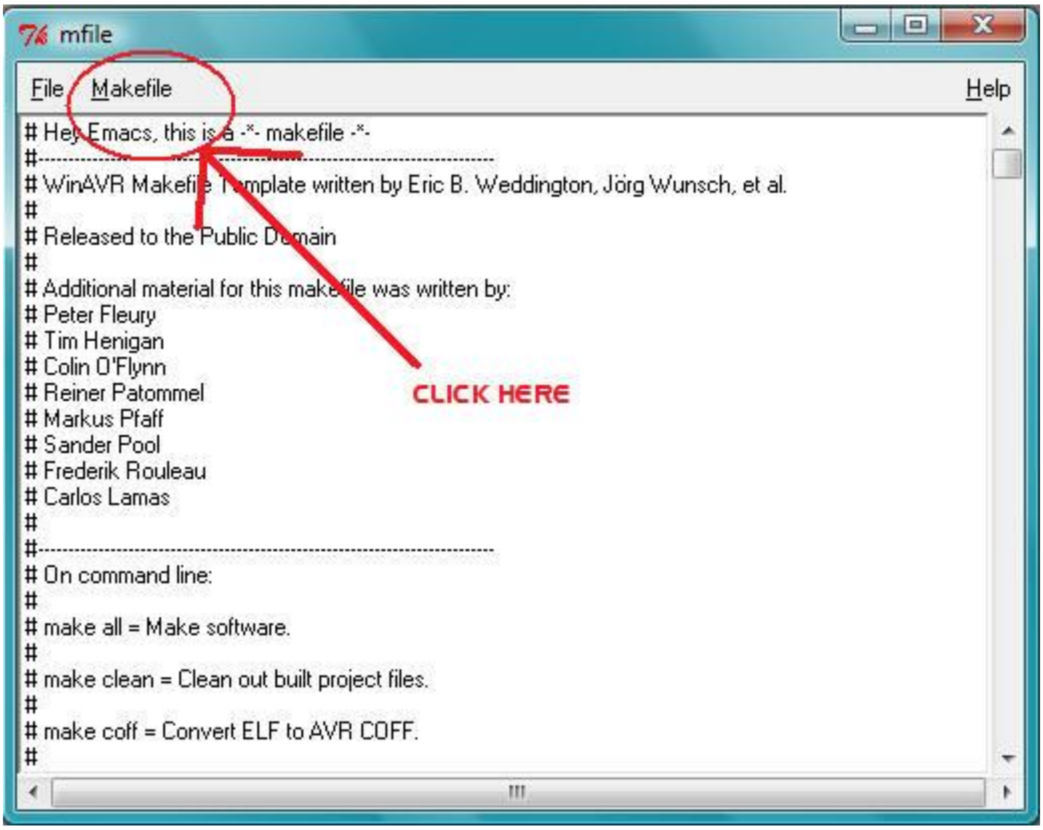
Tutorial by: Sohail Alam
www.robozaa.co.cc

Using WinAVR - Writing the first program

Let us assume that you have installed WinAVR-20090213 (2009 version) on your C: drive. The default install folder is C:\WinAVR-20090313. Let us also assume that you are using Microsoft Windows (any version) as your OS.

