

S32 Design Studio for ARM v2.0

1. Release description

Freescale, a subsidiary of NXP Semiconductors, is pleased to announce the release of the S32 Design Studio for ARM v2.0 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
- GNU Tools for ARM® Embedded Processors (Launchpad) build (4.9.3 20150529)
- ARM64: Linaro GCC 4.9-2015.05
- Libraries included: newlib, newlib-nano and ewl2 (ewl and ewlnano)
- P&E Multilink/Cyclone/OpenSDA (with P&E GDB Server)
- SEGGER J-Link (with SEGGER GDB Server)
- New Project wizard to create application and library projects for supported devices
- Fully integrated S32 SDK for S32K14x EAR release v.0.8.4. For the details on the feature set

Contents

1. Release description	1
1.1. Release content	1
2. What's New	2
2.1. New/Updated device support	2
2.2. New Features	2
2.3. Bug Fixes	3
3. System Requirements for Windows Host	3
3.1. Recommended Configuration	3
3.2. Operational Minimum Configuration	3
3.3. Host Operating System Support.....	3
4. System Requirements for Linux Host	4
4.1. Recommended Configuration	4
4.2. Operational Minimum Configuration	4
4.3. Host Operating System Support.....	4
5. Product WEB page.....	4
6. Installation and Licensing	4
7. Technical Support	5
Appendix A. Known issues and Workarounds.....	6
Appendix B. Performance Considerations	7



of SDK please refer to SDK Release notes and Reference Manuals¹

- SDK management included:
 - Sample Drivers for KEA family (Evaluation grade)
 - FreeMaster Serial Communication driver for KEA and S32K families
 - Automotive Math and Motor Control Libraries for KEA and S32K devices
- Import projects from CodeWarrior for MCU v.10.6 and Kinetis Design Studio for respective supported processors
- IAR v7.x compiler support by new project wizard²
- iSystem, Lauterbach and IAR debuggers support by new project wizard
- Kernel Aware debugging for FreeRTOS, OSEK.
- Devices supported:
 - SKEAZN8, SKEAZN16, SKEAZN32, SKEAZN64, SKEAZ128, SKEAZ64
 - S32K144 v2.0, S32K142, S32K148
 - S32V234
 - MAC57D54H

2. What's New

S32 Design Studio for ARM v2.0 addresses a number of defects and new features:

2.1. New/Updated device support

- S32K142, S32K148

2.2. New Features

- Moved to Eclipse Neon 4.6 Framework
- Updated SEGGER and P&E debug plugins to the latest and greatest versions
- S32 SDK for S32K14x EAR release v.0.8.4 integrated
- AMMCLIB v.1.1.8 integrated
- Restyled S32 Design Studio examples support

¹ Please note that S32K SDK has Early Access Release status, that means that there could be some limitations and issues. The S32K SDK is available for Windows host only.

² The plugins to support IAR, iSystem, Lauterbach are not included and have to be installed from corresponding update site or installation. The current version of GreenHills plugin is not compatible with Eclipse Neon version, so could not be used with this release.

2.3. Bug Fixes

- C preprocessor tool added for C++ projects configuration
- The makefile generation is update to fix issue with building a project on Korean version of Windows - Code page switch command cannot be executed.
- 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
- 4 GB of disk space (when installing full product or updates)
- 3 GB on Windows system disk
- 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
- 3 GB of disk space (when installing full product or updates)
- 2 GB on Windows system disk
- 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 ARM v2.0 supports all flavors and editions of the above operating systems as limited to the requirements of the Java JRE.

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 (when installing full product or updates)
- 3 GB in /tmp directory
- Java 1.8
- USB port for communications with target hardware
- Ethernet port for communications with target hardware (optional)

4.2. Operational Minimum Configuration

- PC with 1.8 GHz Intel® Pentium® compatible processor
- 2 GB of RAM
- 3 GB of disk space (when installing full product or updates)
- 2 GB in /tmp directory
- Java 1.8
- USB port for communications with target hardware

4.3. Host Operating System Support

- Ubuntu 14.04, 16.04 64-bit
- Debian 8 64-bit
- CentOS 7 64-bit

5. Product WEB page

S32 Design Studio for ARM v2.0 product page is <http://www.nxp.com/S32DS>. It contains general information about this product, updates, and download links.

6. Installation and Licensing

To install S32 Design Studio for ARM v2.0, choose the download option that meets your needs.

The installer package contains the complete S32 Design Studio for ARM v2.0 suite and an installer. All data needed by the installer will be downloaded and no other download will be performed³. Run the installation package and a wizard will guide you through the installation process.

During installation the license activation will be requested. The following types of activation supported:

- Online activation, requires the internet access and request for license will be sent automatically
- Offline activation, no internet access required, the offline activation request is generated and should be passed to licensing site to get the activation response, which should be loaded back to installer.

New functionality including support for new devices can be added to S32 Design Studio for ARM v2.0 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 in the Help Menu and all available updates will be displayed. If your computer does not have Internet access, you can download the archive that contains the service pack, update or patch you need from product page and follow the Service Pack Updater procedure posted on the site.

7. Technical Support

S32 Design Studio 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>

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

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 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.
- 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 on IDE start.

Workaround: same activation ID should be entered.
- Problems could be experienced while **debugging of elf file generated by GHS compiler** like missing call stack information, incorrect info on step into function, etc.

Workaround: On occasion GHS generates GDB incompatible debug information preventing the debugger from displaying the call stack. In some cases, without such information gdb can still debug your program. The program build needs to be augmented with a post-build step, select the project in the project explorer view and File->Properties to display the project properties, under C/C++ Build -> Settings select the build steps tab and enter the command gstrip – r=.debug_frame executable. Switch to Instruction Stepping Mode to be able to step through function. If this workaround does not help – please report issue to GreenHills support.
- Hyperlinks** might not work correctly under Windows 10 when Microsoft Edge is default browser

Workaround: Set another browser as default.

NOTE

There are some issues which are introduced by CDT therefore reproduced in S32 Design Studio. Those issues might be fixed when the fix is available with 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.
- Scalability options configure how eclipse deals with large source files.
 - Scalability options:
 - 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 parsing the file
 - Content assist auto activation: content assist activated automatically on trigger sequences, like '!', '::' or '->'.
 - To disable
 - Click menu 'Window' -> 'Preference'
 - Expand 'C/C++' -> 'Editor' -> 'Scalability'
 - Uncheck 'Enable all scalability mode options'
 - 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:
 - Click menu 'Window' -> 'Preference'

- Expand 'C/C++' -> 'Editor' -> 'Content Assist'
- 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, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. 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.

© 2017 NXP B.V.

Document Number: S32DSARM2.0
Rev. 2.0
08/2017

