RELEASE NOTES

GNU Toolchain for Atmel AVR 32-bit Embedded Processors

Introduction

The Atmel AVR® 32-bit GNU Toolchain (3.4.3.820) 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).

Table of Contents

1. Installation Instructions

1.1 System Requirements

1.1.1 Hardware Requirements

AVR 32-bit GNU Toolchain is supported on the following configuration.

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

1.1.2 Software Requirements

- Windows 2000, Windows XP, Windows Vista®, Windows 7 (x86 or x86-64), or Windows 8 (x86 or x86-64)
- Fedora® 13 or 12 (x86 or x86-64), Red Hat® Enterprise Linux® 4/5/6, Ubuntu® Linux 10.04 or 8.04 (x86 or x86-64), or SUSE Linux 11.2 or 11.1 (x86 or x86-64). AVR 32-bit GNU Toolchain may very well work on other distributions. However, those would be untested and unsupported.

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

1.2 Downloading, Installing, and Upgrading

The AVR 32-bit GNU Toolchain provided by Atmel is available for use in the following ways.

1.2.1 Downloading/Installing on Windows

- If you want to try the Atmel AVR 32-bit Toolchain alone, you can download it from here[1]
- If you want to try the Atmel AVR 32-bit GNU Toolchain along with Atmel studio, you can download and install Atmel studio 6.0 or (newer) which will also install the Atmel AVR 32-bit GNU toolchain. See Atmel studio release notes for more details.

1.2.2 Downloading/Installing on Linux

For 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. Linux builds are available from here[2].

Note that if you will develop Linux applications for the AT32AP7000 you must also install the AVR32 Buildroot. For more details on AVR32 Buildroot refer here[3].

1.2.3 Upgrading from previous versions

If AVR 32-bit GNU Toolchain is installed by Atmel Studio installation, refer Atmel Studio release note for more details.

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.

1.2.4 Manifest

1. AVR 32-bit GNU Binutils 2.23.1

   Binary utilities for AVR 32-bit target (including assembler, linker etc).

2. AVR 32-bit GNU Compiler collection (avr32-gcc) 4.4.7

   C, C++ language compiler for the AVR 32-bit target.

[1] http://www.atmel.com/tools/ATMELAVRTOOLCHAINFORWINDOWS.aspx
[2] http://www.atmel.com/tools/ATMELAVRTOOLCHAINFORLINUX.aspx
[3] http://www.atmel.com/tools/MATUREBUILDROOTFORAVR32.aspx

3. Newlib (AVR 32-bit) 1.16.0

   Standard C library for AVR 32-bit.

1.3 Layout

Listed below are some of the directories that you might want to look into, to have a high level understanding of what is packaged inside the Atmel AVR 32-bit GNU Toolchain. The layout is identical in Windows and Linux.

- **INSTALLDIR**

  The directory where the AVR 32-bit GNU Toolchain is installed in the target machine.

- **INSTALLDIR\bin**

  The AVR 32-bit software development programs. This directory should be in your PATH environment variable. (Note: If you are using this toolchain from within Atmel Studio, configure Atmel Studio appropriately). This includes:

  - GNU Binutils
  - GCC

- **INSTALLDIR\avr32\lib**

  The directory which have the AVR 32-bit newlib libraries, startup files, and linker scripts.

- **INSTALLDIR\avr32\include**

  AVR 32-bit newlib header files. This is where the system header files will be searched for by the toolchain.

- **INSTALLDIR\lib**

  GCC libraries, other libraries, and headers.

- **INSTALLDIR\libexec**

  GCC program components.

- **INSTALLDIR\doc**

  Various documentation.

2. Toolset Background

AVR 32-bit GNU toolchain is a collection of executable software development tools for the Atmel AVR 32-bit microcontrollers. These software development tools include:

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

2.1 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 processors/operating systems (backends), and can be configured for multiple different languages (frontends).

The GCC included is targeted for the AVR 32-bit UC3 processors, and is configured to compile C, and 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 AVR32 GCC.

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

GCC compiles a high-level computer lanugage 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.

2.2 Assembler, Linker, and Librarian

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 GNU Assembler
- **avr32-ld**: The GNU Linker
- **avr32-ar**: The GNU Archiver, create, modify, and extract from archives (libraries)
- **avr32-ranlib**: Generate index of archive (library) contents
- **avr32-objcopy**: Copy and translate object files
- **avr32-objdump**: Display information from object files including disassembly
- **avr32-size**: List section size, total size
- **avr32-nm**: List symbol from object files
- **avr32-strings**: List printable strings from files
- **avr32-strip**: Discard symbols
- **avr32-readelf**: Display the contents of ELF file formats

- **avr32-addr2line**: Convert addresses to file and line
- **avr32-c++filt**: Filter to demangle encoded C++ symbols
- **avr32-gdb**: Debugger to debug the target

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

2.3 C Library

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

2.4 Debugging

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

2.5 Source Code

Atmel AVR 32-bit GNU Toolchain uses modified source code of GCC, Binutils, and Newlib. The source code used for building the packaged binaries are available here[1]. The modifications are also available as patches in the above link. Refer to the README for the instructions on how to use the supplied script to build the toolchain.

[1] http://distribute.atmel.no/tools/opensource/Atmel-AVR32-GNU-Toolchain/3.4.3/

3. New and Noteworthy

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

3.1 Supported devices

uc3a0128	uc3a0256	uc3a0512	uc3a0512es
uc3a1128	uc3a1512	uc3a1512es	uc3a3revd
uc3a364	uc3a364s	uc3a3128s	uc3a3256
uc3a3256s	uc3a464	uc3a464s	uc3a4128
uc3a4128s	uc3a4256	uc3a4256s	uc3b064
uc3b0128	uc3b0256es	uc3b0512	uc3b0512revc
uc3b164	uc3b1128	uc3b1256es	uc3b1512
uc3b1512revc	uc3c0512crevc	uc3c1512crevc	uc3c2512crevc
uc3c064c	uc3c0256c	uc3c0512c	uc3c164c
uc3c1128c	uc3c1256c	uc3c264c	uc3c2128c
uc3c2256c	uc3c2512c	uc64d3	uc128d3
uc64d4	uc128d4	uc3l016	uc3l032
uc3l064	uc3l064revb	uc3l0128	uc3l0256
uc64l3u	uc128l3u	uc256l3u	uc64l4u
uc128l4u	uc256l4u		

3.2 Issues Fixed

- Few of the device headers are updated

4. Contact Information and Disclaimer

For support on Atmel AVR 32-bit GNU Toolchain, visit design support[1].

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

4.1 Disclaimer

AVR 32-bit GNU Toolchain is distributed free of charge for the purpose of developing applications for Atmel AVR processors. AVR 32-bit GNU Toolchain comes without any warranty.

[1] http://www.atmel.com/design-support/

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 **| www.atmel.com**