Model-Based Design Toolbox S32K1xx Series

Quick Start Guide

An Embedded Target for the S32K1xx Family of Processors
Version 4.1.0

Target Based Automatic Code Generation Tools
For MATLAB™/Simulink™/Stateflow™ Models working with Simulink Coder ™ and Embedded Coder®

© 2019 NXP Semiconductors. All rights reserved

Summary

1 Installation 1-3
1.1 System Requirements 1-3
1.2 Installation Steps 1-3
1.2.1 Run Add-on installer 1-4
1.2.2 License Registration & Installation 1-7
1.2.3 Setting up the Target Compilers 1-8
1.2.4 Setting the Path for Model-Based Design Toolbox 1-8
2 Run Models 2-9
2.1 Examples Library & Help 2-9
2.2 Hardware Setup 2-11
2.3 Hello World Example 2-12

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.

1.1 System Requirements

For a flowless development experience the minimum recommended PC platform is:

- *Windows® OS*: any x64 processor
- At least 4 GB of RAM
- At least 6 GB of free disk space.
- Internet connectivity for web downloads.

Operating System Supported

	SP Level	64-bit
Windows 7	SP1	X
Windows 10		X

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 package, assuming you have already downloaded the file from NXP's official web page.

To have the toolbox installed and configured properly the following actions should be executed:

1. Run the MATLAB toolbox package file `*.mltbx` downloaded from NXP's Model-Based Design Toolbox web page
2. Register and install the toolbox license file into `..\MATLAB\Add-Ons\Toolboxes\NXP_MBDToolbox_S32K1xx\code\lic`
3. Set the Target Compiler Environment Variables
4. Setup the MATLAB path for Model-Based Design Toolbox

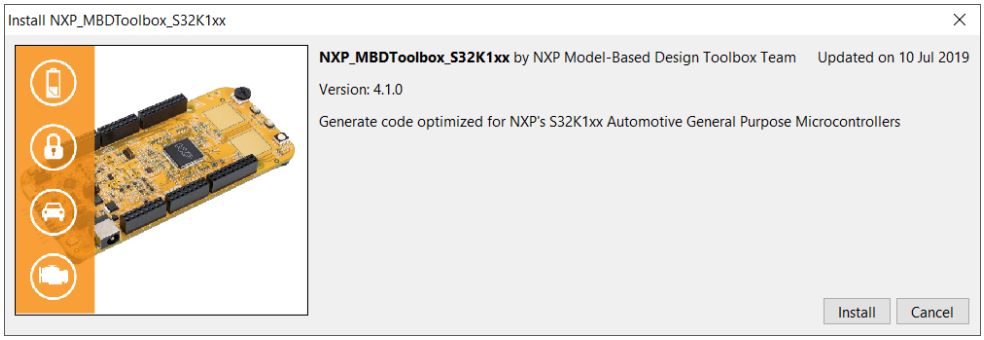
Each of these actions are explained in the following sub-chapters.

1.2.1 **Run Add-on installer**

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.

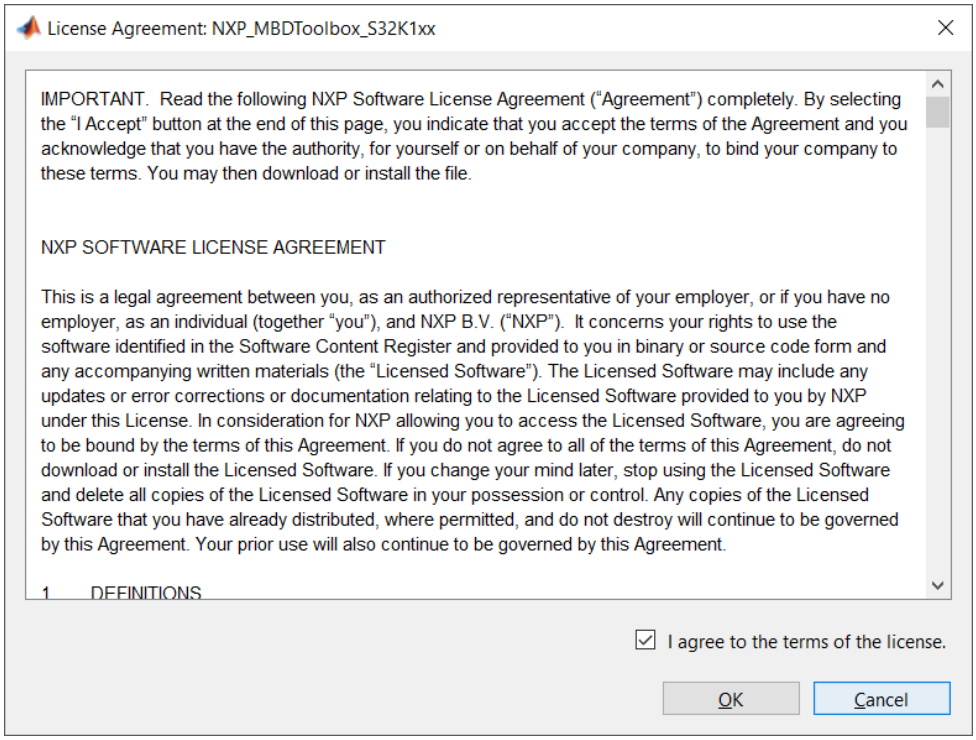
After the MATLAB opens, you will be prompted with the following options:

