

# AVRUSBPRG1

## USB AVR IN SYSTEM PROGRAMMER



2016



## Table of Contents

### **1- Introduction**

**1.1- AVRUSBPRG1 Features**   **1.2- Supported Devices**

**1.3- System Requirement**   **1.4- Pinout**

### **2- AVRUSBPRG1 Driver Installation**

### **3- Running the Software**

### **4- Sample AVR Test Application**

**4.1- Hardware**   **4.2- Software**   **4.3- Downloading**

### **5- Important Note about Fuse Settings**

## **1. Introduction**

**The AVRUSBPRG1 can program all ATMEL AVR 8-bit RISC microcontrollers with ISP Interface via the PC computer USB port.**

### **1.1 AVRUSBPRG Features**

- **Supports all AVR devices with ISP interface.**
- **Very compact size - 3cm x 2cm.**
- **Uses the PC USB port for programming.**
- **Comes with built in USB cable.**
- **Drivers compatible with Windows XP, VISTA, 7, 8, 10.**
- **Programs and verifies both flash and EEPROM.**
- **Supports fuses and lock bit programming and verification.**
- **Powered from USB, does not require external power supply.**
- **Can supply power to the target microcontroller board.**
- **Built in self release protection for short circuit.**

## 1.2 Supported Devices

**Classical AVR**s : AT90S1200 - AT90S2313 - AT90S2333 - AT90S2343 - AT90S4414 - AT90S4434 - AT90S8515 - AT90S8535 -

**ATMEGA AVR**s : ATMEGA103 - ATMEGA128 - ATMEGA1280 - ATMEGA1281 - ATMEGA1284 - ATMEGA128RF1 - ATMEGA16 - ATMEGA161 - ATMEGA162 - ATMEGA163 - ATMEGA164P - ATMEGA168 - ATMEGA169 - ATMEGA2560 - ATMEGA2561 - ATMEGA32 - ATMEGA324P - ATMEGA3250 - ATMEGA328P - ATMEGA329 - ATMEGA329P - ATMEGA48 - ATMEGA64 - ATMEGA640 - ATMEGA644 - ATMEGA644P - ATMEGA645 - ATMEGA6450 - ATMEGA649 - ATMEGA6490 - ATMEGA8 - ATMEGA8515 - ATMEGA8535 - ATMEGA88

**ATtiny AVR**s : ATtiny11 - ATtiny12 - ATtiny13 - ATtiny15 - ATtiny2313 - ATtiny24 - ATtiny25 - ATtiny26 - ATtiny261 - ATtiny44 - ATtiny45 - ATtiny461 - ATtiny84 - ATtiny85 - ATtiny861 - ATtiny88

**CAN AVR**s : AT90CAN128 - AT90CAN32 - AT90CAN64

**PWM AVR**s : AT90PWM2 - AT90PWM2B - AT90PWM3 - AT90PWM3B

**USB AVR**s : AT90USB1286 - AT90USB1287 - AT90USB162 - AT90USB646 - AT90USB647 - AT90USB82 -

## 1.3 System Requirement

- Pentium (Pentium II and above is recommended)
- 64 MB RAM
- 150 MB Free Hard Disk Space for installation of AVR Studio 4.12 or later
- Windows® XP, VISTA, 7, 8, and 10
- USB port, self-powered

