Model-Based Design Toolbox For DSC MC56F8x Series

Quick Start Guide

Automatic Code Generation for the DSC MC56F8x MCUs Version 1.0.0

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 MBDT official download 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 by pressing on the **`Download`** button.
2. Setup the MATLAB path for Model-Based Design Toolbox and generate the appropriate toolchain setting for the user MATLAB environment.

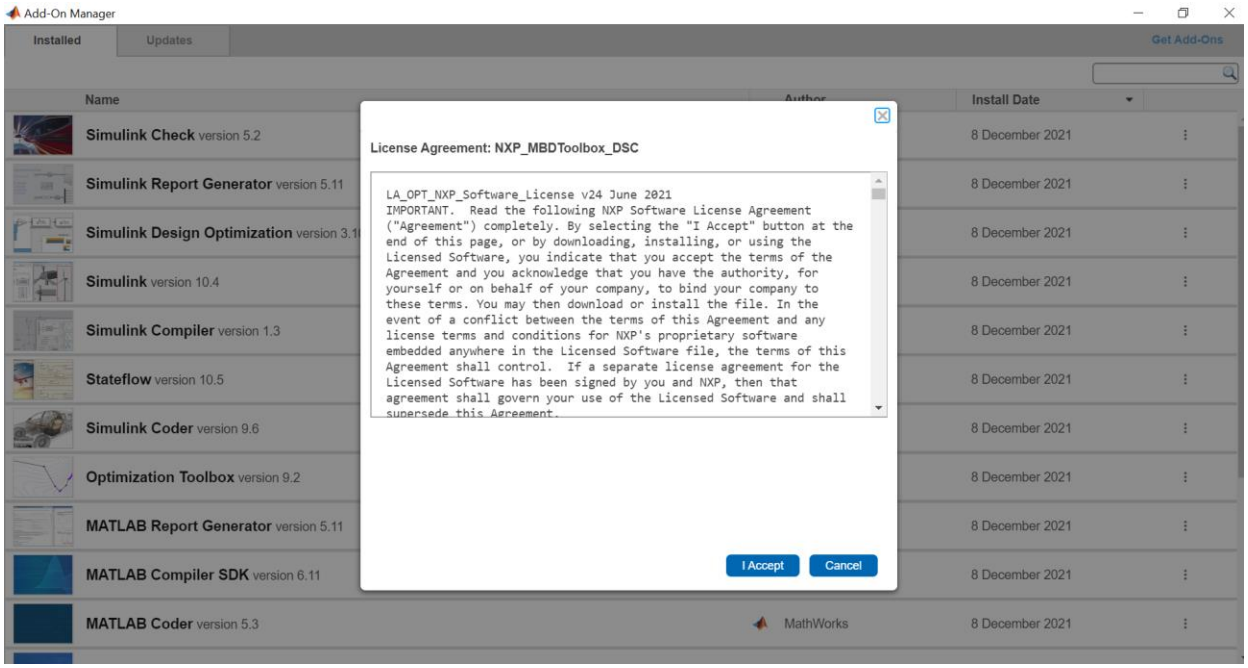
Each of these actions is explained in the following sub-chapters.

1.2.1 **Run Add-on installer**

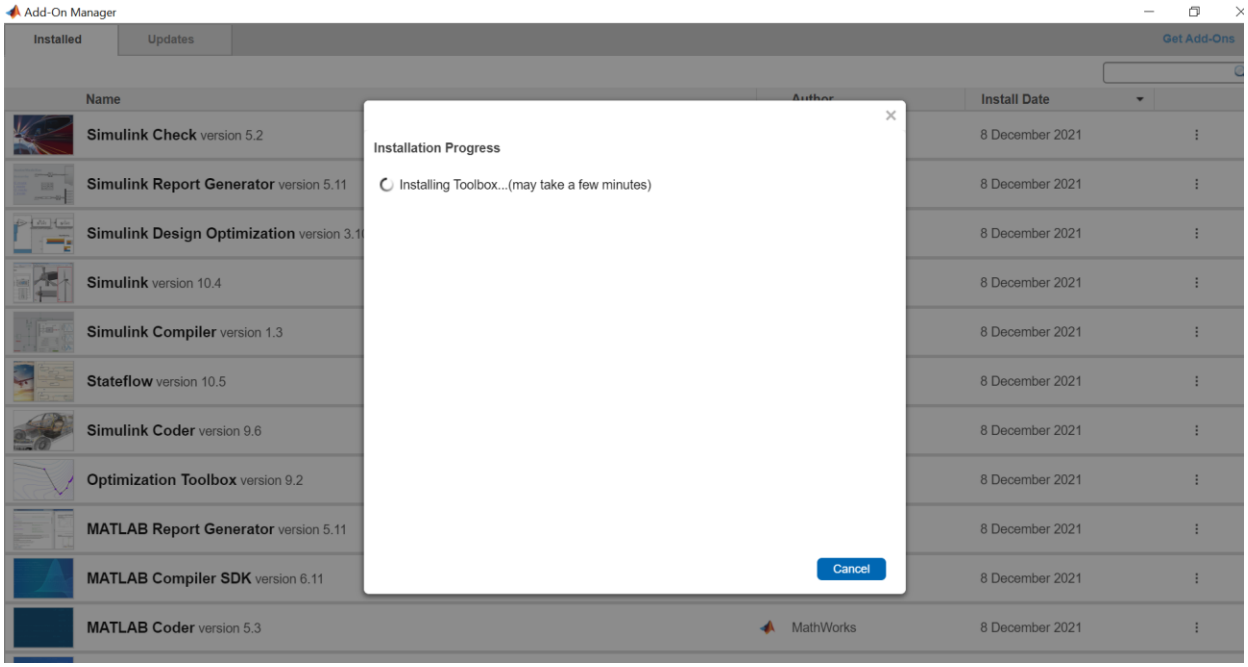
Install the NXP's Model-Based Design Toolbox by double-clicking the `*.mltbx` file. This will activate the MATLAB Add-ons installer that will automatically start the installation process.

