

# Spine Robot Controller

v0.7

Generated by Doxygen 1.8.17



<b>1 Spine Surgery Robot Control Framework</b>	<b>1</b>
1.1 Introduction	1
1.2 Installation	1
1.2.1 1: Opening the box	1
<b>2 Todo List</b>	<b>3</b>
<b>3 Namespace Index</b>	<b>5</b>
3.1 Namespace List	5
<b>4 Hierarchical Index</b>	<b>7</b>
4.1 Class Hierarchy	7
<b>5 Class Index</b>	<b>9</b>
5.1 Class List	9
<b>6 File Index</b>	<b>11</b>
6.1 File List	11
<b>7 Namespace Documentation</b>	<b>13</b>
7.1 EthercatCommunication Namespace Reference	13
7.1.1 Detailed Description	13
7.2 EthercatLifeCycleNode Namespace Reference	13
<b>8 Class Documentation</b>	<b>15</b>
8.1 Controller Struct Reference	15
8.1.1 Detailed Description	15
8.1.2 Member Data Documentation	15
8.1.2.1 blue_button_	16
8.1.2.2 green_button_	16
8.1.2.3 left_d_button_	16
8.1.2.4 left_l_button_	16
8.1.2.5 left_r_button_	16
8.1.2.6 left_rb_button_	16
8.1.2.7 left_start_button_	16
8.1.2.8 left_u_button_	16
8.1.2.9 left_x_axis_	17
8.1.2.10 left_y_axis_	17
8.1.2.11 red_button_	17
8.1.2.12 right_rb_button_	17
8.1.2.13 right_start_button_	17
8.1.2.14 right_x_axis_	17
8.1.2.15 right_y_axis_	17
8.1.2.16 xbox_button_	17
8.1.2.17 yellow_button_	18

8.2 CPositionModeParam Struct Reference	18
8.2.1 Detailed Description	18
8.2.2 Member Data Documentation	18
8.2.2.1 current_controller_gain	18
8.2.2.2 interpolation_time_period	19
8.2.2.3 max_fol_err	19
8.2.2.4 max_gear_input_speed	19
8.2.2.5 max_profile_vel	19
8.2.2.6 motorRatedTorque	19
8.2.2.7 nominalCurrent	19
8.2.2.8 position_control_parameter_set	19
8.2.2.9 profile_acc	19
8.2.2.10 profile_dec	20
8.2.2.11 profile_vel	20
8.2.2.12 quick_stop_dec	20
8.2.2.13 software_position_limit	20
8.2.2.14 torque_constant	20
8.3 CSTorqueModeParam Struct Reference	20
8.3.1 Detailed Description	21
8.3.2 Member Data Documentation	21
8.3.2.1 interpolation_time_period	21
8.3.2.2 max_gear_input_speed	21
8.3.2.3 max_profile_vel	21
8.3.2.4 motorRatedTorque	21
8.3.2.5 nominalCurrent	21
8.3.2.6 profile_acc	21
8.3.2.7 profile_dec	22
8.3.2.8 profile_vel	22
8.3.2.9 quick_stop_dec	22
8.3.2.10 software_position_limit	22
8.3.2.11 torque_constant	22
8.4 CSVelocityModeParam Struct Reference	22
8.4.1 Detailed Description	23
8.4.2 Member Data Documentation	23
8.4.2.1 current_controller_gain	23
8.4.2.2 interpolation_time_period	23
8.4.2.3 max_fol_err	23
8.4.2.4 max_gear_input_speed	23
8.4.2.5 max_profile_vel	24
8.4.2.6 motorRatedTorque	24
8.4.2.7 nominalCurrent	24
8.4.2.8 profile_acc	24

8.4.2.9 profile_dec . . . . .	24
8.4.2.10 profile_vel . . . . .	24
8.4.2.11 quick_stop_dec . . . . .	24
8.4.2.12 software_position_limit . . . . .	24
8.4.2.13 torque_constant . . . . .	25
8.4.2.14 velocity_control_parameter_set . . . . .	25
8.5 EthercatLifeCycleNode::EthercatLifeCycle Class Reference . . . . .	25
8.5.1 Member Typedef Documentation . . . . .	28
8.5.1.1 TLSFAllocator . . . . .	28
8.5.2 Constructor & Destructor Documentation . . . . .	28
8.5.2.1 EthercatLifeCycle() . . . . .	28
8.5.2.2 ~EthercatLifeCycle() . . . . .	28
8.5.3 Member Function Documentation . . . . .	28
8.5.3.1 EnableMotors() . . . . .	28
8.5.3.2 GetComState() . . . . .	28
8.5.3.3 HandleControlNodeCallbacks() . . . . .	29
8.5.3.4 HandleGuiNodeCallbacks() . . . . .	29
8.5.3.5 InitEthercatCommunication() . . . . .	29
8.5.3.6 on_activate() . . . . .	29
8.5.3.7 on_cleanup() . . . . .	30
8.5.3.8 on_configure() . . . . .	30
8.5.3.9 on_deactivate() . . . . .	30
8.5.3.10 on_error() . . . . .	31
8.5.3.11 on_shutdown() . . . . .	31
8.5.3.12 PassCyclicExchange() . . . . .	31
8.5.3.13 PublishAllData() . . . . .	31
8.5.3.14 ReadFromSlaves() . . . . .	32
8.5.3.15 SetComThreadPriorities() . . . . .	32
8.5.3.16 StartEthercatCommunication() . . . . .	32
8.5.3.17 StartPdoExchange() . . . . .	32
8.5.3.18 UpdateCyclicPositionModeParameters() . . . . .	33
8.5.3.19 UpdateMotorStatePositionMode() . . . . .	33
8.5.3.20 UpdateMotorStateVelocityMode() . . . . .	33
8.5.3.21 UpdatePositionModeParameters() . . . . .	33
8.5.3.22 UpdateVelocityModeParameters() . . . . .	33
8.5.3.23 WriteToSlavesInPositionMode() . . . . .	34
8.5.3.24 WriteToSlavesVelocityMode() . . . . .	34
8.5.4 Member Data Documentation . . . . .	34
8.5.4.1 al_state_ . . . . .	34
8.5.4.2 command_ . . . . .	34
8.5.4.3 controller_ . . . . .	34
8.5.4.4 ecat_node_ . . . . .	34

8.5.4.5 emergency_status_ . . . . .	35
8.5.4.6 err_ . . . . .	35
8.5.4.7 ethercat_sched_param_ . . . . .	35
8.5.4.8 ethercat_thread_ . . . . .	35
8.5.4.9 ethercat_thread_attr_ . . . . .	35
8.5.4.10 gui_node_data_ . . . . .	35
8.5.4.11 gui_subscriber_ . . . . .	35
8.5.4.12 joystick_subscriber_ . . . . .	36
8.5.4.13 measurement_time . . . . .	36
8.5.4.14 memory_strategy . . . . .	36
8.5.4.15 motor_state_ . . . . .	36
8.5.4.16 received_data_ . . . . .	36
8.5.4.17 received_data_publisher_ . . . . .	36
8.5.4.18 sent_data_ . . . . .	37
8.5.4.19 sent_data_publisher_ . . . . .	37
8.5.4.20 timer_ . . . . .	37
8.5.4.21 timer_info_ . . . . .	37
8.6 EthercatCommunication::EthercatNode Class Reference . . . . .	37
8.6.1 Constructor & Destructor Documentation . . . . .	39
8.6.1.1 EthercatNode() . . . . .	39
8.6.1.2 ~EthercatNode() . . . . .	39
8.6.2 Member Function Documentation . . . . .	39
8.6.2.1 ActivateMaster() . . . . .	40
8.6.2.2 CheckMasterDomainState() . . . . .	40
8.6.2.3 CheckMasterState() . . . . .	40
8.6.2.4 CheckSlaveConfigurationState() . . . . .	40
8.6.2.5 ConfigDcSync() . . . . .	40
8.6.2.6 ConfigDcSyncDefault() . . . . .	41
8.6.2.7 ConfigureMaster() . . . . .	41
8.6.2.8 ConfigureSlaves() . . . . .	41
8.6.2.9 DeactivateCommunication() . . . . .	42
8.6.2.10 DefineDefaultSlaves() . . . . .	42
8.6.2.11 GetAllSlaveInformation() . . . . .	42
8.6.2.12 GetNumberOfConnectedSlaves() . . . . .	42
8.6.2.13 MapCustomPdos() . . . . .	42
8.6.2.14 MapDefaultPdos() . . . . .	43
8.6.2.15 OpenEthercatMaster() . . . . .	43
8.6.2.16 RegisterDomain() . . . . .	44
8.6.2.17 ReleaseMaster() . . . . .	44
8.6.2.18 SetCustomSlave() . . . . .	44
8.6.2.19 SetCyclicSyncPositionModeParameters() . . . . .	44
8.6.2.20 SetCyclicSyncPositionModeParametersAll() . . . . .	45

8.6.2.21 SetProfilePositionParameters()	45
8.6.2.22 SetProfilePositionParametersAll()	45
8.6.2.23 SetProfileVelocityParameters()	46
8.6.2.24 SetProfileVelocityParametersAll()	46
8.6.2.25 ShutDownEthercatMaster()	46
8.6.2.26 WaitForOperationalMode()	47
8.6.3 Member Data Documentation	47
8.6.3.1 fd	47
8.6.3.2 slaves_	47
8.7 EthercatSlave Class Reference	47
8.7.1 Constructor & Destructor Documentation	48
8.7.1.1 EthercatSlave()	48
8.7.1.2 ~EthercatSlave()	48
8.7.2 Member Function Documentation	49
8.7.2.1 CheckSlaveConfigState()	49
8.7.3 Member Data Documentation	49
8.7.3.1 data_	49
8.7.3.2 homing_param_	49
8.7.3.3 kSync0_shift_	49
8.7.3.4 motor_state_	50
8.7.3.5 offset_	50
8.7.3.6 position_param_	50
8.7.3.7 slave_config_	50
8.7.3.8 slave_config_state_	50
8.7.3.9 slave_info_	50
8.7.3.10 slave_pdo_domain_	51
8.7.3.11 velocity_param_	51
8.8 HomingParam Struct Reference	51
8.8.1 Detailed Description	51
8.8.2 Member Data Documentation	51
8.8.2.1 curr_threshold_homing	52
8.8.2.2 home_offset	52
8.8.2.3 homing_acc	52
8.8.2.4 homing_method	52
8.8.2.5 max_fol_err	52
8.8.2.6 max_profile_vel	52
8.8.2.7 quick_stop_dec	52
8.8.2.8 speed_for_switch_search	53
8.8.2.9 speed_for_zero_search	53
8.9 OffsetPDO Struct Reference	53
8.9.1 Detailed Description	53
8.9.2 Member Data Documentation	54

8.9.2.1 actual_cur . . . . .	54
8.9.2.2 actual_pos . . . . .	54
8.9.2.3 actual_tor . . . . .	54
8.9.2.4 actual_vel . . . . .	54
8.9.2.5 control_word . . . . .	54
8.9.2.6 emergency_switch . . . . .	54
8.9.2.7 error_code . . . . .	54
8.9.2.8 extra_status_reg . . . . .	55
8.9.2.9 l_limit_switch . . . . .	55
8.9.2.10 max_tor . . . . .	55
8.9.2.11 op_mode . . . . .	55
8.9.2.12 op_mode_display . . . . .	55
8.9.2.13 pos_fol_err . . . . .	55
8.9.2.14 profile_acc . . . . .	55
8.9.2.15 profile_dec . . . . .	55
8.9.2.16 profile_vel . . . . .	56
8.9.2.17 quick_stop_dec . . . . .	56
8.9.2.18 r_limit_switch . . . . .	56
8.9.2.19 status_word . . . . .	56
8.9.2.20 target_pos . . . . .	56
8.9.2.21 target_tor . . . . .	56
8.9.2.22 target_vel . . . . .	56
8.10 ProfilePosParam Struct Reference . . . . .	57
8.10.1 Detailed Description . . . . .	57
8.10.2 Member Data Documentation . . . . .	57
8.10.2.1 max_fol_err . . . . .	57
8.10.2.2 max_profile_vel . . . . .	57
8.10.2.3 motion_profile_type . . . . .	57
8.10.2.4 profile_acc . . . . .	58
8.10.2.5 profile_dec . . . . .	58
8.10.2.6 profile_vel . . . . .	58
8.10.2.7 quick_stop_dec . . . . .	58
8.11 ProfileVelocityParam Struct Reference . . . . .	58
8.11.1 Detailed Description . . . . .	58
8.11.2 Member Data Documentation . . . . .	59
8.11.2.1 max_profile_vel . . . . .	59
8.11.2.2 motion_profile_type . . . . .	59
8.11.2.3 profile_acc . . . . .	59
8.11.2.4 profile_dec . . . . .	59
8.11.2.5 quick_stop_dec . . . . .	59
8.12 ReceivedData Struct Reference . . . . .	59
8.12.1 Detailed Description . . . . .	60



8.12.2 Member Data Documentation	60
8.12.2.1 actual_cur	60
8.12.2.2 actual_pos	60
8.12.2.3 actual_tor	60
8.12.2.4 actual_vel	61
8.12.2.5 control_word	61
8.12.2.6 left_limit_switch_val	61
8.12.2.7 max_tor	61
8.12.2.8 op_mode	61
8.12.2.9 op_mode_display	61
8.12.2.10 right_limit_switch_val	61
8.12.2.11 s_emergency_switch_val	61
8.12.2.12 status_word	62
8.12.2.13 target_pos	62
8.12.2.14 target_tor	62
8.12.2.15 target_vel	62
8.12.2.16 tor_offset	62
8.12.2.17 vel_offset	62
8.13 SdoRequest Struct Reference	62
8.13.1 Detailed Description	63
8.13.2 Member Data Documentation	63
8.13.2.1 curr_threshold_homing	63
8.13.2.2 home_offset	63
8.13.2.3 homing_acc	63
8.13.2.4 homing_method	63
8.13.2.5 max_fol_err	64
8.13.2.6 max_profile_vel	64
8.13.2.7 motion_profile_type	64
8.13.2.8 profile_acc	64
8.13.2.9 profile_dec	64
8.13.2.10 profile_vel	64
8.13.2.11 quick_stop_dec	64
8.13.2.12 speed_for_switch_search	64
8.13.2.13 speed_for_zero_search	65
8.14 Timing Class Reference	65
8.14.1 Member Function Documentation	65
8.14.1.1 GetTime()	65
8.14.1.2 MeasureTimeDifference()	65
8.14.1.3 OutInfoToFile()	65
8.14.2 Member Data Documentation	66
8.14.2.1 counter_	66
8.14.2.2 last_start_time_	66

8.14.2.3 time_span . . . . .	66
8.14.2.4 timer_start_ . . . . .	66
8.14.2.5 timing_info_ . . . . .	66
<b>9 File Documentation</b>	<b>67</b>
9.1 /home/veysiadr/spinrobot_ws/src/ecat_pkg/CMakeLists.txt File Reference . . . . .	67
9.1.1 Function Documentation . . . . .	67
9.1.1.1 cmake_minimum_required() . . . . .	67
9.1.1.2 set() . . . . .	68
9.1.2 Variable Documentation . . . . .	68
9.1.2.1 WI . . . . .	68
9.2 /home/veysiadr/spinrobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_globals.hpp File Reference . . . . .	68
9.2.1 Macro Definition Documentation . . . . .	71
9.2.1.1 CLOCK_TO_USE . . . . .	71
9.2.1.2 CYCLIC_POSITION_MODE . . . . .	72
9.2.1.3 DIFF_NS . . . . .	72
9.2.1.4 ENCODER_RESOLUTION . . . . .	72
9.2.1.5 FINAL_SLAVE . . . . .	72
9.2.1.6 FIVE_DEGREE_CCW . . . . .	72
9.2.1.7 FREQUENCY . . . . .	72
9.2.1.8 GEAR_RATIO . . . . .	72
9.2.1.9 INC_PER_ROTATION . . . . .	73
9.2.1.10 MEASURE_TIMING . . . . .	73
9.2.1.11 NUM_OF_SLAVES . . . . .	73
9.2.1.12 PERIOD_MS . . . . .	73
9.2.1.13 PERIOD_NS . . . . .	73
9.2.1.14 PERIOD_US . . . . .	73
9.2.1.15 POSITION_MODE . . . . .	73
9.2.1.16 RESET_BIT . . . . .	74
9.2.1.17 SET_BIT . . . . .	74
9.2.1.18 TEST_BIT . . . . .	74
9.2.1.19 THIRTY_DEGREE_CCW . . . . .	74
9.2.1.20 TIMESPEC2NS . . . . .	74
9.2.1.21 VELOCITY_MODE . . . . .	74
9.2.2 Enumeration Type Documentation . . . . .	74
9.2.2.1 ControlStructureBits . . . . .	74
9.2.2.2 ErrorRegisterBits . . . . .	75
9.2.2.3 MotorStates . . . . .	75
9.2.2.4 OpMode . . . . .	76
9.2.2.5 SensorConfig . . . . .	76
9.2.3 Function Documentation . . . . .	77
9.2.3.1 timespec_add() . . . . .	77

9.2.4 Variable Documentation . . . . .	77
9.2.4.1 g_cycle_time . . . . .	77
9.2.4.2 g_kNsPerSec . . . . .	77
9.2.4.3 g_kNumberOfServoDrivers . . . . .	78
9.2.4.4 g_master . . . . .	78
9.2.4.5 g_master_domain . . . . .	78
9.2.4.6 g_master_domain_state . . . . .	78
9.2.4.7 g_master_state . . . . .	78
9.2.4.8 g_sync_ref_counter . . . . .	78
9.2.4.9 g_sync_timer . . . . .	79
9.2.4.10 sig . . . . .	79
9.3 /home/veysiadm/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_lifecycle.hpp File Reference . . .	79
9.3.1 Typedef Documentation . . . . .	80
9.3.1.1 TLSFAllocator . . . . .	80
9.4 /home/veysiadm/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_node.hpp File Reference . . . .	80
9.5 /home/veysiadm/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_slave.hpp File Reference . . . .	81
9.6 /home/veysiadm/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/object_dictionary.hpp File Reference .	83
9.6.1 Macro Definition Documentation . . . . .	85
9.6.1.1 OD_ACTIVE_FIELDBUS . . . . .	85
9.6.1.2 OD_CHECK_ERROR . . . . .	85
9.6.1.3 OD_CONTROL_STRUCTURE . . . . .	86
9.6.1.4 OD_CONTROL_WORD . . . . .	86
9.6.1.5 OD_CURRENT_ACTUAL_VALUE . . . . .	86
9.6.1.6 OD_CUSTOM_PERSISTENT_MEMORY_1 . . . . .	86
9.6.1.7 OD_CUSTOM_PERSISTENT_MEMORY_2 . . . . .	86
9.6.1.8 OD_CUSTOM_PERSISTENT_MEMORY_3 . . . . .	86
9.6.1.9 OD_CUSTOM_PERSISTENT_MEMORY_4 . . . . .	86
9.6.1.10 OD_DC_CIRCUIT_LINK_VOLTAGE . . . . .	86
9.6.1.11 OD_DIGITAL_INCREMENTAL_ENCODER_1_TYPE . . . . .	87
9.6.1.12 OD_DIGITAL_INPUTS . . . . .	87
9.6.1.13 OD_DIGITAL_OUTPUTS . . . . .	87
9.6.1.14 OD_ERROR_CODE . . . . .	87
9.6.1.15 OD_ERROR_REGISTER . . . . .	87
9.6.1.16 OD_EXTRA_STATUS_REGISTER . . . . .	87
9.6.1.17 OD_GEAR_MAX_INPUT_SPEED . . . . .	87
9.6.1.18 OD_GEAR_MISC_CONFIGURATION . . . . .	87
9.6.1.19 OD_GEAR_REDUCTION_DENOMINATOR . . . . .	88
9.6.1.20 OD_GEAR_REDUCTION_NUMERATOR . . . . .	88
9.6.1.21 OD_INTERPOLATION_TIME_PERIOD . . . . .	88
9.6.1.22 OD_INTERPOLATION_TIME_UNIT . . . . .	88
9.6.1.23 OD_LINEAR_RAMP_TRAPEZOIDAL . . . . .	88
9.6.1.24 OD_MAX_CURRENT . . . . .	88

9.6.1.25 OD_MAX_FOLLOWING_ERROR . . . . .	88
9.6.1.26 OD_MAX_MOTOR_SPEED . . . . .	88
9.6.1.27 OD_MAX_PROFILE_VELOCITY . . . . .	89
9.6.1.28 OD_MOTION_PROFILE_TYPE . . . . .	89
9.6.1.29 OD_MOTOR_DATA_NOMINAL_CURRENT . . . . .	89
9.6.1.30 OD_MOTOR_DATA_NUMBER_OF_POLE_PAIRS . . . . .	89
9.6.1.31 OD_MOTOR_DATA_OUTPUT_CURRENT_LIMIT . . . . .	89
9.6.1.32 OD_MOTOR_DATA_THERMAL_TIME_CONSTANT_WINDINGS . . . . .	89
9.6.1.33 OD_MOTOR_DATA_TORQUE_CONSTANT . . . . .	89
9.6.1.34 OD_OPERATION_MODE . . . . .	89
9.6.1.35 OD_OPERATION_MODE_DISPLAY . . . . .	90
9.6.1.36 OD_POSITION_ACTUAL_VAL . . . . .	90
9.6.1.37 OD_POSITION_DEMAND . . . . .	90
9.6.1.38 OD_POSITON_FOLLOWING_ERROR . . . . .	90
9.6.1.39 OD_POSITONCOUNTS . . . . .	90
9.6.1.40 OD_POWER_SUPPLY_VOLTAGE . . . . .	90
9.6.1.41 OD_PROFILE_ACCELERATION . . . . .	90
9.6.1.42 OD_PROFILE_DECELERATION . . . . .	90
9.6.1.43 OD_PROFILE_VELOCITY . . . . .	91
9.6.1.44 OD_QUICK_STOP_DECELERATION . . . . .	91
9.6.1.45 OD_QUICK_STOP_MODE . . . . .	91
9.6.1.46 OD_SENSOR_CONFIGURATION . . . . .	91
9.6.1.47 OD_STATUS_WORD . . . . .	91
9.6.1.48 OD_STOP_OPTION_CODE . . . . .	91
9.6.1.49 OD_TARGET_POSITION . . . . .	91
9.6.1.50 OD_TARGET_TORQUE . . . . .	92
9.6.1.51 OD_TARGET_VELOCITY . . . . .	92
9.6.1.52 OD_TORQUE_ACTUAL_VALUE . . . . .	92
9.6.1.53 OD_TORQUE_MAX . . . . .	92
9.6.1.54 OD_TORQUE_OFFSET . . . . .	92
9.6.1.55 OD_VELOCITY_ACTUAL_VALUE . . . . .	92
9.6.1.56 OD_VELOCITY_ADDRESS . . . . .	92
9.6.1.57 OD_VELOCITY_ENCODER_RESOLUTION_DEN . . . . .	92
9.6.1.58 OD_VELOCITY_ENCODER_RESOLUTION_NUM . . . . .	93
9.6.1.59 OD_VELOCITY_OFFSET . . . . .	93
9.6.1.60 SM_COMM_RESET . . . . .	93
9.6.1.61 SM_CW_ENABLE_OPERATION . . . . .	93
9.6.1.62 SM_CW_ENABLE_VOLTAGE . . . . .	93
9.6.1.63 SM_CW_FAULT_RESET . . . . .	93
9.6.1.64 SM_CW_OD_HALT . . . . .	93
9.6.1.65 SM_CW_QUICK_STOP . . . . .	93
9.6.1.66 SM_CW_SWITCH_ON . . . . .	94

9.6.1.67 SM_EXPEDITE . . . . .	94
9.6.1.68 SM_FAULT . . . . .	94
9.6.1.69 SM_FAULT2 . . . . .	94
9.6.1.70 SM_FAULT_REACTION_ACTIVE . . . . .	94
9.6.1.71 SM_FAULTREACTIONACTIVE2 . . . . .	94
9.6.1.72 SM_FSAFROMSTATUSWORD . . . . .	94
9.6.1.73 SM_FULL_RESET . . . . .	95
9.6.1.74 SM_GO_ENABLE . . . . .	95
9.6.1.75 SM_GO_READY_TO_SWITCH_ON . . . . .	95
9.6.1.76 SM_GO_SWITCH_ON . . . . .	95
9.6.1.77 SM_GO_SWITCH_ON_DISABLE . . . . .	95
9.6.1.78 SM_NOT_READY_TO_SWITCH_ON . . . . .	95
9.6.1.79 SM_NOT_READY_TO_SWITCH_ON_2 . . . . .	95
9.6.1.80 SM_OPERATION_ENABLED . . . . .	95
9.6.1.81 SM_QUICK_STOP_ACTIVE . . . . .	96
9.6.1.82 SM_QUICKSTOP . . . . .	96
9.6.1.83 SM_READY_TO_SWITCH_ON . . . . .	96
9.6.1.84 SM_RELATIVE_POS . . . . .	96
9.6.1.85 SM_RUN . . . . .	96
9.6.1.86 SM_START . . . . .	96
9.6.1.87 SM_SW_FAULT . . . . .	96
9.6.1.88 SM_SW_INTERNAL_LIMIT_ACTIVE . . . . .	96
9.6.1.89 SM_SW_OPERATION_ENABLED . . . . .	97
9.6.1.90 SM_SW_QUICK_STOP . . . . .	97
9.6.1.91 SM_SW_READY_TO_SWITCH_ON . . . . .	97
9.6.1.92 SM_SW_REMOTE . . . . .	97
9.6.1.93 SM_SW_SWITCH_ON_DISABLED . . . . .	97
9.6.1.94 SM_SW_SWITCHED_ON . . . . .	97
9.6.1.95 SM_SW_TARGET_REACHED . . . . .	97
9.6.1.96 SM_SW_VOLTAGE_ENABLED . . . . .	97
9.6.1.97 SM_SW_WARNING . . . . .	98
9.6.1.98 SM_SWITCH_ON_DISABLED . . . . .	98
9.6.1.99 SM_SWITCH_ON_DISABLED_2 . . . . .	98
9.6.1.100 SM_SWITCHED_ON . . . . .	98
9.7 /home/veysiadm/spinrobot_ws/src/ecat_pkg/include/ecat_pkg/timing.hpp File Reference . . . . .	98
9.7.1 Macro Definition Documentation . . . . .	99
9.7.1.1 NUMBER_OF_SAMPLES . . . . .	99
9.8 /home/veysiadm/spinrobot_ws/src/ecat_pkg/src/ecat_lifecycle.cpp File Reference . . . . .	99
9.9 /home/veysiadm/spinrobot_ws/src/ecat_pkg/src/ecat_node.cpp File Reference . . . . .	100
9.9.1 Variable Documentation . . . . .	100
9.9.1.1 g_master . . . . .	100
9.9.1.2 g_master_domain . . . . .	100

---

9.9.1.3 <a href="#">g_master_domain_state</a>	101
9.9.1.4 <a href="#">g_master_state</a>	101
9.9.1.5 <a href="#">g_sync_ref_counter</a>	101
9.9.1.6 <a href="#">g_sync_timer</a>	101
9.10 <a href="#">/home/veysiadm/spinrobot_ws/src/ecat_pkg/src/ecat_slave.cpp File Reference</a>	101
9.11 <a href="#">/home/veysiadm/spinrobot_ws/src/ecat_pkg/src/main.cpp File Reference</a>	102
9.11.1 Function Documentation	102
9.11.1.1 <a href="#">main()</a>	102
9.11.1.2 <a href="#">signalHandler()</a>	102
9.11.2 Variable Documentation	102
9.11.2.1 <a href="#">ecat_lifecycle_node</a>	103
9.12 <a href="#">/home/veysiadm/spinrobot_ws/src/ecat_pkg/src/timing.cpp File Reference</a>	103
<b>Index</b>	<b>105</b>

## **Chapter 1**

# **Spine Surgery Robot Control Framework**

### **1.1 Introduction**

This is the introduction.

