Model-Based Design Toolbox RADAR

Quick Start Guide

Automatic Code Generation for Radar processors
Version 1.0.0

Target Based Automatic Code Generation Tools
For MATLAB™ models working with MATLAB Coder ™ and Embedded Coder®

Summary

1 Installation

Installing the Model-Based Design Toolbox is the first step in setting up and running automatic C code generation from MATLAB for NXP's embedded target processors and development boards.
The Model-Based Design Toolbox for RADAR supports code generation for:

- ARM Cortex-A53 core
- NXP SPT accelerator
- NXP LAX accelerator

1.1 System Requirements

For a flawless development experience, the minimum recommended PC platform is:

- *Windows® 10 OS*
- At least 4 GB of RAM
- At least 6 GB of free disk space.
- Internet connectivity for web downloads.

1.2 Installation Steps

NXP's Model-Based Design Toolbox is delivered as MATLAB Toolbox Package that can be installed offline or online from MathWorks Add-ons. This document shows how to install the offline/online package.

For the offline package, to have the toolbox installed and configured properly the following actions should be executed, assuming you have already downloaded the file:

1. Run the MATLAB toolbox package file `*.mltbx`.
2. Configure the external tools dependencies, such as the S32 Design Studio and Radar SDK.
3. Set up the MATLAB path for the Model-Based Design Toolbox and generate the right toolchain setting for the user MATLAB environment.

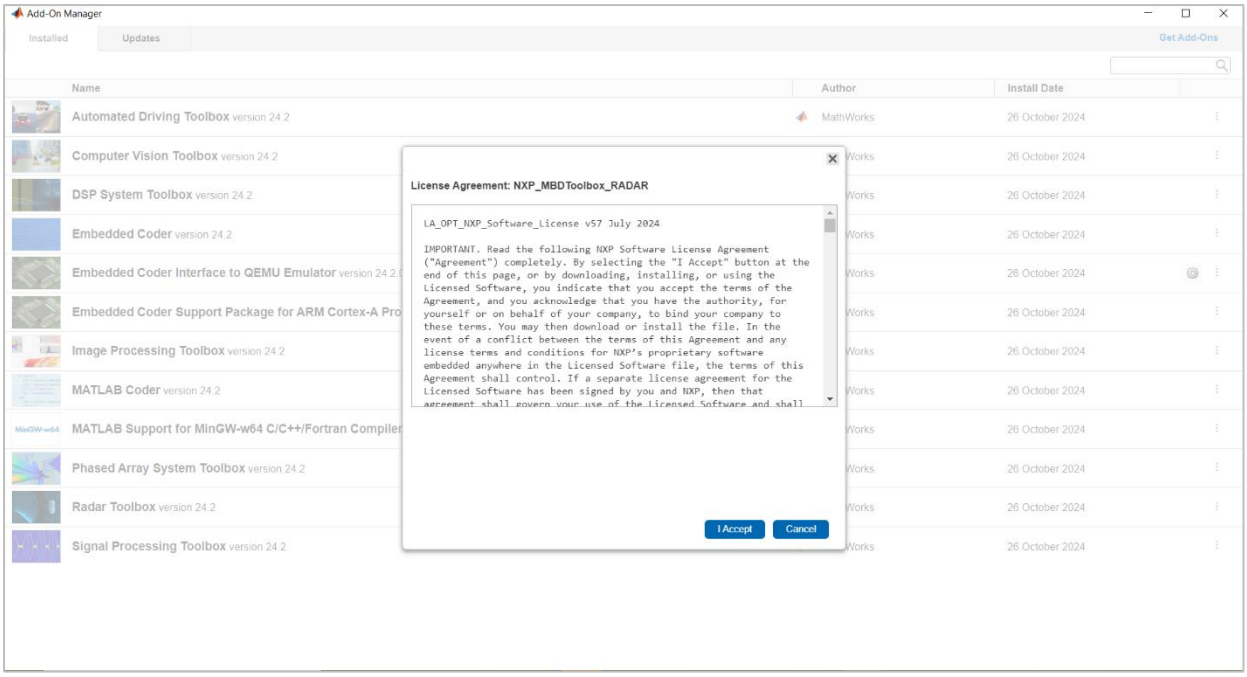
Each of these actions is explained in the following sub-chapters.

1.2.1 **Run Add-on installer (offline)**

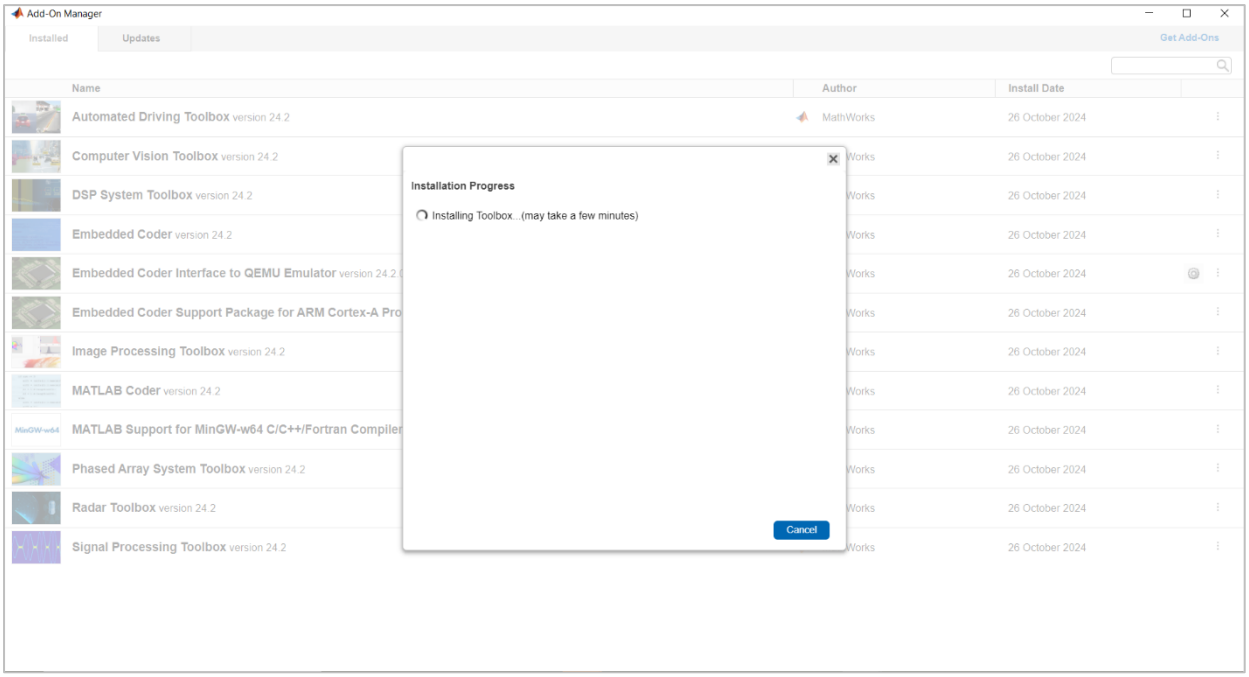
Install the NXP's Model-Based Design Toolbox by double-clicking the `*.mltbx` file. This will activate the MATLAB Add-ons installer that will automatically start the installation process.

After the MATLAB opens, you will be prompted with the following options:

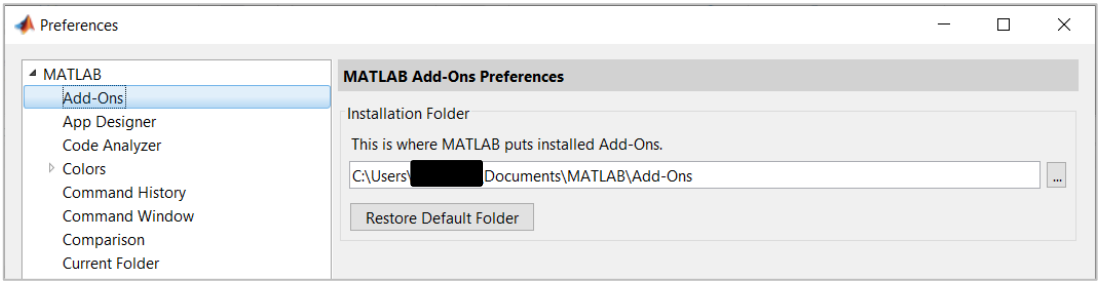
1. Indicate acceptance of the NXP Software License Agreement by selecting "`I Accept`" to proceed.



2. The rest of the process is silent and under MATLAB control. All the files will be automatically copied into the default Add-Ons folder within MATLAB.

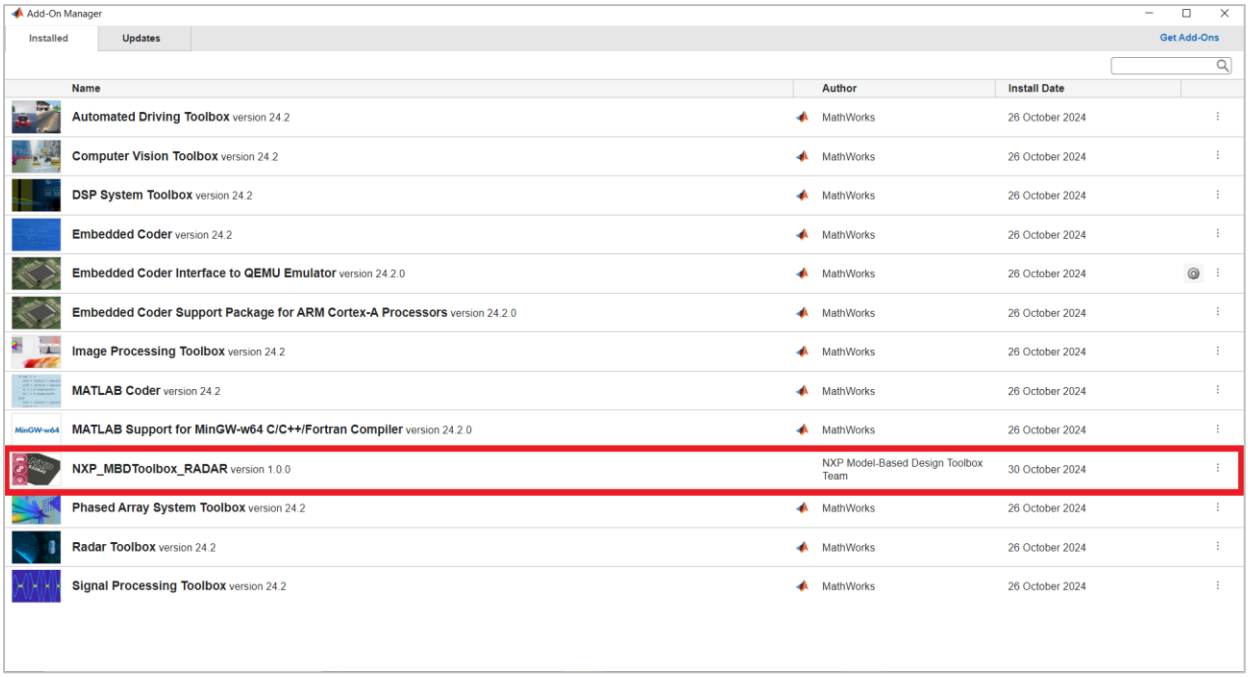


The default Add-Ons Installation Folder can be changed from MATLAB Preferences.

