EtherSpaceLinks JAVA API

Generated by Doxygen 1.8.5

Wed Aug 5 2020 16:21:10

Contents

1 Hardware 1

2 Bug List 3

3 Module Index 5

3.1 Modules . . . 5

4 Hierarchical Index 7

4.1 Class Hierarchy . . . 7

5 Class Index 9

5.1 Class List . . . 9

6 File Index 11

6.1 File List . . . 11

7 Module Documentation 13

7.1 Connection . . . 13

7.1.1 Detailed Description . . . 13

7.1.2 Function Documentation . . . 13

7.1.2.1 device_type . . . 13

7.1.2.2 open . . . 13

7.2 Physical Link Attributes . . . 15

7.2.1 Detailed Description . . . 15

7.2.2 Function Documentation . . . 15

7.2.2.1 link_connected . . . 15

7.2.2.2 set_active_link . . . 15

7.2.2.3 set_mode . . . 16

7.2.2.4 set_mode_portmask . . . 16

7.2.2.5 set_speed . . . 17

7.3 Virtual Link Attributes . . . 18

7.3.1 Detailed Description . . . 18

7.4 Handling Spacewire Traffic . . . 19

7.4.1 Detailed Description . . . 19

7.5 Event handling on spacewire links . . . 20
7.5.1 Detailed Description . . . 20
7.6 Sending data on a spaceiwre link . . . 21
7.6.1 Detailed Description . . . 21
7.6.2 Function Documentation . . . 21
7.6.2.1 flush . . . 21
7.6.2.2 write_packet . . . 21
7.7 Reading data from a spacewire link . . . 23
7.7.1 Detailed Description . . . 23
7.7.2 Function Documentation . . . 23
7.7.2.1 get_packet . . . 23
7.7.2.2 get_rx_flags . . . 24
7.8 TimeTag . . . 25
7.8.1 Detailed Description . . . 25
7.8.2 Function Documentation . . . 25
7.8.2.1 TT_enable_reporting . . . 25
7.9 Error Reporting . . . 26
7.9.1 Detailed Description . . . 26
7.9.2 Function Documentation . . . 26
7.9.2.1 ER_enable_reporting . . . 26
7.10 Error Waveforms . . . 27
7.10.1 Detailed Description . . . 27
7.10.2 Function Documentation . . . 27
7.10.2.1 EW_enable_reporting . . . 27
7.10.2.2 EW_source . . . 27
7.11 Extension codes . . . 29
7.11.1 Detailed Description . . . 30
7.11.2 Variable Documentation . . . 30
7.11.2.1 ATOM . . . 30
7.11.2.2 BARRIER . . . 30
7.11.2.3 Delay . . . 30
7.11.2.4 EEP . . . 30
7.11.2.5 EOP . . . 30
7.11.2.6 ESC . . . 30
7.11.2.7 ESC_EEP . . . 30
7.11.2.8 ESC_EOP . . . 30
7.11.2.9 ESC_ESC . . . 30
7.11.2.10 ESC_FCT . . . 30
7.11.2.11 EVENT . . . 31
7.11.2.12 FCT . . . 31

7.11.2.13 FORWARD . . . 31
7.11.2.14 Header . . . 31
7.11.2.15 HOLD . . . 31
7.11.2.16 JOIN . . . 31
7.11.2.17 Missing_data . . . 31
7.11.2.18 Module . . . 31
7.11.2.19 MOTA . . . 31
7.11.2.20 Multi_byte_extn_start . . . 31
7.11.2.21 ParityError . . . 31
7.11.2.22 PERROR1 . . . 31
7.11.2.23 PERROR2 . . . 31
7.11.2.24 PortSelect . . . 31
7.11.2.25 PortSelect_max . . . 31
7.11.2.26 REPEAT_1 . . . 31
7.11.2.27 REPEAT_2 . . . 31
7.11.2.28 REPEAT_3 . . . 31
7.11.2.29 RESIGN . . . 32
7.11.2.30 STORE . . . 32
7.11.2.31 TimeCode . . . 32
7.11.2.32 Timeout . . . 32
7.11.2.33 TimeTag . . . 32
7.11.2.34 TimeTag_delta . . . 32
7.11.2.35 TimeTag_uncertainty . . . 32
7.11.2.36 TimeZero . . . 32
7.11.2.37 TRUNCATE_1 . . . 32
7.11.2.38 TRUNCATE_2 . . . 32
7.11.2.39 Year . . . 32
7.12 TimeTag mask fields . . . 33
7.12.1 Detailed Description . . . 33
7.12.2 Variable Documentation . . . 33
7.12.2.1 TT . . . 33
7.12.2.2 TT_64 . . . 33
7.12.2.3 TT_report_EEP . . . 33
7.12.2.4 TT_report_EOP . . . 33
7.12.2.5 TT_report_EOP_EEP . . . 34
7.12.2.6 TT_report_ESC_EEP . . . 34
7.12.2.7 TT_report_ESC_EOP . . . 34
7.12.2.8 TT_report_ESC_ESC . . . 34
7.12.2.9 TT_report_fct . . . 34
7.12.2.10 TT_report_first_byte . . . 34

7.12.2.11 TT_report_intermediate_bytes . . . 34
7.12.2.12 TT_report_nothing . . . 34
7.12.2.13 TT_report_null . . . 34
7.12.2.14 TT_report_parity_error . . . 34
7.12.2.15 TT_report_time_code . . . 34
7.12.2.16 TT_report_timeout . . . 34
7.13 Error mask fields . . . 35
7.13.1 Detailed Description . . . 35
7.13.2 Variable Documentation . . . 35
7.13.2.1 ER . . . 35
7.13.2.2 ER_64 . . . 35
7.13.2.3 ER_report_ESC_EEP . . . 35
7.13.2.4 ER_report_ESC_EOP . . . 35
7.13.2.5 ER_report_ESC_ESC . . . 36
7.13.2.6 ER_report_fct . . . 36
7.13.2.7 ER_report_first_fct . . . 36
7.13.2.8 ER_report_first_null . . . 36
7.13.2.9 ER_report_nchar . . . 36
7.13.2.10 ER_report_nothing . . . 36
7.13.2.11 ER_report_null . . . 36
7.13.2.12 ER_report_parity_error . . . 36
7.13.2.13 ER_report_running_error . . . 36
7.13.2.14 ER_report_starting_error . . . 36
7.13.2.15 ER_report_time_code . . . 36
7.13.2.16 ER_report_timeout . . . 36
7.14 Error Waveform Triggers . . . 37
7.14.1 Detailed Description . . . 37
7.14.2 Variable Documentation . . . 37
7.14.2.1 EW . . . 37
7.14.2.2 EW_capture_EEP . . . 37
7.14.2.3 EW_capture_EOP . . . 38
7.14.2.4 EW_capture_ESC_EEP . . . 38
7.14.2.5 EW_capture_ESC_EOP . . . 38
7.14.2.6 EW_capture_ESC_ESC . . . 38
7.14.2.7 EW_capture_excess_data . . . 38
7.14.2.8 EW_capture_excess_FCT . . . 38
7.14.2.9 EW_capture_FCT . . . 38
7.14.2.10 EW_capture_first_fct . . . 38
7.14.2.11 EW_capture_first_null . . . 38
7.14.2.12 EW_capture_nchar . . . 38

7.14.2.13 EW_capture_nothing . . . 38
7.14.2.14 EW_capture_null . . . 38
7.14.2.15 EW_capture_parity_error . . . 38
7.14.2.16 EW_capture_running_error . . . 39
7.14.2.17 EW_capture_starting_error . . . 39
7.14.2.18 EW_capture_time_code . . . 39
7.14.2.19 EW_capture_timeout . . . 39
7.14.2.20 EW_RT . . . 39
7.15 Error Waveform Sources . . . 40
7.15.1 Detailed Description . . . 40
7.15.2 Variable Documentation . . . 40
7.15.2.1 EW_Source_barrier . . . 40
7.15.2.2 EW_Source_local_clock . . . 40
7.15.2.3 EW_Source_port_1 . . . 40
7.15.2.4 EW_Source_port_2 . . . 41
7.15.2.5 EW_Source_port_3 . . . 41
7.15.2.6 EW_Source_port_4 . . . 41
7.15.2.7 EW_Source_port_5 . . . 41
7.15.2.8 EW_Source_port_6 . . . 41
7.15.2.9 EW_Source_port_7 . . . 41
7.15.2.10 EW_Source_port_8 . . . 41
7.15.2.11 EW_Source_SMA_12 . . . 41
7.15.2.12 EW_Source_SMA_34 . . . 41
7.15.2.13 EW_Source_SMA_56 . . . 41
7.15.2.14 EW_Source_SMA_78 . . . 41
7.16 Memory Mapped Addresses . . . 42
7.16.1 Detailed Description . . . 42
7.16.2 Variable Documentation . . . 42
7.16.2.1 ATI_address . . . 42
7.16.2.2 DESCRIPTION_address . . . 42
7.16.2.3 Event_cause_address . . . 42
7.16.2.4 EW_address . . . 42
7.16.2.5 EW_source_address . . . 42
7.16.2.6 FLOW_CONTROL_address . . . 42
7.16.2.7 HWA_address . . . 42
7.16.2.8 IGNORE_address . . . 42
7.16.2.9 LINK_address . . . 42
7.16.2.10 max_packet_data . . . 42
7.16.2.11 NLINKS_address . . . 42
7.16.2.12 OBSERVE_address . . . 43

7.16.2.13 OPTIONS_address . . . 43
7.16.2.14 PC_address . . . 43
7.16.2.15 PG_address . . . 43
7.16.2.16 RX_SPEED_address . . . 43
7.16.2.17 SMA_56_pulse_width_address . . . 43
7.16.2.18 TIMETAG_address . . . 43
7.16.2.19 TX_SPEED_address . . . 43
7.16.2.20 VERSION_address . . . 43
7.17 Error Codes . . . 44
7.17.1 Detailed Description . . . 45
7.17.2 Variable Documentation . . . 45
7.17.2.1 Error_Buffer_Full . . . 45
7.17.2.2 Error_Callback_Return . . . 46
7.17.2.3 Error_CaptureThread_Failed . . . 46
7.17.2.4 Error_EINTR . . . 46
7.17.2.5 Error_EXE_Start_Failed . . . 46
7.17.2.6 Error_File_Move . . . 46
7.17.2.7 Error_File_Not_Present . . . 46
7.17.2.8 Error_FileList_Empty . . . 46
7.17.2.9 Error_Function_Not_Supported . . . 46
7.17.2.10 Error_Host_Unresolvable . . . 46
7.17.2.11 Error_Host_Unresponsive . . . 46
7.17.2.12 Error_Incorrect_Device . . . 46
7.17.2.13 Error_Invalid_Device . . . 46
7.17.2.14 Error_Invalid_File . . . 47
7.17.2.15 Error_Invalid_Link . . . 47
7.17.2.16 Error_IO_Error . . . 47
7.17.2.17 Error_Link_Incorrect . . . 47
7.17.2.18 Error_Link_Not_Connected . . . 47
7.17.2.19 Error_LogFile_Open . . . 47
7.17.2.20 Error_LogFile_Write . . . 47
7.17.2.21 Error_Memory . . . 47
7.17.2.22 Error_Module_Not_Present . . . 47
7.17.2.23 Error_Network . . . 47
7.17.2.24 Error_Network_Format_Error . . . 47
7.17.2.25 Error_NO_Connection . . . 47
7.17.2.26 Error_Not_Known . . . 48
7.17.2.27 Error_Parameter_RangeIncorrect . . . 48
7.17.2.28 Error_ReadHandler_Running . . . 48
7.17.2.29 Error_Receiver_Shutdown . . . 48

7.17.2.30 Error_Receiver_Timeout . . . 48
7.17.2.31 Error_RecFile_Open . . . 48
7.17.2.32 Error_RecFile_Write . . . 48
7.17.2.33 Error_Request_Too_Large . . . 48
7.17.2.34 Error_Response_Mismatch . . . 48
7.17.2.35 Error_Response_Too_Small . . . 48
7.17.2.36 Error_SaveBuf_Overflow_Restore . . . 48
7.17.2.37 Error_SaveBuf_Overflow_Save . . . 48
7.17.2.38 Error_Sequence_Error . . . 49
7.17.2.39 Error_Set_Option_File . . . 49
7.17.2.40 Error_Unknown_System_Type . . . 49
7.17.2.41 Error_WaveForm_Dir_Create . . . 49
7.17.2.42 Error_Would_Block . . . 49
7.17.2.43 Error_Zero_Read . . . 49
7.17.2.44 Option_SO . . . 49
7.18 functions . . . 50
7.18.1 Detailed Description . . . 50
7.18.2 Function Documentation . . . 50
7.18.2.1 device_type . . . 50
7.18.2.2 err . . . 51
7.18.2.3 error_event . . . 52
7.18.2.4 esc_eep . . . 52
7.18.2.5 esc_eop . . . 52
7.18.2.6 esc_esc . . . 52
7.18.2.7 link_selected . . . 52
7.18.2.8 link_timeout . . . 52
7.18.2.9 link_tx_speed . . . 52
7.18.2.10 missing_data . . . 52
7.18.2.11 parity_error . . . 52
7.18.2.12 perror1 . . . 53
7.18.2.13 perror2 . . . 53
7.18.2.14 port_status . . . 53
7.18.2.15 raw_waveform_data . . . 53
7.18.2.16 received_esc_fct . . . 53
7.18.2.17 received_fct . . . 53
7.18.2.18 start_time . . . 53
7.18.2.19 timecode . . . 53
7.18.2.20 timetag . . . 54
7.18.2.21 timetag_uncertainty . . . 54
7.18.2.22 timezero . . . 54

7.18.2.23 unknown_extn_data . . . 54
7.18.2.24 unknown_ram_data . . . 54
7.18.2.25 unknown_special_data . . . 55
7.18.2.26 waveform_data . . . 55

8 Class Documentation **57**

8.1 EtherSpaceLink.Error Class Reference . . . 57
8.1.1 Constructor & Destructor Documentation . . . 57
8.1.1.1 Error . . . 57
8.1.2 Member Function Documentation . . . 57
8.1.2.1 Get . . . 57
8.2 EtherSpaceLink Class Reference . . . 57
8.2.1 Constructor & Destructor Documentation . . . 72
8.2.1.1 EtherSpaceLink . . . 72
8.2.1.2 EtherSpaceLink . . . 72
8.2.2 Member Function Documentation . . . 73
8.2.2.1 abort . . . 73
8.2.2.2 ATI_calibrate . . . 73
8.2.2.3 ATI_calibrate_401 . . . 73
8.2.2.4 ATI_calibrate_408 . . . 73
8.2.2.5 check_record_writes . . . 73
8.2.2.6 close . . . 73
8.2.2.7 connected . . . 74
8.2.2.8 decode_fp16 . . . 74
8.2.2.9 delay . . . 74
8.2.2.10 device_type . . . 74
8.2.2.11 device_type . . . 74
8.2.2.12 dump_max . . . 74
8.2.2.13 EI_flow_control . . . 74
8.2.2.14 EI_flow_control_401 . . . 75
8.2.2.15 EI_flow_control_408 . . . 75
8.2.2.16 EI_ignore_events . . . 75
8.2.2.17 EI_ignore_events_401 . . . 75
8.2.2.18 EI_ignore_events_408 . . . 76
8.2.2.19 empty_frame . . . 76
8.2.2.20 enable_callbacks . . . 76
8.2.2.21 ER_enable_reporting . . . 76
8.2.2.22 ER_enable_reporting_401 . . . 76
8.2.2.23 ER_enable_reporting_408 . . . 76
8.2.2.24 err . . . 76

8.2.2.25 error_event . . . 76
8.2.2.26 esc_eep . . . 76
8.2.2.27 esc_eop . . . 76
8.2.2.28 esc_esc . . . 77
8.2.2.29 esl_delay . . . 77
8.2.2.30 EW_clear . . . 77
8.2.2.31 EW_clear_401 . . . 77
8.2.2.32 EW_clear_408 . . . 77
8.2.2.33 EW_enable_reporting . . . 77
8.2.2.34 EW_enable_reporting_401 . . . 77
8.2.2.35 EW_enable_reporting_408 . . . 77
8.2.2.36 EW_request_data . . . 77
8.2.2.37 EW_request_data_401 . . . 79
8.2.2.38 EW_request_data_408 . . . 79
8.2.2.39 EW_reset . . . 79
8.2.2.40 EW_reset_401 . . . 79
8.2.2.41 EW_reset_408 . . . 79
8.2.2.42 EW_source . . . 79
8.2.2.43 EW_source_401 . . . 80
8.2.2.44 EW_source_408 . . . 80
8.2.2.45 extn_data . . . 80
8.2.2.46 extract_link . . . 80
8.2.2.47 extract_link_state . . . 80
8.2.2.48 extract_rx_speed . . . 80
8.2.2.49 extract_timetag . . . 80
8.2.2.50 extract_timetag_401 . . . 80
8.2.2.51 extract_timetag_408 . . . 80
8.2.2.52 extract_timetag_ns . . . 80
8.2.2.53 extract_timetag_ns_401 . . . 81
8.2.2.54 extract_timetag_ns_408 . . . 81
8.2.2.55 extract_tx_speed . . . 81
8.2.2.56 fastclose . . . 81
8.2.2.57 flow_control . . . 81
8.2.2.58 flush . . . 82
8.2.2.59 flush_record_file . . . 83
8.2.2.60 flush_S . . . 83
8.2.2.61 get_context . . . 83
8.2.2.62 get_control_packet . . . 83
8.2.2.63 get_HWA . . . 84
8.2.2.64 get_HWA_401 . . . 84

8.2.2.65 get_HWA_408 . . . 84
8.2.2.66 get_manufacturer_string . . . 84
8.2.2.67 get_manufacturer_string_401 . . . 84
8.2.2.68 get_manufacturer_string_408 . . . 84
8.2.2.69 get_module_slot . . . 84
8.2.2.70 get_module_string . . . 85
8.2.2.71 get_module_string_401 . . . 85
8.2.2.72 get_module_string_408 . . . 85
8.2.2.73 get_module_type . . . 85
8.2.2.74 get_number_of_links . . . 85
8.2.2.75 get_number_of_links_401 . . . 86
8.2.2.76 get_number_of_links_408 . . . 86
8.2.2.77 get_options . . . 86
8.2.2.78 get_options_string . . . 86
8.2.2.79 get_options_string_401 . . . 86
8.2.2.80 get_options_string_408 . . . 86
8.2.2.81 get_packet . . . 86
8.2.2.82 get_percent_file_read . . . 87
8.2.2.83 get_product_string . . . 87
8.2.2.84 get_product_string_401 . . . 88
8.2.2.85 get_product_string_408 . . . 88
8.2.2.86 get_receive_speed . . . 88
8.2.2.87 get_receive_speed_401 . . . 88
8.2.2.88 get_receive_speed_408 . . . 88
8.2.2.89 get_record_file . . . 88
8.2.2.90 get_record_size . . . 88
8.2.2.91 get_rx_flags . . . 88
8.2.2.92 get_rx_timeout . . . 89
8.2.2.93 get_slot . . . 89
8.2.2.94 get_terminator . . . 89
8.2.2.95 get_timetag_401 . . . 89
8.2.2.96 get_timetag_408 . . . 89
8.2.2.97 get_total_raw_bytes_received . . . 89
8.2.2.98 get_version . . . 89
8.2.2.99 handle_header . . . 90
8.2.2.100 handle_module . . . 90
8.2.2.101 handle_year . . . 90
8.2.2.102 HWA_to_serial_number_string . . . 90
8.2.2.103 link_connected . . . 90
8.2.2.104 link_connected_401 . . . 90

8.2.2.105 link_connected_408 . . . 90
8.2.2.106 link_selected . . . 90
8.2.2.107 link_timeout . . . 90
8.2.2.108 link_tx_speed . . . 91
8.2.2.109 log_packet . . . 91
8.2.2.110 missing_data . . . 91
8.2.2.111 negotiate . . . 91
8.2.2.112 Observe . . . 91
8.2.2.113 Observe_401 . . . 91
8.2.2.114 Observe_408 . . . 91
8.2.2.115 open . . . 91
8.2.2.116 parity_error . . . 92
8.2.2.117 perror1 . . . 92
8.2.2.118 perror2 . . . 92
8.2.2.119 port_status . . . 92
8.2.2.120 ram_io_get_int . . . 92
8.2.2.121 ram_io_write_bytes . . . 92
8.2.2.122 raw_link_address . . . 92
8.2.2.123 raw_ram_rw . . . 92
8.2.2.124 raw_speed_address . . . 92
8.2.2.125 raw_waveform_data . . . 92
8.2.2.126 received_esc_fct . . . 93
8.2.2.127 received_fct . . . 93
8.2.2.128 receivePacket . . . 93
8.2.2.129 record_writes . . . 93
8.2.2.130 request_link_status . . . 93
8.2.2.131 request_link_status_port . . . 93
8.2.2.132 request_rx_speed . . . 94
8.2.2.133 request_tx_speed . . . 94
8.2.2.134 send . . . 94
8.2.2.135 send_ESC . . . 95
8.2.2.136 send_ESC_EEP . . . 95
8.2.2.137 send_ESC_EOP . . . 95
8.2.2.138 send_ESC_ESC . . . 95
8.2.2.139 send_ESC_FCT . . . 95
8.2.2.140 send_timecode . . . 95
8.2.2.141 sendPacket . . . 95
8.2.2.142 sendPacket . . . 95
8.2.2.143 sendPacket . . . 96
8.2.2.144 sendPacket_S . . . 96

8.2.2.145 sendPacket_S . . . 96
8.2.2.146 sendPacket_S . . . 96
8.2.2.147 set_active_link . . . 96
8.2.2.148 set_active_link_401 . . . 97
8.2.2.149 set_active_link_408 . . . 97
8.2.2.150 set_context . . . 97
8.2.2.151 set_EINTR . . . 97
8.2.2.152 set_extension_cb . . . 97
8.2.2.153 set_log_file . . . 97
8.2.2.154 set_max_packet_data . . . 98
8.2.2.155 set_max_packet_data_401 . . . 98
8.2.2.156 set_max_packet_data_408 . . . 98
8.2.2.157 set_mode . . . 98
8.2.2.158 set_mode_401 . . . 99
8.2.2.159 set_mode_408 . . . 99
8.2.2.160 set_mode_portmask . . . 99
8.2.2.161 set_record_file . . . 100
8.2.2.162 set_rx_timeout . . . 100
8.2.2.163 set_rx_timeout_action . . . 100
8.2.2.164 set_slot . . . 101
8.2.2.165 set_special_cb . . . 101
8.2.2.166 set_speed . . . 101
8.2.2.167 set_speed_401 . . . 101
8.2.2.168 set_speed_408 . . . 101
8.2.2.169 set_speed_double . . . 101
8.2.2.170 set_speed_double_401 . . . 102
8.2.2.171 set_speed_double_408 . . . 102
8.2.2.172 set_tx_record_file . . . 102
8.2.2.173 shutdown . . . 102
8.2.2.174 sma_56_pulse_width . . . 102
8.2.2.175 sma_56_pulse_width_401 . . . 103
8.2.2.176 sma_56_pulse_width_408 . . . 103
8.2.2.177 SO_BARRIER . . . 103
8.2.2.178 SO_FORWARD . . . 103
8.2.2.179 SO_JOIN . . . 103
8.2.2.180 SO_STORE . . . 103
8.2.2.181 source . . . 103
8.2.2.182 special_data . . . 103
8.2.2.183 start_time . . . 103
8.2.2.184 system_type . . . 103

8.2.2.185 timecode . . . 104
8.2.2.186 timetag . . . 105
8.2.2.187 timetag_uncertainty . . . 105
8.2.2.188 timezero . . . 105
8.2.2.189 TT_enable_reporting . . . 105
8.2.2.190 TT_enable_reporting_401 . . . 105
8.2.2.191 TT_enable_reporting_408 . . . 105
8.2.2.192 unknown_extn_data . . . 105
8.2.2.193 unknown_ram_data . . . 106
8.2.2.194 unknown_special_data . . . 106
8.2.2.195 update . . . 106
8.2.2.196 update_es . . . 106
8.2.2.197 waveform_data . . . 106
8.2.2.198 what . . . 106
8.2.2.199 write_buffer_empty . . . 106
8.2.2.200 write_buffer_empty_401 . . . 107
8.2.2.201 write_buffer_empty_408 . . . 107
8.2.2.202 write_EXTN . . . 107
8.2.2.203 write_packet . . . 108
8.2.3 Member Data Documentation . . . 108
8.2.3.1 ATI_address . . . 108
8.2.3.2 ATOM . . . 109
8.2.3.3 BARRIER . . . 109
8.2.3.4 barrier . . . 109
8.2.3.5 CALLBACK_EXTENSION_DATA . . . 109
8.2.3.6 CALLBACK_SPECIAL_DATA . . . 109
8.2.3.7 CAPABILITIES . . . 109
8.2.3.8 CONNECT_FILE . . . 109
8.2.3.9 CR . . . 109
8.2.3.10 Delay . . . 109
8.2.3.11 DESCRIPTION_address . . . 109
8.2.3.12 DISCARD_EXTENSION_DATA . . . 109
8.2.3.13 DISCARD_SPECIAL_DATA . . . 109
8.2.3.14 EEP . . . 109
8.2.3.15 EI_ignore_ESC_EEP . . . 109
8.2.3.16 EI_ignore_ESC_EOP . . . 109
8.2.3.17 EI_ignore_ESC_ESC . . . 109
8.2.3.18 EI_ignore_excess_data . . . 109
8.2.3.19 EI_ignore_excess_FCT . . . 109
8.2.3.20 EI_ignore_parity_error . . . 109

8.2.3.21 EI_ignore_timeout . . . 109
8.2.3.22 EI_no_automatic_FCT . . . 109
8.2.3.23 EI_normal_flow_control . . . 109
8.2.3.24 EI_transmit_anyway . . . 109
8.2.3.25 EOP . . . 109
8.2.3.26 ER . . . 109
8.2.3.27 ER_64 . . . 109
8.2.3.28 ER_report_ESC_EEP . . . 110
8.2.3.29 ER_report_ESC_EOP . . . 110
8.2.3.30 ER_report_ESC_ESC . . . 110
8.2.3.31 ER_report_fct . . . 110
8.2.3.32 ER_report_first_fct . . . 110
8.2.3.33 ER_report_first_null . . . 110
8.2.3.34 ER_report_nchar . . . 110
8.2.3.35 ER_report_nothing . . . 110
8.2.3.36 ER_report_null . . . 110
8.2.3.37 ER_report_parity_error . . . 110
8.2.3.38 ER_report_running_error . . . 110
8.2.3.39 ER_report_starting_error . . . 110
8.2.3.40 ER_report_time_code . . . 110
8.2.3.41 ER_report_timeout . . . 111
8.2.3.42 Error_Buffer_Full . . . 111
8.2.3.43 Error_Callback_Return . . . 111
8.2.3.44 Error_CaptureThread_Failed . . . 111
8.2.3.45 Error_EINTR . . . 111
8.2.3.46 Error_EXE_Start_Failed . . . 111
8.2.3.47 Error_File_Move . . . 111
8.2.3.48 Error_File_Not_Present . . . 111
8.2.3.49 Error_FileList_Empty . . . 111
8.2.3.50 Error_Function_Not_Supported . . . 111
8.2.3.51 Error_Host_Unresolvable . . . 111
8.2.3.52 Error_Host_Unresponsive . . . 111
8.2.3.53 Error_Incorrect_Device . . . 112
8.2.3.54 Error_Invalid_Device . . . 112
8.2.3.55 Error_Invalid_File . . . 112
8.2.3.56 Error_Invalid_Link . . . 112
8.2.3.57 Error_IO_Error . . . 112
8.2.3.58 Error_Link_Incorrect . . . 112
8.2.3.59 Error_Link_Not_Connected . . . 112
8.2.3.60 Error_LogFile_Open . . . 112

8.2.3.61 Error_LogFile_Write . . . 112
8.2.3.62 Error_Memory . . . 112
8.2.3.63 Error_Module_Not_Present . . . 112
8.2.3.64 Error_Network . . . 112
8.2.3.65 Error_Network_Format_Error . . . 113
8.2.3.66 Error_NO_Connection . . . 113
8.2.3.67 Error_Not_Known . . . 113
8.2.3.68 Error_Parameter_RangeIncorrect . . . 113
8.2.3.69 Error_ReadHandler_Running . . . 113
8.2.3.70 Error_Receiver_Shutdown . . . 113
8.2.3.71 Error_Receiver_Timeout . . . 113
8.2.3.72 Error_RecFile_Open . . . 113
8.2.3.73 Error_RecFile_Write . . . 113
8.2.3.74 Error_Request_Too_Large . . . 113
8.2.3.75 Error_Response_Mismatch . . . 113
8.2.3.76 Error_Response_Too_Small . . . 113
8.2.3.77 Error_SaveBuf_Overflow_Restore . . . 114
8.2.3.78 Error_SaveBuf_Overflow_Save . . . 114
8.2.3.79 Error_Sequence_Error . . . 114
8.2.3.80 Error_Set_Option_File . . . 114
8.2.3.81 Error_Unknown_System_Type . . . 114
8.2.3.82 Error_WaveForm_Dir_Create . . . 114
8.2.3.83 Error_Would_Block . . . 114
8.2.3.84 Error_Zero_Read . . . 114
8.2.3.85 ESC . . . 114
8.2.3.86 ESC_EEP . . . 114
8.2.3.87 ESC_EOP . . . 114
8.2.3.88 ESC_ESC . . . 114
8.2.3.89 ESC_FCT . . . 115
8.2.3.90 EVENT . . . 115
8.2.3.91 Event_cause_address . . . 115
8.2.3.92 EW . . . 115
8.2.3.93 EW_address . . . 115
8.2.3.94 EW_capture_EEP . . . 115
8.2.3.95 EW_capture_EOP . . . 115
8.2.3.96 EW_capture_ESC_EEP . . . 115
8.2.3.97 EW_capture_ESC_EOP . . . 115
8.2.3.98 EW_capture_ESC_ESC . . . 115
8.2.3.99 EW_capture_excess_data . . . 115
8.2.3.100 EW_capture_excess_FCT . . . 115

8.2.3.101 EW_capture_FCT . . . 115
8.2.3.102 EW_capture_first_fct . . . 115
8.2.3.103 EW_capture_first_null . . . 115
8.2.3.104 EW_capture_nchar . . . 116
8.2.3.105 EW_capture_nothing . . . 116
8.2.3.106 EW_capture_null . . . 116
8.2.3.107 EW_capture_parity_error . . . 116
8.2.3.108 EW_capture_running_error . . . 116
8.2.3.109 EW_capture_starting_error . . . 116
8.2.3.110 EW_capture_time_code . . . 116
8.2.3.111 EW_capture_timeout . . . 116
8.2.3.112 EW_RT . . . 116
8.2.3.113 EW_source_address . . . 116
8.2.3.114 EW_Source_barrier . . . 116
8.2.3.115 EW_Source_local_clock . . . 116
8.2.3.116 EW_Source_port_1 . . . 116
8.2.3.117 EW_Source_port_2 . . . 116
8.2.3.118 EW_Source_port_3 . . . 117
8.2.3.119 EW_Source_port_4 . . . 117
8.2.3.120 EW_Source_port_5 . . . 117
8.2.3.121 EW_Source_port_6 . . . 117
8.2.3.122 EW_Source_port_7 . . . 117
8.2.3.123 EW_Source_port_8 . . . 117
8.2.3.124 EW_Source_SMA_12 . . . 117
8.2.3.125 EW_Source_SMA_34 . . . 117
8.2.3.126 EW_Source_SMA_56 . . . 117
8.2.3.127 EW_Source_SMA_78 . . . 117
8.2.3.128 EXTENSION_DATA_FLAGS . . . 117
8.2.3.129 EXTENSION_SIZE . . . 117
8.2.3.130 EXTN . . . 117
8.2.3.131 FCT . . . 118
8.2.3.132 FLOW_CONTROL_address . . . 118
8.2.3.133 FLUSH . . . 118
8.2.3.134 FORWARD . . . 118
8.2.3.135 Header . . . 118
8.2.3.136 HOLD . . . 118
8.2.3.137 HWA . . . 118
8.2.3.138 HWA_address . . . 118
8.2.3.139 IGNORE_address . . . 118
8.2.3.140 INCOMPLETE . . . 118

