



DEVELOPER'S GUIDE TO PACEMAKER DEVELOPMENT

TUTORIAL 1.2: GETTING STARTED I

SFWRENG/MECHTRON 3K04
McMaster University

September 22, 2025

GETTING STARTED WITH THE MATLAB SIMULINK® ENVIRONMENT

The purpose of this tutorial is to introduce the MATLAB Simulink® environment and provide installation instructions for installing MATLAB Simulink® on a personal computer. **Part A** of the tutorial covers the installation of MATLAB® and Simulink® using the McMaster University campus-wide license. **Part B** of the tutorial covers installing the hardware support package for the FRDM-K64F and the appropriate firmware support. **Part C** of the tutorial covers configuring a basic Simulink® model and exporting the model to a Simulink® template. **Part D** of the tutorial prompts you to take an initial investigation of the Simulink® environment.

Topics Covered

- Introduction to Simulink®
- What is code-generation?
- What is board firmware?
- Installing MATLAB Simulink® and all applicable support packages

At Your Leisure “Model-Based Design for Embedded Control Systems”

1 BACKGROUND

1.1 What is Simulink®?

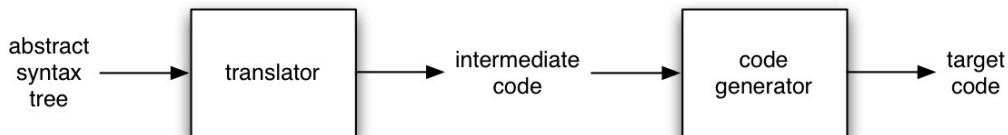
Simulink® is a graphical environment integrated with MATLAB® and a dataflow graphical programming language for simulation and model-based design.

1.2 Why are we interested in Simulink®?

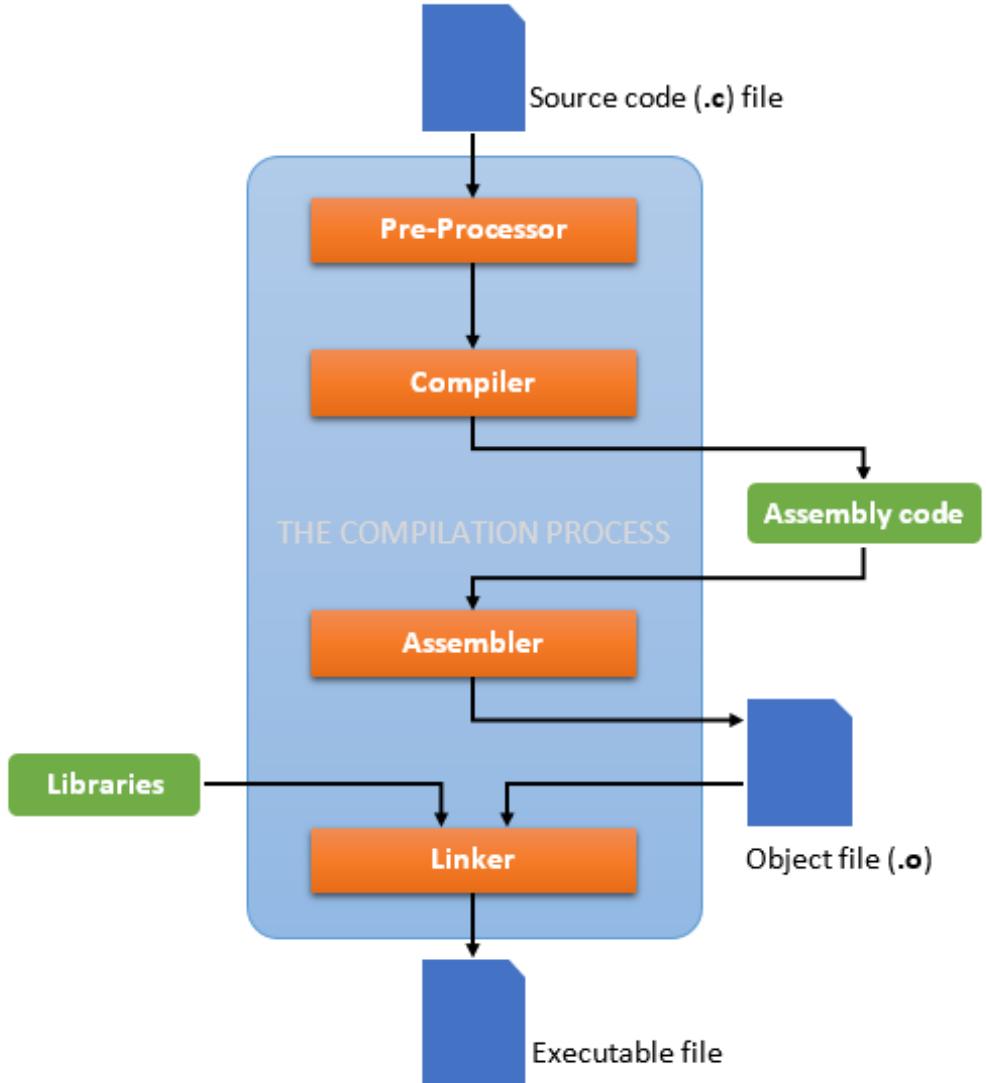
Simulink® is a **model-driven development (MDD)** platform and the leading environment for modelling, simulating and implementing embedded systems. Simulink® supports system-level design, simulation, continuous testing and verification of embedded systems and automatic code generation.

1.3 What is Code-Generation?

A Simulink® model is an abstract description of a system using diagrams. Code-generation (code-gen, for short) is the process of first translating the model into standalone, embedded C code and then optimizing the C code. After the C code is produced, the program may be deployed to the K64F hardware. Compilation is the process of translating the C code and libraries into a single binary file or executable that is flashed to the FRDM-K64F.



(a) Code Generation Process.



(b) Compilation Process

Figure 1: Code Generation and Compilation.

