2022-2026 EcoCAR EV Challenge

Year 3 PCM Development Challenge - Advanced

Lucas Rajotte (PCM) | September, 2024

McMaster EcoCAR (MAC): McMaster University, Hamilton, ON, Canada

Table of Contents

1. Introduction

Vehicles are required to have certain indicator lights and switches installed and functioning in the vehicle to participate in the dynamic events at final competition. The feature outlined below explains how the indicator board should function within the team vehicle. This document outlines both software and physical requirements for the system. It is your job to extract some software requirements to test the provided feature code against. Note that there is no way for you to validate hardware and physical requirements (such as how the physical indicator board should look, or what type of switches to use).

Prior to completing the development challenge, it is recommended that you either watch a few introductory videos to Simulink and Stateflow or complete the Simulink and Stateflow on-ramps to better understand the provided model.

1.1. How-To: Requirements Extraction

Requirements determination is a crucial step in developing a strong control system. Requirements define the ways in which the control system should act, how it should interact with other systems and subsystems, and any unsafe actions that must be mitigated to safely operate the vehicle.

Within the Indicator Board Feature Overview section, the team-added indicator board and its software interface feature for the team controller are outlined in detail. The overall system requirements are defined but are not called out directly. By reading the section, you will be able to extract the software requirements that will be used to test the feature.

To extract software requirements for a feature that interfaces with an HMI board and relays information to the supervisory controller, you can follow these steps:

1.1.1. Understand the Feature Overview:

- Carefully read the section detailing the indicator board and its software interface. Note the specific functionalities and interactions described.

1.1.2. Identify Key Functional Requirements:

- **LED Indicators**: Determine the specific actions for each LED:
    - **Propulsion System Status LED**: The software should control this LED to indicate the propulsion system's status.
    - **HV System Status LED**: The software should control this LED to indicate the high voltage system's status.
    - **CAV Lateral Control Status LED**: The software should control this LED to indicate the status of the CAV lateral control.
    - **CAV Longitudinal Control Status LED**: The software should control this LED to indicate the status of the CAV longitudinal control.
    - **V2X Status LED**: The software should control this LED to indicate the V2X system's status.

- **Mechanical Buttons**: Identify the actions triggered by each button:
    - **Regenerative Braking Enable/Disable Button**: The software should process input to enable or disable regenerative braking.
    - **CAV Longitudinal Control Enable Button**: The software should process input to enable CAV longitudinal control.
    - **CAV Lateral Control Enable Button**: The software should process input to enable CAV lateral control.
    - **V2X Enable Button**: The software should process input to enable the V2X system.

An example of a requirement that could be extracted from the section is:

IB1.2: The HV System Status light shall be ON when the Battery Contactors are closed and Battery is able to provide current to a drive system, without fault, or facilitate on-plug charging.

Where IB stands for Indicator Board (category), and 1.2 is the requirement number within the list. For the sake of the development challenge, you can do IB 1, IB 2, etc.

2. Your Task

2.1. Getting Started

To complete the development challenge, you will need to install MATLAB R2023a with **ALL ADD-ONS.** Be sure that your add-on list includes Simulink Test and Requirements Toolbox, which is where you will develop test cases for the feature and incorporate your requirements within the test suite. In this development challenge, it is your job to review the Indicator Board Feature Overview section. You must extract a minimum of 5 requirements for the indicator board interface feature.

2.2. Model Review

To successfully create tests, you must understand how the feature model works in Simulink. Navigate through the various blocks and Stateflow states in order to translate the feature overview documentation to the provided Simulink model. This is crucial for creating your test assessments.

2.3. Requirements Suite Creation

Once you have your requirements defined, you will need to incorporate them into the Simulink model environment. To do this, you must use the Requirements Toolbox's requirements editor app to create a requirement set. The requirements set file should be named **IndicatorBoard.slreqx.** For a detailed guide on how to do this, please refer to "Using the Requirements and Simulink Test Toolboxes.pdf".