8.2.3.141 JOIN . . . 118
8.2.3.142 LINK . . . 118
8.2.3.143 LINK_address . . . 118
8.2.3.144 LINK_mode_disabled . . . 118
8.2.3.145 LINK_mode_fixed_speed . . . 118
8.2.3.146 LINK_mode_legacy . . . 118
8.2.3.147 LINK_mode_long_timeout . . . 118
8.2.3.148 LINK_mode_master . . . 118
8.2.3.149 LINK_mode_normal . . . 118
8.2.3.150 LINK_mode_slow_speed . . . 119
8.2.3.151 LINK_SPEED . . . 119
8.2.3.152 LINK_state_offset . . . 119
8.2.3.153 LINK_tx_buffer_empty . . . 119
8.2.3.154 MANUFACTURER . . . 119
8.2.3.155 max_packet_data . . . 119
8.2.3.156 Missing_data . . . 119
8.2.3.157 Module . . . 119
8.2.3.158 MOTA . . . 119
8.2.3.159 MSR_state_Connected . . . 119
8.2.3.160 MSR_state_NC . . . 119
8.2.3.161 Multi_byte_extn_start . . . 119
8.2.3.162 NLINKS_address . . . 119
8.2.3.163 OBSERVE_address . . . 119
8.2.3.164 Option_SO . . . 119
8.2.3.165 OPTIONS_address . . . 119
8.2.3.166 ParityError . . . 119
8.2.3.167 PART_EOP_EEP . . . 119
8.2.3.168 PART_EXTN . . . 119
8.2.3.169 PART_SPECIAL . . . 119
8.2.3.170 PC_address . . . 120
8.2.3.171 PERROR1 . . . 120
8.2.3.172 PERROR2 . . . 120
8.2.3.173 PG_address . . . 120
8.2.3.174 PortSelect . . . 120
8.2.3.175 PortSelect_max . . . 120
8.2.3.176 PRODUCT . . . 120
8.2.3.177 ram_rw . . . 120
8.2.3.178 rawtime . . . 120
8.2.3.179 READ_IMMEDIATE . . . 120
8.2.3.180 Receiver_Timeout_Returns_Error . . . 120

8.2.3.181 Receiver_Timeout_Returns_Zero_Part_Pkt . . . 120
8.2.3.182 REPEAT_1 . . . 120
8.2.3.183 REPEAT_2 . . . 120
8.2.3.184 REPEAT_3 . . . 120
8.2.3.185 report_delta . . . 120
8.2.3.186 report_EEP . . . 120
8.2.3.187 report_EOP . . . 120
8.2.3.188 report_ESC_EEP . . . 120
8.2.3.189 report_ESC_EOP . . . 121
8.2.3.190 report_ESC_ESC . . . 121
8.2.3.191 report_excess_data . . . 121
8.2.3.192 report_excess_FCT . . . 121
8.2.3.193 REPORT_EXTENSION_DATA . . . 121
8.2.3.194 report_FCT . . . 121
8.2.3.195 report_first_byte . . . 121
8.2.3.196 report_first_null . . . 121
8.2.3.197 report_mid_bytes . . . 121
8.2.3.198 report_nchar . . . 121
8.2.3.199 report_NULL . . . 121
8.2.3.200 report_parity_error . . . 121
8.2.3.201 REPORT_SPECIAL_DATA . . . 122
8.2.3.202 report_time_code . . . 122
8.2.3.203 report_timeout . . . 122
8.2.3.204 RESIGN . . . 122
8.2.3.205 RETURN_EXTENSION_DATA . . . 122
8.2.3.206 RETURN_SPECIAL_DATA . . . 122
8.2.3.207 router_cs . . . 122
8.2.3.208 router_stats . . . 122
8.2.3.209 router_tables . . . 122
8.2.3.210 rx_link . . . 122
8.2.3.211 RX_SPEED_address . . . 122
8.2.3.212 SF . . . 122
8.2.3.213 SF_disabled . . . 122
8.2.3.214 SF_enabled . . . 122
8.2.3.215 SMA_56_pulse_width_address . . . 122
8.2.3.216 SpaceWire_state_Connecting . . . 122
8.2.3.217 SpaceWire_state_ErrorReset . . . 122
8.2.3.218 SpaceWire_state_ErrorWait . . . 122
8.2.3.219 SpaceWire_state_Ready . . . 122
8.2.3.220 SpaceWire_state_Run . . . 122

8.2.3.221 SpaceWire_state_Started . . . 122
8.2.3.222 SPECIAL . . . 122
8.2.3.223 SPECIAL_DATA_FLAGS . . . 122
8.2.3.224 SPECIAL_SIZE . . . 122
8.2.3.225 STORE . . . 123
8.2.3.226 SYSTEM_TYPE_401 . . . 123
8.2.3.227 SYSTEM_TYPE_408 . . . 123
8.2.3.228 SYSTEM_TYPE_INVALID . . . 123
8.2.3.229 TC_rx . . . 123
8.2.3.230 TC_rx_64 . . . 123
8.2.3.231 TC_rx_report_enabled . . . 123
8.2.3.232 TC_rx_silent . . . 123
8.2.3.233 TC_rx_time_stamp_enabled . . . 123
8.2.3.234 TC_tx . . . 123
8.2.3.235 TC_tx_external_trigger . . . 123
8.2.3.236 TC_tx_format_mask . . . 123
8.2.3.237 TC_tx_increment_6_bits . . . 123
8.2.3.238 TC_tx_increment_7_bits . . . 123
8.2.3.239 TC_tx_increment_8_bits . . . 123
8.2.3.240 TC_tx_no_increment . . . 123
8.2.3.241 TC_tx_no_trigger . . . 123
8.2.3.242 TC_tx_one_code . . . 123
8.2.3.243 TC_tx_regular_trigger . . . 123
8.2.3.244 TC_tx_report_transmission . . . 123
8.2.3.245 TC_tx_trigger_mask . . . 123
8.2.3.246 TC_tx_update_code . . . 123
8.2.3.247 TC_tx_update_interval . . . 123
8.2.3.248 TimeCode . . . 123
8.2.3.249 Timeout . . . 123
8.2.3.250 TimeTag . . . 124
8.2.3.251 TIMETAG_address . . . 124
8.2.3.252 TimeTag_delta . . . 124
8.2.3.253 TimeTag_uncertainty . . . 124
8.2.3.254 TimeZero . . . 124
8.2.3.255 TRUNCATE_1 . . . 124
8.2.3.256 TRUNCATE_2 . . . 124
8.2.3.257 TRUNCATED . . . 124
8.2.3.258 TT . . . 124
8.2.3.259 TT_64 . . . 124
8.2.3.260 TT_now . . . 124

8.2.3.261 TT_report_EEP . . . 124
8.2.3.262 TT_report_EOP . . . 124
8.2.3.263 TT_report_EOP_EEP . . . 124
8.2.3.264 TT_report_ESC_EEP . . . 124
8.2.3.265 TT_report_ESC_EOP . . . 124
8.2.3.266 TT_report_ESC_ESC . . . 125
8.2.3.267 TT_report_fct . . . 125
8.2.3.268 TT_report_first_byte . . . 125
8.2.3.269 TT_report_intermediate_bytes . . . 125
8.2.3.270 TT_report_nothing . . . 125
8.2.3.271 TT_report_null . . . 125
8.2.3.272 TT_report_parity_error . . . 125
8.2.3.273 TT_report_time_code . . . 125
8.2.3.274 TT_report_timeout . . . 125
8.2.3.275 TX_SPEED_address . . . 125
8.2.3.276 version . . . 125
8.2.3.277 VERSION_address . . . 125
8.2.3.278 Year . . . 125

9 File Documentation **127**

9.1 /autogen/EtherSpaceLink.java File Reference . . . 127

Index **128**

Chapter 1

Hardware

Member EtherSpaceLink.write_packet (byte[] netbuff, int length, int flags)

dsi

Chapter 2

Bug List

Member EtherSpaceLink.extract_timetag (byte[] netbuff)

no checking on buffer length

Member EtherSpaceLink.extract_timetag_ns (byte[] netbuff)

no checking on buffer length

Chapter 3

Module Index

3.1 Modules

Here is a list of all modules:

- Connection . . . 13
- Physical Link Attributes . . . 15
- Virtual Link Attributes . . . 18
  - TimeTag . . . 25
    - TimeTag mask fields . . . 33
  - Error Reporting . . . 26
    - Error mask fields . . . 35
  - Error Waveforms . . . 27
    - Error Waveform Triggers . . . 37
    - Error Waveform Sources . . . 40
- Handling Spacewire Traffic . . . 19
  - Event handling on spacewire links . . . 20
    - functions . . . 50
  - Sending data on a spaceiwre link . . . 21
  - Reading data from a spacewire link . . . 23
  - Extension codes . . . 29
- Memory Mapped Addresses . . . 42
- Error Codes . . . 44

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

EtherSpaceLink_IO
    EtherSpaceLink 57
Exception
    EtherSpaceLink.Error 57

Chapter 5

Class Index

5.1 Class List

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

EtherSpaceLink.Error 57
EtherSpaceLink 57

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

/autogen/EtherSpaceLink.java 127

Chapter 7

Module Documentation

7.1 Connection

This file contains the definitions of constants used to drive ESL functions.

Functions

- int EtherSpaceLink.device_type () throws Exception

  returns the device type
- int EtherSpaceLink.open (String ip_address) throws Exception

  opens a connection to the specified device

7.1.1 Detailed Description

This file contains the definitions of constants used to drive ESL functions. (c) 4Links Limited 2000-2019

These functions are used to make a connection to an EtherSpaceLink Device/File

7.1.2 Function Documentation

7.1.2.1 int EtherSpaceLink.device_type ( ) throws Exception

returns the device type

Returns

code indicating the device type

Exceptions

Exception	I/O error has occurred

7.1.2.2 int EtherSpaceLink.open (String *ip_address*) throws Exception

opens a connection to the specified device

Opens a connection the the etherspace link device specified which may be resolvable hostname or an ipv4 address.

A port number can be specified by adding a suffix with :portnumber. For example, 1.2.3.4:9999 will connect to a device at IP 1.2.3.4 with port number 9999

It also reads the table of modules installed in the EtherSpaceLink to an internal buffer, for use by procedures accessing status and module information. When opened, the SpaceWire link will be in the disabled state and its default speed will be 10Mb/s. Module and link parameters can be set immediately but the link must be started (using set_mode()) before data can be transferred over the SpaceWire link.

IPV6 is currently not supported by our devices

Note on the first call to this function we set the SIG_PIPE handler to SIG_IGN.

Parameters

address	The address / address:port specifier

Returns

EtherSpaceLink null if there was an error otherwise a EtherSpaceLink Handle

Exceptions

Exception	I/O error has occurred

7.2 Physical Link Attributes

Functions

- int EtherSpaceLink.set_active_link (int n) throws Exception

  Sets the currently active link.
- int EtherSpaceLink.set_mode (int mode) throws Exception

  set mode of current link
- int EtherSpaceLink.set_speed (int speed) throws Exception

  Sets the transmit speed of the link
  Sets the transmit speed of all of the SpaceWire links on thisEtherSpaceLink unit.
- int EtherSpaceLink.link_connected () throws Exception

  returns if the currently active link is connected
- int EtherSpaceLink.set_mode_portmask (int mode_, int ports_) throws Exception

  set mode of list of links

7.2.1 Detailed Description

Functions and definitions for controlling physical link attributes

7.2.2 Function Documentation

7.2.2.1 int EtherSpaceLink.link_connected ( ) throws Exception

returns if the currently active link is connected

Returns

0 not connected, < 0 if error, 1 connected

Exceptions

Exception	I/O error has occurred

7.2.2.2 int EtherSpaceLink.set_active_link (int *n*) throws Exception

Sets the currently active link.

Parameters

n	the link we want to make active

Returns

< 0 error, 0 success

Exceptions

Exception	I/O error has occurred

7.2.2.3 int EtherSpaceLink.set_mode (int *mode*) throws Exception

set mode of current link

Sets the operating mode of the currently-active SpaceWire link.

After opening a connection, the link is disabled; it must then be enabled into one of its operational modes before data can be transferred.

One of the three modes LINK_mode_disabled, LINK_mode_normal and LINK_mode_legacy should be chosen.

The use of LINK_mode_fixed_speed to set some DSI ports to 10Mb/s, together with the conventional set_speed() mechanism is the only way to run a DSIs links at two different speeds

Parameters

mode	of operation

LINK_mode_disabled

```
The link is idle and silent.
```

LINK_mode_normal

```
Start the link by actively trying to establish contact.
```

LINK_mode_legacy

```
Dont start until activity on the link is seen. Use with SMCS/TSS901 devices.
```

LINK_mode_long_timeout

```
    Extends the timeout period in the link state  machine to provide a potentialy more reliable link
    start at very low data rates (i.e. for slow (lowpower) links near to 2Mb/s). It is necessary to set
    the link speed with an   set_speed() API call before calling set_mode()  with this extra
    LINK_mode_slow_speed.


LINK_mode_fixed_speed

     The link speed remains at its default startup speed
     (10Mb/s nominal; actually within the range 9.8 to 10.2Mb/s)

LINK_mode_slow_speed

     This setting combines the long_timeout and fixed_speed modifiers, thereby also setting the
     initial link speed to the final operating speed.
```

Returns

0 if the request queued , !0 if not

Exceptions

Exception	I/O error has occurred

7.2.2.4 int EtherSpaceLink.set_mode_portmask (int *mode_*, int *ports_*) throws Exception

set mode of list of links

Sets the operating mode of a given set of links

After opening a connection, the link is disabled; it must then be enabled into one of its operational modes before data can be transferred.

The use of LINK_mode_fixed_speed to set some DSI ports to 10Mb/s, together with the conventional set_speed() mechanism is the only way to run a DSIs links at two different speeds. The active port is the highest number listed port in the mask

Parameters

mode	of operation

Returns

0 if the request queued , !0 if not

Exceptions

Exception	I/O error has occurred

7.2.2.5 int EtherSpaceLink.set_speed (int *speed*) throws Exception

Sets the transmit speed of the link

Sets the transmit speed of all of the SpaceWire links on thisEtherSpaceLink unit.

Links set with the additional mode modifier LINK_mode_fixed_speed, which remain at their start-up speed of 10-Mb/s.

Parameters

speed	the number of megabits per second

Returns

0 if request has been put on the wire, !0 if error

Exceptions

Exception	I/O error has occurred

7.3 Virtual Link Attributes

Modules

- TimeTag
- Error Reporting
- Error Waveforms

7.3.1 Detailed Description

Functions and definitions for controlling virtual link attributes

7.4 Handling Spacewire Traffic

Modules

- Event handling on spacewire links
- Sending data on a spaceiwre link
- Reading data from a spacewire link
- Extension codes

7.4.1 Detailed Description

Functions and definitions for handling spacewire traffic

7.5 Event handling on spacewire links

Modules

- functions

7.5.1 Detailed Description

Functions and definitions for handling events on spacewire links

7.6 Sending data on a spaceiwre link

Functions

- int EtherSpaceLink.flush () throws Exception

  transmit any buffered data
  write_packet may queue data for transport, this function puts queued data onto the wire
- int EtherSpaceLink.write_packet (byte[] netbuff, int length, int flags) throws Exception

  queue data for transmission
  Queues message for transmission, if there is no room left in the buffer, the buffer is transmitted. Note, that even the queued data is transmitted the data added to it may not be. If you want to guarantee transmission of this data you need to call flush.

7.6.1 Detailed Description

Functions and definitions for sending data

7.6.2 Function Documentation

7.6.2.1 int EtherSpaceLink.flush ( ) throws Exception

transmit any buffered data

write_packet may queue data for transport, this function puts queued data onto the wire

Parameters

buffer	the buffer to send
length	the length of the buffer to send
flags	indicating how the data to is be treated EOP the data is to be terminated with an EOP EEP the data is to be terminated with an EEP PART_EOP_EEP the data is not yet terminated INCOMPLETE the data is not yet terminated (but queued in such a way on termination it will be sent in one block)

If you logically OR the flags value with FLUSH a network flush is perfomed and the data is transmitted, if this is not performed data will be only transmitted when the network buffer is full or the flush method is called

Returns

0 if sucessfull, !0 if not, errno setup and error code in handle

Exceptions

Exception	I/O error has occurred

7.6.2.2 int EtherSpaceLink.write_packet (byte[] *netbuff,* int *length,* int *flags*) throws Exception

queue data for transmission

Queues message for transmission, if there is no room left in the buffer, the buffer is transmitted. Note, that even the queued data is transmitted the data added to it may not be. If you want to guarantee transmission of this data you need to call flush.

Parameters

buffer	the data to send
length	the size of the buffer to size
flags	additional metadata about the frame we are transmitting

EOP

```
This is the last part, or all, of a data packet; an end-ofpacket  (EOP) is added.
```

EEP

```
This is the last part, or all, of a data packet; an error endof packet (EEP) is added.
```

PART_EOP_EEP

```
This is part of a data packet; no end-of-packet is added. This effectively allows one to send
part packet data, do not rely on this working correctly with other devices as it is not part
of the spacewire specification.
```

EXTN

```
This is a complete extension character sequence. Extension packets have a maximum length of 60 bytes.
```

EherSpaceLink_SPECIAL

```
This is a complete special packet
```

EEP would not normally be used to terminate a packet but is available here to assist with testing where an erroneous packet may usefully be generated. Data is queued in buffers in the API in order to make best use of the TCP/IP stream and may not be sent immediately. flush() should be used to ensure the immediate transmission of any buffered data. The one-character extension sequences may be sent using write_EXTN.

Hardware dsi

Returns

0 if sucessfull, <0 if not (-error number), errno setup and error code in handle

Exceptions

Exception	I/O error has occurred

7.7 Reading data from a spacewire link

Functions

- int EtherSpaceLink.get_rx_flags () throws Exception

  return message flags of last packet data read
  Sometimes it may be necessary to obtain the message flags outside of the read call
- int EtherSpaceLink.get_packet (byte[] netbuff, int offset_, int buffer_length, int sda) throws Exception

  read packet not returning packet type.
  This function is similar to that of read_packet_full, however the rx_flags parameter is not present and as such can be retrieved by calling get_rx_flags

7.7.1 Detailed Description

Functions and definitions for reading data

7.7.2 Function Documentation

7.7.2.1 int EtherSpaceLink.get_packet (byte[] *netbuff,* int *offset_,* int *buffer_length,* int *sda*) throws Exception

read packet not returning packet type.

This function is similar to that of read_packet_full, however the rx_flags parameter is not present and as such can be retrieved by calling get_rx_flags

Parameters

buffer	where to read data into
offset	offset into the above buffer (i.e. data written to buffer+offset)
buffer_length	the number of bytes to read
sda	how to treat special_actions lower 4 bits enumerate to DISCARD_SPECIAL_DATA ignores special data REPORT_SPECIAL_DATA returns special data as -ve return value RETURN_SPECIAL_DATA returns data as normal message CALLBACK_SPECIAL_DATA calls callback upper 4 bits enumberate to DISCARD_EXTENSION_DATA (0) ignores extension data REPORT_EXTENSION_DATA returns extension data as -ve return value RETURN_EXTENSION_DATA returns data as normal message CALLBACK_EXTENSION_DATA calls callback

Returns

< 0 error code

Exceptions

Exception	I/O error has occurred

7.7.2.2 int EtherSpaceLink.get_rx_flags ( ) throws Exception

return message flags of last packet data read

Sometimes it may be necessary to obtain the message flags outside of the read call

This function returns the value of the flags performed by the last read call.

Exceptions

Exception	I/O error has occurred

7.8 TimeTag

Modules

- TimeTag mask fields

Functions

- int EtherSpaceLink.TT_enable_reporting (int when) throws Exception

  Enable timetags for currently active link.

7.8.1 Detailed Description

Functions and definitions for reporting Timetags

7.8.2 Function Documentation

7.8.2.1 int EtherSpaceLink.TT_enable_reporting (int *when*) throws Exception

Enable timetags for currently active link.

Parameters

when	what events generate a timetag

Returns

0 on success or transmitted , $<$ 0 if error

Exceptions

Exception	I/O error has occurred

7.9 Error Reporting

Modules

- Error mask fields

Functions

- int EtherSpaceLink.ER_enable_reporting (int what) throws Exception

  Enables, or disables, error reporting.

7.9.1 Detailed Description

Functions and definitions for reporting Errors

7.9.2 Function Documentation

7.9.2.1 int EtherSpaceLink.ER_enable_reporting (int *what*) throws Exception

Enables, or disables, error reporting.

Parameters

what	error reporting we wish to enable

Returns

0 on success or transmitted , $<$ 0 if error

Exceptions

Exception	I/O error has occurred

7.10 Error Waveforms

Modules

- Error Waveform Triggers
- Error Waveform Sources

Functions

- int EtherSpaceLink.EW_enable_reporting (int what) throws Exception

  Enables, or disables, waveform capture triggers.
  Triggering may be on errors or on other significant events. The parameter what should be set to EW_capture_nothing to disable all reporting, or to a combination of the.
- int EtherSpaceLink.EW_source (int sources) throws Exception

  Selects waveform capture trigger sources.
  Triggering may be on events from ports other than that associated with the capture circuit.

7.10.1 Detailed Description

Functions and definitions for capturing waveforms

7.10.2 Function Documentation

7.10.2.1 int EtherSpaceLink.EW_enable_reporting (int *what*) throws Exception

Enables, or disables, waveform capture triggers.

Triggering may be on errors or on other significant events. The parameter what should be set to EW_capture_nothing to disable all reporting, or to a combination of the.

In addition to the given triggers, a (non-maskable) EVENT in the DSI transmit data stream can also trigger a waveform capture. Each port of a DSI has a waveform capture circuit. Each capture circuit can be triggered by events on its own port, and also on other ports and external events. By default, each capture circuit will respond only to its own port. EW_source() can be used to expand the recognised source of triggers.

Parameters

what	error reporting we wish to enable

Returns

0 on success or transmitted , $<$ 0 if error

Exceptions

Exception	I/O error has occurred

7.10.2.2 int EtherSpaceLink.EW_source (int *sources*) throws Exception

Selects waveform capture trigger sources.

Triggering may be on events from ports other than that associated with the capture circuit.

For example, waveforms may be captured on all ports for an event occurring on only one of them.

Parameters

sources	EW_Source_port_1 Trigger on events from port 1. EW_Source_port_2 Trigger on events from port 2. EW_Source_port_3 Trigger on events from port 3. EW_Source_port_4 Trigger on events from port 4. EW_Source_port_5 Trigger on events from port 5. EW_Source_port_6 Trigger on events from port 6. EW_Source_port_7 Trigger on events from port 7. EW_Source_port_8 Trigger on events from port 8.

EW_Source_SMA_12 -LS, -MS∗ platforms Trigger on a rising edge on SMA connectors 1-2. The threshold level is 0.5 V. EW_Source_SMA_34 -LS, -MS platforms Trigger on a rising edge on SMA connectors 3-4. The threshold level is 0.5 V. EW_Source_SMA_56 -LS, -MS platforms Trigger on a rising edge on SMA connectors 5-6. The threshold level is 0.5 V. EW_Source_SMA_78 -LS, -MS platforms Trigger on a rising edge on SMA connectors 7-8. The threshold level is 0.5 V. EW_Source_barrier SO Trigger when the synchronisation barrier is lifted

Returns

0 if the request was successful

Exceptions

Exception	I/O error has occurred

7.11 Extension codes

Variables

- static final int EtherSpaceLink.FCT = 0x100
- static final int EtherSpaceLink.EEP = 0x101

  Error End of Packet.
- static final int EtherSpaceLink.EOP = 0x102

  End of Packet.
- static final int EtherSpaceLink.ESC = 0x103

  Escape.
- static final int EtherSpaceLink.ESC_FCT = 0x104

  Escape FCT aka a NULL character.
- static final int EtherSpaceLink.ESC_EEP = 0x105

  Escape End of Packet.
- static final int EtherSpaceLink.ESC_EOP = 0x106

  Escape Error of packet.
- static final int EtherSpaceLink.ESC_ESC = 0x107

  Escape Escape.
- static final int EtherSpaceLink.Timeout = 0x108

  Timeout message.
- static final int EtherSpaceLink.ParityError = 0x109

  Parity Error message.
- static final int EtherSpaceLink.PERROR1 = 0x10A

  Error 1 message.
- static final int EtherSpaceLink.PERROR2 = 0x10B

  Error 2 message.
- static final int EtherSpaceLink.STORE = 0x10C
- static final int EtherSpaceLink.FORWARD = 0x10D
- static final int EtherSpaceLink.ATOM = 0x10E
- static final int EtherSpaceLink.MOTA = 0x10F
- static final int EtherSpaceLink.JOIN = 0x110
- static final int EtherSpaceLink.BARRIER = 0x111
- static final int EtherSpaceLink.RESIGN = 0x112
- static final int EtherSpaceLink.EVENT = 0x113
- static final int EtherSpaceLink.Missing_data = 0x114

  Missed data message.
- static final int EtherSpaceLink.HOLD = 0x12F
- static final int EtherSpaceLink.Delay = 0x130
- static final int EtherSpaceLink.PortSelect = 0x140

  Port select message.
- static final int EtherSpaceLink.PortSelect_max = 0x17F

  Max port select message.
- static final int EtherSpaceLink.Multi_byte_extn_start = 0x180
- static final int EtherSpaceLink.TimeTag = 0x188

  Timetag message.
- static final int EtherSpaceLink.TimeTag_delta = 0x182

  Timetag delta message.
- static final int EtherSpaceLink.TimeTag_uncertainty = 0x181

  Timetag uncertain message.
- static final int EtherSpaceLink.TimeCode = 0x191

  Spacewire timecode.

- static final int EtherSpaceLink.Module = 0x192

  Module data.
- static final int EtherSpaceLink.TimeZero = 0x198

  First timecode on the link.
- static final int EtherSpaceLink.TRUNCATE_1 = 0x1A1
- static final int EtherSpaceLink.TRUNCATE_2 = 0x1A2
- static final int EtherSpaceLink.REPEAT_1 = 0x1B1
- static final int EtherSpaceLink.REPEAT_2 = 0x1B2
- static final int EtherSpaceLink.REPEAT_3 = 0x1B3
- static final int EtherSpaceLink.Year = 0x1C8

  Capture start date/time.
- static final int EtherSpaceLink.Header = 0x1CE

  Capture Header containing version and time information.

7.11.1 Detailed Description

7.11.2 Variable Documentation

7.11.2.1 final int EtherSpaceLink.ATOM = 0x10E [static]

7.11.2.2 final int EtherSpaceLink.BARRIER = 0x111 [static]

7.11.2.3 final int EtherSpaceLink.Delay = 0x130 [static]

7.11.2.4 final int EtherSpaceLink.EEP = 0x101 [static]

Error End of Packet.

7.11.2.5 final int EtherSpaceLink.EOP = 0x102 [static]

End of Packet.

7.11.2.6 final int EtherSpaceLink.ESC = 0x103 [static]

Escape.

7.11.2.7 final int EtherSpaceLink.ESC_EEP = 0x105 [static]

Escape End of Packet.

7.11.2.8 final int EtherSpaceLink.ESC_EOP = 0x106 [static]

Escape Error of packet.

7.11.2.9 final int EtherSpaceLink.ESC_ESC = 0x107 [static]

Escape Escape.

7.11.2.10 final int EtherSpaceLink.ESC_FCT = 0x104 [static]

Escape FCT aka a NULL character.

7.11.2.11 final int EtherSpaceLink.EVENT = 0x113 [static]

7.11.2.12 final int EtherSpaceLink.FCT = 0x100 [static]

7.11.2.13 final int EtherSpaceLink.FORWARD = 0x10D [static]

7.11.2.14 final int EtherSpaceLink.Header = 0x1CE [static]

Capture Header containing version and time information.

7.11.2.15 final int EtherSpaceLink.HOLD = 0x12F [static]

7.11.2.16 final int EtherSpaceLink.JOIN = 0x110 [static]

7.11.2.17 final int EtherSpaceLink.Missing_data = 0x114 [static]

Missed data message.

7.11.2.18 final int EtherSpaceLink.Module = 0x192 [static]

Module data.

7.11.2.19 final int EtherSpaceLink.MOTA = 0x10F [static]

7.11.2.20 final int EtherSpaceLink.Multi_byte_extn_start = 0x180 [static]

7.11.2.21 final int EtherSpaceLink.ParityError = 0x109 [static]

Parity Error message.

7.11.2.22 final int EtherSpaceLink.PERROR1 = 0x10A [static]

Error 1 message.

7.11.2.23 final int EtherSpaceLink.PERROR2 = 0x10B [static]

Error 2 message.

7.11.2.24 final int EtherSpaceLink.PortSelect = 0x140 [static]

Port select message.

7.11.2.25 final int EtherSpaceLink.PortSelect_max = 0x17F [static]

Max port select message.

7.11.2.26 final int EtherSpaceLink.REPEAT_1 = 0x1B1 [static]

7.11.2.27 final int EtherSpaceLink.REPEAT_2 = 0x1B2 [static]

7.11.2.28 final int EtherSpaceLink.REPEAT_3 = 0x1B3 [static]

7.11.2.29 final int EtherSpaceLink.RESIGN = 0x112 [static]

7.11.2.30 final int EtherSpaceLink.STORE = 0x10C [static]

7.11.2.31 final int EtherSpaceLink.TimeCode = 0x191 [static]

Spacewire timecode.

7.11.2.32 final int EtherSpaceLink.Timeout = 0x108 [static]

Timeout message.

7.11.2.33 final int EtherSpaceLink.TimeTag = 0x188 [static]

Timetag message.

7.11.2.34 final int EtherSpaceLink.TimeTag_delta = 0x182 [static]

Timetag delta message.

7.11.2.35 final int EtherSpaceLink.TimeTag_uncertainty = 0x181 [static]

Timetag uncertain message.

7.11.2.36 final int EtherSpaceLink.TimeZero = 0x198 [static]

First timecode on the link.

7.11.2.37 final int EtherSpaceLink.TRUNCATE_1 = 0x1A1 [static]

7.11.2.38 final int EtherSpaceLink.TRUNCATE_2 = 0x1A2 [static]

7.11.2.39 final int EtherSpaceLink.Year = 0x1C8 [static]

Capture start date/time.

7.12 TimeTag mask fields

Variables

- static final int EtherSpaceLink.TT = 7
- static final int EtherSpaceLink.TT_64 = 15
- static final int EtherSpaceLink.TT_report_nothing = 0x00

  Report Nothing.
- static final int EtherSpaceLink.TT_report_first_byte = (0x01 | report_first_byte)

  Timetag first byte of packet.
- static final int EtherSpaceLink.TT_report_intermediate_bytes = (0x02 | report_mid_bytes)

  Timetag middle byte.
- static final int EtherSpaceLink.TT_report_EOP_EEP = (0x04 | report_EEP | report_EOP)

  Timetag end of packet markers.
- static final int EtherSpaceLink.TT_report_EEP = (0x04 | report_EEP)

  Timetag report Error End of Packet.
- static final int EtherSpaceLink.TT_report_EOP = (0x04 | report_EOP)

  Timetag report End of Packet.
- static final int EtherSpaceLink.TT_report_time_code = report_time_code

  Timetag report spacewire timecode.
- static final int EtherSpaceLink.TT_report_fct = report_FCT

  Timetag report FCT.
- static final int EtherSpaceLink.TT_report_null = report_NULL

  Timetag report NULL.
- static final int EtherSpaceLink.TT_report_parity_error = report_parity_error

  Timetag report parity error.
- static final int EtherSpaceLink.TT_report_ESC_EOP = report_ESC_EOP

  Timetag report ESC End of Packet.
- static final int EtherSpaceLink.TT_report_ESC_EEP = report_ESC_EEP

  Timetag report ESC Error End of Packet.
- static final int EtherSpaceLink.TT_report_ESC_ESC = report_ESC_ESC

  Timetag report ESC ESC.
- static final int EtherSpaceLink.TT_report_timeout = report_timeout

  Timetag report timeout.

7.12.1 Detailed Description

7.12.2 Variable Documentation

7.12.2.1 final int EtherSpaceLink.TT = 7 [static]

7.12.2.2 final int EtherSpaceLink.TT_64 = 15 [static]

7.12.2.3 final int EtherSpaceLink.TT_report_EEP = (0x04 | report_EEP) [static]

Timetag report Error End of Packet.

7.12.2.4 final int EtherSpaceLink.TT_report_EOP = (0x04 | report_EOP) [static]

Timetag report End of Packet.

