Model-Based Design Toolbox
S32K3xx Series

Release Notes

Automatic Code Generation for the S32K3xx Family of Processors
Version 1.1.0
# Summary

1  Main Features ........................................................................................................ 1-3

2  S32K3xx MCU Support ............................................................................................. 2-4

  2.1  Packages & Derivatives ....................................................................................... 2-4

  2.2  Peripherals and functions ................................................................................. 2-5

3  Model-Based Design Toolbox Features .................................................................. 3-6

  3.1  S32K3xx Simulink Library Blocks ....................................................................... 3-7

  3.2  Support for battery cell controller for BMS applications (MC33775A & MC33664PHY) ................................................. 3-10

  3.3  Motor control examples ...................................................................................... 3-11

  3.4  S32K3xx AMMCLib integration ......................................................................... 3-12

  3.5  S32K3xx FreeMASTER integration ..................................................................... 3-13

  3.6  S32K3xx Design Studio integration ..................................................................... 3-14

  3.7  Support for custom default project .................................................................... 3-15

  3.8  S32K3xx Simulation modes ................................................................................. 3-16

  3.9  S32K3xx Example Library ................................................................................. 3-17

4  Workflow and framework changes ......................................................................... 4-19

  4.1  Using external configuration tools ..................................................................... 4-19

    4.1.1  Modes of operation – BASIC and ADVANCED ............................................. 4-20

    4.1.2  Advantages for using external configuration tools ...................................... 4-20

  4.2  Code generation based on RTD support (MCAL drivers) ................................. 4-21

5  Prerequisites ........................................................................................................... 5-22

  5.1  MATLAB Releases and OSes Supported ............................................................ 5-22

  5.2  Build Toolchain Support ................................................................................... 5-22

  5.3  RTD Support ...................................................................................................... 5-24

  5.4  S32 Configuration Tool Support ......................................................................... 5-25

6  Known Limitations .................................................................................................... 6-26

7  Support Information ................................................................................................. 7-26
1 Main Features

The **NXP’s Model-Based Design Toolbox for S32K3xx Series version 1.1.0** is designed to support S32K344 and S32K312 MCUs into MATLAB/Simulink environment, allowing users to:

- **Design** applications using Model-Based Design methodologies;
- **Simulate** and **Test** Simulink models for S32K3 MCU before deploying the models to the hardware targets;
- **Configure** the MCU peripherals from Simulink Blocks via external configuration tools: NXP S32 Configuration Tools or EB Tresos
- **Generate** the application code automatically without any needs for hand coding C/ASM
- **Deployment** of the application directly from MATLAB/Simulink to the NXP evaluation boards

The main features and functionalities supported the first RTM release (v.1.1.0) are:

- Support for S32K344 and S32K312 MCUs
- Integrated with S32Configuration Tool (v2021.R1.4) and EB Tresos (v27.1.0)
- Integrates the latest Real-Time Drivers (v1.0.0)
- Added support for MC33775A BCC and MC33664 PHY
- Added motor control examples (both for PMSM and BLDC)
- Compatible with MATLAB releases R2020a, R2020b, R2021a, R2021b
- Fully integrated with Simulink Toolchain
- Includes extensive Simulink Library Blocks for S32K344 and S32K312 MCUs, providing dedicated NXP blocks for peripheral components (e.g.: ADC, PWM, CAN, DIO, GPT, SPI, UART)
- Supports FreeMASTER v3.x
- Supports AMMCLib v1.1.26
- Supports AUTOSAR blockset (SW-C deployment)
- Supports download via JTAG and OpenSDA
- Includes an Example library with approximately 100 examples that cover a wide range of topics like:
  - I/O control
  - Timers and scheduling
  - Communication
  - Motor control (3-shunt BLDC and PMSM)
  - Software-in-Loop, Processor-in-Loop, External mode
  - For more details about each of the topics highlighted above please refer to the following chapters.

