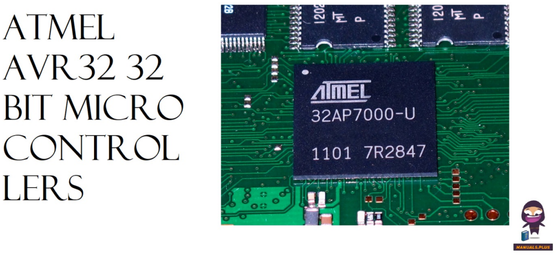
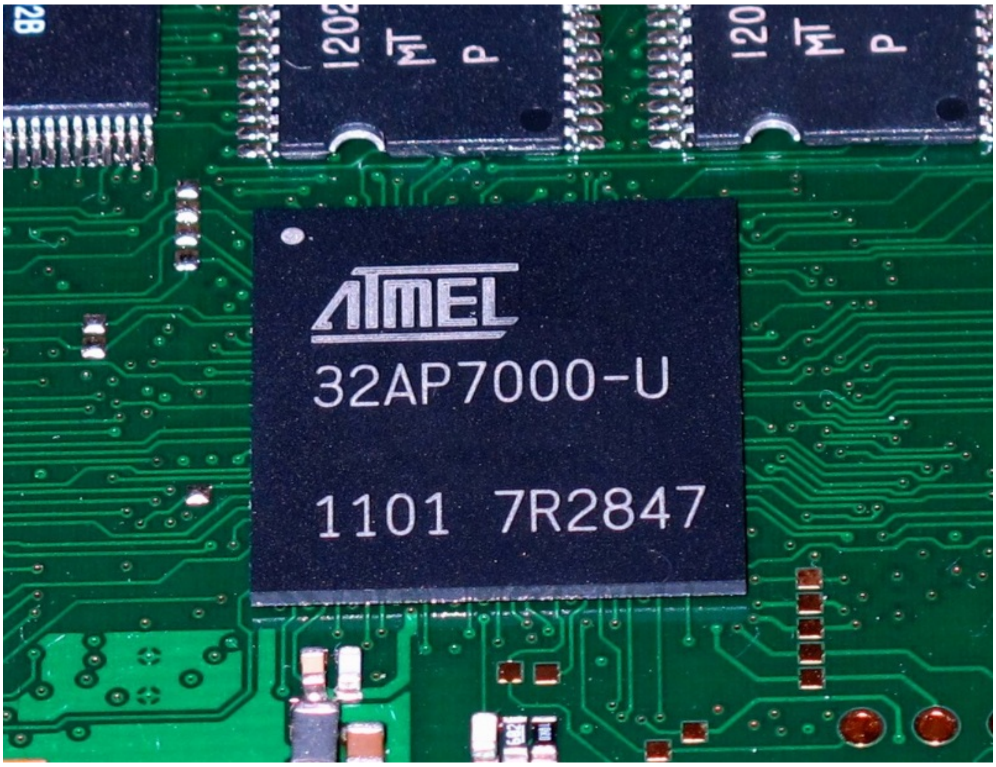


ATMEL AVR32 32 Bit Micro Controllers Instruction Manual

Contents

ATMEL AVR32 32 Bit Micro Controllers



Specifications

- Product Name: AVR32 Studio
- Version: Release 2.6.0
- Supported Processors: Atmel's AVR 32-bit processors
- Supported Microcontrollers: 8/32-bit Microcontrollers
- Tool Support: AVR ONE!, JTAGICE mkII, STK600
- Toolchain Integration: AVR/GNU Toolchain

Product Usage Instructions

Installation

AVR32 Studio is an integrated development environment for writing, debugging, and deploying 32-bit AVR applications. It is distributed by Atmel for free and runs on both Windows and Linux.

System Requirements

- **Hardware requirements:** AVR32 Studio has not been tested on low-resource computers but may run depending on project size.
- **Software requirements:** Not supported on Windows 98, NT, or ME.

Downloading and Installing

- **Installing from product package:** The complete product builds can be found on the AVR Technical Library DVD or downloaded from Atmel's website. Choose Custom Installation for specifying installation location.
- **Installing on Windows:** Download the AVR32 Studio installer from Atmel's website and run it. A Sun Java Runtime Environment will be installed if missing.

AVR32 Studio: Release 2.6.0

AVR32 Studio is an integrated development environment (IDE) for developing 32-bit AVR applications. AVR32 Studio provides a complete set of features including project file management, task management and version control integration (CVS); a C/C++ editor with syntax highlighting, navigation and code completion; a debugger supporting run control including source and instruction-level stepping and breakpoints; registers, memory and I/O views; and target configuration and management. AVR32 Studio is *Built on* Eclipse, enabling easy integration with third party plugins for increased functionality.

AVR32 Studio supports all of Atmel's AVR 32-bit processors. AVR32 Studio supports development and debugging of both standalone (without an operating system) applications and Linux applications (for the AT32AP7 device family). Third party plug-ins exist for debugging other operating systems.

All Atmel tools that support the 32-bit AVR architecture, including the AVR ONE!, JTAGICE mkII and STK600 are supported by AVR32 Studio.

AVR32 Studio integrates with the 32-bit AVR/GNU Toolchain. The GNU C Compiler (GCC) is used for compiling C/C++ programs, while the GNU debugger (GDB) is used for debugging the application on target. Atmel's AVR Utilities, avr32program and avr32gdbproxy, are used for the deployment and debugging of standalone applications as well as target voltage and clock generator adjustments.

Installation Instructions

AVR32 Studio is an integrated development environment for writing, debugging and deploying 32-bit AVR applications. AVR32 Studio is distributed by Atmel free of charge, and runs on both Windows and Linux.

News

This version of AVR32 Studio is an upgrade from release 2.5. The various components AVR32 Studio is based on has been upgraded to the Eclipse Galileo service release 2. This means that a large number of bug fixes, enhancements and other improvements has been included into this release.

- C/C++ Development tooling (108 issues fixed)
- Issue tracker integration, Mylyn (166 issues fixed)
- Eclipse platform (149 issues fixed)
- Target Management/Remote System Explorer (5 issues fixed)

In addition 77 AVR32 Studio bug fixes and enhancements has been implemented. See the *New and Noteworthy*

section for details on the most important changes.

