

S32 Design Studio for Vision 2018.R1

Contents

1. Release Description.....	2
1.1. Release Content.....	2
2. What's New.....	2
2.1. New Features.....	2
2.2. Dropped Features.....	3
2.3. Bug Fixes.....	3
3. System Requirements for Windows Host.....	4
3.1. Recommended Configuration.....	4
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.....	5
4.3. Host Operating System Support.....	5
5. Product WEB page.....	5
6. Installation and Licensing.....	5
7. Technical Support.....	6
Appendix A. Known Issues and Workarounds.....	6
Appendix B. Performance Considerations.....	10

1. Release Description

NXP Semiconductors is pleased to announce the release of the S32 Design Studio for Vision 2018.R1 for Automotive and Ultra-Reliable MCUs. The 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 open-source software to offer designers a straightforward development tool with no code-size limitations.

1.1. Release Content

- Eclipse Neon 4.6 Framework
- Supported MCU: S32V234
- GNU Tools for ARM[®] Embedded Processors (launchpad) build (4.9.3 20150529)
- GNU Bare-Metal Targeted Tools for ARM[®] 32-bit Embedded Processors (GCC version 6.3.1 20170509, build 1574 revision g924fb68)
- GNU Bare-Metal Targeted Tools for ARM[®] 64-bit Embedded Processors (GCC version 6.3.1 20170509, build 1574 revision g924fb68)
- GNU Linux Targeted Tools for ARM[®] 64-bit Embedded Processors (GCC version 6.3.1 20170509, build 1574 revision g924fb68)
- Libraries NewLib, NewLib Nano, EWL, and EWL Nano
- NXP APU compiler version 4.0.1
- APU Emulator
- S32 Debug Probe support provided with S32 Debugger
- P&E Multilink/Cyclone/OpenSDA (with P&E GDB Server)
- Vision SDK v.1.2.0¹
- New project wizard to create application, library and Visual Graph Tools (VGT) projects for supported devices
- Visual Graph Tools to support ISP and APEX2 targeted software development
- EmbSys Registers view
- DDR Configuration and Validation, and DDR Stress Test tools
- Lauterbach debugger supported by the project creation wizard
- Kernel Aware debugging for FreeRTOS, OSEK

2. What's New

S32 Design Studio for Vision 2018.R1 addresses a number of defects and new features.

2.1. New Features

- APU Compiler is updated. Learn the details from the release notes located in S32DS\APUC\doc.

¹S32 Design Studio for Vision 2018.R1 is only compatible with this version of Vision SDK.

- Vision SDK v.1.2.0 integration. Learn the details from the release notes located in S32DS \s32v234_sdk.
- The GCC toolchain v.6.3 is supported
- Debugging via the S32 Debug Probe interface is supported
- The default project structure is modified
- New TEST_A53 and TEST_EMU configurations are available
- The user experience with ISP and APEX2 Visual Graph Tools is enhanced with the following features:
 - The improved wizard for importing ISP graphs from the Vision SDK
 - Emission of ACF source code from APEX2 graph projects.
 - Options to emit ACF source code partially - for APEX2 kernels, graphs, host, or test
 - The location of the emitted code defined by the user
 - Support for the ISP static sequencer
 - The ACF resolver running as part of the ACF builder
 - New APU Kernels can be added to the Kernel Library
- APEX2 project examples are updated to use new configurations. The test configuration works with the code emitted from the APEX graph; the main configurations work with the code from Vision SDK.
- Debugging on APEX2 and ISP is supported
- Environment variables in the SDK path are supported
- The updated documentation set includes the Migration guide to help migrate projects from S32DS for Vision v.2.0 to S32 Design Studio for Vision 2018.R1
- The Getting started page enables navigation through the S32 Design Studio ecosystem and provides collateral materials such as videos and How-To articles
- The latest versions of P&E debug plug-ins and drivers are provided
- The EmbSys Registers view allows import of register values from file and export of register values to file

2.2. Dropped Features

- Support for S32V234 Cortex-A53 APEX2/ISP bareboard is no longer provided by the Vision SDK

2.3. Bug Fixes

- Updated header and register description files to fix issues and synchronize with latest Ref Manual versions
- The Rename Project functionality is improved to cover referenced projects and launch configuration

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
- 8.5 GB of disk space for installation files (full product or updates)
- 3.32 GB of temporary disk space (needed in addition to 8.5 GB to extract installation files)
- 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
- 8.5 GB of disk space for installation files (full product or updates)
- 3.32 GB of temporary disk space (needed in addition to 8.5 GB to extract installation files)
- USB port for communications with target hardware

3.3. Host Operating System Support

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

S32 Design Studio for Vision 2018.R1 supports all flavors and editions of the above operating systems as limited to 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
- 7.5 GB of disk space for installation files (full product or updates)
- 3.1 GB of temporary disk space in /tmp directory (needed in addition to 7.5 GB to extract installation files)
- Java 1.8
- GCC 5.x
- USB port for communications with target hardware

- Ethernet port for communications with target hardware (optional)

Refer to Installation Manual for details on installation prerequisites for Linux platforms.