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

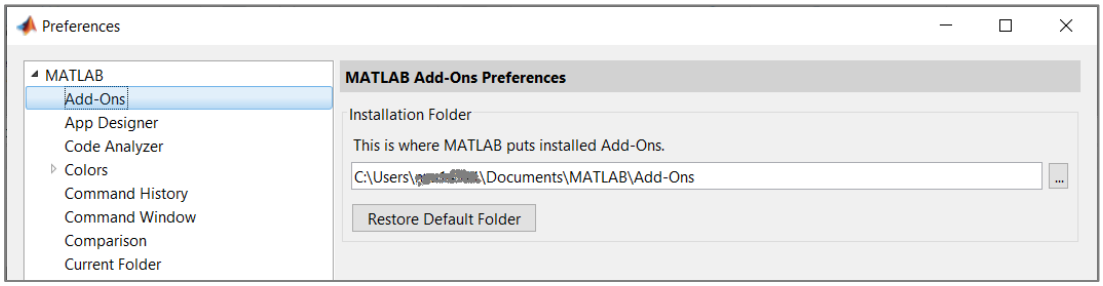
1. Indicate acceptance of the NXP Software License Agreement by selecting "`I Accept`" to proceed.



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

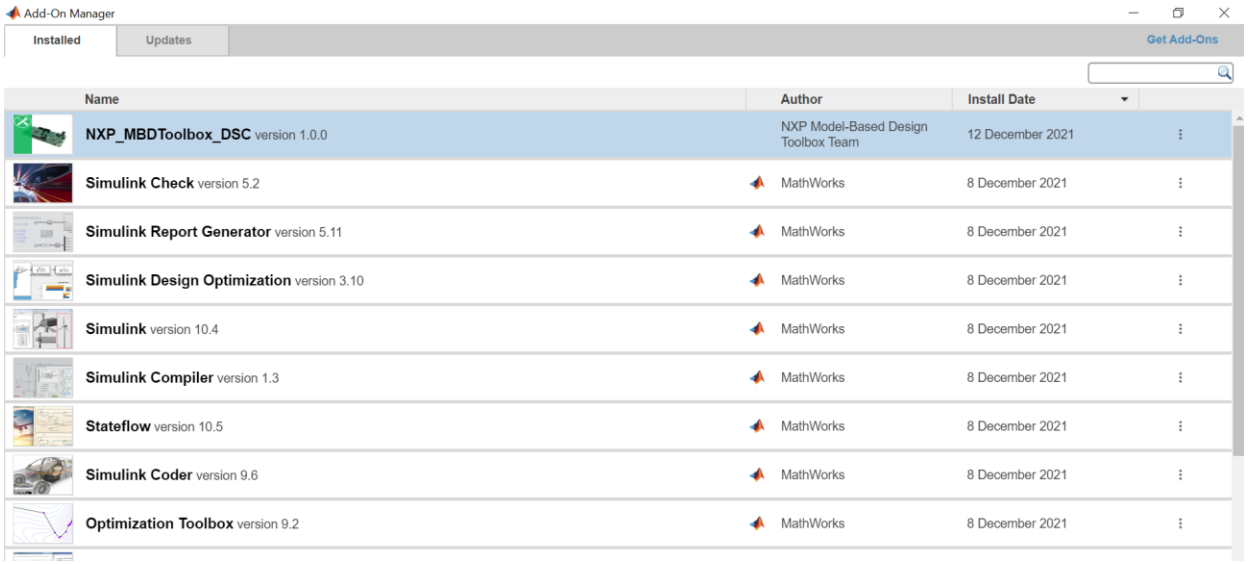


The default location can be changed before installation by changing the Add-Ons path from MATLAB Preferences