### **1.2 Installation**

#### **1.2.1 1: Opening the box**

etc...





## Chapter 2

## Todo List

**Member** [EthercatCommunication::EthercatNode::SetProfileVelocityParametersAll](#) ([ProfileVelocityParam](#)  
&P)

Add error code to all functions. Instead of returning -1.



## Chapter 3

# Namespace Index

### 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">EthercatCommunication</a>	
ROS2 Headers . . . . .	<a href="#">13</a>
<a href="#">EthercatLifeCycleNode</a> . . . . .	<a href="#">13</a>



## Chapter 4

# Hierarchical Index

### 4.1 Class Hierarchy

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

Controller . . . . .	15
CSPositionModeParam . . . . .	18
CSTorqueModeParam . . . . .	20
CSVelocityModeParam . . . . .	22
EthercatCommunication::EthercatNode . . . . .	37
EthercatSlave . . . . .	47
HomingParam . . . . .	51
LifecycleNode	
EthercatLifeCycleNode::EthercatLifeCycle . . . . .	25
OffsetPDO . . . . .	53
ProfilePosParam . . . . .	57
ProfileVelocityParam . . . . .	58
ReceivedData . . . . .	59
SdoRequest . . . . .	62
Timing . . . . .	65



## Chapter 5

# Class Index

### 5.1 Class List

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

<a href="#">Controller</a>	
<a href="#">Controller Parameters</a>	15
<a href="#">CSPositionModeParam</a>	
Struct contains configuration parameters for cyclic sync. position mode	18
<a href="#">CSTorqueModeParam</a>	
Struct contains configuration parameters for cyclic sync. torque mode	20
<a href="#">CSVelocityModeParam</a>	
Struct contains configuration parameters for cyclic sync. velocity mode	22
<a href="#">EthercatLifeCycleNode::EthercatLifeCycle</a>	25
<a href="#">EthercatCommunication::EthercatNode</a>	37
<a href="#">EthercatSlave</a>	47
<a href="#">HomingParam</a>	
Parameters that should be specified in homing mode	51
<a href="#">OffsetPDO</a>	
Offset for PDO entries to register PDOs	53
<a href="#">ProfilePosParam</a>	
Parameters that should be specified in position mode	57
<a href="#">ProfileVelocityParam</a>	
Parameters that should be specified in velocity mode	58
<a href="#">ReceivedData</a>	
Received feedback data from slaves	59
<a href="#">SdoRequest</a>	
EtherCAT SDO request structure for configuration phase	62
<a href="#">Timing</a>	65





## Chapter 6

# File Index

### 6.1 File List

Here is a list of all files with brief descriptions:

/home/veysiadm/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_globals.hpp . . . . .	68
/home/veysiadm/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_lifecycle.hpp . . . . .	79
/home/veysiadm/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_node.hpp . . . . .	80
/home/veysiadm/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_slave.hpp . . . . .	81
/home/veysiadm/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/object_dictionary.hpp . . . . .	83
/home/veysiadm/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/timing.hpp . . . . .	98
/home/veysiadm/spinerobot_ws/src/ecat_pkg/src/ecat_lifecycle.cpp . . . . .	99
/home/veysiadm/spinerobot_ws/src/ecat_pkg/src/ecat_node.cpp . . . . .	100
/home/veysiadm/spinerobot_ws/src/ecat_pkg/src/ecat_slave.cpp . . . . .	101
/home/veysiadm/spinerobot_ws/src/ecat_pkg/src/main.cpp . . . . .	102
/home/veysiadm/spinerobot_ws/src/ecat_pkg/src/timing.cpp . . . . .	103



## Chapter 7

# Namespace Documentation

### 7.1 EthercatCommunication Namespace Reference

ROS2 Headers.

#### Classes

- class [EthercatNode](#)

#### 7.1.1 Detailed Description

ROS2 Headers.

### 7.2 EthercatLifeCycleNode Namespace Reference

#### Classes

- class [EthercatLifeCycle](#)



## Chapter 8

# Class Documentation

### 8.1 Controller Struct Reference

[Controller](#) Parameters.

```
#include <ecat_globals.hpp>
```

#### Public Attributes

- float [left\\_x\\_axis\\_](#)
- float [left\\_y\\_axis\\_](#)
- float [right\\_x\\_axis\\_](#)
- float [right\\_y\\_axis\\_](#)
- uint8\_t [blue\\_button\\_](#)
- uint8\_t [green\\_button\\_](#)
- uint8\_t [red\\_button\\_](#)
- uint8\_t [yellow\\_button\\_](#)
- uint8\_t [left\\_r\\_button\\_](#)
- uint8\_t [left\\_l\\_button\\_](#)
- uint8\_t [left\\_u\\_button\\_](#)
- uint8\_t [left\\_d\\_button\\_](#)
- uint8\_t [left\\_rb\\_button\\_](#)
- uint8\_t [right\\_rb\\_button\\_](#)
- uint8\_t [left\\_start\\_button\\_](#)
- uint8\_t [right\\_start\\_button\\_](#)
- uint8\_t [xbox\\_button\\_](#)

#### 8.1.1 Detailed Description

[Controller](#) Parameters.

#### 8.1.2 Member Data Documentation

#### 8.1.2.1 blue\_button\_

`uint8_t Controller::blue_button_`

#### 8.1.2.2 green\_button\_

`uint8_t Controller::green_button_`

#### 8.1.2.3 left\_d\_button\_

`uint8_t Controller::left_d_button_`

#### 8.1.2.4 left\_l\_button\_

`uint8_t Controller::left_l_button_`

#### 8.1.2.5 left\_r\_button\_

`uint8_t Controller::left_r_button_`

#### 8.1.2.6 left\_rb\_button\_

`uint8_t Controller::left_rb_button_`

#### 8.1.2.7 left\_start\_button\_

`uint8_t Controller::left_start_button_`

#### 8.1.2.8 left\_u\_button\_

`uint8_t Controller::left_u_button_`

#### 8.1.2.9 left\_x\_axis\_

```
float Controller::left_x_axis_
```

#### 8.1.2.10 left\_y\_axis\_

```
float Controller::left_y_axis_
```

#### 8.1.2.11 red\_button\_

```
uint8_t Controller::red_button_
```

#### 8.1.2.12 right\_rb\_button\_

```
uint8_t Controller::right_rb_button_
```

#### 8.1.2.13 right\_start\_button\_

```
uint8_t Controller::right_start_button_
```

#### 8.1.2.14 right\_x\_axis\_

```
float Controller::right_x_axis_
```

#### 8.1.2.15 right\_y\_axis\_

```
float Controller::right_y_axis_
```

#### 8.1.2.16 xbox\_button\_

```
uint8_t Controller::xbox_button_
```

### 8.1.2.17 yellow\_button\_

```
uint8_t Controller::yellow_button_
```

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

- [/home/veysiadr/spinrobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/ecat\\_globals.hpp](#)

## 8.2 CSPositionModeParam Struct Reference

Struct contains configuration parameters for cyclic sync. position mode.

```
#include <ecat_globals.hpp>
```

### Public Attributes

- uint32\_t [nominal\\_current](#)
- uint16\_t [torque\\_constant](#)
- uint32\_t [current\\_controller\\_gain](#)
- uint32\_t [position\\_control\\_parameter\\_set](#)
- uint32\_t [software\\_position\\_limit](#)
- uint16\_t [motor\\_rated\\_torque](#)
- uint32\_t [max\\_gear\\_input\\_speed](#)
- uint32\_t [profile\\_vel](#)
- uint32\_t [profile\\_acc](#)
- uint32\_t [profile\\_dec](#)
- uint32\_t [max\\_fol\\_err](#)
- uint32\_t [max\\_profile\\_vel](#)
- uint32\_t [quick\\_stop\\_dec](#)
- uint32\_t [interpolation\\_time\\_period](#)

### 8.2.1 Detailed Description

Struct contains configuration parameters for cyclic sync. position mode.

### 8.2.2 Member Data Documentation

#### 8.2.2.1 current\_controller\_gain

```
uint32_t CSPositionModeParam::current_controller_gain
```



### 8.2.2.2 interpolation\_time\_period

uint32\_t CSPositionModeParam::interpolation\_time\_period

### 8.2.2.3 max\_fol\_err

uint32\_t CSPositionModeParam::max\_fol\_err

### 8.2.2.4 max\_gear\_input\_speed

uint32\_t CSPositionModeParam::max\_gear\_input\_speed

### 8.2.2.5 max\_profile\_vel

uint32\_t CSPositionModeParam::max\_profile\_vel

### 8.2.2.6 motorRatedTorque

uint16\_t CSPositionModeParam::motorRatedTorque

### 8.2.2.7 nominalCurrent

uint32\_t CSPositionModeParam::nominalCurrent

### 8.2.2.8 positionControlParameterSet

uint32\_t CSPositionModeParam::positionControlParameterSet

### 8.2.2.9 profileAcc

uint32\_t CSPositionModeParam::profileAcc

#### 8.2.2.10 profile\_dec

```
uint32_t CSPositionModeParam::profile_dec
```

#### 8.2.2.11 profile\_vel

```
uint32_t CSPositionModeParam::profile_vel
```

#### 8.2.2.12 quick\_stop\_dec

```
uint32_t CSPositionModeParam::quick_stop_dec
```

#### 8.2.2.13 software\_position\_limit

```
uint32_t CSPositionModeParam::software_position_limit
```

#### 8.2.2.14 torque\_constant

```
uint16_t CSPositionModeParam::torque_constant
```

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

- [/home/veysiadrn/spineroobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/ecat\\_globals.hpp](/home/veysiadrn/spineroobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_globals.hpp)

## 8.3 CSTorqueModeParam Struct Reference

Struct contains configuration parameters for cyclic sync. torque mode.

```
#include <ecat_globals.hpp>
```

### Public Attributes

- uint32\_t [nominal\\_current](#)
- uint16\_t [torque\\_constant](#)
- uint32\_t [software\\_position\\_limit](#)
- uint16\_t [motor\\_rated\\_torque](#)
- uint32\_t [max\\_gear\\_input\\_speed](#)
- uint32\_t [profile\\_vel](#)
- uint32\_t [profile\\_acc](#)
- uint32\_t [profile\\_dec](#)
- uint32\_t [max\\_profile\\_vel](#)
- uint32\_t [quick\\_stop\\_dec](#)
- uint32\_t [interpolation\\_time\\_period](#)

### 8.3.1 Detailed Description

Struct contains configuration parameters for cyclic sync. torque mode.

### 8.3.2 Member Data Documentation

#### 8.3.2.1 interpolation\_time\_period

```
uint32_t CSTorqueModeParam::interpolation_time_period
```

#### 8.3.2.2 max\_gear\_input\_speed

```
uint32_t CSTorqueModeParam::max_gear_input_speed
```

#### 8.3.2.3 max\_profile\_vel

```
uint32_t CSTorqueModeParam::max_profile_vel
```

#### 8.3.2.4 motorRatedTorque

```
uint16_t CSTorqueModeParam::motorRatedTorque
```

#### 8.3.2.5 nominalCurrent

```
uint32_t CSTorqueModeParam::nominalCurrent
```

#### 8.3.2.6 profileAcc

```
uint32_t CSTorqueModeParam::profileAcc
```

### 8.3.2.7 profile\_dec

```
uint32_t CSTorqueModeParam::profile_dec
```

### 8.3.2.8 profile\_vel

```
uint32_t CSTorqueModeParam::profile_vel
```

### 8.3.2.9 quick\_stop\_dec

```
uint32_t CSTorqueModeParam::quick_stop_dec
```

### 8.3.2.10 software\_position\_limit

```
uint32_t CSTorqueModeParam::software_position_limit
```

### 8.3.2.11 torque\_constant

```
uint16_t CSTorqueModeParam::torque_constant
```

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

- [/home/veysiadrn/spinerobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/ecat\\_globals.hpp](/home/veysiadrn/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_globals.hpp)

## 8.4 CSVelocityModeParam Struct Reference

Struct contains configuration parameters for cyclic sync. velocity mode.

```
#include <ecat_globals.hpp>
```

## Public Attributes

- uint32\_t [nominal\\_current](#)
- uint16\_t [torque\\_constant](#)
- uint32\_t [current\\_controller\\_gain](#)
- uint32\_t [velocity\\_control\\_parameter\\_set](#)
- uint32\_t [software\\_position\\_limit](#)
- uint16\_t [motor\\_rated\\_torque](#)
- uint32\_t [max\\_gear\\_input\\_speed](#)
- uint32\_t [profile\\_vel](#)
- uint32\_t [profile\\_acc](#)
- uint32\_t [profile\\_dec](#)
- uint32\_t [max\\_fol\\_err](#)
- uint32\_t [max\\_profile\\_vel](#)
- uint32\_t [quick\\_stop\\_dec](#)
- uint32\_t [interpolation\\_time\\_period](#)

### 8.4.1 Detailed Description

Struct contains configuration parameters for cyclic sync. velocity mode.

### 8.4.2 Member Data Documentation

#### 8.4.2.1 [current\\_controller\\_gain](#)

```
uint32_t CSVelocityModeParam::current_controller_gain
```

#### 8.4.2.2 [interpolation\\_time\\_period](#)

```
uint32_t CSVelocityModeParam::interpolation_time_period
```

#### 8.4.2.3 [max\\_fol\\_err](#)

```
uint32_t CSVelocityModeParam::max_fol_err
```

#### 8.4.2.4 [max\\_gear\\_input\\_speed](#)

```
uint32_t CSVelocityModeParam::max_gear_input_speed
```

#### 8.4.2.5 max\_profile\_vel

```
uint32_t CSVelocityModeParam::max_profile_vel
```

#### 8.4.2.6 motorRatedTorque

```
uint16_t CSVelocityModeParam::motorRatedTorque
```

#### 8.4.2.7 nominalCurrent

```
uint32_t CSVelocityModeParam::nominalCurrent
```

#### 8.4.2.8 profileAcc

```
uint32_t CSVelocityModeParam::profileAcc
```

#### 8.4.2.9 profileDec

```
uint32_t CSVelocityModeParam::profileDec
```

#### 8.4.2.10 profileVel

```
uint32_t CSVelocityModeParam::profileVel
```

#### 8.4.2.11 quickStopDec

```
uint32_t CSVelocityModeParam::quickStopDec
```

#### 8.4.2.12 softwarePositionLimit

```
uint32_t CSVelocityModeParam::softwarePositionLimit
```

#### 8.4.2.13 torque\_constant

```
uint16_t CSVelocityModeParam::torque_constant
```

#### 8.4.2.14 velocity\_control\_parameter\_set

```
uint32_t CSVelocityModeParam::velocity_control_parameter_set
```

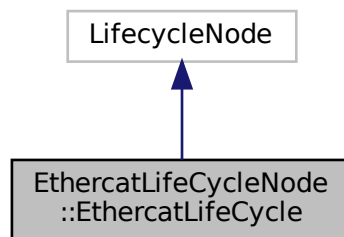
The documentation for this struct was generated from the following file:

- /home/veysiadr/spinrobot\_ws/src/ecat\_pkg/include/ecat\_pkg/[ecat\\_globals.hpp](#)

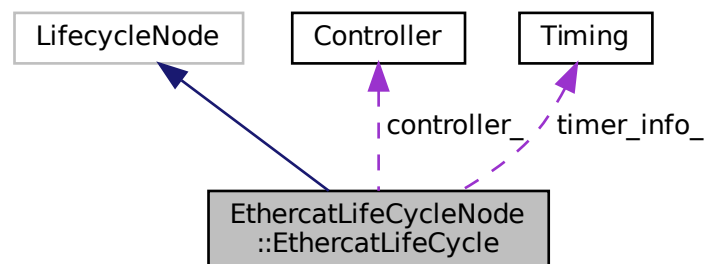
## 8.5 EthercatLifeCycleNode::EthercatLifeCycle Class Reference

```
#include <ecat_lifecycle.hpp>
```

Inheritance diagram for EthercatLifeCycleNode::EthercatLifeCycle:



Collaboration diagram for EthercatLifeCycleNode::EthercatLifeCycle:



## Public Member Functions

- [EthercatLifeCycle](#) ()
- [~EthercatLifeCycle](#) ()

## Private Types

- `template<typename T = void>`  
using [TLSFAllocator](#) = `tlsf_heap_allocator< T >`

## Private Member Functions

- `node_interfaces::LifecycleNodeInterface::CallbackReturn` [on\\_configure](#) (const State &)  
*Ethercat lifecycle node configuration function, node will start with this function For more information about Lifecycle node and it's interfaces check below link : [https://design.ros2.org/articles/node\\_lifecycle.html](https://design.ros2.org/articles/node_lifecycle.html).*
- `node_interfaces::LifecycleNodeInterface::CallbackReturn` [on\\_activate](#) (const State &)  
*Activates Ethercat lifecycle node and starts real-time Ethercat communication. All publishing is done in real-time loop in this active state.*
- `node_interfaces::LifecycleNodeInterface::CallbackReturn` [on\\_deactivate](#) (const State &)  
*Deactivates Ethercat lifecycle node, turns of real-time communication.*
- `node_interfaces::LifecycleNodeInterface::CallbackReturn` [on\\_cleanup](#) (const State &)  
*Cleans up all variables and datas assigned by Ethercat lifecycle node.*
- `node_interfaces::LifecycleNodeInterface::CallbackReturn` [on\\_shutdown](#) (const State &)  
*Shuts down EtherCAT lifecycle node, releases Ethercat master.*
- `node_interfaces::LifecycleNodeInterface::CallbackReturn` [on\\_error](#) (const State &)  
*There isn't any error recovery functionality for this node, just resets nodes. Reconfiguration is needed for restarting communication.*
- void [HandleControlNodeCallbacks](#) (const sensor\_msgs::msg::Joy::SharedPtr msg)  
*This function handles callbacks from control node. It will update received values from controller node.*
- int [SetComThreadPriorities](#) ()  
*Sets Ethercat communication thread's properties After this function called user must call [StartEthercatCommunication\(\)](#) function].*
- int [InitEthercatCommunication](#) ()  
*Encapsulates all configuration steps for the EtherCAT communication with default slaves. And waits for connected slaves to become operational.*
- int [StartEthercatCommunication](#) ()  
*Starts EtherCAT communication.*
- void [StartPdoExchange](#) (void \*instance)  
*Realtime cyclic Pdo exchange function which will constantly read/write values from/to slaves.*
- int [GetComState](#) ()  
*Gets master's communication state.*
- void [ReadFromSlaves](#) ()  
*Reads data from slaves and updates received data structure to be published.*
- int [PublishAllData](#) ()  
*Publishes all data that master received and will be sent.*
- void [EnableMotors](#) ()  
*Enables connected motor drives based on CIA402.*
- void [WriteToSlavesVelocityMode](#) ()  
*Updates data that will be sent to slaves. This updated data will be published as well.*
- void [WriteToSlavesInPositionMode](#) ()



- *Writes target position and control word to motor in profile position mode.*
- void [UpdateVelocityModeParameters](#) ()  
*Acquired data from subscribed controller topic will be assigned as motor speed parameter.*
- void [UpdatePositionModeParameters](#) ()  
*Acquired data from subscribed controller topic will be assigned as motor target position parameter.*
- void [UpdateCyclicPositionModeParameters](#) ()  
*Acquired data from subscribed controller topic will be assigned as motor cyclic target position parameter in configured interpolation time.*
- void [UpdateMotorStateVelocityMode](#) ()  
*Updates motor control word and motor state in velocity mode based on CIA402.*
- void [UpdateMotorStatePositionMode](#) ()  
*Updates motor control word and motor state in position mode based on CIA402 state machine..*
- void [HandleGuiNodeCallbacks](#) (const std\_msgs::msg::UInt8::SharedPtr gui\_sub)  
*This function will handle values from GUI node. Updates parameters based on GUI node inputs.*

## Static Private Member Functions

- static void \* [PassCyclicExchange](#) (void \*arg)  
*Helper function to enter pthread\_create, since pthread's are C function it doesn't accept class member function, to pass class member function this helper function is created.*

## Private Attributes

- rclcpp::TimerBase::SharedPtr [timer\\_](#)
- LifecyclePublisher< ecats\_msgs::msg::DataReceived >::SharedPtr [received\\_data\\_publisher\\_](#)  
*This lifecycle publisher will be used to publish received feedback data from slaves.*
- LifecyclePublisher< ecats\_msgs::msg::DataSent >::SharedPtr [sent\\_data\\_publisher\\_](#)  
*This lifecycle publisher will be used to publish sent data from master to slaves.*
- rclcpp::Subscription< sensor\_msgs::msg::Joy >::SharedPtr [joystick\\_subscriber\\_](#)  
*This subscriber will be used to receive data from controller node.*
- rclcpp::Subscription< std\_msgs::msg::UInt8 >::SharedPtr [gui\\_subscriber\\_](#)
- ecats\_msgs::msg::DataReceived [received\\_data\\_](#)
- ecats\_msgs::msg::DataSent [sent\\_data\\_](#)
- std::unique\_ptr< EthercatNode > [ecat\\_node\\_](#)
- pthread\_t [ethercat\\_thread\\_](#)  
*pthread create required parameters.*
- struct sched\_param [ethercat\\_sched\\_param\\_](#) = {}
- pthread\_attr\_t [ethercat\\_thread\\_attr\\_](#)
- int32\_t [err\\_](#)
- uint8\_t [al\\_state\\_](#) = 0  
*Application layer of slaves seen by master.(INIT/PREOP/SAFEOP/OP)*
- uint32\_t [motor\\_state\\_](#) [[g\\_kNumberOfServoDrivers](#)]
- uint32\_t [command\\_](#) = 0x004F
- [Controller](#) [controller\\_](#)
- uint8\_t [gui\\_node\\_data\\_](#) = 1  
*Values will be sent by controller node and will be assigned to variables below.*
- uint8\_t [emergency\\_status\\_](#) = 1
- rclcpp::memory\_strategy::MemoryStrategy::SharedPtr [memory\\_strategy](#)
- std::int32\_t [measurement\\_time](#) = 0
- [Timing](#) [timer\\_info\\_](#)

## 8.5.1 Member Typedef Documentation

### 8.5.1.1 TLSFAllocator

```
template<typename T = void>
using EthercatLifeCycleNode::EthercatLifeCycle::TLSFAllocator = tlsf_heap_allocator<T> [private]
```

## 8.5.2 Constructor & Destructor Documentation

### 8.5.2.1 EthercatLifeCycle()

```
EthercatLifeCycle::EthercatLifeCycle ( )
```

### 8.5.2.2 ~EthercatLifeCycle()

```
EthercatLifeCycle::~~EthercatLifeCycle ( )
```

## 8.5.3 Member Function Documentation

### 8.5.3.1 EnableMotors()

```
void EthercatLifeCycle::EnableMotors ( ) [private]
```

Enables connected motor drives based on CIA402.

