Introduction To Robotics

Kushagra Nigam-kushagran1@gmail.com

Difference between Microporcessor and Controller

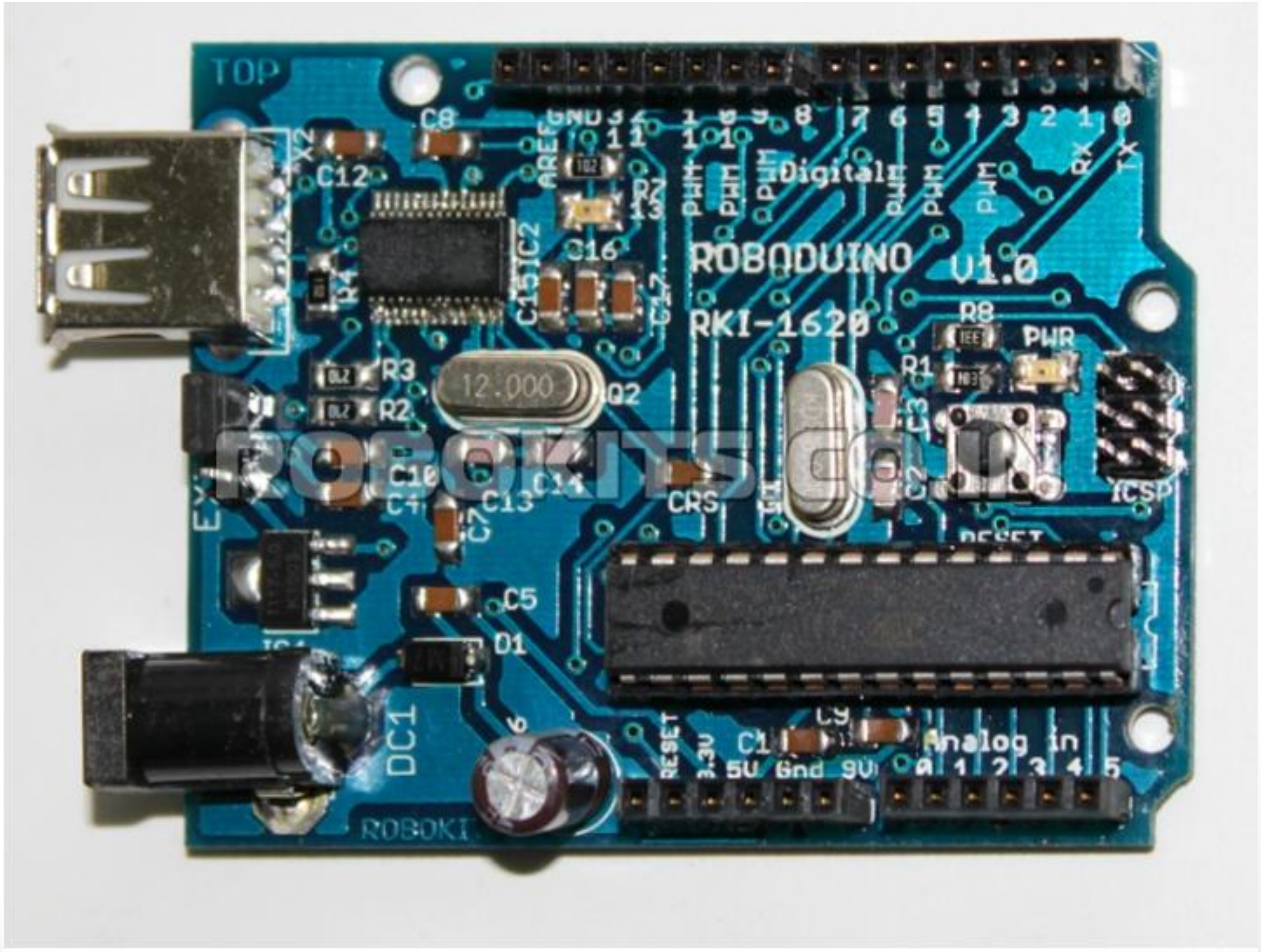
- **Microprocessors:** generally require external components to implement program memory, ram memory and Input/output. Intel's 80186, 80188, and 80386 are examples of microprocessors.

- **Microcontrollers:** incorporate program memory, ram memory and input/output resources internal to the chip. Microchip's pic series and Atmel's AVR series are examples of microcontrollers.

- **Embedded System:** any system consisting of software, hardware and a specific purpose

Kushagra Nigam-
kushagran1@gmail.com

Roboduino Arduino based Board



Kushagra Nigam-
kushagran1@gmail.com

What Is Arduino?

- A free development system based on Atmel AVR 8 bit microcontrollers.

TECH SPECS

Microcontroller	ATmega168
Operating Voltage	2.7-5.5 V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16 KB (ATmega168) of which 0.5 KB is used by bootloader
SRAM	1 KB (ATmega328)
EEPROM	512 B (ATmega168)
Clock Speed	16 MHz

Kushagra Nigam-
kushagran1@gmail.com

Arduino Memory

- **Flash memory**: it's a rewritable non-volatile memory. This means that its content will still be there if you turn off the power. It's a bit like the hard disk on the arduino board. Your program is stored here.

- **RAM:** it's like the RAM in your computer. Its content disappears when you turn of the power but it can be read and written really fast. Every normal variable in your sketch is held in RAM while your sketch runs.

- **EEPROM:** (Electrically Erasable Programmable Read-Only Memory)It's normally used to store settings and other parameters between resets. This memory supports at least 100,000 writes.

Pin Configuration

1)Power pins

- **VIN**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source).

- **5V**. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

- **3V3**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

- **GND**. Ground pins.

Pin Configuration

2)Input and Output

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

-PWM: Provide 8-bit PWM output with the analogWrite() function.

-**LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

Pin Configuration

- **AREF**. Reference voltage for the analog inputs. Used with analogReference().
- **Reset**. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.
- **Analog Pins:**These are used for reading analog inputs.
- **Digital Pins:** The pins on the Arduino can be configured as either inputs or outputs.