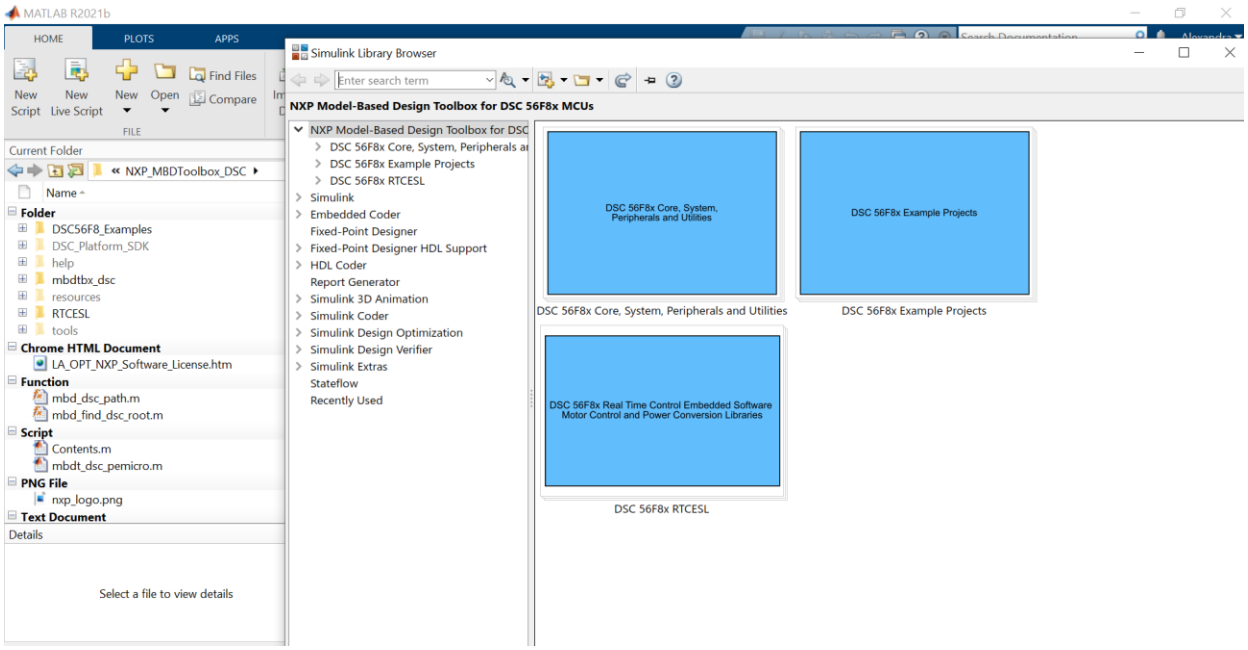


Note: *It is recommended to install the MATLAB and NXP Toolbox into a location that does not contains special characters, empty spaces, or mapped drives.*

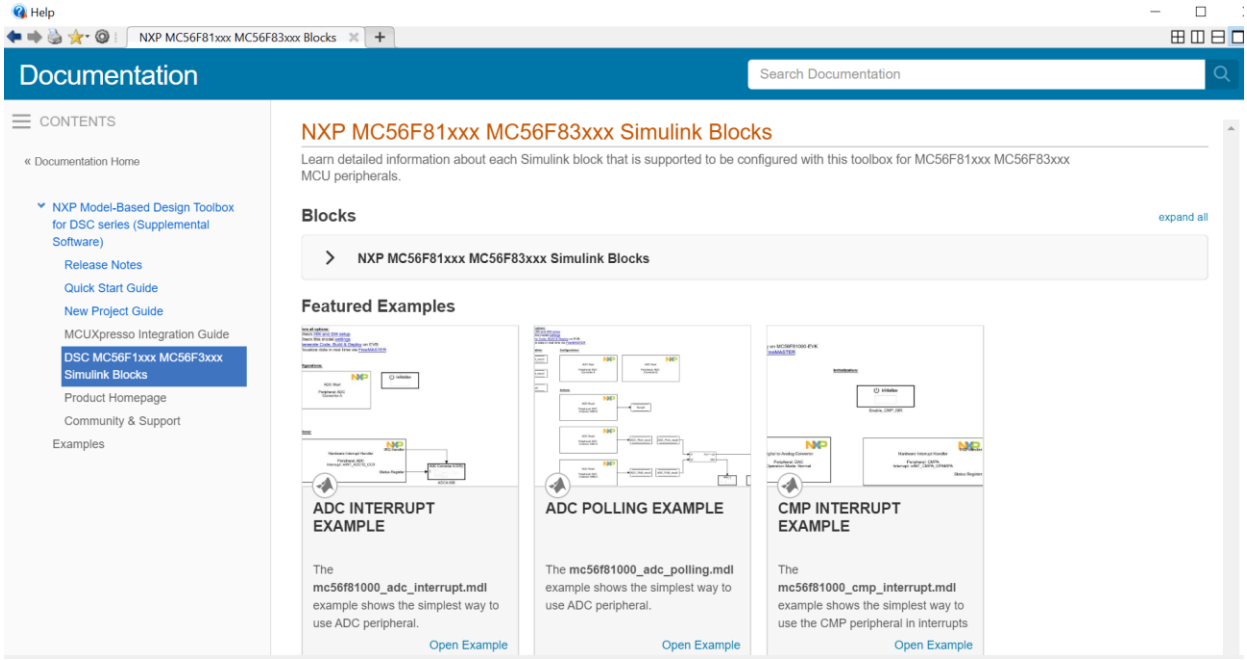
3. After a couple of minutes (4-5min), the NXP's Model-Based Design Toolbox should be visible as a new Add-ons.



