AVR

Programmer

User Manual

24 September, 2014

Document Title	AVR Programmer User Manual
Version	1.3
Date	2014-09-24
Status	Release
Document Control ID	APSLNUR240914

General Notes

TechShop offers this information as a service to its customers, to support application and engineering efforts that use the products designed by TechShop. The information provided is based upon requirements specifically provided to TechShop by the customers. TechShop has not undertaken any independent search for additional relevant information, including any information that may be in the customer's possession. Furthermore, system validation of this product designed by TechShop within a larger electronic system remains the responsibility of the customer or the customer's system integrator. All specifications supplied herein are subject to change.

Copyright

Contents

Version History

Date	Version	Description of change	Developer
2010-01-27	1.00	Origin	Fahad Mirza
2013-03-02	1.2	8051 Feature added;	Fahad Mirza
2014-09-24	1.3	ISP Header rearranged	Nur Mohammad

1. Introduction:

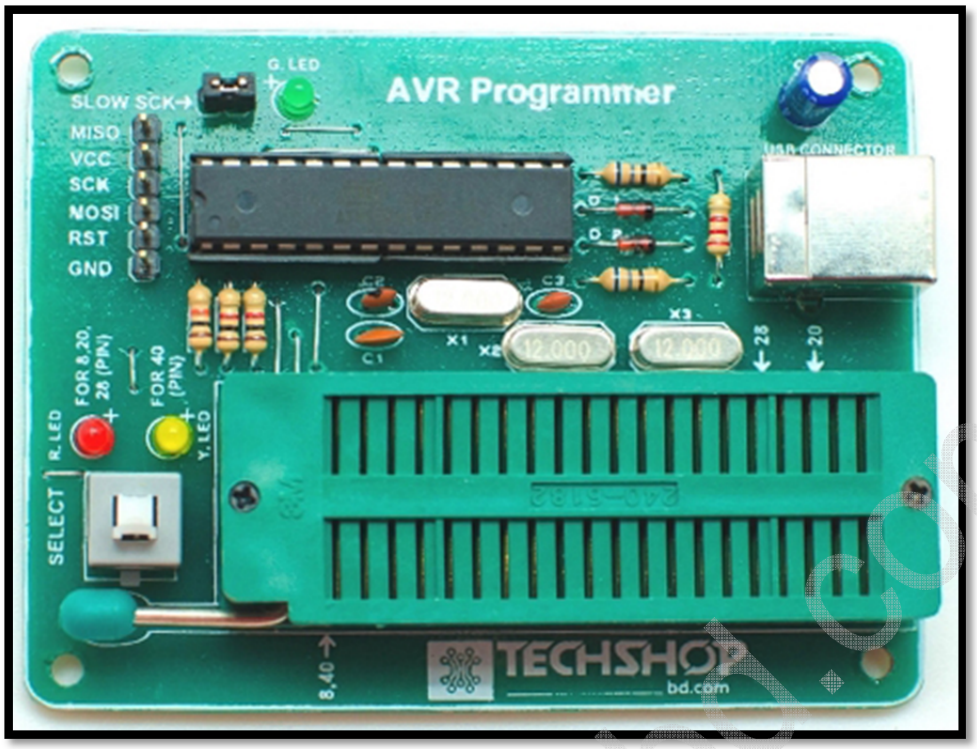
1.1 What is AVR programmer?

AVR Programmer is an USB in circuit programmer for Atmel AVR Microcontrollers. With this programmer you can load .hex files to AVR chips. The programmer uses an USB driver software.

1.2 Technical feature

Some of the features include:

- Connect directly to USB port, can be used with PC or laptop.
- 6 pin ISP interface.
- Support for Linux, Windows XP, Windows Vista, Windows7 (32 and 64 bit).
- Allows you to read or write the microcontroller, EEPROM, firmware, fuse bits.
- 5 kB/sec maximum write speed.
- Powered directly from USB port.
- 3 LEDs: Green, Red, Yellow.
    - Green- Busy Indicator: Glows while hex file is being loaded to the MCU.
    - Red: Should be turned on by the selector switch when the target MCU has 8/20/28 pins.
    - Yellow: Should be turned on by the selector switch when the target MCU has 40 pins.
- Cooperate with: AVRpal, AVRDUDE (with all GUI), WinAVR and more.



⚠ **Cautions:** The 5v supply of the board come directly from USB port of computer; it is advised not to use this power source to power application circuit or device. Wrong connection such as wrong polarity, wrong voltage, shorted might permanently damage your computer.

1.3 Supported Microcontrollers

The table below shows some of the microcontrollers that are supported by the Programmer, so far.

Mega Series				
ATmega48	ATmega8	ATmega88	ATmega8515	ATmega8535
ATmega16	ATmega162	ATmega163	ATmega164	ATmega165
ATmega168	ATmega169	ATmega169P	ATmega32	ATmega324
ATmega325	ATmega3250	ATmega328P	ATmega329	ATmega3290
ATmega64	ATmega640	ATmega644	ATmega645	ATmega6450
ATmega649	ATmega6490	ATmega128	ATmega1280	ATmega1281
ATmega2560	ATmega2561			

Tiny Series				
ATTiny12	ATTiny13	ATTiny15	ATTiny24	ATTiny25
ATTiny26	ATTiny2313	ATTiny44	ATTiny45	ATTiny84
ATTiny85				

8051 Series				
AT89S51	AT89S52	AT89S53	AT89S8252	AT89S8253
AT89S2051	AT89S4051			

1.4 Kit contents

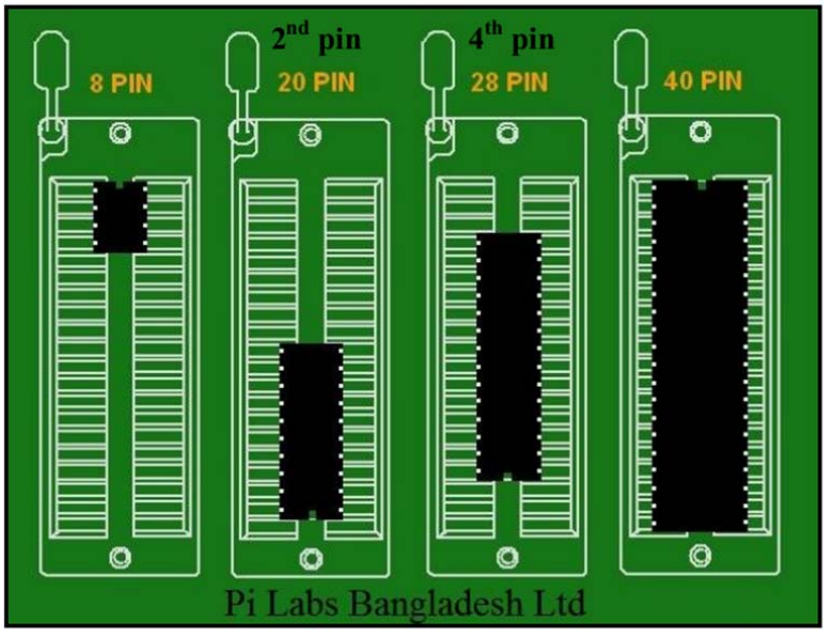
The kit contents:

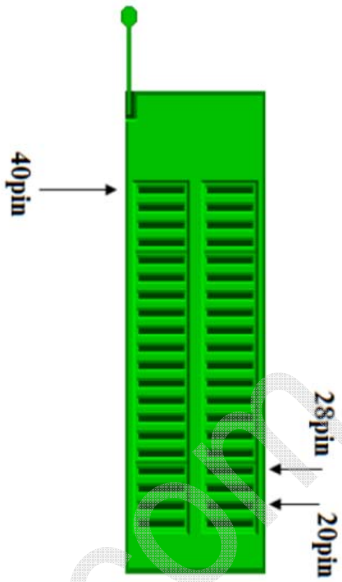
- Programmer, compatible with USBasp.
- Standard USB cable (A to B).
- User Guide DVD

2.Chip Placement

2.1 Zif Socket

Zif socket supports all the AVR chips except 8051(AT89S series). For 8051 MCU you have to use ISP connector.





