Developer's Guide To Pacemaker Development

Tutorial 1.2: Getting Started I

SFWRENG/MECHTRON 3K04
McMaster University

Michael Kehinde
kehindem@mcmaster.ca

October 23, 2020

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®?

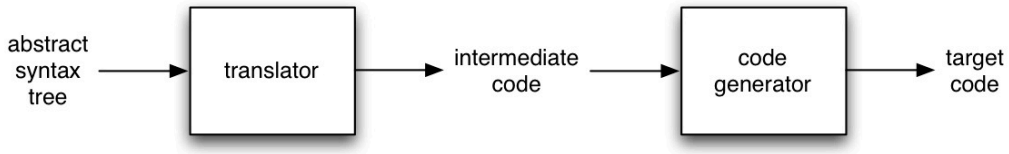
Simulink® is a graphical environment integrated with MATLAB® and a dataflow graphical programming language for simulation and model-based design.

1.2 Why are we interested in Simulink®?

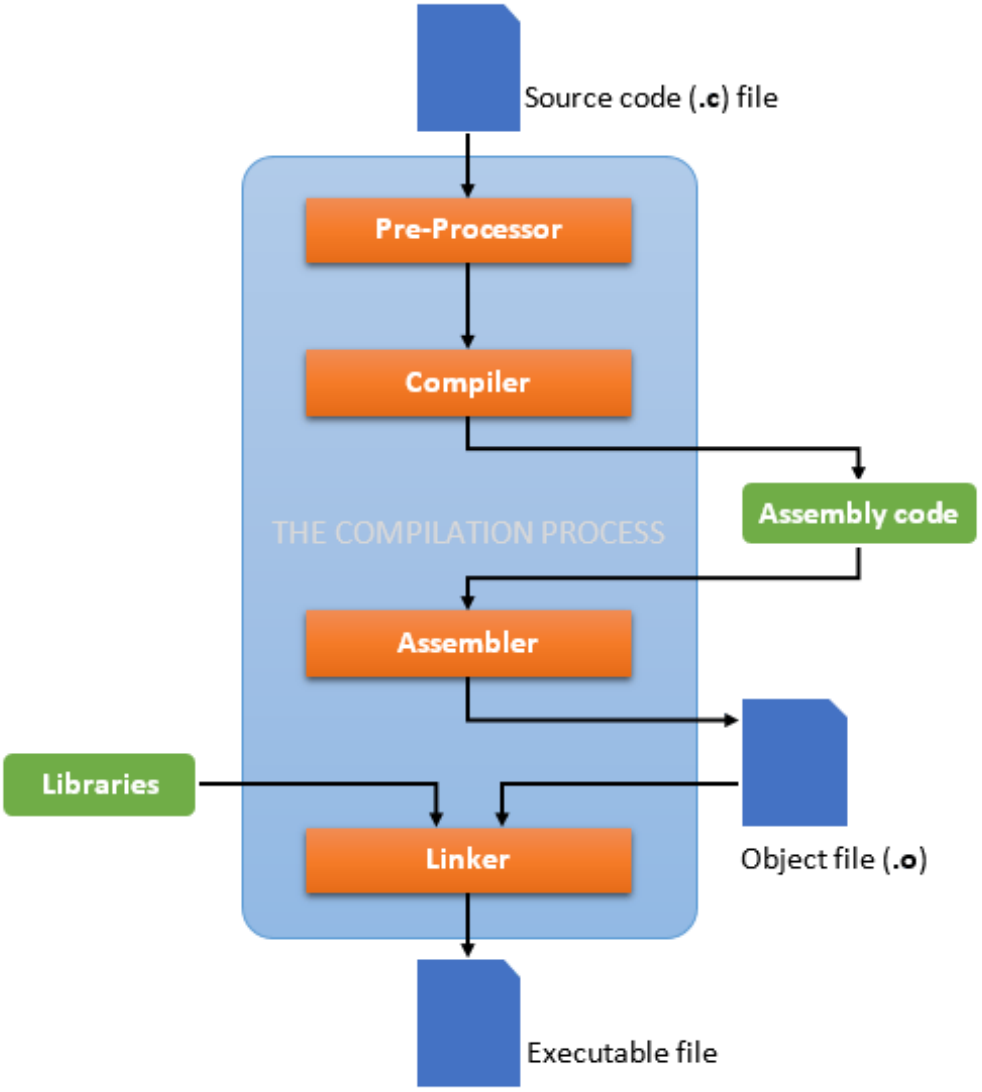
Simulink® is a **model-driven development (MDD)** platform and the leading environment for modelling, simulating and implementing embedded systems. Simulink® supports system-level design, simulation, continuous testing and verification of embedded systems and automatic **code generation**.

