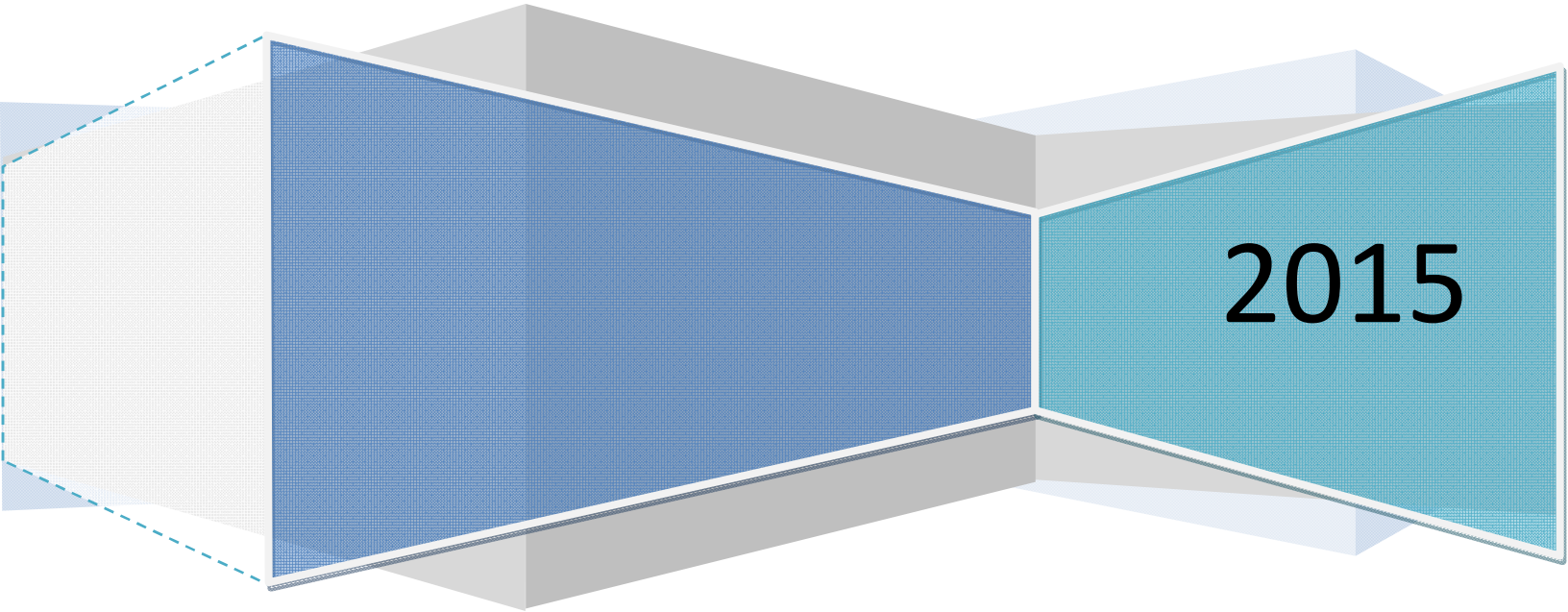
Short Circuit Designs Co., Ltd

ATmega Evaluation KIT REV5
User Manual V1.0



Contact Us

Looking for a low cost engineering solution? Do you have a special idea that is just impossible to implement? Or do you just want to chat about embedded systems? Get in touch with us and we will be glad to help you answer those questions.

E-mail: sales@shortcircuitdesigns.com
support@ shortcircuitdesigns.com

Website: www.shortcircuitdesigns.com

Chapter 1: ATmega Evaluation KIT REV5 System Introduction

Chapter 2: AVR Studio 6.0 IDE Integrated Development Environment

Chapter 3: Installing programming software and USB driver

Chapter 4: ATmega Evaluation KIT System Functional Modules Details

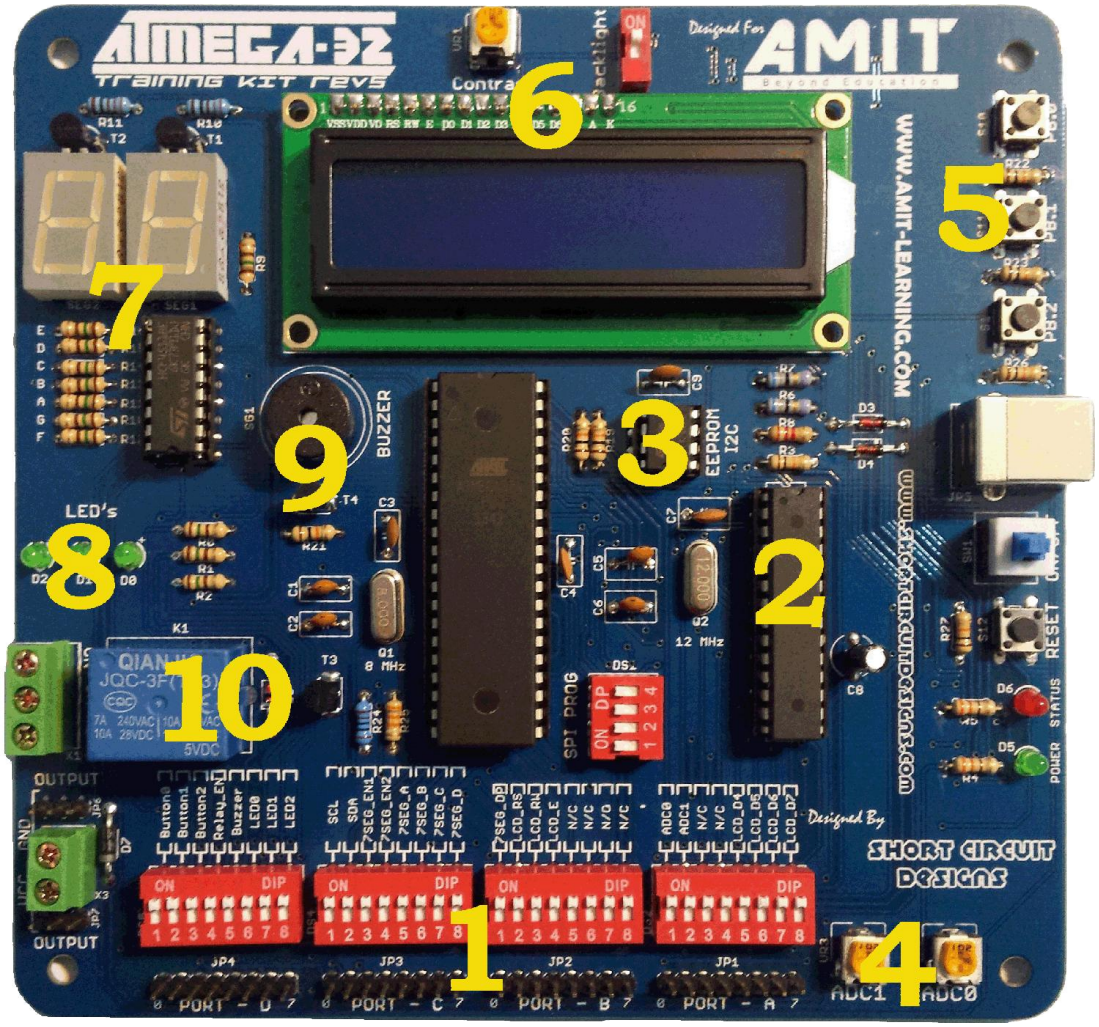
ATmega Evaluation KIT REV5 System Introduction

1.1 Product Overview

ATmega Evaluation KIT is a multifunctional MEGA32/16 microcontroller development platform which has been carefully designed and developed by Short Circuit Designs Co., Ltd. With this product, beginners in the embedded systems track will have all the necessary resources that would enable them to fully master AVR Family microcontroller programming technology in the shortest time possible. It is particularly suitable for self learning for students and/or hobbyists.

The following points may illustrate how your choice was wise.

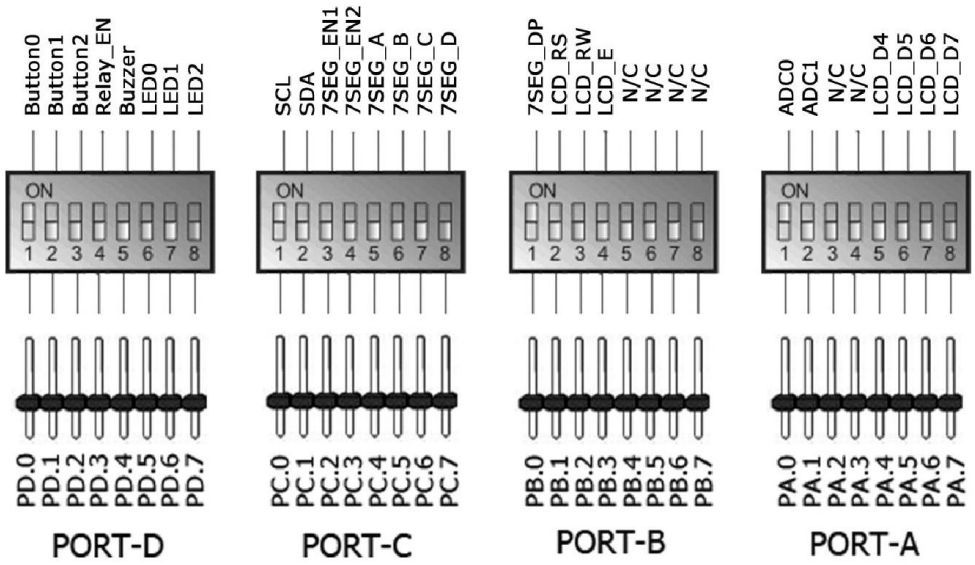
- Optimized modular design
- Superior production technology
- Low selling prices
- Comprehensive technical guidance
- Perfect after service



1.2 Board Resource Introduction

1) General Input / Output Module
2) SPI Programmer Module
3) I^{2}C EEPROM Module
4) A/D converter module
5) Triple Pushbuttons Module
6) 16x2 character LCD module
7) Two- digit 7segment display module
8) 3-bit LED Module
9) Buzzer module
10) Relay Module

1.3 Products Pin Diagram:



- **Three-Bit LED**
  - LED0→PORTD .5
  - LED1→PORTD .6
  - LED2→PORTD .7
- **Buzzer**
  - Buzzer→PORTD .4
- **RELAY**
  - Relay_EN→PORTD .3
- **Pushbutton**
  - Button0→PORTD.0
  - Button1→ PORTD.1
  - Button2→ PORTD.2
- **7 Segment Display**
  - DATA LINES:
    - 7SEG_A→PORTC .4
    - 7SEG_B→PORTC .5
    - 7SEG_C→PORTC .6
    - 7SEG_D→PORTC .7
  - DECIMAL POINT:
    - 7SEG_DP→PORTB .0
  - ENABLE LINES
    - 7SEG_EN1→PORTC .2
    - 7SEG_EN2→PORTC .3

- **I^2C EEPROM Module**
  - SCL →PORTC.0
  - SDA→PORTC.1

- **16X2 CHARACTER LCD**
  - DATA LINES
  - LCD_D4→PORTA .4
  - LCD_D5→PORTA .5
  - LCD_D6→PORTA .6
  - LCD_D7→PORTA .7
  - CONTROL LINES
  - LCD_RS→PORTB .1
  - LCD_RW→PORTB .2
  - LCD_E→PORTB .3

- **ADC**
  - ADC 0 → PORTA.0
  - ADC 1 →PORTA.1

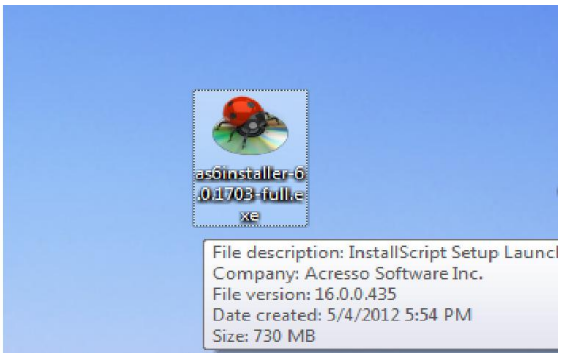
Chapter 2 AVR Studio 6.0 IDE Integrated Development Environment

AVR Studio 6.0 IDE is the powerful software integrated development environment provided by Atmel for its AVR microcontrollers. It allows users to create, record, edit and compile programs for AVR microcontroller series on their own computer systems.

