HOW TO PROGRAM MCU?

Rudra Pratap Suman

Flash Back (Takneek/Techkriti)

- Line Following Robots
- Wireless keyboards
- Wireless Gamepad
- Tachometer
- They were made using Microcontrollers

ELECTRONICS CLUB
IITKANPUR

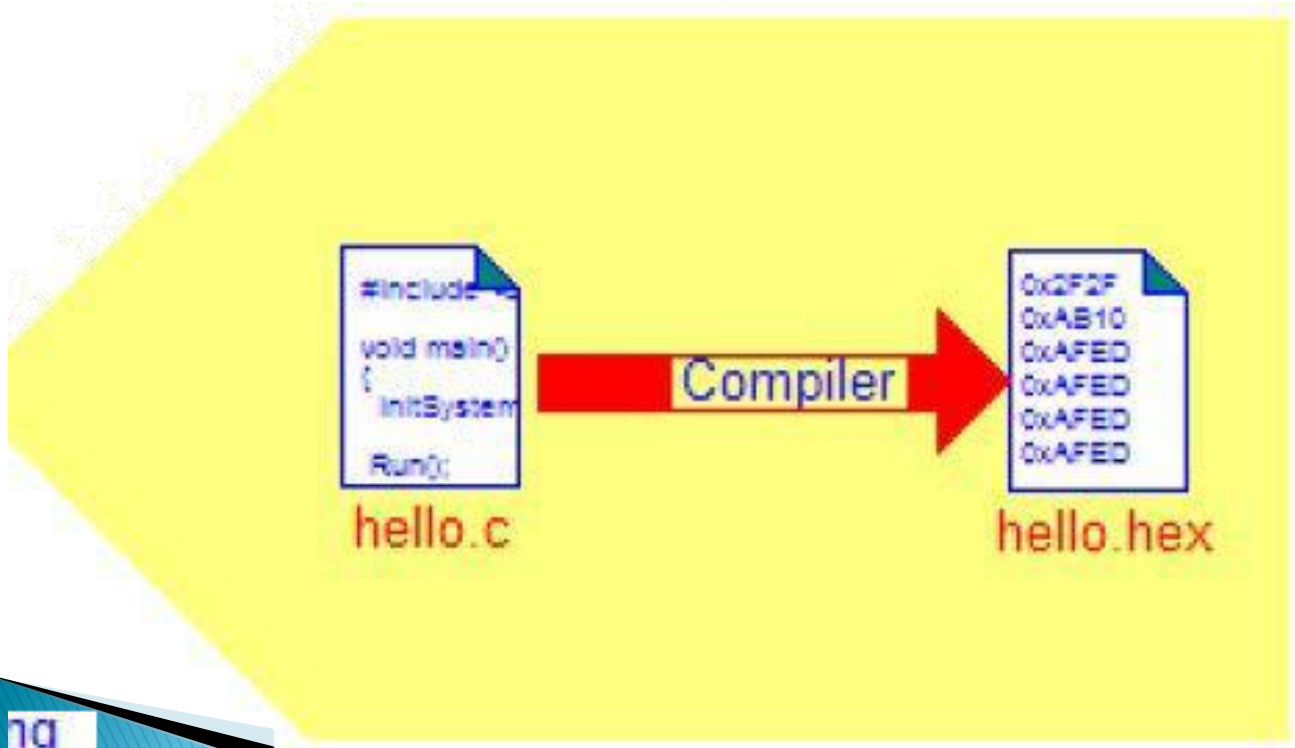
Tools Required -> CVAVR

PC Running IDE for entering,editing and compiling source program.

ELECTRONICS CLUB IITKANPUR

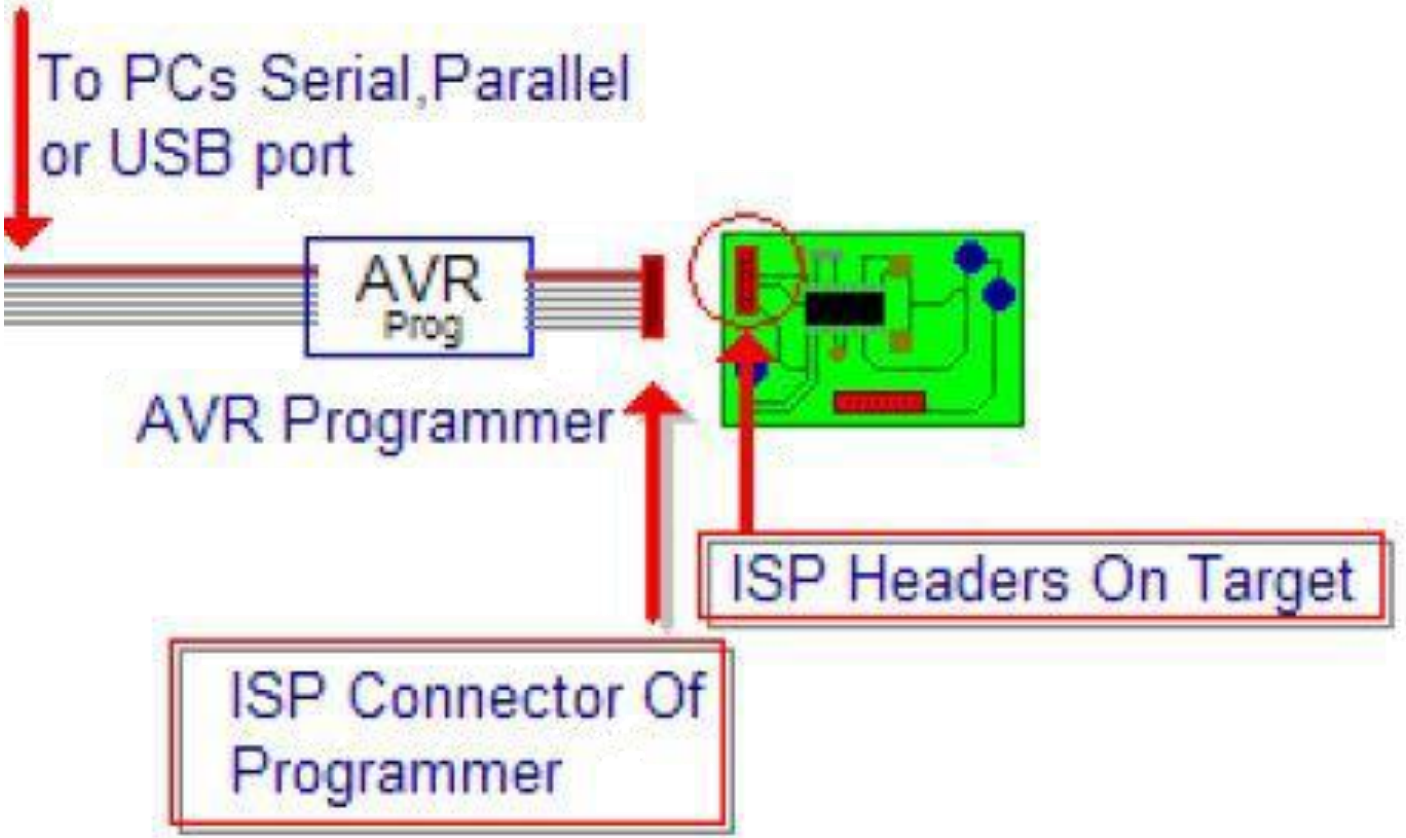
Compiler -> CVAVR

- The code is written in C language so we need to convert it into the format that Atmega understands



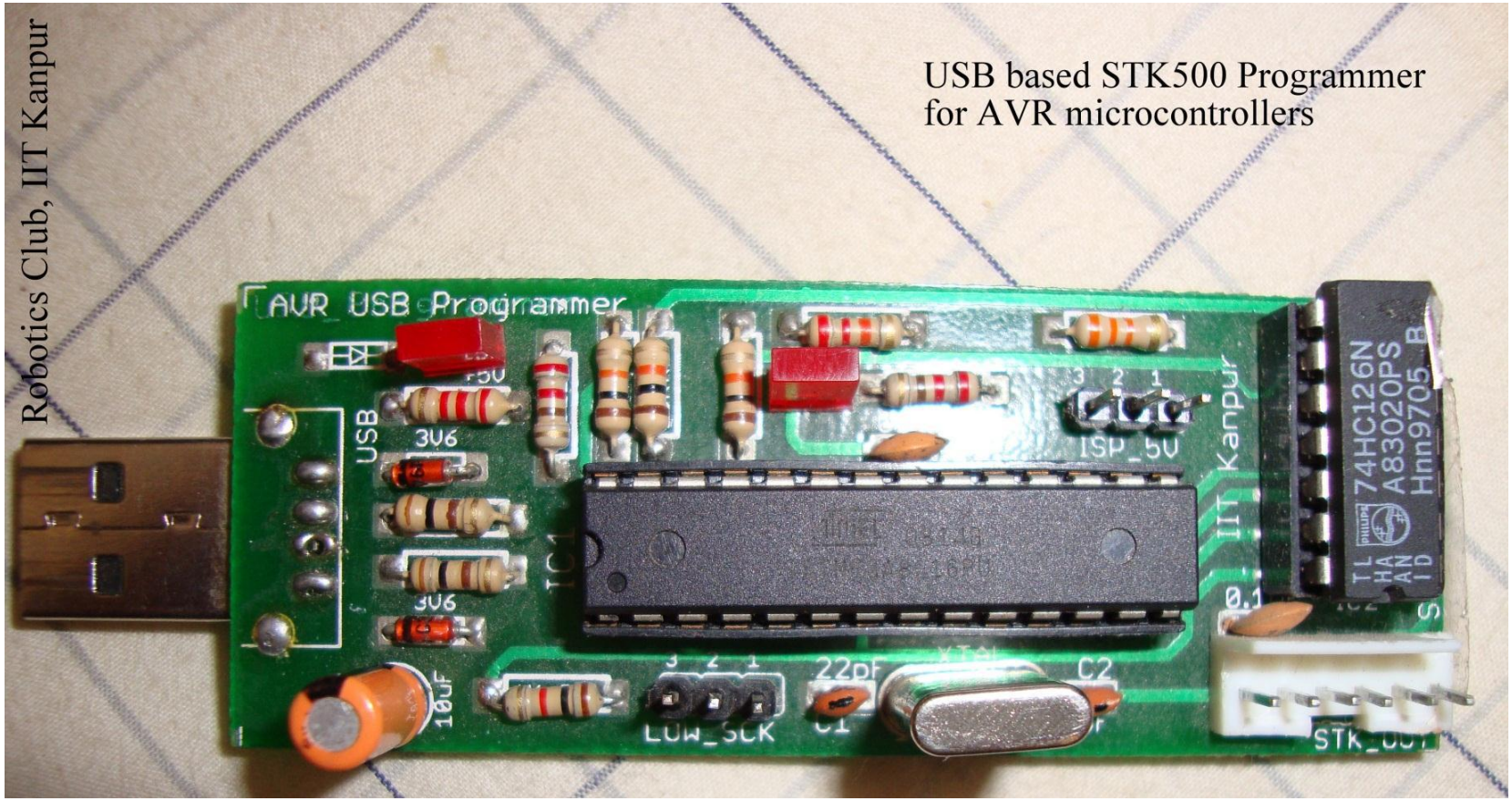
ELECTRONICS CLUB IITKANPUR

Transfer code to Atmega AVR Studio



ELECTRONICS CLUB IITKANPUR

Avr Programmer



ELECTRONICS CLUB IITKANPUR

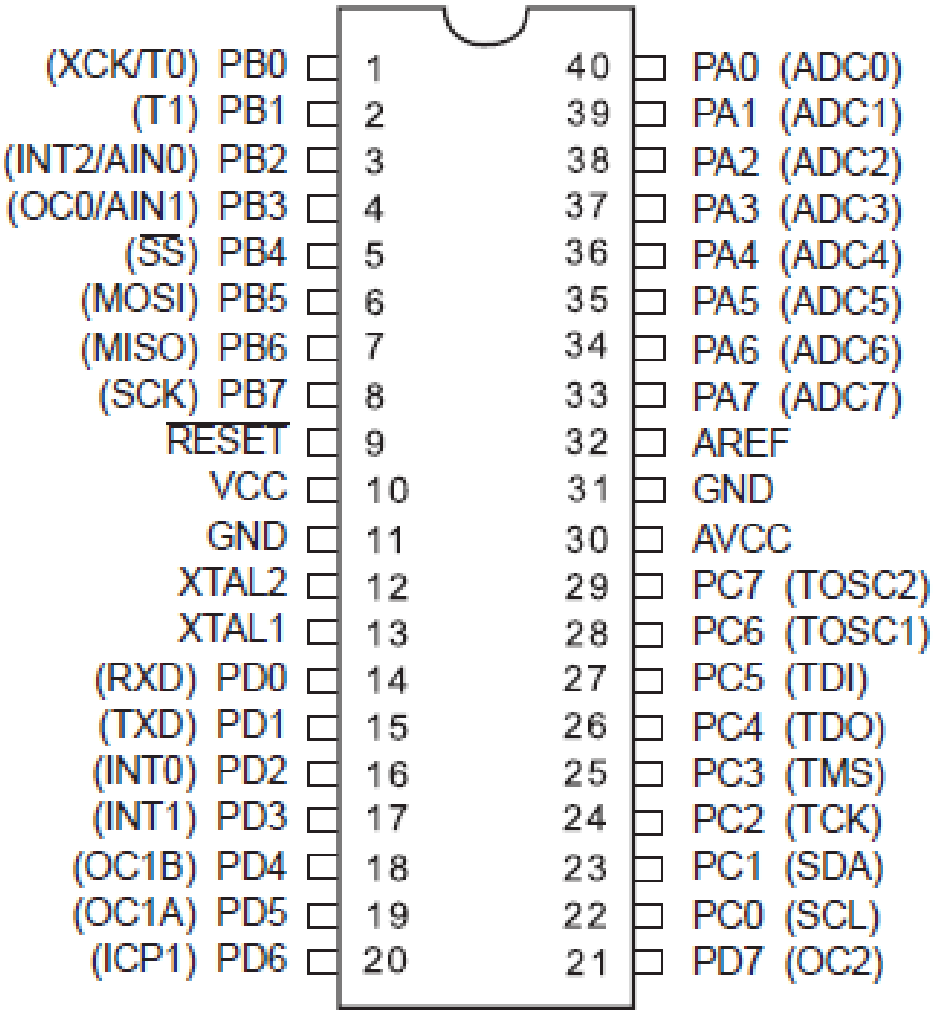
- So we need two softwares overall

a) CVAVR -> Editor and Compiler

b) Avr Studio -> Transfer Code to Atmega

The ATmega16

- 40 pin IC.
- 32 pins for I/O.
- 8 pins reserved.
- I/O pins divided into 4 groups of 8 pins, called ports.
- Ports labeled as A, B, C and D.



