

Model-Based Design Toolbox MPC57xx Series

Quick Start Guide

An Embedded Target for the MPC57xx Family of Processors
Version 3.0.0

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



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 | 2-10 |
| 2.1 | Examples Library & Help..... | 2-10 |
| 2.2 | Hardware Setup | 2-12 |
| 2.3 | BootLoader Setup | 2-13 |
| 2.4 | Hello World Example | 2-17 |

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 `..\My Documents\MATLAB\Add-Ons\Toolboxes\NXP_MBDToolbox_MPC57xx\code\lic`
3. Set the Target Compiler Environment Variables
4. Setup the MATLAB path for Model-Based Design Toolbox

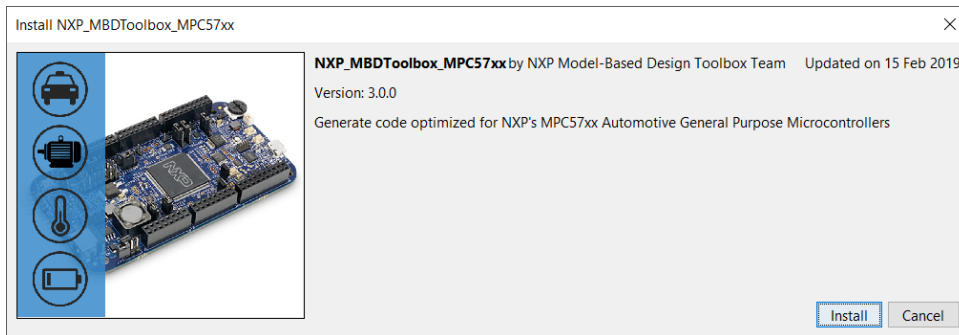
Each of these actions are explained in the following sub-chapters.

1.2.1 Run Add-on installer

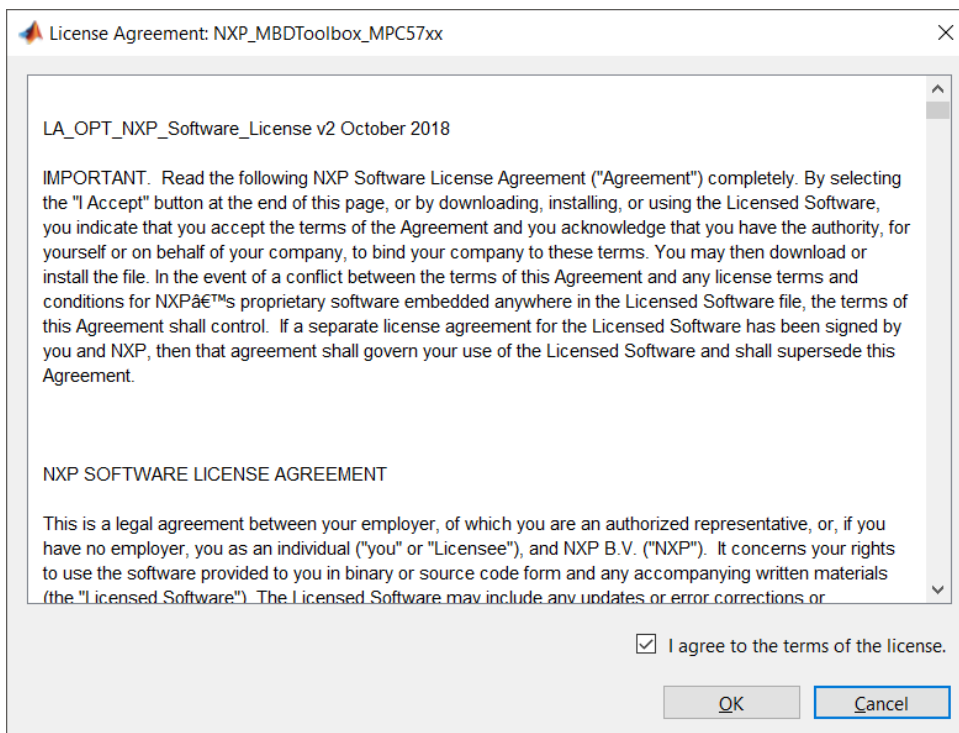
Install the NXP's Model-Based Design Toolbox by double clicking the *.mltbx file. This will activate the MATLAB Add-ons installer that will automatically start the installation process.

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.

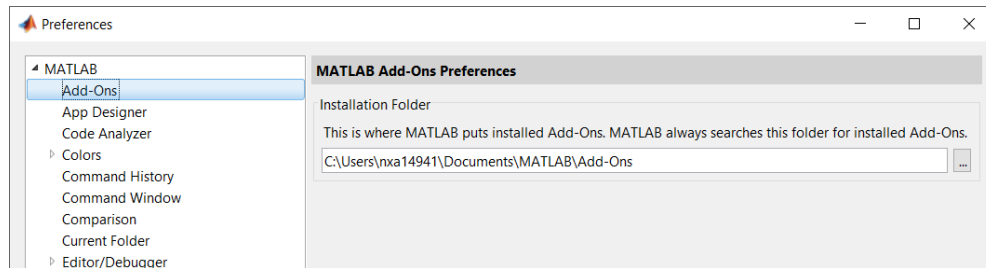


2. Indicate acceptance of the NXP Software License Agreement by selecting "I agree to the terms of the license" to proceed.

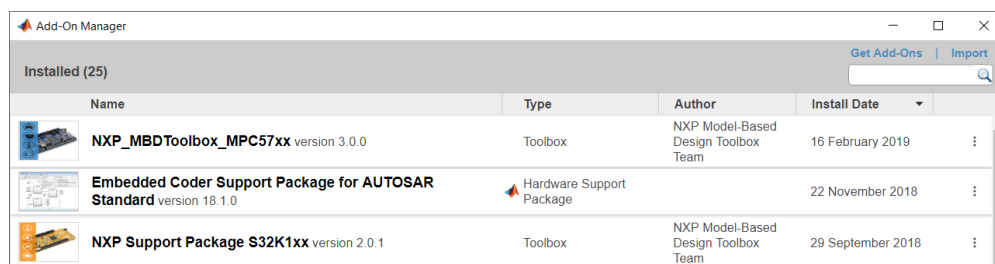
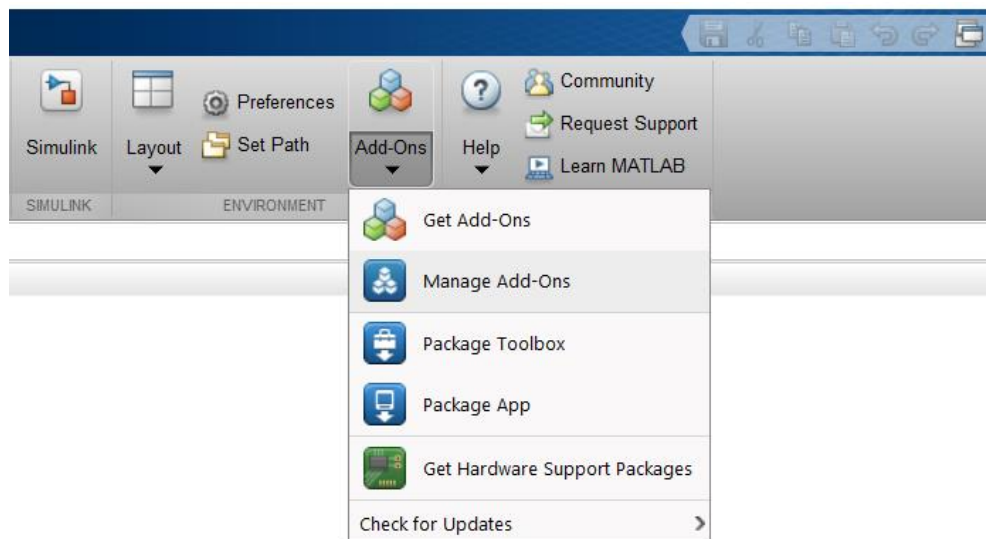


