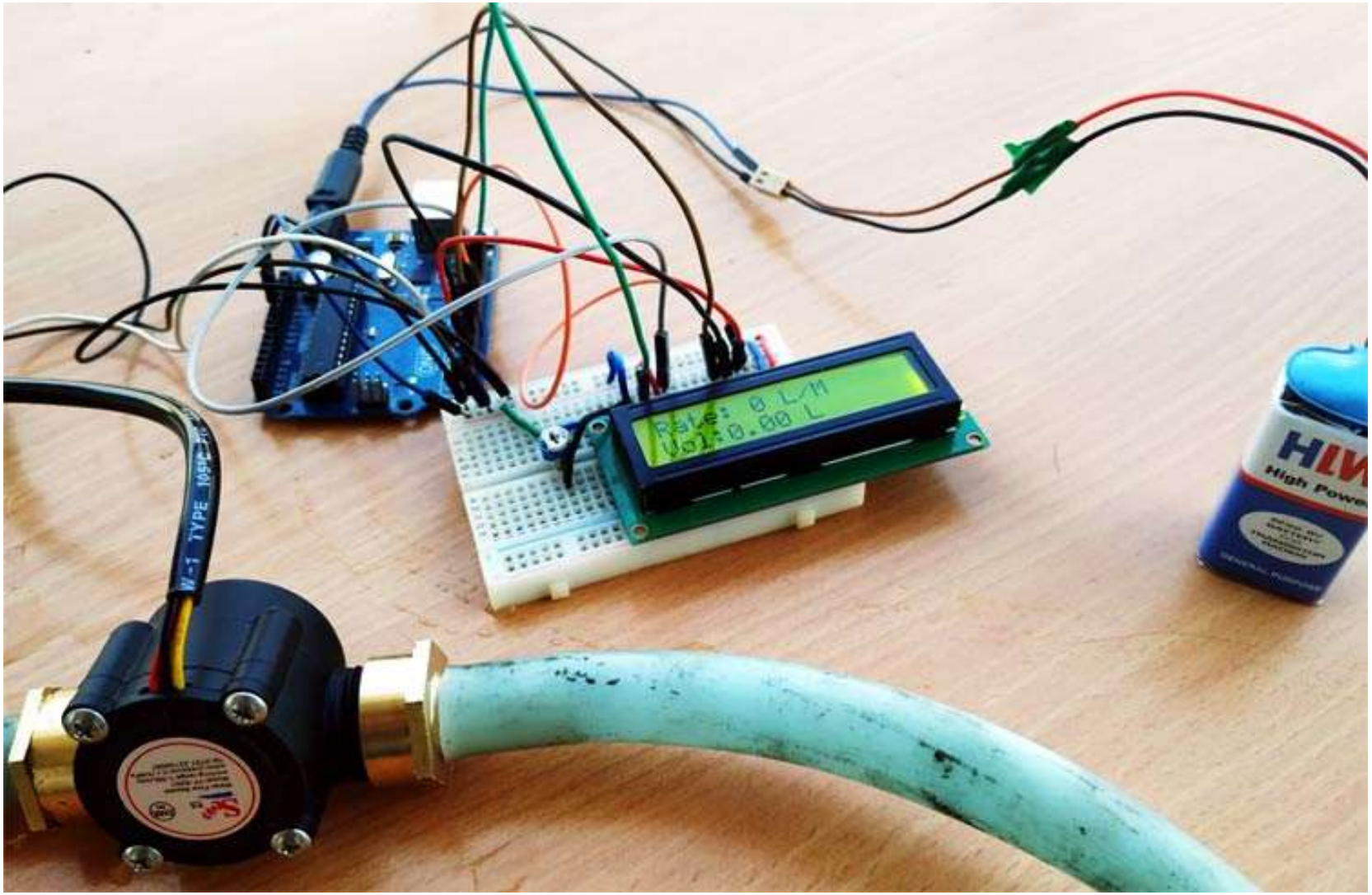
Development Part : 1 _Phase : 3

Measuring water Flow Rate and Volume using Arduino and Flow Sensor for the Given Problem Statement of Smart Water Fountains of an Flow Controlment



