

Behaviour Tree PiCar-V

Generated by Doxygen 1.9.5

1 Sunfounder PiCar Daemon	1
1.1 Installation:	1
1.2 View Logs	1
2 Raspberry Pi Daemon	3
2.1 Install Raspberry Pi Daemon	3
2.2 Enable / Disable Daemon to run on startup	3
2.3 Start / Stop / Restart Daemon	3
2.4 View Logs	3
2.5 Clear Past Logs	3
2.6 Reload Daemon	3
2.7 Change Config	4
2.8 Uninstall Raspberry Pi Daemon	4
2.9 Speedrun	4
3 Raspberry Pi	5
3.1 Main Technologies	5
3.2 Setting up Raspberry Pi	5
4 SETUP	7
5 Raspberry Pi	9
5.1 Installation	9
5.2 Main Technologies	9
5.3 Setting up Raspberry Pi	9
6 SETUP	11
7 Namespace Index	13
7.1 Namespace List	13
8 Hierarchical Index	15
8.1 Class Hierarchy	15
9 Class Index	17
9.1 Class List	17
10 File Index	19
10.1 File List	19
11 Namespace Documentation	21
11.1 behaviour_tree Namespace Reference	21
11.2 car Namespace Reference	21
11.3 car::configuration Namespace Reference	21
11.4 car::display Namespace Reference	21
11.5 car::display::console Namespace Reference	22

11.6 car::display::console::component Namespace Reference	22
11.7 car::display::console::component::debug Namespace Reference	22
11.8 car::display::console::component::main Namespace Reference	22
11.9 car::display::console::component::settings Namespace Reference	22
11.10 car::display::console::screen Namespace Reference	23
11.11 car::plugin Namespace Reference	23
11.12 car::system Namespace Reference	23
11.13 car::system::device Namespace Reference	23
11.14 car::system::device::lidar Namespace Reference	23
11.15 car::system::logging Namespace Reference	24
11.15.1 Typedef Documentation	24
11.15.1.1 vector_sink_mt	24
11.16 car::system::messaging Namespace Reference	24
11.17 car::system::movement Namespace Reference	24
11.18 car::system::movement::controller Namespace Reference	24
12 Class Documentation	25
12.1 car::system::movement::controller::AbstractMovementController Class Reference	25
12.1.1 Member Function Documentation	25
12.1.1.1 initialize()	26
12.1.1.2 setCameraServo1Angle()	26
12.1.1.3 setCameraServo2Angle()	26
12.1.1.4 setFrontWheelsAngle()	26
12.1.1.5 setRearLeftWheelDirectionToBackward()	26
12.1.1.6 setRearLeftWheelDirectionToForward()	27
12.1.1.7 setRearLeftWheelSpeed()	27
12.1.1.8 setRearRightWheelDirectionToBackward()	27
12.1.1.9 setRearRightWheelDirectionToForward()	27
12.1.1.10 setRearRightWheelSpeed()	27
12.1.1.11 setRearWheelsDirectionToBackward()	28
12.1.1.12 setRearWheelsDirectionToForward()	28
12.1.1.13 setRearWheelsSpeed()	28
12.1.1.14 stop()	28
12.1.1.15 terminate()	28
12.2 BackWheels Class Reference	29
12.2.1 Constructor & Destructor Documentation	29
12.2.1.1 BackWheels()	29
12.2.2 Member Function Documentation	29
12.2.2.1 backward()	29
12.2.2.2 calibration()	30
12.2.2.3 caliLeft()	30
12.2.2.4 caliOK()	30

12.2.2.5 caliRight()	30
12.2.2.6 forward()	30
12.2.2.7 getSpeed()	30
12.2.2.8 ready()	30
12.2.2.9 setSpeed()	30
12.2.2.10 stop()	31
12.2.3 Member Data Documentation	31
12.2.3.1 cali_forward_A	31
12.2.3.2 cali_forward_B	31
12.2.3.3 forward_A	31
12.2.3.4 forward_B	31
12.2.3.5 left_wheel	31
12.2.3.6 pca9685	31
12.2.3.7 right_wheel	32
12.2.3.8 speed	32
12.3 behaviour_tree::BehaviourTreeHandler Class Reference	32
12.3.1 Member Function Documentation	33
12.3.1.1 _setBehaviourTree()	33
12.3.1.2 getName()	33
12.3.1.3 handleCommand()	33
12.3.1.4 initialize()	33
12.3.1.5 setBehaviourTree()	33
12.3.1.6 startBehaviourTree()	34
12.3.1.7 stop()	34
12.3.1.8 stopBehaviourTree()	34
12.3.1.9 update()	34
12.3.2 Member Data Documentation	34
12.3.2.1 behaviour_tree	34
12.3.2.2 car_system	34
12.3.2.3 context	35
12.3.2.4 last_connected	35
12.3.2.5 tick_count	35
12.4 car::system::device::CameraDevice Class Reference	35
12.4.1 Constructor & Destructor Documentation	36
12.4.1.1 CameraDevice() [1/3]	36
12.4.1.2 CameraDevice() [2/3]	36
12.4.1.3 CameraDevice() [3/3]	36
12.4.1.4 ~CameraDevice()	36
12.4.2 Member Function Documentation	36
12.4.2.1 create()	37
12.4.2.2 disconnect()	37
12.4.2.3 getFrameBuffer()	37

12.4.2.4 operator=() [1/2]	37
12.4.2.5 operator=() [2/2]	37
12.4.2.6 start()	37
12.4.2.7 stop()	37
12.4.2.8 terminate()	38
12.4.2.9 update()	38
12.4.3 Friends And Related Function Documentation	38
12.4.3.1 DeviceManager	38
12.4.4 Member Data Documentation	38
12.4.4.1 camera_	38
12.4.4.2 camera_mutex_	38
12.4.4.3 configuration	38
12.4.4.4 connected_	39
12.4.4.5 frame_buffer_	39
12.4.4.6 last	39
12.5 car::display::console::CarConsole Class Reference	39
12.5.1 Constructor & Destructor Documentation	39
12.5.1.1 CarConsole()	40
12.5.2 Member Function Documentation	40
12.5.2.1 initialize()	40
12.5.2.2 run()	40
12.5.2.3 terminate()	40
12.5.3 Member Data Documentation	40
12.5.3.1 car_system	40
12.5.3.2 json_configuration	40
12.5.3.3 vector_sink	41
12.6 behaviour_tree::CarContext Class Reference	41
12.6.1 Constructor & Destructor Documentation	41
12.6.1.1 CarContext()	41
12.6.2 Member Function Documentation	41
12.6.2.1 _()	42
12.6.2.2 getCarSystem()	42
12.6.3 Member Data Documentation	42
12.6.3.1 car_system	42
12.7 car::system::CarSystem Class Reference	42
12.7.1 Constructor & Destructor Documentation	43
12.7.1.1 CarSystem()	43
12.7.2 Member Function Documentation	43
12.7.2.1 disconnect()	43
12.7.2.2 getConfiguration()	43
12.7.2.3 getDeviceManager()	44
12.7.2.4 getMessagingSystem()	44

12.7.2.5 getMovementSystem()	44
12.7.2.6 getPlugin()	44
12.7.2.7 initialize()	44
12.7.2.8 reload()	44
12.7.2.9 sendData()	44
12.7.2.10 setConfiguration()	45
12.7.2.11 start()	45
12.7.2.12 stop()	45
12.7.2.13 terminate()	45
12.7.2.14 tryConnect()	45
12.7.2.15 update()	45
12.7.3 Member Data Documentation	45
12.7.3.1 configuration_	45
12.7.3.2 device_manager_	46
12.7.3.3 initialized	46
12.7.3.4 messaging_system_	46
12.7.3.5 movement_system_	46
12.7.3.6 plugin_manager_	46
12.7.3.7 started	46
12.8 car::configuration::Configuration Struct Reference	46
12.8.1 Member Function Documentation	47
12.8.1.1 getCameraFpsInterval()	47
12.8.1.2 setCameraFps()	47
12.8.2 Member Data Documentation	47
12.8.2.1 behaviour_tree_update_ms_interval	47
12.8.2.2 camera_fps	48
12.8.2.3 camera_fps_interval	48
12.8.2.4 camera_index	48
12.8.2.5 host	48
12.8.2.6 lidar_port	48
12.8.2.7 use_camera	48
12.8.2.8 use_lidar	48
12.9 car::display::console::component::main::ConnectButton Class Reference	49
12.9.1 Constructor & Destructor Documentation	49
12.9.1.1 ConnectButton()	49
12.9.2 Member Function Documentation	49
12.9.2.1 element()	49
12.9.3 Member Data Documentation	49
12.9.3.1 button_pressed	50
12.9.3.2 car_system	50
12.9.3.3 main_button	50
12.9.3.4 main_button_text	50

12.9.3.5 main_debounce	50
12.9.3.6 on_connect_failure	50
12.10 car::display::console::component::debug::DebugEnabler Class Reference	50
12.10.1 Member Function Documentation	51
12.10.1.1 getCheckbox()	51
12.10.1.2 getWarningModal()	51
12.10.1.3 isEnabled()	51
12.10.2 Member Data Documentation	51
12.10.2.1 checkbox_value	51
12.10.2.2 component	52
12.10.2.3 debounce	52
12.10.2.4 DEBUG_ENABLE_WARNING_MESSAGE	52
12.10.2.5 DEBUG_MODE_DISABLED_MESSAGE	52
12.10.2.6 DEBUG_MODE_ENABLED_MESSAGE	52
12.10.2.7 DEBUG_MODE_WAIT_MESSAGE	52
12.10.2.8 display_warn_debug_modal	52
12.10.2.9 enabled	53
12.10.2.10 status	53
12.11 car::display::console::component::debug::DebugLidarCheckbox Class Reference	53
12.11.1 Constructor & Destructor Documentation	53
12.11.1.1 DebugLidarCheckbox()	53
12.11.2 Member Function Documentation	54
12.11.2.1 element()	54
12.11.2.2 getLidarMotorSignal()	54
12.11.3 Member Data Documentation	54
12.11.3.1 lidar_motor_checkbox_component	54
12.11.3.2 LIDAR_MOTOR_DISABLED_MESSAGE	54
12.11.3.3 lidar_motor_enabled	54
12.11.3.4 LIDAR_MOTOR_ENABLED_MESSAGE	54
12.11.3.5 lidar_motor_loading_debounce	55
12.11.3.6 lidar_motor_signal	55
12.11.3.7 lidar_motor_status	55
12.12 car::display::console::component::debug::DebugMessagingTextbox Class Reference	55
12.12.1 Constructor & Destructor Documentation	55
12.12.1.1 DebugMessagingTextbox()	56
12.12.2 Member Function Documentation	56
12.12.2.1 element()	56
12.12.3 Member Data Documentation	56
12.12.3.1 message	56
12.12.3.2 message_signal	56
12.12.3.3 messaging_container	56
12.12.3.4 messaging_textbox	56

12.12.3.5 messaging_title	57
12.13 car::display::console::component::debug::DebugMovementRenderer Class Reference	57
12.13.1 Constructor & Destructor Documentation	58
12.13.1.1 DebugMovementRenderer()	58
12.13.2 Member Function Documentation	58
12.13.2.1 element()	58
12.13.2.2 getCameraServo1AngleSliderValue()	58
12.13.2.3 getCameraServo2AngleSliderValue()	58
12.13.2.4 getFrontWheelsAngleSliderValue()	59
12.13.2.5 getRearLeftWheelSpeedSliderValue()	59
12.13.2.6 getRearRightWheelSpeedSliderValue()	59
12.13.2.7 getRearWheelDirectionSignal()	59
12.13.2.8 updateCameraServo1()	59
12.13.2.9 updateCameraServo2()	59
12.13.2.10 updateFrontWheels()	59
12.13.2.11 updateRearWheels()	60
12.13.3 Member Data Documentation	60
12.13.3.1 camera_servo_1_angle_slider	60
12.13.3.2 camera_servo_1_angle_slider_angle	60
12.13.3.3 camera_servo_2_angle_slider	60
12.13.3.4 camera_servo_2_angle_slider_angle	60
12.13.3.5 DEFAULT_FRONT_WHEEL_ANGLE	60
12.13.3.6 DEFAULT_REAR_WHEEL_SPEED	61
12.13.3.7 front_wheels_angle_slider	61
12.13.3.8 front_wheels_angle_slider_value	61
12.13.3.9 previous_camera_servo_1_angle_slider_angle	61
12.13.3.10 previous_camera_servo_2_angle_slider_angle	61
12.13.3.11 previous_front_wheels_angle_slider_value	61
12.13.3.12 previous_rear_left_wheel_speed_slider_value	61
12.13.3.13 previous_rear_right_wheel_speed_slider_value	62
12.13.3.14 previous_rear_wheels_speed_slider_value	62
12.13.3.15 rear_left_wheel_speed_slider	62
12.13.3.16 rear_left_wheel_speed_slider_value	62
12.13.3.17 rear_right_wheel_speed_slider	62
12.13.3.18 rear_right_wheel_speed_slider_value	62
12.13.3.19 rear_wheel_direction	62
12.13.3.20 REAR_WHEEL_DIRECTION_BACKWARD_MESSAGE	63
12.13.3.21 rear_wheel_direction_checkbox_component	63
12.13.3.22 rear_wheel_direction_debounce	63
12.13.3.23 REAR_WHEEL_DIRECTION_FORWARD_MESSAGE	63
12.13.3.24 rear_wheel_direction_signal	63
12.13.3.25 rear_wheel_direction_status	63

12.13.3.26 rear_wheel_menu_entry	63
12.13.3.27 rear_wheel_speed_slider	64
12.13.3.28 rear_wheels_speed_slider_value	64
12.13.3.29 servo_menu_entry	64
12.13.3.30 slider_container	64
12.14 car::system::device::DeviceManager Class Reference	64
12.14.1 Constructor & Destructor Documentation	65
12.14.1.1 DeviceManager()	65
12.14.2 Member Function Documentation	65
12.14.2.1 create()	65
12.14.2.2 getCameraDevice()	65
12.14.2.3 getLidarDevice()	65
12.14.2.4 initialize()	66
12.14.2.5 isRunning()	66
12.14.2.6 start()	66
12.14.2.7 stop()	66
12.14.2.8 terminate()	66
12.14.2.9 update()	66
12.14.3 Member Data Documentation	66
12.14.3.1 camera_device_	66
12.14.3.2 car_system	67
12.14.3.3 is_initialized_	67
12.14.3.4 is_running_	67
12.14.3.5 lidar_device_	67
12.15 car::system::movement::controller::DummyMovementController Class Reference	67
12.15.1 Member Function Documentation	68
12.15.1.1 initialize()	68
12.15.1.2 setCameraServo1Angle()	68
12.15.1.3 setCameraServo2Angle()	68
12.15.1.4 setFrontWheelsAngle()	69
12.15.1.5 setRearLeftWheelDirectionToBackward()	69
12.15.1.6 setRearLeftWheelDirectionToForward()	69
12.15.1.7 setRearLeftWheelSpeed()	69
12.15.1.8 setRearRightWheelDirectionToBackward()	69
12.15.1.9 setRearRightWheelDirectionToForward()	69
12.15.1.10 setRearRightWheelSpeed()	70
12.15.1.11 setRearWheelsDirectionToBackward()	70
12.15.1.12 setRearWheelsDirectionToForward()	70
12.15.1.13 setRearWheelsSpeed()	70
12.15.1.14 stop()	70
12.15.1.15 terminate()	71
12.16 car::system::messaging::MessagingSystem::FirstMessageStruct Struct Reference	71

12.16.1 Member Data Documentation	71
12.16.1.1 condition	71
12.16.1.2 error_message	71
12.16.1.3 uuid	71
12.17 car::configuration::JsonConfiguration Class Reference	72
12.17.1 Constructor & Destructor Documentation	72
12.17.1.1 JsonConfiguration()	72
12.17.2 Member Function Documentation	72
12.17.2.1 getConfigFilePath()	72
12.17.2.2 loadConfiguration()	72
12.17.2.3 setConfigFilePath()	72
12.17.3 Member Data Documentation	73
12.17.3.1 config_file_path	73
12.17.3.2 exe_dir	73
12.18 car::system::device::lidar::LidarDevice Class Reference	73
12.18.1 Member Function Documentation	74
12.18.1.1 disconnect()	74
12.18.1.2 getScanData()	74
12.18.1.3 initialize()	74
12.18.1.4 setScanData()	74
12.18.1.5 start()	74
12.18.1.6 stop()	75
12.18.1.7 terminate()	75
12.18.1.8 update()	75
12.18.2 Friends And Related Function Documentation	75
12.18.2.1 DeviceManager	75
12.18.3 Member Data Documentation	75
12.18.3.1 scan_data_	75
12.19 car::system::device::lidar::LidarDummy Class Reference	76
12.19.1 Constructor & Destructor Documentation	76
12.19.1.1 LidarDummy()	76
12.19.2 Member Function Documentation	76
12.19.2.1 disconnect()	76
12.19.2.2 initialize()	77
12.19.2.3 start()	77
12.19.2.4 stop()	77
12.19.2.5 terminate()	77
12.19.2.6 update()	77
12.20 car::system::device::lidar::LidarScanner Class Reference	78
12.20.1 Constructor & Destructor Documentation	78
12.20.1.1 LidarScanner()	78
12.20.2 Member Function Documentation	79

12.20.2.1 create()	79
12.20.2.2 disconnect()	79
12.20.2.3 initialize()	79
12.20.2.4 start()	79
12.20.2.5 stop()	79
12.20.2.6 terminate()	80
12.20.2.7 update()	80
12.20.3 Member Data Documentation	80
12.20.3.1 configuration_	80
12.20.3.2 lidar_	80
12.20.3.3 running	80
12.20.3.4 scan_data_	80
12.20.3.5 scan_data_mutex_	81
12.20.3.6 scan_generator_	81
12.21 car::display::console::screen::LoggingScreen Class Reference	81
12.21.1 Constructor & Destructor Documentation	81
12.21.1.1 LoggingScreen()	81
12.21.2 Member Function Documentation	81
12.21.2.1 element()	82
12.21.3 Member Data Documentation	82
12.21.3.1 line_elements	82
12.21.3.2 menu	82
12.21.3.3 my_custom_menu	82
12.21.3.4 selected_line	82
12.21.3.5 vector_sink	82
12.22 car::display::console::component::main::MainErrorModal Class Reference	83
12.22.1 Constructor & Destructor Documentation	83
12.22.1.1 MainErrorModal()	83
12.22.2 Member Function Documentation	83
12.22.2.1 element()	83
12.22.2.2 setErrorMessage()	83
12.22.3 Member Data Documentation	83
12.22.3.1 error_element	84
12.22.3.2 error_modal_shown	84
12.22.3.3 main_error_modal	84
12.23 car::display::console::component::main::MainExitModal Class Reference	84
12.23.1 Constructor & Destructor Documentation	84
12.23.1.1 MainExitModal()	84
12.23.2 Member Function Documentation	85
12.23.2.1 element()	85
12.23.3 Member Data Documentation	85
12.23.3.1 exit	85

12.23.3.2	exit_modal_shown	85
12.23.3.3	main_exit_modal	85
12.24	car::display::console::screen::MainScreen Class Reference	85
12.24.1	Constructor & Destructor Documentation	86
12.24.1.1	MainScreen()	86
12.24.2	Member Function Documentation	86
12.24.2.1	element()	86
12.24.3	Member Data Documentation	86
12.24.3.1	box	86
12.24.3.2	car_system	86
12.24.3.3	connect_button	87
12.24.3.4	info	87
12.24.3.5	main_component	87
12.24.3.6	main_error_modal	87
12.24.3.7	main_exit_modal	87
12.24.3.8	main_screen	87
12.25	car::system::messaging::MessagingSystem Class Reference	87
12.25.1	Constructor & Destructor Documentation	88
12.25.1.1	MessagingSystem()	89
12.25.2	Member Function Documentation	89
12.25.2.1	getCommandSignal()	89
12.25.2.2	getDisconnectSignal()	89
12.25.2.3	getFirstMessage()	89
12.25.2.4	getMessageSignal()	89
12.25.2.5	getSelectionSignal()	89
12.25.2.6	getUUID()	90
12.25.2.7	handleMessage()	90
12.25.2.8	initialize()	90
12.25.2.9	initializeWebSocket()	90
12.25.2.10	isConnected()	90
12.25.2.11	onDisconnect()	91
12.25.2.12	onFirstMessage()	91
12.25.2.13	onMessageCallback()	91
12.25.2.14	sendMessage()	91
12.25.2.15	setConfiguration()	91
12.25.2.16	stop()	91
12.25.2.17	terminate()	92
12.25.2.18	tryConnect()	92
12.25.3	Member Data Documentation	92
12.25.3.1	command_signal_	92
12.25.3.2	configuration_	92
12.25.3.3	connected_	92

12.25.3.4 message_signal_	92
12.25.3.5 on_disconnect_signal_	93
12.25.3.6 selection_signal_	93
12.25.3.7 uuid_	93
12.25.3.8 websocket_	93
12.25.3.9 websocket_url_	93
12.26 car::system::movement::MovementSystem Class Reference	93
12.26.1 Constructor & Destructor Documentation	94
12.26.1.1 MovementSystem()	94
12.26.1.2 ~MovementSystem()	94
12.26.2 Member Function Documentation	94
12.26.2.1 initialize()	94
12.26.2.2 setCameraServo1Angle()	95
12.26.2.3 setCameraServo2Angle()	95
12.26.2.4 setFrontWheelsAngle()	95
12.26.2.5 setRearLeftWheelDirectionToBackward()	95
12.26.2.6 setRearLeftWheelDirectionToForward()	95
12.26.2.7 setRearLeftWheelSpeed()	95
12.26.2.8 setRearRightWheelDirectionToBackward()	95
12.26.2.9 setRearRightWheelDirectionToForward()	96
12.26.2.10 setRearRightWheelSpeed()	96
12.26.2.11 setRearWheelsDirectionToBackward()	96
12.26.2.12 setRearWheelsDirectionToForward()	96
12.26.2.13 setRearWheelsSpeed()	96
12.26.2.14 start()	96
12.26.2.15 stop()	96
12.26.2.16 terminate()	97
12.26.3 Member Data Documentation	97
12.26.3.1 movement_controller	97
12.27 car::plugin::Plugin Class Reference	97
12.27.1 Member Function Documentation	97
12.27.1.1 getName()	97
12.27.1.2 initialize()	98
12.27.1.3 stop()	98
12.27.1.4 update()	98
12.28 car::plugin::PluginManager Class Reference	98
12.28.1 Member Function Documentation	99
12.28.1.1 addPlugin()	99
12.28.1.2 getPlugin()	99
12.28.1.3 initialize()	99
12.28.1.4 stop()	99
12.28.1.5 terminate()	99

12.28.1.6 update()	99
12.28.2 Member Data Documentation	99
12.28.2.1 plugins	100
12.29 rpi_daemon Class Reference	100
12.29.1 Member Function Documentation	100
12.29.1.1 connect()	100
12.29.1.2 on_reload()	101
12.29.1.3 on_start()	101
12.29.1.4 on_stop()	101
12.29.1.5 on_update()	101
12.29.1.6 update()	101
12.29.2 Member Data Documentation	101
12.29.2.1 any_configuration_empty	101
12.29.2.2 attempted_to_reconnect	101
12.29.2.3 car_system	102
12.29.2.4 connection_ms_interval	102
12.29.2.5 last_connected	102
12.30 car::display::console::component::settings::SettingsEditConfig Class Reference	102
12.30.1 Constructor & Destructor Documentation	102
12.30.1.1 SettingsEditConfig()	103
12.30.2 Member Function Documentation	103
12.30.2.1 element()	103
12.30.3 Member Data Documentation	103
12.30.3.1 car_system	103
12.30.3.2 input_settings_file_path	103
12.30.3.3 json_configuration	103
12.30.3.4 load_button	103
12.30.3.5 placeholder	104
12.30.3.6 settings_file_path	104
12.31 car::display::console::screen::SettingsScreen Class Reference	104
12.31.1 Constructor & Destructor Documentation	104
12.31.1.1 SettingsScreen()	104
12.31.2 Member Function Documentation	105
12.31.2.1 element()	105
12.31.2.2 update()	105
12.31.3 Member Data Documentation	105
12.31.3.1 car_system	105
12.31.3.2 debug_enabler	105
12.31.3.3 debug_lidar_checkbox	105
12.31.3.4 debug_messaging_text_box	105
12.31.3.5 debug_movement_renderer	106
12.31.3.6 settings_edit_config	106

12.32 TB6612 Class Reference	106
12.32.1 Constructor & Destructor Documentation	106
12.32.1.1 TB6612()	106
12.32.2 Member Function Documentation	107
12.32.2.1 backward()	107
12.32.2.2 forward()	107
12.32.2.3 getMotorPin()	107
12.32.2.4 getPWMPin()	107
12.32.2.5 setOffset()	107
12.32.2.6 setPWM()	107
12.32.2.7 stop()	107
12.32.3 Member Data Documentation	108
12.32.3.1 motor_pin	108
12.32.3.2 offset	108
12.32.3.3 pwm_pin	108
12.33 car::system::logging::VectorSink< Mutex > Class Template Reference	108
12.33.1 Constructor & Destructor Documentation	109
12.33.1.1 VectorSink()	109
12.33.2 Member Function Documentation	109
12.33.2.1 flush_()	109
12.33.2.2 get_log_messages()	109
12.33.2.3 sink_it_()	109
12.33.3 Member Data Documentation	109
12.33.3.1 log_messages	110
12.33.3.2 max_lines	110
13 File Documentation	111
13.1 behaviour_tree/src/main.cpp File Reference	111
13.1.1 Function Documentation	111
13.1.1.1 kbhit()	111
13.1.1.2 main()	112
13.2 daemon/src/main.cpp File Reference	112
13.2.1 Function Documentation	112
13.2.1.1 getLidarDevice()	112
13.2.1.2 getMovementController()	113
13.2.1.3 main()	113
13.2.1.4 terminate_handler()	113
13.3 tui/src/main.cpp File Reference	113
13.3.1 Function Documentation	113
13.3.1.1 getLidarDevice()	114
13.3.1.2 getMovementController()	114
13.3.1.3 main()	114

13.4 common/include/behaviour_tree/BehaviourTreeHandler.hpp File Reference	114
13.5 BehaviourTreeHandler.hpp	115
13.6 common/include/behaviour_tree/CarContext.hpp File Reference	117
13.7 CarContext.hpp	117
13.8 common/include/car/configuration/Configuration.h File Reference	118
13.9 Configuration.h	118
13.10 common/include/car/plugin/Plugin.h File Reference	118
13.11 Plugin.h	119
13.12 common/include/car/plugin/PluginManager.h File Reference	119
13.13 PluginManager.h	120
13.14 common/include/car/system/CarSystem.h File Reference	121
13.15 CarSystem.h	121
13.16 common/include/car/system/device/CameraDevice.h File Reference	122
13.17 CameraDevice.h	123
13.18 common/include/car/system/device/DeviceManager.h File Reference	123
13.19 DeviceManager.h	124
13.20 common/include/car/system/device/lidar/LidarDevice.h File Reference	125
13.21 LidarDevice.h	125
13.22 common/include/car/system/device/lidar/LidarDummy.h File Reference	126
13.23 LidarDummy.h	126
13.24 common/include/car/system/device/lidar/LidarScanner.h File Reference	126
13.25 LidarScanner.h	127
13.26 common/include/car/system/logging/VectorSink.h File Reference	128
13.27 VectorSink.h	129
13.28 common/include/car/system/messaging/MessagingSystem.h File Reference	129
13.29 MessagingSystem.h	130
13.30 common/include/car/system/messaging/StreamType.h File Reference	131
13.30.1 Enumeration Type Documentation	131
13.30.1.1 StreamType	131
13.31 StreamType.h	132
13.32 common/include/car/system/movement/controller/AbstractMovementController.h File Reference	132
13.33 AbstractMovementController.h	132
13.34 common/include/car/system/movement/controller/DeviceMovementController.h File Reference	133
13.35 DeviceMovementController.h	133
13.36 common/include/car/system/movement/controller/DummyMovementController.h File Reference	134
13.37 DummyMovementController.h	134
13.38 common/include/car/system/movement/devices/RearWheel.h File Reference	135
13.39 RearWheel.h	135
13.40 common/include/car/system/movement/devices/Servo.h File Reference	135
13.41 Servo.h	135
13.42 common/include/car/system/movement/MovementSystem.h File Reference	136
13.43 MovementSystem.h	136

13.44 common/src/car/system/CarSystem.cpp File Reference	138
13.45 common/src/car/system/device/CameraDevice.cpp File Reference	138
13.46 common/src/car/system/device/DeviceManager.cpp File Reference	138
13.47 common/src/car/system/messaging/MessagingSystem.cpp File Reference	139
13.48 common/src/car/system/movement/controller/DeviceMovementController.cpp File Reference	139
13.49 common/src/car/system/movement/controller/DummyMovementController.cpp File Reference	139
13.50 common/src/car/system/movement/devices/RearWheel.cpp File Reference	140
13.51 common/src/car/system/movement/devices/Servo.cpp File Reference	140
13.52 common/tests/pca9685/test_front_wheels.cpp File Reference	140
13.52.1 Function Documentation	140
13.52.1.1 main()	140
13.52.1.2 map()	140
13.52.1.3 setAngle()	141
13.52.1.4 setAngleToAnalog()	141
13.52.2 Variable Documentation	141
13.52.2.1 offset	141
13.53 common/tests/tb6612/test_rear_wheels.cpp File Reference	141
13.53.1 Function Documentation	141
13.53.1.1 main()	142
13.53.1.2 test()	142
13.54 daemon/install/README.md File Reference	142
13.55 daemon/README.md File Reference	142
13.56 README.md File Reference	142
13.57 tui/README.md File Reference	142
13.58 repository/packages/t/tb6612/tb6612/include/TB6612.h File Reference	142
13.59 TB6612.h	142
13.60 repository/packages/t/tb6612/tb6612/src/TB6612.cpp File Reference	143
13.61 SETUP.md File Reference	143
13.62 tui/SETUP.md File Reference	143
13.63 tui/src/car/configuration/JsonConfiguration.cxx File Reference	143
13.63.1 Macro Definition Documentation	144
13.63.1.1 JSONCONFIGURATION_CXX	144
13.64 tui/src/car/display/console/CarConsole.cpp File Reference	144
13.65 tui/src/car/display/console/CarConsole.h File Reference	144
13.66 CarConsole.h	145
13.67 tui/src/car/display/console/component/debug/DebugEnabler.cxx File Reference	145
13.67.1 Macro Definition Documentation	146
13.67.1.1 DEBUGENABLER_CXX	146
13.68 tui/src/car/display/console/component/debug/DebugLidarCheckbox.cxx File Reference	146
13.68.1 Macro Definition Documentation	146
13.68.1.1 DEBUGLIDARCHECKBOX_CXX	146
13.69 tui/src/car/display/console/component/debug/DebugMessagingTextbox.cxx File Reference	147

13.69.1 Macro Definition Documentation	147
13.69.1.1 DEBUGMESSAGINGTEXTBOX_CXX	147
13.70 tui/src/car/display/console/component/debug/DebugMovementRenderer.cxx File Reference	147
13.70.1 Macro Definition Documentation	148
13.70.1.1 DEBUGMOVEMENTRENDERER_CXX	148
13.71 tui/src/car/display/console/component/main/ConnectButton.cxx File Reference	148
13.71.1 Macro Definition Documentation	149
13.71.1.1 CONNECTBUTTON_CXX	149
13.72 tui/src/car/display/console/component/main/MainErrorModal.cxx File Reference	149
13.72.1 Macro Definition Documentation	149
13.72.1.1 MAINERRORMODAL_CXX	149
13.73 tui/src/car/display/console/component/main/MainExitModal.cxx File Reference	150
13.73.1 Macro Definition Documentation	150
13.73.1.1 MAINEXITMODAL_CXX	150
13.74 tui/src/car/display/console/component/settings/SettingsEditConfig.cxx File Reference	150
13.74.1 Macro Definition Documentation	151
13.74.1.1 SETTINGSEEDITCONFIG_CXX	151
13.75 tui/src/car/display/console/screen/LoggingScreen.cxx File Reference	151
13.75.1 Macro Definition Documentation	152
13.75.1.1 LOGGINGSCREEN_CXX	152
13.76 tui/src/car/display/console/screen/MainScreen.cxx File Reference	152
13.76.1 Macro Definition Documentation	152
13.76.1.1 MAINSCREEN_CXX	152
13.77 tui/src/car/display/console/screen/SettingsScreen.cxx File Reference	153
13.77.1 Macro Definition Documentation	153
13.77.1.1 SETTINGSSCREEN_CXX	153
Index	155

Chapter 1

Sunfounder PiCar Daemon

1.1 Installation:

Run the follow commands to install the Daemon:

```
sudo chmod +x ./install_script.sh # Grant permission to install_script to run as exe
systemctl stop rpi_daemon # Stop any previous rpi_daemon
sudo ./install_script.sh install # Install the Daemon
sudo systemctl daemon-reload # Reload the Daemon if there was any previous
sudo systemctl start rpi_daemon # Start the new Daemon
```

Or use the single command:

```
sudo chmod +x ./install_script.sh && systemctl stop rpi_daemon && sudo ./install_script.sh install && sudo
systemctl daemon-reload && sudo systemctl start rpi_daemon
```

1.2 View Logs

```
systemctl status rpi_daemon.service journalctl -u rpi\_daemon
```


Chapter 2

Raspberry Pi Daemon

This will only work for Linux.

