Model Advisor Report – Head_Light_Cotrol_Final.slx

Simulink version: 9.1

Model version: 1.9

System: Head_Light_Cotrol_Final

Current run: 14-Sep-2021 16:54:01

Treat as Referenced Model: off

Run Summary

Pass	Fail	Warning	Not Run	Total
✓ 65	✗ 0	⚠ 1	🗒 0	66

Modeling Standards for MAAB

Naming Conventions ✓7 ✗0 ⚠0 🗒0

✓ Check file names

Identify file names with incorrect characters or formatting.

Passed

All files have correct names.

✓ Check folder names

Identify folders using incorrect characters and formatting.

Passed

All folder names have correct formatting.

✓ Check subsystem names

Identify subsystem names that use characters that are not correct in C code.

Passed

All subsystem names use correct characters.

✓ Check port block names

Identify names of Inport or Outport blocks that use characters that are not correct in C code.

Passed

All the Inport or Outport block names use correct characters.

✓ Check character usage in signal labels

Identify signal labels that are not correct for C variable names.

Passed

All signal labels use correct characters.

✓ Check character usage in block names

Identify block names that use characters that are not correct in C code.

Passed

All blocks use correct characters.

✓ Check Simulink bus signal names

Passed

No Simulink bus signals found with unusable characters

Model Architecture ✓4 ✗0 ⚠0 ▤0

✓ Check for mixing basic blocks and subsystems

Identify levels in the model that include basic blocks and subsystems. Each level of a model must be designed with blocks of the same level (for example, only subsystems or only basic blocks).

Passed

The model does not mix basic blocks and subsystems at the same level.

✓ Check unused ports in Variant Subsystems

na_0020: Identify and fix unused ports in Variant subsystems

Passed

No unused ports found in variant subsystems

✓ Check use of default variants

na_0036: Identify variant subsystems that do not use default variants

Passed

All variant subsystems in the model use default variants

✓ Check use of single variable variant conditionals

na_0037: Identify variant subsystems which use multi-variable compound conditions

Passed

No variant subsystems with multiple variable compound conditions found

📁 Model Configuration Options ✓2 ✗0 ⚠0 ▤0

✓ Check Implement logic signals as Boolean data (vs. double)

Identify whether **Implement logic signals as Boolean data (vs. double)** is selected.

Passed

Implement logic signals as Boolean data (vs. double) is selected.

✓ Check model diagnostic parameters

Identify diagnostic parameters that are set to `none`.

Passed

All of the diagnostic parameters are set to `error` or `warning`.

📁 Simulink ✓26 ✗0 ⚠1 ▤0

⚠ Check for Simulink diagrams using nonstandard display attributes

Identify nonstandard display attributes in Simulink diagrams.

Check format settings

Identify incorrect model-level format options.

Passed

The format options are correct.

____ **Check block colors**

Identify blocks using nonstandard colors.

Warning

The following blocks use nonstandard colors:

- Head_Light_Cotrol_Final/Day_Night
- Head_Light_Cotrol_Final/Headlight_Control_System/Day_Night
- Head_Light_Cotrol_Final/Headlight_Control_System/Head_Light_Cotrol/Day_Night" title="Head_Light_Cotrol_Final/Headlight_Control_System/Head_Light_Cotrol/Day_Night
- Head_Light_Cotrol_Final/Headlight_Control_System/Headlight_Left
- Head_Light_Cotrol_Final/Headlight_Control_System/Headlight_Right
- Head_Light_Cotrol_Final/Headlight_Control_System/Road_Types_Signal
- Head_Light_Cotrol_Final/Headlight_Control_System/Speed_Sig
- Head_Light_Cotrol_Final/Headlight_Control_System/Vehicle_Detected_Signal
- Head_Light_Cotrol_Final/Headlight_Left
- Head_Light_Cotrol_Final/Headlight_Right
- Head_Light_Cotrol_Final/Road_Types_Signal
- Head_Light_Cotrol_Final/Speed_Sig
- Head_Light_Cotrol_Final/Vehicle_Detected_Signal

∧ Less

Recommended Action

Set the block foreground color to black and the background color to white.