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



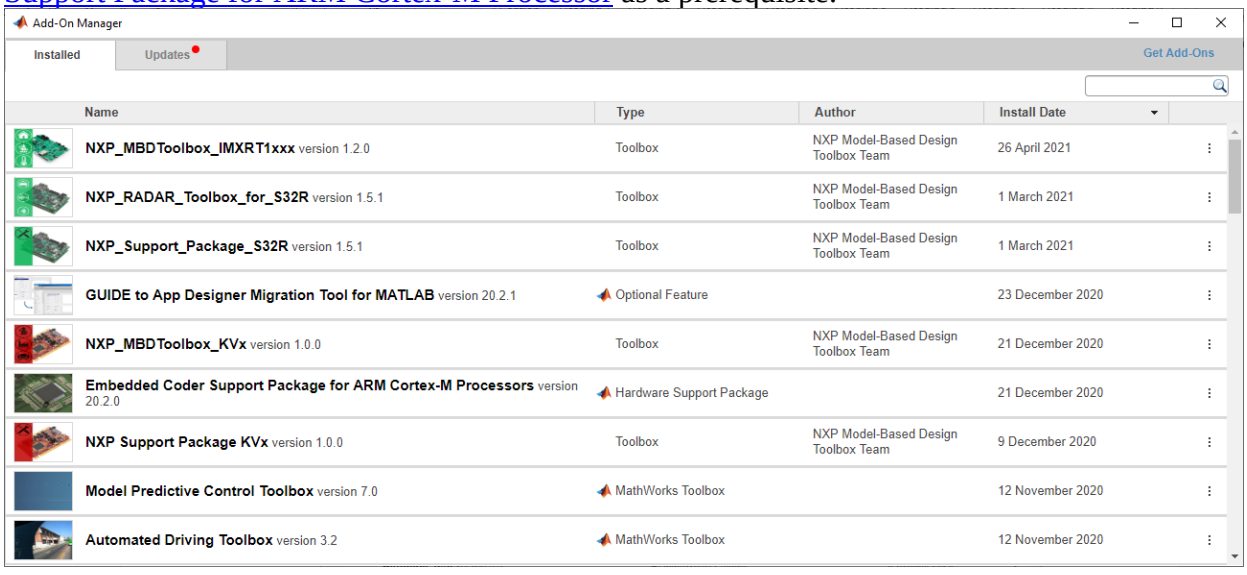
5. NXP's Model-Based Design Toolbox documentation, help, and examples are fully integrated with the MATLAB development environment. Get more details by accessing the standard Help and **Supplemental Software** section



1.2.2 Configure the toolbox

The Model-Based Design Toolbox uses the Toolchain mechanism exposed by the Simulink to enable automatic code generation with the Embedded Coder toolbox.

A custom target was added for the DSP56800 core. Although the DSC target is not ARM based, the custom device registration is using MATLAB API that comes with the Embedded Coder for Cortex M add-on. As a consequence the users are required to install the Embedded Coder Support Package for ARM Cortex-M Processor as a prerequisite.



The toolchain and custom device are configured when `sl_refresh_customizations` is ran from Matlab. This command is run automatically by Simulink when opening for the first time a **Simulink** model in the current Matlab session. Otherwise, if the toolbox is installed on a Matlab session that already was used for development of Simulink models, then the user must run the next command from command window:

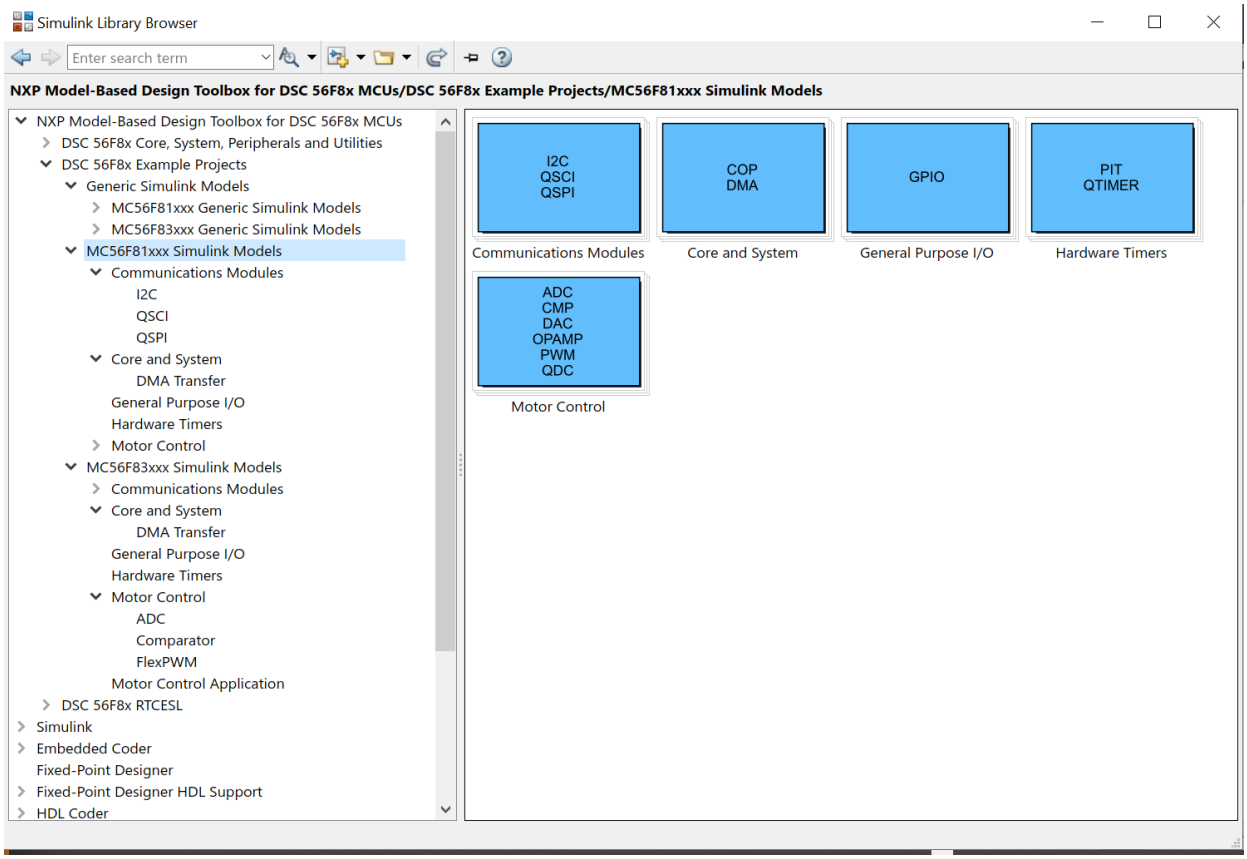
```
>> sl_refresh_customizations
```

2 Run Models

2.1 Examples Library & Help

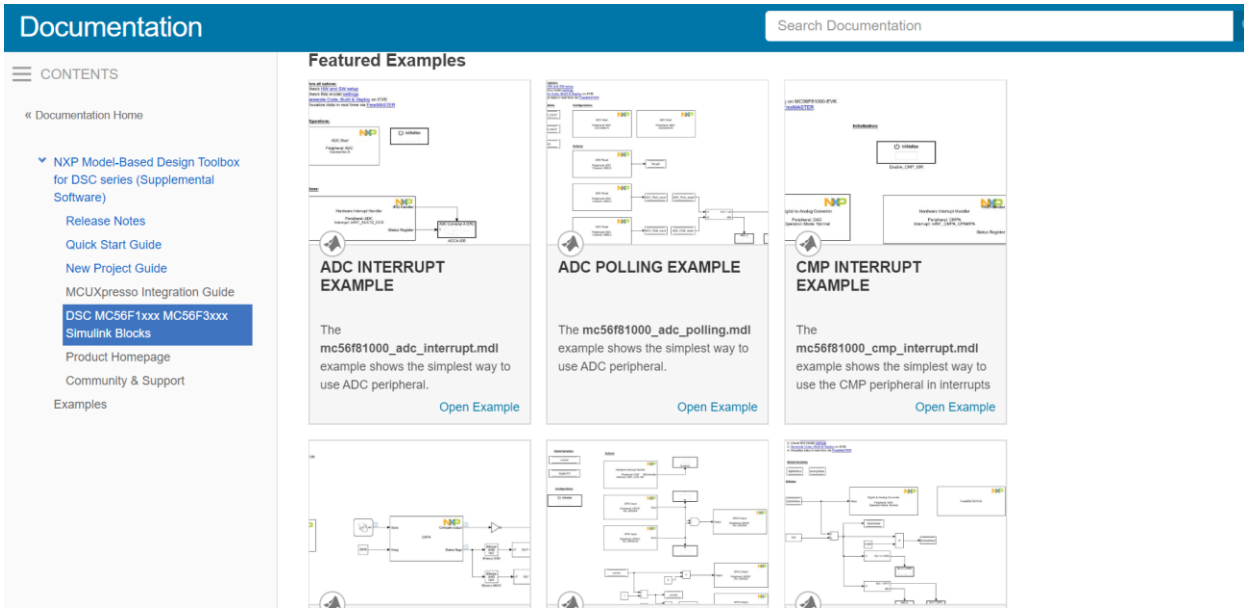
NXP's Model-Based Design Toolbox comes with an Examples Library collection that lets you test different MCU on-chip modules and run complex applications.

The Examples Library `mbd_dsc_examples.slx` can be opened from "`{Model Based Design Install Directory}\DSC56F8_Examples\`" folder or directly from the `Simulink Library Browser` main window



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


Documentation
Search Documentation
CONTENTS
« Documentation Home
NXP Model-Based Design Toolbox for DSC series (Supplemental Software)
Release Notes
Quick Start Guide
New Project Guide
MCUXpresso Integration Guide
DSC MC56F1xxx MC56F3xxx Simulink Blocks
Product Homepage
Community & Support
Examples
Featured Examples
ADC INTERRUPT EXAMPLE
The mc56f81000_adc_interrupt.mdl example shows the simplest way to use ADC peripheral.
Open Example
ADC POLLING EXAMPLE
The mc56f81000_adc_polling.mdl example shows the simplest way to use ADC peripheral.
Open Example
CMP INTERRUPT EXAMPLE
The mc56f81000_cmp_interrupt.mdl example shows the simplest way to use the CMP peripheral in interrupts
Open Example

2.2 Hardware Setup

All examples provided with the Model-Based Design Toolbox were developed on MC56F81000-EVK and MC56F83000-EVK as the primary hardware targets.
The toolbox offer two options to download the application on the target:

- Using the integrated CW Flash Programmer
- Using the PROGDSC flash programmer provided by PE Micro

Before running any example on the either of the boards, the user must select the proper download method. Default the models have set the PROGDSC Flash programmer as download method. For this the user must install on his/her machine the program. This is free of charge and can be downloaded from the PE MICRO site. After installing the program, the user must specify in the model the installation path.