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.

1. Click the following link to access the McMaster University MATLAB Access page: `https://www.mathworks.com/academia/tah-portal/mcmaster-university-31501097.html`. On the webpage, click the "Sign in to get started" button.

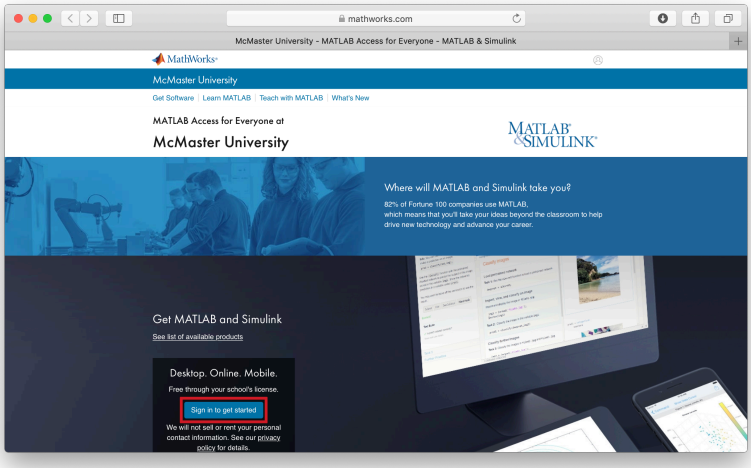
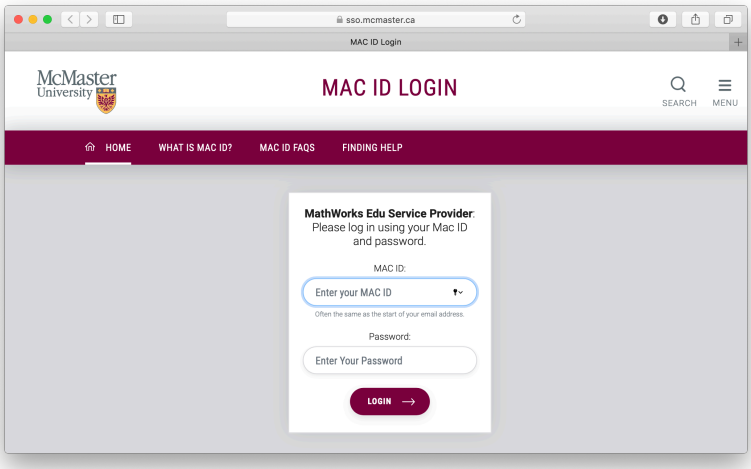
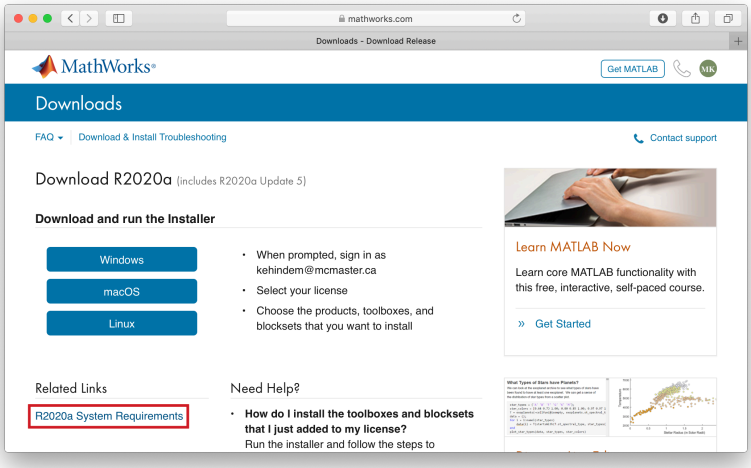


Figure 2: McMaster University MATLAB Access page.

