

## Experiment 1: Blink an LED (UNO and NodeMCU)

Arduino UNO:

```
void setup() {  
  
    pinMode(LED_BUILTIN, OUTPUT);  
  
}  
  
void loop() {  
  
    digitalWrite(LED_BUILTIN, HIGH);  
  
    delay(1000);  
  
    digitalWrite(LED_BUILTIN, LOW);  
  
    delay(1000);  
  
}
```

NodeMCU:

```
#define LED_BUILTIN 2  
  
void setup() {  
  
    pinMode(LED_BUILTIN, OUTPUT);  
  
}  
  
void loop() {  
  
    digitalWrite(LED_BUILTIN, LOW);  
  
    delay(1000);  
  
    digitalWrite(LED_BUILTIN, HIGH);  
  
    delay(1000);  
  
}
```

## Experiment 2(B): Frequently Used Functions (UNO and NodeMCU)

Arduino UNO & NodeMCU:

```
const int ledPin = 13;

void setup() {

    pinMode(ledPin, OUTPUT);

}

void loop() {

    digitalWrite(ledPin, HIGH);

    delay(1000);

    digitalWrite(ledPin, LOW);

    delay(1000);

}
```

### Experiment 3: LED Fading (UNO and NodeMCU)

Arduino UNO & NodeMCU:

```
int led = 9;

int brightness = 0;

int fadeAmount = 5;

void setup() {

    pinMode(led, OUTPUT);

}

void loop() {

    analogWrite(led, brightness);

    brightness += fadeAmount;

    if (brightness <= 0 || brightness >= 255) {
```

```
        fadeAmount = -fadeAmount;

    }

    delay(30);

}
```

## Experiment 4: RGB LED Control (UNO and NodeMCU)

Arduino UNO & NodeMCU:

```
int redPin = 7;

int greenPin = 6;

int bluePin = 5;

void setup() {

    pinMode(redPin, OUTPUT);

    pinMode(greenPin, OUTPUT);

    pinMode(bluePin, OUTPUT);

}

void loop() {

    setColor(255, 0, 0);

    delay(1000);

    setColor(0, 255, 0);

    delay(1000);

    setColor(0, 0, 255);

    delay(1000);

}

void setColor(int redValue, int greenValue, int blueValue) {

    analogWrite(redPin, redValue);
```

```
    analogWrite(greenPin, greenValue);

    analogWrite(bluePin, blueValue);

}
```

## Experiment 5: Ultrasonic Sensor (UNO and NodeMCU)

Arduino UNO & NodeMCU:

```
#define trigPin 11

#define echoPin 12

void setup() {

    Serial.begin(9600);

    pinMode(trigPin, OUTPUT);

    pinMode(echoPin, INPUT);

}

void loop() {

    digitalWrite(trigPin, LOW);

    delayMicroseconds(2);

    digitalWrite(trigPin, HIGH);

    delayMicroseconds(10);

    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH);

    long distance = (duration / 2) / 29.1;

    Serial.print("Distance: ");

    Serial.print(distance);

    Serial.println(" cm");

    delay(1000);

}
```

```
}
```

## Experiment 6: Temperature and Humidity (UNO and NodeMCU)

Arduino UNO & NodeMCU:

```
#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {

    Serial.begin(9600);

    dht.begin();

}

void loop() {

    float humidity = dht.readHumidity();

    float temperature = dht.readTemperature();

    Serial.print("Humidity: ");

    Serial.print(humidity);

    Serial.print("% Temperature: ");

    Serial.print(temperature);

    Serial.println("°C");

    delay(2000);

}
```

## Experiment 7: Servo Motor Control (UNO and NodeMCU)

Arduino UNO & NodeMCU:

```
#include <Servo.h>

Servo myservo;

void setup() {

    myservo.attach(9);

}

void loop() {

    for (int pos = 0; pos <= 180; pos++) {

        myservo.write(pos);

        delay(15);

    }

    for (int pos = 180; pos >= 0; pos--) {

        myservo.write(pos);

        delay(15);

    }

}
```

## Experiment 8: LCD Display (UNO and NodeMCU)

Arduino UNO:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

void setup() {

    lcd.begin(16, 2);

    lcd.print("Hello World!");

}
```

```
}
```

```
void loop() {}
```

NodeMCU:

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
void setup() {
```

```
    lcd.begin();
```

```
    lcd.print("Hello World!");
```

```
}
```

```
void loop() {}
```