7.12.2.5 final int EtherSpaceLink.TT_report_EOP_EEP = (0x04 | report_EEP | report_EOP) [static]

Timetag end of packet markers.

7.12.2.6 final int EtherSpaceLink.TT_report_ESC_EEP = report_ESC_EEP [static]

Timetag report ESC Error End of Packet.

7.12.2.7 final int EtherSpaceLink.TT_report_ESC_EOP = report_ESC_EOP [static]

Timetag report ESC End of Packet.

7.12.2.8 final int EtherSpaceLink.TT_report_ESC_ESC = report_ESC_ESC [static]

Timetag report ESC ESC.

7.12.2.9 final int EtherSpaceLink.TT_report_fct = report_FCT [static]

Timetag report FCT.

7.12.2.10 final int EtherSpaceLink.TT_report_first_byte = (0x01 | report_first_byte) [static]

Timetag first byte of packet.

7.12.2.11 final int EtherSpaceLink.TT_report_intermediate_bytes = (0x02 | report_mid_bytes) [static]

Timetag middle byte.

7.12.2.12 final int EtherSpaceLink.TT_report_nothing = 0x00 [static]

Report Nothing.

7.12.2.13 final int EtherSpaceLink.TT_report_null = report_NULL [static]

Timetag report NULL.

7.12.2.14 final int EtherSpaceLink.TT_report_parity_error = report_parity_error [static]

Timetag report parity error.

7.12.2.15 final int EtherSpaceLink.TT_report_time_code = report_time_code [static]

Timetag report spacewire timecode.

7.12.2.16 final int EtherSpaceLink.TT_report_timeout = report_timeout [static]

Timetag report timeout.

7.13 Error mask fields

Variables

- static final int EtherSpaceLink.ER = 8
- static final int EtherSpaceLink.ER_64 = 16
- static final int EtherSpaceLink.ER_report_nothing = 0x00

  Error reporting report nothing.
- static final int EtherSpaceLink.ER_report_first_null = 0x02

  Error report first null.
- static final int EtherSpaceLink.ER_report_first_fct = 0x04

  Error report first fct.
- static final int EtherSpaceLink.ER_report_running_error = (0x08 | report_parity_error | report_ESC_EOP | report_ESC_EEP | report_ESC_ESC | report_timeout)

  Error report running.
- static final int EtherSpaceLink.ER_report_starting_error = 0x10
- static final int EtherSpaceLink.ER_report_nchar = 0x40
- static final int EtherSpaceLink.ER_report_time_code = (0x80 | report_time_code)

  report time code
- static final int EtherSpaceLink.ER_report_fct = report_FCT

  report FCT
- static final int EtherSpaceLink.ER_report_null = report_NULL

  report null
- static final int EtherSpaceLink.ER_report_parity_error = report_parity_error

  report parity error
- static final int EtherSpaceLink.ER_report_ESC_EOP = report_ESC_EOP

  report Escape End of Packet
- static final int EtherSpaceLink.ER_report_ESC_EEP = report_ESC_EEP

  report Escape Error End of Packet
- static final int EtherSpaceLink.ER_report_ESC_ESC = report_ESC_ESC

  report Escape Escape
- static final int EtherSpaceLink.ER_report_timeout = report_timeout

  report Timeout

7.13.1 Detailed Description

7.13.2 Variable Documentation

7.13.2.1 final int EtherSpaceLink.ER = 8 [static]

7.13.2.2 final int EtherSpaceLink.ER_64 = 16 [static]

7.13.2.3 final int EtherSpaceLink.ER_report_ESC_EEP = report_ESC_EEP [static]

report Escape Error End of Packet

7.13.2.4 final int EtherSpaceLink.ER_report_ESC_EOP = report_ESC_EOP [static]

report Escape End of Packet

7.13.2.5 final int EtherSpaceLink.ER_report_ESC_ESC = report_ESC_ESC [static]

report Escape Escape

7.13.2.6 final int EtherSpaceLink.ER_report_fct = report_FCT [static]

report FCT

7.13.2.7 final int EtherSpaceLink.ER_report_first_fct = 0x04 [static]

Error report first fct.

7.13.2.8 final int EtherSpaceLink.ER_report_first_null = 0x02 [static]

Error report first null.

7.13.2.9 final int EtherSpaceLink.ER_report_nchar = 0x40 [static]

7.13.2.10 final int EtherSpaceLink.ER_report_nothing = 0x00 [static]

Error reporting report nothing.

7.13.2.11 final int EtherSpaceLink.ER_report_null = report_NULL [static]

report null

7.13.2.12 final int EtherSpaceLink.ER_report_parity_error = report_parity_error [static]

report parity error

7.13.2.13 final int EtherSpaceLink.ER_report_running_error = (0x08 | report_parity_error | report_ESC_EOP | report_ESC_EEP | report_ESC_ESC | report_timeout) [static]

Error report running.

7.13.2.14 final int EtherSpaceLink.ER_report_starting_error = 0x10 [static]

7.13.2.15 final int EtherSpaceLink.ER_report_time_code = (0x80 | report_time_code) [static]

report time code

7.13.2.16 final int EtherSpaceLink.ER_report_timeout = report_timeout [static]

report Timeout

7.14 Error Waveform Triggers

Variables

- static final int EtherSpaceLink.EW = 9
- static final int EtherSpaceLink.EW_RT = 13
- static final int EtherSpaceLink.EW_capture_nothing = 0x00
- static final int EtherSpaceLink.EW_capture_first_null = (0x02 | report_first_null)

  trigger on first null
- static final int EtherSpaceLink.EW_capture_first_fct = 0x04

  trigger on first fct
- static final int EtherSpaceLink.EW_capture_running_error = (0x08 | report_parity_error | report_ESC_EOP | report_ESC_EEP | report_ESC_ESC | report_timeout)

  trigger on run error
- static final int EtherSpaceLink.EW_capture_starting_error = 0x10

  trigger on start error
- static final int EtherSpaceLink.EW_capture_nchar = (0x40 | report_nchar)

  trigger on n char
- static final int EtherSpaceLink.EW_capture_time_code = (0x80 | report_time_code)

  trigger on timecode
- static final int EtherSpaceLink.EW_capture_EOP = report_EOP

  trigger on End of Packet
- static final int EtherSpaceLink.EW_capture_EEP = report_EEP

  trigger on Error End of Packet
- static final int EtherSpaceLink.EW_capture_FCT = report_FCT

  trigger on FCT
- static final int EtherSpaceLink.EW_capture_excess_FCT = report_excess_FCT

  trigger on excess fct
- static final int EtherSpaceLink.EW_capture_excess_data = report_excess_data

  trigger on excess data
- static final int EtherSpaceLink.EW_capture_null = report_NULL

  trigger on NULL
- static final int EtherSpaceLink.EW_capture_parity_error = report_parity_error

  trigger on parity error
- static final int EtherSpaceLink.EW_capture_ESC_EOP = report_ESC_EOP

  trigger on Escape End of Packet
- static final int EtherSpaceLink.EW_capture_ESC_EEP = report_ESC_EEP

  trigger on Escape Error End of Packet
- static final int EtherSpaceLink.EW_capture_ESC_ESC = report_ESC_ESC

  trigger on Escape Escape
- static final int EtherSpaceLink.EW_capture_timeout = report_timeout

  trigger on timeout

7.14.1 Detailed Description

7.14.2 Variable Documentation

7.14.2.1 final int EtherSpaceLink.EW = 9 [static]

7.14.2.2 final int EtherSpaceLink.EW_capture_EEP = report_EEP [static]

trigger on Error End of Packet

7.14.2.3 final int EtherSpaceLink.EW_capture_EOP = report_EOP [static]

trigger on End of Packet

7.14.2.4 final int EtherSpaceLink.EW_capture_ESC_EEP = report_ESC_EEP [static]

trigger on Escape Error End of Packet

7.14.2.5 final int EtherSpaceLink.EW_capture_ESC_EOP = report_ESC_EOP [static]

trigger on Escape End of Packet

7.14.2.6 final int EtherSpaceLink.EW_capture_ESC_ESC = report_ESC_ESC [static]

trigger on Escape Escape

7.14.2.7 final int EtherSpaceLink.EW_capture_excess_data = report_excess_data [static]

trigger on excess data

7.14.2.8 final int EtherSpaceLink.EW_capture_excess_FCT = report_excess_FCT [static]

trigger on excess fct

7.14.2.9 final int EtherSpaceLink.EW_capture_FCT = report_FCT [static]

trigger on FCT

7.14.2.10 final int EtherSpaceLink.EW_capture_first_fct = 0x04 [static]

trigger on first fct

7.14.2.11 final int EtherSpaceLink.EW_capture_first_null = (0x02 | report_first_null) [static]

trigger on first null

7.14.2.12 final int EtherSpaceLink.EW_capture_nchar = (0x40 | report_nchar) [static]

trigger on n char

7.14.2.13 final int EtherSpaceLink.EW_capture_nothing = 0x00 [static]

7.14.2.14 final int EtherSpaceLink.EW_capture_null = report_NULL [static]

trigger on NULL

7.14.2.15 final int EtherSpaceLink.EW_capture_parity_error = report_parity_error [static]

trigger on parity error

7.14.2.16 final int EtherSpaceLink.EW_capture_running_error = (0x08 | report_parity_error | report_ESC_EOP | report_ESC_EEP | report_ESC_ESC | report_timeout) [static]

trigger on run error

7.14.2.17 final int EtherSpaceLink.EW_capture_starting_error = 0x10 [static]

trigger on start error

7.14.2.18 final int EtherSpaceLink.EW_capture_time_code = (0x80 | report_time_code) [static]

trigger on timecode

7.14.2.19 final int EtherSpaceLink.EW_capture_timeout = report_timeout [static]

trigger on timeout

7.14.2.20 final int EtherSpaceLink.EW_RT = 13 [static]

7.15 Error Waveform Sources

Variables

- static final int EtherSpaceLink.EW_Source_barrier = 0x0001

  Barrier.
- static final int EtherSpaceLink.EW_Source_port_1 = 0x0002

  Port 1.
- static final int EtherSpaceLink.EW_Source_port_2 = 0x0004

  Port 2.
- static final int EtherSpaceLink.EW_Source_port_3 = 0x0008

  Port 3.
- static final int EtherSpaceLink.EW_Source_port_4 = 0x0010

  Port 4.
- static final int EtherSpaceLink.EW_Source_port_5 = 0x0020

  Port 5.
- static final int EtherSpaceLink.EW_Source_port_6 = 0x0040

  Port 6.
- static final int EtherSpaceLink.EW_Source_port_7 = 0x0080

  Port 7.
- static final int EtherSpaceLink.EW_Source_port_8 = 0x0100

  Port 8.
- static final int EtherSpaceLink.EW_Source_SMA_12 = 0x0200

  SMA 1/2 changing state.
- static final int EtherSpaceLink.EW_Source_SMA_34 = 0x0400

  SMA 3/4 changing state.
- static final int EtherSpaceLink.EW_Source_SMA_56 = 0x0800

  SMA 5/6 changing state.
- static final int EtherSpaceLink.EW_Source_SMA_78 = 0x1000

  SMA 7/8 changing state.
- static final int EtherSpaceLink.EW_Source_local_clock = 0x8000

  Local clock.

7.15.1 Detailed Description

7.15.2 Variable Documentation

7.15.2.1 final int EtherSpaceLink.EW_Source_barrier = 0x0001 [static]

Barrier.

7.15.2.2 final int EtherSpaceLink.EW_Source_local_clock = 0x8000 [static]

Local clock.

7.15.2.3 final int EtherSpaceLink.EW_Source_port_1 = 0x0002 [static]

Port 1.

7.15.2.4 final int EtherSpaceLink.EW_Source_port_2 = 0x0004 [static]

Port 2.

7.15.2.5 final int EtherSpaceLink.EW_Source_port_3 = 0x0008 [static]

Port 3.

7.15.2.6 final int EtherSpaceLink.EW_Source_port_4 = 0x0010 [static]

Port 4.

7.15.2.7 final int EtherSpaceLink.EW_Source_port_5 = 0x0020 [static]

Port 5.

7.15.2.8 final int EtherSpaceLink.EW_Source_port_6 = 0x0040 [static]

Port 6.

7.15.2.9 final int EtherSpaceLink.EW_Source_port_7 = 0x0080 [static]

Port 7.

7.15.2.10 final int EtherSpaceLink.EW_Source_port_8 = 0x0100 [static]

Port 8.

7.15.2.11 final int EtherSpaceLink.EW_Source_SMA_12 = 0x0200 [static]

SMA 1/2 changing state.

7.15.2.12 final int EtherSpaceLink.EW_Source_SMA_34 = 0x0400 [static]

SMA 3/4 changing state.

7.15.2.13 final int EtherSpaceLink.EW_Source_SMA_56 = 0x0800 [static]

SMA 5/6 changing state.

7.15.2.14 final int EtherSpaceLink.EW_Source_SMA_78 = 0x1000 [static]

SMA 7/8 changing state.

7.16 Memory Mapped Addresses

Variables

- static final int EtherSpaceLink.LINK_address = 0x0000
- static final int EtherSpaceLink.TX_SPEED_address = 0x87FD
- static final int EtherSpaceLink.RX_SPEED_address = 0x0001
- static final int EtherSpaceLink.HWA_address = 0x8800
- static final int EtherSpaceLink.VERSION_address = 0x880A
- static final int EtherSpaceLink.DESCRIPTION_address = 0x880B
- static final int EtherSpaceLink.OPTIONS_address = 0x8F60
- static final int EtherSpaceLink.NLINKS_address = 0x8FFF
- static final int EtherSpaceLink.EW_address = 0x1000
- static final int EtherSpaceLink.PC_address = 0x2000
- static final int EtherSpaceLink.PG_address = 0x4000
- static final int EtherSpaceLink.ATI_address = 0x0100
- static final int EtherSpaceLink.OBSERVE_address = 0x0020
- static final int EtherSpaceLink.TIMETAG_address = 0x0030
- static final int EtherSpaceLink.IGNORE_address = 0x0040
- static final int EtherSpaceLink.Event_cause_address = 0x0060
- static final int EtherSpaceLink.EW_source_address = 0x0070
- static final int EtherSpaceLink.FLOW_CONTROL_address = 0x0050
- static final int EtherSpaceLink.SMA_56_pulse_width_address = 0x00F0
- static final int EtherSpaceLink.max_packet_data = 0x0010

7.16.1 Detailed Description

Error codes which API calls may set and be retrieved by the get error call

7.16.2 Variable Documentation

7.16.2.1 final int EtherSpaceLink.ATI_address = 0x0100 [static]

7.16.2.2 final int EtherSpaceLink.DESCRIPTION_address = 0x880B [static]

7.16.2.3 final int EtherSpaceLink.Event_cause_address = 0x0060 [static]

7.16.2.4 final int EtherSpaceLink.EW_address = 0x1000 [static]

7.16.2.5 final int EtherSpaceLink.EW_source_address = 0x0070 [static]

7.16.2.6 final int EtherSpaceLink.FLOW_CONTROL_address = 0x0050 [static]

7.16.2.7 final int EtherSpaceLink.HWA_address = 0x8800 [static]

7.16.2.8 final int EtherSpaceLink.IGNORE_address = 0x0040 [static]

7.16.2.9 final int EtherSpaceLink.LINK_address = 0x0000 [static]

7.16.2.10 final int EtherSpaceLink.max_packet_data = 0x0010 [static]

7.16.2.11 final int EtherSpaceLink.NLINKS_address = 0x8FFF [static]

7.16.2.12 final int EtherSpaceLink.OBSERVE_address = 0x0020 [static]

7.16.2.13 final int EtherSpaceLink.OPTIONS_address = 0x8F60 [static]

7.16.2.14 final int EtherSpaceLink.PC_address = 0x2000 [static]

7.16.2.15 final int EtherSpaceLink.PG_address = 0x4000 [static]

7.16.2.16 final int EtherSpaceLink.RX_SPEED_address = 0x0001 [static]

7.16.2.17 final int EtherSpaceLink.SMA_56_pulse_width_address = 0x00F0 [static]

7.16.2.18 final int EtherSpaceLink.TIMETAG_address = 0x0030 [static]

7.16.2.19 final int EtherSpaceLink.TX_SPEED_address = 0x87FD [static]

7.16.2.20 final int EtherSpaceLink.VERSION_address = 0x880A [static]

7.17 Error Codes

Variables

- static final int EtherSpaceLink.Error_RecFile_Open = -1

  Couldn't open recording file.
- static final int EtherSpaceLink.Error_RecFile_Write = -2

  record_file write failed
- static final int EtherSpaceLink.Error_LogFile_Open = -3

  Couldn't open logging file.
- static final int EtherSpaceLink.Error_LogFile_Write = -4

  log_file write failed
- static final int EtherSpaceLink.Error_Receiver_Timeout = -10

  we have a network timeout timeout
- static final int EtherSpaceLink.Error_Receiver_Shutdown = -11

  peer has performed an orderly shutdown
- static final int EtherSpaceLink.Error_IO_Error = -12

  we have an IO error
- static final int EtherSpaceLink.Error_SaveBuf_Overflow_Save = -15

  Saving the read_packet_full() save_buffer failed.
- static final int EtherSpaceLink.Error_SaveBuf_Overflow_Restore = -16

  Restoring the read_packet_full() save_buffer failed.
- static final int EtherSpaceLink.Error_Function_Not_Supported = -17

  Device does not support the requested function.
- static final int EtherSpaceLink.Error_Network = -18

  Error reading / writing to/from the device.
- static final int EtherSpaceLink.Error_Network_Format_Error = -19

  Error understanding recieved packet.
- static final int EtherSpaceLink.Error_Request_Too_Large = -20

  The I/O request can't be fullfilled by the hardware.
- static final int EtherSpaceLink.Error_Sequence_Error = -21

  Didn't receive expected notification from the hardware.
- static final int EtherSpaceLink.Error_Response_Too_Small = -22

  Response from the device didn't contain enough data.
- static final int EtherSpaceLink.Error_Response_Mismatch = -23

  Response does not match I/O request.
- static final int EtherSpaceLink.Error_Module_Not_Present = -24

  Module not present.
- static final int EtherSpaceLink.Error_Parameter_RangeIncorrect = -25

  Parameter not in range.
- static final int EtherSpaceLink.Error_File_Not_Present = -26

  Requested file is not present.
- static final int EtherSpaceLink.Error_EINTR = -27

  EINTR occurred.
- static final int EtherSpaceLink.Error_Link_Incorrect = -28

  Link number is incorrect.
- static final int EtherSpaceLink.Error_Incorrect_Device = -29

  Connecting to a device which does not support functionality.
- static final int EtherSpaceLink.Error_Memory = -30

  Unable to allocate memory.
- static final int EtherSpaceLink.Error_Host_Unresolvable = -31

Unable to resolve host.

- static final int EtherSpaceLink.Error_Host_Unresponsive = -32

  Unable to connect to host.
- static final int EtherSpaceLink.Error_WaveForm_Dir_Create = -33

  Unable to create waveform directory.
- static final int EtherSpaceLink.Error_Zero_Read = -34

  asked to read zero bytes
- static final int EtherSpaceLink.Error_Set_Option_File = -35

  Asked to set an option when playing back from file.
- static final int EtherSpaceLink.Error_Invalid_Device = -36

  Device is not supported by API.
- static final int EtherSpaceLink.Error_File_Move = -37

  Unable to move file into place.
- static final int EtherSpaceLink.Error_Invalid_File = -38

  Unable to open file.
- static final int EtherSpaceLink.Error_Callback_Return = -39

  Callback has asked for a return.
- static final int EtherSpaceLink.Error_FileList_Empty = -40

  List of files given is empty.
- static final int EtherSpaceLink.Error_Unknown_System_Type = -41

  Unknown type.
- static final int EtherSpaceLink.Error_Not_Known = -42

  API returned 0 as an error should (should not happen)
- static final int EtherSpaceLink.Error_EXE_Start_Failed = -43

  Cannot start executable.
- static final int EtherSpaceLink.Error_NO_Connection = -44

  Link Not estblished.
- static final int EtherSpaceLink.Error_Invalid_Link = -45

  Invalid Link selected.
- static final int EtherSpaceLink.Error_Would_Block = -48

  I/O call would block.
- static final int EtherSpaceLink.Error_Link_Not_Connected = -49

  Link Not Connected.
- static final int EtherSpaceLink.Error_ReadHandler_Running = -50

  There is a read handler running for this connection.
- static final int EtherSpaceLink.Error_Buffer_Full = -51

  can't do non blocking write as buffer is full
- static final int EtherSpaceLink.Error_CaptureThread_Failed = -52

  Capture thread failed.
- static final int EtherSpaceLink.Option_SO = 1

  Option SO module is installed.

7.17.1 Detailed Description

Error codes which API calls may set and be retrieved by the get error call

7.17.2 Variable Documentation

7.17.2.1 final int EtherSpaceLink.Error_Buffer_Full = -51 [static]

can't do non blocking write as buffer is full

7.17.2.2 final int EtherSpaceLink.Error_Callback_Return = -39 [static]

Callback has asked for a return.

7.17.2.3 final int EtherSpaceLink.Error_CaptureThread_Failed = -52 [static]

Capture thread failed.

7.17.2.4 final int EtherSpaceLink.Error_EINTR = -27 [static]

EINTR occurred.

7.17.2.5 final int EtherSpaceLink.Error_EXE_Start_Failed = -43 [static]

Cannot start executable.

7.17.2.6 final int EtherSpaceLink.Error_File_Move = -37 [static]

Unable to move file into place.

7.17.2.7 final int EtherSpaceLink.Error_File_Not_Present = -26 [static]

Requested file is not present.

7.17.2.8 final int EtherSpaceLink.Error_FileList_Empty = -40 [static]

List of files given is empty.

7.17.2.9 final int EtherSpaceLink.Error_Function_Not_Supported = -17 [static]

Device does not support the requested function.

7.17.2.10 final int EtherSpaceLink.Error_Host_Unresolvable = -31 [static]

Unable to resolve host.

7.17.2.11 final int EtherSpaceLink.Error_Host_Unresponsive = -32 [static]

Unable to connect to host.

7.17.2.12 final int EtherSpaceLink.Error_Incorrect_Device = -29 [static]

Connecting to a device which does not support functionality.

7.17.2.13 final int EtherSpaceLink.Error_Invalid_Device = -36 [static]

Device is not supported by API.

7.17.2.14 final int EtherSpaceLink.Error_Invalid_File = -38 [static]

Unable to open file.

7.17.2.15 final int EtherSpaceLink.Error_Invalid_Link = -45 [static]

Invalid Link selected.

7.17.2.16 final int EtherSpaceLink.Error_IO_Error = -12 [static]

we have an IO error

7.17.2.17 final int EtherSpaceLink.Error_Link_Incorrect = -28 [static]

Link number is incorrect.

7.17.2.18 final int EtherSpaceLink.Error_Link_Not_Connected = -49 [static]

Link Not Connected.

7.17.2.19 final int EtherSpaceLink.Error_LogFile_Open = -3 [static]

Couldn't open logging file.

7.17.2.20 final int EtherSpaceLink.Error_LogFile_Write = -4 [static]

log_file write failed

7.17.2.21 final int EtherSpaceLink.Error_Memory = -30 [static]

Unable to allocate memory.

7.17.2.22 final int EtherSpaceLink.Error_Module_Not_Present = -24 [static]

Module not present.

7.17.2.23 final int EtherSpaceLink.Error_Network = -18 [static]

Error reading / writing to/from the device.

7.17.2.24 final int EtherSpaceLink.Error_Network_Format_Error = -19 [static]

Error understanding recieved packet.

7.17.2.25 final int EtherSpaceLink.Error_NO_Connection = -44 [static]

Link Not estblished.

7.17.2.26 final int EtherSpaceLink.Error_Not_Known = -42 [static]

API returned 0 as an error should (should not happen)

7.17.2.27 final int EtherSpaceLink.Error_Parameter_RangeIncorrect = -25 [static]

Parameter not in range.

7.17.2.28 final int EtherSpaceLink.Error_ReadHandler_Running = -50 [static]

There is a read handler running for this connection.

7.17.2.29 final int EtherSpaceLink.Error_Receiver_Shutdown = -11 [static]

peer has performed an orderly shutdown

7.17.2.30 final int EtherSpaceLink.Error_Receiver_Timeout = -10 [static]

we have a network timeout timeout

7.17.2.31 final int EtherSpaceLink.Error_RecFile_Open = -1 [static]

Couldn't open recording file.

7.17.2.32 final int EtherSpaceLink.Error_RecFile_Write = -2 [static]

record_file write failed

7.17.2.33 final int EtherSpaceLink.Error_Request_Too_Large = -20 [static]

The I/O request can't be fullfilled by the hardware.

7.17.2.34 final int EtherSpaceLink.Error_Response_Mismatch = -23 [static]

Response does not match I/O request.

7.17.2.35 final int EtherSpaceLink.Error_Response_Too_Small = -22 [static]

Response from the device didn't contain enough data.

7.17.2.36 final int EtherSpaceLink.Error_SaveBuf_Overflow_Restore = -16 [static]

Restoring the read_packet_full() save_buffer failed.

7.17.2.37 final int EtherSpaceLink.Error_SaveBuf_Overflow_Save = -15 [static]

Saving the read_packet_full() save_buffer failed.

7.17.2.38 final int EtherSpaceLink.Error_Sequence_Error = -21 [static]

Didn't receive expected notification from the hardware.

7.17.2.39 final int EtherSpaceLink.Error_Set_Option_File = -35 [static]

Asked to set an option when playing back from file.

7.17.2.40 final int EtherSpaceLink.Error_Unknown_System_Type = -41 [static]

Unknown type.

7.17.2.41 final int EtherSpaceLink.Error_WaveForm_Dir_Create = -33 [static]

Unable to create waveform directory.

7.17.2.42 final int EtherSpaceLink.Error_Would_Block = -48 [static]

I/O call would block.

7.17.2.43 final int EtherSpaceLink.Error_Zero_Read = -34 [static]

asked to read zero bytes

7.17.2.44 final int EtherSpaceLink.Option_SO = 1 [static]

Option SO module is installed.

7.18 functions

Functions

- boolean EtherSpaceLink.waveform_data (int unit, long time_, int erridx, int port, int ew_port, int ew_event, String fullcause, byte[] special_, int sz)
- boolean EtherSpaceLink.raw_waveform_data (byte[] special_, int sz_)
- boolean EtherSpaceLink.timecode (byte tc_, byte a_, byte b_, byte t_)
- boolean EtherSpaceLink.timetag_uncertainty (int uncertainty_)
- boolean EtherSpaceLink.missing_data ()
- boolean EtherSpaceLink.esc_eep ()
- boolean EtherSpaceLink.esc_eop ()
- boolean EtherSpaceLink.esc_esc ()
- void EtherSpaceLink.link_selected (int link_)
- boolean EtherSpaceLink.link_timeout ()
- boolean EtherSpaceLink.port_status (int link_, double rxspeed_, boolean connected_, int runstatus_)
- boolean EtherSpaceLink.link_tx_speed (int link_, double txspeed_)
- boolean EtherSpaceLink.parity_error ()
- boolean EtherSpaceLink.error_event ()
- boolean EtherSpaceLink.perror1 ()
- boolean EtherSpaceLink.perror2 ()
- boolean EtherSpaceLink.received_fct ()
- boolean EtherSpaceLink.received_esc_fct ()
- boolean EtherSpaceLink.timezero (long time_)
- boolean EtherSpaceLink.timetag (long time_)

  time tag
- boolean EtherSpaceLink.err (double time_, int state_, int error_bits_)

  time tag
- boolean EtherSpaceLink.unknown_ram_data (byte[] data_, int length_, boolean complete_, int data_buffer_position_)
- boolean EtherSpaceLink.unknown_special_data (byte[] data_, int length_, boolean complete_, int data_buffer_position_)
- boolean EtherSpaceLink.unknown_extn_data (byte[] data_, int length_, boolean complete_, int data_buffer_position_)
- boolean EtherSpaceLink.device_type (int type_)
- boolean EtherSpaceLink.start_time (int year, int month, int day, int hour, int minute, int second, long ns)

7.18.1 Detailed Description

These functions are called when an event on a spacewire link occurs

7.18.2 Function Documentation

7.18.2.1 boolean EtherSpaceLink.device_type (int *type_*)

callback informing the application of the device the recording is for

Parameters

*type_*	the device type

7.18.2.2 boolean EtherSpaceLink.err (double *time_*, int *state_*, int *error_bits_*)

time tag

callback informing the application of an error on the currently active rx link

Parameters

*time_*	the current time
*state_*	the state of the port
*error_bits_*	mask of error bits

7.18.2.3 boolean EtherSpaceLink.error_event ( )

callback informing the application that the currently active port has encountered an error with error waveform data available

7.18.2.4 boolean EtherSpaceLink.esc_eep ( )

callback informing the application the device has seen an escape eep frame

7.18.2.5 boolean EtherSpaceLink.esc_eop ( )

callback informing the application the device has seen an escape eop frame

7.18.2.6 boolean EtherSpaceLink.esc_esc ( )

callback informing the application the device has seen an escape escape frame

7.18.2.7 void EtherSpaceLink.link_selected (int *link_*)

callback when the rx link has changed

Parameters

*link_*	the newly active rx link

7.18.2.8 boolean EtherSpaceLink.link_timeout ( )

callback informing the application that the currently rx link has timedout (disconnected?)

7.18.2.9 boolean EtherSpaceLink.link_tx_speed (int *link_*, double *txspeed_*)

callback informing the application that a port has changed TX speed

Parameters

*link_*	the link to which this pertains
*txspeed_*	the TX speed

7.18.2.10 boolean EtherSpaceLink.missing_data ( )

callback informing the application that the device has missed data

7.18.2.11 boolean EtherSpaceLink.parity_error ( )

callback informing the application that the currently active link has had a parity error

7.18.2.12 boolean EtherSpaceLink.perror1 ( )

callback informing the application that the currently active port has had an error

7.18.2.13 boolean EtherSpaceLink.perror2 ( )

callback informing the application that the currently active port has had an error

7.18.2.14 boolean EtherSpaceLink.port_status (int *link_*, double *rxspeed_*, boolean *connected_*, int *runstatus_*)

callback informing the application that a port has changed status

Parameters

*link_*	the link to which this pertains
*rxspeed_*	the RX speed
*connected_*	is the link connected
runstatus	the raw status

7.18.2.15 boolean EtherSpaceLink.raw_waveform_data (byte[] *special_*, int *sz_*)

< Error Code

< Unit

< Error index

7.18.2.16 boolean EtherSpaceLink.received_esc_fct ( )

callback informing the application that an ESC FCT has been received (aka a NULL)

7.18.2.17 boolean EtherSpaceLink.received_fct ( )

callback informing the application that an FCT has been received

7.18.2.18 boolean EtherSpaceLink.start_time (int *year*, int *month*, int *day*, int *hour*, int *minute*, int *second*, long *ns*)

callback informing the application of the time the device started recording

Parameters

year	
month	
day	
minute	
second	
ns	

7.18.2.19 boolean EtherSpaceLink.timecode (byte *tc_*, byte *a_*, byte *b_*, byte *t_*)

callback informing the application of the timecode

Parameters

time	code the timecode

7.18.2.20 boolean EtherSpaceLink.timetag (long *time_*)

time tag

callback informing the application of the currently active rx time

Parameters

*time_*	the current time, the number of 1/10ths of nano seconds since the beginning of the year if synchronised with GPS otherwise the number of 1/10ths since power on.

7.18.2.21 boolean EtherSpaceLink.timetag_uncertainty (int *uncertainty_*)

callback informing the application of the time tag uncertainty

Parameters

*uncertainty_*	the current uncertainty

7.18.2.22 boolean EtherSpaceLink.timezero (long *time_*)

callback informing the application of the first packet received

Parameters

*time_*	the current time, the number of 1/10ths of nano seconds since the beginning of the year if synchronised with GPS otherwise the number of 1/10ths since power on.