ELECTRONICS CLUB IITKANPUR

Basics of C language

- If else block
- If(condition)

```
{
…
  …
 }
 else
{
…
  …
}
```

While & For

- While (conditon)

{

…

…

}

- for(initialisation; condition; increment)

{

…

…

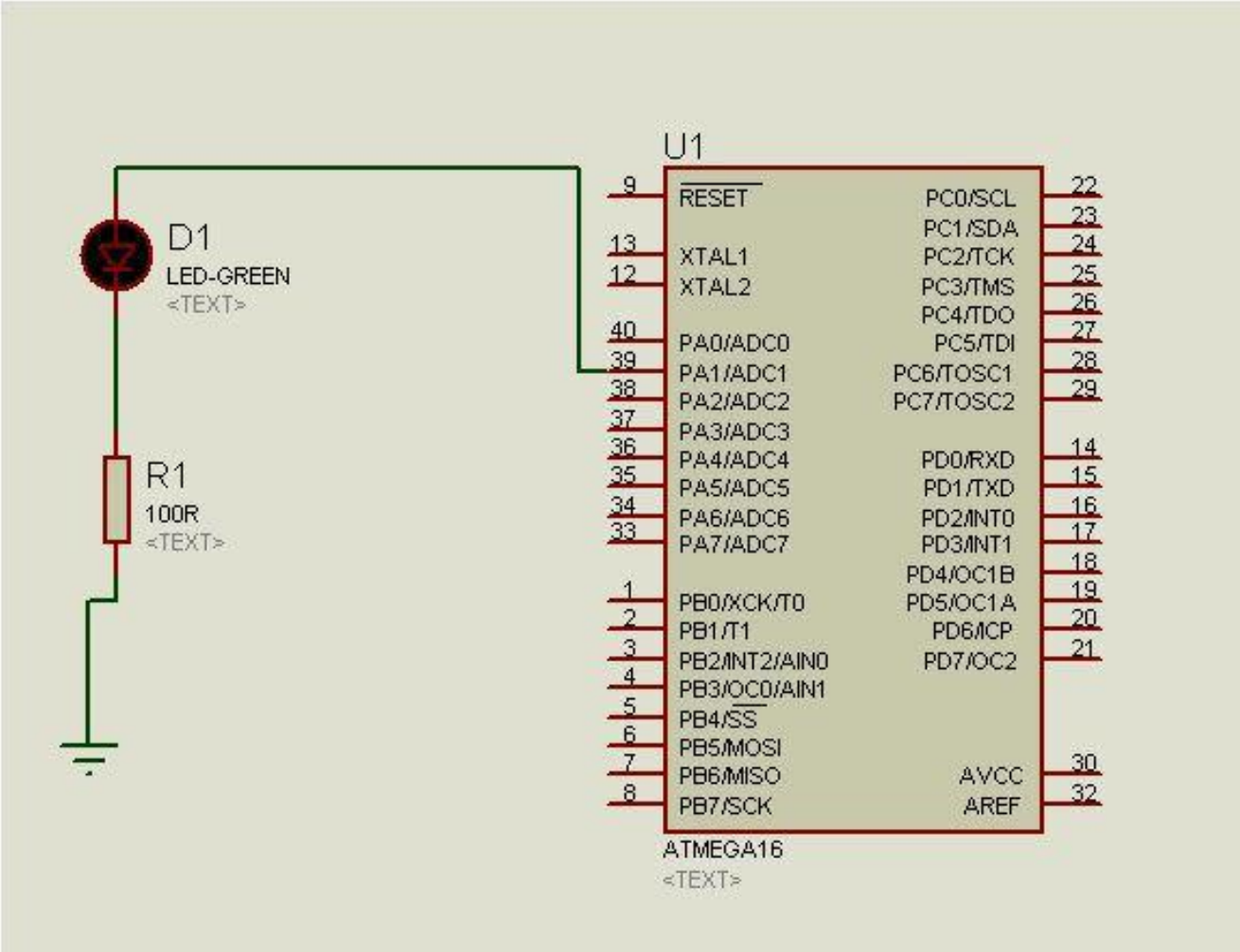
}

Some C operators

- | is bitwise OR. Eg. 10100111 | 11000101 = 11100111
- & is bitwise AND. Eg. 10100111 & 11000101 = 10000101
- ~ is bitwise NOT. Eg. ~10100110 = 01011001
- << is shift left. >> is shift right.