There are two switches for choosing amongMCUs according to their number of pins.

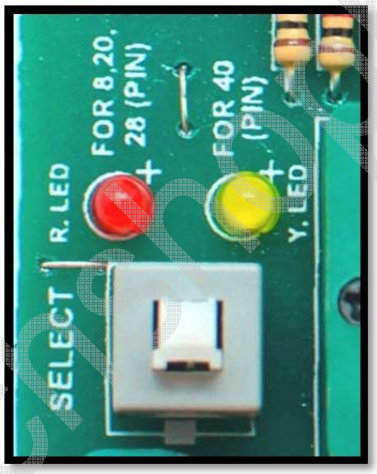


Fig: For 40pin MCUlight the yellow LED. For 28, 20, 8pinturn on the red LED.

2.2 ISP connection

If you are planning to burn chip through ISP connector, you need to follow some steps:

1. Connect the corresponding pin (MISO, MOSI, SCK, RST) of the ISP with the corresponding pin of Target MCU.

2. Supply Vcc and GND. Make sure all the Vcc and GND pins (if there is more than one) are shorted respectively.

3. If the target MCU is set to 'External Clock/Crystal' mode then you have to place a crystal (usually 12-16MHz) between XTAL1 and XTAL2 pin.

4. For AT89S series connect like the schematic below:

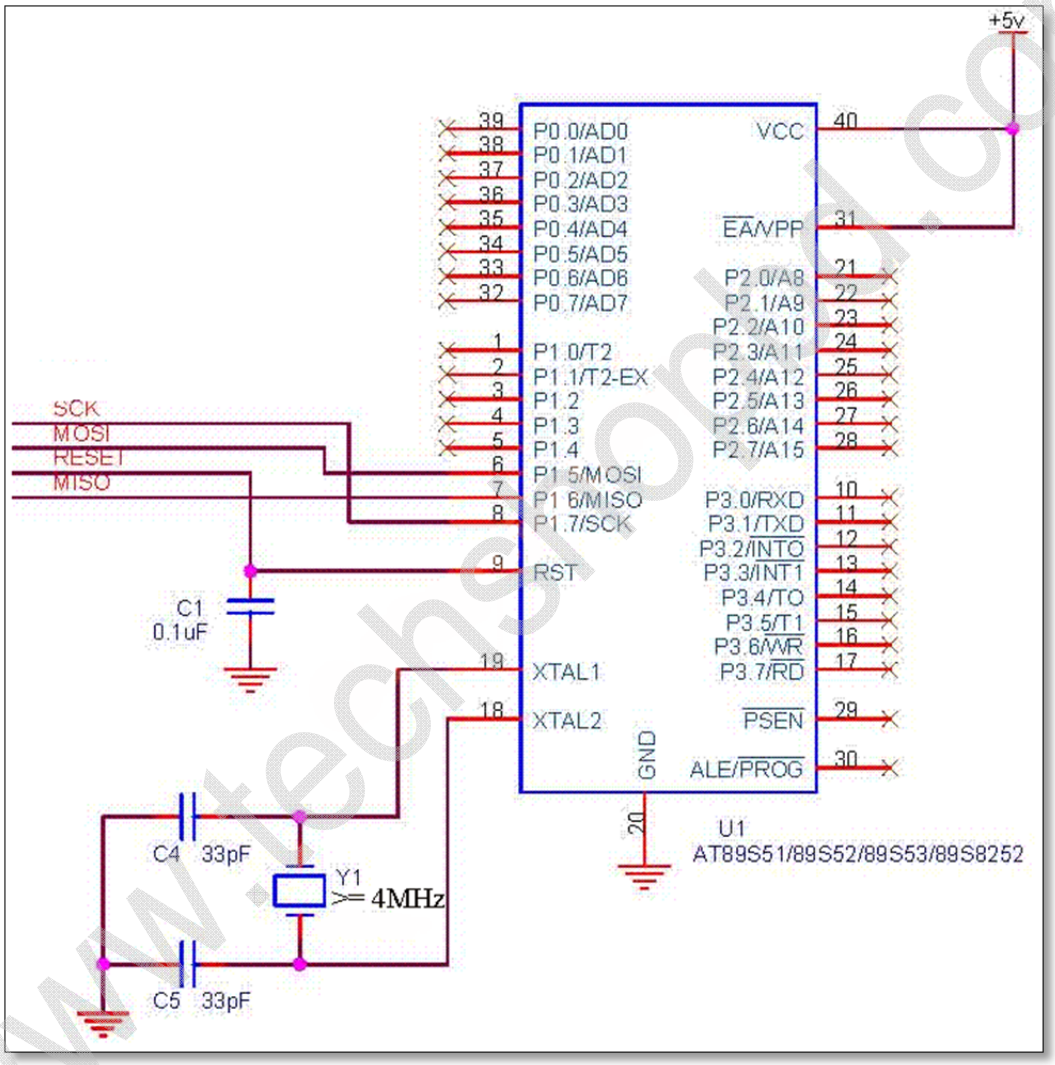


Fig: ISP Connector

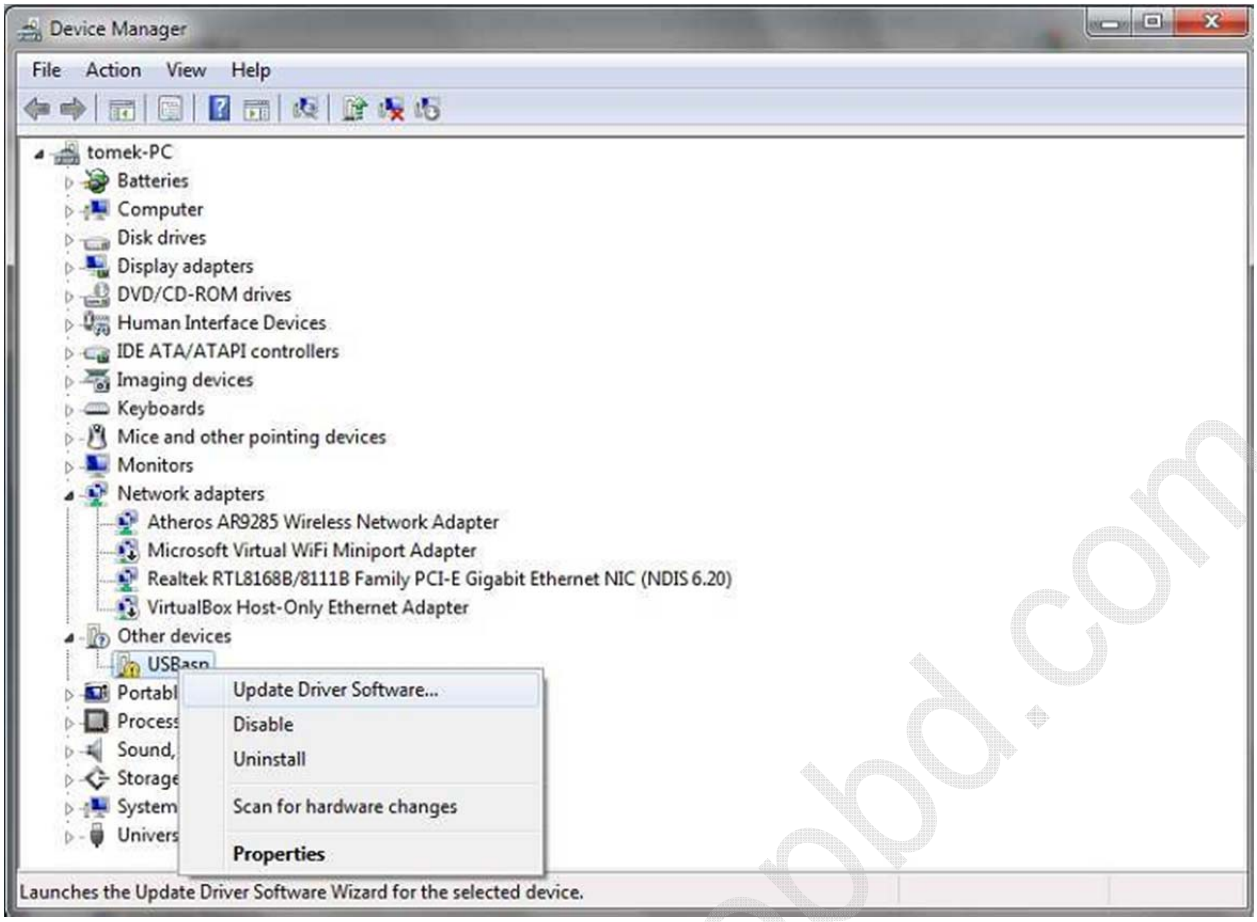
3.Installation

On Linux and Mac OS X no kernel driver is needed. Windows requires a driver. In order to complete the installation, you need to follow several steps. This procedure will only focus on Window 7.

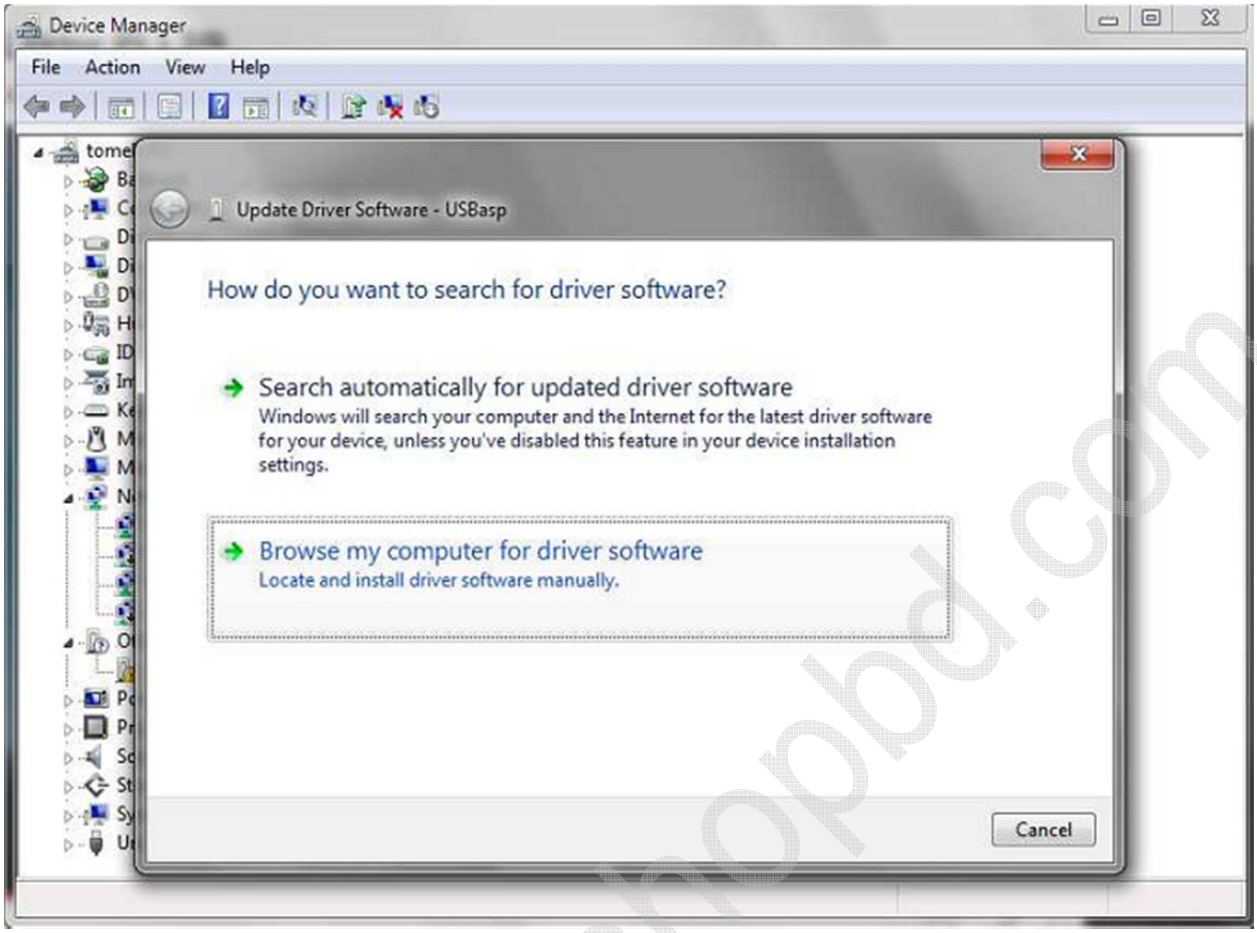
1. Unzip the 'Driver' file (include in the DVD and in our site) and connect the AVR programmer to the USB port of your PC.

