Model-Based Design Toolbox S32Z/E Series

Quick Start Guide

Automatic Code Generation for the S32Z/E Family of Processors
Version 1.3.0

Target Based Automatic Code Generation Tools
For MATLAB™/Simulink™/Stateflow™ Models working with Simulink 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/Simulink for NXP's embedded target processors and development boards.
The S32Z/E Model-Based Design Toolbox supports code generation for:

- ARM Cortex-R52 Cluster 0 Core 0
- ARM Cortex-R52 Cluster 0 Core 1
- ARM Cortex-R52 Cluster 0 Core 2
- ARM Cortex-R52 Cluster 0 Core 3
- ARM Cortex-R52 Cluster 1 Core 0
- ARM Cortex-R52 Cluster 1 Core 1
- ARM Cortex-R52 Cluster 1 Core 2
- ARM Cortex-R52 Cluster 1 Core 3
- SPF2 core

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, as the Simulators and Compilers.
3. Setup the MATLAB path for Model-Based Design Toolbox and generate the appropriate toolchain setting for the user MATLAB environment.

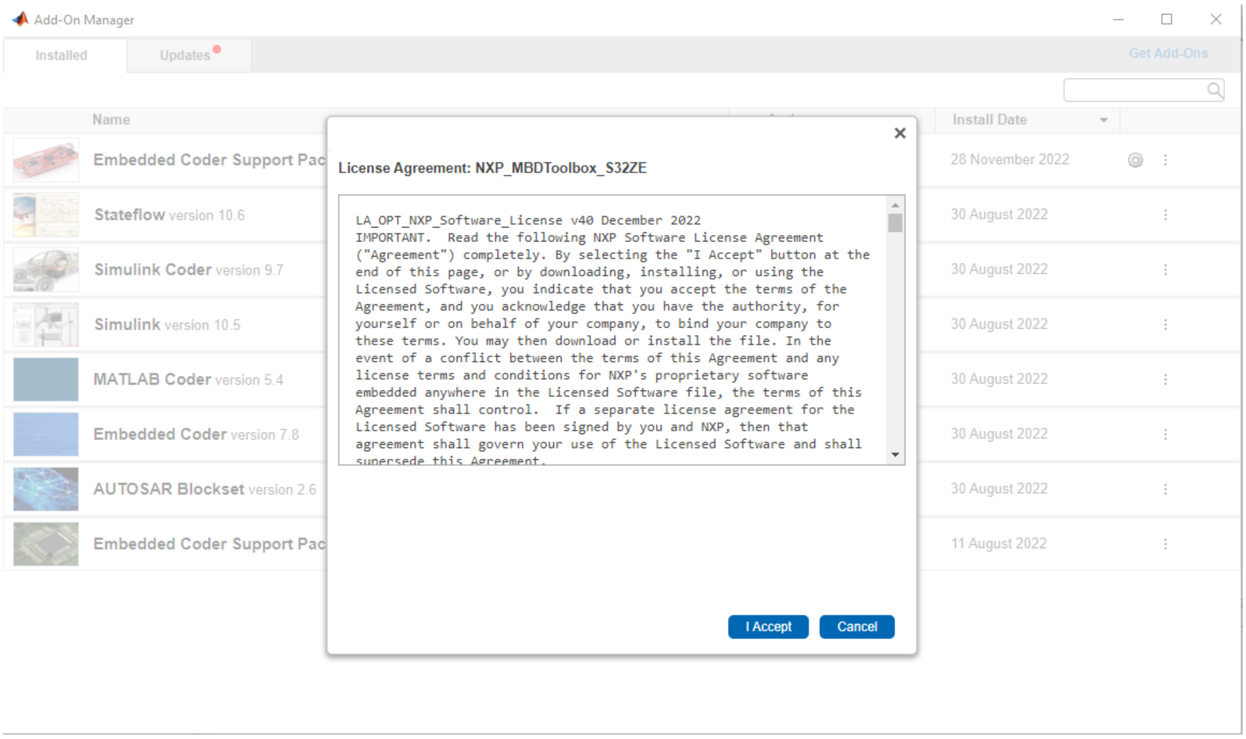
Each of these actions is explained in the following sub-chapters.

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:

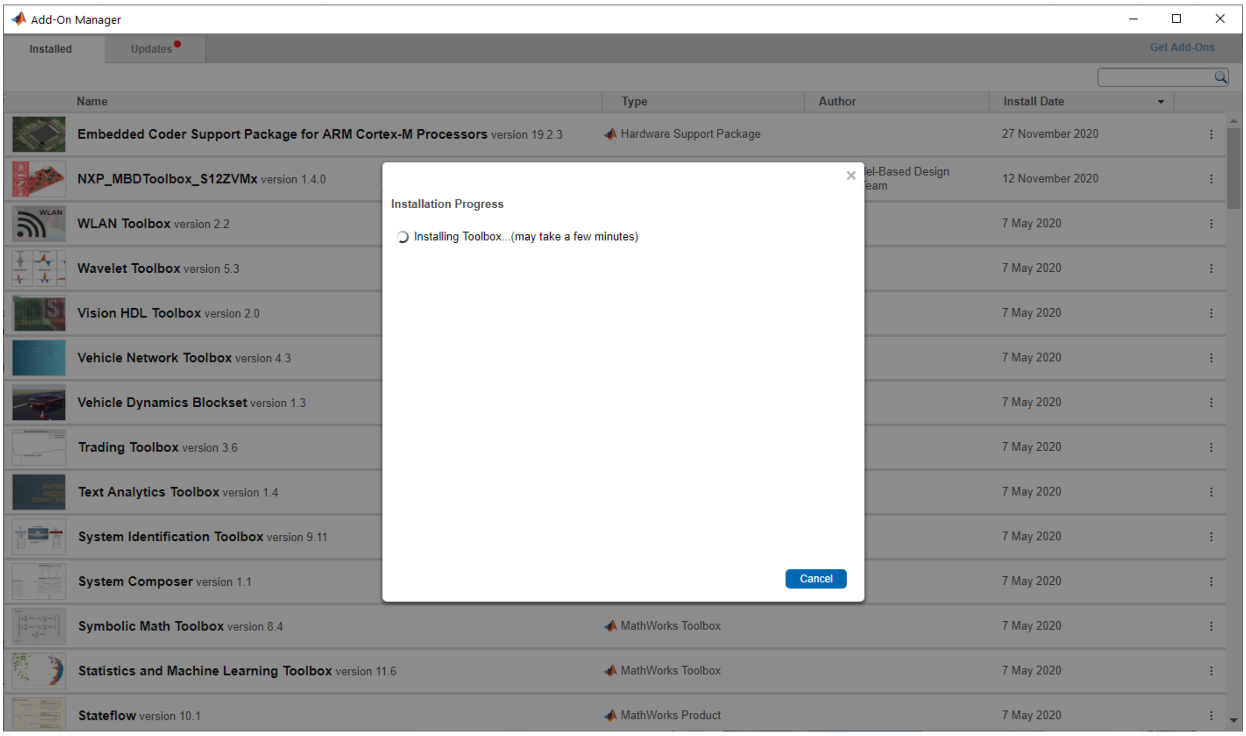
1. Indicate acceptance of the NXP Software License Agreement by selecting “`I Accept`” to proceed.



License Agreement: NXP_MBDToolbox_S32ZE

```
LA_OPT_NXP_Software_License v40 December 2022
IMPORTANT.  Read the following NXP Software License Agreement
("Agreement") completely. By selecting the "I Accept" button at the
end of this page, or by downloading, installing, or using the
Licensed Software, you indicate that you accept the terms of the
Agreement, and you acknowledge that you have the authority, for
yourself or on behalf of your company, to bind your company to
these terms. You may then download or install the file. In the
event of a conflict between the terms of this Agreement and any
license terms and conditions for NXP's proprietary software
embedded anywhere in the Licensed Software file, the terms of this
Agreement shall control.  If a separate license agreement for the
Licensed Software has been signed by you and NXP, then that
agreement shall govern your use of the Licensed Software and shall
supersede this Agreement.
```

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.

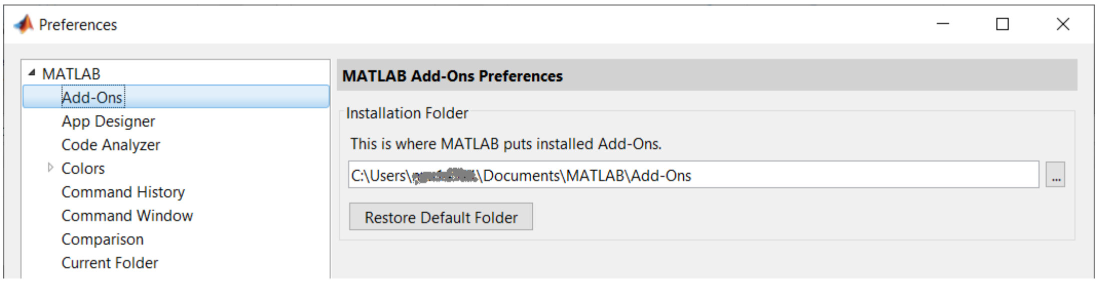


Name	Type	Author	Install Date
Embedded Coder Support Package for ARM Cortex-M Processors version 19.2.3	Hardware Support Package		27 November 2020
NXP_MBDToolbox_S12ZVMx version 1.4.0		el-Based Design eam	12 November 2020
WLAN Toolbox version 2.2			7 May 2020
Wavelet Toolbox version 5.3			7 May 2020
Vision HDL Toolbox version 2.0			7 May 2020
Vehicle Network Toolbox version 4.3			7 May 2020
Vehicle Dynamics Blockset version 1.3			7 May 2020
Trading Toolbox version 3.6			7 May 2020
Text Analytics Toolbox version 1.4			7 May 2020
System Identification Toolbox version 9.11			7 May 2020
System Composer version 1.1			7 May 2020
Symbolic Math Toolbox version 8.4	MathWorks Toolbox		7 May 2020
Statistics and Machine Learning Toolbox version 11.6	MathWorks Toolbox		7 May 2020
Stateflow version 10.1	MathWorks Product		7 May 2020

Installation Progress

Installing Toolbox...(may take a few minutes)

The default location can be changed before installation by changing the Add-Ons path from MATLAB Preferences.



MATLAB Add-Ons Preferences

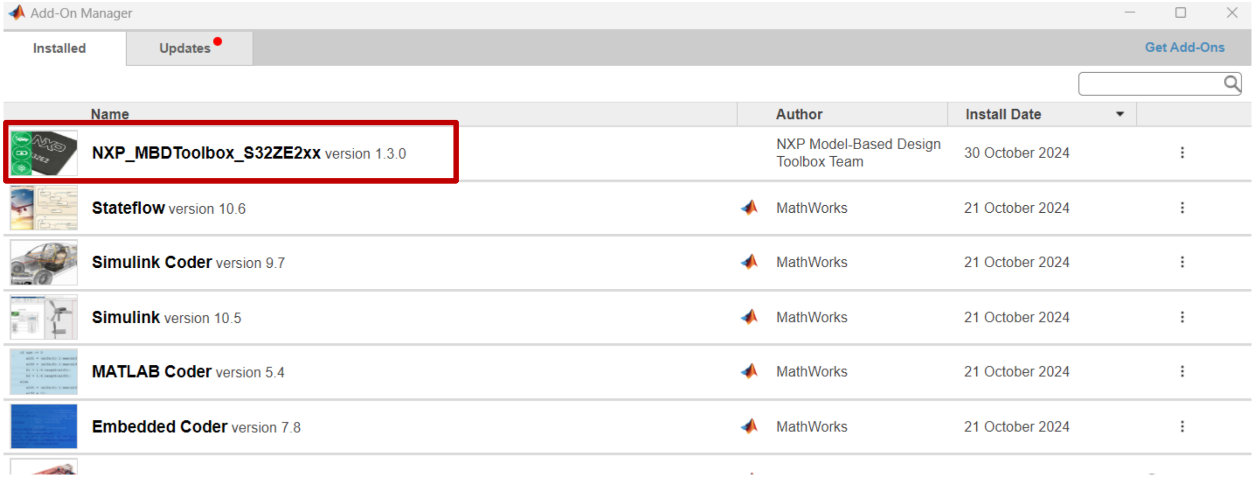
Installation Folder

This is where MATLAB puts installed Add-Ons.

C:\Users\[redacted]\Documents\MATLAB\Add-Ons

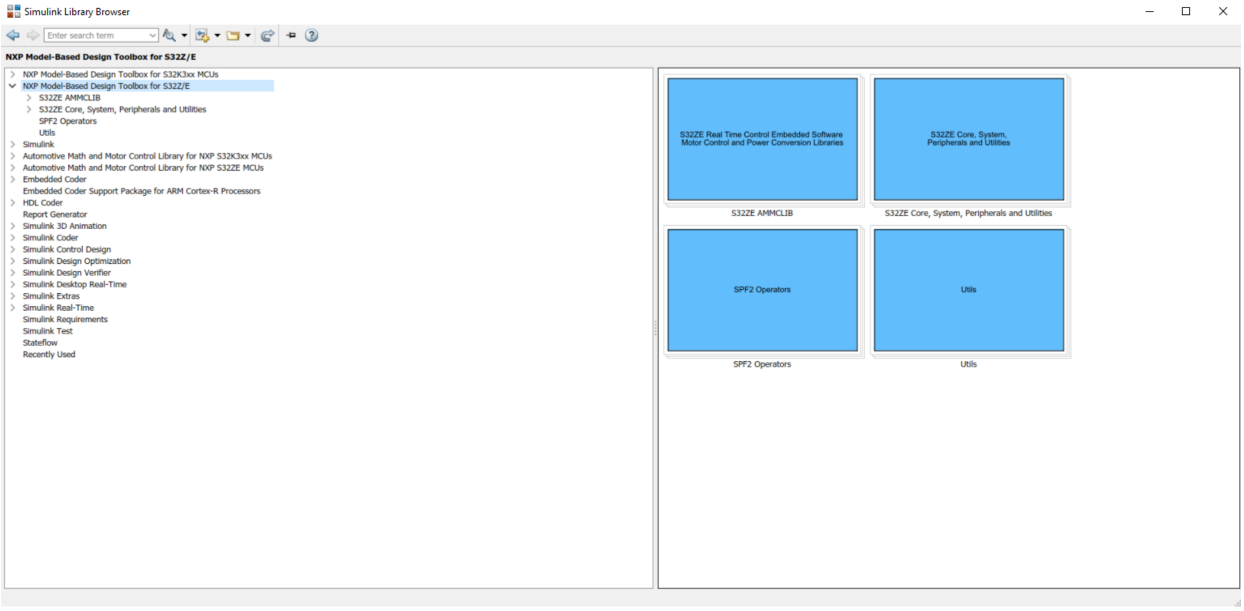
Note: *It is recommended to install the MATLAB and NXP Toolbox into a location that does not contain special characters, empty spaces, or mapped drives.*

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

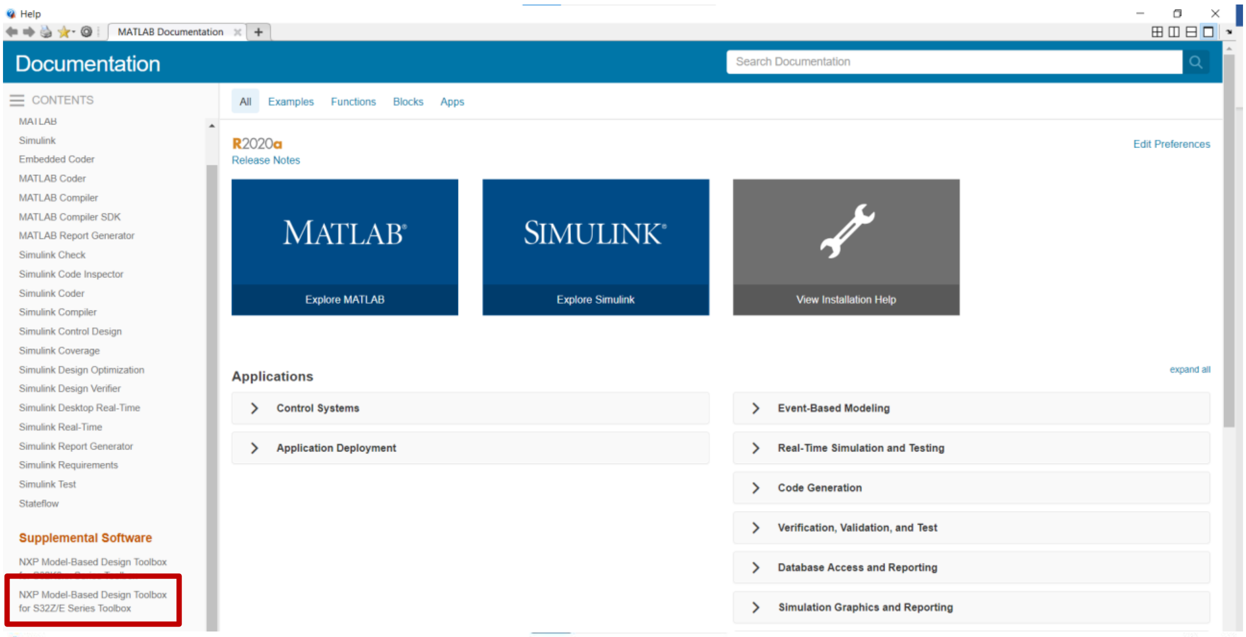


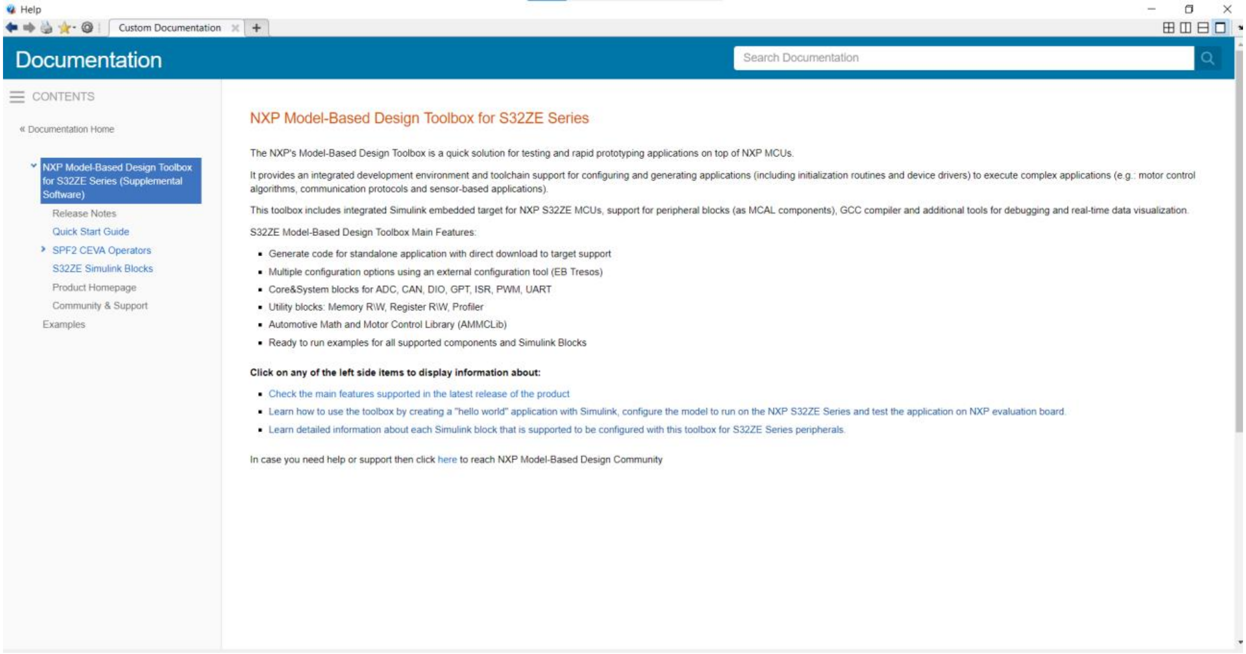
Name	Author	Install Date
NXP_MBDToolbox_S32ZE2xx version 1.3.0	NXP Model-Based Design Toolbox Team	30 October 2024
Stateflow version 10.6	MathWorks	21 October 2024
Simulink Coder version 9.7	MathWorks	21 October 2024
Simulink version 10.5	MathWorks	21 October 2024
MATLAB Coder version 5.4	MathWorks	21 October 2024
Embedded Coder version 7.8	MathWorks	21 October 2024

4. NXP’s Model-Based Design Toolbox layout and Simulink Library are shown below.


Simulink Library Browser
Enter search term
NXP Model-Based Design Toolbox for S32Z/E
NXP Model-Based Design Toolbox for S32K3xx MCUs
NXP Model-Based Design Toolbox for S32Z/E
S32ZE AMMCLIB
S32ZE Core, System, Peripherals and Utilities
SPF2 Operators
Utils
Simulink
Automotive Math and Motor Control Library for NXP S32K3xx MCUs
Automotive Math and Motor Control Library for NXP S32ZE MCUs
Embedded Coder
Embedded Coder Support Package for ARM Cortex-R Processors
HDL Coder
Report Generator
Simulink 3D Animation
Simulink Coder
Simulink Control Design
Simulink Design Optimization
Simulink Design Verifier
Simulink Desktop Real-Time
Simulink Extras
Simulink Real-Time
Simulink Requirements
Simulink Test
Stateflow
Recently Used
S32ZE Real Time Control Embedded Software Motor Control and Power Conversion Libraries
S32ZE AMMCLIB
S32ZE Core, System, Peripherals and Utilities
S32ZE Core, System, Peripherals and Utilities
SPF2 Operators
SPF2 Operators
Utils
Utils

5. 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.





NXP Model-Based Design Toolbox for S32ZE Series

The NXP's Model-Based Design Toolbox is a quick solution for testing and rapid prototyping applications on top of NXP MCUs.

It provides an integrated development environment and toolchain support for configuring and generating applications (including initialization routines and device drivers) to execute complex applications (e.g.: motor control algorithms, communication protocols and sensor-based applications).

This toolbox includes integrated Simulink embedded target for NXP S32ZE MCUs, support for peripheral blocks (as MCAL components), GCC compiler and additional tools for debugging and real-time data visualization.

S32ZE Model-Based Design Toolbox Main Features:

- Generate code for standalone application with direct download to target support
- Multiple configuration options using an external configuration tool (EB Tresos)
- Core&System blocks for ADC, CAN, DIO, GPT, ISR, PWM, UART
- Utility blocks: Memory R\W, Register R\W, Profiler
- Automotive Math and Motor Control Library (AMMCLib)
- Ready to run examples for all supported components and Simulink Blocks

Click on any of the left side items to display information about:

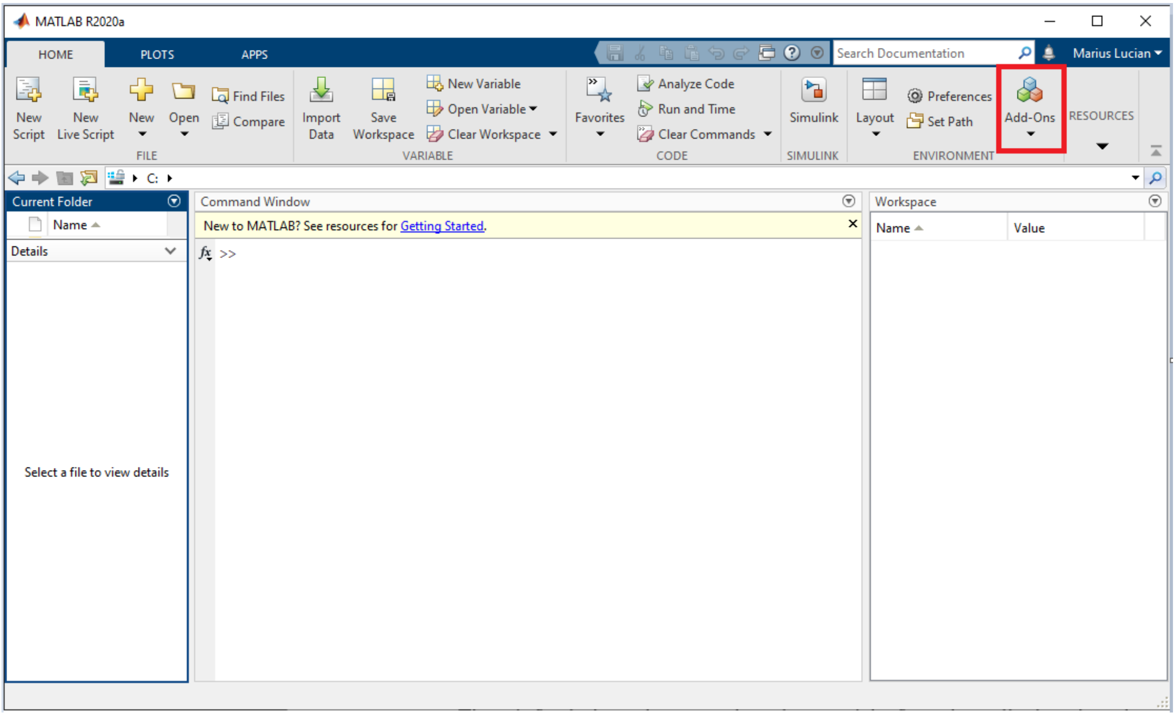
- Check the main features supported in the latest release of the product
- Learn how to use the toolbox by creating a "hello world" application with Simulink, configure the model to run on the NXP S32ZE Series and test the application on NXP evaluation board.
- Learn detailed information about each Simulink block that is supported to be configured with this toolbox for S32ZE Series peripherals.