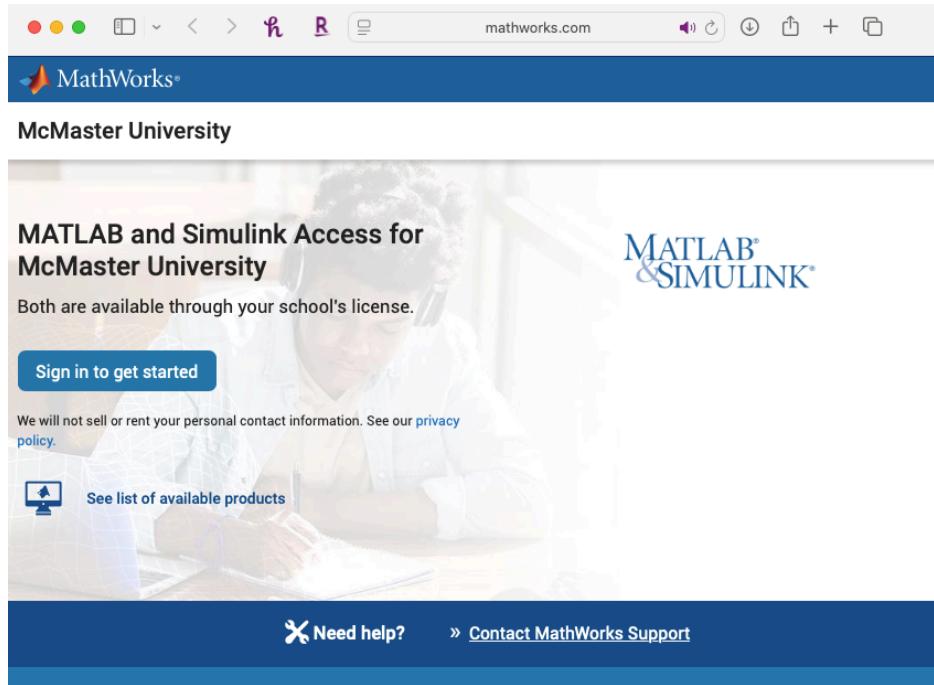
2 TUTORIAL

2.1 Part A — Installing MATLAB® & Simulink®

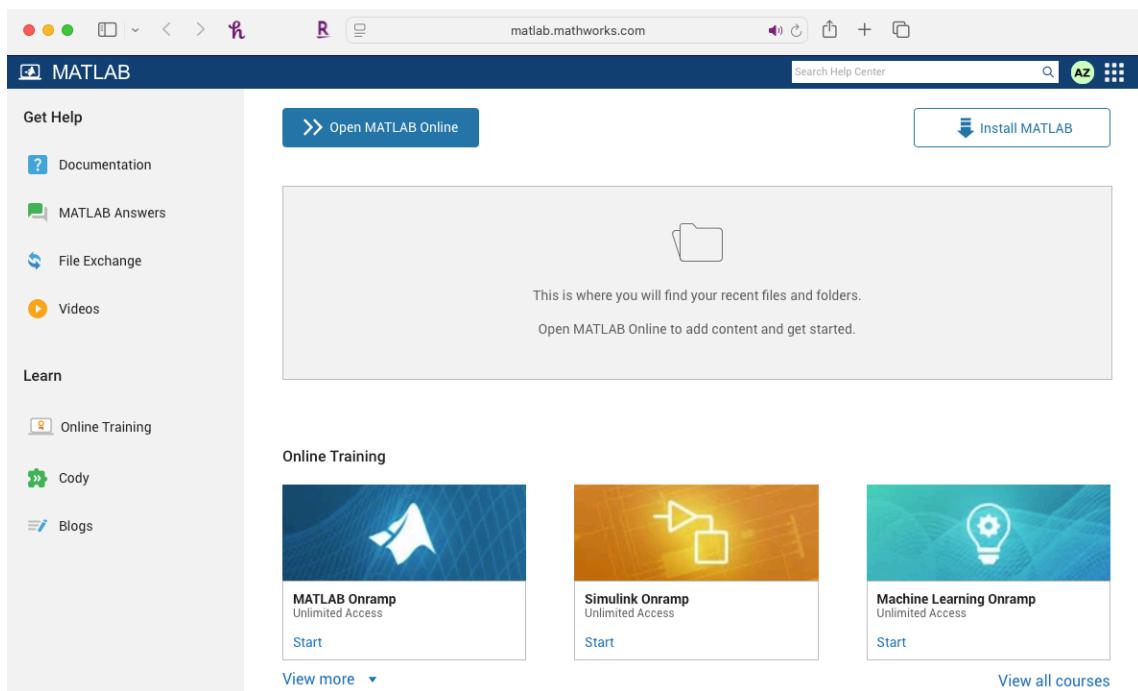
Disclaimer: McMaster students, staff and faculty have free access to a MATLAB® license that includes all the required toolboxes for the project. To install MATLAB® using the McMaster campus-wide license, you will need to have a MACID and a MathWorks® account. If you do not have a MathWorks® account, you will be prompted to set one up during the installation.

Note: If you already have MATLAB installed, it is not necessary to reinstall, although it will take significantly longer. **This software will not work on Macbooks using Apple Silicon ARM chips**

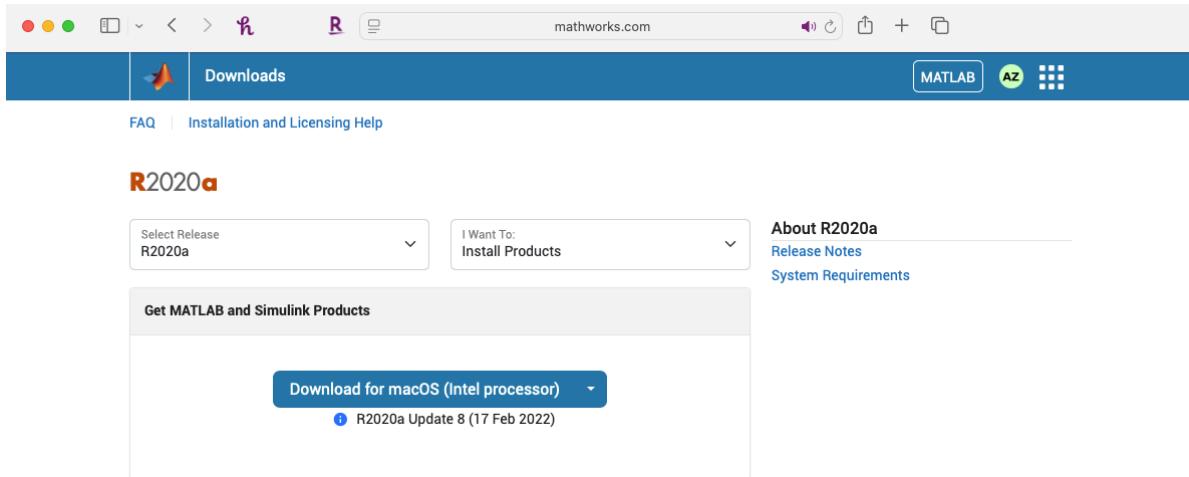
1. Before installing, make sure you have 20GB of disk space available.
2. Click the following link to access the McMaster University MATLAB Access page: <https://www.mathworks.com/academia/tah-portal/mcmaster-university-31501097.html>.



3. Follow the on-screen instructions to create a MathWorks® account with your McMaster email.
4. Click on the top-right button to install Matlab.