2 S32K3xx MCU Support

2.1 Packages & Derivatives

The NXP’s Model-Based Design Toolbox for S32K3xx Series version 1.1.0 supports:

- S32K3xx MCU Packages:
  - S32K344-257MAPBGA
  - S32K344-172MAXQFP
  - S32K312-172MAXQFP

The configurations can be easily changed for each Simulink model from the Model Configuration Parameters menu:
2.2 Peripherals and functions

The NXP’s Model-Based Design Toolbox for S32K3xx Series version 1.1.0 supports the following peripheral components and functions:

- ADC
- PWM
- DIO
- MCL
- CAN
- SPI
- UART
- GPT
- Memory read/write
- Register read/write
- Profiler

The default configuration supported by the toolbox for each peripheral is available inside the Target Hardware Resources panels:

From this panel, the user can access the hardware documentation (Reference Manual) using the Details… button. To update the model configuration, the user can click on the Configure… button to open the NXP S32 Configuration Tool / EB Tresos. The selected external configuration tool will be opened and the user will have access to the entire parameter list for pins / clock / peripherals.

The NXP’s Model-Based Design Toolbox for S32K3xx Series version 1.1.0 has been tested using the official NXP Evaluation Boards for S32K3xx – XS32K3XXEVB-Q257.
3 Model-Based Design Toolbox Features

The NXP’s Model-Based Design Toolbox for S32K3xx Series version 1.1.0 is delivered with complete S32K3xx MCUs Simulink Block Library as shown below.

There are three main categories:
- S32K3xx AMMCLib
- S32K3xx Core, System, Peripherals and Utilities
- S32K3xx Example Projects
3.1 S32K3xx Simulink Library Blocks

All Simulink blocks supported by the NXP Toolbox are designed to offer the best out-of-the-box experience, providing a basic peripheral configuration that covers most of the hardware capabilities. Most of the Simulink Blocks contains just two tabs:

1. **General** Tab allows the user to select what functionality he wants the block to enable (usually this will be a list of MCAL functions). Besides that, the user can edit some parameters (this will be different, depending on the function selection). All peripheral components come with a working configuration (the toolbox default setup). These settings have been chosen to work out-of-the-box for the S32K3x4EVB. The toolbox can be used as-is if no configuration parameters have to be changed. This is the BASIC mode of operation for the toolbox. In the ADVANCED mode, the user can click on the **Configure...** button, which will open an external configuration tool (NXP S32Configuration Tool / EB Tresos). There, the user has full access to all the parameters needed to set up the pins/clock/peripherals.
2. **Parameters** Tab which contains the detailed configuration available. This information is shown for verification purposes.

From any of these blocks, by clicking on the **Configure...** button, the users can open the external configuration tool of choice (NXP S32 Configuration Tools / EB Tresos) to alter the default configuration used by the Simulink model.
Using the external tools, the users can perform ADVANCED configurations and then use these new options in Simulink models. The validation of the configuration and peripheral code generation is done outside of Simulink.
3.2 Support for battery cell controller for BMS applications (MC33775A & MC33664PHY)

The toolbox provides support for the MC33775A and MC33664. The MC33775A is a lithium-ion battery cell controller IC designed for automotive applications which perform ADC conversions of the differential cell voltages and battery temperatures, while the MC33664 is a transceiver physical layer transformer driver, designed to interface the microcontroller with the battery cell controllers through a high speed isolated communication network. The ready-to-run example provided with the MBDT for S32K3, shows how to communicate between the S32K344 and the MC33775A via the MC33664 transceiver. The MCU configures the battery cell controller to perform Primary and Secondary chains conversion, reads the cell voltages conversion results from the MC33775A, and displays the values to the user over the FreeMaster.
3.3 Motor control examples

The toolbox provides examples for both 3-shunt PMSM and BLDC motor control applications. Each of them has a detailed description of the hardware setup and an associated FreeMASTER project which can be used for control and data visualization.
3.4 S32K3xx AMMCLib integration
This toolbox also ships the AMMCLib v1.1.26 blocks. These blocks are essential for building applications with high-performance arithmetic, trigonometric, digital signal processing and math functions.
3.5 S32K3xx FreeMASTER integration