System Requirements

AVR32 Studio is supported under the following configurations.

Hardware requirements

- Minimum processor Pentium 4, 1GHz
- Minimum 512 MB RAM
- Minimum 500 MB free disk space

- Minimum screen resolution of 1024×768

AVR32 Studio has not been tested on computers with less resources, but may run satisfactorily depending on the number and size of projects and the user’s patience.

Software requirements

- Windows 2000, Windows XP, Windows Vista or Windows 7 (x86 or x86-64). Note that since Windows 2000 does not have an “advanced graphics context” certain graphical elements will not be rendered in the desired
- Fedora 13 or 12 (x86 or x86-64), RedHat Enterprise Linux 4 or 5, Ubuntu Linux 10.04 or 8.04 (x86 or x86-64), or SUSE Linux 2 or 11.1 (x86 or x86-64). AVR32 Studio may very well work on other distributions. However those would be untested and unsupported.
- Sun Java 2 Platform version 1.6 or later
- Internet Explorer, Mozilla, or Firefox
- AVR Utilities version 3.0 or later **(See “Downloading and Installing”)**
- AVR Toolchains version 3.0 or later **(See “Downloading and Installing”)**

AVR32 Studio is not supported on Windows 98, NT or ME.

Downloading and Installing

AVR32 Studio requires the “AVR Toolchains” package which contains the C/C++ compilers and linkers. In addition, the “AVR Utilities” is required for programming and debugging. As of this release of AVR32 Studio both these packages are included in the product for certain configurations. There should be no need to install these separately.

However, if you do require a separate installation; the latest versions can be found at the same location as AVR32 Studio. Please install the toolchains and utilities according to the installation instructions given in the accompanying release notes.

As AVR32 Studio is started it will test for the presence of toolchains and utilities packages. If these are not found a warning is issued.

AVR32 Studio can be installed in three ways. Either as a full application, or as a feature set added to pre-existing Eclipse based software using the Eclipse Marketplace Client or the repository directly. The latter method will also allow you to select which features to install.

Installing using Eclipse Marketplace

Note that the Eclipse Marketplace Client is only available in Eclipse 3.6 and newer.

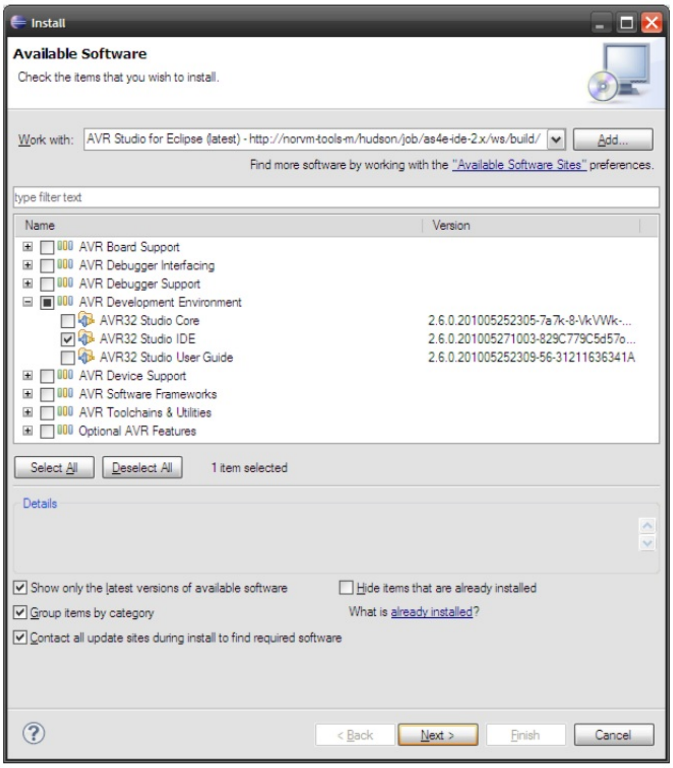
Start your Eclipse based product and open **Help > Eclipse Marketplace…..** Go to the **Search** page and search for

“AVR”. This should list “AVR32 Studio”. Press the entry’s **Install** button. The rest of the process is the same as for installing from a repository.

Installing from repository

When installing from the distribution repository you must already have software based on Eclipse ready. This should contain the Eclipse CDT (C/C++ Development Tooling) components. A good choice would be the “Eclipse IDE for C/C++ Developers” available from **http://www.eclipse.org/downloads**. If required components are not already installed they will be downloaded and installed automatically if possible.

From the main menu; open **Help > Install New Software…** to get the install wizard and add the repository at **http:// distribute.atmel.no/tools/avr32studio/releases/latest/** to the installation sources. If you have the repository as a zip- file you may use that instead.



Install

Available Software

Check the items that you wish to install.

Work with: AVR Studio for Eclipse (latest) - http://norvm-tools-m/hudson/job/as4e-ide-2.x/ws/build/ [Add...]

Find more software by working with the "Available Software Sites" preferences.