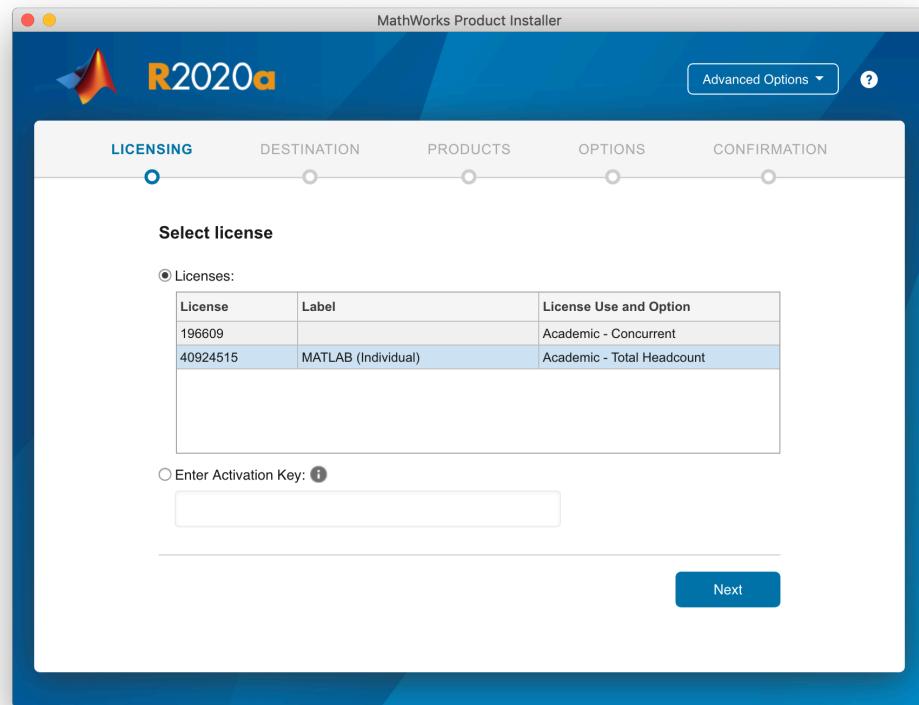


5. Select the release of MATLAB to install. As of the writing of this document, the latest known working version is 2025a



6. Download the installer for your operating system.

7. Run the MATLAB® Product Installer, sign in with your MathWorks® account and accept the License Agreement. When prompted, select the McMaster license. The license number is **40924515**.



! WARNING: Do not skip through the rest of the installation!

8. Confirm your user information on the next page and set the destination folder for the installation. On the “Select Products” page, select the following toolboxes and click “Next”.

Product Name	Download Size
MATLAB®	1095 MB
Simulink®	1539 MB
Embedded Coder®	95 MB
Fixed-Point Designer™	111 MB
MATLAB Coder™	294 MB
Simulink® Check™	38 MB
Simulink Coder™	70 MB
Simulink Coverage™	16 MB
Simulink Design Verifier™	28 MB
Simulink Desktop Real-Time™	25 MB
Simulink Test™	80 MB
Stateflow®	48 MB

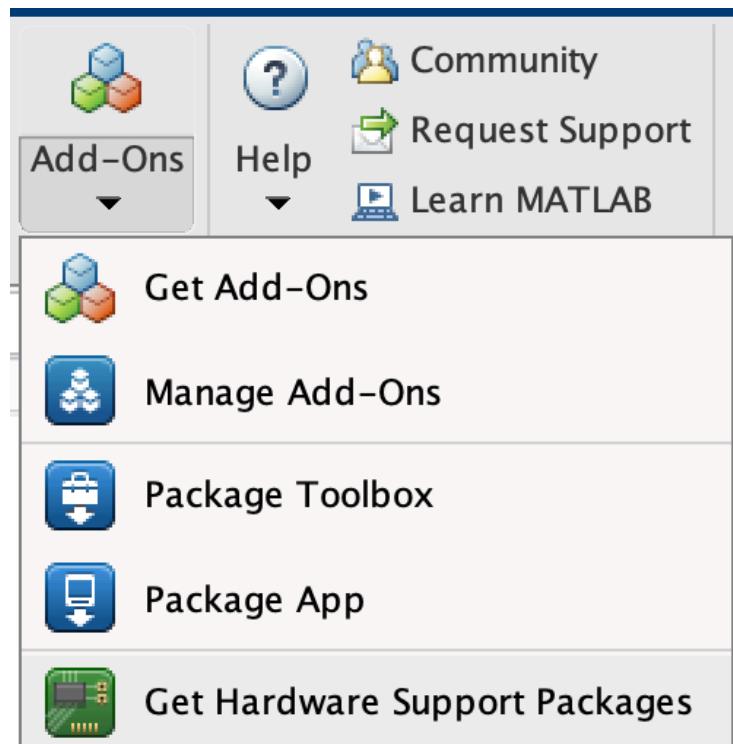
Comment on the above products: the translator in Figure 1a represents Simulink Coder™, and the code generator represents Embedded Coder®. MATLAB Coder™ is a pre-requisite for Simulink Coder™.

Note: if you missed a toolbox, you can install it later within MATLAB®. To find and install the toolbox, go to the **Home** ribbon in the top toolbar. On the **Environment** pane, click the  **Add-Ons** dropdown menu and select **Get Add-Ons**.

2.2 Part B — Installing the Hardware Support Package for the K64F

The support package is an add-on that enables us to use Simulink® with the FRDM-K64F hardware.

1. Open MATLAB. Select the **Home** ribbon in the top toolbar. On the **Environment** pane, click the **Add-Ons** dropdown menu and select **Get Hardware Support Packages**.



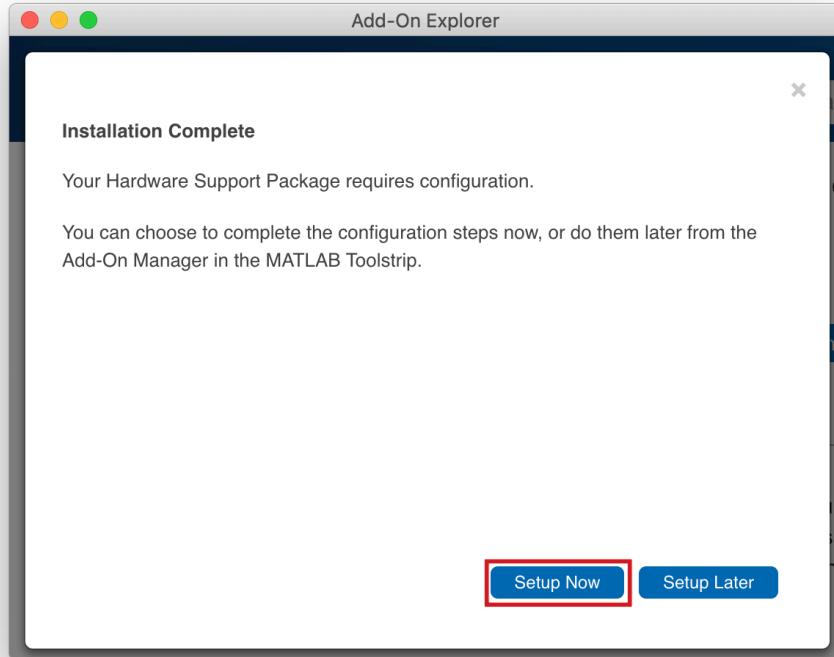
2. Search for the Simulink® Coder Support Package for NXP FRDM-K64F Board add-on using the search filter on the top right of the Add-On Explorer. Install the hardware support package.

