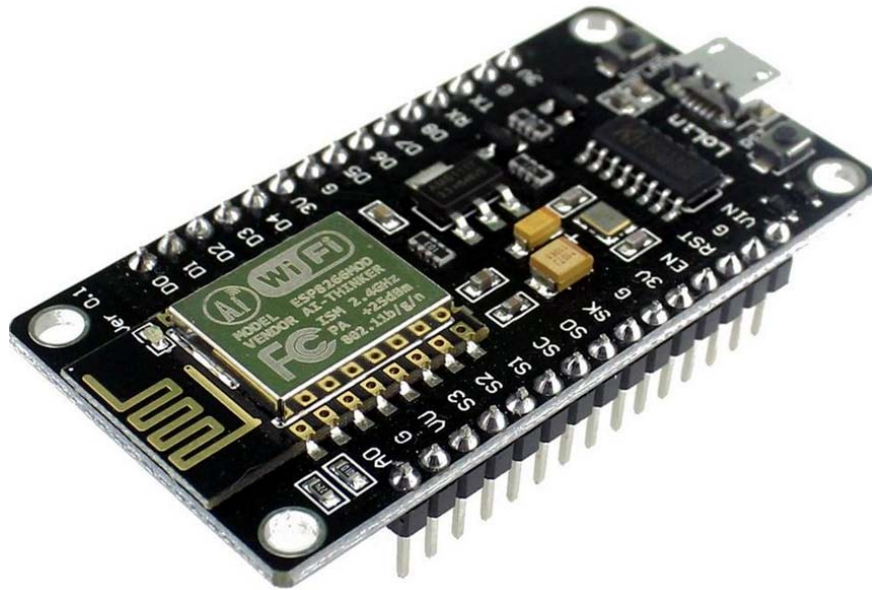


LAMPIRAN

ESP8266 NodeMCU WiFi Devkit



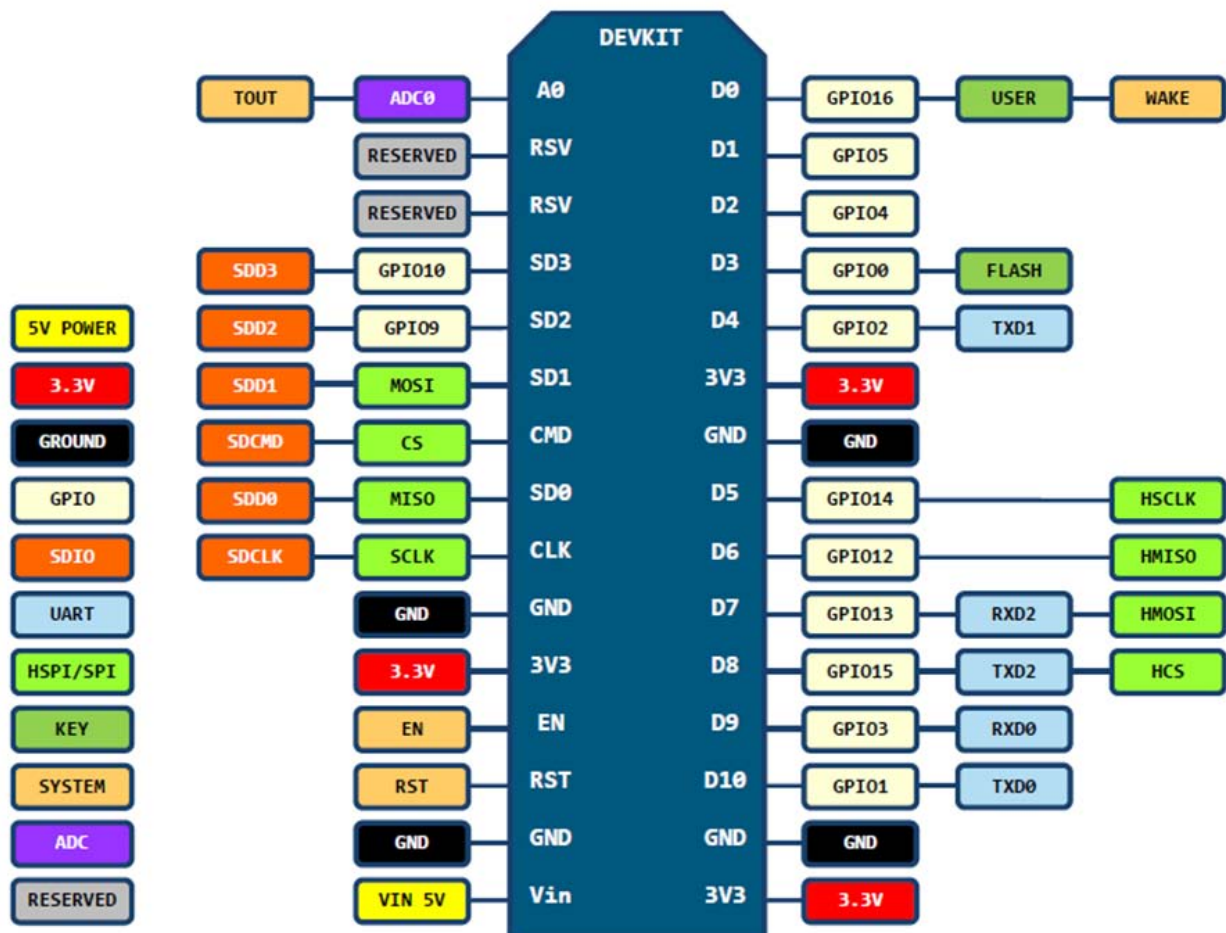
The ESP8266 is the name of a micro controller designed by Espressif Systems. The ESP8266 itself is a self-contained WiFi networking solution offering as a bridge from existing micro controller to WiFi and is also capable of running self-contained applications.