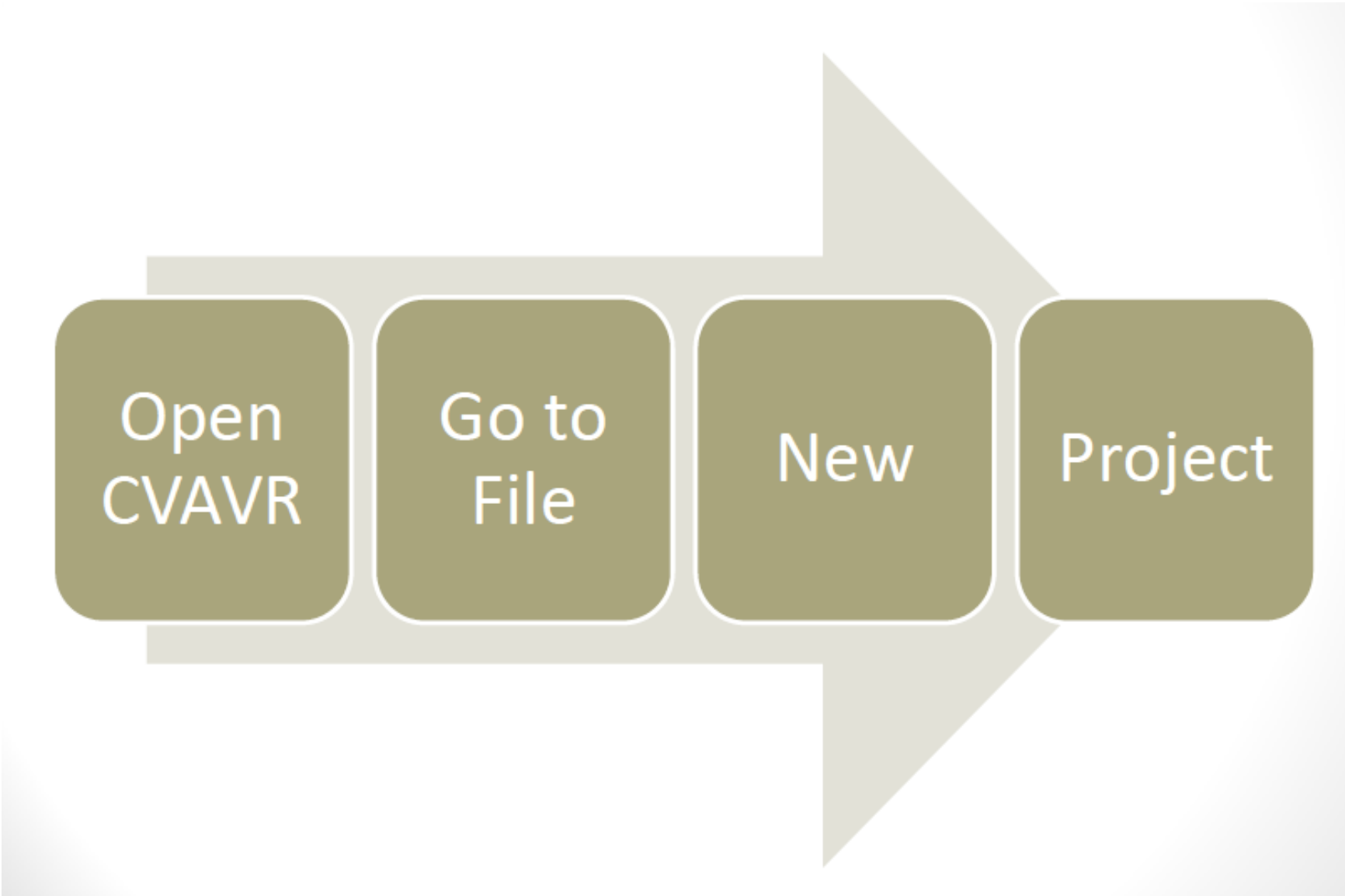
Lets Begin by blinking a simple LED

Circuit Diagram



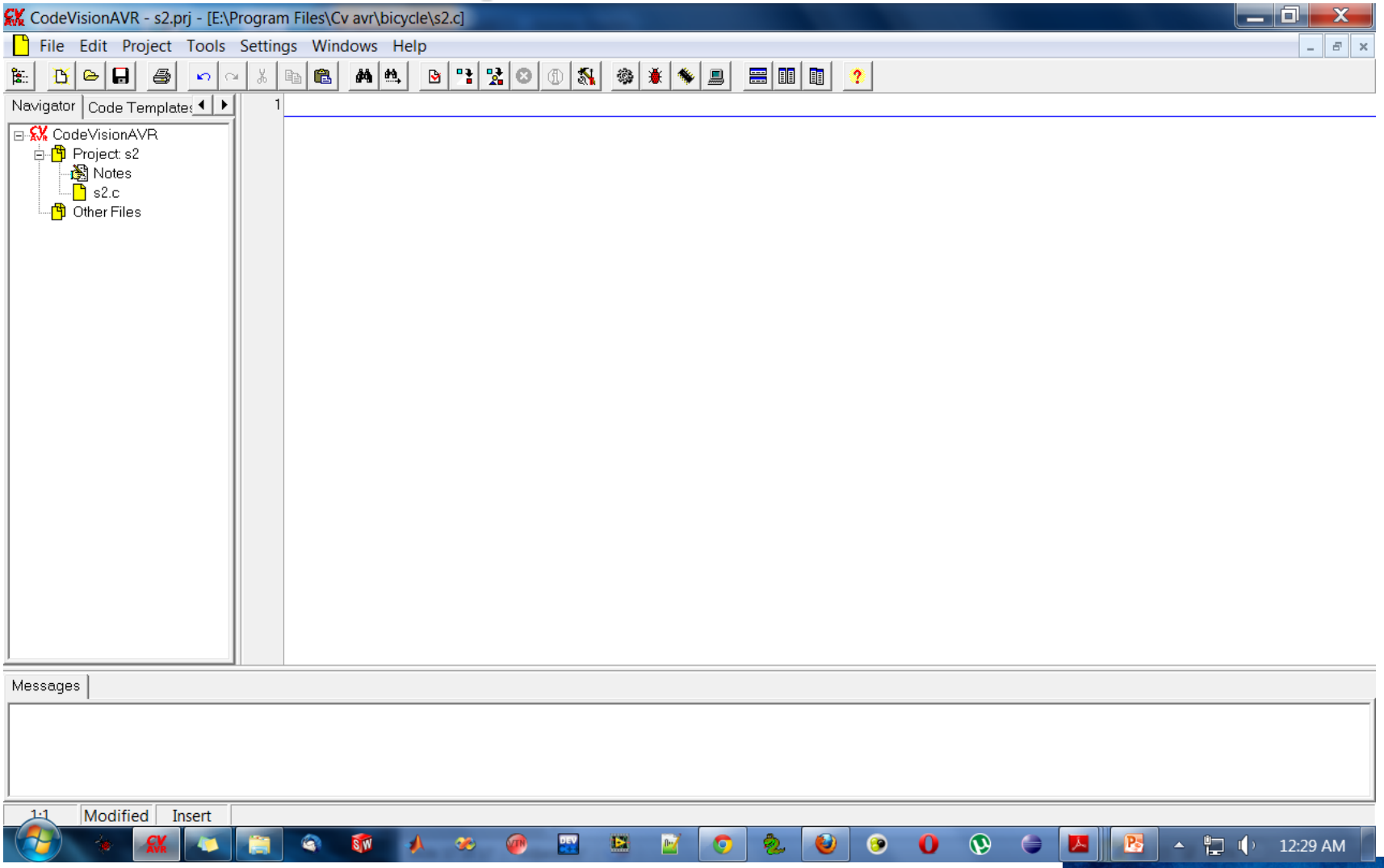
ELECTRONICS CLUB IITKANPUR

Getting Started with CVAVR



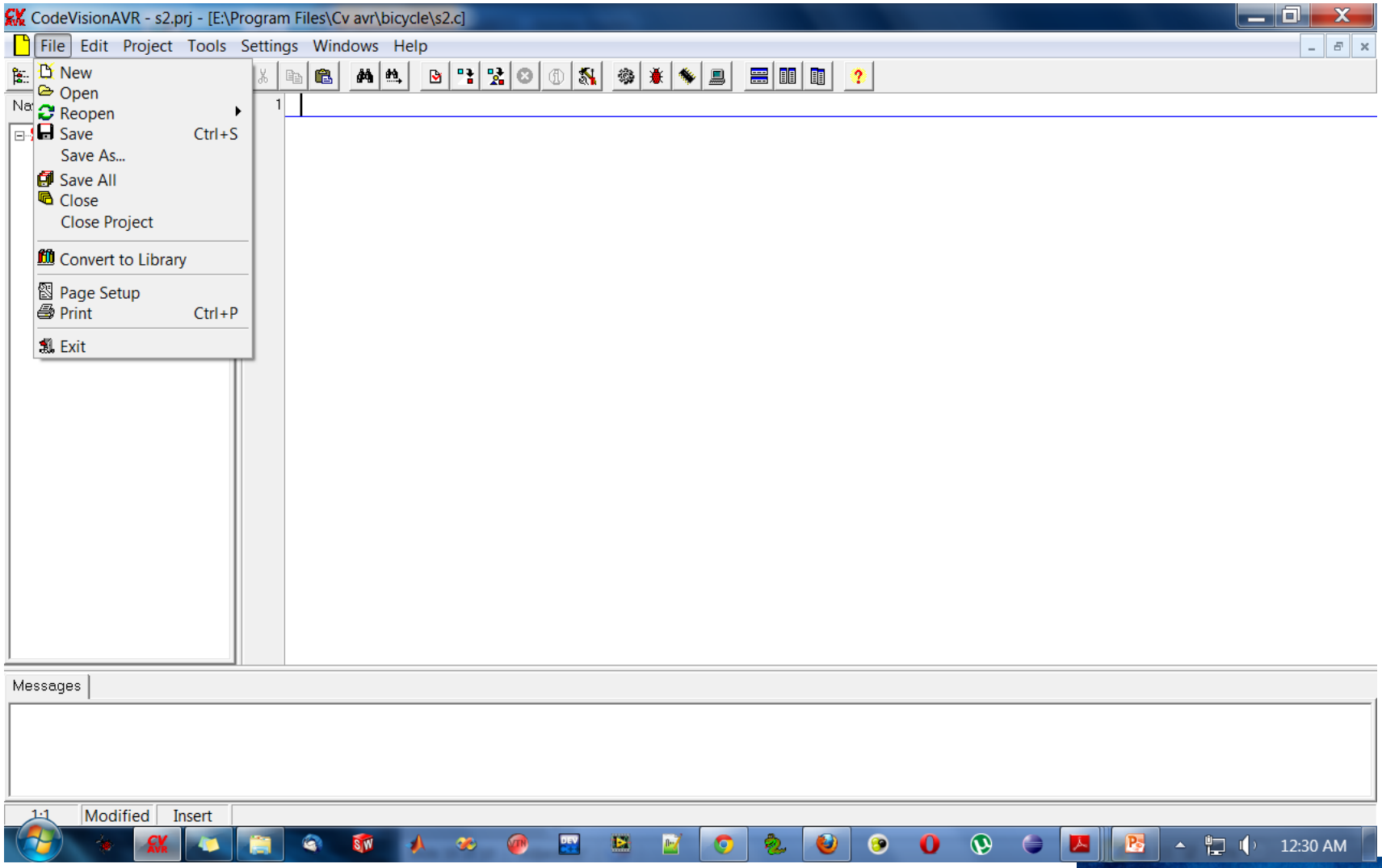
ELECTRONICS CLUB IITKANPUR

Open CVAVR



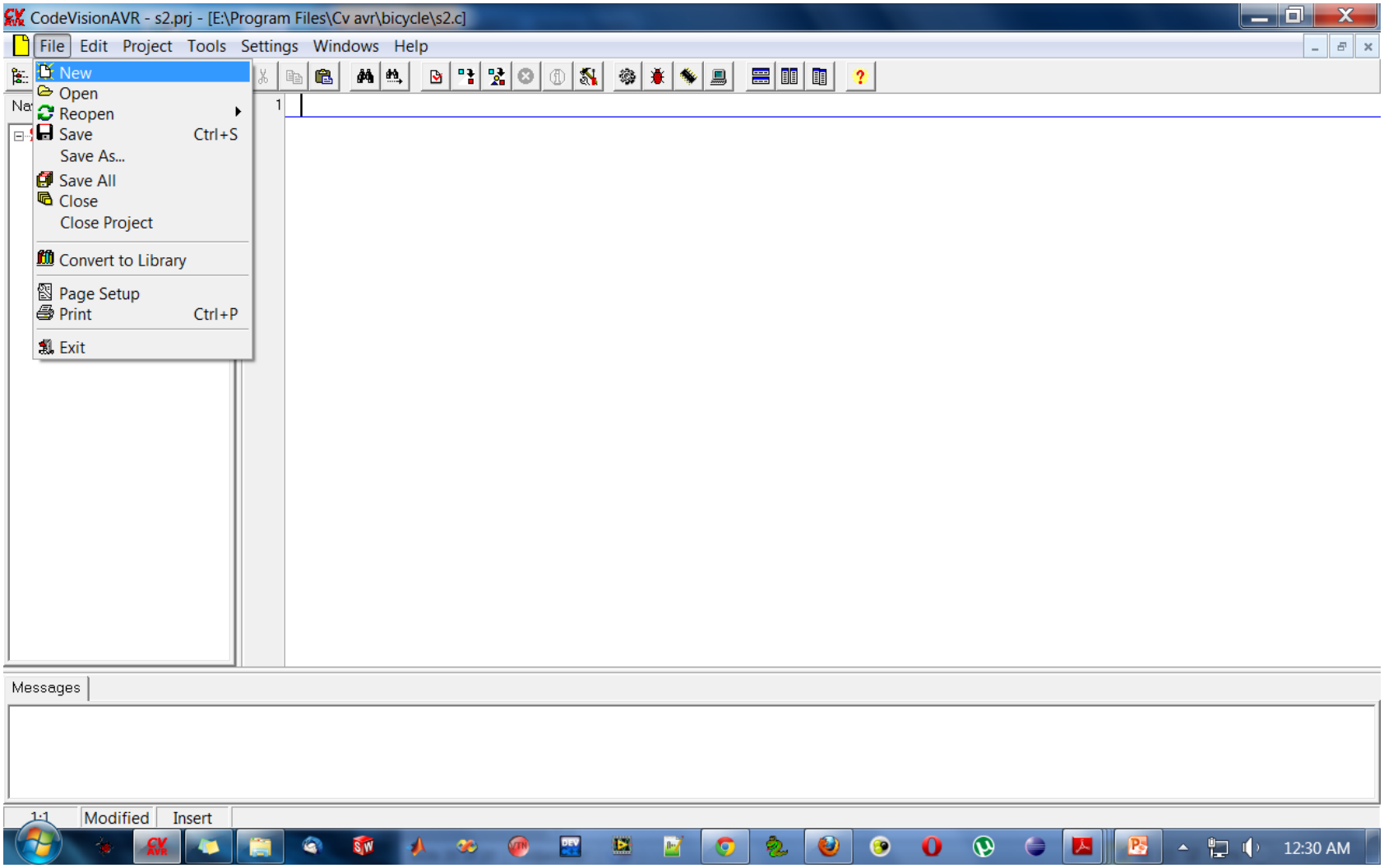
ELECTRONICS CLUB IITKANPUR

Go to File



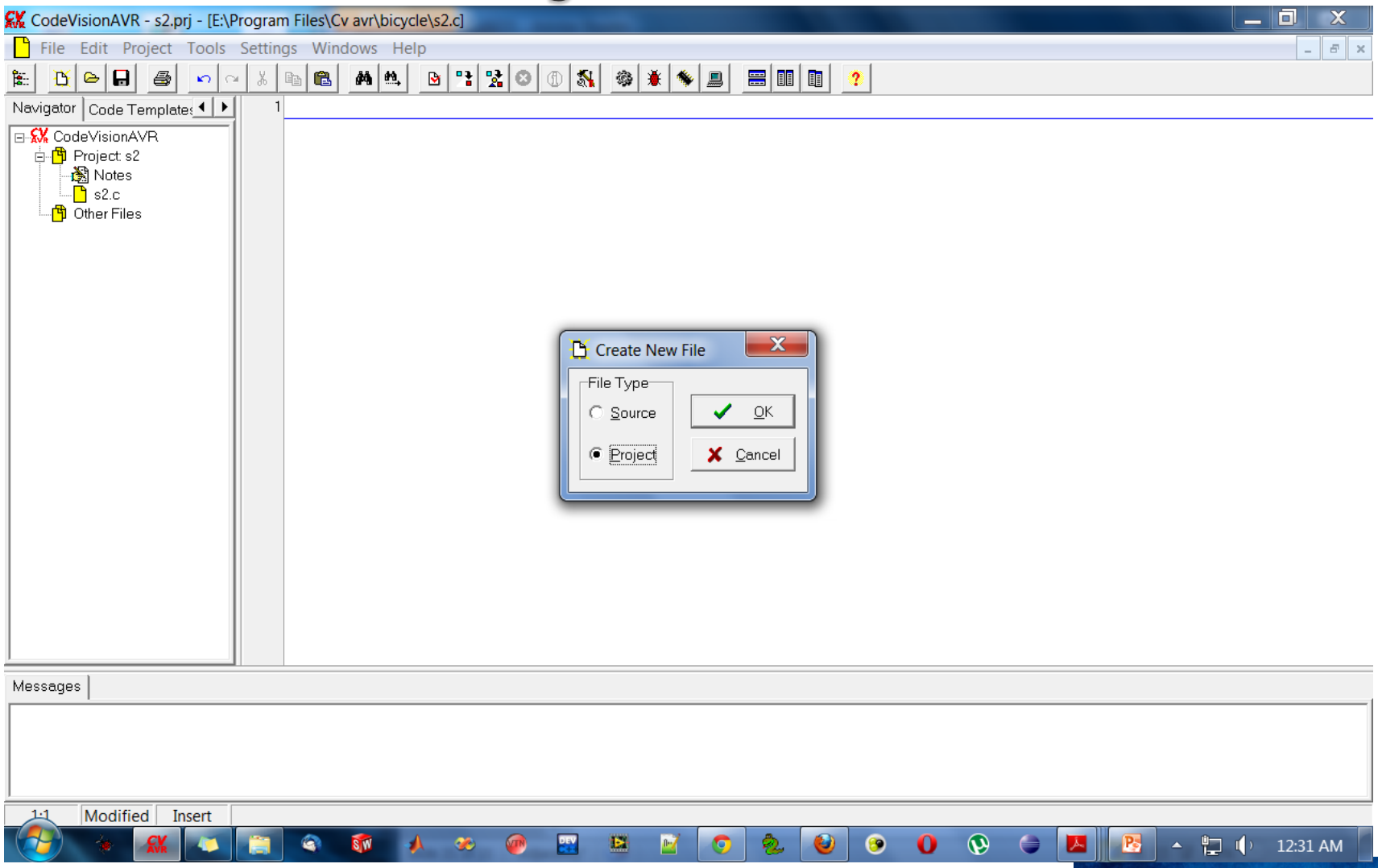
ELECTRONICS CLUB IITKANPUR

Click on New



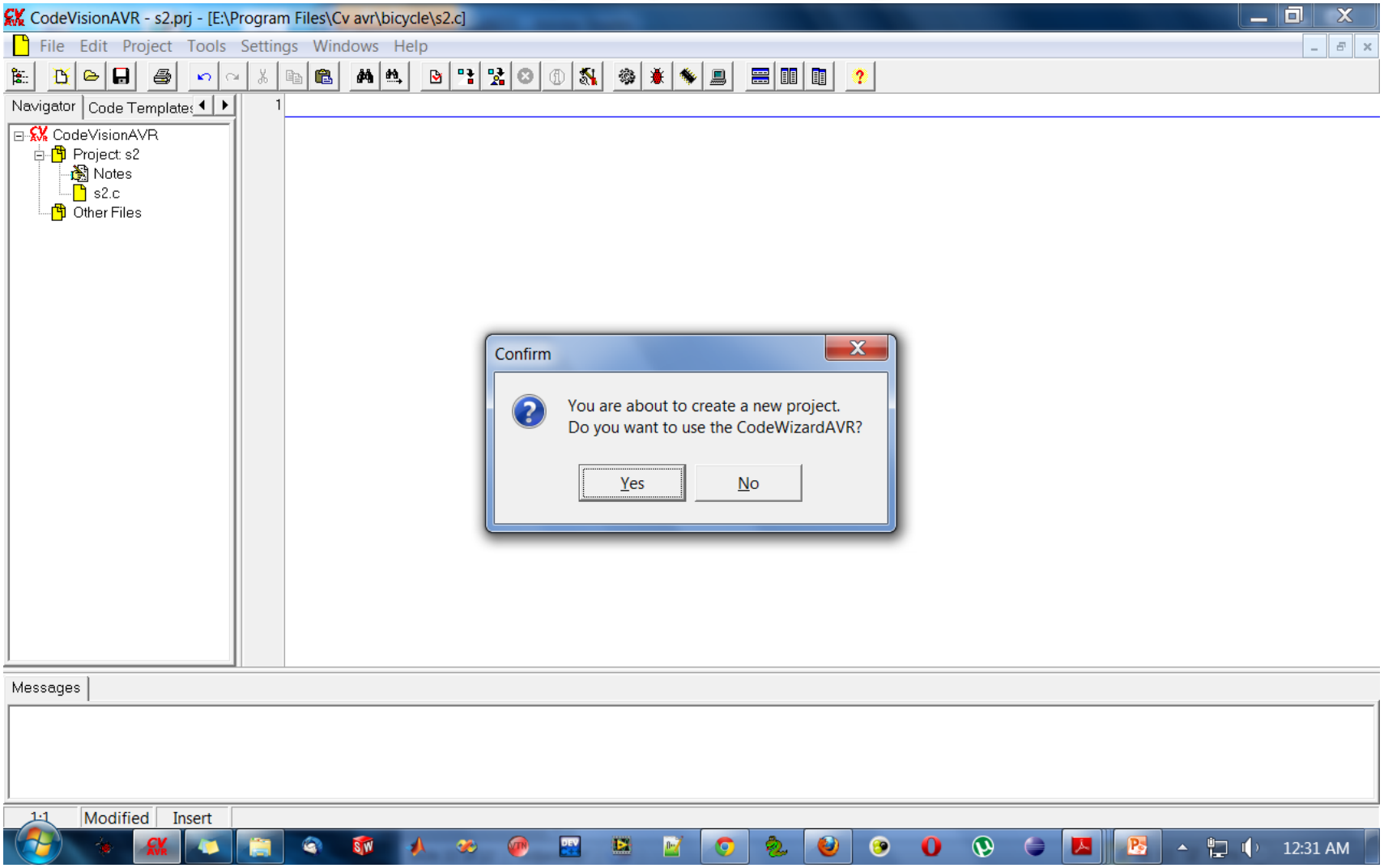
ELECTRONICS CLUB IITKANPUR

Select Project- > Click OK



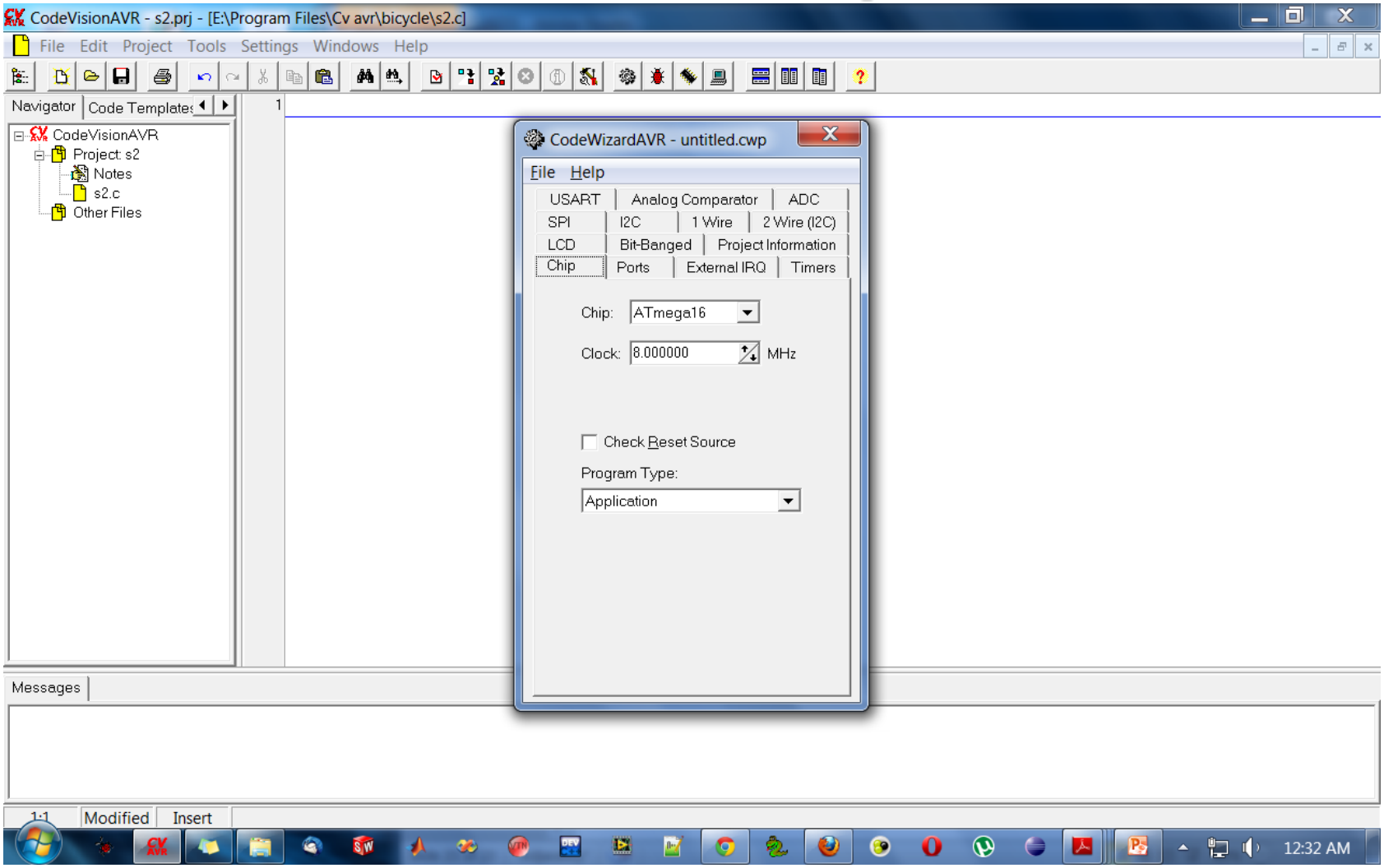
ELECTRONICS CLUB IITKANPUR

Click YES



ELECTRONICS CLUB IITKANPUR

Select Chip



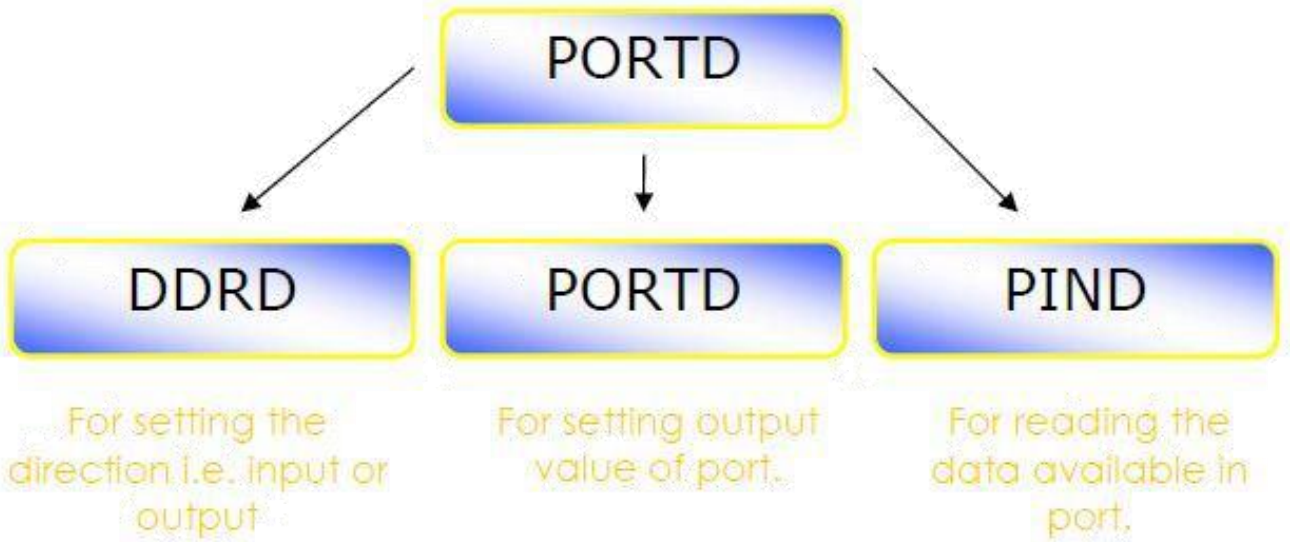
ELECTRONICS CLUB IITKANPUR

Introduction to I/O

- Atmega has total of 40 pins out of which 32 pins can be used as Input or Output
- These 32 pins are divided into 4 groups of 8 pins PORTA, PORTB , PORTC , PORTD

Accessing digital IO in C

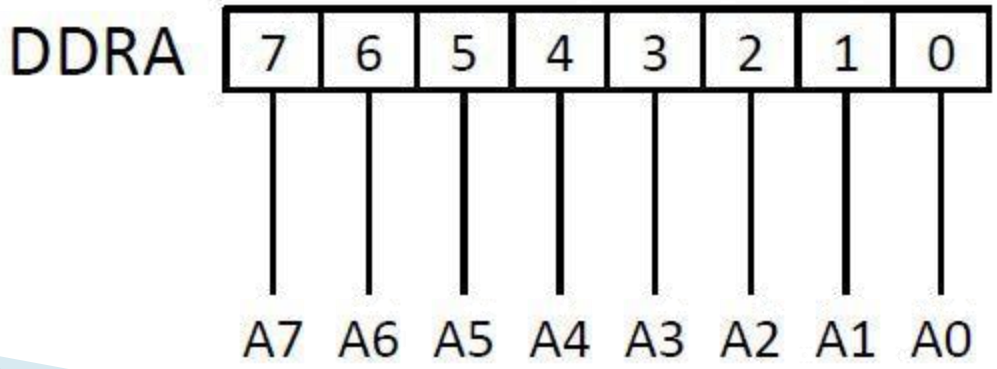
Each PORT in AVR has three related Registers.



