Work book – NODEMCU ESP8266

Learn-Practice-Grow

Contents

NODE MCU ESP8266 Starter Kit

Introduction

Nodemcu is an open-source platform their hardware design is open for edit/modify/build. Nodemcu Development board consists of an ESP8266 Wi-Fi-enabled chip. The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer to the ESP8266 Wi-Fi Module. As Arduino.cc began developing new MCU boards supported non-AVR processors just like the ARM/SAM MCU and utilized in the Arduino Due, they required to change the Arduino IDE so that it would be comparatively easy to modify the IDE to support alternate toolchains to permit Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and therefore the SAM Core. A core is that the assortment of software system elements needed by the Board Manager and therefore the Arduino IDE to compile an Arduino C/C++ supply file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 Wi-Fi SoC, popularly known as the ESP8266 Core for the Arduino IDE. This has become the number one software system development platform for the varied ESP8266-based modules and development boards, as well as Nodemcu.

- The Nodemcu Esp8266 Pin Details
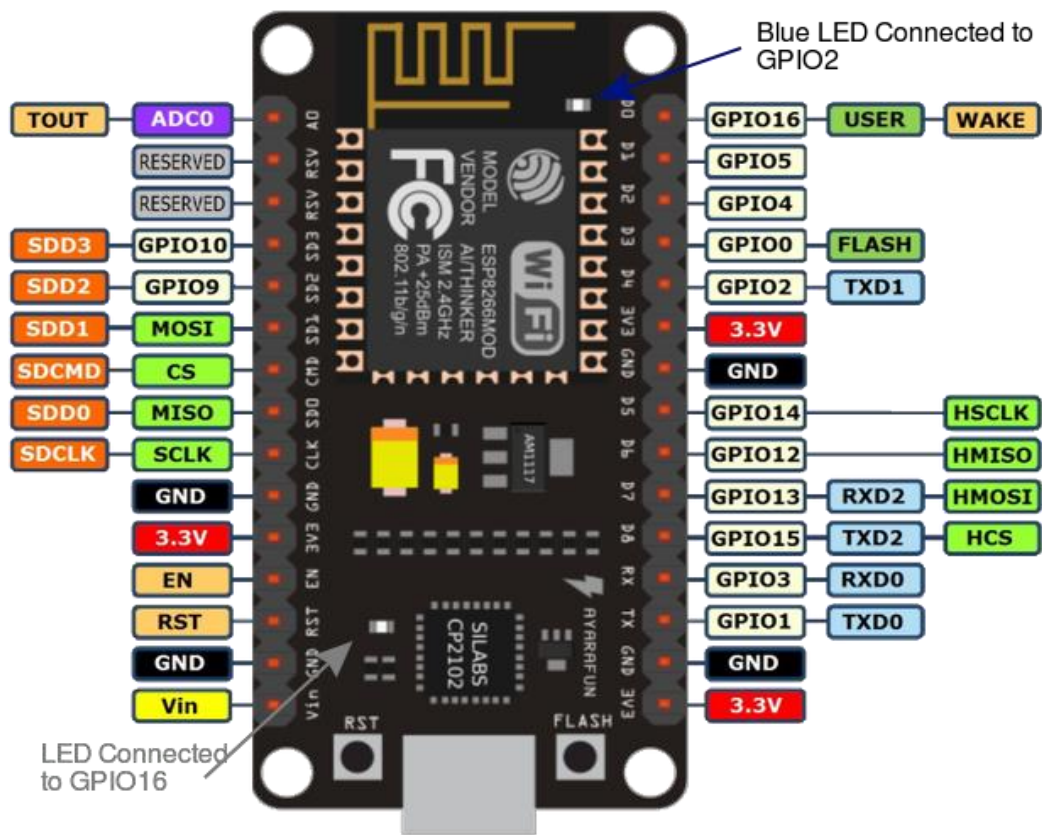
LED circuit
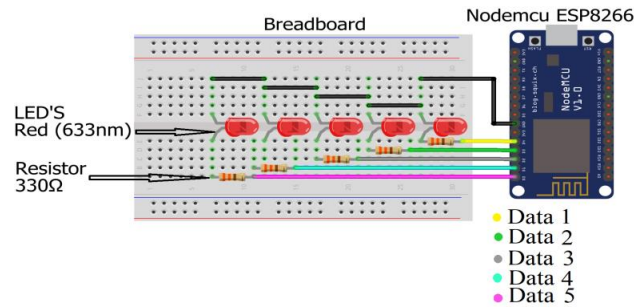
- In this circuit diagram, we have connected five LEDs in a breadboard. We can't connect the led directly into nodemcu esp8266. So that we are connecting with 330-ohm resistor in between nodemcu esp8266 to LEDs.
- The LEDs are work with both analog pins and digital pins in nodemcu esp8266 microcontroller.
- If we are connected with analog means in programming we have to write (analogWrite(pin number, analog value);) or we connected in digital means (digitalWrite(pin number, HIGH or LOW);)