This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly.

1. Specification:

- Voltage:3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5.

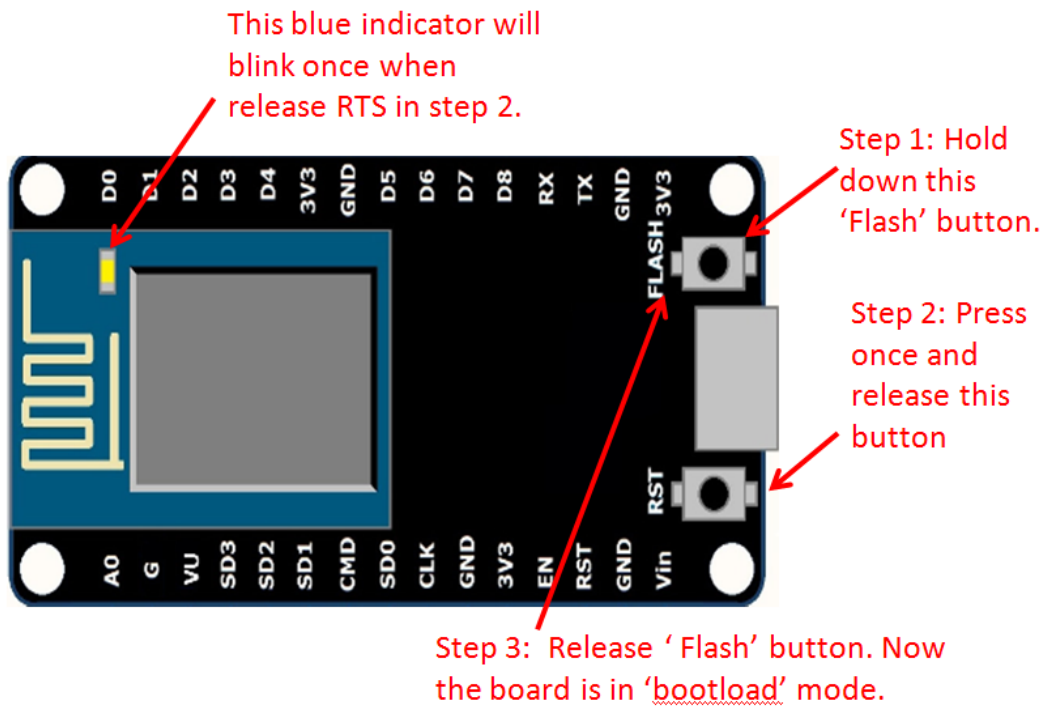
2. Pin Definition:



