# Chapter 2: Your First ESP8266 Project

## 204335 : Microcontroller and IoT

## Part: ESP8266

Suphakit Awiphan

Chiang Mai University

# Chapter 2: Your First ESP8266 Project

204335 Microcontroller and IoT

# Chapter 2 – Your First ESP8266 Project

**Parts You’ll Need for This Chapter**

- ESP8266 board
- USB cable
- LED
- 220 Ω resistor
- Breadboard
- Jumper wires

# Outline

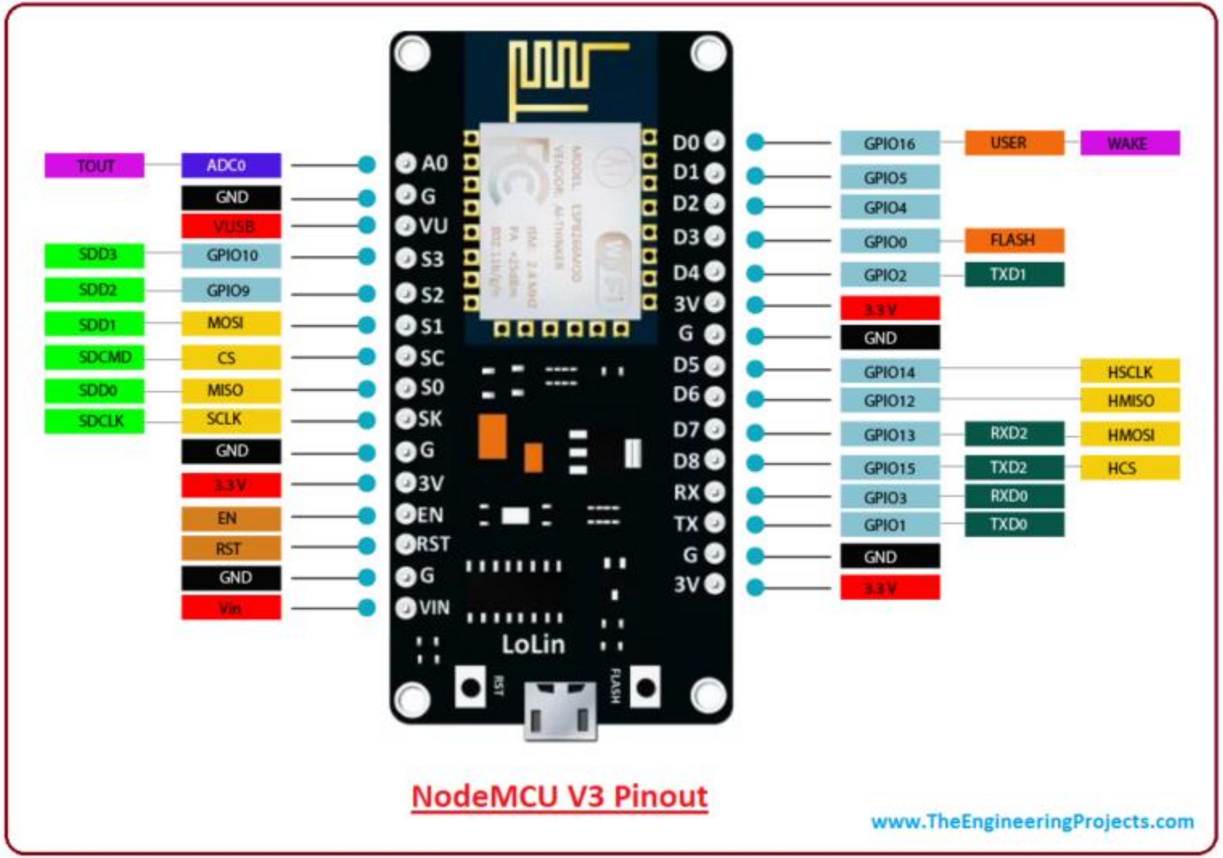
- **Functionalities of ESP8266**
- **Reading digital signals**
- **Reading analog signals**
- **Controlling an LED**
- **Dimming an LED**
- **Controlling a servo motor**
- **Measuring data from a digital sensor**

# 2.1 Functionalities of ESP8266

# Functionalities of ESP8266

- The main features of the ESP8266 can be listed as:
  - Open-source
  - Arduino-like hardware
  - Status LED
  - MicroUSB port
  - Reset/Flash buttons
  - Interactive and Programmable
  - Low cost
  - ESP8266 with inbuilt wifi
  - USB to UART converter
  - GPIO pins