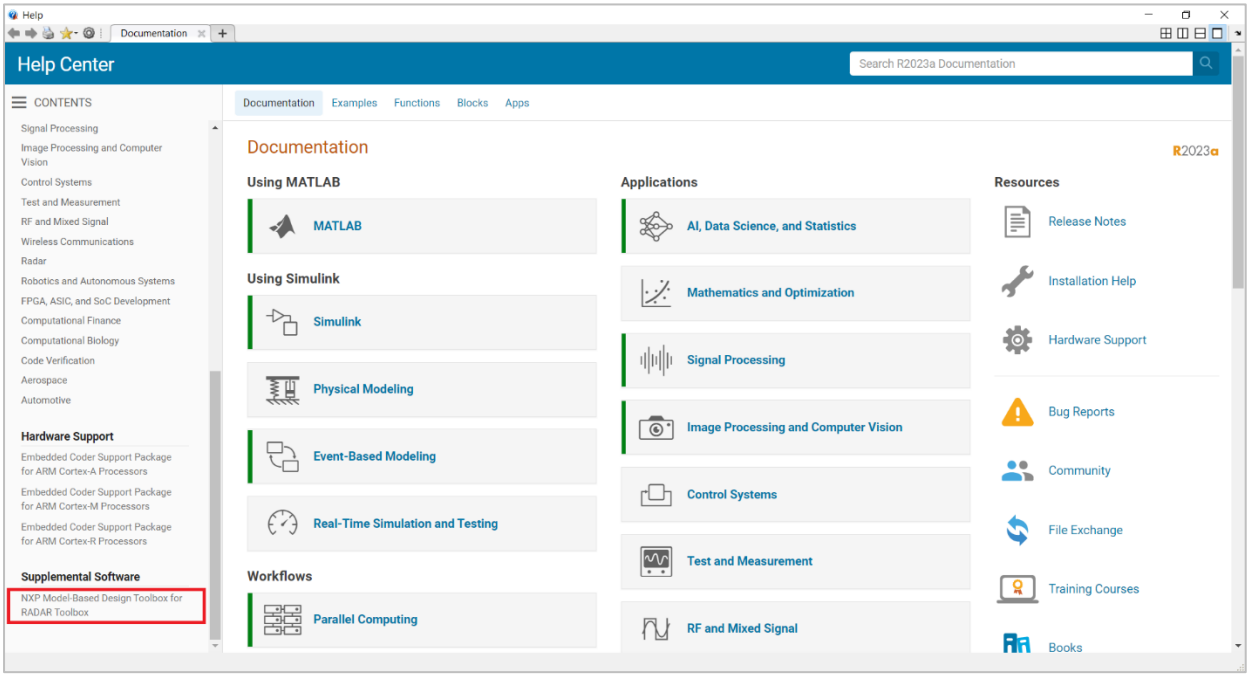


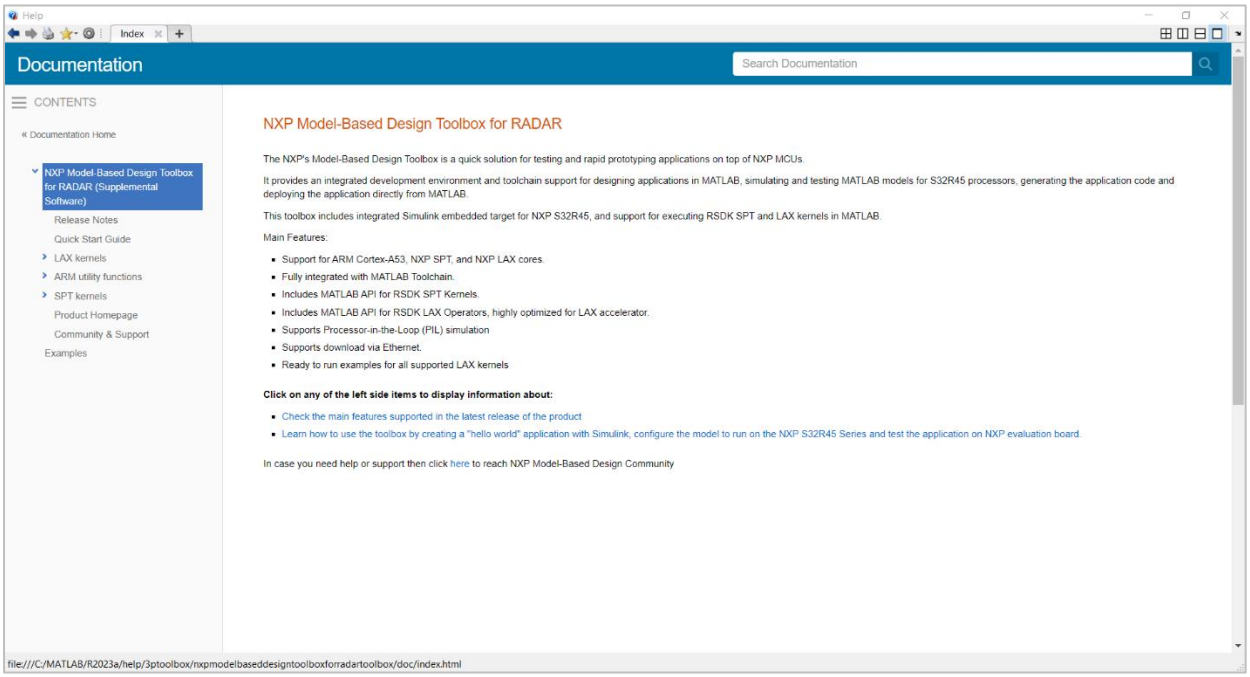
Note: *It is recommended to install the MATLAB and NXP Toolbox in a location that does not contain special characters, empty spaces, or mapped drives.*

3. After several minutes, the NXP's Model-Based Design Toolbox should be visible as a new Add-on.



4. NXP's Model-Based Design Toolbox documentation, help, and examples are fully integrated with the MATLAB development environment. Get more details by accessing the standard Help and **Supplemental Software** section.

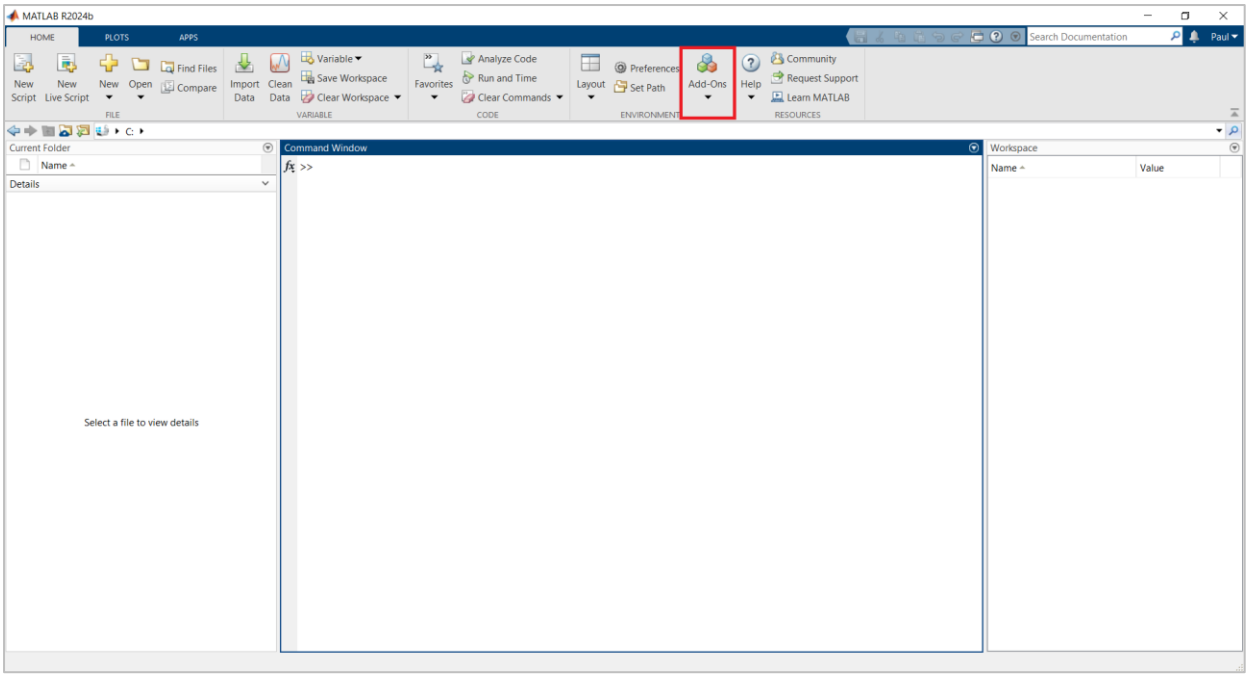




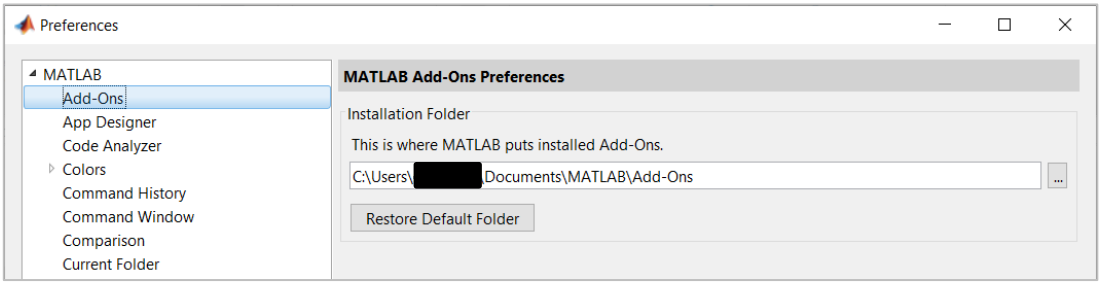
1.2.2 **Install NXP Support Package for RADAR (online)**

This package will guide you through the download, installation, and activation process of the MBDT for RADAR online package.

1. Go to MATLAB Add-Ons



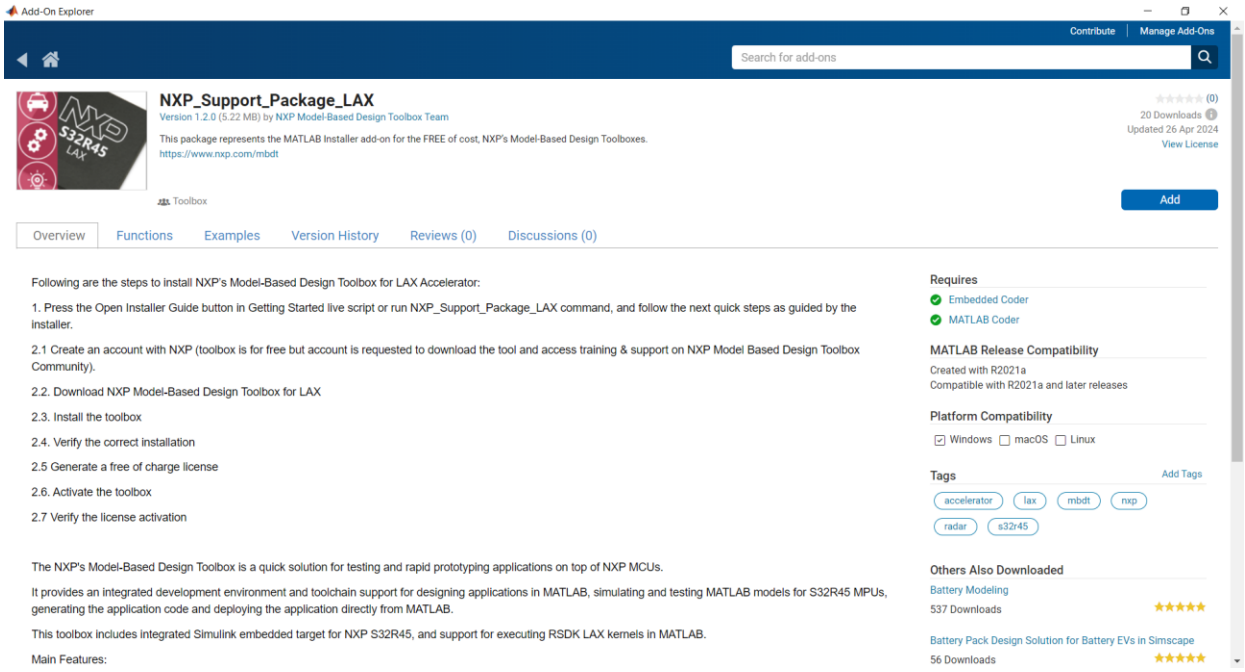
The default Add-Ons Installation Folder can be changed from MATLAB Preferences.



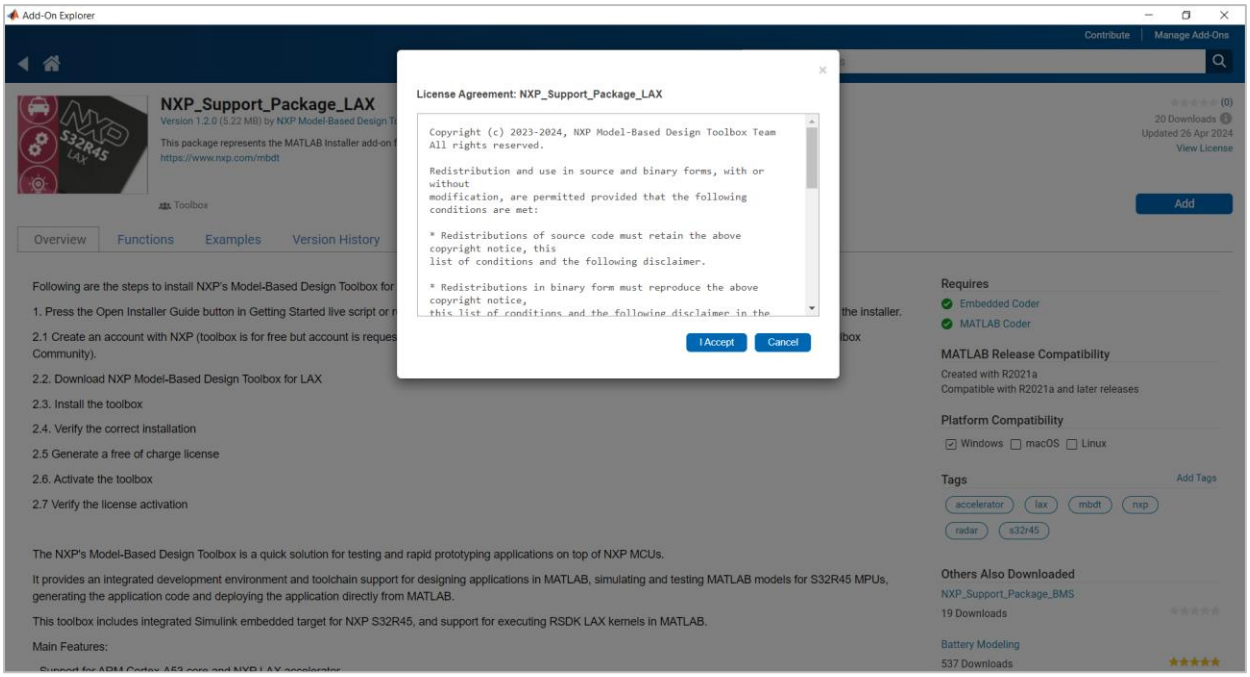
Note: It is recommended to install the MATLAB and NXP Toolbox in a location that does not contain special characters, empty spaces, or mapped drives.

2. Search for the “NXP Support Package RADAR”

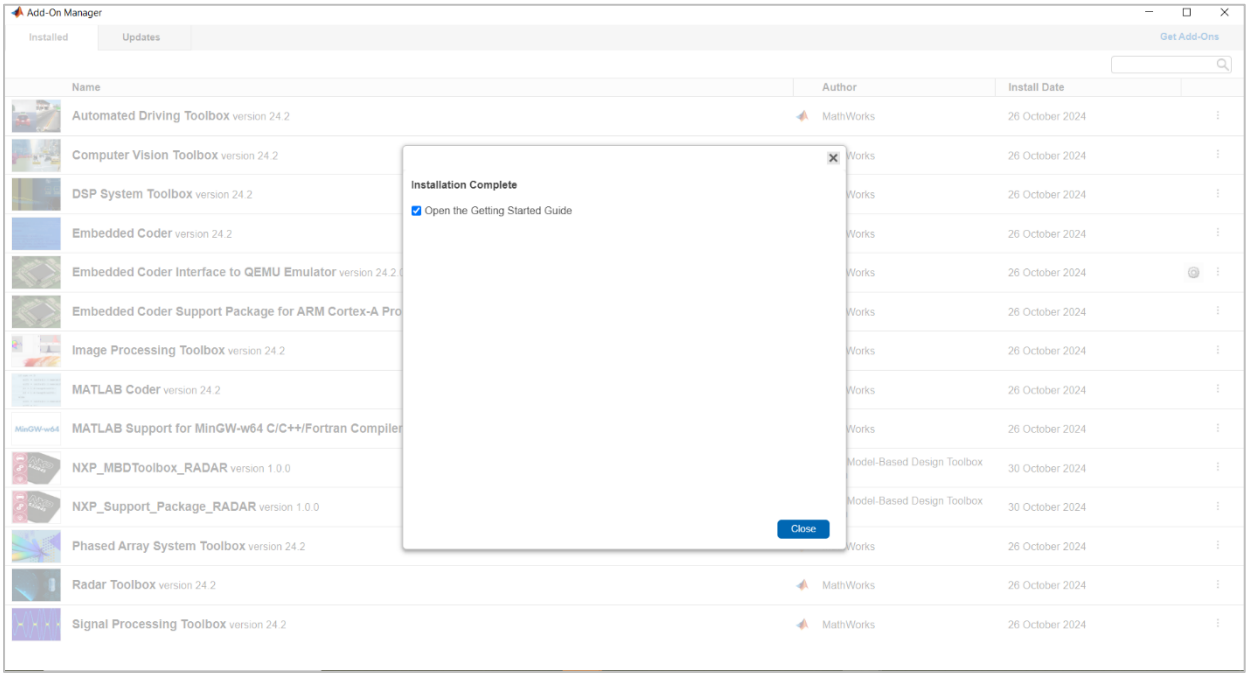
3. Install the “NXP Support Package RADAR” by pressing the **Add** button.



