gclib

2.5.0

Generated by Doxygen 1.14.0

1 Galil Communications Library (gclib) 1

1.1 Installation . . . 1

1.2 Other Languages . . . 1

1.3 Quickstart . . . 1

2 Examples 3

2.1 Controller Addresses . . . 3

2.2 Connection Management . . . 4

2.3 Commands . . . 5

2.4 Errors . . . 6

2.5 Program & Arrays . . . 7

2.6 Unsolicited Data . . . 9

2.7 Galil Connect . . . 12

2.8 Example Project: Record and Replay . . . 13

3 License 19

4 Program Preprocessor 21

4.1 The `preprocessor` argument . . . 21

4.2 In-band Operation . . . 21

4.2.1 The `REM` Comment . . . 21

4.2.2 Double Hash . . . 21

4.2.3 Quad Hash to enable . . . 22

4.2.4 C-style comments . . . 22

4.2.5 Preprocessor Directives . . . 22

4.2.5.1 `##option` . . . 22

4.2.5.2 `##include` . . . 22

4.2.5.3 `##gclib` . . . 22

4.2.6 In-band Support . . . 23

4.3 Preprocessor Options . . . 23

4.3.1 Compression, `--min`, `--max` . . . 23

4.3.1.1 Compression Levels, *n* . . . 23

4.3.2 Code insertion, `--insert` . . . 23

4.3.3 Include Search Paths, `--search`, `-I` . . . 24

4.3.3.1 In-band Example . . . 24

4.3.4 Macro Definition, `--define`, `-D` . . . 24

4.3.5 Conditional Directives, `--ifdef`, `--ifndef` . . . 25

4.4 GDK Support . . . 25

5 Deprecated List 27

6 Topic Index 31

6.1 Topics . . . 31

7 Namespace Index **33**
7.1 Namespace List . . . 33

8 Hierarchical Index **35**
8.1 Class Hierarchy . . . 35

9 Class Index **37**
9.1 Class List . . . 37

10 File Index **39**
10.1 File List . . . 39

11 Topic Documentation **41**
11.1 C . . . 41
11.1.1 Detailed Description . . . 42
11.1.2 Typedef Documentation . . . 42
11.1.2.1 gclib_handle . . . 42
11.1.2.2 gclib_data_record_handle . . . 42
11.1.2.3 gclib_result . . . 43
11.1.2.4 gclib_interrupt_type . . . 43
11.1.2.5 gclib_interrupt_t . . . 43
11.1.2.6 gclib_digital_input_flags . . . 43
11.1.2.7 gclib_axis_flags . . . 43
11.1.3 Enumeration Type Documentation . . . 43
11.1.3.1 gclib_result . . . 43
11.1.3.2 gclib_interrupt_type . . . 43
11.1.3.3 gclib_digital_input_flags . . . 44
11.1.3.4 gclib_axis_flags . . . 44
11.1.4 Function Documentation . . . 45
11.1.4.1 gclib_version() . . . 45
11.1.4.2 gclib_gcaps_version() . . . 45
11.1.5 Connection . . . 45
11.1.5.1 Detailed Description . . . 46
11.1.5.2 Typedef Documentation . . . 46
11.1.5.3 Enumeration Type Documentation . . . 46
11.1.5.4 Function Documentation . . . 47
11.1.5.5 Galil Connect . . . 50
11.1.6 Controller . . . 52
11.1.6.1 Detailed Description . . . 52
11.1.6.2 Function Documentation . . . 53
11.1.6.3 Data Record . . . 54
11.1.6.4 Memory . . . 65
11.1.6.5 Unsolicited Data . . . 67
11.2 C (Legacy) . . . 73

11.2.1 Detailed Description . . . 74
11.2.2 Function Documentation . . . 74
11.2.2.1 GSleep() . . . 74
11.2.2.2 GVersion() . . . 74
11.2.2.3 GError() . . . 75
11.2.3 Connection . . . 75
11.2.3.1 Detailed Description . . . 75
11.2.3.2 Function Documentation . . . 75
11.2.4 Controller . . . 80
11.2.4.1 Detailed Description . . . 80
11.2.4.2 Function Documentation . . . 81
11.2.4.3 Communication . . . 82
11.2.4.4 Memory . . . 85
11.2.4.5 Unsolicited Data . . . 90
11.2.5 Galil Connect . . . 93
11.2.5.1 Detailed Description . . . 93
11.2.5.2 Function Documentation . . . 93

12 Namespace Documentation **97**
12.1 gclib Namespace Reference . . . 97
12.1.1 Enumeration Type Documentation . . . 98
12.1.1.1 ConnectionType . . . 98
12.1.2 Function Documentation . . . 98
12.1.2.1 version() . . . 98
12.1.2.2 addresses() . . . 98
12.1.2.3 ip_requests() . . . 98
12.1.2.4 assign_ip() . . . 99
12.1.2.5 force_gcaps() . . . 99
12.1.2.6 gcaps_version() . . . 99
12.1.2.7 list_servers() . . . 99
12.1.2.8 server() . . . 99
12.1.2.9 set_server() . . . 99
12.1.2.10 published() . . . 99
12.1.2.11 set_published() . . . 99
12.2 gclib::internal Namespace Reference . . . 100

13 Class Documentation **101**
13.1 DataRecord::Amp Struct Reference . . . 101
13.1.1 Detailed Description . . . 101
13.1.2 Constructor & Destructor Documentation . . . 101
13.1.2.1 Amp() . . . 101
13.1.3 Member Function Documentation . . . 101
13.1.3.1 over_current() . . . 101

13.1.3.2 over_voltage() . . . 101
13.1.3.3 over_temp() . . . 102
13.1.3.4 under_voltage() . . . 102
13.1.3.5 electronic_lock_out() . . . 102
13.2 DataRecord::Axis Struct Reference . . . 102
13.2.1 Detailed Description . . . 103
13.2.2 Member Enumeration Documentation . . . 103
13.2.2.1 ModeOfMotion . . . 103
13.2.3 Constructor & Destructor Documentation . . . 103
13.2.3.1 Axis() . . . 103
13.2.4 Member Function Documentation . . . 103
13.2.4.1 motor_off() . . . 103
13.2.4.2 latch_armed() . . . 103
13.2.4.3 final_deceleration() . . . 103
13.2.4.4 stopping() . . . 104
13.2.4.5 slewing() . . . 104
13.2.4.6 mode_of_motion() . . . 104
13.2.4.7 negative_direction_move() . . . 104
13.2.4.8 moving() . . . 104
13.2.4.9 home_phase() . . . 104
13.2.4.10 stepper_mode() . . . 104
13.2.4.11 home_input() . . . 104
13.2.4.12 reverse_limit() . . . 104
13.2.4.13 forward_limit() . . . 104
13.2.4.14 latch_input() . . . 104
13.2.4.15 latch_occurred() . . . 104
13.2.4.16 stop_code() . . . 105
13.2.4.17 reference_position() . . . 105
13.2.4.18 position() . . . 105
13.2.4.19 position_error() . . . 105
13.2.4.20 aux_position() . . . 105
13.2.4.21 velocity() . . . 105
13.2.4.22 torque() . . . 105
13.2.4.23 analog_input() . . . 105
13.2.4.24 halls() . . . 105
13.2.4.25 variable() . . . 105
13.2.4.26 hall_error() . . . 105
13.2.4.27 peak_current() . . . 105
13.3 Controller Class Reference . . . 106
13.3.1 Detailed Description . . . 107
13.3.2 Constructor & Destructor Documentation . . . 107
13.3.2.1 Controller() [1/2] . . . 107

13.3.2.2 Controller() [2/2] . . . 107
13.3.2.3 ~Controller() . . . 107
13.3.3 Member Function Documentation . . . 107
13.3.3.1 open() . . . 107
13.3.3.2 close() . . . 107
13.3.3.3 address() . . . 107
13.3.3.4 connection_type() . . . 107
13.3.3.5 revision_information() . . . 107
13.3.3.6 serial_number() . . . 108
13.3.3.7 command() . . . 108
13.3.3.8 program() . . . 108
13.3.3.9 set_program() . . . 108
13.3.3.10 array() . . . 108
13.3.3.11 set_array() . . . 108
13.3.3.12 set_firmware() . . . 108
13.3.3.13 set_interrupts() . . . 109
13.3.3.14 set_data_records() . . . 109
13.3.3.15 subscribe_messages() [1/2] . . . 109
13.3.3.16 subscribe_messages() [2/2] . . . 110
13.3.3.17 subscribe_interrupts() [1/2] . . . 110
13.3.3.18 subscribe_interrupts() [2/2] . . . 110
13.3.3.19 subscribe_data_records() [1/2] . . . 110
13.3.3.20 subscribe_data_records() [2/2] . . . 111
13.3.3.21 subscribe_progress() . . . 111
13.3.3.22 unsubscribe_messages() . . . 111
13.3.3.23 unsubscribe_interrupts() . . . 111
13.3.3.24 unsubscribe_data_records() . . . 111
13.3.3.25 unsubscribe_progress() . . . 111
13.3.3.26 message() . . . 111
13.3.3.27 interrupt() . . . 112
13.3.3.28 data_record() . . . 112
13.4 DataRecord::CoordinatedMove Struct Reference . . . 112
13.4.1 Detailed Description . . . 112
13.4.2 Constructor & Destructor Documentation . . . 113
13.4.2.1 CoordinatedMove() . . . 113
13.4.3 Member Function Documentation . . . 113
13.4.3.1 segment_count() . . . 113
13.4.3.2 status() . . . 113
13.4.3.3 final_deceleration() . . . 113
13.4.3.4 stopping() . . . 113
13.4.3.5 slewing() . . . 113
13.4.3.6 moving() . . . 113

13.4.3.7 distance() . . . 113
13.4.3.8 buffer_available() . . . 113
13.5 DataRecord Struct Reference . . . 113
13.5.1 Detailed Description . . . 114
13.5.2 Member Enumeration Documentation . . . 114
13.5.2.1 EthernetStatus . . . 114
13.5.3 Constructor & Destructor Documentation . . . 115
13.5.3.1 DataRecord() . . . 115
13.5.4 Member Function Documentation . . . 115
13.5.4.1 sample() . . . 115
13.5.4.2 input() . . . 115
13.5.4.3 output() . . . 115
13.5.4.4 input_bank() . . . 116
13.5.4.5 output_bank() . . . 116
13.5.4.6 error_code() . . . 116
13.5.4.7 contour_segment_count() . . . 116
13.5.4.8 contour_buffer_available() . . . 116
13.5.4.9 thread_running() . . . 116
13.5.4.10 ethernet_status() . . . 116
13.5.4.11 bytes() . . . 116
13.5.4.12 amp() . . . 116
13.5.4.13 axis() . . . 117
13.5.4.14 coordinated_move() . . . 117
13.6 error Class Reference . . . 117
13.6.1 Detailed Description . . . 117
13.7 gclib_interrupt_t Struct Reference . . . 117
13.7.1 Detailed Description . . . 118
13.7.2 Member Data Documentation . . . 118
13.7.2.1 type . . . 118
13.7.2.2 axis . . . 118
13.7.2.3 digital_input . . . 118
13.7.2.4 user_interrupt . . . 118
13.7.2.5 [union] . . . 118
13.7.2.6 status . . . 118
13.8 GDataRecord Union Reference . . . 118
13.8.1 Detailed Description . . . 119
13.8.2 Member Data Documentation . . . 119
13.8.2.1 dmc4000 . . . 119
13.8.2.2 dmc4103 . . . 119
13.8.2.3 dmc50000 . . . 119
13.8.2.4 dmc52000 . . . 119
13.8.2.5 dmc30000 . . . 120

13.8.2.6 dmc2103 . . . 120
13.8.2.7 dmc1806 . . . 120
13.8.2.8 dmc1802 . . . 120
13.8.2.9 rio47000 . . . 120
13.8.2.10 rio47300 . . . 120
13.8.2.11 rio47300_24ex . . . 120
13.8.2.12 rio47162 . . . 120
13.8.2.13 byte_array . . . 120
13.9 GDataRecord1802 Struct Reference . . . 120
13.9.1 Detailed Description . . . 124
13.9.2 Member Data Documentation . . . 124
13.9.2.1 sample_number . . . 124
13.9.2.2 input_bank_0 . . . 124
13.9.2.3 input_bank_1 . . . 124
13.9.2.4 input_bank_2 . . . 124
13.9.2.5 input_bank_3 . . . 124
13.9.2.6 input_bank_4 . . . 124
13.9.2.7 input_bank_5 . . . 124
13.9.2.8 input_bank_6 . . . 124
13.9.2.9 input_bank_7 . . . 125
13.9.2.10 input_bank_8 . . . 125
13.9.2.11 input_bank_9 . . . 125
13.9.2.12 output_bank_0 . . . 125
13.9.2.13 output_bank_1 . . . 125
13.9.2.14 output_bank_2 . . . 125
13.9.2.15 output_bank_3 . . . 125
13.9.2.16 output_bank_4 . . . 125
13.9.2.17 output_bank_5 . . . 125
13.9.2.18 output_bank_6 . . . 125
13.9.2.19 output_bank_7 . . . 126
13.9.2.20 output_bank_8 . . . 126
13.9.2.21 output_bank_9 . . . 126
13.9.2.22 error_code . . . 126
13.9.2.23 general_status . . . 126
13.9.2.24 s_plane_segment_count . . . 126
13.9.2.25 s_plane_move_status . . . 126
13.9.2.26 s_distance . . . 126
13.9.2.27 t_plane_segment_count . . . 126
13.9.2.28 t_plane_move_status . . . 126
13.9.2.29 t_distance . . . 127
13.9.2.30 axis_a_status . . . 127
13.9.2.31 axis_a_switches . . . 127

13.9.2.32 axis_a_stop_code . . . 127
13.9.2.33 axis_a_reference_position . . . 127
13.9.2.34 axis_a_motor_position . . . 127
13.9.2.35 axis_a_position_error . . . 127
13.9.2.36 axis_a_aux_position . . . 127
13.9.2.37 axis_a_velocity . . . 127
13.9.2.38 axis_a_torque . . . 127
13.9.2.39 axis_a_reserved_0 . . . 128
13.9.2.40 axis_a_reserved_1 . . . 128
13.9.2.41 axis_b_status . . . 128
13.9.2.42 axis_b_switches . . . 128
13.9.2.43 axis_b_stop_code . . . 128
13.9.2.44 axis_b_reference_position . . . 128
13.9.2.45 axis_b_motor_position . . . 128
13.9.2.46 axis_b_position_error . . . 128
13.9.2.47 axis_b_aux_position . . . 128
13.9.2.48 axis_b_velocity . . . 128
13.9.2.49 axis_b_torque . . . 129
13.9.2.50 axis_b_reserved_0 . . . 129
13.9.2.51 axis_b_reserved_1 . . . 129
13.9.2.52 axis_c_status . . . 129
13.9.2.53 axis_c_switches . . . 129
13.9.2.54 axis_c_stop_code . . . 129
13.9.2.55 axis_c_reference_position . . . 129
13.9.2.56 axis_c_motor_position . . . 129
13.9.2.57 axis_c_position_error . . . 129
13.9.2.58 axis_c_aux_position . . . 129
13.9.2.59 axis_c_velocity . . . 130
13.9.2.60 axis_c_torque . . . 130
13.9.2.61 axis_c_reserved_0 . . . 130
13.9.2.62 axis_c_reserved_1 . . . 130
13.9.2.63 axis_d_status . . . 130
13.9.2.64 axis_d_switches . . . 130
13.9.2.65 axis_d_stop_code . . . 130
13.9.2.66 axis_d_reference_position . . . 130
13.9.2.67 axis_d_motor_position . . . 130
13.9.2.68 axis_d_position_error . . . 130
13.9.2.69 axis_d_aux_position . . . 131
13.9.2.70 axis_d_velocity . . . 131
13.9.2.71 axis_d_torque . . . 131
13.9.2.72 axis_d_reserved_0 . . . 131
13.9.2.73 axis_d_reserved_1 . . . 131

13.10 GDataRecord1806 Struct Reference . . . 131
13.10.1 Detailed Description . . . 137
13.10.2 Member Data Documentation . . . 138
13.10.2.1 sample_number . . . 138
13.10.2.2 input_bank_0 . . . 138
13.10.2.3 input_bank_1 . . . 138
13.10.2.4 input_bank_2 . . . 138
13.10.2.5 input_bank_3 . . . 138
13.10.2.6 input_bank_4 . . . 138
13.10.2.7 input_bank_5 . . . 138
13.10.2.8 input_bank_6 . . . 138
13.10.2.9 input_bank_7 . . . 138
13.10.2.10 input_bank_8 . . . 138
13.10.2.11 input_bank_9 . . . 139
13.10.2.12 output_bank_0 . . . 139
13.10.2.13 output_bank_1 . . . 139
13.10.2.14 output_bank_2 . . . 139
13.10.2.15 output_bank_3 . . . 139
13.10.2.16 output_bank_4 . . . 139
13.10.2.17 output_bank_5 . . . 139
13.10.2.18 output_bank_6 . . . 139
13.10.2.19 output_bank_7 . . . 139
13.10.2.20 output_bank_8 . . . 139
13.10.2.21 output_bank_9 . . . 140
13.10.2.22 reserved_0 . . . 140
13.10.2.23 reserved_2 . . . 140
13.10.2.24 reserved_4 . . . 140
13.10.2.25 reserved_6 . . . 140
13.10.2.26 reserved_8 . . . 140
13.10.2.27 reserved_10 . . . 140
13.10.2.28 reserved_12 . . . 140
13.10.2.29 reserved_14 . . . 140
13.10.2.30 reserved_16 . . . 140
13.10.2.31 reserved_17 . . . 141
13.10.2.32 reserved_18 . . . 141
13.10.2.33 reserved_19 . . . 141
13.10.2.34 reserved_20 . . . 141
13.10.2.35 reserved_21 . . . 141
13.10.2.36 reserved_22 . . . 141
13.10.2.37 reserved_23 . . . 141
13.10.2.38 error_code . . . 141
13.10.2.39 thread_status . . . 141

13.10.2.40 reserved_24 . . . 141
13.10.2.41 contour_segment_count . . . 142
13.10.2.42 contour_buffer_available . . . 142
13.10.2.43 s_plane_segment_count . . . 142
13.10.2.44 s_plane_move_status . . . 142
13.10.2.45 s_distance . . . 142
13.10.2.46 s_plane_buffer_available . . . 142
13.10.2.47 t_plane_segment_count . . . 142
13.10.2.48 t_plane_move_status . . . 142
13.10.2.49 t_distance . . . 142
13.10.2.50 t_plane_buffer_available . . . 142
13.10.2.51 axis_a_status . . . 143
13.10.2.52 axis_a_switches . . . 143
13.10.2.53 axis_a_stop_code . . . 143
13.10.2.54 axis_a_reference_position . . . 143
13.10.2.55 axis_a_motor_position . . . 143
13.10.2.56 axis_a_position_error . . . 143
13.10.2.57 axis_a_aux_position . . . 143
13.10.2.58 axis_a_velocity . . . 143
13.10.2.59 axis_a_torque . . . 143
13.10.2.60 axis_a_analog_in . . . 143
13.10.2.61 axis_a_reserved_0 . . . 144
13.10.2.62 axis_a_reserved_1 . . . 144
13.10.2.63 axis_a_variable . . . 144
13.10.2.64 axis_b_status . . . 144
13.10.2.65 axis_b_switches . . . 144
13.10.2.66 axis_b_stop_code . . . 144
13.10.2.67 axis_b_reference_position . . . 144
13.10.2.68 axis_b_motor_position . . . 144
13.10.2.69 axis_b_position_error . . . 144
13.10.2.70 axis_b_aux_position . . . 144
13.10.2.71 axis_b_velocity . . . 145
13.10.2.72 axis_b_torque . . . 145
13.10.2.73 axis_b_analog_in . . . 145
13.10.2.74 axis_b_reserved_0 . . . 145
13.10.2.75 axis_b_reserved_1 . . . 145
13.10.2.76 axis_b_variable . . . 145
13.10.2.77 axis_c_status . . . 145
13.10.2.78 axis_c_switches . . . 145
13.10.2.79 axis_c_stop_code . . . 145
13.10.2.80 axis_c_reference_position . . . 145
13.10.2.81 axis_c_motor_position . . . 146

13.10.2.82 axis_c_position_error . . . 146
13.10.2.83 axis_c_aux_position . . . 146
13.10.2.84 axis_c_velocity . . . 146
13.10.2.85 axis_c_torque . . . 146
13.10.2.86 axis_c_analog_in . . . 146
13.10.2.87 axis_c_reserved_0 . . . 146
13.10.2.88 axis_c_reserved_1 . . . 146
13.10.2.89 axis_c_variable . . . 146
13.10.2.90 axis_d_status . . . 146
13.10.2.91 axis_d_switches . . . 147
13.10.2.92 axis_d_stop_code . . . 147
13.10.2.93 axis_d_reference_position . . . 147
13.10.2.94 axis_d_motor_position . . . 147
13.10.2.95 axis_d_position_error . . . 147
13.10.2.96 axis_d_aux_position . . . 147
13.10.2.97 axis_d_velocity . . . 147
13.10.2.98 axis_d_torque . . . 147
13.10.2.99 axis_d_analog_in . . . 147
13.10.2.100 axis_d_reserved_0 . . . 147
13.10.2.101 axis_d_reserved_1 . . . 148
13.10.2.102 axis_d_variable . . . 148
13.10.2.103 axis_e_status . . . 148
13.10.2.104 axis_e_switches . . . 148
13.10.2.105 axis_e_stop_code . . . 148
13.10.2.106 axis_e_reference_position . . . 148
13.10.2.107 axis_e_motor_position . . . 148
13.10.2.108 axis_e_position_error . . . 148
13.10.2.109 axis_e_aux_position . . . 148
13.10.2.110 axis_e_velocity . . . 148
13.10.2.111 axis_e_torque . . . 149
13.10.2.112 axis_e_analog_in . . . 149
13.10.2.113 axis_e_reserved_0 . . . 149
13.10.2.114 axis_e_reserved_1 . . . 149
13.10.2.115 axis_e_variable . . . 149
13.10.2.116 axis_f_status . . . 149
13.10.2.117 axis_f_switches . . . 149
13.10.2.118 axis_f_stop_code . . . 149
13.10.2.119 axis_f_reference_position . . . 149
13.10.2.120 axis_f_motor_position . . . 149
13.10.2.121 axis_f_position_error . . . 150
13.10.2.122 axis_f_aux_position . . . 150
13.10.2.123 axis_f_velocity . . . 150

13.10.2.124 axis_f_torque . . . 150
13.10.2.125 axis_f_analog_in . . . 150
13.10.2.126 axis_f_reserved_0 . . . 150
13.10.2.127 axis_f_reserved_1 . . . 150
13.10.2.128 axis_f_variable . . . 150
13.10.2.129 axis_g_status . . . 150
13.10.2.130 axis_g_switches . . . 150
13.10.2.131 axis_g_stop_code . . . 151
13.10.2.132 axis_g_reference_position . . . 151
13.10.2.133 axis_g_motor_position . . . 151
13.10.2.134 axis_g_position_error . . . 151
13.10.2.135 axis_g_aux_position . . . 151
13.10.2.136 axis_g_velocity . . . 151
13.10.2.137 axis_g_torque . . . 151
13.10.2.138 axis_g_analog_in . . . 151
13.10.2.139 axis_g_reserved_0 . . . 151
13.10.2.140 axis_g_reserved_1 . . . 151
13.10.2.141 axis_g_variable . . . 152
13.10.2.142 axis_h_status . . . 152
13.10.2.143 axis_h_switches . . . 152
13.10.2.144 axis_h_stop_code . . . 152
13.10.2.145 axis_h_reference_position . . . 152
13.10.2.146 axis_h_motor_position . . . 152
13.10.2.147 axis_h_position_error . . . 152
13.10.2.148 axis_h_aux_position . . . 152
13.10.2.149 axis_h_velocity . . . 152
13.10.2.150 axis_h_torque . . . 152
13.10.2.151 axis_h_analog_in . . . 153
13.10.2.152 axis_h_reserved_0 . . . 153
13.10.2.153 axis_h_reserved_1 . . . 153
13.10.2.154 axis_h_variable . . . 153
13.11 GDataRecord2103 Struct Reference . . . 153
13.11.1 Detailed Description . . . 158
13.11.2 Member Data Documentation . . . 158
13.11.2.1 header_0 . . . 158
13.11.2.2 header_1 . . . 158
13.11.2.3 header_2 . . . 158
13.11.2.4 header_3 . . . 158
13.11.2.5 sample_number . . . 158
13.11.2.6 input_bank_0 . . . 158
13.11.2.7 input_bank_1 . . . 158
13.11.2.8 input_bank_2 . . . 158

13.11.2.9 input_bank_3 . . . 158
13.11.2.10 input_bank_4 . . . 159
13.11.2.11 input_bank_5 . . . 159
13.11.2.12 input_bank_6 . . . 159
13.11.2.13 input_bank_7 . . . 159
13.11.2.14 input_bank_8 . . . 159
13.11.2.15 input_bank_9 . . . 159
13.11.2.16 output_bank_0 . . . 159
13.11.2.17 output_bank_1 . . . 159
13.11.2.18 output_bank_2 . . . 159
13.11.2.19 output_bank_3 . . . 159
13.11.2.20 output_bank_4 . . . 160
13.11.2.21 output_bank_5 . . . 160
13.11.2.22 output_bank_6 . . . 160
13.11.2.23 output_bank_7 . . . 160
13.11.2.24 output_bank_8 . . . 160
13.11.2.25 output_bank_9 . . . 160
13.11.2.26 error_code . . . 160
13.11.2.27 general_status . . . 160
13.11.2.28 s_plane_segment_count . . . 160
13.11.2.29 s_plane_move_status . . . 160
13.11.2.30 s_distance . . . 161
13.11.2.31 t_plane_segment_count . . . 161
13.11.2.32 t_plane_move_status . . . 161
13.11.2.33 t_distance . . . 161
13.11.2.34 axis_a_status . . . 161
13.11.2.35 axis_a_switches . . . 161
13.11.2.36 axis_a_stop_code . . . 161
13.11.2.37 axis_a_reference_position . . . 161
13.11.2.38 axis_a_motor_position . . . 161
13.11.2.39 axis_a_position_error . . . 161
13.11.2.40 axis_a_aux_position . . . 162
13.11.2.41 axis_a_velocity . . . 162
13.11.2.42 axis_a_torque . . . 162
13.11.2.43 axis_a_analog_in . . . 162
13.11.2.44 axis_b_status . . . 162
13.11.2.45 axis_b_switches . . . 162
13.11.2.46 axis_b_stop_code . . . 162
13.11.2.47 axis_b_reference_position . . . 162
13.11.2.48 axis_b_motor_position . . . 162
13.11.2.49 axis_b_position_error . . . 162
13.11.2.50 axis_b_aux_position . . . 163

13.11.2.51 axis_b_velocity . . . 163
13.11.2.52 axis_b_torque . . . 163
13.11.2.53 axis_b_analog_in . . . 163
13.11.2.54 axis_c_status . . . 163
13.11.2.55 axis_c_switches . . . 163
13.11.2.56 axis_c_stop_code . . . 163
13.11.2.57 axis_c_reference_position . . . 163
13.11.2.58 axis_c_motor_position . . . 163
13.11.2.59 axis_c_position_error . . . 163
13.11.2.60 axis_c_aux_position . . . 164
13.11.2.61 axis_c_velocity . . . 164
13.11.2.62 axis_c_torque . . . 164
13.11.2.63 axis_c_analog_in . . . 164
13.11.2.64 axis_d_status . . . 164
13.11.2.65 axis_d_switches . . . 164
13.11.2.66 axis_d_stop_code . . . 164
13.11.2.67 axis_d_reference_position . . . 164
13.11.2.68 axis_d_motor_position . . . 164
13.11.2.69 axis_d_position_error . . . 164
13.11.2.70 axis_d_aux_position . . . 165
13.11.2.71 axis_d_velocity . . . 165
13.11.2.72 axis_d_torque . . . 165
13.11.2.73 axis_d_analog_in . . . 165
13.11.2.74 axis_e_status . . . 165
13.11.2.75 axis_e_switches . . . 165
13.11.2.76 axis_e_stop_code . . . 165
13.11.2.77 axis_e_reference_position . . . 165
13.11.2.78 axis_e_motor_position . . . 165
13.11.2.79 axis_e_position_error . . . 165
13.11.2.80 axis_e_aux_position . . . 166
13.11.2.81 axis_e_velocity . . . 166
13.11.2.82 axis_e_torque . . . 166
13.11.2.83 axis_e_analog_in . . . 166
13.11.2.84 axis_f_status . . . 166
13.11.2.85 axis_f_switches . . . 166
13.11.2.86 axis_f_stop_code . . . 166
13.11.2.87 axis_f_reference_position . . . 166
13.11.2.88 axis_f_motor_position . . . 166
13.11.2.89 axis_f_position_error . . . 166
13.11.2.90 axis_f_aux_position . . . 167
13.11.2.91 axis_f_velocity . . . 167
13.11.2.92 axis_f_torque . . . 167

13.11.2.93 axis_f_analog_in . . . 167
13.11.2.94 axis_g_status . . . 167
13.11.2.95 axis_g_switches . . . 167
13.11.2.96 axis_g_stop_code . . . 167
13.11.2.97 axis_g_reference_position . . . 167
13.11.2.98 axis_g_motor_position . . . 167
13.11.2.99 axis_g_position_error . . . 167
13.11.2.100 axis_g_aux_position . . . 168
13.11.2.101 axis_g_velocity . . . 168
13.11.2.102 axis_g_torque . . . 168
13.11.2.103 axis_g_analog_in . . . 168
13.11.2.104 axis_h_status . . . 168
13.11.2.105 axis_h_switches . . . 168
13.11.2.106 axis_h_stop_code . . . 168
13.11.2.107 axis_h_reference_position . . . 168
13.11.2.108 axis_h_motor_position . . . 168
13.11.2.109 axis_h_position_error . . . 168
13.11.2.110 axis_h_aux_position . . . 169
13.11.2.111 axis_h_velocity . . . 169
13.11.2.112 axis_h_torque . . . 169
13.11.2.113 axis_h_analog_in . . . 169
13.12 GDataRecord30000 Struct Reference . . . 169
13.12.1 Detailed Description . . . 170
13.12.2 Member Data Documentation . . . 171
13.12.2.1 header_0 . . . 171
13.12.2.2 header_1 . . . 171
13.12.2.3 header_2 . . . 171
13.12.2.4 header_3 . . . 171
13.12.2.5 sample_number . . . 171
13.12.2.6 input_bank_0 . . . 171
13.12.2.7 input_bank_1 . . . 171
13.12.2.8 output_bank_0 . . . 171
13.12.2.9 output_bank_1 . . . 171
13.12.2.10 error_code . . . 171
13.12.2.11 thread_status . . . 172
13.12.2.12 input_analog_2 . . . 172
13.12.2.13 output_analog_1 . . . 172
13.12.2.14 output_analog_2 . . . 172
13.12.2.15 amplifier_status . . . 172
13.12.2.16 contour_segment_count . . . 172
13.12.2.17 contour_buffer_available . . . 172
13.12.2.18 s_plane_segment_count . . . 172

13.12.2.19 s_plane_move_status . . . 172
13.12.2.20 s_distance . . . 172
13.12.2.21 s_plane_buffer_available . . . 173
13.12.2.22 axis_a_status . . . 173
13.12.2.23 axis_a_switches . . . 173
13.12.2.24 axis_a_stop_code . . . 173
13.12.2.25 axis_a_reference_position . . . 173
13.12.2.26 axis_a_motor_position . . . 173
13.12.2.27 axis_a_position_error . . . 173
13.12.2.28 axis_a_aux_position . . . 173
13.12.2.29 axis_a_velocity . . . 173
13.12.2.30 axis_a_torque . . . 173
13.12.2.31 axis_a_analog_in . . . 174
13.12.2.32 axis_a_halls . . . 174
13.12.2.33 axis_a_reserved . . . 174
13.12.2.34 axis_a_variable . . . 174
13.13 GDataRecord4000 Struct Reference . . . 174
13.13.1 Detailed Description . . . 180
13.13.2 Member Data Documentation . . . 180
13.13.2.1 header_0 . . . 180
13.13.2.2 header_1 . . . 181
13.13.2.3 header_2 . . . 181
13.13.2.4 header_3 . . . 181
13.13.2.5 sample_number . . . 181
13.13.2.6 input_bank_0 . . . 181
13.13.2.7 input_bank_1 . . . 181
13.13.2.8 input_bank_2 . . . 181
13.13.2.9 input_bank_3 . . . 181
13.13.2.10 input_bank_4 . . . 181
13.13.2.11 input_bank_5 . . . 181
13.13.2.12 input_bank_6 . . . 182
13.13.2.13 input_bank_7 . . . 182
13.13.2.14 input_bank_8 . . . 182
13.13.2.15 input_bank_9 . . . 182
13.13.2.16 output_bank_0 . . . 182
13.13.2.17 output_bank_1 . . . 182
13.13.2.18 output_bank_2 . . . 182
13.13.2.19 output_bank_3 . . . 182
13.13.2.20 output_bank_4 . . . 182
13.13.2.21 output_bank_5 . . . 182
13.13.2.22 output_bank_6 . . . 183
13.13.2.23 output_bank_7 . . . 183

13.13.2.24 output_bank_8 . . . 183
13.13.2.25 output_bank_9 . . . 183
13.13.2.26 reserved_0 . . . 183
13.13.2.27 reserved_2 . . . 183
13.13.2.28 reserved_4 . . . 183
13.13.2.29 reserved_6 . . . 183
13.13.2.30 reserved_8 . . . 183
13.13.2.31 reserved_10 . . . 183
13.13.2.32 reserved_12 . . . 184
13.13.2.33 reserved_14 . . . 184
13.13.2.34 ethernet_status_a . . . 184
13.13.2.35 ethernet_status_b . . . 184
13.13.2.36 ethernet_status_c . . . 184
13.13.2.37 ethernet_status_d . . . 184
13.13.2.38 ethernet_status_e . . . 184
13.13.2.39 ethernet_status_f . . . 184
13.13.2.40 ethernet_status_g . . . 184
13.13.2.41 ethernet_status_h . . . 184
13.13.2.42 error_code . . . 185
13.13.2.43 thread_status . . . 185
13.13.2.44 amplifier_status . . . 185
13.13.2.45 contour_segment_count . . . 185
13.13.2.46 contour_buffer_available . . . 185
13.13.2.47 s_plane_segment_count . . . 185
13.13.2.48 s_plane_move_status . . . 185
13.13.2.49 s_distance . . . 185
13.13.2.50 s_plane_buffer_available . . . 185
13.13.2.51 t_plane_segment_count . . . 185
13.13.2.52 t_plane_move_status . . . 186
13.13.2.53 t_distance . . . 186
13.13.2.54 t_plane_buffer_available . . . 186
13.13.2.55 axis_a_status . . . 186
13.13.2.56 axis_a_switches . . . 186
13.13.2.57 axis_a_stop_code . . . 186
13.13.2.58 axis_a_reference_position . . . 186
13.13.2.59 axis_a_motor_position . . . 186
13.13.2.60 axis_a_position_error . . . 186
13.13.2.61 axis_a_aux_position . . . 186
13.13.2.62 axis_a_velocity . . . 187
13.13.2.63 axis_a_torque . . . 187
13.13.2.64 axis_a_analog_in . . . 187
13.13.2.65 axis_a_halls . . . 187

13.13.2.66 axis_a_reserved . . . 187
13.13.2.67 axis_a_variable . . . 187
13.13.2.68 axis_b_status . . . 187
13.13.2.69 axis_b_switches . . . 187
13.13.2.70 axis_b_stop_code . . . 187
13.13.2.71 axis_b_reference_position . . . 187
13.13.2.72 axis_b_motor_position . . . 188
13.13.2.73 axis_b_position_error . . . 188
13.13.2.74 axis_b_aux_position . . . 188
13.13.2.75 axis_b_velocity . . . 188
13.13.2.76 axis_b_torque . . . 188
13.13.2.77 axis_b_analog_in . . . 188
13.13.2.78 axis_b_halls . . . 188
13.13.2.79 axis_b_reserved . . . 188
13.13.2.80 axis_b_variable . . . 188
13.13.2.81 axis_c_status . . . 188
13.13.2.82 axis_c_switches . . . 189
13.13.2.83 axis_c_stop_code . . . 189
13.13.2.84 axis_c_reference_position . . . 189
13.13.2.85 axis_c_motor_position . . . 189
13.13.2.86 axis_c_position_error . . . 189
13.13.2.87 axis_c_aux_position . . . 189
13.13.2.88 axis_c_velocity . . . 189
13.13.2.89 axis_c_torque . . . 189
13.13.2.90 axis_c_analog_in . . . 189
13.13.2.91 axis_c_halls . . . 189
13.13.2.92 axis_c_reserved . . . 190
13.13.2.93 axis_c_variable . . . 190
13.13.2.94 axis_d_status . . . 190
13.13.2.95 axis_d_switches . . . 190
13.13.2.96 axis_d_stop_code . . . 190
13.13.2.97 axis_d_reference_position . . . 190
13.13.2.98 axis_d_motor_position . . . 190
13.13.2.99 axis_d_position_error . . . 190
13.13.2.100 axis_d_aux_position . . . 190
13.13.2.101 axis_d_velocity . . . 190
13.13.2.102 axis_d_torque . . . 191
13.13.2.103 axis_d_analog_in . . . 191
13.13.2.104 axis_d_halls . . . 191
13.13.2.105 axis_d_reserved . . . 191
13.13.2.106 axis_d_variable . . . 191
13.13.2.107 axis_e_status . . . 191

13.13.2.108 axis_e_switches . . . 191
13.13.2.109 axis_e_stop_code . . . 191
13.13.2.110 axis_e_reference_position . . . 191
13.13.2.111 axis_e_motor_position . . . 191
13.13.2.112 axis_e_position_error . . . 192
13.13.2.113 axis_e_aux_position . . . 192
13.13.2.114 axis_e_velocity . . . 192
13.13.2.115 axis_e_torque . . . 192
13.13.2.116 axis_e_analog_in . . . 192
13.13.2.117 axis_e_halls . . . 192
13.13.2.118 axis_e_reserved . . . 192
13.13.2.119 axis_e_variable . . . 192
13.13.2.120 axis_f_status . . . 192
13.13.2.121 axis_f_switches . . . 192
13.13.2.122 axis_f_stop_code . . . 193
13.13.2.123 axis_f_reference_position . . . 193
13.13.2.124 axis_f_motor_position . . . 193
13.13.2.125 axis_f_position_error . . . 193
13.13.2.126 axis_f_aux_position . . . 193
13.13.2.127 axis_f_velocity . . . 193
13.13.2.128 axis_f_torque . . . 193
13.13.2.129 axis_f_analog_in . . . 193
13.13.2.130 axis_f_halls . . . 193
13.13.2.131 axis_f_reserved . . . 193
13.13.2.132 axis_f_variable . . . 194
13.13.2.133 axis_g_status . . . 194
13.13.2.134 axis_g_switches . . . 194
13.13.2.135 axis_g_stop_code . . . 194
13.13.2.136 axis_g_reference_position . . . 194
13.13.2.137 axis_g_motor_position . . . 194
13.13.2.138 axis_g_position_error . . . 194
13.13.2.139 axis_g_aux_position . . . 194
13.13.2.140 axis_g_velocity . . . 194
13.13.2.141 axis_g_torque . . . 194
13.13.2.142 axis_g_analog_in . . . 195
13.13.2.143 axis_g_halls . . . 195
13.13.2.144 axis_g_reserved . . . 195
13.13.2.145 axis_g_variable . . . 195
13.13.2.146 axis_h_status . . . 195
13.13.2.147 axis_h_switches . . . 195
13.13.2.148 axis_h_stop_code . . . 195
13.13.2.149 axis_h_reference_position . . . 195

13.13.2.150 axis_h_motor_position . . . 195
13.13.2.151 axis_h_position_error . . . 195
13.13.2.152 axis_h_aux_position . . . 196
13.13.2.153 axis_h_velocity . . . 196
13.13.2.154 axis_h_torque . . . 196
13.13.2.155 axis_h_analog_in . . . 196
13.13.2.156 axis_h_halls . . . 196
13.13.2.157 axis_h_reserved . . . 196
13.13.2.158 axis_h_variable . . . 196
13.14 GDataRecord47000_ENC Struct Reference . . . 196
13.14.1 Detailed Description . . . 198
13.14.2 Member Data Documentation . . . 198
13.14.2.1 header_0 . . . 198
13.14.2.2 header_1 . . . 198
13.14.2.3 header_2 . . . 198
13.14.2.4 header_3 . . . 198
13.14.2.5 sample_number . . . 198
13.14.2.6 error_code . . . 198
13.14.2.7 general_status . . . 198
13.14.2.8 output_analog_0 . . . 199
13.14.2.9 output_analog_1 . . . 199
13.14.2.10 output_analog_2 . . . 199
13.14.2.11 output_analog_3 . . . 199
13.14.2.12 output_analog_4 . . . 199
13.14.2.13 output_analog_5 . . . 199
13.14.2.14 output_analog_6 . . . 199
13.14.2.15 output_analog_7 . . . 199
13.14.2.16 input_analog_0 . . . 199
13.14.2.17 input_analog_1 . . . 199
13.14.2.18 input_analog_2 . . . 200
13.14.2.19 input_analog_3 . . . 200
13.14.2.20 input_analog_4 . . . 200
13.14.2.21 input_analog_5 . . . 200
13.14.2.22 input_analog_6 . . . 200
13.14.2.23 input_analog_7 . . . 200
13.14.2.24 output_bank_0 . . . 200
13.14.2.25 input_bank_0 . . . 200
13.14.2.26 pulse_count_0 . . . 200
13.14.2.27 zc_variable . . . 200
13.14.2.28 zd_variable . . . 201
13.14.2.29 encoder_0 . . . 201
13.14.2.30 encoder_1 . . . 201

13.14.2.31 encoder_2 . . . 201
13.14.2.32 encoder_3 . . . 201
13.15 GDataRecord47162 Struct Reference . . . 201
13.15.1 Detailed Description . . . 203
13.15.2 Member Data Documentation . . . 203
13.15.2.1 header_0 . . . 203
13.15.2.2 header_1 . . . 203
13.15.2.3 header_2 . . . 203
13.15.2.4 header_3 . . . 203
13.15.2.5 sample_number . . . 203
13.15.2.6 error_code . . . 203
13.15.2.7 general_status . . . 203
13.15.2.8 output_analog_0 . . . 204
13.15.2.9 output_analog_1 . . . 204
13.15.2.10 output_analog_2 . . . 204
13.15.2.11 output_analog_3 . . . 204
13.15.2.12 output_analog_4 . . . 204
13.15.2.13 output_analog_5 . . . 204
13.15.2.14 output_analog_6 . . . 204
13.15.2.15 output_analog_7 . . . 204
13.15.2.16 input_analog_0 . . . 204
13.15.2.17 input_analog_1 . . . 204
13.15.2.18 input_analog_2 . . . 205
13.15.2.19 input_analog_3 . . . 205
13.15.2.20 input_analog_4 . . . 205
13.15.2.21 input_analog_5 . . . 205
13.15.2.22 input_analog_6 . . . 205
13.15.2.23 input_analog_7 . . . 205
13.15.2.24 output_byte_0 . . . 205
13.15.2.25 output_byte_1 . . . 205
13.15.2.26 output_byte_2 . . . 205
13.15.2.27 input_byte_0 . . . 205
13.15.2.28 input_byte_1 . . . 206
13.15.2.29 input_byte_2 . . . 206
13.15.2.30 input_byte_3 . . . 206
13.15.2.31 input_byte_4 . . . 206
13.15.2.32 pulse_count_0 . . . 206
13.15.2.33 zc_variable . . . 206
13.15.2.34 zd_variable . . . 206
13.15.2.35 encoder_0 . . . 206
13.15.2.36 encoder_1 . . . 206
13.15.2.37 encoder_2 . . . 206

13.15.2.38 encoder_3 . . . 207
13.16 GDataRecord47300_24EX Struct Reference . . . 207
13.16.1 Detailed Description . . . 208
13.16.2 Member Data Documentation . . . 208
13.16.2.1 header_0 . . . 208
13.16.2.2 header_1 . . . 208
13.16.2.3 header_2 . . . 208
13.16.2.4 header_3 . . . 209
13.16.2.5 sample_number . . . 209
13.16.2.6 error_code . . . 209
13.16.2.7 general_status . . . 209
13.16.2.8 output_analog_0 . . . 209
13.16.2.9 output_analog_1 . . . 209
13.16.2.10 output_analog_2 . . . 209
13.16.2.11 output_analog_3 . . . 209
13.16.2.12 output_analog_4 . . . 209
13.16.2.13 output_analog_5 . . . 209
13.16.2.14 output_analog_6 . . . 210
13.16.2.15 output_analog_7 . . . 210
13.16.2.16 input_analog_0 . . . 210
13.16.2.17 input_analog_1 . . . 210
13.16.2.18 input_analog_2 . . . 210
13.16.2.19 input_analog_3 . . . 210
13.16.2.20 input_analog_4 . . . 210
13.16.2.21 input_analog_5 . . . 210
13.16.2.22 input_analog_6 . . . 210
13.16.2.23 input_analog_7 . . . 210
13.16.2.24 output_bank_0 . . . 211
13.16.2.25 output_bank_1 . . . 211
13.16.2.26 input_bank_0 . . . 211
13.16.2.27 input_bank_1 . . . 211
13.16.2.28 pulse_count_0 . . . 211
13.16.2.29 zc_variable . . . 211
13.16.2.30 zd_variable . . . 211
13.16.2.31 output_bank_2 . . . 211
13.16.2.32 output_back_3 . . . 211
13.16.2.33 input_bank_2 . . . 211
13.16.2.34 input_bank_3 . . . 212
13.17 GDataRecord47300_ENC Struct Reference . . . 212
13.17.1 Detailed Description . . . 213
13.17.2 Member Data Documentation . . . 213
13.17.2.1 header_0 . . . 213

13.17.2.2 header_1 . . . 213
13.17.2.3 header_2 . . . 213
13.17.2.4 header_3 . . . 214
13.17.2.5 sample_number . . . 214
13.17.2.6 error_code . . . 214
13.17.2.7 general_status . . . 214
13.17.2.8 output_analog_0 . . . 214
13.17.2.9 output_analog_1 . . . 214
13.17.2.10 output_analog_2 . . . 214
13.17.2.11 output_analog_3 . . . 214
13.17.2.12 output_analog_4 . . . 214
13.17.2.13 output_analog_5 . . . 214
13.17.2.14 output_analog_6 . . . 215
13.17.2.15 output_analog_7 . . . 215
13.17.2.16 input_analog_0 . . . 215
13.17.2.17 input_analog_1 . . . 215
13.17.2.18 input_analog_2 . . . 215
13.17.2.19 input_analog_3 . . . 215
13.17.2.20 input_analog_4 . . . 215
13.17.2.21 input_analog_5 . . . 215
13.17.2.22 input_analog_6 . . . 215
13.17.2.23 input_analog_7 . . . 215
13.17.2.24 output_bank_0 . . . 216
13.17.2.25 output_bank_1 . . . 216
13.17.2.26 input_bank_0 . . . 216
13.17.2.27 input_bank_1 . . . 216
13.17.2.28 pulse_count_0 . . . 216
13.17.2.29 zc_variable . . . 216
13.17.2.30 zd_variable . . . 216
13.17.2.31 encoder_0 . . . 216
13.17.2.32 encoder_1 . . . 216
13.17.2.33 encoder_2 . . . 216
13.17.2.34 encoder_3 . . . 217
13.18 GDataRecord52000 Struct Reference . . . 217
13.18.1 Detailed Description . . . 223
13.18.2 Member Data Documentation . . . 223
13.18.2.1 header_0 . . . 223
13.18.2.2 header_1 . . . 223
13.18.2.3 header_2 . . . 223
13.18.2.4 header_3 . . . 224
13.18.2.5 sample_number . . . 224
13.18.2.6 input_bank_0 . . . 224

13.18.2.7 input_bank_1 . . . 224
13.18.2.8 input_bank_2 . . . 224
13.18.2.9 input_bank_3 . . . 224
13.18.2.10 input_bank_4 . . . 224
13.18.2.11 input_bank_5 . . . 224
13.18.2.12 input_bank_6 . . . 224
13.18.2.13 input_bank_7 . . . 224
13.18.2.14 input_bank_8 . . . 225
13.18.2.15 input_bank_9 . . . 225
13.18.2.16 output_bank_0 . . . 225
13.18.2.17 output_bank_1 . . . 225
13.18.2.18 output_bank_2 . . . 225
13.18.2.19 output_bank_3 . . . 225
13.18.2.20 output_bank_4 . . . 225
13.18.2.21 output_bank_5 . . . 225
13.18.2.22 output_bank_6 . . . 225
13.18.2.23 output_bank_7 . . . 225
13.18.2.24 output_bank_8 . . . 226
13.18.2.25 output_bank_9 . . . 226
13.18.2.26 reserved_0 . . . 226
13.18.2.27 reserved_2 . . . 226
13.18.2.28 reserved_4 . . . 226
13.18.2.29 reserved_6 . . . 226
13.18.2.30 reserved_8 . . . 226
13.18.2.31 reserved_10 . . . 226
13.18.2.32 reserved_12 . . . 226
13.18.2.33 ethercat_bank . . . 226
13.18.2.34 reserved_14 . . . 227
13.18.2.35 ethernet_status_a . . . 227
13.18.2.36 ethernet_status_b . . . 227
13.18.2.37 ethernet_status_c . . . 227
13.18.2.38 ethernet_status_d . . . 227
13.18.2.39 ethernet_status_e . . . 227
13.18.2.40 ethernet_status_f . . . 227
13.18.2.41 ethernet_status_g . . . 227
13.18.2.42 ethernet_status_h . . . 227
13.18.2.43 error_code . . . 227
13.18.2.44 thread_status . . . 228
13.18.2.45 amplifier_status . . . 228
13.18.2.46 contour_segment_count . . . 228
13.18.2.47 contour_buffer_available . . . 228
13.18.2.48 s_plane_segment_count . . . 228

13.18.2.49 s_plane_move_status . . . 228
13.18.2.50 s_distance . . . 228
13.18.2.51 s_plane_buffer_available . . . 228
13.18.2.52 t_plane_segment_count . . . 228
13.18.2.53 t_plane_move_status . . . 228
13.18.2.54 t_distance . . . 229
13.18.2.55 t_plane_buffer_available . . . 229
13.18.2.56 axis_a_status . . . 229
13.18.2.57 axis_a_switches . . . 229
13.18.2.58 axis_a_stop_code . . . 229
13.18.2.59 axis_a_reference_position . . . 229
13.18.2.60 axis_a_motor_position . . . 229
13.18.2.61 axis_a_position_error . . . 229
13.18.2.62 axis_a_aux_position . . . 229
13.18.2.63 axis_a_velocity . . . 229
13.18.2.64 axis_a_torque . . . 230
13.18.2.65 axis_a_analog_in . . . 230
13.18.2.66 axis_a_halls . . . 230
13.18.2.67 axis_a_reserved . . . 230
13.18.2.68 axis_a_variable . . . 230
13.18.2.69 axis_b_status . . . 230
13.18.2.70 axis_b_switches . . . 230
13.18.2.71 axis_b_stop_code . . . 230
13.18.2.72 axis_b_reference_position . . . 230
13.18.2.73 axis_b_motor_position . . . 230
13.18.2.74 axis_b_position_error . . . 231
13.18.2.75 axis_b_aux_position . . . 231
13.18.2.76 axis_b_velocity . . . 231
13.18.2.77 axis_b_torque . . . 231
13.18.2.78 axis_b_analog_in . . . 231
13.18.2.79 axis_b_halls . . . 231
13.18.2.80 axis_b_reserved . . . 231
13.18.2.81 axis_b_variable . . . 231
13.18.2.82 axis_c_status . . . 231
13.18.2.83 axis_c_switches . . . 231
13.18.2.84 axis_c_stop_code . . . 232
13.18.2.85 axis_c_reference_position . . . 232
13.18.2.86 axis_c_motor_position . . . 232
13.18.2.87 axis_c_position_error . . . 232
13.18.2.88 axis_c_aux_position . . . 232
13.18.2.89 axis_c_velocity . . . 232
13.18.2.90 axis_c_torque . . . 232

13.18.2.91 axis_c_analog_in . . . 232
13.18.2.92 axis_c_halls . . . 232
13.18.2.93 axis_c_reserved . . . 232
13.18.2.94 axis_c_variable . . . 233
13.18.2.95 axis_d_status . . . 233
13.18.2.96 axis_d_switches . . . 233
13.18.2.97 axis_d_stop_code . . . 233
13.18.2.98 axis_d_reference_position . . . 233
13.18.2.99 axis_d_motor_position . . . 233
13.18.2.100 axis_d_position_error . . . 233
13.18.2.101 axis_d_aux_position . . . 233
13.18.2.102 axis_d_velocity . . . 233
13.18.2.103 axis_d_torque . . . 233
13.18.2.104 axis_d_analog_in . . . 234
13.18.2.105 axis_d_halls . . . 234
13.18.2.106 axis_d_reserved . . . 234
13.18.2.107 axis_d_variable . . . 234
13.18.2.108 axis_e_status . . . 234
13.18.2.109 axis_e_switches . . . 234
13.18.2.110 axis_e_stop_code . . . 234
13.18.2.111 axis_e_reference_position . . . 234
13.18.2.112 axis_e_motor_position . . . 234
13.18.2.113 axis_e_position_error . . . 234
13.18.2.114 axis_e_aux_position . . . 235
13.18.2.115 axis_e_velocity . . . 235
13.18.2.116 axis_e_torque . . . 235
13.18.2.117 axis_e_analog_in . . . 235
13.18.2.118 axis_e_halls . . . 235
13.18.2.119 axis_e_reserved . . . 235
13.18.2.120 axis_e_variable . . . 235
13.18.2.121 axis_f_status . . . 235
13.18.2.122 axis_f_switches . . . 235
13.18.2.123 axis_f_stop_code . . . 235
13.18.2.124 axis_f_reference_position . . . 236
13.18.2.125 axis_f_motor_position . . . 236
13.18.2.126 axis_f_position_error . . . 236
13.18.2.127 axis_f_aux_position . . . 236
13.18.2.128 axis_f_velocity . . . 236
13.18.2.129 axis_f_torque . . . 236
13.18.2.130 axis_f_analog_in . . . 236
13.18.2.131 axis_f_halls . . . 236
13.18.2.132 axis_f_reserved . . . 236

13.18.2.133 axis_f_variable . . . 236
13.18.2.134 axis_g_status . . . 237
13.18.2.135 axis_g_switches . . . 237
13.18.2.136 axis_g_stop_code . . . 237
13.18.2.137 axis_g_reference_position . . . 237
13.18.2.138 axis_g_motor_position . . . 237
13.18.2.139 axis_g_position_error . . . 237
13.18.2.140 axis_g_aux_position . . . 237
13.18.2.141 axis_g_velocity . . . 237
13.18.2.142 axis_g_torque . . . 237
13.18.2.143 axis_g_analog_in . . . 237
13.18.2.144 axis_g_halls . . . 238
13.18.2.145 axis_g_reserved . . . 238
13.18.2.146 axis_g_variable . . . 238
13.18.2.147 axis_h_status . . . 238
13.18.2.148 axis_h_switches . . . 238
13.18.2.149 axis_h_stop_code . . . 238
13.18.2.150 axis_h_reference_position . . . 238
13.18.2.151 axis_h_motor_position . . . 238
13.18.2.152 axis_h_position_error . . . 238
13.18.2.153 axis_h_aux_position . . . 238
13.18.2.154 axis_h_velocity . . . 239
13.18.2.155 axis_h_torque . . . 239
13.18.2.156 axis_h_analog_in . . . 239
13.18.2.157 axis_h_halls . . . 239
13.18.2.158 axis_h_reserved . . . 239
13.18.2.159 axis_h_variable . . . 239
13.19 Interrupt Struct Reference . . . 239
13.19.1 Detailed Description . . . 240
13.19.2 Member Enumeration Documentation . . . 240
13.19.2.1 Type . . . 240
13.19.3 Constructor & Destructor Documentation . . . 240
13.19.3.1 Interrupt() . . . 240
13.19.4 Member Data Documentation . . . 241
13.19.4.1 type . . . 241
13.19.4.2 axis . . . 241
13.19.4.3 digital_input . . . 241
13.19.4.4 user_interrupt . . . 241
13.19.4.5 [union] . . . 241
13.19.4.6 status . . . 241

14 File Documentation **243**

14.1 examples.md File Reference . . . 243
14.2 license.md File Reference . . . 243
14.3 preprocessor.md File Reference . . . 243
14.4 /builds/software/gclib/doc/README.md File Reference . . . 243
14.5 gclib.h File Reference . . . 243
14.5.1 Detailed Description . . . 249
14.6 gclib.h . . . 249
14.7 gclib_compat.h File Reference . . . 252
14.7.1 Macro Definition Documentation . . . 255
14.7.1.1 GCLIB_DEPRECATED . . . 255
14.7.1.2 G_DR . . . 255
14.7.1.3 G_QR . . . 255
14.7.1.4 G_BOUNDS . . . 255
14.7.1.5 G_CR . . . 255
14.7.1.6 G_COMMA . . . 255
14.7.1.7 G_PUBLISH_SERVER . . . 255
14.7.1.8 G_REMOVE_SERVER . . . 255
14.7.1.9 G_USE_INITIAL_TIMEOUT [1/2] . . . 256
14.7.1.10 G_SMALL_BUFFER . . . 256
14.7.1.11 G_HUGE_BUFFER . . . 256
14.7.1.12 G_LINE_BUFFER . . . 256
14.7.1.13 G_UTIL_TIMEOUT . . . 256
14.7.1.14 G_UTIL_TIMEOUT_OVERRIDE . . . 256
14.7.1.15 G_USE_INITIAL_TIMEOUT [2/2] . . . 256
14.7.1.16 G_UTIL_VERSION . . . 256
14.7.1.17 G_UTIL_INFO . . . 256
14.7.1.18 G_UTIL_SLEEP . . . 257
14.7.1.19 G_UTIL_ADDRESSES . . . 257
14.7.1.20 G_UTIL_IPREQUEST . . . 257
14.7.1.21 G_UTIL_ASSIGN . . . 257
14.7.1.22 G_UTIL_DEVICE_INITIALIZE . . . 257
14.7.1.23 G_UTIL_PING . . . 257
14.7.1.24 G_UTIL_ERROR_CONTEXT . . . 257
14.7.1.25 G_UTIL_GCAPS_HOST . . . 257
14.7.1.26 G_UTIL_GCAPS_VERSION . . . 257
14.7.1.27 G_UTIL_GCAPS_KEEPALIVE . . . 257
14.7.1.28 G_UTIL_GCAPS_ADDRESSES . . . 258
14.7.1.29 G_UTIL_GCAPS_IPREQUEST . . . 258
14.7.1.30 G_UTIL_GCAPS_ASSIGN . . . 258
14.7.1.31 G_UTIL_GCAPS_PING . . . 258
14.7.1.32 G_UTIL_GCAPS_LIST_SERVERS . . . 258
14.7.1.33 G_UTIL_GCAPS_PUBLISH_SERVER . . . 258

14.7.1.34 G_UTIL_GCAPS_SET_SERVER . . . 258
14.7.1.35 G_UTIL_GCAPS_SERVER_STATUS . . . 258
14.7.1.36 G_UTIL_GCAPS_REMOTE_CONNECTIONS . . . 258
14.7.1.37 G_UTIL_GCAPS_SERVER_INFO . . . 258
14.7.1.38 G_UTIL_GCAPS_ADDRESSES_GET_REMEMBERED . . . 259
14.7.1.39 G_UTIL_GCAPS_ADDRESSES_SET_REMEMBERED . . . 259
14.7.2 Typedef Documentation . . . 259
14.7.2.1 GReturn . . . 259
14.7.2.2 GCon . . . 259
14.7.2.3 GSize . . . 259
14.7.2.4 GOption . . . 259
14.7.2.5 GCStringOut . . . 259
14.7.2.6 GCStringIn . . . 259
14.7.2.7 GBufOut . . . 259
14.7.2.8 GBufIn . . . 260
14.7.2.9 GStatus . . . 260
14.7.2.10 GMemory . . . 260
14.8 gclib_compat.h . . . 260
14.9 gclib_errors.h File Reference . . . 261
14.9.1 Detailed Description . . . 263
14.9.2 Macro Definition Documentation . . . 263
14.9.2.1 G_NO_ERROR . . . 263
14.9.2.2 G_NO_ERROR_S . . . 263
14.9.2.3 G_GCLIB_ERROR . . . 263
14.9.2.4 G_GCLIB_ERROR_S . . . 264
14.9.2.5 G_GCLIB_UTILITY_ERROR . . . 264
14.9.2.6 G_GCLIB_UTILITY_ERROR_S . . . 264
14.9.2.7 G_GCLIB_UTILITY_IP_TAKEN . . . 264
14.9.2.8 G_GCLIB_UTILITY_IP_TAKEN_S . . . 264
14.9.2.9 G_GCLIB_NON_BLOCKING_READ_EMPTY . . . 264
14.9.2.10 G_GCLIB_NON_BLOCKING_READ_EMPTY_S . . . 264
14.9.2.11 G_GCLIB_POLLING_FAILED . . . 264
14.9.2.12 G_GCLIB_POLLING_FAILED_S . . . 264
14.9.2.13 G_TIMEOUT . . . 264
14.9.2.14 G_TIMEOUT_S . . . 264
14.9.2.15 G_OPEN_ERROR . . . 265
14.9.2.16 G_OPEN_ERROR_S . . . 265
14.9.2.17 G_ALREADY_OPEN . . . 265
14.9.2.18 G_ALREADY_OPEN_S . . . 265
14.9.2.19 G_READ_ERROR . . . 265
14.9.2.20 G_READ_ERROR_S . . . 265
14.9.2.21 G_WRITE_ERROR . . . 265

14.9.2.22 G_WRITE_ERROR_S . . . 265
14.9.2.23 G_INVALID_PREPROCESSOR_OPTIONS . . . 265
14.9.2.24 G_INVALID_PREPROCESSOR_OPTIONS_S . . . 265
14.9.2.25 G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND . . . 265
14.9.2.26 G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND_S . . . 266
14.9.2.27 G_DATA_RECORD_ERROR . . . 266
14.9.2.28 G_DATA_RECORD_ERROR_S . . . 266
14.9.2.29 G_UNSUPPORTED_FUNCTION . . . 266
14.9.2.30 G_UNSUPPORTED_FUNCTION_S . . . 266
14.9.2.31 G_FIRMWARE_LOAD_NOT_SUPPORTED . . . 266
14.9.2.32 G_FIRMWARE_LOAD_NOT_SUPPORTED_S . . . 266
14.9.2.33 G_ARRAY_NOT_DIMENSIONED . . . 266
14.9.2.34 G_ARRAY_NOT_DIMENSIONED_S . . . 266
14.9.2.35 G_CONNECTION_NOT_ESTABLISHED . . . 266
14.9.2.36 G_CONNECTION_NOT_ESTABLISHED_S . . . 266
14.9.2.37 G_ILLEGAL_DATA_IN_PROGRAM . . . 267
14.9.2.38 G_ILLEGAL_DATA_IN_PROGRAM_S . . . 267
14.9.2.39 G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT . . . 267
14.9.2.40 G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT_S . . . 267
14.9.2.41 G_BAD_RESPONSE_QUESTION_MARK . . . 267
14.9.2.42 G_BAD_RESPONSE_QUESTION_MARK_S . . . 267
14.9.2.43 G_BAD_VALUE_RANGE . . . 267
14.9.2.44 G_BAD_VALUE_RANGE_S . . . 267
14.9.2.45 G_BAD_FULL_MEMORY . . . 267
14.9.2.46 G_BAD_FULL_MEMORY_S . . . 267
14.9.2.47 G_BAD_LOST_DATA . . . 267
14.9.2.48 G_BAD_LOST_DATA_S . . . 268
14.9.2.49 G_BAD_FILE . . . 268
14.9.2.50 G_BAD_FILE_S . . . 268
14.9.2.51 G_BAD_ADDRESS . . . 268
14.9.2.52 G_BAD_ADDRESS_S . . . 268
14.9.2.53 G_BAD_FIRMWARE_LOAD . . . 268
14.9.2.54 G_BAD_FIRMWARE_LOAD_S . . . 268
14.9.2.55 G_GCAPS_OPEN_ERROR . . . 268
14.9.2.56 G_GCAPS_OPEN_ERROR_S . . . 268
14.9.2.57 G_GCAPS_SUBSCRIPTION_ERROR . . . 268
14.9.2.58 G_GCAPS_SUBSCRIPTION_ERROR_S . . . 268
14.10 gclib_errors.h . . . 269
14.11 gclib_record.h File Reference . . . 270
14.11.1 Detailed Description . . . 270
14.11.2 Macro Definition Documentation . . . 271
14.11.2.1 GALILDATARECORDMAXLENGTH . . . 271

14.11.3 Typedef Documentation . . . 271
14.11.3.1 UB . . . 271
14.11.3.2 UW . . . 271
14.11.3.3 SW . . . 271
14.11.3.4 SL . . . 271
14.11.3.5 UL . . . 271
14.12 gclib_record.h . . . 271
14.13 gclibo.h File Reference . . . 284
14.13.1 Macro Definition Documentation . . . 285
14.13.1.1 MALLOCBUF . . . 285
14.13.1.2 MAXPROG . . . 285
14.13.1.3 MAXARRAY . . . 285
14.13.1.4 POLLINGINTERVAL . . . 285
14.13.1.5 G_USE_GCAPS . . . 286
14.14 gclibo.h . . . 286
14.15 gclib.hpp File Reference . . . 287
14.16 gclib.hpp . . . 288

Index **293**

Chapter 1

Galil Communications Library (gclib)

The Galil Communications Library (gclib) is a thread-safe C API that allows users to connect to and communicate with Galil controllers. Galil provides multiple wrappers around gclib, allowing for use in C++, Python, Java, C#, Visual Basic, or LabVIEW projects.

1.1 Installation

All languages require the underlying C library to be installed.

Windows Linux

Note

gcaps is a service that is included in the gclib and gdk installers. It allows multiple gclib clients to share a single controller connection, which can be especially useful for unsolicited data.

1.2 Other Languages

Python Java C# / VB

1.3 Quickstart

Follow the instructions to build a simple example program using gclib.

- **C** C99 or newer is supported.
  - **Windows**
    1. Install CMake.
    2. Install one of the supported compilers:
       - MSVC 19.16 (Visual Studio 2017 version 15.9) or later
       - gcc 8.5 or later
    3. Compile and run the Hello World example:

```
> cmake -S "%GCLIB_ROOT%\examples\c" -B c_examples
> cd c_examples
> cmake --build .
> .\hello-world.exe
gclib version is 2.5.0
Hello World!
```

  - **Linux**
    1. Install CMake and gcc.

RHEL / CentOS Stream / Fedora / Rocky	Ubuntu / Debian / Raspberry Pi OS
`$ sudo dnf install cmake gcc`	`$ sudo apt install cmake gcc`

2. Compile and run the Hello World program:

```
$ cmake -S /usr/share/gclib/examples/c -B c_examples
$ cd c_examples
$ cmake --build .
$ ./hello-world
gclib version is 2.5.0
Hello World!
```

- **C++** C++17 or newer is supported.
  - **Windows**
    1. Install CMake.
    2. Install one of the supported compilers:
       * MSVC 19.16 (Visual Studio 2017 version 15.9) or later
       * g++ 8.5 or later
    3. Compile and run the Hello World example:

```
> cmake -S "%GCLIB_ROOT%\examples\cpp" -B cpp_examples
> cd cpp_examples
> cmake --build .
> .\hello-world.exe
gclib version is 2.5.0
Hello World!
```

  - **Linux**
    1. Install CMake and g++.

RHEL / CentOS Stream / Fedora / Rocky	Ubuntu / Debian / Raspberry Pi OS
`$ sudo dnf install cmake gcc-c++`	`$ sudo apt install cmake g++`

2. Compile and run the Hello World program:

```
$ cmake -S /usr/share/gclib/examples/cpp -B cpp_examples
$ cd cpp_examples
$ cmake --build .
$ ./hello-world
gclib version is 2.5.0
Hello World!
```

- **LabVIEW** The LabVIEW wrapper is available on the VIPM community repository. Simply open VIPM, search for 'gclib', and install.

  LabVIEW 2023 or newer is supported.

  Direct install

  Download

  1. Right click the downloaded file and choose "Add to VIPM Library".
  2. Choose "Add To Library & Install" when prompted.
  3. Search for `gclib` in the package list, then double click on it to bring up the package info.
     - To see all available functions, click "Show in Palettes"
     - To see all example VIs, click "Show Examples"

Looking for older versions?

Next
Examples

Chapter 2

Examples

2.1 Controller Addresses

A core feature of gclib is the ability to connect to a controller over ethernet, serial, or PCI, using a single connection API.

- **C** Use `gclib_addresses()` to see the different addresses that are available.
  - **Code**

```
#include <gclib.h>
#include <stdio.h>

int main(int argc, char* argv[]) {
    char addresses[1024];

    gclib_addresses(addresses, sizeof(addresses));
    printf("Address, Model, Serial\n%s\n", addresses);
}
```

  - **Output**

```
> .\addresses.exe
Address, Model, Serial
192.168.0.40, DMC4040 Rev 1.3i, 10601
COM5, DMC-41x3
GALILPCI0
```

  If using an ethernet controller that doesn't have an IP address yet, it will show up in `gclib_ip_requests()`. Use `gclib_assign_ip()` to give the controller an IP. If successful, the controller will begin showing up in `gclib_addresses()` under the new address.

  - **Code**

```
#include <gclib.h>
#include <stdio.h>

int main(int argc, char* argv[]) {
    char result[1024];

    gclib_ip_requests(result, sizeof(result));
    printf("Address, Model, Serial\n%s\n", result);

    gclib_assign_ip("00:50:4C:20:29:69", "192.168.0.40");
}
```

  - **Output**

```
> .\ip-requests.exe
Address, Model, Serial
00:50:4C:20:29:69, DMC4000, 10601
```

- **C++** Use `gclib::addresses()` to see the different addresses that are available.

- **Code**

```
#include <gclib.hpp>
#include <iostream>

int main(int argc, char* argv[]) {
    std::cout « "Address, Model, Serial\n" « gclib::addresses() « "\n";
}
```

- **Output**

```
> .\addresses.exe
Address, Model, Serial
192.168.0.40, DMC4040 Rev 1.3i, 10601
COM5, DMC-41x3
GALILPCI0
```

If using an ethernet controller that doesn't have an IP address yet, it will show up in `gclib::ip_requests()`. Use `gclib::assign_ip()` to give the controller an IP. If successful, the controller will begin showing up in `gclib::addresses()` under the new address.

- **Code**

```
#include <gclib.hpp>
#include <iostream>

int main(int argc, char* argv[]) {
    std::cout « "Address, Model, Serial" « std::endl « gclib::ip_requests() « std::endl;
    gclib::assign_ip("00:50:4C:20:29:69", "192.168.0.40");
}
```

- **Output**

```
> .\ip-requests.exe
Address, Model, Serial
00:50:4C:20:29:69, DMC4000, 10601
```

- **LabVIEW** Use `addresses` to see the different addresses that are available.

  Addresses.vi

  - **Front Panel**
  - **Block Diagram**

  If using an ethernet controller that doesn't have an IP address yet, it will show up in `ip requests`. Use `assign ip` to give the controller an IP. If successful, the controller will begin showing up in `addresses` under the new address. IP Requests.vi

  - **Front Panel**
  - **Block Diagram**

2.2 Connection Management

- **C** To use a controller with gclib, first pass the address to `gclib_open()` to receive a handle to the connection, which will be used in all gclib calls which operate on that connection. After you are done, close the connection with `gclib_close()`.

  Note

  If you need to change the baud rate, call `gclib_set_baud_rate()` after `gclib_open()`.