## 1.4 Pinout

**MOSI** pin of the target AVR

**MISO** pin of the target AVR

**SCK** pin of the target AVR

**SCK** pin of the target AVR

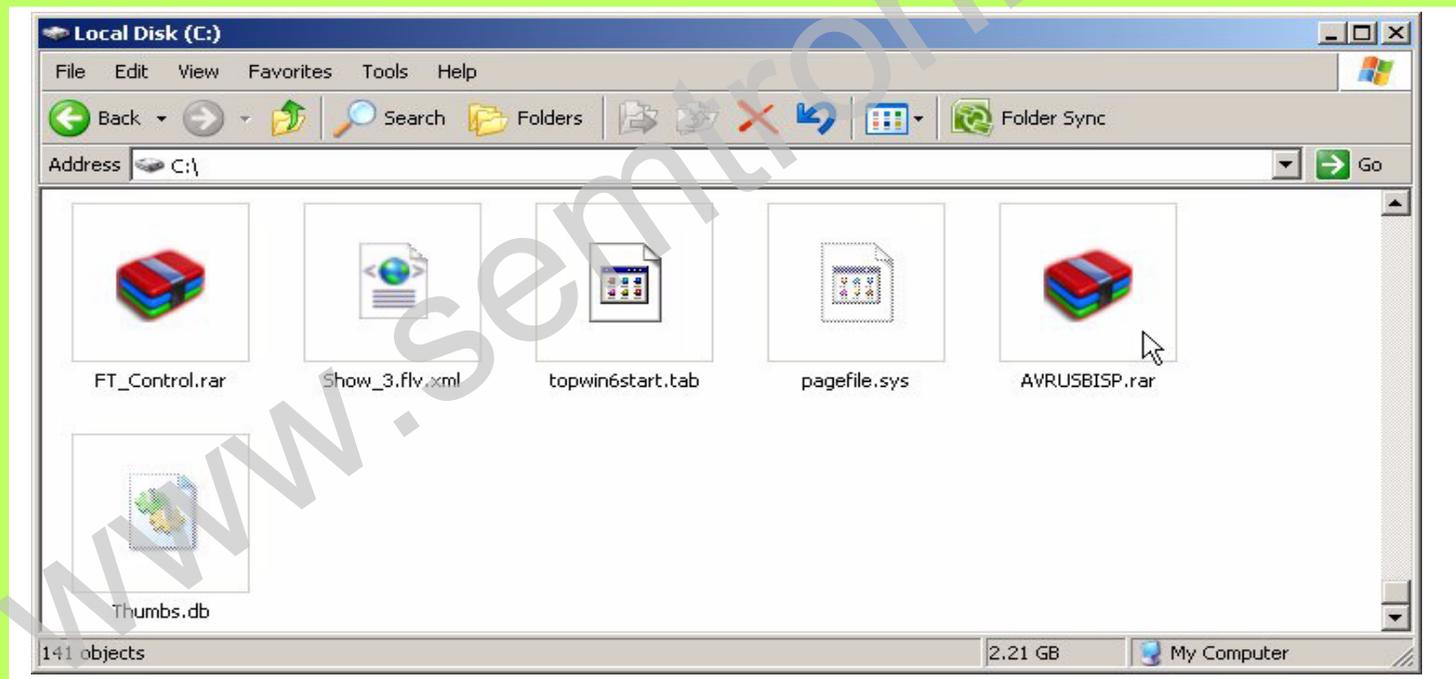
**GND** pin of the target AVR

**VCC** pin of the target AVR

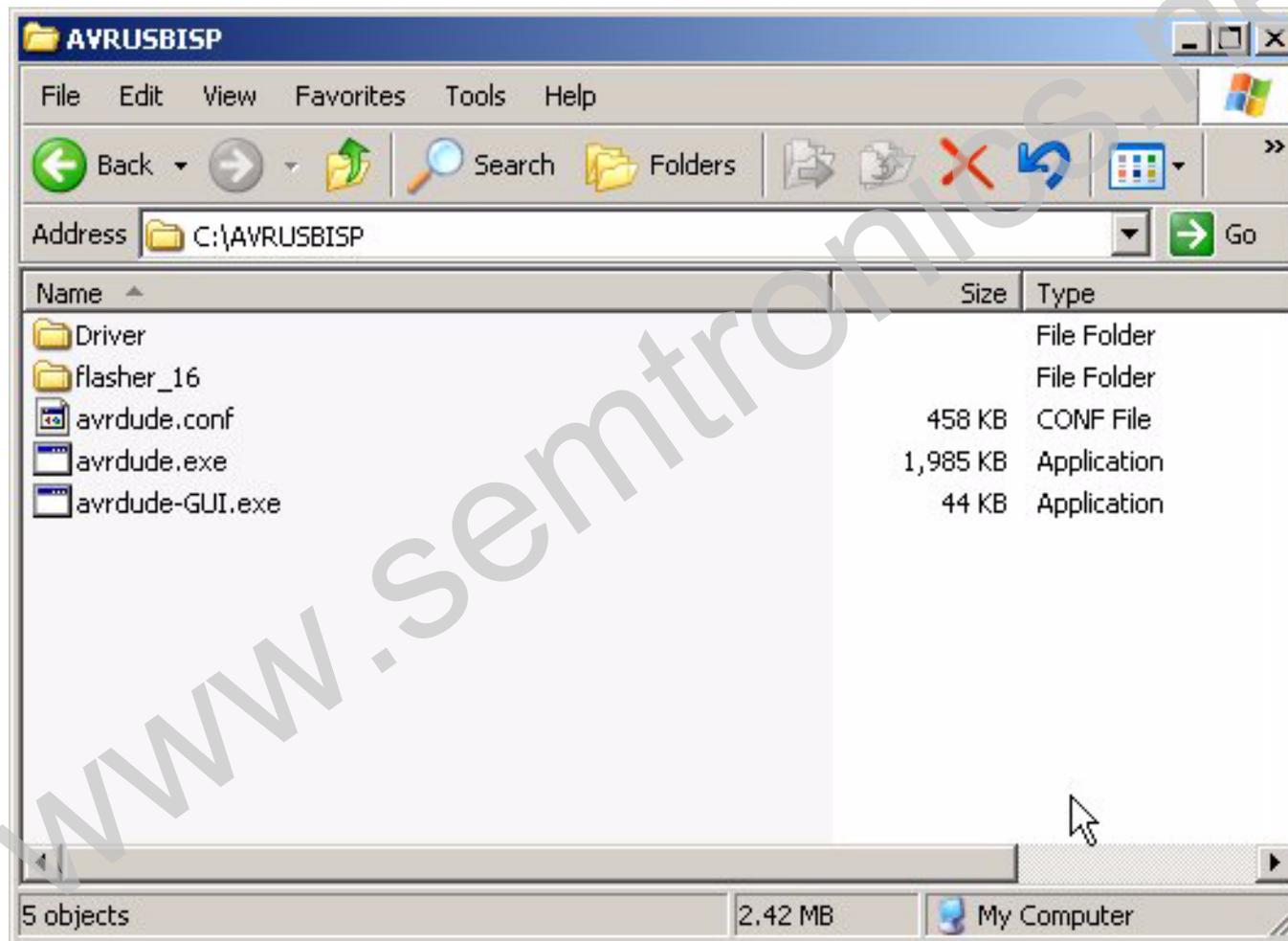


## 2- AVRUSBPRG1 Driver Installation

First, download or copy the file [AVRUSBISP.rar](#) (781kB) to your computer's hard disk.