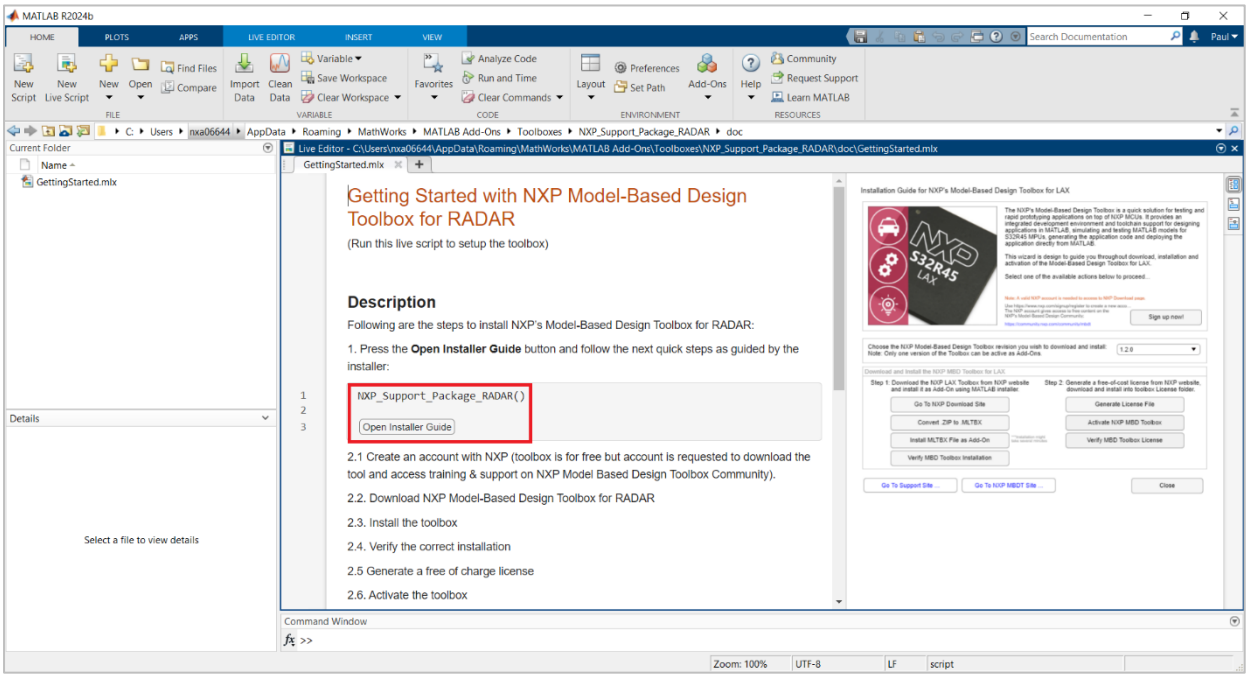
4. Read the License Agreement and press **I Accept.**



5. Once the process is successful, press the **Close** button.



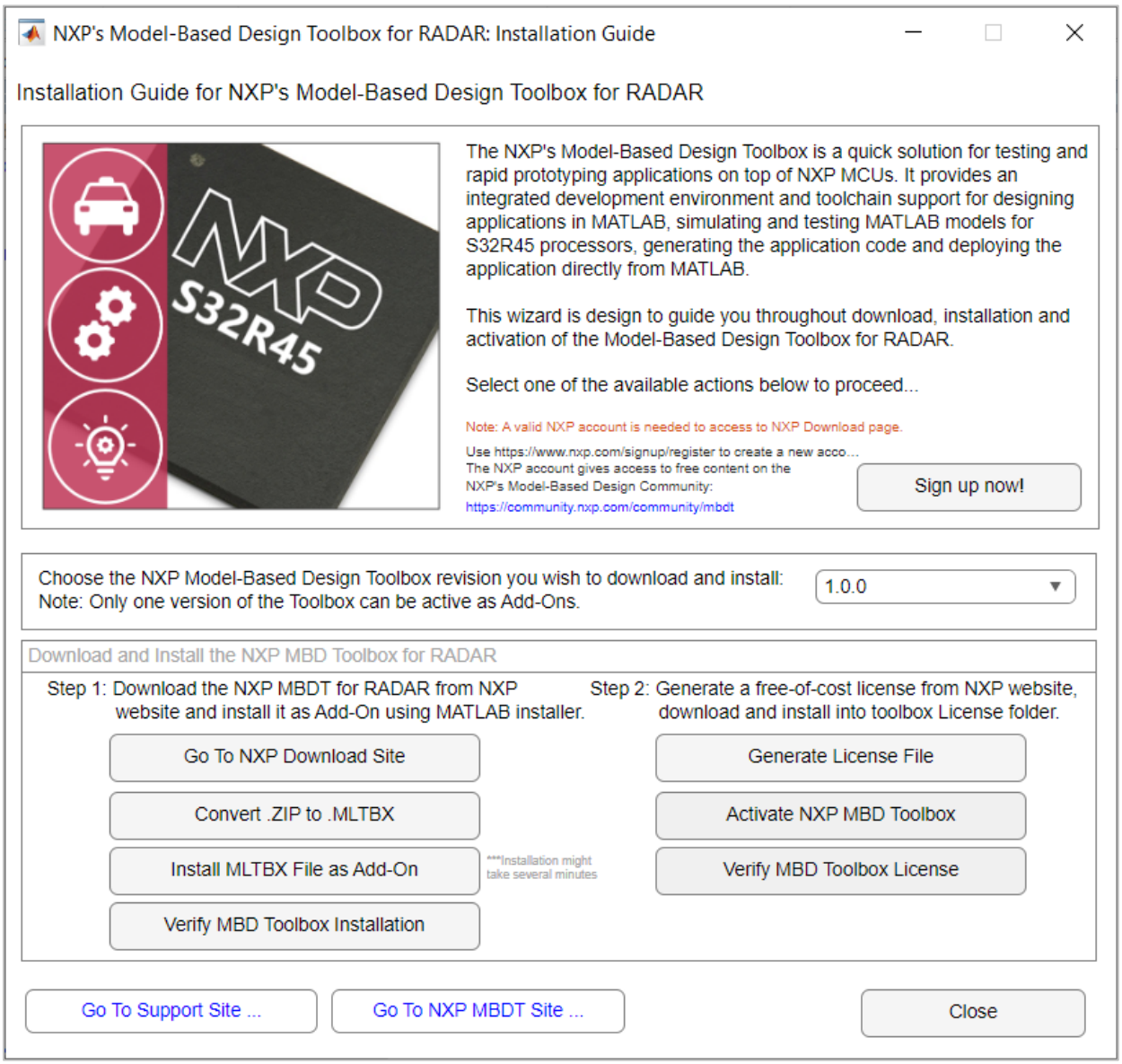
6. Press the **Open Installer Guide** button on the newly opened **Getting Started** page.



7. The **Toolbox Installation Guide** will be opened.

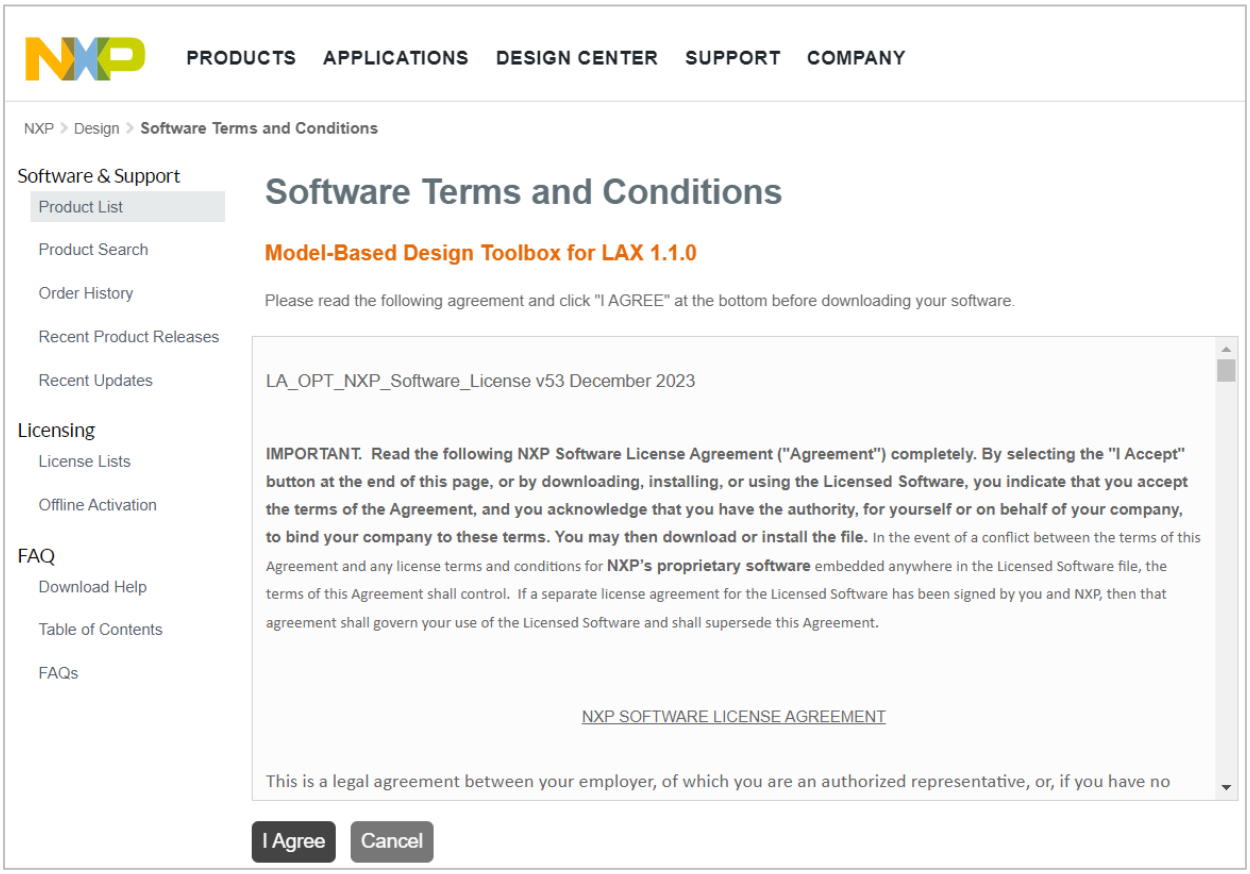
1.2.3 **Install NXP Model-Based Design Toolbox for RADAR**

NXP Toolbox Installation Guide is a graphical user interface guide that helps to download and install the Model-Based Design Toolbox and generate and install the license from the NXP website.



1. Press **Go to NXP Download Site** Button. In the newly opened window, Review the Terms and Conditions as you scroll down, and press the **I Agree** Button.

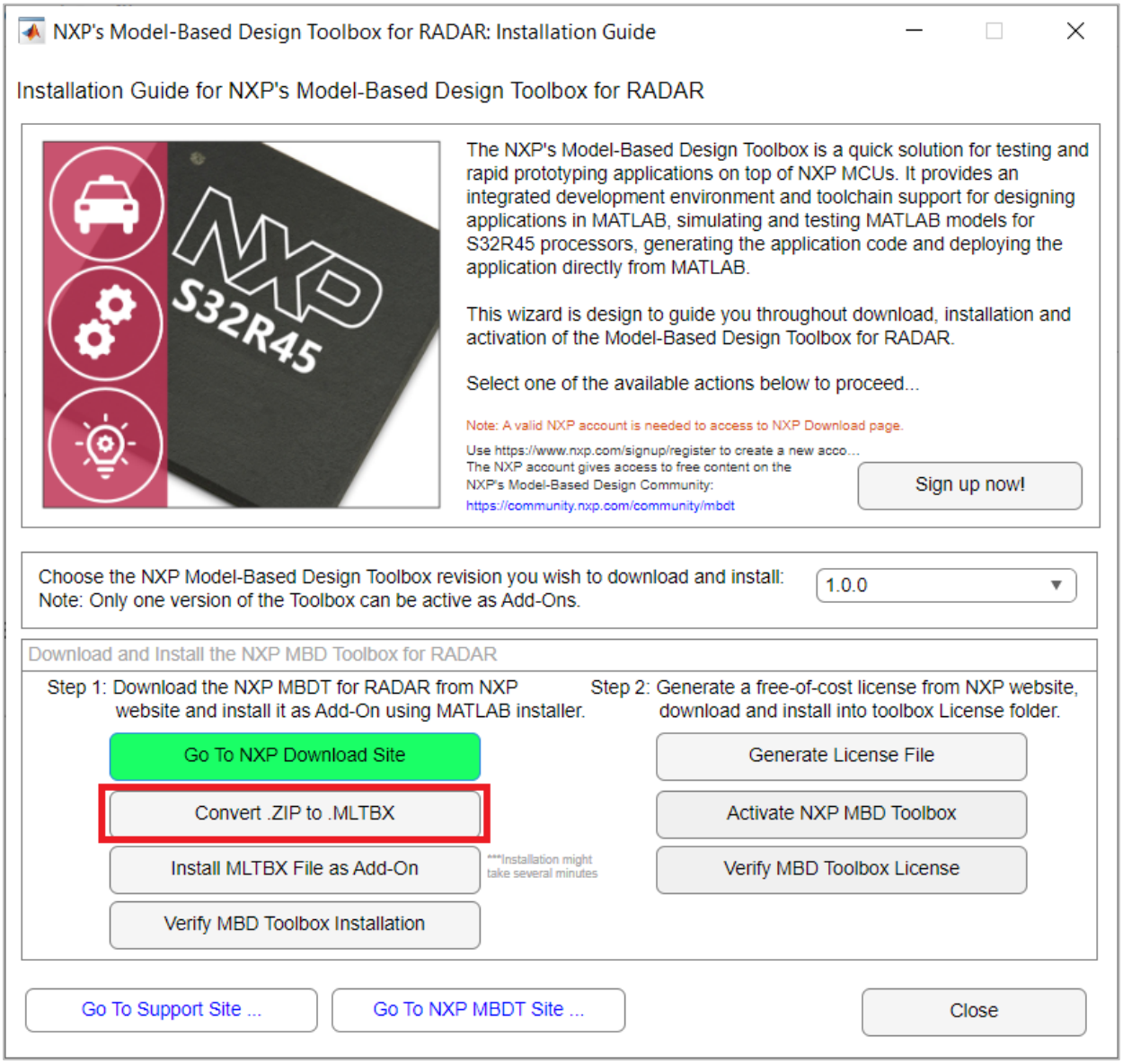
   Note: If the page is not displayed as below, please go to the location presented in chapter 1.2.4 License activation, section 4, Select **Automotive SW – S32R45 Standard Software** -> **Model-Based Design Toolbox for RADAR,** and select the latest release available.