2.1 Install Raspberry Pi Daemon

```
xmake install --admin rpi_daemon
```

2.2 Enable / Disable Daemon to run on startup

```
sudo systemctl [enable|disable] rpi_daemon
```

2.3 Start / Stop / Restart Daemon

```
sudo systemctl [start/stop/restart] rpi_daemon
```

2.4 View Logs

```
systemctl status rpi_daemon.service journalctl -u rpi_daemon
```

2.5 Clear Past Logs

```
sudo journalctl -m --vacuum-time=1s
```

2.6 Reload Daemon

```
sudo systemctl reload rpi_daemon
```

2.7 Change Config

```
sudo nano /etc/rpi_daemon/rpi_daemon.conf
```

2.8 Uninstall Raspberry Pi Daemon

```
xmake uninstall --admin rpi_daemon
```

2.9 Speedrun

```
sudo systemctl stop rpi_daemon && xmake && xmake install --admin rpi_daemon  
&& sudo systemctl daemon-reload && sudo systemctl start rpi_daemon
```


Chapter 3

Raspberry Pi

This sub-project contains three other sub-projects:

- common - Main functionality is stored here
- daemon - Daemon Application for seamless startup and termination
- tui - Terminal User Interface

3.1 Main Technologies

- C++17
- `ixwebsocket`
- `ftxui`
- `rapidjson`

3.2 Setting up Raspberry Pi

If you want to setup the raspberry pi, you can go to `SETUP.md`

Chapter 4

SETUP

Set up the following first:

- Wifi - Use the Mobile Hotspot feature of your device
- SSH - Enable
- I2C - Enable

See if Wifi does not connect: [Raspberrypi Stackexchange](#)

Now connect to the Raspberry Pi with SSH

Run the following commands to get the raspberry pi running:

```
sudo apt-get install build-essential
curl -fsSL https://xmake.io/shget.text | bash
source ~/.xmake/profile
```

Run the following commands before cross compilation:

```
sudo apt update
sudo apt upgrade
sudo apt dist-upgrade
```

or `sudo apt update && sudo apt upgrade && sudo apt dist-upgrade`

Afterwards run `chmod +x ./rpi_tui` on the new executable

Use [CyberDuck](#) to copy the raspberry_pi folder.

Chapter 5

Raspberry Pi

This sub-project handles all the logic of the Sunfounder Raspberry Pi.

The system is broken down to 3 sub systems:

- lidar - Handles the lidar scanner of the Raspberry Pi
- messaging - Handles the websocket of the Raspberry Pi
- movement - Handles the movement of the Sunfounder Raspberry Pi

The system is then displayed by the `CarConsole`.

5.1 Installation

Run the following command to build the sub-project:

```
xmake build raspberry_pi
```

To connect the Raspberry Pi to the backend, you would need the `IPv4 Address` of your Computer/Server and change the host to the value in `IPv4 Address`.

If this doesn't work, you may need to restart your Computer/Server.

5.2 Main Technologies

- C++17
- `ixwebsocket`
- `ftxui`
- `nlohmann_json`

5.3 Setting up Raspberry Pi

If you want to setup the raspberry pi, you can go to `SETUP.md`

Chapter 6

SETUP

Set up the following first:

- Wifi - Use the Mobile Hotspot feature of your device
- SSH - Enable
- I2C - Enable

See if Wifi does not connect: [Raspberrypi Stackexchange](#)

Now connect to the Raspberry Pi with SSH

Run the following commands to get the raspberry pi running:

```
sudo apt-get install build-essential
curl -fsSL https://xmake.io/shget.text | bash
source ~/.xmake/profile
```

Run the following commands before cross compilation:

```
sudo apt update
sudo apt upgrade
sudo apt dist-upgrade
```

or `sudo apt update && sudo apt upgrade && sudo apt dist-upgrade`

Afterwards run `chmod +x ./raspberry_pi` on the new executable

Use [CyberDuck](#) to copy the raspberrypi folder.

Chapter 7

Namespace Index

7.1 Namespace List

Here is a list of all namespaces with brief descriptions:

behaviour_tree	21
car	21
car::configuration	21
car::display	21
car::display::console	22
car::display::console::component	22
car::display::console::component::debug	22
car::display::console::component::main	22
car::display::console::component::settings	22
car::display::console::screen	23
car::plugin	23
car::system	23
car::system::device	23
car::system::device::lidar	23
car::system::logging	24
car::system::messaging	24
car::system::movement	24
car::system::movement::controller	24

Chapter 8

Hierarchical Index

8.1 Class Hierarchy

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

car::system::movement::controller::AbstractMovementController	25
car::system::movement::controller::DummyMovementController	67
BackWheels	29
spdlog::sinks::base_sink	
car::system::logging::VectorSink< Mutex >	108
car::system::device::CameraDevice	35
car::display::console::CarConsole	39
car::configuration::Configuration	46
car::display::console::component::main::ConnectButton	49
Context	
behaviour_tree::CarContext	41
daemon	
rpi_daemon	100
car::display::console::component::debug::DebugEnabler	50
car::display::console::component::debug::DebugLidarCheckbox	53
car::display::console::component::debug::DebugMessagingTextbox	55
car::display::console::component::debug::DebugMovementRenderer	57
car::system::device::DeviceManager	64
std::enable_shared_from_this	
car::system::CarSystem	42
car::system::messaging::MessagingSystem::FirstMessageStruct	71
car::configuration::JsonConfiguration	72
car::system::device::lidar::LidarDevice	73
car::system::device::lidar::LidarDummy	76
car::system::device::lidar::LidarScanner	78
car::display::console::screen::LoggingScreen	81
car::display::console::component::main::MainErrorModal	83
car::display::console::component::main::MainExitModal	84
car::display::console::screen::MainScreen	85
car::system::messaging::MessagingSystem	87
car::system::movement::MovementSystem	93
car::plugin::Plugin	97
behaviour_tree::BehaviourTreeHandler	32
car::plugin::PluginManager	98
car::display::console::component::settings::SettingsEditConfig	102
car::display::console::screen::SettingsScreen	104
TB6612	106

Chapter 9

Class Index

9.1 Class List

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

<code>car::system::movement::controller::AbstractMovementController</code>	25
<code>BackWheels</code>	29
<code>behaviour_tree::BehaviourTreeHandler</code>	32
<code>car::system::device::CameraDevice</code>	35
<code>car::display::console::CarConsole</code>	39
<code>behaviour_tree::CarContext</code>	41
<code>car::system::CarSystem</code>	42
<code>car::configuration::Configuration</code>	46
<code>car::display::console::component::main::ConnectButton</code>	49
<code>car::display::console::component::debug::DebugEnabledler</code>	50
<code>car::display::console::component::debug::DebugLidarCheckbox</code>	53
<code>car::display::console::component::debug::DebugMessagingTextbox</code>	55
<code>car::display::console::component::debug::DebugMovementRenderer</code>	57
<code>car::system::device::DeviceManager</code>	64
<code>car::system::movement::controller::DummyMovementController</code>	67
<code>car::system::messaging::MessagingSystem::FirstMessageStruct</code>	71
<code>car::configuration::JsonConfiguration</code>	72
<code>car::system::device::lidar::LidarDevice</code>	73
<code>car::system::device::lidar::LidarDummy</code>	76
<code>car::system::device::lidar::LidarScanner</code>	78
<code>car::display::console::screen::LoggingScreen</code>	81
<code>car::display::console::component::main::MainErrorModal</code>	83
<code>car::display::console::component::main::MainExitModal</code>	84
<code>car::display::console::screen::MainScreen</code>	85
<code>car::system::messaging::MessagingSystem</code>	87
<code>car::system::movement::MovementSystem</code>	93
<code>car::plugin::Plugin</code>	97
<code>car::plugin::PluginManager</code>	98
<code>rpi_daemon</code>	100
<code>car::display::console::component::settings::SettingsEditConfig</code>	102
<code>car::display::console::screen::SettingsScreen</code>	104
<code>TB6612</code>	106
<code>car::system::logging::VectorSink< Mutex ></code>	108

Chapter 10

File Index

10.1 File List

Here is a list of all files with brief descriptions:

behaviour_tree/src/main.cpp	111
common/include/behaviour_tree/BehaviourTreeHandler.hpp	114
common/include/behaviour_tree/CarContext.hpp	117
common/include/car/configuration/Configuration.h	118
common/include/car/plugin/Plugin.h	118
common/include/car/plugin/PluginManager.h	119
common/include/car/system/CarSystem.h	121
common/include/car/system/device/CameraDevice.h	122
common/include/car/system/device/DeviceManager.h	123
common/include/car/system/device/lidar/LidarDevice.h	125
common/include/car/system/device/lidar/LidarDummy.h	126
common/include/car/system/device/lidar/LidarScanner.h	126
common/include/car/system/logging/VectorSink.h	128
common/include/car/system/messaging/MessagingSystem.h	129
common/include/car/system/messaging/StreamType.h	131
common/include/car/system/movement/MovementSystem.h	136
common/include/car/system/movement/controller/AbstractMovementController.h	132
common/include/car/system/movement/controller/DeviceMovementController.h	133
common/include/car/system/movement/controller/DummyMovementController.h	134
common/include/car/system/movement/devices/RearWheel.h	135
common/include/car/system/movement/devices/Servo.h	135
common/src/car/system/CarSystem.cpp	138
common/src/car/system/device/CameraDevice.cpp	138
common/src/car/system/device/DeviceManager.cpp	138
common/src/car/system/messaging/MessagingSystem.cpp	139
common/src/car/system/movement/controller/DeviceMovementController.cpp	139
common/src/car/system/movement/controller/DummyMovementController.cpp	139
common/src/car/system/movement/devices/RearWheel.cpp	140
common/src/car/system/movement/devices/Servo.cpp	140
common/tests/pca9685/test_front_wheels.cpp	140
common/tests/tb6612/test_rear_wheels.cpp	141
daemon/src/main.cpp	112
repository/packages/t/tb6612/tb6612/include/TB6612.h	142
repository/packages/t/tb6612/tb6612/src/TB6612.cpp	143
tui/src/main.cpp	113

tui/src/car/configuration/JsonConfiguration.cxx	143
tui/src/car/display/console/CarConsole.cpp	144
tui/src/car/display/console/CarConsole.h	144
tui/src/car/display/console/component/debug/DebugEnabler.cxx	145
tui/src/car/display/console/component/debug/DebugLidarCheckbox.cxx	146
tui/src/car/display/console/component/debug/DebugMessagingTextbox.cxx	147
tui/src/car/display/console/component/debug/DebugMovementRenderer.cxx	147
tui/src/car/display/console/component/main/ConnectButton.cxx	148
tui/src/car/display/console/component/main/MainErrorModal.cxx	149
tui/src/car/display/console/component/main/MainExitModal.cxx	150
tui/src/car/display/console/component/settings/SettingsEditConfig.cxx	150
tui/src/car/display/console/screen/LoggingScreen.cxx	151
tui/src/car/display/console/screen/MainScreen.cxx	152
tui/src/car/display/console/screen/SettingsScreen.cxx	153

Chapter 11

Namespace Documentation

11.1 behaviour_tree Namespace Reference

Classes

- class [BehaviourTreeHandler](#)
- class [CarContext](#)

11.2 car Namespace Reference

Namespaces

- namespace [configuration](#)
- namespace [display](#)
- namespace [plugin](#)
- namespace [system](#)

11.3 car::configuration Namespace Reference

Classes

- struct [Configuration](#)
- class [JsonConfiguration](#)

11.4 car::display Namespace Reference

Namespaces

- namespace [console](#)

11.5 car::display::console Namespace Reference

Namespaces

- namespace [component](#)
- namespace [screen](#)

Classes

- class [CarConsole](#)

11.6 car::display::console::component Namespace Reference

Namespaces

- namespace [debug](#)
- namespace [main](#)
- namespace [settings](#)

11.7 car::display::console::component::debug Namespace Reference

Classes

- class [DebugEnabler](#)
- class [DebugLidarCheckbox](#)
- class [DebugMessagingTextbox](#)
- class [DebugMovementRenderer](#)

11.8 car::display::console::component::main Namespace Reference

Classes

- class [ConnectButton](#)
- class [MainErrorModal](#)
- class [MainExitModal](#)

11.9 car::display::console::component::settings Namespace Reference

Classes

- class [SettingsEditConfig](#)

11.10 car::display::console::screen Namespace Reference

Classes

- class [LoggingScreen](#)
- class [MainScreen](#)
- class [SettingsScreen](#)

11.11 car::plugin Namespace Reference

Classes

- class [Plugin](#)
- class [PluginManager](#)

11.12 car::system Namespace Reference

Namespaces

- namespace [device](#)
- namespace [logging](#)
- namespace [messaging](#)
- namespace [movement](#)

Classes

- class [CarSystem](#)

11.13 car::system::device Namespace Reference

Namespaces

- namespace [lidar](#)

Classes

- class [CameraDevice](#)
- class [DeviceManager](#)

11.14 car::system::device::lidar Namespace Reference

Classes

- class [LidarDevice](#)
- class [LidarDummy](#)
- class [LidarScanner](#)

11.15 car::system::logging Namespace Reference

Classes

- class [VectorSink](#)

Typedefs

- using [vector_sink_mt](#) = [VectorSink](#)< std::mutex >

11.15.1 Typedef Documentation

11.15.1.1 vector_sink_mt

using [car::system::logging::vector_sink_mt](#) = typedef [VectorSink](#)<std::mutex>

11.16 car::system::messaging Namespace Reference

Classes

- class [MessagingSystem](#)

11.17 car::system::movement Namespace Reference

Namespaces

- namespace [controller](#)

Classes

- class [MovementSystem](#)

11.18 car::system::movement::controller Namespace Reference

Classes

- class [AbstractMovementController](#)
- class [DummyMovementController](#)

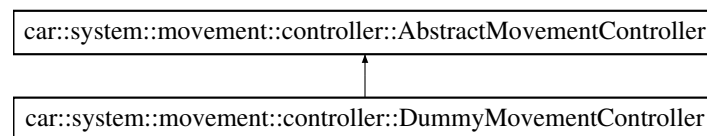
Chapter 12

Class Documentation

12.1 car::system::movement::controller::AbstractMovementController Class Reference

```
#include <AbstractMovementController.h>
```

Inheritance diagram for car::system::movement::controller::AbstractMovementController:



Public Member Functions

- virtual void [initialize](#) ()=0
- virtual void [stop](#) ()=0
- virtual void [terminate](#) ()=0
- virtual void [setRearWheelsSpeed](#) (const int speed)=0
- virtual void [setRearLeftWheelSpeed](#) (const int speed)=0
- virtual void [setRearRightWheelSpeed](#) (const int speed)=0
- virtual void [setFrontWheelsAngle](#) (const float angle)=0
- virtual void [setCameraServo1Angle](#) (const float angle)=0
- virtual void [setCameraServo2Angle](#) (const float angle)=0
- virtual void [setRearWheelsDirectionToForward](#) ()=0
- virtual void [setRearLeftWheelDirectionToForward](#) ()=0
- virtual void [setRearRightWheelDirectionToForward](#) ()=0
- virtual void [setRearWheelsDirectionToBackward](#) ()=0
- virtual void [setRearLeftWheelDirectionToBackward](#) ()=0
- virtual void [setRearRightWheelDirectionToBackward](#) ()=0

12.1.1 Member Function Documentation

12.1.1.1 initialize()

```
virtual void car::system::movement::controller::AbstractMovementController::initialize ( )  
[pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.2 setCameraServo1Angle()

```
virtual void car::system::movement::controller::AbstractMovementController::setCameraServo1↵  
Angle (   
        const float angle ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.3 setCameraServo2Angle()

```
virtual void car::system::movement::controller::AbstractMovementController::setCameraServo2↵  
Angle (   
        const float angle ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.4 setFrontWheelsAngle()

```
virtual void car::system::movement::controller::AbstractMovementController::setFrontWheels↵  
Angle (   
        const float angle ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.5 setRearLeftWheelDirectionToBackward()

```
virtual void car::system::movement::controller::AbstractMovementController::setRearLeftWheel↵  
DirectionToBackward ( ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.6 setRearLeftWheelDirectionToForward()

```
virtual void car::system::movement::controller::AbstractMovementController::setRearLeftWheel↵  
DirectionToForward ( ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.7 setRearLeftWheelSpeed()

```
virtual void car::system::movement::controller::AbstractMovementController::setRearLeftWheel↵  
Speed (   
        const int speed ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.8 setRearRightWheelDirectionToBackward()

```
virtual void car::system::movement::controller::AbstractMovementController::setRearRight↵  
WheelDirectionToBackward ( ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.9 setRearRightWheelDirectionToForward()

```
virtual void car::system::movement::controller::AbstractMovementController::setRearRight↵  
WheelDirectionToForward ( ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.10 setRearRightWheelSpeed()

```
virtual void car::system::movement::controller::AbstractMovementController::setRearRight↵  
WheelSpeed (   
        const int speed ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.11 setRearWheelsDirectionToBackward()

```
virtual void car::system::movement::controller::AbstractMovementController::setRearWheels↵  
DirectionToBackward ( ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.12 setRearWheelsDirectionToForward()

```
virtual void car::system::movement::controller::AbstractMovementController::setRearWheels↵  
DirectionToForward ( ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.13 setRearWheelsSpeed()

```
virtual void car::system::movement::controller::AbstractMovementController::setRearWheelsSpeed  
(  
    const int speed ) [pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.14 stop()

```
virtual void car::system::movement::controller::AbstractMovementController::stop ( ) [pure  
virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

12.1.1.15 terminate()

```
virtual void car::system::movement::controller::AbstractMovementController::terminate ( )  
[pure virtual]
```

Implemented in [car::system::movement::controller::DummyMovementController](#).

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

- [common/include/car/system/movement/controller/AbstractMovementController.h](#)

12.2 BackWheels Class Reference

Public Member Functions

- [BackWheels](#) (const int &bus_number=1)
- void [forward](#) ()
- void [backward](#) ()
- void [stop](#) ()
- int [getSpeed](#) () const
- void [setSpeed](#) (const int &[speed](#))
- void [ready](#) ()
- void [calibration](#) ()
- void [caliLeft](#) ()
- void [caliRight](#) ()
- void [caliOK](#) ()

Public Attributes

- PCA9685 [pca9685](#)

Private Attributes

- std::unique_ptr< [TB6612](#) > [left_wheel](#)
- std::unique_ptr< [TB6612](#) > [right_wheel](#)
- int [forward_A](#)
- int [forward_B](#)
- int [cali_forward_A](#)
- int [cali_forward_B](#)
- int [speed](#)

12.2.1 Constructor & Destructor Documentation

12.2.1.1 BackWheels()

```
BackWheels::BackWheels (  
    const int & bus_number = 1 ) [inline]
```

12.2.2 Member Function Documentation

12.2.2.1 backward()

```
void BackWheels::backward ( ) [inline]
```

12.2.2.2 calibration()

```
void BackWheels::calibration ( ) [inline]
```

12.2.2.3 caliLeft()

```
void BackWheels::caliLeft ( ) [inline]
```

12.2.2.4 caliOK()

```
void BackWheels::caliOK ( ) [inline]
```

12.2.2.5 caliRight()

```
void BackWheels::caliRight ( ) [inline]
```

12.2.2.6 forward()

```
void BackWheels::forward ( ) [inline]
```

12.2.2.7 getSpeed()

```
int BackWheels::getSpeed ( ) const [inline]
```

12.2.2.8 ready()

```
void BackWheels::ready ( ) [inline]
```

12.2.2.9 setSpeed()

```
void BackWheels::setSpeed (
    const int & speed ) [inline]
```

12.2.2.10 stop()

```
void BackWheels::stop ( ) [inline]
```

12.2.3 Member Data Documentation

12.2.3.1 cali_forward_A

```
int BackWheels::cali_forward_A [private]
```

12.2.3.2 cali_forward_B

```
int BackWheels::cali_forward_B [private]
```

12.2.3.3 forward_A

```
int BackWheels::forward_A [private]
```

12.2.3.4 forward_B

```
int BackWheels::forward_B [private]
```

12.2.3.5 left_wheel

```
std::unique_ptr<TB6612> BackWheels::left_wheel [private]
```

12.2.3.6 pca9685

```
PCA9685 BackWheels::pca9685
```

12.2.3.7 right_wheel

```
std::unique_ptr<TB6612> BackWheels::right_wheel [private]
```

12.2.3.8 speed

```
int BackWheels::speed [private]
```

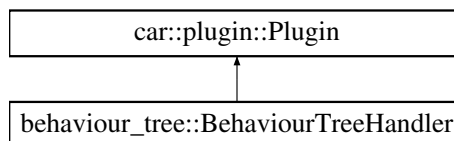
The documentation for this class was generated from the following file:

- [common/tests/tb6612/test_rear_wheels.cpp](#)

12.3 behaviour_tree::BehaviourTreeHandler Class Reference

```
#include <BehaviourTreeHandler.hpp>
```

Inheritance diagram for behaviour_tree::BehaviourTreeHandler:



Public Member Functions

- void [initialize](#) (std::shared_ptr< [car::system::CarSystem](#) > [car_system](#)) final override
- void [handleCommand](#) (const std::string message, const rapidjson::Document &message_json)
- void [setBehaviourTree](#) (const rapidjson::Document &message_json)
- void [startBehaviourTree](#) ()
- void [stopBehaviourTree](#) ()
- void [update](#) () final override
- void [stop](#) () final override
- std::string [getName](#) () final override
- void [_setBehaviourTree](#) (std::shared_ptr< BehaviourTree > [behaviour_tree](#))

Private Attributes

- std::shared_ptr< [car::system::CarSystem](#) > [car_system](#)
- std::shared_ptr< BehaviourTree > [behaviour_tree](#)
- std::shared_ptr< Context > [context](#)
- int [tick_count](#) = 0
- std::chrono::time_point< std::chrono::steady_clock > [last_connected](#)

12.3.1 Member Function Documentation

12.3.1.1 `_setBehaviourTree()`

```
void behaviour_tree::BehaviourTreeHandler::_setBehaviourTree (  
    std::shared_ptr< BehaviourTree > behaviour_tree ) [inline]
```

12.3.1.2 `getName()`

```
std::string behaviour_tree::BehaviourTreeHandler::getName ( ) [inline], [final], [override],  
[virtual]
```

Implements [car::plugin::Plugin](#).

12.3.1.3 `handleCommand()`

```
void behaviour_tree::BehaviourTreeHandler::handleCommand (  
    const std::string message,  
    const rapidjson::Document & message_json ) [inline]
```

12.3.1.4 `initialize()`

```
void behaviour_tree::BehaviourTreeHandler::initialize (  
    std::shared_ptr< car::system::CarSystem > car_system ) [inline], [final], [override],  
[virtual]
```

Implements [car::plugin::Plugin](#).

12.3.1.5 `setBehaviourTree()`

```
void behaviour_tree::BehaviourTreeHandler::setBehaviourTree (  
    const rapidjson::Document & message_json ) [inline]
```

12.3.1.6 startBehaviourTree()

```
void behaviour_tree::BehaviourTreeHandler::startBehaviourTree ( ) [inline]
```

12.3.1.7 stop()

```
void behaviour_tree::BehaviourTreeHandler::stop ( ) [inline], [final], [override], [virtual]
```

Implements [car::plugin::Plugin](#).

12.3.1.8 stopBehaviourTree()

```
void behaviour_tree::BehaviourTreeHandler::stopBehaviourTree ( ) [inline]
```

12.3.1.9 update()

```
void behaviour_tree::BehaviourTreeHandler::update ( ) [inline], [final], [override], [virtual]
```

Implements [car::plugin::Plugin](#).

12.3.2 Member Data Documentation

12.3.2.1 behaviour_tree

```
std::shared_ptr<BehaviourTree> behaviour_tree::BehaviourTreeHandler::behaviour_tree [private]
```

12.3.2.2 car_system

```
std::shared_ptr<car::system::CarSystem> behaviour_tree::BehaviourTreeHandler::car_system  
[private]
```

12.3.2.3 context

```
std::shared_ptr<Context> behaviour_tree::BehaviourTreeHandler::context [private]
```

12.3.2.4 last_connected

```
std::chrono::time_point<std::chrono::steady_clock> behaviour_tree::BehaviourTreeHandler↵
::last_connected [private]
```

12.3.2.5 tick_count

```
int behaviour_tree::BehaviourTreeHandler::tick_count = 0 [private]
```

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

- common/include/behaviour_tree/[BehaviourTreeHandler.hpp](#)

12.4 car::system::device::CameraDevice Class Reference

```
#include <CameraDevice.h>
```

Public Member Functions

- [CameraDevice](#) (std::shared_ptr< [configuration::Configuration](#) > [configuration](#))
- [CameraDevice](#) (const [CameraDevice](#) &)=delete
- [CameraDevice](#) & operator= (const [CameraDevice](#) &)=delete
- [CameraDevice](#) ([CameraDevice](#) &&)=delete
- [CameraDevice](#) & operator= ([CameraDevice](#) &&)=delete
- [~CameraDevice](#) ()=default
- std::string [getFrameBuffer](#) () const

Static Public Member Functions

- static tl::expected< std::unique_ptr< [CameraDevice](#) >, std::string > [create](#) (std::shared_ptr< [configuration::Configuration](#) > [configuration](#))

Protected Member Functions

- void [start](#) ()
- void [update](#) ()
- void [stop](#) ()
- void [disconnect](#) ()
- void [terminate](#) ()

Private Attributes

- `std::shared_ptr< configuration::Configuration > configuration`
- `std::unique_ptr< cv::VideoCapture > camera_`
- `bool connected_ = false`
- `std::string frame_buffer_`
- `std::mutex camera_mutex_`
- `std::chrono::steady_clock::time_point last`

Friends

- class [DeviceManager](#)

12.4.1 Constructor & Destructor Documentation

12.4.1.1 CameraDevice() [1/3]

```
car::system::device::CameraDevice::CameraDevice (  
    std::shared_ptr< configuration::Configuration > configuration ) [inline]
```

12.4.1.2 CameraDevice() [2/3]

```
car::system::device::CameraDevice::CameraDevice (  
    const CameraDevice & ) [delete]
```

12.4.1.3 CameraDevice() [3/3]

```
car::system::device::CameraDevice::CameraDevice (  
    CameraDevice && ) [delete]
```

12.4.1.4 ~CameraDevice()

```
car::system::device::CameraDevice::~~CameraDevice ( ) [default]
```

12.4.2 Member Function Documentation

12.4.2.1 create()

```
tl::expected< std::unique_ptr< CameraDevice >, std::string > car::system::device::CameraDevice::create (
    std::shared_ptr< configuration::Configuration > configuration ) [static]
```

12.4.2.2 disconnect()

```
void car::system::device::CameraDevice::disconnect ( ) [protected]
```

12.4.2.3 getFrameBuffer()

```
std::string car::system::device::CameraDevice::getFrameBuffer ( ) const
```

12.4.2.4 operator=() [1/2]

```
CameraDevice & car::system::device::CameraDevice::operator= (
    CameraDevice && ) [delete]
```

12.4.2.5 operator=() [2/2]

```
CameraDevice & car::system::device::CameraDevice::operator= (
    const CameraDevice & ) [delete]
```

12.4.2.6 start()

```
void car::system::device::CameraDevice::start ( ) [protected]
```

12.4.2.7 stop()

```
void car::system::device::CameraDevice::stop ( ) [protected]
```

12.4.2.8 terminate()

```
void car::system::device::CameraDevice::terminate ( ) [protected]
```

12.4.2.9 update()

```
void car::system::device::CameraDevice::update ( ) [protected]
```

12.4.3 Friends And Related Function Documentation

12.4.3.1 DeviceManager

```
friend class DeviceManager [friend]
```

12.4.4 Member Data Documentation

12.4.4.1 camera_

```
std::unique_ptr<cv::VideoCapture> car::system::device::CameraDevice::camera_ [private]
```

12.4.4.2 camera_mutex_

```
std::mutex car::system::device::CameraDevice::camera_mutex_ [private]
```

12.4.4.3 configuration

```
std::shared_ptr<configuration::Configuration> car::system::device::CameraDevice::configuration  
[private]
```

12.4.4.4 connected_

```
bool car::system::device::CameraDevice::connected_ = false [private]
```

12.4.4.5 frame_buffer_

```
std::string car::system::device::CameraDevice::frame_buffer_ [private]
```

12.4.4.6 last

```
std::chrono::steady_clock::time_point car::system::device::CameraDevice::last [private]
```

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

- common/include/car/system/device/[CameraDevice.h](#)
- common/src/car/system/device/[CameraDevice.cpp](#)

12.5 car::display::console::CarConsole Class Reference

```
#include <CarConsole.h>
```

Public Member Functions

- [CarConsole](#) (std::shared_ptr< [CarSystem](#) > car_system, std::shared_ptr< [JsonConfiguration](#) > json_configuration, std::shared_ptr< [logging::vector_sink_mt](#) > vector_sink)
- void [initialize](#) ()
- void [run](#) ()
- void [terminate](#) ()

Private Attributes

- std::shared_ptr< [CarSystem](#) > car_system
- std::shared_ptr< [JsonConfiguration](#) > json_configuration
- std::shared_ptr< [logging::vector_sink_mt](#) > vector_sink

