Model-Based Design Toolbox S32K1xx Series

Quick Start Guide

An Embedded Target for the S32K1xx Family of Processors
Version 4.2.0

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.

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\lic`
3. Set the Target Compiler Environment Variables
4. Setup the MATLAB path for Model-Based Design Toolbox

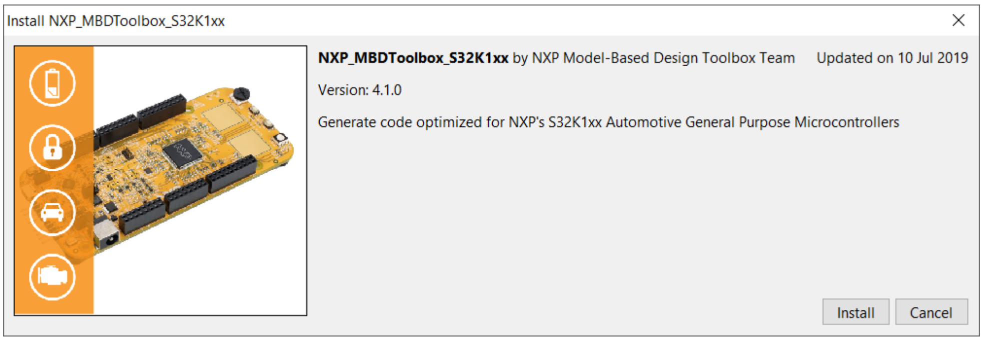
Each of these actions is explained in the following sub-chapters.

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.



Install NXP_MBDToolbox_S32K1xx

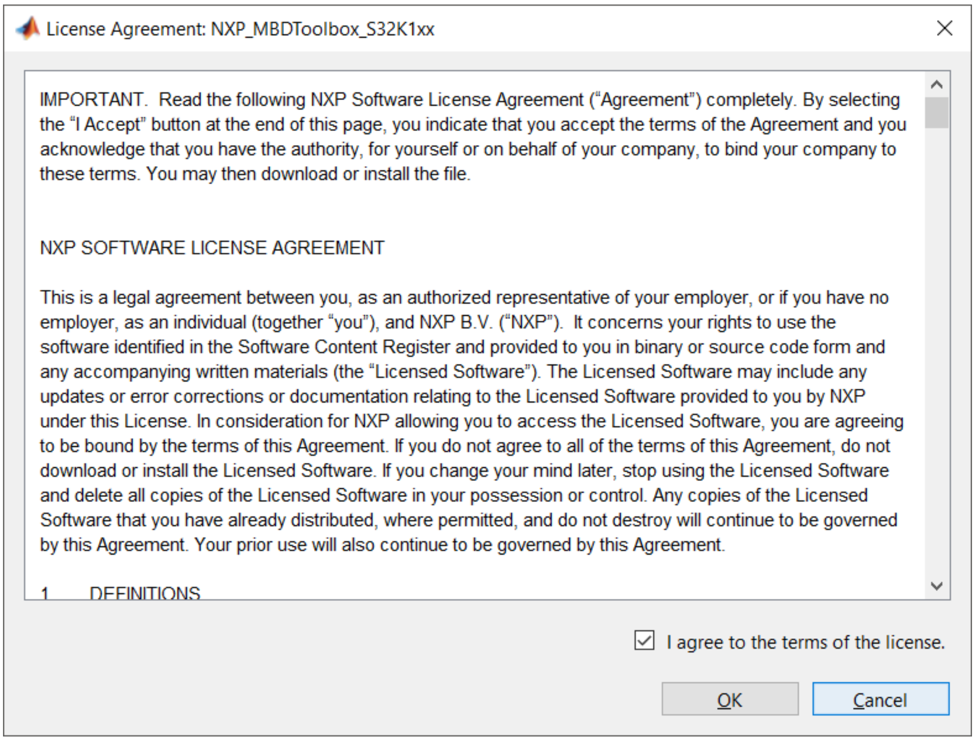
NXP_MBDToolbox_S32K1xx by NXP Model-Based Design Toolbox Team Updated on 10 Jul 2019

Version: 4.1.0

Generate code optimized for NXP's S32K1xx Automotive General Purpose Microcontrollers

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.



License Agreement: NXP_MBDToolbox_S32K1xx

IMPORTANT. Read the following NXP Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, 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.

NXP SOFTWARE LICENSE AGREEMENT

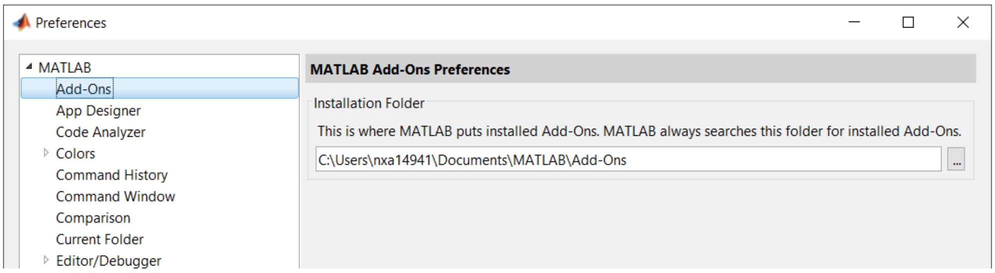
This is a legal agreement between you, as an authorized representative of your employer, or if you have no employer, as an individual (together "you"), and NXP B.V. ("NXP"). It concerns your rights to use the software identified in the Software Content Register and provided to you in binary or source code form and any accompanying written materials (the "Licensed Software"). The Licensed Software may include any updates or error corrections or documentation relating to the Licensed Software provided to you by NXP under this License. In consideration for NXP allowing you to access the Licensed Software, you are agreeing to be bound by the terms of this Agreement. If you do not agree to all of the terms of this Agreement, do not download or install the Licensed Software. If you change your mind later, stop using the Licensed Software and delete all copies of the Licensed Software in your possession or control. Any copies of the Licensed Software that you have already distributed, where permitted, and do not destroy will continue to be governed by this Agreement. Your prior use will also continue to be governed by this Agreement.

