Model-Based Design Toolbox LPC553x Series

Quick Start Guide

Automatic Code Generation for the LPC553x Family of Processors
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
Windows 11		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 install and properly configure the Model-Based Design Toolbox the next steps shall be followed :

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 the Model-Based Design Toolbox and generate the appropriate toolchain setting for the user MATLAB environment.

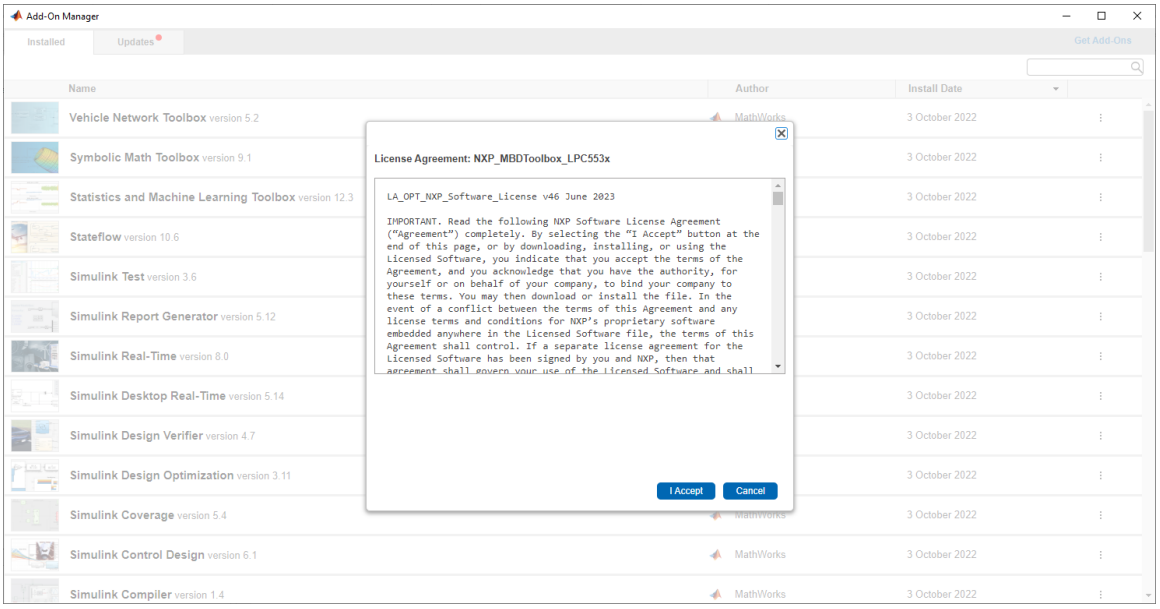
Each of these actions is explained in the following sub-chapters.

1.2.1 **Run Add-on installer**

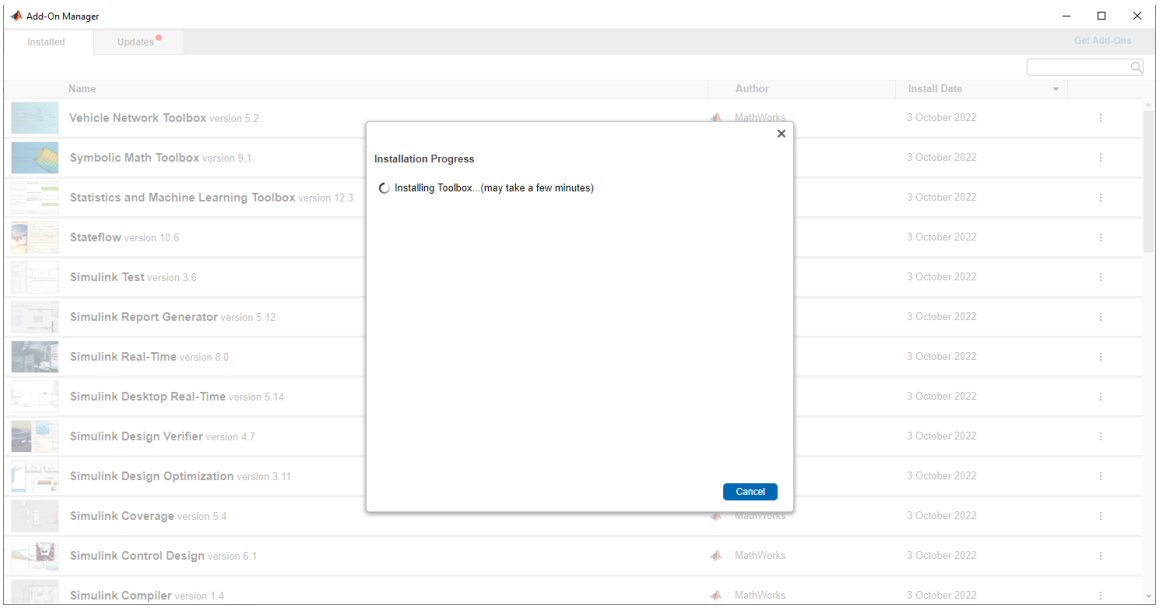
Install the NXP's Model-Based Design Toolbox by double-clicking the `*.mltbx` file. This will activate the MATLAB Add-ons installer that will automatically start the installation process.

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

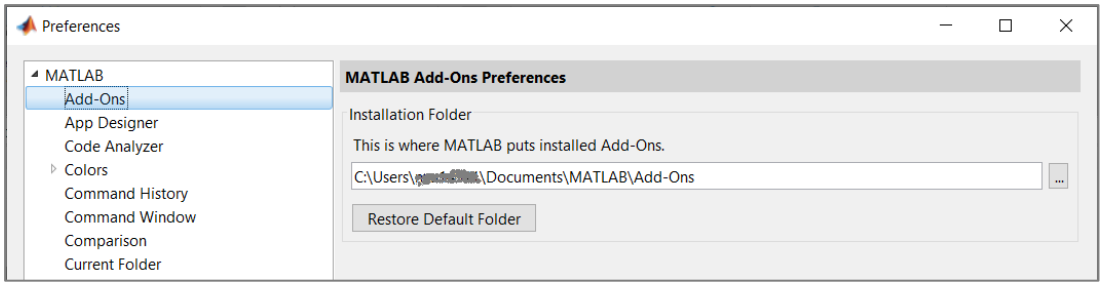
1. Indicate acceptance of the NXP Software License Agreement by selecting "`I Accept`" to proceed.



