**Co-op Guide to Onboarding Process**

**Overview**

Welcome to Linamar!

To prepare you for the start of contributing to our goals and improving your skills in the workforce, we have compiled a document to help guide you through the onboarding process.

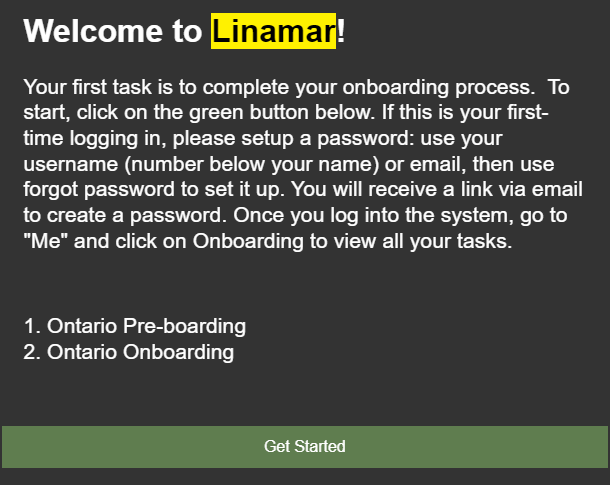
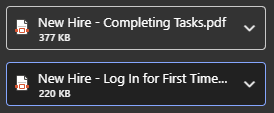
If you have any questions, please don’t hesitate to ask your supervisor or a co-worker. Not only will the onboarding process be finished sooner; we are aiming to create a healthy, growing environment where questions are encouraged!

## Beginning Steps

**Set up workspace**

Along with your acceptance into the position, you should have received an email from Oracle HCM inviting you to complete your onboarding process.

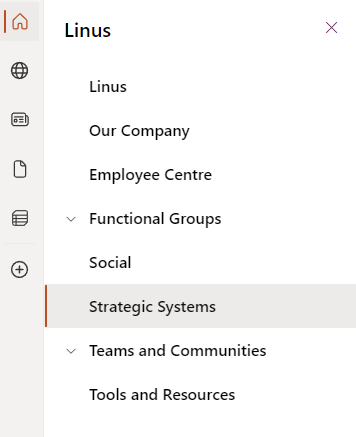
Your name and Linamar ID should be included the email.



HR should provide you with pdf files attached to the email indicating how to log in for the first time and how to complete tasks in Oracle (Right photo above).

You will need to ensure that your work email is listed as a Linamar email, or logging in to services through Linamar might give you trouble later on.

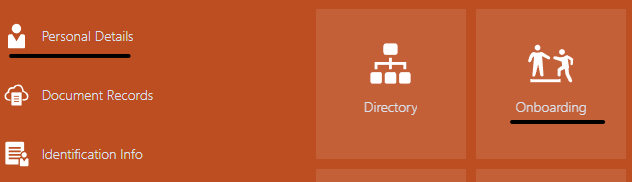
To navigate to Oracle, visit Linus ([Linus - Home (sharepoint.com)](https://linamarcorporation.sharepoint.com/sites/Linus)) and click the Home icon at the side bar. Once the drop down opens, click “Strategic Systems”. Locate “Oracle Human Capital Management (HCM)” and click the hyperlink.



Or just follow the link below:

[Sign In (oraclecloud.com)](https://login-epmd-saasfaprod1.fa.ocs.oraclecloud.com/oam/server/obrareq.cgi?encquery%3De64GqWNWJheDI0mJTo4cJ733gKteMj3dW%2BIQfO5RJ5PcZln1rFTH76BNCuaLKm9hI693804ZCtfVZJQSdy6AA%2FSsk7urgtXmWlA4XW4BMVu3g15bprRXLkyVB5w2lz0IsecL00WiiIJB2eyqaM8HEmCW5r3RGyP9oQ58cihgHLoD%2BbVqvjKuGWsEYFuuWnyfUqJ77wEZ7tboKPBsY4Qiv67IaEozxBp8IYDAYPn28t6sGGY5RMHloXgR3DfUJzZaqNTwy7oBH2v0rNR7XXRYStcd3nIpAK%2BtJPspBgrb6hqF2vfbn%2Fc5FcBlZsKh5sWFtq5Hwg5FK0YDGqbNiZZ92G%2FgJplbDLsTrfGj2bd75%2Fdl9kq4HLRDkOuoG5ICHjGA%2FY2V9X4f9G4AsVsHD7LpyOFcIZ4y1d%2F79R%2F6byh24kzHbeJvMYhlC3syuB%2F38k0MIF%2FX5MjqHxbxeLDPhzXGk8qdHk65xyDmCUNCbCrWpEAAmQuhu%2B6T4c5MoNlP7PiTQeRjjI9MWb67XZFaunqf%2FwPyU4BwaBeDgsnt5yXSQxMrpW0IkW2Vyb9dXcSdeR9BI5fSNZflbbTJ6h9Qnap2ixMloqIVWsEJyg5hT4DmP8elYLvRzN%2BB%2FKSea2QC3AXkamPBVO2hiszlU1VO%2FE5fSnBEU94mj6rYLAPVO5YNkf%2B5CL9TLiSq0WyR7mlRestZNU8M4P0dxYv6e0ff2uw6jxEooKdkKTrGjjM1lE21sAPsuwEIu9HhJYSSsMZy%2FsyjPKL92Vq%2FaSU5OKR813GwqdLKoUT3UT2agfNPE2gD%2BR3HaqrojtjiP0CV6lPhvAr1%2BGvJBPDMF9OuNW%2BFdpy0SpFEdgUnUhPztisYo3sA7eq04zjK1ZfzvJqqea94M96WslS%2FHS0Oc1ec%2FdwFUbf3CQVhyb%2B3XpvfoNpMTUcExnWrYFt1ZlbvfV6VMbC8w7%2B9S2mSk6uTxHwuZwLZ%2BdbWqjP9bMa%2B0FP24cM76ioRSy6f%2Bwa3WMR1B7YsOHGAEBEeUdIoMGDCnCcUZfBqF4YBYQO9Fj9la5ZSd4sDrI3DkzVDyScWifwcs5M6jyr82UUHYda5kKcODR2fIWBo%2F56josd9Aep4fiZ8wr%2FXZyqo7bEJvuDqVPVoFKGMVNdo0YfGDiZ7hKkPETzCtaOd7d2I3G%2FAZEODvCno21%2Fxy0qsSAO1gpWpmDMDfxU7hr7fdozS%2BEGdT0MjQp2Ik5NhphyI%2BIfmv8OMBvJ5fS2WMzAX3zmCWik1uYLb67ABAwNuvBnkcknTid2g13zCeHfYJUCQNx%2FM7dKOrHl9LVzJoBZKcCCD87qvCDRmm%2FZKrGHQnEGZiOMmbYReYkSoqcFpj2Pb5b6sI1ABd%2BCpP6uYrYnq4ifGARw31b%2FAJ1qz%2FLpn37hUFjoiN2yVH52po6GTWaL19tV9qiP%2FcesQxvJsUj%2FQA7HVuB00TVaLwGM8qyrwomtkOD0XpJ3XTeSsRWjnNmnIVYiWTIEaeFOaZ8lUkD70bX9rO7FPwnWS4CGrfDwxqDK6vGJQbAuuFMebACqDD%2BIXDIrEkqfHkJJI00CVugkANQwPluFbO20TRVMi81%2FnaRraM38PnMzLn8JDH%2Bk%2FGXp35PfOWLknXCG9dL0y5kltp4kLuT8SNHr5orCT9qyxewRvrOUW%2Bn1EIb%2FBsYnDQ5Rqrw%3D%3D%20agentid%3DOraFusionApp_11AG%20ver%3D1%20crmethod%3D2%26cksum%3D5f63033d3b4c67fcc3727bb485087e2d1aa21c43&ECID-Context=1.005vGjIvzjg2ZNgLlEc9yZ00028L0009nx%3BkXjE)

Once you are successfully enrolled in Oracle, you will need to complete any tasks related to pre-boarding/on-boarding that you receive, and fill in personal details that are required.



In office, you will begin setting up your workspace.

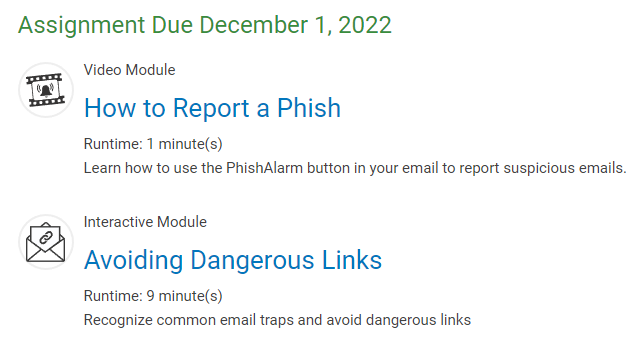
A bundle of paper with instructions will be physically given to you, which you will follow for…

* Microsoft Products setup (Multi Factor Authentication for Microsoft)
  + Email Setup (including 2fa)
  + Teams Setup
  + Adobe Acrobat Setup
* Password naming conventions
* Compatibility mode for browsers (Internet Explorer)
* Computer preference settings
  + Hardware such as printers, headphones, etc.

…and so forth.

Throughout the next couple of weeks, you will be emailed a cyber-security training assignment to go through as well. Please complete within the deadline.