In case you need help or support then click here to reach NXP Model-Based Design Community

1.2.2 Install NXP Support Package for S32Z/E (online)

This package will guide you through the download, installation, and activation process of the MBDT for S32Z/E online package.

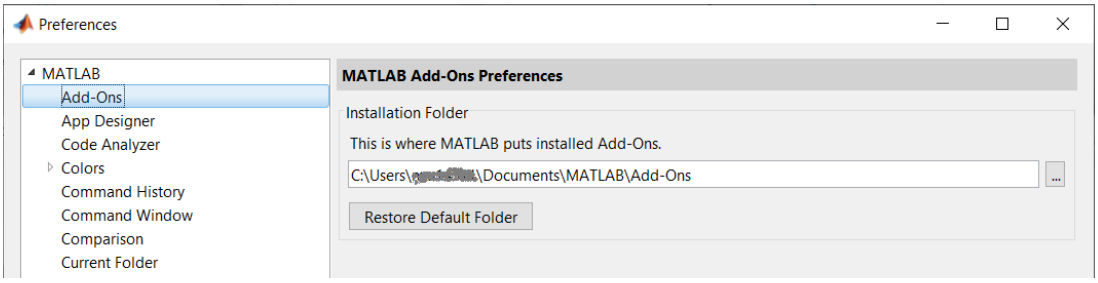
a. Go to MATLAB Add-Ons



New to MATLAB? See resources for Getting Started.

>>

Select a file to view details



MATLAB Add-Ons Preferences

Installation Folder

This is where MATLAB puts installed Add-Ons.

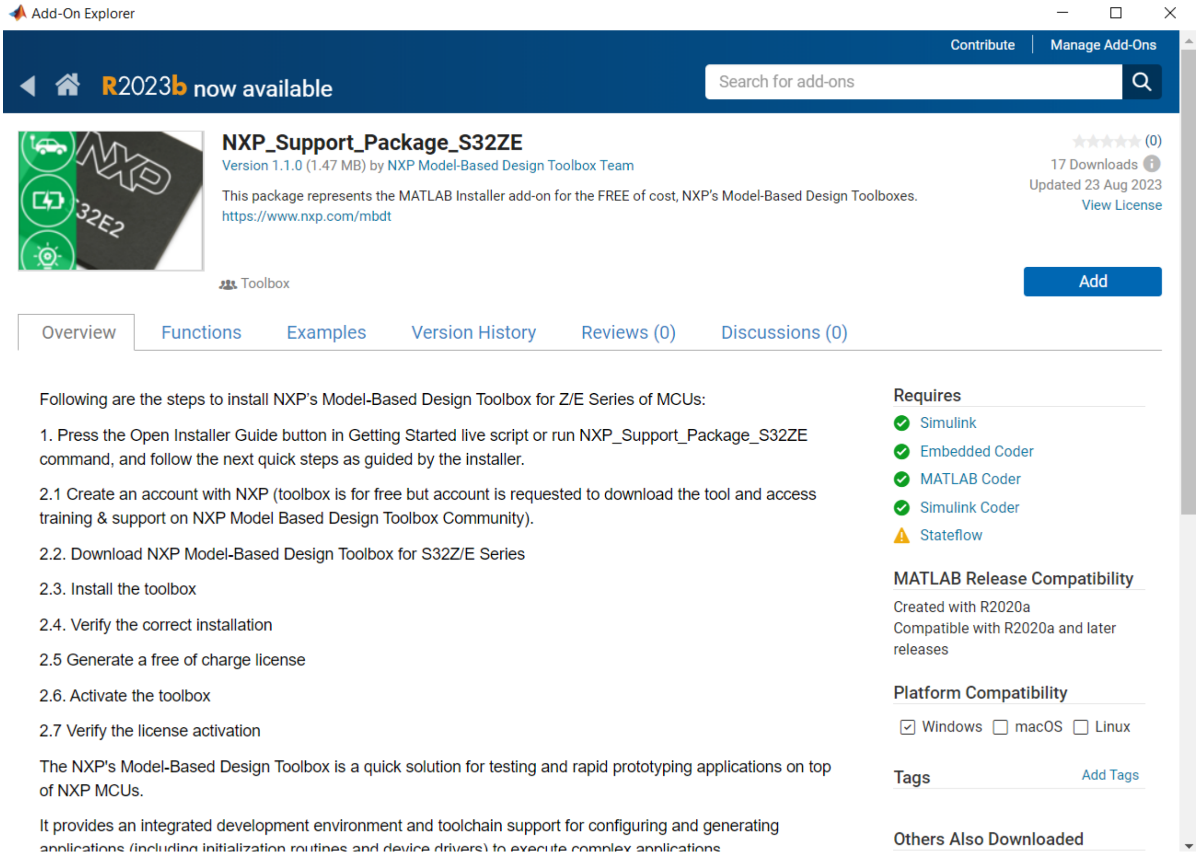
C:\Users\[redacted]\Documents\MATLAB\Add-Ons

The default location can be changed before installation by changing the Add-Ons path from MATLAB Preferences

> **Note:** It is recommended to install the MATLAB and NXP Toolbox into a location that does not contain special characters, empty spaces, or mapped drives. It is also recommended to install the Toolbox into a location with a short path to prevent potential errors related specifically to path length limitations.

b. Search for the “NXP Support Package S32ZE”

c. Install the “NXP Support Package S32ZE” by pressing the **Add** button.



NXP_Support_Package_S32ZE

Version 1.1.0 (1.47 MB) by NXP Model-Based Design Toolbox Team

This package represents the MATLAB Installer add-on for the FREE of cost, NXP's Model-Based Design Toolboxes.

https://www.nxp.com/mbdt

(0)
17 Downloads
Updated 23 Aug 2023
View License

Following are the steps to install NXP's Model-Based Design Toolbox for Z/E Series of MCUs:

1. Press the Open Installer Guide button in Getting Started live script or run NXP_Support_Package_S32ZE command, and follow the next quick steps as guided by the installer.

2.1 Create an account with NXP (toolbox is for free but account is requested to download the tool and access training & support on NXP Model Based Design Toolbox Community).

2.2. Download NXP Model-Based Design Toolbox for S32Z/E Series

2.3. Install the toolbox

2.4. Verify the correct installation

2.5 Generate a free of charge license

2.6. Activate the toolbox

2.7 Verify the license activation

The NXP's Model-Based Design Toolbox is a quick solution for testing and rapid prototyping applications on top of NXP MCUs.

It provides an integrated development environment and toolchain support for configuring and generating applications (including initialization routines and device drivers) to execute complex applications

Requires
- ✓ Simulink
- ✓ Embedded Coder
- ✓ MATLAB Coder
- ✓ Simulink Coder
- ⚠ Stateflow

MATLAB Release Compatibility

Created with R2020a
Compatible with R2020a and later releases

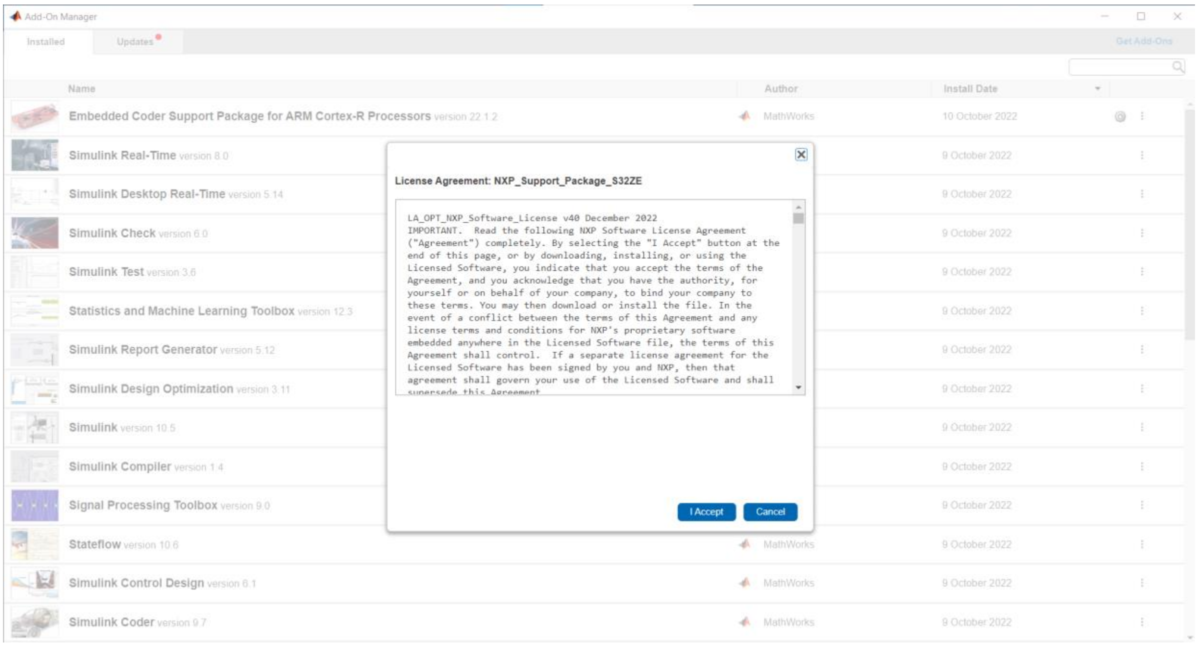
Platform Compatibility

☑ Windows □ macOS □ Linux

Tags

Others Also Downloaded

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



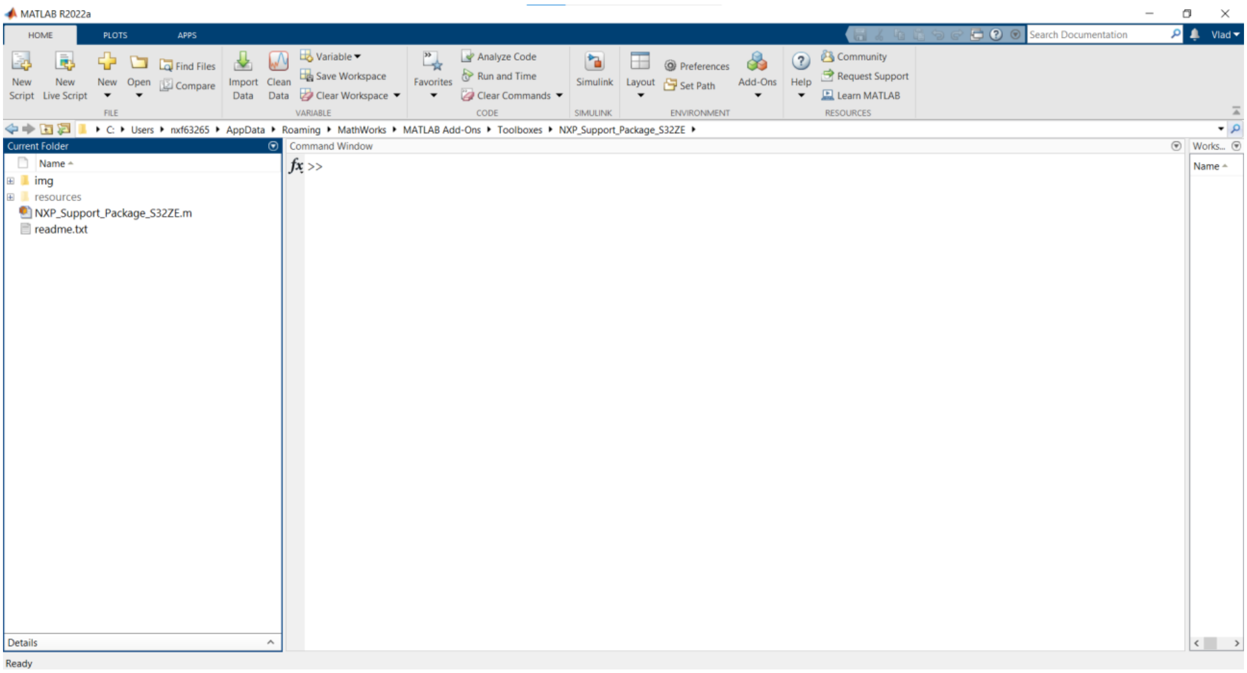
Name	Author	Install Date
Embedded Coder Support Package for ARM Cortex-R Processors version 22.1.2	MathWorks	10 October 2022
Simulink Real-Time version 8.0		9 October 2022
Simulink Desktop Real-Time version 5.14		9 October 2022
Simulink Check version 6.0		9 October 2022
Simulink Test version 3.6		9 October 2022
Statistics and Machine Learning Toolbox version 12.3		9 October 2022
Simulink Report Generator version 5.12		9 October 2022
Simulink Design Optimization version 3.11		9 October 2022
Simulink version 10.5		9 October 2022
Simulink Compiler version 1.4		9 October 2022
Signal Processing Toolbox version 9.0		9 October 2022
Stateflow version 10.6	MathWorks	9 October 2022
Simulink Control Design version 6.1	MathWorks	9 October 2022
Simulink Coder version 9.7	MathWorks	9 October 2022

License Agreement: NXP_Support_Package_S32ZE

```
LA_OPT_NXP_Software_License v40 December 2022
IMPORTANT.  Read the following NXP Software License Agreement
("Agreement") completely. By selecting the "I Accept" button at the
end of this page, or by downloading, installing, or using the
Licensed Software, you indicate that you accept the terms of the
Agreement, and you acknowledge that you have the authority, for
yourself or on behalf of your company, to bind your company to
these terms. You may then download or install the file. In the
event of a conflict between the terms of this Agreement and any
license terms and conditions for NXP's proprietary software
embedded anywhere in the Licensed Software file, the terms of this
Agreement shall control.  If a separate license agreement for the
Licensed Software has been signed by you and NXP, then that
agreement shall govern your use of the Licensed Software and shall
supersede this Agreement
```

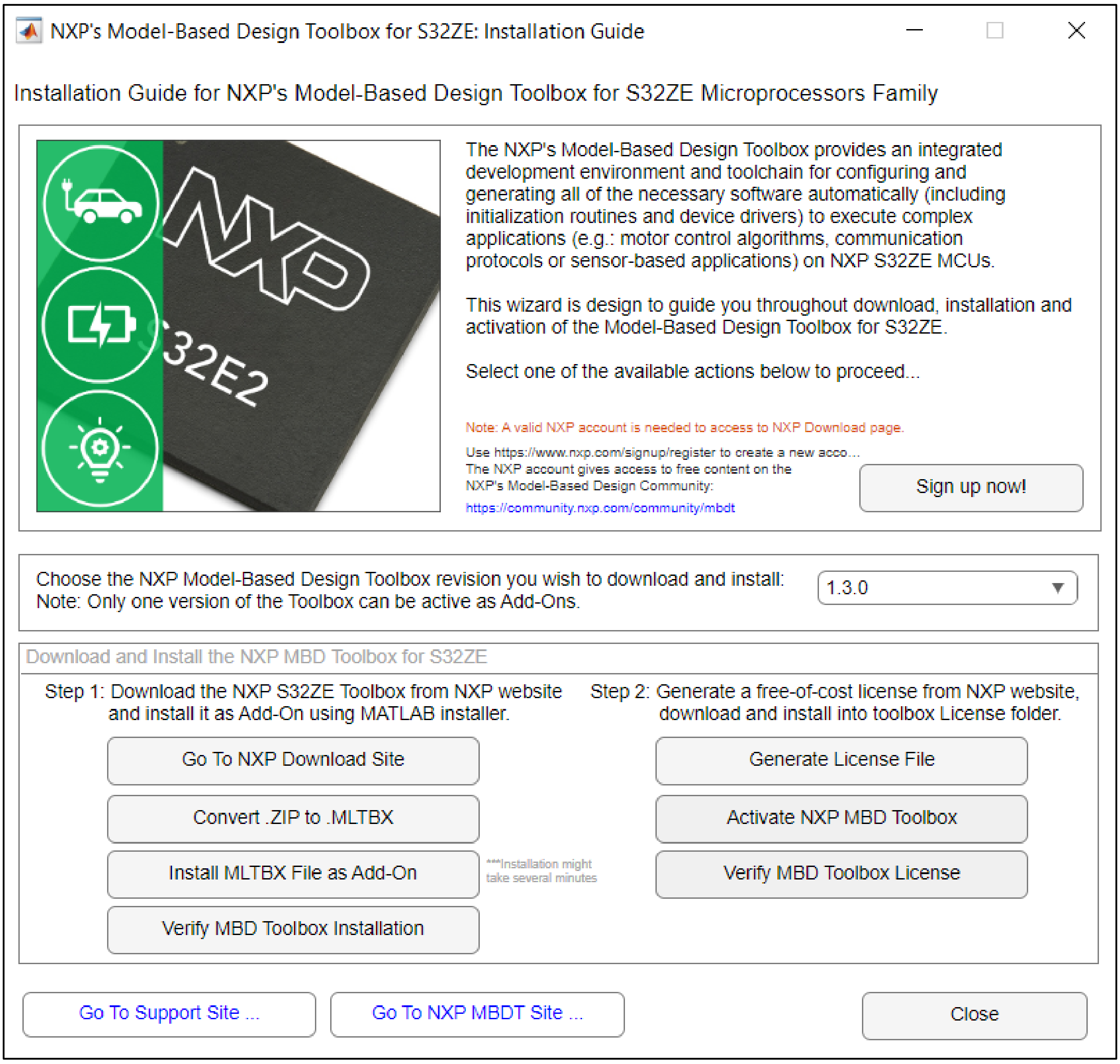
e. Once the process is successful, press the **Open Folder** button under the ellipsis menu.

f. Run the NXP_Support_Package_S32ZE.m script to start the NXP Support Package for S32ZE.



1.2.3 Install NXP Model-Based Design Toolbox for S32Z/E

NXP Support Package for S32Z/E is a graphical user interface guide that helps to download and install the Model-Based Design Toolbox, and also generate and install the license from the NXP website.



NXP's Model-Based Design Toolbox for S32ZE: Installation Guide

Installation Guide for NXP's Model-Based Design Toolbox for S32ZE Microprocessors Family

The NXP's Model-Based Design Toolbox provides an integrated development environment and toolchain for configuring and generating all of the necessary software automatically (including initialization routines and device drivers) to execute complex applications (e.g.: motor control algorithms, communication protocols or sensor-based applications) on NXP S32ZE MCUs.

This wizard is design to guide you throughout download, installation and activation of the Model-Based Design Toolbox for S32ZE.

Select one of the available actions below to proceed...

Note: A valid NXP account is needed to access to NXP Download page.

Use https://www.nxp.com/signup/register to create a new acco...
The NXP account gives access to free content on the NXP's Model-Based Design Community:
https://community.nxp.com/community/mbdt

Choose the NXP Model-Based Design Toolbox revision you wish to download and install:
Note: Only one version of the Toolbox can be active as Add-Ons.

1.3.0

Download and Install the NXP MBD Toolbox for S32ZE

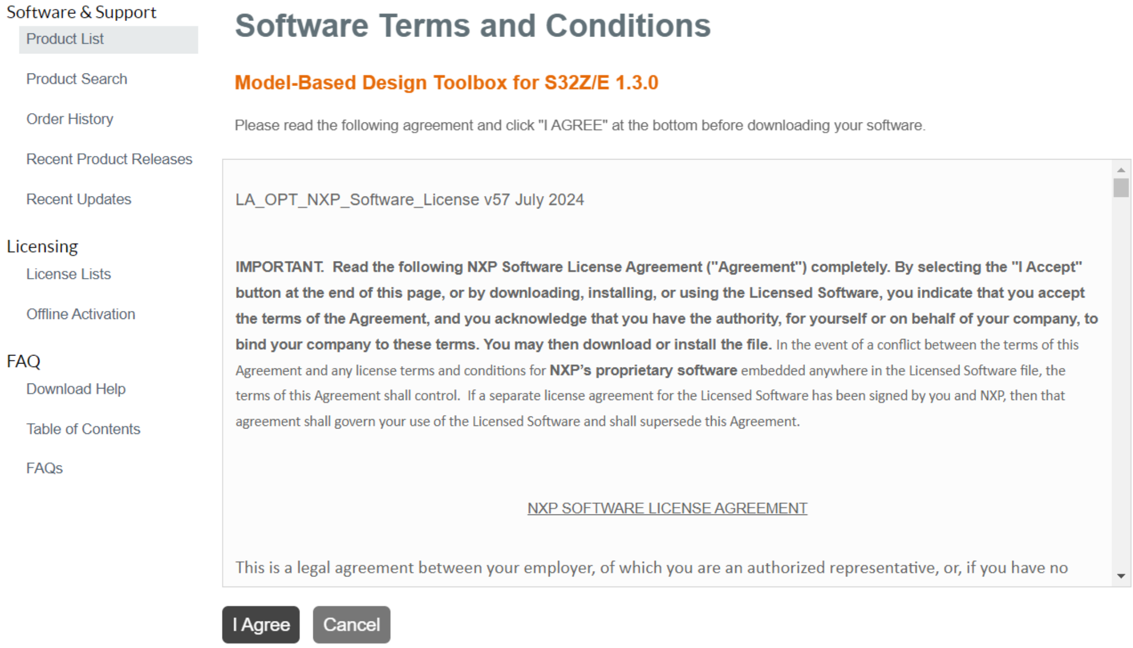
Step 1: Download the NXP S32ZE Toolbox from NXP website and install it as Add-On using MATLAB installer.

Step 2: Generate a free-of-cost license from NXP website, download and install into toolbox License folder.

***Installation might take several minutes

1. Press **Go to NXP Download Site** Button. In the newly opened window, Review the Terms and Conditions as you scroll down, and press **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 - S32Z/E Standard Software** -> **Automotive SW - S32Z/E - MBDT** and select the latest release available.



Software Terms and Conditions

Model-Based Design Toolbox for S32Z/E 1.3.0

Please read the following agreement and click "I AGREE" at the bottom before downloading your software.

