EMBEDDED SYSTEMS TUTORIALS - I

(BASICS OF ARDUINO)

Prepared by,

SHASHANK KARN

TABLE OF CONTENTS

Arduino Programming Basics

- Arduino Programming is based on basic concepts of C/C++ and has a structure of C#.
- It requires initial setup function known as **void setup () {}** and then a continuously executing function known as **void loop () {}**.
- Void setup () function is used to set the parameters such as mode of the pin i.e., pin being used as either output or input. Void loop () function is used to execute a set of instruction continuously.
- Here is the instruction set used commonly

Command	Description
`pinMode(n,INPUT)`	Set pin *n* to act as an input. One-time command at top of program.
`pinMode(n,OUTPUT)`	Set pin *n* to act as an output
`digitalWrite(n,HIGH)`	Set pin *n* to 5V
`digitalWrite(n,LOW)`	Set pin *n* to 0V
`delay(x)`	Pause program for *x* millisec, *x* = 0 to 65,535
`tone(n,f,d)`	Play tone of frequency *f* Hz for *d* millisec on speaker attached to pin *n*
`for()`	Loop. Example: for (i=0;i<3;i++){} Do the instructions enclosed by {} three times
`if (expr) {}`	Conditional branch. If *expr* true, do instructions enclosed by {}
`while (expr) {}`	While *expr* is true, repeat instructions in {} indefinitely

Following are some basic programs.

- Printing Hello world on Serial monitor.

```
void setup()
{
  Serial.begin(9600);
  Serial.println("Hello World");
}
void loop() {}
```

- Printing the reading of a digital pin

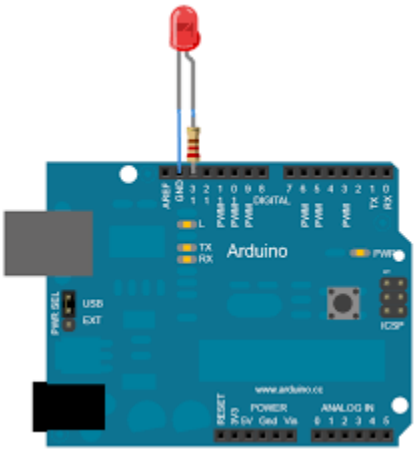
```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.println(digitalRead(2));
  delay(100);
}
```

- Program to demonstrate If-condition

```
int i;

void setup()
{
  Serial.begin(9600);
  i=0;
}

void loop()
{
  i=i+1;
  if (i<30) {
      Serial.println(i);
  }
}
```

Arduino with LED

- Interfacing Diagram.