<https://linamar.wombatsecurity.com/>



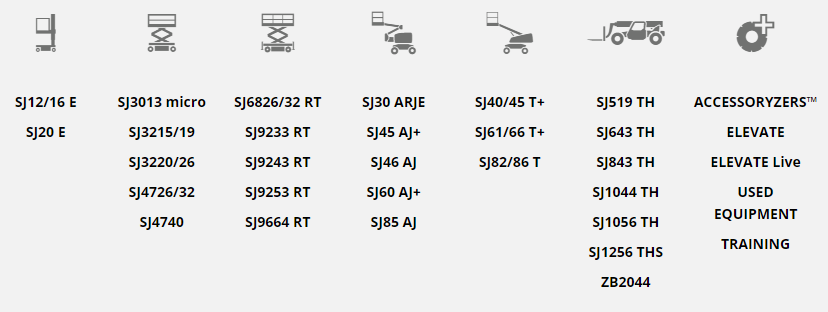
**Machine Familiarization**

Once you are comfortable with navigating your computer in the Linus homepage and all the onboarding tasks are finished, you can begin understanding the machines.

Machines or MEWPs (Mobile Elevating Work Platform) will be the focus of your time and hard work here at Linamar. It is important to get a strong foundational understanding of each design available on the market.

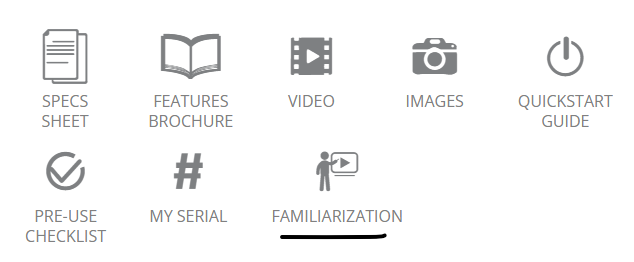
Visit <https://skyjack.com> and position your mouse over the **products** drop down.





Click and read the summary under the overview section for each machine type to learn their purpose and why they differentiate from each other.

Each machine type should contain a familiarization video on certain models in a subcategory.



Please go through each video to get a visual understanding of the basic functions and interlocks of the machine.

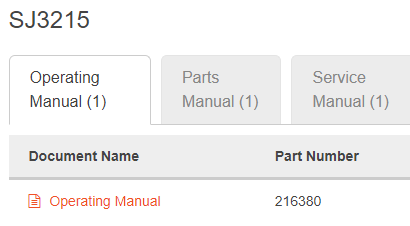
Feel free to explore the website or to play around with the interactive models at the bottom of the webpage for each machine type to review your knowledge.



Since the video is only a brief overview, it is recommended that you read some documentation and try to relate it back to the familiarization to deepen your understanding.

Under the documentation section of each machine model, you can find operating manuals that go more in-depth.





**Plant Tour**

Once you have a solid grasp on the function of the machines, be prepared to see it in-person at the plant!

Your supervisor will plan a day for you to participate in a plant tour, where you will see the parts being welded and the wired connections that contribute to the I/O and ECU communication.

To prepare for the tour, consider the following safety gear you will need to wear:

Safety glasses Ear Plugs Safety shoes

** **

Safety glasses and ear plugs will be given out to you.

Please wear appropriate clothing for the plant (reference agreements on dress code).

For safety shoes purchase, please make sure that a CSA green triangle is visible on the shoe to indicate sole puncture protection for factory standards. If the price is under $113.00, you will be fully reimbursed by HR with receipt proof.



Try looking at your local Walmart or Mark’s for purchasing options, or ask HR for a voucher for the work authority store near Linamar and get the shoes there.

Enjoy the tour and ask questions!