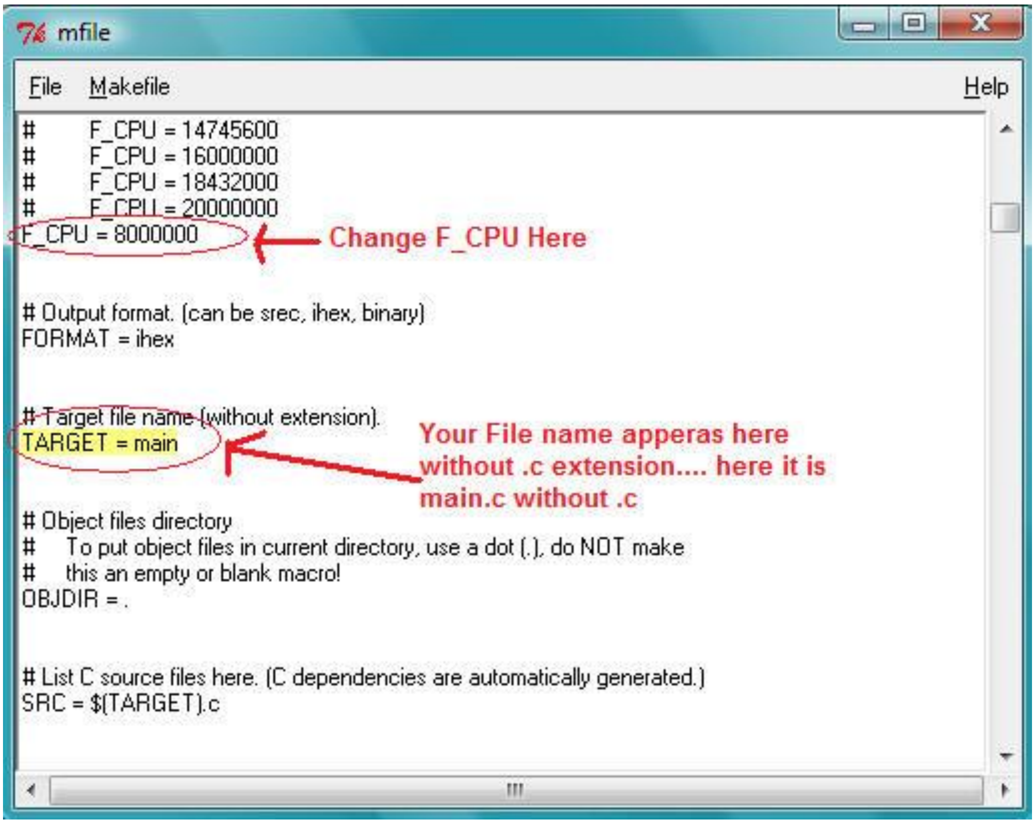
Let us now take a look at how to begin writing codes for our microcontroller.

- The very first step should be creating a folder (by the name say **AVR**) at say **Desktop**. This is the place where you shall save your works. Please create separate folders for each program you write for eg. make a folder led_blink.
- Next open **Programmers Notepad [WinAVR]** from **START -> Programs -> WinAVR-20090313**.
- Save the new file in your above folder (Desktop/AVR/led_blink) by any desired name **with .c as file extension**. As an example save as led.c
- Next open **MFile** (Make File) from **START -> Programs -> WinAVR-20090313**. *Be very careful in the next few steps.*
- Click MakeFile (from menu) -> Enable Editing MakeFile (Last Option).



```
# Hey Emacs, this is a -*- makefile -*-
#----------------------------------------------------------------------------
# WinAVR Makefile Template written by Eric B. Weddington, Jörg Wunsch, et al.
#
# Released to the Public Domain
#
# Additional material for this makefile was written by:
# Peter Fleury
# Tim Henigan
# Colin O'Flynn
# Reiner Patommel
# Markus Pfaff
# Sander Pool
# Frederik Rouleau
# Carlos Lamas
#
#----------------------------------------------------------------------------
# On command line:
#
# make all = Make software.
#
# make clean = Clean out built project files.
#
# make coff = Convert ELF to AVR COFF.
#
```

- Click MakeFile (from menu) -> main file name -> give the **same name** as above **without .c extension**. As for the above example name it as led -> Click OK
- Now scroll a little above to the section F_CPU = 8000000 -> Make the desired change be editing it. If you are using a 16MHz crystal then F_CPU should be equal to 16000000.