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

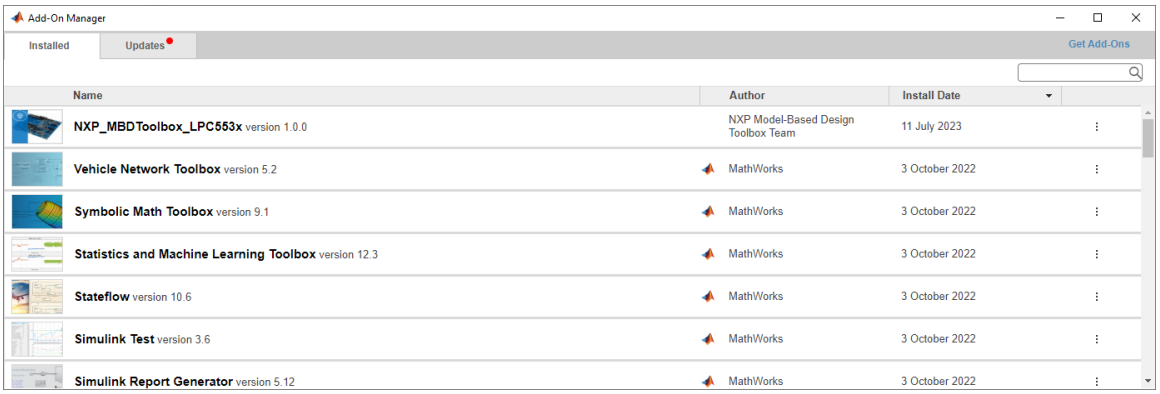


The default location can be changed before installation by changing the Add-Ons path from MATLAB Preferences

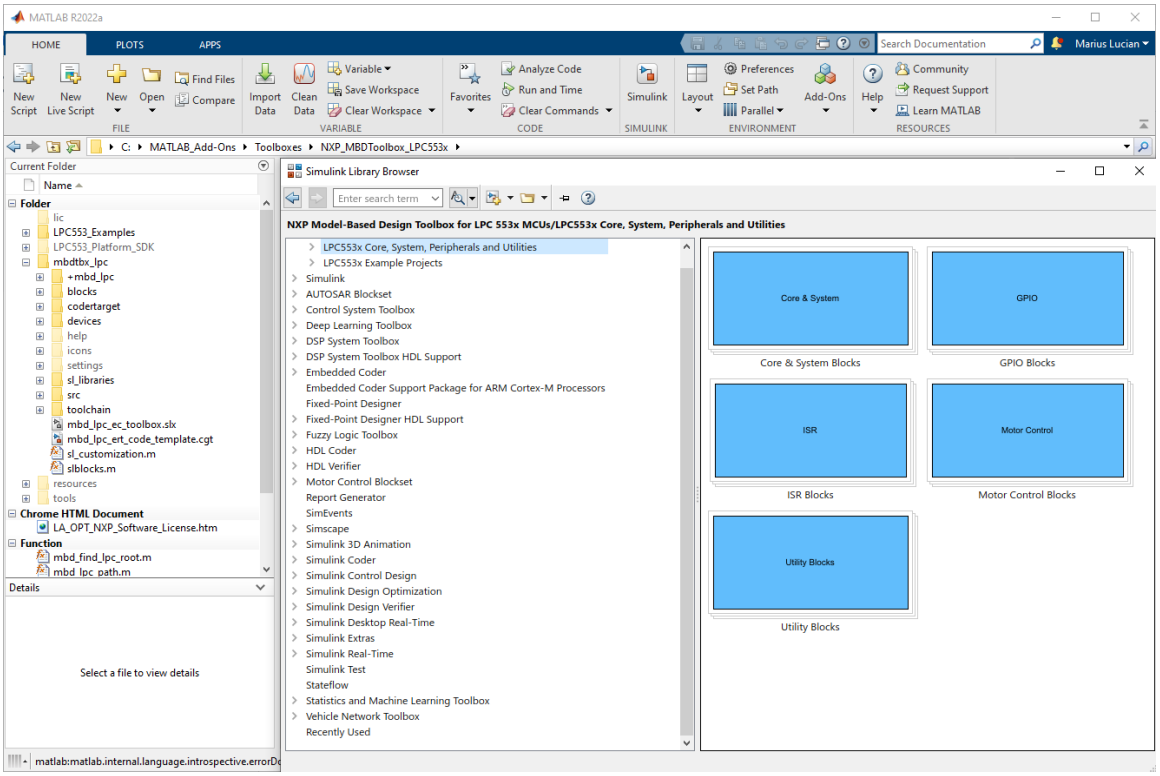


Note: *It is recommended to install the MATLAB and NXP Toolbox into a location that does not contain special characters, empty spaces, or mapped drives.*