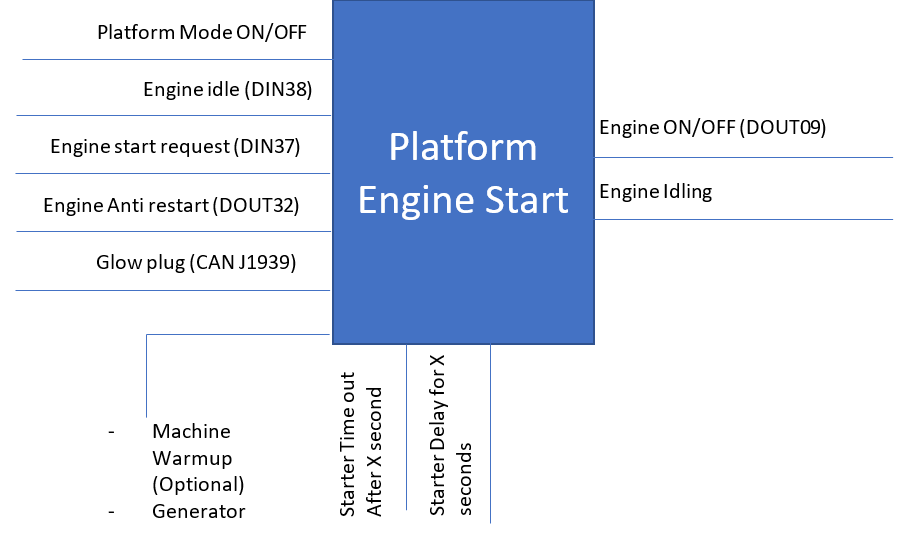
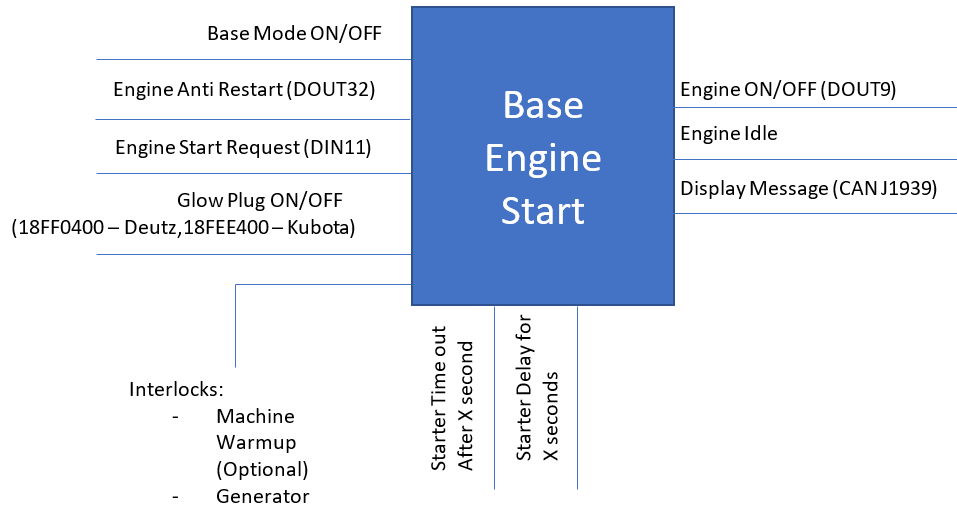
**Review SRS/Theory of Operation**

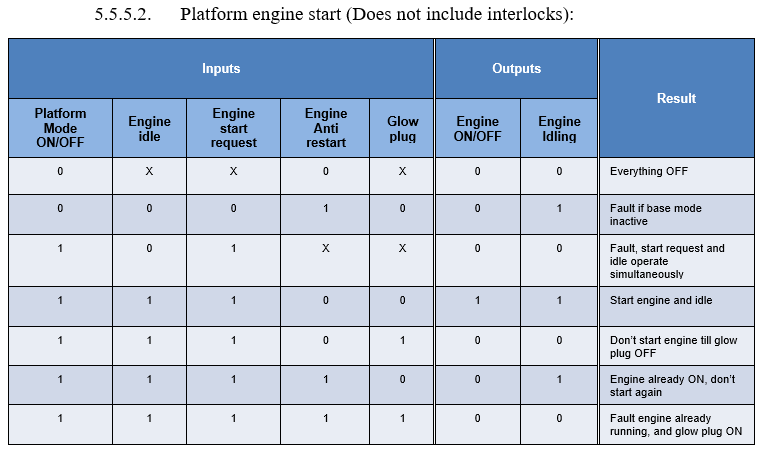
The supervisor should assign/ask you to read through a specific SRS (Software Requirements Specification) or a Theory of Operation for the specific machine they want your focus on.

Location: U:\Engineering\Design\Innovation\General\Projects\In Progress\Model-Based Design\In house controller\2.0 Functional Description & SOW\ 8XT\_85AJ\_SRS\_V0.3



This is not only to connect the information to your understanding to the material, but to also introduce documentation that you will be referencing in the future.

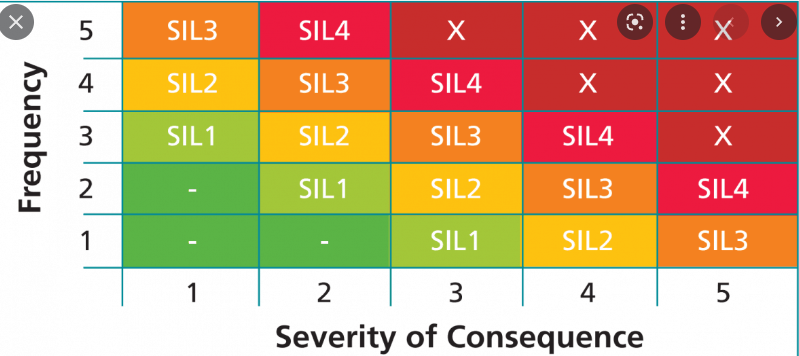
****The ECU (Engine Control Unit) is coded depending on the inputs, outputs, interlocks, dependencies and truth tables located in the document. Therefore, the documents are vital to the creation of the MEWP.



Review each subsection briefly to get a good knowledge base on how the software would impact the results of the I/O combinations.