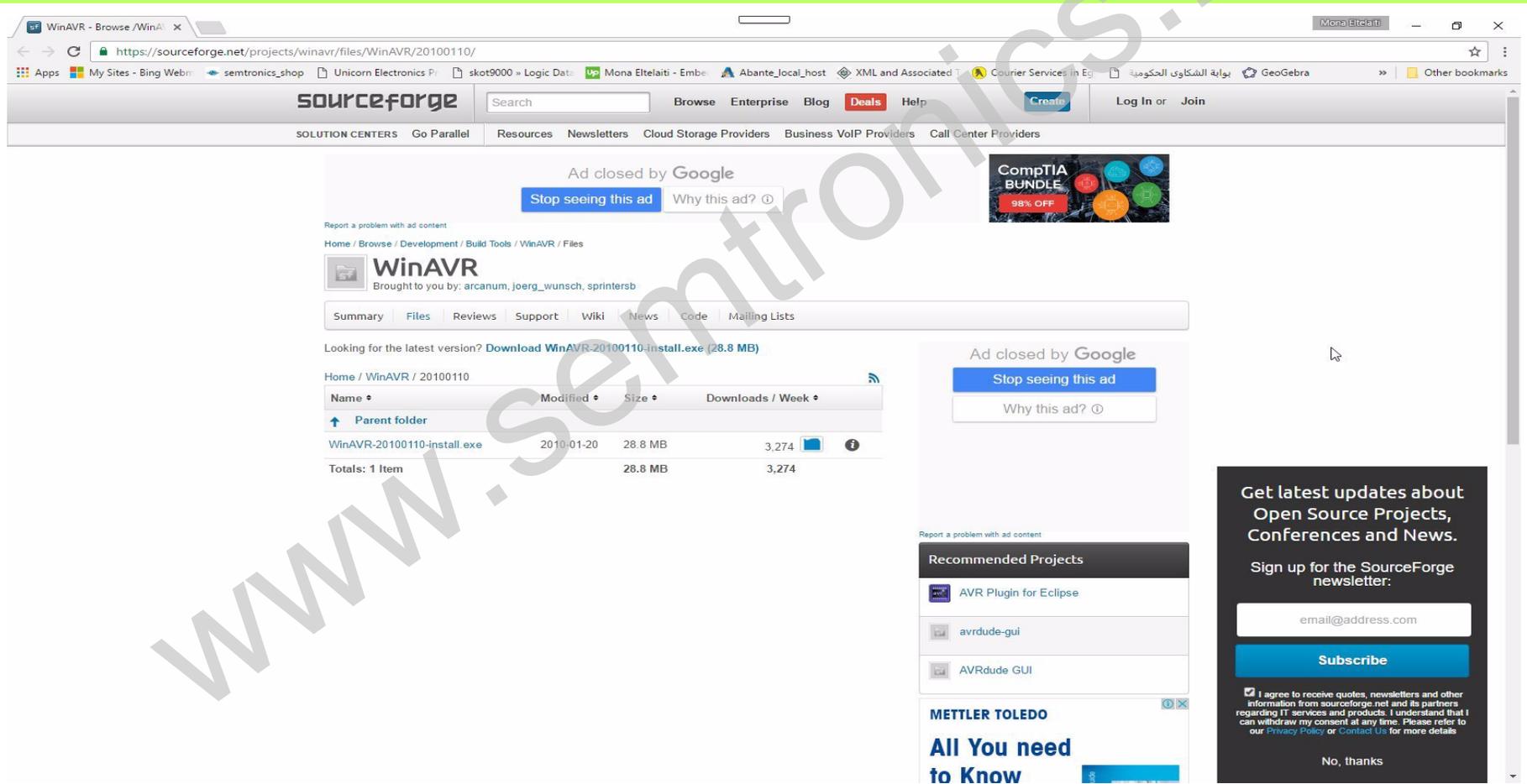


**Unrar the file.**



Download **Winavr** from the following link and copy it to the same folder of **avrusbisp**

<https://sourceforge.net/projects/winavr/files/WinAVR/20100110/>

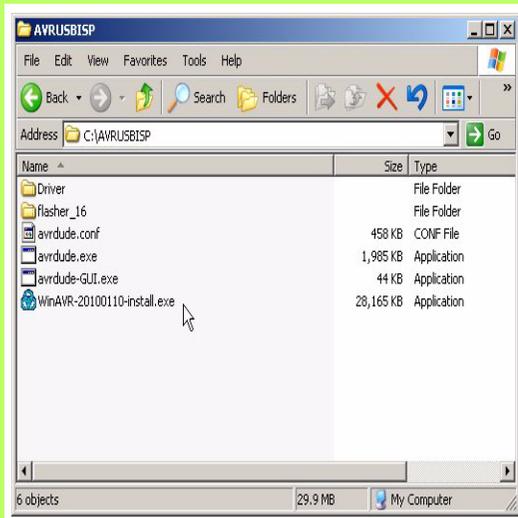


The screenshot shows a web browser window displaying the SourceForge project page for WinAVR. The browser's address bar shows the URL: <https://sourceforge.net/projects/winavr/files/WinAVR/20100110/>. The page header includes the SourceForge logo and navigation links such as "Browse", "Enterprise", "Blog", "Deals", and "Help". Below the header, there are several navigation tabs: "Summary", "Files", "Reviews", "Support", "Wiki", "News", "Code", and "Mailing Lists". The main content area features a table with the following data:

Name	Modified	Size	Downloads / Week
Parent folder			
<a href="#">WinAVR-20100110-install.exe</a>	2010-01-20	28.8 MB	3,274
Totals: 1 Item		28.8 MB	3,274

Below the table, there are several promotional banners and a newsletter sign-up form. The "Recommended Projects" section lists "AVR Plugin for Eclipse", "avrdude-gui", and "AVRdude GUI". The "Mettler Toledo" banner reads "All You need to Know". The newsletter sign-up form includes a text input field for an email address, a "Subscribe" button, and a checkbox for agreeing to receive updates.