1 DEFINITIONS

☑ I agree to the terms of the license.

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 before installation by changing the Add-Ons path from MATLAB Preferences

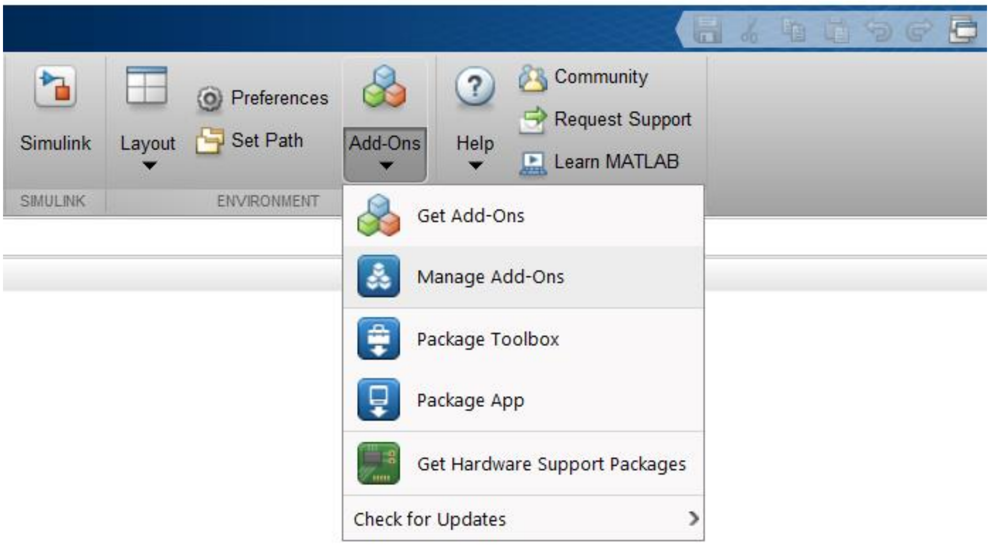


This is where MATLAB puts installed Add-Ons. MATLAB always searches this folder for installed Add-Ons.

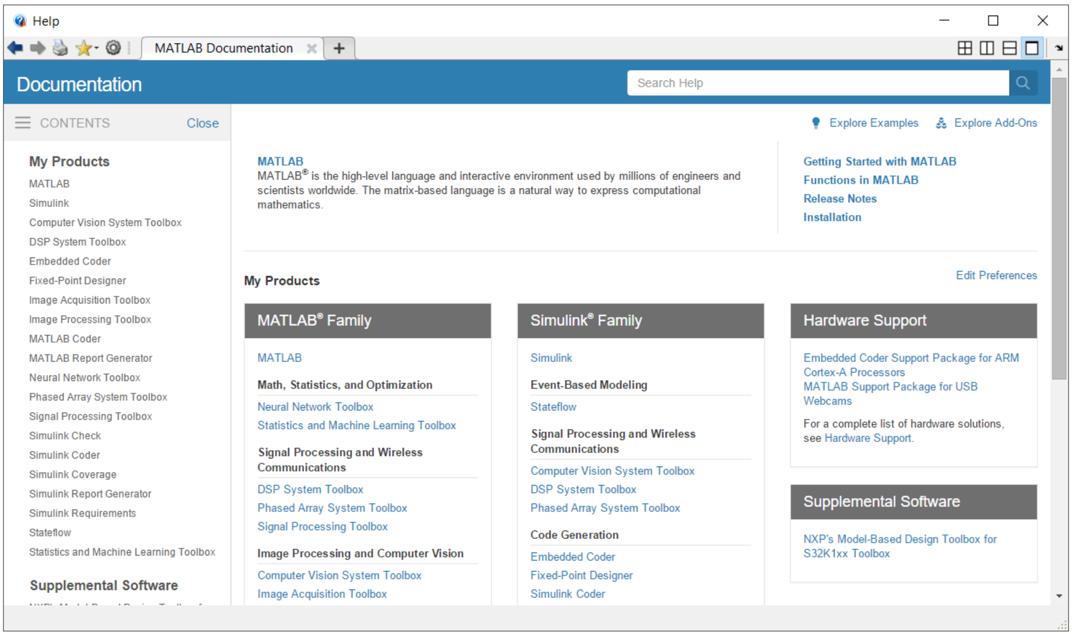
C:\Users\nxa14941\Documents\MATLAB\Add-Ons

   Note: *It is recommended to install the MATLAB and NXP Toolbox into locations that do not contain 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 the MATLAB development environment. Get more details by accessing the standard Help and **Supplemental Software** section



My Products
MATLAB
Simulink
Computer Vision System Toolbox
DSP System Toolbox
Embedded Coder
Fixed-Point Designer
Image Acquisition Toolbox
Image Processing Toolbox
MATLAB Coder
MATLAB Report Generator
Neural Network Toolbox
Phased Array System Toolbox
Signal Processing Toolbox
Simulink Check
Simulink Coder
Simulink Coverage
Simulink Report Generator
Simulink Requirements
Stateflow
Statistics and Machine Learning Toolbox

Supplemental Software

MATLAB
MATLAB® is the high-level language and interactive environment used by millions of engineers and scientists worldwide. The matrix-based language is a natural way to express computational mathematics.