2. Insert programmer to USB port in your computer.

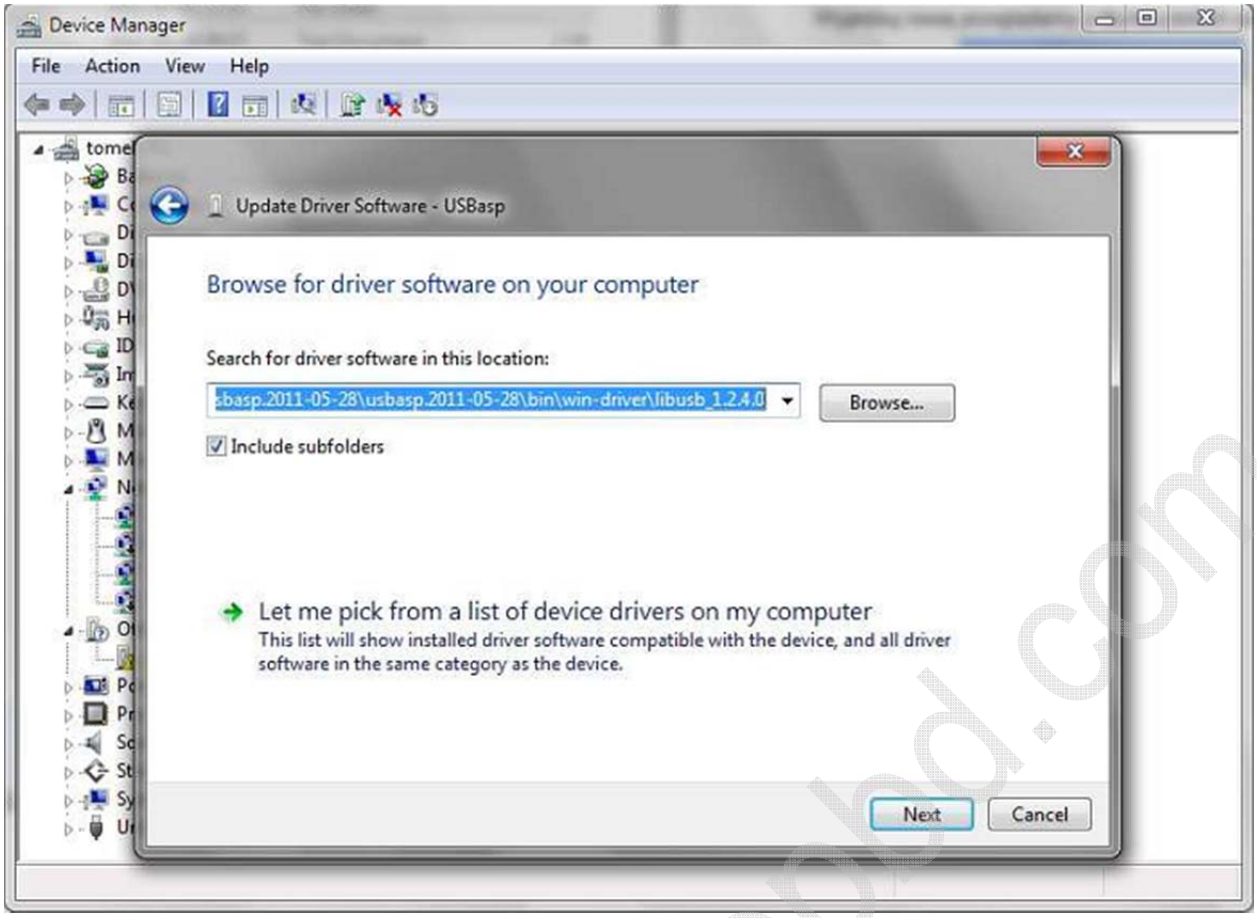
3. Open Device Manager, find the entry for the USBasp and it should be displayed with a yellow alert icon on it. Then right click on the device and select "Update Driver Software".



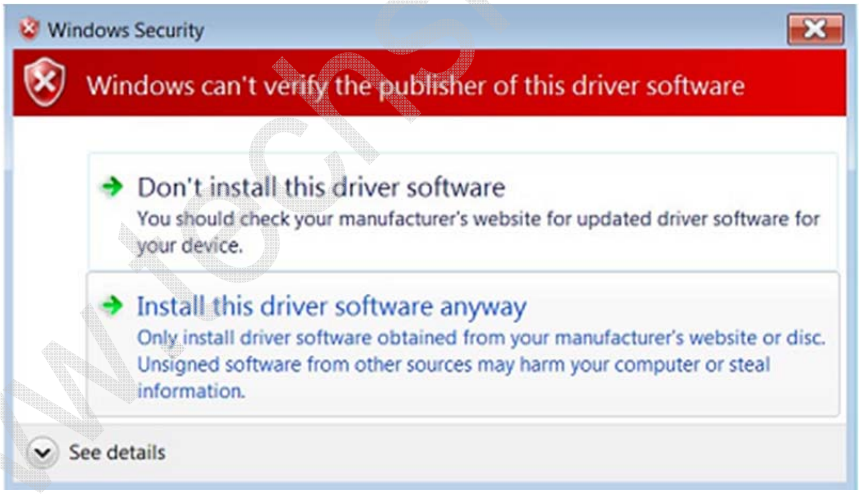
4. After you left click the “Update Driver Software”, it will come out with “How do you want to search for driver software?” Then choose the second one which is “Browse my computer for driver software” and click into it.



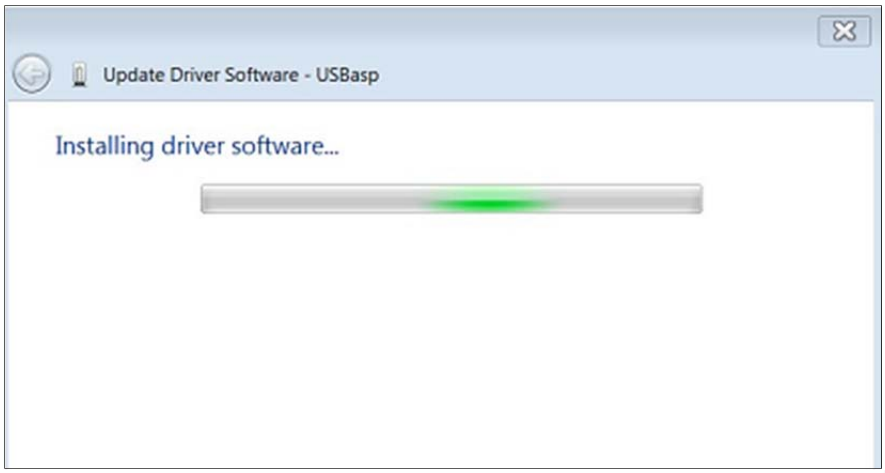
5. After that, you will see the screen which will prompt out “Browse for driver software on your computer”. In this step, you need to select the folder whereyou unzipped the driver files then click “Next”.



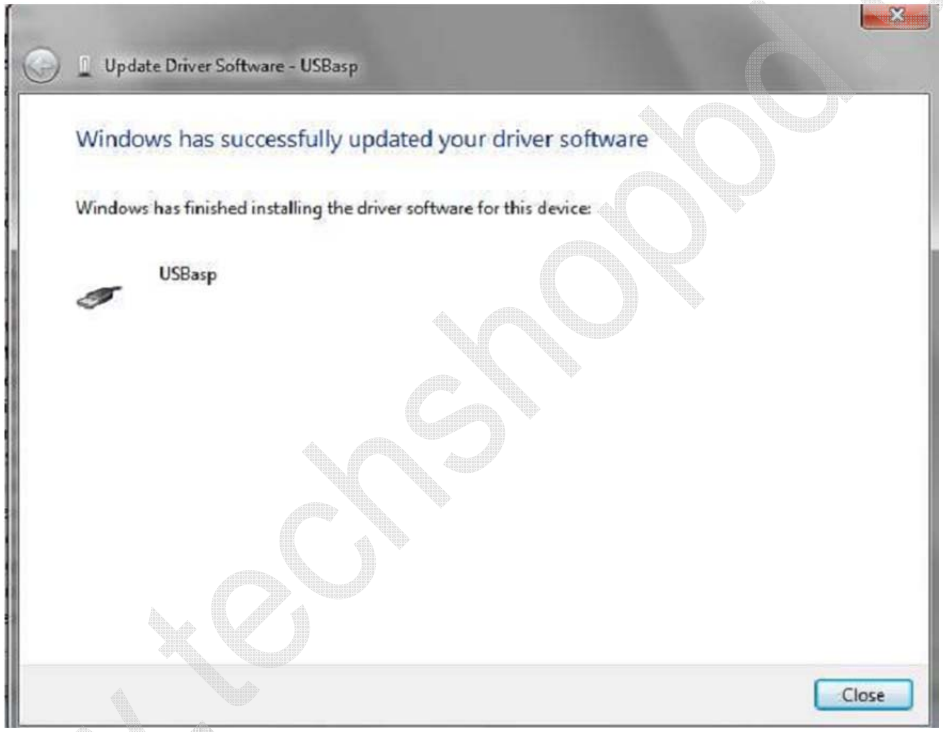
6. Next, the windows will prompt out a "Windows Security" with a red warning dialog. Do not worry about it, and just click "Install this driver software anyway" and the driver will install.



