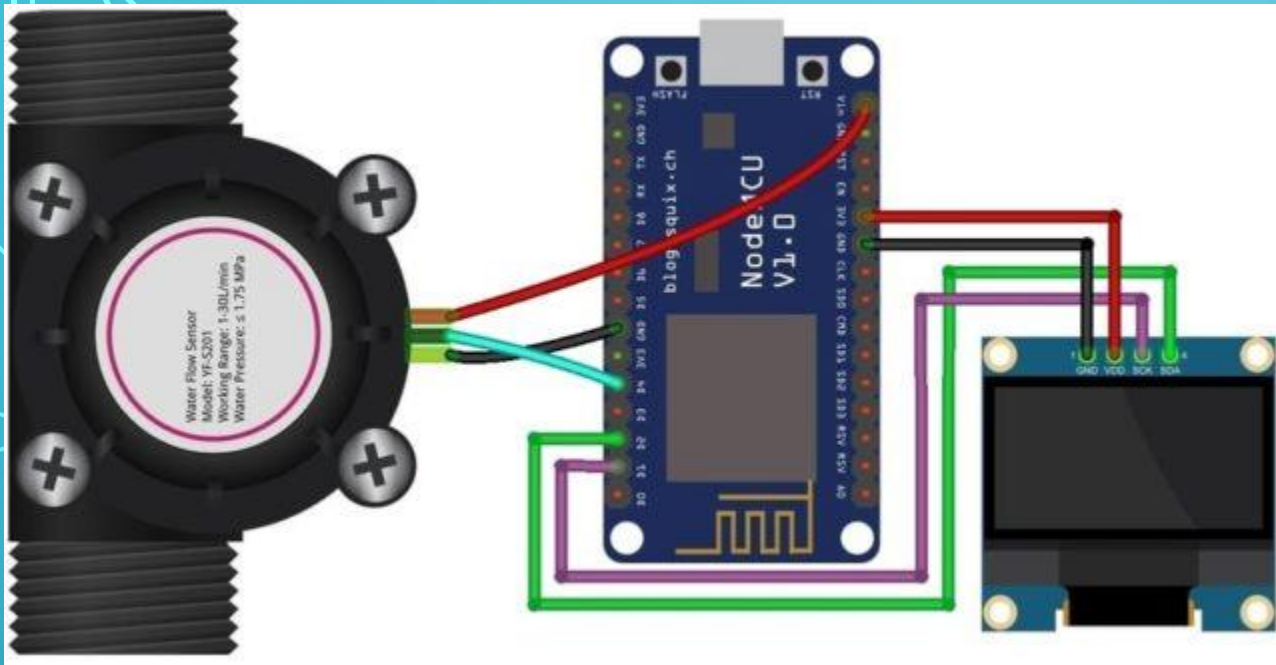


FLOW SENSOR YF-S401
MEASURING FLOW RATE & VOLUME
DISCHARGE USING NODEMCU ON
DISPLAY OLED
ALARM SIGNAL WHEN WATER STARTS
COMING IN TAP

DINESH KUMAR

ISRO SATELLITE CENTER

BANGALORE



1/8 inch Water Flow Sensor - YF-S401



Water flow sensor consists of a plastic valve body, a water rotor, and a hall effect sensor - when water flows through, the internal rotator will change its speed with different rate of flow and hall effect sensor outputs the corresponding pulse signal.

Connections

Red: Power In

Black: Ground

Yellow: Signal Output

```
#include <ESP8266WiFi.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

String apiKey = "KBD1JSZTUKCXJ15V"; // Enter your Write API key from ThingSpeak
const char *ssid = "Xiaomi_A5A0"; // replace with your wifi ssid and wpa2 key
const char *pass = "shuchi00";
const char* server = "api.thingspeak.com";
#define SENSOR 2
long currentMillis = 0;
long previousMillis = 0;
int interval = 1000;
boolean ledState = LOW;
float calibrationFactor = 4.5;
volatile byte pulseCount;
byte pulse1Sec = 0;
float flowRate;
unsigned long flowMilliLitres;
unsigned int totalMilliLitres;
float flowLitres;
float totalLitres;

void IRAM_ATTR pulseCounter()
{
  pulseCount++;
}

WiFiClient client;

void setup()
{
  Serial.begin(115200);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //initialize with the I2C addr 0x3C (128x64)
  display.clearDisplay();
  delay(10);
  pinMode(14, OUTPUT);

  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(SENSOR, INPUT_PULLUP);

  pulseCount = 0;
  flowRate = 0.0;
  flowMilliLitres = 0;
  totalMilliLitres = 0;
  previousMillis = 0;

  attachInterrupt(digitalPinToInterrupt(SENSOR), pulseCounter, FALLING);
}
```

```
void loop()
{
  currentMillis = millis();
  if (currentMillis - previousMillis > interval)
  {

    pulse1Sec = pulseCount;
    pulseCount = 0;

    flowRate = ((1000.0 / (millis() - previousMillis)) * pulse1Sec) / calibrationFactor;
    if (flowRate > 2) {digitalWrite(14,HIGH); } else{digitalWrite(14,LOW);}

    previousMillis = millis();

    flowMilliLitres = (flowRate / 60) * 1000;
    flowLitres = (flowRate / 60);

    // Add the millilitres passed in this second to the cumulative total
    totalMilliLitres += flowMilliLitres;
    totalLitres += flowLitres;

    display.clearDisplay();

    display.setCursor(10,0); //oled display
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.print("Water Flow Meter");

    display.setCursor(0,20); //oled display
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.print("R:");
    display.print(float(flowRate));
    display.setCursor(100,28); //oled display
    display.setTextSize(1);
    display.print("L/M");

    display.setCursor(0,45); //oled display
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.print("V:");
    display.print(totalLitres);
    display.setCursor(100,53); //oled display
    display.setTextSize(1);
    display.print("L");
    display.display();
  }
}
```

```
if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com
{
    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(float(flowRate));
    postStr += "&field2=";
    postStr += String(totalLitres);
    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);

}
client.stop();
}
```