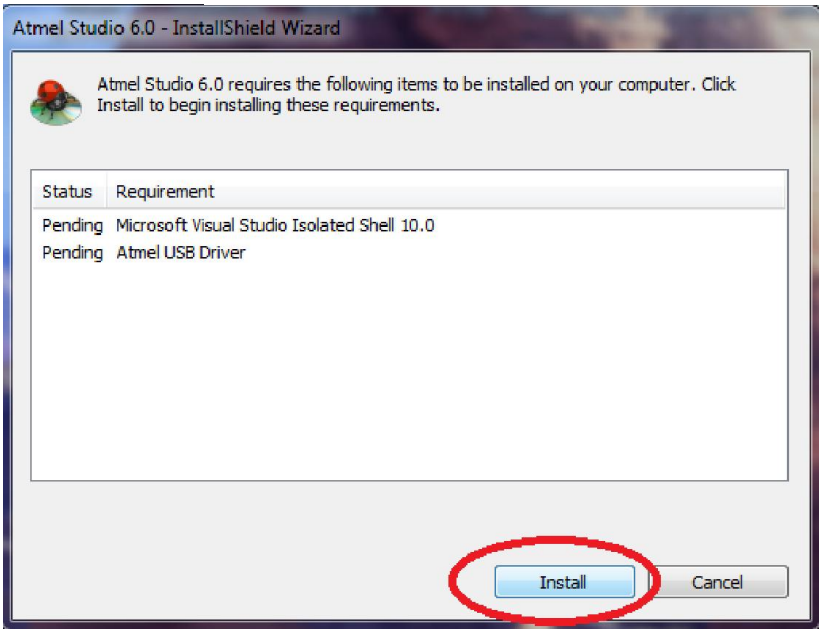
2.1 Installing AVR Studio 6.0

AVR Studio is completely free software offered by Atmel. You can get the latest installation files either from the installation disk included with the KIT or from the following website site: http://goo.gl/dWpKNd .

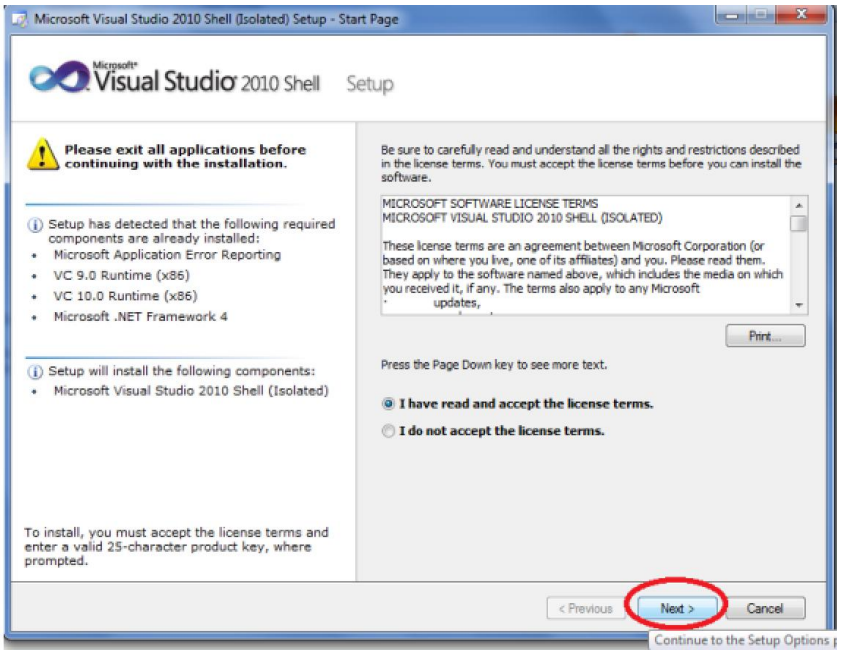
After download is completed, Double click on the installer file to start installation



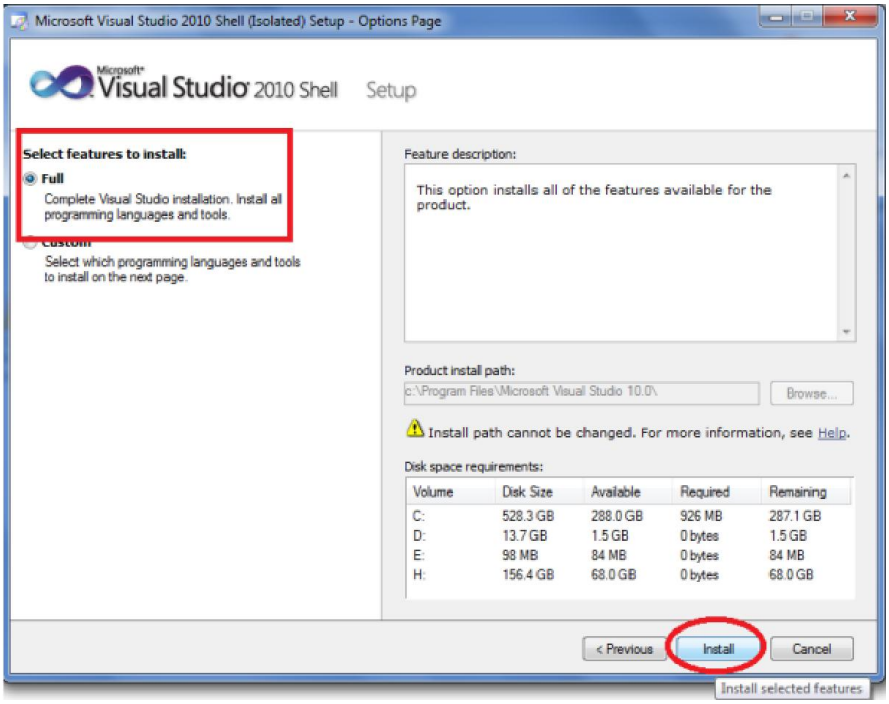
Click on the install option.
If you already have Visual Studio 2010 installed then it won't re-install.



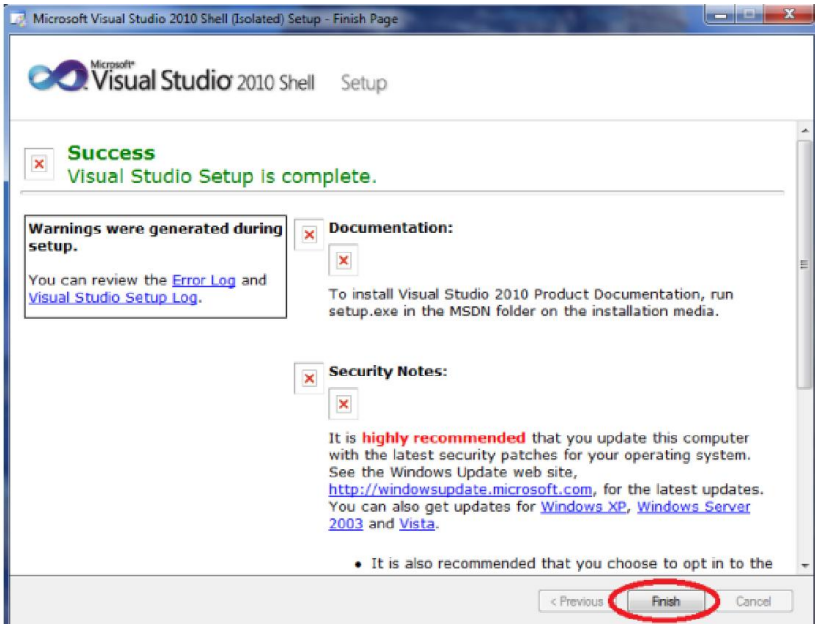
Click Next to proceed



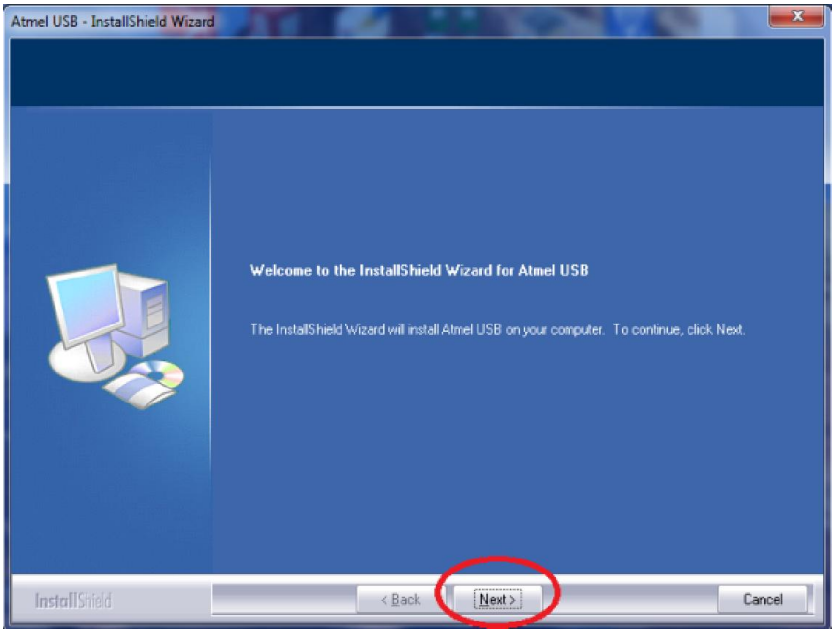
Select FULL Install and Click Next (Default is FULL)



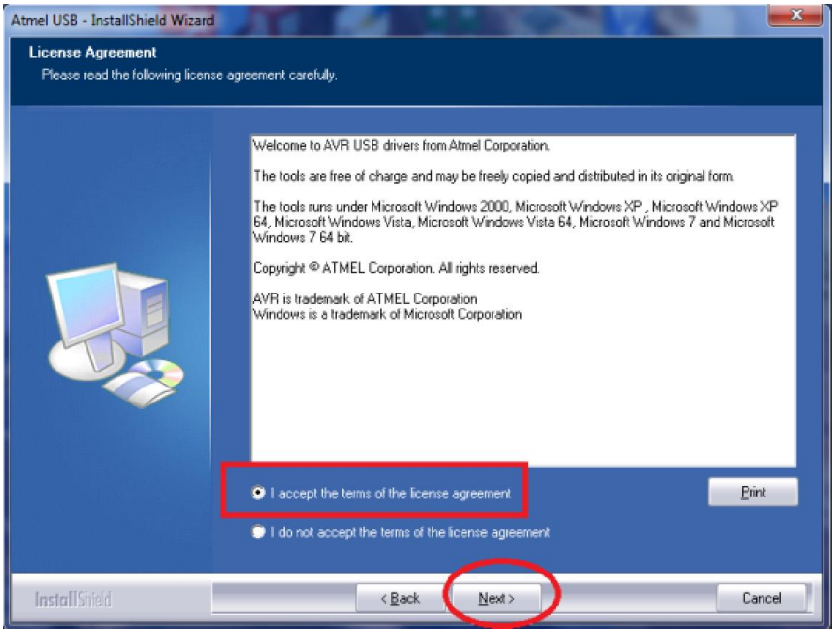
In case error occurs – Can't Write to file ….. config….. Please check for permissions,
just hit Retry to resume install, then click Finish