7. After click it, the next step is to wait a few seconds to let your computer to process the installation of driver software.



8. Now, you can use the programmer to do the programming for themicrocontroller.



4. Software

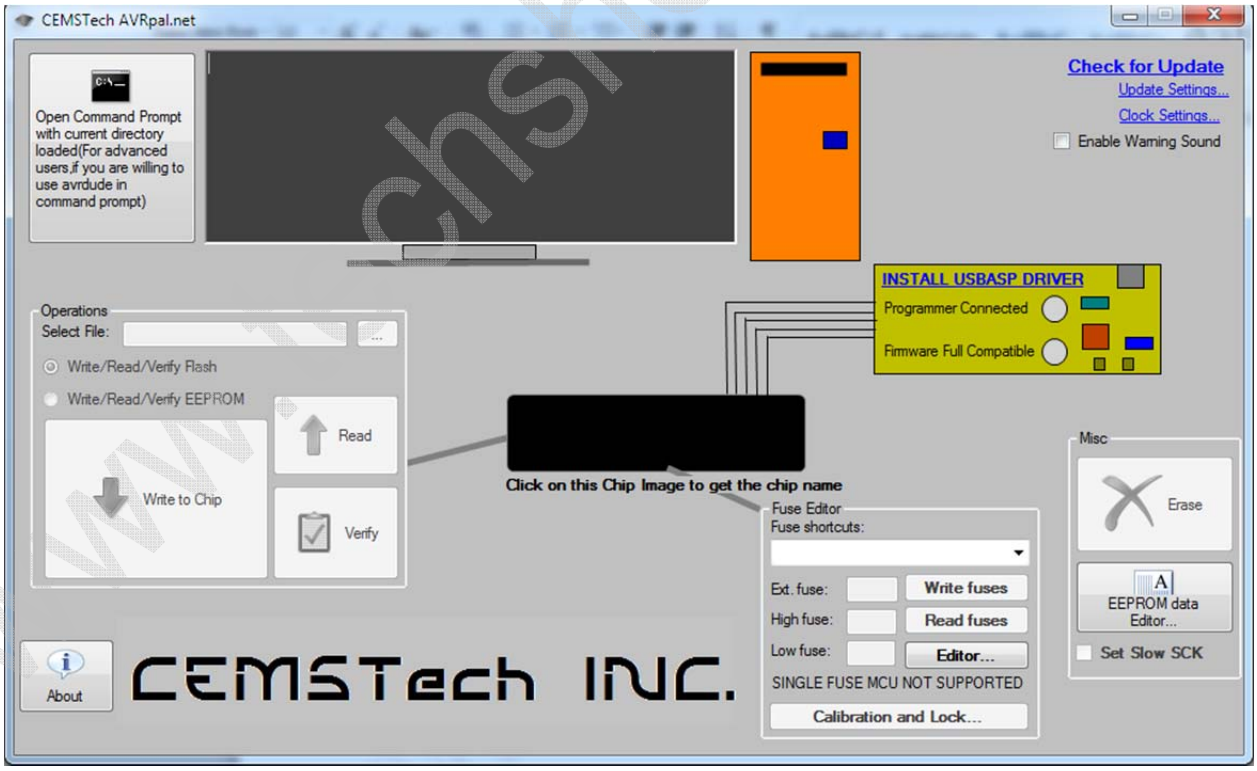
There are varieties of software which can be work also for the programmer.These are including:

- AVRpal – Version 3.1 or later. A great GUI of avrdude. We normally use this software.
- **Khazama AVR Programmer** – An AVRdude GUI for MS Windows.
- **BASCOM-AVR** – Version 1.11.9.6 or later.
- **eXtreme Burner** – An easy to use GUI application.
- **ProgISP 1.72** – Supports various hardware, including USBasp, STK200 etc. For AT89S series we will use this software.

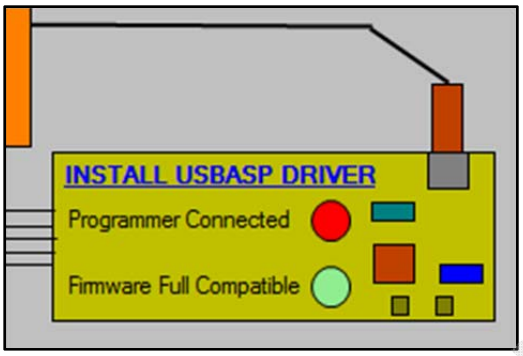
For the list of the software above, we have no responsibility to teach users how to use;users must study themselves in order to use it.

4.1 AVRpal

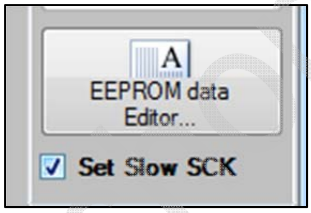
1. At the time I am writing this manual, I have V3.1 in my hand.

