PROJECT 2

Water Flow Sensor

INTRODUCTION: -

There have several case where water usage can suddenly spike, just because of water pipe broken under the ground. The sad thing is you only noticed after you get the latest water bill. So in this tutorial I will share with you on how to interface water flow sensor with ESP32 board. Later we will develop notification system if the water usage exceeds certain value.

COMPONENTS:-

1. **Water Flow Sensor**
2. **Wemos**

APPLICATION:-

Water flow sensors can measure the rate of flow of water either by measuring velocity or displacement. ...

This type of water flow sensor can be used for viscous liquids also.

For working with dirty water and wastewater which may be conductive, Magnetic flow meter is used.

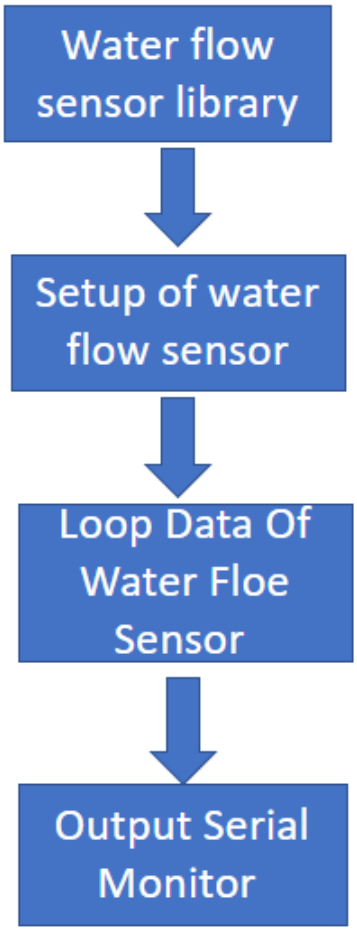
OBJECTIVES: -

1. To monitor the amount of water being supplied and used, the rate of flow of water has to be measured.

2. Water flow sensors are used for this purpose.

3. Water flow sensors are installed at the water source or pipes to measure the rate of flow of water and calculate the amount of water flowed through the pipe.

FLOW CHART :-


Water flow sensor library
Setup of water flow sensor
Loop Data Of Water Floe Sensor
Output Serial Monitor

PROGRAMMING: -

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <WiFiClient.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 = "Alexahome";     // replace with your wifi ssid and wpa2 key

const char *pass = "loranthus";
const char* server = "api.thingspeak.com";

#define LED_BUILTIN 16
#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(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;

    // Because this loop may not complete in exactly 1 second intervals we calculate

    // the number of milliseconds that have passed since the last execution and use
    // that to scale the output. We also apply the calibrationFactor to scale the output
    // based on the number of pulses per second per units
   //of measure (litres/minute in
    // this case) coming from the sensor.
    flowRate = ((1000.0 / (millis() - previousMillis)) * pulse1Sec) / calibrationFactor;
    previousMillis = millis();

    // Divide the flow rate in litres/minute by 60 to determine how many litres have
```

```
    // passed through the sensor in this 1 second interval, then multiply by 1000 to
    // convert to millilitres.
    flowMilliLitres = (flowRate / 60) * 1000;
    flowLitres = (flowRate / 60);

    // Add the millilitres passed in this second to the cumulative total

    totalMilliLitres += flowMilliLitres;
    totalLitres += flowLitres;

    // Print the flow rate for this second in litres / minute
    Serial.print("Flow rate: ");
    Serial.print(float(flowRate));  // Print the integer part of the variable
    Serial.print("L/min");
    Serial.print("\t");       // Print tab space
```

```
   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");

   // Print the cumulative total of litres flowed since
starting
```

```
  Serial.print("Output Liquid Quantity: ");
  Serial.print(totalMilliLitres);
  Serial.print("mL / ");
  Serial.print(totalLitres);
  Serial.println("L");

  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();
}
```

HARDWARE CONNECTION: -

1. **Connect Water Flow Sensor to Wemos.**
2. **Connect Water Flow Sensor Data to D2.**
3. **Connect Water Flow Sensor 5V to 5V.**
4. **Connect Water Flow Sensor GND to GND.**

CURCUIT DIAGRAM: -