2.3 A “Hello World” Example

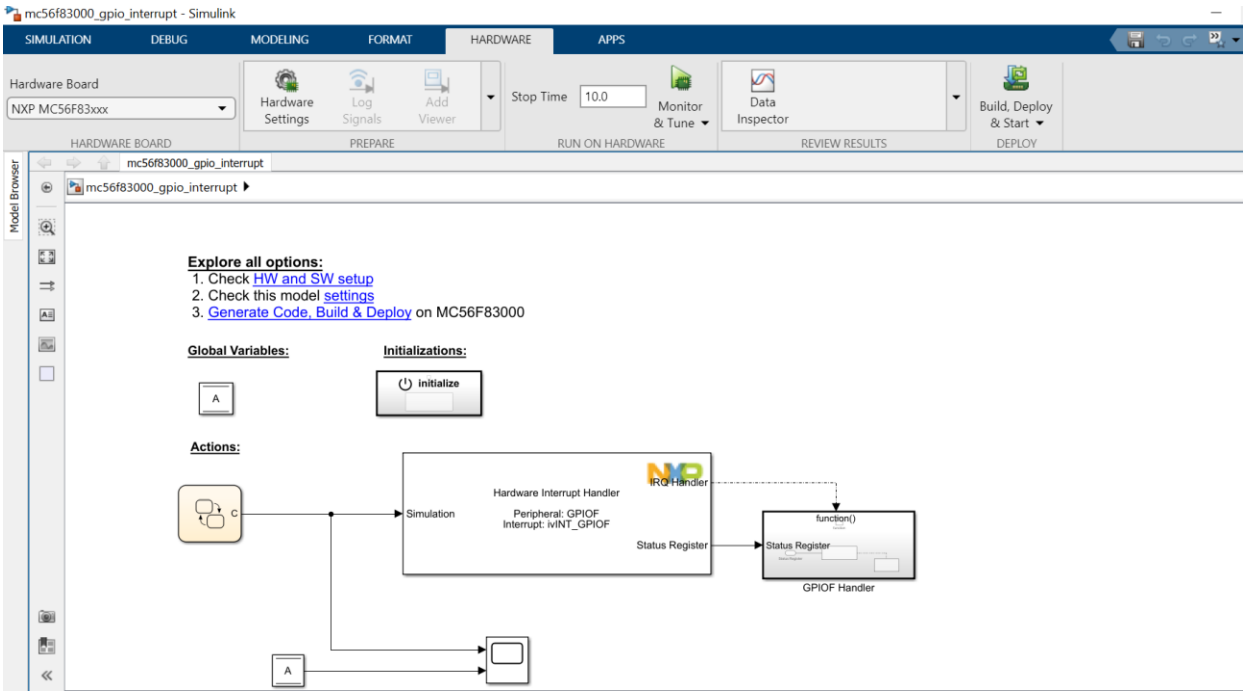
If the hardware setup is completed successfully:

- a virtual COM port is created and visible in `Control Panel -> Device Manager -> Port (COM & LPT)`
- a virtual mass storage device is present

then all ingredients are present for running successfully the Model-Based Design Toolbox for DSC MC56F8x specific examples.

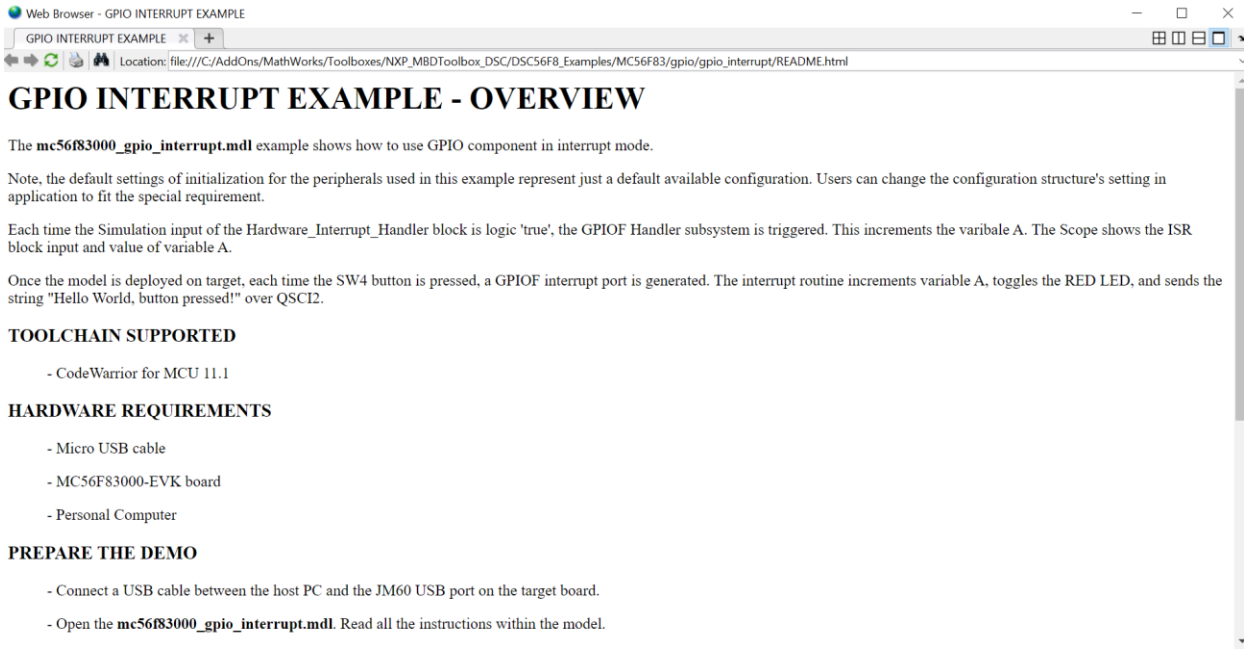
Navigate to “`\DSC56F8_Examples\MC56F83\gpio\gpio_interrupt\`” folder and open the `mc56f83000_gpio_interrupt.mdl` Simulink model.