Getting Started with MATLAB
Functions in MATLAB
Release Notes
Installation

My Products Edit Preferences

MATLAB® Family
MATLAB
Math, Statistics, and Optimization
Neural Network Toolbox
Statistics and Machine Learning Toolbox
Signal Processing and Wireless Communications
DSP System Toolbox
Phased Array System Toolbox
Signal Processing Toolbox
Image Processing and Computer Vision
Computer Vision System Toolbox
Image Acquisition Toolbox

Simulink® Family
Simulink
Event-Based Modeling
Stateflow
Signal Processing and Wireless Communications
Computer Vision System Toolbox
DSP System Toolbox
Phased Array System Toolbox
Code Generation
Embedded Coder
Fixed-Point Designer
Simulink Coder

Hardware Support
Embedded Coder Support Package for ARM Cortex-A Processors
MATLAB Support Package for USB Webcams
For a complete list of hardware solutions, see Hardware Support.

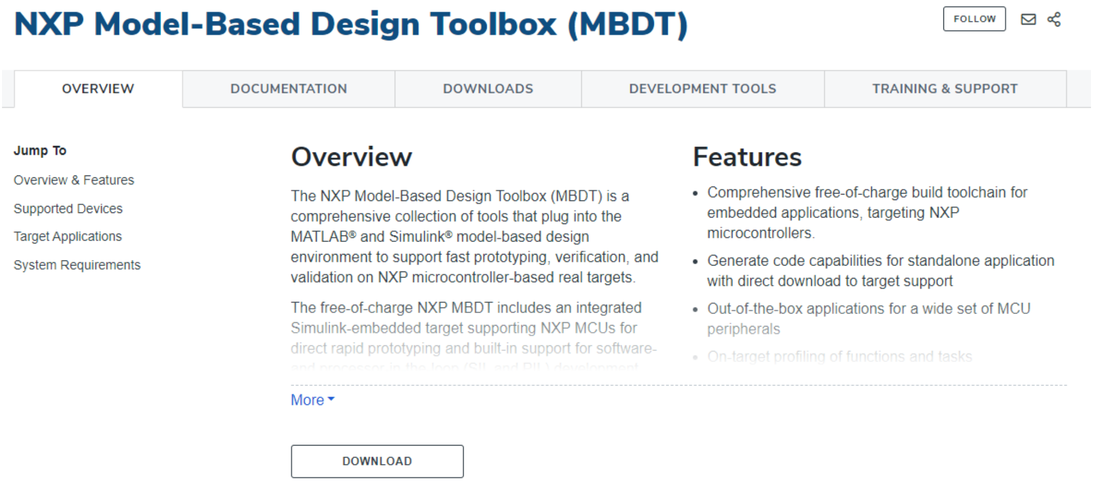
Supplemental Software
NXP's Model-Based Design Toolbox for S32K1xx Toolbox

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`"



NXP Model-Based Design Toolbox (MBDT)

Overview

The NXP Model-Based Design Toolbox (MBDT) is a comprehensive collection of tools that plug into the MATLAB® and Simulink® model-based design environment to support fast prototyping, verification, and validation on NXP microcontroller-based real targets.

The free-of-charge NXP MBDT includes an integrated Simulink-embedded target supporting NXP MCUs for direct rapid prototyping and built-in support for software- and processor-in-the-loop (SIL and PIL) development

Features

- Comprehensive free-of-charge build toolchain for embedded applications, targeting NXP microcontrollers.
- Generate code capabilities for standalone application with direct download to target support
- Out-of-the-box applications for a wide set of MCU peripherals
- On-target profiling of functions and tasks

2. Once you have logged in, the "`Product Information`" page for the 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 the "`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\lic`"