7.18.2.23 boolean EtherSpaceLink.unknown_extn_data (byte[] *data_*, int *length_*, boolean *complete_*, int *data_buffer_position_*)

callback informing the application of unknown extension data

Parameters

*data_*	the data from ram i/o module
*length_*	the length of the data
*complete_*	is the data complete (if not more to come)
*data_buffer_-position_*	the current position in the stream of data

7.18.2.24 boolean EtherSpaceLink.unknown_ram_data (byte[] *data_*, int *length_*, boolean *complete_*, int *data_buffer_position_*)

callback informing the application of data from an unknown module (should never trigger)

Parameters

*data_*	the data from ram i/o module
*length_*	the length of the data
*complete_*	is the data complete (if not more to come)
*data_buffer_position_*	the current position in the stream of data

7.18.2.25 boolean EtherSpaceLink.unknown_special_data (byte[] *data_*, int *length_*, boolean *complete_*, int *data_buffer_position_*)

callback informing the application of unknown special data

Parameters

*data_*	the data from ram i/o module
*length_*	the length of the data
*complete_*	is the data complete (if not more to come)
*data_buffer_position_*	the current position in the stream of data

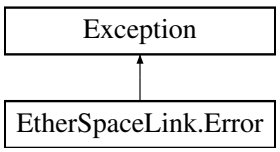
7.18.2.26 boolean EtherSpaceLink.waveform_data (int *unit*, long *time_*, int *erridx*, int *port*, int *ew_port*, int *ew_event*, String *fullcause*, byte[] *special_*, int *sz*)

Chapter 8

Class Documentation

8.1 EtherSpaceLink.Error Class Reference

Inheritance diagram for EtherSpaceLink.Error:



Public Member Functions

- Error (int error)
- int Get ()

8.1.1 Constructor & Destructor Documentation

8.1.1.1 EtherSpaceLink.Error.Error (int *error*)

8.1.2 Member Function Documentation

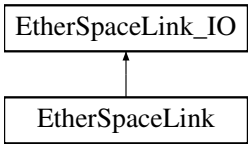
8.1.2.1 int EtherSpaceLink.Error.Get ( )

The documentation for this class was generated from the following file:

- /autogen/EtherSpaceLink.java

8.2 EtherSpaceLink Class Reference

Inheritance diagram for EtherSpaceLink:



Classes

- class Error

Public Member Functions

- EtherSpaceLink (String addr) throws Exception

  Constructor connecting to a network device.
- EtherSpaceLink (File file) throws Exception

  Construtor taking a file generated by record file for playback.
- boolean empty_frame (long sz_)
- int receivePacket (byte[] data, long offset, int f) throws Exception

  Receives packet data.
- boolean connected () throws Exception

  Returns if the currently selected link is connected.
- void set_extension_cb (ESL_CB cb)
- void esl_delay (int ms_)
- int flow_control (int initial_fcts, boolean no_more_fcts, boolean ignore_flow_control) throws Exception

  EI flow_control.
- synchronized int sendPacket_S (byte[] data_buffer, int start, int length, int terminator) throws Exception
- synchronized int flush_S () throws Exception

  Thread safe flush method.
- void set_special_cb (ESL_CB cb)

  Sets special callback handler.
- boolean waveform_data (int unit, long time_, int erridx, int port, int ew_port, int ew_event, String fullcause, byte[] special_, int sz)
- boolean raw_waveform_data (byte[] special_, int sz_)
- boolean timecode (byte tc_, byte a_, byte b_, byte t_)
- boolean timetag_uncertainty (int uncertainty_)
- boolean missing_data ()
- boolean esc_eep ()
- boolean esc_eop ()
- boolean esc_esc ()
- void link_selected (int link_)
- boolean link_timeout ()
- boolean port_status (int link_, double rxspeed_, boolean connected_, int runstatus_)
- boolean link_tx_speed (int link_, double txspeed_)
- boolean parity_error ()
- boolean error_event ()
- boolean perror1 ()
- boolean perror2 ()
- boolean received_fct ()
- boolean received_esc_fct ()
- boolean timezero (long time_)
- boolean timetag (long time_)

  time tag
- boolean err (double time_, int state_, int error_bits_)

  time tag
- boolean unknown_ram_data (byte[] data_, int length_, boolean complete_, int data_buffer_position_)
- boolean unknown_special_data (byte[] data_, int length_, boolean complete_, int data_buffer_position_)
- boolean unknown_extn_data (byte[] data_, int length_, boolean complete_, int data_buffer_position_)
- boolean device_type (int type_)
- boolean start_time (int year, int month, int day, int hour, int minute, int second, long ns)

- boolean handle_year (byte[] extn_data_buffer)
- boolean handle_module (byte[] extn_data_buffer) throws Exception
- boolean handle_header (byte[] extn_data_buffer)
- boolean extn_data (byte[] data_, int length_, boolean complete_, int data_buffer_position_) throws Exception
- boolean raw_link_address (byte[] data_, int length_) throws Exception
- boolean raw_speed_address (byte[] data_, int length_) throws Exception
- boolean raw_ram_rw (byte[] data_, int length_, int data_buffer_position_) throws Exception
- int set_active_link_401 (int n) throws Exception
- int set_mode_401 (int mode) throws Exception
- int set_speed_401 (int speed) throws Exception
- String get_module_string_401 (int module) throws Exception
- String get_product_string_401 () throws Exception
- int link_connected_401 () throws Exception
- int set_speed_double_401 (int speed) throws Exception
- String get_manufacturer_string_401 () throws Exception
- String get_options_string_401 () throws Exception
- int get_HWA_401 (byte[] hwa) throws Exception
- int TT_enable_reporting_401 (int when) throws Exception
- int ER_enable_reporting_401 (int what) throws Exception
- int Observe_401 (int what) throws Exception
- int EW_enable_reporting_401 (int what) throws Exception
- int EW_source_401 (int sources) throws Exception
- int EW_request_data_401 (int port) throws Exception
- int EW_reset_401 (int port) throws Exception
- int EW_clear_401 (int port) throws Exception
- int EI_ignore_events_401 (int what) throws Exception
- int EI_flow_control_401 (int initial_fcts, int flow_control) throws Exception
- double get_timetag_401 (byte[] netbuff) throws Exception
- int write_buffer_empty_401 () throws Exception
- int get_receive_speed_401 () throws Exception
- int get_number_of_links_401 () throws Exception
- int set_max_packet_data_401 (int N) throws Exception
- int ATI_calibrate_401 (int v) throws Exception
- int sma_56_pulse_width_401 (int width) throws Exception
- double extract_timetag_401 (byte[] netbuff) throws Exception
- long extract_timetag_ns_401 (byte[] netbuff) throws Exception
- int ram_io_get_int (int address, int bytes) throws Exception
- int ram_io_write_bytes (int port, int address, byte[] netbuff, int sz) throws Exception
- int set_active_link_408 (int n) throws Exception
- int set_mode_408 (int mode) throws Exception
- int set_speed_408 (int speed) throws Exception
- String get_module_string_408 (int module) throws Exception
- String get_product_string_408 () throws Exception
- int link_connected_408 () throws Exception
- int set_speed_double_408 (int speed) throws Exception
- String get_manufacturer_string_408 () throws Exception
- String get_options_string_408 () throws Exception
- int get_HWA_408 (byte[] hwa) throws Exception
- int TT_enable_reporting_408 (int when) throws Exception
- int ER_enable_reporting_408 (int what) throws Exception
- int Observe_408 (int what) throws Exception
- int EW_enable_reporting_408 (int what) throws Exception
- int EW_source_408 (int sources) throws Exception
- int EW_request_data_408 (int port) throws Exception
- int EW_reset_408 (int port) throws Exception

- int EW_clear_408 (int port) throws Exception
- int EI_ignore_events_408 (int what_) throws Exception
- int EI_flow_control_408 (int initial_fcts, int flow_control) throws Exception
- double get_timetag_408 (byte[] netbuff) throws Exception
- int write_buffer_empty_408 () throws Exception
- int get_receive_speed_408 () throws Exception
- int get_number_of_links_408 () throws Exception
- int set_max_packet_data_408 (int N) throws Exception
- int ATI_calibrate_408 (int v) throws Exception
- int sma_56_pulse_width_408 (int width) throws Exception
- double extract_timetag_408 (byte[] netbuff) throws Exception
- long extract_timetag_ns_408 (byte[] netbuff) throws Exception
- int send (byte[] buffer, int len, int flags) throws Exception

  Low level call to send data on ESL transport.
- int device_type () throws Exception

  returns the device type
- FileOutputStream get_record_file ()

  Retreive current recording file.
- long get_record_size ()

  Return the amount of data written to the current recording file.
- void set_EINTR (int eintr)

  sets behaviour when network i/o is interrupted
- int record_writes ()

  Returns whether errors writing to record file are treated as errors.
- void set_context (Object ctx)

  Set user context can be associated with a EtherSpaceLink handle.
- Object get_context ()

  Return user context associated with a EtherSpaceLink handle.
- int set_rx_timeout (int to) throws Exception

  Sets the rx timeout in milliseconds.
- int get_rx_timeout ()

  Retreive the current recieve timeout (milliseconds)
- String get_version ()

  Used to get the version number of the C API being used.
- double get_percent_file_read ()

  returns how much data has been read from the file
- void abort () throws Exception

  Use to abort a connection with a thread on a read call.
- int close () throws Exception

  Used to close connection to an ESL device or file.
- int shutdown () throws Exception

  Used to terminate a connection to an ESL device.
- int flush () throws Exception

  transmit any buffered data
  write_packet may queue data for transport, this function puts queued data onto the wire
- int get_rx_flags () throws Exception

  return message flags of last packet data read
  Sometimes it may be necessary to obtain the message flags outside of the read call
- int get_packet (byte[] netbuff, int offset_, int buffer_length, int sda) throws Exception

  read packet not returning packet type.
  This function is similar to that of read_packet_full, however the rx_flags parameter is not present and as such can be retrieved by calling get_rx_flags

- int sendPacket (byte[] netbuff, int offset, int length, int flags_) throws Exception
- int write_packet (byte[] netbuff, int length, int flags) throws Exception

  queue data for transmission
  Queues message for transmission, if there is no room left in the buffer, the buffer is transmitted. Note, that even the queued data is transmitted the data added to it may not be. If you want to guarantee transmission of this data you need to call flush.
- int negotiate () throws Exception
- int open (String ip_address) throws Exception

  opens a connection to the specified device
- int set_active_link (int n) throws Exception

  Sets the currently active link.
- int set_mode (int mode) throws Exception

  set mode of current link
- int write_EXTN (int extn) throws Exception

  Sends an extension packet.
- int get_module_slot (int module) throws Exception

  Returns the slot a given module resides in.
- int set_speed (int speed) throws Exception

  Sets the transmit speed of the link
  Sets the transmit speed of all of the SpaceWire links on thisEtherSpaceLink unit.
- int get_control_packet (byte[] control_buffer, int buffer_length, int slot) throws Exception

  request control data for slot
- String get_module_string (int module) throws Exception

  Returns the name of a given module.
- String get_product_string () throws Exception

  Returns an ASCII string indicating the product name.
- int link_connected () throws Exception

  returns if the currently active link is connected
- int set_log_file (String file_name) throws Exception

  Sets the current log file.
- int set_tx_record_file (String file_name) throws Exception

  Sets the tx recording file.
- int set_record_file (String file_name) throws Exception

  Sets the recording file.
- int flush_record_file () throws Exception

  Flush record file.
- void check_record_writes (int on) throws Exception

  Sets up what to do in the case of an error writing to record file.
- long get_total_raw_bytes_received () throws Exception

  returns the total number of bytes read.
- void set_rx_timeout_action (int action) throws Exception

  sets the behaviour on a networktimeout
- void dump_max (int dump_max) throws Exception

  Sets the number of message dumps on any one run.
- double decode_fp16 (int fp16) throws Exception
- int set_speed_double (double speed) throws Exception

  Sets the transmit speed of the link allowing partial Mb speeds.
- int get_slot (int slot) throws Exception

  Returns the module in a given slot.
- int get_module_type (byte[] netbuff) throws Exception

  Return the module type from the given network buffer.
- String get_manufacturer_string () throws Exception

Returns manufacturer string.

- String get_options_string () throws Exception

  Returns an ASCII string indicating the list of options installed on the device.
- int get_HWA (byte[] hwa) throws Exception

  Reads the hardware address (MAC) for the ESL device.
- int TT_enable_reporting (int when) throws Exception

  Enable timetags for currently active link.
- int ER_enable_reporting (int what) throws Exception

  Enables, or disables, error reporting.
- int Observe (int what) throws Exception

  request info from the device
- int EW_enable_reporting (int what) throws Exception

  Enables, or disables, waveform capture triggers.
  Triggering may be on errors or on other significant events. The parameter what should be set to EW_capture_nothing to disable all reporting, or to a combination of the.
- int EW_source (int sources) throws Exception

  Selects waveform capture trigger sources.
  Triggering may be on events from ports other than that associated with the capture circuit.
- int EW_request_data (int port) throws Exception

  When Error Waveform reporting is switched on, it is possible that the device can indicate that it has an error waveform available via an Extension data block of data. If you want to record this data , you must request it. This function queues the request and the device will then send the error waveform data which is sent as a SPECIAL block of data.
- int EW_reset (int port) throws Exception

  Re-arms the capture of error waveforms having previously captured a waveform.
- int EW_clear (int port) throws Exception

  This function is called to clear waveform data from the port.
- int EI_ignore_events (int what) throws Exception

  Ignore events from the EI module, the EI module will disconnect a link in the event of an error, this function allows the link to ignore errors and continue running in the event of a particular error.
- int EI_flow_control (int initial_fcts, int flow_control) throws Exception

  Sets Error Injection flow control characteristics.
- double extract_timetag (byte[] netbuff) throws Exception

  Extract timetag from special data callback.
- long extract_timetag_ns (byte[] netbuff) throws Exception

  Extract timetag from special data callback data in tenths of nano seconds.
- int request_link_status () throws Exception

  Requests the link status.
- int request_link_status_port (int port) throws Exception

  Requests the link status for a given port On the ESL range of devices this can change the active transmission port.
- int request_rx_speed () throws Exception

  Requests the rx speed.
- int request_tx_speed () throws Exception

  requests the tx speed for the device
- int write_buffer_empty () throws Exception

  Returns if the write buffer for the currently active link is empty.
- int get_receive_speed () throws Exception

  Gets the receive speed of the currently active link.
- int get_number_of_links () throws Exception

  Returns the number of links a device has.
- int set_max_packet_data (int N) throws Exception

  Control the data receive compressor - discard data from packet.
- int ATI_calibrate (int v) throws Exception

- int sma_56_pulse_width (int width) throws Exception
- String HWA_to_serial_number_string (byte[] hwa) throws Exception

  Converts MAC hardware to human readable string.
- int get_options () throws Exception

  Returns an integer bitmap indicating the list of options installed on the device.
- int send_timecode (byte tc_) throws Exception

  Sends a timcode packet on the current link.
- int delay (byte delay_) throws Exception
- int send_ESC () throws Exception
- int send_ESC_EOP () throws Exception
- int send_ESC_EEP () throws Exception
- int send_ESC_ESC () throws Exception
- int send_ESC_FCT () throws Exception
- int SO_BARRIER () throws Exception
- int SO_FORWARD () throws Exception
- int SO_JOIN () throws Exception
- int SO_STORE () throws Exception
- void system_type (int syst_) throws Exception

  Forces system type such that platform derived timetags can be read.
- void set_slot (int index_, int type_) throws Exception

  Sets a particular slot to a given slot type (used internally)
- int set_mode_portmask (int mode_, int ports_) throws Exception

  set mode of list of links
- int fastclose () throws Exception

  Sets the SO_LINGER timeout to 0 such that when the connection is closed, it is closed quickly.

Public Attributes

- int rx_link

Static Public Attributes

- static final String version = "ESL_RELID"
- static final int PART_EOP_EEP = 1000

  Error packet.
- static final int SPECIAL = 1003

  We are sending a special frame.
- static final int EXTN = 1005

  We are sending an extension frame.
- static final int PART_EXTN = 1006
- static final int INCOMPLETE = 1008

  Used to build up a single packet for the unit.
- static final int FLUSH = 2048
- static final int PART_SPECIAL = 1004

  A part of a special frame.
- static final int TRUNCATED = 1007

  Aritifical construct for unhandled data.
- static final int SPECIAL_SIZE = 1009

  Returning the amount of special data.
- static final int EXTENSION_SIZE = 1010

  Returning the amount of extension data.

- static final int FCT = 0x100
- static final int EEP = 0x101

  Error End of Packet.
- static final int EOP = 0x102

  End of Packet.
- static final int ESC = 0x103

  Escape.
- static final int ESC_FCT = 0x104

  Escape FCT aka a NULL character.
- static final int ESC_EEP = 0x105

  Escape End of Packet.
- static final int ESC_EOP = 0x106

  Escape Error of packet.
- static final int ESC_ESC = 0x107

  Escape Escape.
- static final int Timeout = 0x108

  Timeout message.
- static final int ParityError = 0x109

  Parity Error message.
- static final int PERROR1 = 0x10A

  Error 1 message.
- static final int PERROR2 = 0x10B

  Error 2 message.
- static final int STORE = 0x10C
- static final int FORWARD = 0x10D
- static final int ATOM = 0x10E
- static final int MOTA = 0x10F
- static final int JOIN = 0x110
- static final int BARRIER = 0x111
- static final int RESIGN = 0x112
- static final int EVENT = 0x113
- static final int Missing_data = 0x114

  Missed data message.
- static final int HOLD = 0x12F
- static final int Delay = 0x130
- static final int PortSelect = 0x140

  Port select message.
- static final int PortSelect_max = 0x17F

  Max port select message.
- static final int Multi_byte_extn_start = 0x180
- static final int TimeTag = 0x188

  Timetag message.
- static final int TimeTag_delta = 0x182

  Timetag delta message.
- static final int TimeTag_uncertainty = 0x181

  Timetag uncertain message.
- static final int TimeCode = 0x191

  Spacewire timecode.
- static final int Module = 0x192

  Module data.
- static final int TimeZero = 0x198

First timecode on the link.

- static final int TRUNCATE_1 = 0x1A1
- static final int TRUNCATE_2 = 0x1A2
- static final int REPEAT_1 = 0x1B1
- static final int REPEAT_2 = 0x1B2
- static final int REPEAT_3 = 0x1B3
- static final int Year = 0x1C8

  Capture start date/time.
- static final int Header = 0x1CE

  Capture Header containing version and time information.
- static final int report_EEP = 0x800000

  EEP error event.
- static final int report_nchar = 0x400000

  character received event
- static final int report_first_null = 0x200000

  First null event.
- static final int report_excess_FCT = 0x100000

  Too many FCTS event.
- static final int report_excess_data = 0x080000

  Too much data sent for # of FCT's.
- static final int report_first_byte = 0x040000

  First byte of packet.
- static final int report_mid_bytes = 0x020000

  Frame mide byte.
- static final int report_EOP = 0x010000

  EOP recieved.
- static final int report_time_code = 0x008000

  Time code received.
- static final int report_FCT = 0x004000

  FCT received.
- static final int report_NULL = 0x002000

  Null received.
- static final int report_parity_error = 0x001000

  Parity Error.
- static final int report_ESC_EOP = 0x000800

  Escape EOP error.
- static final int report_ESC_EEP = 0x000400

  Escape EEP Error.
- static final int report_ESC_ESC = 0x000200

  Escape Escape Error.
- static final int report_timeout = 0x000100

  Link Timeout.
- static final int report_delta = 0x400000
- static final int DISCARD_SPECIAL_DATA = 0x00
- static final int REPORT_SPECIAL_DATA = 0x01
- static final int RETURN_SPECIAL_DATA = 0x02
- static final int CALLBACK_SPECIAL_DATA = 0x03
- static final int SPECIAL_DATA_FLAGS = 0x03
- static final int DISCARD_EXTENSION_DATA = 0x00
- static final int REPORT_EXTENSION_DATA = 0x10
- static final int RETURN_EXTENSION_DATA = 0x20
- static final int CALLBACK_EXTENSION_DATA = 0x30

- static final int EXTENSION_DATA_FLAGS = 0x30
- static final int READ_IMMEDIATE = 0x40
- static final int SpaceWire_state_ErrorReset = 0
- static final int SpaceWire_state_ErrorWait = 1
- static final int SpaceWire_state_Ready = 2
- static final int SpaceWire_state_Started = 3
- static final int SpaceWire_state_Connecting = 4
- static final int SpaceWire_state_Run = 5
- static final int MSR_state_NC = 6
- static final int MSR_state_Connected = 7
- static final int CAPABILITIES = 0
- static final int HWA = 3
- static final int LINK_SPEED = 4
- static final int MANUFACTURER = 1
- static final int PRODUCT = 2
- static final int LINK = 5
- static final int LINK_mode_disabled = 0x01

  Disables the link.
- static final int LINK_mode_normal = 0x02

  Enables the link.
- static final int LINK_mode_legacy = 0x04

  IEEE 1355 (spacewire precursor)
- static final int LINK_mode_master = 0x06

  IEEE 1355 (precurosr)
- static final int LINK_tx_buffer_empty = 0x08
- static final int LINK_state_offset = 4
- static final int LINK_mode_long_timeout = 0x40
- static final int LINK_mode_fixed_speed = 0x80
- static final int LINK_mode_slow_speed = 0xC0
- static final int SF = 6
- static final int SF_disabled = 0x00
- static final int SF_enabled = 0x80
- static final int TT = 7
- static final int TT_64 = 15
- static final int TT_report_nothing = 0x00

  Report Nothing.
- static final int TT_report_first_byte = (0x01 | report_first_byte)

  Timetag first byte of packet.
- static final int TT_report_intermediate_bytes = (0x02 | report_mid_bytes)

  Timetag middle byte.
- static final int TT_report_EOP_EEP = (0x04 | report_EEP | report_EOP)

  Timetag end of packet markers.
- static final int TT_report_EEP = (0x04 | report_EEP)

  Timetag report Error End of Packet.
- static final int TT_report_EOP = (0x04 | report_EOP)

  Timetag report End of Packet.
- static final int TT_report_time_code = report_time_code

  Timetag report spacewire timecode.
- static final int TT_report_fct = report_FCT

  Timetag report FCT.
- static final int TT_report_null = report_NULL

  Timetag report NULL.

- static final int TT_report_parity_error = report_parity_error

  Timetag report parity error.
- static final int TT_report_ESC_EOP = report_ESC_EOP

  Timetag report ESC End of Packet.
- static final int TT_report_ESC_EEP = report_ESC_EEP

  Timetag report ESC Error End of Packet.
- static final int TT_report_ESC_ESC = report_ESC_ESC

  Timetag report ESC ESC.
- static final int TT_report_timeout = report_timeout

  Timetag report timeout.
- static final int ER = 8
- static final int ER_64 = 16
- static final int ER_report_nothing = 0x00

  Error reporting report nothing.
- static final int ER_report_first_null = 0x02

  Error report first null.
- static final int ER_report_first_fct = 0x04

  Error report first fct.
- static final int ER_report_running_error = (0x08 | report_parity_error | report_ESC_EOP | report_ESC_EEP | report_ESC_ESC | report_timeout)

  Error report running.
- static final int ER_report_starting_error = 0x10
- static final int ER_report_nchar = 0x40
- static final int ER_report_time_code = (0x80 | report_time_code)

  report time code
- static final int ER_report_fct = report_FCT

  report FCT
- static final int ER_report_null = report_NULL

  report null
- static final int ER_report_parity_error = report_parity_error

  report parity error
- static final int ER_report_ESC_EOP = report_ESC_EOP

  report Escape End of Packet
- static final int ER_report_ESC_EEP = report_ESC_EEP

  report Escape Error End of Packet
- static final int ER_report_ESC_ESC = report_ESC_ESC

  report Escape Escape
- static final int ER_report_timeout = report_timeout

  report Timeout
- static final int EW = 9
- static final int EW_RT = 13
- static final int EW_capture_nothing = 0x00
- static final int EW_capture_first_null = (0x02 | report_first_null)

  trigger on first null
- static final int EW_capture_first_fct = 0x04

  trigger on first fct
- static final int EW_capture_running_error = (0x08 | report_parity_error | report_ESC_EOP | report_ESC_EEP | report_ESC_ESC | report_timeout)

  trigger on run error
- static final int EW_capture_starting_error = 0x10

  trigger on start error

- static final int EW_capture_nchar = (0x40 | report_nchar)

  trigger on n char
- static final int EW_capture_time_code = (0x80 | report_time_code)

  trigger on timecode
- static final int EW_capture_EOP = report_EOP

  trigger on End of Packet
- static final int EW_capture_EEP = report_EEP

  trigger on Error End of Packet
- static final int EW_capture_FCT = report_FCT

  trigger on FCT
- static final int EW_capture_excess_FCT = report_excess_FCT

  trigger on excess fct
- static final int EW_capture_excess_data = report_excess_data

  trigger on excess data
- static final int EW_capture_null = report_NULL

  trigger on NULL
- static final int EW_capture_parity_error = report_parity_error

  trigger on parity error
- static final int EW_capture_ESC_EOP = report_ESC_EOP

  trigger on Escape End of Packet
- static final int EW_capture_ESC_EEP = report_ESC_EEP

  trigger on Escape Error End of Packet
- static final int EW_capture_ESC_ESC = report_ESC_ESC

  trigger on Escape Escape
- static final int EW_capture_timeout = report_timeout

  trigger on timeout
- static final int EW_Source_barrier = 0x0001

  Barrier.
- static final int EW_Source_port_1 = 0x0002

  Port 1.
- static final int EW_Source_port_2 = 0x0004

  Port 2.
- static final int EW_Source_port_3 = 0x0008

  Port 3.
- static final int EW_Source_port_4 = 0x0010

  Port 4.
- static final int EW_Source_port_5 = 0x0020

  Port 5.
- static final int EW_Source_port_6 = 0x0040

  Port 6.
- static final int EW_Source_port_7 = 0x0080

  Port 7.
- static final int EW_Source_port_8 = 0x0100

  Port 8.
- static final int EW_Source_SMA_12 = 0x0200

  SMA 1/2 changing state.
- static final int EW_Source_SMA_34 = 0x0400

  SMA 3/4 changing state.
- static final int EW_Source_SMA_56 = 0x0800

  SMA 5/6 changing state.
- static final int EW_Source_SMA_78 = 0x1000

SMA 7/8 changing state.

- static final int EW_Source_local_clock = 0x8000

  Local clock.
- static final int TC_rx = 10
- static final int TC_rx_64 = 17
- static final int TC_rx_silent = 0x00
- static final int TC_rx_report_enabled = 0x08
- static final int TC_rx_time_stamp_enabled = 0x40
- static final int TC_tx = 11
- static final int TC_tx_trigger_mask = 0x03
- static final int TC_tx_no_trigger = 0x00
- static final int TC_tx_one_code = 0x01
- static final int TC_tx_external_trigger = 0x02
- static final int TC_tx_regular_trigger = 0x03
- static final int TC_tx_update_interval = 0x04
- static final int TC_tx_update_code = 0x08
- static final int TC_tx_format_mask = 0x30
- static final int TC_tx_no_increment = 0x00
- static final int TC_tx_increment_6_bits = 0x10
- static final int TC_tx_increment_7_bits = 0x20
- static final int TC_tx_increment_8_bits = 0x30
- static final int TC_tx_report_transmission = 0x40
- static final int CR = 14
- static final int router_cs = 18
- static final int router_tables = 19
- static final int router_stats = 20
- static final int ram_rw = 21
- static final int barrier = 22
- static final int TT_now = 23
- static final int EI_ignore_excess_FCT = report_excess_FCT
- static final int EI_ignore_excess_data = report_excess_data
- static final int EI_ignore_parity_error = report_parity_error
- static final int EI_ignore_ESC_EOP = report_ESC_EOP
- static final int EI_ignore_ESC_EEP = report_ESC_EEP
- static final int EI_ignore_ESC_ESC = report_ESC_ESC
- static final int EI_ignore_timeout = report_timeout
- static final int EI_normal_flow_control = 0x00
- static final int EI_transmit_anyway = 0x20
- static final int EI_no_automatic_FCT = 0x10
- static final int LINK_address = 0x0000
- static final int TX_SPEED_address = 0x87FD
- static final int RX_SPEED_address = 0x0001
- static final int HWA_address = 0x8800
- static final int VERSION_address = 0x880A
- static final int DESCRIPTION_address = 0x880B
- static final int OPTIONS_address = 0x8F60
- static final int NLINKS_address = 0x8FFF
- static final int EW_address = 0x1000
- static final int PC_address = 0x2000
- static final int PG_address = 0x4000
- static final int ATI_address = 0x0100
- static final int OBSERVE_address = 0x0020
- static final int TIMETAG_address = 0x0030
- static final int IGNORE_address = 0x0040

- static final int Event_cause_address = 0x0060
- static final int EW_source_address = 0x0070
- static final int FLOW_CONTROL_address = 0x0050
- static final int SMA_56_pulse_width_address = 0x00F0
- static final int max_packet_data = 0x0010
- static final int Error_RecFile_Open = -1

  Couldn't open recording file.
- static final int Error_RecFile_Write = -2

  record_file write failed
- static final int Error_LogFile_Open = -3

  Couldn't open logging file.
- static final int Error_LogFile_Write = -4

  log_file write failed
- static final int Error_Receiver_Timeout = -10

  we have a network timeout timeout
- static final int Error_Receiver_Shutdown = -11

  peer has performed an orderly shutdown
- static final int Error_IO_Error = -12

  we have an IO error
- static final int Error_SaveBuf_Overflow_Save = -15

  Saving the read_packet_full() save_buffer failed.
- static final int Error_SaveBuf_Overflow_Restore = -16

  Restoring the read_packet_full() save_buffer failed.
- static final int Error_Function_Not_Supported = -17

  Device does not support the requested function.
- static final int Error_Network = -18

  Error reading / writing to/from the device.
- static final int Error_Network_Format_Error = -19

  Error understanding recieved packet.
- static final int Error_Request_Too_Large = -20

  The I/O request can't be fullfilled by the hardware.
- static final int Error_Sequence_Error = -21

  Didn't receive expected notification from the hardware.
- static final int Error_Response_Too_Small = -22

  Response from the device didn't contain enough data.
- static final int Error_Response_Mismatch = -23

  Response does not match I/O request.
- static final int Error_Module_Not_Present = -24

  Module not present.
- static final int Error_Parameter_RangeIncorrect = -25

  Parameter not in range.
- static final int Error_File_Not_Present = -26

  Requested file is not present.
- static final int Error_EINTR = -27

  EINTR occurred.
- static final int Error_Link_Incorrect = -28

  Link number is incorrect.
- static final int Error_Incorrect_Device = -29

  Connecting to a device which does not support functionality.
- static final int Error_Memory = -30

  Unable to allocate memory.

- static final int Error_Host_Unresolvable = -31

  Unable to resolve host.
- static final int Error_Host_Unresponsive = -32

  Unable to connect to host.
- static final int Error_WaveForm_Dir_Create = -33

  Unable to create waveform directory.
- static final int Error_Zero_Read = -34

  asked to read zero bytes
- static final int Error_Set_Option_File = -35

  Asked to set an option when playing back from file.
- static final int Error_Invalid_Device = -36

  Device is not supported by API.
- static final int Error_File_Move = -37

  Unable to move file into place.
- static final int Error_Invalid_File = -38

  Unable to open file.
- static final int Error_Callback_Return = -39

  Callback has asked for a return.
- static final int Error_FileList_Empty = -40

  List of files given is empty.
- static final int Error_Unknown_System_Type = -41

  Unknown type.
- static final int Error_Not_Known = -42

  API returned 0 as an error should (should not happen)
- static final int Error_EXE_Start_Failed = -43

  Cannot start executable.
- static final int Error_NO_Connection = -44

  Link Not estblished.
- static final int Error_Invalid_Link = -45

  Invalid Link selected.
- static final int Error_Would_Block = -48

  I/O call would block.
- static final int Error_Link_Not_Connected = -49

  Link Not Connected.
- static final int Error_ReadHandler_Running = -50

  There is a read handler running for this connection.
- static final int Error_Buffer_Full = -51

  can't do non blocking write as buffer is full
- static final int Error_CaptureThread_Failed = -52

  Capture thread failed.
- static final int Option_SO = 1

  Option SO module is installed.
- static final int CONNECT_FILE = (1)
- static final int Receiver_Timeout_Returns_Zero_Part_Pkt = 0
- static final int Receiver_Timeout_Returns_Error = 1
- static final int SYSTEM_TYPE_INVALID = 0
- static final int SYSTEM_TYPE_401 = 1
- static final int SYSTEM_TYPE_408 = 2

Protected Attributes

- long rawtime

Package Functions

- int sendPacket_S (byte[] data, int sz, int terminator) throws Exception

  Queues a packet of data synchronized such that different threads can send data.
- int sendPacket_S (byte[] data, int terminator) throws Exception

  Queues a packet of data synchronized such that different threads can send data.
- int sendPacket (byte[] data, int sz, int terminator) throws Exception

  Queues a packet of data.
- int sendPacket (byte[] data, int terminator) throws Exception

  Queues a packet of data.
- int get_terminator ()
- boolean special_data (byte[] data_, int length_, boolean complete_, int data_buffer_position_) throws Exception
- int extract_link_state (byte[] netbuff, int buflen)
- int extract_link (byte[] netbuff, int buflen)
- double extract_rx_speed (byte[] netbuff, int buflen) throws Exception
- double extract_tx_speed (byte[] netbuff, int buflen) throws Exception
- void enable_callbacks ()

  Enable Out of Band data callbacks.
- void log_packet (int direction, byte[] buffer, int length, int flags)

Static Package Functions

- static String what (int w)
- static int update (int original, String s)
- static int update_es (int original, String s)
- static String source (int w)

8.2.1 Constructor & Destructor Documentation

8.2.1.1 EtherSpaceLink.EtherSpaceLink (String *addr*) throws Exception

Constructor connecting to a network device.