12.5.1 Constructor & Destructor Documentation

12.5.1.1 CarConsole()

```
car::display::console::CarConsole::CarConsole (
    std::shared_ptr< CarSystem > car_system,
    std::shared_ptr< JsonConfiguration > json_configuration,
    std::shared_ptr< logging::vector_sink_mt > vector_sink )
```

12.5.2 Member Function Documentation

12.5.2.1 initialize()

```
void car::display::console::CarConsole::initialize ( )
```

12.5.2.2 run()

```
void car::display::console::CarConsole::run ( )
```

12.5.2.3 terminate()

```
void car::display::console::CarConsole::terminate ( )
```

12.5.3 Member Data Documentation

12.5.3.1 car_system

```
std::shared_ptr<CarSystem> car::display::console::CarConsole::car_system [private]
```

12.5.3.2 json_configuration

```
std::shared_ptr<JsonConfiguration> car::display::console::CarConsole::json_configuration
[private]
```

12.5.3.3 vector_sink

```
std::shared_ptr<logging::vector_sink_mt> car::display::console::CarConsole::vector_sink [private]
```

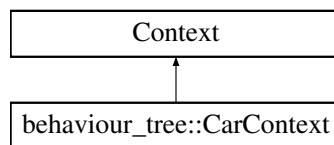
The documentation for this class was generated from the following files:

- [tui/src/car/display/console/CarConsole.h](#)
- [tui/src/car/display/console/CarConsole.cpp](#)

12.6 behaviour_tree::CarContext Class Reference

```
#include <CarContext.hpp>
```

Inheritance diagram for behaviour_tree::CarContext:



Public Member Functions

- [CarContext](#) (std::shared_ptr< BehaviourTree > behaviour_tree, std::shared_ptr< [car::system::CarSystem](#) > [car_system](#))
- std::shared_ptr< [car::system::CarSystem](#) > [getCarSystem](#) () const
- void [_](#) () override

Private Attributes

- std::shared_ptr< [car::system::CarSystem](#) > [car_system](#)

12.6.1 Constructor & Destructor Documentation

12.6.1.1 CarContext()

```
behaviour_tree::CarContext::CarContext (
    std::shared_ptr< BehaviourTree > behaviour_tree,
    std::shared_ptr< car::system::CarSystem > car_system ) [inline]
```

12.6.2 Member Function Documentation

12.6.2.1 _()

```
void behaviour_tree::CarContext::_ ( ) [inline], [override]
```

12.6.2.2 getCarSystem()

```
std::shared_ptr< car::system::CarSystem > behaviour_tree::CarContext::getCarSystem ( ) const  
[inline]
```

12.6.3 Member Data Documentation

12.6.3.1 car_system

```
std::shared_ptr<car::system::CarSystem> behaviour_tree::CarContext::car_system [private]
```

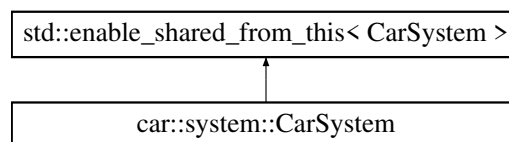
The documentation for this class was generated from the following file:

- common/include/behaviour_tree/[CarContext.hpp](#)

12.7 car::system::CarSystem Class Reference

```
#include <CarSystem.h>
```

Inheritance diagram for car::system::CarSystem:



Public Member Functions

- [CarSystem](#) (std::shared_ptr< [Configuration](#) > configuration, std::unique_ptr< [DeviceManager](#) > device_↵ manager, std::unique_ptr< [MessagingSystem](#) > messaging_system, std::unique_ptr< [MovementSystem](#) > movement_system, std::unique_ptr< [PluginManager](#) > plugin_manager)
 - void [initialize](#) ()
 - void [reload](#) ()
 - void [start](#) ()
 - void [stop](#) ()
 - tl::expected< nullptr_t, std::string > [tryConnect](#) ()
 - void [disconnect](#) ()
 - void [terminate](#) ()
- Only devices should be terminated here since destructor does not work when the program is terminated by the user.*
- void [update](#) ()
 - const std::shared_ptr< [Configuration](#) > [getConfiguration](#) () const
 - void [setConfiguration](#) (std::shared_ptr< [Configuration](#) > configuration)
 - [DeviceManager](#) * [getDeviceManager](#) () const
 - [MessagingSystem](#) * [getMessagingSystem](#) () const
 - [MovementSystem](#) * [getMovementSystem](#) () const
 - template<typename T >
const std::shared_ptr< T > [getPlugin](#) () const

Private Member Functions

- void [sendData](#) ()

Private Attributes

- std::shared_ptr< [Configuration](#) > [configuration_](#)
- const std::unique_ptr< [DeviceManager](#) > [device_manager_](#)
- const std::unique_ptr< [MessagingSystem](#) > [messaging_system_](#)
- const std::unique_ptr< [MovementSystem](#) > [movement_system_](#)
- const std::unique_ptr< [PluginManager](#) > [plugin_manager_](#)
- bool [initialized](#) = false
- bool [started](#) = false

12.7.1 Constructor & Destructor Documentation

12.7.1.1 CarSystem()

```
car::system::CarSystem::CarSystem (
    std::shared_ptr< Configuration > configuration,
    std::unique_ptr< DeviceManager > device\_manager,
    std::unique_ptr< MessagingSystem > messaging\_system,
    std::unique_ptr< MovementSystem > movement\_system,
    std::unique_ptr< PluginManager > plugin\_manager )
```

12.7.2 Member Function Documentation

12.7.2.1 disconnect()

```
void car::system::CarSystem::disconnect ( )
```

12.7.2.2 getConfiguration()

```
const std::shared_ptr< Configuration > car::system::CarSystem::getConfiguration ( ) const
[inline]
```

12.7.2.3 getDeviceManager()

```
DeviceManager * car::system::CarSystem::getDeviceManager ( ) const [inline]
```

12.7.2.4 getMessagingSystem()

```
MessagingSystem * car::system::CarSystem::getMessagingSystem ( ) const [inline]
```

12.7.2.5 getMovementSystem()

```
MovementSystem * car::system::CarSystem::getMovementSystem ( ) const [inline]
```

12.7.2.6 getPlugin()

```
template<typename T >  
const std::shared_ptr< T > car::system::CarSystem::getPlugin ( ) const [inline]
```

12.7.2.7 initialize()

```
void car::system::CarSystem::initialize ( )
```

12.7.2.8 reload()

```
void car::system::CarSystem::reload ( )
```

12.7.2.9 sendData()

```
void car::system::CarSystem::sendData ( ) [private]
```


12.7.2.10 setConfiguration()

```
void car::system::CarSystem::setConfiguration (
    std::shared_ptr< Configuration > configuration )
```

12.7.2.11 start()

```
void car::system::CarSystem::start ( )
```

12.7.2.12 stop()

```
void car::system::CarSystem::stop ( )
```

12.7.2.13 terminate()

```
void car::system::CarSystem::terminate ( )
```

Only devices should be terminated here since destructor does not work when the program is terminated by the user.

12.7.2.14 tryConnect()

```
tl::expected< nullptr_t, std::string > car::system::CarSystem::tryConnect ( )
```

12.7.2.15 update()

```
void car::system::CarSystem::update ( )
```

12.7.3 Member Data Documentation

12.7.3.1 configuration_

```
std::shared_ptr<Configuration> car::system::CarSystem::configuration_ [private]
```

12.7.3.2 device_manager_

```
const std::unique_ptr<DeviceManager> car::system::CarSystem::device_manager_ [private]
```

12.7.3.3 initialized

```
bool car::system::CarSystem::initialized = false [private]
```

12.7.3.4 messaging_system_

```
const std::unique_ptr<MessagingSystem> car::system::CarSystem::messaging_system_ [private]
```

12.7.3.5 movement_system_

```
const std::unique_ptr<MovementSystem> car::system::CarSystem::movement_system_ [private]
```

12.7.3.6 plugin_manager_

```
const std::unique_ptr<PluginManager> car::system::CarSystem::plugin_manager_ [private]
```

12.7.3.7 started

```
bool car::system::CarSystem::started = false [private]
```

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

- [common/include/car/system/CarSystem.h](#)
- [common/src/car/system/CarSystem.cpp](#)

12.8 car::configuration::Configuration Struct Reference

```
#include <Configuration.h>
```

Public Member Functions

- void [setCameraFps](#) (const int [camera_fps](#))
- const int [getCameraFpsInterval](#) ()

Public Attributes

- std::string [host](#) = "127.0.0.1:3000"
- int [camera_index](#) = 0
- bool [use_camera](#) = true
- std::string [lidar_port](#) = ""
- bool [use_lidar](#) = true
- std::chrono::milliseconds [behaviour_tree_update_ms_interval](#) = std::chrono::milliseconds(100)

Private Attributes

- int [camera_fps](#) = 60
- int [camera_fps_interval](#) = 1000

12.8.1 Member Function Documentation

12.8.1.1 [getCameraFpsInterval\(\)](#)

```
const int car::configuration::Configuration::getCameraFpsInterval ( ) [inline]
```

12.8.1.2 [setCameraFps\(\)](#)

```
void car::configuration::Configuration::setCameraFps (
    const int camera_fps ) [inline]
```

12.8.2 Member Data Documentation

12.8.2.1 [behaviour_tree_update_ms_interval](#)

```
std::chrono::milliseconds car::configuration::Configuration::behaviour_tree_update_ms_interval
= std::chrono::milliseconds(100)
```

12.8.2.2 camera_fps

```
int car::configuration::Configuration::camera_fps = 60 [private]
```

12.8.2.3 camera_fps_interval

```
int car::configuration::Configuration::camera_fps_interval = 1000 [private]
```

12.8.2.4 camera_index

```
int car::configuration::Configuration::camera_index = 0
```

12.8.2.5 host

```
std::string car::configuration::Configuration::host = "127.0.0.1:3000"
```

12.8.2.6 lidar_port

```
std::string car::configuration::Configuration::lidar_port = ""
```

12.8.2.7 use_camera

```
bool car::configuration::Configuration::use_camera = true
```

12.8.2.8 use_lidar

```
bool car::configuration::Configuration::use_lidar = true
```

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

- [common/include/car/configuration/Configuration.h](#)

12.9 car::display::console::component::main::ConnectButton Class Reference

Public Member Functions

- [ConnectButton](#) (std::shared_ptr< [CarSystem](#) > *car_system*, Box &box)
- Component [element](#) ()

Public Attributes

- std::function< void(std::string)> [on_connect_failure](#) = [] (std::string _) {}

Private Attributes

- std::shared_ptr< [CarSystem](#) > *car_system*
- bool [main_debounce](#) = false
- bool [button_pressed](#) = false
- std::string [main_button_text](#) = "Start Car Application"
- Component [main_button](#)

12.9.1 Constructor & Destructor Documentation

12.9.1.1 ConnectButton()

```
car::display::console::component::main::ConnectButton::ConnectButton (
    std::shared_ptr< CarSystem > car_system,
    Box & box ) [inline]
```

12.9.2 Member Function Documentation

12.9.2.1 element()

```
Component car::display::console::component::main::ConnectButton::element ( ) [inline]
```

12.9.3 Member Data Documentation

12.9.3.1 button_pressed

```
bool car::display::console::component::main::ConnectButton::button_pressed = false [private]
```

12.9.3.2 car_system

```
std::shared_ptr<CarSystem> car::display::console::component::main::ConnectButton::car_system  
[private]
```

12.9.3.3 main_button

```
Component car::display::console::component::main::ConnectButton::main_button [private]
```

12.9.3.4 main_button_text

```
std::string car::display::console::component::main::ConnectButton::main_button_text = "Start  
Car Application" [private]
```

12.9.3.5 main_debounce

```
bool car::display::console::component::main::ConnectButton::main_debounce = false [private]
```

12.9.3.6 on_connect_failure

```
std::function<void(std::string)> car::display::console::component::main::ConnectButton::on_↵  
connect_failure = [] (std::string _) {}
```

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

- tui/src/car/display/console/component/main/[ConnectButton.cxx](#)

12.10 car::display::console::component::debug::DebugEnableder Class Reference

Public Member Functions

- Component [getCheckbox](#) ()
- ComponentDecorator [getWarningModal](#) ()
- const bool & [isEnabled](#) () const

Private Attributes

- bool `debounce` = false
- bool `enabled` = false
- bool `checkbox_value` = false
- bool `display_warn_debug_modal` = false
- std::string `status` = `DEBUG_MODE_DISABLED_MESSAGE`
- Component `component`

Static Private Attributes

- static constexpr auto `DEBUG_ENABLE_WARNING_MESSAGE` = "Enabling debug mode temporarily disables connecting to online. Are you sure you want to do this?"
- static constexpr auto `DEBUG_MODE_ENABLED_MESSAGE` = "Debug Status: Enabled"
- static constexpr auto `DEBUG_MODE_DISABLED_MESSAGE` = "Debug Status: Disabled"
- static constexpr auto `DEBUG_MODE_WAIT_MESSAGE` = "Debug Status: Waiting for user input..."

12.10.1 Member Function Documentation

12.10.1.1 getCheckbox()

Component car::display::console::component::debug::DebugEnabler::getCheckbox () [inline]

12.10.1.2 getWarningModal()

ComponentDecorator car::display::console::component::debug::DebugEnabler::getWarningModal () [inline]

12.10.1.3 isEnabled()

const bool & car::display::console::component::debug::DebugEnabler::isEnabled () const [inline]

12.10.2 Member Data Documentation

12.10.2.1 checkbox_value

bool car::display::console::component::debug::DebugEnabler::checkbox_value = false [private]

12.10.2.2 component

```
Component car::display::console::component::debug::DebugEnabler::component [private]
```

12.10.2.3 debounce

```
bool car::display::console::component::debug::DebugEnabler::debounce = false [private]
```

12.10.2.4 DEBUG_ENABLE_WARNING_MESSAGE

```
constexpr auto car::display::console::component::debug::DebugEnabler::DEBUG_ENABLE_WARNING_↵  
_MESSAGE = "Enabling debug mode temporarily disables connecting to online. Are you sure you  
want to do this?" [static], [constexpr], [private]
```

12.10.2.5 DEBUG_MODE_DISABLED_MESSAGE

```
constexpr auto car::display::console::component::debug::DebugEnabler::DEBUG_MODE_DISABLED_↵  
MESSAGE = "Debug Status: Disabled" [static], [constexpr], [private]
```

12.10.2.6 DEBUG_MODE_ENABLED_MESSAGE

```
constexpr auto car::display::console::component::debug::DebugEnabler::DEBUG_MODE_ENABLED_↵  
MESSAGE = "Debug Status: Enabled" [static], [constexpr], [private]
```

12.10.2.7 DEBUG_MODE_WAIT_MESSAGE

```
constexpr auto car::display::console::component::debug::DebugEnabler::DEBUG_MODE_WAIT_MESSAGE  
= "Debug Status: Waiting for user input..." [static], [constexpr], [private]
```

12.10.2.8 display_warn_debug_modal

```
bool car::display::console::component::debug::DebugEnabler::display_warn_debug_modal = false  
[private]
```


12.10.2.9 enabled

```
bool car::display::console::component::debug::DebugEnabler::enabled = false [private]
```

12.10.2.10 status

```
std::string car::display::console::component::debug::DebugEnabler::status = DEBUG_MODE_DISABLED_MESSAGE [private]
```

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

- tui/src/car/display/console/component/debug/[DebugEnabler.cxx](#)

12.11 car::display::console::component::debug::DebugLidarCheckbox Class Reference

Public Member Functions

- [DebugLidarCheckbox](#) ()
- Component [element](#) ()
- nod::signal< void(bool)> & [getLidarMotorSignal](#) ()

Private Attributes

- nod::signal< void(bool)> [lidar_motor_signal](#)
- std::string [lidar_motor_status](#) = LIDAR_MOTOR_DISABLED_MESSAGE
- bool [lidar_motor_loading_debounce](#) = false
- bool [lidar_motor_enabled](#) = false
- Component [lidar_motor_checkbox_component](#)

Static Private Attributes

- static constexpr auto [LIDAR_MOTOR_ENABLED_MESSAGE](#) = "Lidar Motor Status: Enabled"
- static constexpr auto [LIDAR_MOTOR_DISABLED_MESSAGE](#) = "Lidar Motor Status: Disconnected"

12.11.1 Constructor & Destructor Documentation

12.11.1.1 DebugLidarCheckbox()

```
car::display::console::component::debug::DebugLidarCheckbox::DebugLidarCheckbox ( ) [inline]
```

12.11.2 Member Function Documentation

12.11.2.1 element()

```
Component car::display::console::component::debug::DebugLidarCheckbox::element ( ) [inline]
```

12.11.2.2 getLidarMotorSignal()

```
nod::signal< void(bool)> & car::display::console::component::debug::DebugLidarCheckbox::get↔  
LidarMotorSignal ( ) [inline]
```

12.11.3 Member Data Documentation

12.11.3.1 lidar_motor_checkbox_component

```
Component car::display::console::component::debug::DebugLidarCheckbox::lidar_motor_checkbox_↔  
component [private]
```

12.11.3.2 LIDAR_MOTOR_DISABLED_MESSAGE

```
constexpr auto car::display::console::component::debug::DebugLidarCheckbox::LIDAR_MOTOR_↔  
DISABLED_MESSAGE = "Lidar Motor Status: Disconnected" [static], [constexpr], [private]
```

12.11.3.3 lidar_motor_enabled

```
bool car::display::console::component::debug::DebugLidarCheckbox::lidar_motor_enabled = false  
[private]
```

12.11.3.4 LIDAR_MOTOR_ENABLED_MESSAGE

```
constexpr auto car::display::console::component::debug::DebugLidarCheckbox::LIDAR_MOTOR_↔  
ENABLED_MESSAGE = "Lidar Motor Status: Enabled" [static], [constexpr], [private]
```

12.11.3.5 lidar_motor_loading_debounce

```
bool car::display::console::component::debug::DebugLidarCheckbox::lidar_motor_loading_debounce
= false [private]
```

12.11.3.6 lidar_motor_signal

```
nod::signal<void(bool)> car::display::console::component::debug::DebugLidarCheckbox::lidar_↵
motor_signal [private]
```

12.11.3.7 lidar_motor_status

```
std::string car::display::console::component::debug::DebugLidarCheckbox::lidar_motor_status =
LIDAR_MOTOR_DISABLED_MESSAGE [private]
```

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

- tui/src/car/display/console/component/debug/[DebugLidarCheckbox.cxx](#)

12.12 car::display::console::component::debug::DebugMessaging↵ Textbox Class Reference

Public Member Functions

- [DebugMessagingTextbox](#) (nod::signal< void(const std::string)> &[message_signal](#))
- ftxui::Component [element](#) ()

Private Attributes

- std::string [message](#)
- Component [messaging_title](#)
- Component [messaging_textbox](#)
- Component [messaging_container](#)
- nod::signal< void(const std::string)> & [message_signal](#)

12.12.1 Constructor & Destructor Documentation

12.12.1.1 DebugMessagingTextbox()

```
car::display::console::component::debug::DebugMessagingTextbox::DebugMessagingTextbox (
    nod::signal< void(const std::string)> & message_signal ) [inline]
```

12.12.2 Member Function Documentation

12.12.2.1 element()

```
ftxui::Component car::display::console::component::debug::DebugMessagingTextbox::element ( )
[inline]
```

12.12.3 Member Data Documentation

12.12.3.1 message

```
std::string car::display::console::component::debug::DebugMessagingTextbox::message [private]
```

12.12.3.2 message_signal

```
nod::signal<void(const std::string)>& car::display::console::component::debug::DebugMessaging←
Textbox::message_signal [private]
```

12.12.3.3 messaging_container

```
Component car::display::console::component::debug::DebugMessagingTextbox::messaging_container
[private]
```

12.12.3.4 messaging_textbox

```
Component car::display::console::component::debug::DebugMessagingTextbox::messaging_textbox
[private]
```

12.12.3.5 messaging_title

Component car::display::console::component::debug::DebugMessagingTextbox::messaging_title
[private]

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

- [tui/src/car/display/console/component/debug/DebugMessagingTextbox.cxx](#)

12.13 car::display::console::component::debug::DebugMovementRenderer Class Reference

Public Member Functions

- [DebugMovementRenderer](#) ()
- [ftxui::Component element](#) ()
- [bool updateFrontWheels](#) ()
- [bool updateCameraServo1](#) ()
- [bool updateCameraServo2](#) ()
- [bool updateRearWheels](#) ()
- [nod::signal< void\(bool\)> & getRearWheelDirectionSignal](#) ()
- [const int getFrontWheelsAngleSliderValue](#) () const
- [const int getCameraServo1AngleSliderValue](#) () const
- [const int getCameraServo2AngleSliderValue](#) () const
- [const int getRearLeftWheelSpeedSliderValue](#) () const
- [const int getRearRightWheelSpeedSliderValue](#) () const

Private Attributes

- [nod::signal< void\(bool\)> rear_wheel_direction_signal](#)
- [int previous_rear_wheels_speed_slider_value = DEFAULT_REAR_WHEEL_SPEED](#)
- [int rear_wheels_speed_slider_value = DEFAULT_REAR_WHEEL_SPEED](#)
- [int previous_rear_left_wheel_speed_slider_value = DEFAULT_REAR_WHEEL_SPEED](#)
- [int rear_left_wheel_speed_slider_value = DEFAULT_REAR_WHEEL_SPEED](#)
- [int previous_rear_right_wheel_speed_slider_value = DEFAULT_REAR_WHEEL_SPEED](#)
- [int rear_right_wheel_speed_slider_value = DEFAULT_REAR_WHEEL_SPEED](#)
- [int previous_front_wheels_angle_slider_value = DEFAULT_FRONT_WHEEL_ANGLE](#)
- [int front_wheels_angle_slider_value = DEFAULT_FRONT_WHEEL_ANGLE](#)
- [int previous_camera_servo_1_angle_slider_angle = DEFAULT_FRONT_WHEEL_ANGLE](#)
- [int camera_servo_1_angle_slider_angle = DEFAULT_FRONT_WHEEL_ANGLE](#)
- [int previous_camera_servo_2_angle_slider_angle = DEFAULT_FRONT_WHEEL_ANGLE](#)
- [int camera_servo_2_angle_slider_angle = DEFAULT_FRONT_WHEEL_ANGLE](#)
- [bool rear_wheel_direction_debounce = false](#)
- [std::string rear_wheel_direction_status = REAR_WHEEL_DIRECTION_FORWARD_MESSAGE](#)
- [bool rear_wheel_direction = true](#)
- [Component rear_wheel_speed_slider](#)
- [Component rear_left_wheel_speed_slider](#)
- [Component rear_right_wheel_speed_slider](#)
- [Component rear_wheel_direction_checkbox_component](#)
- [Component front_wheels_angle_slider](#)
- [Component camera_servo_1_angle_slider](#)
- [Component camera_servo_2_angle_slider](#)
- [Component rear_wheel_menu_entry](#)
- [Component servo_menu_entry](#)
- [Component slider_container](#)

Static Private Attributes

- static constexpr int [DEFAULT_REAR_WHEEL_SPEED](#) = 0
- static constexpr int [DEFAULT_FRONT_WHEEL_ANGLE](#) = 90
- static constexpr auto [REAR_WHEEL_DIRECTION_FORWARD_MESSAGE](#) = "Rear Wheel Direction: Forward"
- static constexpr auto [REAR_WHEEL_DIRECTION_BACKWARD_MESSAGE](#) = "Rear Wheel Direction: Backward"

12.13.1 Constructor & Destructor Documentation

12.13.1.1 DebugMovementRenderer()

```
car::display::console::component::debug::DebugMovementRenderer::DebugMovementRenderer ( )
[inline]
```

12.13.2 Member Function Documentation

12.13.2.1 element()

```
ftxui::Component car::display::console::component::debug::DebugMovementRenderer::element ( )
[inline]
```

12.13.2.2 getCameraServo1AngleSliderValue()

```
const int car::display::console::component::debug::DebugMovementRenderer::getCameraServo1↔
AngleSliderValue ( ) const [inline]
```

12.13.2.3 getCameraServo2AngleSliderValue()

```
const int car::display::console::component::debug::DebugMovementRenderer::getCameraServo2↔
AngleSliderValue ( ) const [inline]
```

12.13.2.4 getFrontWheelsAngleSliderValue()

```
const int car::display::console::component::debug::DebugMovementRenderer::getFrontWheels↵  
AngleSliderValue ( ) const [inline]
```

12.13.2.5 getRearLeftWheelSpeedSliderValue()

```
const int car::display::console::component::debug::DebugMovementRenderer::getRearLeftWheel↵  
SpeedSliderValue ( ) const [inline]
```

12.13.2.6 getRearRightWheelSpeedSliderValue()

```
const int car::display::console::component::debug::DebugMovementRenderer::getRearRightWheel↵  
SpeedSliderValue ( ) const [inline]
```

12.13.2.7 getRearWheelDirectionSignal()

```
nod::signal< void(bool)> & car::display::console::component::debug::DebugMovementRenderer↵  
::getRearWheelDirectionSignal ( ) [inline]
```

12.13.2.8 updateCameraServo1()

```
bool car::display::console::component::debug::DebugMovementRenderer::updateCameraServo1 ( )  
[inline]
```

12.13.2.9 updateCameraServo2()

```
bool car::display::console::component::debug::DebugMovementRenderer::updateCameraServo2 ( )  
[inline]
```

12.13.2.10 updateFrontWheels()

```
bool car::display::console::component::debug::DebugMovementRenderer::updateFrontWheels ( )  
[inline]
```

12.13.2.11 updateRearWheels()

```
bool car::display::console::component::debug::DebugMovementRenderer::updateRearWheels ( )  
[inline]
```

12.13.3 Member Data Documentation

12.13.3.1 camera_servo_1_angle_slider

```
Component car::display::console::component::debug::DebugMovementRenderer::camera_servo_1_↔  
angle_slider [private]
```

12.13.3.2 camera_servo_1_angle_slider_angle

```
int car::display::console::component::debug::DebugMovementRenderer::camera_servo_1_angle_↔  
slider_angle = DEFAULT_FRONT_WHEEL_ANGLE [private]
```

12.13.3.3 camera_servo_2_angle_slider

```
Component car::display::console::component::debug::DebugMovementRenderer::camera_servo_2_↔  
angle_slider [private]
```

12.13.3.4 camera_servo_2_angle_slider_angle

```
int car::display::console::component::debug::DebugMovementRenderer::camera_servo_2_angle_↔  
slider_angle = DEFAULT_FRONT_WHEEL_ANGLE [private]
```

12.13.3.5 DEFAULT_FRONT_WHEEL_ANGLE

```
constexpr int car::display::console::component::debug::DebugMovementRenderer::DEFAULT_FRONT_↔  
WHEEL_ANGLE = 90 [static], [constexpr], [private]
```


12.13.3.6 DEFAULT_REAR_WHEEL_SPEED

```
constexpr int car::display::console::component::debug::DebugMovementRenderer::DEFAULT_REAR_WHEEL_SPEED = 0 [static], [constexpr], [private]
```

12.13.3.7 front_wheels_angle_slider

```
Component car::display::console::component::debug::DebugMovementRenderer::front_wheels_angle_slider [private]
```

12.13.3.8 front_wheels_angle_slider_value

```
int car::display::console::component::debug::DebugMovementRenderer::front_wheels_angle_slider_value = DEFAULT_FRONT_WHEEL_ANGLE [private]
```

12.13.3.9 previous_camera_servo_1_angle_slider_angle

```
int car::display::console::component::debug::DebugMovementRenderer::previous_camera_servo_1_angle_slider_angle = DEFAULT_FRONT_WHEEL_ANGLE [private]
```

12.13.3.10 previous_camera_servo_2_angle_slider_angle

```
int car::display::console::component::debug::DebugMovementRenderer::previous_camera_servo_2_angle_slider_angle = DEFAULT_FRONT_WHEEL_ANGLE [private]
```

12.13.3.11 previous_front_wheels_angle_slider_value

```
int car::display::console::component::debug::DebugMovementRenderer::previous_front_wheels_angle_slider_value = DEFAULT_FRONT_WHEEL_ANGLE [private]
```

12.13.3.12 previous_rear_left_wheel_speed_slider_value

```
int car::display::console::component::debug::DebugMovementRenderer::previous_rear_left_wheel_speed_slider_value = DEFAULT_REAR_WHEEL_SPEED [private]
```

12.13.3.13 previous_rear_right_wheel_speed_slider_value

```
int car::display::console::component::debug::DebugMovementRenderer::previous_rear_right_↵  
wheel_speed_slider_value = DEFAULT_REAR_WHEEL_SPEED [private]
```

12.13.3.14 previous_rear_wheels_speed_slider_value

```
int car::display::console::component::debug::DebugMovementRenderer::previous_rear_wheels_↵  
speed_slider_value = DEFAULT_REAR_WHEEL_SPEED [private]
```

12.13.3.15 rear_left_wheel_speed_slider

```
Component car::display::console::component::debug::DebugMovementRenderer::rear_left_wheel_↵  
speed_slider [private]
```

12.13.3.16 rear_left_wheel_speed_slider_value

```
int car::display::console::component::debug::DebugMovementRenderer::rear_left_wheel_speed_↵  
slider_value = DEFAULT_REAR_WHEEL_SPEED [private]
```

12.13.3.17 rear_right_wheel_speed_slider

```
Component car::display::console::component::debug::DebugMovementRenderer::rear_right_wheel_↵  
speed_slider [private]
```

12.13.3.18 rear_right_wheel_speed_slider_value

```
int car::display::console::component::debug::DebugMovementRenderer::rear_right_wheel_speed_↵  
slider_value = DEFAULT_REAR_WHEEL_SPEED [private]
```

12.13.3.19 rear_wheel_direction

```
bool car::display::console::component::debug::DebugMovementRenderer::rear_wheel_direction =  
true [private]
```

12.13.3.20 REAR_WHEEL_DIRECTION_BACKWARD_MESSAGE

```
constexpr auto car::display::console::component::debug::DebugMovementRenderer::REAR_WHEEL_↵  
DIRECTION_BACKWARD_MESSAGE = "Rear Wheel Direction: Backward" [static], [constexpr], [private]
```

12.13.3.21 rear_wheel_direction_checkbox_component

```
Component car::display::console::component::debug::DebugMovementRenderer::rear_wheel_direction_↵  
_checkbox_component [private]
```

12.13.3.22 rear_wheel_direction_debounce

```
bool car::display::console::component::debug::DebugMovementRenderer::rear_wheel_direction_↵  
debounce = false [private]
```

12.13.3.23 REAR_WHEEL_DIRECTION_FORWARD_MESSAGE

```
constexpr auto car::display::console::component::debug::DebugMovementRenderer::REAR_WHEEL_↵  
DIRECTION_FORWARD_MESSAGE = "Rear Wheel Direction: Forward" [static], [constexpr], [private]
```

12.13.3.24 rear_wheel_direction_signal

```
nod::signal<void(bool)> car::display::console::component::debug::DebugMovementRenderer::rear_↵  
_wheel_direction_signal [private]
```

12.13.3.25 rear_wheel_direction_status

```
std::string car::display::console::component::debug::DebugMovementRenderer::rear_wheel_↵  
direction_status = REAR_WHEEL_DIRECTION_FORWARD_MESSAGE [private]
```