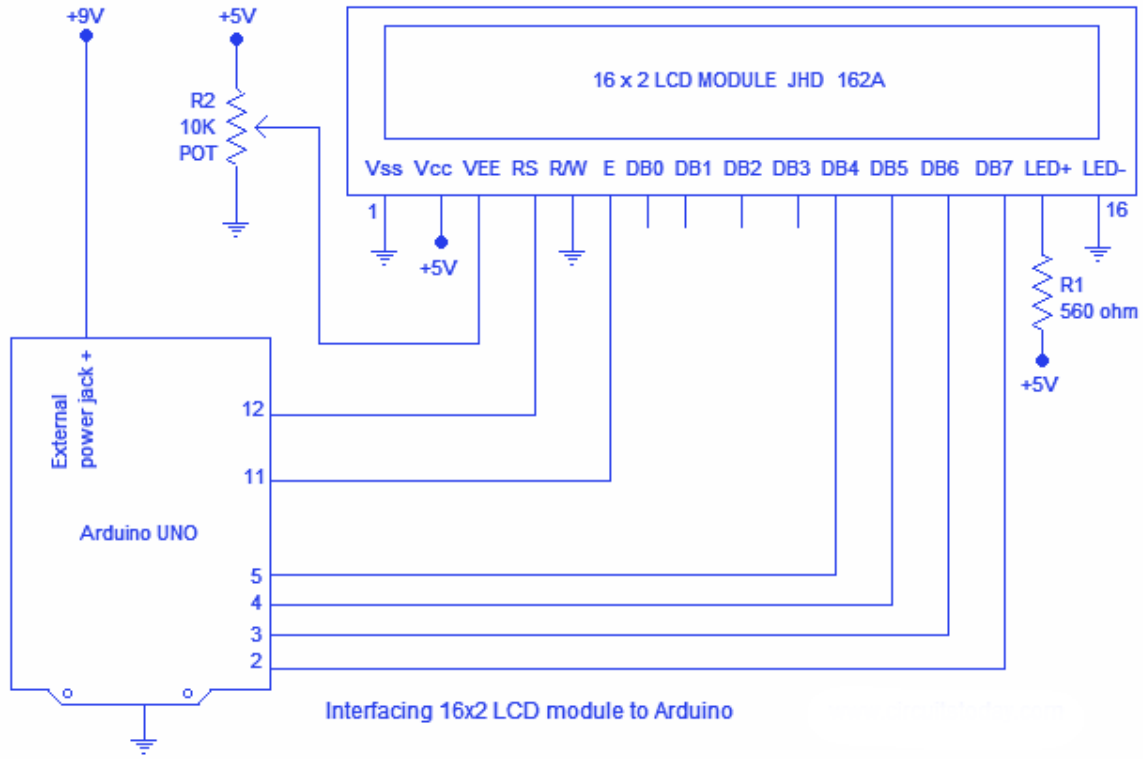
- Program / code.

```
Int led = 13;
void setup(){
  pinMode(led,OUTPUT);
}

Void loop(){

  digitalWrite(led,HIGH);

  delay(1000);

  digitalWrite(led,LOW);

  delay(1000);

}
```

- The above program will turn LED on and off with a delay of 1sec.