  - **Code**

```
#include <gclib.h>
#include <stdio.h>

int main(int argc, char* argv[]) {
    gclib_handle h = 0;
    char address[32], revision[1024];
    uint32_t serial;

    gclib_open(&h, argv[1]);
    gclib_address(h, address, sizeof(address));
    gclib_revision_information(h, revision, sizeof(revision));
    gclib_serial_number(h, &serial);
    printf("Address: %s\nRevision: %s\nSerial: %i\n", address, revision, serial);
    gclib_close(&h);
}
```

- **Output**

```
> .\connection.exe 192.168.0.40
Address: 192.168.0.40
Revision: DMC4040 Rev 1.3i
Serial: 10601
```

Attention

If the program exits before calling `gclib_close()`, the controller may be left in an inconsistent state.

- **C++** To use a controller with gclib, pass the address and an optional baud rate (for serial connections) to `gclib::Controller()` to receive a controller object.

  - **Code**

```
#include <gclib.hpp>
#include <iostream>

int main(int argc, char* argv[]) {
    gclib::Controller dmc(argv[1]);
    std::cout « "Address: " « dmc.address() « "\n";
    std::cout « "Revision: " « dmc.revision_information() « "\n";
    std::cout « "Serial: " « dmc.serial_number() « "\n";
}
```

  - **Output**

```
> .\connection.exe 192.168.0.40
Address: 192.168.0.40
Revision: DMC4040 Rev 1.3i
Serial: 10601
```

- **LabVIEW** To use a controller with gclib, first pass the address to `open` to receive a handle to the connection, which will be used in all gclib calls which operate on that connection. After you are done, close the connection with `close`.

  Connection.vi

  - **Front Panel**

  - **Block Diagram**

Attention

If the program exits before calling `close`, the controller may be left in an inconsistent state.

2.3 Commands

- **C** To issue commands, use `gclib_command()` with an open connection. The following example uses `gclib_command()` to implement a basic terminal.

  - **Code**

```
#include <gclibo.h>
#include <stdio.h>

int main(int argc, char* argv[]) {
    gclib_handle h = NULL;
    char command[80];
    char response[1024];

    gclib_open(&h, argv[1]);
    gclib_revision_information(h, response, sizeof(response));
    printf("Connected to %s.\nPress %s to exit.\n:", response,
        #ifdef WIN32
            "Ctrl+Z then Enter"
        #else
            "Ctrl+D"
        #endif
    );
```

```
    while (fgets(command, sizeof(command), stdin)) {
        gclib_command(h, command, response, sizeof(response));
        printf("%s%s:", response, strlen(response) ? "\n" : "");
    }
    gclib_close(&h);
    printf("\n");
    return 0;
}
```

- **Output**

```
> .\commands.exe 192.168.0.40
Connected to DMC4040 Rev 1.3i.
Press Ctrl+Z then Enter to exit.
:MG "Hello World"
Hello World
:^Z
```

- **C++** To issue commands, use `gclib::Controller::command()` with an open connection. The following example uses `gclib::Controller::command()` to demonstrate using templated return types.

  - **Code**

```
#include <gclib.hpp>
#include <iostream>

int main(int argc, char* argv[]) {
    gclib::Controller dmc(argv[1]);
    std::cout « "Command WH returned " « dmc.command("WH") « "\n";
    std::cout « "Command MG@AN[1] returned " « dmc.command<double>("MG@AN[1]") « "\n";
    std::cout « "Command MG_BN returned " « dmc.command<int>("MG_BN") « "\n";
}
```

  - **Output**

```
> .\commands.exe 192.168.0.40
Command WH returned IHA
Command MG@AN[1] returned 2.583
Command MG_BN returned 10601
```

- **LabVIEW** To issue commands, use `command` with an open connection.

  Commands.vi

  - **Front Panel**

  - **Block Diagram**

2.4 Errors

- **C** All gclib functions provide a `gclib_result` value to indicate the error, with zero (`GCLIB_SUCCESS`) indicating success. Use `gclib_error()` to get a string description of the error.

  - **Code**

```
#include <gclib.h>
#include <stdio.h>

int main(int argc, char* argv[]) {
    gclib_handle h;
    char buf[128];

    gclib_open(&h, argv[1]);

    gclib_result ret = gclib_command(h, "invalid", NULL, 0);
    if (ret == GCLIB_COMMAND_ERROR) {
        printf("Error: %s\n", gclib_error(h));
        gclib_command(h, "TC1", buf, sizeof(buf));
        printf("TC1 returned: %s\n", buf);
    }

    gclib_close(&h);
}
```

- **Output**

```
> .\errors.exe 192.168.0.40
Command "invalid" caused error code 1
TC1 returned: 1 Unrecognized command
```

- **C++** If the underlying C API returns an error, a `gclib::error` exception will be thrown with the error code in `gclib::error::result()`. Use `gclib::error::what()` to get an error string, if available.

  - **Code**

```cpp
#include <gclib.hpp>
#include <iostream>

int main(int argc, char* argv[]) {
    gclib::Controller dmc(argv[1]);
    try {
        dmc.command("invalid");
    } catch (gclib::error& err) {
        std::cout << err.code() << "\n";
        std::cout << "gclib error: " << err.code().value() << "\n";
        std::cout << "error code description: " << err.code().message() << "\n";
        std::cout << "error string: " << err.what() << "\n";
    }
}
```

  - **Output**

```
> .\errors.exe 192.168.0.40
Command "invalid" caused error code 1
TC1 returned: 1 Unrecognized command
```

- **LabVIEW** Standard error chaining is supported by gclib. See `gclib_result` for a full list of error codes. Errors.vi

  - **Front Panel**

  - **Block Diagram**

2.5 Program & Arrays

- **C** Use `gclib_program()` to get the controller's program, and use `gclib_set_program()` to set it.

  - **Code**

```c
#include <gclib.h>
#include <stdio.h>

int main(int argc, char* argv[]) {
    gclib_handle h;
    char program[1024];

    gclib_open(&h, argv[1]);

    gclib_set_program(h,
        "#AUTO\n"
        "MG \"Hello World\"\n"
        "EN", NULL);
    gclib_program(h, program, sizeof(program));
    printf("%s\n", program);

    gclib_close(&h);
}
```

  - **Output**

```
> .\program.exe 192.168.0.40
MG "Hello World"
EN
```

  Use `gclib_array()` to get an array from the controller, and use `gclib_set_array()` to set it. Use the `start` and `end` arguments to transfer only a subset of the array.

- **Code**

```c
#include <gclib.h>
#include <stdio.h>

int main(int argc, char* argv[]) {
    gclib_handle h;
    char array[128];

    gclib_open(&h, argv[1]);
    gclib_command(h, "DM test[5]", NULL, 0);

    gclib_set_array(h, "test", "1,2,3,4,5", 0, 4);
    gclib_array(h, "test", array, sizeof(array), 1, 3);
    printf("%s\n", array);

    gclib_command(h, "DA test[5]", NULL, 0);
    gclib_close(&h);
}
```

- **Output**

```
> .\arrays.exe 192.168.0.40
2.0000, 3.0000, 4.0000
```

- **C++** Use `gclib::Controller::program()` to get the controller's program, and use `gclib::Controller::set_p` to set it.

  - **Code**

```c
#include <gclib.hpp>
#include <iostream>

int main(int argc, char* argv[]) {
    gclib::Controller dmc(argv[1]);

    dmc.set_program(
        "#AUTO\n"
        "MG \"Hello World\"\n"
        "EN");
    std::cout « "" « dmc.program() « "\n";
}
```

  - **Output**

```
> .\program.exe 192.168.0.40
MG "Hello World"
EN
```

  Use `gclib::Controller::array()` to get an array from the controller, and use `gclib::Controller::set_arra` to set it. Use the `start` and `end` arguments to transfer only a subset of the array.

  - **Code**

```c
#include <gclib.hpp>
#include <iostream>

int main(int argc, char* argv[]) {
    gclib::Controller controller(argv[1]);
    controller.command("DM test[5]");

    controller.set_array("test", "1,2,3,4,5", 0, 4);
    std::cout « controller.array("test", 1, 3) « "\n";

    controller.command("DA test[5]");
}
```

  - **Output**

```
> .\arrays.exe 192.168.0.40
2.0000, 3.0000, 4.0000
```

- **LabVIEW** Use `program` to get the controller's program, and use `set program` to set it.

  Program.vi

  - **Front Panel**

  - **Block Diagram**

  Use `array` to get an array from the controller, and use `set array` to set it.

- Use the `first` and `last` inputs to transfer only part of the array.

Arrays.vi

- **Front Panel**

- **Block Diagram**

2.6 Unsolicited Data

The unsolicited data API can be used in blocking mode or callback mode.

Blocking mode is simpler to use, but has tradeoffs. If you want unsolicited data as soon as it arrives, you must block the thread to wait for it, which may not be ideal. If you want to keep the thread running, then you must periodically check for queued unsolicited data, which can add unwanted latency.

Callback mode allows your thread to stay running while enabling immediate response to unsolicited data. When data arrives, your callback function will be invoked on a separate, dedicated thread. Due to this, callback mode can be more complicated to use if you are not familiar with thread synchronization.

- **C**

  - **Blocking** For synchronous usage, subscribe without providing a callback or user data pointer.

    ```
    gclib_subscribe_messages(h, NULL, NULL);
    ```

    Once subscribed, you can then wait a specified amount of time for unsolicited data to arrive.

    ```
    // Time out if message doesn't arrive within one second
    gclib_message(h, message_buf, sizeof(message_buf), 1000);
    ```

    - **Code**

      ```
      #include <gclib.h>
      #include <stdio.h>

      int main(int argc, char* argv[]) {
          gclib_handle h;
          char buf[32];

          gclib_open(&h, argv[1]);

          gclib_subscribe_messages(h, NULL, NULL);
          gclib_subscribe_interrupts(h, NULL, NULL);
          gclib_subscribe_data_records(h, NULL, NULL);

          gclib_set_interrupts(h, GCLIB_PROGRAM_STOPPED, 0, 0);
          gclib_set_data_records(h, 8);

          gclib_set_program(h, "MG \"Hello World\"; EN", NULL);
          gclib_command(h, "XQ", NULL, 0);

          gclib_message(h, buf, sizeof(buf), -1);
          printf("Got message: %s\n", buf);

          gclib_interrupt_t interrupt;
          gclib_interrupt(h, &interrupt, -1);
          printf("Got interrupt status byte: %i\n", interrupt.status);

          gclib_data_record_handle data_record;
          gclib_data_record(h, &data_record, -1);
          printf("Got data record, sample %i\n", gclib_data_record_sample(data_record));

          gclib_set_data_records(h, 0);
          gclib_close(&h);
      }
      ```

    - **Output**

      ```
      > .\unsolicited.exe 192.168.0.40
      Got message Hello World
      Got data record, sample 39306
      Got interrupt 240
      ```

- **Callback** For asynchronous usage, subscribe with a callback function and an arbitrary data pointer (which can be `NULL`). Each time unsolicited data arrives, your callback will be invoked and the pointer will be passed to it.

```
void handle_message(void*, const char* message) {
    printf("Message received: %s", message);
};
gclib_subscribe_messages(h, handle_message, NULL);
```

Attention

The callback will be invoked on a separate thread. Be sure to protect any shared data!

  - **Code**

```
#include <gclib.h>
#include <stdio.h>
#include "sleep.h"

void handle_message(void* user_data, const char* message) {
    printf("Got message: %s\n", message);
}
void handle_interrupt(void* user_data, gclib_interrupt_t interrupt) {
    printf("Got interrupt status byte: %i\n", interrupt.status);
}
void handle_data_record(void* user_data, gclib_data_record_handle data_record) {
    printf("Got data record, sample %i\n", gclib_data_record_sample(data_record));
}

int main(int argc, char* argv[]) {
    gclib_handle h;
    gclib_open(&h, argv[1]);

    gclib_subscribe_messages(h, handle_message, NULL);
    gclib_subscribe_interrupts(h, handle_interrupt, NULL);
    gclib_subscribe_data_records(h, handle_data_record, NULL);

    gclib_set_interrupts(h, GCLIB_PROGRAM_STOPPED, 0, 0);
    gclib_set_data_records(h, 200);

    char buf[16];
    gclib_set_program(h, "WT 100; MG \"Hello World\"; EN", NULL);
    gclib_command(h, "XQ", NULL, 0);

    msleep(300);
    gclib_close(&h);
}
```

  - **Output**

```
> .\callback.exe 192.168.0.40
Got message Hello World
Got data record, sample 39306
Got interrupt 240
```

For a full list of data record fields, see the Data Record API.

Note

Controllers with default settings will not generate interrupts or data records. Use `gclib_set_interrupts()` and `gclib_set_data_records()` to configure your controller if needed.

- **C++**

  - **Blocking** For synchronous usage, subscribe without providing a callback or user data pointer.

```
dmc.subscribe_messages();
```

Once subscribed, you can then wait a specified amount of time for unsolicited data to arrive.

```
// Time out if message doesn't arrive within one second
dmc.message(1000);
```

  - **Code**

```
#include <gclib.hpp>
#include <iostream>

int main(int argc, char* argv[]) {
    gclib::Controller dmc(argv[1]);

    dmc.subscribe_messages();
    dmc.subscribe_interrupts();
    dmc.subscribe_data_records();

    dmc.set_interrupts(gclib::Interrupt::ProgramStopped);
```

```
    dmc.set_data_records(8);

    dmc.set_program("MG \"Hello World\"; EN");
    dmc.command("XQ");

    std::cout « "Got message: " « dmc.message() « "\n";
    std::cout « "Got interrupt type: " « static_cast<unsigned int>(dmc.interrupt().type) «
"\n";
    std::cout « "Got data record, sample " « dmc.data_record().sample() « "\n";

    dmc.set_interrupts(gclib::Interrupt::NoInterrupts);
    dmc.set_data_records(0);
}
```

* **Output**

```
> .\unsolicited.exe 192.168.0.40
Got message Hello World
Got data record, sample 39306
Got interrupt 240
```

– **Callback** For asynchronous usage, subscribe with a callback function. Each time unsolicited data arrives, your callback will be invoked.

```
dmc.subscribe_messages([](std::string message) {
    std::cout « "Message received: " « message;
});
```

Attention

The callback will be invoked on a separate thread. Be sure to protect any shared data!

* **Code**

```
#include <gclib.hpp>
#include <iostream>
#include <chrono>
#include <thread>

int main(int argc, char* argv[]) {
    gclib::Controller dmc(argv[1]);

    dmc.subscribe_messages([](std::string message) {
        std::cout « "Got message: " « message « "\n";
    });
    dmc.subscribe_interrupts([](gclib::Interrupt interrupt) {
        std::cout « "Got interrupt status byte: " « static_cast<unsigned
int>(interrupt.status) « "\n";
    });
    dmc.subscribe_data_records([](gclib::DataRecord data_record) {
        std::cout « "Got data record, sample " « data_record.sample() « "\n";
    });

    dmc.set_interrupts(gclib::Interrupt::ProgramStopped);
    dmc.set_data_records(200);

    dmc.set_program("WT 100; MG \"Hello World\"; EN");
    dmc.command("XQ");

    std::this_thread::sleep_for(std::chrono::milliseconds(300));
}
```

* **Output**

```
> .\callback.exe 192.168.0.40
Got message Hello World
Got data record, sample 39306
Got interrupt 240
```

For a full list of data record fields, see `gclib::DataRecord`.

Note

Controllers with default settings will not generate interrupts or data records. Use `gclib::Controller::set_inter` and `gclib::Controller::set_data_records()` to configure your controller if needed.

- **LabVIEW** To receive unsolicited data, you must first subscribe to the desired unsolicited data using `subscribe messages`, `subscribe interrupts`, or `subscribe data records`.

  Once subscribed, use `message`, `interrupt`, and `data record` alongside an optional timeout to wait for unsolicited data.

  Unsolicited.vi

  – **Front Panel**

- **Block Diagram**

2.7 Galil Connect

Galil Connect allows gclib to issue commands through a remote gcaps server. This makes debugging closed or distant systems much easier.

- **C** On the device hosting the remote gcaps server, use `gclib_set_published()`.
  - **Code**

```c
#include <gclib.h>
#include <stdio.h>
#include <stdbool.h>

int main(int argc, char* argv[]) {
    gclib_set_published(argv[1]);
    printf("Published remote gcaps server \"%s\"\n", argv[1]);
}
```

  - **Output**

```
> .\server.exe pi
Published remote gcaps server "pi"
```

  On the client, use `gclib_list_servers()` to view all available gcaps servers. Pass a server name to `gclib_set_server()` for future gclib calls to be routed through that gcaps server. When done, pass `NULL` to `gclib_set_server()` to disconnect from the remote gcaps server.

  - **Code**

```c
#include <gclib.h>
#include <stdio.h>

int main(int argc, char* argv[]) {
    char buf[1024];
    gclib_handle h;

    gclib_list_servers(buf, sizeof(buf));
    printf("Available servers:\n%s\n", buf);
    gclib_set_server(argv[1]);

    gclib_addresses(buf, sizeof(buf));
    printf("\nAddresses reported by %s:\n%s", argv[1], buf);
    gclib_open(&h, argv[2]);
    gclib_revision_information(h, buf, sizeof(buf));
    printf("\nConnected to %s\n", buf);
    gclib_close(&h);

    gclib_set_server(NULL);
}
```

  - **Output**

```
> .\client.exe pi COM5
Available servers:
pi

Addresses reported by pi:
COM5

Connected to COM5, DMC31010 Rev 1.4f, 12345
```

- **C++** On the device hosting the remote gcaps server, use `gclib::set_published()`.
  - **Code**

```cpp
#include <gclib.hpp>
#include <iostream>

int main(int argc, char* argv[]) {
    gclib::set_published(argv[1]);
    std::cout « "Published remote gcaps server: " « argv[1] « "\n";
}
```

- **Output**

```
> .\server.exe pi
Published remote gcaps server "pi"
```

On the client, use `gclib::list_servers()` to view all available gcaps servers. Pass a server name to `gclib::set_server()` for future gclib calls to be routed through that gcaps server. When done, call `gclib::set_server()` with no arguments to disconnect from the remote gcaps server.

- **Code**

```
#include <gclib.hpp>
#include <iostream>

int main(int argc, char* argv[]) {
    std::cout « "Available servers:\n" « gclib::list_servers() « "\n";

    gclib::set_server(argv[1]);
    std::cout « "\nAddresses reported by " « argv[1] « ":\n" « gclib::addresses() « "\n";

    std::cout « "\nConnected to " « gclib::Controller(argv[2]).revision_information() « "\n";
    gclib::set_server();
}
```

- **Output**

```
> .\client.exe pi COM5
Available servers:
pi

Addresses reported by pi:
COM5

Connected to COM5, DMC31010 Rev 1.4f, 12345
```

- **LabVIEW** On the device hosting the remote gcaps server (in this example a Raspberry Pi), use `set published`.

  Server.vi

  - **Front Panel**

  - **Block Diagram**

  On the client, use `list servers` to view all available gcaps servers. Pass a server name to `set server` for future gclib calls to be routed through that gcaps server. When done, pass an empty string to `set server` to disconnect from the remote gcaps server.

  Client.vi

  - **Front Panel**

  - **Block Diagram**

2.8 Example Project: Record and Replay

This project contains two example programs.

- **C** The 'Record' example uses `RA` in continuous mode along with `gclib_array()` to allow recording movement for an arbitrary amount of time. It produces a file with the recorded positions of Axis A.

  - **Code**

```
#include <gclib.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
```

```
#include "sleep.h"

void check(gclib_result ret, gclib_handle h) {
    char error[128];
    if (ret == GCLIB_SUCCESS) {
        return;
    } else if (ret == GCLIB_COMMAND_ERROR) {
        gclib_command(h, "TC1", error, sizeof(error));
        printf("%s\n", error);
    }

    fprintf(stderr, "Error %i: %s\n", ret, gclib_error(h));
    gclib_close(&h);
    exit(ret);
}

int main(int argc, char* argv[]) {
    if (argc != 3) {
        fprintf(stderr, "Usage: record.exe ADDRESS SECONDS\n");
        return 1;
    }

    char* buf = (char*)malloc(16384);

    gclib_result ret = GCLIB_SUCCESS;
    union GDataRecord record;
    size_t bytesReturned;

    gclib_handle h = NULL;
    if (gclib_open(&h, argv[1]) != GCLIB_SUCCESS) {
        fprintf(stderr, "Failed to open connection to controller");
        exit(1);
    }

    char arrayName[5];
    sprintf(arrayName, "posA");

    gclib_command(h, "MO", NULL, 0);
    gclib_command(h, "DM posA[1000]", NULL, 0);
    gclib_command(h, "RA posA[]", NULL, 0);
    gclib_command(h, "RD _TPA", NULL, 0);
    gclib_command(h, "RC 1,-1000", NULL, 0);

    size_t start = 0, end;
    char* positions = (char*)malloc(16384);
    char* token;
    int count;
    time_t startTime = time(NULL);
    FILE* file = fopen("positions.txt", "w");
    while (difftime(time(NULL), startTime) < atof(argv[2])) {
        msleep(500);
        gclib_command(h, "MG_RD", buf, 16384);
        end = atoi(buf);
        check(gclib_array(h, arrayName, positions, 16384, start, end < start ? 0 : end), h);
        printf("%zu\n", strlen(positions));
        if (end < start) {
            check(gclib_array(h, arrayName, buf, 16384, 0, end), h);
            printf("%zu\n", strlen(buf));
            strcat(positions, ", ");
            strcat(positions, buf);
        }
        count = end - start;
        if (end < start)
            count += 1000;
        for (int i = 0; i < count; i++) {
            token = strtok( i == 0 ? positions : NULL, " ,");
            if (token == NULL)
                break;
            fprintf(file, "%s", token);
            fputc('\n', file);
        }
        start = end;
    }
    fclose(file);

    gclib_command(h, "RC 0", NULL, 0);
    gclib_close(&h);
}
```

– **Output**

```
> .\record.exe 192.168.0.40 5
```

The 'Replay' example uses the file produced by 'Record' along with CM to accurately reproduce the recorded movement. Note that axis A must be properly set up for motion.

– **Code**

```c
#include <gclib.h>
#include <stdio.h>
#include <stdlib.h>
#include "sleep.h"

void check(gclib_result ret, gclib_handle h) {
    char error[128];
    if (ret == GCLIB_SUCCESS)
        return;
    if (ret == GCLIB_COMMAND_ERROR) {
        gclib_command(h, "TC1", error, sizeof(error));
        printf("%s\n", error);
    }
    fprintf(stderr, "Error %i: %s\n", ret, gclib_error(h));
    gclib_close(&h);
    exit(ret);
}

int main(int argc, char* argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Usage: replay.exe ADDRESS\n");
        return 0;
    }

    char* buf = (char*)malloc(16384);

    GCon h = NULL;
    if (gclib_open(&h, argv[1]) != GCLIB_SUCCESS) {
        fprintf(stderr, "Failed to open connection to controller");
        exit(1);
    }

    gclib_command(h, "MG_CM", NULL, 0);
    int contourSpace = atof(buf);

    gclib_command(h, "SH A", NULL, 0);
    gclib_command(h, "CM A", NULL, 0);
    gclib_command(h, "DT -1", NULL, 0);

    FILE* f = fopen("positions.txt", "r");
    if (f == NULL) {
        fprintf(stderr, "Failed to open positions.txt");
        return 1;
    }

    int lastPosition;
    int position;
    fgets(buf, 16384, f);
    lastPosition = atof(buf);
    while (fgets(buf, 16384, f)) {
        position = atof(buf);
        sprintf(buf, "CD %i", position - lastPosition);
        lastPosition = position;
        if (gclib_command(h, buf, NULL, 0) == GCLIB_COMMAND_ERROR) {
            check(gclib_command(h, "MG_TC", buf, 16384), h);
            if ((int)atof(buf) == 32) {
                gclib_command(h, "MG_DT", buf, 16384);
                if ((int)atof(buf) == -1)
                    gclib_command(h, "DT 1", NULL, 0);
                msleep(100);
            }
        }
    }

    do {
        gclib_command(h, "MG_CM", buf, 0);
    } while ((int)atof(buf) < contourSpace);
    fclose(f);

    gclib_command(h, "CD 0=0", NULL, 0);
    gclib_command(h, "MO", NULL, 0);
    gclib_close(&h);
}
```

– Output

```
> .\replay.exe 192.168.0.40
```

- **C++** The 'Record' example uses RA in continuous mode along with `gclib::Controller::array()` to allow recording movement for an arbitrary amount of time. It produces a file with the recorded positions of Axis A.

– Code

```cpp
#include <gclib.hpp>
#include <iostream>
```

```
#include <thread>
#include <chrono>
#include <algorithm>
#include <fstream>

int main(int argc, char* argv[]) {
    if (argc != 3) {
        std::cerr « "Usage: record.exe ADDRESS SECONDS\n";
        return 1;
    }

    gclib::Controller dmc(argv[1]);
    try {
        auto file = std::ofstream("positions.txt", std::ios::trunc);
        if (!file) {
            std::cerr « "Failed to open positions.txt";
            return 1;
        }

        dmc.command("MO");
        dmc.command("DM posA[1000]");
        dmc.command("RA posA[]");
        dmc.command("RD _TPA");
        dmc.command("RC 1,-1000");

        size_t start = 0, end;
        std::string token;
        auto start_time = std::chrono::steady_clock::now();
        while (std::chrono::duration_cast<std::chrono::seconds>(std::chrono::steady_clock::now()
- start_time).count() < std::atoi(argv[2])) {
            std::this_thread::sleep_for(std::chrono::milliseconds{500});
            end = std::stod(dmc.command("MG_RD"));
            std::string positions = dmc.array("posA", start, end < start ? 0 : end);
            if (end < start) {
                positions += ", " + dmc.array("posA", 0, end);
            }
            std::replace(positions.begin(), positions.end(), ',', '\n');
            file « positions « " \n";
            start = end;
        }
    } catch (gclib::error& err) {
        std::cout « "Error " « err.code().value() « ": " « err.what();
        exit(err.code().value());
    }
    dmc.command("RC 0");
}
```

– Output

```
> .\record.exe 192.168.0.40 5
```

The 'Replay' example uses the file produced by 'Record' along with CM to accurately reproduce the recorded movement. Note that axis A must be properly set up for motion.

– Code

```
#include <gclib.hpp>
#include <iostream>
#include <chrono>
#include <thread>
#include <fstream>
#include <string>

int main(int argc, char* argv[]) {
    if (argc != 2) {
        std::cerr « "Usage: replay.exe ADDRESS\n";
        return 0;
    }

    try {
        gclib::Controller dmc(argv[1]);

        int contourSpace = std::stod(dmc.command("MG_CM"));

        dmc.command("SH A");
        dmc.command("CM A");
        dmc.command("DT -1");

        std::ifstream file("positions.txt");
        if (!file) {
            std::cerr « "Failed to open positions.txt";
            return 1;
        }

        int lastPosition;
        int position;
        std::string line;
```

```
            std::getline(file, line);
            lastPosition = std::stod(line);
            while (std::getline(file, line)) {
                position = std::stod(line);
                try {
                    dmc.command("CD " + std::to_string(position - lastPosition));
                } catch (gclib::error& err) {
                    if (err.code().value() != gclib::CommandError) {
                        throw;
                    }
                    if (dmc.command<int>("MG_TC") == 32) {
                        if (dmc.command<int>("MG_DT") == -1) {
                            dmc.command("DT 1");
                        }
                        std::this_thread::sleep_for(std::chrono::milliseconds(100));
                    }
                }
                lastPosition = position;
            }

            while (dmc.command<int>("MG_CM") < contourSpace) {
                std::this_thread::sleep_for(std::chrono::milliseconds(100));
            }
            file.close();

            dmc.command("CD 0=0");
            dmc.command("MO");
        } catch (gclib::error& err) {
            std::cout << "Error " << (int)err.code().value() << ": " << err.what();
            exit(err.code().value());
        }
    }
```

– Output

```
> .\replay.exe 192.168.0.40
```

Chapter 3

License

- gclib is licensed under the MIT-0 License.
- gclib and gcaps use OpenSSL, which is licensed under the Apache 2.0 License.
  - The Java wrapper uses JNA, which is also licensed under the Apache 2.0 License.
- This documentation uses the doxygen-awesome-css theme, which is covered by the MIT License.

Chapter 4

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.

4.1 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.

4.2 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
```

4.2.1 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.

4.2.2 Double Hash

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

4.2.3 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.

4.2.4 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.

4.2.5 Preprocessor Directives

Note

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

4.2.5.1 ##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.

4.2.5.2 ##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 initilization 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.

4.2.5.3 ##gclib

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

4.2.6 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*.

4.3 Preprocessor Options

4.3.1 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`.

4.3.1.1 Compression Levels, *n*

- Level 0 (mandatory)
  1. Remove lines beginning with `REM`.
  2. Remove trailing semicolons.
     (a) Comment blank lines with '.
     (b) Remove white space (space/tab) in front of # (label declarations).
     (c) Remove white space after commands.
     (d) Line ends changed to carriage return.
     (e) Replace leading tabs with double space.
     (f) Replace non-leading tabs with single space.
  3. 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.

4.3.2 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.

4.3.3 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.

4.3.3.1 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
```

4.3.4 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.

4.3.5 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
```

4.4 GDK Support

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

Chapter 5

Deprecated List

Member GAddresses (GCStringOut addresses, GSize addresses_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_addresses()` instead.

Member GArrayDownload (GCon g, const GCStringIn array_name, GOption first, GOption last, GCStringIn buffer)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_set_array()` instead.

Member GArrayDownloadFile (GCon g, GCStringIn file_path)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GArrayUpload (GCon g, const GCStringIn array_name, GOption first, GOption last, GOption delim, GBufOut buffer, GSize buffer_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_array()` instead.

Member GArrayUploadFile (GCon g, GCStringIn file_path, GCStringIn names)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GAssign (GCStringIn ip, GCStringIn mac)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_assign_ip()` instead.

Member GClose (GCon g)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_close()` instead.

Member GCmd (GCon g, GCStringIn command)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GCmdD (GCon g, GCStringIn command, double *value)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GCmdI (GCon g, GCStringIn command, int *value)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GCmdT (GCon g, GCStringIn command, GCStringOut trimmed_response, GSize response_len, GCStringOut *front)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GCommand (GCon g, GCStringIn command, GBufOut buffer, GSize buffer_len, GSize *bytes_returned)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_command()` instead.

Member GError (GReturn rc, GCStringOut error, GSize error_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GFirmwareDownload (GCon g, GCStringIn filepath)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_set_firmware()` instead.

Member GInfo (GCon g, GCStringOut info, GSize info_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_address()`, `gclib_revision_information()`, and `gclib_serial_number()` instead.

Member GInterrupt (GCon g, GStatus *status_byte)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_interrupt()` instead.

Member GIpRequests (GCStringOut requests, GSize requests_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_ip_requests()` instead.

Member GListServers (GCStringOut servers, GSize servers_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_list_servers()` instead.

Member GMessage (GCon g, GCStringOut buffer, GSize buffer_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_message()` instead.

Member GMotionComplete (GCon g, GCStringIn axes)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GOpen (GCStringIn connection_string, GCon *g)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_open()` instead.

Member GProgramDownload (GCon g, GCStringIn program, GCStringIn preprocessor)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_set_program()` instead.

Member GProgramDownloadFile (GCon g, GCStringIn file_path, GCStringIn preprocessor)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GProgramUpload (GCon g, GBufOut buffer, GSize buffer_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_program()` instead.

Member GProgramUploadFile (GCon g, GCStringIn file_path)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GPublishServer (GCStringIn name, GOption publish, GOption save)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_set_published()` instead.

Member GRead (GCon g, GBufOut buffer, GSize buffer_len, GSize *bytes_read)

This function only exists for ABI compatibility, and will be removed in a future gclib version.

Member GRecord (GCon g, union GDataRecord *record, GOption method)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_data_record()` instead.

Member GRecordRate (GCon g, double period_ms)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_set_data_records()` instead.

Member GRemoteConnections (GCStringOut connections, GSize connections_length)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GServerStatus (GCStringOut status, GSize status_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_server()` instead.

Member GSetServer (GCStringIn server_name)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_set_server()` instead.

Member GSetupDownloadFile (GCon g, GCStringIn file_path, GOption options, GCStringOut info, GSize info_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GSleep (unsigned int timeout_ms)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GTimeout (GCon g, short timeout_ms)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GUtility (GCon g, GOption request, GMemory memory1, GMemory memory2)

This function only exists for ABI compatibility, and will be removed in a future gclib version.

Member GVersion (GCStringOut ver, GSize ver_len)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_version()` and `gclib_gcaps_version()` instead.

Member GWaitForBool (GCon g, GCStringIn predicate, int trials)

This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Member GWrite (GCon g, GBufIn buffer, GSize buffer_len)

This function only exists for ABI compatibility, and will be removed in a future gclib version.

Chapter 6

Topic Index

6.1 Topics

Here is a list of all topics with brief descriptions:

C . . . 41
- Connection . . . 45
  - Galil Connect . . . 50
- Controller . . . 52
  - Data Record . . . 54
    - Amp . . . 58
    - Axis . . . 59
    - Coordinated Move . . . 63
  - Memory . . . 65
  - Unsolicited Data . . . 67

C (Legacy) . . . 73
- Connection . . . 75
- Controller . . . 80
  - Communication . . . 82
  - Memory . . . 85
  - Unsolicited Data . . . 90
- Galil Connect . . . 93

Chapter 7

Namespace Index

7.1 Namespace List

Here is a list of all namespaces with brief descriptions:

gclib 97
gclib::internal 100

Chapter 8

Hierarchical Index

8.1 Class Hierarchy

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

DataRecord::Amp . . . 101
DataRecord::Axis . . . 102
Controller . . . 106
DataRecord::CoordinatedMove . . . 112
DataRecord . . . 113
gclib_interrupt_t . . . 117
GDataRecord . . . 118
GDataRecord1802 . . . 120
GDataRecord1806 . . . 131
GDataRecord2103 . . . 153
GDataRecord30000 . . . 169
GDataRecord4000 . . . 174
GDataRecord47000_ENC . . . 196
GDataRecord47162 . . . 201
GDataRecord47300_24EX . . . 207
GDataRecord47300_ENC . . . 212
GDataRecord52000 . . . 217
Interrupt . . . 239
system_error
error . . . 117

Chapter 9

Class Index

9.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

DataRecord::Amp
: Holds amplifier-specific data record fields. Provided by `DataRecord.amp()` 101

DataRecord::Axis
: Holds axis-specific data record fields. Provided by `DataRecord.axis()` 102

Controller
: Manages a connection to a controller 106

DataRecord::CoordinatedMove
: Holds coordinated move specific data record fields. Provided by `DataRecord.coordinated_move()` 112

DataRecord 113

error
: Thrown when the underlying C library returns a nonzero `gclib_result` 117

gclib_interrupt_t
: An interrupt generated by the controller 117

GDataRecord
: Data record union, containing all structs and a generic byte array accessor 118

GDataRecord1802 120

GDataRecord1806
: Data record struct for DMC-1806 controller 131

GDataRecord2103
: Data record struct for DMC-2103 controllers 153

GDataRecord30000
: Data record struct for DMC-30010 controllers 169

GDataRecord4000
: Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0 174

GDataRecord47000_ENC
: Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields 196

GDataRecord47162
: Data record struct for RIO-47162 201

GDataRecord47300_24EX
: Data record struct for RIO-47300 with 24EX I/O daughter board 207

GDataRecord47300_ENC
: Data record struct for RIO-47300. Includes encoder fields 212

GDataRecord52000
: Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40 217

Interrupt 239

Chapter 10

File Index

10.1 File List

Here is a list of all files with brief descriptions:

gclib.h 243
gclib_compat.h 252
gclib_errors.h 261
gclib_record.h 270
gclibo.h 284
gclib.hpp 287

Chapter 11

Topic Documentation

11.1 C

Topics

- Connection
- Controller

Classes

- struct gclib_interrupt_t

  An interrupt generated by the controller.

Typedefs

- typedef struct gclib_context * gclib_handle

  A handle to an open connection given by `gclib_open()` for use in future API calls.
- typedef void * gclib_data_record_handle

  A handle to a data record, for use in Data Record API calls.
- typedef enum gclib_result gclib_result
- typedef enum gclib_interrupt_type gclib_interrupt_type
- typedef struct gclib_interrupt_t gclib_interrupt_t

  An interrupt generated by the controller.
- typedef enum gclib_digital_input_flags gclib_digital_input_flags

  Flags to enable or disable `GCLIB_DIGITAL_INPUT_LOW` interrupts per digital input, for use in `gclib_set_interrupts()`.
- typedef enum gclib_axis_flags gclib_axis_flags

  Flags for building an axis mask, for example in `gclib_set_interrupts()`.

Enumerations

- enum gclib_result {
  GCLIB_SUCCESS ,
  GCLIB_INVALID_ARGUMENT ,
  GCLIB_INTERNAL_ERROR ,
  GCLIB_TIMEOUT ,
  GCLIB_COMMAND_ERROR ,
  GCLIB_BUFFER_TOO_SMALL ,
  GCLIB_NOT_CONNECTED ,
  GCLIB_NOT_SUBSCRIBED }
- enum gclib_interrupt_type {
  GCLIB_NO_INTERRUPTS = 0 ,
  GCLIB_USER_INTERRUPT = 0 ,
  GCLIB_MOTION_COMPLETE = 1 << 0 ,

GCLIB_ALL_AXES_MOTION_COMPLETE = 1 << 8 ,
GCLIB_EXCESS_POSITION_ERROR = 1 << 9 ,
GCLIB_LIMIT_SWITCH = 1 << 10 ,
GCLIB_WATCHDOG_TIMER = 1 << 11 ,
GCLIB_PROGRAM_STOPPED = 1 << 13 ,
GCLIB_COMMAND_DONE = 1 << 14 ,
GCLIB_DIGITAL_INPUT_LOW = 1 << 15 ,
GCLIB_ALL_INTERRUPTS = (1 << 16) - 1 }

- enum gclib_digital_input_flags {
  GCLIB_NO_DIGITAL_INPUTS = 0 ,
  GCLIB_DIGITAL_INPUT_1 = 1 << 0 ,
  GCLIB_DIGITAL_INPUT_2 = 1 << 1 ,
  GCLIB_DIGITAL_INPUT_3 = 1 << 2 ,
  GCLIB_DIGITAL_INPUT_4 = 1 << 3 ,
  GCLIB_DIGITAL_INPUT_5 = 1 << 4 ,
  GCLIB_DIGITAL_INPUT_6 = 1 << 5 ,
  GCLIB_DIGITAL_INPUT_7 = 1 << 6 ,
  GCLIB_DIGITAL_INPUT_8 = 1 << 7 ,
  GCLIB_ALL_DIGITAL_INPUTS = (1 << 8) - 1 }

  Flags to enable or disable `GCLIB_DIGITAL_INPUT_LOW` interrupts per digital input, for use in `gclib_set_interrupts()`.
- enum gclib_axis_flags {
  GCLIB_NO_AXES = 0 ,
  GCLIB_AXIS_A = 1 << 0 ,
  GCLIB_AXIS_B = 1 << 1 ,
  GCLIB_AXIS_C = 1 << 2 ,
  GCLIB_AXIS_D = 1 << 3 ,
  GCLIB_AXIS_E = 1 << 4 ,
  GCLIB_AXIS_F = 1 << 5 ,
  GCLIB_AXIS_G = 1 << 6 ,
  GCLIB_AXIS_H = 1 << 7 ,
  GCLIB_ALL_AXES = (1 << 8) - 1 }

  Flags for building an axis mask, for example in `gclib_set_interrupts()`.

Functions

- gclib_result gclib_version (char *version, size_t len)

  Get library version.
- gclib_result gclib_gcaps_version (char *gcaps_version, size_t len)

  Get library version used by current gcaps server.

11.1.1 Detailed Description

11.1.2 Typedef Documentation

11.1.2.1 gclib_handle

```
typedef struct gclib_context* gclib_handle
```

A handle to an open connection given by `gclib_open()` for use in future API calls.
Definition at line 23 of file gclib.h.

11.1.2.2 gclib_data_record_handle

```
typedef void* gclib_data_record_handle
```

A handle to a data record, for use in Data Record API calls.
Definition at line 25 of file gclib.h.

11.1.2.3 gclib_result

typedef enum gclib_result gclib_result
All gclib function return values. See gclib_error() for a more detailed error string.

11.1.2.4 gclib_interrupt_type

typedef enum gclib_interrupt_type gclib_interrupt_type
All interrupt status types. Used in gclib_set_interrupts() to configure which interrupts should be generated by the controller, and in gclib_interrupt_t once an interrupt has been read from the controller.

11.1.2.5 gclib_interrupt_t

typedef struct gclib_interrupt_t gclib_interrupt_t
An interrupt generated by the controller.

11.1.2.6 gclib_digital_input_flags

typedef enum gclib_digital_input_flags gclib_digital_input_flags
Flags to enable or disable GCLIB_DIGITAL_INPUT_LOW interrupts per digital input, for use in gclib_set_interrupts().

11.1.2.7 gclib_axis_flags

typedef enum gclib_axis_flags gclib_axis_flags
Flags for building an axis mask, for example in gclib_set_interrupts().

11.1.3 Enumeration Type Documentation

11.1.3.1 gclib_result

enum gclib_result
All gclib function return values. See gclib_error() for a more detailed error string.

Enumerator

GCLIB_SUCCESS	
GCLIB_INVALID_ARGUMENT	
GCLIB_INTERNAL_ERROR	
GCLIB_TIMEOUT	
GCLIB_COMMAND_ERROR	
GCLIB_BUFFER_TOO_SMALL	
GCLIB_NOT_CONNECTED	
GCLIB_NOT_SUBSCRIBED	

Definition at line 29 of file gclib.h.

11.1.3.2 gclib_interrupt_type

enum gclib_interrupt_type
All interrupt status types. Used in gclib_set_interrupts() to configure which interrupts should be generated by the controller, and in gclib_interrupt_t once an interrupt has been read from the controller.

Enumerator

GCLIB_NO_INTERRUPTS	
GCLIB_USER_INTERRUPT	
GCLIB_MOTION_COMPLETE	

Enumerator

GCLIB_ALL_AXES_MOTION_COMPLETE	
GCLIB_EXCESS_POSITION_ERROR	Must be reenabled with `gclib_set_interrupts()` after occurrence.
GCLIB_LIMIT_SWITCH	Must be reenabled with `gclib_set_interrupts()` after each occurrence.
GCLIB_WATCHDOG_TIMER	
GCLIB_PROGRAM_STOPPED	If multiple threads are running, this interrupt is only triggered when all threads are finished.
GCLIB_COMMAND_DONE	
GCLIB_DIGITAL_INPUT_LOW	Must be reenabled with `gclib_set_interrupts()` after each occurrence.
GCLIB_ALL_INTERRUPTS	

Definition at line 41 of file gclib.h.

11.1.3.3 gclib_digital_input_flags

`enum gclib_digital_input_flags`
Flags to enable or disable `GCLIB_DIGITAL_INPUT_LOW` interrupts per digital input, for use in `gclib_set_interrupts()`.

Enumerator

GCLIB_NO_DIGITAL_INPUTS	
GCLIB_DIGITAL_INPUT_1	
GCLIB_DIGITAL_INPUT_2	
GCLIB_DIGITAL_INPUT_3	
GCLIB_DIGITAL_INPUT_4	
GCLIB_DIGITAL_INPUT_5	
GCLIB_DIGITAL_INPUT_6	
GCLIB_DIGITAL_INPUT_7	
GCLIB_DIGITAL_INPUT_8	
GCLIB_ALL_DIGITAL_INPUTS	

Definition at line 65 of file gclib.h.

11.1.3.4 gclib_axis_flags

`enum gclib_axis_flags`
Flags for building an axis mask, for example in `gclib_set_interrupts()`.

Enumerator

GCLIB_NO_AXES	
GCLIB_AXIS_A	
GCLIB_AXIS_B	
GCLIB_AXIS_C	
GCLIB_AXIS_D	
GCLIB_AXIS_E	
GCLIB_AXIS_F	
GCLIB_AXIS_G	
GCLIB_AXIS_H	
GCLIB_ALL_AXES	

Definition at line 78 of file gclib.h.

11.1.4 Function Documentation

11.1.4.1 gclib_version()

```
gclib_result gclib_version (
            char * version,
            size_t len)
```

Get library version.

Parameters

version	Output buffer for gclib library version.
len	Length of output buffer.

Returns

GCLIB_BUFFER_TOO_SMALL if output does not fit in buffer.

11.1.4.2 gclib_gcaps_version()

```
gclib_result gclib_gcaps_version (
            char * gcaps_version,
            size_t len)
```

Get library version used by current gcaps server.

Parameters

gcaps_version	Output buffer for gcaps library version.
len	Length of output buffer.

Returns

GCLIB_NOT_CONNECTED if unable to reach gcaps

11.1.5 Connection

Topics

- Galil Connect

Typedefs

- typedef enum gclib_connection_t gclib_connection_t

  Connection type, as returned by `gclib_connection_type()`.

Enumerations

- enum gclib_connection_t {
  GCLIB_ETHERNET = 1 ,
  GCLIB_SERIAL ,
  GCLIB_PCI }

  Connection type, as returned by `gclib_connection_type()`.

Functions

- gclib_result gclib_addresses (char *addresses, size_t len)

  Show address, model, and serial of detected controllers, where available. Detected controllers are newline-separated. Each line is comma-separated.
- gclib_result gclib_ip_requests (char *ip_requests, size_t len)

  Show MAC address, model, and serial of all controllers requesting IP addresses, comma-separated. Use `gclib_assign_ip()` to assign an IP address to controllers returned by this method.
- gclib_result gclib_assign_ip (const char *mac, const char *ip)

  Assign IP to a controller.
- gclib_result gclib_set_dhcp_enabled (bool enabled)

  Enable or disable gcaps DHCP server.
- gclib_result gclib_dhcp_enabled (bool *enabled)

  Check if DHCP server is enabled.
- void gclib_force_gcaps (bool on)

  Force future library calls to use either gcaps or direct connections.
- gclib_result gclib_open (gclib_handle *h, const char *address)

  Open a connection to a controller.
- gclib_result gclib_set_baud_rate (gclib_handle h, size_t baud_rate)

  Set baud rate of serial connection.
- gclib_result gclib_close (gclib_handle *h)

  Close connection to controller.
- const char * gclib_error (gclib_handle h)

  Get library error string.
- gclib_result gclib_address (gclib_handle h, char *address, size_t len)

  Get connection address.
- gclib_result gclib_connection_type (gclib_handle h, gclib_connection_t *type)

  Get connection type.

11.1.5.1 Detailed Description

Discover available controllers and open connections

11.1.5.2 Typedef Documentation

11.1.5.2.1 gclib_connection_t

```
typedef enum gclib_connection_t gclib_connection_t
```

Connection type, as returned by `gclib_connection_type()`.

11.1.5.3 Enumeration Type Documentation

11.1.5.3.1 gclib_connection_t

```
enum gclib_connection_t
```

Connection type, as returned by `gclib_connection_type()`.

Enumerator

GCLIB_ETHERNET	
GCLIB_SERIAL	
GCLIB_PCI	

Definition at line 302 of file gclib.h.

11.1.5.4 Function Documentation

11.1.5.4.1 gclib_addresses()

```
gclib_result gclib_addresses (
            char * addresses,
            size_t len)
```

Show address, model, and serial of detected controllers, where available. Detected controllers are newline-separated. Each line is comma-separated.
Available information by connection type:

- Ethernet connections show address, model, and serial.
- USB connections show address and model.
- Serial and PCI connections show address only.

Example output:

```
192.168.0.40, DMC4040 Rev 1.3i, 10601
COM5, DMC-41x3
GALILPCI0
```

Parameters

addresses	Output buffer
len	Length of output buffer

Returns

GCLIB_BUFFER_TOO_SMALL if output does not fit in buffer.

11.1.5.4.2 gclib_ip_requests()

```
gclib_result gclib_ip_requests (
            char * ip_requests,
            size_t len)
```

Show MAC address, model, and serial of all controllers requesting IP addresses, comma-separated. Use gclib_assign_ip() to assign an IP address to controllers returned by this method.
Example output:

```
00:50:4c:20:29:69, DMC4000, 10601
00:50:4c:58:12:34, DMC2105, 4660
```

Parameters

ip_requests	Output buffer. Example output:
len	Length of output buffer

Returns

GCLIB_BUFFER_TOO_SMALL if output does not fit in buffer.

11.1.5.4.3 gclib_assign_ip()

```
gclib_result gclib_assign_ip (
            const char * mac,
            const char * ip)
```

Assign IP to a controller.

Parameters

mac	The MAC address of the controller requesting an IP. Provided by `gclib_ip_requests()`.
ip	The IP address to assign.

Returns

`GCLIB_INVALID_ARGUMENT` if MAC or IP is invalid, or if assigned IP puts controller in an unreachable subnet.

11.1.5.4.4 gclib_set_dhcp_enabled()

```
gclib_result gclib_set_dhcp_enabled (
            bool enabled)
```

Enable or disable gcaps DHCP server.
The gcaps DHCP server is enabled by default. When disabled, `gclib_ip_requests()` and `gclib_assign_ip()` will no longer function. This setting will persist across restarts.

Parameters

enabled	Set whether the DHCP server should be enabled or disabled.

11.1.5.4.5 gclib_dhcp_enabled()

```
gclib_result gclib_dhcp_enabled (
            bool * enabled)
```

Check if DHCP server is enabled.
Use `gclib_set_dhcp()` to enable or disable the gcaps DHCP server. Persists across restarts.

Parameters

enabled	The status of the DHCP server.

Returns

gclib_result `GCLIB_INVALID_ARGUMENT` if another process is using the DHCP server port (67).

11.1.5.4.6 gclib_force_gcaps()

```
void gclib_force_gcaps (
            bool on)
```

Force future library calls to use either gcaps or direct connections.
The default behavior of gclib is to prefer gcaps, but fall back to direct operations when gcaps is not running. This method allows disabling the default behavior process-wide, but cannot be undone.

Parameters

on	If `true`, all future gclib calls will fail if gcaps not found, rather than falling back to direct operations. If `false`, gcaps will not be used for any future gclib calls.

11.1.5.4.7 gclib_open()

```
gclib_result gclib_open (
            gclib_handle * h,
            const char * address)
```

Open a connection to a controller.

Opens a connection through gcaps, if available. Otherwise, a direct connection will be opened. If fallback is not desirable, see `gclib_force_gcaps()`.

If opening a serial connection, the baud rate will default to `115200`. Call `gclib_set_baud_rate()` if necessary before attempting communications.

Parameters

h	Handle that will be needed for future gclib calls operating on this controller.
address	Address of controller, obtained from `gclib_addresses()`.

Returns

`GCLIB_NOT_CONNECTED` if unable to open connection.

11.1.5.4.8 gclib_set_baud_rate()

```
gclib_result gclib_set_baud_rate (
            gclib_handle h,
            size_t baud_rate)
```

Set baud rate of serial connection.

Parameters

h	Handle to an open connection.
baud_rate	Baud rate to use for serial connection.

Returns

`GCLIB_NOT_CONNECTED` if unable to communicate with controller after setting baud rate.

11.1.5.4.9 gclib_close()

```
gclib_result gclib_close (
            gclib_handle * h)
```

Close connection to controller.

Parameters

h	Handle to an open connection. Will be set to `NULL` on success.

Returns

`GCLIB_INVALID_ARGUMENT` if `h` does not appear to be a valid handle to an open connection.

11.1.5.4.10 gclib_error()

```
const char * gclib_error (
            gclib_handle h)
```

Get library error string.

Pass `NULL` to get the last error generated by gclib functions not taking a `gclib_handle`, such as `gclib_addresses()`.

Warning

The string returned is an internal error string, which may change between releases.

Parameters

h	Handle to an open connection. May be NULL.

Returns

The last error string generated by the library.

11.1.5.4.11 gclib_address()

```
gclib_result gclib_address (
            gclib_handle h,
            char * address,
            size_t len)
```

Get connection address.

Parameters

h	Handle to an open connection.
address	Output buffer for connection address.
len	Size of output buffer.

Returns

GCLIB_BUFFER_TOO_SMALL if output does not fit in buffer.

11.1.5.4.12 gclib_connection_type()

```
gclib_result gclib_connection_type (
            gclib_handle h,
            gclib_connection_t * type)
```

Get connection type.

Parameters

h	Handle to an open connection.
type	Will be set to the type of the connection.

Returns

GCLIB_INVALID_ARGUMENT if h does not refer to an open connection.

11.1.5.5 Galil Connect

Functions

- gclib_result gclib_list_servers (char *servers, size_t len)

  List available gcaps servers separated by newline.
- gclib_result gclib_server (char *name, size_t len)

  Current gcaps server.
- gclib_result gclib_set_server (const char *name)

  Set gcaps server.

- gclib_result gclib_published (char *name, size_t len)

  Provides published status of local gcaps server.
- gclib_result gclib_set_published (const char *name)

  Set published status of local gcaps server.

11.1.5.5.1 Detailed Description

Connect to or host a remote gcaps server. While connected, applicable gclib functions will run on the remote gcaps server. This can be useful for accessing closed or distant systems.

11.1.5.5.2 Function Documentation

11.1.5.5.2.1 gclib_list_servers()

```
gclib_result gclib_list_servers (
            char * servers,
            size_t len)
```

List available gcaps servers separated by newline.

Parameters

servers	Output buffer.
len	Length of output buffer.

Returns

GCLIB_BUFFER_TOO_SMALL if output does not fit in buffer.

11.1.5.5.2.2 gclib_server()

```
gclib_result gclib_server (
            char * name,
            size_t len)
```

Current gcaps server.
If connected to the local gcaps server, returns an empty string.

Parameters

name	Output buffer.
len	Length of output buffer.

Returns

GCLIB_BUFFER_TOO_SMALL if output does not fit in buffer.

11.1.5.5.2.3 gclib_set_server()

```
gclib_result gclib_set_server (
            const char * name)
```

Set gcaps server.

Parameters

name	Name of gcaps server, obtained from gclib_list_servers(). Pass NULL to use the local gcaps server.

Returns

GCLIB_INVALID_ARGUMENT if name does not appear to be a valid server.

11.1.5.5.2.4 gclib_published()

```
gclib_result gclib_published (
            char * name,
            size_t len)
```

Provides published status of local gcaps server.

Parameters

name	If published, will contain name of local gcaps server. NULL otherwise.
len	Length of name buffer.

Returns

GCLIB_NOT_CONNECTED if unable to reach gcaps.

11.1.5.5.2.5 gclib_set_published()

```
gclib_result gclib_set_published (
            const char * name)
```

Set published status of local gcaps server.
Persists across restarts.

Parameters

name	Name of server to publish. If NULL, server will be unpublished.

Returns

GCLIB_NOT_CONNECTED if unable to reach gcaps.

11.1.6 Controller

Topics

- Data Record
- Memory
- Unsolicited Data

  Get unsolicited messages, interrupts, or data records, in blocking or callback mode.

Functions

- gclib_result gclib_revision_information (gclib_handle h, char ∗rev_info, size_t len)

  Get revision information (^R^V) from controller.
- gclib_result gclib_serial_number (gclib_handle h, uint32_t ∗serial_number)

  Get serial number from controller.
- gclib_result gclib_command (gclib_handle h, const char ∗command, char ∗buf, size_t len)

  Issues a command to the controller and provides the response.

11.1.6.1 Detailed Description

Manage a Galil controller

11.1.6.2 Function Documentation

11.1.6.2.1 gclib_revision_information()

```
gclib_result gclib_revision_information (
            gclib_handle h,
            char * rev_info,
            size_t len)
```

Get revision information (^R^V) from controller.

Parameters

h	Handle to an open connection.
rev_info	Output buffer for revision information.
len	Size of output buffer.

Returns

- GCLIB_INVALID_ARGUMENT if handle is NULL.
- GCLIB_NOT_CONNECTED if not connected.
- GCLIB_BUFFER_TOO_SMALL if output does not fit in buffer.

11.1.6.2.2 gclib_serial_number()

```
gclib_result gclib_serial_number (
            gclib_handle h,
            uint32_t * serial_number)
```

Get serial number from controller.

Parameters

h	Handle to an open connection.
serial_number	Will be set to the serial number of the controller, or 0 on failure.

Returns

GCLIB_INTERNAL_ERROR if serial number could not be retrieved.

11.1.6.2.3 gclib_command()

```
gclib_result gclib_command (
            gclib_handle h,
            const char * command,
            char * buf,
            size_t len)
```

Issues a command to the controller and provides the response.
Note that a single command cannot be longer than 80 characters, similar to a program line.

Parameters

h	Handle to an open connection.
command	Command to be sent to the controller. Will be terminated with a carriage return.
buf	Output buffer for controller response.
len	Length of output buffer.

Returns

- GCLIB_TIMEOUT if no response is received from the controller.
- GCLIB_COMMAND_ERROR if the controller responds with ?
- GCLIB_BUFFER_TOO_SMALL if the command response cannot fit in the provided buffer.

11.1.6.3 Data Record

Topics

- Amp
- Axis
- Coordinated Move

Typedefs

- typedef enum gclib_mode_of_motion_t gclib_mode_of_motion_t

  The controller's current mode of motion, given by gclib_data_record_mode_of_motion().
- typedef enum gclib_ethernet_status_t gclib_ethernet_status_t

  The current status of an ethernet handle, given by gclib_data_record_ethernet_status().

Enumerations

- enum gclib_mode_of_motion_t {
  GCLIB_NO_MOTION ,
  GCLIB_CONTOUR ,
  GCLIB_POSITION_ABSOLUTE ,
  GCLIB_POSITION_RELATIVE ,
  GCLIB_FIND_EDGE ,
  GCLIB_FIND_INDEX ,
  GCLIB_HOME ,
  GCLIB_VECTOR_MOVE ,
  GCLIB_LINEAR_MOVE = GCLIB_VECTOR_MOVE }

  The controller's current mode of motion, given by gclib_data_record_mode_of_motion().
- enum gclib_ethernet_status_t {
  HANDLE_FREE = 0 ,
  UDP_SLAVE = 1 ,
  TCP_SLAVE = 2 ,
  UDP_MASTER = -1 ,
  TCP_MASTER = -2 ,
  ESTABLISHING_UDP = -5 ,
  ESTABLISHING_TCP = -6 }

  The current status of an ethernet handle, given by gclib_data_record_ethernet_status().

Functions

- uint16_t gclib_data_record_sample (gclib_data_record_handle h)

  The sample in which this data record was generated.
- bool gclib_data_record_input (gclib_data_record_handle h, size_t index)

  Get an input.
- bool gclib_data_record_output (gclib_data_record_handle h, size_t index)

  Get an output.
- uint8_t gclib_data_record_input_bank (gclib_data_record_handle h, size_t index)

  Get a bank of inputs.
- uint8_t gclib_data_record_output_bank (gclib_data_record_handle h, size_t index)

  Get a bank of outputs.
- uint8_t gclib_data_record_error_code (gclib_data_record_handle h)

Get the error code.

- uint32_t gclib_data_record_contour_segment_count (gclib_data_record_handle h)

  Get the contour segment count.
- uint16_t gclib_data_record_contour_buffer_available (gclib_data_record_handle h)

  Get remaining contour buffer.
- bool gclib_data_record_thread_running (gclib_data_record_handle h, size_t thread)

  Returns true if a given thread is running.
- gclib_ethernet_status_t gclib_data_record_ethernet_status (gclib_data_record_handle h, char handle)

  Get ethernet handle status.
- gclib_result gclib_data_record_bytes (gclib_data_record_handle h, char *data_record, size_t len)

  Copy data record to user buffer.

11.1.6.3.1 Detailed Description

Used to get values from a data record handle.
For obtaining data records, see `gclib_subscribe_data_records()`.

11.1.6.3.2 Typedef Documentation

11.1.6.3.2.1 gclib_mode_of_motion_t

`typedef enum gclib_mode_of_motion_t gclib_mode_of_motion_t`
The controller's current mode of motion, given by `gclib_data_record_mode_of_motion()`.

11.1.6.3.2.2 gclib_ethernet_status_t

`typedef enum gclib_ethernet_status_t gclib_ethernet_status_t`
The current status of an ethernet handle, given by `gclib_data_record_ethernet_status()`.

11.1.6.3.3 Enumeration Type Documentation

11.1.6.3.3.1 gclib_mode_of_motion_t

`enum gclib_mode_of_motion_t`
The controller's current mode of motion, given by `gclib_data_record_mode_of_motion()`.

Enumerator

GCLIB_NO_MOTION	
GCLIB_CONTOUR	
GCLIB_POSITION_ABSOLUTE	
GCLIB_POSITION_RELATIVE	
GCLIB_FIND_EDGE	
GCLIB_FIND_INDEX	
GCLIB_HOME	
GCLIB_VECTOR_MOVE	
GCLIB_LINEAR_MOVE	

Definition at line 632 of file gclib.h.

11.1.6.3.3.2 gclib_ethernet_status_t

`enum gclib_ethernet_status_t`
The current status of an ethernet handle, given by `gclib_data_record_ethernet_status()`.

Enumerator

HANDLE_FREE	

Enumerator

UDP_SLAVE	
TCP_SLAVE	
UDP_MASTER	
TCP_MASTER	
ESTABLISHING_UDP	
ESTABLISHING_TCP	

Definition at line 645 of file gclib.h.

11.1.6.3.4 Function Documentation

11.1.6.3.4.1 gclib_data_record_sample()

```
uint16_t gclib_data_record_sample (
            gclib_data_record_handle h)
```

The sample in which this data record was generated.

11.1.6.3.4.2 gclib_data_record_input()

```
bool gclib_data_record_input (
            gclib_data_record_handle h,
            size_t index)
```

Get an input.
Generally, controllers have 8 inputs and outputs per axis bank.

Note

Inputs are active low, meaning this function returns `true` if current is NOT flowing.

Parameters

h	A data record handle.
index	The index of the input to check. Note that index 0 corresponds to @IN[1].

Returns

`true` if input is active.

11.1.6.3.4.3 gclib_data_record_output()

```
bool gclib_data_record_output (
            gclib_data_record_handle h,
            size_t index)
```

Get an output.

11.1.6.3.4.4 gclib_data_record_input_bank()

```
uint8_t gclib_data_record_input_bank (
            gclib_data_record_handle h,
            size_t index)
```

Get a bank of inputs.

11.1.6.3.4.5 gclib_data_record_output_bank()

```
uint8_t gclib_data_record_output_bank (
            gclib_data_record_handle h,
            size_t index)
```

Get a bank of outputs.

11.1.6.3.4.6 gclib_data_record_error_code()

```
uint8_t gclib_data_record_error_code (
            gclib_data_record_handle h)
```

Get the error code.

11.1.6.3.4.7 gclib_data_record_contour_segment_count()

```
uint32_t gclib_data_record_contour_segment_count (
            gclib_data_record_handle h)
```

Get the contour segment count.

11.1.6.3.4.8 gclib_data_record_contour_buffer_available()

```
uint16_t gclib_data_record_contour_buffer_available (
            gclib_data_record_handle h)
```

Get remaining contour buffer.

11.1.6.3.4.9 gclib_data_record_thread_running()

```
bool gclib_data_record_thread_running (
            gclib_data_record_handle h,
            size_t thread)
```

Returns true if a given thread is running.

11.1.6.3.4.10 gclib_data_record_ethernet_status()

```
gclib_ethernet_status_t gclib_data_record_ethernet_status (
            gclib_data_record_handle h,
            char handle)
```

Get ethernet handle status.

Parameters

h	data record handle.
handle	The ethernet handle to check, A - H.

Returns

gclib_ethernet_status_t The ethernet handle status.

11.1.6.3.4.11 gclib_data_record_bytes()

```
gclib_result gclib_data_record_bytes (
            gclib_data_record_handle h,
            char * data_record,
            size_t len)
```

Copy data record to user buffer.
Useful if your device has a nonstandard data record.

Parameters

h	data record handle.
data_record	pointer to user buffer.
len	size of user buffer.

Returns

GCLIB_BUFFER_TOO_SMALL if output does not fit in buffer.

11.1.6.3.5 Amp

Functions

- bool gclib_data_record_over_current (gclib_data_record_handle h, size_t amp)

  Returns over current status.
- bool gclib_data_record_over_voltage (gclib_data_record_handle h, size_t amp)

  Returns over voltage status.
- bool gclib_data_record_over_temp (gclib_data_record_handle h, size_t amp)

  Returns over temperature status.
- bool gclib_data_record_under_voltage (gclib_data_record_handle h, size_t amp)

  Returns under voltage status.
- bool gclib_data_record_electronic_lock_out (gclib_data_record_handle h, size_t amp)

  Returns electronic lockout (ELO) status for a bank.

11.1.6.3.5.1 Detailed Description

Amp error statuses. Amp errors are latched, and will need to be cleared with `AZ` once the underlying issue has been resolved. See the `command reference` for details.

11.1.6.3.5.2 Function Documentation

gclib_data_record_over_current()

```
bool gclib_data_record_over_current (
            gclib_data_record_handle h,
            size_t amp)
```

Returns over current status.

gclib_data_record_over_voltage()

```
bool gclib_data_record_over_voltage (
            gclib_data_record_handle h,
            size_t amp)
```

Returns over voltage status.

gclib_data_record_over_temp()

```
bool gclib_data_record_over_temp (
            gclib_data_record_handle h,
            size_t amp)
```

Returns over temperature status.

gclib_data_record_under_voltage()

```
bool gclib_data_record_under_voltage (
            gclib_data_record_handle h,
            size_t amp)
```

Returns under voltage status.

gclib_data_record_electronic_lock_out()

```
bool gclib_data_record_electronic_lock_out (
            gclib_data_record_handle h,
            size_t amp)
```

Returns electronic lockout (ELO) status for a bank.

11.1.6.3.6 Axis

Functions

- bool gclib_data_record_motor_off (gclib_data_record_handle h, char axis)

  Get motor off status.
- bool gclib_data_record_latch_armed (gclib_data_record_handle h, char axis)

  Get latch armed.
- bool gclib_data_record_final_deceleration (gclib_data_record_handle h, char axis)

  Returns true when axis is in final deceleration.
- bool gclib_data_record_stopping (gclib_data_record_handle h, char axis)

  Returns true if axis is stopping.
- bool gclib_data_record_slewing (gclib_data_record_handle h, char axis)

  Returns true if axis is slewing.
- gclib_mode_of_motion_t gclib_data_record_mode_of_motion (gclib_data_record_handle h, char axis)

  Returns the mode of motion being profiled.
- bool gclib_data_record_negative_direction_move (gclib_data_record_handle h, char axis)

  Returns true if axis velocity is negative.
- bool gclib_data_record_moving (gclib_data_record_handle h, char axis)

  Returns true if motion is being profiled.
- int gclib_data_record_home_phase (gclib_data_record_handle h, char axis)

  Returns home phase when in `GCLIB_HOME` mode.
- bool gclib_data_record_stepper_mode (gclib_data_record_handle h, char axis)

  Returns true if axis is configured as a stepper motor.
- bool gclib_data_record_home_input (gclib_data_record_handle h, char axis)

  Returns the state of the home input.
- bool gclib_data_record_reverse_limit (gclib_data_record_handle h, char axis)

  Returns the state of the reverse limit.
- bool gclib_data_record_forward_limit (gclib_data_record_handle h, char axis)

  Returns the state of the forward limit.
- bool gclib_data_record_latch_input (gclib_data_record_handle h, char axis)

  Returns the state of the latch input.
- bool gclib_data_record_latch_occurred (gclib_data_record_handle h, char axis)

  Returns whether the latch was triggered.
- uint8_t gclib_data_record_stop_code (gclib_data_record_handle h, char axis)

  Returns a code indicating why the motor has stopped. See the `SC command reference` for details.
- int32_t gclib_data_record_reference_position (gclib_data_record_handle h, char axis)

  Returns the reference position.
- int32_t gclib_data_record_position (gclib_data_record_handle h, char axis)

  Returns the position.
- int32_t gclib_data_record_position_error (gclib_data_record_handle h, char axis)

  Returns the position error.
- int32_t gclib_data_record_aux_position (gclib_data_record_handle h, char axis)

  Returns the auxiliary position.
- int32_t gclib_data_record_velocity (gclib_data_record_handle h, char axis)

  Returns the velocity.
- double gclib_data_record_torque (gclib_data_record_handle h, char axis)

  Returns the torque.
- double gclib_data_record_analog_input (gclib_data_record_handle h, char axis)

  Returns the analog input.
- uint8_t gclib_data_record_halls (gclib_data_record_handle h, char axis)

  Returns the hall state.

- int32_t gclib_data_record_variable (gclib_data_record_handle h, char axis)

  Returns the value of a user variable for an axis.
- bool gclib_data_record_hall_error (gclib_data_record_handle h, char axis)

  Returns true if the amplifier for this axis has triggered a hall error.
- bool gclib_data_record_peak_current (gclib_data_record_handle h, char axis)

  Returns true if the amplifier for this axis has triggered a peak current error.

11.1.6.3.6.1 Detailed Description

Axis-specific values, such as position or torque.

11.1.6.3.6.2 Function Documentation

gclib_data_record_motor_off()

```
bool gclib_data_record_motor_off (
            gclib_data_record_handle h,
            char axis)
```

Get motor off status.

gclib_data_record_latch_armed()

```
bool gclib_data_record_latch_armed (
            gclib_data_record_handle h,
            char axis)
```

Get latch armed.

gclib_data_record_final_deceleration()

```
bool gclib_data_record_final_deceleration (
            gclib_data_record_handle h,
            char axis)
```

Returns true when axis is in final deceleration.

gclib_data_record_stopping()

```
bool gclib_data_record_stopping (
            gclib_data_record_handle h,
            char axis)
```

Returns true if axis is stopping.

gclib_data_record_slewing()

```
bool gclib_data_record_slewing (
            gclib_data_record_handle h,
            char axis)
```

Returns true if axis is slewing.

gclib_data_record_mode_of_motion()

```
gclib_mode_of_motion_t gclib_data_record_mode_of_motion (
            gclib_data_record_handle h,
            char axis)
```

Returns the mode of motion being profiled.

gclib_data_record_negative_direction_move()

```
bool gclib_data_record_negative_direction_move (
            gclib_data_record_handle h,
            char axis)
```

Returns true if axis velocity is negative.

gclib_data_record_moving()

```
bool gclib_data_record_moving (
            gclib_data_record_handle h,
            char axis)
```

Returns true if motion is being profiled.

gclib_data_record_home_phase()

```
int gclib_data_record_home_phase (
            gclib_data_record_handle h,
            char axis)
```

Returns home phase when in GCLIB_HOME mode.

gclib_data_record_stepper_mode()

```
bool gclib_data_record_stepper_mode (
            gclib_data_record_handle h,
            char axis)
```

Returns true if axis is configured as a stepper motor.

gclib_data_record_home_input()

```
bool gclib_data_record_home_input (
            gclib_data_record_handle h,
            char axis)
```

Returns the state of the home input.

gclib_data_record_reverse_limit()

```
bool gclib_data_record_reverse_limit (
            gclib_data_record_handle h,
            char axis)
```

Returns the state of the reverse limit.

gclib_data_record_forward_limit()

```
bool gclib_data_record_forward_limit (
            gclib_data_record_handle h,
            char axis)
```

Returns the state of the forward limit.

gclib_data_record_latch_input()

```
bool gclib_data_record_latch_input (
            gclib_data_record_handle h,
            char axis)
```

Returns the state of the latch input.

gclib_data_record_latch_occurred()

```
bool gclib_data_record_latch_occurred (
            gclib_data_record_handle h,
            char axis)
```

Returns whether the latch was triggered.

gclib_data_record_stop_code()

```
uint8_t gclib_data_record_stop_code (
            gclib_data_record_handle h,
            char axis)
```

Returns a code indicating why the motor has stopped. See the SC command reference for details.

gclib_data_record_reference_position()

```
int32_t gclib_data_record_reference_position (
            gclib_data_record_handle h,
            char axis)
```

Returns the reference position.

gclib_data_record_position()

