- UB [header\\_1](#)  
2nd Byte of Header.
- UB [header\\_2](#)  
3rd Byte of Header.
- UB [header\\_3](#)  
4th Byte of Header.
- UW [sample\\_number](#)  
Sample number.
- UB [error\\_code](#)  
Error code.
- UB [general\\_status](#)  
General status.
- UW [output\\_analog\\_0](#)  
Analog output 0.
- UW [output\\_analog\\_1](#)  
Analog output 1.
- UW [output\\_analog\\_2](#)  
Analog output 2.
- UW [output\\_analog\\_3](#)  
Analog output 3.
- UW [output\\_analog\\_4](#)



- Analog output 4.*
- UW [output\\_analog\\_5](#)  
*Analog output 5.*
- UW [output\\_analog\\_6](#)  
*Analog output 6.*
- UW [output\\_analog\\_7](#)  
*Analog output 7.*
- UW [input\\_analog\\_0](#)  
*Analog input 0.*
- UW [input\\_analog\\_1](#)  
*Analog input 1.*
- UW [input\\_analog\\_2](#)  
*Analog input 2.*
- UW [input\\_analog\\_3](#)  
*Analog input 3.*
- UW [input\\_analog\\_4](#)  
*Analog input 4.*
- UW [input\\_analog\\_5](#)  
*Analog input 5.*
- UW [input\\_analog\\_6](#)  
*Analog input 6.*
- UW [input\\_analog\\_7](#)  
*Analog input 7.*
- UW [output\\_bank\\_0](#)  
*Digital outputs 0-15;.*
- UW [input\\_bank\\_0](#)  
*Digital inputs 0-15;.*
- UL [pulse\\_count\\_0](#)  
*Pulse counter (see PC).*
- SL [zc\\_variable](#)  
*ZC User-defined variable (see ZC).*
- SL [zd\\_variable](#)  
*ZD User-defined variable (see ZD).*
- SL [encoder\\_0](#)  
*Encoder channel 0. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_1](#)  
*Encoder channel 1. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_2](#)  
*Encoder channel 2. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_3](#)  
*Encoder channel 3. Data only valid for parts with -BISS, -QUAD, or -SSI.*

### 12.30.1 Detailed Description

Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields.

Definition at line 1762 of file gclib.cs.

The documentation for this struct was generated from the following file:

- [gclib.cs](#)

## 12.31 GDataRecord47000\_ENC Struct Reference

Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields.

```
#include <gclib_record.h>
```

### Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*Sample number.*
- UB [error\\_code](#)  
*Error code.*
- UB [general\\_status](#)  
*General status.*
- UW [output\\_analog\\_0](#)  
*Analog output 0.*
- UW [output\\_analog\\_1](#)  
*Analog output 1.*
- UW [output\\_analog\\_2](#)  
*Analog output 2.*
- UW [output\\_analog\\_3](#)  
*Analog output 3.*
- UW [output\\_analog\\_4](#)  
*Analog output 4.*
- UW [output\\_analog\\_5](#)  
*Analog output 5.*
- UW [output\\_analog\\_6](#)  
*Analog output 6.*
- UW [output\\_analog\\_7](#)  
*Analog output 7.*
- UW [input\\_analog\\_0](#)  
*Analog input 0.*
- UW [input\\_analog\\_1](#)  
*Analog input 1.*
- UW [input\\_analog\\_2](#)  
*Analog input 2.*
- UW [input\\_analog\\_3](#)  
*Analog input 3.*
- UW [input\\_analog\\_4](#)  
*Analog input 4.*
- UW [input\\_analog\\_5](#)  
*Analog input 5.*
- UW [input\\_analog\\_6](#)  
*Analog input 6.*
- UW [input\\_analog\\_7](#)

- Analog input 7.*
- UW [output\\_bank\\_0](#)
  - Digital outputs 0-15;.*
- UW [input\\_bank\\_0](#)
  - Digital inputs 0-15;.*
- UL [pulse\\_count\\_0](#)
  - Pulse counter (see PC).*
- SL [zc\\_variable](#)
  - ZC User-defined variable (see ZC).*
- SL [zd\\_variable](#)
  - ZD User-defined variable (see ZD).*
- SL [encoder\\_0](#)
  - Encoder channel 0. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_1](#)
  - Encoder channel 1. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_2](#)
  - Encoder channel 2. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_3](#)
  - Encoder channel 3. Data only valid for parts with -BISS, -QUAD, or -SSI.*

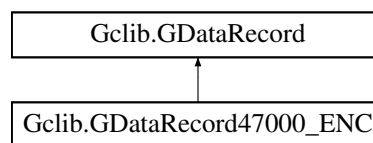
### 12.31.1 Detailed Description

Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields.  
 Definition at line 870 of file `gclib_record.h`.  
 The documentation for this struct was generated from the following file:

- [gclib\\_record.h](#)

## 12.32 Gclib.GDataRecord47000\_ENC Struct Reference

Inheritance diagram for Gclib.GDataRecord47000\_ENC:



### Public Member Functions

- Pulse [counter](#) (see PC). `*/public SL zc_variable`  
*/\*48-51*
- ZC User defined [variable](#) (see ZC). `*/public SL zd_variable`  
*/\*52-55*
- ZD User defined [variable](#) (see ZD). `*/public SL encoder_0`  
*/\*56-59*
- Encoder channel. Data only valid for parts with or SSI.\* `byte_array ()`

## Data Fields

- UB [header\\_0](#)  
*/\*00*
- byte of Header.\*UB [header\\_1](#)  
*/\*01*
- byte of Header.\*UB [header\\_2](#)  
*/\*02*
- byte of Header.\*UB [header\\_3](#)  
*/\*03*
- byte of Header.\*UW [sample\\_number](#)  
*/\*04-05*
- Sample number.\*UB [error\\_code](#)  
*/\*06*
- Error code.\*UB [general\\_status](#)  
*/\*07*
- General status.\*UW [output\\_analog\\_0](#)  
*/\*08-09*
- Analog output.\*UW [output\\_analog\\_1](#)  
*/\*10-11*
- Analog output.\*UW [output\\_analog\\_2](#)  
*/\*12-13*
- Analog output.\*UW [output\\_analog\\_3](#)  
*/\*14-15*
- Analog output.\*UW [output\\_analog\\_4](#)  
*/\*16-17*
- Analog output.\*UW [output\\_analog\\_5](#)  
*/\*18-19*
- Analog output.\*UW [output\\_analog\\_6](#)  
*/\*20-21*
- Analog output.\*UW [output\\_analog\\_7](#)  
*/\*22-23*
- Analog output.\*UW [input\\_analog\\_0](#)  
*/\*24-25*
- Analog input.\*UW [input\\_analog\\_1](#)  
*/\*26-27*
- Analog input.\*UW [input\\_analog\\_2](#)  
*/\*28-29*
- Analog input.\*UW [input\\_analog\\_3](#)  
*/\*30-31*
- Analog input.\*UW [input\\_analog\\_4](#)  
*/\*32-33*
- Analog input.\*UW [input\\_analog\\_5](#)  
*/\*34-35*
- Analog input.\*UW [input\\_analog\\_6](#)  
*/\*36-37*
- Analog input.\*UW [input\\_analog\\_7](#)  
*/\*38-39*
- Analog input.\*UW [output\\_bank\\_0](#)  
*/\*40-41*
- Digital **outputs**

- [\\*UW input\\_bank\\_0](#)  
/\*42-43
- Digital **inputs**
- [\\*UL pulse\\_count\\_0](#)  
/\*44-47
- Encoder channel.Data only valid for parts with **BISS**
- Encoder channel.Data only valid for parts with **QUAD**
- Encoder channel.Data only valid for parts with or SSI.\*SL [encoder\\_1](#)  
/\*60-63
- Encoder channel.Data only valid for parts with or SSI.\*SL [encoder\\_2](#)  
/\*64-67
- Encoder channel.Data only valid for parts with or SSI.\*SL [encoder\\_3](#)  
/\*68-71

### 12.32.1 Detailed Description

Definition at line 1193 of file gclib.vb.

### 12.32.2 Member Function Documentation

#### 12.32.2.1 byte\_array()

Encoder channel.Data only valid for parts with or SSI.\* `byte_array ( )`

Implements [GDataRecord.byte\\_array](#)

Implements [Gclib.GDataRecord](#).

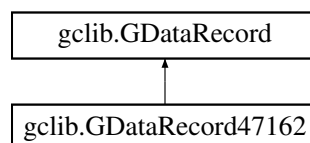
The documentation for this struct was generated from the following file:

- [gclib.vb](#)

## 12.33 gclib.GDataRecord47162 Struct Reference

Data record struct for RIO-47162.

Inheritance diagram for gclib.GDataRecord47162:



### Public Member Functions

- `byte[]` [byte\\_array \( \)](#)  
*Returns the data record as a byte array and allows for access to individual bytes.*

### Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)

- *3rd Byte of Header.*  
UB [header\\_3](#)
- *4th Byte of Header.*  
UW [sample\\_number](#)  
*Sample number.*
- UB [error\\_code](#)  
*Error code.*
- UB [general\\_status](#)  
*General status.*
- UW [output\\_analog\\_0](#)  
*Analog output 0.*
- UW [output\\_analog\\_1](#)  
*Analog output 1.*
- UW [output\\_analog\\_2](#)  
*Analog output 2.*
- UW [output\\_analog\\_3](#)  
*Analog output 3.*
- UW [output\\_analog\\_4](#)  
*Analog output 4.*
- UW [output\\_analog\\_5](#)  
*Analog output 5.*
- UW [output\\_analog\\_6](#)  
*Analog output 6.*
- UW [output\\_analog\\_7](#)  
*Analog output 7.*
- UW [input\\_analog\\_0](#)  
*Analog input 0.*
- UW [input\\_analog\\_1](#)  
*Analog input 1.*
- UW [input\\_analog\\_2](#)  
*Analog input 2.*
- UW [input\\_analog\\_3](#)  
*Analog input 3.*
- UW [input\\_analog\\_4](#)  
*Analog input 4.*
- UW [input\\_analog\\_5](#)  
*Analog input 5.*
- UW [input\\_analog\\_6](#)  
*Analog input 6.*
- UW [input\\_analog\\_7](#)  
*Analog input 7.*
- UB [output\\_byte\\_0](#)  
*Digital outputs 0-7.*
- UB [output\\_byte\\_1](#)  
*Digital outputs 8-15.*
- UB [output\\_byte\\_2](#)  
*Digital outputs 16-23.*
- UB [input\\_byte\\_0](#)  
*Digital inputs 0-7.*
- UB [input\\_byte\\_1](#)  
*Digital inputs 8-15.*

- UB [input\\_byte\\_2](#)  
*Digital inputs 16-23.*
- UB [input\\_byte\\_3](#)  
*Digital inputs 24-31.*
- UB [input\\_byte\\_4](#)  
*Digital inputs 32-39.*
- UL [pulse\\_count\\_0](#)  
*Pulse counter (see PC).*
- SL [zc\\_variable](#)  
*ZC User-defined variable (see ZC).*
- SL [zd\\_variable](#)  
*ZD User-defined variable (see ZD).*
- SL [encoder\\_0](#)  
*Encoder channel 0. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_1](#)  
*Encoder channel 1. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_2](#)  
*Encoder channel 2. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_3](#)  
*Encoder channel 3. Data only valid for parts with -BISS, -QUAD, or -SSI.*

### 12.33.1 Detailed Description

Data record struct for RIO-47162.

Definition at line 1917 of file `gclib.cs`.

The documentation for this struct was generated from the following file:

- [gclib.cs](#)

## 12.34 GDataRecord47162 Struct Reference

Data record struct for RIO-47162.

```
#include <gclib_record.h>
```

### Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*Sample number.*
- UB [error\\_code](#)  
*Error code.*
- UB [general\\_status](#)  
*General status.*
- UW [output\\_analog\\_0](#)  
*Analog output 0.*

- UW [output\\_analog\\_1](#)  
*Analog output 1.*
- UW [output\\_analog\\_2](#)  
*Analog output 2.*
- UW [output\\_analog\\_3](#)  
*Analog output 3.*
- UW [output\\_analog\\_4](#)  
*Analog output 4.*
- UW [output\\_analog\\_5](#)  
*Analog output 5.*
- UW [output\\_analog\\_6](#)  
*Analog output 6.*
- UW [output\\_analog\\_7](#)  
*Analog output 7.*
- UW [input\\_analog\\_0](#)  
*Analog input 0.*
- UW [input\\_analog\\_1](#)  
*Analog input 1.*
- UW [input\\_analog\\_2](#)  
*Analog input 2.*
- UW [input\\_analog\\_3](#)  
*Analog input 3.*
- UW [input\\_analog\\_4](#)  
*Analog input 4.*
- UW [input\\_analog\\_5](#)  
*Analog input 5.*
- UW [input\\_analog\\_6](#)  
*Analog input 6.*
- UW [input\\_analog\\_7](#)  
*Analog input 7.*
- UB [output\\_byte\\_0](#)  
*Digital outputs 0-7.*
- UB [output\\_byte\\_1](#)  
*Digital outputs 8-15.*
- UB [output\\_byte\\_2](#)  
*Digital outputs 16-23.*
- UB [input\\_byte\\_0](#)  
*Digital inputs 0-7.*
- UB [input\\_byte\\_1](#)  
*Digital inputs 8-15.*
- UB [input\\_byte\\_2](#)  
*Digital inputs 16-23.*
- UB [input\\_byte\\_3](#)  
*Digital inputs 24-31.*
- UB [input\\_byte\\_4](#)  
*Digital inputs 32-39.*
- UL [pulse\\_count\\_0](#)  
*Pulse counter (see PC).*
- SL [zc\\_variable](#)  
*ZC User-defined variable (see ZC).*
- SL [zd\\_variable](#)



- ZD User-defined variable (see ZD).*
- SL [encoder\\_0](#)  
*Encoder channel 0. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_1](#)  
*Encoder channel 1. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_2](#)  
*Encoder channel 2. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- SL [encoder\\_3](#)  
*Encoder channel 3. Data only valid for parts with -BISS, -QUAD, or -SSI.*

### 12.34.1 Detailed Description

Data record struct for RIO-47162.

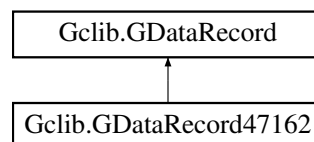
Definition at line 1019 of file `gclib_record.h`.

The documentation for this struct was generated from the following file:

- [gclib\\_record.h](#)

## 12.35 Gclib.GDataRecord47162 Struct Reference

Inheritance diagram for Gclib.GDataRecord47162:



### Public Member Functions

- Pulse [counter](#) (see PC). `*/public SL zc_variable`  
*/\*52-55*
- ZC User defined [variable](#) (see ZC). `*/public SL zd_variable`  
*/\*56-59*
- ZD User defined [variable](#) (see ZD). `*/public SL encoder_0`  
*/\*60-63*
- Encoder channel. Data only valid for parts with or SSI. `* byte_array ()`

### Data Fields

- UB [header\\_0](#)  
*/\*00*
- byte of Header. `*UB header_1`  
*/\*01*
- byte of Header. `*UB header_2`  
*/\*02*
- byte of Header. `*UB header_3`  
*/\*03*
- byte of Header. `*UW sample_number`  
*/\*04-05*
- Sample number. `*UB error_code`  
*/\*06*
- Error code. `*UB general_status`

- /\*07*
  - General status.\*UW [output\\_analog\\_0](#)
- /\*08-09*
  - Analog output.\*UW [output\\_analog\\_1](#)
- /\*10-11*
  - Analog output.\*UW [output\\_analog\\_2](#)
- /\*12-13*
  - Analog output.\*UW [output\\_analog\\_3](#)
- /\*14-15*
  - Analog output.\*UW [output\\_analog\\_4](#)
- /\*16-17*
  - Analog output.\*UW [output\\_analog\\_5](#)
- /\*18-19*
  - Analog output.\*UW [output\\_analog\\_6](#)
- /\*20-21*
  - Analog output.\*UW [output\\_analog\\_7](#)
- /\*22-23*
  - Analog output.\*UW [input\\_analog\\_0](#)
- /\*24-25*
  - Analog input.\*UW [input\\_analog\\_1](#)
- /\*26-27*
  - Analog input.\*UW [input\\_analog\\_2](#)
- /\*28-29*
  - Analog input.\*UW [input\\_analog\\_3](#)
- /\*30-31*
  - Analog input.\*UW [input\\_analog\\_4](#)
- /\*32-33*
  - Analog input.\*UW [input\\_analog\\_5](#)
- /\*34-35*
  - Analog input.\*UW [input\\_analog\\_6](#)
- /\*36-37*
  - Analog input.\*UW [input\\_analog\\_7](#)
- /\*38-39*
  - Analog input.\*UB [output\\_byte\\_0](#)
- /\*40*
  - Digital outputs.\*UB [output\\_byte\\_1](#)
- /\*41*
  - Digital outputs.\*UB [output\\_byte\\_2](#)
- /\*42*
  - Digital outputs.\*UB [input\\_byte\\_0](#)
- /\*43*
  - Digital inputs.\*UB [input\\_byte\\_1](#)
- /\*44*
  - Digital inputs.\*UB [input\\_byte\\_2](#)
- /\*45*
  - Digital inputs.\*UB [input\\_byte\\_3](#)
- /\*46*
  - Digital inputs.\*UB [input\\_byte\\_4](#)
- /\*47*
  - Digital inputs.\*UL [pulse\\_count\\_0](#)
- /\*48-51*

- Encoder channel.Data only valid for parts with **BISS**
- Encoder channel.Data only valid for parts with **QUAD**
- Encoder channel.Data only valid for parts with or SSI.\*SL [encoder\\_1](#)  
/\*64-67
- Encoder channel.Data only valid for parts with or SSI.\*SL [encoder\\_2](#)  
/\*68-71
- Encoder channel.Data only valid for parts with or SSI.\*SL [encoder\\_3](#)  
/\*72-75

### 12.35.1 Detailed Description

Definition at line 1308 of file gclib.vb.

### 12.35.2 Member Function Documentation

#### 12.35.2.1 byte\_array()

Encoder channel.Data only valid for parts with or SSI.\* byte\_array ( )

Implements [GDataRecord.byte\\_array](#)

Implements [Gclib.GDataRecord](#).

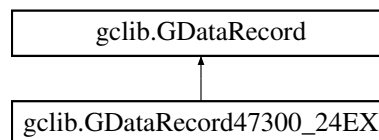
The documentation for this struct was generated from the following file:

- [gclib.vb](#)

## 12.36 gclib.GDataRecord47300\_24EX Struct Reference

Data record struct for RIO-47300 with 24EX I/O daughter board.

Inheritance diagram for gclib.GDataRecord47300\_24EX:



### Public Member Functions

- [byte\[\] byte\\_array \( \)](#)  
*Returns the data record as a byte array and allows for access to individual bytes.*

### Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*Sample number.*

- UB [error\\_code](#)  
*Error code.*
- UB [general\\_status](#)  
*General status.*
- UW [output\\_analog\\_0](#)  
*Analog output 0.*
- UW [output\\_analog\\_1](#)  
*Analog output 1.*
- UW [output\\_analog\\_2](#)  
*Analog output 2.*
- UW [output\\_analog\\_3](#)  
*Analog output 3.*
- UW [output\\_analog\\_4](#)  
*Analog output 4.*
- UW [output\\_analog\\_5](#)  
*Analog output 5.*
- UW [output\\_analog\\_6](#)  
*Analog output 6.*
- UW [output\\_analog\\_7](#)  
*Analog output 7.*
- UW [input\\_analog\\_0](#)  
*Analog input 0.*
- UW [input\\_analog\\_1](#)  
*Analog input 1.*
- UW [input\\_analog\\_2](#)  
*Analog input 2.*
- UW [input\\_analog\\_3](#)  
*Analog input 3.*
- UW [input\\_analog\\_4](#)  
*Analog input 4.*
- UW [input\\_analog\\_5](#)  
*Analog input 5.*
- UW [input\\_analog\\_6](#)  
*Analog input 6.*
- UW [input\\_analog\\_7](#)  
*Analog input 7.*
- UW [output\\_bank\\_0](#)  
*Digital outputs 0-15.*
- UW [output\\_bank\\_1](#)  
*Digital outputs 16-23.*
- UW [input\\_bank\\_0](#)  
*Digital inputs 0-15.*
- UW [input\\_bank\\_1](#)  
*Digital inputs 16-23.*
- UL [pulse\\_count\\_0](#)  
*Pulse counter (see PC)8.*
- SL [zc\\_variable](#)  
*ZC User-defined variable (see ZC).*
- SL [zd\\_variable](#)  
*ZD User-defined variable (see ZD).*
- UW [output\\_bank\\_2](#)

- [UW output\\_back\\_3](#)  
*Digital outputs 24-39. Data only valid for parts with 24EXOUT.*
- [UW input\\_bank\\_2](#)  
*Digital inputs 24-39. Data only valid for parts with 24EXIN.*
- [UW input\\_bank\\_3](#)  
*Digital inputs 40-47. Data only valid for parts with 24EXIN.*

### 12.36.1 Detailed Description

Data record struct for RIO-47300 with 24EX I/O daughter board.

Definition at line 1864 of file `gclib.cs`.

The documentation for this struct was generated from the following file:

- [gclib.cs](#)

## 12.37 GDataRecord47300\_24EX Struct Reference

Data record struct for RIO-47300 with 24EX I/O daughter board.

```
#include <gclib_record.h>
```

### Data Fields

- [UB header\\_0](#)  
*1st Byte of Header.*
- [UB header\\_1](#)  
*2nd Byte of Header.*
- [UB header\\_2](#)  
*3rd Byte of Header.*
- [UB header\\_3](#)  
*4th Byte of Header.*
- [UW sample\\_number](#)  
*Sample number.*
- [UB error\\_code](#)  
*Error code.*
- [UB general\\_status](#)  
*General status.*
- [UW output\\_analog\\_0](#)  
*Analog output 0.*
- [UW output\\_analog\\_1](#)  
*Analog output 1.*
- [UW output\\_analog\\_2](#)  
*Analog output 2.*
- [UW output\\_analog\\_3](#)  
*Analog output 3.*
- [UW output\\_analog\\_4](#)  
*Analog output 4.*
- [UW output\\_analog\\_5](#)  
*Analog output 5.*
- [UW output\\_analog\\_6](#)  
*Analog output 6.*
- [UW output\\_analog\\_7](#)

- Analog output 7.*
- [UW input\\_analog\\_0](#)
  - Analog input 0.*
- [UW input\\_analog\\_1](#)
  - Analog input 1.*
- [UW input\\_analog\\_2](#)
  - Analog input 2.*
- [UW input\\_analog\\_3](#)
  - Analog input 3.*
- [UW input\\_analog\\_4](#)
  - Analog input 4.*
- [UW input\\_analog\\_5](#)
  - Analog input 5.*
- [UW input\\_analog\\_6](#)
  - Analog input 6.*
- [UW input\\_analog\\_7](#)
  - Analog input 7.*
- [UW output\\_bank\\_0](#)
  - Digital outputs 0-15.*
- [UW output\\_bank\\_1](#)
  - Digital outputs 16-23.*
- [UW input\\_bank\\_0](#)
  - Digital inputs 0-15.*
- [UW input\\_bank\\_1](#)
  - Digital inputs 16-23.*
- [UL pulse\\_count\\_0](#)
  - Pulse counter (see PC)8.*
- [SL zc\\_variable](#)
  - ZC User-defined variable (see ZC).*
- [SL zd\\_variable](#)
  - ZD User-defined variable (see ZD).*
- [UW output\\_bank\\_2](#)
  - Digital outputs 24-39. Data only valid for parts with 24EXOUT.*
- [UW output\\_back\\_3](#)
  - Digital outputs 40-47. Data only valid for parts with 24EXOUT.*
- [UW input\\_bank\\_2](#)
  - Digital inputs 24-39. Data only valid for parts with 24EXIN.*
- [UW input\\_bank\\_3](#)
  - Digital inputs 40-47. Data only valid for parts with 24EXIN.*

### 12.37.1 Detailed Description

Data record struct for RIO-47300 with 24EX I/O daughter board.

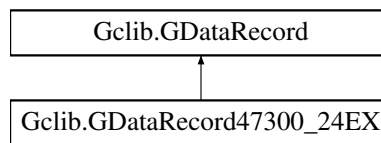
Definition at line 968 of file `gclib_record.h`.

The documentation for this struct was generated from the following file:

- [gclib\\_record.h](#)

## 12.38 Gclib.GDataRecord47300\_24EX Struct Reference

Inheritance diagram for Gclib.GDataRecord47300\_24EX:



### Public Member Functions

- Pulse [counter](#) (see PC) 8. `*/public SL zc_variable`  
`/*52-55`
- ZC User defined [variable](#) (see ZC). `*/public SL zd_variable`  
`/*56-59`
- ZD User defined [variable](#) (see ZD). `*/public UW output_bank_2`  
`/*60-61`
- Digital inputs.Data only valid for parts with.\* [byte\\_array](#) ()

### Data Fields

- UB [header\\_0](#)  
`/*00`
- byte of Header.\*UB [header\\_1](#)  
`/*01`
- byte of Header.\*UB [header\\_2](#)  
`/*02`
- byte of Header.\*UB [header\\_3](#)  
`/*03`
- byte of Header.\*UW [sample\\_number](#)  
`/*04-05`
- Sample number.\*UB [error\\_code](#)  
`/*06`
- Error code.\*UB [general\\_status](#)  
`/*07`
- General status.\*UW [output\\_analog\\_0](#)  
`/*08-09`
- Analog output.\*UW [output\\_analog\\_1](#)  
`/*10-11`
- Analog output.\*UW [output\\_analog\\_2](#)  
`/*12-13`
- Analog output.\*UW [output\\_analog\\_3](#)  
`/*14-15`
- Analog output.\*UW [output\\_analog\\_4](#)  
`/*16-17`
- Analog output.\*UW [output\\_analog\\_5](#)  
`/*18-19`
- Analog output.\*UW [output\\_analog\\_6](#)  
`/*20-21`
- Analog output.\*UW [output\\_analog\\_7](#)  
`/*22-23`

- Analog output.\*UW [input\\_analog\\_0](#)  
/\*24-25
- Analog input.\*UW [input\\_analog\\_1](#)  
/\*26-27
- Analog input.\*UW [input\\_analog\\_2](#)  
/\*28-29
- Analog input.\*UW [input\\_analog\\_3](#)  
/\*30-31
- Analog input.\*UW [input\\_analog\\_4](#)  
/\*32-33
- Analog input.\*UW [input\\_analog\\_5](#)  
/\*34-35
- Analog input.\*UW [input\\_analog\\_6](#)  
/\*36-37
- Analog input.\*UW [input\\_analog\\_7](#)  
/\*38-39
- Analog input.\*UW [output\\_bank\\_0](#)  
/\*40-41
- Digital outputs.\*UW [output\\_bank\\_1](#)  
/\*42-43
- Digital outputs.\*UW [input\\_bank\\_0](#)  
/\*44-45
- Digital inputs.\*UW [input\\_bank\\_1](#)  
/\*46-47
- Digital inputs.\*UL [pulse\\_count\\_0](#)  
/\*48-51
- Digital outputs.Data only valid for parts with.\*UW [output\\_bank\\_3](#)  
/\*62-63
- Digital outputs.Data only valid for parts with.\*UW [input\\_bank\\_2](#)  
/\*64-65
- Digital inputs.Data only valid for parts with.\*UW [input\\_bank\\_3](#)  
/\*66-67

### 12.38.1 Detailed Description

Definition at line 1269 of file gclib.vb.

### 12.38.2 Member Function Documentation

#### 12.38.2.1 `byte_array()`

Digital inputs.Data only valid for parts with.\* `byte_array ( )`

Implements [GDataRecord.byte\\_array](#)

Implements [Gclib.GDataRecord](#).

The documentation for this struct was generated from the following file:

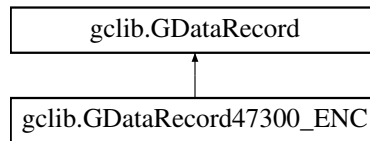
- [gclib.vb](#)



## 12.39 gclib.GDataRecord47300\_ENC Struct Reference

Data record struct for RIO-47300. Includes encoder fields.

Inheritance diagram for gclib.GDataRecord47300\_ENC:



### Public Member Functions

- `byte[] byte_array ()`  
*Returns the data record as a byte array and allows for access to individual bytes.*

### Data Fields

- UB `header_0`  
*1st Byte of Header.*
- UB `header_1`  
*2nd Byte of Header.*
- UB `header_2`  
*3rd Byte of Header.*
- UB `header_3`  
*4th Byte of Header.*
- UW `sample_number`  
*Sample number.*
- UB `error_code`  
*Error code.*
- UB `general_status`  
*General status.*
- UW `output_analog_0`  
*Analog output 0.*
- UW `output_analog_1`  
*Analog output 1.*
- UW `output_analog_2`  
*Analog output 2.*
- UW `output_analog_3`  
*Analog output 3.*
- UW `output_analog_4`  
*Analog output 4.*
- UW `output_analog_5`  
*Analog output 5.*
- UW `output_analog_6`  
*Analog output 6.*
- UW `output_analog_7`  
*Analog output 7.*
- UW `input_analog_0`  
*Analog input 0.*
- UW `input_analog_1`  
*Analog input 1.*

- [UW input\\_analog\\_2](#)  
*Analog input 2.*
- [UW input\\_analog\\_3](#)  
*Analog input 3.*
- [UW input\\_analog\\_4](#)  
*Analog input 4.*
- [UW input\\_analog\\_5](#)  
*Analog input 5.*
- [UW input\\_analog\\_6](#)  
*Analog input 6.*
- [UW input\\_analog\\_7](#)  
*Analog input 7.*
- [UW output\\_bank\\_0](#)  
*Digital outputs 0-15;.*
- [UW output\\_bank\\_1](#)  
*Digital outputs 16-23;.*
- [UW input\\_bank\\_0](#)  
*Digital inputs 0-15;.*
- [UW input\\_bank\\_1](#)  
*Digital inputs 16-23;.*
- [UL pulse\\_count\\_0](#)  
*Pulse counter (see PC).*
- [SL zc\\_variable](#)  
*ZC User-defined variable (see ZC).*
- [SL zd\\_variable](#)  
*ZD User-defined variable (see ZD).*
- [SL encoder\\_0](#)  
*Encoder channel 0. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- [SL encoder\\_1](#)  
*Encoder channel 1. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- [SL encoder\\_2](#)  
*Encoder channel 2. Data only valid for parts with -BISS, -QUAD, or -SSI.*
- [SL encoder\\_3](#)  
*Encoder channel 3. Data only valid for parts with -BISS, -QUAD, or -SSI.*

### 12.39.1 Detailed Description

Data record struct for RIO-47300. Includes encoder fields.

Definition at line 1812 of file gclib.cs.

The documentation for this struct was generated from the following file:

- [gclib.cs](#)

## 12.40 GDataRecord47300\_ENC Struct Reference

Data record struct for RIO-47300. Includes encoder fields.

```
#include <gclib_record.h>
```

## Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*Sample number.*
- UB [error\\_code](#)  
*Error code.*
- UB [general\\_status](#)  
*General status.*
- UW [output\\_analog\\_0](#)  
*Analog output 0.*
- UW [output\\_analog\\_1](#)  
*Analog output 1.*
- UW [output\\_analog\\_2](#)  
*Analog output 2.*
- UW [output\\_analog\\_3](#)  
*Analog output 3.*
- UW [output\\_analog\\_4](#)  
*Analog output 4.*
- UW [output\\_analog\\_5](#)  
*Analog output 5.*
- UW [output\\_analog\\_6](#)  
*Analog output 6.*
- UW [output\\_analog\\_7](#)  
*Analog output 7.*
- UW [input\\_analog\\_0](#)  
*Analog input 0.*
- UW [input\\_analog\\_1](#)  
*Analog input 1.*
- UW [input\\_analog\\_2](#)  
*Analog input 2.*
- UW [input\\_analog\\_3](#)  
*Analog input 3.*
- UW [input\\_analog\\_4](#)  
*Analog input 4.*
- UW [input\\_analog\\_5](#)  
*Analog input 5.*
- UW [input\\_analog\\_6](#)  
*Analog input 6.*
- UW [input\\_analog\\_7](#)  
*Analog input 7.*
- UW [output\\_bank\\_0](#)  
*Digital outputs 0-15;.*
- UW [output\\_bank\\_1](#)



## Data Fields

- UB [header\\_0](#)  
*/\*00*
- byte of Header.\*UB [header\\_1](#)  
*/\*01*
- byte of Header.\*UB [header\\_2](#)  
*/\*02*
- byte of Header.\*UB [header\\_3](#)  
*/\*03*
- byte of Header.\*UW [sample\\_number](#)  
*/\*04-05*
- Sample number.\*UB [error\\_code](#)  
*/\*06*
- Error code.\*UB [general\\_status](#)  
*/\*07*
- General status.\*UW [output\\_analog\\_0](#)  
*/\*08-09*
- Analog output.\*UW [output\\_analog\\_1](#)  
*/\*10-11*
- Analog output.\*UW [output\\_analog\\_2](#)  
*/\*12-13*
- Analog output.\*UW [output\\_analog\\_3](#)  
*/\*14-15*
- Analog output.\*UW [output\\_analog\\_4](#)  
*/\*16-17*
- Analog output.\*UW [output\\_analog\\_5](#)  
*/\*18-19*
- Analog output.\*UW [output\\_analog\\_6](#)  
*/\*20-21*
- Analog output.\*UW [output\\_analog\\_7](#)  
*/\*22-23*
- Analog output.\*UW [input\\_analog\\_0](#)  
*/\*24-25*
- Analog input.\*UW [input\\_analog\\_1](#)  
*/\*26-27*
- Analog input.\*UW [input\\_analog\\_2](#)  
*/\*28-29*
- Analog input.\*UW [input\\_analog\\_3](#)  
*/\*30-31*
- Analog input.\*UW [input\\_analog\\_4](#)  
*/\*32-33*
- Analog input.\*UW [input\\_analog\\_5](#)  
*/\*34-35*
- Analog input.\*UW [input\\_analog\\_6](#)  
*/\*36-37*
- Analog input.\*UW [input\\_analog\\_7](#)  
*/\*38-39*
- Analog input.\*UW [output\\_bank\\_0](#)  
*/\*40-41*
- Digital **outputs**

- [\\*UW output\\_bank\\_1](#)  
/\*42-43
- [\\*UW input\\_bank\\_0](#)  
/\*44-45
- **Digital inputs**
- [\\*UW input\\_bank\\_1](#)  
/\*46-47
- [\\*UL pulse\\_count\\_0](#)  
/\*48-51
- Encoder channel.Data only valid for parts with **BISS**
- Encoder channel.Data only valid for parts with **QUAD**
- Encoder channel.Data only valid for parts with or SSI.\*SL [encoder\\_1](#)  
/\*64-67
- Encoder channel.Data only valid for parts with or SSI.\*SL [encoder\\_2](#)  
/\*68-71
- Encoder channel.Data only valid for parts with or SSI.\*SL [encoder\\_3](#)  
/\*72-75

### 12.41.1 Detailed Description

Definition at line 1230 of file gclib.vb.