type filter text

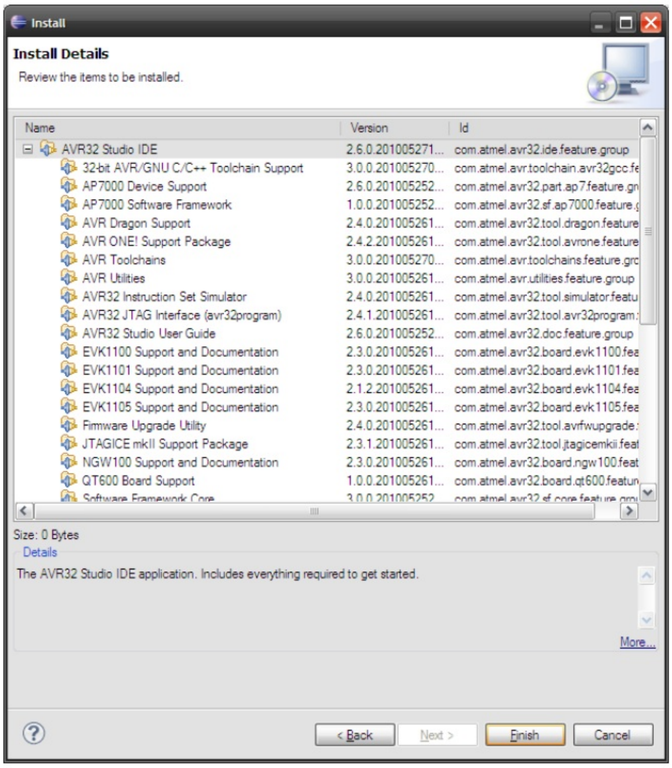
Name	Version
⊞ □ AVR Board Support	
⊞ □ AVR Debugger Interfacing	
⊞ □ AVR Debugger Support	
⊟ ■ AVR Development Environment	
□ AVR32 Studio Core	2.6.0.201005252305-7a7k-8-VkVWk-...
☑ AVR32 Studio IDE	2.6.0.201005271003-829C779C5d57o...
□ AVR32 Studio User Guide	2.6.0.201005252309-56-31211636341A
⊞ □ AVR Device Support	
⊞ □ AVR Software Frameworks	
⊞ □ AVR Toolchains & Utilities	
⊞ □ Optional AVR Features	

[Select All] [Deselect All] 1 item selected

Details

☑ Show only the latest versions of available software
☐ Hide items that are already installed
☑ Group items by category
What is already installed?
☑ Contact all update sites during install to find required software

[< Back] [Next >] [Finish] [Cancel]



Install

Install Details

Review the items to be installed.

Name	Version	Id
⊟ AVR32 Studio IDE	2.6.0.201005271...	com.atmel.avr32.ide.feature.group
32-bit AVR/GNU C/C++ Toolchain Support	3.0.0.201005270...	com.atmel.avr.toolchain.avr32gcc.fe
AP7000 Device Support	2.6.0.201005252...	com.atmel.avr32.part.ap7.feature.gr
AP7000 Software Framework	1.0.0.201005252...	com.atmel.avr32.sf.ap7000.feature.(
AVR Dragon Support	2.4.0.201005261...	com.atmel.avr32.tool.dragon.feature
AVR ONE! Support Package	2.4.2.201005261...	com.atmel.avr32.tool.avrone.feature
AVR Toolchains	3.0.0.201005270...	com.atmel.avr.toolchains.feature.grc
AVR Utilities	3.0.0.201005261...	com.atmel.avr.utilities.feature.group
AVR32 Instruction Set Simulator	2.4.0.201005261...	com.atmel.avr32.tool.simulator.featu
AVR32 JTAG Interface (avr32program)	2.4.1.201005261...	com.atmel.avr32.tool.avr32program.:
AVR32 Studio User Guide	2.6.0.201005252...	com.atmel.avr32.doc.feature.group
EVK1100 Support and Documentation	2.3.0.201005261...	com.atmel.avr32.board.evk1100.fea
EVK1101 Support and Documentation	2.3.0.201005261...	com.atmel.avr32.board.evk1101.fea
EVK1104 Support and Documentation	2.1.2.201005261...	com.atmel.avr32.board.evk1104.fea
EVK1105 Support and Documentation	2.3.0.201005261...	com.atmel.avr32.board.evk1105.fea
Firmware Upgrade Utility	2.4.0.201005261...	com.atmel.avr32.tool.avrfwupgrade.:
JTAGICE mkII Support Package	2.3.1.201005261...	com.atmel.avr32.tool.jtagicemkii.feat
NGW100 Support and Documentation	2.3.0.201005261...	com.atmel.avr32.board.ngw100.feat
QT600 Board Support	1.0.0.201005261...	com.atmel.avr32.board.qt600.featur
Software Framework Core	3.0.0.201005252	com.atmel.avr32.sf.core.feature.grou

Size: 0 Bytes

Details

The AVR32 Studio IDE application. Includes everything required to get started.

More...

[< Back] [Next >] [Finish] [Cancel]

Now select the main IDE feature as shown in the illustration. This is named **AVR32 Studio IDE**. Due to dependencies mechanisms this will automatically select all required features and even download for instance C/C+ + tooling from Eclipse.org. Any of the optional features such as support for obsoleted engineering samples can be installed now or you can add these later.

Although it’s not officially supported you can also install AVR32 Studio from the repository on OS X. However you will also need AVR Toolchain and AVR Utilities for OS X in order to make full use of the IDE. Builds for this platform is currently not available.

You should start by checking off all features except the optional ones which may or may not be interesting as this category contains obsolete or engineering sample support.

Installing from product package

The complete product builds of AVR32 Studio contains everything you need. The software can be found on the AVR Technical Library DVD, or downloaded from Atmel’s website at ***http://www.atmel.com/products/avr32/*** under the “Tools & Software” menu. These builds comes in four different configurations.

- Installer for 32-bit and 64-bit
- Zip-file for 32-bit and 64-bit
- Zip-file for 32-bit
- Zip-file for 64-bit Linux

Installing on Windows

The AVR32 Studio installer can be downloaded from the website as noted above. After downloading, double- click the installer executable file to install. If you wish to specify the location where the AVR32 Studio software is installed, choose “Custom Installation”. The installation software will install a Sun Java Runtime Environment on your computer if it is missing.

There is also a zip-file distribution available for Windows. Simply download and uncompress the file. AVR32 Studio can be launched using the executable found at the root of the new folder.

Note that if you’re running a 64-bit version of the operating system you will have to install a 32-bit version of the Java Runtime.

If device drivers for the debuggers and emulators are not found you will be notified as soon as the IDE starts. It is also possible to install these drivers from the menu. Choose **Help > Install AVR USB Drivers**.