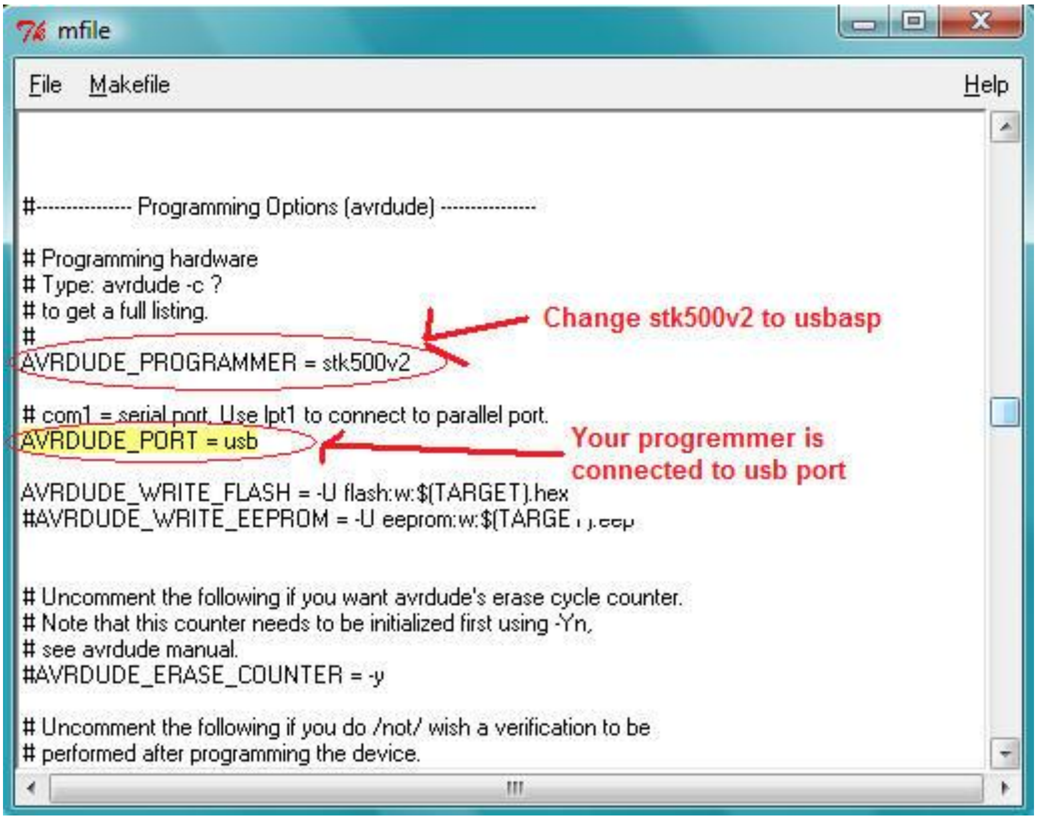


```
#       F_CPU = 14745600
#       F_CPU = 16000000
#       F_CPU = 18432000
#       F_CPU = 20000000
F_CPU = 8000000


# Output format. (can be srec, ihex, binary)
FORMAT = ihex


# Target file name (without extension).
TARGET = main


# Object files directory
#     To put object files in current directory, use a dot (.), do NOT make
#     this an empty or blank macro!
OBJDIR = .


# List C source files here. (C dependencies are automatically generated.)
SRC = $(TARGET).c
```

Change F_CPU Here

Your File name apperas here without .c extension.... here it is main.c without .c

- Click MakeFile (from menu) -> MCU type -> ATmega -> atmega16 (If you are using ATmega16).
- Click MakeFile (from menu) -> Port -> USB (if you are using a usb type of programmer) else choose the COM port where you have connected the programmer.
- Scroll to the section **AVRDUDE_PROGRAMMER = stk500v2.** Change **stk500v2** by **usbasp** if you are using a USBasp programmer. It is the most commonly used programmer. Hence it looks like **AVRDUDE_PROGRAMMER = usbasp**



```
#---------------- Programming Options (avrdude) ----------------

# Programming hardware
# Type: avrdude -c ?
# to get a full listing.
#
AVRDUDE_PROGRAMMER = stk500v2

# com1 = serial port. Use lpt1 to connect to parallel port.
AVRDUDE_PORT = usb

AVRDUDE_WRITE_FLASH = -U flash:w:$(TARGET).hex
#AVRDUDE_WRITE_EEPROM = -U eeprom:w:$(TARGET).eep


# Uncomment the following if you want avrdude's erase cycle counter.
# Note that this counter needs to be initialized first using -Yn,
# see avrdude manual.
#AVRDUDE_ERASE_COUNTER = -y

# Uncomment the following if you do /not/ wish a verification to be
# performed after programming the device.
```

