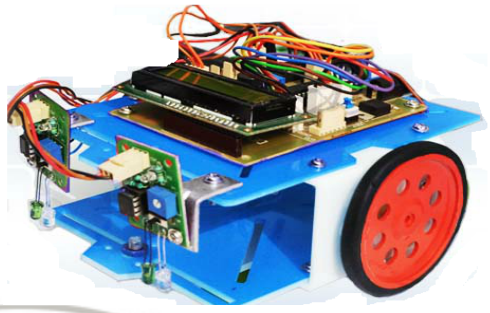
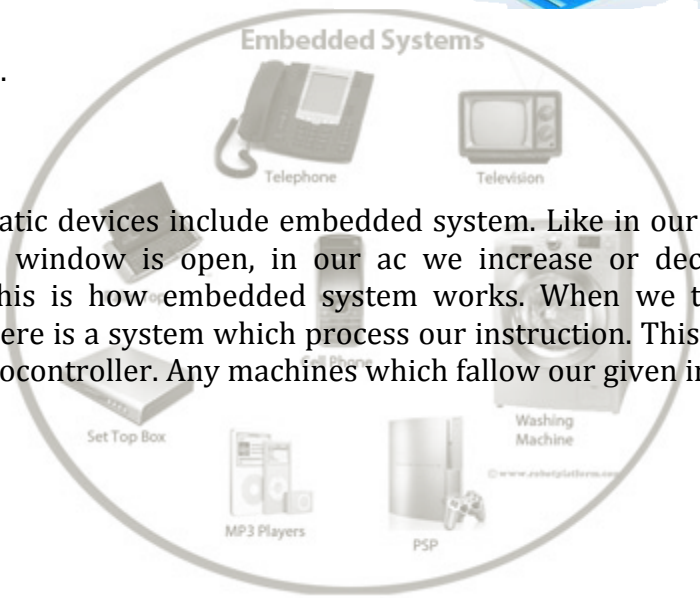
WHAT IS EMBEDDED SYSTEM AND ROBOTICS?

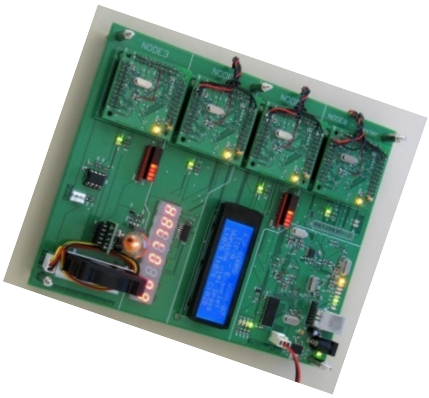


- **We are surrounded by embedded systems**

- Cell phones.
- Washing machines.
- Traffic lights.
- Automatic cars.
- Automobile electronics.



Now a day's almost all automatic devices include embedded system. Like in our car when we press a switch our door window is open, in our ac we increase or decrease the temperature using remote, this is how embedded system works. When we talk about embedded system it means there is a system which process our instruction. This system is called Microprocessor or microcontroller. Any machines which fallow our given instruction is called robot.

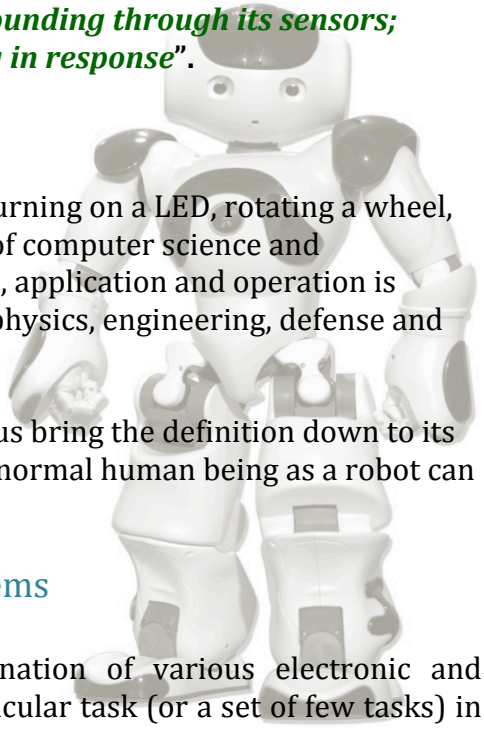


WHAT ARE ROBOTS?

The term robot is derived from Czech word "robota" which means work, or forced work. Nobody has ever given a precise explanation of what a robot is, although each of those definitions more or less means the same.

To make things simpler, we can say that:-

"Robot is a combination of electronics, mechanics and programming (non-programmable in some cases), which senses it's surrounding through its sensors; processes the sensor information and does something in response".

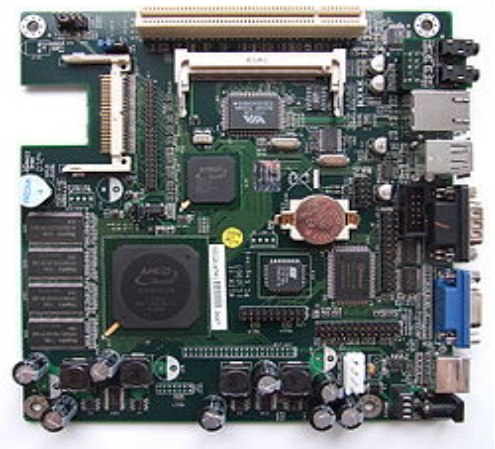


The response can be locomotion or manipulation, like turning on a LED, rotating a wheel, moving an arm, raising an alarm and so on. The branch of computer science and engineering which deals with robot design, construction, application and operation is called **Robotics** with applications in computer science, physics, engineering, defense and even many household devices.

If the above robot definition is too difficult to digest, let us bring the definition down to its very basic level. "Any machine which is recognized by a normal human being as a robot can be considered a **robot**".

Difference between Robots and Embedded systems

Embedded System: Embedded system is a combination of various electronic and mechanical parts which are designed to perform a particular task (or a set of few tasks) in real time with high efficiency and performance. These systems are used in various consumer electronics, medical systems, military applications, etc. Portable music player, cell phones are all examples of embedded systems which have a controller built in to perform specific activities.



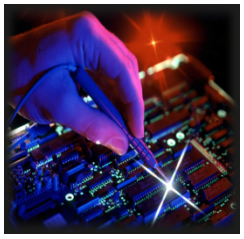
Robots: Robots are theoretically different in that they are equipped with sensors to perceive their environment and actuators to perform particular tasks and can take intelligent decisions.
Although robots and embedded systems seem like two extremes of engineering world, the gap between them is reducing. We already know that washing machines can sense dirt in

cloths and takes intelligent decisions. Air conditioners can sense outside temperature and adjust internal room temperature. These are intelligent embedded systems built inside another bigger system which perceives it environment through its sensors and takes corrective actions, thereby controlling the bigger system.

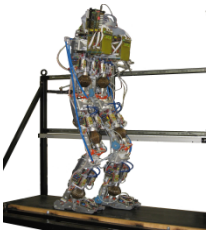
Since most beginners confuse embedded systems with robotics, we will keep embedded systems outside this handout and only use the terms “robots” and “robotics” for our discussions.

ROBOTICS IS INTEGRATION OF

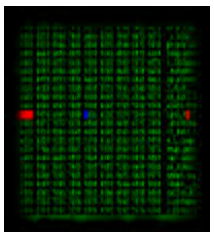
ELECTRONICS+MECHANICS+SOFTWARE=ROBOTICS.



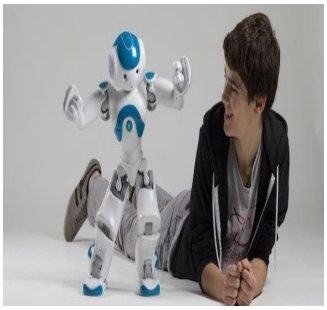
+



+



=



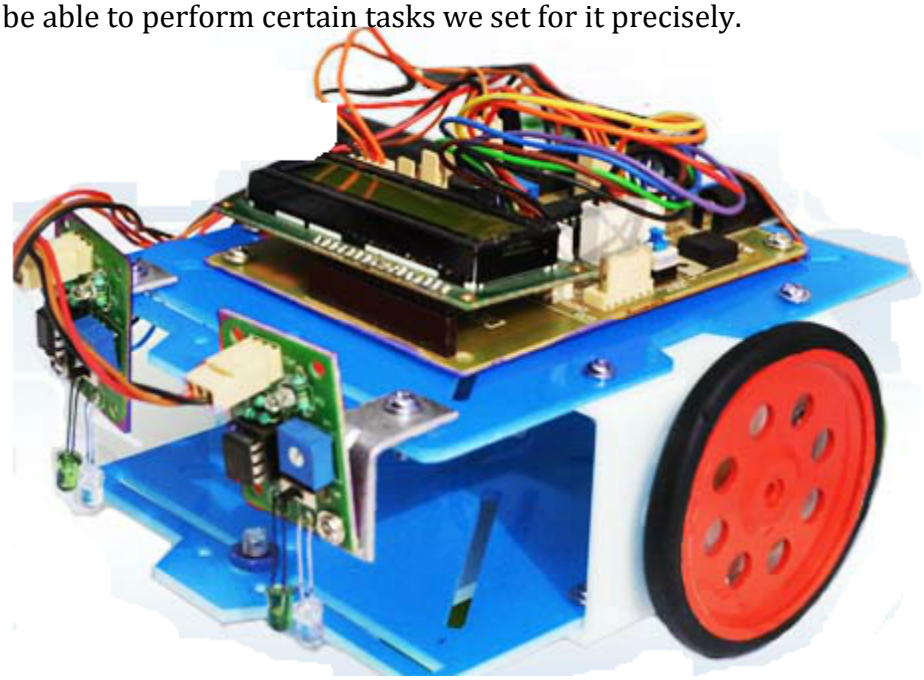
THREE LAWS OF ROBOTICS

Even robots have laws and when you are building a robot, you are *expected* to keep these three laws in mind:

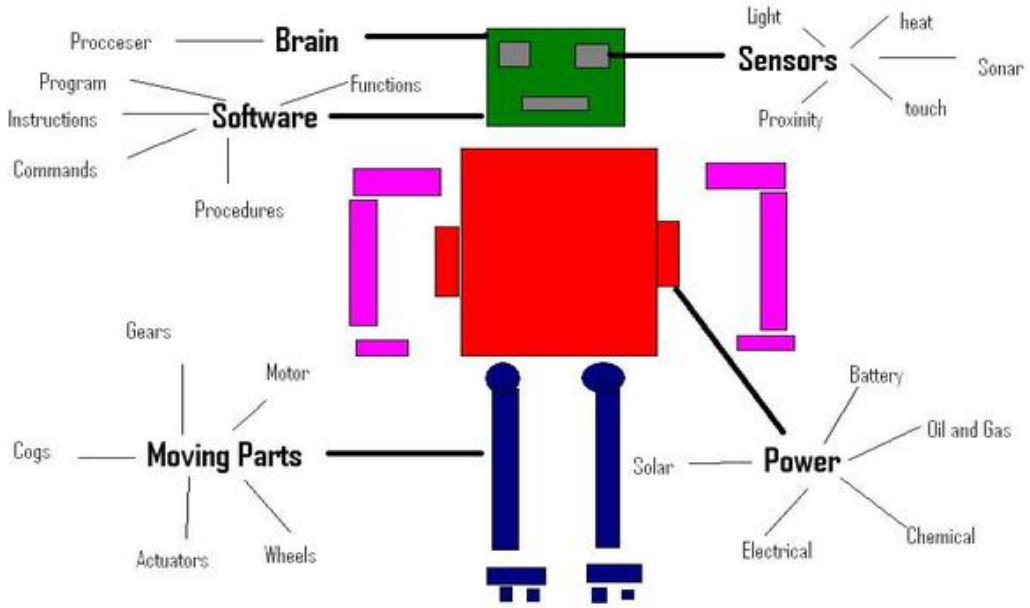
1. A robot may not injure a human being or, through inaction, allow a human being to come to harm
2. A robot must obey any orders given to it by human beings, except where such orders would conflict with the First Law
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law .

What qualifies a machine as an autonomous robot?

- Sensing and perception: get information from its surroundings
- Carry out different tasks: Move or pick place the objects
- Re-programmable: can do different things
- We here aim at building a machine which is capable of controlled locomotion.
- This machine may be human controlled or automatic.
- It must be able to perform certain tasks we set for it precisely.



What constitues a Robot?



BRAIN Microprocessor- controller	EYES Cameras / Light sensors / ranging sonar	EARS Sensors - sound / Infra-red light / magnetic fields
COMMUNICATIONS Data / video / sound transmitters + receivers	SKELETON Mechanical Frame	BALANCE Sensors - orientation
MUSCLES Hydraulic / electric / pneumatic - Actuators(arms etc)	FOOD POWER SOURCE / Charging device	LIMBS Wheels / Legs / Tracks / Propellors - motivation

Brain of the Robot=Microcontroller

Microcontrollers around us

- A microcontroller is s a *computer on a chip*, or if you prefer, a single chip computer .*Micro* suggests that the device is small, and *controller* tells that the device might be used to control objects, processes, or events.
- Another term to describe a microcontroller is ***embedded controller***; because the microcontroller and its support circuits are often built into, or embedded in, the device they control.
- You can find microcontroller in all kinds of things these days, any device that measures, stores, control, calculate or displays information is a candidate for putting a microcontroller inside.
- The largest single use for microcontroller is in automobiles – just think about ever car manufactured today include at least one microcontroller for engine control, and often more to control additional system in the car.
- In desktop computers, you can find microcontrollers inside keyboards, modems, printers and other peripherals.
- In test equipments, microcontroller makes it easy to add features such as ability to store measurements, to create and store user routines, and to display message and waveforms. Consumer products that use microcontrollers include cameras, video recorders, compacts disk players, and ovens. And these are just few examples.

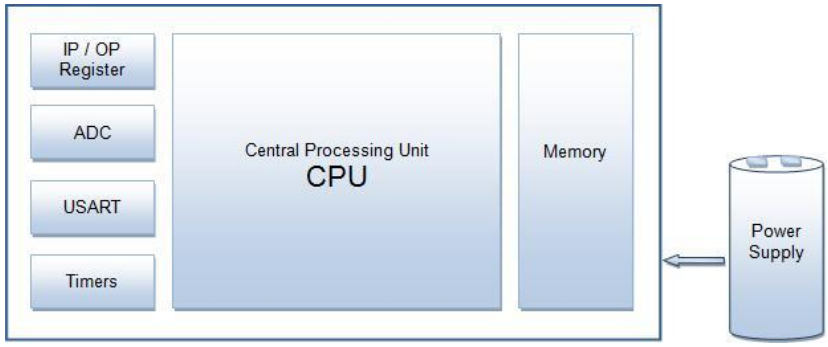
WHAT IS A MICROCONTROLLER?

Microcontrollers are basically *LSI* chips with millions of logic gates constituting an Arithmetic and Logic Unit (*ALU*), accompanying registers, bus control circuitry and an instruction decoder.

* This is the definition in the strictest sense.*

Most modern microcontrollers usually have on chip *RAM* and *ROM* and a few peripherals such as timers, serial interfaces such as a *UART*(Universal Asynchronous Receiver Transmitter), ports and even *ADC's*(Analog to Digital Converters).

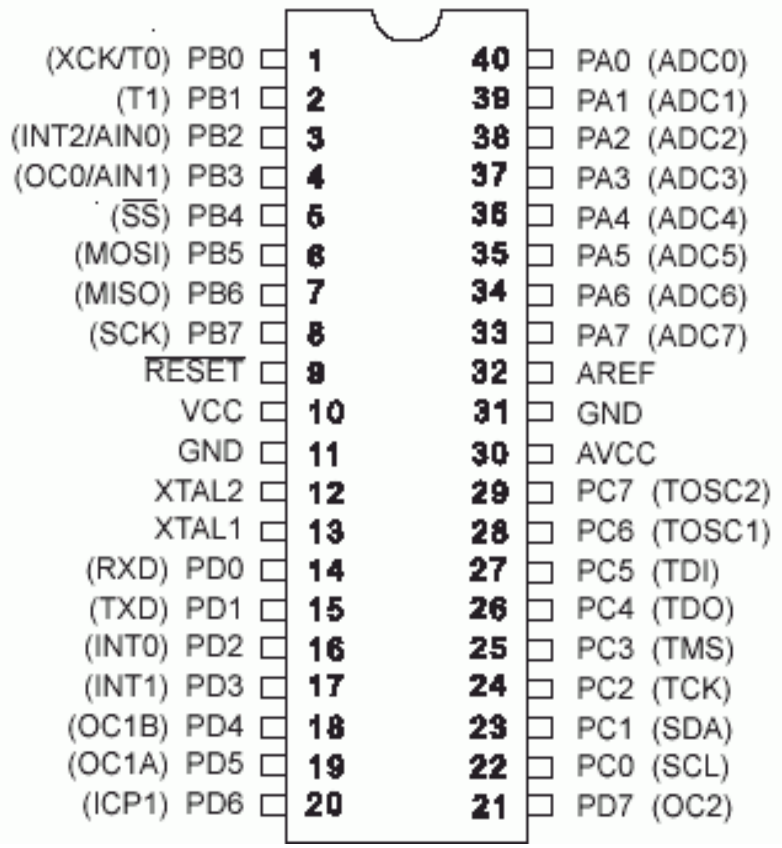
A wide variety of microcontrollers is available today. Some are general purpose, while some are optimized for special applications. A few well known microcontrollers are Intel's 8051, 8085, Pentium, Motorola's 68000 and 68008, Atmel's AVR series, PIC series……..In this workshop we are using Atmega8 (AVR series).



ATMEGA16 MCU

- We will be working on Atmega16 microcontroller, which is a 40-pin IC and belongs to the MegaAVR category of AVR family.
- **Some of the features of Atmega16 are:**
  - **16KB of Flash memory**
  - **1KB of SRAM**
  - **512 Bytes of EEPROM**
  - **Available in 40-Pin DIP**
  - **8- Channel 10-bit ADC**
  - **Two 8-bit Timers/Counters**
  - **One 16-bit Timer/Counter**
  - **4 PWM Channels**
  - **In System Programmer (ISP)**
  - **Serial USART**
  - **SPI Interface**
  - **Digital to Analog Comparator.**

PIN diagram-



Programming with microcontroller:-

Registers-

- AVR is 8 bit microcontroller. All its ports are 8 bit wide. Every port has 3 registers associated with it.
- Every bit in those registers configures pins of particular port.
- Bit 0 of these registers is associated with Pin 0 of the port, Bit 1 of these registers is associated with Pin 1 of the port… and like wise for other bits.
- These three registers are as follows : (x can be replaced by A,B,C,D as per the AVR you are using)
  - DDRx register
  - PORTx register
  - PINx register

DDR-

- Data Direction Registers
- Responsible for configuring the Pins as input/output.
- '1' means that the corresponding pin is configured as output.
- '0' means that the corresponding pin is configured as input.
- So,DDRx=0b00010011 means that the 1st,2nd,and 5th pin of PORT X have been configured as output, while the rest pins are configured as input i.e. they are ready to accept data.

PORT-

- They are directly connected to the pins.
- When the corresponding DDR pin is configured as output, then these registers are used for giving the value to the device(LEDs, Motors).
- If I want to pass '1' to 5th pin of X PORT of the microcontroller which is connected to my device(LEDs, Motors) , I will write:

  PORTx=0b00010000;

- But remember that before passing this value the 5th pin of X port is configured as output using the DDR of that PORT.