Parameters

addr	the address/port to connect to

Exceptions

Exception	I/O error

8.2.1.2 EtherSpaceLink.EtherSpaceLink (File *file*) throws Exception

Construtor taking a file generated by record file for playback.

Parameters

file	the File to playback through the application

Exceptions

Exception	I/O error

8.2.2 Member Function Documentation

8.2.2.1 void EtherSpaceLink.abort ( ) throws Exception

Use to abort a connection with a thread on a read call.

Reading packets can block on an underlying network read and if you have a running thread on that read you will have to wait for a packet to arrive such that it can see you want to terminate the thread.

abort knocks the thread off the read call. Once you have used this call you can't read from the device again.

Exceptions

Exception	I/O error has occurred

8.2.2.2 int EtherSpaceLink.ATI_calibrate (int *v*) throws Exception

Returns

$<$ 0 indicating an error

Exceptions

Exception	I/O error has occurred

8.2.2.3 int EtherSpaceLink.ATI_calibrate_401 (int *v*) throws Exception

8.2.2.4 int EtherSpaceLink.ATI_calibrate_408 (int *v*) throws Exception

8.2.2.5 void EtherSpaceLink.check_record_writes (int *on*) throws Exception

Sets up what to do in the case of an error writing to record file.

This function sets whether the read packet functions return an error when an error occurs writing to the record file.

Generally it is advisable to do this as you may end up with truncated log files

Parameters

on	non zero means we record errors writing to the log file , the default is 0

Exceptions

Exception	I/O error has occurred

8.2.2.6 int EtherSpaceLink.close ( ) throws Exception

Used to close connection to an ESL device or file.

Returns

null to dereference the connection

Exceptions

Exception	I/O error has occurred

8.2.2.7 boolean EtherSpaceLink.connected ( ) throws Exception

Returns if the currently selected link is connected.

Returns

true if connected

Exceptions

Exception	I/O error

8.2.2.8 double EtherSpaceLink.decode_fp16 (int *fp16*) throws Exception

8.2.2.9 int EtherSpaceLink.delay (byte *delay_*) throws Exception

8.2.2.10 boolean EtherSpaceLink.device_type (int *type_*)

callback informing the application of the device the recording is for

Parameters

*type_*	the device type

8.2.2.11 int EtherSpaceLink.device_type ( ) throws Exception

returns the device type

Returns

code indicating the device type

Exceptions

Exception	I/O error has occurred

8.2.2.12 void EtherSpaceLink.dump_max (int *dump_max*) throws Exception

Sets the number of message dumps on any one run.

Parameters

dump_max	number of dumps to do

Exceptions

Exception	I/O error has occurred

8.2.2.13 int EtherSpaceLink.EI_flow_control (int *initial_fcts*, int *flow_control*) throws Exception

Sets Error Injection flow control characteristics.

The ECSS-E-ST-50-12C SpaceWire standard requires at least one flow-control token to be sent to start a link, and the maximum flow-control credit to be 56 N-Chars (as indicated by 7 flow-control tokens). Flow-control tokens are normally issued as space becomes available in the receive buffer. Data tokens (actually N-Chars - data, end-of-packet and error-end-of-packet) may be received up to the number for which credit has been issued. The EI_flow_control() of the DSI allows the link to be set outside the limits defined by the ECSS SpaceWire standard. An initial flow-control issue of 0 tokens should result in a link not starting. An initial flow-control issue of 8 or more FCTs should cause an error. Automatic issuing of flow-control tokens can be suppressed by the DSI, leaving the user to explicitly issue flow-control tokens (see write_EXTN). Also, the transmission of data can be allowed by the DSI despite there being no available credit - in order to test a receivers behaviour. The initial_fcts parameter of EI_flow_control() sets the number of flow-control tokens that are sent when the link starts. This value must be between 0 and 15; otherwise, an error is reported. The flow_control parameter of EI_flow_control() may take the value EI_normal_flow_control, or a combination (logical OR) of one or both of the other values:

Note It will be necessary to enable error reporting for many of these events, using ER_enable_reporting(), before they will be reported back to the application software.

Parameters

initial_fcts	
flow_control	the flow control we want

EI_normal_flow_control Operate the SpaceWire link using the standard SpaceWire flow-control algorithm.

EI_transmit_anyway Allow data to be sent regardless of the amount of flow-control credit available.

EI_no_automatic_FCT Do not automatically send any flowcontrol tokens after the link has started.

Returns

0 on success or transmitted , $<$ 0 if error

Exceptions

Exception	I/O error has occurred

8.2.2.14 int EtherSpaceLink.EI_flow_control_401 (int *initial_fcts,* int *flow_control*) throws Exception

8.2.2.15 int EtherSpaceLink.EI_flow_control_408 (int *initial_fcts,* int *flow_control*) throws Exception

8.2.2.16 int EtherSpaceLink.EI_ignore_events (int *what*) throws Exception

Ignore events from the EI module, the EI module will disconnect a link in the event of an error, this function allows the link to ignore errors and continue running in the event of a particular error.

Parameters

what	events to be ignored from EI module

EI_ignore_excess_FCT EI_ignore_excess_data EI_ignore_parity_error EI_ignore_ESC_ESC EI_ignore_ESC_EOP EI_ignore_ESC_EEP EI_ignore_timeout

Returns

0 on success or transmitted , $<$ 0 if error

Exceptions

Exception	I/O error has occurred

8.2.2.17 int EtherSpaceLink.EI_ignore_events_401 (int *what*) throws Exception

8.2.2.18 int EtherSpaceLink.EI_ignore_events_408 (int *what_*) throws Exception

8.2.2.19 boolean EtherSpaceLink.empty_frame (long *sz_*)

8.2.2.20 void EtherSpaceLink.enable_callbacks ( ) [package]

Enable Out of Band data callbacks.

8.2.2.21 int EtherSpaceLink.ER_enable_reporting (int *what*) throws Exception

Enables, or disables, error reporting.

Parameters

what	error reporting we wish to enable

Returns

0 on success or transmitted , $<$ 0 if error

Exceptions

Exception	I/O error has occurred

8.2.2.22 int EtherSpaceLink.ER_enable_reporting_401 (int *what*) throws Exception

8.2.2.23 int EtherSpaceLink.ER_enable_reporting_408 (int *what*) throws Exception

8.2.2.24 boolean EtherSpaceLink.err (double *time_*, int *state_*, int *error_bits_*)

time tag

callback informing the application of an error on the currently active rx link

Parameters

*time_*	the current time
*state_*	the state of the port
*error_bits_*	mask of error bits

8.2.2.25 boolean EtherSpaceLink.error_event ( )

callback informing the application that the currently active port has encountered an error with error waveform data available

8.2.2.26 boolean EtherSpaceLink.esc_eep ( )

callback informing the application the device has seen an escape eep frame

8.2.2.27 boolean EtherSpaceLink.esc_eop ( )

callback informing the application the device has seen an escape eop frame

8.2.2.28 boolean EtherSpaceLink.esc_esc ( )

callback informing the application the device has seen an escape escape frame

8.2.2.29 void EtherSpaceLink.esl_delay (int *ms_*)

8.2.2.30 int EtherSpaceLink.EW_clear (int *port*) throws Exception

This function is called to clear waveform data from the port.

Parameters

port	the port we want to data from

Returns

!0 on error

Exceptions

Exception	I/O error has occurred

8.2.2.31 int EtherSpaceLink.EW_clear_401 (int *port*) throws Exception

8.2.2.32 int EtherSpaceLink.EW_clear_408 (int *port*) throws Exception

8.2.2.33 int EtherSpaceLink.EW_enable_reporting (int *what*) throws Exception

Enables, or disables, waveform capture triggers.

Triggering may be on errors or on other significant events. The parameter what should be set to EW_capture_-nothing to disable all reporting, or to a combination of the.

In addition to the given triggers, a (non-maskable) EVENT in the DSI transmit data stream can also trigger a waveform capture. Each port of a DSI has a waveform capture circuit. Each capture circuit can be triggered by events on its own port, and also on other ports and external events. By default, each capture circuit will respond only to its own port. EW_source() can be used to expand the recognised source of triggers.

Parameters

what	error reporting we wish to enable

Returns

0 on success or transmitted , $<$ 0 if error

Exceptions

Exception	I/O error has occurred

8.2.2.34 int EtherSpaceLink.EW_enable_reporting_401 (int *what*) throws Exception

8.2.2.35 int EtherSpaceLink.EW_enable_reporting_408 (int *what*) throws Exception

8.2.2.36 int EtherSpaceLink.EW_request_data (int *port*) throws Exception

When Error Waveform reporting is switched on, it is possible that the device can indicate that it has an error waveform available via an Extension data block of data. If you want to record this data , you must request it. This

function queues the request and the device will then send the error waveform data which is sent as a SPECIAL block of data.

Parameters

port	the port we want to data from

Returns

!0 on error

Exceptions

Exception	I/O error has occurred

8.2.2.37 int EtherSpaceLink.EW_request_data_401 (int *port*) throws Exception

8.2.2.38 int EtherSpaceLink.EW_request_data_408 (int *port*) throws Exception

8.2.2.39 int EtherSpaceLink.EW_reset (int *port*) throws Exception

Re-arms the capture of error waveforms having previously captured a waveform.

Parameters

port	the port number we want to reset

Returns

0 on success or transmitted , $<$ 0 if error

Exceptions

Exception	I/O error has occurred

8.2.2.40 int EtherSpaceLink.EW_reset_401 (int *port*) throws Exception

8.2.2.41 int EtherSpaceLink.EW_reset_408 (int *port*) throws Exception

8.2.2.42 int EtherSpaceLink.EW_source (int *sources*) throws Exception

Selects waveform capture trigger sources.

Triggering may be on events from ports other than that associated with the capture circuit.

For example, waveforms may be captured on all ports for an event occurring on only one of them.

Parameters

sources	EW_Source_port_1 Trigger on events from port 1. EW_Source_port_2 Trigger on events from port 2. EW_Source_port_3 Trigger on events from port 3. EW_Source_port_4 Trigger on events from port 4. EW_Source_port_5 Trigger on events from port 5. EW_Source_port_6 Trigger on events from port 6. EW_Source_port_7 Trigger on events from port 7. EW_Source_port_8 Trigger on events from port 8.

EW_Source_SMA_12 -LS, -MS∗ platforms Trigger on a rising edge on SMA connectors 1-2. The threshold level is 0.5 V. EW_Source_SMA_34 -LS, -MS platforms Trigger on a rising edge on SMA connectors 3-4. The threshold level is 0.5 V. EW_Source_SMA_56 -LS, -MS platforms Trigger on a rising edge on SMA connectors 5-6. The threshold level is 0.5 V. EW_Source_SMA_78 -LS, -MS platforms Trigger on a rising edge on SMA connectors 7-8. The threshold level is 0.5 V. EW_Source_barrier SO Trigger when the synchronisation barrier is lifted

Returns

0 if the request was successful

Exceptions

Exception	I/O error has occurred

8.2.2.43 int EtherSpaceLink.EW_source_401 (int *sources*) throws Exception

8.2.2.44 int EtherSpaceLink.EW_source_408 (int *sources*) throws Exception

8.2.2.45 boolean EtherSpaceLink.extn_data (byte[] *data_*, int *length_*, boolean *complete_*, int *data_buffer_position_*) throws Exception

8.2.2.46 int EtherSpaceLink.extract_link (byte[] *netbuff*, int *buflen*) [package]

8.2.2.47 int EtherSpaceLink.extract_link_state (byte[] *netbuff*, int *buflen*) [package]

8.2.2.48 double EtherSpaceLink.extract_rx_speed (byte[] *netbuff*, int *buflen*) throws Exception [package]

8.2.2.49 double EtherSpaceLink.extract_timetag (byte[] *netbuff*) throws Exception

Extract timetag from special data callback.

Returns a timetag from a network buffer, the timetag is returned as tenths of nano seconds since device power on or tenths of nano seconds since the beginning of the year if the device is time synchronized to a GPS source.

Bug no checking on buffer length

Parameters

buffer	for the timetag ,the first int field of the buffer may be the module type

Returns

the timetag

Exceptions

Exception	I/O error has occurred

8.2.2.50 double EtherSpaceLink.extract_timetag_401 (byte[] *netbuff*) throws Exception

8.2.2.51 double EtherSpaceLink.extract_timetag_408 (byte[] *netbuff*) throws Exception

8.2.2.52 long EtherSpaceLink.extract_timetag_ns (byte[] *netbuff*) throws Exception

Extract timetag from special data callback data in tenths of nano seconds.

Returns a timetag from a network buffer, the timetag is returned as tenths of nano seconds since device power on or tenths of nano seconds since the beginning of the year if the device is time synchronized to a GPS source.

Bug no checking on buffer length

Parameters

buffer	for the timetag ,the first int field of the buffer may be the module type

Returns

the timetag

Exceptions

Exception	I/O error has occurred

8.2.2.53 long EtherSpaceLink.extract_timetag_ns_401 (byte[] *netbuff*) throws Exception

8.2.2.54 long EtherSpaceLink.extract_timetag_ns_408 (byte[] *netbuff*) throws Exception

8.2.2.55 double EtherSpaceLink.extract_tx_speed (byte[] *netbuff,* int *buflen*) throws Exception [package]

8.2.2.56 int EtherSpaceLink.fastclose ( ) throws Exception

Sets the SO_LINGER timeout to 0 such that when the connection is closed, it is closed quickly.

Parameters

the	link

Returns

0 if successful

Exceptions

Exception	I/O error has occurred

8.2.2.57 int EtherSpaceLink.flow_control (int *initial_fcts,* boolean *no_more_fcts,* boolean *ignore_flow_control*) throws Exception

EI flow_control.

Parameters

initial_fcts	
no_more_fcts	
ignore_flow_-control	

Exceptions

Exception	on an I/O error

Returns

0 if successful

8.2.2.58 int EtherSpaceLink.flush ( ) throws Exception

transmit any buffered data

write_packet may queue data for transport, this function puts queued data onto the wire

Parameters

buffer	the buffer to send
length	the length of the buffer to send
flags	indicating how the data to is be treated EOP the data is to be terminated with an EOP EEP the data is to be terminated with an EEP PART_EOP_EEP the data is not yet terminated INCOMPLETE the data is not yet terminated (but queued in such a way on termination it will be sent in one block)

If you logically OR the flags value with FLUSH a network flush is perfomed and the data is transmitted, if this is not performed data will be only transmitted when the network buffer is full or the flush method is called

Returns

0 if sucessfull, !0 if not, errno setup and error code in handle

Exceptions

Exception	I/O error has occurred

8.2.2.59 int EtherSpaceLink.flush_record_file ( ) throws Exception

Flush record file.

Flush current recording log file, note that this may cause the application to block whilst data is written

Returns

0 if successful

Exceptions

Exception	I/O error has occurred

8.2.2.60 synchronized int EtherSpaceLink.flush_S ( ) throws Exception

Thread safe flush method.

Exceptions

Exception	I/O error

8.2.2.61 Object EtherSpaceLink.get_context ( )

Return user context associated with a EtherSpaceLink handle.

Returns

the context

Exceptions

Exception	I/O error has occurred

8.2.2.62 int EtherSpaceLink.get_control_packet (byte[] *control_buffer,* int *buffer_length,* int *slot*) throws Exception

request control data for slot

Sends a request for a special packet for the given slot

Parameters

control_buffer	where to save data
buffer_length	the size of the buffer
slot	which slot we are requesting

Returns

$<$ 0 an error occurred, otherwise the size

Exceptions

Exception	I/O error has occurred

8.2.2.63 int EtherSpaceLink.get_HWA (byte[] *hwa*) throws Exception

Reads the hardware address (MAC) for the ESL device.

This function reads the MAC address of the device discarding network traffic.

Parameters

HWA	buffer to save the hardware address

Returns

0 if successful, $<$ 0 if not

Exceptions

Exception	I/O error has occurred

8.2.2.64 int EtherSpaceLink.get_HWA_401 (byte[] *hwa*) throws Exception

8.2.2.65 int EtherSpaceLink.get_HWA_408 (byte[] *hwa*) throws Exception

8.2.2.66 String EtherSpaceLink.get_manufacturer_string ( ) throws Exception

Returns manufacturer string.

Returns

the product string

Exceptions

Exception	I/O error has occurred

8.2.2.67 String EtherSpaceLink.get_manufacturer_string_401 ( ) throws Exception

8.2.2.68 String EtherSpaceLink.get_manufacturer_string_408 ( ) throws Exception

8.2.2.69 int EtherSpaceLink.get_module_slot (int *module*) throws Exception

Returns the slot a given module resides in.

Return the slot containing the module wth the given id This function is generally considered legacy as all devices post ESL use memory mapped I/O and the concept of modules is generally no longer applicable

Parameters

module	the id of the module

Returns

0 if the module is present, otherwise the slot

Exceptions

Exception	I/O error has occurred

8.2.2.70 String EtherSpaceLink.get_module_string (int *module*) throws Exception

Returns the name of a given module.

Parameters

module	the id of the module

Returns

0 if the module is not present, otherwise the string of the module

Exceptions

Exception	I/O error has occurred

8.2.2.71 String EtherSpaceLink.get_module_string_401 (int *module*) throws Exception

8.2.2.72 String EtherSpaceLink.get_module_string_408 (int *module*) throws Exception

8.2.2.73 int EtherSpaceLink.get_module_type (byte[] *netbuff*) throws Exception

Return the module type from the given network buffer.

When data is sent from a module in a special packet, we need to know the type of module originated the message so we can further process the message. This function returns the module type.

Parameters

buffer	the buffer to examine

Returns

the module type , $<$ 0 if an error has occurred

Exceptions

Exception	I/O error has occurred

8.2.2.74 int EtherSpaceLink.get_number_of_links ( ) throws Exception

Returns the number of links a device has.

Note: this function can drop inbound traffic

Returns

$<$ 0 on error, number of links otherwise

Exceptions

Exception	I/O error has occurred

8.2.2.75 int EtherSpaceLink.get_number_of_links_401 ( ) throws Exception

8.2.2.76 int EtherSpaceLink.get_number_of_links_408 ( ) throws Exception

8.2.2.77 int EtherSpaceLink.get_options ( ) throws Exception

Returns an integer bitmap indicating the list of options installed on the device.

Returns

int device options bitmap

Exceptions

Exception	I/O error has occurred

8.2.2.78 String EtherSpaceLink.get_options_string ( ) throws Exception

Returns an ASCII string indicating the list of options installed on the device.

This string is dynamically allocated and can be returned by the free call

Returns

char ∗ device options string

Exceptions

Exception	I/O error has occurred

8.2.2.79 String EtherSpaceLink.get_options_string_401 ( ) throws Exception

8.2.2.80 String EtherSpaceLink.get_options_string_408 ( ) throws Exception

8.2.2.81 int EtherSpaceLink.get_packet (byte[] *netbuff,* int *offset_,* int *buffer_length,* int *sda*) throws Exception

read packet not returning packet type.

This function is similar to that of read_packet_full, however the rx_flags parameter is not present and as such can be retrieved by calling get_rx_flags

Parameters

buffer	where to read data into
offset	offset into the above buffer (i.e. data written to buffer+offset)
buffer_length	the number of bytes to read
sda	how to treat special_actions lower 4 bits enumerate to DISCARD_SPECIAL_DATA ignores special data REPORT_SPECIAL_DATA returns special data as -ve return value RETURN_SPECIAL_DATA returns data as normal message CALLBACK_SPECIAL_DATA calls callback upper 4 bits enumberate to DISCARD_EXTENSION_DATA (0) ignores extension data REPORT_EXTENSION_DATA returns extension data as -ve return value RETURN_EXTENSION_DATA returns data as normal message CALLBACK_EXTENSION_DATA calls callback

Returns

$<$ 0 error code

Exceptions

Exception	I/O error has occurred

8.2.2.82 double EtherSpaceLink.get_percent_file_read ( )

returns how much data has been read from the file

Returns

% file opened

Exceptions

Exception	I/O error has occurred

8.2.2.83 String EtherSpaceLink.get_product_string ( ) throws Exception

Returns an ASCII string indicating the product name.

This string is extracted from the Unicode string actually returned by the unit, by extracting the least significant 8-bits of each character.

This string is dynamically allocated and can be returned by the free call

Returns

char $*$ product string

Exceptions

Exception	I/O error has occurred

8.2.2.84 String EtherSpaceLink.get_product_string_401 ( ) throws Exception

8.2.2.85 String EtherSpaceLink.get_product_string_408 ( ) throws Exception

8.2.2.86 int EtherSpaceLink.get_receive_speed ( ) throws Exception

Gets the receive speed of the currently active link.

Note that this function has the ability to cause frames to be dropped and request_rx_speed should be used instead

Returns

< 0 if an error other gets the speed

Exceptions

Exception	I/O error has occurred

8.2.2.87 int EtherSpaceLink.get_receive_speed_401 ( ) throws Exception

8.2.2.88 int EtherSpaceLink.get_receive_speed_408 ( ) throws Exception

8.2.2.89 FileOutputStream EtherSpaceLink.get_record_file ( )

Retreive current recording file.

Returns

handle on recording file

Exceptions

Exception	I/O error has occurred

8.2.2.90 long EtherSpaceLink.get_record_size ( )

Return the amount of data written to the current recording file.

Returns

long the number of bytes written to the file

Exceptions

Exception	I/O error has occurred

8.2.2.91 int EtherSpaceLink.get_rx_flags ( ) throws Exception

return message flags of last packet data read

Sometimes it may be necessary to obtain the message flags outside of the read call

This function returns the value of the flags performed by the last read call.

Exceptions

Exception	I/O error has occurred

8.2.2.92 int EtherSpaceLink.get_rx_timeout ( )

Retreive the current recieve timeout (milliseconds)

Returns

integer, the current receive timeout in milliseconds

Exceptions

Exception	I/O error has occurred

8.2.2.93 int EtherSpaceLink.get_slot (int *slot*) throws Exception

Returns the module in a given slot.

Parameters

slot	number

Returns

id of the module of the slot

Exceptions

Exception	I/O error has occurred

8.2.2.94 int EtherSpaceLink.get_terminator ( ) [package]

8.2.2.95 double EtherSpaceLink.get_timetag_401 (byte[] *netbuff*) throws Exception

8.2.2.96 double EtherSpaceLink.get_timetag_408 (byte[] *netbuff*) throws Exception

8.2.2.97 long EtherSpaceLink.get_total_raw_bytes_received ( ) throws Exception

returns the total number of bytes read.

Returns

long , the total number of bytes recieved

Exceptions

Exception	I/O error has occurred

8.2.2.98 String EtherSpaceLink.get_version ( )

Used to get the version number of the C API being used.

Note the first character denotes the release status of the API if it starts with an 'r' then it as an officially supported version 'norelease' then it is an internal not for release

The macro version returns the version of the interface

Returns

String API version number

Exceptions

Exception	I/O error has occurred

8.2.2.99 boolean EtherSpaceLink.handle_header (byte[ ] *extn_data_buffer*)

8.2.2.100 boolean EtherSpaceLink.handle_module (byte[ ] *extn_data_buffer*) throws Exception

8.2.2.101 boolean EtherSpaceLink.handle_year (byte[ ] *extn_data_buffer*)

8.2.2.102 String EtherSpaceLink.HWA_to_serial_number_string (byte[ ] *hwa*) throws Exception

Converts MAC hardware to human readable string.

Parameters

HWA	buffer to hardware buffer

Returns

char $*$ the product string

Exceptions

Exception	I/O error has occurred

8.2.2.103 int EtherSpaceLink.link_connected ( ) throws Exception

returns if the currently active link is connected

Returns

0 not connected, $<$ 0 if error, 1 connected

Exceptions

Exception	I/O error has occurred

8.2.2.104 int EtherSpaceLink.link_connected_401 ( ) throws Exception

8.2.2.105 int EtherSpaceLink.link_connected_408 ( ) throws Exception

8.2.2.106 void EtherSpaceLink.link_selected (int *link_*)

callback when the rx link has changed

Parameters

*link_*	the newly active rx link

8.2.2.107 boolean EtherSpaceLink.link_timeout ( )

callback informing the application that the currently rx link has timedout (disconnected?)

8.2.2.108 boolean EtherSpaceLink.link_tx_speed (int *link_*, double *txspeed_*)

callback informing the application that a port has changed TX speed

Parameters

*link_*	the link to which this pertains
*txspeed_*	the TX speed

8.2.2.109 void EtherSpaceLink.log_packet (int *direction*, byte[] *buffer*, int *length*, int *flags*) [package]

8.2.2.110 boolean EtherSpaceLink.missing_data ( )

callback informing the application that the device has missed data

8.2.2.111 int EtherSpaceLink.negotiate ( ) throws Exception

8.2.2.112 int EtherSpaceLink.Observe (int *what*) throws Exception

request info from the device

Parameters

address	the address
bytes	the number of bytes to read

Returns

$<$ 0 if an error has been sent

Exceptions

Exception	I/O error has occurred

8.2.2.113 int EtherSpaceLink.Observe_401 (int *what*) throws Exception

8.2.2.114 int EtherSpaceLink.Observe_408 (int *what*) throws Exception

8.2.2.115 int EtherSpaceLink.open (String *ip_address*) throws Exception

opens a connection to the specified device

Opens a connection the the etherspace link device specified which may be resolvable hostname or an ipv4 address.

A port number can be specified by adding a suffix with :portnumber. For example, 1.2.3.4:9999 will connect to a device at IP 1.2.3.4 with port number 9999

It also reads the table of modules installed in the EtherSpaceLink to an internal buffer, for use by procedures accessing status and module information. When opened, the SpaceWire link will be in the disabled state and its default speed will be 10Mb/s. Module and link parameters can be set immediately but the link must be started (using set_mode()) before data can be transferred over the SpaceWire link.

IPV6 is currently not supported by our devices

Note on the first call to this function we set the SIG_PIPE handler to SIG_IGN.

Parameters

address	The address / address:port specifier

Returns

EtherSpaceLink null if there was an error otherwise a EtherSpaceLink Handle

Exceptions

Exception	I/O error has occurred

8.2.2.116 boolean EtherSpaceLink.parity_error ( )

callback informing the application that the currently active link has had a parity error

8.2.2.117 boolean EtherSpaceLink.perror1 ( )

callback informing the application that the currently active port has had an error

8.2.2.118 boolean EtherSpaceLink.perror2 ( )

callback informing the application that the currently active port has had an error

8.2.2.119 boolean EtherSpaceLink.port_status (int *link_*, double *rxspeed_*, boolean *connected_*, int *runstatus_*)

callback informing the application that a port has changed status

Parameters

*link_*	the link to which this pertains
*rxspeed_*	the RX speed
*connected_*	is the link connected
runstatus	the raw status

8.2.2.120 int EtherSpaceLink.ram_io_get_int (int *address*, int *bytes*) throws Exception

8.2.2.121 int EtherSpaceLink.ram_io_write_bytes (int *port*, int *address*, byte[] *netbuff*, int *sz*) throws Exception

8.2.2.122 boolean EtherSpaceLink.raw_link_address (byte[] *data_*, int *length_*) throws Exception

8.2.2.123 boolean EtherSpaceLink.raw_ram_rw (byte[] *data_*, int *length_*, int *data_buffer_position_*) throws Exception

8.2.2.124 boolean EtherSpaceLink.raw_speed_address (byte[] *data_*, int *length_*) throws Exception

8.2.2.125 boolean EtherSpaceLink.raw_waveform_data (byte[] *special_*, int *sz_*)

< Error Code

< Unit

< Error index

8.2.2.126 boolean EtherSpaceLink.received_esc_fct ( )

callback informing the application that an ESC FCT has been received (aka a NULL)

8.2.2.127 boolean EtherSpaceLink.received_fct ( )

callback informing the application that an FCT has been received

8.2.2.128 int EtherSpaceLink.receivePacket (byte[] *data,* long *offset,* int *f*) throws Exception

Receives packet data.

Parameters

data	destination for packet data
offset	offset into the above array
f	flags for reading special data

Returns

the number of bytes read

Exceptions

Exception	I/O error

8.2.2.129 int EtherSpaceLink.record_writes ( )

Returns whether errors writing to record file are treated as errors.

Returns

integer, 0 errors writing to file are ignored, !0 if they are acted on

Exceptions

Exception	I/O error has occurred

8.2.2.130 int EtherSpaceLink.request_link_status ( ) throws Exception

Requests the link status.

Returns

$<$ 0 if an error, otherwise queued for transmission

Exceptions

Exception	I/O error has occurred

8.2.2.131 int EtherSpaceLink.request_link_status_port (int *port*) throws Exception

Requests the link status for a given port On the ESL range of devices this can change the active transmission port.

Parameters

port	which port you want status on

Returns

$<$ 0 if an error, otherwise queued for transmission

Exceptions

Exception	I/O error has occurred

8.2.2.132 int EtherSpaceLink.request_rx_speed ( ) throws Exception

Requests the rx speed.

Returns

$<$ 0 if an error, otherwise queued for transmission

Exceptions

Exception	I/O error has occurred

8.2.2.133 int EtherSpaceLink.request_tx_speed ( ) throws Exception

requests the tx speed for the device

Returns

$<$ 0 if an error, otherwise queued for transmission

Exceptions

Exception	I/O error has occurred

8.2.2.134 int EtherSpaceLink.send (byte[] *buffer,* int *len,* int *flags*) throws Exception

Low level call to send data on ESL transport.

This is a low level call which sends some data on an ESL transport, handling EINTR and EAGAIN. It is really applicable if you are sending raw data to the socket

Parameters

buffer	the data to send
size	the number of bytes to send
flags	transport flags

Returns

number of bytes sent, $<$ 0 if error (-ve error code)

Exceptions

Exception	I/O error has occurred

8.2.2.135 int EtherSpaceLink.send_ESC ( ) throws Exception

8.2.2.136 int EtherSpaceLink.send_ESC_EEP ( ) throws Exception

8.2.2.137 int EtherSpaceLink.send_ESC_EOP ( ) throws Exception

8.2.2.138 int EtherSpaceLink.send_ESC_ESC ( ) throws Exception

8.2.2.139 int EtherSpaceLink.send_ESC_FCT ( ) throws Exception

8.2.2.140 int EtherSpaceLink.send_timecode (byte *tc_*) throws Exception

Sends a timcode packet on the current link.

Parameters

n	the link we want to make active

Returns

$<$ 0 error, 0 success

Exceptions

Exception	I/O error has occurred

8.2.2.141 int EtherSpaceLink.sendPacket (byte[ ] *data,* int *sz,* int *terminator*) throws Exception [package]

Queues a packet of data.

Parameters

data,the	data we want to send
sz	the size of the data we want to send
terminator	EOP, EEP, or PART_EOP_EEP

Returns

the number of bytes sent

Exceptions

Exception	I/O error

8.2.2.142 int EtherSpaceLink.sendPacket (byte[ ] *data,* int *terminator*) throws Exception [package]

Queues a packet of data.

Parameters

data,the	data we want to send
terminator	EOP, EEP, or PART_EOP_EEP

Returns

the number of bytes sent

Exceptions

Exception	I/O error

8.2.2.143 int EtherSpaceLink.sendPacket (byte[] *netbuff,* int *offset,* int *length,* int *flags_*) throws Exception

8.2.2.144 int EtherSpaceLink.sendPacket_S (byte[] *data,* int *sz,* int *terminator*) throws Exception [package]

Queues a packet of data synchronized such that different threads can send data.