### 8.5.3.2 GetComState()

```
int EthercatLifeCycle::GetComState ( ) [private]
```

Gets master's communication state.

See also

`ec_al_state_t`

Returns

Application layer state for master.

### 8.5.3.3 HandleControlNodeCallbacks()

```
void EthercatLifeCycle::HandleControlNodeCallbacks (
    const sensor_msgs::msg::Joy::SharedPtr msg ) [private]
```

This function handles callbacks from control node. It will update received values from controller node.

### 8.5.3.4 HandleGuiNodeCallbacks()

```
void EthercatLifeCycle::HandleGuiNodeCallbacks (
    const std_msgs::msg::UInt8::SharedPtr gui_sub ) [private]
```

This function will handle values from GUI node. Updates parameters based on GUI node inputs.

### 8.5.3.5 InitEthercatCommunication()

```
int EthercatLifeCycle::InitEthercatCommunication ( ) [private]
```

Encapsulates all configuration steps for the EtherCAT communication with default slaves. And waits for connected slaves to become operational.

#### Returns

0 if succesful otherwise -1.

### 8.5.3.6 on\_activate()

```
node_interfaces::LifecycleNodeInterface::CallbackReturn EthercatLifeCycle::on_activate (
    const State & ) [private]
```

Activates Ethercat lifecycle node and starts real-time Ethercat communication. All publishing is done in real-time loop in this active state.

#### Returns

Success if activation succesfull,otherwise FAILURE

#### 8.5.3.7 on\_cleanup()

```
node_interfaces::LifecycleNodeInterface::CallbackReturn EthercatLifeCycle::on_cleanup (
    const State & ) [private]
```

Cleans up all variables and datas assigned by Ethercat lifecycle node.

##### Returns

Success if cleanup succesfull,otherwise FAILURE

#### 8.5.3.8 on\_configure()

```
node_interfaces::LifecycleNodeInterface::CallbackReturn EthercatLifeCycle::on_configure (
    const State & ) [private]
```

Ethercat lifecycle node configuration function, node will start with this function For more information about Lifecyclenode and it's interfaces check below link : [https://design.ros2.org/articles/node\\_lifecycle.html](https://design.ros2.org/articles/node_lifecycle.html).

##### See also

`node_interfaces::LifecycleNodeInterface::CallbackReturn`

##### Returns

Success if configuration succesfull,otherwise FAILURE

#### 8.5.3.9 on\_deactivate()

```
node_interfaces::LifecycleNodeInterface::CallbackReturn EthercatLifeCycle::on_deactivate (
    const State & ) [private]
```

Deactivates Ethercat lifecycle node, turns of real-time communication.

##### Returns

Success if deactivation succesfull,otherwise FAILURE

#### 8.5.3.10 on\_error()

```
node_interfaces::LifecycleNodeInterface::CallbackReturn EthercatLifeCycle::on_error (
    const State & ) [private]
```

There isn't any error recovery functionality for this node, just resets nodes. Reconfiguration is needed for restarting communication.

##### Returns

Success

#### 8.5.3.11 on\_shutdown()

```
node_interfaces::LifecycleNodeInterface::CallbackReturn EthercatLifeCycle::on_shutdown (
    const State & ) [private]
```

Shuts down EtherCAT lifecycle node, releases Ethercat master.

##### Returns

Success if shut down succesfull,otherwise FAILURE

#### 8.5.3.12 PassCyclycExchange()

```
void * EthercatLifeCycle::PassCyclycExchange (
    void * arg ) [static], [private]
```

Helper function to enter pthread\_create, since pthread's are C function it doesn't accept class member function, to pass class member function this helper function is created.

##### Parameters

<i>arg</i>	Pointer to current class instance.
------------	------------------------------------

##### Returns

void\*

#### 8.5.3.13 PublishAllData()

```
int EthercatLifeCycle::PublishAllData ( ) [private]
```

Publishes all data that master received and will be sent.

**Returns**

0 if succesfull otherwise -1.

**8.5.3.14 ReadFromSlaves()**

```
void EthercatLifeCycle::ReadFromSlaves ( ) [private]
```

Reads data from slaves and updates received data structure to be published.

**8.5.3.15 SetComThreadPriorities()**

```
int EthercatLifeCycle::SetComThreadPriorities ( ) [private]
```

Sets Ethercat communication thread's properties After this function called user must call [StartEthercatCommunication\(\)](#) function].

**Returns**

0 if succesfull, otherwise -1.

**8.5.3.16 StartEthercatCommunication()**

```
int EthercatLifeCycle::StartEthercatCommunication ( ) [private]
```

Starts EtherCAT communcation.

**Returns**

0 if succesfull, otherwise -1.

**8.5.3.17 StartPdoExchange()**

```
void EthercatLifeCycle::StartPdoExchange (
    void * instance ) [private]
```

Realtime cyclic Pdo exchange function which will constantly read/write values from/to slaves.

**Parameters**

<i>arg</i>	Used during pthread_create function to pass variables to realtime task.
------------	---

**Returns**

NULL

**8.5.3.18 UpdateCyclicPositionModeParameters()**

```
void EthercatLifeCycle::UpdateCyclicPositionModeParameters ( ) [private]
```

Acquired data from subscribed controller topic will be assigned as motor cyclic target position parameter in configured interpolation time.

**8.5.3.19 UpdateMotorStatePositionMode()**

```
void EthercatLifeCycle::UpdateMotorStatePositionMode ( ) [private]
```

Updates motor control word and motor state in position mode based on CIA402 state machine,.

**8.5.3.20 UpdateMotorStateVelocityMode()**

```
void EthercatLifeCycle::UpdateMotorStateVelocityMode ( ) [private]
```

Updates motor control word and motor state in velocity mode based on CIA402.

**8.5.3.21 UpdatePositionModeParameters()**

```
void EthercatLifeCycle::UpdatePositionModeParameters ( ) [private]
```

Acquired data from subscribed controller topic will be assigned as motor target position parameter.

**8.5.3.22 UpdateVelocityModeParameters()**

```
void EthercatLifeCycle::UpdateVelocityModeParameters ( ) [private]
```

Acquired data from subscribed controller topic will be assigned as motor speed parameter.

#### 8.5.3.23 WriteToSlavesInPositionMode()

```
void EthercatLifeCycle::WriteToSlavesInPositionMode ( ) [private]
```

Writes target position and control word to motor in profile position mode.

#### 8.5.3.24 WriteToSlavesVelocityMode()

```
void EthercatLifeCycle::WriteToSlavesVelocityMode ( ) [private]
```

Updates data that will be sent to slaves. This updated data will be published as well.

### 8.5.4 Member Data Documentation

#### 8.5.4.1 al\_state\_

```
uint8_t EthercatLifeCycleNode::EthercatLifeCycle::al_state_ = 0 [private]
```

Application layer of slaves seen by master.(INIT/PREOP/SAFEOP/OP)

#### 8.5.4.2 command\_

```
uint32_t EthercatLifeCycleNode::EthercatLifeCycle::command_ = 0x004F [private]
```

#### 8.5.4.3 controller\_

```
Controller EthercatLifeCycleNode::EthercatLifeCycle::controller_ [private]
```

#### 8.5.4.4 ecat\_node\_

```
std::unique_ptr<EthercatNode> EthercatLifeCycleNode::EthercatLifeCycle::ecat_node_ [private]
```



#### 8.5.4.5 emergency\_status\_

```
uint8_t EthercatLifeCycleNode::EthercatLifeCycle::emergency_status_ = 1 [private]
```

#### 8.5.4.6 err\_

```
int32_t EthercatLifeCycleNode::EthercatLifeCycle::err_ [private]
```

#### 8.5.4.7 ethercat\_sched\_param\_

```
struct sched_param EthercatLifeCycleNode::EthercatLifeCycle::ethercat_sched_param_ = {} [private]
```

#### 8.5.4.8 ethercat\_thread\_

```
pthread_t EthercatLifeCycleNode::EthercatLifeCycle::ethercat_thread_ [private]
```

pthread create required parameters.

#### 8.5.4.9 ethercat\_thread\_attr\_

```
pthread_attr_t EthercatLifeCycleNode::EthercatLifeCycle::ethercat_thread_attr_ [private]
```

#### 8.5.4.10 gui\_node\_data\_

```
uint8_t EthercatLifeCycleNode::EthercatLifeCycle::gui_node_data_ = 1 [private]
```

Values will be sent by controller node and will be assigned to variables below.

#### 8.5.4.11 gui\_subscriber\_

```
rcldcpp::Subscription<std_msgs::msg::UInt8>::SharedPtr EthercatLifeCycleNode::EthercatLifeCycle::gui_subscriber_ [private]
```

#### 8.5.4.12 joystick\_subscriber\_

```
rclcpp::Subscription<sensor_msgs::msg::Joy>::SharedPtr EthercatLifeCycleNode::EthercatLifeCycle::joystick_subscriber_ [private]
```

This subscriber will be used to receive data from controller node.

#### 8.5.4.13 measurement\_time

```
std::int32_t EthercatLifeCycleNode::EthercatLifeCycle::measurement_time = 0 [private]
```

#### 8.5.4.14 memory\_strategy

```
rclcpp::memory_strategy::MemoryStrategy::SharedPtr EthercatLifeCycleNode::EthercatLifeCycle::memory_strategy [private]
```

##### Initial value:

```
=
    std::make_shared<AllocatorMemoryStrategy<TLSFAllocator<void>>>()
```

#### 8.5.4.15 motor\_state\_

```
uint32_t EthercatLifeCycleNode::EthercatLifeCycle::motor_state_[g_kNumberOfServoDrivers] [private]
```

#### 8.5.4.16 received\_data\_

```
ecat_msgs::msg::DataReceived EthercatLifeCycleNode::EthercatLifeCycle::received_data_ [private]
```

#### 8.5.4.17 received\_data\_publisher\_

```
LifecyclePublisher<ecat_msgs::msg::DataReceived>::SharedPtr EthercatLifeCycleNode::EthercatLifeCycle::received_data_publisher_ [private]
```

This lifecycle publisher will be used to publish received feedback data from slaves.

#### 8.5.4.18 sent\_data\_

```
ecat_msgs::msg::DataSent EthercatLifeCycleNode::EthercatLifeCycle::sent_data_ [private]
```

#### 8.5.4.19 sent\_data\_publisher\_

```
LifecyclePublisher<ecat_msgs::msg::DataSent>::SharedPtr EthercatLifeCycleNode::EthercatLifeCycle::sent_data_publisher_ [private]
```

This lifecycle publisher will be used to publish sent data from master to slaves.

#### 8.5.4.20 timer\_

```
rclcpp::TimerBase::SharedPtr EthercatLifeCycleNode::EthercatLifeCycle::timer_ [private]
```

#### 8.5.4.21 timer\_info\_

```
Timing EthercatLifeCycleNode::EthercatLifeCycle::timer_info_ [private]
```

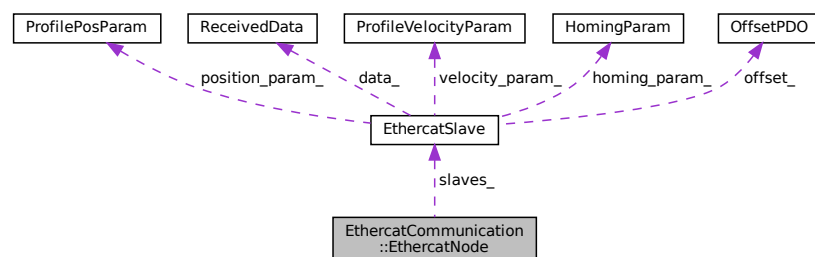
The documentation for this class was generated from the following files:

- [/home/veysiadm/spinerobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/ecat\\_lifecycle.hpp](#)
- [/home/veysiadm/spinerobot\\_ws/src/ecat\\_pkg/src/ecat\\_lifecycle.cpp](#)

## 8.6 EthercatCommunication::EthercatNode Class Reference

```
#include <ecat_node.hpp>
```

Collaboration diagram for EthercatCommunication::EthercatNode:



## Public Member Functions

- [EthercatNode](#) ()
- [~EthercatNode](#) ()
- int [ConfigureMaster](#) ()  
*Requests master instance and creates a domain for a master.*
- void [DefineDefaultSlaves](#) ()  
*Defines default connected slaves based on number of slaves. Specifies its position, vendor id , product code etc. Default connected slaves considered implementation specific. In our case it will be 3 motors and one EasyCAT slave.*
- void [SetCustomSlave](#) ([EthercatSlave](#) c\_slave, int position)  
*Passes your defined slave to [EthercatNode](#) class.*
- int [ConfigureSlaves](#) ()  
*Obtains slave configuration for all slaves w.r.t master.*
- int [SetProfilePositionParameters](#) ([ProfilePosParam](#) &P, int position)  
*Set mode to ProfilePositionMode with specified parameters for servo drive on that position.*
- int [SetProfilePositionParametersAll](#) ([ProfilePosParam](#) &P)  
*Set the mode to ProfilePositionMode with specified Parameters for all servo drives on the bus.*
- int [SetProfileVelocityParameters](#) ([ProfileVelocityParam](#) &P, int position)  
*Set mode to ProfileVelocityMode with specified parameters for servo drive on that position.*
- int [SetProfileVelocityParametersAll](#) ([ProfileVelocityParam](#) &P)  
*Set mode to ProfileVelocityMode with specified parameters for all servo drives on the bus.*
- int [SetCyclicSyncPositionModeParameters](#) ([CSPositionModeParam](#) &P, int position)  
*Set the Cyclic Sync Position Mode Parameters for slave in specified physical position w.r.t. master.*
- int [SetCyclicSyncPositionModeParametersAll](#) ([CSPositionModeParam](#) &P)  
*Sets the Cyclic Synchronous Position Mode Parameters for all connected motor driver slaves.*
- int [MapDefaultPdos](#) ()  
*Maps default PDOs for our spine surgery robot implementation.*
- int [MapCustomPdos](#) ([ec\\_sync\\_info\\_t](#) \*syncs, [ec\\_pdo\\_entry\\_reg\\_t](#) \*pdo\_entry\_reg, int position)  
*Map Custom PDO based on your PDO mappings.*
- void [ConfigDcSyncDefault](#) ()  
*Configures DC sync for our default configuration.*
- void [ConfigDcSync](#) ([uint16\\_t](#) assign\_activate, int position)  
*Configures DC synchronization for specified slave position.*
- void [CheckSlaveConfigurationState](#) ()  
*This function will check slave's application layer states. (INIT/PREOP/SAFEOP/OP)*
- int [CheckMasterState](#) ()  
*This function will check master's state, in terms of number of responding slaves and their application layer states.*
- void [CheckMasterDomainState](#) ()  
*Reads the state of a domain. Stores the domain state in the given state structure. Using this method, the process data exchange can be monitored in realtime.*
- int [ActivateMaster](#) ()  
*Activates master, after this function call realtime operation can start.*
- int [RegisterDomain](#) ()  
*Registers domain for each slave. This method has to be called after [ecrt\\_master\\_activate\(\)](#) to get the mapped domain process data memory.*
- int [WaitForOperationalMode](#) ()  
*Puts all slave to operational mode. User must call this before entering real-time operation. Reason for this function is that, master and slave has to do several exchange before becoming operational. So this function does exchange between master and slaves for up to 10 sec, could finish earlier. If timeout occurs it will return -1.*
- int [OpenEthercatMaster](#) ()  
*Opens EtherCAT master via command line tool if it's not already on.*
- int [GetNumberOfConnectedSlaves](#) ()

*Get the Number Of physically Connected Slaves to the bus.And checks if specified NUM\_OF\_SLAVES is correct.*

- void [GetAllSlaveInformation](#) ()

*Get the information of physically connected slaves to the master. This function will return connected slave's vendor id, product code.*

- void [DeactivateCommunication](#) ()

*Deactivates slaves and can be called in real-time.*

- void [ReleaseMaster](#) ()

*Deactivates and releases master shouldn't be called in real-time.*

- int [ShutDownEthercatMaster](#) ()

*Shutdowns EtherCAT master via command line tool if it's not already off.*

## Public Attributes

- [EthercatSlave slaves\\_](#) [NUM\_OF\_SLAVES]

## Private Attributes

- int [fd](#)

*File descriptor to open and wake master from CLI.*

## 8.6.1 Constructor & Destructor Documentation

### 8.6.1.1 EthercatNode()

```
EthercatNode::EthercatNode ( )
```

### 8.6.1.2 ~EthercatNode()

```
EthercatNode::~~EthercatNode ( )
```

## 8.6.2 Member Function Documentation

### 8.6.2.1 ActivateMaster()

```
int EthercatNode::ActivateMaster ( )
```

Activates master, after this function call realtime operation can start.

#### Warning

Before activating master all configuration should be done

After calling this function you have to register domain(s) and start realtime task.

#### Returns

0 if succesful, otherwise -1.

### 8.6.2.2 CheckMasterDomainState()

```
void EthercatNode::CheckMasterDomainState ( )
```

Reads the state of a domain. Stores the domain state in the given state structure. Using this method, the process data exchange can be monitored in realtime.

### 8.6.2.3 CheckMasterState()

```
int EthercatNode::CheckMasterState ( )
```

This function will check master's state, in terms of number of responding slaves and their application layer states.

#### Returns

0 if succesful, otherwise -1

#### See also

ec\_master\_state\_t structure.

### 8.6.2.4 CheckSlaveConfigurationState()

```
void EthercatNode::CheckSlaveConfigurationState ( )
```

This function will check slave's application layer states. (INIT/PREOP/SAFEOP/OP)

### 8.6.2.5 ConfigDcSync()

```
void EthercatNode::ConfigDcSync (
    uint16_t assign_activate,
    int position )
```

Configures DC synchronization for specified slave position.

## Parameters

<i>assign_activate</i>	Activating DC synchronization for slave. 0x300 for Elmo   and same for EasyCAT
------------------------	--

## Note

Assign activate parameters specified in slaves ESI file

## Parameters

<i>position</i>	
-----------------	--

### 8.6.2.6 ConfigDcSyncDefault()

```
void EthercatNode::ConfigDcSyncDefault ( )
```

Configures DC sync for our default configuration.

### 8.6.2.7 ConfigureMaster()

```
int EthercatNode::ConfigureMaster ( )
```

Requests master instance and creates a domain for a master.

## Note

Keep in mind that created master and domain are global variables.

## Returns

0 if succesful otherwise -1.

### 8.6.2.8 ConfigureSlaves()

```
int EthercatNode::ConfigureSlaves ( )
```

Obtains slave configuration for all slaves w.r.t master.

## Returns

0 if succesfull, otherwise -1.

### 8.6.2.9 DeactivateCommunication()

```
void EthercatNode::DeactivateCommunication ( )
```

Deactivates slaves and can be called in real-time.

### 8.6.2.10 DefineDefaultSlaves()

```
void EthercatCommunication::EthercatNode::DefineDefaultSlaves ( )
```

Defines default connected slaves based on number of slaves. Specifies its position, vendor id , product code etc. Default connected slaves considered implementation specific. In our case it will be 3 motors and one EasyCAT slave.

### 8.6.2.11 GetAllSlaveInformation()

```
void EthercatNode::GetAllSlaveInformation ( )
```

Get the information of physically connected slaves to the master. This function will return connected slave's vendor id, product code.

### 8.6.2.12 GetNumberOfConnectedSlaves()

```
int EthercatNode::GetNumberOfConnectedSlaves ( )
```

Get the Number Of physically Connected Slaves to the bus.And checks if specified NUM\_OF\_SLAVES is correct.

#### Returns

0 if NUM\_OF\_SLAVES setting is correct, otherwise -1.

### 8.6.2.13 MapCustomPdos()

```
int EthercatNode::MapCustomPdos (
    ec_sync_info_t * syncs,
    ec_pdo_entry_reg_t * pdo_entry_reg,
    int position )
```

Map Custom PDO based on your PDO mappings.

#### Note

You have to specify slave syncs and slave pdo registers before using function



## Parameters

<i>S</i>	<a href="#">EthercatSlave</a> instance
<i>position</i>	Physical position of your slave w.r.t master

## Returns

0 if succesfull, -1 otherwise.

#### 8.6.2.14 MapDefaultPdos()

```
int EthercatNode::MapDefaultPdos ( )
```

Maps default PDOs for our spine surgery robot implementation.

## Note

This method is specific for our spinnerobot implementation. If you have different topology or different servo drives use

## See also

[MapCustomPdos\(\)](#) function of modify this function based on your needs.

## Returns

0 if succesfull, otherwise -1.

This part is specific for our Custom EASYCAT slave configuration To create your custom slave and variables you can add variables to

## See also

[OffsetPDO](#) struct. Also you have add your variables to received data structure, you may have to create your custom msg files as well.

#### 8.6.2.15 OpenEthercatMaster()

```
int EthercatNode::OpenEthercatMaster ( )
```

Opens EtherCAT master via command line tool if it's not already on.

## Returns

0 if succesfull, otherwise -1.

### 8.6.2.16 RegisterDomain()

```
int EthercatNode::RegisterDomain ( )
```

Registers domain for each slave. This method has to be called after `ecrt_master_activate()` to get the mapped domain process data memory.

#### Returns

0 if succesful , otherwise -1

### 8.6.2.17 ReleaseMaster()

```
void EthercatNode::ReleaseMaster ( )
```

Deactivates and releases master shouldn't be called in real-time.

### 8.6.2.18 SetCustomSlave()

```
void EthercatNode::SetCustomSlave (
    EthercatSlave c_slave,
    int position )
```

Passes your defined slave to [EthercatNode](#) class.

#### Parameters

<i>c_slave</i>	first create your own <a href="#">EthercatSlave</a> instance and modify it then pass it to configuration.
<i>position</i>	specify the physical connection position for your custom configured slave.

### 8.6.2.19 SetCyclicSyncPositionModeParameters()

```
int EthercatNode::SetCyclicSyncPositionModeParameters (
    CSPositionModeParam & P,
    int position )
```

Set the Cyclic Sync Position Mode Parameters for slave in specified physical position w.r.t. master.

#### Parameters

<i>P</i>	Cyclic Sync. Position Mode Parameters.
<i>position</i>	Physical position of slave to be configured

**Returns**

0 if sucessfull, otherwise -1.

**8.6.2.20 SetCyclicSyncPositionModeParametersAll()**

```
int EthercatNode::SetCyclicSyncPositionModeParametersAll (
    CSPositionModeParam & P )
```

Sets the Cyclic Synchronous Position Mode Parameters for all connected motor driver slaves.

**Returns**

0 if sucessful, otherwise -1.

**8.6.2.21 SetProfilePositionParameters()**

```
int EthercatNode::SetProfilePositionParameters (
    ProfilePosParam & P,
    int position )
```

Set mode to ProfilePositionMode with specified parameters for servo drive on that position.

**Parameters**

<i>P</i>	Profile position parameter structure specified by user.
<i>position</i>	Slave position

**Returns**

0 if succesfull, otherwise -1.

**8.6.2.22 SetProfilePositionParametersAll()**

```
int EthercatNode::SetProfilePositionParametersAll (
    ProfilePosParam & P )
```

Set the mode to ProfilePositionMode with specified Parameters for all servo drives on the bus.

**Parameters**

<i>P</i>	Profile position parameter structure specified by user.
----------	---

**Returns**

0 if succesful, otherwise -1.

**8.6.2.23 SetProfileVelocityParameters()**

```
int EthercatNode::SetProfileVelocityParameters (
    ProfileVelocityParam & P,
    int position )
```

Set mode to ProfileVelocityMode with specified parameters for servo drive on that position.

**Parameters**

<i>P</i>	Profile velocity parameter structure specified by user.
<i>position</i>	Slave position

**Returns**

0 if succesful, -1 otherwise.