### 12.41.2 Member Function Documentation

#### 12.41.2.1 `byte_array()`

Encoder channel.Data only valid for parts with or SSI.\* `byte_array ( )`

Implements [GDataRecord.byte\\_array](#)

Implements [Gclib.GDataRecord](#).

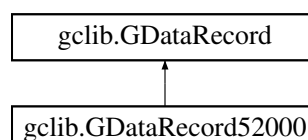
The documentation for this struct was generated from the following file:

- [gclib.vb](#)

## 12.42 `gclib.GDataRecord52000` Struct Reference

Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40.

Inheritance diagram for `gclib.GDataRecord52000`:



### Public Member Functions

- `byte[]` [byte\\_array \( \)](#)

*Returns the data record as a byte array and allows for access to individual bytes.*

## Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*sample number.*
- UB [input\\_bank\\_0](#)  
*general input bank 0 (inputs 1-8).*
- UB [input\\_bank\\_1](#)  
*general input bank 1 (inputs 9-16).*
- UB [input\\_bank\\_2](#)  
*general input bank 2 (inputs 17-24).*
- UB [input\\_bank\\_3](#)  
*general input bank 3 (inputs 25-32).*
- UB [input\\_bank\\_4](#)  
*general input bank 4 (inputs 33-40).*
- UB [input\\_bank\\_5](#)  
*general input bank 5 (inputs 41-48).*
- UB [input\\_bank\\_6](#)  
*general input bank 6 (inputs 49-56).*
- UB [input\\_bank\\_7](#)  
*general input bank 7 (inputs 57-64).*
- UB [input\\_bank\\_8](#)  
*general input bank 8 (inputs 65-72).*
- UB [input\\_bank\\_9](#)  
*general input bank 9 (inputs 73-80).*
- UB [output\\_bank\\_0](#)  
*general output bank 0 (outputs 1-8).*
- UB [output\\_bank\\_1](#)  
*general output bank 1 (outputs 9-16).*
- UB [output\\_bank\\_2](#)  
*general output bank 2 (outputs 17-24).*
- UB [output\\_bank\\_3](#)  
*general output bank 3 (outputs 25-32).*
- UB [output\\_bank\\_4](#)  
*general output bank 4 (outputs 33-40).*
- UB [output\\_bank\\_5](#)  
*general output bank 5 (outputs 41-48).*
- UB [output\\_bank\\_6](#)  
*general output bank 6 (outputs 49-56).*
- UB [output\\_bank\\_7](#)  
*general output bank 7 (outputs 57-64).*
- UB [output\\_bank\\_8](#)  
*general output bank 8 (outputs 65-72).*
- UB [output\\_bank\\_9](#)

- general output bank 9 (outputs 73-80).*
- SW [reserved\\_0](#)  
*Reserved.*
- SW [reserved\\_2](#)  
*Reserved.*
- SW [reserved\\_4](#)  
*Reserved.*
- SW [reserved\\_6](#)  
*Reserved.*
- SW [reserved\\_8](#)  
*Reserved.*
- SW [reserved\\_10](#)  
*Reserved.*
- SW [reserved\\_12](#)  
*Reserved.*
- UB [ethercat\\_bank](#)  
*EtherCAT Bank Indicator.*
- UB [reserved\\_14](#)  
*Reserved.*
- UB [ethernet\\_status\\_a](#)  
*Ethernet Handle A Status.*
- UB [ethernet\\_status\\_b](#)  
*Ethernet Handle B Status.*
- UB [ethernet\\_status\\_c](#)  
*Ethernet Handle C Status.*
- UB [ethernet\\_status\\_d](#)  
*Ethernet Handle D Status.*
- UB [ethernet\\_status\\_e](#)  
*Ethernet Handle E Status.*
- UB [ethernet\\_status\\_f](#)  
*Ethernet Handle F Status.*
- UB [ethernet\\_status\\_g](#)  
*Ethernet Handle G Status.*
- UB [ethernet\\_status\\_h](#)  
*Ethernet Handle H Status.*
- UB [error\\_code](#)  
*error code.*
- UB [thread\\_status](#)  
*thread status*
- UL [amplifier\\_status](#)  
*Amplifier Status.*
- UL [contour\\_segment\\_count](#)  
*Segment Count for Contour Mode.*
- UW [contour\\_buffer\\_available](#)  
*Buffer space remaining, Contour Mode.*
- UW [s\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)  
*coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*



- UW [s\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, S Plane.*
- UW [t\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for T plane.*
- UW [t\\_plane\\_move\\_status](#)  
*Coordinated move status for T plane.*
- SL [t\\_distance](#)  
*distance traveled in coordinated move for T plane.*
- UW [t\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, T Plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)  
*A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*
- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SL [axis\\_a\\_torque](#)  
*A axis torque.*
- UW [axis\\_a\\_analog\\_in](#)  
*A axis analog input.*
- UB [axis\\_a\\_halls](#)  
*A Hall Input Status.*
- UB [axis\\_a\\_reserved](#)  
*Reserved.*
- SL [axis\\_a\\_variable](#)  
*A User-defined variable (ZA).*
- UW [axis\\_b\\_status](#)  
*B axis status.*
- UB [axis\\_b\\_switches](#)  
*B axis switches.*
- UB [axis\\_b\\_stop\\_code](#)  
*B axis stop code.*
- SL [axis\\_b\\_reference\\_position](#)  
*B axis reference position.*
- SL [axis\\_b\\_motor\\_position](#)  
*B axis motor position.*
- SL [axis\\_b\\_position\\_error](#)  
*B axis position error.*
- SL [axis\\_b\\_aux\\_position](#)  
*B axis auxiliary position.*
- SL [axis\\_b\\_velocity](#)

- B axis velocity.*
- SL [axis\\_b\\_torque](#)
  - B axis torque.*
- UW [axis\\_b\\_analog\\_in](#)
  - B axis analog input.*
- UB [axis\\_b\\_halls](#)
  - B Hall Input Status.*
- UB [axis\\_b\\_reserved](#)
  - Reserved.*
- SL [axis\\_b\\_variable](#)
  - B User-defined variable (ZA).*
- UW [axis\\_c\\_status](#)
  - C axis status.*
- UB [axis\\_c\\_switches](#)
  - C axis switches.*
- UB [axis\\_c\\_stop\\_code](#)
  - C axis stop code.*
- SL [axis\\_c\\_reference\\_position](#)
  - C axis reference position.*
- SL [axis\\_c\\_motor\\_position](#)
  - C axis motor position.*
- SL [axis\\_c\\_position\\_error](#)
  - C axis position error.*
- SL [axis\\_c\\_aux\\_position](#)
  - C axis auxiliary position.*
- SL [axis\\_c\\_velocity](#)
  - C axis velocity.*
- SL [axis\\_c\\_torque](#)
  - C axis torque.*
- UW [axis\\_c\\_analog\\_in](#)
  - C axis analog input.*
- UB [axis\\_c\\_halls](#)
  - C Hall Input Status.*
- UB [axis\\_c\\_reserved](#)
  - Reserved.*
- SL [axis\\_c\\_variable](#)
  - C User-defined variable (ZA).*
- UW [axis\\_d\\_status](#)
  - D axis status.*
- UB [axis\\_d\\_switches](#)
  - D axis switches.*
- UB [axis\\_d\\_stop\\_code](#)
  - D axis stop code.*
- SL [axis\\_d\\_reference\\_position](#)
  - D axis reference position.*
- SL [axis\\_d\\_motor\\_position](#)
  - D axis motor position.*
- SL [axis\\_d\\_position\\_error](#)
  - D axis position error.*
- SL [axis\\_d\\_aux\\_position](#)
  - D axis auxiliary position.*

- SL [axis\\_d\\_velocity](#)  
*D axis velocity.*
- SL [axis\\_d\\_torque](#)  
*D axis torque.*
- UW [axis\\_d\\_analog\\_in](#)  
*D axis analog input.*
- UB [axis\\_d\\_halls](#)  
*D Hall Input Status.*
- UB [axis\\_d\\_reserved](#)  
*Reserved.*
- SL [axis\\_d\\_variable](#)  
*D User-defined variable (ZA).*
- UW [axis\\_e\\_status](#)  
*E axis status.*
- UB [axis\\_e\\_switches](#)  
*E axis switches.*
- UB [axis\\_e\\_stop\\_code](#)  
*E axis stop code.*
- SL [axis\\_e\\_reference\\_position](#)  
*E axis reference position.*
- SL [axis\\_e\\_motor\\_position](#)  
*E axis motor position.*
- SL [axis\\_e\\_position\\_error](#)  
*E axis position error.*
- SL [axis\\_e\\_aux\\_position](#)  
*E axis auxiliary position.*
- SL [axis\\_e\\_velocity](#)  
*E axis velocity.*
- SL [axis\\_e\\_torque](#)  
*E axis torque.*
- UW [axis\\_e\\_analog\\_in](#)  
*E axis analog input.*
- UB [axis\\_e\\_halls](#)  
*E Hall Input Status.*
- UB [axis\\_e\\_reserved](#)  
*Reserved.*
- SL [axis\\_e\\_variable](#)  
*E User-defined variable (ZA).*
- UW [axis\\_f\\_status](#)  
*F axis status.*
- UB [axis\\_f\\_switches](#)  
*F axis switches.*
- UB [axis\\_f\\_stop\\_code](#)  
*F axis stop code.*
- SL [axis\\_f\\_reference\\_position](#)  
*F axis reference position.*
- SL [axis\\_f\\_motor\\_position](#)  
*F axis motor position.*
- SL [axis\\_f\\_position\\_error](#)  
*F axis position error.*
- SL [axis\\_f\\_aux\\_position](#)

- F axis auxiliary position.*
- SL [axis\\_f\\_velocity](#)  
*F axis velocity.*
- SL [axis\\_f\\_torque](#)  
*F axis torque.*
- UW [axis\\_f\\_analog\\_in](#)  
*F axis analog input.*
- UB [axis\\_f\\_halls](#)  
*F Hall Input Status.*
- UB [axis\\_f\\_reserved](#)  
*Reserved.*
- SL [axis\\_f\\_variable](#)  
*F User-defined variable (ZA).*
- UW [axis\\_g\\_status](#)  
*G axis status.*
- UB [axis\\_g\\_switches](#)  
*G axis switches.*
- UB [axis\\_g\\_stop\\_code](#)  
*G axis stop code.*
- SL [axis\\_g\\_reference\\_position](#)  
*G axis reference position.*
- SL [axis\\_g\\_motor\\_position](#)  
*G axis motor position.*
- SL [axis\\_g\\_position\\_error](#)  
*G axis position error.*
- SL [axis\\_g\\_aux\\_position](#)  
*G axis auxiliary position.*
- SL [axis\\_g\\_velocity](#)  
*G axis velocity.*
- SL [axis\\_g\\_torque](#)  
*G axis torque.*
- UW [axis\\_g\\_analog\\_in](#)  
*G axis analog input.*
- UB [axis\\_g\\_halls](#)  
*G Hall Input Status.*
- UB [axis\\_g\\_reserved](#)  
*Reserved.*
- SL [axis\\_g\\_variable](#)  
*G User-defined variable (ZA).*
- UW [axis\\_h\\_status](#)  
*H axis status.*
- UB [axis\\_h\\_switches](#)  
*H axis switches.*
- UB [axis\\_h\\_stop\\_code](#)  
*H axis stop code.*
- SL [axis\\_h\\_reference\\_position](#)  
*H axis reference position.*
- SL [axis\\_h\\_motor\\_position](#)  
*H axis motor position.*
- SL [axis\\_h\\_position\\_error](#)  
*H axis position error.*

- SL [axis\\_h\\_aux\\_position](#)  
*H axis auxiliary position.*
- SL [axis\\_h\\_velocity](#)  
*H axis velocity.*
- SL [axis\\_h\\_torque](#)  
*H axis torque.*
- UW [axis\\_h\\_analog\\_in](#)  
*H axis analog input.*
- UB [axis\\_h\\_halls](#)  
*H Hall Input Status.*
- UB [axis\\_h\\_reserved](#)  
*Reserved.*
- SL [axis\\_h\\_variable](#)  
*H User-defined variable (ZA).*

### 12.42.1 Detailed Description

Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40. Definition at line 1102 of file gclib.cs.

The documentation for this struct was generated from the following file:

- [gclib.cs](#)

## 12.43 GDataRecord52000 Struct Reference

Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40. `#include <gclib_record.h>`

### Data Fields

- UB [header\\_0](#)  
*1st Byte of Header.*
- UB [header\\_1](#)  
*2nd Byte of Header.*
- UB [header\\_2](#)  
*3rd Byte of Header.*
- UB [header\\_3](#)  
*4th Byte of Header.*
- UW [sample\\_number](#)  
*sample number.*
- UB [input\\_bank\\_0](#)  
*general input bank 0 (inputs 1-8).*
- UB [input\\_bank\\_1](#)  
*general input bank 1 (inputs 9-16).*
- UB [input\\_bank\\_2](#)  
*general input bank 2 (inputs 17-24).*
- UB [input\\_bank\\_3](#)  
*general input bank 3 (inputs 25-32).*
- UB [input\\_bank\\_4](#)  
*general input bank 4 (inputs 33-40).*
- UB [input\\_bank\\_5](#)  
*general input bank 5 (inputs 41-48).*

- UB [input\\_bank\\_6](#)  
*general input bank 6 (inputs 49-56).*
- UB [input\\_bank\\_7](#)  
*general input bank 7 (inputs 57-64).*
- UB [input\\_bank\\_8](#)  
*general input bank 8 (inputs 65-72).*
- UB [input\\_bank\\_9](#)  
*general input bank 9 (inputs 73-80).*
- UB [output\\_bank\\_0](#)  
*general output bank 0 (outputs 1-8).*
- UB [output\\_bank\\_1](#)  
*general output bank 1 (outputs 9-16).*
- UB [output\\_bank\\_2](#)  
*general output bank 2 (outputs 17-24).*
- UB [output\\_bank\\_3](#)  
*general output bank 3 (outputs 25-32).*
- UB [output\\_bank\\_4](#)  
*general output bank 4 (outputs 33-40).*
- UB [output\\_bank\\_5](#)  
*general output bank 5 (outputs 41-48).*
- UB [output\\_bank\\_6](#)  
*general output bank 6 (outputs 49-56).*
- UB [output\\_bank\\_7](#)  
*general output bank 7 (outputs 57-64).*
- UB [output\\_bank\\_8](#)  
*general output bank 8 (outputs 65-72).*
- UB [output\\_bank\\_9](#)  
*general output bank 9 (outputs 73-80).*
- SW [reserved\\_0](#)  
*Reserved.*
- SW [reserved\\_2](#)  
*Reserved.*
- SW [reserved\\_4](#)  
*Reserved.*
- SW [reserved\\_6](#)  
*Reserved.*
- SW [reserved\\_8](#)  
*Reserved.*
- SW [reserved\\_10](#)  
*Reserved.*
- SW [reserved\\_12](#)  
*Reserved.*
- UB [ethernet\\_bank](#)  
*EtherCAT Bank Indicator.*
- UB [reserved\\_14](#)  
*Reserved.*
- UB [ethernet\\_status\\_a](#)  
*Ethernet Handle A Status.*
- UB [ethernet\\_status\\_b](#)  
*Ethernet Handle B Status.*
- UB [ethernet\\_status\\_c](#)

- *Ethernet Handle C Status.*
- UB [ethernet\\_status\\_d](#)  
*Ethernet Handle D Status.*
- UB [ethernet\\_status\\_e](#)  
*Ethernet Handle E Status.*
- UB [ethernet\\_status\\_f](#)  
*Ethernet Handle F Status.*
- UB [ethernet\\_status\\_g](#)  
*Ethernet Handle G Status.*
- UB [ethernet\\_status\\_h](#)  
*Ethernet Handle H Status.*
- UB [error\\_code](#)  
*error code.*
- UB [thread\\_status](#)  
*thread status*
- UL [amplifier\\_status](#)  
*Amplifier Status.*
- UL [contour\\_segment\\_count](#)  
*Segment Count for Contour Mode.*
- UW [contour\\_buffer\\_available](#)  
*Buffer space remaining, Contour Mode.*
- UW [s\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for S plane.*
- UW [s\\_plane\\_move\\_status](#)  
*coordinated move status for S plane.*
- SL [s\\_distance](#)  
*distance traveled in coordinated move for S plane.*
- UW [s\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, S Plane.*
- UW [t\\_plane\\_segment\\_count](#)  
*segment count of coordinated move for T plane.*
- UW [t\\_plane\\_move\\_status](#)  
*Coordinated move status for T plane.*
- SL [t\\_distance](#)  
*distance traveled in coordinated move for T plane.*
- UW [t\\_plane\\_buffer\\_available](#)  
*Buffer space remaining, T Plane.*
- UW [axis\\_a\\_status](#)  
*A axis status.*
- UB [axis\\_a\\_switches](#)  
*A axis switches.*
- UB [axis\\_a\\_stop\\_code](#)  
*A axis stop code.*
- SL [axis\\_a\\_reference\\_position](#)  
*A axis reference position.*
- SL [axis\\_a\\_motor\\_position](#)  
*A axis motor position.*
- SL [axis\\_a\\_position\\_error](#)  
*A axis position error.*
- SL [axis\\_a\\_aux\\_position](#)  
*A axis auxiliary position.*

- SL [axis\\_a\\_velocity](#)  
*A axis velocity.*
- SL [axis\\_a\\_torque](#)  
*A axis torque.*
- UW [axis\\_a\\_analog\\_in](#)  
*A axis analog input.*
- UB [axis\\_a\\_halls](#)  
*A Hall Input Status.*
- UB [axis\\_a\\_reserved](#)  
*Reserved.*
- SL [axis\\_a\\_variable](#)  
*A User-defined variable (ZA).*
- UW [axis\\_b\\_status](#)  
*B axis status.*
- UB [axis\\_b\\_switches](#)  
*B axis switches.*
- UB [axis\\_b\\_stop\\_code](#)  
*B axis stop code.*
- SL [axis\\_b\\_reference\\_position](#)  
*B axis reference position.*
- SL [axis\\_b\\_motor\\_position](#)  
*B axis motor position.*
- SL [axis\\_b\\_position\\_error](#)  
*B axis position error.*
- SL [axis\\_b\\_aux\\_position](#)  
*B axis auxiliary position.*
- SL [axis\\_b\\_velocity](#)  
*B axis velocity.*
- SL [axis\\_b\\_torque](#)  
*B axis torque.*
- UW [axis\\_b\\_analog\\_in](#)  
*B axis analog input.*
- UB [axis\\_b\\_halls](#)  
*B Hall Input Status.*
- UB [axis\\_b\\_reserved](#)  
*Reserved.*
- SL [axis\\_b\\_variable](#)  
*B User-defined variable (ZA).*
- UW [axis\\_c\\_status](#)  
*C axis status.*
- UB [axis\\_c\\_switches](#)  
*C axis switches.*
- UB [axis\\_c\\_stop\\_code](#)  
*C axis stop code.*
- SL [axis\\_c\\_reference\\_position](#)  
*C axis reference position.*
- SL [axis\\_c\\_motor\\_position](#)  
*C axis motor position.*
- SL [axis\\_c\\_position\\_error](#)  
*C axis position error.*
- SL [axis\\_c\\_aux\\_position](#)



- C axis auxiliary position.*
- SL [axis\\_c\\_velocity](#)  
*C axis velocity.*
- SL [axis\\_c\\_torque](#)  
*C axis torque.*
- UW [axis\\_c\\_analog\\_in](#)  
*C axis analog input.*
- UB [axis\\_c\\_halls](#)  
*C Hall Input Status.*
- UB [axis\\_c\\_reserved](#)  
*Reserved.*
- SL [axis\\_c\\_variable](#)  
*C User-defined variable (ZA).*
- UW [axis\\_d\\_status](#)  
*D axis status.*
- UB [axis\\_d\\_switches](#)  
*D axis switches.*
- UB [axis\\_d\\_stop\\_code](#)  
*D axis stop code.*
- SL [axis\\_d\\_reference\\_position](#)  
*D axis reference position.*
- SL [axis\\_d\\_motor\\_position](#)  
*D axis motor position.*
- SL [axis\\_d\\_position\\_error](#)  
*D axis position error.*
- SL [axis\\_d\\_aux\\_position](#)  
*D axis auxiliary position.*
- SL [axis\\_d\\_velocity](#)  
*D axis velocity.*
- SL [axis\\_d\\_torque](#)  
*D axis torque.*
- UW [axis\\_d\\_analog\\_in](#)  
*D axis analog input.*
- UB [axis\\_d\\_halls](#)  
*D Hall Input Status.*
- UB [axis\\_d\\_reserved](#)  
*Reserved.*
- SL [axis\\_d\\_variable](#)  
*D User-defined variable (ZA).*
- UW [axis\\_e\\_status](#)  
*E axis status.*
- UB [axis\\_e\\_switches](#)  
*E axis switches.*
- UB [axis\\_e\\_stop\\_code](#)  
*E axis stop code.*
- SL [axis\\_e\\_reference\\_position](#)  
*E axis reference position.*
- SL [axis\\_e\\_motor\\_position](#)  
*E axis motor position.*
- SL [axis\\_e\\_position\\_error](#)  
*E axis position error.*

- SL [axis\\_e\\_aux\\_position](#)  
*E axis auxiliary position.*
- SL [axis\\_e\\_velocity](#)  
*E axis velocity.*
- SL [axis\\_e\\_torque](#)  
*E axis torque.*
- UW [axis\\_e\\_analog\\_in](#)  
*E axis analog input.*
- UB [axis\\_e\\_halls](#)  
*E Hall Input Status.*
- UB [axis\\_e\\_reserved](#)  
*Reserved.*
- SL [axis\\_e\\_variable](#)  
*E User-defined variable (ZA).*
- UW [axis\\_f\\_status](#)  
*F axis status.*
- UB [axis\\_f\\_switches](#)  
*F axis switches.*
- UB [axis\\_f\\_stop\\_code](#)  
*F axis stop code.*
- SL [axis\\_f\\_reference\\_position](#)  
*F axis reference position.*
- SL [axis\\_f\\_motor\\_position](#)  
*F axis motor position.*
- SL [axis\\_f\\_position\\_error](#)  
*F axis position error.*
- SL [axis\\_f\\_aux\\_position](#)  
*F axis auxiliary position.*
- SL [axis\\_f\\_velocity](#)  
*F axis velocity.*
- SL [axis\\_f\\_torque](#)  
*F axis torque.*
- UW [axis\\_f\\_analog\\_in](#)  
*F axis analog input.*
- UB [axis\\_f\\_halls](#)  
*F Hall Input Status.*
- UB [axis\\_f\\_reserved](#)  
*Reserved.*
- SL [axis\\_f\\_variable](#)  
*F User-defined variable (ZA).*
- UW [axis\\_g\\_status](#)  
*G axis status.*
- UB [axis\\_g\\_switches](#)  
*G axis switches.*
- UB [axis\\_g\\_stop\\_code](#)  
*G axis stop code.*
- SL [axis\\_g\\_reference\\_position](#)  
*G axis reference position.*
- SL [axis\\_g\\_motor\\_position](#)  
*G axis motor position.*
- SL [axis\\_g\\_position\\_error](#)

- G axis position error.*
- SL [axis\\_g\\_aux\\_position](#)
  - G axis auxiliary position.*
- SL [axis\\_g\\_velocity](#)
  - G axis velocity.*
- SL [axis\\_g\\_torque](#)
  - G axis torque.*
- UW [axis\\_g\\_analog\\_in](#)
  - G axis analog input.*
- UB [axis\\_g\\_halls](#)
  - G Hall Input Status.*
- UB [axis\\_g\\_reserved](#)
  - Reserved.*
- SL [axis\\_g\\_variable](#)
  - G User-defined variable (ZA).*
- UW [axis\\_h\\_status](#)
  - H axis status.*
- UB [axis\\_h\\_switches](#)
  - H axis switches.*
- UB [axis\\_h\\_stop\\_code](#)
  - H axis stop code.*
- SL [axis\\_h\\_reference\\_position](#)
  - H axis reference position.*
- SL [axis\\_h\\_motor\\_position](#)
  - H axis motor position.*
- SL [axis\\_h\\_position\\_error](#)
  - H axis position error.*
- SL [axis\\_h\\_aux\\_position](#)
  - H axis auxiliary position.*
- SL [axis\\_h\\_velocity](#)
  - H axis velocity.*
- SL [axis\\_h\\_torque](#)
  - H axis torque.*
- UW [axis\\_h\\_analog\\_in](#)
  - H axis analog input.*
- UB [axis\\_h\\_halls](#)
  - H Hall Input Status.*
- UB [axis\\_h\\_reserved](#)
  - Reserved.*
- SL [axis\\_h\\_variable](#)
  - H User-defined variable (ZA).*

### 12.43.1 Detailed Description

Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40.

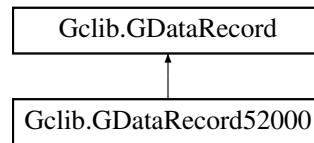
Definition at line 218 of file `gclib_record.h`.

The documentation for this struct was generated from the following file:

- [gclib\\_record.h](#)

## 12.44 Gclib.GDataRecord52000 Struct Reference

Inheritance diagram for Gclib.GDataRecord52000:



### Public Member Functions

- general input **bank** (inputs 1-8). *\*/public UB input\_bank\_1*  
*/\*07*
- general input **bank** (inputs 9-16). *\*/public UB input\_bank\_2*  
*/\*08*
- general input **bank** (inputs 17-24). *\*/public UB input\_bank\_3*  
*/\*09*
- general input **bank** (inputs 25-32). *\*/public UB input\_bank\_4*  
*/\*10*
- general input **bank** (inputs 33-40). *\*/public UB input\_bank\_5*  
*/\*11*
- general input **bank** (inputs 41-48). *\*/public UB input\_bank\_6*  
*/\*12*
- general input **bank** (inputs 49-56). *\*/public UB input\_bank\_7*  
*/\*13*
- general input **bank** (inputs 57-64). *\*/public UB input\_bank\_8*  
*/\*14*
- general input **bank** (inputs 65-72). *\*/public UB input\_bank\_9*  
*/\*15*
- general input **bank** (inputs 73-80). *\*/public UB output\_bank\_0*  
*/\*16*
- general output **bank** (outputs 1-8). *\*/public UB output\_bank\_1*  
*/\*17*
- general output **bank** (outputs 9-16). *\*/public UB output\_bank\_2*  
*/\*18*
- general output **bank** (outputs 17-24). *\*/public UB output\_bank\_3*  
*/\*19*
- general output **bank** (outputs 25-32). *\*/public UB output\_bank\_4*  
*/\*20*
- general output **bank** (outputs 33-40). *\*/public UB output\_bank\_5*  
*/\*21*
- general output **bank** (outputs 41-48). *\*/public UB output\_bank\_6*  
*/\*22*
- general output **bank** (outputs 49-56). *\*/public UB output\_bank\_7*  
*/\*23*
- general output **bank** (outputs 57-64). *\*/public UB output\_bank\_8*  
*/\*24*
- general output **bank** (outputs 65-72). *\*/public UB output\_bank\_9*  
*/\*25*
- general output **bank** (outputs 73-80). *\*/public SW reserved\_0*  
*/\*26-27*
- H User defined variable.\*[] **byte\_array** ()

## Data Fields

- UB [header\\_0](#)  
*/\*00*
- byte of Header.\*UB [header\\_1](#)  
*/\*01*
- byte of Header.\*UB [header\\_2](#)  
*/\*02*
- byte of Header.\*UB [header\\_3](#)  
*/\*03*
- byte of Header.\*UW [sample\\_number](#)  
*/\*04-05*
- sample number.\*UB [input\\_bank\\_0](#)  
*/\*06*
- Reserved.\*SW [reserved\\_2](#)  
*/\*28-29*
- Reserved.\*SW [reserved\\_4](#)  
*/\*30-31*
- Reserved.\*SW [reserved\\_6](#)  
*/\*32-33*
- Reserved.\*SW [reserved\\_8](#)  
*/\*34-35*
- Reserved.\*SW [reserved\\_10](#)  
*/\*36-37*
- Reserved.\*SW [reserved\\_12](#)  
*/\*38-39*
- Reserved.\*UB [ethercat\\_bank](#)  
*/\*40*
- EtherCAT Bank Indicator.\*UB [reserved\\_14](#)  
*/\*41*
- Reserved.\*UB [ethernet\\_status\\_a](#)  
*/\*42*
- Ethernet Handle A Status.\*UB [ethernet\\_status\\_b](#)  
*/\*43*
- Ethernet Handle B Status.\*UB [ethernet\\_status\\_c](#)  
*/\*44*
- Ethernet Handle C Status.\*UB [ethernet\\_status\\_d](#)  
*/\*45*
- Ethernet Handle D Status.\*UB [ethernet\\_status\\_e](#)  
*/\*46*
- Ethernet Handle E Status.\*UB [ethernet\\_status\\_f](#)  
*/\*47*
- Ethernet Handle F Status.\*UB [ethernet\\_status\\_g](#)  
*/\*48*
- Ethernet Handle G Status.\*UB [ethernet\\_status\\_h](#)  
*/\*49*
- Ethernet Handle H Status.\*UB [error\\_code](#)  
*/\*50*
- [error](#) code.\*UB [thread\\_status](#)  
*/\*51*
- thread status \*UL [amplifier\\_status](#)

- /\*52-55
- Amplifier Status.\*UL [contour\\_segment\\_count](#)
- /\*56-59
- Segment Count for Contour Mode.\*UW [contour\\_buffer\\_available](#)
- /\*60-61
- Buffer space **remaining**
- Buffer space Contour Mode.\*UW [s\\_plane\\_segment\\_count](#)
- /\*62-63
- segment count of coordinated move for S plane.\*UW [s\\_plane\\_move\\_status](#)
- /\*64-65
- coordinated move status for S plane.\*SL [s\\_distance](#)
- /\*66-69
- distance traveled in coordinated move for S plane.\*UW [s\\_plane\\_buffer\\_available](#)
- /\*70-71
- Buffer space S Plane.\*UW [t\\_plane\\_segment\\_count](#)
- /\*72-73
- segment count of coordinated move for T plane.\*UW [t\\_plane\\_move\\_status](#)
- /\*74-75
- Coordinated move status for T plane.\*SL [t\\_distance](#)
- /\*76-79
- distance traveled in coordinated move for T plane.\*UW [t\\_plane\\_buffer\\_available](#)
- /\*80-81
- Buffer space T Plane.\*UW [axis\\_a\\_status](#)
- /\*82-83
- A axis status.\*UB [axis\\_a\\_switches](#)
- /\*84
- A axis switches.\*UB [axis\\_a\\_stop\\_code](#)
- /\*85
- A axis stop code.\*SL [axis\\_a\\_reference\\_position](#)
- /\*86-89
- A axis reference position.\*SL [axis\\_a\\_motor\\_position](#)
- /\*90-93
- A axis motor position.\*SL [axis\\_a\\_position\\_error](#)
- /\*94-97
- A axis position error.\*SL [axis\\_a\\_aux\\_position](#)
- /\*98-101
- A axis auxiliary position.\*SL [axis\\_a\\_velocity](#)
- /\*102-105
- A axis velocity.\*SL [axis\\_a\\_torque](#)
- /\*106-109
- A axis torque.\*UW [axis\\_a\\_analog\\_in](#)
- /\*110-111
- A axis analog input.\*UB [axis\\_a\\_halls](#)
- /\*112
- A Hall Input Status.\*UB [axis\\_a\\_reserved](#)
- /\*113
- Reserved.\*SL [axis\\_a\\_variable](#)
- /\*114-117
- A User defined variable.\*[] UW [axis\\_b\\_status](#)
- /\*118-119
- B axis status.\*UB [axis\\_b\\_switches](#)

- /\* 120*
- B axis switches.\*UB [axis\\_b\\_stop\\_code](#)
- /\* 121*
- B axis stop code.\*SL [axis\\_b\\_reference\\_position](#)
- /\* 122-125*
- B axis reference position.\*SL [axis\\_b\\_motor\\_position](#)
- /\* 126-129*
- B axis motor position.\*SL [axis\\_b\\_position\\_error](#)
- /\* 130-133*
- B axis position error.\*SL [axis\\_b\\_aux\\_position](#)
- /\* 134-137*
- B axis auxiliary position.\*SL [axis\\_b\\_velocity](#)
- /\* 138-141*
- B axis velocity.\*SL [axis\\_b\\_torque](#)
- /\* 142-145*
- B axis torque.\*UW [axis\\_b\\_analog\\_in](#)
- /\* 146-147*
- B axis analog input.\*UB [axis\\_b\\_halls](#)
- /\* 148*
- B Hall Input Status.\*UB [axis\\_b\\_reserved](#)
- /\* 149*
- Reserved.\*SL [axis\\_b\\_variable](#)
- /\* 150-153*
- B User defined variable.\*[] UW [axis\\_c\\_status](#)
- /\* 154-155*
- C axis status.\*UB [axis\\_c\\_switches](#)
- /\* 156*
- C axis switches.\*UB [axis\\_c\\_stop\\_code](#)
- /\* 157*
- C axis stop code.\*SL [axis\\_c\\_reference\\_position](#)
- /\* 158-161*
- C axis reference position.\*SL [axis\\_c\\_motor\\_position](#)
- /\* 162-165*
- C axis motor position.\*SL [axis\\_c\\_position\\_error](#)
- /\* 166-169*
- C axis position error.\*SL [axis\\_c\\_aux\\_position](#)
- /\* 170-173*
- C axis auxiliary position.\*SL [axis\\_c\\_velocity](#)
- /\* 174-177*
- C axis velocity.\*SL [axis\\_c\\_torque](#)
- /\* 178-181*
- C axis torque.\*UW [axis\\_c\\_analog\\_in](#)
- /\* 182-183*
- C axis analog input.\*UB [axis\\_c\\_halls](#)
- /\* 184*
- C Hall Input Status.\*UB [axis\\_c\\_reserved](#)
- /\* 185*
- Reserved.\*SL [axis\\_c\\_variable](#)
- /\* 186-189*
- C User defined variable.\*[] UW [axis\\_d\\_status](#)
- /\* 190-191*

