

## Behaviour Tree PiCar-V

Generated by Doxygen 1.9.5



<b>1 Behaviour Tree PiCar Daemon</b>	<b>1</b>
1.1 Installation:	1
1.2 View Logs	1
<b>2 Raspberry Pi Daemon</b>	<b>3</b>
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
<b>3 Raspberry Pi</b>	<b>5</b>
3.1 Main Technologies	5
3.2 Setting up Raspberry Pi	5
<b>4 Behaviour Tree PiCar-V Setup</b>	<b>7</b>
<b>5 Raspberry Pi</b>	<b>9</b>
5.1 Installation	9
5.2 Main Technologies	9
5.3 Setting up Raspberry Pi	9
<b>6 Namespace Index</b>	<b>11</b>
6.1 Namespace List	11
<b>7 Hierarchical Index</b>	<b>13</b>
7.1 Class Hierarchy	13
<b>8 Class Index</b>	<b>15</b>
8.1 Class List	15
<b>9 File Index</b>	<b>17</b>
9.1 File List	17
<b>10 Namespace Documentation</b>	<b>19</b>
10.1 behaviour_tree Namespace Reference	19
10.2 car Namespace Reference	19
10.3 car::configuration Namespace Reference	19
10.4 car::display Namespace Reference	19
10.5 car::display::console Namespace Reference	20
10.6 car::display::console::component Namespace Reference	20
10.7 car::display::console::component::debug Namespace Reference	20

10.8 car::display::console::component::main Namespace Reference . . . . .	20
10.9 car::display::console::component::settings Namespace Reference . . . . .	20
10.10 car::display::console::screen Namespace Reference . . . . .	21
10.11 car::plugin Namespace Reference . . . . .	21
10.12 car::system Namespace Reference . . . . .	21
10.13 car::system::device Namespace Reference . . . . .	21
10.14 car::system::device::lidar Namespace Reference . . . . .	21
10.15 car::system::logging Namespace Reference . . . . .	22
10.15.1 Typedef Documentation . . . . .	22
10.15.1.1 vector_sink_mt . . . . .	22
10.16 car::system::messaging Namespace Reference . . . . .	22
10.17 car::system::movement Namespace Reference . . . . .	22
10.18 car::system::movement::controller Namespace Reference . . . . .	22
<b>11 Class Documentation</b> . . . . .	<b>23</b>
11.1 car::system::movement::controller::AbstractMovementController Class Reference . . . . .	23
11.1.1 Member Function Documentation . . . . .	23
11.1.1.1 initialize() . . . . .	24
11.1.1.2 setCameraServo1Angle() . . . . .	24
11.1.1.3 setCameraServo2Angle() . . . . .	24
11.1.1.4 setFrontWheelsAngle() . . . . .	24
11.1.1.5 setRearLeftWheelDirectionToBackward() . . . . .	24
11.1.1.6 setRearLeftWheelDirectionToForward() . . . . .	25
11.1.1.7 setRearLeftWheelSpeed() . . . . .	25
11.1.1.8 setRearRightWheelDirectionToBackward() . . . . .	25
11.1.1.9 setRearRightWheelDirectionToForward() . . . . .	25
11.1.1.10 setRearRightWheelSpeed() . . . . .	25
11.1.1.11 setRearWheelsDirectionToBackward() . . . . .	26
11.1.1.12 setRearWheelsDirectionToForward() . . . . .	26
11.1.1.13 setRearWheelsSpeed() . . . . .	26
11.1.1.14 stop() . . . . .	26
11.1.1.15 terminate() . . . . .	26
11.2 BackWheels Class Reference . . . . .	27
11.2.1 Constructor & Destructor Documentation . . . . .	27
11.2.1.1 BackWheels() . . . . .	27
11.2.2 Member Function Documentation . . . . .	27
11.2.2.1 backward() . . . . .	27
11.2.2.2 calibration() . . . . .	28
11.2.2.3 caliLeft() . . . . .	28
11.2.2.4 caliOK() . . . . .	28
11.2.2.5 caliRight() . . . . .	28
11.2.2.6 forward() . . . . .	28

11.2.2.7 getSpeed()	28
11.2.2.8 ready()	28
11.2.2.9 setSpeed()	28
11.2.2.10 stop()	29
11.2.3 Member Data Documentation	29
11.2.3.1 cali_forward_A	29
11.2.3.2 cali_forward_B	29
11.2.3.3 forward_A	29
11.2.3.4 forward_B	29
11.2.3.5 left_wheel	29
11.2.3.6 pca9685	29
11.2.3.7 right_wheel	30
11.2.3.8 speed	30
11.3 behaviour_tree::BehaviourTreeHandler Class Reference	30
11.3.1 Member Function Documentation	31
11.3.1.1 _setBehaviourTree()	31
11.3.1.2 getName()	31
11.3.1.3 handleCommand()	31
11.3.1.4 initialize()	31
11.3.1.5 setBehaviourTree()	31
11.3.1.6 startBehaviourTree()	32
11.3.1.7 stop()	32
11.3.1.8 stopBehaviourTree()	32
11.3.1.9 update()	32
11.3.2 Member Data Documentation	32
11.3.2.1 behaviour_tree	32
11.3.2.2 car_system	32
11.3.2.3 context	33
11.3.2.4 last_connected	33
11.3.2.5 tick_count	33
11.4 car::system::device::CameraDevice Class Reference	33
11.4.1 Constructor & Destructor Documentation	34
11.4.1.1 CameraDevice() [1/3]	34
11.4.1.2 CameraDevice() [2/3]	34
11.4.1.3 CameraDevice() [3/3]	34
11.4.1.4 ~CameraDevice()	34
11.4.2 Member Function Documentation	34
11.4.2.1 create()	35
11.4.2.2 disconnect()	35
11.4.2.3 getFrameBuffer()	35
11.4.2.4 operator=() [1/2]	35
11.4.2.5 operator=() [2/2]	35

11.4.2.6 start()	35
11.4.2.7 stop()	35
11.4.2.8 terminate()	36
11.4.2.9 update()	36
11.4.3 Friends And Related Function Documentation	36
11.4.3.1 DeviceManager	36
11.4.4 Member Data Documentation	36
11.4.4.1 camera_	36
11.4.4.2 camera_mutex_	36
11.4.4.3 configuration	36
11.4.4.4 connected_	37
11.4.4.5 frame_buffer_	37
11.4.4.6 last	37
11.5 car::display::console::CarConsole Class Reference	37
11.5.1 Constructor & Destructor Documentation	37
11.5.1.1 CarConsole()	38
11.5.2 Member Function Documentation	38
11.5.2.1 initialize()	38
11.5.2.2 run()	38
11.5.2.3 terminate()	38
11.5.3 Member Data Documentation	38
11.5.3.1 car_system	38
11.5.3.2 json_configuration	38
11.5.3.3 vector_sink	39
11.6 behaviour_tree::CarContext Class Reference	39
11.6.1 Constructor & Destructor Documentation	39
11.6.1.1 CarContext()	39
11.6.2 Member Function Documentation	39
11.6.2.1 _()	40
11.6.2.2 getCarSystem()	40
11.6.3 Member Data Documentation	40
11.6.3.1 car_system	40
11.7 car::system::CarSystem Class Reference	40
11.7.1 Constructor & Destructor Documentation	41
11.7.1.1 CarSystem()	41
11.7.2 Member Function Documentation	41
11.7.2.1 disconnect()	41
11.7.2.2 getConfiguration()	41
11.7.2.3 getDeviceManager()	42
11.7.2.4 getMessagingSystem()	42
11.7.2.5 getMovementSystem()	42
11.7.2.6 getPlugin()	42

11.7.2.7 initialize()	42
11.7.2.8 reload()	42
11.7.2.9 sendData()	42
11.7.2.10 setConfiguration()	43
11.7.2.11 start()	43
11.7.2.12 stop()	43
11.7.2.13 terminate()	43
11.7.2.14 tryConnect()	43
11.7.2.15 update()	43
11.7.3 Member Data Documentation	43
11.7.3.1 configuration_	43
11.7.3.2 device_manager_	44
11.7.3.3 initialized	44
11.7.3.4 messaging_system_	44
11.7.3.5 movement_system_	44
11.7.3.6 plugin_manager_	44
11.7.3.7 started	44
11.8 car::configuration::Configuration Struct Reference	44
11.8.1 Member Function Documentation	45
11.8.1.1 getCameraFpsInterval()	45
11.8.1.2 setCameraFps()	45
11.8.2 Member Data Documentation	45
11.8.2.1 behaviour_tree_update_ms_interval	45
11.8.2.2 camera_fps	46
11.8.2.3 camera_fps_interval	46
11.8.2.4 camera_index	46
11.8.2.5 host	46
11.8.2.6 lidar_port	46
11.8.2.7 use_camera	46
11.8.2.8 use_lidar	46
11.9 car::display::console::component::main::ConnectButton Class Reference	47
11.9.1 Constructor & Destructor Documentation	47
11.9.1.1 ConnectButton()	47
11.9.2 Member Function Documentation	47
11.9.2.1 element()	47
11.9.3 Member Data Documentation	47
11.9.3.1 button_pressed	48
11.9.3.2 car_system	48
11.9.3.3 main_button	48
11.9.3.4 main_button_text	48
11.9.3.5 main_debounce	48
11.9.3.6 on_connect_failure	48

11.10 car::display::console::component::debug::DebugEnabler Class Reference	48
11.10.1 Member Function Documentation	49
11.10.1.1 getCheckbox()	49
11.10.1.2 getWarningModal()	49
11.10.1.3 isEnabled()	49
11.10.2 Member Data Documentation	49
11.10.2.1 checkbox_value	49
11.10.2.2 component	50
11.10.2.3 debounce	50
11.10.2.4 DEBUG_ENABLE_WARNING_MESSAGE	50
11.10.2.5 DEBUG_MODE_DISABLED_MESSAGE	50
11.10.2.6 DEBUG_MODE_ENABLED_MESSAGE	50
11.10.2.7 DEBUG_MODE_WAIT_MESSAGE	50
11.10.2.8 display_warn_debug_modal	50
11.10.2.9 enabled	51
11.10.2.10 status	51
11.11 car::display::console::component::debug::DebugLidarCheckbox Class Reference	51
11.11.1 Constructor & Destructor Documentation	51
11.11.1.1 DebugLidarCheckbox()	51
11.11.2 Member Function Documentation	52
11.11.2.1 element()	52
11.11.2.2 getLidarMotorSignal()	52
11.11.3 Member Data Documentation	52
11.11.3.1 lidar_motor_checkbox_component	52
11.11.3.2 LIDAR_MOTOR_DISABLED_MESSAGE	52
11.11.3.3 lidar_motor_enabled	52
11.11.3.4 LIDAR_MOTOR_ENABLED_MESSAGE	52
11.11.3.5 lidar_motor_loading_debounce	53
11.11.3.6 lidar_motor_signal	53
11.11.3.7 lidar_motor_status	53
11.12 car::display::console::component::debug::DebugMessagingTextbox Class Reference	53
11.12.1 Constructor & Destructor Documentation	53
11.12.1.1 DebugMessagingTextbox()	54
11.12.2 Member Function Documentation	54
11.12.2.1 element()	54
11.12.3 Member Data Documentation	54
11.12.3.1 message	54
11.12.3.2 message_signal	54
11.12.3.3 messaging_container	54
11.12.3.4 messaging_textbox	54
11.12.3.5 messaging_title	55
11.13 car::display::console::component::debug::DebugMovementRenderer Class Reference	55



11.13.1 Constructor & Destructor Documentation	56
11.13.1.1 DebugMovementRenderer()	56
11.13.2 Member Function Documentation	56
11.13.2.1 element()	56
11.13.2.2 getCameraServo1AngleSliderValue()	56
11.13.2.3 getCameraServo2AngleSliderValue()	56
11.13.2.4 getFrontWheelsAngleSliderValue()	57
11.13.2.5 getRearLeftWheelSpeedSliderValue()	57
11.13.2.6 getRearRightWheelSpeedSliderValue()	57
11.13.2.7 getRearWheelDirectionSignal()	57
11.13.2.8 updateCameraServo1()	57
11.13.2.9 updateCameraServo2()	57
11.13.2.10 updateFrontWheels()	57
11.13.2.11 updateRearWheels()	58
11.13.3 Member Data Documentation	58
11.13.3.1 camera_servo_1_angle_slider	58
11.13.3.2 camera_servo_1_angle_slider_angle	58
11.13.3.3 camera_servo_2_angle_slider	58
11.13.3.4 camera_servo_2_angle_slider_angle	58
11.13.3.5 DEFAULT_FRONT_WHEEL_ANGLE	58
11.13.3.6 DEFAULT_REAR_WHEEL_SPEED	59
11.13.3.7 front_wheels_angle_slider	59
11.13.3.8 front_wheels_angle_slider_value	59
11.13.3.9 previous_camera_servo_1_angle_slider_angle	59
11.13.3.10 previous_camera_servo_2_angle_slider_angle	59
11.13.3.11 previous_front_wheels_angle_slider_value	59
11.13.3.12 previous_rear_left_wheel_speed_slider_value	59
11.13.3.13 previous_rear_right_wheel_speed_slider_value	60
11.13.3.14 previous_rear_wheels_speed_slider_value	60
11.13.3.15 rear_left_wheel_speed_slider	60
11.13.3.16 rear_left_wheel_speed_slider_value	60
11.13.3.17 rear_right_wheel_speed_slider	60
11.13.3.18 rear_right_wheel_speed_slider_value	60
11.13.3.19 rear_wheel_direction	60
11.13.3.20 REAR_WHEEL_DIRECTION_BACKWARD_MESSAGE	61
11.13.3.21 rear_wheel_direction_checkbox_component	61
11.13.3.22 rear_wheel_direction_debounce	61
11.13.3.23 REAR_WHEEL_DIRECTION_FORWARD_MESSAGE	61
11.13.3.24 rear_wheel_direction_signal	61
11.13.3.25 rear_wheel_direction_status	61
11.13.3.26 rear_wheel_menu_entry	61
11.13.3.27 rear_wheel_speed_slider	62

11.13.3.28 rear_wheels_speed_slider_value . . . . .	62
11.13.3.29 servo_menu_entry . . . . .	62
11.13.3.30 slider_container . . . . .	62
11.14 car::system::device::DeviceManager Class Reference . . . . .	62
11.14.1 Constructor & Destructor Documentation . . . . .	63
11.14.1.1 DeviceManager() . . . . .	63
11.14.2 Member Function Documentation . . . . .	63
11.14.2.1 create() . . . . .	63
11.14.2.2 getCameraDevice() . . . . .	63
11.14.2.3 getLidarDevice() . . . . .	63
11.14.2.4 initialize() . . . . .	64
11.14.2.5 isRunning() . . . . .	64
11.14.2.6 start() . . . . .	64
11.14.2.7 stop() . . . . .	64
11.14.2.8 terminate() . . . . .	64
11.14.2.9 update() . . . . .	64
11.14.3 Member Data Documentation . . . . .	64
11.14.3.1 camera_device_ . . . . .	64
11.14.3.2 car_system . . . . .	65
11.14.3.3 is_initialized_ . . . . .	65
11.14.3.4 is_running_ . . . . .	65
11.14.3.5 lidar_device_ . . . . .	65
11.15 car::system::movement::controller::DummyMovementController Class Reference . . . . .	65
11.15.1 Member Function Documentation . . . . .	66
11.15.1.1 initialize() . . . . .	66
11.15.1.2 setCameraServo1Angle() . . . . .	66
11.15.1.3 setCameraServo2Angle() . . . . .	66
11.15.1.4 setFrontWheelsAngle() . . . . .	67
11.15.1.5 setRearLeftWheelDirectionToBackward() . . . . .	67
11.15.1.6 setRearLeftWheelDirectionToForward() . . . . .	67
11.15.1.7 setRearLeftWheelSpeed() . . . . .	67
11.15.1.8 setRearRightWheelDirectionToBackward() . . . . .	67
11.15.1.9 setRearRightWheelDirectionToForward() . . . . .	67
11.15.1.10 setRearRightWheelSpeed() . . . . .	68
11.15.1.11 setRearWheelsDirectionToBackward() . . . . .	68
11.15.1.12 setRearWheelsDirectionToForward() . . . . .	68
11.15.1.13 setRearWheelsSpeed() . . . . .	68
11.15.1.14 stop() . . . . .	68
11.15.1.15 terminate() . . . . .	69
11.16 car::system::messaging::MessagingSystem::FirstMessageStruct Struct Reference . . . . .	69
11.16.1 Member Data Documentation . . . . .	69
11.16.1.1 condition . . . . .	69