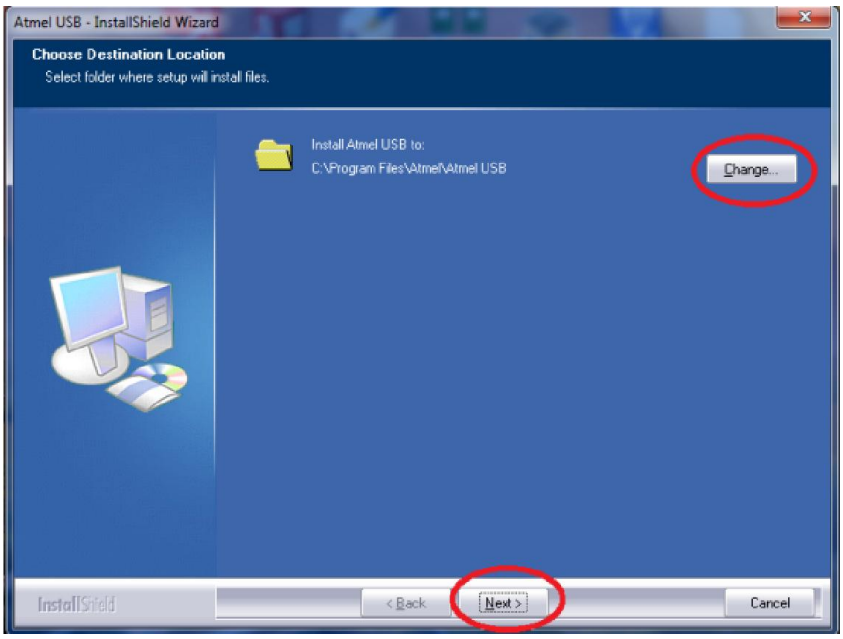
Now to install ATMEL USB Driver
Click Next



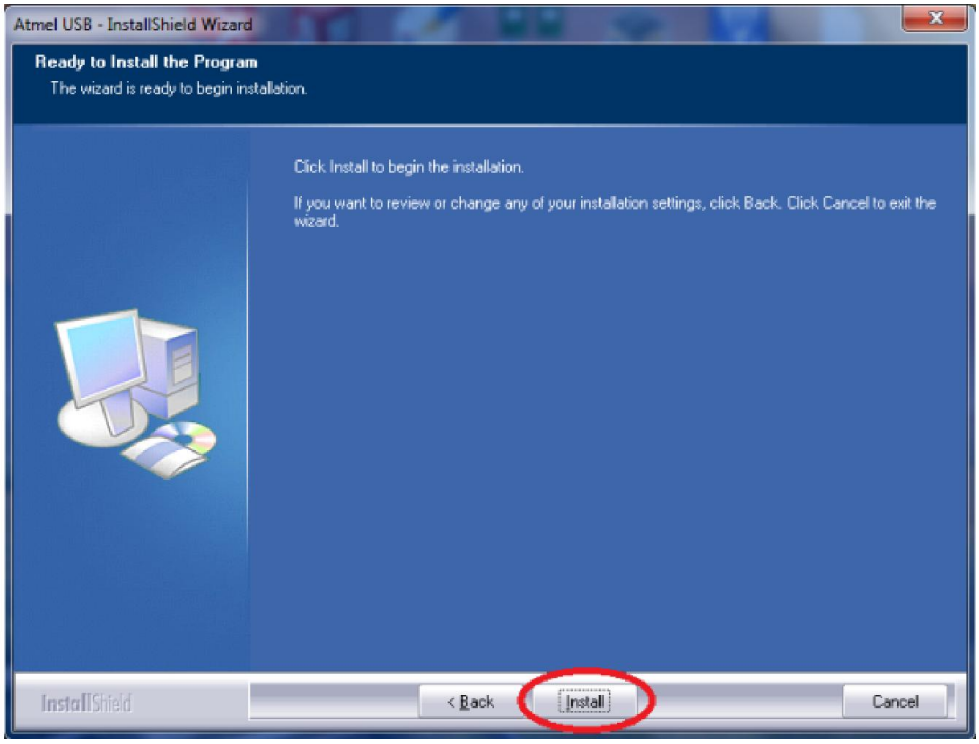
Accept the Terms and Conditions and Click Next