2.4. Test Suite and Requirements Tests Creation

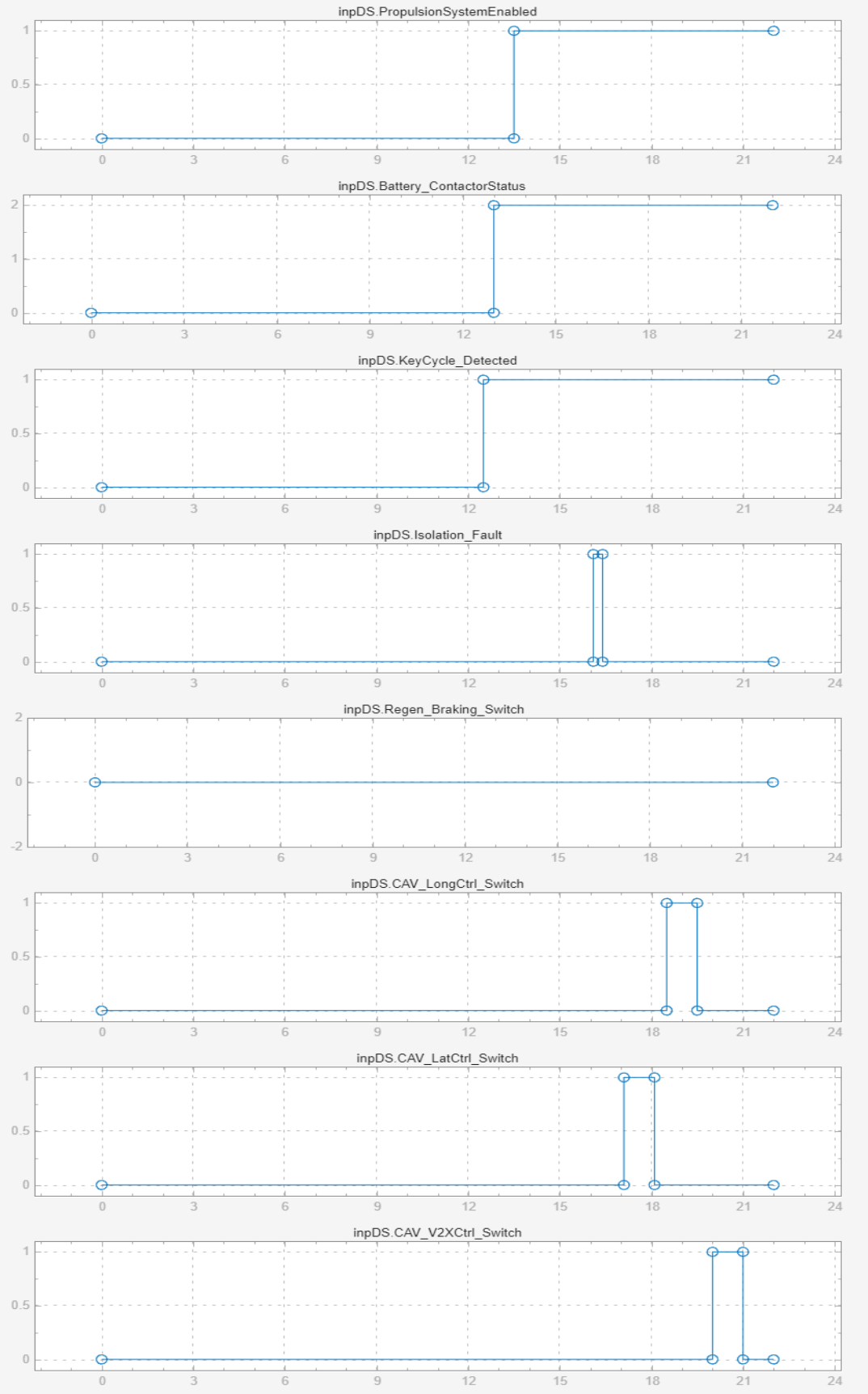
Once your requirement set is created, you will need to create your test suite to evaluate the feature. To do this, you must use the Simulink Test toolbox's Test Manager app to create your test suite. The test suite file should be named **IndicatorBoard.mldatx**. For a detailed guide on how to do this, please refer to "Using the Requirements and Simulink Test Toolboxes.pdf".