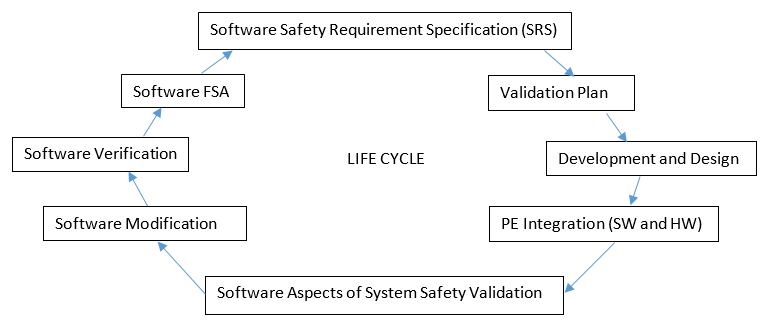
**Summary/Review of IEC 61508**

The IEC takes a high precedent to not only the software lifecycle, but how safety is viewed and treated during creation. SIL level is at a higher risk priority because working with machinery can lead to injury/death.



Safety Requirements are an integral part of the lifecycle in development and integration, and will be developed throughout every stage of the process. Validation plans and test harnesses will also be used in abundance to test the safety requirements themselves and follow standards.

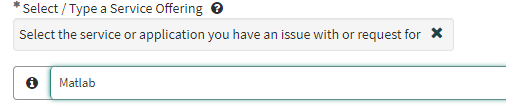
You will be handed a specific version of the IEC [IEC 61508] to read through and summarize your findings. The summary should include the objective of each stage and the requirements associated.

Be sure to include memorable images explaining complex concepts, and encapsulate the main focus of each stage. The point of the summary is not to summarize, but to present your understanding of the standard in a shortened manner.

Once the summary is done, please present your findings to your supervisor/team. Don’t worry, it should be an informal session of you passing on your learnings to your superiors to make sure that you understand the basic idea of the IEC.

**Matlab license/installation**

Now that you understand the importance of safety in the development cycle, you will need to implement your learnings. Your supervisor will provide you with a development/test license for you to install. If you require assistance with installation, use Service Now to make an incident request for Matlab installation.



After the installation is complete and the path is correctly set, it is time to start the Onramp.

**Matlab Onramp**

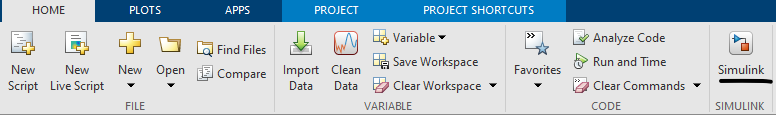
Complete the Onramp lessons for both Matlab and Simulink – it is designed to ease you into using the software as a complete beginner.

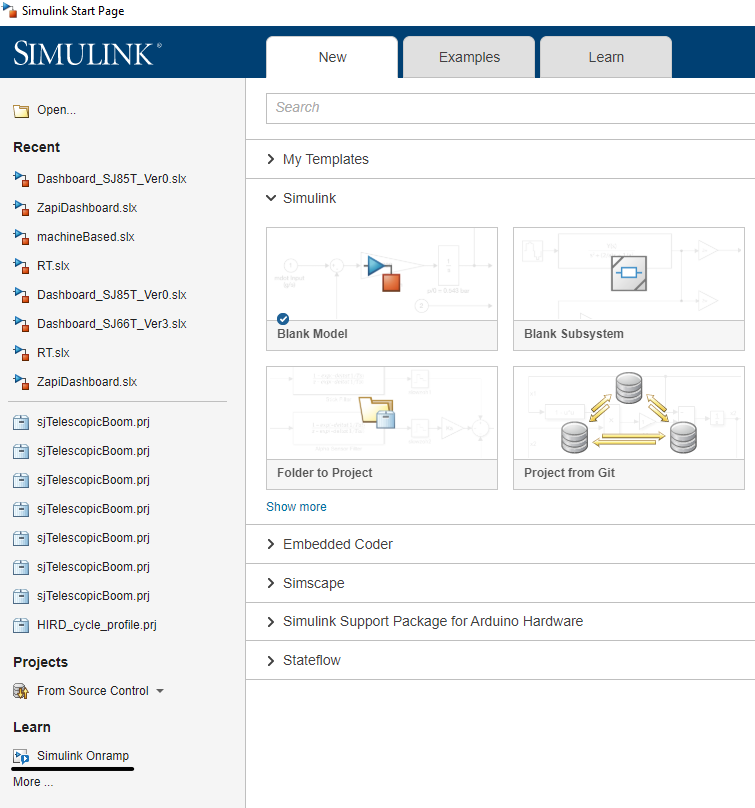
The Onramp for Matlab can be found the [www.mathworks.com](http://www.mathworks.com) webpage using the search bar. It will use the online version of Matlab to grade your answers:

[MATLAB Onramp - MATLAB & Simulink Tutorial (mathworks.com)](https://www.mathworks.com/learn/tutorials/matlab-onramp.html)

**Simulik Onramp**

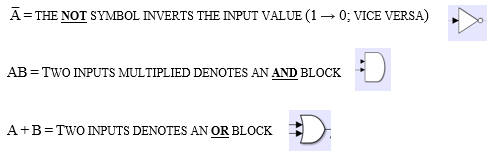
After Matlab Onramp is completed, open Matlab locally and access Simulink. Once you land on the Simulink Start Page, at the side bar you should spot the Simulink Onramp tutorial under “Learn”.