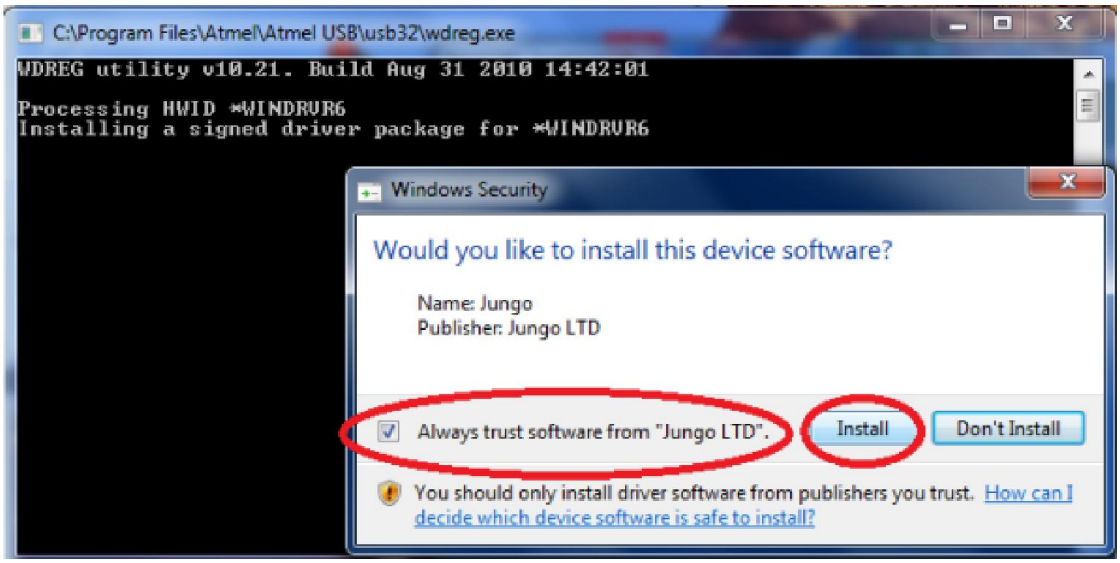
Click Next



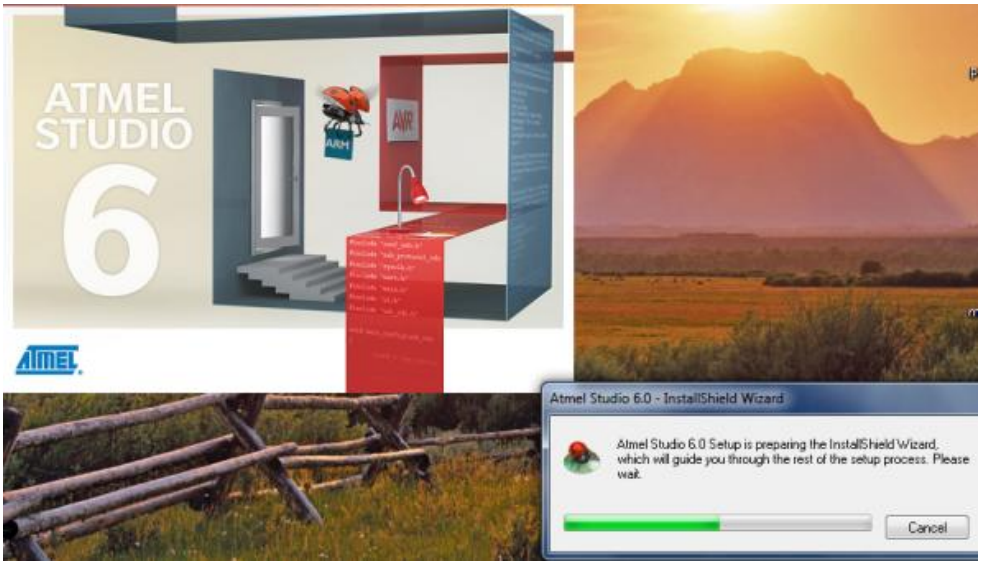
Click Install



Check Always Trust software from “Jungo LTD” option and Click Install

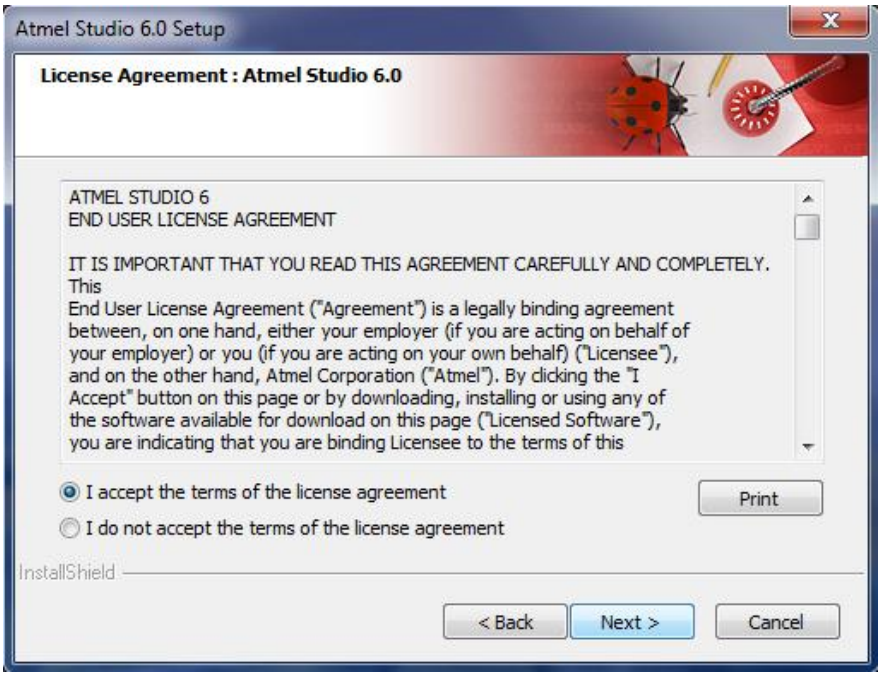


Click Finish, then you should see an Atmel studio splash screen loading after

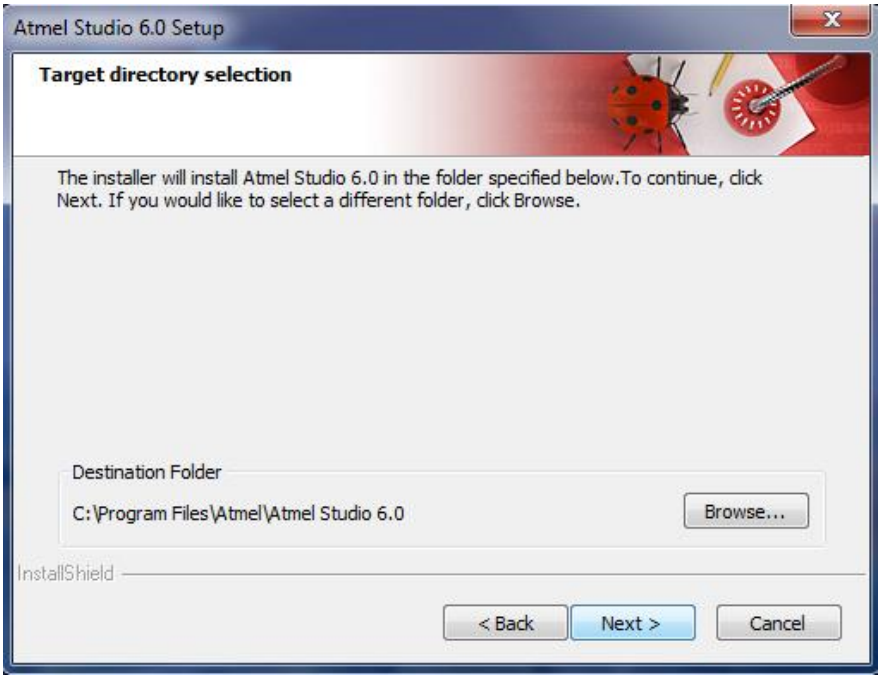


Click Next to proceed

Accept the Terms and Conditions and Click Next



Click Next, Next

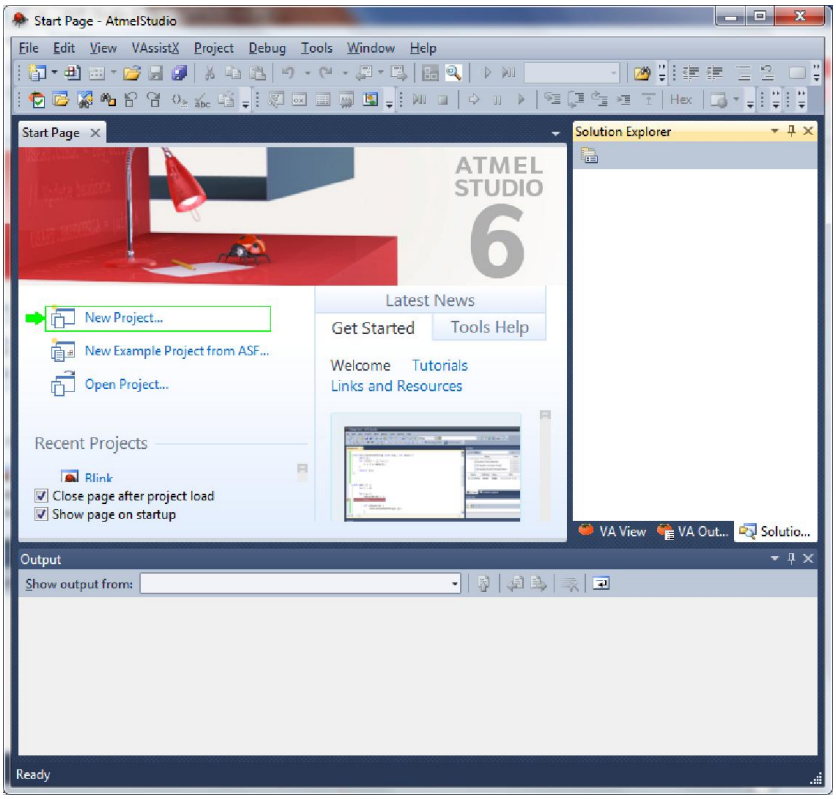


Click on Finish (Advanced programmers … please avoid associating .c/.h/.asm)

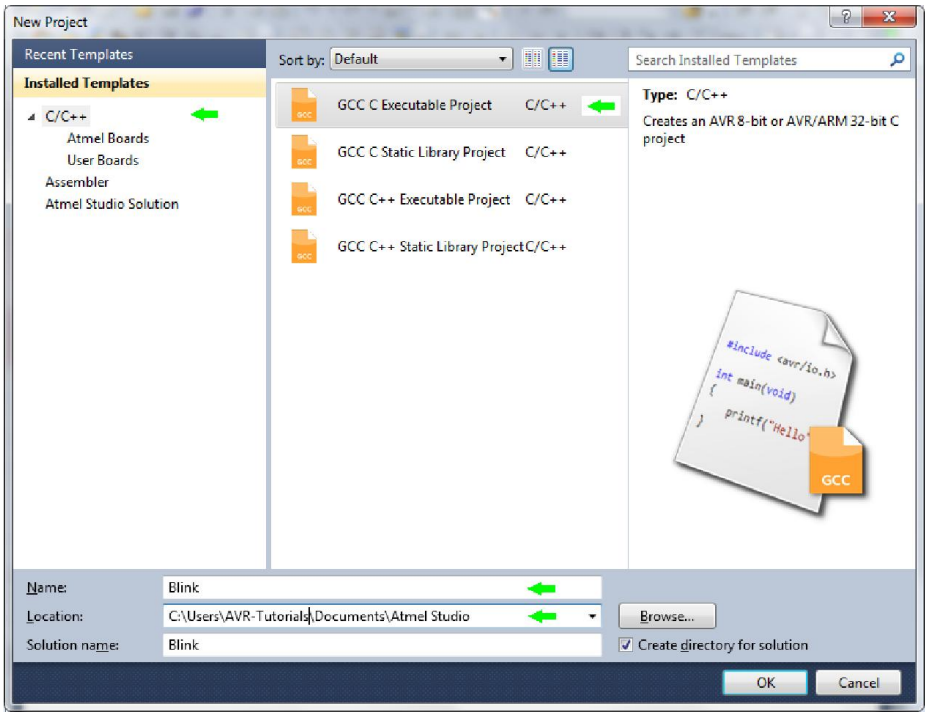
2.2 Creating a new C project in AVR Studio 6

Step 1: To create a C project first start AVR Studio 6 by going to the start menu on your PC select **Atmel AVR Tools** then **AVR Studio 6**. See the splash screen for AVR Studio 6, as shown figure below, this indicates that AVR Studio 6 is starting up. Just wait for a moment.

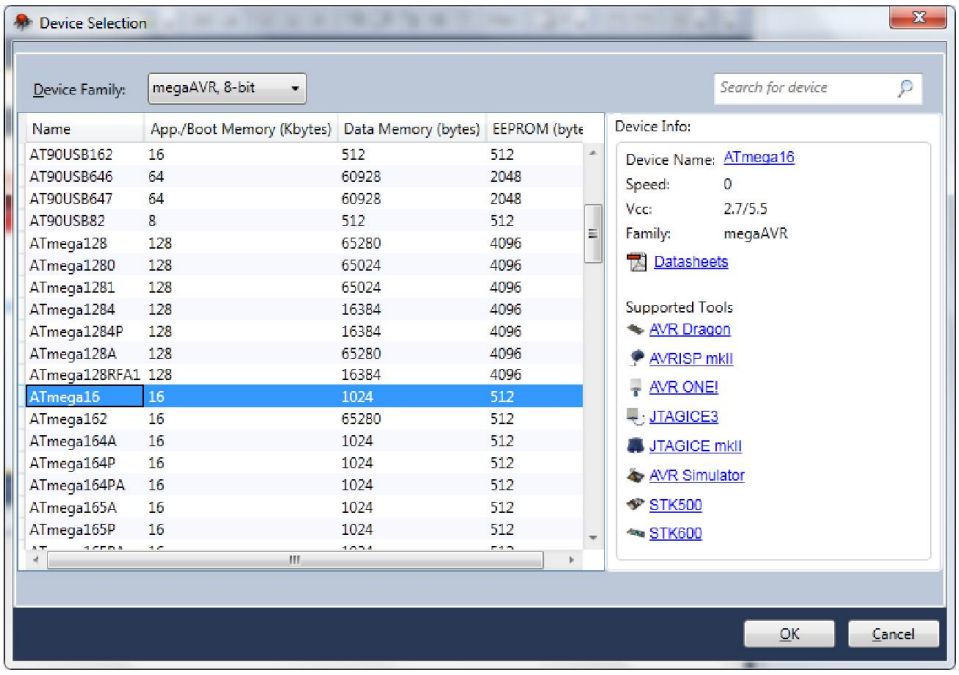
After AVR Studio 6 starts the AVR Studio 6 Start Page will appear as shown in the figure below.



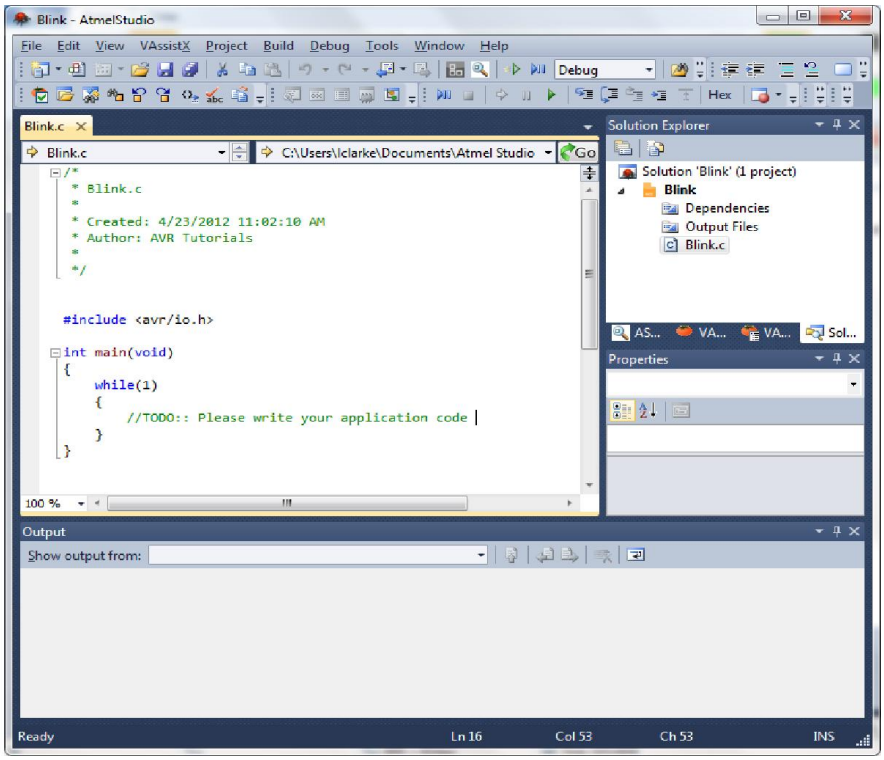
Step 2: Click on **New Project** pointed to by the green arrow in the diagram above to start a new project. The following window will appear.



Step 3: To start a new C project select **C** from the panel to the left pointed to by the green arrow. Also type the **Filename** and **Location** pointed to by the green arrows at the bottom of the window. The following window will then appear.



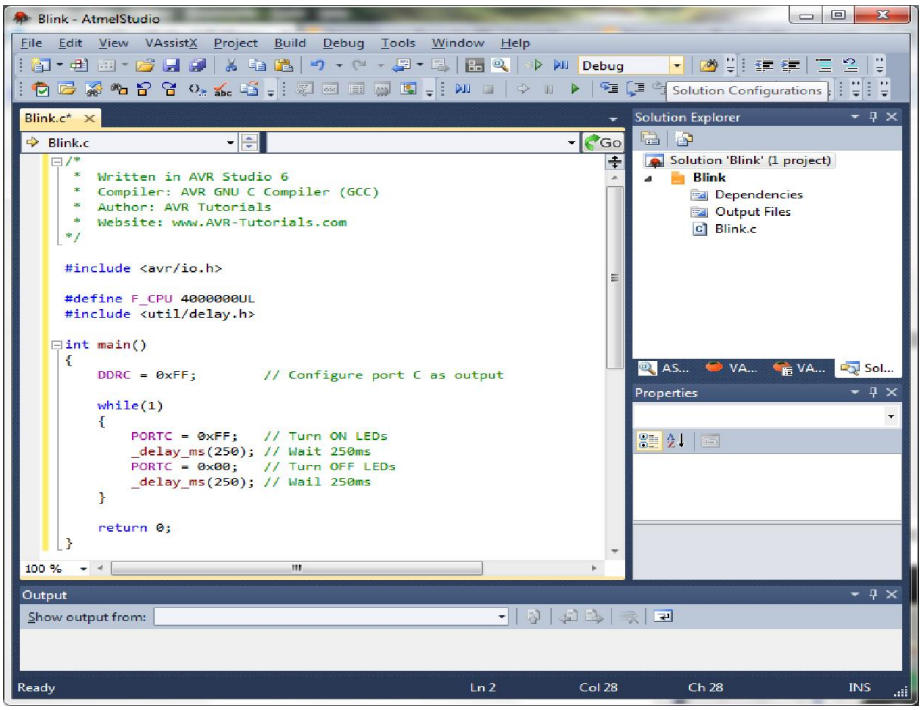
Step 4: The window above is the device selection screen for AVR Studio 6. Scroll down and select the microcontroller you will be using. The following window will then appear.