1. The NXP's Model-Based Design Toolbox Installation Wizard dialog will appear. Click "`Install`" to proceed.



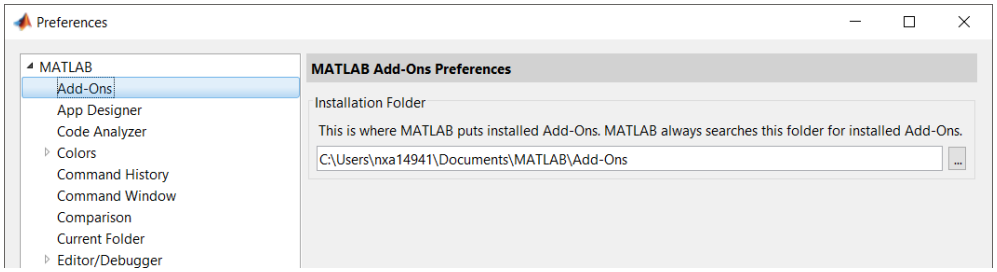
Note: *Depending on the MATLAB version used to install the MLTBX files, this dialogue might be visible or not*

2. Indicate acceptance of the NXP Software License Agreement by selecting "`I agree to the terms of the license`" to proceed.



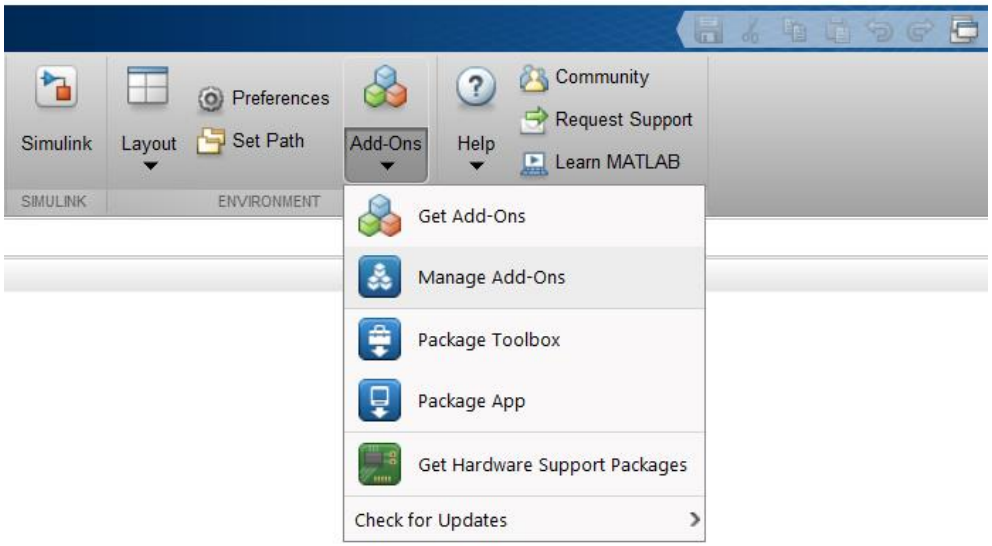
3. Click "OK" to start the MATLAB installation process. The rest of the process is silent and under MATLAB control. All the files will be automatically copied into default Add-Ons folder within the MATLAB

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

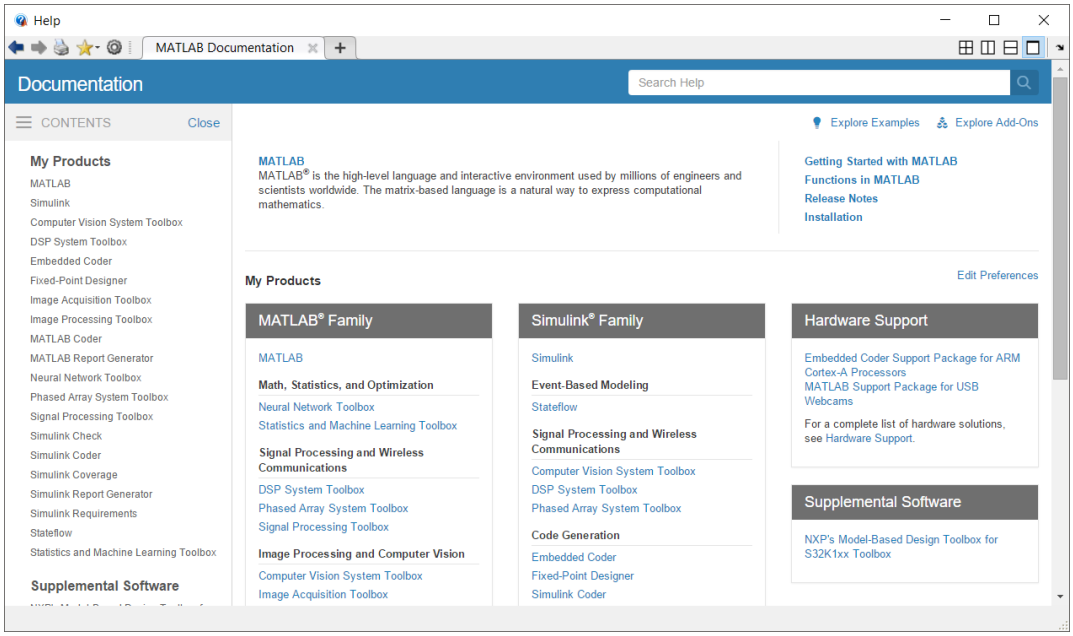


Note: *It is recommended to install the MATLAB and NXP Toolbox into location that do not contains special characters, empty spaces, or mapped drives.*