A screenshot of the MATLAB Add-On Explorer. At the top, there is a search bar with the text 'Simulink Coder Support Package for NXP FRDM-K64F Board'. Below the search bar, the package information is displayed:
- Image: A photograph of the NXP FRDM-K64F Board.
- Name: **Simulink Coder Support Package for NXP FRDM-K64F Board**
- Author: by MathWorks Embedded Coder Team STAFF
- Description: Generate and deploy code for NXP FRDM-K64F Board.
- Rating: ★★★★☆ 7 Ratings
- Downloads: 13 Downloads
- Last Updated: Updated 18 Mar 2020
- Buttons: 'Learn More' and 'Install ▾'
Below the main listing, there is a 'Download Only...' button.

Simulink® Coder™ Support Package for NXP™ FRDM-K64F Board enables you to create and run Simulink® models on an NXP FRDM-

Requires

- Follow the on-screen instructions to complete the installation. When the installation is complete, click “Setup Now”.



Note: if you will need to set up the support package later, you will be able to access the setup menu by navigating to **Home** > **Environment** > **Add-Ons** ▾ > **Manage Add-Ons**. Click the gear icon next to the FRDM-K64F board support package.

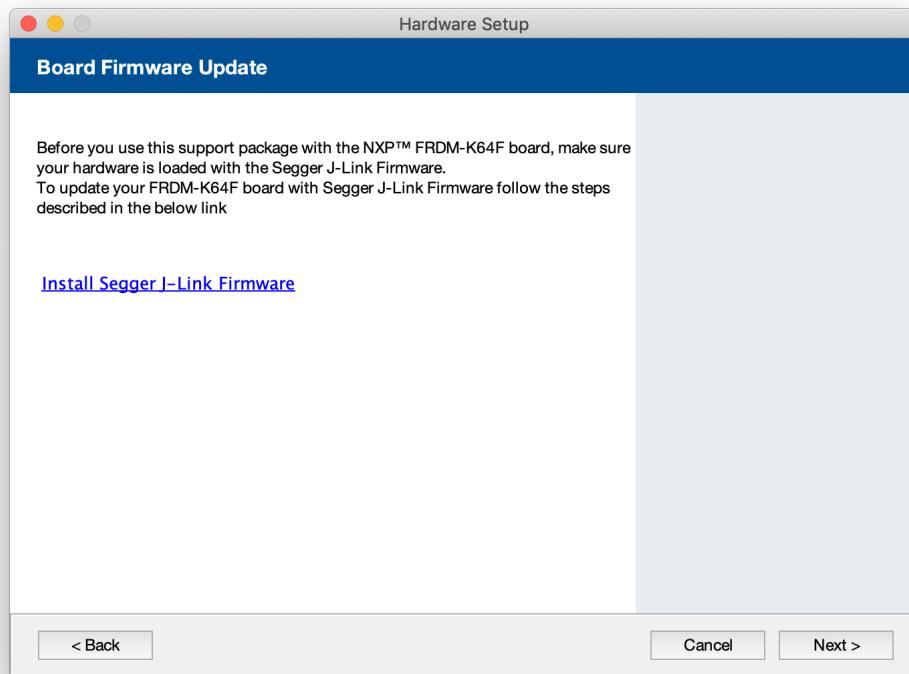
Name	Type	Author	Install Date	⋮
 Simulink Coder Support Package for NXP FRDM-K64F Board version 20.1.0	Hardware Support Package		13 September 2020	

Disclaimer: If you are using a MacOS system with window-snapping or accessibility apps and you encounter a situation where the MATLAB® UI freezes when you attempt the next step, you will need to disable those apps and then restart your computer before attempting the hardware setup. (You can re-enable them after completing the setup).

- On the “Select Firmware” page, choose **Segger’s J-Link OpenSDA V2 firmware** from the firmware drop-down menu options. Recall from Tutorial 1.1 that firmware can be thought of as a computer program that is not intended to change. One of the main functions of the board

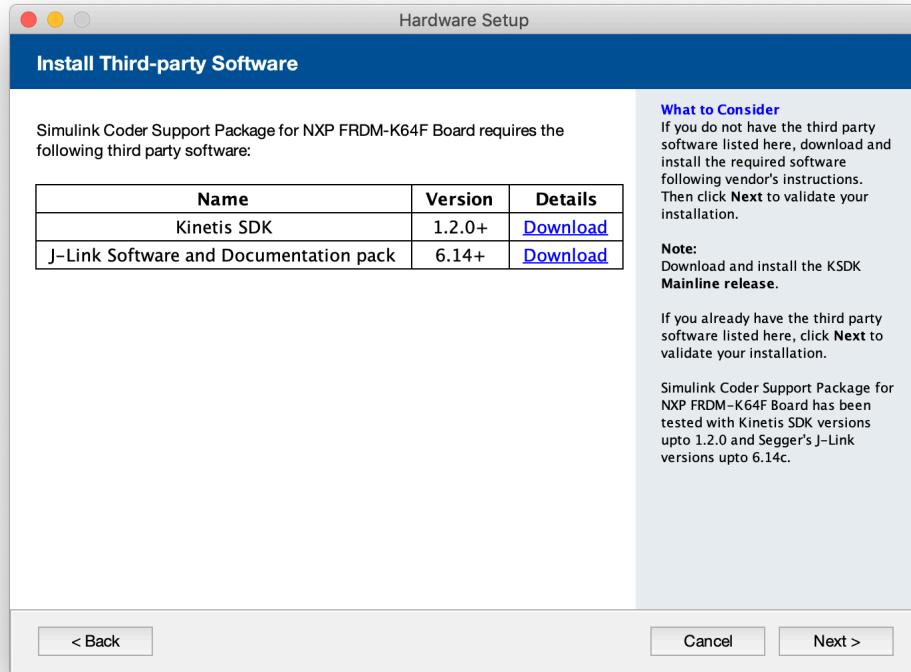
firmware is to enable programming and debugging application with the FRDM-K64F processor from your personal computer. We will be using the Segger's J-Link OpenSDA V2 firmware for the project due to its enhanced data transfer and debugging abilities.

5. Proceed to the next page. Click “Next” to skip the Board Firmware Update. This step has already been completed for you.



6. Follow the on-screen instructions to download the third-party software.
Note: the third-party software makes up the libraries segment of the illustration in Figure 1. Make sure you install **Kinetis SDK 1.2.0 mainline release** and not the newer KSDK 1.3.0 to avoid any issues that may occur. Install the SDK into your home folder or C drive. Also install version **V6.20a** of the J-Link Software and Documentation pack. The total installation size for both software should be about 2.5GB.