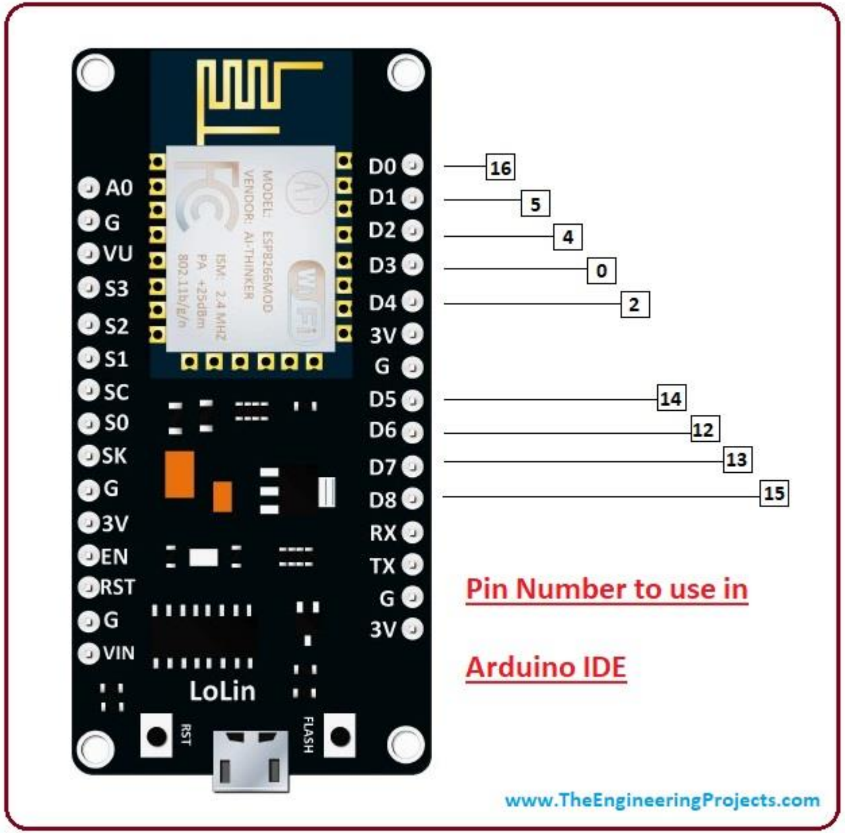
# Pin Configuration



NodeMCU V3 Pinout

www.TheEngineeringProjects.com

# Pin Configuration



Pin Number to use in

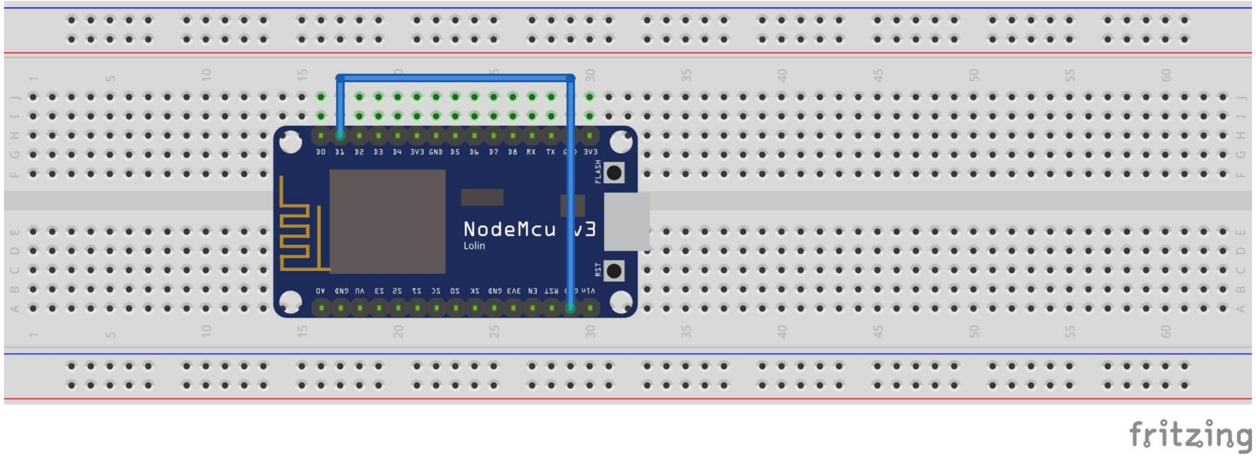
Arduino IDE

www.TheEngineeringProjects.com

# 2.2 Reading Digital Signals

# Getting Ready

- Connect a jumper wire from pin 5 (D1) to the GND pin.