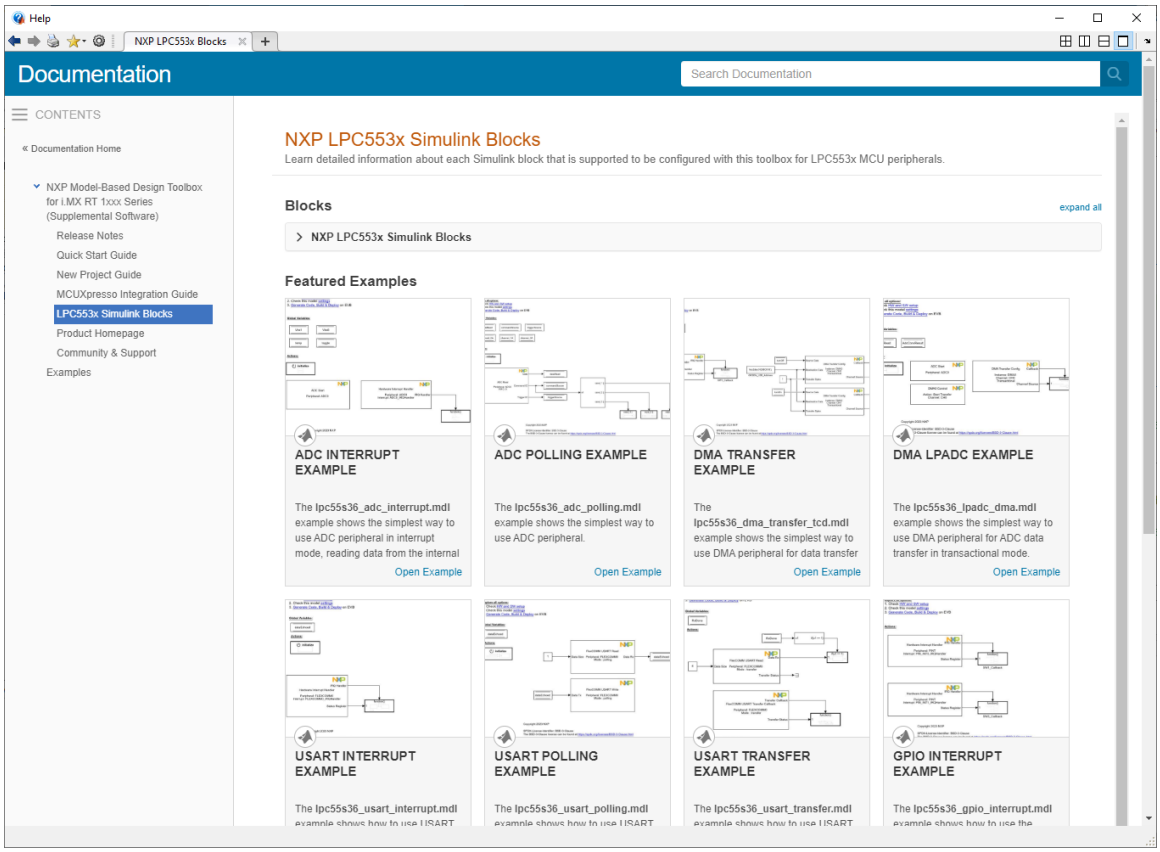
3. After a couple of minutes (~5min), the NXP's Model-Based Design Toolbox should be visible as a new Add-ons.



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.