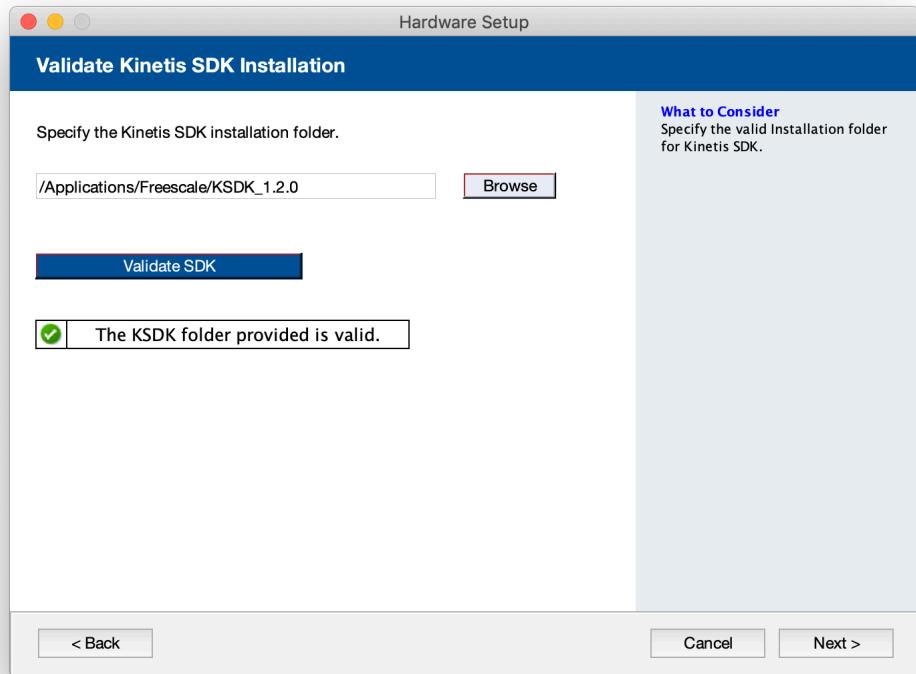
Note: you will need to set up an NXP account to download Kinetis SDK. For the company name, you may enter “McMaster University”.



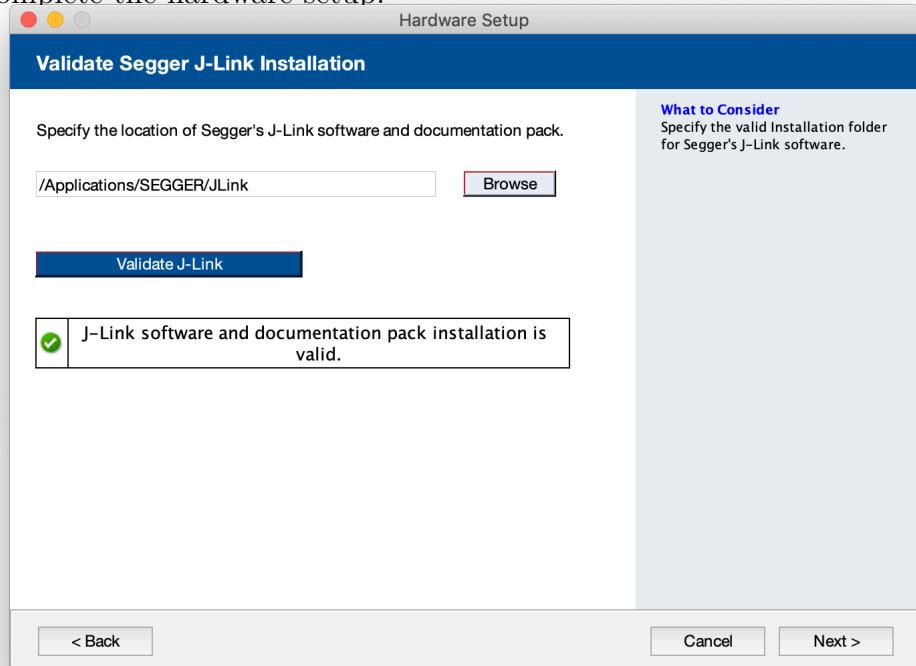
7. If the link is invalid for Kinetis SDK, try this link. Scroll down to the Downloads section, and click on the DOWNLOAD button next to Kinetis SDK. In the Product Information page, change the tab to Previous to find the right version and download the KSDK v1.2.0 Mainline release.

Version	Description	Date Available
1.1.0	KSDK v1.1.0 Mainline releases	Dec 20, 2014
1.0.0	KSDK v1.0.0 Mainline releases	Dec 20, 2014
1.1.0	KSDK v1.1.0 standalone releases	Mar 20, 2015
1.2.0	KSDK v1.2.0 Mainline release	May 6, 2015
1.2.0	CyaSSL for Kinetis SDK v1.2.0 with MQX RTOS releases and patches	May 6, 2015
1.2.0	KSDK v1.2.0 Standalone releases	Jul 6, 2015
1.2.0	KSDK v1.2.0 Board Support Patches	Jun 30, 2015
1.0.0	KSDK v1.0.0 standalone releases	Jan 14, 2015

8. Validate the Kinetis SDK installation.



9. Validate the J-Link Software and Documentation pack installation and complete the hardware setup.



If you are getting a “Hardware Setup Error” go to Troubleshooting

10. The FRDM K64F REV E3, REV E4 and REV F is not fully configured in the R2020a release with respect to pin D8.

As a workaround, in order to enable the pin,

- i. Enter the following command on MATLAB Command window.
Copy and paste the command **exactly** as it appears below. **The command appears as two lines below but needs to be executed as one.**

```
open([codertarget.freedomk64f.internal.getSpPkgRootDir,  
      '/src/mw_sdk_interface.c']);
```

- ii. Within the file that is opened, find the following line:

```
{ GPIO_MAKE_PIN(GPIOA_IDX, 0), MW_NOT_USED},//  
PTA0, D8
```

and replace it with

```
{ GPIO_MAKE_PIN(GPIOC_IDX, 12), MW_NOT_USED},//  
PTC12, D8
```

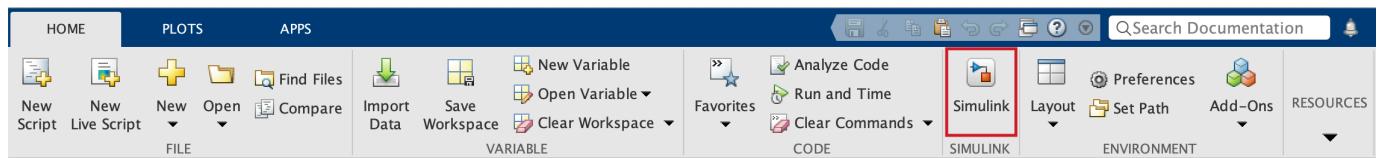
Do not get any of the letters wrong. Double click the entry from the first text field above, copy it and use the **CTRL**+**F** keys to find the line in the file. Remove that line from the file. Then double click the entry in the second text field above, copy it and paste it at the same position in the file.

- iii. Save and close the file.

