IOT Part B Lab Manual

1. Write a Node MCU program to accomplish the following tasks:
a. Blink a led connected to any digital pin of the Node MCU board.

Aim:

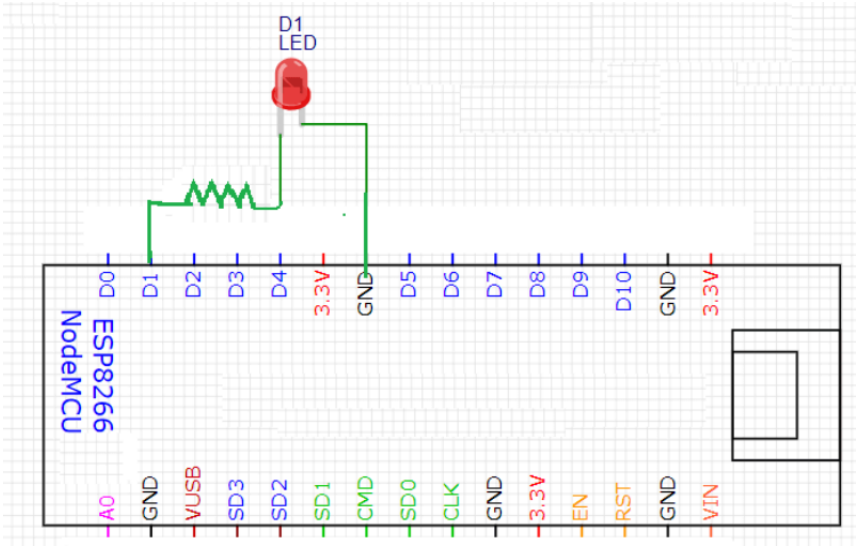
To control the state of an LED using a microcontroller (Node MCU Esp8266).

Application:

Traffic Light Simulation:

The Blink concept can be expanded to simulate a traffic light system. By connecting LEDs to different digital pins, one can create a program that mimics the changing states of a traffic light.

Circuit Diagram:



Components Required:

1. Node MCU Board
2. Micro USB Cable
3. 220Ω Resistor
4. LED
5. Jumper Wires

Procedure:

- **Open Arduino IDE, go to the File and click on the Preferences.**
- **In the Additional Boards Manager enter the URL and then click OK.**
- **open the tools in that select Board and click on the Boards Manager.**
- **Type the module name ESP8266 and select ESP8266 by community and click on the install button.**
- **To run the esp8266 with Arduino we have to select the Board:" and then change it to Node MCU 1.0 (ESP-12E Module).**

- Connect the positive/longer terminal (Anode) of the LED to the Resistor and Resistor to the digital pin of the node MCU board, i.e., PIN **D1**.
- Connect the negative/shorter terminal (Cathode) of the LED to the GND pin of the Node MCU board using the jumper wire.
- Connect the Micro USB cable.
- Select the board and serial port.
- Upload the sketch or code to the Node MCU.
- The LED will dim and light for the specified duration.

Code:

```
void setup()
{
pinMode(D1, OUTPUT);
}
void loop()
{
digitalWrite(D1, HIGH);
delay(1000);
digitalWrite(D1, LOW);
delay(1000);
}
```

Result:

D1 turns ON for 1 second.
D1 turns OFF for 1 second.
The cycle repeats continuously.

b. Fade an LED connected to any digital pin of the Node MCU board. Include the necessary steps to set up the circuit.

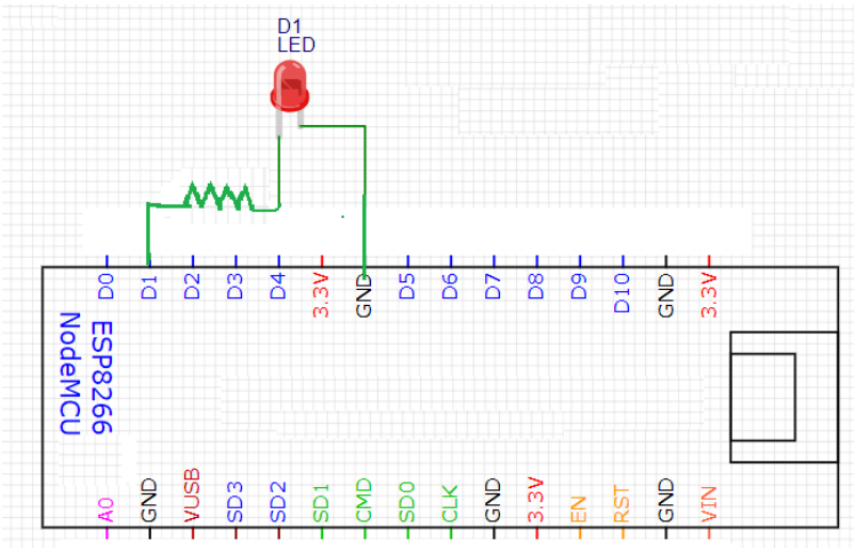
Aim:

To control the brightness of an LED using a microcontroller (Node MCU) through the Fade operation.