---

11.16.1.2 error_message . . . . .	69
11.16.1.3 uuid . . . . .	69
11.17 car::configuration::JsonConfiguration Class Reference . . . . .	70
11.17.1 Constructor & Destructor Documentation . . . . .	70
11.17.1.1 JsonConfiguration() . . . . .	70
11.17.2 Member Function Documentation . . . . .	70
11.17.2.1 getConfigFilePath() . . . . .	70
11.17.2.2 loadConfiguration() . . . . .	70
11.17.2.3 setConfigFilePath() . . . . .	70
11.17.3 Member Data Documentation . . . . .	71
11.17.3.1 config_file_path . . . . .	71
11.17.3.2 exe_dir . . . . .	71
11.18 car::system::device::lidar::LidarDevice Class Reference . . . . .	71
11.18.1 Member Function Documentation . . . . .	72
11.18.1.1 disconnect() . . . . .	72
11.18.1.2 getScanData() . . . . .	72
11.18.1.3 initialize() . . . . .	72
11.18.1.4 setScanData() . . . . .	72
11.18.1.5 start() . . . . .	72
11.18.1.6 stop() . . . . .	73
11.18.1.7 terminate() . . . . .	73
11.18.1.8 update() . . . . .	73
11.18.2 Friends And Related Function Documentation . . . . .	73
11.18.2.1 DeviceManager . . . . .	73
11.18.3 Member Data Documentation . . . . .	73
11.18.3.1 scan_data_ . . . . .	73
11.19 car::system::device::lidar::LidarDummy Class Reference . . . . .	74
11.19.1 Constructor & Destructor Documentation . . . . .	74
11.19.1.1 LidarDummy() . . . . .	74
11.19.2 Member Function Documentation . . . . .	74
11.19.2.1 disconnect() . . . . .	74
11.19.2.2 initialize() . . . . .	75
11.19.2.3 start() . . . . .	75
11.19.2.4 stop() . . . . .	75
11.19.2.5 terminate() . . . . .	75
11.19.2.6 update() . . . . .	75
11.20 car::system::device::lidar::LidarScanner Class Reference . . . . .	76
11.20.1 Constructor & Destructor Documentation . . . . .	76
11.20.1.1 LidarScanner() . . . . .	76
11.20.2 Member Function Documentation . . . . .	77
11.20.2.1 create() . . . . .	77
11.20.2.2 disconnect() . . . . .	77

11.20.2.3 initialize()	77
11.20.2.4 start()	77
11.20.2.5 stop()	77
11.20.2.6 terminate()	78
11.20.2.7 update()	78
11.20.3 Member Data Documentation	78
11.20.3.1 configuration_	78
11.20.3.2 lidar_	78
11.20.3.3 running	78
11.20.3.4 scan_data_	78
11.20.3.5 scan_data_mutex_	79
11.20.3.6 scan_generator_	79
11.21 car::display::console::screen::LoggingScreen Class Reference	79
11.21.1 Constructor & Destructor Documentation	79
11.21.1.1 LoggingScreen()	79
11.21.2 Member Function Documentation	79
11.21.2.1 element()	80
11.21.3 Member Data Documentation	80
11.21.3.1 line_elements	80
11.21.3.2 menu	80
11.21.3.3 my_custom_menu	80
11.21.3.4 selected_line	80
11.21.3.5 vector_sink	80
11.22 car::display::console::component::main::MainErrorModal Class Reference	81
11.22.1 Constructor & Destructor Documentation	81
11.22.1.1 MainErrorModal()	81
11.22.2 Member Function Documentation	81
11.22.2.1 element()	81
11.22.2.2 setErrorMessage()	81
11.22.3 Member Data Documentation	81
11.22.3.1 error_element	82
11.22.3.2 error_modal_shown	82
11.22.3.3 main_error_modal	82
11.23 car::display::console::component::main::MainExitModal Class Reference	82
11.23.1 Constructor & Destructor Documentation	82
11.23.1.1 MainExitModal()	82
11.23.2 Member Function Documentation	83
11.23.2.1 element()	83
11.23.3 Member Data Documentation	83
11.23.3.1 exit	83
11.23.3.2 exit_modal_shown	83
11.23.3.3 main_exit_modal	83

11.24 car::display::console::screen::MainScreen Class Reference	83
11.24.1 Constructor & Destructor Documentation	84
11.24.1.1 MainScreen()	84
11.24.2 Member Function Documentation	84
11.24.2.1 element()	84
11.24.3 Member Data Documentation	84
11.24.3.1 box	84
11.24.3.2 car_system	84
11.24.3.3 connect_button	85
11.24.3.4 info	85
11.24.3.5 main_component	85
11.24.3.6 main_error_modal	85
11.24.3.7 main_exit_modal	85
11.24.3.8 main_screen	85
11.25 car::system::messaging::MessagingSystem Class Reference	85
11.25.1 Constructor & Destructor Documentation	86
11.25.1.1 MessagingSystem()	87
11.25.2 Member Function Documentation	87
11.25.2.1 getCommandSignal()	87
11.25.2.2 getDisconnectSignal()	87
11.25.2.3 getFirstMessage()	87
11.25.2.4 getMessageSignal()	87
11.25.2.5 getSelectionSignal()	87
11.25.2.6 getUUID()	88
11.25.2.7 handleMessage()	88
11.25.2.8 initialize()	88
11.25.2.9 initializeWebSocket()	88
11.25.2.10 isConnected()	88
11.25.2.11 onDisconnect()	89
11.25.2.12 onFirstMessage()	89
11.25.2.13 onMessageCallback()	89
11.25.2.14 sendMessage()	89
11.25.2.15 setConfiguration()	89
11.25.2.16 stop()	89
11.25.2.17 terminate()	90
11.25.2.18 tryConnect()	90
11.25.3 Member Data Documentation	90
11.25.3.1 command_signal_	90
11.25.3.2 configuration_	90
11.25.3.3 connected_	90
11.25.3.4 message_signal_	90
11.25.3.5 on_disconnect_signal_	91

11.25.3.6 selection_signal_ . . . . .	91
11.25.3.7 uuid_ . . . . .	91
11.25.3.8 websocket_ . . . . .	91
11.25.3.9 websocket_url_ . . . . .	91
11.26 car::system::movement::MovementSystem Class Reference . . . . .	91
11.26.1 Constructor & Destructor Documentation . . . . .	92
11.26.1.1 MovementSystem() . . . . .	92
11.26.1.2 ~MovementSystem() . . . . .	92
11.26.2 Member Function Documentation . . . . .	92
11.26.2.1 initialize() . . . . .	92
11.26.2.2 setCameraServo1Angle() . . . . .	93
11.26.2.3 setCameraServo2Angle() . . . . .	93
11.26.2.4 setFrontWheelsAngle() . . . . .	93
11.26.2.5 setRearLeftWheelDirectionToBackward() . . . . .	93
11.26.2.6 setRearLeftWheelDirectionToForward() . . . . .	93
11.26.2.7 setRearLeftWheelSpeed() . . . . .	93
11.26.2.8 setRearRightWheelDirectionToBackward() . . . . .	93
11.26.2.9 setRearRightWheelDirectionToForward() . . . . .	94
11.26.2.10 setRearRightWheelSpeed() . . . . .	94
11.26.2.11 setRearWheelsDirectionToBackward() . . . . .	94
11.26.2.12 setRearWheelsDirectionToForward() . . . . .	94
11.26.2.13 setRearWheelsSpeed() . . . . .	94
11.26.2.14 start() . . . . .	94
11.26.2.15 stop() . . . . .	94
11.26.2.16 terminate() . . . . .	95
11.26.3 Member Data Documentation . . . . .	95
11.26.3.1 movement_controller . . . . .	95
11.27 car::plugin::Plugin Class Reference . . . . .	95
11.27.1 Member Function Documentation . . . . .	95
11.27.1.1 getName() . . . . .	95
11.27.1.2 initialize() . . . . .	96
11.27.1.3 stop() . . . . .	96
11.27.1.4 update() . . . . .	96
11.28 car::plugin::PluginManager Class Reference . . . . .	96
11.28.1 Member Function Documentation . . . . .	97
11.28.1.1 addPlugin() . . . . .	97
11.28.1.2 getPlugin() . . . . .	97
11.28.1.3 initialize() . . . . .	97
11.28.1.4 stop() . . . . .	97
11.28.1.5 terminate() . . . . .	97
11.28.1.6 update() . . . . .	97
11.28.2 Member Data Documentation . . . . .	97

11.28.2.1 plugins	98
11.29 rpi_daemon Class Reference	98
11.29.1 Member Function Documentation	98
11.29.1.1 connect()	98
11.29.1.2 on_reload()	99
11.29.1.3 on_start()	99
11.29.1.4 on_stop()	99
11.29.1.5 on_update()	99
11.29.1.6 update()	99
11.29.2 Member Data Documentation	99
11.29.2.1 any_configuration_empty	99
11.29.2.2 attempted_to_reconnect	99
11.29.2.3 car_system	100
11.29.2.4 connection_ms_interval	100
11.29.2.5 last_connected	100
11.30 car::display::console::component::settings::SettingsEditConfig Class Reference	100
11.30.1 Constructor & Destructor Documentation	100
11.30.1.1 SettingsEditConfig()	101
11.30.2 Member Function Documentation	101
11.30.2.1 element()	101
11.30.3 Member Data Documentation	101
11.30.3.1 car_system	101
11.30.3.2 input_settings_file_path	101
11.30.3.3 json_configuration	101
11.30.3.4 load_button	101
11.30.3.5 placeholder	102
11.30.3.6 settings_file_path	102
11.31 car::display::console::screen::SettingsScreen Class Reference	102
11.31.1 Constructor & Destructor Documentation	102
11.31.1.1 SettingsScreen()	102
11.31.2 Member Function Documentation	103
11.31.2.1 element()	103
11.31.2.2 update()	103
11.31.3 Member Data Documentation	103
11.31.3.1 car_system	103
11.31.3.2 debug_enabler	103
11.31.3.3 debug_lidar_checkbox	103
11.31.3.4 debug_messaging_text_box	103
11.31.3.5 debug_movement_renderer	104
11.31.3.6 settings_edit_config	104
11.32 TB6612 Class Reference	104
11.32.1 Constructor & Destructor Documentation	104

11.32.1.1 TB6612()	104
11.32.2 Member Function Documentation	105
11.32.2.1 backward()	105
11.32.2.2 forward()	105
11.32.2.3 getMotorPin()	105
11.32.2.4 getPWMPin()	105
11.32.2.5 setOffset()	105
11.32.2.6 setPWM()	105
11.32.2.7 stop()	105
11.32.3 Member Data Documentation	106
11.32.3.1 motor_pin	106
11.32.3.2 offset	106
11.32.3.3 pwm_pin	106
11.33 car::system::logging::VectorSink< Mutex > Class Template Reference	106
11.33.1 Constructor & Destructor Documentation	107
11.33.1.1 VectorSink()	107
11.33.2 Member Function Documentation	107
11.33.2.1 flush_()	107
11.33.2.2 get_log_messages()	107
11.33.2.3 sink_it_()	107
11.33.3 Member Data Documentation	107
11.33.3.1 log_messages	108
11.33.3.2 max_lines	108
<b>12 File Documentation</b>	<b>109</b>
12.1 behaviour_tree/src/main.cpp File Reference	109
12.1.1 Function Documentation	109
12.1.1.1 kbhit()	109
12.1.1.2 main()	110
12.2 daemon/src/main.cpp File Reference	110
12.2.1 Function Documentation	110
12.2.1.1 getLidarDevice()	110
12.2.1.2 getMovementController()	111
12.2.1.3 main()	111
12.2.1.4 terminate_handler()	111
12.3 tui/src/main.cpp File Reference	111
12.3.1 Function Documentation	111
12.3.1.1 getLidarDevice()	112
12.3.1.2 getMovementController()	112
12.3.1.3 main()	112
12.4 common/include/behaviour_tree/BehaviourTreeHandler.hpp File Reference	112
12.5 BehaviourTreeHandler.hpp	113



12.6 common/include/behaviour_tree/CarContext.hpp File Reference . . . . .	115
12.7 CarContext.hpp . . . . .	115
12.8 common/include/car/configuration/Configuration.h File Reference . . . . .	116
12.9 Configuration.h . . . . .	116
12.10 common/include/car/plugin/Plugin.h File Reference . . . . .	116
12.11 Plugin.h . . . . .	117
12.12 common/include/car/plugin/PluginManager.h File Reference . . . . .	117
12.13 PluginManager.h . . . . .	118
12.14 common/include/car/system/CarSystem.h File Reference . . . . .	119
12.15 CarSystem.h . . . . .	119
12.16 common/include/car/system/device/CameraDevice.h File Reference . . . . .	120
12.17 CameraDevice.h . . . . .	121
12.18 common/include/car/system/device/DeviceManager.h File Reference . . . . .	121
12.19 DeviceManager.h . . . . .	122
12.20 common/include/car/system/device/lidar/LidarDevice.h File Reference . . . . .	123
12.21 LidarDevice.h . . . . .	123
12.22 common/include/car/system/device/lidar/LidarDummy.h File Reference . . . . .	124
12.23 LidarDummy.h . . . . .	124
12.24 common/include/car/system/device/lidar/LidarScanner.h File Reference . . . . .	124
12.25 LidarScanner.h . . . . .	125
12.26 common/include/car/system/logging/VectorSink.h File Reference . . . . .	126
12.27 VectorSink.h . . . . .	127
12.28 common/include/car/system/messaging/MessagingSystem.h File Reference . . . . .	127
12.29 MessagingSystem.h . . . . .	128
12.30 common/include/car/system/messaging/StreamType.h File Reference . . . . .	129
12.30.1 Enumeration Type Documentation . . . . .	129
12.30.1.1 StreamType . . . . .	129
12.31 StreamType.h . . . . .	130
12.32 common/include/car/system/movement/controller/AbstractMovementController.h File Reference . . . . .	130
12.33 AbstractMovementController.h . . . . .	130
12.34 common/include/car/system/movement/controller/DeviceMovementController.h File Reference . . . . .	131
12.35 DeviceMovementController.h . . . . .	131
12.36 common/include/car/system/movement/controller/DummyMovementController.h File Reference . . . . .	132
12.37 DummyMovementController.h . . . . .	132
12.38 common/include/car/system/movement/devices/RearWheel.h File Reference . . . . .	133
12.39 RearWheel.h . . . . .	133
12.40 common/include/car/system/movement/devices/Servo.h File Reference . . . . .	133
12.41 Servo.h . . . . .	133
12.42 common/include/car/system/movement/MovementSystem.h File Reference . . . . .	134
12.43 MovementSystem.h . . . . .	134
12.44 common/src/car/system/CarSystem.cpp File Reference . . . . .	136
12.45 common/src/car/system/device/CameraDevice.cpp File Reference . . . . .	136

