

S32 Design Studio for ARM 2.2

Contents

1. Release Description.....	2
1.1. Release content.....	2
1.2. Supported devices.....	3
2. What's New.....	3
2.1. New/Updated device support.....	3
2.2. Removed device support.....	3
2.3. New features.....	3
3. System Requirements for Windows Host.....	3
3.1. Recommended configuration.....	3
3.2. Operational minimum configuration.....	4
3.3. Host operating system support.....	4
4. System Requirements for Linux Host.....	4
4.1. Recommended configuration.....	4
4.2. Operational minimum configuration.....	4
4.3. Host operating system support.....	5
5. Product Web Page.....	5
6. Installation and Licensing.....	5
7. Technical Support.....	5
Appendix A. Known issues and Workarounds.....	6
Appendix B. Performance Considerations.....	8



1. Release Description

NXP Semiconductors is pleased to announce the release of the S32 Design Studio for ARM 2.2 for Automotive and Ultra-Reliable MCUs. S32 Design Studio is based on the Eclipse open development platform and integrates the Eclipse IDE, GNU Compiler Collection (GCC), GNU Debugger (GDB), and other software to offer designers a straightforward development tool with no code-size limitations.

1.1. Release content

- Eclipse Neon 4.6 Framework
- GNU Bare-Metal Targeted Tools for Arm[®] 32-bit Embedded Processors (GCC version 6.3.1 20170509, revision g7fea41d)
- GNU Tools for Arm[®] Embedded Processors (Launchpad) (GCC version 4.9.3 20150529, revision 227977)
- Libraries: NewLib, NewLib Nano, EWL, and EWL Nano ¹
- Semihosting for Arm[®] 32-bit Bare-Metal Targeted toolchain
- MSYS2 32bit version 1.0.0
- GDB 7.12.1 with Python support
- The wizards for creating application, library projects and projects from project examples ²
- The S32DS Extensions and Updates tool
- Support for importing projects from CodeWarrior for MCU v.10.6 and Kinetis Design Studio
- Support for importing MCAL configuration to a custom SDK
- IAR v7.x and v8.11.2 compiler support
- Green Hills compiler support
- SEGGER J-Link (with SEGGER GDB Server)
- iSystem, Lauterbach and IAR debuggers support³
- PEMicro debugger support (P&E Multilink/Cyclone/OpenSDA)
- Kernel Aware debugging for FreeRTOS, OSEK
- MQX 4.2 for MAC57D54H
- EmbSys Registers view support
- FreeMASTER Serial Communication Driver v2.0
- Sample Drivers for KEA family (Evaluation grade)
- SDK management
- Integrated S32SDK RTM 3.0.2
- Integrated AMMCLIB SDKs 1.1.18
- The Getting Started page

¹The availability of libraries depends on the toolchain version and the core type.

²Support for wizards and the project examples are provided in the device specific software packages.

³The plug-ins to support third-party compilers or debuggers (IAR, GreenHills, iSystem, Lauterbach) are not included in the installation package and have to be installed from the corresponding sites.

1.2. Supported devices

- SKEAZN8, SKEAZN16, SKEAZN32, SKEAZN64, SKEAZ64, SKEAZ128
- S32K116, S32K118, S32K142, S32K144, S32K146, S32K148
- MAC57D54H
- MWCT1014S, MWCT1015S, MWCT1016S

2. What's New

This release comes with several new features, improvements and bug fixes.

2.1. New/Updated device support

- S32K116, S32K118
- MWCT1014S, MWCT1015S, MWCT1016S

2.2. Removed device support

- S32V234
- AArch64 toolchain

Note: Support of removed devices is provided in the S32 Design Studio 3.2.

2.3. New features

- The S32DS Extensions and Updates tool for automatic lookup and on-demand installation of software packages with SDKs and support for the NXP ARM[®] based processor families.
- The Getting Started page enables navigation through the S32 Design Studio structure and provides collateral materials such as videos and How-To articles.
- The default product directory structure is modified.