- Click “OK” to start the MATLAB installation process. The rest of the process is silent and under MATLAB control. All the files will be automatically copied into default Add-Ons folder within the MATLAB

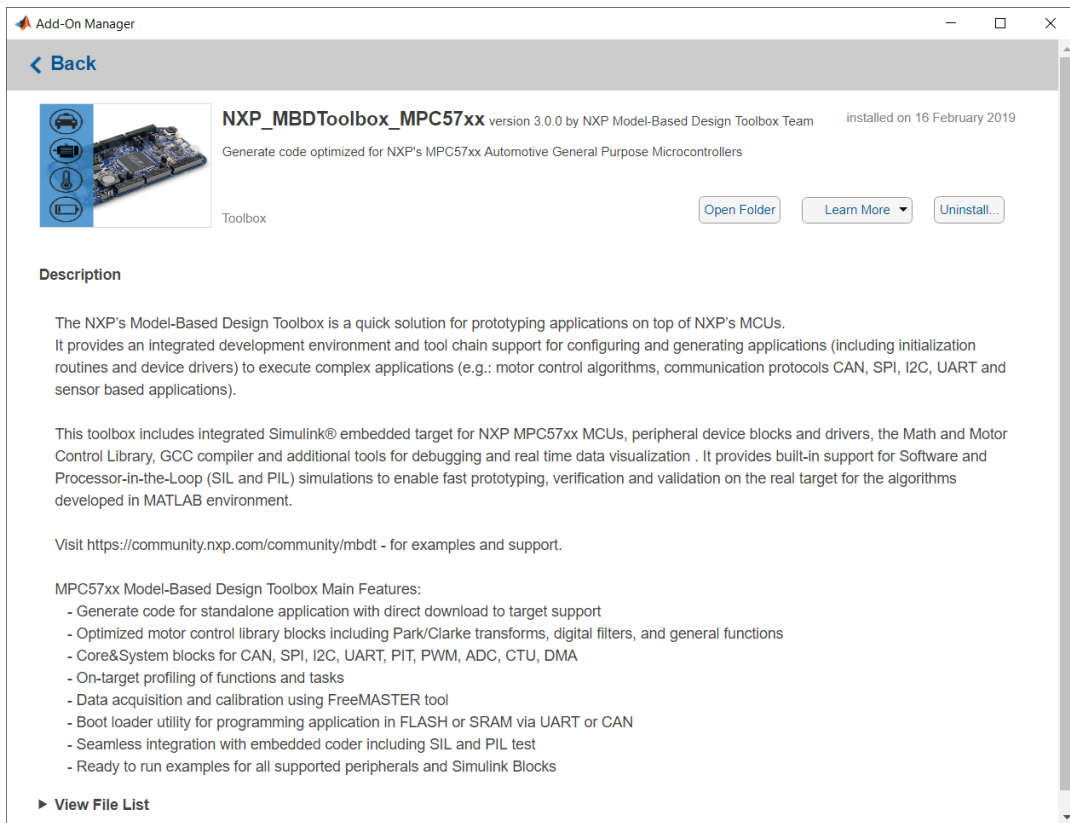
The default location can be changed prior to installation by changing the Add-Ons path from MATLAB Preferences



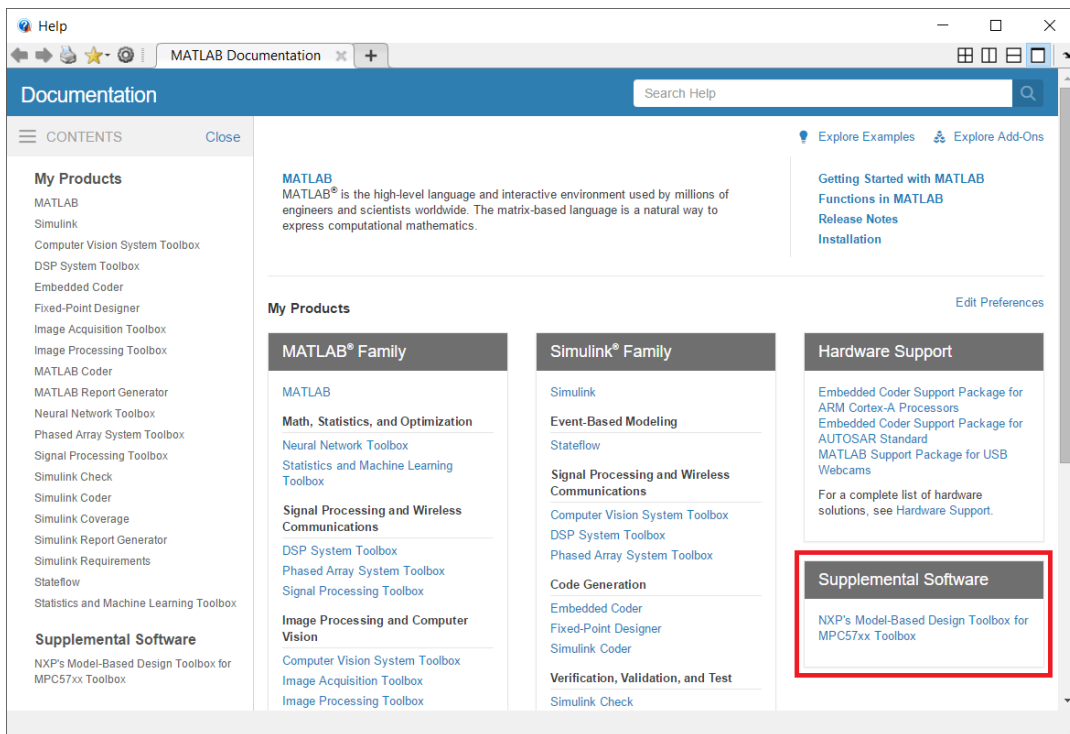
- After a couple of minutes (2-5min), the NXP’s Model-Based Design Toolbox should be visible as a new Add-ons.



- More details about the NXP’s Model-Based Design Toolbox can be found by clicking on View Details



6. 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

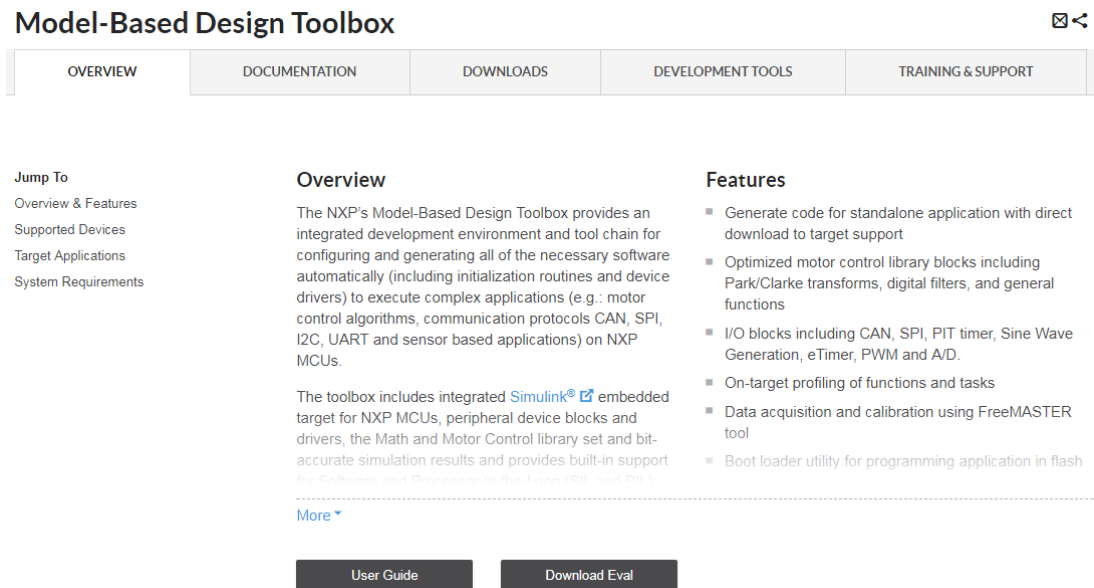


1.2.2 License Registration & Installation

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”