4. After a couple of minutes (2-5min), the NXP's Model-Based Design Toolbox should be visible as a new Add-ons.



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

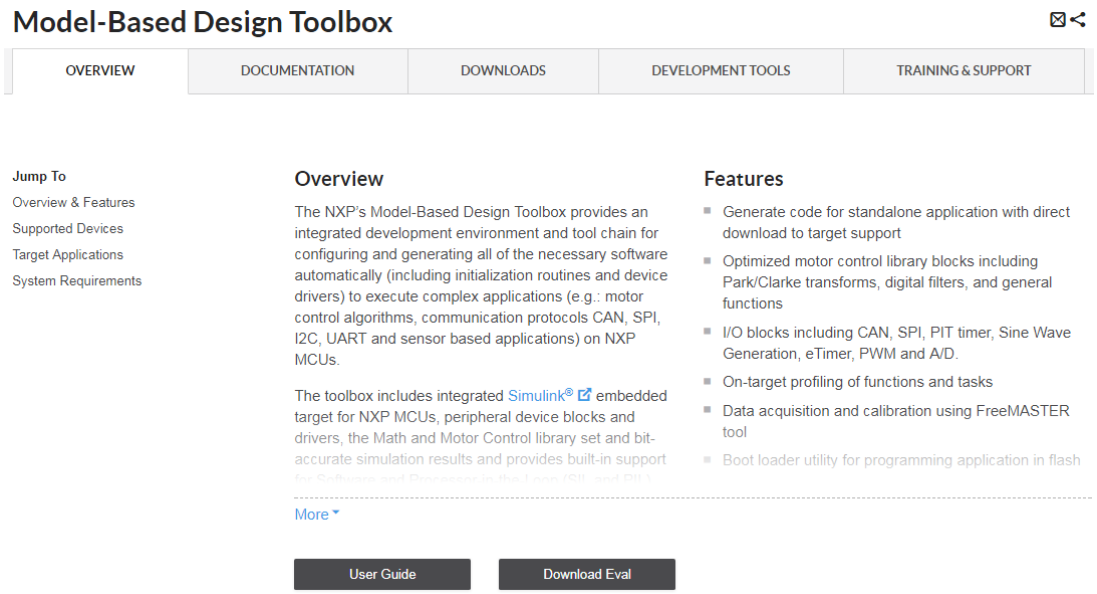


1.2.2 License Registration & Installation

The NXP's Model Based Design Toolbox is available free of charge, however, a license is required. If you need to get a license it can be obtained by following the path outlined below.
If you encounter issues getting a license please submit a ticket at https://www.nxp.com/support/support:SUPPORTHOME describing the issue.

Please perform the following steps to obtain your license:

1. Go to www.nxp.com/mbdt and click on "`Download Eval`"



2. Once you have logged in, the "`Product Information`" page for Model-Based Design Toolbox appears.

   Select the appropriate toolbox you wish you evaluate to bring up the Software Terms and Conditions page.

3. Click "`I Agree`" to consent to the software license agreement.

4. If you need to download the tool, click on the linked file name. Otherwise, click on "`License Keys`" tab.

5. Verify the correct tool is identified and click on "`Generate`".

6. Enter your Disk Serial Number as the `Host ID`. If you do not know your Disk Serial Number, go to "`Locating the Host ID`" in the document **Model_Based_Design_Toolbox_S32K1xx_Series_Licensing_Guide** to learn how to locate your Disk Serial Number, which is needed to generate your license.

7. Click "`Generate`"

8. Either click on "`Save All`" or copy and paste the file into a text editor and save the file as "`license.lic`" to the folder "`..\MATLAB\Add-Ons\Toolboxes\NXP_MBDToolbox_S32K1xx\code\lic`"

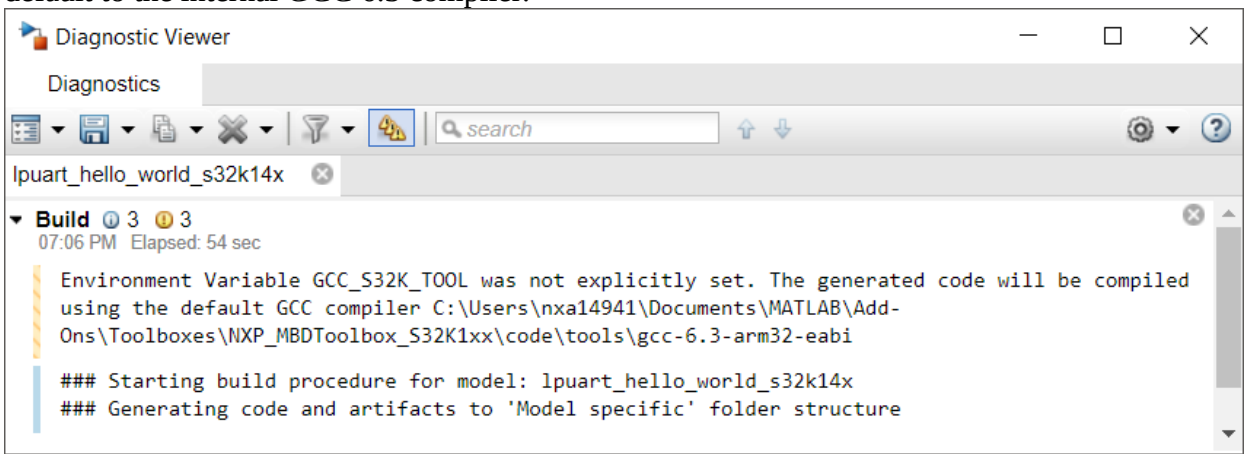
1.2.3 Setting up the Target Compilers

The target compiler for Model-Based Design Toolbox needs to be configured. Use the notation below to setup these compiler environmental variables. Ensure a system environment variable called <COMPILER_STRING>_TOOL, corresponding to the compiler(s) you have installed, is defined to compiler path value as shown below:

```
GCC_S32K_TOOL = {Toolbox installation path}\tools\gcc-6.3-arm32-eabi
IAR_TOOL = {IAR installation path}/IAR Systems/Embedded Workbench 7.3
GHS_TOOL = {GHS installation path}/multi517
```