To ensure you retained your knowledge, try some practice questions using Cody or any practice problems online!  
[MATLAB Cody - MATLAB Central (mathworks.com)](https://www.mathworks.com/matlabcentral/cody/)

Simulink will be an integral factor to generating the code through logic gates, so keeping the basic blocks in mind for understanding the logic gates will be important.



If you are developing logic circuits and have a test license, you will need to know how to create a test harness, to eliminate manual testing with truth tables.

Follow the PowerPoint provided for guidance:

U:\Engineering\Design\Innovation\General\Projects\In Progress\Model-Based Design\In house controller\7.0 Presentations\Test Harness Guide\_v2.0.pptx

# Coding the ECU

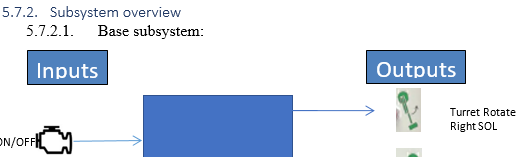
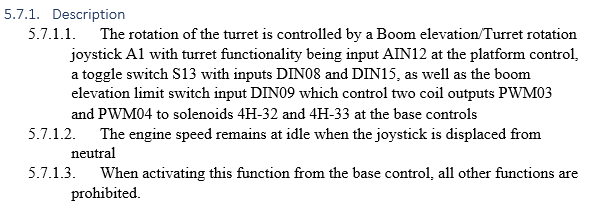
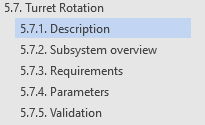
**Mapping architecture**

Once you have mastered Simulink and Matlab, you are ready to incorporate functional blocks into logical circuits for the controller.

When starting to build a function block, the SRS/Theory of Operation is an important tool to be used as a guideline with not only I/O, but validation truth tables as well. The logic will dictate how the machine will function.

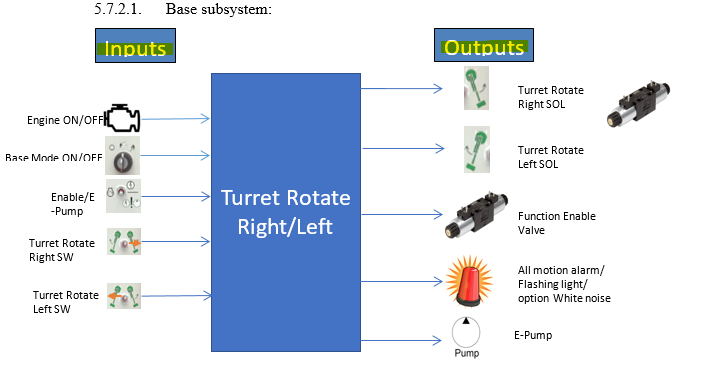
The document is used as the blueprint for Simulink development.

Go to the heading of the block you are focusing on building, and read through all the *Description* and the *Subsystem Overview.*



The *Subsystem Overview* contains the inputs and outputs that the machine will utilize to complete each function. Familiarize yourself with the hardware and why it is activated for the according block.

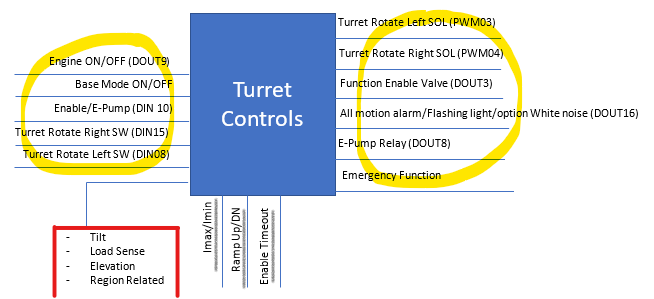
For example purposes, we will only examining a base subsystem, yet there should be a platform counterpart.



Once you’ve understood the inputs and outputs of the subsystem, proceed to the *Requirements* the Simulink model needs to achieve.

Make sure to focus on the *Requirements* sub section, as it contains all the I/O pins/signals that will be incorporated into the logic, parameters that affect certain resulting outputs, and interlocks that influence said function.

The impacting interlocks [boxed in red] as well as the parameters [underlined] are included to demonstrate their effect on the logic development.



To check parameter range and description, check under the *Parameters* section of the functional requirement.

Once all resulting conditions and edge cases have been established and the information is ready to be incorporated, the logical circuit development begins.

**Opening the ECU model**

The ECU model with all logic implementations is stored in the U drive.