Program:

```
int LED1 = D0; int LED2= D1;
int LED3 = D2; int LED4 = D3; int LED5 = D4;
void setup() {
 pinMode(LED1, OUTPUT); pinMode(LED2, OUTPUT); pinMode(LED3, OUTPUT);
pinMode(LED4, OUTPUT); pinMode(LED5, OUTPUT);
}
void loop() {
digitalWrite(LED1, HIGH);   // turn the LED on (HIGH is the voltage level)
digitalWrite(LED2, HIGH);
digitalWrite(LED3, HIGH);
digitalWrite(LED4, HIGH);
digitalWrite(LED5, HIGH);    delay(1000);   // wait for a second
digitalWrite(LED1, LOW);    // turn the LED off by making the voltage LOW
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
digitalWrite(LED4, LOW);
digitalWrite(LED5, LOW);     delay(1000);   // wait for a second
}
```

LDR Sensor
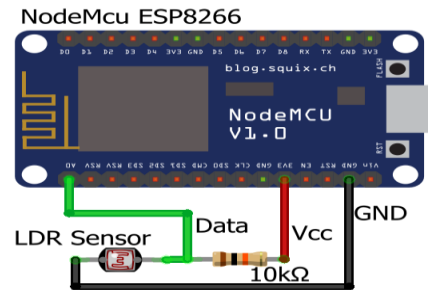
- The LDR sensor (Light Dependent Resistor) is also called a photoresistor or a cadmium sulfide (CdS) cell. This optoelectronic device is mostly used to find the light intensity, and light and dark activated switching sensor.
- An LDR or a photoresistor is a device whose resistivity is a function of the incident electromagnetic radiation. Hence, they are light-sensitive devices. They are also called as photoconductors, photoconductive cells or simply photocells
- LDR sensor output is analog values. So that we are connecting with A0 pin in nodemcu.

Program

```
int LDR = A0;
void setup() {
 Serial.begin(9600);
 pinMode(LDR,INPUT);
}
void loop() {
 int ldr = analogRead (LDR);
 Serial.print("LDR:");
 Serial.println(ldr);
 if(ldr == 1)
 {
   Serial.println("yes");
 }
 else
 {
   Serial.println("no");
 }
 delay(1000);
}
```

Water Level Sensor
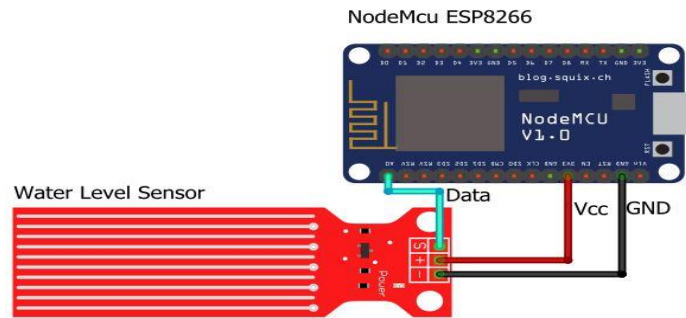
- The Water Level Depth Detection sensing element for Arduino has operational voltage DC3-5V and operational current less than 20mA.
- The sensing element is that the Analog type that produces analog output signals according to the water pressure with its Detection space of 40x16mm.
- The Water Level sensor is easy to find the values and cost-effective with a high-level drop recognition sensor by having a series of parallel wires exposed traces measured droplets/water volume therefore on see the water level. Easy to finish water to analog signal conversion and output analog values may directly browse the Arduino development board to achieve the level alarm result.
- Water level sensor output is analog values. So that we are connecting with A0 pin in nodemcu microcontroller.