PIN-

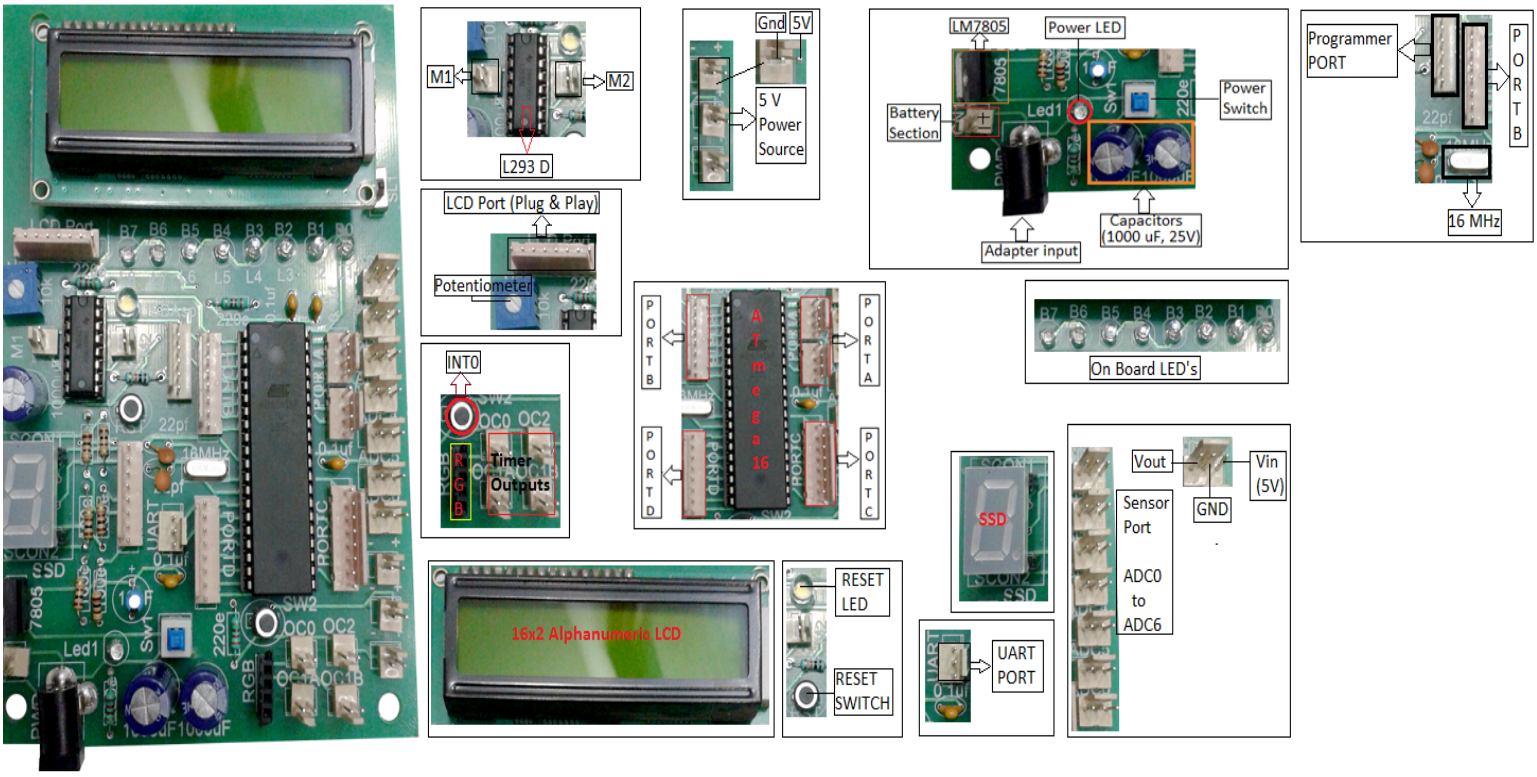
- They are directly connected to the pins.
- When the corresponding DDR pin is configured as input, then these registers are used for taking the value from the device (sensors).
- If I want to read '1' on 5th pin of X PORT of the microcontroller which is connected to my device(sensors) , I will write:

  PINx=0b00010000;

- But remember that before reading the value from the 5th pin of X port, the corresponding pin is to be configured as input using the DDR of that PORT.

MEGATron™ Board

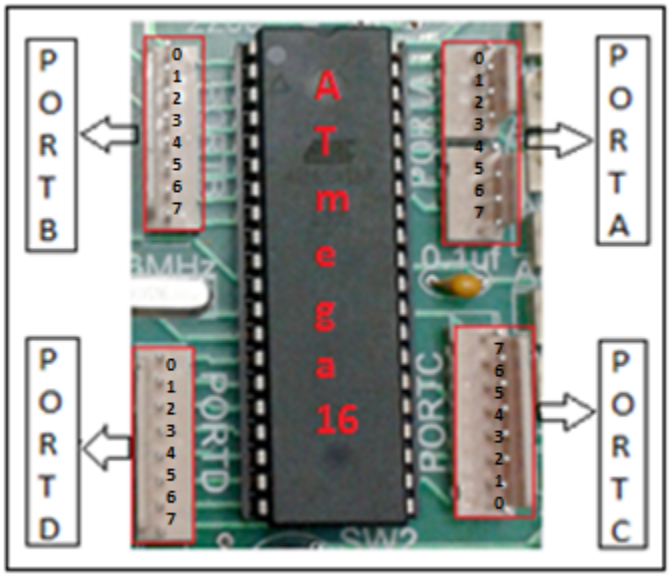
This development board is manufactured for learning the basic and advance concepts for beginners and professionals as well. This board contains a lot of features covering almost all section of AVR microcontroller. This board is suitable for many controllers like ATmega16, ATmega32, ATmega163, ATmega164P, ATmega1284P, ATmega324P, ATmega644, ATmega644P, ATmega8535 etc.



Features:

1. 4 Ports output connectors for general use.
2. 7 Ports for ADC.
3. 8 LED output for showing the pattern.
4. USART output.
5. On-Board Power supply.
6. 4 PWM output.
7. Hardware Interrupt.
8. Port for 16X2 alphanumeric LCD.
9. Buzzer for sound signal.
10. DC motor output using L293D.
11. Fair power supply distribution.
12. Touch screen connector.
13. On-board Seven Segment Display Connector point.
14. Connector for RGB LED

ATmega16 PORTs Pinout:



1st Program-

- **LED interfacing:-**

```
#include<avr/io.h>

int main()

{

DDRB=0b00000100; //declare portb2 pin as output

PORTB=0b00000100; //give value 1(5v) to portb

}
```

AVR studio:-

As we have been discussed in that embedded system is basically a combination of Hardware with the software. Here combination means that software is buried inside the hardware to perform specific tasks automatically as shown in figure below.

For performing different task on the computer, a compiler is communication medium that translate source language into target language. Compiler allows the user to perform customized task on machine. Initially for writing compiler machine language was used. After some development assembly language and these days high level language are used for

writing compiler. We need compiler because it allows us to communicate with hardware. It is also use to cover the "GAP" between Humans and the computer language. Computer can understand only one language that binary language consists of only two digits are 0 and 1.

Binary language is also called machine language. When there is only Machine Language then programmers write their compilers in this language. But it is very difficult and tedious job. The role of compiler is take source code written in high level language (Java, C++, VB.Net etc). The High Level Languages are easily understood by humans. So compiler converts the program written in formal language (Source language) into machine language (target language). As we know that computers can easily understand machine language. There are different programs related to compiler that works before compilation such as editor, preprocessor, assembler, linker or loader, debugger and profiler.

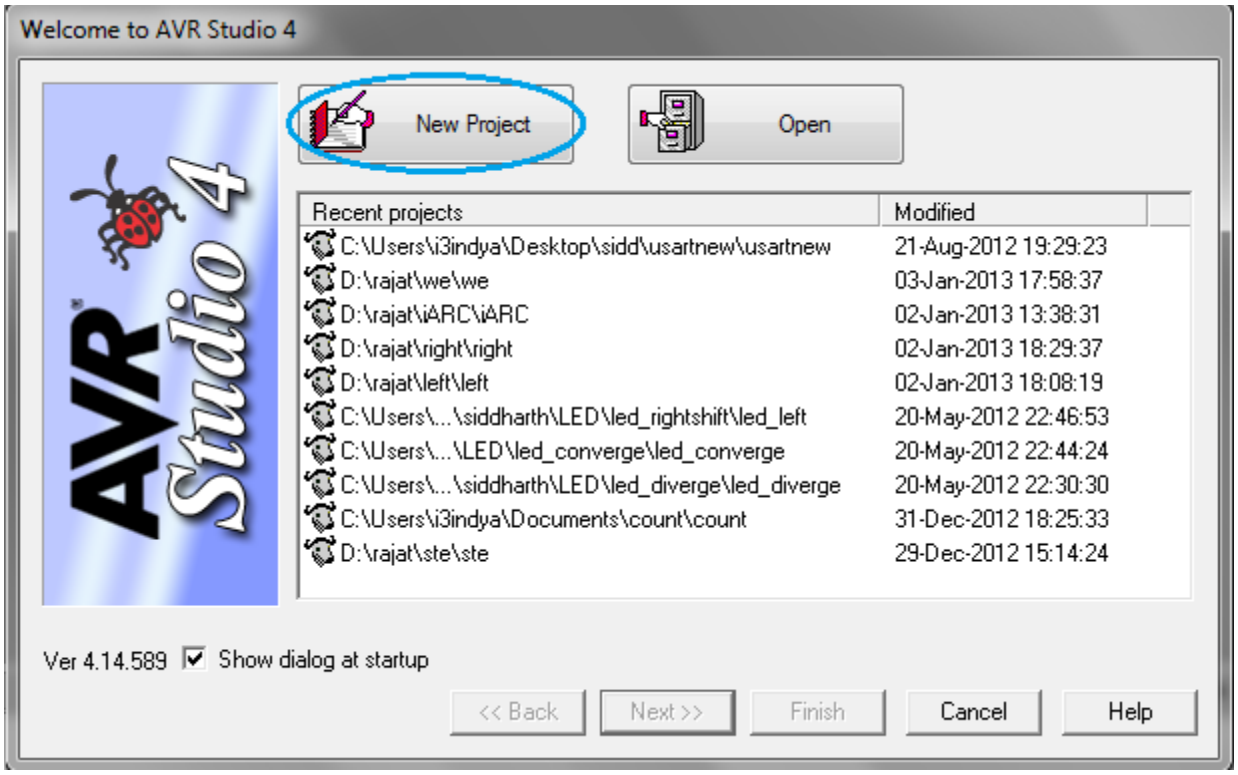
Here we use AVR studio for writing program and AVR GCC for compilation