Arduino with LCD 16*2 Screen

- Interfacing diagram

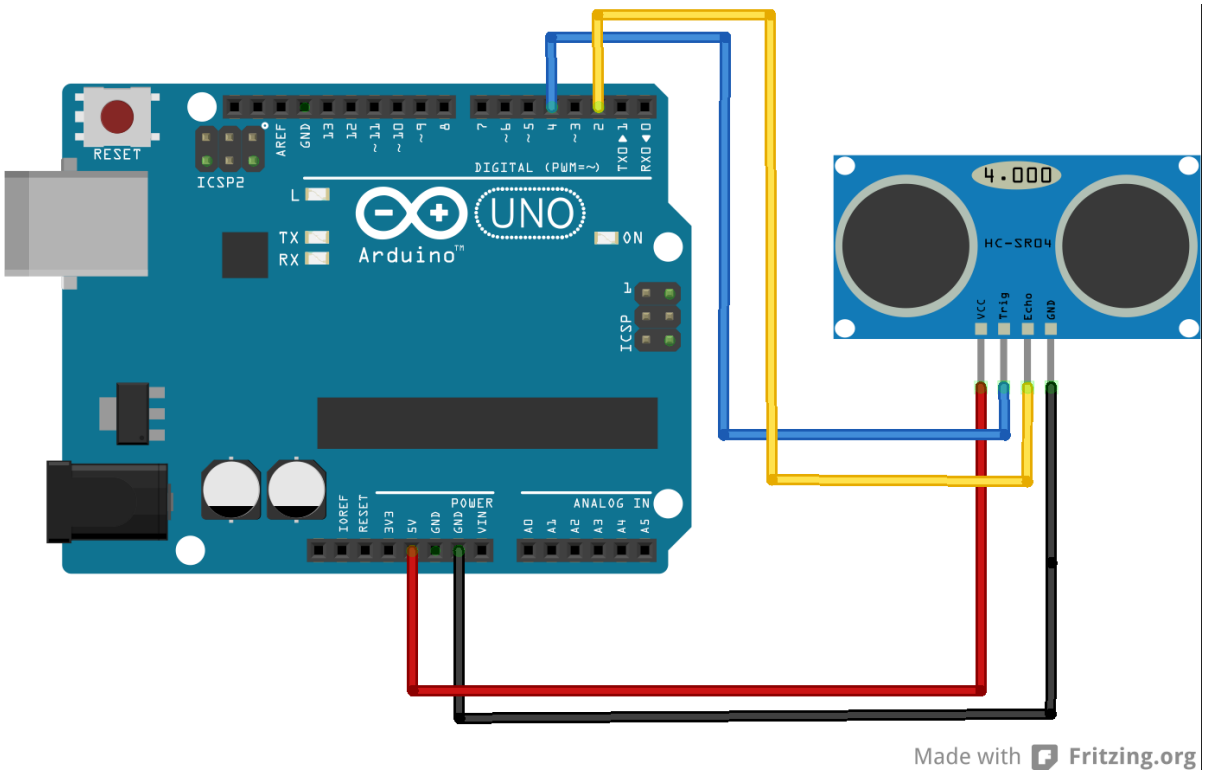


- Program / code

```
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}
void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

Arduino with Ultrasonic sound sensor

- Interface diagram.



- Program / code.
- Using header file.

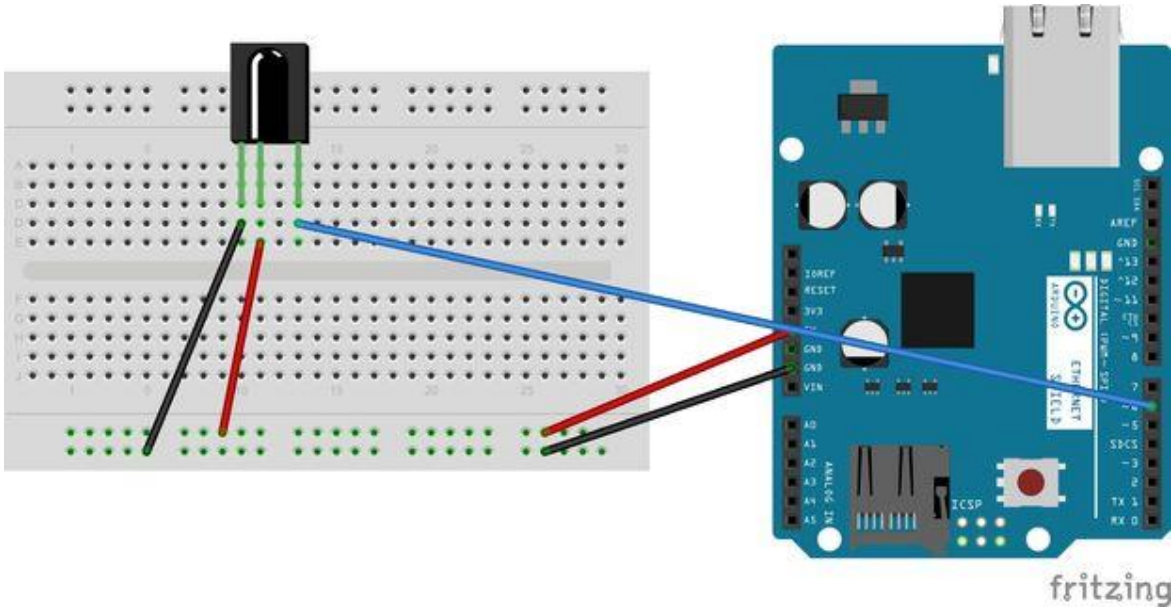
```
#include <Ultrasonic.h>

Ultrasonic Ultrasonic(2, 4);

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

void loop() {
 Serial.println("Distance in CM : ");
 // Pass INC as a parameter to get the distance in inches
 Serial.println(Ultrasonic.distanceRead());
 delay(1000);
}
```

- Without using header.

