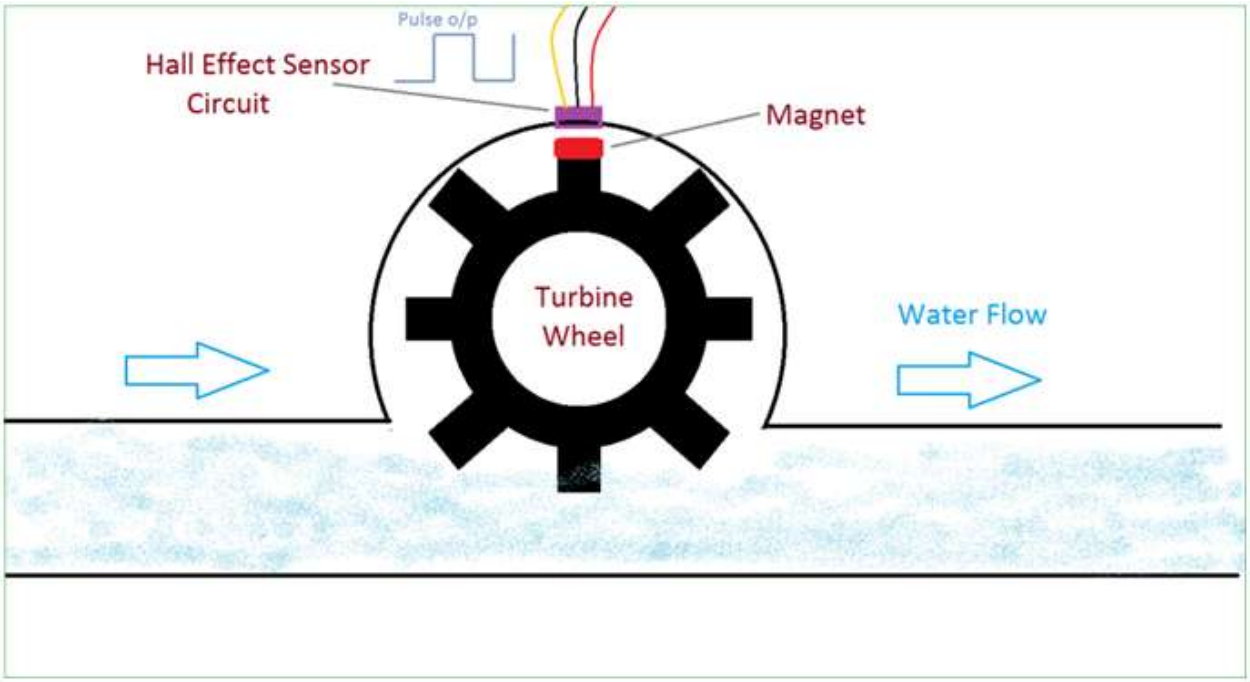
Water Flow Sensor YF-S201 Arduino Interface

It is very simple to measure the water or liquid flow by using water flow sensor YF-S201 with Arduino.

This Article describes about the water flow sensor and How the water flow sensor works then how to interface water flow sensor with Arduino.

To take control on volume we need to measure, water is essential to every thing, here this article helps you to built water flow meter to measure the volume of water flow through pipelines.

