

---

## **GNU Toolchain for Microchip AVR8 Embedded Processors**

---

### **Introduction**

---

The AVR 8-bit GNU Toolchain (3.6.2.1778) supports all AVR 8-bit devices. The AVR 8-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 (AVR-libc) and GNU Debugger (GDB).

## Table of Contents

|  |           |
|--|-----------|
| Introduction .....                                   | 1         |
| <b>1. Installation Instructions .....</b>            | <b>3</b>  |
| 1.1. System requirements .....                       | 3         |
| 1.1.1. Hardware requirements .....                   | 3         |
| 1.1.2. Software Requirements .....                   | 3         |
| 1.2. Downloading, Installing and Upgrading .....     | 3         |
| 1.2.1. Downloading/Installing on Windows .....       | 3         |
| 1.2.2. Downloading/Installing on Linux and Mac ..... | 3         |
| 1.2.3. Upgrading from previous versions .....        | 3         |
| 1.3. Layout .....                                    | 3         |
| <b>2. Toolset Background .....</b>                   | <b>5</b>  |
| 2.1. Component Versions .....                        | 5         |
| 2.2. Compiler .....                                  | 5         |
| 2.3. Assembler, Linker, Librarian and More .....     | 5         |
| 2.4. C Library .....                                 | 6         |
| 2.5. Debugging .....                                 | 6         |
| 2.6. Source Code .....                               | 6         |
| <b>3. Bugs and New Features .....</b>                | <b>7</b>  |
| 3.1. Notable Bugs Fixed .....                        | 7         |
| 3.2. Known Issues .....                              | 7         |
| <b>4. Supported Devices .....</b>                    | <b>8</b>  |
| <b>5. Contact Information and Disclaimer .....</b>   | <b>10</b> |
| 5.1. Disclaimer .....                                | 10        |

# 1. Installation Instructions

## 1.1 System requirements

### 1.1.1 Hardware requirements

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

AVR 8-bit GNU Toolchain has not been tested on computers with less resources, but may run satisfactorily depending on the number and size of the projects and the user's patience.

### 1.1.2 Software Requirements

- Windows 2000, Windows XP, Windows Vista, Windows 7 (x86 or x86-64) or Windows 8 (x86 or x86-64)
- AVR 8-bit GNU Toolchain is not supported on Windows 98, NT or ME.
- The toolchain should work on the Linux distributions Fedora, RedHat Enterprise, Arch Linux and Ubuntu for both 32-bits and 64-bits architecture. AVR 8-bit GNU Toolchain may very well work on other distributions. However those are untested and unsupported.

## 1.2 Downloading, Installing and Upgrading

The AVR8 GNU toolchain provided by Microchip is available for download and install in one of the following ways.

### 1.2.1 Downloading/Installing on Windows

- If you want to try the AVR8 GNU toolchain alone, you can download it from Microchip's website
- If you want to try the AVR8 GNU Toolchain along with Atmel Studio, you can download and install Atmel Studio 7 or (newer) which will also install the AVR8 GNU toolchain. See Atmel Studio release notes for more details.

### 1.2.2 Downloading/Installing on Linux and Mac

For Linux and Mac, the AVR8 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 from where you want to run it from. Linux and Mac builds are available from Microchip's website.

### 1.2.3 Upgrading from previous versions

If the AVR8 GNU Toolchain is installed by Atmel Studio installation, refer Atmel Studio documentation to upgrade.

If the toolchain is installed separately using one of the (Windows, Linux, Mac) installers, upgrading is not supported. You can install the new package side-by-side of the old package and use it.

## 1.3 Layout

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

`<install_dir>` = The directory where you installed AVR 8-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
  - GDB

- `<install_dir>\avr\lib`  
avr-libc libraries, startup files, linker scripts, and stuff.
- `<install_dir>\avr\include`  
avr-libc header files for AVR 8-bit.
- `<install_dir>\avr\include\avr`  
header files specific to the AVR 8-bit MCU. This is where, for example, `#include <avr/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.

## 2. Toolset Background

AVR 8-bit GNU Toolchain is a collection of executable, open source software development tools for the Microchip AVR 8-bit series of microcontrollers. It includes the GNU GCC compiler for C and C++.

### 2.1 Component Versions

GCC: 5.4.0

binutils: 2.26.20160125

avr-libc: "2.0.0"

gdb: 7.8

### 2.2 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 (back-ends), and can be configured for multiple different languages (front-ends).

The GCC included in AVR 8-bit GNU Toolchain is targeted for the AVR 8-bit microcontroller and is configured to compile C or C++.

**CAUTION:** There are caveats on using C++. See the avr-libc FAQ. C++ language is not fully supported and has some limitations. libstdc++ is unsupported.