12.13.3.26 rear_wheel_menu_entry

```
Component car::display::console::component::debug::DebugMovementRenderer::rear_wheel_menu_↵  
entry [private]
```

12.13.3.27 rear_wheel_speed_slider

```
Component car::display::console::component::debug::DebugMovementRenderer::rear_wheel_speed_slider [private]
```

12.13.3.28 rear_wheels_speed_slider_value

```
int car::display::console::component::debug::DebugMovementRenderer::rear_wheels_speed_slider_value = DEFAULT_REAR_WHEEL_SPEED [private]
```

12.13.3.29 servo_menu_entry

```
Component car::display::console::component::debug::DebugMovementRenderer::servo_menu_entry [private]
```

12.13.3.30 slider_container

```
Component car::display::console::component::debug::DebugMovementRenderer::slider_container [private]
```

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

- tui/src/car/display/console/component/debug/[DebugMovementRenderer.cxx](#)

12.14 car::system::device::DeviceManager Class Reference

```
#include <DeviceManager.h>
```

Public Member Functions

- [DeviceManager](#) (std::unique_ptr< [CameraDevice](#) > camera_device, std::unique_ptr< [lidar::LidarDevice](#) > lidar_device)
- [CameraDevice](#) * [getCameraDevice](#) ()
- [lidar::LidarDevice](#) * [getLidarDevice](#) ()
- const bool [isRunning](#) () const
- void [initialize](#) (std::shared_ptr< [system::CarSystem](#) > car_system)
- void [start](#) ()
- void [update](#) ()
- void [stop](#) ()
- void [terminate](#) ()

Static Public Member Functions

- static tl::expected< std::unique_ptr< [DeviceManager](#) >, std::string > [create](#) (std::shared_ptr< [Configuration](#) > configuration)

Private Attributes

- std::shared_ptr< [car::system::CarSystem](#) > [car_system](#)
- bool [is_initialized_](#) = false
- bool [is_running_](#) = false
- std::unique_ptr< [lidar::LidarDevice](#) > [lidar_device_](#)
- std::unique_ptr< [CameraDevice](#) > [camera_device_](#)

12.14.1 Constructor & Destructor Documentation

12.14.1.1 DeviceManager()

```
car::system::device::DeviceManager::DeviceManager (
    std::unique_ptr< CameraDevice > camera\_device,
    std::unique_ptr< lidar::LidarDevice > lidar\_device ) [inline]
```

12.14.2 Member Function Documentation

12.14.2.1 create()

```
tl::expected< std::unique_ptr< DeviceManager >, std::string > car::system::device::DeviceManager::create (
    std::shared_ptr< Configuration > configuration ) [static]
```

12.14.2.2 getCameraDevice()

```
CameraDevice * car::system::device::DeviceManager::getCameraDevice ( ) [inline]
```

12.14.2.3 getLidarDevice()

```
lidar::LidarDevice * car::system::device::DeviceManager::getLidarDevice ( ) [inline]
```

12.14.2.4 initialize()

```
void car::system::device::DeviceManager::initialize (
    std::shared_ptr< system::CarSystem > car_system )
```

12.14.2.5 isRunning()

```
const bool car::system::device::DeviceManager::isRunning ( ) const [inline]
```

12.14.2.6 start()

```
void car::system::device::DeviceManager::start ( )
```

12.14.2.7 stop()

```
void car::system::device::DeviceManager::stop ( )
```

12.14.2.8 terminate()

```
void car::system::device::DeviceManager::terminate ( )
```

12.14.2.9 update()

```
void car::system::device::DeviceManager::update ( )
```

12.14.3 Member Data Documentation

12.14.3.1 camera_device_

```
std::unique_ptr<CameraDevice> car::system::device::DeviceManager::camera_device_ [private]
```

12.14.3.2 car_system

```
std::shared_ptr<car::system::CarSystem> car::system::device::DeviceManager::car_system [private]
```

12.14.3.3 is_initialized_

```
bool car::system::device::DeviceManager::is_initialized_ = false [private]
```

12.14.3.4 is_running_

```
bool car::system::device::DeviceManager::is_running_ = false [private]
```

12.14.3.5 lidar_device_

```
std::unique_ptr<lidar::LidarDevice> car::system::device::DeviceManager::lidar_device_ [private]
```

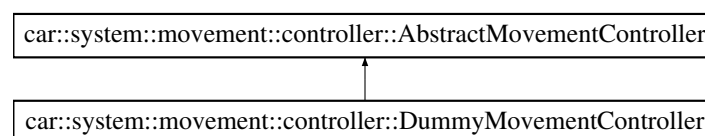
The documentation for this class was generated from the following files:

- common/include/car/system/device/[DeviceManager.h](#)
- common/src/car/system/device/[DeviceManager.cpp](#)

12.15 car::system::movement::controller::DummyMovementController Class Reference

```
#include <DummyMovementController.h>
```

Inheritance diagram for car::system::movement::controller::DummyMovementController:



Public Member Functions

- void [initialize](#) () final override
- void [stop](#) () final override
- void [terminate](#) () final override
- void [setRearWheelsSpeed](#) (const int speed) final override
- void [setRearLeftWheelSpeed](#) (const int speed) final override
- void [setRearRightWheelSpeed](#) (const int speed) final override
- void [setFrontWheelsAngle](#) (const float angle) final override
- void [setCameraServo1Angle](#) (const float angle) final override
- void [setCameraServo2Angle](#) (const float angle) final override
- void [setRearWheelsDirectionToForward](#) () final override
- void [setRearLeftWheelDirectionToForward](#) () final override
- void [setRearRightWheelDirectionToForward](#) () final override
- void [setRearWheelsDirectionToBackward](#) () final override
- void [setRearLeftWheelDirectionToBackward](#) () final override
- void [setRearRightWheelDirectionToBackward](#) () final override

12.15.1 Member Function Documentation

12.15.1.1 [initialize\(\)](#)

```
void car::system::movement::controller::DummyMovementController::initialize ( ) [inline],  
[final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.2 [setCameraServo1Angle\(\)](#)

```
void car::system::movement::controller::DummyMovementController::setCameraServo1Angle (  
    const float angle ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.3 [setCameraServo2Angle\(\)](#)

```
void car::system::movement::controller::DummyMovementController::setCameraServo2Angle (  
    const float angle ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.4 setFrontWheelsAngle()

```
void car::system::movement::controller::DummyMovementController::setFrontWheelsAngle (
    const float angle ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.5 setRearLeftWheelDirectionToBackward()

```
void car::system::movement::controller::DummyMovementController::setRearLeftWheelDirectionTo←
Backward ( ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.6 setRearLeftWheelDirectionToForward()

```
void car::system::movement::controller::DummyMovementController::setRearLeftWheelDirectionTo←
Forward ( ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.7 setRearLeftWheelSpeed()

```
void car::system::movement::controller::DummyMovementController::setRearLeftWheelSpeed (
    const int speed ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.8 setRearRightWheelDirectionToBackward()

```
void car::system::movement::controller::DummyMovementController::setRearRightWheelDirection←
ToBackward ( ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.9 setRearRightWheelDirectionToForward()

```
void car::system::movement::controller::DummyMovementController::setRearRightWheelDirection←
ToForward ( ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.10 setRearRightWheelSpeed()

```
void car::system::movement::controller::DummyMovementController::setRearRightWheelSpeed (
    const int speed ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.11 setRearWheelsDirectionToBackward()

```
void car::system::movement::controller::DummyMovementController::setRearWheelsDirectionTo↵
Backward ( ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.12 setRearWheelsDirectionToForward()

```
void car::system::movement::controller::DummyMovementController::setRearWheelsDirectionTo↵
Forward ( ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.13 setRearWheelsSpeed()

```
void car::system::movement::controller::DummyMovementController::setRearWheelsSpeed (
    const int speed ) [final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.14 stop()

```
void car::system::movement::controller::DummyMovementController::stop ( ) [final], [override],
[virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

12.15.1.15 terminate()

```
void car::system::movement::controller::DummyMovementController::terminate ( ) [inline],  
[final], [override], [virtual]
```

Implements [car::system::movement::controller::AbstractMovementController](#).

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

- common/include/car/system/movement/controller/[DummyMovementController.h](#)
- common/src/car/system/movement/controller/[DummyMovementController.cpp](#)

12.16 car::system::messaging::MessagingSystem::FirstMessageStruct Struct Reference

```
#include <MessagingSystem.h>
```

Public Attributes

- std::string [error_message](#)
- std::string [uuid](#)
- std::condition_variable [condition](#)

12.16.1 Member Data Documentation

12.16.1.1 condition

```
std::condition_variable car::system::messaging::MessagingSystem::FirstMessageStruct::condition
```

12.16.1.2 error_message

```
std::string car::system::messaging::MessagingSystem::FirstMessageStruct::error_message
```

12.16.1.3 uuid

```
std::string car::system::messaging::MessagingSystem::FirstMessageStruct::uuid
```

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

- common/include/car/system/messaging/[MessagingSystem.h](#)

12.17 car::configuration::JsonConfiguration Class Reference

Public Member Functions

- [JsonConfiguration](#) (std::string [exe_dir](#))
- void [setConfigFilePath](#) (std::string [config_file_path](#))
- const std::string & [getConfigFilePath](#) () const
- tl::expected< [Configuration](#), std::string > [loadConfiguration](#) ()

Private Attributes

- const std::string [exe_dir](#)
- std::string [config_file_path](#)

12.17.1 Constructor & Destructor Documentation

12.17.1.1 JsonConfiguration()

```
car::configuration::JsonConfiguration::JsonConfiguration (  
    std::string exe_dir ) [inline]
```

12.17.2 Member Function Documentation

12.17.2.1 getConfigFilePath()

```
const std::string & car::configuration::JsonConfiguration::getConfigFilePath ( ) const [inline]
```

12.17.2.2 loadConfiguration()

```
tl::expected< Configuration, std::string > car::configuration::JsonConfiguration::loadConfiguration  
( ) [inline]
```

12.17.2.3 setConfigFilePath()

```
void car::configuration::JsonConfiguration::setConfigFilePath (  
    std::string config_file_path ) [inline]
```

12.17.3 Member Data Documentation

12.17.3.1 config_file_path

```
std::string car::configuration::JsonConfiguration::config_file_path [private]
```

12.17.3.2 exe_dir

```
const std::string car::configuration::JsonConfiguration::exe_dir [private]
```

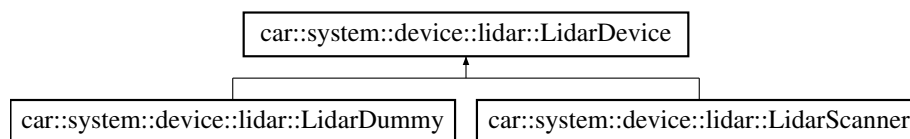
The documentation for this class was generated from the following file:

- tui/src/car/configuration/[JsonConfiguration.cxx](#)

12.18 car::system::device::lidar::LidarDevice Class Reference

```
#include <LidarDevice.h>
```

Inheritance diagram for car::system::device::lidar::LidarDevice:



Public Member Functions

- std::vector< Measure > [getScanData](#) () const
- virtual void [start](#) ()=0
- virtual void [update](#) ()=0
- virtual void [stop](#) ()=0
- virtual void [initialize](#) ()=0
- virtual void [terminate](#) ()=0
- virtual void [disconnect](#) ()=0

Protected Member Functions

- void [setScanData](#) (const std::vector< Measure > &scan_data)

Protected Attributes

- std::vector< Measure > [scan_data_](#)

Friends

- class [DeviceManager](#)

12.18.1 Member Function Documentation

12.18.1.1 disconnect()

```
virtual void car::system::device::lidar::LidarDevice::disconnect ( ) [pure virtual]
```

Implemented in [car::system::device::lidar::LidarDummy](#), and [car::system::device::lidar::LidarScanner](#).

12.18.1.2 getScanData()

```
std::vector< Measure > car::system::device::lidar::LidarDevice::getScanData ( ) const [inline]
```

12.18.1.3 initialize()

```
virtual void car::system::device::lidar::LidarDevice::initialize ( ) [pure virtual]
```

Implemented in [car::system::device::lidar::LidarDummy](#), and [car::system::device::lidar::LidarScanner](#).

12.18.1.4 setScanData()

```
void car::system::device::lidar::LidarDevice::setScanData (
    const std::vector< Measure > & scan_data ) [inline], [protected]
```

12.18.1.5 start()

```
virtual void car::system::device::lidar::LidarDevice::start ( ) [pure virtual]
```

Implemented in [car::system::device::lidar::LidarDummy](#), and [car::system::device::lidar::LidarScanner](#).

12.18.1.6 stop()

```
virtual void car::system::device::lidar::LidarDevice::stop ( ) [pure virtual]
```

Implemented in [car::system::device::lidar::LidarDummy](#), and [car::system::device::lidar::LidarScanner](#).

12.18.1.7 terminate()

```
virtual void car::system::device::lidar::LidarDevice::terminate ( ) [pure virtual]
```

Implemented in [car::system::device::lidar::LidarDummy](#), and [car::system::device::lidar::LidarScanner](#).

12.18.1.8 update()

```
virtual void car::system::device::lidar::LidarDevice::update ( ) [pure virtual]
```

Implemented in [car::system::device::lidar::LidarDummy](#), and [car::system::device::lidar::LidarScanner](#).

12.18.2 Friends And Related Function Documentation

12.18.2.1 DeviceManager

```
friend class DeviceManager [friend]
```

12.18.3 Member Data Documentation

12.18.3.1 scan_data_

```
std::vector<Measure> car::system::device::lidar::LidarDevice::scan_data_ [protected]
```

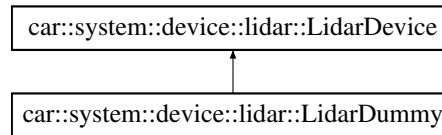
The documentation for this class was generated from the following file:

- [common/include/car/system/device/lidar/LidarDevice.h](#)

12.19 car::system::device::lidar::LidarDummy Class Reference

```
#include <LidarDummy.h>
```

Inheritance diagram for car::system::device::lidar::LidarDummy:



Public Member Functions

- [LidarDummy](#) ()
- void [start](#) () final override
- void [update](#) () final override
- void [stop](#) () final override
- void [initialize](#) () final override
- void [terminate](#) () final override
- void [disconnect](#) () final override

Additional Inherited Members

12.19.1 Constructor & Destructor Documentation

12.19.1.1 LidarDummy()

```
car::system::device::lidar::LidarDummy::LidarDummy ( ) [inline]
```

12.19.2 Member Function Documentation

12.19.2.1 disconnect()

```
void car::system::device::lidar::LidarDummy::disconnect ( ) [inline], [final], [override],  
[virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.19.2.2 initialize()

```
void car::system::device::lidar::LidarDummy::initialize ( ) [inline], [final], [override],  
[virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.19.2.3 start()

```
void car::system::device::lidar::LidarDummy::start ( ) [inline], [final], [override], [virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.19.2.4 stop()

```
void car::system::device::lidar::LidarDummy::stop ( ) [inline], [final], [override], [virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.19.2.5 terminate()

```
void car::system::device::lidar::LidarDummy::terminate ( ) [inline], [final], [override],  
[virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.19.2.6 update()

```
void car::system::device::lidar::LidarDummy::update ( ) [inline], [final], [override], [virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

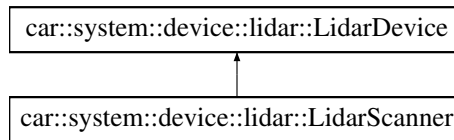
The documentation for this class was generated from the following file:

- [common/include/car/system/device/lidar/LidarDummy.h](#)

12.20 car::system::device::lidar::LidarScanner Class Reference

```
#include <LidarScanner.h>
```

Inheritance diagram for car::system::device::lidar::LidarScanner:



Public Member Functions

- [LidarScanner](#) (std::shared_ptr< [configuration::Configuration](#) > configuration, std::unique_ptr< RPLidar > lidar)
- void [start](#) () final override
- void [update](#) () final override
- void [stop](#) () final override
- void [initialize](#) () final override
- void [disconnect](#) () final override
- void [terminate](#) () final override

Static Public Member Functions

- static tl::expected< std::unique_ptr< [LidarScanner](#) >, std::string > [create](#) (std::shared_ptr< [configuration::Configuration](#) > configuration) noexcept

Private Attributes

- std::atomic_bool [running](#) = false
- std::shared_ptr< [configuration::Configuration](#) > [configuration_](#)
- std::vector< Measure > [scan_data_](#)
- std::unique_ptr< RPLidar > [lidar_](#)
- std::variant< std::function< std::vector< Measure >()>, nullptr_t > [scan_generator_](#) = nullptr
- std::mutex [scan_data_mutex_](#)

Additional Inherited Members

12.20.1 Constructor & Destructor Documentation

12.20.1.1 LidarScanner()

```

car::system::device::lidar::LidarScanner::LidarScanner (
    std::shared_ptr< configuration::Configuration > configuration,
    std::unique_ptr< RPLidar > lidar ) [inline]
  
```

12.20.2 Member Function Documentation

12.20.2.1 create()

```
static tl::expected< std::unique_ptr< LidarScanner >, std::string > car::system::device↵  
::lidar::LidarScanner::create (   
    std::shared_ptr< configuration::Configuration > configuration ) [inline], [static],  
[noexcept]
```

12.20.2.2 disconnect()

```
void car::system::device::lidar::LidarScanner::disconnect ( ) [inline], [final], [override],  
[virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.20.2.3 initialize()

```
void car::system::device::lidar::LidarScanner::initialize ( ) [inline], [final], [override],  
[virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.20.2.4 start()

```
void car::system::device::lidar::LidarScanner::start ( ) [inline], [final], [override], [virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.20.2.5 stop()

```
void car::system::device::lidar::LidarScanner::stop ( ) [inline], [final], [override], [virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.20.2.6 terminate()

```
void car::system::device::lidar::LidarScanner::terminate ( ) [inline], [final], [override],  
[virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.20.2.7 update()

```
void car::system::device::lidar::LidarScanner::update ( ) [inline], [final], [override],  
[virtual]
```

Implements [car::system::device::lidar::LidarDevice](#).

12.20.3 Member Data Documentation

12.20.3.1 configuration_

```
std::shared_ptr<configuration::Configuration> car::system::device::lidar::LidarScanner::configuration←  
_ [private]
```

12.20.3.2 lidar_

```
std::unique_ptr<RPLidar> car::system::device::lidar::LidarScanner::lidar_ [private]
```

12.20.3.3 running

```
std::atomic_bool car::system::device::lidar::LidarScanner::running = false [private]
```

12.20.3.4 scan_data_

```
std::vector<Measure> car::system::device::lidar::LidarScanner::scan_data_ [private]
```

12.20.3.5 scan_data_mutex_

```
std::mutex car::system::device::lidar::LidarScanner::scan_data_mutex_ [private]
```

12.20.3.6 scan_generator_

```
std::variant<std::function<std::vector<Measure>()>, nullptr_t> car::system::device::lidar↔  
::LidarScanner::scan_generator_ = nullptr [private]
```

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

- common/include/car/system/device/lidar/[LidarScanner.h](#)

12.21 car::display::console::screen::LoggingScreen Class Reference

Public Member Functions

- [LoggingScreen](#) (std::shared_ptr< [logging::vector_sink_mt](#) > [vector_sink](#))
- Component [element](#) ()

Private Attributes

- int [selected_line](#) = 0
- std::shared_ptr< [logging::vector_sink_mt](#) > [vector_sink](#)
- Component [menu](#)
- Component [my_custom_menu](#)
- ftxui::Elements [line_elements](#)

12.21.1 Constructor & Destructor Documentation

12.21.1.1 LoggingScreen()

```
car::display::console::screen::LoggingScreen::LoggingScreen (
    std::shared_ptr< logging::vector\_sink\_mt > vector\_sink ) [inline]
```

12.21.2 Member Function Documentation

12.21.2.1 element()

```
Component car::display::console::screen::LoggingScreen::element ( ) [inline]
```

12.21.3 Member Data Documentation

12.21.3.1 line_elements

```
ftxui::Elements car::display::console::screen::LoggingScreen::line_elements [private]
```

12.21.3.2 menu

```
Component car::display::console::screen::LoggingScreen::menu [private]
```

12.21.3.3 my_custom_menu

```
Component car::display::console::screen::LoggingScreen::my_custom_menu [private]
```

12.21.3.4 selected_line

```
int car::display::console::screen::LoggingScreen::selected_line = 0 [private]
```

12.21.3.5 vector_sink

```
std::shared_ptr<logging::vector_sink_mt> car::display::console::screen::LoggingScreen::vector←  
_sink [private]
```

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

- [tui/src/car/display/console/screen/LoggingScreen.cxx](#)

12.22 car::display::console::component::main::MainErrorModal Class Reference

Public Member Functions

- [MainErrorModal](#) ()
- Component [element](#) ()
- void [setErrorMessage](#) (std::string message)

Public Attributes

- bool [error_modal_shown](#) = false

Private Attributes

- Component [main_error_modal](#)
- Element [error_element](#)

12.22.1 Constructor & Destructor Documentation

12.22.1.1 MainErrorModal()

```
car::display::console::component::main::MainErrorModal::MainErrorModal ( ) [inline]
```

12.22.2 Member Function Documentation

12.22.2.1 element()

```
Component car::display::console::component::main::MainErrorModal::element ( ) [inline]
```

12.22.2.2 setErrorMessage()

```
void car::display::console::component::main::MainErrorModal::setErrorMessage (
    std::string message ) [inline]
```

12.22.3 Member Data Documentation

12.22.3.1 error_element

Element car::display::console::component::main::MainErrorModal::error_element [private]

12.22.3.2 error_modal_shown

bool car::display::console::component::main::MainErrorModal::error_modal_shown = false

12.22.3.3 main_error_modal

Component car::display::console::component::main::MainErrorModal::main_error_modal [private]

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

- tui/src/car/display/console/component/main/[MainErrorModal.cxx](#)

12.23 car::display::console::component::main::MainExitModal Class Reference

Public Member Functions

- [MainExitModal](#) (std::function< void()> [exit](#))
- Component [element](#) ()

Public Attributes

- bool [exit_modal_shown](#) = false

Private Attributes

- std::function< void()> [exit](#)
- Component [main_exit_modal](#)

12.23.1 Constructor & Destructor Documentation

12.23.1.1 MainExitModal()

```
car::display::console::component::main::MainExitModal::MainExitModal (
    std::function< void()> exit ) [inline]
```


12.23.2 Member Function Documentation

12.23.2.1 element()

Component car::display::console::component::main::MainExitModal::element () [inline]

12.23.3 Member Data Documentation

12.23.3.1 exit

std::function<void()> car::display::console::component::main::MainExitModal::exit [private]

12.23.3.2 exit_modal_shown

bool car::display::console::component::main::MainExitModal::exit_modal_shown = false

12.23.3.3 main_exit_modal

Component car::display::console::component::main::MainExitModal::main_exit_modal [private]

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

- tui/src/car/display/console/component/main/[MainExitModal.cxx](#)

12.24 car::display::console::screen::MainScreen Class Reference

Public Member Functions

- [MainScreen](#) (std::shared_ptr< [CarSystem](#) > car_system, std::function< void()> exit)
- Component [element](#) ()

Private Attributes

- `std::shared_ptr< CarSystem > car_system`
- `Box box`
- `ConnectButton connect_button`
- `MainExitModal main_exit_modal`
- `MainErrorModal main_error_modal`
- `Component info`
- `Component main_screen`
- `Component main_component`

12.24.1 Constructor & Destructor Documentation

12.24.1.1 MainScreen()

```
car::display::console::screen::MainScreen::MainScreen (  
    std::shared_ptr< CarSystem > car_system,  
    std::function< void()> exit ) [inline]
```

12.24.2 Member Function Documentation

12.24.2.1 element()

```
Component car::display::console::screen::MainScreen::element ( ) [inline]
```

12.24.3 Member Data Documentation

12.24.3.1 box

```
Box car::display::console::screen::MainScreen::box [private]
```

12.24.3.2 car_system

```
std::shared_ptr<CarSystem> car::display::console::screen::MainScreen::car_system [private]
```

12.24.3.3 connect_button

`ConnectButton` car::display::console::screen::MainScreen::connect_button [private]

12.24.3.4 info

`Component` car::display::console::screen::MainScreen::info [private]

12.24.3.5 main_component

`Component` car::display::console::screen::MainScreen::main_component [private]

12.24.3.6 main_error_modal

`MainErrorModal` car::display::console::screen::MainScreen::main_error_modal [private]

12.24.3.7 main_exit_modal

`MainExitModal` car::display::console::screen::MainScreen::main_exit_modal [private]

12.24.3.8 main_screen

`Component` car::display::console::screen::MainScreen::main_screen [private]

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

- tui/src/car/display/console/screen/[MainScreen.cxx](#)

12.25 car::system::messaging::MessagingSystem Class Reference

```
#include <MessagingSystem.h>
```

Classes

- struct [FirstMessageStruct](#)

Public Member Functions

- [MessagingSystem](#) ()
- void [initialize](#) (std::shared_ptr< [configuration::Configuration](#) > configuration)
Initializes the use of Websockets and initializes the Signals.
- void [initializeWebSocket](#) ()
Creates a new WebSocket object for use.
- const tl::expected< nullptr_t, std::string > [tryConnect](#) ()
Attempts to connect to the WebSocket server and retrieves the first message from the WebSocket (Should be UUID)
- void [stop](#) ()
- void [terminate](#) ()
- void [setConfiguration](#) (std::shared_ptr< [configuration::Configuration](#) > configuration)
- nod::signal< void(const std::string, const rapidjson::Document &)> & [getCommandSignal](#) ()
- nod::signal< void(const std::string, const rapidjson::Document &)> & [getSelectionSignal](#) ()
- nod::signal< void(const std::string)> & [getMessageSignal](#) ()
- nod::signal< void(const std::string)> & [getDisconnectSignal](#) ()
- void [onMessageCallback](#) (const ix::WebSocketMessagePtr &msg) const
- void [onDisconnect](#) (const std::string)
- const std::string [getUUID](#) () const
- void [handleMessage](#) (const std::string &message) const
Sends out signals depending on the type of message.
- void [sendMessage](#) (const std::string &message)
- void [onFirstMessage](#) (const ix::WebSocketMessagePtr &msg, [FirstMessageStruct](#) &first_message_struct)
Actually retrieves the First Message from the WebSocket to put into [FirstMessageStruct](#).
- const bool [isConnected](#) () const

Public Attributes

- nod::signal< void(std::string)> [on_disconnect_signal_](#)
- nod::signal< void(const std::string)> [message_signal_](#)
- nod::signal< void(const std::string, const rapidjson::Document &)> [command_signal_](#)
- nod::signal< void(const std::string, const rapidjson::Document &)> [selection_signal_](#)

Private Member Functions

- tl::expected< std::string, std::string > [getFirstMessage](#) ()
Waits and retrieves the first message when connecting to a websocket.

Private Attributes

- std::shared_ptr< [configuration::Configuration](#) > [configuration_](#)
- std::unique_ptr< ix::WebSocket > [websocket_](#)
- std::string [websocket_url_](#)
- std::string [uuid_](#)
- bool [connected_](#) = false

12.25.1 Constructor & Destructor Documentation

12.25.1.1 MessagingSystem()

```
car::system::messaging::MessagingSystem::MessagingSystem ( )
```

12.25.2 Member Function Documentation

12.25.2.1 getCommandSignal()

```
nod::signal< void(const std::string, const rapidjson::Document &)> & car::system::messaging↵  
::MessagingSystem::getCommandSignal ( ) [inline]
```

12.25.2.2 getDisconnectSignal()

```
nod::signal< void(const std::string)> & car::system::messaging::MessagingSystem::getDisconnect↵  
Signal ( ) [inline]
```

12.25.2.3 getFirstMessage()

```
tl::expected< std::string, std::string > car::system::messaging::MessagingSystem::getFirst↵  
Message ( ) [private]
```

Waits and retrieves the first message when connecting to a websocket.

Returns

tl::expected<std::string, std::string>

12.25.2.4 getMessageSignal()

```
nod::signal< void(const std::string)> & car::system::messaging::MessagingSystem::getMessage↵  
Signal ( ) [inline]
```

12.25.2.5 getSelectionSignal()

```
nod::signal< void(const std::string, const rapidjson::Document &)> & car::system::messaging↵  
::MessagingSystem::getSelectionSignal ( ) [inline]
```

12.25.2.6 getUUID()

```
const std::string car::system::messaging::MessagingSystem::getUUID ( ) const [inline]
```

12.25.2.7 handleMessage()

```
void car::system::messaging::MessagingSystem::handleMessage (
    const std::string & message ) const
```

Sends out signals depending on the type of message.

Parameters

<i>message</i>	
----------------	--

12.25.2.8 initialize()

```
void car::system::messaging::MessagingSystem::initialize (
    std::shared_ptr< configuration::Configuration > configuration )
```

Initializes the use of Websockets and initializes the Signals.

Parameters

<i>configuration</i>	
----------------------	--

12.25.2.9 initializeWebSocket()

```
void car::system::messaging::MessagingSystem::initializeWebSocket ( )
```

Creates a new WebSocket object for use.

12.25.2.10 isConnected()

```
const bool car::system::messaging::MessagingSystem::isConnected ( ) const [inline]
```

12.25.2.11 onDisconnect()

```
void car::system::messaging::MessagingSystem::onDisconnect (
    const std::string message )
```

12.25.2.12 onFirstMessage()

```
void car::system::messaging::MessagingSystem::onFirstMessage (
    const ix::WebSocketMessagePtr & msg,
    FirstMessageStruct & first_message_struct )
```

Actually retrieves the First Message from the Websocket to put into [FirstMessageStruct](#).

Parameters

<i>msg</i>	
<i>first_message_struct</i>	

12.25.2.13 onMessageCallback()

```
void car::system::messaging::MessagingSystem::onMessageCallback (
    const ix::WebSocketMessagePtr & msg ) const
```

12.25.2.14 sendMessage()

```
void car::system::messaging::MessagingSystem::sendMessage (
    const std::string & message )
```

12.25.2.15 setConfiguration()

```
void car::system::messaging::MessagingSystem::setConfiguration (
    std::shared_ptr< configuration::Configuration > configuration )
```

12.25.2.16 stop()

```
void car::system::messaging::MessagingSystem::stop ( )
```

12.25.2.17 terminate()

```
void car::system::messaging::MessagingSystem::terminate ( )
```

12.25.2.18 tryConnect()

```
const tl::expected< nullptr_t, std::string > car::system::messaging::MessagingSystem::try↵
Connect ( )
```

Attempts to connect to the Websocket server and retrieves the first message from the Websocket (Should be UUID)

Returns

```
const tl::expected<nullptr_t, std::string>
```

12.25.3 Member Data Documentation**12.25.3.1 command_signal_**

```
nod::signal<void(const std::string, const rapidjson::Document&)> car::system::messaging::↵
MessagingSystem::command_signal_
```

12.25.3.2 configuration_

```
std::shared_ptr<configuration::Configuration> car::system::messaging::MessagingSystem::configuration↵
_ [private]
```

12.25.3.3 connected_

```
bool car::system::messaging::MessagingSystem::connected_ = false [private]
```

12.25.3.4 message_signal_

```
nod::signal<void(const std::string)> car::system::messaging::MessagingSystem::message_signal_↵
—
```


12.25.3.5 on_disconnect_signal_