**8.6.2.24 SetProfileVelocityParametersAll()**

```
int EthercatNode::SetProfileVelocityParametersAll (
    ProfileVelocityParam & P )
```

Set mode to ProfileVelocityMode with specified parameters for all servo drives on the bus.

**Parameters**

<i>P</i>	Profile velocity parameter structure specified by user.
----------	---

**Returns**

0 if succesfull, -1 otherwise.

**Todo** Add error code to all functions. Instead of returning -1.

**8.6.2.25 ShutDownEthercatMaster()**

```
int EthercatNode::ShutDownEthercatMaster ( )
```

Shutdowns EtherCAT master via command line tool if it's not already off.

**Returns**

0 if succesfull, otherwise -1.

**8.6.2.26 WaitForOperationalMode()**

```
int EthercatNode::WaitForOperationalMode ( )
```

Puts all slave to operational mode. User must call this before entering real-time operation. Reason for this function is that, master and slave has to do several exchange before becoming operational. So this function does exchange between master and slaves for up to 10 sec, could finish earlier. If timeout occurs it will return -1.

**Returns**

0 if succesfull, otherwise -1.

**8.6.3 Member Data Documentation****8.6.3.1 fd**

```
int EthercatCommunication::EthercatNode::fd [private]
```

File descriptor to open and wake master from CLI.

**8.6.3.2 slaves\_**

```
EthercatSlave EthercatCommunication::EthercatNode::slaves_[NUM_OF_SLAVES]
```

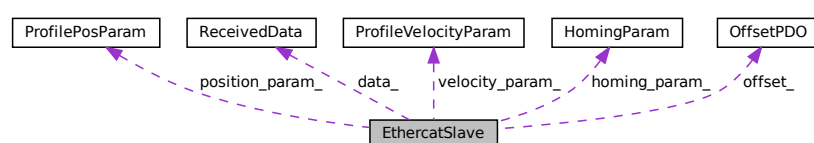
The documentation for this class was generated from the following files:

- /home/veysiadm/spinerobot\_ws/src/ecat\_pkg/include/ecat\_pkg/ecat\_node.hpp
- /home/veysiadm/spinerobot\_ws/src/ecat\_pkg/src/ecat\_node.cpp

**8.7 EthercatSlave Class Reference**

```
#include <ecat_slave.hpp>
```

Collaboration diagram for EthercatSlave:



## Public Member Functions

- [EthercatSlave](#) ()
- [~EthercatSlave](#) ()
- [int CheckSlaveConfigState](#) ()

*This function will check slave's application layer states. (INIT/PREOP/SAFEOP/OP)*

## Public Attributes

- [ec\\_slave\\_config\\_t](#) \* [slave\\_config\\_](#)  
*Slave configuration parameters, assigned to each slave.*
- [ec\\_slave\\_config\\_state\\_t](#) [slave\\_config\\_state\\_](#)  
*Slave state handle to check if slave is online and slaves state machine status(INIT/PREOP/SAFEOP/OP)*
- [uint8\\_t](#) \* [slave\\_pdo\\_domain\\_](#)  
*PDO domain for data exchange.*
- [int32\\_t](#) [motor\\_state\\_](#)  
*Variable for checking motor state.*
- [ec\\_slave\\_info\\_t](#) [slave\\_info\\_](#)  
*Slave information data structure. This structure contains all information related to slave. It will be used to get slave's information from master.*
- [OffsetPDO](#) [offset\\_](#)  
*Offset for PDO entries to assign pdo registers.*
- [ReceivedData](#) [data\\_](#)  
*Received data from servo drivers.*
- [ProfileVelocityParam](#) [velocity\\_param\\_](#)  
*Slave velocity parameters.*
- [ProfilePosParam](#) [position\\_param\\_](#)  
*Slave position parameters.*
- [HomingParam](#) [homing\\_param\\_](#)

## Static Public Attributes

- [const static uint32\\_t](#) [kSync0\\_shift\\_](#) = 0  
*DC sync shift setting, zero will give best synchronization.*

## 8.7.1 Constructor & Destructor Documentation

### 8.7.1.1 EthercatSlave()

```
EthercatSlave::EthercatSlave ( )
```

### 8.7.1.2 ~EthercatSlave()

```
EthercatSlave::~~EthercatSlave ( )
```

## 8.7.2 Member Function Documentation

### 8.7.2.1 CheckSlaveConfigState()

```
int EthercatSlave::CheckSlaveConfigState ( )
```

This function will check slave's application layer states. (INIT/PREOP/SAFEOP/OP)

#### Note

This function shouldn't be called in real time context. For diagnosis you can use CheckDomainState() encapsulation in ecat\_node.

#### Returns

0 if succesful.

## 8.7.3 Member Data Documentation

### 8.7.3.1 data\_

```
ReceivedData EthercatSlave::data_
```

Received data from servo drivers.

### 8.7.3.2 homing\_param\_

```
HomingParam EthercatSlave::homing_param_
```

### 8.7.3.3 kSync0\_shift\_

```
const static uint32_t EthercatSlave::kSync0_shift_ = 0 [static]
```

DC sync shift setting, zero will give best synchronization.

#### 8.7.3.4 motor\_state\_

```
int32_t EthercatSlave::motor_state_
```

Variable for checking motor state.

#### 8.7.3.5 offset\_

```
OffsetPDO EthercatSlave::offset_
```

Offset for PDO entries to assign pdo registers.

#### 8.7.3.6 position\_param\_

```
ProfilePosParam EthercatSlave::position_param_
```

Slave position parameters.

#### 8.7.3.7 slave\_config\_

```
ec_slave_config_t* EthercatSlave::slave_config_
```

Slave configuration parameters, assigned to each slave.

#### 8.7.3.8 slave\_config\_state\_

```
ec_slave_config_state_t EthercatSlave::slave_config_state_
```

Slave state handle to check if slave is online and slaves state machine status(INIT/PREOP/SAFEOP/OP)

#### 8.7.3.9 slave\_info\_

```
ec_slave_info_t EthercatSlave::slave_info_
```

Slave information data structure. This structure contains all information related to slave. It will be used to get slave's information from master.



### 8.7.3.10 slave\_pdo\_domain\_

```
uint8_t* EthercatSlave::slave_pdo_domain_
```

PDO domain for data exchange.

### 8.7.3.11 velocity\_param\_

```
ProfileVelocityParam EthercatSlave::velocity_param_
```

Slave velocity parameters.

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

- [/home/veysiadm/spinrobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/ecat\\_slave.hpp](#)
- [/home/veysiadm/spinrobot\\_ws/src/ecat\\_pkg/src/ecat\\_slave.cpp](#)

## 8.8 HomingParam Struct Reference

Parameters that should be specified in homing mode.

```
#include <ecat_globals.hpp>
```

### Public Attributes

- uint32\_t [max\\_fol\\_err](#)
- uint32\_t [max\\_profile\\_vel](#)
- uint32\_t [quick\\_stop\\_dec](#)
- uint32\_t [speed\\_for\\_switch\\_search](#)
- uint32\_t [speed\\_for\\_zero\\_search](#)
- uint32\_t [homing\\_acc](#)
- uint16\_t [curr\\_threshold\\_homing](#)  
*Used when homing by touching mechanical limit and sensing current.*
- int32\_t [home\\_offset](#)  
*Amount to move away from the sensed limit*
- int8\_t [homing\\_method](#)

### 8.8.1 Detailed Description

Parameters that should be specified in homing mode.

### 8.8.2 Member Data Documentation

### 8.8.2.1 curr\_threshold\_homing

```
uint16_t HomingParam::curr_threshold_homing
```

Used when homing by touching mechanical limit and sensing current.

### 8.8.2.2 home\_offset

```
int32_t HomingParam::home_offset
```

Amount to move away from the sensed limit

### 8.8.2.3 homing\_acc

```
uint32_t HomingParam::homing_acc
```

### 8.8.2.4 homing\_method

```
int8_t HomingParam::homing_method
```

### 8.8.2.5 max\_fol\_err

```
uint32_t HomingParam::max_fol_err
```

### 8.8.2.6 max\_profile\_vel

```
uint32_t HomingParam::max_profile_vel
```

### 8.8.2.7 quick\_stop\_dec

```
uint32_t HomingParam::quick_stop_dec
```

### 8.8.2.8 speed\_for\_switch\_search

```
uint32_t HomingParam::speed_for_switch_search
```

### 8.8.2.9 speed\_for\_zero\_search

```
uint32_t HomingParam::speed_for_zero_search
```

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

- [/home/veysiadr/spinrobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/ecat\\_globals.hpp](#)

## 8.9 OffsetPDO Struct Reference

offset for PDO entries to register PDOs.

```
#include <ecat_globals.hpp>
```

### Public Attributes

- uint32\_t [target\\_pos](#)
- uint32\_t [target\\_vel](#)
- uint32\_t [target\\_tor](#)
- uint32\_t [max\\_tor](#)
- uint32\_t [control\\_word](#)
- uint32\_t [op\\_mode](#)
- uint32\_t [profile\\_acc](#)
- uint32\_t [profile\\_dec](#)
- uint32\_t [quick\\_stop\\_dec](#)
- uint32\_t [profile\\_vel](#)
- uint32\_t [actual\\_pos](#)
- uint32\_t [pos\\_fol\\_err](#)
- uint32\_t [actual\\_vel](#)
- uint32\_t [actual\\_cur](#)
- uint32\_t [actual\\_tor](#)
- uint32\_t [status\\_word](#)
- uint32\_t [op\\_mode\\_display](#)
- uint32\_t [error\\_code](#)
- uint32\_t [extra\\_status\\_reg](#)
- uint32\_t [r\\_limit\\_switch](#)
- uint32\_t [l\\_limit\\_switch](#)
- uint32\_t [emergency\\_switch](#)

### 8.9.1 Detailed Description

offset for PDO entries to register PDOs.

## 8.9.2 Member Data Documentation

### 8.9.2.1 actual\_cur

`uint32_t OffsetPDO::actual_cur`

### 8.9.2.2 actual\_pos

`uint32_t OffsetPDO::actual_pos`

### 8.9.2.3 actual\_tor

`uint32_t OffsetPDO::actual_tor`

### 8.9.2.4 actual\_vel

`uint32_t OffsetPDO::actual_vel`

### 8.9.2.5 control\_word

`uint32_t OffsetPDO::control_word`

### 8.9.2.6 emergency\_switch

`uint32_t OffsetPDO::emergency_switch`

### 8.9.2.7 error\_code

`uint32_t OffsetPDO::error_code`

#### 8.9.2.8 extra\_status\_reg

uint32\_t OffsetPDO::extra\_status\_reg

#### 8.9.2.9 l\_limit\_switch

uint32\_t OffsetPDO::l\_limit\_switch

#### 8.9.2.10 max\_tor

uint32\_t OffsetPDO::max\_tor

#### 8.9.2.11 op\_mode

uint32\_t OffsetPDO::op\_mode

#### 8.9.2.12 op\_mode\_display

uint32\_t OffsetPDO::op\_mode\_display

#### 8.9.2.13 pos\_fol\_err

uint32\_t OffsetPDO::pos\_fol\_err

#### 8.9.2.14 profile\_acc

uint32\_t OffsetPDO::profile\_acc

#### 8.9.2.15 profile\_dec

uint32\_t OffsetPDO::profile\_dec

#### 8.9.2.16 profile\_vel

```
uint32_t OffsetPDO::profile_vel
```

#### 8.9.2.17 quick\_stop\_dec

```
uint32_t OffsetPDO::quick_stop_dec
```

#### 8.9.2.18 r\_limit\_switch

```
uint32_t OffsetPDO::r_limit_switch
```

#### 8.9.2.19 status\_word

```
uint32_t OffsetPDO::status_word
```

#### 8.9.2.20 target\_pos

```
uint32_t OffsetPDO::target_pos
```

#### 8.9.2.21 target\_tor

```
uint32_t OffsetPDO::target_tor
```

#### 8.9.2.22 target\_vel

```
uint32_t OffsetPDO::target_vel
```

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

- [/home/veysiadrn/spinrobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/ecat\\_globals.hpp](#)

## 8.10 ProfilePosParam Struct Reference

Parameters that should be specified in position mode.

```
#include <ecat_globals.hpp>
```

### Public Attributes

- uint32\_t [profile\\_vel](#)
- uint32\_t [profile\\_acc](#)
- uint32\_t [profile\\_dec](#)
- uint32\_t [max\\_fol\\_err](#)
- uint32\_t [max\\_profile\\_vel](#)
- uint32\_t [quick\\_stop\\_dec](#)
- uint16\_t [motion\\_profile\\_type](#)

### 8.10.1 Detailed Description

Parameters that should be specified in position mode.

### 8.10.2 Member Data Documentation

#### 8.10.2.1 max\_fol\_err

```
uint32_t ProfilePosParam::max_fol_err
```

#### 8.10.2.2 max\_profile\_vel

```
uint32_t ProfilePosParam::max_profile_vel
```

#### 8.10.2.3 motion\_profile\_type

```
uint16_t ProfilePosParam::motion_profile_type
```

#### 8.10.2.4 profile\_acc

```
uint32_t ProfilePosParam::profile_acc
```

#### 8.10.2.5 profile\_dec

```
uint32_t ProfilePosParam::profile_dec
```

#### 8.10.2.6 profile\_vel

```
uint32_t ProfilePosParam::profile_vel
```

#### 8.10.2.7 quick\_stop\_dec

```
uint32_t ProfilePosParam::quick_stop_dec
```

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

- [/home/veysiadm/spinerobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/ecat\\_globals.hpp](#)

## 8.11 ProfileVelocityParam Struct Reference

Parameters that should be specified in velocity mode.

```
#include <ecat_globals.hpp>
```

### Public Attributes

- uint32\_t [max\\_profile\\_vel](#)
- uint32\_t [quick\\_stop\\_dec](#)
- uint32\_t [profile\\_acc](#)
- uint32\_t [profile\\_dec](#)
- uint16\_t [motion\\_profile\\_type](#)

#### 8.11.1 Detailed Description

Parameters that should be specified in velocity mode.



## 8.11.2 Member Data Documentation

### 8.11.2.1 max\_profile\_vel

`uint32_t ProfileVelocityParam::max_profile_vel`

### 8.11.2.2 motion\_profile\_type

`uint16_t ProfileVelocityParam::motion_profile_type`

### 8.11.2.3 profile\_acc

`uint32_t ProfileVelocityParam::profile_acc`

### 8.11.2.4 profile\_dec

`uint32_t ProfileVelocityParam::profile_dec`

### 8.11.2.5 quick\_stop\_dec

`uint32_t ProfileVelocityParam::quick_stop_dec`

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

- `/home/veysiadr/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_globals.hpp`

## 8.12 ReceivedData Struct Reference

Received feedback data from slaves.

```
#include <ecat_globals.hpp>
```

## Public Attributes

- `int32_t` [target\\_pos](#)
- `int32_t` [target\\_vel](#)
- `int16_t` [target\\_tor](#)
- `int16_t` [max\\_tor](#)
- `uint16_t` [control\\_word](#)
- `OpMode` [op\\_mode](#)
- `int32_t` [vel\\_offset](#)
- `int16_t` [tor\\_offset](#)
- `int32_t` [actual\\_pos](#)
- `int32_t` [actual\\_vel](#)
- `int16_t` [actual\\_cur](#)
- `int16_t` [actual\\_tor](#)
- `uint16_t` [status\\_word](#)
- `int8_t` [op\\_mode\\_display](#)
- `uint8_t` [left\\_limit\\_switch\\_val](#)
- `uint8_t` [right\\_limit\\_switch\\_val](#)
- `uint8_t` [s\\_emergency\\_switch\\_val](#)

### 8.12.1 Detailed Description

Received feedback data from slaves.

### 8.12.2 Member Data Documentation

#### 8.12.2.1 `actual_cur`

```
int16_t ReceivedData::actual_cur
```

#### 8.12.2.2 `actual_pos`

```
int32_t ReceivedData::actual_pos
```

#### 8.12.2.3 `actual_tor`

```
int16_t ReceivedData::actual_tor
```

#### 8.12.2.4 actual\_vel

`int32_t ReceivedData::actual_vel`

#### 8.12.2.5 control\_word

`uint16_t ReceivedData::control_word`

#### 8.12.2.6 left\_limit\_switch\_val

`uint8_t ReceivedData::left_limit_switch_val`

#### 8.12.2.7 max\_tor

`int16_t ReceivedData::max_tor`

#### 8.12.2.8 op\_mode

`OpMode ReceivedData::op_mode`

#### 8.12.2.9 op\_mode\_display

`int8_t ReceivedData::op_mode_display`

#### 8.12.2.10 right\_limit\_switch\_val

`uint8_t ReceivedData::right_limit_switch_val`

#### 8.12.2.11 s\_emergency\_switch\_val

`uint8_t ReceivedData::s_emergency_switch_val`

#### 8.12.2.12 status\_word

```
uint16_t ReceivedData::status_word
```

#### 8.12.2.13 target\_pos

```
int32_t ReceivedData::target_pos
```

#### 8.12.2.14 target\_tor

```
int16_t ReceivedData::target_tor
```

#### 8.12.2.15 target\_vel

```
int32_t ReceivedData::target_vel
```

#### 8.12.2.16 tor\_offset

```
int16_t ReceivedData::tor_offset
```

#### 8.12.2.17 vel\_offset

```
int32_t ReceivedData::vel_offset
```

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

- [/home/veysiadr/spinrobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/ecat\\_globals.hpp](#)

## 8.13 SdoRequest Struct Reference

EtherCAT SDO request structure for configuration phase.

```
#include <ecat_globals.hpp>
```

## Public Attributes

- `ec_sdo_request` \* [profile\\_acc](#)
- `ec_sdo_request` \* [profile\\_dec](#)
- `ec_sdo_request` \* [profile\\_vel](#)
- `ec_sdo_request` \* [quick\\_stop\\_dec](#)
- `ec_sdo_request` \* [motion\\_profile\\_type](#)
- `ec_sdo_request` \* [max\\_profile\\_vel](#)
- `ec_sdo_request` \* [max\\_fol\\_err](#)
- `ec_sdo_request` \* [speed\\_for\\_switch\\_search](#)
- `ec_sdo_request` \* [speed\\_for\\_zero\\_search](#)
- `ec_sdo_request` \* [homing\\_acc](#)
- `ec_sdo_request` \* [curr\\_threshold\\_homing](#)
- `ec_sdo_request` \* [home\\_offset](#)
- `ec_sdo_request` \* [homing\\_method](#)

### 8.13.1 Detailed Description

EtherCAT SDO request structure for configuration phase.

### 8.13.2 Member Data Documentation

#### 8.13.2.1 `curr_threshold_homing`

```
ec_sdo_request* SdoRequest::curr_threshold_homing
```

#### 8.13.2.2 `home_offset`

```
ec_sdo_request* SdoRequest::home_offset
```

#### 8.13.2.3 `homing_acc`

```
ec_sdo_request* SdoRequest::homing_acc
```

#### 8.13.2.4 `homing_method`

```
ec_sdo_request* SdoRequest::homing_method
```

**8.13.2.5 max\_fol\_err**

ec\_sdo\_request\* SdoRequest::max\_fol\_err

**8.13.2.6 max\_profile\_vel**

ec\_sdo\_request\* SdoRequest::max\_profile\_vel

**8.13.2.7 motion\_profile\_type**

ec\_sdo\_request\* SdoRequest::motion\_profile\_type

**8.13.2.8 profile\_acc**

ec\_sdo\_request\* SdoRequest::profile\_acc

**8.13.2.9 profile\_dec**

ec\_sdo\_request\* SdoRequest::profile\_dec

**8.13.2.10 profile\_vel**

ec\_sdo\_request\* SdoRequest::profile\_vel

**8.13.2.11 quick\_stop\_dec**

ec\_sdo\_request\* SdoRequest::quick\_stop\_dec

**8.13.2.12 speed\_for\_switch\_search**

ec\_sdo\_request\* SdoRequest::speed\_for\_switch\_search

### 8.13.2.13 speed\_for\_zero\_search

```
ec_sdo_request* SdoRequest::speed_for_zero_search
```

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

- [/home/veysiadm/spineroobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/ecat\\_globals.hpp](/home/veysiadm/spineroobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_globals.hpp)

## 8.14 Timing Class Reference

```
#include <timing.hpp>
```

### Public Member Functions

- void [GetTime](#) ()
- void [MeasureTimeDifference](#) ()
- void [OutInfoToFile](#) ()

### Public Attributes

- std::chrono::high\_resolution\_clock::time\_point [timer\\_start\\_](#)
- std::chrono::high\_resolution\_clock::time\_point [last\\_start\\_time\\_](#)
- std::chrono::duration< long, std::micro > [time\\_span](#)
- std::vector< long > [timing\\_info\\_](#) = std::vector<long>(NUMBER\_OF\_SAMPLES)
- uint32\_t [counter\\_](#) = 0

### 8.14.1 Member Function Documentation

#### 8.14.1.1 GetTime()

```
void Timing::GetTime ( )
```

#### 8.14.1.2 MeasureTimeDifference()

```
void Timing::MeasureTimeDifference ( )
```

#### 8.14.1.3 OutInfoToFile()

```
void Timing::OutInfoToFile ( )
```

## 8.14.2 Member Data Documentation

### 8.14.2.1 counter\_

```
uint32_t Timing::counter_ = 0
```

### 8.14.2.2 last\_start\_time\_

```
std::chrono::high_resolution_clock::time_point Timing::last_start_time_
```

### 8.14.2.3 time\_span

```
std::chrono::duration<long, std::micro> Timing::time_span
```

### 8.14.2.4 timer\_start\_

```
std::chrono::high_resolution_clock::time_point Timing::timer_start_
```

### 8.14.2.5 timing\_info\_

```
std::vector<long> Timing::timing_info_ = std::vector<long>(NUMBER_OF_SAMPLES)
```

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

- [/home/veysiadrn/spinrobot\\_ws/src/ecat\\_pkg/include/ecat\\_pkg/timing.hpp](#)
- [/home/veysiadrn/spinrobot\\_ws/src/ecat\\_pkg/src/timing.cpp](#)



## Chapter 9

# File Documentation

### 9.1 /home/veysiadr/spinerobot\_ws/src/ecat\_pkg/CMakeLists.txt File Reference

#### Functions

- `cmake_minimum_required` (VERSION 3.5) `project(ecat_pkg)` `if(NOT CMAKE_C_STANDARD)` `set(CMAKE_C_STANDARD 99)` `endif()` `if(NOT CMAKE_CXX_STANDARD)` `set(CMAKE_CXX_STANDARD 14)` `endif()` `add_compile_options(-g -w -Wall -Wextra -Wpedantic -I/opt/etherlab/include -L/opt/etherlab/lib -lethercat -lpthread -lrt -Wl`
- `rpath` `opt etherlab lib` `set(etherlab_include/opt/etherlab/include)` `set(etherlab_lib/opt/etherlab/lib/libethercat.so.1.1.0)` `set(ecat_node_include ~/spinerobot_ws/src/ecat_pkg/include/ecat_pkg/)` `set(node_name "ecat_node")` `find_package(ament_cmake REQUIRED)` `find_package(rclcpp REQUIRED)` `find_package(rclcpp_lifecycle REQUIRED)` `find_package(rttest)` `find_package(tlsf_cpp)` `find_package(ecat_msgs REQUIRED)` `find_package(sensor_msgs REQUIRED)` `add_executable(ecat_node src/main.cpp src/ecat_node.cpp src/ecat_slave.cpp src/ecat_lifecycle.cpp src/timing.cpp)` `target_include_directories(ecat_node PUBLIC $<BUILD_INTERFACE`