4.2. Operational Minimum Configuration

- PC with 1.8 GHz Intel® Pentium® compatible processor
- 2 GB of RAM
- 7.5 GB of disk space for installation files (full product or updates)
- 3.1 GB of temporary disk space in /tmp directory (needed in addition to 7.5 GB to extract installation files)
- 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

5. Product WEB page

S32DS Vision 2018.R1 product page is <http://www.nxp.com/S32DS-VISION>. It contains general information about this product, updates, and download links.

6. Installation and Licensing

To install S32 Design Studio for Vision 2018.R1, choose the download option that meets your needs. The installation package contains the complete S32 Design Studio for Vision 2018.R1 suite and the installer. All data needed by the installer will be downloaded and no other download will be performed².

Run the installation package for the installer wizard to guide you through the installation process.

Note: Installation of S32 Design Studio for Vision 2018.R1 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.

²The plugins to support 3rd party compilers or debuggers are not included and have to be installed from the corresponding update site or package.

New functionality, including support for new devices, can be added to S32 Design Studio for Vision 2018.R1 with service packs, updates and patches. Service packs add specific support for new devices. Updates and patches correct software defects and add general functionality affecting more than one device family.

New support can be added directly from the Internet or from a downloaded archive. If your computer is connected to the Internet, select **Install New Software** on the **Help** menu to view all updates available. If your computer is not connected to the Internet, you can download the required service pack, update or patch from the product Web page and follow the Service Pack Updater procedure posted on the site.

7. Technical Support

S32 Design Studio general issues are tracked through the S32DS Public NXP Community space:

<https://community.nxp.com/community/s32/s32ds>

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>

Appendix A. Known Issues and Workarounds

- **Conditional watchpoints and breakpoints:** Conditional breakpoints and watchpoints, including those using ignore counts, do not work always.
Workaround: Do not use conditions for breakpoints and watchpoints, instead check for condition in the code and set a normal breakpoint.
- **Uninstallation of P&E drivers:** The P&E Device Drivers item will remain in the Control Panel/ Programs and Features folder after uninstallation of S32 Design Studio, if user will try to use this item to uninstall – error message appeared, the user should ignore it. But the drivers will remain in the system.
Workaround: P&E drivers should be uninstalled before the product uninstallation. But one should be careful if several products are using P&E drivers.
- **Disassembly view** content might be destroyed occasionally.
Workaround: Close disassembly view and open it again using menu **Window > Show View > Disassembly**.
- **Views** are not updated instantly after executing a command in the GDB console.
Workaround: Use the `step` command to see refreshed views.
- **USB driver conflict** between P&E and Xilinx software: P&E Drivers for Windows utilize 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. This is because 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: Please 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 drivers issue persists, please submit a support request to P&E Microcomputer Systems:

<http://www.pemicro.com/support/index.cfm>

- **Missing license** could be reported when you start IDE.

Workaround: Same activation ID should be entered.

- **Hyperlinks** might not work correctly on Windows 10 when Microsoft Edge is configured as the default browser.

Workaround: Set another browser as the default.

- **Welcome page** page may be displayed blank on Linux.

Workaround: Install Webkit1 for GTK2 using the following command:

```
sudo apt-get install libwebkitgtk-1.0-0
```

- **Build failure:** Project builds may fail on a non-English version of Windows. The error occurs if the path to the linker file includes non-ASCII characters. The allowed characters are: A-z, 0-9, '-' (dash), '_' (underscore), '/' (slash), '.' (period).