```
nod::signal<void(std::string)> car::system::messaging::MessagingSystem::on_disconnect_signal_↵  
_
```

12.25.3.6 selection_signal_

```
nod::signal<void(const std::string, const rapidjson::Document&)> car::system::messaging::↵  
MessagingSystem::selection_signal_
```

12.25.3.7 uuid_

```
std::string car::system::messaging::MessagingSystem::uuid_ [private]
```

12.25.3.8 websocket_

```
std::unique_ptr<ix::WebSocket> car::system::messaging::MessagingSystem::websocket_ [private]
```

12.25.3.9 websocket_url_

```
std::string car::system::messaging::MessagingSystem::websocket_url_ [private]
```

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

- common/include/car/system/messaging/[MessagingSystem.h](#)
- common/src/car/system/messaging/[MessagingSystem.cpp](#)

12.26 car::system::movement::MovementSystem Class Reference

```
#include <MovementSystem.h>
```

Public Member Functions

- [MovementSystem](#) (std::unique_ptr< [AbstractMovementController](#) > [movement_controller](#))
- void [initialize](#) ()
- void [start](#) ()
- void [stop](#) ()
- void [terminate](#) ()
- void [setRearWheelsSpeed](#) (const int speed) const
- void [setRearLeftWheelSpeed](#) (const int speed) const
- void [setRearRightWheelSpeed](#) (const int speed) const
- void [setFrontWheelsAngle](#) (const float angle) const
- void [setCameraServo1Angle](#) (const float angle) const
- void [setCameraServo2Angle](#) (const float angle) const
- void [setRearWheelsDirectionToForward](#) () const
- void [setRearLeftWheelDirectionToForward](#) () const
- void [setRearRightWheelDirectionToForward](#) () const
- void [setRearWheelsDirectionToBackward](#) () const
- void [setRearLeftWheelDirectionToBackward](#) () const
- void [setRearRightWheelDirectionToBackward](#) () const
- [~MovementSystem](#) ()

Private Attributes

- std::unique_ptr< [AbstractMovementController](#) > [movement_controller](#)

12.26.1 Constructor & Destructor Documentation

12.26.1.1 MovementSystem()

```
car::system::movement::MovementSystem::MovementSystem (
    std::unique_ptr< AbstractMovementController > movement\_controller ) [inline]
```

12.26.1.2 ~MovementSystem()

```
car::system::movement::MovementSystem::~MovementSystem ( ) [inline]
```

12.26.2 Member Function Documentation

12.26.2.1 initialize()

```
void car::system::movement::MovementSystem::initialize ( ) [inline]
```

12.26.2.2 setCameraServo1Angle()

```
void car::system::movement::MovementSystem::setCameraServo1Angle (
    const float angle ) const [inline]
```

12.26.2.3 setCameraServo2Angle()

```
void car::system::movement::MovementSystem::setCameraServo2Angle (
    const float angle ) const [inline]
```

12.26.2.4 setFrontWheelsAngle()

```
void car::system::movement::MovementSystem::setFrontWheelsAngle (
    const float angle ) const [inline]
```

12.26.2.5 setRearLeftWheelDirectionToBackward()

```
void car::system::movement::MovementSystem::setRearLeftWheelDirectionToBackward ( ) const
[inline]
```

12.26.2.6 setRearLeftWheelDirectionToForward()

```
void car::system::movement::MovementSystem::setRearLeftWheelDirectionToForward ( ) const [inline]
```

12.26.2.7 setRearLeftWheelSpeed()

```
void car::system::movement::MovementSystem::setRearLeftWheelSpeed (
    const int speed ) const [inline]
```

12.26.2.8 setRearRightWheelDirectionToBackward()

```
void car::system::movement::MovementSystem::setRearRightWheelDirectionToBackward ( ) const
[inline]
```

12.26.2.9 setRearRightWheelDirectionToForward()

```
void car::system::movement::MovementSystem::setRearRightWheelDirectionToForward ( ) const  
[inline]
```

12.26.2.10 setRearRightWheelSpeed()

```
void car::system::movement::MovementSystem::setRearRightWheelSpeed (   
    const int speed ) const [inline]
```

12.26.2.11 setRearWheelsDirectionToBackward()

```
void car::system::movement::MovementSystem::setRearWheelsDirectionToBackward ( ) const [inline]
```

12.26.2.12 setRearWheelsDirectionToForward()

```
void car::system::movement::MovementSystem::setRearWheelsDirectionToForward ( ) const [inline]
```

12.26.2.13 setRearWheelsSpeed()

```
void car::system::movement::MovementSystem::setRearWheelsSpeed (   
    const int speed ) const [inline]
```

12.26.2.14 start()

```
void car::system::movement::MovementSystem::start ( ) [inline]
```

12.26.2.15 stop()

```
void car::system::movement::MovementSystem::stop ( ) [inline]
```

12.26.2.16 terminate()

```
void car::system::movement::MovementSystem::terminate ( ) [inline]
```

12.26.3 Member Data Documentation

12.26.3.1 movement_controller

```
std::unique_ptr<AbstractMovementController> car::system::movement::MovementSystem::movement_↔  
controller [private]
```

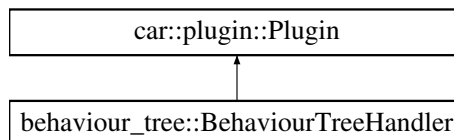
The documentation for this class was generated from the following file:

- common/include/car/system/movement/[MovementSystem.h](#)

12.27 car::plugin::Plugin Class Reference

```
#include <Plugin.h>
```

Inheritance diagram for car::plugin::Plugin:



Public Member Functions

- virtual void [initialize](#) (std::shared_ptr< [car::system::CarSystem](#) > car_system)=0
- virtual void [update](#) ()=0
- virtual void [stop](#) ()=0
- virtual std::string [getName](#) ()=0

12.27.1 Member Function Documentation

12.27.1.1 getName()

```
virtual std::string car::plugin::Plugin::getName ( ) [pure virtual]
```

Implemented in [behaviour_tree::BehaviourTreeHandler](#).

12.27.1.2 initialize()

```
virtual void car::plugin::Plugin::initialize (
    std::shared_ptr< car::system::CarSystem > car_system ) [pure virtual]
```

Implemented in [behaviour_tree::BehaviourTreeHandler](#).

12.27.1.3 stop()

```
virtual void car::plugin::Plugin::stop ( ) [pure virtual]
```

Implemented in [behaviour_tree::BehaviourTreeHandler](#).

12.27.1.4 update()

```
virtual void car::plugin::Plugin::update ( ) [pure virtual]
```

Implemented in [behaviour_tree::BehaviourTreeHandler](#).

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

- common/include/car/plugin/[Plugin.h](#)

12.28 car::plugin::PluginManager Class Reference

```
#include <PluginManager.h>
```

Public Member Functions

- void [initialize](#) (std::shared_ptr< [system::CarSystem](#) > car_system)
- void [update](#) ()
- void [stop](#) ()
- void [terminate](#) ()
- void [addPlugin](#) (std::shared_ptr< [Plugin](#) > plugin)
- template<typename T >
std::shared_ptr< T > [getPlugin](#) ()

Private Attributes

- std::vector< std::shared_ptr< [Plugin](#) > > [plugins](#)

12.28.1 Member Function Documentation

12.28.1.1 addPlugin()

```
void car::plugin::PluginManager::addPlugin (
    std::shared_ptr< Plugin > plugin ) [inline]
```

12.28.1.2 getPlugin()

```
template<typename T >
std::shared_ptr< T > car::plugin::PluginManager::getPlugin ( ) [inline]
```

12.28.1.3 initialize()

```
void car::plugin::PluginManager::initialize (
    std::shared_ptr< system::CarSystem > car_system ) [inline]
```

12.28.1.4 stop()

```
void car::plugin::PluginManager::stop ( ) [inline]
```

12.28.1.5 terminate()

```
void car::plugin::PluginManager::terminate ( ) [inline]
```

12.28.1.6 update()

```
void car::plugin::PluginManager::update ( ) [inline]
```

12.28.2 Member Data Documentation

12.28.2.1 plugins

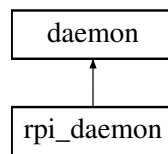
```
std::vector<std::shared_ptr<Plugin> > car::plugin::PluginManager::plugins [private]
```

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

- common/include/car/plugin/[PluginManager.h](#)

12.29 rpi_daemon Class Reference

Inheritance diagram for rpi_daemon:



Public Member Functions

- void [on_start](#) (const INIReader reader) override
- void [update](#) ()
- void [connect](#) (const std::chrono::time_point< std::chrono::steady_clock > &now)
- void [on_update](#) () override
- void [on_stop](#) () override
- void [on_reload](#) (const INIReader reader) override

Private Attributes

- std::shared_ptr< [CarSystem](#) > [car_system](#)
- bool [any_configuration_empty](#) = false
- bool [attempted_to_reconnect](#) = false
- std::chrono::milliseconds [connection_ms_interval](#) = std::chrono::milliseconds(1000)
- std::chrono::time_point< std::chrono::steady_clock > [last_connected](#)

12.29.1 Member Function Documentation

12.29.1.1 connect()

```
void rpi_daemon::connect (
    const std::chrono::time_point< std::chrono::steady_clock > & now ) [inline]
```


12.29.1.2 on_reload()

```
void rpi_daemon::on_reload (
    const INIReader reader ) [inline], [override]
```

12.29.1.3 on_start()

```
void rpi_daemon::on_start (
    const INIReader reader ) [inline], [override]
```

12.29.1.4 on_stop()

```
void rpi_daemon::on_stop ( ) [inline], [override]
```

12.29.1.5 on_update()

```
void rpi_daemon::on_update ( ) [inline], [override]
```

12.29.1.6 update()

```
void rpi_daemon::update ( ) [inline]
```

12.29.2 Member Data Documentation

12.29.2.1 any_configuration_empty

```
bool rpi_daemon::any_configuration_empty = false [private]
```

12.29.2.2 attempted_to_reconnect

```
bool rpi_daemon::attempted_to_reconnect = false [private]
```

12.29.2.3 car_system

```
std::shared_ptr<CarSystem> rpi_daemon::car_system [private]
```

12.29.2.4 connection_ms_interval

```
std::chrono::milliseconds rpi_daemon::connection_ms_interval = std::chrono::milliseconds(1000)
[private]
```

12.29.2.5 last_connected

```
std::chrono::time_point<std::chrono::steady_clock> rpi_daemon::last_connected [private]
```

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

- daemon/src/main.cpp

12.30 car::display::console::component::settings::SettingsEditConfig Class Reference

Public Member Functions

- [SettingsEditConfig](#) (std::shared_ptr< [system::CarSystem](#) > [car_system](#), std::shared_ptr< [JsonConfiguration](#) > [json_configuration](#))
- Component [element](#) ()

Private Attributes

- std::shared_ptr< [system::CarSystem](#) > [car_system](#)
- std::shared_ptr< [configuration::JsonConfiguration](#) > [json_configuration](#)
- std::string [placeholder](#) = "settings/config.jsonc"
- std::string [settings_file_path](#) = "settings/config.jsonc"
- Component [input_settings_file_path](#)
- Component [load_button](#)

12.30.1 Constructor & Destructor Documentation

12.30.1.1 SettingsEditConfig()

```
car::display::console::component::settings::SettingsEditConfig::SettingsEditConfig (
    std::shared_ptr< system::CarSystem > car_system,
    std::shared_ptr< JsonConfiguration > json_configuration ) [inline]
```

12.30.2 Member Function Documentation

12.30.2.1 element()

```
Component car::display::console::component::settings::SettingsEditConfig::element ( ) [inline]
```

12.30.3 Member Data Documentation

12.30.3.1 car_system

```
std::shared_ptr<system::CarSystem> car::display::console::component::settings::SettingsEdit↔
Config::car_system [private]
```

12.30.3.2 input_settings_file_path

```
Component car::display::console::component::settings::SettingsEditConfig::input_settings_↔
file_path [private]
```

12.30.3.3 json_configuration

```
std::shared_ptr<configuration::JsonConfiguration> car::display::console::component::settings↔
::SettingsEditConfig::json_configuration [private]
```

12.30.3.4 load_button

```
Component car::display::console::component::settings::SettingsEditConfig::load_button [private]
```

12.30.3.5 placeholder

```
std::string car::display::console::component::settings::SettingsEditConfig::placeholder =
"settings/config.jsonc" [private]
```

12.30.3.6 settings_file_path

```
std::string car::display::console::component::settings::SettingsEditConfig::settings_file_path
= "settings/config.jsonc" [private]
```

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

- tui/src/car/display/console/component/settings/[SettingsEditConfig.cxx](#)

12.31 car::display::console::screen::SettingsScreen Class Reference

Public Member Functions

- [SettingsScreen](#) (std::shared_ptr< [CarSystem](#) > car_system, std::shared_ptr< [JsonConfiguration](#) > json_configuration)
- Component [element](#) ()
- void [update](#) ()

Private Attributes

- std::shared_ptr< [CarSystem](#) > car_system
- [SettingsEditConfig](#) settings_edit_config
- [DebugEnabler](#) debug_enabler
- [DebugLidarCheckbox](#) debug_lidar_checkbox
- [DebugMovementRenderer](#) debug_movement_renderer
- [DebugMessagingTextbox](#) debug_messaging_text_box

12.31.1 Constructor & Destructor Documentation

12.31.1.1 SettingsScreen()

```
car::display::console::screen::SettingsScreen::SettingsScreen (
    std::shared_ptr< CarSystem > car_system,
    std::shared_ptr< JsonConfiguration > json_configuration ) [inline]
```

12.31.2 Member Function Documentation

12.31.2.1 element()

Component car::display::console::screen::SettingsScreen::element () [inline]

12.31.2.2 update()

void car::display::console::screen::SettingsScreen::update () [inline]

12.31.3 Member Data Documentation

12.31.3.1 car_system

std::shared_ptr<CarSystem> car::display::console::screen::SettingsScreen::car_system [private]

12.31.3.2 debug_enabler

DebugEnabledler car::display::console::screen::SettingsScreen::debug_enabler [private]

12.31.3.3 debug_lidar_checkbox

DebugLidarCheckbox car::display::console::screen::SettingsScreen::debug_lidar_checkbox [private]

12.31.3.4 debug_messaging_text_box

DebugMessagingTextbox car::display::console::screen::SettingsScreen::debug_messaging_text_box
[private]

12.31.3.5 debug_movement_renderer

```
DebugMovementRenderer car::display::console::screen::SettingsScreen::debug_movement_renderer  
[private]
```

12.31.3.6 settings_edit_config

```
SettingsEditConfig car::display::console::screen::SettingsScreen::settings_edit_config [private]
```

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

- [tui/src/car/display/console/screen/SettingsScreen.cxx](#)

12.32 TB6612 Class Reference

```
#include <TB6612.h>
```

Public Member Functions

- [TB6612](#) (int [motor_pin](#), int [pwm_pin](#))
- void [setPWM](#) (int value)
- void [forward](#) ()
- void [backward](#) ()
- void [stop](#) ()
- void [setOffset](#) (bool [offset](#))
- const int & [getMotorPin](#) () const
- const int & [getPWMPin](#) () const

Private Attributes

- const int [motor_pin](#)
- const int [pwm_pin](#)
- bool [offset](#) = true

12.32.1 Constructor & Destructor Documentation

12.32.1.1 TB6612()

```
TB6612::TB6612 (  
    int motor_pin,  
    int pwm_pin )
```

12.32.2 Member Function Documentation

12.32.2.1 backward()

```
void TB6612::backward ( )
```

12.32.2.2 forward()

```
void TB6612::forward ( )
```

12.32.2.3 getMotorPin()

```
const int & TB6612::getMotorPin ( ) const
```

12.32.2.4 getPWMPin()

```
const int & TB6612::getPWMPin ( ) const
```

12.32.2.5 setOffset()

```
void TB6612::setOffset (
    bool offset )
```

12.32.2.6 setPWM()

```
void TB6612::setPWM (
    int value )
```

12.32.2.7 stop()

```
void TB6612::stop ( )
```

12.32.3 Member Data Documentation

12.32.3.1 motor_pin

```
const int TB6612::motor_pin [private]
```

12.32.3.2 offset

```
bool TB6612::offset = true [private]
```

12.32.3.3 pwm_pin

```
const int TB6612::pwm_pin [private]
```

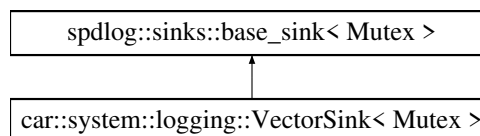
The documentation for this class was generated from the following files:

- repository/packages/t/tb6612/tb6612/include/TB6612.h
- repository/packages/t/tb6612/tb6612/src/TB6612.cpp

12.33 car::system::logging::VectorSink< Mutex > Class Template Reference

```
#include <VectorSink.h>
```

Inheritance diagram for car::system::logging::VectorSink< Mutex >:



Public Member Functions

- [VectorSink](#) (int [max_lines](#))
- void [sink_it_](#) (const spdlog::details::log_msg &msg) override
- void [flush_](#) () override
- const std::vector< std::string > & [get_log_messages](#) () const

Private Attributes

- const int [max_lines](#)
- std::vector< std::string > [log_messages](#)

12.33.1 Constructor & Destructor Documentation

12.33.1.1 VectorSink()

```
template<typename Mutex >
car::system::logging::VectorSink< Mutex >::VectorSink (
    int max_lines ) [inline]
```

12.33.2 Member Function Documentation

12.33.2.1 flush_()

```
template<typename Mutex >
void car::system::logging::VectorSink< Mutex >::flush_ ( ) [inline], [override]
```

12.33.2.2 get_log_messages()

```
template<typename Mutex >
const std::vector< std::string > & car::system::logging::VectorSink< Mutex >::get_log_↵
messages ( ) const [inline]
```

12.33.2.3 sink_it_()

```
template<typename Mutex >
void car::system::logging::VectorSink< Mutex >::sink_it_ (
    const spdlog::details::log_msg & msg ) [inline], [override]
```

12.33.3 Member Data Documentation

12.33.3.1 log_messages

```
template<typename Mutex >  
std::vector<std::string> car::system::logging::VectorSink< Mutex >::log_messages [private]
```

12.33.3.2 max_lines

```
template<typename Mutex >  
const int car::system::logging::VectorSink< Mutex >::max_lines [private]
```

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

- [common/include/car/system/logging/VectorSink.h](#)

Chapter 13

File Documentation

13.1 behaviour_tree/src/main.cpp File Reference

```
#include <iostream>
#include <chrono>
#include <filesystem>
#include <memory>
#include <cxxopts.hpp>
#include "car/system/CarSystem.h"
#include "car/system/device/lidar/LidarScanner.h"
#include "car/system/device/lidar/LidarDummy.h"
#include "car/system/movement/controller/DummyMovementController.h"
#include "car/system/movement/controller/DeviceMovementController.h"
#include "car/plugin/PluginManager.h"
#include "behaviour_tree/BehaviourTreeParser.hpp"
#include "behaviour_tree/node/custom/CarCustomNodeParser.hpp"
#include "behaviour_tree/BehaviourTreeHandler.hpp"
#include <thread>
#include <unistd.h>
#include <termios.h>
```

Functions

- int [kbhit](#) (void)
- int [main](#) (int argc, const char *argv[])

13.1.1 Function Documentation

13.1.1.1 kbhit()

```
int kbhit (
    void )
```

13.1.1.2 main()

```
int main (
    int argc,
    const char * argv[] )
```

13.2 daemon/src/main.cpp File Reference

```
#include <iostream>
#include <chrono>
#include <filesystem>
#include <memory>
#include <daemonpp/daemon.hpp>
#include <cpptrace/cpptrace.hpp>
#include <fmt/format.h>
#include <spdlog/sinks/callback_sink.h>
#include "car/system/CarSystem.h"
#include "car/system/device/lidar/LidarScanner.h"
#include "car/system/device/lidar/LidarDummy.h"
#include "car/system/movement/controller/DummyMovementController.h"
#include "car/system/movement/controller/DeviceMovementController.h"
#include "behaviour_tree/BehaviourTreeHandler.hpp"
#include "car/plugin/PluginManager.h"
```

Classes

- class [rpi_daemon](#)

Functions

- `std::unique_ptr< LidarDevice > getLidarDevice (std::shared_ptr< Configuration > configuration)`
- `std::unique_ptr< AbstractMovementController > getMovementController ()`
- `void terminate_handler ()`
- `int main (int argc, const char *argv[])`

13.2.1 Function Documentation

13.2.1.1 getLidarDevice()

```
std::unique_ptr< lidar::LidarDevice > getLidarDevice (
    std::shared_ptr< Configuration > configuration )
```

13.2.1.2 getMovementController()

```
std::unique_ptr< AbstractMovementController > getMovementController ( )
```

13.2.1.3 main()

```
int main (
    int argc,
    const char * argv[] )
```

13.2.1.4 terminate_handler()

```
void terminate_handler ( )
```

13.3 tui/src/main.cpp File Reference

```
#include <optional>
#include <string>
#include <thread>
#include <chrono>
#include <fmt/format.h>
#include "car/display/console/CarConsole.h"
#include "car/configuration/JsonConfiguration.cxx"
#include "car/system/CarSystem.h"
#include "car/system/device/DeviceManager.h"
#include "car/system/device/lidar/LidarDevice.h"
#include "car/system/device/lidar/LidarDummy.h"
#include "car/system/device/lidar/LidarScanner.h"
#include "car/system/device/CameraDevice.h"
#include "car/system/movement/controller/DummyMovementController.h"
#include "car/system/movement/controller/DeviceMovementController.h"
#include "car/plugin/PluginManager.h"
#include "car/system/logging/VectorSink.h"
#include "behaviour_tree/BehaviourTreeHandler.hpp"
```

Functions

- `std::unique_ptr< LidarDevice > getLidarDevice (std::shared_ptr< Configuration > configuration)`
- `std::unique_ptr< AbstractMovementController > getMovementController ()`
- `int main (int argc, char *argv[])`

13.3.1 Function Documentation

13.3.1.1 `getLidarDevice()`

```
std::unique_ptr< LidarDevice > getLidarDevice (
    std::shared_ptr< Configuration > configuration )
```

13.3.1.2 `getMovementController()`

```
std::unique_ptr< AbstractMovementController > getMovementController ( )
```

13.3.1.3 `main()`

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

13.4 `common/include/behaviour_tree/BehaviourTreeHandler.hpp` File Reference

```
#include <string>
#include <vector>
#include <nod/nod.hpp>
#include "utils/Utility.hpp"
#include "car/plugin/Plugin.h"
#include "behaviour_tree/BehaviourTreeParser.hpp"
#include "behaviour_tree/node/custom/CarCustomNodeParser.hpp"
#include "CarContext.hpp"
```

Classes

- class `behaviour_tree::BehaviourTreeHandler`

Namespaces

- namespace `behaviour_tree`

13.5 BehaviourTreeHandler.hpp

[Go to the documentation of this file.](#)

```

1  #ifndef BEHAVIOURTREEHANDLER_HPP
2  #define BEHAVIOURTREEHANDLER_HPP
3
4  #pragma once
5
6  #include <string>
7  #include <vector>
8
9  #include <nod/nod.hpp>
10
11 #include "utils/Utility.hpp"
12
13 #include "car/plugin/Plugin.h"
14
15 #include "behaviour_tree/BehaviourTreeParser.hpp"
16 #include "behaviour_tree/node/custom/CarCustomNodeParser.hpp"
17
18 #include "CarContext.hpp"
19
20 namespace behaviour_tree
21 {
22     class BehaviourTreeHandler : public car::plugin::Plugin
23     {
24     public:
25         void initialize(std::shared_ptr<car::system::CarSystem> car_system) final override
26         {
27             this->car_system = car_system;
28             // The BehaviourTreeParser does not come with a CustomNodeParser since each program can have
29             // a different set of Action nodes
30
31             BehaviourTreeParser::instance().setCustomNodeParser(std::make_shared<node::custom::CarCustomNodeParser>(CarCustomNodeParser));
32
33             this->car_system->getMessagingSystem()->getCommandSignal().connect(std::bind(&BehaviourTreeHandler::handleCommand,
34             this, std::placeholders::_1, std::placeholders::_2));
35         }
36
37         void handleCommand(const std::string message, const rapidjson::Document &message_json)
38         {
39             const std::string command = message_json["command"].GetString();
40             if (command != "behaviour_tree")
41             {
42                 spdlog::error(R"(The property "command" does not match "behaviour_tree", {})", command);
43                 return;
44             }
45             if (!message_json.HasMember("action") || !message_json["action"].IsString())
46             {
47                 spdlog::error(R"(The property "action" does not exist in the given json.)");
48                 return;
49             }
50             const std::string action = message_json["action"].GetString();
51             switch (utils::hash(action))
52             {
53             case utils::hash("set"):
54             {
55                 this->setBehaviourTree(message_json);
56                 break;
57             }
58             case utils::hash("start"):
59             {
60                 this->startBehaviourTree();
61                 break;
62             }
63             case utils::hash("stop"):
64             {
65                 this->stopBehaviourTree();
66                 break;
67             }
68             default:
69                 spdlog::error(R"(The property "action" does not match "set" or "start", {})", action);
70                 break;
71             };
72         }
73
74         void setBehaviourTree(const rapidjson::Document &message_json)
75         {
76             if (!message_json.HasMember("data") || !message_json["data"].IsString())
77             {
78                 spdlog::error(R"(The property "data" does not exist in the given json.)");
79                 return;
80             }
81             try
82             {
83

```

```

79         auto maybe_behaviour_tree =
BehaviourTreeParser::instance().parseXML(message_json["data"].GetString());
80         if (!maybe_behaviour_tree.has_value())
81         {
82             spdlog::error(R"(Unable to parse the given behaviour tree | {})",
maybe_behaviour_tree.error());
83             return;
84         }
85         auto &behaviour_tree = maybe_behaviour_tree.value();
86         spdlog::info("Behaviour tree parsed successfully | {}", behaviour_tree->toString());
87         this->_setBehaviourTree(behaviour_tree);
88     }
89     catch (std::exception &e)
90     {
91         spdlog::error("An error has occurred while parsing the given behaviour tree: {}",
e.what());
92     }
93 }
94
95 void startBehaviourTree()
96 {
97     assert(this->car_system != nullptr);
98     if (this->behaviour_tree == nullptr)
99     {
100         spdlog::error("The Behaviour tree has not been set");
101         return;
102     }
103     this->behaviour_tree->resetCycles();
104     this->tick_count = 0;
105     std::shared_ptr<Context> context = std::make_shared<CarContext>(this->behaviour_tree,
this->car_system);
106     this->context = context;
107     spdlog::info("Starting the given Behaviour tree");
108 }
109
110 void stopBehaviourTree()
111 {
112     assert(this->car_system != nullptr);
113     this->context = nullptr;
114     spdlog::info("Stopped any Behaviour Tree context");
115 }
116
117 void update() final override
118 {
119     if (this->context == nullptr)
120     {
121         return;
122     }
123     if (this->context->canRun())
124     {
125         const std::chrono::time_point<std::chrono::steady_clock> now =
std::chrono::steady_clock::now();
126         // TODO:
127         if (now - this->last_connected >=
this->car_system->getConfiguration()->behaviour_tree_update_ms_interval) {
128             this->context->update(this->tick_count);
129             this->tick_count++;
130             this->last_connected = now;
131         }
132     }
133     else
134     {
135         this->context = nullptr;
136     }
137 }
138
139 void stop() final override
140 {
141     this->context = nullptr;
142 }
143
144 std::string getName() final override
145 {
146     return "BehaviourTreeHandler";
147 }
148
149 void _setBehaviourTree(std::shared_ptr<BehaviourTree> behaviour_tree)
150 {
151     this->behaviour_tree = behaviour_tree;
152 }
153
154 private:
155     std::shared_ptr<car::system::CarSystem> car_system;
156
157     std::shared_ptr<BehaviourTree> behaviour_tree;
158     std::shared_ptr<Context> context;
159

```



```
160         int tick_count = 0;
161
162         // This is initialized as 0
163         std::chrono::time_point<std::chrono::steady_clock> last_connected;
164     };
165 } // namespace behaviour_tree
166
167 #endif
```

13.6 common/include/behaviour_tree/CarContext.hpp File Reference

```
#include "car/system/CarSystem.h"
#include "behaviour_tree/Context.h"
```

Classes

- class `behaviour_tree::CarContext`

Namespaces

- namespace `behaviour_tree`

13.7 CarContext.hpp

[Go to the documentation of this file.](#)

```
1 #ifndef BEHAVIOUR_TREE_CARCONTEXT_HPP
2 #define BEHAVIOUR_TREE_CARCONTEXT_HPP
3
4 #pragma once
5
6 #include "car/system/CarSystem.h"
7 #include "behaviour_tree/Context.h"
8
9 namespace behaviour_tree
10 {
11     class CarContext : public Context
12     {
13     public:
14         CarContext(std::shared_ptr<BehaviourTree> behaviour_tree, std::shared_ptr<car::system::CarSystem>
car_system) : Context(std::move(behaviour_tree)), car_system(std::move(car_system))
15         {
16         }
17
18         std::shared_ptr<car::system::CarSystem> getCarSystem() const
19         {
20             return this->car_system;
21         }
22
23         void _() override{};
24
25     private:
26         std::shared_ptr<car::system::CarSystem> car_system;
27     };
28 }
29
30 #endif
```

13.8 common/include/car/configuration/Configuration.h File Reference

```
#include <chrono>
#include <optional>
#include <string>
#include <tl/expected.hpp>
```

Classes

- struct [car::configuration::Configuration](#)

Namespaces

- namespace [car](#)
- namespace [car::configuration](#)

13.9 Configuration.h

[Go to the documentation of this file.](#)

```
1 #ifndef CONFIGURATION_H
2 #define CONFIGURATION_H
3
4 #pragma once
5
6 #include <chrono>
7 #include <optional>
8 #include <string>
9
10 #include <tl/expected.hpp>
11
12 namespace car::configuration
13 {
14     struct Configuration
15     {
16         std::string host = "127.0.0.1:3000";
17
18         int camera_index = 0;
19         void setCameraFps(const int camera_fps)
20         {
21             this->camera_fps = camera_fps;
22             this->camera_fps_interval = 1000 / camera_fps;
23         }
24         const int getCameraFpsInterval() { return this->camera_fps_interval; }
25         bool use_camera = true;
26
27         std::string lidar_port = "";
28         bool use_lidar = true;
29
30         std::chrono::milliseconds behaviour_tree_update_ms_interval = std::chrono::milliseconds(100);
31
32     private:
33         int camera_fps = 60;
34         int camera_fps_interval = 1000;
35     };
36 };
37
38 #endif
```

13.10 common/include/car/plugin/Plugin.h File Reference

```
#include <string>
#include <memory>
```

Classes

- class [car::plugin::Plugin](#)

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::plugin](#)

13.11 Plugin.h

[Go to the documentation of this file.](#)

```
1 #ifndef PLUGIN_H
2 #define PLUGIN_H
3
4 #pragma once
5
6 #include <string>
7 #include <memory>
8
9 namespace car::system
10 {
11     class CarSystem;
12 }
13
14 namespace car::plugin
15 {
16     class Plugin
17     {
18     public:
19         virtual void initialize(std::shared_ptr<car::system::CarSystem> car_system) = 0;
20         virtual void update() = 0;
21         virtual void stop() = 0;
22         virtual std::string getName() = 0;
23     };
24 }
25
26 #endif
```

13.12 common/include/car/plugin/PluginManager.h File Reference