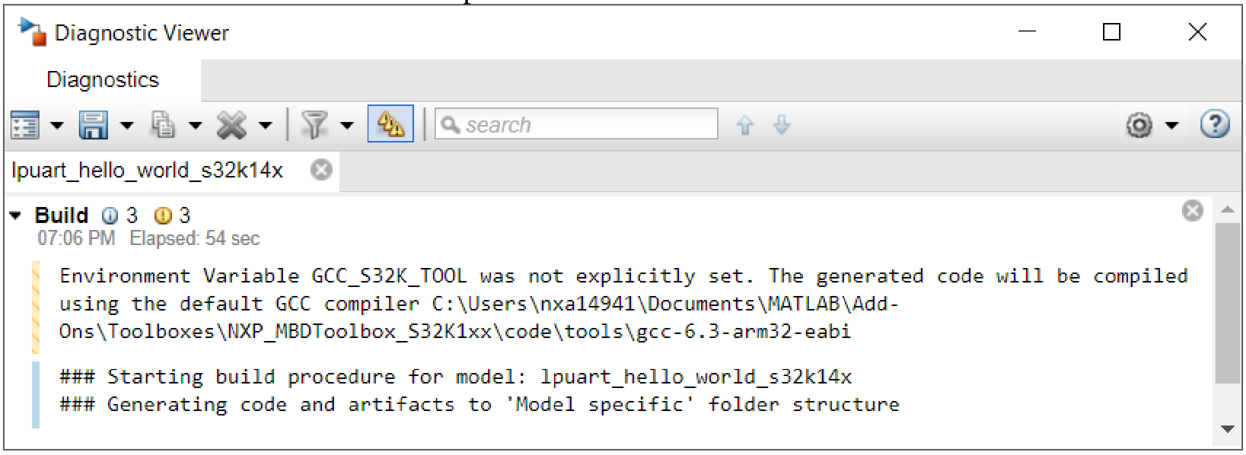
1.2.3 Setting up the Target Compilers

The target compiler for the 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.



```
Environment Variable GCC_S32K_TOOL was not explicitly set. The generated code will be compiled
using the default GCC compiler C:\Users\nxa14941\Documents\MATLAB\Add-
Ons\Toolboxes\NXP_MBDToolbox_S32K1xx\code\tools\gcc-6.3-arm32-eabi

### Starting build procedure for model: lpuart_hello_world_s32k14x
### Generating code and artifacts to 'Model specific' folder structure
```

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\`) and running the "`mbd_s32k_path`" script.

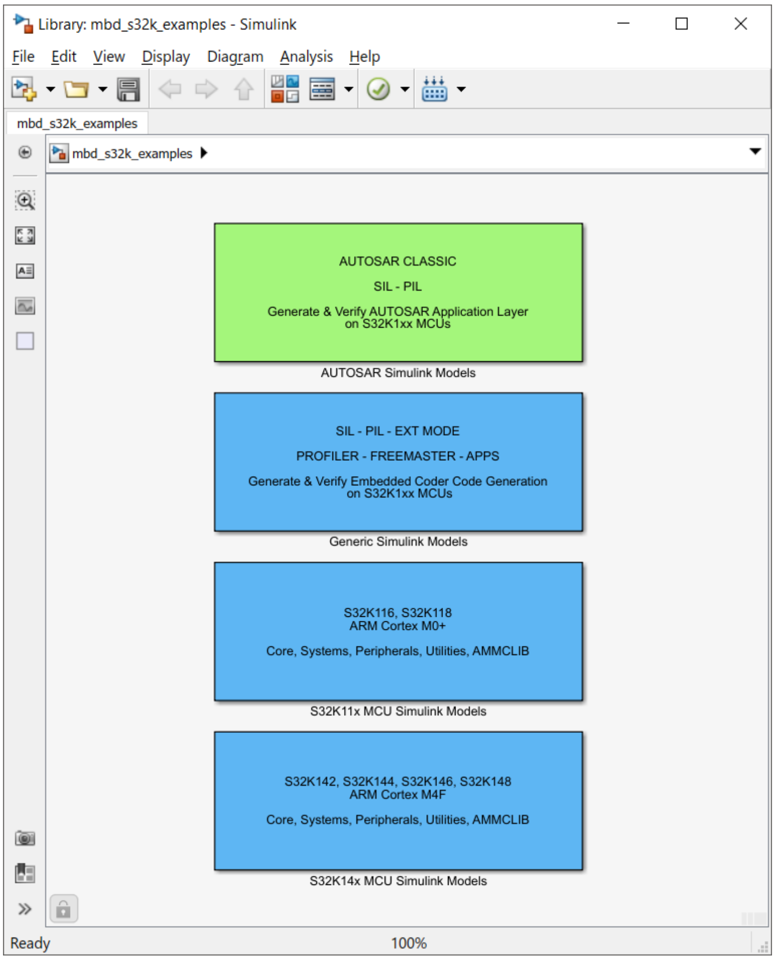
```
>> mbd_s32k_path
Treating                        'C:\Users\b21307\Documents\MATLAB\Add-
Ons\Toolboxes\NXP_MBDToolbox_S32K1xx\code' as MBD Toolbox installation
root.
MBD Toolbox path prepended.
Successful.
>>
```