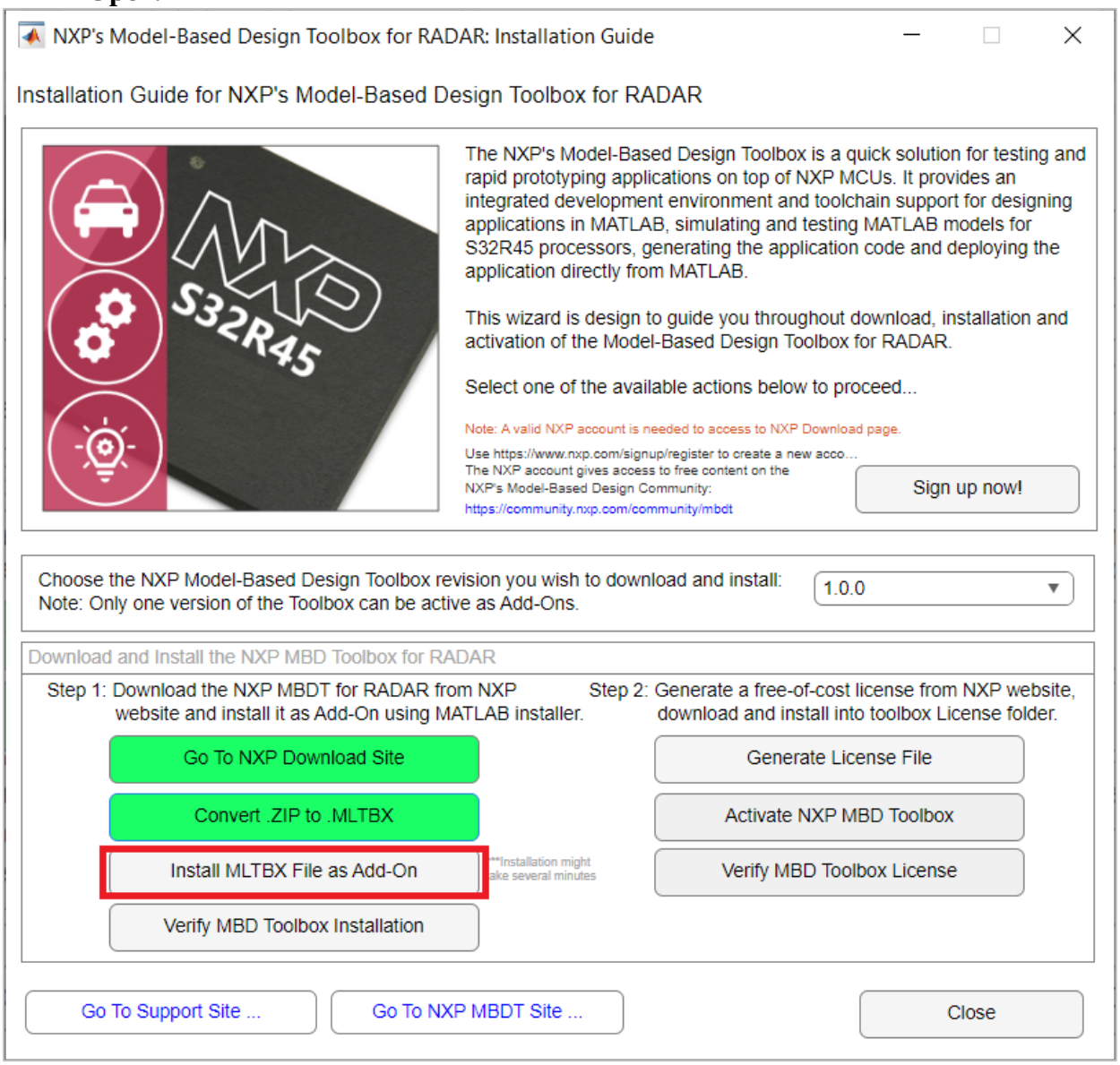
2. Download the **SW32_MBDT_RADAR_1.0.0_D2410.mltbx** file.

   Note: The downloaded file has the **.zip** extension instead of **.mltbx**. The next step helps to convert to the right format.

3. Go back to **NXP Toolbox Installation Guide** and press the **Convert .ZIP to. MLTBX** button. In the newly opened Browsing window, select the file downloaded and press **Open.**

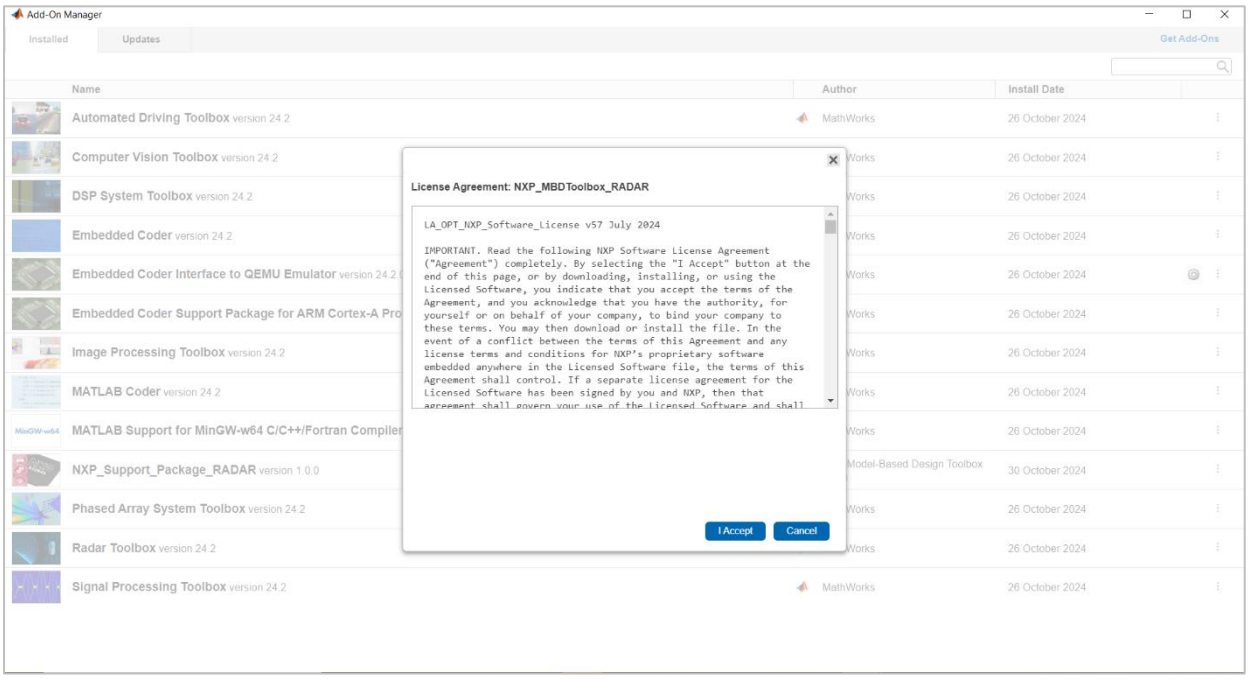


4. Go back to **NXP Toolbox Installation Guide** and select the **Install MLTBX File as Add-On** button. In the newly opened window, browse for the MLTBX file and press **Open.**

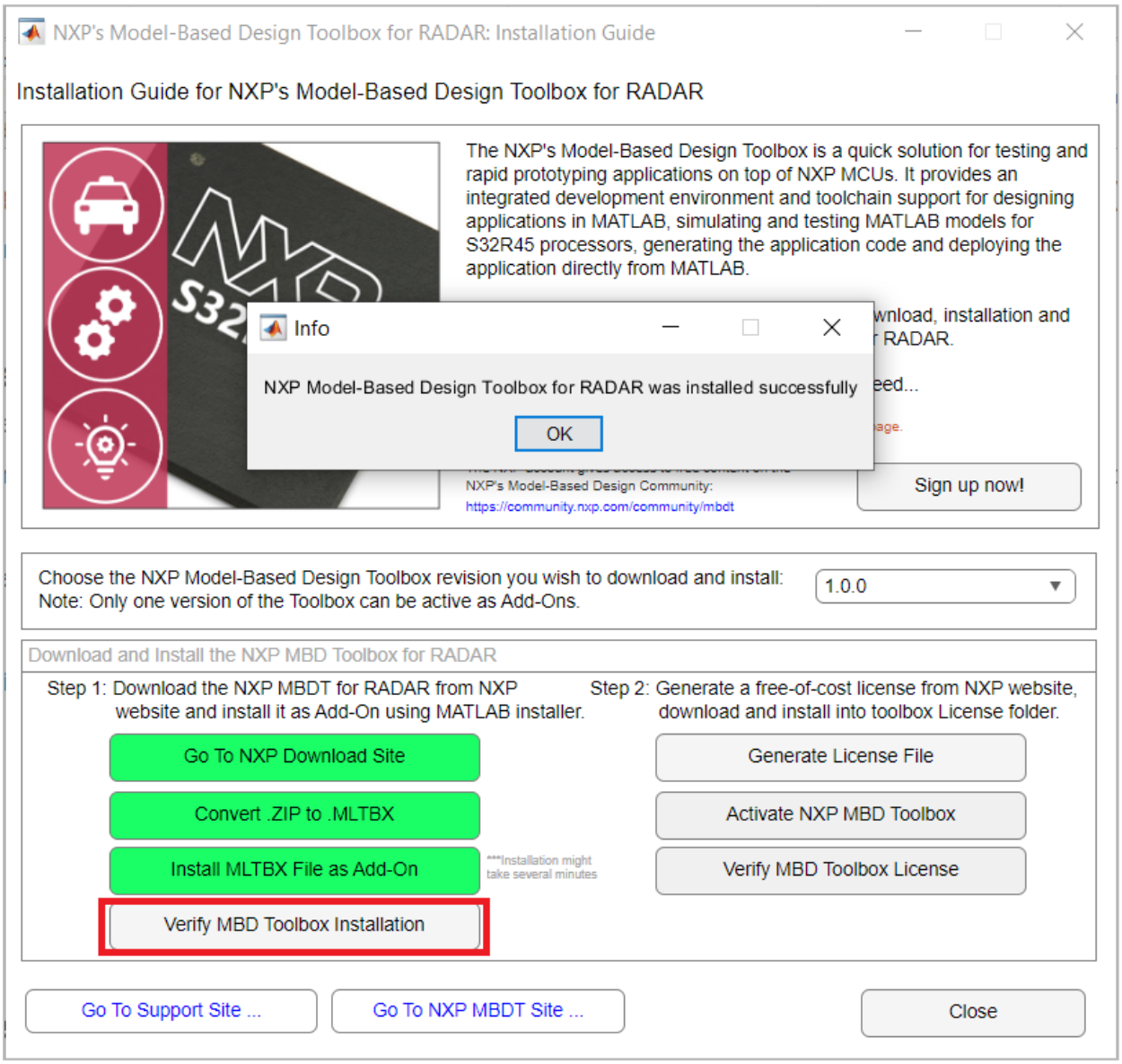


5. In the MATLAB Add-On Manager, Review the Terms and Conditions as you scroll down, and press the **I Accept** Button. This action starts MBDT for RADAR toolbox installation process.

   Note: Installation might take several minutes.



