

Model Based Design Toolbox

Release Notes

An Embedded Target for the MPC574xP Family of Processors

Version 2.0.0

Target Based Automatic Code Generation Tools

For MATLAB™/Simulink™/Stateflow™ Models working with Simulink Coder™ and Embedded Coder®

Model Based Design Toolbox Release Notes is for use with the Model Based Design Toolbox, an embedded target and block set library for MATLAB/Simulink/Stateflow supporting model based design and code generation.

MATLAB, Simulink, Stateflow, Handle Graphics, and Real-Time Workshop are registered trademarks, and TargetBox is a trademark of The MathWorks, Inc.

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.

Other product or brand names are trademarks or registered trademarks of their respective holders.

© *COPYRIGHT NXP Semiconductors*
2017 All Rights Reserved

1	What's New	1-4
2	MCU Parts Supported	2-5
3	MATLAB Releases Supported	3-5
4	Compiler Support.....	4-6
5	Math and Motor Control Function Support	5-6
6	FreeMASTER Support.....	6-6
7	On-Target Profiling Support.....	7-6
8	Processor-In-the-Loop Support (PIL)	8-7
9	Example Models	9-7
10	Limitations	10-7
11	Known Issues	11-7
12	Additional Information	12-8
12.1	Setting the Path for Model Based Design Toolbox	12-8
12.2	Setting up the Target Compilers.....	12-8
13	Release History	13-8
14	Support Information.....	14-8

1 What's New

1.1 Features:

- ✓ *Support for new Panther XDEVKIT-MPC5744P board (ARDUINO style) which will work with the new chassis XDEVKIT-MOTORGD for motor control applications.*
- ✓ *Integrated latest Automotive Math and Motor Control Library release 1.1.7.*
- ✓ *Support for latest MATLAB versions including 64 bits (2015/2016 a/b)*
- ✓ *New DMA blocks - allowing ADC sampled data to be transferred to memory without CPU intervention through DMA module. A usual scenario of using CTU to trigger ADC sampling and then transfer data via DMA support into memory can now be used.*
- ✓ *New LINFlexD blocks for serial communication support now allow data send/receive operations through UART.*
- ✓ *New Memory Read/Write blocks were added and they can now be used to read/write any memory zone.*
- ✓ *New “Custom Initialization” block was added and it can be used to extend the configuration of any module outside the default setup. Custom Initialization block - allows read/write of any register to have any desired modification before the model first step.*
- ✓ *Added support for S32 Design Studio for Power Compiler.*
- ✓ *New Advanced Motor Control blocks, were added and now new functions like Track Observer or Back EMF observer are provided as Simulink Blocks.*
- ✓ *Aligned ADC clock frequency from 20MHz to 80MHz (max speed).*
- ✓ *New ADC channel configuration block was designed to be used only for configuration of an ADC channel. Previous version was doing configuration and data capture inside the same block not allowing DMA configuration.*

- ✓ *New Diagnostics panel – can be used to enable/disable multiple consistency checks.*
- ✓ *New Bootloader build to support UART1 communication.*
- ✓ *Integrated FreeMASTER release 2.0.2.*

1.2 Fixed issues:

- ✓ *CTU trigger was not working correctly when an external signal was used to trigger ADC.*
- ✓ *ADC sampling rate was not close to 100 kHz value (value was increased 20 times from previous version)*
- ✓ *Generated CAN configuration was not correct. The Clock source and self-received field were not reflecting the mask settings.*
- ✓ *Invalid number of CTU FIFOs was displayed in CTU configuration mask.*
- ✓ *SIL to PIL conversion script did not generate the correct settings.*
- ✓ *The ISR of ADC Watchdog Threshold was not triggering correctly.*
- ✓ *Incorrect priority level interval was shown for all blocks that are using the PSR register.*
- ✓ *Wrong projects settings were generated for MPC5743 and MPC5742.*
- ✓ *SRAM initialization was not done correctly when BAM is used which leads to ECC errors.*

2 MCU Parts Supported

This toolbox supports the MPC5744P, MPC5743P, and MPC5742P MCUs. The tool supports operation with a 40 MHz external XTAL.

3 MATLAB Releases Supported

This tool supports the following MATLAB releases R2015ASP1, R2015B, R2016A and R2016B 32-Bit and 64-Bit versions.