```
int32_t gclib_data_record_position (
            gclib_data_record_handle h,
            char axis)
```

Returns the position.

gclib_data_record_position_error()

```
int32_t gclib_data_record_position_error (
            gclib_data_record_handle h,
            char axis)
```

Returns the position error.

gclib_data_record_aux_position()

```
int32_t gclib_data_record_aux_position (
            gclib_data_record_handle h,
            char axis)
```

Returns the auxiliary position.

gclib_data_record_velocity()

```
int32_t gclib_data_record_velocity (
            gclib_data_record_handle h,
            char axis)
```

Returns the velocity.

gclib_data_record_torque()

```
double gclib_data_record_torque (
            gclib_data_record_handle h,
            char axis)
```

Returns the torque.

gclib_data_record_analog_input()

```
double gclib_data_record_analog_input (
            gclib_data_record_handle h,
            char axis)
```

Returns the analog input.

gclib_data_record_halls()

```
uint8_t gclib_data_record_halls (
            gclib_data_record_handle h,
            char axis)
```

Returns the hall state.

gclib_data_record_variable()

```
int32_t gclib_data_record_variable (
            gclib_data_record_handle h,
            char axis)
```

Returns the value of a user variable for an axis.

gclib_data_record_hall_error()

```
bool gclib_data_record_hall_error (
            gclib_data_record_handle h,
            char axis)
```

Returns true if the amplifier for this axis has triggered a hall error.

gclib_data_record_peak_current()

```
bool gclib_data_record_peak_current (
            gclib_data_record_handle h,
            char axis)
```

Returns true if the amplifier for this axis has triggered a peak current error.

11.1.6.3.7 Coordinated Move

Functions

- uint16_t gclib_data_record_coordinated_move_segment_count (gclib_data_record_handle h, char plane)

  Returns the segment count for the current move.
- uint16_t gclib_data_record_coordinated_move_status (gclib_data_record_handle h, char plane)

  Returns the coordinated move status word.
- bool gclib_data_record_coordinated_move_final_deceleration (gclib_data_record_handle h, char plane)

  Returns true if in final deceleration.
- bool gclib_data_record_coordinated_move_stopping (gclib_data_record_handle h, char plane)

  Returns true if stopping.
- bool gclib_data_record_coordinated_move_slewing (gclib_data_record_handle h, char plane)

  Returns true if slewing.
- bool gclib_data_record_coordinated_move_moving (gclib_data_record_handle h, char plane)

  Returns true if moving.
- int32_t gclib_data_record_coordinated_move_distance (gclib_data_record_handle h, char plane)

  Returns the distance covered in the current move.
- uint16_t gclib_data_record_coordinated_move_buffer_available (gclib_data_record_handle h, char plane)

  Returns the available buffer space.

11.1.6.3.7.1 Detailed Description

Coordinated move values, relevant for `LM` and `CM` modes of motion.

11.1.6.3.7.2 Function Documentation

gclib_data_record_coordinated_move_segment_count()

```
uint16_t gclib_data_record_coordinated_move_segment_count (
            gclib_data_record_handle h,
            char plane)
```

Returns the segment count for the current move.

gclib_data_record_coordinated_move_status()

```
uint16_t gclib_data_record_coordinated_move_status (
            gclib_data_record_handle h,
            char plane)
```

Returns the coordinated move status word.

gclib_data_record_coordinated_move_final_deceleration()

```
bool gclib_data_record_coordinated_move_final_deceleration (
            gclib_data_record_handle h,
            char plane)
```

Returns true if in final deceleration.

gclib_data_record_coordinated_move_stopping()

```
bool gclib_data_record_coordinated_move_stopping (
            gclib_data_record_handle h,
            char plane)
```

Returns true if stopping.

gclib_data_record_coordinated_move_slewing()

```
bool gclib_data_record_coordinated_move_slewing (
            gclib_data_record_handle h,
            char plane)
```

Returns true if slewing.

gclib_data_record_coordinated_move_moving()

```
bool gclib_data_record_coordinated_move_moving (
            gclib_data_record_handle h,
            char plane)
```

Returns true if moving.

gclib_data_record_coordinated_move_distance()

```
int32_t gclib_data_record_coordinated_move_distance (
            gclib_data_record_handle h,
            char plane)
```

Returns the distance covered in the current move.

gclib_data_record_coordinated_move_buffer_available()

```
uint16_t gclib_data_record_coordinated_move_buffer_available (
            gclib_data_record_handle h,
            char plane)
```

Returns the available buffer space.

11.1.6.4 Memory

Functions

- gclib_result gclib_program (gclib_handle h, char ∗program, size_t len)

  Get the controller's current program.
- gclib_result gclib_set_program (gclib_handle h, const char ∗program, const char ∗insert)

  Set the program on the controller.
- gclib_result gclib_array (gclib_handle h, const char ∗name, char ∗buf, size_t len, size_t start, size_t end)

  Get an array from the controller.
- gclib_result gclib_set_array (gclib_handle h, const char ∗name, const char ∗buf, size_t start, size_t end)

  Set an array on the controller.
- gclib_result gclib_set_firmware (gclib_handle h, const char ∗file_path)

  Set the firmware on the controller.

11.1.6.4.1 Detailed Description

Manage controller firmware, program, and arrays.

11.1.6.4.2 Function Documentation

11.1.6.4.2.1 gclib_program()

```
gclib_result gclib_program (
            gclib_handle h,
            char * program,
            size_t len)
```

Get the controller's current program.

Parameters

h	Handle to an open connection.
program	Output buffer for program.
len	Length of output buffer.

Returns

GCLIB_BUFFER_TOO_SMALL if program does not fit in buffer.

11.1.6.4.2.2 gclib_set_program()

```
gclib_result gclib_set_program (
            gclib_handle h,
            const char * program,
            const char * insert)
```

Set the program on the controller.

Parameters

h	Handle to an open connection.
program	Program to write to controller memory. Lines may be separated by \r, \r\n, or \n.
insert	Either a line number or a label name to start downloading from.

Returns

GCLIB_INVALID_ARGUMENT if program is not accepted by controller (for example, if a program is already running).

11.1.6.4.2.3 gclib_array()

```
gclib_result gclib_array (
            gclib_handle h,
            const char * name,
            char * buf,
            size_t len,
            size_t start,
            size_t end)
```

Get an array from the controller.

Parameters

h	Handle to an open connection.
name	Array name on controller.
buf	Output buffer for comma-separated array values.
len	Length of output buffer.
start	Index to start reading array.
end	Index to stop reading array.

Returns

GCLIB_INVALID_ARGUMENT if array does not exist on controller.

11.1.6.4.2.4 gclib_set_array()

```
gclib_result gclib_set_array (
            gclib_handle h,
            const char * name,
            const char * buf,
            size_t start,
            size_t end)
```

Set an array on the controller.
Array will be automatically dimensioned unless one of the following is true:

- Start or end are nonzero
- Controller is currently recording arrays
- Modbus array write is enabled

Parameters

h	Handle to open connection.
name	Array name on controller (will be dimensioned if not present).
buf	Array buffer. Elements may be separated by `,`, `\r`, or `\r\n`. Do not use spaces.
start	Index to start writing over an existing array.
end	Index to stop writing over an existing array. Leave 0 to write entire `buf` to controller.

Returns

GCLIB_INVALID_ARGUMENT if array does not exist but cannot be automatically dimensioned, or start / end are invalid for array.

11.1.6.4.2.5 gclib_set_firmware()

```
gclib_result gclib_set_firmware (
            gclib_handle h,
            const char * file_path)
```

Set the firmware on the controller.

Parameters

h	Handle to open connection.
file_path	Path to firmware file.

Returns

GCLIB_INVALID_ARGUMENT if firmware file cannot be opened, or is rejected by the controller.

11.1.6.5 Unsolicited Data

Get unsolicited messages, interrupts, or data records, in blocking or callback mode.

Functions

- gclib_result gclib_set_interrupts (gclib_handle h, gclib_interrupt_type interrupt_mask, gclib_axis_flags motion_complete_axes, gclib_digital_input_flags digital_inputs)

  Configure which interrupts will be generated by the controller.
- gclib_result gclib_set_data_records (gclib_handle h, size_t period_ms)

  Configure controller data records.
- gclib_result gclib_subscribe_messages (gclib_handle h, void(callback)(void ∗user_data, const char ∗message), void ∗user_data)

  Subscribe to unsolicited messages.
- gclib_result gclib_subscribe_interrupts (gclib_handle h, void(callback)(void ∗user_data, gclib_interrupt_t interrupt), void ∗user_data)

  Subscribe to interrupts.
- gclib_result gclib_subscribe_data_records (gclib_handle h, void(callback)(void ∗user_data, gclib_data_record_handle data_record), void ∗user_data)

  Subscribe to data records.
- gclib_result gclib_subscribe_progress (gclib_handle h, void(callback)(void ∗user_data, size_t current, size_t max), void ∗user_data)

  Subscribe to progress for gclib_set_program(), gclib_set_array(), and gclib_set_firmware().
- gclib_result gclib_unsubscribe (gclib_handle h, void ∗callback)

  Unsubscribe from messages, interrupts, data records, or progress.
- gclib_result gclib_message (gclib_handle h, char ∗message, size_t len, int timeout)

  Get a queued unsolicited message, or wait up to timeout ms for one to arrive.
- gclib_result gclib_interrupt (gclib_handle h, gclib_interrupt_t ∗interrupt, int timeout)

  Get a queued interrupt, or wait up to timeout ms for one to arrive.
- gclib_result gclib_data_record (gclib_handle h, gclib_data_record_handle ∗data_record, int timeout)

  Get a queued data record, or wait up to timeout ms for one to arrive.

11.1.6.5.1 Detailed Description

Get unsolicited messages, interrupts, or data records, in blocking or callback mode.

11.1.6.5.1.1 Blocking Mode

Blocking mode is simpler to use, but has tradeoffs. If you want unsolicited data as soon as it arrives, you must block the thread to wait for it, which may not be ideal. If you want to keep the thread running, then you must periodically check for queued unsolicited data, which can add unwanted latency.
To subscribe in blocking mode, simply do not pass a callback or data pointer when you subscribe. These methods will begin queueing any unsolicited data sent by the controller.

- **C**

```
gclib_subscribe_messages(h, NULL, NULL);
gclib_subscribe_interrupts(h, NULL, NULL);
gclib_subscribe_data_records(h, NULL, NULL);
```

- **C++**

```
dmc.subscribe_messages();
dmc.subscribe_interrupts();
dmc.subscribe_data_records();
```

- **Python**
- **Java**
- **C#**
- **VB**
- **LabVIEW**

11.1.6.5.1.2 Callback Mode

Callback mode allows your thread to stay running while enabling immediate response to unsolicited data. When data arrives, your callback function will be invoked on a separate, dedicated thread. Due to this, callback mode can be more complicated to use if you are not familiar with thread synchronization.

- **C** To subscribe in callback mode, pass a callback function and an optional data pointer when you subscribe. When unsolicited data arrives, your callback will be invoked and passed the data pointer. Use this to allow the callback function access to any data it needs, and to allow data to persist between callbacks.

```
void handle_message(void* user_data, const char* message) {
    printf("Message received: %s\n", message);
}
void handle_interrupt(void* user_data, gclib_interrupt_t interrupt) {
    printf("Interrupt received, type %i, raw status byte %i": interrupt.type, interrupt.status);
}
void handle_data_record(void* user_data, gclib_data_record_handle data_record) {
    printf("Data record received, sample %i", gclib_data_record_sample(data_record));
}

gclib_subscribe_messages(h, handle_messages, user_data);
gclib_subscribe_interrupts(h, handle_interrupts, user_data);
gclib_subscribe_data_records(h, handle_data_records, user_data);
```

- **C++** To subscribe in callback mode, pass a callback function when you subscribe. When unsolicited data arrives, your callback will be invoked. Use lambda captures or `std::bind` to allow the callback function access to any data it needs, and to allow data to persist between callbacks.

```
dmc.subscribe_messages([user_data]{
    std::cout "Message received: " « messages « "\n";
});
dmc.subscribe_interrupts([user_data]{
    std::cout « "Interrupt received, type " « interrupt.type « ", raw status byte: " « interrupt.status «
      "\n";
});
dmc.subscribe_data_records([user_data](DataRecord data_record){
    std::cout « "Data record received, sample " « data_record.sample();
});
```

- **Python**
- **Java**
- **C#**
- **VB**
- **LabVIEW**

User Data

The user data pointer can be used to pass arbitrary data to the callback function. The following example uses a struct to provide the callback access to the gclib handle itself, a count to increment when messages are received, and an atomic bool to signal when the program is ready to exit. Note that the handle is safe to use concurrently, and the count doesn't need to be atomic (since it is only used in the callback thread). As soon as the tenth message is received, the program is stopped and the connection is closed from the callback method, and the main thread is signaled to exit.

```
#include <gclib.h>
#include <stdio.h>
#include <stdatomic.h>
#include "sleep.h"

struct context {
    gclib_handle h; // gclib is thread-safe
    int count; // Only used in the callback thread
    atomic_bool done; // Used in both threads
};
void handle_message(void* user_data, const char* message) {
    struct context* ctx = user_data;
    ctx->count = ctx->count + 1;
    printf("Received message #%i at TIME %i\n", ctx->count, (int)atof(message));
    if (ctx->count == 10) {
        gclib_command(ctx->h, "ST", NULL, 0);
        gclib_close(&ctx->h);
        ctx->done = true;
    }
};

int main(int argc, char* argv[]) {
    struct context ctx = {NULL, 0, 0};
    gclib_open(&ctx.h, argv[1]);
    gclib_subscribe_messages(ctx.h, handle_message, &ctx);
    gclib_set_program(ctx.h, "#msg\nMGTIME\nWT100\nJP#msg\nEN", NULL);
    gclib_command(ctx.h, "XQ", NULL, 0);
    while (!ctx.done) {
        msleep(100 * 1000); // Do work
    }
}
```

11.1.6.5.2 Function Documentation

11.1.6.5.2.1 gclib_set_interrupts()

```
gclib_result gclib_set_interrupts (
            gclib_handle h,
            gclib_interrupt_type interrupt_mask,
            gclib_axis_flags motion_complete_axes,
            gclib_digital_input_flags digital_inputs)
```

Configure which interrupts will be generated by the controller.
By default, the controller will not generate any interrupts.
Controllers can be configured to send only an exact list of desired interrupts, to avoid unnecessary processing and network traffic.
Example usage:

```
gclib_set_interrupts(h, GCLIB_ALL_INTERRUPTS, 0, 0); // Enable all interrupts.
gclib_set_interrupts(h,
    GCLIB_MOTION_COMPLETE | GCLIB_DIGITAL_INPUT_LOW, // Only enable two interrupts
    GCLIB_AXIS_A | GCLIB_AXIS_F, // Only trigger GCLIB_MOTION_COMPLETE on axis A or F
    GCLIB_DIGITAL_INPUT_1 | GCLIB_DIGITAL_INPUT_4); // Only trigger GCLIB_DIGITAL_INPUT_LOW on digital input
      1 and 4.
gclib_set_interrupts(h, 0, 0, 0); // Disable interrupts
```

See `gclib_subscribe_interrupts()` to enable receiving interrupts from the controller.

Parameters

h	Handle to an open connection.
interrupt_mask	Bitmask of desired interrupts. Use `GCLIB_NO_INTERRUPTS` (0) to disable interrupts. Note that User Interrupts cannot be disabled.
axis_mask	When `GCLIB_MOTION_COMPLETE` is in `interrupt_mask`, this axis mask controls which axes will trigger a `GCLIB_MOTION_COMPLETE` interrupt. interrupts. If `0`, all axes will be selected.

Parameters

digital_input_mask	When GCLIB_DIGITAL_INPUT_LOW is in interrupt_mask, this bitmask selects which digital inputs will trigger an interrupt. If 0, all digital inputs will be selected.

Returns

GCLIB_TIMEOUT if no response from controller after configuring interrupts.

11.1.6.5.2.2 gclib_set_data_records()

```
gclib_result gclib_set_data_records (
            gclib_handle h,
            size_t period_ms)
```

Configure controller data records.
By default, data records are not sent by the controller.
See gclib_subscribe_data_records() to enable receiving data records from the controller.

Parameters

h	Handle to an open connection.
period_ms	Period between data records. Pass 0 to disable data records.

Returns

GCLIB_TIMEOUT if no response from controller after configuring data records.

11.1.6.5.2.3 gclib_subscribe_messages()

```
gclib_result gclib_subscribe_messages (
            gclib_handle h,
            void(callback)(void *user_data, const char *message) ,
            void * user_data)
```

Subscribe to unsolicited messages.
This method registers a callback to be run every time an unsolicited message is received from the controller.

Note

The callback will be invoked on a separate, dedicated thread. Be careful when accessing any shared state.

When a callback is not provided, messages are instead queued for later retrieval by gclib_message().
On ethernet connections, a handle is opened for unsolicited data (if not already open) and the controller is configured to use it for messages.

Parameters

h	Handle to an open connection.
callback	Pass a callback function to get messages asynchronously, or NULL if you would like to use the gclib_message() blocking API. Replaces any previously set message callback.
user_data	An arbitrary pointer that will be passed to your callback function. Can be used to provide state that persists between callbacks.

Returns

GCLIB_NOT_SUBSCRIBED if unable to subscribe to messages.

11.1.6.5.2.4 gclib_subscribe_interrupts()

```
gclib_result gclib_subscribe_interrupts (
            gclib_handle h,
            void(callback)(void *user_data, gclib_interrupt_t interrupt) ,
            void * user_data)
```

Subscribe to interrupts.
This method registers a callback to be run every time an interrupt is received from the controller. See `gclib_set_interrupts()` to enable and configure controller interrupts.

Note

The callback will be invoked on a separate, dedicated thread. Be careful when accessing any shared state.

When a callback is not provided, interrupts are instead queued for later retrieval by `gclib_interrupt()`.
On ethernet connections, a handle is opened for unsolicited data (if not already open) and the controller is configured to use it for interrupts.

Parameters

h	Handle to an open connection.
callback	Pass a callback function to get interrupts asynchronously, or `NULL` if you would like to use the `gclib_interrupt()` blocking API. Replaces any previously set interrupt callback.
user_data	An arbitrary pointer that will be passed to your callback function. Can be used to provide state that persists between callbacks.

Returns

`GCLIB_NOT_SUBSCRIBED` if unable to subscribe to interrupts.

11.1.6.5.2.5 gclib_subscribe_data_records()

```
gclib_result gclib_subscribe_data_records (
            gclib_handle h,
            void(callback)(void *user_data, gclib_data_record_handle data_record) ,
            void * user_data)
```

Subscribe to data records.
This method registers a callback to be run every time a data record is received from the controller. See `gclib_set_data_records()` to enable data records and configure how often they will be generated by the controller.

Note

The callback will be invoked on a separate, dedicated thread. Be careful when accessing any shared state.

When a callback is not provided, data records are instead queued for later retrieval by `gclib_data_record()`.
On ethernet connections, a handle is opened for unsolicited data (if not already open) and the controller is configured to use it for data records.

Parameters

h	Handle to an open connection.
callback	Pass a callback function to get data records asynchronously, or `NULL` if you would like to use the `gclib_data_record()` blocking API. Replaces any previously registered data record callback.
user_data	An arbitrary pointer that will be passed to your callback function. Can be used to provide state that persists between callbacks.

Returns

`GCLIB_NOT_SUBSCRIBED` if unable to subscribe to data records.

11.1.6.5.2.6 gclib_subscribe_progress()

```
gclib_result gclib_subscribe_progress (
            gclib_handle h,
            void(callback)(void *user_data, size_t current, size_t max) ,
            void * user_data)
```

Subscribe to progress for `gclib_set_program()`, `gclib_set_array()`, and `gclib_set_firmware()`.

Note

The callback will be invoked on a separate, dedicated thread. Be careful when accessing any shared state.

Parameters

h	Handle to an open connection.
callback	A callback function that will get called anytime part of a large write is completed. Replaces any previously registered progress callback.
user_data	An arbitrary pointer that will be passed to your callback function. Can be used to provide state that persists between callbacks.

Returns

`GCLIB_INVALID_ARGUMENT` if **callback** is `NULL`.

11.1.6.5.2.7 gclib_unsubscribe()

```
gclib_result gclib_unsubscribe (
            gclib_handle h,
            void * callback)
```

Unsubscribe from messages, interrupts, data records, or progress.

Parameters

h	Handle to an open connection.
callback	The callback function that was previously used to subscribe. If `NULL`, all subscriptions will be cleared.

Returns

`GCLIB_INVALID_ARGUMENT` if **callback** doesn't match any existing subscription.

11.1.6.5.2.8 gclib_message()

```
gclib_result gclib_message (
            gclib_handle h,
            char * message,
            size_t len,
            int timeout)
```

Get a queued unsolicited message, or wait up to `timeout` ms for one to arrive.

Parameters

h	Handle to an open connection.
message	Output buffer for the message.
len	Length of output buffer.
timeout	Timeout value in milliseconds. If `-1`, wait forever.

Returns

- GCLIB_TIMEOUT if no message arrives for timeout ms.
- GCLIB_BUFFER_TOO_SMALL if output does not fit in buffer.

11.1.6.5.2.9 gclib_interrupt()

```
gclib_result gclib_interrupt (
             gclib_handle h,
             gclib_interrupt_t * interrupt,
             int timeout)
```

Get a queued interrupt, or wait up to timeout ms for one to arrive.

Parameters

h	Handle to an open connection.
interrupt	Output interrupt.
timeout	Timeout value in milliseconds. If −1, wait forever.

Returns

GCLIB_TIMEOUT if no interrupt arrives after timeout ms.

11.1.6.5.2.10 gclib_data_record()

```
gclib_result gclib_data_record (
             gclib_handle h,
             gclib_data_record_handle * data_record,
             int timeout)
```

Get a queued data record, or wait up to timeout ms for one to arrive.
The returned handle remains valid until the next call to gclib_data_record().

Parameters

h	Handle to an open connection.
data_record	Output data record handle.
timeout	Timeout value in milliseconds. If −1, wait forever.

Returns

GCLIB_TIMEOUT if no data record arrives after timeout ms.

11.2 C (Legacy)

For backwards compatibility with gclib 2.2.1 and below.

Topics

- Connection

  Discover available controllers and open connections.
- Controller

  Manage a Galil controller.
- Galil Connect

  Host or connect to a remote gcaps instance.

Functions

- GCLIB_DEPRECATED void GSleep (unsigned int timeout_ms)
- GCLIB_DEPRECATED GReturn 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_DEPRECATED void GError (GReturn rc, GCStringOut error, GSize error_len)

  Provides a human-readable description string for return codes.

11.2.1 Detailed Description

For backwards compatibility with gclib 2.2.1 and below.

11.2.2 Function Documentation

11.2.2.1 GSleep()

```
GCLIB_DEPRECATED void GSleep (
            unsigned int timeout_ms)
```

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Parameters

timeout_ms	The timeout, in milliseconds, to block before returning.

See GWaitForBool() for an example.

11.2.2.2 GVersion()

```
GCLIB_DEPRECATED GReturn GVersion (
            GCStringOut ver,
            GSize ver_len)
```

Uses GUtility(), G_UTIL_VERSION and G_UTIL_GCAPS_VERSION to provide the library and gcaps version numbers.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_version()` and `gclib_gcaps_version()` instead.

Parameters

ver	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so.
ver_len	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.

11.2.2.3 GError()

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

Provides a human-readable description string for return codes.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Parameters

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

11.2.3 Connection

Discover available controllers and open connections.

Functions

- GCLIB_DEPRECATED GReturn GOpen (GCStringIn connection_string, GCon ∗g)

  Open a connection to a Galil Controller.
- GCLIB_DEPRECATED GReturn GClose (GCon g)

  Closes a connection to a Galil Controller.
- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn GInfo (GCon g, GCStringOut info, GSize info_len)

  Uses GUtility() and G_UTIL_INFO to provide a useful connection string.
- GCLIB_DEPRECATED GReturn GTimeout (GCon g, short timeout_ms)

  Uses GUtility() and G_UTIL_TIMEOUT_OVERRIDE to set the library timeout.
- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn 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.

11.2.3.1 Detailed Description

Discover available controllers and open connections.

11.2.3.2 Function Documentation

11.2.3.2.1 GOpen()

```
GCLIB_DEPRECATED GReturn GOpen (
            GCStringIn connection_string,
            GCon * g)
```

Open a connection to a Galil Controller.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_open()` instead.

Parameters

connection_string	Null-terminated connection string. See table below.
g	Pointer to user's GCon variable. On success, the library will fill the user's variable with the handle to use for the rest of the connection. A valid g 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	**Simple address to hardware**	*IP address*, *PCI*, *COM port*	--address COM1
-a	shorthand for --address	See *Address Ranges* below	-a GALILPCI1
{no switch}	--address is implicit for any lone token		192.168.0.42
--baud	**Baud rate**	(115200), *valid baud...*	COM2 --baud 19200
-b	shorthand for --baud		COM3 -b 38400
--direct	**Connect directly to hardware instead of via** gcaps		-a GALILPCI2 --direct
-d	shorthand for --direct		GALILPCI2 -d
--subscribe	**Subscribe to messages, data records, and/or interrupts**	(NONE), MG, DR, EI, ALL	192.168.0.42 --subscribe MG
-s	shorthand for --subscribe		192.168.0.42 -s DR -s EI
--timeout	**timeout in ms**	(5000), *0-65535*	192.168.0.42 --timeout 5000
-t	shorthand for --timeout		GALILPCI2 -t 500
The following address switches are ignored as of gclib 2.4.0. They are still accepted for backwards compatibility.			
--command	**Command-and-response socket protocol**	(TCP), UDP	192.168.0.42 --command TCP
-c	shorthand for --command		192.168.0.42 -c UDP
--unsolicited	**Unsolicited socket protocol**	(UDP), NONE	192.168.0.42 --unsolicited NONE
-u	shorthand for --unsolicited		192.168.1.42 -u UDP
--p1	**Primary port for command-and-response traffic**	(23), *valid port number*	192.168.0.42 --p1 5000
--p2	**Secondary port for unsolicited traffic**	(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

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.

11.2.3.2.2 GClose()

```
GCLIB_DEPRECATED GReturn GClose (
            GCon g)
```

Closes a connection to a Galil Controller.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_close()` instead.

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.

Parameters

g	Connection's handle.

Returns

The success status or error code of the function. See gclib_errors.h for possible values.

11.2.3.2.3 GAddresses()

```
GCLIB_DEPRECATED GReturn 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.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_addresses()` instead.

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

addresses	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.
addresses_len	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.

11.2.3.2.4 GInfo()

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

Uses GUtility() and G_UTIL_INFO to provide a useful connection string.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_address()`, `gclib_revision_information()`, and `gclib_serial_number()` instead.

Parameters

g	Connection's handle.
info	Buffer to hold the output string. Buffer will be null terminated, even if the data must be truncated to do so.
info_len	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
```

11.2.3.2.5 GTimeout()

```
GCLIB_DEPRECATED GReturn GTimeout (
            GCon g,
            short timeout_ms)
```

Uses GUtility() and G_UTIL_TIMEOUT_OVERRIDE to set the library timeout.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Parameters

g	Connection's handle.
timeout_ms	The value to be used for the timeout. Use `G_USE_INITIAL_TIMEOUT` to set the timeout back to the initial GOpen() value, `--timeout`.

Returns

The success status or error code of the function. See gclib_errors.h for possible values.

11.2.3.2.6 GIpRequests()

```
GCLIB_DEPRECATED GReturn 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.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_ip_requests()` instead.

Parameters

requests	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.
requests_len	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 infomation 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
```

11.2.3.2.7 GAssign()

```
GCLIB_DEPRECATED GReturn 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.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_assign_ip()` instead.

Parameters

ip	The null-terminated ip address to assign. The hardware should not yet have an IP address.
mac	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, specfify 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.

11.2.4 Controller

Manage a Galil controller.

Topics

- Communication

  Send commands.
- Memory

  Manage controller memory, such as program and arrays.
- Unsolicited Data

  Receive messages, interrupts, and data records.

Functions

- GCLIB_DEPRECATED GReturn GRead (GCon g, GBufOut buffer, GSize buffer_len, GSize *bytes_read)

  Does nothing and returns G_GCLIB_ERROR.
- GCLIB_DEPRECATED GReturn GWrite (GCon g, GBufIn buffer, GSize buffer_len)

  Does nothing and returns G_GCLIB_ERROR.
- GCLIB_DEPRECATED GReturn GUtility (GCon g, GOption request, GMemory memory1, GMemory memory2)

  Does nothing and returns G_GCLIB_ERROR.
- GCLIB_DEPRECATED GReturn GWaitForBool (GCon g, GCStringIn predicate, int trials)

  Blocking call that returns when the controller evaluates the predicate as true.
- GCLIB_DEPRECATED GReturn GMotionComplete (GCon g, GCStringIn axes)

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

11.2.4.1 Detailed Description

Manage a Galil controller.

11.2.4.2 Function Documentation

11.2.4.2.1 GRead()

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

Does nothing and returns G_GCLIB_ERROR.

Deprecated This function only exists for ABI compatibility, and will be removed in a future gclib version.

11.2.4.2.2 GWrite()

```
GCLIB_DEPRECATED GReturn GWrite (
            GCon g,
            GBufIn buffer,
            GSize buffer_len)
```

Does nothing and returns G_GCLIB_ERROR.

Deprecated This function only exists for ABI compatibility, and will be removed in a future gclib version.

11.2.4.2.3 GUtility()

```
GCLIB_DEPRECATED GReturn GUtility (
            GCon g,
            GOption request,
            GMemory memory1,
            GMemory memory2)
```

Does nothing and returns G_GCLIB_ERROR.

Deprecated This function only exists for ABI compatibility, and will be removed in a future gclib version.

11.2.4.2.4 GWaitForBool()

```
GCLIB_DEPRECATED GReturn GWaitForBool (
            GCon g,
            GCStringIn predicate,
            int trials)
```

Blocking call that returns when the controller evaluates the predicate as true.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

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

g	Connection's handle.
predicate	A null-terminated string containing the predicate to be polled. The predicate will be enclosed in parentheses and used in the command `MG (predicate)` to return the value.
trials	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.

11.2.4.2.5 GMotionComplete()

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

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

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

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

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

Returns

The success status or error code of the function. See gclib_errors.h for possible values.

11.2.4.3 Communication

Send commands.

Functions

- GCLIB_DEPRECATED GReturn GCommand (GCon g, GCStringIn command, GBufOut buffer, GSize buffer_len, GSize *bytes_returned)

  Performs a command-and-response transaction on the connection.
- GCLIB_DEPRECATED GReturn GCmd (GCon g, GCStringIn command)

  Wrapper around GCommand for use when the return value is not desired.
- GCLIB_DEPRECATED GReturn GCmdT (GCon g, GCStringIn command, GCStringOut trimmed_response, GSize response_len, GCStringOut *front)

  Wrapper around GCommand that trims the response.
- GCLIB_DEPRECATED GReturn GCmdI (GCon g, GCStringIn command, int *value)

  Wrapper around GCommand that provides the return value of a command parsed into an int.
- GCLIB_DEPRECATED GReturn GCmdD (GCon g, GCStringIn command, double *value)

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

11.2.4.3.1 Detailed Description

Send commands.

11.2.4.3.2 Function Documentation

11.2.4.3.2.1 GCommand()

```
GCLIB_DEPRECATED GReturn GCommand (
            GCon g,
            GCStringIn command,
            GBufOut buffer,
```

```
            GSize buffer_len,
            GSize * bytes_returned)
```

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

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_command()` instead.

Parameters

g	Connection's handle.
command	Null-terminated command string to send to the controller. The library will append a carriage return to the command string.
buffer	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.
buffer_len	The size of the response buffer.
bytes_returned	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.

11.2.4.3.2.2 GCmd()

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

Wrapper around GCommand for use when the return value is not desired.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

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

Parameters

g	Connection's handle.
command	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.

11.2.4.3.2.3 GCmdT()

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

Wrapper around GCommand that trims the response.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

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

g	Connection's handle.
command	Null-terminated command string to send to the controller.
trimmed_response	The trimmed response from the controller. Trailing space is trimmed by null terminating any trailing spaces, carriage returns, or line feeds.
response_len	The length of the trimmed_response buffer.
front	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.

11.2.4.3.2.4 GCmdI()

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

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

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

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

Parameters

g	Connection's handle.
command	Null-terminated command string to send to the controller.
value	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.

11.2.4.3.2.5 GCmdD()

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

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

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

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

g	Connection's handle.
command	Null-terminated command string to send to the controller.
value	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.

11.2.4.4 Memory

Manage controller memory, such as program and arrays.

Functions

- GCLIB_DEPRECATED GReturn GProgramDownload (GCon g, GCStringIn program, GCStringIn preprocessor)

  Downloads a program to the controller's program buffer.
- GCLIB_DEPRECATED GReturn GProgramUpload (GCon g, GBufOut buffer, GSize buffer_len)

  Uploads a program from the controller's program buffer.
- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn 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_DEPRECATED GReturn GFirmwareDownload (GCon g, GCStringIn filepath)

  Upgrade firmware.
- GCLIB_DEPRECATED GReturn GProgramDownloadFile (GCon g, GCStringIn file_path, GCStringIn preprocessor)

  Program download from file.
- GCLIB_DEPRECATED GReturn GProgramUploadFile (GCon g, GCStringIn file_path)

  Program upload to file.
- GCLIB_DEPRECATED GReturn GArrayDownloadFile (GCon g, GCStringIn file_path)

  Array download from file.
- GCLIB_DEPRECATED GReturn GArrayUploadFile (GCon g, GCStringIn file_path, GCStringIn names)

  Array upload to file.
- GCLIB_DEPRECATED GReturn GSetupDownloadFile (GCon g, GCStringIn file_path, GOption options, GCStringOut info, GSize info_len)

  Download a saved controller configuration from a file.

11.2.4.4.1 Detailed Description

Manage controller memory, such as program and arrays.

11.2.4.4.2 Function Documentation

11.2.4.4.2.1 GProgramDownload()

```
GCLIB_DEPRECATED GReturn GProgramDownload (
            GCon g,
            GCStringIn program,
            GCStringIn preprocessor)
```

Downloads a program to the controller's program buffer.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_set_program()` instead.

Parameters

g	Connection's handle.
program	Null-terminated program for download.
preprocessor	Options string for preprocessing the program before sending it to the controller. Null allows the library to use defaults for the download. See the Program Preprocessor documentation for options.

Returns

The success status or error code of the function. See gclib_errors.h for possible values.

11.2.4.4.2.2 GProgramUpload()

```
GCLIB_DEPRECATED GReturn GProgramUpload (
            GCon g,
            GBufOut buffer,
            GSize buffer_len)
```

Uploads a program from the controller's program buffer.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_program()` instead.

Parameters

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

Returns

The success status or error code of the function. See gclib_errors.h for possible values.

11.2.4.4.2.3 GArrayDownload()

```
GCLIB_DEPRECATED GReturn 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.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_set_array()` instead.

Warning

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

Parameters

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

Returns

The success status or error code of the function. See gclib_errors.h for possible values.

11.2.4.4.2.4 GArrayUpload()

```
GCLIB_DEPRECATED GReturn 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.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_array()` instead.

Parameters

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

Returns

The success status or error code of the function. See gclib_errors.h for possible values.

11.2.4.4.2.5 GFirmwareDownload()

```
GCLIB_DEPRECATED GReturn GFirmwareDownload (
            GCon g,
            GCStringIn filepath)
```

Upgrade firmware.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_set_firmware()` instead.

Parameters

g	Connection's handle.
filepath	The full file path to the Galil-supplied firmware hex file. See `http://www.galil.com/downloads/firmware`

Returns

The success status or error code of the function. See gclib_errors.h for possible values.

11.2.4.4.2.6 GProgramDownloadFile()

```
GCLIB_DEPRECATED GReturn GProgramDownloadFile (
            GCon g,
            GCStringIn file_path,
            GCStringIn preprocessor)
```

Program download from file.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Parameters

g	Connection's handle.
file_path	Null-terminated string containing the path to the program file.
preprocessor	Options string for preprocessing the program before sending it to the controller. See GProgramDownload().

Returns

The success status or error code of the function. See gclib_errors.h for possible values.

11.2.4.4.2.7 GProgramUploadFile()

```
GCLIB_DEPRECATED GReturn GProgramUploadFile (
            GCon g,
            GCStringIn file_path)
```

Program upload to file.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Parameters

g	Connection's handle.
file_path	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.

11.2.4.4.2.8 GArrayDownloadFile()

```
GCLIB_DEPRECATED GReturn GArrayDownloadFile (
            GCon g,
            GCStringIn file_path)
```

Array download from file.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

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

Parameters

g	Connection's handle.
file_path	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.

11.2.4.4.2.9 GArrayUploadFile()

```
GCLIB_DEPRECATED GReturn GArrayUploadFile (
            GCon g,
            GCStringIn file_path,
            GCStringIn names)
```

Array upload to file.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Uploads the entire controller array table or a subset and saves the data as a csv file specified by `file_path`.

Parameters

g	Connection's handle.
file_path	Null-terminated string containing the path to the array file, file will be overwritten if it exists.
names	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.

11.2.4.4.2.10 GSetupDownloadFile()

```
GCLIB_DEPRECATED GReturn GSetupDownloadFile (
            GCon g,
            GCStringIn file_path,
            GOption options,
            GCStringOut info,
            GSize info_len)
```

Download a saved controller configuration from a file.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Parameters

g	Connection's handle.
file_path	Null-terminated string containing the path to the gcb file.
options	Bit mask to determine what configuration data to download. See below for all options.
info	Optional pointer to a buffer to store the controller info. If no info is needed, specify as NULL.
info_len	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	**KPA**, **KIA**, **KDA**, etc...
3	0x0008	Restore variables	Variables are listed by the **LV** command
4	0x0010	Restore arrays	Arrays are listed by the **LA** command
5	0x0020	Restore program	The program is listed by the **LS** command
31	0x8000	Ignore errors	Ignore invalid parameter errors and continue restoring data. GSetupDownloadFile() 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 = G`
// 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;
```

11.2.4.5 Unsolicited Data

Receive messages, interrupts, and data records.

Functions

- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn GMessage (GCon g, GCStringOut buffer, GSize buffer_len)

  Provides access to unsolicited messages from the controller.
- GCLIB_DEPRECATED GReturn GInterrupt (GCon g, GStatus *status_byte)

  Provides access to PCI and UDP interrupts from the controller.
- GCLIB_DEPRECATED GReturn GRecordRate (GCon g, double period_ms)

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

11.2.4.5.1 Detailed Description

Receive messages, interrupts, and data records.

11.2.4.5.2 Function Documentation

11.2.4.5.2.1 GRecord()

```
GCLIB_DEPRECATED GReturn 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.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_data_record()` instead.

Parameters

g	Connection's handle.
record	A pointer to the user's DataRecord union to hold the copy.
method	Determines the method for acquiring the data. • G_QR: QR is used via command-and-response. • G_DR: DR is used for asynchronous acquisition.

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 reecord, 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.

11.2.4.5.2.2 GMessage()

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

Provides access to unsolicited messages from the controller.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use gclib_message() instead.

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

g	Connection's handle.
buffer	The buffer to write the message data. The buffer will be null terminated.
buffer_len	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 am HMI or display panel. See MG and CF.

11.2.4.5.2.3 GInterrupt()

```
GCLIB_DEPRECATED GReturn GInterrupt (
            GCon g,
            GStatus * status_byte)
```

Provides access to PCI and UDP interrupts from the controller.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_interrupt()` instead.

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

g	Connection's handle.
status_byte	A pointer to a GStatus 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.

11.2.4.5.2.4 GRecordRate()

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

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

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use gclib_set_data_records() instead.

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

Parameters

g	Connection's handle.
period_ms	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.

11.2.5 Galil Connect

Host or connect to a remote gcaps instance.

Functions

- GCLIB_DEPRECATED GReturn GSetServer (GCStringIn server_name)

  Uses GUtility(), G_UTIL_GCAPS_SET_SERVER to set the new active server.
- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn GPublishServer (GCStringIn name, GOption publish, GOption save)

  Uses GUtility(), G_UTIL_GCAPS_PUBLISH_SERVER to publish local gcaps server to the local network.
- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn 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.

11.2.5.1 Detailed Description

Host or connect to a remote gcaps instance.

11.2.5.2 Function Documentation

11.2.5.2.1 GSetServer()

```
GCLIB_DEPRECATED GReturn GSetServer (
            GCStringIn server_name)
```

Uses GUtility(), G_UTIL_GCAPS_SET_SERVER to set the new active server.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use gclib_set_server() instead.

Note

This function is only available on Windows 10 and Linux.

Parameters

server_name	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.

11.2.5.2.2 GListServers()

```
GCLIB_DEPRECATED GReturn 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.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_list_servers()` instead.

Note

This function is only available on Windows 10 and Linux.

Parameters

servers	The buffer to hold the list of available gcaps servers
servers_len	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.

11.2.5.2.3 GPublishServer()

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

Uses GUtility(), G_UTIL_GCAPS_PUBLISH_SERVER to publish local gcaps server to the local network.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_set_published()` instead.

Note

This function is only available on Windows 10 and Linux.

Parameters

name	The name of the server to publish or remove
publish	Option to publish or remove server from network
save	Option to save this configuration for future reboots. This option is no longer functional as of gcaps 2.4.0 , since now gcaps always saves its configuration.

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"

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.

11.2.5.2.4 GServerStatus()

```
GCLIB_DEPRECATED GReturn 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.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release. Use `gclib_server()` instead.

Note

This function is only available on Windows 10 and Linux.

Parameters

status	The buffer to hold the status of the local gcaps server
status_len	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.

11.2.5.2.5 GRemoteConnections()

```
GCLIB_DEPRECATED GReturn 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.

Deprecated This function is part of the deprecated C (Legacy) API, which will be removed in a future release.

Note

This function is only available on Windows 10 and Linux.

Parameters

connections	The buffer to hold the list of remote IP addresses currently connected to your hardware
connections_len	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.

Chapter 12

Namespace Documentation

12.1 gclib Namespace Reference

Namespaces

- namespace internal

Classes

- class Controller

  Manages a connection to a controller.
- struct DataRecord
- class error

  Thrown when the underlying C library returns a nonzero `gclib_result`.
- struct Interrupt

Enumerations

- enum ConnectionType {
  Ethernet = GCLIB_ETHERNET ,
  Serial = GCLIB_SERIAL ,
  Pci = GCLIB_PCI }

  Returned by `Controller::connection_type()`.

Functions

- std::string version ()

  Installed gclib version.
- std::string addresses ()

  Show address, model, and serial of detected controllers, where available. Detected controllers are newline-separated. Each line is comma-separated.
- std::string ip_requests ()

  Show MAC address, model, and serial of all controllers requesting IP addresses. Detected controllers are newline-separated. Use `assign_ip()` to assign an IP address to controllers returned by this method.
- void assign_ip (std::string mac, std::string ip)
- void force_gcaps (bool on)

  Force future library calls to use either gcaps or direct connections.
- std::string gcaps_version ()

  Installed gcaps version, as reported by current gcaps server.
- std::string list_servers ()

  List available gcaps servers separated by newline.
- std::string server ()

Current gcaps server, or an empty string if connected to the local gcaps server.

- void set_server (std::string name={})
- std::string published ()

  Return name of local gcaps server, or an empty string if unpublished.
- void set_published (std::string name={})

  Publish local gcaps server as `name`, or unpublish if `name` is empty.

12.1.1 Enumeration Type Documentation

12.1.1.1 ConnectionType

`enum ConnectionType`
Returned by `Controller::connection_type()`.

Enumerator

Ethernet	
Serial	
Pci	

Definition at line 55 of file gclib.hpp.

12.1.2 Function Documentation

12.1.2.1 version()

`std::string version () [inline]`
Installed gclib version.
Definition at line 242 of file gclib.hpp.

12.1.2.2 addresses()

`std::string addresses () [inline]`
Show address, model, and serial of detected controllers, where available. Detected controllers are newline-separated. Each line is comma-separated.
Available information by connection type:

- Ethernet connections show address, model, and serial.
- USB connections show address and model.
- Serial and PCI connections show address only.

Example output:

```
192.168.0.40, DMC4040 Rev 1.3i, 10601
COM5, DMC-41x3
GALILPCI0
```

Returns

`std::string` Newline-separated info on detected controllers.

Definition at line 261 of file gclib.hpp.

12.1.2.3 ip_requests()

`std::string ip_requests () [inline]`
Show MAC address, model, and serial of all controllers requesting IP addresses. Detected controllers are newline-separated. Use `assign_ip()` to assign an IP address to controllers returned by this method.
Example output:

```
00:50:4c:20:29:69, DMC4000, 10601
00:50:4c:58:12:34, DMC2105, 4660
```

Definition at line 273 of file gclib.hpp.

12.1.2.4 assign_ip()

```
void assign_ip (
            std::string mac,
            std::string ip)  [inline]
```

Definition at line 274 of file gclib.hpp.

12.1.2.5 force_gcaps()

```
void force_gcaps (
            bool on)  [inline]
```

Force future library calls to use either gcaps or direct connections.
The default behavior of gclib is to prefer gcaps, but fall back to direct operations when gcaps is not running. This method allows disabling the default behavior process-wide, but cannot be undone.

Parameters

on	If `true`, all future gclib calls will fail if gcaps not found, rather than falling back to direct operations. If `false`, gcaps will not be used for any future gclib calls.

Definition at line 287 of file gclib.hpp.

12.1.2.6 gcaps_version()

```
std::string gcaps_version ()  [inline]
```

Installed gcaps version, as reported by current gcaps server.
Definition at line 289 of file gclib.hpp.

12.1.2.7 list_servers()

```
std::string list_servers ()  [inline]
```

List available gcaps servers separated by newline.
Definition at line 291 of file gclib.hpp.

12.1.2.8 server()

```
std::string server ()  [inline]
```

Current gcaps server, or an empty string if connected to the local gcaps server.
Definition at line 293 of file gclib.hpp.

12.1.2.9 set_server()

```
void set_server (
            std::string name = {})  [inline]
```

Use a gcaps server as reported by `list_servers()` for all future gcaps communications. Pass an empty string to use the local gcaps server.
Definition at line 296 of file gclib.hpp.

12.1.2.10 published()

```
std::string published ()  [inline]
```

Return name of local gcaps server, or an empty string if unpublished.
Definition at line 298 of file gclib.hpp.

12.1.2.11 set_published()

```
void set_published (
            std::string name = {})  [inline]
```

Publish local gcaps server as `name`, or unpublish if `name` is empty.
Definition at line 300 of file gclib.hpp.

12.2 gclib::internal Namespace Reference

Chapter 13

Class Documentation

13.1 DataRecord::Amp Struct Reference

Holds amplifier-specific data record fields. Provided by `DataRecord.amp()`.
`#include <gclib.hpp>`

Public Member Functions

- Amp (gclib_data_record_handle h, size_t amp)
- bool over_current ()
- bool over_voltage ()
- bool over_temp ()
- bool under_voltage ()
- bool electronic_lock_out ()

13.1.1 Detailed Description

Holds amplifier-specific data record fields. Provided by `DataRecord.amp()`.

Note

Amp errors are latched, and will need to be cleared with `AZ` once the underlying issue has been resolved. See the `command reference` for details.

Definition at line 129 of file gclib.hpp.

13.1.2 Constructor & Destructor Documentation

13.1.2.1 Amp()

```
Amp (
            gclib_data_record_handle h,
            size_t amp)  [inline]
```

Definition at line 130 of file gclib.hpp.

13.1.3 Member Function Documentation

13.1.3.1 over_current()

```
bool over_current ()  [inline]
```

Definition at line 131 of file gclib.hpp.

13.1.3.2 over_voltage()

```
bool over_voltage ()  [inline]
```

Definition at line 132 of file gclib.hpp.

13.1.3.3 over_temp()

```
bool over_temp () [inline]
```
Definition at line 133 of file gclib.hpp.

13.1.3.4 under_voltage()

```
bool under_voltage () [inline]
```
Definition at line 134 of file gclib.hpp.

13.1.3.5 electronic_lock_out()

```
bool electronic_lock_out () [inline]
```
Definition at line 135 of file gclib.hpp.
The documentation for this struct was generated from the following file:

- gclib.hpp

13.2 DataRecord::Axis Struct Reference

Holds axis-specific data record fields. Provided by `DataRecord.axis()`.
`#include <gclib.hpp>`

Public Types

- enum ModeOfMotion {
  NoMotion = GCLIB_NO_MOTION ,
  Contour = GCLIB_CONTOUR ,
  PositionAbsolute = GCLIB_POSITION_ABSOLUTE ,
  PositionRelative = GCLIB_POSITION_RELATIVE ,
  FindEdge = GCLIB_FIND_EDGE ,
  FindIndex = GCLIB_FIND_INDEX ,
  Home = GCLIB_HOME ,
  VectorMove = GCLIB_VECTOR_MOVE ,
  LinearMove = GCLIB_LINEAR_MOVE }

Public Member Functions

- Axis (gclib_data_record_handle h, char axis)
- bool motor_off ()
- bool latch_armed ()
- bool final_deceleration ()
- bool stopping ()
- bool slewing ()
- ModeOfMotion mode_of_motion ()
- bool negative_direction_move ()
- bool moving ()
- int home_phase ()
- bool stepper_mode ()
- bool home_input ()
- bool reverse_limit ()
- bool forward_limit ()
- bool latch_input ()
- bool latch_occurred ()
- uint8_t stop_code ()
- int32_t reference_position ()
- int32_t position ()
- int32_t position_error ()

- int32_t aux_position ()
- int32_t velocity ()
- double torque ()
- double analog_input ()
- int8_t halls ()
- int32_t variable ()
- bool hall_error ()
- bool peak_current ()

13.2.1 Detailed Description

Holds axis-specific data record fields. Provided by `DataRecord.axis()`.
Definition at line 141 of file gclib.hpp.

13.2.2 Member Enumeration Documentation

13.2.2.1 ModeOfMotion

```
enum ModeOfMotion
```

Enumerator

NoMotion	
Contour	
PositionAbsolute	
PositionRelative	
FindEdge	
FindIndex	
Home	
VectorMove	
LinearMove	

Definition at line 143 of file gclib.hpp.

13.2.3 Constructor & Destructor Documentation

13.2.3.1 Axis()

```
Axis (
            gclib_data_record_handle h,
            char axis)  [inline]
```

Definition at line 142 of file gclib.hpp.

13.2.4 Member Function Documentation

13.2.4.1 motor_off()

```
bool motor_off ()  [inline]
```

Definition at line 154 of file gclib.hpp.

13.2.4.2 latch_armed()

```
bool latch_armed ()  [inline]
```

Definition at line 155 of file gclib.hpp.

13.2.4.3 final_deceleration()

```
bool final_deceleration ()  [inline]
```

Definition at line 156 of file gclib.hpp.

13.2.4.4 stopping()

```
bool stopping () [inline]
```
Definition at line 157 of file gclib.hpp.

13.2.4.5 slewing()

```
bool slewing () [inline]
```
Definition at line 158 of file gclib.hpp.

13.2.4.6 mode_of_motion()

```
ModeOfMotion mode_of_motion () [inline]
```
Definition at line 159 of file gclib.hpp.

13.2.4.7 negative_direction_move()

```
bool negative_direction_move () [inline]
```
Definition at line 160 of file gclib.hpp.

13.2.4.8 moving()

```
bool moving () [inline]
```
Definition at line 161 of file gclib.hpp.

13.2.4.9 home_phase()

```
int home_phase () [inline]
```
Definition at line 162 of file gclib.hpp.

13.2.4.10 stepper_mode()

```
bool stepper_mode () [inline]
```
Definition at line 163 of file gclib.hpp.

13.2.4.11 home_input()

```
bool home_input () [inline]
```
Definition at line 164 of file gclib.hpp.

13.2.4.12 reverse_limit()

```
bool reverse_limit () [inline]
```
Definition at line 165 of file gclib.hpp.

13.2.4.13 forward_limit()

```
bool forward_limit () [inline]
```
Definition at line 166 of file gclib.hpp.

13.2.4.14 latch_input()

```
bool latch_input () [inline]
```
Definition at line 167 of file gclib.hpp.

13.2.4.15 latch_occurred()

```
bool latch_occurred () [inline]
```
Definition at line 168 of file gclib.hpp.

13.2.4.16 stop_code()

```
uint8_t stop_code () [inline]
```
Definition at line 169 of file gclib.hpp.

13.2.4.17 reference_position()

```
int32_t reference_position () [inline]
```
Definition at line 170 of file gclib.hpp.

13.2.4.18 position()

```
int32_t position () [inline]
```
Definition at line 171 of file gclib.hpp.

13.2.4.19 position_error()

```
int32_t position_error () [inline]
```
Definition at line 172 of file gclib.hpp.

13.2.4.20 aux_position()

```
int32_t aux_position () [inline]
```
Definition at line 173 of file gclib.hpp.

13.2.4.21 velocity()

```
int32_t velocity () [inline]
```
Definition at line 174 of file gclib.hpp.

13.2.4.22 torque()

```
double torque () [inline]
```
Definition at line 175 of file gclib.hpp.

13.2.4.23 analog_input()

```
double analog_input () [inline]
```
Definition at line 176 of file gclib.hpp.

13.2.4.24 halls()

```
int8_t halls () [inline]
```
Definition at line 177 of file gclib.hpp.

13.2.4.25 variable()

```
int32_t variable () [inline]
```
Definition at line 178 of file gclib.hpp.

13.2.4.26 hall_error()

```
bool hall_error () [inline]
```
Definition at line 179 of file gclib.hpp.

13.2.4.27 peak_current()

```
bool peak_current () [inline]
```
Definition at line 180 of file gclib.hpp.

The documentation for this struct was generated from the following file:

- gclib.hpp

13.3 Controller Class Reference

Manages a connection to a controller.
`#include <gclib.hpp>`

Public Member Functions

- Controller (std::string address)

  Open a connection to a controller reported by `addresses()`.
- Controller (std::string address, size_t baud_rate)

  Open a serial connection to a controller reported by `addresses()`.
- ~Controller ()
- void open ()

  Open the connection. If connection is already open, it will be closed and reopened.
- void close ()
- std::string address ()

  Returns the address that was used to connect to the controller.
- ConnectionType connection_type ()

  Returns the type of address that was used to connect to the controller.
- std::string revision_information ()

  Returns controller ^R^V output.
- uint32_t serial_number ()
- template<typename T = std::string>
  T command (std::string command)

  Issues a command to the controller and provides the response.
- std::string program ()
- void set_program (std::string program, std::string insert="")
- std::string array (std::string name, size_t start=0, size_t end=0)
- void set_array (std::string name, std::string array, size_t start=0, size_t end=0)
- void set_firmware (std::string file_path)
- void set_interrupts (Interrupt::Type mask, std::bitset< 8 > motion_complete_axes=0, std::bitset< 8 > digital_inputs=0)

  Configure which interrupts will be generated by the controller.
- void set_data_records (size_t period_ms)

  Configure controller data records.
- void subscribe_messages (std::function< void(std::string message)> callback)

  Subscribe to unsolicited messages.
- void subscribe_messages ()

  Subscribe to unsolicited messages.
- void subscribe_interrupts (std::function< void(Interrupt interrupt)> callback)

  Subscribe to interrupts.
- void subscribe_interrupts ()

  Subscribe to interrupts.
- void subscribe_data_records (std::function< void(DataRecord data_record)> callback)

  Subscribe to data records.
- void subscribe_data_records ()

  Subscribe to data records.
- void subscribe_progress (std::function< void(size_t current, size_t max)> callback)

  Register a progress callback.
- void unsubscribe_messages ()
- void unsubscribe_interrupts ()
- void unsubscribe_data_records ()
- void unsubscribe_progress ()
- std::string message (int timeout=-1)
- Interrupt interrupt (int timeout=-1)
- DataRecord data_record (int timeout=-1)

13.3.1 Detailed Description

Manages a connection to a controller.
Definition at line 303 of file gclib.hpp.

13.3.2 Constructor & Destructor Documentation

13.3.2.1 Controller() [1/2]

```
Controller (
            std::string address)  [inline]
```

Open a connection to a controller reported by `addresses()`.
Definition at line 306 of file gclib.hpp.

13.3.2.2 Controller() [2/2]

```
Controller (
            std::string address,
            size_t baud_rate)  [inline]
```

Open a serial connection to a controller reported by `addresses()`.
Definition at line 311 of file gclib.hpp.

13.3.2.3 ~Controller()

```
~Controller ()  [inline]
```

Definition at line 315 of file gclib.hpp.

13.3.3 Member Function Documentation

13.3.3.1 open()

```
void open ()  [inline]
```

Open the connection. If connection is already open, it will be closed and reopened.
Definition at line 318 of file gclib.hpp.

13.3.3.2 close()

```
void close ()  [inline]
```

Definition at line 324 of file gclib.hpp.

13.3.3.3 address()

```
std::string address ()  [inline]
```

Returns the address that was used to connect to the controller.
Definition at line 327 of file gclib.hpp.

13.3.3.4 connection_type()

```
ConnectionType connection_type ()  [inline]
```

Returns the type of address that was used to connect to the controller.
Definition at line 329 of file gclib.hpp.

13.3.3.5 revision_information()

```
std::string revision_information ()  [inline]
```

Returns controller ^R^V output.
Definition at line 332 of file gclib.hpp.

13.3.3.6 serial_number()

```
uint32_t serial_number ()  [inline]
```

Definition at line 333 of file gclib.hpp.

13.3.3.7 command()

```
template<typename T = std::string>
T command (
            std::string command)  [inline]
```

Issues a command to the controller and provides the response.
Note that a single command cannot be longer than 80 characters, similar to a program line.

Parameters

command	Command to be sent to the controller. Will be terminated with a carriage return.

Returns

The controller response.

Definition at line 345 of file gclib.hpp.

13.3.3.8 program()

```
std::string program ()  [inline]
```

Definition at line 354 of file gclib.hpp.

13.3.3.9 set_program()

```
void set_program (
            std::string program,
            std::string insert = "")  [inline]
```

Definition at line 355 of file gclib.hpp.

13.3.3.10 array()

```
std::string array (
            std::string name,
            size_t start = 0,
            size_t end = 0)  [inline]
```

Definition at line 356 of file gclib.hpp.

13.3.3.11 set_array()

```
void set_array (
            std::string name,
            std::string array,
            size_t start = 0,
            size_t end = 0)  [inline]
```

Definition at line 357 of file gclib.hpp.

13.3.3.12 set_firmware()

```
void set_firmware (
            std::string file_path)  [inline]
```

Definition at line 358 of file gclib.hpp.

13.3.3.13 set_interrupts()

```
void set_interrupts (
            Interrupt::Type mask,
            std::bitset< 8 > motion_complete_axes = 0,
            std::bitset< 8 > digital_inputs = 0)  [inline]
```

Configure which interrupts will be generated by the controller.
By default, the controller will not generate any interrupts.
Controllers can be configured to send only an exact list of desired interrupts, to avoid unnecessary processing and network traffic.
Example usage:

```
controller.set_interrupts(AllInterrupts); // Enable all interrupts.
controller.set_interrupts(
    MotionComplete | DigitalInputLow, // Only enable two interrupts
    {0, 7}, // Only trigger MotionComplete on axis A or F
    {0, 3}); // Only trigger DigitalInputLow on digital input 1 and 4.
controller.set_interrupts(); // Disable interrupts
```

See `subscribe_interrupts()` to enable receiving interrupts from the controller.

Parameters

interrupt_mask	Bitmask of desired interrupts. Use `::NoInterrupts` (`0`) to disable interrupts. Note that User Interrupts cannot be disabled.
axis_mask	When `::MotionComplete` is in `interrupt_mask`, this axis mask controls which axes will trigger a `::MotionComplete` interrupt. If `0` (default), all axes will be selected.
digital_input_mask	When `::DigitalInputLow` is in `interrupt_mask`, this bitmask selects which digital inputs will trigger an interrupt. If `0` (default), all digital inputs will be selected.

Definition at line 390 of file gclib.hpp.

13.3.3.14 set_data_records()

```
void set_data_records (
            size_t period_ms)  [inline]
```

Configure controller data records.
By default, data records are not sent by the controller.
See `subscribe_data_records()` to enable receiving data records from the controller.

Parameters

period_ms	Period (in milliseconds) between data records. Pass 0 (default) to disable data records.

Definition at line 403 of file gclib.hpp.

13.3.3.15 subscribe_messages() [1/2]

```
void subscribe_messages (
            std::function< void(std::string message)> callback)  [inline]
```

Subscribe to unsolicited messages.
This method registers a callback to be run every time an unsolicited message is received from the controller.

Note

The callback will be invoked on a separate, dedicated thread. Be careful when accessing any shared state.

On ethernet connections, a handle is opened for unsolicited data (if not already open) and the controller is configured to use it for messages.

Parameters

callback	A callback function, which will be invoked each time a message arrives. Replaces any previously set message callback.

Definition at line 420 of file gclib.hpp.

13.3.3.16 subscribe_messages() [2/2]

```
void subscribe_messages ()  [inline]
```

Subscribe to unsolicited messages.
This method begins queueing unsolicited messages for later retrieval by `message()`.
On ethernet connections, a handle is opened for unsolicited data (if not already open) and the controller is configured to use it for messages.
Definition at line 433 of file gclib.hpp.

13.3.3.17 subscribe_interrupts() [1/2]

```
void subscribe_interrupts (
            std::function< void(Interrupt interrupt)> callback)  [inline]
```

Subscribe to interrupts.
This method registers a callback to be run every time an interrupt is received from the controller. See `set_interrupts()` to enable and configure controller interrupts.

Note

The callback will be invoked on a separate, dedicated thread. Be careful when accessing any shared state.

On ethernet connections, a handle is opened for unsolicited data (if not already open) and the controller is configured to use it for interrupts.

Parameters

callback	A callback function, which will be invoked each time an interrupt arrives. Replaces any previously set message callback.

Definition at line 453 of file gclib.hpp.

13.3.3.18 subscribe_interrupts() [2/2]

```
void subscribe_interrupts ()  [inline]
```

Subscribe to interrupts.
This method begins queueing unsolicited interrupts for later retrieval by `interrupt()`.
See `set_interrupts()` to enable and configure controller interrupts.
On ethernet connections, a handle is opened for unsolicited data (if not already open) and the controller is configured to use it for interrupts.
Definition at line 469 of file gclib.hpp.

13.3.3.19 subscribe_data_records() [1/2]

```
void subscribe_data_records (
            std::function< void(DataRecord data_record)> callback)  [inline]
```

Subscribe to data records.
This method registers a callback to be run every time a data record is received from the controller. See `set_data_records()` to enable data records and configure how often they will be generated by the controller.

Note

The callback will be invoked on a separate, dedicated thread. Be careful when accessing any shared state.

On ethernet connections, a handle is opened for unsolicited data (if not already open) and the controller is configured to use it for data records.

Parameters

callback	A callback function, which will be invoked each time a data record arrives. Replaces any previously set message callback.

Definition at line 489 of file gclib.hpp.

13.3.3.20 subscribe_data_records() [2/2]

```
void subscribe_data_records ()  [inline]
```

Subscribe to data records.
This method begins queueing unsolicited interrupts for later retrieval by `data_record()`.
On ethernet connections, a handle is opened for unsolicited data (if not already open) and the controller is configured to use it for data records.
Definition at line 502 of file gclib.hpp.

13.3.3.21 subscribe_progress()

```
void subscribe_progress (
            std::function< void(size_t current, size_t max)> callback)  [inline]
```

Register a progress callback.
This method registers a callback to be run every time data is written to the controller during a `set_firmware`, `set_array`, or `set_program` call.

Note

The callback will be invoked on a separate, dedicated thread. Be careful when accessing any shared state.

Parameters

callback	A callback function, which will be invoked each time data is written to the controller. Replaces any previously set message callback.

Definition at line 518 of file gclib.hpp.

13.3.3.22 unsubscribe_messages()

```
void unsubscribe_messages ()  [inline]
```

Definition at line 523 of file gclib.hpp.

13.3.3.23 unsubscribe_interrupts()

```
void unsubscribe_interrupts ()  [inline]
```

Definition at line 524 of file gclib.hpp.

13.3.3.24 unsubscribe_data_records()

```
void unsubscribe_data_records ()  [inline]
```

Definition at line 525 of file gclib.hpp.

13.3.3.25 unsubscribe_progress()

```
void unsubscribe_progress ()  [inline]
```

Definition at line 526 of file gclib.hpp.

13.3.3.26 message()

```
std::string message (
            int timeout = -1)  [inline]
```

Get a queued message, or wait up to `timeout` ms for one to arrive.

Parameters

timeout	Timeout value in milliseconds. If `-1`, wait forever.

Definition at line 530 of file gclib.hpp.

13.3.3.27 interrupt()

```
Interrupt interrupt (
            int timeout = -1)  [inline]
```

Get a queued interrupt, or wait up to `timeout` ms for one to arrive.

Parameters

timeout	Timeout value in milliseconds. If `-1`, wait forever.

Definition at line 533 of file gclib.hpp.

13.3.3.28 data_record()

```
DataRecord data_record (
            int timeout = -1)  [inline]
```

Get a queued data record, or wait up to `timeout` ms for one to arrive.

Parameters

timeout	Timeout value in milliseconds. If `-1`, wait forever.

Definition at line 536 of file gclib.hpp.
The documentation for this class was generated from the following file:

- gclib.hpp

13.4 DataRecord::CoordinatedMove Struct Reference

Holds coordinated move specific data record fields. Provided by `DataRecord.coordinated_move()`.
`#include <gclib.hpp>`

Public Member Functions

- CoordinatedMove (gclib_data_record_handle h, char plane)
- uint16_t segment_count ()
- uint16_t status ()
- bool final_deceleration ()
- bool stopping ()
- bool slewing ()
- bool moving ()
- int32_t distance ()
- uint16_t buffer_available ()

13.4.1 Detailed Description

Holds coordinated move specific data record fields. Provided by `DataRecord.coordinated_move()`.
Definition at line 186 of file gclib.hpp.

13.4.2 Constructor & Destructor Documentation

13.4.2.1 CoordinatedMove()

```
CoordinatedMove (
            gclib_data_record_handle h,
            char plane)  [inline]
```

Definition at line 187 of file gclib.hpp.

13.4.3 Member Function Documentation

13.4.3.1 segment_count()

```
uint16_t segment_count ()  [inline]
```

Definition at line 188 of file gclib.hpp.

13.4.3.2 status()

```
uint16_t status ()  [inline]
```

Definition at line 189 of file gclib.hpp.

13.4.3.3 final_deceleration()

```
bool final_deceleration ()  [inline]
```

Definition at line 190 of file gclib.hpp.

13.4.3.4 stopping()

```
bool stopping ()  [inline]
```

Definition at line 191 of file gclib.hpp.

13.4.3.5 slewing()

```
bool slewing ()  [inline]
```

Definition at line 192 of file gclib.hpp.

13.4.3.6 moving()

```
bool moving ()  [inline]
```

Definition at line 193 of file gclib.hpp.

13.4.3.7 distance()

```
int32_t distance ()  [inline]
```

Definition at line 194 of file gclib.hpp.

13.4.3.8 buffer_available()

```
uint16_t buffer_available ()  [inline]
```

Definition at line 195 of file gclib.hpp.

The documentation for this struct was generated from the following file:

- gclib.hpp

13.5 DataRecord Struct Reference

```
#include <gclib.hpp>
```

Classes

- struct Amp

  Holds amplifier-specific data record fields. Provided by `DataRecord.amp()`.
- struct Axis

  Holds axis-specific data record fields. Provided by `DataRecord.axis()`.
- struct CoordinatedMove

  Holds coordinated move specific data record fields. Provided by `DataRecord.coordinated_move()`.

Public Types

- enum EthernetStatus {
  HandleFree = HANDLE_FREE ,
  UdpSlave = UDP_SLAVE ,
  TcpSlave = TCP_SLAVE ,
  UdpMaster = UDP_MASTER ,
  TcpMaster = TCP_MASTER ,
  EstablishingUdp = ESTABLISHING_UDP ,
  EstablishingTcp = ESTABLISHING_TCP }

Public Member Functions

- DataRecord (gclib_data_record_handle h)
- uint16_t sample ()
- bool input (size_t index)
- bool output (size_t index)
- std::bitset< 8 > input_bank (size_t index)
- std::bitset< 8 > output_bank (size_t index)
- uint8_t error_code ()
- uint32_t contour_segment_count ()
- uint16_t contour_buffer_available ()
- bool thread_running (size_t thread)
- EthernetStatus ethernet_status (char handle)
- std::vector< uint8_t > bytes ()

  Get data record as raw bytes.
- Amp amp (size_t amp)

  Get an instance of `DataRecord.Amp`, which provides amplifier-specific data record fields.
- Axis axis (char axis)

  Get an instance of `DataRecord.Axis`, which provides axis-specific data record fields.
- CoordinatedMove coordinated_move (char plane)

  Get an instance of `DataRecord.CoordinatedMove`, which provides CM-specific data record fields.

13.5.1 Detailed Description

Definition at line 115 of file gclib.hpp.

13.5.2 Member Enumeration Documentation

13.5.2.1 EthernetStatus

`enum EthernetStatus`

Enumerator

HandleFree	
UdpSlave	
TcpSlave	

Enumerator

UdpMaster	
TcpMaster	
EstablishingUdp	
EstablishingTcp	

Definition at line 117 of file gclib.hpp.

13.5.3 Constructor & Destructor Documentation

13.5.3.1 DataRecord()

```
DataRecord (
            gclib_data_record_handle h)  [inline]
```

Definition at line 116 of file gclib.hpp.

13.5.4 Member Function Documentation

13.5.4.1 sample()

```
uint16_t sample ()  [inline]
```

Definition at line 200 of file gclib.hpp.

13.5.4.2 input()

```
bool input (
            size_t index)  [inline]
```

Note

Inputs are active low, meaning this function returns `true` if current is NOT flowing.

Parameters

index	The index of the input to check. Note that index 0 corresponds to @IN[1].

Returns

`true` if input is active.

Definition at line 208 of file gclib.hpp.

13.5.4.3 output()

```
bool output (
            size_t index)  [inline]
```

Parameters

index	The index of the output to check. Note that index 0 corresponds to @OUT[1].

Returns

`true` if output is active.

Definition at line 213 of file gclib.hpp.

13.5.4.4 input_bank()

```
std::bitset< 8 > input_bank (
            size_t index)  [inline]
```

Definition at line 214 of file gclib.hpp.

13.5.4.5 output_bank()

```
std::bitset< 8 > output_bank (
            size_t index)  [inline]
```

Definition at line 215 of file gclib.hpp.

13.5.4.6 error_code()

```
uint8_t error_code ()  [inline]
```

Definition at line 216 of file gclib.hpp.

13.5.4.7 contour_segment_count()

```
uint32_t contour_segment_count ()  [inline]
```

Definition at line 217 of file gclib.hpp.

13.5.4.8 contour_buffer_available()

```
uint16_t contour_buffer_available ()  [inline]
```

Definition at line 218 of file gclib.hpp.

13.5.4.9 thread_running()

```
bool thread_running (
            size_t thread)  [inline]
```

Definition at line 219 of file gclib.hpp.

13.5.4.10 ethernet_status()

```
EthernetStatus ethernet_status (
            char handle)  [inline]
```

Parameters

handle	The ethernet handle to check, `A` - `H`.

Definition at line 223 of file gclib.hpp.

13.5.4.11 bytes()

```
std::vector< uint8_t > bytes ()  [inline]
```

Get data record as raw bytes.
Useful if your device has a nonstandard data record.
Definition at line 229 of file gclib.hpp.

13.5.4.12 amp()

```
Amp amp (
            size_t amp)  [inline]
```

Get an instance of `DataRecord.Amp`, which provides amplifier-specific data record fields.
Definition at line 234 of file gclib.hpp.

13.5.4.13 axis()