#### Variables

- `rpath` `Wl`

#### 9.1.1 Function Documentation

##### 9.1.1.1 `cmake_minimum_required()`

```
cmake_minimum_required (
    VERSION 3.5 )
```

### 9.1.1.2 set()

```
rpath opt etherlab lib set (
    etherlab_include/opt/etherlab/ include )
```

## 9.1.2 Variable Documentation

### 9.1.2.1 Wl

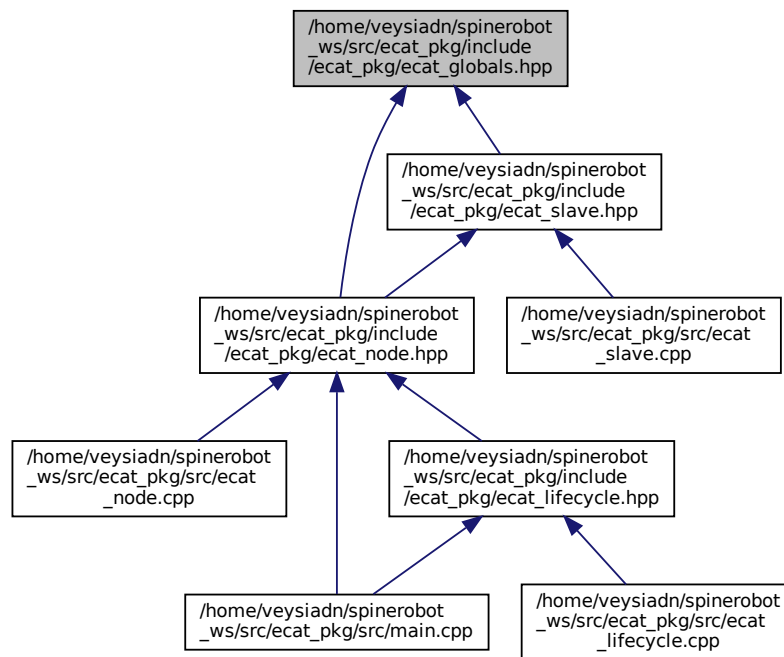
```
rpath Wl
```

## 9.2 /home/veysiadm/spinerobot\_ws/src/ecat\_pkg/include/ecat\_pkg/ecat\_globals.hpp File Reference

```
#include <iostream>
#include <cstring>
#include <limits.h>
#include <stdlib.h>
#include <pthread.h>
#include <errno.h>
#include <signal.h>
#include <stdio.h>
#include <string.h>
#include <sys/resource.h>
#include <sys/time.h>
#include <sys/types.h>
#include <unistd.h>
#include <time.h>
#include <sys/mman.h>
#include <malloc.h>
#include <sched.h>
#include <chrono>
#include <memory>
#include "ecrt.h"
#include "object_dictionary.hpp"
Include dependency graph for ecat_globals.hpp:
```



This graph shows which files directly or indirectly include this file:



## Classes

- struct [Controller](#)  
*Controller Parameters.*
- struct [OffsetPDO](#)  
*offset for PDO entries to register PDOs.*
- struct [ReceivedData](#)  
*Received feedback data from slaves.*
- struct [SdoRequest](#)  
*EtherCAT SDO request structure for configuration phase.*
- struct [ProfilePosParam](#)  
*Parameters that should be specified in position mode.*
- struct [CSPositionModeParam](#)  
*Struct contains configuration parameters for cyclic sync. position mode.*
- struct [CSVelocityModeParam](#)  
*Struct contains configuration parameters for cyclic sync. velocity mode.*
- struct [CSTorqueModeParam](#)  
*Struct contains configuration parameters for cyclic sync. torque mode.*
- struct [HomingParam](#)  
*Parameters that should be specified in homing mode.*
- struct [ProfileVelocityParam](#)  
*Parameters that should be specified in velocity mode.*

## Macros

- `#define NUM_OF_SLAVES 2`  
*Object dictionary paramaters PDO index and default values in here.*
- `#define FREQUENCY 1000`  
*Ethercat PDO exchange loop frequency in Hz.*
- `#define MEASURE_TIMING 1`  
*If you want to measure timings leave it as one, otherwise make it 0.*
- `#define VELOCITY_MODE 0`  
*set this to 1 if you want to use it in velocity mode (and set position mode 0)*
- `#define POSITION_MODE 0`  
*set this to 1 if you want to use it in position mode (and set velocity mode 0)*
- `#define CYCLIC_POSITION_MODE 1`  
*set this to 1 if you want to use it in cyclic synchronous position mode (and set velocity mode 0)*
- `#define GEAR_RATIO 103`  
*Define your own motor properties to configure increments in position mode.*
- `#define ENCODER_RESOLUTION 1024`
- `#define INC_PER_ROTATION GEAR_RATIO*ENCODER_RESOLUTION*4`
- `#define FIVE_DEGREE_CCW int(INC_PER_ROTATION/72)`
- `#define THIRTY_DEGREE_CCW int(INC_PER_ROTATION/12)`
- `#define PERIOD_NS (g_kNsPerSec/FREQUENCY)`  
*EtherCAT communication period in nanoseconds.*
- `#define PERIOD_US (PERIOD_NS / 1000)`
- `#define PERIOD_MS (PERIOD_US / 1000)`
- `#define FINAL_SLAVE (NUM_OF_SLAVES-1)`
- `#define TEST_BIT(NUM, N) ((NUM & (1 << N))>>N)`  
*To sync every cycle.*
- `#define SET_BIT(NUM, N) (NUM | (1 << N))`
- `#define RESET_BIT(NUM, N) (NUM & ~(1 << N))`
- `#define TIMESPEC2NS(T) ((uint64_t) (T).tv_sec * g_kNsPerSec + (T).tv_nsec)`
- `#define DIFF_NS(A, B) (((B).tv_sec - (A).tv_sec) * g_kNsPerSec + (B).tv_nsec - (A).tv_nsec)`
- `#define CLOCK_TO_USE CLOCK_MONOTONIC`

## Enumerations

- `enum OpMode {`  
`kProfilePosition = 1, kProfileVelocity = 3, kProfileTorque = 4, kHoming = 6,`  
`kInterpolatedPosition = 7, kCSPosition = 8, kCSVelocity = 9, kCSTorque = 10 }`  
*Motor operation modes.*
- `enum MotorStates {`  
`kReadyToSwitchOn = 1, kSwitchedOn, kOperationEnabled, kFault,`  
`kVoltageEnabled, kQuickStop, kSwitchOnDisabled, kWarning,`  
`kRemote, kTargetReached, kInternalLimitActivate }`  
*CIA 402 state machine motor states.*
- `enum ErrorRegisterBits {`  
`kGenericError = 0, kCurrentError, kVoltageError, kTemperatureError,`  
`kCommunicationError, kDeviceProfileSpecificError, kReserved, kMotionError }`
- `enum SensorConfig {`  
`kSensor1TypeNone = 0, kSensor1TypeDigitalIncrementalEncoder1 = 1, kSensor2TypeNone = 0, kSensor2TypeDigitalIncrementalEncoder2 = 256,`  
`kSensor2TypeAnalogIncrementalEncoderSinCos = 512, kSensor2TypeSSIAbsoluteEncoder = 768, kSensor3TypeNone = 0, kSensor3TypeDigitalHallSensor = 131072 }`  
*Sensor Configuration for motor for more information.*

- enum `ControlStructureBits` {  
`kCurrentControlStructure` = 0, `kVelocityControlStructure` = 4, `kPositionControlStructure` = 8, `kGearLocation` = 12,  
`kProcessValueReference` = 14, `kMainSensor` = 16, `kAuxiliarySensor` = 20, `kMountingPositionSensor1` = 24,  
`kMountingPositionSensor2` = 26, `kMountingPositionSensor3` = 28 }

*Control structure configuration for control mechanism to select sensor structure specific to hardware.*

## Functions

- struct timespec `timespec_add` (struct timespec time1, struct timespec time2)

*Add two timespec struct.*

## Variables

- const uint32\_t `g_kNumberOfServoDrivers` = 1  
*Number of connected servo drives.*
- const uint32\_t `g_kNsPerSec` = 1000000000  
*Nanoseconds per second.*
- static volatile sig\_atomic\_t `sig` = 1
- ec\_master\_t \* `g_master`  
*Extern global variable declaration.*
- ec\_master\_state\_t `g_master_state`  
*EtherCAT master.*
- ec\_domain\_t \* `g_master_domain`  
*EtherCAT master state.*
- ec\_domain\_state\_t `g_master_domain_state`  
*Ethercat data passing master domain.*
- struct timespec `g_sync_timer`  
*EtherCAT master domain state.*
- const struct timespec `g_cycle_time` = {0, PERIOD\_NS}  
*timer for DC sync .*
- uint32\_t `g_sync_ref_counter`  
*cycletime settings in ns.*

## 9.2.1 Macro Definition Documentation

### 9.2.1.1 CLOCK\_TO\_USE

```
#define CLOCK_TO_USE CLOCK_MONOTONIC
```

### 9.2.1.2 CYCLIC\_POSITION\_MODE

```
#define CYCLIC_POSITION_MODE 1
```

set this to 1 if you want to use it in cyclic synchronous position mode (and set velocity mode 0)

### 9.2.1.3 DIFF\_NS

```
#define DIFF_NS(  
    A,  
    B ) (((B).tv_sec - (A).tv_sec) * g_kNsPerSec + (B).tv_nsec - (A).tv_nsec)
```

### 9.2.1.4 ENCODER\_RESOLUTION

```
#define ENCODER_RESOLUTION 1024
```

### 9.2.1.5 FINAL\_SLAVE

```
#define FINAL_SLAVE (NUM_OF_SLAVES-1)
```

### 9.2.1.6 FIVE\_DEGREE\_CCW

```
#define FIVE_DEGREE_CCW int(INC_PER_ROTATION/72)
```

### 9.2.1.7 FREQUENCY

```
#define FREQUENCY 1000
```

Ethercat PDO exchange loop frequency in Hz.

### 9.2.1.8 GEAR\_RATIO

```
#define GEAR_RATIO 103
```

Define your own motor properties to configure increments in position mode.

### 9.2.1.9 INC\_PER\_ROTATION

```
#define INC_PER_ROTATION GEAR_RATIO*ENCODER_RESOLUTION*4
```

### 9.2.1.10 MEASURE\_TIMING

```
#define MEASURE_TIMING 1
```

If you want to measure timings leave it as one, otherwise make it 0.

### 9.2.1.11 NUM\_OF\_SLAVES

```
#define NUM_OF_SLAVES 2
```

Object dictionary paramaters PDO index and default values in here.

IgH EtherCAT library header file the user-space real-time interface library. IgH, EtherCAT related functions and data types. USER SHOULD DEFINE THIS AREAS Total number of connected slave to the bus.

### 9.2.1.12 PERIOD\_MS

```
#define PERIOD_MS (PERIOD_US / 1000)
```

### 9.2.1.13 PERIOD\_NS

```
#define PERIOD_NS (g_kNsPerSec/FREQUENCY)
```

EtherCAT communication period in nanoseconds.

### 9.2.1.14 PERIOD\_US

```
#define PERIOD_US (PERIOD_NS / 1000)
```

### 9.2.1.15 POSITION\_MODE

```
#define POSITION_MODE 0
```

set this to 1 if you want to use it in position mode (and set velocity mode 0)

### 9.2.1.16 RESET\_BIT

```
#define RESET_BIT(
    NUM,
    N ) (NUM & ~(1 << N))
```

### 9.2.1.17 SET\_BIT

```
#define SET_BIT(
    NUM,
    N ) (NUM | (1 << N))
```

### 9.2.1.18 TEST\_BIT

```
#define TEST_BIT(
    NUM,
    N ) ((NUM & (1 << N))>>N)
```

To sync every cycle.

### 9.2.1.19 THIRTY\_DEGREE\_CCW

```
#define THIRTY_DEGREE_CCW int(INC_PER_ROTATION/12)
```

### 9.2.1.20 TIMESPEC2NS

```
#define TIMESPEC2NS(
    T ) ((uint64_t) (T).tv_sec * g_kNsPerSec + (T).tv_nsec)
```

### 9.2.1.21 VELOCITY\_MODE

```
#define VELOCITY_MODE 0
```

set this to 1 if you want to use it in velocity mode (and set position mode 0)

## 9.2 Enumeration Type Documentation

### 9.2.2.1 ControlStructureBits

```
enum ControlStructureBits
```

Control structure configuration for control mechanism to select sensor structure specific to hardware.

See also

EPOS4-Firmware-Specification pg. 140



**Enumerator**

kCurrentControlStructure	These are bit locations not values for values.  See also  EPOS4-Firmware-Specification pg. 140 !!!
kVelocityControlStructure	0-3 , 4 bits. Val : 1 - PI current controller
kPositionControlStructure	4-7, 4bits. Val : 0 - None   1 - PI Vecolity controller (low pass filter)   2 - PI velocity controller (observer)
kGearLocation	8-11 , 4bits. Val : 0 - None   1 - PID position controller
kProcessValueReference	1 bit Val : 0 - None   1 - Gear Mounted on system
kMainSensor	14-15 2 bits. Val : 0 - On motor (or undefined)   1 - On gear
kAuxiliarySensor	16-19 4 bits. Val : 0 - None   1 - Sensor 1   2 - Sensor 2   3 - Sensor 3
kMountingPositionSensor1	20-23 4 bits. Val : 0 - None   1 - Sensor 1   2 - Sensor 2   3 - Sensor 3
kMountingPositionSensor2	24-25 2 bits. Val : 0 - On motor (or undefined)   1 - On gear
kMountingPositionSensor3	26-27 2 bits. Val : 0 - On motor (or undefined)   1 - On gear

**9.2.2.2 ErrorRegisterBits**

```
enum ErrorRegisterBits
```

**Enumerator**

kGenericError	
kCurrentError	
kVoltageError	
kTemperatureError	
kCommunicationError	
kDeviceProfileSpecificError	
kReserved	
kMotionError	

**9.2.2.3 MotorStates**

```
enum MotorStates
```

CIA 402 state machine motor states.

**Enumerator**

kReadyToSwitchOn	
kSwitchedOn	

**Enumerator**

kOperationEnabled	
kFault	
kVoltageEnabled	
kQuickStop	
kSwitchOnDisabled	
kWarning	
kRemote	
kTargetReached	
kInternalLimitActivate	

**9.2.2.4 OpMode**

enum [OpMode](#)

Motor operation modes.

**Enumerator**

kProfilePosition	
kProfileVelocity	
kProfileTorque	
kHoming	
kInterpolatedPosition	
kCSPosition	
kCSVelocity	
kCSTorque	

**9.2.2.5 SensorConfig**

enum [SensorConfig](#)

Sensor Configuration for motor for more information.

**See also**

EPOS4-Firmware-Specification Pg.138

**Enumerator**

kSensor1TypeNone	
kSensor1TypeDigitalIncrementalEncoder1	
kSensor2TypeNone	

## Enumerator

kSensor2TypeDigitalIncrementalEncoder2	
kSensor2TypeAnalogIncrementalEncoderSinCos	
kSensor2TypeSSIAbsoluteEncoder	
kSensor3TypeNone	
kSensor3TypeDigitalHallSensor	

## 9.2.3 Function Documentation

### 9.2.3.1 timespec\_add()

```
struct timespec timespec_add (
    struct timespec time1,
    struct timespec time2 ) [inline]
```

Add two timespec struct.

## Parameters

<i>time1</i>	Timespec struct 1
<i>time2</i>	Timespec struct 2

## Returns

Addition result

## 9.2.4 Variable Documentation

### 9.2.4.1 g\_cycle\_time

```
const struct timespec g_cycle_time = {0, PERIOD_NS}
```

timer for DC sync .

### 9.2.4.2 g\_kNsPerSec

```
const uint32_t g_kNsPerSec = 1000000000
```

Nanoseconds per second.

#### 9.2.4.3 g\_kNumberOfServoDrivers

```
const uint32_t g_kNumberOfServoDrivers = 1
```

Number of connected servo drives.

#### 9.2.4.4 g\_master

```
ec_master_t* g_master
```

Extern global variable declaration.

#### 9.2.4.5 g\_master\_domain

```
ec_domain_t* g_master_domain
```

EtherCAT master state.

#### 9.2.4.6 g\_master\_domain\_state

```
ec_domain_state_t g_master_domain_state
```

Ethercat data passing master domain.

#### 9.2.4.7 g\_master\_state

```
ec_master_state_t g_master_state
```

EtherCAT master.

#### 9.2.4.8 g\_sync\_ref\_counter

```
uint32_t g_sync_ref_counter
```

cycletime settings in ns.

#### 9.2.4.9 g\_sync\_timer

```
struct timespec g_sync_timer
```

EtherCAT master domain state.

#### 9.2.4.10 sig

```
volatile sig_atomic_t sig = 1 [static]
```

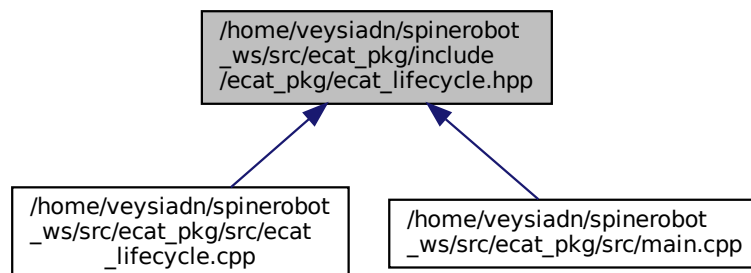
### 9.3 /home/veysiadm/spinerobot\_ws/src/ecat\_pkg/include/ecat\_pkg/ecat\_lifecycle.hpp File Reference

```
#include "ecat_node.hpp"
#include "timing.hpp"
#include <rclcpp_lifecycle/lifecycle_node.hpp>
#include <rclcpp_lifecycle/lifecycle_publisher.hpp>
#include "std_msgs/msg/u_int8.hpp"
#include "ecat_msgs/msg/data_received.hpp"
#include "ecat_msgs/msg/data_sent.hpp"
#include <rclcpp/strategies/message_pool_memory_strategy.hpp>
#include <rclcpp/strategies/allocator_memory_strategy.hpp>
#include <rttest/rttest.h>
#include <tlsf_cpp/tlsf.hpp>
```

Include dependency graph for ecat\_lifecycle.hpp:

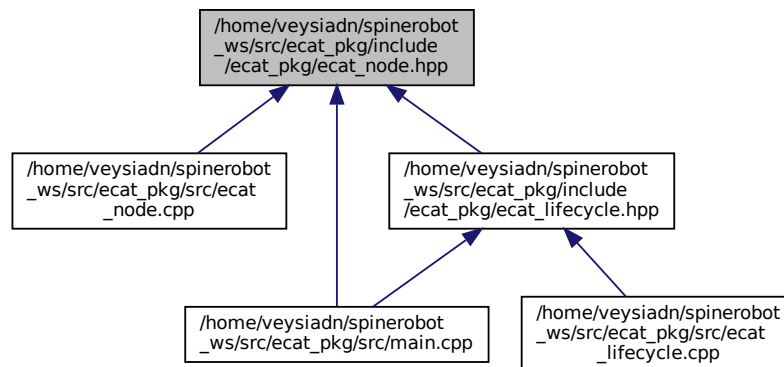


This graph shows which files directly or indirectly include this file:





This graph shows which files directly or indirectly include this file:



## Classes

- class `EthercatCommunication::EthercatNode`

## Namespaces

- EthercatCommunication

### ROS2 Headers.

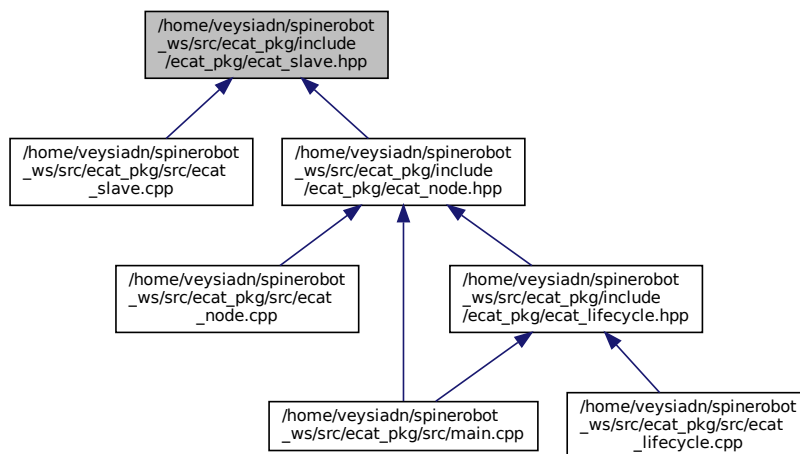
## 9.5 /home/veysiadm/spinerobot\_ws/src/ecat\_pkg/include/ecat\_pkg/ecat\_↵ \_slave.hpp File Reference

```
#include "ecat_globals.hpp"
```

Include dependency graph for ecat\_slave.hpp:



This graph shows which files directly or indirectly include this file:



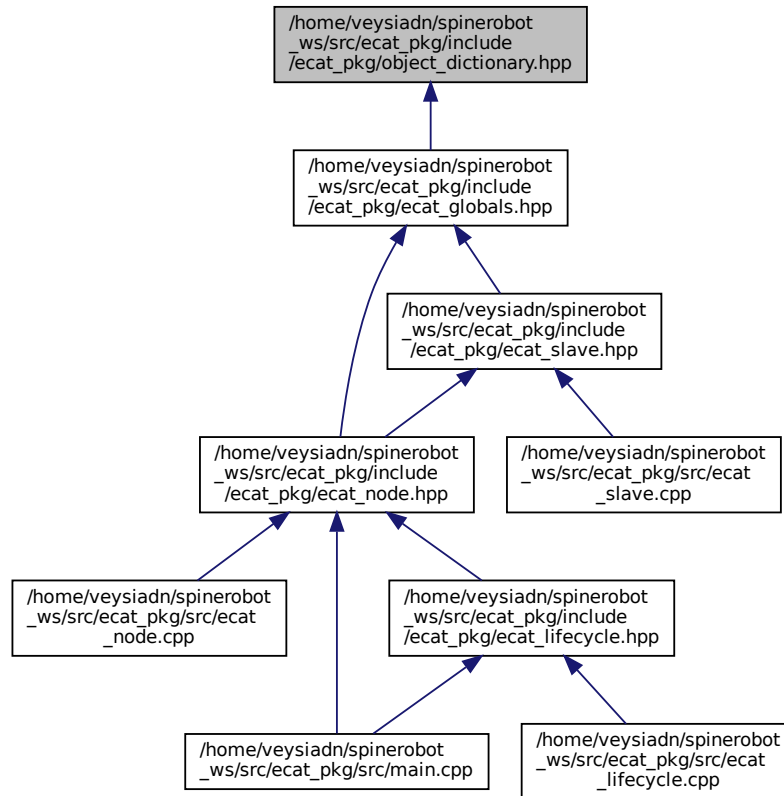
## Classes

- class [EthercatSlave](#)



## 9.6 /home/veysiadm/spinerobot\_ws/src/ecat\_pkg/include/ecat\_pkg/object\_dictionary.hpp File Reference

This graph shows which files directly or indirectly include this file:



### Macros

- `#define OD_CONTROL_WORD 0x6040,0x00`
- `#define OD_STATUS_WORD 0x6041,0x00`
- `#define OD_OPERATION_MODE 0x6060,0x00`
- `#define OD_OPERATION_MODE_DISPLAY 0x6061,0x00`
- `#define OD_TARGET_POSITION 0x607A,0x00`
- `#define OD_POSITION_ACTUAL_VAL 0x6064,0x00`
- `#define OD_POSITION_DEMAND 0x6062,0x00`
- `#define OD_POSITON_FOLLOWING_ERROR 0x60F4,0X00`
- `#define OD_POSITONCOUNTS 0x6063,0x00`
- `#define OD_MAX_FOLLOWING_ERROR 0x6065,0x00`
- `#define OD_VELOCITY_ACTUAL_VALUE 0x606C,0x00`
- `#define OD_VELOCITY_ADDRESS 0x6069,0x00`
- `#define OD_TARGET_VELOCITY 0x60FF,0x00`
- `#define OD_VELOCITY_OFFSET 0x60B1,0x00`
- `#define OD_INTERPOLATION_TIME_PERIOD 0X60C2,0X01`

- #define OD\_INTERPOLATION\_TIME\_UNIT 0X60C2,0X02
- #define OD\_MOTOR\_DATA\_NOMINAL\_CURRENT 0X3001,0X01
- #define OD\_MOTOR\_DATA\_OUTPUT\_CURRENT\_LIMIT 0X3001,0X02
- #define OD\_MOTOR\_DATA\_NUMBER\_OF\_POLE\_PAIRS 0X3001,0X03
- #define OD\_MOTOR\_DATA\_THERMAL\_TIME\_CONSTANT\_WINDINGS 0X3001,0X04
- #define OD\_MOTOR\_DATA\_TORQUE\_CONSTANT 0X3001,0X05
- #define OD\_GEAR\_REDUCTION\_NUMERATOR 0X3003,0X01
- #define OD\_GEAR\_REDUCTION\_DENOMINATOR 0X3003,0X02
- #define OD\_GEAR\_MAX\_INPUT\_SPEED 0X3003,0X03
- #define OD\_GEAR\_MISC\_CONFIGURATION 0X3003,0X04
- #define OD\_DIGITAL\_INCREMENTAL\_ENCODER\_1\_TYPE 0X3010,0X02
- #define OD\_ERROR\_REGISTER 0X1001,0X00
- #define OD\_ACTIVE\_FIELDBUS 0X2010,0X00
- #define OD\_CUSTOM\_PERSISTENT\_MEMORY\_1 0X210C,0X01
- #define OD\_CUSTOM\_PERSISTENT\_MEMORY\_2 0X210C,0X02
- #define OD\_CUSTOM\_PERSISTENT\_MEMORY\_3 0X210C,0X03
- #define OD\_CUSTOM\_PERSISTENT\_MEMORY\_4 0X210C,0X04
- #define OD\_POWER\_SUPPLY\_VOLTAGE 0X2200,0X01
- #define OD\_SENSOR\_CONFIGURATION 0X3000,0X01