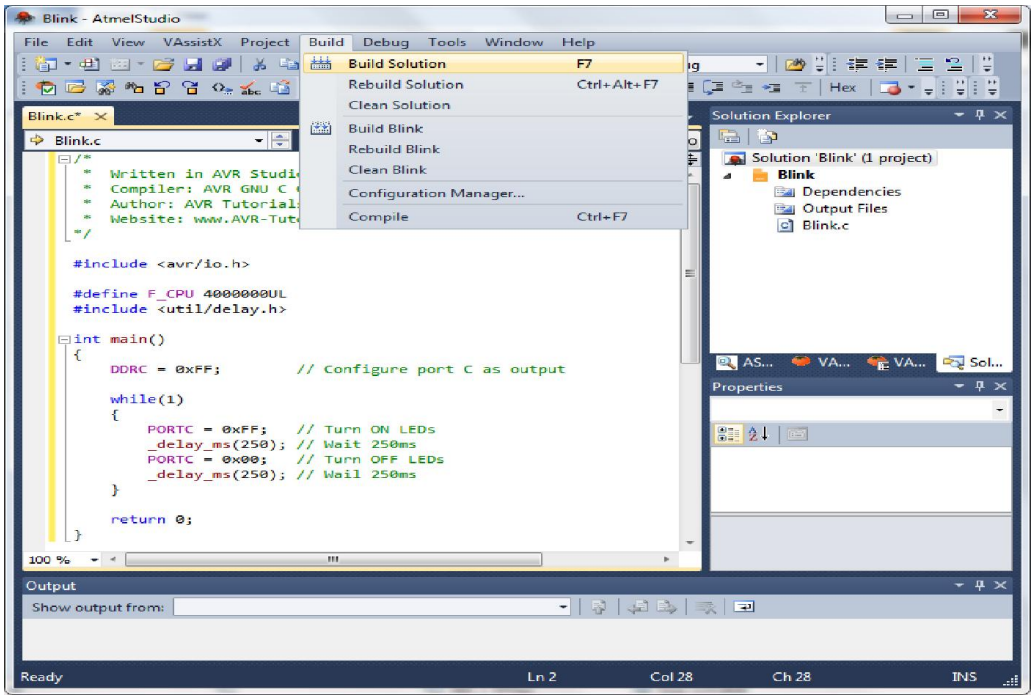
This is the AVR studio 6 editor where you type your C program. The editor starts your C program for you by providing you with the structure shown in the editor of the figure above.

2.3 Generating a HEX File in AVR Studio 6 (Compiling)

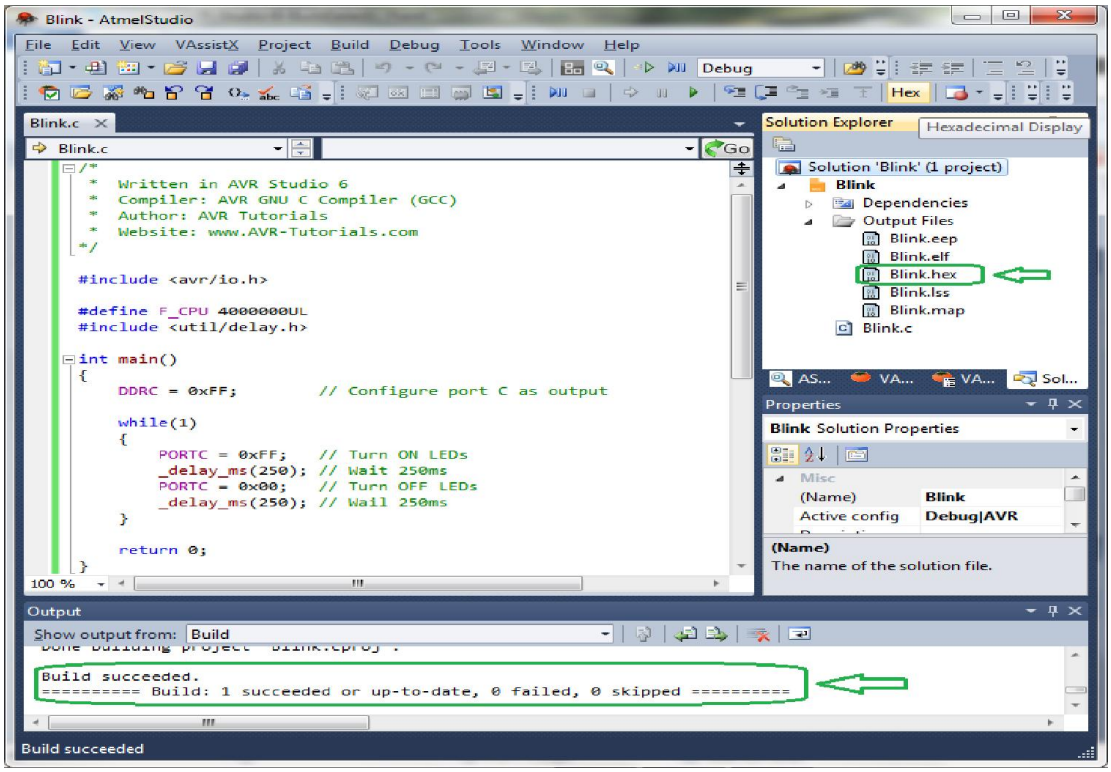
We will be generating/creating the hex file for the AVR C code shown in the AVR Studio 6 editor below.



After typing your AVR program may it be C or assembly to generate the **hex** file, go to **Build** menu and click **Build Solution**. See the figure below.



If the AVR C or Assembly program was build successfully a message will display at the bottom of the AVR Studio 6 Editor indicating that Build succeeded. See the bottom of the figure below. Also to the left of the editor if you expand the output folder you will see a file with a **.hex** extension. This is the actual HEX file that was generated.



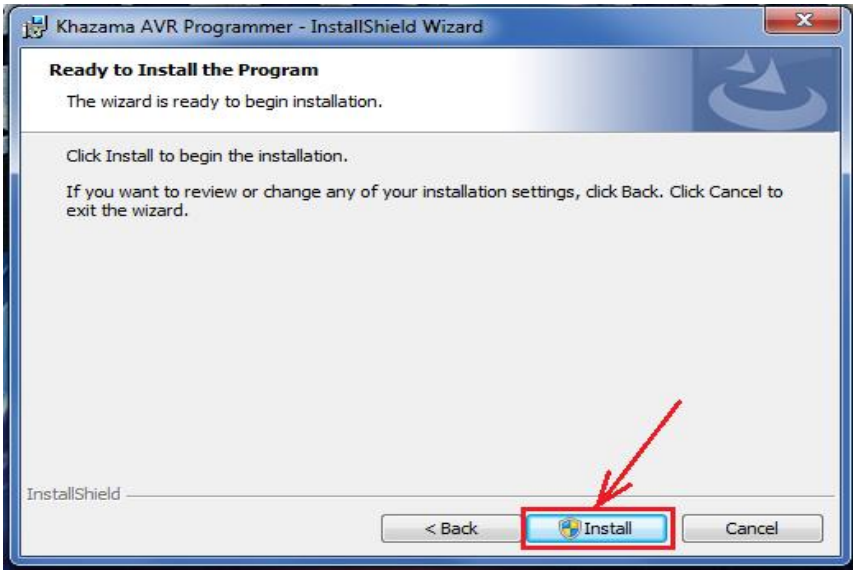
Chapter 3: Installing programming software and USB driver

Khazama is an AVR Programmer GUI which is an easy, fast, reliable and simple program to use for programming HEX files onto nearly all AVR Microcontrollers. Khazama programmer can be located in the CD provided with the KIT , ot you can download it from the following website : http://goo.gl/NEcGPz .

3.1 Installing Khazama Programmer GUI

Step1: Run the Khazma programmer setup file, follow the wizard for successful installation

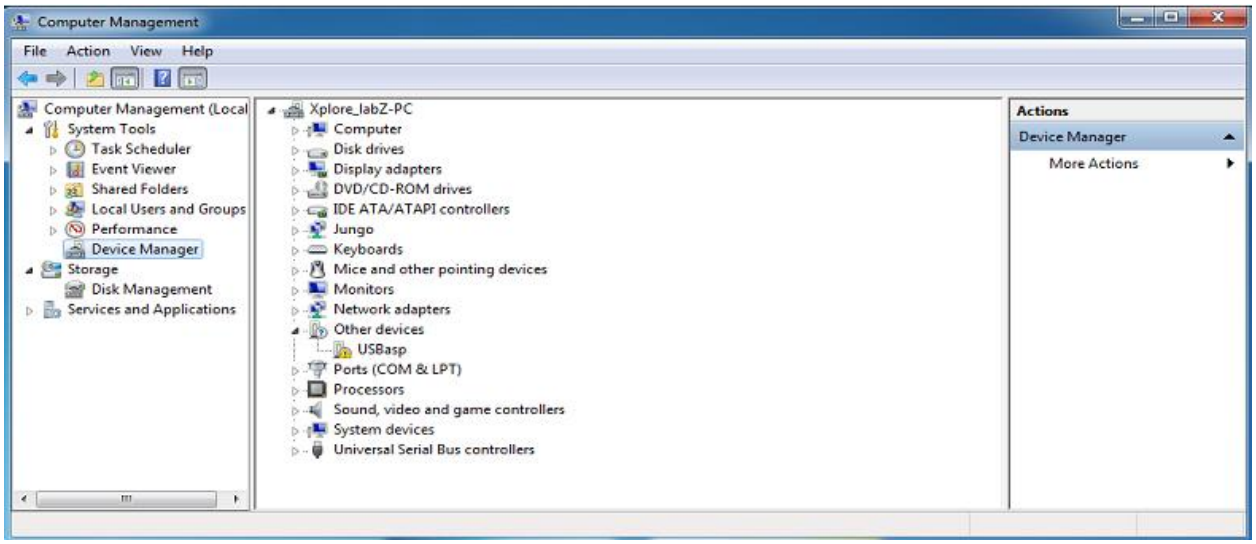
Step2: Click on Install then wait for installion to finish, and then click on Finish to close installation



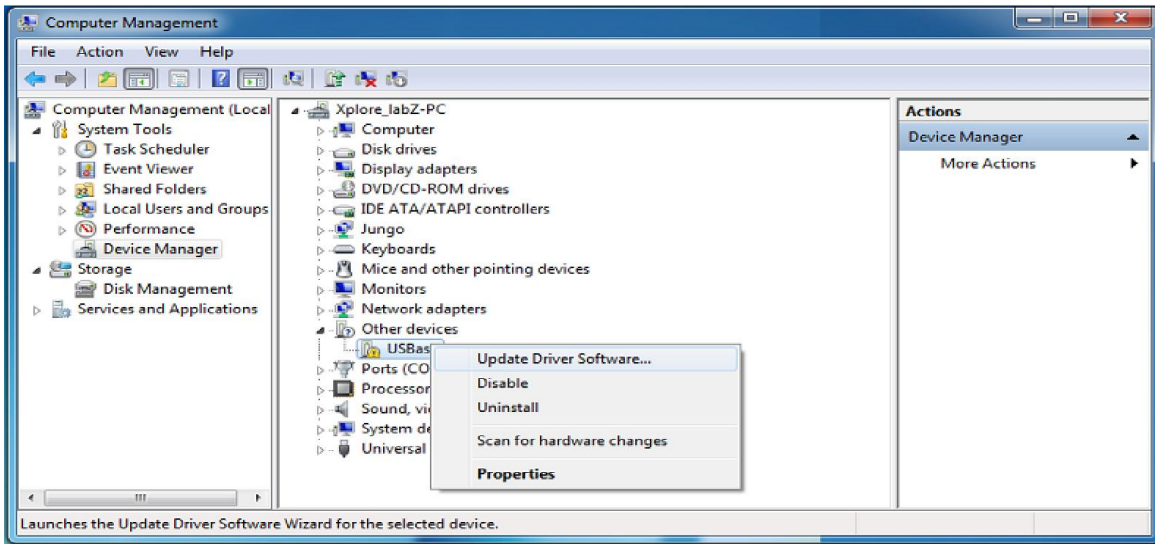
3.2 Installing USB Programmer (USBasp) Driver