```
Axis axis (
            char axis)  [inline]
```

Get an instance of DataRecord.Axis, which provides axis-specific data record fields.
Definition at line 235 of file gclib.hpp.

13.5.4.14 coordinated_move()

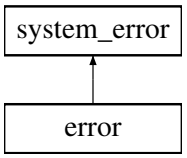
```
CoordinatedMove coordinated_move (
            char plane)  [inline]
```

Get an instance of DataRecord.CoordinatedMove, which provides CM-specific data record fields.
Definition at line 236 of file gclib.hpp.
The documentation for this struct was generated from the following file:

- gclib.hpp

13.6 error Class Reference

Thrown when the underlying C library returns a nonzero gclib_result.
`#include <gclib.hpp>`
Inheritance diagram for error:



13.6.1 Detailed Description

Thrown when the underlying C library returns a nonzero gclib_result.
Definition at line 49 of file gclib.hpp.
The documentation for this class was generated from the following file:

- gclib.hpp

13.7 gclib_interrupt_t Struct Reference

An interrupt generated by the controller.
`#include <gclib.h>`

Public Attributes

- gclib_interrupt_type type
- union {
  char axis
  If `type` *is* `GCLIB_MOTION_COMPLETE`*, holds the axis that triggered the interrupt.*
  uint8_t digital_input
  If `type` *is* `GCLIB_DIGITAL_INPUT_LOW`*, holds the digital input that triggered the interrupt.*
  uint8_t user_interrupt
  If `type` *is* `GCLIB_USER_INTERRUPT`*, holds the user interrupt that was triggered.*
  };
- uint8_t status
  The raw status byte generated by the controller. See the `EI command reference` *for all possible values.*

13.7.1 Detailed Description

An interrupt generated by the controller.
Definition at line 55 of file gclib.h.

13.7.2 Member Data Documentation

13.7.2.1 type

`gclib_interrupt_type type`
Definition at line 56 of file gclib.h.

13.7.2.2 axis

`char axis`
If `type` is `GCLIB_MOTION_COMPLETE`, holds the axis that triggered the interrupt.
Definition at line 58 of file gclib.h.

13.7.2.3 digital_input

`uint8_t digital_input`
If `type` is `GCLIB_DIGITAL_INPUT_LOW`, holds the digital input that triggered the interrupt.
Definition at line 59 of file gclib.h.

13.7.2.4 user_interrupt

`uint8_t user_interrupt`
If `type` is `GCLIB_USER_INTERRUPT`, holds the user interrupt that was triggered.
Definition at line 60 of file gclib.h.

13.7.2.5 [union]

`union { ... }`

13.7.2.6 status

`uint8_t status`
The raw status byte generated by the controller. See the `EI command reference` for all possible values.
Definition at line 62 of file gclib.h.
The documentation for this struct was generated from the following file:

- gclib.h

13.8 GDataRecord Union Reference

Data record union, containing all structs and a generic byte array accessor.
`#include <gclib_record.h>`

Public Attributes

- 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.

13.8.1 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 1078 of file gclib_record.h.

13.8.2 Member Data Documentation

13.8.2.1 dmc4000

`struct GDataRecord4000 dmc4000`

The DMC-4000 data record.
Definition at line 1080 of file gclib_record.h.

13.8.2.2 dmc4103

`struct GDataRecord4000 dmc4103`

The DMC-4103 data record.
Definition at line 1081 of file gclib_record.h.

13.8.2.3 dmc50000

`struct GDataRecord4000 dmc50000`

The DMC-50000 data record.
Definition at line 1082 of file gclib_record.h.

13.8.2.4 dmc52000

`struct GDataRecord52000 dmc52000`

The DMC-52000 data record.
Definition at line 1084 of file gclib_record.h.

13.8.2.5 dmc30000

`struct GDataRecord30000 dmc30000`
The DMC-30000 data record.
Definition at line 1086 of file gclib_record.h.

13.8.2.6 dmc2103

`struct GDataRecord2103 dmc2103`
The DMC-21x3 data record.
Definition at line 1088 of file gclib_record.h.

13.8.2.7 dmc1806

`struct GDataRecord1806 dmc1806`
The DMC-1806 data record.
Definition at line 1090 of file gclib_record.h.

13.8.2.8 dmc1802

`struct GDataRecord1802 dmc1802`
The DMC-1802 data record.
Definition at line 1092 of file gclib_record.h.

13.8.2.9 rio47000

`struct GDataRecord47000_ENC rio47000`
The RIO-471xx & 472xx data record, including encoder support.
Definition at line 1094 of file gclib_record.h.

13.8.2.10 rio47300

`struct GDataRecord47300_ENC rio47300`
The RIO 473xx data record, including encoder support.
Definition at line 1095 of file gclib_record.h.

13.8.2.11 rio47300_24ex

`struct GDataRecord47300_24EX rio47300_24ex`
The RIO 473xx data record, with 24EXOUT/24EXIN support.
Definition at line 1096 of file gclib_record.h.

13.8.2.12 rio47162

`struct GDataRecord47162 rio47162`
The RIO 47162 data record.
Definition at line 1097 of file gclib_record.h.

13.8.2.13 byte_array

`unsigned char byte_array[GALILDATARECORDMAXLENGTH]`
Generic byte array for offsets.
Definition at line 1099 of file gclib_record.h.
The documentation for this union was generated from the following file:

- gclib_record.h

13.9 GDataRecord1802 Struct Reference

`#include <gclib_record.h>`

Public Attributes

- 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.

13.9.1 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 723 of file gclib_record.h.

13.9.2 Member Data Documentation

13.9.2.1 sample_number

```
UW sample_number
```

sample number.
Definition at line 728 of file gclib_record.h.

13.9.2.2 input_bank_0

```
UB input_bank_0
```

general input bank 0 (inputs 1-8).
Definition at line 730 of file gclib_record.h.

13.9.2.3 input_bank_1

```
UB input_bank_1
```

general input bank 1 (inputs 9-16).
Definition at line 731 of file gclib_record.h.

13.9.2.4 input_bank_2

```
UB input_bank_2
```

general input bank 2 (inputs 17-24).
Definition at line 732 of file gclib_record.h.

13.9.2.5 input_bank_3

```
UB input_bank_3
```

general input bank 3 (inputs 25-32).
Definition at line 733 of file gclib_record.h.

13.9.2.6 input_bank_4

```
UB input_bank_4
```

general input bank 4 (inputs 33-40).
Definition at line 734 of file gclib_record.h.

13.9.2.7 input_bank_5

```
UB input_bank_5
```

general input bank 5 (inputs 41-48).
Definition at line 735 of file gclib_record.h.

13.9.2.8 input_bank_6

```
UB input_bank_6
```

general input bank 6 (inputs 49-56).
Definition at line 736 of file gclib_record.h.

13.9.2.9 input_bank_7

`UB input_bank_7`
general input bank 7 (inputs 57-64).
Definition at line 737 of file gclib_record.h.

13.9.2.10 input_bank_8

`UB input_bank_8`
general input bank 8 (inputs 65-72).
Definition at line 738 of file gclib_record.h.

13.9.2.11 input_bank_9

`UB input_bank_9`
general input bank 9 (inputs 73-80).
Definition at line 739 of file gclib_record.h.

13.9.2.12 output_bank_0

`UB output_bank_0`
general output bank 0 (outputs 1-8).
Definition at line 741 of file gclib_record.h.

13.9.2.13 output_bank_1

`UB output_bank_1`
general output bank 1 (outputs 9-16).
Definition at line 742 of file gclib_record.h.

13.9.2.14 output_bank_2

`UB output_bank_2`
general output bank 2 (outputs 17-24).
Definition at line 743 of file gclib_record.h.

13.9.2.15 output_bank_3

`UB output_bank_3`
general output bank 3 (outputs 25-32).
Definition at line 744 of file gclib_record.h.

13.9.2.16 output_bank_4

`UB output_bank_4`
general output bank 4 (outputs 33-40).
Definition at line 745 of file gclib_record.h.

13.9.2.17 output_bank_5

`UB output_bank_5`
general output bank 5 (outputs 41-48).
Definition at line 746 of file gclib_record.h.

13.9.2.18 output_bank_6

`UB output_bank_6`
general output bank 6 (outputs 49-56).
Definition at line 747 of file gclib_record.h.

13.9.2.19 output_bank_7

`UB output_bank_7`
general output bank 7 (outputs 57-64).
Definition at line 748 of file gclib_record.h.

13.9.2.20 output_bank_8

`UB output_bank_8`
general output bank 8 (outputs 65-72).
Definition at line 749 of file gclib_record.h.

13.9.2.21 output_bank_9

`UB output_bank_9`
general output bank 9 (outputs 73-80).
Definition at line 750 of file gclib_record.h.

13.9.2.22 error_code

`UB error_code`
error code.
Definition at line 752 of file gclib_record.h.

13.9.2.23 general_status

`UB general_status`
general status
Definition at line 753 of file gclib_record.h.

13.9.2.24 s_plane_segment_count

`UW s_plane_segment_count`
segment count of coordinated move for S plane.
Definition at line 755 of file gclib_record.h.

13.9.2.25 s_plane_move_status

`UW s_plane_move_status`
coordinated move status for S plane.
Definition at line 756 of file gclib_record.h.

13.9.2.26 s_distance

`SL s_distance`
distance traveled in coordinated move for S plane.
Definition at line 757 of file gclib_record.h.

13.9.2.27 t_plane_segment_count

`UW t_plane_segment_count`
segment count of coordinated move for T plane.
Definition at line 759 of file gclib_record.h.

13.9.2.28 t_plane_move_status

`UW t_plane_move_status`
Coordinated move status for T plane.
Definition at line 760 of file gclib_record.h.

13.9.2.29 t_distance

`SL t_distance`
distance traveled in coordinated move for T plane.
Definition at line 761 of file gclib_record.h.

13.9.2.30 axis_a_status

`UW axis_a_status`
A axis status.
Definition at line 763 of file gclib_record.h.

13.9.2.31 axis_a_switches

`UB axis_a_switches`
A axis switches.
Definition at line 764 of file gclib_record.h.

13.9.2.32 axis_a_stop_code

`UB axis_a_stop_code`
A axis stop code.
Definition at line 765 of file gclib_record.h.

13.9.2.33 axis_a_reference_position

`SL axis_a_reference_position`
A axis reference position.
Definition at line 766 of file gclib_record.h.

13.9.2.34 axis_a_motor_position

`SL axis_a_motor_position`
A axis motor position.
Definition at line 767 of file gclib_record.h.

13.9.2.35 axis_a_position_error

`SL axis_a_position_error`
A axis position error.
Definition at line 768 of file gclib_record.h.

13.9.2.36 axis_a_aux_position

`SL axis_a_aux_position`
A axis auxiliary position.
Definition at line 769 of file gclib_record.h.

13.9.2.37 axis_a_velocity

`SL axis_a_velocity`
A axis velocity.
Definition at line 770 of file gclib_record.h.

13.9.2.38 axis_a_torque

`SW axis_a_torque`
A axis torque.
Definition at line 771 of file gclib_record.h.

13.9.2.39 axis_a_reserved_0

`UB axis_a_reserved_0`
Reserved.
Definition at line 772 of file gclib_record.h.

13.9.2.40 axis_a_reserved_1

`UB axis_a_reserved_1`
Reserved.
Definition at line 773 of file gclib_record.h.

13.9.2.41 axis_b_status

`UW axis_b_status`
B axis status.
Definition at line 775 of file gclib_record.h.

13.9.2.42 axis_b_switches

`UB axis_b_switches`
B axis switches.
Definition at line 776 of file gclib_record.h.

13.9.2.43 axis_b_stop_code

`UB axis_b_stop_code`
B axis stop code.
Definition at line 777 of file gclib_record.h.

13.9.2.44 axis_b_reference_position

`SL axis_b_reference_position`
B axis reference position.
Definition at line 778 of file gclib_record.h.

13.9.2.45 axis_b_motor_position

`SL axis_b_motor_position`
B axis motor position.
Definition at line 779 of file gclib_record.h.

13.9.2.46 axis_b_position_error

`SL axis_b_position_error`
B axis position error.
Definition at line 780 of file gclib_record.h.

13.9.2.47 axis_b_aux_position

`SL axis_b_aux_position`
B axis auxiliary position.
Definition at line 781 of file gclib_record.h.

13.9.2.48 axis_b_velocity

`SL axis_b_velocity`
B axis velocity.
Definition at line 782 of file gclib_record.h.

13.9.2.49 axis_b_torque

`SW axis_b_torque`
B axis torque.
Definition at line 783 of file gclib_record.h.

13.9.2.50 axis_b_reserved_0

`UB axis_b_reserved_0`
Reserved.
Definition at line 784 of file gclib_record.h.

13.9.2.51 axis_b_reserved_1

`UB axis_b_reserved_1`
Reserved.
Definition at line 785 of file gclib_record.h.

13.9.2.52 axis_c_status

`UW axis_c_status`
C axis status.
Definition at line 787 of file gclib_record.h.

13.9.2.53 axis_c_switches

`UB axis_c_switches`
C axis switches.
Definition at line 788 of file gclib_record.h.

13.9.2.54 axis_c_stop_code

`UB axis_c_stop_code`
C axis stop code.
Definition at line 789 of file gclib_record.h.

13.9.2.55 axis_c_reference_position

`SL axis_c_reference_position`
C axis reference position.
Definition at line 790 of file gclib_record.h.

13.9.2.56 axis_c_motor_position

`SL axis_c_motor_position`
C axis motor position.
Definition at line 791 of file gclib_record.h.

13.9.2.57 axis_c_position_error

`SL axis_c_position_error`
C axis position error.
Definition at line 792 of file gclib_record.h.

13.9.2.58 axis_c_aux_position

`SL axis_c_aux_position`
C axis auxiliary position.
Definition at line 793 of file gclib_record.h.

13.9.2.59 axis_c_velocity

`SL axis_c_velocity`
C axis velocity.
Definition at line 794 of file gclib_record.h.

13.9.2.60 axis_c_torque

`SW axis_c_torque`
C axis torque.
Definition at line 795 of file gclib_record.h.

13.9.2.61 axis_c_reserved_0

`UB axis_c_reserved_0`
Reserved.
Definition at line 796 of file gclib_record.h.

13.9.2.62 axis_c_reserved_1

`UB axis_c_reserved_1`
Reserved.
Definition at line 797 of file gclib_record.h.

13.9.2.63 axis_d_status

`UW axis_d_status`
D axis status.
Definition at line 799 of file gclib_record.h.

13.9.2.64 axis_d_switches

`UB axis_d_switches`
D axis switches.
Definition at line 800 of file gclib_record.h.

13.9.2.65 axis_d_stop_code

`UB axis_d_stop_code`
D axis stop code.
Definition at line 801 of file gclib_record.h.

13.9.2.66 axis_d_reference_position

`SL axis_d_reference_position`
D axis reference position.
Definition at line 802 of file gclib_record.h.

13.9.2.67 axis_d_motor_position

`SL axis_d_motor_position`
D axis motor position.
Definition at line 803 of file gclib_record.h.

13.9.2.68 axis_d_position_error

`SL axis_d_position_error`
D axis position error.
Definition at line 804 of file gclib_record.h.

13.9.2.69 axis_d_aux_position

`SL axis_d_aux_position`
D axis auxiliary position.
Definition at line 805 of file gclib_record.h.

13.9.2.70 axis_d_velocity

`SL axis_d_velocity`
D axis velocity.
Definition at line 806 of file gclib_record.h.

13.9.2.71 axis_d_torque

`SW axis_d_torque`
D axis torque.
Definition at line 807 of file gclib_record.h.

13.9.2.72 axis_d_reserved_0

`UB axis_d_reserved_0`
Reserved.
Definition at line 808 of file gclib_record.h.

13.9.2.73 axis_d_reserved_1

`UB axis_d_reserved_1`
Reserved.
Definition at line 809 of file gclib_record.h.
The documentation for this struct was generated from the following file:

- gclib_record.h

13.10 GDataRecord1806 Struct Reference

Data record struct for DMC-1806 controller.
`#include <gclib_record.h>`

Public Attributes

- 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).

13.10.1 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 amplfifier status (bytes 52-55).
4. No axis-specific hall input status.

Definition at line 405 of file gclib_record.h.

13.10.2 Member Data Documentation

13.10.2.1 sample_number

`UW sample_number`
sample number.
Definition at line 409 of file gclib_record.h.

13.10.2.2 input_bank_0

`UB input_bank_0`
general input bank 0 (inputs 1-8).
Definition at line 411 of file gclib_record.h.

13.10.2.3 input_bank_1

`UB input_bank_1`
general input bank 1 (inputs 9-16).
Definition at line 412 of file gclib_record.h.

13.10.2.4 input_bank_2

`UB input_bank_2`
general input bank 2 (inputs 17-24).
Definition at line 413 of file gclib_record.h.

13.10.2.5 input_bank_3

`UB input_bank_3`
general input bank 3 (inputs 25-32).
Definition at line 414 of file gclib_record.h.

13.10.2.6 input_bank_4

`UB input_bank_4`
general input bank 4 (inputs 33-40).
Definition at line 415 of file gclib_record.h.

13.10.2.7 input_bank_5

`UB input_bank_5`
general input bank 5 (inputs 41-48).
Definition at line 416 of file gclib_record.h.

13.10.2.8 input_bank_6

`UB input_bank_6`
general input bank 6 (inputs 49-56).
Definition at line 417 of file gclib_record.h.

13.10.2.9 input_bank_7

`UB input_bank_7`
general input bank 7 (inputs 57-64).
Definition at line 418 of file gclib_record.h.

13.10.2.10 input_bank_8

`UB input_bank_8`
general input bank 8 (inputs 65-72).
Definition at line 419 of file gclib_record.h.

13.10.2.11 input_bank_9

`UB input_bank_9`
general input bank 9 (inputs 73-80).
Definition at line 420 of file gclib_record.h.

13.10.2.12 output_bank_0

`UB output_bank_0`
general output bank 0 (outputs 1-8).
Definition at line 422 of file gclib_record.h.

13.10.2.13 output_bank_1

`UB output_bank_1`
general output bank 1 (outputs 9-16).
Definition at line 423 of file gclib_record.h.

13.10.2.14 output_bank_2

`UB output_bank_2`
general output bank 2 (outputs 17-24).
Definition at line 424 of file gclib_record.h.

13.10.2.15 output_bank_3

`UB output_bank_3`
general output bank 3 (outputs 25-32).
Definition at line 425 of file gclib_record.h.

13.10.2.16 output_bank_4

`UB output_bank_4`
general output bank 4 (outputs 33-40).
Definition at line 426 of file gclib_record.h.

13.10.2.17 output_bank_5

`UB output_bank_5`
general output bank 5 (outputs 41-48).
Definition at line 427 of file gclib_record.h.

13.10.2.18 output_bank_6

`UB output_bank_6`
general output bank 6 (outputs 49-56).
Definition at line 428 of file gclib_record.h.

13.10.2.19 output_bank_7

`UB output_bank_7`
general output bank 7 (outputs 57-64).
Definition at line 429 of file gclib_record.h.

13.10.2.20 output_bank_8

`UB output_bank_8`
general output bank 8 (outputs 65-72).
Definition at line 430 of file gclib_record.h.

13.10.2.21 output_bank_9

`UB output_bank_9`
general output bank 9 (outputs 73-80).
Definition at line 431 of file gclib_record.h.

13.10.2.22 reserved_0

`SW reserved_0`
Reserved.
Definition at line 433 of file gclib_record.h.

13.10.2.23 reserved_2

`SW reserved_2`
Reserved.
Definition at line 434 of file gclib_record.h.

13.10.2.24 reserved_4

`SW reserved_4`
Reserved.
Definition at line 435 of file gclib_record.h.

13.10.2.25 reserved_6

`SW reserved_6`
Reserved.
Definition at line 436 of file gclib_record.h.

13.10.2.26 reserved_8

`SW reserved_8`
Reserved.
Definition at line 437 of file gclib_record.h.

13.10.2.27 reserved_10

`SW reserved_10`
Reserved.
Definition at line 438 of file gclib_record.h.

13.10.2.28 reserved_12

`SW reserved_12`
Reserved.
Definition at line 439 of file gclib_record.h.

13.10.2.29 reserved_14

`SW reserved_14`
Reserved.
Definition at line 440 of file gclib_record.h.

13.10.2.30 reserved_16

`UB reserved_16`
Reserved.
Definition at line 442 of file gclib_record.h.

13.10.2.31 reserved_17

`UB reserved_17`
Reserved.
Definition at line 443 of file gclib_record.h.

13.10.2.32 reserved_18

`UB reserved_18`
Reserved.
Definition at line 444 of file gclib_record.h.

13.10.2.33 reserved_19

`UB reserved_19`
Reserved.
Definition at line 445 of file gclib_record.h.

13.10.2.34 reserved_20

`UB reserved_20`
Reserved.
Definition at line 446 of file gclib_record.h.

13.10.2.35 reserved_21

`UB reserved_21`
Reserved.
Definition at line 447 of file gclib_record.h.

13.10.2.36 reserved_22

`UB reserved_22`
Reserved.
Definition at line 448 of file gclib_record.h.

13.10.2.37 reserved_23

`UB reserved_23`
Reserved.
Definition at line 449 of file gclib_record.h.

13.10.2.38 error_code

`UB error_code`
error code.
Definition at line 451 of file gclib_record.h.

13.10.2.39 thread_status

`UB thread_status`
thread status.
Definition at line 452 of file gclib_record.h.

13.10.2.40 reserved_24

`UL reserved_24`
Reserved.
Definition at line 453 of file gclib_record.h.

13.10.2.41 contour_segment_count

`UL contour_segment_count`
Segment Count for Contour Mode.
Definition at line 455 of file gclib_record.h.

13.10.2.42 contour_buffer_available

`UW contour_buffer_available`
Buffer space remaining, Contour Mode.
Definition at line 456 of file gclib_record.h.

13.10.2.43 s_plane_segment_count

`UW s_plane_segment_count`
segment count of coordinated move for S plane.
Definition at line 458 of file gclib_record.h.

13.10.2.44 s_plane_move_status

`UW s_plane_move_status`
coordinated move status for S plane.
Definition at line 459 of file gclib_record.h.

13.10.2.45 s_distance

`SL s_distance`
distance traveled in coordinated move for S plane.
Definition at line 460 of file gclib_record.h.

13.10.2.46 s_plane_buffer_available

`UW s_plane_buffer_available`
Buffer space remaining, S Plane.
Definition at line 461 of file gclib_record.h.

13.10.2.47 t_plane_segment_count

`UW t_plane_segment_count`
segment count of coordinated move for T plane.
Definition at line 463 of file gclib_record.h.

13.10.2.48 t_plane_move_status

`UW t_plane_move_status`
Coordinated move status for T plane.
Definition at line 464 of file gclib_record.h.

13.10.2.49 t_distance

`SL t_distance`
distance traveled in coordinated move for T plane.
Definition at line 465 of file gclib_record.h.

13.10.2.50 t_plane_buffer_available

`UW t_plane_buffer_available`
Buffer space remaining, T Plane.
Definition at line 466 of file gclib_record.h.

13.10.2.51 axis_a_status

`UW axis_a_status`
A axis status.
Definition at line 468 of file gclib_record.h.

13.10.2.52 axis_a_switches

`UB axis_a_switches`
A axis switches.
Definition at line 469 of file gclib_record.h.

13.10.2.53 axis_a_stop_code

`UB axis_a_stop_code`
A axis stop code.
Definition at line 470 of file gclib_record.h.

13.10.2.54 axis_a_reference_position

`SL axis_a_reference_position`
A axis reference position.
Definition at line 471 of file gclib_record.h.

13.10.2.55 axis_a_motor_position

`SL axis_a_motor_position`
A axis motor position.
Definition at line 472 of file gclib_record.h.

13.10.2.56 axis_a_position_error

`SL axis_a_position_error`
A axis position error.
Definition at line 473 of file gclib_record.h.

13.10.2.57 axis_a_aux_position

`SL axis_a_aux_position`
A axis auxiliary position.
Definition at line 474 of file gclib_record.h.

13.10.2.58 axis_a_velocity

`SL axis_a_velocity`
A axis velocity.
Definition at line 475 of file gclib_record.h.

13.10.2.59 axis_a_torque

`SL axis_a_torque`
A axis torque.
Definition at line 476 of file gclib_record.h.

13.10.2.60 axis_a_analog_in

`UW axis_a_analog_in`
A axis analog input.
Definition at line 477 of file gclib_record.h.

13.10.2.61 axis_a_reserved_0

`UB axis_a_reserved_0`
Reserved.
Definition at line 478 of file gclib_record.h.

13.10.2.62 axis_a_reserved_1

`UB axis_a_reserved_1`
Reserved.
Definition at line 479 of file gclib_record.h.

13.10.2.63 axis_a_variable

`SL axis_a_variable`
A User-defined variable (ZA).
Definition at line 480 of file gclib_record.h.

13.10.2.64 axis_b_status

`UW axis_b_status`
B axis status.
Definition at line 482 of file gclib_record.h.

13.10.2.65 axis_b_switches

`UB axis_b_switches`
B axis switches.
Definition at line 483 of file gclib_record.h.

13.10.2.66 axis_b_stop_code

`UB axis_b_stop_code`
B axis stop code.
Definition at line 484 of file gclib_record.h.

13.10.2.67 axis_b_reference_position

`SL axis_b_reference_position`
B axis reference position.
Definition at line 485 of file gclib_record.h.

13.10.2.68 axis_b_motor_position

`SL axis_b_motor_position`
B axis motor position.
Definition at line 486 of file gclib_record.h.

13.10.2.69 axis_b_position_error

`SL axis_b_position_error`
B axis position error.
Definition at line 487 of file gclib_record.h.

13.10.2.70 axis_b_aux_position

`SL axis_b_aux_position`
B axis auxiliary position.
Definition at line 488 of file gclib_record.h.

13.10.2.71 axis_b_velocity

`SL axis_b_velocity`
B axis velocity.
Definition at line 489 of file gclib_record.h.

13.10.2.72 axis_b_torque

`SL axis_b_torque`
B axis torque.
Definition at line 490 of file gclib_record.h.

13.10.2.73 axis_b_analog_in

`UW axis_b_analog_in`
B axis analog input.
Definition at line 491 of file gclib_record.h.

13.10.2.74 axis_b_reserved_0

`UB axis_b_reserved_0`
Reserved.
Definition at line 492 of file gclib_record.h.

13.10.2.75 axis_b_reserved_1

`UB axis_b_reserved_1`
Reserved.
Definition at line 493 of file gclib_record.h.

13.10.2.76 axis_b_variable

`SL axis_b_variable`
B User-defined variable (ZA).
Definition at line 494 of file gclib_record.h.

13.10.2.77 axis_c_status

`UW axis_c_status`
C axis status.
Definition at line 496 of file gclib_record.h.

13.10.2.78 axis_c_switches

`UB axis_c_switches`
C axis switches.
Definition at line 497 of file gclib_record.h.

13.10.2.79 axis_c_stop_code

`UB axis_c_stop_code`
C axis stop code.
Definition at line 498 of file gclib_record.h.

13.10.2.80 axis_c_reference_position

`SL axis_c_reference_position`
C axis reference position.
Definition at line 499 of file gclib_record.h.

13.10.2.81 axis_c_motor_position

SL axis_c_motor_position
C axis motor position.
Definition at line 500 of file gclib_record.h.

13.10.2.82 axis_c_position_error

SL axis_c_position_error
C axis position error.
Definition at line 501 of file gclib_record.h.

13.10.2.83 axis_c_aux_position

SL axis_c_aux_position
C axis auxiliary position.
Definition at line 502 of file gclib_record.h.

13.10.2.84 axis_c_velocity

SL axis_c_velocity
C axis velocity.
Definition at line 503 of file gclib_record.h.

13.10.2.85 axis_c_torque

SL axis_c_torque
C axis torque.
Definition at line 504 of file gclib_record.h.

13.10.2.86 axis_c_analog_in

UW axis_c_analog_in
C axis analog input.
Definition at line 505 of file gclib_record.h.

13.10.2.87 axis_c_reserved_0

UB axis_c_reserved_0
Reserved.
Definition at line 506 of file gclib_record.h.

13.10.2.88 axis_c_reserved_1

UB axis_c_reserved_1
Reserved.
Definition at line 507 of file gclib_record.h.

13.10.2.89 axis_c_variable

SL axis_c_variable
C User-defined variable (ZA).
Definition at line 508 of file gclib_record.h.

13.10.2.90 axis_d_status

UW axis_d_status
D axis status.
Definition at line 510 of file gclib_record.h.

13.10.2.91 axis_d_switches

UB axis_d_switches
D axis switches.
Definition at line 511 of file gclib_record.h.

13.10.2.92 axis_d_stop_code

UB axis_d_stop_code
D axis stop code.
Definition at line 512 of file gclib_record.h.

13.10.2.93 axis_d_reference_position

SL axis_d_reference_position
D axis reference position.
Definition at line 513 of file gclib_record.h.

13.10.2.94 axis_d_motor_position

SL axis_d_motor_position
D axis motor position.
Definition at line 514 of file gclib_record.h.

13.10.2.95 axis_d_position_error

SL axis_d_position_error
D axis position error.
Definition at line 515 of file gclib_record.h.

13.10.2.96 axis_d_aux_position

SL axis_d_aux_position
D axis auxiliary position.
Definition at line 516 of file gclib_record.h.

13.10.2.97 axis_d_velocity

SL axis_d_velocity
D axis velocity.
Definition at line 517 of file gclib_record.h.

13.10.2.98 axis_d_torque

SL axis_d_torque
D axis torque.
Definition at line 518 of file gclib_record.h.

13.10.2.99 axis_d_analog_in

UW axis_d_analog_in
D axis analog input.
Definition at line 519 of file gclib_record.h.

13.10.2.100 axis_d_reserved_0

UB axis_d_reserved_0
Reserved.
Definition at line 520 of file gclib_record.h.

13.10.2.101 axis_d_reserved_1

`UB axis_d_reserved_1`
Reserved.
Definition at line 521 of file gclib_record.h.

13.10.2.102 axis_d_variable

`SL axis_d_variable`
D User-defined variable (ZA).
Definition at line 522 of file gclib_record.h.

13.10.2.103 axis_e_status

`UW axis_e_status`
E axis status.
Definition at line 524 of file gclib_record.h.

13.10.2.104 axis_e_switches

`UB axis_e_switches`
E axis switches.
Definition at line 525 of file gclib_record.h.

13.10.2.105 axis_e_stop_code

`UB axis_e_stop_code`
E axis stop code.
Definition at line 526 of file gclib_record.h.

13.10.2.106 axis_e_reference_position

`SL axis_e_reference_position`
E axis reference position.
Definition at line 527 of file gclib_record.h.

13.10.2.107 axis_e_motor_position

`SL axis_e_motor_position`
E axis motor position.
Definition at line 528 of file gclib_record.h.

13.10.2.108 axis_e_position_error

`SL axis_e_position_error`
E axis position error.
Definition at line 529 of file gclib_record.h.

13.10.2.109 axis_e_aux_position

`SL axis_e_aux_position`
E axis auxiliary position.
Definition at line 530 of file gclib_record.h.

13.10.2.110 axis_e_velocity

`SL axis_e_velocity`
E axis velocity.
Definition at line 531 of file gclib_record.h.

13.10.2.111 axis_e_torque

SL axis_e_torque
E axis torque.
Definition at line 532 of file gclib_record.h.

13.10.2.112 axis_e_analog_in

UW axis_e_analog_in
E axis analog input.
Definition at line 533 of file gclib_record.h.

13.10.2.113 axis_e_reserved_0

UB axis_e_reserved_0
Reserved.
Definition at line 534 of file gclib_record.h.

13.10.2.114 axis_e_reserved_1

UB axis_e_reserved_1
Reserved.
Definition at line 535 of file gclib_record.h.

13.10.2.115 axis_e_variable

SL axis_e_variable
E User-defined variable (ZA).
Definition at line 536 of file gclib_record.h.

13.10.2.116 axis_f_status

UW axis_f_status
F axis status.
Definition at line 538 of file gclib_record.h.

13.10.2.117 axis_f_switches

UB axis_f_switches
F axis switches.
Definition at line 539 of file gclib_record.h.

13.10.2.118 axis_f_stop_code

UB axis_f_stop_code
F axis stop code.
Definition at line 540 of file gclib_record.h.

13.10.2.119 axis_f_reference_position

SL axis_f_reference_position
F axis reference position.
Definition at line 541 of file gclib_record.h.

13.10.2.120 axis_f_motor_position

SL axis_f_motor_position
F axis motor position.
Definition at line 542 of file gclib_record.h.

13.10.2.121 axis_f_position_error

`SL axis_f_position_error`
F axis position error.
Definition at line 543 of file gclib_record.h.

13.10.2.122 axis_f_aux_position

`SL axis_f_aux_position`
F axis auxiliary position.
Definition at line 544 of file gclib_record.h.

13.10.2.123 axis_f_velocity

`SL axis_f_velocity`
F axis velocity.
Definition at line 545 of file gclib_record.h.

13.10.2.124 axis_f_torque

`SL axis_f_torque`
F axis torque.
Definition at line 546 of file gclib_record.h.

13.10.2.125 axis_f_analog_in

`UW axis_f_analog_in`
F axis analog input.
Definition at line 547 of file gclib_record.h.

13.10.2.126 axis_f_reserved_0

`UB axis_f_reserved_0`
Reserved.
Definition at line 548 of file gclib_record.h.

13.10.2.127 axis_f_reserved_1

`UB axis_f_reserved_1`
Reserved.
Definition at line 549 of file gclib_record.h.

13.10.2.128 axis_f_variable

`SL axis_f_variable`
F User-defined variable (ZA).
Definition at line 550 of file gclib_record.h.

13.10.2.129 axis_g_status

`UW axis_g_status`
G axis status.
Definition at line 552 of file gclib_record.h.

13.10.2.130 axis_g_switches

`UB axis_g_switches`
G axis switches.
Definition at line 553 of file gclib_record.h.

13.10.2.131 axis_g_stop_code

`UB axis_g_stop_code`
G axis stop code.
Definition at line 554 of file gclib_record.h.

13.10.2.132 axis_g_reference_position

`SL axis_g_reference_position`
G axis reference position.
Definition at line 555 of file gclib_record.h.

13.10.2.133 axis_g_motor_position

`SL axis_g_motor_position`
G axis motor position.
Definition at line 556 of file gclib_record.h.

13.10.2.134 axis_g_position_error

`SL axis_g_position_error`
G axis position error.
Definition at line 557 of file gclib_record.h.

13.10.2.135 axis_g_aux_position

`SL axis_g_aux_position`
G axis auxiliary position.
Definition at line 558 of file gclib_record.h.

13.10.2.136 axis_g_velocity

`SL axis_g_velocity`
G axis velocity.
Definition at line 559 of file gclib_record.h.

13.10.2.137 axis_g_torque

`SL axis_g_torque`
G axis torque.
Definition at line 560 of file gclib_record.h.

13.10.2.138 axis_g_analog_in

`UW axis_g_analog_in`
G axis analog input.
Definition at line 561 of file gclib_record.h.

13.10.2.139 axis_g_reserved_0

`UB axis_g_reserved_0`
Reserved.
Definition at line 562 of file gclib_record.h.

13.10.2.140 axis_g_reserved_1

`UB axis_g_reserved_1`
Reserved.
Definition at line 563 of file gclib_record.h.

13.10.2.141 axis_g_variable

`SL axis_g_variable`
G User-defined variable (ZA).
Definition at line 564 of file gclib_record.h.

13.10.2.142 axis_h_status

`UW axis_h_status`
H axis status.
Definition at line 566 of file gclib_record.h.

13.10.2.143 axis_h_switches

`UB axis_h_switches`
H axis switches.
Definition at line 567 of file gclib_record.h.

13.10.2.144 axis_h_stop_code

`UB axis_h_stop_code`
H axis stop code.
Definition at line 568 of file gclib_record.h.

13.10.2.145 axis_h_reference_position

`SL axis_h_reference_position`
H axis reference position.
Definition at line 569 of file gclib_record.h.

13.10.2.146 axis_h_motor_position

`SL axis_h_motor_position`
H axis motor position.
Definition at line 570 of file gclib_record.h.

13.10.2.147 axis_h_position_error

`SL axis_h_position_error`
H axis position error.
Definition at line 571 of file gclib_record.h.

13.10.2.148 axis_h_aux_position

`SL axis_h_aux_position`
H axis auxiliary position.
Definition at line 572 of file gclib_record.h.

13.10.2.149 axis_h_velocity

`SL axis_h_velocity`
H axis velocity.
Definition at line 573 of file gclib_record.h.

13.10.2.150 axis_h_torque

`SL axis_h_torque`
H axis torque.
Definition at line 574 of file gclib_record.h.

13.10.2.151 axis_h_analog_in

`UW axis_h_analog_in`
H axis analog input.
Definition at line 575 of file gclib_record.h.

13.10.2.152 axis_h_reserved_0

`UB axis_h_reserved_0`
Reserved.
Definition at line 576 of file gclib_record.h.

13.10.2.153 axis_h_reserved_1

`UB axis_h_reserved_1`
Reserved.
Definition at line 577 of file gclib_record.h.

13.10.2.154 axis_h_variable

`SL axis_h_variable`
H User-defined variable (ZA).
Definition at line 578 of file gclib_record.h.
The documentation for this struct was generated from the following file:

- gclib_record.h

13.11 GDataRecord2103 Struct Reference

Data record struct for DMC-2103 controllers.
`#include <gclib_record.h>`

Public Attributes

- 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.

13.11.1 Detailed Description

Data record struct for DMC-2103 controllers.
Definition at line 582 of file gclib_record.h.

13.11.2 Member Data Documentation

13.11.2.1 header_0

`UB header_0`
1st Byte of Header.
Definition at line 587 of file gclib_record.h.

13.11.2.2 header_1

`UB header_1`
2nd Byte of Header.
Definition at line 588 of file gclib_record.h.

13.11.2.3 header_2

`UB header_2`
3rd Byte of Header.
Definition at line 589 of file gclib_record.h.

13.11.2.4 header_3

`UB header_3`
4th Byte of Header.
Definition at line 590 of file gclib_record.h.

13.11.2.5 sample_number

`UW sample_number`
sample number.
Definition at line 592 of file gclib_record.h.

13.11.2.6 input_bank_0

`UB input_bank_0`
general input bank 0 (inputs 1-8).
Definition at line 594 of file gclib_record.h.

13.11.2.7 input_bank_1

`UB input_bank_1`
general input bank 1 (inputs 9-16).
Definition at line 595 of file gclib_record.h.

13.11.2.8 input_bank_2

`UB input_bank_2`
general input bank 2 (inputs 17-24).
Definition at line 596 of file gclib_record.h.

13.11.2.9 input_bank_3

`UB input_bank_3`
general input bank 3 (inputs 25-32).
Definition at line 597 of file gclib_record.h.

13.11.2.10 input_bank_4

`UB input_bank_4`
general input bank 4 (inputs 33-40).
Definition at line 598 of file gclib_record.h.

13.11.2.11 input_bank_5

`UB input_bank_5`
general input bank 5 (inputs 41-48).
Definition at line 599 of file gclib_record.h.

13.11.2.12 input_bank_6

`UB input_bank_6`
general input bank 6 (inputs 49-56).
Definition at line 600 of file gclib_record.h.

13.11.2.13 input_bank_7

`UB input_bank_7`
general input bank 7 (inputs 57-64).
Definition at line 601 of file gclib_record.h.

13.11.2.14 input_bank_8

`UB input_bank_8`
general input bank 8 (inputs 65-72).
Definition at line 602 of file gclib_record.h.

13.11.2.15 input_bank_9

`UB input_bank_9`
general input bank 9 (inputs 73-80).
Definition at line 603 of file gclib_record.h.

13.11.2.16 output_bank_0

`UB output_bank_0`
general output bank 0 (outputs 1-8).
Definition at line 605 of file gclib_record.h.

13.11.2.17 output_bank_1

`UB output_bank_1`
general output bank 1 (outputs 9-16).
Definition at line 606 of file gclib_record.h.

13.11.2.18 output_bank_2

`UB output_bank_2`
general output bank 2 (outputs 17-24).
Definition at line 607 of file gclib_record.h.

13.11.2.19 output_bank_3

`UB output_bank_3`
general output bank 3 (outputs 25-32).
Definition at line 608 of file gclib_record.h.

13.11.2.20 output_bank_4

`UB output_bank_4`
general output bank 4 (outputs 33-40).
Definition at line 609 of file gclib_record.h.

13.11.2.21 output_bank_5

`UB output_bank_5`
general output bank 5 (outputs 41-48).
Definition at line 610 of file gclib_record.h.

13.11.2.22 output_bank_6

`UB output_bank_6`
general output bank 6 (outputs 49-56).
Definition at line 611 of file gclib_record.h.

13.11.2.23 output_bank_7

`UB output_bank_7`
general output bank 7 (outputs 57-64).
Definition at line 612 of file gclib_record.h.

13.11.2.24 output_bank_8

`UB output_bank_8`
general output bank 8 (outputs 65-72).
Definition at line 613 of file gclib_record.h.

13.11.2.25 output_bank_9

`UB output_bank_9`
general output bank 9 (outputs 73-80).
Definition at line 614 of file gclib_record.h.

13.11.2.26 error_code

`UB error_code`
error code.
Definition at line 616 of file gclib_record.h.

13.11.2.27 general_status

`UB general_status`
general status
Definition at line 617 of file gclib_record.h.

13.11.2.28 s_plane_segment_count

`UW s_plane_segment_count`
segment count of coordinated move for S plane.
Definition at line 619 of file gclib_record.h.

13.11.2.29 s_plane_move_status

`UW s_plane_move_status`
coordinated move status for S plane.
Definition at line 620 of file gclib_record.h.

13.11.2.30 s_distance

SL s_distance
distance traveled in coordinated move for S plane.
Definition at line 621 of file gclib_record.h.

13.11.2.31 t_plane_segment_count

UW t_plane_segment_count
segment count of coordinated move for T plane.
Definition at line 623 of file gclib_record.h.

13.11.2.32 t_plane_move_status

UW t_plane_move_status
Coordinated move status for T plane.
Definition at line 624 of file gclib_record.h.

13.11.2.33 t_distance

SL t_distance
distance traveled in coordinated move for T plane.
Definition at line 625 of file gclib_record.h.

13.11.2.34 axis_a_status

UW axis_a_status
A axis status.
Definition at line 627 of file gclib_record.h.

13.11.2.35 axis_a_switches

UB axis_a_switches
A axis switches.
Definition at line 628 of file gclib_record.h.

13.11.2.36 axis_a_stop_code

UB axis_a_stop_code
A axis stop code.
Definition at line 629 of file gclib_record.h.

13.11.2.37 axis_a_reference_position

SL axis_a_reference_position
A axis reference position.
Definition at line 630 of file gclib_record.h.

13.11.2.38 axis_a_motor_position

SL axis_a_motor_position
A axis motor position.
Definition at line 631 of file gclib_record.h.

13.11.2.39 axis_a_position_error

SL axis_a_position_error
A axis position error.
Definition at line 632 of file gclib_record.h.

13.11.2.40 axis_a_aux_position

`SL axis_a_aux_position`
A axis auxiliary position.
Definition at line 633 of file gclib_record.h.

13.11.2.41 axis_a_velocity

`SL axis_a_velocity`
A axis velocity.
Definition at line 634 of file gclib_record.h.

13.11.2.42 axis_a_torque

`SW axis_a_torque`
A axis torque.
Definition at line 635 of file gclib_record.h.

13.11.2.43 axis_a_analog_in

`UW axis_a_analog_in`
A axis analog input.
Definition at line 636 of file gclib_record.h.

13.11.2.44 axis_b_status

`UW axis_b_status`
B axis status.
Definition at line 638 of file gclib_record.h.

13.11.2.45 axis_b_switches

`UB axis_b_switches`
B axis switches.
Definition at line 639 of file gclib_record.h.

13.11.2.46 axis_b_stop_code

`UB axis_b_stop_code`
B axis stop code.
Definition at line 640 of file gclib_record.h.

13.11.2.47 axis_b_reference_position

`SL axis_b_reference_position`
B axis reference position.
Definition at line 641 of file gclib_record.h.

13.11.2.48 axis_b_motor_position

`SL axis_b_motor_position`
B axis motor position.
Definition at line 642 of file gclib_record.h.

13.11.2.49 axis_b_position_error

`SL axis_b_position_error`
B axis position error.
Definition at line 643 of file gclib_record.h.

13.11.2.50 axis_b_aux_position

SL axis_b_aux_position
B axis auxiliary position.
Definition at line 644 of file gclib_record.h.

13.11.2.51 axis_b_velocity

SL axis_b_velocity
B axis velocity.
Definition at line 645 of file gclib_record.h.

13.11.2.52 axis_b_torque

SW axis_b_torque
B axis torque.
Definition at line 646 of file gclib_record.h.

13.11.2.53 axis_b_analog_in

UW axis_b_analog_in
B axis analog input.
Definition at line 647 of file gclib_record.h.

13.11.2.54 axis_c_status

UW axis_c_status
C axis status.
Definition at line 649 of file gclib_record.h.

13.11.2.55 axis_c_switches

UB axis_c_switches
C axis switches.
Definition at line 650 of file gclib_record.h.

13.11.2.56 axis_c_stop_code

UB axis_c_stop_code
C axis stop code.
Definition at line 651 of file gclib_record.h.

13.11.2.57 axis_c_reference_position

SL axis_c_reference_position
C axis reference position.
Definition at line 652 of file gclib_record.h.

13.11.2.58 axis_c_motor_position

SL axis_c_motor_position
C axis motor position.
Definition at line 653 of file gclib_record.h.

13.11.2.59 axis_c_position_error

SL axis_c_position_error
C axis position error.
Definition at line 654 of file gclib_record.h.

13.11.2.60 axis_c_aux_position

`SL axis_c_aux_position`
C axis auxiliary position.
Definition at line 655 of file gclib_record.h.

13.11.2.61 axis_c_velocity

`SL axis_c_velocity`
C axis velocity.
Definition at line 656 of file gclib_record.h.

13.11.2.62 axis_c_torque

`SW axis_c_torque`
C axis torque.
Definition at line 657 of file gclib_record.h.

13.11.2.63 axis_c_analog_in

`UW axis_c_analog_in`
C axis analog input.
Definition at line 658 of file gclib_record.h.

13.11.2.64 axis_d_status

`UW axis_d_status`
D axis status.
Definition at line 660 of file gclib_record.h.

13.11.2.65 axis_d_switches

`UB axis_d_switches`
D axis switches.
Definition at line 661 of file gclib_record.h.

13.11.2.66 axis_d_stop_code

`UB axis_d_stop_code`
D axis stop code.
Definition at line 662 of file gclib_record.h.

13.11.2.67 axis_d_reference_position

`SL axis_d_reference_position`
D axis reference position.
Definition at line 663 of file gclib_record.h.

13.11.2.68 axis_d_motor_position

`SL axis_d_motor_position`
D axis motor position.
Definition at line 664 of file gclib_record.h.

13.11.2.69 axis_d_position_error

`SL axis_d_position_error`
D axis position error.
Definition at line 665 of file gclib_record.h.

13.11.2.70 axis_d_aux_position

SL axis_d_aux_position
D axis auxiliary position.
Definition at line 666 of file gclib_record.h.

13.11.2.71 axis_d_velocity

SL axis_d_velocity
D axis velocity.
Definition at line 667 of file gclib_record.h.

13.11.2.72 axis_d_torque

SW axis_d_torque
D axis torque.
Definition at line 668 of file gclib_record.h.

13.11.2.73 axis_d_analog_in

UW axis_d_analog_in
D axis analog input.
Definition at line 669 of file gclib_record.h.

13.11.2.74 axis_e_status

UW axis_e_status
E axis status.
Definition at line 671 of file gclib_record.h.

13.11.2.75 axis_e_switches

UB axis_e_switches
E axis switches.
Definition at line 672 of file gclib_record.h.

13.11.2.76 axis_e_stop_code

UB axis_e_stop_code
E axis stop code.
Definition at line 673 of file gclib_record.h.

13.11.2.77 axis_e_reference_position

SL axis_e_reference_position
E axis reference position.
Definition at line 674 of file gclib_record.h.

13.11.2.78 axis_e_motor_position

SL axis_e_motor_position
E axis motor position.
Definition at line 675 of file gclib_record.h.

13.11.2.79 axis_e_position_error

SL axis_e_position_error
E axis position error.
Definition at line 676 of file gclib_record.h.

13.11.2.80 axis_e_aux_position

`SL axis_e_aux_position`
E axis auxiliary position.
Definition at line 677 of file gclib_record.h.

13.11.2.81 axis_e_velocity

`SL axis_e_velocity`
E axis velocity.
Definition at line 678 of file gclib_record.h.

13.11.2.82 axis_e_torque

`SW axis_e_torque`
E axis torque.
Definition at line 679 of file gclib_record.h.

13.11.2.83 axis_e_analog_in

`UW axis_e_analog_in`
E axis analog input.
Definition at line 680 of file gclib_record.h.

13.11.2.84 axis_f_status

`UW axis_f_status`
F axis status.
Definition at line 682 of file gclib_record.h.

13.11.2.85 axis_f_switches

`UB axis_f_switches`
F axis switches.
Definition at line 683 of file gclib_record.h.

13.11.2.86 axis_f_stop_code

`UB axis_f_stop_code`
F axis stop code.
Definition at line 684 of file gclib_record.h.

13.11.2.87 axis_f_reference_position

`SL axis_f_reference_position`
F axis reference position.
Definition at line 685 of file gclib_record.h.

13.11.2.88 axis_f_motor_position

`SL axis_f_motor_position`
F axis motor position.
Definition at line 686 of file gclib_record.h.

13.11.2.89 axis_f_position_error

`SL axis_f_position_error`
F axis position error.
Definition at line 687 of file gclib_record.h.

13.11.2.90 axis_f_aux_position

`SL axis_f_aux_position`
F axis auxiliary position.
Definition at line 688 of file gclib_record.h.

13.11.2.91 axis_f_velocity

`SL axis_f_velocity`
F axis velocity.
Definition at line 689 of file gclib_record.h.

13.11.2.92 axis_f_torque

`SW axis_f_torque`
F axis torque.
Definition at line 690 of file gclib_record.h.

13.11.2.93 axis_f_analog_in

`UW axis_f_analog_in`
F axis analog input.
Definition at line 691 of file gclib_record.h.

13.11.2.94 axis_g_status

`UW axis_g_status`
G axis status.
Definition at line 693 of file gclib_record.h.

13.11.2.95 axis_g_switches

`UB axis_g_switches`
G axis switches.
Definition at line 694 of file gclib_record.h.

13.11.2.96 axis_g_stop_code

`UB axis_g_stop_code`
G axis stop code.
Definition at line 695 of file gclib_record.h.

13.11.2.97 axis_g_reference_position

`SL axis_g_reference_position`
G axis reference position.
Definition at line 696 of file gclib_record.h.

13.11.2.98 axis_g_motor_position

`SL axis_g_motor_position`
G axis motor position.
Definition at line 697 of file gclib_record.h.

13.11.2.99 axis_g_position_error

`SL axis_g_position_error`
G axis position error.
Definition at line 698 of file gclib_record.h.

13.11.2.100 axis_g_aux_position

`SL axis_g_aux_position`
G axis auxiliary position.
Definition at line 699 of file gclib_record.h.

13.11.2.101 axis_g_velocity

`SL axis_g_velocity`
G axis velocity.
Definition at line 700 of file gclib_record.h.

13.11.2.102 axis_g_torque

`SW axis_g_torque`
G axis torque.
Definition at line 701 of file gclib_record.h.

13.11.2.103 axis_g_analog_in

`UW axis_g_analog_in`
G axis analog input.
Definition at line 702 of file gclib_record.h.

13.11.2.104 axis_h_status

`UW axis_h_status`
H axis status.
Definition at line 704 of file gclib_record.h.

13.11.2.105 axis_h_switches

`UB axis_h_switches`
H axis switches.
Definition at line 705 of file gclib_record.h.

13.11.2.106 axis_h_stop_code

`UB axis_h_stop_code`
H axis stop code.
Definition at line 706 of file gclib_record.h.

13.11.2.107 axis_h_reference_position

`SL axis_h_reference_position`
H axis reference position.
Definition at line 707 of file gclib_record.h.

13.11.2.108 axis_h_motor_position

`SL axis_h_motor_position`
H axis motor position.
Definition at line 708 of file gclib_record.h.

13.11.2.109 axis_h_position_error

`SL axis_h_position_error`
H axis position error.
Definition at line 709 of file gclib_record.h.

13.11.2.110 axis_h_aux_position

SL axis_h_aux_position
H axis auxiliary position.
Definition at line 710 of file gclib_record.h.

13.11.2.111 axis_h_velocity

SL axis_h_velocity
H axis velocity.
Definition at line 711 of file gclib_record.h.

13.11.2.112 axis_h_torque

SW axis_h_torque
H axis torque.
Definition at line 712 of file gclib_record.h.

13.11.2.113 axis_h_analog_in

UW axis_h_analog_in
H axis analog input.
Definition at line 713 of file gclib_record.h.
The documentation for this struct was generated from the following file:

- gclib_record.h

13.12 GDataRecord30000 Struct Reference

Data record struct for DMC-30010 controllers.
`#include <gclib_record.h>`

Public Attributes

- 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).

13.12.1 Detailed Description

Data record struct for DMC-30010 controllers.
Definition at line 814 of file gclib_record.h.

13.12.2 Member Data Documentation

13.12.2.1 header_0

`UB header_0`
1st Byte of Header.
Definition at line 819 of file gclib_record.h.

13.12.2.2 header_1

`UB header_1`
2nd Byte of Header.
Definition at line 820 of file gclib_record.h.

13.12.2.3 header_2

`UB header_2`
3rd Byte of Header.
Definition at line 821 of file gclib_record.h.

13.12.2.4 header_3

`UB header_3`
4th Byte of Header.
Definition at line 822 of file gclib_record.h.

13.12.2.5 sample_number

`UW sample_number`
sample number.
Definition at line 824 of file gclib_record.h.

13.12.2.6 input_bank_0

`UB input_bank_0`
general input bank 0 (inputs 1-8).
Definition at line 826 of file gclib_record.h.

13.12.2.7 input_bank_1

`UB input_bank_1`
general input bank 1 (inputs 9-16).
Definition at line 827 of file gclib_record.h.

13.12.2.8 output_bank_0

`UB output_bank_0`
general output bank 0 (outputs 1-8).
Definition at line 829 of file gclib_record.h.

13.12.2.9 output_bank_1

`UB output_bank_1`
general output bank 1 (outputs 9-16).
Definition at line 830 of file gclib_record.h.

13.12.2.10 error_code

`UB error_code`
error code.
Definition at line 832 of file gclib_record.h.

13.12.2.11 thread_status

`UB thread_status`
thread status.
Definition at line 833 of file gclib_record.h.

13.12.2.12 input_analog_2

`UW input_analog_2`
Analog input 2. 1 is in axis data, see axis_a_analog_in.
Definition at line 835 of file gclib_record.h.

13.12.2.13 output_analog_1

`UW output_analog_1`
Analog output 1.
Definition at line 837 of file gclib_record.h.

13.12.2.14 output_analog_2

`UW output_analog_2`
Analog output 2.
Definition at line 838 of file gclib_record.h.

13.12.2.15 amplifier_status

`UL amplifier_status`
Amplifier Status.
Definition at line 840 of file gclib_record.h.

13.12.2.16 contour_segment_count

`UL contour_segment_count`
Segment Count for Contour Mode.
Definition at line 842 of file gclib_record.h.

13.12.2.17 contour_buffer_available

`UW contour_buffer_available`
Buffer space remaining, Contour Mode.
Definition at line 843 of file gclib_record.h.

13.12.2.18 s_plane_segment_count

`UW s_plane_segment_count`
segment count of coordinated move for S plane.
Definition at line 845 of file gclib_record.h.

13.12.2.19 s_plane_move_status

`UW s_plane_move_status`
coordinated move status for S plane.
Definition at line 846 of file gclib_record.h.

13.12.2.20 s_distance

`SL s_distance`
distance traveled in coordinated move for S plane.
Definition at line 847 of file gclib_record.h.

13.12.2.21 s_plane_buffer_available

`UW s_plane_buffer_available`
Buffer space remaining, S Plane.
Definition at line 848 of file gclib_record.h.

13.12.2.22 axis_a_status

`UW axis_a_status`
A axis status.
Definition at line 850 of file gclib_record.h.

13.12.2.23 axis_a_switches

`UB axis_a_switches`
A axis switches.
Definition at line 851 of file gclib_record.h.

13.12.2.24 axis_a_stop_code

`UB axis_a_stop_code`
A axis stop code.
Definition at line 852 of file gclib_record.h.

13.12.2.25 axis_a_reference_position

`SL axis_a_reference_position`
A axis reference position.
Definition at line 853 of file gclib_record.h.

13.12.2.26 axis_a_motor_position

`SL axis_a_motor_position`
A axis motor position.
Definition at line 854 of file gclib_record.h.

13.12.2.27 axis_a_position_error

`SL axis_a_position_error`
A axis position error.
Definition at line 855 of file gclib_record.h.

13.12.2.28 axis_a_aux_position

`SL axis_a_aux_position`
A axis auxiliary position.
Definition at line 856 of file gclib_record.h.

13.12.2.29 axis_a_velocity

`SL axis_a_velocity`
A axis velocity.
Definition at line 857 of file gclib_record.h.

13.12.2.30 axis_a_torque

`SL axis_a_torque`
A axis torque.
Definition at line 858 of file gclib_record.h.

13.12.2.31 axis_a_analog_in

`UW axis_a_analog_in`
A axis analog input.
Definition at line 859 of file gclib_record.h.

13.12.2.32 axis_a_halls

`UB axis_a_halls`
A Hall Input Status.
Definition at line 860 of file gclib_record.h.

13.12.2.33 axis_a_reserved

`UB axis_a_reserved`
Reserved.
Definition at line 861 of file gclib_record.h.

13.12.2.34 axis_a_variable

`SL axis_a_variable`
A User-defined variable (ZA).
Definition at line 862 of file gclib_record.h.
The documentation for this struct was generated from the following file:

- gclib_record.h

13.13 GDataRecord4000 Struct Reference

Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0.
`#include <gclib_record.h>`

Public Attributes

- 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).

13.13.1 Detailed Description

Data record struct for DMC-4000 controllers, including 4000, 4200, 4103, and 500x0.
Definition at line 31 of file gclib_record.h.

13.13.2 Member Data Documentation

13.13.2.1 header_0

```
UB header_0
```

1st Byte of Header.
Definition at line 36 of file gclib_record.h.

13.13.2.2 header_1

`UB header_1`
2nd Byte of Header.
Definition at line 37 of file gclib_record.h.

13.13.2.3 header_2

`UB header_2`
3rd Byte of Header.
Definition at line 38 of file gclib_record.h.

13.13.2.4 header_3

`UB header_3`
4th Byte of Header.
Definition at line 39 of file gclib_record.h.

13.13.2.5 sample_number

`UW sample_number`
sample number.
Definition at line 41 of file gclib_record.h.

13.13.2.6 input_bank_0

`UB input_bank_0`
general input bank 0 (inputs 1-8).
Definition at line 43 of file gclib_record.h.

13.13.2.7 input_bank_1

`UB input_bank_1`
general input bank 1 (inputs 9-16).
Definition at line 44 of file gclib_record.h.

13.13.2.8 input_bank_2

`UB input_bank_2`
general input bank 2 (inputs 17-24).
Definition at line 45 of file gclib_record.h.

13.13.2.9 input_bank_3

`UB input_bank_3`
general input bank 3 (inputs 25-32).
Definition at line 46 of file gclib_record.h.

13.13.2.10 input_bank_4

`UB input_bank_4`
general input bank 4 (inputs 33-40).
Definition at line 47 of file gclib_record.h.

13.13.2.11 input_bank_5

`UB input_bank_5`
general input bank 5 (inputs 41-48).
Definition at line 48 of file gclib_record.h.

13.13.2.12 input_bank_6

`UB input_bank_6`
general input bank 6 (inputs 49-56).
Definition at line 49 of file gclib_record.h.

13.13.2.13 input_bank_7

`UB input_bank_7`
general input bank 7 (inputs 57-64).
Definition at line 50 of file gclib_record.h.

13.13.2.14 input_bank_8

`UB input_bank_8`
general input bank 8 (inputs 65-72).
Definition at line 51 of file gclib_record.h.

13.13.2.15 input_bank_9

`UB input_bank_9`
general input bank 9 (inputs 73-80).
Definition at line 52 of file gclib_record.h.

13.13.2.16 output_bank_0

`UB output_bank_0`
general output bank 0 (outputs 1-8).
Definition at line 54 of file gclib_record.h.

13.13.2.17 output_bank_1

`UB output_bank_1`
general output bank 1 (outputs 9-16).
Definition at line 55 of file gclib_record.h.

13.13.2.18 output_bank_2

`UB output_bank_2`
general output bank 2 (outputs 17-24).
Definition at line 56 of file gclib_record.h.

13.13.2.19 output_bank_3

`UB output_bank_3`
general output bank 3 (outputs 25-32).
Definition at line 57 of file gclib_record.h.

13.13.2.20 output_bank_4

`UB output_bank_4`
general output bank 4 (outputs 33-40).
Definition at line 58 of file gclib_record.h.

13.13.2.21 output_bank_5

`UB output_bank_5`
general output bank 5 (outputs 41-48).
Definition at line 59 of file gclib_record.h.

13.13.2.22 output_bank_6

`UB output_bank_6`
general output bank 6 (outputs 49-56).
Definition at line 60 of file gclib_record.h.

13.13.2.23 output_bank_7

`UB output_bank_7`
general output bank 7 (outputs 57-64).
Definition at line 61 of file gclib_record.h.

13.13.2.24 output_bank_8

`UB output_bank_8`
general output bank 8 (outputs 65-72).
Definition at line 62 of file gclib_record.h.

13.13.2.25 output_bank_9

`UB output_bank_9`
general output bank 9 (outputs 73-80).
Definition at line 63 of file gclib_record.h.

13.13.2.26 reserved_0

`SW reserved_0`
Reserved.
Definition at line 65 of file gclib_record.h.

13.13.2.27 reserved_2

`SW reserved_2`
Reserved.
Definition at line 66 of file gclib_record.h.

13.13.2.28 reserved_4

`SW reserved_4`
Reserved.
Definition at line 67 of file gclib_record.h.

13.13.2.29 reserved_6

`SW reserved_6`
Reserved.
Definition at line 68 of file gclib_record.h.

13.13.2.30 reserved_8

`SW reserved_8`
Reserved.
Definition at line 69 of file gclib_record.h.

13.13.2.31 reserved_10

`SW reserved_10`
Reserved.
Definition at line 70 of file gclib_record.h.

13.13.2.32 reserved_12

`SW reserved_12`
Reserved.
Definition at line 71 of file gclib_record.h.

13.13.2.33 reserved_14

`SW reserved_14`
Reserved.
Definition at line 72 of file gclib_record.h.

13.13.2.34 ethernet_status_a

`UB ethernet_status_a`
Ethernet Handle A Status.
Definition at line 74 of file gclib_record.h.

13.13.2.35 ethernet_status_b

`UB ethernet_status_b`
Ethernet Handle B Status.
Definition at line 75 of file gclib_record.h.

13.13.2.36 ethernet_status_c

`UB ethernet_status_c`
Ethernet Handle C Status.
Definition at line 76 of file gclib_record.h.

13.13.2.37 ethernet_status_d

`UB ethernet_status_d`
Ethernet Handle D Status.
Definition at line 77 of file gclib_record.h.

13.13.2.38 ethernet_status_e

`UB ethernet_status_e`
Ethernet Handle E Status.
Definition at line 78 of file gclib_record.h.

13.13.2.39 ethernet_status_f

`UB ethernet_status_f`
Ethernet Handle F Status.
Definition at line 79 of file gclib_record.h.

13.13.2.40 ethernet_status_g

`UB ethernet_status_g`
Ethernet Handle G Status.
Definition at line 80 of file gclib_record.h.

13.13.2.41 ethernet_status_h

`UB ethernet_status_h`
Ethernet Handle H Status.
Definition at line 81 of file gclib_record.h.

13.13.2.42 error_code

`UB error_code`
error code.
Definition at line 83 of file gclib_record.h.

13.13.2.43 thread_status

`UB thread_status`
thread status
Definition at line 84 of file gclib_record.h.

13.13.2.44 amplifier_status

`UL amplifier_status`
Amplifier Status.
Definition at line 85 of file gclib_record.h.

13.13.2.45 contour_segment_count

`UL contour_segment_count`
Segment Count for Contour Mode.
Definition at line 87 of file gclib_record.h.

13.13.2.46 contour_buffer_available

`UW contour_buffer_available`
Buffer space remaining, Contour Mode.
Definition at line 88 of file gclib_record.h.

13.13.2.47 s_plane_segment_count

`UW s_plane_segment_count`
segment count of coordinated move for S plane.
Definition at line 90 of file gclib_record.h.

13.13.2.48 s_plane_move_status

`UW s_plane_move_status`
coordinated move status for S plane.
Definition at line 91 of file gclib_record.h.

13.13.2.49 s_distance

`SL s_distance`
distance traveled in coordinated move for S plane.
Definition at line 92 of file gclib_record.h.

13.13.2.50 s_plane_buffer_available

`UW s_plane_buffer_available`
Buffer space remaining, S Plane.
Definition at line 93 of file gclib_record.h.

13.13.2.51 t_plane_segment_count

`UW t_plane_segment_count`
segment count of coordinated move for T plane.
Definition at line 95 of file gclib_record.h.

13.13.2.52 t_plane_move_status

`UW t_plane_move_status`
Coordinated move status for T plane.
Definition at line 96 of file gclib_record.h.

13.13.2.53 t_distance

`SL t_distance`
distance traveled in coordinated move for T plane.
Definition at line 97 of file gclib_record.h.

13.13.2.54 t_plane_buffer_available

`UW t_plane_buffer_available`
Buffer space remaining, T Plane.
Definition at line 98 of file gclib_record.h.

13.13.2.55 axis_a_status

`UW axis_a_status`
A axis status.
Definition at line 100 of file gclib_record.h.

13.13.2.56 axis_a_switches

`UB axis_a_switches`
A axis switches.
Definition at line 101 of file gclib_record.h.

13.13.2.57 axis_a_stop_code

`UB axis_a_stop_code`
A axis stop code.
Definition at line 102 of file gclib_record.h.

13.13.2.58 axis_a_reference_position

`SL axis_a_reference_position`
A axis reference position.
Definition at line 103 of file gclib_record.h.

13.13.2.59 axis_a_motor_position

`SL axis_a_motor_position`
A axis motor position.
Definition at line 104 of file gclib_record.h.

13.13.2.60 axis_a_position_error

`SL axis_a_position_error`
A axis position error.
Definition at line 105 of file gclib_record.h.

13.13.2.61 axis_a_aux_position

`SL axis_a_aux_position`
A axis auxiliary position.
Definition at line 106 of file gclib_record.h.

13.13.2.62 axis_a_velocity

`SL axis_a_velocity`
A axis velocity.
Definition at line 107 of file gclib_record.h.

13.13.2.63 axis_a_torque

`SL axis_a_torque`
A axis torque.
Definition at line 108 of file gclib_record.h.

13.13.2.64 axis_a_analog_in

`UW axis_a_analog_in`
A axis analog input.
Definition at line 109 of file gclib_record.h.

13.13.2.65 axis_a_halls

`UB axis_a_halls`
A Hall Input Status.
Definition at line 110 of file gclib_record.h.

13.13.2.66 axis_a_reserved

`UB axis_a_reserved`
Reserved.
Definition at line 111 of file gclib_record.h.

13.13.2.67 axis_a_variable

`SL axis_a_variable`
A User-defined variable (ZA).
Definition at line 112 of file gclib_record.h.

13.13.2.68 axis_b_status

`UW axis_b_status`
B axis status.
Definition at line 114 of file gclib_record.h.

13.13.2.69 axis_b_switches

`UB axis_b_switches`
B axis switches.
Definition at line 115 of file gclib_record.h.

13.13.2.70 axis_b_stop_code

`UB axis_b_stop_code`
B axis stop code.
Definition at line 116 of file gclib_record.h.

13.13.2.71 axis_b_reference_position

`SL axis_b_reference_position`
B axis reference position.
Definition at line 117 of file gclib_record.h.

13.13.2.72 axis_b_motor_position

`SL axis_b_motor_position`
B axis motor position.
Definition at line 118 of file gclib_record.h.

13.13.2.73 axis_b_position_error

`SL axis_b_position_error`
B axis position error.
Definition at line 119 of file gclib_record.h.

13.13.2.74 axis_b_aux_position

`SL axis_b_aux_position`
B axis auxiliary position.
Definition at line 120 of file gclib_record.h.

13.13.2.75 axis_b_velocity

`SL axis_b_velocity`
B axis velocity.
Definition at line 121 of file gclib_record.h.

13.13.2.76 axis_b_torque

`SL axis_b_torque`
B axis torque.
Definition at line 122 of file gclib_record.h.

13.13.2.77 axis_b_analog_in

`UW axis_b_analog_in`
B axis analog input.
Definition at line 123 of file gclib_record.h.

13.13.2.78 axis_b_halls

`UB axis_b_halls`
B Hall Input Status.
Definition at line 124 of file gclib_record.h.

13.13.2.79 axis_b_reserved

`UB axis_b_reserved`
Reserved.
Definition at line 125 of file gclib_record.h.

13.13.2.80 axis_b_variable

`SL axis_b_variable`
B User-defined variable (ZA).
Definition at line 126 of file gclib_record.h.

13.13.2.81 axis_c_status

`UW axis_c_status`
C axis status.
Definition at line 128 of file gclib_record.h.

13.13.2.82 axis_c_switches

`UB axis_c_switches`
C axis switches.
Definition at line 129 of file gclib_record.h.

13.13.2.83 axis_c_stop_code

`UB axis_c_stop_code`
C axis stop code.
Definition at line 130 of file gclib_record.h.

13.13.2.84 axis_c_reference_position

`SL axis_c_reference_position`
C axis reference position.
Definition at line 131 of file gclib_record.h.

13.13.2.85 axis_c_motor_position

`SL axis_c_motor_position`
C axis motor position.
Definition at line 132 of file gclib_record.h.

13.13.2.86 axis_c_position_error

`SL axis_c_position_error`
C axis position error.
Definition at line 133 of file gclib_record.h.

13.13.2.87 axis_c_aux_position

`SL axis_c_aux_position`
C axis auxiliary position.
Definition at line 134 of file gclib_record.h.

13.13.2.88 axis_c_velocity

`SL axis_c_velocity`
C axis velocity.
Definition at line 135 of file gclib_record.h.

13.13.2.89 axis_c_torque

`SL axis_c_torque`
C axis torque.
Definition at line 136 of file gclib_record.h.

13.13.2.90 axis_c_analog_in

`UW axis_c_analog_in`
C axis analog input.
Definition at line 137 of file gclib_record.h.

13.13.2.91 axis_c_halls

`UB axis_c_halls`
C Hall Input Status.
Definition at line 138 of file gclib_record.h.

13.13.2.92 axis_c_reserved

UB axis_c_reserved
Reserved.
Definition at line 139 of file gclib_record.h.

13.13.2.93 axis_c_variable

SL axis_c_variable
C User-defined variable (ZA).
Definition at line 140 of file gclib_record.h.

13.13.2.94 axis_d_status

UW axis_d_status
D axis status.
Definition at line 142 of file gclib_record.h.

13.13.2.95 axis_d_switches

UB axis_d_switches
D axis switches.
Definition at line 143 of file gclib_record.h.

13.13.2.96 axis_d_stop_code

UB axis_d_stop_code
D axis stop code.
Definition at line 144 of file gclib_record.h.

13.13.2.97 axis_d_reference_position

SL axis_d_reference_position
D axis reference position.
Definition at line 145 of file gclib_record.h.

13.13.2.98 axis_d_motor_position

SL axis_d_motor_position
D axis motor position.
Definition at line 146 of file gclib_record.h.

13.13.2.99 axis_d_position_error

SL axis_d_position_error
D axis position error.
Definition at line 147 of file gclib_record.h.

13.13.2.100 axis_d_aux_position

SL axis_d_aux_position
D axis auxiliary position.
Definition at line 148 of file gclib_record.h.

13.13.2.101 axis_d_velocity

SL axis_d_velocity
D axis velocity.
Definition at line 149 of file gclib_record.h.

13.13.2.102 axis_d_torque

SL axis_d_torque
D axis torque.
Definition at line 150 of file gclib_record.h.

13.13.2.103 axis_d_analog_in

UW axis_d_analog_in
D axis analog input.
Definition at line 151 of file gclib_record.h.

13.13.2.104 axis_d_halls

UB axis_d_halls
D Hall Input Status.
Definition at line 152 of file gclib_record.h.

13.13.2.105 axis_d_reserved

UB axis_d_reserved
Reserved.
Definition at line 153 of file gclib_record.h.

13.13.2.106 axis_d_variable

SL axis_d_variable
D User-defined variable (ZA).
Definition at line 154 of file gclib_record.h.

13.13.2.107 axis_e_status

UW axis_e_status
E axis status.
Definition at line 156 of file gclib_record.h.

13.13.2.108 axis_e_switches

UB axis_e_switches
E axis switches.
Definition at line 157 of file gclib_record.h.

13.13.2.109 axis_e_stop_code

UB axis_e_stop_code
E axis stop code.
Definition at line 158 of file gclib_record.h.

13.13.2.110 axis_e_reference_position

SL axis_e_reference_position
E axis reference position.
Definition at line 159 of file gclib_record.h.

13.13.2.111 axis_e_motor_position

SL axis_e_motor_position
E axis motor position.
Definition at line 160 of file gclib_record.h.

13.13.2.112 axis_e_position_error

`SL axis_e_position_error`
E axis position error.
Definition at line 161 of file gclib_record.h.

13.13.2.113 axis_e_aux_position

`SL axis_e_aux_position`
E axis auxiliary position.
Definition at line 162 of file gclib_record.h.

13.13.2.114 axis_e_velocity

`SL axis_e_velocity`
E axis velocity.
Definition at line 163 of file gclib_record.h.

13.13.2.115 axis_e_torque

`SL axis_e_torque`
E axis torque.
Definition at line 164 of file gclib_record.h.

13.13.2.116 axis_e_analog_in

`UW axis_e_analog_in`
E axis analog input.
Definition at line 165 of file gclib_record.h.

13.13.2.117 axis_e_halls

`UB axis_e_halls`
E Hall Input Status.
Definition at line 166 of file gclib_record.h.

13.13.2.118 axis_e_reserved

`UB axis_e_reserved`
Reserved.
Definition at line 167 of file gclib_record.h.

13.13.2.119 axis_e_variable

`SL axis_e_variable`
E User-defined variable (ZA).
Definition at line 168 of file gclib_record.h.

13.13.2.120 axis_f_status

`UW axis_f_status`
F axis status.
Definition at line 170 of file gclib_record.h.

13.13.2.121 axis_f_switches

`UB axis_f_switches`
F axis switches.
Definition at line 171 of file gclib_record.h.

13.13.2.122 axis_f_stop_code

`UB axis_f_stop_code`
F axis stop code.
Definition at line 172 of file gclib_record.h.

13.13.2.123 axis_f_reference_position

`SL axis_f_reference_position`
F axis reference position.
Definition at line 173 of file gclib_record.h.

13.13.2.124 axis_f_motor_position

`SL axis_f_motor_position`
F axis motor position.
Definition at line 174 of file gclib_record.h.

13.13.2.125 axis_f_position_error

`SL axis_f_position_error`
F axis position error.
Definition at line 175 of file gclib_record.h.

13.13.2.126 axis_f_aux_position

`SL axis_f_aux_position`
F axis auxiliary position.
Definition at line 176 of file gclib_record.h.

13.13.2.127 axis_f_velocity

`SL axis_f_velocity`
F axis velocity.
Definition at line 177 of file gclib_record.h.

13.13.2.128 axis_f_torque

`SL axis_f_torque`
F axis torque.
Definition at line 178 of file gclib_record.h.

13.13.2.129 axis_f_analog_in

`UW axis_f_analog_in`
F axis analog input.
Definition at line 179 of file gclib_record.h.

13.13.2.130 axis_f_halls

`UB axis_f_halls`
F Hall Input Status.
Definition at line 180 of file gclib_record.h.

13.13.2.131 axis_f_reserved

`UB axis_f_reserved`
Reserved.
Definition at line 181 of file gclib_record.h.

13.13.2.132 axis_f_variable

`SL axis_f_variable`
F User-defined variable (ZA).
Definition at line 182 of file gclib_record.h.

13.13.2.133 axis_g_status

`UW axis_g_status`
G axis status.
Definition at line 184 of file gclib_record.h.

13.13.2.134 axis_g_switches

`UB axis_g_switches`
G axis switches.
Definition at line 185 of file gclib_record.h.

13.13.2.135 axis_g_stop_code

`UB axis_g_stop_code`
G axis stop code.
Definition at line 186 of file gclib_record.h.

13.13.2.136 axis_g_reference_position

`SL axis_g_reference_position`
G axis reference position.
Definition at line 187 of file gclib_record.h.

13.13.2.137 axis_g_motor_position

`SL axis_g_motor_position`
G axis motor position.
Definition at line 188 of file gclib_record.h.

13.13.2.138 axis_g_position_error

`SL axis_g_position_error`
G axis position error.
Definition at line 189 of file gclib_record.h.

13.13.2.139 axis_g_aux_position

`SL axis_g_aux_position`
G axis auxiliary position.
Definition at line 190 of file gclib_record.h.

13.13.2.140 axis_g_velocity

`SL axis_g_velocity`
G axis velocity.
Definition at line 191 of file gclib_record.h.

13.13.2.141 axis_g_torque

`SL axis_g_torque`
G axis torque.
Definition at line 192 of file gclib_record.h.

13.13.2.142 axis_g_analog_in

`UW axis_g_analog_in`
G axis analog input.
Definition at line 193 of file gclib_record.h.

13.13.2.143 axis_g_halls

`UB axis_g_halls`
G Hall Input Status.
Definition at line 194 of file gclib_record.h.

13.13.2.144 axis_g_reserved

`UB axis_g_reserved`
Reserved.
Definition at line 195 of file gclib_record.h.

13.13.2.145 axis_g_variable

`SL axis_g_variable`
G User-defined variable (ZA).
Definition at line 196 of file gclib_record.h.

13.13.2.146 axis_h_status

`UW axis_h_status`
H axis status.
Definition at line 198 of file gclib_record.h.

13.13.2.147 axis_h_switches

`UB axis_h_switches`
H axis switches.
Definition at line 199 of file gclib_record.h.

13.13.2.148 axis_h_stop_code

`UB axis_h_stop_code`
H axis stop code.
Definition at line 200 of file gclib_record.h.

13.13.2.149 axis_h_reference_position

`SL axis_h_reference_position`
H axis reference position.
Definition at line 201 of file gclib_record.h.

13.13.2.150 axis_h_motor_position

`SL axis_h_motor_position`
H axis motor position.
Definition at line 202 of file gclib_record.h.

13.13.2.151 axis_h_position_error

`SL axis_h_position_error`
H axis position error.
Definition at line 203 of file gclib_record.h.

13.13.2.152 axis_h_aux_position

SL axis_h_aux_position
H axis auxiliary position.
Definition at line 204 of file gclib_record.h.

13.13.2.153 axis_h_velocity

SL axis_h_velocity
H axis velocity.
Definition at line 205 of file gclib_record.h.

13.13.2.154 axis_h_torque

SL axis_h_torque
H axis torque.
Definition at line 206 of file gclib_record.h.

13.13.2.155 axis_h_analog_in

UW axis_h_analog_in
H axis analog input.
Definition at line 207 of file gclib_record.h.

13.13.2.156 axis_h_halls

UB axis_h_halls
H Hall Input Status.
Definition at line 208 of file gclib_record.h.

13.13.2.157 axis_h_reserved

UB axis_h_reserved
Reserved.
Definition at line 209 of file gclib_record.h.

13.13.2.158 axis_h_variable

SL axis_h_variable
H User-defined variable (ZA).
Definition at line 210 of file gclib_record.h.
The documentation for this struct was generated from the following file:

- gclib_record.h

13.14 GDataRecord47000_ENC Struct Reference

Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields.
`#include <gclib_record.h>`

Public Attributes

- 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.

13.14.1 Detailed Description

Data record struct for RIO-471xx and RIO-472xx PLCs. Includes encoder fields.
Definition at line 866 of file gclib_record.h.

13.14.2 Member Data Documentation

13.14.2.1 header_0