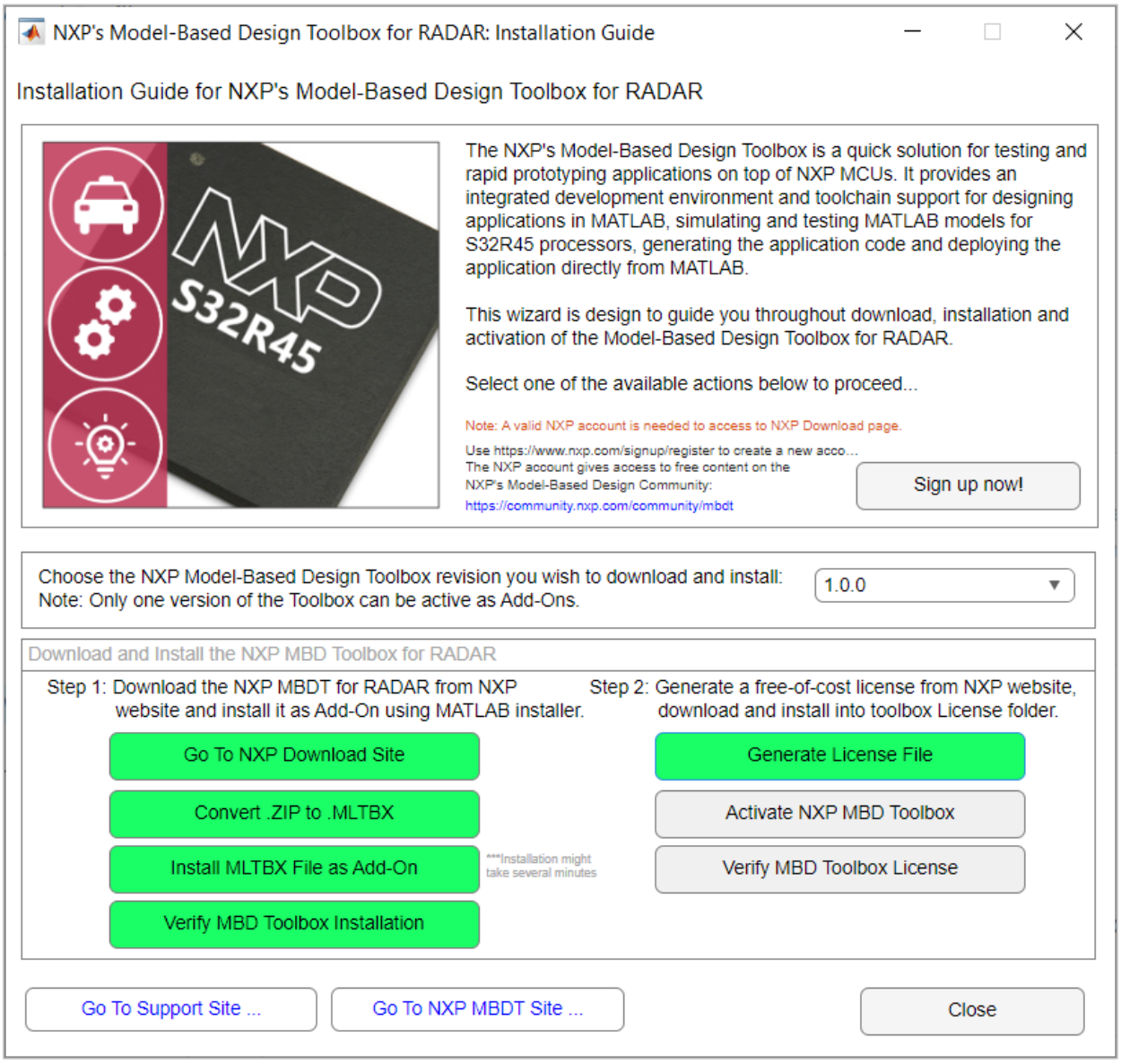
6. Once the installation is complete, go back to **NXP Toolbox Installation Guide** and press the **Verify MBD Toolbox Installation button**.



1.2.4 Generate and Activate NXP Model-Based Design Toolbox for RADAR license

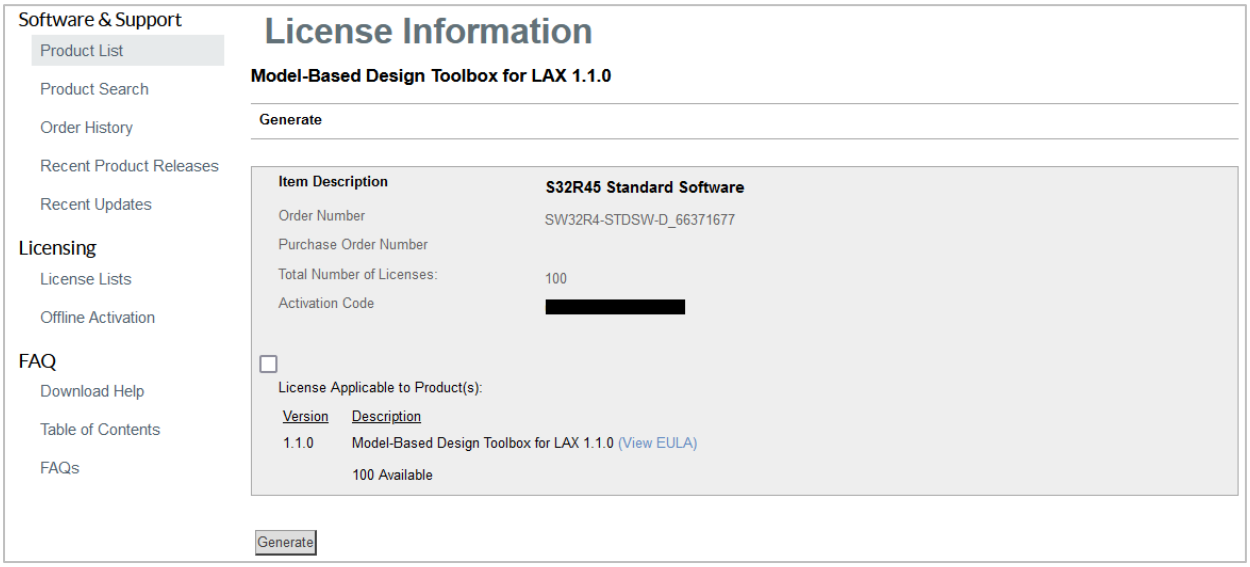
The following steps guide you on how to achieve the license for MBDT for RADAR toolbox.

1. Press the Generate License File button in the **NXP Toolbox Installation Guide.**

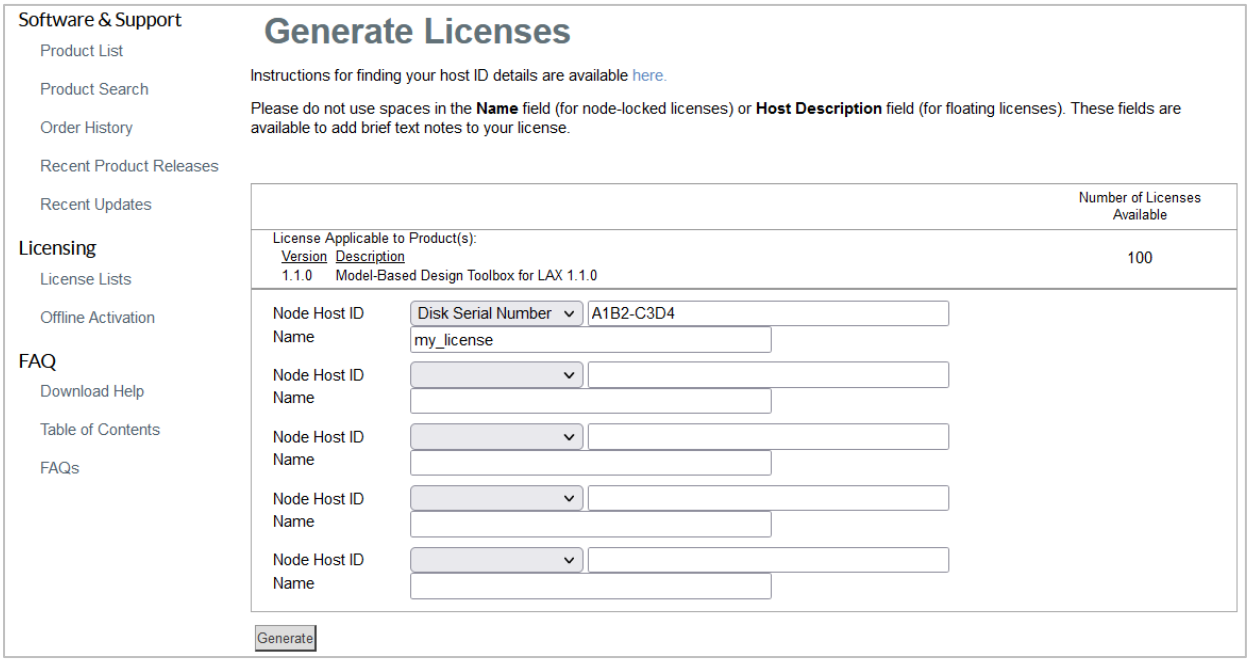


2. In the newly opened webpage, select the checkbox as shown below, and press the generate button.

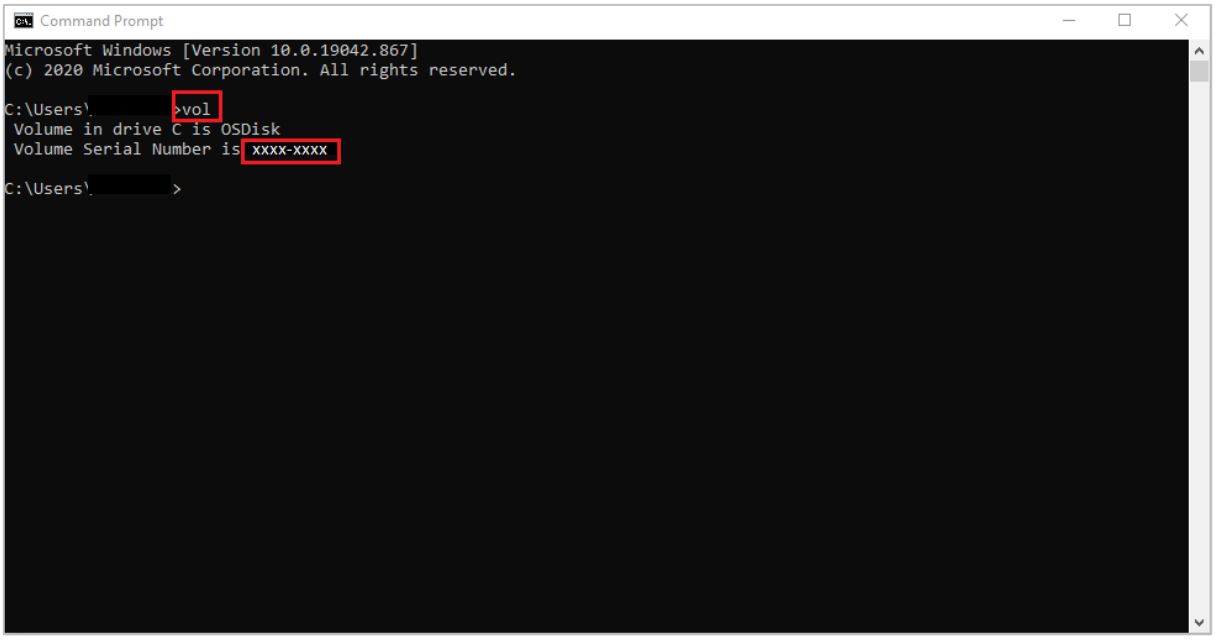
   Note: If a similar webpage as shown below is not being displayed, please go to the same page as described in the previous section, bullet 2, where the next tab "License keys".



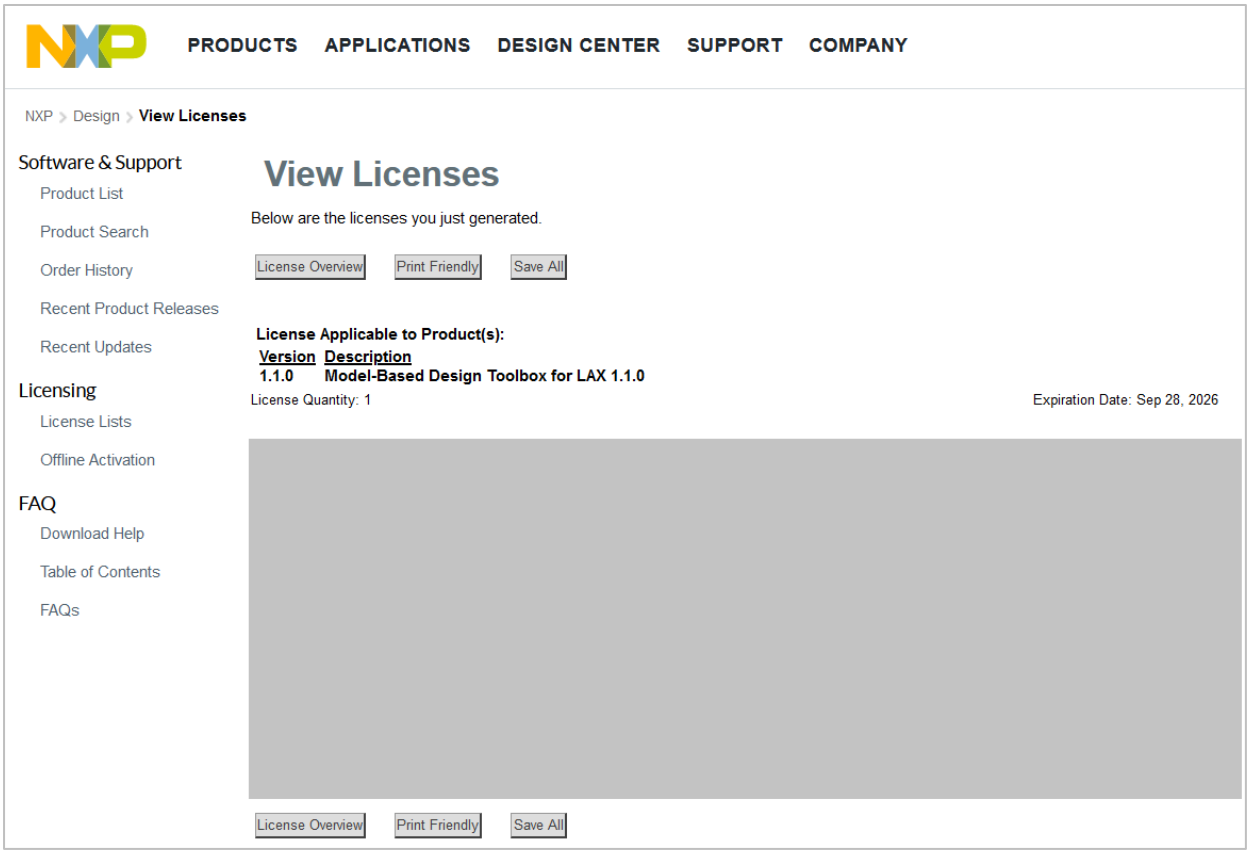
3. Select Disk Serial Number and type the host id number. Give a name to the license and press the Generate button.