The screenshot shows the 'Model-Based Design Toolbox' website. At the top, there is a navigation bar with tabs for 'OVERVIEW', 'DOCUMENTATION', 'DOWNLOADS', 'DEVELOPMENT TOOLS', and 'TRAINING & SUPPORT'. The 'OVERVIEW' tab is selected. Below the navigation bar, there is a 'Jump To' section with links for 'Overview & Features', 'Supported Devices', 'Target Applications', and 'System Requirements'. The main content area is divided into three columns: 'Overview', 'Features', and 'User Guide'. The 'Overview' column contains text describing the toolbox's capabilities and its integration with Simulink. The 'Features' column lists several key features, such as code generation for standalone applications, optimized motor control blocks, and data acquisition tools. At the bottom of the page, there are two buttons: 'User Guide' and '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 License Installation** 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_MPC57xx\code\lic”

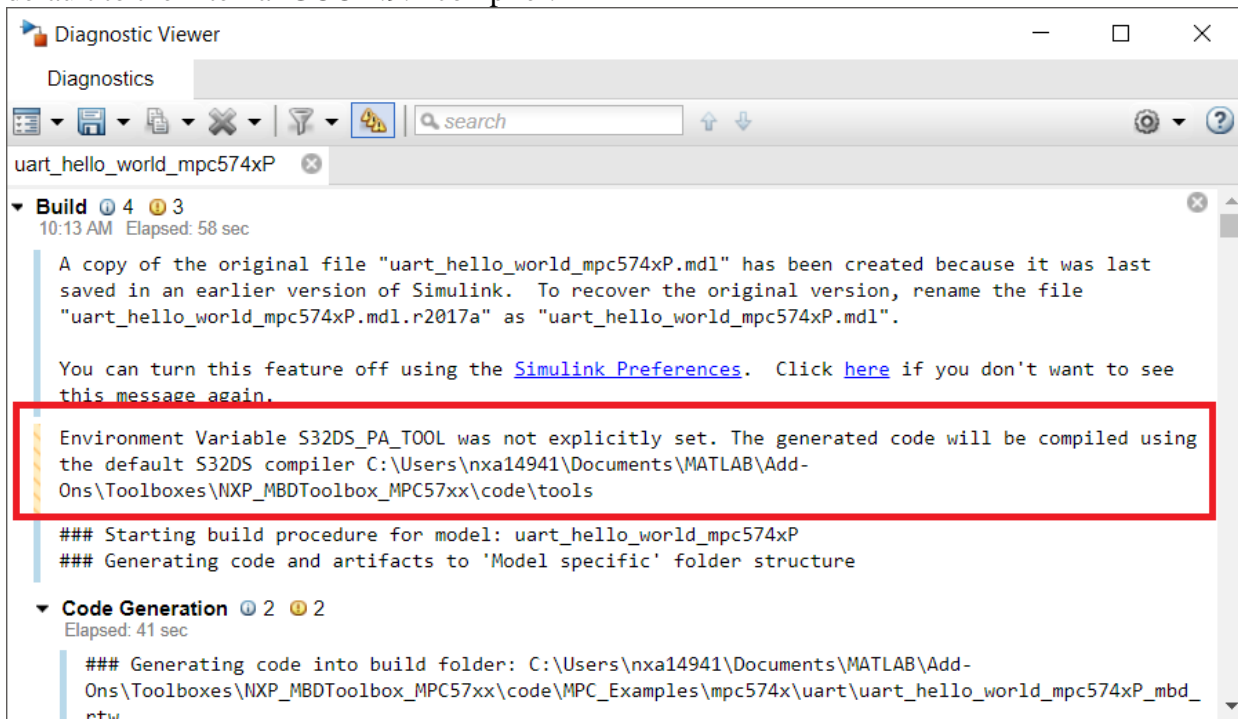
1.2.3 Setting up the Target Compilers

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:

```
S32DS_PA_TOOL = {Toolbox installation path}/tools or  
                {S32 Design Studio installation path}/  
GHS_TOOL = {GHS installation path}/comp_201516/  
DIAB_TOOL = {DIAB installation path}/diab/5.9.6.2/
```

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 4.9.4 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 MATLAB 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_MPC57xx\code) and running the "mbd_mpc_path" script.


```
>> mbd_mpc_path
Treating 'C:\Users\xxxxx\Documents\MATLAB\Add-
Ons\Toolboxes\NXP_MBDToolbox_MPC57xx\code' as MBD Toolbox installation
root.
MBD Toolbox path prepended.
Successful.
>>
```

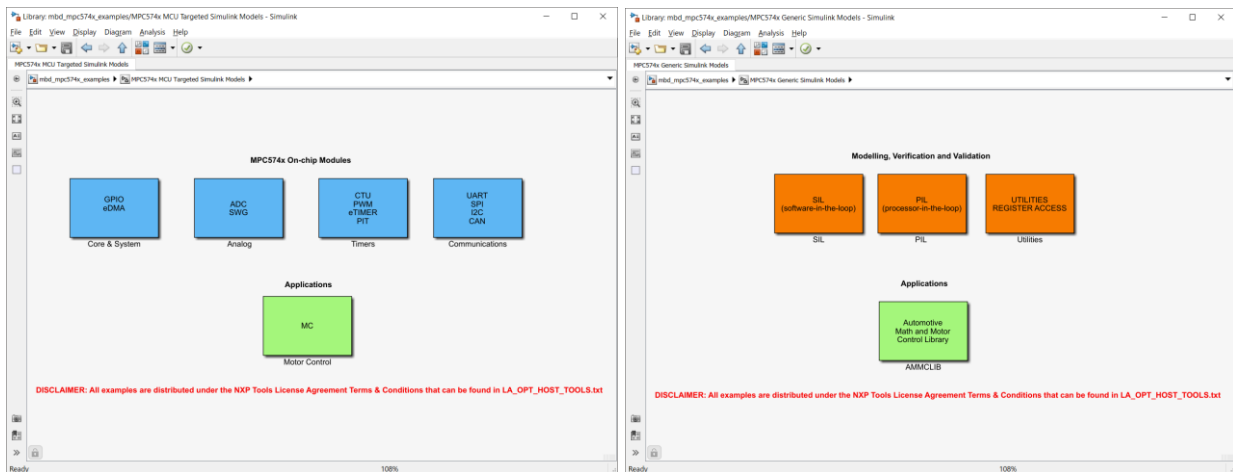
2 Run Models

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_mpc574x_examples.mdl` can be opened from “{Model Based Design Install Directory}\MPC_Examples\” folder.