Because this GCC is targeted for the AVR 8-bit MCUs, the main executable that is created is prefixed with the target name: ``avr-gcc`` (with `.exe` extension on MS Windows). It is also referred to as AVR GCC.

``avr-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: ``avr-cpp``. 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.

### 2.3 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.

Note that while the assembler uses the same mnemonics as proposed by Microchip, the "glue" (pseudo-ops, operators, expression syntax) is derived from the common assembler syntax used in Unix assemblers, so it is not directly compatible to Microchip AVR assembler source files.

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

- `avr-as`: The Assembler.
- `avr-ld`: The Linker.
- `avr-ar`: Create, modify, and extract from archives (libraries).
- `avr-ranlib`: Generate index to archive (library) contents.
- `avr-objcopy`: Copy and translate object files.
- `avr-objdump`: Display information from object files including disassembly.
- `avr-size`: List section sizes and total size.
- `avr-nm`: List symbols from object files.
- `avr-strings`: List printable strings from files.
- `avr-strip`: Discard symbols.

- *avr-readelf*: Display the contents of ELF format files.
- *avr-addr2line*: Convert addresses to file and line.
- *avr-c++filt*: Filter to demangle encoded C++ symbols.
- *avr-gdb*: GDB, the GNU debugger, allows you to see what is going on 'inside' another program targeted to AVR, while it executes.

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

## 2.4 C Library

avr-libc is the Standard C Library for AVR 8-bit GCC. It contains many of the standard C routines, and many non-standard routines that are specific and useful for the AVR 8-bit MCUs.

In addition to avr-libc libraries, Host IO library (libhostio.a) is integrated to this toolchain. This Host IO library allows the target to use the host's file system and console I/O to perform various avr I/O operations.

*NOTE:* The actual library is currently split into two main parts, libc.a and libm.a, where the latter contains mathematical functions (everything mentioned in <math.h>, and a bit more). Also, there are additional libraries which allow a customization of the printf and scanf function families. avr-libc contains documentation on how to use (and build) the entire toolset, including code examples. The avr-libc user manual also contains the FAQ on using the toolset.

## 2.5 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. The GNU debugger is now shipped along with the toolchain.

## 2.6 Source Code

AVR8 GNU Toolchain uses modified source code from GCC, Binutils and AVR-LibC. The source code and the build scripts used for building the packaged binaries are available in Microchip's website.

Please refer to the README for the instructions on how to use the supplied script to build the toolchain.

## 3. Bugs and New Features

### 3.1 Notable Bugs Fixed

**Issue #AVRTC-870:**

Updated eeprom\_is\_ready for avrxmega3 devices with new NVM control registers.

**Issue #AVRTC-871:**

An intermittent segmentation fault has been corrected.

**Issue #AVRTC-872:**

FUSE MEMORY SIZE updated for ATtiny4/5/9/10/20/40 devices.

**Issue #AVRTC-876:**

Enabled clock\_prescale\_get/set functions for ATmega324PB and ATmega328PB devices.

**Issue #XC8-1796:**

Programs that exceeded that available RAM were not detected by the compiler in some situations, resulting in a runtime code failure.

**Issue #XC8-1822:**

Loop optimization causes an internal compiler error on Windows.

**Issue #XC8-1826:**

Unused volatile memory access (SFR reads, for e.g.) was optimized away in certain cases. This broke code that relied on the access for its side effects.

**Issue #XC8-1739:**

Fix PR 24564 - link fails for some rcalls/rjumps with wraparound.

**Issue #XC8-1889:**

Fix PR 24571 - Relaxation does not shorten jmp or call to target at pc-relative range boundary

### 3.2 Known Issues

**Issue #AVRTC-731:**

For AVRTINY architecture, libgcc implementation has some known limitations. Standard C / Math library implementation is very limited or not present.

**Issue #AVRTC-732:**

Program memory images beyond 128KBytes are supported by the toolchain, subject to the limitations mentioned in "3.17.4.1 EIND and Devices with more than 128 Ki Bytes of Flash" at <http://gcc.gnu.org/onlinedocs/gcc/AVR-Options.html>

**Issue #AVRTC-733:**

Named address spaces are supported by the toolchain, subject to the limitations mentioned in "6.16.1 AVR Named Address Spaces" at <http://gcc.gnu.org/onlinedocs/gcc/Named-Address-Spaces.html#AVR%20Named%20Address%20Spaces>

## 4. Supported Devices

Most of the AVR8 devices are supported by this toolchain. Users can get new devices support from Microchip Device Family Packs (DFP). Download DFPs from [here](http://packs.download.atmel.com/)<sup>1</sup>.