```
// defines pins numbers
const int trigPin = 2;
const int echoPin = 4;
// defines variables
long duration;
int distance;
void setup() {
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.begin(9600); // Starts the serial communication
}
void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in
microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
delay(800);
}
```

Arduino with IR-decoder

- Interfacing diagram.

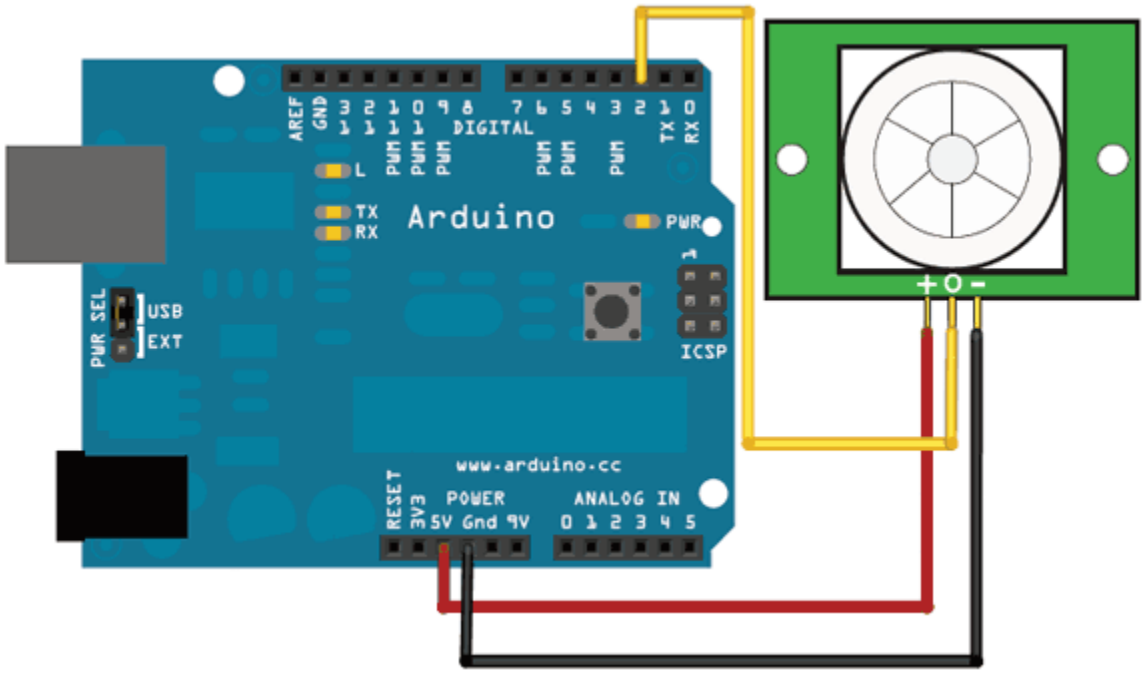


- Programm / code.