D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

3. Using Arduino IDE

- When you release the 'RST' button, the blue indicator will blink once, this means its ready to bootload.



Once the ESP board is in bootload mode, upload the sketch via the IDE, Figure 3-2.

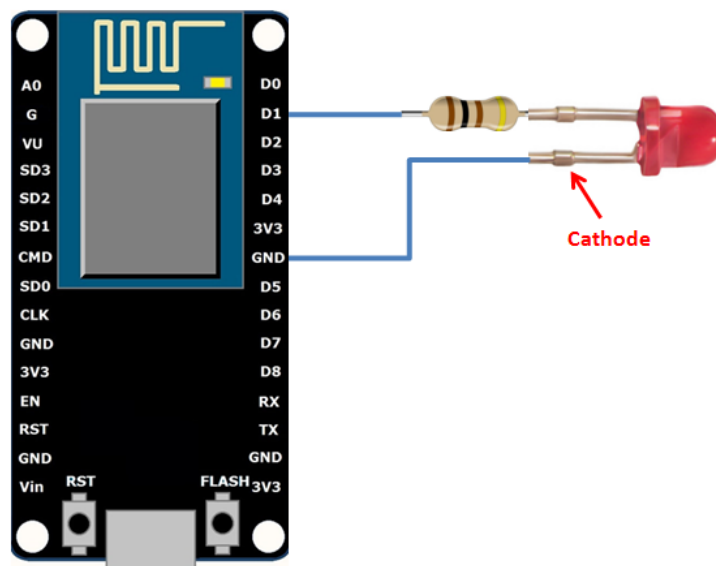
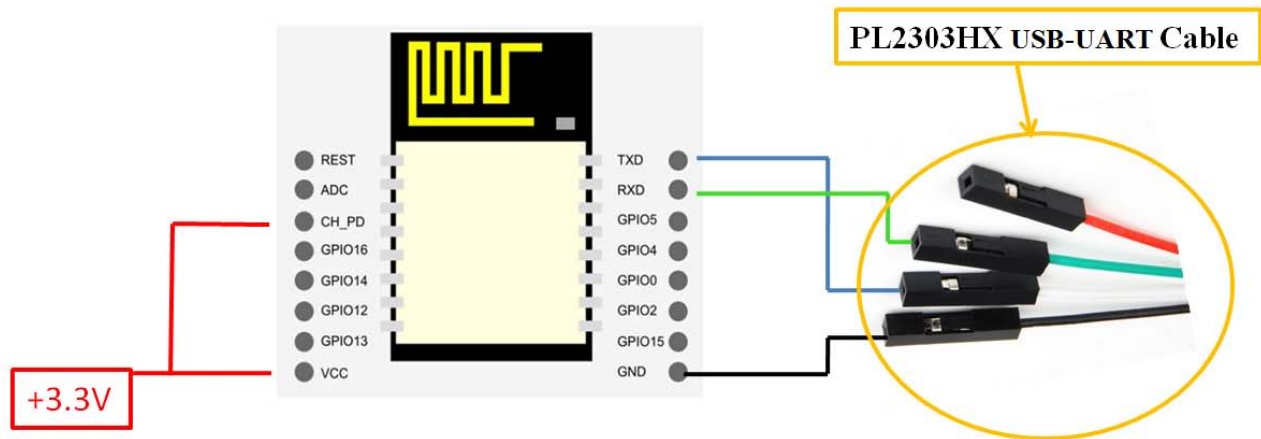