Change stk500v2 to usbasp

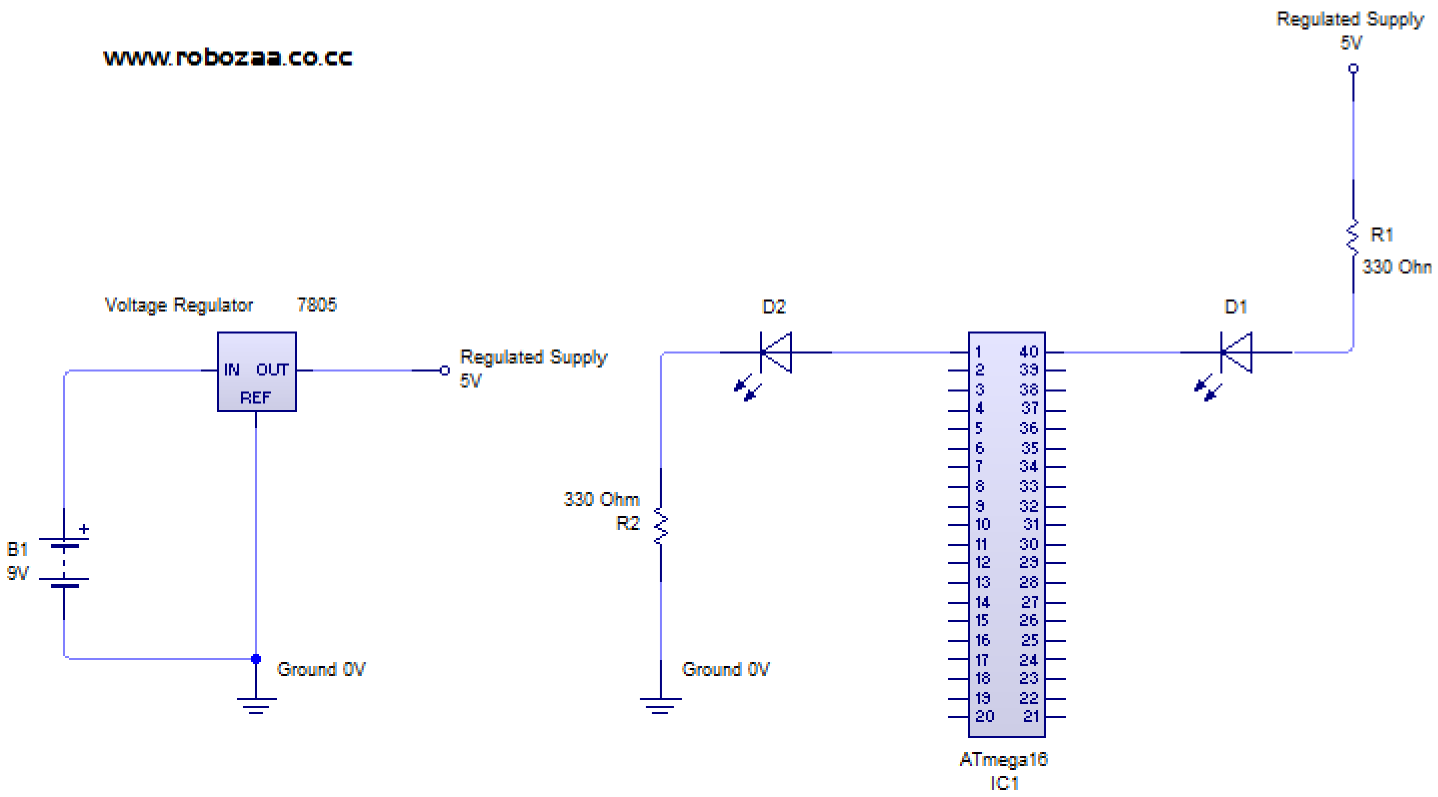
Your programmer is connected to usb port

- Click MakeFile (from menu) -> Programmer (if you are NOT using a USBasp programmer) and choose your programmer.
- Now the last step -> file -> save as -> browse to the folder where you have saved the file led.c (Desktop/AVR/led_blink) then save it. Make sure the file name is not changed, it should be named as **Makefile**
- Now you are all done to write your code and compile it and even burn it to your MCU with the help of WinAVR.