### Using DFPs with this toolchain:

- Download DFP which has required device support. (e.g. ATmega328PB is part of ATmega Series DFP.)
- Unzip downloaded \*.atpack file to packs directory (e.g. /home/packs/).
- Invoke avr-gcc with additional option -B to tell gcc where to look for device specific information and -I for device header include path.

*e.g. avr-gcc -mmcu=atmega328pb -B /home/packs/Atmel.ATmega\_DFP.1.0.86/gcc/dev/atmega328pb/ -I /home/packs/Atmel.ATmega\_DFP.1.0.86/include/*

#### avr2

|           |           |           |           |
|-----------|-----------|-----------|-----------|
| at90s2313 | at90s2343 | at90s4414 | at90s8515 |
| at90s2323 | attiny22  | at90s4433 | at90c8534 |
| at90s2333 | attiny26  | at90s4434 | at90s8535 |

#### avr25

|             |            |            |           |
|-------------|------------|------------|-----------|
| ata5272     | attiny4313 | attiny85   | attiny87  |
| ata6616c    | attiny44   | attiny261  | attiny48  |
| attiny13    | attiny44a  | attiny261a | attiny88  |
| attiny13a   | attiny441  | attiny461  | attiny828 |
| attiny2313  | attiny84   | attiny461a | attiny841 |
| attiny2313a | attiny84a  | attiny861  | at86rf401 |
| attiny24    | attiny25   | attiny861a |           |
| attiny24a   | attiny45   | attiny43u  |           |

#### avr3

|            |          |
|------------|----------|
| at43usb355 | at76c711 |
|------------|----------|

#### avr31

|           |            |
|-----------|------------|
| atmega103 | at43usb320 |
|-----------|------------|

#### avr35

|           |            |            |            |
|-----------|------------|------------|------------|
| ata5505   | at90usb82  | atmega16u2 | attiny1634 |
| ata6617c  | at90usb162 | atmega32u2 |            |
| ata664251 | atmega8u2  | attiny167  |            |

#### avr4

|          |            |            |           |
|----------|------------|------------|-----------|
| ata6285  | atmega48a  | atmega88pa | at90pwm2b |
| ata6286  | atmega48p  | atmega88pb | at90pwm3  |
| ata6289  | atmega48pa | atmega8515 | at90pwm3b |
| ata6612c | atmega48pb | atmega8535 | at90pwm81 |
| atmega8  | atmega88   | atmega8hva |           |
| atmega8a | atmega88a  | at90pwm1   |           |
| atmega48 | atmega88p  | at90pwm2   |           |

#### avr5

|             |                 |              |              |
|-------------|-----------------|--------------|--------------|
| ata5702m322 | atmega168pb     | atmega329a   | atmega649p   |
| ata5782     | atmega169       | atmega329p   | atmega6490   |
| ata8210     | atmega169a      | atmega329pa  | atmega16hva  |
| ata5790     | atmega169p      | atmega3290   | atmega16hva2 |
| ata5790n    | atmega169pa     | atmega3290a  | atmega32hvb  |
| ata5791     | atmega16hvb     | atmega3290p  | atmega6490a  |
| ata5795     | atmega16hvbrevb | atmega3290pa | atmega6490p  |
| ata5831     | atmega16m1      | atmega32c1   | atmega64c1   |
| ata8510     | atmega16u4      | atmega32m1   | atmega64m1   |
| ata6613c    | atmega32a       | atmega32u4   | atmega64hve  |
| ata6614q    | atmega32        | atmega32u6   | atmega64hve2 |
| atmega16    | atmega323       | atmega406    | atmega64rfr2 |