The minimum recommended PC platform is:

- *Windows® OS*: Intel® Pentium® 4 processor, 2 GHz or faster, Intel® Xeon™, Intel® Core™, AMD Athlon™ 64, AMD Opteron™, or later
- At least 4 GB of RAM
- At least 3 GB of free disk space.
- Internet connectivity for web downloads.

Operating System Hosts

Windows			
Host OS	SP Level	32-bit	64-bit
Windows 7	SP1	X	X
Windows 8.1	U1	X	X
Windows 10		X	X

4 Compiler Support

The following compilers are supported and have been tested with the following releases.

Compiler Supported	Release Version
Green Hills MULTI for PowerPC	V2015.1
WindRiver DIAB	V5.9.4.8
S32 DS for Power	V1.1

5 Math and Motor Control Function Support

All functions in the Automotive Math and Motor Control Functions Library (version 1.1.7) are supported as blocks for simulation, SIL, PIL and embedded target code generation.

6 FreeMASTER Support

The Model Based Design Toolbox has built-in code generation support for FreeMASTER through Serial. All features of FreeMASTER are supported with exception of flash programming capability.

7 On-Target Profiling Support

The Model Based Design Toolbox provides blocks to be used for On-Target function profiling that returns results in units of clock cycles of execution per execution iteration.

8 Processor-In-the-Loop Support (PIL)

The Model Based Design Toolbox provides PIL support for purposes of ASIL software development processes, “Model PIL Block” (Model Reference) and “PIL Block” modes of operation are supported “Top Model PIL” mode is not supported. PIL contains full support for Math and Motor Control Blocks, and limited support for peripheral blocks. No support for interrupts exist in supported PIL modes of execution, therefore no blocks with interrupts are supported in PIL mode.

9 Example Models

Example models for different peripheral and motor control blocks are located in the Examples directory `mbdtbx_pnt/examples`. These are working examples of different configurations and use cases of peripheral blocks.

10 Limitations

There is a suite of limitations in compatibility with the previous version (released in 2013) generated by the transition to MATLAB 64 bits, refactoring of the components, integration of new updated (with significant API changes) version of AM & MC libraries and tool name’s change.

While AM & MC libraries updates must be reviewed individually by the user for the peripheral blocks updates, a script to automatically update mdl model files with correct information is AVAILABLE to facilitate existing project conversion.

The script is part of Model Based Design Toolbox for Panther rev 2, at is presented as a function called “`mbd_convert2PNT20`” available in `/mscripts` folder directly from MATLAB prompt. This script converts only peripheral Simulink blocks e.g. ADC (but again not the AMMCLIB blocks).

The script works with the following syntax:

```
MATLAB>
```

```
mbd_convert2PNT20(list_of_mdl_files_to_be_converted_from_current_folder)
```

or a call with empty list converts all mdl files from current folder.

Using the information from the old model a new mdl file is created in current folder with the suffix “`_pnt_2_0`”. The script updates the name of the Configuration block, compiler settings, Toolbox name, target file, make command and consistency checks callbacks.

11 Known Issues

At the release time of the tool, for the new Panther XDEVKIT-MPC5744P board (ARDUINO style) an issue was noticed with the CAN signals strength. The issue is probably related to a hardware condition and not a configuration problem. While it is still under investigation complete testing for CAN could not be completed for this specific board. The Panther Evaluation Board works fine for CAN driver.

12 Additional Information

12.1 Setting the Path for Model Based Design Toolbox

In order for MATLAB to recognize the Model Based Design Toolbox, the path needs to be setup in the MATLAB environment. This is done by navigating the MATLAB Current Directory to the MBD Toolbox/mbdtbx_pnt installation directory and running the “mbd_pnt_path” script.

12.2 Setting up the Target Compilers

The target compiler for Model Based Design Toolbox to use will need to be configured. Use the script commands 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:

```
DIAB_TOOL = C:\WindRiver\diab\5.9.4.8\  
GHS_TOOL = C:\ghs\comp_201516  
S32DS_PA_TOOL = C:\Freescale\S32_Power_v1.1
```

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

13 Release History

None.

14 Support Information

To order NXP products or literature, consult your local sales representative.

For technical support please sign on to the following NXP Model Based Software Design Tools Community: <https://community.nxp.com/community/mbdt>