12.46 common/src/car/system/device/DeviceManager.cpp File Reference . . . . .	136
12.47 common/src/car/system/messaging/MessagingSystem.cpp File Reference . . . . .	137
12.48 common/src/car/system/movement/controller/DeviceMovementController.cpp File Reference . . . . .	137
12.49 common/src/car/system/movement/controller/DummyMovementController.cpp File Reference . . . . .	137
12.50 common/src/car/system/movement/devices/RearWheel.cpp File Reference . . . . .	138
12.51 common/src/car/system/movement/devices/Servo.cpp File Reference . . . . .	138
12.52 common/tests/pca9685/test_front_wheels.cpp File Reference . . . . .	138
12.52.1 Function Documentation . . . . .	138
12.52.1.1 main() . . . . .	138
12.52.1.2 map() . . . . .	138
12.52.1.3 setAngle() . . . . .	139
12.52.1.4 setAngleToAnalog() . . . . .	139
12.52.2 Variable Documentation . . . . .	139
12.52.2.1 offset . . . . .	139
12.53 common/tests/tb6612/test_rear_wheels.cpp File Reference . . . . .	139
12.53.1 Function Documentation . . . . .	139
12.53.1.1 main() . . . . .	140
12.53.1.2 test() . . . . .	140
12.54 daemon/install/README.md File Reference . . . . .	140
12.55 daemon/README.md File Reference . . . . .	140
12.56 README.md File Reference . . . . .	140
12.57 tui/README.md File Reference . . . . .	140
12.58 repository/packages/t/tb6612/tb6612/include/TB6612.h File Reference . . . . .	140
12.59 TB6612.h . . . . .	140
12.60 repository/packages/t/tb6612/tb6612/src/TB6612.cpp File Reference . . . . .	141
12.61 SETUP.md File Reference . . . . .	141
12.62 tui/src/car/configuration/JsonConfiguration.cxx File Reference . . . . .	141
12.62.1 Macro Definition Documentation . . . . .	142
12.62.1.1 JSONCONFIGURATION_CXX . . . . .	142
12.63 tui/src/car/display/console/CarConsole.cpp File Reference . . . . .	142
12.64 tui/src/car/display/console/CarConsole.h File Reference . . . . .	142
12.65 CarConsole.h . . . . .	143
12.66 tui/src/car/display/console/component/debug/DebugEnabler.cxx File Reference . . . . .	143
12.66.1 Macro Definition Documentation . . . . .	144
12.66.1.1 DEBUGENABLER_CXX . . . . .	144
12.67 tui/src/car/display/console/component/debug/DebugLidarCheckbox.cxx File Reference . . . . .	144
12.67.1 Macro Definition Documentation . . . . .	144
12.67.1.1 DEBUGLIDARCHECKBOX_CXX . . . . .	144
12.68 tui/src/car/display/console/component/debug/DebugMessagingTextbox.cxx File Reference . . . . .	145
12.68.1 Macro Definition Documentation . . . . .	145
12.68.1.1 DEBUGMESSAGINGTEXTBOX_CXX . . . . .	145
12.69 tui/src/car/display/console/component/debug/DebugMovementRenderer.cxx File Reference . . . . .	145

12.69.1 Macro Definition Documentation . . . . .	146
12.69.1.1 DEBUGMOVEMENTRENDERER_CXX . . . . .	146
12.70 tui/src/car/display/console/component/main/ConnectButton.cxx File Reference . . . . .	146
12.70.1 Macro Definition Documentation . . . . .	147
12.70.1.1 CONNECTBUTTON_CXX . . . . .	147
12.71 tui/src/car/display/console/component/main/MainErrorModal.cxx File Reference . . . . .	147
12.71.1 Macro Definition Documentation . . . . .	147
12.71.1.1 MAINERRORMODAL_CXX . . . . .	147
12.72 tui/src/car/display/console/component/main/MainExitModal.cxx File Reference . . . . .	148
12.72.1 Macro Definition Documentation . . . . .	148
12.72.1.1 MAINEXITMODAL_CXX . . . . .	148
12.73 tui/src/car/display/console/component/settings/SettingsEditConfig.cxx File Reference . . . . .	148
12.73.1 Macro Definition Documentation . . . . .	149
12.73.1.1 SETTINGSEITCONFIG_CXX . . . . .	149
12.74 tui/src/car/display/console/screen/LoggingScreen.cxx File Reference . . . . .	149
12.74.1 Macro Definition Documentation . . . . .	150
12.74.1.1 LOGGINGSCREEN_CXX . . . . .	150
12.75 tui/src/car/display/console/screen/MainScreen.cxx File Reference . . . . .	150
12.75.1 Macro Definition Documentation . . . . .	150
12.75.1.1 MAINSCREEN_CXX . . . . .	150
12.76 tui/src/car/display/console/screen/SettingsScreen.cxx File Reference . . . . .	151
12.76.1 Macro Definition Documentation . . . . .	151
12.76.1.1 SETTINGSSCREEN_CXX . . . . .	151
<b>Index</b>	<b>153</b>



# Chapter 1

## Behaviour Tree 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

# Behaviour Tree PiCar-V 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`



## Chapter 5

# Raspberry Pi

This sub-project handles all the logic of the Sunfounder PiCar.

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 PiCar

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

# Namespace Index

### 6.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">behaviour_tree</a>	19
<a href="#">car</a>	19
<a href="#">car::configuration</a>	19
<a href="#">car::display</a>	19
<a href="#">car::display::console</a>	20
<a href="#">car::display::console::component</a>	20
<a href="#">car::display::console::component::debug</a>	20
<a href="#">car::display::console::component::main</a>	20
<a href="#">car::display::console::component::settings</a>	20
<a href="#">car::display::console::screen</a>	21
<a href="#">car::plugin</a>	21
<a href="#">car::system</a>	21
<a href="#">car::system::device</a>	21
<a href="#">car::system::device::lidar</a>	21
<a href="#">car::system::logging</a>	22
<a href="#">car::system::messaging</a>	22
<a href="#">car::system::movement</a>	22
<a href="#">car::system::movement::controller</a>	22





## Chapter 7

# Hierarchical Index

### 7.1 Class Hierarchy

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

car::system::movement::controller::AbstractMovementController . . . . .	23
car::system::movement::controller::DummyMovementController . . . . .	65
BackWheels . . . . .	27
spdlog::sinks::base_sink	
car::system::logging::VectorSink< Mutex > . . . . .	106
car::system::device::CameraDevice . . . . .	33
car::display::console::CarConsole . . . . .	37
car::configuration::Configuration . . . . .	44
car::display::console::component::main::ConnectButton . . . . .	47
Context	
behaviour_tree::CarContext . . . . .	39
daemon	
rpi_daemon . . . . .	98
car::display::console::component::debug::DebugEnabler . . . . .	48
car::display::console::component::debug::DebugLidarCheckbox . . . . .	51
car::display::console::component::debug::DebugMessagingTextbox . . . . .	53
car::display::console::component::debug::DebugMovementRenderer . . . . .	55
car::system::device::DeviceManager . . . . .	62
std::enable_shared_from_this	
car::system::CarSystem . . . . .	40
car::system::messaging::MessagingSystem::FirstMessageStruct . . . . .	69
car::configuration::JsonConfiguration . . . . .	70
car::system::device::lidar::LidarDevice . . . . .	71
car::system::device::lidar::LidarDummy . . . . .	74
car::system::device::lidar::LidarScanner . . . . .	76
car::display::console::screen::LoggingScreen . . . . .	79
car::display::console::component::main::MainErrorModal . . . . .	81
car::display::console::component::main::MainExitModal . . . . .	82
car::display::console::screen::MainScreen . . . . .	83
car::system::messaging::MessagingSystem . . . . .	85
car::system::movement::MovementSystem . . . . .	91
car::plugin::Plugin . . . . .	95
behaviour_tree::BehaviourTreeHandler . . . . .	30
car::plugin::PluginManager . . . . .	96
car::display::console::component::settings::SettingsEditConfig . . . . .	100
car::display::console::screen::SettingsScreen . . . . .	102
TB6612 . . . . .	104



## Chapter 8

# Class Index

### 8.1 Class List

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

<a href="#">car::system::movement::controller::AbstractMovementController</a>	23
<a href="#">BackWheels</a>	27
<a href="#">behaviour_tree::BehaviourTreeHandler</a>	30
<a href="#">car::system::device::CameraDevice</a>	33
<a href="#">car::display::console::CarConsole</a>	37
<a href="#">behaviour_tree::CarContext</a>	39
<a href="#">car::system::CarSystem</a>	40
<a href="#">car::configuration::Configuration</a>	44
<a href="#">car::display::console::component::main::ConnectButton</a>	47
<a href="#">car::display::console::component::debug::DebugEnabler</a>	48
<a href="#">car::display::console::component::debug::DebugLidarCheckbox</a>	51
<a href="#">car::display::console::component::debug::DebugMessagingTextbox</a>	53
<a href="#">car::display::console::component::debug::DebugMovementRenderer</a>	55
<a href="#">car::system::device::DeviceManager</a>	62
<a href="#">car::system::movement::controller::DummyMovementController</a>	65
<a href="#">car::system::messaging::MessagingSystem::FirstMessageStruct</a>	69
<a href="#">car::configuration::JsonConfiguration</a>	70
<a href="#">car::system::device::lidar::LidarDevice</a>	71
<a href="#">car::system::device::lidar::LidarDummy</a>	74
<a href="#">car::system::device::lidar::LidarScanner</a>	76
<a href="#">car::display::console::screen::LoggingScreen</a>	79
<a href="#">car::display::console::component::main::MainErrorModal</a>	81
<a href="#">car::display::console::component::main::MainExitModal</a>	82
<a href="#">car::display::console::screen::MainScreen</a>	83
<a href="#">car::system::messaging::MessagingSystem</a>	85
<a href="#">car::system::movement::MovementSystem</a>	91
<a href="#">car::plugin::Plugin</a>	95
<a href="#">car::plugin::PluginManager</a>	96
<a href="#">rpi_daemon</a>	98
<a href="#">car::display::console::component::settings::SettingsEditConfig</a>	100
<a href="#">car::display::console::screen::SettingsScreen</a>	102
<a href="#">TB6612</a>	104
<a href="#">car::system::logging::VectorSink&lt; Mutex &gt;</a>	106



## Chapter 9

# File Index

### 9.1 File List

Here is a list of all files with brief descriptions:

behaviour_tree/src/main.cpp . . . . .	109
common/include/behaviour_tree/BehaviourTreeHandler.hpp . . . . .	112
common/include/behaviour_tree/CarContext.hpp . . . . .	115
common/include/car/configuration/Configuration.h . . . . .	116
common/include/car/plugin/Plugin.h . . . . .	116
common/include/car/plugin/PluginManager.h . . . . .	117
common/include/car/system/CarSystem.h . . . . .	119
common/include/car/system/device/CameraDevice.h . . . . .	120
common/include/car/system/device/DeviceManager.h . . . . .	121
common/include/car/system/device/lidar/LidarDevice.h . . . . .	123
common/include/car/system/device/lidar/LidarDummy.h . . . . .	124
common/include/car/system/device/lidar/LidarScanner.h . . . . .	124
common/include/car/system/logging/VectorSink.h . . . . .	126
common/include/car/system/messaging/MessagingSystem.h . . . . .	127
common/include/car/system/messaging/StreamType.h . . . . .	129
common/include/car/system/movement/MovementSystem.h . . . . .	134
common/include/car/system/movement/controller/AbstractMovementController.h . . . . .	130
common/include/car/system/movement/controller/DeviceMovementController.h . . . . .	131
common/include/car/system/movement/controller/DummyMovementController.h . . . . .	132
common/include/car/system/movement/devices/RearWheel.h . . . . .	133
common/include/car/system/movement/devices/Servo.h . . . . .	133
common/src/car/system/CarSystem.cpp . . . . .	136
common/src/car/system/device/CameraDevice.cpp . . . . .	136
common/src/car/system/device/DeviceManager.cpp . . . . .	136
common/src/car/system/messaging/MessagingSystem.cpp . . . . .	137
common/src/car/system/movement/controller/DeviceMovementController.cpp . . . . .	137
common/src/car/system/movement/controller/DummyMovementController.cpp . . . . .	137
common/src/car/system/movement/devices/RearWheel.cpp . . . . .	138
common/src/car/system/movement/devices/Servo.cpp . . . . .	138
common/tests/pca9685/test_front_wheels.cpp . . . . .	138
common/tests/tb6612/test_rear_wheels.cpp . . . . .	139
daemon/src/main.cpp . . . . .	110
repository/packages/t/tb6612/tb6612/include/TB6612.h . . . . .	140
repository/packages/t/tb6612/tb6612/src/TB6612.cpp . . . . .	141
tui/src/main.cpp . . . . .	111

tui/src/car/configuration/JsonConfiguration.cxx . . . . .	141
tui/src/car/display/console/CarConsole.cpp . . . . .	142
tui/src/car/display/console/CarConsole.h . . . . .	142
tui/src/car/display/console/component/debug/DebugEnabler.cxx . . . . .	143
tui/src/car/display/console/component/debug/DebugLidarCheckbox.cxx . . . . .	144
tui/src/car/display/console/component/debug/DebugMessagingTextbox.cxx . . . . .	145
tui/src/car/display/console/component/debug/DebugMovementRenderer.cxx . . . . .	145
tui/src/car/display/console/component/main/ConnectButton.cxx . . . . .	146
tui/src/car/display/console/component/main/MainErrorModal.cxx . . . . .	147
tui/src/car/display/console/component/main/MainExitModal.cxx . . . . .	148
tui/src/car/display/console/component/settings/SettingsEditConfig.cxx . . . . .	148
tui/src/car/display/console/screen/LoggingScreen.cxx . . . . .	149
tui/src/car/display/console/screen/MainScreen.cxx . . . . .	150
tui/src/car/display/console/screen/SettingsScreen.cxx . . . . .	151

## Chapter 10

# Namespace Documentation

### 10.1 behaviour\_tree Namespace Reference

#### Classes

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

### 10.2 car Namespace Reference

#### Namespaces

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

### 10.3 car::configuration Namespace Reference

#### Classes

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

### 10.4 car::display Namespace Reference

#### Namespaces

- namespace [console](#)

## 10.5 car::display::console Namespace Reference

### Namespaces

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

### Classes

- class [CarConsole](#)

## 10.6 car::display::console::component Namespace Reference

### Namespaces

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

## 10.7 car::display::console::component::debug Namespace Reference

### Classes

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

## 10.8 car::display::console::component::main Namespace Reference

### Classes

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

## 10.9 car::display::console::component::settings Namespace Reference

### Classes

- class [SettingsEditConfig](#)



## 10.10 car::display::console::screen Namespace Reference

### Classes

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

## 10.11 car::plugin Namespace Reference

### Classes

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

## 10.12 car::system Namespace Reference

### Namespaces

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

### Classes

- class [CarSystem](#)

## 10.13 car::system::device Namespace Reference

### Namespaces

- namespace [lidar](#)

### Classes

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

## 10.14 car::system::device::lidar Namespace Reference

### Classes

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

## 10.15 car::system::logging Namespace Reference

### Classes

- class [VectorSink](#)

### Typedefs

- using [vector\\_sink\\_mt](#) = [VectorSink](#)< std::mutex >

### 10.15.1 Typedef Documentation

#### 10.15.1.1 vector\_sink\_mt

using [car::system::logging::vector\\_sink\\_mt](#) = typedef [VectorSink](#)<std::mutex>

## 10.16 car::system::messaging Namespace Reference

### Classes

- class [MessagingSystem](#)

## 10.17 car::system::movement Namespace Reference

### Namespaces

- namespace [controller](#)

### Classes

- class [MovementSystem](#)

## 10.18 car::system::movement::controller Namespace Reference

### Classes

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

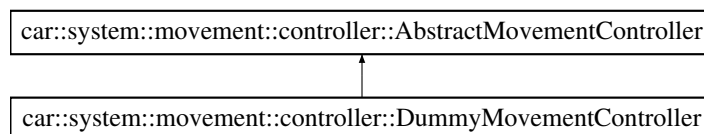
## Chapter 11

# Class Documentation

### 11.1 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

#### 11.1.1 Member Function Documentation

#### 11.1.1.1 initialize()

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

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

#### 11.1.1.2 setCameraServo1Angle()

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

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

#### 11.1.1.3 setCameraServo2Angle()

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

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

#### 11.1.1.4 setFrontWheelsAngle()

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

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

#### 11.1.1.5 setRearLeftWheelDirectionToBackward()

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

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

#### 11.1.1.6 setRearLeftWheelDirectionToForward()

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

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

#### 11.1.1.7 setRearLeftWheelSpeed()

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

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

#### 11.1.1.8 setRearRightWheelDirectionToBackward()

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

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

#### 11.1.1.9 setRearRightWheelDirectionToForward()

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

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

#### 11.1.1.10 setRearRightWheelSpeed()

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

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

#### 11.1.1.11 `setRearWheelsDirectionToBackward()`

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

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

#### 11.1.1.12 `setRearWheelsDirectionToForward()`

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

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

#### 11.1.1.13 `setRearWheelsSpeed()`

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

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

#### 11.1.1.14 `stop()`

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

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

#### 11.1.1.15 `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](#)

## 11.2 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](#)

## 11.2.1 Constructor & Destructor Documentation

### 11.2.1.1 BackWheels()

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

## 11.2.2 Member Function Documentation

### 11.2.2.1 backward()

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

#### 11.2.2.2 calibration()

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

#### 11.2.2.3 caliLeft()

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

#### 11.2.2.4 caliOK()

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

#### 11.2.2.5 caliRight()

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

#### 11.2.2.6 forward()

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

#### 11.2.2.7 getSpeed()

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

#### 11.2.2.8 ready()

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

#### 11.2.2.9 setSpeed()

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



### 11.2.2.10 stop()

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

## 11.2.3 Member Data Documentation

### 11.2.3.1 cali\_forward\_A

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

### 11.2.3.2 cali\_forward\_B

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

### 11.2.3.3 forward\_A

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

### 11.2.3.4 forward\_B

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

### 11.2.3.5 left\_wheel

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

### 11.2.3.6 pca9685

```
PCA9685 BackWheels::pca9685
```

### 11.2.3.7 right\_wheel

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

### 11.2.3.8 speed

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

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

- [common/tests/tb6612/test\\_rear\\_wheels.cpp](#)