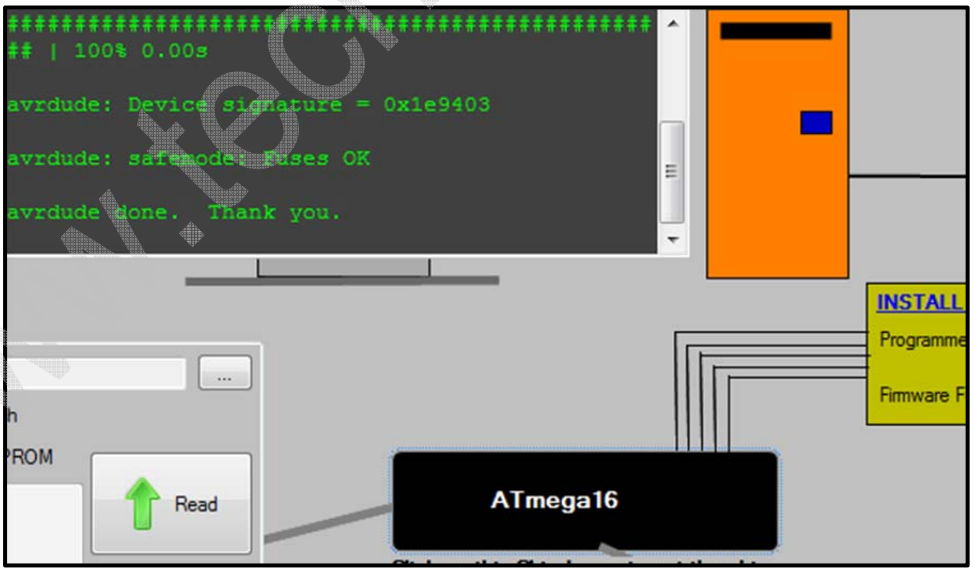


2. If AVRpal.net detect you programmer, then “Programmer Connected” and “Firmware Full Compatible” LED will lit.

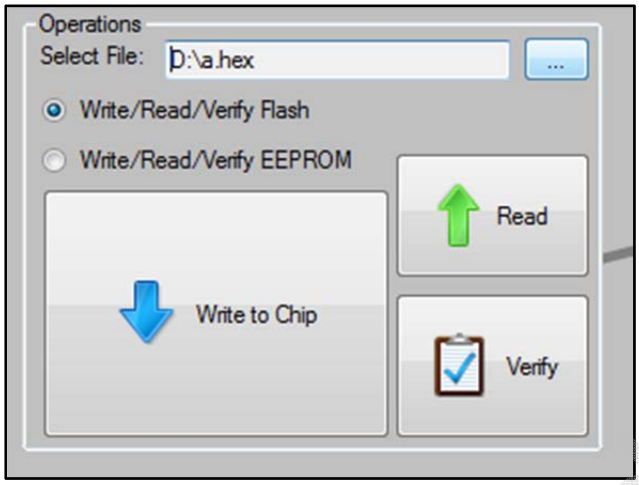


3. Now click on the “Black Box” (which actually represents IC -_-), and software will detect which MCU is present in the zif socket. Make sure you checked “Set Slow SCK”. A detail on “Slow SCK” isdiscussed in “Fuse Bytes” section.

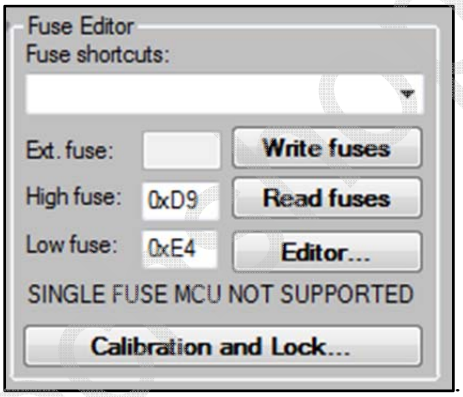




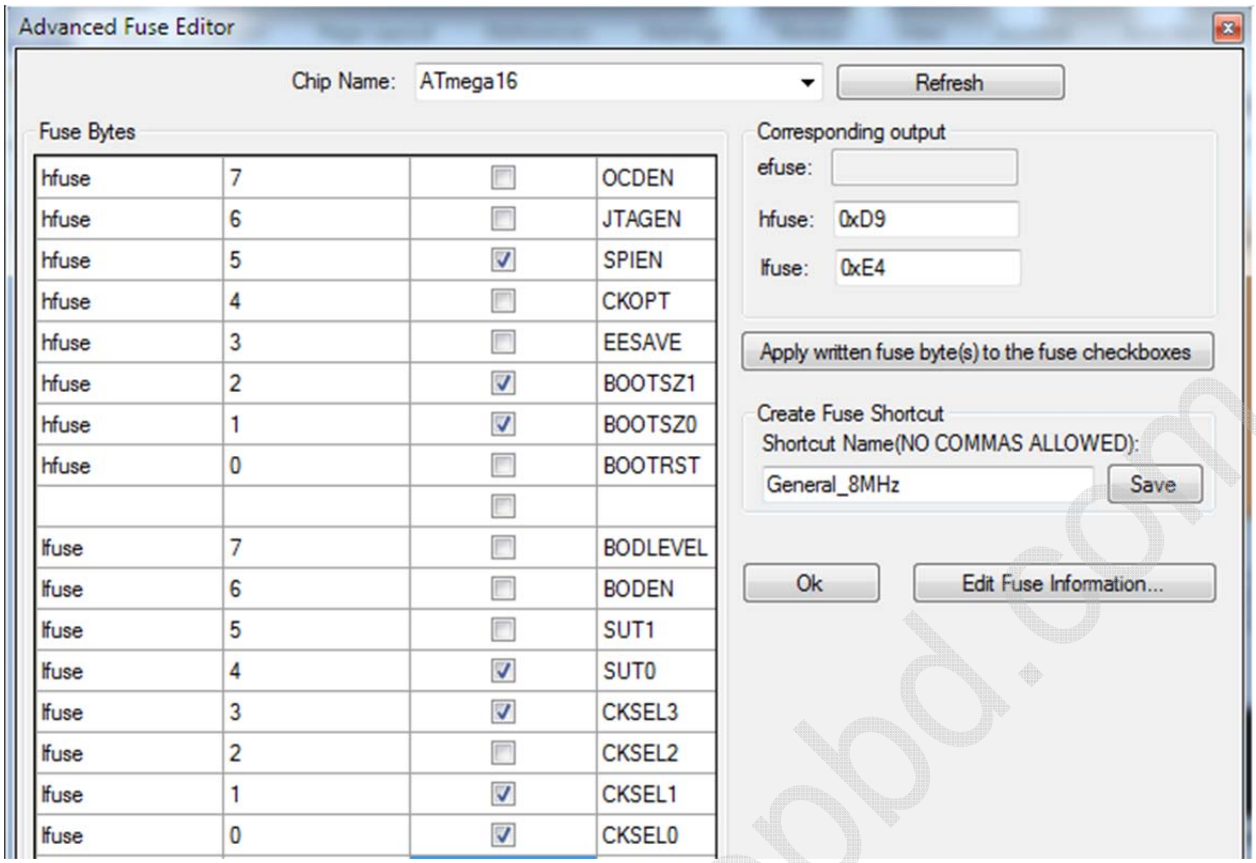
4. So, it detects my MCU (ATmega16). To load “Hex” file browse it from “Select File”.



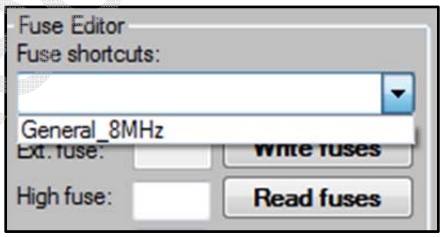
5. You can also read/write fuses. Take great care when you change the fuse bits. If do not know anything about fuse bits, please read “Fuse bytes” section first



6. By using Fuse Editor you can save fuse byte for specific MCU.For instance, I frequently use L=0xE4 / H=0xD9 fuse bytes. So I named it “General_8MHz” and then press “Save”.