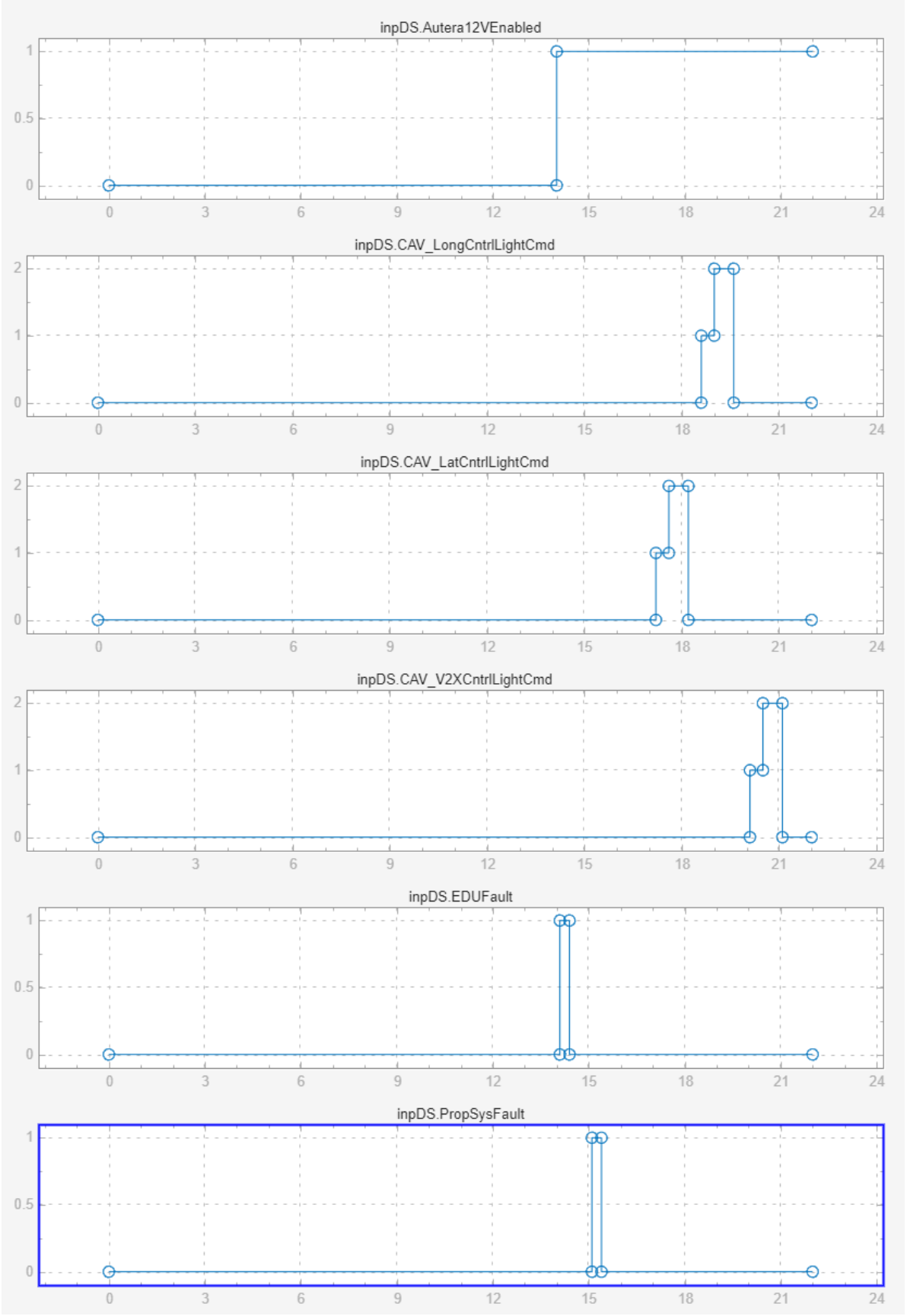
2.4.1. System Under Test Settings

Your system under test settings should be configured to use the IndicatorBoard.slx model and should have a stop time of 22 seconds. Please refer to the aforementioned guide which outlines how to do this.