Please refer the following if you haven't already, before proceeding.

1. Basics of MCU
2. Ports and Registers

Now we write our code for blinking some LEDs. The circuit diagram is given as :



Explanation of the above circuit:

There are two LEDs D1 and D2. D1 connected to PA0 (Port A Pin 0) and Vcc (+5V) through a resistor. Similarly D2 connected to PB0 (Port B Pin 0) but grounded (0V) through a resistor.

It is important to know when will these two LEDs will glow.

Both will glow ONLY when Port A and B, Pin 0 are made OUTPUT. Also remember when a port pin is made output two conditions may arise: pin may be made high (+5V) or low (0V) as and when required.

D1 will glow only when it receives LOW (0V) from PA0. Similarly D2 will glow only when it receives HIGH (+5V) from PB0.

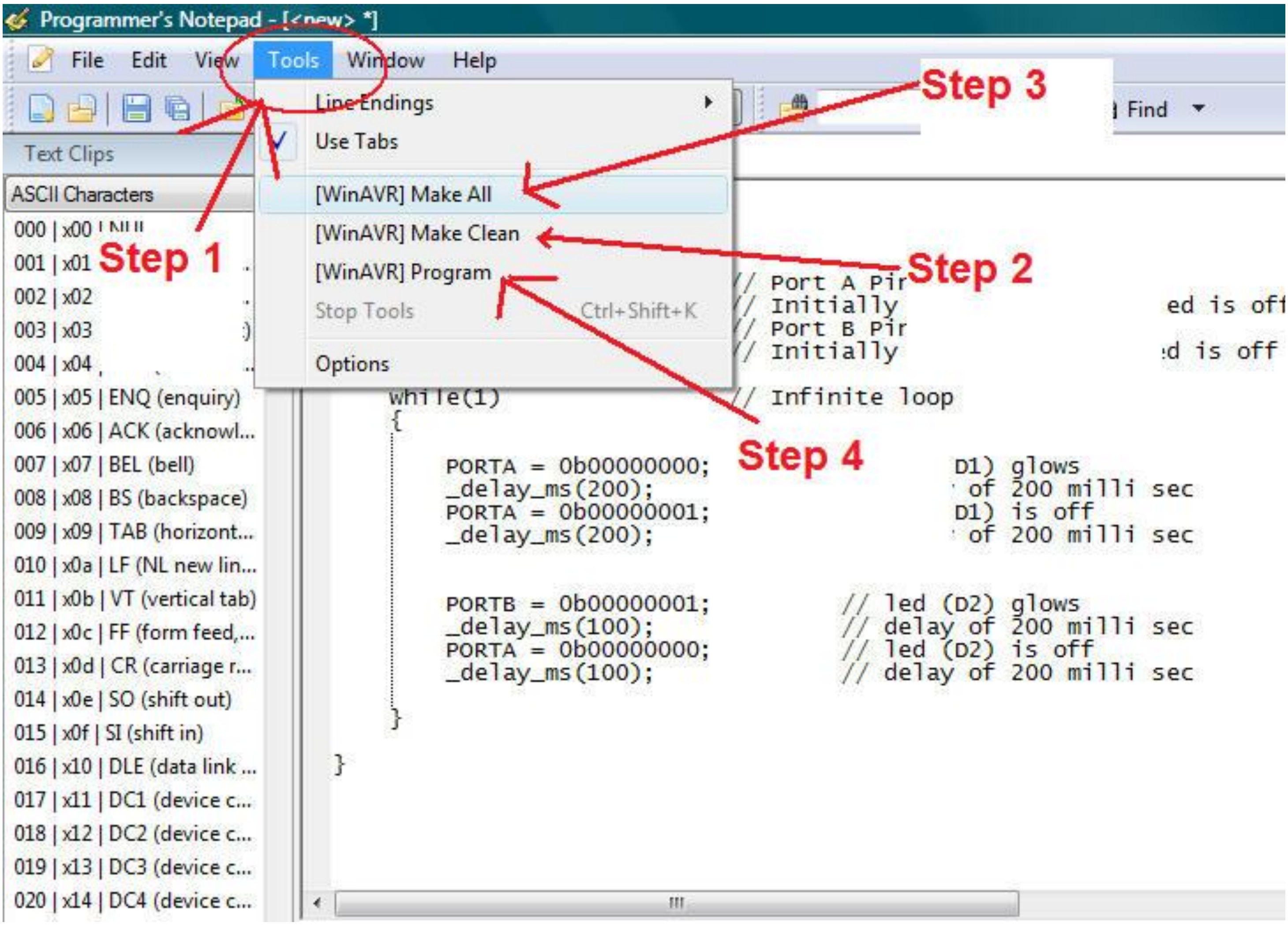
Now we write the code in Programmer's Notepad [WinAVR].

```
#include <avr/io.h>          // This contains the definitions of the terms used
#include <util/delay.h>      // This contains the definition of delay function


void main()
{
DDRA = 0b00000001; // Port A Pin 0 made output
PORTA = 0xff; // Initially Pin 0 is high so led is off
DDRB= 0b00000001; // Port B Pin 0 made output
PORTB= 0; // Initially Pin 0 is low so led is off

while(1) // Infinite loop
{
```

```
PORTA = 0b00000000; // led (D1) glows
_delay_ms(200); // delay of 200 millisecond
PORTA = 0b00000001; // led (D1) is off
_delay_ms(200); // delay of 200 millisecond


PORTB = 0b00000001; // led (D2) glows
_delay_ms(100); // delay of 200 millisecond
PORTA = 0b00000000; // led (D2) is off
_delay_ms(100); // delay of 200 millisecond

}

}
```