So

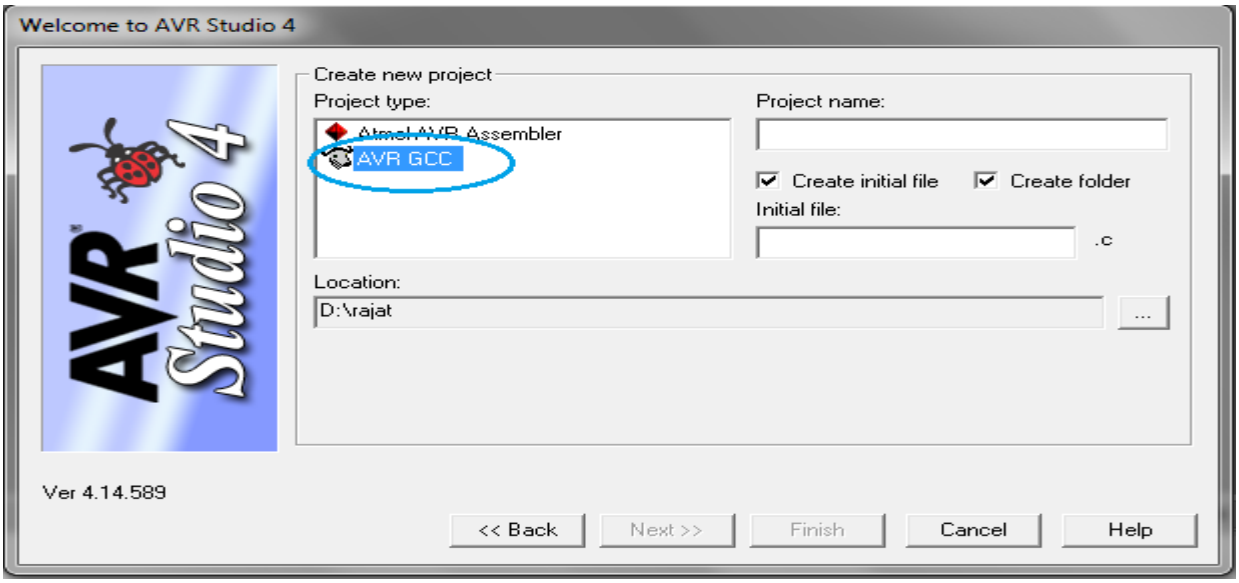
1. Install WINAVR
2. Install AVR studio

Now open AVR studio-

1. New Project



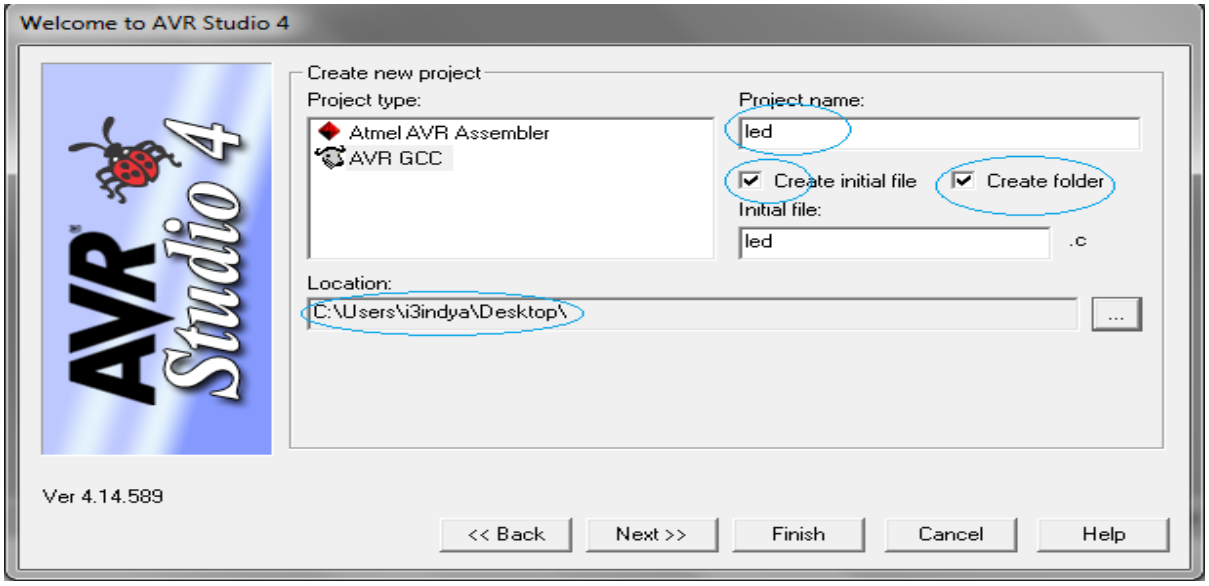
2. Select AVR GCC compiler



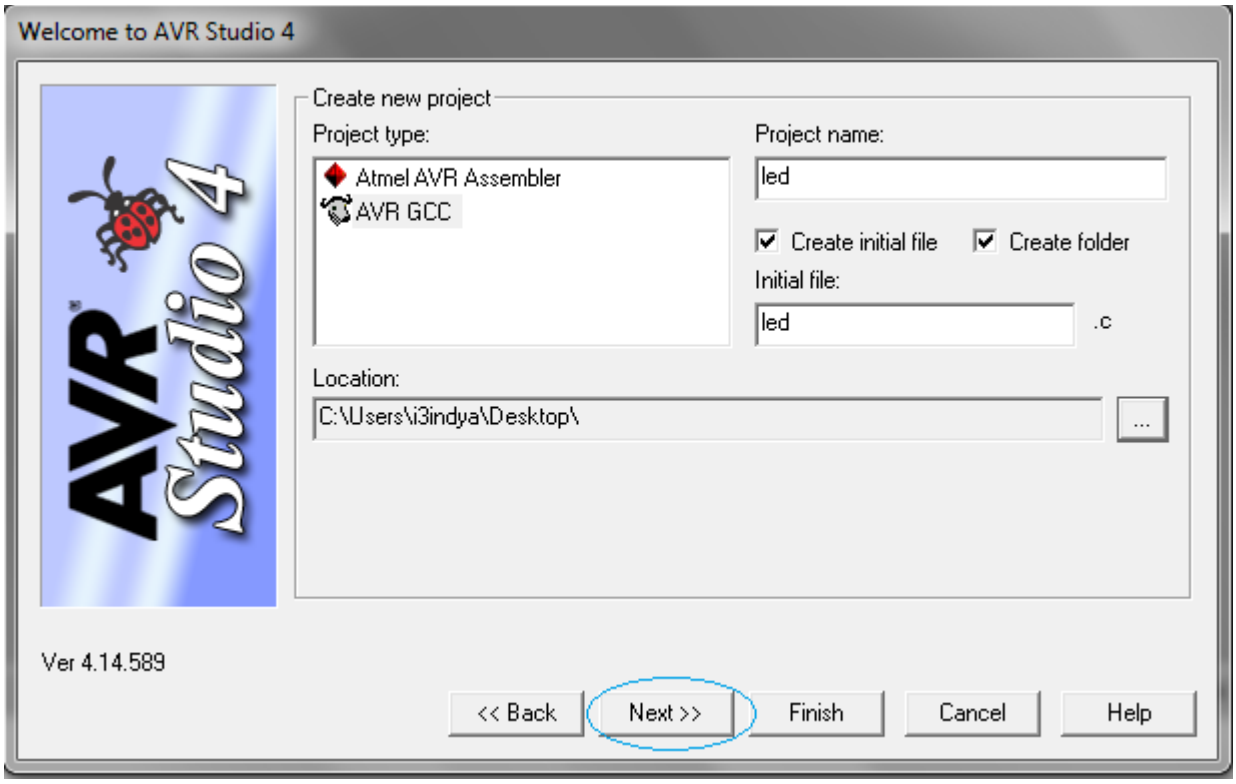
3. Write project name (anything like: led)

* Don't give space while writing the project name.