Parameters

data,the	data we want to send
sz	the size of the data we want to send
terminator	EOP, EEP, or PART_EOP_EEP

Returns

the number of bytes sent

Exceptions

Exception	I/O error

8.2.2.145 int EtherSpaceLink.sendPacket_S (byte[] *data,* int *terminator*) throws Exception [package]

Queues a packet of data synchronized such that different threads can send data.

Parameters

data,the	data we want to send
terminator	EOP, EEP, or PART_EOP_EEP

Returns

the number of bytes sent

Exceptions

Exception	I/O error

8.2.2.146 synchronized int EtherSpaceLink.sendPacket_S (byte[] *data_buffer,* int *start,* int *length,* int *terminator*) throws Exception

8.2.2.147 int EtherSpaceLink.set_active_link (int *n*) throws Exception

Sets the currently active link.

Parameters

n	the link we want to make active

Returns

< 0 error, 0 success

Exceptions

Exception	I/O error has occurred

8.2.2.148 int EtherSpaceLink.set_active_link_401 (int *n*) throws Exception

8.2.2.149 int EtherSpaceLink.set_active_link_408 (int *n*) throws Exception

8.2.2.150 void EtherSpaceLink.set_context (Object *ctx*)

Set user context can be associated with a EtherSpaceLink handle.

Parameters

context	arbitrary user defined context / structure

Exceptions

Exception	I/O error has occurred

8.2.2.151 void EtherSpaceLink.set_EINTR (int *eintr*)

sets behaviour when network i/o is interrupted

When the API is reading data, it may be using a blocking read call. Normally it soaks up EINTR and EAGAIN calls when reading data, thus if a signal is received the application may not be able to handle any action set up by the signal handler.

Setting the EINTR flag means that an application on a blocking read will return and the error code will be Error_EINTR

Parameters

enable	returning EINTR on read

Exceptions

Exception	I/O error has occurred

8.2.2.152 void EtherSpaceLink.set_extension_cb (ESL_CB *cb*)

extension callback

Parameters

cb	the callback handler

8.2.2.153 int EtherSpaceLink.set_log_file (String *file_name*) throws Exception

Sets the current log file.

Creates a log file which will contain a record of data transmitted and received. Data from this file can be later analysed and/or replayed.

If there is an extant log file in then it closed.

if the file name is not null then it is used for logging

Entries in the log file will begin with the text: Rx@p for data received on port p Tx@p for data transmitted on port p

Alternatively, if a SpaceWire link id has been set for a particular EtherSpaceLink unit:

Rx@id@p for data received on port p Tx@id@p for data transmitted on port p

Parameters

file_name	the name of the logging file

Returns

int, 0 if successful , non zero if not

Exceptions

Exception	I/O error has occurred

8.2.2.154 int EtherSpaceLink.set_max_packet_data (int *N*) throws Exception

Control the data receive compressor - discard data from packet.

Parameters

link	the connection to the hardware device
N	the number of bytes to be forwarded N = 0 to 254 allow N bytes to be forwarded N = 255 allow all bytes to be forwarded return 0 if sucessful and !0 in error situation

Exceptions

Exception	I/O error has occurred

8.2.2.155 int EtherSpaceLink.set_max_packet_data_401 (int *N*) throws Exception

8.2.2.156 int EtherSpaceLink.set_max_packet_data_408 (int *N*) throws Exception

8.2.2.157 int EtherSpaceLink.set_mode (int *mode*) throws Exception

set mode of current link

Sets the operating mode of the currently-active SpaceWire link.

After opening a connection, the link is disabled; it must then be enabled into one of its operational modes before data can be transferred.

One of the three modes LINK_mode_disabled, LINK_mode_normal and LINK_mode_legacy should be chosen.

The use of LINK_mode_fixed_speed to set some DSI ports to 10Mb/s, together with the conventional set_speed() mechanism is the only way to run a DSIs links at two different speeds

Parameters

mode	of operation

LINK_mode_disabled

```
 The link is idle and silent.
```

LINK_mode_normal

```
Start the link by actively trying to establish contact.
```

LINK_mode_legacy

```
Dont start until activity on the link is seen. Use with SMCS/TSS901 devices.
```

LINK_mode_long_timeout

```
    Extends the timeout period in the link state  machine to provide a potentialy more reliable link
    start at very low data rates (i.e. for slow (lowpower) links near to 2Mb/s). It is necessary to set
    the link speed with an   set_speed() API call before calling set_mode()  with this extra
    LINK_mode_slow_speed.


 LINK_mode_fixed_speed

     The link speed remains at its default startup speed
     (10Mb/s nominal; actually within the range 9.8 to 10.2Mb/s)

 LINK_mode_slow_speed

     This setting combines the long_timeout and fixed_speed modifiers, thereby also setting the
     initial link speed to the final operating speed.
```

Returns

0 if the request queued , !0 if not

Exceptions

Exception	I/O error has occurred

8.2.2.158 int EtherSpaceLink.set_mode_401 (int *mode*) throws Exception

8.2.2.159 int EtherSpaceLink.set_mode_408 (int *mode*) throws Exception

8.2.2.160 int EtherSpaceLink.set_mode_portmask (int *mode_*, int *ports_*) throws Exception

set mode of list of links

Sets the operating mode of a given set of links

After opening a connection, the link is disabled; it must then be enabled into one of its operational modes before data can be transferred.

The use of LINK_mode_fixed_speed to set some DSI ports to 10Mb/s, together with the conventional set_speed() mechanism is the only way to run a DSIs links at two different speeds. The active port is the highest number listed port in the mask

Parameters

mode	of operation

Returns

0 if the request queued , !0 if not

Exceptions

Exception	I/O error has occurred

8.2.2.161 int EtherSpaceLink.set_record_file (String *file_name*) throws Exception

Sets the recording file.

Used to set the recording file if there is a recording file in operation it will be closed

Parameters

file_name	the name of the logging file

Returns

int, 0 if successful , non zero if not

Exceptions

Exception	I/O error has occurred

8.2.2.162 int EtherSpaceLink.set_rx_timeout (int *to*) throws Exception

Sets the rx timeout in milliseconds.

Sets the maximum period that an read_packet command will wait for data before returning. When the timeout expires, the read_packet command will return with as much data as it has received, if any. This is the maximum period of waiting after the last received data, not a delay from issuing the read_packet command.

Set the current recieve timeout (milliseconds)

Parameters

to	the new timeout

Exceptions

Exception	I/O error has occurred

8.2.2.163 void EtherSpaceLink.set_rx_timeout_action (int *action*) throws Exception

sets the behaviour on a networktimeout

Network errors are treated as a generic errors, but there are circumstances where one wants to know that an explicit timeout has occurred.

Parameters

returns_error	0 timeouts are returned as generic error !0 timeout errors are identified

Exceptions

Exception	I/O error has occurred

8.2.2.164 void EtherSpaceLink.set_slot (int *index_*, int *type_*) throws Exception

Sets a particular slot to a given slot type (used internally)

Parameters

slot	no
system	type

Exceptions

Exception	I/O error has occurred

8.2.2.165 void EtherSpaceLink.set_special_cb (ESL_CB *cb*)

Sets special callback handler.

Parameters

cb	the callback handler

8.2.2.166 int EtherSpaceLink.set_speed (int *speed*) throws Exception

Sets the transmit speed of the link

Sets the transmit speed of all of the SpaceWire links on thisEtherSpaceLink unit.

Links set with the additional mode modifier LINK_mode_fixed_speed, which remain at their start-up speed of 10-Mb/s.

Parameters

speed	the number of megabits per second

Returns

0 if request has been put on the wire, !0 if error

Exceptions

Exception	I/O error has occurred

8.2.2.167 int EtherSpaceLink.set_speed_401 (int *speed*) throws Exception

8.2.2.168 int EtherSpaceLink.set_speed_408 (int *speed*) throws Exception

8.2.2.169 int EtherSpaceLink.set_speed_double (double *speed*) throws Exception

Sets the transmit speed of the link allowing partial Mb speeds.

Sets the transmit speed of all of the SpaceWire links on this EtherSpaceLink unit.

The links on an EtherSpaceLink unit can be set to a range of speeds; see the platform description for details.

Notice that the link speed in set_speed_double() is treated in units of bits/s, if the value is greater than 1 000 000, and in units of Mb/s otherwise. This is unlike set_speed, which uses Mb/s only.

All links on each EtherSpaceLink unit that are set using set_mode() to LINK_mode_normal run at the same speed. The only exception to this are links set with the additional mode modifier LINK_mode_fixed_speed, which remain at their start-up speed of 10Mb/s.

Parameters

speed	the number of megabits per second

Returns

0 if request has been put on the wire, !0 if error

Exceptions

Exception	I/O error has occurred

8.2.2.170 int EtherSpaceLink.set_speed_double_401 (int *speed*) throws Exception

8.2.2.171 int EtherSpaceLink.set_speed_double_408 (int *speed*) throws Exception

8.2.2.172 int EtherSpaceLink.set_tx_record_file (String *file_name*) throws Exception

Sets the tx recording file.

Used to set the recording file for recording transmitted space wire data

Parameters

file_name	the name of the logging file

Returns

int, 0 if successful , non zero if not

Exceptions

Exception	I/O error has occurred

8.2.2.173 int EtherSpaceLink.shutdown ( ) throws Exception

Used to terminate a connection to an ESL device.

Generally when things have gone seriously wrong try to reset the TCP.IP networking to the device.

Returns

null to dereference the connection

Exceptions

Exception	I/O error has occurred

8.2.2.174 int EtherSpaceLink.sma_56_pulse_width (int *width*) throws Exception

Parameters

width	the pulse width for the SMA connector

It is possible to connect an SMA connector This sets the pulse width of the device

Returns

< 0 indicating an error

Exceptions

Exception	I/O error has occurred

8.2.2.175 int EtherSpaceLink.sma_56_pulse_width_401 (int *width*) throws Exception

8.2.2.176 int EtherSpaceLink.sma_56_pulse_width_408 (int *width*) throws Exception

8.2.2.177 int EtherSpaceLink.SO_BARRIER ( ) throws Exception

8.2.2.178 int EtherSpaceLink.SO_FORWARD ( ) throws Exception

8.2.2.179 int EtherSpaceLink.SO_JOIN ( ) throws Exception

8.2.2.180 int EtherSpaceLink.SO_STORE ( ) throws Exception

8.2.2.181 static String EtherSpaceLink.source (int *w*) [static],[package]

8.2.2.182 boolean EtherSpaceLink.special_data (byte[] *data_*, int *length_*, boolean *complete_*, int *data_buffer_position_*) throws Exception [package]

8.2.2.183 boolean EtherSpaceLink.start_time (int *year*, int *month*, int *day*, int *hour*, int *minute*, int *second*, long *ns*)

callback informing the application of the time the device started recording

Parameters

year	
month	
day	
minute	
second	
ns	

8.2.2.184 void EtherSpaceLink.system_type (int *syst_*) throws Exception

Forces system type such that platform derived timetags can be read.

Parameters

system	type

Exceptions

Exception	I/O error has occurred

8.2.2.185 boolean EtherSpaceLink.timecode (byte *tc_*, byte *a_*, byte *b_*, byte *t_*)

callback informing the application of the timecode

Parameters

time	code the timecode

8.2.2.186 boolean EtherSpaceLink.timetag (long *time_*)

time tag

callback informing the application of the currently active rx time

Parameters

*time_*	the current time, the number of 1/10ths of nano seconds since the beginning of the year if synchronised with GPS otherwise the number of 1/10ths since power on.

8.2.2.187 boolean EtherSpaceLink.timetag_uncertainty (int *uncertainty_*)

callback informing the application of the time tag uncertainty

Parameters

*uncertainty_*	the current uncertainty

8.2.2.188 boolean EtherSpaceLink.timezero (long *time_*)

callback informing the application of the first packet received

Parameters

*time_*	the current time, the number of 1/10ths of nano seconds since the beginning of the year if synchronised with GPS otherwise the number of 1/10ths since power on.

8.2.2.189 int EtherSpaceLink.TT_enable_reporting (int *when*) throws Exception

Enable timetags for currently active link.

Parameters

when	what events generate a timetag

Returns

0 on success or transmitted , $<$ 0 if error

Exceptions

Exception	I/O error has occurred

8.2.2.190 int EtherSpaceLink.TT_enable_reporting_401 (int *when*) throws Exception

8.2.2.191 int EtherSpaceLink.TT_enable_reporting_408 (int *when*) throws Exception

8.2.2.192 boolean EtherSpaceLink.unknown_extn_data (byte[] *data_*, int *length_*, boolean *complete_*, int *data_buffer_position_*)

callback informing the application of unknown extension data

Parameters

*data_*	the data from ram i/o module
*length_*	the length of the data
*complete_*	is the data complete (if not more to come)
*data_buffer_position_*	the current position in the stream of data

8.2.2.193 boolean EtherSpaceLink.unknown_ram_data (byte[] *data_*, int *length_*, boolean *complete_*, int *data_buffer_position_*)

callback informing the application of data from an unknown module (should never trigger)

Parameters

*data_*	the data from ram i/o module
*length_*	the length of the data
*complete_*	is the data complete (if not more to come)
*data_buffer_position_*	the current position in the stream of data

8.2.2.194 boolean EtherSpaceLink.unknown_special_data (byte[] *data_*, int *length_*, boolean *complete_*, int *data_buffer_position_*)

callback informing the application of unknown special data

Parameters

*data_*	the data from ram i/o module
*length_*	the length of the data
*complete_*	is the data complete (if not more to come)
*data_buffer_position_*	the current position in the stream of data

8.2.2.195 static int EtherSpaceLink.update (int *original*, String *s*) [static],[package]

8.2.2.196 static int EtherSpaceLink.update_es (int *original*, String *s*) [static],[package]

8.2.2.197 boolean EtherSpaceLink.waveform_data (int *unit*, long *time_*, int *erridx*, int *port*, int *ew_port*, int *ew_event*, String *fullcause*, byte[] *special_*, int *sz*)

8.2.2.198 static String EtherSpaceLink.what (int *w*) [static],[package]

8.2.2.199 int EtherSpaceLink.write_buffer_empty ( ) throws Exception

Returns if the write buffer for the currently active link is empty.

Returns

1 if empty, 0 if not and $<$ 0 in an error situation

Exceptions

Exception	I/O error has occurred

8.2.2.200 int EtherSpaceLink.write_buffer_empty_401 ( ) throws Exception

8.2.2.201 int EtherSpaceLink.write_buffer_empty_408 ( ) throws Exception

8.2.2.202 int EtherSpaceLink.write_EXTN (int *extn*) throws Exception

Sends an extension packet.

Sends a single extension data character via the EtherSpaceLink unit to the currently-selected SpaceWire link

Parameters

EXTN	id of the extension to write EOP End of Packet EEP Error End of Packet STORE Store flag - part of a store-and-forward sequence FORWARD Forward flag - part of a store-and-forward sequence port_select to port_select + 15 Select this SpaceWire port for data transmission; subsequent data transmitted will be sent to port n, where port_select = n = port_select + 15 ESC EI Escape - follow with another character (e.g. ESC + data is a time-code) FCT EI Flow-control token ESC_EOP EI Escape and end-of-packet (ECSS error). ESC_EEP EI Escape and error-end-of-packet (ECSS error). ESC_ESC EI Escape and escape (ECSS error). ESC_FCT EI Escape and flow-control token (null). ParityError EI Invert the parity bit (and cause an ECSS error). EVENT EW Insert a flag - generate an event. JOIN SO Join a synchronisation RESIGN SO Leave a synchronisation BARRIER SO Synchronisation point DELAY to DELAY + 15 EI,SO Insert delays: values 0 to 15 represent 1 to 16 * 4-bits additional time a [(1 to 16) * 4 + 1 bits total] HOLD CO Hold the following characters until a low TimeCode EI To send an ECSS time-code, a second byte is required for the time-code valu

Returns

0 if the packet has been queued, !0 if an error occurred queuing the extension

Exceptions

Exception	I/O error has occurred

8.2.2.203 int EtherSpaceLink.write_packet (byte[] *netbuff,* int *length,* int *flags*) throws Exception

queue data for transmission

Queues message for transmission, if there is no room left in the buffer, the buffer is transmitted. Note, that even the queued data is transmitted the data added to it may not be. If you want to guarantee transmission of this data you need to call flush.

Parameters

buffer	the data to send
length	the size of the buffer to size
flags	additional metadata about the frame we are transmitting

EOP

```
This is the last part, or all, of a data packet; an end-ofpacket  (EOP) is added.
```

EEP

```
This is the last part, or all, of a data packet; an error endof packet (EEP) is added.
```

PART_EOP_EEP

```
This is part of a data packet; no end-of-packet is added. This effectively allows one to send
part packet data, do not rely on this working correctly with other devices as it is not part
of the spacewire specification.
```

EXTN

```
This is a complete extension character sequence. Extension packets have a maximum length of 60 bytes.
```

EherSpaceLink_SPECIAL

```
This is a complete special packet
```

EEP would not normally be used to terminate a packet but is available here to assist with testing where an erroneous packet may usefully be generated. Data is queued in buffers in the API in order to make best use of the TCP/IP stream and may not be sent immediately. flush() should be used to ensure the immediate transmission of any buffered data. The one-character extension sequences may be sent using write_EXTN.

Hardware dsi

Returns

0 if sucessfull, <0 if not (-error number), errno setup and error code in handle

Exceptions

Exception	I/O error has occurred

8.2.3 Member Data Documentation

8.2.3.1 final int EtherSpaceLink.ATI_address = 0x0100 [static]

8.2.3.2 final int EtherSpaceLink.ATOM = 0x10E [static]

8.2.3.3 final int EtherSpaceLink.BARRIER = 0x111 [static]

8.2.3.4 final int EtherSpaceLink.barrier = 22 [static]

8.2.3.5 final int EtherSpaceLink.CALLBACK_EXTENSION_DATA = 0x30 [static]

8.2.3.6 final int EtherSpaceLink.CALLBACK_SPECIAL_DATA = 0x03 [static]

8.2.3.7 final int EtherSpaceLink.CAPABILITIES = 0 [static]

8.2.3.8 final int EtherSpaceLink.CONNECT_FILE = (1) [static]

8.2.3.9 final int EtherSpaceLink.CR = 14 [static]

8.2.3.10 final int EtherSpaceLink.Delay = 0x130 [static]

8.2.3.11 final int EtherSpaceLink.DESCRIPTION_address = 0x880B [static]

8.2.3.12 final int EtherSpaceLink.DISCARD_EXTENSION_DATA = 0x00 [static]

8.2.3.13 final int EtherSpaceLink.DISCARD_SPECIAL_DATA = 0x00 [static]

8.2.3.14 final int EtherSpaceLink.EEP = 0x101 [static]

Error End of Packet.

8.2.3.15 final int EtherSpaceLink.EI_ignore_ESC_EEP = report_ESC_EEP [static]

8.2.3.16 final int EtherSpaceLink.EI_ignore_ESC_EOP = report_ESC_EOP [static]

8.2.3.17 final int EtherSpaceLink.EI_ignore_ESC_ESC = report_ESC_ESC [static]

8.2.3.18 final int EtherSpaceLink.EI_ignore_excess_data = report_excess_data [static]

8.2.3.19 final int EtherSpaceLink.EI_ignore_excess_FCT = report_excess_FCT [static]

8.2.3.20 final int EtherSpaceLink.EI_ignore_parity_error = report_parity_error [static]

8.2.3.21 final int EtherSpaceLink.EI_ignore_timeout = report_timeout [static]

8.2.3.22 final int EtherSpaceLink.EI_no_automatic_FCT = 0x10 [static]

8.2.3.23 final int EtherSpaceLink.EI_normal_flow_control = 0x00 [static]

8.2.3.24 final int EtherSpaceLink.EI_transmit_anyway = 0x20 [static]

8.2.3.25 final int EtherSpaceLink.EOP = 0x102 [static]

End of Packet.

8.2.3.26 final int EtherSpaceLink.ER = 8 [static]

8.2.3.27 final int EtherSpaceLink.ER_64 = 16 [static]

8.2.3.28 final int EtherSpaceLink.ER_report_ESC_EEP = report_ESC_EEP [static]

report Escape Error End of Packet

8.2.3.29 final int EtherSpaceLink.ER_report_ESC_EOP = report_ESC_EOP [static]

report Escape End of Packet

8.2.3.30 final int EtherSpaceLink.ER_report_ESC_ESC = report_ESC_ESC [static]

report Escape Escape

8.2.3.31 final int EtherSpaceLink.ER_report_fct = report_FCT [static]

report FCT

8.2.3.32 final int EtherSpaceLink.ER_report_first_fct = 0x04 [static]

Error report first fct.

8.2.3.33 final int EtherSpaceLink.ER_report_first_null = 0x02 [static]

Error report first null.

8.2.3.34 final int EtherSpaceLink.ER_report_nchar = 0x40 [static]

8.2.3.35 final int EtherSpaceLink.ER_report_nothing = 0x00 [static]

Error reporting report nothing.

8.2.3.36 final int EtherSpaceLink.ER_report_null = report_NULL [static]

report null

8.2.3.37 final int EtherSpaceLink.ER_report_parity_error = report_parity_error [static]

report parity error

8.2.3.38 final int EtherSpaceLink.ER_report_running_error = (0x08 | report_parity_error | report_ESC_EOP | report_ESC_EEP | report_ESC_ESC | report_timeout) [static]

Error report running.

8.2.3.39 final int EtherSpaceLink.ER_report_starting_error = 0x10 [static]

8.2.3.40 final int EtherSpaceLink.ER_report_time_code = (0x80 | report_time_code) [static]

report time code

8.2.3.41 final int EtherSpaceLink.ER_report_timeout = report_timeout [static]

report Timeout

8.2.3.42 final int EtherSpaceLink.Error_Buffer_Full = -51 [static]

can't do non blocking write as buffer is full

8.2.3.43 final int EtherSpaceLink.Error_Callback_Return = -39 [static]

Callback has asked for a return.

8.2.3.44 final int EtherSpaceLink.Error_CaptureThread_Failed = -52 [static]

Capture thread failed.

8.2.3.45 final int EtherSpaceLink.Error_EINTR = -27 [static]

EINTR occurred.

8.2.3.46 final int EtherSpaceLink.Error_EXE_Start_Failed = -43 [static]

Cannot start executable.

8.2.3.47 final int EtherSpaceLink.Error_File_Move = -37 [static]

Unable to move file into place.

8.2.3.48 final int EtherSpaceLink.Error_File_Not_Present = -26 [static]

Requested file is not present.

8.2.3.49 final int EtherSpaceLink.Error_FileList_Empty = -40 [static]

List of files given is empty.

8.2.3.50 final int EtherSpaceLink.Error_Function_Not_Supported = -17 [static]

Device does not support the requested function.

8.2.3.51 final int EtherSpaceLink.Error_Host_Unresolvable = -31 [static]

Unable to resolve host.

8.2.3.52 final int EtherSpaceLink.Error_Host_Unresponsive = -32 [static]

Unable to connect to host.

8.2.3.53 final int EtherSpaceLink.Error_Incorrect_Device = -29 [static]

Connecting to a device which does not support functionality.

8.2.3.54 final int EtherSpaceLink.Error_Invalid_Device = -36 [static]

Device is not supported by API.

8.2.3.55 final int EtherSpaceLink.Error_Invalid_File = -38 [static]

Unable to open file.

8.2.3.56 final int EtherSpaceLink.Error_Invalid_Link = -45 [static]

Invalid Link selected.

8.2.3.57 final int EtherSpaceLink.Error_IO_Error = -12 [static]

we have an IO error

8.2.3.58 final int EtherSpaceLink.Error_Link_Incorrect = -28 [static]

Link number is incorrect.

8.2.3.59 final int EtherSpaceLink.Error_Link_Not_Connected = -49 [static]

Link Not Connected.

8.2.3.60 final int EtherSpaceLink.Error_LogFile_Open = -3 [static]

Couldn't open logging file.

8.2.3.61 final int EtherSpaceLink.Error_LogFile_Write = -4 [static]

log_file write failed

8.2.3.62 final int EtherSpaceLink.Error_Memory = -30 [static]

Unable to allocate memory.

8.2.3.63 final int EtherSpaceLink.Error_Module_Not_Present = -24 [static]

Module not present.

8.2.3.64 final int EtherSpaceLink.Error_Network = -18 [static]

Error reading / writing to/from the device.

8.2.3.65 final int EtherSpaceLink.Error_Network_Format_Error = -19 [static]

Error understanding recieved packet.

8.2.3.66 final int EtherSpaceLink.Error_NO_Connection = -44 [static]

Link Not estblished.

8.2.3.67 final int EtherSpaceLink.Error_Not_Known = -42 [static]

API returned 0 as an error should (should not happen)

8.2.3.68 final int EtherSpaceLink.Error_Parameter_RangeIncorrect = -25 [static]

Parameter not in range.

8.2.3.69 final int EtherSpaceLink.Error_ReadHandler_Running = -50 [static]

There is a read handler running for this connection.

8.2.3.70 final int EtherSpaceLink.Error_Receiver_Shutdown = -11 [static]

peer has performed an orderly shutdown

8.2.3.71 final int EtherSpaceLink.Error_Receiver_Timeout = -10 [static]

we have a network timeout timeout

8.2.3.72 final int EtherSpaceLink.Error_RecFile_Open = -1 [static]

Couldn't open recording file.

8.2.3.73 final int EtherSpaceLink.Error_RecFile_Write = -2 [static]

record_file write failed

8.2.3.74 final int EtherSpaceLink.Error_Request_Too_Large = -20 [static]

The I/O request can't be fullfilled by the hardware.

8.2.3.75 final int EtherSpaceLink.Error_Response_Mismatch = -23 [static]

Response does not match I/O request.

8.2.3.76 final int EtherSpaceLink.Error_Response_Too_Small = -22 [static]

Response from the device didn't contain enough data.

8.2.3.77 final int EtherSpaceLink.Error_SaveBuf_Overflow_Restore = -16 [static]

Restoring the read_packet_full() save_buffer failed.

8.2.3.78 final int EtherSpaceLink.Error_SaveBuf_Overflow_Save = -15 [static]

Saving the read_packet_full() save_buffer failed.

8.2.3.79 final int EtherSpaceLink.Error_Sequence_Error = -21 [static]

Didn't receive expected notification from the hardware.

8.2.3.80 final int EtherSpaceLink.Error_Set_Option_File = -35 [static]

Asked to set an option when playing back from file.

8.2.3.81 final int EtherSpaceLink.Error_Unknown_System_Type = -41 [static]

Unknown type.

8.2.3.82 final int EtherSpaceLink.Error_WaveForm_Dir_Create = -33 [static]

Unable to create waveform directory.

8.2.3.83 final int EtherSpaceLink.Error_Would_Block = -48 [static]

I/O call would block.

8.2.3.84 final int EtherSpaceLink.Error_Zero_Read = -34 [static]

asked to read zero bytes

8.2.3.85 final int EtherSpaceLink.ESC = 0x103 [static]

Escape.

8.2.3.86 final int EtherSpaceLink.ESC_EEP = 0x105 [static]

Escape End of Packet.

8.2.3.87 final int EtherSpaceLink.ESC_EOP = 0x106 [static]

Escape Error of packet.

8.2.3.88 final int EtherSpaceLink.ESC_ESC = 0x107 [static]

Escape Escape.

8.2.3.89 final int EtherSpaceLink.ESC_FCT = 0x104 [static]

Escape FCT aka a NULL character.

8.2.3.90 final int EtherSpaceLink.EVENT = 0x113 [static]

8.2.3.91 final int EtherSpaceLink.Event_cause_address = 0x0060 [static]

8.2.3.92 final int EtherSpaceLink.EW = 9 [static]

8.2.3.93 final int EtherSpaceLink.EW_address = 0x1000 [static]

8.2.3.94 final int EtherSpaceLink.EW_capture_EEP = report_EEP [static]

trigger on Error End of Packet

8.2.3.95 final int EtherSpaceLink.EW_capture_EOP = report_EOP [static]

trigger on End of Packet

8.2.3.96 final int EtherSpaceLink.EW_capture_ESC_EEP = report_ESC_EEP [static]

trigger on Escape Error End of Packet

8.2.3.97 final int EtherSpaceLink.EW_capture_ESC_EOP = report_ESC_EOP [static]

trigger on Escape End of Packet

8.2.3.98 final int EtherSpaceLink.EW_capture_ESC_ESC = report_ESC_ESC [static]

trigger on Escape Escape

8.2.3.99 final int EtherSpaceLink.EW_capture_excess_data = report_excess_data [static]

trigger on excess data

8.2.3.100 final int EtherSpaceLink.EW_capture_excess_FCT = report_excess_FCT [static]

trigger on excess fct

8.2.3.101 final int EtherSpaceLink.EW_capture_FCT = report_FCT [static]

trigger on FCT

8.2.3.102 final int EtherSpaceLink.EW_capture_first_fct = 0x04 [static]

trigger on first fct

8.2.3.103 final int EtherSpaceLink.EW_capture_first_null = (0x02 | report_first_null) [static]

trigger on first null

8.2.3.104 final int EtherSpaceLink.EW_capture_nchar = (0x40 | report_nchar) [static]

trigger on n char

8.2.3.105 final int EtherSpaceLink.EW_capture_nothing = 0x00 [static]

8.2.3.106 final int EtherSpaceLink.EW_capture_null = report_NULL [static]

trigger on NULL

8.2.3.107 final int EtherSpaceLink.EW_capture_parity_error = report_parity_error [static]

trigger on parity error

8.2.3.108 final int EtherSpaceLink.EW_capture_running_error = (0x08 | report_parity_error | report_ESC_EOP | report_ESC_EEP | report_ESC_ESC | report_timeout) [static]

trigger on run error

8.2.3.109 final int EtherSpaceLink.EW_capture_starting_error = 0x10 [static]

trigger on start error

8.2.3.110 final int EtherSpaceLink.EW_capture_time_code = (0x80 | report_time_code) [static]

trigger on timecode

8.2.3.111 final int EtherSpaceLink.EW_capture_timeout = report_timeout [static]

trigger on timeout

8.2.3.112 final int EtherSpaceLink.EW_RT = 13 [static]

8.2.3.113 final int EtherSpaceLink.EW_source_address = 0x0070 [static]

8.2.3.114 final int EtherSpaceLink.EW_Source_barrier = 0x0001 [static]

Barrier.

8.2.3.115 final int EtherSpaceLink.EW_Source_local_clock = 0x8000 [static]

Local clock.

8.2.3.116 final int EtherSpaceLink.EW_Source_port_1 = 0x0002 [static]

Port 1.

8.2.3.117 final int EtherSpaceLink.EW_Source_port_2 = 0x0004 [static]

Port 2.

8.2.3.118 final int EtherSpaceLink.EW_Source_port_3 = 0x0008 [static]

Port 3.

8.2.3.119 final int EtherSpaceLink.EW_Source_port_4 = 0x0010 [static]

Port 4.

8.2.3.120 final int EtherSpaceLink.EW_Source_port_5 = 0x0020 [static]

Port 5.

8.2.3.121 final int EtherSpaceLink.EW_Source_port_6 = 0x0040 [static]

Port 6.

8.2.3.122 final int EtherSpaceLink.EW_Source_port_7 = 0x0080 [static]

Port 7.

8.2.3.123 final int EtherSpaceLink.EW_Source_port_8 = 0x0100 [static]

Port 8.

8.2.3.124 final int EtherSpaceLink.EW_Source_SMA_12 = 0x0200 [static]

SMA 1/2 changing state.

8.2.3.125 final int EtherSpaceLink.EW_Source_SMA_34 = 0x0400 [static]

SMA 3/4 changing state.

8.2.3.126 final int EtherSpaceLink.EW_Source_SMA_56 = 0x0800 [static]

SMA 5/6 changing state.

8.2.3.127 final int EtherSpaceLink.EW_Source_SMA_78 = 0x1000 [static]

SMA 7/8 changing state.

8.2.3.128 final int EtherSpaceLink.EXTENSION_DATA_FLAGS = 0x30 [static]

8.2.3.129 final int EtherSpaceLink.EXTENSION_SIZE = 1010 [static]

Returning the amount of extension data.

8.2.3.130 final int EtherSpaceLink.EXTN = 1005 [static]

We are sending an extension frame.

8.2.3.131 final int EtherSpaceLink.FCT = 0x100 [static]

8.2.3.132 final int EtherSpaceLink.FLOW_CONTROL_address = 0x0050 [static]

8.2.3.133 final int EtherSpaceLink.FLUSH = 2048 [static]

8.2.3.134 final int EtherSpaceLink.FORWARD = 0x10D [static]

8.2.3.135 final int EtherSpaceLink.Header = 0x1CE [static]

Capture Header containing version and time information.