ELECTRONICS CLUB IITKANPUR

Data Direction register (DDR)

- This sets direction for all pins (32)
- Direction for these pins can be Input or Output
- To blink an LED we need to set pin as “OUTPUT” but “HOW“ ?
- DDRA = 0b00000001 ;
- DDRA = 0x01 ;
- 1 Stands for Output & 0 stands for Input



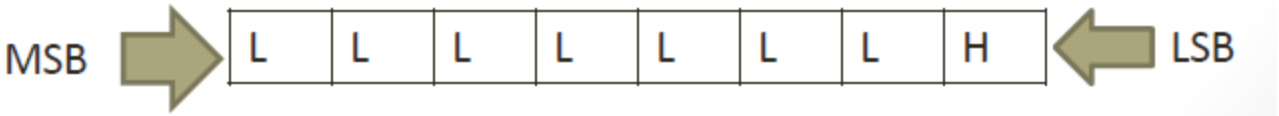
ELECTRONICS CLUB IITKANPUR

What Next ?

- We have set the Pin as Output
- What else do we need to light the LED ??
- Supply of 5 Volts !!! This is given by PORT Register

PORT Register

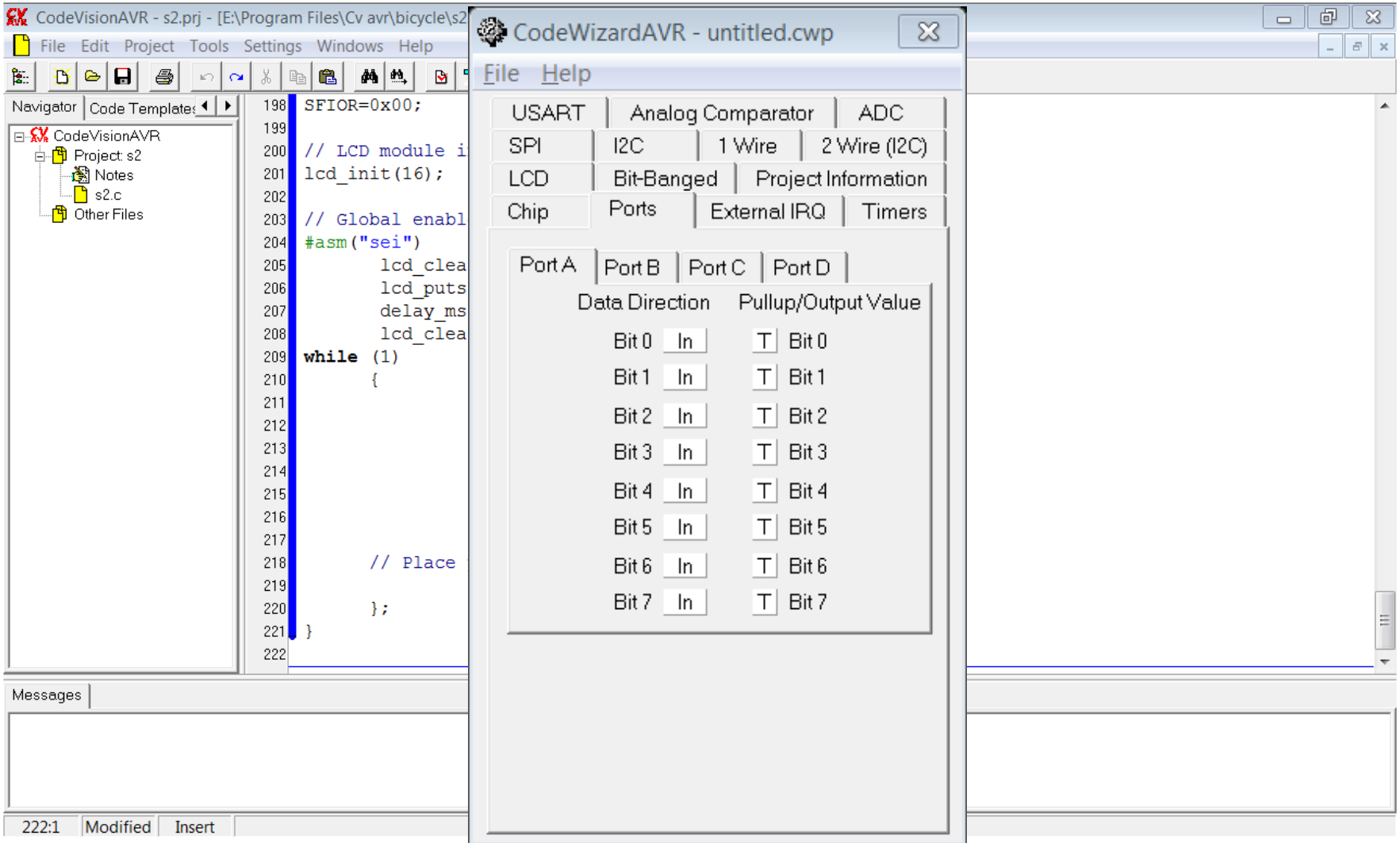
- Only after you have set the Pin to Output you can control them through this Register
- It is a 8 bit register . It corresponds to the pin in same manner as that of DDR Register
- Used to set output value (0 or 1) only if the corresponding Pin has been set as output by DDR Register
- PORTA= 0b 00000001; or
- PORTA= 0x01 ;
- 1 stands for 5V
- 0 stands for 0V



ELECTRONICS CLUB IITKANPUR

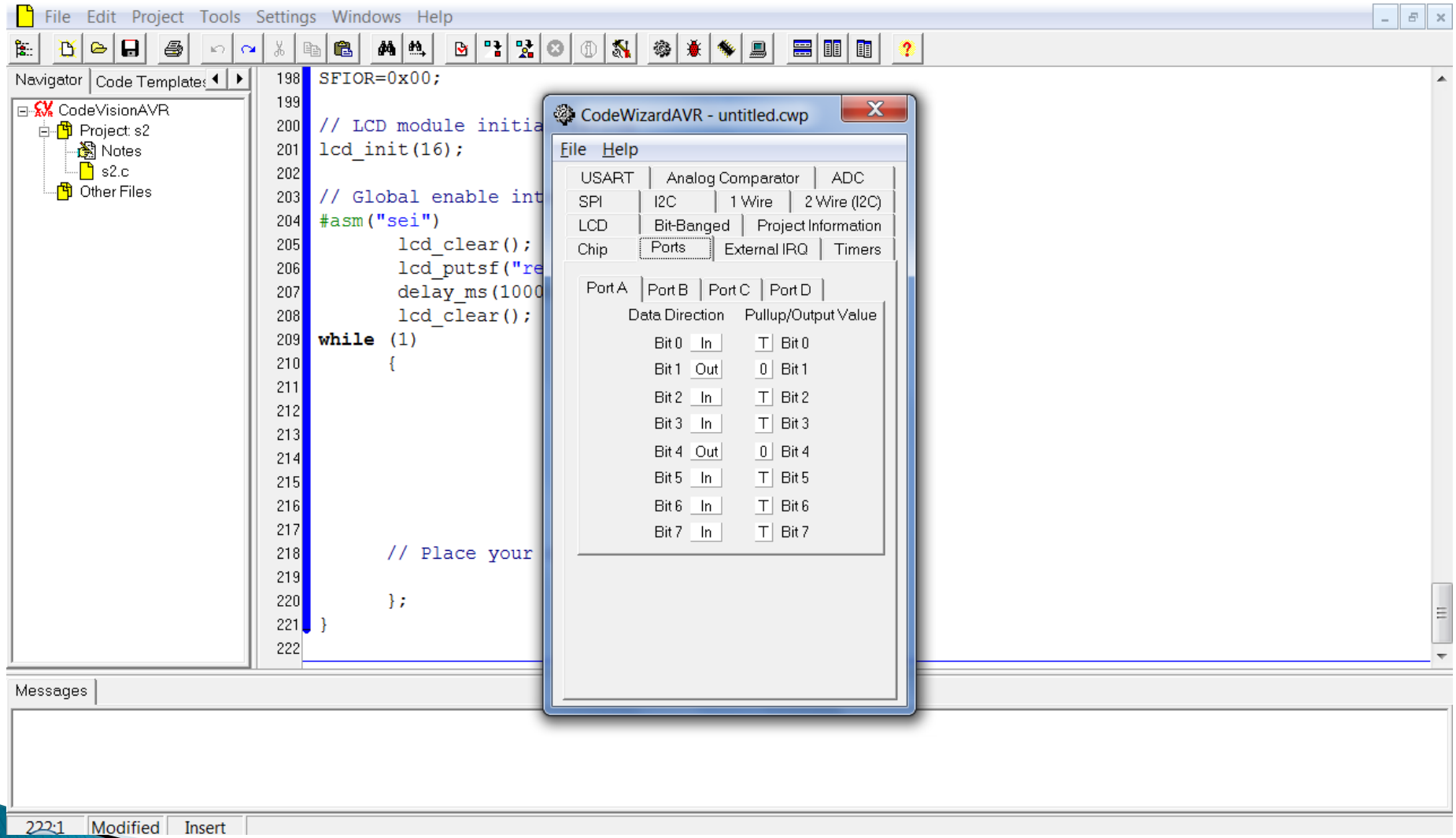
Setting I/O

Go to Ports

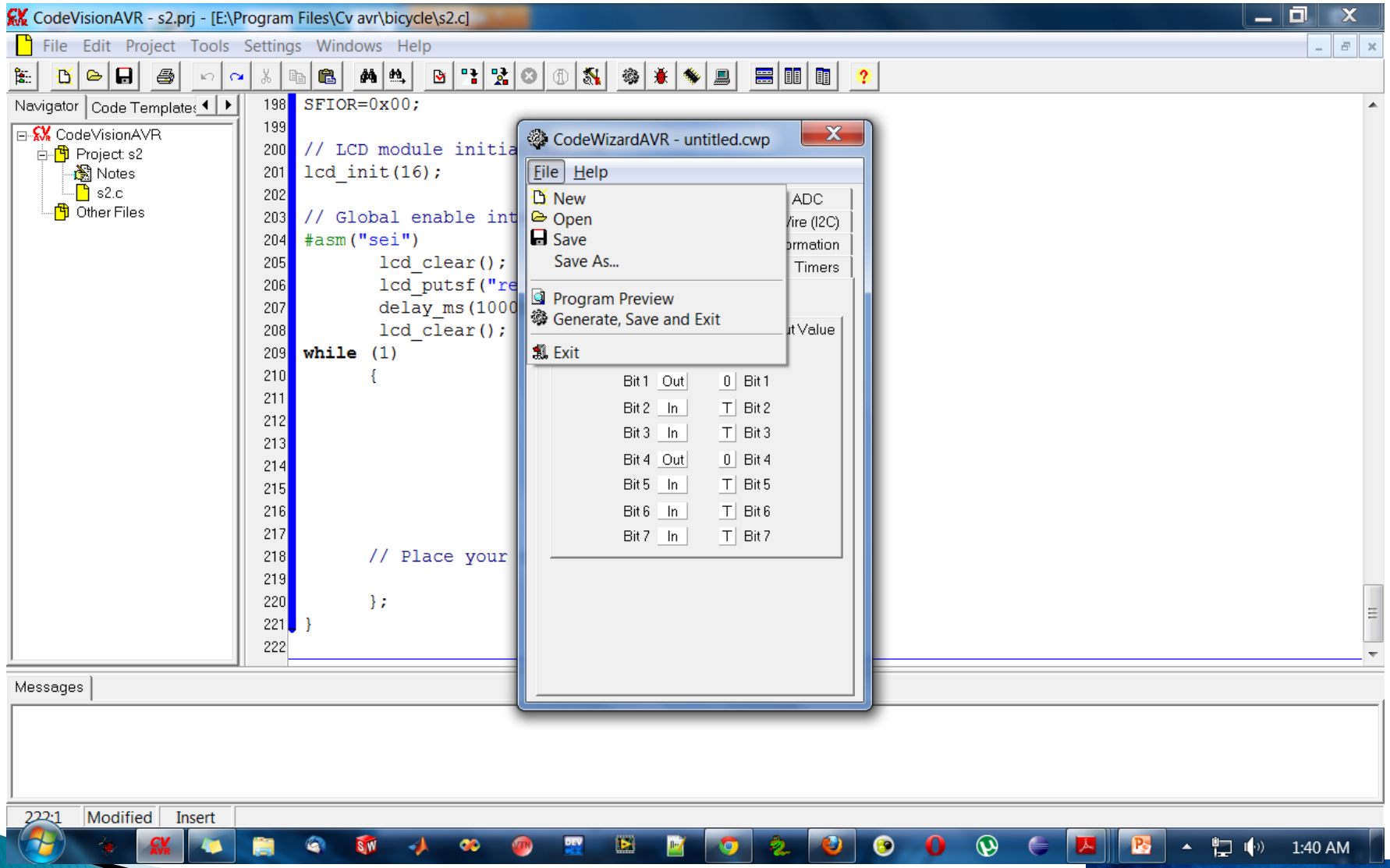


ELECTRONICS CLUB IITKANPUR

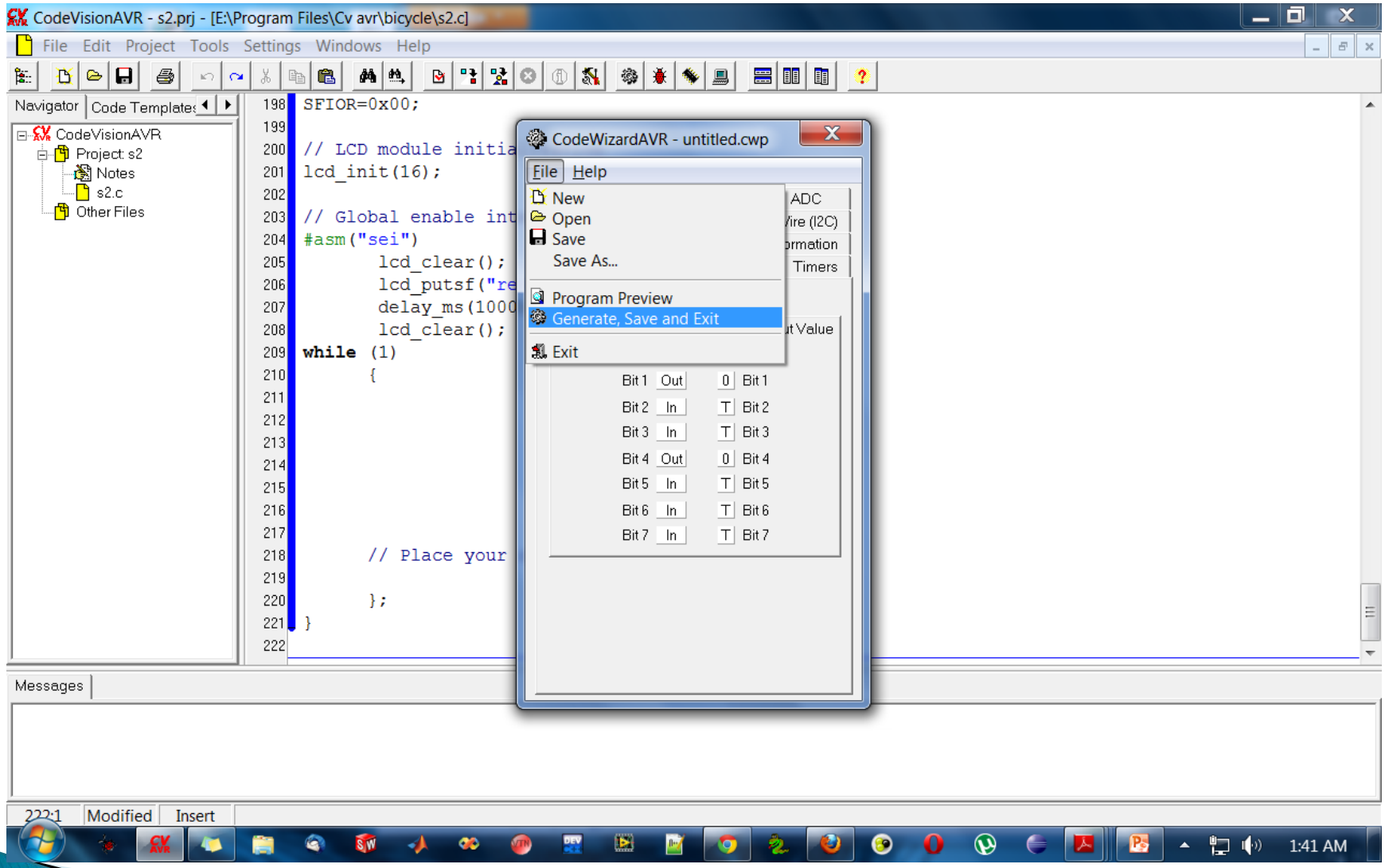
- Click on In to make that pin Output
- Can do so for all four ports