Note: *Paths shown are for illustration, your installation path may be different. Once environmental variables are setup you will need to restart MATLAB for the IDE environment to see these system variables.*

In case there is no compiler installed, the NXP's Model-Based Design Toolbox is going to default to the internal GCC 6.3 compiler.



1.2.4 Setting the Path for Model-Based Design Toolbox

If the toolbox installation as Add-ons was successful, then the MATLAB will automatically detect the NXP's Model-Based Design Toolbox. Nonetheless, in special circumstances you might need to run a special script to instruct the MALAB to recognize the NXP's Model Based Design Toolbox. This is done by changing the MATLAB Current Directory to the toolbox installation directory (e.g.: ..\MATLAB\Add-Ons\Toolboxes\NXP_MBDToolbox_S32K1xx\code) and running the "mbd_s32k_path" script.

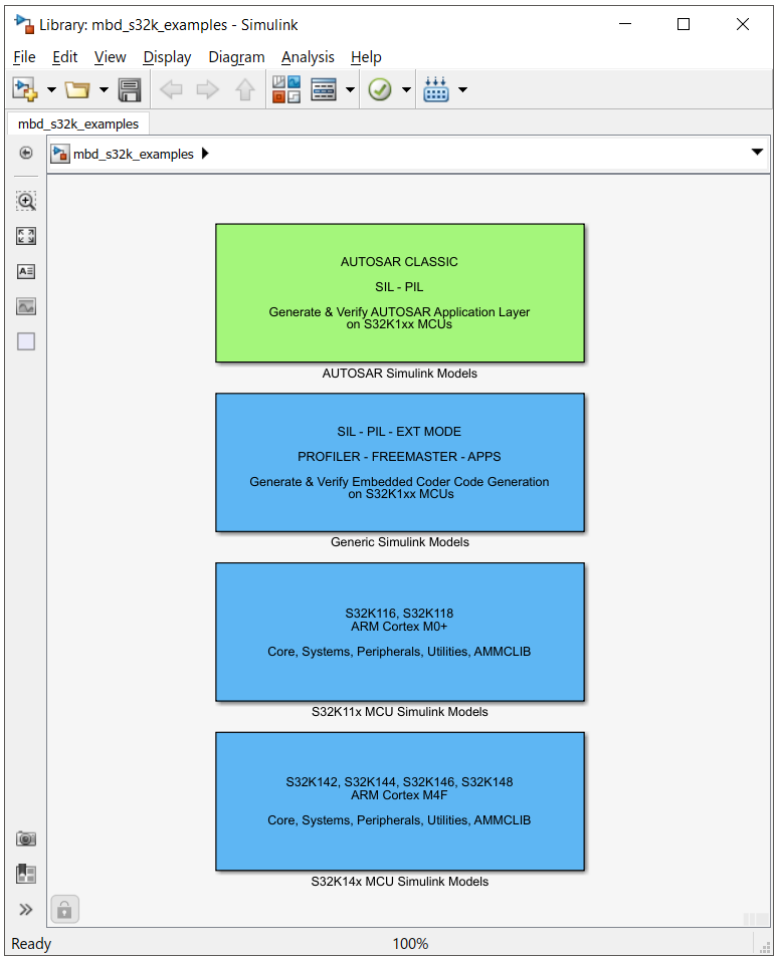
```
>> mbd_s32k_path
Treating 'C:\Users\b21307\Documents\MATLAB\Add-
Ons\Toolboxes\NXP_MBDToolbox_S32K1xx\code' as MBD Toolbox installation
root.
MBD Toolbox path prepended.
Successful.
>>
```