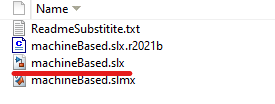
Location: U:\Engineering\Design\Innovation\General\Projects\In Progress\Model-Based Design\In house controller\2.0 Functional Description & SOW\SoftwareDevelopment

Open Matlab and set the file path to match to the latest version of the archived Simulink model. Click the folder with the green arrow near the file path.

NOTE: Please do not set the file path in to the U drive. Copy the archived folder onto your personal drive to avoid overwriting existing changes.



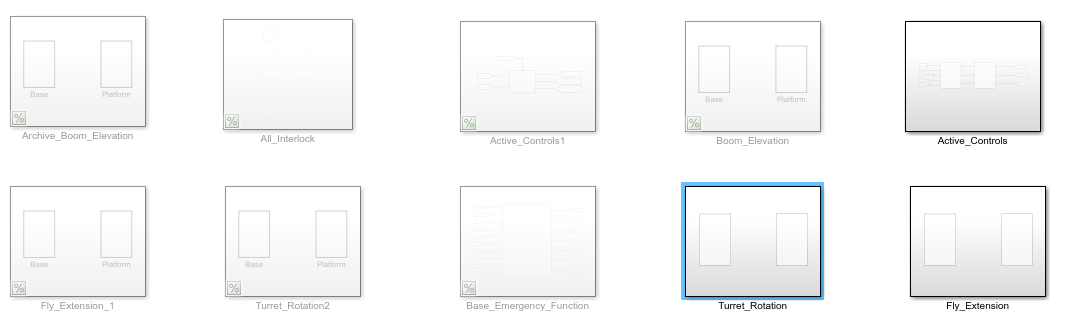
Locate the file named machineBased.slx which holds the design model of the logic diagrams inserted into the ECU. Double click to open it, and start at the highest level model.

**Connecting I/O and Parameters**

Once you have the model page open, start navigating deeper into the submodules. Double click the submodule block to enter it. Please navigate to the FunctionBlock submodule level, where all the functional logic and interlocks are stored.



****

Add a submodule of the function you are working on.

Check the *Requirements* section of the SRS to see what submodules the logic needs to be split up into, and add those accordingly into the submodule you just created. Name the submodules accordingly.



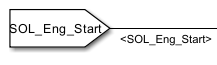
Create a submodule within all the existing modules to hold the logic for each functionality of the ECU. First, we will connect the inputs and outputs.

Incorporate the I/O pins as goto and from blocks that are processed from the Input\_Data\_Processing submodule (connects to Data\_Processing).

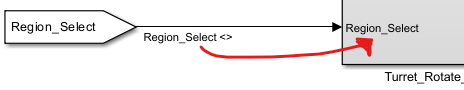
To translate the pin name to the proper signal naming convention, please refer to the excel sheet located in the path below.

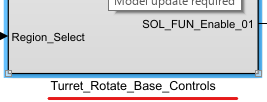
Location: U:\Engineering\Design\Innovation\General\Projects\In Progress\Model-Based Design\In house controller\2.0 Functional Description & SOW\SystemLevel\_IO\_DWG\_New.xlsx

The excel sheet contains all mapping from the actual pin to the named signal. Reference it to navigate the slx model.



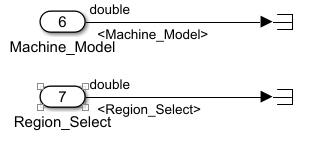
Make sure to name the signal to keep the model organized and keep tracking easier. It will auto name the inputs for the submodule connected containing the logic.



 Connect the outputs using the same naming conventions and methods as the input connection. The logic circuit building can commence if all connections are established.

Don’t forget to name the logic submodule before proceeding.

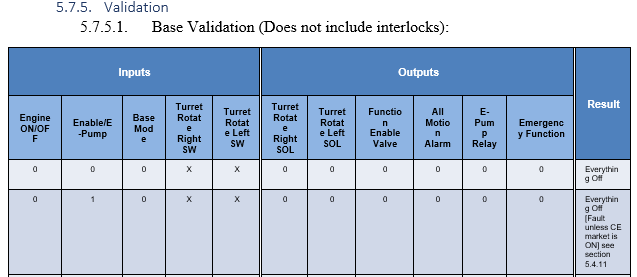
Feel free to use terminator blocks for inputs inside the logic submodule that are planned to be integrated but their role is not yet established (aka parameters) or they will interfere with current development.



**Creating Logic Circuits**

Prior to starting connecting blocks, refer back to the SRS or other blocks to catch similarities between certain functionalities. Most of the time, you can **copy & paste** a circuit (or a part of one) that have the same I/O and logic to save time.

Go to the *Validation* section of the SRS and look at the truth table for the specific function.



If another function’s truth table looks similar, it is also a good opportunity to take advantage of already built functions.

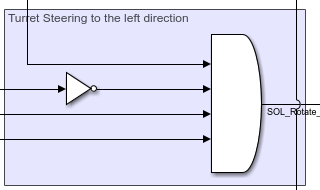
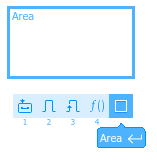
Analyzing all cases that lead to a result can reveal certain dependencies within inputs that directly certain outputs and doesn’t impact others.