```
#include <vector>
#include <memory>
#include "utils/Utility.hpp"
#include "utils/TypeName.hpp"
#include "Plugin.h"
```

Classes

- class [car::plugin::PluginManager](#)

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::plugin](#)

13.13 PluginManager.h

[Go to the documentation of this file.](#)

```

1  #ifndef PLUGIN_MANAGER_H
2  #define PLUGIN_MANAGER_H
3
4  #pragma once
5
6  #include <vector>
7  #include <memory>
8
9  #include "utils/Utility.hpp"
10 #include "utils/TypeName.hpp"
11
12 #include "Plugin.h"
13
14 namespace car::system
15 {
16     class CarSystem;
17 }
18
19 namespace car::plugin
20 {
21     class PluginManager
22     {
23     public:
24         void initialize(std::shared_ptr<system::CarSystem> car_system)
25         {
26             for (std::shared_ptr<Plugin>& plugin : this->plugins)
27             {
28                 plugin->initialize(car_system);
29             }
30         }
31
32         void update()
33         {
34             for (std::shared_ptr<Plugin>& plugin : this->plugins)
35             {
36                 plugin->update();
37             }
38         }
39
40         void stop()
41         {
42             for (std::shared_ptr<Plugin>& plugin : this->plugins)
43             {
44                 plugin->stop();
45             }
46         }
47
48         void terminate()
49         {
50             this->stop();
51         }
52
53         void addPlugin(std::shared_ptr<Plugin> plugin)
54         {
55             this->plugins.push_back(plugin);
56         }
57
58         template<typename T>
59         std::shared_ptr<T> getPlugin()
60         {
61             static_assert(std::is_base_of<Plugin, T>::value, "T must be a Plugin");
62             std::string type_name = std::string(utils::TypeName<T>());
63             type_name = utils::getStringAfterLastColon(type_name);
64
65             for (std::shared_ptr<Plugin>& plugin : this->plugins)
66             {
67                 if (plugin->getName() == type_name)
68                 {
69                     return plugin;
70                 }
71             }
72
73             return nullptr;
74         }
75
76     private:
77         std::vector<std::shared_ptr<Plugin>> plugins;
78     };
79 }
80
81
82 #endif

```

13.14 common/include/car/system/CarSystem.h File Reference

```
#include <memory>
#include "car/configuration/Configuration.h"
#include "car/system/device/DeviceManager.h"
#include "car/system/messaging/MessagingSystem.h"
#include "car/system/movement/MovementSystem.h"
#include "car/plugin/PluginManager.h"
```

Classes

- class [car::system::CarSystem](#)

Namespaces

- namespace [car](#)
- namespace [car::system](#)

13.15 CarSystem.h

[Go to the documentation of this file.](#)

```
1 #ifndef CARSYSTEM_H
2 #define CARSYSTEM_H
3
4 #pragma once
5
6 #include <memory>
7
8 #include "car/configuration/Configuration.h"
9
10 #include "car/system/device/DeviceManager.h"
11 #include "car/system/messaging/MessagingSystem.h"
12 #include "car/system/movement/MovementSystem.h"
13
14 #include "car/plugin/PluginManager.h"
15
16 using namespace car::configuration;
17 using namespace car::plugin;
18 using namespace car::system::device;
19 using namespace car::system::messaging;
20 using namespace car::system::movement;
21
22 namespace car::system
23 {
24     // Make sure this is stored as a shared_ptr
25     class CarSystem : public std::enable_shared_from_this<CarSystem>
26     {
27     public:
28         CarSystem(
29             std::shared_ptr<Configuration> configuration,
30             std::unique_ptr<DeviceManager> device_manager,
31             std::unique_ptr<MessagingSystem> messaging_system,
32             std::unique_ptr<MovementSystem> movement_system,
33             std::unique_ptr<PluginManager> plugin_manager);
34
35         void initialize();
36         void reload();
37
38         void start();
39         void stop();
40
41         tl::expected<nullptr_t, std::string> tryConnect();
42         void disconnect();
43
44         void terminate();
45     }
```

```

46     void update();
47
48     const std::shared_ptr<Configuration> getConfiguration() const { return this->configuration_; };
49     void setConfiguration(std::shared_ptr<Configuration> configuration);
50
51     DeviceManager *getDeviceManager() const
52     {
53         return this->device_manager_.get();
54     }
55
56     MessagingSystem *getMessagingSystem() const
57     {
58         return this->messaging_system_.get();
59     }
60
61     MovementSystem *getMovementSystem() const
62     {
63         return this->movement_system_.get();
64     }
65
66     template <typename T>
67     const std::shared_ptr<T> getPlugin() const { return this->plugin_manager_->getPlugin<T>(); }
68
69 private:
70     void sendData();
71
72     std::shared_ptr<Configuration> configuration_;
73
74     const std::unique_ptr<DeviceManager> device_manager_;
75     const std::unique_ptr<MessagingSystem> messaging_system_;
76     const std::unique_ptr<MovementSystem> movement_system_;
77     const std::unique_ptr<PluginManager> plugin_manager_;
78
79     bool initialized = false;
80     bool started = false;
81 };
82 }
83
84 #endif

```

13.16 common/include/car/system/device/CameraDevice.h File Reference

```

#include <vector>
#include <tl/expected.hpp>
#include <opencv2/opencv.hpp>
#include "car/configuration/Configuration.h"

```

Classes

- class [car::system::device::CameraDevice](#)

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::system::device](#)

13.17 CameraDevice.h

[Go to the documentation of this file.](#)

```

1  #ifndef CAMERADEVICE_H
2  #define CAMERADEVICE_H
3
4  #pragma once
5
6  #include <vector>
7
8  #include <tl/expected.hpp>
9  #include <opencv2/opencv.hpp>
10
11 #include "car/configuration/Configuration.h"
12
13 namespace car::system::device
14 {
15     class DeviceManager;
16     class CameraDevice
17     {
18     public:
19         CameraDevice(std::shared_ptr<configuration::Configuration> configuration) :
20             configuration(configuration) {}
21
22         CameraDevice(const CameraDevice&) = delete;
23         CameraDevice& operator=(const CameraDevice&) = delete;
24
25         CameraDevice(CameraDevice&&) = delete;
26         CameraDevice& operator=(CameraDevice&&) = delete;
27
28         ~CameraDevice() = default;
29
30     public:
31         [[nodiscard]] static tl::expected<std::unique_ptr<CameraDevice>, std::string>
32             create(std::shared_ptr<configuration::Configuration> configuration);
33         std::string getFrameBuffer() const;
34
35     protected:
36         void start();
37         void update();
38         void stop();
39         void disconnect();
40         void terminate();
41
42     friend class DeviceManager;
43
44     private:
45         std::shared_ptr<configuration::Configuration> configuration;
46
47         std::unique_ptr<cv::VideoCapture> camera_;
48
49         bool connected_ = false;
50         std::string frame_buffer_;
51
52         std::mutex camera_mutex_;
53
54         std::chrono::steady_clock::time_point last;
55     };
56 #endif

```

13.18 common/include/car/system/device/DeviceManager.h File Reference

```

#include <memory>
#include <tl/expected.hpp>
#include "car/configuration/Configuration.h"
#include "CameraDevice.h"
#include "lidar/LidarDevice.h"
#include "lidar/LidarScanner.h"

```

Classes

- class `car::system::device::DeviceManager`

Namespaces

- namespace `car`
- namespace `car::system`
- namespace `car::system::device`

13.19 DeviceManager.h

[Go to the documentation of this file.](#)

```

1 #ifndef DEVICE_MANAGER_H
2 #define DEVICE_MANAGER_H
3
4 #pragma once
5
6 #include <memory>
7
8 #include <tl/expected.hpp>
9
10 #include "car/configuration/Configuration.h"
11
12 #include "CameraDevice.h"
13 #include "lidar/LidarDevice.h"
14 #include "lidar/LidarScanner.h"
15
16 using namespace car::configuration;
17
18 namespace car::system
19 {
20     class CarSystem;
21 }
22
23 namespace car::system::device
24 {
25     class DeviceManager {
26     public:
27         [[nodiscard]] static tl::expected<std::unique_ptr<DeviceManager>, std::string>
28         create(std::shared_ptr<Configuration> configuration);
29
30         DeviceManager(std::unique_ptr<CameraDevice> camera_device, std::unique_ptr<lidar::LidarDevice>
31         lidar_device) :
32             camera_device_(std::move(camera_device)),
33             lidar_device_(std::move(lidar_device))
34         {
35         }
36
37         CameraDevice* getCameraDevice() {
38             return this->camera_device_.get();
39         }
40
41         lidar::LidarDevice* getLidarDevice() {
42             return this->lidar_device_.get();
43         }
44
45         const bool isRunning() const {
46             return this->is_running_;
47         }
48
49         void initialize(std::shared_ptr<system::CarSystem> car_system);
50         void start();
51         void update();
52         void stop();
53         void terminate();
54
55     private:
56         std::shared_ptr<car::system::CarSystem> car_system;
57
58         bool is_initialized_ = false;
59         bool is_running_ = false;
60
61         std::unique_ptr<lidar::LidarDevice> lidar_device_;
62         std::unique_ptr<CameraDevice> camera_device_;
63     };
64 }
65 #endif

```


13.20 common/include/car/system/device/lidar/LidarDevice.h File Reference

```
#include <vector>
#include <rapidjson/document.h>
#include <RPLidar.h>
```

Classes

- class [car::system::device::lidar::LidarDevice](#)

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::system::device](#)
- namespace [car::system::device::lidar](#)

13.21 LidarDevice.h

[Go to the documentation of this file.](#)

```
1 #ifndef LIDARDEVICE_H
2 #define LIDARDEVICE_H
3
4 #pragma once
5
6 #include <vector>
7
8 #include <rapidjson/document.h>
9
10 #include <RPLidar.h>
11
12 using namespace rplidar;
13
14 namespace car::system::device {
15     class DeviceManager;
16 }
17
18 namespace car::system::device::lidar
19 {
20     class LidarDevice
21     {
22     public:
23         std::vector<Measure> getScanData() const { return this->scan_data_; }
24
25         virtual void start() = 0;
26         virtual void update() = 0;
27         virtual void stop() = 0;
28
29         virtual void initialize() = 0;
30         virtual void terminate() = 0;
31         virtual void disconnect() = 0;
32
33     protected:
34         friend class DeviceManager;
35
36         void setScanData(const std::vector<Measure>& scan_data)
37         {
38             this->scan_data_ = scan_data;
39         }
40
41         std::vector<Measure> scan_data_;
42     };
43 }
44
45 #endif
```

13.22 common/include/car/system/device/lidar/LidarDummy.h File Reference

```
#include <fstream>
#include <spdlog/spdlog.h>
#include "LidarDevice.h"
```

Classes

- class [car::system::device::lidar::LidarDummy](#)

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::system::device](#)
- namespace [car::system::device::lidar](#)

13.23 LidarDummy.h

[Go to the documentation of this file.](#)

```
1 #ifndef LIDARDUMMY_H
2 #define LIDARDUMMY_H
3
4 #pragma once
5
6 #include <fstream>
7 #include <spdlog/spdlog.h>
8
9 #include "LidarDevice.h"
10
11 namespace car::system::device::lidar
12 {
13     class LidarDummy final : public LidarDevice
14     {
15     public:
16         LidarDummy()
17         {
18             spdlog::warn("Currently using the LidarDummy");
19         };
20
21         void start() final override {};
22         void update() final override {};
23         void stop() final override {};
24         void initialize() final override {};
25         void terminate() final override {};
26         void disconnect() final override {};
27
28     private:
29     };
30 }
31
32 #endif
```

13.24 common/include/car/system/device/lidar/LidarScanner.h File Reference

```
#include "LidarDevice.h"
#include <memory>
#include <variant>
#include <RPLidar.h>
#include <tl/expected.hpp>
#include "car/configuration/Configuration.h"
```

Classes

- class `car::system::device::lidar::LidarScanner`

Namespaces

- namespace `car`
- namespace `car::system`
- namespace `car::system::device`
- namespace `car::system::device::lidar`

13.25 LidarScanner.h

[Go to the documentation of this file.](#)

```

1 #ifndef LIDARSCANNER_H
2 #define LIDARSCANNER_H
3
4 #pragma once
5
6 #include "LidarDevice.h"
7
8 #include <memory>
9 #include <variant>
10
11 #include <RPLidar.h>
12 #include <tl/expected.hpp>
13
14 #include "car/configuration/Configuration.h"
15
16 using namespace rplidar;
17
18 namespace car::system::device::lidar
19 {
20     class LidarScanner final : public LidarDevice
21     {
22     public:
23         [[nodiscard]] static tl::expected<std::unique_ptr<LidarScanner>, std::string>
24         create(std::shared_ptr<configuration::Configuration> configuration) noexcept
25         {
26             auto maybe_lidar = RPLidar::create(configuration->lidar_port);
27             if (maybe_lidar.has_value())
28             {
29                 return std::make_unique<LidarScanner>(configuration, std::move(maybe_lidar.value()));
30             }
31             else
32             {
33                 return tl::make_unexpected(maybe_lidar.error());
34             }
35         }
36
37         // Do not call this constructor directly. Use the create method instead.
38         LidarScanner(std::shared_ptr<configuration::Configuration> configuration,
39                     std::unique_ptr<RPLidar> lidar) : configuration_(configuration), lidar_(std::move(lidar)) {}
40
41         void start() final override
42         {
43             this->running = true;
44             this->lidar_->start_motor();
45             std::lock_guard<std::mutex> lock(this->scan_data_mutex_);
46             this->scan_generator_ = this->lidar_->iter_scans();
47         }
48
49         void update() final override
50         {
51             if (this->running) {
52                 std::lock_guard<std::mutex> lock(this->scan_data_mutex_);
53                 const auto& scan_generator =
54                     std::get<std::function<std::vector<Measure>()>>(this->scan_generator_);
55                 this->set_scan_data(scan_generator());
56             }
57         }
58
59         void stop() final override

```

```

58     {
59         if (this->running) {
60             this->running = false;
61             std::lock_guard<std::mutex> lock(this->scan_data_mutex_);
62             this->scan_generator_ = nullptr;
63             this->lidar_->stop();
64             this->lidar_->stop_motor();
65         }
66     }
67
68     void initialize() final override
69     {
70     };
71
72     void disconnect() final override
73     {
74         if (this->running) {
75             this->running = false;
76             std::lock_guard<std::mutex> lock(this->scan_data_mutex_);
77             this->scan_generator_ = nullptr;
78             this->lidar_->disconnect();
79         }
80     }
81
82     void terminate() final override
83     {
84         this->stop();
85         this->disconnect();
86     }
87
88     private:
89         std::atomic_bool running = false;
90
91         std::shared_ptr<configuration::Configuration> configuration_;
92
93         std::vector<Measure> scan_data_;
94
95         std::unique_ptr<RPLidar> lidar_;
96         std::variant<std::function<std::vector<Measure>()>, nullptr_t> scan_generator_ = nullptr;
97
98         std::mutex scan_data_mutex_;
99     };
100 }
101
102 #endif

```

13.26 common/include/car/system/logging/VectorSink.h File Reference

```

#include <algorithm>
#include <vector>
#include <fmt/format.h>
#include <spdlog/sinks/base_sink.h>
#include <spdlog/details/synchronous_factory.h>
#include <iostream>

```

Classes

- class [car::system::logging::VectorSink< Mutex >](#)

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::system::logging](#)

Typedefs

- using `car::system::logging::vector_sink_mt` = `VectorSink< std::mutex >`

13.27 VectorSink.h

[Go to the documentation of this file.](#)

```

1 #ifndef VECTORSINK_CXX
2 #define VECTORSINK_CXX
3
4 #include <algorithm>
5 #include <vector>
6
7 #include <fmt/format.h>
8
9 #include <spdlog/sinks/base_sink.h>
10 #include <spdlog/details/synchronous_factory.h>
11 #include <iostream>
12
13 namespace car::system::logging
14 {
15     template <typename Mutex>
16     class VectorSink : public spdlog::sinks::base_sink<Mutex>
17     {
18     public:
19         VectorSink(int max_lines) : max_lines(max_lines)
20         {
21         }
22
23         void sink_it_(const spdlog::details::log_msg &msg) override
24         {
25             spdlog::memory_buf_t formatted;
26             spdlog::sinks::base_sink<Mutex>::formatter_>format(msg, formatted);
27             if (this->log_messages.size() < this->max_lines)
28             {
29                 this->log_messages.push_back(std::string(formatted.data(), formatted.size()));
30             }
31             else
32             {
33                 std::rotate(this->log_messages.begin(), this->log_messages.begin() + 1,
34                     this->log_messages.end());
35                 this->log_messages[this->log_messages.size() - 1] = std::string(formatted.data(),
36                     formatted.size());
37             }
38         };
39
40         void flush_() override
41         {
42             this->log_messages.clear();
43         };
44
45         const std::vector<std::string> &get_log_messages() const
46         {
47             return this->log_messages;
48         }
49
50     private:
51         const int max_lines;
52
53         std::vector<std::string> log_messages;
54     };
55
56     using vector_sink_mt = VectorSink<std::mutex>;
57 }
58 #endif

```

13.28 common/include/car/system/messaging/MessagingSystem.h File Reference

```

#include <functional>
#include <memory>
#include <ixwebsocket/IXNetSystem.h>

```

```
#include <ixwebsocket/IXWebSocket.h>
#include <nod/nod.hpp>
#include <rapidjson/rapidjson.h>
#include <rapidjson/document.h>
#include "utils/Utility.hpp"
#include "car/configuration/Configuration.h"
```

Classes

- class [car::system::messaging::MessagingSystem](#)
- struct [car::system::messaging::MessagingSystem::FirstMessageStruct](#)

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::system::messaging](#)

13.29 MessagingSystem.h

[Go to the documentation of this file.](#)

```
1 #ifndef MESSAGINGSYSTEM_H
2 #define MESSAGINGSYSTEM_H
3
4 #pragma once
5
6 #include <functional>
7 #include <memory>
8
9 #include <ixwebsocket/IXNetSystem.h>
10 #include <ixwebsocket/IXWebSocket.h>
11
12 #include <nod/nod.hpp>
13
14 #include <rapidjson/rapidjson.h>
15 #include <rapidjson/document.h>
16
17 #include "utils/Utility.hpp"
18
19 #include "car/configuration/Configuration.h"
20
21 namespace car::system::messaging
22 {
23     class MessagingSystem
24     {
25     public:
26         MessagingSystem();
27
28         void initialize(std::shared_ptr<configuration::Configuration> configuration);
29         void initializeWebSocket();
30         const tl::expected<nullptr_t, std::string> tryConnect();
31         void stop();
32         void terminate();
33
34         // Necessary for the reloading the configuration
35         void setConfiguration(std::shared_ptr<configuration::Configuration> configuration);
36
37         nod::signal<void(const std::string, const rapidjson::Document*)>& getCommandSignal() { return
this->command_signal_; }
38         nod::signal<void(const std::string, const rapidjson::Document*)>& getSelectionSignal() { return
this->selection_signal_; }
39         nod::signal<void(const std::string)>& getMessageSignal() { return this->message_signal_; }
40         nod::signal<void(const std::string)>& getDisconnectSignal() { return this->on_disconnect_signal_;
}
41
42         void onMessageCallback(const ix::WebSocketMessagePtr& msg) const;
43         void onDisconnect(const std::string);
```

```

44
45     const std::string getUUID() const { return this->uuid_; }
46     void handleMessage(const std::string& message) const;
47     void sendMessage(const std::string& message);
48
49     struct FirstMessageStruct
50     {
51         std::string error_message;
52         std::string uuid;
53         std::condition_variable condition;
54     };
55     void onFirstMessage(const ix::WebSocketMessagePtr& msg, FirstMessageStruct&
first_message_struct);
56
57     const bool isConnected() const { return this->connected_; }
58
59     nod::signal<void(std::string)> on_disconnect_signal_;
60
61     nod::signal<void(const std::string)> message_signal_;
62     nod::signal<void(const std::string, const rapidjson::Document&)> command_signal_;
63     nod::signal<void(const std::string, const rapidjson::Document&)> selection_signal_;
64
65     private:
66         tl::expected<std::string, std::string> getFirstMessage();
67
68         std::shared_ptr<configuration::Configuration> configuration_;
69
70         std::unique_ptr<ix::WebSocket> websocket_;
71         std::string websocket_url_;
72
73         std::string uuid_;
74
75         bool connected_ = false;
76     };
77 };
78
79 #endif

```

13.30 common/include/car/system/messaging/StreamType.h File Reference

Enumerations

- enum [StreamType](#) { [None](#) = 0 , [Lidar](#) , [Camera](#) , [Both](#) }

13.30.1 Enumeration Type Documentation

13.30.1.1 StreamType

enum [StreamType](#)

Enumerator

None	
Lidar	
Camera	
Both	

13.31 StreamType.h

[Go to the documentation of this file.](#)

```
1 #ifndef STREAM_TYPE_H
2 #define STREAM_TYPE_H
3
4 #pragma once
5
6 enum StreamType {
7     None = 0,
8     Lidar,
9     Camera,
10    Both,
11 };
12
13 #endif
```

13.32 common/include/car/system/movement/controller/AbstractMovementController.h File Reference

Classes

- class [car::system::movement::controller::AbstractMovementController](#)

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::system::movement](#)
- namespace [car::system::movement::controller](#)

13.33 AbstractMovementController.h

[Go to the documentation of this file.](#)

```
1 #ifndef ABSTRACTWHEELCONTROLLER_H
2 #define ABSTRACTWHEELCONTROLLER_H
3
4 #pragma once
5
6 namespace car::system::movement::controller
7 {
8     class AbstractMovementController
9     {
10     public:
11         virtual void initialize() = 0;
12         virtual void stop() = 0;
13         virtual void terminate() = 0;
14
15         virtual void setRearWheelsSpeed(const int speed) = 0;
16
17         virtual void setRearLeftWheelSpeed(const int speed) = 0;
18         virtual void setRearRightWheelSpeed(const int speed) = 0;
19
20         virtual void setFrontWheelsAngle(const float angle) = 0;
21         virtual void setCameraServo1Angle(const float angle) = 0;
22         virtual void setCameraServo2Angle(const float angle) = 0;
23
24         virtual void setRearWheelsDirectionToForward() = 0;
25         virtual void setRearLeftWheelDirectionToForward() = 0;
26         virtual void setRearRightWheelDirectionToForward() = 0;
27
28         virtual void setRearWheelsDirectionToBackward() = 0;
29         virtual void setRearLeftWheelDirectionToBackward() = 0;
30         virtual void setRearRightWheelDirectionToBackward() = 0;
31     };
32 } // namespace car::system::movement::controller
33
34 #endif
```


13.34 common/include/car/system/movement/controller/DeviceMovementController.h File Reference

13.35 DeviceMovementController.h

[Go to the documentation of this file.](#)

```

1 #ifndef __linux__
2 #ifndef DEVICEMOVEMENTCONTROLLER_H
3 #define DEVICEMOVEMENTCONTROLLER_H
4
5 #pragma once
6
7 #include <memory>
8
9 #include "AbstractMovementController.h"
10
11 #include "car/system/movement/devices/Servo.h"
12 #include "car/system/movement/devices/RearWheel.h"
13
14 using namespace car::system::movement::devices;
15
16 namespace car::system::movement::controller
17 {
18     static constexpr int Motor_A = 17;
19     static constexpr int Motor_B = 27;
20     static constexpr int PWM_A = 4;
21     static constexpr int PWM_B = 5;
22
23     static constexpr int MIN_PULSE_WIDTH = 900;
24     static constexpr int MAX_PULSE_WIDTH = 2100;
25     static constexpr int FREQUENCY = 50;
26
27     static constexpr int BUS_NUMBER = 1;
28
29     class DeviceMovementController : public AbstractMovementController
30     {
31     public:
32         [[nodiscard]] DeviceMovementController();
33
34         void initialize() final override;
35
36         void stop() final override;
37
38         void terminate() final override;
39
40         void setRearWheelsSpeed(const int speed) final override;
41
42         void setRearLeftWheelSpeed(const int speed) final override;
43
44         void setRearRightWheelSpeed(const int speed) final override;
45
46         void setFrontWheelsAngle(const float angle) final override;
47
48         void setCameraServo1Angle(const float angle) final override;
49
50         void setCameraServo2Angle(const float angle) final override;
51
52         void setRearWheelsDirectionToForward() final override;
53
54         void setRearLeftWheelDirectionToForward() final override;
55
56         void setRearRightWheelDirectionToForward() final override;
57
58         void setRearWheelsDirectionToBackward() final override;
59
60         void setRearLeftWheelDirectionToBackward() final override;
61
62         void setRearRightWheelDirectionToBackward() final override;
63
64     private:
65         std::shared_ptr<PCA9685> pwm;
66
67         std::unique_ptr<Servo> front_wheels_;
68         std::unique_ptr<Servo> camera_servo_1_;
69         std::unique_ptr<Servo> camera_servo_2_;
70
71         std::unique_ptr<RearWheel> rear_left_wheel_;
72         std::unique_ptr<RearWheel> rear_right_wheel_;
73     };
74 } // namespace car::system::movement::controller
75

```

```
76 #endif
77 #endif // __linux__
```

13.36 common/include/car/system/movement/controller/DummyMovementController.h File Reference

```
#include "AbstractMovementController.h"
```

Classes

- class `car::system::movement::controller::DummyMovementController`

Namespaces

- namespace `car`
- namespace `car::system`
- namespace `car::system::movement`
- namespace `car::system::movement::controller`

13.37 DummyMovementController.h

[Go to the documentation of this file.](#)

```
1 #ifndef DUMMYWHEELCONTROLLER_H
2 #define DUMMYWHEELCONTROLLER_H
3
4 #pragma once
5
6 #include "AbstractMovementController.h"
7
8 namespace car::system::movement::controller
9 {
10     class DummyMovementController : public AbstractMovementController
11     {
12     public:
13         void initialize() final override {};
14
15         void stop() final override;
16
17         void terminate() final override {};
18
19         void setRearWheelsSpeed(const int speed) final override;
20
21         void setRearLeftWheelSpeed(const int speed) final override;
22
23         void setRearRightWheelSpeed(const int speed) final override;
24
25         void setFrontWheelsAngle(const float angle) final override;
26
27         void setCameraServo1Angle(const float angle) final override;
28
29         void setCameraServo2Angle(const float angle) final override;
30
31         void setRearWheelsDirectionToForward() final override;
32
33         void setRearLeftWheelDirectionToForward() final override;
34
35         void setRearRightWheelDirectionToForward() final override;
36
37         void setRearWheelsDirectionToBackward() final override;
38
39         void setRearLeftWheelDirectionToBackward() final override;
40
41         void setRearRightWheelDirectionToBackward() final override;
42
43     private:
44     };
45 } // namespace car::system::movement::controller
46
47 #endif
```

13.38 common/include/car/system/movement/devices/RearWheel.h File Reference

13.39 RearWheel.h

[Go to the documentation of this file.](#)

```
1 #ifndef __linux__
2 #ifndef REARWHEEL_H
3 #define REARWHEEL_H
4
5 #include <memory>
6
7 #include <PCA9685.h>
8 #include <TB6612.h>
9
10 // Made with the help of ChatGPT
11
12 namespace car::system::movement::devices
13 {
14     class RearWheel
15     {
16     public:
17         RearWheel(std::shared_ptr<PCA9685> pwm, std::unique_ptr<TB6612> motor);
18
19         void forward();
20
21         void backward();
22
23         void stop();
24
25         int getSpeed() const;
26
27         void setSpeed(const int speed);
28
29         void ready();
30
31     private:
32         std::shared_ptr<PCA9685> pwm_;
33         std::unique_ptr<TB6612> motor_;
34
35         int speed_;
36     };
37 } // namespace car::system::movement::wheels
38
39 #endif
40 #endif
```

13.40 common/include/car/system/movement/devices/Servo.h File Reference

13.41 Servo.h

[Go to the documentation of this file.](#)

```
1 #ifndef __linux__
2 #ifndef SERVO_H
3 #define SERVO_H
4
5 #include <algorithm>
6 #include <memory>
7
8 #include <PCA9685.h>
9
10 namespace car::system::movement::devices
11 {
12     class Servo
13     {
14     private:
15         static int map(int x, int in_min, int in_max, int out_min, int out_max)
16         {
17             return ((x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min);
18         }
19     };
20 }
21
```

```

22     }
23
24     static constexpr int MIN_PULSE_WIDTH = 900;
25     static constexpr int MAX_PULSE_WIDTH = 2100;
26     static constexpr int FREQUENCY = 50;
27
28     public:
29         Servo(std::shared_ptr<PCA9685> pwm, int channel);
30
31         // Some of the code was from: https://github.com/chaoticmachinery/pca9685
32         int getAnalogAngle() const;
33
34         int getAngle() const;
35
36         // Some of the code was from: https://github.com/chaoticmachinery/pca9685
37         void setAngle(const int angle);
38
39         void reset();
40
41     private:
42         const std::shared_ptr<PCA9685> pwm_;
43         const int channel_;
44
45         int angle_;
46     };
47 } // namespace car::system::movement::wheels
48
49 #endif
50 #endif // __linux__

```

13.42 common/include/car/system/movement/MovementSystem.h File Reference

```

#include <memory>
#include "car/system/movement/controller/AbstractMovementController.h"

```

Classes

- class [car::system::movement::MovementSystem](#)

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::system::movement](#)

13.43 MovementSystem.h

[Go to the documentation of this file.](#)

```

1 #ifndef MOVEMENTSYSTEM_H
2 #define MOVEMENTSYSTEM_H
3
4 #pragma once
5
6 #include <memory>
7
8 #include "car/system/movement/controller/AbstractMovementController.h"
9
10 using namespace car::system::movement::controller;
11
12 namespace car::system::movement
13 {
14     class MovementSystem

```

```

15     {
16     public:
17         MovementSystem(std::unique_ptr<AbstractMovementController> movement_controller) :
            movement_controller(std::move(movement_controller)) {};
18
19         void initialize()
20         {
21             this->movement_controller->initialize();
22         }
23
24         void start()
25         {
26         }
27
28         void stop()
29         {
30             this->movement_controller->stop();
31         }
32
33         void terminate()
34         {
35             this->movement_controller->terminate();
36         }
37
38 #pragma region Wheels
39         void setRearWheelsSpeed(const int speed) const
40         {
41             this->movement_controller->setRearWheelsSpeed(speed);
42         }
43
44         void setRearLeftWheelSpeed(const int speed) const
45         {
46             this->movement_controller->setRearLeftWheelSpeed(speed);
47         }
48
49         void setRearRightWheelSpeed(const int speed) const
50         {
51             this->movement_controller->setRearRightWheelSpeed(speed);
52         }
53
54         void setFrontWheelsAngle(const float angle) const
55         {
56             this->movement_controller->setFrontWheelsAngle(angle);
57         }
58
59         void setCameraServo1Angle(const float angle) const
60         {
61             this->movement_controller->setCameraServo1Angle(angle);
62         }
63
64         void setCameraServo2Angle(const float angle) const
65         {
66             this->movement_controller->setCameraServo2Angle(angle);
67         }
68
69         void setRearWheelsDirectionToForward() const
70         {
71             this->movement_controller->setRearWheelsDirectionToForward();
72         }
73
74         void setRearLeftWheelDirectionToForward() const
75         {
76             this->movement_controller->setRearLeftWheelDirectionToForward();
77         }
78
79         void setRearRightWheelDirectionToForward() const
80         {
81             this->movement_controller->setRearRightWheelDirectionToForward();
82         }
83
84         void setRearWheelsDirectionToBackward() const
85         {
86             this->movement_controller->setRearWheelsDirectionToBackward();
87         }
88
89         void setRearLeftWheelDirectionToBackward() const
90         {
91             this->movement_controller->setRearLeftWheelDirectionToBackward();
92         }
93
94         void setRearRightWheelDirectionToBackward() const
95         {
96             this->movement_controller->setRearRightWheelDirectionToBackward();
97         }
98 #pragma endregion
99
100     ~MovementSystem() {};

```

