

Lab Practical 9

Problem Statement:

Write a program to control the color of the LED by turning 3 different potentiometers. One will be read for the value of Red, one for the value of Green, and one for the value of Blue.

//Code:

```
int red_light_pin= 5;
int green_light_pin = 6;
int blue_light_pin = 3;

unsigned int red,green,blue;

void setup() {
  pinMode(red_light_pin, OUTPUT);
  pinMode(green_light_pin, OUTPUT);
  pinMode(blue_light_pin, OUTPUT);
}

void loop() {
  red = analogRead(A0); // reads the value from the specified analog pin present on the particular Arduino
  board and returns a number between 0 and 1023 that is proportional to the amount of voltage being applied
  to the pin.
  red = (red/4);
  green = analogRead(A2);
  green = (green/4);
  blue = analogRead(A3);
  blue = (blue/4);

  RGB_color(255-red, 255-green, 255-blue); // turning potentiometer will show change in color
  delay(1000);

  /*RGB_color(0, 255, 255); // Red (show by removing comments without turning
  delay(1000);
  RGB_color(255, 0, 255); // Green
  delay(1000);
  RGB_color(255, 255, 0); // Blue
  delay(1000);
  RGB_color(0, 0, 125); // Raspberry
  delay(1000);
  RGB_color(255, 0, 0); // Cyan
  delay(1000);
  RGB_color(0, 255, 0); // Magenta
  delay(1000);
  RGB_color(0, 0, 255); // Yellow
  delay(1000);
  RGB_color(0, 0, 0); // White
  delay(1000);
  */
}
```

```
void RGB_color(int red_light_value, int green_light_value, int blue_light_value)
{
  analogWrite(red_light_pin, red_light_value);
  analogWrite(green_light_pin, green_light_value);
  analogWrite(blue_light_pin, blue_light_value);
}
```

OUTPUT:



Lab Practical 10

Problem Statement:

Write a program read the temperature sensor and send the values to the serial monitor on the computer.

//Code:

```
#include <SimpleDHT.h>
```

```
int pinDHT11 = A1;  
SimpleDHT11 dht11(pinDHT11);
```

```
void setup()  
{  
  Serial.begin(115200);  
}
```

```
void loop() {  
  // start working...  
  Serial.println("=====");  
  Serial.println("Sample DHT11...");  
  
  // read without samples.  
  byte temperature = 0;  
  byte humidity = 0;  
  int err = SimpleDHTErrSuccess;  
  if ((err = dht11.read(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess)  
  {  
    Serial.print("Read DHT11 failed, err=");  
    Serial.print(SimpleDHTErrCode(err));  
    Serial.print(",");  
    Serial.println(SimpleDHTErrDuration(err));  
    delay(1000);  
    return;  
  }  
}
```

```
Serial.print("Sample OK: ");  
Serial.print((int)temperature); Serial.print(" *C, ");  
Serial.print((int)humidity); Serial.println(" H");
```

```
// DHT11 sampling rate is 1HZ.  
delay(1500);  
}
```

OUTPUT:

```
Sample DHT11...
Sample OK: 33 *C, 36 H
=====
Sample DHT11...
Sample OK: 33 *C, 36 H
=====
Sample DHT11...
Sample OK: 33 *C, 36 H
=====
Sample DHT11...
Sample OK: 33 *C, 36 H
=====
Sample DHT11...
Sample OK: 33 *C, 36 H
=====
Sample DHT11...
Sample OK: 33 *C, 36 H
=====
Sample DHT11...
Sample OK: 33 *C, 35 H
=====
Sample DHT11...
Sample OK: 33 *C, 35 H
```

Lab Practical 11

Problem Statement:

Write a program using piezo element and use it to play a tune after someone knocks.

//Code:

```
const int buzzer = A1;    // LED connected to digital pin 13
const int knockSensor = A1; // the piezo is connected to analog pin 0
const int threshold = 400; // threshold value to decide when the detected sound is a knock or
not

// these variables will change:
int sensorReading = 0;    // variable to store the value read from the sensor pin

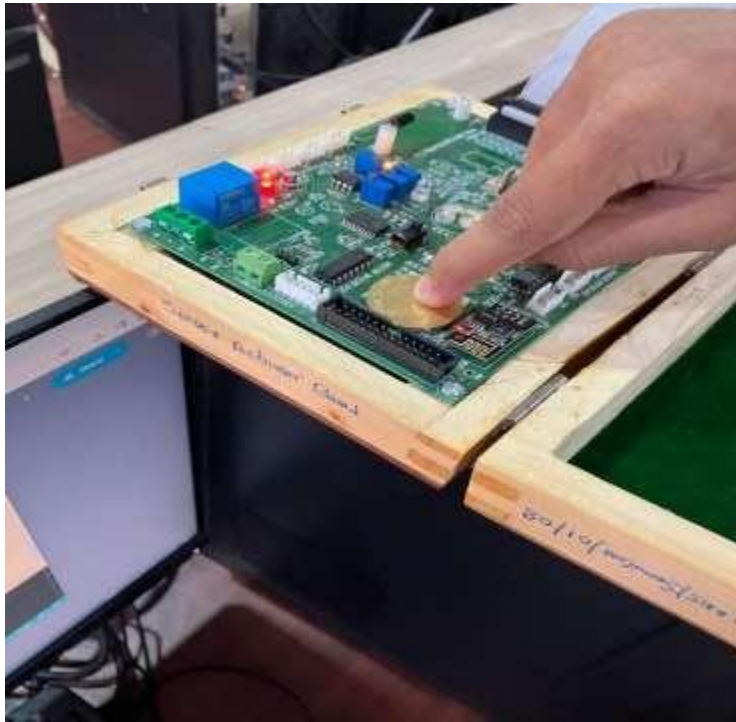
void setup() {
  pinMode(buzzer,INPUT);
}

void loop() {
  // read the sensor and store it in the variable sensorReading:
  pinMode(buzzer,INPUT);
  sensorReading = analogRead(knockSensor);

  // if the sensor reading is greater than the threshold:
  if (sensorReading >= threshold) {
    pinMode(buzzer,OUTPUT);
    tone(buzzer,261);
    // Waits some time to turn off
    delay(200);
    //Turns the buzzer off
    noTone(buzzer);
    // Sounds the buzzer at the frequency relative to the note D in Hz
    tone(buzzer,293);
    delay(200);
    noTone(buzzer);
    // Sounds the buzzer at the frequency relative to the note E in Hz
    tone(buzzer,329);
    delay(200);
    noTone(buzzer);
    // Sounds the buzzer at the frequency relative to the note F in Hz
    tone(buzzer,349);
    delay(200);
    noTone(buzzer);
```

```
// Sounds the buzzer at the frequency relative to the note G in Hz  
tone(buzzer,392);  
delay(200);  
noTone(buzzer);  
}  
delay(100); // delay to avoid overloading the serial port buffer  
}
```

OUTPUT:



Problem Statement:

Lab Practical 12

Understanding the connectivity of Raspberry-Pi /Beagle board circuit / Arduino with IR sensor. Write an application to detect obstacle and notify user using LEDs.

//Code:

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(4,OUTPUT);  
    pinMode(9,INPUT);  
    digitalWrite(4,HIGH);  
}  
  
void loop() {  
    if(digitalRead(9)== 1)  
        digitalWrite(4,HIGH);  
    else  
        digitalWrite(4,LOW);  
}
```

OUTPUT:

Before obstacle comes,

After the obstacle comes