Determining which inputs are needed for which outputs can help in establishing the logic circuit and making it as efficient as it needs to be.

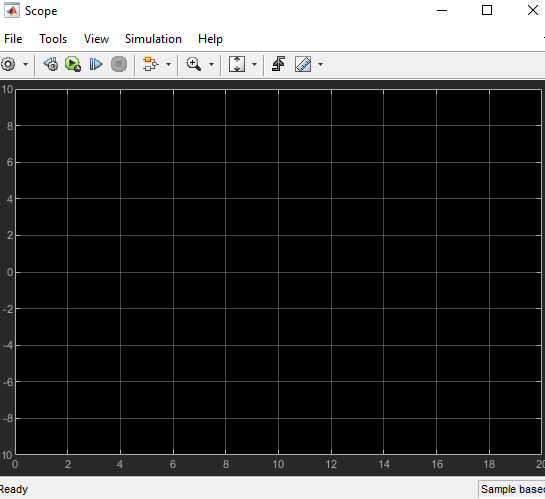
Please reference the document located in the path below to get an idea of how we utilize interdependencies to build circuits:   
<file:///U:\Engineering\Design\Innovation\General\Projects\In%20Progress\Model-Based%20Design\In%20house%20controller\16.%20Lesson%20Learned\Method%20of%20Developing%20Logical%20Circuits%20through%20Truth%20Tables.docx>

Keep the circuit neat and create areas around certain AND or OR blocks to explain which output the resulting signal is leading to.

Click and drag around the block you want to label. A menu should pop out at the bottom of the area. Choose the fifth option labelled “Area→”. Name the area accordingly.



You can unit test while building with the Scope block. Attaching the scope to a signal and running the model. Double clicking the block will open up a separate UI with a graphed signal.





Once you run the program, the scope will show a yellow circled value on the y-axis revealing the value of the signal connected to the scope based on the input and logic.

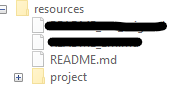
The unit testing can commence by using multiple input scopes or using the test harness with the tutorial specified above.

Once all the results are verified and a test harness report is generated, move on to the next submodule of the function or the next function block.

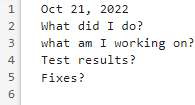
**Backups and READMEs**

Constantly save your progress on developments and update a created README with the progress you have made thus far.

The README update should contain:

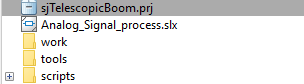
* Date of the update
* Any completed/in-progress blocks
* Current progress of the blocks
  + Awaiting testing?
  + Fully tested?
  + Output of testing
* Any obstacles you encountered
  + Explanation for fix if it is found
  + Error messages
* All progress achieved
* Plan for completion until next update

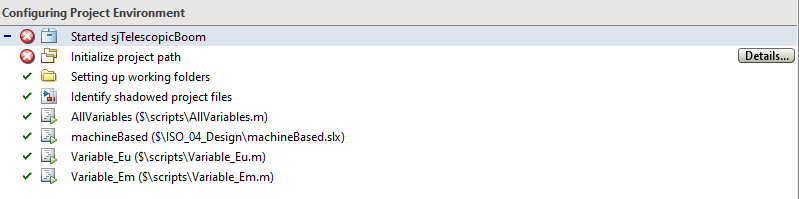
The README file can be found/kept in the resources folder when opening a project instance. It should contain updates of all your work.

Once you finish up updating all of the work for the time allocated and updated your personal README file, you will either save the project or export a backup.

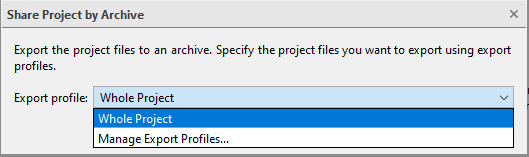
A backup is done once/twice per week to ensure progress doesn’t get overwritten and all updates are organized.

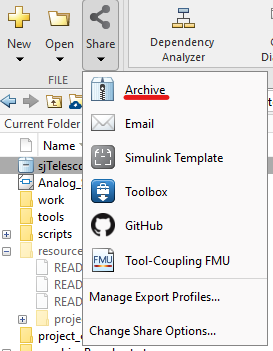
To do a backup, navigate to the project file opened in the MATLAB application. Open the prj tab, and then navigate to the project header.







Press the share icon and choose “Archive”. This should save an archived file with all the folders combined.

When saving, please save in the file path below. Go one folder further into the path by choosing the folder assigned to your name.

Location: U:\Engineering\Design\Innovation\General\Projects\In Progress\Model-Based Design\In house controller\2.0 Functional Description & SOW\SoftwareDevelopment\Backup\[Insert Name]

Ensure the backup is saved correctly, and continue working on the project.

Good luck with your term!