LA_OPT_NXP_Software_License v57 July 2024

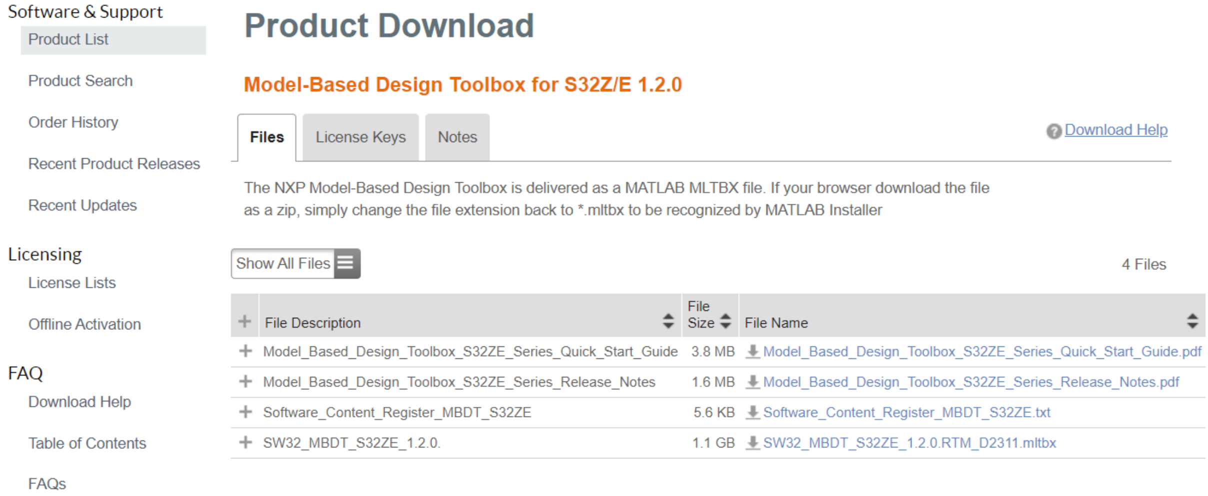
IMPORTANT. Read the following NXP Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, or by downloading, installing, or using the Licensed Software, you indicate that you accept the terms of the Agreement, and you acknowledge that you have the authority, for yourself or on behalf of your company, to bind your company to these terms. You may then download or install the file. In the event of a conflict between the terms of this Agreement and any license terms and conditions for **NXP's proprietary software** embedded anywhere in the Licensed Software file, the terms of this Agreement shall control. If a separate license agreement for the Licensed Software has been signed by you and NXP, then that agreement shall govern your use of the Licensed Software and shall supersede this Agreement.

NXP SOFTWARE LICENSE AGREEMENT

This is a legal agreement between your employer, of which you are an authorized representative, or, if you have no

2. Download the SW32_MBDT_S32ZE_1.3.0. mltbx file.

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



Product Download

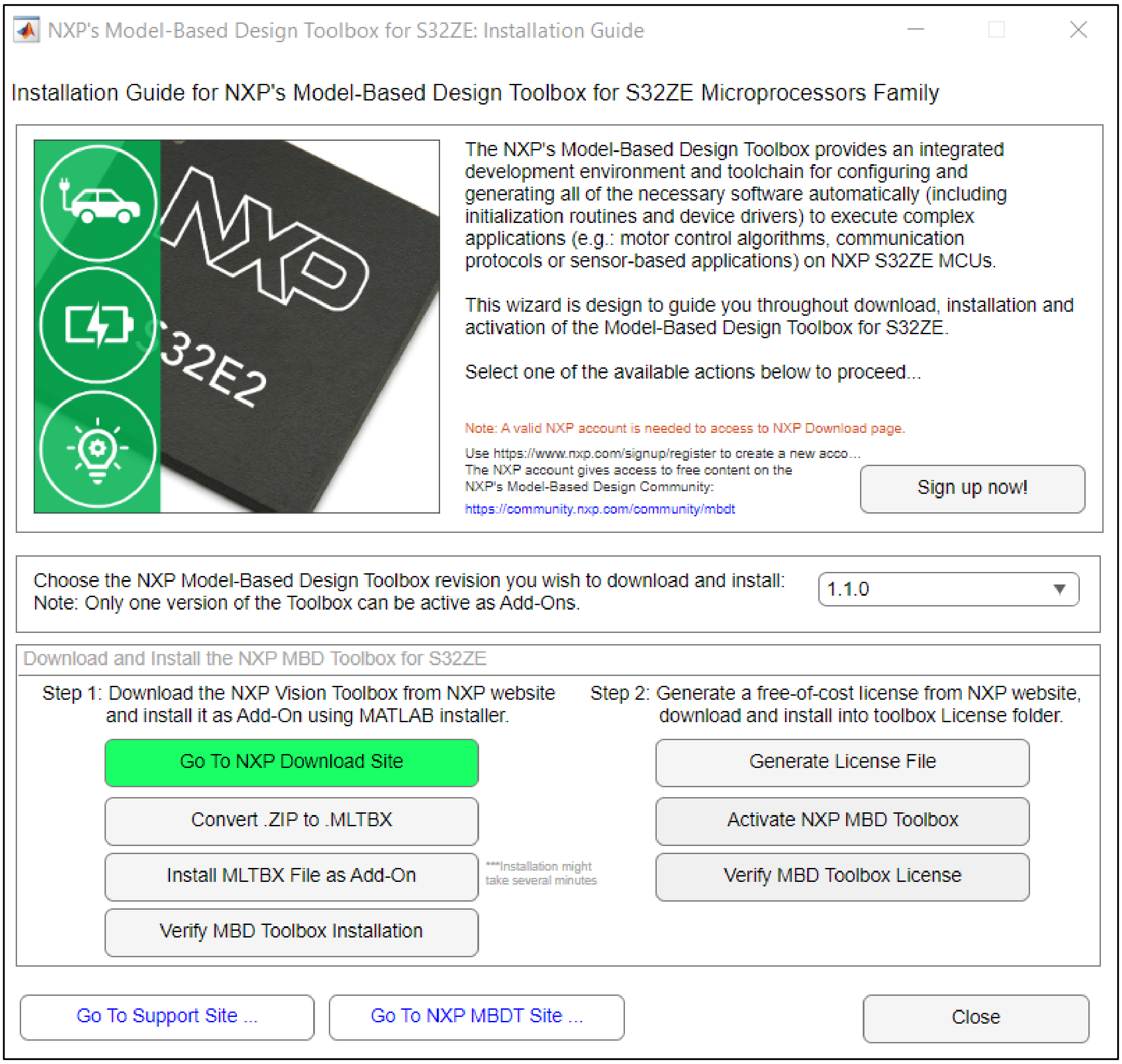
Model-Based Design Toolbox for S32Z/E 1.2.0

The NXP Model-Based Design Toolbox is delivered as a MATLAB MLTBX file. If your browser download the file as a zip, simply change the file extension back to *.mltbx to be recognized by MATLAB Installer

4 Files

File Description	File Size	File Name
Model_Based_Design_Toolbox_S32ZE_Series_Quick_Start_Guide	3.8 MB	Model_Based_Design_Toolbox_S32ZE_Series_Quick_Start_Guide.pdf
Model_Based_Design_Toolbox_S32ZE_Series_Release_Notes	1.6 MB	Model_Based_Design_Toolbox_S32ZE_Series_Release_Notes.pdf
Software_Content_Register_MBDT_S32ZE	5.6 KB	Software_Content_Register_MBDT_S32ZE.txt
SW32_MBDT_S32ZE_1.2.0.	1.1 GB	SW32_MBDT_S32ZE_1.2.0.RTM_D2311.mltbx

3. Go back to **NXP Support Package for S32ZE** and press the **Convert .ZIP to .MLTBX** button. In the newly opened Browsing window, select the file downloaded and press **Open.**



Installation Guide for NXP's Model-Based Design Toolbox for S32ZE Microprocessors Family

The NXP's Model-Based Design Toolbox provides an integrated development environment and toolchain for configuring and generating all of the necessary software automatically (including initialization routines and device drivers) to execute complex applications (e.g.: motor control algorithms, communication protocols or sensor-based applications) on NXP S32ZE MCUs.

This wizard is design to guide you throughout download, installation and activation of the Model-Based Design Toolbox for S32ZE.

Select one of the available actions below to proceed...

Note: A valid NXP account is needed to access to NXP Download page.

Use https://www.nxp.com/signup/register to create a new acco...
The NXP account gives access to free content on the NXP's Model-Based Design Community:
https://community.nxp.com/community/mbdt

Choose the NXP Model-Based Design Toolbox revision you wish to download and install:
Note: Only one version of the Toolbox can be active as Add-Ons.

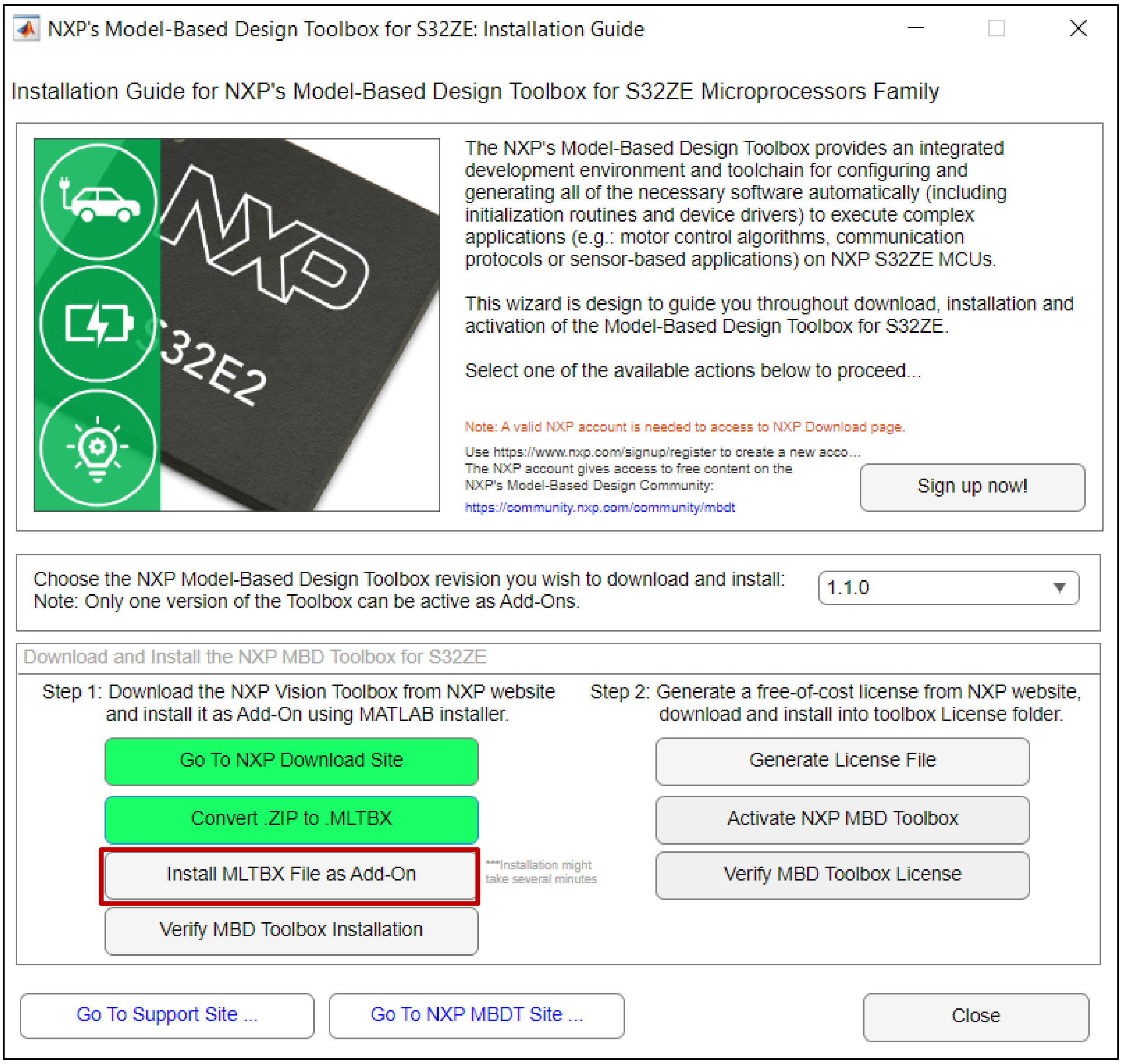
Download and Install the NXP MBD Toolbox for S32ZE

Step 1: Download the NXP Vision Toolbox from NXP website and install it as Add-On using MATLAB installer.

Step 2: Generate a free-of-cost license from NXP website, download and install into toolbox License folder.

***Installation might take several minutes

4. Go back to **NXP Support Package for S32ZE** and select the **Install MLTBX File as Add-On** button. In the newly opened window, browse for the MLTBX file and press **Open.**



Installation Guide for NXP's Model-Based Design Toolbox for S32ZE Microprocessors Family

The NXP's Model-Based Design Toolbox provides an integrated development environment and toolchain for configuring and generating all of the necessary software automatically (including initialization routines and device drivers) to execute complex applications (e.g.: motor control algorithms, communication protocols or sensor-based applications) on NXP S32ZE MCUs.

This wizard is design to guide you throughout download, installation and activation of the Model-Based Design Toolbox for S32ZE.

Select one of the available actions below to proceed...

Note: A valid NXP account is needed to access to NXP Download page.

Use https://www.nxp.com/signup/register to create a new acco...
The NXP account gives access to free content on the NXP's Model-Based Design Community:
https://community.nxp.com/community/mbdt

Choose the NXP Model-Based Design Toolbox revision you wish to download and install: 1.1.0
Note: Only one version of the Toolbox can be active as Add-Ons.

Download and Install the NXP MBD Toolbox for S32ZE

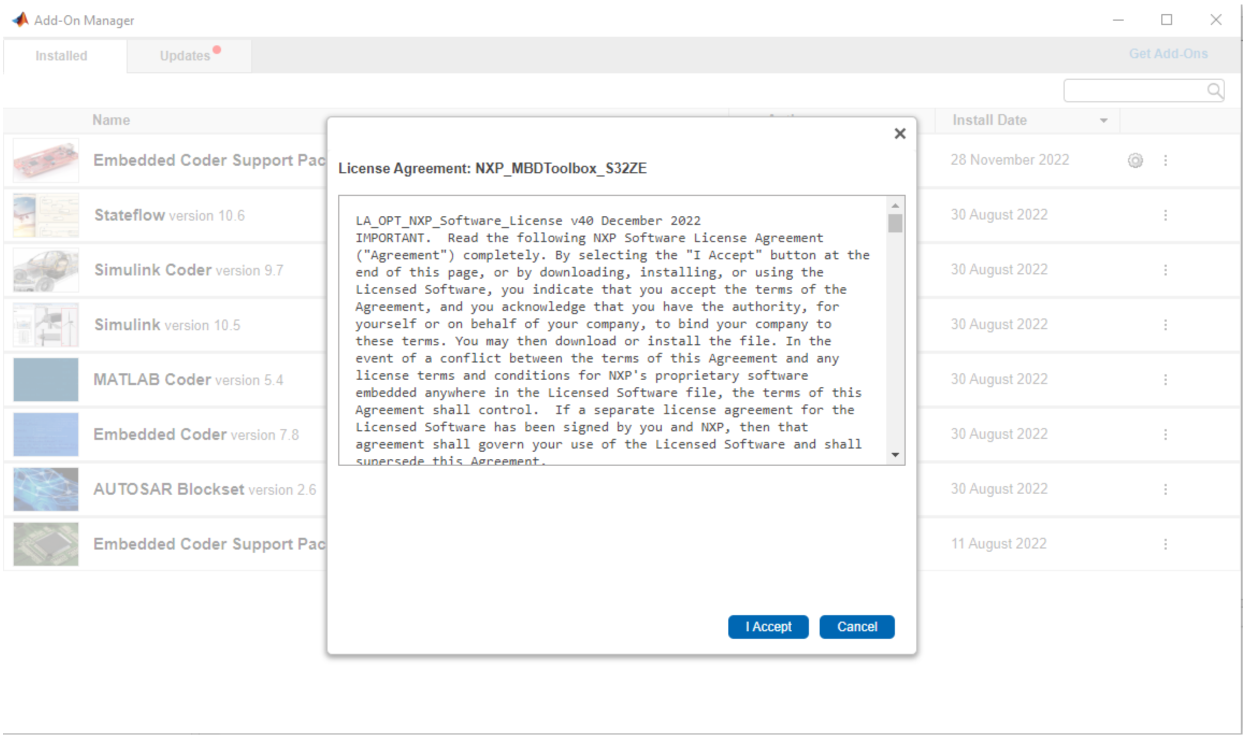
Step 1: Download the NXP Vision Toolbox from NXP website and install it as Add-On using MATLAB installer.

Step 2: Generate a free-of-cost license from NXP website, download and install into toolbox License folder.

***Installation might take several minutes

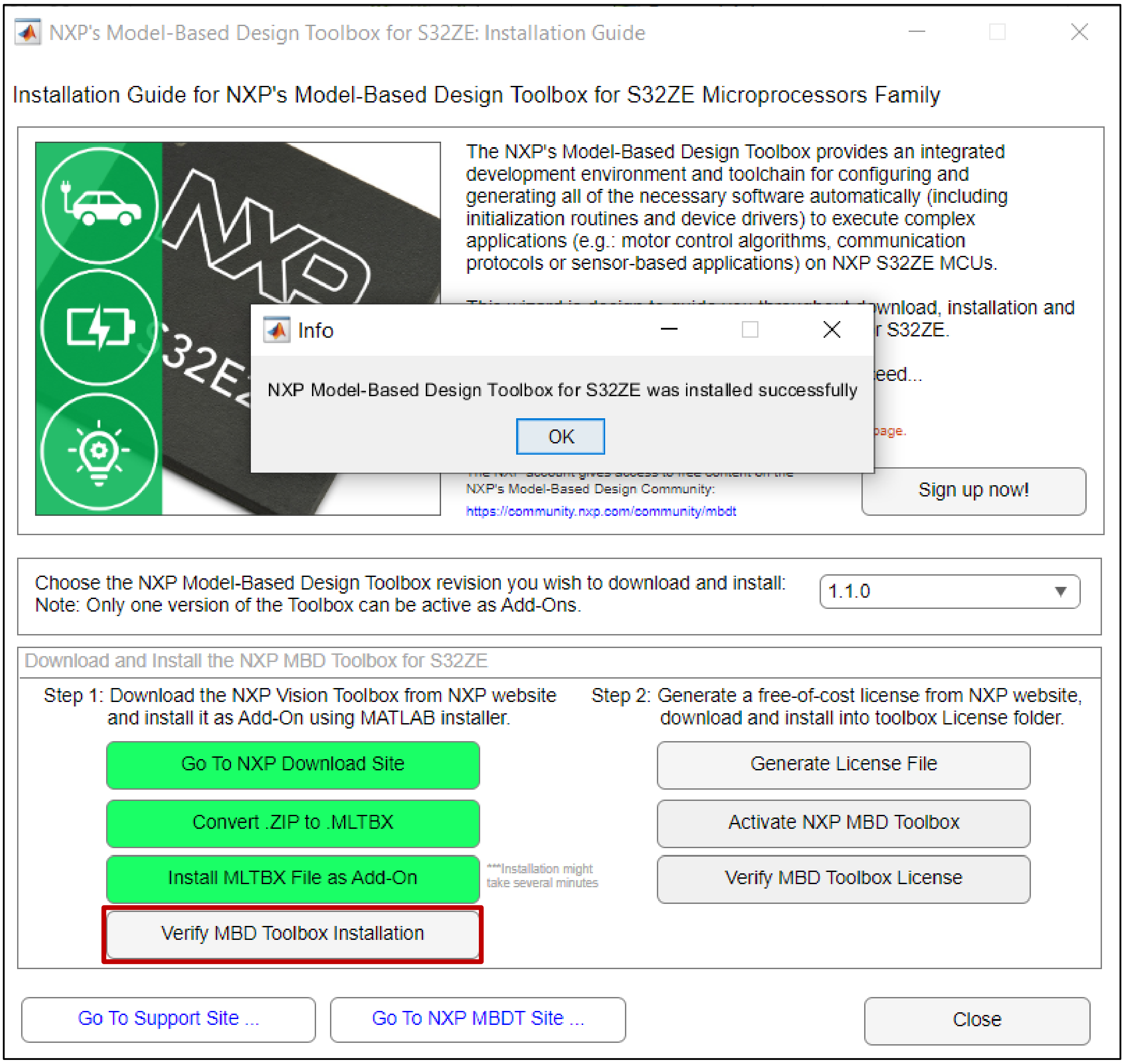
5. In the MATLAB Add-On Manager, Review the Terms and Conditions as you scroll down, and press **I Accept** Button. This action starts MBDT for S32Z/E Toolbox installation process.

   Note: Installation might take several minutes.



```
LA_OPT_NXP_Software_License v40 December 2022
IMPORTANT.  Read the following NXP Software License Agreement
("Agreement") completely. By selecting the "I Accept" button at the
end of this page, or by downloading, installing, or using the
Licensed Software, you indicate that you accept the terms of the
Agreement, and you acknowledge that you have the authority, for
yourself or on behalf of your company, to bind your company to
these terms. You may then download or install the file. In the
event of a conflict between the terms of this Agreement and any
license terms and conditions for NXP's proprietary software
embedded anywhere in the Licensed Software file, the terms of this
Agreement shall control.  If a separate license agreement for the
Licensed Software has been signed by you and NXP, then that
agreement shall govern your use of the Licensed Software and shall
supersede this Agreement.
```

6. Once the installation is complete, go back to **NXP Support Package for S32ZE** and press the **Verify MBD Toolbox Installation button**.



NXP's Model-Based Design Toolbox for S32ZE: Installation Guide

Installation Guide for NXP's Model-Based Design Toolbox for S32ZE Microprocessors Family

The NXP's Model-Based Design Toolbox provides an integrated development environment and toolchain for configuring and generating all of the necessary software automatically (including initialization routines and device drivers) to execute complex applications (e.g.: motor control algorithms, communication protocols or sensor-based applications) on NXP S32ZE MCUs.

wnload, installation and r S32ZE.

eed...

page.

The NXP account gives access to free content on the NXP's Model-Based Design Community:
https://community.nxp.com/community/mbdt

Info

NXP Model-Based Design Toolbox for S32ZE was installed successfully

Choose the NXP Model-Based Design Toolbox revision you wish to download and install:
Note: Only one version of the Toolbox can be active as Add-Ons.

1.1.0

Download and Install the NXP MBD Toolbox for S32ZE

Step 1: Download the NXP Vision Toolbox from NXP website and install it as Add-On using MATLAB installer.