Figure3-1: Connection diagram for the blinking test

5.2 Schematics

To upload code to your ESP8266, you should connect your ESP8266 to your [PL2303HX USB-UART](#) Programming Cable like the figure below:



5.3 Writing Your Lua Script

Below is your script to blink an LED.

```
lighton=0
pin=4
gpio.mode(pin,gpio.OUTPUT)
tmr.alarm(1,2000,1,function()
    if lighton==0 then
        lighton=1
        gpio.write(pin,gpio.HIGH)
    else
        lighton=0
        gpio.write(pin,gpio.LOW)
    end
end)
```

The screenshot shows a code editor window titled 'init.lua' with the following Lua code:

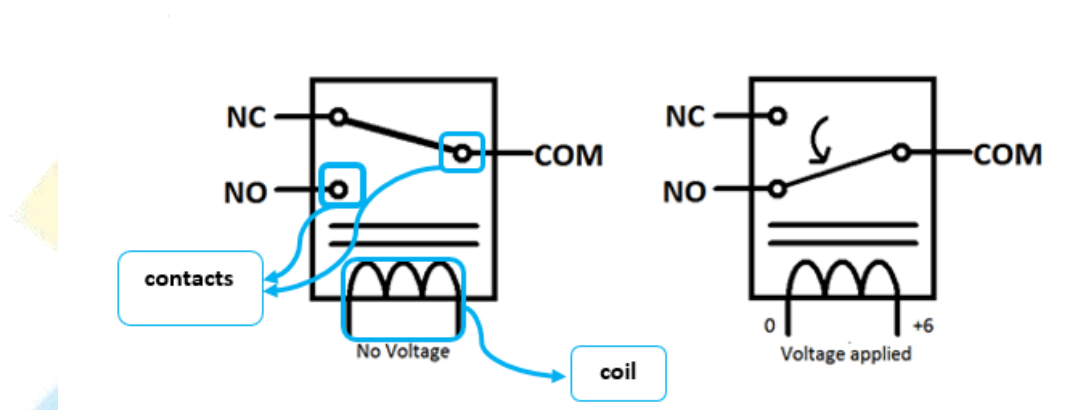
```
1 lighton=0
2 pin=4
3 gpio.mode(pin,gpio.OUTPUT)
4 tmr.alarm(1,1000,1,function()
5     if lighton==0 then
6         lighton=1
7         gpio.write(pin,gpio.HIGH)
8     else
9         lighton=0
10        gpio.write(pin,gpio.LOW)
11    end
12 end)
13
```

Right now you don't need to worry how this code works, but how you can upload it to your ESP8266.

RELAY MODULES

RELAY WORKING IDEA

Relays consist of three pins normally open pin , normally closed pin, common pin and coil. When coil power on magntic field is generated the contacts connected to each other.