```
#include <IRremote.h>
int RECV_PIN = 6;
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup(){
 Serial.begin(9600);
 irrecv.enableIRIn(); // Start the receiver
}
void loop() {
 if (irrecv.decode(&results)) {
  Serial.println(results.value, HEX);
  irrecv.resume(); // Receive the next value
 }
 delay(100);
}
```

Arduino with PIR-motion sensor

- Interfacing diagram.



- Program / code.

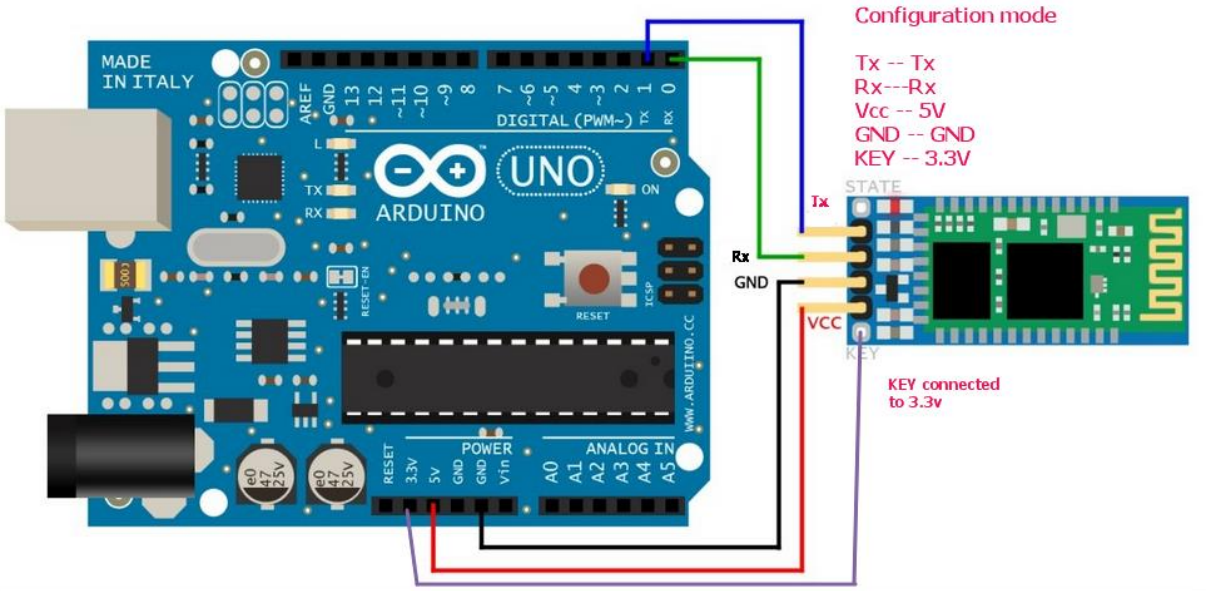
```
int calibrationTime = 30;

//the time when the sensor outputs a low impulse
long unsigned int lowIn;

//the amount of milliseconds the sensor has to be low
//before we assume all motion has stopped
long unsigned int pause = 100;

boolean lockLow = true;
boolean takeLowTime;

int pirPin = 3;    //the digital pin connected to the PIR sensor's output
int ledPin = 12;


/////////////////////////////
```

```
//SETUP
void setup(){
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(pirPin, LOW);

  //give the sensor some time to calibrate
  Serial.print("calibrating sensor ");
    for(int i = 0; i < calibrationTime; i++){
      Serial.print(".");
      delay(1000);
      }
    Serial.println(" done");
    Serial.println("SENSOR ACTIVE");
    delay(50);
  }

////////////////////////////
//LOOP
void loop(){

     if(digitalRead(pirPin) == HIGH){
       digitalWrite(ledPin, HIGH);   //the led visualizes the sensors
output pin state
       delay(100);
       digitalWrite(ledPin,LOW);
       delay(100);
       if(lockLow){
         //makes sure we wait for a transition to LOW before any further
output is made:
         lockLow = false;
         Serial.println("---");
         Serial.print("motion detected at ");
         Serial.print(millis()/1000);
```

```
        Serial.println(" sec");
        delay(50);
        }
        takeLowTime = true;
      }

    if(digitalRead(pirPin) == LOW){
      digitalWrite(ledPin, LOW);  //the led visualizes the sensors output
pin state