# Install Winavr



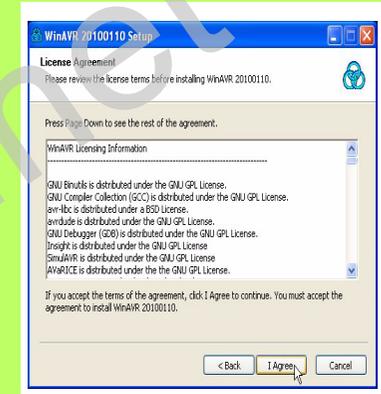
1



2



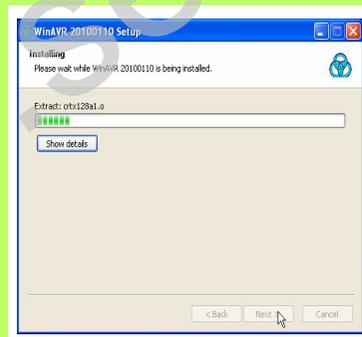
3



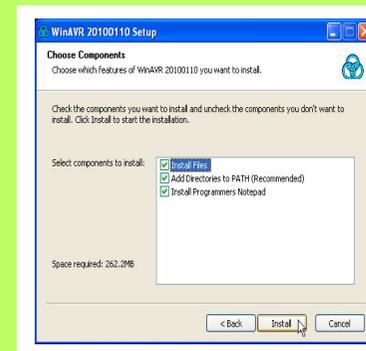
4



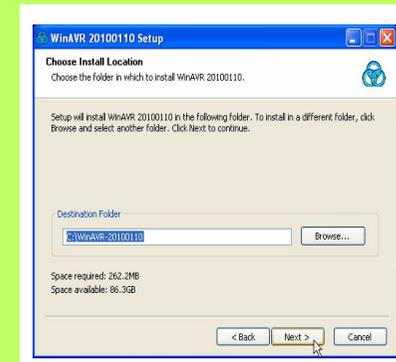
8



7

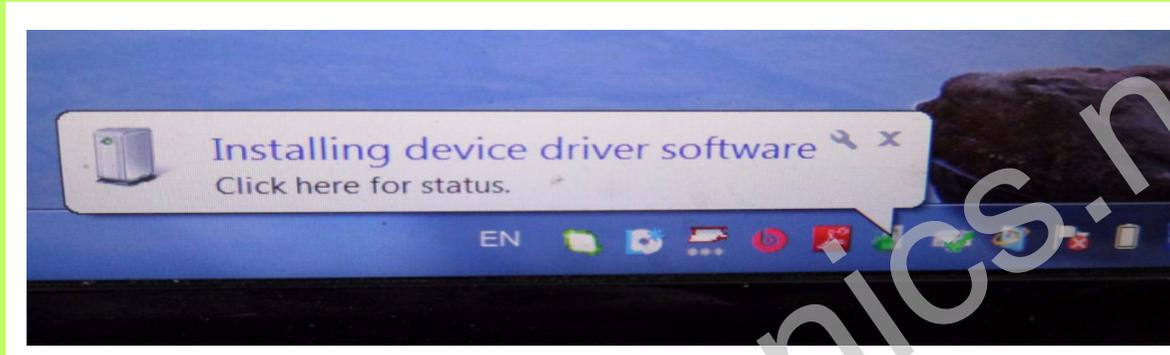


6



5

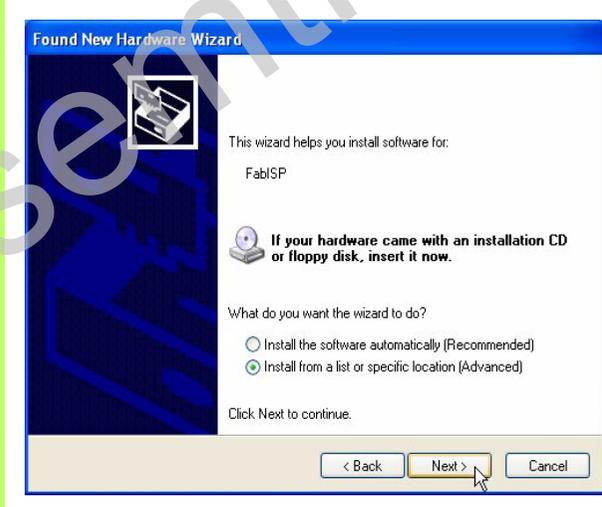
## Connect **AVRUSBPRG1** to the USB port and install the drivers



1



2



3

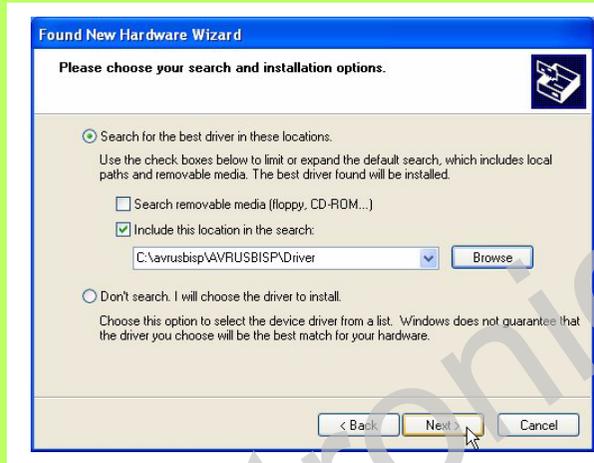


4

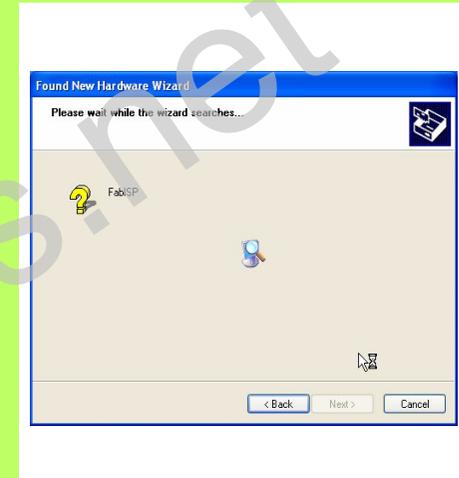
# Continue AVRUSBPRG1 USB driver installation