2 Run Models

2.1 Examples Library & Help

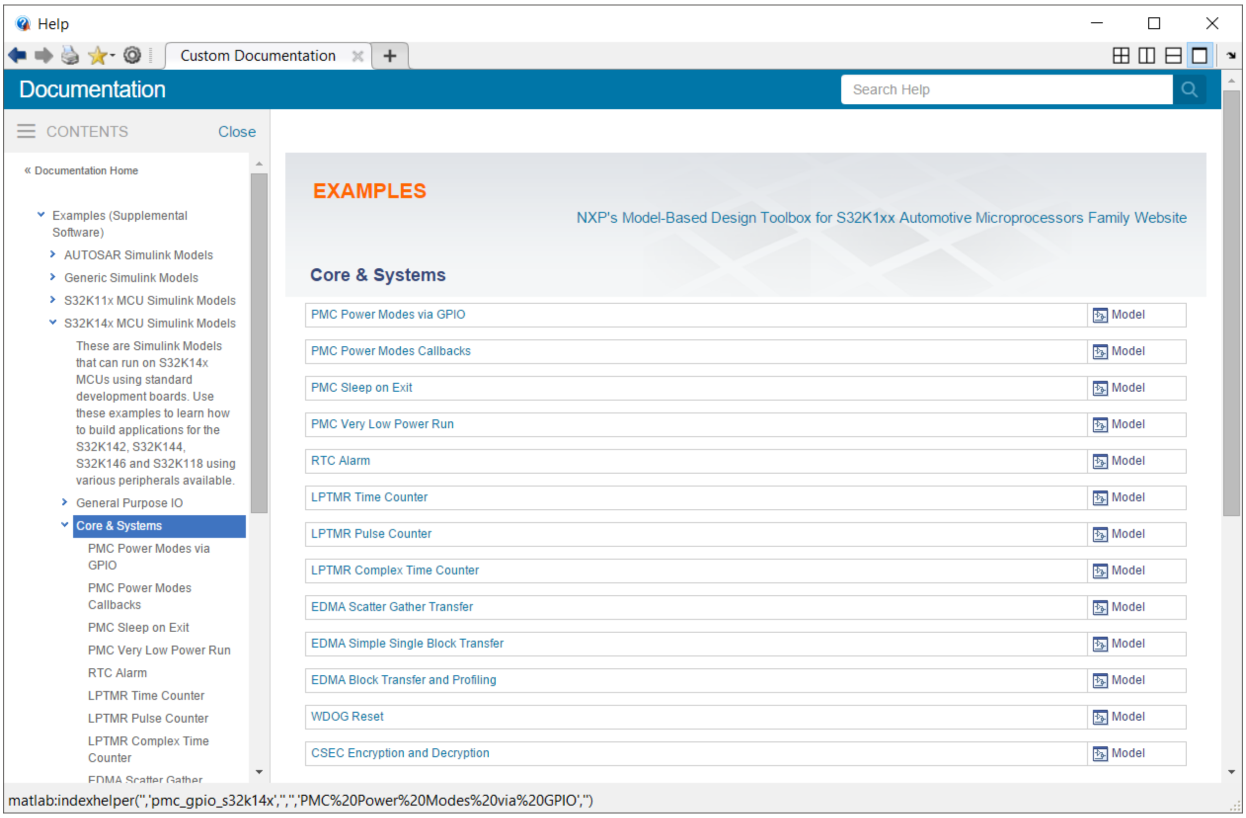
NXP’s Model-Based Design Toolbox comes with an Examples Library collection that lets you test different MCU on-chip modules and run complex applications.

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


Library: mbd_s32k_examples - Simulink
File Edit View Display Diagram Analysis Help
mbd_s32k_examples
mbd_s32k_examples
AUTOSAR CLASSIC
SIL - PIL
Generate & Verify AUTOSAR Application Layer on S32K1xx MCUs
AUTOSAR Simulink Models
SIL - PIL - EXT MODE
PROFILER - FREEMASTER - APPS
Generate & Verify Embedded Coder Code Generation on S32K1xx MCUs
Generic Simulink Models
S32K116, S32K118
ARM Cortex M0+
Core, Systems, Peripherals, Utilities, AMMCLIB
S32K11x MCU Simulink Models
S32K142, S32K144, S32K146, S32K148
ARM Cortex M4F
Core, Systems, Peripherals, Utilities, AMMCLIB
S32K14x MCU Simulink Models
Ready
100%

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

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



- 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
      - EDMA Scatter Gather

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

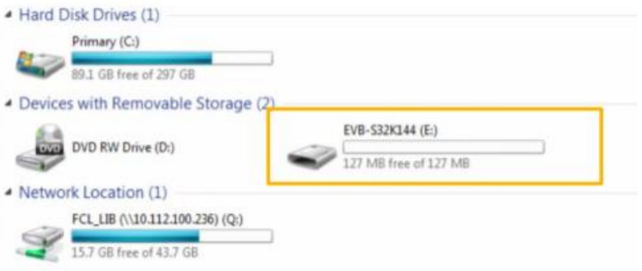
```
matlab:indexhelper('','pmc_gpio_s32k14x','','','PMC%20Power%20Modes%20via%20GPIO','')
```

2.2 Hardware Setup

All examples provided with the Model-Based Design Toolbox were developed on S32K144EVB-Q100 as the 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 assign 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;



Hard Disk Drives (1)

Primary (C:)
89.1 GB free of 297 GB

Devices with Removable Storage (2)

DVD RW Drive (D:)