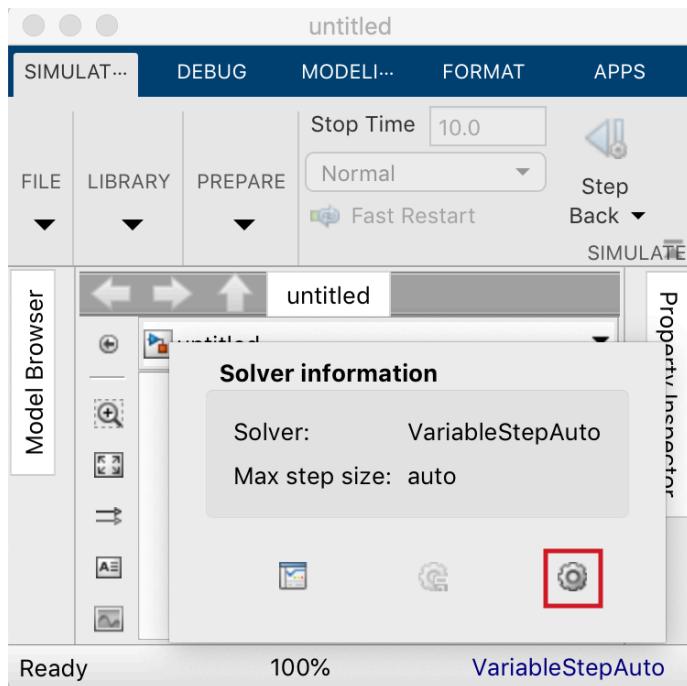
2.3 Part C — Simulink® Model Configuration

Simulink® Embedded Coder generates C and C++ code from Simulink® Models for use on Embedded Processors. Everytime a new model is created in Simulink®, it must be configured to generate code for the FRDM-K64F board. In this part of the tutorial, you will configure a new Simulink model and save it as a template. When you need to create a new model for the FRDM-K64F, you can load the template to reuse the settings of the model instead of repeating the following steps every time.

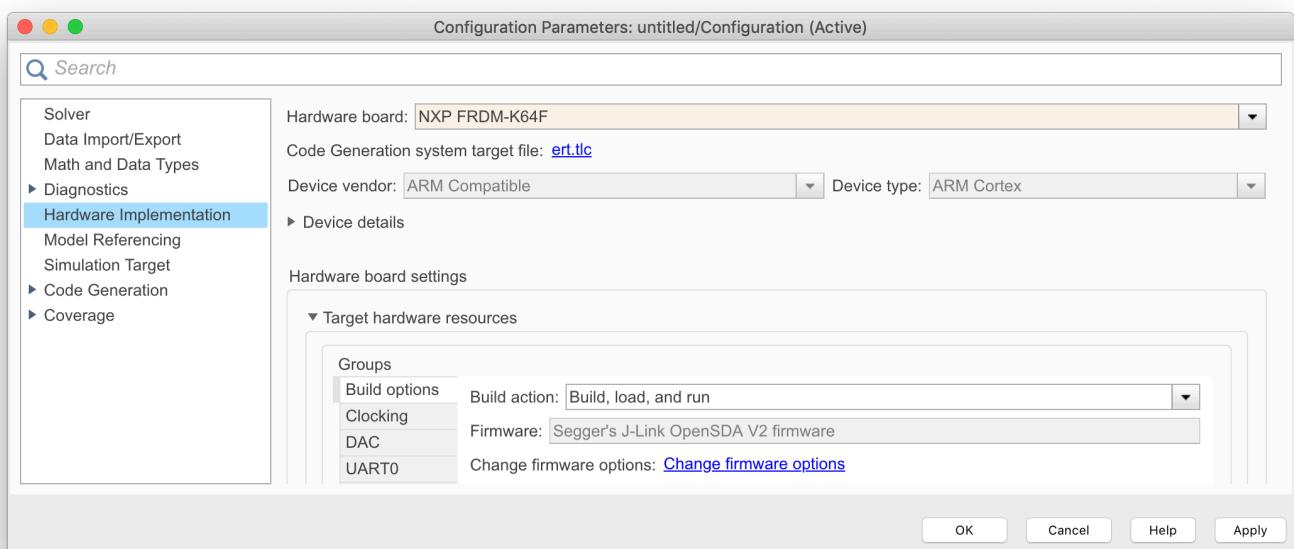
1. Navigate to the Simulink® pane on the Home ribbon. Click “**Simulink**” to open Simulink. Create a blank Simulink® model.



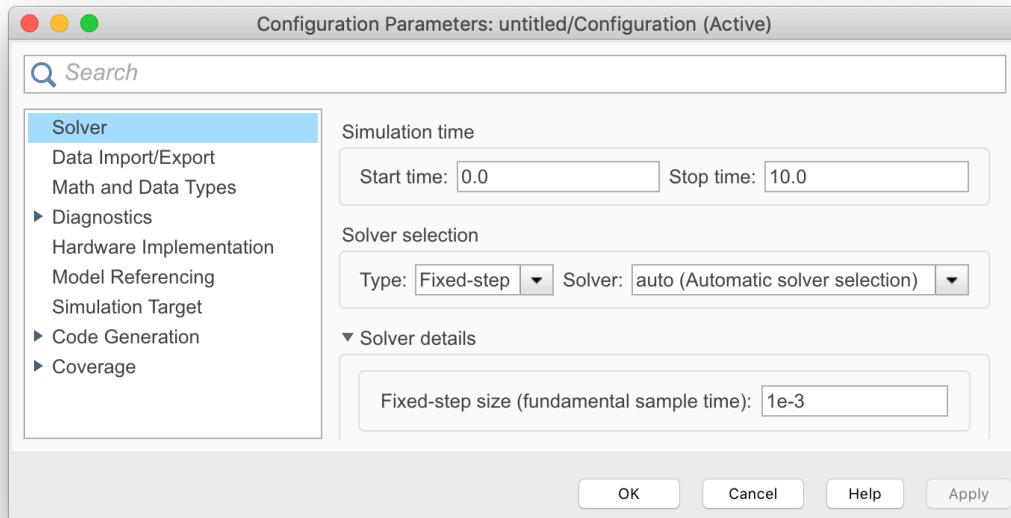
2. Use the keyboard shortcut **CMD+E** (or **CTRL+E** on Windows) to open the “**Configuration Parameters**” window. You can also get there from **Modeling > Model Settings > Model Settings** on the top toolbar, or by clicking on the blue text in the bottom right of the Simulink model and then clicking on the gear icon that shows up on the small box.



- Click “Hardware Implementation” then select NXP FRDM-K64F in the “Hardware board” drop-down menu. Make sure the firmware shown under “Target hardware resources” is “Segger’s J-Link OpenSDA V2 firmware”. Leave the other options as the default. If the firmware is not “Segger’s J-Link OpenSDA V2 firmware”, click “Change firmware options” and follow the instructions on the screen to change the firmware to “Segger’s J-Link OpenSDA V2 firmware”.