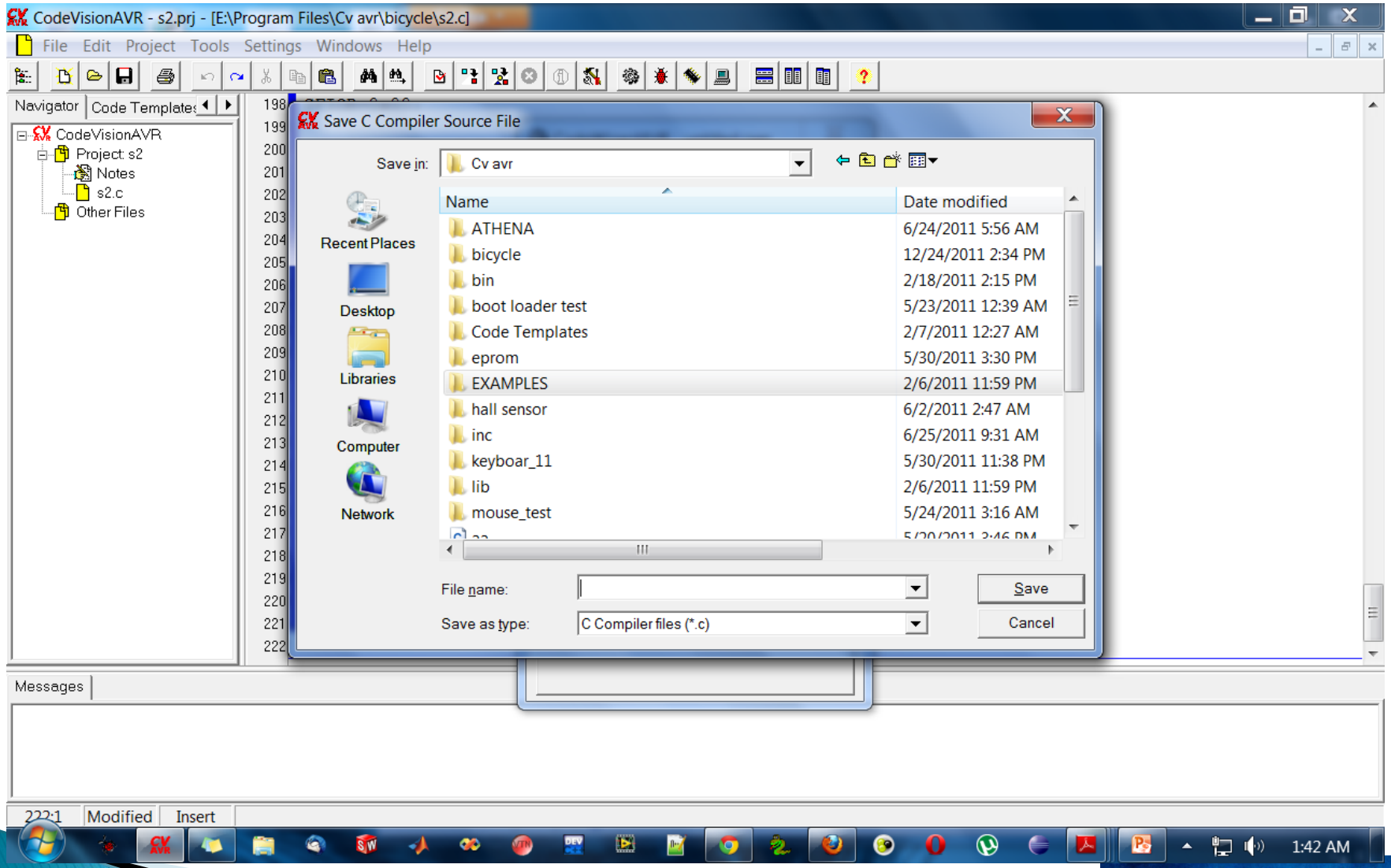
Click on File



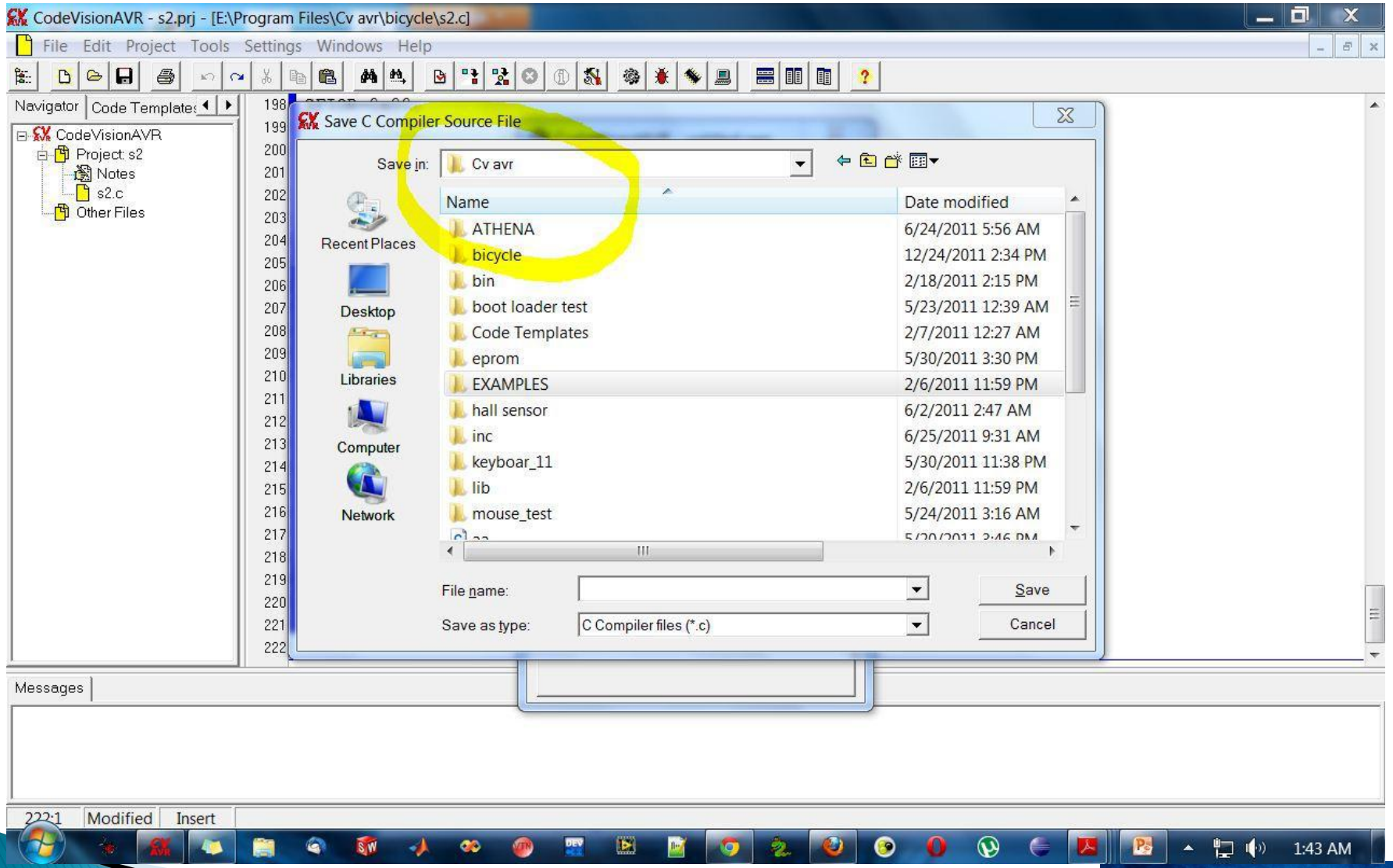
Generate Save and Exit



Enter name (3 times)

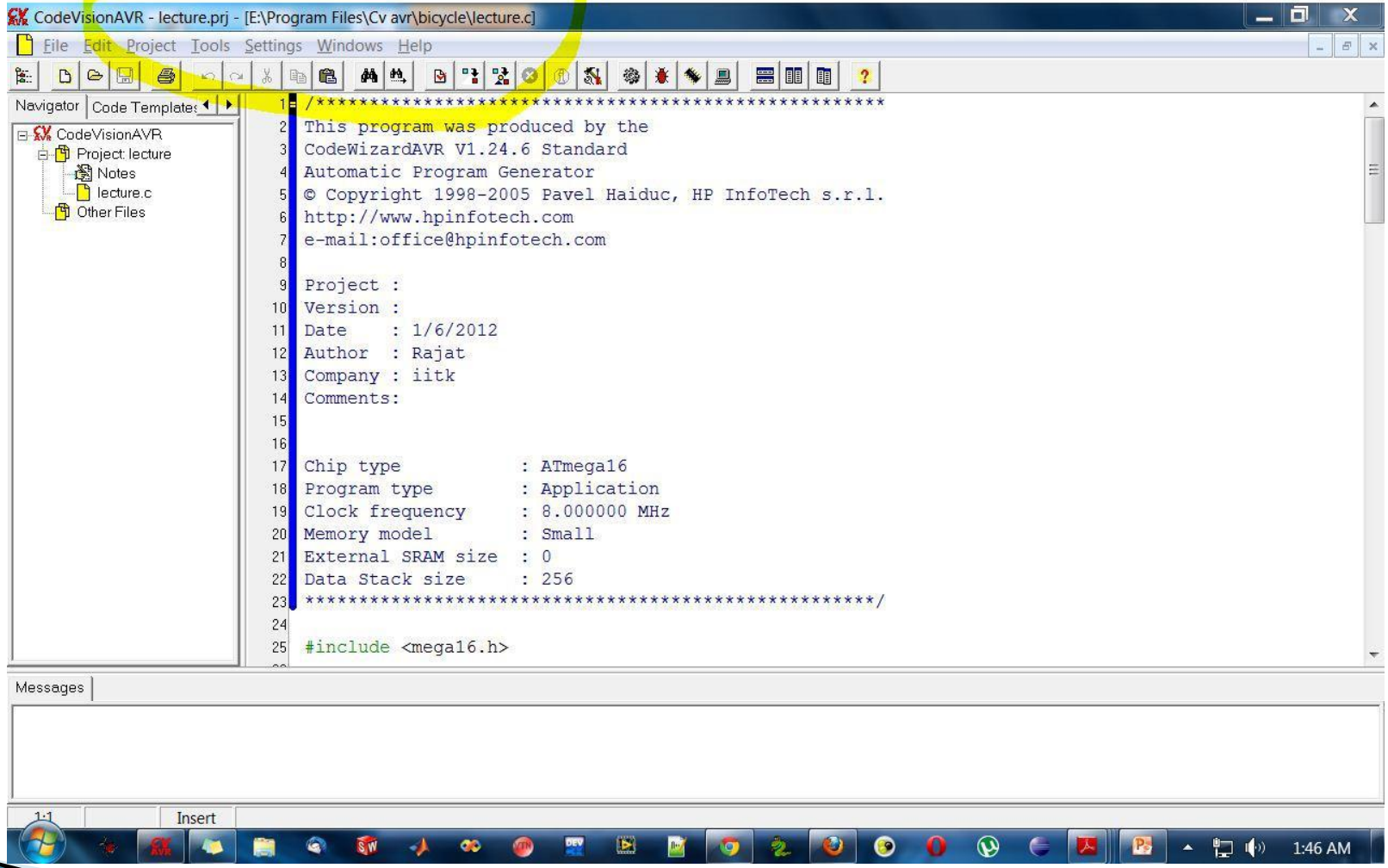


Where is the code stored ?



Then Click Save

Name of Project & Location



Writing the Code

- NOTE : We write our code in While block
- While (1)

```
{
 PORTA.1=1; // sets the Pin to 5 volts
 PORTA.1=0; // sets the Pin to 0 volts
 }
```

- This makes the LED to blink but we cannot see blinking !!!

- This is because Atmega runs at a frequency of 8000000 Hz
- We need to introduce delay so as to see blinking
- Use header file delay.h
- Function to be used · delay_ms(time in millis);
- While (1) {

```
delay_ms(1000);
PORTA.1=1;
delay_ms(1000);
PORTA.1=0;
}
```

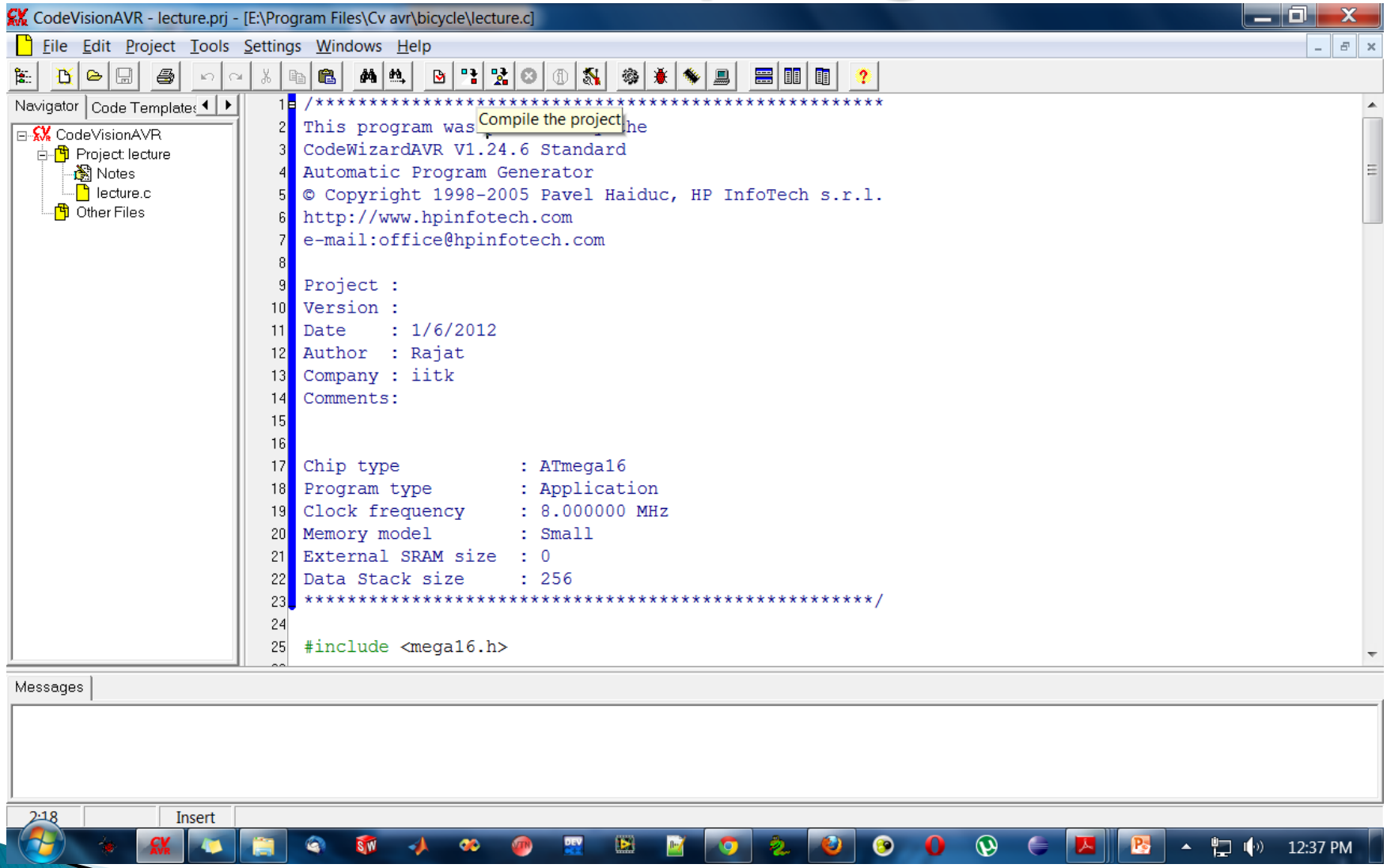
- This is because Atmega runs at a frequency of 8000000 Hz
- We need to introduce delay so as to see blinking
- Use header file delay.h
- Function to be used · delay_ms(time in millis);
- While (1) {

```
delay_ms(1000);
PORTA.1=1;
delay_ms(1000);
PORTA.1=0;
}
```