```
UB header_0
```

1st Byte of Header.
Definition at line 871 of file gclib_record.h.

13.14.2.2 header_1

```
UB header_1
```

2nd Byte of Header.
Definition at line 872 of file gclib_record.h.

13.14.2.3 header_2

```
UB header_2
```

3rd Byte of Header.
Definition at line 873 of file gclib_record.h.

13.14.2.4 header_3

```
UB header_3
```

4th Byte of Header.
Definition at line 874 of file gclib_record.h.

13.14.2.5 sample_number

```
UW sample_number
```

Sample number.
Definition at line 876 of file gclib_record.h.

13.14.2.6 error_code

```
UB error_code
```

Error code.
Definition at line 877 of file gclib_record.h.

13.14.2.7 general_status

```
UB general_status
```

General status.
Definition at line 878 of file gclib_record.h.

13.14.2.8 output_analog_0

`UW output_analog_0`
Analog output 0.
Definition at line 880 of file gclib_record.h.

13.14.2.9 output_analog_1

`UW output_analog_1`
Analog output 1.
Definition at line 881 of file gclib_record.h.

13.14.2.10 output_analog_2

`UW output_analog_2`
Analog output 2.
Definition at line 882 of file gclib_record.h.

13.14.2.11 output_analog_3

`UW output_analog_3`
Analog output 3.
Definition at line 883 of file gclib_record.h.

13.14.2.12 output_analog_4

`UW output_analog_4`
Analog output 4.
Definition at line 884 of file gclib_record.h.

13.14.2.13 output_analog_5

`UW output_analog_5`
Analog output 5.
Definition at line 885 of file gclib_record.h.

13.14.2.14 output_analog_6

`UW output_analog_6`
Analog output 6.
Definition at line 886 of file gclib_record.h.

13.14.2.15 output_analog_7

`UW output_analog_7`
Analog output 7.
Definition at line 887 of file gclib_record.h.

13.14.2.16 input_analog_0

`UW input_analog_0`
Analog input 0.
Definition at line 889 of file gclib_record.h.

13.14.2.17 input_analog_1

`UW input_analog_1`
Analog input 1.
Definition at line 890 of file gclib_record.h.

13.14.2.18 input_analog_2

`UW input_analog_2`
Analog input 2.
Definition at line 891 of file gclib_record.h.

13.14.2.19 input_analog_3

`UW input_analog_3`
Analog input 3.
Definition at line 892 of file gclib_record.h.

13.14.2.20 input_analog_4

`UW input_analog_4`
Analog input 4.
Definition at line 893 of file gclib_record.h.

13.14.2.21 input_analog_5

`UW input_analog_5`
Analog input 5.
Definition at line 894 of file gclib_record.h.

13.14.2.22 input_analog_6

`UW input_analog_6`
Analog input 6.
Definition at line 895 of file gclib_record.h.

13.14.2.23 input_analog_7

`UW input_analog_7`
Analog input 7.
Definition at line 896 of file gclib_record.h.

13.14.2.24 output_bank_0

`UW output_bank_0`
Digital outputs 0-15;.
Definition at line 898 of file gclib_record.h.

13.14.2.25 input_bank_0

`UW input_bank_0`
Digital inputs 0-15;.
Definition at line 900 of file gclib_record.h.

13.14.2.26 pulse_count_0

`UL pulse_count_0`
Pulse counter (see PC).
Definition at line 902 of file gclib_record.h.

13.14.2.27 zc_variable

`SL zc_variable`
ZC User-defined variable (see ZC).
Definition at line 903 of file gclib_record.h.

13.14.2.28 zd_variable

SL zd_variable
ZD User-defined variable (see ZD).
Definition at line 904 of file gclib_record.h.

13.14.2.29 encoder_0

SL encoder_0
Encoder channel 0. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 906 of file gclib_record.h.

13.14.2.30 encoder_1

SL encoder_1
Encoder channel 1. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 907 of file gclib_record.h.

13.14.2.31 encoder_2

SL encoder_2
Encoder channel 2. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 908 of file gclib_record.h.

13.14.2.32 encoder_3

SL encoder_3
Encoder channel 3. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 909 of file gclib_record.h.
The documentation for this struct was generated from the following file:

- gclib_record.h

13.15 GDataRecord47162 Struct Reference

Data record struct for RIO-47162.
`#include <gclib_record.h>`

Public Attributes

- 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.

13.15.1 Detailed Description

Data record struct for RIO-47162.
Definition at line 1015 of file gclib_record.h.

13.15.2 Member Data Documentation

13.15.2.1 header_0

`UB header_0`
1st Byte of Header.
Definition at line 1019 of file gclib_record.h.

13.15.2.2 header_1

`UB header_1`
2nd Byte of Header.
Definition at line 1020 of file gclib_record.h.

13.15.2.3 header_2

`UB header_2`
3rd Byte of Header.
Definition at line 1021 of file gclib_record.h.

13.15.2.4 header_3

`UB header_3`
4th Byte of Header.
Definition at line 1022 of file gclib_record.h.

13.15.2.5 sample_number

`UW sample_number`
Sample number.
Definition at line 1024 of file gclib_record.h.

13.15.2.6 error_code

`UB error_code`
Error code.
Definition at line 1025 of file gclib_record.h.

13.15.2.7 general_status

`UB general_status`
General status.
Definition at line 1026 of file gclib_record.h.

13.15.2.8 output_analog_0

`UW output_analog_0`
Analog output 0.
Definition at line 1028 of file gclib_record.h.

13.15.2.9 output_analog_1

`UW output_analog_1`
Analog output 1.
Definition at line 1029 of file gclib_record.h.

13.15.2.10 output_analog_2

`UW output_analog_2`
Analog output 2.
Definition at line 1030 of file gclib_record.h.

13.15.2.11 output_analog_3

`UW output_analog_3`
Analog output 3.
Definition at line 1031 of file gclib_record.h.

13.15.2.12 output_analog_4

`UW output_analog_4`
Analog output 4.
Definition at line 1032 of file gclib_record.h.

13.15.2.13 output_analog_5

`UW output_analog_5`
Analog output 5.
Definition at line 1033 of file gclib_record.h.

13.15.2.14 output_analog_6

`UW output_analog_6`
Analog output 6.
Definition at line 1034 of file gclib_record.h.

13.15.2.15 output_analog_7

`UW output_analog_7`
Analog output 7.
Definition at line 1035 of file gclib_record.h.

13.15.2.16 input_analog_0

`UW input_analog_0`
Analog input 0.
Definition at line 1037 of file gclib_record.h.

13.15.2.17 input_analog_1

`UW input_analog_1`
Analog input 1.
Definition at line 1038 of file gclib_record.h.

13.15.2.18 input_analog_2

`UW input_analog_2`
Analog input 2.
Definition at line 1039 of file gclib_record.h.

13.15.2.19 input_analog_3

`UW input_analog_3`
Analog input 3.
Definition at line 1040 of file gclib_record.h.

13.15.2.20 input_analog_4

`UW input_analog_4`
Analog input 4.
Definition at line 1041 of file gclib_record.h.

13.15.2.21 input_analog_5

`UW input_analog_5`
Analog input 5.
Definition at line 1042 of file gclib_record.h.

13.15.2.22 input_analog_6

`UW input_analog_6`
Analog input 6.
Definition at line 1043 of file gclib_record.h.

13.15.2.23 input_analog_7

`UW input_analog_7`
Analog input 7.
Definition at line 1044 of file gclib_record.h.

13.15.2.24 output_byte_0

`UB output_byte_0`
Digital outputs 0-7.
Definition at line 1046 of file gclib_record.h.

13.15.2.25 output_byte_1

`UB output_byte_1`
Digital outputs 8-15.
Definition at line 1047 of file gclib_record.h.

13.15.2.26 output_byte_2

`UB output_byte_2`
Digital outputs 16-23.
Definition at line 1048 of file gclib_record.h.

13.15.2.27 input_byte_0

`UB input_byte_0`
Digital inputs 0-7.
Definition at line 1050 of file gclib_record.h.

13.15.2.28 input_byte_1

```
UB input_byte_1
```

Digital inputs 8-15.
Definition at line 1051 of file gclib_record.h.

13.15.2.29 input_byte_2

```
UB input_byte_2
```

Digital inputs 16-23.
Definition at line 1052 of file gclib_record.h.

13.15.2.30 input_byte_3

```
UB input_byte_3
```

Digital inputs 24-31.
Definition at line 1053 of file gclib_record.h.

13.15.2.31 input_byte_4

```
UB input_byte_4
```

Digital inputs 32-39.
Definition at line 1054 of file gclib_record.h.

13.15.2.32 pulse_count_0

```
UL pulse_count_0
```

Pulse counter (see PC).
Definition at line 1056 of file gclib_record.h.

13.15.2.33 zc_variable

```
SL zc_variable
```

ZC User-defined variable (see ZC).
Definition at line 1057 of file gclib_record.h.

13.15.2.34 zd_variable

```
SL zd_variable
```

ZD User-defined variable (see ZD).
Definition at line 1058 of file gclib_record.h.

13.15.2.35 encoder_0

```
SL encoder_0
```

Encoder channel 0. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 1060 of file gclib_record.h.

13.15.2.36 encoder_1

```
SL encoder_1
```

Encoder channel 1. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 1061 of file gclib_record.h.

13.15.2.37 encoder_2

```
SL encoder_2
```

Encoder channel 2. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 1062 of file gclib_record.h.

13.15.2.38 encoder_3

SL encoder_3
Encoder channel 3. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 1063 of file gclib_record.h.
The documentation for this struct was generated from the following file:

- gclib_record.h

13.16 GDataRecord47300_24EX Struct Reference

Data record struct for RIO-47300 with 24EX I/O daughter board.
`#include <gclib_record.h>`

Public Attributes

- 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.

13.16.1 Detailed Description

Data record struct for RIO-47300 with 24EX I/O daughter board.
Definition at line 964 of file gclib_record.h.

13.16.2 Member Data Documentation

13.16.2.1 header_0

```
UB header_0
```

1st Byte of Header.
Definition at line 969 of file gclib_record.h.

13.16.2.2 header_1

```
UB header_1
```

2nd Byte of Header.
Definition at line 970 of file gclib_record.h.

13.16.2.3 header_2

```
UB header_2
```

3rd Byte of Header.
Definition at line 971 of file gclib_record.h.

13.16.2.4 header_3

`UB header_3`
4th Byte of Header.
Definition at line 972 of file gclib_record.h.

13.16.2.5 sample_number

`UW sample_number`
Sample number.
Definition at line 974 of file gclib_record.h.

13.16.2.6 error_code

`UB error_code`
Error code.
Definition at line 975 of file gclib_record.h.

13.16.2.7 general_status

`UB general_status`
General status.
Definition at line 976 of file gclib_record.h.

13.16.2.8 output_analog_0

`UW output_analog_0`
Analog output 0.
Definition at line 978 of file gclib_record.h.

13.16.2.9 output_analog_1

`UW output_analog_1`
Analog output 1.
Definition at line 979 of file gclib_record.h.

13.16.2.10 output_analog_2

`UW output_analog_2`
Analog output 2.
Definition at line 980 of file gclib_record.h.

13.16.2.11 output_analog_3

`UW output_analog_3`
Analog output 3.
Definition at line 981 of file gclib_record.h.

13.16.2.12 output_analog_4

`UW output_analog_4`
Analog output 4.
Definition at line 982 of file gclib_record.h.

13.16.2.13 output_analog_5

`UW output_analog_5`
Analog output 5.
Definition at line 983 of file gclib_record.h.

13.16.2.14 output_analog_6

`UW output_analog_6`
Analog output 6.
Definition at line 984 of file gclib_record.h.

13.16.2.15 output_analog_7

`UW output_analog_7`
Analog output 7.
Definition at line 985 of file gclib_record.h.

13.16.2.16 input_analog_0

`UW input_analog_0`
Analog input 0.
Definition at line 987 of file gclib_record.h.

13.16.2.17 input_analog_1

`UW input_analog_1`
Analog input 1.
Definition at line 988 of file gclib_record.h.

13.16.2.18 input_analog_2

`UW input_analog_2`
Analog input 2.
Definition at line 989 of file gclib_record.h.

13.16.2.19 input_analog_3

`UW input_analog_3`
Analog input 3.
Definition at line 990 of file gclib_record.h.

13.16.2.20 input_analog_4

`UW input_analog_4`
Analog input 4.
Definition at line 991 of file gclib_record.h.

13.16.2.21 input_analog_5

`UW input_analog_5`
Analog input 5.
Definition at line 992 of file gclib_record.h.

13.16.2.22 input_analog_6

`UW input_analog_6`
Analog input 6.
Definition at line 993 of file gclib_record.h.

13.16.2.23 input_analog_7

`UW input_analog_7`
Analog input 7.
Definition at line 994 of file gclib_record.h.

13.16.2.24 output_bank_0

`UW output_bank_0`
Digital outputs 0-15.
Definition at line 996 of file gclib_record.h.

13.16.2.25 output_bank_1

`UW output_bank_1`
Digital outputs 16-23.
Definition at line 997 of file gclib_record.h.

13.16.2.26 input_bank_0

`UW input_bank_0`
Digital inputs 0-15.
Definition at line 999 of file gclib_record.h.

13.16.2.27 input_bank_1

`UW input_bank_1`
Digital inputs 16-23.
Definition at line 1000 of file gclib_record.h.

13.16.2.28 pulse_count_0

`UL pulse_count_0`
Pulse counter (see PC)8.
Definition at line 1002 of file gclib_record.h.

13.16.2.29 zc_variable

`SL zc_variable`
ZC User-defined variable (see ZC).
Definition at line 1003 of file gclib_record.h.

13.16.2.30 zd_variable

`SL zd_variable`
ZD User-defined variable (see ZD).
Definition at line 1004 of file gclib_record.h.

13.16.2.31 output_bank_2

`UW output_bank_2`
Digital outputs 24-39. Data only valid for parts with 24EXOUT.
Definition at line 1006 of file gclib_record.h.

13.16.2.32 output_back_3

`UW output_back_3`
Digital outputs 40-47. Data only valid for parts with 24EXOUT.
Definition at line 1007 of file gclib_record.h.

13.16.2.33 input_bank_2

`UW input_bank_2`
Digital inputs 24-39. Data only valid for parts with 24EXIN.
Definition at line 1009 of file gclib_record.h.

13.16.2.34 input_bank_3

`UW input_bank_3`
Digital inputs 40-47. Data only valid for parts with 24EXIN.
Definition at line 1010 of file gclib_record.h.
The documentation for this struct was generated from the following file:

- gclib_record.h

13.17 GDataRecord47300_ENC Struct Reference

Data record struct for RIO-47300. Includes encoder fields.
`#include <gclib_record.h>`

Public Attributes

- 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.

13.17.1 Detailed Description

Data record struct for RIO-47300. Includes encoder fields.
Definition at line 914 of file gclib_record.h.

13.17.2 Member Data Documentation

13.17.2.1 header_0

```
UB header_0
```

1st Byte of Header.
Definition at line 919 of file gclib_record.h.

13.17.2.2 header_1

```
UB header_1
```

2nd Byte of Header.
Definition at line 920 of file gclib_record.h.

13.17.2.3 header_2

```
UB header_2
```

3rd Byte of Header.
Definition at line 921 of file gclib_record.h.

13.17.2.4 header_3

UB header_3
4th Byte of Header.
Definition at line 922 of file gclib_record.h.

13.17.2.5 sample_number

UW sample_number
Sample number.
Definition at line 924 of file gclib_record.h.

13.17.2.6 error_code

UB error_code
Error code.
Definition at line 925 of file gclib_record.h.

13.17.2.7 general_status

UB general_status
General status.
Definition at line 926 of file gclib_record.h.

13.17.2.8 output_analog_0

UW output_analog_0
Analog output 0.
Definition at line 928 of file gclib_record.h.

13.17.2.9 output_analog_1

UW output_analog_1
Analog output 1.
Definition at line 929 of file gclib_record.h.

13.17.2.10 output_analog_2

UW output_analog_2
Analog output 2.
Definition at line 930 of file gclib_record.h.

13.17.2.11 output_analog_3

UW output_analog_3
Analog output 3.
Definition at line 931 of file gclib_record.h.

13.17.2.12 output_analog_4

UW output_analog_4
Analog output 4.
Definition at line 932 of file gclib_record.h.

13.17.2.13 output_analog_5

UW output_analog_5
Analog output 5.
Definition at line 933 of file gclib_record.h.

13.17.2.14 output_analog_6

`UW output_analog_6`
Analog output 6.
Definition at line 934 of file gclib_record.h.

13.17.2.15 output_analog_7

`UW output_analog_7`
Analog output 7.
Definition at line 935 of file gclib_record.h.

13.17.2.16 input_analog_0

`UW input_analog_0`
Analog input 0.
Definition at line 937 of file gclib_record.h.

13.17.2.17 input_analog_1

`UW input_analog_1`
Analog input 1.
Definition at line 938 of file gclib_record.h.

13.17.2.18 input_analog_2

`UW input_analog_2`
Analog input 2.
Definition at line 939 of file gclib_record.h.

13.17.2.19 input_analog_3

`UW input_analog_3`
Analog input 3.
Definition at line 940 of file gclib_record.h.

13.17.2.20 input_analog_4

`UW input_analog_4`
Analog input 4.
Definition at line 941 of file gclib_record.h.

13.17.2.21 input_analog_5

`UW input_analog_5`
Analog input 5.
Definition at line 942 of file gclib_record.h.

13.17.2.22 input_analog_6

`UW input_analog_6`
Analog input 6.
Definition at line 943 of file gclib_record.h.

13.17.2.23 input_analog_7

`UW input_analog_7`
Analog input 7.
Definition at line 944 of file gclib_record.h.

13.17.2.24 output_bank_0

`UW output_bank_0`
Digital outputs 0-15;.
Definition at line 946 of file gclib_record.h.

13.17.2.25 output_bank_1

`UW output_bank_1`
Digital outputs 16-23;.
Definition at line 947 of file gclib_record.h.

13.17.2.26 input_bank_0

`UW input_bank_0`
Digital inputs 0-15;.
Definition at line 949 of file gclib_record.h.

13.17.2.27 input_bank_1

`UW input_bank_1`
Digital inputs 16-23;.
Definition at line 950 of file gclib_record.h.

13.17.2.28 pulse_count_0

`UL pulse_count_0`
Pulse counter (see PC).
Definition at line 952 of file gclib_record.h.

13.17.2.29 zc_variable

`SL zc_variable`
ZC User-defined variable (see ZC).
Definition at line 953 of file gclib_record.h.

13.17.2.30 zd_variable

`SL zd_variable`
ZD User-defined variable (see ZD).
Definition at line 954 of file gclib_record.h.

13.17.2.31 encoder_0

`SL encoder_0`
Encoder channel 0. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 956 of file gclib_record.h.

13.17.2.32 encoder_1

`SL encoder_1`
Encoder channel 1. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 957 of file gclib_record.h.

13.17.2.33 encoder_2

`SL encoder_2`
Encoder channel 2. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 958 of file gclib_record.h.

13.17.2.34 encoder_3

SL encoder_3
Encoder channel 3. Data only valid for parts with -BISS, -QUAD, or -SSI.
Definition at line 959 of file gclib_record.h.
The documentation for this struct was generated from the following file:

- gclib_record.h

13.18 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>`

Public Attributes

- 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).

13.18.1 Detailed Description

Data record struct for DMC-52000 controller. Same as DMC-4000, with bank indicator added at byte 40.
Definition at line 214 of file gclib_record.h.

13.18.2 Member Data Documentation

13.18.2.1 header_0

```
UB header_0
```

1st Byte of Header.
Definition at line 219 of file gclib_record.h.

13.18.2.2 header_1

```
UB header_1
```

2nd Byte of Header.
Definition at line 220 of file gclib_record.h.

13.18.2.3 header_2

```
UB header_2
```

3rd Byte of Header.
Definition at line 221 of file gclib_record.h.

13.18.2.4 header_3

`UB header_3`
4th Byte of Header.
Definition at line 222 of file gclib_record.h.

13.18.2.5 sample_number

`UW sample_number`
sample number.
Definition at line 224 of file gclib_record.h.

13.18.2.6 input_bank_0

`UB input_bank_0`
general input bank 0 (inputs 1-8).
Definition at line 226 of file gclib_record.h.

13.18.2.7 input_bank_1

`UB input_bank_1`
general input bank 1 (inputs 9-16).
Definition at line 227 of file gclib_record.h.

13.18.2.8 input_bank_2

`UB input_bank_2`
general input bank 2 (inputs 17-24).
Definition at line 228 of file gclib_record.h.

13.18.2.9 input_bank_3

`UB input_bank_3`
general input bank 3 (inputs 25-32).
Definition at line 229 of file gclib_record.h.

13.18.2.10 input_bank_4

`UB input_bank_4`
general input bank 4 (inputs 33-40).
Definition at line 230 of file gclib_record.h.

13.18.2.11 input_bank_5

`UB input_bank_5`
general input bank 5 (inputs 41-48).
Definition at line 231 of file gclib_record.h.

13.18.2.12 input_bank_6

`UB input_bank_6`
general input bank 6 (inputs 49-56).
Definition at line 232 of file gclib_record.h.

13.18.2.13 input_bank_7

`UB input_bank_7`
general input bank 7 (inputs 57-64).
Definition at line 233 of file gclib_record.h.

13.18.2.14 input_bank_8

`UB input_bank_8`
general input bank 8 (inputs 65-72).
Definition at line 234 of file gclib_record.h.

13.18.2.15 input_bank_9

`UB input_bank_9`
general input bank 9 (inputs 73-80).
Definition at line 235 of file gclib_record.h.

13.18.2.16 output_bank_0

`UB output_bank_0`
general output bank 0 (outputs 1-8).
Definition at line 237 of file gclib_record.h.

13.18.2.17 output_bank_1

`UB output_bank_1`
general output bank 1 (outputs 9-16).
Definition at line 238 of file gclib_record.h.

13.18.2.18 output_bank_2

`UB output_bank_2`
general output bank 2 (outputs 17-24).
Definition at line 239 of file gclib_record.h.

13.18.2.19 output_bank_3

`UB output_bank_3`
general output bank 3 (outputs 25-32).
Definition at line 240 of file gclib_record.h.

13.18.2.20 output_bank_4

`UB output_bank_4`
general output bank 4 (outputs 33-40).
Definition at line 241 of file gclib_record.h.

13.18.2.21 output_bank_5

`UB output_bank_5`
general output bank 5 (outputs 41-48).
Definition at line 242 of file gclib_record.h.

13.18.2.22 output_bank_6

`UB output_bank_6`
general output bank 6 (outputs 49-56).
Definition at line 243 of file gclib_record.h.

13.18.2.23 output_bank_7

`UB output_bank_7`
general output bank 7 (outputs 57-64).
Definition at line 244 of file gclib_record.h.

13.18.2.24 output_bank_8

`UB output_bank_8`
general output bank 8 (outputs 65-72).
Definition at line 245 of file gclib_record.h.

13.18.2.25 output_bank_9

`UB output_bank_9`
general output bank 9 (outputs 73-80).
Definition at line 246 of file gclib_record.h.

13.18.2.26 reserved_0

`SW reserved_0`
Reserved.
Definition at line 248 of file gclib_record.h.

13.18.2.27 reserved_2

`SW reserved_2`
Reserved.
Definition at line 249 of file gclib_record.h.

13.18.2.28 reserved_4

`SW reserved_4`
Reserved.
Definition at line 250 of file gclib_record.h.

13.18.2.29 reserved_6

`SW reserved_6`
Reserved.
Definition at line 251 of file gclib_record.h.

13.18.2.30 reserved_8

`SW reserved_8`
Reserved.
Definition at line 252 of file gclib_record.h.

13.18.2.31 reserved_10

`SW reserved_10`
Reserved.
Definition at line 253 of file gclib_record.h.

13.18.2.32 reserved_12

`SW reserved_12`
Reserved.
Definition at line 254 of file gclib_record.h.

13.18.2.33 ethercat_bank

`UB ethercat_bank`
EtherCAT Bank Indicator.
Definition at line 255 of file gclib_record.h.

13.18.2.34 reserved_14

```
UB reserved_14
```

Reserved.
Definition at line 256 of file gclib_record.h.

13.18.2.35 ethernet_status_a

```
UB ethernet_status_a
```

Ethernet Handle A Status.
Definition at line 258 of file gclib_record.h.

13.18.2.36 ethernet_status_b

```
UB ethernet_status_b
```

Ethernet Handle B Status.
Definition at line 259 of file gclib_record.h.

13.18.2.37 ethernet_status_c

```
UB ethernet_status_c
```

Ethernet Handle C Status.
Definition at line 260 of file gclib_record.h.

13.18.2.38 ethernet_status_d

```
UB ethernet_status_d
```

Ethernet Handle D Status.
Definition at line 261 of file gclib_record.h.

13.18.2.39 ethernet_status_e

```
UB ethernet_status_e
```

Ethernet Handle E Status.
Definition at line 262 of file gclib_record.h.

13.18.2.40 ethernet_status_f

```
UB ethernet_status_f
```

Ethernet Handle F Status.
Definition at line 263 of file gclib_record.h.

13.18.2.41 ethernet_status_g

```
UB ethernet_status_g
```

Ethernet Handle G Status.
Definition at line 264 of file gclib_record.h.

13.18.2.42 ethernet_status_h

```
UB ethernet_status_h
```

Ethernet Handle H Status.
Definition at line 265 of file gclib_record.h.

13.18.2.43 error_code

```
UB error_code
```

error code.
Definition at line 267 of file gclib_record.h.

13.18.2.44 thread_status

`UB thread_status`
thread status
Definition at line 268 of file gclib_record.h.

13.18.2.45 amplifier_status

`UL amplifier_status`
Amplifier Status.
Definition at line 269 of file gclib_record.h.

13.18.2.46 contour_segment_count

`UL contour_segment_count`
Segment Count for Contour Mode.
Definition at line 271 of file gclib_record.h.

13.18.2.47 contour_buffer_available

`UW contour_buffer_available`
Buffer space remaining, Contour Mode.
Definition at line 272 of file gclib_record.h.

13.18.2.48 s_plane_segment_count

`UW s_plane_segment_count`
segment count of coordinated move for S plane.
Definition at line 274 of file gclib_record.h.

13.18.2.49 s_plane_move_status

`UW s_plane_move_status`
coordinated move status for S plane.
Definition at line 275 of file gclib_record.h.

13.18.2.50 s_distance

`SL s_distance`
distance traveled in coordinated move for S plane.
Definition at line 276 of file gclib_record.h.

13.18.2.51 s_plane_buffer_available

`UW s_plane_buffer_available`
Buffer space remaining, S Plane.
Definition at line 277 of file gclib_record.h.

13.18.2.52 t_plane_segment_count

`UW t_plane_segment_count`
segment count of coordinated move for T plane.
Definition at line 279 of file gclib_record.h.

13.18.2.53 t_plane_move_status

`UW t_plane_move_status`
Coordinated move status for T plane.
Definition at line 280 of file gclib_record.h.

13.18.2.54 t_distance

SL t_distance
distance traveled in coordinated move for T plane.
Definition at line 281 of file gclib_record.h.

13.18.2.55 t_plane_buffer_available

UW t_plane_buffer_available
Buffer space remaining, T Plane.
Definition at line 282 of file gclib_record.h.

13.18.2.56 axis_a_status

UW axis_a_status
A axis status.
Definition at line 284 of file gclib_record.h.

13.18.2.57 axis_a_switches

UB axis_a_switches
A axis switches.
Definition at line 285 of file gclib_record.h.

13.18.2.58 axis_a_stop_code

UB axis_a_stop_code
A axis stop code.
Definition at line 286 of file gclib_record.h.

13.18.2.59 axis_a_reference_position

SL axis_a_reference_position
A axis reference position.
Definition at line 287 of file gclib_record.h.

13.18.2.60 axis_a_motor_position

SL axis_a_motor_position
A axis motor position.
Definition at line 288 of file gclib_record.h.

13.18.2.61 axis_a_position_error

SL axis_a_position_error
A axis position error.
Definition at line 289 of file gclib_record.h.

13.18.2.62 axis_a_aux_position

SL axis_a_aux_position
A axis auxiliary position.
Definition at line 290 of file gclib_record.h.

13.18.2.63 axis_a_velocity

SL axis_a_velocity
A axis velocity.
Definition at line 291 of file gclib_record.h.

13.18.2.64 axis_a_torque

SL axis_a_torque
A axis torque.
Definition at line 292 of file gclib_record.h.

13.18.2.65 axis_a_analog_in

UW axis_a_analog_in
A axis analog input.
Definition at line 293 of file gclib_record.h.

13.18.2.66 axis_a_halls

UB axis_a_halls
A Hall Input Status.
Definition at line 294 of file gclib_record.h.

13.18.2.67 axis_a_reserved

UB axis_a_reserved
Reserved.
Definition at line 295 of file gclib_record.h.

13.18.2.68 axis_a_variable

SL axis_a_variable
A User-defined variable (ZA).
Definition at line 296 of file gclib_record.h.

13.18.2.69 axis_b_status

UW axis_b_status
B axis status.
Definition at line 298 of file gclib_record.h.

13.18.2.70 axis_b_switches

UB axis_b_switches
B axis switches.
Definition at line 299 of file gclib_record.h.

13.18.2.71 axis_b_stop_code

UB axis_b_stop_code
B axis stop code.
Definition at line 300 of file gclib_record.h.

13.18.2.72 axis_b_reference_position

SL axis_b_reference_position
B axis reference position.
Definition at line 301 of file gclib_record.h.

13.18.2.73 axis_b_motor_position

SL axis_b_motor_position
B axis motor position.
Definition at line 302 of file gclib_record.h.

13.18.2.74 axis_b_position_error

SL axis_b_position_error
B axis position error.
Definition at line 303 of file gclib_record.h.

13.18.2.75 axis_b_aux_position

SL axis_b_aux_position
B axis auxiliary position.
Definition at line 304 of file gclib_record.h.

13.18.2.76 axis_b_velocity

SL axis_b_velocity
B axis velocity.
Definition at line 305 of file gclib_record.h.

13.18.2.77 axis_b_torque

SL axis_b_torque
B axis torque.
Definition at line 306 of file gclib_record.h.

13.18.2.78 axis_b_analog_in

UW axis_b_analog_in
B axis analog input.
Definition at line 307 of file gclib_record.h.

13.18.2.79 axis_b_halls

UB axis_b_halls
B Hall Input Status.
Definition at line 308 of file gclib_record.h.

13.18.2.80 axis_b_reserved

UB axis_b_reserved
Reserved.
Definition at line 309 of file gclib_record.h.

13.18.2.81 axis_b_variable

SL axis_b_variable
B User-defined variable (ZA).
Definition at line 310 of file gclib_record.h.

13.18.2.82 axis_c_status

UW axis_c_status
C axis status.
Definition at line 312 of file gclib_record.h.

13.18.2.83 axis_c_switches

UB axis_c_switches
C axis switches.
Definition at line 313 of file gclib_record.h.

13.18.2.84 axis_c_stop_code

`UB axis_c_stop_code`
C axis stop code.
Definition at line 314 of file gclib_record.h.

13.18.2.85 axis_c_reference_position

`SL axis_c_reference_position`
C axis reference position.
Definition at line 315 of file gclib_record.h.

13.18.2.86 axis_c_motor_position

`SL axis_c_motor_position`
C axis motor position.
Definition at line 316 of file gclib_record.h.

13.18.2.87 axis_c_position_error

`SL axis_c_position_error`
C axis position error.
Definition at line 317 of file gclib_record.h.

13.18.2.88 axis_c_aux_position

`SL axis_c_aux_position`
C axis auxiliary position.
Definition at line 318 of file gclib_record.h.

13.18.2.89 axis_c_velocity

`SL axis_c_velocity`
C axis velocity.
Definition at line 319 of file gclib_record.h.

13.18.2.90 axis_c_torque

`SL axis_c_torque`
C axis torque.
Definition at line 320 of file gclib_record.h.

13.18.2.91 axis_c_analog_in

`UW axis_c_analog_in`
C axis analog input.
Definition at line 321 of file gclib_record.h.

13.18.2.92 axis_c_halls

`UB axis_c_halls`
C Hall Input Status.
Definition at line 322 of file gclib_record.h.

13.18.2.93 axis_c_reserved

`UB axis_c_reserved`
Reserved.
Definition at line 323 of file gclib_record.h.

13.18.2.94 axis_c_variable

`SL axis_c_variable`
C User-defined variable (ZA).
Definition at line 324 of file gclib_record.h.

13.18.2.95 axis_d_status

`UW axis_d_status`
D axis status.
Definition at line 326 of file gclib_record.h.

13.18.2.96 axis_d_switches

`UB axis_d_switches`
D axis switches.
Definition at line 327 of file gclib_record.h.

13.18.2.97 axis_d_stop_code

`UB axis_d_stop_code`
D axis stop code.
Definition at line 328 of file gclib_record.h.

13.18.2.98 axis_d_reference_position

`SL axis_d_reference_position`
D axis reference position.
Definition at line 329 of file gclib_record.h.

13.18.2.99 axis_d_motor_position

`SL axis_d_motor_position`
D axis motor position.
Definition at line 330 of file gclib_record.h.

13.18.2.100 axis_d_position_error

`SL axis_d_position_error`
D axis position error.
Definition at line 331 of file gclib_record.h.

13.18.2.101 axis_d_aux_position

`SL axis_d_aux_position`
D axis auxiliary position.
Definition at line 332 of file gclib_record.h.

13.18.2.102 axis_d_velocity

`SL axis_d_velocity`
D axis velocity.
Definition at line 333 of file gclib_record.h.

13.18.2.103 axis_d_torque

`SL axis_d_torque`
D axis torque.
Definition at line 334 of file gclib_record.h.

13.18.2.104 axis_d_analog_in

`UW axis_d_analog_in`
D axis analog input.
Definition at line 335 of file gclib_record.h.

13.18.2.105 axis_d_halls

`UB axis_d_halls`
D Hall Input Status.
Definition at line 336 of file gclib_record.h.

13.18.2.106 axis_d_reserved

`UB axis_d_reserved`
Reserved.
Definition at line 337 of file gclib_record.h.

13.18.2.107 axis_d_variable

`SL axis_d_variable`
D User-defined variable (ZA).
Definition at line 338 of file gclib_record.h.

13.18.2.108 axis_e_status

`UW axis_e_status`
E axis status.
Definition at line 340 of file gclib_record.h.

13.18.2.109 axis_e_switches

`UB axis_e_switches`
E axis switches.
Definition at line 341 of file gclib_record.h.

13.18.2.110 axis_e_stop_code

`UB axis_e_stop_code`
E axis stop code.
Definition at line 342 of file gclib_record.h.

13.18.2.111 axis_e_reference_position

`SL axis_e_reference_position`
E axis reference position.
Definition at line 343 of file gclib_record.h.

13.18.2.112 axis_e_motor_position

`SL axis_e_motor_position`
E axis motor position.
Definition at line 344 of file gclib_record.h.

13.18.2.113 axis_e_position_error

`SL axis_e_position_error`
E axis position error.
Definition at line 345 of file gclib_record.h.

13.18.2.114 axis_e_aux_position

SL axis_e_aux_position
E axis auxiliary position.
Definition at line 346 of file gclib_record.h.

13.18.2.115 axis_e_velocity

SL axis_e_velocity
E axis velocity.
Definition at line 347 of file gclib_record.h.

13.18.2.116 axis_e_torque

SL axis_e_torque
E axis torque.
Definition at line 348 of file gclib_record.h.

13.18.2.117 axis_e_analog_in

UW axis_e_analog_in
E axis analog input.
Definition at line 349 of file gclib_record.h.

13.18.2.118 axis_e_halls

UB axis_e_halls
E Hall Input Status.
Definition at line 350 of file gclib_record.h.

13.18.2.119 axis_e_reserved

UB axis_e_reserved
Reserved.
Definition at line 351 of file gclib_record.h.

13.18.2.120 axis_e_variable

SL axis_e_variable
E User-defined variable (ZA).
Definition at line 352 of file gclib_record.h.

13.18.2.121 axis_f_status

UW axis_f_status
F axis status.
Definition at line 354 of file gclib_record.h.

13.18.2.122 axis_f_switches

UB axis_f_switches
F axis switches.
Definition at line 355 of file gclib_record.h.

13.18.2.123 axis_f_stop_code

UB axis_f_stop_code
F axis stop code.
Definition at line 356 of file gclib_record.h.

13.18.2.124 axis_f_reference_position

SL axis_f_reference_position
F axis reference position.
Definition at line 357 of file gclib_record.h.

13.18.2.125 axis_f_motor_position

SL axis_f_motor_position
F axis motor position.
Definition at line 358 of file gclib_record.h.

13.18.2.126 axis_f_position_error

SL axis_f_position_error
F axis position error.
Definition at line 359 of file gclib_record.h.

13.18.2.127 axis_f_aux_position

SL axis_f_aux_position
F axis auxiliary position.
Definition at line 360 of file gclib_record.h.

13.18.2.128 axis_f_velocity

SL axis_f_velocity
F axis velocity.
Definition at line 361 of file gclib_record.h.

13.18.2.129 axis_f_torque

SL axis_f_torque
F axis torque.
Definition at line 362 of file gclib_record.h.

13.18.2.130 axis_f_analog_in

UW axis_f_analog_in
F axis analog input.
Definition at line 363 of file gclib_record.h.

13.18.2.131 axis_f_halls

UB axis_f_halls
F Hall Input Status.
Definition at line 364 of file gclib_record.h.

13.18.2.132 axis_f_reserved

UB axis_f_reserved
Reserved.
Definition at line 365 of file gclib_record.h.

13.18.2.133 axis_f_variable

SL axis_f_variable
F User-defined variable (ZA).
Definition at line 366 of file gclib_record.h.

13.18.2.134 axis_g_status

`UW axis_g_status`
G axis status.
Definition at line 368 of file gclib_record.h.

13.18.2.135 axis_g_switches

`UB axis_g_switches`
G axis switches.
Definition at line 369 of file gclib_record.h.

13.18.2.136 axis_g_stop_code

`UB axis_g_stop_code`
G axis stop code.
Definition at line 370 of file gclib_record.h.

13.18.2.137 axis_g_reference_position

`SL axis_g_reference_position`
G axis reference position.
Definition at line 371 of file gclib_record.h.

13.18.2.138 axis_g_motor_position

`SL axis_g_motor_position`
G axis motor position.
Definition at line 372 of file gclib_record.h.

13.18.2.139 axis_g_position_error

`SL axis_g_position_error`
G axis position error.
Definition at line 373 of file gclib_record.h.

13.18.2.140 axis_g_aux_position

`SL axis_g_aux_position`
G axis auxiliary position.
Definition at line 374 of file gclib_record.h.

13.18.2.141 axis_g_velocity

`SL axis_g_velocity`
G axis velocity.
Definition at line 375 of file gclib_record.h.

13.18.2.142 axis_g_torque

`SL axis_g_torque`
G axis torque.
Definition at line 376 of file gclib_record.h.

13.18.2.143 axis_g_analog_in

`UW axis_g_analog_in`
G axis analog input.
Definition at line 377 of file gclib_record.h.

13.18.2.144 axis_g_halls

`UB axis_g_halls`
G Hall Input Status.
Definition at line 378 of file gclib_record.h.

13.18.2.145 axis_g_reserved

`UB axis_g_reserved`
Reserved.
Definition at line 379 of file gclib_record.h.

13.18.2.146 axis_g_variable

`SL axis_g_variable`
G User-defined variable (ZA).
Definition at line 380 of file gclib_record.h.

13.18.2.147 axis_h_status

`UW axis_h_status`
H axis status.
Definition at line 382 of file gclib_record.h.

13.18.2.148 axis_h_switches

`UB axis_h_switches`
H axis switches.
Definition at line 383 of file gclib_record.h.

13.18.2.149 axis_h_stop_code

`UB axis_h_stop_code`
H axis stop code.
Definition at line 384 of file gclib_record.h.

13.18.2.150 axis_h_reference_position

`SL axis_h_reference_position`
H axis reference position.
Definition at line 385 of file gclib_record.h.

13.18.2.151 axis_h_motor_position

`SL axis_h_motor_position`
H axis motor position.
Definition at line 386 of file gclib_record.h.

13.18.2.152 axis_h_position_error

`SL axis_h_position_error`
H axis position error.
Definition at line 387 of file gclib_record.h.

13.18.2.153 axis_h_aux_position

`SL axis_h_aux_position`
H axis auxiliary position.
Definition at line 388 of file gclib_record.h.

13.18.2.154 axis_h_velocity

SL axis_h_velocity
H axis velocity.
Definition at line 389 of file gclib_record.h.

13.18.2.155 axis_h_torque

SL axis_h_torque
H axis torque.
Definition at line 390 of file gclib_record.h.

13.18.2.156 axis_h_analog_in

UW axis_h_analog_in
H axis analog input.
Definition at line 391 of file gclib_record.h.

13.18.2.157 axis_h_halls

UB axis_h_halls
H Hall Input Status.
Definition at line 392 of file gclib_record.h.

13.18.2.158 axis_h_reserved

UB axis_h_reserved
Reserved.
Definition at line 393 of file gclib_record.h.

13.18.2.159 axis_h_variable

SL axis_h_variable
H User-defined variable (ZA).
Definition at line 394 of file gclib_record.h.
The documentation for this struct was generated from the following file:

- gclib_record.h

13.19 Interrupt Struct Reference

```
#include <gclib.hpp>
```

Public Types

- enum Type {
  NoInterrupts = GCLIB_NO_INTERRUPTS ,
  UserInterrupt = GCLIB_USER_INTERRUPT ,
  MotionComplete = GCLIB_MOTION_COMPLETE ,
  AllAxesMotionComplete = GCLIB_ALL_AXES_MOTION_COMPLETE ,
  ExcessPositionError = GCLIB_EXCESS_POSITION_ERROR ,
  LimitSwitch = GCLIB_LIMIT_SWITCH ,
  WatchdogTimer = GCLIB_WATCHDOG_TIMER ,
  ProgramStopped = GCLIB_PROGRAM_STOPPED ,
  CommandDone = GCLIB_COMMAND_DONE ,
  DigitalInputLow = GCLIB_DIGITAL_INPUT_LOW ,
  AllInterrupts = GCLIB_ALL_INTERRUPTS }

Public Member Functions

- Interrupt (gclib_interrupt_t interrupt)

Public Attributes

- Type type
- union {
  char axis
  If `type` is `MotionComplete`, holds the axis that triggered the interrupt.
  uint8_t digital_input
  If `type` is `DigitalInputLow`, holds the digital input that triggered the interrupt.
  uint8_t user_interrupt
  If `type` is `UserInterrupt`, holds the user interrupt that was triggered.
  };

  Interrupt context. Only valid for `MotionComplete`, `DigitalInputLow`, and `UserInterrupt`.
- uint8_t status
  The raw status byte generated by the controller. See the `EI command reference` for all possible values.

13.19.1 Detailed Description

An unsolicited interrupt. Use `set_interrupts` to configure which types are generated by the controller, and `subscribe_interrupts` to receive generated interrupts from the controller.
Definition at line 88 of file gclib.hpp.

13.19.2 Member Enumeration Documentation

13.19.2.1 Type

`enum Type`
Flags that can be combined to select which interrupts should be enabled by `set_interrupts()`. Also used to categorize interrupts such as `MotionComplete` which correspond to a range of status bytes.

Enumerator

NoInterrupts	
UserInterrupt	
MotionComplete	
AllAxesMotionComplete	
ExcessPositionError	Must be reenabled with `set_interrupts()` after occurrence.
LimitSwitch	Must be reenabled with `set_interrupts()` after each occurrence.
WatchdogTimer	
ProgramStopped	If multiple threads are running, this interrupt is only triggered when all threads are finished.
CommandDone	
DigitalInputLow	Must be reenabled with `set_interrupts()` after each occurrence.
AllInterrupts	

Definition at line 92 of file gclib.hpp.

13.19.3 Constructor & Destructor Documentation

13.19.3.1 Interrupt()

```
Interrupt (
            gclib_interrupt_t interrupt)  [inline]
```

Definition at line 89 of file gclib.hpp.

13.19.4 Member Data Documentation

13.19.4.1 type

`Type type`
Definition at line 105 of file gclib.hpp.

13.19.4.2 axis

`char axis`
If `type` is `MotionComplete`, holds the axis that triggered the interrupt.
Definition at line 108 of file gclib.hpp.

13.19.4.3 digital_input

`uint8_t digital_input`
If `type` is `DigitalInputLow`, holds the digital input that triggered the interrupt.
Definition at line 109 of file gclib.hpp.

13.19.4.4 user_interrupt

`uint8_t user_interrupt`
If `type` is `UserInterrupt`, holds the user interrupt that was triggered.
Definition at line 110 of file gclib.hpp.

13.19.4.5 [union]

`union { ... }`
Interrupt context. Only valid for `MotionComplete`, `DigitalInputLow`, and `UserInterrupt`.

13.19.4.6 status

`uint8_t status`
The raw status byte generated by the controller. See the `EI command reference` for all possible values.
Definition at line 112 of file gclib.hpp.
The documentation for this struct was generated from the following file:

- gclib.hpp

Chapter 14

File Documentation

14.1 examples.md File Reference

14.2 license.md File Reference

14.3 preprocessor.md File Reference

14.4 /builds/software/gclib/doc/README.md File Reference

14.5 gclib.h File Reference

```
#include <string.h>
#include <stdlib.h>
#include <stdint.h>
#include <stdbool.h>
#include <stdarg.h>
#include "gclib_compat.h"
```

Classes

- struct gclib_interrupt_t

  An interrupt generated by the controller.

Typedefs

- typedef struct gclib_context * gclib_handle

  A handle to an open connection given by `gclib_open()` for use in future API calls.
- typedef void * gclib_data_record_handle

  A handle to a data record, for use in Data Record API calls.
- typedef enum gclib_result gclib_result
- typedef enum gclib_interrupt_type gclib_interrupt_type
- typedef struct gclib_interrupt_t gclib_interrupt_t

  An interrupt generated by the controller.
- typedef enum gclib_digital_input_flags gclib_digital_input_flags

  Flags to enable or disable `GCLIB_DIGITAL_INPUT_LOW` interrupts per digital input, for use in `gclib_set_interrupts()`.
- typedef enum gclib_axis_flags gclib_axis_flags

  Flags for building an axis mask, for example in `gclib_set_interrupts()`.
- typedef enum gclib_connection_t gclib_connection_t

  Connection type, as returned by `gclib_connection_type()`.
- typedef enum gclib_mode_of_motion_t gclib_mode_of_motion_t

The controller's current mode of motion, given by `gclib_data_record_mode_of_motion()`.

- typedef enum gclib_ethernet_status_t gclib_ethernet_status_t

  The current status of an ethernet handle, given by `gclib_data_record_ethernet_status()`.

Enumerations

- enum gclib_result {
  GCLIB_SUCCESS ,
  GCLIB_INVALID_ARGUMENT ,
  GCLIB_INTERNAL_ERROR ,
  GCLIB_TIMEOUT ,
  GCLIB_COMMAND_ERROR ,
  GCLIB_BUFFER_TOO_SMALL ,
  GCLIB_NOT_CONNECTED ,
  GCLIB_NOT_SUBSCRIBED }
- enum gclib_interrupt_type {
  GCLIB_NO_INTERRUPTS = 0 ,
  GCLIB_USER_INTERRUPT = 0 ,
  GCLIB_MOTION_COMPLETE = 1 << 0 ,
  GCLIB_ALL_AXES_MOTION_COMPLETE = 1 << 8 ,
  GCLIB_EXCESS_POSITION_ERROR = 1 << 9 ,
  GCLIB_LIMIT_SWITCH = 1 << 10 ,
  GCLIB_WATCHDOG_TIMER = 1 << 11 ,
  GCLIB_PROGRAM_STOPPED = 1 << 13 ,
  GCLIB_COMMAND_DONE = 1 << 14 ,
  GCLIB_DIGITAL_INPUT_LOW = 1 << 15 ,
  GCLIB_ALL_INTERRUPTS = (1 << 16) - 1 }
- enum gclib_digital_input_flags {
  GCLIB_NO_DIGITAL_INPUTS = 0 ,
  GCLIB_DIGITAL_INPUT_1 = 1 << 0 ,
  GCLIB_DIGITAL_INPUT_2 = 1 << 1 ,
  GCLIB_DIGITAL_INPUT_3 = 1 << 2 ,
  GCLIB_DIGITAL_INPUT_4 = 1 << 3 ,
  GCLIB_DIGITAL_INPUT_5 = 1 << 4 ,
  GCLIB_DIGITAL_INPUT_6 = 1 << 5 ,
  GCLIB_DIGITAL_INPUT_7 = 1 << 6 ,
  GCLIB_DIGITAL_INPUT_8 = 1 << 7 ,
  GCLIB_ALL_DIGITAL_INPUTS = (1 << 8) - 1 }

  Flags to enable or disable `GCLIB_DIGITAL_INPUT_LOW` interrupts per digital input, for use in `gclib_set_interrupts()`.
- enum gclib_axis_flags {
  GCLIB_NO_AXES = 0 ,
  GCLIB_AXIS_A = 1 << 0 ,
  GCLIB_AXIS_B = 1 << 1 ,
  GCLIB_AXIS_C = 1 << 2 ,
  GCLIB_AXIS_D = 1 << 3 ,
  GCLIB_AXIS_E = 1 << 4 ,
  GCLIB_AXIS_F = 1 << 5 ,
  GCLIB_AXIS_G = 1 << 6 ,
  GCLIB_AXIS_H = 1 << 7 ,
  GCLIB_ALL_AXES = (1 << 8) - 1 }

  Flags for building an axis mask, for example in `gclib_set_interrupts()`.
- enum gclib_connection_t {
  GCLIB_ETHERNET = 1 ,
  GCLIB_SERIAL ,
  GCLIB_PCI }

  Connection type, as returned by `gclib_connection_type()`.

- enum gclib_mode_of_motion_t {
  GCLIB_NO_MOTION ,
  GCLIB_CONTOUR ,
  GCLIB_POSITION_ABSOLUTE ,
  GCLIB_POSITION_RELATIVE ,
  GCLIB_FIND_EDGE ,
  GCLIB_FIND_INDEX ,
  GCLIB_HOME ,
  GCLIB_VECTOR_MOVE ,
  GCLIB_LINEAR_MOVE = GCLIB_VECTOR_MOVE }

  The controller's current mode of motion, given by `gclib_data_record_mode_of_motion()`.
- enum gclib_ethernet_status_t {
  HANDLE_FREE = 0 ,
  UDP_SLAVE = 1 ,
  TCP_SLAVE = 2 ,
  UDP_MASTER = -1 ,
  TCP_MASTER = -2 ,
  ESTABLISHING_UDP = -5 ,
  ESTABLISHING_TCP = -6 }

  The current status of an ethernet handle, given by `gclib_data_record_ethernet_status()`.

Functions

- gclib_result gclib_version (char *version, size_t len)

  Get library version.
- gclib_result gclib_gcaps_version (char *gcaps_version, size_t len)

  Get library version used by current gcaps server.
- gclib_result gclib_addresses (char *addresses, size_t len)

  Show address, model, and serial of detected controllers, where available. Detected controllers are newline-separated. Each line is comma-separated.
- gclib_result gclib_ip_requests (char *ip_requests, size_t len)

  Show MAC address, model, and serial of all controllers requesting IP addresses, comma-separated. Use `gclib_assign_ip()` to assign an IP address to controllers returned by this method.
- gclib_result gclib_assign_ip (const char *mac, const char *ip)

  Assign IP to a controller.
- gclib_result gclib_set_dhcp_enabled (bool enabled)

  Enable or disable gcaps DHCP server.
- gclib_result gclib_dhcp_enabled (bool *enabled)

  Check if DHCP server is enabled.
- gclib_result gclib_list_servers (char *servers, size_t len)

  List available gcaps servers separated by newline.
- gclib_result gclib_server (char *name, size_t len)

  Current gcaps server.
- gclib_result gclib_set_server (const char *name)

  Set gcaps server.
- gclib_result gclib_published (char *name, size_t len)

  Provides published status of local gcaps server.
- gclib_result gclib_set_published (const char *name)

  Set published status of local gcaps server.
- void gclib_force_gcaps (bool on)

  Force future library calls to use either gcaps or direct connections.
- gclib_result gclib_open (gclib_handle *h, const char *address)

  Open a connection to a controller.
- gclib_result gclib_set_baud_rate (gclib_handle h, size_t baud_rate)

Set baud rate of serial connection.

- gclib_result gclib_close (gclib_handle *h)

  Close connection to controller.

- const char * gclib_error (gclib_handle h)

  Get library error string.

- gclib_result gclib_address (gclib_handle h, char *address, size_t len)

  Get connection address.

- gclib_result gclib_connection_type (gclib_handle h, gclib_connection_t *type)

  Get connection type.

- gclib_result gclib_revision_information (gclib_handle h, char *rev_info, size_t len)

  Get revision information (^R^V) from controller.

- gclib_result gclib_serial_number (gclib_handle h, uint32_t *serial_number)

  Get serial number from controller.

- gclib_result gclib_command (gclib_handle h, const char *command, char *buf, size_t len)

  Issues a command to the controller and provides the response.

- gclib_result gclib_set_interrupts (gclib_handle h, gclib_interrupt_type interrupt_mask, gclib_axis_flags motion_complete_axes, gclib_digital_input_flags digital_inputs)

  Configure which interrupts will be generated by the controller.

- gclib_result gclib_set_data_records (gclib_handle h, size_t period_ms)

  Configure controller data records.

- gclib_result gclib_subscribe_messages (gclib_handle h, void(callback)(void *user_data, const char *message), void *user_data)

  Subscribe to unsolicited messages.

- gclib_result gclib_subscribe_interrupts (gclib_handle h, void(callback)(void *user_data, gclib_interrupt_t interrupt), void *user_data)

  Subscribe to interrupts.

- gclib_result gclib_subscribe_data_records (gclib_handle h, void(callback)(void *user_data, gclib_data_record_handle data_record), void *user_data)

  Subscribe to data records.

- gclib_result gclib_subscribe_progress (gclib_handle h, void(callback)(void *user_data, size_t current, size_t max), void *user_data)

  Subscribe to progress for `gclib_set_program()`, `gclib_set_array()`, and `gclib_set_firmware()`.

- gclib_result gclib_unsubscribe (gclib_handle h, void *callback)

  Unsubscribe from messages, interrupts, data records, or progress.

- gclib_result gclib_message (gclib_handle h, char *message, size_t len, int timeout)

  Get a queued unsolicited message, or wait up to `timeout` ms for one to arrive.

- gclib_result gclib_interrupt (gclib_handle h, gclib_interrupt_t *interrupt, int timeout)

  Get a queued interrupt, or wait up to `timeout` ms for one to arrive.

- gclib_result gclib_data_record (gclib_handle h, gclib_data_record_handle *data_record, int timeout)

  Get a queued data record, or wait up to `timeout` ms for one to arrive.

- gclib_result gclib_program (gclib_handle h, char *program, size_t len)

  Get the controller's current program.

- gclib_result gclib_set_program (gclib_handle h, const char *program, const char *insert)

  Set the program on the controller.

- gclib_result gclib_array (gclib_handle h, const char *name, char *buf, size_t len, size_t start, size_t end)

  Get an array from the controller.

- gclib_result gclib_set_array (gclib_handle h, const char *name, const char *buf, size_t start, size_t end)

  Set an array on the controller.

- gclib_result gclib_set_firmware (gclib_handle h, const char *file_path)

  Set the firmware on the controller.

- uint16_t gclib_data_record_sample (gclib_data_record_handle h)

The sample in which this data record was generated.

- bool gclib_data_record_input (gclib_data_record_handle h, size_t index)

  Get an input.
- bool gclib_data_record_output (gclib_data_record_handle h, size_t index)

  Get an output.
- uint8_t gclib_data_record_input_bank (gclib_data_record_handle h, size_t index)

  Get a bank of inputs.
- uint8_t gclib_data_record_output_bank (gclib_data_record_handle h, size_t index)

  Get a bank of outputs.
- uint8_t gclib_data_record_error_code (gclib_data_record_handle h)

  Get the error code.
- uint32_t gclib_data_record_contour_segment_count (gclib_data_record_handle h)

  Get the contour segment count.
- uint16_t gclib_data_record_contour_buffer_available (gclib_data_record_handle h)

  Get remaining contour buffer.
- bool gclib_data_record_thread_running (gclib_data_record_handle h, size_t thread)

  Returns true if a given thread is running.
- gclib_ethernet_status_t gclib_data_record_ethernet_status (gclib_data_record_handle h, char handle)

  Get ethernet handle status.
- gclib_result gclib_data_record_bytes (gclib_data_record_handle h, char *data_record, size_t len)

  Copy data record to user buffer.
- bool gclib_data_record_motor_off (gclib_data_record_handle h, char axis)

  Get motor off status.
- bool gclib_data_record_latch_armed (gclib_data_record_handle h, char axis)

  Get latch armed.
- bool gclib_data_record_final_deceleration (gclib_data_record_handle h, char axis)

  Returns true when axis is in final deceleration.
- bool gclib_data_record_stopping (gclib_data_record_handle h, char axis)

  Returns true if axis is stopping.
- bool gclib_data_record_slewing (gclib_data_record_handle h, char axis)

  Returns true if axis is slewing.
- gclib_mode_of_motion_t gclib_data_record_mode_of_motion (gclib_data_record_handle h, char axis)

  Returns the mode of motion being profiled.
- bool gclib_data_record_negative_direction_move (gclib_data_record_handle h, char axis)

  Returns true if axis velocity is negative.
- bool gclib_data_record_moving (gclib_data_record_handle h, char axis)

  Returns true if motion is being profiled.
- int gclib_data_record_home_phase (gclib_data_record_handle h, char axis)

  Returns home phase when in `GCLIB_HOME` mode.
- bool gclib_data_record_stepper_mode (gclib_data_record_handle h, char axis)

  Returns true if axis is configured as a stepper motor.
- bool gclib_data_record_home_input (gclib_data_record_handle h, char axis)

  Returns the state of the home input.
- bool gclib_data_record_reverse_limit (gclib_data_record_handle h, char axis)

  Returns the state of the reverse limit.
- bool gclib_data_record_forward_limit (gclib_data_record_handle h, char axis)

  Returns the state of the forward limit.
- bool gclib_data_record_latch_input (gclib_data_record_handle h, char axis)

  Returns the state of the latch input.
- bool gclib_data_record_latch_occurred (gclib_data_record_handle h, char axis)

  Returns whether the latch was triggered.

- uint8_t gclib_data_record_stop_code (gclib_data_record_handle h, char axis)

  Returns a code indicating why the motor has stopped. See the SC command reference for details.
- int32_t gclib_data_record_reference_position (gclib_data_record_handle h, char axis)

  Returns the reference position.
- int32_t gclib_data_record_position (gclib_data_record_handle h, char axis)

  Returns the position.
- int32_t gclib_data_record_position_error (gclib_data_record_handle h, char axis)

  Returns the position error.
- int32_t gclib_data_record_aux_position (gclib_data_record_handle h, char axis)

  Returns the auxiliary position.
- int32_t gclib_data_record_velocity (gclib_data_record_handle h, char axis)

  Returns the velocity.
- double gclib_data_record_torque (gclib_data_record_handle h, char axis)

  Returns the torque.
- double gclib_data_record_analog_input (gclib_data_record_handle h, char axis)

  Returns the analog input.
- uint8_t gclib_data_record_halls (gclib_data_record_handle h, char axis)

  Returns the hall state.
- int32_t gclib_data_record_variable (gclib_data_record_handle h, char axis)

  Returns the value of a user variable for an axis.
- bool gclib_data_record_hall_error (gclib_data_record_handle h, char axis)

  Returns true if the amplifier for this axis has triggered a hall error.
- bool gclib_data_record_peak_current (gclib_data_record_handle h, char axis)

  Returns true if the amplifier for this axis has triggered a peak current error.
- bool gclib_data_record_over_current (gclib_data_record_handle h, size_t amp)

  Returns over current status.
- bool gclib_data_record_over_voltage (gclib_data_record_handle h, size_t amp)

  Returns over voltage status.
- bool gclib_data_record_over_temp (gclib_data_record_handle h, size_t amp)

  Returns over temperature status.
- bool gclib_data_record_under_voltage (gclib_data_record_handle h, size_t amp)

  Returns under voltage status.
- bool gclib_data_record_electronic_lock_out (gclib_data_record_handle h, size_t amp)

  Returns electronic lockout (ELO) status for a bank.
- uint16_t gclib_data_record_coordinated_move_segment_count (gclib_data_record_handle h, char plane)

  Returns the segment count for the current move.
- uint16_t gclib_data_record_coordinated_move_status (gclib_data_record_handle h, char plane)

  Returns the coordinated move status word.
- bool gclib_data_record_coordinated_move_final_deceleration (gclib_data_record_handle h, char plane)

  Returns true if in final deceleration.
- bool gclib_data_record_coordinated_move_stopping (gclib_data_record_handle h, char plane)

  Returns true if stopping.
- bool gclib_data_record_coordinated_move_slewing (gclib_data_record_handle h, char plane)

  Returns true if slewing.
- bool gclib_data_record_coordinated_move_moving (gclib_data_record_handle h, char plane)

  Returns true if moving.
- int32_t gclib_data_record_coordinated_move_distance (gclib_data_record_handle h, char plane)

  Returns the distance covered in the current move.
- uint16_t gclib_data_record_coordinated_move_buffer_available (gclib_data_record_handle h, char plane)

  Returns the available buffer space.

14.5.1 Detailed Description

Defines the interface for the Galil C Library (GCLIB).
Definition in file gclib.h.

14.6 gclib.h

Go to the documentation of this file.