Check canvas colors

Identify canvases that are not white.

Passed

All diagrams use a white canvas.

Check diagram zoom

Identify diagrams that do not have zoom factor set to 100 %.

Note: Zoom factors can differ for each instance of a model diagram opened in Simulink Editor

Passed

All canvases have a zoom factor of 100 percent.

✓ Check font formatting

Identify inconsistent formatting of text.

Passed

All block and line fonts have consistent formatting.

The following font characteristics are used in the model/subsystem. Font characteristics are sorted by number of occurrences. The most common characteristics are bold.

Font Name	Font Size	Font Style
Arial	**10**	**normal**

Input Parameters Selection

Name	Value

Font Name	Common
Font Size	Common
Font Style	Common

✓ Check positioning and configuration of ports

Identify input and output ports with incorrect positioning and configurations.

Check Inport blocks position

Identify Inport blocks that result in left-flowing signals.

Passed

There are no Inport blocks in the model that result in left-flowing signals.

Check Outport block position

Identify Outport blocks that result in left-flowing signals.

Passed

There are no Outport blocks in the model that result in left-flowing signals.

Check port orientation

Identify port blocks with nondefault orientation.

Passed

All ports use the default orientation.

Check for duplicate Inports blocks

Identify duplicate Inport blocks.

Passed

All Inport blocks in the model are used once.

✓ Check visibility of block port names

Identify port block names that are not uniformly displayed. The block names must all be displayed or none displayed. Library blocks are an exception to this rule. This check ignores masked and subsystem blocks.

Check for incorrect port name display

Identify ports that are incorrectly displaying names.

Passed

Subsystem blocks are correctly displayed.

Check for incorrect subsystem port name display

Identify subsystems that are incorrectly displaying names.

Passed

Subsystem blocks are correctly displayed.

Input Parameters Selection

Name	Value
Display all port names (Diagram > Format > Show Block Name).	true

---

✓ Check display for port blocks

Identify Inport and Outport blocks that do not specify `Port number` for the **Icon display** block parameter.

Passed

All port blocks display the port number.

✓ Check whether block names appear below blocks

Identify blocks where the name is not displayed below the block.

Passed

All blocks have names displayed below the block.

✓ Check the display attributes of block names

Identify whether to display block names.

Check for blocks with hidden names and obvious function

Identify block names that are displayed but can be hidden due to obvious behavior.

Passed

All blocks with obvious behavior have hidden names.

Check for non-descriptive displayed block names

Identify block names that are displayed but should be hidden due to a lack of a descriptive name.

Passed

All displayed names provide descriptive information.

Check for missing block names

Identify block names that are hidden but should be displayed to show a descriptive name.

Passed

All displayed names provide descriptive information.

✓ Check position of Trigger and Enable blocks

Identify Trigger and Enable blocks that are not located at the top of the subsystem diagram.

Passed

The model does not contain Trigger and Enable blocks, or the blocks are located at the top of the subsystem diagram.

✓ Check for nondefault block attributes

Identify blocks that use and fail to display nondefault values.

Passed

Model displays all block parameter values that are not default values.

✓ Check for matching port and signal names

Identify mismatches between names of ports and corresponding signals.

Passed

All signal and port names match.

✓ Check Trigger and Enable block names

Match the name of the Trigger and Enable blocks to the name of the signal that triggers the subsystem. If the signal is not named, use any name for the block.

Check Trigger block names

Identify Trigger blocks that do not match the names of the signals to which they are connected.

Passed

All Trigger blocks in the system are correctly named.

Check for root level Trigger block

Identify Trigger blocks placed at the root level of the model.

Passed

No Trigger blocks where found at the root level.

Check Enable block names

Identify Enable blocks that do not match the names of the signals to which they are connected.

Passed

All Enable blocks in the system are correctly named.

Check for root level Enable block

Identify Enable blocks placed at the root level of the model.

Passed

No Enable blocks where found at the root level.