Adding Utilities and Toolchains to the PATH

The Windows distribution of AVR32 Studio comes with AVR Utilities and AVR Toolchains as plug-ins. As these are unpacked when installed it is possible to add the binaries within to the system PATH. Hence making it possible to use these even outside of AVR32 Studio. Depending on where you installed the IDE the paths to the binaries are:

- C:\Program Files\Atmel\AVR Tools\AVR32 Studio\plugins\com.atmel.avr.toolchains.win32.x86_3.0.0.<*build-id*>\os\win32\x86\bin
- C:\Program Files\Atmel\AVR Tools\AVR32 Studio\plugins\com.atmel.avr.utilities.win32.x86_3.0.0.<*build-id*>\os\win32\x86\bin

Installing on Linux

On Linux, AVR32 Studio is only available as a ZIP archive which can be extracted using the unzip utility. Simply extract to the location where you want the application to run from.

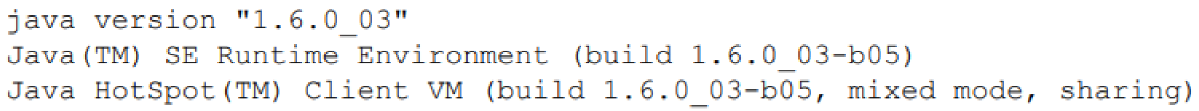
Note that if you will develop Linux applications for the AT32AP7000 you must also install the AVR32 Buildroot.

If device drivers for the debuggers and emulators are not found you will be notified as soon as the IDE starts. It is also possible to install these drivers from the menu. Choose **Help > Install AVR USB Drivers**.

IMPORTANT: The Java runtime environments shipped with many Linux distributions are not compatible with AVR32 Studio. A Java Runtime (or JDK) **1.6** is required. Consult your Linux distribution’s documentation for instructions on installing Sun Java, or download it from Sun’s website at http://java.sun.com/. Specifically, any reference to a Java version 1.7 indicates that an incompatible version is being used.

We recommend installing AVR32 Studio into a directory that is writable for the user(s). This simplifies the process of adding or updating the product. On a single-user machine, you can typically extract the AVR32 Studio ZIP file into your home directory. This creates a directory containing the product files.

To run AVR32 Studio, execute the avr32studio program from the avr32studio directory. If you experience problems, make sure the correct java is being used by running java -version which should give output similarly to this:



```
java version "1.6.0_03"
Java(TM) SE Runtime Environment (build 1.6.0_03-b05)
Java HotSpot(TM) Client VM (build 1.6.0_03-b05, mixed mode, sharing)
```

Sun Java on Ubuntu

You can install Sun’s Java on Ubuntu using the following commands from a shell:

```
sudo apt-get install sun-java6-jre
```

RedHat Enterprise Linux 4

Note that you may need to set the environment variable <code>MOZILLA_FIVE_HOME</code> to the folder containing your Firefox install. e.g.

```
export MOZILLA_FIVE_HOME=/usr/lib/firefox-1.0.4
```

or, if using a tcsh:

```
setenv MOZILLA_FIVE_HOME /usr/lib/firefox-1.0.4
```

in order for the welcome page to work.

Adding Utilities and Toolchains to the PATH

The Linux distribution of AVR32 Studio comes with AVR Utilities and AVR Toolchains as plug-ins. As these are unpacked when installed it is possible to add the binaries within to the system PATH. Hence making it possible to use these even outside of AVR32 Studio. Depending on where you installed the IDE the paths to the binaries are:

- On 32-bit Linux hosts
  - /usr/local/as4e-ide/plugins/com.atmel.avr.toolchains.win32.x86_3.0.0.*<build-id>*/os/linux/x86/bin
  - /usr/local/as4e-ide/plugins/com.atmel.avr.utilities.win32.x86_3.0.0.*<build-id>*/os/linux/x86/bin
- On 64-bit Linux hosts
  - /usr/local/as4e-ide/plugins/com.atmel.avr.toolchains.win32.x86_3.0.0.*<build-id>*/os/linux/x86_64/bin
  - /usr/local/as4e-ide/plugins/com.atmel.avr.utilities.win32.x86_3.0.0.*<build-id>*/os/linux/x86_64/bin

Upgrading from previous versions

Due to changes in the provisioning mechanisms it is not possible to upgrade from versions earlier than 2.5.0 into version 2.6.0. A fresh install must be made. However you may continue to use your existing workspace.

Standalone projects created with AVR32 Studio 2.0.1 or newer do not have to be updated. Older projects should be converted to the 2.0.1 format. Linux projects created with releases older than AVR32 Studio 2.1.0 must be converted. See the user guide chapter about upgrading projects for more details.

Contact Information

For support on AVR32 Studio please contact avr32@atmel.com.

Users of AVR32 Studio are also welcome to discuss on the AVRFreaks website forum for AVR32 Software Tools.

Disclaimer and Credits

AVR32 Studio is distributed free of charge for the purpose of developing applications for Atmel AVR processors. Use for other purposes are not permitted; see the software license agreement for details. AVR32 Studio comes without any warranty.

Copyright 2006-2010 Atmel Corporation. All rights reserved. ATMEL, logo and combinations thereof, Everywhere You Are, AVR, AVR32, and others, are the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Windows, Internet Explorer and Windows Vista are either registered trademarks or trademarks