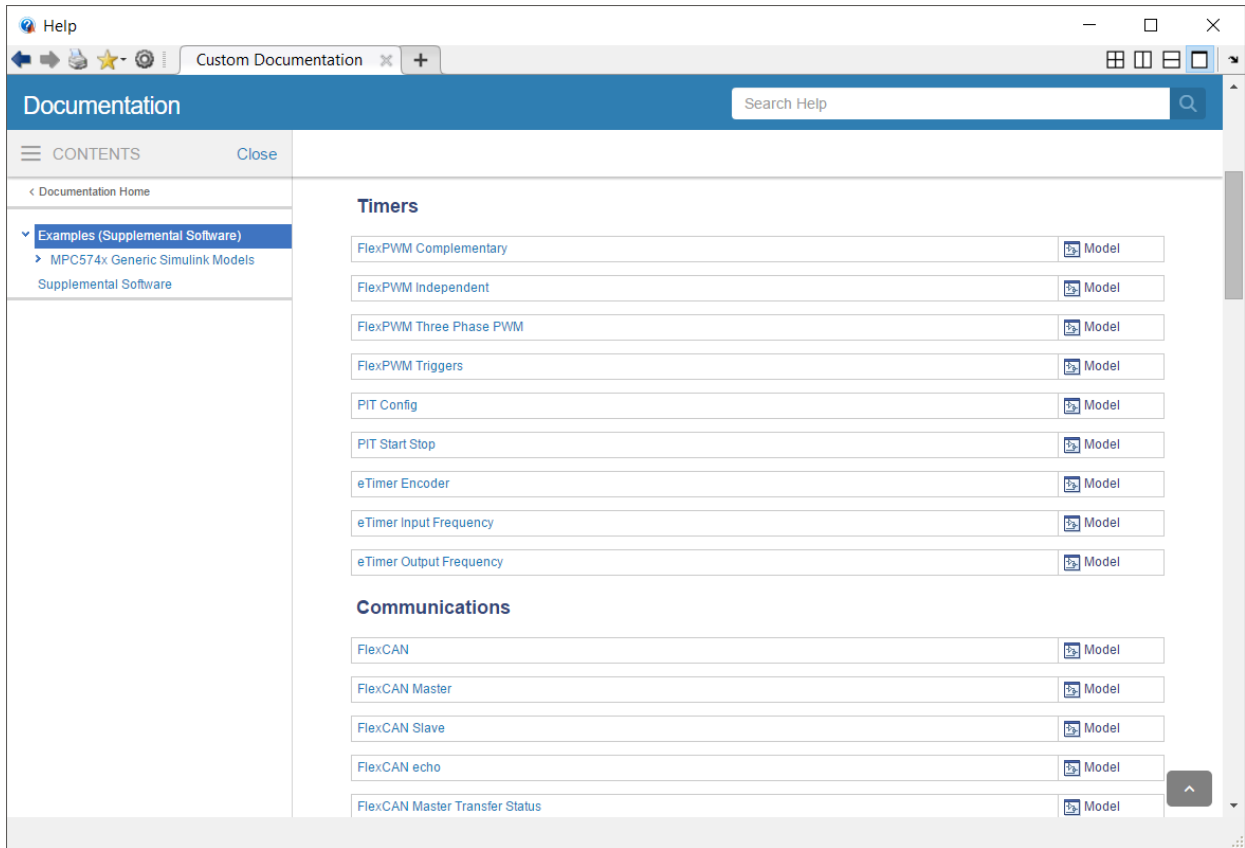
Note: the example library is divided into two sections: custom & generic examples. The generic examples can be executed on all the processors and boards supported with none or minimal changes. The custom examples are targeted for a specific processor and board and are highly dependent on peripherals and pins available.



Each category contains multiple examples that showcase different Model-Based Design Toolbox capabilities that are clarified in three layers:

- **On-chip modules** and peripherals examples covers the basic functionalities of the MCU;
- **Modelling, Verification and Validation** examples covers the SIL, PIL and additional blocks supported by toolbox to help faster prototyping;
- **Applications** covers the AMMCLIB and Motor Control scenarios;

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



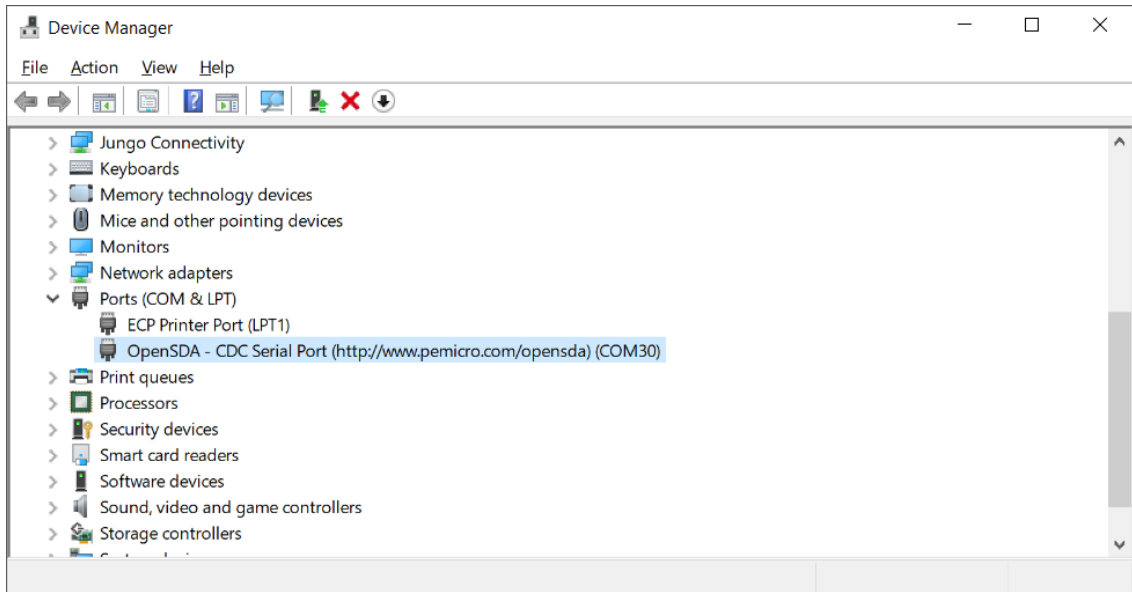
2.2 Hardware Setup

All examples provided with the Model-Based Design Toolbox were developed on DEVKIT-MPC5744P as primary target. Additional information about this development kit can be found on NXP official web page [here](#).



Before running any example on the DEVKIT-MPC5744P 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 J13 jumper on positions 2-3 to power the DEVKIT-MPC5744P via the USB micro connector;
2. Connect the EVB micro USB connector J12 to a host PC USB connector using supplied USB cable;
3. Allow the PC to automatically configure the USB drivers if needed. Windows OS should automatically detect the DEVKIT-MPC5744P and should assigned a virtual COM port to the host;