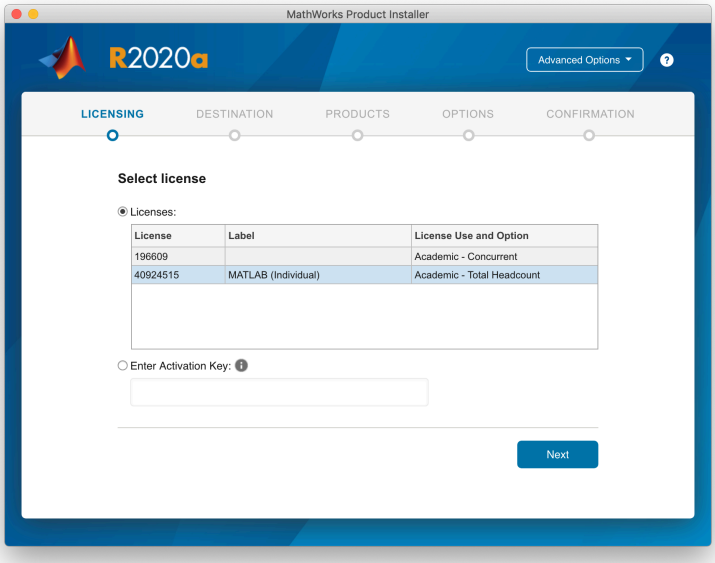
2. Follow the on-screen instructions to log in with your MACID, followed by your MathWorks® account.



3. Click the R2020a button to install the R2020a release. On the following page, click the link under the "Related Links" section on the bottom left corner of the screen to review the System Requirements. If your system is not supported for R2020a, notify your TA.



4. Download the installer for your operating system.
5. 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**.



6. 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™.

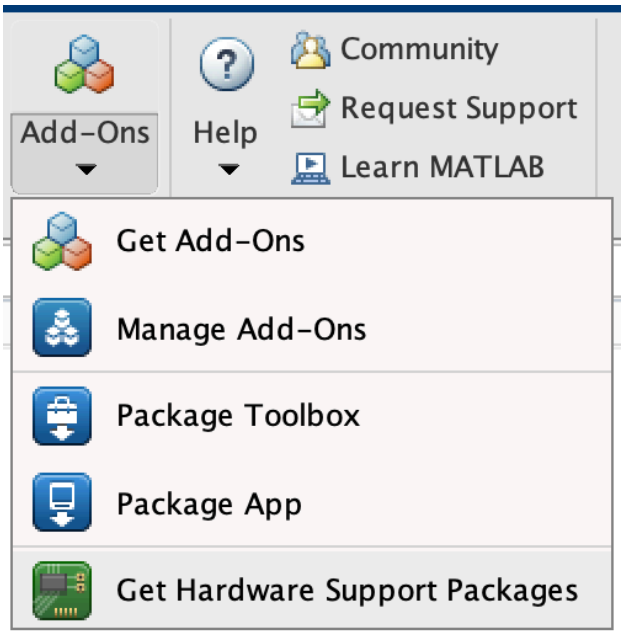
Note: if you missed a toolbox, you can install it later within MATLAB®. To find and install the toolbox, go to the **Home** ribbon in the top toolbar. On the **Environment** pane, click the **Add-Ons** ▽ dropdown menu and select **Package Toolbox**.

7. Confirm your options and begin the installation. Make sure you have 16GB of disk space available. (The total download size for MATLAB® and products will be less than 3.4GB and the installation size should be about 12GB).

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

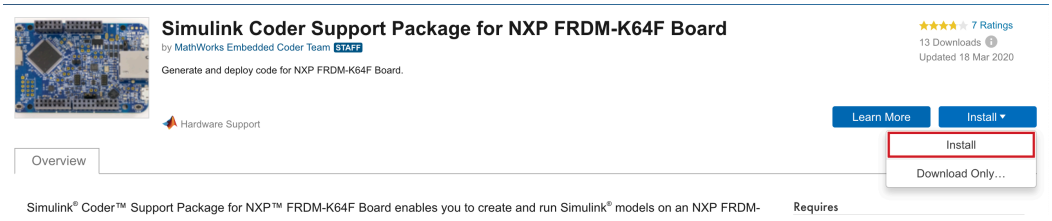
The support package is an add-on that enables us to use Simulink® with the FRDM-K64F hardware.

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**.