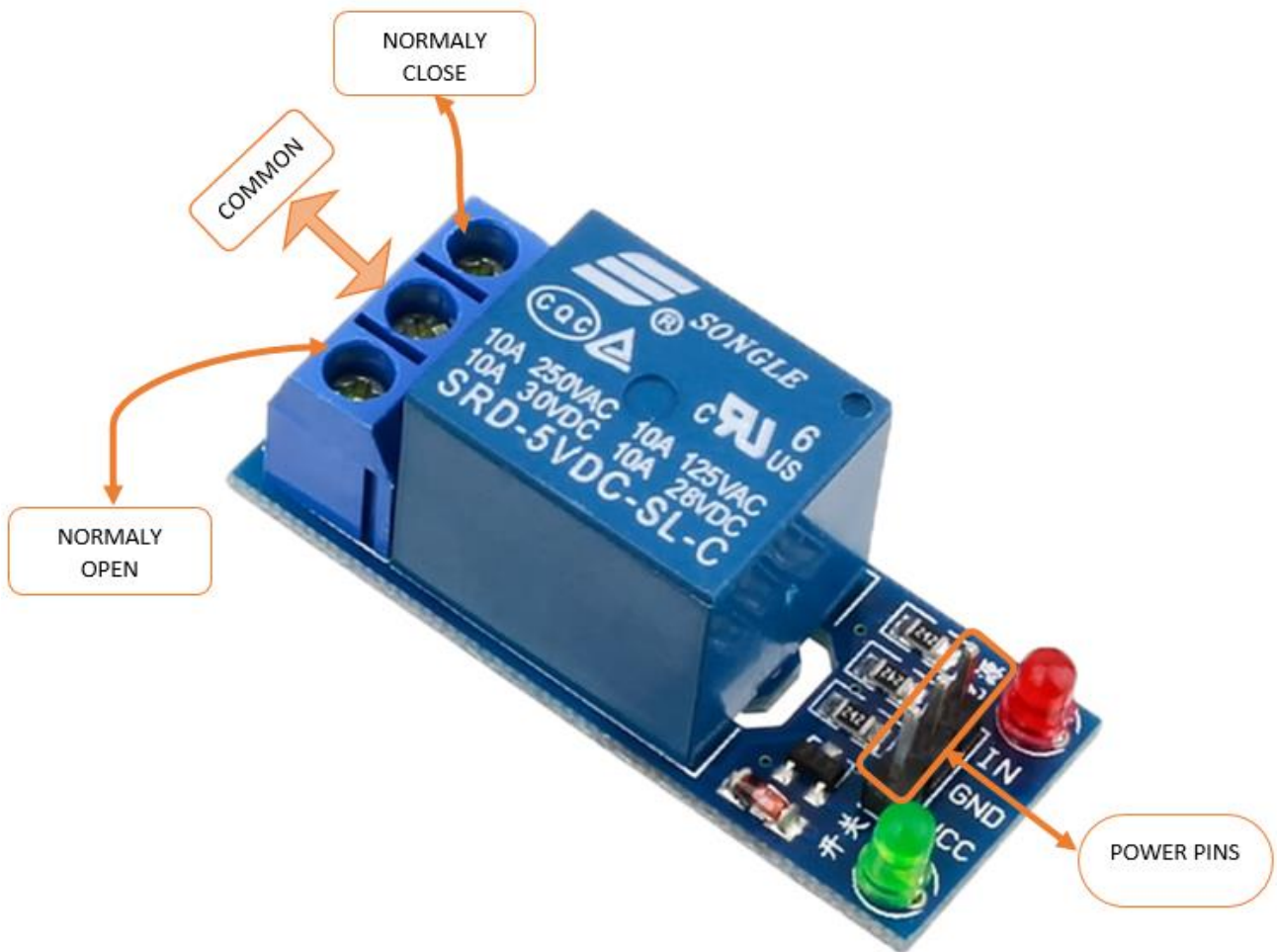


Relay modules 1-channel features

- Contact current 10A and 250V AC or 30V DC.
- Each channel has indication LED.
- Coil voltage 12V per channel.
- Kit operating voltage 5-12 V
- Input signal 3-5 V for each channel.
- Three pins for normally open and closed for each channel.

How to connect relay module with Arduino

As shown in relay working idea it depends on magnetic field generated from the coil so there is power isolation between the coil and the switching pins so coils can be easily powered from Arduino by connecting VCC and GND pins from Arduino kit to the relay module kit after that we choose Arduino output pins depending on the number of relays needed in project designed and set these pins to output and make it out high (5 V) to control the coil that allow controlling of switching process.



NOTE : whatever was the relay channels number the pinconfiguration is the same for every channel except the power pins (VCC and GND) are for the board itself. The input signal (IN) pin for every relay.