2 Run Models

2.1 Examples Library & Help

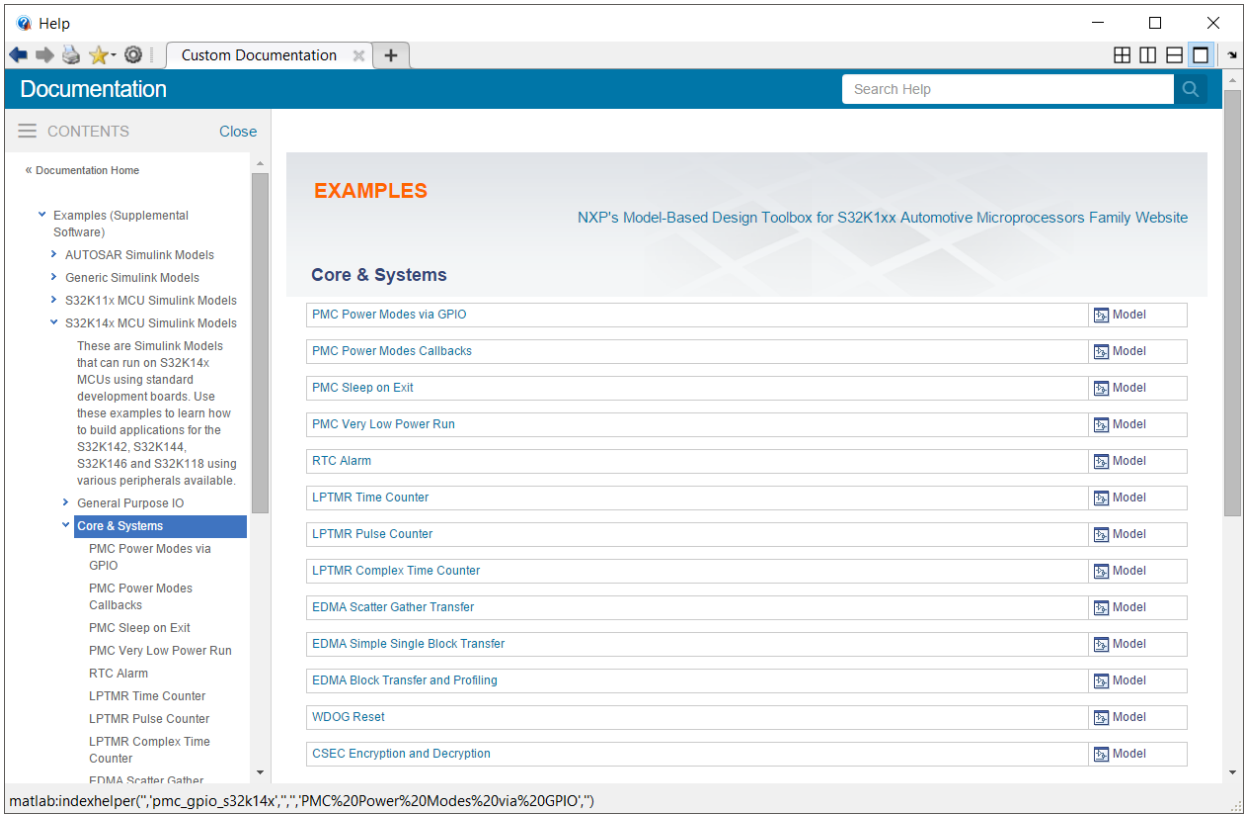
NXP's Model Based Design Toolbox comes with an Examples Library collection that let you test different MCU on-chip modules and run complex applications.

The Examples Library `mbd_s32k_examples.mdl` can be opened from "`{Model Based Design Install Directory}\S32_Examples\`" folder.



Each category contains multiple examples that showcase different Model-Based Design Toolbox capabilities that are categorized in different groups.

The examples are also available from standard MATLAB Help for NXP's Model-Based Design Toolbox Example


Help
Custom Documentation
Documentation
Search Help
CONTENTS
Close
« Documentation Home
Examples (Supplemental Software)
AUTOSAR Simulink Models
Generic Simulink Models
S32K11x MCU Simulink Models
S32K14x MCU Simulink Models
These are Simulink Models that can run on S32K14x MCUs using standard development boards. Use these examples to learn how to build applications for the S32K142, S32K144, S32K146 and S32K118 using various peripherals available.
General Purpose IO
Core & Systems
PMC Power Modes via GPIO
PMC Power Modes Callbacks
PMC Sleep on Exit
PMC Very Low Power Run
RTC Alarm
LPTMR Time Counter
LPTMR Pulse Counter
LPTMR Complex Time Counter
EXAMPLES
NXP's Model-Based Design Toolbox for S32K1xx Automotive Microprocessors Family Website
Core & Systems
PMC Power Modes via GPIO Model
PMC Power Modes Callbacks Model
PMC Sleep on Exit Model
PMC Very Low Power Run Model
RTC Alarm Model
LPTMR Time Counter Model
LPTMR Pulse Counter Model
LPTMR Complex Time Counter Model
EDMA Scatter Gather Transfer Model
EDMA Simple Single Block Transfer Model
EDMA Block Transfer and Profiling Model
WDOG Reset Model
CSEC Encryption and Decryption Model
matlab:indexhelper('','pmc_gpio_s32k14x','','','PMC%20Power%20Modes%20via%20GPIO','')

2.2 Hardware Setup

All examples provided with the Model-Based Design Toolbox were developed on S32K144EVB-Q100 as primary target. Additional information about this development kit can be found on NXP official web page here.