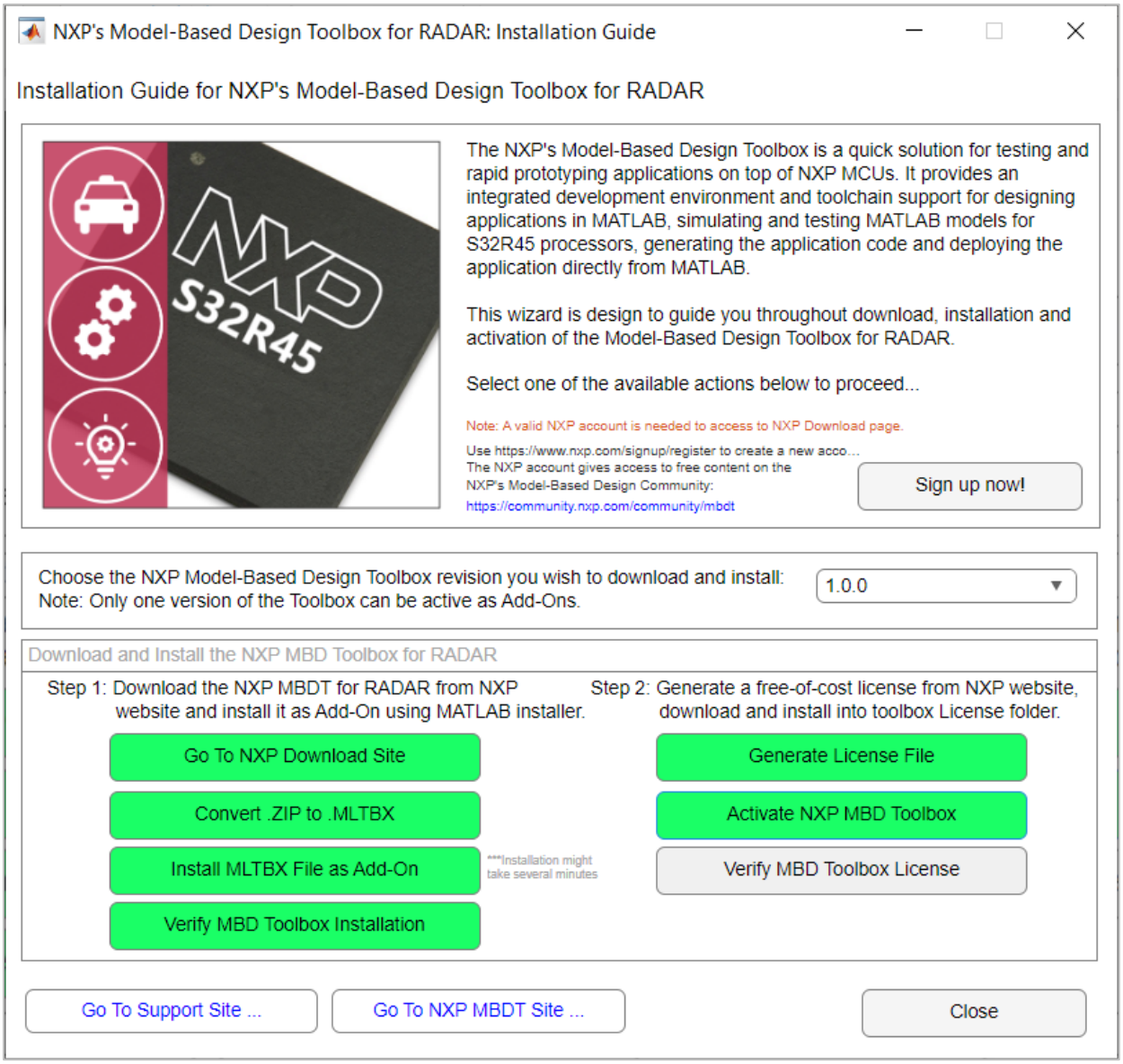
4. To find the host ID for your hard drive, please open a Windows Command Prompt and execute the “vol” command.



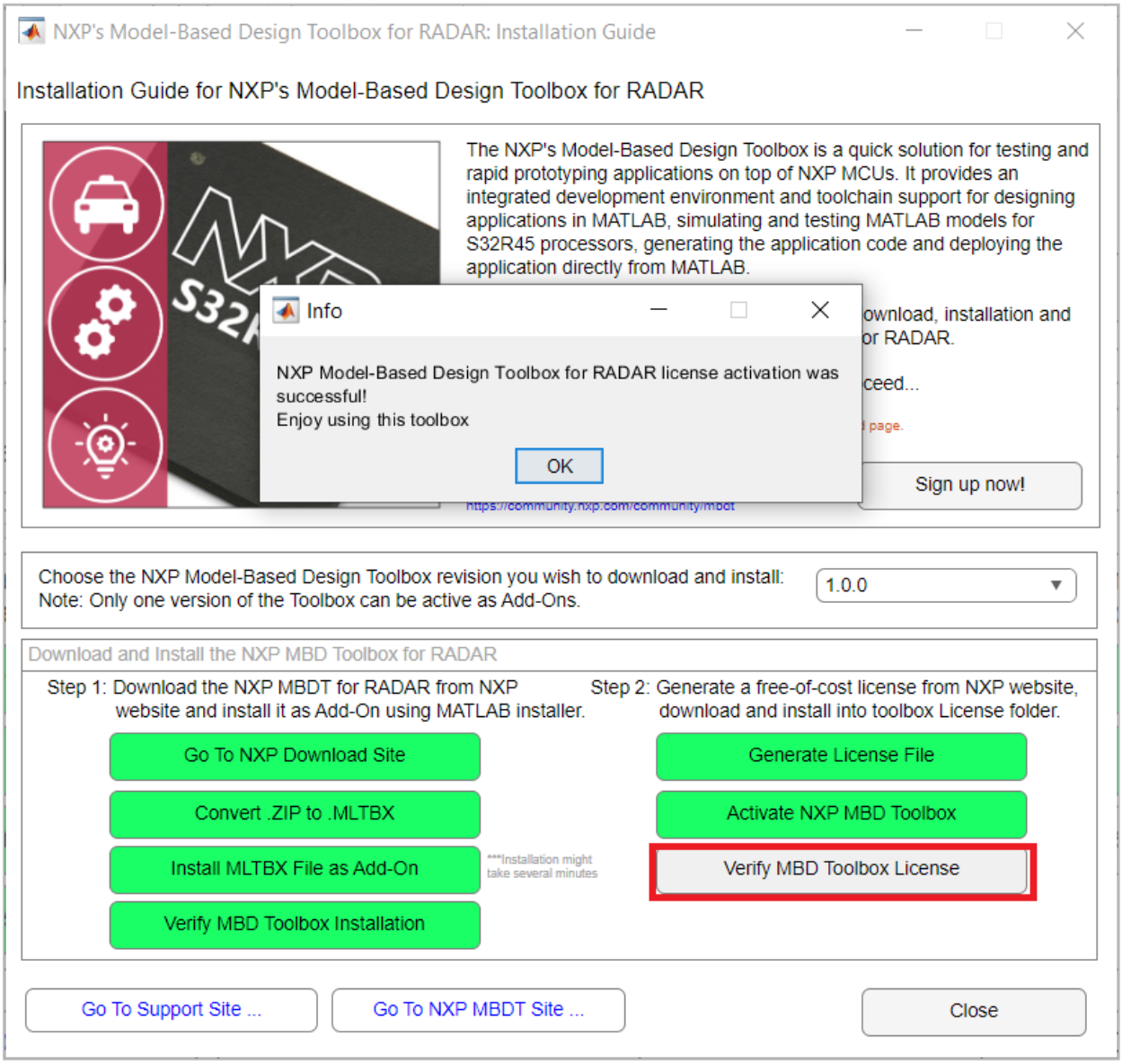
5. Now that the license has been successfully generated, press the Save all button.



6. Back to **NXP Toolbox Installation Guide,** press the **Activate NXP MBD Toolbox** button. In the newly opened window, browse for the downloaded license.dat or license.lic file, and press Open.

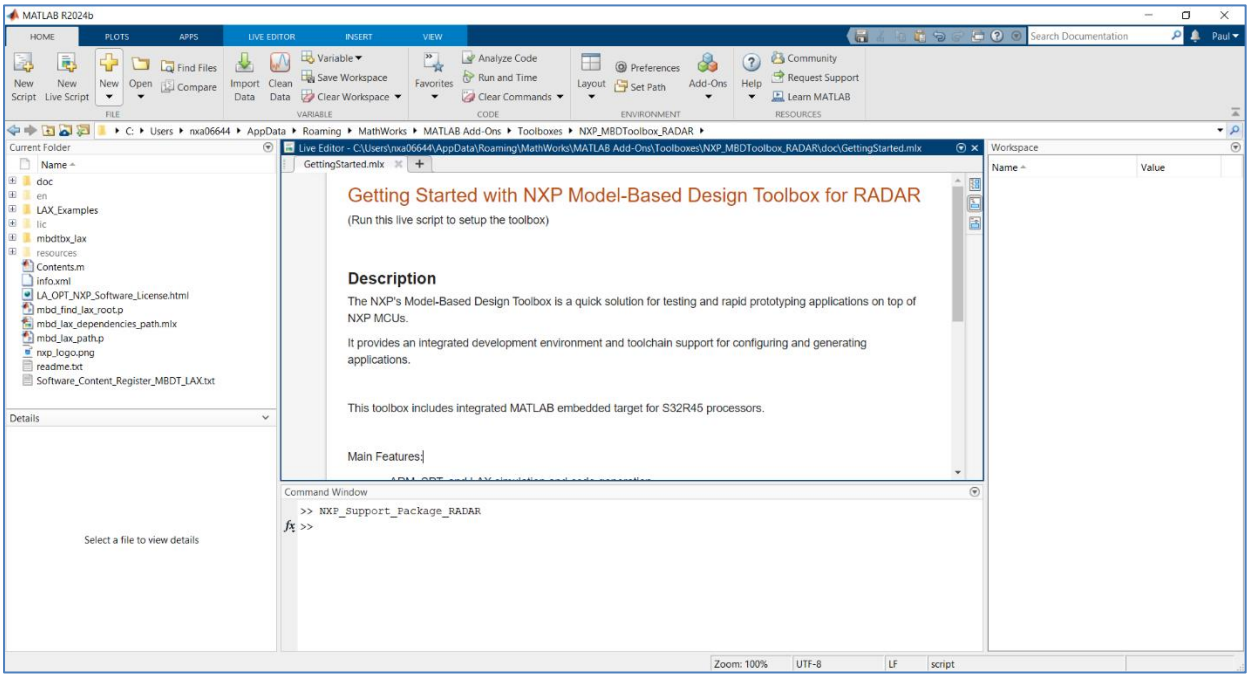


7. The last step is to check the license activation status, by pressing the **Verify MBD Toolbox License** button. If everything went well, a similar popup window as below will be displayed.



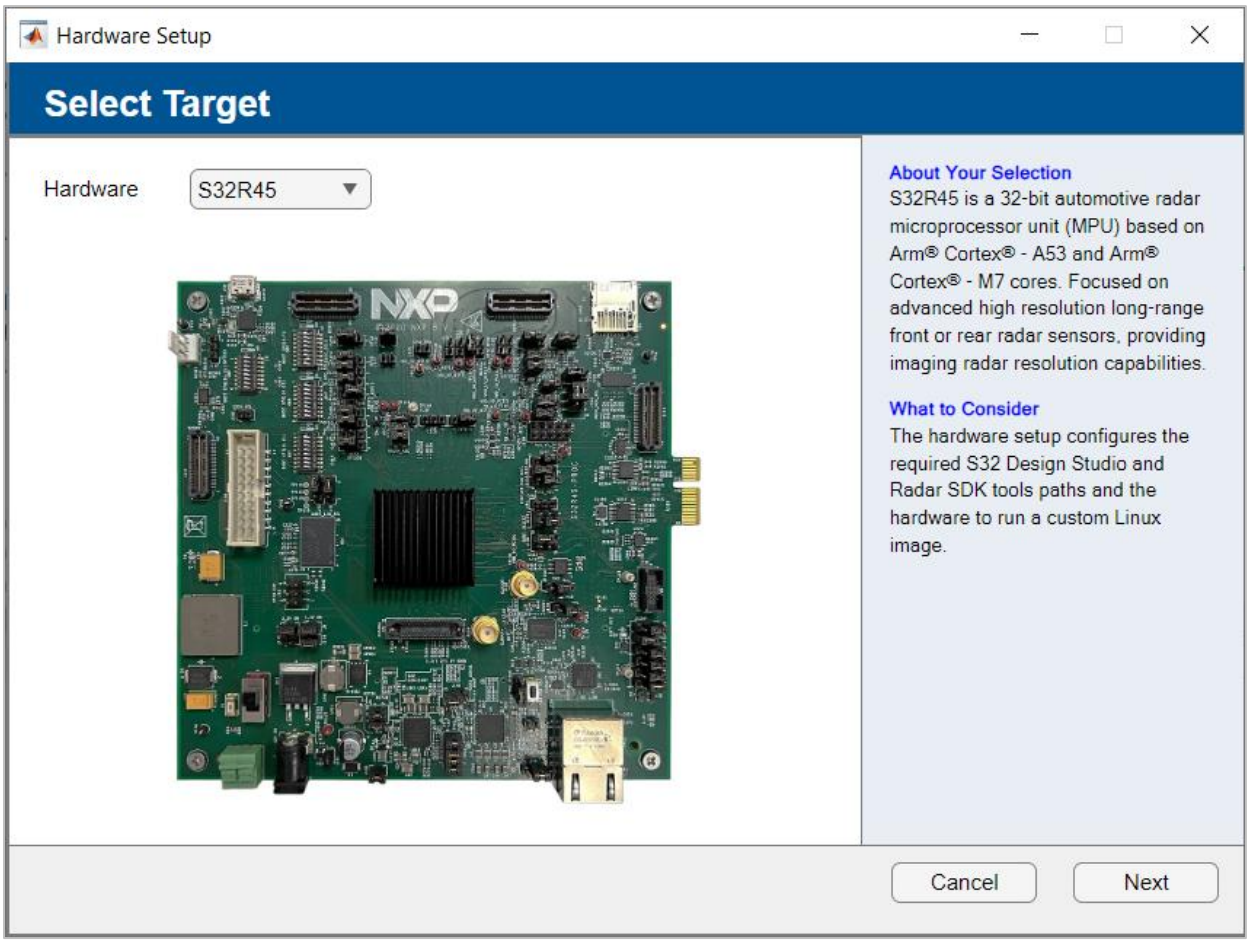
1.2.5 **Model-Based Design Toolbox Configuration**

After the installation has finished the *GettingStarted.mlx* is opened. To run it go to the LIVE EDITOR tab and push the Run button. Running it will open the Setup GUI that helps the user to configure the toolbox.