```
00001
00003
00004 #pragma once
00005
00006 #ifdef __cplusplus
00007 extern "C" {
00008 #endif
00009
00010 #include <string.h>
00011 #include <stdlib.h>
00012 #include <stdint.h>
00013 #include <stdbool.h>
00014 #include <stdarg.h>
00015
00016 #include "gclib_compat.h"
00017
00018 #ifndef _WIN32
00019 #pragma GCC visibility push(default)
00020 #endif
00021
00023 typedef struct gclib_context* gclib_handle;
00025 typedef void* gclib_data_record_handle;
00026
00029 typedef enum gclib_result {
00030     GCLIB_SUCCESS,
00031     GCLIB_INVALID_ARGUMENT,
00032     GCLIB_INTERNAL_ERROR,
00033     GCLIB_TIMEOUT,
00034     GCLIB_COMMAND_ERROR,
00035     GCLIB_BUFFER_TOO_SMALL,
00036     GCLIB_NOT_CONNECTED,
00037     GCLIB_NOT_SUBSCRIBED
00038 } gclib_result;
00041 typedef enum gclib_interrupt_type {
00042     GCLIB_NO_INTERRUPTS = 0,
00043     GCLIB_USER_INTERRUPT = 0, // Cannot be enabled or disabled
00044     GCLIB_MOTION_COMPLETE = 1 << 0,
00045     GCLIB_ALL_AXES_MOTION_COMPLETE = 1 << 8,
00046     GCLIB_EXCESS_POSITION_ERROR = 1 << 9,
00047     GCLIB_LIMIT_SWITCH = 1 << 10,
00048     GCLIB_WATCHDOG_TIMER = 1 << 11,
00049     GCLIB_PROGRAM_STOPPED = 1 << 13,
00050     GCLIB_COMMAND_DONE = 1 << 14,
00051     GCLIB_DIGITAL_INPUT_LOW = 1 << 15,
00052     GCLIB_ALL_INTERRUPTS = (1 << 16) - 1,
00053 } gclib_interrupt_type;
00055 typedef struct gclib_interrupt_t {
00056     gclib_interrupt_type type;
00057     union {
00058         char axis;
00059         uint8_t digital_input;
00060         uint8_t user_interrupt;
00061     };
00062     uint8_t status;
00063 } gclib_interrupt_t;
00065 typedef enum gclib_digital_input_flags {
00066     GCLIB_NO_DIGITAL_INPUTS = 0,
00067     GCLIB_DIGITAL_INPUT_1 = 1 << 0,
00068     GCLIB_DIGITAL_INPUT_2 = 1 << 1,
00069     GCLIB_DIGITAL_INPUT_3 = 1 << 2,
00070     GCLIB_DIGITAL_INPUT_4 = 1 << 3,
00071     GCLIB_DIGITAL_INPUT_5 = 1 << 4,
00072     GCLIB_DIGITAL_INPUT_6 = 1 << 5,
00073     GCLIB_DIGITAL_INPUT_7 = 1 << 6,
00074     GCLIB_DIGITAL_INPUT_8 = 1 << 7,
00075     GCLIB_ALL_DIGITAL_INPUTS = (1 << 8) - 1
00076 } gclib_digital_input_flags;
00078 typedef enum gclib_axis_flags {
00079     GCLIB_NO_AXES = 0,
00080     GCLIB_AXIS_A = 1 << 0,
00081     GCLIB_AXIS_B = 1 << 1,
00082     GCLIB_AXIS_C = 1 << 2,
00083     GCLIB_AXIS_D = 1 << 3,
00084     GCLIB_AXIS_E = 1 << 4,
00085     GCLIB_AXIS_F = 1 << 5,
```

```
00086     GCLIB_AXIS_G = 1 << 6,
00087     GCLIB_AXIS_H = 1 << 7,
00088     GCLIB_ALL_AXES = (1 << 8) - 1
00089 } gclib_axis_flags;
00090
00091
00098 gclib_result gclib_version(char* version, size_t len);
00099
00106 gclib_result gclib_gcaps_version(char* gcaps_version, size_t len);
00107
00111
00112
00132 gclib_result gclib_addresses(char* addresses, size_t len);
00133
00147 gclib_result gclib_ip_requests(char* ip_requests, size_t len);
00148
00157 gclib_result gclib_assign_ip(const char* mac, const char* ip);
00158
00167 gclib_result gclib_set_dhcp_enabled(bool enabled);
00168
00178 gclib_result gclib_dhcp_enabled(bool* enabled);
00179
00185
00186
00193 gclib_result gclib_list_servers(char* servers, size_t len);
00194
00203 gclib_result gclib_server(char* name, size_t len);
00204
00212 gclib_result gclib_set_server(const char* name);
00213
00221 gclib_result gclib_published(char* name, size_t len);
00222
00230 gclib_result gclib_set_published(const char* name);
00231
00233
00234
00245 void gclib_force_gcaps(bool on);
00246
00260 gclib_result gclib_open(gclib_handle* h, const char* address);
00261
00269 gclib_result gclib_set_baud_rate(gclib_handle h, size_t baud_rate);
00270
00277 gclib_result gclib_close(gclib_handle* h);
00278
00291 const char* gclib_error(gclib_handle h);
00292
00300 gclib_result gclib_address(gclib_handle h, char* address, size_t len);
00302 typedef enum gclib_connection_t {
00303     GCLIB_ETHERNET = 1,
00304     GCLIB_SERIAL,
00305     GCLIB_PCI
00306 } gclib_connection_t;
00307
00314 gclib_result gclib_connection_type(gclib_handle h, gclib_connection_t* type);
00315
00320
00321
00332 gclib_result gclib_revision_information(gclib_handle h, char* rev_info, size_t len);
00333
00341 gclib_result gclib_serial_number(gclib_handle h, uint32_t* serial_number);
00342
00358 gclib_result gclib_command(gclib_handle h, const char* command, char* buf, size_t len);
00359
00360
00367
00368
00401 gclib_result gclib_set_interrupts(gclib_handle h, gclib_interrupt_type interrupt_mask,
      gclib_axis_flags motion_complete_axes, gclib_digital_input_flags digital_inputs);
00402
00415 gclib_result gclib_set_data_records(gclib_handle h, size_t period_ms);
00416
00440 gclib_result gclib_subscribe_messages(gclib_handle h, void(callback)(void* user_data, const char*
      message), void* user_data);
00441
00466 gclib_result gclib_subscribe_interrupts(gclib_handle h, void(callback)(void* user_data,
      gclib_interrupt_t interrupt), void* user_data);
00467
00492 gclib_result gclib_subscribe_data_records(gclib_handle h, void(callback)(void* user_data,
      gclib_data_record_handle data_record), void* user_data);
00493
00509 gclib_result gclib_subscribe_progress(gclib_handle h, void(callback)(void* user_data, size_t current,
      size_t max), void* user_data);
00510
00519 gclib_result gclib_unsubscribe(gclib_handle h, void* callback);
00520
00532 gclib_result gclib_message(gclib_handle h, char* message, size_t len, int timeout);
00533
```

```
00542 gclib_result gclib_interrupt(gclib_handle h, gclib_interrupt_t* interrupt, int timeout);
00543
00554 gclib_result gclib_data_record(gclib_handle h, gclib_data_record_handle* data_record, int timeout);
00555
00560
00561
00569 gclib_result gclib_program(gclib_handle h, char* program, size_t len);
00570
00580 gclib_result gclib_set_program(gclib_handle h, const char* program, const char* insert);
00581
00592 gclib_result gclib_array(gclib_handle h, const char* name, char* buf, size_t len, size_t start, size_t
      end);
00593
00611 gclib_result gclib_set_array(gclib_handle h, const char* name, const char* buf, size_t start, size_t
      end);
00612
00620 gclib_result gclib_set_firmware(gclib_handle h, const char* file_path);
00621
00623
00630
00632 typedef enum gclib_mode_of_motion_t {
00633     GCLIB_NO_MOTION,
00634     GCLIB_CONTOUR,
00635     GCLIB_POSITION_ABSOLUTE,
00636     GCLIB_POSITION_RELATIVE,
00637     GCLIB_FIND_EDGE,
00638     GCLIB_FIND_INDEX,
00639     GCLIB_HOME,
00640     GCLIB_VECTOR_MOVE,
00641     GCLIB_LINEAR_MOVE = GCLIB_VECTOR_MOVE
00642 } gclib_mode_of_motion_t;
00643
00645 typedef enum gclib_ethernet_status_t {
00646     HANDLE_FREE = 0,
00647     UDP_SLAVE = 1,
00648     TCP_SLAVE = 2,
00649     UDP_MASTER = -1,
00650     TCP_MASTER = -2,
00651     ESTABLISHING_UDP = -5,
00652     ESTABLISHING_TCP = -6
00653 } gclib_ethernet_status_t;
00654
00656 uint16_t gclib_data_record_sample(gclib_data_record_handle h);
00657
00669 bool gclib_data_record_input(gclib_data_record_handle h, size_t index);
00671 bool gclib_data_record_output(gclib_data_record_handle h, size_t index);
00673 uint8_t gclib_data_record_input_bank(gclib_data_record_handle h, size_t index);
00675 uint8_t gclib_data_record_output_bank(gclib_data_record_handle h, size_t index);
00677 uint8_t gclib_data_record_error_code(gclib_data_record_handle h);
00679 uint32_t gclib_data_record_contour_segment_count(gclib_data_record_handle h);
00681 uint16_t gclib_data_record_contour_buffer_available(gclib_data_record_handle h);
00683 bool gclib_data_record_thread_running(gclib_data_record_handle h, size_t thread);
00684
00691 gclib_ethernet_status_t gclib_data_record_ethernet_status(gclib_data_record_handle h, char handle);
00692
00702 gclib_result gclib_data_record_bytes(gclib_data_record_handle h, char* data_record, size_t len);
00706
00708 bool gclib_data_record_motor_off(gclib_data_record_handle h, char axis);
00710 bool gclib_data_record_latch_armed(gclib_data_record_handle h, char axis);
00712 bool gclib_data_record_final_deceleration(gclib_data_record_handle h, char axis);
00714 bool gclib_data_record_stopping(gclib_data_record_handle h, char axis);
00716 bool gclib_data_record_slewing(gclib_data_record_handle h, char axis);
00718 gclib_mode_of_motion_t gclib_data_record_mode_of_motion(gclib_data_record_handle h, char axis);
00720 bool gclib_data_record_negative_direction_move(gclib_data_record_handle h, char axis);
00722 bool gclib_data_record_moving(gclib_data_record_handle h, char axis);
00724 int gclib_data_record_home_phase(gclib_data_record_handle h, char axis);
00726 bool gclib_data_record_stepper_mode(gclib_data_record_handle h, char axis);
00728 bool gclib_data_record_home_input(gclib_data_record_handle h, char axis);
00730 bool gclib_data_record_reverse_limit(gclib_data_record_handle h, char axis);
00732 bool gclib_data_record_forward_limit(gclib_data_record_handle h, char axis);
00734 bool gclib_data_record_latch_input(gclib_data_record_handle h, char axis);
00736 bool gclib_data_record_latch_occurred(gclib_data_record_handle h, char axis);
00738 uint8_t gclib_data_record_stop_code(gclib_data_record_handle h, char axis);
00740 int32_t gclib_data_record_reference_position(gclib_data_record_handle h, char axis);
00742 int32_t gclib_data_record_position(gclib_data_record_handle h, char axis);
00744 int32_t gclib_data_record_position_error(gclib_data_record_handle h, char axis);
00746 int32_t gclib_data_record_aux_position(gclib_data_record_handle h, char axis);
00748 int32_t gclib_data_record_velocity(gclib_data_record_handle h, char axis);
00750 double gclib_data_record_torque(gclib_data_record_handle h, char axis);
00752 double gclib_data_record_analog_input(gclib_data_record_handle h, char axis);
00754 uint8_t gclib_data_record_halls(gclib_data_record_handle h, char axis);
00756 int32_t gclib_data_record_variable(gclib_data_record_handle h, char axis);
00758 bool gclib_data_record_hall_error(gclib_data_record_handle h, char axis);
00760 bool gclib_data_record_peak_current(gclib_data_record_handle h, char axis);
00767
00769 bool gclib_data_record_over_current(gclib_data_record_handle h, size_t amp);
00771 bool gclib_data_record_over_voltage(gclib_data_record_handle h, size_t amp);
```

```
00773 bool gclib_data_record_over_temp(gclib_data_record_handle h, size_t amp);
00775 bool gclib_data_record_under_voltage(gclib_data_record_handle h, size_t amp);
00777 bool gclib_data_record_electronic_lock_out(gclib_data_record_handle h, size_t amp);
00782
00784 uint16_t gclib_data_record_coordinated_move_segment_count(gclib_data_record_handle h, char plane);
00786 uint16_t gclib_data_record_coordinated_move_status(gclib_data_record_handle h, char plane);
00788 bool gclib_data_record_coordinated_move_final_deceleration(gclib_data_record_handle h, char plane);
00790 bool gclib_data_record_coordinated_move_stopping(gclib_data_record_handle h, char plane);
00792 bool gclib_data_record_coordinated_move_slewing(gclib_data_record_handle h, char plane);
00794 bool gclib_data_record_coordinated_move_moving(gclib_data_record_handle h, char plane);
00796 int32_t gclib_data_record_coordinated_move_distance(gclib_data_record_handle h, char plane);
00798 uint16_t gclib_data_record_coordinated_move_buffer_available(gclib_data_record_handle h, char plane);
00801
00802 // Undoc methods for gclibo
00803 void gclib_set_compat();
00804 gclib_result gclib_set_timeout(gclib_handle h, size_t timeout);
00805
00806 #ifdef __cplusplus
00807 }
00808 #endif
00809
00810 #ifndef _WIN32
00811 #pragma GCC visibility pop
00812 #endif
00813
```

14.7 gclib_compat.h File Reference

```
#include "gclib_record.h"
#include "gclib_errors.h"
```

Macros

- #define GCLIB_DEPRECATED
- #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_USE_INITIAL_TIMEOUT -1

  For GTimeout(), use this value to use initial GOpen() timeout (–timeout).
- #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 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

For GTimeout(), use 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.

Typedefs

- typedef int GReturn

  Every function returns a value of type GReturn. See gclib_errors.h for possible values.
- typedef struct gclib_context * 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_DEPRECATED GReturn GOpen (GCStringIn connection_string, GCon *g)

  Open a connection to a Galil Controller.
- GCLIB_DEPRECATED GReturn GClose (GCon g)

  Closes a connection to a Galil Controller.
- GCLIB_DEPRECATED GReturn GCommand (GCon g, GCStringIn command, GBufOut buffer, GSize buffer_len, GSize *bytes_returned)

  Performs a command-and-response transaction on the connection.
- GCLIB_DEPRECATED GReturn GProgramDownload (GCon g, GCStringIn program, GCStringIn preprocessor)

  Downloads a program to the controller's program buffer.
- GCLIB_DEPRECATED GReturn GProgramUpload (GCon g, GBufOut buffer, GSize buffer_len)

  Uploads a program from the controller's program buffer.
- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn 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_DEPRECATED GReturn 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_DEPRECATED GReturn GMessage (GCon g, GCStringOut buffer, GSize buffer_len)

  Provides access to unsolicited messages from the controller.
- GCLIB_DEPRECATED GReturn GInterrupt (GCon g, GStatus *status_byte)

  Provides access to PCI and UDP interrupts from the controller.
- GCLIB_DEPRECATED GReturn GFirmwareDownload (GCon g, GCStringIn filepath)

  Upgrade firmware.
- GCLIB_DEPRECATED GReturn GRead (GCon g, GBufOut buffer, GSize buffer_len, GSize *bytes_read)

Does nothing and returns G_GCLIB_ERROR.

- GCLIB_DEPRECATED GReturn GWrite (GCon g, GBufIn buffer, GSize buffer_len)

  Does nothing and returns G_GCLIB_ERROR.

- GCLIB_DEPRECATED GReturn GUtility (GCon g, GOption request, GMemory memory1, GMemory memory2)

  Does nothing and returns G_GCLIB_ERROR.

14.7.1 Macro Definition Documentation

14.7.1.1 GCLIB_DEPRECATED

```
#define GCLIB_DEPRECATED
```

Definition at line 35 of file gclib_compat.h.

14.7.1.2 G_DR

```
#define G_DR 1
```

Value for GRecord() `method` variable for acquiring a data record via DR mode.
Definition at line 50 of file gclib_compat.h.

14.7.1.3 G_QR

```
#define G_QR 0
```

Value for GRecord() `method` variable for acquiring a data record via QR mode.
Definition at line 51 of file gclib_compat.h.

14.7.1.4 G_BOUNDS

```
#define G_BOUNDS -1
```

For functions that take range options, e.g. GArrayUpload(), use this value for full range.
Definition at line 52 of file gclib_compat.h.

14.7.1.5 G_CR

```
#define G_CR 0
```

For GArrayUpload(), use this value in the delim field to delimit with carriage returns.
Definition at line 53 of file gclib_compat.h.

14.7.1.6 G_COMMA

```
#define G_COMMA 1
```

For GArrayUpload(), use this value in the delim field to delimit with commas.
Definition at line 54 of file gclib_compat.h.

14.7.1.7 G_PUBLISH_SERVER

```
#define G_PUBLISH_SERVER 1
```

For GPublishServer(), use this value to publish server to local network.
Definition at line 55 of file gclib_compat.h.

14.7.1.8 G_REMOVE_SERVER

```
#define G_REMOVE_SERVER 0
```

For GPublishServer(), use this value to remove server from local network.
Definition at line 56 of file gclib_compat.h.

14.7.1.9 G_USE_INITIAL_TIMEOUT [1/2]

```
#define G_USE_INITIAL_TIMEOUT -1
```

For GTimeout(), use this value to use initial GOpen() timeout (–timeout).
GUtility(), for timeout override. Set `G_UTIL_TIMEOUT_OVERRIDE` to this value to use initial GOpen() timeout (`--timeout`).
Definition at line 57 of file gclib_compat.h.

14.7.1.10 G_SMALL_BUFFER

```
#define G_SMALL_BUFFER 1024
```

Most reads from Galil are small. This value will easily hold most, e.g. TH, TZ, etc.
Definition at line 60 of file gclib_compat.h.

14.7.1.11 G_HUGE_BUFFER

```
#define G_HUGE_BUFFER 524288
```

Most reads from Galil hardware are small. This value will hold the largest array or program upload/download possible.
Definition at line 61 of file gclib_compat.h.

14.7.1.12 G_LINE_BUFFER

```
#define G_LINE_BUFFER 80
```

For writes, via command interpreter, to the Galil.
Definition at line 62 of file gclib_compat.h.

14.7.1.13 G_UTIL_TIMEOUT

```
#define G_UTIL_TIMEOUT 1
```

GUtility(), Access to timeout.
Definition at line 65 of file gclib_compat.h.

14.7.1.14 G_UTIL_TIMEOUT_OVERRIDE

```
#define G_UTIL_TIMEOUT_OVERRIDE 2
```

GUtility(), read/write access to timeout override.
Definition at line 66 of file gclib_compat.h.

14.7.1.15 G_USE_INITIAL_TIMEOUT [2/2]

```
#define G_USE_INITIAL_TIMEOUT -1
```

For GTimeout(), use this value to use initial GOpen() timeout (–timeout).
GUtility(), for timeout override. Set `G_UTIL_TIMEOUT_OVERRIDE` to this value to use initial GOpen() timeout (`--timeout`).
Definition at line 57 of file gclib_compat.h.

14.7.1.16 G_UTIL_VERSION

```
#define G_UTIL_VERSION 128
```

GUtility(), get a library version string.
Definition at line 68 of file gclib_compat.h.

14.7.1.17 G_UTIL_INFO

```
#define G_UTIL_INFO 129
```

GUtility(), get a connection info string.
Definition at line 69 of file gclib_compat.h.

14.7.1.18 G_UTIL_SLEEP

```
#define G_UTIL_SLEEP 130
```

GUtility(), specify an interval to sleep.
Definition at line 70 of file gclib_compat.h.

14.7.1.19 G_UTIL_ADDRESSES

```
#define G_UTIL_ADDRESSES 131
```

GUtility(), get a list of available connections.
Definition at line 71 of file gclib_compat.h.

14.7.1.20 G_UTIL_IPREQUEST

```
#define G_UTIL_IPREQUEST 132
```

GUtility(), get a list of hardware requesting IPs.
Definition at line 72 of file gclib_compat.h.

14.7.1.21 G_UTIL_ASSIGN

```
#define G_UTIL_ASSIGN 133
```

GUtility(), assign IP addresses over the network.
Definition at line 73 of file gclib_compat.h.

14.7.1.22 G_UTIL_DEVICE_INITIALIZE

```
#define G_UTIL_DEVICE_INITIALIZE 134
```

GUtility(), sends CF, CW, EO etc. to initialize the connection. Useful after RS or other reset.
Definition at line 74 of file gclib_compat.h.

14.7.1.23 G_UTIL_PING

```
#define G_UTIL_PING 135
```

GUtility(), uses ICMP ping to determine if an IP address is reachable and assigned.
Definition at line 75 of file gclib_compat.h.

14.7.1.24 G_UTIL_ERROR_CONTEXT

```
#define G_UTIL_ERROR_CONTEXT 136
```

GUtility(), provides additional error context, where available.
Definition at line 76 of file gclib_compat.h.

14.7.1.25 G_UTIL_GCAPS_HOST

```
#define G_UTIL_GCAPS_HOST 256
```

Definition at line 78 of file gclib_compat.h.

14.7.1.26 G_UTIL_GCAPS_VERSION

```
#define G_UTIL_GCAPS_VERSION 257
```

GUtility(), get the version of the gcaps server.
Definition at line 79 of file gclib_compat.h.

14.7.1.27 G_UTIL_GCAPS_KEEPALIVE

```
#define G_UTIL_GCAPS_KEEPALIVE 258
```

GUtility(), Deprecated 20210119. No longer functional.
Definition at line 80 of file gclib_compat.h.

14.7.1.28 G_UTIL_GCAPS_ADDRESSES

```
#define G_UTIL_GCAPS_ADDRESSES 259
```

GUtility(), get a list of available connections from the gcaps server.
Definition at line 81 of file gclib_compat.h.

14.7.1.29 G_UTIL_GCAPS_IPREQUEST

```
#define G_UTIL_GCAPS_IPREQUEST 260
```

GUtility(), get a list of hardware requesting IPs from the gcaps server.
Definition at line 82 of file gclib_compat.h.

14.7.1.30 G_UTIL_GCAPS_ASSIGN

```
#define G_UTIL_GCAPS_ASSIGN 261
```

GUtility(), assign IP addresses over the network from the gcaps server.
Definition at line 83 of file gclib_compat.h.

14.7.1.31 G_UTIL_GCAPS_PING

```
#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.
Definition at line 84 of file gclib_compat.h.

14.7.1.32 G_UTIL_GCAPS_LIST_SERVERS

```
#define G_UTIL_GCAPS_LIST_SERVERS 263
```

GUtility(), get a list of all available gcaps servers on the local network.
Definition at line 85 of file gclib_compat.h.

14.7.1.33 G_UTIL_GCAPS_PUBLISH_SERVER

```
#define G_UTIL_GCAPS_PUBLISH_SERVER 264
```

GUtility(), make local gcaps server discoverable by other gcaps servers on the local network.
Definition at line 86 of file gclib_compat.h.

14.7.1.34 G_UTIL_GCAPS_SET_SERVER

```
#define G_UTIL_GCAPS_SET_SERVER 265
```

GUtility(), set the new active gcaps server.
Definition at line 87 of file gclib_compat.h.

14.7.1.35 G_UTIL_GCAPS_SERVER_STATUS

```
#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.
Definition at line 88 of file gclib_compat.h.

14.7.1.36 G_UTIL_GCAPS_REMOTE_CONNECTIONS

```
#define G_UTIL_GCAPS_REMOTE_CONNECTIONS 267
```

GUtility(), get a list of remote addresses connected to local server.
Definition at line 89 of file gclib_compat.h.

14.7.1.37 G_UTIL_GCAPS_SERVER_INFO

```
#define G_UTIL_GCAPS_SERVER_INFO 268
```

Definition at line 90 of file gclib_compat.h.

14.7.1.38 G_UTIL_GCAPS_ADDRESSES_GET_REMEMBERED

```
#define G_UTIL_GCAPS_ADDRESSES_GET_REMEMBERED 269
```

GUtility(), returns true if gcaps is remembering ip assignments.
Definition at line 91 of file gclib_compat.h.

14.7.1.39 G_UTIL_GCAPS_ADDRESSES_SET_REMEMBERED

```
#define G_UTIL_GCAPS_ADDRESSES_SET_REMEMBERED 270
```

GUtility(), sets if gcaps should remember ip assignments.
Definition at line 92 of file gclib_compat.h.

14.7.2 Typedef Documentation

14.7.2.1 GReturn

```
typedef int GReturn
```

Every function returns a value of type GReturn. See gclib_errors.h for possible values.
Definition at line 94 of file gclib_compat.h.

14.7.2.2 GCon

```
typedef struct gclib_context* GCon
```

Connection handle. Unique for each connection in process. Assigned a non-zero value in GOpen().
Definition at line 95 of file gclib_compat.h.

14.7.2.3 GSize

```
typedef unsigned int GSize
```

Size of buffers, etc.
Definition at line 96 of file gclib_compat.h.

14.7.2.4 GOption

```
typedef int GOption
```

Option integer for various formatting, etc.
Definition at line 97 of file gclib_compat.h.

14.7.2.5 GCStringOut

```
typedef char* GCStringOut
```

C-string output from the library. Implies null-termination.
Definition at line 98 of file gclib_compat.h.

14.7.2.6 GCStringIn

```
typedef const char* GCStringIn
```

C-string input to the library. Implies null-termination.
Definition at line 99 of file gclib_compat.h.

14.7.2.7 GBufOut

```
typedef char* GBufOut
```

Data output from the library. No null-termination implied. Returned values may be null-terminated, see function documentation for details.
Definition at line 100 of file gclib_compat.h.

14.7.2.8 GBufIn

```
typedef const char* GBufIn
```

Data input to the library. No null-termination, function will have a GSize to indicate bytes to write .
Definition at line 101 of file gclib_compat.h.

14.7.2.9 GStatus

```
typedef unsigned char GStatus
```

Interrupt status byte.
Definition at line 102 of file gclib_compat.h.

14.7.2.10 GMemory

```
typedef void* GMemory
```

Pointer to untyped memory for use in GUtility().
Definition at line 103 of file gclib_compat.h.

14.8 gclib_compat.h

Go to the documentation of this file.

```
00001
00022
00027 #ifndef GCLIB_COMPAT_H
00028 #define GCLIB_COMPAT_H
00029
00030 #ifdef _MSC_VER
00031 #define GCLIB_DEPRECATED __declspec(deprecated)
00032 #elif defined(__GNUC__) || defined(__clang__)
00033 #define GCLIB_DEPRECATED __attribute__((__deprecated__))
00034 #else
00035 #define GCLIB_DEPRECATED
00036 #endif
00037
00038 #include "gclib_record.h" // Galil data record structs and unions.
00039 #include "gclib_errors.h" // GReturn error code values and strings.
00040
00041 #ifdef __cplusplus
00042 extern "C" {
00043 #endif
00044
00045 #ifndef _WIN32
00046 #pragma GCC visibility push(default)
00047 #endif
00048
00049 // Constants for function arguments
00050 #define G_DR 1
00051 #define G_QR 0
00052 #define G_BOUNDS -1
00053 #define G_CR 0
00054 #define G_COMMA 1
00055 #define G_PUBLISH_SERVER 1
00056 #define G_REMOVE_SERVER 0
00057 #define G_USE_INITIAL_TIMEOUT -1
00058
00059 // Convenience constants
00060 #define G_SMALL_BUFFER 1024
00061 #define G_HUGE_BUFFER 524288
00062 #define G_LINE_BUFFER 80
00063
00064 //Constants for GUtility()
00065 #define G_UTIL_TIMEOUT 1
00066 #define G_UTIL_TIMEOUT_OVERRIDE 2
00067 #define G_USE_INITIAL_TIMEOUT -1
00068 #define G_UTIL_VERSION 128
00069 #define G_UTIL_INFO 129
00070 #define G_UTIL_SLEEP 130
00071 #define G_UTIL_ADDRESSES 131
00072 #define G_UTIL_IPREQUEST 132
00073 #define G_UTIL_ASSIGN 133
00074 #define G_UTIL_DEVICE_INITIALIZE 134
00075 #define G_UTIL_PING 135
00076 #define G_UTIL_ERROR_CONTEXT 136
00077
00078 #define G_UTIL_GCAPS_HOST 256
00079 #define G_UTIL_GCAPS_VERSION 257
00080 #define G_UTIL_GCAPS_KEEPALIVE 258
```

```
00081 #define G_UTIL_GCAPS_ADDRESSES 259
00082 #define G_UTIL_GCAPS_IPREQUEST 260
00083 #define G_UTIL_GCAPS_ASSIGN 261
00084 #define G_UTIL_GCAPS_PING 262
00085 #define G_UTIL_GCAPS_LIST_SERVERS 263
00086 #define G_UTIL_GCAPS_PUBLISH_SERVER 264
00087 #define G_UTIL_GCAPS_SET_SERVER 265
00088 #define G_UTIL_GCAPS_SERVER_STATUS 266
00089 #define G_UTIL_GCAPS_REMOTE_CONNECTIONS 267
00090 #define G_UTIL_GCAPS_SERVER_INFO 268
00091 #define G_UTIL_GCAPS_ADDRESSES_GET_REMEMBERED 269
00092 #define G_UTIL_GCAPS_ADDRESSES_SET_REMEMBERED 270
00093
00094 typedef int GReturn;
00095 typedef struct gclib_context* GCon;
00096 typedef unsigned int GSize;
00097 typedef int GOption;
00098 typedef char* GCStringOut;
00099 typedef const char* GCStringIn;
00100 typedef char* GBufOut;
00101 typedef const char* GBufIn;
00102 typedef unsigned char GStatus;
00103 typedef void* GMemory;
00104
00106 GCLIB_DEPRECATED GReturn GOpen(GCStringIn connection_string, GCon* g);
00150
00152 GCLIB_DEPRECATED GReturn GClose(GCon g);
00166
00168 GCLIB_DEPRECATED GReturn GCommand(GCon g, GCStringIn command, GBufOut buffer, GSize buffer_len, GSize*
      bytes_returned);
00182
00184 GCLIB_DEPRECATED GReturn GProgramDownload(GCon g, GCStringIn program, GCStringIn preprocessor);
00196
00198 GCLIB_DEPRECATED GReturn GProgramUpload(GCon g, GBufOut buffer, GSize buffer_len);
00208
00210 GCLIB_DEPRECATED GReturn GArrayDownload(GCon g, const GCStringIn array_name, GOption first, GOption
      last, GCStringIn buffer);
00224
00226 GCLIB_DEPRECATED GReturn GArrayUpload(GCon g, const GCStringIn array_name, GOption first, GOption
      last, GOption delim, GBufOut buffer, GSize buffer_len);
00240
00242 GCLIB_DEPRECATED GReturn GRecord(GCon g, union GDataRecord* record, GOption method);
00266
00268 GCLIB_DEPRECATED GReturn GMessage(GCon g, GCStringOut buffer, GSize buffer_len);
00299
00301 GCLIB_DEPRECATED GReturn GInterrupt(GCon g, GStatus* status_byte);
00320
00322 GCLIB_DEPRECATED GReturn GFirmwareDownload(GCon g, GCStringIn filepath);
00331
00333 GCLIB_DEPRECATED GReturn GRead(GCon g, GBufOut buffer, GSize buffer_len, GSize* bytes_read);
00339 GCLIB_DEPRECATED GReturn GWrite(GCon g, GBufIn buffer, GSize buffer_len);
00345 GCLIB_DEPRECATED GReturn GUtility(GCon g, GOption request, GMemory memory1, GMemory memory2);
00350
00351 #ifdef __cplusplus
00352 } //extern "C"
00353 #endif
00354
00355 #ifndef _WIN32
00356 #pragma GCC visibility pop
00357 #endif
00358
00359 #endif // GCLIB_COMPAT_H
```

14.9 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.
- #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 overriden by GTimeout().
- #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()"

14.9.1 Detailed Description

Defines values for the Galil C Library return codes and error strings.
Definition in file gclib_errors.h.

14.9.2 Macro Definition Documentation

14.9.2.1 G_NO_ERROR

```
#define G_NO_ERROR 0
```

Return value if function succeeded.
Definition at line 9 of file gclib_errors.h.

14.9.2.2 G_NO_ERROR_S

```
#define G_NO_ERROR_S "no error"
```

Definition at line 10 of file gclib_errors.h.

14.9.2.3 G_GCLIB_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.
Definition at line 12 of file gclib_errors.h.

14.9.2.4 G_GCLIB_ERROR_S

```
#define G_GCLIB_ERROR_S "gclib unexpected error"
```
Definition at line 13 of file gclib_errors.h.

14.9.2.5 G_GCLIB_UTILITY_ERROR

```
#define G_GCLIB_UTILITY_ERROR -2
```
An invalid request value was specified to GUtility.
Definition at line 15 of file gclib_errors.h.

14.9.2.6 G_GCLIB_UTILITY_ERROR_S

```
#define G_GCLIB_UTILITY_ERROR_S "invalid request value or bad arguments were specified to
GUtility()"
```
Definition at line 16 of file gclib_errors.h.

14.9.2.7 G_GCLIB_UTILITY_IP_TAKEN

```
#define G_GCLIB_UTILITY_IP_TAKEN -3
```
The IP cannot be assigned because ping returned a reply.
Definition at line 18 of file gclib_errors.h.

14.9.2.8 G_GCLIB_UTILITY_IP_TAKEN_S

```
#define G_GCLIB_UTILITY_IP_TAKEN_S "ip address is already taken by a device on the network"
```
Definition at line 19 of file gclib_errors.h.

14.9.2.9 G_GCLIB_NON_BLOCKING_READ_EMPTY

```
#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.
Definition at line 21 of file gclib_errors.h.

14.9.2.10 G_GCLIB_NON_BLOCKING_READ_EMPTY_S

```
#define G_GCLIB_NON_BLOCKING_READ_EMPTY_S "data was not waiting for a zero-timeout read"
```
Definition at line 22 of file gclib_errors.h.

14.9.2.11 G_GCLIB_POLLING_FAILED

```
#define G_GCLIB_POLLING_FAILED -5
```
GWaitForBool out of polling trials.
Definition at line 24 of file gclib_errors.h.

14.9.2.12 G_GCLIB_POLLING_FAILED_S

```
#define G_GCLIB_POLLING_FAILED_S "exit condition not met in specified polling period"
```
Definition at line 25 of file gclib_errors.h.

14.9.2.13 G_TIMEOUT

```
#define G_TIMEOUT -1100
```
Operation timed out. Timeout is set by the –timeout option in GOpen() and can be overriden by GTimeout().
Definition at line 27 of file gclib_errors.h.

14.9.2.14 G_TIMEOUT_S

```
#define G_TIMEOUT_S "device timed out"
```
Definition at line 28 of file gclib_errors.h.

14.9.2.15 G_OPEN_ERROR

```
#define G_OPEN_ERROR -1101
```
Device could not be opened. E.G. Serial port or PCI device already open.
Definition at line 30 of file gclib_errors.h.

14.9.2.16 G_OPEN_ERROR_S

```
#define G_OPEN_ERROR_S "device failed to open"
```
Definition at line 31 of file gclib_errors.h.

14.9.2.17 G_ALREADY_OPEN

```
#define G_ALREADY_OPEN -1111
```
Serial or PCI file has a flock placed on it, presumably by another gclib connection.
Definition at line 33 of file gclib_errors.h.

14.9.2.18 G_ALREADY_OPEN_S

```
#define G_ALREADY_OPEN_S "Serial or PCI port already open"
```
Definition at line 34 of file gclib_errors.h.

14.9.2.19 G_READ_ERROR

```
#define G_READ_ERROR -1103
```
Device read failed. E.G. Socket was closed by remote host. See G_UTIL_GCAPS_KEEPALIVE.
Definition at line 36 of file gclib_errors.h.

14.9.2.20 G_READ_ERROR_S

```
#define G_READ_ERROR_S "device read error"
```
Definition at line 37 of file gclib_errors.h.

14.9.2.21 G_WRITE_ERROR

```
#define G_WRITE_ERROR -1104
```
Device write failed. E.G. Socket was closed by remote host. See G_UTIL_GCAPS_KEEPALIVE.
Definition at line 39 of file gclib_errors.h.

14.9.2.22 G_WRITE_ERROR_S

```
#define G_WRITE_ERROR_S "device write error"
```
Definition at line 40 of file gclib_errors.h.

14.9.2.23 G_INVALID_PREPROCESSOR_OPTIONS

```
#define G_INVALID_PREPROCESSOR_OPTIONS -1204
```
GProgramDownload was called with a bad preprocessor directive.
Definition at line 42 of file gclib_errors.h.

14.9.2.24 G_INVALID_PREPROCESSOR_OPTIONS_S

```
#define G_INVALID_PREPROCESSOR_OPTIONS_S "preprocessor did not recognize options"
```
Definition at line 43 of file gclib_errors.h.

14.9.2.25 G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND

```
#define G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND -1106
```
GCommand() was called with an illegal command, e.g. ED, DL or QD.
Definition at line 45 of file gclib_errors.h.

14.9.2.26 G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND_S

```
#define G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND_S "illegal command passed to command call"
```
Definition at line 46 of file gclib_errors.h.

14.9.2.27 G_DATA_RECORD_ERROR

```
#define G_DATA_RECORD_ERROR -1107
```
Data record error, e.g. DR attempted on serial connection.
Definition at line 48 of file gclib_errors.h.

14.9.2.28 G_DATA_RECORD_ERROR_S

```
#define G_DATA_RECORD_ERROR_S "data record error"
```
Definition at line 49 of file gclib_errors.h.

14.9.2.29 G_UNSUPPORTED_FUNCTION

```
#define G_UNSUPPORTED_FUNCTION -1109
```
Function cannot be called on this bus. E.G. GInterrupt() on serial.
Definition at line 51 of file gclib_errors.h.

14.9.2.30 G_UNSUPPORTED_FUNCTION_S

```
#define G_UNSUPPORTED_FUNCTION_S "function not supported on this communication bus"
```
Definition at line 52 of file gclib_errors.h.

14.9.2.31 G_FIRMWARE_LOAD_NOT_SUPPORTED

```
#define G_FIRMWARE_LOAD_NOT_SUPPORTED -1110
```
Firmware is not supported on this bus, e.g. Ethernet for the DMC-21x3 series.
Definition at line 54 of file gclib_errors.h.

14.9.2.32 G_FIRMWARE_LOAD_NOT_SUPPORTED_S

```
#define G_FIRMWARE_LOAD_NOT_SUPPORTED_S "firmware cannot be loaded on this communication bus
to this hardware"
```
Definition at line 55 of file gclib_errors.h.

14.9.2.33 G_ARRAY_NOT_DIMENSIONED

```
#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.
Definition at line 57 of file gclib_errors.h.

14.9.2.34 G_ARRAY_NOT_DIMENSIONED_S

```
#define G_ARRAY_NOT_DIMENSIONED_S "array not dimensioned on controller or wrong size"
```
Definition at line 58 of file gclib_errors.h.

14.9.2.35 G_CONNECTION_NOT_ESTABLISHED

```
#define G_CONNECTION_NOT_ESTABLISHED -1201
```
Function was called with no connection.
Definition at line 60 of file gclib_errors.h.

14.9.2.36 G_CONNECTION_NOT_ESTABLISHED_S

```
#define G_CONNECTION_NOT_ESTABLISHED_S "connection to hardware not established"
```
Definition at line 61 of file gclib_errors.h.

14.9.2.37 G_ILLEGAL_DATA_IN_PROGRAM

```
#define G_ILLEGAL_DATA_IN_PROGRAM -1202
```

Data to download not valid, e.g. \ in data.
Definition at line 63 of file gclib_errors.h.

14.9.2.38 G_ILLEGAL_DATA_IN_PROGRAM_S

```
#define G_ILLEGAL_DATA_IN_PROGRAM_S "illegal ASCII character in program"
```

Definition at line 64 of file gclib_errors.h.

14.9.2.39 G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT

```
#define G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT -1203
```

Program preprocessor could not compress the program within the user's constraints.
Definition at line 66 of file gclib_errors.h.

14.9.2.40 G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT_S

```
#define G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT_S "program cannot be compressed to fit on the
controller"
```

Definition at line 67 of file gclib_errors.h.

14.9.2.41 G_BAD_RESPONSE_QUESTION_MARK

```
#define G_BAD_RESPONSE_QUESTION_MARK -10000
```

Operation received a ?, indicating controller has a TC error.
Definition at line 69 of file gclib_errors.h.

14.9.2.42 G_BAD_RESPONSE_QUESTION_MARK_S

```
#define G_BAD_RESPONSE_QUESTION_MARK_S "question mark returned by controller"
```

Definition at line 70 of file gclib_errors.h.

14.9.2.43 G_BAD_VALUE_RANGE

```
#define G_BAD_VALUE_RANGE -10002
```

Bad value or range, e.g. GCon *g* variable passed to function was bad.
Definition at line 72 of file gclib_errors.h.

14.9.2.44 G_BAD_VALUE_RANGE_S

```
#define G_BAD_VALUE_RANGE_S "value passed to function was bad or out of range"
```

Definition at line 73 of file gclib_errors.h.

14.9.2.45 G_BAD_FULL_MEMORY

```
#define G_BAD_FULL_MEMORY -10003
```

Not enough memory for an operation, e.g. all connections allowed for a process already taken.
Definition at line 75 of file gclib_errors.h.

14.9.2.46 G_BAD_FULL_MEMORY_S

```
#define G_BAD_FULL_MEMORY_S "operation could not complete because of a memory error"
```

Definition at line 76 of file gclib_errors.h.

14.9.2.47 G_BAD_LOST_DATA

```
#define G_BAD_LOST_DATA -10004
```

Lost data, e.g. GCommand() response buffer was too small for the controller's response.
Definition at line 78 of file gclib_errors.h.

14.9.2.48 G_BAD_LOST_DATA_S

```
#define G_BAD_LOST_DATA_S "data was lost due to buffer or fifo limitations"
```
Definition at line 79 of file gclib_errors.h.

14.9.2.49 G_BAD_FILE

```
#define G_BAD_FILE -10005
```
Bad file path, bad file contents, or bad write.
Definition at line 81 of file gclib_errors.h.

14.9.2.50 G_BAD_FILE_S

```
#define G_BAD_FILE_S "file was not found, contents are invalid, or write failed"
```
Definition at line 82 of file gclib_errors.h.

14.9.2.51 G_BAD_ADDRESS

```
#define G_BAD_ADDRESS -10006
```
Bad address.
Definition at line 84 of file gclib_errors.h.

14.9.2.52 G_BAD_ADDRESS_S

```
#define G_BAD_ADDRESS_S "a bad address was specified in open"
```
Definition at line 85 of file gclib_errors.h.

14.9.2.53 G_BAD_FIRMWARE_LOAD

```
#define G_BAD_FIRMWARE_LOAD -10008
```
Bad firmware upgrade.
Definition at line 87 of file gclib_errors.h.

14.9.2.54 G_BAD_FIRMWARE_LOAD_S

```
#define G_BAD_FIRMWARE_LOAD_S "Firmware upgrade failed"
```
Definition at line 88 of file gclib_errors.h.

14.9.2.55 G_GCAPS_OPEN_ERROR

```
#define G_GCAPS_OPEN_ERROR -20000
```
gcaps connection couldn't open. Server is not running or is not reachable.
Definition at line 90 of file gclib_errors.h.

14.9.2.56 G_GCAPS_OPEN_ERROR_S

```
#define G_GCAPS_OPEN_ERROR_S "gcaps connection could not be opened"
```
Definition at line 91 of file gclib_errors.h.

14.9.2.57 G_GCAPS_SUBSCRIPTION_ERROR

```
#define G_GCAPS_SUBSCRIPTION_ERROR -20002
```
GMessage(), GRecord(), GInterrupt() called on a connection without –subscribe switch.
Definition at line 93 of file gclib_errors.h.

14.9.2.58 G_GCAPS_SUBSCRIPTION_ERROR_S

```
#define G_GCAPS_SUBSCRIPTION_ERROR_S "function requires subscription not specified in GOpen()"
```
Definition at line 94 of file gclib_errors.h.

14.10 gclib_errors.h

Go to the documentation of this file.

```
00001
00006 #ifndef I_0DD3687F_47D0_454B_ADB2_CBAB0ED46FCE
00007 #define I_0DD3687F_47D0_454B_ADB2_CBAB0ED46FCE
00008
00009 #define G_NO_ERROR 0
00010 #define G_NO_ERROR_S "no error"
00011
00012 #define G_GCLIB_ERROR -1
00013 #define G_GCLIB_ERROR_S "gclib unexpected error"
00014
00015 #define G_GCLIB_UTILITY_ERROR -2
00016 #define G_GCLIB_UTILITY_ERROR_S "invalid request value or bad arguments were specified to GUtility()"
00017
00018 #define G_GCLIB_UTILITY_IP_TAKEN -3
00019 #define G_GCLIB_UTILITY_IP_TAKEN_S "ip address is already taken by a device on the network"
00020
00021 #define G_GCLIB_NON_BLOCKING_READ_EMPTY -4
00022 #define G_GCLIB_NON_BLOCKING_READ_EMPTY_S "data was not waiting for a zero-timeout read"
00023
00024 #define G_GCLIB_POLLING_FAILED -5
00025 #define G_GCLIB_POLLING_FAILED_S "exit condition not met in specified polling period"
00026
00027 #define G_TIMEOUT -1100
00028 #define G_TIMEOUT_S "device timed out"
00029
00030 #define G_OPEN_ERROR -1101
00031 #define G_OPEN_ERROR_S "device failed to open"
00032
00033 #define G_ALREADY_OPEN -1111
00034 #define G_ALREADY_OPEN_S "Serial or PCI port already open"
00035
00036 #define G_READ_ERROR -1103
00037 #define G_READ_ERROR_S "device read error"
00038
00039 #define G_WRITE_ERROR -1104
00040 #define G_WRITE_ERROR_S "device write error"
00041
00042 #define G_INVALID_PREPROCESSOR_OPTIONS -1204
00043 #define G_INVALID_PREPROCESSOR_OPTIONS_S "preprocessor did not recognize options"
00044
00045 #define G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND -1106
00046 #define G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND_S "illegal command passed to command call"
00047
00048 #define G_DATA_RECORD_ERROR -1107
00049 #define G_DATA_RECORD_ERROR_S "data record error"
00050
00051 #define G_UNSUPPORTED_FUNCTION -1109
00052 #define G_UNSUPPORTED_FUNCTION_S "function not supported on this communication bus"
00053
00054 #define G_FIRMWARE_LOAD_NOT_SUPPORTED -1110
00055 #define G_FIRMWARE_LOAD_NOT_SUPPORTED_S "firmware cannot be loaded on this communication bus to this
     hardware"
00056
00057 #define G_ARRAY_NOT_DIMENSIONED -1200
00058 #define G_ARRAY_NOT_DIMENSIONED_S "array not dimensioned on controller or wrong size"
00059
00060 #define G_CONNECTION_NOT_ESTABLISHED -1201
00061 #define G_CONNECTION_NOT_ESTABLISHED_S "connection to hardware not established"
00062
00063 #define G_ILLEGAL_DATA_IN_PROGRAM -1202
00064 #define G_ILLEGAL_DATA_IN_PROGRAM_S "illegal ASCII character in program"
00065
00066 #define G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT -1203
00067 #define G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT_S "program cannot be compressed to fit on the controller"
00068
00069 #define G_BAD_RESPONSE_QUESTION_MARK -10000
00070 #define G_BAD_RESPONSE_QUESTION_MARK_S "question mark returned by controller"
00071
00072 #define G_BAD_VALUE_RANGE -10002
00073 #define G_BAD_VALUE_RANGE_S "value passed to function was bad or out of range"
00074
00075 #define G_BAD_FULL_MEMORY -10003
00076 #define G_BAD_FULL_MEMORY_S "operation could not complete because of a memory error"
00077
00078 #define G_BAD_LOST_DATA -10004
00079 #define G_BAD_LOST_DATA_S "data was lost due to buffer or fifo limitations"
00080
00081 #define G_BAD_FILE -10005
00082 #define G_BAD_FILE_S "file was not found, contents are invalid, or write failed"
00083
00084 #define G_BAD_ADDRESS -10006
00085 #define G_BAD_ADDRESS_S "a bad address was specified in open"
00086
```

```
00087 #define G_BAD_FIRMWARE_LOAD -10008
00088 #define G_BAD_FIRMWARE_LOAD_S "Firmware upgrade failed"
00089
00090 #define G_GCAPS_OPEN_ERROR -20000
00091 #define G_GCAPS_OPEN_ERROR_S "gcaps connection could not be opened"
00092
00093 #define G_GCAPS_SUBSCRIPTION_ERROR -20002
00094 #define G_GCAPS_SUBSCRIPTION_ERROR_S "function requires subscription not specified in GOpen()"
00095
00096 #endif //I_0DD3687F_47D0_454B_ADB2_CBAB0ED46FCE
```

14.11 gclib_record.h File Reference

`#include <stdint.h>`

Classes

- 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

14.11.1 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`.
Definition in file gclib_record.h.

14.11.2 Macro Definition Documentation

14.11.2.1 GALILDATARECORDMAXLENGTH

```
#define GALILDATARECORDMAXLENGTH 512
```

Max size for any Galil data record, equal to dual port ram size of PCI.
Definition at line 28 of file gclib_record.h.

14.11.3 Typedef Documentation

14.11.3.1 UB

```
typedef uint8_t UB
```

Definition at line 12 of file gclib_record.h.

14.11.3.2 UW

```
typedef uint16_t UW
```

Definition at line 13 of file gclib_record.h.

14.11.3.3 SW

```
typedef int16_t SW
```

Definition at line 14 of file gclib_record.h.

14.11.3.4 SL

```
typedef int32_t SL
```

Definition at line 15 of file gclib_record.h.

14.11.3.5 UL

```
typedef uint32_t UL
```

Definition at line 16 of file gclib_record.h.

14.12 gclib_record.h

Go to the documentation of this file.

```
00001
00006
00007 #ifndef I_210405D9_D9EF_484F_8258_BB29A1BBA217
00008 #define I_210405D9_D9EF_484F_8258_BB29A1BBA217
00009
00010 //typedefs to keep the layout of the structs clean and matching the Galil user manual docs
00011 #include <stdint.h>
00012 typedef uint8_t UB;  //unsigned byte
00013 typedef uint16_t UW; //unsigned word
00014 typedef int16_t SW;  //signed word
00015 typedef int32_t SL;  //signed long
00016 typedef uint32_t UL; //unsigned long
00017
00018 #if defined(_MSC_VER) || defined(__GNUC__) || defined(__BORLANDC__)
00019 #define PACKOK
00020 #endif
00021
00022 #ifdef PACKOK
00023 #pragma pack(1)
00024 #else
00025 #error "Need to set structure packing for compiler"
00026 #endif
00027
00028 #define GALILDATARECORDMAXLENGTH 512
00029
00031 struct GDataRecord4000
00032 {
00033
00034     /*Offset   type name        description*/
00035
00036     /*00*/      UB    header_0;
00037     /*01*/      UB    header_1;
```

```
00038     /*02*/      UB    header_2;
00039     /*03*/      UB    header_3;
00040
00041     /*04-05*/   UW    sample_number;
00042
00043     /*06*/      UB    input_bank_0;
00044     /*07*/      UB    input_bank_1;
00045     /*08*/      UB    input_bank_2;
00046     /*09*/      UB    input_bank_3;
00047     /*10*/      UB    input_bank_4;
00048     /*11*/      UB    input_bank_5;
00049     /*12*/      UB    input_bank_6;
00050     /*13*/      UB    input_bank_7;
00051     /*14*/      UB    input_bank_8;
00052     /*15*/      UB    input_bank_9;
00053
00054     /*16*/      UB    output_bank_0;
00055     /*17*/      UB    output_bank_1;
00056     /*18*/      UB    output_bank_2;
00057     /*19*/      UB    output_bank_3;
00058     /*20*/      UB    output_bank_4;
00059     /*21*/      UB    output_bank_5;
00060     /*22*/      UB    output_bank_6;
00061     /*23*/      UB    output_bank_7;
00062     /*24*/      UB    output_bank_8;
00063     /*25*/      UB    output_bank_9;
00064
00065     /*26-27*/   SW  reserved_0;
00066     /*28-29*/   SW    reserved_2;
00067     /*30-31*/   SW    reserved_4;
00068     /*32-33*/   SW    reserved_6;
00069     /*34-35*/   SW    reserved_8;
00070     /*36-37*/   SW    reserved_10;
00071     /*38-39*/   SW    reserved_12;
00072     /*40-41*/   SW    reserved_14;
00073
00074     /*42*/      UB    ethernet_status_a;
00075     /*43*/      UB    ethernet_status_b;
00076     /*44*/      UB    ethernet_status_c;
00077     /*45*/      UB    ethernet_status_d;
00078     /*46*/      UB    ethernet_status_e;
00079     /*47*/      UB    ethernet_status_f;
00080     /*48*/      UB    ethernet_status_g;
00081     /*49*/      UB    ethernet_status_h;
00082
00083     /*50*/      UB    error_code;
00084     /*51*/      UB    thread_status;
00085     /*52-55*/   UL    amplifier_status;
00086
00087     /*56-59*/   UL    contour_segment_count;
00088     /*60-61*/   UW    contour_buffer_available;
00089
00090     /*62-63*/   UW    s_plane_segment_count;
00091     /*64-65*/   UW    s_plane_move_status;
00092     /*66-69*/   SL    s_distance;
00093     /*70-71*/   UW    s_plane_buffer_available;
00094
00095     /*72-73*/   UW    t_plane_segment_count;
00096     /*74-75*/   UW    t_plane_move_status;
00097     /*76-79*/   SL    t_distance;
00098     /*80-81*/   UW    t_plane_buffer_available;
00099
00100     /*82-83*/   UW    axis_a_status;
00101     /*84*/      UB    axis_a_switches;
00102     /*85*/      UB    axis_a_stop_code;
00103     /*86-89*/   SL    axis_a_reference_position;
00104     /*90-93*/   SL    axis_a_motor_position;
00105     /*94-97*/   SL    axis_a_position_error;
00106     /*98-101*/  SL    axis_a_aux_position;
00107     /*102-105*/ SL    axis_a_velocity;
00108     /*106-109*/ SL    axis_a_torque;
00109     /*110-111*/ UW  axis_a_analog_in;
00110     /*112*/     UB    axis_a_halls;
00111     /*113*/     UB    axis_a_reserved;
00112     /*114-117*/ SL    axis_a_variable;
00113
00114     /*118-119*/ UW    axis_b_status;
00115     /*120*/     UB    axis_b_switches;
00116     /*121*/     UB    axis_b_stop_code;
00117     /*122-125*/ SL    axis_b_reference_position;
00118     /*126-129*/ SL    axis_b_motor_position;
00119     /*130-133*/ SL    axis_b_position_error;
00120     /*134-137*/ SL    axis_b_aux_position;
00121     /*138-141*/ SL    axis_b_velocity;
00122     /*142-145*/ SL    axis_b_torque;
00123     /*146-147*/ UW  axis_b_analog_in;
00124     /*148*/     UB    axis_b_halls;
```

```
00125     /*149*/     UB    axis_b_reserved;
00126     /*150-153*/ SL    axis_b_variable;
00127
00128     /*154-155*/ UW    axis_c_status;
00129     /*156*/     UB    axis_c_switches;
00130     /*157*/     UB    axis_c_stop_code;
00131     /*158-161*/ SL    axis_c_reference_position;
00132     /*162-165*/ SL    axis_c_motor_position;
00133     /*166-169*/ SL    axis_c_position_error;
00134     /*170-173*/ SL    axis_c_aux_position;
00135     /*174-177*/ SL    axis_c_velocity;
00136     /*178-181*/ SL    axis_c_torque;
00137     /*182-183*/ UW    axis_c_analog_in;
00138     /*184*/     UB    axis_c_halls;
00139     /*185*/     UB    axis_c_reserved;
00140     /*186-189*/ SL    axis_c_variable;
00141
00142     /*190-191*/ UW    axis_d_status;
00143     /*192*/     UB    axis_d_switches;
00144     /*193*/     UB    axis_d_stop_code;
00145     /*194-197*/ SL    axis_d_reference_position;
00146     /*198-201*/ SL    axis_d_motor_position;
00147     /*202-205*/ SL    axis_d_position_error;
00148     /*206-209*/ SL    axis_d_aux_position;
00149     /*210-213*/ SL    axis_d_velocity;
00150     /*214-217*/ SL    axis_d_torque;
00151     /*218-219*/ UW  axis_d_analog_in;
00152     /*220*/     UB    axis_d_halls;
00153     /*221*/     UB    axis_d_reserved;
00154     /*222-225*/ SL    axis_d_variable;
00155
00156     /*226-227*/ UW    axis_e_status;
00157     /*228*/     UB    axis_e_switches;
00158     /*229*/     UB    axis_e_stop_code;
00159     /*230-233*/ SL    axis_e_reference_position;
00160     /*234-237*/ SL    axis_e_motor_position;
00161     /*238-241*/ SL    axis_e_position_error;
00162     /*242-245*/ SL    axis_e_aux_position;
00163     /*246-249*/ SL    axis_e_velocity;
00164     /*250-253*/ SL    axis_e_torque;
00165     /*254-255*/ UW  axis_e_analog_in;
00166     /*256*/     UB    axis_e_halls;
00167     /*257*/     UB    axis_e_reserved;
00168     /*258-261*/ SL    axis_e_variable;
00169
00170     /*262-263*/ UW    axis_f_status;
00171     /*264*/     UB    axis_f_switches;
00172     /*265*/     UB    axis_f_stop_code;
00173     /*266-269*/ SL    axis_f_reference_position;
00174     /*270-273*/ SL    axis_f_motor_position;
00175     /*274-277*/ SL    axis_f_position_error;
00176     /*278-281*/ SL    axis_f_aux_position;
00177     /*282-285*/ SL    axis_f_velocity;
00178     /*286-289*/ SL    axis_f_torque;
00179     /*290-291*/ UW    axis_f_analog_in;
00180     /*292*/     UB    axis_f_halls;
00181     /*293*/     UB    axis_f_reserved;
00182     /*294-297*/ SL    axis_f_variable;
00183
00184     /*298-299*/ UW    axis_g_status;
00185     /*300*/     UB    axis_g_switches;
00186     /*301*/     UB    axis_g_stop_code;
00187     /*302-305*/ SL    axis_g_reference_position;
00188     /*306-309*/ SL    axis_g_motor_position;
00189     /*310-313*/ SL    axis_g_position_error;
00190     /*314-317*/ SL    axis_g_aux_position;
00191     /*318-321*/ SL    axis_g_velocity;
00192     /*322-325*/ SL    axis_g_torque;
00193     /*326-327*/ UW  axis_g_analog_in;
00194     /*328*/     UB    axis_g_halls;
00195     /*329*/     UB    axis_g_reserved;
00196     /*330-333*/ SL    axis_g_variable;
00197
00198     /*334-335*/ UW    axis_h_status;
00199     /*336*/     UB    axis_h_switches;
00200     /*337*/     UB    axis_h_stop_code;
00201     /*338-341*/ SL    axis_h_reference_position;
00202     /*342-345*/ SL    axis_h_motor_position;
00203     /*346-349*/ SL    axis_h_position_error;
00204     /*350-353*/ SL    axis_h_aux_position;
00205     /*354-357*/ SL    axis_h_velocity;
00206     /*358-361*/ SL    axis_h_torque;
00207     /*362-363*/ UW  axis_h_analog_in;
00208     /*364*/     UB    axis_h_halls;
00209     /*365*/     UB    axis_h_reserved;
00210     /*366-369*/ SL    axis_h_variable;
00211 }; //DataRecord4000
```

```
00212
00214 struct GDataRecord52000
00215 {
00216
00217     /*Offset   type name        description*/
00218
00219     /*00*/      UB    header_0;
00220     /*01*/      UB    header_1;
00221     /*02*/      UB    header_2;
00222     /*03*/      UB    header_3;
00223
00224     /*04-05*/   UW    sample_number;
00225
00226     /*06*/      UB    input_bank_0;
00227     /*07*/      UB    input_bank_1;
00228     /*08*/      UB    input_bank_2;
00229     /*09*/      UB    input_bank_3;
00230     /*10*/      UB    input_bank_4;
00231     /*11*/      UB    input_bank_5;
00232     /*12*/      UB    input_bank_6;
00233     /*13*/      UB    input_bank_7;
00234     /*14*/      UB    input_bank_8;
00235     /*15*/      UB    input_bank_9;
00236
00237     /*16*/      UB    output_bank_0;
00238     /*17*/      UB    output_bank_1;
00239     /*18*/      UB    output_bank_2;
00240     /*19*/      UB    output_bank_3;
00241     /*20*/      UB    output_bank_4;
00242     /*21*/      UB    output_bank_5;
00243     /*22*/      UB    output_bank_6;
00244     /*23*/      UB    output_bank_7;
00245     /*24*/      UB    output_bank_8;
00246     /*25*/      UB    output_bank_9;
00247
00248     /*26-27*/   SW  reserved_0;
00249     /*28-29*/   SW    reserved_2;
00250     /*30-31*/   SW    reserved_4;
00251     /*32-33*/   SW    reserved_6;
00252     /*34-35*/   SW    reserved_8;
00253     /*36-37*/   SW    reserved_10;
00254     /*38-39*/   SW    reserved_12;
00255     /*40*/      UB  ethercat_bank;
00256     /*41*/      UB    reserved_14;
00257
00258     /*42*/      UB    ethernet_status_a;
00259     /*43*/      UB    ethernet_status_b;
00260     /*44*/      UB    ethernet_status_c;
00261     /*45*/      UB    ethernet_status_d;
00262     /*46*/      UB    ethernet_status_e;
00263     /*47*/      UB    ethernet_status_f;
00264     /*48*/      UB    ethernet_status_g;
00265     /*49*/      UB    ethernet_status_h;
00266
00267     /*50*/      UB    error_code;
00268     /*51*/      UB    thread_status;
00269     /*52-55*/   UL    amplifier_status;
00270
00271     /*56-59*/   UL    contour_segment_count;
00272     /*60-61*/   UW    contour_buffer_available;
00273
00274     /*62-63*/   UW    s_plane_segment_count;
00275     /*64-65*/   UW    s_plane_move_status;
00276     /*66-69*/   SL    s_distance;
00277     /*70-71*/   UW    s_plane_buffer_available;
00278
00279     /*72-73*/   UW    t_plane_segment_count;
00280     /*74-75*/   UW    t_plane_move_status;
00281     /*76-79*/   SL    t_distance;
00282     /*80-81*/   UW    t_plane_buffer_available;
00283
00284     /*82-83*/   UW    axis_a_status;
00285     /*84*/      UB    axis_a_switches;
00286     /*85*/      UB    axis_a_stop_code;
00287     /*86-89*/   SL    axis_a_reference_position;
00288     /*90-93*/   SL    axis_a_motor_position;
00289     /*94-97*/   SL    axis_a_position_error;
00290     /*98-101*/  SL    axis_a_aux_position;
00291     /*102-105*/ SL    axis_a_velocity;
00292     /*106-109*/ SL    axis_a_torque;
00293     /*110-111*/ UW  axis_a_analog_in;
00294     /*112*/     UB    axis_a_halls;
00295     /*113*/     UB    axis_a_reserved;
00296     /*114-117*/ SL    axis_a_variable;
00297
00298     /*118-119*/ UW    axis_b_status;
00299     /*120*/     UB    axis_b_switches;
```

```
00300     /*121*/     UB    axis_b_stop_code;
00301     /*122-125*/ SL    axis_b_reference_position;
00302     /*126-129*/ SL    axis_b_motor_position;
00303     /*130-133*/ SL    axis_b_position_error;
00304     /*134-137*/ SL    axis_b_aux_position;
00305     /*138-141*/ SL    axis_b_velocity;
00306     /*142-145*/ SL    axis_b_torque;
00307     /*146-147*/ UW  axis_b_analog_in;
00308     /*148*/     UB    axis_b_halls;
00309     /*149*/     UB    axis_b_reserved;
00310     /*150-153*/ SL    axis_b_variable;
00311
00312     /*154-155*/ UW    axis_c_status;
00313     /*156*/     UB    axis_c_switches;
00314     /*157*/     UB    axis_c_stop_code;
00315     /*158-161*/ SL    axis_c_reference_position;
00316     /*162-165*/ SL    axis_c_motor_position;
00317     /*166-169*/ SL    axis_c_position_error;
00318     /*170-173*/ SL    axis_c_aux_position;
00319     /*174-177*/ SL    axis_c_velocity;
00320     /*178-181*/ SL    axis_c_torque;
00321     /*182-183*/ UW    axis_c_analog_in;
00322     /*184*/     UB    axis_c_halls;
00323     /*185*/     UB    axis_c_reserved;
00324     /*186-189*/ SL    axis_c_variable;
00325
00326     /*190-191*/ UW    axis_d_status;
00327     /*192*/     UB    axis_d_switches;
00328     /*193*/     UB    axis_d_stop_code;
00329     /*194-197*/ SL    axis_d_reference_position;
00330     /*198-201*/ SL    axis_d_motor_position;
00331     /*202-205*/ SL    axis_d_position_error;
00332     /*206-209*/ SL    axis_d_aux_position;
00333     /*210-213*/ SL    axis_d_velocity;
00334     /*214-217*/ SL    axis_d_torque;
00335     /*218-219*/ UW  axis_d_analog_in;
00336     /*220*/     UB    axis_d_halls;
00337     /*221*/     UB    axis_d_reserved;
00338     /*222-225*/ SL    axis_d_variable;
00339
00340     /*226-227*/ UW    axis_e_status;
00341     /*228*/     UB    axis_e_switches;
00342     /*229*/     UB    axis_e_stop_code;
00343     /*230-233*/ SL    axis_e_reference_position;
00344     /*234-237*/ SL    axis_e_motor_position;
00345     /*238-241*/ SL    axis_e_position_error;
00346     /*242-245*/ SL    axis_e_aux_position;
00347     /*246-249*/ SL    axis_e_velocity;
00348     /*250-253*/ SL    axis_e_torque;
00349     /*254-255*/ UW  axis_e_analog_in;
00350     /*256*/     UB    axis_e_halls;
00351     /*257*/     UB    axis_e_reserved;
00352     /*258-261*/ SL    axis_e_variable;
00353
00354     /*262-263*/ UW    axis_f_status;
00355     /*264*/     UB    axis_f_switches;
00356     /*265*/     UB    axis_f_stop_code;
00357     /*266-269*/ SL    axis_f_reference_position;
00358     /*270-273*/ SL    axis_f_motor_position;
00359     /*274-277*/ SL    axis_f_position_error;
00360     /*278-281*/ SL    axis_f_aux_position;
00361     /*282-285*/ SL    axis_f_velocity;
00362     /*286-289*/ SL    axis_f_torque;
00363     /*290-291*/ UW    axis_f_analog_in;
00364     /*292*/     UB    axis_f_halls;
00365     /*293*/     UB    axis_f_reserved;
00366     /*294-297*/ SL    axis_f_variable;
00367
00368     /*298-299*/ UW    axis_g_status;
00369     /*300*/     UB    axis_g_switches;
00370     /*301*/     UB    axis_g_stop_code;
00371     /*302-305*/ SL    axis_g_reference_position;
00372     /*306-309*/ SL    axis_g_motor_position;
00373     /*310-313*/ SL    axis_g_position_error;
00374     /*314-317*/ SL    axis_g_aux_position;
00375     /*318-321*/ SL    axis_g_velocity;
00376     /*322-325*/ SL    axis_g_torque;
00377     /*326-327*/ UW  axis_g_analog_in;
00378     /*328*/     UB    axis_g_halls;
00379     /*329*/     UB    axis_g_reserved;
00380     /*330-333*/ SL    axis_g_variable;
00381
00382     /*334-335*/ UW    axis_h_status;
00383     /*336*/     UB    axis_h_switches;
00384     /*337*/     UB    axis_h_stop_code;
00385     /*338-341*/ SL    axis_h_reference_position;
00386     /*342-345*/ SL    axis_h_motor_position;
```

```
00387     /*346-349*/ SL    axis_h_position_error;
00388     /*350-353*/ SL    axis_h_aux_position;
00389     /*354-357*/ SL    axis_h_velocity;
00390     /*358-361*/ SL    axis_h_torque;
00391     /*362-363*/ UW  axis_h_analog_in;
00392     /*364*/     UB    axis_h_halls;
00393     /*365*/     UB    axis_h_reserved;
00394     /*366-369*/ SL    axis_h_variable;
00395 }; //DataRecord52000
00396
00398
00405 struct GDataRecord1806
00406 {
00407     /*Offset   type name        description*/
00408
00409     /*00-01*/   UW    sample_number;
00410
00411     /*02*/      UB    input_bank_0;
00412     /*03*/      UB    input_bank_1;
00413     /*04*/      UB    input_bank_2;
00414     /*05*/      UB    input_bank_3;
00415     /*06*/      UB    input_bank_4;
00416     /*07*/      UB    input_bank_5;
00417     /*08*/      UB    input_bank_6;
00418     /*09*/      UB    input_bank_7;
00419     /*10*/      UB    input_bank_8;
00420     /*11*/      UB    input_bank_9;
00421
00422     /*12*/      UB    output_bank_0;
00423     /*13*/      UB    output_bank_1;
00424     /*14*/      UB    output_bank_2;
00425     /*15*/      UB    output_bank_3;
00426     /*16*/      UB    output_bank_4;
00427     /*17*/      UB    output_bank_5;
00428     /*18*/      UB    output_bank_6;
00429     /*19*/      UB    output_bank_7;
00430     /*20*/      UB    output_bank_8;
00431     /*21*/      UB    output_bank_9;
00432
00433     /*22-23*/   SW  reserved_0;
00434     /*24-25*/   SW    reserved_2;
00435     /*26-27*/   SW    reserved_4;
00436     /*28-29*/   SW    reserved_6;
00437     /*30-31*/   SW    reserved_8;
00438     /*32-33*/   SW    reserved_10;
00439     /*34-35*/   SW    reserved_12;
00440     /*36-37*/   SW    reserved_14;
00441
00442     /*38*/      UB    reserved_16;
00443     /*39*/      UB    reserved_17;
00444     /*40*/      UB    reserved_18;
00445     /*41*/      UB    reserved_19;
00446     /*42*/      UB    reserved_20;
00447     /*43*/      UB    reserved_21;
00448     /*44*/      UB    reserved_22;
00449     /*45*/      UB    reserved_23;
00450
00451     /*46*/      UB    error_code;
00452     /*47*/      UB    thread_status;
00453     /*48-51*/   UL    reserved_24;
00454
00455     /*52-55*/   UL    contour_segment_count;
00456     /*56-57*/   UW    contour_buffer_available;
00457
00458     /*58-59*/   UW    s_plane_segment_count;
00459     /*60-61*/   UW    s_plane_move_status;
00460     /*62-65*/   SL    s_distance;
00461     /*66-67*/   UW    s_plane_buffer_available;
00462
00463     /*68-69*/   UW    t_plane_segment_count;
00464     /*70-71*/   UW    t_plane_move_status;
00465     /*72-75*/   SL    t_distance;
00466     /*76-77*/   UW    t_plane_buffer_available;
00467
00468     /*78-79*/   UW    axis_a_status;
00469     /*80*/      UB    axis_a_switches;
00470     /*81*/      UB    axis_a_stop_code;
00471     /*82-85*/   SL    axis_a_reference_position;
00472     /*86-89*/   SL    axis_a_motor_position;
00473     /*90-93*/   SL    axis_a_position_error;
00474     /*94-97*/   SL    axis_a_aux_position;
00475     /*98-101*/  SL    axis_a_velocity;
00476     /*102-105*/ SL    axis_a_torque;
00477     /*106-107*/ UW  axis_a_analog_in;
00478     /*108*/     UB    axis_a_reserved_0;
00479     /*109*/     UB    axis_a_reserved_1;
00480     /*110-113*/ SL    axis_a_variable;
```

