### ***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](https://circuitdigest.com/microcontroller-projects/arduino-interrupt-tutorial-with-examples).

**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;**

**}**​​

The if function ensures that for every one second the code inside it runs. In this way, we can count the number of frequencies produces by the water flow sensor per second. The flow rate pulse characteristics from the datasheet are given that frequency is 7.5 multiplied by flow rate.  So the flow rate is frequency / 7.5. After finding flow rate which is in liters/minute, divide it by 60 to convert it into liter/sec. This value is added to the vol variable for every one second.

**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.

**else {**

**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");**

**}**

***Source 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("Petrol 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");  
    }  
   }  
}