```
101
102     private:
103         std::unique_ptr<AbstractMovementController> movement_controller;
104     };
105 };
106
107 #endif
```

13.44 common/src/car/system/CarSystem.cpp File Reference

```
#include "car/system/CarSystem.h"
#include <memory>
#include <rapidjson/rapidjson.h>
#include <rapidjson/document.h>
#include <rapidjson/stringbuffer.h>
#include <rapidjson/writer.h>
#include <tobiaslocker_base64/base64.hpp>
#include "car/configuration/Configuration.h"
#include "car/system/device/DeviceManager.h"
#include "car/system/device/lidar/LidarDevice.h"
#include "car/system/device/CameraDevice.h"
#include "car/system/messaging/MessagingSystem.h"
#include "car/system/movement/MovementSystem.h"
#include "car/plugin/PluginManager.h"
```

Namespaces

- namespace `car`
- namespace `car::system`

13.45 common/src/car/system/device/CameraDevice.cpp File Reference

```
#include "car/system/device/CameraDevice.h"
```

Namespaces

- namespace `car`
- namespace `car::system`
- namespace `car::system::device`

13.46 common/src/car/system/device/DeviceManager.cpp File Reference

```
#include "car/system/device/DeviceManager.h"
#include "car/system/CarSystem.h"
```

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::system::device](#)

13.47 common/src/car/system/messaging/MessagingSystem.cpp File Reference

```
#include "car/system/messaging/MessagingSystem.h"  
#include <functional>  
#include <memory>  
#include <ixwebsocket/IXNetSystem.h>  
#include <ixwebsocket/IXWebSocket.h>  
#include <nod/nod.hpp>  
#include <spdlog/spdlog.h>  
#include <rapidjson/rapidjson.h>  
#include <rapidjson/document.h>  
#include <fmt/format.h>  
#include "car/configuration/Configuration.h"
```

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::system::messaging](#)

13.48 common/src/car/system/movement/controller/DeviceMovement↵ Controller.cpp File Reference

13.49 common/src/car/system/movement/controller/DummyMovement↵ Controller.cpp File Reference

```
#include "car/system/movement/controller/DummyMovementController.h"  
#include <spdlog/spdlog.h>
```

Namespaces

- namespace [car](#)
- namespace [car::system](#)
- namespace [car::system::movement](#)
- namespace [car::system::movement::controller](#)

13.50 common/src/car/system/movement/devices/RearWheel.cpp File Reference

13.51 common/src/car/system/movement/devices/Servo.cpp File Reference

13.52 common/tests/pca9685/test_front_wheels.cpp File Reference

```
#include "PCA9685.h"
#include <iostream>
#include <algorithm>
#include <thread>
```

Functions

- int [setAngle](#) (int &angle, PCA9685 pwm, int channel)
- int [map](#) (int x, int in_min, int in_max, int out_min, int out_max)
- int [setAngleToAnalog](#) (int angle)
- int [main](#) ()

Variables

- int [offset](#) = 0

13.52.1 Function Documentation

13.52.1.1 main()

```
int main ( )
```

13.52.1.2 map()

```
int map (
    int x,
    int in_min,
    int in_max,
    int out_min,
    int out_max )
```

Following method clamps the x to in_min and in_max. Afterwards, it puts the result of that into the range of out_min and out_max

13.52.1.3 `setAngle()`

```
int setAngle (
    int & angle,
    PCA9685 pwm,
    int channel )
```

13.52.1.4 `setAngleToAnalog()`

```
int setAngleToAnalog (
    int angle )
```

13.52.2 Variable Documentation

13.52.2.1 `offset`

```
int offset = 0
```

13.53 common/tests/tb6612/test_rear_wheels.cpp File Reference

```
#include <pigpio.h>
#include <iostream>
#include <memory>
#include <thread>
#include <chrono>
#include <algorithm>
#include "PCA9685.h"
#include "TB6612.h"
```

Classes

- class [BackWheels](#)

Functions

- void [test](#) ()
- int [main](#) ()

13.53.1 Function Documentation

13.53.1.1 main()

```
int main ( )
```

13.53.1.2 test()

```
void test ( )
```

13.54 daemon/install/README.md File Reference

13.55 daemon/README.md File Reference

13.56 README.md File Reference

13.57 tui/README.md File Reference

13.58 repository/packages/t/tb6612/tb6612/include/TB6612.h File Reference

```
#include "pigpio.h"  
#include "pigpiod_if2.h"
```

Classes

- class [TB6612](#)

13.59 TB6612.h

[Go to the documentation of this file.](#)

```
1 #ifndef TB6612_HPP  
2 #define TB6612_HPP  
3  
4 #pragma once  
5  
6 // Made with the help of ChatGPT  
7  
8 #include "pigpio.h"  
9 #include "pigpiod_if2.h"  
10  
11 class TB6612  
12 {  
13 public:  
14     TB6612(int motor_pin, int pwm_pin);  
15  
16     void setPWM(int value);  
17
```

```
18     void forward();
19
20     void backward();
21
22     void stop();
23
24     void setOffset(bool offset);
25
26     const int &getMotorPin() const;
27
28     const int &getPWMPin() const;
29
30 private:
31     const int motor_pin;
32     const int pwm_pin;
33     bool offset = true;
34 };
35
36 #endif
```

13.60 repository/packages/t/tb6612/tb6612/src/TB6612.cpp File Reference

```
#include "TB6612.h"
```

13.61 SETUP.md File Reference

13.62 tui/SETUP.md File Reference

13.63 tui/src/car/configuration/JsonConfiguration.cxx File Reference

```
#include <iostream>
#include <fstream>
#include <variant>
#include <optional>
#include <rapidjson/document.h>
#include <rapidjson/istreamwrapper.h>
#include <spdlog/spdlog.h>
#include <fmt/format.h>
#include <tl/expected.hpp>
#include "car/configuration/Configuration.h"
```

Classes

- class [car::configuration::JsonConfiguration](#)

Namespaces

- namespace [car](#)
- namespace [car::configuration](#)

Macros

- `#define` [JSONCONFIGURATION_CXX](#)

13.63.1 Macro Definition Documentation

13.63.1.1 JSONCONFIGURATION_CXX

```
#define JSONCONFIGURATION_CXX
```

13.64 tui/src/car/display/console/CarConsole.cpp File Reference

```
#include "CarConsole.h"  
#include <ftxui/component/component.hpp>  
#include <ftxui/component/screen_interactive.hpp>  
#include <ftxui/dom/elements.hpp>  
#include <ftxui/component/loop.hpp>  
#include <nod/nod.hpp>
```

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)

13.65 tui/src/car/display/console/CarConsole.h File Reference

```
#include <memory>  
#include "car/system/CarSystem.h"  
#include "car/system/logging/VectorSink.h"  
#include "screen/MainScreen.cxx"  
#include "screen/SettingsScreen.cxx"  
#include "screen/LoggingScreen.cxx"
```

Classes

- class [car::display::console::CarConsole](#)

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)

13.66 CarConsole.h

[Go to the documentation of this file.](#)

```

1 #ifndef CARCONSOLE_H
2 #define CARCONSOLE_H
3
4 #pragma once
5
6 #include <memory>
7
8 #include "car/system/CarSystem.h"
9 #include "car/system/logging/VectorSink.h"
10
11 #include "screen/MainScreen.cxx"
12 #include "screen/SettingsScreen.cxx"
13 #include "screen/LoggingScreen.cxx"
14
15 using namespace car::system;
16 using namespace car::display::console::screen;
17
18 namespace car::display::console
19 {
20     class CarConsole
21     {
22     public:
23         CarConsole(std::shared_ptr<CarSystem> car_system, std::shared_ptr<JsonConfiguration>
                json_configuration, std::shared_ptr<logging::vector_sink_mt> vector_sink);
24
25         void initialize();
26
27         void run();
28
29         void terminate();
30
31     private:
32         std::shared_ptr<CarSystem> car_system;
33         std::shared_ptr<JsonConfiguration> json_configuration;
34         std::shared_ptr<logging::vector_sink_mt> vector_sink;
35     };
36 }
37
38 #endif

```

13.67 tui/src/car/display/console/component/debug/DebugEnabler.cxx

File Reference

```

#include <nod/nod.hpp>
#include <ftxui/component/component.hpp>

```

Classes

- class [car::display::console::component::debug::DebugEnabler](#)

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)
- namespace [car::display::console::component](#)
- namespace [car::display::console::component::debug](#)

Macros

- #define [DEBUGENABLER_CXX](#)

13.67.1 Macro Definition Documentation

13.67.1.1 DEBUGENABLER_CXX

```
#define DEBUGENABLER_CXX
```

13.68 tui/src/car/display/console/component/debug/DebugLidarCheckbox.cxx File Reference

```
#include <nod/nod.hpp>
#include <ftxui/component/component.hpp>
```

Classes

- class [car::display::console::component::debug::DebugLidarCheckbox](#)

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)
- namespace [car::display::console::component](#)
- namespace [car::display::console::component::debug](#)

Macros

- `#define` [DEBUGLIDARCHECKBOX_CXX](#)

13.68.1 Macro Definition Documentation

13.68.1.1 DEBUGLIDARCHECKBOX_CXX

```
#define DEBUGLIDARCHECKBOX_CXX
```

13.69 tui/src/car/display/console/component/debug/DebugMessagingTextbox.cxx File Reference

```
#include <nod/nod.hpp>
#include <ftxui/component/component.hpp>
```

Classes

- class [car::display::console::component::debug::DebugMessagingTextbox](#)

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)
- namespace [car::display::console::component](#)
- namespace [car::display::console::component::debug](#)

Macros

- `#define` [DEBUGMESSAGINGTEXTBOX_CXX](#)

13.69.1 Macro Definition Documentation

13.69.1.1 DEBUGMESSAGINGTEXTBOX_CXX

```
#define DEBUGMESSAGINGTEXTBOX_CXX
```

13.70 tui/src/car/display/console/component/debug/DebugMovementRenderer.cxx File Reference

```
#include <nod/nod.hpp>
#include <ftxui/component/component.hpp>
```

Classes

- class [car::display::console::component::debug::DebugMovementRenderer](#)

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)
- namespace [car::display::console::component](#)
- namespace [car::display::console::component::debug](#)

Macros

- `#define` [DEBUGMOVEMENTRENDERER_CXX](#)

13.70.1 Macro Definition Documentation

13.70.1.1 DEBUGMOVEMENTRENDERER_CXX

```
#define DEBUGMOVEMENTRENDERER_CXX
```

13.71 tui/src/car/display/console/component/main/ConnectButton.cxx File Reference

```
#include <ftxui/component/component.hpp>  
#include "car/system/CarSystem.h"
```

Classes

- class [car::display::console::component::main::ConnectButton](#)

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)
- namespace [car::display::console::component](#)
- namespace [car::display::console::component::main](#)

Macros

- `#define` [CONNECTBUTTON_CXX](#)

13.71.1 Macro Definition Documentation

13.71.1.1 CONNECTBUTTON_CXX

```
#define CONNECTBUTTON_CXX
```

13.72 tui/src/car/display/console/component/main/MainErrorModal.cxx File Reference

```
#include <ftxui/component/component.hpp>  
#include "car/system/CarSystem.h"
```

Classes

- class [car::display::console::component::main::MainErrorModal](#)

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)
- namespace [car::display::console::component](#)
- namespace [car::display::console::component::main](#)

Macros

- #define [MAINERRORMODAL_CXX](#)

13.72.1 Macro Definition Documentation

13.72.1.1 MAINERRORMODAL_CXX

```
#define MAINERRORMODAL_CXX
```

13.73 `tui/src/car/display/console/component/main/MainExitModal.cxx` File Reference

```
#include <ftxui/component/component.hpp>
#include "car/system/CarSystem.h"
```

Classes

- class `car::display::console::component::main::MainExitModal`

Namespaces

- namespace `car`
- namespace `car::display`
- namespace `car::display::console`
- namespace `car::display::console::component`
- namespace `car::display::console::component::main`

Macros

- `#define MAINEXITMODAL_CXX`

13.73.1 Macro Definition Documentation

13.73.1.1 `MAINEXITMODAL_CXX`

```
#define MAINEXITMODAL_CXX
```

13.74 `tui/src/car/display/console/component/settings/SettingsEditConfig.cxx` File Reference

```
#include <ftxui/component/component.hpp>
#include "car/system/CarSystem.h"
#include "../../../../../configuration/JsonConfiguration.cxx"
```

Classes

- class `car::display::console::component::settings::SettingsEditConfig`

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)
- namespace [car::display::console::component](#)
- namespace [car::display::console::component::settings](#)

Macros

- `#define` [SETTINGSEDTCONFIG_CXX](#)

13.74.1 Macro Definition Documentation

13.74.1.1 SETTINGSEDTCONFIG_CXX

```
#define SETTINGSEDTCONFIG_CXX
```

13.75 tui/src/car/display/console/screen/LoggingScreen.cxx File Reference

```
#include <ftxui/component/component.hpp>  
#include <spdlog/spdlog.h>  
#include "car/system/logging/VectorSink.h"
```

Classes

- class [car::display::console::screen::LoggingScreen](#)

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)
- namespace [car::display::console::screen](#)

Macros

- `#define` [LOGGINGSCREEN_CXX](#)

13.75.1 Macro Definition Documentation

13.75.1.1 LOGGINGSCREEN_CXX

```
#define LOGGINGSCREEN_CXX
```

13.76 tui/src/car/display/console/screen/MainScreen.cxx File Reference

```
#include <memory>
#include <ftxui/component/component.hpp>
#include "car/system/CarSystem.h"
#include "../component/main/ConnectButton.cxx"
#include "../component/main/MainExitModal.cxx"
#include "../component/main/MainErrorModal.cxx"
```

Classes

- class [car::display::console::screen::MainScreen](#)

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)
- namespace [car::display::console::screen](#)

Macros

- #define [MAINSCREEN_CXX](#)

13.76.1 Macro Definition Documentation

13.76.1.1 MAINSCREEN_CXX

```
#define MAINSCREEN_CXX
```

13.77 tui/src/car/display/console/screen/SettingsScreen.cxx File Reference

```
#include <memory>
#include <ftxui/component/component.hpp>
#include "car/system/CarSystem.h"
#include "../../configuration/JsonConfiguration.cxx"
#include "../component/settings/SettingsEditConfig.cxx"
#include "../component/debug/DebugEnabler.cxx"
#include "../component/debug/DebugLidarCheckbox.cxx"
#include "../component/debug/DebugMovementRenderer.cxx"
#include "../component/debug/DebugMessagingTextbox.cxx"
```

Classes

- class [car::display::console::screen::SettingsScreen](#)

Namespaces

- namespace [car](#)
- namespace [car::display](#)
- namespace [car::display::console](#)
- namespace [car::display::console::screen](#)

Macros

- #define [SETTINGSSCREEN_CXX](#)

13.77.1 Macro Definition Documentation

13.77.1.1 SETTINGSSCREEN_CXX