## 11.3 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](#)

### 11.3.1 Member Function Documentation

#### 11.3.1.1 `_setBehaviourTree()`

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

#### 11.3.1.2 `getName()`

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

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

#### 11.3.1.3 `handleCommand()`

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

#### 11.3.1.4 `initialize()`

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

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

#### 11.3.1.5 `setBehaviourTree()`

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

#### 11.3.1.6 startBehaviourTree()

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

#### 11.3.1.7 stop()

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

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

#### 11.3.1.8 stopBehaviourTree()

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

#### 11.3.1.9 update()

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

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

### 11.3.2 Member Data Documentation

#### 11.3.2.1 behaviour\_tree

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

#### 11.3.2.2 car\_system

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

### 11.3.2.3 context

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

### 11.3.2.4 last\_connected

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

### 11.3.2.5 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](#)

## 11.4 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](#)

## 11.4.1 Constructor & Destructor Documentation

### 11.4.1.1 CameraDevice() [1/3]

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

### 11.4.1.2 CameraDevice() [2/3]

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

### 11.4.1.3 CameraDevice() [3/3]

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

### 11.4.1.4 ~CameraDevice()

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

## 11.4.2 Member Function Documentation

#### 11.4.2.1 create()

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

#### 11.4.2.2 disconnect()

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

#### 11.4.2.3 getFrameBuffer()

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

#### 11.4.2.4 operator=() [1/2]

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

#### 11.4.2.5 operator=() [2/2]

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

#### 11.4.2.6 start()

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

#### 11.4.2.7 stop()

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

#### 11.4.2.8 terminate()

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

#### 11.4.2.9 update()

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

### 11.4.3 Friends And Related Function Documentation

#### 11.4.3.1 DeviceManager

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

### 11.4.4 Member Data Documentation

#### 11.4.4.1 camera\_

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

#### 11.4.4.2 camera\_mutex\_

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

#### 11.4.4.3 configuration

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



#### 11.4.4.4 connected\_

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

#### 11.4.4.5 frame\_buffer\_

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

#### 11.4.4.6 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](#)

## 11.5 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

### 11.5.1 Constructor & Destructor Documentation

#### 11.5.1.1 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 )
```

### 11.5.2 Member Function Documentation

#### 11.5.2.1 initialize()

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

#### 11.5.2.2 run()

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

#### 11.5.2.3 terminate()

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

### 11.5.3 Member Data Documentation

#### 11.5.3.1 car\_system

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

#### 11.5.3.2 json\_configuration

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

### 11.5.3.3 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](#)

## 11.6 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](#)

### 11.6.1 Constructor & Destructor Documentation

#### 11.6.1.1 CarContext()

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

### 11.6.2 Member Function Documentation

### 11.6.2.1 \_()

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

### 11.6.2.2 getCarSystem()

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

## 11.6.3 Member Data Documentation

### 11.6.3.1 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](#)

## 11.7 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

## 11.7.1 Constructor & Destructor Documentation

### 11.7.1.1 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 )
```

## 11.7.2 Member Function Documentation

### 11.7.2.1 disconnect()

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

### 11.7.2.2 getConfiguration()

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

### 11.7.2.3 getDeviceManager()

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

### 11.7.2.4 getMessagingSystem()

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

### 11.7.2.5 getMovementSystem()

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

### 11.7.2.6 getPlugin()

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

### 11.7.2.7 initialize()

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

### 11.7.2.8 reload()

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

### 11.7.2.9 sendData()

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

#### 11.7.2.10 setConfiguration()

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

#### 11.7.2.11 start()

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

#### 11.7.2.12 stop()

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

#### 11.7.2.13 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.

#### 11.7.2.14 tryConnect()

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

#### 11.7.2.15 update()

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

### 11.7.3 Member Data Documentation

#### 11.7.3.1 configuration\_

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

### 11.7.3.2 device\_manager\_

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

### 11.7.3.3 initialized

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

### 11.7.3.4 messaging\_system\_

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

### 11.7.3.5 movement\_system\_

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

### 11.7.3.6 plugin\_manager\_

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

### 11.7.3.7 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](#)

## 11.8 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

### 11.8.1 Member Function Documentation

#### 11.8.1.1 [getCameraFpsInterval\(\)](#)

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

#### 11.8.1.2 [setCameraFps\(\)](#)

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

### 11.8.2 Member Data Documentation

#### 11.8.2.1 [behaviour\\_tree\\_update\\_ms\\_interval](#)

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

### 11.8.2.2 camera\_fps

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

### 11.8.2.3 camera\_fps\_interval

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

### 11.8.2.4 camera\_index

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

### 11.8.2.5 host

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

### 11.8.2.6 lidar\_port

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

### 11.8.2.7 use\_camera

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

### 11.8.2.8 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](#)

## 11.9 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](#)

### 11.9.1 Constructor & Destructor Documentation

#### 11.9.1.1 ConnectButton()

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

### 11.9.2 Member Function Documentation

#### 11.9.2.1 element()

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

### 11.9.3 Member Data Documentation

### 11.9.3.1 button\_pressed

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

### 11.9.3.2 car\_system

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

### 11.9.3.3 main\_button

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

### 11.9.3.4 main\_button\_text

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

### 11.9.3.5 main\_debounce

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

### 11.9.3.6 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](#)

## 11.10 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..."

## 11.10.1 Member Function Documentation

### 11.10.1.1 getCheckbox()

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

### 11.10.1.2 getWarningModal()

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

### 11.10.1.3 isEnabled()

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

## 11.10.2 Member Data Documentation

### 11.10.2.1 checkbox\_value

bool car::display::console::component::debug::DebugEnabler::checkbox\_value = false [private]

### 11.10.2.2 component

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

### 11.10.2.3 debounce

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

### 11.10.2.4 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]
```

### 11.10.2.5 DEBUG\_MODE\_DISABLED\_MESSAGE

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

### 11.10.2.6 DEBUG\_MODE\_ENABLED\_MESSAGE

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

### 11.10.2.7 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]
```

### 11.10.2.8 display\_warn\_debug\_modal

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

### 11.10.2.9 enabled

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

### 11.10.2.10 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

## 11.11 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"

### 11.11.1 Constructor & Destructor Documentation

#### 11.11.1.1 DebugLidarCheckbox()

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

## 11.11.2 Member Function Documentation

### 11.11.2.1 element()

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

### 11.11.2.2 getLidarMotorSignal()

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

## 11.11.3 Member Data Documentation

### 11.11.3.1 lidar\_motor\_checkbox\_component

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

### 11.11.3.2 LIDAR\_MOTOR\_DISABLED\_MESSAGE

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

### 11.11.3.3 lidar\_motor\_enabled

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

### 11.11.3.4 LIDAR\_MOTOR\_ENABLED\_MESSAGE

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



### 11.11.3.5 lidar\_motor\_loading\_debounce

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

### 11.11.3.6 lidar\_motor\_signal

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

### 11.11.3.7 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](#)

## 11.12 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](#)

### 11.12.1 Constructor & Destructor Documentation

### 11.12.1.1 DebugMessagingTextbox()

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

## 11.12.2 Member Function Documentation

### 11.12.2.1 element()

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

## 11.12.3 Member Data Documentation

### 11.12.3.1 message

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

### 11.12.3.2 message\_signal

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

### 11.12.3.3 messaging\_container

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

### 11.12.3.4 messaging\_textbox

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

### 11.12.3.5 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](#)

## 11.13 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"

## 11.13.1 Constructor & Destructor Documentation

### 11.13.1.1 DebugMovementRenderer()

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

## 11.13.2 Member Function Documentation

### 11.13.2.1 element()

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

### 11.13.2.2 getCameraServo1AngleSliderValue()

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

### 11.13.2.3 getCameraServo2AngleSliderValue()

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

#### 11.13.2.4 getFrontWheelsAngleSliderValue()

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

#### 11.13.2.5 getRearLeftWheelSpeedSliderValue()

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

#### 11.13.2.6 getRearRightWheelSpeedSliderValue()

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

#### 11.13.2.7 getRearWheelDirectionSignal()

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

#### 11.13.2.8 updateCameraServo1()

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

#### 11.13.2.9 updateCameraServo2()

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

#### 11.13.2.10 updateFrontWheels()

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

### 11.13.2.11 updateRearWheels()

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

## 11.13.3 Member Data Documentation

### 11.13.3.1 camera\_servo\_1\_angle\_slider

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

### 11.13.3.2 camera\_servo\_1\_angle\_slider\_angle

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

### 11.13.3.3 camera\_servo\_2\_angle\_slider

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

### 11.13.3.4 camera\_servo\_2\_angle\_slider\_angle

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

### 11.13.3.5 DEFAULT\_FRONT\_WHEEL\_ANGLE

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

### 11.13.3.6 DEFAULT\_REAR\_WHEEL\_SPEED

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

### 11.13.3.7 front\_wheels\_angle\_slider

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

### 11.13.3.8 front\_wheels\_angle\_slider\_value

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

### 11.13.3.9 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]
```

### 11.13.3.10 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]
```

### 11.13.3.11 previous\_front\_wheels\_angle\_slider\_value

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

### 11.13.3.12 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]
```

#### 11.13.3.13 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]
```

#### 11.13.3.14 previous\_rear\_wheels\_speed\_slider\_value

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

#### 11.13.3.15 rear\_left\_wheel\_speed\_slider

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

#### 11.13.3.16 rear\_left\_wheel\_speed\_slider\_value

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

#### 11.13.3.17 rear\_right\_wheel\_speed\_slider

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

#### 11.13.3.18 rear\_right\_wheel\_speed\_slider\_value

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

#### 11.13.3.19 rear\_wheel\_direction

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



#### 11.13.3.20 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]
```

#### 11.13.3.21 rear\_wheel\_direction\_checkbox\_component

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

#### 11.13.3.22 rear\_wheel\_direction\_debounce

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

#### 11.13.3.23 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]
```

#### 11.13.3.24 rear\_wheel\_direction\_signal

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

#### 11.13.3.25 rear\_wheel\_direction\_status

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

#### 11.13.3.26 rear\_wheel\_menu\_entry

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

### 11.13.3.27 rear\_wheel\_speed\_slider

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

### 11.13.3.28 rear\_wheels\_speed\_slider\_value

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

### 11.13.3.29 servo\_menu\_entry

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

### 11.13.3.30 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](#)

## 11.14 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\\_](#)

## 11.14.1 Constructor & Destructor Documentation

### 11.14.1.1 DeviceManager()

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

## 11.14.2 Member Function Documentation

### 11.14.2.1 create()

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

### 11.14.2.2 getCameraDevice()

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

### 11.14.2.3 getLidarDevice()

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

#### 11.14.2.4 initialize()

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

#### 11.14.2.5 isRunning()

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

#### 11.14.2.6 start()

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

#### 11.14.2.7 stop()

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

#### 11.14.2.8 terminate()

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

#### 11.14.2.9 update()

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

### 11.14.3 Member Data Documentation

#### 11.14.3.1 camera\_device\_

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

### 11.14.3.2 car\_system

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

### 11.14.3.3 is\_initialized\_

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

### 11.14.3.4 is\_running\_

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

### 11.14.3.5 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](#)

## 11.15 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

### 11.15.1 Member Function Documentation

#### 11.15.1.1 [initialize\(\)](#)

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

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

#### 11.15.1.2 [setCameraServo1Angle\(\)](#)

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

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

#### 11.15.1.3 [setCameraServo2Angle\(\)](#)

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

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

#### 11.15.1.4 setFrontWheelsAngle()

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

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

#### 11.15.1.5 setRearLeftWheelDirectionToBackward()

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

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

#### 11.15.1.6 setRearLeftWheelDirectionToForward()

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

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

#### 11.15.1.7 setRearLeftWheelSpeed()

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

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

#### 11.15.1.8 setRearRightWheelDirectionToBackward()

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

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

#### 11.15.1.9 setRearRightWheelDirectionToForward()

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

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

#### 11.15.1.10 `setRearRightWheelSpeed()`

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

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

#### 11.15.1.11 `setRearWheelsDirectionToBackward()`

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

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

#### 11.15.1.12 `setRearWheelsDirectionToForward()`

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

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

#### 11.15.1.13 `setRearWheelsSpeed()`

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

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

#### 11.15.1.14 `stop()`

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

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



### 11.15.1.15 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](#)

## 11.16 car::system::messaging::MessagingSystem::FirstMessageStruct Struct Reference

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

### Public Attributes

- std::string [error\\_message](#)
- std::string [uuid](#)
- std::condition\_variable [condition](#)

