
AVR 32-bit GNU Toolchain: Release 3.4.0.332

The AVR 32-bit GNU Toolchain supports all AVR 32-bit devices. The AVR 32-bit Toolchain is based on the free and open-source GCC compiler. The toolchain includes compiler, assembler, linker and binutils (GCC and Binutils), Standard C library (Newlib).



8/32-bit **AVR**[®]
Microcontrollers

Release 3.4.0.332





Installation Instructions

System Requirements

AVR 32-bit GNU Toolchain is supported under the following configurations

Hardware requirements

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

AVR 32-bit GNU Toolchain 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).
- 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 11.2 or 11.1 (x86 or x86-64). AVR 32bit GNU Toolchain may very well work on other distributions. However those would be untested and unsupported.

AVR 32-bit GNU Toolchain is not supported on Windows 98, NT or ME.

Downloading and Installing

The package comes in two forms.

- As part of a standalone installer (avr-toolchain-installer)
- As part of Atmel Studio 6 installer

It can be downloaded from Atmel's website at <http://www.atmel.com>

Installing on Windows

Atmel Studio 6 installation procedure will also install AVR Toolchain. See Atmel Studio 6 release notes for more details.

In order to install using standalone installer, the AVR Toolchain installer can be downloaded from Atmel website. After downloading the installer, double-click the executable file to install. You may use "Custom Installation" in order to install in a specific location.

Installing on Linux

On Linux AVR 32-bit GNU Toolchain is available as a TAR.GZ archive which can be extracted using the 'tar' utility. In order to install, simply extract to the location where you want the toolchain to run from.

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

Upgrading from previous versions

If it is installed via Atmel Studio 6 it will be upgraded through the Atmel Studio 6 upgrade. See Atmel Studio 6 release notes for more information.

If you have used the standalone installer on MS-Windows, you might do a clean upgrade by first un-installing the old version or just upgrade using the latest installer.

On Linux, if you have it unpacked to a local folder, you just delete the old folder and unpack the latest version in a new folder.

Manifest

1. AVR GNU Binutils 2.22
 - Binary utilities for AVR 32-bit target (including assembler, linker, etc.).
2. AVR 32-bit GNU Compiler Collection (avr32-gcc) 4.4.3
 - C language and C++ language compiler for AVR 32-bit target.
3. Newlib (for AVR 32-bit) 1.16.0
 - Standard C Library for AVR 32-bit

Layout

Listed below are some directories you might want to know about.

`<install_dir>` = The directory where you installed AVR 32-bit GNU Toolchain.

- `<install_dir>\bin`
 - The AVR software development programs. This directory should be in your `PATH` environment variable. This includes:
 - GNU Binutils
 - GCC
- `<install_dir>\avr32\lib`
 - avr32-newlib libraries, startup files, linker scripts, and stuff.
- `<install_dir>\avr32\include`
 - avr32-newlib header files. This is where, for example, `#include <string.h>` comes from.
- `<install_dir>\avr32\include\avr32`
 - avr32-newlib header files specific to the AVR32 microprocessor. This is where, for example, `#include <avr32/io.h>` comes from.
- `<install_dir>\lib`
 - GCC libraries, other libraries, headers and stuff.
- `<install_dir>\libexec`
 - GCC program components
- `<install_dir>\doc`
 - Various documentation.
- `<install_dir>\source`
 - Documentation on where to find the source code for the various projects and source code patches that were used to build the tools.

Toolset Background

AVR 32-bit GNU Toolchain is a collection of executable software development tools for the Atmel AVR 32-bit processor.

These software development tools include:

1. Compiler

2. Assembler
3. Linker
4. Archiver
5. File converter
6. Other file utilities
7. C Library

Compiler

The compiler is the GNU Compiler Collection, or GCC. This compiler is incredibly flexible and can be hosted on many platforms, it can target many different different processors/operating systems (back-ends), and can be configured for multiple different languages (front-ends).

The GCC included is targeted for the AVR 32-bit processor, and is configured to compile C, or C++.

Because this GCC is targeted for the AVR 32-bit, the main executable that is created is prefixed with the target name: ``avr32-gcc``. It is also referred to as AVR 32-bit GCC.

``avr32-gcc`` is just a "driver" program only. The compiler itself is called ``cc1.exe`` for C, or ``cc1plus.exe`` for C++. Also, the preprocessor ``cpp.exe`` will usually automatically be prepended with the target name: ``avr32-cpp.exe``. The actual set of component programs called is usually derived from the suffix of each source code file being processed.

GCC compiles a high-level computer language into assembly, and that is all. It cannot work alone. GCC is coupled with another project, GNU Binutils, which provides the assembler, linker, librarian and more. Since GCC is just a "driver" program, it can automatically call the assembler and linker directly to build the final program.