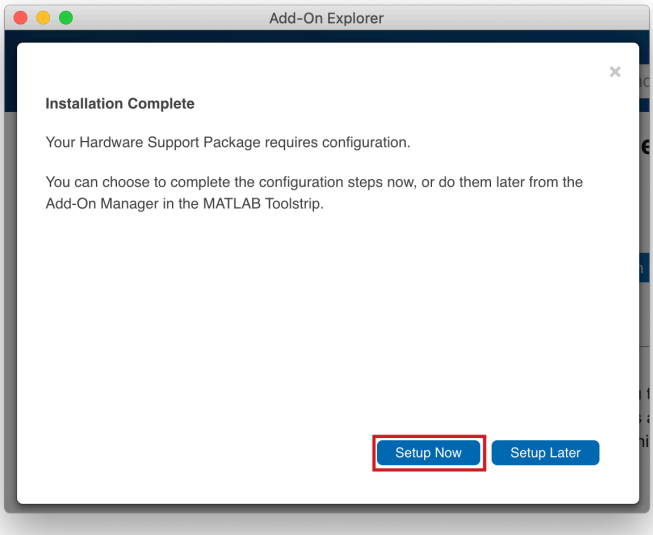


2. Search for the `Simulink® Coder Support Package for NXP FRDM-K64F`

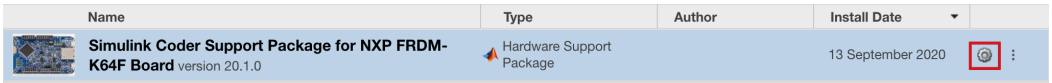
`Board` add-on using the search filter on the top right of the Add-On Explorer. Install the hardware support package.



3. Follow the on-screen instructions to complete the installation. When the installation is complete, click **"Setup Now"**.



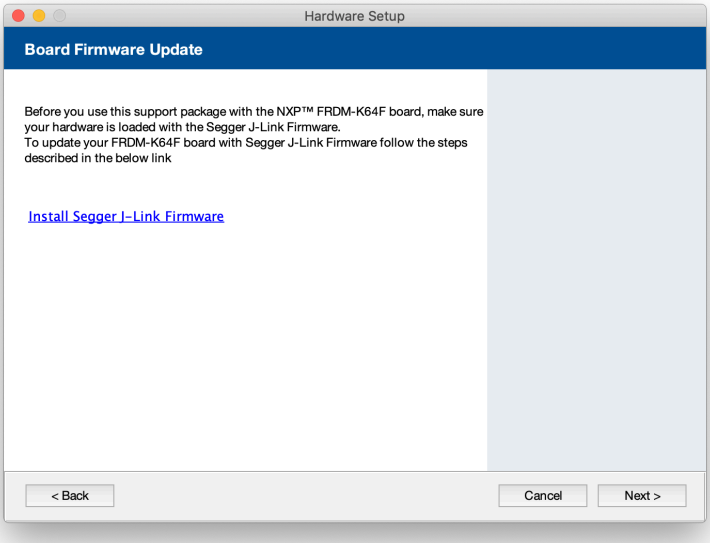
Note: if you will need to set up the support package later, you will be able to access the setup menu by navigating to **Home** > **Environment** > **Add-Ons** ▽ > **Manage Add-Ons**. Click the gear icon next to the FRDM-K64F board support package.



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).

4. 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.

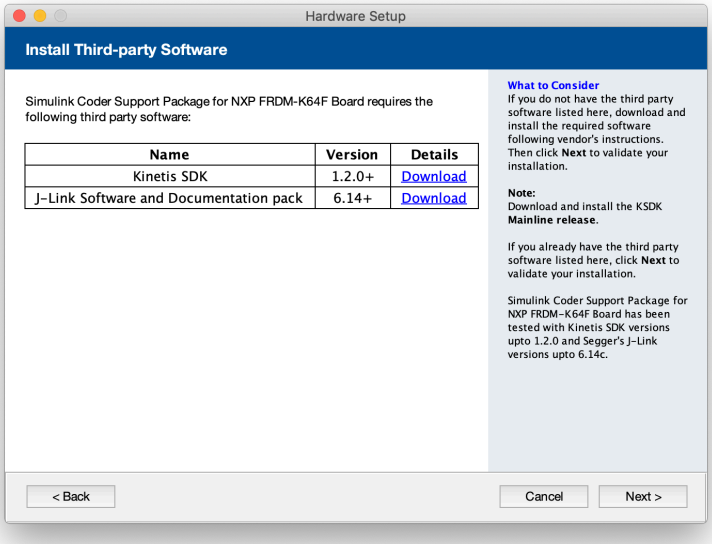
5. Proceed to the next page. Click "Next" to skip the Board Firmware Update. This step has already been completed for you.