3. System Requirements for Windows Host

3.1. Recommended configuration

- PC with 2.6 GHz Intel[®] Pentium[®] compatible processor or better
- 4 GB of RAM
- 4.5 GB of disk space
- 3 GB of temporary storage (required only during the product installation)
- USB port for communications with target hardware
- Ethernet port for communications with target hardware (optional)

3.2. Operational minimum configuration

- PC with 1.8 GHz Intel® Pentium® compatible processor
- 2 GB of RAM
- 4.5 GB of disk space
- 3 GB of temporary storage (required only during the product installation)
- USB port for communications with target hardware

3.3. Host operating system support

- Microsoft® Windows® 7 32-bit and 64-bit
- Microsoft® Windows® 8/8.1 32-bit and 64-bit
- Microsoft® Windows® 10 32-bit and 64-bit

S32 Design Studio for ARM 2.2 supports all editions of the operating systems listed above and is limited only by the requirements of the Java Runtime Environment.

4. System Requirements for Linux Host

4.1. Recommended configuration

- PC with 2.6 GHz Intel® Pentium® compatible processor or better
- 4 GB of RAM
- 4 GB of disk space for installation files (full product or updates)
- 2.5 GB of temporary storage (required only during the product installation)
- Java 1.8
- USB port for communications with target hardware
- Ethernet port for communications with target hardware (optional)

S32 Design Studio for ARM 2.2 Installation Guide specifies the installation prerequisites for Linux platforms.

4.2. Operational minimum configuration

- PC with 1.8 GHz Intel® Pentium® compatible processor
- 2 GB of RAM
- 4 GB of disk space for installation files (full product or updates)
- 2.5 GB of temporary storage (required only during the product installation)
- Java 1.8
- USB port for communications with target hardware

4.3. Host operating system support

- Ubuntu LTS 16.04 64-bit
- Debian 8 64-bit
- CentOS 7 64-bit

S32 Design Studio for ARM 2.2 supports all editions of the operating systems listed above and is limited only by the requirements of the Java Runtime Environment.

5. Product Web Page

S32 Design Studio for ARM 2.2 is available for download from the product Web page at www.nxp.com.

6. Installation and Licensing

The S32 Design Studio for ARM 2.2 installation package contains the base tools and the installer. Support for the NXP Arm[®] based processor families is provided with additional software packages. The packages can be installed on the product from the S32DS Extensions and Updates tool or downloaded from the product Web page.

Run the installation package. The wizard will guide you through the installation process.

Note: Installation of S32 Design Studio for ARM 2.2 from the command line interface in the console or silent mode is not supported.

During installation, license activation is requested. The following activation types are supported:

- Online activation requires access to the Internet. A license activation request is sent automatically.
- Offline activation requires no Internet access. A license activation request is generated and needs to be passed to the licensing site manually. The activation response is loaded from the site to the installer manually.

New functionality can be added to S32 Design Studio for ARM 2.2 with additional software packages, updates and patches. Software packages add support for specific NXP Arm[®] based processor families. Updates and patches extend general functionality of the product and correct software defects.

New functionality can be added directly from the Internet or from a downloaded archive. If your computer is connected to the Internet, select **S32DS Extensions and Updates** on the **Help** menu to find and install all updates and packages available. If your computer does not have access to the Internet, you can download software packages, updates, and patches from the product page and install them from the archive files using the S32DS Extensions and Updates tool.