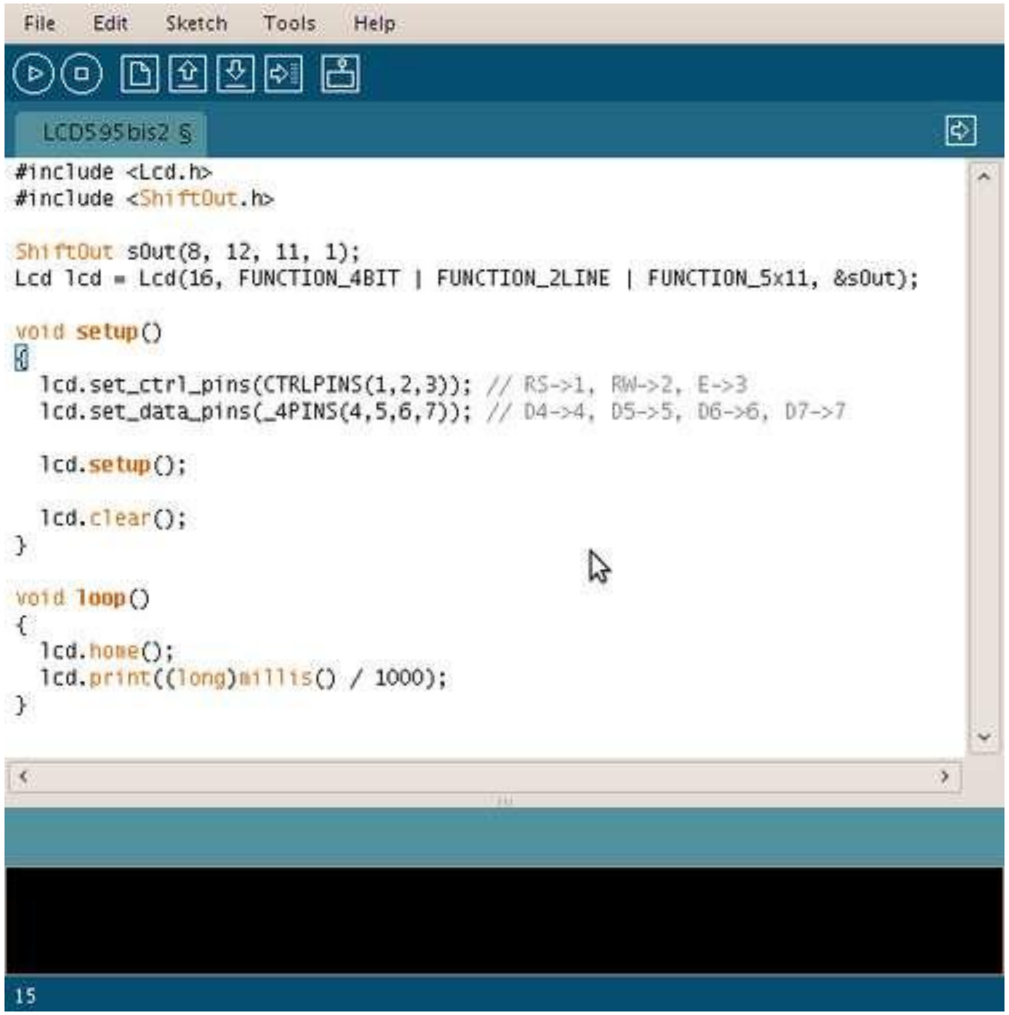
Kushagra Nigam-
kushagran1@gmail.com

Digital Voltage levels

- Digital Low-> 0 (Roughly 0V to 0.7V).
- Digital High-> 1 (Roughly 2.3V to 3.3V).

Kushagra Nigam-
kushagran1@gmail.com

Arduino IDE



```
#include <Lcd.h>
#include <ShiftOut.h>

ShiftOut sOut(8, 12, 11, 1);
Lcd lcd = Lcd(16, FUNCTION_4BIT | FUNCTION_2LINE | FUNCTION_5x11, &sOut);

void setup()
{
  lcd.set_ctrl_pins(CTRLPINS(1,2,3)); // RS->1, RW->2, E->3
  lcd.set_data_pins(_4PINS(4,5,6,7)); // D4->4, D5->5, D6->6, D7->7

  lcd.setup();

  lcd.clear();
}

void loop()
{
  lcd.home();
  lcd.print((long)millis() / 1000);
}
```

Basic Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference() –To change the upper limit of reference voltage(0-5V by default) for ADCs on analog pins
- analogRead()
- analogWrite() – PWM

analogReference(type)

<u>type:-</u>

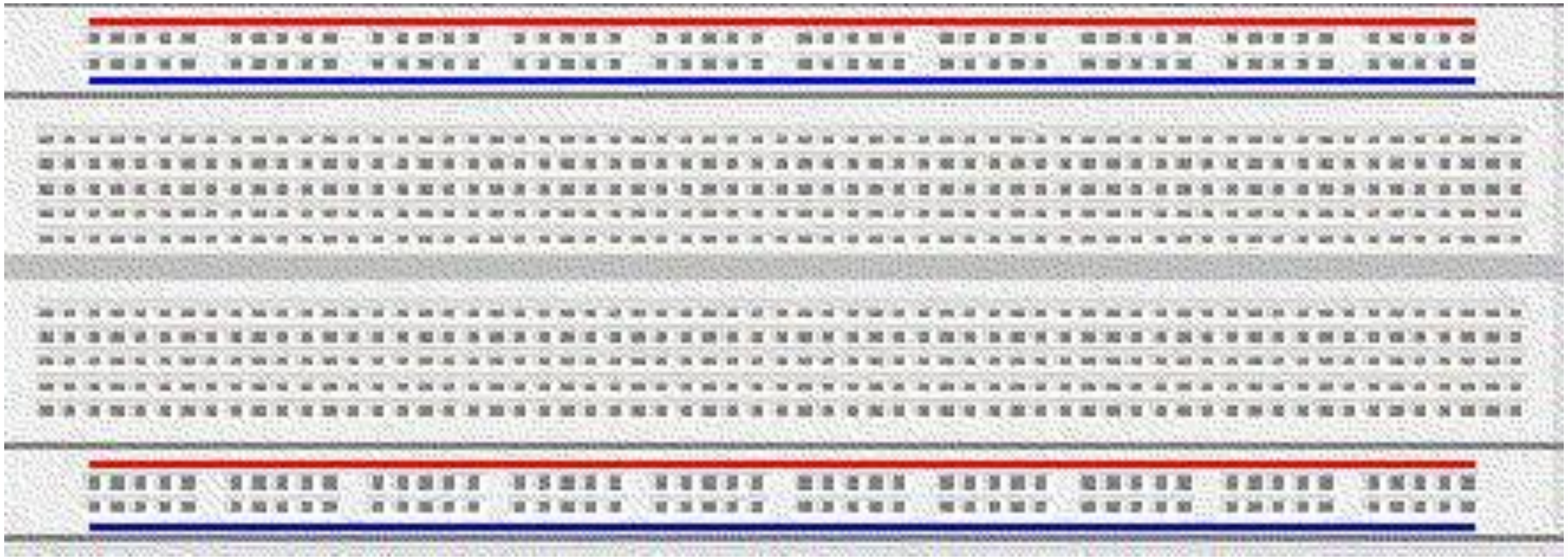
- DEFAULT: the default analog reference of 5 volts (on 5V Arduino boards) or 3.3 volts (on 3.3V Arduino boards)
- INTERNAL: an built-in reference, equal to 1.1 volts on the ATmega168 or ATmega328 and 2.56 volts on the ATmega8 (not available on the Arduino Mega)
- INTERNAL1V1: a built-in 1.1V reference (Arduino Mega only)
- INTERNAL2V56: a built-in 2.56V reference (Arduino Mega only)
- EXTERNAL: the voltage applied to the AREF pin (0 to 5V only) is used as the reference.

Basic Functions

Serial Communication

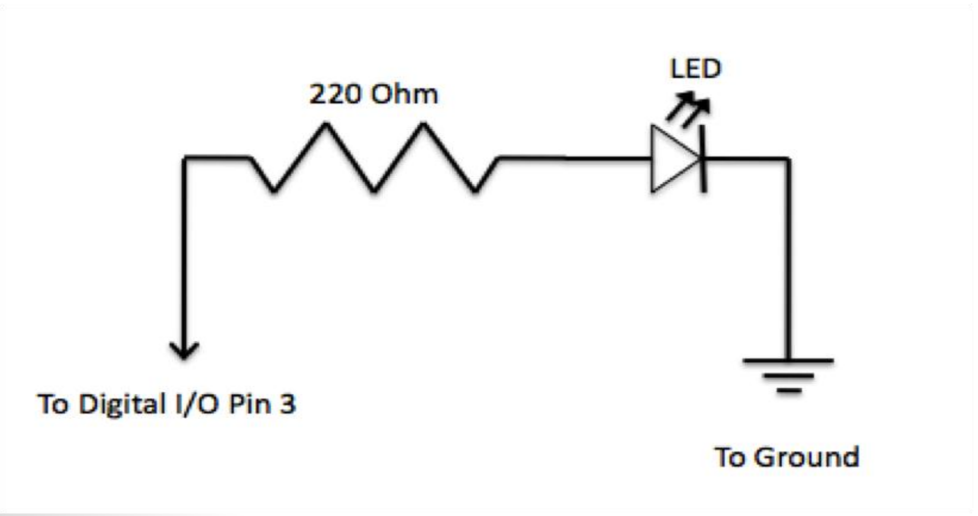
- begin()
- end()
- available()
- read()
- peek()
- flush()
- print()
- println()
- write()