✓ Check signal line labels

Identify blocks that require labeled signals. A subset of source and destination blocks require labeled signals.

Check source block labels

The following source blocks require labeled signals: Inport, From, DataStoreRead, SubSystem, Constant, BusSelector, Demux, Selector. If the signal name is visible on the block, this rule is considered met.

Passed

All blocks that require labeled sources have labeled signals.

Check destination block labels

The following destination blocks require labeled signals: Outport, Goto, DataStoreWrite, BusCreator, Mux, SubSystem. If the signal name is visible on the source block, this rule is considered met.

Passed

All blocks that require labeled sources have labeled signals.

✓ Check for propagated signal labels

Identify propagated labels on signal lines.

Check subsystem input labels

Identify subsystem inputs that are labeled and display propagated signals.

Passed

All inputs to the subsystem have labels and display propagated signals.

Check subsystem output labels

Identify outputs from subsystems that are labeled and display signal propagation.

Passed

All outputs from the subsystem have labels and display propagated signals.

Signal propagation for nonsubsystem blocks

Identify the signal propagation status for both transformative and nontransformative blocks.

Passed

All outputs from non subsystem blocks correctly use labels and display propagated signals.

✓ Check for unconnected ports and signal lines

Identify unconnected block input ports, output ports, and signal lines.

Passed

All lines and ports in the model are connected.

✓ Check for prohibited blocks in discrete controllers

Identify blocks that are not allowed in discrete controllers. Prohibited blocks include all continuous blocks and some source and sink blocks.

Passed

All blocks in the model are allowed in discrete controllers.

✓ Check for blocks not recommended for C/C++ production code deployment

Identify blocks not supported by code generation or not recommended for C/C++ production code deployment.

Passed

Blocks not recommended for C/C++ production code deployment were not found in the model or subsystem.

✓ Check for prohibited sink blocks

Identify sink blocks that must be removed prior to code generation.

Passed

There are no prohibited blocks in the subsystem.

✓ Check scope of From and Goto blocks

Identify incorrect scoping of From and Goto blocks. For signal flows, From and Goto blocks must use local scope. Control flow can use global scope.

Passed

All From and Goto blocks are used correctly.

✓ Check usage of Switch blocks

Identify Switch blocks that do not use Boolean inputs for the switch condition (input 2), and do not use `u2 ~= 0` for the **Criteria for passing first input** block parameter.

Check Switch block parameters

Identify Switch blocks with the parameter **Criteria for passing first input** not set to `u2 ~= 0`.

Passed

The block parameter **Criteria for passing first input** is correctly configured.

Check for Boolean switch condition

Identify blocks that do not use Boolean signal switch conditions (input 2).

Passed

The switch condition is a Boolean signal.

✓ Check usage of Relational Operator blocks

Identify Relational Operator blocks that connect to constants with the first (upper) input value.

Passed

All Relational Operator blocks with constant input values are configured correct.

✓ Check for indexing in blocks

Identify blocks with inconsistent indexing. To increase code efficiency, use zero-based indexing.

Passed

All blocks in the system use a consistent indexing method.

✓ Check usage of buses and Mux blocks

Bus signal treated as vector

Identify bus signals in the model that are treated as vectors by the Simulink software.

Passed

The model uses bus signals properly. Model is configured to detect future changes that might result in improper bus signal usage.

✓ Check usage of tunable parameters in blocks

Identify tunable parameters used to specify expressions, data type conversions, or indexing operations.

Passed

Tunable parameters are not used in the model.

✓ Check orientation of Subsystem blocks

Check for subsystems that do not have the correct orientation. Blocks with the correct orientation have inputs on the left and outputs on the right.

Passed

All subsystem blocks use the correct orientation.

✓ Check fundamental logical and numerical operations

Check input data types of blocks meant for numeric operations

Passed

No numeric blocks found with boolean inputs.