```
#define SETTINGSSCREEN_CXX
```


Index

- - behaviour_tree::CarContext, [41](#)
- _setBehaviourTree
 - behaviour_tree::BehaviourTreeHandler, [33](#)
- ~CameraDevice
 - car::system::device::CameraDevice, [36](#)
- ~MovementSystem
 - car::system::movement::MovementSystem, [94](#)
- addPlugin
 - car::plugin::PluginManager, [99](#)
- any_configuration_empty
 - rpi_daemon, [101](#)
- attempted_to_reconnect
 - rpi_daemon, [101](#)
- backward
 - BackWheels, [29](#)
 - TB6612, [107](#)
- BackWheels, [29](#)
 - backward, [29](#)
 - BackWheels, [29](#)
 - cali_forward_A, [31](#)
 - cali_forward_B, [31](#)
 - calibration, [29](#)
 - caliLeft, [30](#)
 - caliOK, [30](#)
 - caliRight, [30](#)
 - forward, [30](#)
 - forward_A, [31](#)
 - forward_B, [31](#)
 - getSpeed, [30](#)
 - left_wheel, [31](#)
 - pca9685, [31](#)
 - ready, [30](#)
 - right_wheel, [31](#)
 - setSpeed, [30](#)
 - speed, [32](#)
 - stop, [30](#)
- behaviour_tree, [21](#)
 - behaviour_tree::BehaviourTreeHandler, [34](#)
- behaviour_tree/src/main.cpp, [111](#)
- behaviour_tree::BehaviourTreeHandler, [32](#)
 - _setBehaviourTree, [33](#)
 - behaviour_tree, [34](#)
 - car_system, [34](#)
 - context, [34](#)
 - getName, [33](#)
 - handleCommand, [33](#)
 - initialize, [33](#)
 - last_connected, [35](#)
 - setBehaviourTree, [33](#)
 - startBehaviourTree, [33](#)
 - stop, [34](#)
 - stopBehaviourTree, [34](#)
 - tick_count, [35](#)
 - update, [34](#)
- behaviour_tree::CarContext, [41](#)
 - _, [41](#)
 - car_system, [42](#)
 - CarContext, [41](#)
 - getCarSystem, [42](#)
- behaviour_tree_update_ms_interval
 - car::configuration::Configuration, [47](#)
- Both
 - StreamType.h, [131](#)
- box
 - car::display::console::screen::MainScreen, [86](#)
- button_pressed
 - car::display::console::component::main::ConnectButton, [49](#)
- cali_forward_A
 - BackWheels, [31](#)
- cali_forward_B
 - BackWheels, [31](#)
- calibration
 - BackWheels, [29](#)
- caliLeft
 - BackWheels, [30](#)
- caliOK
 - BackWheels, [30](#)
- caliRight
 - BackWheels, [30](#)
- Camera
 - StreamType.h, [131](#)
- camera_
 - car::system::device::CameraDevice, [38](#)
- camera_device_
 - car::system::device::DeviceManager, [66](#)
- camera_fps
 - car::configuration::Configuration, [47](#)
- camera_fps_interval
 - car::configuration::Configuration, [48](#)
- camera_index
 - car::configuration::Configuration, [48](#)
- camera_mutex_
 - car::system::device::CameraDevice, [38](#)
- camera_servo_1_angle_slider

- car::display::console::component::debug::DebugMovementRenderer, 60
- camera_servo_1_angle_slider_angle
 - car::display::console::component::debug::DebugMovementRenderer, 60
- camera_servo_2_angle_slider
 - car::display::console::component::debug::DebugMovementRenderer, 60
- camera_servo_2_angle_slider_angle
 - car::display::console::component::debug::DebugMovementRenderer, 60
- CameraDevice
 - car::system::device::CameraDevice, 36
- car, 21
- car::configuration, 21
- car::configuration::Configuration, 46
 - behaviour_tree_update_ms_interval, 47
 - camera_fps, 47
 - camera_fps_interval, 48
 - camera_index, 48
 - getCameraFpsInterval, 47
 - host, 48
 - lidar_port, 48
 - setCameraFps, 47
 - use_camera, 48
 - use_lidar, 48
- car::configuration::JsonConfiguration, 72
 - config_file_path, 73
 - exe_dir, 73
 - getConfigFilePath, 72
 - JsonConfiguration, 72
 - loadConfiguration, 72
 - setConfigFilePath, 72
- car::display, 21
- car::display::console, 22
- car::display::console::CarConsole, 39
 - car_system, 40
 - CarConsole, 39
 - initialize, 40
 - json_configuration, 40
 - run, 40
 - terminate, 40
 - vector_sink, 40
- car::display::console::component, 22
- car::display::console::component::debug, 22
- car::display::console::component::debug::DebugEnabler, 50
 - checkbox_value, 51
 - component, 51
 - debounce, 52
 - DEBUG_ENABLE_WARNING_MESSAGE, 52
 - DEBUG_MODE_DISABLED_MESSAGE, 52
 - DEBUG_MODE_ENABLED_MESSAGE, 52
 - DEBUG_MODE_WAIT_MESSAGE, 52
 - display_warn_debug_modal, 52
 - enabled, 52
 - getCheckbox, 51
 - getWarningModal, 51
- car::display::console::component::debug::DebugMovementRenderer, 51
 - is_enabled, 51
 - status, 53
- car::display::console::component::debug::DebugLidarCheckbox, 53
 - DebugLidarCheckbox, 53
 - element, 54
 - lidar_motor_signal, 54
 - lidar_motor_checkbox_component, 54
 - LIDAR_MOTOR_DISABLED_MESSAGE, 54
 - lidar_motor_enabled, 54
 - LIDAR_MOTOR_ENABLED_MESSAGE, 54
 - lidar_motor_loading_debounce, 54
 - lidar_motor_signal, 55
 - lidar_motor_status, 55
- car::display::console::component::debug::DebugMessagingTextbox, 55
 - DebugMessagingTextbox, 55
 - element, 56
 - message, 56
 - message_signal, 56
 - messaging_container, 56
 - messaging_textbox, 56
 - messaging_title, 56
- car::display::console::component::debug::DebugMovementRenderer, 57
 - camera_servo_1_angle_slider, 60
 - camera_servo_1_angle_slider_angle, 60
 - camera_servo_2_angle_slider, 60
 - camera_servo_2_angle_slider_angle, 60
 - DebugMovementRenderer, 58
 - DEFAULT_FRONT_WHEEL_ANGLE, 60
 - DEFAULT_REAR_WHEEL_SPEED, 60
 - element, 58
 - front_wheels_angle_slider, 61
 - front_wheels_angle_slider_value, 61
 - getCameraServo1AngleSliderValue, 58
 - getCameraServo2AngleSliderValue, 58
 - getFrontWheelsAngleSliderValue, 58
 - getRearLeftWheelSpeedSliderValue, 59
 - getRearRightWheelSpeedSliderValue, 59
 - getRearWheelDirectionSignal, 59
 - previous_camera_servo_1_angle_slider_angle, 61
 - previous_camera_servo_2_angle_slider_angle, 61
 - previous_front_wheels_angle_slider_value, 61
 - previous_rear_left_wheel_speed_slider_value, 61
 - previous_rear_right_wheel_speed_slider_value, 61
 - previous_rear_wheels_speed_slider_value, 62
 - rear_left_wheel_speed_slider, 62
 - rear_left_wheel_speed_slider_value, 62
 - rear_right_wheel_speed_slider, 62
 - rear_right_wheel_speed_slider_value, 62
 - rear_wheel_direction, 62
 - REAR_WHEEL_DIRECTION_BACKWARD_MESSAGE, 62
 - rear_wheel_direction_checkbox_component, 63
 - rear_wheel_direction_debounce, 63

- REAR_WHEEL_DIRECTION_FORWARD_MESSAGE, 63
- rear_wheel_direction_signal, 63
- rear_wheel_direction_status, 63
- rear_wheel_menu_entry, 63
- rear_wheel_speed_slider, 63
- rear_wheels_speed_slider_value, 64
- servo_menu_entry, 64
- slider_container, 64
- updateCameraServo1, 59
- updateCameraServo2, 59
- updateFrontWheels, 59
- updateRearWheels, 59
- car::display::console::component::main, 22
- car::display::console::component::main::ConnectButton, 49
 - button_pressed, 49
 - car_system, 50
 - ConnectButton, 49
 - element, 49
 - main_button, 50
 - main_button_text, 50
 - main_debounce, 50
 - on_connect_failure, 50
- car::display::console::component::main::MainErrorModal, 83
 - element, 83
 - error_element, 83
 - error_modal_shown, 84
 - main_error_modal, 84
 - MainErrorModal, 83
 - setErrorMessage, 83
- car::display::console::component::main::MainExitModal, 84
 - element, 85
 - exit, 85
 - exit_modal_shown, 85
 - main_exit_modal, 85
 - MainExitModal, 84
- car::display::console::component::settings, 22
- car::display::console::component::settings::SettingsEditConfig, 102
 - car_system, 103
 - element, 103
 - input_settings_file_path, 103
 - json_configuration, 103
 - load_button, 103
 - placeholder, 103
 - settings_file_path, 104
 - SettingsEditConfig, 102
- car::display::console::screen, 23
- car::display::console::screen::LoggingScreen, 81
 - element, 81
 - line_elements, 82
 - LoggingScreen, 81
 - menu, 82
 - my_custom_menu, 82
 - selected_line, 82
 - vector_sink, 82
- car::display::console::screen::MainScreen, 85
 - box, 86
 - car_system, 86
 - connect_button, 86
 - element, 86
 - info, 87
 - main_component, 87
 - main_error_modal, 87
 - main_exit_modal, 87
 - main_screen, 87
 - MainScreen, 86
- car::display::console::screen::SettingsScreen, 104
 - car_system, 105
 - debug_enabler, 105
 - debug_lidar_checkbox, 105
 - debug_messaging_text_box, 105
 - debug_movement_renderer, 105
 - element, 105
 - settings_edit_config, 106
 - SettingsScreen, 104
 - update, 105
- car::plugin, 23
- car::plugin::Plugin, 97
 - getName, 97
 - initialize, 97
 - stop, 98
 - update, 98
- car::plugin::PluginManager, 98
 - addPlugin, 99
 - getPlugin, 99
 - initialize, 99
 - plugins, 99
 - stop, 99
 - terminate, 99
 - update, 99
- car::system, 23
- car::system::CarSystem, 42
 - CarSystem, 43
 - configuration_, 45
 - device_manager_, 45
 - disconnect, 43
 - getConfiguration, 43
 - getDeviceManager, 43
 - getMessagingSystem, 44
 - getMovementSystem, 44
 - getPlugin, 44
 - initialize, 44
 - initialized, 46
 - messaging_system_, 46
 - movement_system_, 46
 - plugin_manager_, 46
 - reload, 44
 - sendData, 44
 - setConfiguration, 44
 - start, 45
 - started, 46
 - stop, 45

- terminate, 45
 - tryConnect, 45
 - update, 45
- car::system::device, 23
- car::system::device::CameraDevice, 35
 - ~CameraDevice, 36
 - camera_, 38
 - camera_mutex_, 38
 - CameraDevice, 36
 - configuration, 38
 - connected_, 38
 - create, 36
 - DeviceManager, 38
 - disconnect, 37
 - frame_buffer_, 39
 - getFrameBuffer, 37
 - last, 39
 - operator=, 37
 - start, 37
 - stop, 37
 - terminate, 37
 - update, 38
- car::system::device::DeviceManager, 64
 - camera_device_, 66
 - car_system, 66
 - create, 65
 - DeviceManager, 65
 - getCameraDevice, 65
 - getLidarDevice, 65
 - initialize, 65
 - is_initialized_, 67
 - is_running_, 67
 - isRunning, 66
 - lidar_device_, 67
 - start, 66
 - stop, 66
 - terminate, 66
 - update, 66
- car::system::device::lidar, 23
- car::system::device::lidar::LidarDevice, 73
 - DeviceManager, 75
 - disconnect, 74
 - getScanData, 74
 - initialize, 74
 - scan_data_, 75
 - setScanData, 74
 - start, 74
 - stop, 74
 - terminate, 75
 - update, 75
- car::system::device::lidar::LidarDummy, 76
 - disconnect, 76
 - initialize, 76
 - LidarDummy, 76
 - start, 77
 - stop, 77
 - terminate, 77
 - update, 77
- car::system::device::lidar::LidarScanner, 78
 - configuration_, 80
 - create, 79
 - disconnect, 79
 - initialize, 79
 - lidar_, 80
 - LidarScanner, 78
 - running, 80
 - scan_data_, 80
 - scan_data_mutex_, 80
 - scan_generator_, 81
 - start, 79
 - stop, 79
 - terminate, 79
 - update, 80
- car::system::logging, 24
 - vector_sink_mt, 24
- car::system::logging::VectorSink< Mutex >, 108
 - flush_, 109
 - get_log_messages, 109
 - log_messages, 109
 - max_lines, 110
 - sink_it_, 109
 - VectorSink, 109
- car::system::messaging, 24
- car::system::messaging::MessagingSystem, 87
 - command_signal_, 92
 - configuration_, 92
 - connected_, 92
 - getCommandSignal, 89
 - getDisconnectSignal, 89
 - getFirstMessage, 89
 - getMessageSignal, 89
 - getSelectionSignal, 89
 - getUUID, 89
 - handleMessage, 90
 - initialize, 90
 - initializeWebSocket, 90
 - isConnected, 90
 - message_signal_, 92
 - MessagingSystem, 88
 - on_disconnect_signal_, 92
 - onDisconnect, 90
 - onFirstMessage, 91
 - onMessageCallback, 91
 - selection_signal_, 93
 - sendMessage, 91
 - setConfiguration, 91
 - stop, 91
 - terminate, 91
 - tryConnect, 92
 - uuid_, 93
 - websocket_, 93
 - websocket_url_, 93
- car::system::messaging::MessagingSystem::FirstMessageStruct, 71
 - condition, 71
 - error_message, 71

- uuid, 71
- car::system::movement, 24
- car::system::movement::controller, 24
- car::system::movement::controller::AbstractMovementController, 25
 - initialize, 25
 - setCameraServo1Angle, 26
 - setCameraServo2Angle, 26
 - setFrontWheelsAngle, 26
 - setRearLeftWheelDirectionToBackward, 26
 - setRearLeftWheelDirectionToForward, 26
 - setRearLeftWheelSpeed, 27
 - setRearRightWheelDirectionToBackward, 27
 - setRearRightWheelDirectionToForward, 27
 - setRearRightWheelSpeed, 27
 - setRearWheelsDirectionToBackward, 27
 - setRearWheelsDirectionToForward, 28
 - setRearWheelsSpeed, 28
 - stop, 28
 - terminate, 28
- car::system::movement::controller::DummyMovementController, 67
 - initialize, 68
 - setCameraServo1Angle, 68
 - setCameraServo2Angle, 68
 - setFrontWheelsAngle, 68
 - setRearLeftWheelDirectionToBackward, 69
 - setRearLeftWheelDirectionToForward, 69
 - setRearLeftWheelSpeed, 69
 - setRearRightWheelDirectionToBackward, 69
 - setRearRightWheelDirectionToForward, 69
 - setRearRightWheelSpeed, 69
 - setRearWheelsDirectionToBackward, 70
 - setRearWheelsDirectionToForward, 70
 - setRearWheelsSpeed, 70
 - stop, 70
 - terminate, 70
- car::system::movement::MovementSystem, 93
 - ~MovementSystem, 94
 - initialize, 94
 - movement_controller, 97
 - MovementSystem, 94
 - setCameraServo1Angle, 94
 - setCameraServo2Angle, 95
 - setFrontWheelsAngle, 95
 - setRearLeftWheelDirectionToBackward, 95
 - setRearLeftWheelDirectionToForward, 95
 - setRearLeftWheelSpeed, 95
 - setRearRightWheelDirectionToBackward, 95
 - setRearRightWheelDirectionToForward, 95
 - setRearRightWheelSpeed, 96
 - setRearWheelsDirectionToBackward, 96
 - setRearWheelsDirectionToForward, 96
 - setRearWheelsSpeed, 96
 - start, 96
 - stop, 96
 - terminate, 96
- car_system
 - behaviour_tree::BehaviourTreeHandler, 34
 - behaviour_tree::CarContext, 42
 - car::display::console::CarConsole, 40
 - car::display::console::component::main::ConnectButton, 50
 - car::display::console::component::settings::SettingsEditConfig, 103
 - car::display::console::screen::MainScreen, 86
 - car::display::console::screen::SettingsScreen, 105
 - car::system::device::DeviceManager, 66
 - rpi_daemon, 101
 - CarConsole
 - car::display::console::CarConsole, 39
 - CarContext
 - behaviour_tree::CarContext, 41
 - CarSystem
 - car::system::CarSystem, 43
 - checkbox_value
 - car::display::console::component::debug::DebugEnabler, 51
 - color_and_signal_
 - car::system::messaging::MessagingSystem, 92
 - common/include/behaviour_tree/BehaviourTreeHandler.hpp, 114, 115
 - common/include/behaviour_tree/CarContext.hpp, 117
 - common/include/car/configuration/Configuration.h, 118
 - common/include/car/plugin/Plugin.h, 118, 119
 - common/include/car/plugin/PluginManager.h, 119, 120
 - common/include/car/system/CarSystem.h, 121
 - common/include/car/system/device/CameraDevice.h, 122, 123
 - common/include/car/system/device/DeviceManager.h, 123, 124
 - common/include/car/system/device/lidar/LidarDevice.h, 125
 - common/include/car/system/device/lidar/LidarDummy.h, 126
 - common/include/car/system/device/lidar/LidarScanner.h, 126, 127
 - common/include/car/system/logging/VectorSink.h, 128, 129
 - common/include/car/system/messaging/MessagingSystem.h, 129, 130
 - common/include/car/system/messaging/StreamType.h, 131, 132
 - common/include/car/system/movement/controller/AbstractMovementController.h, 132
 - common/include/car/system/movement/controller/DeviceMovementController.h, 133
 - common/include/car/system/movement/controller/DummyMovementController.h, 134
 - common/include/car/system/movement/devices/RearWheel.h, 135
 - common/include/car/system/movement/devices/Servo.h, 135
 - common/include/car/system/movement/MovementSystem.h, 136
 - common/src/car/system/CarSystem.cpp, 138

- common/src/car/system/device/CameraDevice.cpp, 138
- common/src/car/system/device/DeviceManager.cpp, 138
- common/src/car/system/messaging/MessagingSystem.cpp, 139
- common/src/car/system/movement/controller/DeviceMovementController.cpp, 139
- common/src/car/system/movement/controller/DummyMovementController.cpp, 139
- common/src/car/system/movement/devices/RearWheel.cpp, 140
- common/src/car/system/movement/devices/Servo.cpp, 140
- common/tests/pca9685/test_front_wheels.cpp, 140
- common/tests/tb6612/test_rear_wheels.cpp, 141
- component
 - car::display::console::component::debug::DebugEnabler, 51
- condition
 - car::system::messaging::MessagingSystem::FirstMessageFinal, 71
- config_file_path
 - car::configuration::JsonConfiguration, 73
- configuration
 - car::system::device::CameraDevice, 38
- configuration_
 - car::system::CarSystem, 45
 - car::system::device::lidar::LidarScanner, 80
 - car::system::messaging::MessagingSystem, 92
- connect
 - rpi_daemon, 100
- connect_button
 - car::display::console::screen::MainScreen, 86
- ConnectButton
 - car::display::console::component::main::ConnectButton, 49
- ConnectButton.cxx
 - CONNECTBUTTON_CXX, 149
- CONNECTBUTTON_CXX
 - ConnectButton.cxx, 149
- connected_
 - car::system::device::CameraDevice, 38
 - car::system::messaging::MessagingSystem, 92
- connection_ms_interval
 - rpi_daemon, 102
- context
 - behaviour_tree::BehaviourTreeHandler, 34
- create
 - car::system::device::CameraDevice, 36
 - car::system::device::DeviceManager, 65
 - car::system::device::lidar::LidarScanner, 79
- daemon/install/README.md, 142
- daemon/README.md, 142
- daemon/src/main.cpp, 112
- debounce
 - car::display::console::component::debug::DebugEnabler, 52
- DEBUG_ENABLE_WARNING_MESSAGE
 - car::display::console::component::debug::DebugEnabler, 52
 - debug_enabler
 - car::display::console::screen::SettingsScreen, 105
 - debug_lidar_checkbox
 - car::display::console::screen::SettingsScreen, 105
 - debug_messaging_text_box
 - car::display::console::screen::SettingsScreen, 105
 - DEBUG_MODE_DISABLED_MESSAGE
 - car::display::console::component::debug::DebugEnabler, 52
 - DEBUG_MODE_ENABLED_MESSAGE
 - car::display::console::component::debug::DebugEnabler, 52
 - DEBUG_MODE_WAIT_MESSAGE
 - car::display::console::component::debug::DebugEnabler, 52
 - debug_movement_renderer
 - car::display::console::screen::SettingsScreen, 105
 - DebugEnabler.cxx
 - DEBUGENABLER_CXX, 146
 - DEBUGENABLER_CXX
 - DebugEnabler.cxx, 146
 - DebugLidarCheckbox
 - car::display::console::component::debug::DebugLidarCheckbox, 53
 - DebugLidarCheckbox.cxx
 - DEBUGLIDARCHECKBOX_CXX, 146
 - DEBUGLIDARCHECKBOX_CXX
 - DebugLidarCheckbox.cxx, 146
 - DebugMessagingTextbox
 - car::display::console::component::debug::DebugMessagingTextbox, 55
 - DebugMessagingTextbox.cxx
 - DEBUGMESSAGINGTEXTBOX_CXX, 147
 - DEBUGMESSAGINGTEXTBOX_CXX
 - DebugMessagingTextbox.cxx, 147
 - DebugMovementRenderer
 - car::display::console::component::debug::DebugMovementRenderer, 58
 - DebugMovementRenderer.cxx
 - DEBUGMOVEMENTRENDERER_CXX, 148
 - DEBUGMOVEMENTRENDERER_CXX
 - DebugMovementRenderer.cxx, 148
 - DEFAULT_FRONT_WHEEL_ANGLE
 - car::display::console::component::debug::DebugMovementRenderer, 60
 - DEFAULT_REAR_WHEEL_SPEED
 - car::display::console::component::debug::DebugMovementRenderer, 60
 - device_manager_
 - car::system::CarSystem, 45
 - DeviceManager
 - car::system::device::CameraDevice, 38
 - car::system::device::DeviceManager, 65
 - car::system::device::lidar::LidarDevice, 75
 - disconnect
 - car::system::CarSystem, 43

car::system::device::CameraDevice, 37
 car::system::device::lidar::LidarDevice, 74
 car::system::device::lidar::LidarDummy, 76
 car::system::device::lidar::LidarScanner, 79
 display_warn_debug_modal
 car::display::console::component::debug::DebugEnabler, 52
 element
 car::display::console::component::debug::DebugLidarCheckbox, 54
 car::display::console::component::debug::DebugMessagingSystem, 56
 car::display::console::component::debug::DebugMovementRenderer, 58
 car::display::console::component::main::ConnectButton, 49
 car::display::console::component::main::MainErrorModal, 83
 car::display::console::component::main::MainExitModal, 85
 car::display::console::component::settings::SettingsEditControl, 103
 car::display::console::screen::LoggingScreen, 81
 car::display::console::screen::MainScreen, 86
 car::display::console::screen::SettingsScreen, 105
 enabled
 car::display::console::component::debug::DebugEnabler, 52
 error_element
 car::display::console::component::main::MainErrorModal, 83
 error_message
 car::system::messaging::MessagingSystem::FirstMessageSignal, 71
 error_modal_shown
 car::display::console::component::main::MainErrorModal, 84
 exe_dir
 car::configuration::JsonConfiguration, 73
 exit
 car::display::console::component::main::MainExitModal, 85
 exit_modal_shown
 car::display::console::component::main::MainExitModal, 85
 flush_
 car::system::logging::VectorSink< Mutex >, 109
 forward
 BackWheels, 30
 TB6612, 107
 forward_A
 BackWheels, 31
 forward_B
 BackWheels, 31
 frame_buffer
 car::system::device::CameraDevice, 39
 front_wheels_angle_slider
 car::display::console::component::debug::DebugMovementRenderer, 61
 front_wheels_angle_slider_value
 car::display::console::component::debug::DebugMovementRenderer, 61
 get_log_messages
 car::system::logging::VectorSink< Mutex >, 109
 getCameraDevice
 car::system::device::DeviceManager, 65
 getCameraFpsInterval
 car::configuration::Configuration, 47
 getCameraServo1AngleSliderValue
 car::display::console::component::debug::DebugMovementRenderer, 58
 getCameraServo2AngleSliderValue
 car::display::console::component::debug::DebugMovementRenderer, 58
 getCarSystem
 behaviour_tree::CarContext, 42
 getCheckbox
 car::display::console::component::debug::DebugEnabler, 51
 getCommandSignal
 car::system::messaging::MessagingSystem, 89
 getConfigFilePath
 car::configuration::JsonConfiguration, 72
 getConfiguration
 car::system::CarSystem, 43
 getDeviceManager
 car::system::CarSystem, 43
 getDisconnectSignal
 car::system::messaging::MessagingSystem, 89
 getFirstMessage
 car::system::messaging::MessagingSystem, 89
 getFrameBuffer
 car::system::device::CameraDevice, 37
 getFrontWheelsAngleSliderValue
 car::display::console::component::debug::DebugMovementRenderer, 58
 getLidarDevice
 car::system::device::DeviceManager, 65
 main.cpp, 112, 113
 getLidarMotorSignal
 car::display::console::component::debug::DebugLidarCheckbox, 54
 getMessageSignal
 car::system::messaging::MessagingSystem, 89
 getMessagingSystem
 car::system::CarSystem, 44
 getMotorPin
 TB6612, 107
 getMovementController
 main.cpp, 112, 114
 getMovementSystem
 car::system::CarSystem, 44
 getName
 behaviour_tree::BehaviourTreeHandler, 33
 car::plugin::Plugin, 97

- getPlugin
 - car::plugin::PluginManager, 99
 - car::system::CarSystem, 44
- getPWMPin
 - TB6612, 107
- getRearLeftWheelSpeedSliderValue
 - car::display::console::component::debug::DebugMovementController, 59
- getRearRightWheelSpeedSliderValue
 - car::display::console::component::debug::DebugMovementController, 59
- getRearWheelDirectionSignal
 - car::display::console::component::debug::DebugMovementController, 59
- getScanData
 - car::system::device::lidar::LidarDevice, 74
- getSelectionSignal
 - car::system::messaging::MessagingSystem, 89
- getSpeed
 - BackWheels, 30
- getUUID
 - car::system::messaging::MessagingSystem, 89
- getWarningModal
 - car::display::console::component::debug::DebugEnabler, 51
- handleCommand
 - behaviour_tree::BehaviourTreeHandler, 33
- handleMessage
 - car::system::messaging::MessagingSystem, 90
- host
 - car::configuration::Configuration, 48
- info
 - car::display::console::screen::MainScreen, 87
- initialize
 - behaviour_tree::BehaviourTreeHandler, 33
 - car::display::console::CarConsole, 40
 - car::plugin::Plugin, 97
 - car::plugin::PluginManager, 99
 - car::system::CarSystem, 44
 - car::system::device::DeviceManager, 65
 - car::system::device::lidar::LidarDevice, 74
 - car::system::device::lidar::LidarDummy, 76
 - car::system::device::lidar::LidarScanner, 79
 - car::system::messaging::MessagingSystem, 90
 - car::system::movement::controller::AbstractMovementController, 25
 - car::system::movement::controller::DummyMovementController, 68
 - car::system::movement::MovementSystem, 94
- initialized
 - car::system::CarSystem, 46
- initializeWebSocket
 - car::system::messaging::MessagingSystem, 90
- input_settings_file_path
 - car::display::console::component::settings::SettingsEditConfig, 103
- is_initialized_
 - car::system::device::DeviceManager, 67
- is_running_
 - car::system::device::DeviceManager, 67
- isConnected
 - car::system::messaging::MessagingSystem, 90
- isEnabled
 - car::display::console::component::debug::DebugEnabler, 51
- isRunning
 - car::system::device::DeviceManager, 66
- json_configuration
 - car::display::console::CarConsole, 40
 - car::display::console::component::settings::SettingsEditConfig, 103
- JsonConfiguration
 - car::configuration::JsonConfiguration, 72
- JsonConfiguration.cxx
 - JSONCONFIGURATION_CXX, 144
- JSONCONFIGURATION_CXX
 - JsonConfiguration.cxx, 144
- kbhit
 - main.cpp, 111
- last
 - car::system::device::CameraDevice, 39
- last_connected
 - behaviour_tree::BehaviourTreeHandler, 35
 - rpi_daemon, 102
- left_wheel
 - BackWheels, 31
- Lidar
 - StreamType.h, 131
- lidar_
 - car::system::device::lidar::LidarScanner, 80
- lidar_device_
 - car::system::device::DeviceManager, 67
- lidar_motor_checkbox_component
 - car::display::console::component::debug::DebugLidarCheckbox, 54
- LIDAR_MOTOR_DISABLED_MESSAGE
 - car::display::console::component::debug::DebugLidarCheckbox, 54
- lidar_motor_enabled
 - car::display::console::component::debug::DebugLidarCheckbox, 54
- LIDAR_MOTOR_ENABLED_MESSAGE
 - car::display::console::component::debug::DebugLidarCheckbox, 54
- lidar_motor_loading_debounce
 - car::display::console::component::debug::DebugLidarCheckbox, 54
- lidar_motor_signal
 - car::display::console::component::debug::DebugLidarCheckbox, 55
- lidar_motor_status
 - car::display::console::component::debug::DebugLidarCheckbox, 55

- lidar_port
 - car::configuration::Configuration, 48
- LidarDummy
 - car::system::device::lidar::LidarDummy, 76
- LidarScanner
 - car::system::device::lidar::LidarScanner, 78
- line_elements
 - car::display::console::screen::LoggingScreen, 82
- load_button
 - car::display::console::component::settings::SettingsEditControl, 103
- loadConfiguration
 - car::configuration::JsonConfiguration, 72
- log_messages
 - car::system::logging::VectorSink< Mutex >, 109
- LoggingScreen
 - car::display::console::screen::LoggingScreen, 81
- LoggingScreen.cxx
 - LOGGINGSCREEN_CXX, 152
- LOGGINGSCREEN_CXX
 - LoggingScreen.cxx, 152
- main
 - main.cpp, 111, 113, 114
 - test_front_wheels.cpp, 140
 - test_rear_wheels.cpp, 141
- main.cpp
 - getLidarDevice, 112, 113
 - getMovementController, 112, 114
 - kbhit, 111
 - main, 111, 113, 114
 - terminate_handler, 113
- main_button
 - car::display::console::component::main::ConnectButton, 50
- main_button_text
 - car::display::console::component::main::ConnectButton, 50
- main_component
 - car::display::console::screen::MainScreen, 87
- main_debounce
 - car::display::console::component::main::ConnectButton, 50
- main_error_modal
 - car::display::console::component::main::MainErrorModal, 84
 - car::display::console::screen::MainScreen, 87
- main_exit_modal
 - car::display::console::component::main::MainExitModal, 85
 - car::display::console::screen::MainScreen, 87
- main_screen
 - car::display::console::screen::MainScreen, 87
- MainErrorModal
 - car::display::console::component::main::MainErrorModal, 83
- MainErrorModal.cxx
 - MAINERRORMODAL_CXX, 149
- MAINERRORMODAL_CXX
 - MainErrorModal.cxx, 149
- MainExitModal
 - car::display::console::component::main::MainExitModal, 84
- MainExitModal.cxx
 - MAINEXITMODAL_CXX, 150
- MAINEXITMODAL_CXX
 - MainExitModal.cxx, 150
- MainScreen
 - car::display::console::screen::MainScreen, 86
- MainScreen.cxx
 - MAINSCREEN_CXX, 152
- MAINSCREEN_CXX
 - MainScreen.cxx, 152
- map
 - test_front_wheels.cpp, 140
- max_lines
 - car::system::logging::VectorSink< Mutex >, 110
- menu
 - car::display::console::screen::LoggingScreen, 82
- message
 - car::display::console::component::debug::DebugMessagingTextbox, 56
- message_signal
 - car::display::console::component::debug::DebugMessagingTextbox, 56
- message_signal_
 - car::system::messaging::MessagingSystem, 92
- messaging_container
 - car::display::console::component::debug::DebugMessagingTextbox, 56
- messaging_system_
 - car::system::CarSystem, 46
- messaging_textbox
 - car::display::console::component::debug::DebugMessagingTextbox, 56
- messaging_title
 - car::display::console::component::debug::DebugMessagingTextbox, 56
- MessagingSystem
 - car::system::messaging::MessagingSystem, 88
- motor_pin
 - TB6612, 108
- movement_controller
 - car::system::movement::MovementSystem, 97
- movement_system_
 - car::system::CarSystem, 46
- MovementSystem
 - car::system::movement::MovementSystem, 94
- my_custom_menu
 - car::display::console::screen::LoggingScreen, 82
- None
 - StreamType.h, 131
- offset
 - TB6612, 108
 - test_front_wheels.cpp, 141
- on_connect_failure

car::display::console::component::main::ConnectButton, 50
 on_disconnect_signal_
 car::system::messaging::MessagingSystem, 92
 on_reload
 rpi_daemon, 100
 on_start
 rpi_daemon, 101
 on_stop
 rpi_daemon, 101
 on_update
 rpi_daemon, 101
 onDisconnect
 car::system::messaging::MessagingSystem, 90
 onFirstMessage
 car::system::messaging::MessagingSystem, 91
 onMessageCallback
 car::system::messaging::MessagingSystem, 91
 operator=
 car::system::device::CameraDevice, 37
 pca9685
 BackWheels, 31
 placeholder
 car::display::console::component::settings::SettingsEditConfig, 103
 plugin_manager_
 car::system::CarSystem, 46
 plugins
 car::plugin::PluginManager, 99
 previous_camera_servo_1_angle_slider_angle
 car::display::console::component::debug::DebugMovementRenderer, 61
 previous_camera_servo_2_angle_slider_angle
 car::display::console::component::debug::DebugMovementRenderer, 61
 previous_front_wheels_angle_slider_value
 car::display::console::component::debug::DebugMovementRenderer, 61
 previous_rear_left_wheel_speed_slider_value
 car::display::console::component::debug::DebugMovementRenderer, 61
 previous_rear_right_wheel_speed_slider_value
 car::display::console::component::debug::DebugMovementRenderer, 61
 previous_rear_wheels_speed_slider_value
 car::display::console::component::debug::DebugMovementRenderer, 62
 pwm_pin
 TB6612, 108
 README.md, 142
 ready
 BackWheels, 30
 rear_left_wheel_speed_slider
 car::display::console::component::debug::DebugMovementRenderer, 62
 rear_left_wheel_speed_slider_value
 car::display::console::component::main::ConnectButton, 50
 car::display::console::component::debug::DebugMovementRenderer, 62
 car::display::console::component::debug::DebugMovementRenderer, 62
 rear_right_wheel_speed_slider_value
 car::display::console::component::debug::DebugMovementRenderer, 62
 rear_wheel_direction
 car::display::console::component::debug::DebugMovementRenderer, 62
 REAR_WHEEL_DIRECTION_BACKWARD_MESSAGE
 car::display::console::component::debug::DebugMovementRenderer, 62
 rear_wheel_direction_checkbox_component
 car::display::console::component::debug::DebugMovementRenderer, 63
 rear_wheel_direction_debounce
 car::display::console::component::debug::DebugMovementRenderer, 63
 REAR_WHEEL_DIRECTION_FORWARD_MESSAGE
 car::display::console::component::debug::DebugMovementRenderer, 63
 rear_wheel_direction_signal
 car::display::console::component::debug::DebugMovementRenderer, 63
 rear_wheel_direction_status
 car::display::console::component::debug::DebugMovementRenderer, 63
 rear_wheel_menu_entry
 car::display::console::component::debug::DebugMovementRenderer, 63
 rear_wheel_speed_slider
 car::display::console::component::debug::DebugMovementRenderer, 63
 rear_wheels_speed_slider_value
 car::display::console::component::debug::DebugMovementRenderer, 63
 reload
 car::system::CarSystem, 44
 repository/packages/tb6612/tb6612/include/TB6612.h, 142
 repository/packages/tb6612/tb6612/src/TB6612.cpp, 142
 right_wheel
 BackWheels, 31
 rpi_daemon, 100
 any_configuration_empty, 101
 attempted_to_reconnect, 101
 car_system, 101
 connect, 100
 connection_ms_interval, 102
 last_connected, 102
 on_reload, 100
 on_start, 101
 on_stop, 101
 on_update, 101
 update, 101

- run
 - car::display::console::CarConsole, 40
- running
 - car::system::device::lidar::LidarScanner, 80
- scan_data_
 - car::system::device::lidar::LidarDevice, 75
 - car::system::device::lidar::LidarScanner, 80
- scan_data_mutex_
 - car::system::device::lidar::LidarScanner, 80
- scan_generator_
 - car::system::device::lidar::LidarScanner, 81
- selected_line
 - car::display::console::screen::LoggingScreen, 82
- selection_signal_
 - car::system::messaging::MessagingSystem, 93
- sendData
 - car::system::CarSystem, 44
- sendMessage
 - car::system::messaging::MessagingSystem, 91
- servo_menu_entry
 - car::display::console::component::debug::DebugMovementRenderer, 64
- setAngle
 - test_front_wheels.cpp, 140
- setAngleToAnalog
 - test_front_wheels.cpp, 141
- setBehaviourTree
 - behaviour_tree::BehaviourTreeHandler, 33
- setCameraFps
 - car::configuration::Configuration, 47
- setCameraServo1Angle
 - car::system::movement::controller::AbstractMovementController, 26
 - car::system::movement::controller::DummyMovementController, 68
 - car::system::movement::MovementSystem, 94
- setCameraServo2Angle
 - car::system::movement::controller::AbstractMovementController, 26
 - car::system::movement::controller::DummyMovementController, 68
 - car::system::movement::MovementSystem, 95
- setConfigFilePath
 - car::configuration::JsonConfiguration, 72
- setConfiguration
 - car::system::CarSystem, 44
 - car::system::messaging::MessagingSystem, 91
- setErrorMessage
 - car::display::console::component::main::MainErrorModal, 83
- setFrontWheelsAngle
 - car::system::movement::controller::AbstractMovementController, 26
 - car::system::movement::controller::DummyMovementController, 68
 - car::system::movement::MovementSystem, 95
- setOffset
 - TB6612, 107
- setPWM
 - TB6612, 107
- setRearLeftWheelDirectionToBackward
 - car::system::movement::controller::AbstractMovementController, 26
 - car::system::movement::controller::DummyMovementController, 69
 - car::system::movement::MovementSystem, 95
- setRearLeftWheelDirectionToForward
 - car::system::movement::controller::AbstractMovementController, 26
 - car::system::movement::controller::DummyMovementController, 69
 - car::system::movement::MovementSystem, 95
- setRearLeftWheelSpeed
 - car::system::movement::controller::AbstractMovementController, 27
 - car::system::movement::controller::DummyMovementController, 69
 - car::system::movement::MovementSystem, 95
- setRearRightWheelDirectionToBackward
 - car::system::movement::controller::AbstractMovementController, 27
 - car::system::movement::controller::DummyMovementController, 69
 - car::system::movement::MovementSystem, 95
- setRearRightWheelDirectionToForward
 - car::system::movement::controller::AbstractMovementController, 27
 - car::system::movement::controller::DummyMovementController, 69
 - car::system::movement::MovementSystem, 95
- setRearRightWheelSpeed
 - car::system::movement::controller::AbstractMovementController, 27
 - car::system::movement::controller::DummyMovementController, 69
 - car::system::movement::MovementSystem, 96
- setRearWheelsDirectionToBackward
 - car::system::movement::controller::AbstractMovementController, 27
 - car::system::movement::controller::DummyMovementController, 70
 - car::system::movement::MovementSystem, 96
- setRearWheelsDirectionToForward
 - car::system::movement::controller::AbstractMovementController, 28
 - car::system::movement::controller::DummyMovementController, 70
 - car::system::movement::MovementSystem, 96
- setRearWheelsSpeed
 - car::system::movement::controller::AbstractMovementController, 28
 - car::system::movement::controller::DummyMovementController, 70
 - car::system::movement::MovementSystem, 96
- setScanData
 - car::system::device::lidar::LidarDevice, 74

- setSpeed
 - BackWheels, 30
- settings_edit_config
 - car::display::console::screen::SettingsScreen, 106
- settings_file_path
 - car::display::console::component::settings::SettingsEditConfig, 104
- SettingsEditConfig
 - car::display::console::component::settings::SettingsEditConfig, 102
- SettingsEditConfig.cxx
 - SETTINGSEDTCONFIG_CXX, 151
- SETTINGSEDTCONFIG_CXX
 - SettingsEditConfig.cxx, 151
- SettingsScreen
 - car::display::console::screen::SettingsScreen, 104
- SettingsScreen.cxx
 - SETTINGSSCREEN_CXX, 153
- SETTINGSSCREEN_CXX
 - SettingsScreen.cxx, 153
- SETUP.md, 143
- sink_it_
 - car::system::logging::VectorSink< Mutex >, 109
- slider_container
 - car::display::console::component::debug::DebugMovementHandler, 64
- speed
 - BackWheels, 32
- start
 - car::system::CarSystem, 45
 - car::system::device::CameraDevice, 37
 - car::system::device::DeviceManager, 66
 - car::system::device::lidar::LidarDevice, 74
 - car::system::device::lidar::LidarDummy, 77
 - car::system::device::lidar::LidarScanner, 79
 - car::system::movement::MovementSystem, 96
- startBehaviourTree
 - behaviour_tree::BehaviourTreeHandler, 33
- started
 - car::system::CarSystem, 46
- status
 - car::display::console::component::debug::DebugEnabler, 53
- stop
 - BackWheels, 30
 - behaviour_tree::BehaviourTreeHandler, 34
 - car::plugin::Plugin, 98
 - car::plugin::PluginManager, 99
 - car::system::CarSystem, 45
 - car::system::device::CameraDevice, 37
 - car::system::device::DeviceManager, 66
 - car::system::device::lidar::LidarDevice, 74
 - car::system::device::lidar::LidarDummy, 77
 - car::system::device::lidar::LidarScanner, 79
 - car::system::messaging::MessagingSystem, 91
 - car::system::movement::controller::AbstractMovementController, 28
- car::system::movement::controller::DummyMovementController, 70
- car::system::movement::MovementSystem, 96
- TB6612, 107
- stopBehaviourTree
 - behaviour_tree::BehaviourTreeHandler, 34
- StreamType
 - StreamType.h, 131
- StreamType.h
 - Both, 131
 - Camera, 131
 - Lidar, 131
 - None, 131
 - StreamType, 131
- TB6612, 106
 - backward, 107
 - forward, 107
 - getMotorPin, 107
 - getPWMPin, 107
 - motor_pin, 108
 - offset, 108
 - pwm_pin, 108
 - setOffset, 107
 - setPWM, 107
 - stop, 107
 - TB6612, 106
- terminate
 - car::display::console::CarConsole, 40
 - car::plugin::PluginManager, 99
 - car::system::CarSystem, 45
 - car::system::device::CameraDevice, 37
 - car::system::device::DeviceManager, 66
 - car::system::device::lidar::LidarDevice, 75
 - car::system::device::lidar::LidarDummy, 77
 - car::system::device::lidar::LidarScanner, 79
 - car::system::messaging::MessagingSystem, 91
 - car::system::movement::controller::AbstractMovementController, 28
 - car::system::movement::controller::DummyMovementController, 70
 - car::system::movement::MovementSystem, 96
- terminate_handler
 - main.cpp, 113
- test
 - test_rear_wheels.cpp, 142
- test_front_wheels.cpp
 - main, 140
 - map, 140
 - offset, 141
 - setAngle, 140
 - setAngleToAnalog, 141
- test_rear_wheels.cpp
 - main, 141
 - test, 142
- tick_count
 - behaviour_tree::BehaviourTreeHandler, 35
- tryConnect
 - car::system::CarSystem, 45

- car::system::messaging::MessagingSystem, 92
- tui/README.md, 142
- tui/SETUP.md, 143
- tui/src/car/configuration/JsonConfiguration.cxx, 143
- tui/src/car/display/console/CarConsole.cpp, 144
- tui/src/car/display/console/CarConsole.h, 144, 145
- tui/src/car/display/console/component/debug/DebugEnabler.cxx, 145
- tui/src/car/display/console/component/debug/DebugLidarCheckbox.cxx, 146
- tui/src/car/display/console/component/debug/DebugMessagingTextBox.cxx, 147
- tui/src/car/display/console/component/debug/DebugMovementRenderer.cxx, 147
- tui/src/car/display/console/component/main/ConnectButton.cxx, 148
- tui/src/car/display/console/component/main/MainErrorModal.cxx, 149
- tui/src/car/display/console/component/main/MainExitModal.cxx, 150
- tui/src/car/display/console/component/settings/SettingsEditConfig.cxx, 150
- tui/src/car/display/console/screen/LoggingScreen.cxx, 151
- tui/src/car/display/console/screen/MainScreen.cxx, 152
- tui/src/car/display/console/screen/SettingsScreen.cxx, 153
- tui/src/main.cpp, 113
- update
 - behaviour_tree::BehaviourTreeHandler, 34
 - car::display::console::screen::SettingsScreen, 105
 - car::plugin::Plugin, 98
 - car::plugin::PluginManager, 99
 - car::system::CarSystem, 45
 - car::system::device::CameraDevice, 38
 - car::system::device::DeviceManager, 66
 - car::system::device::lidar::LidarDevice, 75
 - car::system::device::lidar::LidarDummy, 77
 - car::system::device::lidar::LidarScanner, 80
 - rpi_daemon, 101
- updateCameraServo1
 - car::display::console::component::debug::DebugMovementRenderer, 59
- updateCameraServo2
 - car::display::console::component::debug::DebugMovementRenderer, 59
- updateFrontWheels
 - car::display::console::component::debug::DebugMovementRenderer, 59
- updateRearWheels
 - car::display::console::component::debug::DebugMovementRenderer, 59
- use_camera
 - car::configuration::Configuration, 48
- use_lidar
 - car::configuration::Configuration, 48
- uuid
 - car::system::messaging::MessagingSystem::FirstMessageStruct, 71
 - uuid_
 - car::system::messaging::MessagingSystem, 93
 - vector_sink
 - car::display::console::CarConsole, 40
 - car::display::console::screen::LoggingScreen, 82
 - vector_sink_mt
 - car::system::logging, 24
 - VectorSink
 - car::system::logging::VectorSink< Mutex >, 109
 - websocket_
 - car::system::messaging::MessagingSystem, 93
 - websocket_url_
 - car::system::messaging::MessagingSystem, 93