*Object Dictionary for Axis configurations.*

- #define OD\_CONTROL\_STRUCTURE 0X3000,0X02
- #define OD\_PROFILE\_VELOCITY 0x6081,0x00
- #define OD\_MAX\_PROFILE\_VELOCITY 0x6080,0x00
- #define OD\_MAX\_MOTOR\_SPEED 0X607F,0X00
- #define OD\_PROFILE\_ACCELERATION 0x6083,0x00
- #define OD\_PROFILE\_DECELERATION 0x6084,0x00
- #define OD\_QUICK\_STOP\_DECELERATION 0x6085,0x00
- #define OD\_MOTION\_PROFILE\_TYPE 0x6086,0x00
- #define OD\_LINEAR\_RAMP\_TRAPEZOIDAL 0x00,0x00
- #define OD\_VELOCITY\_ENCODER\_RESOLUTION\_NUM 0x6094,0x01
- #define OD\_VELOCITY\_ENCODER\_RESOLUTION\_DEN 0x6094,0x02
- #define OD\_DIGITAL\_INPUTS 0x60FD,0x00
- #define OD\_DIGITAL\_OUTPUTS 0x60FE,0x01
- #define OD\_DC\_CIRCUIT\_LINK\_VOLTAGE 0x6079,0x00
- #define OD\_TARGET\_TORQUE 0x6071,0x00
- #define OD\_TORQUE\_MAX 0x6072,0x00
- #define OD\_TORQUE\_ACTUAL\_VALUE 0x6077,0x00
- #define OD\_TORQUE\_OFFSET 0x60b2,0x00
- #define OD\_MAX\_CURRENT 0x6073, 0x00
- #define OD\_CURRENT\_ACTUAL\_VALUE 0x6078, 0x00
- #define OD\_ERROR\_CODE 0x603F, 0x00
- #define OD\_EXTRA\_STATUS\_REGISTER 0x2085, 0x00
- #define OD\_CHECK\_ERROR 0x1002,0x00
- #define OD\_QUICK\_STOP\_MODE 0x605A,0x00
- #define OD\_STOP\_OPTION\_CODE 0x605D,0x00
- #define SM\_COMM\_RESET 0x81
- #define SM\_FULL\_RESET 0x82
- #define SM\_START 0x01
- #define SM\_GO\_READY\_TO\_SWITCH\_ON 0x06
- #define SM\_GO\_SWITCH\_ON 0x07
- #define SM\_GO\_ENABLE 0X0F
- #define SM\_GO\_SWITCH\_ON\_DISABLE 0x00
- #define SM\_RUN 0x1F
- #define SM\_EXPEDITE 0x3F

- #define SM\_QUICKSTOP 0x02
- #define SM\_RELATIVE\_POS 0X7F
- #define SM\_FSAFROMSTATUSWORD(SW) (SW & 0x006f)
- #define SM\_NOT\_READY\_TO\_SWITCH\_ON 0b00000000
- #define SM\_NOT\_READY\_TO\_SWITCH\_ON\_2 0b00100000
- #define SM\_SWITCH\_ON\_DISABLED 0b01000000
- #define SM\_SWITCH\_ON\_DISABLED\_2 0b01100000
- #define SM\_READY\_TO\_SWITCH\_ON 0b00100001
- #define SM\_SWITCHED\_ON 0b00100011
- #define SM\_OPERATION\_ENABLED 0b00100111
- #define SM\_QUICK\_STOP\_ACTIVE 0b00000111
- #define SM\_FAULT\_REACTION\_ACTIVE 0b00001111
- #define SM\_FAULTREACTIONACTIVE2 0b00101111
- #define SM\_FAULT 0b00001000
- #define SM\_FAULT2 0b00101000
- #define SM\_SW\_READY\_TO\_SWITCH\_ON 0x0001
- #define SM\_SW\_SWITCHED\_ON 0x0002
- #define SM\_SW\_OPERATION\_ENABLED 0x0004
- #define SM\_SW\_FAULT 0x0008
- #define SM\_SW\_VOLTAGE\_ENABLED 0x0010
- #define SM\_SW\_QUICK\_STOP 0x0020
- #define SM\_SW\_SWITCH\_ON\_DISABLED 0x0040
- #define SM\_SW\_WARNING 0x0080
- #define SM\_SW\_REMOTE 0x0200
- #define SM\_SW\_TARGET\_REACHED 0x0400
- #define SM\_SW\_INTERNAL\_LIMIT\_ACTIVE 0x0800
- #define SM\_CW\_SWITCH\_ON 0x0001
- #define SM\_CW\_ENABLE\_VOLTAGE 0x0002
- #define SM\_CW\_QUICK\_STOP 0x0004
- #define SM\_CW\_ENABLE\_OPERATION 0x0008
- #define SM\_CW\_FAULT\_RESET 0x0080
- #define SM\_CW\_OD\_HALT 0x0100

## 9.6.1 Macro Definition Documentation

### 9.6.1.1 OD\_ACTIVE\_FIELDBUS

```
#define OD_ACTIVE_FIELDBUS 0X2010,0X00
```

### 9.6.1.2 OD\_CHECK\_ERROR

```
#define OD_CHECK_ERROR 0x1002,0x00
```

### 9.6.1.3 OD\_CONTROL\_STRUCTURE

```
#define OD_CONTROL_STRUCTURE 0X3000,0X02
```

### 9.6.1.4 OD\_CONTROL\_WORD

```
#define OD_CONTROL_WORD 0x6040,0x00
```

### 9.6.1.5 OD\_CURRENT\_ACTUAL\_VALUE

```
#define OD_CURRENT_ACTUAL_VALUE 0x6078, 0x00
```

### 9.6.1.6 OD\_CUSTOM\_PERSISTENT\_MEMORY\_1

```
#define OD_CUSTOM_PERSISTENT_MEMORY_1 0X210C,0X01
```

### 9.6.1.7 OD\_CUSTOM\_PERSISTENT\_MEMORY\_2

```
#define OD_CUSTOM_PERSISTENT_MEMORY_2 0X210C,0X02
```

### 9.6.1.8 OD\_CUSTOM\_PERSISTENT\_MEMORY\_3

```
#define OD_CUSTOM_PERSISTENT_MEMORY_3 0X210C,0X03
```

### 9.6.1.9 OD\_CUSTOM\_PERSISTENT\_MEMORY\_4

```
#define OD_CUSTOM_PERSISTENT_MEMORY_4 0X210C,0X04
```

### 9.6.1.10 OD\_DC\_CIRCUIT\_LINK\_VOLTAGE

```
#define OD_DC_CIRCUIT_LINK_VOLTAGE 0x6079,0x00
```

#### 9.6.1.11 OD\_DIGITAL\_INCREMENTAL\_ENCODER\_1\_TYPE

```
#define OD_DIGITAL_INCREMENTAL_ENCODER_1_TYPE 0X3010,0X02
```

#### 9.6.1.12 OD\_DIGITAL\_INPUTS

```
#define OD_DIGITAL_INPUTS 0x60FD,0x00
```

#### 9.6.1.13 OD\_DIGITAL\_OUTPUTS

```
#define OD_DIGITAL_OUTPUTS 0x60FE,0x01
```

#### 9.6.1.14 OD\_ERROR\_CODE

```
#define OD_ERROR_CODE 0x603F, 0x00
```

#### 9.6.1.15 OD\_ERROR\_REGISTER

```
#define OD_ERROR_REGISTER 0X1001,0X00
```

#### 9.6.1.16 OD\_EXTRA\_STATUS\_REGISTER

```
#define OD_EXTRA_STATUS_REGISTER 0x2085, 0x00
```

#### 9.6.1.17 OD\_GEAR\_MAX\_INPUT\_SPEED

```
#define OD_GEAR_MAX_INPUT_SPEED 0X3003,0X03
```

#### 9.6.1.18 OD\_GEAR\_MISC\_CONFIGURATION

```
#define OD_GEAR_MISC_CONFIGURATION 0X3003,0X04
```

#### 9.6.1.19 OD\_GEAR\_REDUCTION\_DENOMINATOR

```
#define OD_GEAR_REDUCTION_DENOMINATOR 0X3003,0X02
```

#### 9.6.1.20 OD\_GEAR\_REDUCTION\_NUMERATOR

```
#define OD_GEAR_REDUCTION_NUMERATOR 0X3003,0X01
```

#### 9.6.1.21 OD\_INTERPOLATION\_TIME\_PERIOD

```
#define OD_INTERPOLATION_TIME_PERIOD 0X60C2,0X01
```

#### 9.6.1.22 OD\_INTERPOLATION\_TIME\_UNIT

```
#define OD_INTERPOLATION_TIME_UNIT 0X60C2,0X02
```

#### 9.6.1.23 OD\_LINEAR\_RAMP\_TRAPEZOIDAL

```
#define OD_LINEAR_RAMP_TRAPEZOIDAL 0x00,0x00
```

#### 9.6.1.24 OD\_MAX\_CURRENT

```
#define OD_MAX_CURRENT 0x6073, 0x00
```

#### 9.6.1.25 OD\_MAX\_FOLLOWING\_ERROR

```
#define OD_MAX_FOLLOWING_ERROR 0x6065,0x00
```

#### 9.6.1.26 OD\_MAX\_MOTOR\_SPEED

```
#define OD_MAX_MOTOR_SPEED 0X607F,0X00
```

#### 9.6.1.27 OD\_MAX\_PROFILE\_VELOCITY

```
#define OD_MAX_PROFILE_VELOCITY 0x6080,0x00
```

#### 9.6.1.28 OD\_MOTION\_PROFILE\_TYPE

```
#define OD_MOTION_PROFILE_TYPE 0x6086,0x00
```

#### 9.6.1.29 OD\_MOTOR\_DATA\_NOMINAL\_CURRENT

```
#define OD_MOTOR_DATA_NOMINAL_CURRENT 0X3001,0X01
```

#### 9.6.1.30 OD\_MOTOR\_DATA\_NUMBER\_OF\_POLE\_PAIRS

```
#define OD_MOTOR_DATA_NUMBER_OF_POLE_PAIRS 0X3001,0X03
```

#### 9.6.1.31 OD\_MOTOR\_DATA\_OUTPUT\_CURRENT\_LIMIT

```
#define OD_MOTOR_DATA_OUTPUT_CURRENT_LIMIT 0X3001,0X02
```

#### 9.6.1.32 OD\_MOTOR\_DATA\_THERMAL\_TIME\_CONSTANT\_WINDINGS

```
#define OD_MOTOR_DATA_THERMAL_TIME_CONSTANT_WINDINGS 0X3001,0X04
```

#### 9.6.1.33 OD\_MOTOR\_DATA\_TORQUE\_CONSTANT

```
#define OD_MOTOR_DATA_TORQUE_CONSTANT 0X3001,0X05
```

#### 9.6.1.34 OD\_OPERATION\_MODE

```
#define OD_OPERATION_MODE 0x6060,0x00
```

**9.6.1.35 OD\_OPERATION\_MODE\_DISPLAY**

```
#define OD_OPERATION_MODE_DISPLAY 0x6061,0x00
```

**9.6.1.36 OD\_POSITION\_ACTUAL\_VAL**

```
#define OD_POSITION_ACTUAL_VAL 0x6064,0x00
```

**9.6.1.37 OD\_POSITION\_DEMAND**

```
#define OD_POSITION_DEMAND 0x6062,0x00
```

**9.6.1.38 OD\_POSITON\_FOLLOWING\_ERROR**

```
#define OD_POSITON_FOLLOWING_ERROR 0x60F4,0X00
```

**9.6.1.39 OD\_POSITONCOUNTS**

```
#define OD_POSITONCOUNTS 0x6063,0x00
```

**9.6.1.40 OD\_POWER\_SUPPLY\_VOLTAGE**

```
#define OD_POWER_SUPPLY_VOLTAGE 0X2200,0X01
```

**9.6.1.41 OD\_PROFILE\_ACCELERATION**

```
#define OD_PROFILE_ACCELERATION 0x6083,0x00
```

**9.6.1.42 OD\_PROFILE\_DECELERATION**

```
#define OD_PROFILE_DECELERATION 0x6084,0x00
```



#### 9.6.1.43 OD\_PROFILE\_VELOCITY

```
#define OD_PROFILE_VELOCITY 0x6081,0x00
```

#### 9.6.1.44 OD\_QUICK\_STOP\_DECELERATION

```
#define OD_QUICK_STOP_DECELERATION 0x6085,0x00
```

#### 9.6.1.45 OD\_QUICK\_STOP\_MODE

```
#define OD_QUICK_STOP_MODE 0x605A,0x00
```

#### 9.6.1.46 OD\_SENSOR\_CONFIGURATION

```
#define OD_SENSOR_CONFIGURATION 0X3000,0X01
```

Object Dictionary for Axis configurations.

#### 9.6.1.47 OD\_STATUS\_WORD

```
#define OD_STATUS_WORD 0x6041,0x00
```

#### 9.6.1.48 OD\_STOP\_OPTION\_CODE

```
#define OD_STOP_OPTION_CODE 0x605D,0x00
```

#### 9.6.1.49 OD\_TARGET\_POSITION

```
#define OD_TARGET_POSITION 0x607A,0x00
```

#### 9.6.1.50 OD\_TARGET\_TORQUE

```
#define OD_TARGET_TORQUE 0x6071,0x00
```

#### 9.6.1.51 OD\_TARGET\_VELOCITY

```
#define OD_TARGET_VELOCITY 0x60FF,0x00
```

#### 9.6.1.52 OD\_TORQUE\_ACTUAL\_VALUE

```
#define OD_TORQUE_ACTUAL_VALUE 0x6077,0x00
```

#### 9.6.1.53 OD\_TORQUE\_MAX

```
#define OD_TORQUE_MAX 0x6072,0x00
```

#### 9.6.1.54 OD\_TORQUE\_OFFSET

```
#define OD_TORQUE_OFFSET 0x60b2,0x00
```

#### 9.6.1.55 OD\_VELOCITY\_ACTUAL\_VALUE

```
#define OD_VELOCITY_ACTUAL_VALUE 0x606C,0x00
```

#### 9.6.1.56 OD\_VELOCITY\_ADDRESS

```
#define OD_VELOCITY_ADDRESS 0x6069,0x00
```

#### 9.6.1.57 OD\_VELOCITY\_ENCODER\_RESOLUTION\_DEN

```
#define OD_VELOCITY_ENCODER_RESOLUTION_DEN 0x6094,0x02
```

#### 9.6.1.58 OD\_VELOCITY\_ENCODER\_RESOLUTION\_NUM

```
#define OD_VELOCITY_ENCODER_RESOLUTION_NUM 0x6094,0x01
```

#### 9.6.1.59 OD\_VELOCITY\_OFFSET

```
#define OD_VELOCITY_OFFSET 0x60B1,0x00
```

#### 9.6.1.60 SM\_COMM\_RESET

```
#define SM_COMM_RESET 0x81
```

#### 9.6.1.61 SM\_CW\_ENABLE\_OPERATION

```
#define SM_CW_ENABLE_OPERATION 0x0008
```

#### 9.6.1.62 SM\_CW\_ENABLE\_VOLTAGE

```
#define SM_CW_ENABLE_VOLTAGE 0x0002
```

#### 9.6.1.63 SM\_CW\_FAULT\_RESET

```
#define SM_CW_FAULT_RESET 0x0080
```

#### 9.6.1.64 SM\_CW\_OD\_HALT

```
#define SM_CW_OD_HALT 0x0100
```

#### 9.6.1.65 SM\_CW\_QUICK\_STOP

```
#define SM_CW_QUICK_STOP 0x0004
```

#### 9.6.1.66 SM\_CW\_SWITCH\_ON

```
#define SM_CW_SWITCH_ON 0x0001
```

#### 9.6.1.67 SM\_EXPEDITE

```
#define SM_EXPEDITE 0x3F
```

#### 9.6.1.68 SM\_FAULT

```
#define SM_FAULT 0b00001000
```

#### 9.6.1.69 SM\_FAULT2

```
#define SM_FAULT2 0b00101000
```

#### 9.6.1.70 SM\_FAULT\_REACTION\_ACTIVE

```
#define SM_FAULT_REACTION_ACTIVE 0b00001111
```

#### 9.6.1.71 SM\_FAULTREACTIONACTIVE2

```
#define SM_FAULTREACTIONACTIVE2 0b00101111
```

#### 9.6.1.72 SM\_FSAFROMSTATUSWORD

```
#define SM_FSAFROMSTATUSWORD(  
    SW ) (SW & 0x006f)
```

#### 9.6.1.73 SM\_FULL\_RESET

```
#define SM_FULL_RESET 0x82
```

#### 9.6.1.74 SM\_GO\_ENABLE

```
#define SM_GO_ENABLE 0X0F
```

#### 9.6.1.75 SM\_GO\_READY\_TO\_SWITCH\_ON

```
#define SM_GO_READY_TO_SWITCH_ON 0x06
```

#### 9.6.1.76 SM\_GO\_SWITCH\_ON

```
#define SM_GO_SWITCH_ON 0x07
```

#### 9.6.1.77 SM\_GO\_SWITCH\_ON\_DISABLE

```
#define SM_GO_SWITCH_ON_DISABLE 0x00
```

#### 9.6.1.78 SM\_NOT\_READY\_TO\_SWITCH\_ON

```
#define SM_NOT_READY_TO_SWITCH_ON 0b00000000
```

#### 9.6.1.79 SM\_NOT\_READY\_TO\_SWITCH\_ON\_2

```
#define SM_NOT_READY_TO_SWITCH_ON_2 0b00100000
```

#### 9.6.1.80 SM\_OPERATION\_ENABLED

```
#define SM_OPERATION_ENABLED 0b00100111
```

#### 9.6.1.81 SM\_QUICK\_STOP\_ACTIVE

```
#define SM_QUICK_STOP_ACTIVE 0b00000111
```

#### 9.6.1.82 SM\_QUICKSTOP

```
#define SM_QUICKSTOP 0x02
```

#### 9.6.1.83 SM\_READY\_TO\_SWITCH\_ON

```
#define SM_READY_TO_SWITCH_ON 0b00100001
```

#### 9.6.1.84 SM\_RELATIVE\_POS

```
#define SM_RELATIVE_POS 0X7F
```

#### 9.6.1.85 SM\_RUN

```
#define SM_RUN 0x1F
```

#### 9.6.1.86 SM\_START

```
#define SM_START 0x01
```

#### 9.6.1.87 SM\_SW\_FAULT

```
#define SM_SW_FAULT 0x0008
```

#### 9.6.1.88 SM\_SW\_INTERNAL\_LIMIT\_ACTIVE

```
#define SM_SW_INTERNAL_LIMIT_ACTIVE 0x0800
```

#### 9.6.1.89 SM\_SW\_OPERATION\_ENABLED

```
#define SM_SW_OPERATION_ENABLED 0x0004
```

#### 9.6.1.90 SM\_SW\_QUICK\_STOP

```
#define SM_SW_QUICK_STOP 0x0020
```

#### 9.6.1.91 SM\_SW\_READY\_TO\_SWITCH\_ON

```
#define SM_SW_READY_TO_SWITCH_ON 0x0001
```

#### 9.6.1.92 SM\_SW\_REMOTE

```
#define SM_SW_REMOTE 0x0200
```

#### 9.6.1.93 SM\_SW\_SWITCH\_ON\_DISABLED

```
#define SM_SW_SWITCH_ON_DISABLED 0x0040
```

#### 9.6.1.94 SM\_SW\_SWITCHED\_ON

```
#define SM_SW_SWITCHED_ON 0x0002
```

#### 9.6.1.95 SM\_SW\_TARGET\_REACHED

```
#define SM_SW_TARGET_REACHED 0x0400
```

#### 9.6.1.96 SM\_SW\_VOLTAGE\_ENABLED

```
#define SM_SW_VOLTAGE_ENABLED 0x0010
```

### 9.6.1.97 SM\_SW\_WARNING

```
#define SM_SW_WARNING 0x0080
```

### 9.6.1.98 SM\_SWITCH\_ON\_DISABLED

```
#define SM_SWITCH_ON_DISABLED 0b01000000
```

### 9.6.1.99 SM\_SWITCH\_ON\_DISABLED\_2

```
#define SM_SWITCH_ON_DISABLED_2 0b01100000
```

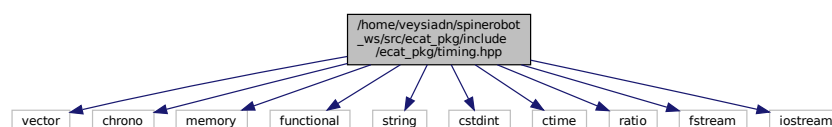
### 9.6.1.100 SM\_SWITCHED\_ON

```
#define SM_SWITCHED_ON 0b00100011
```

## 9.7 /home/veysiadm/spinrobot\_ws/src/ecat\_pkg/include/ecat\_pkg/timing.hpp File Reference

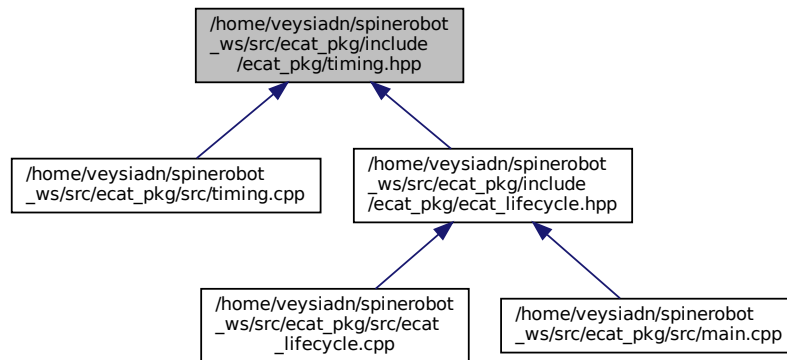
```
#include <vector>
#include <chrono>
#include <memory>
#include <functional>
#include <string>
#include <cstdint>
#include <ctime>
#include <ratio>
#include <fstream>
#include <iostream>
```

Include dependency graph for timing.hpp:





This graph shows which files directly or indirectly include this file:



## Classes

- class [Timing](#)

## Macros

- `#define` [NUMBER\\_OF\\_SAMPLES](#) 11E4

### 9.7.1 Macro Definition Documentation

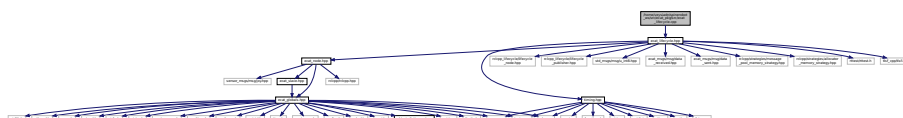
#### 9.7.1.1 NUMBER\_OF\_SAMPLES

```
#define NUMBER_OF_SAMPLES 11E4
```

## 9.8 /home/veysiadn/spineroobot\_ws/src/ecat\_pkg/src/ecat\_lifecycle.cpp File Reference

```
#include <ecat_lifecycle.hpp>
```

Include dependency graph for `ecat_lifecycle.cpp`:

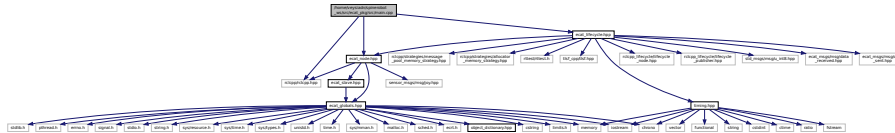






## 9.11 /home/veysiadm/spinerobot\_ws/src/ecat\_pkg/src/main.cpp File Reference

```
#include "rclcpp/rclcpp.hpp"
#include "ecat_node.hpp"
#include "ecat_lifecycle.hpp"
Include dependency graph for main.cpp:
```



### Functions

- void [signalHandler](#) (int)
- int [main](#) (int argc, char \*\*argv)

### Variables

- std::unique\_ptr< [EthercatLifeCycleNode::EthercatLifeCycle](#) > [ecat\\_lifecycle\\_node](#)

### 9.11.1 Function Documentation

#### 9.11.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

#### 9.11.1.2 signalHandler()

```
void signalHandler (
    int )
```

### 9.11.2 Variable Documentation

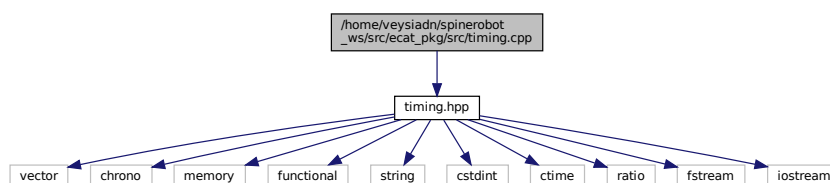
### 9.11.2.1 ecat\_lifecycle\_node

```
std::unique_ptr<EthercatLifeCycleNode::EthercatLifeCycle> ecat_lifecycle_node
```