EVB-S32K144 (E:)
127 MB free of 127 MB

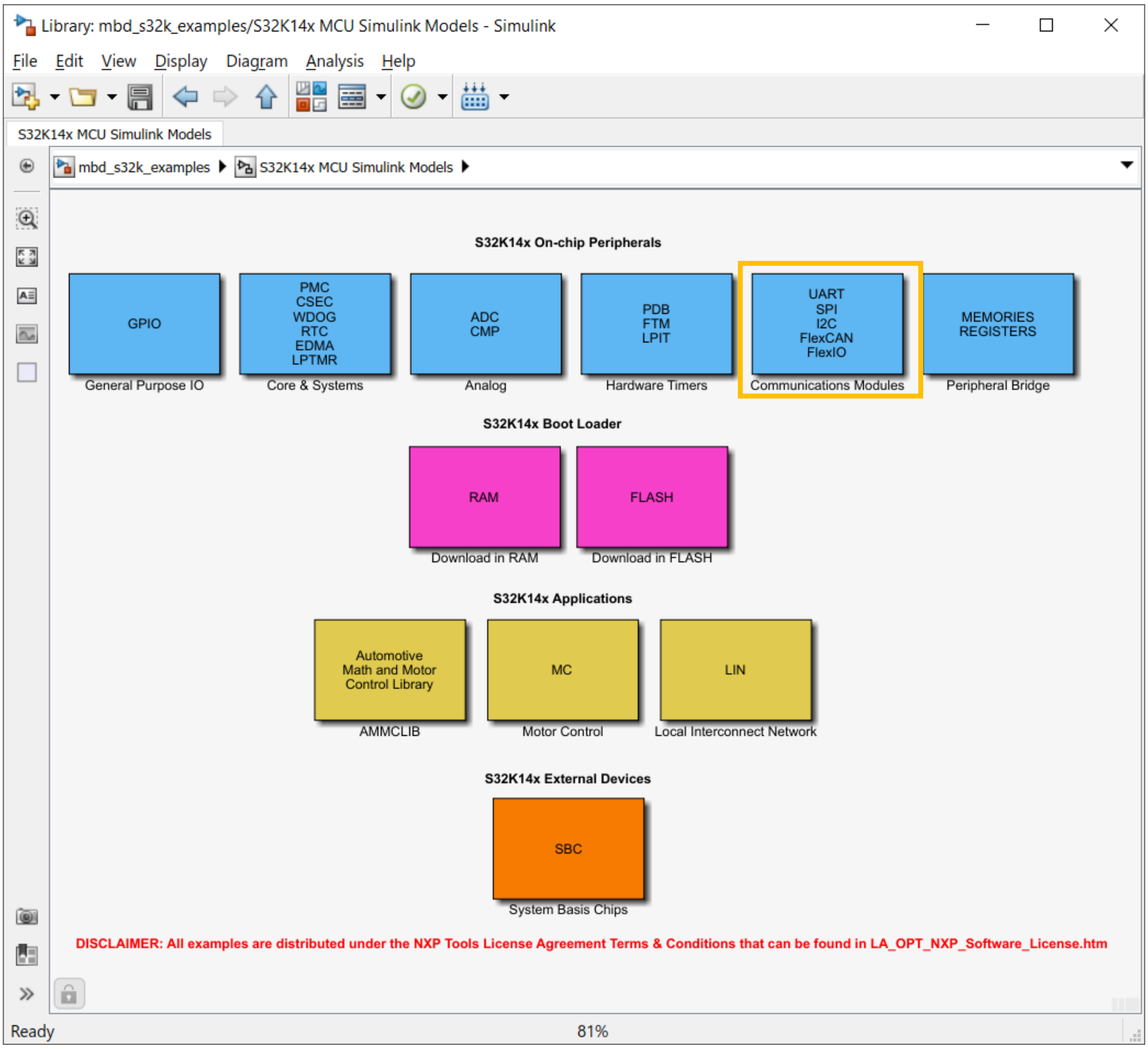
Network Location (1)

FCL_LIB (\\10.112.100.236) (Q:)
15.7 GB free of 43.7 GB

2.3 Hello World Example

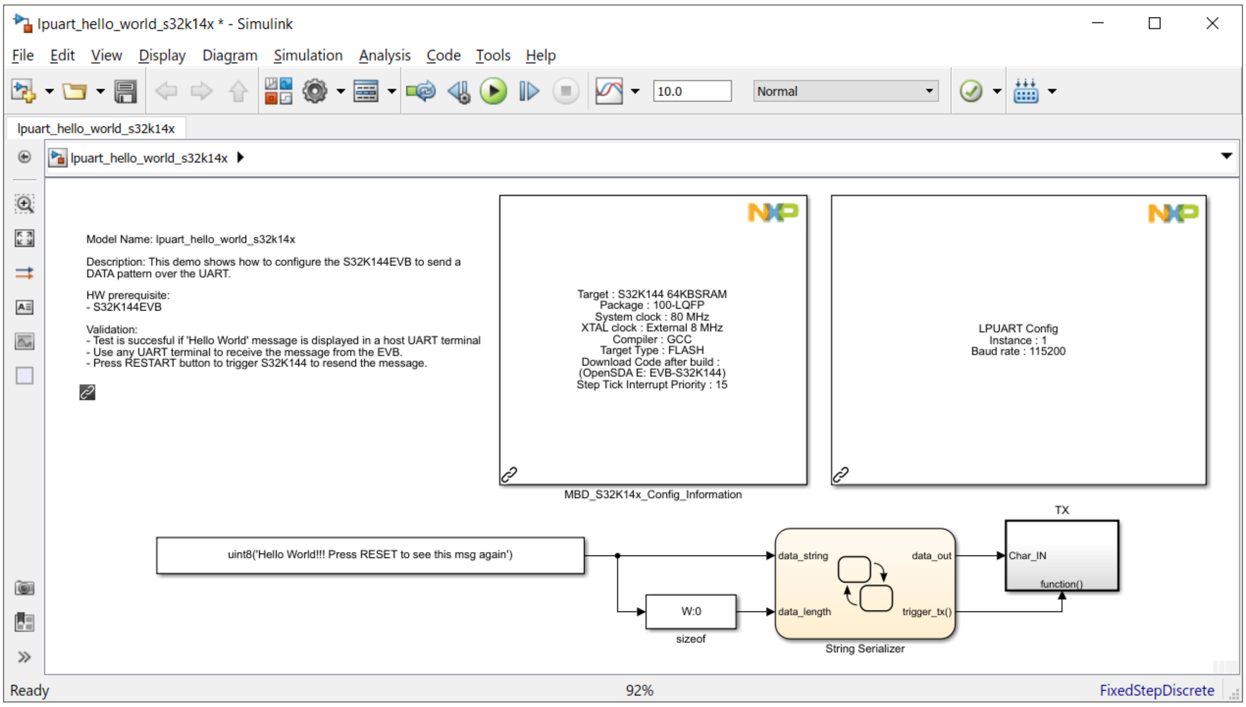
If the hardware setup is completed successfully and a virtual COM port is created and visible in `Control Panel -> Device Manager -> Port (COM & LPT)` and a virtual mass storage device are present, then all ingredients for running a Model-Based Design Toolbox for S32K1xx Example.