```
00481
00482     /*114-115*/ UW    axis_b_status;
00483     /*116*/     UB    axis_b_switches;
00484     /*117*/     UB    axis_b_stop_code;
00485     /*118-121*/ SL    axis_b_reference_position;
00486     /*122-125*/ SL    axis_b_motor_position;
00487     /*126-129*/ SL    axis_b_position_error;
00488     /*130-133*/ SL    axis_b_aux_position;
00489     /*134-137*/ SL    axis_b_velocity;
00490     /*138-141*/ SL    axis_b_torque;
00491     /*142-143*/ UW  axis_b_analog_in;
00492     /*144*/     UB    axis_b_reserved_0;
00493     /*145*/     UB    axis_b_reserved_1;
00494     /*146-149*/ SL    axis_b_variable;
00495
00496     /*150-151*/ UW    axis_c_status;
00497     /*152*/     UB    axis_c_switches;
00498     /*153*/     UB    axis_c_stop_code;
00499     /*154-157*/ SL    axis_c_reference_position;
00500     /*158-161*/ SL    axis_c_motor_position;
00501     /*162-165*/ SL    axis_c_position_error;
00502     /*166-169*/ SL    axis_c_aux_position;
00503     /*170-173*/ SL    axis_c_velocity;
00504     /*174-177*/ SL    axis_c_torque;
00505     /*178-179*/ UW    axis_c_analog_in;
00506     /*180*/     UB    axis_c_reserved_0;
00507     /*181*/     UB    axis_c_reserved_1;
00508     /*182-185*/ SL    axis_c_variable;
00509
00510     /*186-187*/ UW    axis_d_status;
00511     /*188*/     UB    axis_d_switches;
00512     /*189*/     UB    axis_d_stop_code;
00513     /*190-193*/ SL    axis_d_reference_position;
00514     /*194-197*/ SL    axis_d_motor_position;
00515     /*198-201*/ SL    axis_d_position_error;
00516     /*202-205*/ SL    axis_d_aux_position;
00517     /*206-209*/ SL    axis_d_velocity;
00518     /*210-213*/ SL    axis_d_torque;
00519     /*214-215*/ UW  axis_d_analog_in;
00520     /*216*/     UB    axis_d_reserved_0;
00521     /*217*/     UB    axis_d_reserved_1;
00522     /*218-221*/ SL    axis_d_variable;
00523
00524     /*222-223*/ UW    axis_e_status;
00525     /*224*/     UB    axis_e_switches;
00526     /*225*/     UB    axis_e_stop_code;
00527     /*226-229*/ SL    axis_e_reference_position;
00528     /*230-233*/ SL    axis_e_motor_position;
00529     /*234-237*/ SL    axis_e_position_error;
00530     /*238-241*/ SL    axis_e_aux_position;
00531     /*242-245*/ SL    axis_e_velocity;
00532     /*256-249*/ SL    axis_e_torque;
00533     /*250-251*/ UW  axis_e_analog_in;
00534     /*252*/     UB    axis_e_reserved_0;
00535     /*253*/     UB    axis_e_reserved_1;
00536     /*254-257*/ SL    axis_e_variable;
00537
00538     /*258-259*/ UW    axis_f_status;
00539     /*260*/     UB    axis_f_switches;
00540     /*261*/     UB    axis_f_stop_code;
00541     /*262-265*/ SL    axis_f_reference_position;
00542     /*266-269*/ SL    axis_f_motor_position;
00543     /*270-273*/ SL    axis_f_position_error;
00544     /*274-277*/ SL    axis_f_aux_position;
00545     /*278-281*/ SL    axis_f_velocity;
00546     /*282-285*/ SL    axis_f_torque;
00547     /*286-287*/ UW    axis_f_analog_in;
00548     /*288*/     UB    axis_f_reserved_0;
00549     /*289*/     UB    axis_f_reserved_1;
00550     /*290-293*/ SL    axis_f_variable;
00551
00552     /*294-295*/ UW    axis_g_status;
00553     /*296*/     UB    axis_g_switches;
00554     /*297*/     UB    axis_g_stop_code;
00555     /*298-301*/ SL    axis_g_reference_position;
00556     /*302-305*/ SL    axis_g_motor_position;
00557     /*306-309*/ SL    axis_g_position_error;
00558     /*310-313*/ SL    axis_g_aux_position;
00559     /*314-317*/ SL    axis_g_velocity;
00560     /*318-321*/ SL    axis_g_torque;
00561     /*322-323*/ UW  axis_g_analog_in;
00562     /*324*/     UB    axis_g_reserved_0;
00563     /*325*/     UB    axis_g_reserved_1;
00564     /*326-329*/ SL    axis_g_variable;
00565
00566     /*330-331*/ UW    axis_h_status;
00567     /*332*/     UB    axis_h_switches;
```

```
00568     /*333*/     UB    axis_h_stop_code;
00569     /*334-337*/ SL    axis_h_reference_position;
00570     /*338-341*/ SL    axis_h_motor_position;
00571     /*342-345*/ SL    axis_h_position_error;
00572     /*346-349*/ SL    axis_h_aux_position;
00573     /*350-353*/ SL    axis_h_velocity;
00574     /*354-357*/ SL    axis_h_torque;
00575     /*358-359*/ UW  axis_h_analog_in;
00576     /*360*/     UB    axis_h_reserved_0;
00577     /*361*/     UB    axis_h_reserved_1;
00578     /*362-365*/ SL    axis_h_variable;
00579 }; //DataRecord1806
00580
00582 struct GDataRecord2103
00583 {
00584
00585     /*Offset   type name        description*/
00586
00587     /*00*/      UB    header_0;
00588     /*01*/      UB    header_1;
00589     /*02*/      UB    header_2;
00590     /*03*/      UB    header_3;
00591
00592     /*04-05*/   UW    sample_number;
00593
00594     /*06*/      UB    input_bank_0;
00595     /*07*/      UB    input_bank_1;
00596     /*08*/      UB    input_bank_2;
00597     /*09*/      UB    input_bank_3;
00598     /*10*/      UB    input_bank_4;
00599     /*11*/      UB    input_bank_5;
00600     /*12*/      UB    input_bank_6;
00601     /*13*/      UB    input_bank_7;
00602     /*14*/      UB    input_bank_8;
00603     /*15*/      UB    input_bank_9;
00604
00605     /*16*/      UB    output_bank_0;
00606     /*17*/      UB    output_bank_1;
00607     /*18*/      UB    output_bank_2;
00608     /*19*/      UB    output_bank_3;
00609     /*20*/      UB    output_bank_4;
00610     /*21*/      UB    output_bank_5;
00611     /*22*/      UB    output_bank_6;
00612     /*23*/      UB    output_bank_7;
00613     /*24*/      UB    output_bank_8;
00614     /*25*/      UB    output_bank_9;
00615
00616     /*26*/      UB    error_code;
00617     /*27*/      UB    general_status;
00618
00619     /*28-29*/   UW    s_plane_segment_count;
00620     /*30-31*/   UW    s_plane_move_status;
00621     /*32-35*/   SL    s_distance;
00622
00623     /*36-37*/   UW    t_plane_segment_count;
00624     /*38-39*/   UW    t_plane_move_status;
00625     /*40-43*/   SL    t_distance;
00626
00627     /*44-45*/   UW    axis_a_status;
00628     /*46*/      UB    axis_a_switches;
00629     /*47*/      UB    axis_a_stop_code;
00630     /*48-51*/   SL    axis_a_reference_position;
00631     /*52-55*/   SL    axis_a_motor_position;
00632     /*56-59*/   SL    axis_a_position_error;
00633     /*60-63*/   SL    axis_a_aux_position;
00634     /*64-67*/   SL    axis_a_velocity;
00635     /*68-69*/   SW    axis_a_torque;
00636     /*70-71*/   UW  axis_a_analog_in;
00637
00638     /*72-73*/   UW    axis_b_status;
00639     /*74*/      UB    axis_b_switches;
00640     /*75*/      UB    axis_b_stop_code;
00641     /*76-79*/   SL    axis_b_reference_position;
00642     /*80-83*/   SL    axis_b_motor_position;
00643     /*84-87*/   SL    axis_b_position_error;
00644     /*88-91*/   SL    axis_b_aux_position;
00645     /*92-95*/   SL    axis_b_velocity;
00646     /*96-97*/   SW    axis_b_torque;
00647     /*98-99*/   UW  axis_b_analog_in;
00648
00649     /*100-101*/ UW    axis_c_status;
00650     /*102*/     UB    axis_c_switches;
00651     /*103*/     UB    axis_c_stop_code;
00652     /*104-107*/ SL    axis_c_reference_position;
00653     /*108-111*/ SL    axis_c_motor_position;
00654     /*112-115*/ SL    axis_c_position_error;
00655     /*116-119*/ SL    axis_c_aux_position;
```

```
00656     /*120-123*/ SL    axis_c_velocity;
00657     /*124-125*/ SW    axis_c_torque;
00658     /*126-127*/ UW    axis_c_analog_in;
00659
00660     /*128-129*/ UW    axis_d_status;
00661     /*130*/     UB    axis_d_switches;
00662     /*131*/     UB    axis_d_stop_code;
00663     /*132-135*/ SL    axis_d_reference_position;
00664     /*136-139*/ SL    axis_d_motor_position;
00665     /*140-143*/ SL    axis_d_position_error;
00666     /*144-147*/ SL    axis_d_aux_position;
00667     /*148-151*/ SL    axis_d_velocity;
00668     /*152-153*/ SW    axis_d_torque;
00669     /*154-155*/ UW  axis_d_analog_in;
00670
00671     /*156-157*/ UW    axis_e_status;
00672     /*158*/     UB    axis_e_switches;
00673     /*159*/     UB    axis_e_stop_code;
00674     /*160-163*/ SL    axis_e_reference_position;
00675     /*164-167*/ SL    axis_e_motor_position;
00676     /*168-171*/ SL    axis_e_position_error;
00677     /*172-175*/ SL    axis_e_aux_position;
00678     /*176-179*/ SL    axis_e_velocity;
00679     /*180-181*/ SW    axis_e_torque;
00680     /*182-183*/ UW  axis_e_analog_in;
00681
00682     /*184-185*/ UW    axis_f_status;
00683     /*186*/     UB    axis_f_switches;
00684     /*187*/     UB    axis_f_stop_code;
00685     /*188-191*/ SL    axis_f_reference_position;
00686     /*192-195*/ SL    axis_f_motor_position;
00687     /*196-199*/ SL    axis_f_position_error;
00688     /*200-203*/ SL    axis_f_aux_position;
00689     /*204-207*/ SL    axis_f_velocity;
00690     /*208-209*/ SW    axis_f_torque;
00691     /*210-211*/ UW    axis_f_analog_in;
00692
00693     /*212-213*/ UW    axis_g_status;
00694     /*214*/     UB    axis_g_switches;
00695     /*215*/     UB    axis_g_stop_code;
00696     /*216-219*/ SL    axis_g_reference_position;
00697     /*220-223*/ SL    axis_g_motor_position;
00698     /*224-227*/ SL    axis_g_position_error;
00699     /*228-231*/ SL    axis_g_aux_position;
00700     /*232-235*/ SL    axis_g_velocity;
00701     /*236-237*/ SW    axis_g_torque;
00702     /*238-239*/ UW  axis_g_analog_in;
00703
00704     /*240-241*/ UW    axis_h_status;
00705     /*242*/     UB    axis_h_switches;
00706     /*243*/     UB    axis_h_stop_code;
00707     /*244-247*/ SL    axis_h_reference_position;
00708     /*248-251*/ SL    axis_h_motor_position;
00709     /*252-255*/ SL    axis_h_position_error;
00710     /*256-259*/ SL    axis_h_aux_position;
00711     /*260-263*/ SL    axis_h_velocity;
00712     /*264-265*/ SW    axis_h_torque;
00713     /*266-267*/ UW  axis_h_analog_in;
00714 }; //DataRecord2013
00715
00718
00723 struct GDataRecord1802
00724 {
00725
00726     /*Offset   type name        description*/
00727
00728     /*00-01*/   UW    sample_number;
00729
00730     /*02*/      UB    input_bank_0;
00731     /*03*/      UB    input_bank_1;
00732     /*04*/      UB    input_bank_2;
00733     /*05*/      UB    input_bank_3;
00734     /*06*/      UB    input_bank_4;
00735     /*07*/      UB    input_bank_5;
00736     /*08*/      UB    input_bank_6;
00737     /*09*/      UB    input_bank_7;
00738     /*10*/      UB    input_bank_8;
00739     /*11*/      UB    input_bank_9;
00740
00741     /*12*/      UB    output_bank_0;
00742     /*13*/      UB    output_bank_1;
00743     /*14*/      UB    output_bank_2;
00744     /*15*/      UB    output_bank_3;
00745     /*16*/      UB    output_bank_4;
00746     /*17*/      UB    output_bank_5;
00747     /*18*/      UB    output_bank_6;
00748     /*19*/      UB    output_bank_7;
```

```
00749     /*20*/      UB    output_bank_8;
00750     /*21*/      UB    output_bank_9;
00751
00752     /*22*/      UB    error_code;
00753     /*23*/      UB    general_status;
00754
00755     /*24-25*/   UW    s_plane_segment_count;
00756     /*26-27*/   UW    s_plane_move_status;
00757     /*28-31*/   SL    s_distance;
00758
00759     /*32-33*/   UW    t_plane_segment_count;
00760     /*34-35*/   UW    t_plane_move_status;
00761     /*36-39*/   SL    t_distance;
00762
00763     /*40-41*/   UW    axis_a_status;
00764     /*42*/      UB    axis_a_switches;
00765     /*43*/      UB    axis_a_stop_code;
00766     /*44-47*/   SL    axis_a_reference_position;
00767     /*48-51*/   SL    axis_a_motor_position;
00768     /*52-55*/   SL    axis_a_position_error;
00769     /*56-59*/   SL    axis_a_aux_position;
00770     /*60-63*/   SL    axis_a_velocity;
00771     /*64-65*/   SW    axis_a_torque;
00772     /*66*/      UB    axis_a_reserved_0;
00773     /*67*/      UB    axis_a_reserved_1;
00774
00775     /*68-69*/   UW    axis_b_status;
00776     /*70*/      UB    axis_b_switches;
00777     /*71*/      UB    axis_b_stop_code;
00778     /*72-75*/   SL    axis_b_reference_position;
00779     /*76-79*/   SL    axis_b_motor_position;
00780     /*80-83*/   SL    axis_b_position_error;
00781     /*84-87*/   SL    axis_b_aux_position;
00782     /*88-91*/   SL    axis_b_velocity;
00783     /*92-93*/   SW    axis_b_torque;
00784     /*94*/      UB    axis_b_reserved_0;
00785     /*95*/      UB    axis_b_reserved_1;
00786
00787     /*96-97*/   UW    axis_c_status;
00788     /*98*/      UB    axis_c_switches;
00789     /*99*/      UB    axis_c_stop_code;
00790     /*100-103*/ SL    axis_c_reference_position;
00791     /*104-107*/ SL    axis_c_motor_position;
00792     /*108-111*/ SL    axis_c_position_error;
00793     /*112-115*/ SL    axis_c_aux_position;
00794     /*116-119*/ SL    axis_c_velocity;
00795     /*120-121*/ SW    axis_c_torque;
00796     /*122*/     UB    axis_c_reserved_0;
00797     /*123*/     UB    axis_c_reserved_1;
00798
00799     /*124-125*/ UW    axis_d_status;
00800     /*126*/     UB    axis_d_switches;
00801     /*127*/     UB    axis_d_stop_code;
00802     /*128-131*/ SL    axis_d_reference_position;
00803     /*132-135*/ SL    axis_d_motor_position;
00804     /*136-139*/ SL    axis_d_position_error;
00805     /*140-143*/ SL    axis_d_aux_position;
00806     /*144-147*/ SL    axis_d_velocity;
00807     /*148-149*/ SW    axis_d_torque;
00808     /*150*/     UB    axis_d_reserved_0;
00809     /*151*/     UB    axis_d_reserved_1;
00810
00811 }; //DataRecord1802
00812
00814 struct GDataRecord30000
00815 {
00816
00817     /*Offset   type name        description*/
00818
00819     /*00*/      UB    header_0;
00820     /*01*/      UB    header_1;
00821     /*02*/      UB    header_2;
00822     /*03*/      UB    header_3;
00823
00824     /*04-05*/   UW    sample_number;
00825
00826     /*06*/      UB    input_bank_0;
00827     /*07*/      UB    input_bank_1;
00828
00829     /*08*/      UB    output_bank_0;
00830     /*09*/      UB    output_bank_1;
00831
00832     /*10*/      UB    error_code;
00833     /*11*/      UB    thread_status;
00834
00835     /*12-13*/   UW    input_analog_2;
00836
```

```
00837     /*14-15*/   UW    output_analog_1;
00838     /*16-17*/   UW    output_analog_2;
00839
00840     /*18-21*/   UL    amplifier_status;
00841
00842     /*22-25*/   UL    contour_segment_count;
00843     /*26-27*/   UW    contour_buffer_available;
00844
00845     /*28-29*/   UW    s_plane_segment_count;
00846     /*30-31*/   UW    s_plane_move_status;
00847     /*32-35*/   SL    s_distance;
00848     /*36-37*/   UW    s_plane_buffer_available;
00849
00850     /*38-39*/   UW    axis_a_status;
00851     /*40*/      UB    axis_a_switches;
00852     /*41*/      UB    axis_a_stop_code;
00853     /*42-45*/   SL    axis_a_reference_position;
00854     /*46-49*/   SL    axis_a_motor_position;
00855     /*50-53*/   SL    axis_a_position_error;
00856     /*54-57*/   SL    axis_a_aux_position;
00857     /*58-61*/   SL    axis_a_velocity;
00858     /*62-65*/   SL    axis_a_torque;
00859     /*66-67*/   UW  axis_a_analog_in;
00860     /*68*/      UB    axis_a_halls;
00861     /*69*/      UB    axis_a_reserved;
00862     /*70-73*/   SL    axis_a_variable;
00863 }; //DataRecord30000
00864
00866 struct GDataRecord47000_ENC
00867 {
00868
00869     /*Offset   type name        description*/
00870
00871     /*00*/      UB    header_0;
00872     /*01*/      UB    header_1;
00873     /*02*/      UB    header_2;
00874     /*03*/      UB    header_3;
00875
00876     /*04-05*/   UW    sample_number;
00877     /*06*/      UB    error_code;
00878     /*07*/      UB    general_status;
00879
00880     /*08-09*/   UW    output_analog_0;
00881     /*10-11*/   UW    output_analog_1;
00882     /*12-13*/   UW    output_analog_2;
00883     /*14-15*/   UW    output_analog_3;
00884     /*16-17*/   UW    output_analog_4;
00885     /*18-19*/   UW    output_analog_5;
00886     /*20-21*/   UW    output_analog_6;
00887     /*22-23*/   UW    output_analog_7;
00888
00889     /*24-25*/   UW    input_analog_0;
00890     /*26-27*/   UW    input_analog_1;
00891     /*28-29*/   UW    input_analog_2;
00892     /*30-31*/   UW    input_analog_3;
00893     /*32-33*/   UW    input_analog_4;
00894     /*34-35*/   UW    input_analog_5;
00895     /*36-37*/   UW    input_analog_6;
00896     /*38-39*/   UW    input_analog_7;
00897
00898     /*40-41*/   UW    output_bank_0;
00899
00900     /*42-43*/   UW    input_bank_0;
00901
00902     /*44-47*/   UL    pulse_count_0;
00903     /*48-51*/   SL    zc_variable;
00904     /*52-55*/   SL    zd_variable;
00905
00906     /*56-59*/   SL    encoder_0;
00907     /*60-63*/   SL    encoder_1;
00908     /*64-67*/   SL    encoder_2;
00909     /*68-71*/   SL    encoder_3;
00910
00911 }; //GDataRecord47000_ENC
00912
00914 struct GDataRecord47300_ENC
00915 {
00916
00917     /*Offset   type name        description*/
00918
00919     /*00*/      UB    header_0;
00920     /*01*/      UB    header_1;
00921     /*02*/      UB    header_2;
00922     /*03*/      UB    header_3;
00923
00924     /*04-05*/   UW    sample_number;
00925     /*06*/      UB    error_code;
```

```
00926     /*07*/      UB    general_status;
00927
00928     /*08-09*/   UW    output_analog_0;
00929     /*10-11*/   UW    output_analog_1;
00930     /*12-13*/   UW    output_analog_2;
00931     /*14-15*/   UW    output_analog_3;
00932     /*16-17*/   UW    output_analog_4;
00933     /*18-19*/   UW    output_analog_5;
00934     /*20-21*/   UW    output_analog_6;
00935     /*22-23*/   UW    output_analog_7;
00936
00937     /*24-25*/   UW    input_analog_0;
00938     /*26-27*/   UW    input_analog_1;
00939     /*28-29*/   UW    input_analog_2;
00940     /*30-31*/   UW    input_analog_3;
00941     /*32-33*/   UW    input_analog_4;
00942     /*34-35*/   UW    input_analog_5;
00943     /*36-37*/   UW    input_analog_6;
00944     /*38-39*/   UW    input_analog_7;
00945
00946     /*40-41*/   UW    output_bank_0;
00947     /*42-43*/   UW    output_bank_1;
00948
00949     /*44-45*/   UW    input_bank_0;
00950     /*46-47*/   UW    input_bank_1;
00951
00952     /*48-51*/   UL    pulse_count_0;
00953     /*52-55*/   SL    zc_variable;
00954     /*56-59*/   SL    zd_variable;
00955
00956     /*60-63*/   SL    encoder_0;
00957     /*64-67*/   SL    encoder_1;
00958     /*68-71*/   SL    encoder_2;
00959     /*72-75*/   SL    encoder_3;
00960
00961 }; //GDataRecord47300_ENC
00962
00964 struct GDataRecord47300_24EX
00965 {
00966
00967     /*Offset   type name        description*/
00968
00969     /*00*/      UB    header_0;
00970     /*01*/      UB    header_1;
00971     /*02*/      UB    header_2;
00972     /*03*/      UB    header_3;
00973
00974     /*04-05*/   UW    sample_number;
00975     /*06*/      UB    error_code;
00976     /*07*/      UB    general_status;
00977
00978     /*08-09*/   UW    output_analog_0;
00979     /*10-11*/   UW    output_analog_1;
00980     /*12-13*/   UW    output_analog_2;
00981     /*14-15*/   UW    output_analog_3;
00982     /*16-17*/   UW    output_analog_4;
00983     /*18-19*/   UW    output_analog_5;
00984     /*20-21*/   UW    output_analog_6;
00985     /*22-23*/   UW    output_analog_7;
00986
00987     /*24-25*/   UW    input_analog_0;
00988     /*26-27*/   UW    input_analog_1;
00989     /*28-29*/   UW    input_analog_2;
00990     /*30-31*/   UW    input_analog_3;
00991     /*32-33*/   UW    input_analog_4;
00992     /*34-35*/   UW    input_analog_5;
00993     /*36-37*/   UW    input_analog_6;
00994     /*38-39*/   UW    input_analog_7;
00995
00996     /*40-41*/   UW    output_bank_0;
00997     /*42-43*/   UW    output_bank_1;
00998
00999     /*44-45*/   UW    input_bank_0;
01000     /*46-47*/   UW    input_bank_1;
01001
01002     /*48-51*/   UL    pulse_count_0;
01003     /*52-55*/   SL    zc_variable;
01004     /*56-59*/   SL    zd_variable;
01005
01006     /*60-61*/   UW output_bank_2;
01007     /*62-63*/   UW output_back_3;
01008
01009     /*64-65*/   UW input_bank_2;
01010     /*66-67*/   UW input_bank_3;
01011
01012 }; //GDataRecord47300_24EX
01013
```

```
01015 struct GDataRecord47162
01016 {
01017     /*Offset   type name        description*/
01018
01019     /*00*/      UB    header_0;
01020     /*01*/      UB    header_1;
01021     /*02*/      UB    header_2;
01022     /*03*/      UB    header_3;
01023
01024     /*04-05*/   UW    sample_number;
01025     /*06*/      UB    error_code;
01026     /*07*/      UB    general_status;
01027
01028     /*08-09*/   UW    output_analog_0;
01029     /*10-11*/   UW    output_analog_1;
01030     /*12-13*/   UW    output_analog_2;
01031     /*14-15*/   UW    output_analog_3;
01032     /*16-17*/   UW    output_analog_4;
01033     /*18-19*/   UW    output_analog_5;
01034     /*20-21*/   UW    output_analog_6;
01035     /*22-23*/   UW    output_analog_7;
01036
01037     /*24-25*/   UW    input_analog_0;
01038     /*26-27*/   UW    input_analog_1;
01039     /*28-29*/   UW    input_analog_2;
01040     /*30-31*/   UW    input_analog_3;
01041     /*32-33*/   UW    input_analog_4;
01042     /*34-35*/   UW    input_analog_5;
01043     /*36-37*/   UW    input_analog_6;
01044     /*38-39*/   UW    input_analog_7;
01045
01046     /*40*/      UB    output_byte_0;
01047     /*41*/      UB    output_byte_1;
01048     /*42*/      UB    output_byte_2;
01049
01050     /*43*/      UB    input_byte_0;
01051     /*44*/      UB    input_byte_1;
01052     /*45*/      UB    input_byte_2;
01053     /*46*/      UB    input_byte_3;
01054     /*47*/      UB    input_byte_4;
01055
01056     /*48-51*/   UL    pulse_count_0;
01057     /*52-55*/   SL    zc_variable;
01058     /*56-59*/   SL    zd_variable;
01059
01060     /*60-63*/   SL    encoder_0;
01061     /*64-67*/   SL    encoder_1;
01062     /*68-71*/   SL    encoder_2;
01063     /*72-75*/   SL    encoder_3;
01064
01065 }; //GDataRecord47162
01066
01068
01078 union GDataRecord
01079 {
01080     struct GDataRecord4000 dmc4000;
01081     struct GDataRecord4000 dmc4103;
01082     struct GDataRecord4000 dmc50000;
01083
01084     struct GDataRecord52000 dmc52000;
01085
01086     struct GDataRecord30000 dmc30000;
01087
01088     struct GDataRecord2103 dmc2103;
01089
01090     struct GDataRecord1806 dmc1806;
01091
01092     struct GDataRecord1802 dmc1802;
01093
01094     struct GDataRecord47000_ENC rio47000;
01095     struct GDataRecord47300_ENC rio47300;
01096     struct GDataRecord47300_24EX rio47300_24ex;
01097     struct GDataRecord47162 rio47162;
01098
01099     unsigned char byte_array[GALILDATARECORDMAXLENGTH];
01100 };
01101
01102
01103 #ifdef PACKOK
01104 #pragma pack() //return pack to default
01105 #else
01106 #error "Need to return structure packing for compiler"
01107 #endif
01108
01109 #endif //I_210405D9_D9EF_484F_8258_BB29A1BBA217
```

14.13 gclibo.h File Reference

```
#include "gclib.h"
```

Macros

- #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.

Functions

- GCLIB_DEPRECATED void GSleep (unsigned int timeout_ms)
- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn 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_DEPRECATED GReturn GInfo (GCon g, GCStringOut info, GSize info_len)

  Uses GUtility() and G_UTIL_INFO to provide a useful connection string.
- GCLIB_DEPRECATED GReturn GTimeout (GCon g, short timeout_ms)

  Uses GUtility() and G_UTIL_TIMEOUT_OVERRIDE to set the library timeout.
- GCLIB_DEPRECATED GReturn GCmd (GCon g, GCStringIn command)

  Wrapper around GCommand for use when the return value is not desired.
- GCLIB_DEPRECATED GReturn GCmdT (GCon g, GCStringIn command, GCStringOut trimmed_response, GSize response_len, GCStringOut *front)

  Wrapper around GCommand that trims the response.
- GCLIB_DEPRECATED GReturn GCmdI (GCon g, GCStringIn command, int *value)

  Wrapper around GCommand that provides the return value of a command parsed into an int.
- GCLIB_DEPRECATED GReturn GCmdD (GCon g, GCStringIn command, double *value)

  Wrapper around GCommand that provides the return value of a command parsed into a double.
- GCLIB_DEPRECATED GReturn GWaitForBool (GCon g, GCStringIn predicate, int trials)

  Blocking call that returns when the controller evaluates the predicate as true.
- GCLIB_DEPRECATED GReturn GMotionComplete (GCon g, GCStringIn axes)

  Blocking call that returns once all axes specified have completed their motion.
- GCLIB_DEPRECATED GReturn GRecordRate (GCon g, double period_ms)

  Sets the asynchronous data record to a user-specified period via DR.
- GCLIB_DEPRECATED GReturn GProgramDownloadFile (GCon g, GCStringIn file_path, GCStringIn preprocessor)

  Program download from file.
- GCLIB_DEPRECATED GReturn GProgramUploadFile (GCon g, GCStringIn file_path)

  Program upload to file.
- GCLIB_DEPRECATED GReturn GArrayDownloadFile (GCon g, GCStringIn file_path)

Array download from file.

- GCLIB_DEPRECATED GReturn GArrayUploadFile (GCon g, GCStringIn file_path, GCStringIn names)

  Array upload to file.

- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn GSetServer (GCStringIn server_name)

  Uses GUtility(), G_UTIL_GCAPS_SET_SERVER to set the new active server.

- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn GPublishServer (GCStringIn name, GOption publish, GOption save)

  Uses GUtility(), G_UTIL_GCAPS_PUBLISH_SERVER to publish local gcaps server to the local network.

- GCLIB_DEPRECATED GReturn 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_DEPRECATED GReturn 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_DEPRECATED GReturn 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_DEPRECATED void GError (GReturn rc, GCStringOut error, GSize error_len)

  Provides a human-readable description string for return codes.

- GCLIB_DEPRECATED GReturn GSetupDownloadFile (GCon g, GCStringIn file_path, GOption options, GCStringOut info, GSize info_len)

  Download a saved controller configuration from a file.

14.13.1 Macro Definition Documentation

14.13.1.1 MALLOCBUF

```
#define MALLOCBUF G_HUGE_BUFFER
```

Malloc used for large program and array uploads.
Definition at line 13 of file gclibo.h.

14.13.1.2 MAXPROG

```
#define MAXPROG MALLOCBUF
```

Maximum size for a program.
Definition at line 14 of file gclibo.h.

14.13.1.3 MAXARRAY

```
#define MAXARRAY MALLOCBUF
```

Maximum size for an array table upload.
Definition at line 15 of file gclibo.h.

14.13.1.4 POLLINGINTERVAL

```
#define POLLINGINTERVAL 100
```

Interval, in milliseconds, for polling commands, e.g. GWaitForBool().
Definition at line 16 of file gclibo.h.

14.13.1.5 G_USE_GCAPS

```
#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.
Definition at line 17 of file gclibo.h.

14.14 gclibo.h

Go to the documentation of this file.

```
00001 #pragma once
00002
00003 #include "gclib.h"
00004
00005 #ifndef _WIN32
00006 #pragma GCC visibility push(default)
00007 #endif
00008
00009 #ifdef __cplusplus
00010 extern "C" {
00011 #endif
00012
00013 #define MALLOCBUF G_HUGE_BUFFER
00014 #define MAXPROG MALLOCBUF
00015 #define MAXARRAY MALLOCBUF
00016 #define POLLINGINTERVAL 100
00017 #define G_USE_GCAPS
00018
00019 GCLIB_DEPRECATED void GSleep(unsigned int timeout_ms);
00027
00029 GCLIB_DEPRECATED GReturn GVersion(GCStringOut ver, GSize ver_len);
00052
00054 GCLIB_DEPRECATED GReturn GAddresses(GCStringOut addresses, GSize addresses_len);
00085
00087 GCLIB_DEPRECATED GReturn GInfo(GCon g, GCStringOut info, GSize info_len);
00102
00103
00105 GCLIB_DEPRECATED GReturn GTimeout(GCon g, short timeout_ms);
00115
00116
00118 GCLIB_DEPRECATED GReturn GCmd(GCon g, GCStringIn command);
00131
00132
00134 GCLIB_DEPRECATED GReturn GCmdT(GCon g, GCStringIn command, GCStringOut trimmed_response, GSize
      response_len, GCStringOut* front);
00151
00152
00154 GCLIB_DEPRECATED GReturn GCmdI(GCon g, GCStringIn command, int* value);
00168
00169
00171 GCLIB_DEPRECATED GReturn GCmdD(GCon g, GCStringIn command, double* value);
00186
00188 GCLIB_DEPRECATED GReturn GWaitForBool(GCon g, GCStringIn predicate, int trials);
00211
00213 GCLIB_DEPRECATED GReturn GMotionComplete(GCon g, GCStringIn axes);
00229
00231 GCLIB_DEPRECATED GReturn GRecordRate(GCon g, double period_ms);
00244
00245
00247 GCLIB_DEPRECATED GReturn GProgramDownloadFile(GCon g, GCStringIn file_path, GCStringIn preprocessor);
00258
00259
00261 GCLIB_DEPRECATED GReturn GProgramUploadFile(GCon g, GCStringIn file_path);
00271
00272
00274 GCLIB_DEPRECATED GReturn GArrayDownloadFile(GCon g, GCStringIn file_path);
00287
00288
00290 GCLIB_DEPRECATED GReturn GArrayUploadFile(GCon g, GCStringIn file_path, GCStringIn names);
00304
00305
00307 GCLIB_DEPRECATED GReturn GIpRequests(GCStringOut requests, GSize requests_len);
00334
00336 GCLIB_DEPRECATED GReturn GSetServer(GCStringIn server_name);
00354
00356 GCLIB_DEPRECATED GReturn GListServers(GCStringOut servers, GSize servers_len);
00375
00377 GCLIB_DEPRECATED GReturn GPublishServer(GCStringIn name, GOption publish, GOption save);
00399
00401 GCLIB_DEPRECATED GReturn GServerStatus(GCStringOut status, GSize status_len);
00423
00424
```

```
00426 GCLIB_DEPRECATED GReturn GRemoteConnections(GCStringOut connections, GSize connections_length);
00448
00450 GCLIB_DEPRECATED GReturn GAssign(GCStringIn ip, GCStringIn mac);
00474
00476 GCLIB_DEPRECATED void GError(GReturn rc, GCStringOut error, GSize error_len);
00485
00487 GCLIB_DEPRECATED GReturn GSetupDownloadFile(GCon g, GCStringIn file_path, GOption options, GCStringOut
      info, GSize info_len);
00545
00546 #ifdef __cplusplus
00547 } //extern "C"
00548 #endif
00549
00550 #ifndef _WIN32
00551 #pragma GCC visibility pop
00552 #endif
```

14.15 gclib.hpp File Reference

```
#include <gclib.h>
#include <string>
#include <vector>
#include <functional>
#include <system_error>
#include <bitset>
```

Classes

- class error

  Thrown when the underlying C library returns a nonzero `gclib_result`.
- struct Interrupt
- struct DataRecord
- struct DataRecord::Amp

  Holds amplifier-specific data record fields. Provided by `DataRecord.amp()`.
- struct DataRecord::Axis

  Holds axis-specific data record fields. Provided by `DataRecord.axis()`.
- struct DataRecord::CoordinatedMove

  Holds coordinated move specific data record fields. Provided by `DataRecord.coordinated_move()`.
- class Controller

  Manages a connection to a controller.

Namespaces

- namespace gclib
- namespace gclib::internal

Enumerations

- enum ConnectionType {
  Ethernet = GCLIB_ETHERNET ,
  Serial = GCLIB_SERIAL ,
  Pci = GCLIB_PCI }

  Returned by `Controller::connection_type()`.

Functions

- std::string version ()

  Installed gclib version.
- std::string addresses ()

Show address, model, and serial of detected controllers, where available. Detected controllers are newline-separated. Each line is comma-separated.

- std::string ip_requests ()

  Show MAC address, model, and serial of all controllers requesting IP addresses. Detected controllers are newline-separated. Use `assign_ip()` to assign an IP address to controllers returned by this method.

- void assign_ip (std::string mac, std::string ip)
- void force_gcaps (bool on)

  Force future library calls to use either gcaps or direct connections.

- std::string gcaps_version ()

  Installed gcaps version, as reported by current gcaps server.

- std::string list_servers ()

  List available gcaps servers separated by newline.

- std::string server ()

  Current gcaps server, or an empty string if connected to the local gcaps server.

- void set_server (std::string name={})
- std::string published ()

  Return name of local gcaps server, or an empty string if unpublished.

- void set_published (std::string name={})

  Publish local gcaps server as `name`, or unpublish if `name` is empty.

14.16 gclib.hpp

Go to the documentation of this file.

```
00001 #pragma once
00002 #include <gclib.h>
00003 #include <string>
00004 #include <vector>
00005 #include <functional>
00006 #include <system_error>
00007 #include <bitset>
00008
00009 namespace gclib {
00011     enum result {
00012         InvalidArgument = GCLIB_INVALID_ARGUMENT,
00013         InternalError = GCLIB_INTERNAL_ERROR,
00014         Timeout = GCLIB_TIMEOUT,
00015         CommandError = GCLIB_COMMAND_ERROR,
00016         BufferTooSmall = GCLIB_BUFFER_TOO_SMALL,
00017         NotConnected = GCLIB_NOT_CONNECTED,
00018         NotSubscribed = GCLIB_NOT_SUBSCRIBED
00019     };
00021     const class error_category : public std::error_category {
00022     public:
00023         using std::error_category::error_category;
00024         virtual const char* name() const noexcept override {
00025             return "gclib::error::category";
00026         }
00027         virtual std::string message(int result) const override {
00028             switch (result) {
00029                 case InvalidArgument:
00030                     return "An invalid argument was passed";
00031                 case InternalError:
00032                     return "An internal error occurred";
00033                 case Timeout:
00034                     return "The operation timed out";
00035                 case CommandError:
00036                     return "The controller returned a question mark";
00037                 case BufferTooSmall:
00038                     return "The data could not fit in the provided buffer";
00039                 case NotConnected:
00040                     return "The controller or gcaps is not connected";
00041                 case NotSubscribed:
00042                     return "Not subscribed to unsolicited data";
00043                 default:
00044                     return "Unrecognized gclib result";
00045             }
00046         }
00047     } error_category;
00049     class error : public std::system_error {
00050     public:
00051         using std::system_error::system_error;
```

```
00052     };
00053
00055     enum ConnectionType {
00056         Ethernet = GCLIB_ETHERNET,
00057         Serial = GCLIB_SERIAL,
00058         Pci = GCLIB_PCI
00059     };
00060
00062     namespace internal {
00064         const size_t HUGE_BUFFER = 524288;
00066         inline void check(gclib_result result, gclib_handle h = NULL) {
00067             if (result) {
00068                 std::string err = gclib_error(h);
00069                 if (err.empty()) {
00070                     throw error(std::error_code(static_cast<enum result>(result), error_category));
00071                 }
00072                 throw error(std::error_code(static_cast<enum result>(result), error_category), err);
00073             }
00074         }
00076         template <typename ...Args>
00077         inline std::string wrap(std::function<gclib_result(char*, size_t, Args...)> func, Args...
      args) {
00078             char* huge_buffer = static_cast<char*>(std::malloc(sizeof(char) * HUGE_BUFFER));
00079             check(func(huge_buffer, sizeof(char) * HUGE_BUFFER, args...));
00080             std::string result(huge_buffer);
00081             std::free(huge_buffer);
00082             return result;
00083         }
00084     };
00085
00088     struct Interrupt {
00089         Interrupt(gclib_interrupt_t interrupt) : type{static_cast<Type>(interrupt.type)},
      axis{interrupt.axis}, status{interrupt.status} {}
00092         enum Type {
00093             NoInterrupts = GCLIB_NO_INTERRUPTS,
00094             UserInterrupt = GCLIB_USER_INTERRUPT,
00095             MotionComplete = GCLIB_MOTION_COMPLETE,
00096             AllAxesMotionComplete = GCLIB_ALL_AXES_MOTION_COMPLETE,
00097             ExcessPositionError = GCLIB_EXCESS_POSITION_ERROR,
00098             LimitSwitch = GCLIB_LIMIT_SWITCH,
00099             WatchdogTimer = GCLIB_WATCHDOG_TIMER,
00100             ProgramStopped = GCLIB_PROGRAM_STOPPED,
00101             CommandDone = GCLIB_COMMAND_DONE,
00102             DigitalInputLow = GCLIB_DIGITAL_INPUT_LOW,
00103             AllInterrupts = GCLIB_ALL_INTERRUPTS
00104         };
00105         Type type;
00107         union {
00108             char axis;
00109             uint8_t digital_input;
00110             uint8_t user_interrupt;
00111         };
00112         uint8_t status;
00113     };
00114
00115     struct DataRecord {
00116         DataRecord(gclib_data_record_handle h) : h_(h) {}
00117         enum EthernetStatus {
00118             HandleFree = HANDLE_FREE,
00119             UdpSlave = UDP_SLAVE,
00120             TcpSlave = TCP_SLAVE,
00121             UdpMaster = UDP_MASTER,
00122             TcpMaster = TCP_MASTER,
00123             EstablishingUdp = ESTABLISHING_UDP,
00124             EstablishingTcp = ESTABLISHING_TCP
00125         };
00129         struct Amp {
00130             Amp(gclib_data_record_handle h, size_t amp) : h_{h}, amp_{amp} {}
00131             bool over_current() { return gclib_data_record_over_current(h_, amp_); }
00132             bool over_voltage() { return gclib_data_record_over_voltage(h_, amp_); }
00133             bool over_temp() { return gclib_data_record_over_temp(h_, amp_); }
00134             bool under_voltage() { return gclib_data_record_under_voltage(h_, amp_); }
00135             bool electronic_lock_out() { return gclib_data_record_electronic_lock_out(h_, amp_); }
00136         private:
00137             gclib_data_record_handle h_;
00138             size_t amp_;
00139         };
00141         struct Axis {
00142             Axis(gclib_data_record_handle h, char axis) : h_{h}, axis_{axis} {}
00143             enum ModeOfMotion {
00144                 NoMotion = GCLIB_NO_MOTION,
00145                 Contour = GCLIB_CONTOUR,
00146                 PositionAbsolute = GCLIB_POSITION_ABSOLUTE,
00147                 PositionRelative = GCLIB_POSITION_RELATIVE,
00148                 FindEdge = GCLIB_FIND_EDGE,
00149                 FindIndex = GCLIB_FIND_INDEX,
00150                 Home = GCLIB_HOME,
```

```
00151                 VectorMove = GCLIB_VECTOR_MOVE,
00152                 LinearMove = GCLIB_LINEAR_MOVE
00153             };
00154             bool motor_off() { return gclib_data_record_motor_off(h_, axis_); }
00155             bool latch_armed() { return gclib_data_record_latch_armed(h_, axis_); }
00156             bool final_deceleration() { return gclib_data_record_final_deceleration(h_, axis_); }
00157             bool stopping() { return gclib_data_record_stopping(h_, axis_); }
00158             bool slewing() { return gclib_data_record_slewing(h_, axis_); }
00159             ModeOfMotion mode_of_motion() { return
      static_cast<ModeOfMotion>(gclib_data_record_mode_of_motion(h_, axis_)); }
00160             bool negative_direction_move() { return gclib_data_record_negative_direction_move(h_,
      axis_); }
00161             bool moving() { return gclib_data_record_moving(h_, axis_); }
00162             int home_phase() { return gclib_data_record_home_phase(h_, axis_); }
00163             bool stepper_mode() { return gclib_data_record_stepper_mode(h_, axis_); }
00164             bool home_input() { return gclib_data_record_home_input(h_, axis_); }
00165             bool reverse_limit() { return gclib_data_record_reverse_limit(h_, axis_); }
00166             bool forward_limit() { return gclib_data_record_forward_limit(h_, axis_); }
00167             bool latch_input() { return gclib_data_record_latch_input(h_, axis_); }
00168             bool latch_occurred() { return gclib_data_record_latch_occurred(h_, axis_); }
00169             uint8_t stop_code() { return gclib_data_record_stop_code(h_, axis_); }
00170             int32_t reference_position() { return gclib_data_record_reference_position(h_, axis_); }
00171             int32_t position() { return gclib_data_record_position(h_, axis_); }
00172             int32_t position_error() { return gclib_data_record_position_error(h_, axis_); }
00173             int32_t aux_position() { return gclib_data_record_aux_position(h_, axis_); }
00174             int32_t velocity() { return gclib_data_record_velocity(h_, axis_); }
00175             double torque() { return gclib_data_record_torque(h_, axis_); }
00176             double analog_input() { return gclib_data_record_analog_input(h_, axis_); }
00177             int8_t halls() { return gclib_data_record_halls(h_, axis_); }
00178             int32_t variable() { return gclib_data_record_variable(h_, axis_); }
00179             bool hall_error() { return gclib_data_record_hall_error(h_, axis_); }
00180             bool peak_current() { return gclib_data_record_peak_current(h_, axis_); }
00181         private:
00182             gclib_data_record_handle h_;
00183             char axis_;
00184         };
00186         struct CoordinatedMove {
00187             CoordinatedMove(gclib_data_record_handle h, char plane) : h_{h}, plane_{plane} {}
00188             uint16_t segment_count() { return gclib_data_record_coordinated_move_segment_count(h_,
      plane_); }
00189             uint16_t status() { return gclib_data_record_coordinated_move_status(h_, plane_); }
00190             bool final_deceleration() { return
      gclib_data_record_coordinated_move_final_deceleration(h_, plane_); }
00191             bool stopping() { return gclib_data_record_coordinated_move_stopping(h_, plane_); }
00192             bool slewing() { return gclib_data_record_coordinated_move_slewing(h_, plane_); }
00193             bool moving() { return gclib_data_record_coordinated_move_moving(h_, plane_); }
00194             int32_t distance() { return gclib_data_record_coordinated_move_distance(h_, plane_); }
00195             uint16_t buffer_available() { return
      gclib_data_record_coordinated_move_buffer_available(h_, plane_); }
00196         private:
00197             gclib_data_record_handle h_;
00198             char plane_;
00199         };
00200         uint16_t sample() { return gclib_data_record_sample(h_); }
00201
00208         bool input(size_t index) { return gclib_data_record_input(h_, index); }
00209
00213         bool output(size_t index) { return gclib_data_record_output(h_, index); }
00214         std::bitset<8> input_bank(size_t index) { return gclib_data_record_input_bank(h_, index); }
00215         std::bitset<8> output_bank(size_t index) { return gclib_data_record_output_bank(h_, index); }
00216         uint8_t error_code() { return gclib_data_record_error_code(h_); }
00217         uint32_t contour_segment_count() { return gclib_data_record_contour_segment_count(h_); }
00218         uint16_t contour_buffer_available() { return gclib_data_record_contour_buffer_available(h_); }
00219         bool thread_running(size_t thread) { return gclib_data_record_thread_running(h_, thread); }
00220
00223         EthernetStatus ethernet_status(char handle) { return
      static_cast<EthernetStatus>(gclib_data_record_ethernet_status(h_, handle)); }
00224
00229         std::vector<uint8_t> bytes() {
00230             std::string result(1024, '\0');
00231             internal::check(gclib_data_record_bytes(h_, result.data(), result.size()));
00232             return std::vector<uint8_t>(result.begin(), result.end());
00233         }
00234         Amp amp(size_t amp) { return Amp(h_, amp); }
00235         Axis axis(char axis) { return Axis(h_, axis); }
00236         CoordinatedMove coordinated_move(char plane) { return CoordinatedMove(h_, plane); }
00237     private:
00238         gclib_data_record_handle h_;
00239     };
00240
00242     inline std::string version() { return internal::wrap({&gclib_version}); }
00243
00261     inline std::string addresses() { return internal::wrap({&gclib_addresses}); }
00262
00273     inline std::string ip_requests() { return internal::wrap({&gclib_ip_requests}); }
00274     inline void assign_ip(std::string mac, std::string ip) {
      internal::check(gclib_assign_ip(mac.data(), ip.data())); }
```

```
00275
00276
00287     inline void force_gcaps(bool on) { gclib_force_gcaps(on); }
00289     inline std::string gcaps_version() { return internal::wrap({&gclib_gcaps_version}); }
00291     inline std::string list_servers() { return internal::wrap({gclib_list_servers}); }
00293     inline std::string server() { return internal::wrap({gclib_server}); }
00296     inline void set_server(std::string name = {}) { internal::check(gclib_set_server(name.empty() ?
      nullptr : name.c_str())); }
00298     inline std::string published() { return internal::wrap({&gclib_published}); }
00300     inline void set_published(std::string name = {}) {
      internal::check(gclib_set_published(name.empty() ? nullptr : name.c_str())); }
00301
00303     class Controller {
00304     public:
00306         Controller(std::string address) : address_{address} {
00307             open();
00308             connection_type_ = connection_type();
00309         }
00311         Controller(std::string address, size_t baud_rate) : address_{address}, baud_rate_{baud_rate} {
00312             open();
00313             connection_type_ = connection_type();
00314         }
00315         ~Controller() { close(); }
00316
00318         void open() {
00319             check(gclib_open(&h_, address_.data()));
00320             if (connection_type_ == Serial) {
00321                 gclib_set_baud_rate(h_, baud_rate_);
00322             }
00323         }
00324         void close() { gclib_close(&h_); }
00325
00327         std::string address() { return wrap({&gclib_address}); }
00329         ConnectionType connection_type() { return
      static_cast<ConnectionType>(wrap<gclib_connection_t>({&gclib_connection_type})); }
00330
00332         std::string revision_information() { return wrap({&gclib_revision_information}); }
00333         uint32_t serial_number() { return wrap<uint32_t>({&gclib_serial_number}); }
00334
00344         template <typename T = std::string>
00345         T command(std::string command) {
00346             std::string str = wrap({[command](gclib_handle h, char* buf, size_t len){ return
      gclib_command(h, command.c_str(), buf, len); }});
00347             if constexpr (std::is_integral<T>() || std::is_floating_point<T>()) {
00348                 return std::stod(str);
00349             } else {
00350                 return str;
00351             }
00352         };
00353
00354         std::string program() { return wrap({&gclib_program}); }
00355         void set_program(std::string program, std::string insert = "") { check(gclib_set_program(h_,
      program.c_str(), insert.c_str())); }
00356         std::string array(std::string name, size_t start = 0, size_t end = 0) { return
      wrap({[name](gclib_handle h, char* buf, size_t len, size_t start, size_t end){ return gclib_array(h,
      name.c_str(), buf, len, start, end); }}, start, end); }
00357         void set_array(std::string name, std::string array, size_t start = 0, size_t end = 0) {
      check(gclib_set_array(h_, name.c_str(), array.c_str(), start, end)); }
00358         void set_firmware(std::string file_path) { check(gclib_set_firmware(h_, file_path.c_str())); }
00359
00360
00390         void set_interrupts(Interrupt::Type mask, std::bitset<8> motion_complete_axes = 0,
      std::bitset<8> digital_inputs = 0) {
00391             check(gclib_set_interrupts(h_, static_cast<gclib_interrupt_type>(mask),
      gclib_axis_flags(motion_complete_axes.to_ulong()),
      gclib_digital_input_flags(digital_inputs.to_ulong())));
00392         }
00393
00403         void set_data_records(size_t period_ms) { check(gclib_set_data_records(h_, period_ms)); }
00404
00420         void subscribe_messages(std::function<void(std::string message)> callback) {
00421             message_callback_ = std::function([callback](const char* message){ callback(message); });
00422             check(gclib_subscribe_messages(h_, wrap_message_callback, &message_callback_));
00423         }
00424
00433         void subscribe_messages() {
00434             check(gclib_subscribe_messages(h_, nullptr, nullptr));
00435         }
00436
00453         void subscribe_interrupts(std::function<void(Interrupt interrupt)> callback) {
00454             interrupt_callback_ = std::function([callback](gclib_interrupt_t interrupt){
      callback(interrupt); });
00455             check(gclib_subscribe_interrupts(h_, wrap_interrupt_callback, &interrupt_callback_));
00456         }
00457
00469         void subscribe_interrupts() {
00470             check(gclib_subscribe_interrupts(h_, nullptr, nullptr));
```

```
00471         }
00472
00489         void subscribe_data_records(std::function<void(DataRecord data_record)> callback) {
00490             data_record_callback_ = std::function([callback](gclib_data_record_handle h){
    callback(DataRecord(h)); });
00491             check(gclib_subscribe_data_records(h_, wrap_data_record_callback,
    &data_record_callback_));
00492         }
00493
00502         void subscribe_data_records() {
00503             check(gclib_subscribe_data_records(h_, nullptr, nullptr));
00504         };
00505
00518         void subscribe_progress(std::function<void(size_t current, size_t max)> callback) {
00519             progress_callback_ = callback;
00520             check(gclib_subscribe_progress(h_, wrap_progress_callback, &progress_callback_));
00521         };
00522
00523         void unsubscribe_messages() { check(gclib_unsubscribe(h_, (void*)&wrap_message_callback)); }
00524         void unsubscribe_interrupts() { check(gclib_unsubscribe(h_, (void*)&wrap_interrupt_callback));
    }
00525         void unsubscribe_data_records() { check(gclib_unsubscribe(h_,
    (void*)&wrap_data_record_callback)); }
00526         void unsubscribe_progress() { check(gclib_unsubscribe(h_, (void*)&wrap_progress_callback)); }
00527
00530         std::string message(int timeout = -1) { return wrap({&gclib_message}, timeout); }
00533         Interrupt interrupt(int timeout = -1) { return wrap<gclib_interrupt_t>({&gclib_interrupt},
    timeout); }
00536         DataRecord data_record(int timeout = -1) { return
    wrap<gclib_data_record_handle>({&gclib_data_record}, timeout); }
00537
00538     private:
00539         std::string address_;
00540         size_t baud_rate_{115200};
00541         ConnectionType connection_type_;
00542         void check(gclib_result result) {
00543             internal::check(result, h_);
00544         }
00545         template <typename ...Args>
00546         std::string wrap(std::function<gclib_result(gclib_handle, char*, size_t, Args...)> func,
    Args... args) {
00547             char* huge_buffer = static_cast<char*>(std::malloc(sizeof(char) * internal::HUGE_BUFFER));
00548             check(func(h_, huge_buffer, sizeof(char) * internal::HUGE_BUFFER, args...));
00549             std::string result(huge_buffer);
00550             std::free(huge_buffer);
00551             return result;
00552         }
00553         template <typename T, typename ...Args>
00554         T wrap(std::function<gclib_result(gclib_handle, T*, Args...)> func, Args... args) {
00555             T result;
00556             check(func(h_, &result, args...));
00557             return result;
00558         }
00559         gclib_handle h_{nullptr};
00560
00561         static inline void wrap_message_callback(void* context, const char* message){
    (*(std::function<void(const char*)>*)(context))(message); }
00562         static inline void wrap_interrupt_callback(void* context, gclib_interrupt_t interrupt){
    (*(std::function<void(gclib_interrupt_t)>*)(context))(interrupt); }
00563         static inline void wrap_data_record_callback(void* context, gclib_data_record_handle
    data_record){ (*(std::function<void(gclib_data_record_handle)>*)(context))(data_record); }
00564         static inline void wrap_progress_callback(void* context, size_t current, size_t max){
    (*(std::function<void(size_t, size_t)>*)(context))(current, max); }
00565         std::function<void(const char*)> message_callback_;
00566         std::function<void(gclib_interrupt_t)> interrupt_callback_;
00567         std::function<void(gclib_data_record_handle)> data_record_callback_;
00568         std::function<void(size_t, size_t)> progress_callback_;
00569     };
00570 }
```

Index

/builds/software/gclib/doc/README.md, 243
~Controller
    Controller, 107

address
    Controller, 107
addresses
    gclib, 98
AllAxesMotionComplete
    Interrupt, 240
AllInterrupts
    Interrupt, 240
Amp, 58
    DataRecord::Amp, 101
    gclib_data_record_electronic_lock_out, 58
    gclib_data_record_over_current, 58
    gclib_data_record_over_temp, 58
    gclib_data_record_over_voltage, 58
    gclib_data_record_under_voltage, 58
amp
    DataRecord, 116
amplifier_status
    GDataRecord30000, 172
    GDataRecord4000, 185
    GDataRecord52000, 228
analog_input
    DataRecord::Axis, 105
array
    Controller, 108
assign_ip
    gclib, 98
aux_position
    DataRecord::Axis, 105
Axis, 59
    DataRecord::Axis, 103
    gclib_data_record_analog_input, 62
    gclib_data_record_aux_position, 62
    gclib_data_record_final_deceleration, 60
    gclib_data_record_forward_limit, 61
    gclib_data_record_hall_error, 63
    gclib_data_record_halls, 63
    gclib_data_record_home_input, 61
    gclib_data_record_home_phase, 61
    gclib_data_record_latch_armed, 60
    gclib_data_record_latch_input, 61
    gclib_data_record_latch_occurred, 61
    gclib_data_record_mode_of_motion, 60
    gclib_data_record_motor_off, 60
    gclib_data_record_moving, 61
    gclib_data_record_negative_direction_move, 60
    gclib_data_record_peak_current, 63
    gclib_data_record_position, 62
    gclib_data_record_position_error, 62
    gclib_data_record_reference_position, 62
    gclib_data_record_reverse_limit, 61
    gclib_data_record_slewing, 60
    gclib_data_record_stepper_mode, 61
    gclib_data_record_stop_code, 62
    gclib_data_record_stopping, 60
    gclib_data_record_torque, 62
    gclib_data_record_variable, 63
    gclib_data_record_velocity, 62
axis
    DataRecord, 116
    gclib_interrupt_t, 118
    Interrupt, 241
axis_a_analog_in
    GDataRecord1806, 143
    GDataRecord2103, 162
    GDataRecord30000, 173
    GDataRecord4000, 187
    GDataRecord52000, 230
axis_a_aux_position
    GDataRecord1802, 127
    GDataRecord1806, 143
    GDataRecord2103, 161
    GDataRecord30000, 173
    GDataRecord4000, 186
    GDataRecord52000, 229
axis_a_halls
    GDataRecord30000, 174
    GDataRecord4000, 187
    GDataRecord52000, 230
axis_a_motor_position
    GDataRecord1802, 127
    GDataRecord1806, 143
    GDataRecord2103, 161
    GDataRecord30000, 173
    GDataRecord4000, 186
    GDataRecord52000, 229
axis_a_position_error
    GDataRecord1802, 127
    GDataRecord1806, 143
    GDataRecord2103, 161
    GDataRecord30000, 173
    GDataRecord4000, 186
    GDataRecord52000, 229
axis_a_reference_position
    GDataRecord1802, 127

GDataRecord1806, 143
GDataRecord2103, 161
GDataRecord30000, 173
GDataRecord4000, 186
GDataRecord52000, 229
axis_a_reserved
GDataRecord30000, 174
GDataRecord4000, 187
GDataRecord52000, 230
axis_a_reserved_0
GDataRecord1802, 127
GDataRecord1806, 143
axis_a_reserved_1
GDataRecord1802, 128
GDataRecord1806, 144
axis_a_status
GDataRecord1802, 127
GDataRecord1806, 142
GDataRecord2103, 161
GDataRecord30000, 173
GDataRecord4000, 186
GDataRecord52000, 229
axis_a_stop_code
GDataRecord1802, 127
GDataRecord1806, 143
GDataRecord2103, 161
GDataRecord30000, 173
GDataRecord4000, 186
GDataRecord52000, 229
axis_a_switches
GDataRecord1802, 127
GDataRecord1806, 143
GDataRecord2103, 161
GDataRecord30000, 173
GDataRecord4000, 186
GDataRecord52000, 229
axis_a_torque
GDataRecord1802, 127
GDataRecord1806, 143
GDataRecord2103, 162
GDataRecord30000, 173
GDataRecord4000, 187
GDataRecord52000, 229
axis_a_variable
GDataRecord1806, 144
GDataRecord30000, 174
GDataRecord4000, 187
GDataRecord52000, 230
axis_a_velocity
GDataRecord1802, 127
GDataRecord1806, 143
GDataRecord2103, 162
GDataRecord30000, 173
GDataRecord4000, 186
GDataRecord52000, 229
axis_b_analog_in
GDataRecord1806, 145
GDataRecord2103, 163
GDataRecord4000, 188
GDataRecord52000, 231
axis_b_aux_position
GDataRecord1802, 128
GDataRecord1806, 144
GDataRecord2103, 162
GDataRecord4000, 188
GDataRecord52000, 231
axis_b_halls
GDataRecord4000, 188
GDataRecord52000, 231
axis_b_motor_position
GDataRecord1802, 128
GDataRecord1806, 144
GDataRecord2103, 162
GDataRecord4000, 187
GDataRecord52000, 230
axis_b_position_error
GDataRecord1802, 128
GDataRecord1806, 144
GDataRecord2103, 162
GDataRecord4000, 188
GDataRecord52000, 230
axis_b_reference_position
GDataRecord1802, 128
GDataRecord1806, 144
GDataRecord2103, 162
GDataRecord4000, 187
GDataRecord52000, 230
axis_b_reserved
GDataRecord4000, 188
GDataRecord52000, 231
axis_b_reserved_0
GDataRecord1802, 129
GDataRecord1806, 145
axis_b_reserved_1
GDataRecord1802, 129
GDataRecord1806, 145
axis_b_status
GDataRecord1802, 128
GDataRecord1806, 144
GDataRecord2103, 162
GDataRecord4000, 187
GDataRecord52000, 230
axis_b_stop_code
GDataRecord1802, 128
GDataRecord1806, 144
GDataRecord2103, 162
GDataRecord4000, 187
GDataRecord52000, 230
axis_b_switches
GDataRecord1802, 128
GDataRecord1806, 144
GDataRecord2103, 162
GDataRecord4000, 187
GDataRecord52000, 230
axis_b_torque
GDataRecord1802, 128

GDataRecord1806, 145
GDataRecord2103, 163
GDataRecord4000, 188
GDataRecord52000, 231
axis_b_variable
GDataRecord1806, 145
GDataRecord4000, 188
GDataRecord52000, 231
axis_b_velocity
GDataRecord1802, 128
GDataRecord1806, 144
GDataRecord2103, 163
GDataRecord4000, 188
GDataRecord52000, 231
axis_c_analog_in
GDataRecord1806, 146
GDataRecord2103, 164
GDataRecord4000, 189
GDataRecord52000, 232
axis_c_aux_position
GDataRecord1802, 129
GDataRecord1806, 146
GDataRecord2103, 163
GDataRecord4000, 189
GDataRecord52000, 232
axis_c_halls
GDataRecord4000, 189
GDataRecord52000, 232
axis_c_motor_position
GDataRecord1802, 129
GDataRecord1806, 145
GDataRecord2103, 163
GDataRecord4000, 189
GDataRecord52000, 232
axis_c_position_error
GDataRecord1802, 129
GDataRecord1806, 146
GDataRecord2103, 163
GDataRecord4000, 189
GDataRecord52000, 232
axis_c_reference_position
GDataRecord1802, 129
GDataRecord1806, 145
GDataRecord2103, 163
GDataRecord4000, 189
GDataRecord52000, 232
axis_c_reserved
GDataRecord4000, 189
GDataRecord52000, 232
axis_c_reserved_0
GDataRecord1802, 130
GDataRecord1806, 146
axis_c_reserved_1
GDataRecord1802, 130
GDataRecord1806, 146
axis_c_status
GDataRecord1802, 129
GDataRecord1806, 145
GDataRecord2103, 163
GDataRecord4000, 188
GDataRecord52000, 231
axis_c_stop_code
GDataRecord1802, 129
GDataRecord1806, 145
GDataRecord2103, 163
GDataRecord4000, 189
GDataRecord52000, 231
axis_c_switches
GDataRecord1802, 129
GDataRecord1806, 145
GDataRecord2103, 163
GDataRecord4000, 188
GDataRecord52000, 231
axis_c_torque
GDataRecord1802, 130
GDataRecord1806, 146
GDataRecord2103, 164
GDataRecord4000, 189
GDataRecord52000, 232
axis_c_variable
GDataRecord1806, 146
GDataRecord4000, 190
GDataRecord52000, 232
axis_c_velocity
GDataRecord1802, 129
GDataRecord1806, 146
GDataRecord2103, 164
GDataRecord4000, 189
GDataRecord52000, 232
axis_d_analog_in
GDataRecord1806, 147
GDataRecord2103, 165
GDataRecord4000, 191
GDataRecord52000, 233
axis_d_aux_position
GDataRecord1802, 130
GDataRecord1806, 147
GDataRecord2103, 164
GDataRecord4000, 190
GDataRecord52000, 233
axis_d_halls
GDataRecord4000, 191
GDataRecord52000, 234
axis_d_motor_position
GDataRecord1802, 130
GDataRecord1806, 147
GDataRecord2103, 164
GDataRecord4000, 190
GDataRecord52000, 233
axis_d_position_error
GDataRecord1802, 130
GDataRecord1806, 147
GDataRecord2103, 164
GDataRecord4000, 190
GDataRecord52000, 233
axis_d_reference_position

GDataRecord1802, 130
GDataRecord1806, 147
GDataRecord2103, 164
GDataRecord4000, 190
GDataRecord52000, 233
axis_d_reserved
GDataRecord4000, 191
GDataRecord52000, 234
axis_d_reserved_0
GDataRecord1802, 131
GDataRecord1806, 147
axis_d_reserved_1
GDataRecord1802, 131
GDataRecord1806, 147
axis_d_status
GDataRecord1802, 130
GDataRecord1806, 146
GDataRecord2103, 164
GDataRecord4000, 190
GDataRecord52000, 233
axis_d_stop_code
GDataRecord1802, 130
GDataRecord1806, 147
GDataRecord2103, 164
GDataRecord4000, 190
GDataRecord52000, 233
axis_d_switches
GDataRecord1802, 130
GDataRecord1806, 146
GDataRecord2103, 164
GDataRecord4000, 190
GDataRecord52000, 233
axis_d_torque
GDataRecord1802, 131
GDataRecord1806, 147
GDataRecord2103, 165
GDataRecord4000, 190
GDataRecord52000, 233
axis_d_variable
GDataRecord1806, 148
GDataRecord4000, 191
GDataRecord52000, 234
axis_d_velocity
GDataRecord1802, 131
GDataRecord1806, 147
GDataRecord2103, 165
GDataRecord4000, 190
GDataRecord52000, 233
axis_e_analog_in
GDataRecord1806, 149
GDataRecord2103, 166
GDataRecord4000, 192
GDataRecord52000, 235
axis_e_aux_position
GDataRecord1806, 148
GDataRecord2103, 165
GDataRecord4000, 192
GDataRecord52000, 234
axis_e_halls
GDataRecord4000, 192
GDataRecord52000, 235
axis_e_motor_position
GDataRecord1806, 148
GDataRecord2103, 165
GDataRecord4000, 191
GDataRecord52000, 234
axis_e_position_error
GDataRecord1806, 148
GDataRecord2103, 165
GDataRecord4000, 191
GDataRecord52000, 234
axis_e_reference_position
GDataRecord1806, 148
GDataRecord2103, 165
GDataRecord4000, 191
GDataRecord52000, 234
axis_e_reserved
GDataRecord4000, 192
GDataRecord52000, 235
axis_e_reserved_0
GDataRecord1806, 149
axis_e_reserved_1
GDataRecord1806, 149
axis_e_status
GDataRecord1806, 148
GDataRecord2103, 165
GDataRecord4000, 191
GDataRecord52000, 234
axis_e_stop_code
GDataRecord1806, 148
GDataRecord2103, 165
GDataRecord4000, 191
GDataRecord52000, 234
axis_e_switches
GDataRecord1806, 148
GDataRecord2103, 165
GDataRecord4000, 191
GDataRecord52000, 234
axis_e_torque
GDataRecord1806, 148
GDataRecord2103, 166
GDataRecord4000, 192
GDataRecord52000, 235
axis_e_variable
GDataRecord1806, 149
GDataRecord4000, 192
GDataRecord52000, 235
axis_e_velocity
GDataRecord1806, 148
GDataRecord2103, 166
GDataRecord4000, 192
GDataRecord52000, 235
axis_f_analog_in
GDataRecord1806, 150
GDataRecord2103, 167
GDataRecord4000, 193

GDataRecord52000, 236
axis_f_aux_position
GDataRecord1806, 150
GDataRecord2103, 166
GDataRecord4000, 193
GDataRecord52000, 236
axis_f_halls
GDataRecord4000, 193
GDataRecord52000, 236
axis_f_motor_position
GDataRecord1806, 149
GDataRecord2103, 166
GDataRecord4000, 193
GDataRecord52000, 236
axis_f_position_error
GDataRecord1806, 149
GDataRecord2103, 166
GDataRecord4000, 193
GDataRecord52000, 236
axis_f_reference_position
GDataRecord1806, 149
GDataRecord2103, 166
GDataRecord4000, 193
GDataRecord52000, 235
axis_f_reserved
GDataRecord4000, 193
GDataRecord52000, 236
axis_f_reserved_0
GDataRecord1806, 150
axis_f_reserved_1
GDataRecord1806, 150
axis_f_status
GDataRecord1806, 149
GDataRecord2103, 166
GDataRecord4000, 192
GDataRecord52000, 235
axis_f_stop_code
GDataRecord1806, 149
GDataRecord2103, 166
GDataRecord4000, 192
GDataRecord52000, 235
axis_f_switches
GDataRecord1806, 149
GDataRecord2103, 166
GDataRecord4000, 192
GDataRecord52000, 235
axis_f_torque
GDataRecord1806, 150
GDataRecord2103, 167
GDataRecord4000, 193
GDataRecord52000, 236
axis_f_variable
GDataRecord1806, 150
GDataRecord4000, 193
GDataRecord52000, 236
axis_f_velocity
GDataRecord1806, 150
GDataRecord2103, 167
GDataRecord4000, 193
GDataRecord52000, 236
axis_g_analog_in
GDataRecord1806, 151
GDataRecord2103, 168
GDataRecord4000, 194
GDataRecord52000, 237
axis_g_aux_position
GDataRecord1806, 151
GDataRecord2103, 167
GDataRecord4000, 194
GDataRecord52000, 237
axis_g_halls
GDataRecord4000, 195
GDataRecord52000, 237
axis_g_motor_position
GDataRecord1806, 151
GDataRecord2103, 167
GDataRecord4000, 194
GDataRecord52000, 237
axis_g_position_error
GDataRecord1806, 151
GDataRecord2103, 167
GDataRecord4000, 194
GDataRecord52000, 237
axis_g_reference_position
GDataRecord1806, 151
GDataRecord2103, 167
GDataRecord4000, 194
GDataRecord52000, 237
axis_g_reserved
GDataRecord4000, 195
GDataRecord52000, 238
axis_g_reserved_0
GDataRecord1806, 151
axis_g_reserved_1
GDataRecord1806, 151
axis_g_status
GDataRecord1806, 150
GDataRecord2103, 167
GDataRecord4000, 194
GDataRecord52000, 236
axis_g_stop_code
GDataRecord1806, 150
GDataRecord2103, 167
GDataRecord4000, 194
GDataRecord52000, 237
axis_g_switches
GDataRecord1806, 150
GDataRecord2103, 167
GDataRecord4000, 194
GDataRecord52000, 237
axis_g_torque
GDataRecord1806, 151
GDataRecord2103, 168
GDataRecord4000, 194
GDataRecord52000, 237
axis_g_variable

GDataRecord1806, 151
GDataRecord4000, 195
GDataRecord52000, 238
axis_g_velocity
GDataRecord1806, 151
GDataRecord2103, 168
GDataRecord4000, 194
GDataRecord52000, 237
axis_h_analog_in
GDataRecord1806, 152
GDataRecord2103, 169
GDataRecord4000, 196
GDataRecord52000, 239
axis_h_aux_position
GDataRecord1806, 152
GDataRecord2103, 168
GDataRecord4000, 195
GDataRecord52000, 238
axis_h_halls
GDataRecord4000, 196
GDataRecord52000, 239
axis_h_motor_position
GDataRecord1806, 152
GDataRecord2103, 168
GDataRecord4000, 195
GDataRecord52000, 238
axis_h_position_error
GDataRecord1806, 152
GDataRecord2103, 168
GDataRecord4000, 195
GDataRecord52000, 238
axis_h_reference_position
GDataRecord1806, 152
GDataRecord2103, 168
GDataRecord4000, 195
GDataRecord52000, 238
axis_h_reserved
GDataRecord4000, 196
GDataRecord52000, 239
axis_h_reserved_0
GDataRecord1806, 153
axis_h_reserved_1
GDataRecord1806, 153
axis_h_status
GDataRecord1806, 152
GDataRecord2103, 168
GDataRecord4000, 195
GDataRecord52000, 238
axis_h_stop_code
GDataRecord1806, 152
GDataRecord2103, 168
GDataRecord4000, 195
GDataRecord52000, 238
axis_h_switches
GDataRecord1806, 152
GDataRecord2103, 168
GDataRecord4000, 195
GDataRecord52000, 238
axis_h_torque
GDataRecord1806, 152
GDataRecord2103, 169
GDataRecord4000, 196
GDataRecord52000, 239
axis_h_variable
GDataRecord1806, 153
GDataRecord4000, 196
GDataRecord52000, 239
axis_h_velocity
GDataRecord1806, 152
GDataRecord2103, 169
GDataRecord4000, 196
GDataRecord52000, 238

buffer_available
DataRecord::CoordinatedMove, 113
byte_array
GDataRecord, 120
bytes
DataRecord, 116