## 9.12 /home/veysiadm/spinrobot\_ws/src/ecat\_pkg/src/timing.cpp File Reference

```
#include "timing.hpp"
```

Include dependency graph for timing.cpp:





# Index

~/home/veysiadm/spineroobot_ws/src/ecat_pkg/CMakeLists.txt	checkMasterDomainState
67	EthercatCommunication::EthercatNode, 40
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_slave.cpp	checkMasterState
68	EthercatCommunication::EthercatNode, 40
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_slave_config.cpp	checkSlaveConfigState
79	EthercatSlave, 49
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_slave_config.h	checkSlaveConfigurationState
80	EthercatCommunication::EthercatNode, 40
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/include/ecat_pkg/ecat_slave.h	check_slave.cpp
81	ecat_globals.hpp, 71
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/include/ecat_pkg/object_dictionary.hpp	check_minimum_required
83	CMakeLists.txt, 67
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/include/ecat_pkg/timing.h	CMakeLists.txt
98	cmake_minimum_required, 67
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/src/ecat_lifecycle.cpp	set, 67
99	WI, 68
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/src/ecat_node.cpp	command_
100	EthercatLifeCycleNode::EthercatLifeCycle, 34
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/src/ecat_slave.cpp	ConfigDcSync
101	EthercatCommunication::EthercatNode, 40
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/src/main.cpp	ConfigDcSyncDefault
102	EthercatCommunication::EthercatNode, 41
~/home/veysiadm/spineroobot_ws/src/ecat_pkg/src/timing.cpp	ConfigureMaster
103	EthercatCommunication::EthercatNode, 41
~EthercatLifeCycle	ConfigureSlaves
EthercatLifeCycleNode::EthercatLifeCycle, 28	EthercatCommunication::EthercatNode, 41
~EthercatNode	control_word
EthercatCommunication::EthercatNode, 39	OffsetPDO, 54
~EthercatSlave	ReceivedData, 61
EthercatSlave, 48	Controller, 15
ActivateMaster	blue_button_, 15
EthercatCommunication::EthercatNode, 39	green_button_, 16
actual_cur	left_d_button_, 16
OffsetPDO, 54	left_l_button_, 16
ReceivedData, 60	left_r_button_, 16
actual_pos	left_rb_button_, 16
OffsetPDO, 54	left_start_button_, 16
ReceivedData, 60	left_u_button_, 16
actual_tor	left_x_axis_, 16
OffsetPDO, 54	left_y_axis_, 17
ReceivedData, 60	red_button_, 17
actual_vel	right_rb_button_, 17
OffsetPDO, 54	right_start_button_, 17
ReceivedData, 60	right_x_axis_, 17
al_state_	right_y_axis_, 17
EthercatLifeCycleNode::EthercatLifeCycle, 34	xbox_button_, 17
blue_button_	yellow_button_, 17
Controller, 15	controller_
	EthercatLifeCycleNode::EthercatLifeCycle, 34

- ControlStructureBits
  - ecat\_globals.hpp, 74
- counter\_
  - Timing, 66
- CSPositionModeParam, 18
  - current\_controller\_gain, 18
  - interpolation\_time\_period, 18
  - max\_fol\_err, 19
  - max\_gear\_input\_speed, 19
  - max\_profile\_vel, 19
  - motorRatedTorque, 19
  - nominal\_current, 19
  - position\_control\_parameter\_set, 19
  - profile\_acc, 19
  - profile\_dec, 19
  - profile\_vel, 20
  - quick\_stop\_dec, 20
  - software\_position\_limit, 20
  - torque\_constant, 20
- CSTorqueModeParam, 20
  - interpolation\_time\_period, 21
  - max\_gear\_input\_speed, 21
  - max\_profile\_vel, 21
  - motorRatedTorque, 21
  - nominal\_current, 21
  - profile\_acc, 21
  - profile\_dec, 21
  - profile\_vel, 22
  - quick\_stop\_dec, 22
  - software\_position\_limit, 22
  - torque\_constant, 22
- CSVelocityModeParam, 22
  - current\_controller\_gain, 23
  - interpolation\_time\_period, 23
  - max\_fol\_err, 23
  - max\_gear\_input\_speed, 23
  - max\_profile\_vel, 23
  - motorRatedTorque, 24
  - nominal\_current, 24
  - profile\_acc, 24
  - profile\_dec, 24
  - profile\_vel, 24
  - quick\_stop\_dec, 24
  - software\_position\_limit, 24
  - torque\_constant, 24
  - velocity\_control\_parameter\_set, 25
- curr\_threshold\_homing
  - HomingParam, 51
  - SdoRequest, 63
- current\_controller\_gain
  - CSPositionModeParam, 18
  - CSVelocityModeParam, 23
- CYCLIC\_POSITION\_MODE
  - ecat\_globals.hpp, 71
- data\_
  - EthercatSlave, 49
- DeactivateCommunication
  - EthercatCommunication::EthercatNode, 41
- DefineDefaultSlaves
  - EthercatCommunication::EthercatNode, 42
- DIFF\_NS
  - ecat\_globals.hpp, 72
- ecat\_globals.hpp
  - CLOCK\_TO\_USE, 71
  - ControlStructureBits, 74
  - CYCLIC\_POSITION\_MODE, 71
  - DIFF\_NS, 72
  - ENCODER\_RESOLUTION, 72
  - ErrorRegisterBits, 75
  - FINAL\_SLAVE, 72
  - FIVE\_DEGREE\_CCW, 72
  - FREQUENCY, 72
  - g\_cycle\_time, 77
  - g\_kNsPerSec, 77
  - g\_kNumberOfServoDrivers, 77
  - g\_master, 78
  - g\_master\_domain, 78
  - g\_master\_domain\_state, 78
  - g\_master\_state, 78
  - g\_sync\_ref\_counter, 78
  - g\_sync\_timer, 78
  - GEAR\_RATIO, 72
  - INC\_PER\_ROTATION, 72
  - kAuxiliarySensor, 75
  - kCommunicationError, 75
  - kCSPosition, 76
  - kCSTorque, 76
  - kCSVelocity, 76
  - kCurrentControlStructure, 75
  - kCurrentError, 75
  - kDeviceProfileSpecificError, 75
  - kFault, 76
  - kGearLocation, 75
  - kGenericError, 75
  - kHoming, 76
  - kInternalLimitActivate, 76
  - kInterpolatedPosition, 76
  - kMainSensor, 75
  - kMotionError, 75
  - kMountingPositionSensor1, 75
  - kMountingPositionSensor2, 75
  - kMountingPositionSensor3, 75
  - kOperationEnabled, 76
  - kPositionControlStructure, 75
  - kProcessValueReference, 75
  - kProfilePosition, 76
  - kProfileTorque, 76
  - kProfileVelocity, 76
  - kQuickStop, 76
  - kReadyToSwitchOn, 75
  - kRemote, 76
  - kReserved, 75
  - kSensor1TypeDigitalIncrementalEncoder1, 76
  - kSensor1TypeNone, 76
  - kSensor2TypeAnalogIncrementalEncoderSinCos, 77



- kSensor2TypeDigitalIncrementalEncoder2, 77
- kSensor2TypeNone, 76
- kSensor2TypeSSIAbsoluteEncoder, 77
- kSensor3TypeDigitalHallSensor, 77
- kSensor3TypeNone, 77
- kSwitchedOn, 75
- kSwitchOnDisabled, 76
- kTargetReached, 76
- kTemperatureError, 75
- kVelocityControlStructure, 75
- kVoltageEnabled, 76
- kVoltageError, 75
- kWarning, 76
- MEASURE\_TIMING, 73
- MotorStates, 75
- NUM\_OF\_SLAVES, 73
- OpMode, 76
- PERIOD\_MS, 73
- PERIOD\_NS, 73
- PERIOD\_US, 73
- POSITION\_MODE, 73
- RESET\_BIT, 73
- SensorConfig, 76
- SET\_BIT, 74
- sig, 79
- TEST\_BIT, 74
- THIRTY\_DEGREE\_CCW, 74
- TIMESPEC2NS, 74
- timespec\_add, 77
- VELOCITY\_MODE, 74
- ecat\_lifecycle.hpp
  - TLSFAllocator, 80
- ecat\_lifecycle\_node
  - main.cpp, 102
- ecat\_node.cpp
  - g\_master, 100
  - g\_master\_domain, 100
  - g\_master\_domain\_state, 100
  - g\_master\_state, 101
  - g\_sync\_ref\_counter, 101
  - g\_sync\_timer, 101
- ecat\_node\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 34
- emergency\_status\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 34
- emergency\_switch
  - OffsetPDO, 54
- EnableMotors
  - EthercatLifeCycleNode::EthercatLifeCycle, 28
- ENCODER\_RESOLUTION
  - ecat\_globals.hpp, 72
- err\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 35
- error\_code
  - OffsetPDO, 54
- ErrorRegisterBits
  - ecat\_globals.hpp, 75
- ethercat\_sched\_param\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 35
- ethercat\_thread\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 35
- ethercat\_thread\_attr\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 35
- EthercatCommunication, 13
- EthercatCommunication::EthercatNode, 37
  - ~EthercatNode, 39
  - ActivateMaster, 39
  - CheckMasterDomainState, 40
  - CheckMasterState, 40
  - CheckSlaveConfigurationState, 40
  - ConfigDcSync, 40
  - ConfigDcSyncDefault, 41
  - ConfigureMaster, 41
  - ConfigureSlaves, 41
  - DeactivateCommunication, 41
  - DefineDefaultSlaves, 42
  - EthercatNode, 39
  - fd, 47
  - GetAllSlaveInformation, 42
  - GetNumberOfConnectedSlaves, 42
  - MapCustomPdcs, 42
  - MapDefaultPdcs, 43
  - OpenEthercatMaster, 43
  - RegisterDomain, 43
  - ReleaseMaster, 44
  - SetCustomSlave, 44
  - SetCyclicSyncPositionModeParameters, 44
  - SetCyclicSyncPositionModeParametersAll, 45
  - SetProfilePositionParameters, 45
  - SetProfilePositionParametersAll, 45
  - SetProfileVelocityParameters, 46
  - SetProfileVelocityParametersAll, 46
  - ShutDownEthercatMaster, 46
  - slaves\_, 47
  - WaitForOperationalMode, 47
- EthercatLifeCycle
  - EthercatLifeCycleNode::EthercatLifeCycle, 28
- EthercatLifeCycleNode, 13
- EthercatLifeCycleNode::EthercatLifeCycle, 25
  - ~EthercatLifeCycle, 28
  - al\_state\_, 34
  - command\_, 34
  - controller\_, 34
  - ecat\_node\_, 34
  - emergency\_status\_, 34
  - EnableMotors, 28
  - err\_, 35
  - ethercat\_sched\_param\_, 35
  - ethercat\_thread\_, 35
  - ethercat\_thread\_attr\_, 35
  - EthercatLifeCycle, 28
  - GetComState, 28
  - gui\_node\_data\_, 35
  - gui\_subscriber\_, 35
  - HandleControlNodeCallbacks, 28
  - HandleGuiNodeCallbacks, 29

- InitEthercatCommunication, 29
- joystick\_subscriber\_, 35
- measurement\_time, 36
- memory\_strategy, 36
- motor\_state\_, 36
- on\_activate, 29
- on\_cleanup, 29
- on\_configure, 30
- on\_deactivate, 30
- on\_error, 30
- on\_shutdown, 31
- PassCyclicExchange, 31
- PublishAllData, 31
- ReadFromSlaves, 32
- received\_data\_, 36
- received\_data\_publisher\_, 36
- sent\_data\_, 36
- sent\_data\_publisher\_, 37
- SetComThreadPriorities, 32
- StartEthercatCommunication, 32
- StartPdoExchange, 32
- timer\_, 37
- timer\_info\_, 37
- TLSFAllocator, 28
- UpdateCyclicPositionModeParameters, 33
- UpdateMotorStatePositionMode, 33
- UpdateMotorStateVelocityMode, 33
- UpdatePositionModeParameters, 33
- UpdateVelocityModeParameters, 33
- WriteToSlavesInPositionMode, 33
- WriteToSlavesVelocityMode, 34
- EthercatNode
  - EthercatCommunication::EthercatNode, 39
- EthercatSlave, 47
  - ~EthercatSlave, 48
  - CheckSlaveConfigState, 49
  - data\_, 49
  - EthercatSlave, 48
  - homing\_param\_, 49
  - kSync0\_shift\_, 49
  - motor\_state\_, 49
  - offset\_, 50
  - position\_param\_, 50
  - slave\_config\_, 50
  - slave\_config\_state\_, 50
  - slave\_info\_, 50
  - slave\_pdo\_domain\_, 50
  - velocity\_param\_, 51
- extra\_status\_reg
  - OffsetPDO, 54
- fd
  - EthercatCommunication::EthercatNode, 47
- FINAL\_SLAVE
  - ecat\_globals.hpp, 72
- FIVE\_DEGREE\_CCW
  - ecat\_globals.hpp, 72
- FREQUENCY
  - ecat\_globals.hpp, 72
- g\_cycle\_time
  - ecat\_globals.hpp, 77
- g\_kNsPerSec
  - ecat\_globals.hpp, 77
- g\_kNumberOfServoDrivers
  - ecat\_globals.hpp, 77
- g\_master
  - ecat\_globals.hpp, 78
  - ecat\_node.cpp, 100
- g\_master\_domain
  - ecat\_globals.hpp, 78
  - ecat\_node.cpp, 100
- g\_master\_domain\_state
  - ecat\_globals.hpp, 78
  - ecat\_node.cpp, 100
- g\_master\_state
  - ecat\_globals.hpp, 78
  - ecat\_node.cpp, 101
- g\_sync\_ref\_counter
  - ecat\_globals.hpp, 78
  - ecat\_node.cpp, 101
- g\_sync\_timer
  - ecat\_globals.hpp, 78
  - ecat\_node.cpp, 101
- GEAR\_RATIO
  - ecat\_globals.hpp, 72
- GetAllSlaveInformation
  - EthercatCommunication::EthercatNode, 42
- GetComState
  - EthercatLifeCycleNode::EthercatLifeCycle, 28
- GetNumberOfConnectedSlaves
  - EthercatCommunication::EthercatNode, 42
- GetTime
  - Timing, 65
- green\_button\_
  - Controller, 16
- gui\_node\_data\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 35
- gui\_subscriber\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 35
- HandleControlNodeCallbacks
  - EthercatLifeCycleNode::EthercatLifeCycle, 28
- HandleGuiNodeCallbacks
  - EthercatLifeCycleNode::EthercatLifeCycle, 29
- home\_offset
  - HomingParam, 52
  - SdoRequest, 63
- homing\_acc
  - HomingParam, 52
  - SdoRequest, 63
- homing\_method
  - HomingParam, 52
  - SdoRequest, 63
- homing\_param\_
  - EthercatSlave, 49
- HomingParam, 51
  - curr\_threshold\_homing, 51
  - home\_offset, 52

- homing\_acc, [52](#)
- homing\_method, [52](#)
- max\_fol\_err, [52](#)
- max\_profile\_vel, [52](#)
- quick\_stop\_dec, [52](#)
- speed\_for\_switch\_search, [52](#)
- speed\_for\_zero\_search, [53](#)
- INC\_PER\_ROTATION
  - ecat\_globals.hpp, [72](#)
- InitEthercatCommunication
  - EthercatLifeCycleNode::EthercatLifeCycle, [29](#)
- interpolation\_time\_period
  - CSPositionModeParam, [18](#)
  - CSTorqueModeParam, [21](#)
  - CSVelocityModeParam, [23](#)
- joystick\_subscriber\_
  - EthercatLifeCycleNode::EthercatLifeCycle, [35](#)
- kAuxiliarySensor
  - ecat\_globals.hpp, [75](#)
- kCommunicationError
  - ecat\_globals.hpp, [75](#)
- kCSPosition
  - ecat\_globals.hpp, [76](#)
- kCSTorque
  - ecat\_globals.hpp, [76](#)
- kCSVelocity
  - ecat\_globals.hpp, [76](#)
- kCurrentControlStructure
  - ecat\_globals.hpp, [75](#)
- kCurrentError
  - ecat\_globals.hpp, [75](#)
- kDeviceProfileSpecificError
  - ecat\_globals.hpp, [75](#)
- kFault
  - ecat\_globals.hpp, [76](#)
- kGearLocation
  - ecat\_globals.hpp, [75](#)
- kGenericError
  - ecat\_globals.hpp, [75](#)
- kHoming
  - ecat\_globals.hpp, [76](#)
- kInternalLimitActivate
  - ecat\_globals.hpp, [76](#)
- kInterpolatedPosition
  - ecat\_globals.hpp, [76](#)
- kMainSensor
  - ecat\_globals.hpp, [75](#)
- kMotionError
  - ecat\_globals.hpp, [75](#)
- kMountingPositionSensor1
  - ecat\_globals.hpp, [75](#)
- kMountingPositionSensor2
  - ecat\_globals.hpp, [75](#)
- kMountingPositionSensor3
  - ecat\_globals.hpp, [75](#)
- kOperationEnabled
  - ecat\_globals.hpp, [76](#)
- kPositionControlStructure
  - ecat\_globals.hpp, [75](#)
- kProcessValueReference
  - ecat\_globals.hpp, [75](#)
- kProfilePosition
  - ecat\_globals.hpp, [76](#)
- kProfileTorque
  - ecat\_globals.hpp, [76](#)
- kProfileVelocity
  - ecat\_globals.hpp, [76](#)
- kQuickStop
  - ecat\_globals.hpp, [76](#)
- kReadyToSwitchOn
  - ecat\_globals.hpp, [75](#)
- kRemote
  - ecat\_globals.hpp, [76](#)
- kReserved
  - ecat\_globals.hpp, [75](#)
- kSensor1TypeDigitalIncrementalEncoder1
  - ecat\_globals.hpp, [76](#)
- kSensor1TypeNone
  - ecat\_globals.hpp, [76](#)
- kSensor2TypeAnalogIncrementalEncoderSinCos
  - ecat\_globals.hpp, [77](#)
- kSensor2TypeDigitalIncrementalEncoder2
  - ecat\_globals.hpp, [77](#)
- kSensor2TypeNone
  - ecat\_globals.hpp, [76](#)
- kSensor2TypeSSIAbsoluteEncoder
  - ecat\_globals.hpp, [77](#)
- kSensor3TypeDigitalHallSensor
  - ecat\_globals.hpp, [77](#)
- kSensor3TypeNone
  - ecat\_globals.hpp, [77](#)
- kSwitchedOn
  - ecat\_globals.hpp, [75](#)
- kSwitchOnDisabled
  - ecat\_globals.hpp, [76](#)
- kSync0\_shift\_
  - EthercatSlave, [49](#)
- kTargetReached
  - ecat\_globals.hpp, [76](#)
- kTemperatureError
  - ecat\_globals.hpp, [75](#)
- kVelocityControlStructure
  - ecat\_globals.hpp, [75](#)
- kVoltageEnabled
  - ecat\_globals.hpp, [76](#)
- kVoltageError
  - ecat\_globals.hpp, [75](#)
- kWarning
  - ecat\_globals.hpp, [76](#)
- l\_limit\_switch
  - OffsetPDO, [55](#)
- last\_start\_time\_
  - Timing, [66](#)
- left\_d\_button\_
  -

- Controller, 16
- left\_l\_button\_
  - Controller, 16
- left\_limit\_switch\_val
  - ReceivedData, 61
- left\_r\_button\_
  - Controller, 16
- left\_rb\_button\_
  - Controller, 16
- left\_start\_button\_
  - Controller, 16
- left\_u\_button\_
  - Controller, 16
- left\_x\_axis\_
  - Controller, 16
- left\_y\_axis\_
  - Controller, 17
- main
  - main.cpp, 102
- main.cpp
  - ecat\_lifecycle\_node, 102
  - main, 102
  - signalHandler, 102
- MapCustomPdos
  - EthercatCommunication::EthercatNode, 42
- MapDefaultPdos
  - EthercatCommunication::EthercatNode, 43
- max\_fol\_err
  - CSPositionModeParam, 19
  - CSVelocityModeParam, 23
  - HomingParam, 52
  - ProfilePosParam, 57
  - SdoRequest, 63
- max\_gear\_input\_speed
  - CSPositionModeParam, 19
  - CSTorqueModeParam, 21
  - CSVelocityModeParam, 23
- max\_profile\_vel
  - CSPositionModeParam, 19
  - CSTorqueModeParam, 21
  - CSVelocityModeParam, 23
  - HomingParam, 52
  - ProfilePosParam, 57
  - ProfileVelocityParam, 59
  - SdoRequest, 64
- max\_tor
  - OffsetPDO, 55
  - ReceivedData, 61
- MEASURE\_TIMING
  - ecat\_globals.hpp, 73
- measurement\_time
  - EthercatLifeCycleNode::EthercatLifeCycle, 36
- MeasureTimeDifference
  - Timing, 65
- memory\_strategy
  - EthercatLifeCycleNode::EthercatLifeCycle, 36
- motion\_profile\_type
  - ProfilePosParam, 57
- ProfileVelocityParam, 59
- SdoRequest, 64
- motorRatedTorque
  - CSPositionModeParam, 19
  - CSTorqueModeParam, 21
  - CSVelocityModeParam, 24
- motor\_state\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 36
  - EthercatSlave, 49
- MotorStates
  - ecat\_globals.hpp, 75
- nominal\_current
  - CSPositionModeParam, 19
  - CSTorqueModeParam, 21
  - CSVelocityModeParam, 24
- NUM\_OF\_SLAVES
  - ecat\_globals.hpp, 73
- NUMBER\_OF\_SAMPLES
  - timing.hpp, 99
- object\_dictionary.hpp
  - OD\_ACTIVE\_FIELDBUS, 85
  - OD\_CHECK\_ERROR, 85
  - OD\_CONTROL\_STRUCTURE, 85
  - OD\_CONTROL\_WORD, 86
  - OD\_CURRENT\_ACTUAL\_VALUE, 86
  - OD\_CUSTOM\_PERSISTENT\_MEMORY\_1, 86
  - OD\_CUSTOM\_PERSISTENT\_MEMORY\_2, 86
  - OD\_CUSTOM\_PERSISTENT\_MEMORY\_3, 86
  - OD\_CUSTOM\_PERSISTENT\_MEMORY\_4, 86
  - OD\_DC\_CIRCUIT\_LINK\_VOLTAGE, 86
  - OD\_DIGITAL\_INCREMENTAL\_ENCODER\_1\_TYPE, 86
  - OD\_DIGITAL\_INPUTS, 87
  - OD\_DIGITAL\_OUTPUTS, 87
  - OD\_ERROR\_CODE, 87
  - OD\_ERROR\_REGISTER, 87
  - OD\_EXTRA\_STATUS\_REGISTER, 87
  - OD\_GEAR\_MAX\_INPUT\_SPEED, 87
  - OD\_GEAR\_MISC\_CONFIGURATION, 87
  - OD\_GEAR\_REDUCTION\_DENOMINATOR, 87
  - OD\_GEAR\_REDUCTION\_NUMERATOR, 88
  - OD\_INTERPOLATION\_TIME\_PERIOD, 88
  - OD\_INTERPOLATION\_TIME\_UNIT, 88
  - OD\_LINEAR\_RAMP\_TRAPEZOIDAL, 88
  - OD\_MAX\_CURRENT, 88
  - OD\_MAX\_FOLLOWING\_ERROR, 88
  - OD\_MAX\_MOTOR\_SPEED, 88
  - OD\_MAX\_PROFILE\_VELOCITY, 88
  - OD\_MOTION\_PROFILE\_TYPE, 89
  - OD\_MOTOR\_DATA\_NOMINAL\_CURRENT, 89
  - OD\_MOTOR\_DATA\_NUMBER\_OF\_POLE\_PAIRS, 89
  - OD\_MOTOR\_DATA\_OUTPUT\_CURRENT\_LIMIT, 89
  - OD\_MOTOR\_DATA\_THERMAL\_TIME\_CONSTANT\_WINDINGS, 89
  - OD\_MOTOR\_DATA\_TORQUE\_CONSTANT, 89