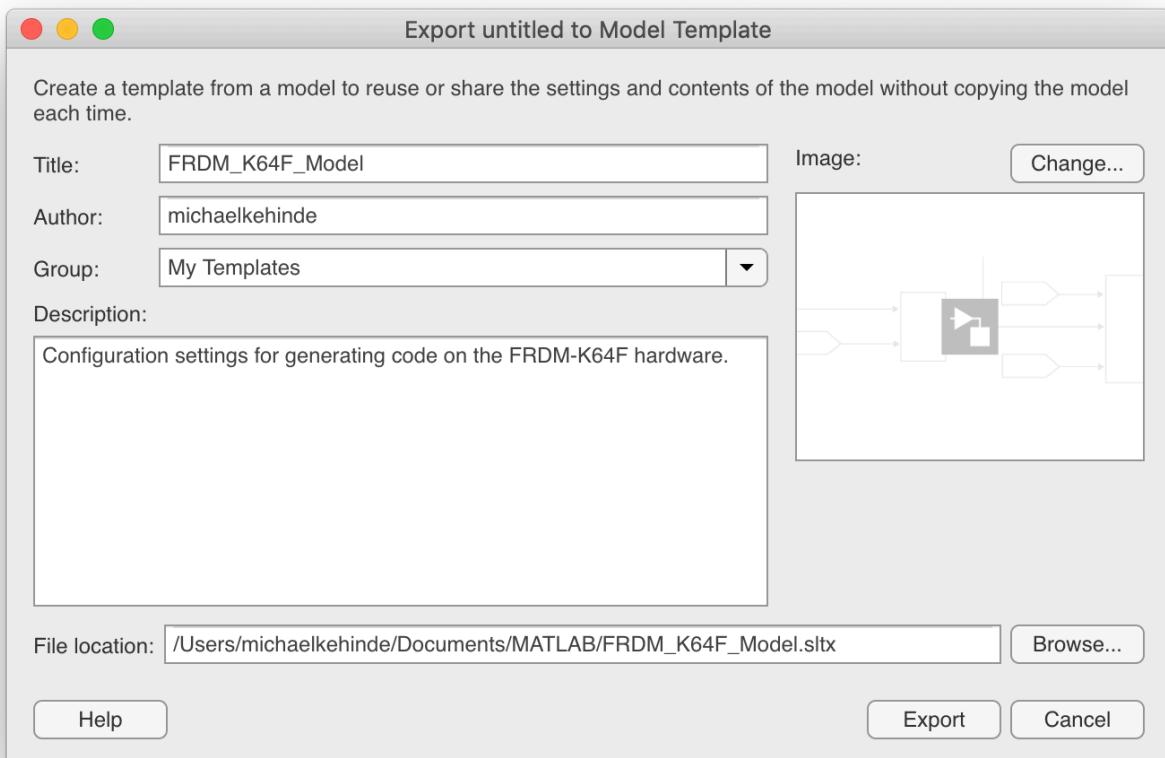


4. On the pane on the left hand side, click “Solver”. Under **Solver selection**, make sure the solver type is **“Fixed-step”** and leave the **Solver** on auto.



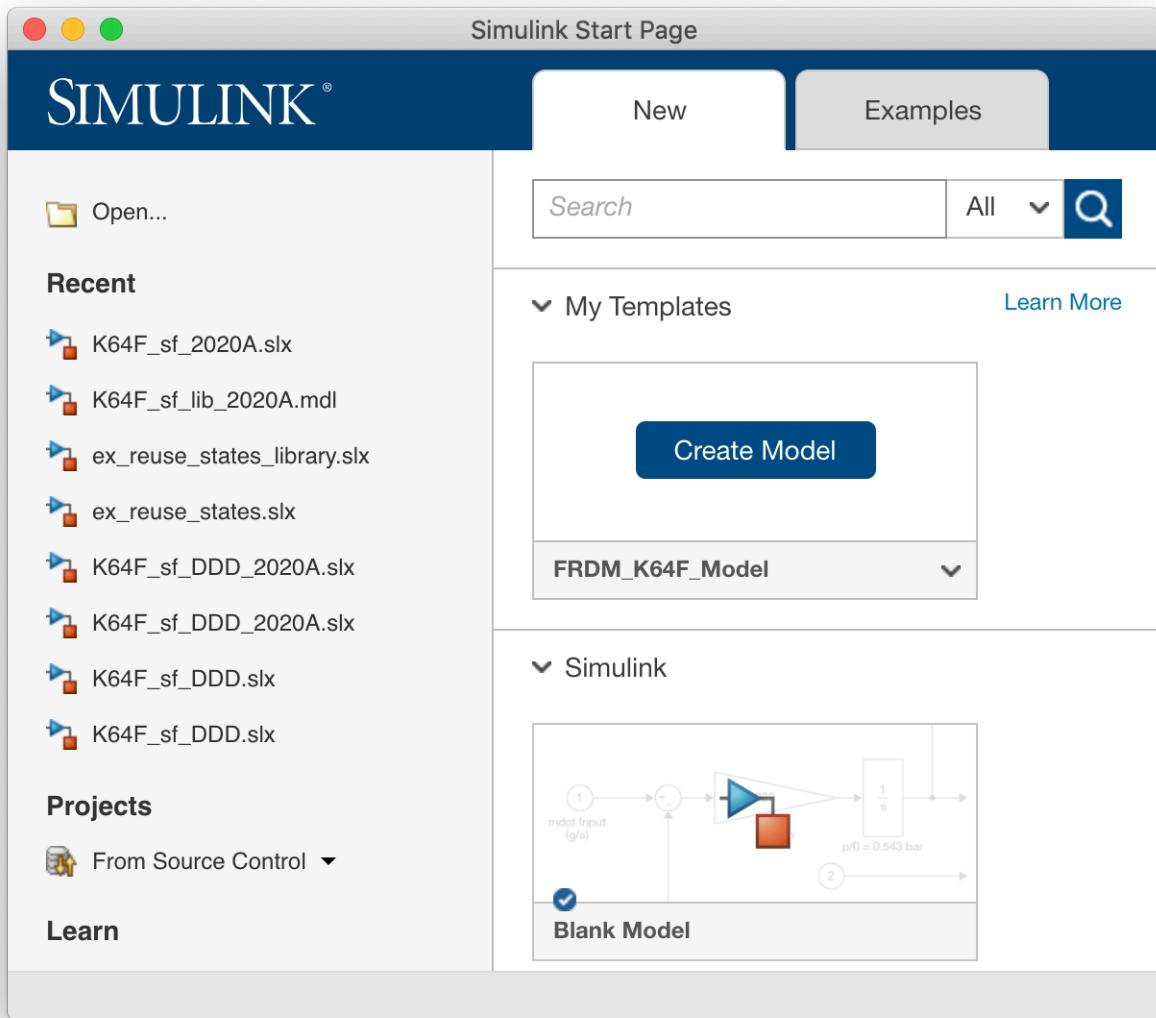
5. Set the Fixed-step size to 1e-3. **Fixed-step size (Fundamental Sample Time)** is a parameter used to specify the resolution of timing in the simulation typically less than or equal to the lowest tolerance in the model. For example, if you wanted to deliver a pulse every 1000 milliseconds with a tolerance of +/- 1 milliseconds you have to choose a fixed-step size \leq 1ms. You can change the step size by going to Solver options, expanding **Additional options** then changing the Fixed-step size to the desired step size in seconds.