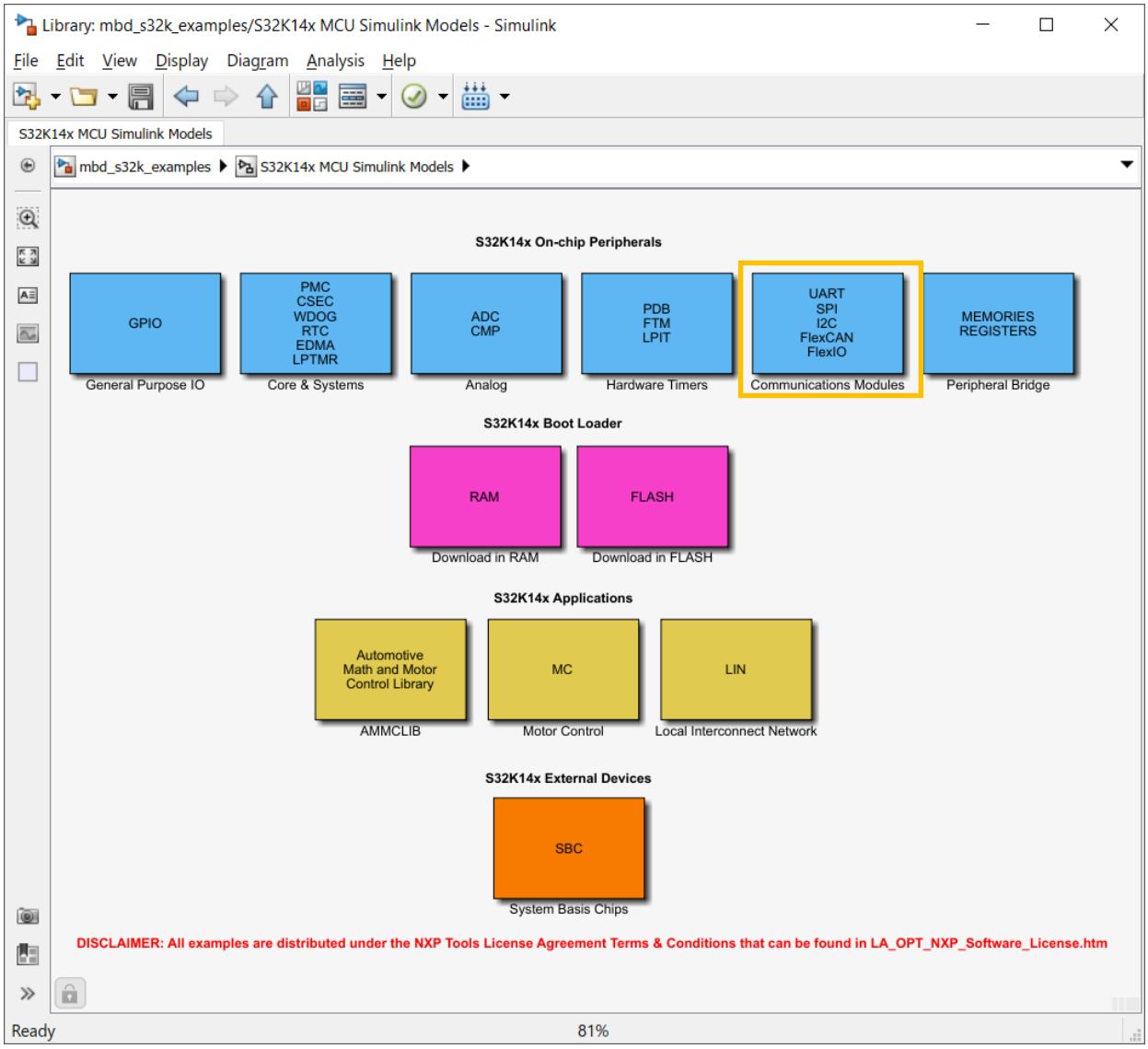
Before running any example on the S32K144EVB-Q100 a proper communication setup between the board and the host PC must be enabled. Please follow the next steps to ensure a working setup:

1. Place J107 jumper on positions 2-3 to power the S32K144EVB via the USB micro connector;
2. Place J104 jumper on positions 2-3 to enable the S12K144 MCU reset;
3. Connect the EVB micro USB connector J7 to a host PC USB connector using supplied USB cable;
4. Allow the PC to automatically configure the USB drivers if needed. Windows OS should automatically detect the S32K144 EVB and should assigned a virtual COM port and a virtual mass storage device to the host;
5. When powered through USB, LEDs D2 and D3 should light green;
6. Once the board is recognized, it should appear as a mass storage device in your PC with the name EVB-S32K144;