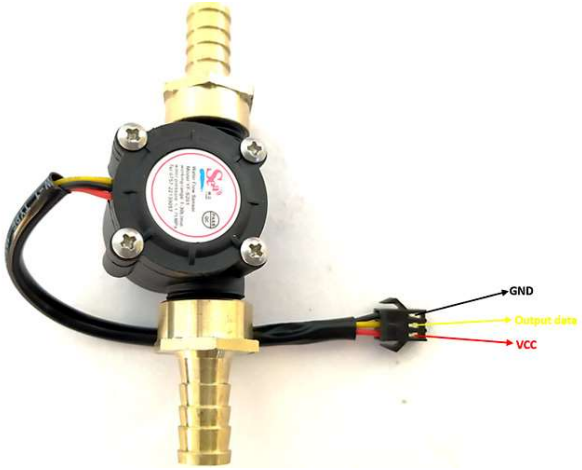
In this project, we'll create a water flow sensor with an Arduino, connecting it to an LCD screen. The Arduino will display the water volume passing through the valve. We've opted for the YF-S201 sensor, which uses hall effect for flow rate detection.

Components Required :

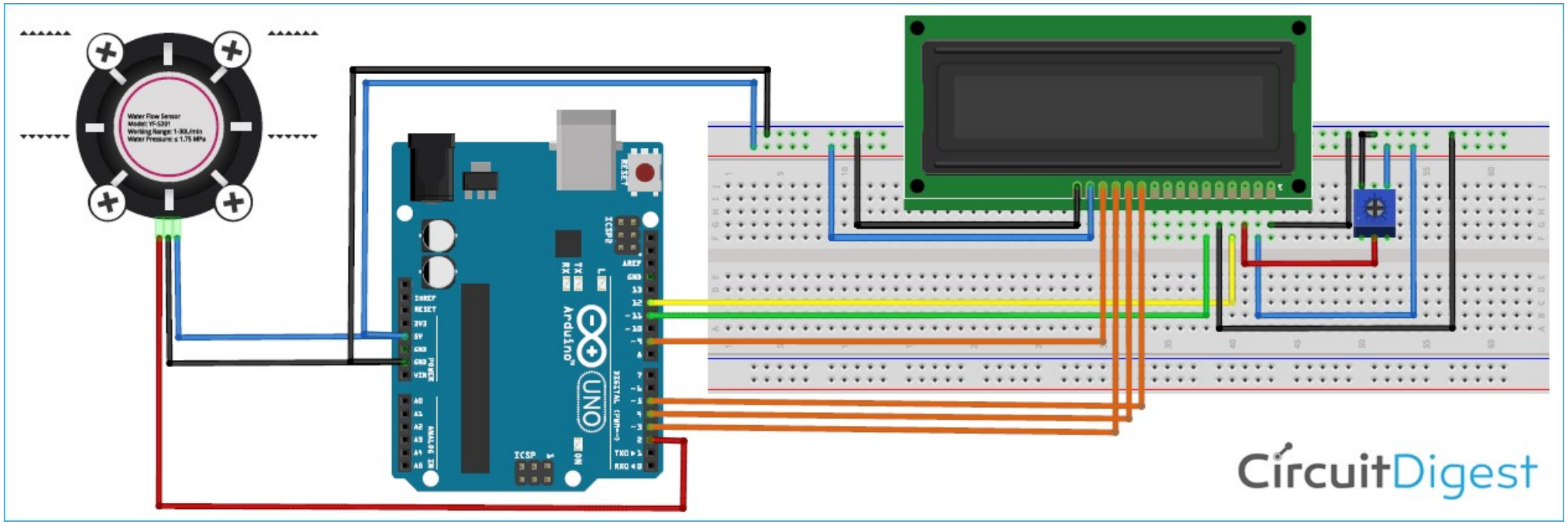
1.Water Flow Sensor
2.Arduino UNO
3.LCD (16x2)
4.Connector with internal threading
5.Connecting wires
6.Pipe