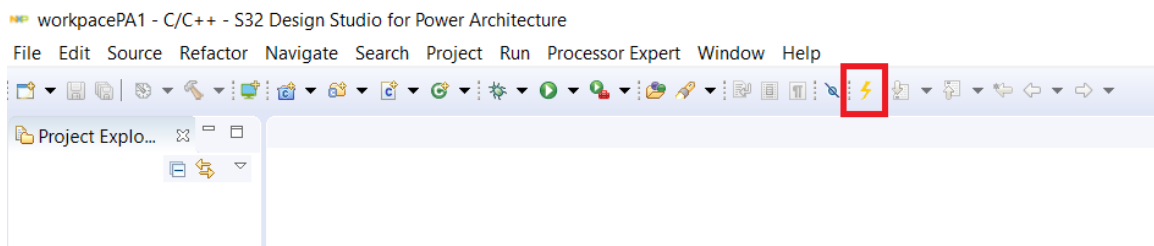


4. When powered through USB, LEDs DS1, DS2 and DS5 should light green;

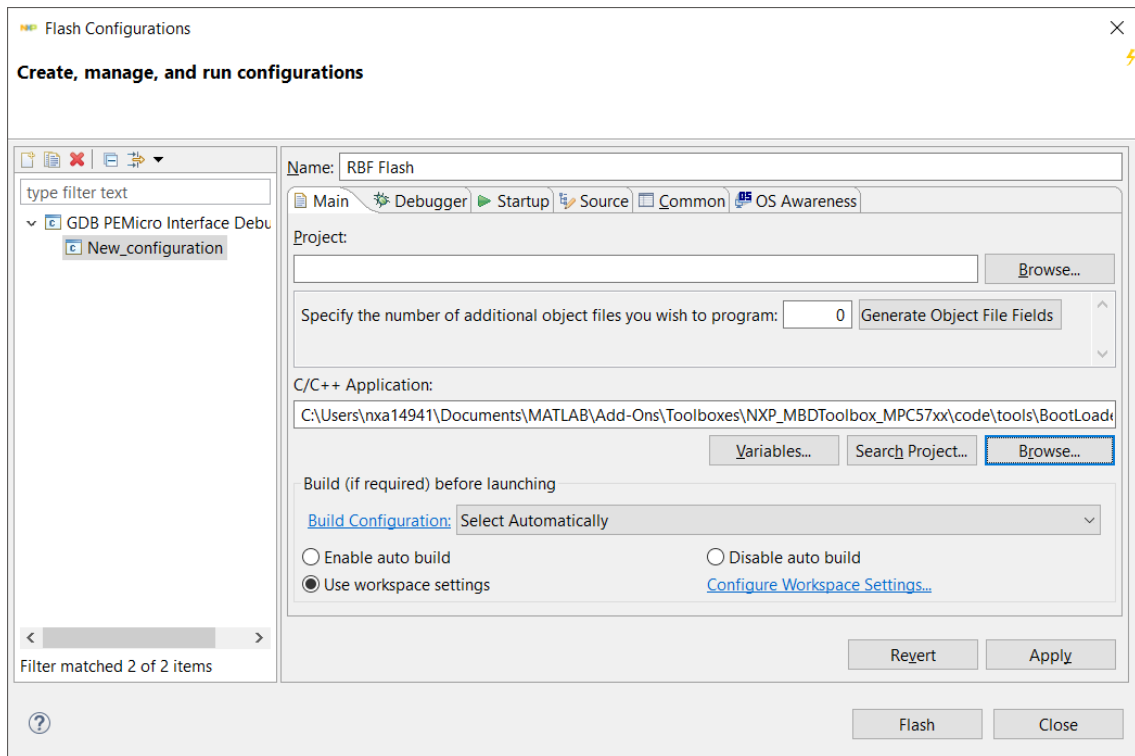
2.3 BootLoader Setup

To be able to download the applications from MATLAB/Simulink to the target we need to program a special application called RBF into the MCU flash memory. For the first time programming you need to have the [S32 Design Studio for PA IDE](#) installed. Follow the next steps to flash the RBF image to the processor flash memory.

1. Click on the **Flash icon** to open the wizard that allows you to connect with the target and flash the boot loader algorithm

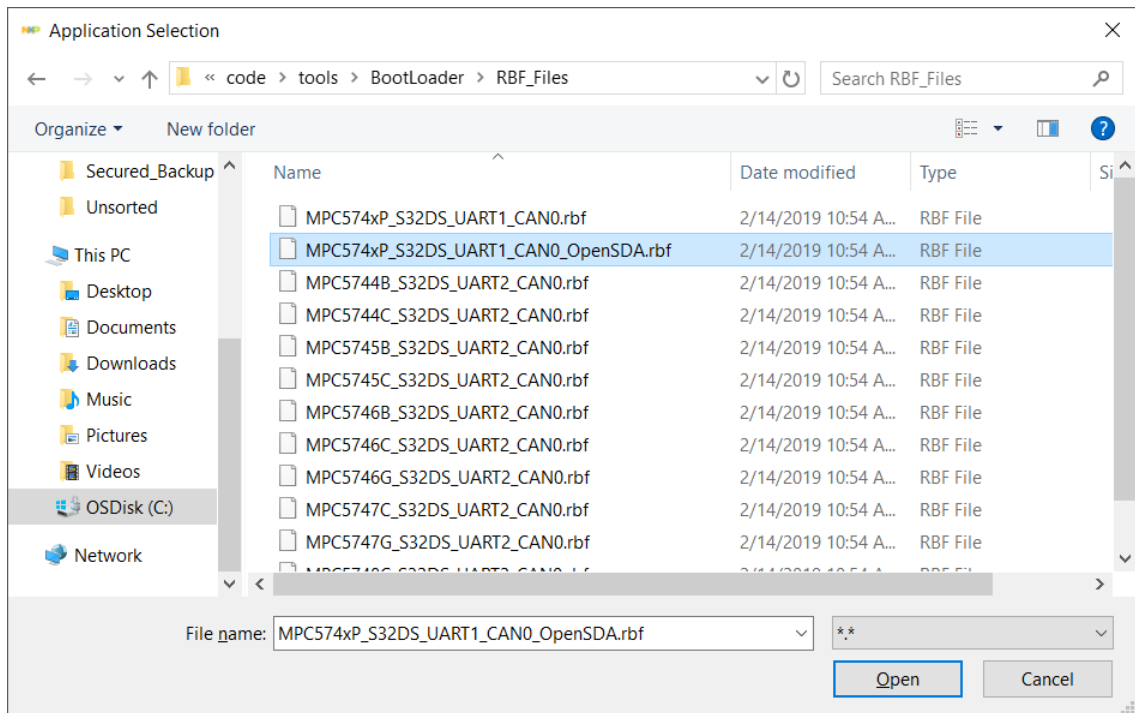


2. The Flash Wizard will appear on your screen. Add a **new** configuration and name it RBF Flash.

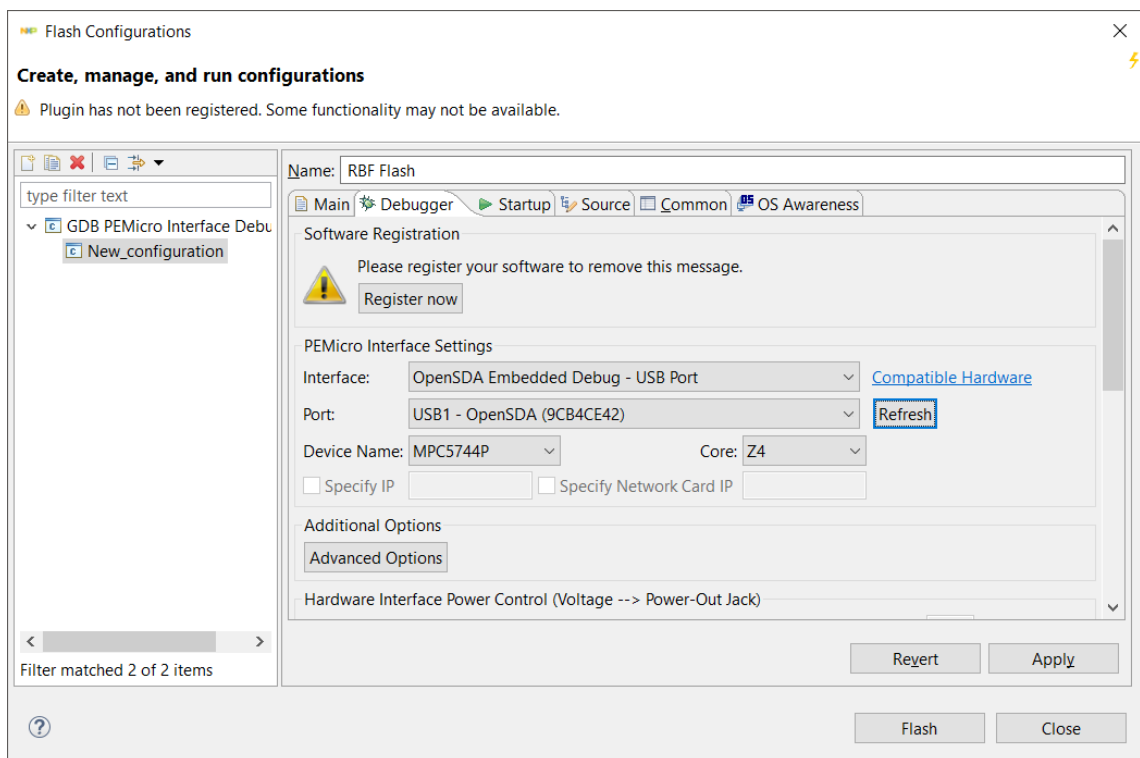


3. Click on the **C/C++ Application Browse** button to navigate to the {Model-Based Design Toolbox installation folder}\code\tools\BootLoader\RBF_Files and choose the appropriate RBF image to be flashed on your setup.