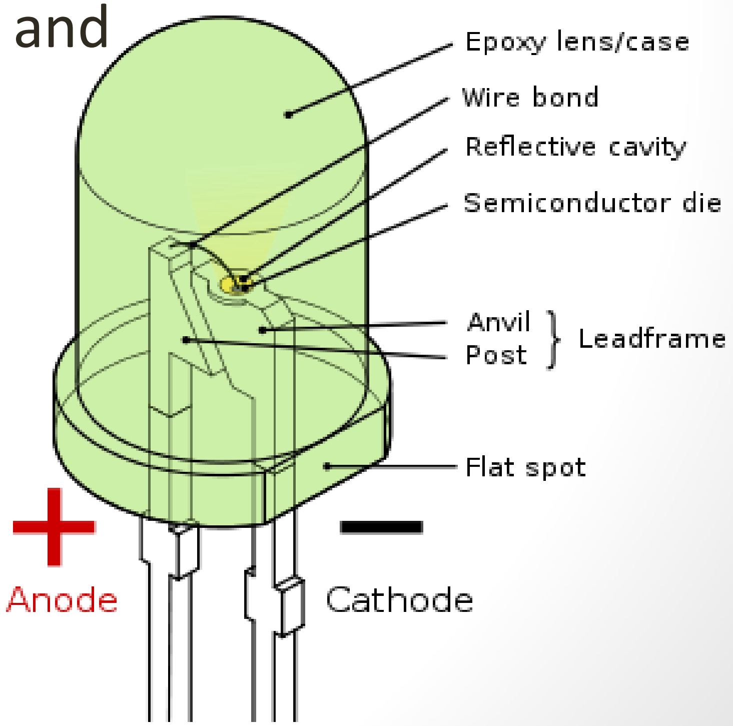
Bread Board



Activity-Blinking An LED

- LED->Forward Biased or Reverse Biased?
- Light Emitted by Radiative Recombination of Electrons and Holes.
- Normally rated as 20 -25 mA at around 2 V. Resistor in series is required to limit voltage and hence current through LED.





Kushagra Nigam-
kushagran1@gmail.com

Activity-Blinking An LED

- **Sketch:**

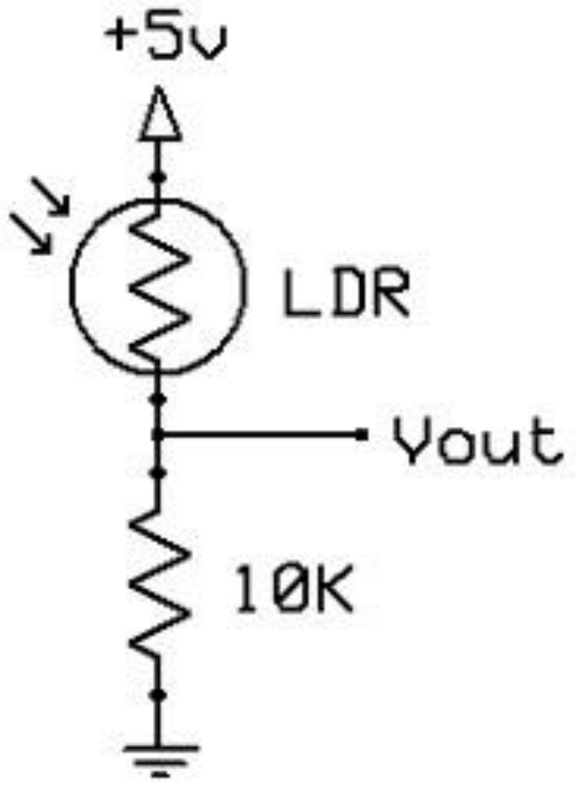
```
void setup()
{ // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              //WHY DELAY???
}
```

ADC (ANALOG TO DIGITAL CONVERSION)

- uC can't process analog signals.
- **Resolution** – No. of bits to show the values.
- Arduino has 10-bits resolution i.e. it has 2 registers whose 10 out of total 16 bits are used to represent the ADC values.
- Example-Temperature Sensor- 0 to 100 deg C ->0V to 5 V
- 1 Bit Resolution.
- 2 Bit Resolution.
- 10 Bit Resolution.

Kushagra Nigam-
kushagran1@gmail.com

Activity-Reading LDR and Potentiometer



->Vout- Analog Input Pin.

->LDRs or Light Dependent (Photo) Resistor are very useful especially in light/dark sensor circuits.

Kushagra Nigam-
kushagran1@gmail.com

Activity-Reading LDR and Potentiometer

- **Sketch:**

```
int sensorValue;
Int sensorPin=A0;
void setup()
{//Analog pins are input pins
}
void loop() {
 // read the value from the sensor:
 sensorValue = analogRead(sensorPin);
 //HOW TO DISPLAY??? Use Serial Comm.
}
```

Serial Communication

- Used for communication between the Arduino board and a computer or other devices.
- All Arduino boards have at least one serial port (also known as a UART or USART): Serial.
- It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB.
- Thus, if you use these functions, you cannot also use pins 0 and 1 for digital input or output.
- Baud Rate of communication: Symbols per second that are to be transmitted .
- 1 Symbol can be 1 bit or 2bit or n bits long depending upon the protocol used.
- In our case Baud Rate is bit rate.

Basic Functions

Serial Communication

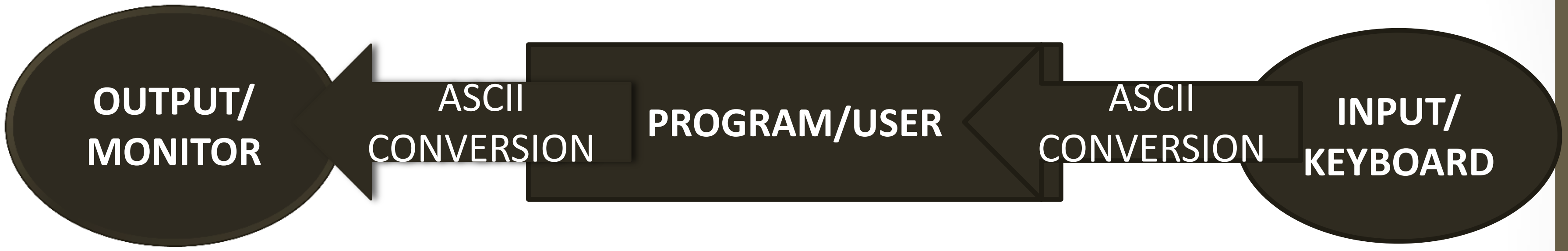
- Serial.begin(baudrate) //To start the communication
- Serial.end() //To end communication
- Serial.available() //To check if any serial data is available for microcontroller
- Serial.read() //To read incoming data into a variable
- Serial.flush() //Very important to empty the buffer
- Serial.print() //To print values strings characters
- Serial.println() //To print values strings characters in new line
- Serial.write() //Writes binary data to the serial port (series of bytes)

SOME BASICS

- **INTEGER CONSTANTS** int x=1;
- **CHARACTER CONSTANTS** char x=‘c’;
- **STRING LITERALS** char a[10]=‘HELLO’;
- **ASCII VALUES :** 0 to 9 -> 48 to 57

  a to z -> 97 to 122

  A to Z -> 65 to 90



Kushagra Nigam-
kushagran1@gmail.com

ACTIVITY ON SERIAL COMM.