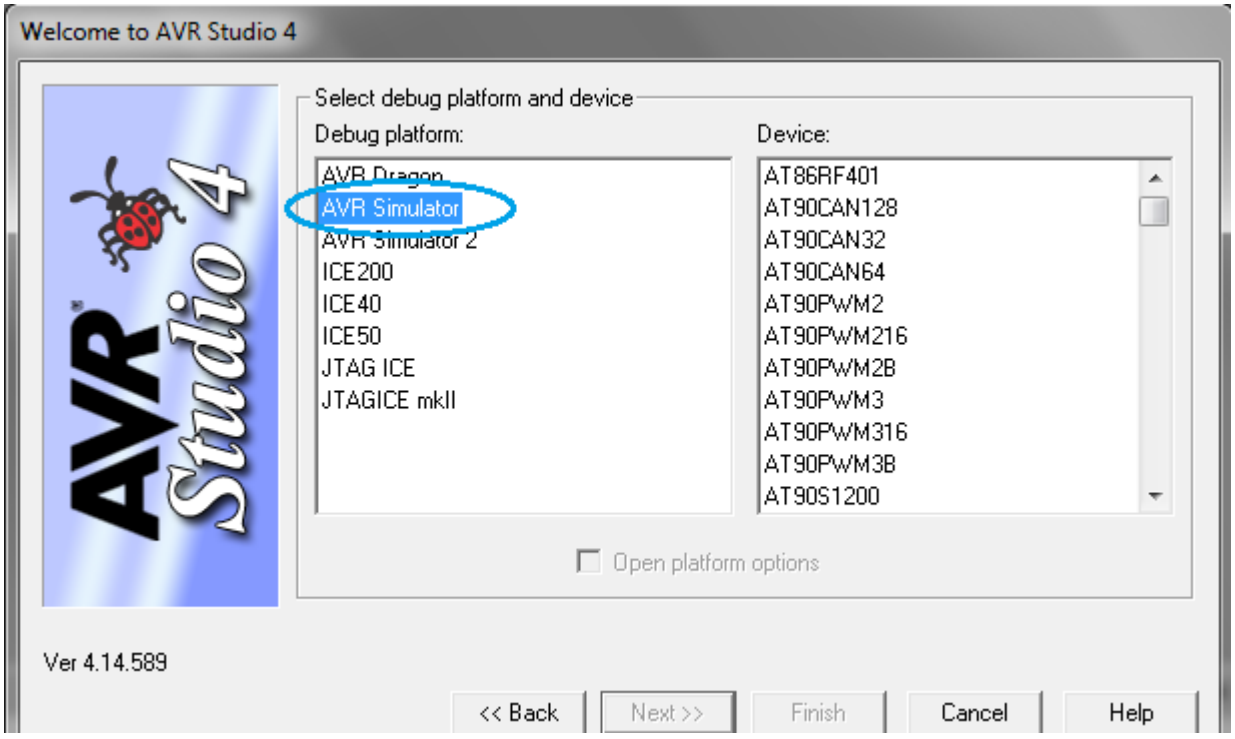
Select create initial file, create folder and give location (where you want to store program)



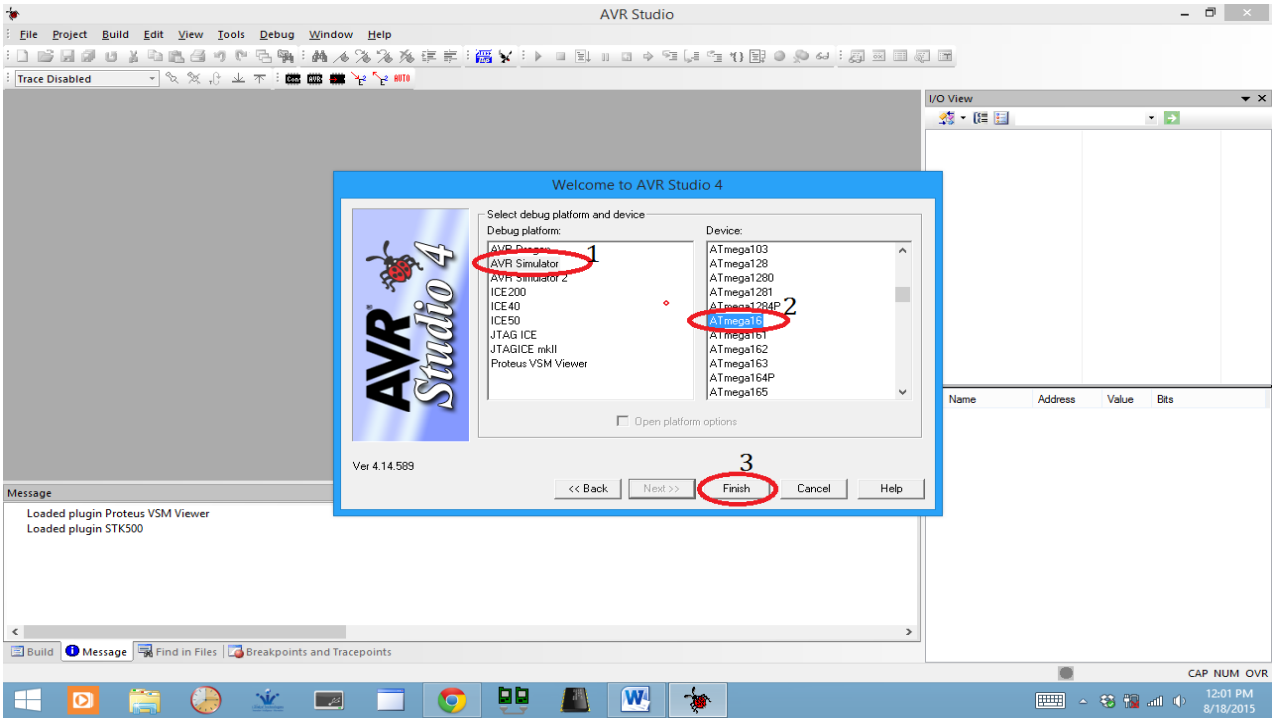
4. Select next



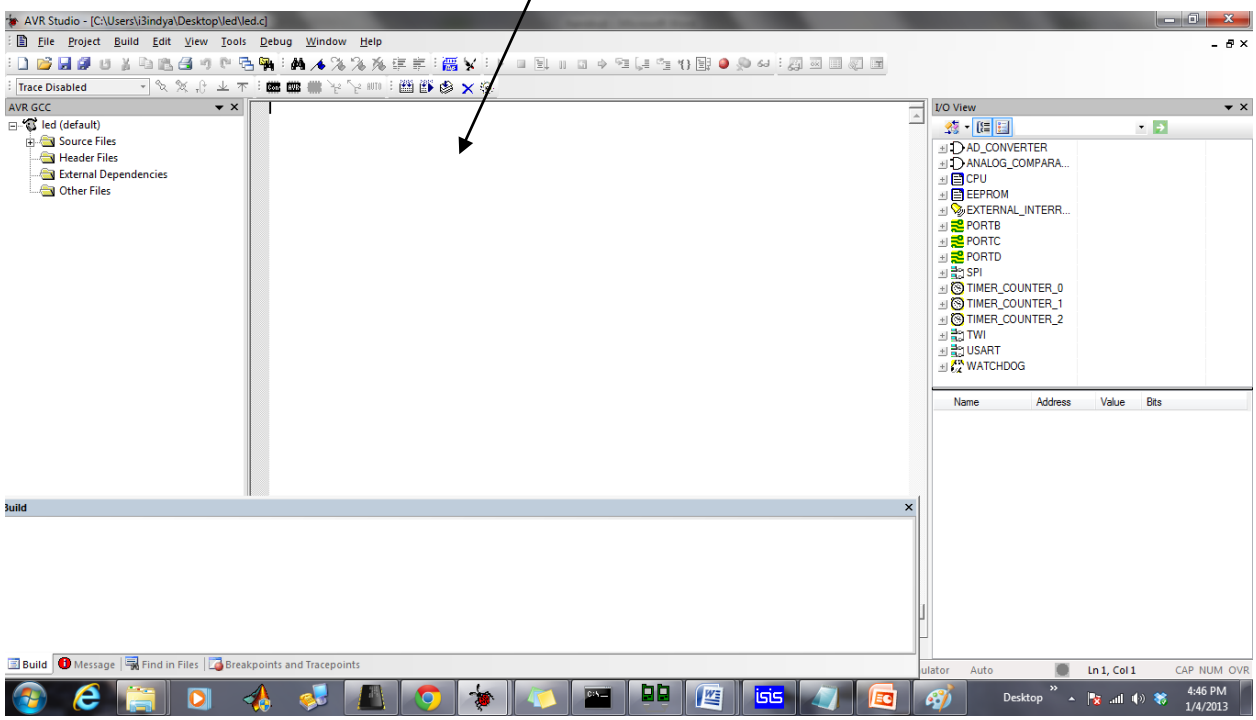
5. Then select AVR simulator



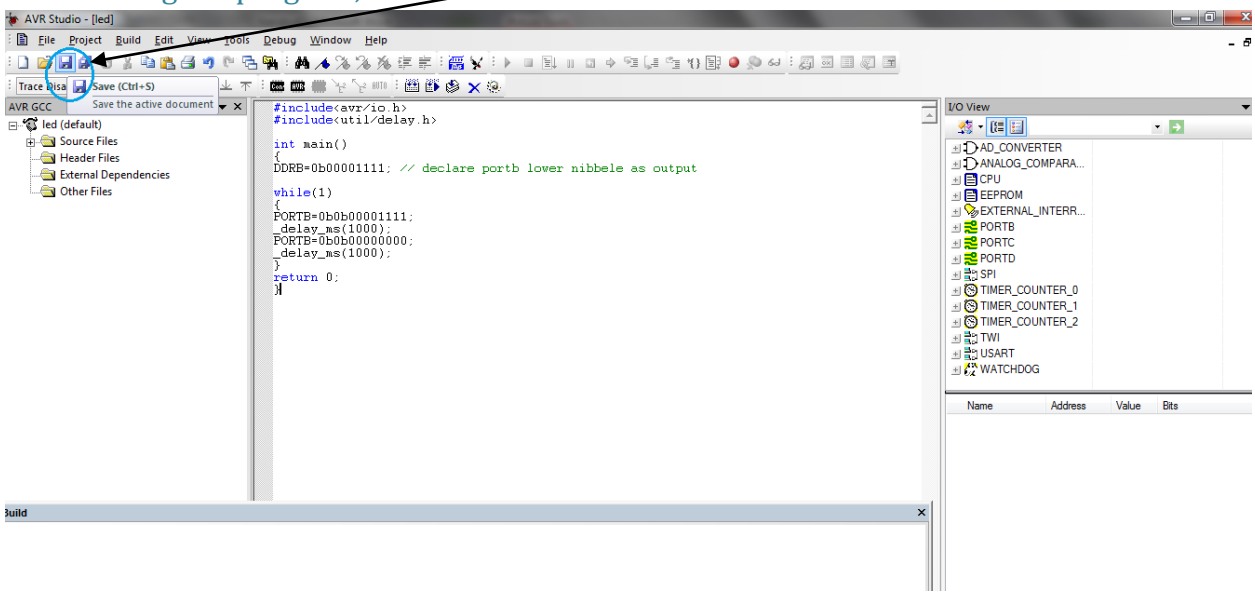
6. Then in right side select AT mega16 and click on finish