### 11.16.1 Member Data Documentation

#### 11.16.1.1 condition

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

#### 11.16.1.2 error\_message

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

#### 11.16.1.3 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](#)

## 11.17 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](#)

### 11.17.1 Constructor & Destructor Documentation

#### 11.17.1.1 JsonConfiguration()

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

### 11.17.2 Member Function Documentation

#### 11.17.2.1 getConfigFilePath()

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

#### 11.17.2.2 loadConfiguration()

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

#### 11.17.2.3 setConfigFilePath()

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

### 11.17.3 Member Data Documentation

#### 11.17.3.1 config\_file\_path

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

#### 11.17.3.2 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](#)

## 11.18 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](#)

## 11.18.1 Member Function Documentation

### 11.18.1.1 disconnect()

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

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

### 11.18.1.2 getScanData()

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

### 11.18.1.3 initialize()

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

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

### 11.18.1.4 setScanData()

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

### 11.18.1.5 start()

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

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

### 11.18.1.6 stop()

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

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

### 11.18.1.7 terminate()

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

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

### 11.18.1.8 update()

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

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

## 11.18.2 Friends And Related Function Documentation

### 11.18.2.1 DeviceManager

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

## 11.18.3 Member Data Documentation

### 11.18.3.1 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](#)

## 11.19 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

#### 11.19.1 Constructor & Destructor Documentation

##### 11.19.1.1 LidarDummy()

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

#### 11.19.2 Member Function Documentation

##### 11.19.2.1 disconnect()

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

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

### 11.19.2.2 initialize()

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

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

### 11.19.2.3 start()

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

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

### 11.19.2.4 stop()

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

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

### 11.19.2.5 terminate()

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

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

### 11.19.2.6 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](#)

## 11.20 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

#### 11.20.1 Constructor & Destructor Documentation

##### 11.20.1.1 LidarScanner()

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



## 11.20.2 Member Function Documentation

### 11.20.2.1 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]
```

### 11.20.2.2 disconnect()

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

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

### 11.20.2.3 initialize()

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

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

### 11.20.2.4 start()

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

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

### 11.20.2.5 stop()

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

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

### 11.20.2.6 terminate()

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

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

### 11.20.2.7 update()

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

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

## 11.20.3 Member Data Documentation

### 11.20.3.1 configuration\_

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

### 11.20.3.2 lidar\_

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

### 11.20.3.3 running

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

### 11.20.3.4 scan\_data\_

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

### 11.20.3.5 scan\_data\_mutex\_

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

### 11.20.3.6 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](#)

## 11.21 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](#)

### 11.21.1 Constructor & Destructor Documentation

#### 11.21.1.1 LoggingScreen()

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

### 11.21.2 Member Function Documentation

### 11.21.2.1 element()

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

## 11.21.3 Member Data Documentation

### 11.21.3.1 line\_elements

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

### 11.21.3.2 menu

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

### 11.21.3.3 my\_custom\_menu

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

### 11.21.3.4 selected\_line

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

### 11.21.3.5 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](#)

## 11.22 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](#)

### 11.22.1 Constructor & Destructor Documentation

#### 11.22.1.1 MainErrorModal()

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

### 11.22.2 Member Function Documentation

#### 11.22.2.1 element()

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

#### 11.22.2.2 setErrorMessage()

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

### 11.22.3 Member Data Documentation

### 11.22.3.1 error\_element

Element car::display::console::component::main::MainErrorModal::error\_element [private]

### 11.22.3.2 error\_modal\_shown

bool car::display::console::component::main::MainErrorModal::error\_modal\_shown = false

### 11.22.3.3 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](#)

## 11.23 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](#)

## 11.23.1 Constructor & Destructor Documentation

### 11.23.1.1 MainExitModal()

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

## 11.23.2 Member Function Documentation

### 11.23.2.1 element()

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

## 11.23.3 Member Data Documentation

### 11.23.3.1 exit

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

### 11.23.3.2 exit\_modal\_shown

bool car::display::console::component::main::MainExitModal::exit\_modal\_shown = false

### 11.23.3.3 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](#)

## 11.24 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`

## 11.24.1 Constructor & Destructor Documentation

### 11.24.1.1 MainScreen()

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

## 11.24.2 Member Function Documentation

### 11.24.2.1 element()

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

## 11.24.3 Member Data Documentation

### 11.24.3.1 box

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

### 11.24.3.2 car\_system

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



### 11.24.3.3 connect\_button

`ConnectButton` car::display::console::screen::MainScreen::connect\_button [private]

### 11.24.3.4 info

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

### 11.24.3.5 main\_component

`Component` car::display::console::screen::MainScreen::main\_component [private]

### 11.24.3.6 main\_error\_modal

`MainErrorModal` car::display::console::screen::MainScreen::main\_error\_modal [private]

### 11.24.3.7 main\_exit\_modal

`MainExitModal` car::display::console::screen::MainScreen::main\_exit\_modal [private]

### 11.24.3.8 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](#)

## 11.25 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

## 11.25.1 Constructor & Destructor Documentation

### 11.25.1.1 MessagingSystem()

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

## 11.25.2 Member Function Documentation

### 11.25.2.1 getCommandSignal()

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

### 11.25.2.2 getDisconnectSignal()

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

### 11.25.2.3 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>

### 11.25.2.4 getMessageSignal()

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

### 11.25.2.5 getSelectionSignal()

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

#### 11.25.2.6 getUUID()

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

#### 11.25.2.7 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>	
----------------	--

#### 11.25.2.8 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>	
----------------------	--

#### 11.25.2.9 initializeWebSocket()

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

Creates a new WebSocket object for use.

#### 11.25.2.10 isConnected()

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

#### 11.25.2.11 onDisconnect()

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

#### 11.25.2.12 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>	

#### 11.25.2.13 onMessageCallback()

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

#### 11.25.2.14 sendMessage()

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

#### 11.25.2.15 setConfiguration()

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

#### 11.25.2.16 stop()

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

**11.25.2.17 terminate()**

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

**11.25.2.18 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>
```

**11.25.3 Member Data Documentation****11.25.3.1 command\_signal\_**

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

**11.25.3.2 configuration\_**

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

**11.25.3.3 connected\_**

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

**11.25.3.4 message\_signal\_**

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

### 11.25.3.5 on\_disconnect\_signal\_

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

### 11.25.3.6 selection\_signal\_

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

### 11.25.3.7 uuid\_

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

### 11.25.3.8 websocket\_

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

### 11.25.3.9 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](#)

## 11.26 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](#)

## 11.26.1 Constructor & Destructor Documentation

### 11.26.1.1 MovementSystem()

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

### 11.26.1.2 ~MovementSystem()

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

## 11.26.2 Member Function Documentation

### 11.26.2.1 initialize()

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



### 11.26.2.2 setCameraServo1Angle()

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

### 11.26.2.3 setCameraServo2Angle()

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

### 11.26.2.4 setFrontWheelsAngle()

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

### 11.26.2.5 setRearLeftWheelDirectionToBackward()

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

### 11.26.2.6 setRearLeftWheelDirectionToForward()

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

### 11.26.2.7 setRearLeftWheelSpeed()

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

### 11.26.2.8 setRearRightWheelDirectionToBackward()

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

#### 11.26.2.9 setRearRightWheelDirectionToForward()

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

#### 11.26.2.10 setRearRightWheelSpeed()

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

#### 11.26.2.11 setRearWheelsDirectionToBackward()

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

#### 11.26.2.12 setRearWheelsDirectionToForward()

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

#### 11.26.2.13 setRearWheelsSpeed()

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

#### 11.26.2.14 start()

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

#### 11.26.2.15 stop()

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

### 11.26.2.16 terminate()

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

## 11.26.3 Member Data Documentation

### 11.26.3.1 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](#)

## 11.27 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

### 11.27.1 Member Function Documentation

#### 11.27.1.1 getName()

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

Implemented in [behaviour\\_tree::BehaviourTreeHandler](#).

### 11.27.1.2 initialize()

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

Implemented in [behaviour\\_tree::BehaviourTreeHandler](#).

### 11.27.1.3 stop()

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

Implemented in [behaviour\\_tree::BehaviourTreeHandler](#).

### 11.27.1.4 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](#)

## 11.28 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](#)

## 11.28.1 Member Function Documentation

### 11.28.1.1 addPlugin()

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

### 11.28.1.2 getPlugin()

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

### 11.28.1.3 initialize()

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

### 11.28.1.4 stop()

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

### 11.28.1.5 terminate()

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

### 11.28.1.6 update()

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

## 11.28.2 Member Data Documentation

### 11.28.2.1 plugins

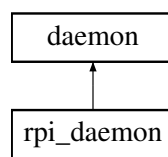
```
std::vector<std::shared_ptr<Plugin> > car::plugin::PluginManager::plugins [private]
```

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

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

## 11.29 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](#)

## 11.29.1 Member Function Documentation

### 11.29.1.1 connect()

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

### 11.29.1.2 on\_reload()

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

### 11.29.1.3 on\_start()

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

### 11.29.1.4 on\_stop()

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

### 11.29.1.5 on\_update()

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

### 11.29.1.6 update()

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

## 11.29.2 Member Data Documentation

### 11.29.2.1 any\_configuration\_empty

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

### 11.29.2.2 attempted\_to\_reconnect

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

### 11.29.2.3 car\_system

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

### 11.29.2.4 connection\_ms\_interval

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

### 11.29.2.5 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

## 11.30 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](#)

### 11.30.1 Constructor & Destructor Documentation



### 11.30.1.1 SettingsEditConfig()

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

## 11.30.2 Member Function Documentation

### 11.30.2.1 element()

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

## 11.30.3 Member Data Documentation

### 11.30.3.1 car\_system

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

### 11.30.3.2 input\_settings\_file\_path

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

### 11.30.3.3 json\_configuration

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

### 11.30.3.4 load\_button

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

### 11.30.3.5 placeholder

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

### 11.30.3.6 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](#)

## 11.31 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

## 11.31.1 Constructor & Destructor Documentation

### 11.31.1.1 SettingsScreen()

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

## 11.31.2 Member Function Documentation

### 11.31.2.1 element()

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

### 11.31.2.2 update()

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

## 11.31.3 Member Data Documentation

### 11.31.3.1 car\_system

std::shared\_ptr<CarSystem> car::display::console::screen::SettingsScreen::car\_system [private]

### 11.31.3.2 debug\_enabler

DebugEnabledler car::display::console::screen::SettingsScreen::debug\_enabler [private]

### 11.31.3.3 debug\_lidar\_checkbox

DebugLidarCheckbox car::display::console::screen::SettingsScreen::debug\_lidar\_checkbox [private]

### 11.31.3.4 debug\_messaging\_text\_box

DebugMessagingTextbox car::display::console::screen::SettingsScreen::debug\_messaging\_text\_box  
[private]

### 11.31.3.5 debug\_movement\_renderer

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

### 11.31.3.6 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](#)

## 11.32 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

### 11.32.1 Constructor & Destructor Documentation

#### 11.32.1.1 TB6612()

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

## 11.32.2 Member Function Documentation

### 11.32.2.1 backward()

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

### 11.32.2.2 forward()

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

### 11.32.2.3 getMotorPin()

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

### 11.32.2.4 getPWMPin()

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

### 11.32.2.5 setOffset()

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

### 11.32.2.6 setPWM()

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

### 11.32.2.7 stop()

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

### 11.32.3 Member Data Documentation

#### 11.32.3.1 motor\_pin

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

#### 11.32.3.2 offset

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

#### 11.32.3.3 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

## 11.33 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](#)

## 11.33.1 Constructor & Destructor Documentation

### 11.33.1.1 VectorSink()

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

## 11.33.2 Member Function Documentation

### 11.33.2.1 flush\_()

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

### 11.33.2.2 get\_log\_messages()

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

### 11.33.2.3 sink\_it\_()

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

## 11.33.3 Member Data Documentation

#### 11.33.3.1 log\_messages

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

#### 11.33.3.2 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 12

# File Documentation

### 12.1 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[])

#### 12.1.1 Function Documentation

##### 12.1.1.1 kbhit()

```
int kbhit (
    void )
```

### 12.1.1.2 main()

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

## 12.2 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[])`

### 12.2.1 Function Documentation

#### 12.2.1.1 getLidarDevice()

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

### 12.2.1.2 getMovementController()

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

### 12.2.1.3 main()

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

### 12.2.1.4 terminate\_handler()

```
void terminate_handler ( )
```

## 12.3 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[])

### 12.3.1 Function Documentation

### 12.3.1.1 `getLidarDevice()`

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

### 12.3.1.2 `getMovementController()`

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

### 12.3.1.3 `main()`

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

## 12.4 `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](#)

## 12.5 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

```

## 12.6 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`

## 12.7 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

```

## 12.8 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](#)

## 12.9 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
```

## 12.10 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](#)

## 12.11 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
```

## 12.12 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](#)

## 12.13 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

```

## 12.14 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](#)

## 12.15 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

```

## 12.16 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](#)

## 12.17 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) :
            configuration(configuration) {}
20
21         CameraDevice(const CameraDevice&) = delete;
22         CameraDevice& operator=(const CameraDevice&) = delete;
23
24         CameraDevice(CameraDevice&&) = delete;
25         CameraDevice& operator=(CameraDevice&&) = delete;
26
27         ~CameraDevice() = default;
28
29     public:
30         [[nodiscard]] static tl::expected<std::unique_ptr<CameraDevice>, std::string>
            create(std::shared_ptr<configuration::Configuration> configuration);
31         std::string getFrameBuffer() const;
32
33     protected:
34         void start();
35         void update();
36         void stop();
37         void disconnect();
38         void terminate();
39
40         friend class DeviceManager;
41
42     private:
43         std::shared_ptr<configuration::Configuration> configuration;
44
45         std::unique_ptr<cv::VideoCapture> camera_;
46
47         bool connected_ = false;
48         std::string frame_buffer_;
49
50         std::mutex camera_mutex_;
51
52         std::chrono::steady_clock::time_point last;
53     };
54 }
55
56 #endif

```

## 12.18 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`

## 12.19 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

```

## 12.20 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](#)

## 12.21 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
```

## 12.22 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](#)

## 12.23 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
```

## 12.24 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`

## 12.25 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

```

## 12.26 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 >`

## 12.27 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         std::vector<std::string> log_messages;
53     };
54     using vector_sink_mt = VectorSink<std::mutex>;
55 }
56 #endif

```

## 12.28 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](#)

## 12.29 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

```

## 12.30 common/include/car/system/messaging/StreamType.h File Reference

### Enumerations

- enum [StreamType](#) { [None](#) = 0 , [Lidar](#) , [Camera](#) , [Both](#) }

### 12.30.1 Enumeration Type Documentation

#### 12.30.1.1 StreamType

enum [StreamType](#)

##### Enumerator

None	
Lidar	
Camera	
Both	

## 12.31 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
```

## 12.32 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](#)

## 12.33 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
```

## 12.34 common/include/car/system/movement/controller/DeviceMovementController.h File Reference

### 12.35 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__

```

## 12.36 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](#)

## 12.37 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

```



## 12.38 common/include/car/system/movement/devices/RearWheel.h File Reference

### 12.39 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

```

## 12.40 common/include/car/system/movement/devices/Servo.h File Reference

### 12.41 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__

```

## 12.42 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](#)

## 12.43 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
```

## 12.44 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](#)

## 12.45 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](#)

## 12.46 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](#)

## 12.47 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](#)

## 12.48 common/src/car/system/movement/controller/DeviceMovementController.cpp File Reference ↩

## 12.49 common/src/car/system/movement/controller/DummyMovementController.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](#)

## 12.50 common/src/car/system/movement/devices/RearWheel.cpp File Reference

## 12.51 common/src/car/system/movement/devices/Servo.cpp File Reference

## 12.52 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

### 12.52.1 Function Documentation

#### 12.52.1.1 main()

```
int main ( )
```

#### 12.52.1.2 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

### 12.52.1.3 `setAngle()`

```
int setAngle (
    int & angle,
    PCA9685 pwm,
    int channel )
```

### 12.52.1.4 `setAngleToAnalog()`

```
int setAngleToAnalog (
    int angle )
```

## 12.52.2 Variable Documentation

### 12.52.2.1 `offset`

```
int offset = 0
```

## 12.53 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](#) ()

### 12.53.1 Function Documentation

### 12.53.1.1 main()

```
int main ( )
```

### 12.53.1.2 test()

```
void test ( )
```

## 12.54 daemon/install/README.md File Reference

## 12.55 daemon/README.md File Reference

## 12.56 README.md File Reference

## 12.57 tui/README.md File Reference

## 12.58 repository/packages/t/tb6612/tb6612/include/TB6612.h File Reference

```
#include "pigpio.h"  
#include "pigpiod_if2.h"
```

### Classes

- class [TB6612](#)

## 12.59 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
```

## 12.60 repository/packages/t/tb6612/tb6612/src/TB6612.cpp File Reference

```
#include "TB6612.h"
```

## 12.61 SETUP.md File Reference

## 12.62 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](#)

## 12.62.1 Macro Definition Documentation

### 12.62.1.1 JSONCONFIGURATION\_CXX

```
#define JSONCONFIGURATION_CXX
```

## 12.63 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](#)

## 12.64 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](#)

## 12.65 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

```

## 12.66 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](#)

## 12.66.1 Macro Definition Documentation

### 12.66.1.1 DEBUGENABLER\_CXX

```
#define DEBUGENABLER_CXX
```

## 12.67 [tui/src/car/display/console/component/debug/DebugLidar](#) Checkbox.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](#)

## 12.67.1 Macro Definition Documentation

### 12.67.1.1 DEBUGLIDARCHECKBOX\_CXX

```
#define DEBUGLIDARCHECKBOX_CXX
```

## 12.68 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](#)

### 12.68.1 Macro Definition Documentation

#### 12.68.1.1 DEBUGMESSAGINGTEXTBOX\_CXX

```
#define DEBUGMESSAGINGTEXTBOX_CXX
```

## 12.69 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](#)

### 12.69.1 Macro Definition Documentation

#### 12.69.1.1 DEBUGMOVEMENTRENDERER\_CXX

```
#define DEBUGMOVEMENTRENDERER_CXX
```

## 12.70 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](#)

## 12.70.1 Macro Definition Documentation

### 12.70.1.1 CONNECTBUTTON\_CXX

```
#define CONNECTBUTTON_CXX
```

## 12.71 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](#)

## 12.71.1 Macro Definition Documentation

### 12.71.1.1 MAINERRORMODAL\_CXX

```
#define MAINERRORMODAL_CXX
```

## 12.72 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](#)

### 12.72.1 Macro Definition Documentation

#### 12.72.1.1 MAINEXITMODAL\_CXX

```
#define MAINEXITMODAL_CXX
```

## 12.73 tui/src/car/display/console/component/settings/SettingsEdit↵ Config.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](#)