- **CODE1:**

```
void setup()
{ Serial.begin(9600);
}
void loop()
{Serial.write(48);// 48 corresponds to 0 Alternative way use ‘0’
delay(50);
}
```

ACTIVITY ON SERIAL COMM.

Code2:

```
void setup() {
  Serial.begin(9600);
}
int x=0;
void loop()
{
while(x<10)
 {
  Serial.print(x);//USE Serial.println(x); to print in next line
  x++;
  delay(50);
 }
}
//print() function employs automatic conversion to ascii value
To give space use Serial.print("\t");
To give newline use Serial.print("\n");
```

ACTIVITY ON SERIAL COMM.

Code3:

```
void setup()
{Serial.begin(9600);
}
int x=0;
void loop()
{x=Serial.read();
if(x==49)//ASCII value  49 corresponds to 1
Serial.println("true");
}
//ALTERNATIVE?? -> Equivalent character constant
Declare x as char x;
and use if(x=='1');
```

ACTIVITY ON ADC

- CIRCUIT:- Connect the A0 pin of arduino to 3.3V, 5V and GND one by one and read the corresponding ADC values.

Sketch:

```
int sensorValue;
Int sensorPin=A0;
void setup()
{//Analog pins are input pins
 Serial.begin(9600);
//To initialize serial communication at 9600 Baud rate
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue, DEC);  //Serial.print(sensorValue)
}
```

Kushagra Nigam-
kushagran1@gmail.com

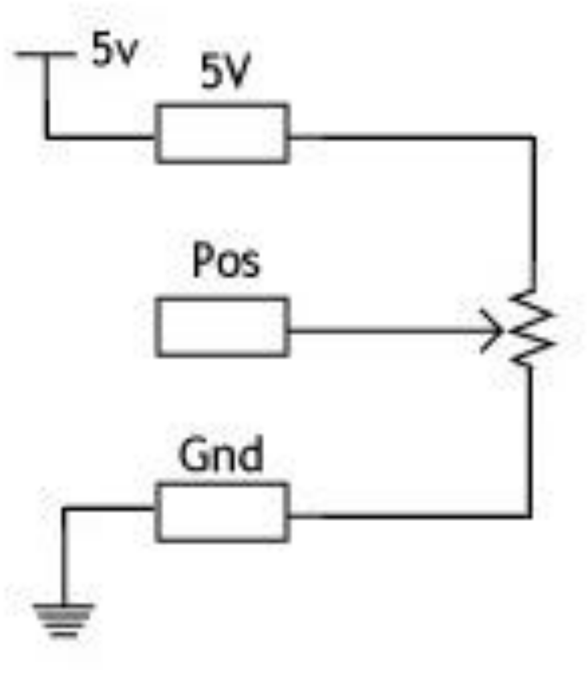
Activity-Reading LDR and Potentiometer

- **Sketch:**

```
int sensorValue;
Int sensorPin=A0;
void setup()
{//Analog pins are input pins
 Serial.begin(9600);
//To initialize serial communication at 9600 Baud rate
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue, DEC);  //Serial.print(sensorValue)
}
```

Kushagra Nigam-
kushagran1@gmail.com

Activity-Reading LDR and Potentiometer



PWM-Pulse Width Modulation

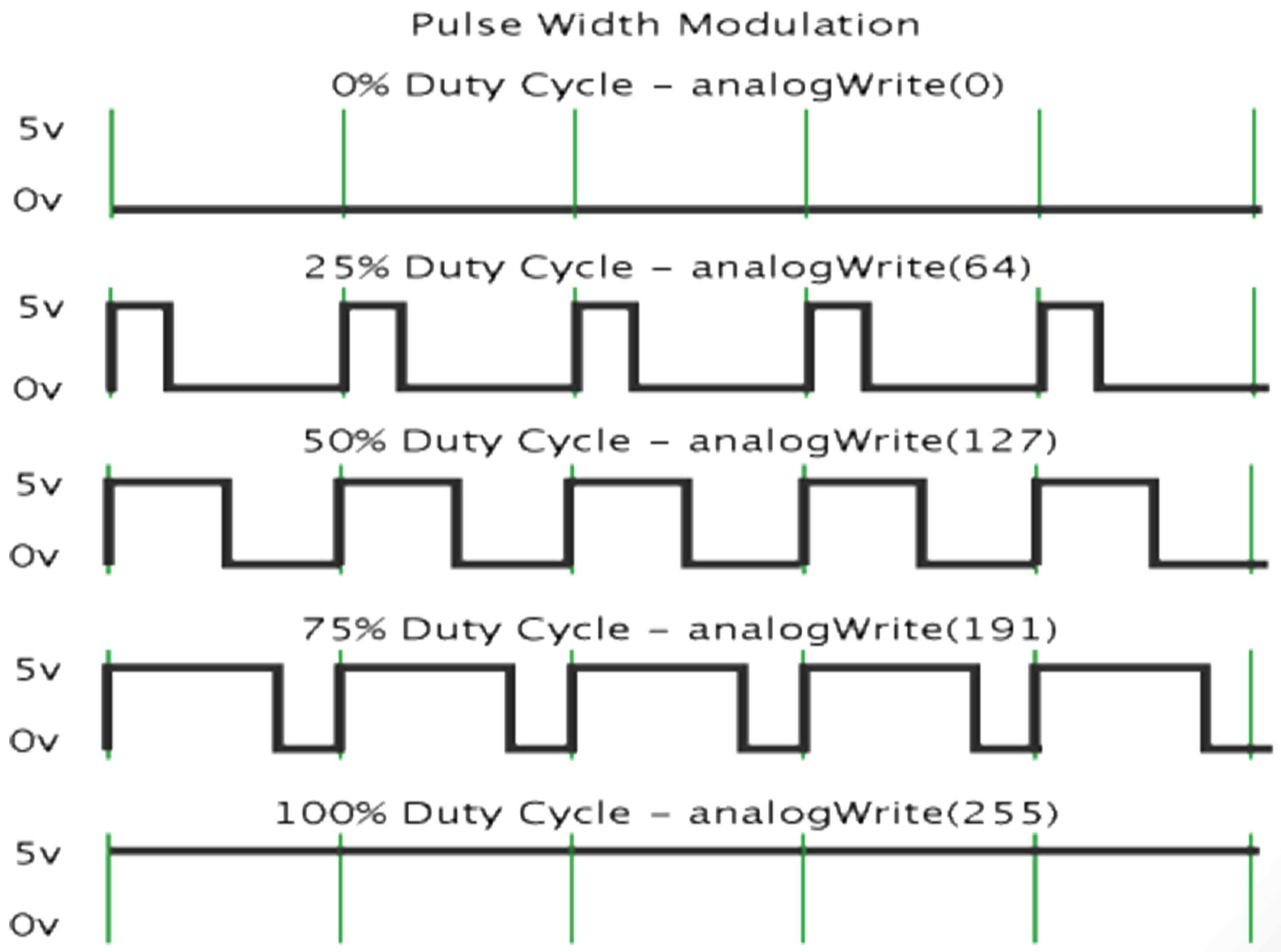
- **How to control brightness of LED???**

- Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off.

- Arduino has analogWrite() function that operates on digital pins labeled as PWM.

- **Duty Cycle:** % of total time for which signal is high.