8.2.3.136 final int EtherSpaceLink.HOLD = 0x12F [static]

8.2.3.137 final int EtherSpaceLink.HWA = 3 [static]

8.2.3.138 final int EtherSpaceLink.HWA_address = 0x8800 [static]

8.2.3.139 final int EtherSpaceLink.IGNORE_address = 0x0040 [static]

8.2.3.140 final int EtherSpaceLink.INCOMPLETE = 1008 [static]

Used to build up a single packet for the unit.

8.2.3.141 final int EtherSpaceLink.JOIN = 0x110 [static]

8.2.3.142 final int EtherSpaceLink.LINK = 5 [static]

8.2.3.143 final int EtherSpaceLink.LINK_address = 0x0000 [static]

8.2.3.144 final int EtherSpaceLink.LINK_mode_disabled = 0x01 [static]

Disables the link.

8.2.3.145 final int EtherSpaceLink.LINK_mode_fixed_speed = 0x80 [static]

8.2.3.146 final int EtherSpaceLink.LINK_mode_legacy = 0x04 [static]

IEEE 1355 (spacewire precursor)

8.2.3.147 final int EtherSpaceLink.LINK_mode_long_timeout = 0x40 [static]

8.2.3.148 final int EtherSpaceLink.LINK_mode_master = 0x06 [static]

IEEE 1355 (precurosr)

8.2.3.149 final int EtherSpaceLink.LINK_mode_normal = 0x02 [static]

Enables the link.

8.2.3.150 final int EtherSpaceLink.LINK_mode_slow_speed = 0xC0 [static]

8.2.3.151 final int EtherSpaceLink.LINK_SPEED = 4 [static]

8.2.3.152 final int EtherSpaceLink.LINK_state_offset = 4 [static]

8.2.3.153 final int EtherSpaceLink.LINK_tx_buffer_empty = 0x08 [static]

8.2.3.154 final int EtherSpaceLink.MANUFACTURER = 1 [static]

8.2.3.155 final int EtherSpaceLink.max_packet_data = 0x0010 [static]

8.2.3.156 final int EtherSpaceLink.Missing_data = 0x114 [static]

Missed data message.

8.2.3.157 final int EtherSpaceLink.Module = 0x192 [static]

Module data.

8.2.3.158 final int EtherSpaceLink.MOTA = 0x10F [static]

8.2.3.159 final int EtherSpaceLink.MSR_state_Connected = 7 [static]

8.2.3.160 final int EtherSpaceLink.MSR_state_NC = 6 [static]

8.2.3.161 final int EtherSpaceLink.Multi_byte_extn_start = 0x180 [static]

8.2.3.162 final int EtherSpaceLink.NLINKS_address = 0x8FFF [static]

8.2.3.163 final int EtherSpaceLink.OBSERVE_address = 0x0020 [static]

8.2.3.164 final int EtherSpaceLink.Option_SO = 1 [static]

Option SO module is installed.

8.2.3.165 final int EtherSpaceLink.OPTIONS_address = 0x8F60 [static]

8.2.3.166 final int EtherSpaceLink.ParityError = 0x109 [static]

Parity Error message.

8.2.3.167 final int EtherSpaceLink.PART_EOP_EEP = 1000 [static]

Error packet.

8.2.3.168 final int EtherSpaceLink.PART_EXTN = 1006 [static]

8.2.3.169 final int EtherSpaceLink.PART_SPECIAL = 1004 [static]

A part of a special frame.

8.2.3.170 final int EtherSpaceLink.PC_address = 0x2000 [static]

8.2.3.171 final int EtherSpaceLink.PERROR1 = 0x10A [static]

Error 1 message.

8.2.3.172 final int EtherSpaceLink.PERROR2 = 0x10B [static]

Error 2 message.

8.2.3.173 final int EtherSpaceLink.PG_address = 0x4000 [static]

8.2.3.174 final int EtherSpaceLink.PortSelect = 0x140 [static]

Port select message.

8.2.3.175 final int EtherSpaceLink.PortSelect_max = 0x17F [static]

Max port select message.

8.2.3.176 final int EtherSpaceLink.PRODUCT = 2 [static]

8.2.3.177 final int EtherSpaceLink.ram_rw = 21 [static]

8.2.3.178 long EtherSpaceLink.rawtime [protected]

8.2.3.179 final int EtherSpaceLink.READ_IMMEDIATE = 0x40 [static]

8.2.3.180 final int EtherSpaceLink.Receiver_Timeout_Returns_Error = 1 [static]

8.2.3.181 final int EtherSpaceLink.Receiver_Timeout_Returns_Zero_Part_Pkt = 0 [static]

8.2.3.182 final int EtherSpaceLink.REPEAT_1 = 0x1B1 [static]

8.2.3.183 final int EtherSpaceLink.REPEAT_2 = 0x1B2 [static]

8.2.3.184 final int EtherSpaceLink.REPEAT_3 = 0x1B3 [static]

8.2.3.185 final int EtherSpaceLink.report_delta = 0x400000 [static]

8.2.3.186 final int EtherSpaceLink.report_EEP = 0x800000 [static]

EEP error event.

8.2.3.187 final int EtherSpaceLink.report_EOP = 0x010000 [static]

EOP recieved.

8.2.3.188 final int EtherSpaceLink.report_ESC_EEP = 0x000400 [static]

Escape EEP Error.

8.2.3.189 final int EtherSpaceLink.report_ESC_EOP = 0x000800 [static]

Escape EOP error.

8.2.3.190 final int EtherSpaceLink.report_ESC_ESC = 0x000200 [static]

Escape Escape Error.

8.2.3.191 final int EtherSpaceLink.report_excess_data = 0x080000 [static]

Too much data sent for # of FCT's.

8.2.3.192 final int EtherSpaceLink.report_excess_FCT = 0x100000 [static]

Too many FCTS event.

8.2.3.193 final int EtherSpaceLink.REPORT_EXTENSION_DATA = 0x10 [static]

8.2.3.194 final int EtherSpaceLink.report_FCT = 0x004000 [static]

FCT received.

8.2.3.195 final int EtherSpaceLink.report_first_byte = 0x040000 [static]

First byte of packet.

8.2.3.196 final int EtherSpaceLink.report_first_null = 0x200000 [static]

First null event.

8.2.3.197 final int EtherSpaceLink.report_mid_bytes = 0x020000 [static]

Frame mide byte.

8.2.3.198 final int EtherSpaceLink.report_nchar = 0x400000 [static]

character received event

8.2.3.199 final int EtherSpaceLink.report_NULL = 0x002000 [static]

Null received.

8.2.3.200 final int EtherSpaceLink.report_parity_error = 0x001000 [static]

Parity Error.

8.2.3.201 final int EtherSpaceLink.REPORT_SPECIAL_DATA = 0x01 [static]

8.2.3.202 final int EtherSpaceLink.report_time_code = 0x008000 [static]

Time code received.

8.2.3.203 final int EtherSpaceLink.report_timeout = 0x000100 [static]

Link Timeout.

8.2.3.204 final int EtherSpaceLink.RESIGN = 0x112 [static]

8.2.3.205 final int EtherSpaceLink.RETURN_EXTENSION_DATA = 0x20 [static]

8.2.3.206 final int EtherSpaceLink.RETURN_SPECIAL_DATA = 0x02 [static]

8.2.3.207 final int EtherSpaceLink.router_cs = 18 [static]

8.2.3.208 final int EtherSpaceLink.router_stats = 20 [static]

8.2.3.209 final int EtherSpaceLink.router_tables = 19 [static]

8.2.3.210 int EtherSpaceLink.rx_link

8.2.3.211 final int EtherSpaceLink.RX_SPEED_address = 0x0001 [static]

8.2.3.212 final int EtherSpaceLink.SF = 6 [static]

8.2.3.213 final int EtherSpaceLink.SF_disabled = 0x00 [static]

8.2.3.214 final int EtherSpaceLink.SF_enabled = 0x80 [static]

8.2.3.215 final int EtherSpaceLink.SMA_56_pulse_width_address = 0x00F0 [static]

8.2.3.216 final int EtherSpaceLink.SpaceWire_state_Connecting = 4 [static]

8.2.3.217 final int EtherSpaceLink.SpaceWire_state_ErrorReset = 0 [static]

8.2.3.218 final int EtherSpaceLink.SpaceWire_state_ErrorWait = 1 [static]

8.2.3.219 final int EtherSpaceLink.SpaceWire_state_Ready = 2 [static]

8.2.3.220 final int EtherSpaceLink.SpaceWire_state_Run = 5 [static]

8.2.3.221 final int EtherSpaceLink.SpaceWire_state_Started = 3 [static]

8.2.3.222 final int EtherSpaceLink.SPECIAL = 1003 [static]

We are sending a special frame.

8.2.3.223 final int EtherSpaceLink.SPECIAL_DATA_FLAGS = 0x03 [static]

8.2.3.224 final int EtherSpaceLink.SPECIAL_SIZE = 1009 [static]

Returning the amount of special data.

8.2.3.225 final int EtherSpaceLink.STORE = 0x10C [static]

8.2.3.226 final int EtherSpaceLink.SYSTEM_TYPE_401 = 1 [static]

8.2.3.227 final int EtherSpaceLink.SYSTEM_TYPE_408 = 2 [static]

8.2.3.228 final int EtherSpaceLink.SYSTEM_TYPE_INVALID = 0 [static]

8.2.3.229 final int EtherSpaceLink.TC_rx = 10 [static]

8.2.3.230 final int EtherSpaceLink.TC_rx_64 = 17 [static]

8.2.3.231 final int EtherSpaceLink.TC_rx_report_enabled = 0x08 [static]

8.2.3.232 final int EtherSpaceLink.TC_rx_silent = 0x00 [static]

8.2.3.233 final int EtherSpaceLink.TC_rx_time_stamp_enabled = 0x40 [static]

8.2.3.234 final int EtherSpaceLink.TC_tx = 11 [static]

8.2.3.235 final int EtherSpaceLink.TC_tx_external_trigger = 0x02 [static]

8.2.3.236 final int EtherSpaceLink.TC_tx_format_mask = 0x30 [static]

8.2.3.237 final int EtherSpaceLink.TC_tx_increment_6_bits = 0x10 [static]

8.2.3.238 final int EtherSpaceLink.TC_tx_increment_7_bits = 0x20 [static]

8.2.3.239 final int EtherSpaceLink.TC_tx_increment_8_bits = 0x30 [static]

8.2.3.240 final int EtherSpaceLink.TC_tx_no_increment = 0x00 [static]

8.2.3.241 final int EtherSpaceLink.TC_tx_no_trigger = 0x00 [static]

8.2.3.242 final int EtherSpaceLink.TC_tx_one_code = 0x01 [static]

8.2.3.243 final int EtherSpaceLink.TC_tx_regular_trigger = 0x03 [static]

8.2.3.244 final int EtherSpaceLink.TC_tx_report_transmission = 0x40 [static]

8.2.3.245 final int EtherSpaceLink.TC_tx_trigger_mask = 0x03 [static]

8.2.3.246 final int EtherSpaceLink.TC_tx_update_code = 0x08 [static]

8.2.3.247 final int EtherSpaceLink.TC_tx_update_interval = 0x04 [static]

8.2.3.248 final int EtherSpaceLink.TimeCode = 0x191 [static]

Spacewire timecode.

8.2.3.249 final int EtherSpaceLink.Timeout = 0x108 [static]

Timeout message.

8.2.3.250 final int EtherSpaceLink.TimeTag = 0x188 [static]

Timetag message.

8.2.3.251 final int EtherSpaceLink.TIMETAG_address = 0x0030 [static]

8.2.3.252 final int EtherSpaceLink.TimeTag_delta = 0x182 [static]

Timetag delta message.

8.2.3.253 final int EtherSpaceLink.TimeTag_uncertainty = 0x181 [static]

Timetag uncertain message.

8.2.3.254 final int EtherSpaceLink.TimeZero = 0x198 [static]

First timecode on the link.

8.2.3.255 final int EtherSpaceLink.TRUNCATE_1 = 0x1A1 [static]

8.2.3.256 final int EtherSpaceLink.TRUNCATE_2 = 0x1A2 [static]

8.2.3.257 final int EtherSpaceLink.TRUNCATED = 1007 [static]

Aritifical construct for unhandled data.

8.2.3.258 final int EtherSpaceLink.TT = 7 [static]

8.2.3.259 final int EtherSpaceLink.TT_64 = 15 [static]

8.2.3.260 final int EtherSpaceLink.TT_now = 23 [static]

8.2.3.261 final int EtherSpaceLink.TT_report_EEP = (0x04 | report_EEP) [static]

Timetag report Error End of Packet.

8.2.3.262 final int EtherSpaceLink.TT_report_EOP = (0x04 | report_EOP) [static]

Timetag report End of Packet.

8.2.3.263 final int EtherSpaceLink.TT_report_EOP_EEP = (0x04 | report_EEP | report_EOP) [static]

Timetag end of packet markers.

8.2.3.264 final int EtherSpaceLink.TT_report_ESC_EEP = report_ESC_EEP [static]

Timetag report ESC Error End of Packet.

8.2.3.265 final int EtherSpaceLink.TT_report_ESC_EOP = report_ESC_EOP [static]

Timetag report ESC End of Packet.

8.2.3.266 final int EtherSpaceLink.TT_report_ESC_ESC = report_ESC_ESC [static]

Timetag report ESC ESC.

8.2.3.267 final int EtherSpaceLink.TT_report_fct = report_FCT [static]

Timetag report FCT.

8.2.3.268 final int EtherSpaceLink.TT_report_first_byte = (0x01 | report_first_byte) [static]

Timetag first byte of packet.

8.2.3.269 final int EtherSpaceLink.TT_report_intermediate_bytes = (0x02 | report_mid_bytes) [static]

Timetag middle byte.

8.2.3.270 final int EtherSpaceLink.TT_report_nothing = 0x00 [static]

Report Nothing.

8.2.3.271 final int EtherSpaceLink.TT_report_null = report_NULL [static]

Timetag report NULL.

8.2.3.272 final int EtherSpaceLink.TT_report_parity_error = report_parity_error [static]

Timetag report parity error.

8.2.3.273 final int EtherSpaceLink.TT_report_time_code = report_time_code [static]

Timetag report spacewire timecode.

8.2.3.274 final int EtherSpaceLink.TT_report_timeout = report_timeout [static]

Timetag report timeout.

8.2.3.275 final int EtherSpaceLink.TX_SPEED_address = 0x87FD [static]

8.2.3.276 final String EtherSpaceLink.version = "ESL_RELID" [static]

8.2.3.277 final int EtherSpaceLink.VERSION_address = 0x880A [static]

8.2.3.278 final int EtherSpaceLink.Year = 0x1C8 [static]

Capture start date/time.

The documentation for this class was generated from the following file:

- /autogen/EtherSpaceLink.java

Chapter 9

File Documentation

9.1 /autogen/EtherSpaceLink.java File Reference

Classes

- class EtherSpaceLink
- class EtherSpaceLink.Error

Index

/autogen/EtherSpaceLink.java, 127

ATI_address
    EtherSpaceLink, 108
    Memory Mapped Addresses, 42
ATI_calibrate
    EtherSpaceLink, 73
ATI_calibrate_401
    EtherSpaceLink, 73
ATI_calibrate_408
    EtherSpaceLink, 73
ATOM
    EtherSpaceLink, 108
    Extension codes, 30
abort
    EtherSpaceLink, 73

BARRIER
    EtherSpaceLink, 109
    Extension codes, 30
barrier
    EtherSpaceLink, 109

CAPABILITIES
    EtherSpaceLink, 109
CONNECT_FILE
    EtherSpaceLink, 109
CR
    EtherSpaceLink, 109
check_record_writes
    EtherSpaceLink, 73
close
    EtherSpaceLink, 73
connected
    EtherSpaceLink, 74
Connection, 13
    device_type, 13
    open, 13

DESCRIPTION_address
    EtherSpaceLink, 109
    Memory Mapped Addresses, 42
decode_fp16
    EtherSpaceLink, 74
Delay
    EtherSpaceLink, 109
    Extension codes, 30
delay
    EtherSpaceLink, 74
device_type
    Connection, 13
    EtherSpaceLink, 74
    functions, 50
dump_max
    EtherSpaceLink, 74

EEP
    EtherSpaceLink, 109
    Extension codes, 30
EI_flow_control
    EtherSpaceLink, 74
EI_flow_control_401
    EtherSpaceLink, 75
EI_flow_control_408
    EtherSpaceLink, 75
EI_ignore_ESC_EEP
    EtherSpaceLink, 109
EI_ignore_ESC_EOP
    EtherSpaceLink, 109
EI_ignore_ESC_ESC
    EtherSpaceLink, 109
EI_ignore_events
    EtherSpaceLink, 75
EI_ignore_events_401
    EtherSpaceLink, 75
EI_ignore_events_408
    EtherSpaceLink, 75
EI_ignore_excess_FCT
    EtherSpaceLink, 109
EI_ignore_excess_data
    EtherSpaceLink, 109
EI_ignore_parity_error
    EtherSpaceLink, 109
EI_ignore_timeout
    EtherSpaceLink, 109
EI_no_automatic_FCT
    EtherSpaceLink, 109
EI_normal_flow_control
    EtherSpaceLink, 109
EI_transmit_anyway
    EtherSpaceLink, 109
EOP
    EtherSpaceLink, 109
    Extension codes, 30
ER
    Error mask fields, 35
    EtherSpaceLink, 109
ER_64
    Error mask fields, 35
    EtherSpaceLink, 109

ER_enable_reporting
Error Reporting, 26
EtherSpaceLink, 76
ER_enable_reporting_401
EtherSpaceLink, 76
ER_enable_reporting_408
EtherSpaceLink, 76
ER_report_ESC_EEP
Error mask fields, 35
EtherSpaceLink, 109
ER_report_ESC_EOP
Error mask fields, 35
EtherSpaceLink, 110
ER_report_ESC_ESC
Error mask fields, 35
EtherSpaceLink, 110
ER_report_fct
Error mask fields, 36
EtherSpaceLink, 110
ER_report_first_fct
Error mask fields, 36
EtherSpaceLink, 110
ER_report_first_null
Error mask fields, 36
EtherSpaceLink, 110
ER_report_nchar
Error mask fields, 36
EtherSpaceLink, 110
ER_report_nothing
Error mask fields, 36
EtherSpaceLink, 110
ER_report_null
Error mask fields, 36
EtherSpaceLink, 110
ER_report_parity_error
Error mask fields, 36
EtherSpaceLink, 110
ER_report_running_error
Error mask fields, 36
EtherSpaceLink, 110
ER_report_starting_error
Error mask fields, 36
EtherSpaceLink, 110
ER_report_time_code
Error mask fields, 36
EtherSpaceLink, 110
ER_report_timeout
Error mask fields, 36
EtherSpaceLink, 110
ESC
EtherSpaceLink, 114
Extension codes, 30
ESC_EEP
EtherSpaceLink, 114
Extension codes, 30
ESC_EOP
EtherSpaceLink, 114
Extension codes, 30
ESC_ESC
EtherSpaceLink, 114
Extension codes, 30
ESC_FCT
EtherSpaceLink, 114
Extension codes, 30
EVENT
EtherSpaceLink, 115
Extension codes, 30
EW
Error Waveform Triggers, 37
EtherSpaceLink, 115
EW_RT
Error Waveform Triggers, 39
EtherSpaceLink, 116
EW_Source_SMA_12
Error Waveform Sources, 41
EtherSpaceLink, 117
EW_Source_SMA_34
Error Waveform Sources, 41
EtherSpaceLink, 117
EW_Source_SMA_56
Error Waveform Sources, 41
EtherSpaceLink, 117
EW_Source_SMA_78
Error Waveform Sources, 41
EtherSpaceLink, 117
EW_Source_barrier
Error Waveform Sources, 40
EtherSpaceLink, 116
EW_Source_local_clock
Error Waveform Sources, 40
EtherSpaceLink, 116
EW_Source_port_1
Error Waveform Sources, 40
EtherSpaceLink, 116
EW_Source_port_2
Error Waveform Sources, 40
EtherSpaceLink, 116
EW_Source_port_3
Error Waveform Sources, 41
EtherSpaceLink, 116
EW_Source_port_4
Error Waveform Sources, 41
EtherSpaceLink, 117
EW_Source_port_5
Error Waveform Sources, 41
EtherSpaceLink, 117
EW_Source_port_6
Error Waveform Sources, 41
EtherSpaceLink, 117
EW_Source_port_7
Error Waveform Sources, 41
EtherSpaceLink, 117
EW_Source_port_8
Error Waveform Sources, 41
EtherSpaceLink, 117
EW_address

EtherSpaceLink, 115
Memory Mapped Addresses, 42
EW_capture_EEP
Error Waveform Triggers, 37
EtherSpaceLink, 115
EW_capture_EOP
Error Waveform Triggers, 37
EtherSpaceLink, 115
EW_capture_ESC_EEP
Error Waveform Triggers, 38
EtherSpaceLink, 115
EW_capture_ESC_EOP
Error Waveform Triggers, 38
EtherSpaceLink, 115
EW_capture_ESC_ESC
Error Waveform Triggers, 38
EtherSpaceLink, 115
EW_capture_FCT
Error Waveform Triggers, 38
EtherSpaceLink, 115
EW_capture_excess_FCT
Error Waveform Triggers, 38
EtherSpaceLink, 115
EW_capture_excess_data
Error Waveform Triggers, 38
EtherSpaceLink, 115
EW_capture_first_fct
Error Waveform Triggers, 38
EtherSpaceLink, 115
EW_capture_first_null
Error Waveform Triggers, 38
EtherSpaceLink, 115
EW_capture_nchar
Error Waveform Triggers, 38
EtherSpaceLink, 115
EW_capture_nothing
Error Waveform Triggers, 38
EtherSpaceLink, 116
EW_capture_null
Error Waveform Triggers, 38
EtherSpaceLink, 116
EW_capture_parity_error
Error Waveform Triggers, 38
EtherSpaceLink, 116
EW_capture_running_error
Error Waveform Triggers, 38
EtherSpaceLink, 116
EW_capture_starting_error
Error Waveform Triggers, 39
EtherSpaceLink, 116
EW_capture_time_code
Error Waveform Triggers, 39
EtherSpaceLink, 116
EW_capture_timeout
Error Waveform Triggers, 39
EtherSpaceLink, 116
EW_clear
EtherSpaceLink, 77
EW_clear_401
EtherSpaceLink, 77
EW_clear_408
EtherSpaceLink, 77
EW_enable_reporting
Error Waveforms, 27
EtherSpaceLink, 77
EW_enable_reporting_401
EtherSpaceLink, 77
EW_enable_reporting_408
EtherSpaceLink, 77
EW_request_data
EtherSpaceLink, 77
EW_request_data_401
EtherSpaceLink, 79
EW_request_data_408
EtherSpaceLink, 79
EW_reset
EtherSpaceLink, 79
EW_reset_401
EtherSpaceLink, 79
EW_reset_408
EtherSpaceLink, 79
EW_source
Error Waveforms, 27
EtherSpaceLink, 79
EW_source_401
EtherSpaceLink, 80
EW_source_408
EtherSpaceLink, 80
EW_source_address
EtherSpaceLink, 116
Memory Mapped Addresses, 42
EXTENSION_SIZE
EtherSpaceLink, 117
EXTN
EtherSpaceLink, 117
empty_frame
EtherSpaceLink, 76
enable_callbacks
EtherSpaceLink, 76
err
EtherSpaceLink, 76
functions, 50
Error
EtherSpaceLink::Error, 57
Error Codes, 44
Error_Buffer_Full, 45
Error_Callback_Return, 45
Error_CaptureThread_Failed, 46
Error_EINTR, 46
Error_EXE_Start_Failed, 46
Error_File_Move, 46
Error_File_Not_Present, 46
Error_FileList_Empty, 46
Error_Function_Not_Supported, 46
Error_Host_Unresolvable, 46
Error_Host_Unresponsive, 46

Error_IO_Error, 47
Error_Incorrect_Device, 46
Error_Invalid_Device, 46
Error_Invalid_File, 46
Error_Invalid_Link, 47
Error_Link_Incorrect, 47
Error_Link_Not_Connected, 47
Error_LogFile_Open, 47
Error_LogFile_Write, 47
Error_Memory, 47
Error_Module_Not_Present, 47
Error_NO_Connection, 47
Error_Network, 47
Error_Network_Format_Error, 47
Error_Not_Known, 47
Error_Parameter_RangeIncorrect, 48
Error_ReadHandler_Running, 48
Error_RecFile_Open, 48
Error_RecFile_Write, 48
Error_Receiver_Shutdown, 48
Error_Receiver_Timeout, 48
Error_Request_Too_Large, 48
Error_Response_Mismatch, 48
Error_Response_Too_Small, 48
Error_SaveBuf_Overflow_Restore, 48
Error_SaveBuf_Overflow_Save, 48
Error_Sequence_Error, 48
Error_Set_Option_File, 49
Error_Unknown_System_Type, 49
Error_WaveForm_Dir_Create, 49
Error_Would_Block, 49
Error_Zero_Read, 49
Option_SO, 49
Error mask fields, 35
ER, 35
ER_64, 35
ER_report_ESC_EEP, 35
ER_report_ESC_EOP, 35
ER_report_ESC_ESC, 35
ER_report_fct, 36
ER_report_first_fct, 36
ER_report_first_null, 36
ER_report_nchar, 36
ER_report_nothing, 36
ER_report_null, 36
ER_report_parity_error, 36
ER_report_running_error, 36
ER_report_starting_error, 36
ER_report_time_code, 36
ER_report_timeout, 36
Error Reporting, 26
ER_enable_reporting, 26
Error Waveform Sources, 40
EW_Source_SMA_12, 41
EW_Source_SMA_34, 41
EW_Source_SMA_56, 41
EW_Source_SMA_78, 41
EW_Source_barrier, 40
EW_Source_local_clock, 40
EW_Source_port_1, 40
EW_Source_port_2, 40
EW_Source_port_3, 41
EW_Source_port_4, 41
EW_Source_port_5, 41
EW_Source_port_6, 41
EW_Source_port_7, 41
EW_Source_port_8, 41
Error Waveform Triggers, 37
EW, 37
EW_RT, 39
EW_capture_EEP, 37
EW_capture_EOP, 37
EW_capture_ESC_EEP, 38
EW_capture_ESC_EOP, 38
EW_capture_ESC_ESC, 38
EW_capture_FCT, 38
EW_capture_excess_FCT, 38
EW_capture_excess_data, 38
EW_capture_first_fct, 38
EW_capture_first_null, 38
EW_capture_nchar, 38
EW_capture_nothing, 38
EW_capture_null, 38
EW_capture_parity_error, 38
EW_capture_running_error, 38
EW_capture_starting_error, 39
EW_capture_time_code, 39
EW_capture_timeout, 39
Error Waveforms, 27
EW_enable_reporting, 27
EW_source, 27
Error_Buffer_Full
Error Codes, 45
EtherSpaceLink, 111
Error_Callback_Return
Error Codes, 45
EtherSpaceLink, 111
Error_CaptureThread_Failed
Error Codes, 46
EtherSpaceLink, 111
Error_EINTR
Error Codes, 46
EtherSpaceLink, 111
Error_EXE_Start_Failed
Error Codes, 46
EtherSpaceLink, 111
Error_File_Move
Error Codes, 46
EtherSpaceLink, 111
Error_File_Not_Present
Error Codes, 46
EtherSpaceLink, 111
Error_FileList_Empty
Error Codes, 46
EtherSpaceLink, 111
Error_Function_Not_Supported

Error Codes, 46
EtherSpaceLink, 111
Error_Host_Unresolvable
Error Codes, 46
EtherSpaceLink, 111
Error_Host_Unresponsive
Error Codes, 46
EtherSpaceLink, 111
Error_IO_Error
Error Codes, 47
EtherSpaceLink, 112
Error_Incorrect_Device
Error Codes, 46
EtherSpaceLink, 111
Error_Invalid_Device
Error Codes, 46
EtherSpaceLink, 112
Error_Invalid_File
Error Codes, 46
EtherSpaceLink, 112
Error_Invalid_Link
Error Codes, 47
EtherSpaceLink, 112
Error_Link_Incorrect
Error Codes, 47
EtherSpaceLink, 112
Error_Link_Not_Connected
Error Codes, 47
EtherSpaceLink, 112
Error_LogFile_Open
Error Codes, 47
EtherSpaceLink, 112
Error_LogFile_Write
Error Codes, 47
EtherSpaceLink, 112
Error_Memory
Error Codes, 47
EtherSpaceLink, 112
Error_Module_Not_Present
Error Codes, 47
EtherSpaceLink, 112
Error_NO_Connection
Error Codes, 47
EtherSpaceLink, 113
Error_Network
Error Codes, 47
EtherSpaceLink, 112
Error_Network_Format_Error
Error Codes, 47
EtherSpaceLink, 112
Error_Not_Known
Error Codes, 47
EtherSpaceLink, 113
Error_Parameter_RangeIncorrect
Error Codes, 48
EtherSpaceLink, 113
Error_ReadHandler_Running
Error Codes, 48
EtherSpaceLink, 113
Error_RecFile_Open
Error Codes, 48
EtherSpaceLink, 113
Error_RecFile_Write
Error Codes, 48
EtherSpaceLink, 113
Error_Receiver_Shutdown
Error Codes, 48
EtherSpaceLink, 113
Error_Receiver_Timeout
Error Codes, 48
EtherSpaceLink, 113
Error_Request_Too_Large
Error Codes, 48
EtherSpaceLink, 113
Error_Response_Mismatch
Error Codes, 48
EtherSpaceLink, 113
Error_Response_Too_Small
Error Codes, 48
EtherSpaceLink, 113
Error_SaveBuf_Overflow_Restore
Error Codes, 48
EtherSpaceLink, 113
Error_SaveBuf_Overflow_Save
Error Codes, 48
EtherSpaceLink, 114
Error_Sequence_Error
Error Codes, 48
EtherSpaceLink, 114
Error_Set_Option_File
Error Codes, 49
EtherSpaceLink, 114
Error_Unknown_System_Type
Error Codes, 49
EtherSpaceLink, 114
Error_WaveForm_Dir_Create
Error Codes, 49
EtherSpaceLink, 114
Error_Would_Block
Error Codes, 49
EtherSpaceLink, 114
Error_Zero_Read
Error Codes, 49
EtherSpaceLink, 114
error_event
EtherSpaceLink, 76
functions, 52
esc_eep
EtherSpaceLink, 76
functions, 52
esc_eop
EtherSpaceLink, 76
functions, 52
esc_esc
EtherSpaceLink, 76
functions, 52

esl_delay
EtherSpaceLink, 77
EtherSpaceLink, 57
ATI_address, 108
ATI_calibrate, 73
ATI_calibrate_401, 73
ATI_calibrate_408, 73
ATOM, 108
abort, 73
BARRIER, 109
barrier, 109
CAPABILITIES, 109
CONNECT_FILE, 109
CR, 109
check_record_writes, 73
close, 73
connected, 74
DESCRIPTION_address, 109
decode_fp16, 74
Delay, 109
delay, 74
device_type, 74
dump_max, 74
EEP, 109
EI_flow_control, 74
EI_flow_control_401, 75
EI_flow_control_408, 75
EI_ignore_ESC_EEP, 109
EI_ignore_ESC_EOP, 109
EI_ignore_ESC_ESC, 109
EI_ignore_events, 75
EI_ignore_events_401, 75
EI_ignore_events_408, 75
EI_ignore_excess_FCT, 109
EI_ignore_excess_data, 109
EI_ignore_parity_error, 109
EI_ignore_timeout, 109
EI_no_automatic_FCT, 109
EI_normal_flow_control, 109
EI_transmit_anyway, 109
EOP, 109
ER, 109
ER_64, 109
ER_enable_reporting, 76
ER_enable_reporting_401, 76
ER_enable_reporting_408, 76
ER_report_ESC_EEP, 109
ER_report_ESC_EOP, 110
ER_report_ESC_ESC, 110
ER_report_fct, 110
ER_report_first_fct, 110
ER_report_first_null, 110
ER_report_nchar, 110
ER_report_nothing, 110
ER_report_null, 110
ER_report_parity_error, 110
ER_report_running_error, 110
ER_report_starting_error, 110
ER_report_time_code, 110
ER_report_timeout, 110
ESC, 114
ESC_EEP, 114
ESC_EOP, 114
ESC_ESC, 114
ESC_FCT, 114
EVENT, 115
EW, 115
EW_RT, 116
EW_Source_SMA_12, 117
EW_Source_SMA_34, 117
EW_Source_SMA_56, 117
EW_Source_SMA_78, 117
EW_Source_barrier, 116
EW_Source_local_clock, 116
EW_Source_port_1, 116
EW_Source_port_2, 116
EW_Source_port_3, 116
EW_Source_port_4, 117
EW_Source_port_5, 117
EW_Source_port_6, 117
EW_Source_port_7, 117
EW_Source_port_8, 117
EW_address, 115
EW_capture_EEP, 115
EW_capture_EOP, 115
EW_capture_ESC_EEP, 115
EW_capture_ESC_EOP, 115
EW_capture_ESC_ESC, 115
EW_capture_FCT, 115
EW_capture_excess_FCT, 115
EW_capture_excess_data, 115
EW_capture_first_fct, 115
EW_capture_first_null, 115
EW_capture_nchar, 115
EW_capture_nothing, 116
EW_capture_null, 116
EW_capture_parity_error, 116
EW_capture_running_error, 116
EW_capture_starting_error, 116
EW_capture_time_code, 116
EW_capture_timeout, 116
EW_clear, 77
EW_clear_401, 77
EW_clear_408, 77
EW_enable_reporting, 77
EW_enable_reporting_401, 77
EW_enable_reporting_408, 77
EW_request_data, 77
EW_request_data_401, 79
EW_request_data_408, 79
EW_reset, 79
EW_reset_401, 79
EW_reset_408, 79
EW_source, 79
EW_source_401, 80
EW_source_408, 80