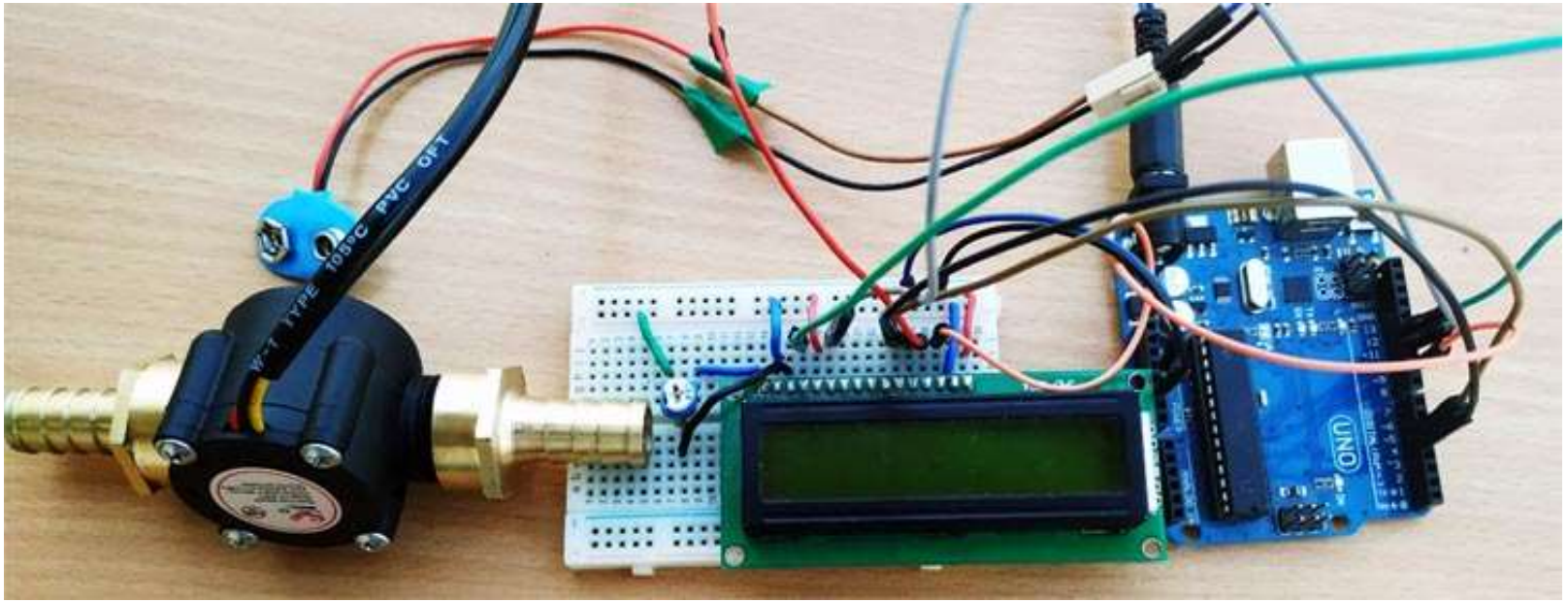
YFS201 Water Flow Sensor

The sensor has 3 wires RED, YELLOW, and BLACK as shown in the figure below. The red wire is used for supply voltage which ranges from 5V to 18V and the black wire is connected to GND. The yellow wire is used for output(pulses), which can be read by an MCU. To connect with the pipe and water flow sensor, I used two connectors with a female thread as shown below.



Circuit Diagram





S.NO	Water Flow sensor pin	Arduino Pins
1	Red Wire	5V
2	Black	GND
3	Yellow	A0

S.No	LCD	Arduino
1	Vss	GND(ground rail of breadboard)
2	VDD	5V (Positive rail of the breadboard)
3	For connection with V0 check the above note	
4	RS	12
5	RW	GND
6	E	11
7	D7	9
8	D6 to D3	3 to 5

Arduino Water Flow Sensor Code

The complete water flow sensor Arduino code is given at the bottom of the page. The explanation of the code is as follows.