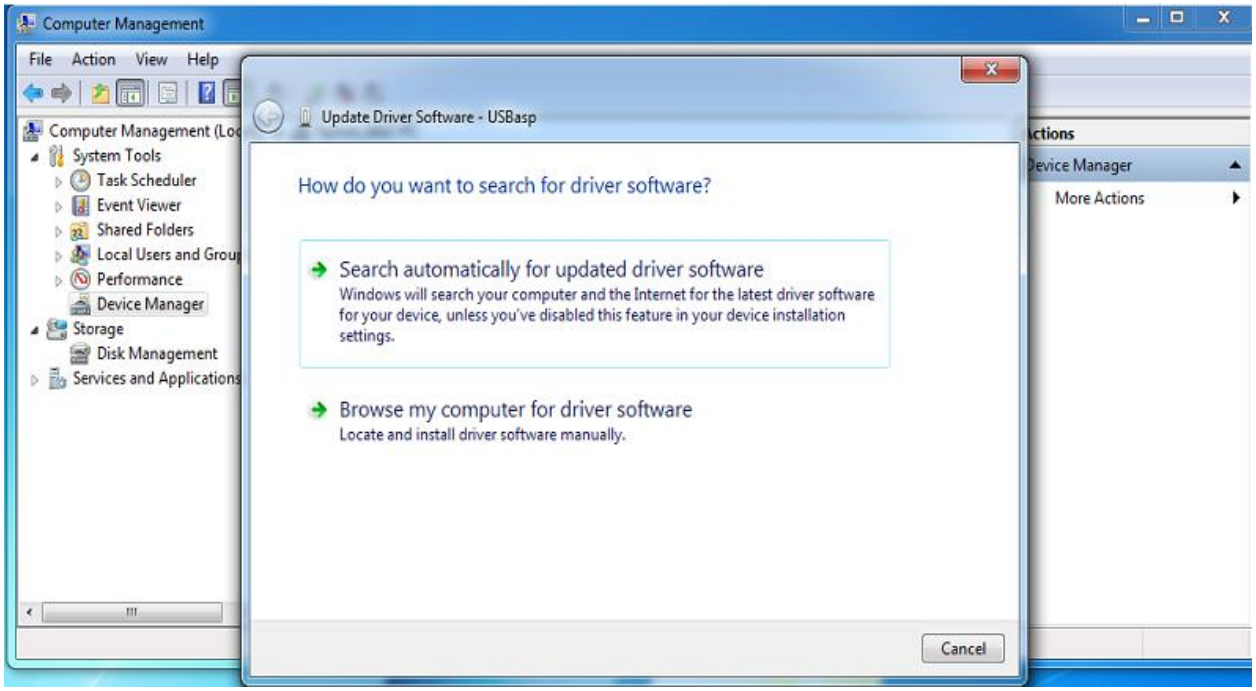
Step1: Connect the KIT using a USB (Type B) cable to a PC USB Port (Type A) . Since the USB driver has not been installed, below message will be displayed.

Step2: The device will be listed in other devices; it will be indicated by a yellow exclamation mark as shown below.



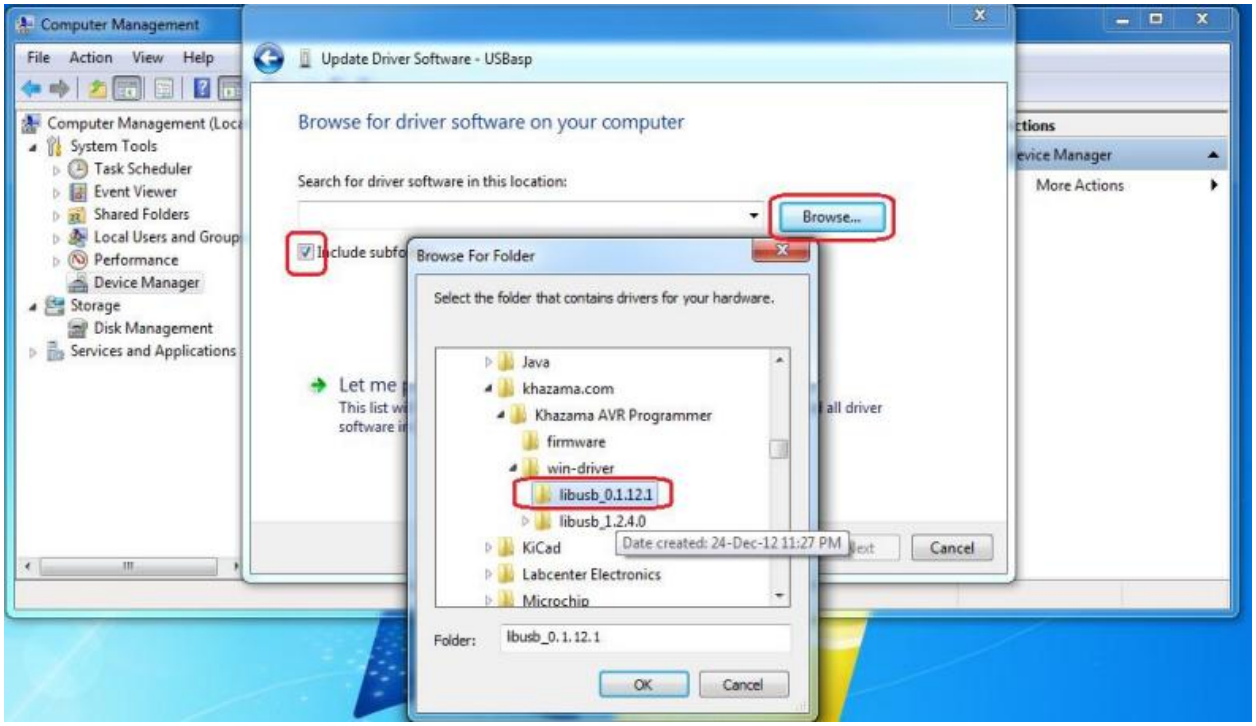
Step3: Before proceeding with driver installation please check if the Operating System is 32 bit or 64 bit so that correct driver can be installed. Update the drivers by browsing to the specific folder.



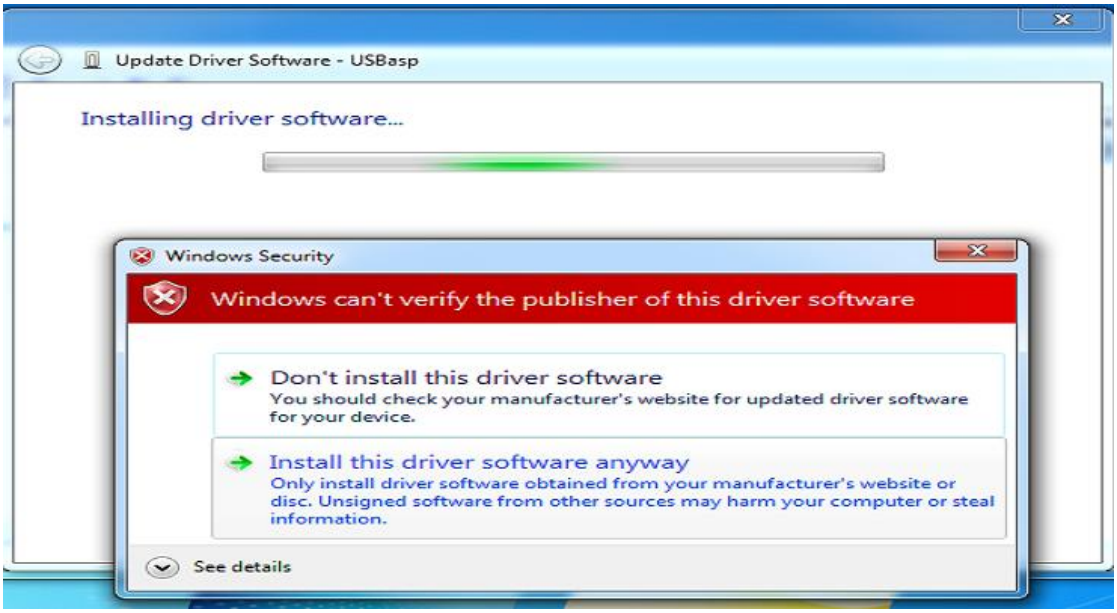


Step 4: The driver can be installed as shown in the figure below, the driver can be found in the installed files of Khazama Programmer as shown in the path below:

C:\Program Files\khazama.com\Khazama AVR Programmer\win-driver\libusb_0.1.12.1

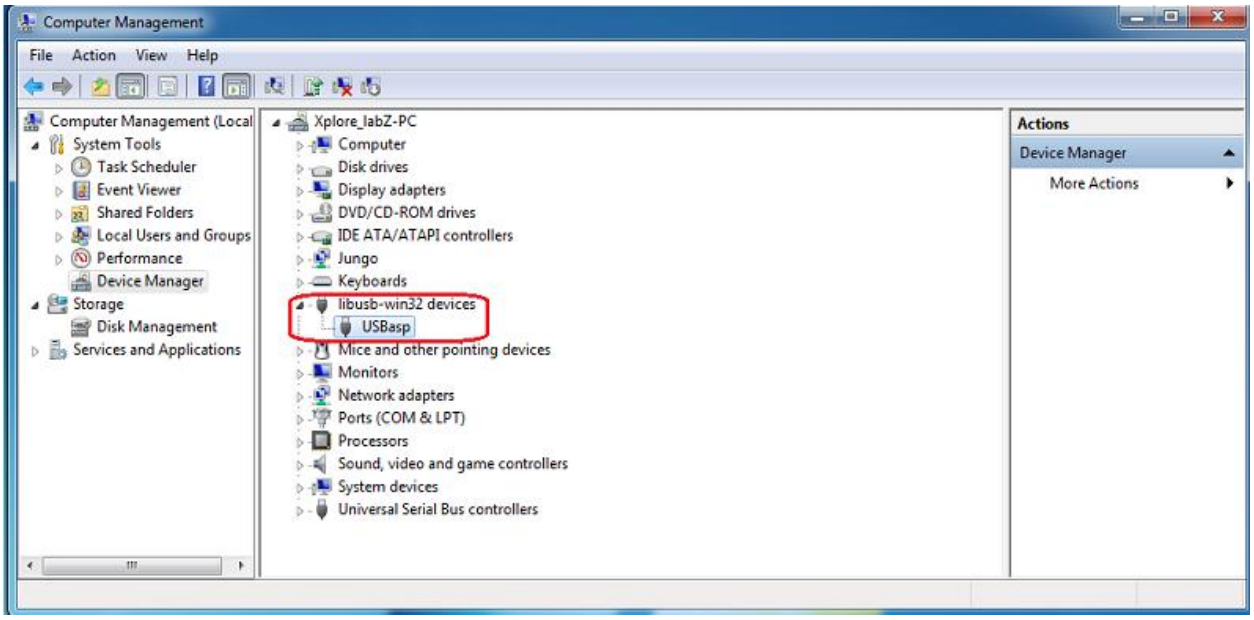


Step 5: Ignore the message and proceed with the installation



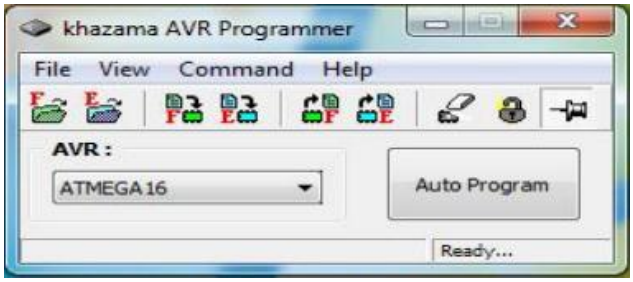
Step 6: The driver will be installed successfully as shown. Else check if the proper driver file was selected for the operating system installed.

