

### ***//EXP 1 LED GLOW***

```
const int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(2000);
  digitalWrite(ledPin, LOW);
  delay(2000);
} -----
```

### ***//EXP 2 LED GLOW RASP***

```
import RPi.GPIO as GPIO
import time
led_pin = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(led_pin, GPIO.OUT)
try:
  while True:
    GPIO.output(led_pin, GPIO.HIGH)
    time.sleep(2) # Wait for 2 seconds
    GPIO.output(led_pin, GPIO.LOW)
    time.sleep(2) # Wait for 2 seconds
except KeyboardInterrupt:
  GPIO.cleanup() -----
```

### ***//EXP3 READ FROM TEMP SENSOR LM-35***

```
const int lm35_pin = A1; /* LM35 O/P pin */
void setup() {
  Serial.begin(9600);
}

void loop() {
  int temp_adc_val;
  float temp_val;
  temp_adc_val = analogRead(lm35_pin); /* Read Temperature */
  temp_val = (temp_adc_val * 4.88); /* Convert adc value to equivalent
voltage */
  temp_val = (temp_val/10); /* LM35 gives output of 10mv/°C */
}
```

```

Serial.print("Temperature = ");
Serial.print(temp_val);
Serial.print("Degree Celsius");
delay(1000);
}

//DHT11
#include <DHT.h>

#define DHTPIN 2    // DHT11 connected to digital pin 2
#define DHTTYPE DHT11 // Define the sensor type
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    Serial.begin(9600);
    dht.begin();
}void loop() {
    float temp = dht.readTemperature();
    float humidity = dht.readHumidity();
    Serial.print("Temperature: ");
    Serial.print(temp);
    Serial.print(" °C ");
    Serial.print("Humidity: ");
    Serial.print(humidity);
    Serial.println(" %");
    delay(2000); // Wait for 2 seconds
} -----

//EXP4 SPI
#include <SPI.h>

void setup() {
    SPI.begin(); // Initialize SPI
    Serial.begin(9600); // Start Serial Monitor
    pinMode(SS, OUTPUT); // Set Slave Select (SS) as output
}void loop() {
    digitalWrite(SS, LOW); // Activate SPI device
    SPI.transfer(0x01); // Send a sample byte (0x01)
    digitalWrite(SS, HIGH); // Deactivate SPI device
    Serial.println("Data sent via SPI: 0x01"); // Print confirmation
    delay(1000); // Wait 1 second before repeating
} -----

```

### ***//EXP 5 I2C***

```
#include <Wire.h>

void setup() {
    Wire.begin(); // Initialize I2C communication
    Serial.begin(9600); // Start Serial Monitor
}

void loop() {
    Wire.requestFrom(0x68, 1); // Request 1 byte from device at address 0x68
    while (Wire.available()) { // Check if data is available
        char c = Wire.read(); // Read received byte
        Serial.print("Received Data: ");
        Serial.println(c); // Print received data
    }
    delay(500); // Wait before requesting again
} -----
```

### ***//EXP 6 BLUETOOTH HC-05***

```
#include <SoftwareSerial.h>

SoftwareSerial BT(10, 11); // RX, TX for HC-05

void setup() {
    BT.begin(9600); // Start Bluetooth
    pinMode(13, OUTPUT); // Buzzer pin
}

void loop() {
    if (BT.available()) {
        char data = BT.read(); // Read Bluetooth data
        if (data == '1') // If '1' received
            digitalWrite(13, HIGH); // Buzzer ON
        else if (data == '0') // If '0' received
            digitalWrite(13, LOW); // Buzzer OFF
    }
} -----=
```

### ***//EXP 7 UART***

```
void setup() {
    Serial.begin(9600); // Initialize serial communication at 9600 bps
}

void loop() {
    if (Serial.available() > 0) {
```

```
String input = Serial.readString(); // Read input from Serial Monitor
Serial.println("Received:" + input);
}
delay(100);
} -----
```

***//EXP 8 SERIAL communication SAME AS UART***

```
void setup() {
Serial.begin(9600); // Initialize serial communication at 9600 bps
}
void loop() {
if (Serial.available() > 0) {
String input = Serial.readString(); // Read input from Serial Monitor
Serial.println("Received: " + input);
}
delay(100);
} -----
```

***//EXP 9 BLYNK APP***

```
#define BLYNK_TEMPLATE_ID "TMPL3-aV5TAVh"
#define BLYNK_TEMPLATE_NAME "blink led"
#define BLYNK_AUTH_TOKEN "oQoeqyki5aNXv9HRrt7EjmkX--JBih9"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[]=BLYNK_AUTH_TOKEN;
char SSID[]=YOUR_USERNAME;
char PASS[]=YOUR_PASSWORD;
int LED_PIN=D4;
void setup()
{
Serial.begin(115200);
Blynk.begin(auth,ssid,pass);
pinMode(LED_PIN,OUTPUT);
}
void loop()
{
Blynk.run();
} -----
```

### ***//EXP 10 OBSTACLE WITH IR SENSOR***

```
#define IRpin 2
#define ledPin 13
void setup() {
    // put your setup code here, to run once:
    pinMode(IRpin,INPUT);
    pinMode(ledPin,OUTPUT);
}
void loop() {
    // put your main code here, to run repeatedly:
    int IRread = digitalRead(IRpin);
    digitalWrite(ledPin,LOW);
    if(IRread == 0){
        digitalWrite(ledPin,HIGH);
    }
} -----
```

### ***//EXP 11 OBSTACLE WITH ULTRA SONIC***

```
#define TRIG 9
#define ECHO 10
#define LED 13
void setup() {
    pinMode(TRIG, OUTPUT);
    pinMode(ECHO, INPUT);
    pinMode(LED, OUTPUT);
}
void loop() {
    digitalWrite(TRIG, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG, LOW);
    long duration = pulseIn(ECHO, HIGH);
    int distance = duration * 0.034 / 2; // Convert to cm
    if (distance > 0 && distance < 10) { // If obstacle is within 10 cm
        digitalWrite(LED, HIGH); // Turn LED ON
    } else {
        digitalWrite(LED, LOW); // Turn LED OFF
    }
}
```

```
}  
}
```

-----  
*//EXP 12 DISTANCE USING ULTRA SONIC*

```
#define TRIG 9
```

```
#define ECHO 10
```

```
void setup() {
```

```
    pinMode(TRIG, OUTPUT);
```

```
    pinMode(ECHO, INPUT);
```

```
    Serial.begin(9600); // Start Serial Monitor
```

```
}
```

```
void loop() {
```

```
    long duration;
```

```
    int distance;
```

```
    // Send trigger pulse
```

```
    digitalWrite(TRIG, LOW);
```

```
    delayMicroseconds(2);
```

```
    digitalWrite(TRIG, HIGH);
```

```
    delayMicroseconds(10);
```

```
    digitalWrite(TRIG, LOW);
```

```
    // Read echo pulse
```

```
    duration = pulseIn(ECHO, HIGH);
```

```
    // Convert duration to distance (in cm)
```

```
    distance = duration * 0.034 / 2;
```

```
    // Print distance on Serial Monitor
```

```
    Serial.print("Distance: ");
```

```
    Serial.print(distance);
```

```
    Serial.println(" cm");
```

```
    delay(500); // Small delay before next measurement
```

```
}
```