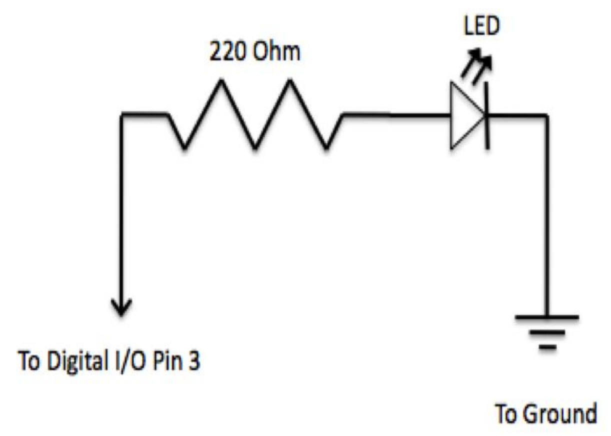
PWM-Pulse Width Modulation


Pulse Width Modulation
0% Duty Cycle – analogWrite(0)
5v
0v
25% Duty Cycle – analogWrite(64)
5v
0v
50% Duty Cycle – analogWrite(127)
5v
0v
75% Duty Cycle – analogWrite(191)
5v
0v
100% Duty Cycle – analogWrite(255)
5v
0v

Activity-Controlling the brightness of LED

Sketch:

```
int ledPin = 9;    // LED connected to digital pwm pin 9

void setup()  {
  // nothing happens in setup
}

void loop()  {
  // fade in from min to max in increments of 5 points:
  for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {
  // sets the value (range from 0 to 255):
   analogWrite(ledPin, fadeValue);
  // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }

  // fade out from max to min in increments of 5 points:
  for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {
  // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
  // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}
```



Kushagra Nigam-
kushagran1@gmail.com

Activity-Controlling the brightness of LED using potentiometer

- **Sketch:**

```
int ledPin = 9;
int sensorValue;
Int sensorPin=A0;
void setup()
{//Analog pins are input pins
 Serial.begin(9600);
//To initialize serial communication at 9600 Baud rate
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue, DEC);  //Serial.print(sensorValue)
  int val = map(sensorValue, 0, 1023, 0, 255);
  analogWrite(9, val);
}
```

EEPROM

- Memory whose values are kept when the board is turned off (like a tiny hard drive). This library enables you to read and write those bytes.
- **ATMEGA 168** - 512 B EEPROM

Functions:-

EEPROM.read(address);//Reads a byte from the EEPROM.

Parameters

address: the location to read from, starting from 0 (*int*)

Returns

the value stored in that location (*byte*)

Kushagra Nigam-
kushagran1@gmail.com

EEPROM

- **EEPROM.write(address, value)//WRITES A BYTE TO EEPROM**

Parameters

address: the location to write to, starting from 0 (*int*)

value: the value to write, from 0 to 255 (*byte*)

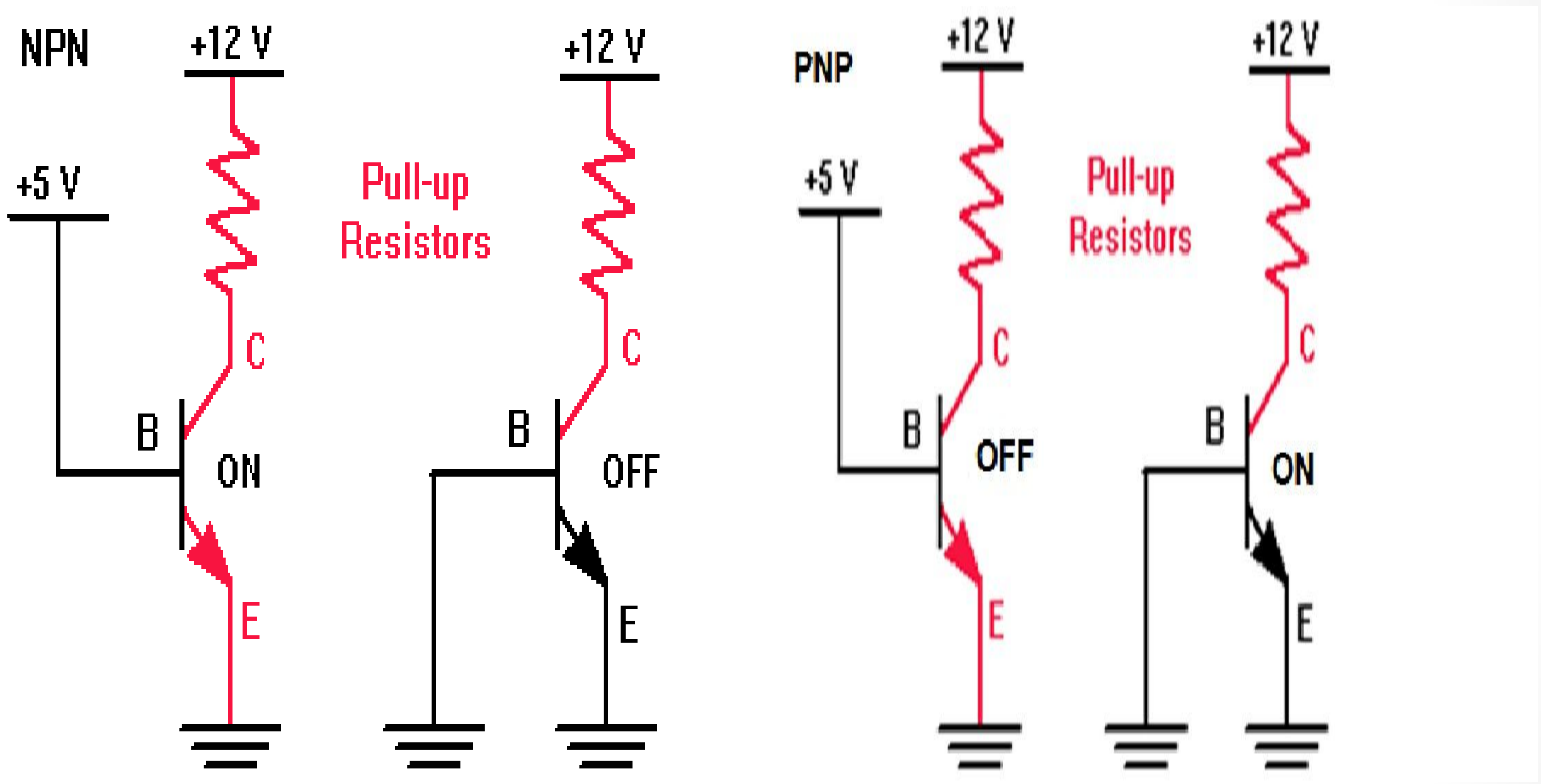
ACTIVITY EEPROM

```
#include <EEPROM.h>
 void setup()
{
for (int i = 0; i < 512; i++)
EEPROM.write(i, i);
}
 void loop()
{ }
//Code written in setup function  for single time execution
//void loop though is still required for error free compilation
```

Kushagra Nigam-
kushagran1@gmail.com

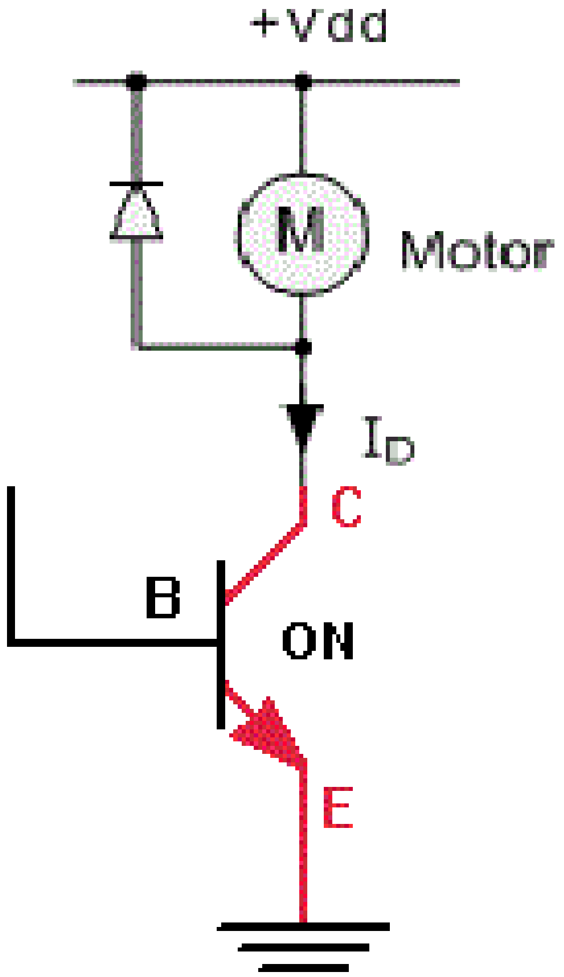
```
#include <EEPROM.h>

 int a = 0; int value;

void setup()
 { Serial.begin(9600);
 }

void loop()
 { value = EEPROM.read(a);

Serial.print(a);

Serial.print("\t");

Serial.print(value);

Serial.println();

a = a + 1;
if (a == 512)
 a = 0;
delay(500);
 }
```