Step 7: Verify the programmer installation in the device manager as shown below, now it would recognize the device LIBUSB devices.

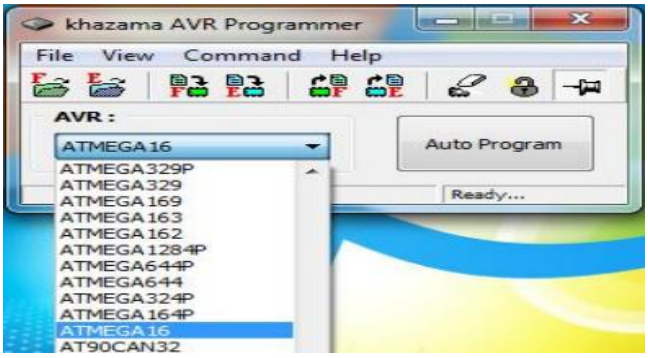


3.3 Flashing HEX File into the Microcontroller

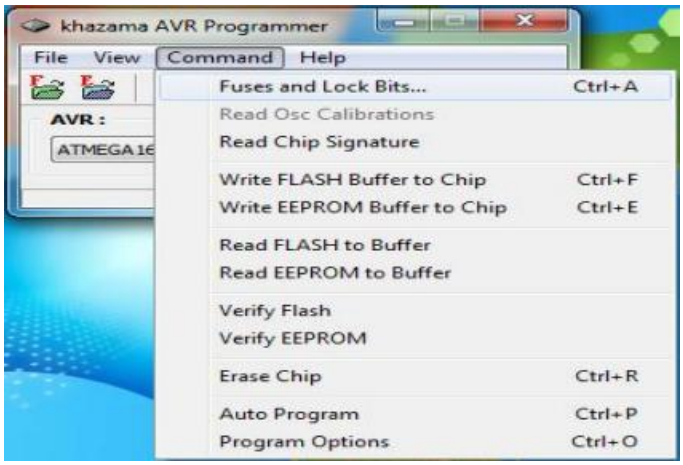
Step 1: Run the Khazama AVR Programmer.



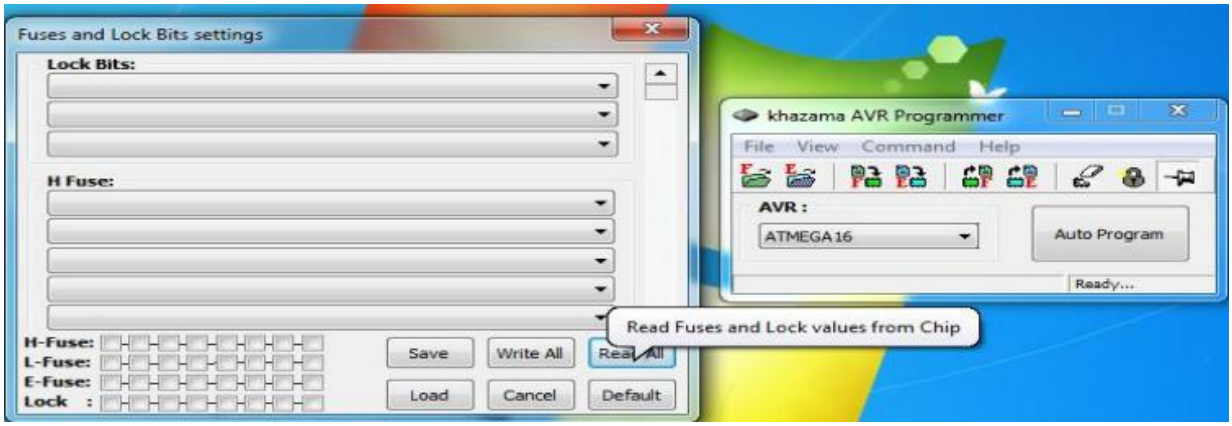
Step 2: Choose your target device (eg: ATMEGA16 from the drop down menu.



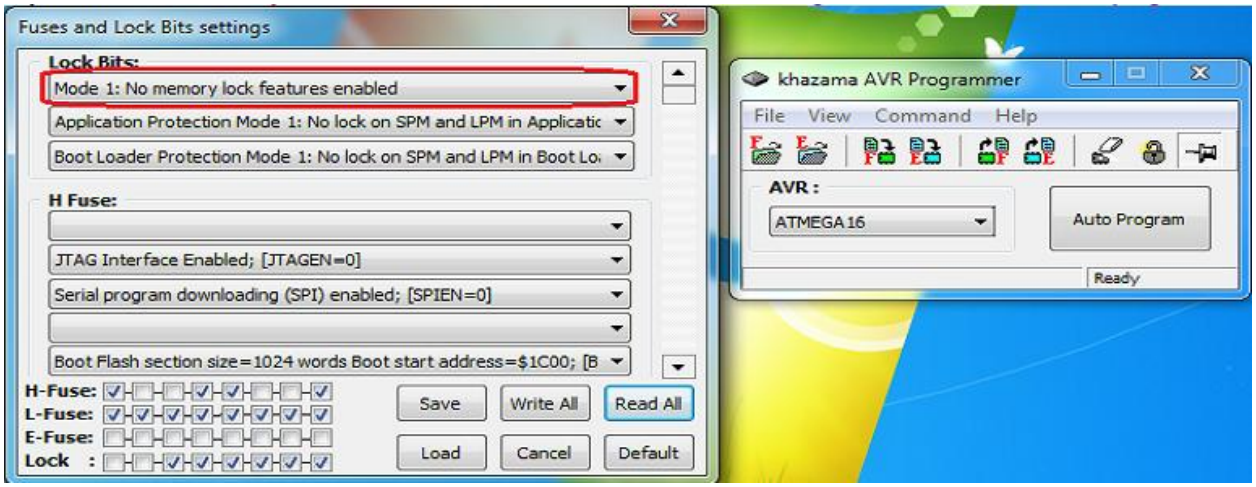
Step 3: For configuring the fuse and lock bits: Select the Fuses and Lock bits from the Command menu.



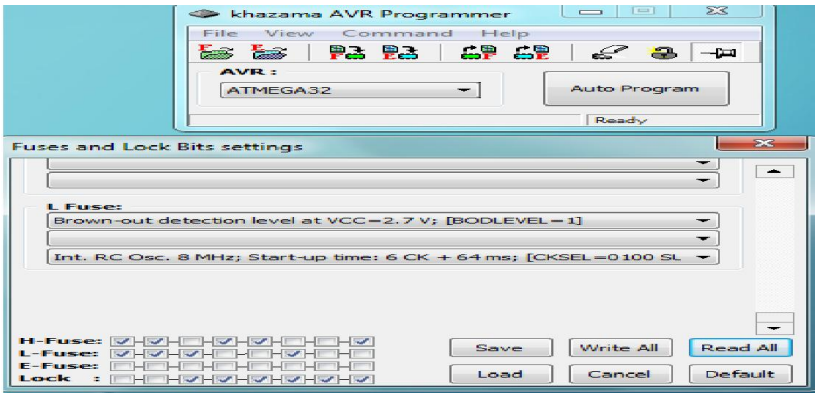
Step 4: A new window opens to read/write the fuse bits. Now read the fuse bits of the target. If the below window doesn't open, then double check your target IC is properly seated.



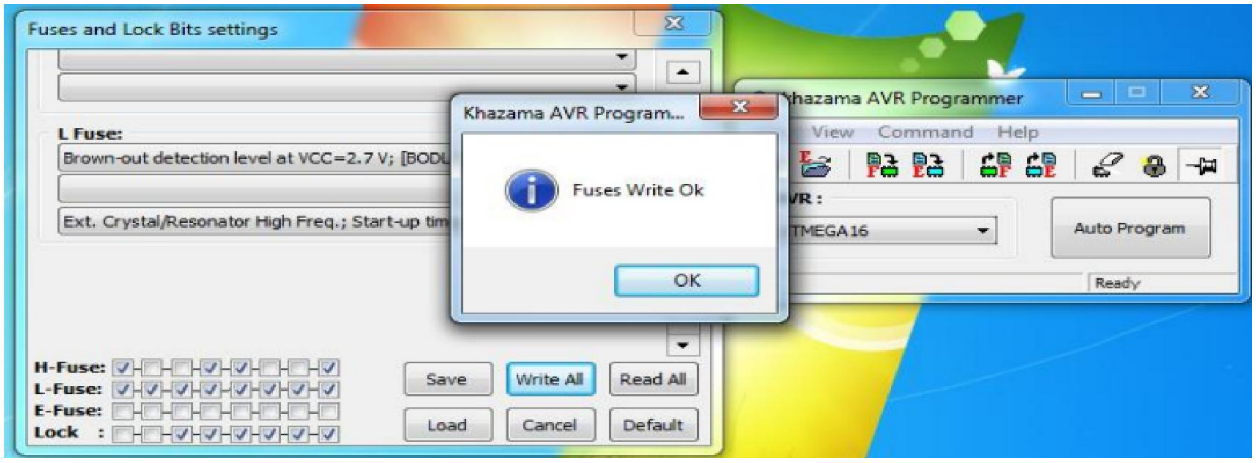
Step 5: **Mode 1: No memory lock Feature should be selected else the controller gets locked and cannot be reprogrammed !**



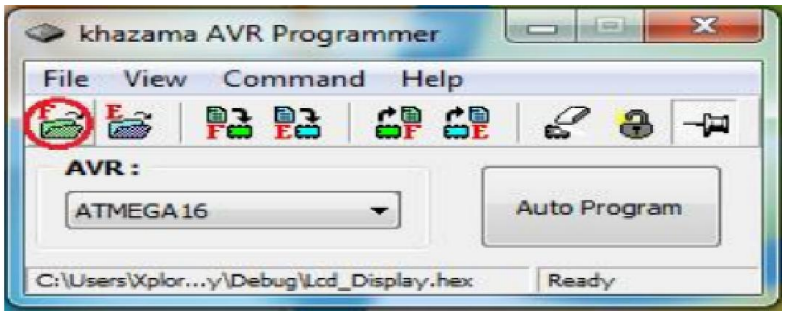
Step6: the Source of the clock should be selected from the drop down menu as shown below. Here the Int. RC Osc. 8 MHz is selected as the target device is running at 8 MHz freq.