EW_source_address, 116
EXTENSION_SIZE, 117
EXTN, 117
empty_frame, 76
enable_callbacks, 76
err, 76
Error_Buffer_Full, 111
Error_Callback_Return, 111
Error_CaptureThread_Failed, 111
Error_EINTR, 111
Error_EXE_Start_Failed, 111
Error_File_Move, 111
Error_File_Not_Present, 111
Error_FileList_Empty, 111
Error_Function_Not_Supported, 111
Error_Host_Unresolvable, 111
Error_Host_Unresponsive, 111
Error_IO_Error, 112
Error_Incorrect_Device, 111
Error_Invalid_Device, 112
Error_Invalid_File, 112
Error_Invalid_Link, 112
Error_Link_Incorrect, 112
Error_Link_Not_Connected, 112
Error_LogFile_Open, 112
Error_LogFile_Write, 112
Error_Memory, 112
Error_Module_Not_Present, 112
Error_NO_Connection, 113
Error_Network, 112
Error_Network_Format_Error, 112
Error_Not_Known, 113
Error_Parameter_RangeIncorrect, 113
Error_ReadHandler_Running, 113
Error_RecFile_Open, 113
Error_RecFile_Write, 113
Error_Receiver_Shutdown, 113
Error_Receiver_Timeout, 113
Error_Request_Too_Large, 113
Error_Response_Mismatch, 113
Error_Response_Too_Small, 113
Error_SaveBuf_Overflow_Restore, 113
Error_SaveBuf_Overflow_Save, 114
Error_Sequence_Error, 114
Error_Set_Option_File, 114
Error_Unknown_System_Type, 114
Error_WaveForm_Dir_Create, 114
Error_Would_Block, 114
Error_Zero_Read, 114
error_event, 76
esc_eep, 76
esc_eop, 76
esc_esc, 76
esl_delay, 77
EtherSpaceLink, 72
EtherSpaceLink, 72
Event_cause_address, 115
extn_data, 80
extract_link, 80
extract_link_state, 80
extract_rx_speed, 80
extract_timetag, 80
extract_timetag_401, 80
extract_timetag_408, 80
extract_timetag_ns, 80
extract_timetag_ns_401, 81
extract_timetag_ns_408, 81
extract_tx_speed, 81
FCT, 117
FLOW_CONTROL_address, 118
FLUSH, 118
FORWARD, 118
fastclose, 81
flow_control, 81
flush, 81
flush_S, 83
flush_record_file, 83
get_HWA, 84
get_HWA_401, 84
get_HWA_408, 84
get_context, 83
get_control_packet, 83
get_manufacturer_string, 84
get_manufacturer_string_401, 84
get_manufacturer_string_408, 84
get_module_slot, 84
get_module_string, 85
get_module_string_401, 85
get_module_string_408, 85
get_module_type, 85
get_number_of_links, 85
get_number_of_links_401, 86
get_number_of_links_408, 86
get_options, 86
get_options_string, 86
get_options_string_401, 86
get_options_string_408, 86
get_packet, 86
get_percent_file_read, 87
get_product_string, 87
get_product_string_401, 88
get_product_string_408, 88
get_receive_speed, 88
get_receive_speed_401, 88
get_receive_speed_408, 88
get_record_file, 88
get_record_size, 88
get_rx_flags, 88
get_rx_timeout, 89
get_slot, 89
get_terminator, 89
get_timetag_401, 89
get_timetag_408, 89
get_total_raw_bytes_received, 89
get_version, 89
HOLD, 118

HWA, 118
HWA_address, 118
HWA_to_serial_number_string, 90
handle_header, 90
handle_module, 90
handle_year, 90
Header, 118
IGNORE_address, 118
INCOMPLETE, 118
JOIN, 118
LINK, 118
LINK_SPEED, 119
LINK_address, 118
LINK_mode_disabled, 118
LINK_mode_fixed_speed, 118
LINK_mode_legacy, 118
LINK_mode_long_timeout, 118
LINK_mode_master, 118
LINK_mode_normal, 118
LINK_mode_slow_speed, 118
LINK_state_offset, 119
LINK_tx_buffer_empty, 119
link_connected, 90
link_connected_401, 90
link_connected_408, 90
link_selected, 90
link_timeout, 90
link_tx_speed, 90
log_packet, 91
MANUFACTURER, 119
MOTA, 119
MSR_state_Connected, 119
MSR_state_NC, 119
max_packet_data, 119
Missing_data, 119
missing_data, 91
Module, 119
Multi_byte_extn_start, 119
NLINKS_address, 119
negotiate, 91
OBSERVE_address, 119
OPTIONS_address, 119
Observe, 91
Observe_401, 91
Observe_408, 91
open, 91
Option_SO, 119
PART_EOP_EEP, 119
PART_EXTN, 119
PART_SPECIAL, 119
PC_address, 119
PERROR1, 120
PERROR2, 120
PG_address, 120
PRODUCT, 120
parity_error, 92
ParityError, 119
perror1, 92
perror2, 92
port_status, 92
PortSelect, 120
PortSelect_max, 120
READ_IMMEDIATE, 120
REPEAT_1, 120
REPEAT_2, 120
REPEAT_3, 120
RESIGN, 122
RX_SPEED_address, 122
ram_io_get_int, 92
ram_io_write_bytes, 92
ram_rw, 120
raw_link_address, 92
raw_ram_rw, 92
raw_speed_address, 92
raw_waveform_data, 92
rawtime, 120
receivePacket, 93
received_esc_fct, 92
received_fct, 93
Receiver_Timeout_Returns_Error, 120
Receiver_Timeout_Returns_Zero_Part_Pkt, 120
record_writes, 93
report_EEP, 120
report_EOP, 120
report_ESC_EEP, 120
report_ESC_EOP, 120
report_ESC_ESC, 121
report_FCT, 121
report_NULL, 121
report_delta, 120
report_excess_FCT, 121
report_excess_data, 121
report_first_byte, 121
report_first_null, 121
report_mid_bytes, 121
report_nchar, 121
report_parity_error, 121
report_time_code, 122
report_timeout, 122
request_link_status, 93
request_link_status_port, 93
request_rx_speed, 94
request_tx_speed, 94
router_cs, 122
router_stats, 122
router_tables, 122
rx_link, 122
SF, 122
SF_disabled, 122
SF_enabled, 122
SMA_56_pulse_width_address, 122
SO_BARRIER, 103
SO_FORWARD, 103
SO_JOIN, 103
SO_STORE, 103
SPECIAL, 122

SPECIAL_DATA_FLAGS, 122
SPECIAL_SIZE, 122
STORE, 122
SYSTEM_TYPE_401, 123
SYSTEM_TYPE_408, 123
send, 94
send_ESC, 95
send_ESC_EEP, 95
send_ESC_EOP, 95
send_ESC_ESC, 95
send_ESC_FCT, 95
send_timecode, 95
sendPacket, 95, 96
sendPacket_S, 96
set_EINTR, 97
set_active_link, 96
set_active_link_401, 97
set_active_link_408, 97
set_context, 97
set_extension_cb, 97
set_log_file, 97
set_max_packet_data, 98
set_max_packet_data_401, 98
set_max_packet_data_408, 98
set_mode, 98
set_mode_401, 99
set_mode_408, 99
set_mode_portmask, 99
set_record_file, 100
set_rx_timeout, 100
set_rx_timeout_action, 100
set_slot, 101
set_special_cb, 101
set_speed, 101
set_speed_401, 101
set_speed_408, 101
set_speed_double, 101
set_speed_double_401, 102
set_speed_double_408, 102
set_tx_record_file, 102
shutdown, 102
sma_56_pulse_width, 102
sma_56_pulse_width_401, 103
sma_56_pulse_width_408, 103
source, 103
SpaceWire_state_Connecting, 122
SpaceWire_state_ErrorReset, 122
SpaceWire_state_ErrorWait, 122
SpaceWire_state_Ready, 122
SpaceWire_state_Run, 122
SpaceWire_state_Started, 122
special_data, 103
start_time, 103
system_type, 103
TC_rx, 123
TC_rx_64, 123
TC_rx_report_enabled, 123
TC_rx_silent, 123
TC_rx_time_stamp_enabled, 123
TC_tx, 123
TC_tx_external_trigger, 123
TC_tx_format_mask, 123
TC_tx_increment_6_bits, 123
TC_tx_increment_7_bits, 123
TC_tx_increment_8_bits, 123
TC_tx_no_increment, 123
TC_tx_no_trigger, 123
TC_tx_one_code, 123
TC_tx_regular_trigger, 123
TC_tx_report_transmission, 123
TC_tx_trigger_mask, 123
TC_tx_update_code, 123
TC_tx_update_interval, 123
TIMETAG_address, 124
TRUNCATE_1, 124
TRUNCATE_2, 124
TRUNCATED, 124
TT, 124
TT_64, 124
TT_enable_reporting, 105
TT_enable_reporting_401, 105
TT_enable_reporting_408, 105
TT_now, 124
TT_report_EEP, 124
TT_report_EOP, 124
TT_report_EOP_EEP, 124
TT_report_ESC_EEP, 124
TT_report_ESC_EOP, 124
TT_report_ESC_ESC, 124
TT_report_fct, 125
TT_report_first_byte, 125
TT_report_intermediate_bytes, 125
TT_report_nothing, 125
TT_report_null, 125
TT_report_parity_error, 125
TT_report_time_code, 125
TT_report_timeout, 125
TX_SPEED_address, 125
TimeCode, 123
TimeTag, 123
TimeTag_delta, 124
TimeTag_uncertainty, 124
TimeZero, 124
timecode, 103
Timeout, 123
timetag, 105
timetag_uncertainty, 105
timezero, 105
unknown_extn_data, 105
unknown_ram_data, 106
unknown_special_data, 106
update, 106
update_es, 106
VERSION_address, 125
version, 125
waveform_data, 106

what, 106
write_EXTN, 107
write_buffer_empty, 106
write_buffer_empty_401, 107
write_buffer_empty_408, 107
write_packet, 108
Year, 125
EtherSpaceLink.Error, 57
EtherSpaceLink::Error
Error, 57
Get, 57
Event handling on spacewire links, 20
Event_cause_address
EtherSpaceLink, 115
Memory Mapped Addresses, 42
Extension codes, 29
ATOM, 30
BARRIER, 30
Delay, 30
EEP, 30
EOP, 30
ESC, 30
ESC_EEP, 30
ESC_EOP, 30
ESC_ESC, 30
ESC_FCT, 30
EVENT, 30
FCT, 31
FORWARD, 31
HOLD, 31
Header, 31
JOIN, 31
MOTA, 31
Missing_data, 31
Module, 31
Multi_byte_extn_start, 31
PERROR1, 31
PERROR2, 31
ParityError, 31
PortSelect, 31
PortSelect_max, 31
REPEAT_1, 31
REPEAT_2, 31
REPEAT_3, 31
RESIGN, 31
STORE, 32
TRUNCATE_1, 32
TRUNCATE_2, 32
TimeCode, 32
TimeTag, 32
TimeTag_delta, 32
TimeTag_uncertainty, 32
TimeZero, 32
Timeout, 32
Year, 32
extn_data
EtherSpaceLink, 80
extract_link
EtherSpaceLink, 80
extract_link_state
EtherSpaceLink, 80
extract_rx_speed
EtherSpaceLink, 80
extract_timetag
EtherSpaceLink, 80
extract_timetag_401
EtherSpaceLink, 80
extract_timetag_408
EtherSpaceLink, 80
extract_timetag_ns
EtherSpaceLink, 80
extract_timetag_ns_401
EtherSpaceLink, 81
extract_timetag_ns_408
EtherSpaceLink, 81
extract_tx_speed
EtherSpaceLink, 81

FCT
EtherSpaceLink, 117
Extension codes, 31
FLOW_CONTROL_address
EtherSpaceLink, 118
Memory Mapped Addresses, 42
FLUSH
EtherSpaceLink, 118
FORWARD
EtherSpaceLink, 118
Extension codes, 31
fastclose
EtherSpaceLink, 81
flow_control
EtherSpaceLink, 81
flush
EtherSpaceLink, 81
Sending data on a spaceiwre link, 21
flush_S
EtherSpaceLink, 83
flush_record_file
EtherSpaceLink, 83
functions, 50
device_type, 50
err, 50
error_event, 52
esc_eep, 52
esc_eop, 52
esc_esc, 52
link_selected, 52
link_timeout, 52
link_tx_speed, 52
missing_data, 52
parity_error, 52
perror1, 52
perror2, 53
port_status, 53
raw_waveform_data, 53
received_esc_fct, 53

received_fct, 53
start_time, 53
timecode, 53
timetag, 54
timetag_uncertainty, 54
timezero, 54
unknown_extn_data, 54
unknown_ram_data, 54
unknown_special_data, 55
waveform_data, 55

Get
EtherSpaceLink::Error, 57
get_HWA
EtherSpaceLink, 84
get_HWA_401
EtherSpaceLink, 84
get_HWA_408
EtherSpaceLink, 84
get_context
EtherSpaceLink, 83
get_control_packet
EtherSpaceLink, 83
get_manufacturer_string
EtherSpaceLink, 84
get_manufacturer_string_401
EtherSpaceLink, 84
get_manufacturer_string_408
EtherSpaceLink, 84
get_module_slot
EtherSpaceLink, 84
get_module_string
EtherSpaceLink, 85
get_module_string_401
EtherSpaceLink, 85
get_module_string_408
EtherSpaceLink, 85
get_module_type
EtherSpaceLink, 85
get_number_of_links
EtherSpaceLink, 85
get_number_of_links_401
EtherSpaceLink, 86
get_number_of_links_408
EtherSpaceLink, 86
get_options
EtherSpaceLink, 86
get_options_string
EtherSpaceLink, 86
get_options_string_401
EtherSpaceLink, 86
get_options_string_408
EtherSpaceLink, 86
get_packet
EtherSpaceLink, 86
Reading data from a spacewire link, 23
get_percent_file_read
EtherSpaceLink, 87
get_product_string
EtherSpaceLink, 87
get_product_string_401
EtherSpaceLink, 88
get_product_string_408
EtherSpaceLink, 88
get_receive_speed
EtherSpaceLink, 88
get_receive_speed_401
EtherSpaceLink, 88
get_receive_speed_408
EtherSpaceLink, 88
get_record_file
EtherSpaceLink, 88
get_record_size
EtherSpaceLink, 88
get_rx_flags
EtherSpaceLink, 88
Reading data from a spacewire link, 24
get_rx_timeout
EtherSpaceLink, 89
get_slot
EtherSpaceLink, 89
get_terminator
EtherSpaceLink, 89
get_timetag_401
EtherSpaceLink, 89
get_timetag_408
EtherSpaceLink, 89
get_total_raw_bytes_received
EtherSpaceLink, 89
get_version
EtherSpaceLink, 89

HOLD
EtherSpaceLink, 118
Extension codes, 31
HWA
EtherSpaceLink, 118
HWA_address
EtherSpaceLink, 118
Memory Mapped Addresses, 42
HWA_to_serial_number_string
EtherSpaceLink, 90
handle_header
EtherSpaceLink, 90
handle_module
EtherSpaceLink, 90
handle_year
EtherSpaceLink, 90
Handling Spacewire Traffic, 19
Header
EtherSpaceLink, 118
Extension codes, 31

IGNORE_address
EtherSpaceLink, 118
Memory Mapped Addresses, 42
INCOMPLETE
EtherSpaceLink, 118

JOIN
EtherSpaceLink, 118
Extension codes, 31

LINK
EtherSpaceLink, 118
LINK_SPEED
EtherSpaceLink, 119
LINK_address
EtherSpaceLink, 118
Memory Mapped Addresses, 42
LINK_mode_disabled
EtherSpaceLink, 118
LINK_mode_fixed_speed
EtherSpaceLink, 118
LINK_mode_legacy
EtherSpaceLink, 118
LINK_mode_long_timeout
EtherSpaceLink, 118
LINK_mode_master
EtherSpaceLink, 118
LINK_mode_normal
EtherSpaceLink, 118
LINK_mode_slow_speed
EtherSpaceLink, 118
LINK_state_offset
EtherSpaceLink, 119
LINK_tx_buffer_empty
EtherSpaceLink, 119
link_connected
EtherSpaceLink, 90
Physical Link Attributes, 15
link_connected_401
EtherSpaceLink, 90
link_connected_408
EtherSpaceLink, 90
link_selected
EtherSpaceLink, 90
functions, 52
link_timeout
EtherSpaceLink, 90
functions, 52
link_tx_speed
EtherSpaceLink, 90
functions, 52
log_packet
EtherSpaceLink, 91

MANUFACTURER
EtherSpaceLink, 119
MOTA
EtherSpaceLink, 119
Extension codes, 31
MSR_state_Connected
EtherSpaceLink, 119
MSR_state_NC
EtherSpaceLink, 119
max_packet_data
EtherSpaceLink, 119
Memory Mapped Addresses, 42
Memory Mapped Addresses, 42
ATI_address, 42
DESCRIPTION_address, 42
EW_address, 42
EW_source_address, 42
Event_cause_address, 42
FLOW_CONTROL_address, 42
HWA_address, 42
IGNORE_address, 42
LINK_address, 42
max_packet_data, 42
NLINKS_address, 42
OBSERVE_address, 42
OPTIONS_address, 43
PC_address, 43
PG_address, 43
RX_SPEED_address, 43
SMA_56_pulse_width_address, 43
TIMETAG_address, 43
TX_SPEED_address, 43
VERSION_address, 43
Missing_data
EtherSpaceLink, 119
Extension codes, 31
missing_data
EtherSpaceLink, 91
functions, 52
Module
EtherSpaceLink, 119
Extension codes, 31
Multi_byte_extn_start
EtherSpaceLink, 119
Extension codes, 31

NLINKS_address
EtherSpaceLink, 119
Memory Mapped Addresses, 42
negotiate
EtherSpaceLink, 91

OBSERVE_address
EtherSpaceLink, 119
Memory Mapped Addresses, 42
OPTIONS_address
EtherSpaceLink, 119
Memory Mapped Addresses, 43
Observe
EtherSpaceLink, 91
Observe_401
EtherSpaceLink, 91
Observe_408
EtherSpaceLink, 91
open
Connection, 13
EtherSpaceLink, 91
Option_SO
Error Codes, 49
EtherSpaceLink, 119

PART_EOP_EEP
EtherSpaceLink, 119
PART_EXTN
EtherSpaceLink, 119
PART_SPECIAL
EtherSpaceLink, 119
PC_address
EtherSpaceLink, 119
Memory Mapped Addresses, 43
PERROR1
EtherSpaceLink, 120
Extension codes, 31
PERROR2
EtherSpaceLink, 120
Extension codes, 31
PG_address
EtherSpaceLink, 120
Memory Mapped Addresses, 43
PRODUCT
EtherSpaceLink, 120
parity_error
EtherSpaceLink, 92
functions, 52
ParityError
EtherSpaceLink, 119
Extension codes, 31
perror1
EtherSpaceLink, 92
functions, 52
perror2
EtherSpaceLink, 92
functions, 53
Physical Link Attributes, 15
link_connected, 15
set_active_link, 15
set_mode, 15
set_mode_portmask, 16
set_speed, 17
port_status
EtherSpaceLink, 92
functions, 53
PortSelect
EtherSpaceLink, 120
Extension codes, 31
PortSelect_max
EtherSpaceLink, 120
Extension codes, 31

READ_IMMEDIATE
EtherSpaceLink, 120
REPEAT_1
EtherSpaceLink, 120
Extension codes, 31
REPEAT_2
EtherSpaceLink, 120
Extension codes, 31
REPEAT_3
EtherSpaceLink, 120
Extension codes, 31
REPORT_SPECIAL_DATA
EtherSpaceLink, 121
RESIGN
EtherSpaceLink, 122
Extension codes, 31
RETURN_SPECIAL_DATA
EtherSpaceLink, 122
RX_SPEED_address
EtherSpaceLink, 122
Memory Mapped Addresses, 43
ram_io_get_int
EtherSpaceLink, 92
ram_io_write_bytes
EtherSpaceLink, 92
ram_rw
EtherSpaceLink, 120
raw_link_address
EtherSpaceLink, 92
raw_ram_rw
EtherSpaceLink, 92
raw_speed_address
EtherSpaceLink, 92
raw_waveform_data
EtherSpaceLink, 92
functions, 53
rawtime
EtherSpaceLink, 120
Reading data from a spacewire link, 23
get_packet, 23
get_rx_flags, 24
receivePacket
EtherSpaceLink, 93
received_esc_fct
EtherSpaceLink, 92
functions, 53
received_fct
EtherSpaceLink, 93
functions, 53
Receiver_Timeout_Returns_Error
EtherSpaceLink, 120
Receiver_Timeout_Returns_Zero_Part_Pkt
EtherSpaceLink, 120
record_writes
EtherSpaceLink, 93
report_EEP
EtherSpaceLink, 120
report_EOP
EtherSpaceLink, 120
report_ESC_EEP
EtherSpaceLink, 120
report_ESC_EOP
EtherSpaceLink, 120
report_ESC_ESC
EtherSpaceLink, 121
report_FCT
EtherSpaceLink, 121
report_NULL
EtherSpaceLink, 121

report_delta
EtherSpaceLink, 120
report_excess_FCT
EtherSpaceLink, 121
report_excess_data
EtherSpaceLink, 121
report_first_byte
EtherSpaceLink, 121
report_first_null
EtherSpaceLink, 121
report_mid_bytes
EtherSpaceLink, 121
report_nchar
EtherSpaceLink, 121
report_parity_error
EtherSpaceLink, 121
report_time_code
EtherSpaceLink, 122
report_timeout
EtherSpaceLink, 122
request_link_status
EtherSpaceLink, 93
request_link_status_port
EtherSpaceLink, 93
request_rx_speed
EtherSpaceLink, 94
request_tx_speed
EtherSpaceLink, 94
router_cs
EtherSpaceLink, 122
router_stats
EtherSpaceLink, 122
router_tables
EtherSpaceLink, 122
rx_link
EtherSpaceLink, 122

SF
EtherSpaceLink, 122
SF_disabled
EtherSpaceLink, 122
SF_enabled
EtherSpaceLink, 122
SMA_56_pulse_width_address
EtherSpaceLink, 122
Memory Mapped Addresses, 43
SO_BARRIER
EtherSpaceLink, 103
SO_FORWARD
EtherSpaceLink, 103
SO_JOIN
EtherSpaceLink, 103
SO_STORE
EtherSpaceLink, 103
SPECIAL
EtherSpaceLink, 122
SPECIAL_DATA_FLAGS
EtherSpaceLink, 122
SPECIAL_SIZE
EtherSpaceLink, 122
STORE
EtherSpaceLink, 122
Extension codes, 32
SYSTEM_TYPE_401
EtherSpaceLink, 123
SYSTEM_TYPE_408
EtherSpaceLink, 123
SYSTEM_TYPE_INVALID
EtherSpaceLink, 123
send
EtherSpaceLink, 94
send_ESC
EtherSpaceLink, 95
send_ESC_EEP
EtherSpaceLink, 95
send_ESC_EOP
EtherSpaceLink, 95
send_ESC_ESC
EtherSpaceLink, 95
send_ESC_FCT
EtherSpaceLink, 95
send_timecode
EtherSpaceLink, 95
sendPacket
EtherSpaceLink, 95, 96
sendPacket_S
EtherSpaceLink, 96
Sending data on a spaceiwre link, 21
flush, 21
write_packet, 21
set_EINTR
EtherSpaceLink, 97
set_active_link
EtherSpaceLink, 96
Physical Link Attributes, 15
set_active_link_401
EtherSpaceLink, 97
set_active_link_408
EtherSpaceLink, 97
set_context
EtherSpaceLink, 97
set_extension_cb
EtherSpaceLink, 97
set_log_file
EtherSpaceLink, 97
set_max_packet_data
EtherSpaceLink, 98
set_max_packet_data_401
EtherSpaceLink, 98
set_max_packet_data_408
EtherSpaceLink, 98
set_mode
EtherSpaceLink, 98
Physical Link Attributes, 15
set_mode_401
EtherSpaceLink, 99
set_mode_408

EtherSpaceLink, 99
set_mode_portmask
EtherSpaceLink, 99
Physical Link Attributes, 16
set_record_file
EtherSpaceLink, 100
set_rx_timeout
EtherSpaceLink, 100
set_rx_timeout_action
EtherSpaceLink, 100
set_slot
EtherSpaceLink, 101
set_special_cb
EtherSpaceLink, 101
set_speed
EtherSpaceLink, 101
Physical Link Attributes, 17
set_speed_401
EtherSpaceLink, 101
set_speed_408
EtherSpaceLink, 101
set_speed_double
EtherSpaceLink, 101
set_speed_double_401
EtherSpaceLink, 102
set_speed_double_408
EtherSpaceLink, 102
set_tx_record_file
EtherSpaceLink, 102
shutdown
EtherSpaceLink, 102
sma_56_pulse_width
EtherSpaceLink, 102
sma_56_pulse_width_401
EtherSpaceLink, 103
sma_56_pulse_width_408
EtherSpaceLink, 103
source
EtherSpaceLink, 103
SpaceWire_state_Connecting
EtherSpaceLink, 122
SpaceWire_state_ErrorReset
EtherSpaceLink, 122
SpaceWire_state_ErrorWait
EtherSpaceLink, 122
SpaceWire_state_Ready
EtherSpaceLink, 122
SpaceWire_state_Run
EtherSpaceLink, 122
SpaceWire_state_Started
EtherSpaceLink, 122
special_data
EtherSpaceLink, 103
start_time
EtherSpaceLink, 103
functions, 53
system_type
EtherSpaceLink, 103
TC_rx
EtherSpaceLink, 123
TC_rx_64
EtherSpaceLink, 123
TC_rx_report_enabled
EtherSpaceLink, 123
TC_rx_silent
EtherSpaceLink, 123
TC_rx_time_stamp_enabled
EtherSpaceLink, 123
TC_tx
EtherSpaceLink, 123
TC_tx_external_trigger
EtherSpaceLink, 123
TC_tx_format_mask
EtherSpaceLink, 123
TC_tx_increment_6_bits
EtherSpaceLink, 123
TC_tx_increment_7_bits
EtherSpaceLink, 123
TC_tx_increment_8_bits
EtherSpaceLink, 123
TC_tx_no_increment
EtherSpaceLink, 123
TC_tx_no_trigger
EtherSpaceLink, 123
TC_tx_one_code
EtherSpaceLink, 123
TC_tx_regular_trigger
EtherSpaceLink, 123
TC_tx_report_transmission
EtherSpaceLink, 123
TC_tx_trigger_mask
EtherSpaceLink, 123
TC_tx_update_code
EtherSpaceLink, 123
TC_tx_update_interval
EtherSpaceLink, 123
TIMETAG_address
EtherSpaceLink, 124
Memory Mapped Addresses, 43
TRUNCATE_1
EtherSpaceLink, 124
Extension codes, 32
TRUNCATE_2
EtherSpaceLink, 124
Extension codes, 32
TRUNCATED
EtherSpaceLink, 124
TT
EtherSpaceLink, 124
TimeTag mask fields, 33
TT_64
EtherSpaceLink, 124
TimeTag mask fields, 33
TT_enable_reporting
EtherSpaceLink, 105
TimeTag, 25

TT_enable_reporting_401
EtherSpaceLink, 105
TT_enable_reporting_408
EtherSpaceLink, 105
TT_now
EtherSpaceLink, 124
TT_report_EEP
EtherSpaceLink, 124
TimeTag mask fields, 33
TT_report_EOP
EtherSpaceLink, 124
TimeTag mask fields, 33
TT_report_EOP_EEP
EtherSpaceLink, 124
TimeTag mask fields, 33
TT_report_ESC_EEP
EtherSpaceLink, 124
TimeTag mask fields, 34
TT_report_ESC_EOP
EtherSpaceLink, 124
TimeTag mask fields, 34
TT_report_ESC_ESC
EtherSpaceLink, 124
TimeTag mask fields, 34
TT_report_fct
EtherSpaceLink, 125
TimeTag mask fields, 34
TT_report_first_byte
EtherSpaceLink, 125
TimeTag mask fields, 34
TT_report_intermediate_bytes
EtherSpaceLink, 125
TimeTag mask fields, 34
TT_report_nothing
EtherSpaceLink, 125
TimeTag mask fields, 34
TT_report_null
EtherSpaceLink, 125
TimeTag mask fields, 34
TT_report_parity_error
EtherSpaceLink, 125
TimeTag mask fields, 34
TT_report_time_code
EtherSpaceLink, 125
TimeTag mask fields, 34
TT_report_timeout
EtherSpaceLink, 125
TimeTag mask fields, 34
TX_SPEED_address
EtherSpaceLink, 125
Memory Mapped Addresses, 43
TimeCode
EtherSpaceLink, 123
Extension codes, 32
TimeTag, 25
EtherSpaceLink, 123
Extension codes, 32
TT_enable_reporting, 25
TimeTag mask fields, 33
TT, 33
TT_64, 33
TT_report_EEP, 33
TT_report_EOP, 33
TT_report_EOP_EEP, 33
TT_report_ESC_EEP, 34
TT_report_ESC_EOP, 34
TT_report_ESC_ESC, 34
TT_report_fct, 34
TT_report_first_byte, 34
TT_report_intermediate_bytes, 34
TT_report_nothing, 34
TT_report_null, 34
TT_report_parity_error, 34
TT_report_time_code, 34
TT_report_timeout, 34
TimeTag_delta
EtherSpaceLink, 124
Extension codes, 32
TimeTag_uncertainty
EtherSpaceLink, 124
Extension codes, 32
TimeZero
EtherSpaceLink, 124
Extension codes, 32
timecode
EtherSpaceLink, 103
functions, 53
Timeout
EtherSpaceLink, 123
Extension codes, 32
timetag
EtherSpaceLink, 105
functions, 54
timetag_uncertainty
EtherSpaceLink, 105
functions, 54
timezero
EtherSpaceLink, 105
functions, 54

unknown_extn_data
EtherSpaceLink, 105
functions, 54
unknown_ram_data
EtherSpaceLink, 106
functions, 54
unknown_special_data
EtherSpaceLink, 106
functions, 55
update
EtherSpaceLink, 106
update_es
EtherSpaceLink, 106

VERSION_address
EtherSpaceLink, 125
Memory Mapped Addresses, 43

version
EtherSpaceLink, 125
Virtual Link Attributes, 18

waveform_data
EtherSpaceLink, 106
functions, 55
what
EtherSpaceLink, 106
write_EXTN
EtherSpaceLink, 107
write_buffer_empty
EtherSpaceLink, 106
write_buffer_empty_401
EtherSpaceLink, 107
write_buffer_empty_408
EtherSpaceLink, 107
write_packet
EtherSpaceLink, 108
Sending data on a spaceiwre link, 21

Year
EtherSpaceLink, 125
Extension codes, 32