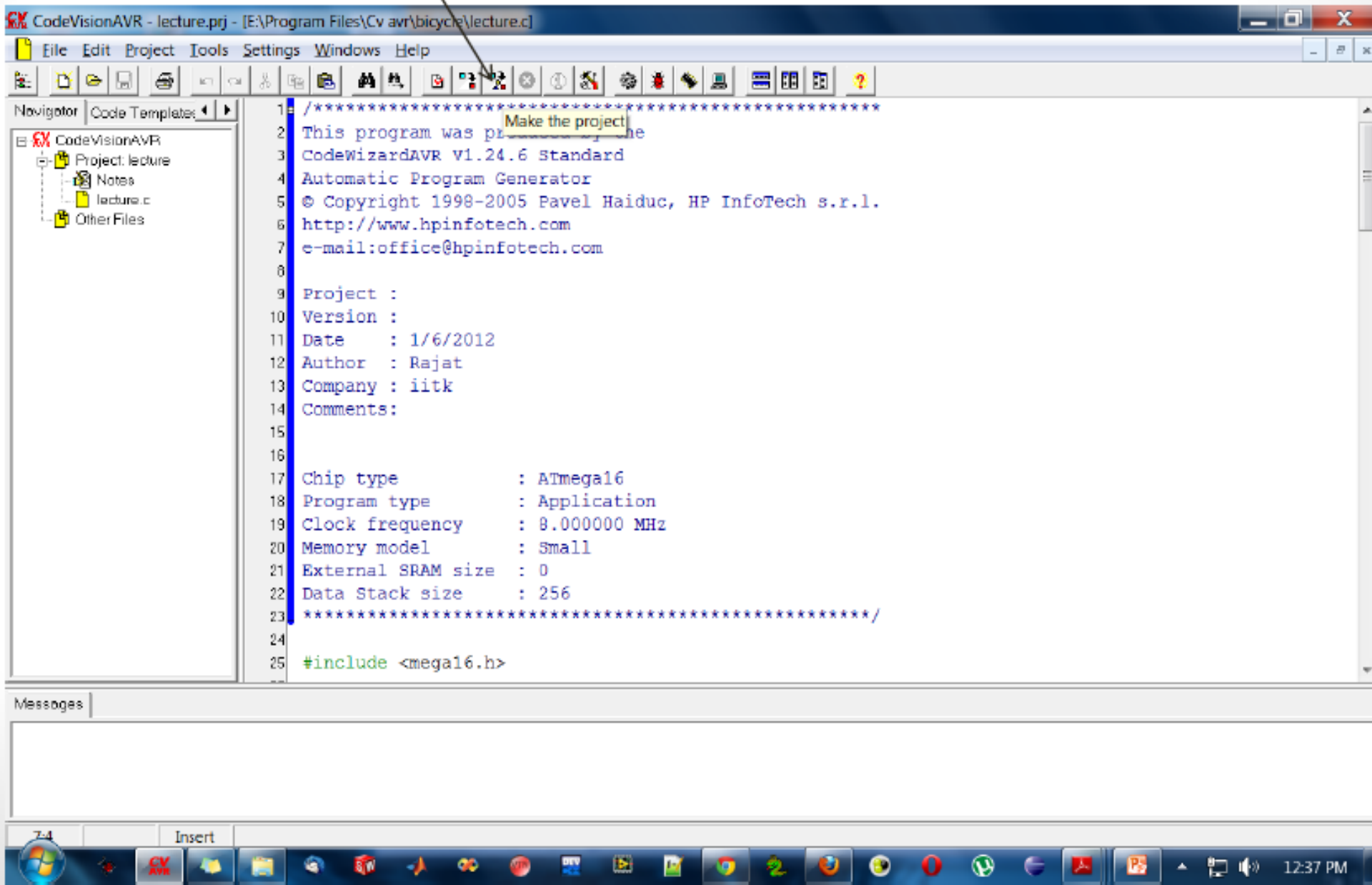
How to compile

- Code is written in C language but Atmega understands Hex file
- so we need to convert the C file to Hex file

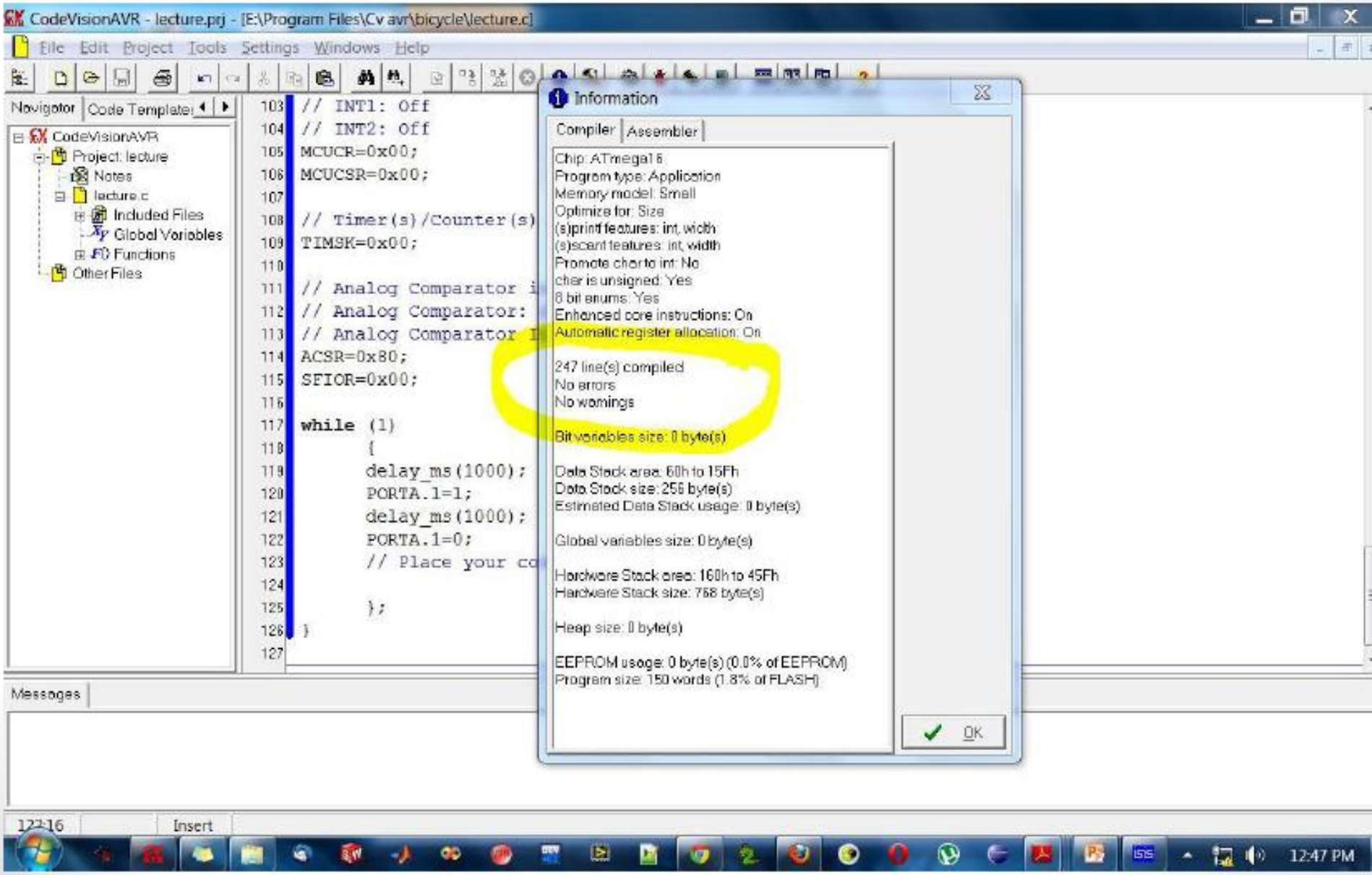
Compiling



Make the Project



Check for errors



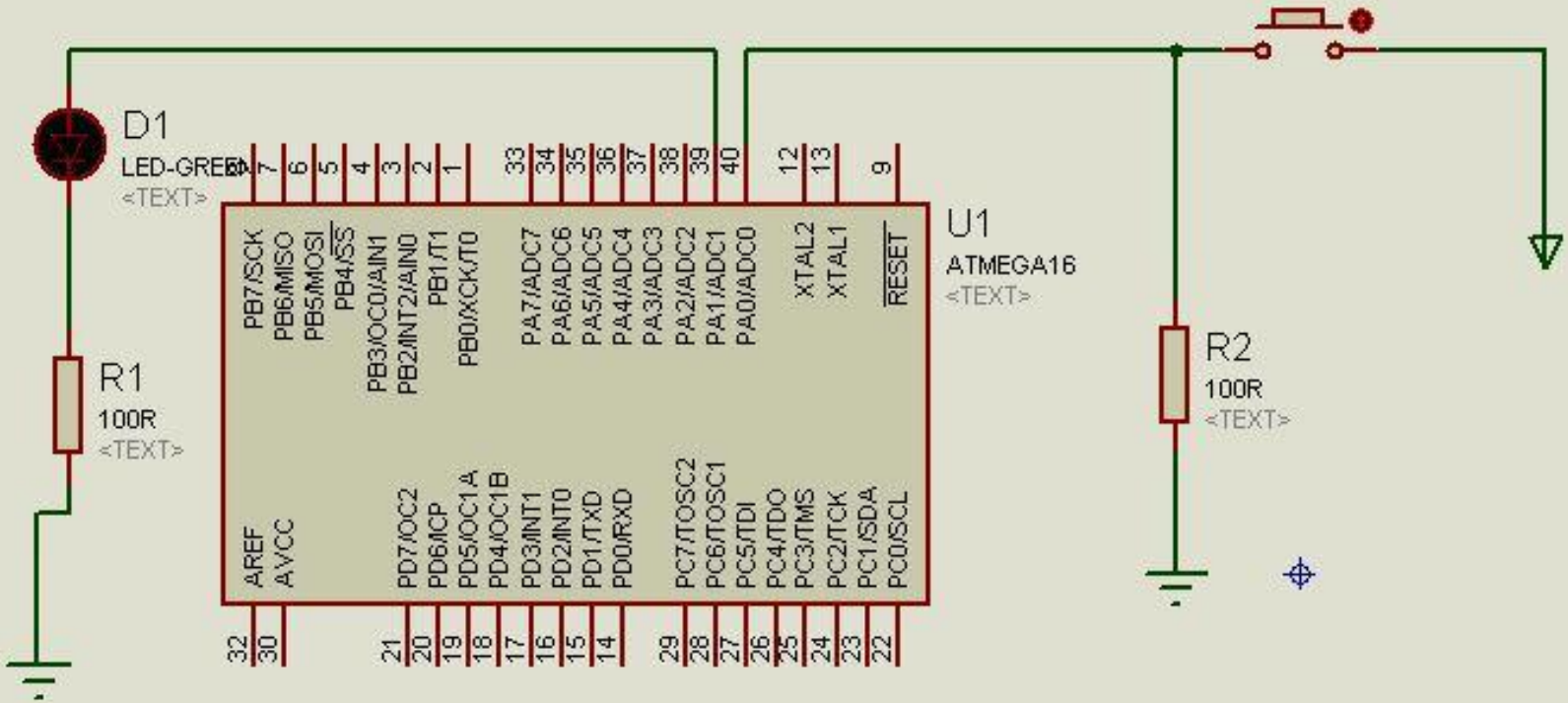
Hex File

- You can find the Hex file in Bin folder or the EXE folder of the directory where You installed CVAVR
- So we Have our Code ready
- Feed this code to Atmega using Programmer
- Lets see the code in action

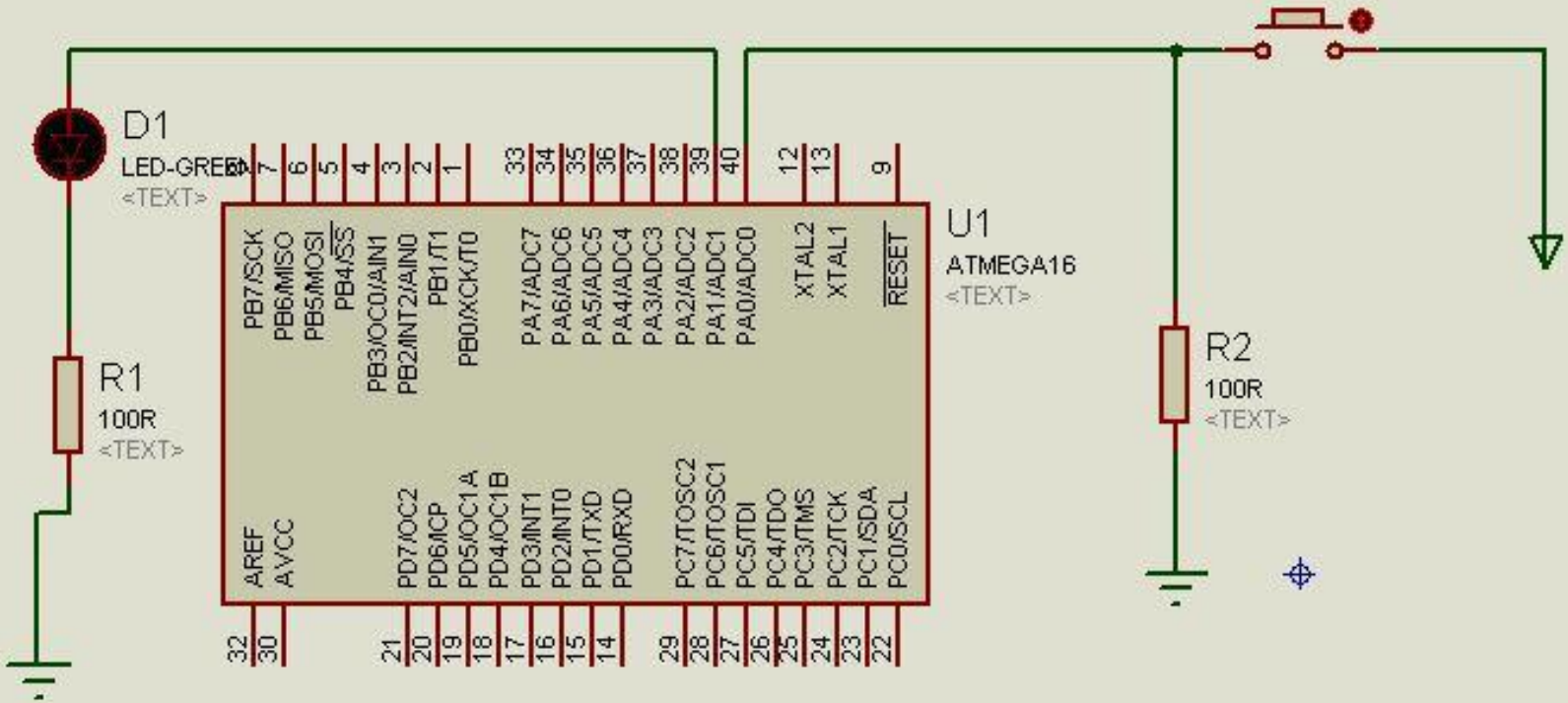
Lets add an Input

- Most Common Input - Button
- Since we have already made A0 as Input we connect a button to that pin
- If button is pressed light the LED else turn it off
- First draw the Circuit Diagram