This toolbox has support for FreeMASTER 3.x – this is a user-friendly real-time debug monitor and data visualization tool that enables runtime configuration and tuning of embedded software applications. With this support, users can build models and monitor various variables, display them on multiple scopes and even add widgets (gauges, sliders, etc).
3.6 S32K3xx Design Studio integration

The toolbox provides users the opportunity to export the code generated from Simulink (using the S32K3xx toolbox) and easily import it into S32Design Studio. This functionality can be useful if the model is to be integrated into an already existing, larger project.
3.7 Support for custom default project

The toolbox provides support for users to create their own custom default project. This could be very useful when having a custom board design – the configuration for it needing to be created only once. After that configuration is saved as a custom default project, it can be used for every model that is being developed.
3.8 S32K3xx Simulation modes

The toolbox provides support for the following Simulation modes:
- Software-in-Loop (SIL)
- Processor-in-Loop (PIL)
- External Mode

Software-in-Loop
A SIL simulation compiles and runs the generated code on the user’s development computer. One can use such a simulation to detect early defects and fix them.

Processor-in-loop
In a PIL simulation, the generated code runs on the target hardware. The results of the PIL simulation are transferred to Simulink to verify the numerical equivalence of the simulation and the code generation results. The PIL verification process is a crucial part of the design cycle to ensure that the behavior of the deployment code matches the design.

External mode
In this mode, the user can leverage Simulink to link algorithm code with the I/O driver code generated from the I/O blocks (from MATLAB). The resulting executable runs on the development computer and exchanges parameter data with Simulink via a shared memory interface.
3.9 S32K3xx Example Library

The Examples Library represents a collection of Simulink models that let you test different MCU on-chip modules and run complex applications.

The examples are grouped in different layers that mimic a typical development flow: starting with basic building blocks that expose the MCU HW functionalities up to more complex applications that incorporate multiple building blocks.

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.

The examples are also available from the MATLAB help page.
DIO Example - OVERVIEW

The s32k3xx_dio.m file example shows how to use DIO ReadChannel, WriteChannel and FlipChannel functionalities.

After the application is deployed on target, the D03 LED should blink RED every 0.1 second. If SW4 is pressed, D02 should turn on BLUE. The state of the D03 RED LED is stored inside the red led_level variable. The versionInfo variable stores the version information of this module, including model id, vendor id and vendor specific version numbers.

Note: the initialization settings for the components used in this example represent just a default EFR Teens available configuration. Users can change the configuration of the application to fit special requirements.

TOOLCHAIN SUPPORTED
- S32 Design Studio 3.4

HARDWARE REQUIREMENTS
- Mini micro USB cable
- EVB-S32K3XX board
- Personal Computer

PREPARE THE DEMO
- Connect a USB cable between the host PC and the OpenSDA USB port on the target board.
- Open the s32k3xx_dio.m. Read all the instructions within the model.
- Build and Download the program to the target board.

RUNNING THE DEMO
- The D03 LED should blink RED every 0.1 second. If SW4 is pressed, D02 should turn on BLUE.

CUSTOMIZATION OPTIONS
Fully customizable for any input and output pins.
4 Workflow and framework changes

4.1 Using external configuration tools

One of the major differences between older toolboxes (e.g. for S32K1xx) and MBDT for S32K3xx is the introduction of a new workflow, where the modeling and configuration are split. In other toolboxes, every peripheral had associated a Configuration Block. There, the user could set up some important parameters (e.g.: UART baud rate):
4.1.1 **Modes of operation – BASIC and ADVANCED**

Starting with this release, all peripheral Configuration Blocks are no longer needed. Instead, we provide a default configuration for all the peripherals we support. Users can see the default parameters for every such block, going in the mask, searching for the ‘Configuration Parameters’ tab. This use case is what we call the BASIC mode of operation – where there is no additional configuration to be made (everything is based on the default one), so no interaction with an external configuration tool is needed.