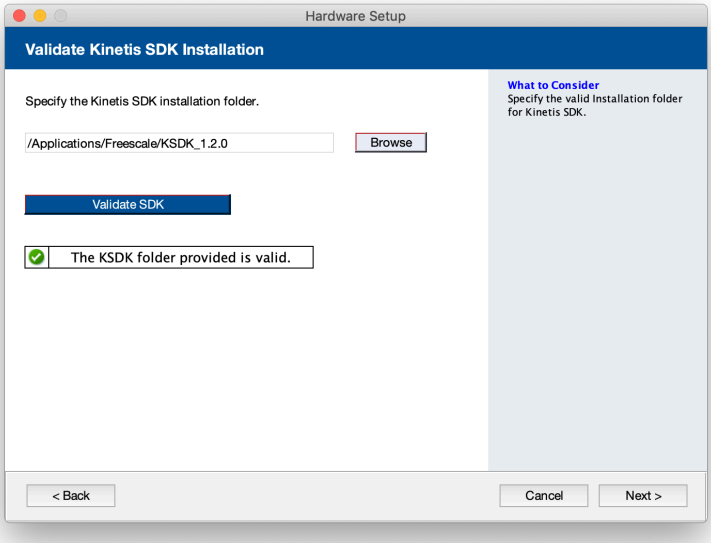
6. Follow the on-screen instructions to download the third-party software. **Note:** the third-party software makes up the libraries segment of the illustration in Figure 1. Make sure you install **Kinetis SDK 1.2.0 mainline release** and not the newer KSDK 1.3.0 to avoid any issues that may occur. Install 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.

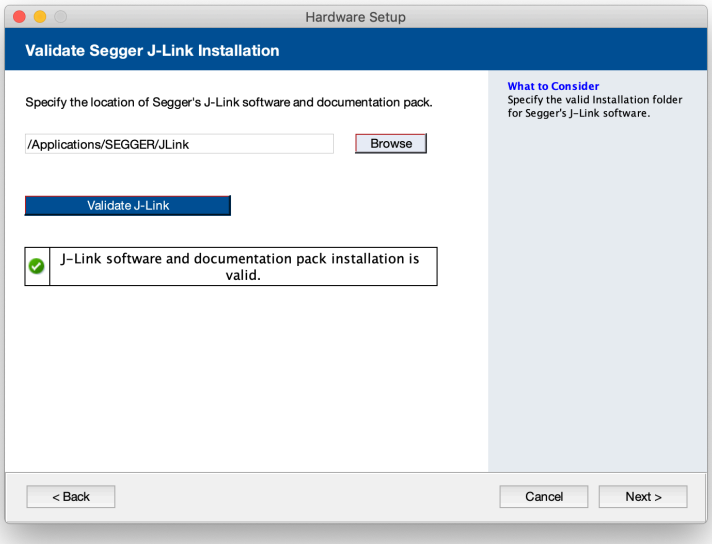
Note: you will need to set up an NXP account to download Kinetis SDK. For the company name, you may enter "McMaster University".



7. Validate the Kinetis SDK installation.



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



9. 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. Make sure to copy both lines together.

   ```
   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
   ```

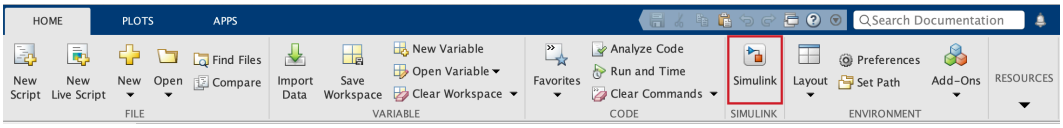
   Do not get any of the letters wrong. Double click the entry from the first text field above, copy it and use the CTRL + F keys to find the line in the file. Remove that line from the file. Then double click the entry in the second text field above, copy it and paste it at the same position in the file.

   iii. Save and close the file.

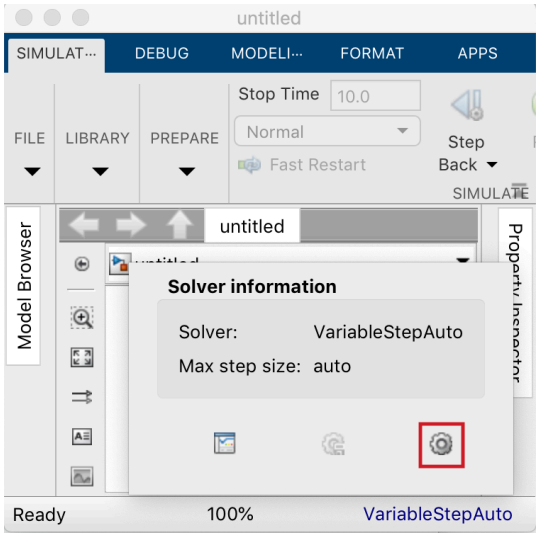
2.3 Part C — Simulink® Model Configuration

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 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.