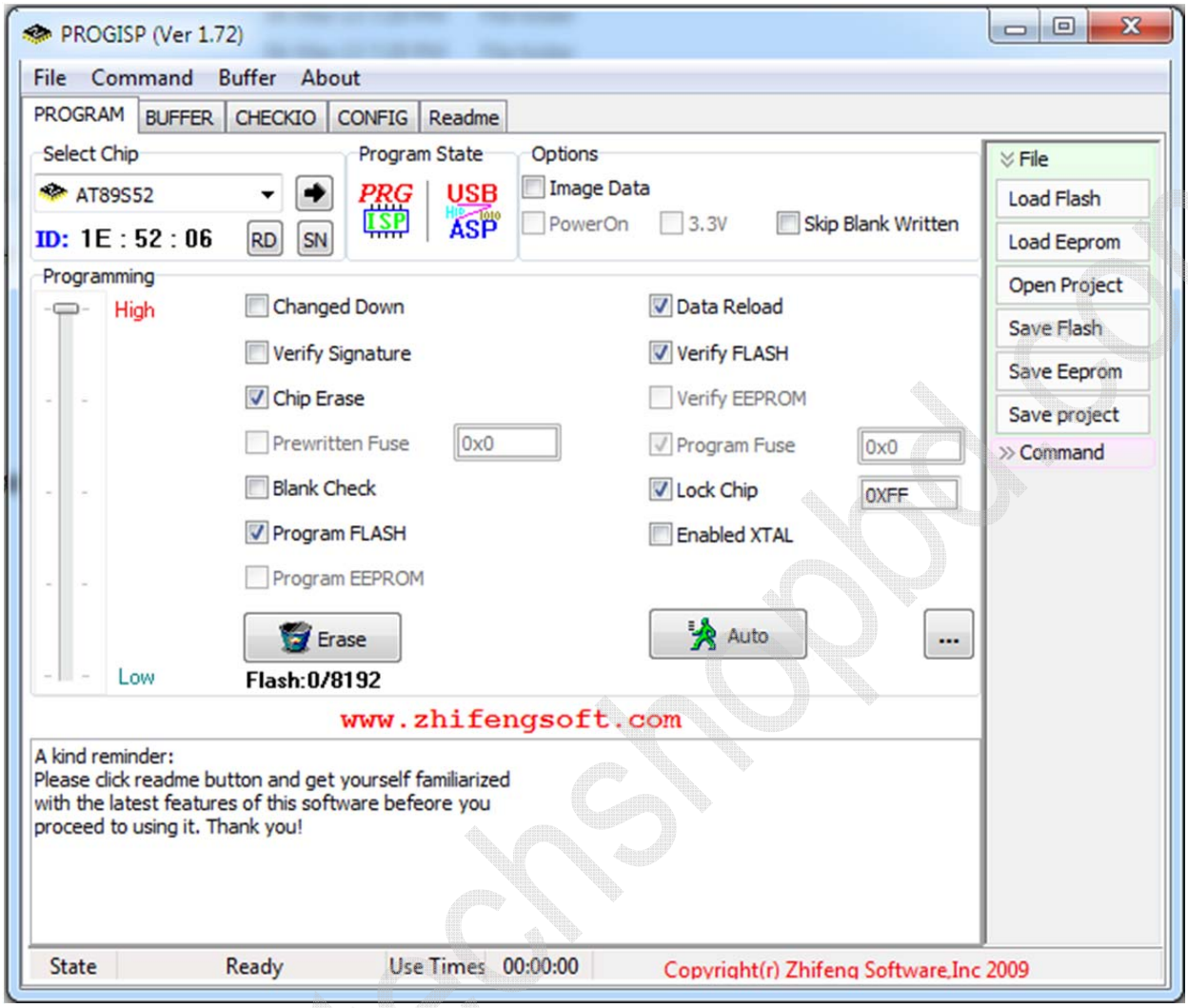


7. Now if you click on the “Fuse Shortcuts” drop down menu, you’ll see “General_8MHz”. If you select it, corresponding fuse bytes will load. If there is another MCU in zif socket other than ATmega16, it won’t be available, which prevents write wrong fuse bytes!!



4.2 ProgISP

1. Run “progisp.exe”.



2. From “Select Chip” choose your desired chip (like: AT89S52). Signature byte will show beside ID (i.e. 0x1E5206).

3. Assuming you already connect your MCU with ISP, so now press ‘RD’. If the signature bytes match it will show “Read ID successfully”.

4. Browse through “File” and “Command” menu for more options.

5. Fuse Bytes

I guess many of you were confused when programming AVR fuse bytes. I get many newbie questions like I programmed AVR but it doesn't work. 90% of them always set wrong fuse bytes and make them DEAD or unusable. In this tutorial first I will discuss about fuse bits and then how to active so called "DEAD" chips.

5.1 Introduction

Fuses are an extremely important part programming a chip, but are rarely explained thoroughly. You only need to set them once, but if you don't do it right, it's a disaster!

You know about flash, EEPROM and RAM as parts of the chip. What I did not mention is that there are also 3 bytes of permanent (by permanent I mean that they stick around after power goes out, but you can change them as many times as you'd like) storage called the fuses. The fuses determine how the chip will act, whether it has a bootloader, what speed and voltage it likes to run at, etc. Note that despite being called 'fuses' they are re-settable and don't have anything to do with protection from overpowering (like the fuses in a home).

The fuses are documented in the datasheets, but the best way to examine the fuses is to look at a fuse calculator such as in "MikroC" compiler. Please collect the latest version. When I am writing this tutorial I have version 5.6 in my hand.

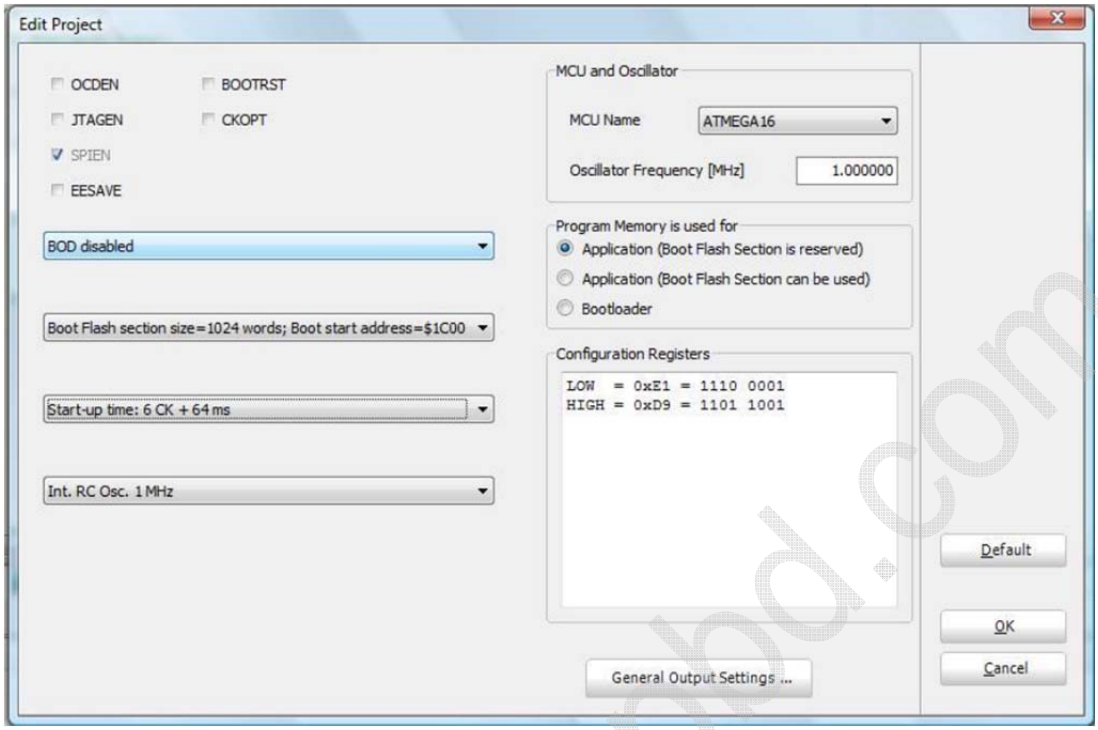
If you want to enable something in AVR what you do? Most probably you set corresponding bit as '1', right? For Fuse bit its opposite. Here „1" means un-programmed or disable and '0' means programmed or enable.