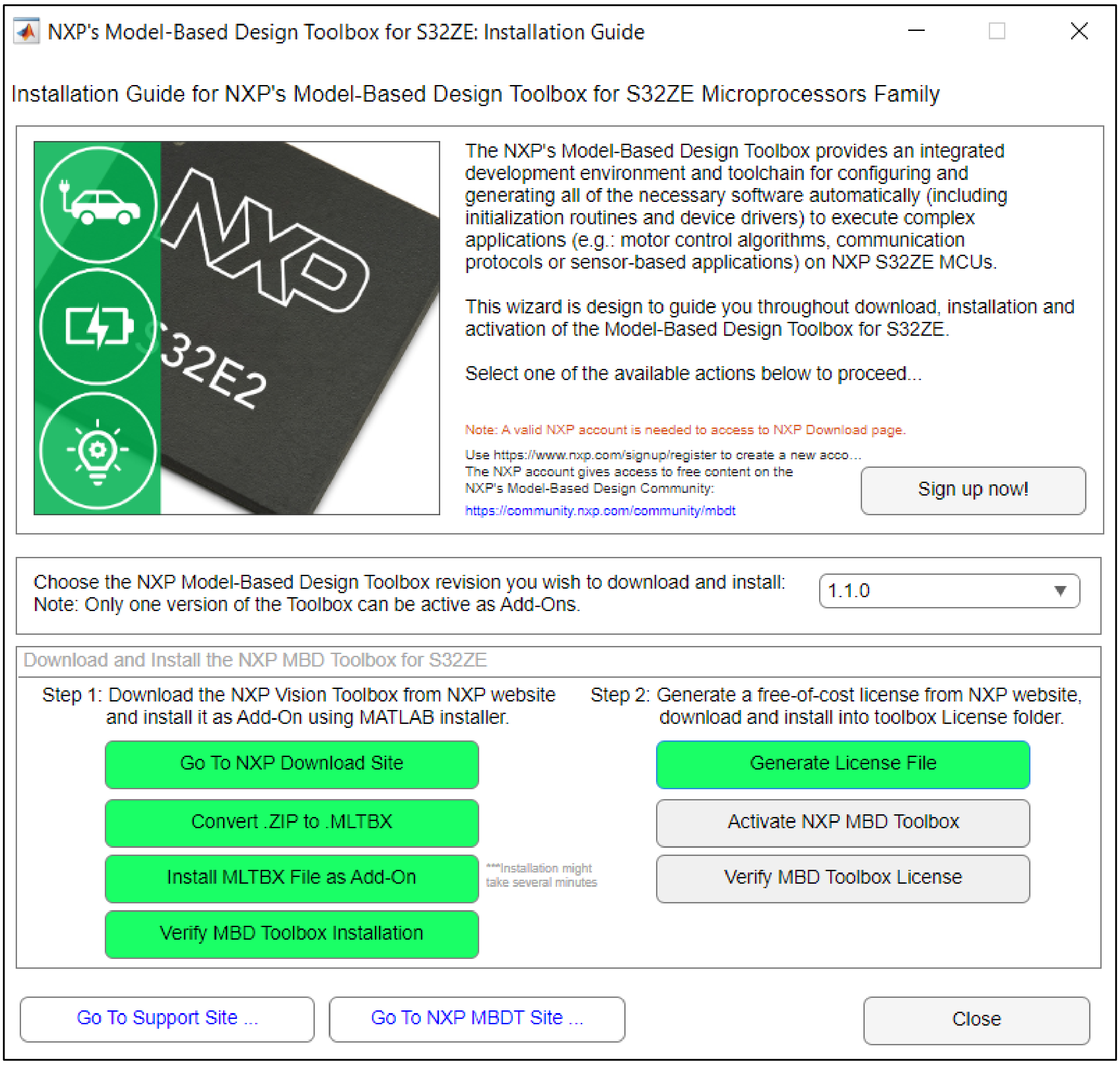
Step 2: Generate a free-of-cost license from NXP website, download and install into toolbox License folder.

***Installation might take several minutes

1.2.4 Generate and Activate NXP Model-Based Design Toolbox for S32Z/E license

The following steps guides you on how to achieve the license for S32Z/E Toolbox.

1. Press the Generate License File in the **NXP Support Package for S32ZE.**



NXP's Model-Based Design Toolbox for S32ZE: Installation Guide

Installation Guide for NXP's Model-Based Design Toolbox for S32ZE Microprocessors Family

The NXP's Model-Based Design Toolbox provides an integrated development environment and toolchain for configuring and generating all of the necessary software automatically (including initialization routines and device drivers) to execute complex applications (e.g.: motor control algorithms, communication protocols or sensor-based applications) on NXP S32ZE MCUs.

This wizard is design to guide you throughout download, installation and activation of the Model-Based Design Toolbox for S32ZE.

Select one of the available actions below to proceed...

Note: A valid NXP account is needed to access to NXP Download page.

Use https://www.nxp.com/signup/register to create a new acco...
The NXP account gives access to free content on the
NXP's Model-Based Design Community:
https://community.nxp.com/community/mbdt

Choose the NXP Model-Based Design Toolbox revision you wish to download and install: 1.1.0
Note: Only one version of the Toolbox can be active as Add-Ons.

Download and Install the NXP MBD Toolbox for S32ZE

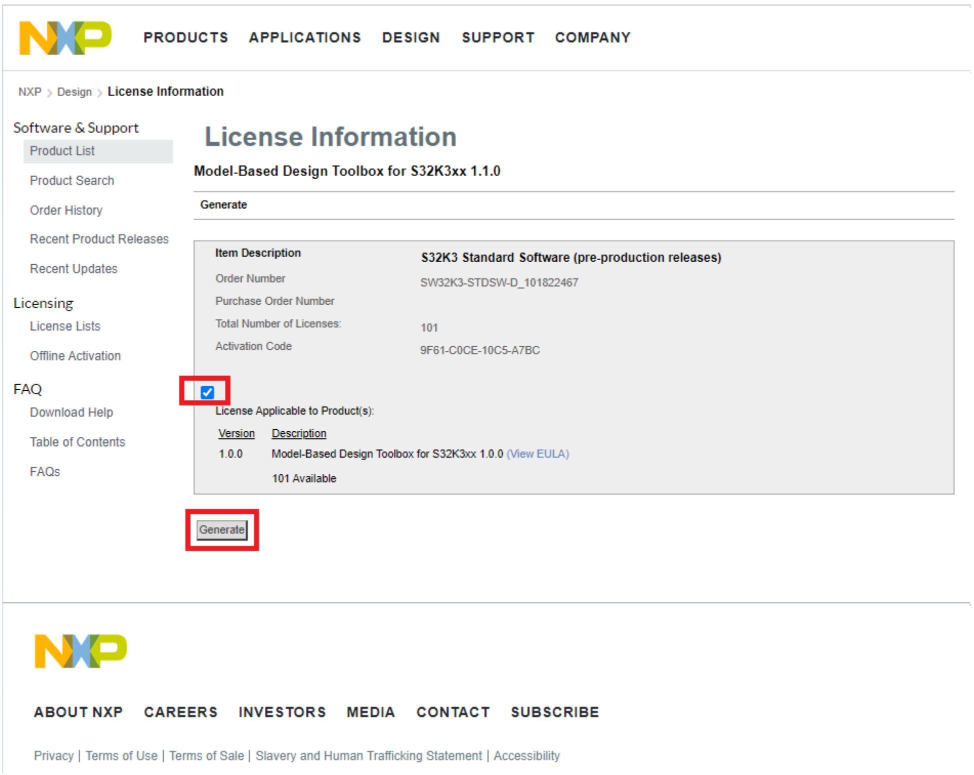
Step 1: Download the NXP Vision Toolbox from NXP website and install it as Add-On using MATLAB installer.

Step 2: Generate a free-of-cost license from NXP website, download and install into toolbox License folder.

***Installation might take several minutes

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, under the next tab “License keys”.



License Information

Model-Based Design Toolbox for S32K3xx 1.1.0

Generate

Item Description	S32K3 Standard Software (pre-production releases)
Order Number	SW32K3-STDSW-D_101822467
Purchase Order Number	
Total Number of Licenses:	101
Activation Code	9F61-C0CE-10C5-A7BC

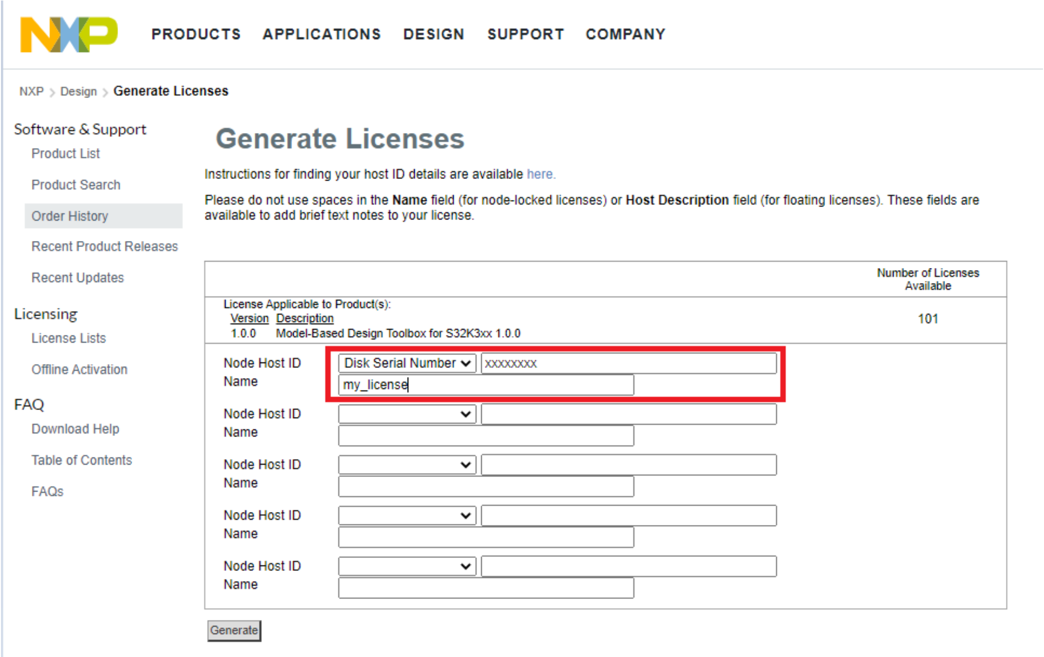
☑

License Applicable to Product(s):

Version	Description
1.0.0	Model-Based Design Toolbox for S32K3xx 1.0.0 (View EULA) 101 Available

Generate
ABOUT NXP
CAREERS
INVESTORS
MEDIA
CONTACT
SUBSCRIBE
Privacy | Terms of Use | Terms of Sale | Slavery and Human Trafficking Statement | Accessibility

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



Generate Licenses

Instructions for finding your host ID details are available here.

Please do not use spaces in the **Name** field (for node-locked licenses) or **Host Description** field (for floating licenses). These fields are available to add brief text notes to your license.

	Number of Licenses Available
License Applicable to Product(s): Version Description 1.0.0 Model-Based Design Toolbox for S32K3xx 1.0.0	101

Node Host ID: Disk Serial Number xxxxxxxx
Name: my_license

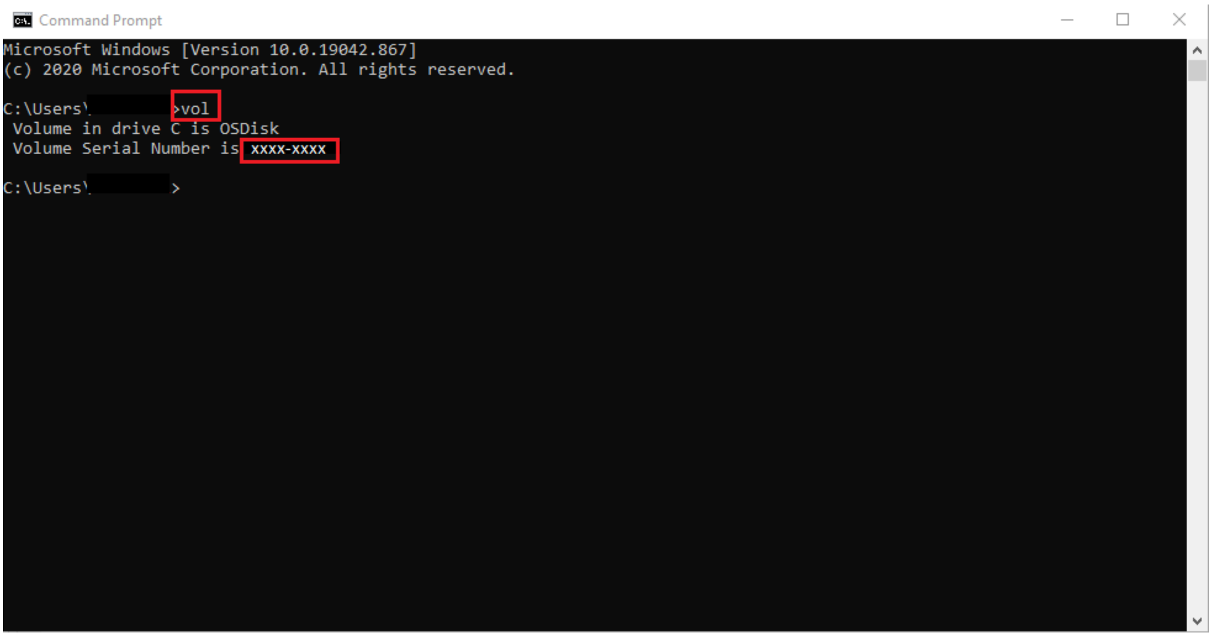
Node Host ID
Name

Node Host ID
Name

Node Host ID
Name

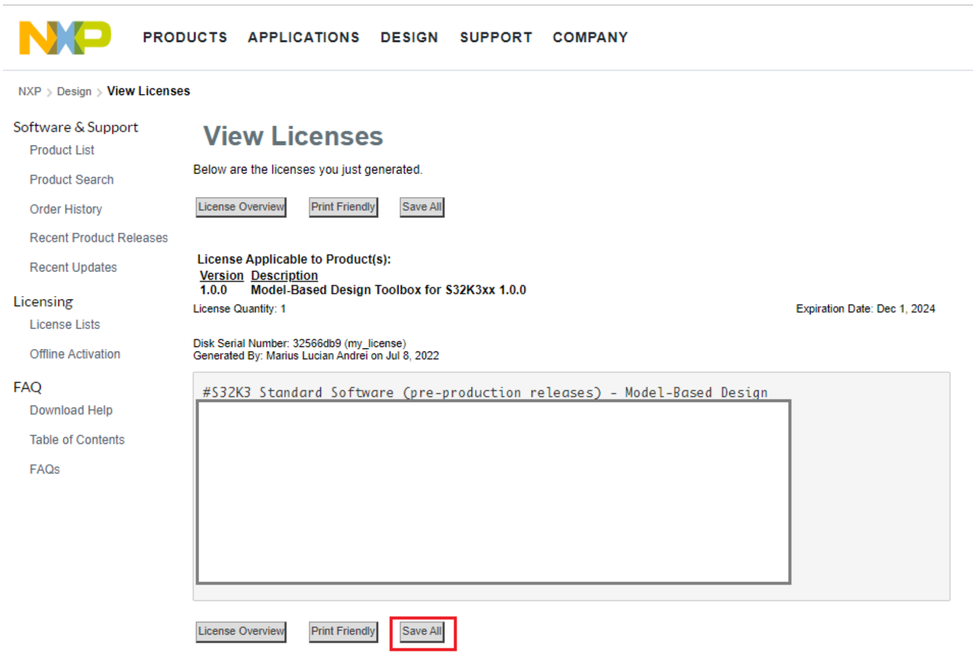
Node Host ID
Name

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



```
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\[redacted]>vol
 Volume in drive C is OSDisk
 Volume Serial Number is xxxx-xxxx

C:\Users\[redacted]>
```

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



View Licenses

Below are the licenses you just generated.

License Applicable to Product(s):

Version	Description
1.0.0	Model-Based Design Toolbox for S32K3xx 1.0.0

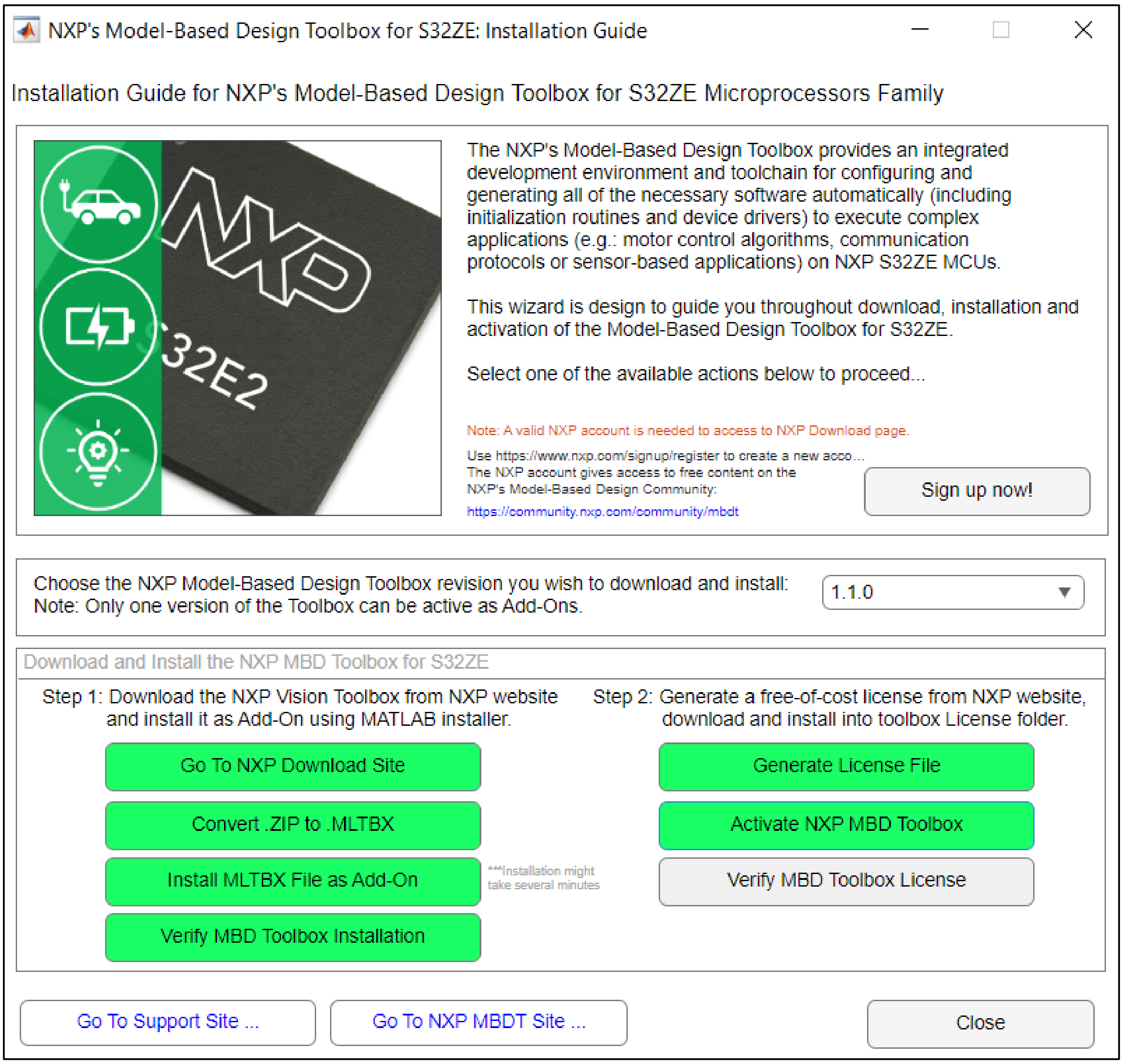
License Quantity: 1

Expiration Date: Dec 1, 2024

Disk Serial Number: 32566db9 (my_license)
Generated By: Marius Lucian Andrei on Jul 8, 2022

```
#S32K3 Standard Software (pre-production releases) - Model-Based Design
```

6. Back to **NXP Support Package for S32ZE,** 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.



NXP's Model-Based Design Toolbox for S32ZE: Installation Guide

Installation Guide for NXP's Model-Based Design Toolbox for S32ZE Microprocessors Family

The NXP's Model-Based Design Toolbox provides an integrated development environment and toolchain for configuring and generating all of the necessary software automatically (including initialization routines and device drivers) to execute complex applications (e.g.: motor control algorithms, communication protocols or sensor-based applications) on NXP S32ZE MCUs.

This wizard is design to guide you throughout download, installation and activation of the Model-Based Design Toolbox for S32ZE.

Select one of the available actions below to proceed...

Note: A valid NXP account is needed to access to NXP Download page.

Use https://www.nxp.com/signup/register to create a new acco... The NXP account gives access to free content on the NXP's Model-Based Design Community: https://community.nxp.com/community/mbdt

Choose the NXP Model-Based Design Toolbox revision you wish to download and install: 1.1.0
Note: Only one version of the Toolbox can be active as Add-Ons.

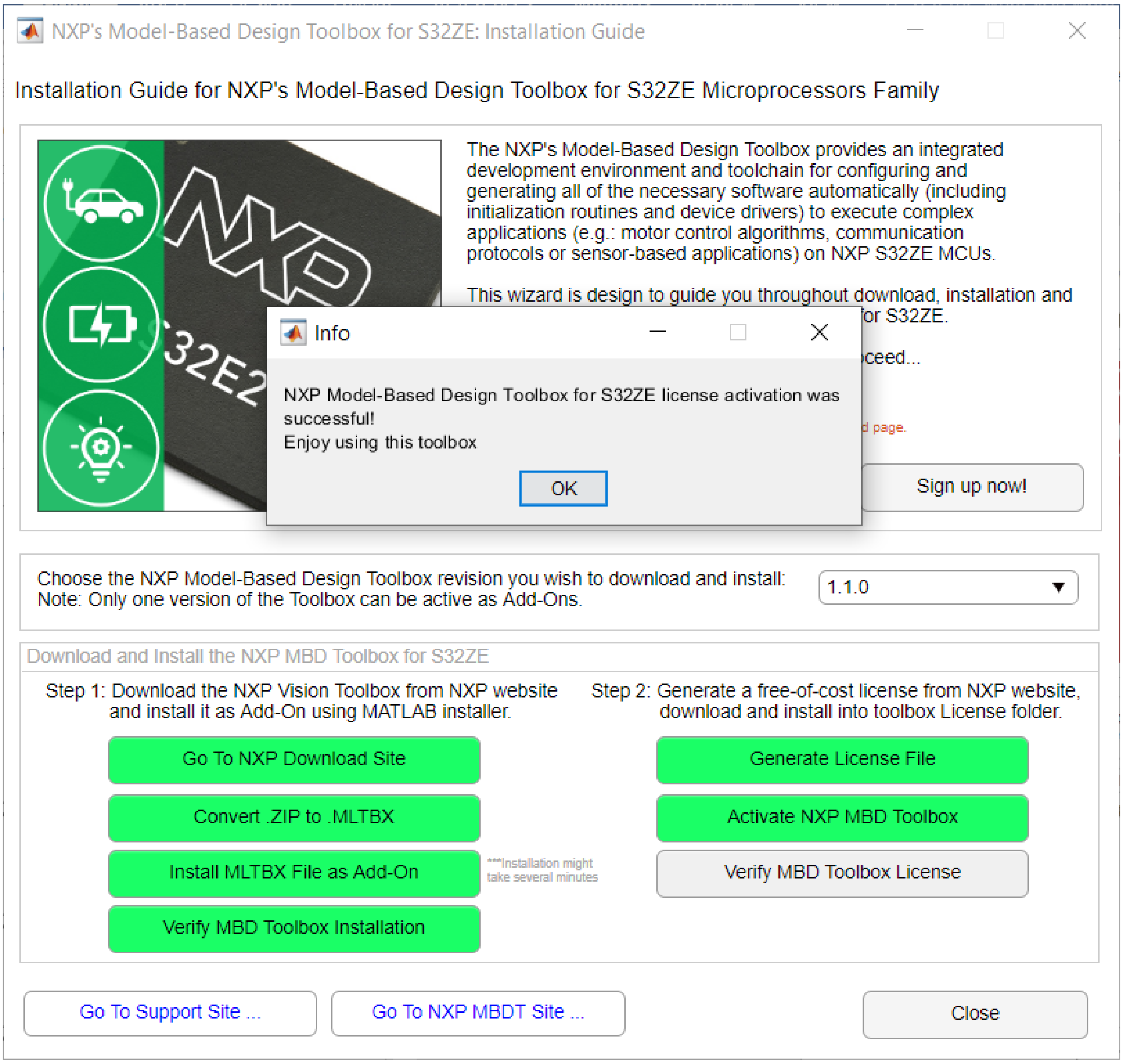
Download and Install the NXP MBD Toolbox for S32ZE

Step 1: Download the NXP Vision Toolbox from NXP website and install it as Add-On using MATLAB installer.

Step 2: Generate a free-of-cost license from NXP website, download and install into toolbox License folder.

***Installation might take several minutes

7. Final 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.



NXP's Model-Based Design Toolbox for S32ZE: Installation Guide

Installation Guide for NXP's Model-Based Design Toolbox for S32ZE Microprocessors Family

The NXP's Model-Based Design Toolbox provides an integrated development environment and toolchain for configuring and generating all of the necessary software automatically (including initialization routines and device drivers) to execute complex applications (e.g.: motor control algorithms, communication protocols or sensor-based applications) on NXP S32ZE MCUs.