2.4.2. Test Case Inputs

For your requirements tests, you can use the same set of inputs. All input plots are shown in the two images below and will need to be replicated for your tests. Once you create your inputs for the first test case, you can add them to each subsequent test case. Please refer to the aforementioned guide which outlines how to do create or add inputs to your test case.


inpDS.PropulsionSystemEnabled
inpDS.Battery_ContactorStatus
inpDS.KeyCycle_Detected
inpDS.Isolation_Fault
inpDS.Regen_Braking_Switch
inpDS.CAV_LongCtrl_Switch
inpDS.CAV_LatCtrl_Switch
inpDS.CAV_V2XCtrl_Switch


inpDS.Autera12VEnabled
inpDS.CAV_LongCntrlLightCmd
inpDS.CAV_LatCntrlLightCmd
inpDS.CAV_V2XCntrlLightCmd
inpDS.EDUFault
inpDS.PropSysFault

2.4.3. Test Suite Baseline Criteria

For your requirements tests, you can skip adding baseline data. It will not be necessary for the development challenge.

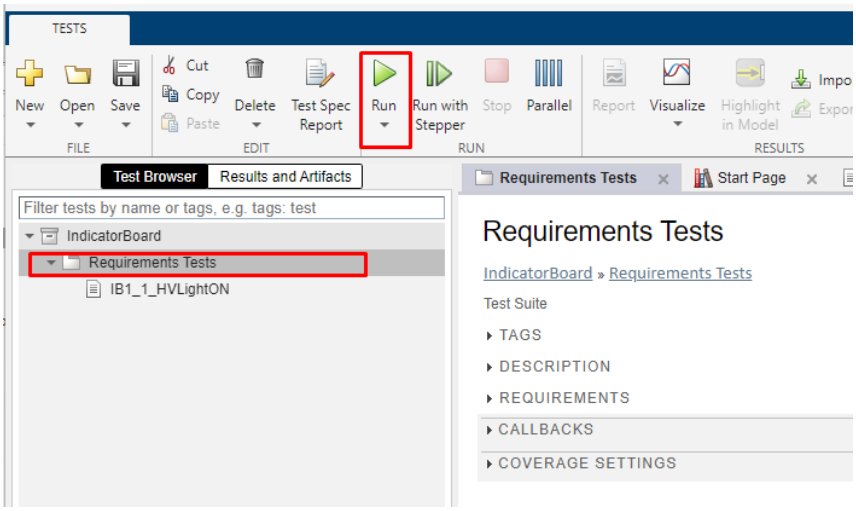
2.4.4. Logical and Temporal Assessments for your Tests

Each requirement must be evaluated using a logical and temporal assessment within the test case. For the development challenge, each requirement must have its own assessment, and its own test case. Therefore, each test case shall only have one assessment. Please refer to the guide which outlines how to do this in detail.

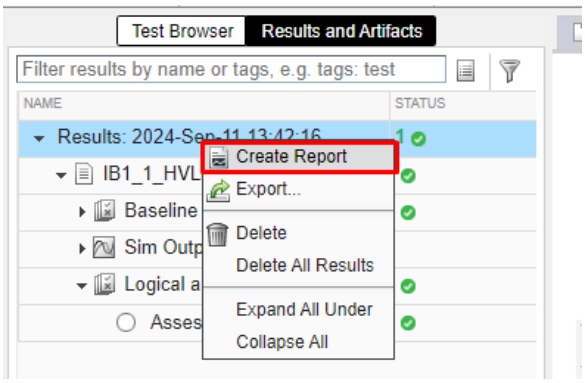
2.4.5. Final Submission/Presentation

Each development challenge team is required to prepare a final submission and presentation. The final submission will be submitted after the dev challenge has concluded. All teams will be required to attend the development challenge day at McMaster Automotive Resource Centre (MARC), where you will present a science fair-style presentation of your work and results. This is an informal presentation, where you will walk the evaluators through your development process, results, challenges, etc.

Final Submission: All teams are required to create a test report from the results of your testing. To do this correctly, you will need to run the entire test suite, as opposed to running each individual test case one at a time. To do so, select the test suite from the test browser and press the run button in the top banner.



Once all tests have completed, navigate to the Results and Artifacts tab of the left navigation panel. Right-click the Result and select the ‘Create Report’ option, shown below:



This will open a pop-up window. Ensure your settings match the image below, with the Author section being filled with all teammember macids, and the File Name being set correctly. Your file should be named Team#_Tests (e.g. Team1_Tests).