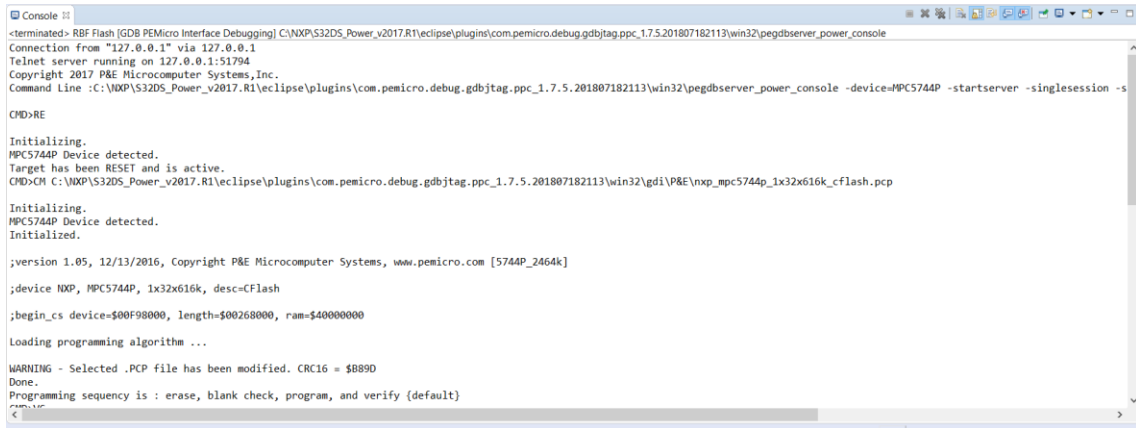
Note: each file contains in its name the interfaces allowed for Flash Tool download communication. Choose the one that is suitable for your processor and communication interface. In this case, we are going to use the UART1(openSDA) to download the code from host PC to DEVKIT-MPC5744P board.



4. Configure the **Debugger** interface to connect with the target.



5. Click on **Flash** button to start the flash programming sequence. The debugger will connect with the target selected and will start to flash the RBF image into the non-volatile memory of the target.



```
<terminated> RBF Flash [GDB PEMicro Interface Debugging] C:\NXP\S32DS_Power_v2017.R1\eclipse\plugins\com.pemicro.debug.gdbjtag.ppc_1.7.5.201807182113\win32\pegdbserver_power_console
Connection from "127.0.0.1" via 127.0.0.1
Telnet server running on 127.0.0.1:51794
Copyright 2017 P&E Microcomputer Systems, Inc.
Command Line :C:\NXP\S32DS_Power_v2017.R1\eclipse\plugins\com.pemicro.debug.gdbjtag.ppc_1.7.5.201807182113\win32\pegdbserver_power_console -device=MPC5744P -startserver -singleession -s
CMD>RE

Initializing.
MPC5744P Device detected.
Target has been RESET and is active.
CMD>CM C:\NXP\S32DS_Power_v2017.R1\eclipse\plugins\com.pemicro.debug.gdbjtag.ppc_1.7.5.201807182113\win32\gdi\NXP\mpc5744p_1x32x616k_cflash.pcp

Initializing.
MPC5744P Device detected.
Initialized.

;version 1.05, 12/13/2016, Copyright P&E Microcomputer Systems, www.pemicro.com [5744P_2464k]
;device NXP, MPC5744P, 1x32x616k, desc=CFlash
;begin_cs device=$00F98000, length=$00268000, ram=$40000000

Loading programming algorithm ...

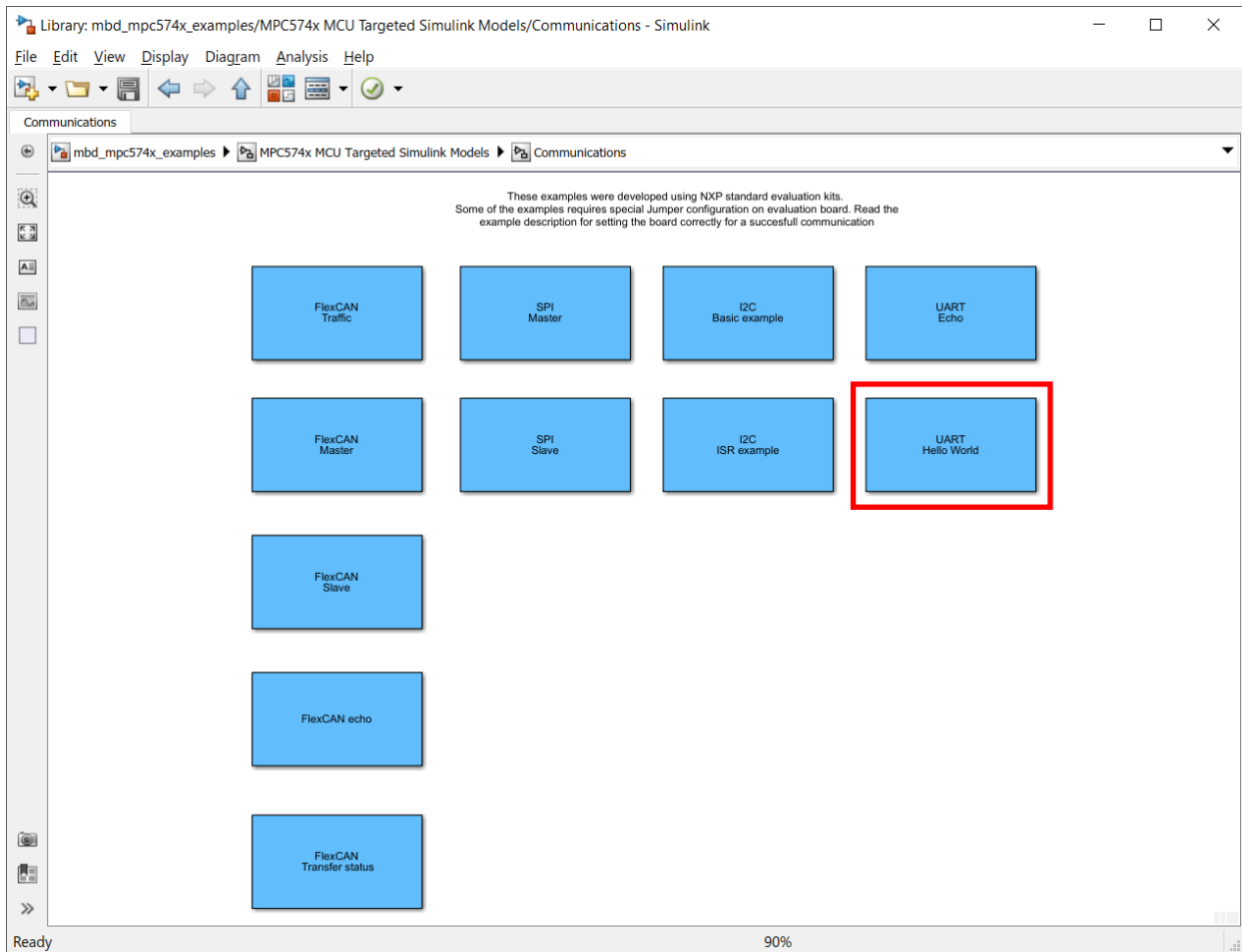
WARNING - Selected .PCP file has been modified. CRC16 = $B89D
Done.
Programming sequency is : erase, blank check, program, and verify {default}
```

At this point the target should be ready for receiving applications from MATLAB/Simulink environment over the UART interface connected via OpenSDA.

