LEAKAGE DETECTION USING WATER FLOW SENSOR

By

NGABOYERA Eric

MUKAMANA Florentine

NYIRANSABIMANA Esperance

Introduction

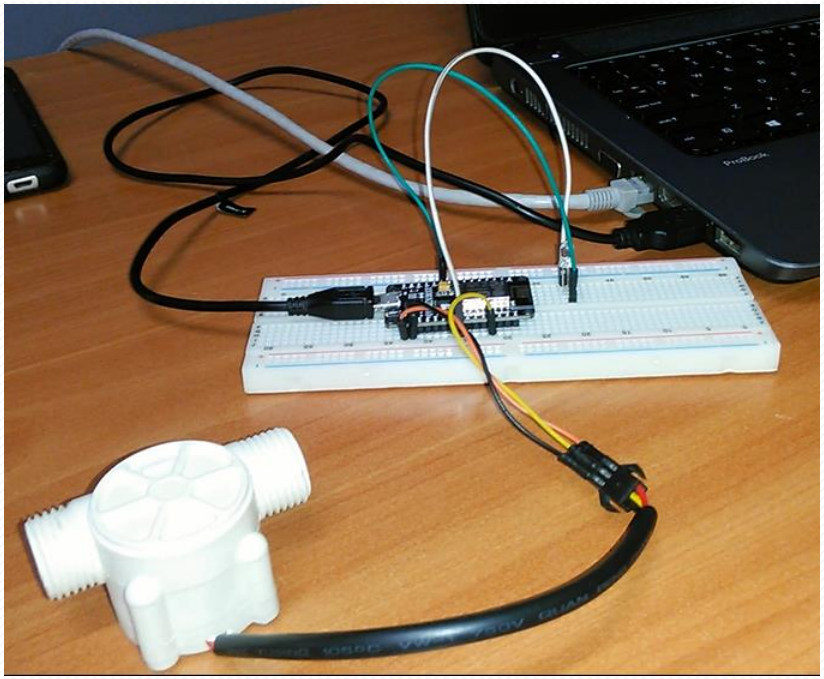
- Water flow sensor is used to detect the leakage
- It has a small rotating rotor to induce electricity within the yellow wire as a signal to the NodeMCU.

- In this project, we don’t have water pump, we use a blow within that flow to generate the signal to the NodeMCU.

PROJECT OBJECTIVES

- To detect and to determine the location of the damage when there is a leak by real time monitoring.
- To monitor the flow of water
- To alert people or the user of the pipeline when there is a leakage.

Project diagram



Hardware components

1. Water flow sensor
2. NodeMCU
3. Jumper wires
4. LED
5. Bread board
6. USB cable

Hardware connection

- The water flow sensor has three wires. Which are connected in the following way:
  - Black is connected to the ground of NodeMCU
  - Yellow is connected to digital pin 2 of the NodeMCU.
  - Red is connected to VCC of NodeMCU.
- The LED is connected to digital pin D5 and to ground of the nodeMCU

Software connection

- The software is also as for Arduino uno or another board like Arduino IDE but we have to include the driver for WiFi.
- The module ESP8266 is used
- In the code we have pubsubclient which has to upload the code to MQTT blocker.

Project codes

arduino_uno_flow_sensor_a | Arduino 1.8.9

File Edit Sketch Tools Help

arduino_uno_flow_sensor_a

```
#include<ESP8266WiFi.h>
#include<PubSubClient.h>
const char* mqtt_server = "broker.mqttdashboard.com";
WiFiClient espclient;
PubSubClient client(espclient);
const char* ssid = "Esperance";
const char* password = "Esperance2";
volatile int FlowPulse; //measuring the rising edges of the signal;
 int Calc;
int flowsensor = D3;     //The pin-2 location of the sensor Always use this pin as we are using interrupt 0
int ledPin= D5;
void callback(char* topic, byte* payload, unsigned int length)
{
} //end callback
void reconnect() {
  while (!client.connected())
  {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str()))
    {
      Serial.println("connected");
      client.publish("eric/sensor/data","Flow Sensor Readings:");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(6000);
    }
  }
} //end reconnect()
```

Project Codes Cont'd.

```
void setup_wifi(){
   WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED)
    {
      delay(500);
      Serial.print(".");
    }
  randomSeed(micros());
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
client.setServer(mqtt_server, 1883);
}
void rpm ()     //This is the function that the interupt calls
{
  FlowPulse++;  //This function measures the rising and falling edge of the hall effect sensors signal
}
void setup() {
  pinMode(ledPin,OUTPUT);
   pinMode(flowsensor, INPUT); //initializes digital pin 2 as an input
   Serial.begin(9600);         //This is the setup function where the serial port is initialised,
   attachInterrupt(0,rpm, RISING); //and the interrupt is attached on Pin 2 (INT 0)
   setup_wifi();
   reconnect();
client.setServer(mqtt_server, 1883);
}
```

Project Codes Cont'd.

```
void loop() {
  if (!client.connected()) {
  reconnect();
  }
  client.loop();
  client.setCallback(callback);
  client.loop();

 FlowPulse = 0;       //Set NbTops to 0 ready for calculations
 sei();             //Enables interrupts
 delay (1000);       //Wait 1 second
 cli();             //Disable interrupts
 Calc = (FlowPulse * 60 / 7.5); //(Pulse frequency x 60) / 7.5Q, = flow rate in L/hour
 if(Calc>50 & Calc<300)
 {
  digitalWrite(ledPin,HIGH);
 Serial.println("there is sufficient flow in the pipe");
 }
 else
 {
 digitalWrite(ledPin,LOW);
 Serial.println("leakage is detected");
 }
 Serial.print (Calc, DEC); //Prints the number calculated above
 Serial.println (" L/hour"); //Prints "L/hour"
 Serial.print("Publishing Data to Mqtt Broker: ");
 char str[16];
 itoa( Calc,str,10);
 Serial.println(str);
 client.publish("eric/sensor/data",str);
 client.publish("water flow data",str);
 Serial.println("SENT!!!!!!!!!");
 }
```

Project result

- To calculate data from sensor, the flow sensor will calculate data then will save data in Calc.
- The memory Calc will save the data.
- The condition is that the leakage will be happen when the calc is between 50 and 300 in order to avoid interference.

Simulation

- When you blog in the sensor , the serial monitor shaw various data.
- When the shown data is below 50 and above 300, there is no signal means that there is sufficient flow and the LED will be OFF.
- When the shown data is between 50 and300, there is signal means that there is a leakage and the LED will be ON.
- The code also will be uploaded to the MQTT when you blog.

Conclusion

- As The Recommendation To The User, when you have a water pump you have to adjust the value we have used above 3000 because water pump has enough power .

More info on the following links:

https://www.youtube.com/watch?v=FH84hfbNjZY&feature=youtu.be

https://www.youtube.com/watch?v=DjV5gEbKGS4&feature=youtu.be


Thank You
So Much!