

## Bootloader

The common method to program a microcontroller is to use a programmer for that particular microcontroller. An alternative is to write a small program (a bootloader) into the flash memory of the microcontroller which allows code and EEPROM data to be transmitted over a serial cable and written to the microcontroller.

A bootloader has to be written to the flash memory just once using a conventional programmer. The bootloader is programmed such that when the bootloader start condition is satisfied it receives data via a predetermined interface (eg, UART) and writes these into the program memory at predetermined locations.

We would be using the AVR Butterfly Bootloader ([click here](#) to download). For programming using the Bootloader we would be using AVRdude which is part of the WinAVR package.

Instructions for setting up the main.c and makefile are also given in the readme file.

1. Select the MCU type to be used.
2. Keep the bootloader size at its default 512 words.
3. Set the baudrate making sure it's the same as the one set for the PORT used.
4. Set the CPU frequency for eg, `#define F_CPU 8000000` for 8MHz
5. Now select the startup mode.
  - **START\_SIMPLE:** In this mode the MCU checks the start condition on reset. The start condition is a given pin being grounded. If the start condition is satisfied the MCU starts listening through UART and programs the flash according to commands received. The startpin can be set by the BLPORT, BLDDR, BLPIN and BLPNUM variables. Set the pin in BLPNUM variable and corresponding PORT, DDR and PIN values in BLPORT, BLDDR and BLPNUM variables correspondingly. If the startpin is not satisfied then the program already stored in flash memory is run.
  - **START\_WAIT:** In this mode the MCU waits for a specified time for incoming data. If data is received the flash memory is written according to commands received. After the timeout, the program already stored in flash memory is run. The timeout value can be set by the WAIT\_VALUE variable (the variable stores the number of 10 ms steps)
6. Now make the file by running Command Prompt. Go to the folder and execute make command. The HEX file would be generated. If some changes are to be made in the source

file after generating HEX file first run make clean which deletes the present HEX files and then run make.

7. Program the bootloader to the MCU. Program the "Boot Flash section size" (BOOTSZ fuses) according to the boot-size selected in the makefile i.e. BOOTSZ=00 for boot-size 1024 words on ATmega16. Enable the Boot Reset Vector fuse ie BOTRST=0.
8. Now reset the MCU, fulfilling the start condition.
9. Start avrdude.

### AVRDude interface:

Type avrdude -h to get a list of commands and options.

To program the MCU type the following command:

```
avrdude -p <mcu_type> -c butterfly -P <com_port> -b <baud_rate> -U  
<memory_type>:w:<filename>.hex
```

<mcu\_type>

m8 for ATMEGA8, m16 for ATMEGA16 and so on.

<memory\_type>

Use flash for programming to the flash memory eeprom for writing the EEPROM memory.

Remember to set the Baud rate if non- standard RS-232 baud rate is being used or AVRDUDE will be unable to connect.

To run the program reset the MCU. If START\_SIMPLE is being used now set the startpin which was grounded earlier to live. This passes the control to the application flash section and the program loaded in the application flash memory is run. If START\_WAIT mode is being used the MCU automatically runs the program stored in application flash memory.

Abhishek Srivastava

Kapil Dev Advani

Sakaar Khurana

Vatsal Sharan