5



6



7



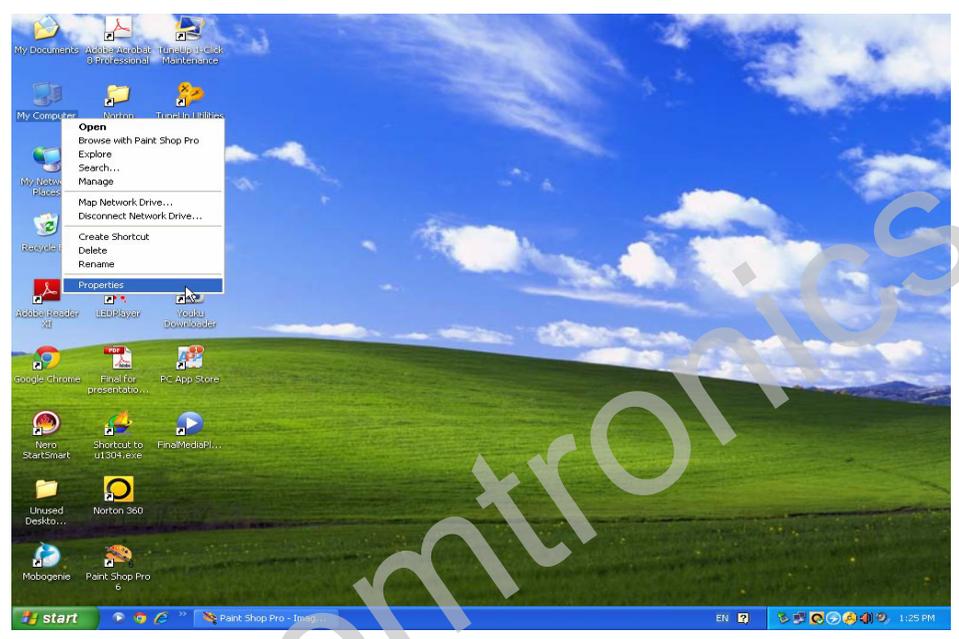
9



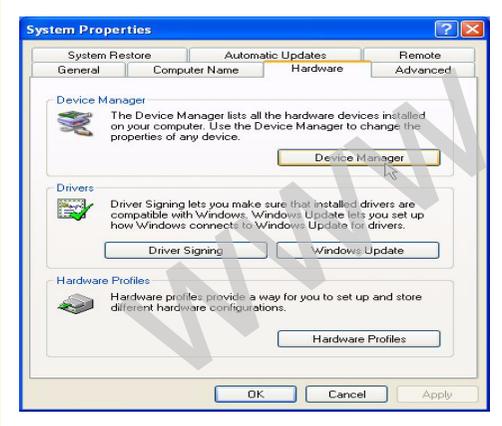
8



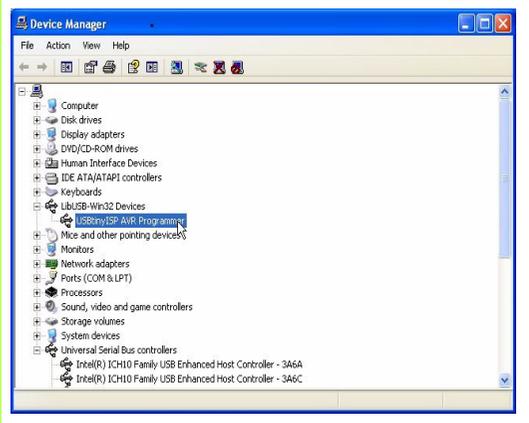
# Check that the driver has been installed successfully