Water Sensor Module User's Manual



I. Notice

(1) did not carefully read the instructions before you do not give the driver board is powered !
Avoid faulty wiring caused permanent damage to the drive plate .

(2) Please carefully check pin function , attention condensed identifier , correct wiring ! Do not
reverse the power cord , resulting in
Electronic devices burned.

2. the product introduction

2013 latest Water Sensor is a Easy to use, compact and lightweight , high cost of water , droplets
identification and detection sensors. This sensor is working The principle is to measure the size
of the trace amount of water droplets through the line with a series of parallel wires exposed .
And domestic and foreign Products compared not only small , powerful, and cleverly designed
with the following features : First, the amount of water to simulate Conversion ; Second, plasticity
, based on the sensor output analog values ; Third, low power consumption , high sensitivity ;
Fourth, can Directly connected to a microprocessor or other logic circuitry , and the controller
board for a variety of , for example : ArduinoController , STC microcontroller , AVR microcontroller
and so on.

3, the specification parameters

- 1 Product Name: water level sensor
- 2 Item :. K-0135
- 3 Operating voltage :. DC5V
- 4 Working current : less than 20mA
- 5 Sensor Type : Analog
- 6 detection area :. 40mm x16mm
- 7 Production process :. FR4 double-sided HASL
- 8 mounting hole size : 3.0mm
- 9 user-friendly design : half-moon -slip handle depression
- 10 Working temperature :. 10 °C -30 °C
- 11 Operating Humidity : 10% ~ 90 % non -condensing
- 12 Weight :. 3g
- 13 Product Dimensions : 65mm x 20mm x 8mm

4., the test Water Sensor Module

We use the Arduino controller to be tested , need to use hardware devices as follows :

- 1, Arduino controller × 1
- 2, Arduino sensor expansion board × 1
- 3, Water Sensor Module × 1
- 4, 3P sensor cable × 2
- 5, IR & LED Modue (red) × 1
- 6, USB data communication cable × 1

Water Sensor DuPont line will be connected to the Arduino sensor expansion board interface A1. The use of sensors

The red line will be connected to the Arduino piranha light sensor expansion board D8. After completing the hardware connection , the code is compiled

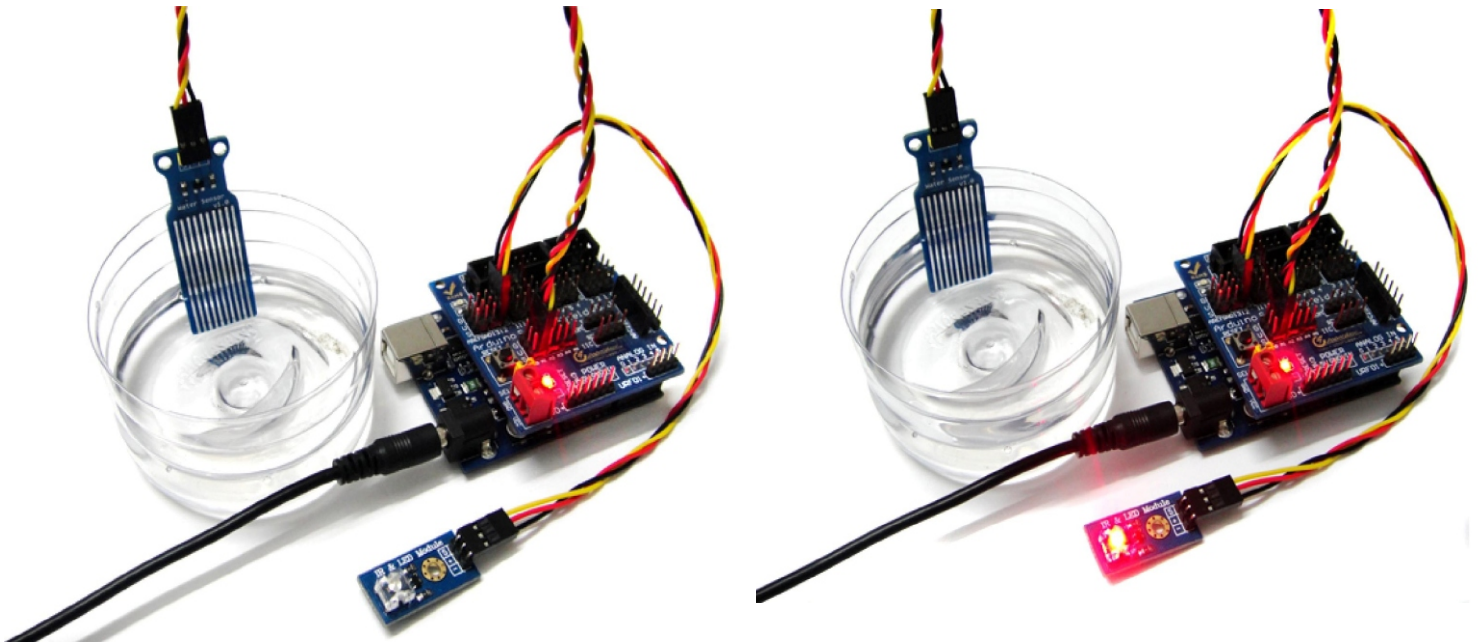
After downloading the Arduino inside .