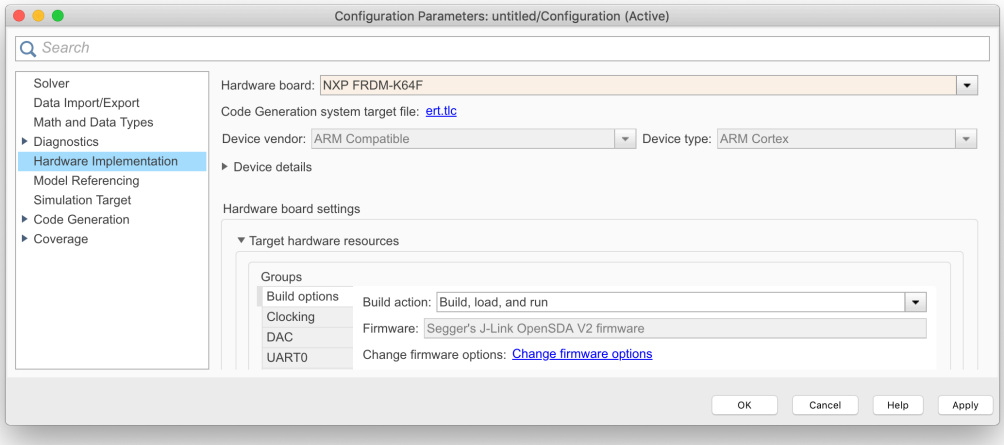


2. Use the keyboard shortcut CMD+E (or CTRL+E on Windows) to open the **"Configuration Parameters"** window. You can also get there from **Modeling** » **Model Settings** » **Model Settings** on the top toolbar, or by clicking on the blue text in the bottom right of the Simulink model and then clicking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

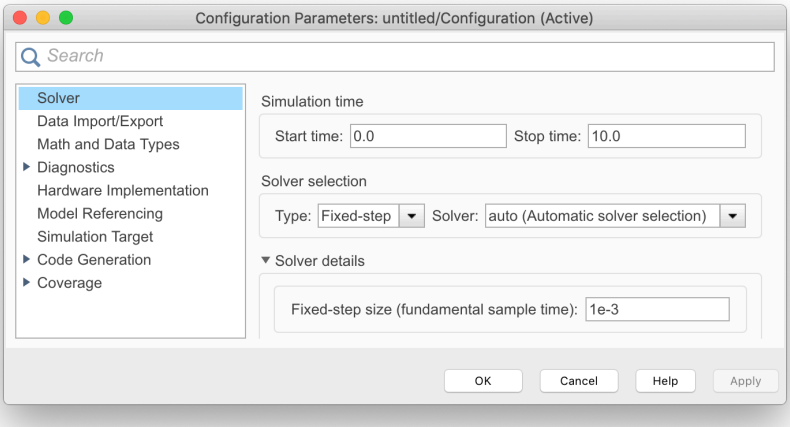


3. Click "Hardware Implementation" then select NXP FRDM-K64F in the **"Hardware board"** drop-down menu. Make sure the firmware

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"**.

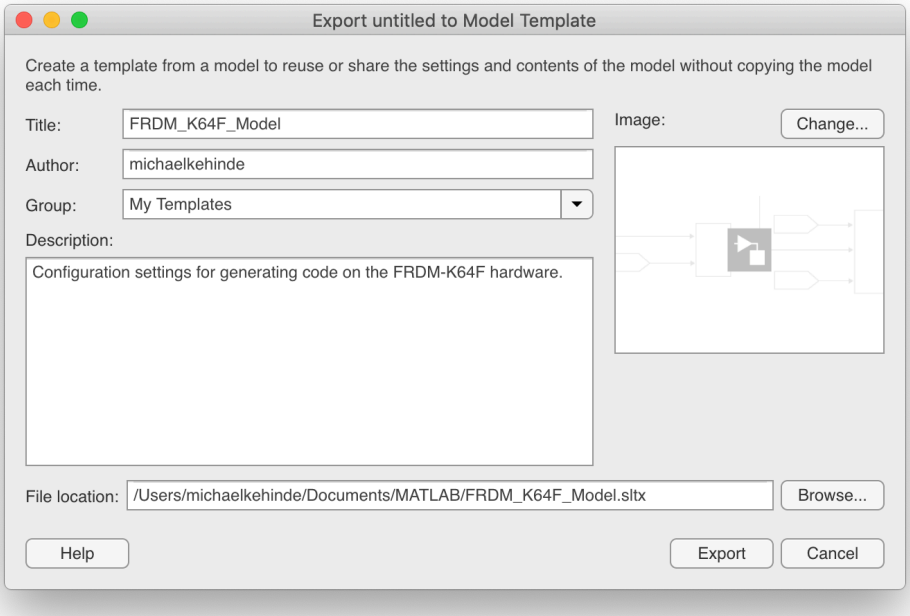


4. On the pane on the left hand side, click "Solver". Under **Solver selection**, make sure the solver **type** is **"Fixed-step"** and leave the **Solver** on auto.