1



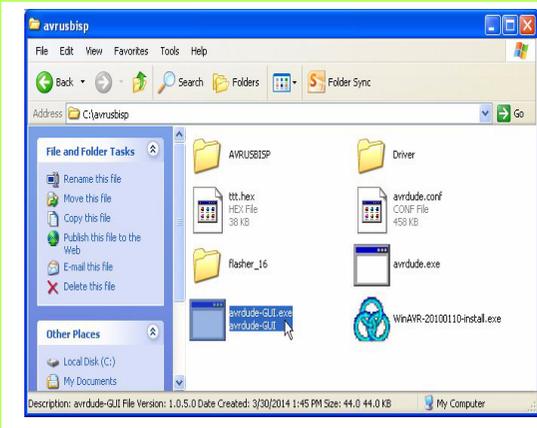
2



3

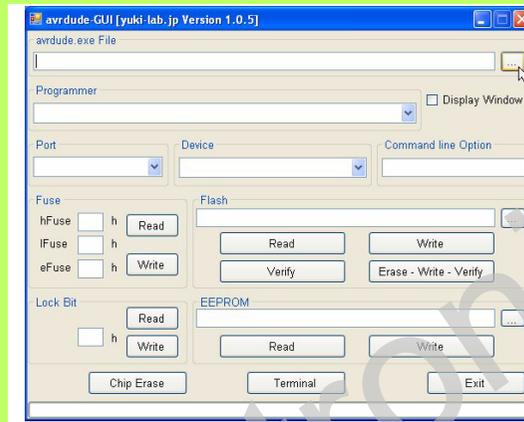
### 3- Running the Software

Run **avrdude-GUI.exe**

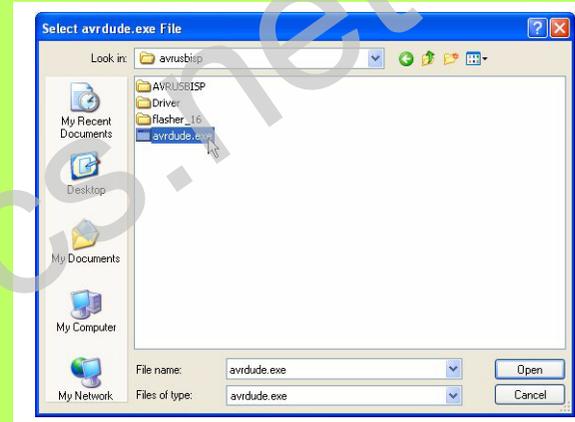


1

Select the path for **avrdude.exe**

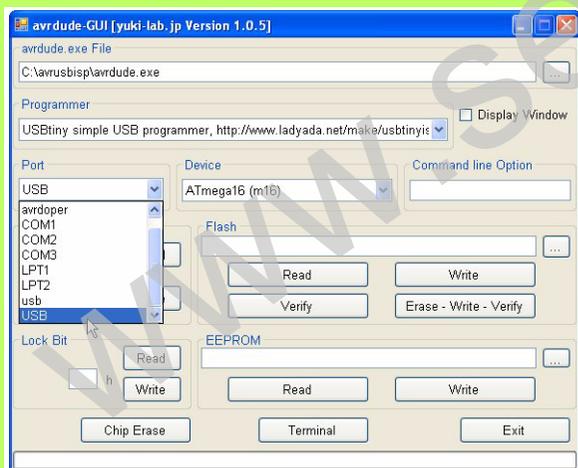


2



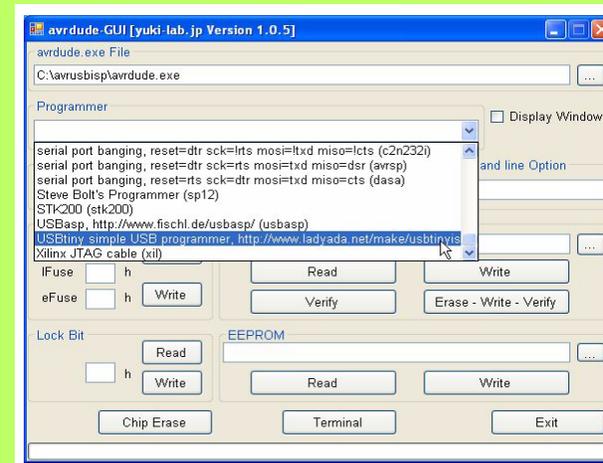
3

Select the **USB and Atmega16**



5

Select the programmer type



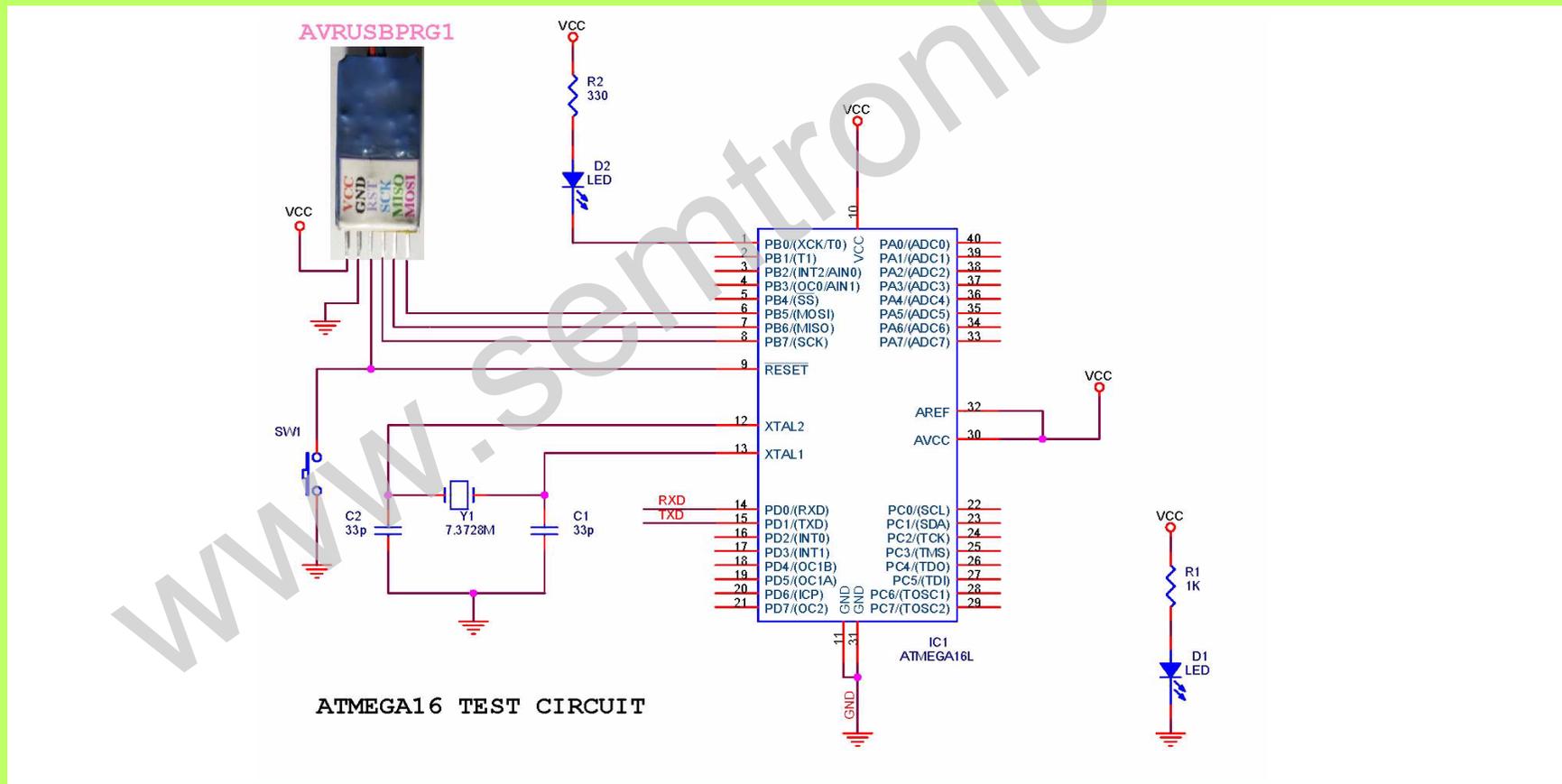
4

## 4. Sample AVR Test Application

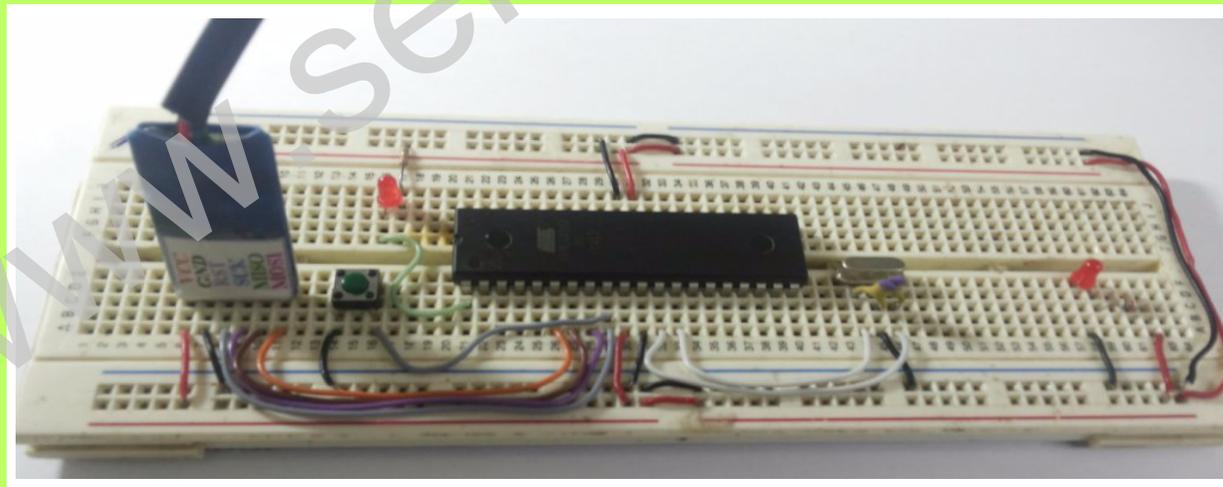
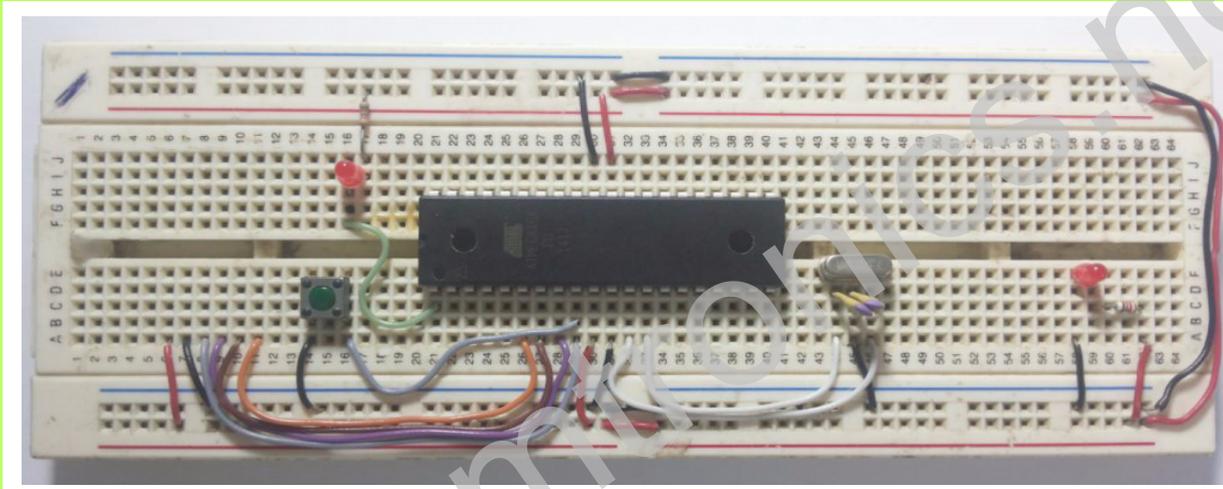
The following example illustrates the hardware and software requirements for building a sample flasher test application for ATMEGA16.