D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX G 3V3
FLASH
NodeMcu v3
Lolin
RST
A0 GND VU S3 S2 S1 SC SO SK GND 3V3 EN RST G Vin
1 5 10 15 20 25 30 35 40 45 50 55 60
A B C D E F G H I J
fritzing

# How to Do It

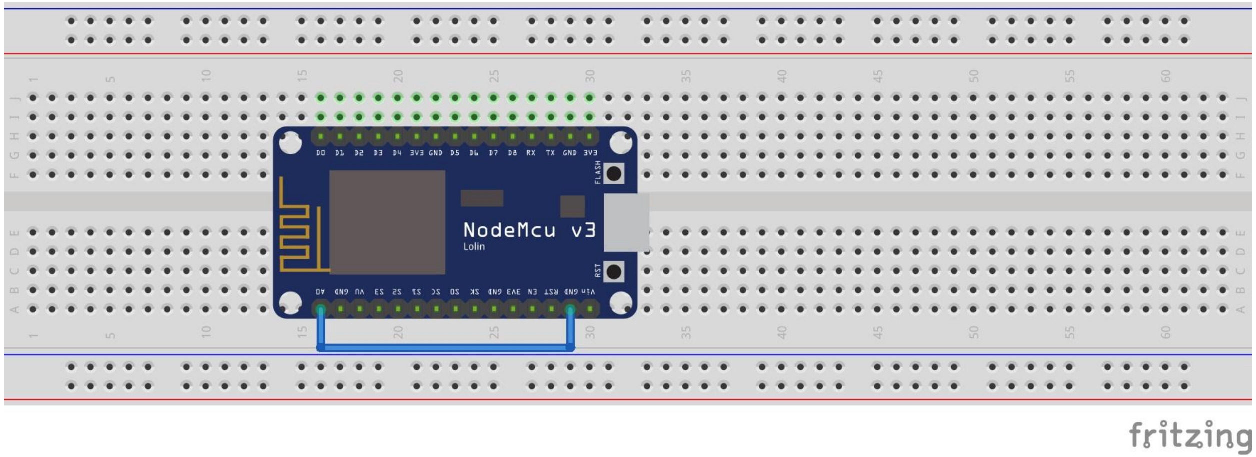
- Configure pin 5 (D1) as an input, then read it using the digitalRead() function and display the state of the input signal on the serial monitor. This will be repeated every 1 second:

```
// LED pin
int inputPin = 5;
int val = 0;

void setup() {
   Serial.begin(9600);
   pinMode(inputPin, INPUT);
}

void loop() {
   // read pin
   val = digitalRead(inputPin);
   // display state of input pin
   Serial.println(val);
   delay(1000);
}
```

# 2.3 Reading Analog Signals

# Getting Ready

- Connect a jumper wire from the analog ADC pin to the GND pin


D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
FLASH
NodeMcu v3
Lolin
RST
A0 GND VU S3 S2 S1 SC SO SK GND 3V3 EN RST GND Vin
fritzing

# How to Do It

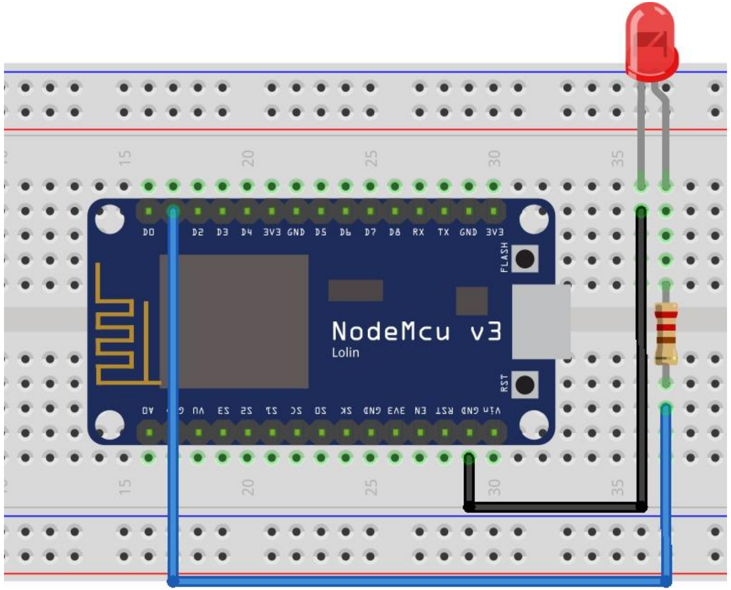
- use the analogRead() function to read the analog signal on the ADC pin and display the analog signal value on the serial monitor. This will be repeated every 1 second

```
// LED pin
int val = 0;

void setup() {
   Serial.begin(9600);
}

void loop() {
   // read pin
   val = analogRead(A0);
   // display state of input pin
   Serial.println(val);
   delay(1000);
}
```