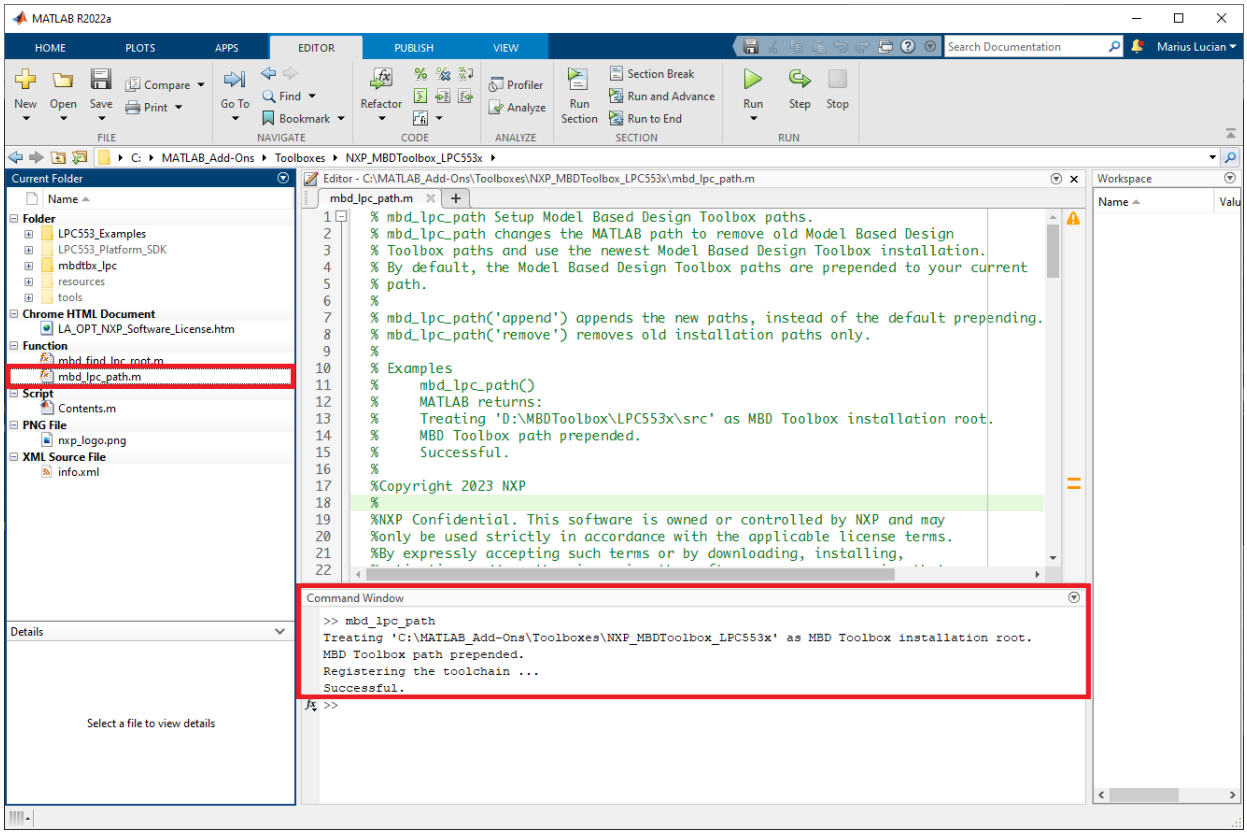


1.2.2 Setting the Path for Model-Based Design Toolbox and Toolchain Generation

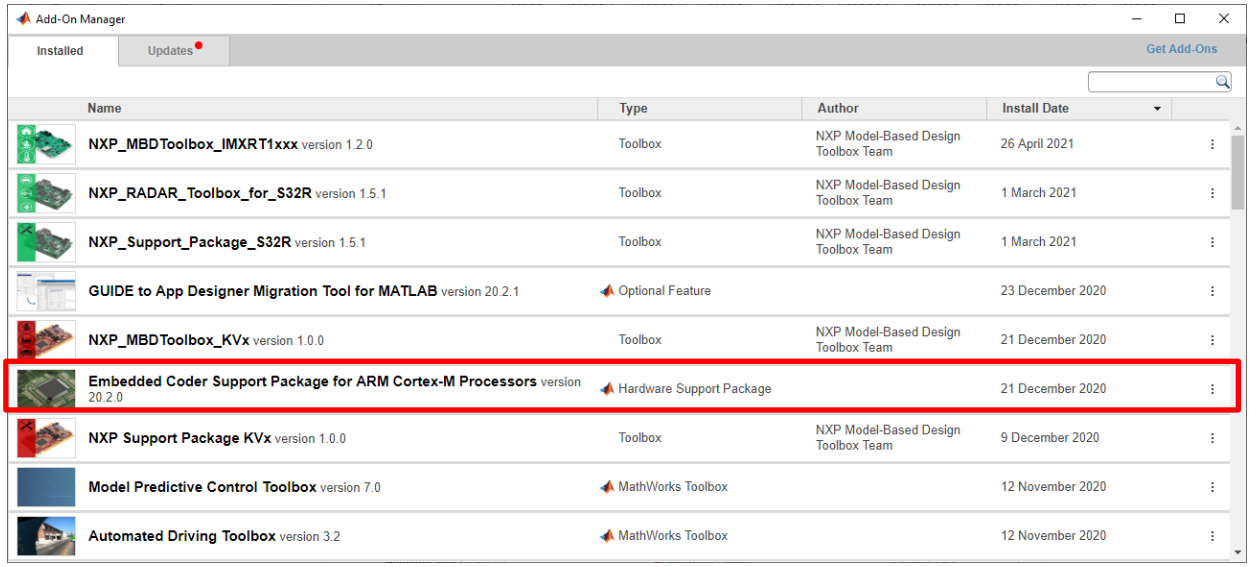
The Model-Based Design Toolbox uses the Toolchain mechanism exposed by the Simulink to enable automatic code generation with the Embedded Coder toolbox. By default, the toolchain is configured for the MATLAB R2022a, R2022b, R2023a releases. For newer MATLAB releases, the user needs to execute a toolbox 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_LPC553x\`) and running the "`mbd_lpc_path.m`" script.

```
>> cd (mbd_mbd_find_lpc_root)
>> mbd_lpc_path
Treating …\MATLAB\Add-Ons\Toolboxes\NXP_MBDToolbox_LPC553x' as MBD
Toolbox installation root.
MBD Toolbox path prepended.
Registering the toolchain ...
Successful.
>>
```



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



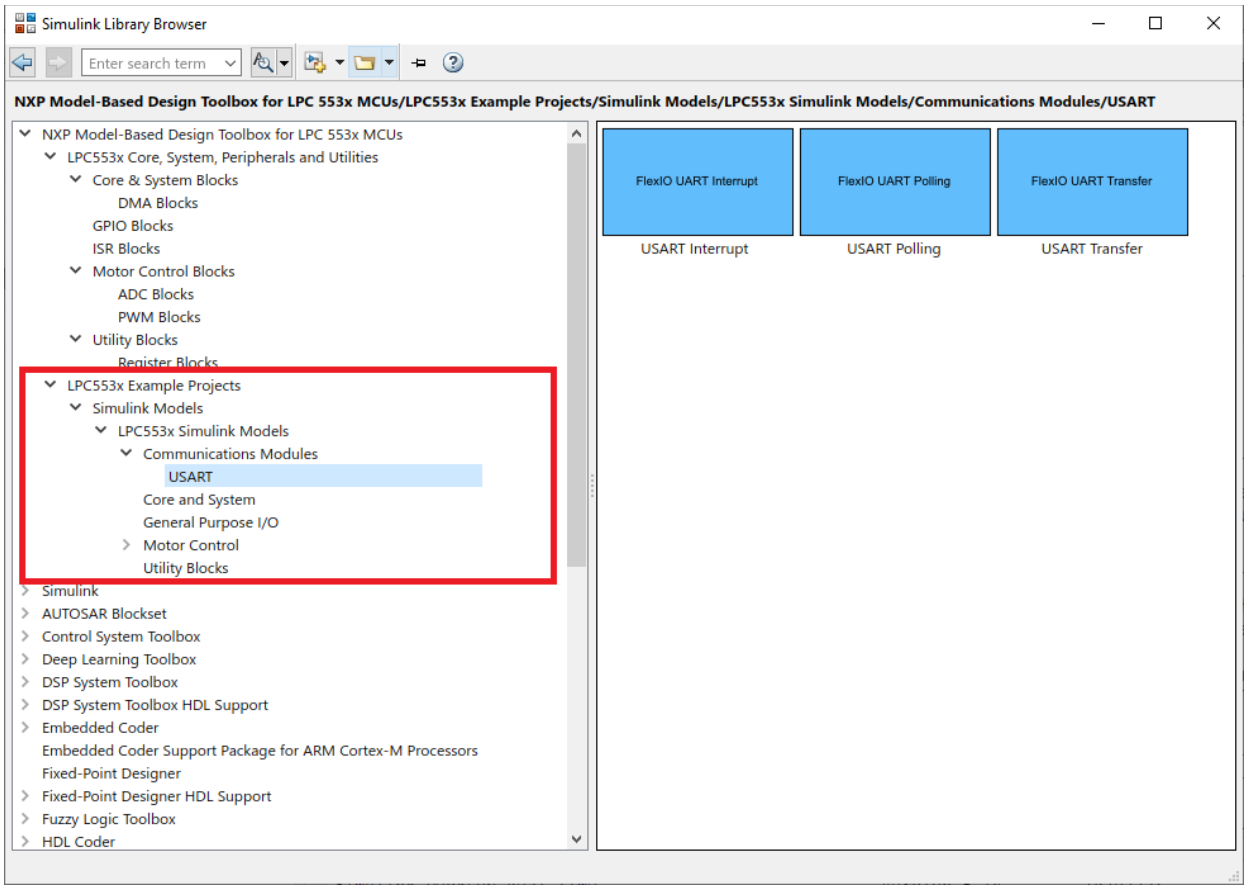
The "`mbd_lpc_path.m`" script verifies the user setup dependencies and will issue instructions for a successful installation and configuration of the toolbox.

