**IOTL**

**Experiment 4**

// Define the pin where the LED is connected

int const ledPin = 12; // On most Arduino boards, the built-in LED is connected to pin 13

void setup() {

// Initialize the digital pin as an output

pinMode(ledPin, OUTPUT);

}

void loop() {

// Turn the LED on

digitalWrite(ledPin, HIGH);

// Wait for 1000 milliseconds (1 second)

delay(1000);

// Turn the LED off

digitalWrite(ledPin, LOW);

// Wait for 1000 milliseconds (1 second)

delay(1000);

}

**Experiment 5**

#define echoPin 2

#define trigPin 3

#define ledPin 13

long duration;

int distance;

void setup() {

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT); // FIX: echoPin must be INPUT

pinMode(ledPin, OUTPUT);

Serial.begin(9600);

Serial.println("Distance measurement using Arduino Uno:");

}

void loop() {

// Trigger the ultrasonic sensor

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH); // FIX: missing trigger pulse

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Read the echo time

duration = pulseIn(echoPin, HIGH);

// Calculate distance in cm

distance = duration \* 0.0344 / 2;

// Print distance

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

// Turn LED on if distance ≤ 10cm

if (distance <= 100) {

digitalWrite(ledPin, HIGH);

} else {

digitalWrite(ledPin, LOW);

}

delay(200); // Optional: small delay to reduce serial spam

}

**Experiment 6**

const int greenLED=9;

const int yellowLED=12;

const int redLED=11;

int counter= 0;

void setup(){

pinMode (greenLED,OUTPUT);

pinMode (yellowLED,OUTPUT);

pinMode (redLED,OUTPUT);

}

void loop(){

counter++;

if (counter>255){

counter=0;

}

digitalWrite(greenLED,LOW);

digitalWrite(yellowLED,LOW);

digitalWrite(redLED,LOW);

if (counter<100){

digitalWrite(greenLED,HIGH);

}

else if(counter>=101 && counter<=200){

digitalWrite(yellowLED,HIGH);

}

else if(counter>200){

digitalWrite(redLED,HIGH);

}

delay(20);

}

**Experiment 7**

// Define LED pins

const int greenPin = 6;

const int yellowPin = 2;

const int redPin = 3;

void setup() {

Serial.begin(9600); // Use 9600 for compatibility with Serial Monitor

pinMode(greenPin, OUTPUT);

pinMode(yellowPin, OUTPUT);

pinMode(redPin, OUTPUT);

// Turn off all LEDs at the start

digitalWrite(greenPin, LOW);

digitalWrite(yellowPin, LOW);

digitalWrite(redPin, LOW);

}

void loop() {

if (Serial.available() > 0) {

char input = Serial.read(); // Read incoming character

// Call functions based on input

if (input == 'b') {

blinkGreenLight();

} else if (input == 'g') {

illuminateGreen();

} else if (input == 'r') {

illuminateRed();

} else if (input == 'y') {

illuminateYellow();

}

}

}

// Blink green LED 20 times

void blinkGreenLight() {

for (int i = 0; i < 20; i++) {

digitalWrite(greenPin, HIGH);

delay(500);

digitalWrite(greenPin, LOW);

delay(500);

}

}

// Turn on only the green LED

void illuminateGreen() {

digitalWrite(greenPin, HIGH);

digitalWrite(yellowPin, LOW);

digitalWrite(redPin, LOW);

}

// Turn on only the red LED

void illuminateRed() {

digitalWrite(greenPin, LOW);

digitalWrite(yellowPin, LOW);

digitalWrite(redPin, HIGH);

}

// Turn on only the yellow LED

void illuminateYellow() {

digitalWrite(greenPin, LOW);

digitalWrite(yellowPin, HIGH);

digitalWrite(redPin, LOW);

}

**Experiment 8 (AND)**

const int inputPin1=2;

const int inputPin2=3;

const int outputPin=13;

void setup(){

pinMode(inputPin1,INPUT);

pinMode(inputPin2,INPUT);

pinMode(outputPin,OUTPUT);