      if(takeLowTime){
       lowIn = millis();          //save the time of the transition from high
to LOW
       takeLowTime = false;       //make sure this is only done at the
start of a LOW phase
       }
      //if the sensor is low for more than the given pause,
      //we assume that no more motion is going to happen
      if(!lockLow && millis() - lowIn > pause){
          //makes sure this block of code is only executed again after
          //a new motion sequence has been detected
          lockLow = true;
          Serial.print("motion ended at ");      //output
          Serial.print((millis() - pause)/1000);
          Serial.println(" sec");
          delay(50);
          }
      }
  }
```

Arduino with HC-05/06 blue-tooth module

- Interfacing diagram.



- Program / code.

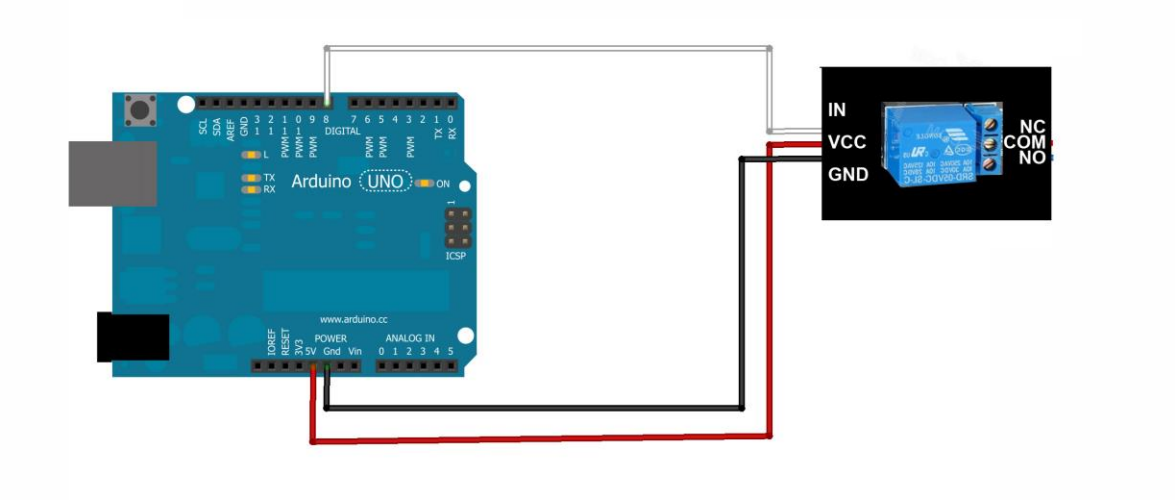
  There is no specific program for Bluetooth module as it just reads serial values received from a connected device.

  Here is a program to read values from Bluetooth module:

```
char state;
void setup(){
Serial.begin(9600);
}
void loop(){
Serial.available();
state = Serial.read();
Serial.println(state);
}
```

Arduino with 5V Relay

- Interfacing diagram.



- Program / code.

  Code for relay is similar to that for LED, only difference is that relay switches for digital logic '0' or say 'LOW'.

```
Int relay = 3;
void setup(){
  pinMode(relay,OUTPUT);
}
Void loop(){
  digitalWrite(relay,LOW);
  delay(1000);
  digitalWrite(relay,HIGH);
  delay(1000);
}
```

Mini Project – Distance Sensor

- As for now we know how to interface ultrasonic sensor and LCD with the Arduino, we can now make a simple project which can be used to measure distance.
- Interfacing diagram can be derived from the above interfacing diagrams.
- Program / code.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
// defines variables
long duration;
int distance;
void setup() {
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
lcd.begin(16,2); // Starts the serial communication
pinMode(8,OUTPUT);
pinMode(7,OUTPUT);
}
void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in
microseconds
```

```
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
lcd.setCursor(1,0);
lcd.print("Distance:");
lcd.print(distance);
lcd.print("cm..");
if (distance < 14)
{
 digitalWrite(8,HIGH);
 lcd.setCursor(0,1);
 lcd.println("too close.....");
}
else if(distance > 15 && distance < 40 )
{
 digitalWrite(7,HIGH);
 digitalWrite(8,LOW);
 lcd.setCursor(0,1);
 lcd.println("enough distance.....");
}
else
{
 digitalWrite(7,LOW);
 digitalWrite(8,LOW);
 lcd.setCursor(0,1);
 lcd.println(".............");
}
delay(100);
}
```

