Interfacing of RGB LED with Arduino and display all possible colours.

```
          int redPin= 9;
int greenPin = 6;  int
bluePin = 5; void setup()
          {
          pinMode(redPin, OUTPUT); pinMode(greenPin,
          OUTPUT); pinMode(bluePin, OUTPUT);
          }
          void loop()
          {    int
          i,j,k;
           for(i=0;i<255;i++)
           {
             analogWrite(redPin, i);
          analogWrite(greenPin, 128);
          analogWrite(bluePin, 128);     delay(1000);
           }
           for(j=0;j<255;j++)
           {
             analogWrite(greenPin, j);
          analogWrite(bluePin, 128);
          analogWrite(redPin, 128);     delay(1000);
           }
           for(k=0;k<255;k++)
           {
             analogWrite(bluePin, k);
          analogWrite(redPin, 128);
          analogWrite(greenPin, 128);     delay(1000);
           }
          }
```

Interfacing of LDR with Arduino and program for displaying the light Intensity

```
int ldrPin = A0; // Analog input pin for LDR int ldrValue; // Variable to store LDR value  void setup()
{
pinMode(ldrPin,INPUT);
Serial.begin(9600); // Initialize serial communication for debugging
}
void loop()
{
  int readValue;
float realValue;
  readValue = analogRead(ldrPin);
realValue = (5.0/1024.0)*readValue;
  Serial.println(realValue);
}
```

Interfacing of DC motor with Arduino and speed control using PWM.

Clockwise and anti clockwise

```
// Define motor control pins
int motorPin1 = 9;  // Motor input 1
int motorPin2 = 10; // Motor input 2

void setup() {
  // Set motor control pins as OUTPUT
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
}

void loop() {
  // Rotate clockwise
  digitalWrite(motorPin1, HIGH);
  digitalWrite(motorPin2, LOW);

  // Set PWM speed (0 to 255)
  analogWrite(motorPin2, 150);  // Adjust speed as needed

  // Delay for 2 seconds
  delay(2000);

  // Rotate anticlockwise
  digitalWrite(motorPin1, LOW);
  digitalWrite(motorPin2, HIGH);

  // Set PWM speed (0 to 255)
  analogWrite(motorPin2, 100);  // Adjust speed as needed

  // Delay for 2 seconds
```

```
  delay(2000);

  // Stop the motor
  digitalWrite(motorPin1, LOW);
  digitalWrite(motorPin2, LOW);

  // Delay for 2 seconds before repeating the loop
  delay(2000);
}
```

LCD name and roll no.

```
#include <LiquidCrystal.h>

// Initialize the LCD with the appropriate pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // Set up the LCD columns and rows
  lcd.begin(16, 2);
}

void loop() {
  // Clear the LCD screen
  lcd.clear();

  // Display your name on the first line
  lcd.setCursor(0, 0);
  lcd.print("Your Name");

  // Display your roll number on the second line
  lcd.setCursor(0, 1);
  lcd.print("Roll Number");

  // Delay for a while before clearing the screen again
  delay(2000);
}
```

Lm35

```
#define sensorPin A0

void setup() {

Serial.begin(9600);

}

void loop() {                          // Get a reading from the temperature sensor:

int reading = analogRead(sensorPin);          //Convert digital data into analog by multiplying by 5000 and dividing by 1024

float voltage = reading * (5000 / 1024.0);    // Convert the voltage into the temperature in degree Celsius: float  float

temperatureC = voltage / 10;

float temperatureF=(temperatureC*1.8)+32;          // Converting to Fahrenheit// Print the temperature in Celsius into the Serial Monitor:

Serial.print("Temperature in Celsius = ");

Serial.print(temperatureC);

Serial.println("C");                    // Print the temperature in Celsius into the Serial Monitor:

Serial.print("Temperature in Fahrenheit = ");

Serial.print(temperatureF);

Serial.println("F"); Serial.print("\n"); delay(1000);

// wait a second between readings

}
```

Bluetooth

```
#include <SoftwareSerial.h>

SoftwareSerial bluetooth(10, 11); // RX, TX

int ledPin1 = 2;
int ledPin2 = 3;
int ledPin3 = 4;
int ledPin4 = 5;

void setup() {
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);

  // Serial communication with Bluetooth module
  Serial.begin(9600);
  bluetooth.begin(9600);
}

void loop() {
  if (bluetooth.available() > 0) {
    char command = bluetooth.read();

    // Turn on all LEDs
    if (command == '1') {
      digitalWrite(ledPin1, HIGH);
      digitalWrite(ledPin2, HIGH);
      digitalWrite(ledPin3, HIGH);
      digitalWrite(ledPin4, HIGH);
```

```
    }
    // Turn off all LEDs
    else if (command == '0') {
      digitalWrite(ledPin1, LOW);
      digitalWrite(ledPin2, LOW);
      digitalWrite(ledPin3, LOW);
      digitalWrite(ledPin4, LOW);
    }
  }
}
```

IR sensor with Raspberry pi

```
import time import RPi.GPIO as GPIO
RUNNING = True
HIGH  = 1
LOW  = 0
DetectPin = 4
 led = 8
def InitSystem():
        GPIO.setmode(GPIO.BCM)
GPIO.setup(DetectPin,GPIO.IN,pull_up_down=GPIO.PUD_UP)
        GPIO.setup(led,GPIO.OUT)
        return

def DetectPerson():
        while True:
                input_state = GPIO.input(DetectPin)
                time.sleep(0.3)
                if input_state == 0:
                        return LOW               else:
                        return HIGH
try:
        print ("\nCounting using IR LED\n")
print  ("-----------------------------------------------\n")
        InitSystem()
        count =0;
        while RUNNING:
                state = DetectPerson()
                if state == LOW:
```

```
                    count+=1
                    print ("person count =%d" %count)
                    GPIO.output(led,LOW)
                time.sleep(1)
                    GPIO.output(led,HIGH)

# If CTRL+C is pressed the main loop is broken except
KeyboardInterrupt:
    RUNNING = False


# Actions under 'finally' will always be called
finally:
    # Stop and finish cleanly so the pins
    # are available to be used again
    GPIO.cleanup()
```