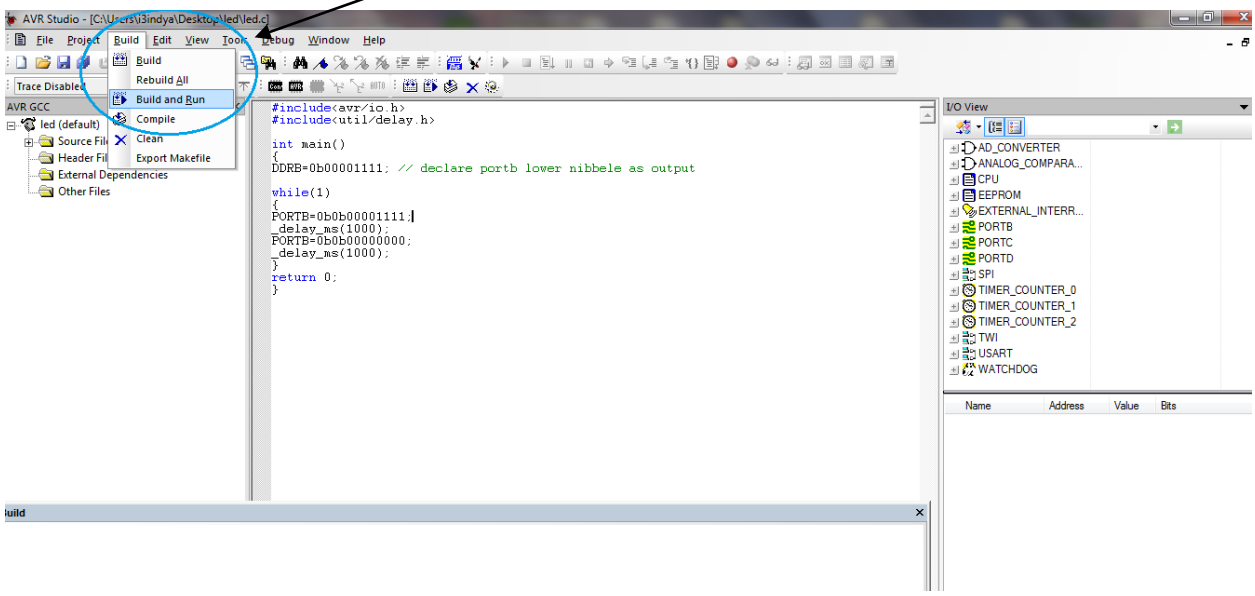
7. Now this is the plate form to write program



8. After writing the program, click on save it



9. Then click on Build and then build and run

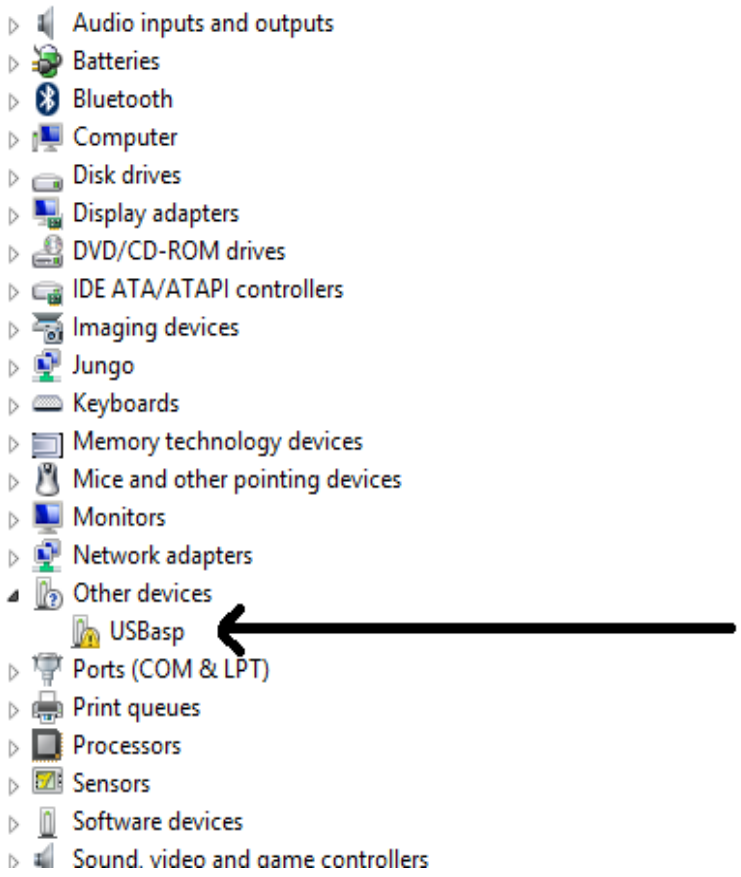


Now your program is compiled and a HEX file is generated in store folder

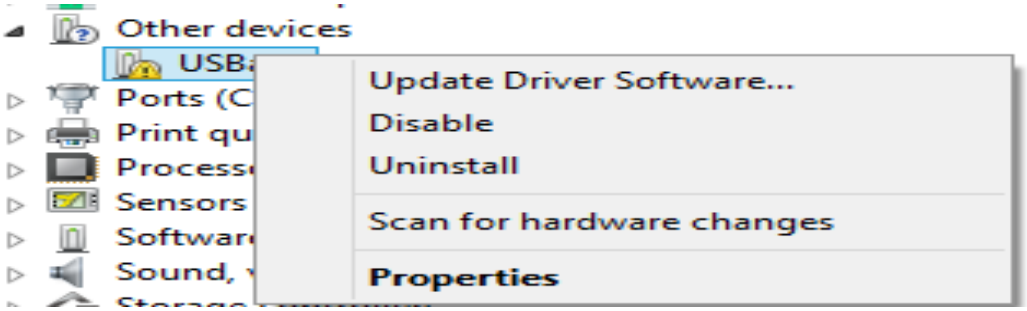
We have made all the hardware requirements for our first program! Next we will see how to configure our PC for Programming.

We need programmer drivers for connecting our programmer with PC. . As said earlier we are using the programmer USBasp. So we will see from where to get the drivers for this programmer. Follow these steps.

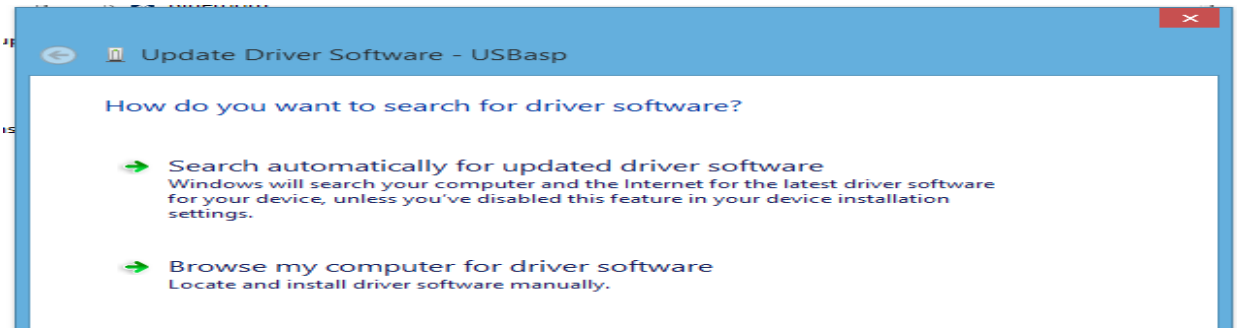
➔ Insert USBasp programmer, then open Device Manager. There you'll find USBasp in yellow mark.



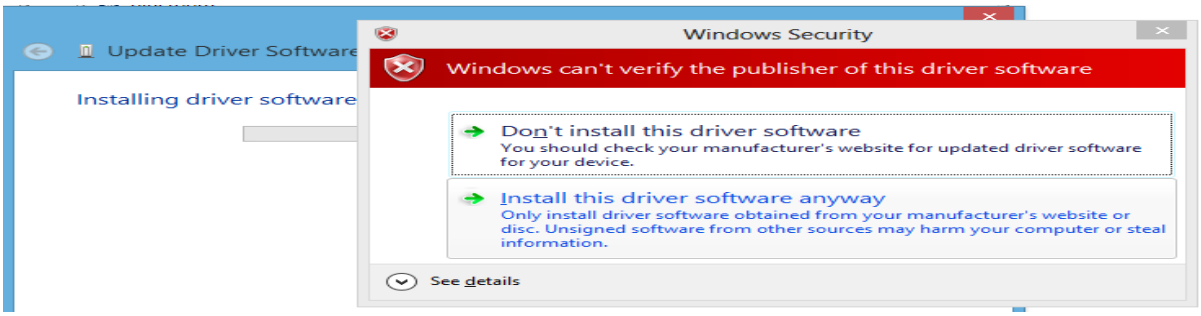
➔ Right click on the yellow marked USBasp, and then click on Update Driver.