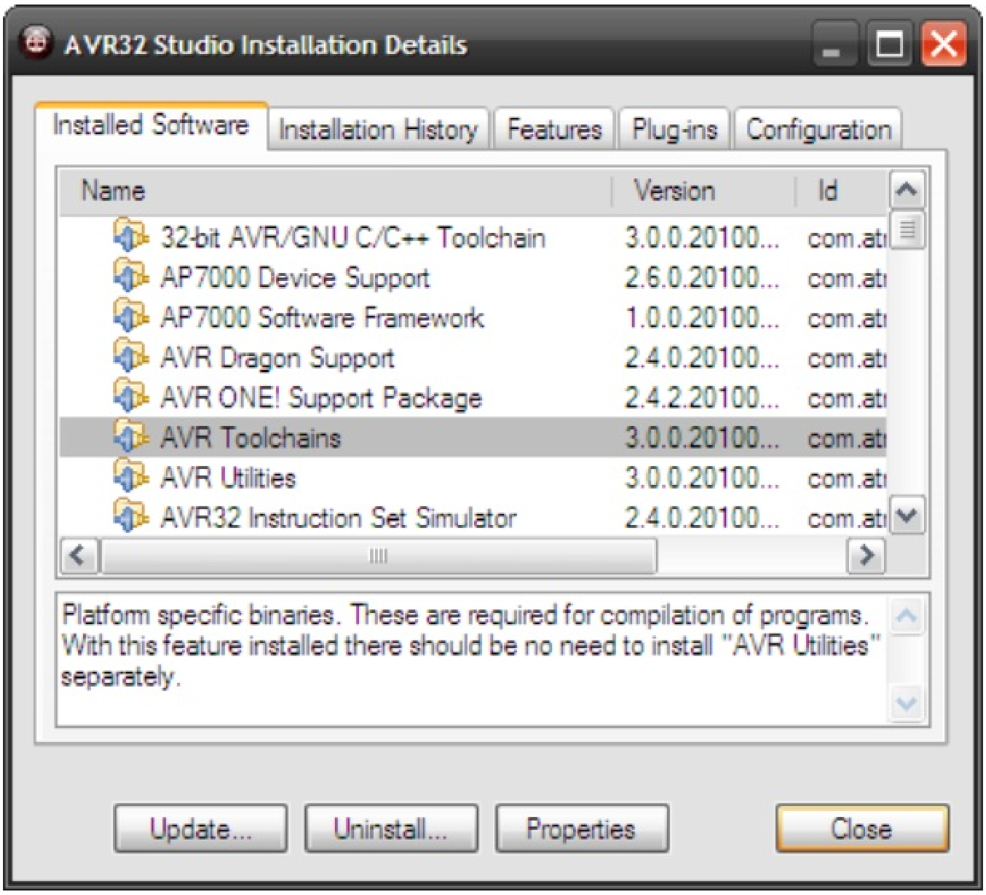
of Microsoft Corporation in the United States and/or other countries. Linux is the registered trademark of Linus Torvalds in the United States and other countries. *Built on Eclipse* is a trademark of Eclipse Foundation, Inc. Sun and Java are registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Mozilla and Firefox are registered trademarks of the Mozilla Foundation. Fedora is a trademark of Red Hat, Inc. SUSE is a trademark of Novell, Inc. Other terms and product names may be the trademarks of others.

New and Noteworthy

This chapter lists new and noteworthy items for the 2.6.0 release.

Workbench

Batteries included

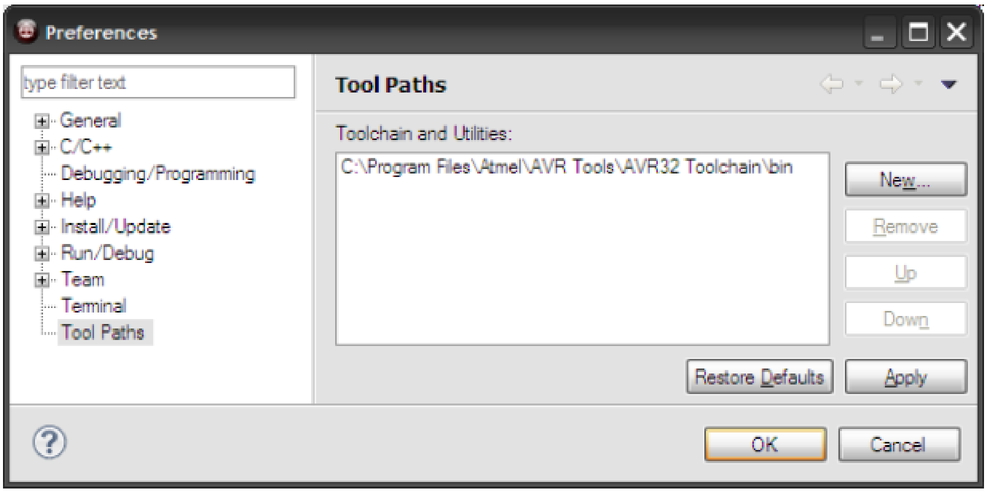


Name	Version	Id
32-bit AVR/GNU C/C++ Toolchain	3.0.0.20100...	com.at
AP7000 Device Support	2.6.0.20100...	com.at
AP7000 Software Framework	1.0.0.20100...	com.at
AVR Dragon Support	2.4.0.20100...	com.at
AVR ONE! Support Package	2.4.2.20100...	com.at
AVR Toolchains	3.0.0.20100...	com.at
AVR Utilities	3.0.0.20100...	com.at
AVR32 Instruction Set Simulator	2.4.0.20100...	com.at

Platform specific binaries. These are required for compilation of programs. With this feature installed there should be no need to install "AVR Utilities" separately.

The *AVR Toolchain* package along with *AVR Utilities* is now included into the product build for certain configurations. This means there should be no need to install these separately. All the software you need to start developing AVR applications is included. If you do install either package separately the included versions will still be present and must be removed if the external version is to be used. This can be done through **Help > About AVR32 Studio > Installation Details**.