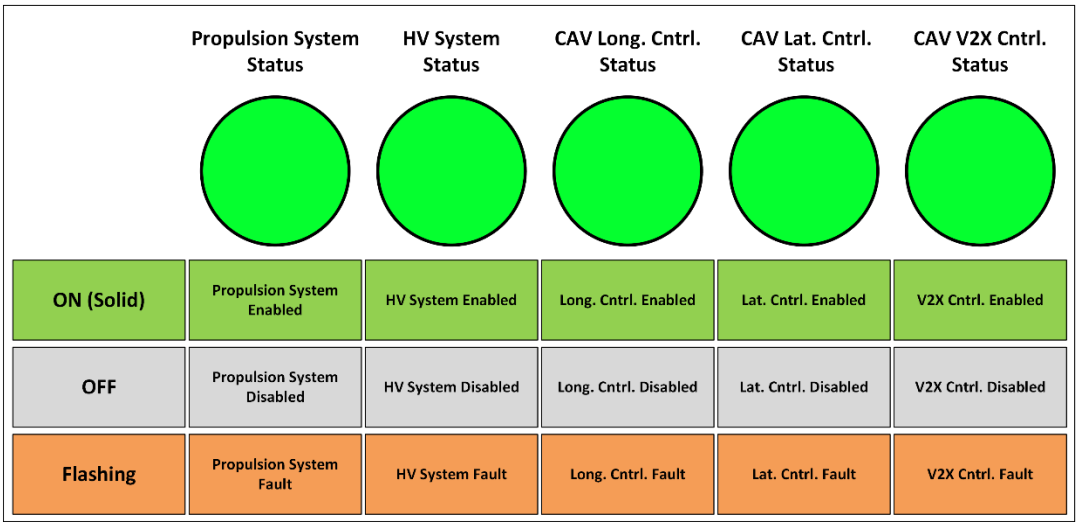
Dev Challenge Presentation: Each team must create a presentation for the final day of the development challenge. This presentation can be made using powerpoint or prezi and must also include a quick walk-through of your test suite live in Simulink. Note that teams will be required to bring their own laptop for their presentation. Your presentation should outline the task, your processes to complete the task, results, challenges faced, and any other information you feel is relevant. Be prepared to answer questions at the end of your presentation.

3. Indicator Board Feature Overview

Each indicator light must be a physical LED and cannot be integrated into a digital display. Additionally, the indicator lights must meet the following requirements:

1. The light is to be illuminated by a **solid LED** when the system is enabled and can respond to control input, without fault.
2. The light is to be illuminated by a **flashing LED** when there is a system-relevant fault present and detected by the vehicle.
3. The light is to be **turned off** when the system is **disabled.**

For the HV System Status Light, it should rely on Contactor Status signal only but should also use the IMD (Insulation Monitoring Device) to report isolation faults.

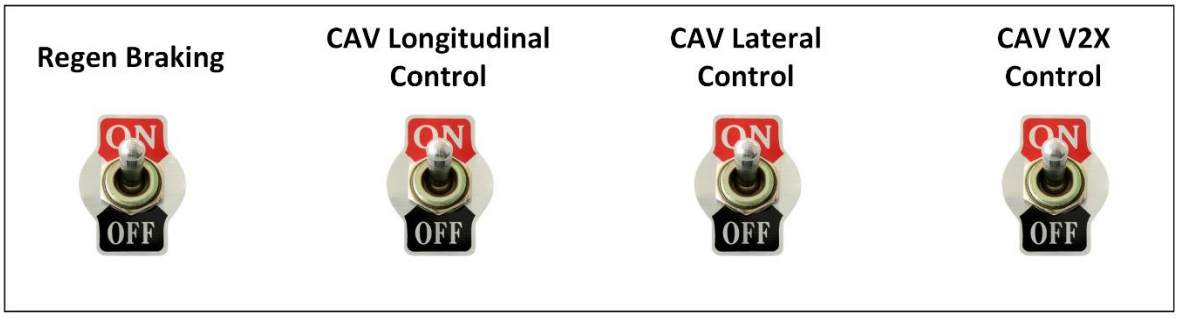


	Propulsion System Status	HV System Status	CAV Long. Cntrl. Status	CAV Lat. Cntrl. Status	CAV V2X Cntrl. Status
ON (Solid)	Propulsion System Enabled	HV System Enabled	Long. Cntrl. Enabled	Lat. Cntrl. Enabled	V2X Cntrl. Enabled
OFF	Propulsion System Disabled	HV System Disabled	Long. Cntrl. Disabled	Lat. Cntrl. Disabled	V2X Cntrl. Disabled
Flashing	Propulsion System Fault	HV System Fault	Long. Cntrl. Fault	Lat. Cntrl. Fault	V2X Cntrl. Fault