This wizard is design to guide you throughout download, installation and
or S32ZE.

ceed...

d page.

Info

NXP Model-Based Design Toolbox for S32ZE license activation was successful!
Enjoy using this toolbox

Choose the NXP Model-Based Design Toolbox revision you wish to download and install:
Note: Only one version of the Toolbox can be active as Add-Ons.

1.1.0

Download and Install the NXP MBD Toolbox for S32ZE

Step 1: Download the NXP Vision Toolbox from NXP website and install it as Add-On using MATLAB installer.

Step 2: Generate a free-of-cost license from NXP website, download and install into toolbox License folder.

***Installation might take several minutes

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

The Model-Based Design Toolbox for S32Z/E platform uses external tools to set up the Simulators and Compilers for ARM R52 core and CEVA SPF2 core. By default, the toolchain is configured to use the default installation paths for these tools. In order to set these paths, the user needs to edit a mlx-script with his/her installation paths for the external dependencies.

This is done by changing the MATLAB Current Directory to the toolbox installation directory (e.g.: `..\MATLAB\Add-Ons\Toolboxes\NXP_MBDToolbox_S32ZE\`) and edit the "`mbd_s32ze_dependencies_path.mlx`" script, changing the variables:

- "`gccCompilerPath`" for the desired GCC compiler path; default is the one provided within the toolbox.
- "`cevaToolboxPath`" for the CEVA toolchain path.
- "`spf2cePath`" for SPF2 Code Enablement package path.
- "`ipcfPath`" for IPCF package path.
- "`rtdPath`" for the RTD path; default is the one provided within the toolbox.

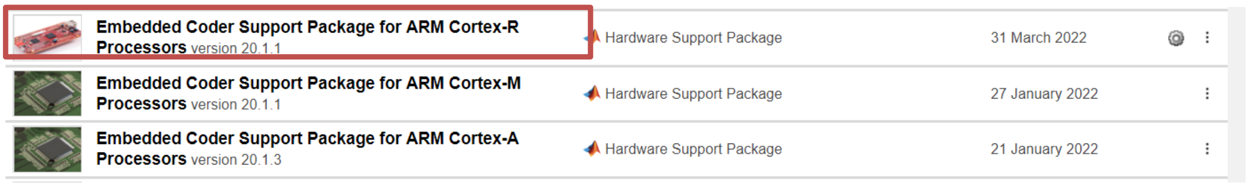
After the script was edited accordingly, it must be run for the changes to be applied.

The Model-Based Design Toolbox uses the Toolchain mechanism exposed by the MATLAB/Simulink to enable automatic code generation with Embedded Coder toolbox. By default, the toolchain is configured for the MATLAB 2022a release. For any other MATLAB release, the user needs to execute a toolbox m-script to generate the appropriate 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_S32ZE\`) and running the "`mbd_s32ze_path.m`" script.

```
>> mbd_s32ze_path
Treating '...\S32ZE\src' as MBD Toolbox installation root.
MBD Toolbox path prepended.
NXP S32 Design Studio GCC (S32ZE) toolchain is already registered ...
NXP CEVA Toolbox LLVM (S32ZE) toolchain is already registered ...
Registering the R52 coder-target ...
No compatible target currently available for NXP S32ZE2xx. Creating
one...
Creating folders for the target 'NXP S32ZE2xx' in the folder
'...\S32ZE\src\mbdtbx_s32ze\codertarget\'...
Creating the framework for the target 'NXP S32ZE2xx'...
Registering the target 'NXP S32ZE2xx'...
Done.
Successful.
```

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



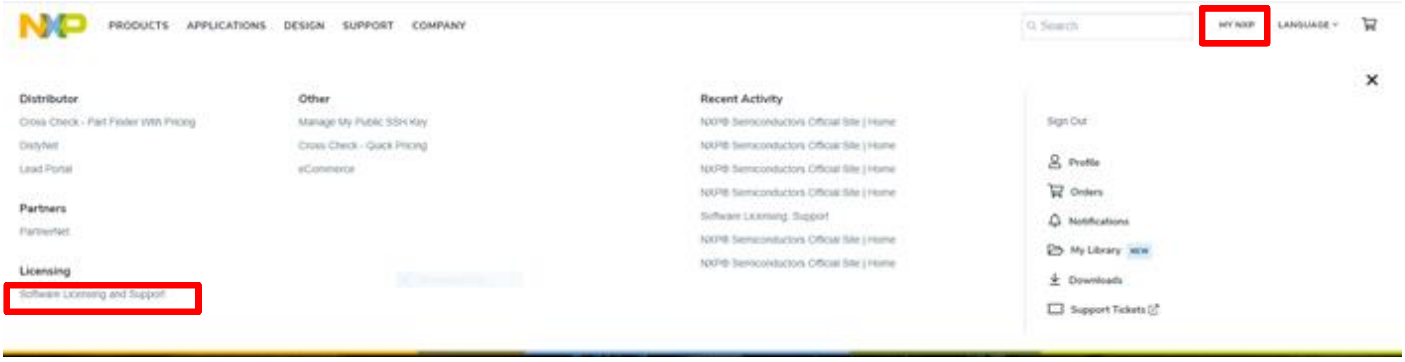
Embedded Coder Support Package for ARM Cortex-R Processors version 20.1.1	Hardware Support Package	31 March 2022	⋮
Embedded Coder Support Package for ARM Cortex-M Processors version 20.1.1	Hardware Support Package	27 January 2022	⋮
Embedded Coder Support Package for ARM Cortex-A Processors version 20.1.3	Hardware Support Package	21 January 2022	⋮

The “`mbd_s32ze_path.m`” script verifies the user setup dependencies and will issue instructions for a successful installation and configuration of the toolbox.

1.2.6 Installing EB Tresos (optional step)

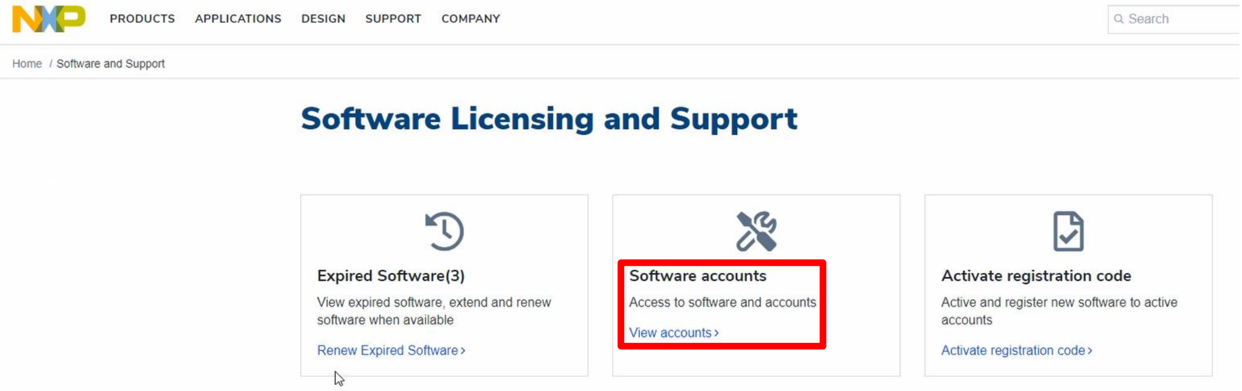
Model-based Design Toolbox for S32Z/E provides support for 1 external configuration tool – EB Tresos. To install this product, you will need to follow these instructions:

Go to the nxp.com website, log into your account (or make one for free). Then go to My NXP -> Licensing -> Software Licensing and Support:



PRODUCTS
APPLICATIONS
DESIGN
SUPPORT
COMPANY
Search
MY NXP
LANGUAGE
Distributor
Cross Check - Part Finder With Pricing
DistyNet
Lead Portal
Partners
PartnerNet
Licensing
Software Licensing and Support
Other
Manage My Public SSH Key
Cross Check - Quick Pricing
eCommerce
Recent Activity
NXP® Semiconductors Official Site | Home
NXP® Semiconductors Official Site | Home
NXP® Semiconductors Official Site | Home
NXP® Semiconductors Official Site | Home
Software Licensing Support
NXP® Semiconductors Official Site | Home
NXP® Semiconductors Official Site | Home
Sign Out
Profile
Orders
Notifications
My Library
NEW
Downloads
Support Tickets

For the next step go to Software accounts -> View accounts



Software Licensing and Support

Expired Software(3)

View expired software, extend and renew software when available

Renew Expired Software ›

Software accounts

Access to software and accounts

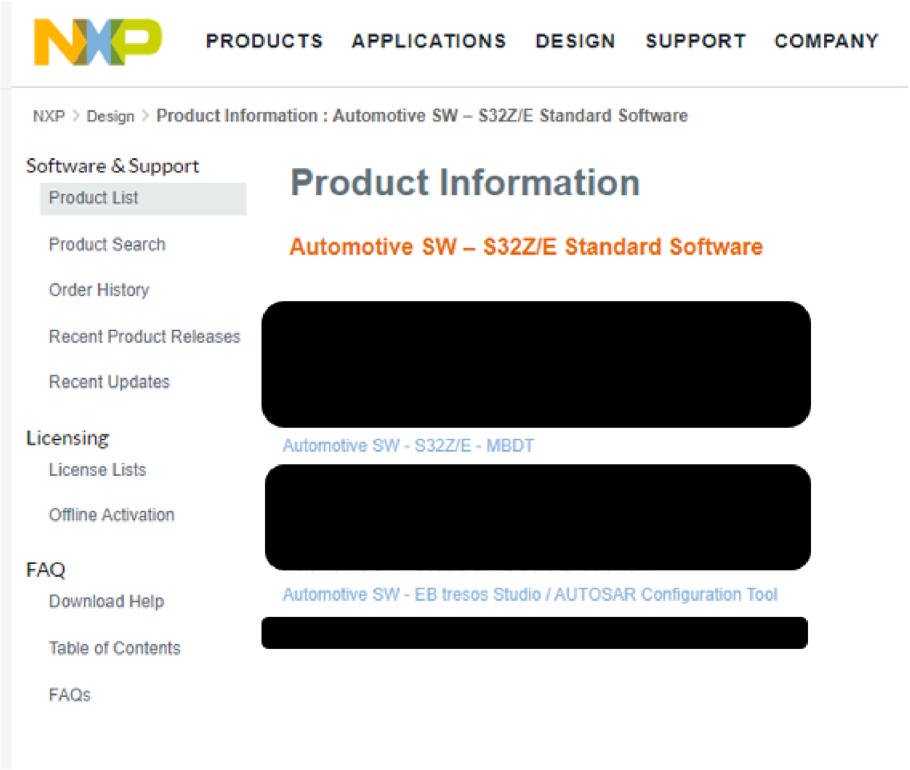
View accounts ›

Activate registration code

Active and register new software to active accounts

Activate registration code ›

You will find a Product list on the next page from which you should select **Automotive SW – S32Z/E Standard Software.** From the Product Information page, you should find the item names **Automotive SW – EB tresos Studio / AUTOSAR Configuration Tool**.



NXP > Design > Product Information : Automotive SW – S32Z/E Standard Software

Product Information

Automotive SW – S32Z/E Standard Software

[redacted]

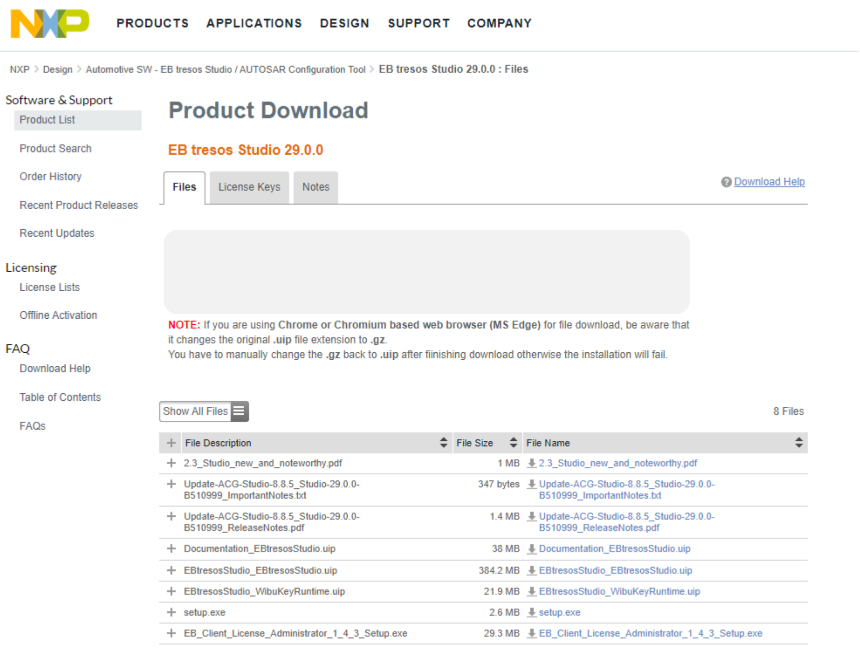
Automotive SW - S32Z/E - MBDT

[redacted]

Automotive SW - EB tresos Studio / AUTOSAR Configuration Tool

[redacted]

From the next page, select **EB tresos Studio 29.0.0**. Read the Software Terms and Conditions on the following page and click on **I Agree**.



Product Download

EB tresos Studio 29.0.0

NOTE: If you are using **Chrome or Chromium based web browser (MS Edge)** for file download, be aware that it changes the original **.uip** file extension to **.gz**.
You have to manually change the **.gz** back to **.uip** after fiinishing download otherwise the installation will fail.

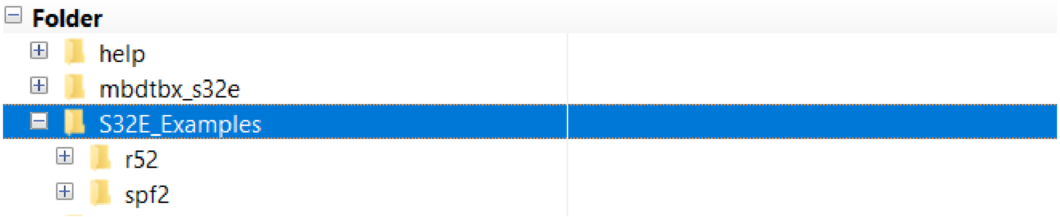
File Description	File Size	File Name
2.3_Studio_new_and_noteworthy.pdf	1 MB	2.3_Studio_new_and_noteworthy.pdf
Update-ACG-Studio-8.8.5_Studio-29.0.0-B510999_ImportantNotes.txt	347 bytes	Update-ACG-Studio-8.8.5_Studio-29.0.0-B510999_ImportantNotes.txt
Update-ACG-Studio-8.8.5_Studio-29.0.0-B510999_ReleaseNotes.pdf	1.4 MB	Update-ACG-Studio-8.8.5_Studio-29.0.0-B510999_ReleaseNotes.pdf
Documentation_EBtresosStudio.uip	38 MB	Documentation_EBtresosStudio.uip
EBtresosStudio_EBtresosStudio.uip	384.2 MB	EBtresosStudio_EBtresosStudio.uip
EBtresosStudio_WibuKeyRuntime.uip	21.9 MB	EBtresosStudio_WibuKeyRuntime.uip
setup.exe	2.6 MB	setup.exe
EB_Client_License_Administrator_1_4_3_Setup.exe	29.3 MB	EB_Client_License_Administrator_1_4_3_Setup.exe

From this page, download (minimum) the files shown in the picture above. Note that when the download is complete, the .uip files might have been renamed to .gz or .zip. You will have to manually change the extensions of those files back to .uip, then simply run the **setup.exe** file. From that step, the EB Tresos wizard will guide you through the installation. Additionally, you can download from the same location (shown in the picture above) the EB Tresos installation guide (available in .pdf format). Note that you will be required a license for this configuration tool, but it is made available on that same page (highlighted in yellow), for free.

2 Run Models

2.1 MATLAB

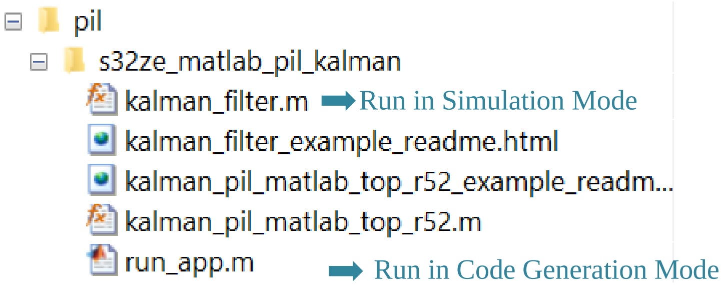
NXP's Model-Based Design Toolbox provides the MATLAB code generation/simulation capabilities to run the examples for both R52 and SPF2 cores. The examples are separated in two folders, one for R52 and one for SPF2.



Each category contains examples that can be run on the Hardware Target and for the SPF2 examples there is also the option to be run on the CEVA SensPro Simulator.

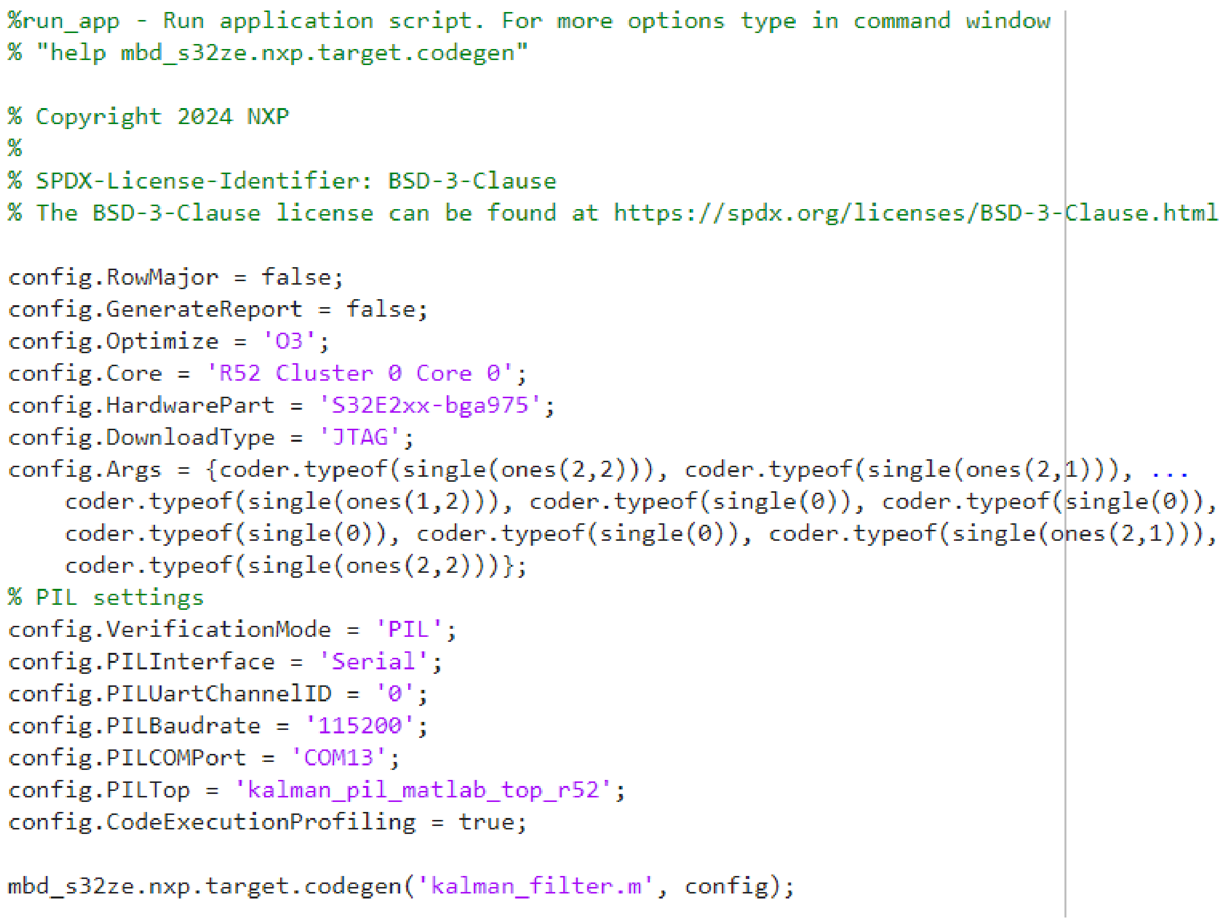
2.1.1 Run R52 Examples

In order to run the R52 examples, the user must go into the example directory and run the run_app.m script and this will start the code generation process. The example can be run also in Simulation mode by running the script with the example name.



pil
s32ze_matlab_pil_kalman
kalman_filter.m
Run in Simulation Mode
kalman_filter_example_readme.html
kalman_pil_matlab_top_r52_example_readm...
kalman_pil_matlab_top_r52.m
run_app.m
Run in Code Generation Mode

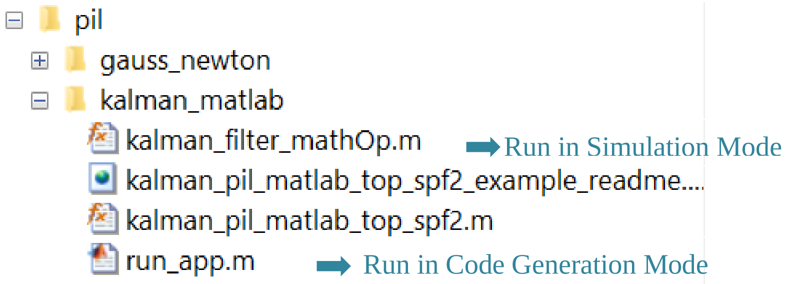
In the run_app script the user can select different options, like target name (R52 Cluste *x* Core *y*), optimization level, download type, etc.



```
%run_app - Run application script. For more options type in command window
% "help mbd_s32ze.nxp.target.codegen"

% Copyright 2024 NXP
%
% SPDX-License-Identifier: BSD-3-Clause
% The BSD-3-Clause license can be found at https://spdx.org/licenses/BSD-3-Clause.html

config.RowMajor = false;
config.GenerateReport = false;
config.Optimize = 'O3';
config.Core = 'R52 Cluster 0 Core 0';
config.HardwarePart = 'S32E2xx-bga975';
config.DownloadType = 'JTAG';
config.Args = {coder.typeof(single(ones(2,2))), coder.typeof(single(ones(2,1))), ...
    coder.typeof(single(ones(1,2))), coder.typeof(single(0)), coder.typeof(single(0)),
    coder.typeof(single(0)), coder.typeof(single(0)), coder.typeof(single(ones(2,1))),
    coder.typeof(single(ones(2,2)))};
% PIL settings
config.VerificationMode = 'PIL';
config.PILInterface = 'Serial';
config.PILUartChannelID = '0';
config.PILBaudrate = '115200';
config.PILCOMPort = 'COM13';
config.PILTop = 'kalman_pil_matlab_top_r52';
config.CodeExecutionProfiling = true;

mbd_s32ze.nxp.target.codegen('kalman_filter.m', config);
```