Arduino experimental code .

```
int analogPin = 1; // level sensor connected to an analog port
int led = 12; // Piranha LED connected to digital port 12
int val = 0; // define a variable val initial value of 0
int data = 0; // define a variable data initial value of 0
void setup ()
{
  pinMode (led, OUTPUT); // define led to an output pin
  Serial.begin (9600); // set the baud rate to 9600
}

void loop ()
{
  val = analogRead (analogPin); // read the analog value to the variable val
  if (val > 700) { // determine whether more than 700 variables val
    digitalWrite (led, HIGH); when // variable val is greater than 700 , Piranha LED lights
  }
  else {
    digitalWrite (led, LOW); when // variable val is less than 700 , the lamp goes out piranha
  }
  data = val; // variable is assigned to the variable data val
  Serial.println (data); // Serial print variable data
  delay (100);
}
```

After these steps are completed, we test the low water level, see experimental phenomena :



The water level does not reach the warning value , piranhas lamp is not lit
Water level reaches and exceeds the alert value , piranha lights , initiate alarm.

Turbidity sensor SKU: SEN0189



Contents

- [1 Introduction](#)
- [2 Specification](#)
- [3 Connection Diagram](#)
- [4 Examples](#)

Introduction

The turbidity sensor detects water quality by measuring the levels of turbidity. It uses light to detect suspended particles in water by measuring the light transmittance and scattering rate, which changes with the amount of total suspended solids (TSS) in water. As the TSS increases, the liquid turbidity level increases.

Turbidity sensors are used to measure water quality in rivers and streams, wastewater and effluent measurements, control instrumentation for settling ponds, sediment transport research and laboratory measurements.

This sensor provides analog and digital signal output modes. The threshold is adjustable when in digital signal mode. You can select the mode according to your MCU.