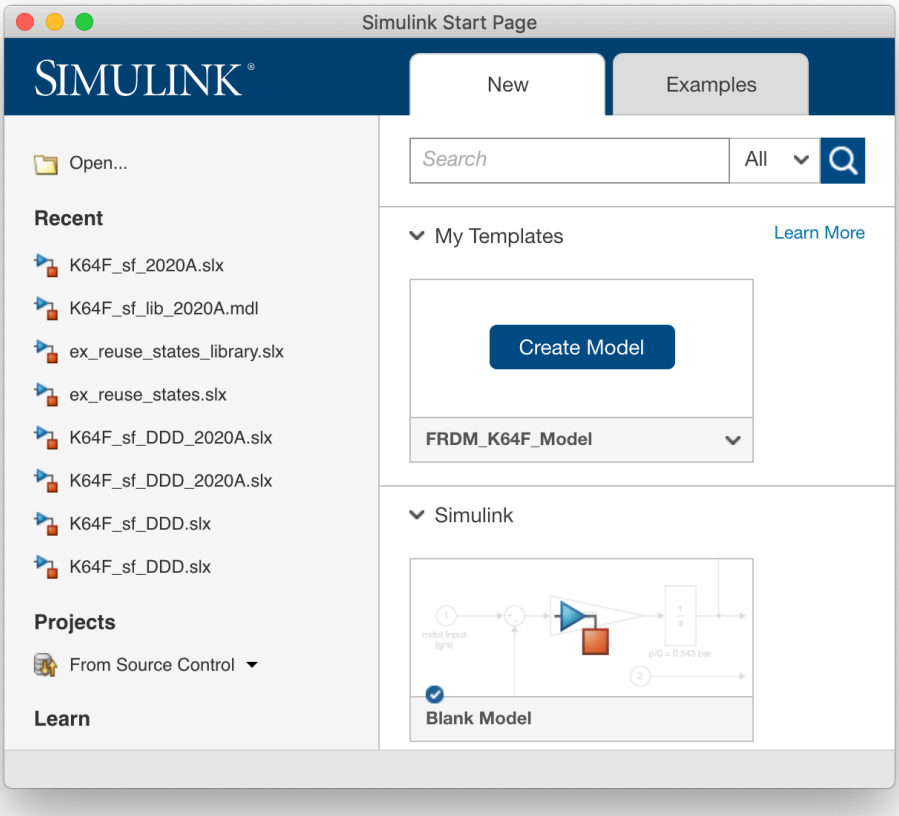
5. Set the Fixed-step size to 1e-3. **Fixed-step size (Fundamental Sample Time)** is a parameter used to specify the resolution of timing in the simulation typically less than or equal to the lowest tolerance in the model. For example, if you wanted to deliver a pulse every 1000 milliseconds with a tolerance of +- 1 milliseconds you have to choose a fixed-step size <= 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. Click **"Simulink"** to open Simulink. 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 Board.

3 Troubleshooting

3.1 Board Firmware Update

If you encounter a problem where the FRDM-K64F microcontroller refuses to flash or always enters the bootloader or maintenance mode (i.e. the board shows up as the "BOOTLOADER" when it is connected to your computer), you may need to reflash the board firmware. (Recall the discussion of **bootloader** from **Tutorial 1.1**).

To replace the firmware on the MCU with the Segger J-Link OpenSDA firmware, connect one end of your micro-USB cable to the **OpenSDA USB** input on the FRDM-K64F board and the other end to the USB input on your system. Follow instructions on the webpage **"Install Segger J-Link Firmware"** to update the FRDM-K64F board with the Segger J-Link firmware. If you are using a Mac system, refer to the note below.

After the firmware has been updated successfully, the OpenSDA LED by the reset button will be solid red. The LED will stay on until you flash a program to the board.

Note: Due to limitations of the native Mac file explorer, replacing the firmware to use it as a J-Link cannot be done on Mac systems using the drag and drop operation. If you are using a Mac system, you will need to flash the FRDM-K64F using the command line interface. Replace Step 4 and Step 5 in the Segger J-Link Firmware update instructions with following steps:

1. Open a **Terminal** window (you can use Spotlight Search to find Terminal).
2. Double click the the following command to select it. Then copy and paste the command into a text editor of your choice.