2 Run Models

2.1 Examples Library & Help

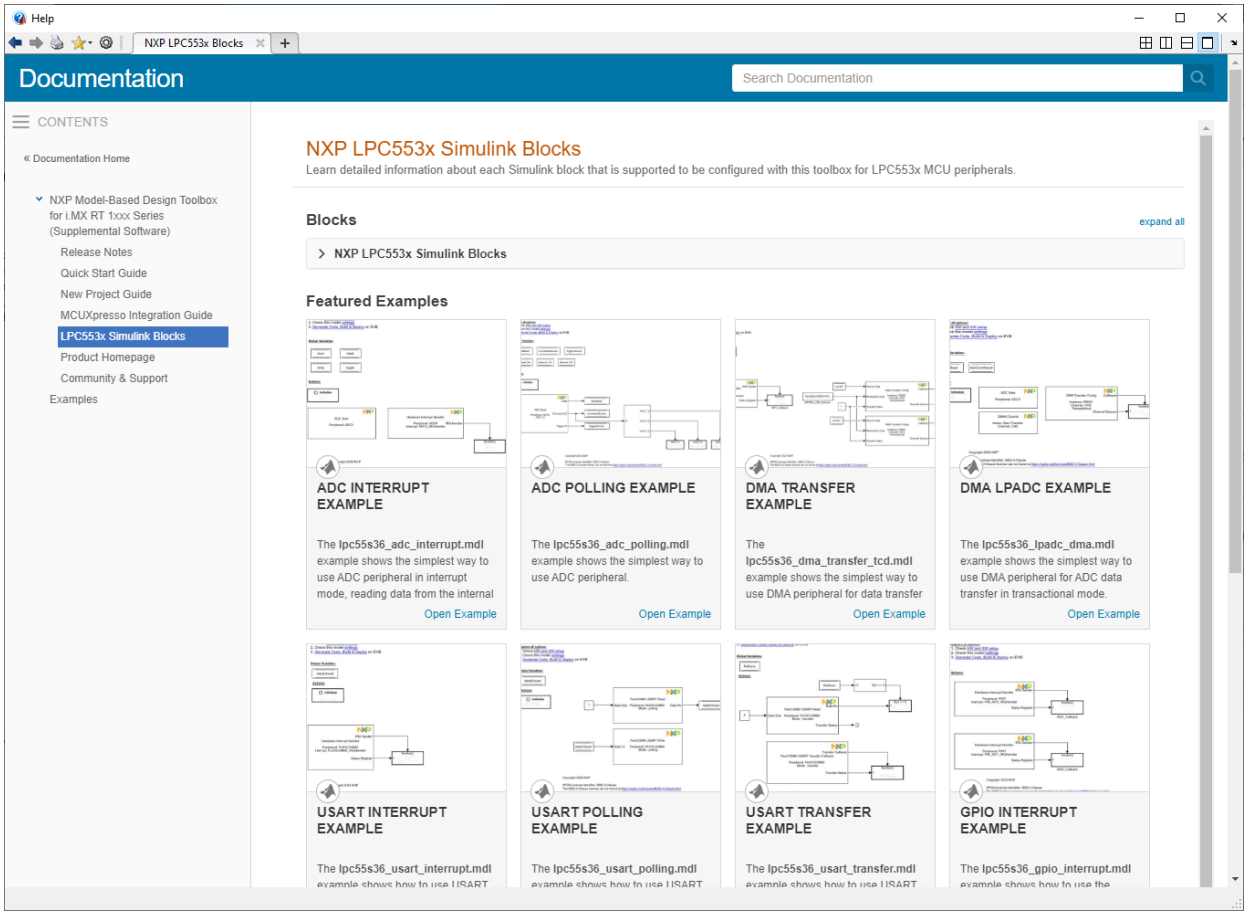
NXP's Model-Based Design Toolbox comes with an Examples Library collection that lets you test different MCU on-chip modules and run complex applications.

The Examples Library `mbd_lpc553_examples.slx` can be opened from "`{Model Based Design Install Directory}\LPC553_Examples\`" folder or directly from the `Simulink Library Browser` main window



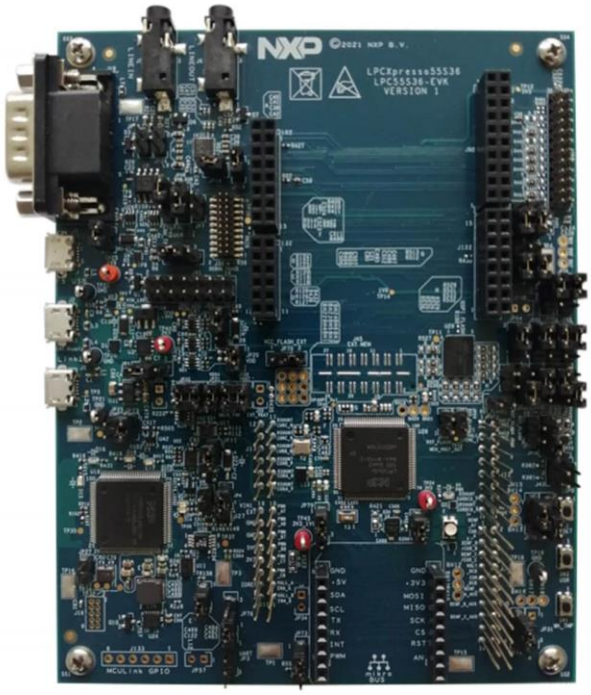
Each category contains multiple examples that showcase different Model-Based Design Toolbox capabilities that are categorized into different groups.