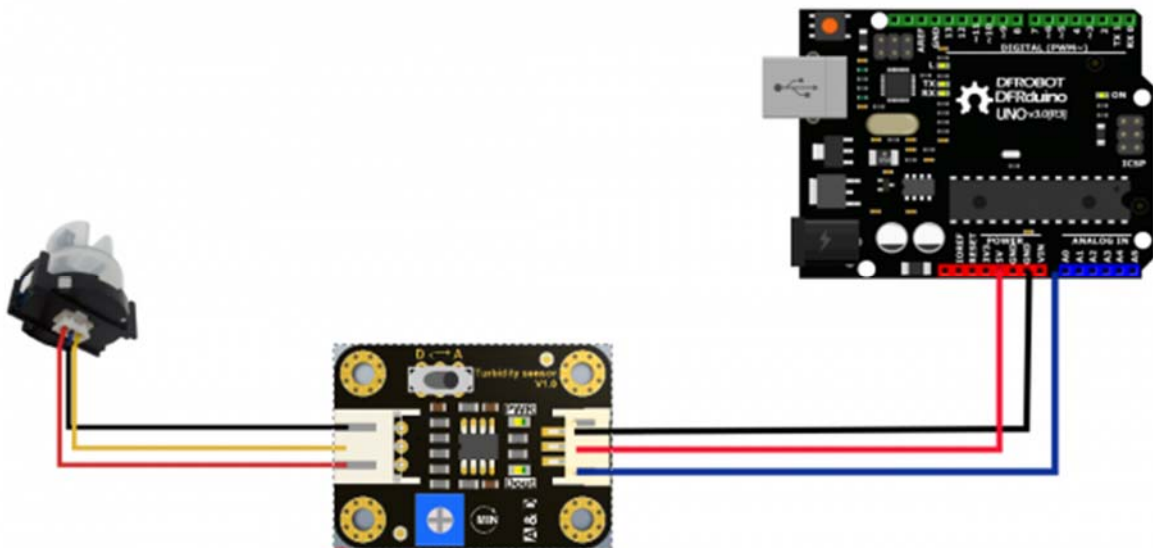


Note: The top of probe is not waterproof.

Specification

- Operating Voltage: 5V DC
- Operating Current: 40mA (MAX)
- Response Time : <500ms
- Insulation Resistance: 100M (Min)
- Output Method:
Analog output: 0-4.5V
Digital Output: High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)
- Operating Temperature: 5°C~90°C
- Storage Temperature: -10°C~90°C
- Weight: 30g
- Adapter Dimensions: 38mm*28mm*10mm/1.5inches *1.1inches*0.4inches

Connection Diagram



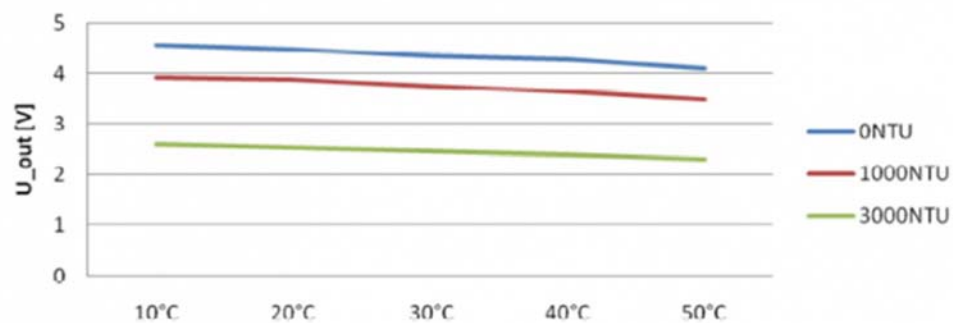
```

void setup(){
  pinMode(ledPin, OUTPUT);      // Set ledPin to output mode
  pinMode(sensor_in, INPUT);    //Set the turbidity sensor pin to input
  mode
}

void loop(){
  if(digitalRead(sensor_in)==LOW){      //read sensor signal
    digitalWrite(ledPin, HIGH);  // if sensor is LOW, then turn on
  }else{
    digitalWrite(ledPin, LOW);    // if sensor is HIGH, then turn off
    the led
  }
}
}

```

This is a reference chart for the mapping from the output voltage to the NTU according to different temperature. e.g. If you leave the sensor in the pure water, that is NTU < 0.5, it should output "4.1±0.3V" when temperature is 10~50°C.



characteristic curve "Voltage ----Temperature

Note: In the diagram, the unit measuring turbidity is shown as NTU, also it is known as JTU (Jackson Turbidity Unit), 1JTU = 1NTU = 1 mg/L. Refer to Turbidity wikipedia