- D axis status.\*UB [axis\\_d\\_switches](#)  
/\* 192
- D axis switches.\*UB [axis\\_d\\_stop\\_code](#)  
/\* 193
- D axis stop code.\*SL [axis\\_d\\_reference\\_position](#)  
/\* 194-197
- D axis reference position.\*SL [axis\\_d\\_motor\\_position](#)  
/\* 198-201
- D axis motor position.\*SL [axis\\_d\\_position\\_error](#)  
/\* 202-205
- D axis position error.\*SL [axis\\_d\\_aux\\_position](#)  
/\* 206-209
- D axis auxiliary position.\*SL [axis\\_d\\_velocity](#)  
/\* 210-213
- D axis velocity.\*SL [axis\\_d\\_torque](#)  
/\* 214-217
- D axis torque.\*UW [axis\\_d\\_analog\\_in](#)  
/\* 218-219
- D axis analog input.\*UB [axis\\_d\\_halls](#)  
/\* 220
- D Hall Input Status.\*UB [axis\\_d\\_reserved](#)  
/\* 221
- Reserved.\*SL [axis\\_d\\_variable](#)  
/\* 222-225
- D User defined variable.\*[] UW [axis\\_e\\_status](#)  
/\* 226-227
- E axis status.\*UB [axis\\_e\\_switches](#)  
/\* 228
- E axis switches.\*UB [axis\\_e\\_stop\\_code](#)  
/\* 229
- E axis stop code.\*SL [axis\\_e\\_reference\\_position](#)  
/\* 230-233
- E axis reference position.\*SL [axis\\_e\\_motor\\_position](#)  
/\* 234-237
- E axis motor position.\*SL [axis\\_e\\_position\\_error](#)  
/\* 238-241
- E axis position error.\*SL [axis\\_e\\_aux\\_position](#)  
/\* 242-245
- E axis auxiliary position.\*SL [axis\\_e\\_velocity](#)  
/\* 246-249
- E axis velocity.\*SL [axis\\_e\\_torque](#)  
/\* 250-253
- E axis torque.\*UW [axis\\_e\\_analog\\_in](#)  
/\* 254-255
- E axis analog input.\*UB [axis\\_e\\_halls](#)  
/\* 256
- E Hall Input Status.\*UB [axis\\_e\\_reserved](#)  
/\* 257
- Reserved.\*SL [axis\\_e\\_variable](#)  
/\* 258-261
- E User defined variable.\*[] UW [axis\\_f\\_status](#)



- /\*262-263*
  - F axis status.\*UB [axis\\_f\\_switches](#)
- /\*264*
  - F axis switches.\*UB [axis\\_f\\_stop\\_code](#)
- /\*265*
  - F axis stop code.\*SL [axis\\_f\\_reference\\_position](#)
- /\*266-269*
  - F axis reference position.\*SL [axis\\_f\\_motor\\_position](#)
- /\*270-273*
  - F axis motor position.\*SL [axis\\_f\\_position\\_error](#)
- /\*274-277*
  - F axis position error.\*SL [axis\\_f\\_aux\\_position](#)
- /\*278-281*
  - F axis auxiliary position.\*SL [axis\\_f\\_velocity](#)
- /\*282-285*
  - F axis velocity.\*SL [axis\\_f\\_torque](#)
- /\*286-289*
  - F axis torque.\*UW [axis\\_f\\_analog\\_in](#)
- /\*290-291*
  - F axis analog input.\*UB [axis\\_f\\_halls](#)
- /\*292*
  - F Hall Input Status.\*UB [axis\\_f\\_reserved](#)
- /\*293*
  - Reserved.\*SL [axis\\_f\\_variable](#)
- /\*294-297*
  - Reserved.\*SL [axis\\_f\\_variable](#)
- /\*298-299*
  - F User defined variable.\*[] UW [axis\\_g\\_status](#)
- /\*298-299*
  - F User defined variable.\*[] UW [axis\\_g\\_status](#)
- /\*300*
  - G axis status.\*UB [axis\\_g\\_switches](#)
- /\*301*
  - G axis switches.\*UB [axis\\_g\\_stop\\_code](#)
- /\*301*
  - G axis stop code.\*UB [axis\\_g\\_stop\\_code](#)
- /\*302-305*
  - G axis stop code.\*SL [axis\\_g\\_reference\\_position](#)
- /\*302-305*
  - G axis stop code.\*SL [axis\\_g\\_reference\\_position](#)
- /\*306-309*
  - G axis reference position.\*SL [axis\\_g\\_motor\\_position](#)
- /\*306-309*
  - G axis reference position.\*SL [axis\\_g\\_motor\\_position](#)
- /\*310-313*
  - G axis motor position.\*SL [axis\\_g\\_position\\_error](#)
- /\*310-313*
  - G axis motor position.\*SL [axis\\_g\\_position\\_error](#)
- /\*314-317*
  - G axis position error.\*SL [axis\\_g\\_aux\\_position](#)
- /\*314-317*
  - G axis position error.\*SL [axis\\_g\\_aux\\_position](#)
- /\*318-321*
  - G axis auxiliary position.\*SL [axis\\_g\\_velocity](#)
- /\*318-321*
  - G axis auxiliary position.\*SL [axis\\_g\\_velocity](#)
- /\*322-325*
  - G axis velocity.\*SL [axis\\_g\\_torque](#)
- /\*322-325*
  - G axis velocity.\*SL [axis\\_g\\_torque](#)
- /\*326-327*
  - G axis torque.\*UW [axis\\_g\\_analog\\_in](#)
- /\*326-327*
  - G axis torque.\*UW [axis\\_g\\_analog\\_in](#)
- /\*328*
  - G axis analog input.\*UB [axis\\_g\\_halls](#)
- /\*328*
  - G axis analog input.\*UB [axis\\_g\\_halls](#)
- /\*329*
  - G Hall Input Status.\*UB [axis\\_g\\_reserved](#)
- /\*329*
  - G Hall Input Status.\*UB [axis\\_g\\_reserved](#)
- /\*330-333*
  - Reserved.\*SL [axis\\_g\\_variable](#)
- /\*330-333*
  - Reserved.\*SL [axis\\_g\\_variable](#)

- G User defined variable.\*[] UW [axis\\_h\\_status](#)  
/\*334-335
- H axis status.\*UB [axis\\_h\\_switches](#)  
/\*336
- H axis switches.\*UB [axis\\_h\\_stop\\_code](#)  
/\*337
- H axis stop code.\*SL [axis\\_h\\_reference\\_position](#)  
/\*338-341
- H axis reference position.\*SL [axis\\_h\\_motor\\_position](#)  
/\*342-345
- H axis motor position.\*SL [axis\\_h\\_position\\_error](#)  
/\*346-349
- H axis position error.\*SL [axis\\_h\\_aux\\_position](#)  
/\*350-353
- H axis auxiliary position.\*SL [axis\\_h\\_velocity](#)  
/\*354-357
- H axis velocity.\*SL [axis\\_h\\_torque](#)  
/\*358-361
- H axis torque.\*UW [axis\\_h\\_analog\\_in](#)  
/\*362-363
- H axis analog input.\*UB [axis\\_h\\_halls](#)  
/\*364
- H Hall Input Status.\*UB [axis\\_h\\_reserved](#)  
/\*365
- Reserved.\*SL [axis\\_h\\_variable](#)  
/\*366-369

### 12.44.1 Detailed Description

Definition at line 635 of file gclib.vb.

### 12.44.2 Member Function Documentation

#### 12.44.2.1 `byte_array()`

H User defined variable.\* [] `byte_array ( )`

Implements [GDataRecord.byte\\_array](#)

Implements [Gclib.GDataRecord](#).

The documentation for this struct was generated from the following file:

- [gclib.vb](#)

## 12.45 H\_ArrayData Struct Reference

Structure to create a linked list for array data.

## Data Fields

- char **name** [16]
- char \* **data**
- int **len**
- int **elements**
- int **index**
- struct [H\\_ArrayData](#) \* **next**
- struct [H\\_ArrayData](#) \* **tail**
- int **count**

### 12.45.1 Detailed Description

Structure to create a linked list for array data.

Definition at line 20 of file arrays.c.

The documentation for this struct was generated from the following file:

- [arrays.c](#)

## 12.46 IP\_Assigner\_Example Class Reference

Assigns controller an IP Address given a serial number and a 1 byte address.

### Static Public Member Functions

- static int [Main](#) (string[] args)  
*Main function for the IP Assigner example.*

### 12.46.1 Detailed Description

Assigns controller an IP Address given a serial number and a 1 byte address.

The first argument should be the Serial # of a [Galil](#) Controller.

The second argument should be a 1 Byte value that will be used as the final byte in the newly assigned IP Address.

For VB.NET, see definition in file [ip\\_assigner\\_example.vb](#)

Definition at line 25 of file ip\_assigner\_example.cs.

### 12.46.2 Member Function Documentation

#### 12.46.2.1 Main()

```
static int Main (
    string[] args ) [inline], [static]
```

Main function for the IP Assigner example.

#### Parameters

<i>args</i>	An array of command line arguments.
-------------	-------------------------------------

#### Returns

The success status or error code of the function.

The first argument should be the Serial # of a [Galil](#) Controller.

The second argument should be a 1 Byte value that will be used as the final byte in the newly assigned IP Address.

Definition at line 35 of file ip\_assigner\_example.cs.

References `Examples.GALIL_EXAMPLE_ERROR`, `Examples.GALIL_EXAMPLE_OK`, `gclib.GClose()`, `Examples.↵  
IP_Assigner()`, and `Examples.PrintError()`.

The documentation for this class was generated from the following file:

- [ip\\_assigner\\_example.cs](#)

## 12.47 Jog\_Example Class Reference

Accepts user-input at the command line to control the speed of the controller in Jog mode.

### Static Public Member Functions

- static int `Main` (string[] args)  
*Main function for the jog example.*

#### 12.47.1 Detailed Description

Accepts user-input at the command line to control the speed of the controller in Jog mode.

The first argument should be the IP Address of a [Galil](#) controller.

For VB.NET, see definition in file [jog\\_example.vb](#)

Definition at line 23 of file `jog_example.cs`.

#### 12.47.2 Member Function Documentation

##### 12.47.2.1 Main()

```
static int Main (
    string[] args ) [inline], [static]
```

Main function for the jog example.

##### Parameters

<code>args</code>	An array of command line arguments.
-------------------	-------------------------------------

##### Returns

The success status or error code of the function.

The first argument should be the IP Address of a [Galil](#) controller.

Definition at line 31 of file `jog_example.cs`.

References `Examples.GALIL_EXAMPLE_ERROR`, `Examples.GALIL_EXAMPLE_OK`, `gclib.GClose()`, `gclib.↵  
GOpen()`, `Examples.Jog()`, and `Examples.PrintError()`.

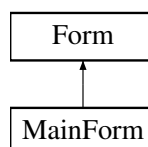
The documentation for this class was generated from the following file:

- [jog\\_example.cs](#)

## 12.48 MainForm Class Reference

Demonstrates using `gclib` in a Windows Form, including using a second thread to free the GUI.

Inheritance diagram for `MainForm`:



### 12.48.1 Detailed Description

Demonstrates using gclib in a Windows Form, including using a second thread to free the GUI.  
 Demonstrates using gclib in a Windows Form, including using a second thread to free the GUI.

#### Remarks

Handles the MyBase.Load event.

Definition at line 22 of file Form1.cs.

The documentation for this class was generated from the following files:

- [Form1.cs](#)
- [Form1.vb](#)

## 12.49 Message\_Example Class Reference

Demonstrates how to handle and interpret messages from the controller.

### Static Public Member Functions

- static int [Main](#) (string[] args)  
*Main function for the message example.*

### 12.49.1 Detailed Description

Demonstrates how to handle and interpret messages from the controller.

The first argument should be the IP Address of a [Galil](#) controller.

For VB.NET, see definition in file [message\\_example.vb](#)

Definition at line 23 of file message\_example.cs.

### 12.49.2 Member Function Documentation

#### 12.49.2.1 Main()

```
static int Main (
    string[] args ) [inline], [static]
```

Main function for the message example.

#### Parameters

<i>args</i>	An array of command line arguments.
-------------	-------------------------------------

#### Returns

The success status or error code of the function.

The first argument should be the IP Address of a [Galil](#) controller.

Definition at line 31 of file message\_example.cs.

References [Examples.GALIL\\_EXAMPLE\\_ERROR](#), [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GClose\(\)](#), [gclib.GOpen\(\)](#), [Examples.Message\(\)](#), and [Examples.PrintError\(\)](#).

The documentation for this class was generated from the following file:

- [message\\_example.cs](#)

## 12.50 Motion\_Complete\_Example Class Reference

Uses controller interrupts to detect when motion is complete.

## Static Public Member Functions

- static int [Main](#) (string[] args)

*Main function for the Motion Complete example.*

### 12.50.1 Detailed Description

Uses controller interrupts to detect when motion is complete.

The first argument should be the IP Address of a [Galil](#) controller.

For VB.NET, see definition in file [motion\\_complete\\_example.vb](#)

Definition at line 23 of file motion\_complete\_example.cs.

### 12.50.2 Member Function Documentation

#### 12.50.2.1 Main()

```
static int Main (
    string[] args ) [inline], [static]
```

Main function for the Motion Complete example.

#### Parameters

<i>args</i>	An array of command line arguments.
-------------	-------------------------------------

#### Returns

The success status or error code of the function.

The first argument should be the IP Address of a [Galil](#) controller.

Definition at line 31 of file motion\_complete\_example.cs.

References [Examples.GALIL\\_EXAMPLE\\_ERROR](#), [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GClose\(\)](#), [gclib.GOpen\(\)](#), [Examples.Motion\\_Complete\(\)](#), and [Examples.PrintError\(\)](#).

The documentation for this class was generated from the following file:

- [motion\\_complete\\_example.cs](#)

## 12.51 Position\_Tracking\_Example Class Reference

Places controller into position tracking mode. Accepts user-defined positional values at the command line.

### Static Public Member Functions

- static int [Main](#) (string[] args)

*Main function for the position tracking example.*

#### 12.51.1 Detailed Description

Places controller into position tracking mode. Accepts user-defined positional values at the command line.

The first argument should be the IP Address of a [Galil](#) controller.

The second argument is optional and defines the default speed of the controller in Position Tracking mode.

For VB.NET, see definition in file [position\\_tracking\\_example.vb](#)

Definition at line 26 of file position\_tracking\_example.cs.

#### 12.51.2 Member Function Documentation

### 12.51.2.1 Main()

```
static int Main (
    string[] args ) [inline], [static]
```

Main function for the position tracking example.

#### Parameters

<code>args</code>	An array of command line arguments.
-------------------	-------------------------------------

#### Returns

The success status or error code of the function.

The first argument should be the IP Address of a [Galil](#) controller. The second argument is optional and defines the default speed of the controller in Position Tracking mode.

Definition at line 36 of file `position_tracking_example.cs`.

References `Examples.GALIL_EXAMPLE_ERROR`, `Examples.GALIL_EXAMPLE_OK`, `gclib.GClose()`, `gclib.GOpen()`, `Examples.Position_Tracking()`, and `Examples.PrintError()`.

The documentation for this class was generated from the following file:

- [position\\_tracking\\_example.cs](#)

## 12.52 Program Class Reference

### 12.52.1 Detailed Description

Definition at line 11 of file `Program.cs`.

The documentation for this class was generated from the following file:

- [Program.cs](#)

## 12.53 gclib.py Class Reference

### Public Member Functions

- def `__init__` (self)
- def `__del__` (self)
- def `GOpen` (self, address)
- def `GClose` (self)
- def `GCommand` (self, command)
- def `GSleep` (self, val)
- def `GVersion` (self)
- def `GServerStatus` (self)
- def `GSetServer` (self, server\_name)
- def `GListServers` (self)
- def `GPublishServer` (self, server\_name, publish, save)
- def `GRemoteConnections` (self)
- def `GInfo` (self)
- def `GlpRequests` (self)
- def `GAssign` (self, ip, mac)
- def `GAddresses` (self)
- def `GProgramDownload` (self, program, preprocessor='')
- def `GProgramUpload` (self)
- def `GProgramDownloadFile` (self, file\_path, preprocessor='')
- def `GProgramUploadFile` (self, file\_path)
- def `GArrayDownload` (self, name, first, last, array\_data)

- def `GArrayUploadFile` (self, file\_path, names=[])
- def `GArrayDownloadFile` (self, file\_path)
- def `GArrayUpload` (self, name, first, last)
- def `GTimeout` (self, timeout)
- def `timeout` (self)
- def `timeout` (self, timeout)
- def `GFirmwareDownload` (self, file\_path)
- def `GMessage` (self)
- def `GMotionComplete` (self, axes)
- def `GInterrupt` (self)
- def `GSetupDownloadFile` (self, file\_path, options)

### 12.53.1 Detailed Description

Represents a single Python connection to a Galil Controller or PLC.

Definition at line 141 of file `gclib.py`.

### 12.53.2 Constructor & Destructor Documentation

#### 12.53.2.1 `__init__()`

```
def __init__ (
    self )
```

Constructor for the Connection class. Initializes `gclib`'s handle and read buffer.

Definition at line 144 of file `gclib.py`.

References `gclib.py._buf`, `gclib.py._gcon`, and `gclib.py._timeout`.

#### 12.53.2.2 `__del__()`

```
def __del__ (
    self )
```

Destructor for the Connection class. Ensures `close` gets called to release Galil resource (Sockets, Kernel Drive).

Definition at line 151 of file `gclib.py`.

References `gclib.py._gcon`, `gclib.GClose()`, `GclibJava.GClose()`, `Gclib.GClose()`, `GclibJava.Gclib.GClose()`, and `gclib.py.GClose()`.

### 12.53.3 Member Function Documentation

#### 12.53.3.1 `GAddresses()`

```
def GAddresses (
    self )
```

Provides a dictionary of all available connection addresses.

Returns a dictionary mapping 'address' -> 'revision reports', where possible  
e.g. {}

Definition at line 269 of file `gclib.py`.

References `gclib.py._buf`.



### 12.53.3.2 GArrayDownload()

```
def GArrayDownload (
    self,
    name,
    first,
    last,
    array_data )
```

Downloads array data to a pre-dimensioned array in the controller's array table. array\_data should be a list of values (e.g. int or float)

Definition at line 329 of file gclib.py.  
References gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.3 GArrayDownloadFile()

```
def GArrayDownloadFile (
    self,
    file_path )
```

Downloads a csv file containing array data at file\_path.

Definition at line 361 of file gclib.py.  
References gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.4 GArrayUpload()

```
def GArrayUpload (
    self,
    name,
    first,
    last )
```

Uploads array data from the controller's array table.

Definition at line 371 of file gclib.py.  
References gclib.py.\_buf, gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.5 GArrayUploadFile()

```
def GArrayUploadFile (
    self,
    file_path,
    names = [] )
```

Uploads the entire controller array table or a subset and saves the data as a csv file specified by file\_path. names is optional and should be a list of array names on the controller.

Definition at line 344 of file gclib.py.  
References gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.6 GAssign()

```
def GAssign (
    self,
    ip,
    mac )
```

Assigns IP address over the Ethernet to a controller at a given MAC address.  
Linux/OS X users must be root to use GAssign() and have UDP access to send on port 68.

Definition at line 258 of file gclib.py.

### 12.53.3.7 GClose()

```
def GClose (
    self )
```

Closes a connection to a Galil Controller.

Definition at line 171 of file gclib.py.

References gclib.py.\_gcon.

Referenced by gclib.py.\_\_del\_\_().

### 12.53.3.8 GCommand()

```
def GCommand (
    self,
    command )
```

Performs a command-and-response transaction on the connection.  
Trims the response.

Definition at line 181 of file gclib.py.

References gclib.py.\_buf, gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.9 GFirmwareDownload()

```
def GFirmwareDownload (
    self,
    file_path )
```

Upgrade firmware.

Definition at line 411 of file gclib.py.

References gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.10 GInfo()

```
def GInfo (
    self )
```

Provides a useful connection string. Please include the output of this function on all support cases.

Definition at line 230 of file gclib.py.

References gclib.py.\_buf, and gclib.py.\_gcon.

### 12.53.3.11 GInterrupt()

```
def GInterrupt (
    self )
```

Provides access to PCI and UDP interrupts from the controller.

Definition at line 439 of file gclib.py.

References gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.12 GIpRequests()

```
def GIpRequests (
    self )
```

Provides a dictionary of all Galil controllers requesting IP addresses via BOOT-P or DHCP.

Returns a dictionary mapping 'model-serial' --> 'mac address'  
e.g. {'DMC4000-783': '00:50:4c:20:03:0f', 'DMC4103-9998': '00:50:4c:38:27:0e'}

Linux/OS X users must be root to use GIpRequests() and have UDP access to bind and listen on port 67.

Definition at line 238 of file gclib.py.

References gclib.py.\_buf.

### 12.53.3.13 GMessage()

```
def GMessage (
    self )
```

Provides access to unsolicited messages from the controller.

Definition at line 421 of file gclib.py.

References gclib.py.\_buf, gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.14 GMotionComplete()

```
def GMotionComplete (
    self,
    axes )
```

Blocking call that returns once all axes specified have completed their motion.

Definition at line 430 of file gclib.py.

References gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.15 GOpen()

```
def GOpen (
    self,
    address )
```

Opens a connection a galil controller.  
See the gclib docs for address string formatting.

Definition at line 161 of file gclib.py.

References gclib.py.\_gcon.

### 12.53.3.16 GProgramDownload()

```
def GProgramDownload (
    self,
    program,
    preprocessor = "" )
```

Downloads a program to the controller's program buffer.  
See the gclib docs for preprocessor options.

Definition at line 288 of file gclib.py.

References gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.17 GProgramDownloadFile()

```
def GProgramDownloadFile (
    self,
    file_path,
    preprocessor = "" )
```

Program download from file.  
See the gclib docs for preprocessor options.

Definition at line 309 of file gclib.py.  
References gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.18 GProgramUpload()

```
def GProgramUpload (
    self )
```

Uploads a program from the controller's program buffer.

Definition at line 300 of file gclib.py.  
References gclib.py.\_buf, gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.19 GProgramUploadFile()

```
def GProgramUploadFile (
    self,
    file_path )
```

Program upload to file.

Definition at line 320 of file gclib.py.  
References gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.20 GSetupDownloadFile()

```
def GSetupDownloadFile (
    self,
    file_path,
    options )
```

Downloads specified sectors from a Galil compressed backup (gcb) file to a controller.

Returns a dictionary with the controller information stored in the gcb file.  
If options is specified as 0, an additional "options" key will be in the dictionary indicating the info sector

Definition at line 448 of file gclib.py.  
References gclib.py.\_buf, gclib.py.\_cc(), and gclib.py.\_gcon.

### 12.53.3.21 GSleep()

```
def GSleep (
    self,
    val )
```

Provides a blocking sleep call which can be useful for timing-based chores.

Definition at line 193 of file gclib.py.

**12.53.3.22 GTimeout()**

```
def GTimeout (
    self,
    timeout )
```

Set the library timeout. Set to -1 to use the initial library timeout, as specified in GOpen.

Definition at line 385 of file gclib.py.

References gclib.py.\_cc(), gclib.py.\_gcon, and gclib.py.\_timeout.

Referenced by gclib.py.timeout().

**12.53.3.23 GVersion()**

```
def GVersion (
    self )
```

Provides the gclib version number. Please include the output of this function on all support cases.

Definition at line 201 of file gclib.py.

References gclib.py.\_buf.

**12.53.3.24 timeout() [1/2]**

```
def timeout (
    self )
```

Convenience property read access to timeout value. If -1, gclib uses the initial library timeout, as specified

Definition at line 396 of file gclib.py.

References gclib.py.\_timeout.

Referenced by gclib.py.timeout().

**12.53.3.25 timeout() [2/2]**

```
def timeout (
    self,
    timeout )
```

Convenience property write access to timeout value. Set to -1 to use the initial library timeout, as specified

Definition at line 403 of file gclib.py.

References gclib.GTimeout(), Gclib.GTimeout(), GclibJava.Gclibo.GTimeout(), gclib.py.GTimeout(), GclibJava.G

Timeout(), and gclib.py.timeout().

The documentation for this class was generated from the following file:

- [gclib.py](#)

**12.54 Record\_Position\_Example Class Reference**

Takes two file paths at the command line to hold positional data for Axis A and Axis B. Positional data is saved to the two files until an analog input value changes.

**Static Public Member Functions**

- static int [Main](#) (string[] args)

*Main function for the Record Position example.*

### 12.54.1 Detailed Description

Takes two file paths at the command line to hold positional data for Axis A and Axis B. Positional data is saved to the two files until an analog input value changes.

The first argument should be the IP Address of a [Galil](#) controller.

The second argument should be a path to a file to save Axis A positional data.

The third argument should be a path to a file to save Axis B positional data.

For VB.NET, see definition in file [record\\_position\\_example.vb](#)

Definition at line 27 of file [record\\_position\\_example.cs](#).

### 12.54.2 Member Function Documentation

#### 12.54.2.1 Main()

```
static int Main (
    string[] args ) [inline], [static]
```

Main function for the Record Position example.

#### Parameters

<i>args</i>	An array of command line arguments.
-------------	-------------------------------------

#### Returns

The success status or error code of the function.

The first argument should be the IP Address of a [Galil](#) controller.

The second argument should be a path to a file to save Axis A positional data.

The third argument should be a path to a file to save Axis B positional data.

Definition at line 38 of file [record\\_position\\_example.cs](#).

References [Examples.GALIL\\_EXAMPLE\\_ERROR](#), [Examples.GALIL\\_EXAMPLE\\_OK](#), [gclib.GClose\(\)](#), [gclib.GOpen\(\)](#), [Examples.PrintError\(\)](#), and [Examples.Record\\_Position\(\)](#).

The documentation for this class was generated from the following file:

- [record\\_position\\_example.cs](#)

## 12.55 Remote\_Client\_Example Class Reference

Demonstrates various uses of [GListServers\(\)](#) and [GSetServer\(\)](#)

### Static Public Member Functions

- static int [Main](#) ()

*Main function for the Remote Client example.*

#### 12.55.1 Detailed Description

Demonstrates various uses of [GListServers\(\)](#) and [GSetServer\(\)](#)

This example requires no command line arguments.

For VB.NET, see definition in file [remote\\_client\\_example.vb](#)

Definition at line 24 of file [remote\\_client\\_example.cs](#).

#### 12.55.2 Member Function Documentation

### 12.55.2.1 Main()

```
static int Main ( ) [inline], [static]
```

Main function for the Remote Client example.

#### Returns

The success status or error code of the function.

The first argument is an optional name to publish your client under.

Definition at line 31 of file `remote_client_example.cs`.

References `Examples.GALIL_EXAMPLE_ERROR`, `Examples.GALIL_EXAMPLE_OK`, and `Examples.Remote_Client()`.

The documentation for this class was generated from the following file:

- [remote\\_client\\_example.cs](#)

## 12.56 Remote\_Server\_Example Class Reference

Demonstrates various uses of [GPublishServer\(\)](#)

### Static Public Member Functions

- static int [Main](#) (string[] args)  
*Main function for the Remote Server example.*

### 12.56.1 Detailed Description

Demonstrates various uses of [GPublishServer\(\)](#)

The first argument is an optional name to publish your server under.

For VB.NET, see definition in file [remote\\_server\\_example.vb](#)

Definition at line 24 of file `remote_server_example.cs`.

### 12.56.2 Member Function Documentation

#### 12.56.2.1 Main()

```
static int Main (
    string[] args ) [inline], [static]
```

Main function for the Remote Server example.

#### Parameters

<i>args</i>	An array of command line arguments.
-------------	-------------------------------------

#### Returns

The success status or error code of the function.

The first argument is optional and defines the name to publish your server under.

Definition at line 32 of file `remote_server_example.cs`.

References `Examples.GALIL_EXAMPLE_ERROR`, `Examples.GALIL_EXAMPLE_OK`, and `Examples.Remote_Server()`.

The documentation for this class was generated from the following file:

- [remote\\_server\\_example.cs](#)

## 12.57 Source Struct Reference

### Public Member Functions

- **Source** (int byte=0, std::string type="Ux", int bit=-1, std::string units="", std::string description="", double scale=1, double offset=0)

### Data Fields

- int **byte**
- std::string **type**
- int **bit**
- std::string **units**
- std::string **description**
- double **scale**
- double **offset**

### 12.57.1 Detailed Description

Definition at line 16 of file `gcl_galil.h`.

The documentation for this struct was generated from the following file:

- [gcl\\_galil.h](#)

## 12.58 Vector\_Mode\_Example Class Reference

Takes a path to a file at the command line holding vector commands for the controller. The controller is placed into vector mode and commands are read from the file and sent to the controller.

### Static Public Member Functions

- static int [Main](#) (string[] args)  
*Main function for the vector mode example.*

### 12.58.1 Detailed Description

Takes a path to a file at the command line holding vector commands for the controller. The controller is placed into vector mode and commands are read from the file and sent to the controller.

The first argument should be the IP Address of a [Galil](#) controller. The second argument should be a path to a file containing vector commands.

For VB.NET, see definition in file [vector\\_mode\\_example.vb](#)

Definition at line 26 of file `vector_mode_example.cs`.

### 12.58.2 Member Function Documentation

#### 12.58.2.1 Main()

```
static int Main (
    string[] args ) [inline], [static]
```

Main function for the vector mode example.

#### Parameters

<i>args</i>	An array of command line arguments.
-------------	-------------------------------------



**Returns**

The success status or error code of the function.

The first argument should be the IP Address of a [Galil](#) controller. The second argument should be a path to a file containing vector commands.

Definition at line 35 of file `vector_mode_example.cs`.

References `Examples.GALIL_EXAMPLE_ERROR`, `Examples.GALIL_EXAMPLE_OK`, `gclib.GClose()`, `gclib.GOpen()`, `Examples.PrintError()`, and `Examples.Vector_Mode()`.

The documentation for this class was generated from the following file:

- [vector\\_mode\\_example.cs](#)



# Chapter 13

## File Documentation

### 13.1 arrays.c File Reference

```
#include "gclibo.h"  
#include <stdlib.h>  
#include <string.h>  
#include <stdio.h>  
#include <math.h>  
#include <zlib.h>
```

#### Data Structures

- struct [H\\_ArrayData](#)  
*Structure to create a linked list for array data.*

#### Typedefs

- typedef struct [H\\_ArrayData](#) **ArrayNode**

#### Functions

- void [H\\_InitArrayNode](#) ([ArrayNode](#) \*node)  
*Function to initialize the memory of a new node.*
- [GReturn H\\_AddArray](#) ([ArrayNode](#) \*head, char \*name, char \*data)  
*Add an ArrayData node to the linked list.*
- void [H\\_FreeArrays](#) ([ArrayNode](#) \*node)  
*Frees all memory downstream of node. After passing list head to this function, all memory is freed and the head node is invalid.*
- [GReturn H\\_UploadArrayToList](#) ([GCon](#) g, [ArrayNode](#) \*head, char \*name)  
*Uploads a particular array and adds it to the linked list.*
- [GReturn H\\_CreateArrayNode](#) ([ArrayNode](#) \*head, char \*name)  
*Creates a buffer on the heap to write data, and adds it to the linked list.*
- [GReturn H\\_ArrayAddElement](#) ([ArrayNode](#) \*node, [GCStringIn](#) element)  
*Adds an array element to an array node.*
- [GReturn H\\_DownloadArraysFromList](#) ([GCon](#) g, [ArrayNode](#) \*head, int fail)  
*Walks through the array linked list, downloading each.*
- [GReturn H\\_WriteArrayCsv](#) ([ArrayNode](#) \*head, [GCStringIn](#) file\_path)  
*After filling the array list, this function is called to write out the CSV.*
- [GReturn H\\_ArrayDownloadFromMemory](#) ([GCon](#) g, const char \*array\_data, int fail)  
*Helper function to download a block of arrays to the controller.*

- [GReturn H\\_DownloadData](#) ([GCon](#) g, const char \*data, int fail)  
*Helper function to send a string of commands to the controller, one at a time.*
- char \* [H\\_FindSector](#) (char \*arr, int arr\_size, int index)  
*Function that returns a pointer to the start of the specified sector in the GCB data.*
- [GReturn GCALL GArrayDownloadFile](#) ([GCon](#) g, [GCStringIn](#) file\_path)  
*Array download from file.*
- [GReturn GCALL GArrayUploadFile](#) ([GCon](#) g, [GCStringIn](#) file\_path, [GCStringIn](#) names)  
*Array upload to file.*
- [GReturn GCALL GSetupDownloadFile](#) ([GCon](#) g, [GCStringIn](#) file\_path, [GOption](#) options, [GCStringOut](#) info, [GSize](#) info\_len)  
*Download a saved controller configuration from a file.*

### 13.1.1 Detailed Description

Function calls for uploading and downloading arrays with CSV files. Also contains functions for support of [GSetupDownloadFile\(\)](#).

#### Warning

All helper functions (names beginning with H\_) are subject to change without notice

### 13.1.2 Function Documentation

#### 13.1.2.1 GArrayDownloadFile()

```
GReturn GCALL GArrayDownloadFile (
    GCon g,
    GCStringIn file_path )
```

Array download from file.

Downloads a csv file containing array data at `file_path`. If the arrays don't exist, they will be dimensioned.

#### Parameters

<code>g</code>	Connection's handle.
<code>file_path</code>	Null-terminated string containing the path to the array file.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_arrays.cpp](#) for an example.

Definition at line 380 of file `arrays.c`.

#### 13.1.2.2 GArrayUploadFile()

```
GReturn GCALL GArrayUploadFile (
    GCon g,
    GCStringIn file_path,
    GCStringIn names )
```

Array upload to file.

Uploads the entire controller array table or a subset and saves the data as a csv file specified by `file_path`.

#### Parameters

<code>g</code>	Connection's handle.
----------------	----------------------

## Parameters

<i>file_path</i>	Null-terminated string containing the path to the array file, file will be overwritten if it exists.
<i>names</i>	Null-terminated string containing the arrays to upload, delimited with space. "" or null uploads all arrays listed in LA.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_arrays.cpp](#) for an example.

Definition at line 408 of file arrays.c.

## 13.1.2.3 GSetupDownloadFile()

```
GReturn GCALL GSetupDownloadFile (
    GCon g,
    GCStringIn file_path,
    GOption options,
    GCStringOut info,
    GSize info_len )
```

Download a saved controller configuration from a file.

## Parameters

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the gcb file.
<i>options</i>	Bit mask to determine what configuration data to download. See below for all options.
<i>info</i>	Optional pointer to a buffer to store the controller info. If no info is needed, specify as NULL.
<i>info_len</i>	Length of optional info buffer. If no info is needed, specify as NULL.

## Returns

The success status or error code of the function. If the options parameter is set to 0, the return value will be a bit mask indicating which sectors in the specified GCB are not empty. Otherwise, see [gclib\\_errors.h](#) for possible error values.

## Note

By default, [GSetupDownloadFile\(\)](#) will stop immediately if an error is encountered downloading data. This can be overridden in the options parameter. For example, you may want to override the error if you have a backup from an 8-axis controller and want to restore the parameters for the first 4 axes to a 4-axis controller.