NPN and PNP diode as switch



Kushagra Nigam-
kushagran1@gmail.com

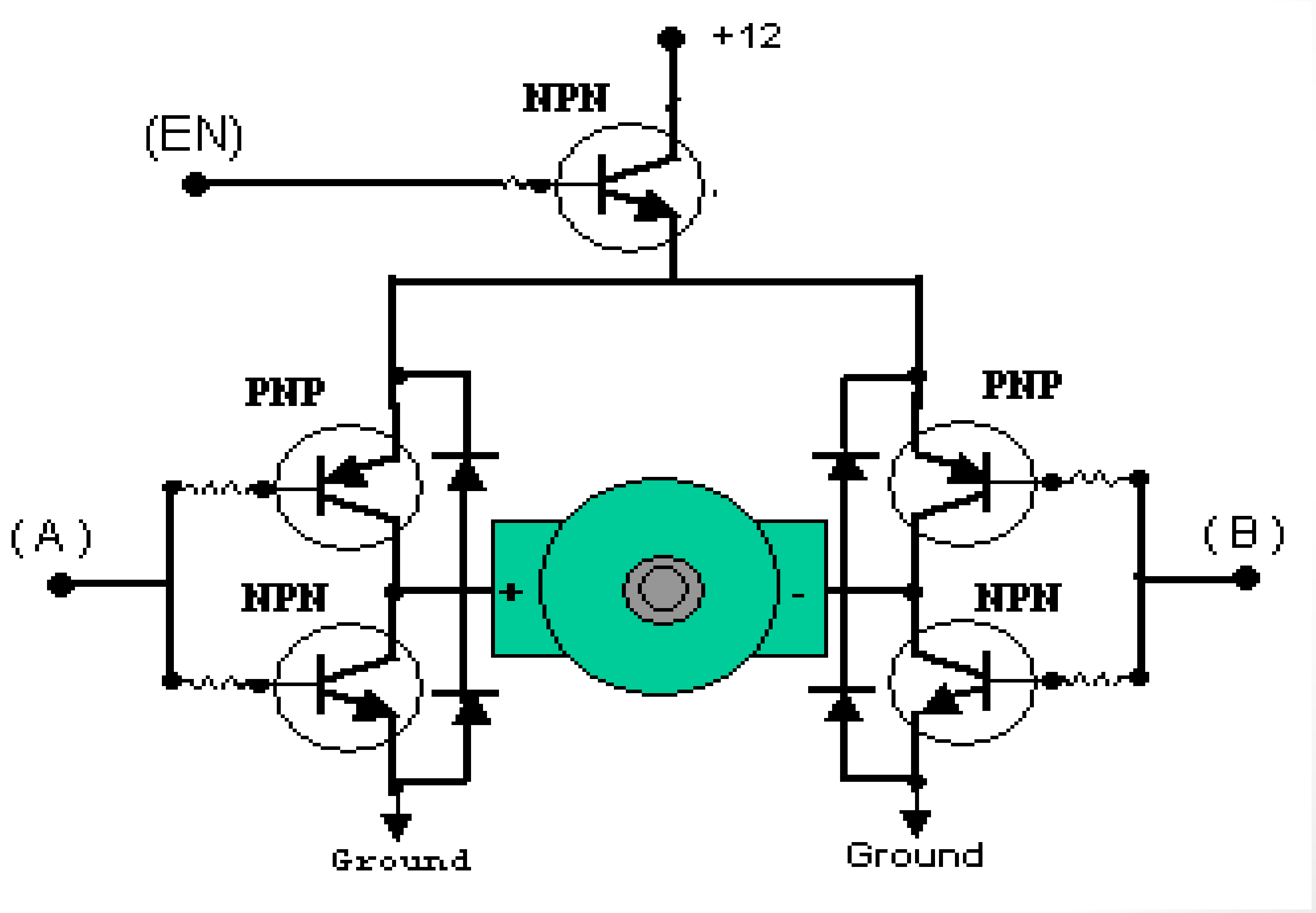
Motor Driver Working



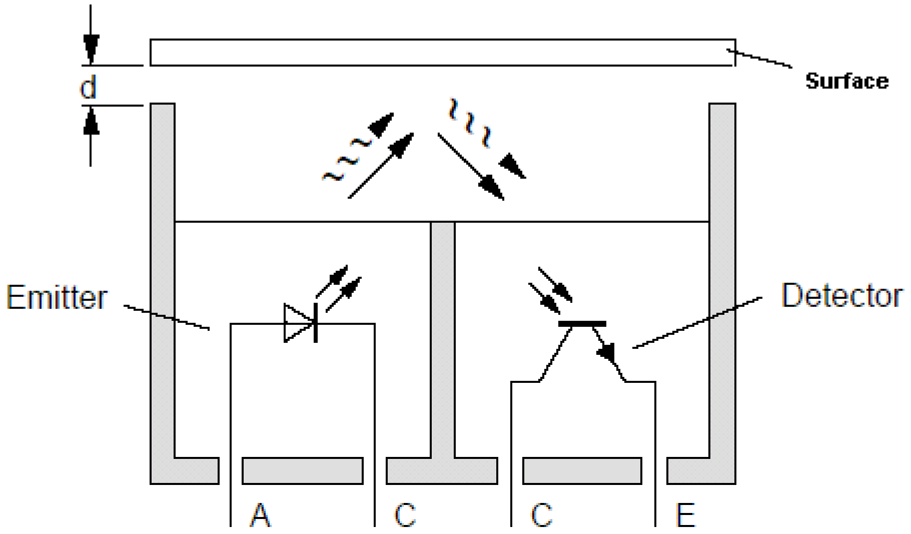
Kushagra Nigam- kushagran1@gmail.com

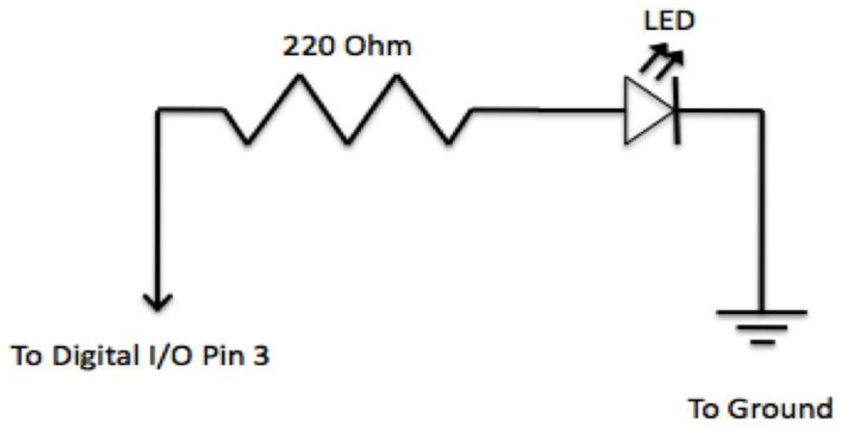
L293D

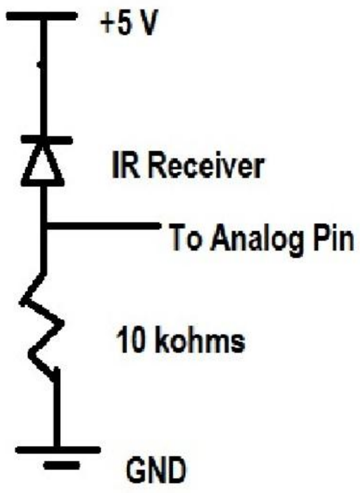
- It is a Dual H-Bridge Motor driver IC