Program

```
int Level = A0;
void setup() {
 Serial.begin(9600);
 pinMode(Level,INPUT);
}
void loop() {
 int level = analogRead(Level);
 Serial.print("Level:");
 Serial.println(level);
 if(level == 1)
 {
   Serial.println("yes");
 }
 else
 {
   Serial.println("no");
 }
 delay(1000);
}
```
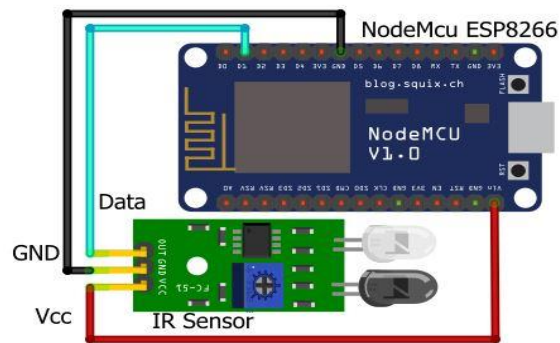
PIR Sensor
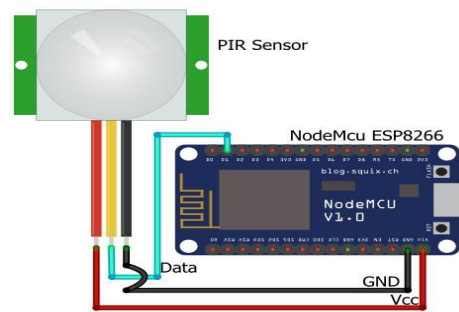
- The PIR sensor (passive infrared sensor) is using to measures infrared light radiating from objects in its field of view. PIR sensor is mostly used to find the motion of objects.
- The PIR sensor output is a digital signal. So we are connecting with digital pins in nodemcu esp8266 microcontroller. But we can connect in any digital pins in controller
- All most the sensor are given some voltage only in the output side that we are converting into digital or analog value to the microcontroller
- Using this sensor we can do any type of controlling the electronics devices or electronic drives

Program

```
int PIR = D2;
void setup() {
 Serial.begin(9600);
 pinMode(PIR,INPUT);
}
void loop() {
 int pir = digitalRead(PIR);
 Serial.print("PIR:");
 Serial.println(pir);
 if(pir == 1)
 {
  Serial.println("yes");
 }
 else
 {
  Serial.println("no");
 }
 delay(1000);
}
```
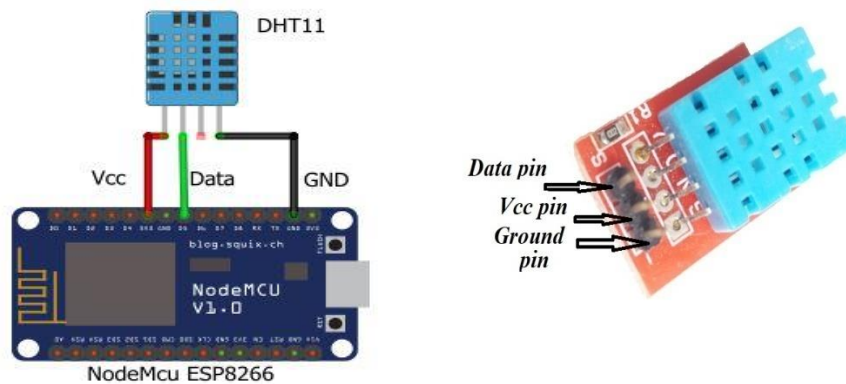
IR Sensor

- The IR sensor is using to find the obstacle and IR sensor is working transmitter and receiver. the infrared led is emitting the light energy when the light energy is getting interrupt the means the photodiode receives the light energy and the input energy is converting into digital values using LM358 IC.
- In the market the analog type sensor also available. So the connection depends upon the sensor type if it's analog means in nodemcu esp8266 we have only one analog pin that is A0 pin.
- If the sensor is digital means we have 0 to 8 digital pins so that we connect with any digital pins.