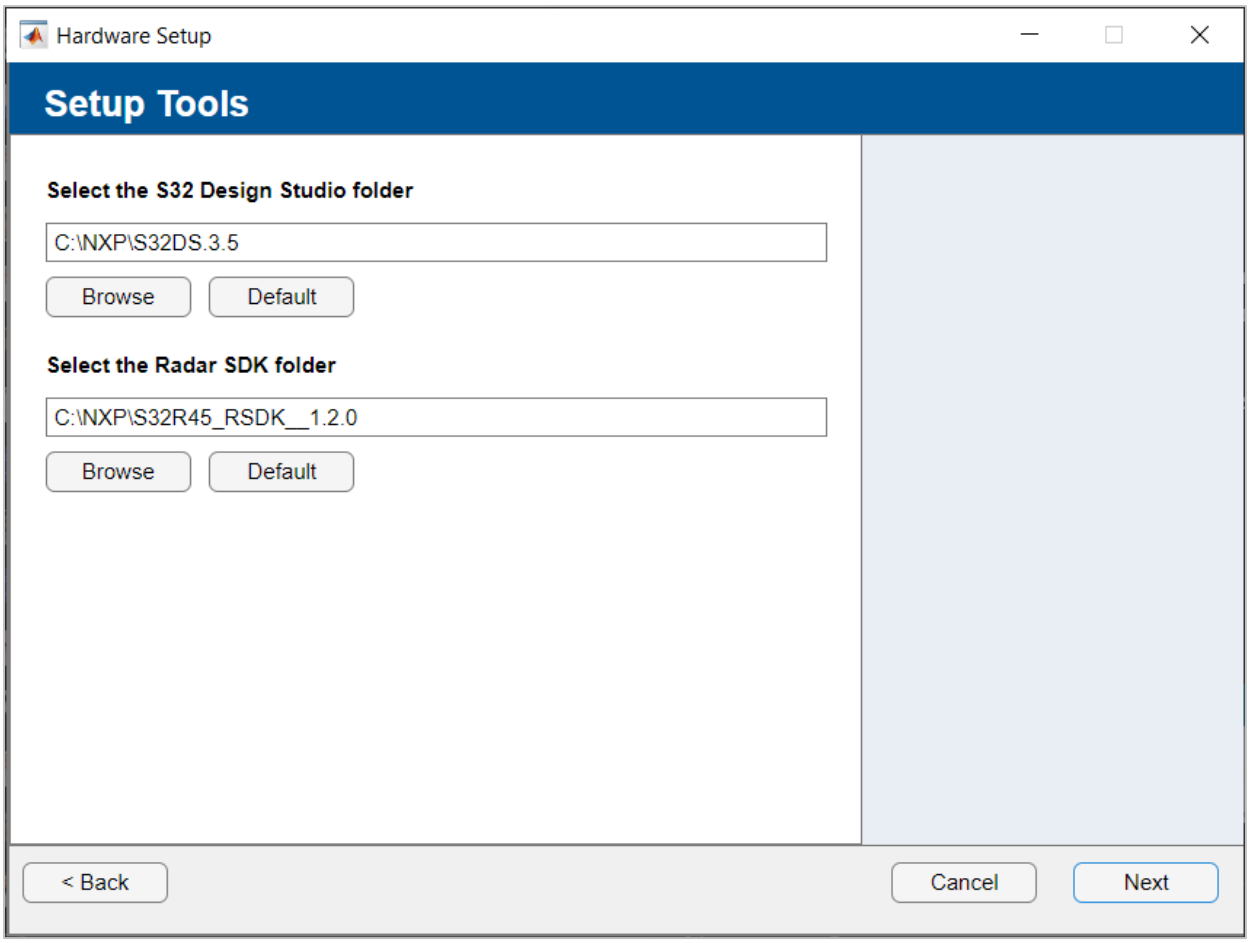
1. **Select the hardware board**

We are going to select S32R45 and push the Next button.



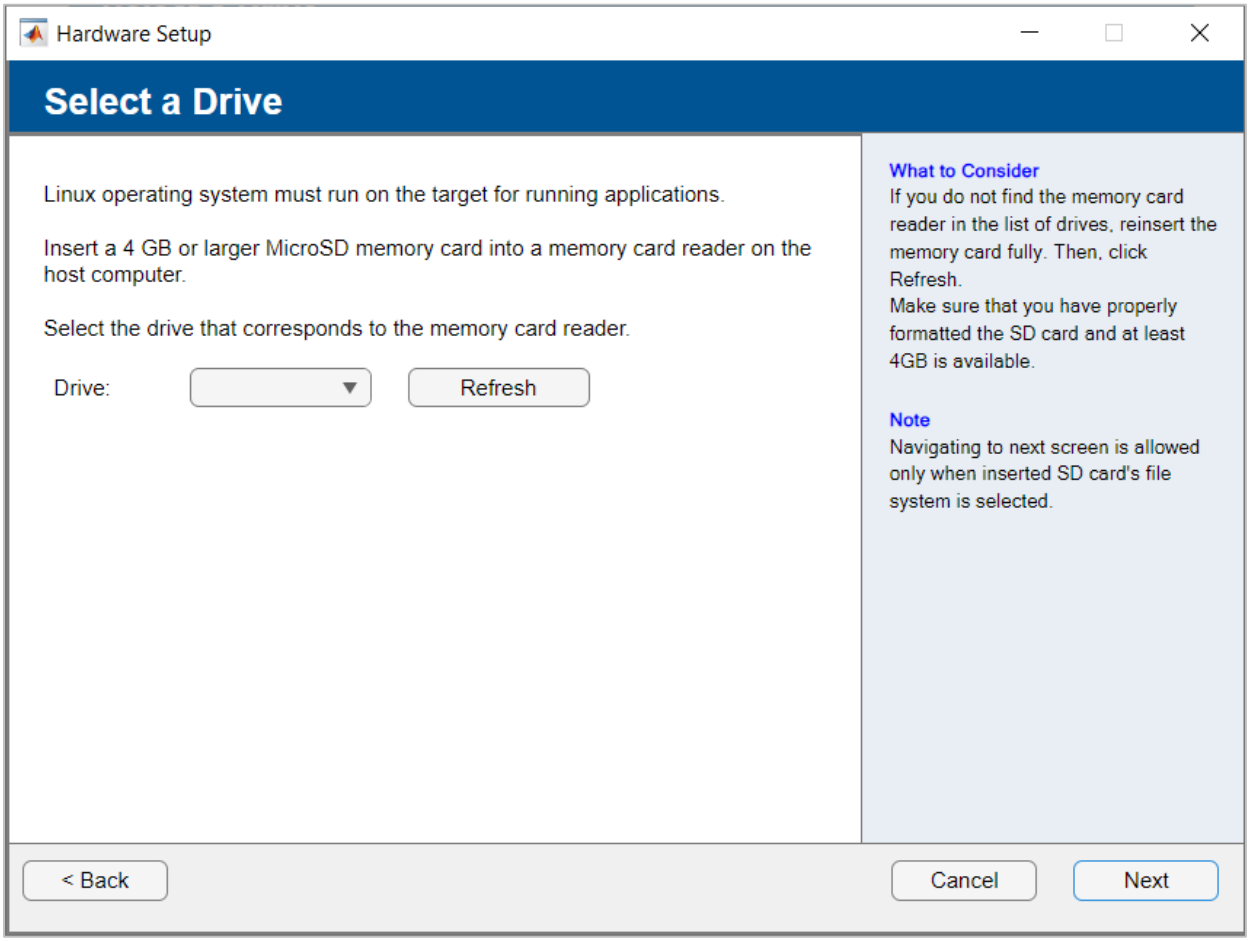
2. Setup the external tools

Select the needed external tools paths.

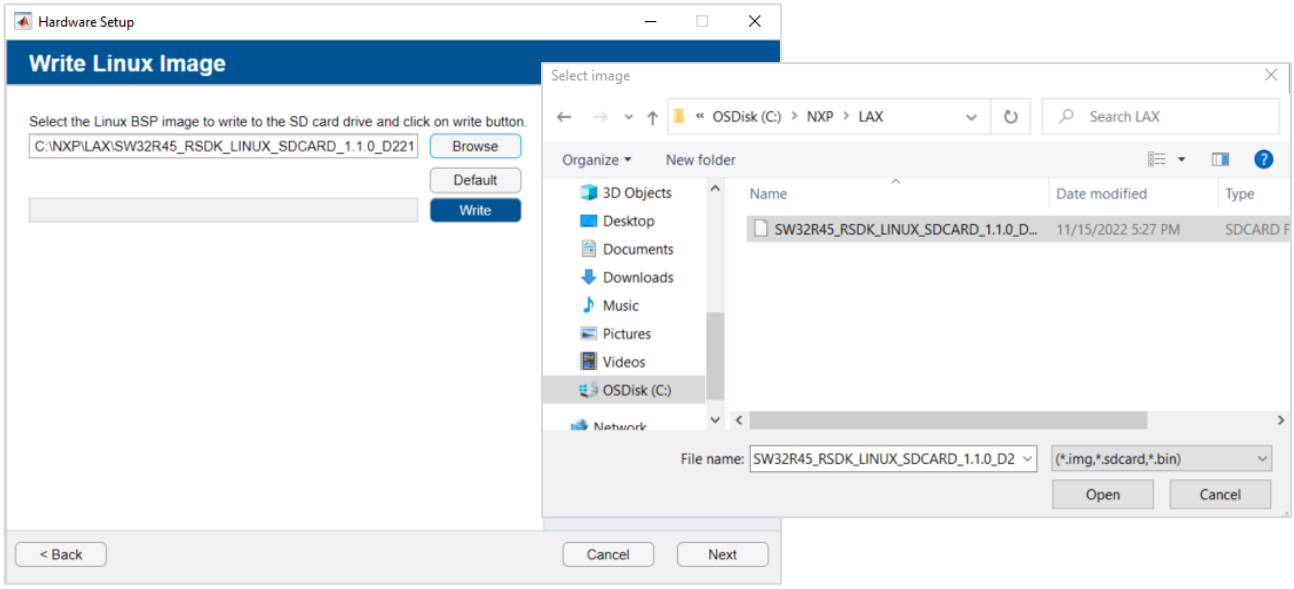


3. Write Linux on microSD card

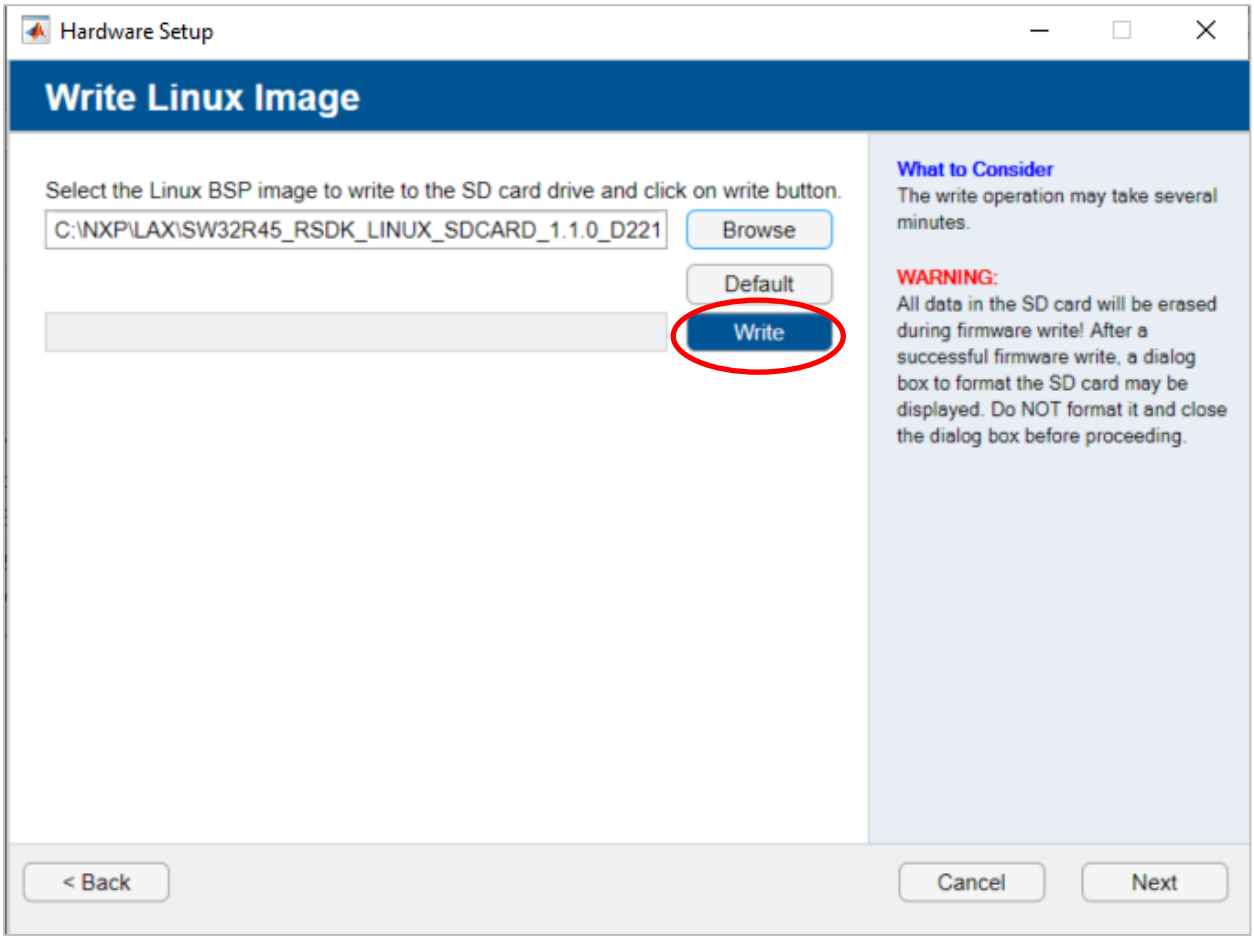
The next page helps you to write Linux on a microSD card. First, insert the microSD card, then push the Refresh button to actualize the drive list. If the microSD card is available in Windows, the list should not be empty. Select the drive from the list and push the Next button.