# 2.4 Controlling an LED

# Getting Ready

- Connect one end of the 220 Ω resistor to the positive leg of the LED
- Connect the other end of the resistor to another rail of the breadboard
- Connect one end of the jumper wire to that rail and the other end of the jumper wire to pin 5 (D1)



- Take another jumper wire and connect one of its ends to the negative leg of the LED
- Connect the other end to the GND pin

# How to Do It

- Use the digitalWrite() function to output a HIGH signal on pin 5 for a duration of 1 second
- Output a low signal on pin 5 for a duration of one second.
- This will be repeated over and over again to blink the LED

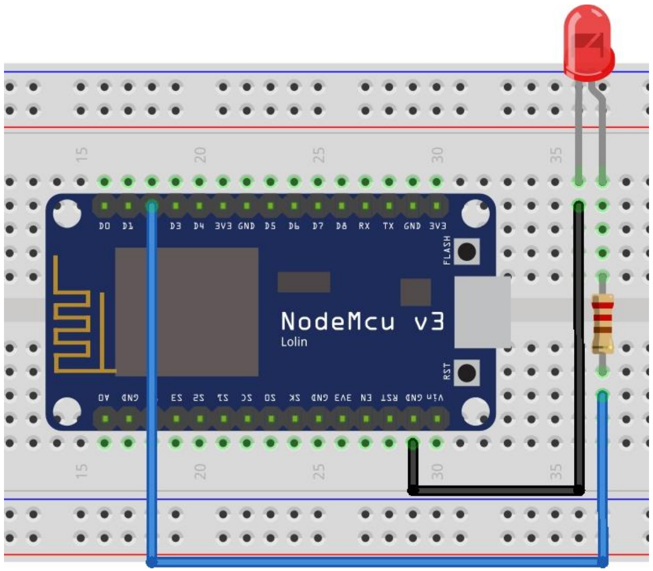
```
// LED pin
int ledPin = 5;

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

void loop() {
   // ON
   digitalWrite(ledPin, HIGH);
   delay(1000);
   // OFF
   digitalWrite(ledPin, LOW);
   delay(1000);
}
```

# 2.5 Dimming an LED

# Getting Ready

- Connect one end of the 220 Ω resistor to the positive leg of the LED
- Connect the other end of the resistor to another rail of the breadboard
- Connect one end of the jumper wire to that rail and the other end of the jumper wire to pin 4 (D2)

- Take another jumper wire and connect one of its ends to the negative leg of the LED
- Connect the other end to the GND pin


15
20
25
30
35
D0 D1 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
FLASH
NodeMcu v3
Lolin
RST
A0 GND S3 S2 S1 SC S0 SK GND 3V3 EN RST GND Vin

# How to Do It

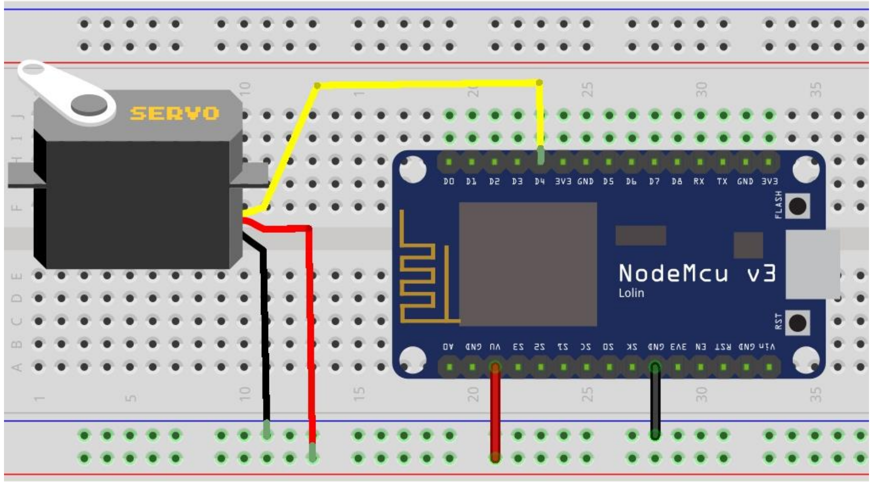
- Use the analogWrite() function to gradually reduce the duty cycle of the output signal to dim the LED.
- This will in turn reduce the brightness of the LED slowly until it completely turns off

```
int ledPin = 4;                  // LED pin
int fadeValue = 1023;            //duty cycle

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

void loop() {
   analogWrite(ledPin, fadeValue);       // glow LED
   if(fadeValue > 0)
        fadeValue --;                    // decrease duty cycle by 1
   delay(5);
}
```