Now we are ready to compile and burn our program to the mcu.

Follow the steps shown in the picture. Make sure your programmer (usbasp) is connected to the PC or Laptop and also to your MCU and also make sure that the power (Vcc) of your MCU is ON before proceeding to Step 4.


Programmer's Notepad - [<new> *]
File
Edit
View
Tools
Window
Help
Find
Text Clips
ASCII Characters
000 | x00 | NULL
001 | x01
002 | x02
003 | x03
004 | x04
005 | x05 | ENQ (enquiry)
006 | x06 | ACK (acknowl...
007 | x07 | BEL (bell)
008 | x08 | BS (backspace)
009 | x09 | TAB (horizont...
010 | x0a | LF (NL new lin...
011 | x0b | VT (vertical tab)
012 | x0c | FF (form feed,...
013 | x0d | CR (carriage r...
014 | x0e | SO (shift out)
015 | x0f | SI (shift in)
016 | x10 | DLE (data link ...
017 | x11 | DC1 (device c...
018 | x12 | DC2 (device c...
019 | x13 | DC3 (device c...
020 | x14 | DC4 (device c...
Line Endings
Use Tabs
[WinAVR] Make All
[WinAVR] Make Clean
[WinAVR] Program
Stop Tools
Ctrl+Shift+K
Options
Step 1
Step 2
Step 3
Step 4

```
        // Port A Pir                                        ed is off
        // Initially
        // Port B Pir
        // Initially                                         :d is off
    while(1)                // Infinite loop
    {

        PORTA = 0b00000000;                     D1) glows
        _delay_ms(200);                       ' of 200 milli sec
        PORTA = 0b00000001;                     D1) is off
        _delay_ms(200);                       ' of 200 milli sec


        PORTB = 0b00000001;             // led (D2) glows
        _delay_ms(100);                 // delay of 200 milli sec
        PORTA = 0b00000000;             // led (D2) is off
        _delay_ms(100);                 // delay of 200 milli sec

    }

}
```

Hoooollaaa.... you are done.