If both *info* and *info\_len* are not NULL, the controller information will be provided regardless of the options parameter. The options parameter is a bit mask. If options is set to 0, [GSetupDownloadFile\(\)](#) will return a bit mask indicating which sectors in the specified GCB are not empty. The following contains a list of all currently available options:

Bit	Value	Function	Description
1	0x0002	Restore parameters	<b>KPA, KIA, KDA</b> , etc...
3	0x0008	Restore variables	Variables are listed by the <b>LV</b> command
4	0x0010	Restore arrays	Arrays are listed by the <b>LA</b> command
5	0x0020	Restore program	The program is listed by the <b>LS</b> command
31	0x8000	Ignore errors	Ignore invalid parameter errors and continue restoring data. <a href="#">GSetupDownloadFile()</a> will still stop immediately if a connection issue or other fatal error is encountered

**Usage example:**

```
GCon g;
GOption opt = 0;
GCStringOut info;
GSize info_len = 4096;
GReturn rc = GOpen("192.168.0.50", &g);
if (rc) return rc;
// Call GSetupDownloadFile() with options set to 0 so we can get the non-empty sector bit mask
opt = GSetupDownloadFile(g, "C:\\path\\to\\gcb\\file.gcb", 0, NULL, NULL);
info = (GCStringOut)malloc(sizeof(GCStringOut) * info_len);
// Call GSetupDownloadFile() with the bit mask returned in the previous function call
rc = GSetupDownloadFile(g, "C:\\path\\to\\gcb\\file.gcb", opt, info, info_len);
printf("Info:\n\n%s", info);
GClose(g);
free(info);
return rc;
```

Definition at line 476 of file arrays.c.

**13.1.2.4 H\_DownloadArraysFromList()**

```
GReturn H_DownloadArraysFromList (
    GCon g,
    ArrayNode * head,
    int fail )
```

Walks through the array linked list, downloading each.

**Warning**

This function will call DA and DM which modifies the controllers' array table. This should NOT be done while running record array (see RA/RC/RD) or while using the MODBUS array sharing feature (see ME). To prevent any possibility of array table issues, dimension all the arrays used in the applications with the appropriate lengths before use and comment out the *array table modification* section below.

Definition at line 136 of file arrays.c.

**13.2 commands.cpp File Reference**

```
#include "examples.h"
#include <iostream>
```

**Functions**

- [GReturn commands](#) (GCon g)  
*Demonstrates various uses of GCommand() and GUtility().*

**13.2.1 Detailed Description**

Function calls for the Command Example Project.

**13.2.2 Function Documentation****13.2.2.1 commands()**

```
GReturn commands (
    GCon g )
```

Demonstrates various uses of [GCommand\(\)](#) and [GUtility\(\)](#).

**Parameters**

<i>g</i>	Connection's handle.
----------	----------------------

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [commands\\_example.cpp](#) for an example.

Definition at line 16 of file `commands.cpp`.

## 13.3 `commands.cs` File Reference

**Data Structures**

- class [Examples](#)

*Provides a class of shared constants and methods for gclib's example projects.*

### 13.3.1 Detailed Description

Function calls for the Command Example Project.

For VB.NET, see definition in file [commands.vb](#)

## 13.4 `Commands.vb` File Reference

**Functions**

- int [Commands](#) (Gclib [gclib](#))

*Demonstrates various uses of [GCommand\(\)](#) And basic controller queries.*

**Variables**

- partial Module [Examples](#)

## 13.5 `commands_example.cpp` File Reference

```
#include "examples.h"
```

**Functions**

- int [main](#) (int *argc*, char \**argv*[])

*Main function for Commands Example.*

### 13.5.1 Detailed Description

See [commands\(\)](#) for implementation of logic

### 13.5.2 Function Documentation

#### 13.5.2.1 `main()`

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file [commands\\_example.cpp](#).

References [G\\_SMALL\\_BUFFER](#).

## 13.6 [commands\\_example.cs](#) File Reference

### Data Structures

- class [Commands\\_Example](#)

*Demonstrates various uses of [GCommand\(\)](#) and basic controller queries.*

### 13.6.1 Detailed Description

See [Commands\(\)](#) for implementation of logic

For VB.NET, see definition in file [commands\\_example.vb](#)

## 13.7 [Commands\\_Example.vb](#) File Reference

### Functions

- int [Main](#) ()

## 13.8 [contour.cpp](#) File Reference

```
#include "examples.h"
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
```



## Functions

- bool `load_buf` (`GCon g`, const `std::vector< int > &positions_A`, const `std::vector< int > &positions_B`, int `capacity`, int `&cmd`)  
*Loads contour buffer with commands from the given text file.*
- `std::vector< int > csv_to_vector` (`ifstream &is`)  
*Converts a file of comma separated values to a vector.*
- `GReturn contour` (`GCon g`, char `*fileA`, char `*fileB`)  
*Record user's training and plays back training through contour mode.*

### 13.8.1 Detailed Description

Function calls for the Contour Example project

### 13.8.2 Function Documentation

#### 13.8.2.1 contour()

```
GReturn contour (
    GCon g,
    char * fileA,
    char * fileB )
```

Record user's training and plays back training through contour mode.

#### Parameters

<code>g</code>	Connection's handle.
<code>fileA</code>	A Path to a text file where training for Axis A will be recorded.
<code>fileB</code>	A Path to a text file where training for Axis B will be recorded.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [contour\\_example.cpp](#) for an example.

Definition at line 20 of file `contour.cpp`.

## 13.9 contour.cs File Reference

### Data Structures

- class [Examples](#)  
*Provides a class of shared constants and methods for gclib's example projects.*

### 13.9.1 Detailed Description

Function calls for the Contour Example Project.

For VB.NET, see definition in file [contour.vb](#)

## 13.10 Contour.vb File Reference

### Functions

- int `Contour` (`Gclib gclib`, string `fileA`, string `fileB`)

## Variables

- `positions_A` As `List< string >`

## 13.11 contour\_example.cpp File Reference

```
#include "examples.h"
#include <iostream>
```

## Functions

- `int main (int argc, char *argv[])`  
*Main function for Commands Example.*

### 13.11.1 Detailed Description

See [contour\(\)](#) for implementation of logic

### 13.11.2 Function Documentation

#### 13.11.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file `commands_example.cpp`.

References `G_SMALL_BUFFER`.

## 13.12 contour\_example.cs File Reference

### Data Structures

- class [Contour\\_Example](#)

*Record user's training and plays back training through contour mode.*

#### 13.12.1 Detailed Description

See [Contour\(\)](#) for implementation of logic

For VB.NET, see definition in file [contour\\_example.vb](#)

## 13.13 Contour\_Example.vb File Reference

### Functions

- int [Main](#) ()

## 13.14 example.py File Reference

### Functions

- def [main](#) ()

## 13.15 examples.cs File Reference

### Data Structures

- class [Examples](#)

*Provides a class of shared constants and methods for gclib's example projects.*

#### 13.15.1 Detailed Description

Shared methods and constants for gclib example projects.

For VB.NET, see definition in file [examples.vb](#)

## 13.16 examples.h File Reference

```
#include "gclib.h"
#include "gclibo.h"
#include <iostream>
#include <cstdio>
```

### Macros

- `#define _CRT_SECURE_NO_WARNINGS`
- `#define GALIL_EXAMPLE_OK 0`
- `#define GALIL_EXAMPLE_ERROR -100`

## Functions

- void `e` (`GReturn rc`)  
*A trivial, C++ style return code check used in [Galil's](#) examples and demos.*
- void `error` (`GCon g`, `GReturn rc`)  
*An example of error handling and debugging information.*
- int `pause` ()  
*Pauses console apps for a user key stroke.*
- `GReturn position_tracking` (`GCon g`, int speed=5000)  
*Puts controller into Position Tracking Mode and accepts user-entered positions.*
- `GReturn jog` (`GCon g`)  
*Puts controller into Jog Mode and accepts user input to adjust the speed.*
- `GReturn vector` (`GCon g`, char \*file)  
*Puts controller into Vector Mode and accepts a file defining vector points.*
- `GReturn ip_assigner` (char \*serial\_num, int address)  
*Assigns controller an IP Adress given a serial number and a 1 byte address.*
- `GReturn commands` (`GCon g`)  
*Demonstrates various uses of [GCommand\(\)](#) and [GUtility\(\)](#).*
- `GReturn motion_complete` (`GCon g`)  
*Uses interrupts to track when the motion of controller is completed.*
- `GReturn message` (`GCon g`)  
*Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.*
- `GReturn record_position` (`GCon g`, char \*fileA, char \*fileB)  
*Record user's training and saves to a text file.*
- `GReturn contour` (`GCon g`, char \*fileA, char \*fileB)  
*Record user's training and plays back training through contour mode.*
- `GReturn remote_server` (const char \*server\_name)  
*Publishes local gcaps server to the network.*
- `GReturn remote_client` ()  
*Lists available remote servers and allows connection to remote server.*

### 13.16.1 Detailed Description

Header file for [Galil](#) `gclib` example projects.

### 13.16.2 Function Documentation

#### 13.16.2.1 `commands()`

`GReturn` `commands` (  
    `GCon g` )

Demonstrates various uses of [GCommand\(\)](#) and [GUtility\(\)](#).

#### Parameters

<code>g</code>	Connection's handle.
----------------	----------------------

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [commands\\_example.cpp](#) for an example.

Definition at line 16 of file `commands.cpp`.

### 13.16.2.2 contour()

```
GReturn contour (
    GCon g,
    char * fileA,
    char * fileB )
```

Record user's training and plays back training through contour mode.

#### Parameters

<i>g</i>	Connection's handle.
<i>fileA</i>	A Path to a text file where training for Axis A will be recorded.
<i>fileB</i>	A Path to a text file where training for Axis B will be recorded.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [contour\\_example.cpp](#) for an example.

Definition at line 20 of file `contour.cpp`.

### 13.16.2.3 e()

```
void e (
    GReturn rc ) [inline]
```

A trivial, C++ style return code check used in [Galil's](#) examples and demos.

Throws GReturn if return value is not G\_NO\_ERROR. See [Commands\\_Example.cpp](#) for example usage and catch() handler.

Definition at line 33 of file `examples.h`.

References G\_NO\_ERROR.

Referenced by `GclibJava::GArrayUpload()`, `ip_assigner()`, and `remote_server()`.

### 13.16.2.4 ip\_assigner()

```
GReturn ip_assigner (
    char * serial_num,
    int address )
```

Assigns controller an IP Address given a serial number and a 1 byte address.

#### Parameters

<i>serial_num</i>	Serial Number of the controller.
<i>address</i>	A 1 byte address that defines the last byte of the IP Address.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [ip\\_assigner\\_example.cpp](#) for an example.

This function will listen on the network for controllers requesting an IP Address. If a detected controller matches the serial number provided by the user, a new IP Address will be assigned based on the first 3 bytes of the detected IP Address combined with the user defined 1 byte address.

Definition at line 26 of file `ip_assigner.cpp`.

References `e()`, `G_SMALL_BUFFER`, `GlpRequests()`, and `string_split()`.

### 13.16.2.5 jog()

```
GReturn jog (
    GCon g )
```

Puts controller into Jog Mode and accepts user input to adjust the speed.

#### Parameters

<i>g</i>	Connection's handle.
----------	----------------------

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [jog\\_example.cpp](#) for an example.

Key	Usage
q	Quit Jogging
a	-2000 counts / second
s	-500 counts / second
d	+500 counts / second
f	+2000 counts / second
r	Direction Reversal

Definition at line 29 of file jog.cpp.

### 13.16.2.6 message()

```
GReturn message (
    GCon g )
```

Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.

#### Parameters

<i>g</i>	Connection's handle.
----------	----------------------

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [message\\_example.cpp](#) for an example.

Definition at line 14 of file message.cpp.

Referenced by Examples::Message().

### 13.16.2.7 motion\_complete()

```
GReturn motion_complete (
    GCon g )
```

Uses interrupts to track when the motion of controller is completed.

#### Parameters

<i>g</i>	Connection's handle.
----------	----------------------

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [motion\\_complete\\_example.cpp](#) for an example.

Definition at line 18 of file `motion_complete.cpp`.

**13.16.2.8 position\_tracking()**

```
GReturn position_tracking (
    GCon g,
    int speed = 5000 )
```

Puts controller into Position Tracking Mode and accepts user-entered positions.

**Parameters**

<i>g</i>	Connection's handle.
<i>speed</i>	Optional speed of the controller in Position Tracking Mode. Default value of 5000.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [position\\_tracking\\_example.cpp](#) for an example.

Definition at line 15 of file `position_tracking.cpp`.

**13.16.2.9 record\_position()**

```
GReturn record_position (
    GCon g,
    char * fileA,
    char * fileB )
```

Record user's training and saves to a text file.

**Parameters**

<i>g</i>	Connection's handle.
<i>fileA</i>	A Path to a text file where training for Axis A will be recorded.
<i>fileB</i>	A Path to a text file where training for Axis B will be recorded.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [record\\_position\\_example.cpp](#) for an example.

Definition at line 20 of file `record_position.cpp`.

**13.16.2.10 remote\_client()**

```
GReturn remote_client ( )
```

Lists available remote servers and allows connection to remote server.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [remote\\_client\\_example](#) for an example.

Key	Usage
q	Quit
s	List available servers on then network
h	List available hardware on the current server
0-9	Connect to server instance by number
l	Connect back to local server

Definition at line 89 of file `remote_client.cpp`.  
References `G_SMALL_BUFFER`.

### 13.16.2.11 `remote_server()`

**GReturn** `remote_server (`  
                   `const char * server_name )`

Publishes local gcaps server to the network.

#### Parameters

<i>Name</i>	to publish server under.
-------------	--------------------------

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See `remote_server_example` for an example.

Key	Usage
q	Quit
p	Publish this server to the network
r	Remove this server from the network

Definition at line 39 of file `remote_server.cpp`.  
References `e()`, `G_SMALL_BUFFER`, and `GPublishServer()`.

### 13.16.2.12 `vector()`

**GReturn** `vector (`  
                   `GCon g,`  
                   `char * file )`

Puts controller into Vector Mode and accepts a file defining vector points.

#### Parameters

<i>g</i>	Connection's handle.
<i>file</i>	A Path to a file that defines vector commands.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See `vector_example.cpp` for an example.

Example text file:

```
VP -2219,-2667
VP -2523,-2832
VP 2844,-1425
VP 728,1971
```



```
VP 2127,183
VP -997,688
VP 725,-1893
VP 527,2899
VP -37,2523
VP 1277,1425
VP 857,2388
VP 1096,-1694
CR 1000,0,90
```

Definition at line 36 of file vector.cpp.

## 13.17 Examples.vb File Reference

### Functions

- void [PrintError](#) (Gclib [gclib](#), Exception ex)

*Prints the exception to the console And queries the controller for the most recent error message.*

### Variables

- const int [GALIL\\_EXAMPLE\\_OK](#) = 0
- const [GALIL\\_EXAMPLE\\_ERROR](#) = -100

## 13.18 Form1.cs File Reference

### Data Structures

- class [MainForm](#)

*Demonstrates using gclib in a Windows Form, including using a second thread to free the GUI.*

### 13.18.1 Detailed Description

Function calls for the C# examples.

For VB.NET, see definition in file [Form1.vb](#)

## 13.19 Form1.vb File Reference

### Data Structures

- class [MainForm](#)

*Demonstrates using gclib in a Windows Form, including using a second thread to free the GUI.*

## 13.20 Galil.h File Reference

```
#include <string>
#include <vector>
```

### Data Structures

- class [Galil](#)

## 13.21 gcl\_datarecord.cpp File Reference

```
#include "gcl_galil.h"
```

## 13.22 gcl\_galil.cpp File Reference

```
#include "gcl_galil.h"
```

### Functions

- void `ec` ([GReturn](#) rc)
- string [AddressConvert](#) (const string &gcl\_address)  
*Takes a GCL address string and returns the equivalent gclib address string.*

## 13.23 gcl\_galil.h File Reference

```
#include "gclibo.h"  
#include <unordered_map>  
#include <sstream>  
#include <iomanip>  
#include <algorithm>  
#include "Galil.h"
```

### Data Structures

- struct [Source](#)
- class [GalilPrivate](#)

### Macros

- #define **MAKEDLL**
- #define **TRAFFICBUF** 4096

### Functions

- void `ec` ([GReturn](#) rc)

## 13.24 gcl\_simple.cpp File Reference

```
#include "Galil.h"  
#include <iostream>
```

### Functions

- int `main` ()

## 13.25 gclib.cs File Reference

### Data Structures

- class [gclib](#)  
*Provides a class that binds to gclib's unmanaged dll. Wraps each call and provides a more user-friendly interface for use in C#.*
- interface [gclib.GDataRecord](#)
- struct [gclib.GDataRecord4000](#)  
*Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0.*

- struct [gclib.GDataRecord52000](#)  
*Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40.*
- struct [gclib.GDataRecord1806](#)  
*Data record struct for DMC-1806 controller.*
- struct [gclib.GDataRecord2103](#)  
*Data record struct for DMC-2103 controllers.*
- struct [gclib.GDataRecord1802](#)  
*Data record struct for DMC-1802 controllers.*
- struct [gclib.GDataRecord30000](#)  
*Data record struct for DMC-30010 controllers.*
- struct [gclib.GDataRecord47000\\_ENC](#)  
*Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields.*
- struct [gclib.GDataRecord47300\\_ENC](#)  
*Data record struct for RIO-47300. Includes encoder fields.*
- struct [gclib.GDataRecord47300\\_24EX](#)  
*Data record struct for RIO-47300 with 24EX I/O daughter board.*
- struct [gclib.GDataRecord47162](#)  
*Data record struct for RIO-47162.*

## Typedefs

- using **UB** = System.Byte
- using **UW** = System.UInt16
- using **SW** = System.Int16
- using **SL** = System.Int32
- using **UL** = System.UInt32

## 13.26 gclib.h File Reference

```
#include "gclib_record.h"
#include "gclib_errors.h"
```

## Macros

- #define **GCLIB\_DLL\_EXPORTED**
- #define [GCALL](#) \_\_stdcall  
*Specify calling convention for Windows.*
- #define [G\\_DR](#) 1  
*Value for [GRecord\(\)](#) method variable for acquiring a data record via DR mode.*
- #define [G\\_QR](#) 0  
*Value for [GRecord\(\)](#) method variable for acquiring a data record via QR mode.*
- #define [G\\_BOUNDS](#) -1  
*For functions that take range options, e.g. [GArrayUpload\(\)](#), use this value for full range.*
- #define [G\\_CR](#) 0  
*For [GArrayUpload\(\)](#), use this value in the delim field to delimit with carriage returns.*
- #define [G\\_COMMA](#) 1  
*For [GArrayUpload\(\)](#), use this value in the delim field to delimit with commas.*
- #define [G\\_PUBLISH\\_SERVER](#) 1  
*For [GPublishServer\(\)](#), use this value to publish server to local network.*
- #define [G\\_REMOVE\\_SERVER](#) 0  
*For [GPublishServer\(\)](#), use this value to remove server from local network.*

- #define `G_UTIL_TIMEOUT` 1  
*GUtility()*, Access to timeout.
- #define `G_UTIL_TIMEOUT_OVERRIDE` 2  
*GUtility()*, read/write access to timeout override.
- #define `G_USE_INITIAL_TIMEOUT` -1  
*GUtility()*, for timeout override. Set `G_UTIL_TIMEOUT_OVERRIDE` to this value to use initial `GOpen()` timeout (`--timeout`).
- #define `G_UTIL_VERSION` 128  
*GUtility()*, get a library version string.
- #define `G_UTIL_INFO` 129  
*GUtility()*, get a connection info string.
- #define `G_UTIL_SLEEP` 130  
*GUtility()*, specify an interval to sleep.
- #define `G_UTIL_ADDRESSES` 131  
*GUtility()*, get a list of available connections.
- #define `G_UTIL_IPREQUEST` 132  
*GUtility()*, get a list of hardware requesting IPs.
- #define `G_UTIL_ASSIGN` 133  
*GUtility()*, assign IP addresses over the network.
- #define `G_UTIL_DEVICE_INITIALIZE` 134  
*GUtility()*, sends CF, CW, EO etc. to initialize the connection. Useful after RS or other reset.
- #define `G_UTIL_PING` 135  
*GUtility()*, uses ICMP ping to determine if an IP address is reachable and assigned.
- #define `G_UTIL_ERROR_CONTEXT` 136  
*GUtility()*, provides additional error context, where available.
- #define `G_UTIL_GCAPS_HOST` 256
- #define `G_UTIL_GCAPS_VERSION` 257  
*GUtility()*, get the version of the `gcaps` server.
- #define `G_UTIL_GCAPS_KEEPALIVE` 258  
*GUtility()*, Deprecated 20210119. No longer functional.
- #define `G_UTIL_GCAPS_ADDRESSES` 259  
*GUtility()*, get a list of available connections from the `gcaps` server.
- #define `G_UTIL_GCAPS_IPREQUEST` 260  
*GUtility()*, get a list of hardware requesting IPs from the `gcaps` server.
- #define `G_UTIL_GCAPS_ASSIGN` 261  
*GUtility()*, assign IP addresses over the network from the `gcaps` server.
- #define `G_UTIL_GCAPS_PING` 262  
*GUtility()*, uses ICMP ping to determine if an IP address is reachable and assigned. Ping sent from the `gcaps` server.
- #define `G_UTIL_GCAPS_LIST_SERVERS` 263  
*GUtility()*, get a list of all available `gcaps` servers on the local network.
- #define `G_UTIL_GCAPS_PUBLISH_SERVER` 264  
*GUtility()*, make local `gcaps` server discoverable by other `gcaps` servers on the local network.
- #define `G_UTIL_GCAPS_SET_SERVER` 265  
*GUtility()*, set the new active `gcaps` server.
- #define `G_UTIL_GCAPS_SERVER_STATUS` 266  
*GUtility()*, get information on the local server's name and if it is published to the local network.
- #define `G_UTIL_GCAPS_REMOTE_CONNECTIONS` 267  
*GUtility()*, get a list of remote addresses connected to local server.
- #define `G_UTIL_GCAPS_SERVER_INFO` 268
- #define `G_UTIL_GCAPS_ADDRESSES_GET_REMEMBERED` 269  
*GUtility()*, returns true if `gcaps` is remembering ip assignments.

- #define [G\\_UTIL\\_GCAPS\\_ADDRESSES\\_SET\\_REMEMBERED](#) 270  
*GUtility()*, sets if gcaps should remember ip assignments.
- #define [G\\_SMALL\\_BUFFER](#) 1024  
Most reads from *Galil* are small. This value will easily hold most, e.g. TH, TZ, etc.
- #define [G\\_HUGE\\_BUFFER](#) 524288  
Most reads from *Galil* hardware are small. This value will hold the largest array or program upload/download possible.
- #define [G\\_LINE\\_BUFFER](#) 80  
For writes, via command interpreter, to the *Galil*.

## Typedefs

- typedef int [GReturn](#)  
Every function returns a value of type *GReturn*. See [gclib\\_errors.h](#) for possible values.
- typedef void \* [GCon](#)  
Connection handle. Unique for each connection in process. Assigned a non-zero value in [GOpen\(\)](#).
- typedef unsigned int [GSize](#)  
Size of buffers, etc.
- typedef int [GOption](#)  
Option integer for various formatting, etc.
- typedef char \* [GCStringOut](#)  
C-string output from the library. Implies null-termination.
- typedef const char \* [GCStringIn](#)  
C-string input to the library. Implies null-termination.
- typedef char \* [GBufOut](#)  
Data output from the library. No null-termination implied. Returned values may be null-terminated, see function documentation for details.
- typedef const char \* [GBufIn](#)  
Data input to the library. No null-termination, function will have a *GSize* to indicate bytes to write .
- typedef unsigned char [GStatus](#)  
Interrupt status byte.
- typedef void \* [GMemory](#)  
Pointer to untyped memory for use in [GUtility\(\)](#).

## Functions

- [GCLIB\\_DLL\\_EXPORTED GReturn GCALL GOpen](#) ([GCStringIn](#) address, [GCon](#) \*g)  
Open a connection to a *Galil* Controller.
- [GCLIB\\_DLL\\_EXPORTED GReturn GCALL GClose](#) ([GCon](#) g)  
Closes a connection to a *Galil* Controller.
- [GCLIB\\_DLL\\_EXPORTED GReturn GCALL GRead](#) ([GCon](#) g, [GBufOut](#) buffer, [GSize](#) buffer\_len, [GSize](#) \*bytes\_read)  
Performs a read on the connection.
- [GCLIB\\_DLL\\_EXPORTED GReturn GCALL GWrite](#) ([GCon](#) g, [GBufIn](#) buffer, [GSize](#) buffer\_len)  
Performs a write on the connection.
- [GCLIB\\_DLL\\_EXPORTED GReturn GCALL GCommand](#) ([GCon](#) g, [GCStringIn](#) command, [GBufOut](#) buffer, [GSize](#) buffer\_len, [GSize](#) \*bytes\_returned)  
Performs a command-and-response transaction on the connection.
- [GCLIB\\_DLL\\_EXPORTED GReturn GCALL GProgramDownload](#) ([GCon](#) g, [GCStringIn](#) program, [GCStringIn](#) preprocessor)  
Downloads a program to the controller's program buffer.
- [GCLIB\\_DLL\\_EXPORTED GReturn GCALL GProgramUpload](#) ([GCon](#) g, [GBufOut](#) buffer, [GSize](#) buffer\_len)  
Uploads a program from the controller's program buffer.

- GCLIB\_DLL\_EXPORTED `GReturn GCALL GArrayDownload` (`GCon g`, `const GCStringIn array_name`, `GOption first`, `GOption last`, `GCStringIn buffer`)  
*Downloads array data to a pre-dimensioned array in the controller's array table.*
- GCLIB\_DLL\_EXPORTED `GReturn GCALL GArrayUpload` (`GCon g`, `const GCStringIn array_name`, `GOption first`, `GOption last`, `GOption delim`, `GBufOut buffer`, `GSize buffer_len`)  
*Uploads array data from the controller's array table.*
- GCLIB\_DLL\_EXPORTED `GReturn GCALL GRecord` (`GCon g`, `union GDataRecord *record`, `GOption method`)  
*Provides a fresh copy of the controller's data record. Data is cast into a union, `GDataRecord`.*
- GCLIB\_DLL\_EXPORTED `GReturn GCALL GMessage` (`GCon g`, `GCStringOut buffer`, `GSize buffer_len`)  
*Provides access to unsolicited messages from the controller.*
- GCLIB\_DLL\_EXPORTED `GReturn GCALL GInterrupt` (`GCon g`, `GStatus *status_byte`)  
*Provides access to PCI and UDP interrupts from the controller.*
- GCLIB\_DLL\_EXPORTED `GReturn GCALL GFirmwareDownload` (`GCon g`, `GCStringIn filepath`)  
*Upgrade firmware.*
- GCLIB\_DLL\_EXPORTED `GReturn GCALL GUtility` (`GCon g`, `GOption request`, `GMemory memory1`, `GMemory memory2`)  
*Provides read/write access to driver settings and convenience features based on the request variable.*

### 13.26.1 Detailed Description

Defines the interface for the Galil C Library (GCLIB).

### 13.26.2 Function Documentation

#### 13.26.2.1 GArrayDownload()

```
GCLIB_DLL_EXPORTED GReturn GCALL GArrayDownload (
    GCon g,
    const GCStringIn array_name,
    GOption first,
    GOption last,
    GCStringIn buffer )
```

Downloads array data to a pre-dimensioned array in the controller's array table.

#### Warning

The array must already exist on the controller and be sufficient dimension to hold the desired array data, e.g. via DM.

#### Parameters

<i>g</i>	Connection's handle.
<i>array_name</i>	Null-terminated string containing the name of the array to download. Must match the array name used in DM.
<i>first</i>	The first element of the array for sub-array downloads. <code>G_BOUNDS</code> to omit.
<i>last</i>	The last element of the array for sub-array downloads. <code>G_BOUNDS</code> to omit.
<i>buffer</i>	Buffer containing the null-terminated data to be sent to the controller. The array data may be separated with <i>carriage return</i> , <i>carriage return + line feed</i> , or a <i>comma</i> . No spaces.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_arrays.cpp](#) for an example.

**13.26.2.2 GArrayUpload()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GArrayUpload (
    GCon g,
    const GCStringIn array_name,
    GOption first,
    GOption last,
    GOption delim,
    GBufOut buffer,
    GSize buffer_len )
```

Uploads array data from the controller's array table.

**Parameters**

<i>g</i>	Connection's handle.
<i>array_name</i>	Null-terminated string containing the name of the array to upload.
<i>first</i>	The first element of the array for sub-array uploads. <code>G_BOUNDS</code> to omit.
<i>last</i>	The last element of the array for sub-array uploads. <code>G_BOUNDS</code> to omit.
<i>delim</i>	Sets the delimiter between array elements in the returned data, <code>G_CR</code> specifies carriage return, <code>G_COMMA</code> specifies comma.
<i>buffer</i>	Buffer to receive the uploaded data. The data will be null terminated unless function returns <code>G_BAD_LOST_DATA</code> due to the buffer being too small to hold the data.
<i>buffer_len</i>	The length of the receive buffer.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_arrays.cpp](#) for an example.

**13.26.2.3 GClose()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GClose (
    GCon g )
```

Closes a connection to a [Galil](#) Controller.

**Attention**

*gclib* requires that `GClose()` be called whenever a program is finished with a controller. This includes when a program closes. A rule of thumb is that for every `GOpen()` call on a given connection, a `GClose()` call should be found on every code path. Failing to call `GClose()` may cause controller resources to not be released or can hang the process if there are outstanding asynchronous operations. The latter can occur, for example, if a call to `GRead()` times out and the process exits without calling `GClose()`. In this case, `GRead()` still has an outstanding asynchronous read pending. `GClose()` will terminate this operation allowing the process to exit correctly.

**Parameters**

<i>g</i>	Connection's handle.
----------	----------------------

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_examples.cpp](#) for an example.

**13.26.2.4 GCommand()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GCommand (
    GCon g,
    GCStringIn command,
    GBufOut buffer,
    GSize buffer_len,
    GSize * bytes_returned )
```

Performs a *command-and-response* transaction on the connection.

**Parameters**

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller. The library will append a carriage return to the command string.
<i>buffer</i>	Buffer for the response. Will be filled with the response from the controller. The data will be null terminated unless the function returns <code>G_BAD_LOST_DATA</code> due to the buffer being too small to hold the data.
<i>buffer_len</i>	The size of the response buffer.
<i>bytes_returned</i>	The size of the data returned from the controller. This does not include null termination. This argument may be null if the value is not desired.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

**13.26.2.5 GFirmwareDownload()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GFirmwareDownload (
    GCon g,
    GCStringIn filepath )
```

Upgrade firmware.

**Parameters**

<i>g</i>	Connection's handle.
<i>filepath</i>	The full file path to the Galil-supplied firmware hex file. See <a href="http://www.galil.com/downloads/firmware">http://www.galil.com/downloads/firmware</a>

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

```
ec(GInfo(g, buf, sizeof(buf))); //get conntroller info
cout << buf << '\n'; //print the info
ec(GFirmwareDownload(g, "F:/1806.dmc/dmc-1806-r11a.hex"));
ec(GInfo(g, buf, sizeof(buf))); //get the info again
cout << buf << '\n';
// example output:
// GALILPCI1, DMC1846 Rev 1.1a-CM, 4232
// GALILPCI1, DMC1846 Rev 1.1a, 4232
```



### 13.26.2.6 GInterrupt()

```
GCLIB_DLL_EXPORTED GReturn GCALL GInterrupt (
    GCon g,
    GStatus * status_byte )
```

Provides access to PCI and UDP interrupts from the controller.

Interrupts can be generated automatically by the firmware on important events via `EI` (Enable Interrupt) or by the user in embedded DMC code via `UI` (User Interrupt). To use this function, `-s EI` must be used in the `GOpen()` address string to subscribe to interrupts.

#### Parameters

<code>g</code>	Connection's handle.
<code>status_byte</code>	A pointer to a <code>GStatus</code> to receive the status byte.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

`GInterrupt()` will block until an interrupt is received, or the function times out.

#### Note

If this function is called with a timeout of zero, a non-blocking read is performed. If interrupt data is waiting in the interrupt queue, the oldest byte will be popped off the queue. If there is no interrupt data queued, but there is data waiting in the socket or PCI FIFO, one read will be performed to process the waiting data. If new data is still not found after these two attempts, `G_GCLIB_NON_BLOCKING_READ_EMPTY` will be returned.

See [x\\_ginterrupt.cpp](#) for an example. See [x\\_nonblocking.cpp](#) for an example of non-blocking usage.

### 13.26.2.7 GMessage()

```
GCLIB_DLL_EXPORTED GReturn GCALL GMessage (
    GCon g,
    GStringOut buffer,
    GSize buffer_len )
```

Provides access to unsolicited messages from the controller.

To use this function, `-s MG` must be used in the `GOpen()` address string to subscribe to messages. Unsolicited bytes must be flagged by the high-bit setting, `CW 1`. The driver will automatically set this when subscribing to messages. The user should not overwrite this setting.

Unsolicited messages are data generated by the controller that are not in response to a command, a data record, or an interrupt. Examples follow.

1. Data generated by the `MG` command from embedded code. `MG` sent from the host is solicited.
2. Any command in an embedded program that returns data, e.g. `TP, RP, var=?`
3. A run time error in an embedded program, e.g. `?55 i=var`

#### Note

Messages are unframed byte streams. There is no guarantee that the user will get complete messages or single messages in a call to `GMessage()`. If multiple messages have been sent from the controller since the last call to `GMessage()`, they will all be placed in the buffer, separated by newline characters.

#### Parameters

<code>g</code>	Connection's handle.
<code>buffer</code>	The buffer to write the message data. The buffer will be null terminated.
<code>buffer_len</code>	The length of the user's buffer.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

[GMessage\(\)](#) will block until a message is received, or the function times out.

**Note**

If this function is called with a timeout of zero, a non-blocking read is performed. If message data has been processed since the last time the function was called, this data will be returned. If there is no processed message data, but there is data waiting in the socket or PCI FIFO, one read will be performed to process the waiting data. If new data is still not found after these two attempts, `G_GCLIB_NON_BLOCKING_READ_EMPTY` will be returned.

**Warning**

When sending message streams through [gcaps](#), the following non-printable bytes are illegal, `$00-$07` and `$10-$17`. These bytes may be routed to a third party device such as an HMI or display panel. See MG and CF.

See [x\\_gmessage.cpp](#) for an example. See [x\\_nonblocking.cpp](#) for an example of non-blocking usage.