Circuit Diagram



Circuit Diagram



PIN Register

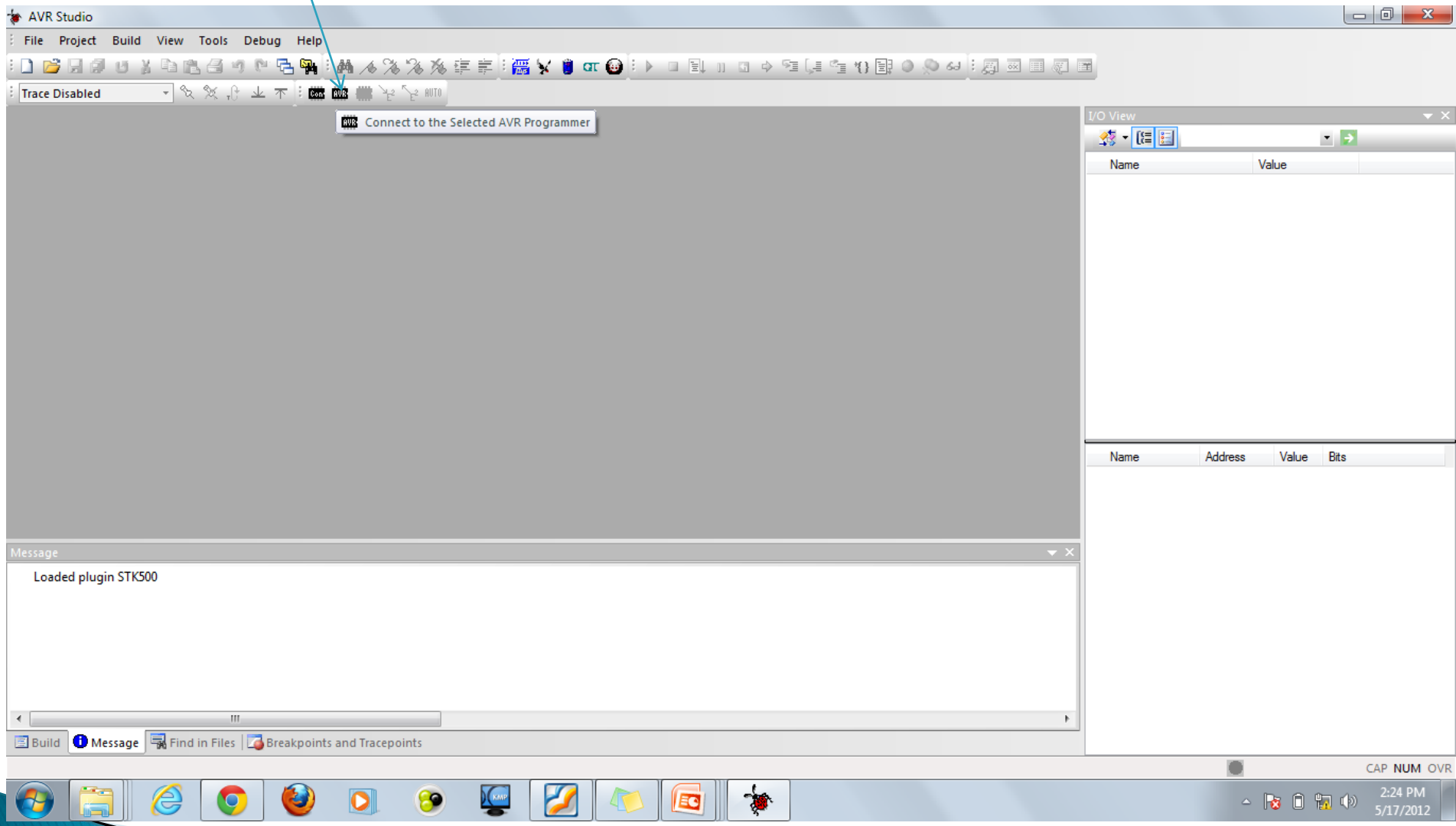
- It is a 8 bit register . It corresponds to the pin in same manner as that of DDR Register
- It is used to read voltage at a pin
- To be used only after the pin has been set as input by DDR register

Using Pin Register

- int a; // Define the variable a to store the value of voltage
- a=PINA.0; // read value at pin A.0 (make sure it is input)
- If (a==1) // if voltage is 5V {
- PORTA.1=1; // Light the LED
- }
- else
- {
- PORTA.1=0; // Turn off the LED
- }

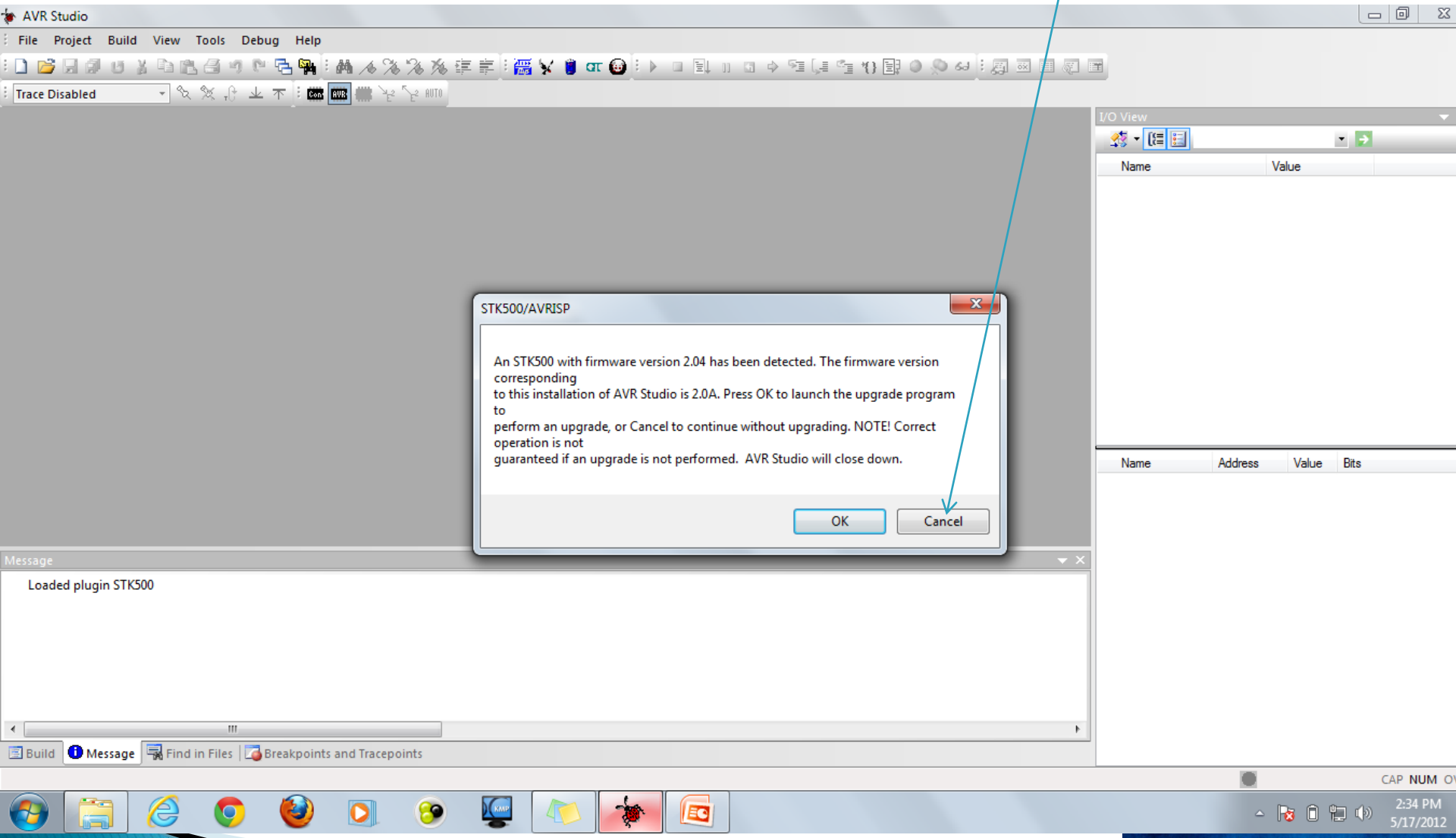
AVR Studio4

Click on this icon

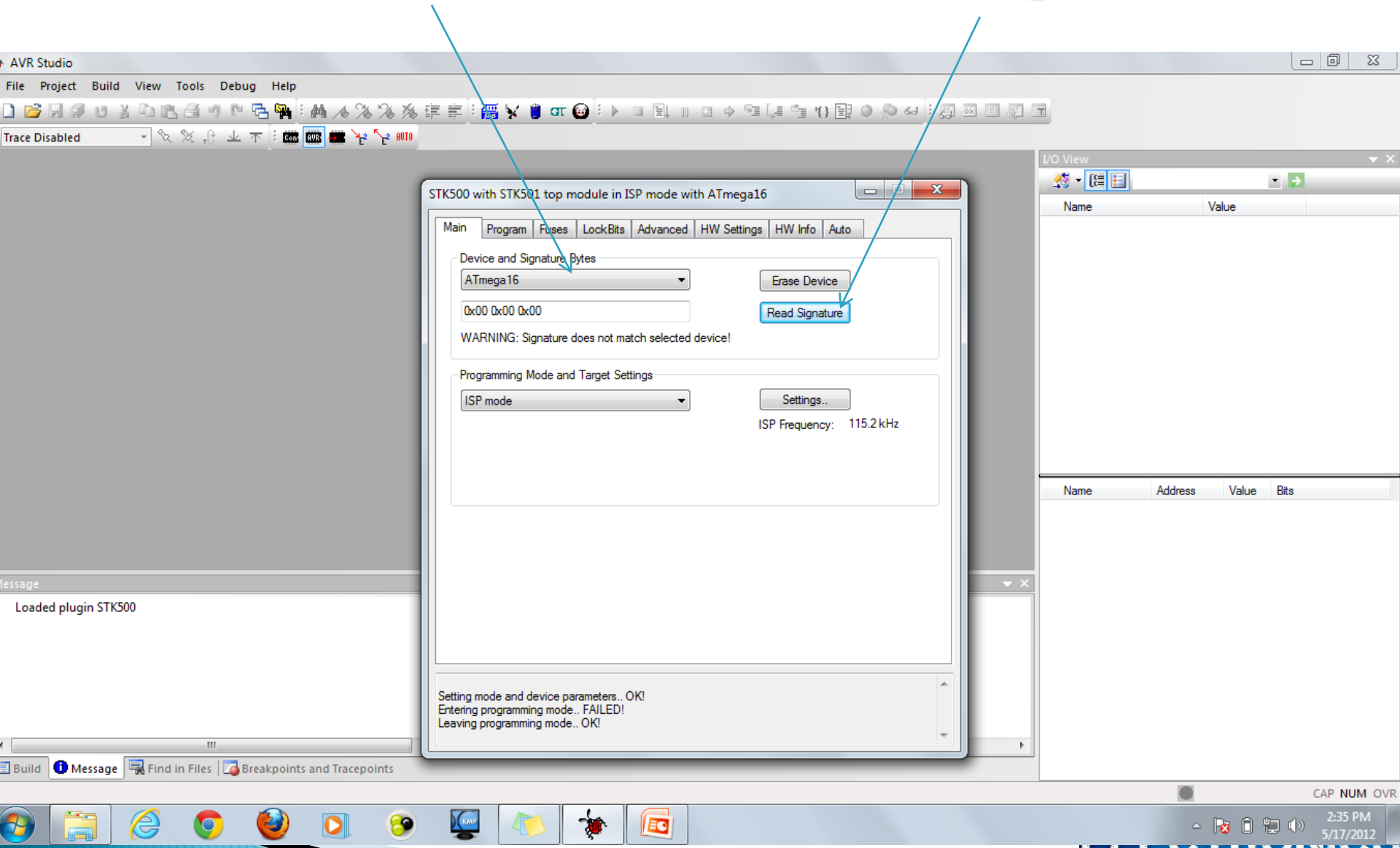


ELECTRONICS CLUB IITKANPUR

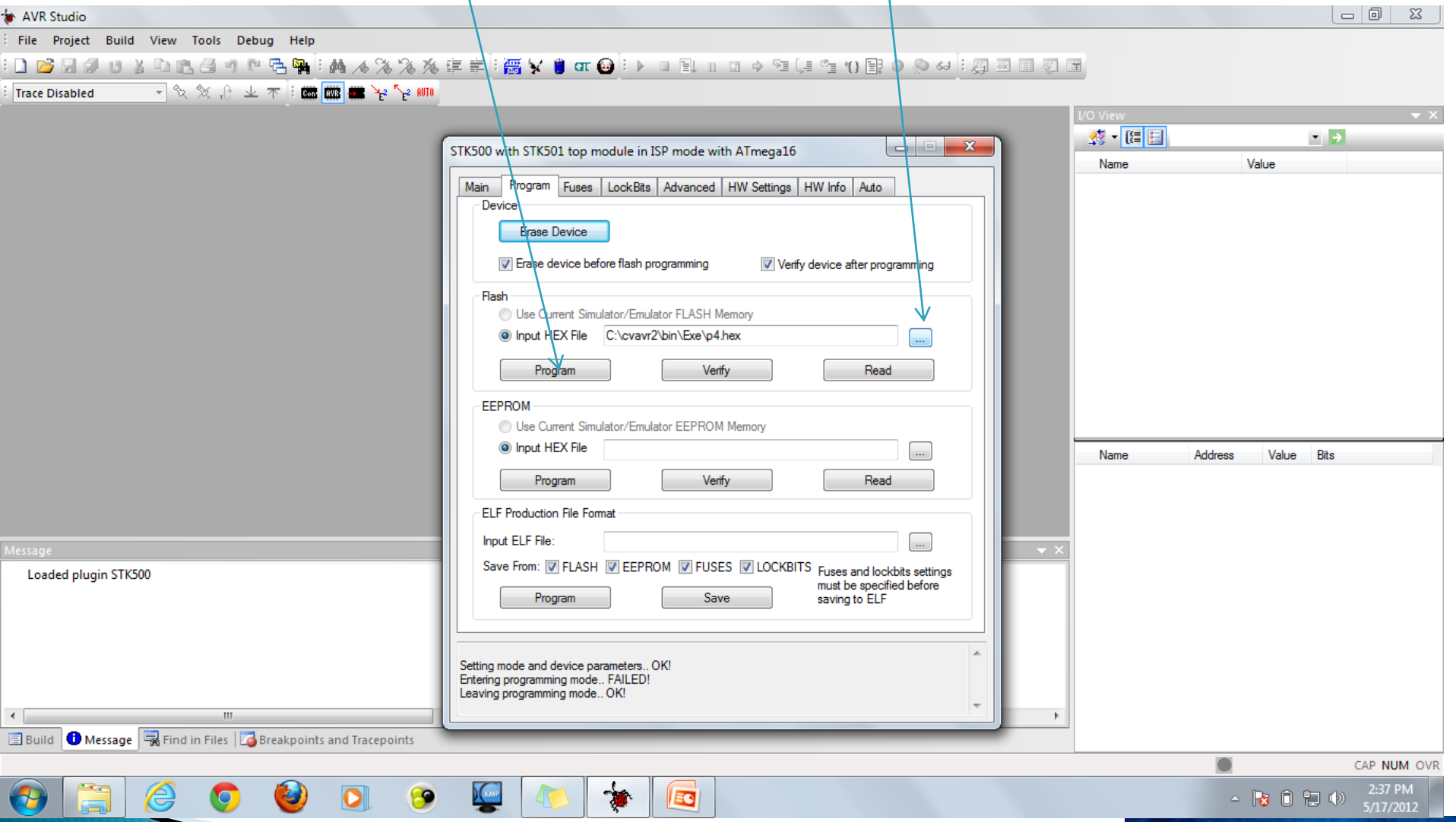
Click Cancel



Select Device and Read signature



Select Hex file and then click program



ELECTRONICS CLUB IITKANPUR

Difference from Other

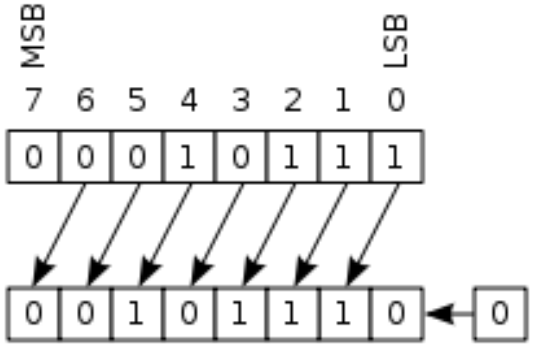
- You can’t write PORTA.1 in this
- User friendly
- Easy interface
- Can add external library