IR SENSOR

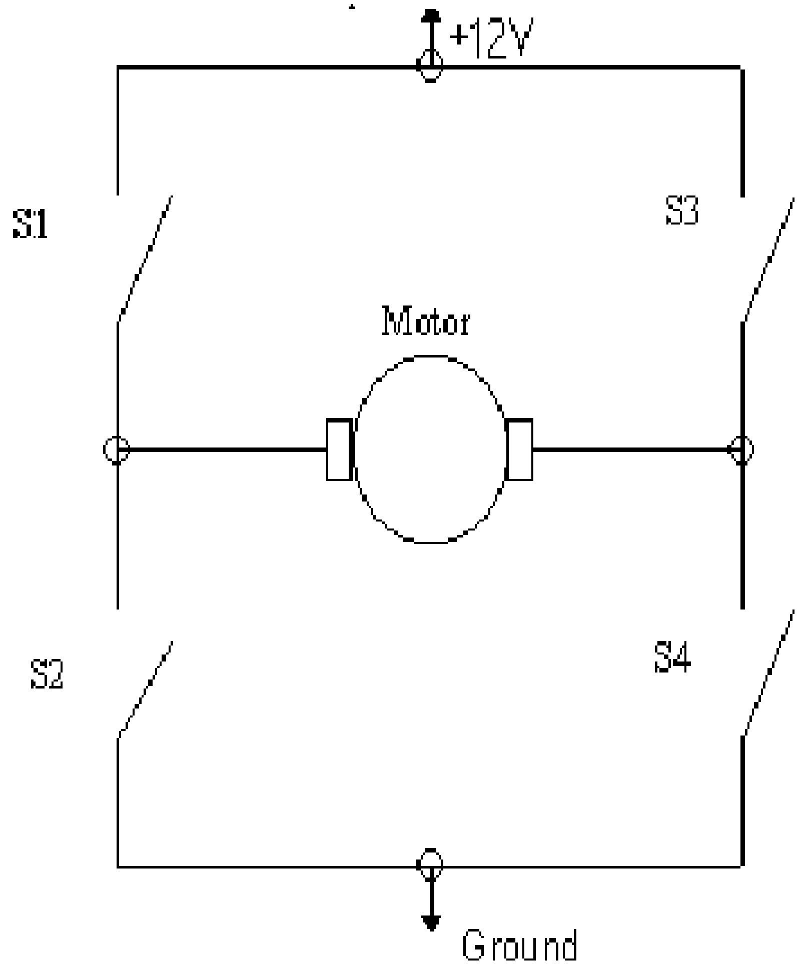






Kushagra Nigam-
kushagran1@gmail.com

H Bridge Circuit

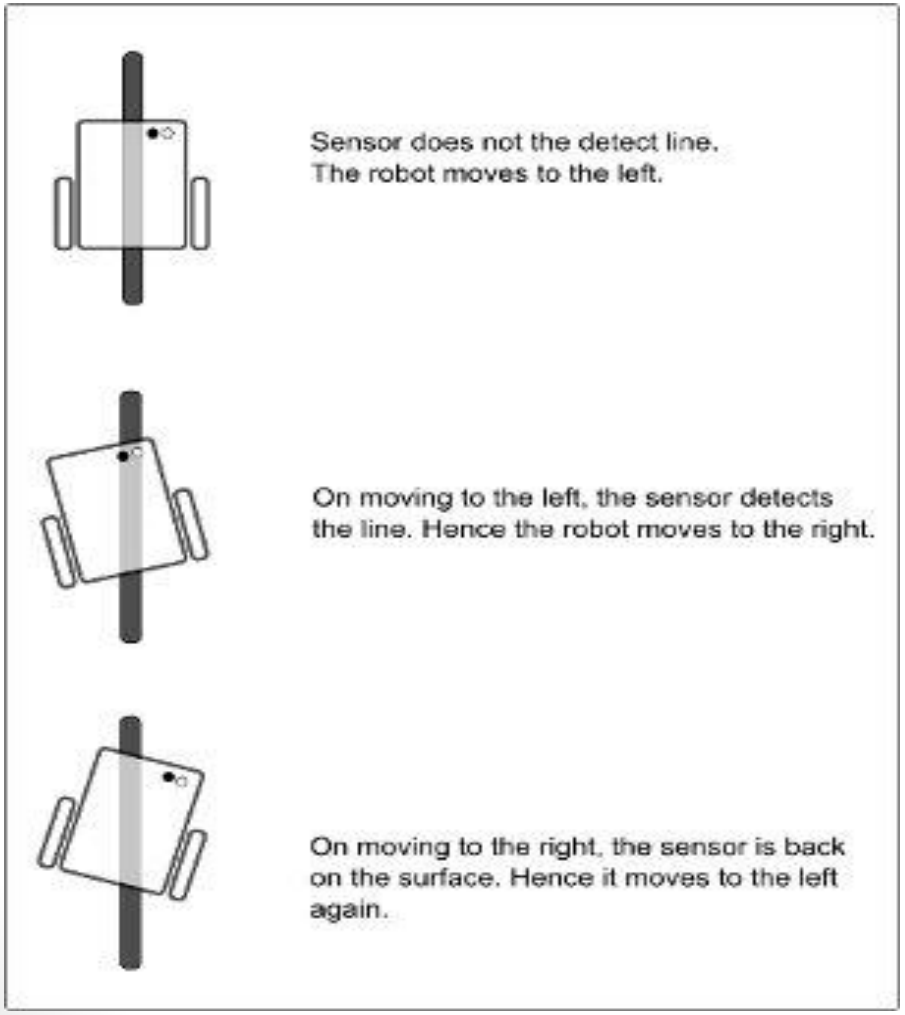


H-Bridge Switch

Switches Closed	Motor
S1 S3	Off
S2 S4	Off
S1 S4	Forward
S3 S2	Reverse

LINFOLLOWING BASICS

->Linfollowing using one IR transreceiver :