}

void loop(){

int inputstate1=digitalRead(inputPin1);

int inputstate2=digitalrzead(inputPin2);

if (inputstate1==HIGH && inputstate2==HIGH){

digitalWrite (outputPin,HIGH);

}

else{

digitalWrite(outputPin,LOW);

}

delay(100);

}

**Experiment 8 (OR)**

const int inputPin1=2;

const int inputPin2=3;

const int outputPin=13;

void setup(){

pinMode(inputPin1,INPUT);

pinMode(inputPin2,INPUT);

pinMode(outputPin,OUTPUT);

}

void loop(){

int inputstate1=digitalRead(inputPin1);

int inputstate2=digitalRead(inputPin2);

if (inputstate1==HIGH || inputstate2==HIGH){

digitalWrite (outputPin,HIGH);

}

else{

digitalWrite(outputPin,LOW);

}

delay(100);

}

**Experiment 9**

void setup(){

Serial.begin(9600);

while(!Serial);

Serial.println("Please enter a no.=");

}

void loop(){

if (Serial.available()>0){

int number=Serial.parseInt();

int squared=number\*number;

Serial.print("The square of");

Serial.print( number );

Serial.print(" is ");

Serial.println(squared);

delay(1000);

Serial.println("Please enter a new number: ");

}

}

**Experiment 10**

void setup(){

Serial.begin(9600);

while(!Serial);

Serial.println("Please enter a no.=");

}

void loop(){

if (Serial.available()>0){

float number=Serial.parseFloat();

float result=sqrt(number);

Serial.print("The square root of");

Serial.print( number );

Serial.print(" is ");

Serial.println(result);

delay(1000);

Serial.println("Please enter a new number: ");

}

}

**Experiment 11**

void setup(){

Serial.begin(9600);

while(!Serial);

Serial.println("Please enter a no.=");

}

void loop(){

if (Serial.available()>0){

float number=Serial.parseFloat();

float result=cbrt(number);

Serial.print("The cube root of");

Serial.print( number );

Serial.print(" is ");

Serial.println(result);

delay(1000);

Serial.println("Please enter a new number: ");

}

}

**Experiment 12**

void setup(){

Serial.begin(9600);

while(!Serial);

Serial.println("Please enter a no.=");

}

void loop(){

if (Serial.available()>0){

int number=Serial.parseInt();

int cube=number\*number\*number;

Serial.print("The cube of");

Serial.print( number );

Serial.print(" is ");

Serial.println(cube);

delay(1000);

Serial.println("Please enter a new number: ");

}

}

**Experiment 13**

const int redPin = 9;

const int greenPin = 10;

const int yellowPin = 11;

const int potPin = A0;

void setup() {

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

pinMode(yellowPin, OUTPUT);

Serial.begin(9600); // Use 9600 for compatibility with Serial Monitor

}

void loop() {

int potValue = analogRead (potPin);

int brightness= map(potValue,0,123,0,225);

analogWrite (redPin, brightness);

analogWrite (greenPin, brightness);

analogWrite (yellowPin, brightness);

Serial.print(“Potentiometer Value=”);

Serial.println(potValue);

delay(100);

}

**Experiment 14**

const int pirsensorpin = 7; // PIR sensor input pin

const int ledpin = 13; // LED output pin

void setup() {

// Initialize the PIR sensor as input and the LED as output

pinMode(pirsensorpin, INPUT);

pinMode(ledpin, OUTPUT);

// Initialize serial communication

Serial.begin(9600); // Baud rate should be 9600 or something commonly used

}

void loop() {

int motionDetected = digitalRead(pirsensorpin); // Read the state of the PIR sensor

// Print the PIR sensor current state for debugging

Serial.print("PIR sensor state: ");

Serial.println(motionDetected);

if (motionDetected == HIGH) {

digitalWrite(ledpin, HIGH); // Turn on LED when motion is detected

Serial.println("Motion Detected!");

} else {

digitalWrite(ledpin, LOW); // Turn off LED when no motion is detected

Serial.println("No motion detected.");

}

delay(500); // Wait for half a second before checking again

}