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


Help
NXP LPC553x Blocks
Documentation
Search Documentation
CONTENTS
« Documentation Home
NXP Model-Based Design Toolbox for i.MX RT 1xxx Series (Supplemental Software)
Release Notes
Quick Start Guide
New Project Guide
MCUXpresso Integration Guide
LPC553x Simulink Blocks
Product Homepage
Community & Support
Examples
NXP LPC553x Simulink Blocks
Learn detailed information about each Simulink block that is supported to be configured with this toolbox for LPC553x MCU peripherals.
Blocks
expand all
NXP LPC553x Simulink Blocks
Featured Examples
ADC INTERRUPT EXAMPLE
The lpc55s36_adc_interrupt.mdl example shows the simplest way to use ADC peripheral in interrupt mode, reading data from the internal
Open Example
ADC POLLING EXAMPLE
The lpc55s36_adc_polling.mdl example shows the simplest way to use ADC peripheral.
Open Example
DMA TRANSFER EXAMPLE
The lpc55s36_dma_transfer_tcd.mdl example shows the simplest way to use DMA peripheral for data transfer
Open Example
DMA LPADC EXAMPLE
The lpc55s36_lpadc_dma.mdl example shows the simplest way to use DMA peripheral for ADC data transfer in transactional mode.
Open Example
USART INTERRUPT EXAMPLE
The lpc55s36_usart_interrupt.mdl
USART POLLING EXAMPLE
The lpc55s36_usart_polling.mdl
USART TRANSFER EXAMPLE
The lpc55s36_usart_transfer.mdl
GPIO INTERRUPT EXAMPLE
The lpc55s36_gpio_interrupt.mdl

2.2 Hardware Setup

All examples provided with the Model-Based Design Toolbox were developed on **LPCXpresso55S36 Development Board** as the primary hardware target. Additional information about this development kit can be found on NXP official web page here.

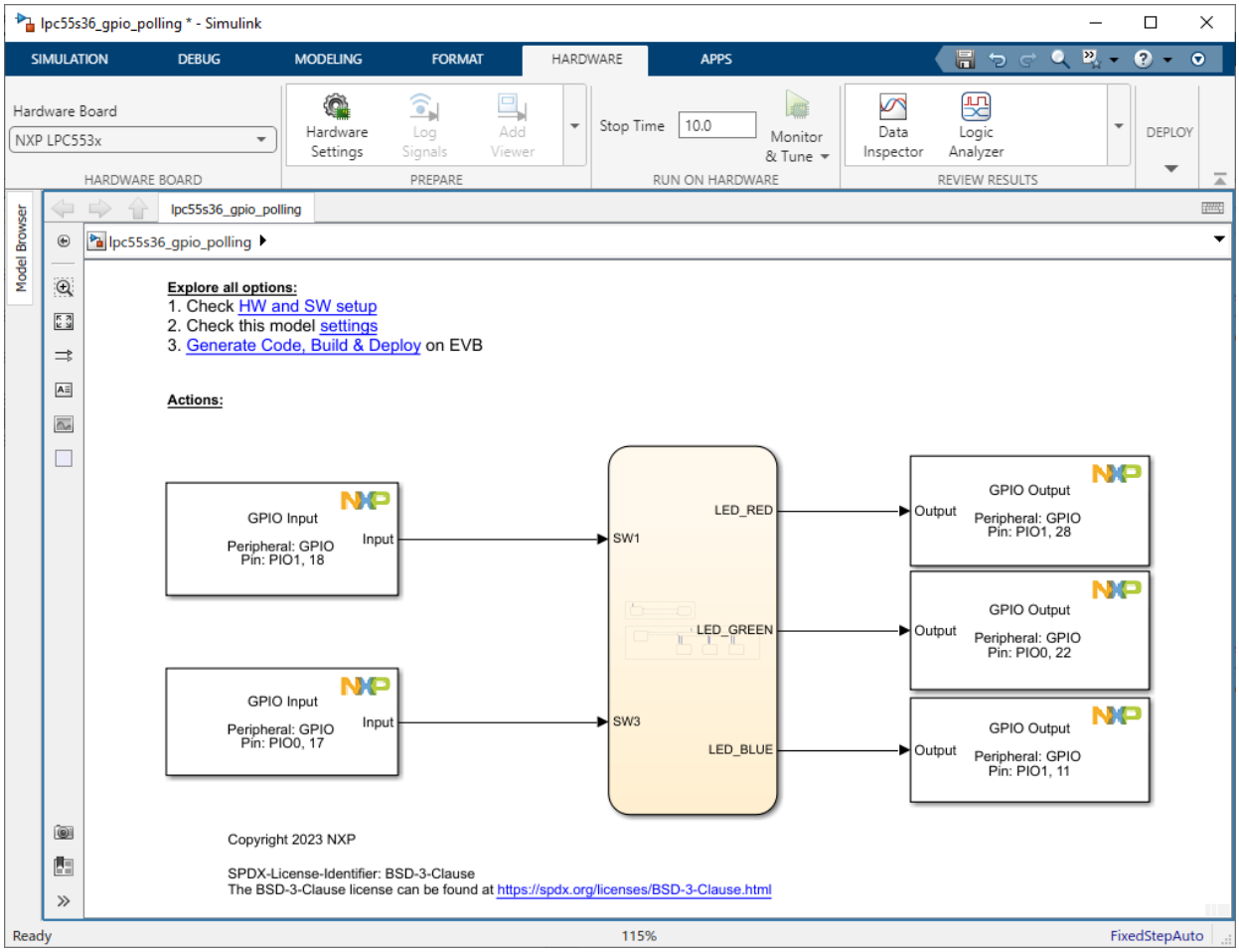


Before running any example on the **LPCXpresso55S36 Development Board**, please connect the board to the PC via a microUSB cable on the Link2 J1 USB connector.