Program

```
int IR = D2;
void setup() {
 Serial.begin(9600);
 pinMode(IR,INPUT);
}
void loop() {
 int ir = digitalRead(IR);
 Serial.print("PIR:");
 Serial.println(ir);
 if(ir == 1)
 {
   Serial.println("yes");
 }
 else
 {
   Serial.println("no");
 }
 delay(1000);
}
```

DHT 11 Sensor

- This is the connection of the DHT 11 sensor with nodemcu esp8266. We can connect the data pin in any digital pins in nodemcu microcontroller.
- The DHT 11 sensor is working with library only   #include "DHT.h" this is the header file for DHT 11 sensor.
- From this sensor, we get the two digital value one is temperature and another one humidity

Program

```
#include<dht.h>
dht DHT;
#define DHT11_PIN D0
void setup() {
Serial.begin(9600);
Serial.println("welcome to TechPonder Humidity and temperature Detector");    }
void loop() {   // READ DATA
int chk = DHT.read11(DHT11_PIN);
Serial.println(" Humidity " );
Serial.println(DHT.humidity, 1);
Serial.println(" Temparature ");
Serial.println(DHT.temperature, 1);
delay(2000);
}
```

Relay with motor
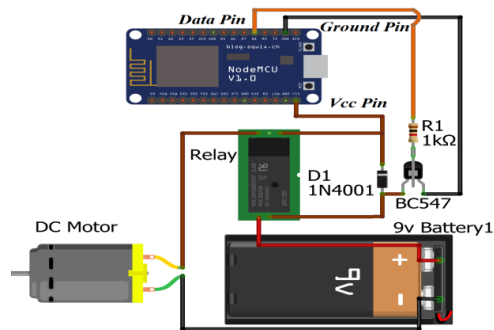
- The relay is called an electronic switch using this switch we can make on and off conditions of electrical devices.
- In this circuit diagram, we are connecting 5v DC motor in the relay board the motor power supply is given by 9V battery form in this circuit.
- Relay is working with digital input only. So we can connect the relay in any digital pins in nodemcu esp8266.

Program

```
const int LED1 = D3;
char inchar;
void setup()
{
 Serial.begin(115200);
 pinMode(LED1, OUTPUT);
 digitalWrite(LED1, LOW);
}
void loop()
{
 if (Serial.available())
 {
  inchar = Serial.read();
  if (inchar == 'A')
  {
   Serial.println("ON State");
   digitalWrite(LED1, HIGH);
  }
  else if (inchar == 'B')
  {
   Serial.println("OFF State");
   digitalWrite(LED1, LOW);

  }
  delay(100);
 }
}
```

Servo motor
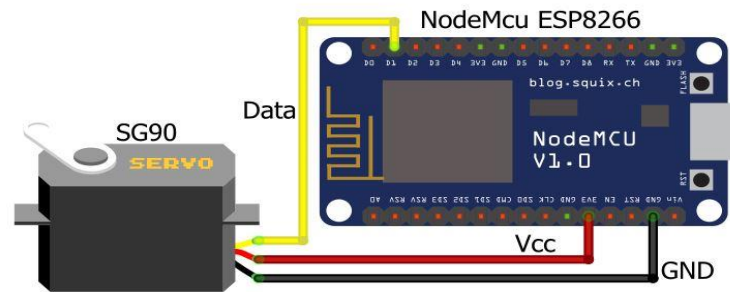
- The Servo motor SG90. light-weight with high output. The servo will rotate just about a hundred and eighty degrees (90 in each direction) and works rather like the standard types however smaller. you'll use any servo code, hardware or library to control these servos.
- Servo motor is controlled by sending an electrical pulse of variable width or pulse width modulation (PWM), through the control wire. The servo motor expects to check a pulse every twenty milliseconds (ms) and also the length of the pulse can determine however far the motor turns.

Program

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo // twelve servo objects can be
created on most boards
int pos = 0;    // variable to store the servo position
void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

Hardware tools

- Nodemcu ESP8266
- Breadboard
- DHT 11
- LDR sensor
- Servo motor SG90
- water level sensor
- PIR sensor
- IR sensor
- relay board with dc motor
- buzzer circuit
- LED circuit
- jumper wires

Software Tools

- Arduino IDE