<sup>1</sup> <http://packs.download.atmel.com/>



|   |  |  |  |
|---|--|--|--|
| atmega16a<br>atmega161<br>atmega162<br>atmega163<br>atmega164a<br>atmega164p<br>atmega164pa<br>atmega165<br>atmega165a<br>atmega165p<br>atmega165pa<br>atmega168<br>atmega168a<br>atmega168p<br>atmega168pa | atmega324a<br>atmega324p<br>atmega324pa<br>atmega325<br>atmega325a<br>atmega325p<br>atmega325pa<br>atmega3250<br>atmega3250a<br>atmega3250p<br>atmega3250pa<br>atmega328<br>atmega328p<br>atmega328pb<br>atmega329 | atmega64<br>atmega64a<br>atmega640<br>atmega644<br>atmega644a<br>atmega644p<br>atmega644pa<br>atmega645<br>atmega645a<br>atmega645p<br>atmega6450<br>atmega6450a<br>atmega6450p<br>atmega649<br>atmega649a | atmega644rfr2<br>atmega32hvbrevb<br>at90can32<br>at90can64<br>at90pwm161<br>at90pwm216<br>at90pwm316<br>at90scr100<br>at90usb646<br>at90usb647<br>at94k<br>m3000 |
| <b>avr51</b><br><br>atmega128<br>atmega128a<br>atmega1280   | <br><br>atmega1281<br>atmega1284<br>atmega1284p  | <br><br>atmega128rfa1<br>atmega128rfr2<br>atmega1284rfr2   | <br><br>at90can128<br>at90usb1286<br>at90usb1287   |
| <b>avr6</b><br><br>atmega2560   | <br><br>atmega2561   | <br><br>atmega256rfr2  | <br><br>atmega2564rfr2   |
| <b>avrxmega2</b><br><br>atxmega8e5<br>atxmega16a4<br>atxmega16d4<br>atxmega16e5   | <br><br>atxmega32a4<br>atxmega32c3<br>atxmega32d3<br>atxmega32d4   | <br><br>atxmega16a4u<br>atxmega16c4<br>atxmega32a4u<br>atxmega32c4   | <br><br>atxmega32e5  |
| <b>avrxmega3</b><br><br>attiny212<br>attiny214<br>attiny412<br>attiny414  | <br><br>attiny416<br>attiny417<br>attiny814<br>attiny816   | <br><br>attiny817<br>attiny1614<br>attiny1616<br>attiny1617  | <br><br>attiny3216<br>attiny3217   |
| <b>avrxmega4</b><br><br>atxmega64a3<br>atxmega64d3  | <br><br>atxmega64a3u<br>atxmega64a4u   | <br><br>atxmega64b1<br>atxmega64b3   | <br><br>atxmega64c3<br>atxmega64d4   |
| <b>avrxmega5</b><br><br>atxmega64a1   | <br><br>atxmega64a1u   |  |  |
| <b>avrxmega6</b><br><br>atxmega128a3<br>atxmega128d3<br>atxmega192a3<br>atxmega192d3<br>atxmega256a3  | <br><br>atxmega256a3b<br>atxmega256a3bu<br>atxmega256d3<br>atxmega128a3u<br>atxmega128b1   | <br><br>atxmega128b3<br>atxmega128c3<br>atxmega128d4<br>atxmega192a3u<br>atxmega192c3  | <br><br>atxmega256a3u<br>atxmega256c3<br>atxmega384c3<br>atxmega384d3  |
| <b>avrxmega7</b><br><br>atxmega128a1  | <br><br>atxmega128a1u  | <br><br>atxmega128a4u  |  |
| <b>avrtiny</b><br><br>attiny4<br>attiny5  | <br><br>attiny9<br>attiny10  | <br><br>attiny20<br>attiny40   |  |
| <b>avr1</b><br><br>at90s1200<br>attiny11  | <br><br>attiny12<br>attiny15   | <br><br>attiny28   |  |

## **5. Contact Information and Disclaimer**

Users of AVR 8-bit GNU Toolchain are also welcome to discuss on the AVRFreaks website forum for AVR Software Tools.

### **5.1 Disclaimer**

AVR 8-bit GNU Toolchain is distributed free of charge for the purpose of developing applications for Microchip AVR processors. AVR 8-bit GNU Toolchain comes without any warranty.



**Microchip Technology Inc.** 2355 West Chandler Blvd., Chandler, Arizona, USA T: (+1)(480) 792-7200 F: (+1)(480) 792-7277 |

**[www.microchip.com](http://www.microchip.com)**

© 2019 Microchip Technology Inc. / Rev.: 42372A-MCU-06/2019

Microchip<sup>®</sup>, Microchip logo and combinations thereof, Enabling Unlimited Possibilities<sup>®</sup>, AVR<sup>®</sup>, tinyAVR<sup>®</sup>, XMEGA<sup>®</sup>, megaAVR<sup>®</sup>, and others are registered trademarks or trademarks of Microchip Technology Inc. in U.S. and other countries. Windows<sup>®</sup>, and others, are registered trademarks of Microsoft Corporation in U.S. and or other countries. Other terms and product names may be trademarks of others.

**DISCLAIMER:** The information in this document is provided in connection with Microchip products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Microchip products. EXCEPT AS SET FORTH IN THE MICROCHIP TERMS AND CONDITIONS OF SALES LOCATED ON THE MICROCHIP WEBSITE, MICROCHIP ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL MICROCHIP BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Microchip makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Microchip does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Microchip products are not suitable for, and shall not be used in, automotive applications. Microchip products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

**SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER:** Microchip products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death ("Safety-Critical Applications") without an Microchip officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Microchip products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by Microchip as military-grade. Microchip products are not designed nor intended for use in automotive applications unless specifically designated by Microchip as automotive-grade.