C, 41
GCLIB_ALL_AXES, 44
GCLIB_ALL_AXES_MOTION_COMPLETE, 44
GCLIB_ALL_DIGITAL_INPUTS, 44
GCLIB_ALL_INTERRUPTS, 44
GCLIB_AXIS_A, 44
GCLIB_AXIS_B, 44
GCLIB_AXIS_C, 44
GCLIB_AXIS_D, 44
GCLIB_AXIS_E, 44
GCLIB_AXIS_F, 44
gclib_axis_flags, 43, 44
GCLIB_AXIS_G, 44
GCLIB_AXIS_H, 44
GCLIB_BUFFER_TOO_SMALL, 43
GCLIB_COMMAND_DONE, 44
GCLIB_COMMAND_ERROR, 43
gclib_data_record_handle, 42
GCLIB_DIGITAL_INPUT_1, 44
GCLIB_DIGITAL_INPUT_2, 44
GCLIB_DIGITAL_INPUT_3, 44
GCLIB_DIGITAL_INPUT_4, 44
GCLIB_DIGITAL_INPUT_5, 44
GCLIB_DIGITAL_INPUT_6, 44
GCLIB_DIGITAL_INPUT_7, 44
GCLIB_DIGITAL_INPUT_8, 44
gclib_digital_input_flags, 43, 44
GCLIB_DIGITAL_INPUT_LOW, 44
GCLIB_EXCESS_POSITION_ERROR, 44
gclib_gcaps_version, 45
gclib_handle, 42
GCLIB_INTERNAL_ERROR, 43
gclib_interrupt_t, 43
gclib_interrupt_type, 43
GCLIB_INVALID_ARGUMENT, 43
GCLIB_LIMIT_SWITCH, 44
GCLIB_MOTION_COMPLETE, 43

GCLIB_NO_AXES, 44
GCLIB_NO_DIGITAL_INPUTS, 44
GCLIB_NO_INTERRUPTS, 43
GCLIB_NOT_CONNECTED, 43
GCLIB_NOT_SUBSCRIBED, 43
GCLIB_PROGRAM_STOPPED, 44
gclib_result, 42, 43
GCLIB_SUCCESS, 43
GCLIB_TIMEOUT, 43
GCLIB_USER_INTERRUPT, 43
gclib_version, 45
GCLIB_WATCHDOG_TIMER, 44
C (Legacy), 73
GError, 74
GSleep, 74
GVersion, 74
close
Controller, 107
command
Controller, 108
CommandDone
Interrupt, 240
Communication, 82
GCmd, 83
GCmdD, 84
GCmdI, 84
GCmdT, 83
GCommand, 82
Connection, 45, 75
GAddresses, 77
GAssign, 79
gclib_address, 50
gclib_addresses, 47
gclib_assign_ip, 47
gclib_close, 49
gclib_connection_t, 46
gclib_connection_type, 50
gclib_dhcp_enabled, 48
gclib_error, 49
GCLIB_ETHERNET, 46
gclib_force_gcaps, 48
gclib_ip_requests, 47
gclib_open, 48
GCLIB_PCI, 46
GCLIB_SERIAL, 46
gclib_set_baud_rate, 49
gclib_set_dhcp_enabled, 48
GClose, 77
GInfo, 78
GIpRequests, 79
GOpen, 75
GTimeout, 78
connection_type
Controller, 107
ConnectionType
gclib, 98
Contour
DataRecord::Axis, 103
contour_buffer_available
DataRecord, 116
GDataRecord1806, 142
GDataRecord30000, 172
GDataRecord4000, 185
GDataRecord52000, 228
contour_segment_count
DataRecord, 116
GDataRecord1806, 141
GDataRecord30000, 172
GDataRecord4000, 185
GDataRecord52000, 228
Controller, 52, 80, 106
~Controller, 107
address, 107
array, 108
close, 107
command, 108
connection_type, 107
Controller, 107
data_record, 112
gclib_command, 53
gclib_revision_information, 53
gclib_serial_number, 53
GMotionComplete, 81
GRead, 81
GUtility, 81
GWaitForBool, 81
GWrite, 81
interrupt, 112
message, 111
open, 107
program, 108
revision_information, 107
serial_number, 107
set_array, 108
set_data_records, 109
set_firmware, 108
set_interrupts, 108
set_program, 108
subscribe_data_records, 110, 111
subscribe_interrupts, 110
subscribe_messages, 109, 110
subscribe_progress, 111
unsubscribe_data_records, 111
unsubscribe_interrupts, 111
unsubscribe_messages, 111
unsubscribe_progress, 111
Coordinated Move, 63
gclib_data_record_coordinated_move_buffer_available, 64
gclib_data_record_coordinated_move_distance, 64
gclib_data_record_coordinated_move_final_deceleration, 64
gclib_data_record_coordinated_move_moving, 64
gclib_data_record_coordinated_move_segment_count, 64

gclib_data_record_coordinated_move_slewing, 64
gclib_data_record_coordinated_move_status, 64
gclib_data_record_coordinated_move_stopping, 64
coordinated_move
DataRecord, 117
CoordinatedMove
DataRecord::CoordinatedMove, 113

Data Record, 54
ESTABLISHING_TCP, 56
ESTABLISHING_UDP, 56
GCLIB_CONTOUR, 55
gclib_data_record_bytes, 57
gclib_data_record_contour_buffer_available, 57
gclib_data_record_contour_segment_count, 57
gclib_data_record_error_code, 56
gclib_data_record_ethernet_status, 57
gclib_data_record_input, 56
gclib_data_record_input_bank, 56
gclib_data_record_output, 56
gclib_data_record_output_bank, 56
gclib_data_record_sample, 56
gclib_data_record_thread_running, 57
gclib_ethernet_status_t, 55
GCLIB_FIND_EDGE, 55
GCLIB_FIND_INDEX, 55
GCLIB_HOME, 55
GCLIB_LINEAR_MOVE, 55
gclib_mode_of_motion_t, 55
GCLIB_NO_MOTION, 55
GCLIB_POSITION_ABSOLUTE, 55
GCLIB_POSITION_RELATIVE, 55
GCLIB_VECTOR_MOVE, 55
HANDLE_FREE, 55
TCP_MASTER, 56
TCP_SLAVE, 56
UDP_MASTER, 56
UDP_SLAVE, 56
data_record
Controller, 112
DataRecord, 113
amp, 116
axis, 116
bytes, 116
contour_buffer_available, 116
contour_segment_count, 116
coordinated_move, 117
DataRecord, 115
error_code, 116
EstablishingTcp, 115
EstablishingUdp, 115
ethernet_status, 116
EthernetStatus, 114
HandleFree, 114
input, 115
input_bank, 115
output, 115
output_bank, 116
sample, 115
TcpMaster, 115
TcpSlave, 114
thread_running, 116
UdpMaster, 115
UdpSlave, 114
DataRecord::Amp, 101
Amp, 101
electronic_lock_out, 102
over_current, 101
over_temp, 101
over_voltage, 101
under_voltage, 102
DataRecord::Axis, 102
analog_input, 105
aux_position, 105
Axis, 103
Contour, 103
final_deceleration, 103
FindEdge, 103
FindIndex, 103
forward_limit, 104
hall_error, 105
halls, 105
Home, 103
home_input, 104
home_phase, 104
latch_armed, 103
latch_input, 104
latch_occurred, 104
LinearMove, 103
mode_of_motion, 104
ModeOfMotion, 103
motor_off, 103
moving, 104
negative_direction_move, 104
NoMotion, 103
peak_current, 105
position, 105
position_error, 105
PositionAbsolute, 103
PositionRelative, 103
reference_position, 105
reverse_limit, 104
slewing, 104
stepper_mode, 104
stop_code, 104
stopping, 103
torque, 105
variable, 105
VectorMove, 103
velocity, 105
DataRecord::CoordinatedMove, 112
buffer_available, 113
CoordinatedMove, 113
distance, 113
final_deceleration, 113
moving, 113

segment_count, 113
slewing, 113
status, 113
stopping, 113
Deprecated List, 27
digital_input
gclib_interrupt_t, 118
Interrupt, 241
DigitalInputLow
Interrupt, 240
distance
DataRecord::CoordinatedMove, 113
dmc1802
GDataRecord, 120
dmc1806
GDataRecord, 120
dmc2103
GDataRecord, 120
dmc30000
GDataRecord, 119
dmc4000
GDataRecord, 119
dmc4103
GDataRecord, 119
dmc50000
GDataRecord, 119
dmc52000
GDataRecord, 119

electronic_lock_out
DataRecord::Amp, 102
encoder_0
GDataRecord47000_ENC, 201
GDataRecord47162, 206
GDataRecord47300_ENC, 216
encoder_1
GDataRecord47000_ENC, 201
GDataRecord47162, 206
GDataRecord47300_ENC, 216
encoder_2
GDataRecord47000_ENC, 201
GDataRecord47162, 206
GDataRecord47300_ENC, 216
encoder_3
GDataRecord47000_ENC, 201
GDataRecord47162, 206
GDataRecord47300_ENC, 216
error, 117
error_code
DataRecord, 116
GDataRecord1802, 126
GDataRecord1806, 141
GDataRecord2103, 160
GDataRecord30000, 171
GDataRecord4000, 184
GDataRecord47000_ENC, 198
GDataRecord47162, 203
GDataRecord47300_24EX, 209
GDataRecord47300_ENC, 214
GDataRecord52000, 227
ESTABLISHING_TCP
Data Record, 56
ESTABLISHING_UDP
Data Record, 56
EstablishingTcp
DataRecord, 115
EstablishingUdp
DataRecord, 115
ethercat_bank
GDataRecord52000, 226
Ethernet
gclib, 98
ethernet_status
DataRecord, 116
ethernet_status_a
GDataRecord4000, 184
GDataRecord52000, 227
ethernet_status_b
GDataRecord4000, 184
GDataRecord52000, 227
ethernet_status_c
GDataRecord4000, 184
GDataRecord52000, 227
ethernet_status_d
GDataRecord4000, 184
GDataRecord52000, 227
ethernet_status_e
GDataRecord4000, 184
GDataRecord52000, 227
ethernet_status_f
GDataRecord4000, 184
GDataRecord52000, 227
ethernet_status_g
GDataRecord4000, 184
GDataRecord52000, 227
ethernet_status_h
GDataRecord4000, 184
GDataRecord52000, 227
EthernetStatus
DataRecord, 114
Examples, 3
examples.md, 243
ExcessPositionError
Interrupt, 240

final_deceleration
DataRecord::Axis, 103
DataRecord::CoordinatedMove, 113
FindEdge
DataRecord::Axis, 103
FindIndex
DataRecord::Axis, 103
force_gcaps
gclib, 99
forward_limit
DataRecord::Axis, 104

G_ALREADY_OPEN

gclib_errors.h, 265
G_ALREADY_OPEN_S
gclib_errors.h, 265
G_ARRAY_NOT_DIMENSIONED
gclib_errors.h, 266
G_ARRAY_NOT_DIMENSIONED_S
gclib_errors.h, 266
G_BAD_ADDRESS
gclib_errors.h, 268
G_BAD_ADDRESS_S
gclib_errors.h, 268
G_BAD_FILE
gclib_errors.h, 268
G_BAD_FILE_S
gclib_errors.h, 268
G_BAD_FIRMWARE_LOAD
gclib_errors.h, 268
G_BAD_FIRMWARE_LOAD_S
gclib_errors.h, 268
G_BAD_FULL_MEMORY
gclib_errors.h, 267
G_BAD_FULL_MEMORY_S
gclib_errors.h, 267
G_BAD_LOST_DATA
gclib_errors.h, 267
G_BAD_LOST_DATA_S
gclib_errors.h, 267
G_BAD_RESPONSE_QUESTION_MARK
gclib_errors.h, 267
G_BAD_RESPONSE_QUESTION_MARK_S
gclib_errors.h, 267
G_BAD_VALUE_RANGE
gclib_errors.h, 267
G_BAD_VALUE_RANGE_S
gclib_errors.h, 267
G_BOUNDS
gclib_compat.h, 255
G_COMMA
gclib_compat.h, 255
G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND
gclib_errors.h, 265
G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND_S
gclib_errors.h, 265
G_CONNECTION_NOT_ESTABLISHED
gclib_errors.h, 266
G_CONNECTION_NOT_ESTABLISHED_S
gclib_errors.h, 266
G_CR
gclib_compat.h, 255
G_DATA_RECORD_ERROR
gclib_errors.h, 266
G_DATA_RECORD_ERROR_S
gclib_errors.h, 266
G_DR
gclib_compat.h, 255
G_FIRMWARE_LOAD_NOT_SUPPORTED
gclib_errors.h, 266
G_FIRMWARE_LOAD_NOT_SUPPORTED_S
gclib_errors.h, 266
G_GCAPS_OPEN_ERROR
gclib_errors.h, 268
G_GCAPS_OPEN_ERROR_S
gclib_errors.h, 268
G_GCAPS_SUBSCRIPTION_ERROR
gclib_errors.h, 268
G_GCAPS_SUBSCRIPTION_ERROR_S
gclib_errors.h, 268
G_GCLIB_ERROR
gclib_errors.h, 263
G_GCLIB_ERROR_S
gclib_errors.h, 263
G_GCLIB_NON_BLOCKING_READ_EMPTY
gclib_errors.h, 264
G_GCLIB_NON_BLOCKING_READ_EMPTY_S
gclib_errors.h, 264
G_GCLIB_POLLING_FAILED
gclib_errors.h, 264
G_GCLIB_POLLING_FAILED_S
gclib_errors.h, 264
G_GCLIB_UTILITY_ERROR
gclib_errors.h, 264
G_GCLIB_UTILITY_ERROR_S
gclib_errors.h, 264
G_GCLIB_UTILITY_IP_TAKEN
gclib_errors.h, 264
G_GCLIB_UTILITY_IP_TAKEN_S
gclib_errors.h, 264
G_HUGE_BUFFER
gclib_compat.h, 256
G_ILLEGAL_DATA_IN_PROGRAM
gclib_errors.h, 266
G_ILLEGAL_DATA_IN_PROGRAM_S
gclib_errors.h, 267
G_INVALID_PREPROCESSOR_OPTIONS
gclib_errors.h, 265
G_INVALID_PREPROCESSOR_OPTIONS_S
gclib_errors.h, 265
G_LINE_BUFFER
gclib_compat.h, 256
G_NO_ERROR
gclib_errors.h, 263
G_NO_ERROR_S
gclib_errors.h, 263
G_OPEN_ERROR
gclib_errors.h, 264
G_OPEN_ERROR_S
gclib_errors.h, 265
G_PUBLISH_SERVER
gclib_compat.h, 255
G_QR
gclib_compat.h, 255
G_READ_ERROR
gclib_errors.h, 265
G_READ_ERROR_S
gclib_errors.h, 265
G_REMOVE_SERVER

gclib_compat.h, 255
G_SMALL_BUFFER
gclib_compat.h, 256
G_TIMEOUT
gclib_errors.h, 264
G_TIMEOUT_S
gclib_errors.h, 264
G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT
gclib_errors.h, 267
G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT_S
gclib_errors.h, 267
G_UNSUPPORTED_FUNCTION
gclib_errors.h, 266
G_UNSUPPORTED_FUNCTION_S
gclib_errors.h, 266
G_USE_GCAPS
gclibo.h, 285
G_USE_INITIAL_TIMEOUT
gclib_compat.h, 255, 256
G_UTIL_ADDRESSES
gclib_compat.h, 257
G_UTIL_ASSIGN
gclib_compat.h, 257
G_UTIL_DEVICE_INITIALIZE
gclib_compat.h, 257
G_UTIL_ERROR_CONTEXT
gclib_compat.h, 257
G_UTIL_GCAPS_ADDRESSES
gclib_compat.h, 257
G_UTIL_GCAPS_ADDRESSES_GET_REMEMBERED
gclib_compat.h, 258
G_UTIL_GCAPS_ADDRESSES_SET_REMEMBERED
gclib_compat.h, 259
G_UTIL_GCAPS_ASSIGN
gclib_compat.h, 258
G_UTIL_GCAPS_HOST
gclib_compat.h, 257
G_UTIL_GCAPS_IPREQUEST
gclib_compat.h, 258
G_UTIL_GCAPS_KEEPALIVE
gclib_compat.h, 257
G_UTIL_GCAPS_LIST_SERVERS
gclib_compat.h, 258
G_UTIL_GCAPS_PING
gclib_compat.h, 258
G_UTIL_GCAPS_PUBLISH_SERVER
gclib_compat.h, 258
G_UTIL_GCAPS_REMOTE_CONNECTIONS
gclib_compat.h, 258
G_UTIL_GCAPS_SERVER_INFO
gclib_compat.h, 258
G_UTIL_GCAPS_SERVER_STATUS
gclib_compat.h, 258
G_UTIL_GCAPS_SET_SERVER
gclib_compat.h, 258
G_UTIL_GCAPS_VERSION
gclib_compat.h, 257
G_UTIL_INFO
gclib_compat.h, 256
G_UTIL_IPREQUEST
gclib_compat.h, 257
G_UTIL_PING
gclib_compat.h, 257
G_UTIL_SLEEP
gclib_compat.h, 256
G_UTIL_TIMEOUT
gclib_compat.h, 256
G_UTIL_TIMEOUT_OVERRIDE
gclib_compat.h, 256
G_UTIL_VERSION
gclib_compat.h, 256
G_WRITE_ERROR
gclib_errors.h, 265
G_WRITE_ERROR_S
gclib_errors.h, 265
GAddresses
Connection, 77
Galil Communications Library (gclib), 1
Galil Connect, 50, 93
gclib_list_servers, 51
gclib_published, 51
gclib_server, 51
gclib_set_published, 52
gclib_set_server, 51
GListServers, 94
GPublishServer, 94
GRemoteConnections, 95
GServerStatus, 95
GSetServer, 93
GALILDATARECORDMAXLENGTH
gclib_record.h, 271
GArrayDownload
Memory, 86
GArrayDownloadFile
Memory, 88
GArrayUpload
Memory, 86
GArrayUploadFile
Memory, 88
GAssign
Connection, 79
GBufIn
gclib_compat.h, 259
GBufOut
gclib_compat.h, 259
gcaps_version
gclib, 99
gclib, 97
addresses, 98
assign_ip, 98
ConnectionType, 98
Ethernet, 98
force_gcaps, 99
gcaps_version, 99
ip_requests, 98
list_servers, 99

Pci, 98
published, 99
Serial, 98
server, 99
set_published, 99
set_server, 99
version, 98
gclib.h, 243
gclib.hpp, 287
gclib::internal, 100
gclib_address
Connection, 50
gclib_addresses
Connection, 47
GCLIB_ALL_AXES
C, 44
GCLIB_ALL_AXES_MOTION_COMPLETE
C, 44
GCLIB_ALL_DIGITAL_INPUTS
C, 44
GCLIB_ALL_INTERRUPTS
C, 44
gclib_array
Memory, 65
gclib_assign_ip
Connection, 47
GCLIB_AXIS_A
C, 44
GCLIB_AXIS_B
C, 44
GCLIB_AXIS_C
C, 44
GCLIB_AXIS_D
C, 44
GCLIB_AXIS_E
C, 44
GCLIB_AXIS_F
C, 44
gclib_axis_flags
C, 43, 44
GCLIB_AXIS_G
C, 44
GCLIB_AXIS_H
C, 44
GCLIB_BUFFER_TOO_SMALL
C, 43
gclib_close
Connection, 49
gclib_command
Controller, 53
GCLIB_COMMAND_DONE
C, 44
GCLIB_COMMAND_ERROR
C, 43
gclib_compat.h, 252
G_BOUNDS, 255
G_COMMA, 255
G_CR, 255
G_DR, 255
G_HUGE_BUFFER, 256
G_LINE_BUFFER, 256
G_PUBLISH_SERVER, 255
G_QR, 255
G_REMOVE_SERVER, 255
G_SMALL_BUFFER, 256
G_USE_INITIAL_TIMEOUT, 255, 256
G_UTIL_ADDRESSES, 257
G_UTIL_ASSIGN, 257
G_UTIL_DEVICE_INITIALIZE, 257
G_UTIL_ERROR_CONTEXT, 257
G_UTIL_GCAPS_ADDRESSES, 257
G_UTIL_GCAPS_ADDRESSES_GET_REMEMBERED, 258
G_UTIL_GCAPS_ADDRESSES_SET_REMEMBERED, 259
G_UTIL_GCAPS_ASSIGN, 258
G_UTIL_GCAPS_HOST, 257
G_UTIL_GCAPS_IPREQUEST, 258
G_UTIL_GCAPS_KEEPALIVE, 257
G_UTIL_GCAPS_LIST_SERVERS, 258
G_UTIL_GCAPS_PING, 258
G_UTIL_GCAPS_PUBLISH_SERVER, 258
G_UTIL_GCAPS_REMOTE_CONNECTIONS, 258
G_UTIL_GCAPS_SERVER_INFO, 258
G_UTIL_GCAPS_SERVER_STATUS, 258
G_UTIL_GCAPS_SET_SERVER, 258
G_UTIL_GCAPS_VERSION, 257
G_UTIL_INFO, 256
G_UTIL_IPREQUEST, 257
G_UTIL_PING, 257
G_UTIL_SLEEP, 256
G_UTIL_TIMEOUT, 256
G_UTIL_TIMEOUT_OVERRIDE, 256
G_UTIL_VERSION, 256
GBufIn, 259
GBufOut, 259
GCLIB_DEPRECATED, 255
GCon, 259
GCStringIn, 259
GCStringOut, 259
GMemory, 260
GOption, 259
GReturn, 259
GSize, 259
GStatus, 260
gclib_connection_t
Connection, 46
gclib_connection_type
Connection, 50
GCLIB_CONTOUR
Data Record, 55
gclib_data_record
Unsolicited Data, 73
gclib_data_record_analog_input
Axis, 62
gclib_data_record_aux_position

Axis, 62
gclib_data_record_bytes
Data Record, 57
gclib_data_record_contour_buffer_available
Data Record, 57
gclib_data_record_contour_segment_count
Data Record, 57
gclib_data_record_coordinated_move_buffer_available
Coordinated Move, 64
gclib_data_record_coordinated_move_distance
Coordinated Move, 64
gclib_data_record_coordinated_move_final_deceleration
Coordinated Move, 64
gclib_data_record_coordinated_move_moving
Coordinated Move, 64
gclib_data_record_coordinated_move_segment_count
Coordinated Move, 64
gclib_data_record_coordinated_move_slewing
Coordinated Move, 64
gclib_data_record_coordinated_move_status
Coordinated Move, 64
gclib_data_record_coordinated_move_stopping
Coordinated Move, 64
gclib_data_record_electronic_lock_out
Amp, 58
gclib_data_record_error_code
Data Record, 56
gclib_data_record_ethernet_status
Data Record, 57
gclib_data_record_final_deceleration
Axis, 60
gclib_data_record_forward_limit
Axis, 61
gclib_data_record_hall_error
Axis, 63
gclib_data_record_halls
Axis, 63
gclib_data_record_handle
C, 42
gclib_data_record_home_input
Axis, 61
gclib_data_record_home_phase
Axis, 61
gclib_data_record_input
Data Record, 56
gclib_data_record_input_bank
Data Record, 56
gclib_data_record_latch_armed
Axis, 60
gclib_data_record_latch_input
Axis, 61
gclib_data_record_latch_occurred
Axis, 61
gclib_data_record_mode_of_motion
Axis, 60
gclib_data_record_motor_off
Axis, 60
gclib_data_record_moving
Axis, 61
gclib_data_record_negative_direction_move
Axis, 60
gclib_data_record_output
Data Record, 56
gclib_data_record_output_bank
Data Record, 56
gclib_data_record_over_current
Amp, 58
gclib_data_record_over_temp
Amp, 58
gclib_data_record_over_voltage
Amp, 58
gclib_data_record_peak_current
Axis, 63
gclib_data_record_position
Axis, 62
gclib_data_record_position_error
Axis, 62
gclib_data_record_reference_position
Axis, 62
gclib_data_record_reverse_limit
Axis, 61
gclib_data_record_sample
Data Record, 56
gclib_data_record_slewing
Axis, 60
gclib_data_record_stepper_mode
Axis, 61
gclib_data_record_stop_code
Axis, 62
gclib_data_record_stopping
Axis, 60
gclib_data_record_thread_running
Data Record, 57
gclib_data_record_torque
Axis, 62
gclib_data_record_under_voltage
Amp, 58
gclib_data_record_variable
Axis, 63
gclib_data_record_velocity
Axis, 62
GCLIB_DEPRECATED
gclib_compat.h, 255
gclib_dhcp_enabled
Connection, 48
GCLIB_DIGITAL_INPUT_1
C, 44
GCLIB_DIGITAL_INPUT_2
C, 44
GCLIB_DIGITAL_INPUT_3
C, 44
GCLIB_DIGITAL_INPUT_4
C, 44
GCLIB_DIGITAL_INPUT_5
C, 44
GCLIB_DIGITAL_INPUT_6

C, 44
GCLIB_DIGITAL_INPUT_7
C, 44
GCLIB_DIGITAL_INPUT_8
C, 44
gclib_digital_input_flags
C, 43, 44
GCLIB_DIGITAL_INPUT_LOW
C, 44
gclib_error
Connection, 49
gclib_errors.h, 261
G_ALREADY_OPEN, 265
G_ALREADY_OPEN_S, 265
G_ARRAY_NOT_DIMENSIONED, 266
G_ARRAY_NOT_DIMENSIONED_S, 266
G_BAD_ADDRESS, 268
G_BAD_ADDRESS_S, 268
G_BAD_FILE, 268
G_BAD_FILE_S, 268
G_BAD_FIRMWARE_LOAD, 268
G_BAD_FIRMWARE_LOAD_S, 268
G_BAD_FULL_MEMORY, 267
G_BAD_FULL_MEMORY_S, 267
G_BAD_LOST_DATA, 267
G_BAD_LOST_DATA_S, 267
G_BAD_RESPONSE_QUESTION_MARK, 267
G_BAD_RESPONSE_QUESTION_MARK_S, 267
G_BAD_VALUE_RANGE, 267
G_BAD_VALUE_RANGE_S, 267
G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND, 265
G_COMMAND_CALLED_WITH_ILLEGAL_COMMAND_S, 265
G_CONNECTION_NOT_ESTABLISHED, 266
G_CONNECTION_NOT_ESTABLISHED_S, 266
G_DATA_RECORD_ERROR, 266
G_DATA_RECORD_ERROR_S, 266
G_FIRMWARE_LOAD_NOT_SUPPORTED, 266
G_FIRMWARE_LOAD_NOT_SUPPORTED_S, 266
G_GCAPS_OPEN_ERROR, 268
G_GCAPS_OPEN_ERROR_S, 268
G_GCAPS_SUBSCRIPTION_ERROR, 268
G_GCAPS_SUBSCRIPTION_ERROR_S, 268
G_GCLIB_ERROR, 263
G_GCLIB_ERROR_S, 263
G_GCLIB_NON_BLOCKING_READ_EMPTY, 264
G_GCLIB_NON_BLOCKING_READ_EMPTY_S, 264
G_GCLIB_POLLING_FAILED, 264
G_GCLIB_POLLING_FAILED_S, 264
G_GCLIB_UTILITY_ERROR, 264
G_GCLIB_UTILITY_ERROR_S, 264
G_GCLIB_UTILITY_IP_TAKEN, 264
G_GCLIB_UTILITY_IP_TAKEN_S, 264
G_ILLEGAL_DATA_IN_PROGRAM, 266
G_ILLEGAL_DATA_IN_PROGRAM_S, 267
G_INVALID_PREPROCESSOR_OPTIONS, 265
G_INVALID_PREPROCESSOR_OPTIONS_S, 265
G_NO_ERROR, 263
G_NO_ERROR_S, 263
G_OPEN_ERROR, 264
G_OPEN_ERROR_S, 265
G_READ_ERROR, 265
G_READ_ERROR_S, 265
G_TIMEOUT, 264
G_TIMEOUT_S, 264
G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT, 267
G_UNABLE_TO_COMPRESS_PROGRAM_TO_FIT_S, 267
G_UNSUPPORTED_FUNCTION, 266
G_UNSUPPORTED_FUNCTION_S, 266
G_WRITE_ERROR, 265
G_WRITE_ERROR_S, 265
GCLIB_ETHERNET
Connection, 46
gclib_ethernet_status_t
Data Record, 55
GCLIB_EXCESS_POSITION_ERROR
C, 44
GCLIB_FIND_EDGE
Data Record, 55
GCLIB_FIND_INDEX
Data Record, 55
gclib_force_gcaps
Connection, 48
gclib_gcaps_version
C, 45
gclib_handle
C, 42
GCLIB_HOME
Data Record, 55
GCLIB_INTERNAL_ERROR
C, 43
gclib_interrupt
Unsolicited Data, 73
gclib_interrupt_t, 117
axis, 118
C, 43
digital_input, 118
status, 118
type, 118
user_interrupt, 118
gclib_interrupt_type
C, 43
GCLIB_INVALID_ARGUMENT
C, 43
gclib_ip_requests
Connection, 47
GCLIB_LIMIT_SWITCH
C, 44
GCLIB_LINEAR_MOVE
Data Record, 55

gclib_list_servers
Galil Connect, 51
gclib_message
Unsolicited Data, 72
gclib_mode_of_motion_t
Data Record, 55
GCLIB_MOTION_COMPLETE
C, 43
GCLIB_NO_AXES
C, 44
GCLIB_NO_DIGITAL_INPUTS
C, 44
GCLIB_NO_INTERRUPTS
C, 43
GCLIB_NO_MOTION
Data Record, 55
GCLIB_NOT_CONNECTED
C, 43
GCLIB_NOT_SUBSCRIBED
C, 43
gclib_open
Connection, 48
GCLIB_PCI
Connection, 46
GCLIB_POSITION_ABSOLUTE
Data Record, 55
GCLIB_POSITION_RELATIVE
Data Record, 55
gclib_program
Memory, 65
GCLIB_PROGRAM_STOPPED
C, 44
gclib_published
Galil Connect, 51
gclib_record.h, 270
GALILDATARECORDMAXLENGTH, 271
SL, 271
SW, 271
UB, 271
UL, 271
UW, 271
gclib_result
C, 42, 43
gclib_revision_information
Controller, 53
GCLIB_SERIAL
Connection, 46
gclib_serial_number
Controller, 53
gclib_server
Galil Connect, 51
gclib_set_array
Memory, 66
gclib_set_baud_rate
Connection, 49
gclib_set_data_records
Unsolicited Data, 70
gclib_set_dhcp_enabled
Connection, 48
gclib_set_firmware
Memory, 66
gclib_set_interrupts
Unsolicited Data, 69
gclib_set_program
Memory, 65
gclib_set_published
Galil Connect, 52
gclib_set_server
Galil Connect, 51
gclib_subscribe_data_records
Unsolicited Data, 71
gclib_subscribe_interrupts
Unsolicited Data, 70
gclib_subscribe_messages
Unsolicited Data, 70
gclib_subscribe_progress
Unsolicited Data, 71
GCLIB_SUCCESS
C, 43
GCLIB_TIMEOUT
C, 43
gclib_unsubscribe
Unsolicited Data, 72
GCLIB_USER_INTERRUPT
C, 43
GCLIB_VECTOR_MOVE
Data Record, 55
gclib_version
C, 45
GCLIB_WATCHDOG_TIMER
C, 44
gclibo.h, 284
G_USE_GCAPS, 285
MALLOCBUF, 285
MAXARRAY, 285
MAXPROG, 285
POLLINGINTERVAL, 285
GClose
Connection, 77
GCmd
Communication, 83
GCmdD
Communication, 84
GCmdI
Communication, 84
GCmdT
Communication, 83
GCommand
Communication, 82
GCon
gclib_compat.h, 259
GCStringIn
gclib_compat.h, 259
GCStringOut
gclib_compat.h, 259
GDataRecord, 118

byte_array, 120
dmc1802, 120
dmc1806, 120
dmc2103, 120
dmc30000, 119
dmc4000, 119
dmc4103, 119
dmc50000, 119
dmc52000, 119
rio47000, 120
rio47162, 120
rio47300, 120
rio47300_24ex, 120
GDataRecord1802, 120
axis_a_aux_position, 127
axis_a_motor_position, 127
axis_a_position_error, 127
axis_a_reference_position, 127
axis_a_reserved_0, 127
axis_a_reserved_1, 128
axis_a_status, 127
axis_a_stop_code, 127
axis_a_switches, 127
axis_a_torque, 127
axis_a_velocity, 127
axis_b_aux_position, 128
axis_b_motor_position, 128
axis_b_position_error, 128
axis_b_reference_position, 128
axis_b_reserved_0, 129
axis_b_reserved_1, 129
axis_b_status, 128
axis_b_stop_code, 128
axis_b_switches, 128
axis_b_torque, 128
axis_b_velocity, 128
axis_c_aux_position, 129
axis_c_motor_position, 129
axis_c_position_error, 129
axis_c_reference_position, 129
axis_c_reserved_0, 130
axis_c_reserved_1, 130
axis_c_status, 129
axis_c_stop_code, 129
axis_c_switches, 129
axis_c_torque, 130
axis_c_velocity, 129
axis_d_aux_position, 130
axis_d_motor_position, 130
axis_d_position_error, 130
axis_d_reference_position, 130
axis_d_reserved_0, 131
axis_d_reserved_1, 131
axis_d_status, 130
axis_d_stop_code, 130
axis_d_switches, 130
axis_d_torque, 131
axis_d_velocity, 131
error_code, 126
general_status, 126
input_bank_0, 124
input_bank_1, 124
input_bank_2, 124
input_bank_3, 124
input_bank_4, 124
input_bank_5, 124
input_bank_6, 124
input_bank_7, 124
input_bank_8, 125
input_bank_9, 125
output_bank_0, 125
output_bank_1, 125
output_bank_2, 125
output_bank_3, 125
output_bank_4, 125
output_bank_5, 125
output_bank_6, 125
output_bank_7, 125
output_bank_8, 126
output_bank_9, 126
s_distance, 126
s_plane_move_status, 126
s_plane_segment_count, 126
sample_number, 124
t_distance, 126
t_plane_move_status, 126
t_plane_segment_count, 126
GDataRecord1806, 131
axis_a_analog_in, 143
axis_a_aux_position, 143
axis_a_motor_position, 143
axis_a_position_error, 143
axis_a_reference_position, 143
axis_a_reserved_0, 143
axis_a_reserved_1, 144
axis_a_status, 142
axis_a_stop_code, 143
axis_a_switches, 143
axis_a_torque, 143
axis_a_variable, 144
axis_a_velocity, 143
axis_b_analog_in, 145
axis_b_aux_position, 144
axis_b_motor_position, 144
axis_b_position_error, 144
axis_b_reference_position, 144
axis_b_reserved_0, 145
axis_b_reserved_1, 145
axis_b_status, 144
axis_b_stop_code, 144
axis_b_switches, 144
axis_b_torque, 145
axis_b_variable, 145
axis_b_velocity, 144
axis_c_analog_in, 146
axis_c_aux_position, 146

axis_c_motor_position, 145
axis_c_position_error, 146
axis_c_reference_position, 145
axis_c_reserved_0, 146
axis_c_reserved_1, 146
axis_c_status, 145
axis_c_stop_code, 145
axis_c_switches, 145
axis_c_torque, 146
axis_c_variable, 146
axis_c_velocity, 146
axis_d_analog_in, 147
axis_d_aux_position, 147
axis_d_motor_position, 147
axis_d_position_error, 147
axis_d_reference_position, 147
axis_d_reserved_0, 147
axis_d_reserved_1, 147
axis_d_status, 146
axis_d_stop_code, 147
axis_d_switches, 146
axis_d_torque, 147
axis_d_variable, 148
axis_d_velocity, 147
axis_e_analog_in, 149
axis_e_aux_position, 148
axis_e_motor_position, 148
axis_e_position_error, 148
axis_e_reference_position, 148
axis_e_reserved_0, 149
axis_e_reserved_1, 149
axis_e_status, 148
axis_e_stop_code, 148
axis_e_switches, 148
axis_e_torque, 148
axis_e_variable, 149
axis_e_velocity, 148
axis_f_analog_in, 150
axis_f_aux_position, 150
axis_f_motor_position, 149
axis_f_position_error, 149
axis_f_reference_position, 149
axis_f_reserved_0, 150
axis_f_reserved_1, 150
axis_f_status, 149
axis_f_stop_code, 149
axis_f_switches, 149
axis_f_torque, 150
axis_f_variable, 150
axis_f_velocity, 150
axis_g_analog_in, 151
axis_g_aux_position, 151
axis_g_motor_position, 151
axis_g_position_error, 151
axis_g_reference_position, 151
axis_g_reserved_0, 151
axis_g_reserved_1, 151
axis_g_status, 150
axis_g_stop_code, 150
axis_g_switches, 150
axis_g_torque, 151
axis_g_variable, 151
axis_g_velocity, 151
axis_h_analog_in, 152
axis_h_aux_position, 152
axis_h_motor_position, 152
axis_h_position_error, 152
axis_h_reference_position, 152
axis_h_reserved_0, 153
axis_h_reserved_1, 153
axis_h_status, 152
axis_h_stop_code, 152
axis_h_switches, 152
axis_h_torque, 152
axis_h_variable, 153
axis_h_velocity, 152
contour_buffer_available, 142
contour_segment_count, 141
error_code, 141
input_bank_0, 138
input_bank_1, 138
input_bank_2, 138
input_bank_3, 138
input_bank_4, 138
input_bank_5, 138
input_bank_6, 138
input_bank_7, 138
input_bank_8, 138
input_bank_9, 138
output_bank_0, 139
output_bank_1, 139
output_bank_2, 139
output_bank_3, 139
output_bank_4, 139
output_bank_5, 139
output_bank_6, 139
output_bank_7, 139
output_bank_8, 139
output_bank_9, 139
reserved_0, 140
reserved_10, 140
reserved_12, 140
reserved_14, 140
reserved_16, 140
reserved_17, 140
reserved_18, 141
reserved_19, 141
reserved_2, 140
reserved_20, 141
reserved_21, 141
reserved_22, 141
reserved_23, 141
reserved_24, 141
reserved_4, 140
reserved_6, 140
reserved_8, 140

s_distance, 142
s_plane_buffer_available, 142
s_plane_move_status, 142
s_plane_segment_count, 142
sample_number, 138
t_distance, 142
t_plane_buffer_available, 142
t_plane_move_status, 142
t_plane_segment_count, 142
thread_status, 141
GDataRecord2103, 153
axis_a_analog_in, 162
axis_a_aux_position, 161
axis_a_motor_position, 161
axis_a_position_error, 161
axis_a_reference_position, 161
axis_a_status, 161
axis_a_stop_code, 161
axis_a_switches, 161
axis_a_torque, 162
axis_a_velocity, 162
axis_b_analog_in, 163
axis_b_aux_position, 162
axis_b_motor_position, 162
axis_b_position_error, 162
axis_b_reference_position, 162
axis_b_status, 162
axis_b_stop_code, 162
axis_b_switches, 162
axis_b_torque, 163
axis_b_velocity, 163
axis_c_analog_in, 164
axis_c_aux_position, 163
axis_c_motor_position, 163
axis_c_position_error, 163
axis_c_reference_position, 163
axis_c_status, 163
axis_c_stop_code, 163
axis_c_switches, 163
axis_c_torque, 164
axis_c_velocity, 164
axis_d_analog_in, 165
axis_d_aux_position, 164
axis_d_motor_position, 164
axis_d_position_error, 164
axis_d_reference_position, 164
axis_d_status, 164
axis_d_stop_code, 164
axis_d_switches, 164
axis_d_torque, 165
axis_d_velocity, 165
axis_e_analog_in, 166
axis_e_aux_position, 165
axis_e_motor_position, 165
axis_e_position_error, 165
axis_e_reference_position, 165
axis_e_status, 165
axis_e_stop_code, 165
axis_e_switches, 165
axis_e_torque, 166
axis_e_velocity, 166
axis_f_analog_in, 167
axis_f_aux_position, 166
axis_f_motor_position, 166
axis_f_position_error, 166
axis_f_reference_position, 166
axis_f_status, 166
axis_f_stop_code, 166
axis_f_switches, 166
axis_f_torque, 167
axis_f_velocity, 167
axis_g_analog_in, 168
axis_g_aux_position, 167
axis_g_motor_position, 167
axis_g_position_error, 167
axis_g_reference_position, 167
axis_g_status, 167
axis_g_stop_code, 167
axis_g_switches, 167
axis_g_torque, 168
axis_g_velocity, 168
axis_h_analog_in, 169
axis_h_aux_position, 168
axis_h_motor_position, 168
axis_h_position_error, 168
axis_h_reference_position, 168
axis_h_status, 168
axis_h_stop_code, 168
axis_h_switches, 168
axis_h_torque, 169
axis_h_velocity, 169
error_code, 160
general_status, 160
header_0, 158
header_1, 158
header_2, 158
header_3, 158
input_bank_0, 158
input_bank_1, 158
input_bank_2, 158
input_bank_3, 158
input_bank_4, 158
input_bank_5, 159
input_bank_6, 159
input_bank_7, 159
input_bank_8, 159
input_bank_9, 159
output_bank_0, 159
output_bank_1, 159
output_bank_2, 159
output_bank_3, 159
output_bank_4, 159
output_bank_5, 160
output_bank_6, 160
output_bank_7, 160
output_bank_8, 160

output_bank_9, 160
s_distance, 160
s_plane_move_status, 160
s_plane_segment_count, 160
sample_number, 158
t_distance, 161
t_plane_move_status, 161
t_plane_segment_count, 161
GDataRecord30000, 169
amplifier_status, 172
axis_a_analog_in, 173
axis_a_aux_position, 173
axis_a_halls, 174
axis_a_motor_position, 173
axis_a_position_error, 173
axis_a_reference_position, 173
axis_a_reserved, 174
axis_a_status, 173
axis_a_stop_code, 173
axis_a_switches, 173
axis_a_torque, 173
axis_a_variable, 174
axis_a_velocity, 173
contour_buffer_available, 172
contour_segment_count, 172
error_code, 171
header_0, 171
header_1, 171
header_2, 171
header_3, 171
input_analog_2, 172
input_bank_0, 171
input_bank_1, 171
output_analog_1, 172
output_analog_2, 172
output_bank_0, 171
output_bank_1, 171
s_distance, 172
s_plane_buffer_available, 172
s_plane_move_status, 172
s_plane_segment_count, 172
sample_number, 171
thread_status, 171
GDataRecord4000, 174
amplifier_status, 185
axis_a_analog_in, 187
axis_a_aux_position, 186
axis_a_halls, 187
axis_a_motor_position, 186
axis_a_position_error, 186
axis_a_reference_position, 186
axis_a_reserved, 187
axis_a_status, 186
axis_a_stop_code, 186
axis_a_switches, 186
axis_a_torque, 187
axis_a_variable, 187
axis_a_velocity, 186
axis_b_analog_in, 188
axis_b_aux_position, 188
axis_b_halls, 188
axis_b_motor_position, 187
axis_b_position_error, 188
axis_b_reference_position, 187
axis_b_reserved, 188
axis_b_status, 187
axis_b_stop_code, 187
axis_b_switches, 187
axis_b_torque, 188
axis_b_variable, 188
axis_b_velocity, 188
axis_c_analog_in, 189
axis_c_aux_position, 189
axis_c_halls, 189
axis_c_motor_position, 189
axis_c_position_error, 189
axis_c_reference_position, 189
axis_c_reserved, 189
axis_c_status, 188
axis_c_stop_code, 189
axis_c_switches, 188
axis_c_torque, 189
axis_c_variable, 190
axis_c_velocity, 189
axis_d_analog_in, 191
axis_d_aux_position, 190
axis_d_halls, 191
axis_d_motor_position, 190
axis_d_position_error, 190
axis_d_reference_position, 190
axis_d_reserved, 191
axis_d_status, 190
axis_d_stop_code, 190
axis_d_switches, 190
axis_d_torque, 190
axis_d_variable, 191
axis_d_velocity, 190
axis_e_analog_in, 192
axis_e_aux_position, 192
axis_e_halls, 192
axis_e_motor_position, 191
axis_e_position_error, 191
axis_e_reference_position, 191
axis_e_reserved, 192
axis_e_status, 191
axis_e_stop_code, 191
axis_e_switches, 191
axis_e_torque, 192
axis_e_variable, 192
axis_e_velocity, 192
axis_f_analog_in, 193
axis_f_aux_position, 193
axis_f_halls, 193
axis_f_motor_position, 193
axis_f_position_error, 193
axis_f_reference_position, 193

axis_f_reserved, 193
axis_f_status, 192
axis_f_stop_code, 192
axis_f_switches, 192
axis_f_torque, 193
axis_f_variable, 193
axis_f_velocity, 193
axis_g_analog_in, 194
axis_g_aux_position, 194
axis_g_halls, 195
axis_g_motor_position, 194
axis_g_position_error, 194
axis_g_reference_position, 194
axis_g_reserved, 195
axis_g_status, 194
axis_g_stop_code, 194
axis_g_switches, 194
axis_g_torque, 194
axis_g_variable, 195
axis_g_velocity, 194
axis_h_analog_in, 196
axis_h_aux_position, 195
axis_h_halls, 196
axis_h_motor_position, 195
axis_h_position_error, 195
axis_h_reference_position, 195
axis_h_reserved, 196
axis_h_status, 195
axis_h_stop_code, 195
axis_h_switches, 195
axis_h_torque, 196
axis_h_variable, 196
axis_h_velocity, 196
contour_buffer_available, 185
contour_segment_count, 185
error_code, 184
ethernet_status_a, 184
ethernet_status_b, 184
ethernet_status_c, 184
ethernet_status_d, 184
ethernet_status_e, 184
ethernet_status_f, 184
ethernet_status_g, 184
ethernet_status_h, 184
header_0, 180
header_1, 180
header_2, 181
header_3, 181
input_bank_0, 181
input_bank_1, 181
input_bank_2, 181
input_bank_3, 181
input_bank_4, 181
input_bank_5, 181
input_bank_6, 181
input_bank_7, 182
input_bank_8, 182
input_bank_9, 182
output_bank_0, 182
output_bank_1, 182
output_bank_2, 182
output_bank_3, 182
output_bank_4, 182
output_bank_5, 182
output_bank_6, 182
output_bank_7, 183
output_bank_8, 183
output_bank_9, 183
reserved_0, 183
reserved_10, 183
reserved_12, 183
reserved_14, 184
reserved_2, 183
reserved_4, 183
reserved_6, 183
reserved_8, 183
s_distance, 185
s_plane_buffer_available, 185
s_plane_move_status, 185
s_plane_segment_count, 185
sample_number, 181
t_distance, 186
t_plane_buffer_available, 186
t_plane_move_status, 185
t_plane_segment_count, 185
thread_status, 185
GDataRecord47000_ENC, 196
encoder_0, 201
encoder_1, 201
encoder_2, 201
encoder_3, 201
error_code, 198
general_status, 198
header_0, 198
header_1, 198
header_2, 198
header_3, 198
input_analog_0, 199
input_analog_1, 199
input_analog_2, 199
input_analog_3, 200
input_analog_4, 200
input_analog_5, 200
input_analog_6, 200
input_analog_7, 200
input_bank_0, 200
output_analog_0, 198
output_analog_1, 199
output_analog_2, 199
output_analog_3, 199
output_analog_4, 199
output_analog_5, 199
output_analog_6, 199
output_analog_7, 199
output_bank_0, 200
pulse_count_0, 200

sample_number, 198
zc_variable, 200
zd_variable, 200
GDataRecord47162, 201
encoder_0, 206
encoder_1, 206
encoder_2, 206
encoder_3, 206
error_code, 203
general_status, 203
header_0, 203
header_1, 203
header_2, 203
header_3, 203
input_analog_0, 204
input_analog_1, 204
input_analog_2, 204
input_analog_3, 205
input_analog_4, 205
input_analog_5, 205
input_analog_6, 205
input_analog_7, 205
input_byte_0, 205
input_byte_1, 205
input_byte_2, 206
input_byte_3, 206
input_byte_4, 206
output_analog_0, 203
output_analog_1, 204
output_analog_2, 204
output_analog_3, 204
output_analog_4, 204
output_analog_5, 204
output_analog_6, 204
output_analog_7, 204
output_byte_0, 205
output_byte_1, 205
output_byte_2, 205
pulse_count_0, 206
sample_number, 203
zc_variable, 206
zd_variable, 206
GDataRecord47300_24EX, 207
error_code, 209
general_status, 209
header_0, 208
header_1, 208
header_2, 208
header_3, 208
input_analog_0, 210
input_analog_1, 210
input_analog_2, 210
input_analog_3, 210
input_analog_4, 210
input_analog_5, 210
input_analog_6, 210
input_analog_7, 210
input_bank_0, 211
input_bank_1, 211
input_bank_2, 211
input_bank_3, 211
output_analog_0, 209
output_analog_1, 209
output_analog_2, 209
output_analog_3, 209
output_analog_4, 209
output_analog_5, 209
output_analog_6, 209
output_analog_7, 210
output_back_3, 211
output_bank_0, 210
output_bank_1, 211
output_bank_2, 211
pulse_count_0, 211
sample_number, 209
zc_variable, 211
zd_variable, 211
GDataRecord47300_ENC, 212
encoder_0, 216
encoder_1, 216
encoder_2, 216
encoder_3, 216
error_code, 214
general_status, 214
header_0, 213
header_1, 213
header_2, 213
header_3, 213
input_analog_0, 215
input_analog_1, 215
input_analog_2, 215
input_analog_3, 215
input_analog_4, 215
input_analog_5, 215
input_analog_6, 215
input_analog_7, 215
input_bank_0, 216
input_bank_1, 216
output_analog_0, 214
output_analog_1, 214
output_analog_2, 214
output_analog_3, 214
output_analog_4, 214
output_analog_5, 214
output_analog_6, 214
output_analog_7, 215
output_bank_0, 215
output_bank_1, 216
pulse_count_0, 216
sample_number, 214
zc_variable, 216
zd_variable, 216
GDataRecord52000, 217
amplifier_status, 228
axis_a_analog_in, 230
axis_a_aux_position, 229

axis_a_halls, 230
axis_a_motor_position, 229
axis_a_position_error, 229
axis_a_reference_position, 229
axis_a_reserved, 230
axis_a_status, 229
axis_a_stop_code, 229
axis_a_switches, 229
axis_a_torque, 229
axis_a_variable, 230
axis_a_velocity, 229
axis_b_analog_in, 231
axis_b_aux_position, 231
axis_b_halls, 231
axis_b_motor_position, 230
axis_b_position_error, 230
axis_b_reference_position, 230
axis_b_reserved, 231
axis_b_status, 230
axis_b_stop_code, 230
axis_b_switches, 230
axis_b_torque, 231
axis_b_variable, 231
axis_b_velocity, 231
axis_c_analog_in, 232
axis_c_aux_position, 232
axis_c_halls, 232
axis_c_motor_position, 232
axis_c_position_error, 232
axis_c_reference_position, 232
axis_c_reserved, 232
axis_c_status, 231
axis_c_stop_code, 231
axis_c_switches, 231
axis_c_torque, 232
axis_c_variable, 232
axis_c_velocity, 232
axis_d_analog_in, 233
axis_d_aux_position, 233
axis_d_halls, 234
axis_d_motor_position, 233
axis_d_position_error, 233
axis_d_reference_position, 233
axis_d_reserved, 234
axis_d_status, 233
axis_d_stop_code, 233
axis_d_switches, 233
axis_d_torque, 233
axis_d_variable, 234
axis_d_velocity, 233
axis_e_analog_in, 235
axis_e_aux_position, 234
axis_e_halls, 235
axis_e_motor_position, 234
axis_e_position_error, 234
axis_e_reference_position, 234
axis_e_reserved, 235
axis_e_status, 234
axis_e_stop_code, 234
axis_e_switches, 234
axis_e_torque, 235
axis_e_variable, 235
axis_e_velocity, 235
axis_f_analog_in, 236
axis_f_aux_position, 236
axis_f_halls, 236
axis_f_motor_position, 236
axis_f_position_error, 236
axis_f_reference_position, 235
axis_f_reserved, 236
axis_f_status, 235
axis_f_stop_code, 235
axis_f_switches, 235
axis_f_torque, 236
axis_f_variable, 236
axis_f_velocity, 236
axis_g_analog_in, 237
axis_g_aux_position, 237
axis_g_halls, 237
axis_g_motor_position, 237
axis_g_position_error, 237
axis_g_reference_position, 237
axis_g_reserved, 238
axis_g_status, 236
axis_g_stop_code, 237
axis_g_switches, 237
axis_g_torque, 237
axis_g_variable, 238
axis_g_velocity, 237
axis_h_analog_in, 239
axis_h_aux_position, 238
axis_h_halls, 239
axis_h_motor_position, 238
axis_h_position_error, 238
axis_h_reference_position, 238
axis_h_reserved, 239
axis_h_status, 238
axis_h_stop_code, 238
axis_h_switches, 238
axis_h_torque, 239
axis_h_variable, 239
axis_h_velocity, 238
contour_buffer_available, 228
contour_segment_count, 228
error_code, 227
ethercat_bank, 226
ethernet_status_a, 227
ethernet_status_b, 227
ethernet_status_c, 227
ethernet_status_d, 227
ethernet_status_e, 227
ethernet_status_f, 227
ethernet_status_g, 227
ethernet_status_h, 227
header_0, 223
header_1, 223

header_2, 223
header_3, 223
input_bank_0, 224
input_bank_1, 224
input_bank_2, 224
input_bank_3, 224
input_bank_4, 224
input_bank_5, 224
input_bank_6, 224
input_bank_7, 224
input_bank_8, 224
input_bank_9, 225
output_bank_0, 225
output_bank_1, 225
output_bank_2, 225
output_bank_3, 225
output_bank_4, 225
output_bank_5, 225
output_bank_6, 225
output_bank_7, 225
output_bank_8, 225
output_bank_9, 226
reserved_0, 226
reserved_10, 226
reserved_12, 226
reserved_14, 226
reserved_2, 226
reserved_4, 226
reserved_6, 226
reserved_8, 226
s_distance, 228
s_plane_buffer_available, 228
s_plane_move_status, 228
s_plane_segment_count, 228
sample_number, 224
t_distance, 228
t_plane_buffer_available, 229
t_plane_move_status, 228
t_plane_segment_count, 228
thread_status, 227
general_status
GDataRecord1802, 126
GDataRecord2103, 160
GDataRecord47000_ENC, 198
GDataRecord47162, 203
GDataRecord47300_24EX, 209
GDataRecord47300_ENC, 214
GError
C (Legacy), 74
GFirmwareDownload
Memory, 87
GInfo
Connection, 78
GInterrupt
Unsolicited Data, 92
GIpRequests
Connection, 79
GListServers
Galil Connect, 94
GMemory
gclib_compat.h, 260
GMessage
Unsolicited Data, 91
GMotionComplete
Controller, 81
GOpen
Connection, 75
GOption
gclib_compat.h, 259
GProgramDownload
Memory, 85
GProgramDownloadFile
Memory, 87
GProgramUpload
Memory, 86
GProgramUploadFile
Memory, 88
GPublishServer
Galil Connect, 94
GRead
Controller, 81
GRecord
Unsolicited Data, 90
GRecordRate
Unsolicited Data, 92
GRemoteConnections
Galil Connect, 95
GReturn
gclib_compat.h, 259
GServerStatus
Galil Connect, 95
GSetServer
Galil Connect, 93
GSetupDownloadFile
Memory, 89
GSize
gclib_compat.h, 259
GSleep
C (Legacy), 74
GStatus
gclib_compat.h, 260
GTimeout
Connection, 78
GUtility
Controller, 81
GVersion
C (Legacy), 74
GWaitForBool
Controller, 81
GWrite
Controller, 81

hall_error
DataRecord::Axis, 105
halls
DataRecord::Axis, 105
HANDLE_FREE

Data Record, 55
HandleFree
DataRecord, 114
header_0
GDataRecord2103, 158
GDataRecord30000, 171
GDataRecord4000, 180
GDataRecord47000_ENC, 198
GDataRecord47162, 203
GDataRecord47300_24EX, 208
GDataRecord47300_ENC, 213
GDataRecord52000, 223
header_1
GDataRecord2103, 158
GDataRecord30000, 171
GDataRecord4000, 180
GDataRecord47000_ENC, 198
GDataRecord47162, 203
GDataRecord47300_24EX, 208
GDataRecord47300_ENC, 213
GDataRecord52000, 223
header_2
GDataRecord2103, 158
GDataRecord30000, 171
GDataRecord4000, 181
GDataRecord47000_ENC, 198
GDataRecord47162, 203
GDataRecord47300_24EX, 208
GDataRecord47300_ENC, 213
GDataRecord52000, 223
header_3
GDataRecord2103, 158
GDataRecord30000, 171
GDataRecord4000, 181
GDataRecord47000_ENC, 198
GDataRecord47162, 203
GDataRecord47300_24EX, 208
GDataRecord47300_ENC, 213
GDataRecord52000, 223
Home
DataRecord::Axis, 103
home_input
DataRecord::Axis, 104
home_phase
DataRecord::Axis, 104

input
DataRecord, 115
input_analog_0
GDataRecord47000_ENC, 199
GDataRecord47162, 204
GDataRecord47300_24EX, 210
GDataRecord47300_ENC, 215
input_analog_1
GDataRecord47000_ENC, 199
GDataRecord47162, 204
GDataRecord47300_24EX, 210
GDataRecord47300_ENC, 215
input_analog_2
GDataRecord30000, 172
GDataRecord47000_ENC, 199
GDataRecord47162, 204
GDataRecord47300_24EX, 210
GDataRecord47300_ENC, 215
input_analog_3
GDataRecord47000_ENC, 200
GDataRecord47162, 205
GDataRecord47300_24EX, 210
GDataRecord47300_ENC, 215
input_analog_4
GDataRecord47000_ENC, 200
GDataRecord47162, 205
GDataRecord47300_24EX, 210
GDataRecord47300_ENC, 215
input_analog_5
GDataRecord47000_ENC, 200
GDataRecord47162, 205
GDataRecord47300_24EX, 210
GDataRecord47300_ENC, 215
input_analog_6
GDataRecord47000_ENC, 200
GDataRecord47162, 205
GDataRecord47300_24EX, 210
GDataRecord47300_ENC, 215
input_analog_7
GDataRecord47000_ENC, 200
GDataRecord47162, 205
GDataRecord47300_24EX, 210
GDataRecord47300_ENC, 215
input_bank
DataRecord, 115
input_bank_0
GDataRecord1802, 124
GDataRecord1806, 138
GDataRecord2103, 158
GDataRecord30000, 171
GDataRecord4000, 181
GDataRecord47000_ENC, 200
GDataRecord47300_24EX, 211
GDataRecord47300_ENC, 216
GDataRecord52000, 224
input_bank_1
GDataRecord1802, 124
GDataRecord1806, 138
GDataRecord2103, 158
GDataRecord30000, 171
GDataRecord4000, 181
GDataRecord47300_24EX, 211
GDataRecord47300_ENC, 216
GDataRecord52000, 224
input_bank_2
GDataRecord1802, 124
GDataRecord1806, 138
GDataRecord2103, 158
GDataRecord4000, 181
GDataRecord47300_24EX, 211
GDataRecord52000, 224

input_bank_3
GDataRecord1802, 124
GDataRecord1806, 138
GDataRecord2103, 158
GDataRecord4000, 181
GDataRecord47300_24EX, 211
GDataRecord52000, 224
input_bank_4
GDataRecord1802, 124
GDataRecord1806, 138
GDataRecord2103, 158
GDataRecord4000, 181
GDataRecord52000, 224
input_bank_5
GDataRecord1802, 124
GDataRecord1806, 138
GDataRecord2103, 159
GDataRecord4000, 181
GDataRecord52000, 224
input_bank_6
GDataRecord1802, 124
GDataRecord1806, 138
GDataRecord2103, 159
GDataRecord4000, 181
GDataRecord52000, 224
input_bank_7
GDataRecord1802, 124
GDataRecord1806, 138
GDataRecord2103, 159
GDataRecord4000, 182
GDataRecord52000, 224
input_bank_8
GDataRecord1802, 125
GDataRecord1806, 138
GDataRecord2103, 159
GDataRecord4000, 182
GDataRecord52000, 224
input_bank_9
GDataRecord1802, 125
GDataRecord1806, 138
GDataRecord2103, 159
GDataRecord4000, 182
GDataRecord52000, 225
input_byte_0
GDataRecord47162, 205
input_byte_1
GDataRecord47162, 205
input_byte_2
GDataRecord47162, 206
input_byte_3
GDataRecord47162, 206
input_byte_4
GDataRecord47162, 206
Interrupt, 239
AllAxesMotionComplete, 240
AllInterrupts, 240
axis, 241
CommandDone, 240
digital_input, 241
DigitalInputLow, 240
ExcessPositionError, 240
Interrupt, 240
LimitSwitch, 240
MotionComplete, 240
NoInterrupts, 240
ProgramStopped, 240
status, 241
Type, 240
type, 241
user_interrupt, 241
UserInterrupt, 240
WatchdogTimer, 240
interrupt
Controller, 112
ip_requests
gclib, 98

latch_armed
DataRecord::Axis, 103
latch_input
DataRecord::Axis, 104
latch_occurred
DataRecord::Axis, 104
License, 19
license.md, 243
LimitSwitch
Interrupt, 240
LinearMove
DataRecord::Axis, 103
list_servers
gclib, 99

MALLOCBUF
gclibo.h, 285
MAXARRAY
gclibo.h, 285
MAXPROG
gclibo.h, 285
Memory, 65, 85
GArrayDownload, 86
GArrayDownloadFile, 88
GArrayUpload, 86
GArrayUploadFile, 88
gclib_array, 65
gclib_program, 65
gclib_set_array, 66
gclib_set_firmware, 66
gclib_set_program, 65
GFirmwareDownload, 87
GProgramDownload, 85
GProgramDownloadFile, 87
GProgramUpload, 86
GProgramUploadFile, 88
GSetupDownloadFile, 89
message
Controller, 111
mode_of_motion

DataRecord::Axis, 104
ModeOfMotion
DataRecord::Axis, 103
MotionComplete
Interrupt, 240
motor_off
DataRecord::Axis, 103
moving
DataRecord::Axis, 104
DataRecord::CoordinatedMove, 113

negative_direction_move
DataRecord::Axis, 104
NoInterrupts
Interrupt, 240
NoMotion
DataRecord::Axis, 103

open
Controller, 107
output
DataRecord, 115
output_analog_0
GDataRecord47000_ENC, 198
GDataRecord47162, 203
GDataRecord47300_24EX, 209
GDataRecord47300_ENC, 214
output_analog_1
GDataRecord30000, 172
GDataRecord47000_ENC, 199
GDataRecord47162, 204
GDataRecord47300_24EX, 209
GDataRecord47300_ENC, 214
output_analog_2
GDataRecord30000, 172
GDataRecord47000_ENC, 199
GDataRecord47162, 204
GDataRecord47300_24EX, 209
GDataRecord47300_ENC, 214
output_analog_3
GDataRecord47000_ENC, 199
GDataRecord47162, 204
GDataRecord47300_24EX, 209
GDataRecord47300_ENC, 214
output_analog_4
GDataRecord47000_ENC, 199
GDataRecord47162, 204
GDataRecord47300_24EX, 209
GDataRecord47300_ENC, 214
output_analog_5
GDataRecord47000_ENC, 199
GDataRecord47162, 204
GDataRecord47300_24EX, 209
GDataRecord47300_ENC, 214
output_analog_6
GDataRecord47000_ENC, 199
GDataRecord47162, 204
GDataRecord47300_24EX, 209
GDataRecord47300_ENC, 214
output_analog_7
GDataRecord47000_ENC, 199
GDataRecord47162, 204
GDataRecord47300_24EX, 210
GDataRecord47300_ENC, 215
output_back_3
GDataRecord47300_24EX, 211
output_bank
DataRecord, 116
output_bank_0
GDataRecord1802, 125
GDataRecord1806, 139
GDataRecord2103, 159
GDataRecord30000, 171
GDataRecord4000, 182
GDataRecord47000_ENC, 200
GDataRecord47300_24EX, 210
GDataRecord47300_ENC, 215
GDataRecord52000, 225
output_bank_1
GDataRecord1802, 125
GDataRecord1806, 139
GDataRecord2103, 159
GDataRecord30000, 171
GDataRecord4000, 182
GDataRecord47300_24EX, 211
GDataRecord47300_ENC, 216
GDataRecord52000, 225
output_bank_2
GDataRecord1802, 125
GDataRecord1806, 139
GDataRecord2103, 159
GDataRecord4000, 182
GDataRecord47300_24EX, 211
GDataRecord52000, 225
output_bank_3
GDataRecord1802, 125
GDataRecord1806, 139
GDataRecord2103, 159
GDataRecord4000, 182
GDataRecord52000, 225
output_bank_4
GDataRecord1802, 125
GDataRecord1806, 139
GDataRecord2103, 159
GDataRecord4000, 182
GDataRecord52000, 225
output_bank_5
GDataRecord1802, 125
GDataRecord1806, 139
GDataRecord2103, 160
GDataRecord4000, 182
GDataRecord52000, 225
output_bank_6
GDataRecord1802, 125
GDataRecord1806, 139
GDataRecord2103, 160
GDataRecord4000, 182

GDataRecord52000, 225
output_bank_7
GDataRecord1802, 125
GDataRecord1806, 139
GDataRecord2103, 160
GDataRecord4000, 183
GDataRecord52000, 225
output_bank_8
GDataRecord1802, 126
GDataRecord1806, 139
GDataRecord2103, 160
GDataRecord4000, 183
GDataRecord52000, 225
output_bank_9
GDataRecord1802, 126
GDataRecord1806, 139
GDataRecord2103, 160
GDataRecord4000, 183
GDataRecord52000, 226
output_byte_0
GDataRecord47162, 205
output_byte_1
GDataRecord47162, 205
output_byte_2
GDataRecord47162, 205
over_current
DataRecord::Amp, 101
over_temp
DataRecord::Amp, 101
over_voltage
DataRecord::Amp, 101

Pci
gclib, 98
peak_current
DataRecord::Axis, 105
POLLINGINTERVAL
gclibo.h, 285
position
DataRecord::Axis, 105
position_error
DataRecord::Axis, 105
PositionAbsolute
DataRecord::Axis, 103
PositionRelative
DataRecord::Axis, 103
preprocessor.md, 243
program
Controller, 108
Program Preprocessor, 21
ProgramStopped
Interrupt, 240
published
gclib, 99
pulse_count_0
GDataRecord47000_ENC, 200
GDataRecord47162, 206
GDataRecord47300_24EX, 211
GDataRecord47300_ENC, 216
reference_position
DataRecord::Axis, 105
reserved_0
GDataRecord1806, 140
GDataRecord4000, 183
GDataRecord52000, 226
reserved_10
GDataRecord1806, 140
GDataRecord4000, 183
GDataRecord52000, 226
reserved_12
GDataRecord1806, 140
GDataRecord4000, 183
GDataRecord52000, 226
reserved_14
GDataRecord1806, 140
GDataRecord4000, 184
GDataRecord52000, 226
reserved_16
GDataRecord1806, 140
reserved_17
GDataRecord1806, 140
reserved_18
GDataRecord1806, 141
reserved_19
GDataRecord1806, 141
reserved_2
GDataRecord1806, 140
GDataRecord4000, 183
GDataRecord52000, 226
reserved_20
GDataRecord1806, 141
reserved_21
GDataRecord1806, 141
reserved_22
GDataRecord1806, 141
reserved_23
GDataRecord1806, 141
reserved_24
GDataRecord1806, 141
reserved_4
GDataRecord1806, 140
GDataRecord4000, 183
GDataRecord52000, 226
reserved_6
GDataRecord1806, 140
GDataRecord4000, 183
GDataRecord52000, 226
reserved_8
GDataRecord1806, 140
GDataRecord4000, 183
GDataRecord52000, 226
reverse_limit
DataRecord::Axis, 104
revision_information
Controller, 107
rio47000
GDataRecord, 120

rio47162
GDataRecord, 120
rio47300
GDataRecord, 120
rio47300_24ex
GDataRecord, 120

s_distance
GDataRecord1802, 126
GDataRecord1806, 142
GDataRecord2103, 160
GDataRecord30000, 172
GDataRecord4000, 185
GDataRecord52000, 228
s_plane_buffer_available
GDataRecord1806, 142
GDataRecord30000, 172
GDataRecord4000, 185
GDataRecord52000, 228
s_plane_move_status
GDataRecord1802, 126
GDataRecord1806, 142
GDataRecord2103, 160
GDataRecord30000, 172
GDataRecord4000, 185
GDataRecord52000, 228
s_plane_segment_count
GDataRecord1802, 126
GDataRecord1806, 142
GDataRecord2103, 160
GDataRecord30000, 172
GDataRecord4000, 185
GDataRecord52000, 228
sample
DataRecord, 115
sample_number
GDataRecord1802, 124
GDataRecord1806, 138
GDataRecord2103, 158
GDataRecord30000, 171
GDataRecord4000, 181
GDataRecord47000_ENC, 198
GDataRecord47162, 203
GDataRecord47300_24EX, 209
GDataRecord47300_ENC, 214
GDataRecord52000, 224
segment_count
DataRecord::CoordinatedMove, 113
Serial
gclib, 98
serial_number
Controller, 107
server
gclib, 99
set_array
Controller, 108
set_data_records
Controller, 109
set_firmware
Controller, 108
set_interrupts
Controller, 108
set_program
Controller, 108
set_published
gclib, 99
set_server
gclib, 99
SL
gclib_record.h, 271
slewing
DataRecord::Axis, 104
DataRecord::CoordinatedMove, 113
status
DataRecord::CoordinatedMove, 113
gclib_interrupt_t, 118
Interrupt, 241
stepper_mode
DataRecord::Axis, 104
stop_code
DataRecord::Axis, 104
stopping
DataRecord::Axis, 103
DataRecord::CoordinatedMove, 113
subscribe_data_records
Controller, 110, 111
subscribe_interrupts
Controller, 110
subscribe_messages
Controller, 109, 110
subscribe_progress
Controller, 111
SW
gclib_record.h, 271

t_distance
GDataRecord1802, 126
GDataRecord1806, 142
GDataRecord2103, 161
GDataRecord4000, 186
GDataRecord52000, 228
t_plane_buffer_available
GDataRecord1806, 142
GDataRecord4000, 186
GDataRecord52000, 229
t_plane_move_status
GDataRecord1802, 126
GDataRecord1806, 142
GDataRecord2103, 161
GDataRecord4000, 185
GDataRecord52000, 228
t_plane_segment_count
GDataRecord1802, 126
GDataRecord1806, 142
GDataRecord2103, 161
GDataRecord4000, 185
GDataRecord52000, 228
TCP_MASTER

Data Record, 56
TCP_SLAVE
Data Record, 56
TcpMaster
DataRecord, 115
TcpSlave
DataRecord, 114
thread_running
DataRecord, 116
thread_status
GDataRecord1806, 141
GDataRecord30000, 171
GDataRecord4000, 185
GDataRecord52000, 227
torque
DataRecord::Axis, 105
Type
Interrupt, 240
type
gclib_interrupt_t, 118
Interrupt, 241

UB
gclib_record.h, 271
UDP_MASTER
Data Record, 56
UDP_SLAVE
Data Record, 56
UdpMaster
DataRecord, 115
UdpSlave
DataRecord, 114
UL
gclib_record.h, 271
under_voltage
DataRecord::Amp, 102
Unsolicited Data, 67, 90
gclib_data_record, 73
gclib_interrupt, 73
gclib_message, 72
gclib_set_data_records, 70
gclib_set_interrupts, 69
gclib_subscribe_data_records, 71
gclib_subscribe_interrupts, 70
gclib_subscribe_messages, 70
gclib_subscribe_progress, 71
gclib_unsubscribe, 72
GInterrupt, 92
GMessage, 91
GRecord, 90
GRecordRate, 92
unsubscribe_data_records
Controller, 111
unsubscribe_interrupts
Controller, 111
unsubscribe_messages
Controller, 111
unsubscribe_progress
Controller, 111
user_interrupt
gclib_interrupt_t, 118
Interrupt, 241
UserInterrupt
Interrupt, 240
UW
gclib_record.h, 271

variable
DataRecord::Axis, 105
VectorMove
DataRecord::Axis, 103
velocity
DataRecord::Axis, 105
version
gclib, 98

WatchdogTimer
Interrupt, 240

zc_variable
GDataRecord47000_ENC, 200
GDataRecord47162, 206
GDataRecord47300_24EX, 211
GDataRecord47300_ENC, 216
zd_variable
GDataRecord47000_ENC, 200
GDataRecord47162, 206
GDataRecord47300_24EX, 211
GDataRecord47300_ENC, 216