6. Navigate to the **Save** drop-down menu under the Simulation ribbon.
Export the model to a template and use it when building subsequent models.



2.4 Part D — Exploring Simulink

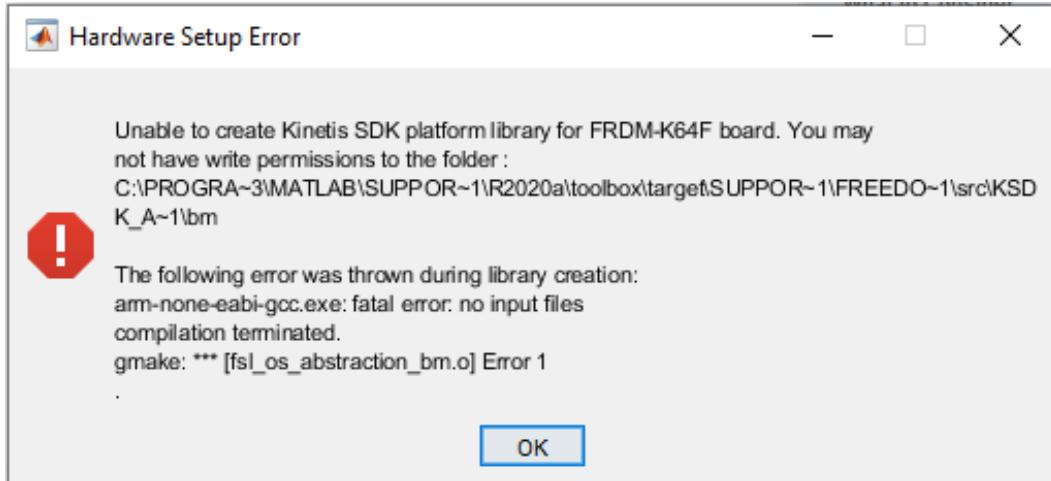
1. Navigate to the Simulink® pane on the Home ribbon. Open “Simulink”, then create a new model using the template you made in **Part C**.



2. Take some time to get acquainted with the Simulink UI. (You may be spending lots of time with Simulink this semester). Navigate through each ribbon and get a feel for what features are available. The next tutorial will involve building a basic model and flashing the K64F. If you would like some additional resources, you can begin exploring Getting Started with Simulink Coder Support Package for NXP FRDM-K64F.

3 TROUBLESHOOTING

3.1 Hardware Setup Error



This error is due to a MATLAB bug. Follow the steps in this discussion to resolve the error: <https://www.mathworks.com/matlabcentral/answers/515771-why-does-the-setup-stage-fail-for-simulink-coder-support-package-for-nxp-frdm-k64f-board>

If you are having trouble finding the makefiles, follow these steps:

1. Make sure that Kinetis SDK is installed in it's default location.

C:/Freescale/KSDK_1.2.0

2. Enter the following command on the MATLAB command line. Make sure to copy and paste all three lines together:

```
open([matlabshared.supportpkg.getSupportPackageRoot,
      '/toolbox/target/supportpackages/freedomk64f/src/',
      'ksdk_armgcc_lib/bm/ksdk_platform_lib_bm.mk'])
```

3. Press **Enter** on your keyboard.

4. Enter the following line at the beginning of the file that opens:

SHELL=C:/Windows/System32/cmd.exe

5. Find the line:

[KSDK_ROOT_DIR = \\$\(MW_KSDK_TOOLS_PATH\)](KSDK_ROOT_DIR = $(MW_KSDK_TOOLS_PATH)

And replace it with (The path should be the same as what is validated for Kinetis SDK):

`KSDK_ROOT_DIR = C:/Freescale/KSDK_1.2.0`

6. Save and close the file.
7. Enter the following command on the MATLAB command line. Make sure to copy and paste all three lines together:

```
open([matlabshared.supportpkg.getSupportPackageRoot,  
      '/toolbox/target/supportpackages/freedomk64f/src/',  
      'ksdk_armgcc_lib/freertos/ksdk_platform_lib_freertos.mk'])
```

8. Press **Enter** on your keyboard.
9. Enter the following line at the beginning of the file that opens:

SHELL=C:/Windows/System32/cmd.exe

10. Find the line:

`KSDK_ROOT_DIR = $(MW_KSDK_TOOLS_PATH)`

And replace it with (The path should be the same as what is validated for Kinetis SDK):

`KSDK_ROOT_DIR = C:/Freescale/KSDK_1.2.0`

11. Save and close the file.
12. Complete the setup process for the Simulink Code Support Package.
13. If none of this works - ask your classmates or TAs if anyone has made it past these steps.

Copy their files from (change path for your version):

`C:\ProgramData\MATLAB\SupportPackages\R2025a\toolbox\target\
supportpackages\freedomk64f\src\ksdk_armgcc_lib`

And replace those your files with theirs at that location

3.2 Missing Third Party Library Files

```
in file included from
/Users/[REDACTED]/Documents/MATLAB/SupportPackages/R2020a/toolbox/target/s
IIO.c:6:0:
/Users/[REDACTED]/Documents/MATLAB/SupportPackages/R2020a/toolbox/target/s
k_interface.h:9:10: fatal error: fsl_gpio_driver.h: No such file or directory
#include "fsl_gpio_driver.h"
^~~~~~
compilation terminated.
gmake: *** [MW_digitalIO.o] Error 1

Error(s) encountered while building "Model\_1":
### Failed to generate all binary outputs.

Component: Simulink | Category: Block diagram error
```

This error is due to an issue with the Kinetis SDK installation. Either the path you provided in the installer was invalid or the files got corrupted when you were downloading it.

1. Repeat the Hardware Setup and make sure the path you provided points to the KSDK_1.2.0 folder. Try to validate the library as in Part B — Step 7.
2. Otherwise try to reinstall the hardware support package.

3.3 Corrupted Model

In some rare instances, if your model refuses to build or shows a similar error to Section 3.3 even after reinstalling the support packages:

1. Make sure you have a valid file name with no spaces.
2. Copy the “autosave” file into new directory, rename to .slx and build the backup.
3. Otherwise, create a new project and copy the contents of the model over.

4 REVISION HISTORY

Version	Date	Modification	Modified by
1.0	Sept. 16, 2017	Initial R2017a FRDMK64F document	Ayesh, Mostafa
2.0	Sept. 13, 2020	Updated instructions; new template	Kehinde, Michael
2.1	Sept. 14, 2020	Description of MATLAB products	Kehinde, Michael
2.2	Sept. 20, 2020	Added Troubleshooting section	Kehinde, Michael
2.3	Sept. 24, 2020	Updated Troubleshooting section	Kehinde, Michael
2.4	Oct. 22, 2020	Improved accessibility of code snippets	Kehinde, Michael
3.0	Sept. 2, 2025	Updated links and some processes	Zavaleta, Angela
3.1	Sept 22, 2025	Trouble shooting improvements	Kapustin, Vasily