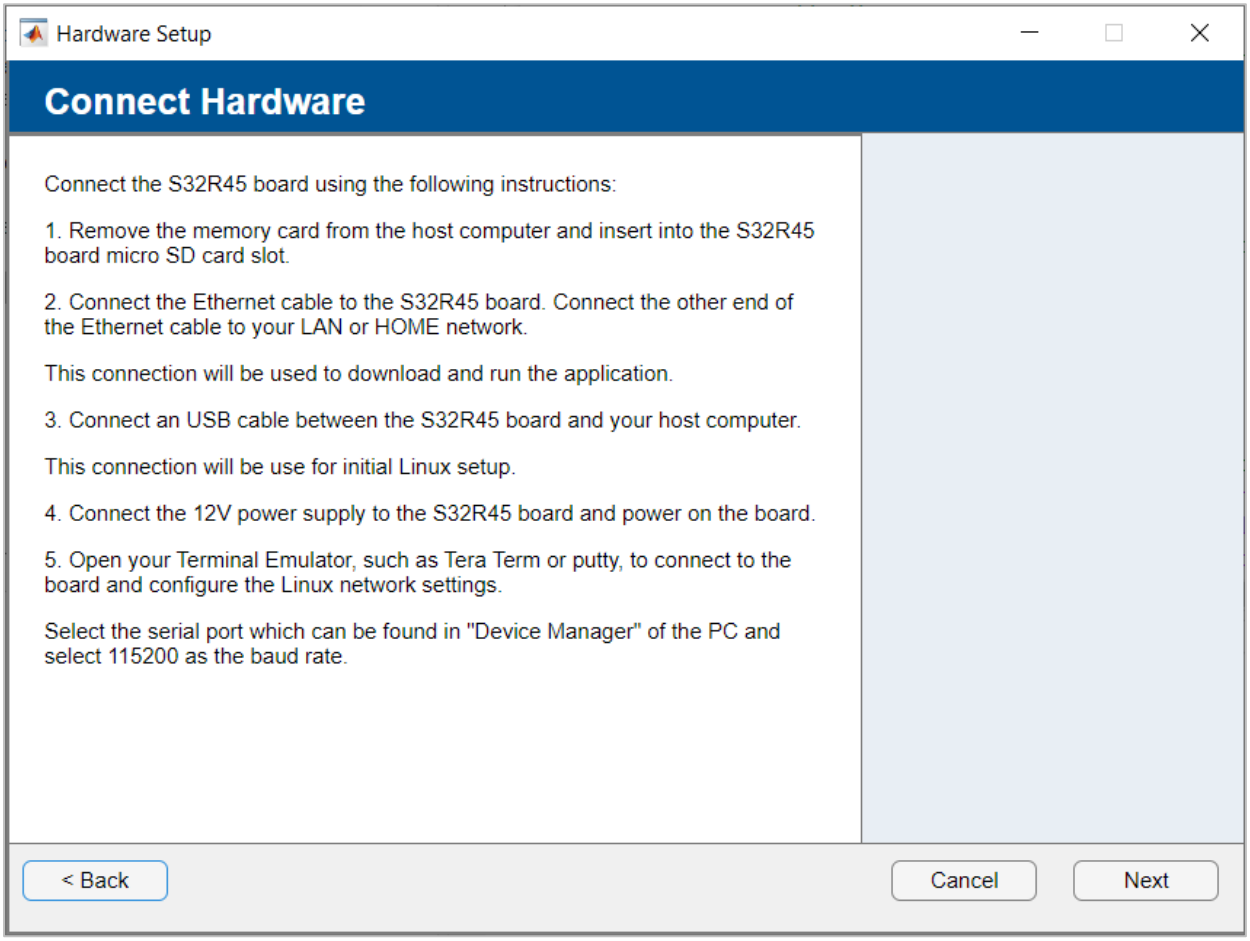
Select the Linux image using the Browse button.



Finally, push the Write button to write the Linux image.

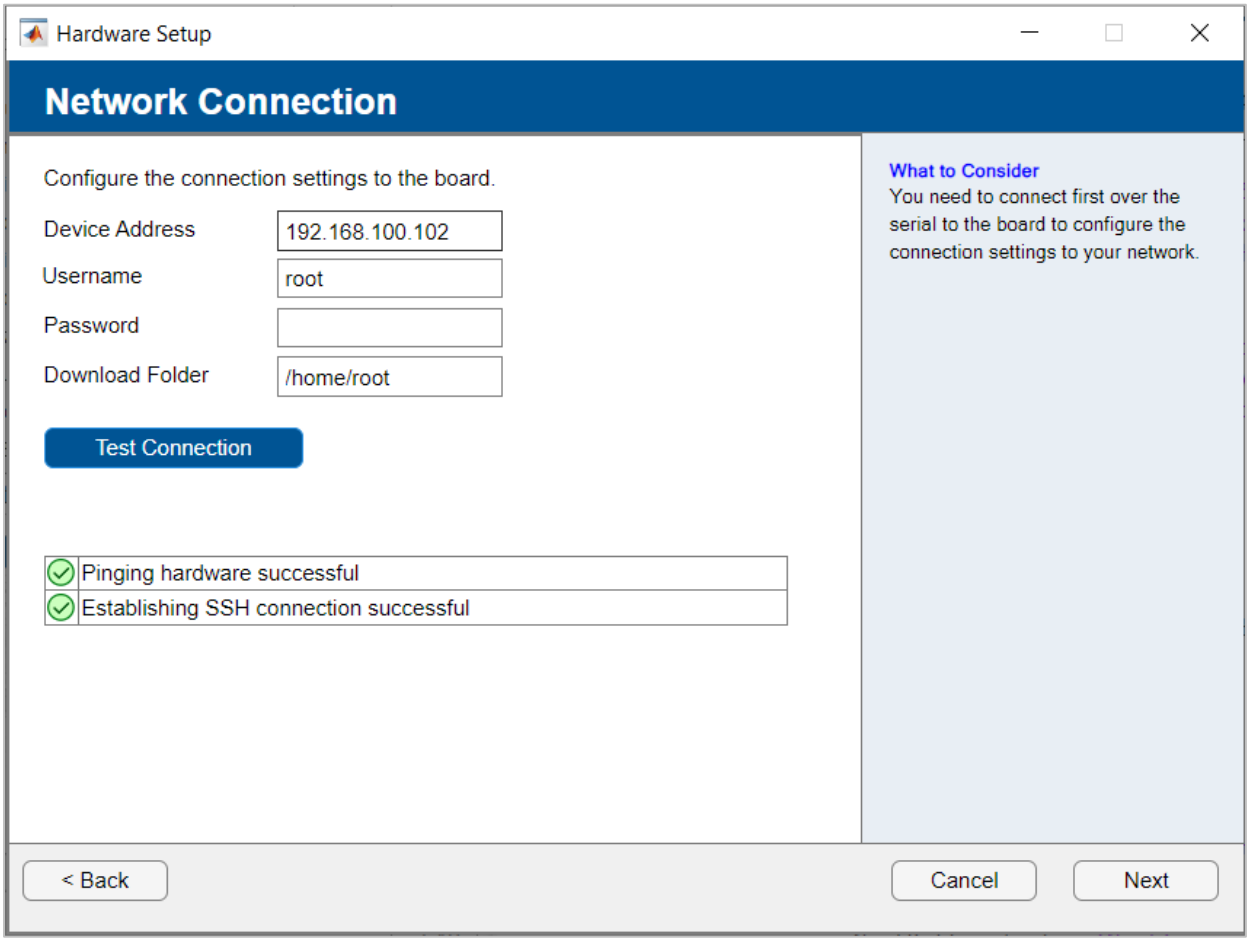


4. Get additional software



5. **Setup network connection**

Fill in the Device Address, Username, Password, and Download folder, and then push the Test Connection button. This tests if the device address is available and if the SSH commands can be run.



Setup done

Congratulations! You are ready to use the Model-Based Design for RADAR toolbox.

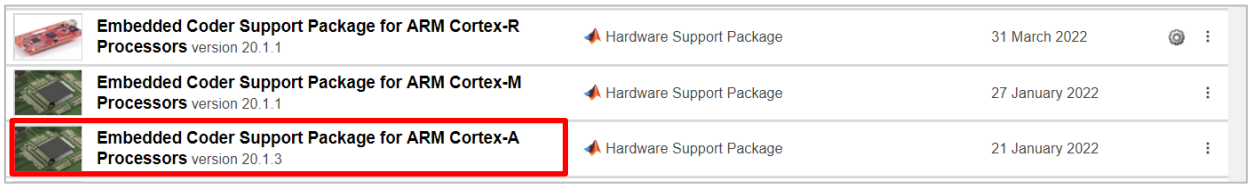
1.2.6 Setting the Path for Model-Based Design Toolbox and Toolchain Generation

The Model-Based Design Toolbox for RADAR uses the Toolchain mechanism exposed by the MATLAB to enable automatic code generation with the Embedded Coder toolbox. By default, the toolchain is configured for the MATLAB 2023a-2024b releases. For any other MATLAB release, the user needs to execute a toolbox m-script to generate the proper settings for his/her installation environment.

This is done by changing the MATLAB Current Directory to the toolbox installation directory (e.g.: ..\MATLAB\Add-Ons\Toolboxes\NXP_MBDToolbox_RADAR\) and running the "mbd_lax_path.m" script.

```
>> mbd_lax_path
Treating 'C:\MATLABAddOns\Toolboxes\NXP_MBDToolbox_RADAR\' as MBD
Toolbox installation root.
MBD Toolbox path prepended.
NXP S32 Design Studio GCC (S32R45) toolchain is already registered ...
No compatible target is currently available for NXP S32R45 - A53.
Creating one...
Creating folders for the target 'NXP S32R45 - A53' in the folder
'C:\MATLABAddOns\Toolboxes\NXP_MBDToolbox_RADAR\mbdtbx_lax\codertarget
\2023a\s32r45'...
Creating the framework for the target 'NXP S32R45 - A53'...
Registering the target 'NXP S32R45 - A53'...
Done.
Successful.
```

This mechanism requires users to install the Embedded Coder Support Package for ARM Cortex-A Processor as a prerequisite.



The "mbd_lax_path.m" script verifies the user setup dependencies and will issue instructions for a successful installation and configuration of the toolbox.

2 How to create, run and build LAX example applications

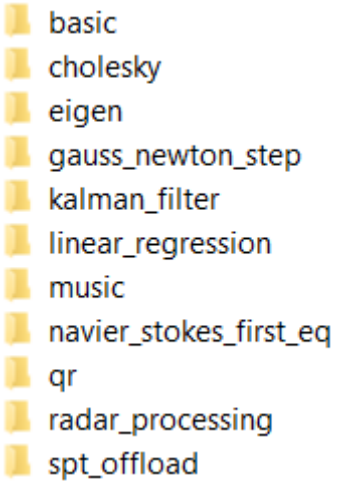
2.1.1 S32R45 Linux Setup

Before running any example on the S32R45 board for the first time, Linux must be configured to load the needed drivers. To do so, after logging into the Linux environment type the following commands. These configurations will be persistent, no need to reproduce them after each Linux reboot.

- echo "oal_driver" >> /etc/modules
- echo "rsdk_lax_driver" >> /etc/modules
- echo "rsdk_spt_driver" >> /etc/modules

2.1.2 Create LAX example

NXP's Model-Based Design Toolbox provides MATLAB code generation/simulation capabilities to run the examples for both A53 and LAX cores. The examples are separated into multiple folders.



Each category contains examples that can be run on the ARM core with graph functions that are running on the LAX accelerator.

Each LAX example contains a main function that will run on the ARM A53 core. The part of the code that will run on the LAX accelerator will be part of the lax graph function defined as *laxGraphName*, at it will be executed as *outputs = arm.laxRemoteExec(coreID, @laxGraphName, inputs)*.

```
function main_linear_regression()
    x = single([-6 -4 -1 0 3 5 8])';
    y = single([18 13 6 4 -1 -8 -15])';

    % plot the points
    if isempty(coder.target)
        hold on
        plot(x, y,  'x');
    end

    % y = b1*x+ b0;
    X = [x ones(size(x))];

    % Call the LAX Graph function
    b = arm.laxRemoteExec(@lax_func, X, y);

    % Plot the curve fit and the points
    if isempty(coder.target)
        plot(x, b(1)*x + b(2), 'b');
        hold('on');
        plot(x, y, 'rx');
        legend('curve fit', 'data');
    end
end
```

```
function b = lax_func(X, y)  %codegen
   coder.inline('always');

   b = inv(X.'*X)*X.'*y;
end
```

For more information about the available MATLAB lax operators see the MATLAB LAX Operators help section within the toolbox.

2.1.3 **Run LAX example**

To run the LAX examples, the user must go into the example directory and run the *build_example.m* script and this will start the code generation process. The example can be run also in Simulation mode by running the main function script with the example name.

In the build_example script, the user can select different options, optimization levels, connectivity settings, IO, etc.

```
config = struct();
config.HardwarePart = 'S32R45';
config.Hostname = '192.168.1.2';
config.Username = 'root';
config.Password = '';
config.Deploy = true;
config.DeployPath = '~/examples/linear_regression/';
config.PutFiles = {};
config.GetFiles = {};
config.BuildConfig = 'Faster Runs';

lax_codegen('main_linear_regression.m', config);
```

How to Reach Us:

Home Page:
www.nxp.com

Web Support:
www.nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

NXP Semiconductor reserves the right to make changes without further notice to any products herein. NXP Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. NXP Semiconductor does not convey any license under its patent rights nor the rights of others. NXP Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the NXP Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use NXP Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold NXP Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that NXP Semiconductor was negligent regarding the design or manufacture of the part.

MATLAB, Simulink, Stateflow, Handle Graphics, and Real-Time Workshop are registered trademarks, and TargetBox is a trademark of The MathWorks, Inc.
Microsoft and .NET Framework are trademarks of Microsoft Corporation.
Flexera Software, Flexlm, and FlexNet Publisher are registered trademarks or trademarks of Flexera Software, Inc. and/or InstallShield Co. Inc. in the United States of America and/or other countries.
NXP, the NXP logo, CodeWarrior and ColdFire are trademarks of NXP Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Flexis and Processor Expert are trademarks of NXP Semiconductor, Inc. All other product or service names are the property of their respective owners