**13.26.2.8 GOpen()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GOpen (
    GCStringIn address,
    GCon * g )
```

Open a connection to a [Galil](#) Controller.

**Parameters**

<i>address</i>	Null-terminated address string. See table below.
<i>g</i>	Pointer to user's <code>GCon</code> variable. On success, the library will fill the user's variable with the handle to use for the rest of the connection. A valid <code>g</code> value is nonzero.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

address switch	Meaning	Arguments (default), other options	Examples
<code>--address</code>	<b>Simple address to hardware</b>	<i>IP address, PCI, COM port</i>	<code>--address COM1</code>
<code>-a</code>	shorthand for <code>--address</code>	See <i>Address Ranges</i> below	<code>-a GALILPCI1</code>
{no switch}	<code>--address</code> is implicit for any lone token		192.168.0.42
<code>--baud</code>	<b>Baud rate</b>	(115200), <i>valid baud...</i>	<code>COM2 --baud 19200</code>
<code>-b</code>	shorthand for <code>--baud</code>		<code>COM3 -b 38400</code>
<code>--command</code>	<b>Command-and-response socket protocol</b>	(TCP), UDP	192.168.0.42 <code>--command TCP</code>
<code>-c</code>	shorthand for <code>--command</code>		192.168.0.42 <code>-c UDP</code>
<code>--direct</code>	<b>Connect directly to hardware instead of via <a href="#">gcaps</a></b>		<code>-a GALILPCI2 --direct</code>
<code>-d</code>	shorthand for <code>--direct</code>		<code>GALILPCI2 -d</code>

address switch	Meaning	Arguments (default), other options	Examples
--subscribe	<b>Subscribe to messages, data records, and/or interrupts</b>	(NONE), MG, DR, EI, ALL	192.168.0.42 --subscribe MG
-s	shorthand for --subscribe		192.168.0.42 -s DR -s EI
--timeout	<b>timeout in ms</b>	(5000), 0-65535	192.168.0.42 --timeout 5000
-t	shorthand for --timeout		GALILPCI2 -t 500
--unsolicited	<b>Unsolicited socket protocol</b>	(UDP), NONE	192.168.0.42 --unsolicited NONE
-u	shorthand for --unsolicited		192.168.1.42 -u UDP
<b>The following address switches are deprecated and will be unavailable starting July 1st, 2020.</b>			
--p1	<b>Primary port for command-and-response traffic</b>	(23), <i>valid port number</i>	192.168.0.42 --p1 5000
--p2	<b>Secondary port for unsolicited traffic</b>	(60007), <i>valid port number</i>	192.168.0.42 --p2 5000

Operating System	Address Range	Notes
Windows	COM1 - COM256	RS232 and USB-to-serial
Linux	/dev/ttyS0 - /dev/ttyS255	RS232
Linux	/dev/ttyUSB0 - /dev/ttyUSB255	USB-to-serial, e.g. DMC-4103
Windows	GALILPCI1 - GALILPCI8	PCI
Linux	/dev/galilpci0 - /dev/galilpci7	PCI

See [x\\_examples.cpp](#) for an example.

When connecting to a network device, if the command-and-response socket is opened successfully but the unsolicited socket fails, [GOpen\(\)](#) will still complete successfully. This allows connection to a [Galil](#) controller when only one Ethernet handle is available. Unsolicited traffic will not be accessible in this case.

### 13.26.2.9 GProgramDownload()

```
GCLIB_DLL_EXPORTED GReturn GCALL GProgramDownload (
    GCon g,
    GCStringIn program,
    GCStringIn preprocessor )
```

Downloads a program to the controller's program buffer.

#### Parameters

<i>g</i>	Connection's handle.
<i>program</i>	Null-terminated program for download.
<i>preprocessor</i>	Options string for preprocessing the program before sending it to the controller. Null allows the library to use defaults for the download. See the <a href="#">Program Preprocessor</a> documentation for options.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_programs.cpp](#) for an example.

**13.26.2.10 GProgramUpload()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GProgramUpload (
    GCon g,
    GBufOut buffer,
    GSize buffer_len )
```

Uploads a program from the controller's program buffer.

**Parameters**

<i>g</i>	Connection's handle.
<i>buffer</i>	Buffer to receive the controller's program. The data will be null terminated unless function returns <code>G_BAD_LOST_DATA</code> due to the buffer being too small to hold the data.
<i>buffer_len</i>	The length of the receive buffer.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_programs.cpp](#) for an example.

**13.26.2.11 GRead()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GRead (
    GCon g,
    GBufOut buffer,
    GSize buffer_len,
    GSize * bytes_read )
```

Performs a read on the connection.

**Parameters**

<i>g</i>	Connection's handle.
<i>buffer</i>	The user's read buffer.
<i>buffer_len</i>	The length of the user's read buffer.
<i>bytes_read</i>	Pointer to a <code>GSize</code> which will be filled with the number of bytes read upon return.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

**Warning**

This function is deprecated and will be removed in a future `gclib` version. Please contact [Galil](#) for needs not covered by the other `gclib` functions.

Unsolicited messages may be returned in the read data. The high bit of each message byte will be set unless the user changes the CW setting. Interrupts and Data Records are always filtered from a read.

See [x\\_gread\\_gwrite.cpp](#) for an example.

**13.26.2.12 GRecord()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GRecord (
    GCon g,
```

```
union GDataRecord * record,
    GOption method )
```

Provides a fresh copy of the controller's data record. Data is cast into a union, [GDataRecord](#).

#### Parameters

<i>g</i>	Connection's handle.
<i>record</i>	A pointer to the user's DataRecord union to hold the copy.
<i>method</i>	Determines the method for acquiring the data. <ul style="list-style-type: none"> <li>• <code>G_QR</code>: QR is used via command-and-response.</li> <li>• <code>G_DR</code>: DR is used for asynchronous acquisition.</li> </ul>

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

When using `G_DR`, the asynchronous data record must already be set up.

- `-s DR` must be used in the [GOpen\(\)](#) `address` string to subscribe to records. The driver will automatically set the second argument of `DR`, where applicable.
- [GRecordRate\(\)](#) should be issued to set `DR` to an appropriate interval, `n`. The interval must be no faster than the rate at which [GRecord\(\)](#) is called.
- If [GRecord\(\)](#) is called more slowly than the data record rate, stale data will be returned until [GRecord\(\)](#) has been called once for each record sent by the controller.

[GRecord\(\)](#) will block until the data record is received, or the transaction times out.

#### Note

If this function is called with a timeout of zero and the `G_DR` method, a non-blocking read is performed. If a data record has been processed since the last time the function was called, this data will be returned. If there is not a processed data record, but there is data waiting in the socket or PCI FIFO, one read will be performed to process the waiting data. If new data is still not found after these two attempts, `G_GCLIB_↔NON_BLOCKING_READ_EMPTY` will be returned.

See [x\\_grecord.cpp](#) for an example. See [x\\_nonblocking.cpp](#) for an example of non-blocking usage.

#### 13.26.2.13 GUtility()

```
GCLIB_DLL_EXPORTED GReturn GCALL GUtility (
    GCon g,
    GOption request,
    GMemory memory1,
    GMemory memory2 )
```

Provides read/write access to driver settings and convenience features based on the request variable.

#### Note

The open source library, [gclibo.h](#), has wrappers for most of these utilities.

#### Parameters

<i>g</i>	Connection's handle.
<i>request</i>	Defines the request. Input/Output and type of memory are implicit in the value of request. The following lists the supported request values.

- [G\\_UTIL\\_TIMEOUT](#) Read initial timeout value, as specified in [GOpen\(\)](#) via `--timeout` switch.
  - `memory1` is output and must be an `unsigned short*`.
  - `memory2` is ignored, use null.
- [G\\_UTIL\\_TIMEOUT\\_OVERRIDE](#) See [GTimeout\(\)](#). Write/Read override timeout value.
  - `memory1` is input. If nonnull, value must be a `short*` holding the override, in milliseconds, for the timeout. Write `G_USE_INITIAL_TIMEOUT` to use initial timeout. If null, no write occurs.
  - `memory2` is output. If nonnull, value must be a `short*` which will be filled with the current override. `G_USE_INITIAL_TIMEOUT` indicates initial timeout used. If null, no read occurs. `memory2` is processed before `'memory1'`.
- [G\\_UTIL\\_VERSION](#) See [GVersion\(\)](#). Returns the library version. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is output, and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_INFO](#) See [GInfo\(\)](#). Returns information about the connection.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_SLEEP](#) See [GSleep\(\)](#). Platform-independent, non-busy, sleep. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is input and must be an `unsigned int*`, units are milliseconds.
  - `memory2` is ignored, use null.
- [G\\_UTIL\\_ADDRESSES](#) see [GAddresses\(\)](#). Provides a `\n` delimited listing of all available IP addresses, PCI addresses, and COM ports. A valid connection (`g`) is not necessary, i.e. `g` may be null. The suffix `-d` will be appended to each address to indicate these addresses are available via direct connection. See [G\\_UTIL\\_↔GCAPS\\_ADDRESSES](#) for addresses through [gcaps](#).
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_IPREQUEST](#) see [GIpRequests\(\)](#). Listens and returns a `\n` delimited listing of [Galil](#) MAC addresses sending BOOT-P or DHCP requests. The function will listen, and block, for roughly 5 seconds. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_ASSIGN](#) see [GAssign\(\)](#). Provides a method to assign an IP address given a [Galil](#) MAC address. A valid connection (`g`) is not necessary, i.e. `g` may be null.
  - `memory1` is input and must be a `char*` containing the null terminated address that is to be assigned. e.g. `"192.168.0.43"`.
  - `memory2` is input and must be a `char*` containing the null terminated controller MAC address. e.g. `"00:50:4C:20:01:23"`.
- [G\\_UTIL\\_DEVICE\\_INITIALIZE](#) Provides a method to reinitialize a connection after a reset, e.g. an RS command. Depending on the device type, the appropriate commands will be sent to configure the communication bus for optimal performance.
  - `memory1` is ignored, use null.

- `memory2` is ignored, use `null`.
- [G\\_UTIL\\_PING](#) Uses ICMP ping to determine if an IP address is reachable and assigned. A valid connection (`g`) is not necessary, i.e. `g` may be `null`.
  - `memory1` is input and must be a `char*` containing the null terminated address that is to be pinged. e.g. `"192.168.0.43"`.
  - `memory2` is output and must be an `int*`. The value will be set to zero if the ping times out, and nonzero if a ping reply is returned.
- [G\\_UTIL\\_ERROR\\_CONTEXT](#) More error detail for the last error on `GCon`, where available. The internal error message is cleared upon read.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.

The following request values are for use with a `@ref gcaps` server.

- [G\\_UTIL\\_GCAPS\\_VERSION](#) see [GVersion\(\)](#). Returns the `gcaps` server version. A valid connection (`g`) is not necessary, i.e. `g` may be `null`. This operation will connect to the server to determine the version.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_GCAPS\\_ADDRESSES](#) see [GAddresses\(\)](#). Provides a `\n` delimited listing of all available IP addresses, PCI addresses, and COM ports as available from the `gcaps` server. A valid connection (`g`) is not necessary, i.e. `g` may be `null`.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_GCAPS\\_IPREQUEST](#) see [GIpRequests\(\)](#). Connects to `gcaps` and returns a `\n` delimited listing of `Galil` MAC addresses sending BOOT-P or DHCP requests. A valid connection (`g`) is not necessary, i.e. `g` may be `null`.
  - `memory1` is output and must be a `char*`. Data will be null terminated, even if the data must be truncated to do so.
  - `memory2` is input and must be an `unsigned int*` holding the length of the buffer in `memory1`.
- [G\\_UTIL\\_GCAPS\\_ASSIGN](#) see [GAssign\(\)](#). Provides a method to assign an IP address through `gcaps` given a `Galil` MAC address. A valid connection (`g`) is not necessary, i.e. `g` may be `null`.
  - `memory1` is input and must be a `char*` containing the null terminated address that is to be assigned. e.g. `"192.168.0.43"`.
  - `memory2` is input and must be a `char*` containing the null terminated controller MAC address. e.g. `"00:50:4C:20:01:23"`.
- [G\\_UTIL\\_GCAPS\\_PING](#) Uses ICMP ping to determine if an IP address is reachable and assigned. Ping sent from the `gcaps` server. A valid connection (`g`) is not necessary, i.e. `g` may be `null`.
  - `memory1` is input and must be a `char*` containing the null terminated address that is to be pinged. e.g. `"192.168.0.43"`.
  - `memory2` is output and must be an `int*`. The value will be set to zero if the ping times out, and nonzero if a ping reply is returned.

**Parameters**

<i>memory1</i>	An untyped pointer to data required for request. The data type is defined by the request variable.
<i>memory2</i>	An untyped pointer to data required for request. The data type is defined by the request variable.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See the following functions from gclibo, the open source portion, for implementation of several [GUtility\(\)](#) requests.:

- [GAddresses\(\)](#)
- [GAssign\(\)](#)
- [GInfo\(\)](#)
- [GIpRequests\(\)](#)
- [GSleep\(\)](#)
- [GTimeout\(\)](#)
- [GVersion\(\)](#)

Referenced by [GAddresses\(\)](#), [GAssign\(\)](#), [GIpRequests\(\)](#), [GListServers\(\)](#), [GRemoteConnections\(\)](#), [GServer↔Status\(\)](#), [GSetServer\(\)](#), [GSleep\(\)](#), and [GVersion\(\)](#).

**13.26.2.14 GWrite()**

```
GCLIB_DLL_EXPORTED GReturn GCALL GWrite (
    GCon g,
    GBufIn buffer,
    GSize buffer_len )
```

Performs a write on the connection.

**Parameters**

<i>g</i>	Connection's handle.
<i>buffer</i>	The user's write buffer. To send a <a href="#">Galil</a> command, a terminating carriage return is usually required.
<i>buffer_len</i>	The length of the data in the buffer.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values. If `G_NO_ERROR` is returned, all bytes were written.

**Warning**

This function is deprecated and will be removed in a future gclib version. Please contact [Galil](#) for needs not covered by the other gclib functions.

See [x\\_gread\\_gwrite.cpp](#) for an example.

## 13.27 gclib.py File Reference

**Data Structures**

- class [gclib.GclibError](#)
- class [gclib.py](#)



## 13.28 gclib.vb File Reference

### Data Structures

- class [Gclib](#)  
*Provides a class that binds to gclib's unmanaged dll. Wraps each call and provides a more user-friendly interface for use in Visual Basic.*
- interface [Gclib.GDataRecord](#)
- struct [Gclib.GDataRecord4000](#)
- struct [Gclib.GDataRecord52000](#)
- struct [Gclib.GDataRecord1806](#)
- struct [Gclib.GDataRecord2103](#)
- struct [Gclib.GDataRecord1802](#)
- struct [Gclib.GDataRecord30000](#)
- struct [Gclib.GDataRecord47000\\_ENC](#)
- struct [Gclib.GDataRecord47300\\_ENC](#)
- struct [Gclib.GDataRecord47300\\_24EX](#)
- struct [Gclib.GDataRecord47162](#)

### Typedefs

- using **UB** = System.Byte
- using **UW** = System.UInt16
- using **SW** = System.Int16
- using **SL** = System.Int32
- using **UL** = System.UInt32
- using **GReturn** = System.Int32
- using **GCon** = System.IntPtr
- using **GSize** = System.UInt32
- using **GCStringOut** = System.Text.StringBuilder
- using **GCStringIn** = System.String
- using **GBufOut** = System.Text.StringBuilder
- using **GBufIn** = System.String
- using **GStatus** = System.Byte

### Variables

- using **System**
- Module **LibraryPath**
- const string **GclibDllPath** = "C:\Program Files[\Galil\gclib\dl]x86\gclib.dll"
- const string **GcliboDllPath** = "C:\Program Files[\Galil\gclib\dl]x86\gclibo.dll"

## 13.29 gclib\_errors.h File Reference

### Macros

- #define [G\\_NO\\_ERROR](#) 0  
*Return value if function succeeded.*
- #define **G\_NO\_ERROR\_S** "no error"
- #define [G\\_GCLIB\\_ERROR](#) -1  
*General library error. Indicates internal API caught an unexpected error. Contact [Galil](#) support if this error is returned, [softwaresupport@galil.com](mailto:softwaresupport@galil.com).*
- #define **G\_GCLIB\_ERROR\_S** "gclib unexpected error"
- #define [G\\_GCLIB\\_UTILITY\\_ERROR](#) -2  
*An invalid request value was specified to GUtility.*

- #define **G\_GCLIB\_UTILITY\_ERROR\_S** "invalid request value or bad arguments were specified to [GUtility\(\)](#)"
- #define **G\_GCLIB\_UTILITY\_IP\_TAKEN** -3
 

*The IP cannot be assigned because ping returned a reply.*
- #define **G\_GCLIB\_UTILITY\_IP\_TAKEN\_S** "ip address is already taken by a device on the network"
- #define **G\_GCLIB\_NON\_BLOCKING\_READ\_EMPTY** -4
 

*GMessage, GInterrupt, and GRecord can be called with a zero timeout. If there wasn't data waiting in memory, this error is returned.*
- #define **G\_GCLIB\_NON\_BLOCKING\_READ\_EMPTY\_S** "data was not waiting for a zero-timeout read"
- #define **G\_GCLIB\_POLLING\_FAILED** -5
 

*GWaitForBool out of polling trials.*
- #define **G\_GCLIB\_POLLING\_FAILED\_S** "exit condition not met in specified polling period"
- #define **G\_TIMEOUT** -1100
 

*Operation timed out. Timeout is set by the `-timeout` option in [GOpen\(\)](#) and can be overridden by [GSetting\(\)](#).*
- #define **G\_TIMEOUT\_S** "device timed out"
- #define **G\_OPEN\_ERROR** -1101
 

*Device could not be opened. E.G. Serial port or PCI device already open.*
- #define **G\_OPEN\_ERROR\_S** "device failed to open"
- #define **G\_ALREADY\_OPEN** -1111
 

*Serial or PCI file has a flock placed on it, presumably by another gclib connection.*
- #define **G\_ALREADY\_OPEN\_S** "Serial or PCI port already open"
- #define **G\_READ\_ERROR** -1103
 

*Device read failed. E.G. Socket was closed by remote host. See [G\\_UTIL\\_GCAPS\\_KEEPALIVE](#).*
- #define **G\_READ\_ERROR\_S** "device read error"
- #define **G\_WRITE\_ERROR** -1104
 

*Device write failed. E.G. Socket was closed by remote host. See [G\\_UTIL\\_GCAPS\\_KEEPALIVE](#).*
- #define **G\_WRITE\_ERROR\_S** "device write error"
- #define **G\_INVALID\_PREPROCESSOR\_OPTIONS** -1204
 

*GProgramDownload was called with a bad preprocessor directive.*
- #define **G\_INVALID\_PREPROCESSOR\_OPTIONS\_S** "preprocessor did not recognize options"
- #define **G\_COMMAND\_CALLED\_WITH\_ILLEGAL\_COMMAND** -1106
 

*GCommand() was called with an illegal command, e.g. ED, DL or QD.*
- #define **G\_COMMAND\_CALLED\_WITH\_ILLEGAL\_COMMAND\_S** "illegal command passed to command call"
- #define **G\_DATA\_RECORD\_ERROR** -1107
 

*Data record error, e.g. DR attempted on serial connection.*
- #define **G\_DATA\_RECORD\_ERROR\_S** "data record error"
- #define **G\_UNSUPPORTED\_FUNCTION** -1109
 

*Function cannot be called on this bus. E.G. [GInterrupt\(\)](#) on serial.*
- #define **G\_UNSUPPORTED\_FUNCTION\_S** "function not supported on this communication bus"
- #define **G\_FIRMWARE\_LOAD\_NOT\_SUPPORTED** -1110
 

*Firmware is not supported on this bus, e.g. Ethernet for the DMC-21x3 series.*
- #define **G\_FIRMWARE\_LOAD\_NOT\_SUPPORTED\_S** "firmware cannot be loaded on this communication bus to this hardware"
- #define **G\_ARRAY\_NOT\_DIMENSIONED** -1200
 

*Array operation was called on an array that was not in the controller's array table, see LA command.*
- #define **G\_ARRAY\_NOT\_DIMENSIONED\_S** "array not dimensioned on controller or wrong size"
- #define **G\_CONNECTION\_NOT\_ESTABLISHED** -1201
 

*Function was called with no connection.*
- #define **G\_CONNECTION\_NOT\_ESTABLISHED\_S** "connection to hardware not established"
- #define **G\_ILLEGAL\_DATA\_IN\_PROGRAM** -1202
 

*Data to download not valid, e.g. \ in data.*
- #define **G\_ILLEGAL\_DATA\_IN\_PROGRAM\_S** "illegal ASCII character in program"

- #define [G\\_UNABLE\\_TO\\_COMPRESS\\_PROGRAM\\_TO\\_FIT](#) -1203  
*Program preprocessor could not compress the program within the user's constraints.*
- #define [G\\_UNABLE\\_TO\\_COMPRESS\\_PROGRAM\\_TO\\_FIT\\_S](#) "program cannot be compressed to fit on the controller"
- #define [G\\_BAD\\_RESPONSE\\_QUESTION\\_MARK](#) -10000  
*Operation received a ?, indicating controller has a TC error.*
- #define [G\\_BAD\\_RESPONSE\\_QUESTION\\_MARK\\_S](#) "question mark returned by controller"
- #define [G\\_BAD\\_VALUE\\_RANGE](#) -10002  
*Bad value or range, e.g. GCon g variable passed to function was bad.*
- #define [G\\_BAD\\_VALUE\\_RANGE\\_S](#) "value passed to function was bad or out of range"
- #define [G\\_BAD\\_FULL\\_MEMORY](#) -10003  
*Not enough memory for an operation, e.g. all connections allowed for a process already taken.*
- #define [G\\_BAD\\_FULL\\_MEMORY\\_S](#) "operation could not complete because of a memory error"
- #define [G\\_BAD\\_LOST\\_DATA](#) -10004  
*Lost data, e.g. GCommand() response buffer was too small for the controller's response.*
- #define [G\\_BAD\\_LOST\\_DATA\\_S](#) "data was lost due to buffer or fifo limitations"
- #define [G\\_BAD\\_FILE](#) -10005  
*Bad file path, bad file contents, or bad write.*
- #define [G\\_BAD\\_FILE\\_S](#) "file was not found, contents are invalid, or write failed"
- #define [G\\_BAD\\_ADDRESS](#) -10006  
*Bad address.*
- #define [G\\_BAD\\_ADDRESS\\_S](#) "a bad address was specified in open"
- #define [G\\_BAD\\_FIRMWARE\\_LOAD](#) -10008  
*Bad firmware upgrade.*
- #define [G\\_BAD\\_FIRMWARE\\_LOAD\\_S](#) "Firmware upgrade failed"
- #define [G\\_GCAPS\\_OPEN\\_ERROR](#) -20000  
*gcaps connection couldn't open. Server is not running or is not reachable.*
- #define [G\\_GCAPS\\_OPEN\\_ERROR\\_S](#) "gcaps connection could not be opened"
- #define [G\\_GCAPS\\_SUBSCRIPTION\\_ERROR](#) -20002  
*GMessage(), GRecord(), GInterrupt() called on a connection without –subscribe switch.*
- #define [G\\_GCAPS\\_SUBSCRIPTION\\_ERROR\\_S](#) "function requires subscription not specified in GOpen()"

### 13.29.1 Detailed Description

Defines values for the [Galil C Library](#) return codes and error strings.

## 13.30 gclib\_record.h File Reference

```
#include <stdint.h>
```

### Data Structures

- struct [GDataRecord4000](#)  
*Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0.*
- struct [GDataRecord52000](#)  
*Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40.*
- struct [GDataRecord1806](#)  
*Data record struct for DMC-1806 controller.*
- struct [GDataRecord2103](#)  
*Data record struct for DMC-2103 controllers.*
- struct [GDataRecord1802](#)

- struct [GDataRecord30000](#)  
*Data record struct for DMC-30010 controllers.*
- struct [GDataRecord47000\\_ENC](#)  
*Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields.*
- struct [GDataRecord47300\\_ENC](#)  
*Data record struct for RIO-47300. Includes encoder fields.*
- struct [GDataRecord47300\\_24EX](#)  
*Data record struct for RIO-47300 with 24EX I/O daughter board.*
- struct [GDataRecord47162](#)  
*Data record struct for RIO-47162.*
- union [GDataRecord](#)  
*Data record union, containing all structs and a generic byte array accessor.*

## Macros

- #define [GALILDATARECORDMAXLENGTH](#) 512  
*Max size for any Galil data record, equal to dual port ram size of PCI.*

## Typedefs

- typedef uint8\_t **UB**
- typedef uint16\_t **UW**
- typedef int16\_t **SW**
- typedef int32\_t **SL**
- typedef uint32\_t **UL**

### 13.30.1 Detailed Description

Defines a union for data records. Each supported controller has a struct member in the union with named record types. Offsets into the data record can also be used by referencing the member `byte_array`.

## 13.31 GclibJava.java File Reference

### Data Structures

- class [GclibJava](#)
- interface [GclibJava.Gclib](#)
- interface [GclibJava.Gclibo](#)

### Packages

- package [gclibjava](#)

## 13.32 GclibJavaException.java File Reference

### Data Structures

- class [GclibJavaException](#)

### Packages

- package [gclibjava](#)

## 13.33 gclibo.c File Reference

```
#include "gclibo.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <math.h>
#include <time.h>
```

### Functions

- GCLIB\_DLL\_EXPORTED void **GCALL GSleep** (unsigned int timeout\_ms)  
*Uses [GUtility\(\)](#) and [G\\_UTIL\\_SLEEP](#) to provide a blocking sleep call which can be useful for timing-based chores.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GVersion** (GCStringOut ver, GSize ver\_len)  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_VERSION](#) and [G\\_UTIL\\_GCAPS\\_VERSION](#) to provide the library and [gcaps](#) version numbers.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GInfo** (GCon g, GCStringOut info, GSize info\_len)  
*Uses [GUtility\(\)](#) and [G\\_UTIL\\_INFO](#) to provide a useful connection string.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GAddresses** (GCStringOut addresses, GSize addresses\_len)  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_ADDRESSES](#) or [G\\_UTIL\\_ADDRESSES](#) to provide a listing of all available connection addresses.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GTimeout** (GCon g, short timeout\_ms)  
*Uses [GUtility\(\)](#) and [G\\_UTIL\\_TIMEOUT\\_OVERRIDE](#) to set the library timeout.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GAssign** (GCStringIn ip, GCStringIn mac)  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_ASSIGN](#) or [G\\_UTIL\\_ASSIGN](#) to assign an IP address over the Ethernet to a controller at a given MAC address.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GIpRequests** (GCStringOut requests, GSize requests\_len)  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_IPREQUEST](#) or [G\\_UTIL\\_IPREQUEST](#) to provide a list of all [Gallil](#) controllers requesting IP addresses via BOOT-P or DHCP.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GSetServer** (GCStringIn server\_name)  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_SET\\_SERVER](#) to set the new active server.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GServerStatus** (GCStringOut status, GSize status\_len)  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_SERVER\\_STATUS](#) to get information on the local server name and if it is published to the local network.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GListServers** (GCStringOut servers, GSize servers\_len)  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_LIST\\_SERVERS](#) to provide a list of all available [gcaps](#) services on the local network.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GPublishServer** (GCStringIn name, GOption publish, GOption save)  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_PUBLISH\\_SERVER](#) to publish local [gcaps](#) server to the local network.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GRemoteConnections** (GCStringOut connections, GSize connections\_length)  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_REMOTE\\_CONNECTIONS](#) to get a list of remote addresses connected to the local server.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GCmd** (GCon g, GCStringIn command)  
*Wrapper around [GCommand](#) for use when the return value is not desired.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GCmdT** (GCon g, GCStringIn command, GCStringOut trimmed←\_response, GSize response\_len, GCStringOut \*front)  
*Wrapper around [GCommand](#) that trims the response.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GCmdI** (GCon g, GCStringIn command, int \*value)  
*Wrapper around [GCommand](#) that provides the return value of a command parsed into an int.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GCmdD** (GCon g, GCStringIn command, double \*value)  
*Wrapper around [GCommand](#) that provides the return value of a command parsed into a double.*

- GCLIB\_DLL\_EXPORTED [GReturn GCALL GMotionComplete](#) ([GCon g](#), [GCStringIn axes](#))  
*Blocking call that returns once all axes specified have completed their motion.*
- GCLIB\_DLL\_EXPORTED [GReturn GCALL GWaitForBool](#) ([GCon g](#), [GCStringIn predicate](#), [int trials](#))  
*Blocking call that returns when the controller evaluates the predicate as true.*
- GCLIB\_DLL\_EXPORTED [GReturn GCALL GRecordRate](#) ([GCon g](#), [double period\\_ms](#))  
*Sets the asynchronous data record to a user-specified period via DR.*
- GCLIB\_DLL\_EXPORTED [GReturn GCALL GProgramDownloadFile](#) ([GCon g](#), [GCStringIn file\\_path](#), [GCStringIn preprocessor](#))  
*Program download from file.*
- GCLIB\_DLL\_EXPORTED [GReturn GCALL GProgramUploadFile](#) ([GCon g](#), [GCStringIn file\\_path](#))  
*Program upload to file.*
- GCLIB\_DLL\_EXPORTED [void GCALL GError](#) ([GReturn rc](#), [GCStringOut error](#), [GSize error\\_len](#))  
*Provides a human-readable description string for return codes.*

### 13.33.1 Detailed Description

Partial implementation of [gclibo.h](#)

### 13.33.2 Function Documentation

#### 13.33.2.1 GAddresses()

```
GReturn GCALL GAddresses (
    GCStringOut addresses,
    GSize addresses_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_ADDRESSES](#) or [G\\_UTIL\\_ADDRESSES](#) to provide a listing of all available connection addresses.

#### Note

Serial ports are listed, e.g. COM1. Upon open, it may be necessary to specify a baud rate for the controller, e.g. `--baud 19200`. Default baud is 115200. See [GOpen\(\)](#).

#### Parameters

<i>addresses</i>	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so. See below for more information.
<i>addresses_len</i>	Length of buffer.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

If [gcaps](#) is available, the listing will come from the server via [G\\_UTIL\\_GCAPS\\_ADDRESSES](#). In the absence of the server, gclib will use [G\\_UTIL\\_ADDRESSES](#) to generate the list.

- Ethernet controllers will be listed as *ip\_address, revision\_report, network\_adapter\_name, network\_adapter←\_ip\_address*. If an IP address is unreachable via ping, the address will be in parentheses.
- PCI controllers will be listed by their identifier, e.g. GALILPCI1.
- Serial ports will be listed by their identifier, e.g. COM1.

```
10.1.3.91, DMC4020 Rev 1.2e, LAN, 10.1.3.10
192.168.0.63, DMC4040 Rev 1.2f, Static, 192.168.0.41
(192.0.0.42), RIO47102 Rev 1.1j, Static, 192.168.0.41
GALILPCI1
```

COM1  
COM2

#### Note

[GAddresses\(\)](#) will take up to 1 second to look for [gcaps](#).

See [x\\_examples.cpp](#) for an example.

Definition at line 54 of file [gclibo.c](#).

References [G\\_NO\\_ERROR](#), [G\\_UTIL\\_ADDRESSES](#), [G\\_UTIL\\_GCAPS\\_ADDRESSES](#), and [GUtility\(\)](#).

### 13.33.2.2 GAssign()

```
GReturn GCALL GAssign (
    GCStringIn ip,
    GCStringIn mac )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_ASSIGN](#) or [G\\_UTIL\\_ASSIGN](#) to assign an IP address over the Ethernet to a controller at a given MAC address.

#### Parameters

<i>ip</i>	The null-terminated ip address to assign. The hardware should not yet have an IP address.
<i>mac</i>	The null-terminated MAC address of the hardware.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

On Linux and Mac, the desired IP address will be pinged prior to the assignment. If the ping is returned, [GAssign\(\)](#) will return [G\\_GCLIB\\_UTILITY\\_IP\\_TAKEN](#).

If [gcaps](#) is available, the assign will be performed from the server via [G\\_UTIL\\_GCAPS\\_ASSIGN](#). [gcaps](#) will remember the assignment and will automatically assign the desired IP address should the controller ever request one again, e.g. after a controller master reset. To clear the remembered IP address from [gcaps](#), call [GAssign\(\)](#) with a blank string in place of the ip address. To remove all remembered ip addresses, specify a blank string for the mac address.

In the absence of the server, [gclib](#) will use [G\\_UTIL\\_ASSIGN](#) to assign. [GAssign\(\)](#) will take up to 1 second to look for [gcaps](#). When not using [gcaps](#), Linux/OS X users must be root to use [GAssign\(\)](#) and have UDP access to send on port 68.

See [x\\_examples.cpp](#) for an example.

Definition at line 70 of file [gclibo.c](#).

References [G\\_GCLIB\\_UTILITY\\_IP\\_TAKEN](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_ASSIGN](#), [G\\_UTIL\\_GCAPS\\_ASSIGN](#), [G\\_UTIL\\_GCAPS\\_PING](#), [G\\_UTIL\\_PING](#), and [GUtility\(\)](#).

### 13.33.2.3 GCmd()

```
GReturn GCALL GCmd (
    GCon g,
    GCStringIn command )
```

Wrapper around [GCommand](#) for use when the return value is not desired.

The returned data is still checked for error, e.g. ? or timeout, but is not brought out through the prototype.

#### Parameters

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 237 of file `gclibo.c`.

**13.33.2.4 GCmdD()**

```
GReturn GCALL GCmdD (
    GCon g,
    GCStringIn command,
    double * value )
```

Wrapper around GCommand that provides the return value of a command parsed into a double.

Use this function to retrieve the full Galil 4.2 range, e.g. for a variable value with fractional data, or the value of an Analog input or Output.

**Parameters**

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.
<i>value</i>	Pointer to a double that will be filled with the return value.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 289 of file `gclibo.c`.

**13.33.2.5 GCmdI()**

```
GReturn GCALL GCmdI (
    GCon g,
    GCStringIn command,
    int * value )
```

Wrapper around GCommand that provides the return value of a command parsed into an int.

Use this function to get most values including TP, RP, TE, Digital I/O states, etc.

**Parameters**

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.
<i>value</i>	Pointer to an int that will be filled with the return value.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 278 of file `gclibo.c`.

**13.33.2.6 GCmdT()**

```
GReturn GCALL GCmdT (
    GCon g,
```



```

GCStringIn  command,
GCStringOut trimmed_response,
GSize      response_len,
GCStringOut * front )

```

Wrapper around GCommand that trims the response.

For use when the return value is desired, is ASCII (not binary), and the response should be trimmed of trailing colon, whitespace, and optionally leading space.

#### Parameters

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.
<i>trimmed_response</i>	The trimmed response from the controller. Trailing space is trimmed by null terminating any trailing spaces, carriage returns, or line feeds.
<i>response_len</i>	The length of the trimmed_response buffer.
<i>front</i>	If non-null, upon return *front will point to the first non-space character in trimmed_response. This allows trimming the front of the string without modifying the user's buffer pointer, which may be allocated on the heap.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 243 of file gclibo.c.

#### 13.33.2.7 GError()

```

void GCALL GError (
    GReturn rc,
    GCStringOut error,
    GSize error_len )

```

Provides a human-readable description string for return codes.

#### Parameters

<i>rc</i>	The return code to lookup.
<i>error</i>	The buffer to fill with the error text. Buffer will be null terminated, even if the data must be truncated to do so.
<i>error_len</i>	The length of the error buffer.

See [x\\_examples.cpp](#) for an example.

Definition at line 459 of file gclibo.c.

References G\_NO\_ERROR.

#### 13.33.2.8 GInfo()

```

GReturn GCALL GInfo (
    GCon g,
    GCStringOut info,
    GSize info_len )

```

Uses [GUtility\(\)](#) and [G\\_UTIL\\_INFO](#) to provide a useful connection string.