# 2.6 Controlling a Servo Motor

# Getting Ready

- Use jumper cables to connect the power wires of the servo to the ESP8266 power pins
- Connect the positive terminal to the USB pin and the negative terminal to the GND pin
- Connect the signal wire of the servo motor to GPIO pin 2 (D4)


SERVO
NodeMcu v3
Lolin
D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
FLASH
RST
A0 GND VU S3 S2 S1 SC S0 SK GND 3V3 EN RST GND Vin
5
10
15
20
25
30
35
A B C D E F G H I J

# How to Do It

- With the servo library, we will use the attach() function to define the signal pin that the servo motor is connected to the write() function to instruct the servo to move to a specified position
- To demonstrate how the two functions are used, we will move the servo motor position from 0 degrees to 180 degrees and then back to 0 degrees, and repeat this forever:

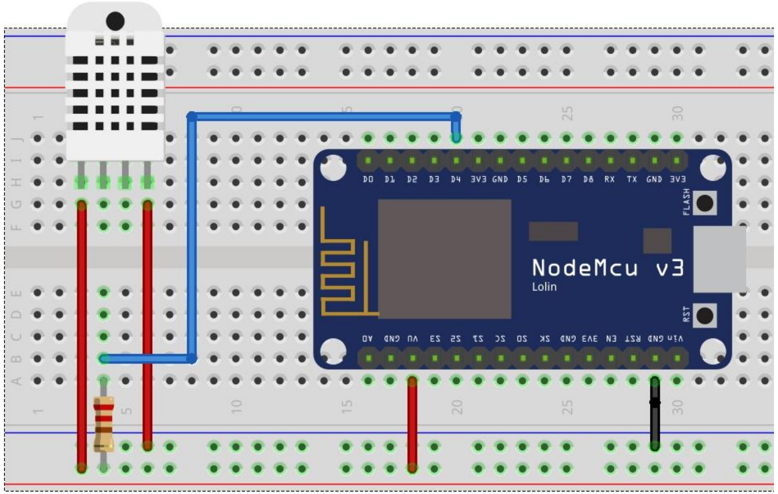
# How to Do It

```
#include <Servo.h>
Servo myservo; // create servo object to control a servo
void setup(){
   myservo.attach(2); // attach the servo on GPIO 2
}
void loop(){
   int pos; // holds the position the servo should move to
   // goes from 0 degrees to 180 degrees
   // in steps of 1 degree
   for(pos = 0; pos <= 180; pos += 1){
     myservo.write(pos); // move servo to position in var pos
     delay(15); // waits 15ms to reach the position
   }

   // goes from 180 degrees to 0 degrees
   // in steps of 1 degree
   for(pos = 180; pos>=0; pos-=1){
     myservo.write(pos); // move servo to position in var pos
     delay(15); // waits 15ms to reach the position
   }
}
```

# 2.7 Measuring Data From Digital Sensor

# Getting Ready

- Connect a 10 kΩ pull up resistor to the DHT22 data pin
- Connect the VCC pin and GND pin to the 3.3V pin and GND pin
- Connect the data pin of the DHT22 to GPIO 2 (D4)



D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
FLASH
NodeMcu v3
Lolin
RST
A0 GND VU S3 S2 S1 SC SO SK GND 3V3 EN RST GND Vin

# How to Do It

- To measure temperature and humidity readings from the DHT22 sensor, we use the DHT library from Adafruit.
- The library can be found at this link: https://github.com/adafruit/DHT-sensor-library.
- The library handles the reading of digital signals and the conversion of the digital signals to more understandable formats, such as degrees Celsius
- All this data can be accessed through the use of some library functions such as readTemperature(), readHumidity(), and computeHeatIndex()

# How to Do It

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
void setup() {
   Serial.begin(9600);
   dht.begin();
}

void loop() {
   // Wait a few seconds between measurements.
   delay(2000);
   // get humidity reading
   float h = dht.readHumidity();
   // get temperature reading in Celsius
   float t = dht.readTemperature();
   // get temperature reading in Fahrenheit
   float f = dht.readTemperature(true);
  // Check if any reads failed and exit early
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // display data on serial monitor
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.println(" *F");
}
```