OD\_OPERATION\_MODE, [89](#)  
OD\_OPERATION\_MODE\_DISPLAY, [89](#)  
OD\_POSITION\_ACTUAL\_VAL, [90](#)  
OD\_POSITION\_DEMAND, [90](#)  
OD\_POSITON\_FOLLOWING\_ERROR, [90](#)  
OD\_POSITONCOUNTS, [90](#)  
OD\_POWER\_SUPPLY\_VOLTAGE, [90](#)  
OD\_PROFILE\_ACCELERATION, [90](#)  
OD\_PROFILE\_DECELERATION, [90](#)  
OD\_PROFILE\_VELOCITY, [90](#)  
OD\_QUICK\_STOP\_DECELERATION, [91](#)  
OD\_QUICK\_STOP\_MODE, [91](#)  
OD\_SENSOR\_CONFIGURATION, [91](#)  
OD\_STATUS\_WORD, [91](#)  
OD\_STOP\_OPTION\_CODE, [91](#)  
OD\_TARGET\_POSITION, [91](#)  
OD\_TARGET\_TORQUE, [91](#)  
OD\_TARGET\_VELOCITY, [92](#)  
OD\_TORQUE\_ACTUAL\_VALUE, [92](#)  
OD\_TORQUE\_MAX, [92](#)  
OD\_TORQUE\_OFFSET, [92](#)  
OD\_VELOCITY\_ACTUAL\_VALUE, [92](#)  
OD\_VELOCITY\_ADDRESS, [92](#)  
OD\_VELOCITY\_ENCODER\_RESOLUTION\_DEN, [92](#)  
OD\_VELOCITY\_ENCODER\_RESOLUTION\_NUM, [92](#)  
OD\_VELOCITY\_OFFSET, [93](#)  
SM\_COMM\_RESET, [93](#)  
SM\_CW\_ENABLE\_OPERATION, [93](#)  
SM\_CW\_ENABLE\_VOLTAGE, [93](#)  
SM\_CW\_FAULT\_RESET, [93](#)  
SM\_CW\_OD\_HALT, [93](#)  
SM\_CW\_QUICK\_STOP, [93](#)  
SM\_CW\_SWITCH\_ON, [93](#)  
SM\_EXPEDITE, [94](#)  
SM\_FAULT, [94](#)  
SM\_FAULT2, [94](#)  
SM\_FAULT\_REACTION\_ACTIVE, [94](#)  
SM\_FAULTREACTIONACTIVE2, [94](#)  
SM\_FSAFROMSTATUSWORD, [94](#)  
SM\_FULL\_RESET, [94](#)  
SM\_GO\_ENABLE, [95](#)  
SM\_GO\_READY\_TO\_SWITCH\_ON, [95](#)  
SM\_GO\_SWITCH\_ON, [95](#)  
SM\_GO\_SWITCH\_ON\_DISABLE, [95](#)  
SM\_NOT\_READY\_TO\_SWITCH\_ON, [95](#)  
SM\_NOT\_READY\_TO\_SWITCH\_ON\_2, [95](#)  
SM\_OPERATION\_ENABLED, [95](#)  
SM\_QUICK\_STOP\_ACTIVE, [95](#)  
SM\_QUICKSTOP, [96](#)  
SM\_READY\_TO\_SWITCH\_ON, [96](#)  
SM\_RELATIVE\_POS, [96](#)  
SM\_RUN, [96](#)  
SM\_START, [96](#)  
SM\_SW\_FAULT, [96](#)  
SM\_SW\_INTERNAL\_LIMIT\_ACTIVE, [96](#)  
SM\_SW\_OPERATION\_ENABLED, [96](#)  
SM\_SW\_QUICK\_STOP, [97](#)  
SM\_SW\_READY\_TO\_SWITCH\_ON, [97](#)  
SM\_SW\_REMOTE, [97](#)  
SM\_SW\_SWITCH\_ON\_DISABLED, [97](#)  
SM\_SW\_SWITCHED\_ON, [97](#)  
SM\_SW\_TARGET\_REACHED, [97](#)  
SM\_SW\_VOLTAGE\_ENABLED, [97](#)  
SM\_SW\_WARNING, [97](#)  
SM\_SWITCH\_ON\_DISABLED, [98](#)  
SM\_SWITCH\_ON\_DISABLED\_2, [98](#)  
SM\_SWITCHED\_ON, [98](#)  
OD\_ACTIVE\_FIELDBUS  
    object\_dictionary.hpp, [85](#)  
OD\_CHECK\_ERROR  
    object\_dictionary.hpp, [85](#)  
OD\_CONTROL\_STRUCTURE  
    object\_dictionary.hpp, [85](#)  
OD\_CONTROL\_WORD  
    object\_dictionary.hpp, [86](#)  
OD\_CURRENT\_ACTUAL\_VALUE  
    object\_dictionary.hpp, [86](#)  
OD\_CUSTOM\_PERSISTENT\_MEMORY\_1  
    object\_dictionary.hpp, [86](#)  
OD\_CUSTOM\_PERSISTENT\_MEMORY\_2  
    object\_dictionary.hpp, [86](#)  
OD\_CUSTOM\_PERSISTENT\_MEMORY\_3  
    object\_dictionary.hpp, [86](#)  
OD\_CUSTOM\_PERSISTENT\_MEMORY\_4  
    object\_dictionary.hpp, [86](#)  
OD\_DC\_CIRCUIT\_LINK\_VOLTAGE  
    object\_dictionary.hpp, [86](#)  
OD\_DIGITAL\_INCREMENTAL\_ENCODER\_1\_TYPE  
    object\_dictionary.hpp, [86](#)  
OD\_DIGITAL\_INPUTS  
    object\_dictionary.hpp, [87](#)  
OD\_DIGITAL\_OUTPUTS  
    object\_dictionary.hpp, [87](#)  
OD\_ERROR\_CODE  
    object\_dictionary.hpp, [87](#)  
OD\_ERROR\_REGISTER  
    object\_dictionary.hpp, [87](#)  
OD\_EXTRA\_STATUS\_REGISTER  
    object\_dictionary.hpp, [87](#)  
OD\_GEAR\_MAX\_INPUT\_SPEED  
    object\_dictionary.hpp, [87](#)  
OD\_GEAR\_MISC\_CONFIGURATION  
    object\_dictionary.hpp, [87](#)  
OD\_GEAR\_REDUCTION\_DENOMINATOR  
    object\_dictionary.hpp, [87](#)  
OD\_GEAR\_REDUCTION\_NUMERATOR  
    object\_dictionary.hpp, [88](#)  
OD\_INTERPOLATION\_TIME\_PERIOD  
    object\_dictionary.hpp, [88](#)  
OD\_INTERPOLATION\_TIME\_UNIT  
    object\_dictionary.hpp, [88](#)  
OD\_LINEAR\_RAMP\_TRAPEZOIDAL  
    object\_dictionary.hpp, [88](#)  
OD\_MAX\_CURRENT

- object\_dictionary.hpp, [88](#)
- OD\_MAX\_FOLLOWING\_ERROR
  - object\_dictionary.hpp, [88](#)
- OD\_MAX\_MOTOR\_SPEED
  - object\_dictionary.hpp, [88](#)
- OD\_MAX\_PROFILE\_VELOCITY
  - object\_dictionary.hpp, [88](#)
- OD\_MOTION\_PROFILE\_TYPE
  - object\_dictionary.hpp, [89](#)
- OD\_MOTOR\_DATA\_NOMINAL\_CURRENT
  - object\_dictionary.hpp, [89](#)
- OD\_MOTOR\_DATA\_NUMBER\_OF\_POLE\_PAIRS
  - object\_dictionary.hpp, [89](#)
- OD\_MOTOR\_DATA\_OUTPUT\_CURRENT\_LIMIT
  - object\_dictionary.hpp, [89](#)
- OD\_MOTOR\_DATA\_THERMAL\_TIME\_CONSTANT\_WINDOW
  - object\_dictionary.hpp, [89](#)
- OD\_MOTOR\_DATA\_TORQUE\_CONSTANT
  - object\_dictionary.hpp, [89](#)
- OD\_OPERATION\_MODE
  - object\_dictionary.hpp, [89](#)
- OD\_OPERATION\_MODE\_DISPLAY
  - object\_dictionary.hpp, [89](#)
- OD\_POSITION\_ACTUAL\_VAL
  - object\_dictionary.hpp, [90](#)
- OD\_POSITION\_DEMAND
  - object\_dictionary.hpp, [90](#)
- OD\_POSITON\_FOLLOWING\_ERROR
  - object\_dictionary.hpp, [90](#)
- OD\_POSITONCOUNTS
  - object\_dictionary.hpp, [90](#)
- OD\_POWER\_SUPPLY\_VOLTAGE
  - object\_dictionary.hpp, [90](#)
- OD\_PROFILE\_ACCELERATION
  - object\_dictionary.hpp, [90](#)
- OD\_PROFILE\_DECELERATION
  - object\_dictionary.hpp, [90](#)
- OD\_PROFILE\_VELOCITY
  - object\_dictionary.hpp, [90](#)
- OD\_QUICK\_STOP\_DECELERATION
  - object\_dictionary.hpp, [91](#)
- OD\_QUICK\_STOP\_MODE
  - object\_dictionary.hpp, [91](#)
- OD\_SENSOR\_CONFIGURATION
  - object\_dictionary.hpp, [91](#)
- OD\_STATUS\_WORD
  - object\_dictionary.hpp, [91](#)
- OD\_STOP\_OPTION\_CODE
  - object\_dictionary.hpp, [91](#)
- OD\_TARGET\_POSITION
  - object\_dictionary.hpp, [91](#)
- OD\_TARGET\_TORQUE
  - object\_dictionary.hpp, [91](#)
- OD\_TARGET\_VELOCITY
  - object\_dictionary.hpp, [92](#)
- OD\_TORQUE\_ACTUAL\_VALUE
  - object\_dictionary.hpp, [92](#)
- OD\_TORQUE\_MAX
  - object\_dictionary.hpp, [92](#)
- OD\_TORQUE\_OFFSET
  - object\_dictionary.hpp, [92](#)
- OD\_VELOCITY\_ACTUAL\_VALUE
  - object\_dictionary.hpp, [92](#)
- OD\_VELOCITY\_ADDRESS
  - object\_dictionary.hpp, [92](#)
- OD\_VELOCITY\_ENCODER\_RESOLUTION\_DEN
  - object\_dictionary.hpp, [92](#)
- OD\_VELOCITY\_ENCODER\_RESOLUTION\_NUM
  - object\_dictionary.hpp, [92](#)
- OD\_VELOCITY\_OFFSET
  - object\_dictionary.hpp, [93](#)
- offset\_
  - EthercatSlave, [50](#)
- OffsetPDO, [53](#)
  - actual\_cur, [54](#)
  - actual\_pos, [54](#)
  - actual\_tor, [54](#)
  - actual\_vel, [54](#)
  - control\_word, [54](#)
  - emergency\_switch, [54](#)
  - error\_code, [54](#)
  - extra\_status\_reg, [54](#)
  - l\_limit\_switch, [55](#)
  - max\_tor, [55](#)
  - op\_mode, [55](#)
  - op\_mode\_display, [55](#)
  - pos\_fol\_err, [55](#)
  - profile\_acc, [55](#)
  - profile\_dec, [55](#)
  - profile\_vel, [55](#)
  - quick\_stop\_dec, [56](#)
  - r\_limit\_switch, [56](#)
  - status\_word, [56](#)
  - target\_pos, [56](#)
  - target\_tor, [56](#)
  - target\_vel, [56](#)
- on\_activate
  - EthercatLifeCycleNode::EthercatLifeCycle, [29](#)
- on\_cleanup
  - EthercatLifeCycleNode::EthercatLifeCycle, [29](#)
- on\_configure
  - EthercatLifeCycleNode::EthercatLifeCycle, [30](#)
- on\_deactivate
  - EthercatLifeCycleNode::EthercatLifeCycle, [30](#)
- on\_error
  - EthercatLifeCycleNode::EthercatLifeCycle, [30](#)
- on\_shutdown
  - EthercatLifeCycleNode::EthercatLifeCycle, [31](#)
- op\_mode
  - OffsetPDO, [55](#)
  - ReceivedData, [61](#)
- op\_mode\_display
  - OffsetPDO, [55](#)
  - ReceivedData, [61](#)
- OpenEthercatMaster
  - EthercatCommunication::EthercatNode, [43](#)



- OpMode
  - ecat\_globals.hpp, 76
- OutInfoToFile
  - Timing, 65
- PassCyclicExchange
  - EthercatLifeCycleNode::EthercatLifeCycle, 31
- PERIOD\_MS
  - ecat\_globals.hpp, 73
- PERIOD\_NS
  - ecat\_globals.hpp, 73
- PERIOD\_US
  - ecat\_globals.hpp, 73
- pos\_fol\_err
  - OffsetPDO, 55
- position\_control\_parameter\_set
  - CSPositionModeParam, 19
- POSITION\_MODE
  - ecat\_globals.hpp, 73
- position\_param\_
  - EthercatSlave, 50
- profile\_acc
  - CSPositionModeParam, 19
  - CSTorqueModeParam, 21
  - CSVelocityModeParam, 24
  - OffsetPDO, 55
  - ProfilePosParam, 57
  - ProfileVelocityParam, 59
  - SdoRequest, 64
- profile\_dec
  - CSPositionModeParam, 19
  - CSTorqueModeParam, 21
  - CSVelocityModeParam, 24
  - OffsetPDO, 55
  - ProfilePosParam, 58
  - ProfileVelocityParam, 59
  - SdoRequest, 64
- profile\_vel
  - CSPositionModeParam, 20
  - CSTorqueModeParam, 22
  - CSVelocityModeParam, 24
  - OffsetPDO, 55
  - ProfilePosParam, 58
  - SdoRequest, 64
- ProfilePosParam, 57
  - max\_fol\_err, 57
  - max\_profile\_vel, 57
  - motion\_profile\_type, 57
  - profile\_acc, 57
  - profile\_dec, 58
  - profile\_vel, 58
  - quick\_stop\_dec, 58
- ProfileVelocityParam, 58
  - max\_profile\_vel, 59
  - motion\_profile\_type, 59
  - profile\_acc, 59
  - profile\_dec, 59
  - quick\_stop\_dec, 59
- PublishAllData
  - EthercatLifeCycleNode::EthercatLifeCycle, 31
- quick\_stop\_dec
  - CSPositionModeParam, 20
  - CSTorqueModeParam, 22
  - CSVelocityModeParam, 24
  - HomingParam, 52
  - OffsetPDO, 56
  - ProfilePosParam, 58
  - ProfileVelocityParam, 59
  - SdoRequest, 64
- r\_limit\_switch
  - OffsetPDO, 56
- ReadFromSlaves
  - EthercatLifeCycleNode::EthercatLifeCycle, 32
- received\_data\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 36
- received\_data\_publisher\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 36
- ReceivedData, 59
  - actual\_cur, 60
  - actual\_pos, 60
  - actual\_tor, 60
  - actual\_vel, 60
  - control\_word, 61
  - left\_limit\_switch\_val, 61
  - max\_tor, 61
  - op\_mode, 61
  - op\_mode\_display, 61
  - right\_limit\_switch\_val, 61
  - s\_emergency\_switch\_val, 61
  - status\_word, 61
  - target\_pos, 62
  - target\_tor, 62
  - target\_vel, 62
  - tor\_offset, 62
  - vel\_offset, 62
- red\_button\_
  - Controller, 17
- RegisterDomain
  - EthercatCommunication::EthercatNode, 43
- ReleaseMaster
  - EthercatCommunication::EthercatNode, 44
- RESET\_BIT
  - ecat\_globals.hpp, 73
- right\_limit\_switch\_val
  - ReceivedData, 61
- right\_rb\_button\_
  - Controller, 17
- right\_start\_button\_
  - Controller, 17
- right\_x\_axis\_
  - Controller, 17
- right\_y\_axis\_
  - Controller, 17
- s\_emergency\_switch\_val
  - ReceivedData, 61

- SdoRequest, 62
  - curr\_threshold\_homing, 63
  - home\_offset, 63
  - homing\_acc, 63
  - homing\_method, 63
  - max\_fol\_err, 63
  - max\_profile\_vel, 64
  - motion\_profile\_type, 64
  - profile\_acc, 64
  - profile\_dec, 64
  - profile\_vel, 64
  - quick\_stop\_dec, 64
  - speed\_for\_switch\_search, 64
  - speed\_for\_zero\_search, 64
- SensorConfig
  - ecat\_globals.hpp, 76
- sent\_data\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 36
- sent\_data\_publisher\_
  - EthercatLifeCycleNode::EthercatLifeCycle, 37
- set
  - CMakeLists.txt, 67
- SET\_BIT
  - ecat\_globals.hpp, 74
- SetComThreadPriorities
  - EthercatLifeCycleNode::EthercatLifeCycle, 32
- SetCustomSlave
  - EthercatCommunication::EthercatNode, 44
- SetCyclicSyncPositionModeParameters
  - EthercatCommunication::EthercatNode, 44
- SetCyclicSyncPositionModeParametersAll
  - EthercatCommunication::EthercatNode, 45
- SetProfilePositionParameters
  - EthercatCommunication::EthercatNode, 45
- SetProfilePositionParametersAll
  - EthercatCommunication::EthercatNode, 45
- SetProfileVelocityParameters
  - EthercatCommunication::EthercatNode, 46
- SetProfileVelocityParametersAll
  - EthercatCommunication::EthercatNode, 46
- ShutDownEthercatMaster
  - EthercatCommunication::EthercatNode, 46
- sig
  - ecat\_globals.hpp, 79
- signalHandler
  - main.cpp, 102
- slave\_config\_
  - EthercatSlave, 50
- slave\_config\_state\_
  - EthercatSlave, 50
- slave\_info\_
  - EthercatSlave, 50
- slave\_pdo\_domain\_
  - EthercatSlave, 50
- slaves\_
  - EthercatCommunication::EthercatNode, 47
- SM\_COMM\_RESET
  - object\_dictionary.hpp, 93
- SM\_CW\_ENABLE\_OPERATION
  - object\_dictionary.hpp, 93
- SM\_CW\_ENABLE\_VOLTAGE
  - object\_dictionary.hpp, 93
- SM\_CW\_FAULT\_RESET
  - object\_dictionary.hpp, 93
- SM\_CW\_OD\_HALT
  - object\_dictionary.hpp, 93
- SM\_CW\_QUICK\_STOP
  - object\_dictionary.hpp, 93
- SM\_CW\_SWITCH\_ON
  - object\_dictionary.hpp, 93
- SM\_EXPEDITE
  - object\_dictionary.hpp, 94
- SM\_FAULT
  - object\_dictionary.hpp, 94
- SM\_FAULT2
  - object\_dictionary.hpp, 94
- SM\_FAULT\_REACTION\_ACTIVE
  - object\_dictionary.hpp, 94
- SM\_FAULTREACTIONACTIVE2
  - object\_dictionary.hpp, 94
- SM\_FSAFROMSTATUSWORD
  - object\_dictionary.hpp, 94
- SM\_FULL\_RESET
  - object\_dictionary.hpp, 94
- SM\_GO\_ENABLE
  - object\_dictionary.hpp, 95
- SM\_GO\_READY\_TO\_SWITCH\_ON
  - object\_dictionary.hpp, 95
- SM\_GO\_SWITCH\_ON
  - object\_dictionary.hpp, 95
- SM\_GO\_SWITCH\_ON\_DISABLE
  - object\_dictionary.hpp, 95
- SM\_NOT\_READY\_TO\_SWITCH\_ON
  - object\_dictionary.hpp, 95
- SM\_NOT\_READY\_TO\_SWITCH\_ON\_2
  - object\_dictionary.hpp, 95
- SM\_OPERATION\_ENABLED
  - object\_dictionary.hpp, 95
- SM\_QUICK\_STOP\_ACTIVE
  - object\_dictionary.hpp, 95
- SM\_QUICKSTOP
  - object\_dictionary.hpp, 96
- SM\_READY\_TO\_SWITCH\_ON
  - object\_dictionary.hpp, 96
- SM\_RELATIVE\_POS
  - object\_dictionary.hpp, 96
- SM\_RUN
  - object\_dictionary.hpp, 96
- SM\_START
  - object\_dictionary.hpp, 96
- SM\_SW\_FAULT
  - object\_dictionary.hpp, 96
- SM\_SW\_INTERNAL\_LIMIT\_ACTIVE
  - object\_dictionary.hpp, 96
- SM\_SW\_OPERATION\_ENABLED
  - object\_dictionary.hpp, 96



- SM\_SW\_QUICK\_STOP
  - object\_dictionary.hpp, [97](#)
- SM\_SW\_READY\_TO\_SWITCH\_ON
  - object\_dictionary.hpp, [97](#)
- SM\_SW\_REMOTE
  - object\_dictionary.hpp, [97](#)
- SM\_SW\_SWITCH\_ON\_DISABLED
  - object\_dictionary.hpp, [97](#)
- SM\_SW\_SWITCHED\_ON
  - object\_dictionary.hpp, [97](#)
- SM\_SW\_TARGET\_REACHED
  - object\_dictionary.hpp, [97](#)
- SM\_SW\_VOLTAGE\_ENABLED
  - object\_dictionary.hpp, [97](#)
- SM\_SW\_WARNING
  - object\_dictionary.hpp, [97](#)
- SM\_SWITCH\_ON\_DISABLED
  - object\_dictionary.hpp, [98](#)
- SM\_SWITCH\_ON\_DISABLED\_2
  - object\_dictionary.hpp, [98](#)
- SM\_SWITCHED\_ON
  - object\_dictionary.hpp, [98](#)
- software\_position\_limit
  - CSPositionModeParam, [20](#)
  - CSTorqueModeParam, [22](#)
  - CSVelocityModeParam, [24](#)
- speed\_for\_switch\_search
  - HomingParam, [52](#)
  - SdoRequest, [64](#)
- speed\_for\_zero\_search
  - HomingParam, [53](#)
  - SdoRequest, [64](#)
- StartEthercatCommunication
  - EthercatLifeCycleNode::EthercatLifeCycle, [32](#)
- StartPdoExchange
  - EthercatLifeCycleNode::EthercatLifeCycle, [32](#)
- status\_word
  - OffsetPDO, [56](#)
  - ReceivedData, [61](#)
- target\_pos
  - OffsetPDO, [56](#)
  - ReceivedData, [62](#)
- target\_tor
  - OffsetPDO, [56](#)
  - ReceivedData, [62](#)
- target\_vel
  - OffsetPDO, [56](#)
  - ReceivedData, [62](#)
- TEST\_BIT
  - ecat\_globals.hpp, [74](#)
- THIRTY\_DEGREE\_CCW
  - ecat\_globals.hpp, [74](#)
- time\_span
  - Timing, [66](#)
- timer\_
  - EthercatLifeCycleNode::EthercatLifeCycle, [37](#)
- timer\_info\_
  - EthercatLifeCycleNode::EthercatLifeCycle, [37](#)
- timer\_start\_
  - Timing, [66](#)
- TIMESPEC2NS
  - ecat\_globals.hpp, [74](#)
- timespec\_add
  - ecat\_globals.hpp, [77](#)
- Timing, [65](#)
  - counter\_, [66](#)
  - GetTime, [65](#)
  - last\_start\_time\_, [66](#)
  - MeasureTimeDifference, [65](#)
  - OutInfoToFile, [65](#)
  - time\_span, [66](#)
  - timer\_start\_, [66](#)
  - timing\_info\_, [66](#)
- timing.hpp
  - NUMBER\_OF\_SAMPLES, [99](#)
- timing\_info\_
  - Timing, [66](#)
- TLSFAllocator
  - ecat\_lifecycle.hpp, [80](#)
  - EthercatLifeCycleNode::EthercatLifeCycle, [28](#)
- tor\_offset
  - ReceivedData, [62](#)
- torque\_constant
  - CSPositionModeParam, [20](#)
  - CSTorqueModeParam, [22](#)
  - CSVelocityModeParam, [24](#)
- UpdateCyclicPositionModeParameters
  - EthercatLifeCycleNode::EthercatLifeCycle, [33](#)
- UpdateMotorStatePositionMode
  - EthercatLifeCycleNode::EthercatLifeCycle, [33](#)
- UpdateMotorStateVelocityMode
  - EthercatLifeCycleNode::EthercatLifeCycle, [33](#)
- UpdatePositionModeParameters
  - EthercatLifeCycleNode::EthercatLifeCycle, [33](#)
- UpdateVelocityModeParameters
  - EthercatLifeCycleNode::EthercatLifeCycle, [33](#)
- vel\_offset
  - ReceivedData, [62](#)
- velocity\_control\_parameter\_set
  - CSVelocityModeParam, [25](#)
- VELOCITY\_MODE
  - ecat\_globals.hpp, [74](#)
- velocity\_param\_
  - EthercatSlave, [51](#)
- WaitForOperationalMode
  - EthercatCommunication::EthercatNode, [47](#)
- WI
  - CMakeLists.txt, [68](#)
- WriteToSlavesInPositionMode
  - EthercatLifeCycleNode::EthercatLifeCycle, [33](#)
- WriteToSlavesVelocityMode
  - EthercatLifeCycleNode::EthercatLifeCycle, [34](#)
- xbox\_button\_

Controller, [17](#)

yellow\_button\_  
Controller, [17](#)