2.1.2 **Run SPF2 Examples**

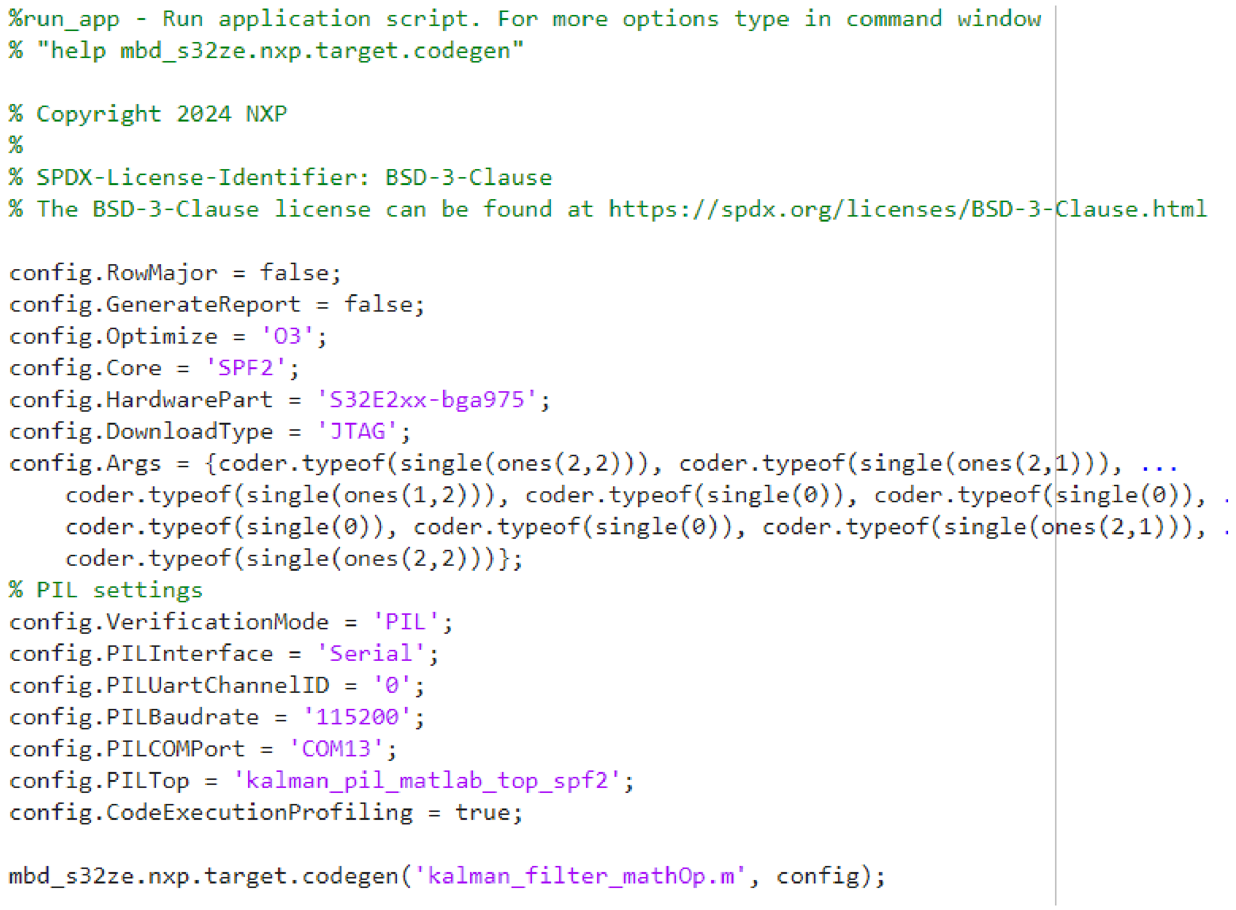
In order to run the SPF2 examples, the user must go into the example directory and run the run_app.m script and this will start the code generation process. The example can be run also in Simulation mode by running the script with the example name.



```
⊟ pil
  ⊞ gauss_newton
  ⊟ kalman_matlab
      kalman_filter_mathOp.m
      kalman_pil_matlab_top_spf2_example_readme....
      kalman_pil_matlab_top_spf2.m
      run_app.m
```

➡ Run in Simulation Mode

➡ Run in Code Generation Mode

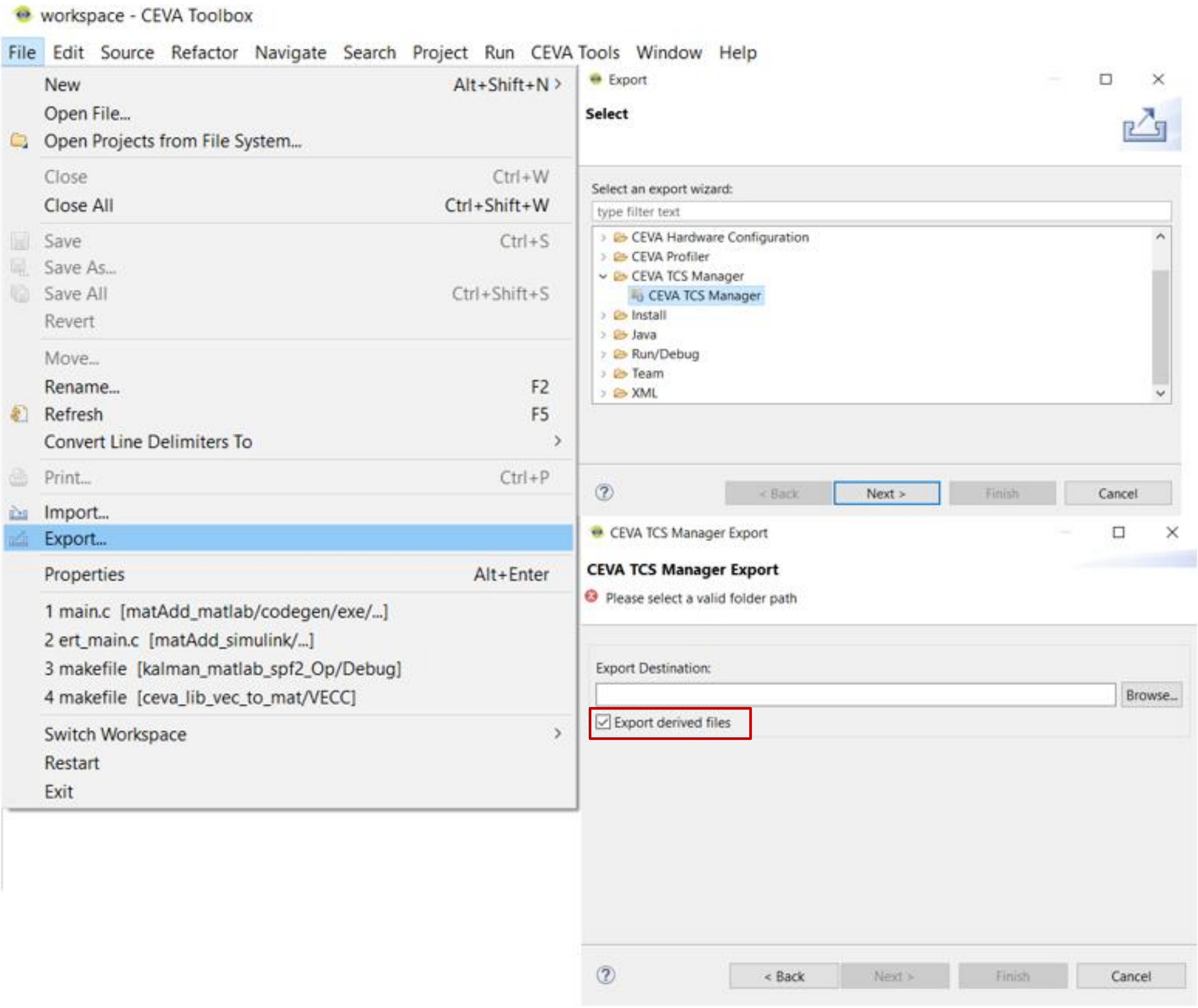


```
%run_app - Run application script. For more options type in command window
% "help mbd_s32ze.nxp.target.codegen"

% Copyright 2024 NXP
%
% SPDX-License-Identifier: BSD-3-Clause
% The BSD-3-Clause license can be found at https://spdx.org/licenses/BSD-3-Clause.html

config.RowMajor = false;
config.GenerateReport = false;
config.Optimize = 'O3';
config.Core = 'SPF2';
config.HardwarePart = 'S32E2xx-bga975';
config.DownloadType = 'JTAG';
config.Args = {coder.typeof(single(ones(2,2))), coder.typeof(single(ones(2,1))), ...
    coder.typeof(single(ones(1,2))), coder.typeof(single(0)), coder.typeof(single(0)), .
    coder.typeof(single(0)), coder.typeof(single(0)), coder.typeof(single(ones(2,1))), .
    coder.typeof(single(ones(2,2)))};
% PIL settings
config.VerificationMode = 'PIL';
config.PILInterface = 'Serial';
config.PILUartChannelID = '0';
config.PILBaudrate = '115200';
config.PILCOMPort = 'COM13';
config.PILTop = 'kalman_pil_matlab_top_spf2';
config.CodeExecutionProfiling = true;

mbd_s32ze.nxp.target.codegen('kalman_filter_mathOp.m', config);
```

In the run_app script the user can select different options, like target name (SPF2), optimization level, download type, etc. The examples for SPF2 core can also be run on *CEVA SensPro Simulator* and for that the user must change the download type from *JTAG* to *CEVA-SensPro Simulator*, which is both instruction set and cycle accurate.
In the SPF2 Examples the user can find the 41 CEVA SPF2 Operators that can be used both in Simulation Mode and Code Generation Mode. These operators are mathematical functions with matrices and vectors, and they are highly optimized for CEVA SPF2 accelerator. The user can also find one Kalman example implemented using basic functions for matrix/vector operations and one using the SPF2 operators.

For SPF2 Examples to work on CEVA-SensPro Simulator, the Target Configuration File (TCS) is needed. If the user does not specify a custom TCS file, then the default one within the toolbox will be used. The TCS file full path can be specified using the `config.tcs` option.

The TCS file can be created in CEVA SDT Toolbox and then exported. This can be done using the export option provided in the CEVA Toolbox.



Please select a valid folder path

☑ Export derived files

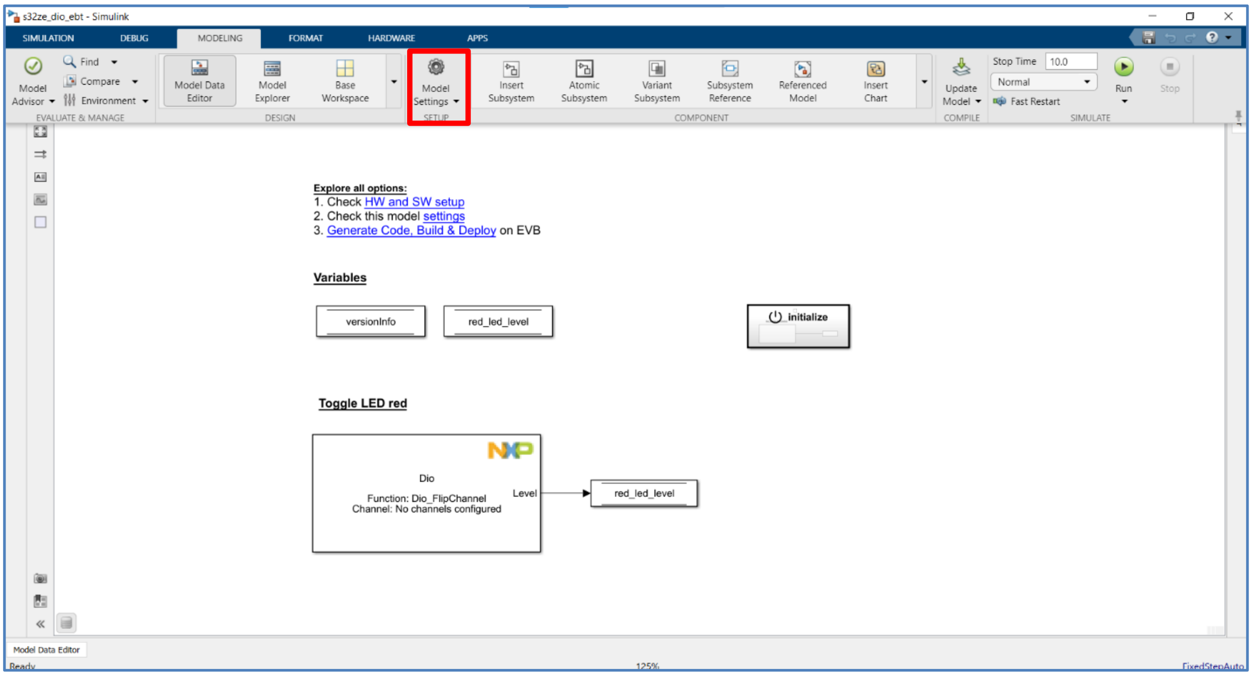
Note: *Be careful to check the Export derived files*

2.2 Simulink

As in MATLAB case, the Simulink examples work on both Simulation and Code Generation Mode and can be found also in the two separated folders for R52 and SPF2 with the MATLAB examples.

2.2.1 Run R52 Examples

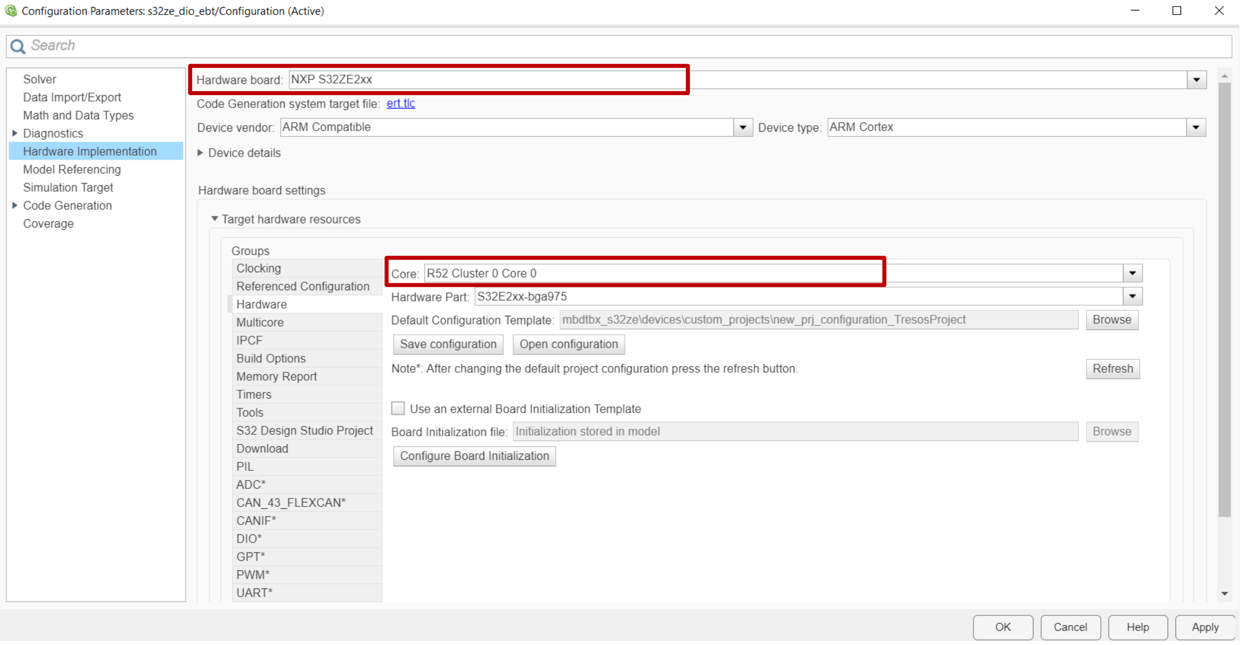
In order to run R52 examples on Simulink, the user has to open the Simulink model and to select the hardware board for R52 core from Model Settings – Hardware Implementation – Hardware - Core. Also in the Hardware board settings the user can find options like Download Type, Clocking value, etc.



Explore all options:
1. Check HW and SW setup
2. Check this model settings
3. Generate Code, Build & Deploy on EVB

Variables

Toggle LED red



Configuration Parameters: s32ze_dio_ebt/Configuration (Active)

Hardware Implementation

Hardware board: NXP S32ZE2xx

Code Generation system target file: ert.tlc

Device vendor: ARM Compatible

Device type: ARM Cortex

Hardware board settings

Target hardware resources

Core: R52 Cluster 0 Core 0

Hardware Part: S32E2xx-bga975

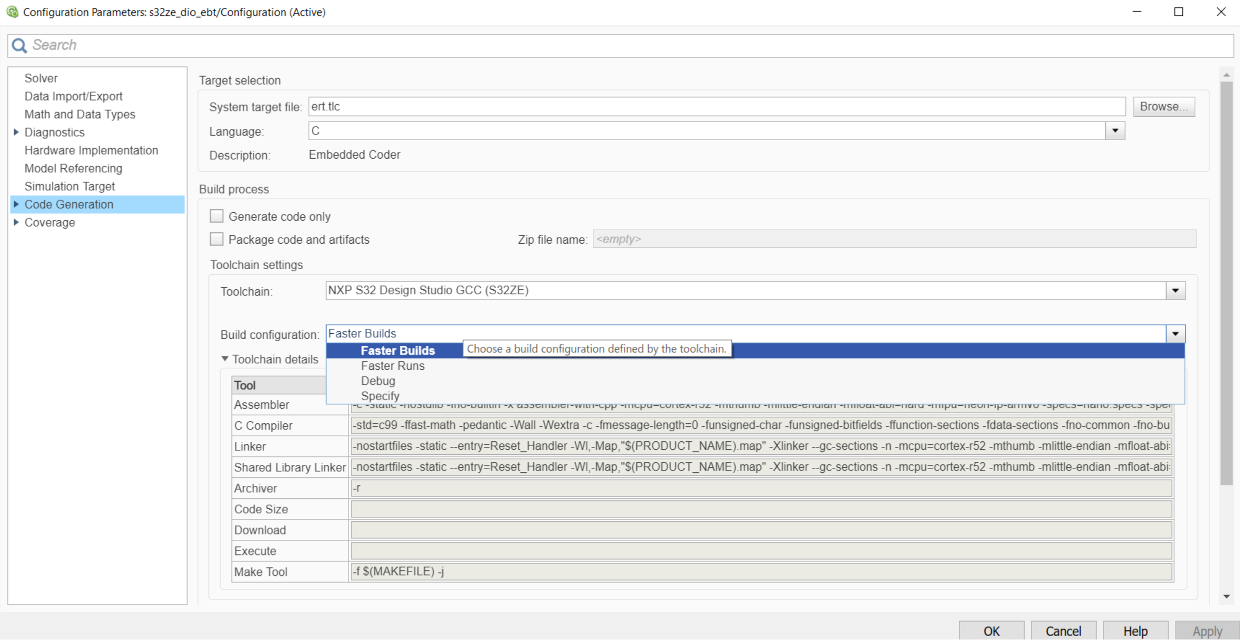
Default Configuration Template: mbdtbx_s32ze\devices\custom_projects\new_prj_configuration_TresosProject

Note*: After changing the default project configuration press the refresh button.

□ Use an external Board Initialization Template

Board Initialization file: Initialization stored in model

In the Code Generation tab, the user can select the desired Build Configuration or he/she can create a new one by selecting the Specify option.



Configuration Parameters: s32ze_dio_ebt/Configuration (Active)

Code Generation

Target selection

System target file: ert.tlc

Language: C

Description: Embedded Coder

Build process

□ Generate code only

□ Package code and artifacts

Zip file name: <empty>

Toolchain settings

Toolchain: NXP S32 Design Studio GCC (S32ZE)

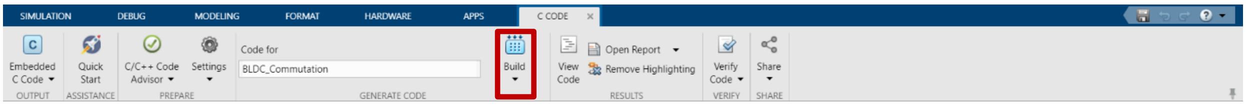
Build configuration: Faster Builds

Choose a build configuration defined by the toolchain.

Toolchain details

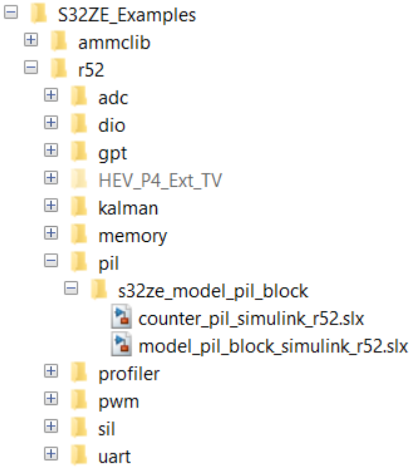
Tool	
Assembler	-c -static -nostdlib -fno-builtin -x assembler-with-cpp -mcpu=cortex-r52 -mthumb -mlittle-endian -mfloat-abi=hard -mfpu=neon-fp-armv8 -specs=nano.specs -spe
C Compiler	-std=c99 -ffast-math -pedantic -Wall -Wextra -c -fmessage-length=0 -funsigned-char -funsigned-bitfields -ffunction-sections -fdata-sections -fno-common -fno-bu
Linker	-nostartfiles -static --entry=Reset_Handler -Wl,-Map,"$(PRODUCT_NAME).map" -Xlinker --gc-sections -n -mcpu=cortex-r52 -mthumb -mlittle-endian -mfloat-abi:
Shared Library Linker	-nostartfiles -static --entry=Reset_Handler -Wl,-Map,"$(PRODUCT_NAME).map" -Xlinker --gc-sections -n -mcpu=cortex-r52 -mthumb -mlittle-endian -mfloat-abi:
Archiver	-r
Code Size	
Download	
Execute	
Make Tool	-f $(MAKEFILE) -j

To run the example in Simulation Mode, the user has to hit the run button and for Code Generation the user must hit the build button from Embedded Coder/C Code tab.



Code for: BLDC_Commutation

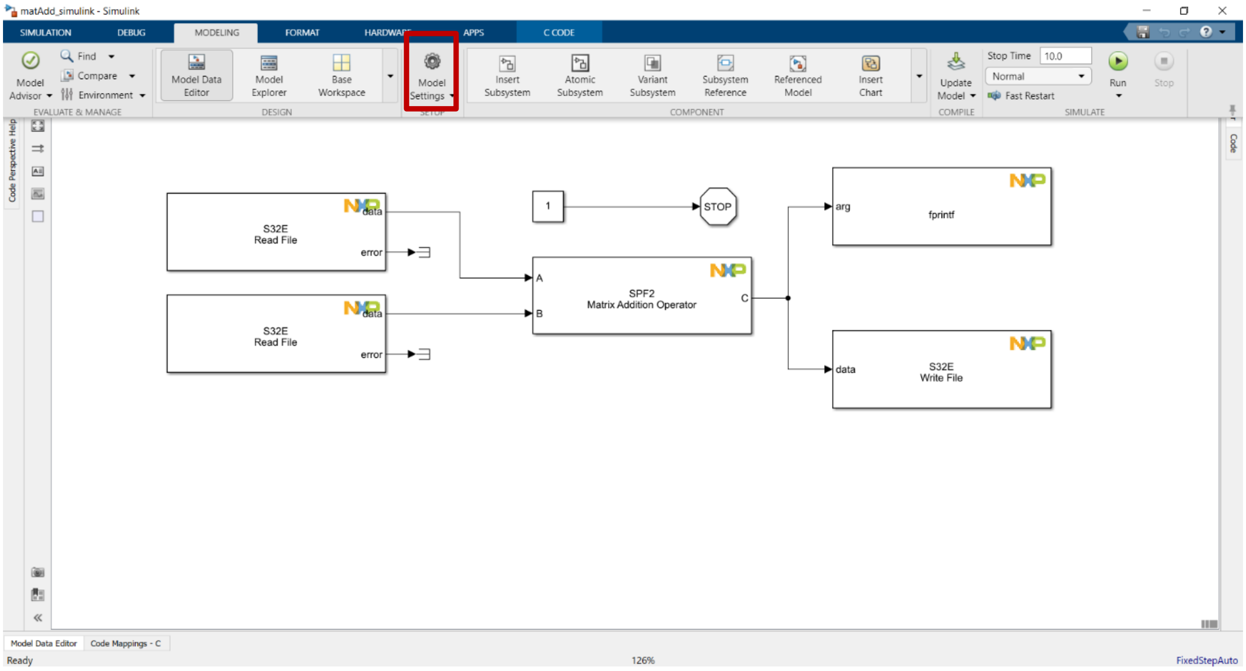
The R52 Simulink Examples also contain PIL examples that can be run using the JTAG download option.