Workaround: Open the project properties and go to the **C/C++ Build > Settings > Tool Settings > Standard S32DS C/C++ Linker > General** page. In the **Script files (-T)** field, ensure that the path to the linker file uses allowed characters and all included variables are resolved to ASCII strings. If necessary, relocate the linker file or rename the folders. If the issue is caused by a variable, use the relative path. For example, the **Script files (-T)** field shows the following default path to the linker file:

```
${ProjDirPath}/Project_Settings/Linker_Files/<project_name>.ld
```

The `${ProjDirPath}` variable holds the project folder path that may contain characters out the allowed range. If this be the case, replace the default path with the relative path that excludes the use of the `${ProjDirPath}` variable:

```
../Project_Settings/Linker_Files/<project_name>.ld
```

The update takes effect after you click **Apply**.

- **Build failure:** Building of a project may fail with unresolved symbol claims because the header files with symbol definitions were skipped from indexing due to the file size restrictions.

Workaround: Open the preferences (**Window > Preferences**) and go to **C/C++ > Indexer**. Make sure the C/C++ indexer is enabled. Modify the **Skip files larger than** and **Skip included files larger than** settings so that no files in your project are skipped from indexing. Under **Cache limits**, raise the value of the **Absolute Limit** setting for the header file cache to 256 MB. Click **Apply** and rebuild the project.

- **Debugging on ISP with S32 Debugger may fail:** When debugging an ISP application example, the execution of the program may be terminated unexpectedly after the S32 Debugger ISP configuration has been loaded.

An attempt to debug the program on the ISP coprocessor may fail after the first breakpoint is encountered. The failure occurs because the breakpoint freezes the application's ISP frame, while the stream from the camera continues. After resume, the ISP frame cannot be synchronized with the stream and causes the program failure.

Workaround: No workaround available.

- **Remote Linux debug session terminated while debugging on ISP:** The overall structure of the application with ISP support assumes complex hardware relationship between the Cortex-A53 core, sequencer and ISP coprocessor. The termination occurs because of synchronization problem while timeout checking.

Workaround: To keep the debug session alive, use the ISP application with the static sequencer support and set the Heart Beat Source to MIPI.

- **The source file is not found during debugging on APEX2:** When debugging an APEX2 application with S32 Debugger on Linux, the debug session shows the "Can't find a source file..." error message. The error occurs because the APEX2 libraries include the Windows-style slashes.

Workaround: Click the **Locate File...** button to locate the kernel files manually, or update the source lookup paths on the **Source** tab of the debug configuration settings.

- **The EmbSys Registers view** does not support some core specific registers such as NVIC, SysTick, and other.

Workaround: Access the core specific registers in the **Memory** view.

- **License check failure when building on Linux:** When building a project on Linux with the parallel build option enabled, the build may fail with the error message “ License check failed. Feature not found (APU-COMPILER_v1.1, APUCompiler, 1.1) ” .

Workaround: Disable the parallel build option and rebuild the project.

- **Unexpected closure of S32 Design Studio for Vision 2018.R1:** An attempt to open the **Product Licenses** window right after the product launch may terminate the product.

Workaround: Restart the application and wait for a few seconds before opening the **Product Licenses** window.

- **An ISP hardware breakpoint appears in the wrong debug context:** When debugging an ISP application example remotely using the S32 Debugger ISP configuration, a hardware breakpoint set on an ISP address appears in the context of the remote Linux application rather than in the ISP context.

Workaround: No workaround available.

- **Upload of an executable file may fail after reset of the board:** When starting a remote debug session for an ISP Linux application example, the first attempt to upload the executable to the target fails after reset of the board.

Workaround: Upload the file again, or restart S32 Design Studio for Vision 2018.R1.

- **The EmbSys Registers view cannot read some registers:** Some peripheral registers are displayed as “ not read ” when clicked to be updated. This happens because the **EmbSys Registers** view tries to read the reserved memory areas when reading some peripheral registers. The registers cannot be updated and are displayed as “ not read ” .

Workaround: Access the “ not read ” register values in the **Memory** view.

- **USB connection failure when debugging with S32 Debug Probe on a Linux VM:** A debug session fails on a Linux virtual machine with S32 Debug Probe connected through USB. Debugging with S32 Debug Probe connected through Ethernet can be done successfully.

Workaround: Connect the probe to USB. Set up the debug configuration to use the Ethernet connection rather than USB. Specify the virtual IP address of the probe in the connection settings of the debug configuration. To obtain the required value, run the following command:

```
sudo ifconfig -a
```

The displayed output includes a section for the virtual Ethernet interface created on Linux for communication with the probe over the physical USB link. The *inet addr* value in this section is a local IP address starting with “ 169.254 ” . Take the *HWAddr* value in the section and convert the last two bytes into decimal notation. Append the resulting numbers to “ 169.254. ” to obtain the required virtual IP address.