### 4.1 Hardware

The shown circuit illustrates the schematic for the sample flasher application



**The photos of the complete circuit connected in the breadboard are shown before connecting the AVRUSBPRG and after connecting it.**



## 4.2 Software

You can develop the program for the controller using any C compiler like the Atmel Studio, Codevision, MikroC, or IAR. The accompanied software comes with a complete Codevision project for ATMEGA16. The C code is listed below

```
/****** flasher_16.C *****/
// This program flashes a LED connected to PB.0 every 1/4 second
// ATMEGA16 microcontroller with 7.3728MHz crystal
#include <mega16.h> /* special function register declarations */

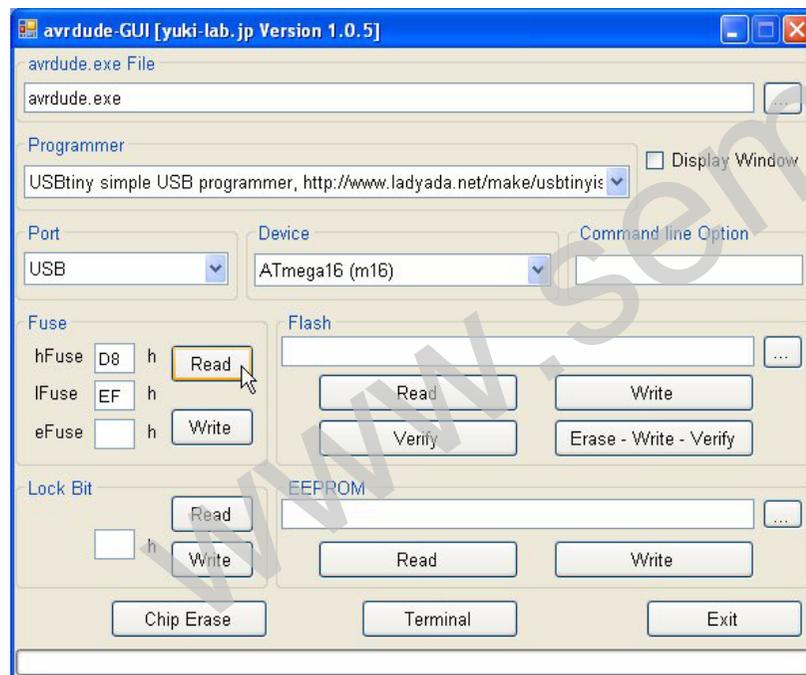
unsigned int i;
#define LED PORTB.0

void main (void)
{
    DDRB=0x01; //Program PORTB.0 as output
    while (1)
    {
        LED = 0; // Turn on the LED
        for(i=0;i<65000;i++); // Delay 1 S
        LED = 1; // Turn off the LED
        for(i=0;i<65000;i++); // Delay 1 S
    }
}
```

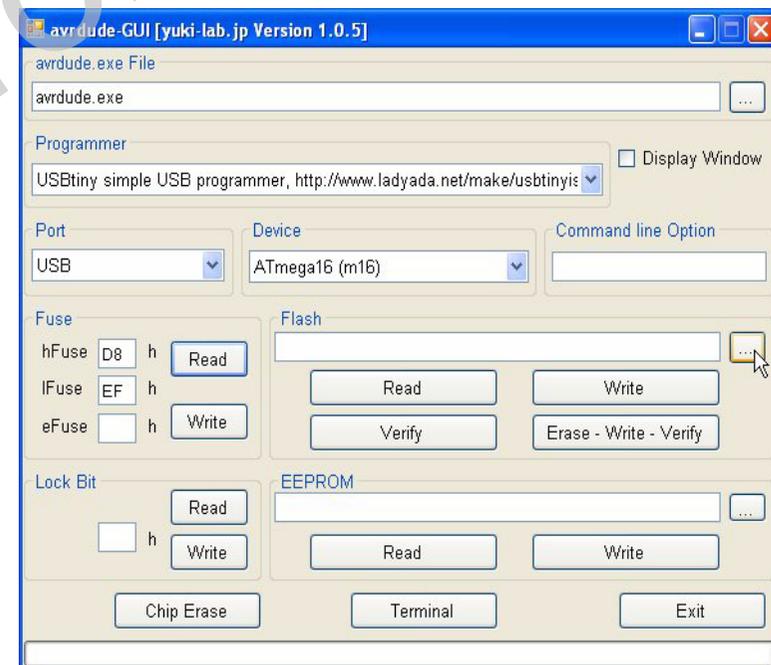
## 4.3 Downloading

After compiling the C, the resulting HEX file can be downloaded into the microcontroller using AVRUSBPRG1 software

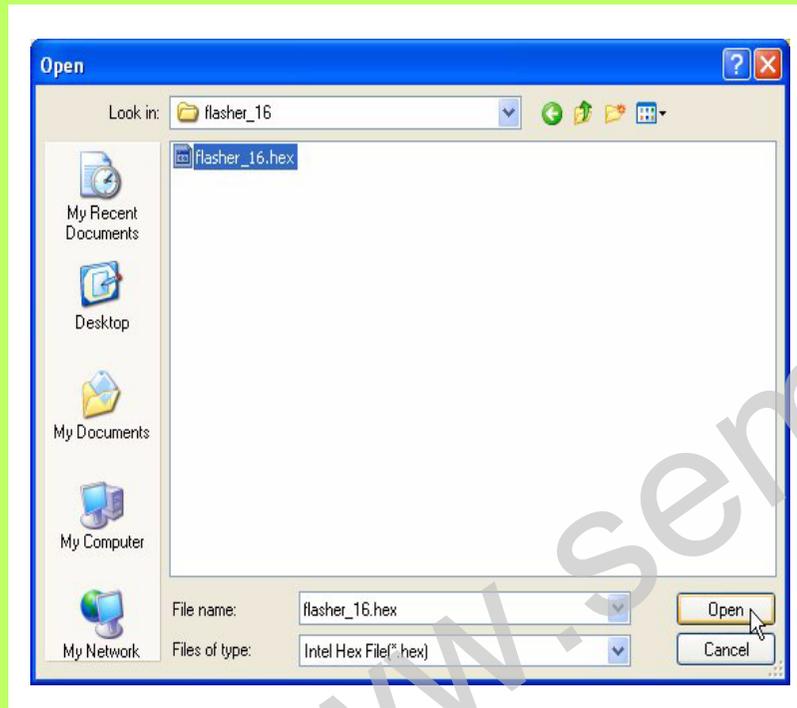
Click on the Fuse Read button to ensure successful recognition of the target microcontroller