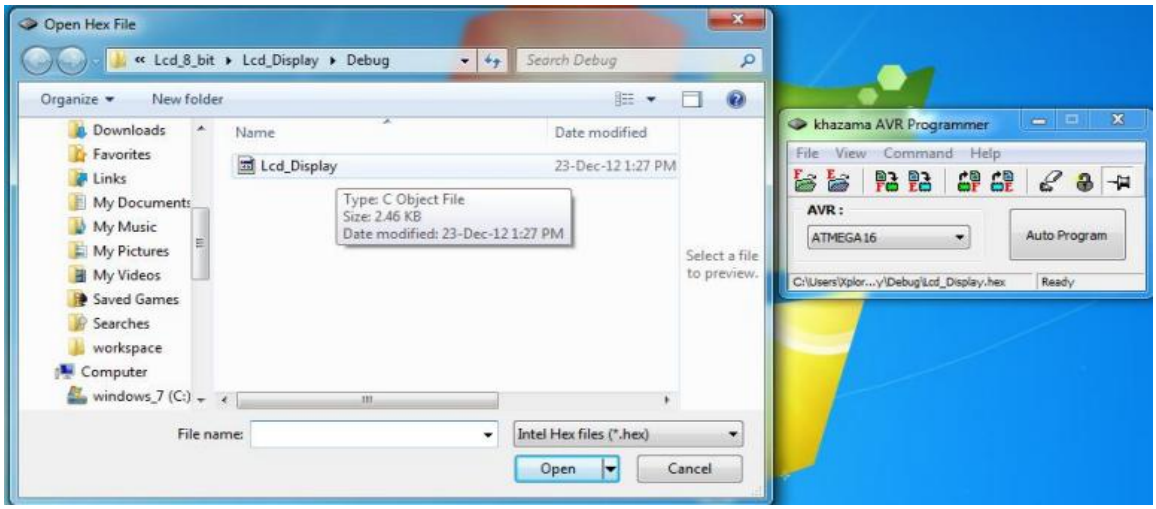


Step 7: After configuring the lock and fuse bits, write them back into the target controller. Double check the memory lock and crystal features as shown in step 4 and 5 before writing the fuse bits, if you want to save this setting as default ,click on Save.



Step 8: Browse and select the hex file that needs to be flashed into the target device.





Step9: Click the flash option as show below to program the target device or AutoProgram if you are using default or saved settings.

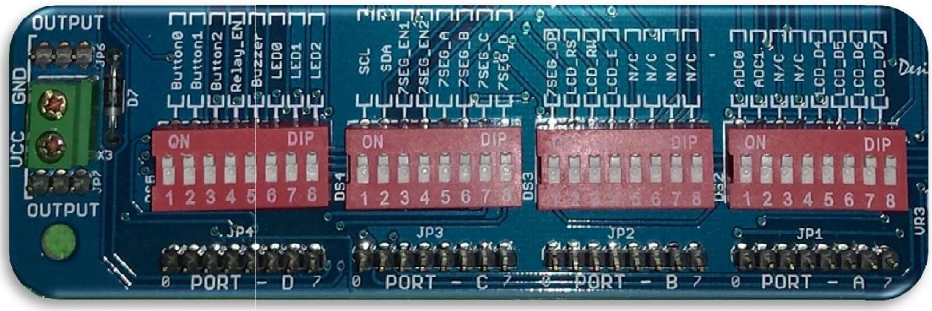
Step10: Once the target is programmed a new pop up window will be displayed to notify the same. And the controller starts executing and no external hard reset is required.

Chapter 4: ATmega Evaluation KIT System Functional Modules Details

This chapter will describe in more details the functional modules of the ATmega Evaluation KIT v1.0c System, by illustrating the schematics of the modules and any vital points that need to be noted.

4.1 General Input / Output Module

The General Input / Output Pins can be interfaced any external peripheral. Users can build their own circuit externally and interface theirs with the board's input/output pins; those pins are directly connected to the microcontroller's ports.



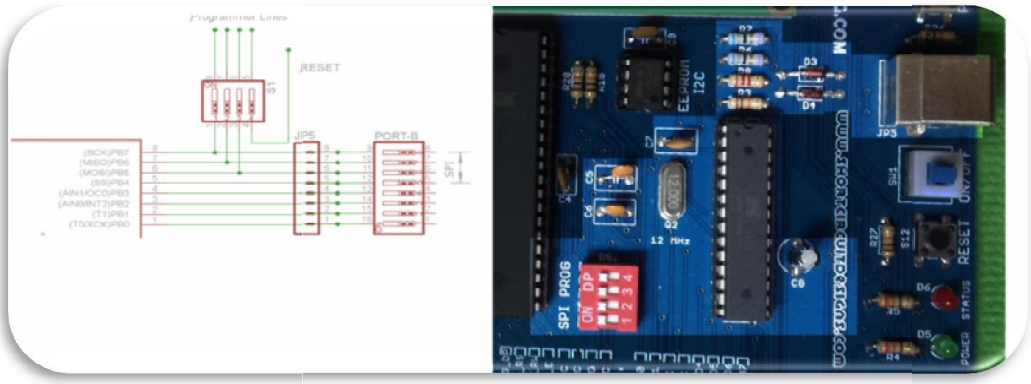
This module consists of the following main components:

1) 32 pin chip all I/O (PORTS A/ B/C/D)
2) Dip switches to enable or disable any built-in peripherals, isolate the I/O general port pin headers.
3) Power supply VCC and GND **[Output Terminal Only]** .

4.2 SPI Programmer Module

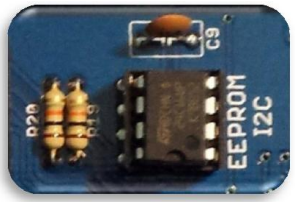
This module mainly shows the SPI Communication protocol between the following:

1) Built-in USB programmer and microcontroller's SPI pins [PORT-B.4, PORT-B.5, PORT-B.6 and PORT-B.7).
2) General I/O pin headers and the microcontroller's SPI pins [PORT-B.4, PORT-B.5, PORT-B.6 and PORT-B.7)



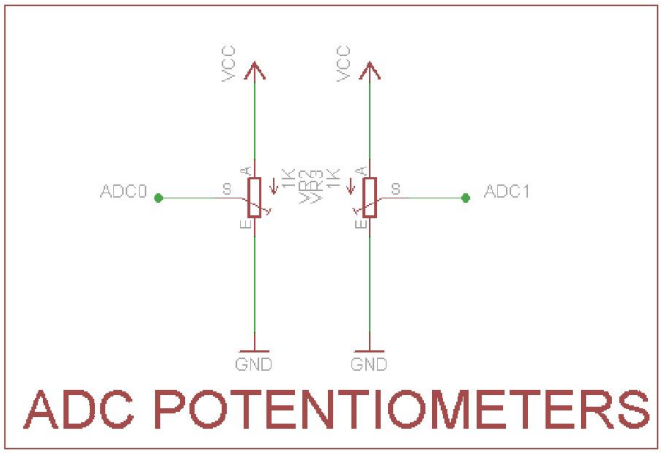
Note: Always make sure the 4way DIP switch is ON while Programming the target MCU, always close the DIP switch when not using the programming so as not to experience any malfunctioning.

4.3 I2C EEPROM Module



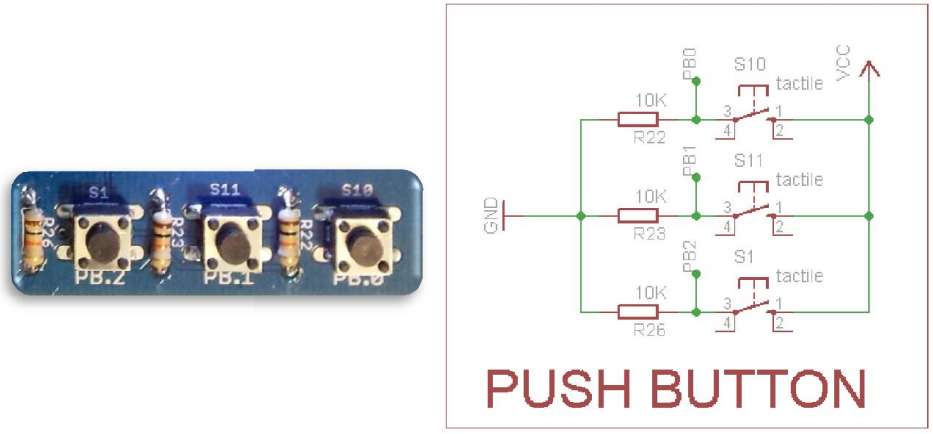
A 24C16 Atmel production EEPROM was chosen as an application for the I2C serial protocol. The communication with this module is via the two I2C lines which are SCL and SDA connected to PORTC.0 and PORTC.1 respectively. The 16K EEPROM enables the user to store and read data to and from the module by properly implementing the I2C protocol. For more details on how to communicate with the chip, please refer to the manufacturer's datasheet.

4.4 A/D Converter Module



Two Separate potentiometers are connected to PORTA, by changing the value of the resistance the input voltage value varies , allowing the user to manipulate the ADC accordingly. AREF is equipped with noise canceling (decoupling capacitor) in order to achieve stable internal reference voltages. VREF can be selected as either AVCC (5V), internal 2.56V reference. General I/O pin header at Port A can be used as Input for ADC.

4.5 Triple Pushbuttons Module

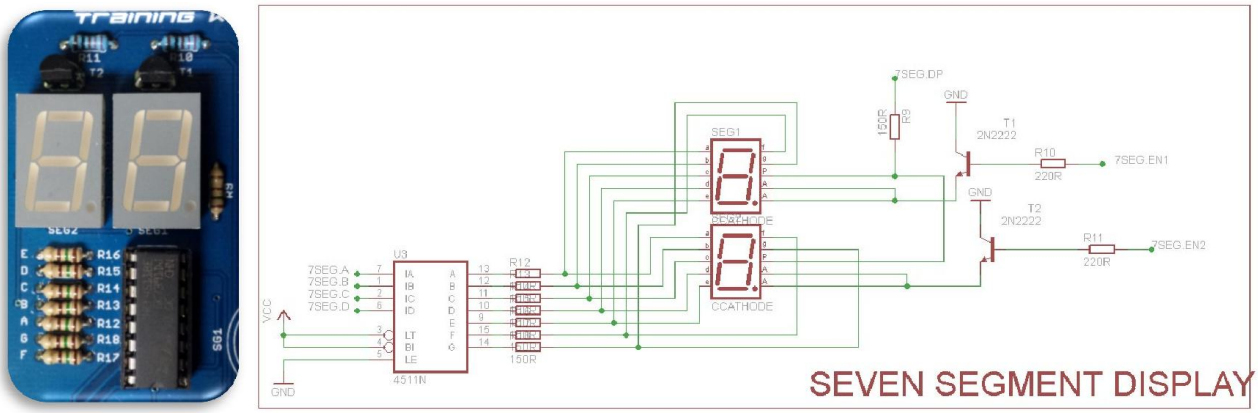


Three separate push buttons are connected to PORTD, they are active **high**. The figure below illustrates how the push buttons are connected and located.

4.6 16x2 Character LCD Module

The LCD module is connected to the MCU in 4-bit mode, data and control lines are connected as shown in figure below. There's a backlight switch on the upper left corner of the board which is used to switch on/off the light of the screen. A potentiometer as shown in the figure below is used to adjust the contrast of the lcd when needed.

4.7 Two- digital Display Module



This module consists of the following major components:

1) 7 segment display.
2) Enable Circuit.
3) BCD Driver.

Description of this module is as follows:

1) The MCU's data lines are connected to the display units through the Driver IC; both are commonly connected to most significant part of PORTC.
2) The enable of the 7-segment is connected to the PORTC of MCU.
3) The two digit 7segment displays are common cathode; the decimal point (7SEG_DP) is active **high** because it is the only pin from the 7segment display having a direct connection to the MCU pins, on the other hand, the data lines of the display are connected first to the BCD decoder (4511) then to the MCU, data sent from the MCU us always decimal then the decoder converts them into output binary.

The schematic of this module is shown in figure above.

4.8 3-bit LED Module

This module consists of 3 parallel LEDs; all of them are connected to PORT-D.

4.9 Buzzer Module

Description about this module is as follows:

1) The buzzer is connected to the PORT D of MCU, figure below shows the connection. This pin is also a PWM so application may depend on your output wave.

4.10 Relay Module

The relay onboard operates at 5V and is connected to the microcontroller through an NPN transistor which enables/disables the coil of the relay that in return deflects the blade of the relay, where the Common pole is stationary. When the coil in the relay is enabled, the Common changes its connection either to the NC or NO. The 3-pin Screw Terminal can handle around 6 Amperes of load current due to the width of PCB tracks at the 3-pin screw terminal. The Figure above illustrates the connections performed.