Application:

Dimmable Lighting System: The Fade concept can be extended to simulate a dimmable lighting system. By connecting LEDs to different digital pins and applying Fade logic, one can create a program that mimics the smooth transition of light intensity, providing a customizable ambient lighting experience.

Circuit Diagram:



Components Required:

1. Node MCU ESP8266 Board
2. Micro USB Cable
3. 220Ω Resistor
4. LED
5. Jumper Wires

Procedure:

- Connect the positive/longer terminal (Anode) of the LED to the Resistor and Resistor to the digital pin of the node MCU board, i.e., PIN **D1**.
- Connect the negative/shorter terminal (Cathode) of the LED to the GND pin of the Node MCU board using the jumper wire.
- Connect the Micro USB cable.
- Select the board and serial port.
- Upload the sketch or code to the Node MCU.
- The LED will dim and light for the specified duration.

Code:

```
int led = D1;
int brightness = 0;
int fadeAmount = 5;
 void setup()
 {
pinMode(led, OUTPUT);
 }
 void loop()
 {
analogWrite(led, brightness);
 brightness = brightness + fadeAmount;
 if (brightness <= 0 || brightness >= 255)
{
fadeAmount = -fadeAmount;
}
delay(10);
}
```

Result:

- The LED connected to the digital pin **D1** fades in and out smoothly, creating a visually pleasing transition of brightness.
- The fadeAmount variable controls the rate of brightness change, and the delay(10) introduces a short delay between each intensity adjustment, contributing to the smooth fading effect.

- The LED's brightness transitions from 0 to 255 and vice versa in a continuous loop.

2. Write a Node MCU program to control the position of the servo motor.

Aim:

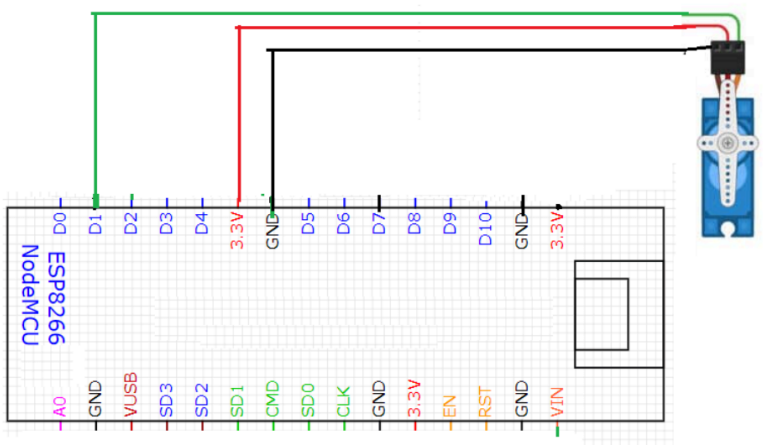
To control the position of a servo motor using a microcontroller (Node MCU) through the Servo Motor operation.

Application:

Door/Gate Control System:

The Servo Motor concept can be extended to control doors or gates in home automation systems. By connecting the servo to a door or gate mechanism, the Arduino can precisely control the position of the servo motor to open or close the door.

Circuit Diagram:



Components Required:

1. Node MCU board
2. Micro USB cable
3. Servo Motor
4. Jumper Wires

Procedure:

- The servo motor has a female connector with three pins. The darkest or even black(brown) one is usually the ground. Connect this to the Node MCU board GND.
- Connect the power cable that in all standards should be red to 3.3V on the Node MCU.
- Connect the remaining line on the servo connector (Orange) to a digital pin on the Node MCU. (**D1**) using jumper wires.
- Connect the Micro USB cable.
- Select the board and serial port in the Arduino IDE.
- Upload the sketch or code on the board.

Code:

```
#include<Servo.h>
Servo myservo;
int pos = 0;
void setup()
{
myservo.attach(D1);
}
void loop()
{
for (pos = 0; pos <= 180; pos += 1)
{
myservo.write(pos);
delay(15);
}
for (pos = 180; pos >= 0; pos -= 1)
{
myservo.write(pos);
delay(15);
}
}
```

Result:

- The servo motor connected to pin D1 smoothly sweeps from 0 to 180 degrees in one direction.

- After reaching 180 degrees, the servo motor smoothly sweeps back from 180 to 0 degrees.
- This sweeping motion repeats continuously, creating a visually perceivable back-and-forth movement.

3.Write a Node MCU program to detect obstacles using IR sensor.