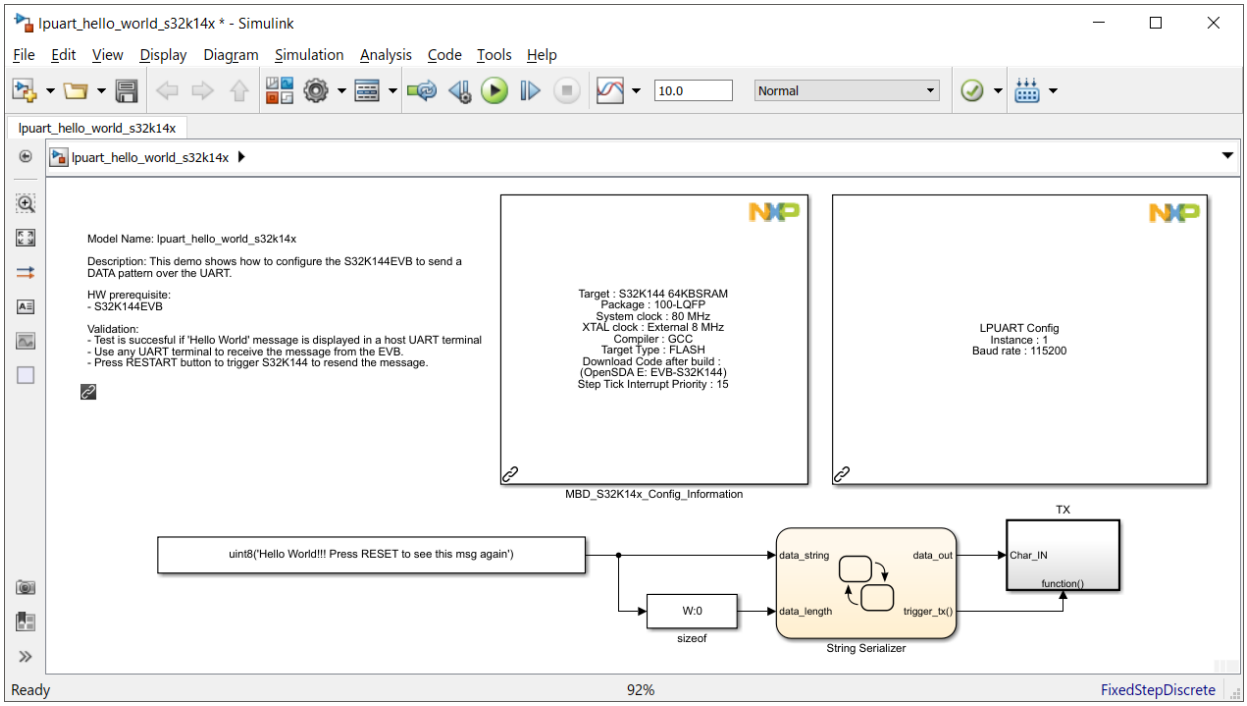
2.3 Hello World Example

If the hardware setup is completed successful and a virtual COM port is created and visible in `Control Panel -> Device Manager -> Port (COM & LPT)` and a virtual mass storage device is present, then all ingredients for running a Model-Based Design Toolbox for S32K1xx Example.

Open the Examples Library and go to Communications Modules for S32K14x MCU and double click on UART Hello World block.

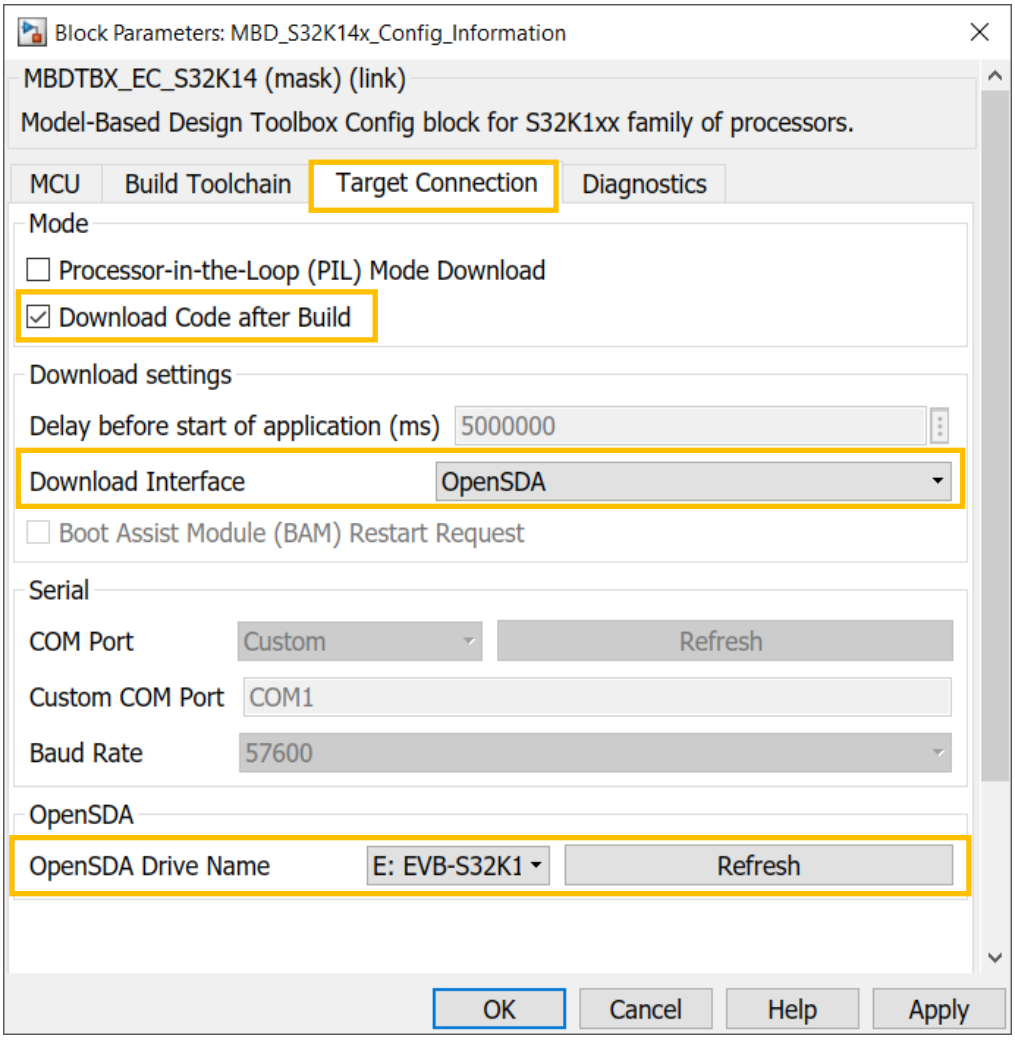


The Simulink `lpuart_hello_world_s32k14x.mdl` model will open. This model programs the S32K144 MCU to sent a message over the UART.