The switches can only be activated by a human either when instructed or if an emergency event occurs. Additionally, switches must meet the following requirements:

1. Mechanically activated, **two-state switch** that is mechanically latched and secured in each state.

- Switch states may only change states as a result of physical interaction by the driver.
- Clear indication of the current state of the switch, with the indication state being independent of programmable logic (e.g., on/off state labeled clearly at the switch).



3.1. Explanation For ON/OFF/FLASH Light

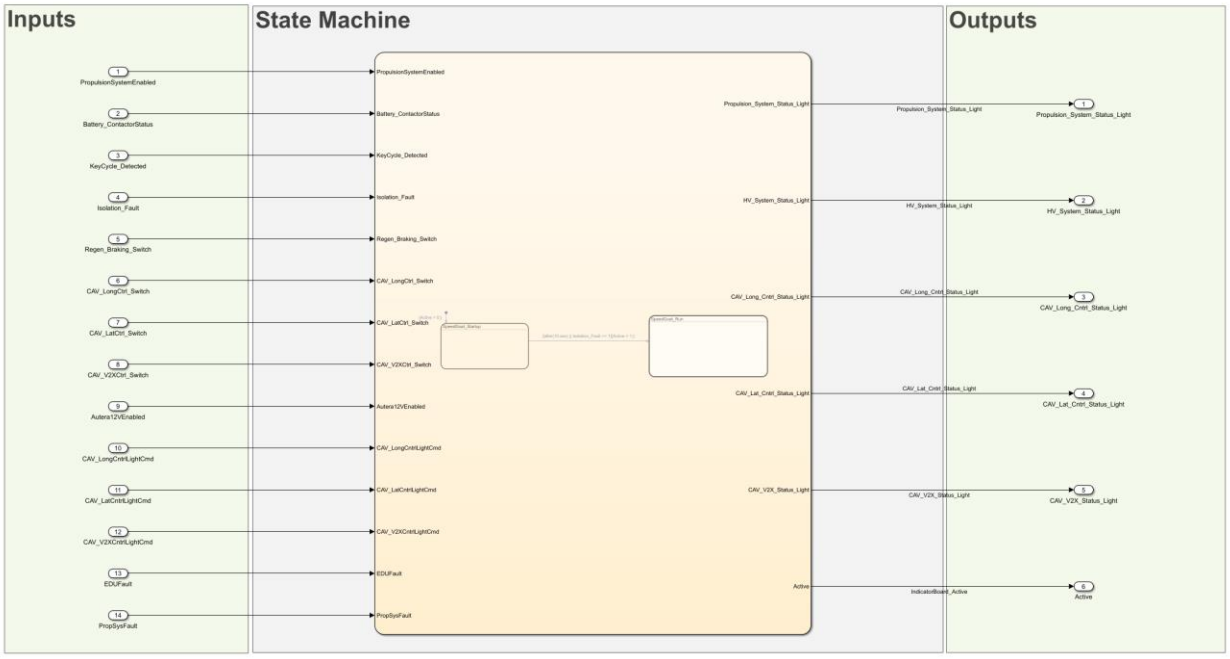
Indicator Light	ON (Solid Light)	Flashing	OFF (No Light)
Propulsion System Status	Illuminated when any drive mode is enabled and will respond to control inputs. Full torque available. E.g. The driver shifts from parking to drive	Flashing when faults are present that prevents the use of the propulsions system.	OFF when the vehicle is unable to respond to control inputs.
HV System Status	Illuminated when the Energy storage system contactors are closed and Energy storage system is able to provide current to a drive system, without fault, or facilitate on-plug charging. (This includes off and charging conditions)	Ground Fault Indicator: Illuminated within 60s of when the vehicle ground fault monitor detects a ground fault.	OFF when Energy storage system Contactors are open
CAV Longitudinal Control Status	Illuminated when the longitudinal CAV system is physically connected to and able to communicate with the CSC, i.e. (Long. control system is enabled and ready to provide commands.). Must be illuminated to initialize any CAV feature.	Flashing when faults are present that prevents the use of any longitudinal control feature.	OFF when the “CAV Longitudinal Control” switch is in the off position.
CAV Lateral Control Status	Illuminated when the lateral CAV system is physically connected to and able to communicate with the CSC, i.e. (Lat. control system is enabled and ready to provide the features.). Must be illuminated to initialize any lateral control features.	Flashing when faults are present that prevents the use of any lateral control feature.	OFF when the “CAV Lateral Control” switch is in the off position.
CAV V2X Status	Illuminated when use of V2X information for CAV features is allowable.	Flashing when faults are present that prevent the use of any V2X data retrieval e.g., disconnected C-V2X radio.	OFF when the “CAV V2X Control” switch is in the off position.