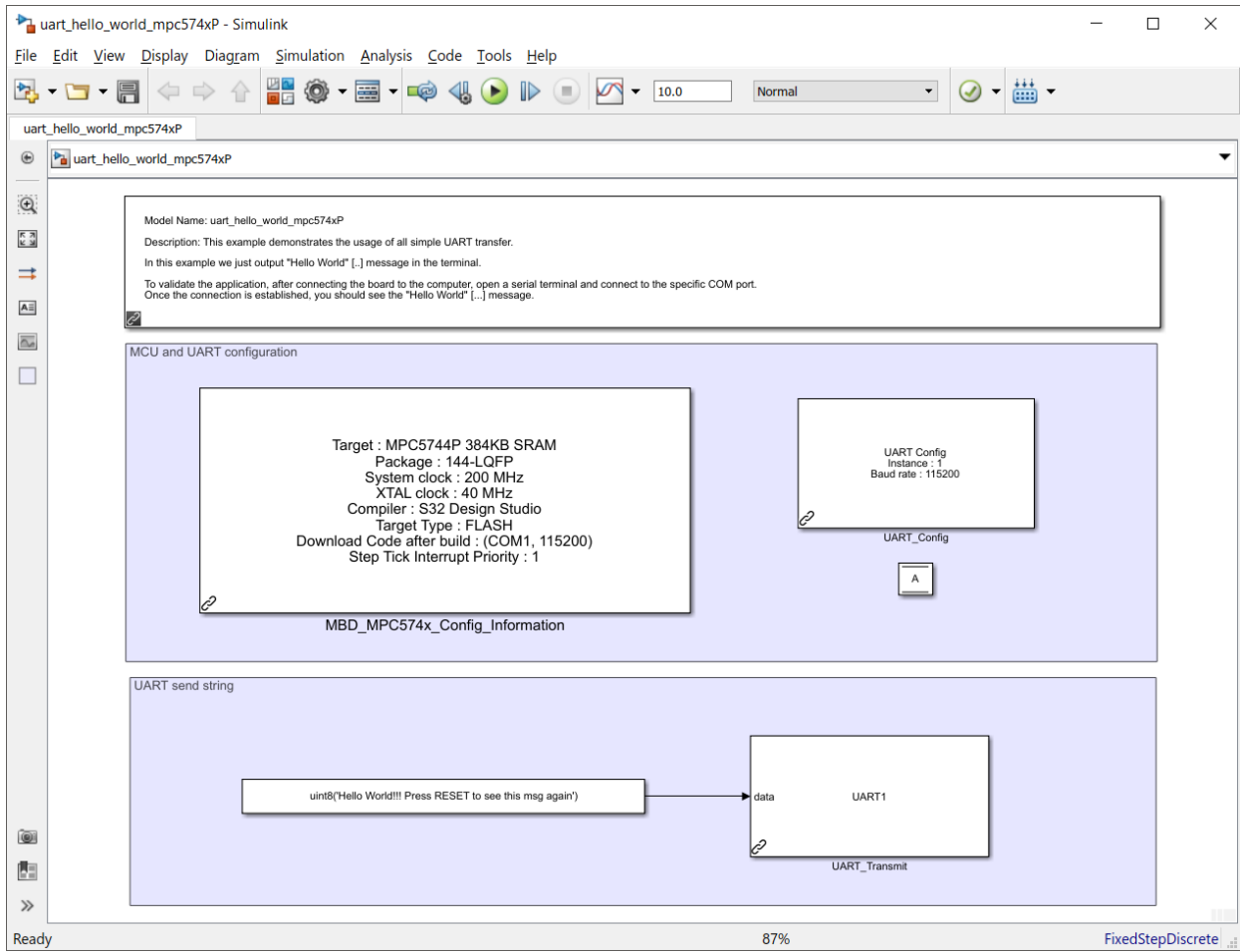
2.4 Hello World Example

If the hardware and BootLoader setup is completed successful and a virtual COM port is created and visible in Control Panel -> Device Manager -> Port (COM & LPT), then all requirements for running a Model-Based Design Toolbox for MPC57xx Examples are fulfilled.

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

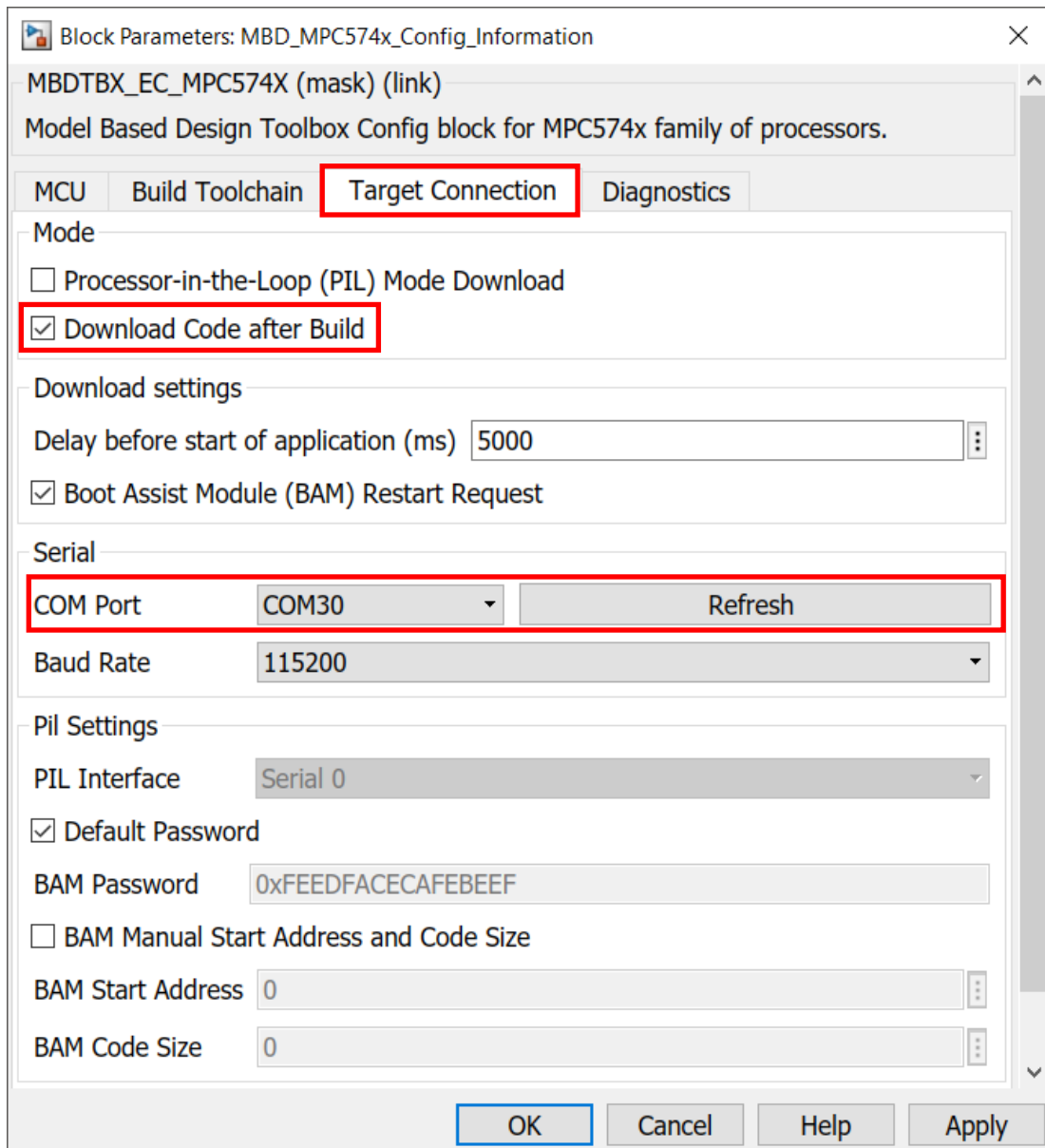


The Simulink `uart_hello_world_mpc574xP.mdl` model will open. This model programs the MPC5744P MCU to sent a message over the UART.