   ```
   sudo mount -u -w -o sync /Volumes/BOOTLOADER; cp -X <path to interface firm
   ```

3. Replace the following placeholder *(angle brackets inclusive)* from the command above

   <path to interface firmware file>

   with the **full path** to the *02_OpenSDA_FRDM-K64F.bin* binary on your system. Copy and paste the modified command into the

Terminal window.

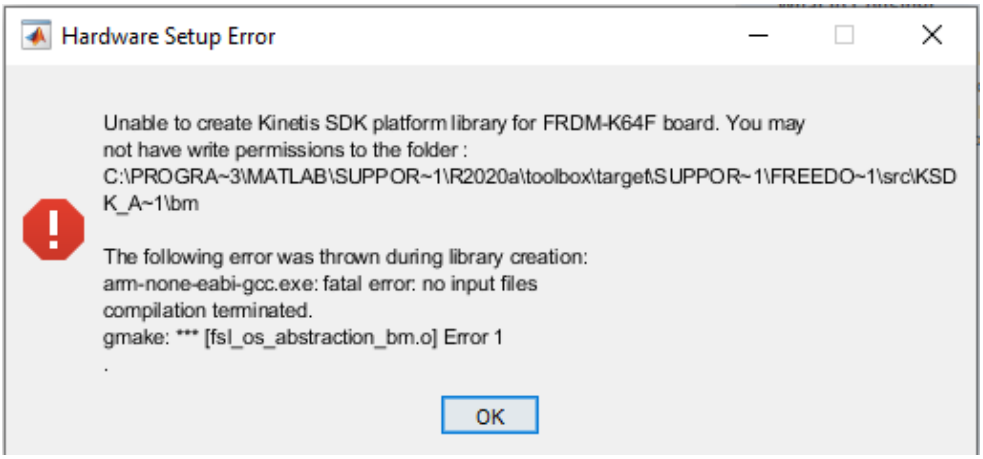
For example, if your binary is on your desktop, then your command would be

```
sudo mount -u -w -o sync /Volumes/BOOTLOADER; cp -X
    ~/Desktop/02_OpenSDA_FRDM-K64F.bin
    /Volumes/BOOTLOADER/
```

Note If you are having trouble with determining the path, you can move the binary to your desktop and copy and paste the command here in Step 3 verbatim. Alternatively, you can simply copy and paste first part of the command `sudo mount -u -w -o sync /Volumes/BOOTLOADER; cp -X` into the Terminal window. Then *click and drag the binary file to the Terminal window.* Then copy and paste the `/Volumes/BOOTLOADER/` part of the command.

4. Press Enter to run the command.

3.2 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. 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'])
   ```

2. Press Enter on your keyboard.

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

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

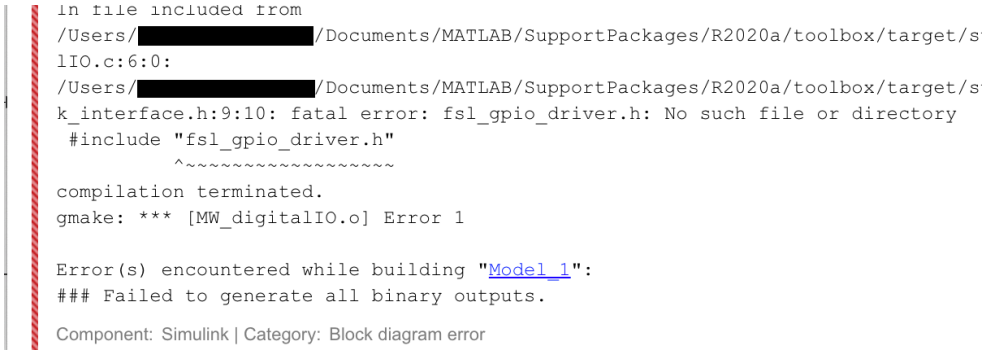
4. Save and close the file.

5. 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'])
   ```

6. Press Enter on your keyboard.
7. Enter the following line at the beginning of the file that opens: `SHELL=C:/Windows/System32/cmd.exe`
8. Save and close the file.
9. Complete the setup process for the Simulink Code Support Package.

3.3 Missing Third Party Library Files



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.

(a) 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.

(b) Otherwise try to reinstall the hardware support package.

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