Always remember these.

Fuse bit = 0 => fuse bit is PROGRAMMED

Fuse bit = 1 => fuse bit is UN-PROGRAMMED

Open a project. I am assuming that you are using ATmega16. Go to "Project Menu" and click on "Edit Project" (Shifht+Ctrl+E). A window will open just like below:



SPIEN bit is for programming. It should be enable for load program into chip. If you disable it you can't program no more. So 'MikroC' make this fixed and you can't change it. If you check other bits (Like: JTAGEN) you will see the corresponding bits in "Configuration Registers" box (bottom right) changed to Zero. For more about other fuse bits browse ATmega16's datasheet, page 255. I like to mention one thing before I go to next stage. By default in ATmega16, JTAG enabled. That's why you can't use PORTC2-5 as digital I/O. So you have to disable it.

5.2 Brown Out Detection (BOD)

The first drop down menu is "Brown Out Detect". These fuses set what voltage to turn the **Brownout**protection circuitry on. A brownout for a chip means that the power voltage is too low for it to runreliably at the speed of the clock.

For example, the ATTiny2313 can run as fast at 20MHz but only if the power voltage is between 4.5V and 5.5V. If the voltage is lower than that, it may behave erratically, erasing or overwriting the RAM

and EEPROM. It may also start running random piece of the flash program. To keep it from doing that, set the brownout voltage to 4.3V, then if the voltage dips, the chip will turn off until the voltage returns. It will then reset and start over.

If the chip is meant to run at 5V, set the brown-out to 4.3V. If the chip can run as low as 3.3V you can set the brown-out to 1.8V. If the chip is a 'low voltage compatible' chip such as the attiny2313V (which can run as low as 1.8V if its clocked at 4MHz or less) then you can set the brownout to 1.8V.

For simplicity, disable it.

5.3 Clock Selection

The 2nd option is how the chip is clocked. Every CPU uses a clock, in general one assembly code instruction is run every clock cycle. The one in your PC has a clock that runs at 1GHz or higher. This little chip runs much slower. If you look at the menu you'll see a huge list of options, but looking carefully you'll see there are two groupings, the **Clock Source** and the **Startup Time**.

The Clock Source can be either of the following:

```
External Clock, Internal 8MHz clock, Internal 4MHz clock,
Internal 128KHz clock, External Crystal (0.4-0.9 MHz), External
Crystal (0.9MHz  -  3.0MHz), External Crystal (3.0MHz - 8.0MHz)
or External Crystal (8.0MHz +)
```

External Clock means that a square wave is being input into the **CLOCK-IN** pin. This is pretty rare unless you have a clock generating chip. Don't use this unless you're sure you mean to.

Internal Clock means that there's a little oscillator inside the chip, it's not very precise but good for most projects that don't have fine timing issues. The clock varies with temperature and the power supply voltage. You can choose from a 1MHz, 2MHz, 4MHz or 8MHz clock. Having an internal oscillator means we don't need to wire up a crystal and we can use the clock pins for our own nefarious purposes.

External Crystal,If you need a special clock rate, like 3.58MHz or 12MHz or a high precision clock that won't drift with the temperature, you'll want an external crystal or oscillator.

Crystal Oscillator

Ceramic Oscillator

Clock settings have following options:

Beginners usually use internal 1MHz to 8 MHz. Sochoose internal oscillator. For this tutorial I choose Int. RC Osc. 1 MHz.

5.4 StartupTime

The Startup time can be either of the following: 6CK + 0 ms, 6CK + 4 ms, 6CK + 64 ms.

The Startup Time is just how long the clock source needs to stable from when power is first applied. Always go with the longest setting 6CK + 64ms unless you know for a fact your clock source needs less time and 64ms is too long to wait.

The "Start-up Time" menu has following options. Choose the 64ms:

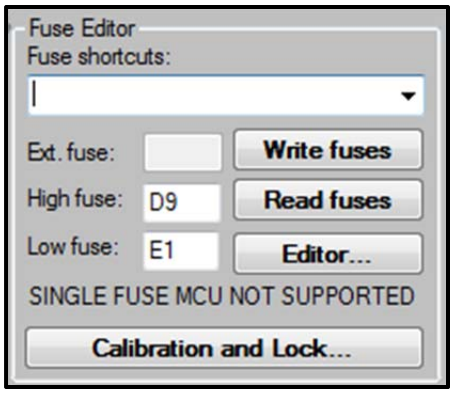
By default, chips that come from the factory have the Internal 1 MHz clock with 6CK + 0ms Startup.So, in configuration register box I see the fuse bits are: LOW: 0xE1 HIGH: 0xD9

It's time to load this bit into chip.

5.5 Writing Fuse Bytes (By AVRpal.net):

If your chip is brand new, then by default its clock frequency set into internal 1 MHz. So, in AVRpal.net you have to check "Set Slow SCK" box. If you change the fuse bits for higher frequency then you can either "check" or "uncheck" Slow SCK, your choice. The difference is, when "Slow SCK" is checked the program will load slowly and vice versa. REMEMBER, if clock set to 1 MHZ, you have to check "Slow SCK" box. Otherwise it won't work.

So write fuse bytes in corresponding box and press 'Write fuses'.



6 .Troubleshooting!

I think the above tutorial will help you set fuse bits properly. But though what if you set wrong fuse bit? (Most of the time it happens with pony prog user). Here comes the solution:

If your programmer does not detect you chip:

1) First check, is "SLOW SCK" checked (in AVRpal)? If not, then put 'Check' and try again.

2) If till problem remains then put a crystal (12 or 16MHz, I prefer 16MHz) between XTAL1 and XTAL2 pin (if you are using ISP connector to program), with "SLOW SCK" checked. Most of the time problem will solve in this stage.

3) If till problem exist then maybe you disable SPIEN bit, which is unlikely, a rare situation, but definitely it isn't for wrong clock fuse bit. If that is the situation then you need high voltage programmer (like TopWin). Also you can use "Fuse bit Doctor" (Google it!).

4) Check the Target MCU's Vcc and GND pin; see if they get between 4.5 to 5.5V.

5) 'Short Test'of MOSI, MISO, SCK, RST pin between your programmer and your chip.

I hope above five steps will solve your problem.

7 .Warranty

Product warranty is valid for 6 months. Warranty only applies to manufacturing defect. Damage caused by misuse is not covered under warranty. Warranty does not cover freight cost for both ways.

Servicing is free for life time.

Thank you for using TechShop's Product.

LIMITATIONS

Hardware:

The circuitry of this programmer can only be used for programming 5V target systems. For other systems a level converter is needed.