Enhanced tool handling



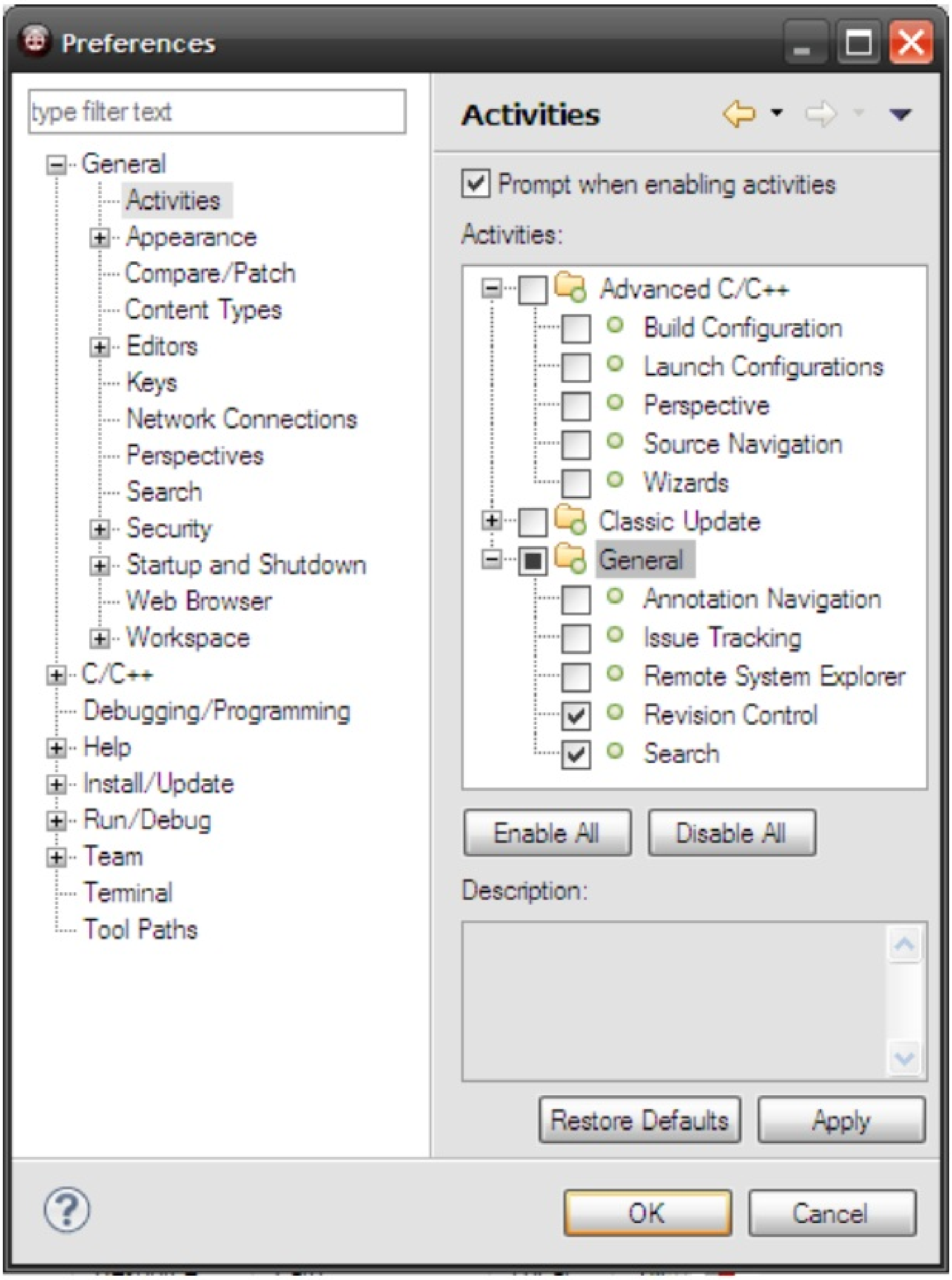
Tool Paths

Toolchain and Utilities:

C:\Program Files\Atmel\AVR Tools\AVR32 Toolchain\bin

Previously AVR32 Studio would use the system PATH or AVR32_HOME variables to figure out where the *AVR Utilities* and *AVR Toolchains* were installed. This mechanism has now been changed so that it is possible to configure which search path to use. The preference setting dialog can be found at **Window > Preferences >**

Simplified user interface



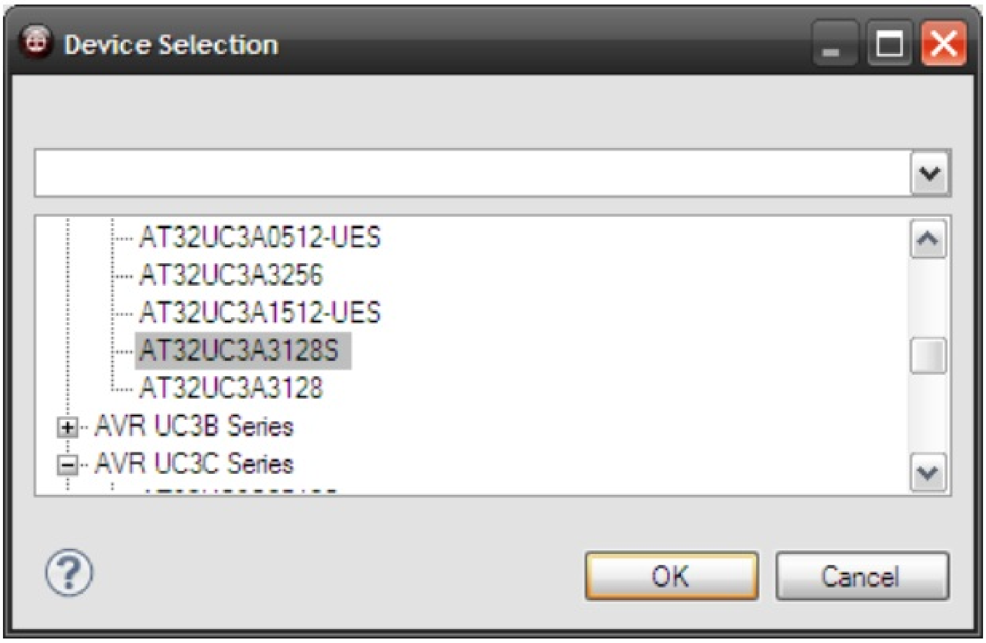
☑ Prompt when enabling activities

- □ Advanced C/C++
  - □ Build Configuration
  - □ Launch Configurations
  - □ Perspective
  - □ Source Navigation
  - □ Wizards
- □ Classic Update
- ■ General
  - □ Annotation Navigation
  - □ Issue Tracking
  - □ Remote System Explorer
  - ☑ Revision Control
  - ☑ Search

Tool Paths. The automatically determined value will still serve as the default value. Note that if the *AVR Utilities* and *AVR Toolchains* are installed as a part the IDE (as described above) the paths specified here will get lower priority.

The user interface has been simplified and several of the more “advanced” features have been hidden away. However these are still available and can be activated by altering the preference settings at **Preferences > General > Activities**.

Improved device selection



The device selection dialog has been improved. It will now allow you to perform a simple substring search for a device name and it will remember the devices last used. Full names are now used for all devices. The new project wizard will always start with the last used device if any.

New features added

Report #9558: AVR C project from template should use the board MCU.