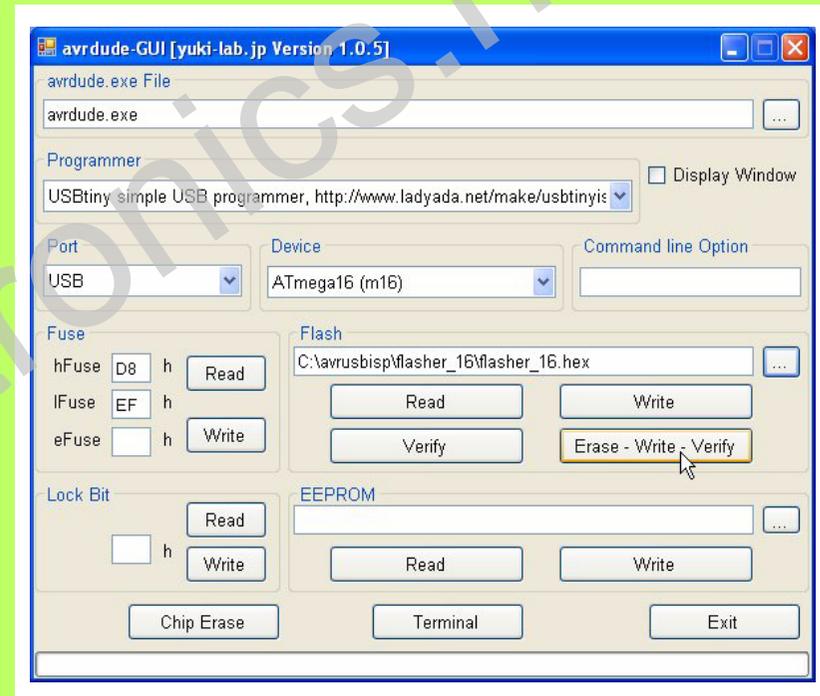
Click on the Flash open file button to load the .hex file



**Load the .hex file**



**Click on the Erase, Write, Verify button to program the code into the microcontroller**



**Now the LED connected to PORTB.0 will flash indicating the successful process.**

## **5- Important Note about Fuse Settings**

**It is worth to mention the importance of proper setting of the fuse and lock bits of the microcontroller. wrong settings may make the microcontroller undetectable by the software.**

**It is recommended to use an AVR calculator program to set the options and get the fuse values. One example is the calculator at the following link.**

**[eleccelerator/fusecalc/](http://www.eleccelerator.com/fusecalc/)**

**After calculating the fuse bits values, you can copy them into the [avrdude-gui](#) software and then program the fuses and lock bits.**

Select Chip: ATmega16 (current) Go

LOW Fuse Presets:

- Brown-out detection enabled; [BODEN=0]
- Brown-out detection level at VCC=2.7 V; [BODLEVEL=1]
- Ext. Crystal/Resonator High Freq.; Start-up time: 1K CK + 0 ms; [CKSEL=1110 SUT=10]

HIGH Fuse Presets:

- Boot Flash section size=128 words Boot start address=\$1F80; [BOOTSZ=11]
- Boot Reset vector Enabled (default address=\$0000); [BOOTRST=0]
- CKOPT fuse (operation dependent of CKSEL fuses); [CKOPT=0]
- JTAG Interface Enabled; [JTAGEN=0]
- On-Chip Debug Enabled; [OCDEN=0]
- Preserve EEPROM memory through the Chip Erase cycle; [EESAVE=0]
- Serial program downloading (SPI) enabled; [SPIEN=0] \*

LOCKBIT Fuse Presets:

- Application Protection Mode 1: No lock on SPM and LPM in Application Section
- Boot Loader Protection Mode 1: No lock on SPM and LPM in Boot Loader Section
- Mode 1: No memory lock features enabled

Manual Fuse Bit Manipulation

Remember:  = programmed = 0;  = unprogrammed = 1

Bit	LOW	HIGH	LOCKBIT
7	<input type="checkbox"/> BODLEVEL	<input type="checkbox"/> OCDEN	<input type="checkbox"/> Bit 7
6	<input type="checkbox"/> BODEN	<input type="checkbox"/> JTAGEN	<input type="checkbox"/> Bit 6
5	<input type="checkbox"/> SUT1	<input checked="" type="checkbox"/> SPIEN *	<input type="checkbox"/> Bit 5
4	<input checked="" type="checkbox"/> SUT0	<input type="checkbox"/> CKOPT	<input type="checkbox"/> Bit 4
3	<input type="checkbox"/> CKSEL3	<input type="checkbox"/> EESAVE	<input type="checkbox"/> Bit 3
2	<input type="checkbox"/> CKSEL2	<input type="checkbox"/> BOOTSZ1	<input type="checkbox"/> Bit 2
1	<input type="checkbox"/> CKSEL1	<input type="checkbox"/> BOOTSZ0	<input type="checkbox"/> Bit 1
0	<input checked="" type="checkbox"/> CKSEL0	<input type="checkbox"/> BOOTRST	<input type="checkbox"/> Bit 0
Default	0xC1	0x99	0xFF
Apply	0xEE	0xDF	0xFF

```
AVRDUDE -U lfuse:w:0xEE:m -U hfuse:w:0xDF:m -U lock:w:0xFF:m
```

Perma-Link: <http://eleccelerator.com/fusecalc/fusecalc.php?chip=atmega16&LOW=EE&HIGH=DF&LOCKBIT=FF>

This is a javascript based AVR fuse calculator. You must have javascript enabled to use this. There are lots of places with mouse-over text (alt-text) to provide some help.