➔ Now update the driver using "Browse my computer for Driver Software"

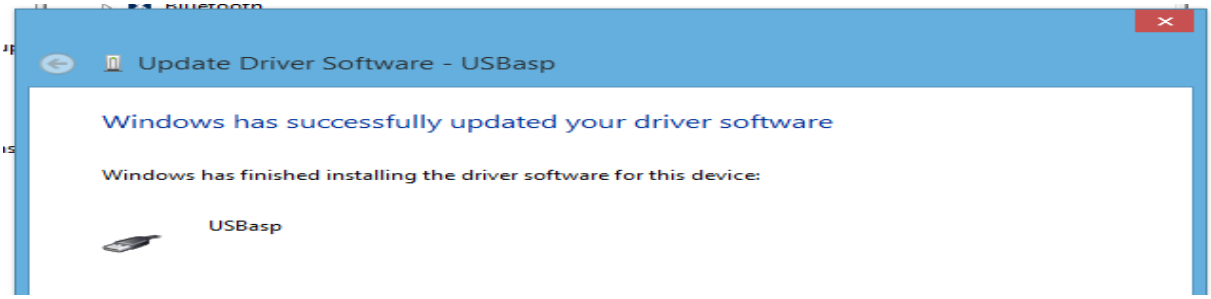


➔ Provide the location where the Driver for USBasp exists. (Refer to the CD provided.)



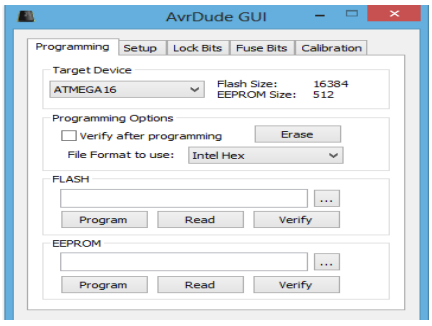
➔ Select "install this driver software anyway" option.

➔ After installation close the window.

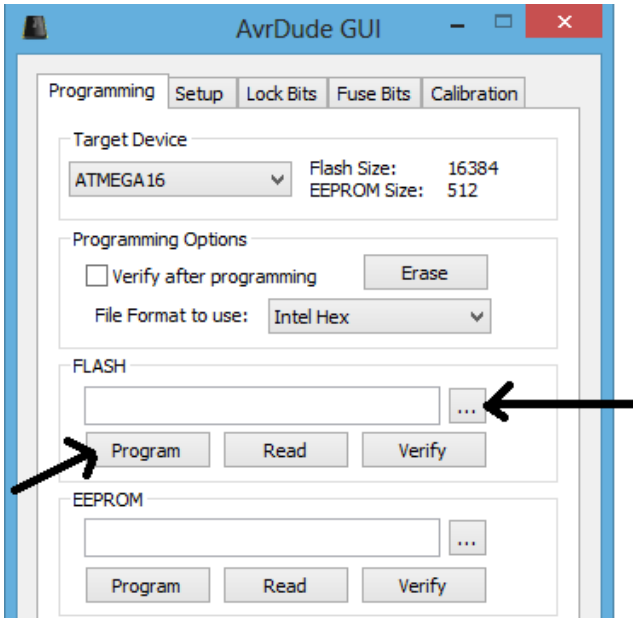


Getting the microcontroller ready for programming.

➔ After installation open "AVR Dude GUI" folder from the CD.



- Burn the program (.HEX file) on "Flash."



➢ MOTOR INTERFACING

An electric motor is an electromechanical device that converts electrical energy into mechanical energy. Most electric motors operate through the interaction of magnetic fields and Current-carrying conductors to generate force. These are the following type of motors.

1. DC motor
2. Stepper motor
3. Servo motor
4. Universal motor

DC motor:-

A DC motor is an electric motor that runs on direct current (DC) electricity. DC motor has simply two wires after giving them DC supply. It will rotate in one direction and its direction can be changed by changing the polarity. DC motors can operate directly from rechargeable batteries, providing the motive power for the first electric vehicles.

Principle of operation:-

- In any electric motor, operation is based on simple electromagnetism.
- The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

We don't connect DC motor with microcontroller directly because to run motor needs some more current than the output of microcontroller. To motor drive we need current but for the motor speed we need voltage. So to connect motor with microcontroller we need an interfacing IC which is also called driver IC. There are different types of driver IC like L293D, L298D, ULN2803 and so many….

We use L293D as a driver IC to interface with microcontroller-

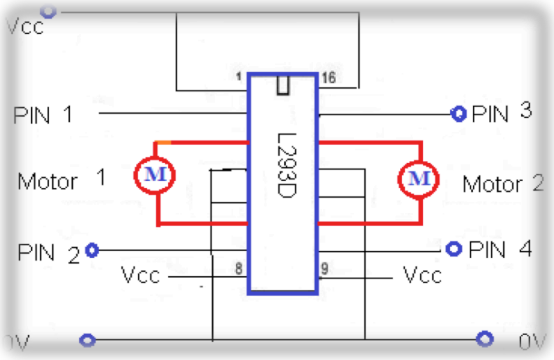
L293D- The L293D is ideal for controlling the forward/reverse/brake motions of small DC motors controlled by a microcontroller such as an AVR, 8051 microcontroller. The L293D is a quadruple push-pull 4 channel driver capable of delivering 600mA per channel. One l293D is capable of driving two DC motor at a time.

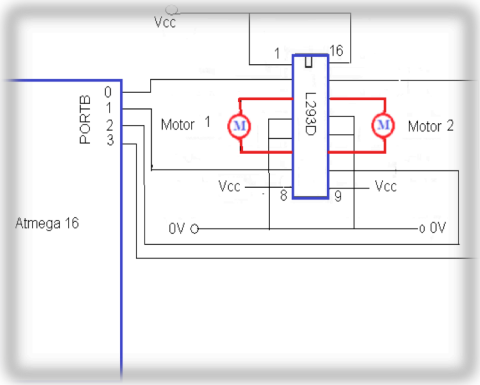
Features:-

- 600mA output current capability per driver.
- Wide supply voltage range 4.5 volt to 36 volt.
- Separate input-logic supply.

PIN diagram- It is a 16 pin IC in which four input pins and four output pins. This single IC has a property to drive two Dc motor at a time. If we give 1st pin of this Ic to Vcc then left part of this IC is enable and if we give Vcc to pin no 9 then right part of this IC is enable. And if we provide both pins 1& 9 Vcc then both sides input are enable.

ENABLE	1	16	V_{CC}
INPUT 1	2	15	INPUT 4
OUTPUT 1	3	14	OUTPUT 4
GND	4	13	GND
GND	5	12	GND
OUTPUT 2	6	11	OUTPUT 3
INPUT 2	7	10	INPUT 3
Vs	8	16	ENABLE





We connect four input pin of L293D with any PORT of the Atmega 8. Here we interface L293d with 2nd, 3rd ,4th and 5th pin of PORTB.

Program- To rotate our motor in forward and backward direction

```
int main()
{
DDRB=0xff;
  While(1)
   {
      PORTB=0b00100100;    //both motors move in forward direction
      _delay_ms (5000);         // forward movement for 5 second
     PORTB=0b00011000;    // both motors move in reverse direction
      _delay_ms (5000);        // backward movement for 5 second
   }
return 0;
}
```

Program- To rotate our robot forward, right, left and backward direction

```
int main()
{
DDRB=0xff;
  While(1)
   {
      PORTB=0b00100100;      //Robot move in forward direction
      _delay_ms (5000);          // forward movement for 5 second
     PORTB=0b00011000;       // Robot move in reverse direction
      _delay_ms (5000);          // backward movement for 5 second
```

```
        PORTB=0b00100000;      //Robot move in right direction
        _delay_ms (5000);            // right movement for 50  second
       PORTB=0b00000100;      // Robot move in left direction
        _delay_ms (5000);            // left movement for 5 second
           }
return 0;
}
```

➢ LCD INTERFACING

LCD stands for "**Liquid Crystal Display**". As the name suggest that liquid crystal display, so it's a kind of display which is made by using liquid and crystal. Wonder 'Liquid Crystal'? It's basically a new state of matter 'Forth State', in which a type of crystal is melt up to a stage where it is at converting stage in liquid. This state of matter is quite useful for displaying values. There are numerous types of liquid used for this purpose as pneumatic liquid and so on. Pneumatic liquid works on the concept of polarization which consists with twisted and untwisted pairs, means there'll be twisted and untwisted pair crystals of this pneumatic liquid, whenever any datum is to be printed onto display that time these twisted pair crystal turns to untwisted pair and then light will block which is continuously passing through backlight LED. As these crystal becomes untwisted that particular crystal block the light and that datum is printed on display.

Let's not confuse with this technology of polarization for displaying any data. As we are talking about embedded systems. so, we come directly on the topic. There are two types of LCD's used now a days for displaying data, even in Nokia display screens in 90's used.

1. Alphanumeric LCD
2. Graphic LCD

Let's talk about alphanumeric LCD first. There are various models available for alphanumeric LCD in market now days like 16*1, 16*2, 16*4 and so many. Let's see what their names tells about them, a 16*2 LCD means that there are 16 columns and two rows in which 16 characters can be printed in a single line and there are two lines. So, 32 characters are visible on this LCD at a time.

Alphanumeric LCD

The diagram shown above is for Alphanumeric LCD. There'll lines for displaying characters and at a time only 16 characters can be displayed for 16*x series Alphanumeric LCD.

Pin Description:

Pin. No	Function	Name
1.	Ground (0V)	Ground
2.	Supply voltage; 5V (4.7V – 5.3V)	VCC
3.	Contrast adjustment; through a variable resistor	VEE
4.	Selects command register when low; and data register when high	Register Select
5.	Low to write to the register; High to read from the register	Read/ Write
6.	Sends data to data pins when a high to low pulse is given	Enable
7.	Data bit-0	DB0
8.	Data bit-1	DB1
9.	Data bit-2	DB2
10.	Data bit-3	DB3
11.	Data bit-4	DB4
12.	Data bit-5	DB5
13.	Data bit-6	DB6
14.	Data bit-7	DB7
15.	Backlight V_{CC} (5V)	LED+
16.	Backlight Ground (0V)	LCD-

Before going to start interfacing part we should first know how characters can display on display screen and which type of characters can display. Actually there is a limitation on the characters that can be displayed, we can't display Urdu, Arabic, Punjabi characters and so on. English characters and numbers, arrow keys and some more character only can displayed. But if we want to display any special characters like: bell, any special figure, these can also be displayed but first we would be creating these special characters and then would be using same like we'll use other characters.