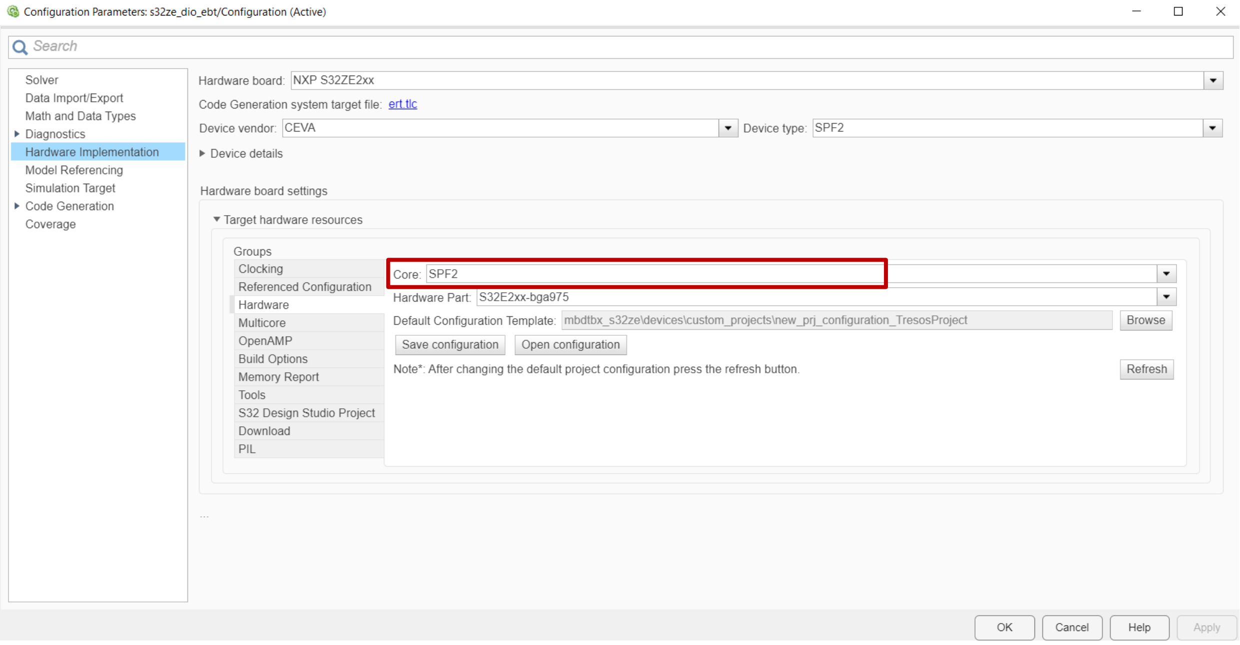
```
S32ZE_Examples
  ammclib
  r52
    adc
    dio
    gpt
    HEV_P4_Ext_TV
    kalman
    memory
    pil
      s32ze_model_pil_block
        counter_pil_simulink_r52.slx
        model_pil_block_simulink_r52.slx
    profiler
    pwm
    sil
    uart
```

2.2.2 **Run SPF2 Examples**

In order to run SPF2 examples on Simulink, the user has to open the Simulink model and select the hardware board for SPF2 core from Model Settings – Hardware Implementation – Hardware Core.



Also in the Hardware board settings the user can find options like Download Type, Export, Clocking value, etc.



Configuration Parameters: s32ze_dio_ebt/Configuration (Active)

Hardware board: NXP S32ZE2xx

Code Generation system target file: ert.tlc

Device vendor: CEVA

Device type: SPF2

Hardware board settings

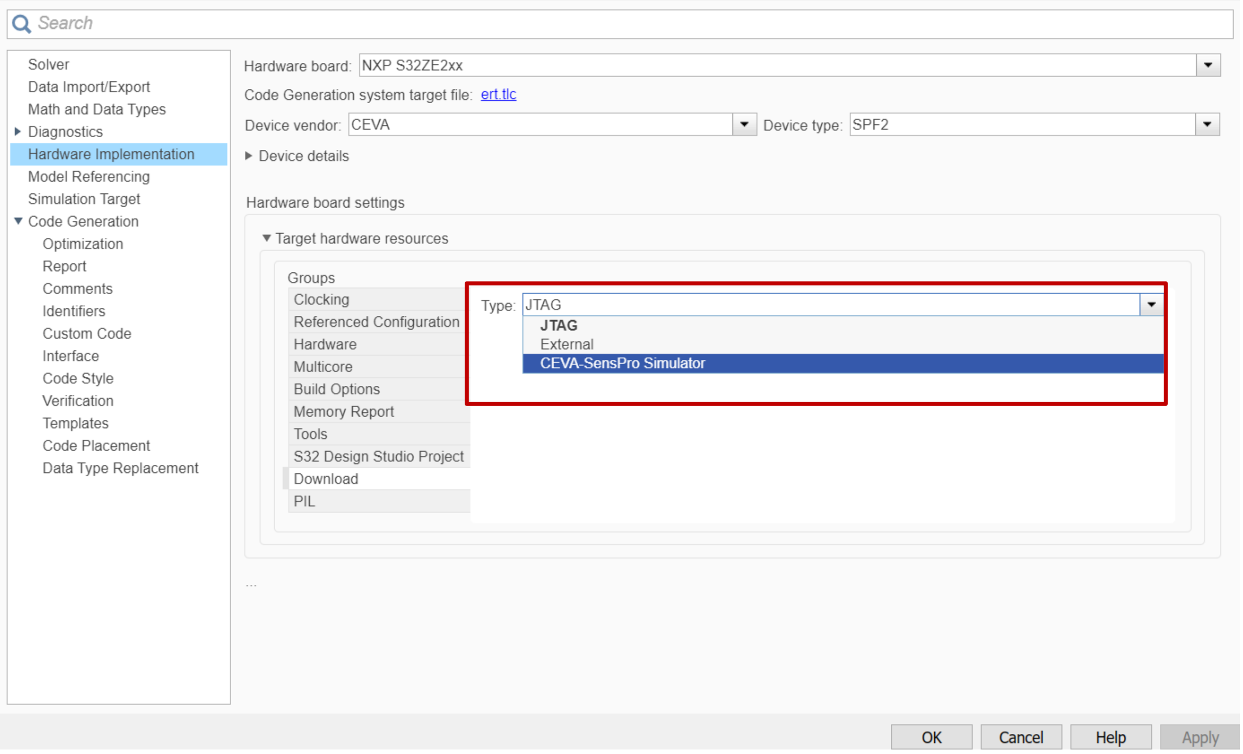
Core: SPF2

Hardware Part: S32E2xx-bga975

Default Configuration Template: mbdtbx_s32ze\devices\custom_projects\new_prj_configuration_TresosProject

Note*: After changing the default project configuration press the refresh button.

In the Download tab the user can select between *JTAG* and *CEVA-SensPro Simulator* and for the last one he/she has to check the "Use Custom CEVA TCS" for using a custom location for TCS file (*see Run SPF2 Examples on MATLAB* for exporting TCS file from CEVA SDT Toolbox).



Hardware board: NXP S32ZE2xx

Code Generation system target file: ert.tlc

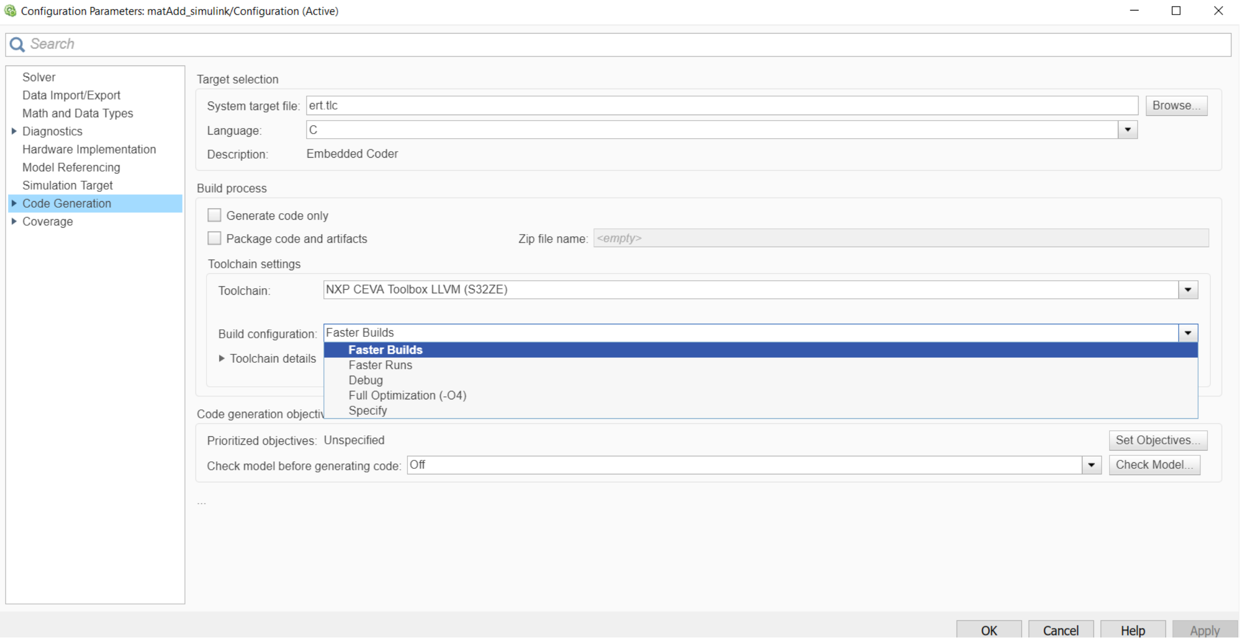
Device vendor: CEVA

Device type: SPF2

Hardware board settings

Type: JTAG

In the Code Generation tab, the user can select the desired Build Configuration or he/she can create a new one by selecting the Specify option.


Configuration Parameters: matAdd_simulink/Configuration (Active)

Target selection

System target file: ert.tlc

Language: C

Description: Embedded Coder

Build process

☐ Generate code only

☐ Package code and artifacts   Zip file name: <empty>

Toolchain settings

Toolchain: NXP CEVA Toolbox LLVM (S32ZE)

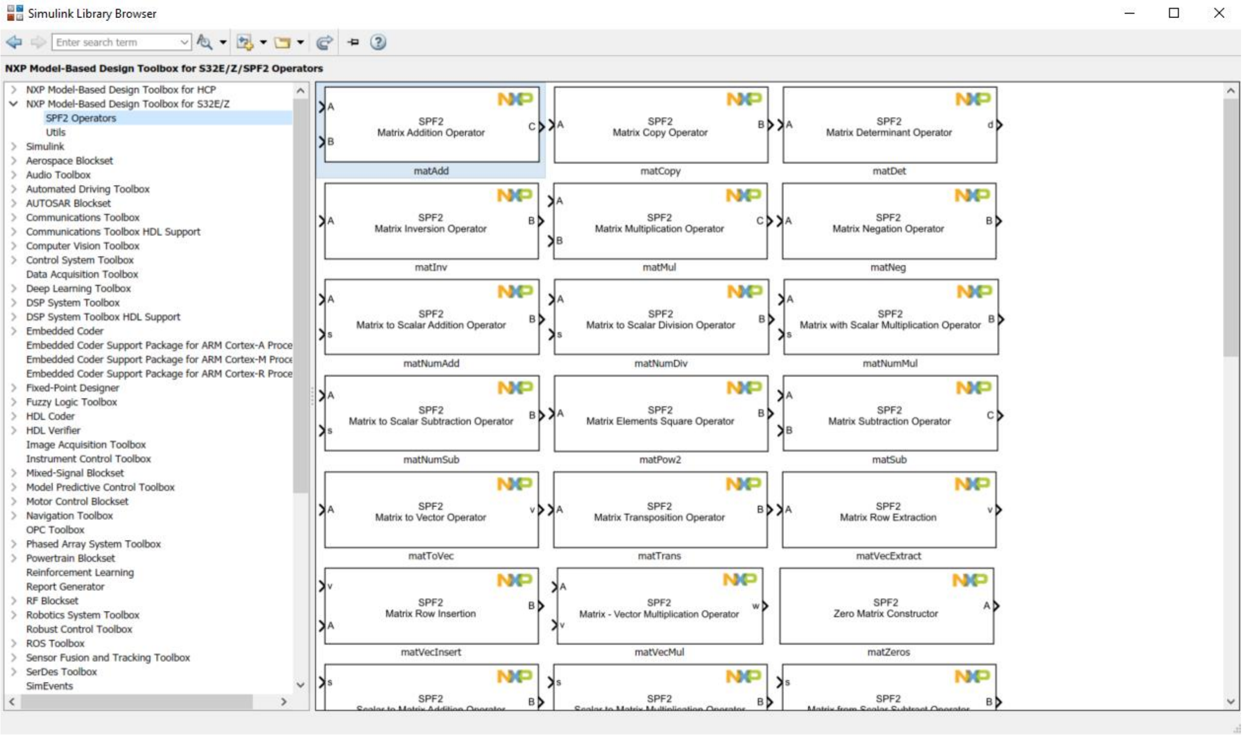
Build configuration: Faster Builds

Code generation objectives

Prioritized objectives: Unspecified

Check model before generating code: Off

The SPF2 Simulink Examples also contain the 41 CEVA SPF2 operators that are highly optimized for SPF2 accelerator and work both in Simulation and Code Generation Mode. These operators can be found in Simulink Library Browser.


Simulink Library Browser

NXP Model-Based Design Toolbox for S32E/Z/SPF2 Operators

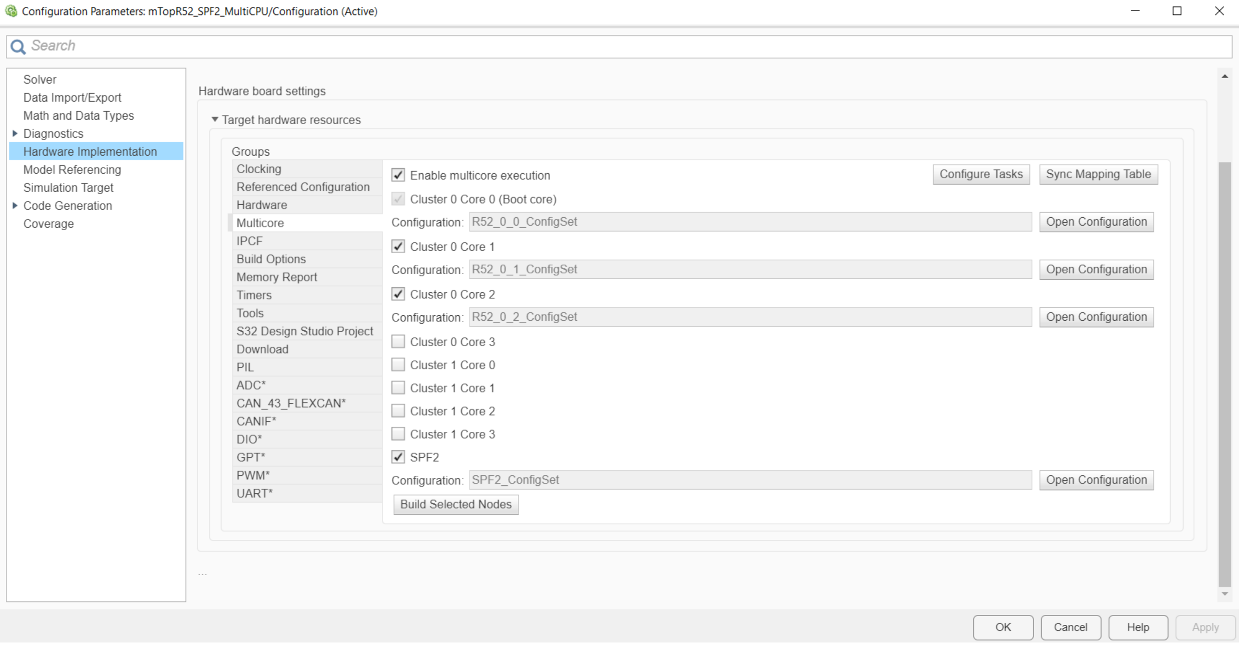
Enter search term
NXP Model-Based Design Toolbox for HCP
NXP Model-Based Design Toolbox for S32E/Z
SPF2 Operators
Utils
Simulink
Aerospace Blockset
Audio Toolbox
Automated Driving Toolbox
AUTOSAR Blockset
Communications Toolbox
Communications Toolbox HDL Support
Computer Vision Toolbox
Control System Toolbox
Data Acquisition Toolbox
Deep Learning Toolbox
DSP System Toolbox
DSP System Toolbox HDL Support
Embedded Coder
Embedded Coder Support Package for ARM Cortex-A Proce
Embedded Coder Support Package for ARM Cortex-M Proce
Embedded Coder Support Package for ARM Cortex-R Proce
Fixed-Point Designer
Fuzzy Logic Toolbox
HDL Coder
HDL Verifier
Image Acquisition Toolbox
Instrument Control Toolbox
Mixed-Signal Blockset
Model Predictive Control Toolbox
Motor Control Blockset
Navigation Toolbox
OPC Toolbox
Phased Array System Toolbox
Powertrain Blockset
Reinforcement Learning
Report Generator
RF Blockset
Robotics System Toolbox
Robust Control Toolbox
ROS Toolbox
Sensor Fusion and Tracking Toolbox
SerDes Toolbox
SimEvents
SPF2 Matrix Addition Operator
matAdd
SPF2 Matrix Copy Operator
matCopy
SPF2 Matrix Determinant Operator
matDet
SPF2 Matrix Inversion Operator
matInv
SPF2 Matrix Multiplication Operator
matMul
SPF2 Matrix Negation Operator
matNeg
SPF2 Matrix to Scalar Addition Operator
matNumAdd
SPF2 Matrix to Scalar Division Operator
matNumDiv
SPF2 Matrix with Scalar Multiplication Operator
matNumMul
SPF2 Matrix to Scalar Subtraction Operator
matNumSub
SPF2 Matrix Elements Square Operator
matPow2
SPF2 Matrix Subtraction Operator
matSub
SPF2 Matrix to Vector Operator
matToVec
SPF2 Matrix Transposition Operator
matTrans
SPF2 Matrix Row Extraction
matVecExtract
SPF2 Matrix Row Insertion
matVecInsert
SPF2 Matrix - Vector Multiplication Operator
matVecMul
SPF2 Zero Matrix Constructor
matZeros
SPF2 Scalar to Matrix Addition Operator
SPF2 Scalar to Matrix Multiplication Operator
SPF2 Matrix from Scalar Subtract Operator

2.3 Multicore Example

The `“\S32ZE_Examples\multicore\mTopR52_SPF2_MultiCPU.mdl”` example shows how to use Concurrent Execution to run a Multicore application on R52 and SPF2 cores. This model illustrates the explicit partitioning functionality in Simulink that allows running different Simulink models on different cores. In this case, the R52 Core 0 Cluster 0 is the boot core that will start the remote cores. One remote core is the SPF2 core, which will offload some parts of the computation using SPF2 Math Operators. The other remote cores, R52 Core 1 Cluster 0 and R52 Core 2 Cluster 0 are doing the same computations, but in this case the R52 Core 2 is sending data, while the R52 Core 1 is doing the math operations. The results of the SPF2 and R52 computations are then validated on the R52 Core 0 and acknowledge messages are sent to the terminal using UART.

To build the multicore application, go to *Model Settings* -> *Hardware Implementation* -> *Target hardware resources* -> *Groups* -> *Multicore* and check *Enable multicore execution* then press the *Build Selected Nodes* button.

The Multicore Top Model is shared between all the cores mapped within the model, so for each core there is an associated Simulink Configuration. Each core configuration can be accessed and modified using the *Open Configuration* button. The Simulink referenced configuration feature is not supported in Multicore usecase, so the user must check that all the models involved in the multicore application have valid configurations (same hardware part, same peripheral configurations between models mapped to the same core, etc.).


Configuration Parameters: mTopR52_SPF2_MultiCPU/Configuration (Active)

Hardware board settings

Target hardware resources

Groups

☑ Enable multicore execution

☑ Cluster 0 Core 0 (Boot core)

Configuration: R52_0_0_ConfigSet

☑ Cluster 0 Core 1

Configuration: R52_0_1_ConfigSet

☑ Cluster 0 Core 2

Configuration: R52_0_2_ConfigSet

☐ Cluster 0 Core 3

☐ Cluster 1 Core 0

☐ Cluster 1 Core 1

☐ Cluster 1 Core 2

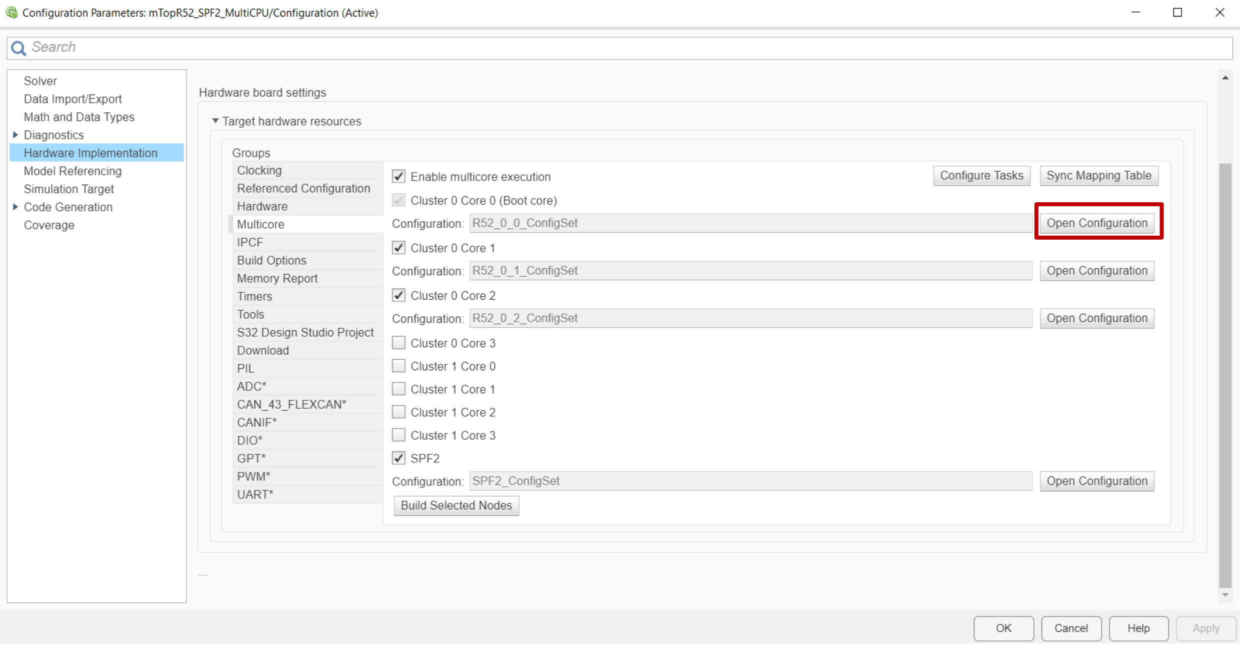
☐ Cluster 1 Core 3

☑ SPF2

Configuration: SPF2_ConfigSet

Simulink Concurrent Execution GUI can be accessed by clicking the *Configure Tasks* button. For more information about explicit partitioning in Simulink, please visit MathWorks websites Concepts in multicore programming, Setup partitions in explicit partitioning.

Peripherals are initialized only by the R52 Cluster 0 Core 0. If other core needs to initialize any peripheral, this must be done by modifying the default board configuration. The board initialization sequence is per model and it is common for all the cores participating in the multicore model. To modify the board initialization, from the Multicore tab, open the configuration for one of the cores used in the application.