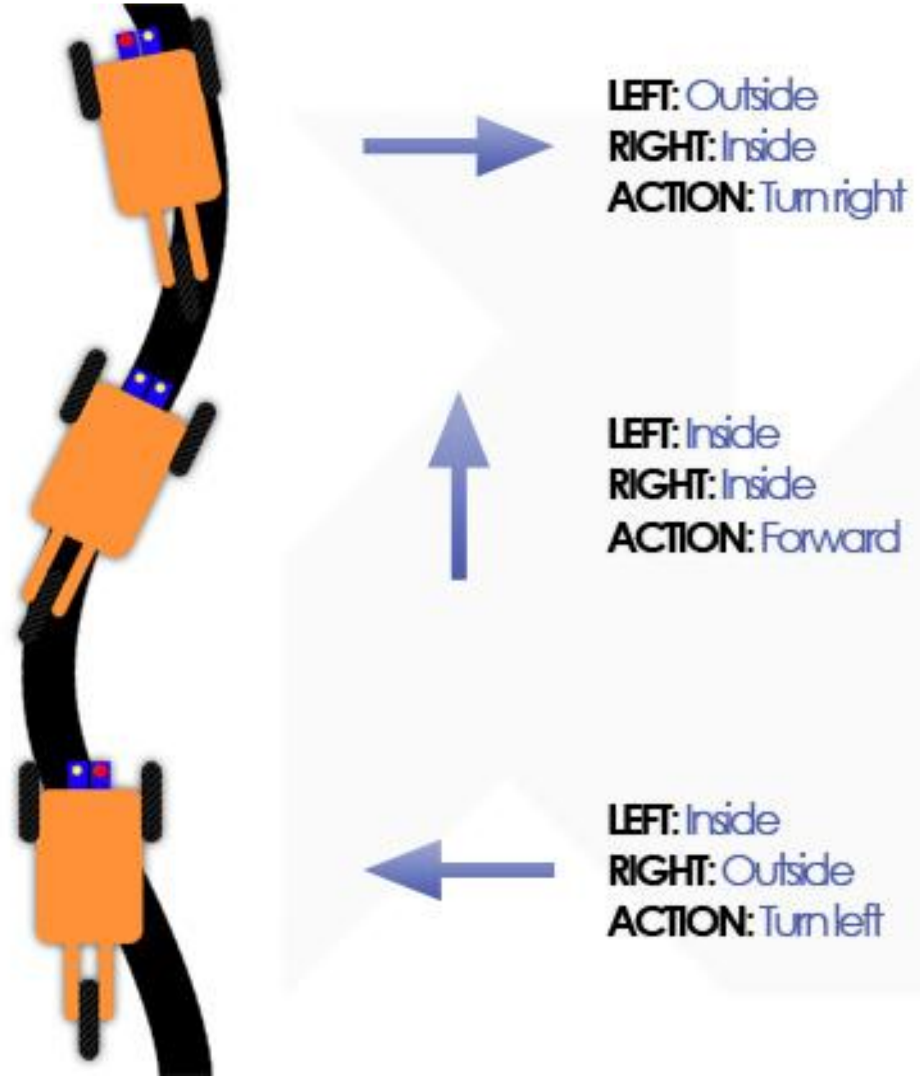
Sensor does not the detect line.
The robot moves to the left.

On moving to the left, the sensor detects the line. Hence the robot moves to the right.

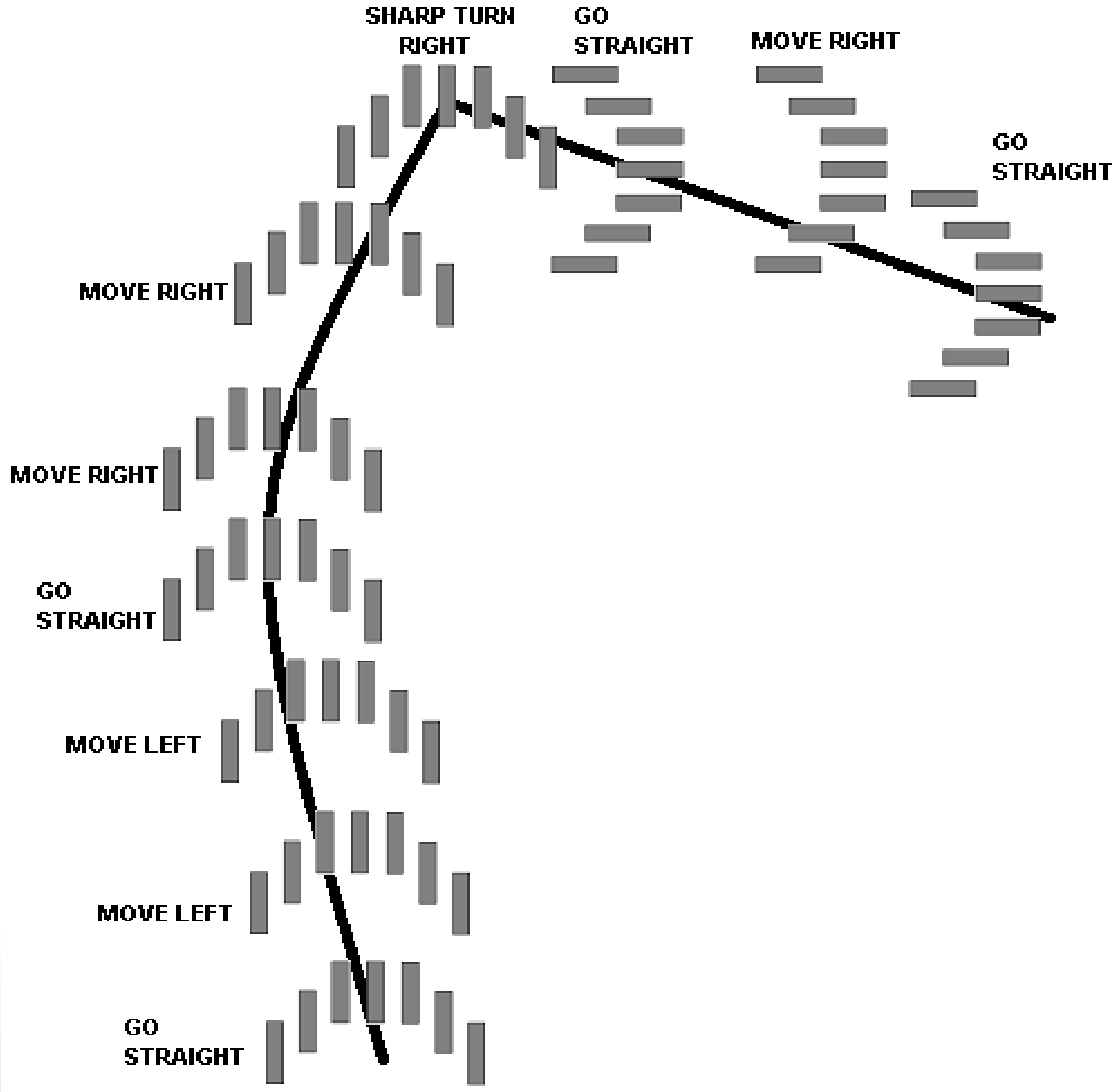
On moving to the right, the sensor is back on the surface. Hence it moves to the left again.

LINFOLLOWING BASICS

- ->Linfollowing using two IR transreceivers :



Linfollowing Sensor Array



Kushagra Nigam-
kushagran1@gmail.com

Activity-LineFollowing

ADD ON COMPONENTS

- PCB-Printed Circuit Board
- Bread Board-For testing of circuits
- Soldering Iron and solder wire
- MultiCutter
- Multimeter
- Arduino Shield
- LEDs
- Resistors (10 ohms, 1 ohms)
- Switches (with or without locking)
- Other Sensors depending on your future projects.

Sites listed on

http://robotronics.yolasite.com
Presentation and Tutorials tab

Kushagra Nigam-
kushagran1@gmail.com

Wide Range of Sensors

- PIR (Pyro electric Infrared Sensor):-Use to detect living object presence.
- Ultrasonic Range Detector.
- Accelerometer Sensor Module.
- LM35D temperature Sensor IC.

Kushagra Nigam- kushagran1@gmail.com

THANK YOU

Any Queries:

Contact :- Jinal Shah (jinal4141@gmail.com)

Kushagra Nigam (kushagran1@gmail.com)

Vishal Pathak (vishalpathak24@gmail.com)

Slide Available on http://robotronics.yolasite.com

Presentations and tutorials tab