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



DISCLAIMER: All examples are distributed under the NXP Tools License Agreement Terms & Conditions that can be found in LA_OPT_NXP_Software_License.htm

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



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.

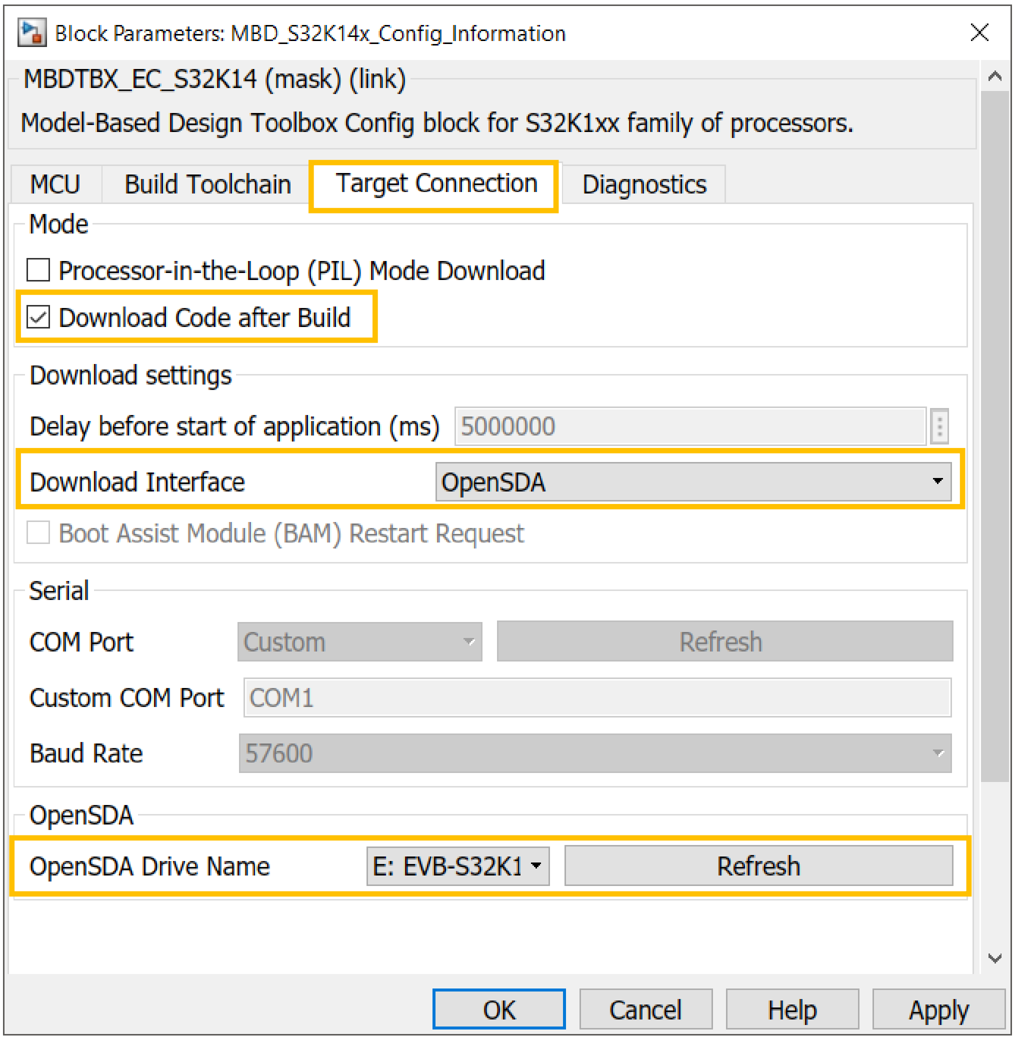
Target : S32K144 64KBSRAM
Package : 100-LQFP
System clock : 80 MHz
XTAL clock : External 8 MHz
Compiler : GCC
Target Type : FLASH
Download Code after build :
(OpenSDA E: EVB-S32K144)
Step Tick Interrupt Priority : 15

MBD_S32K14x_Config_Information

LPUART Config
Instance : 1
Baud rate : 115200

Follow the next steps to run the example:

1. Double click on `MBD_S32K1xx_Config_Information` block and set up the `Download Config` parameters according to with your PC and HW setup



Block Parameters: MBD_S32K14x_Config_Information

MBDTBX_EC_S32K14 (mask) (link)

Model-Based Design Toolbox Config block for S32K1xx family of processors.