This model programs the MC56F83000-EVK to sent a “Hello World” type of message over the UART each time the USER BTN on the board (SW4) is pressed. The BLUE LED should toggle and the message should appear inside the UART terminal.

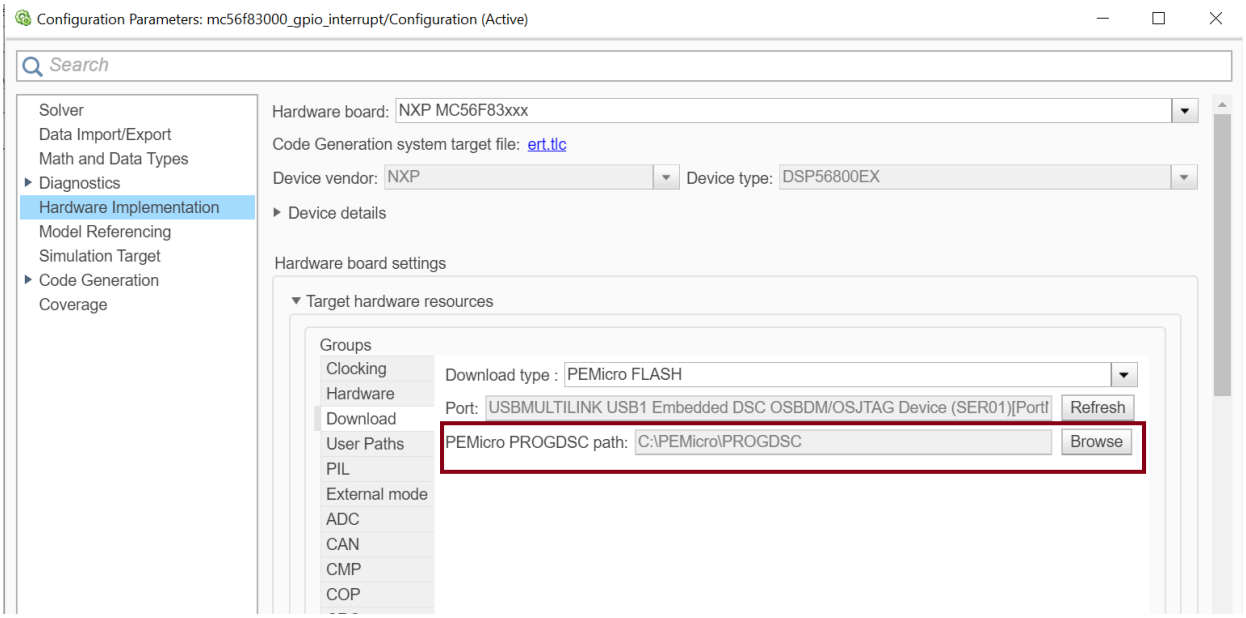


Follow the next steps to run the example:

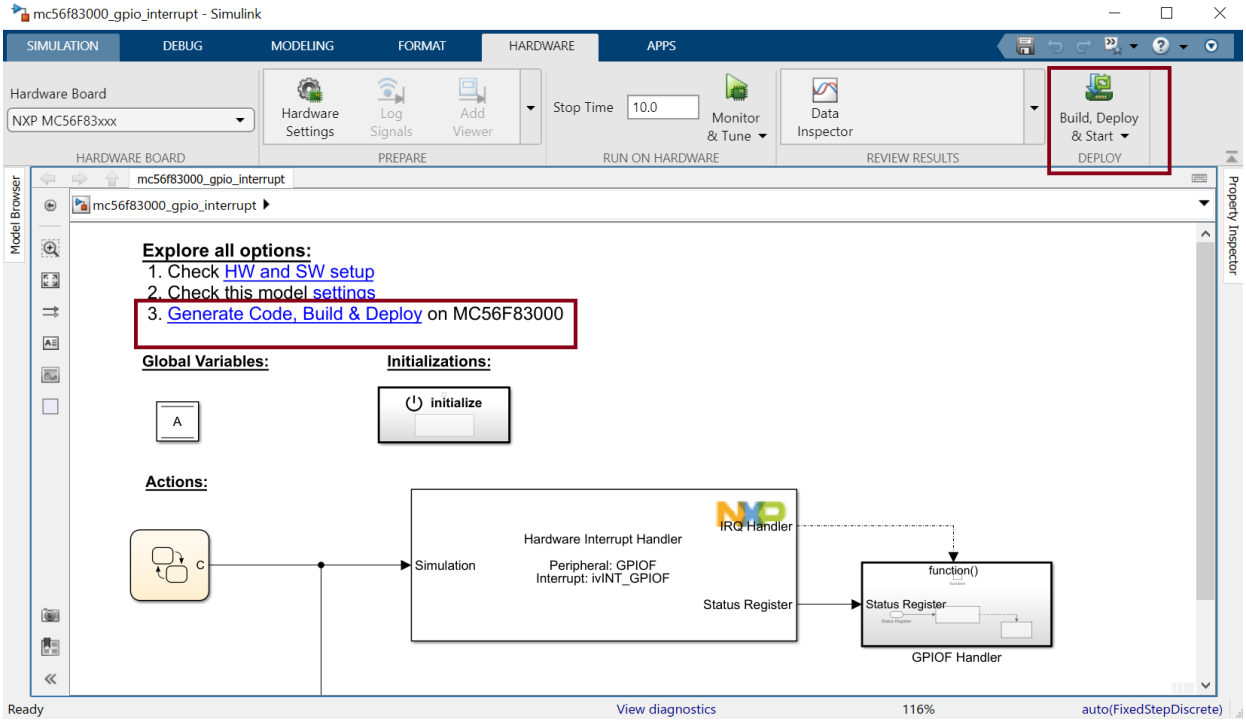
1. Open and README.html file to understand the hardware and software requirements for running the application



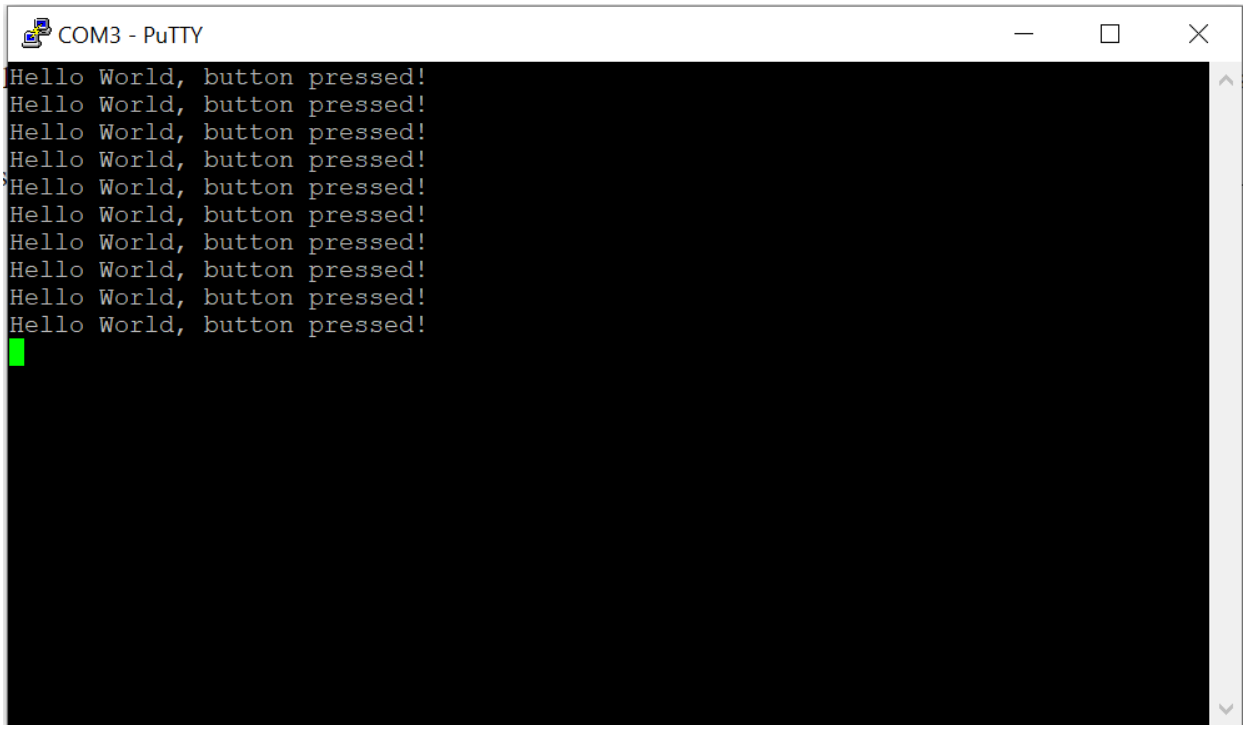
2. Open the Simulink `Model Configuration Parameters` and select the download method and, if needed, the path to the PROGDSC program.



3. 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.



4. Open any UART terminal (e.g.: puTTY.exe) 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. Now, press the USER BTN on the board (SW4). The board should send "`Hello World, button pressed!`" message over the UART and the UART terminal should display it.



Congratulations! You succeeded with running your first example created with Model-Based Design Toolbox for DSC MC56F8x MCUs

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