Assembler, Linker, Librarian and More

GNU Binutils is a collection of binary utilities. This also includes the assembler, `as`. Sometimes you will see it referenced as GNU `as` or `gas`. Binutils includes the linker, `ld`; the librarian or archiver, `ar`. There are many other programs included that provide various functionality.

Binutils is configured for the AVR 32-bit target and each of the programs is prefixed with the target name. So you have programs such as:

- **avr32-as**: The Assembler.
- **avr32-ld**: The Linker.
- **avr32-ar**: Create, modify, and extract from archives (libraries).
- **avr32-ranlib**: Generate index to archive (library) contents.
- **avr32-objcopy**: Copy and translate object files.
- **avr32-objdump**: Display information from object files including disassembly.
- **avr32-size**: List section sizes and total size.
- **avr32-nm**: List symbols from object files.
- **avr32-strings**: List printable strings from files.
- **avr32-strip**: Discard symbols.
- **avr32-readelf**: Display the contents of ELF format files.
- **avr32-addr2line**: Convert addresses to file and line.
- **avr32-c++filt**: Filter to demangle encoded C++ symbols.

See the binutils user manual for more information on what each program can do.

C Library

Newlib is the Standard C Library for AVR 32-bit GCC. Newlib is a C library intended for use on embedded systems. It is a conglomeration of several library parts. The library is ported to support the AVR 32-bit processor.

Debugging

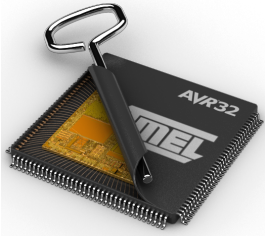
Atmel Studio 6 provides a debugger and also provides simulators for the parts that can be used for debugging as well. Note that `Atmel Studio 6` is currently free to the public, but it is not Open Source.

New and Noteworthy

This chapter lists new and noteworthy items for the AVR 32-bit GNU Toolchain release.

AVR 32-bit GNU Toolchain

Supported devices



AVR 32-bit GNU Toolchain supports the following devices:

Note:- Devices which are newly supported in this release are marked with *

uc3a0128	uc3a0256	uc3a0512	uc3a0512es	uc3a1128
uc3a1512	uc3a1512es	uc3a3revd	uc3a364	uc3a364s
uc3a3128s	uc3a3256	uc3a3256s	uc3a464	uc3a464s
uc3a4128s	uc3a4256	uc3a4256s	uc3b064	uc3b0128
uc3b0256es	uc3b0512	uc3b0512revc	uc3b164	uc3b1128
uc3b1256es	uc3b1512	uc3b1512revc	uc64d3	uc128d3
uc128d4	uc3c0512crevc	uc3c1512crevc	uc3c2512crevc	uc3l0256
uc3l064	uc3l032	uc3l016	uc3l064revb	uc3c064c
uc3c0256c	uc3c0512c	uc3c164c	uc3c1128c	uc3c1256c
uc3c264c	uc3c2128c	uc3c2256c	uc3c2512c	

Improvements

- PR 239: New option: `-munsafe-math` : Compiler has been generating FPU instructions to emulate division and sqrt when `-funsafe-math-optimizations` was enabled. This option will enable emission of instructions that may be unsafe/approximate in the IEEE standard perspective but may improve speed. This has been modified such that `-munsafe-math` will enable this behavior. The option `-funsafe-math-optimization` by default will enable `-munsafe-math`. This is to allow user to use the AVR32 specific non-standard operations without enabling all other unsafe math optimizations.
- PR 200: Changes to `-mhard-float` option: Previously it was required to pass `-mhard-float` even when the device supports FPU in order to generate floating point instructions instead of using software emulation of floating point operations. Now this has been changed. When `-mpart=` specifies a device that supports FPU, the floating point instructions emission is turned on by default. Passing `-msoft-float` will turn this off. Passing `-mhard-float` to any device that doesn't support FPU will result in compilation error. `-mhard-float` option is no more a reverse of `-msoft-float`. `-mhard-float` is deprecated, from now on.

Component Upgrades

- Binutils upgraded to 2.22 from 2.20.1

Issues Fixed

- PR 170: Decoding "sub" instruction argument is incorrect as assembler checks the incorrect range. This is now fixed
- PR 102-1: Alignment done twice by assembler when it is expected to do it once. This is now fixed
- PR 102-2: Wrong displacement computation by assembler. This is now fixed



Contact Information

For support on AVR 32-bit GNU Toolchain please contact avr32@atmel.com.

Users of AVR 32-bit GNU Toolchain are also welcome to discuss on the AVRfreaks website forum for AVR32 Software Tools.

Disclaimer and Credits

AVR 32-bit GNU Toolchain 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. AVR 32-bit GNU Toolchain comes without any warranty.

Copyright 2006-2012 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.