- **Wireless Internet connection failure:** When debugging with S32 Debug Probe connected through USB on a machine connected to a wireless network, the Wi-Fi connection fails after a minute and the user can no longer have access to the Internet. The behavior is related to the Windows policy that prevents simultaneous connectivity between domain and non-domain networks over Ethernet and Wi-Fi at the same time.

Workaround: Ask the network administrator to allow simultaneous domain and non-domain connections for the user account (recommended). Or, use a wired Ethernet connection to the domain network, if available. Or, connect to an external Wi-Fi network, then connect to the domain through a VPN client.

- **The Breakpoints view allows watchpoints above the limit:** The **Breakpoints** view and the **Debugger Console** view (GDB Client) may show a different number of watchpoints. This happens because the GDB Client restricts the number of watchpoints by four while the **Breakpoints** view does not. For watchpoints exceeding the limit of four the GDB console shows the error message: “ There are not enough available hardware resources for this watchpoint ” .

Workaround: Avoid setting more than four watchpoints in the code.

- **Debug session errors may not appear in the console:** When starting a debug session, errors in the connection parameters, in the initialization script, and some others may not appear in the debug session output (for instance, in the semihosting console when debugging with S32 Debugger). These errors are directed to the GDB traces console which is disabled by default.

Workaround: Before launching a debug session, make sure that the GDB traces console is enabled. To enable the GDB traces console, go to **Window > Preferences > C/C++ > Debug > GDB** and select the option **Show the GDB traces consoles with character limit**.

- **The P&E hardware is not detected on CentOS:** The PEMicro Interface Settings display empty list of ports on the Debugger tab. The port over which debug communication is conducted cannot be selected, the P&E debug connection is not set.

Workaround: Install the latest PEMicro update: click **Help** > **Install New Software**, select the PEMicro site from the **Work with** drop-down list, and follow the installer instructions.

- The **Debugging Console** view may display in red font the information that is not critical for debugging, such as warning messages, parameters, and other.

Workaround: No workaround available.

- **Watchpoints set on complex data types are not hit:** When debugging on a target connected with S32 Debug Probe, a debug session ignores a watchpoint set on a variable of a complex data type (such as a structure or other). A watchpoint set on an item of a basic data type inside a complex variable works correctly.

Workaround: Avoid setting watchpoints at complex data types.

- **The debugged source file is not detected:** When a debug session on ISP is started, at first stop the console displays the “No source available 0x...” message rather than the name of the source file being debugged.

Workaround: Wait and then step through the code. The source files and the views get in sync automatically.

- **Meaningless text displayed in the Disassembly view:** The **Disassembly** view may display arbitrary characters instead of some addresses.

Workaround: No workaround available. The expected behavior is restored by itself when stepping through the code.

- **The expanded groups of registers are collapsed unexpectedly:** When stepping rapidly through the assembly code, the expanded groups of registers in the **Registers** view get collapsed after each step.

Workaround: No workaround available.

Note: There are some issues which are introduced by CDT and therefore reproduced in S32 Design Studio. These issues might be fixed when the fix is available with a newer CDT version and when S32 Design Studio migrates to it.

Appendix B. Performance Considerations

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

1. To maximize performance, the S32 Design Studio tools should be installed on a computer with the recommended system configuration. While the tools will operate on a computer with the minimum configuration, the 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 bandwidth. If performance is slow and you do not need these options, turn them off.
4. These scalability options configure how Eclipse deals with large source files:

- Editor live parsing: impacts parsing while typing, Outline View, semantic highlighting, folding, etc.
- Semantic highlighting: C/C++ identifiers are colored
- Syntax coloring: coloring of keywords, comments and literals
- Parsing-based content assist proposals: content assist proposals which require file parsing
- Content assist auto activation: content assist activated automatically on trigger sequences such as '!', '::' and '->'

To disable the scalability options:

- a. Click menu **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 a developer must type to create code. The Content Assist plug-in consists of components that predict what a developer will type, based on the current context, scope and prefix.

To disable Content Assist Auto Activation:

- a. Click menu **Window > Preferences**.
- b. Expand **C/C++ > Editor > Content Assist**.
- c. Uncheck all the options for **Auto-Activation**.



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, QorIQ Onverge, 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.

© 2018 NXP B.V.

Document Number: S32DSVISION2018.R1

Rev. 1.0

10/2018

