Model Based Design Toolbox

Release Notes

An Embedded Target for the S32K 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®
### Summary

1. **What is new** ................................................................. 1-3
2. **MCU Support** ............................................................... 2-4
   2.1 Packages & Derivatives .............................................. 2-4
   2.2 Peripherals & Devices .............................................. 2-4
3. **Model Based Design Toolbox Features** .................................. 3-5
   3.1 S32K14x Blocks .................................................. 3-5
   3.2 S32K14x Automotive Math and Motor Control .................... 3-6
   3.3 S32K14x Example Library ........................................ 3-7
4. **Model Based Design Toolbox Extras** ...................................... 4-9
   4.1 FreeMASTER Support ............................................ 4-9
   4.2 On-Target Profiling Support .................................... 4-9
   4.3 Processor-In-the-Loop Support (PIL) .......................... 4-9
   4.4 Boot Loader ..................................................... 4-9
   4.5 Miscellaneous ................................................... 4-9
5. **Prerequisites** .................................................................. 5-10
   5.1 MATLAB Releases and OSes Supported .......................... 5-10
   5.2 Compiler Support .................................................. 5-10
6. **Known Limitations** ............................................................. 6-11
7. **Support Information** .......................................................... 7-11

---

*Model Based Design Toolbox 2.0.0*

*License Installation* for S32K family of processors
1 What is new

The latest Model Based Design Toolbox designed to support S32K MCU brings the following important enhancements:

- Support for the new S32K144 MCU and S32K144EVB-Q100 evaluation board that can work with the new chasses XDEVKIT-MOTORGD for motor control and XDEVKIT-COMM for CAN and LIN communication;

- Integrates the latest Automotive Math and Motor Control Library release 1.1.8 for ARM M4 cores;
- Integrates the latest SDK release version 1.0.0 for S32K MCU;
- Integrates FreeMASTER release 2.0.2;
- Support for latest MATLAB versions including 64 bits (2015b, 2016a/b and 2017a);
- New Example Library for main S32K peripherals and comprehensive applications;
- 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;

For more details please refer to the following chapters.
2 MCU Support

2.1 Packages & Derivatives
This release supports the S32K144 MCU Packages:
- 100 BGA;
- 64 LQFP;
- 100 LQFP;

The toolbox supports operation with 8MHz and 40MHz external XTAL and MCU system clock frequencies of 80MHz and 112MHz.

2.2 Peripherals & Devices
The peripherals, devices and functions supported by Model Based Design Toolbox are highlighted on top of the S32K144 block diagram.
3 Model Based Design Toolbox Features

The Model Based Design Toolbox for S32K Series consists in three main Simulink Libraries:

- S32K14x Blocks;
- S32K14x Automotive Math and Motor Control;
- S32K14x Examples;

3.1 S32K14x Blocks

This groups all the Simulink blocks designed to support the S32K MCU peripherals and devices. It also exposes specific Simulink blocks needed for MCU configuration, interfacing with FreeMASTER, custom code initialization and profiling.
The main functionalities supported are:

- **Peripheral Interface Blocks** – *General Purpose Digital Input and Output* with Interrupts and Glitch filtering;
- **Motor Control Blocks** – includes *Flex Timer Module* for independent / complementary / 3-phase for asymmetric or symmetric PWM generation with different resolution, frequency update and dead-time insertion at run-time, edge capture for single / dual / quadrature and interrupts, *Compare* unit with interrupts generation, *Analog to Digital Convertor* with SW an HW interrupts and *Programmable Delay Block* to generates HW interrupts from different HW triggers to different modules;
- **Communication Blocks** – supports configuration, data read, data write and Interrupts for UART, SPI, I2C and CAN;
- **Utility Blocks** – supports miscellaneous functionalities helpful for application development as generic timer interrupts, custom code, profiling, data read/write;

### 3.2 S32K14x Automotive Math and Motor Control

S32K14x Automotive Math and Motor Control supports the ARM M4 core based devices and consists of several sub-libraries, functionally connected.