We are using the header file of the LCD, which eases our interfacing the LCD with Arduino, and the pins 12,11,5,4,3,9 are allotted for data transfer between LCD and Arduino. The sensor's output pin is connected to pin 2 of Arduino UNO.

```
volatile int flow_frequency; // Measures flow sensor pulses
// Calculated litres/hour
 float vol = 0.0,l_minute;
unsigned char flowsensor = 2; // Sensor Input
unsigned long currentTime;
unsigned long cloopTime;
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 9);
```

This function is an interrupt service routine and this will be called whenever there is an interrupt signal at pin2 of Arduino UNO. For every interrupt signal, the count of the variable flow_frequency will be increased by 1. For more details on the interrupts and their working, you can read this article on Arduino interrupts.

```
void flow () // Interrupt function
{
   flow_frequency++;
}
```

In the void setup, we tell the MCU that the pin 2 of the Arduino UNO is used as INPUT by giving command pinMode(pin, OUTPUT). By using attachInterrupt command, whenever there is a rise in the signal at pin 2, the flow function is called. This increases the count in the variable flow_frequency by 1. The current time and cloopTime are used for the code to run in every 1 second.

```
void setup()
{
   pinMode(flowsensor, INPUT);
   digitalWrite(flowsensor, HIGH);
   Serial.begin(9600);
   lcd.begin(16, 2);
   attachInterrupt(digitalPinToInterrupt(flowsensor), flow, RISING); // Setup Interrupt
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Water Flow Meter");
   lcd.setCursor(0,1);
   lcd.print("Circuit Digest");
   currentTime = millis();
   cloopTime = currentTime;
}
```

```
void loop ()
{
   currentTime = millis();
   // Every second, calculate and print litres/hour
   if(currentTime >= (cloopTime + 1000))
   {
    cloopTime = currentTime; // Updates cloopTime
    if(flow_frequency != 0){
      // Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.
      l_minute = (flow_frequency / 7.5); // (Pulse frequency x 60 min) / 7.5Q = flowrate in L/hour
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Rate: ");
      lcd.print(l_minute);
      lcd.print(" L/M");
      l_minute = l_minute/60;
      lcd.setCursor(0,1);
      vol = vol +l_minute;
      lcd.print("Vol:");
      lcd.print(vol);
      lcd.print(" L");
      flow_frequency = 0; // Reset Counter
      Serial.print(l_minute, DEC); // Print litres/hour
      Serial.println(" L/Sec");
        }
```

The else function works when there is no output from the water flow sensor within the given time span.

Code :

```
/*
YF- S201 Water Flow Sensor
Water Flow Sensor output processed to read in litres/hour
Adaptation Courtesy: hobbytronics.co.uk
*/
volatile int flow_frequency; // Measures flow sensor pulses
// Calculated litres/hour
 float vol = 0.0,l_minute;
unsigned char flowsensor = 2; // Sensor Input
unsigned long currentTime;
unsigned long cloopTime;
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 9);
void flow () // Interrupt function
{
  flow_frequency++;
}
void setup()
{
  pinMode(flowsensor, INPUT);
  digitalWrite(flowsensor, HIGH); // Optional Internal Pull-Up
  Serial.begin(9600);
  lcd.begin(16, 2);
  attachInterrupt(digitalPinToInterrupt(flowsensor), flow, RISING); // Setup Interrupt
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Water Flow Meter");
  lcd.setCursor(0,1);
  lcd.print("Circuit Digest");
  currentTime = millis();
  cloopTime = currentTime;
}
void loop ()
{
```

```
  currentTime = millis();
  // Every second, calculate and print litres/hour
  if(currentTime >= (cloopTime + 1000))
  {
   cloopTime = currentTime; // Updates cloopTime
   if(flow_frequency != 0){
    // Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.
    l_minute = (flow_frequency / 7.5); // (Pulse frequency x 60 min) / 7.5Q = flowrate in L/hour
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Rate: ");
    lcd.print(l_minute);
    lcd.print(" L/M");
    l_minute = l_minute/60;
    lcd.setCursor(0,1);
    vol = vol +l_minute;
    lcd.print("Vol:");
    lcd.print(vol);
    lcd.print(" L");
    flow_frequency = 0; // Reset Counter
    Serial.print(l_minute, DEC); // Print litres/hour
    Serial.println(" L/Sec");
   }
   else {
    Serial.println(" flow rate = 0 ");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Rate: ");
    lcd.print( flow_frequency );
    lcd.print(" L/M");
    lcd.setCursor(0,1);
    lcd.print("Vol:");
    lcd.print(vol);
    lcd.print(" L");
   }
  }
}
```