Aim:

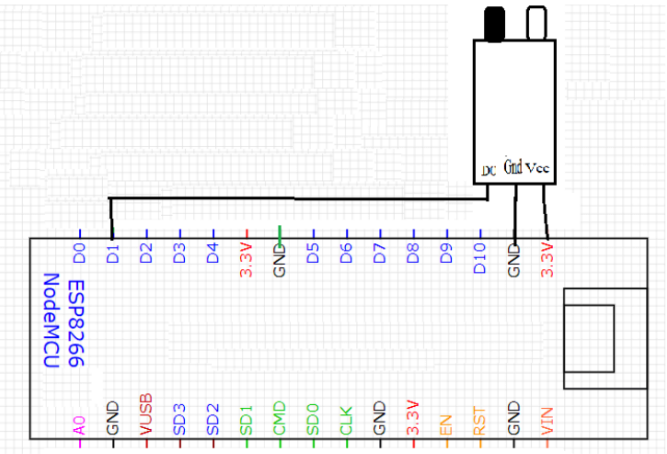
To interface an Infrared (IR) sensor with a microcontroller (Node MCU) and detect the obstacle with it.

Application:

Distance Measurement:
The IR sensor's analog output can be utilized for distance measurement. As an object approaches or moves away from the sensor,the analog voltage changes, allowing for proximity or distance detection.

Obstacle Avoidance:
In robotics, IR sensors are often employed for obstacle avoidance. The analog readings can be used to identify the presence and proximity of obstacles, enabling the robot to navigate around them.

Circuit Diagram:



Components Required:

1. Node MCU board
2. Micro USB cable
3. IR Sensor
4. Jumper Wires

Procedure:

- Connect the VCC pin of IR Sensor to the 3.3V power supply pin on the Node MCU Board.
- Connect the OUT pin to the digital pin **D1** on the Node MCU board.
- Connect the ground pin (GND) to the Ground (GND) pin on the Node MCU board using jumper wires.
- Connect the Micro USB cable.
- Select the board and serial port in the Arduino IDE.
- Upload the sketch or code on the board.

Code:

```
void setup()
{
pinMode (D1, INPUT);
Serial.begin (9600);
}
void loop()
{
int sensordata = digitalRead(D1);
if (sensordata == HIGH)
{
Serial.print("Path is clear");
Serial.println();
delay(500);
}
else
{
Serial.print("Stop something is ahead");
Serial.println();
delay(500);
```

```
}
delay(2000);
}
```

Result:

Whenever an obstacle is present in front of the IR sensor it will detect and display that “Stop something is ahead” on the serial monitor otherwise it will display that “Path is clear”.

4.Design a program to control an LED based on ambient light conditions using an LDR sensor and Node MCU.

Aim:

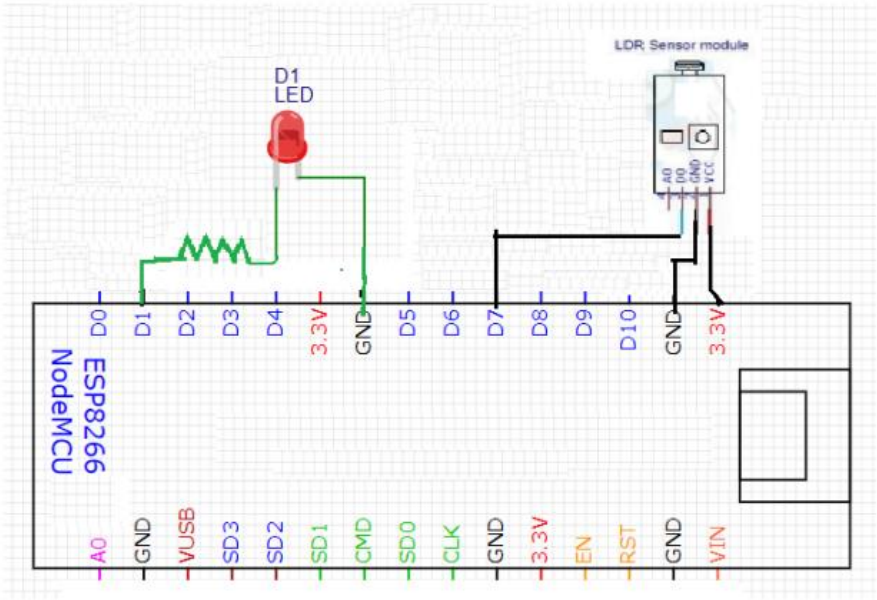
To interface a Light-Dependent Resistor (LDR) sensor with a microcontroller (Node MCU), to demonstrate how the LED can be turned on or off based on the amount of light detected by the LDR.

Application:

Ambient Light Control:

This experiment serves as the foundation for creating systems that adjust ambient lighting based on the surrounding light intensity. For example, it can be applied to automatically control indoor lighting based on natural light conditions.

