

# Programing AVR on Ubuntu with USBasp for beginners

Hiran Jayaweera / August 17, 2014

Microcontroller is a small chip on a computer. AVR is one of the microcontroller families developed by [Atmel](#). One main reason for getting such a fame for AVR is the availability of a large number of free and open source development tools compared to other similar microccontroller families. [Arduino](#), the most famous open hardware platform among hobbies, also use AVR as its processor (in most of their boards).

Followings are some of the tutorials written in this site about Arduino

This tutorial is about how to program an AVR using USBasp programmer on Ubuntu (I am using 14.04.1 now).

## Requirements

1. PC running on Ubuntu (14.04.1)
2. USBasp programmer
3. Atmel ATmega 32A chip connected to the USBasp programmer properly and some LEDs connected to it for verifying ([This is a guide on how to connect USBasp to an AVR](#))

## Installing software packages

In order to start programing AVR you require some tools such as a compiler, libraries, programmer (downloader), linker, assembler, e



- gcc-avr (compiller)
- binutils-avr (set of tools such as assembler, linker)
- [avr-libc](#) (C libraries specifically made for avr)
- [avrdude](#) (software programmer)

In the terminal window, run this command

```
sudo apt-get install gcc-avr avrdude avr-libc binutils-avr
```

In case of fedora

```
dnf install avr-gcc avr-binutils avr-libc avr-gdb avrdude
```

## Writing your first for AVR program on Ubuntu

Open your favorite editor and type following code, and save it as led.c

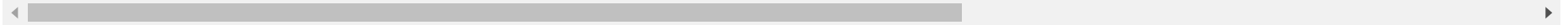
[?](#)

```
1      #define F_CPU 1000000UL
2
3
4      #include <avr/io.h>
5      #include <util/delay.h>
6
7      int main(void) {
8
9
10         DDRC = 255;
11
12         while(1){
13
14             PORTC=255;
15             _delay_ms(200);
16
17             PORTC=0;
18             _delay_ms(200);
19         }
20
21         return 0;
22     }
```

---

## Creating a Makefile

It is a good idea to have Makefile for compiling the c code. Download and extract [Makefile](#) file and use the Makefile available inside the archive. This make file was generated MFILE tool under windows 7. Copy Makefile and run `make all`



You can use "make", "make all", "make clean" and "make program" commands

## Downloading the firmware

We use avrdude for downloading the generated hex file to the device. Connect your USBasp programmer to the computer and the microcontroller.

```
sudo avrdude -p m32 -c usbasp -e -U flash:w:led.hex
```

in above code -p followed by m32 is for part number ATmega 32A; -c is for providing the programmer's name (hardware programmer); -e is for erasing the device before programming.

Refer [avrdude Linux man](#) page for more information



August 17, 2014 in Misc, Tutorials.