7. Technical Support

- S32 Design Studio general issues are tracked through the S32DS Public NXP Community space: <https://community.nxp.com/community/s32/s32ds>

- The S32 SDK issues are tracked through the S32 SDK Public NXP Community space:
<https://community.nxp.com/community/s32/s32sdk>
- For confidential cases and cases which cannot be publicly shared on NXP Community please follow the steps described here:
<https://community.nxp.com/docs/DOC-329745>
- Submit a support request for P&E Microcomputer Systems:
<https://www.pemicro.com/support/>

Appendix A. Known issues and Workarounds

- **Conditional watchpoints and breakpoints:** Conditional breakpoints and watchpoints, including those using ignore counts, may not work in some cases.
Workaround: Avoid using conditions for breakpoints and watchpoints, instead check for condition in the code and set a normal breakpoint.
- **Disassembly view** content might be destroyed occasionally.
Workaround: Close the Disassembly view and open it again by selecting **Window > Show View > Disassembly** on the menu.
- **USB driver conflict** between P&E and Xilinx software: P&E Drivers for Windows use the 3rd party service Jungo WinDriver to enable USB support under windows XP/7/8.x/10. Xilinx also uses WinDriver for their driver packages, but older versions. Starting with the P&E Driver package 11.7, a conflict between P&E Drivers and software IDEs that rely on older Jungo driver packages might occur. Both the old and new driver packages utilize the same .sys file (windrvr6.sys). When the new P&E Drivers are installed, Xilinx tools stop working after 30 days because the new .sys file requires a new license that the older Xilinx software does not have.
Workaround: Follow the steps in P&E's FAQ 255 for installing an older version of P&E Drivers (11.1) to resolve Xilinx conflict under Windows XP/7/8.x: http://www.pemicro.com/website_management/faq_view.cfm?id=225. If the drivers issue persists, submit a support request for P&E Microcomputer Systems.
- The **SDK attach mechanism** may report error about missing files if more than one SDK attached and they have conflicting files.
- The error is reported for project with **IAR toolchain**, if the Properties view is active and "NXP S32K144" is selected in Project Explorer.
Workaround: Do not select "NXP S32K144" in Project Explorer, it is the IAR plug-in issue.
- **Uninstallation of P&E drivers:** The P&E Device Drivers are not uninstalled with uninstallation of S32 Design Studio. After that, an attempt to uninstall them manually can cause errors.

Workaround: If you do not need the P&E drivers for other products and devices, uninstall them before the product uninstallation.

- **Hyperlinks** may not work correctly when Microsoft Edge is configured as the default browser.

Workaround: Set a different browser as the default one.

- **Views** are not updated instantly after executing a command in the GDB console.

Workaround: Use the `step` command to see refreshed views.

- The **Getting Started** page may be displayed blank in Linux.

Workaround: Install Webkit1 for GTK2 using the following command: `sudo apt-get install libwebkitgtk-1.0-0`

- The **Getting Started** page may fail to load content at startup of S32 Design Studio for ARM 2.2. “The page can't be displayed” error message may be shown instead.

Workaround: Refresh the **Getting Started** page by pressing the F5 key.

- **Debugging an elf file generated by the GHS compiler** may not work correctly: missing call stack information, incorrect info on step into function, etc.

Workaround: In case of GHS generates the GDB incompatible debug information, it prevents the debugger displaying the call stack. Sometimes gdb can still debug your program without this information. The post-build step should be executed: open the project properties, click **C/C++ Build > Settings** and select the **Build Steps** tab, then add the following command: `gstrip -r=.debug_frame`. Switch to Instruction Stepping Mode to be able to step through function. If this workaround does not help, please report issue to the GreenHills support.

- **Creating new project in the root of drive C:\ fails:** When creating a new project in the root directory, the project wizard shows the "Can't create directory: C:\project_name" error message. Windows does not allow user to create files in this location.

Workaround: Create a subdirectory in the root directory (for example, C:\projects) and specify this location in the project creation wizard.

- **Linker error** may occur on low-RAM devices (for example, SKEAZN16) when a project is created in the project creation wizard with the C++ and EWL options but later the Library option is changed to "newlib"/"newlib_nano".