LCD works in two mode

1. 4 bit mode
2. 8 bit mode

We are here using 4 bit mode means we are using 4 data pins for sending data and commands to lcd

➔

Now before printing something in our LCD open your software folder from your cd

Copy "i3indya" folder and paste it on

>> **C:\WinAVR-20080610\avr\include**

Now write down your first program to print on LCD

>>Program- to print a string on LCD

>>

```
#include<avr/io.h>
#include<util/delay.h>
#include<i3indya/lcd16.h>
 int main()
{
lcd_init(); //initialize the lcd and select port on which you want to connect
while(1)
{
lcd_gotoxy(0,1);    //select the position where you want to print character
lcd_printc("String");
}
return 0;
}
```

➢ SENSOR INTERFACING

A sensor is an instrument that responds to a physical stimulus (such as heat, light, sound, pressure, magnetism, or motion.) It collects and measures data regarding some property of a phenomenon, object, or material. Typical sensors are cameras, radiometers and scanners, lasers, radio frequency receivers, radar systems, sonar, thermal devices, seismographs, magnetometers, gravimeters…….In robotics also there are too many sensors we are going to discuss here IR sensor

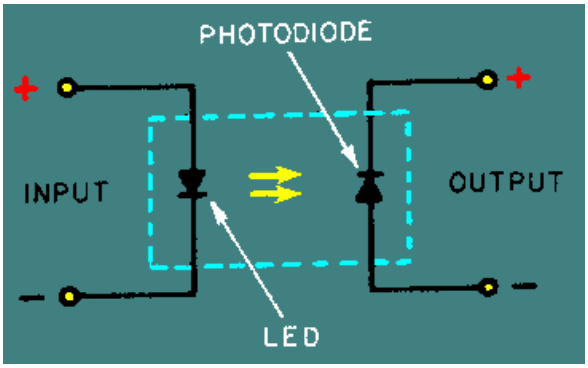
IR sensor- IR sensor is the combination of IR transmitter and IR receiver which works on the phenomena of photo diode

PHOTO DIODES: A **photodiode** is a type of photo detector capable of converting light into either current or voltage, depending upon the mode of operation.

Most photodiodes will look like the picture to the right, that is, similar to a light emitting diode. They will have two leads, or wires, coming from the bottom. The shorter end of the two is the cathode, while the longer end is the anode.

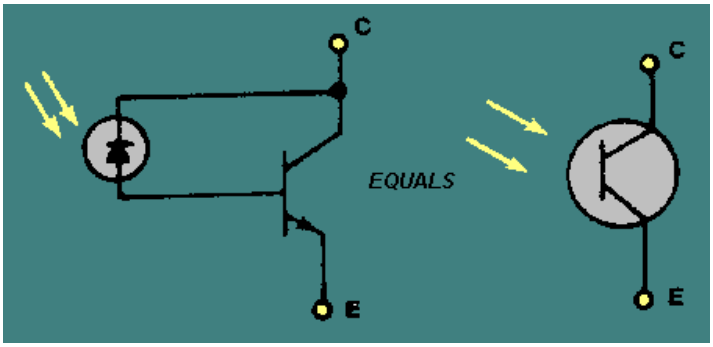
Principle of operation

A photodiode is a PN junction or PIN structure. When a photon of sufficient energy strikes the diode, it excites an electron thereby creating a mobile electron and a positively charged electron hole. If the absorption occurs in the junction's depletion region, or one diffusion length away from it, these carriers are swept from the junction by the built-in field of the depletion region. Thus holes move toward the anode, and electrons toward the cathode, and a photocurrent is produced.

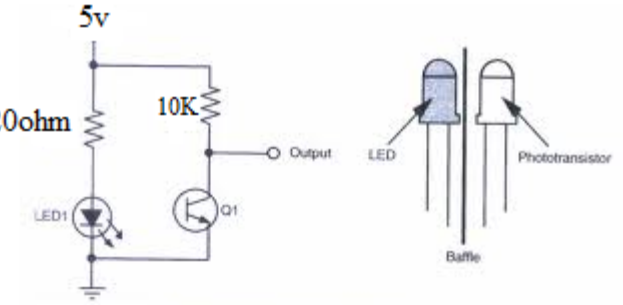


PHOTOTRANSISTOR:

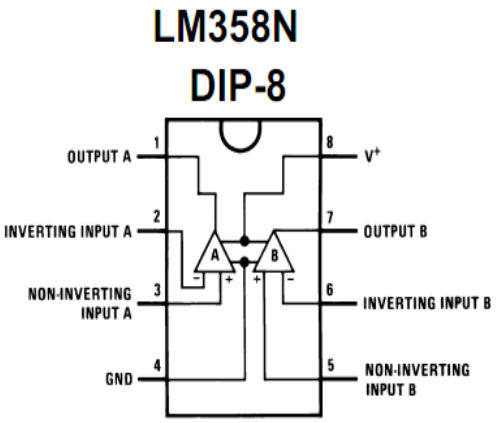
A second optoelectronic device that conducts current when exposed to light is the PHOTOTRANSISTOR. A phototransistor, however, is much more sensitive to light and produces more output current for a given light intensity that does a photodiode. Figure shows one type of phototransistor, which is made by placing a photodiode in the base circuit of an NPN transistor. Light falling on the photodiode changes the base current of the transistor, causing the collector current to be amplified. Phototransistors may also be of the PNP type, with the photodiode placed in the base-collector circuit.



The complete circuit of IR sensor in analog mode



To convert this analog output to digital output we need a comparator IC LM358.



Sample program-

Obstacle avoider- In this program when any obstacle is in front of our robot it will turn to right otherwise it rotating in forward direction.

```
#include<avr/io.h>
#include<util/delay.h>
int main()
{
DDRC=0x00;  //declare port A as input
DDRB=0xff;   // declare PORTB as output port
while(1)
{

int x=PINC & 0B00000001;

if(x==0x01)

{PORTB=0b00001000;} // if obstacle is in front of robot then right turn

else

{PORTB=0b00100100;}    //else our robot move forward direction

 }

return 0;

}
```

Line follower

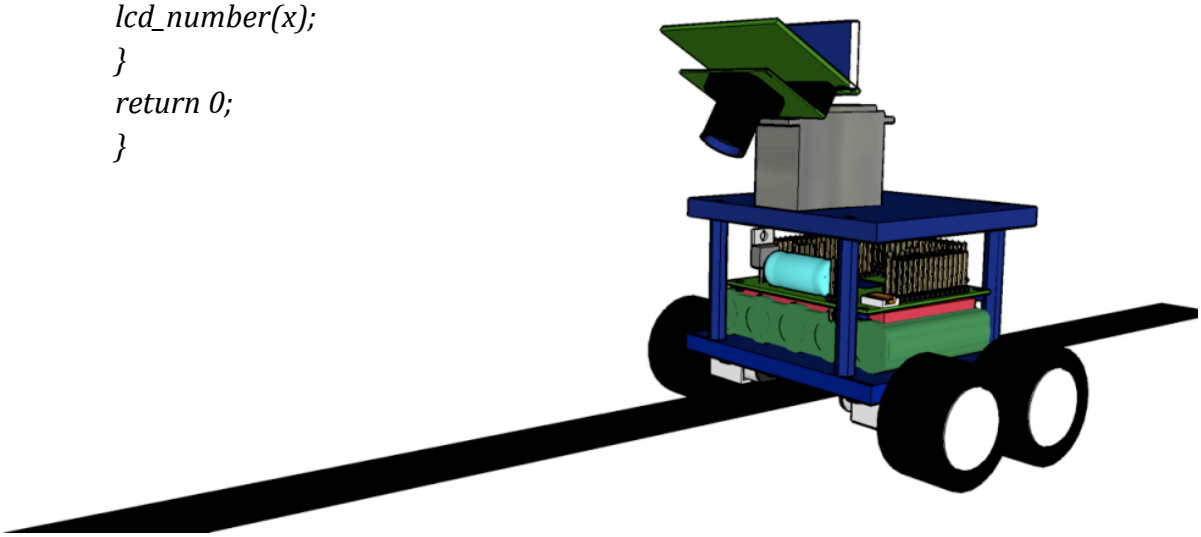
```
#include<avr/io.h>
#include<util/delay.h>

int main()
{
DDRC=0x00;
DDRB=0xff;
while(1)
{
```

```
int e=PINC & 0b00000011;
if(e==0b00000011){PORTB=0b00100100;}
else if(e==0b00000001){PORTB=0b00000100;}
else if(e==0b00000010){PORTB=0b00100000;}
else{PORTB=0x00;}
}
return 0;
}
```

Sensors in Analog Mode:-

```
#include<avr/io.h>
#include<util/delay.h>
#include<i3indya/lcd16.h>
#include<i3indya/adc16.h>
 int main()
{
lcd_init(); //initialize the lcd and select port on which you want to connect
adc_init();
while(1)
{
lcd_gotoxy(0,1);    //select the position where you want to print character
lcd_printc("ADC_Value");
int x=adc_read(0); //first cannel ADC read value
lcd_gotoxy(3,1);
lcd_number(x);
}
return 0;
}
```



i3indya Distance Learning Program

To know click HERE

To know click HERE

Talk to Expert

For more details call at
+91 - 956060 5666
(Extension : 3)
Timings : 10am - 8pm

REGISTRATION PROCEDURE FOR TRAINING

»**Register for Embedded Systems & Robotics Winter Training online. Click Here.**

»**Register for Ethical Hacking & Cyber Security Winter Training online. Click Here.**
» **Download** the Training Registration Form here
» To confirm your seat send Training Registration Form & Course Fee via D.D. in favor of **"I THREE INFOTECH PVT LTD"** payable at New Delhi to the following address:

i3indya Technologies
C-56, 3rd Floor above Bata Showrrom
Metro Pillar no. 81, Preet Vihar, New Delhi - 110092