It is no longer required to specify which device to use when creating a new project using the “AVR32 C Project From Template”. The device specified in the template will automatically be used.

Report #10477: Added support for QT600 development kit.

QT600 offers a powerful environment for the designer to evaluate and design touch-based solutions. The QT600’s scalable design allows the designer to use their own touch sensor boards with various microcontroller boards or to connect QT600 sensor boards directly to their own application.

Report #11205: Include the UC3 software framework version 1.7.

The UC3 Software framework provides software drivers and libraries to build any application for AVR32 UC3 devices. It has been designed to help develop and glue together the different components of a software design, and to be easily integrable into an operating system (OS) as well as to operate in a stand-alone way. This release contains version 1.7 of the software framework.

Report #11273: Add a “simplified” perspective/mode.

The user interface has been simplified and many of the more advanced features has been hidden away. These are still available and can be activated using the preference settings found at “General > Activities”.

Report #11625: Include AVR Utilities as an (optional) plug-in.

AVR Utilities are now included into the product build. This means there should be no need to install these separately on Windows or Linux. If you do install AVR Utilities separately the included version will still be used and must be removed if the external version is to be used.

Report #11628: Include AVR Toolchain as an (optional) plug-in.

The AVR Toolchains are now included into the product build. This means there should be no need to install these separately on Windows or Linux. If you do install AVR Toolchains separately the included version will still be used and must be removed if the external version is to be used.

Notable bugs fixed

Report #8963: Interrupt trigged during breakpoint halt causes debugger to loose track.

Interrupt triggered during breakpoint halt causes debugger to loose track

Report #10725: Changes in included header files do not trigger build.

When an included header file placed in a sub-folder of a project is changed it will not trigger a re-build of the project. Simply pressing CTRL+B or by other means invoking a build will not do anything as the change has not been detected. A clean build must be performed instead. Note that a change in a source file will trigger a new build.

Report #11226: Buttons functionality problem with GTK+ 2.18.

AVR32 Studio does not work properly with GTK+ 2.18. Various buttons are not enabled and the GUI does not paint as expected. This problem is caused by incompatibility between this new version of GTK and Eclipse SWT. Executing "export GDK_NATIVE_WINDOWS=true" before launching AVR32 Studio should restore normal behaviour. See **https://bugs.eclipse.org/bugs/show_bug.cgi?id=291257** for more information.

Report #7497: Improve behaviour when source file cannot be found while debugging.

When entering in debug mode, if an external library is used and not found, the debugger is stopped.

Report #9462: Drivers Include Path not set in AVR32 CPP project.

Executing the UC3 software framework wizard on a C++ project would not update all project settings. For instance the include path would be left out. This has now been fixed.

Report #9828: PM/GCCTRL5 is missing from device description.

The AVR32 Register view in AVR32 Studio is not working properly and sometimes it is missing

Report #10818: Strange target configuration behaviour.

When using a shortcut ("target" > Debug > "project") to debug a target the device could be changed to that of the project. However the "board" if set would not change and may cause an invalid configuration. This has been fixed.

Report #10907: AVR32 Studio framework plug-in issue.

Running the software framework wizard on a project created using a previous versions of the software framework would not update changed files unless the files had been changed locally. Changed files will now also be upgraded to the latest version. A dialog will ask for confirmation before overwriting files.

Report #11167: "UC3 Software Framework" disappeared.

Closing a project that had a software framework link would also close the link for all other projects using the same software framework. This has been fixed.

Report #11318: Device setting on source file defaults to “ap7000”.

In certain cases when having build settings for a specific file; the default device (AP7000) would kick in so that “-mpart=ap7000” would be applied. This has been fixed.

Report #11584: JTAGICE mkII debug launch delay (ticket 577114).

When using debugging on Ubuntu Karmic there was a long pause (30 sec) after connecting to the trace port on avr32gdbproxy. This has been fixed and debugging proceeds as normal.

Report #11021: Update IDE documentation and rename “AVR32” to “32-bit AVR”.

Due to the rebranding of AVR32 into AVR use of “AVR32” is changed into “32-bit AVR” in the documentation. Certain elements in the user interface is renamed from “AVR32” to “AVR”. The name of the IDE is still “AVR32 Studio”.

Known issues

Report #11836: Cannot start AUX trace on EVK1105.

All modes of AUX trace (buffered/streaming) cannot be used on an EVK1105. There is no workaround for now except using NanoTrace.

Report #5716: AVR32Studio unresponsive when stepping through for loop.

Stepping over a line of source code that results in a large amount of machine instructions to be executed (typically empty for or while loops used for delays) will cause AVR32 Studio to be unresponsive. To regain control, terminate the launch. To step over such code line, use breakpoints and the resume (F8) function.

Report #7280: editor vertical ruler context menu confuses tracepoints with breakpoints.

If a breakpoint and a tracepoint is located on the same source line it is not possible to open the breakpoint’s properties from the context (right-click) menu. In such cases, access the breakpoint from the Breakpoints view.

Report #7596: Display of the assembly lines.

The contents of the Disassembly view may be displayed non-sequential depending on the output of the compiler. Typically, the presentation of for-loops or optimized code may be unfamiliar to some users.

Report #8525: META Can’t expand structs for peripheral with write-only registers.

When attempting to expand structs pointing to peripheral memory containing write-only registers (for instance for struct avr32_usart_t), an error “Duplicate variable object name” occurs.

Report #10857: DMACA registers can not be displayed.

The DMACA registers for the UC3A3 do not properly display when in the debugger. They remain constant despite any changes… both the register view and the memory view show FB forever in that memory range. The service access bus (SAB) can not access DMACA registers. There is no workaround.

Report #7099: Verify when programming for debug launch.

The launch configuration setting “Verify memory after programming” will not be effective for debug launches.

Report #7370: ‘includes’ folder from Project Explorer only display includes from debug target.

The Includes folder for projects will only display includes for the Debug configuration.

Report #7707: file redirection in post-build or pre-build doesn’t work.

It is not possible to use redirection in Pre-build or Post-build steps. A workaround is to create an external command (i.e a .bat file) that performs the necessary redirection.