How to Access PortA.1

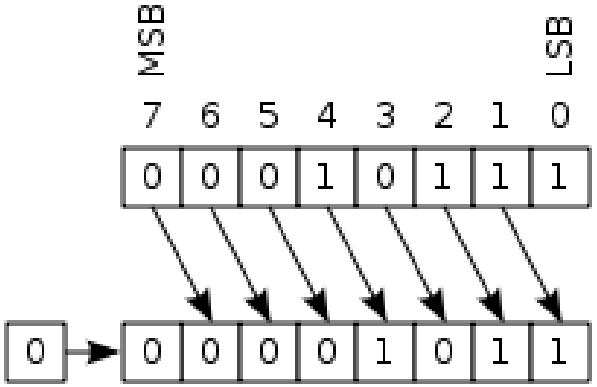
- Using simple bit wise C operation
- For PINA.1
- Value_PINA1=(PINA & 0b00000010)>>1
- For PORTA.1
- PORTA=PORTA | 0b00000010 // PORTA.1=1
- PORTA=PORTA & 0b11111101 // PORTA.1=0

How to Access PortA.x

- Left shit operator
- <<



- Right shift operator
- >>



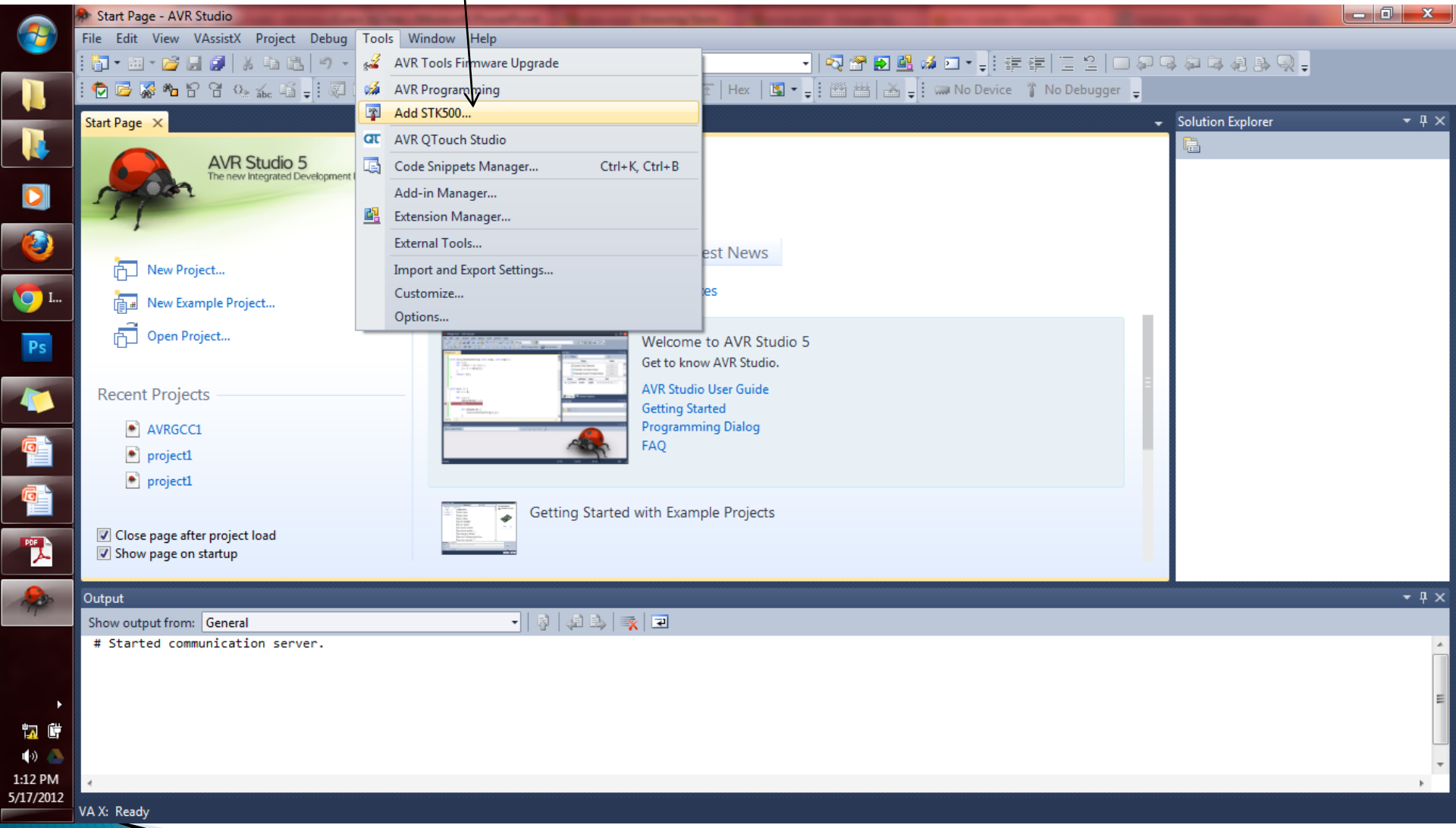
ELECTRONICS CLUB IITKANPUR

How to Access PortA.x

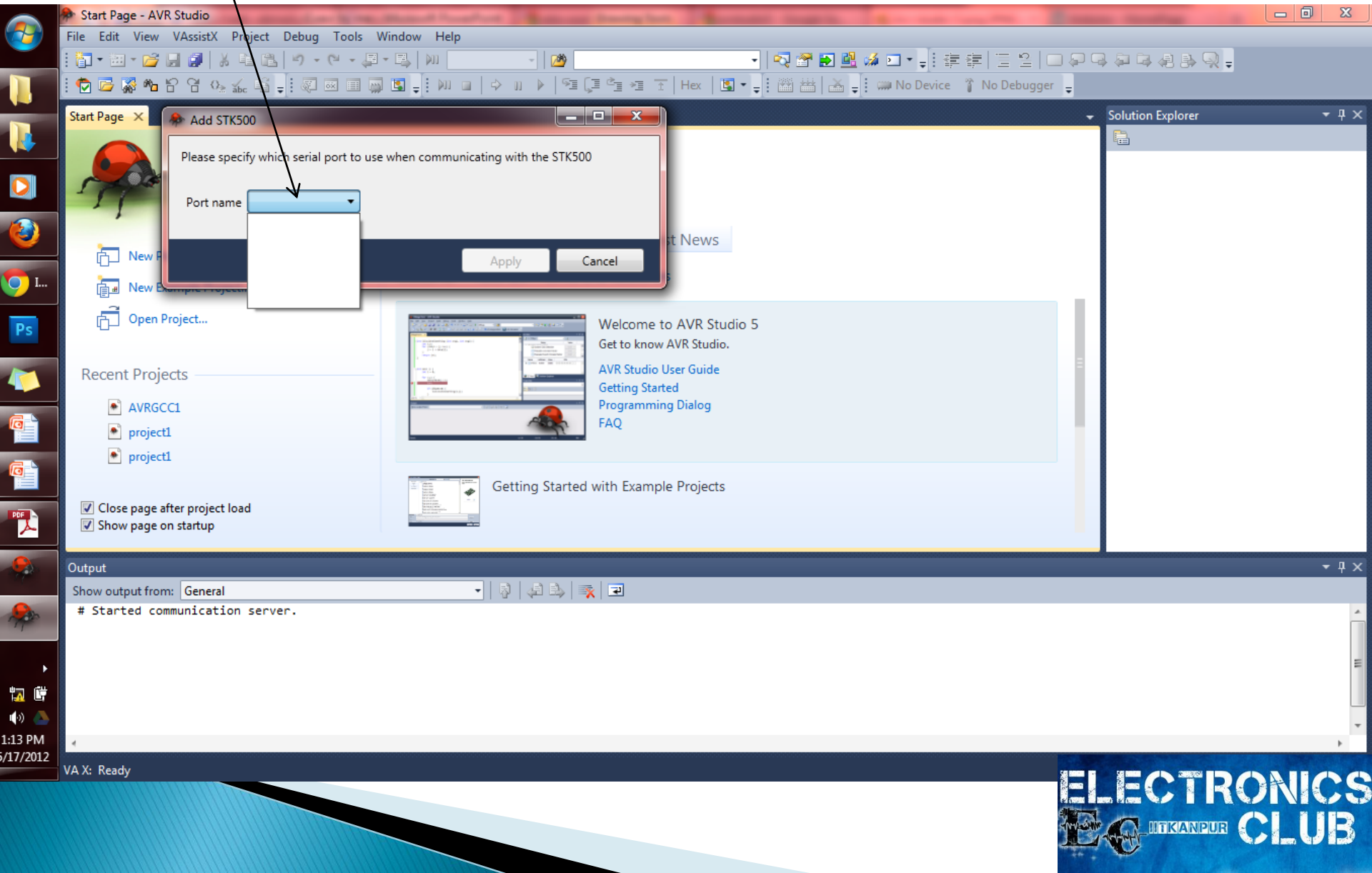
- For PINA.x
- Value_PINAx=(PINA & (1<<x))>>x
- For PORTA.1
- PORTA=PORTA | (1<<x) // to set PORTA.1=1
- PORTA=PORTA & (~(1<<x)) // set PORTA.1=0

Programming

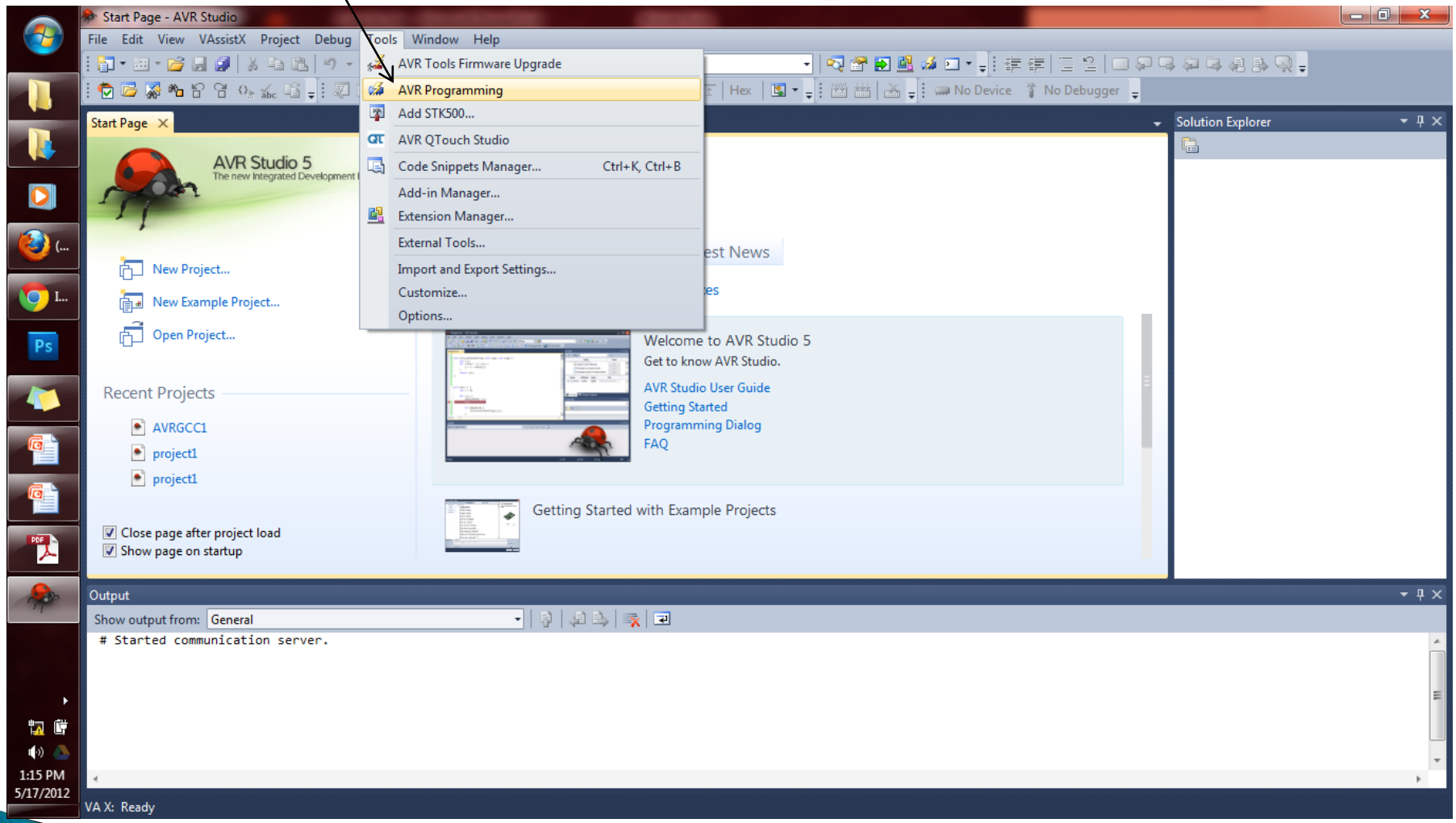
Add Stk500



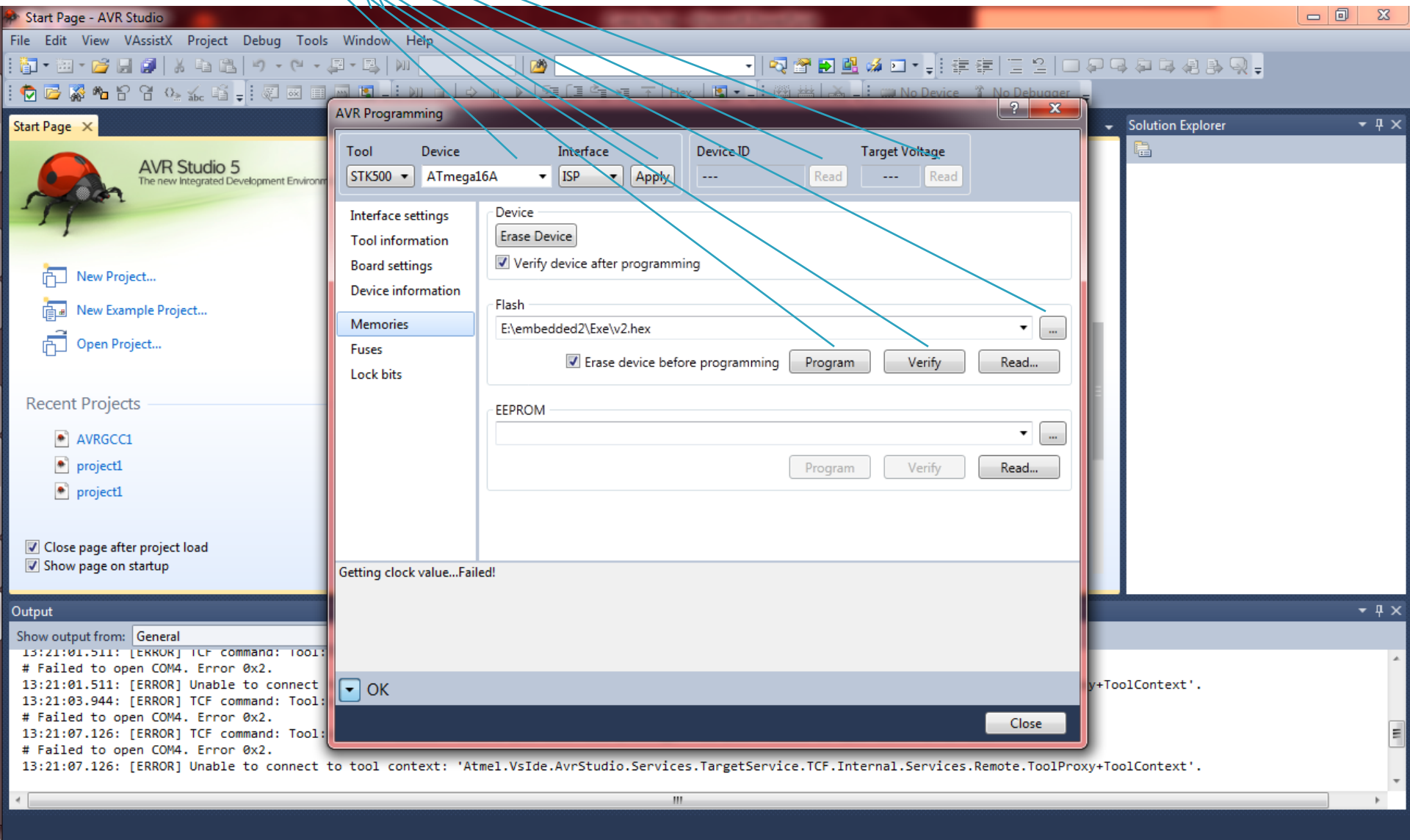
Select Com Port



Select AVR Programming



- Select Device→Click Apply→Read Device ID→Read target Voltage→Choose Hex File→Then Program



Thank you

ELECTRONICS CLUB
IITKANPUR