MCU | Build Toolchain | Target Connection | Diagnostics

Mode

☐ Processor-in-the-Loop (PIL) Mode Download

☑ Download Code after Build

Download settings

Delay before start of application (ms): 5000000

Download Interface: OpenSDA

☐ Boot Assist Module (BAM) Restart Request

Serial

COM Port: Custom — Refresh

Custom COM Port: COM1

Baud Rate: 57600

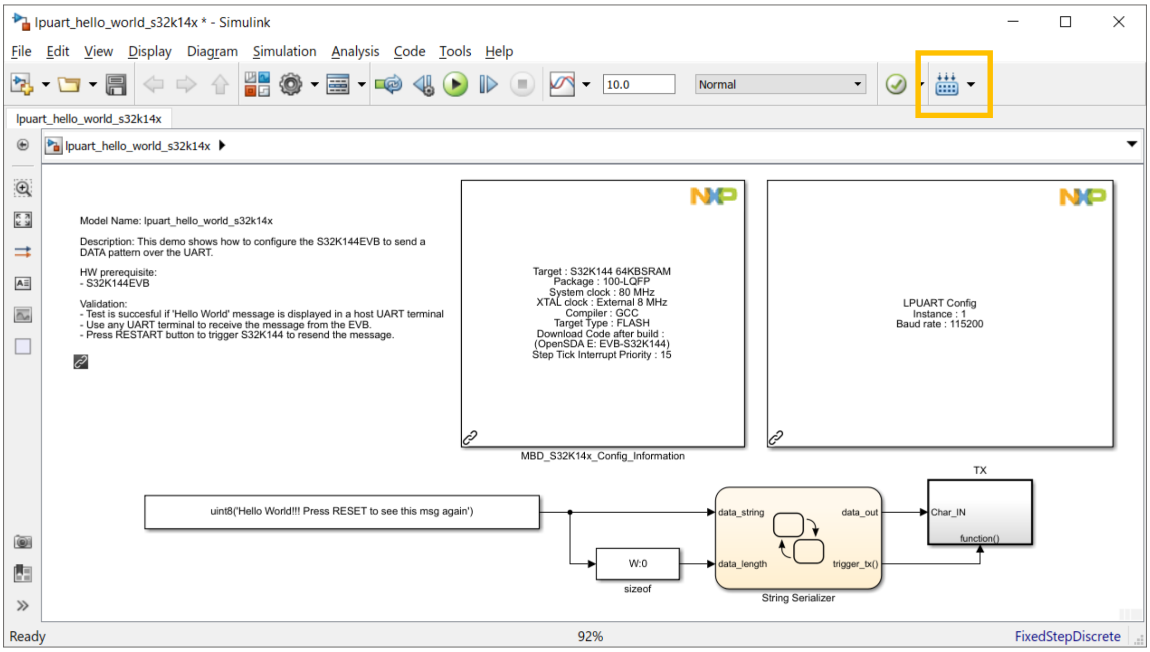
OpenSDA

OpenSDA Drive Name: E: EVB-S32K1 — Refresh

OK | Cancel | Help | Apply

2. `Apply` and close this window.

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



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

LPUART Config
Instance : 1
Baud rate : 115200

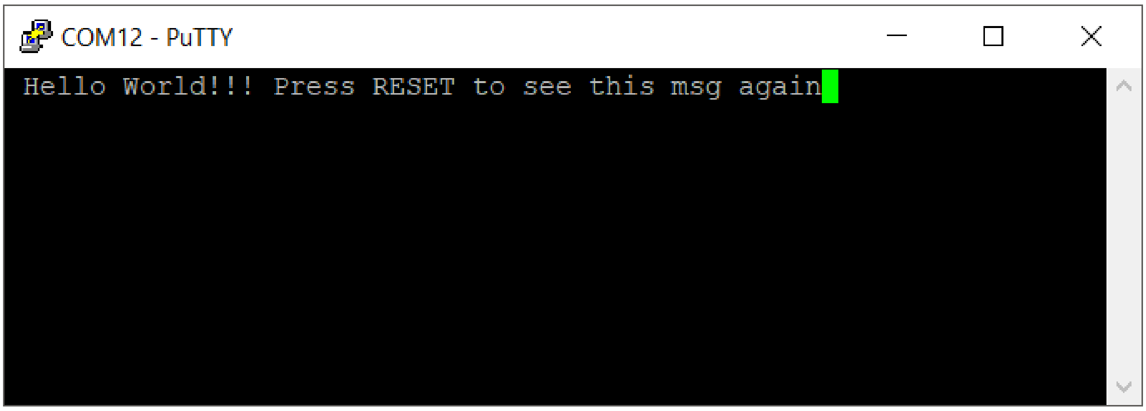
uint8('Hello World!!! Press RESET to see this msg again')

lpuart_hello_world_s32k14x * - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
10.0
Normal
lpuart_hello_world_s32k14x
lpuart_hello_world_s32k14x
MBD_S32K14x_Config_Information
TX
data_string
data_out
Char_IN
function()
W:0
sizeof
data_length
trigger_tx()
String Serializer
Ready
92%
FixedStepDiscrete

4. Open any UART terminal for the virtual COM port assigned 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.



```
Hello World!!! Press RESET to see this msg again
```

Congratulations! You succeeded 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