#### Parameters

<i>g</i>	Connection's handle.
----------	----------------------

## Parameters

<i>info</i>	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so.
<i>info_len</i>	Length of buffer.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

The response is *address, revision\_report, serial\_number*. For example:

```
COM2, RIO47102 Rev 1.1j, 37290
```

See [x\\_examples.cpp](#) for an example.

Definition at line 49 of file `gclibo.c`.

### 13.33.2.9 GIpRequests()

```
GReturn GCALL GIpRequests (
    GCStringOut requests,
    GSize requests_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_IPREQUEST](#) or [G\\_UTIL\\_IPREQUEST](#) to provide a list of all [Galil](#) controllers requesting IP addresses via BOOT-P or DHCP.

## Parameters

<i>requests</i>	The buffer to hold the list of requesting controllers. Data will be null terminated, even if the data must be truncated to do so. See below for more information.
<i>requests_len</i>	The length of the requests buffer.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

[GIpRequests\(\)](#) will block up to 5 seconds while listening for requests.

If [gcaps](#) is available, the listing will come from the server via [G\\_UTIL\\_GCAPS\\_IPREQUEST](#). In the absence of the server, `gclib` will use [G\\_UTIL\\_IPREQUEST](#) to generate the list. [GIpRequests\(\)](#) will take up to 1 second to look for [gcaps](#). When not using [gcaps](#), Linux/OS X users must be root to use [GIpRequests\(\)](#) and have UDP access to bind and listen on port 67.

Each line of the returned data will be of the form *model, serial\_number, MAC\_address, network\_adapter\_name, network\_adapter\_ip\_address, remembered\_ip\_assignment*. See [GAssign\(\)](#) for more information about remembered IP assignments. The following is an example output.

```
DMC2000, 34023, 00:50:4C:00:84:E7, enp5s0, 192.168.42.92, 192.168.42.200
DMC2105, 7, 00:50:4C:58:00:07, enp5s0, 192.168.42.92, 0.0.0.0
DMC2105, 13, 00:50:4C:58:00:0D, enp5s0, 192.168.42.92, 0.0.0.0
```

See [x\\_examples.cpp](#) for an example.

Definition at line 106 of file `gclibo.c`.

References [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_IPREQUEST](#), [G\\_UTIL\\_IPREQUEST](#), [GSleep\(\)](#), and [GUtility\(\)](#).

Referenced by [ip\\_assigner\(\)](#).

### 13.33.2.10 GListServers()

```
GReturn GCALL GListServers (
    GCStringOut servers,
    GSize servers_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_LIST\\_SERVERS](#) to provide a list of all available [gcaps](#) services on the local network.

**Note**

This function is only available on Windows 10 and Linux.

**Parameters**

<i>servers</i>	The buffer to hold the list of available gcaps servers
<i>servers_len</i>	The length of the servers buffer

This function is used to find a list of available gcaps servers that have made themselves "Discoverable". The list of available servers are separated by a newline '\n' character.

**Attention**

This function will always use your local gcaps server, regardless of which server you have set as your active server.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 169 of file gclibo.c.

References G\_GCAPS\_OPEN\_ERROR, G\_NO\_ERROR, G\_UTIL\_GCAPS\_LIST\_SERVERS, and GUtility().

**13.33.2.11 GMotionComplete()**

```
GReturn GCALL GMotionComplete (
    GCon g,
    GCStringIn axes )
```

Blocking call that returns once all axes specified have completed their motion.

**Note**

This function uses a profiled motion indicator, not the position of the encoder. E.G. see the difference between AM (profiled) and MC (encoder-based).

Although using the `_BGm` operand is the most generally compatible method, there are higher-performance ways to check for motion complete by using the data record, or interrupts. See examples [x\\_dr\\_motioncomplete\(\)](#) and [x\\_ei\\_motioncomplete\(\)](#).

**Parameters**

<i>g</i>	Connection's handle.
<i>axes</i>	A null-terminated string containing a multiple-axes mask. Every character in the string should be a valid argument to <code>MG_BGm</code> , i.e. XYZWABCDEFHST.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gmotioncomplete.cpp](#) for an example.

Definition at line 300 of file gclibo.c.

**13.33.2.12 GProgramDownloadFile()**

```
GReturn GCALL GProgramDownloadFile (
    GCon g,
    GCStringIn file_path,
    GCStringIn preprocessor )
```

Program download from file.

#### Parameters

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the program file.
<i>preprocessor</i>	Options string for preprocessing the program before sending it to the controller. See <a href="#">GProgramDownload()</a> .

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_programs.cpp](#) for an example.

Definition at line 387 of file `gclibo.c`.

#### 13.33.2.13 GProgramUploadFile()

```
GReturn GCALL GProgramUploadFile (
    GCon g,
    GCStringIn file_path )
```

Program upload to file.

#### Parameters

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the program file, file will be overwritten if it exists.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_programs.cpp](#) for an example.

Definition at line 430 of file `gclibo.c`.

#### 13.33.2.14 GPublishServer()

```
GReturn GCALL GPublishServer (
    GCStringIn name,
    GOption publish,
    GOption save )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_PUBLISH\\_SERVER](#) to publish local gcaps server to the local network.

#### Note

This function is only available on Windows 10 and Linux.

#### Parameters

<i>name</i>	The name of the server to publish or remove
<i>publish</i>	Option to publish or remove server from network
<i>save</i>	Option to save this configuration for future reboots

This function is used to make your local gcaps server "Discoverable" or "Invisible"  
publish Option:

Set to 1 to publish server to the network and make "Discoverable"  
 Set to 0 to remove server from the network and make "Invisible"

save Option:

Set to 1 to save the configuration for future reboots of the server  
 Set to 0 to use this configuration once, and not overwrite previous server settings

#### Attention

This function will always use your local gcaps server, regardless of which server you have set as your active server.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 189 of file gclibo.c.  
 Referenced by `remote_server()`.

#### 13.33.2.15 GRecordRate()

```
GReturn GCALL GRecordRate (
    GCon g,
    double period_ms )
```

Sets the asynchronous data record to a user-specified period via DR.  
 Takes TM and product type into account and sets the DR period to the period requested by the user, if possible.

#### Parameters

<i>g</i>	Connection's handle.
<i>period_ms</i>	Period, in milliseconds, to set up for the asynchronous data record.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_grecord.cpp](#) for an example.  
 Definition at line 342 of file gclibo.c.

#### 13.33.2.16 GRemoteConnections()

```
GReturn GCALL GRemoteConnections (
    GCStringOut connections,
    GSize connections_length )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_REMOTE\\_CONNECTIONS](#) to get a list of remote addresses connected to the local server.

#### Note

This function is only available on Windows 10 and Linux.

#### Parameters

<i>connections</i>	The buffer to hold the list of remote IP addresses currently connected to your hardware
<i>connections_len</i>	The length of the connections buffer

This function is used to find a list of IP Addresses of machines that currently have open connections to your local

hardware. If another user sets your local server as their active server, and then opens a connection to your hardware, their IP Address will appear in this list.

The list of IP addresses are separated by a newline '\n' character.

#### Attention

This function will always use your local gcaps server, regardless of which server you have set as your active server.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 217 of file gclibo.c.

References [G\\_GCAPS\\_OPEN\\_ERROR](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_REMOTE\\_CONNECTIONS](#), and [GUtility\(\)](#).

### 13.33.2.17 GServerStatus()

```
GReturn GCALL GServerStatus (
    GCStringOut status,
    GSize status_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_SERVER\\_STATUS](#) to get information on the local server name and if it is published to the local network.

#### Note

This function is only available on Windows 10 and Linux.

#### Parameters

<i>status</i>	The buffer to hold the status of the local gcaps server
<i>status_len</i>	The length of the status buffer

This function is used to find the status of your local gcaps server. Use this function to determine the name your server is currently using, and whether or not your gcaps server is currently set to "Discoverable" or "Invisible"

The status buffer will be filled in the form of "[Server Name], [Discoverable]"

For example, for a server with the name "Example Server" that is set to "Discoverable", the status buffer would contain "Example Server, true".

#### Attention

This function will always use your local gcaps server, regardless of which server you have set as your active server.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 149 of file gclibo.c.

References [G\\_GCAPS\\_OPEN\\_ERROR](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_SERVER\\_STATUS](#), and [GUtility\(\)](#).

### 13.33.2.18 GSetServer()

```
GReturn GCALL GSetServer (
    GCStringIn server_name )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_SET\\_SERVER](#) to set the new active server.

**Note**

This function is only available on Windows 10 and Linux.

**Parameters**

<i>server_name</i>	The name of the server to set as your new active server.
--------------------	--

Use this function in conjunction with [GListServers\(\)](#). Choose a name received from [GListServers\(\)](#) to set as your new active server.

After setting a new active server, all gclib calls will route through that new active server, unless explicitly noted otherwise.

To set your active server back to your local server, simply pass "Local" to [GSetServer\(\)](#):

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 128 of file gclibo.c.

References [G\\_GCAPS\\_OPEN\\_ERROR](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_SET\\_SERVER](#), and [GUtility\(\)](#).

**13.33.2.19 GSleep()**

```
void GCALL GSleep (
    unsigned int timeout_ms )
```

Uses [GUtility\(\)](#) and [G\\_UTIL\\_SLEEP](#) to provide a blocking sleep call which can be useful for timing-based chores.

**Parameters**

<i>timeout_ms</i>	The timeout, in milliseconds, to block before returning.
-------------------	--

See [GWaitForBool\(\)](#) for an example.

Definition at line 24 of file gclibo.c.

References [G\\_UTIL\\_SLEEP](#), and [GUtility\(\)](#).

Referenced by [GlpRequests\(\)](#).

**13.33.2.20 GTimeout()**

```
GReturn GCALL GTimeout (
    GCon g,
    short timeout_ms )
```

Uses [GUtility\(\)](#) and [G\\_UTIL\\_TIMEOUT\\_OVERRIDE](#) to set the library timeout.

**Parameters**

<i>g</i>	Connection's handle.
<i>timeout_ms</i>	The value to be used for the timeout. Use <a href="#">G_USE_INITIAL_TIMEOUT</a> to set the timeout back to the initial <a href="#">GOpen()</a> value, <code>--timeout</code> .

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) and [x\\_gread\\_gwrite.cpp](#) for examples.

Definition at line 65 of file gclibo.c.

### 13.33.2.21 GVersion()

```
GReturn GCALL GVersion (
    GCStringOut ver,
    GSize ver_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_VERSION](#) and [G\\_UTIL\\_GCAPS\\_VERSION](#) to provide the library and [gcaps](#) version numbers.

#### Parameters

<i>ver</i>	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so.
<i>ver_len</i>	Length of buffer.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

The version number of gclib is provided first. If the [gcaps](#) server can be found, its version will be provided after a space.

Example with gcaps version.

```
154.190.329 1.0.0.82
```

Example with gclib version only.

```
154.190.329
```

#### Note

[GVersion\(\)](#) will take up to 1 second to look for [gcaps](#).

See [x\\_examples.cpp](#) for an example.

Definition at line 29 of file [gclibo.c](#).

References [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_VERSION](#), [G\\_UTIL\\_VERSION](#), and [GUtility\(\)](#).

### 13.33.2.22 GWaitForBool()

```
GReturn GCALL GWaitForBool (
    GCon g,
    GCStringIn predicate,
    int trials )
```

Blocking call that returns when the controller evaluates the predicate as true.

Polls the message command (MG) to check the value of predicate. Polling will continue until the controller responds with a nonzero value or the number of polling trials is reached.

The amount of time until the function fails with [G\\_GCLIB\\_POLLING\\_FAILED](#) is roughly (trials \* [POLLINGINTERVAL](#)) milliseconds.

#### Parameters

<i>g</i>	Connection's handle.
<i>predicate</i>	A null-terminated string containing the predicate to be polled. The predicate will be enclosed in parentheses and used in the command <code>MG (predicate)</code> to return the value.
<i>trials</i>	The number of polling cycles to perform looking for a nonzero value. Use -1 to poll indefinitely.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [GMotionComplete\(\)](#) for an example.

Definition at line 318 of file [gclibo.c](#).



## 13.34 gclibo.h File Reference

```
#include "gclib.h"
```

### Macros

- #define **GCLIB\_DLL\_EXPORTED**
- #define **GCALL \_\_stdcall**
- #define **MALLOCBUF G\_HUGE\_BUFFER**  
*Malloc used for large program and array uploads.*
- #define **MAXPROG MALLOCBUF**  
*Maximum size for a program.*
- #define **MAXARRAY MALLOCBUF**  
*Maximum size for an array table upload.*
- #define **POLLINGINTERVAL 100**  
*Interval, in milliseconds, for polling commands, e.g. `GWaitForBool()`.*
- #define **G\_USE\_GCAPS**  
*Use the GCAPS server in `GAddresses()`, `GAssign()`, `GlpRequests()`, and `GVersion()`. To avoid GCAPS, comment out this line and recompile, <http://galil.com/sw/pub/all/doc/gclib/html/gclibo.html>.*

### Functions

- GCLIB\_DLL\_EXPORTED void **GCALL GSleep** (unsigned int timeout\_ms)  
*Uses `GUtility()` and `G_UTIL_SLEEP` to provide a blocking sleep call which can be useful for timing-based chores.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GVersion** (GCStringOut ver, GSize ver\_len)  
*Uses `GUtility()`, `G_UTIL_VERSION` and `G_UTIL_GCAPS_VERSION` to provide the library and `gcaps` version numbers.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GAddresses** (GCStringOut addresses, GSize addresses\_len)  
*Uses `GUtility()`, `G_UTIL_GCAPS_ADDRESSES` or `G_UTIL_ADDRESSES` to provide a listing of all available connection addresses.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GInfo** (GCon g, GCStringOut info, GSize info\_len)  
*Uses `GUtility()` and `G_UTIL_INFO` to provide a useful connection string.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GTimeout** (GCon g, short timeout\_ms)  
*Uses `GUtility()` and `G_UTIL_TIMEOUT_OVERRIDE` to set the library timeout.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GCmd** (GCon g, GCStringIn command)  
*Wrapper around `GCommand` for use when the return value is not desired.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GCmdT** (GCon g, GCStringIn command, GCStringOut trimmed\_response, GSize response\_len, GCStringOut \*front)  
*Wrapper around `GCommand` that trims the response.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GCmdI** (GCon g, GCStringIn command, int \*value)  
*Wrapper around `GCommand` that provides the return value of a command parsed into an int.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GCmdD** (GCon g, GCStringIn command, double \*value)  
*Wrapper around `GCommand` that provides the return value of a command parsed into a double.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GWaitForBool** (GCon g, GCStringIn predicate, int trials)  
*Blocking call that returns when the controller evaluates the predicate as true.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GMotionComplete** (GCon g, GCStringIn axes)  
*Blocking call that returns once all axes specified have completed their motion.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GRecordRate** (GCon g, double period\_ms)  
*Sets the asynchronous data record to a user-specified period via `DR`.*
- GCLIB\_DLL\_EXPORTED **GReturn GCALL GProgramDownloadFile** (GCon g, GCStringIn file\_path, GCStringIn preprocessor)  
*Program download from file.*

- `GCLIB_DLL_EXPORTED GReturn GCALL GProgramUploadFile (GCon g, GCStringIn file_path)`  
*Program upload to file.*
- `GCLIB_DLL_EXPORTED GReturn GCALL GArrayDownloadFile (GCon g, GCStringIn file_path)`  
*Array download from file.*
- `GCLIB_DLL_EXPORTED GReturn GCALL GArrayUploadFile (GCon g, GCStringIn file_path, GCStringIn names)`  
*Array upload to file.*
- `GCLIB_DLL_EXPORTED GReturn GCALL GIpRequests (GCStringOut requests, GSize requests_len)`  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_IPREQUEST](#) or [G\\_UTIL\\_IPREQUEST](#) to provide a list of all *Galil* controllers requesting IP addresses via BOOT-P or DHCP.*
- `GCLIB_DLL_EXPORTED GReturn GCALL GSetServer (GCStringIn server_name)`  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_SET\\_SERVER](#) to set the new active server.*
- `GCLIB_DLL_EXPORTED GReturn GCALL GListServers (GCStringOut servers, GSize servers_len)`  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_LIST\\_SERVERS](#) to provide a list of all available *gcaps* services on the local network.*
- `GCLIB_DLL_EXPORTED GReturn GCALL GPublishServer (GCStringIn name, GOption publish, GOption save)`  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_PUBLISH\\_SERVER](#) to publish local *gcaps* server to the local network.*
- `GCLIB_DLL_EXPORTED GReturn GCALL GServerStatus (GCStringOut status, GSize status_len)`  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_SERVER\\_STATUS](#) to get information on the local server name and if it is published to the local network.*
- `GCLIB_DLL_EXPORTED GReturn GCALL GRemoteConnections (GCStringOut connections, GSize connections_length)`  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_REMOTE\\_CONNECTIONS](#) to get a list of remote addresses connected to the local server.*
- `GCLIB_DLL_EXPORTED GReturn GCALL GAssign (GCStringIn ip, GCStringIn mac)`  
*Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_ASSIGN](#) or [G\\_UTIL\\_ASSIGN](#) to assign an IP address over the Ethernet to a controller at a given MAC address.*
- `GCLIB_DLL_EXPORTED void GCALL GError (GReturn rc, GCStringOut error, GSize error_len)`  
*Provides a human-readable description string for return codes.*
- `GCLIB_DLL_EXPORTED GReturn GCALL GSetupDownloadFile (GCon g, GCStringIn file_path, GOption options, GCStringOut info, GSize info_len)`  
*Download a saved controller configuration from a file.*

### 13.34.1 Detailed Description

Open-source convenience functions for [Galil C Lib](#). Please email [softwarefeedback@galil.com](mailto:softwarefeedback@galil.com) with suggestions for useful/missing functions.

### 13.34.2 Function Documentation

#### 13.34.2.1 GAddresses()

```
GReturn GCALL GAddresses (
    GCStringOut addresses,
    GSize addresses_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_ADDRESSES](#) or [G\\_UTIL\\_ADDRESSES](#) to provide a listing of all available connection addresses.

#### Note

Serial ports are listed, e.g. COM1. Upon open, it may be necessary to specify a baud rate for the controller, e.g. `--baud 19200`. Default baud is 115200. See [GOpen\(\)](#).

## Parameters

<i>addresses</i>	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so. See below for more information.
<i>addresses_len</i>	Length of buffer.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

If [gcaps](#) is available, the listing will come from the server via [G\\_UTIL\\_GCAPS\\_ADDRESSES](#). In the absence of the server, gclib will use [G\\_UTIL\\_ADDRESSES](#) to generate the list.

- Ethernet controllers will be listed as *ip\_address, revision\_report, network\_adapter\_name, network\_adapter→\_ip\_address*. If an IP address is unreachable via ping, the address will be in parentheses.
- PCI controllers will be listed by their identifier, e.g. GALILPCI1.
- Serial ports will be listed by their identifier, e.g. COM1.

```
10.1.3.91, DMC4020 Rev 1.2e, LAN, 10.1.3.10
192.168.0.63, DMC4040 Rev 1.2f, Static, 192.168.0.41
(192.0.0.42), RIO47102 Rev 1.1j, Static, 192.168.0.41
GALILPCI1
COM1
COM2
```

## Note

[GAddresses\(\)](#) will take up to 1 second to look for [gcaps](#).

See [x\\_examples.cpp](#) for an example.

Definition at line 54 of file [gclibo.c](#).

References [G\\_NO\\_ERROR](#), [G\\_UTIL\\_ADDRESSES](#), [G\\_UTIL\\_GCAPS\\_ADDRESSES](#), and [GUtility\(\)](#).

## 13.34.2.2 GArrayDownloadFile()

```
GCLIB_DLL_EXPORTED GReturn GCALL GArrayDownloadFile (
    GCon g,
    GCStringIn file_path )
```

Array download from file.

Downloads a csv file containing array data at *file\_path*. If the arrays don't exist, they will be dimensioned.

## Parameters

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the array file.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_arrays.cpp](#) for an example.

Definition at line 380 of file [arrays.c](#).

## 13.34.2.3 GArrayUploadFile()

```
GCLIB_DLL_EXPORTED GReturn GCALL GArrayUploadFile (
    GCon g,
```

```

GCStringIn file_path,
GCStringIn names )

```

Array upload to file.

Uploads the entire controller array table or a subset and saves the data as a csv file specified by `file_path`.

#### Parameters

<code>g</code>	Connection's handle.
<code>file_path</code>	Null-terminated string containing the path to the array file, file will be overwritten if it exists.
<code>names</code>	Null-terminated string containing the arrays to upload, delimited with space. "" or null uploads all arrays listed in LA.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_arrays.cpp](#) for an example.

Definition at line 408 of file `arrays.c`.

### 13.34.2.4 GAssign()

```

GReturn GCALL GAssign (
    GCStringIn ip,
    GCStringIn mac )

```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_ASSIGN](#) or [G\\_UTIL\\_ASSIGN](#) to assign an IP address over the Ethernet to a controller at a given MAC address.

#### Parameters

<code>ip</code>	The null-terminated ip address to assign. The hardware should not yet have an IP address.
<code>mac</code>	The null-terminated MAC address of the hardware.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

On Linux and Mac, the desired IP address will be pinged prior to the assignment. If the ping is returned, [GAssign\(\)](#) will return [G\\_GCLIB\\_UTILITY\\_IP\\_TAKEN](#).

If [gcaps](#) is available, the assign will be performed from the server via [G\\_UTIL\\_GCAPS\\_ASSIGN](#). [gcaps](#) will remember the assignment and will automatically assign the desired IP address should the controller ever request one again, e.g. after a controller master reset. To clear the remembered IP address from [gcaps](#), call [GAssign\(\)](#) with a blank string in place of the ip address. To remove all remembered ip addresses, specify a blank string for the mac address.

In the absence of the server, [gclib](#) will use [G\\_UTIL\\_ASSIGN](#) to assign. [GAssign\(\)](#) will take up to 1 second to look for [gcaps](#). When not using [gcaps](#), Linux/OS X users must be root to use [GAssign\(\)](#) and have UDP access to send on port 68.

See [x\\_examples.cpp](#) for an example.

Definition at line 70 of file `gclibo.c`.

References [G\\_GCLIB\\_UTILITY\\_IP\\_TAKEN](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_ASSIGN](#), [G\\_UTIL\\_GCAPS\\_ASSIGN](#), [G\\_UTIL\\_ASSIGN](#), [G\\_UTIL\\_GCAPS\\_PING](#), [G\\_UTIL\\_PING](#), and [GUtility\(\)](#).

### 13.34.2.5 GCmd()

```

GReturn GCALL GCmd (
    GCon g,
    GCStringIn command )

```

Wrapper around GCommand for use when the return value is not desired. The returned data is still checked for error, e.g. ? or timeout, but is not brought out through the prototype.

#### Parameters

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 237 of file `gclibo.c`.

### 13.34.2.6 GCmdD()

```
GReturn GCALL GCmdD (
    GCon g,
    GCStringIn command,
    double * value )
```

Wrapper around GCommand that provides the return value of a command parsed into a double. Use this function to retrieve the full [Galil 4.2](#) range, e.g. for a variable value with fractional data, or the value of an Analog input or Output.

#### Parameters

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.
<i>value</i>	Pointer to a double that will be filled with the return value.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 289 of file `gclibo.c`.

### 13.34.2.7 GCmdI()

```
GReturn GCALL GCmdI (
    GCon g,
    GCStringIn command,
    int * value )
```

Wrapper around GCommand that provides the return value of a command parsed into an int. Use this function to get most values including TP, RP, TE, Digital I/O states, etc.

#### Parameters

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.
<i>value</i>	Pointer to an int that will be filled with the return value.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 278 of file `gclibo.c`.

**13.34.2.8 GCmdT()**

```
GReturn GCALL GCmdT (
    GCon g,
    GCStringIn command,
    GCStringOut trimmed_response,
    GSize response_len,
    GCStringOut * front )
```

Wrapper around GCommand that trims the response.

For use when the return value is desired, is ASCII (not binary), and the response should be trimmed of trailing colon, whitespace, and optionally leading space.

**Parameters**

<i>g</i>	Connection's handle.
<i>command</i>	Null-terminated command string to send to the controller.
<i>trimmed_response</i>	The trimmed response from the controller. Trailing space is trimmed by null terminating any trailing spaces, carriage returns, or line feeds.
<i>response_len</i>	The length of the trimmed_response buffer.
<i>front</i>	If non-null, upon return *front will point to the first non-space character in trimmed_response. This allows trimming the front of the string without modifying the user's buffer pointer, which may be allocated on the heap.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) for an example.

Definition at line 243 of file `gclibo.c`.

**13.34.2.9 GError()**

```
void GCALL GError (
    GReturn rc,
    GCStringOut error,
    GSize error_len )
```

Provides a human-readable description string for return codes.

**Parameters**

<i>rc</i>	The return code to lookup.
<i>error</i>	The buffer to fill with the error text. Buffer will be null terminated, even if the data must be truncated to do so.
<i>error_len</i>	The length of the error buffer.

See [x\\_examples.cpp](#) for an example.

Definition at line 459 of file `gclibo.c`.

References `G_NO_ERROR`.

### 13.34.2.10 GInfo()

```
GReturn GCALL GInfo (
    GCon g,
    GCStringOut info,
    GSize info_len )
```

Uses [GUtility\(\)](#) and [G\\_UTIL\\_INFO](#) to provide a useful connection string.

#### Parameters

<i>g</i>	Connection's handle.
<i>info</i>	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so.
<i>info_len</i>	Length of buffer.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

The response is *address*, *revision\_report*, *serial\_number*. For example:

```
COM2, RIO47102 Rev 1.1j, 37290
```

See [x\\_examples.cpp](#) for an example.

Definition at line 49 of file `gclibo.c`.

### 13.34.2.11 GIpRequests()

```
GReturn GCALL GIpRequests (
    GCStringOut requests,
    GSize requests_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_IPREQUEST](#) or [G\\_UTIL\\_IPREQUEST](#) to provide a list of all [Galil](#) controllers requesting IP addresses via BOOT-P or DHCP.

#### Parameters

<i>requests</i>	The buffer to hold the list of requesting controllers. Data will be null terminated, even if the data must be truncated to do so. See below for more information.
<i>requests_len</i>	The length of the requests buffer.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

[GIpRequests\(\)](#) will block up to 5 seconds while listening for requests.

If [gcaps](#) is available, the listing will come from the server via [G\\_UTIL\\_GCAPS\\_IPREQUEST](#). In the absence of the server, `gclib` will use [G\\_UTIL\\_IPREQUEST](#) to generate the list. [GIpRequests\(\)](#) will take up to 1 second to look for [gcaps](#). When not using [gcaps](#), Linux/OS X users must be root to use [GIpRequests\(\)](#) and have UDP access to bind and listen on port 67.

Each line of the returned data will be of the form *model*, *serial\_number*, *MAC\_address*, *network\_adapter\_name*, *network\_adapter\_ip\_address*, *remembered\_ip\_assignment*. See [GAssign\(\)](#) for more information about remembered IP assignments. The following is an example output.

```
DMC2000, 34023, 00:50:4C:00:84:E7, enp5s0, 192.168.42.92, 192.168.42.200
DMC2105, 7, 00:50:4C:58:00:07, enp5s0, 192.168.42.92, 0.0.0.0
DMC2105, 13, 00:50:4C:58:00:0D, enp5s0, 192.168.42.92, 0.0.0.0
```

See [x\\_examples.cpp](#) for an example.

Definition at line 106 of file `gclibo.c`.

References [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_IPREQUEST](#), [G\\_UTIL\\_IPREQUEST](#), [GSleep\(\)](#), and [GUtility\(\)](#).

Referenced by [ip\\_assigner\(\)](#).

### 13.34.2.12 GListServers()

```
GReturn GCALL GListServers (
    GCStringOut servers,
    GSize servers_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_LIST\\_SERVERS](#) to provide a list of all available gcaps services on the local network.

#### Note

This function is only available on Windows 10 and Linux.

#### Parameters

<i>servers</i>	The buffer to hold the list of available gcaps servers
<i>servers_len</i>	The length of the servers buffer

This function is used to find a list of available gcaps servers that have made themselves "Discoverable". The list of available servers are separated by a newline '\n' character.

#### Attention

This function will always use your local gcaps server, regardless of which server you have set as your active server.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 169 of file `gclibo.c`.

References [G\\_GCAPS\\_OPEN\\_ERROR](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_LIST\\_SERVERS](#), and [GUtility\(\)](#).

### 13.34.2.13 GMotionComplete()

```
GReturn GCALL GMotionComplete (
    GCon g,
    GCStringIn axes )
```

Blocking call that returns once all axes specified have completed their motion.

#### Note

This function uses a profiled motion indicator, not the position of the encoder. E.G. see the difference between AM (profiled) and MC (encoder-based).

Although using the `_BGm` operand is the most generally compatible method, there are higher-performance ways to check for motion complete by using the data record, or interrupts. See examples [x\\_dr\\_motioncomplete\(\)](#) and [x\\_ei\\_motioncomplete\(\)](#).

#### Parameters

<i>g</i>	Connection's handle.
<i>axes</i>	A null-terminated string containing a multiple-axes mask. Every character in the string should be a valid argument to <code>MG_BGm</code> , i.e. XYZWABCEFGHST.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gmotioncomplete.cpp](#) for an example.

Definition at line 300 of file `gclibo.c`.



**13.34.2.14 GProgramDownloadFile()**

```
GReturn GCALL GProgramDownloadFile (
    GCon g,
    GCStringIn file_path,
    GCStringIn preprocessor )
```

Program download from file.

**Parameters**

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the program file.
<i>preprocessor</i>	Options string for preprocessing the program before sending it to the controller. See <a href="#">GProgramDownload()</a> .

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_programs.cpp](#) for an example.

Definition at line 387 of file [gclibo.c](#).

**13.34.2.15 GProgramUploadFile()**

```
GReturn GCALL GProgramUploadFile (
    GCon g,
    GCStringIn file_path )
```

Program upload to file.

**Parameters**

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the program file, file will be overwritten if it exists.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_programs.cpp](#) for an example.

Definition at line 430 of file [gclibo.c](#).

**13.34.2.16 GPublishServer()**

```
GReturn GCALL GPublishServer (
    GCStringIn name,
    GOption publish,
    GOption save )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_PUBLISH\\_SERVER](#) to publish local gcaps server to the local network.

**Note**

This function is only available on Windows 10 and Linux.

## Parameters

<i>name</i>	The name of the server to publish or remove
<i>publish</i>	Option to publish or remove server from network
<i>save</i>	Option to save this configuration for future reboots

This function is used to make your local gcaps server "Discoverable" or "Invisible"  
 publish Option:  
 Set to 1 to publish server to the network and make "Discoverable"  
 Set to 0 to remove server from the network and make "Invisible"

save Option:  
 Set to 1 to save the configuration for future reboots of the server  
 Set to 0 to use this configuration once, and not overwrite previous server settings

## Attention

This function will always use your local gcaps server, regardless of which server you have set as your active server.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 189 of file `gclibo.c`.  
 Referenced by `remote_server()`.

**13.34.2.17 GRecordRate()**

```
GReturn GCALL GRecordRate (
    GCon g,
    double period_ms )
```

Sets the asynchronous data record to a user-specified period via DR.  
 Takes TM and product type into account and sets the DR period to the period requested by the user, if possible.

## Parameters

<i>g</i>	Connection's handle.
<i>period_ms</i>	Period, in milliseconds, to set up for the asynchronous data record.

## Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_grecord.cpp](#) for an example.  
 Definition at line 342 of file `gclibo.c`.

**13.34.2.18 GRemoteConnections()**

```
GReturn GCALL GRemoteConnections (
    GCStringOut connections,
    GSize connections_length )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_REMOTE\\_CONNECTIONS](#) to get a list of remote addresses connected to the local server.

## Note

This function is only available on Windows 10 and Linux.

**Parameters**

<i>connections</i>	The buffer to hold the list of remote IP addresses currently connected to your hardware
<i>connections_len</i>	The length of the connections buffer

This function is used to find a list of IP Addresses of machines that currently have open connections to your local hardware. If another user sets your local server as their active server, and then opens a connection to your hardware, their IP Address will appear in this list.

The list of IP addresses are separated by a newline '\n' character.

**Attention**

This function will always use your local gcaps server, regardless of which server you have set as your active server.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 217 of file gclibo.c.

References [G\\_GCAPS\\_OPEN\\_ERROR](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_REMOTE\\_CONNECTIONS](#), and [GUtility\(\)](#).

**13.34.2.19 GServerStatus()**

```
GReturn GCALL GServerStatus (
    GCStringOut status,
    GSize status_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_SERVER\\_STATUS](#) to get information on the local server name and if it is published to the local network.

**Note**

This function is only available on Windows 10 and Linux.

**Parameters**

<i>status</i>	The buffer to hold the status of the local gcaps server
<i>status_len</i>	The length of the status buffer

This function is used to find the status of your local gcaps server. Use this function to determine the name your server is currently using, and whether or not your gcaps server is currently set to "Discoverable" or "Invisible"

The status buffer will be filled in the form of "[Server Name], [Discoverable]"

For example, for a server with the name "Example Server" that is set to "Discoverable", the status buffer would contain "Example Server, true".

**Attention**

This function will always use your local gcaps server, regardless of which server you have set as your active server.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 149 of file gclibo.c.

References [G\\_GCAPS\\_OPEN\\_ERROR](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_SERVER\\_STATUS](#), and [GUtility\(\)](#).

### 13.34.2.20 GSetServer()

```
GReturn GCALL GSetServer (
    GCStringIn server_name )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_GCAPS\\_SET\\_SERVER](#) to set the new active server.

#### Note

This function is only available on Windows 10 and Linux.

#### Parameters

<i>server_name</i>	The name of the server to set as your new active server.
--------------------	--

Use this function in conjunction with [GListServers\(\)](#). Choose a name received from [GListServers\(\)](#) to set as your new active server.

After setting a new active server, all gclib calls will route through that new active server, unless explicitly noted otherwise.

To set your active server back to your local server, simply pass "Local" to [GSetServer\(\)](#):

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

Definition at line 128 of file gclibo.c.

References [G\\_GCAPS\\_OPEN\\_ERROR](#), [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_SET\\_SERVER](#), and [GUtility\(\)](#).

### 13.34.2.21 GSetupDownloadFile()

```
GCLIB_DLL_EXPORTED GReturn GCALL GSetupDownloadFile (
    GCon g,
    GCStringIn file_path,
    GOption options,
    GCStringOut info,
    GSize info_len )
```

Download a saved controller configuration from a file.

#### Parameters

<i>g</i>	Connection's handle.
<i>file_path</i>	Null-terminated string containing the path to the gcb file.
<i>options</i>	Bit mask to determine what configuration data to download. See below for all options.
<i>info</i>	Optional pointer to a buffer to store the controller info. If no info is needed, specify as NULL.
<i>info_len</i>	Length of optional info buffer. If no info is needed, specify as NULL.

#### Returns

The success status or error code of the function. If the options parameter is set to 0, the return value will be a bit mask indicating which sectors in the specified GCB are not empty. Otherwise, see [gclib\\_errors.h](#) for possible error values.