Configuration Parameters: mTopR52_SPF2_MultiCPU/Configuration (Active)

Search

Hardware board settings

Target hardware resources

☑ Enable multicore execution

☑ Cluster 0 Core 0 (Boot core)

Configuration: R52_0_0_ConfigSet

☑ Cluster 0 Core 1

Configuration: R52_0_1_ConfigSet

☑ Cluster 0 Core 2

Configuration: R52_0_2_ConfigSet

□ Cluster 0 Core 3

□ Cluster 1 Core 0

□ Cluster 1 Core 1

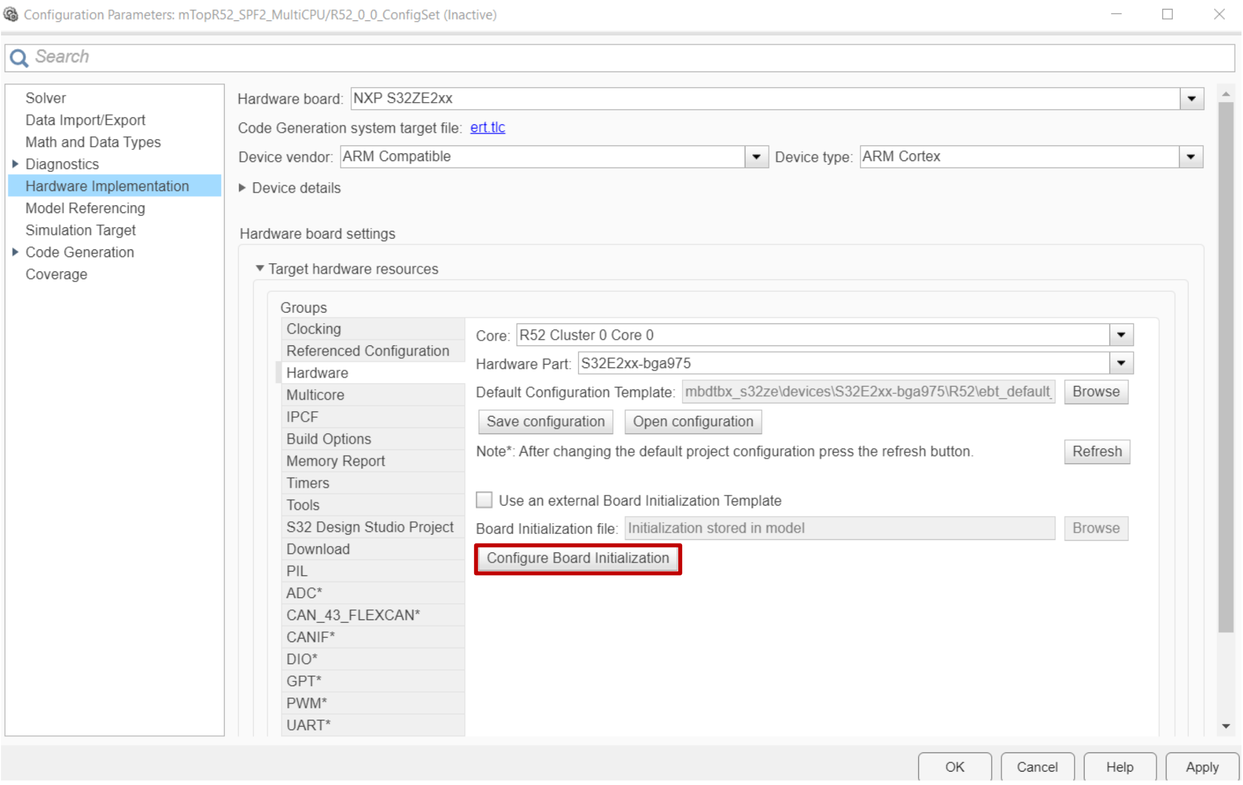
□ Cluster 1 Core 2

□ Cluster 1 Core 3

☑ SPF2

Configuration: SPF2_ConfigSet

For the selected configuration go to *Hardware – Configure Board Initialization*



Configuration Parameters: mTopR52_SPF2_MultiCPU/R52_0_0_ConfigSet (Inactive)

Search

Hardware board: NXP S32ZE2xx

Code Generation system target file: ert.tlc

Device vendor: ARM Compatible

Device type: ARM Cortex

Device details

Hardware board settings

Target hardware resources

Core: R52 Cluster 0 Core 0

Hardware Part: S32E2xx-bga975

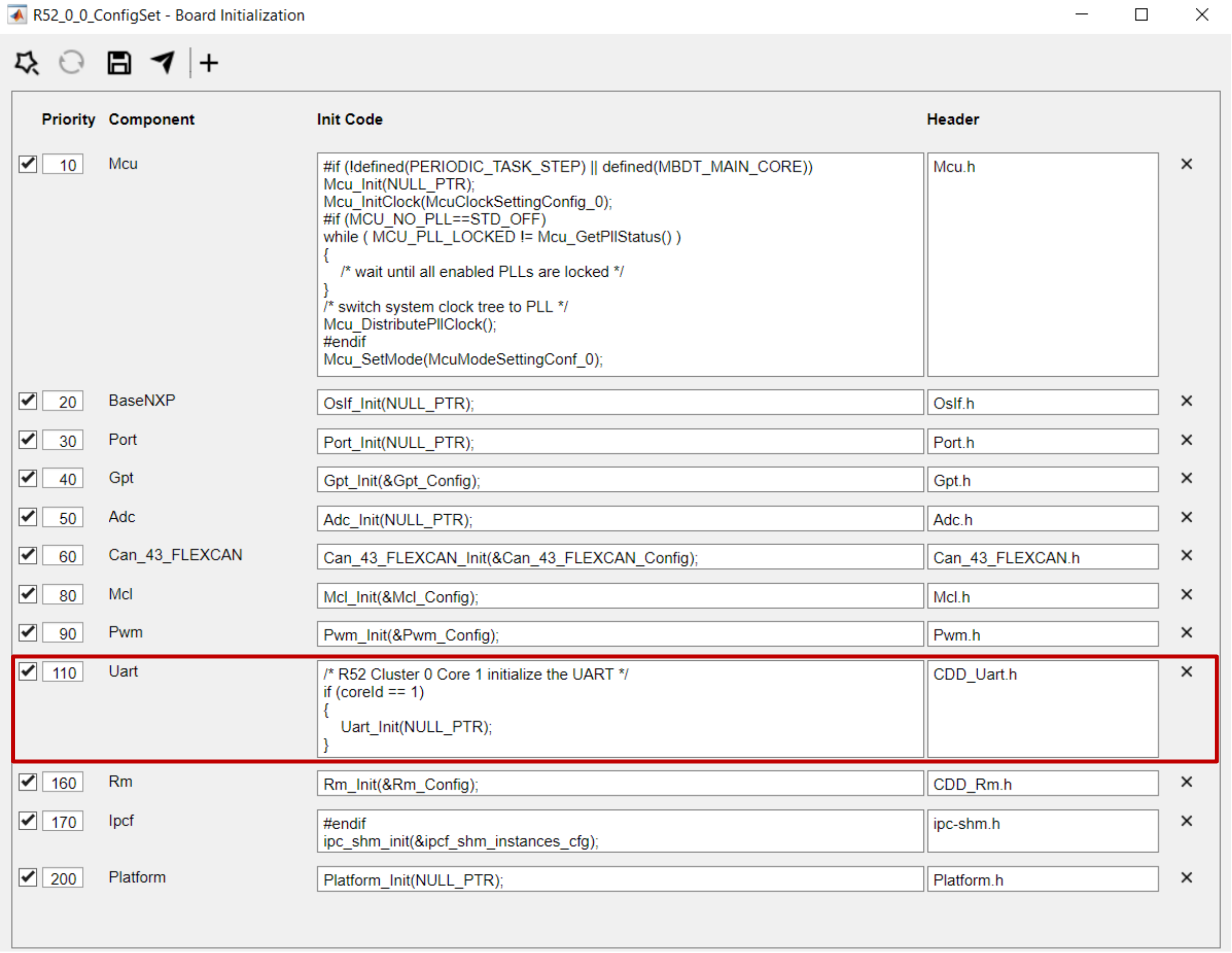
Default Configuration Template: mbdtbx_s32ze\devices\S32E2xx-bga975\R52\ebt_default

Note*: After changing the default project configuration press the refresh button.

□ Use an external Board Initialization Template

Board Initialization file: Initialization stored in model

Modify the board initialization as needed (e.g. Cluster 0 Core 1 initialize the UART).



R52_0_0_ConfigSet - Board Initialization

	Priority	Component	Init Code	Header	

☑	10	Mcu	#if (!defined(PERIODIC_TASK_STEP) || defined(MBDT_MAIN_CORE))
			Mcu_Init(NULL_PTR);
			Mcu_InitClock(McuClockSettingConfig_0);
			#if (MCU_NO_PLL==STD_OFF)
			while (MCU_PLL_LOCKED != Mcu_GetPllStatus())
			{
			    /* wait until all enabled PLLs are locked */
			}
			/* switch system clock tree to PLL */
			Mcu_DistributePllClock();
			#endif
			Mcu_SetMode(McuModeSettingConf_0);
				Mcu.h	×

☑	20	BaseNXP	OsIf_Init(NULL_PTR);	OsIf.h	×
☑	30	Port	Port_Init(NULL_PTR);	Port.h	×
☑	40	Gpt	Gpt_Init(&Gpt_Config);	Gpt.h	×
☑	50	Adc	Adc_Init(NULL_PTR);	Adc.h	×
☑	60	Can_43_FLEXCAN	Can_43_FLEXCAN_Init(&Can_43_FLEXCAN_Config);	Can_43_FLEXCAN.h	×
☑	80	Mcl	Mcl_Init(&Mcl_Config);	Mcl.h	×
☑	90	Pwm	Pwm_Init(&Pwm_Config);	Pwm.h	×

☑	110	Uart	/* R52 Cluster 0 Core 1 initialize the UART */
			if (coreId == 1)
			{
			    Uart_Init(NULL_PTR);
			}
				CDD_Uart.h	×

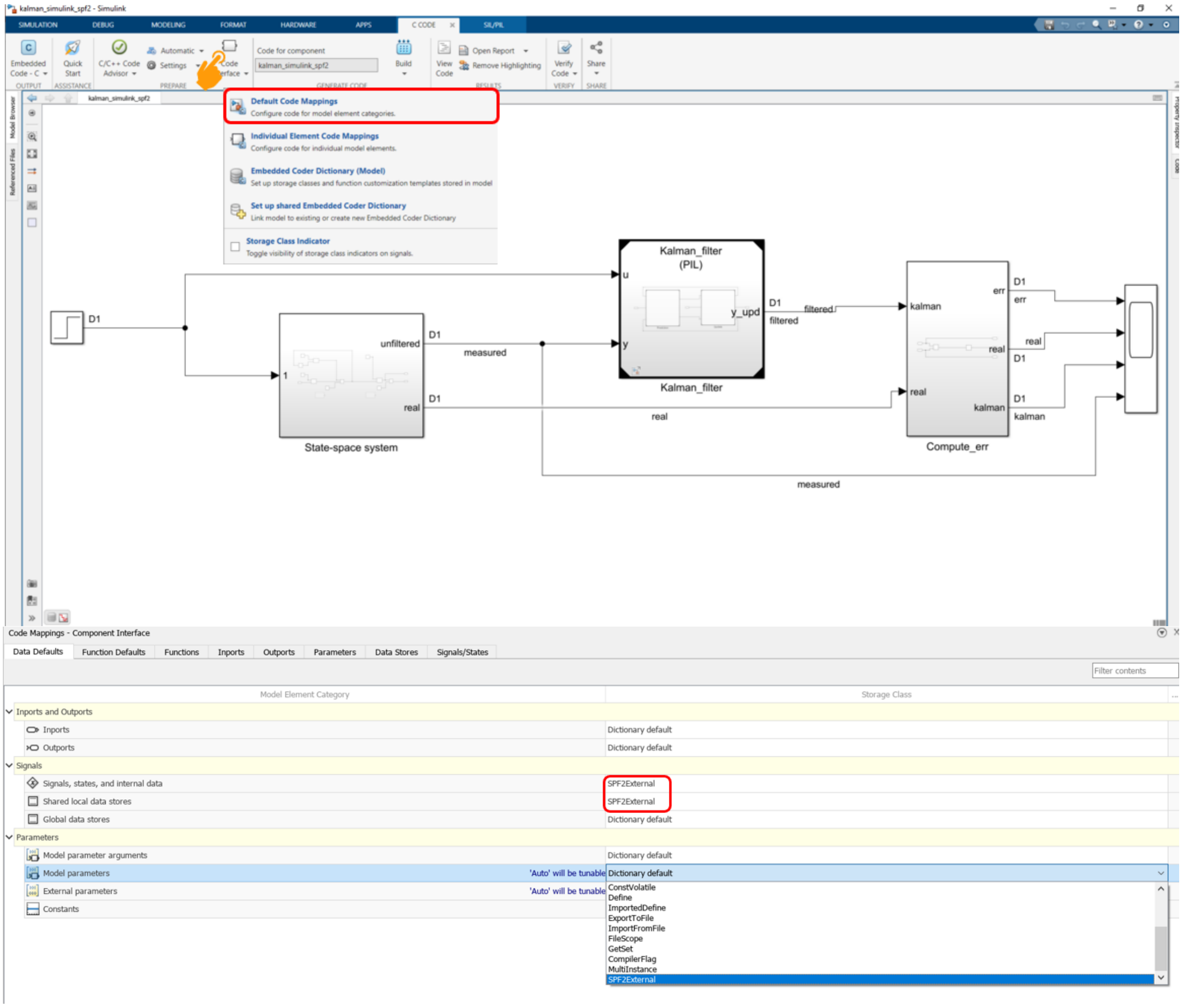
☑	160	Rm	Rm_Init(&Rm_Config);	CDD_Rm.h	×

☑	170	Ipcf	#endif
			ipc_shm_init(&ipcf_shm_instances_cfg);
				ipc-shm.h	×

☑	200	Platform	Platform_Init(NULL_PTR);	Platform.h	×

2.4 Memory Optimizer SPF2

The memory optimizer allows the user to select the memory section where he/she wants to map different model elements.

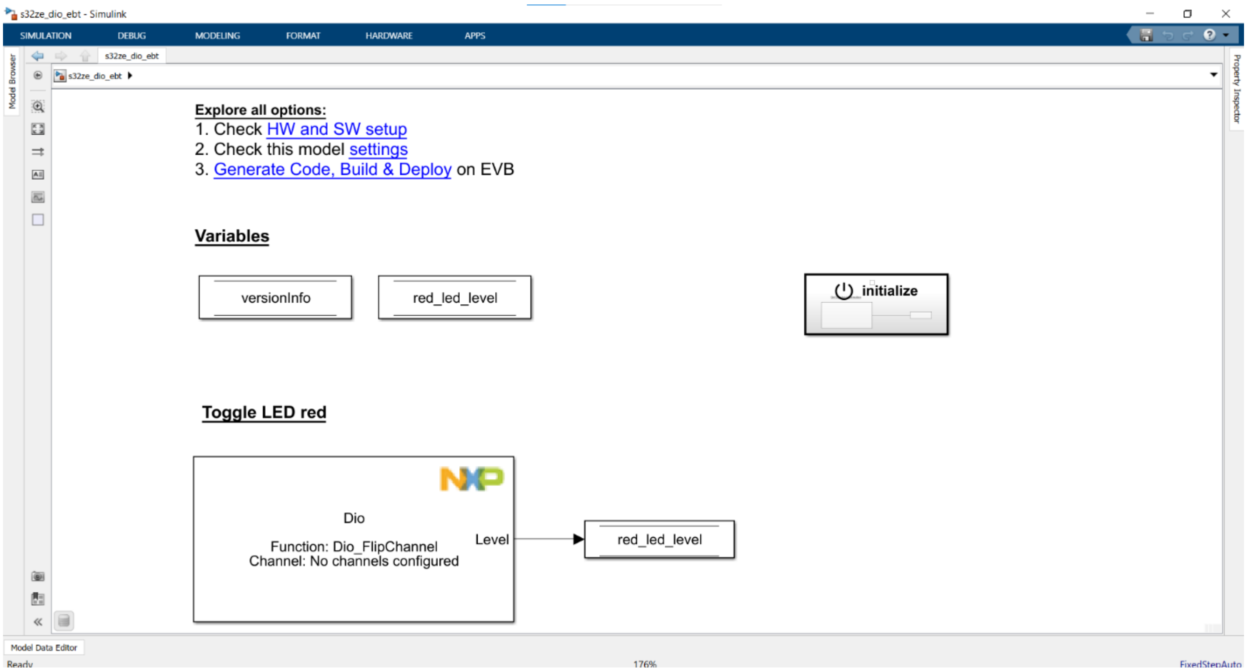


Model Element Category		Storage Class
Inports and Outports		
Inports		Dictionary default
Outports		Dictionary default
Signals		
Signals, states, and internal data		SPF2External
Shared local data stores		SPF2External
Global data stores		Dictionary default
Parameters		
Model parameter arguments		Dictionary default
Model parameters	'Auto' will be tunable	Dictionary default
External parameters	'Auto' will be tunable	ConstVolatile Define ImportedDefine ExportToFile ImportFromFile FileScope GetSet CompilerFlag MultiInstance SPF2External
Constants		

2.5 A “Hello World” Example

If the hardware setup is completed successfully, then all ingredients are present for running successfully the Model-Based Design Toolbox for S32Z/E specific examples. The examples delivered by the toolbox are targeting the S32Z27x and S32E27x processors of the S32ZE family. Moreover, for the supported derivatives, the Model-Based Design Toolbox provides examples using EB Tresos to demonstrate the interaction with the configuration tools it provides integration with.

Navigate to "`\S32ZE_Examples\dio`" folder and open the model according to the hardware used and the desired configuration tool (e.g `s32ze_dio_ebt.mdl`)



Explore all options:

1. Check HW and SW setup
2. Check this model settings
3. Generate Code, Build & Deploy on EVB

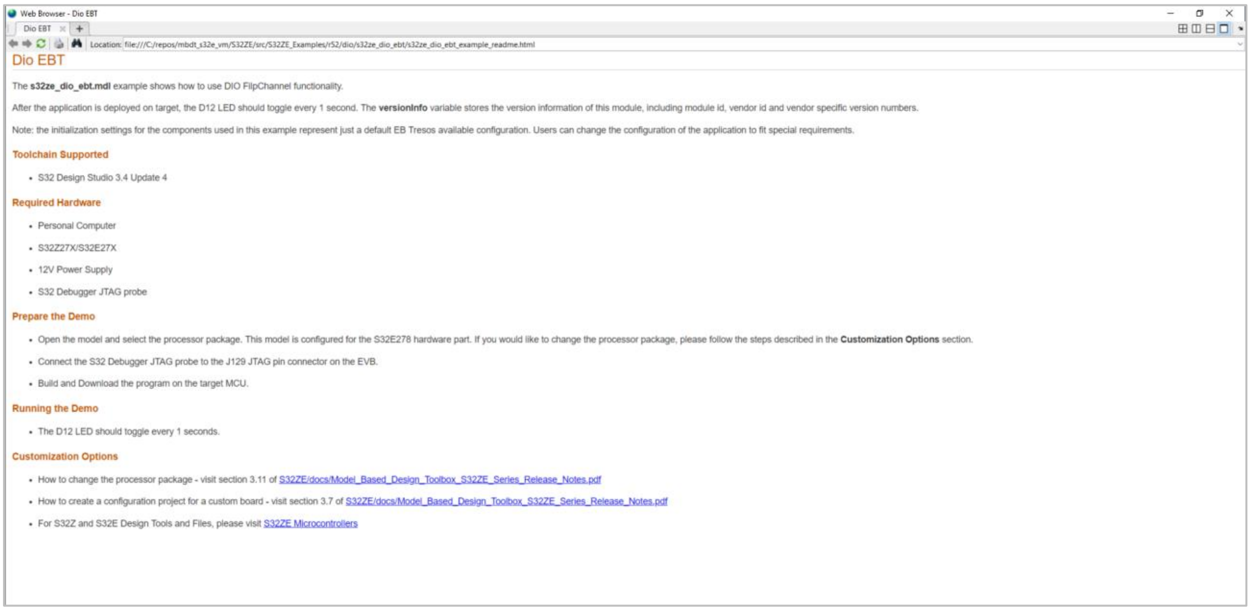
Variables

Toggle LED red

This model programs the S32E27X-DC EVB to toggle the D12 red LED every 1 seconds.

Follow the next steps to run the example:

1. Open and README.html file to understand the hardware and software requirements for running the application



Web Browser - Dio EBT
Dio EBT
Location: file:///C:/repos/mbdt_s32e_vm/S32ZE/src/S32ZE_Examples/r52/dio/s32ze_dio_ebt/s32ze_dio_ebt_example_readme.html

Dio EBT

The **s32ze_dio_ebt.mdl** example shows how to use DIO FlipChannel functionality.

After the application is deployed on target, the D12 LED should toggle every 1 second. The **versionInfo** variable stores the version information of this module, including module id, vendor id and vendor specific version numbers.

Note: the initialization settings for the components used in this example represent just a default EB Tresos available configuration. Users can change the configuration of the application to fit special requirements.

Toolchain Supported

- S32 Design Studio 3.4 Update 4

Required Hardware

- Personal Computer
- S32Z27X/S32E27X
- 12V Power Supply
- S32 Debugger JTAG probe

Prepare the Demo

- Open the model and select the processor package. This model is configured for the S32E278 hardware part. If you would like to change the processor package, please follow the steps described in the **Customization Options** section.
- Connect the S32 Debugger JTAG probe to the J129 JTAG pin connector on the EVB.
- Build and Download the program on the target MCU.

Running the Demo

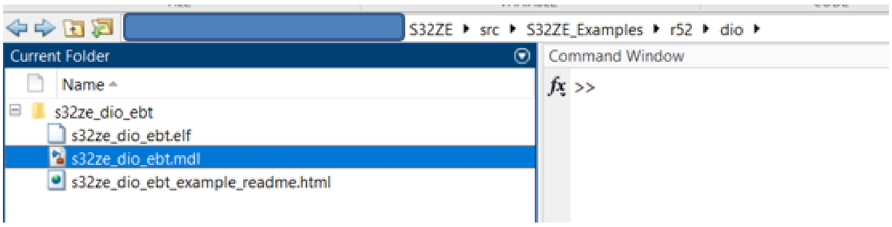
- The D12 LED should toggle every 1 seconds.

Customization Options

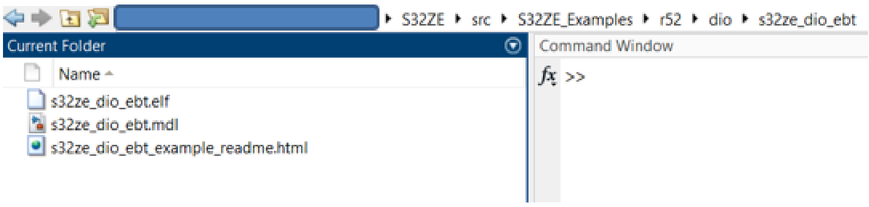
- How to change the processor package - visit section 3.11 of S32ZE/docs/Model_Based_Design_Toolbox_S32ZE_Series_Release_Notes.pdf
- How to create a configuration project for a custom board - visit section 3.7 of S32ZE/docs/Model_Based_Design_Toolbox_S32ZE_Series_Release_Notes.pdf
- For S32Z and S32E Design Tools and Files, please visit S32ZE Microcontrollers

2. Press the `Build Model` button and wait until the code is generated, compiled, and downloaded to the evaluation board. Alternatively, you can press on the text highlighted in the model to start the process automatically.

3. The build of the model must be done while the user is inside the folder of the model.



Incorrect



▸ S32ZE ▸ src ▸ S32ZE_Examples ▸ r52 ▸ dio ▸ s32ze_dio_ebt
Current Folder
Name
s32ze_dio_ebt.elf
s32ze_dio_ebt.mdl
s32ze_dio_ebt_example_readme.html
Command Window
fx >>

Correct

If you see the LEDs toggling, congratulations! You succeeded in running your first example created with

Model-Based Design Toolbox for S32Z/E

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