### 12.73.1 Macro Definition Documentation

#### 12.73.1.1 SETTINGSEDTCONFIG\_CXX

```
#define SETTINGSEDTCONFIG_CXX
```

## 12.74 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](#)

## 12.74.1 Macro Definition Documentation

### 12.74.1.1 LOGGINGSCREEN\_CXX

```
#define LOGGINGSCREEN_CXX
```

## 12.75 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](#)

## 12.75.1 Macro Definition Documentation

### 12.75.1.1 MAINSCREEN\_CXX

```
#define MAINSCREEN_CXX
```

## 12.76 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](#)

#### 12.76.1 Macro Definition Documentation

##### 12.76.1.1 SETTINGSSCREEN\_CXX

```
#define SETTINGSSCREEN_CXX
```



# Index

- - behaviour\_tree::CarContext, 39
- \_setBehaviourTree
  - behaviour\_tree::BehaviourTreeHandler, 31
- ~CameraDevice
  - car::system::device::CameraDevice, 34
- ~MovementSystem
  - car::system::movement::MovementSystem, 92
- addPlugin
  - car::plugin::PluginManager, 97
- any\_configuration\_empty
  - rpi\_daemon, 99
- attempted\_to\_reconnect
  - rpi\_daemon, 99
- backward
  - BackWheels, 27
  - TB6612, 105
- BackWheels, 27
  - backward, 27
  - BackWheels, 27
  - cali\_forward\_A, 29
  - cali\_forward\_B, 29
  - calibration, 27
  - caliLeft, 28
  - caliOK, 28
  - caliRight, 28
  - forward, 28
  - forward\_A, 29
  - forward\_B, 29
  - getSpeed, 28
  - left\_wheel, 29
  - pca9685, 29
  - ready, 28
  - right\_wheel, 29
  - setSpeed, 28
  - speed, 30
  - stop, 28
- behaviour\_tree, 19
  - behaviour\_tree::BehaviourTreeHandler, 32
- behaviour\_tree/src/main.cpp, 109
- behaviour\_tree::BehaviourTreeHandler, 30
  - \_setBehaviourTree, 31
  - behaviour\_tree, 32
  - car\_system, 32
  - context, 32
  - getName, 31
  - handleCommand, 31
  - initialize, 31
  - last\_connected, 33
  - setBehaviourTree, 31
  - startBehaviourTree, 31
  - stop, 32
  - stopBehaviourTree, 32
  - tick\_count, 33
  - update, 32
- behaviour\_tree::CarContext, 39
  - \_, 39
  - car\_system, 40
  - CarContext, 39
  - getCarSystem, 40
- behaviour\_tree\_update\_ms\_interval
  - car::configuration::Configuration, 45
- Both
  - StreamType.h, 129
- box
  - car::display::console::screen::MainScreen, 84
- button\_pressed
  - car::display::console::component::main::ConnectButton, 47
- cali\_forward\_A
  - BackWheels, 29
- cali\_forward\_B
  - BackWheels, 29
- calibration
  - BackWheels, 27
- caliLeft
  - BackWheels, 28
- caliOK
  - BackWheels, 28
- caliRight
  - BackWheels, 28
- Camera
  - StreamType.h, 129
- camera\_
  - car::system::device::CameraDevice, 36
- camera\_device\_
  - car::system::device::DeviceManager, 64
- camera\_fps
  - car::configuration::Configuration, 45
- camera\_fps\_interval
  - car::configuration::Configuration, 46
- camera\_index
  - car::configuration::Configuration, 46
- camera\_mutex\_
  - car::system::device::CameraDevice, 36
- camera\_servo\_1\_angle\_slider

- car::display::console::component::debug::DebugMovementRenderer, 58
- camera\_servo\_1\_angle\_slider\_angle
  - car::display::console::component::debug::DebugMovementRenderer, 58
- camera\_servo\_2\_angle\_slider
  - car::display::console::component::debug::DebugMovementRenderer, 58
- camera\_servo\_2\_angle\_slider\_angle
  - car::display::console::component::debug::DebugMovementRenderer, 58
- CameraDevice
  - car::system::device::CameraDevice, 34
- car, 19
- car::configuration, 19
- car::configuration::Configuration, 44
  - behaviour\_tree\_update\_ms\_interval, 45
  - camera\_fps, 45
  - camera\_fps\_interval, 46
  - camera\_index, 46
  - getCameraFpsInterval, 45
  - host, 46
  - lidar\_port, 46
  - setCameraFps, 45
  - use\_camera, 46
  - use\_lidar, 46
- car::configuration::JsonConfiguration, 70
  - config\_file\_path, 71
  - exe\_dir, 71
  - getConfigFilePath, 70
  - JsonConfiguration, 70
  - loadConfiguration, 70
  - setConfigFilePath, 70
- car::display, 19
- car::display::console, 20
- car::display::console::CarConsole, 37
  - car\_system, 38
  - CarConsole, 37
  - initialize, 38
  - json\_configuration, 38
  - run, 38
  - terminate, 38
  - vector\_sink, 38
- car::display::console::component, 20
- car::display::console::component::debug, 20
- car::display::console::component::debug::DebugEnabler, 48
  - checkbox\_value, 49
  - component, 49
  - debounce, 50
  - DEBUG\_ENABLE\_WARNING\_MESSAGE, 50
  - DEBUG\_MODE\_DISABLED\_MESSAGE, 50
  - DEBUG\_MODE\_ENABLED\_MESSAGE, 50
  - DEBUG\_MODE\_WAIT\_MESSAGE, 50
  - display\_warn\_debug\_modal, 50
  - enabled, 50
  - getCheckbox, 49
  - getWarningModal, 49
- car::display::console::component::debug::DebugMovementRenderer, 58
  - is\_enabled, 49
  - status, 51
- car::display::console::component::debug::DebugLidarCheckbox, 51
  - DebugLidarCheckbox, 51
  - element, 52
  - lidar\_motor\_signal, 52
  - lidar\_motor\_checkbox\_component, 52
  - LIDAR\_MOTOR\_DISABLED\_MESSAGE, 52
  - lidar\_motor\_enabled, 52
  - LIDAR\_MOTOR\_ENABLED\_MESSAGE, 52
  - lidar\_motor\_loading\_debounce, 52
  - lidar\_motor\_signal, 53
  - lidar\_motor\_status, 53
- car::display::console::component::debug::DebugMessagingTextbox, 53
  - DebugMessagingTextbox, 53
  - element, 54
  - message, 54
  - message\_signal, 54
  - messaging\_container, 54
  - messaging\_textbox, 54
  - messaging\_title, 54
- car::display::console::component::debug::DebugMovementRenderer, 55
  - camera\_servo\_1\_angle\_slider, 58
  - camera\_servo\_1\_angle\_slider\_angle, 58
  - camera\_servo\_2\_angle\_slider, 58
  - camera\_servo\_2\_angle\_slider\_angle, 58
  - DebugMovementRenderer, 56
  - DEFAULT\_FRONT\_WHEEL\_ANGLE, 58
  - DEFAULT\_REAR\_WHEEL\_SPEED, 58
  - element, 56
  - front\_wheels\_angle\_slider, 59
  - front\_wheels\_angle\_slider\_value, 59
  - getCameraServo1AngleSliderValue, 56
  - getCameraServo2AngleSliderValue, 56
  - getFrontWheelsAngleSliderValue, 56
  - getRearLeftWheelSpeedSliderValue, 57
  - getRearRightWheelSpeedSliderValue, 57
  - getRearWheelDirectionSignal, 57
  - previous\_camera\_servo\_1\_angle\_slider\_angle, 59
  - previous\_camera\_servo\_2\_angle\_slider\_angle, 59
  - previous\_front\_wheels\_angle\_slider\_value, 59
  - previous\_rear\_left\_wheel\_speed\_slider\_value, 59
  - previous\_rear\_right\_wheel\_speed\_slider\_value, 59
  - previous\_rear\_wheels\_speed\_slider\_value, 60
  - rear\_left\_wheel\_speed\_slider, 60
  - rear\_left\_wheel\_speed\_slider\_value, 60
  - rear\_right\_wheel\_speed\_slider, 60
  - rear\_right\_wheel\_speed\_slider\_value, 60
  - rear\_wheel\_direction, 60
  - REAR\_WHEEL\_DIRECTION\_BACKWARD\_MESSAGE, 60
  - rear\_wheel\_direction\_checkbox\_component, 61
  - rear\_wheel\_direction\_debounce, 61

- REAR\_WHEEL\_DIRECTION\_FORWARD\_MESSAGE, 61
- rear\_wheel\_direction\_signal, 61
- rear\_wheel\_direction\_status, 61
- rear\_wheel\_menu\_entry, 61
- rear\_wheel\_speed\_slider, 61
- rear\_wheels\_speed\_slider\_value, 62
- servo\_menu\_entry, 62
- slider\_container, 62
- updateCameraServo1, 57
- updateCameraServo2, 57
- updateFrontWheels, 57
- updateRearWheels, 57
- car::display::console::component::main, 20
- car::display::console::component::main::ConnectButton, 47
  - button\_pressed, 47
  - car\_system, 48
  - ConnectButton, 47
  - element, 47
  - main\_button, 48
  - main\_button\_text, 48
  - main\_debounce, 48
  - on\_connect\_failure, 48
- car::display::console::component::main::MainErrorModal, 81
  - element, 81
  - error\_element, 81
  - error\_modal\_shown, 82
  - main\_error\_modal, 82
  - MainErrorModal, 81
  - setErrorMessage, 81
- car::display::console::component::main::MainExitModal, 82
  - element, 83
  - exit, 83
  - exit\_modal\_shown, 83
  - main\_exit\_modal, 83
  - MainExitModal, 82
- car::display::console::component::settings, 20
- car::display::console::component::settings::SettingsEditConfig, 100
  - car\_system, 101
  - element, 101
  - input\_settings\_file\_path, 101
  - json\_configuration, 101
  - load\_button, 101
  - placeholder, 101
  - settings\_file\_path, 102
  - SettingsEditConfig, 100
- car::display::console::screen, 21
- car::display::console::screen::LoggingScreen, 79
  - element, 79
  - line\_elements, 80
  - LoggingScreen, 79
  - menu, 80
  - my\_custom\_menu, 80
  - selected\_line, 80
  - vector\_sink, 80
- car::display::console::screen::MainScreen, 83
  - box, 84
  - car\_system, 84
  - connect\_button, 84
  - element, 84
  - info, 85
  - main\_component, 85
  - main\_error\_modal, 85
  - main\_exit\_modal, 85
  - main\_screen, 85
  - MainScreen, 84
- car::display::console::screen::SettingsScreen, 102
  - car\_system, 103
  - debug\_enabler, 103
  - debug\_lidar\_checkbox, 103
  - debug\_messaging\_text\_box, 103
  - debug\_movement\_renderer, 103
  - element, 103
  - settings\_edit\_config, 104
  - SettingsScreen, 102
  - update, 103
- car::plugin, 21
- car::plugin::Plugin, 95
  - getName, 95
  - initialize, 95
  - stop, 96
  - update, 96
- car::plugin::PluginManager, 96
  - addPlugin, 97
  - getPlugin, 97
  - initialize, 97
  - plugins, 97
  - stop, 97
  - terminate, 97
  - update, 97
- car::system, 21
- car::system::CarSystem, 40
  - CarSystem, 41
  - configuration\_, 43
  - device\_manager\_, 43
  - disconnect, 41
  - getConfiguration, 41
  - getDeviceManager, 41
  - getMessagingSystem, 42
  - getMovementSystem, 42
  - getPlugin, 42
  - initialize, 42
  - initialized, 44
  - messaging\_system\_, 44
  - movement\_system\_, 44
  - plugin\_manager\_, 44
  - reload, 42
  - sendData, 42
  - setConfiguration, 42
  - start, 43
  - started, 44
  - stop, 43

- terminate, 43
  - tryConnect, 43
  - update, 43
- car::system::device, 21
- car::system::device::CameraDevice, 33
  - ~CameraDevice, 34
  - camera\_, 36
  - camera\_mutex\_, 36
  - CameraDevice, 34
  - configuration, 36
  - connected\_, 36
  - create, 34
  - DeviceManager, 36
  - disconnect, 35
  - frame\_buffer\_, 37
  - getFrameBuffer, 35
  - last, 37
  - operator=, 35
  - start, 35
  - stop, 35
  - terminate, 35
  - update, 36
- car::system::device::DeviceManager, 62
  - camera\_device\_, 64
  - car\_system, 64
  - create, 63
  - DeviceManager, 63
  - getCameraDevice, 63
  - getLidarDevice, 63
  - initialize, 63
  - is\_initialized\_, 65
  - is\_running\_, 65
  - isRunning, 64
  - lidar\_device\_, 65
  - start, 64
  - stop, 64
  - terminate, 64
  - update, 64
- car::system::device::lidar, 21
- car::system::device::lidar::LidarDevice, 71
  - DeviceManager, 73
  - disconnect, 72
  - getScanData, 72
  - initialize, 72
  - scan\_data\_, 73
  - setScanData, 72
  - start, 72
  - stop, 72
  - terminate, 73
  - update, 73
- car::system::device::lidar::LidarDummy, 74
  - disconnect, 74
  - initialize, 74
  - LidarDummy, 74
  - start, 75
  - stop, 75
  - terminate, 75
  - update, 75
- car::system::device::lidar::LidarScanner, 76
  - configuration\_, 78
  - create, 77
  - disconnect, 77
  - initialize, 77
  - lidar\_, 78
  - LidarScanner, 76
  - running, 78
  - scan\_data\_, 78
  - scan\_data\_mutex\_, 78
  - scan\_generator\_, 79
  - start, 77
  - stop, 77
  - terminate, 77
  - update, 78
- car::system::logging, 22
  - vector\_sink\_mt, 22
- car::system::logging::VectorSink< Mutex >, 106
  - flush\_, 107
  - get\_log\_messages, 107
  - log\_messages, 107
  - max\_lines, 108
  - sink\_it\_, 107
  - VectorSink, 107
- car::system::messaging, 22
- car::system::messaging::MessagingSystem, 85
  - command\_signal\_, 90
  - configuration\_, 90
  - connected\_, 90
  - getCommandSignal, 87
  - getDisconnectSignal, 87
  - getFirstMessage, 87
  - getMessageSignal, 87
  - getSelectionSignal, 87
  - getUUID, 87
  - handleMessage, 88
  - initialize, 88
  - initializeWebSocket, 88
  - isConnected, 88
  - message\_signal\_, 90
  - MessagingSystem, 86
  - on\_disconnect\_signal\_, 90
  - onDisconnect, 88
  - onFirstMessage, 89
  - onMessageCallback, 89
  - selection\_signal\_, 91
  - sendMessage, 89
  - setConfiguration, 89
  - stop, 89
  - terminate, 89
  - tryConnect, 90
  - uuid\_, 91
  - websocket\_, 91
  - websocket\_url\_, 91
- car::system::messaging::MessagingSystem::FirstMessageStruct, 69
  - condition, 69
  - error\_message, 69