#### Note

By default, [GSetupDownloadFile\(\)](#) will stop immediately if an error is encountered downloading data. This can be overridden in the options parameter. For example, you may want to override the error if you have a backup from an 8-axis controller and want to restore the parameters for the first 4 axes to a 4-axis controller.

If both info and info\_len are not NULL, the controller information will be provided regardless of the options parameter.

The options parameter is a bit mask. If options is set to 0, [GSetupDownloadFile\(\)](#) will return a bit mask indicating which sectors in the specified GCB are not empty. The following contains a list of all currently available options:

Bit	Value	Function	Description
1	0x0002	Restore parameters	<b>KPA, KIA, KDA</b> , etc...
3	0x0008	Restore variables	Variables are listed by the <b>LV</b> command
4	0x0010	Restore arrays	Arrays are listed by the <b>LA</b> command
5	0x0020	Restore program	The program is listed by the <b>LS</b> command
31	0x8000	Ignore errors	Ignore invalid parameter errors and continue restoring data. <a href="#">GSetupDownloadFile()</a> will still stop immediately if a connection issue or other fatal error is encountered

Usage example:

```
GCon g;
GOption opt = 0;
GCStringOut info;
GSize info_len = 4096;
GReturn rc = GOpen("192.168.0.50", &g);
if (rc) return rc;
// Call GSetupDownloadFile() with options set to 0 so we can get the non-empty sector bit mask
opt = GSetupDownloadFile(g, "C:\\path\\to\\gcb\\file.gcb", 0, NULL, NULL);
info = (GCStringOut)malloc(sizeof(GCStringOut) * info_len);
// Call GSetupDownloadFile() with the bit mask returned in the previous function call
rc = GSetupDownloadFile(g, "C:\\path\\to\\gcb\\file.gcb", opt, info, info_len);
printf("Info:\n\n%s", info);
GClose(g);
free(info);
return rc;
```

Definition at line 476 of file arrays.c.

### 13.34.2.22 GSleep()

```
void GCALL GSleep (
    unsigned int timeout_ms )
```

Uses [GUtility\(\)](#) and [G\\_UTIL\\_SLEEP](#) to provide a blocking sleep call which can be useful for timing-based chores.

Parameters

<i>timeout_ms</i>	The timeout, in milliseconds, to block before returning.
-------------------	--

See [GWaitForBool\(\)](#) for an example.

Definition at line 24 of file gclibo.c.

References [G\\_UTIL\\_SLEEP](#), and [GUtility\(\)](#).

Referenced by [GlpRequests\(\)](#).

### 13.34.2.23 GTimeout()

```
GReturn GCALL GTimeout (
    GCon g,
    short timeout_ms )
```

Uses [GUtility\(\)](#) and [G\\_UTIL\\_TIMEOUT\\_OVERRIDE](#) to set the library timeout.

Parameters

<i>g</i>	Connection's handle.
<i>timeout_ms</i>	The value to be used for the timeout. Use <a href="#">G_USE_INITIAL_TIMEOUT</a> to set the timeout back to the initial <a href="#">GOpen()</a> value, <code>--timeout</code> .

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [x\\_gcommand.cpp](#) and [x\\_gread\\_gwrite.cpp](#) for examples.

Definition at line 65 of file `gclibo.c`.

**13.34.2.24 GVersion()**

```
GReturn GCALL GVersion (
    GCStringOut ver,
    GSize ver_len )
```

Uses [GUtility\(\)](#), [G\\_UTIL\\_VERSION](#) and [G\\_UTIL\\_GCAPS\\_VERSION](#) to provide the library and [gcaps](#) version numbers.

**Parameters**

<i>ver</i>	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so.
<i>ver_len</i>	Length of buffer.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

The version number of `gclib` is provided first. If the `gcaps` server can be found, its version will be provided after a space.

Example with `gcaps` version.

```
154.190.329 1.0.0.82
```

Example with `gclib` version only.

```
154.190.329
```

**Note**

[GVersion\(\)](#) will take up to 1 second to look for `gcaps`.

See [x\\_examples.cpp](#) for an example.

Definition at line 29 of file `gclibo.c`.

References [G\\_NO\\_ERROR](#), [G\\_UTIL\\_GCAPS\\_VERSION](#), [G\\_UTIL\\_VERSION](#), and [GUtility\(\)](#).

**13.34.2.25 GWaitForBool()**

```
GReturn GCALL GWaitForBool (
    GCon g,
    GCStringIn predicate,
    int trials )
```

Blocking call that returns when the controller evaluates the predicate as true.

Polls the message command (MG) to check the value of predicate. Polling will continue until the controller responds with a nonzero value or the number of polling trials is reached.

The amount of time until the function fails with [G\\_GCLIB\\_POLLING\\_FAILED](#) is roughly (trials \* [POLLINGINTERVAL](#)) milliseconds.

**Parameters**

<i>g</i>	Connection's handle.
<i>predicate</i>	A null-terminated string containing the predicate to be polled. The predicate will be enclosed in parentheses and used in the command <code>MG (predicate)</code> to return the value.
<i>trials</i>	The number of polling cycles to perform looking for a nonzero value. Use -1 to poll indefinitely.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [GMotionComplete\(\)](#) for an example.

Definition at line 318 of file gclibo.c.

## 13.35 GclibTest.java File Reference

**Data Structures**

- class [GclibTest](#)

## 13.36 ip\_assigner.cpp File Reference

```
#include "examples.h"
#include <iostream>
#include <vector>
#include <string.h>
```

**Typedefs**

- typedef [std::vector](#)< string > **tokens**

**Functions**

- tokens [string\\_split](#) (const string &str, const string &token)  
*Splits a string into a vector based on a token.*
- [GReturn ip\\_assigner](#) (char \*serial\_num, int address)  
*Assigns controller an IP Adress given a serial number and a 1 byte address.*

### 13.36.1 Detailed Description

Function calls for the IP Assigner Example Project.

### 13.36.2 Function Documentation

#### 13.36.2.1 ip\_assigner()

```
GReturn ip_assigner (
    char * serial_num,
    int address )
```

Assigns controller an IP Address given a serial number and a 1 byte address.

**Parameters**

<i>serial_num</i>	Serial Number of the controller.
<i>address</i>	A 1 byte address that defines the last byte of the IP Address.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [ip\\_assigner\\_example.cpp](#) for an example.

This function will listen on the network for controllers requesting an IP Address. If a detected controller matches the serial number provided by the user, a new IP Address will be assigned based on the first 3 bytes of the detected IP Address combined with the user defined 1 byte address.

Definition at line 26 of file `ip_assigner.cpp`.

References `e()`, `G_SMALL_BUFFER`, `GIpRequests()`, and `string_split()`.

## 13.37 ip\_assigner.cs File Reference

### Data Structures

- class [Examples](#)

*Provides a class of shared constants and methods for gclib's example projects.*

### 13.37.1 Detailed Description

Function calls for the IP Assigner Example Project.

For VB.NET, see definition in file [ip\\_assigner.vb](#)

## 13.38 IP\_Assigner.vb File Reference

### Functions

- int `IP_Assigner` (Gclib [gclib](#), string `serial_num`, byte `address`)

## 13.39 ip\_assigner\_example.cpp File Reference

```
#include "examples.h"
#include <iostream>
```

### Functions

- int `main` (int `argc`, char \*`argv[]`)

*Main function for Commands Example.*

### 13.39.1 Detailed Description

See [ip\\_assigner\(\)](#) for implementation of logic

### 13.39.2 Function Documentation

#### 13.39.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.



Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file [commands\\_example.cpp](#).

References [G\\_SMALL\\_BUFFER](#).

## 13.40 ip\_assigner\_example.cs File Reference

### Data Structures

- class [IP\\_Assigner\\_Example](#)

*Assigns controller an IP Adress given a serial number and a 1 byte address.*

### 13.40.1 Detailed Description

See [IP\\_Assigner\(\)](#) for implementation of logic

For VB.NET, see definition in file [ip\\_assigner\\_example.vb](#)

## 13.41 IP\_Assigner\_Example.vb File Reference

### Functions

- int [Main](#) ()

## 13.42 jog.cpp File Reference

```
#include "examples.h"
```

```
#include <conio.h>
```

```
#include <iostream>
```

### Functions

- [GReturn jog](#) ([GCon](#) g)

*Puts controller into Jog Mode and accepts user input to adjust the speed.*

### 13.42.1 Detailed Description

Function calls for the Jog Example Project.

### 13.42.2 Function Documentation

#### 13.42.2.1 jog()

```
GReturn jog (
    GCon g )
```

Puts controller into Jog Mode and accepts user input to adjust the speed.

#### Parameters

<i>g</i>	Connection's handle.
----------	----------------------

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [jog\\_example.cpp](#) for an example.

Key	Usage
q	Quit Jogging
a	-2000 counts / second
s	-500 counts / second
d	+500 counts / second
f	+2000 counts / second
r	Direction Reversal

Definition at line 29 of file jog.cpp.

## 13.43 jog.cs File Reference

### Data Structures

- class [Examples](#)  
*Provides a class of shared constants and methods for gclib's example projects.*

### 13.43.1 Detailed Description

Function calls for the Jog Example Project.

For VB.NET, see definition in file [jog.vb](#)

## 13.44 Jog.vb File Reference

### Functions

- int **Jog** (Gclib [gclib](#))

## 13.45 jog\_example.cpp File Reference

```
#include "examples.h"
```

```
#include <iostream>
```

## Functions

- int [main](#) (int argc, char \*argv[])  
*Main function for Commands Example.*

### 13.45.1 Detailed Description

See [jog\(\)](#) for implementation of logic

### 13.45.2 Function Documentation

#### 13.45.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file `commands_example.cpp`.

References `G_SMALL_BUFFER`.

## 13.46 jog\_example.cs File Reference

### Data Structures

- class [Jog\\_Example](#)

*Accepts user-input at the command line to control the speed of the controller in Jog mode.*

### 13.46.1 Detailed Description

See [Jog\(\)](#) for implementation of logic

For VB.NET, see definition in file [jog\\_example.vb](#)

## 13.47 Jog\_Example.vb File Reference

### Functions

- int [Main](#) ()

## 13.48 main.cpp File Reference

```
#include "Galil.h"
#include <iostream>
#include <iomanip>
#include <Windows.h>
```

### Functions

- int [run](#) (int argc, char \*argv[])
- int [main](#) (int argc, char \*argv[])

## 13.49 message.cpp File Reference

```
#include "examples.h"
#include <iostream>
#include <string.h>
```

### Functions

- [GReturn message](#) ([GCon g](#))

*Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.*

### 13.49.1 Detailed Description

Function calls for the Message Example project

### 13.49.2 Function Documentation

#### 13.49.2.1 message()

```
GReturn message (
    GCon g )
```

Demonstrates how to receive messages from the controller and detect differences in Trace and crashed code.

#### Parameters

<i>g</i>	Connection's handle.
----------	----------------------

### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [message\\_example.cpp](#) for an example.

Definition at line 14 of file `message.cpp`.

Referenced by `Examples::Message()`.

## 13.50 message.cs File Reference

### Data Structures

- class [Examples](#)

*Provides a class of shared constants and methods for gclib's example projects.*

### 13.50.1 Detailed Description

Function calls for the Message Example Project.

For VB.NET, see definition in file [message.vb](#)

## 13.51 Message.vb File Reference

### Functions

- [Message](#) (Gclib [gclib](#))

## 13.52 message\_example.cpp File Reference

```
#include "examples.h"  
#include <iostream>
```

### Functions

- [main](#) (int argc, char \*argv[])

*Main function for Commands Example.*

### 13.52.1 Detailed Description

See [message\(\)](#) for implementation of logic

### 13.52.2 Function Documentation

#### 13.52.2.1 main()

```
int main (  
    int argc,  
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file `commands_example.cpp`.

References `G_SMALL_BUFFER`.

## 13.53 message\_example.cs File Reference

### Data Structures

- class [Message\\_Example](#)

*Demonstrates how to handle and interpret messages from the controller.*

### 13.53.1 Detailed Description

See [Message\(\)](#) for implementation of logic

For VB.NET, see definition in file [message\\_example.vb](#)

## 13.54 Message\_Example.vb File Reference

### Functions

- int [Main](#) ()

## 13.55 motion\_complete.cpp File Reference

```
#include "examples.h"
#include <iostream>
#include <string.h>
```

### Functions

- int [check\\_interrupts](#) ([GCon](#) g, [GCStringIn](#) axes)

*Monitors interrupt status on the given axes and returns when interrupts are fired.*

- [GReturn motion\\_complete](#) (GCon *g*)

*Uses interrupts to track when the motion of controller is completed.*

### 13.55.1 Detailed Description

Function calls for the Motion Complete Example Project.

### 13.55.2 Function Documentation

#### 13.55.2.1 motion\_complete()

`GReturn motion_complete (`  
     `GCon g )`

Uses interrupts to track when the motion of controller is completed.

#### Parameters

<i>g</i>	Connection's handle.
----------	----------------------

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [motion\\_complete\\_example.cpp](#) for an example.

Definition at line 18 of file [motion\\_complete.cpp](#).

## 13.56 motion\_complete.cs File Reference

### Data Structures

- class [Examples](#)

*Provides a class of shared constants and methods for gclib's example projects.*

### 13.56.1 Detailed Description

Function calls for the Motion Complete Example Project.

For VB.NET, see definition in file [motion\\_complete.vb](#)

## 13.57 Motion\_Complete.vb File Reference

### Functions

- [Motion\\_Complete](#) (Gclib [gclib](#))

## 13.58 motion\_complete\_example.cpp File Reference

```
#include "examples.h"
#include <iostream>
```

### Functions

- [int main](#) (int argc, char \*argv[])

*Main function for Commands Example.*

### 13.58.1 Detailed Description

See [motion\\_complete\(\)](#) for implementation of logic

### 13.58.2 Function Documentation

#### 13.58.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file [commands\\_example.cpp](#).

References [G\\_SMALL\\_BUFFER](#).

## 13.59 motion\_complete\_example.cs File Reference

### Data Structures

- class [Motion\\_Complete\\_Example](#)

*Uses controller interrupts to detect when motion is complete.*

### 13.59.1 Detailed Description

See [Motion\\_Complete\(\)](#) for implementation of logic

For VB.NET, see definition in file [motion\\_complete\\_example.vb](#)



## 13.60 Motion\_Complete\_Example.vb File Reference

### Functions

- int [Main](#) ()

## 13.61 position\_tracking.cpp File Reference

```
#include "examples.h"
#include <iostream>
```

### Functions

- [GReturn position\\_tracking](#) ([GCon](#) g, int speed=5000)  
*Puts controller into Position Tracking Mode and accepts user-entered positions.*

#### 13.61.1 Detailed Description

Function calls for the Position Tracking Example Project.

#### 13.61.2 Function Documentation

##### 13.61.2.1 position\_tracking()

```
GReturn position_tracking (
    GCon g,
    int speed = 5000 )
```

Puts controller into Position Tracking Mode and accepts user-entered positions.

##### Parameters

<i>g</i>	Connection's handle.
<i>speed</i>	Optional speed of the controller in Position Tracking Mode. Default value of 5000.

##### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [position\\_tracking\\_example.cpp](#) for an example.  
Definition at line 15 of file position\_tracking.cpp.

## 13.62 position\_tracking.cs File Reference

### Data Structures

- class [Examples](#)  
*Provides a class of shared constants and methods for gclib's example projects.*

#### 13.62.1 Detailed Description

Function calls for the Position Tracking Example Project.

For VB.NET, see definition in file [position\\_tracking.vb](#)

## 13.63 Position\_Tracking.vb File Reference

### Functions

- **Position\_Tracking** (Gclib [gclib](#), int speed)

## 13.64 position\_tracking\_example.cpp File Reference

```
#include "examples.h"
#include <iostream>
```

### Functions

- int **main** (int argc, char \*argv[])  
*Main function for Commands Example.*

### 13.64.1 Detailed Description

See [position\\_tracking\(\)](#) for implementation of logic

### 13.64.2 Function Documentation

#### 13.64.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file `commands_example.cpp`.

References `G_SMALL_BUFFER`.

## 13.65 position\_tracking\_example.cs File Reference

### Data Structures

- class [Position\\_Tracking\\_Example](#)  
*Places controller into position tracking mode. Accepts user-defined positional values at the command line.*

#### 13.65.1 Detailed Description

See [Position\\_Tracking\(\)](#) for implementation of logic

For VB.NET, see definition in file [position\\_tracking\\_example.vb](#)

## 13.66 Position\_Tracking\_Example.vb File Reference

### Functions

- int [Main](#) ()

## 13.67 Program.cs File Reference

### Data Structures

- class [Program](#)

## 13.68 record\_position.cpp File Reference

```
#include "examples.h"  
#include <iostream>  
#include <fstream>
```

### Macros

- `#define G_LASTINDEX 999`

### Functions

- void [write\\_array\\_to\\_file](#) ([GCon](#) g, `ofstream &os`, `const char *array_name`, `int previous_rd`, `int rd`)  
*Grabs data from array on controller and writes it to the given text file.*
- [GReturn record\\_position](#) ([GCon](#) g, `char *fileA`, `char *fileB`)  
*Record user's training and saves to a text file.*

#### 13.68.1 Detailed Description

Function calls for the Record Position Example project

#### 13.68.2 Function Documentation

### 13.68.2.1 record\_position()

```
GReturn record_position (
    GCon g,
    char * fileA,
    char * fileB )
```

Record user's training and saves to a text file.

#### Parameters

<i>g</i>	Connection's handle.
<i>fileA</i>	A Path to a text file where training for Axis A will be recorded.
<i>fileB</i>	A Path to a text file where training for Axis B will be recorded.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [record\\_position\\_example.cpp](#) for an example.

Definition at line 20 of file [record\\_position.cpp](#).

## 13.69 record\_position.cs File Reference

### Data Structures

- class [Examples](#)

*Provides a class of shared constants and methods for gclib's example projects.*

### 13.69.1 Detailed Description

Function calls for the Record Position Example Project.

For VB.NET, see definition in file [record\\_position.vb](#)

## 13.70 Record\_Position.vb File Reference

### Functions

- [Record\\_Position](#) (Gclib [gclib](#), string fileA, string fileB)

## 13.71 record\_position\_example.cpp File Reference

```
#include "examples.h"
#include <iostream>
```

### Functions

- int [main](#) (int argc, char \*argv[])

*Main function for Commands Example.*

### 13.71.1 Detailed Description

See [record\\_position\(\)](#) for implementation of logic

## 13.71.2 Function Documentation

### 13.71.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file [commands\\_example.cpp](#).

References [G\\_SMALL\\_BUFFER](#).

## 13.72 record\_position\_example.cs File Reference

### Data Structures

- class [Record\\_Position\\_Example](#)

*Takes two file paths at the command line to hold positional data for Axis A and Axis B. Positional data is saved to the two files until an analog input value changes.*

### 13.72.1 Detailed Description

See [Record\\_Position\(\)](#) for implementation of logic

For VB.NET, see definition in file [record\\_position\\_example.vb](#)

## 13.73 Record\_Position\_Example.vb File Reference

### Functions

- int [Main](#) ()

## 13.74 remote\_client.cpp File Reference

```
#include "examples.h"
#include <iostream>
#include <vector>
#include <string>
#include <conio.h>
```

### Functions

- void **print\_client\_message** (const char \*[message](#))
- void **print\_servers\_list** (const [std::vector](#)< std::string > &server\_list)
- void **servers\_to\_list** ([std::vector](#)< std::string > &server\_list, std::string servers)
- [GReturn remote\\_client](#) ()

*Lists available remote servers and allows connection to remote server.*

### 13.74.1 Detailed Description

Function calls for the Remote Client Example Project.

### 13.74.2 Function Documentation

#### 13.74.2.1 remote\_client()

[GReturn remote\\_client](#) ( )

Lists available remote servers and allows connection to remote server.

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [remote\\_client\\_example](#) for an example.

Key	Usage
q	Quit
s	List available servers on then network
h	List available hardware on the current server
0-9	Connect to server instance by number
l	Connect back to local server

Definition at line 89 of file [remote\\_client.cpp](#).

References [G\\_SMALL\\_BUFFER](#).

## 13.75 Remote\_Client.cs File Reference

### Data Structures

- class [Examples](#)

*Provides a class of shared constants and methods for gclib's example projects.*

### 13.75.1 Detailed Description

Function calls for the Remote Client Example Project.

For VB.NET, see definition in file [remote\\_client.vb](#)

## 13.76 Remote\_Client.vb File Reference

### Functions

- int [Remote\\_Client](#) ()  
*Demonstrates various uses of [GListServers\(\)](#) and [GSetServer\(\)](#)*

## 13.77 remote\_client\_example.cpp File Reference

```
#include "examples.h"
#include <iostream>
```

### Functions

- int [main](#) (int argc, char \*argv[])  
*Main function for Commands Example.*

### 13.77.1 Detailed Description

See [remote\\_client\(\)](#) for implementation of logic

### 13.77.2 Function Documentation

#### 13.77.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file `commands_example.cpp`.

References `G_SMALL_BUFFER`.

## 13.78 remote\_client\_example.cs File Reference

### Data Structures

- class [Remote\\_Client\\_Example](#)  
*Demonstrates various uses of [GListServers\(\)](#) and [GSetServer\(\)](#)*

### 13.78.1 Detailed Description

See [Remote\\_Client\(\)](#) for implementation of logic

For VB.NET, see definition in file [remote\\_client\\_example.vb](#)

## 13.79 Remote\_Client\_Example.vb File Reference

### Functions

- int [Main](#) ()

## 13.80 remote\_server.cpp File Reference

```
#include "examples.h"
#include <iostream>
#include <conio.h>
```

### Functions

- void [print\\_server\\_message](#) (const char \*[message](#))
- [GReturn remote\\_server](#) (const char \*[server\\_name](#))

*Publishes local gcaps server to the network.*

### 13.80.1 Detailed Description

Function calls for the Remote Server Example Project.

### 13.80.2 Function Documentation



### 13.80.2.1 remote\_server()

```
GReturn remote_server (
    const char * server_name )
```

Publishes local gcaps server to the network.

#### Parameters

<i>Name</i>	to publish server under.
-------------	--------------------------

#### Returns

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See `remote_server_example` for an example.

Key	Usage
q	Quit
p	Publish this server to the network
r	Remove this server from the network

Definition at line 39 of file `remote_server.cpp`.

References `e()`, `G_SMALL_BUFFER`, and `GPublishServer()`.

## 13.81 Remote\_Server.cs File Reference

### Data Structures

- class [Examples](#)

*Provides a class of shared constants and methods for gclib's example projects.*

### 13.81.1 Detailed Description

Function calls for the Remote Server Example Project.

For VB.NET, see definition in file [remote\\_server.vb](#)

## 13.82 Remote\_Server.vb File Reference

### Functions

- int [Remote\\_Server](#) (string server\_name)

*Demonstrates various uses of [GPublishServer\(\)](#)*

## 13.83 remote\_server\_example.cpp File Reference

```
#include "examples.h"
#include <iostream>
#include <string>
```

### Functions

- int [main](#) (int argc, char \*argv[])

*Main function for Commands Example.*

### 13.83.1 Detailed Description

See [remote\\_server\(\)](#) for implementation of logic

### 13.83.2 Function Documentation

#### 13.83.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file `commands_example.cpp`.

References `G_SMALL_BUFFER`.

## 13.84 remote\_server\_example.cs File Reference

### Data Structures

- class [Remote\\_Server\\_Example](#)

*Demonstrates various uses of [GPublishServer\(\)](#)*

### 13.84.1 Detailed Description

See [Remote\\_Server\(\)](#) for implementation of logic

For VB.NET, see definition in file [remote\\_server\\_example.vb](#)

## 13.85 Remote\_Server\_Example.vb File Reference

### Functions

- int [Main](#) ()

## 13.86 setup.py File Reference

### Variables

- name
- version
- description
- author
- author\_email
- url
- py\_modules

## 13.87 vector.cpp File Reference

```
#include "examples.h"  
#include <iostream>  
#include <string>  
#include <fstream>
```

### Functions

- bool [load\\_buffer](#) ([GCon](#) g, ifstream &fs, int capacity)
- [GReturn](#) vector ([GCon](#) g, char \*file)

*Puts controller into Vector Mode and accepts a file defining vector points.*

#### 13.87.1 Detailed Description

Function calls the Vector Mode Example Project.

#### 13.87.2 Function Documentation

##### 13.87.2.1 load\_buffer()

```
bool load_buffer (  
    GCon g,  
    ifstream & fs,  
    int capacity )
```

Loads vector buffer with commands from the given text file.

Returns false when there are no more lines in the text file

Definition at line 88 of file vector.cpp.

##### 13.87.2.2 vector()

```
GReturn vector (  
    GCon g,  
    char * file )
```

Puts controller into Vector Mode and accepts a file defining vector points.

**Parameters**

<i>g</i>	Connection's handle.
<i>file</i>	A Path to a file that defines vector commands.

**Returns**

The success status or error code of the function. See [gclib\\_errors.h](#) for possible values.

See [vector\\_example.cpp](#) for an example.

**Example text file:**

```
VP -2219,-2667
VP -2523,-2832
VP 2844,-1425
VP 728,1971
VP 2127,183
VP -997,688
VP 725,-1893
VP 527,2899
VP -37,2523
VP 1277,1425
VP 857,2388
VP 1096,-1694
CR 1000,0,90
```

Definition at line 36 of file vector.cpp.

## 13.88 vector\_example.cpp File Reference

```
#include "examples.h"
#include <iostream>
```

**Functions**

- `int main (int argc, char *argv[])`  
*Main function for Commands Example.*

### 13.88.1 Detailed Description

See [vector\(\)](#) for implementation of logic

### 13.88.2 Function Documentation

#### 13.88.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Main function for Commands Example.

Main function for Vector Mode Example.

Main function for Remote Server Example.

Main function for Record Position Example.

Main function for Position Tracking Example.

Main Function for Motion Complete Example.

Main function for Message Example.

Main function for Jog Example.

Main function for IP Assigner Example.

Main function for Contour Example.

[commands\\_example.cpp](#) takes one arguments at the command line: an IP Address to a [Galil](#) controllers.

[contour\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[ip\\_assigner\\_example.cpp](#) takes two arguments at the command line: a Serial Number of a [Galil](#) controller and 1 byte address.

[jog\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller. When the program is run the controller will be at rest. Press a key at the console to adjust the speed of the controller.

[message\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[motion\\_complete\\_example.cpp](#) takes one argument at the command line: an IP Address to a [Galil](#) controller.

[position\\_tracking\\_example.cpp](#) takes up to two arguments at the command line: an IP Address to a [Galil](#) controller and an optional speed value. If only one argument is provided the program will default to a speed value of 5000.

[record\\_position\\_example.cpp](#) takes three arguments at the command line: an IP Address to a [Galil](#) controller and a two text files to hold the positional data for two axes.

[remote\\_client\\_example.cpp](#) takes no arguments at the command line.

[remote\\_server\\_example.cpp](#) takes one argument at the command line: the name you wish to publish your server under.

[vector\\_example.cpp](#) takes two arguments at the command line: an IP Address to a [Galil](#) controller and a path to a text file defining vector points. When the program is run the controller will be put into vector mode and loaded with the points defined in the text file. The controller will run until it reaches all points defined in the text file.

Definition at line 18 of file `commands_example.cpp`.

References `G_SMALL_BUFFER`.

## 13.89 vector\_mode.cs File Reference

### Data Structures

- class [Examples](#)

*Provides a class of shared constants and methods for gclib's example projects.*

### 13.89.1 Detailed Description

Function calls for the Vector Mode Example Project.

For VB.NET, see definition in file [vector\\_mode.vb](#)

## 13.90 Vector\_Mode.vb File Reference

### Functions

- int `Vector_Mode` (Gclib [gclib](#), string file)

## 13.91 vector\_mode\_example.cs File Reference

### Data Structures

- class [Vector\\_Mode\\_Example](#)

*Takes a path to a file at the command line holding vector commands for the controller. The controller is placed into vector mode and commands are read from the file and sent to the controller.*

### 13.91.1 Detailed Description

See [Vector\\_Mode\(\)](#) for implementation of logic

For VB.NET, see definition in file [vector\\_mode\\_example.vb](#)

## 13.92 Vector\_Mode\_Example.vb File Reference

### Functions

- int [Main](#) ()

## 13.93 x\_arrays.cpp File Reference

```
#include "x_examples.h"
```

### Functions

- int [x\\_arrays](#) (GCon g)  
*Example [GArrayDownload\(\)](#) and [GArrayUpload\(\)](#) usage.*

### 13.93.1 Detailed Description

Example [GArrayDownload\(\)](#) and [GArrayUpload\(\)](#) usage.

### 13.93.2 Function Documentation

#### 13.93.2.1 x\_arrays()

```
int x_arrays (  
    GCon g )
```

Example [GArrayDownload\(\)](#) and [GArrayUpload\(\)](#) usage.  
Demonstrates array download and upload.  
Definition at line 9 of file `x_arrays.cpp`.

## 13.94 x\_examples.cpp File Reference

```
#include "x_examples.h"  
#include <iomanip>
```

### Functions

- int [main](#) (int argc, char \*argv[])

### 13.94.1 Detailed Description

Examples [main\(\)](#). Calls example code.

## 13.95 x\_examples.h File Reference

```
#include <iostream>  
#include <string>  
#include <cstdio>  
#include <cstring>  
#include "gclib.h"  
#include "gclibo.h"
```

## Macros

- #define `_CRT_SECURE_NO_WARNINGS`
- #define `GALIL_EXAMPLE_OK` `G_NO_ERROR`
- #define `GALIL_EXAMPLE_ERROR` `-100`

## Functions

- void `x_e` (`GReturn` rc)  
*A trivial, C++ style return code check used in Galil's examples and demos.*
- int `x_gcommand` (`GCon` g)  
*Example `GCommand()` usage.*
- int `x_gmotioncomplete` (`GCon` g)  
*Example `GMotionComplete()` usage.*
- int `x_gread_gwrite` (`GCon` g)  
*Example `GRead()` and `GWrite()` usage.*
- int `x_programs` (`GCon` g)  
*Example `GProgramDownload()` and `GProgramUpload()` usage.*
- int `x_arrays` (`GCon` g)  
*Example `GArrayDownload()` and `GArrayUpload()` usage.*
- int `x_grecord` (`GCon` g)  
*Example `GRecord()` usage.*
- int `x_dr_motioncomplete` (`GCon` g, `GCStringIn` axes)  
*Example of `MotionComplete` with data records.*
- int `x_gmessage` (`GCon` g)  
*Example `GMessage()` usage.*
- int `x_ginterrupt` (`GCon` g)  
*Example `GInterrupt()` usage.*
- int `x_ei_motioncomplete` (`GCon` g, `GCStringIn` axes)  
*Example of `MotionComplete` with interrupts.*
- int `x_nonblocking` (`GCon` g)  
*Examples of using non-blocking operation of `GRecord()`, `GInterrupt()`, and `GMessage()`.*

### 13.95.1 Detailed Description

Header file for Galil gclib examples. All example functions start with the letter e. Example function calls demonstrating the use of library functions start with `x_`.

### 13.95.2 Function Documentation

#### 13.95.2.1 `x_arrays()`

```
int x_arrays (
    GCon g )
```

Example `GArrayDownload()` and `GArrayUpload()` usage.

Demonstrates array download and upload.

Definition at line 9 of file `x_arrays.cpp`.

**13.95.2.2 x\_dr\_motioncomplete()**

```
int x_dr_motioncomplete (
    GCon g,
    GCStringIn axes )
```

Example of MotionComplete with data records.

Uses axis status in data record to determine when motion has completed.

**Parameters**

<i>g</i>	Connection's handle.
<i>axes</i>	Mult-axis mask for determining motion complete. ABCDEFGH valid.

Definition at line 92 of file `x_grecord.cpp`.

**13.95.2.3 x\_e()**

```
void x_e (
    GReturn rc ) [inline]
```

A trivial, C++ style return code check used in [Galil's](#) examples and demos.

Throws GReturn if return value is not G\_NO\_ERROR. See [x\\_examples.cpp](#) for example usage and catch() handler.

Definition at line 30 of file `x_examples.h`.

References G\_NO\_ERROR.

**13.95.2.4 x\_ei\_motioncomplete()**

```
int x_ei_motioncomplete (
    GCon g,
    GCStringIn axes )
```

Example of MotionComplete with interrupts.

Uses motion complete status bytes to determine when motion has completed.

**Parameters**

<i>g</i>	Connection's handle.
<i>axes</i>	Mult-axis mask for determining motion complete. ABCDEFGH valid. Axes must be in motion when function is called.

Definition at line 85 of file `x_ginterrupt.cpp`.

**13.95.2.5 x\_gcommand()**

```
int x_gcommand (
    GCon g )
```

Example [GCommand\(\)](#) usage.

Examples of [GCommand\(\)](#) and open-source wrappers like [GCmd\(\)](#).

Definition at line 9 of file `x_gcommand.cpp`.

**13.95.2.6 x\_ginterrupt()**

```
int x_ginterrupt (
    GCon g )
```

Example [GInterrupt\(\)](#) usage.

Demonstrates retrieving status bytes via UI, and a MotionComplete function with interrupts.



**Warning**

This function will attempt to move motors.

Definition at line 9 of file x\_ginterrupt.cpp.

**13.95.2.7 x\_gmessage()**

```
int x_gmessage (
    GCon g )
```

Example [GMessage\(\)](#) usage.

Demonstrates retrieving messages.

Definition at line 9 of file x\_gmessage.cpp.

**13.95.2.8 x\_gmotioncomplete()**

```
int x_gmotioncomplete (
    GCon g )
```

Example [GMotionComplete\(\)](#) usage.

**Warning**

This function will attempt to move motors.

Definition at line 9 of file x\_gmotioncomplete.cpp.

**13.95.2.9 x\_gread\_gwrite()**

```
int x_gread_gwrite (
    GCon g )
```

Example [GRead\(\)](#) and [GWrite\(\)](#) usage.

Demonstrates usage of Read/Write operations.

Definition at line 9 of file x\_gread\_gwrite.cpp.

**13.95.2.10 x\_grecord()**

```
int x_grecord (
    GCon g )
```

Example [GRecord\(\)](#) usage.

Demonstrates QR and DR data records with struct names and pointer arithmetic. Also demonstrates a MotionComplete function with data records.

**Warning**

This function will attempt to move motors.

Definition at line 10 of file x\_grecord.cpp.

**13.95.2.11 x\_nonblocking()**

```
int x_nonblocking (
    GCon g )
```

Examples of using non-blocking operation of [GRecord\(\)](#), [GInterrupt\(\)](#), and [GMessage\(\)](#).

Typical usage of the asynchronous data streams is to call the function with a given timeout. The function will then block until the desired data is received of the timeout occurs. However, to check for available messages, asynchronous records, or interrupts, the user can set a timeout of zero and the functions will return waiting data. See the source of this example for more detail.

Definition at line 22 of file x\_nonblocking.cpp.

### 13.95.2.12 x\_programs()