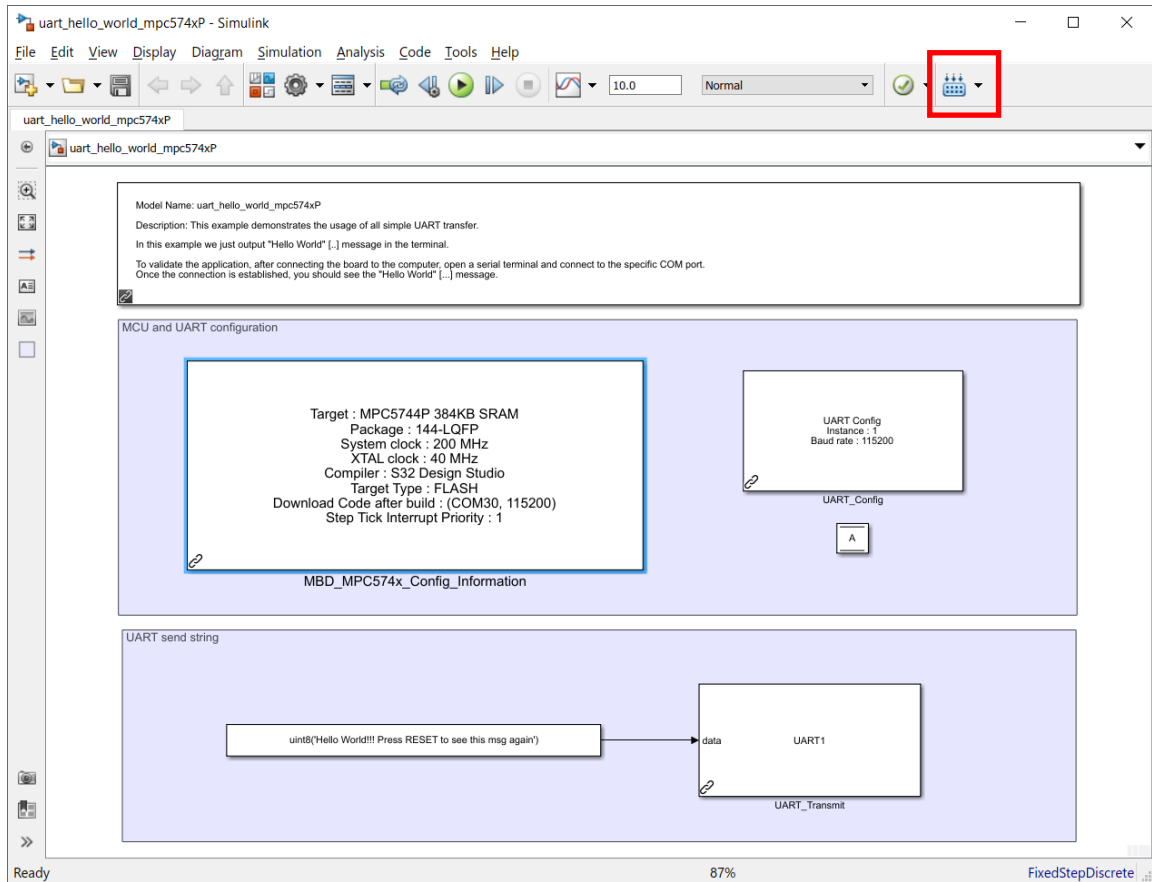
Follow the next steps to run the example:

1. Double click on MBD_MPC574x_Config_Information block and setup the Download Config parameters according with your PC and HW setup

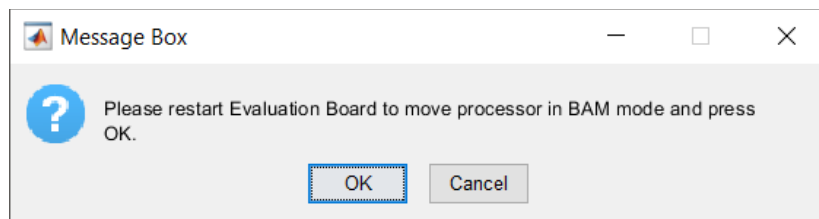


2. Apply and close this window.

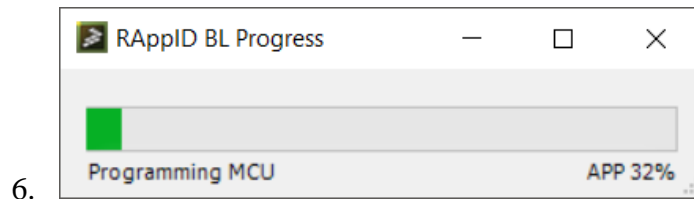
- Press **Build Model** button and wait until the code is generated, compiled and downloaded to the evaluation board.



- Wait until the application is built and the following message is displayed on the screen. Press the board reset button SW3 and then (within 5 seconds) press the OK button to start the application programming to the target.

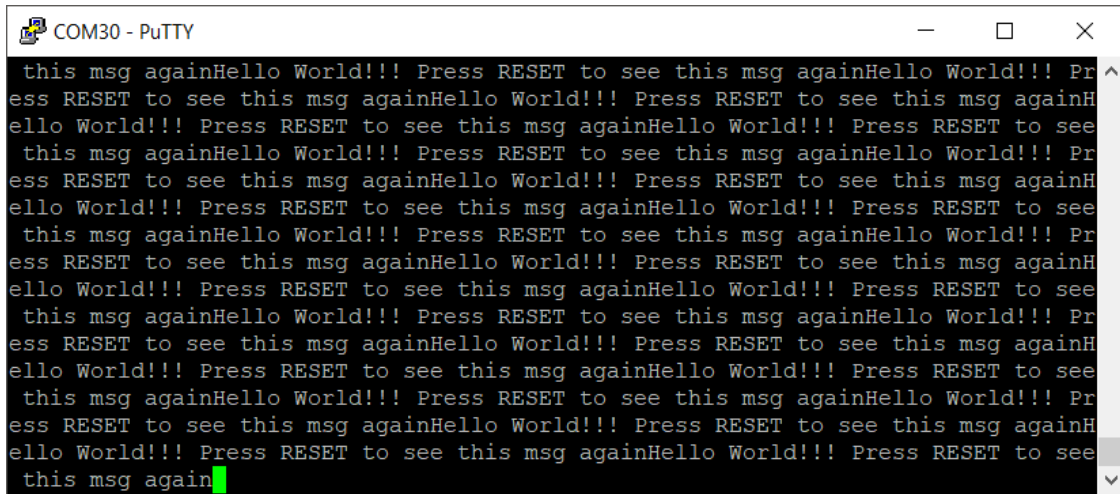


- Wait until the BootLoader finishes the flashing of the application to the target processor.



- Open any UART terminal for the virtual COM port assigned and set up the baud rate at 115200, data bits 8 and parity none.

8. Press the reset button on the evaluation board.
9. The MPC5744P MCU sends “Hello World!!! Press RESET to see this msg again” message over the UART and the UART terminal should display it.



The image shows a screenshot of a PuTTY terminal window titled "COM30 - PuTTY". The terminal displays a repeating message: "this msg againHello World!!! Press RESET to see this msg again". The message is repeated multiple times, with a green cursor at the end of the last line. The terminal window has standard window controls (minimize, maximize, close) in the top right corner.

Congradulations! You succeeded with running your first example created with Model-Based Design Toolbox for MPC57xx

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, FlexIm, 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.