Check input data types of blocks meant for logical operations

Passed

No logic operation blocks found with non-boolean data types at input

---

✓ Check usage of merge blocks

na_0032: Identify usage of merge blocks in Simulink

Passed

No merge blocks found with inconsistent inputs

✓ Check logical expressions in If blocks

Checks If blocks for complex usage of primary expressions within a logical expression

Passed

Logical expressions inside If blocks are simple

📁 Stateflow ✓20 ✖0 ⚠0 🗒0

✓ Check usage of exclusive and default states in state machines

Identify Stateflow charts and substates that incorrectly use or define exclusive and default states.

Check Stateflow charts for exclusive states

Identify Stateflow charts that have singular exclusive (OR) states.

Passed

The Stateflow charts do not have singular exclusive (OR) states.

Check Stateflow charts for undefined default states

Identify Stateflow charts that do not define default states.

Passed

Each Stateflow chart defines a default state.

Check for multiple states assigned as the default state

At the root level in the Stateflow hierarchy only one state should be assigned as the default.

Passed

The root level of the chart has only one default state assigned.

Check for substates with singular OR states

States configured as OR should always be part of a group of states.

Passed

No singular OR states were detected.

Check for substates without default states defined

At every level in the Stateflow hierarchy a default state should be assigned.

Passed

All substates have default states assigned.

Check for substates with multiple default states defined

At every level in the Stateflow hierarchy only one state should be assigned as the default.

Passed

All levels of the chart have only one default state assigned.

✓ Check transition orientations in flow charts

Identify transitions in Stateflow flow charts that are drawn incorrectly.

Check for conditions drawn horizontally

Condition expressions should be drawn on the horizontal segments of flow charts.

Passed

All condition expressions were drawn horizontally.

Check for action transitions drawn vertically

Transition actions should be drawn on the vertical segments of flow charts.

Passed

All transition actions were drawn vertically.

Check for junctions for default transitions

All Junctions in a flow chart should have a default exit transition.

Passed

All Junctions have a default exit transition.

Check for transitions that combine condition and action

Flow charts should not combine condition evaluations and action expressions in a single transition.

Passed

No combined expressions were found in the chart.

✓ Check entry formatting in State blocks in Stateflow charts

Identify missing line breaks between entry action (en), during action (du), and exit action (ex) entries in states. Identify missing line breaks after semicolons (;) in statements.

Passed

No Stateflow charts were found.

✓ Check return value assignments of graphical functions in Stateflow charts

Identify graphical functions with multiple assignments of return values in Stateflow charts.

Passed

No Stateflow charts were found.

✓ Check default transition placement in Stateflow charts

Identify all groupings of states that do not have a default transition or do not have the default state as the top-most state.

Passed

No Stateflow charts were found.

✓ Check for Strong Data Typing with Simulink I/O

Identify whether Stateflow charts have **Use Strong Data Typing with Simulink I/O** cleared.

Passed

No Stateflow charts were found.

✓ Check Stateflow data objects with local scope

Identify Stateflow data objects with local scope that are not scoped at the chart level or below.

Passed

No Stateflow charts were found.

✓ Check usage of return values from a graphical function in Stateflow charts

Identify calls to graphical functions that are used in conditional expressions.

Passed

No conditional expressions containing calls to graphical functions were found.

✓ Check for MATLAB expressions in Stateflow charts

Identify Stateflow objects that use MATLAB expressions that are not suitable for code generation.
Note: This check applies only to Stateflow charts that use C as the action language.

Passed

No Stateflow objects that use MATLAB expressions were found.

✓ Check for pointers in Stateflow charts

Identify pointer operations on custom code variables.
Note: This check applies only to Stateflow charts that use C as the action language.

Passed
No pointer operations were found.

✓ Check for event broadcasts in Stateflow charts

Identify undirected event broadcasts that might cause recursion during simulation and generate inefficient code.

Passed
No instances of undirected event broadcast were found.

✓ Check transition actions in Stateflow charts

Identify missing line breaks between transition actions.

Passed
All transition actions are formatted correctly.