If something needs to be edited/added/removed, users can open an external configuration tool of choice (NXP S32Configuration Tools or EB Tresos). There, all the parameters of peripherals, pins and clock are available. The Simulink model is linked to the external configuration tools and is aware of changes made, so simply building the model one more time/clicking on the ‘Refresh’ button of the block fetches the new values of the parameters. This is the ADVANCED mode, where changes to the default configuration need to be made. Note that NXP S32Configuration Tools is shipped with our toolbox, so no extra installation process needed; EB Tresos needs to be installed and licensed (free of charge), following the steps in the MBDT Quick Start Guide.

4.1.2 **Advantages for using external configuration tools**

There are a couple of key advantages to using an external configuration tool instead of the peripheral configuration blocks. Firstly, using these dedicated tools, the user has access to all parameters for the peripherals. Additionally, full access to pins and clock configuration is provided (while the older framework has limited support for this). Secondly, splitting the modeling (algorithms, state machines, etc) from the configuration means they can be done in parallel. This can be an advantage when applications are developed in larger teams (one engineer might work on the configuration for all pins, clock, peripherals while another engineer can focus on modeling a complex algorithm in Simulink). Additionally, different configuration setups can be easily switched, without having to change anything inside the Simulink model.
4.2 Code generation based on RTD support (MCAL drivers)

In this release, the code generation is based on the Real Time Drivers. The peripheral support is different from other releases (e.g. for S32K1xx), not having a per-peripheral strategy, but a per MCAL component/driver one. For example, LPIT (low power interrupt timer) and STM (system timer module) support will not be done via separate blocks, but within one GPT (general purpose timer) block. The structure will be similar for all such blocks: a list of supported functions (MCAL standard functions and additional functions, added for optimization purposes) and a few parameters (depending on selected functionality).
5 Prerequisites

5.1 MATLAB Releases and OSes Supported
This toolbox is developed and tested to support the following MATLAB releases:
- R2020a;
- R2020b;
- R2021a;
- R2021b;

For a flawless 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

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

5.2 Build Toolchain 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>V10.2.0</td>
</tr>
</tbody>
</table>

The target compiler for the Model-Based Design Toolbox needs to be configured.

The Model-Based Design Toolbox uses the Toolchain mechanism exposed by the Simulink to enable automatic code generation with Embedded and Simulink Coder toolbox. By default, the toolchain is configured for the MATLAB 2020a, 2020b, 2021a and 2021b releases. For any other MATLAB release, 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\NXPMBDToolbox_S32K3xx\`) and running the “mbd_s32k3_path.m” script.

```bash
>> mbd_s32k3_path
Treating 'C:\Users\nxa14941\Documents\MATLAB\Add-Ons\Toolboxes\NXPMBDToolbox_S32K3xx\' 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_s32k3_path.m” script verifies the user setup dependencies and will issue instructions for a successful installation and configuration of the toolbox.

The toolchain can be further enhanced using the Simulink Model Configuration Parameters menu:
5.3 RTD Support

The toolbox is delivered with RTD 1.0.0 support. The user can change the RTD package from the Simulink Model Configuration Parameters menu.
5.4 S32 Configuration Tool Support

The toolbox is delivered with Configuration Tools v2021.R1.4. The user can change the Configuration Tools version used by the Simulink from the Simulink Model Configuration Parameters menu.
6 Known Limitations

The list of known limitations can be found in the readme.txt file that is delivered with the toolbox and can be consulted in the MATLAB Add-on installation folder of the Model-Based Design Toolbox for S32K3xx Series.

7 Support Information

For technical support please sign on to the following NXP’s Model-Based Design Toolbox Community: https://community.nxp.com/t5/NXP-Model-Based-Design-Tools/bd-p/mbdt