Circuit Diagram:



Components Required:

1. Node MCU Board
2. Micro USB cable
3. LDR Sensor
4. LED
5. 220Ω Resistor
6. Jumper Wires

Procedure:

- Connect the VCC pin of LDR Sensor to the 3.3V power supply pin on the Node MCU Board.
- Connect the OUT pin to the digital pin **D7** on the Node MCU.
- Connect the ground pin (GND) of the LDR sensor to the Ground (GND) pin on the Node MCU board using jumper wires.
- Connect the positive/longer terminal (Anode) of the LED to the Resistor and Resistor to the digital output pin of the board, i.e., PIN **D1**.
- Connect the negative/shorter terminal (Cathode) of the LED to the GND pin of the UNO board using the jumper wire.
- Connect the Micro USB cable.
- Select the board and serial port in the Arduino IDE.
- Upload the sketch or code on the board.

Code:

```
void setup()
{
pinMode(D1,OUTPUT);  //LED
pinMode(D7,INPUT);  //LDR
Serial.begin(9600);
}
void loop()
{
int resistance= digitalRead(D7);
if (resistance==HIGH)
{
Serial.println(" it is dark time , LED is turned ON");
digitalWrite(D1,HIGH);
delay(500);
```

```
}
else
{
Serial.println(“ it is day time , LED is turned OFF”);
digitalWrite(D1, LOW);
delay(500);
}
delay(2000);
}
```

Result:

- When the light is detected from the LDR sensor it will and display that **“It's day. LED is turned off**” on the serial monitor and the LED should be turned off.
- When the light is not detected from the LDR sensor it will display that **“It's dark. LED is turned on”** and the LED should be turned on.
- The system continuously monitors the LDR sensor and categorizes light intensity based on predefined thresholds.

5.Write a Node MCU program to measure and display the distance of an object from the sensor using ultrasonic sensor.

Aim:

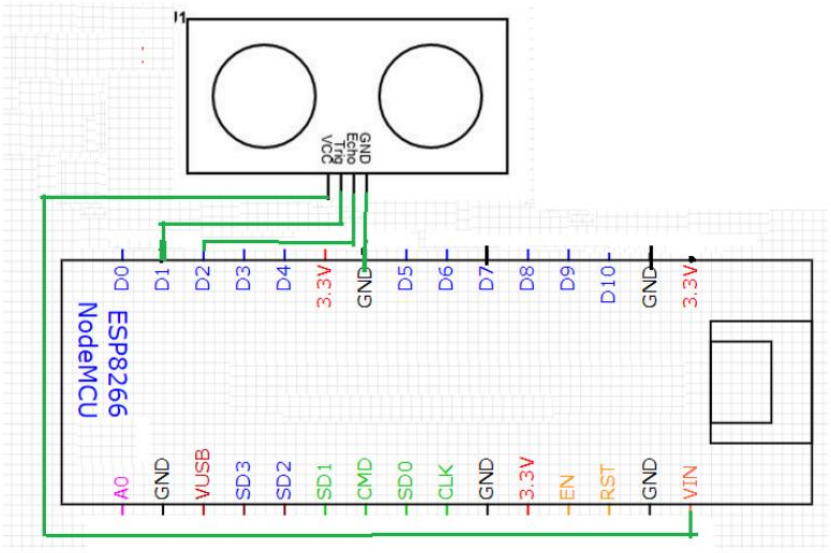
To interface an ultrasonic sensor with a microcontroller (Node MCU), aiming to measure and display the distance of an object from the sensor.

Application:

Distance Measurement System:

This experiment serves as the foundation for creating distance measurement systems. Ultrasonic sensors are commonly used in robotics, parking assistance systems, and object detection applications.

Circuit Diagram:



Components Required:

1. Node MCU Board
2. Micro USB Cable
3. Ultrasonic Sensor
4. Jumper Wires

Procedure:

- Connect the VCC pin of Ultrasonic Sensor to the 3.3V power supply pin on the Node MCU Board.
- Connect the trig pin to the digital pin **D1** on the Node MCU.
- Connect the echo pin to the digital pin **D2** on the Node MCU.
- Connect the ground pin (GND) of the Ultrasonic sensor to the Ground (GND) pin on the Node MCU board using jumper wires.
- Connect the Micro USB cable.
- Select the board and serial port in the Arduino IDE.
- Upload the sketch or code on the board.

Code:

```
#define echoPin D2
#define trigPin D1
```

```
long duration;
int distance;
void setup()
{
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Serial.begin(9600);
}
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
delay(500);
}
```

Result:

- Serial monitor displays "Distance: [Distance in cm]" every 500 milliseconds.
- The ultrasonic sensor continuously measures the distance, updating the value based on the object's proximity.