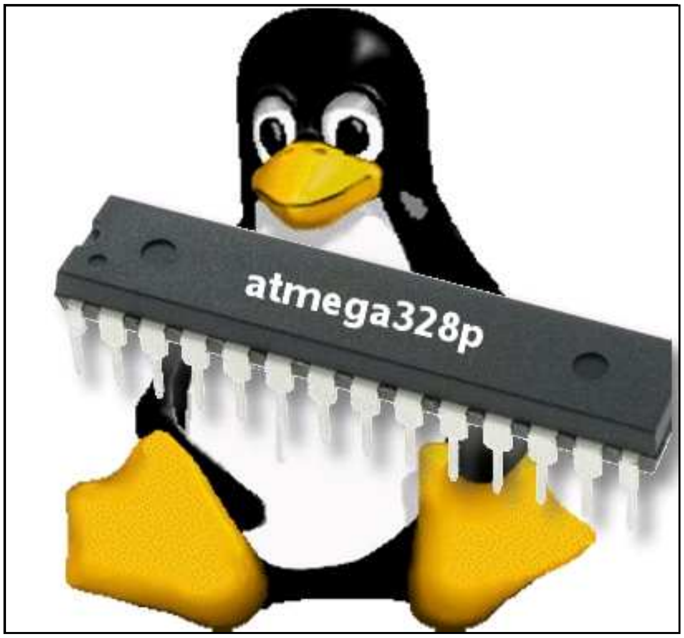
Linux: AVR microcontroller programming in C



Abstract:

The free avr-gcc development environment was originally written for Linux but until recently you had to compile the whole tool-chain yourself under Linux. Most distributions did not included it and there were only ready to use binaries available for Windows and Mac.

It is not necessarily a bad thing if you have to compile everything from the sources but it takes more time.

The tuxgraphics article Building your own avr-gcc environment with atmega328p capabilities, Linux described how to build the tool-chain and it has actually a ready to use binary in the download section of that article.

Atmel, the manufacturer of the AVR chips, provides now officially an "Atmel AVR Toolchain for Linux". It is tar-ball with ready to use binaries.

Installation of the avr tool chain

The "Atmel AVR Toolchain for Linux" can be downloaded from Atmel at:

http://www.atmel.com/tools/ATMELAVRTOOLCHAINFORLINUX.aspx.

Atmel wants you to fill out a registration form and after that they allow you to download it. The software is free open source software. A link to copies of those images is available at the end of this article. This is to ensure that the article remains valid even if Atmel should change the software in future.

We will be installing the whole thing at /usr/local/avr8-gnu-toolchain-VERSION with the softlink /usr/local/avr pointing to the most recent version. /usr/local/avr/bin would then be the directory where the compiler is and you would add that to your PATH environment:

in case of bash: PATH=/usr/local/avr/bin:${PATH} export PATH in case of tcsh: setenv PATH /usr/local /avr/bin:${PATH}
You don't have to do it that way with the softlink but it gives you an easy way to switch between versions if you need to.

The example given here uses the tar-ball named avr8-gnu-toolchain-3.2.3.314-linux.any.x86.tar.gz

```
Download the avr8-gnu-toolchain-3.2.3.314-linux.any.x86.tar.gz from Atmel
 and save it in /tmp

Install it like this:
 cd /usr/local/
 mkdir avr8-gnu-toolchain-3.2.3.314
 cd avr8-gnu-toolchain-3.2.3.314
 tar zxvf /tmp/avr8-gnu-toolchain-3.2.3.314-linux.any.x86.tar.gz

Create the softlink:
 ln -s avr8-gnu-toolchain-3.2.3.314/avr8-gnu-toolchain-linux_x86 avr
Test the link:
 cd /usr/local/avr
```

This was easy. The installation is done. You can use now use the avr-gcc compiler. The only thing that is missing from the tar-ball that Atmel provides is the programmer software avrdude. You would need to install it in addition. Installing avrdude is really easy:

Download avrdude from http://savannah.nongnu.org/projects/avrdude and save it in /tmp

The following example uses avrdude-5.11.tar.gz

```
tar zxvf /tmp/avrdude-5.11.tar.gz
cd avrdude-5.11/
CC=gcc
export CC
./configure --prefix=/usr/local/avr8-gnu-toolchain-3.2.3.314/avr8-gnu-toolchain-linux_x86
make

To install the compiled avrdude software now into the directory
/usr/local/avr8-gnu-toolchain-3.2.3.314/avr8-gnu-toolchain-linux_x86
you just type:

make install
```

For use with avrusb500 you should set /dev/ttyUSB0 as a default serial device in the avrdude.conf. Edit the file /usr/local/avr8-gnu-toolchain-3.2.3.314/avr8-gnu-toolchain-linux_x86/etc/avrdude.conf as follows:

```
#default_serial     = "/dev/ttyS0";
default_serial     = "/dev/ttyUSB0"; # avrusb500
```

That is: comment out the existing default_serial line and add the one shown above.

That's it! You are good to go.

This is really all you need in terms of toolchain for avr-microcontroller programming. All the Linux distributions come already with good text editors.

Example projects from tuxgraphics do all include makefiles. To compile such an example you would first read and follow the instructions in the included readme.htm file and then just type:

```
make
```

to compile it. "make load" would be used to compile and load it into the microcontroller using an avrusb500 programmer.

Functions that are not just plain C but specific to the avr-microcontroller are documented in the Modules index of the avr-libc user manual
The documentation that comes with the library is very good it is worth while to look at it if you want to do some serious programming.

References/Download

- The old article Building your own avr-gcc environment with atmega328p capabilities, Linux
- Tuxgraphics electronics section: http://electronics.tuxgraphics.org
- Download exact version of the avrtoolchain software that this document was written for: http://images.tuxgraphics.org/avrtoolchain-3.2.3/

2012-07-11, generated by tuxgrparser version 2.57