Report #11834: FLASHC example for AT32UC3A0512UES does not compile with AVR32 Studio 2.6.

The linker script used in this version of the UC3 Software Framework was written for an older version of the compiler and will not work with the current release. If you need to do development on these older UC3 devices, please use the 2.5 release of AVR32 Studio with accompanying toolchain.

Supported Devices

The following tables lists all supported tools and devices and shows which tools support debugging and programming of the various devices.

We have three kinds of support. “Control” support means that the device can only be programmed and controlled through the target context menu. By “debug” we mean a starting a debugging session through the launch mechanism and that the target context menu can be used. Similarily “run” means programming and starting the application through the launch mechanism (but no debugging). “Full” means that all these kinds are supported.

Required firmware versions

Debugger/programmer	**Firmware version**
AVR Dragon	MCU 6.11:MCU_S1 6.11
AVR ONE!	MCU 4.16:FPGA 4.0:FPGA 3.0:FPGA 2.0
JTAGICE mkII	MCU 6.6:MCU_S1 6.6
QT600	MCU 1.5
STK600	MCU 2.11:MCU_S1 2.1:MCU_S2 2.1

AVR AP7 Series

AVR Dragon		AVR ONE !	AVR32 Simulator	JTAGICE mkII	QT600	STK600	USB DFU
AT32AP7000	Full	Full	N/A	Full	N/A	N/A	N/A

AVR UC3A Series

AVR Dragon		AVR ONE !	AVR32 Simulator	JTAGICE mkII	QT600	STK600	USB DFU
AT32UC3A0128	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A0256	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A0512	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A0512-UES	Full	Full	Debug	Full	N/A	N/A	Control
AT32UC3A1128	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A1256	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A1512	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A1512-UES	N/A	N/A	Debug	N/A	N/A	N/A	Control
AT32UC3A3128	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A3128S	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A3256	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A3256S	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A364	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3A364S	Full	Full	Debug	Full	N/A	Run	Control

AVR UC3B Series

AVR Dragon		AVR ONE !	AVR32 Simulator	JTAGICE mkII	QT600	STK600	USB DFU
AT32UC3B0128	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3B0256	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3B0256-UES	Full	Full	Debug	Full	N/A	N/A	Control

AVR Dragon		AVR ONE !	AVR32 Simulator	JTAGICE mkII	QT600	STK600	USB DFU
AT32UC3B0512	N/A	Full	Debug	Full	N/A	Run	Control
AT32UC3B0512 (Revision C)	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3B064	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3B1128	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3B1256	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3B1256-UES	N/A	N/A	Debug	N/A	N/A	N/A	Control
AT32UC3B164	Full	Full	Debug	Full	N/A	Run	Control

AVR UC3C Series

AVR Dragon		AVR ONE !	AVR32 Simulator	JTAGICE mkII	QT600	STK600	USB DFU
AT32UC3C0512C (Revision C)	Full	Full	N/A	Full	N/A	Run	Control
AT32UC3C1512C (Revision C)	Full	Full	N/A	Full	N/A	Run	Control
AT32UC3C2512C (Revision C)	Full	Full	N/A	Full	N/A	Run	Control

AVR UC3L Series

AVR Dragon		AVR ONE !	AVR32 Simulator	JTAGICE mkII	QT600	STK600	USB DFU
AT32UC3L016	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3L032	Full	Full	Debug	Full	N/A	Run	Control
AT32UC3L064	Full	Full	Debug	Full	Run	Run	Control
AT32UC3L064 (Revision B)	Full	Full	N/A	Full	N/A	Run	Control

FAQ

Q: What processors are supported by AVR32 Studio?

A: AVR32 Studio supports all of Atmel’s AVR 32-bit processors.

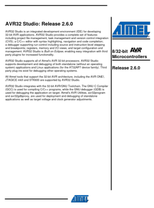
Q: Can AVR32 Studio be installed on Windows 98 or NT?

A: No, AVR32 Studio is not supported on Windows 98 or NT.

Q: Where can I find the AVR Toolchains package required for AVR32 Studio?

A: The AVR Toolchains package can be found on Atmel’s website under the Tools & Software menu.

Documents / Resources

 AVR32 Studio: Release 2.6.0 AVR32 Studio is an integrated development environment (IDE) for developing 32-bit AVR applications. AVR32 Studio provides a complete set of features including project file management, task management and version control integration (CVS); a C/C++ editor with syntax highlighting, navigation and code completion; a debugger supporting run control including source and instruction-level stepping and breakpoints; registers, memory and I/O views; and target configuration and management. AVR32 Studio is *Built on Eclipse*, enabling easy integration with third party plugins for increased functionality. AVR32 Studio supports all of Atmel's AVR 32-bit processors. AVR32 Studio supports development and debugging of both standalone (without an operating system) applications and Linux applications (for the AT32AP7 device family). Third party plug-ins exist for debugging other operating systems. All Atmel tools that support the 32-bit AVR architecture, including the AVR ONE!, JTAGICE mkII and STK600 are supported by AVR32 Studio. AVR32 Studio integrates with the 32-bit AVR/GNU Toolchain. The GNU C Compiler (GCC) is used for compiling C/C++ programs, while the GNU debugger (GDB) is used for debugging the application on target. Atmel's AVR Utilities, avr32program and avr32gdbproxy, are used for deployment and debugging of standalone applications as well as target voltage and clock generator adjustments. 8/32-bit AVR Microcontrollers Release 2.6.0	**ATMEL AVR32 32 Bit Micro Controllers** [pdf] Instruction Manual AVR ONE, JTAGICE mkII, STK600, AVR32 32 Bit Micro Controllers, AVR32, 32 Bit Micro Controllers, Bit Micro Controllers, Micro Controllers, Controllers

References

- **The Community for Open Collaboration and Innovation \| The Eclipse Foundation**
- **Oracle Java Technologies \| Oracle**
- **Empowering Innovation \| Microchip Technology**
- **AVR Freaks**
- **Eclipse Downloads \| The Eclipse Foundation**
- **Bug List**
- **Bug List**
- **Bug List**
- **291257 – [Widgets] Buttons functionality problem with GTK+ 2.18**
- **User Manual**