3.2. Explanation For ON/OFF Switch

Switch	Purpose	Engaged/ON	Disengaged/OFF
Regenerative Braking	This switch is used during On-Road Safety Evaluations to disable regenerative braking to ensure the vehicle is able stop solely on hydraulic friction brakes.	The vehicle is able to utilize regenerative braking as a part of the normal vehicle control strategy.	The vehicle shall never use traction motors to apply negative torque during braking maneuvers.
CAV Longitudinal Control	This switch will allow automated longitudinal control commands to be sent by CAV Supervisory Controller(s), enabling operators to initiate automated longitudinal control features. Once the switch is turned on, operators will still need to initiate features using other buttons in the vehicle.	Vehicle longitudinal control features are allowed and may be initiated by operator.	Vehicle shall operate as if it has no longitudinal control features. - CAV Supervisory Controller(s) shall not send messages to propulsion system controller that would request longitudinal control actions. - Propulsion Supervisory Controller shall not send any automated torque requests to propulsion system components, nor shall it use any CAV Supervisory Controller(s) data to adjust propulsion system commands - Perception and connectivity systems shall still operate and report detections, if engaged.
CAV Lateral Control	This switch will allow automated lateral control commands to be sent by the CAV Supervisory Controller(s), enabling operators to initiate automated lateral control features. Once the switch is turned to on, operators will still need to initiate features using other buttons in the vehicle.	Vehicle lateral control features are allowed and may be initiated by operator.	Vehicle shall operate as if it has no lateral control features. - Propulsion Supervisory Controller shall not send any automated torque requests to propulsion system components, nor shall it use any CAV Supervisory Controller(s) data to adjust propulsion system commands.
CAV V2X Control	This switch will allow the use of V2X information for CAV features when available. This switch should have no effect when longitudinal control switch is turned off.	- Vehicle CAV features requiring V2X are allowed and may be initiated by operator.	- Vehicle shall be able to engage longitudinal and lateral control without V2X features if permitted per control switches. - Propulsion Supervisory Controller shall not use any V2X data or control requests coming from the CAV Supervisory Controller(s) - Perception system shall still operate and report data.