Workaround: Open the linker file and delete or comment the following symbol definition `EXTERN(_ZN10__cxxabiv119__terminate_handlerE)`

- Compatibility of the GDB based debuggers and the Green Hills compiler cannot be guaranteed. It is recommended to use Lauterbach TRACE32 Debugger for project with the Green Hills compiler.
- When two **IAR Workbenches** (7.x and 8.x) are installed and user switches to new workspace, paths variables for them are lost and there is no sync in Eclipse to provide double variable dialog showing at the same time. It results in the **User Operation is Waiting** dialog displaying forever when importing projects with **IAR toolchain**.

Workaround:

- # Set at least one IAR path variable through the **Preferences** window before starting to work with IAR-depended projects (recommended).
 - # Stop root **User waiting operation** monitor through the progress dialog (not recommended).
- Project importing or exporting with **ProjectInfo.xml** mechanics is designed to be used only for C projects - there is no ability to find out if options are provided for C or C++ linker (entire design restriction). All the **KDS** projects have C++ building nature and linker parameters, so, this mechanics cannot be used for them neither export nor import.

Note: There are some issues which are introduced by Eclipse CDT and are therefore reproduced in the S32 Design Studio. These issues might be fixed when the fix is available in the future version of Eclipse CDT and the S32 Design Studio migrates to the updated CDT.

Appendix B. Performance Considerations

The following suggestions will help you keep the S32 Design Studio tools running at a respectable performance level.

1. To maximize the performance, the S32 Design Studio tools should be installed on a computer with the recommended configuration. While the tools will operate on a computer with the operational minimum configuration, limited hardware will restrict its ability to function at desired performance levels.
2. Close unused projects. Eclipse caches files for all open projects in the workspace. If you need multiple projects open, try to limit the number of projects to no more than 10.
3. The Eclipse IDE provides several options that provide user assistance tools. These options, however, use memory and CPU resources. If the performance is slow, and you do not need these options, turn them off.
4. Scalability mode options configure how Eclipse deals with large source files and may affect the overall performance of the IDE. Turn off these options to maximize the performance. The following scalability mode options are available: editor live parsing, semantic highlighting, syntax coloring, parsing-based content assist proposals, and content assist auto activation.

To disable:

- a. Choose **Window > Preferences**
 - b. Expand **C/C++ > Editor > Scalability**
 - c. Uncheck **Enable all scalability mode options**
5. Content Assist Auto Activation can reduce the number of keystrokes one must type to create the code. The Content Assist plug-in consists of components that predict what keystroke you may type next, based on the current context, scope and prefix. Turn off this predictor to maximize the performance.

To disable:

- a. Choose **Window > Preferences**
- b. Expand **C/C++ > Editor > Content Assist**

- c. Uncheck all the options for **Auto-Activation**
6. For faster serial communication response (for example, UART connection), you can change the OS latency timer settings. The latency timer is a form of time-out mechanism for the read buffer of FTDI devices. The default value for the latency timer is 16 ms. You can decrease the latency value to maximize the performance:
 - On Windows, go to the port properties, open advanced port settings and select the required latency value from the drop-down menu.
 - On Linux, open the terminal and set the required value using the `echo` or `setserial` commands. For example:

```
echo 2 > /sys/bus/usb-serial/devices/ttyUSB0/latency_timer
```



How to Reach Us:

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP 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 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 does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Registered trademarks: NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTest, CodeWarrior, ColdFire, ColdFire+, C-Ware, Energy Efficient Solutions logo, Kinetis, mobileGT, PowerQUICC, Processor Expert, QorIQ, Qorivva, StarCore, Symphony, and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. and Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, Layerscape, MagniV, MXC, Platform in a Package, QorIQonverge, QUICC Engine, Ready Play, SafeAssure, SafeAssure logo, SMARTMOS, Tower, TurboLink, Vybrid, and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

Arm, the Arm logo, and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. mbed is a trademark of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

IEEE nnn, nnn, and nnn are registered trademarks of the Institute of Electrical and Electronics Engineers, Inc. (IEEE). This product is not endorsed or approved by the IEEE. Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

Document Number: S32DSARM22

Rev. 1.0

01/2020