- uuid, 69
- car::system::movement, 22
- car::system::movement::controller, 22
- car::system::movement::controller::AbstractMovementController, 23
  - initialize, 23
  - setCameraServo1Angle, 24
  - setCameraServo2Angle, 24
  - setFrontWheelsAngle, 24
  - setRearLeftWheelDirectionToBackward, 24
  - setRearLeftWheelDirectionToForward, 24
  - setRearLeftWheelSpeed, 25
  - setRearRightWheelDirectionToBackward, 25
  - setRearRightWheelDirectionToForward, 25
  - setRearRightWheelSpeed, 25
  - setRearWheelsDirectionToBackward, 25
  - setRearWheelsDirectionToForward, 26
  - setRearWheelsSpeed, 26
  - stop, 26
  - terminate, 26
- car::system::movement::controller::DummyMovementController, 65
  - initialize, 66
  - setCameraServo1Angle, 66
  - setCameraServo2Angle, 66
  - setFrontWheelsAngle, 66
  - setRearLeftWheelDirectionToBackward, 67
  - setRearLeftWheelDirectionToForward, 67
  - setRearLeftWheelSpeed, 67
  - setRearRightWheelDirectionToBackward, 67
  - setRearRightWheelDirectionToForward, 67
  - setRearRightWheelSpeed, 67
  - setRearWheelsDirectionToBackward, 68
  - setRearWheelsDirectionToForward, 68
  - setRearWheelsSpeed, 68
  - stop, 68
  - terminate, 68
- car::system::movement::MovementSystem, 91
  - ~MovementSystem, 92
  - initialize, 92
  - movement\_controller, 95
  - MovementSystem, 92
  - setCameraServo1Angle, 92
  - setCameraServo2Angle, 93
  - setFrontWheelsAngle, 93
  - setRearLeftWheelDirectionToBackward, 93
  - setRearLeftWheelDirectionToForward, 93
  - setRearLeftWheelSpeed, 93
  - setRearRightWheelDirectionToBackward, 93
  - setRearRightWheelDirectionToForward, 93
  - setRearRightWheelSpeed, 94
  - setRearWheelsDirectionToBackward, 94
  - setRearWheelsDirectionToForward, 94
  - setRearWheelsSpeed, 94
  - start, 94
  - stop, 94
  - terminate, 94
- car\_system
  - behaviour\_tree::BehaviourTreeHandler, 32
  - behaviour\_tree::CarContext, 40
  - car::display::console::CarConsole, 38
  - car::display::console::component::main::ConnectButton, 48
  - car::display::console::component::settings::SettingsEditConfig, 101
  - car::display::console::screen::MainScreen, 84
  - car::display::console::screen::SettingsScreen, 103
  - car::system::device::DeviceManager, 64
  - rpi\_daemon, 99
  - CarConsole
    - car::display::console::CarConsole, 37
  - CarContext
    - behaviour\_tree::CarContext, 39
  - CarSystem
    - car::system::CarSystem, 41
  - checkbox\_value
    - car::display::console::component::debug::DebugEnabler, 49
  - color\_and\_signal\_
    - car::system::messaging::MessagingSystem, 90
  - common/include/behaviour\_tree/BehaviourTreeHandler.hpp, 112, 113
  - common/include/behaviour\_tree/CarContext.hpp, 115
  - common/include/car/configuration/Configuration.h, 116
  - common/include/car/plugin/Plugin.h, 116, 117
  - common/include/car/plugin/PluginManager.h, 117, 118
  - common/include/car/system/CarSystem.h, 119
  - common/include/car/system/device/CameraDevice.h, 120, 121
  - common/include/car/system/device/DeviceManager.h, 121, 122
  - common/include/car/system/device/lidar/LidarDevice.h, 123
  - common/include/car/system/device/lidar/LidarDummy.h, 124
  - common/include/car/system/device/lidar/LidarScanner.h, 124, 125
  - common/include/car/system/logging/VectorSink.h, 126, 127
  - common/include/car/system/messaging/MessagingSystem.h, 127, 128
  - common/include/car/system/messaging/StreamType.h, 129, 130
  - common/include/car/system/movement/controller/AbstractMovementController.h, 130
  - common/include/car/system/movement/controller/DeviceMovementController.h, 131
  - common/include/car/system/movement/controller/DummyMovementController.h, 132
  - common/include/car/system/movement/devices/RearWheel.h, 133
  - common/include/car/system/movement/devices/Servo.h, 133
  - common/include/car/system/movement/MovementSystem.h, 134
  - common/src/car/system/CarSystem.cpp, 136

- common/src/car/system/device/CameraDevice.cpp, 136
- common/src/car/system/device/DeviceManager.cpp, 136
- common/src/car/system/messaging/MessagingSystem.cpp, 137
- common/src/car/system/movement/controller/DeviceMovementController.cpp, 137
- common/src/car/system/movement/controller/DummyMovementController.cpp, 137
- common/src/car/system/movement/devices/RearWheel.cpp, 138
- common/src/car/system/movement/devices/Servo.cpp, 138
- common/tests/pca9685/test\_front\_wheels.cpp, 138
- common/tests/tb6612/test\_rear\_wheels.cpp, 139
- component
  - car::display::console::component::debug::DebugEnabler, 49
- condition
  - car::system::messaging::MessagingSystem::FirstMessageFinalizer, 69
- config\_file\_path
  - car::configuration::JsonConfiguration, 71
- configuration
  - car::system::device::CameraDevice, 36
- configuration\_
  - car::system::CarSystem, 43
  - car::system::device::lidar::LidarScanner, 78
  - car::system::messaging::MessagingSystem, 90
- connect
  - rpi\_daemon, 98
- connect\_button
  - car::display::console::screen::MainScreen, 84
- ConnectButton
  - car::display::console::component::main::ConnectButton, 47
- ConnectButton.cxx
  - CONNECTBUTTON\_CXX, 147
- CONNECTBUTTON\_CXX
  - ConnectButton.cxx, 147
- connected\_
  - car::system::device::CameraDevice, 36
  - car::system::messaging::MessagingSystem, 90
- connection\_ms\_interval
  - rpi\_daemon, 100
- context
  - behaviour\_tree::BehaviourTreeHandler, 32
- create
  - car::system::device::CameraDevice, 34
  - car::system::device::DeviceManager, 63
  - car::system::device::lidar::LidarScanner, 77
- daemon/install/README.md, 140
- daemon/README.md, 140
- daemon/src/main.cpp, 110
- debounce
  - car::display::console::component::debug::DebugEnabler, 50
- DEBUG\_ENABLE\_WARNING\_MESSAGE
- car::display::console::component::debug::DebugEnabler, 50
  - debug\_enabler
- car::display::console::screen::SettingsScreen, 103
  - debug\_lidar\_checkbox
  - debug\_messaging\_text\_box
- car::display::console::screen::SettingsScreen, 103
  - DEBUG\_MODE\_DISABLED\_MESSAGE
- car::display::console::component::debug::DebugEnabler, 50
  - DEBUG\_MODE\_ENABLED\_MESSAGE
- car::display::console::component::debug::DebugEnabler, 50
  - DEBUG\_MODE\_WAIT\_MESSAGE
- car::display::console::component::debug::DebugEnabler, 50
  - debug\_movement\_renderer
- car::display::console::screen::SettingsScreen, 103
  - DebugEnabler.cxx
  - DEBUGENABLER\_CXX, 144
  - DEBUGENABLER\_CXX
  - DebugEnabler.cxx, 144
  - DebugLidarCheckbox
  - car::display::console::component::debug::DebugLidarCheckbox, 51
  - DebugLidarCheckbox.cxx
  - DEBUGLIDARCHECKBOX\_CXX, 144
  - DEBUGLIDARCHECKBOX\_CXX
  - DebugLidarCheckbox.cxx, 144
  - DebugMessagingTextbox
  - car::display::console::component::debug::DebugMessagingTextbox, 53
  - DebugMessagingTextbox.cxx
  - DEBUGMESSAGINGTEXTBOX\_CXX, 145
  - DEBUGMESSAGINGTEXTBOX\_CXX
  - DebugMessagingTextbox.cxx, 145
  - DebugMovementRenderer
  - car::display::console::component::debug::DebugMovementRenderer, 56
  - DebugMovementRenderer.cxx
  - DEBUGMOVEMENTRENDERER\_CXX, 146
  - DEBUGMOVEMENTRENDERER\_CXX
  - DebugMovementRenderer.cxx, 146
  - DEFAULT\_FRONT\_WHEEL\_ANGLE
  - car::display::console::component::debug::DebugMovementRenderer, 58
  - DEFAULT\_REAR\_WHEEL\_SPEED
  - car::display::console::component::debug::DebugMovementRenderer, 58
  - device\_manager\_
    - car::system::CarSystem, 43
  - DeviceManager
    - car::system::device::CameraDevice, 36
    - car::system::device::DeviceManager, 63
    - car::system::device::lidar::LidarDevice, 73
  - disconnect
    - car::system::CarSystem, 41

- car::system::device::CameraDevice, 35
- car::system::device::lidar::LidarDevice, 72
- car::system::device::lidar::LidarDummy, 74
- car::system::device::lidar::LidarScanner, 77
- display\_warn\_debug\_modal
  - car::display::console::component::debug::DebugEnabler, 50
- element
  - car::display::console::component::debug::DebugLidarCheckbox, 52
  - car::display::console::component::debug::DebugMessagingSystem, 54
  - car::display::console::component::debug::DebugMovementRenderer, 56
  - car::display::console::component::main::ConnectButton, 47
  - car::display::console::component::main::MainErrorModal, 81
  - car::display::console::component::main::MainExitModal, 83
  - car::display::console::component::settings::SettingsEditControl, 101
  - car::display::console::screen::LoggingScreen, 79
  - car::display::console::screen::MainScreen, 84
  - car::display::console::screen::SettingsScreen, 103
- enabled
  - car::display::console::component::debug::DebugEnabler, 50
- error\_element
  - car::display::console::component::main::MainErrorModal, 81
- error\_message
  - car::system::messaging::MessagingSystem::FirstMessageSignal, 69
- error\_modal\_shown
  - car::display::console::component::main::MainErrorModal, 82
- exe\_dir
  - car::configuration::JsonConfiguration, 71
- exit
  - car::display::console::component::main::MainExitModal, 83
- exit\_modal\_shown
  - car::display::console::component::main::MainExitModal, 83
- flush\_
  - car::system::logging::VectorSink< Mutex >, 107
- forward
  - BackWheels, 28
  - TB6612, 105
- forward\_A
  - BackWheels, 29
- forward\_B
  - BackWheels, 29
- frame\_buffer\_
  - car::system::device::CameraDevice, 37
- front\_wheels\_angle\_slider
  - car::display::console::component::debug::DebugMovementRenderer, 59
  - front\_wheels\_angle\_slider\_value
    - car::display::console::component::debug::DebugMovementRenderer, 59
  - get\_log\_messages
    - car::system::logging::VectorSink< Mutex >, 107
  - getCameraDevice
    - car::system::device::DeviceManager, 63
  - getCameraFpsInterval
    - car::configuration::Configuration, 45
  - getCameraServo1AngleSliderValue
    - car::display::console::component::debug::DebugMovementRenderer, 56
  - getCameraServo2AngleSliderValue
    - car::display::console::component::debug::DebugMovementRenderer, 56
  - getCarSystem
    - behaviour\_tree::CarContext, 40
  - getCheckbox
    - car::display::console::component::debug::DebugEnabler, 49
  - getCommandSignal
    - car::system::messaging::MessagingSystem, 87
  - getConfigFilePath
    - car::configuration::JsonConfiguration, 70
  - getConfiguration
    - car::system::CarSystem, 41
  - getDeviceManager
    - car::system::CarSystem, 41
  - getDisconnectSignal
    - car::system::messaging::MessagingSystem, 87
  - getFirstMessage
    - car::system::messaging::MessagingSystem, 87
  - getFrameBuffer
    - car::system::device::CameraDevice, 35
  - getFrontWheelsAngleSliderValue
    - car::display::console::component::debug::DebugMovementRenderer, 56
  - getLidarDevice
    - car::system::device::DeviceManager, 63
  - main.cpp, 110, 111
  - getLidarMotorSignal
    - car::display::console::component::debug::DebugLidarCheckbox, 52
  - getMessageSignal
    - car::system::messaging::MessagingSystem, 87
  - getMessagingSystem
    - car::system::CarSystem, 42
  - getMotorPin
    - TB6612, 105
  - getMovementController
    - main.cpp, 110, 112
  - getMovementSystem
    - car::system::CarSystem, 42
  - getName
    - behaviour\_tree::BehaviourTreeHandler, 31
    - car::plugin::Plugin, 95

- getPlugin
  - car::plugin::PluginManager, 97
  - car::system::CarSystem, 42
- getPWMPin
  - TB6612, 105
- getRearLeftWheelSpeedSliderValue
  - car::display::console::component::debug::DebugMovementController, 57
- getRearRightWheelSpeedSliderValue
  - car::display::console::component::debug::DebugMovementController, 57
- getRearWheelDirectionSignal
  - car::display::console::component::debug::DebugMovementController, 57
- getScanData
  - car::system::device::lidar::LidarDevice, 72
- getSelectionSignal
  - car::system::messaging::MessagingSystem, 87
- getSpeed
  - BackWheels, 28
- getUUID
  - car::system::messaging::MessagingSystem, 87
- getWarningModal
  - car::display::console::component::debug::DebugEnabler, 49
- handleCommand
  - behaviour\_tree::BehaviourTreeHandler, 31
- handleMessage
  - car::system::messaging::MessagingSystem, 88
- host
  - car::configuration::Configuration, 46
- info
  - car::display::console::screen::MainScreen, 85
- initialize
  - behaviour\_tree::BehaviourTreeHandler, 31
  - car::display::console::CarConsole, 38
  - car::plugin::Plugin, 95
  - car::plugin::PluginManager, 97
  - car::system::CarSystem, 42
  - car::system::device::DeviceManager, 63
  - car::system::device::lidar::LidarDevice, 72
  - car::system::device::lidar::LidarDummy, 74
  - car::system::device::lidar::LidarScanner, 77
  - car::system::messaging::MessagingSystem, 88
  - car::system::movement::controller::AbstractMovementController, 23
  - car::system::movement::controller::DummyMovementController, 66
  - car::system::movement::MovementSystem, 92
- initialized
  - car::system::CarSystem, 44
- initializeWebsocket
  - car::system::messaging::MessagingSystem, 88
- input\_settings\_file\_path
  - car::display::console::component::settings::SettingsEditConfig, 101
- is\_initialized\_
  - car::system::device::DeviceManager, 65
- is\_running\_
  - car::system::device::DeviceManager, 65
- isConnected
  - car::system::messaging::MessagingSystem, 88
- isEnabled
  - car::display::console::component::debug::DebugEnabler, 49
- isRunning
  - car::system::device::DeviceManager, 64
- json\_configuration
  - car::display::console::CarConsole, 38
  - car::display::console::component::settings::SettingsEditConfig, 101
- JsonConfiguration
  - car::configuration::JsonConfiguration, 70
- JsonConfiguration.cxx
  - JSONCONFIGURATION\_CXX, 142
- JSONCONFIGURATION\_CXX
  - JsonConfiguration.cxx, 142
- kbhit
  - main.cpp, 109
- last
  - car::system::device::CameraDevice, 37
- last\_connected
  - behaviour\_tree::BehaviourTreeHandler, 33
  - rpi\_daemon, 100
- left\_wheel
  - BackWheels, 29
- Lidar
  - StreamType.h, 129
- lidar\_
  - car::system::device::lidar::LidarScanner, 78
- lidar\_device\_
  - car::system::device::DeviceManager, 65
- lidar\_motor\_checkbox\_component
  - car::display::console::component::debug::DebugLidarCheckbox, 52
- LIDAR\_MOTOR\_DISABLED\_MESSAGE
  - car::display::console::component::debug::DebugLidarCheckbox, 52
- lidar\_motor\_enabled
  - car::display::console::component::debug::DebugLidarCheckbox, 52
- LIDAR\_MOTOR\_ENABLED\_MESSAGE
  - car::display::console::component::debug::DebugLidarCheckbox, 52
- lidar\_motor\_loading\_debounce
  - car::display::console::component::debug::DebugLidarCheckbox, 52
- lidar\_motor\_signal
  - car::display::console::component::debug::DebugLidarCheckbox, 53
- lidar\_motor\_status
  - car::display::console::component::debug::DebugLidarCheckbox, 53