2.3 A “Hello World” Example

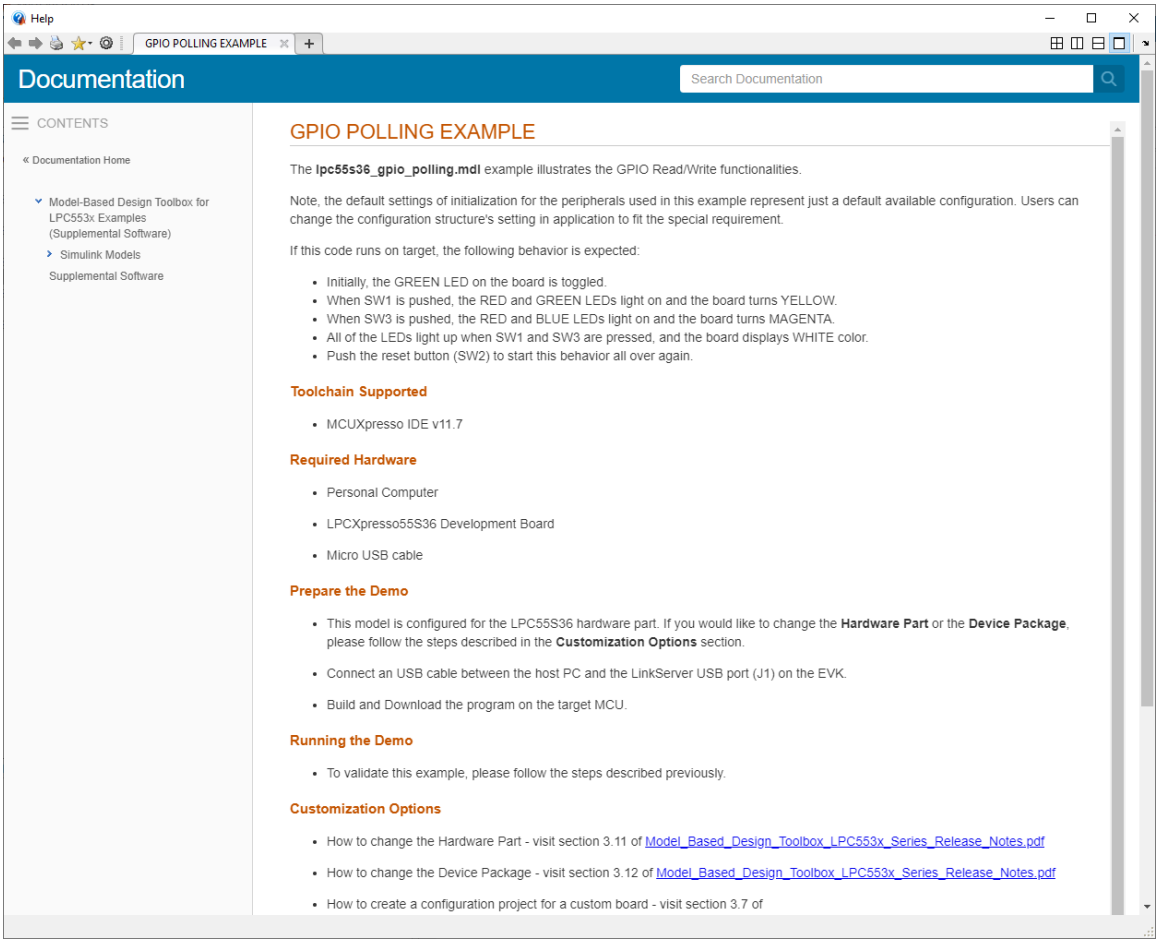
Navigate to “`\LPC553_Examples\gpio\gpio_polling`” folder and open the `lpc55s36_gpio_polling.mdl` Simulink model.

This model illustrates the GPIO Read/Write functionlities by toggling the RGB LED in GREEN. The push buttons controlls the RGB LED BLUE, GREEN and RED on the LPCXpresso55S36 Development Board.

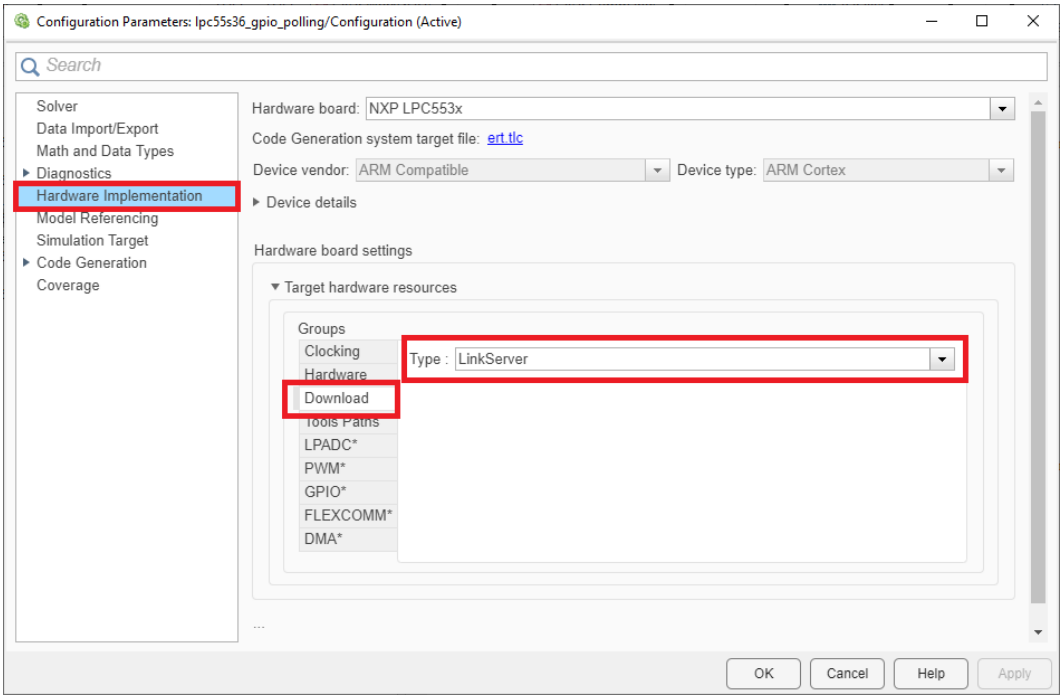


Follow the next steps to run the example:

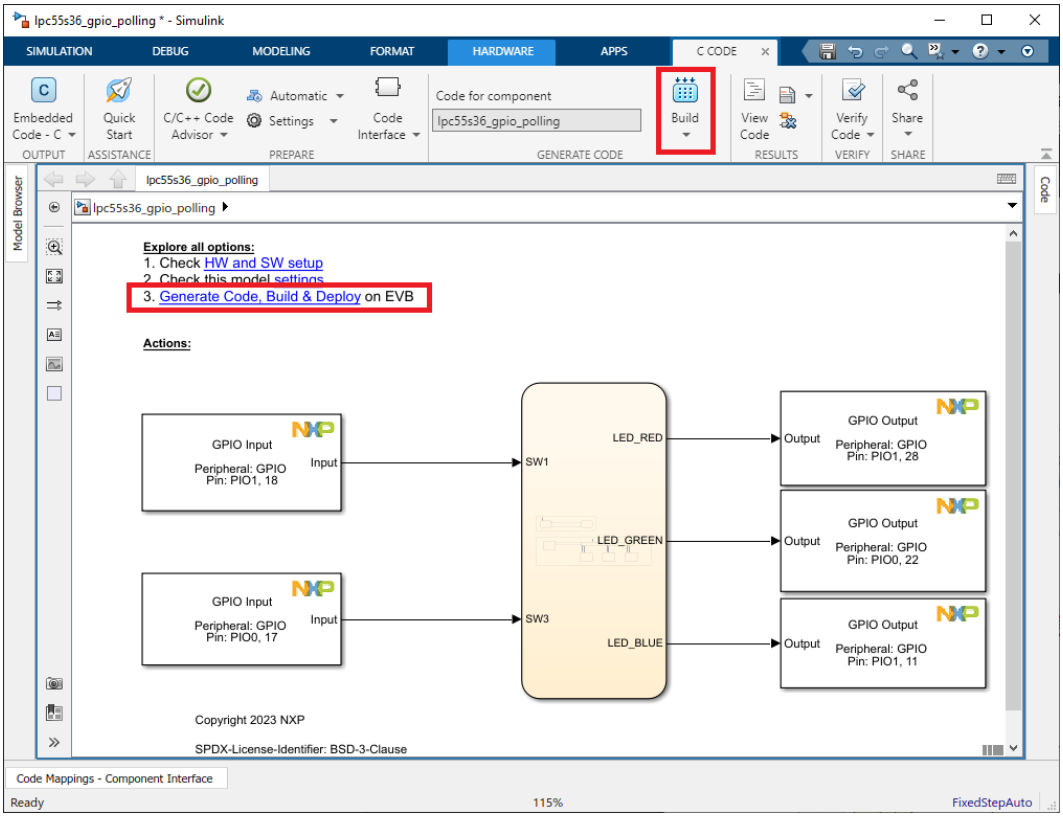
1. Open lpc55s36_gpio_polling_example_readme.html file to understand the hardware and software requirements for running the application



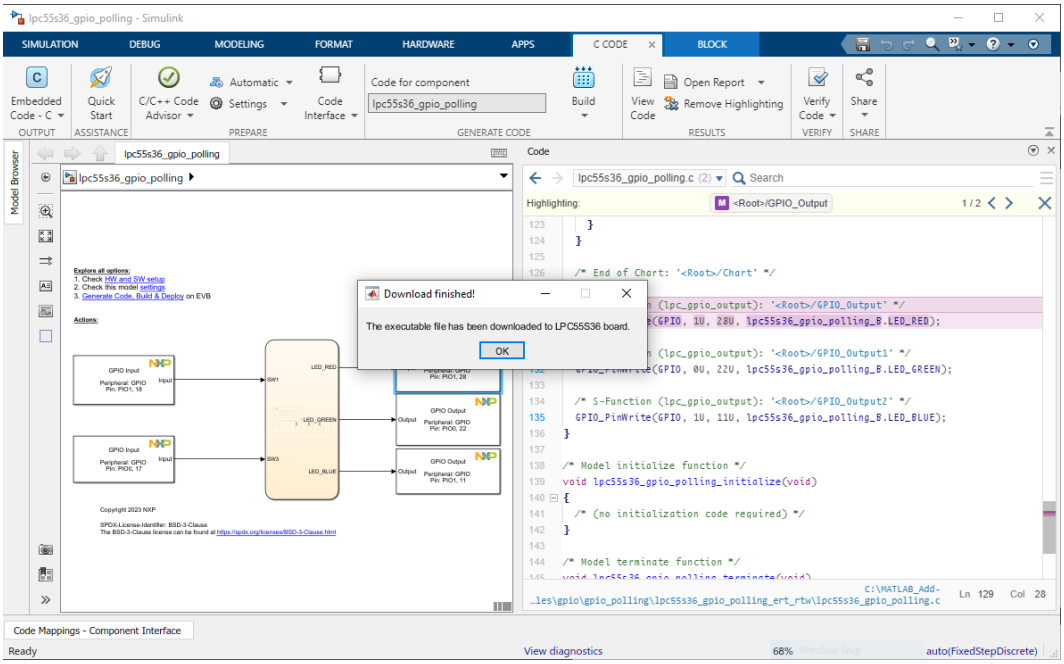
2. Open the Simulink `Model Configuration Parameters` and select the appropriate LinkServer drive used to application download to the LPCXpresso55S36 Development Board.



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. Once the code has been generated, built, and deployed on the target, the “Download finished!” popup appears informing that the code has been successfully deployed on the target.



If successful, the RGB LED shall toggle in Green. If SW1 is pressed, the RGB LED lights on YELLOW. If SW3 is pressed, RGB LED lights on MAGENTA. If both SW1 and SW3 are pressed simultaneously, the RGB LED turns on WHITE.

Congratulations! You succeeded with running your first example created with Model-Based Design Toolbox for LPC553x.

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