Lab Practical 5

Problem Statement:
Write a program using Raspberry-pi to control LED (One or more ON/OFF) or Blinking

//Code:

```
import time
from gpiozero import LED
led1 =LED(7)
led2 =LED(22)
led3 =LED(23)
led4 =LED(25)
while True:
    try:
      led1.off()
      time.sleep(0.5)
      led1.on()
      led2.off()
      time.sleep(0.5)
      led2.on()
      led3.off()
      time.sleep(0.5)
      led3.on()
      led4.off()
      time.sleep(0.5)
      led4.on()
      time.sleep(0.5)
     except KeyboardInterrupt:
      print(“Closing”)
      exit()
```

Lab Practical 6

Problem Statement:

Create a program that illuminates the green LED if the counter is less than 100, illuminate the yellow LED if the counter is between 101 and 200 and illuminates the red LED if the counter is greater than 200.

//Code:

```
import time
from gpiozero import LED
led1 = LED(7)
led2 = LED(22)
led3 = LED(23)
number=0

while True:
        time.sleep(0.2)
        if number<=100:
            led1.off()
            led2.on()
            led3.on()
        elif number>201 and number<=300:
            led1.on()
            led2.off()
            led3.on()
        elif number>101 and number<=200:
            led1.on()
            led2.on()
            led3.off()

        number=number+1
```

Lab Practical 7

Problem Statement:

Create a program so that when the user enters ‘b’ the green light blinks, ‘g’ the green light is illuminated ‘y’ the yellow light is illuminated and ‘r’ the red light is illuminated.

//Code:

```
import time
from gpiozero import LED
led1 = LED(7)
led2 = LED(22)
led3 = LED(23)
led4 = LED(25)

led1.on()
led2.on()
led3.on()
led4.on()

key=input("Enter Character")
print(key)

while True:
    time.sleep(0.2)
    key=input("Enter Character")
    print (key)
    if key=='g' or key=='G':
        led1.off()
        led2.on()
        led3.on()
        led4.on()
    elif key=='r' or key=='R':
        led1.on()
        led2.off()
        led3.on()
        led4.on()
    elif key=='y' or key=='Y':
        led1.on()
        led2.on()
        led3.off()
        led4.on()
    else :
        led1.on()
        led2.on()
        led3.on()
```

Lab Practical 8

Problem Statement:

Write a program that asks the user for a number and outputs the number squared that is entered.

//Code:

```
Num=int(input("Enter the number:"))
sq=num*num
print("The square of ",num," is: ",sq)
```

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);
}
```

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);
}
```

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
}
```

Lab Practical 12

Problem Statement:

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);
}
```

Lab Practical 13

Problem Statement:

Study of ThingSpeak – an API and Web Service for the Internet of Things (Mini Project: Same can be done parallel with PBL).

//Code :

```
import requests
import time
import random

channel_id = 394744 # PUT CHANNEL ID HERE
write_key = 'AAOG612RJT5LW21R' # PUT YOUR WRITE KEY HERE
read_key = 'TSPKJB230K65QPJQ' # PUT YOUR READ KEY HERE
if __name__ == "__main__":
        while True:
                val = random.randint(0,99)
                 #print(val)
                  URL = 'http://api.thingspeak.com/update?api_key=%s&field1=%s'%(write_key,val)
                #print(URL)
                resp = requests.get(url=URL)
                 #print(resp.json())
                # free account has an api limit of 15sec
                time.sleep(30)
```

Code:

```
import requests
import json
import time

Key = 'TSPKJB230K65QPJQ'
channel = 394744

while True:
   try:
         #URL = 'http://api.thingspeak.com/channels/%s/feeds/last.json?' %(channel)
         URL = 'http://api.thingspeak.com/channels/%s/field/1' %(channel)
         #print(URL)
         resp = requests.get(url=URL)
         #print(resp)
         resp = resp.json()['feeds'][-1]['field1']
         print(resp)
         time.sleep(0.3)
   except KeyboardInterrupt:
  exit()
```