```
int x_programs (
    GCon g )
```

Example [GProgramDownload\(\)](#) and [GProgramUpload\(\)](#) usage.  
Demonstrates program download and upload including compression.  
Definition at line 9 of file `x_programs.cpp`.

## 13.96 x\_gcommand.cpp File Reference

```
#include "x_examples.h"
```

### Functions

- int [x\\_gcommand](#) (GCon g)  
*Example [GCommand\(\)](#) usage.*

#### 13.96.1 Detailed Description

Example [GCommand\(\)](#) usage.

#### 13.96.2 Function Documentation

##### 13.96.2.1 x\_gcommand()

```
int x_gcommand (
    GCon g )
```

Example [GCommand\(\)](#) usage.  
Examples of [GCommand\(\)](#) and open-source wrappers like [GCmd\(\)](#).  
Definition at line 9 of file `x_gcommand.cpp`.

## 13.97 x\_ginterrupt.cpp File Reference

```
#include "x_examples.h"
```

### Functions

- int [x\\_ginterrupt](#) (GCon g)  
*Example [GInterrupt\(\)](#) usage.*
- int [x\\_ei\\_motioncomplete](#) (GCon g, GCStringIn axes)  
*Example of [MotionComplete](#) with interrupts.*

#### 13.97.1 Detailed Description

Example [GInterrupt\(\)](#) usage.

#### 13.97.2 Function Documentation

### 13.97.2.1 x\_ei\_motioncomplete()

```
int x_ei_motioncomplete (
    GCon g,
    GCStringIn axes )
```

Example of MotionComplete with interrupts.

Uses motion complete status bytes to determine when motion has completed.

#### Parameters

<i>g</i>	Connection's handle.
<i>axes</i>	Multi-axis mask for determining motion complete. ABCDEFGH valid. Axes must be in motion when function is called.

Definition at line 85 of file x\_ginterrupt.cpp.

### 13.97.2.2 x\_ginterrupt()

```
int x_ginterrupt (
    GCon g )
```

Example [GInterrupt\(\)](#) usage.

Demonstrates retrieving status bytes via UI, and a MotionComplete function with interrupts.

#### Warning

This function will attempt to move motors.

Definition at line 9 of file x\_ginterrupt.cpp.

## 13.98 x\_gmessage.cpp File Reference

```
#include "x_examples.h"
```

### Functions

- int [x\\_gmessage](#) (GCon g)  
*Example [GMessage\(\)](#) usage.*

### 13.98.1 Detailed Description

Example [GMessage\(\)](#) usage.

### 13.98.2 Function Documentation

#### 13.98.2.1 x\_gmessage()

```
int x_gmessage (
    GCon g )
```

Example [GMessage\(\)](#) usage.

Demonstrates retrieving messages.

Definition at line 9 of file x\_gmessage.cpp.

## 13.99 x\_gmotioncomplete.cpp File Reference

```
#include "x_examples.h"
```

### Functions

- int [x\\_gmotioncomplete](#) ([GCon g](#))  
*Example [GMotionComplete\(\)](#) usage.*

#### 13.99.1 Detailed Description

Example [GMotionComplete\(\)](#) usage.

#### 13.99.2 Function Documentation

##### 13.99.2.1 x\_gmotioncomplete()

```
int x_gmotioncomplete (  
    GCon g )
```

Example [GMotionComplete\(\)](#) usage.

##### Warning

This function will attempt to move motors.

Definition at line 9 of file [x\\_gmotioncomplete.cpp](#).

## 13.100 x\_gread\_gwrite.cpp File Reference

```
#include "x_examples.h"
```

### Functions

- int [x\\_gread\\_gwrite](#) ([GCon g](#))  
*Example [GRead\(\)](#) and [GWrite\(\)](#) usage.*

#### 13.100.1 Detailed Description

Example [GRead\(\)](#) and [GWrite\(\)](#) usage.

#### 13.100.2 Function Documentation

##### 13.100.2.1 x\_gread\_gwrite()

```
int x_gread_gwrite (  
    GCon g )
```

Example [GRead\(\)](#) and [GWrite\(\)](#) usage.

Demonstrates usage of Read/Write operations.

Definition at line 9 of file [x\\_gread\\_gwrite.cpp](#).

## 13.101 x\_grecord.cpp File Reference

```
#include "x_examples.h"
```

### Functions

- int [x\\_grecord](#) ([GCon g](#))  
*Example [GRecord\(\)](#) usage.*
- int [x\\_dr\\_motioncomplete](#) ([GCon g](#), [GCStringIn axes](#))  
*Example of [MotionComplete](#) with data records.*

#### 13.101.1 Detailed Description

Example [GRecord\(\)](#) usage.

#### 13.101.2 Function Documentation

##### 13.101.2.1 x\_dr\_motioncomplete()

```
int x_dr_motioncomplete (
    GCon g,
    GCStringIn axes )
```

Example of [MotionComplete](#) with data records.

Uses axis status in data record to determine when motion has completed.

##### Parameters

<i>g</i>	Connection's handle.
<i>axes</i>	Multi-axis mask for determining motion complete. ABCDEFGH valid.

Definition at line 92 of file [x\\_grecord.cpp](#).

##### 13.101.2.2 x\_grecord()

```
int x_grecord (
    GCon g )
```

Example [GRecord\(\)](#) usage.

Demonstrates QR and DR data records with struct names and pointer arithmetic. Also demonstrates a [MotionComplete](#) function with data records.

##### Warning

This function will attempt to move motors.

Definition at line 10 of file [x\\_grecord.cpp](#).

## 13.102 x\_nonblocking.cpp File Reference

```
#include "x_examples.h"
#include <iomanip>
```

### Functions

- void [progress](#) ()

- int `x_nonblocking` (`GCon g`)

*Examples of using non-blocking operation of [GRecord\(\)](#), [GInterrupt\(\)](#), and [GMessage\(\)](#).*

## Variables

- int `cur` = 0
- char `chars` [] = { '|', '\\, '-', '/' }

### 13.102.1 Detailed Description

Example usage of `GMessage`, `GRecord`, and `GInterrupt` for non-blocking operation.

### 13.102.2 Function Documentation

#### 13.102.2.1 `x_nonblocking()`

```
int x_nonblocking (
    GCon g )
```

Examples of using non-blocking operation of [GRecord\(\)](#), [GInterrupt\(\)](#), and [GMessage\(\)](#).

Typical usage of the asynchronous data streams is to call the function with a given timeout. The function will then block until the desired data is received or the timeout occurs. However, to check for available messages, asynchronous records, or interrupts, the user can set a timeout of zero and the functions will return waiting data. See the source of this example for more detail.

Definition at line 22 of file `x_nonblocking.cpp`.

## 13.103 `x_programs.cpp` File Reference

```
#include "x_examples.h"
```

## Functions

- int `x_programs` (`GCon g`)

*Example [GProgramDownload\(\)](#) and [GProgramUpload\(\)](#) usage.*

### 13.103.1 Detailed Description

Example [GProgramDownload\(\)](#) and [GProgramUpload\(\)](#) usage.

### 13.103.2 Function Documentation

#### 13.103.2.1 `x_programs()`

```
int x_programs (
    GCon g )
```

Example [GProgramDownload\(\)](#) and [GProgramUpload\(\)](#) usage.

Demonstrates program download and upload including compression.

Definition at line 9 of file `x_programs.cpp`.

## 13.104 x\_simple.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "gclibo.h"
```

### Functions

- void **check** ([GReturn](#) rc)
- int **main** ()

### Variables

- [GCon](#) **g** = 0

#### 13.104.1 Detailed Description

A very simple example for using gclib. See [x\\_examples.cpp](#) for more in-depth examples.





# Index

- `__del__`
    - `gclib.py`, 296
  - `__init__`
    - `gclib.py`, 296
- API, 77
  - `GAddresses`, 84
  - `GArrayDownload`, 84
  - `GArrayDownloadFile`, 85
  - `GArrayUpload`, 85
  - `GArrayUploadFile`, 86
  - `GAssign`, 86
  - `GClose`, 87
  - `GCmd`, 87
  - `GCmdD`, 88
  - `GCmdl`, 88
  - `GCmdT`, 89
  - `GCommand`, 89
  - `GError`, 90
  - `GFirmwareDownload`, 90
  - `GInfo`, 90
  - `GInterrupt`, 91
  - `GIpRequests`, 91
  - `GListServers`, 92
  - `GMessage`, 92
  - `GMotionComplete`, 93
  - `GOpen`, 94
  - `GProgramDownload`, 95
  - `GProgramDownloadFile`, 95
  - `GProgramUpload`, 96
  - `GProgramUploadFile`, 96
  - `GPublishServer`, 97
  - `GRead`, 97
  - `GRecord`, 98
  - `GRecordRate`, 98
  - `GRemoteConnections`, 99
  - `GServerStatus`, 99
  - `GSetServer`, 100
  - `GSetupDownloadFile`, 100
  - `GSleep`, 101
  - `GTimeout`, 102
  - `GUtility`, 102
  - `GVersion`, 105
  - `GWaitForBool`, 105
  - `GWrite`, 106
- `arrays.c`, 307
  - `GArrayDownloadFile`, 308
  - `GArrayUploadFile`, 308
  - `GSetupDownloadFile`, 309
  - `H_DownloadArraysFromList`, 310
- `byte_array`
  - `Gclib.GDataRecord1802`, 187
  - `Gclib.GDataRecord1806`, 207
  - `Gclib.GDataRecord2103`, 222
  - `Gclib.GDataRecord30000`, 227
  - `Gclib.GDataRecord4000`, 248
  - `Gclib.GDataRecord47000_ENC`, 253
  - `Gclib.GDataRecord47162`, 259
  - `Gclib.GDataRecord47300_24EX`, 264
  - `Gclib.GDataRecord47300_ENC`, 270
  - `Gclib.GDataRecord52000`, 290
- C++ examples, 106
  - `commands`, 108
  - `contour`, 108
  - `e`, 109
  - `ip_assigner`, 109
  - `jog`, 109
  - `load_buffer`, 111
  - `main`, 111
  - `message`, 112
  - `motion_complete`, 112
  - `position_tracking`, 112
  - `record_position`, 113
  - `remote_client`, 113
  - `remote_server`, 113
  - `vector`, 114
- C#/VB API, 121
- C#/VB examples, 114
  - `Commands`, 116
  - `Contour`, 116
  - `IP_Assigner`, 117
  - `Jog`, 117
  - `Message`, 118
  - `Motion_Complete`, 118
  - `Position_Tracking`, 119
  - `Record_Position`, 119
  - `Remote_Client`, 120
  - `Remote_Server`, 120
  - `Vector_Mode`, 120
- Commands
  - C#/VB examples, 116
  - Examples, 129
  - examples, 124
- `commands`
  - C++ examples, 108
  - `commands.cpp`, 310
  - `examples.h`, 316
- `commands.cpp`, 310
  - `commands`, 310

- commands.cs, 311
- Commands.vb, 311
- Commands\_Example, 127
  - Main, 127
- commands\_example.cpp, 311
  - main, 311
- commands\_example.cs, 312
- Commands\_Example.vb, 312
- Contour
  - C#/VB examples, 116
  - Examples, 130
- contour
  - C++ examples, 108
  - contour.cpp, 313
  - examples.h, 317
- contour.cpp, 312
  - contour, 313
- contour.cs, 313
- Contour.vb, 313
- Contour\_Example, 128
  - Main, 128
- contour\_example.cpp, 314
  - main, 314
- contour\_example.cs, 315
- Contour\_Example.vb, 315
- e
  - C++ examples, 109
  - examples.h, 317
- example.py, 315
- Examples, 128
  - Commands, 129
  - Contour, 130
  - examples, 125
  - IP\_Assigner, 130
  - Jog, 130
  - Message, 131
  - Motion\_Complete, 131
  - Position\_Tracking, 132
  - PrintError, 132
  - Record\_Position, 133
  - Remote\_Client, 133
  - Remote\_Server, 133
  - Vector\_Mode, 134
- examples, 123
  - Commands, 124
  - Examples, 125
  - Main, 124
  - PrintError, 125
  - Remote\_Client, 125
  - Remote\_Server, 125
- examples.cs, 315
- examples.h, 315
  - commands, 316
  - contour, 317
  - e, 317
  - ip\_assigner, 317
  - jog, 317
  - message, 318
  - motion\_complete, 318
  - position\_tracking, 319
  - record\_position, 319
  - remote\_client, 319
  - remote\_server, 320
  - vector, 320
- Examples.vb, 321
- finalize
  - GclibJava, 164
- Form1.cs, 321
- Form1.vb, 321
- GAddresses
  - API, 84
  - Gclib, 152
  - gclib, 138
  - gclib.py, 296
  - GclibJava, 164
  - gclibo.c, 342
  - gclibo.h, 354
- Galil, 135
- Galil.h, 321
- GalilPrivate, 136
- GArrayDownload
  - API, 84
  - Gclib, 152
  - gclib, 139
  - gclib.h, 326
  - gclib.py, 296
  - GclibJava, 164
- GArrayDownloadFile
  - API, 85
  - arrays.c, 308
  - Gclib, 152
  - gclib, 139
  - gclib.py, 297
  - GclibJava, 165
  - gclibo.h, 355
- GArrayUpload
  - API, 85
  - Gclib, 153
  - gclib, 140
  - gclib.h, 327
  - gclib.py, 297
  - GclibJava, 165
- GArrayUploadFile
  - API, 86
  - arrays.c, 308
  - Gclib, 153
  - gclib, 140
  - gclib.py, 297
  - GclibJava, 165, 166
  - gclibo.h, 355
- GAssign
  - API, 86
  - Gclib, 154
  - gclib, 140
  - gclib.py, 297

- GclibJava, 166
- gclibo.c, 343
- gclibo.h, 356
- gcl\_datarecord.cpp, 321
- gcl\_galil.cpp, 322
- gcl\_galil.h, 322
- gcl\_simple.cpp, 322
- Gclib, 150
  - GAddresses, 152
  - GArrayDownload, 152
  - GArrayDownloadFile, 152
  - GArrayUpload, 153
  - GArrayUploadFile, 153
  - GAssign, 154
  - GClose, 154
  - GCmdD, 154
  - GCmdI, 154
  - GCommand, 155
  - GFirmwareDownload, 155
  - GInfo, 156
  - GInterrupt, 156
  - GIpRequests, 156
  - GListServers, 156
  - GMessage, 156
  - GMotionComplete, 157
  - GOpen, 157
  - GProgramDownload, 157
  - GProgramDownloadFile, 158
  - GProgramUpload, 158
  - GProgramUploadFile, 158
  - GPublishServer, 159
  - GRead, 159
  - GRecord, 159
  - GRecordRate, 160
  - GRemoteConnections, 160
  - GServerStatus, 160
  - GSetServer, 160
  - GSetupDownloadFile, 161
  - GTimeout, 161
  - GVersion, 162
  - GWrite, 162
  - New, 162
- gclib, 136
  - GAddresses, 138
  - GArrayDownload, 139
  - GArrayDownloadFile, 139
  - GArrayUpload, 140
  - GArrayUploadFile, 140
  - GAssign, 140
  - gclib, 138
  - GClose, 141
  - GCmdD, 141
  - GCmdI, 141
  - GCommand, 142
  - GFirmwareDownload, 142
  - GInfo, 143
  - GInterrupt, 143
  - GIpRequests, 143
  - GListServers, 143
  - GMessage, 144
  - GMotionComplete, 144
  - GOpen, 144
  - GProgramDownload, 145
  - GProgramDownloadFile, 145
  - GProgramUpload, 146
  - GProgramUploadFile, 146
  - GPublishServer, 146
  - GRead, 147
  - GRecord< T >, 147
  - GRecordRate, 147
  - GRemoteConnections, 148
  - GServerStatus, 148
  - GSetServer, 148
  - GSetupDownloadFile, 149
  - GTimeout, 149
  - GVersion, 149
  - GWrite, 149
- gclib.cs, 322
- gclib.GclibError, 162
- Gclib.GDataRecord, 176
- gclib.GDataRecord, 174
- Gclib.GDataRecord1802, 184
  - byte\_array, 187
- gclib.GDataRecord1802, 177
- Gclib.GDataRecord1806, 201
  - byte\_array, 207
- gclib.GDataRecord1806, 188
- Gclib.GDataRecord2103, 217
  - byte\_array, 222
- gclib.GDataRecord2103, 208
- Gclib.GDataRecord30000, 226
  - byte\_array, 227
- gclib.GDataRecord30000, 222
- Gclib.GDataRecord4000, 241
  - byte\_array, 248
- gclib.GDataRecord4000, 228
- Gclib.GDataRecord47000\_ENC, 251
  - byte\_array, 253
- gclib.GDataRecord47000\_ENC, 248
- Gclib.GDataRecord47162, 257
  - byte\_array, 259
- gclib.GDataRecord47162, 253
- Gclib.GDataRecord47300\_24EX, 263
  - byte\_array, 264
- gclib.GDataRecord47300\_24EX, 259
- Gclib.GDataRecord47300\_ENC, 268
  - byte\_array, 270
- gclib.GDataRecord47300\_ENC, 265
- Gclib.GDataRecord52000, 284
  - byte\_array, 290
- gclib.GDataRecord52000, 270
- gclib.h, 323
  - GArrayDownload, 326
  - GArrayUpload, 327
  - GClose, 327
  - GCommand, 328

- GFirmwareDownload, 328
- GInterrupt, 328
- GMessage, 329
- GOpen, 330
- GProgramDownload, 331
- GProgramUpload, 332
- GRead, 332
- GRecord, 332
- GUtility, 333
- GWrite, 336
- gclib.py, 295, 336
  - \_\_del\_\_, 296
  - \_\_init\_\_, 296
- GAddresses, 296
- GArrayDownload, 296
- GArrayDownloadFile, 297
- GArrayUpload, 297
- GArrayUploadFile, 297
- GAssign, 297
- GClose, 298
- GCommand, 298
- GFirmwareDownload, 298
- GInfo, 298
- GInterrupt, 298
- GlpRequests, 298
- GMessage, 299
- GMotionComplete, 299
- GOpen, 299
- GProgramDownload, 299
- GProgramDownloadFile, 299
- GProgramUpload, 300
- GProgramUploadFile, 300
- GSetupDownloadFile, 300
- GSleep, 300
- GTimeout, 300
- GVersion, 301
- timeout, 301
- gclib.vb, 337
- gclib\_errors.h, 337
- gclib\_example, 126
- gclib\_record.h, 339
- GclibJava, 163
  - finalize, 164
  - GAddresses, 164
  - GArrayDownload, 164
  - GArrayDownloadFile, 165
  - GArrayUpload, 165
  - GArrayUploadFile, 165, 166
  - GAssign, 166
  - GclibJava, 164
  - GClose, 166
  - GCommand, 167
  - GFirmwareDownload, 167
  - GInfo, 167
  - GInterrupt, 168
  - GlpRequests, 168
  - GListServers, 168
  - GMessage, 168
  - GOpen, 169
  - GProgramDownload, 169, 170
  - GProgramDownloadFile, 170
  - GProgramUpload, 171
  - GProgramUploadFile, 171
  - GPublishServer, 171
  - GRemoteConnections, 172
  - GServerStatus, 172
  - GSetServer, 172
  - GSleep, 172
  - GTimeout, 173
  - GVersion, 173
- gclibjava, 126
- GclibJava.java, 340
- GclibJavaException, 173
- GclibJavaException.java, 340
- gclibo.c, 341
  - GAddresses, 342
  - GAssign, 343
  - GCmd, 343
  - GCmdD, 344
  - GCmdI, 344
  - GCmdT, 344
  - GError, 345
  - GInfo, 345
  - GlpRequests, 346
  - GListServers, 346
  - GMotionComplete, 347
  - GProgramDownloadFile, 347
  - GProgramUploadFile, 348
  - GPublishServer, 348
  - GRecordRate, 349
  - GRemoteConnections, 349
  - GServerStatus, 350
  - GSetServer, 350
  - GSleep, 351
  - GTimeout, 351
  - GVersion, 351
  - GWaitForBool, 352
- gclibo.h, 353
  - GAddresses, 354
  - GArrayDownloadFile, 355
  - GArrayUploadFile, 355
  - GAssign, 356
  - GCmd, 356
  - GCmdD, 357
  - GCmdI, 357
  - GCmdT, 358
  - GError, 358
  - GInfo, 358
  - GlpRequests, 359
  - GListServers, 359
  - GMotionComplete, 360
  - GProgramDownloadFile, 361
  - GProgramUploadFile, 361
  - GPublishServer, 361
  - GRecordRate, 362
  - GRemoteConnections, 362

- GServerStatus, [363](#)
- GSetServer, [363](#)
- GSetupDownloadFile, [364](#)
- GSleep, [365](#)
- GTimeout, [365](#)
- GVersion, [366](#)
- GWaitForBool, [366](#)
- GcLibTest, [174](#)
  - main, [174](#)
- GcLibTest.java, [367](#)
- GClose
  - API, [87](#)
  - GcLib, [154](#)
  - gclib, [141](#)
  - gclib.h, [327](#)
  - gclib.py, [298](#)
  - GcLibJava, [166](#)
- GCmd
  - API, [87](#)
  - gclibo.c, [343](#)
  - gclibo.h, [356](#)
- GCmdD
  - API, [88](#)
  - GcLib, [154](#)
  - gclib, [141](#)
  - gclibo.c, [344](#)
  - gclibo.h, [357](#)
- GCmdl
  - API, [88](#)
  - GcLib, [154](#)
  - gclib, [141](#)
  - gclibo.c, [344](#)
  - gclibo.h, [357](#)
- GCmdT
  - API, [89](#)
  - gclibo.c, [344](#)
  - gclibo.h, [358](#)
- GCommand
  - API, [89](#)
  - GcLib, [155](#)
  - gclib, [142](#)
  - gclib.h, [328](#)
  - gclib.py, [298](#)
  - GcLibJava, [167](#)
- GDataRecord, [175](#)
- GDataRecord1802, [181](#)
- GDataRecord1806, [194](#)
- GDataRecord2103, [212](#)
- GDataRecord30000, [224](#)
- GDataRecord4000, [234](#)
- GDataRecord47000\_ENC, [250](#)
- GDataRecord47162, [255](#)
- GDataRecord47300\_24EX, [261](#)
- GDataRecord47300\_ENC, [266](#)
- GDataRecord52000, [277](#)
- GError
  - API, [90](#)
  - gclibo.c, [345](#)
  - gclibo.h, [358](#)
- GFirmwareDownload
  - API, [90](#)
  - GcLib, [155](#)
  - gclib, [142](#)
  - gclib.h, [328](#)
  - gclib.py, [298](#)
  - GcLibJava, [167](#)
- GInfo
  - API, [90](#)
  - GcLib, [156](#)
  - gclib, [143](#)
  - gclib.py, [298](#)
  - GcLibJava, [167](#)
  - gclibo.c, [345](#)
  - gclibo.h, [358](#)
- GInterrupt
  - API, [91](#)
  - GcLib, [156](#)
  - gclib, [143](#)
  - gclib.h, [328](#)
  - gclib.py, [298](#)
  - GcLibJava, [168](#)
- GlpRequests
  - API, [91](#)
  - GcLib, [156](#)
  - gclib, [143](#)
  - gclib.py, [298](#)
  - GcLibJava, [168](#)
  - gclibo.c, [346](#)
  - gclibo.h, [359](#)
- GListServers
  - API, [92](#)
  - GcLib, [156](#)
  - gclib, [143](#)
  - GcLibJava, [168](#)
  - gclibo.c, [346](#)
  - gclibo.h, [359](#)
- GMessage
  - API, [92](#)
  - GcLib, [156](#)
  - gclib, [144](#)
  - gclib.h, [329](#)
  - gclib.py, [299](#)
  - GcLibJava, [168](#)
- GMotionComplete
  - API, [93](#)
  - GcLib, [157](#)
  - gclib, [144](#)
  - gclib.py, [299](#)
  - gclibo.c, [347](#)
  - gclibo.h, [360](#)
- GOpen
  - API, [94](#)
  - GcLib, [157](#)
  - gclib, [144](#)
  - gclib.h, [330](#)
  - gclib.py, [299](#)

- GclibJava, 169
- GProgramDownload
  - API, 95
  - Gclib, 157
  - gclib, 145
  - gclib.h, 331
  - gclib.py, 299
  - GclibJava, 169, 170
- GProgramDownloadFile
  - API, 95
  - Gclib, 158
  - gclib, 145
  - gclib.py, 299
  - GclibJava, 170
  - gclibo.c, 347
  - gclibo.h, 361
- GProgramUpload
  - API, 96
  - Gclib, 158
  - gclib, 146
  - gclib.h, 332
  - gclib.py, 300
  - GclibJava, 171
- GProgramUploadFile
  - API, 96
  - Gclib, 158
  - gclib, 146
  - gclib.py, 300
  - GclibJava, 171
  - gclibo.c, 348
  - gclibo.h, 361
- GPublishServer
  - API, 97
  - Gclib, 159
  - gclib, 146
  - GclibJava, 171
  - gclibo.c, 348
  - gclibo.h, 361
- GRead
  - API, 97
  - Gclib, 159
  - gclib, 147
  - gclib.h, 332
- GRecord
  - API, 98
  - Gclib, 159
  - gclib.h, 332
- GRecord< T >
  - gclib, 147
- GRecordRate
  - API, 98
  - Gclib, 160
  - gclib, 147
  - gclibo.c, 349
  - gclibo.h, 362
- GRemoteConnections
  - API, 99
  - Gclib, 160
  - gclib, 148
  - GclibJava, 172
  - gclibo.c, 349
  - gclibo.h, 362
- GServerStatus
  - API, 99
  - Gclib, 160
  - gclib, 148
  - GclibJava, 172
  - gclibo.c, 350
  - gclibo.h, 363
- GSetServer
  - API, 100
  - Gclib, 160
  - gclib, 148
  - GclibJava, 172
  - gclibo.c, 350
  - gclibo.h, 363
- GSetupDownloadFile
  - API, 100
  - arrays.c, 309
  - Gclib, 161
  - gclib, 149
  - gclib.py, 300
  - gclibo.h, 364
- GSleep
  - API, 101
  - gclib.py, 300
  - GclibJava, 172
  - gclibo.c, 351
  - gclibo.h, 365
- GTimeout
  - API, 102
  - Gclib, 161
  - gclib, 149
  - gclib.py, 300
  - GclibJava, 173
  - gclibo.c, 351
  - gclibo.h, 365
- GUtility
  - API, 102
  - gclib.h, 333
- GVersion
  - API, 105
  - Gclib, 162
  - gclib, 149
  - gclib.py, 301
  - GclibJava, 173
  - gclibo.c, 351
  - gclibo.h, 366
- GWaitForBool
  - API, 105
  - gclibo.c, 352
  - gclibo.h, 366
- GWrite
  - API, 106
  - Gclib, 162
  - gclib, 149

- gclib.h, 336
- H\_ArrayData, 290
- H\_DownloadArraysFromList
  - arrays.c, 310
- IP\_Assigner
  - C#/VB examples, 117
  - Examples, 130
- ip\_assigner
  - C++ examples, 109
  - examples.h, 317
  - ip\_assigner.cpp, 367
- ip\_assigner.cpp, 367
  - ip\_assigner, 367
- ip\_assigner.cs, 368
- IP\_Assigner.vb, 368
- IP\_Assigner\_Example, 291
  - Main, 291
- ip\_assigner\_example.cpp, 368
  - main, 368
- ip\_assigner\_example.cs, 369
- IP\_Assigner\_Example.vb, 369
- Jog
  - C#/VB examples, 117
  - Examples, 130
- jog
  - C++ examples, 109
  - examples.h, 317
  - jog.cpp, 370
- jog.cpp, 369
  - jog, 370
- jog.cs, 370
- Jog.vb, 370
- Jog\_Example, 292
  - Main, 292
- jog\_example.cpp, 370
  - main, 371
- jog\_example.cs, 371
- Jog\_Example.vb, 372
- load\_buffer
  - C++ examples, 111
  - vector.cpp, 387
- Main
  - Commands\_Example, 127
  - Contour\_Example, 128
  - examples, 124
  - IP\_Assigner\_Example, 291
  - Jog\_Example, 292
  - Message\_Example, 293
  - Motion\_Complete\_Example, 294
  - Position\_Tracking\_Example, 294
  - Record\_Position\_Example, 302
  - Remote\_Client\_Example, 302
  - Remote\_Server\_Example, 303
  - Vector\_Mode\_Example, 304
- main
  - C++ examples, 111
  - commands\_example.cpp, 311
  - contour\_example.cpp, 314
  - GclibTest, 174
  - ip\_assigner\_example.cpp, 368
  - jog\_example.cpp, 371
  - message\_example.cpp, 373
  - motion\_complete\_example.cpp, 376
  - position\_tracking\_example.cpp, 378
  - record\_position\_example.cpp, 381
  - remote\_client\_example.cpp, 383
  - remote\_server\_example.cpp, 386
  - vector\_example.cpp, 388
- main.cpp, 372
- MainForm, 292
- Message
  - C#/VB examples, 118
  - Examples, 131
- message
  - C++ examples, 112
  - examples.h, 318
  - message.cpp, 372
- message.cpp, 372
  - message, 372
- message.cs, 373
- Message.vb, 373
- Message\_Example, 293
  - Main, 293
- message\_example.cpp, 373
  - main, 373
- message\_example.cs, 374
- Message\_Example.vb, 374
- Motion\_Complete
  - C#/VB examples, 118
  - Examples, 131
- motion\_complete
  - C++ examples, 112
  - examples.h, 318
  - motion\_complete.cpp, 375
- motion\_complete.cpp, 374
  - motion\_complete, 375
- motion\_complete.cs, 375
- Motion\_Complete.vb, 375
- Motion\_Complete\_Example, 293
  - Main, 294
- motion\_complete\_example.cpp, 375
  - main, 376
- motion\_complete\_example.cs, 376
- Motion\_Complete\_Example.vb, 377
- New
  - Gclib, 162
- Position\_Tracking
  - C#/VB examples, 119
  - Examples, 132
- position\_tracking
  - C++ examples, 112

- examples.h, 319
  - position\_tracking.cpp, 377
- position\_tracking.cpp, 377
  - position\_tracking, 377
- position\_tracking.cs, 377
- Position\_Tracking.vb, 378
- Position\_Tracking\_Example, 294
  - Main, 294
- position\_tracking\_example.cpp, 378
  - main, 378
- position\_tracking\_example.cs, 379
- Position\_Tracking\_Example.vb, 379
- PrintError
  - Examples, 132
  - examples, 125
- Program, 295
- Program.cs, 379
- Record\_Position
  - C#/VB examples, 119
  - Examples, 133
- record\_position
  - C++ examples, 113
  - examples.h, 319
  - record\_position.cpp, 379
- record\_position.cpp, 379
  - record\_position, 379
- record\_position.cs, 380
- Record\_Position.vb, 380
- Record\_Position\_Example, 301
  - Main, 302
- record\_position\_example.cpp, 380
  - main, 381
- record\_position\_example.cs, 381
- Record\_Position\_Example.vb, 382
- Remote\_Client
  - C#/VB examples, 120
  - Examples, 133
  - examples, 125
- remote\_client
  - C++ examples, 113
  - examples.h, 319
  - remote\_client.cpp, 382
- remote\_client.cpp, 382
  - remote\_client, 382
- Remote\_Client.cs, 382
- Remote\_Client.vb, 383
- Remote\_Client\_Example, 302
  - Main, 302
- remote\_client\_example.cpp, 383
  - main, 383
- remote\_client\_example.cs, 384
- Remote\_Client\_Example.vb, 384
- Remote\_Server
  - C#/VB examples, 120
  - Examples, 133
  - examples, 125
- remote\_server
  - C++ examples, 113
  - examples.h, 320
  - remote\_server.cpp, 384
- remote\_server.cpp, 384
  - remote\_server, 384
- Remote\_Server.cs, 385
- Remote\_Server.vb, 385
- Remote\_Server\_Example, 303
  - Main, 303
- remote\_server\_example.cpp, 385
  - main, 386
- remote\_server\_example.cs, 386
- Remote\_Server\_Example.vb, 387
- setup.py, 387
- Source, 304
- timeout
  - gclib.py, 301
- vb, 126
- vector
  - C++ examples, 114
  - examples.h, 320
  - vector.cpp, 387
- vector.cpp, 387
  - load\_buffer, 387
  - vector, 387
- vector\_example.cpp, 388
  - main, 388
- Vector\_Mode
  - C#/VB examples, 120
  - Examples, 134
- vector\_mode.cs, 389
- Vector\_Mode.vb, 389
- Vector\_Mode\_Example, 304
  - Main, 304
- vector\_mode\_example.cs, 389
- Vector\_Mode\_Example.vb, 390
- x\_arrays
  - x\_arrays.cpp, 390
  - x\_examples.h, 391
- x\_arrays.cpp, 390
  - x\_arrays, 390
- x\_dr\_motioncomplete
  - x\_examples.h, 391
  - x\_grecord.cpp, 397
- x\_e
  - x\_examples.h, 392
- x\_ei\_motioncomplete
  - x\_examples.h, 392
  - x\_ginterrupt.cpp, 394
- x\_examples.cpp, 390
- x\_examples.h, 390
  - x\_arrays, 391
  - x\_dr\_motioncomplete, 391
  - x\_e, 392
  - x\_ei\_motioncomplete, 392
  - x\_gcommand, 392



- x\_ginterrupt, 392
- x\_gmessage, 393
- x\_gmotioncomplete, 393
- x\_gread\_gwrite, 393
- x\_grecord, 393
- x\_nonblocking, 393
- x\_programs, 393
- x\_gcommand
  - x\_examples.h, 392
  - x\_gcommand.cpp, 394
- x\_gcommand.cpp, 394
  - x\_gcommand, 394
- x\_ginterrupt
  - x\_examples.h, 392
  - x\_ginterrupt.cpp, 395
- x\_ginterrupt.cpp, 394
  - x\_ei\_motioncomplete, 394
  - x\_ginterrupt, 395
- x\_gmessage
  - x\_examples.h, 393
  - x\_gmessage.cpp, 395
- x\_gmessage.cpp, 395
  - x\_gmessage, 395
- x\_gmotioncomplete
  - x\_examples.h, 393
  - x\_gmotioncomplete.cpp, 396
- x\_gmotioncomplete.cpp, 396
  - x\_gmotioncomplete, 396
- x\_gread\_gwrite
  - x\_examples.h, 393
  - x\_gread\_gwrite.cpp, 396
- x\_gread\_gwrite.cpp, 396
  - x\_gread\_gwrite, 396
- x\_grecord
  - x\_examples.h, 393
  - x\_grecord.cpp, 397
- x\_grecord.cpp, 397
  - x\_dr\_motioncomplete, 397
  - x\_grecord, 397
- x\_nonblocking
  - x\_examples.h, 393
  - x\_nonblocking.cpp, 398
- x\_nonblocking.cpp, 397
  - x\_nonblocking, 398
- x\_programs
  - x\_examples.h, 393
  - x\_programs.cpp, 398
- x\_programs.cpp, 398
  - x\_programs, 398
- x\_simple.c, 399