- lidar\_port
  - car::configuration::Configuration, 46
- LidarDummy
  - car::system::device::lidar::LidarDummy, 74
- LidarScanner
  - car::system::device::lidar::LidarScanner, 76
- line\_elements
  - car::display::console::screen::LoggingScreen, 80
- load\_button
  - car::display::console::component::settings::SettingsEditControl, 101
- loadConfiguration
  - car::configuration::JsonConfiguration, 70
- log\_messages
  - car::system::logging::VectorSink< Mutex >, 107
- LoggingScreen
  - car::display::console::screen::LoggingScreen, 79
- LoggingScreen.cxx
  - LOGGINGSCREEN\_CXX, 150
- LOGGINGSCREEN\_CXX
  - LoggingScreen.cxx, 150
- main
  - main.cpp, 109, 111, 112
  - test\_front\_wheels.cpp, 138
  - test\_rear\_wheels.cpp, 139
- main.cpp
  - getLidarDevice, 110, 111
  - getMovementController, 110, 112
  - kbhit, 109
  - main, 109, 111, 112
  - terminate\_handler, 111
- main\_button
  - car::display::console::component::main::ConnectButton, 48
- main\_button\_text
  - car::display::console::component::main::ConnectButton, 48
- main\_component
  - car::display::console::screen::MainScreen, 85
- main\_debounce
  - car::display::console::component::main::ConnectButton, 48
- main\_error\_modal
  - car::display::console::component::main::MainErrorModal, 82
  - car::display::console::screen::MainScreen, 85
- main\_exit\_modal
  - car::display::console::component::main::MainExitModal, 83
  - car::display::console::screen::MainScreen, 85
- main\_screen
  - car::display::console::screen::MainScreen, 85
- MainErrorModal
  - car::display::console::component::main::MainErrorModal, 81
- MainErrorModal.cxx
  - MAINERRORMODAL\_CXX, 147
- MAINERRORMODAL\_CXX
  - MainErrorModal.cxx, 147
- MainExitModal
  - car::display::console::component::main::MainExitModal, 82
- MainExitModal.cxx
  - MAINEXITMODAL\_CXX, 148
- MAINEXITMODAL\_CXX
  - MainExitModal.cxx, 148
- MainScreen
  - car::display::console::screen::MainScreen, 84
- MainScreen.cxx
  - MAINSCREEN\_CXX, 150
- MAINSCREEN\_CXX
  - MainScreen.cxx, 150
- map
  - test\_front\_wheels.cpp, 138
- max\_lines
  - car::system::logging::VectorSink< Mutex >, 108
- menu
  - car::display::console::screen::LoggingScreen, 80
- message
  - car::display::console::component::debug::DebugMessagingTextbox, 54
- message\_signal
  - car::display::console::component::debug::DebugMessagingTextbox, 54
- message\_signal\_
  - car::system::messaging::MessagingSystem, 90
- messaging\_container
  - car::display::console::component::debug::DebugMessagingTextbox, 54
- messaging\_system\_
  - car::system::CarSystem, 44
- messaging\_textbox
  - car::display::console::component::debug::DebugMessagingTextbox, 54
- messaging\_title
  - car::display::console::component::debug::DebugMessagingTextbox, 54
- MessagingSystem
  - car::system::messaging::MessagingSystem, 86
- motor\_pin
  - TB6612, 106
- movement\_controller
  - car::system::movement::MovementSystem, 95
- movement\_system\_
  - car::system::CarSystem, 44
- MovementSystem
  - car::system::movement::MovementSystem, 92
- my\_custom\_menu
  - car::display::console::screen::LoggingScreen, 80
- None
  - StreamType.h, 129
- offset
  - TB6612, 106
  - test\_front\_wheels.cpp, 139
- on\_connect\_failure

car::display::console::component::main::ConnectButton, 48  
 on\_disconnect\_signal\_  
   car::system::messaging::MessagingSystem, 90  
 on\_reload  
   rpi\_daemon, 98  
 on\_start  
   rpi\_daemon, 99  
 on\_stop  
   rpi\_daemon, 99  
 on\_update  
   rpi\_daemon, 99  
 onDisconnect  
   car::system::messaging::MessagingSystem, 88  
 onFirstMessage  
   car::system::messaging::MessagingSystem, 89  
 onMessageCallback  
   car::system::messaging::MessagingSystem, 89  
 operator=  
   car::system::device::CameraDevice, 35  
 pca9685  
   BackWheels, 29  
 placeholder  
   car::display::console::component::settings::SettingsEditConfig, 101  
 plugin\_manager\_  
   car::system::CarSystem, 44  
 plugins  
   car::plugin::PluginManager, 97  
 previous\_camera\_servo\_1\_angle\_slider\_angle  
   car::display::console::component::debug::DebugMovementRenderer, 59  
 previous\_camera\_servo\_2\_angle\_slider\_angle  
   car::display::console::component::debug::DebugMovementRenderer, 59  
 previous\_front\_wheels\_angle\_slider\_value  
   car::display::console::component::debug::DebugMovementRenderer, 59  
 previous\_rear\_left\_wheel\_speed\_slider\_value  
   car::display::console::component::debug::DebugMovementRenderer, 59  
 previous\_rear\_right\_wheel\_speed\_slider\_value  
   car::display::console::component::debug::DebugMovementRenderer, 59  
 previous\_rear\_wheels\_speed\_slider\_value  
   car::display::console::component::debug::DebugMovementRenderer, 60  
 pwm\_pin  
   TB6612, 106  
 README.md, 140  
 ready  
   BackWheels, 28  
 rear\_left\_wheel\_speed\_slider  
   car::display::console::component::debug::DebugMovementRenderer, 60  
 rear\_left\_wheel\_speed\_slider\_value  
   car::display::console::component::main::ConnectButton, 48  
   car::display::console::component::debug::DebugMovementRenderer, 60  
   car::display::console::component::debug::DebugMovementRenderer, 60  
   rear\_right\_wheel\_speed\_slider\_value  
   car::display::console::component::debug::DebugMovementRenderer, 60  
   rear\_wheel\_direction  
   car::display::console::component::debug::DebugMovementRenderer, 60  
   REAR\_WHEEL\_DIRECTION\_BACKWARD\_MESSAGE  
   car::display::console::component::debug::DebugMovementRenderer, 60  
   rear\_wheel\_direction\_checkbox\_component  
   car::display::console::component::debug::DebugMovementRenderer, 61  
   rear\_wheel\_direction\_debounce  
   car::display::console::component::debug::DebugMovementRenderer, 61  
   REAR\_WHEEL\_DIRECTION\_FORWARD\_MESSAGE  
   car::display::console::component::debug::DebugMovementRenderer, 61  
   rear\_wheel\_direction\_signal  
   car::display::console::component::debug::DebugMovementRenderer, 61  
   rear\_wheel\_direction\_status  
   car::display::console::component::debug::DebugMovementRenderer, 61  
   rear\_wheel\_menu\_entry  
   car::display::console::component::debug::DebugMovementRenderer, 61  
   rear\_wheel\_speed\_slider  
   car::display::console::component::debug::DebugMovementRenderer, 61  
   rear\_wheels\_speed\_slider\_value  
   car::display::console::component::debug::DebugMovementRenderer, 62  
 reload  
   car::system::CarSystem, 42  
 repository/packages/tb6612/tb6612/include/TB6612.h, 140  
 repository/packages/tb6612/tb6612/src/TB6612.cpp, 140  
 right\_wheel  
   BackWheels, 29  
   rpi\_daemon, 98  
   any\_configuration\_empty, 99  
   attempted\_to\_reconnect, 99  
   car\_system, 99  
   connect, 98  
   connection\_ms\_interval, 100  
   last\_connected, 100  
   on\_reload, 98  
   on\_start, 99  
   on\_stop, 99  
   on\_update, 99  
   update, 99



- run
  - car::display::console::CarConsole, 38
- running
  - car::system::device::lidar::LidarScanner, 78
- scan\_data\_
  - car::system::device::lidar::LidarDevice, 73
  - car::system::device::lidar::LidarScanner, 78
- scan\_data\_mutex\_
  - car::system::device::lidar::LidarScanner, 78
- scan\_generator\_
  - car::system::device::lidar::LidarScanner, 79
- selected\_line
  - car::display::console::screen::LoggingScreen, 80
- selection\_signal\_
  - car::system::messaging::MessagingSystem, 91
- sendData
  - car::system::CarSystem, 42
- sendMessage
  - car::system::messaging::MessagingSystem, 89
- servo\_menu\_entry
  - car::display::console::component::debug::DebugMovementRenderer, 62
- setAngle
  - test\_front\_wheels.cpp, 138
- setAngleToAnalog
  - test\_front\_wheels.cpp, 139
- setBehaviourTree
  - behaviour\_tree::BehaviourTreeHandler, 31
- setCameraFps
  - car::configuration::Configuration, 45
- setCameraServo1Angle
  - car::system::movement::controller::AbstractMovementController, 24
  - car::system::movement::controller::DummyMovementController, 66
  - car::system::movement::MovementSystem, 92
- setCameraServo2Angle
  - car::system::movement::controller::AbstractMovementController, 24
  - car::system::movement::controller::DummyMovementController, 66
  - car::system::movement::MovementSystem, 93
- setConfigFilePath
  - car::configuration::JsonConfiguration, 70
- setConfiguration
  - car::system::CarSystem, 42
  - car::system::messaging::MessagingSystem, 89
- setErrorMessage
  - car::display::console::component::main::MainErrorModal, 81
- setFrontWheelsAngle
  - car::system::movement::controller::AbstractMovementController, 24
  - car::system::movement::controller::DummyMovementController, 66
  - car::system::movement::MovementSystem, 93
- setOffset
  - TB6612, 105
- setPWM
  - TB6612, 105
- setRearLeftWheelDirectionToBackward
  - car::system::movement::controller::AbstractMovementController, 24
  - car::system::movement::controller::DummyMovementController, 67
  - car::system::movement::MovementSystem, 93
- setRearLeftWheelDirectionToForward
  - car::system::movement::controller::AbstractMovementController, 24
  - car::system::movement::controller::DummyMovementController, 67
  - car::system::movement::MovementSystem, 93
- setRearLeftWheelSpeed
  - car::system::movement::controller::AbstractMovementController, 25
  - car::system::movement::controller::DummyMovementController, 67
  - car::system::movement::MovementSystem, 93
- setRearRightWheelDirectionToBackward
  - car::system::movement::controller::AbstractMovementController, 25
  - car::system::movement::controller::DummyMovementController, 67
  - car::system::movement::MovementSystem, 93
- setRearRightWheelDirectionToForward
  - car::system::movement::controller::AbstractMovementController, 25
  - car::system::movement::controller::DummyMovementController, 67
  - car::system::movement::MovementSystem, 93
- setRearRightWheelSpeed
  - car::system::movement::controller::AbstractMovementController, 25
  - car::system::movement::controller::DummyMovementController, 67
  - car::system::movement::MovementSystem, 94
- setRearWheelsDirectionToBackward
  - car::system::movement::controller::AbstractMovementController, 25
  - car::system::movement::controller::DummyMovementController, 68
  - car::system::movement::MovementSystem, 94
- setRearWheelsDirectionToForward
  - car::system::movement::controller::AbstractMovementController, 26
  - car::system::movement::controller::DummyMovementController, 68
  - car::system::movement::MovementSystem, 94
- setRearWheelsSpeed
  - car::system::movement::controller::AbstractMovementController, 26
  - car::system::movement::controller::DummyMovementController, 68
  - car::system::movement::MovementSystem, 94
- setScanData
  - car::system::device::lidar::LidarDevice, 72

- setSpeed
  - BackWheels, 28
- settings\_edit\_config
  - car::display::console::screen::SettingsScreen, 104
- settings\_file\_path
  - car::display::console::component::settings::SettingsEditConfig, 102
- SettingsEditConfig
  - car::display::console::component::settings::SettingsEditConfig, 100
- SettingsEditConfig.cxx
  - SETTINGSEDTCONFIG\_CXX, 149
- SETTINGSEDTCONFIG\_CXX
  - SettingsEditConfig.cxx, 149
- SettingsScreen
  - car::display::console::screen::SettingsScreen, 102
- SettingsScreen.cxx
  - SETTINGSSCREEN\_CXX, 151
- SETTINGSSCREEN\_CXX
  - SettingsScreen.cxx, 151
- SETUP.md, 141
- sink\_it\_
  - car::system::logging::VectorSink< Mutex >, 107
- slider\_container
  - car::display::console::component::debug::DebugMovementController, 62
- speed
  - BackWheels, 30
- start
  - car::system::CarSystem, 43
  - car::system::device::CameraDevice, 35
  - car::system::device::DeviceManager, 64
  - car::system::device::lidar::LidarDevice, 72
  - car::system::device::lidar::LidarDummy, 75
  - car::system::device::lidar::LidarScanner, 77
  - car::system::movement::MovementSystem, 94
- startBehaviourTree
  - behaviour\_tree::BehaviourTreeHandler, 31
- started
  - car::system::CarSystem, 44
- status
  - car::display::console::component::debug::DebugEnabler, 51
- stop
  - BackWheels, 28
  - behaviour\_tree::BehaviourTreeHandler, 32
  - car::plugin::Plugin, 96
  - car::plugin::PluginManager, 97
  - car::system::CarSystem, 43
  - car::system::device::CameraDevice, 35
  - car::system::device::DeviceManager, 64
  - car::system::device::lidar::LidarDevice, 72
  - car::system::device::lidar::LidarDummy, 75
  - car::system::device::lidar::LidarScanner, 77
  - car::system::messaging::MessagingSystem, 89
  - car::system::movement::controller::AbstractMovementController, 26
  - car::system::movement::controller::DummyMovementController, 68
  - car::system::movement::MovementSystem, 94
  - car::system::movement::TB6612, 105
  - stopBehaviourTree
  - behaviour\_tree::BehaviourTreeHandler, 32
  - StreamType
  - StreamType.h, 129
  - StreamType.h
  - Both, 129
  - Camera, 129
  - Lidar, 129
  - None, 129
  - StreamType, 129
  - TB6612, 104
  - backward, 105
  - forward, 105
  - getMotorPin, 105
  - getPWMPin, 105
  - motor\_pin, 106
  - offset, 106
  - pwm\_pin, 106
  - setOffset, 105
  - setPWM, 105
  - stop, 105
  - TB6612, 104
  - terminate
    - car::display::console::CarConsole, 38
    - car::plugin::PluginManager, 97
    - car::system::CarSystem, 43
    - car::system::device::CameraDevice, 35
    - car::system::device::DeviceManager, 64
    - car::system::device::lidar::LidarDevice, 73
    - car::system::device::lidar::LidarDummy, 75
    - car::system::device::lidar::LidarScanner, 77
    - car::system::messaging::MessagingSystem, 89
    - car::system::movement::controller::AbstractMovementController, 26
    - car::system::movement::controller::DummyMovementController, 68
    - car::system::movement::MovementSystem, 94
    - terminate\_handler
    - main.cpp, 111
  - test
    - test\_rear\_wheels.cpp, 140
  - test\_front\_wheels.cpp
    - main, 138
    - map, 138
    - offset, 139
    - setAngle, 138
    - setAngleToAnalog, 139
  - test\_rear\_wheels.cpp
    - main, 139
    - test, 140
  - tick\_count
  - behaviour\_tree::BehaviourTreeHandler, 33
  - tryConnect
    - car::system::CarSystem, 43



- car::system::messaging::MessagingSystem, 90
- tui/README.md, 140
- tui/src/car/configuration/JsonConfiguration.cxx, 141
- tui/src/car/display/console/CarConsole.cpp, 142
- tui/src/car/display/console/CarConsole.h, 142, 143
- tui/src/car/display/console/component/debug/DebugEnabler.cxx, 143
- tui/src/car/display/console/component/debug/DebugLidarChecker.cxx, 144
- tui/src/car/display/console/component/debug/DebugMessagingTextbox.cxx, 145
- tui/src/car/display/console/component/debug/DebugMovementRenderer.cxx, 145
- tui/src/car/display/console/component/main/ConnectButton.cxx, 146
- tui/src/car/display/console/component/main/MainErrorModal.cxx, 147
- tui/src/car/display/console/component/main/MainExitModal.cxx, 148
- tui/src/car/display/console/component/settings/SettingsEditConfig.cxx, 148
- tui/src/car/display/console/screen/LoggingScreen.cxx, 149
- tui/src/car/display/console/screen/MainScreen.cxx, 150
- tui/src/car/display/console/screen/SettingsScreen.cxx, 151
- tui/src/main.cpp, 111
- update
  - behaviour\_tree::BehaviourTreeHandler, 32
  - car::display::console::screen::SettingsScreen, 103
  - car::plugin::Plugin, 96
  - car::plugin::PluginManager, 97
  - car::system::CarSystem, 43
  - car::system::device::CameraDevice, 36
  - car::system::device::DeviceManager, 64
  - car::system::device::lidar::LidarDevice, 73
  - car::system::device::lidar::LidarDummy, 75
  - car::system::device::lidar::LidarScanner, 78
  - rpi\_daemon, 99
- updateCameraServo1
  - car::display::console::component::debug::DebugMovementRenderer, 57
- updateCameraServo2
  - car::display::console::component::debug::DebugMovementRenderer, 57
- updateFrontWheels
  - car::display::console::component::debug::DebugMovementRenderer, 57
- updateRearWheels
  - car::display::console::component::debug::DebugMovementRenderer, 57
- use\_camera
  - car::configuration::Configuration, 46
- use\_lidar
  - car::configuration::Configuration, 46
- uuid
  - car::system::messaging::MessagingSystem::FirstMessageStruct, 69
- uuid\_
  - car::system::messaging::MessagingSystem, 91
- vector\_sink
  - car::display::console::CarConsole, 38
  - car::display::console::screen::LoggingScreen, 80
- vector\_sink\_mt
  - car::system::logging, 22
- VectorSink
  - car::system::logging::VectorSink< Mutex >, 107
- websocket
  - car::system::messaging::MessagingSystem, 91
- websocket\_url\_
  - car::system::messaging::MessagingSystem, 91