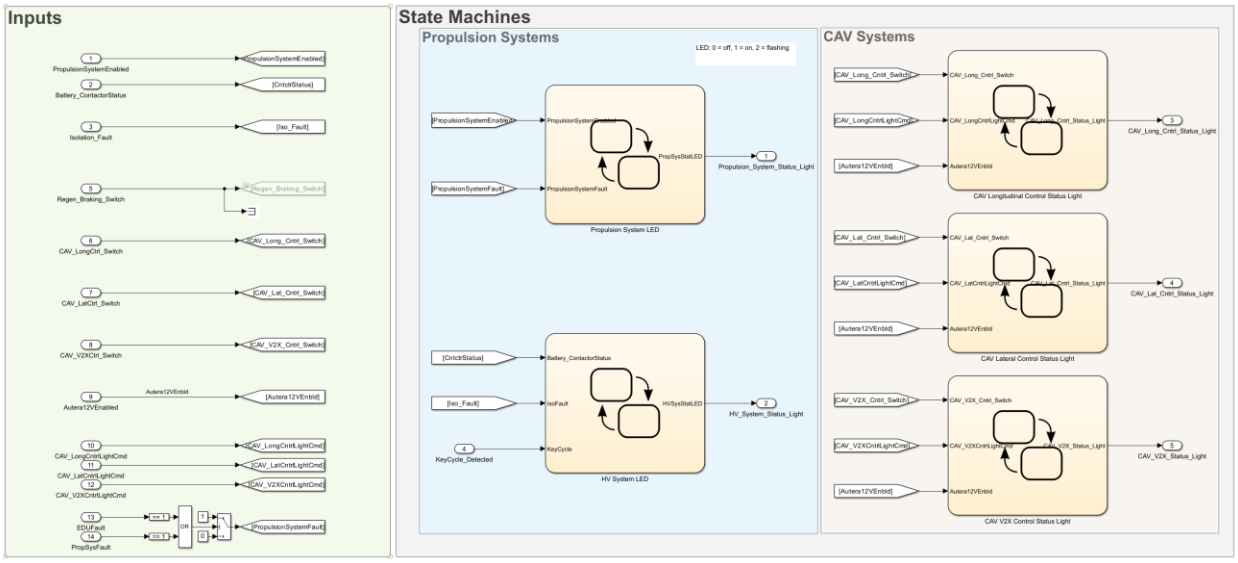
3.3. Simulink Model Overview

The provided Simulink Model is for the indicator board interface feature in our team controller. This model reacts to inputs from the indicator board switches, and using data from the vehicle, sends control signals to actuate the LED lights on the indicator board.



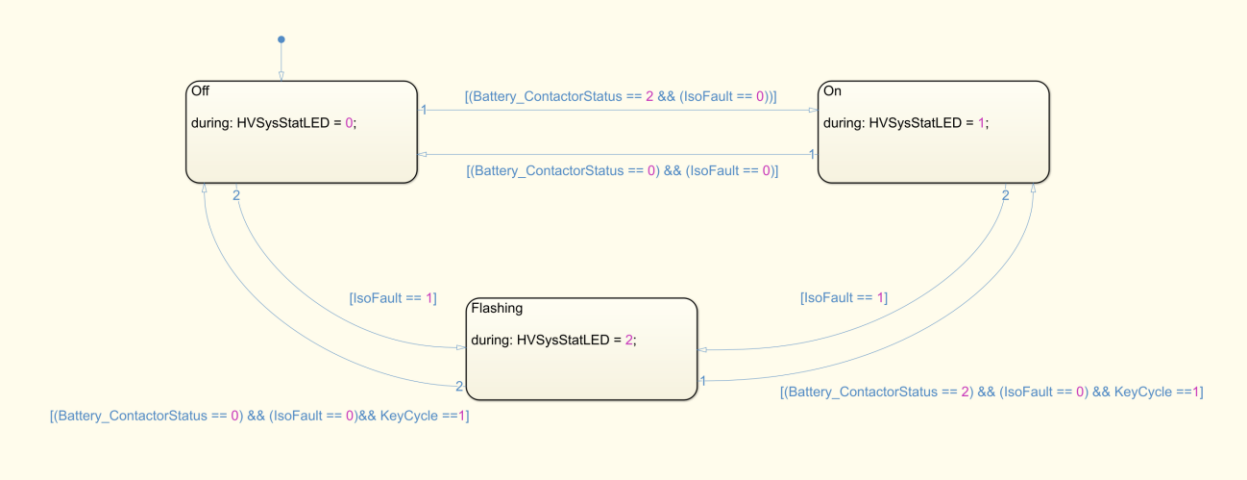
The state machine consists of two primary states – a startup indicator and the main control code. The startup indicator sequence is run every time the controller turns on, flashing a light sequence once the controller software has fully loaded. After 10 seconds, or if there is an isolation fault detected on the HV bus, the 'Active' output variable is set high, and the run state is entered.

3.3.1. Run State



The run state is a Simulink state within the Stateflow chart, consisting of Simulink block logic, and 5 additional sub – Stateflow charts, one to control each LED. Within these Stateflow charts, logical checks are completed to determine whether the LED light should be ON, FLASHING, or OFF based on the values of the input variables.

An example of this is the HV System LED Stateflow chart, shown below.



If you are unsure of what a specific block does, you can right-click on the block and select ‘Help’, this will open a MATLAB help window with detailed documentation on the specific block and how to use it.