Mini Project – Bluetooth automation

- This project involves the basic automation using a simple android app and Bluetooth module.
- Program / code.

```
#define R1 3
#define R2 4
#define R3 5
#define R4 6
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
pinMode(R1,OUTPUT);
pinMode(R2,OUTPUT);
pinMode(R3,OUTPUT);
pinMode(R4,OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
 Serial.available();       //checking the availability of Arduino board for Serial comuniication.
 char state = Serial.read();//Declaring a variable for recieving the input from the smartphone.
 int flag=0;               //Seting flag value to zero.

  if(state == 'A'){
   //Put code here for performing Function 1
   digitalWrite(R1,0);
  flag=1;
  }
   if(state == 'a'){
  //Put code here for stopping Function 1
    digitalWrite(R1,1);
    flag=1;
  }
```

```
 if(state == 'B'){
  //Put code here for performing Function 2
   digitalWrite(R2,0);
    flag=1;
 }
  if(state == 'b'){
 //Put code here for stopping Function 2
   digitalWrite(R2,1);
   flag=1;
 }
 if(state == 'C'){
 //Put code here for performing Function 3
   digitalWrite(R3,0);
   flag=1;
 }
  if(state == 'c'){
 //Put code here for stopping Function 3
  digitalWrite(R3,1);
   flag=1;
  }
 if(state == 'D'){
 //Put code here for performing Function 4
   digitalWrite(R4,0);
   flag=1;
 }
  if(state == 'd'){
 //Put code here for stopping Function 4
 digitalWrite(R4,1);
  flag=1;
 }
}
```

Mini Project – IR automation

- This project involves use of sm0038 IR decoder as the interface for automation.
- Program / code.

```
#include <IRremote.h>

int RECV_PIN = 11;

int ena=5;//the enable pins should be connected to a pwm pin on arduino.
int enb=10;
int ln1=9;// the line pins should be connected to non-pwm pins on arduino.
int ln2=8;
int ln3=7;
int ln4=6;

IRrecv irrecv(RECV_PIN);
int val=0;
decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
pinMode(ena,OUTPUT);
pinMode(enb,OUTPUT);
pinMode(ln1,OUTPUT);
pinMode(ln2,OUTPUT);
pinMode(ln3,OUTPUT);
pinMode(ln4,OUTPUT);
}
```

```
void loop() {
  if (irrecv.decode(&results))
  {
  // Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
    val=results.value;
    Serial.println(results.value);
  }

     int flag=0;




if (val == 16) {

    analogWrite(ena,255);
     analogWrite(enb,255);
     digitalWrite(ln1,HIGH);
     digitalWrite(ln2,LOW);
     digitalWrite(ln3,HIGH);
     digitalWrite(ln4,LOW);

     flag = 1;
}
   if (val == 2320) {

    analogWrite(ena,0);
     analogWrite(enb,0);
     digitalWrite(ln1,LOW);
     digitalWrite(ln2,LOW);
     digitalWrite(ln3,LOW);
     digitalWrite(ln4,LOW);

     flag = 0;
```

```
}
 if (val == 2064) {

   analogWrite(ena,255);
    analogWrite(enb,255);
    digitalWrite(ln1,LOW);
    digitalWrite(ln2,HIGH);
    digitalWrite(ln3,LOW);
    digitalWrite(ln4,HIGH);

    flag = 1;
}
if (val == 3088) {

   analogWrite(ena,255);
    analogWrite(enb,255);
    digitalWrite(ln1,HIGH);
    digitalWrite(ln2,LOW);
    digitalWrite(ln3,LOW);
    digitalWrite(ln4,LOW);

    flag = 1;
}
 if (val == 528) {

   analogWrite(ena,255);
    analogWrite(enb,255);
    digitalWrite(ln1,LOW);
    digitalWrite(ln2,LOW);
    digitalWrite(ln3,LOW);
    digitalWrite(ln4,HIGH);

    flag = 1;
}
}
```