lpuart_hello_world_s32k14x * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
10.0
Normal
lpuart_hello_world_s32k14x
Model Name: lpuart_hello_world_s32k14x
Description: This demo shows how to configure the S32K144EVB to send a DATA pattern over the UART.
HW prerequisite:
- S32K144EVB
Validation:
- Test is succesful if 'Hello World' message is displayed in a host UART terminal
- Use any UART terminal to receive the message from the EVB.
- Press RESTART button to trigger S32K144 to resend the message.
Target : S32K144 64KBSRAM
Package : 100-LQFP
System clock : 80 MHz
XTAL clock : External 8 MHz
Compiler : GCC
Target Type : FLASH
Download Code after build :
(OpenSDA E: EVB-S32K144)
Step Tick Interrupt Priority : 15
MBD_S32K14x_Config_Information
LPUART Config
Instance : 1
Baud rate : 115200
uint8('Hello World!!! Press RESET to see this msg again')
W:0
sizeof
data_string
data_length
data_out
trigger_tx()
String Serializer
TX
Char_IN
function()
Ready
92%
FixedStepDiscrete

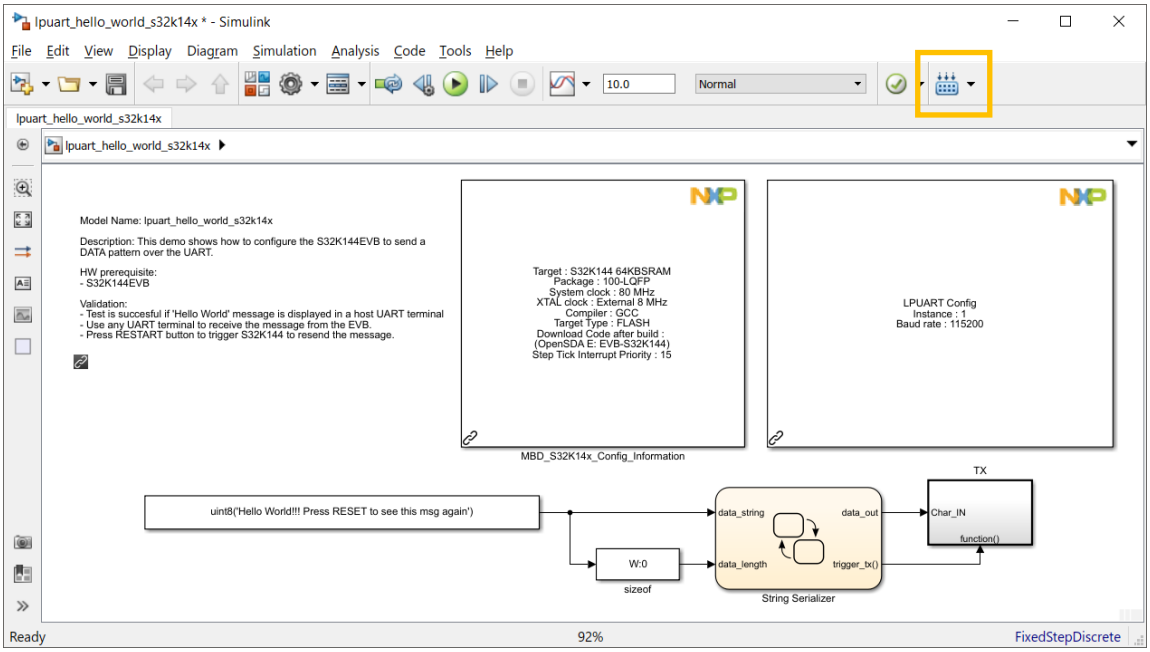
Follow the next steps to run the example:

1. Double click on `MBD_S32K11x_Config_Information` block and setup the `Download Config` parameters according with your PC and HW setup

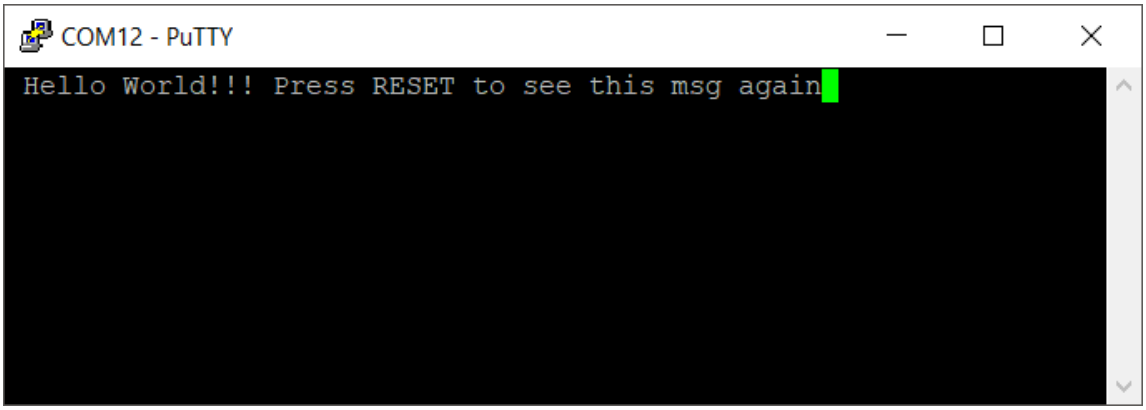


2. `Apply` and close this windows.

3. Press Build Model button and wait until the code is generated, compiled and downloaded to the evaluation board.



4. Open any UART terminal for the virtual COM port assinged and set up the baud rate at 115200, data bits 8 and parity none.
5. Press the reset button on the evaluation board.
6. The S32K144 MCU sends "`Hello World!!! Press RESET to see this msg again`" message over the UART and the UART terminal should display it.



Congradulations! You succedded with running your first example created with Model-Based Design Toolbox for S32K1xx

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

©2019 NXP Semiconductors. All rights reserved.

© 2019 NXP Semiconductors. All rights reserved