✓ Check for bitwise operations in Stateflow charts

Identify bitwise operators (&, |, and ^) in Stateflow charts. If **Enable C-bit Operations** is selected for a chart, only bitwise operators in expressions containing Boolean data types are reported. Otherwise, all bitwise operators are reported for the chart.
Note: This check applies only to Stateflow charts that use C as the action language.

Identify bitwise operators in charts with 'Enable C-bit Operations' selected.
Display Stateflow charts with **Enable C-bit Operations** selected. Identify bitwise operators (&, |, and ^) in expressions containing Boolean data types.

Passed
The following charts have **Enable C-bit Operations** selected:
N/A

No bitwise operators in expressions containing Boolean data types were found in the above charts.

Identify bitwise operators in charts with 'Enable C-bit Operations' cleared

Display Stateflow charts with **Enable C-bit Operations** cleared. Identify bitwise operators (&, |, and ^) in expressions.

Passed

The following charts have **Enable C-bit Operations** cleared:
N/A

No bitwise operators in expressions with Boolean data types were found in the above charts.

✓ Check for unary minus operations on unsigned integers in Stateflow charts

Identify unary minus operations applied to unsigned integers in Stateflow objects.

Passed

No unary minus operations applied to unsigned integers in Stateflow objects were found.

✓ Check for comparison operations in Stateflow charts

Identify comparison operations with different data types in Stateflow objects.

Passed

No comparison operations with different data types were found.

✓ Check for equality operations between floating-point expressions in Stateflow charts

Identify equal to operations (==) in expressions where at least one side of the expression is a floating-point variable or constant.

Passed

No equal to operations in expressions where at least one side of the expression is a floating-point variable or constant were found.

✓ Check for mismatches between names of Stateflow ports and associated signals

Identify mismatches between names of Stateflow ports and the associated signals.

Passed

No Stateflow charts were found.

✓ Check nested states in Stateflow charts

na_0038: Check nesting levels of states in Stateflow charts.

Passed

No deeply nested states found.

✓ Check use of Simulink in Stateflow charts

na_0039: Check use of Stateflow charts nested inside Simulink functions used in Stateflow.

Passed

No Stateflow charts found nested inside Simulink functions used in Stateflow.

✓ Check number of Stateflow states per container

na_0040: Identify Stateflow containers with number of viewable states more than threshold.

Passed

No Stateflow containers found with number of states exceeding the maximum limit

MATLAB Functions ✓6 ✗0 ⚠0 ▤0

✓ Check input and output settings of MATLAB Functions

Identify MATLAB Functions that have inputs, outputs, or parameters with inherited complexity, data type, or size properties.

Passed

No MATLAB Functions found in the model or subsystem.

✓ Check MATLAB Function metrics

Identify MATLAB Functions that violate complexity limits.

Passed

No MATLAB Function with metrics violations were found.

The following metrics were determined for the model or subsystem.

Legend:

- LoC : Total lines of code
- ELoC : Effective lines of code
- CLoC : Comment lines of code
- DC : Density of comments
- CYC : Cyclomatic complexity

>

Input Parameters Selection

Name	Value
Maximum effective lines of code per function	60
Minimum density of comments	0.2
Maximum cyclomatic complexity per function	15

✓ Check MATLAB code for global variables

Check for global variables in MATLAB code

Check for global variables in MATLAB code used in MATLAB Function blocks

Passed

No MATLAB Function blocks found

Check for global variables in MATLAB functions defined in Stateflow charts

Passed

No MATLAB functions defined in Stateflow charts found

Check for global variables in called MATLAB functions

Passed

No external MATLAB functions found

✓ Check usage of restricted variable names

Checks whether variable names used in MATLAB Function blocks are reserved for C/C++/MATLAB keywords

Passed

No variable names conflict with reserved keywords

✓ Check usage of character vector inside MATLAB Function block

Checks whether character vectors are being used inside MATLAB Function blocks

Passed

No character vectors found in MATLAB Function block

✓ Check usage of recommended patterns for Switch/Case statements

Checks whether non-constant variables are used in Switch/Case arguments.

Passed

Non-constant variables are not used as Switch/Case arguments