All functions in the Automotive Math and Motor Control Functions Library v1.1.8 are supported as blocks for simulation and embedded target code generation for:

- `bam_f16` – 16-bit fixed-point;
- `bam_f32` – 32-bit fixed-point;
- `bam_flt` – single precision floating point;

The main functionalities supported are:
- Mathematical Function Library (MLIB) – supports basic mathematical operations such as addition, multiplication, etc;
- General Function Library (GFLIB) – supports basic trigonometric and general math functions such as sine, cosine, tan, hysteresis, limit, etc;
- General Digital Filters Library (GDFLIB) – comprising digital IIR and FIR filters designed to be used in a motor control application;
- General Motor Control Library (GMCLIB) – supports standard algorithms used for motor control such as Clarke/Park transformations, Space Vector Modulation, etc;
- Advanced Motor Control Function Library (AMCLIB) – comprising advanced algorithms used for motor control purposes;

### 3.3 S32K14x Example Library

S32K14x Examples Library represents a collection of Simulink models that let you test different MCU on-chip modules and run complex applications. These examples are grouped in different layers that mimics a typical development flow: starting with basic building blocks that expose the MCU HW functionalities, build SIL and PIL models for verification and validation purposed and ending up with more complex applications that incorporates multiple building blocks.

The main functionalities supported are:
- On-chip modules and peripherals examples layer covers the basic functionalities of the S32K MCU;
- Modelling, Verification and Validation examples layer covers the SIL, PIL and additional blocks supported by MBD Toolbox to help faster prototyping;
- Applications layer covers the AMMCLIB and Motor Control scenarios;

The Simulink models shown as examples are enhanced with a comprehensive description to help users understand better the functionality that is exercised, hardware setup instructions whenever are necessary and a result validation section.
4 Model Based Design Toolbox Extras

The Model Based Design Toolbox enables additional functionalities that are not MCU specific but could help faster prototyping, validation and verification of the developed models.

4.1 FreeMASTER Support

The Model Based Design Toolbox has built-in code generation support for FreeMASTER through LPUART and FlexCAN interface support. The FlexCAN interface works with Vector CAN hardware, and IXAAT CAN interface cards.
All features of FreeMASTER are supported with exception of flash programming capability.

4.2 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.

4.3 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.

4.4 Boot Loader

Internal Boot Loader is a standalone application which requires the Microsoft .NET Framework version 4.0 installed on PC. It may be required to download the package from http://www.microsoft.com/download and install if you are going to use PIL and internal Boot Loader. The board should be configured to work with BAM to use Internal Boot Loader. Please check board documentation.

4.5 Miscellaneous

It is possible to add the user's files into a project supported using the user defined script, please refer to reference manual for details.
Model Referencing is supported but there are limitations with ISRs, please refer to reference manual for details.
5 Prerequisites

5.1 MATLAB Releases and OSes Supported
This toolbox is developed and tested to supports the following MATLAB releases:
- R2015B;
- R2016A;
- R2016B;
- R2017A;
- 32/64-Bit versions;

The minimum recommended PC platform is:
- Windows® OS: any Intel® or AMD® x86-64 processor
- At least 4 GB of RAM
- At least 6 GB of free disk space.
- Internet connectivity for web downloads.

### Operating System Hosts

<table>
<thead>
<tr>
<th>Operating System</th>
<th>SP Level</th>
<th>32-bit</th>
<th>64-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 7</td>
<td>SP1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows 8.1</td>
<td>U1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows 10</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

5.2 Compiler Support
The following compilers are supported:

<table>
<thead>
<tr>
<th>Compiler Supported</th>
<th>Release Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCC for ARM Embedded Processors</td>
<td>V4.9.3</td>
</tr>
<tr>
<td>IAR ANSI C/C++ Compiler</td>
<td>V7.5.3</td>
</tr>
<tr>
<td>GreenHills MULTI for ARM</td>
<td>2015.1.4</td>
</tr>
</tbody>
</table>

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:

```
GCC_S32K_TOOL = C:/NXP/S32DS_ARM_v1.3/Cross_Tools/gcc-arm-none-eabi-4_9
IAR_TOOL = C:/Program Files (x86)/IAR Systems/Embedded Workbench 7.3
GHS_TOOL = C:/ghs/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 environment to see these.
6 Known Limitations
- Due to an issue related with Matlab 2017a makefile generation that is currently under Mathworks investigation the PIL_Block are not functional with this 2017a Matlab release. Revert to Matlab 2015b, 2016a or 2016b releases instead.

- The Simulink Function-Call Split block cannot be mixed with Model Based Design Toolbox blocks.

7 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