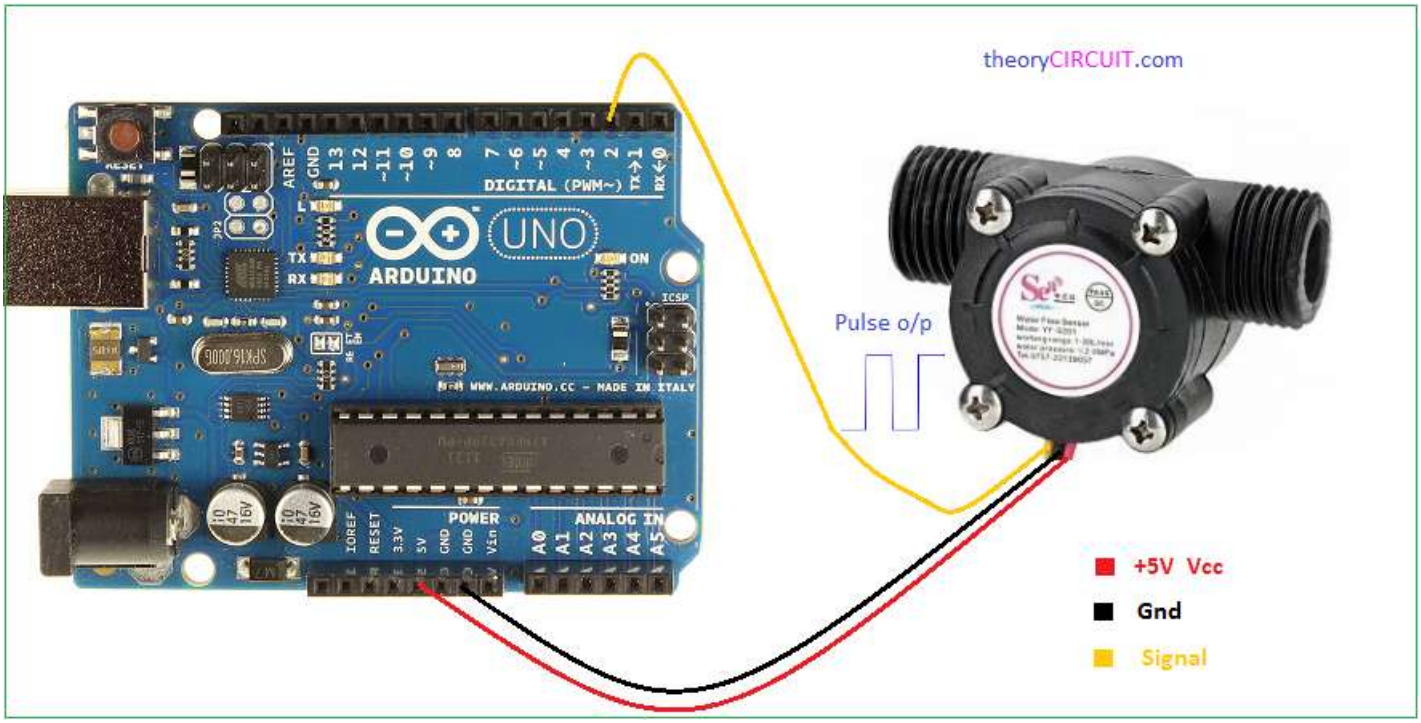


This illustration gives detailed working method of hall effect sensor based water flow sensor, a turbine wheel embed with magnet is placed on a closed plastic envelop and a Hall effect sensor placed, When the water flows through the pipeline, it makes the turbine wheel to rotate and hence the magnet flux interferes the hall sensor, the rate of interference is depends on the speed of water flow, so the hall effect sensor produce pulse signal output, this pulse output can be calculated as water volume.

YF-S201 water flow sensor

This water flow sensor has only three wires and it can be easily interfaced between any microcontroller and Arduino board. It requires only +5V Vcc and gives pulse output, the sensor needs to be tightly fitted between water pipeline.

Arduino Hookup



Connect the +5V wire to Arduino power pin 5V and Ground pin to Gnd then connect Signal pin to Digital pin D2, this sensor has control circuit hence there is no need for pull up resistor, some sensor requires pull up resistors refer datasheet of water flow sensor before concluding hookup.

Arduino Code for water flow meter

```
/*
Arduino Water flow meter
YF- S201 Hall Effect Water Flow Sensor
Water Flow Sensor output processed to read in litres/hour
*/
volatile int flow_frequency; // Measures flow sensor pulses
unsigned int l_hour; // Calculated litres/hour
unsigned char flowsensor = 2; // Sensor Input
unsigned long currentTime;
unsigned long cloopTime;


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

```
void setup()
{
   pinMode(flowsensor, INPUT);
   digitalWrite(flowsensor, HIGH); // Optional Internal Pull-Up
   Serial.begin(9600);
   attachInterrupt(0, flow, RISING); // Setup Interrupt

   sei(); // Enable interrupts
   currentTime = millis();
   cloopTime = currentTime;
}

void loop ()
{
   currentTime = millis();

   // Every second, calculate and print litres/hour
   if(currentTime >= (cloopTime + 1000))
   {
      cloopTime = currentTime; // Updates cloopTime

      // Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.
      l_hour = (flow_frequency * 60 / 7.5);
      // (Pulse frequency x 60 min) / 7.5Q = flowrate in L/hour

      flow_frequency = 0; // Reset Counter
      Serial.print(l_hour, DEC); // Print litres/hour
      Serial.println(" L/hour");
   }
}
```

Water Flow Sensor

Introduction

In this tutorial you will learn how to use one water flow sensor with an Arduino board.

The water flow sensor consists of a plastic valve body, a water rotor and a hall-effect sensor.
When the water flows through the rotor, rotor rolls and the speed of it changes with a different rate of flow. The hall-effect sensor outputs the corresponding pulse signal.

This type of sensor can be found on different diameters, water pressure (MPa) and flow rate (L/m) ranges. Make sure to select one that will cover your needs. The sensor that I have it has 20mm diameter, <1.75Mpa water pressure and ~30 L/m flow rate range.

In this tutorial we will use the serial monitor for printing the water flow rate in liters per hour and the total of liters flowed since starting.

So let's get started!

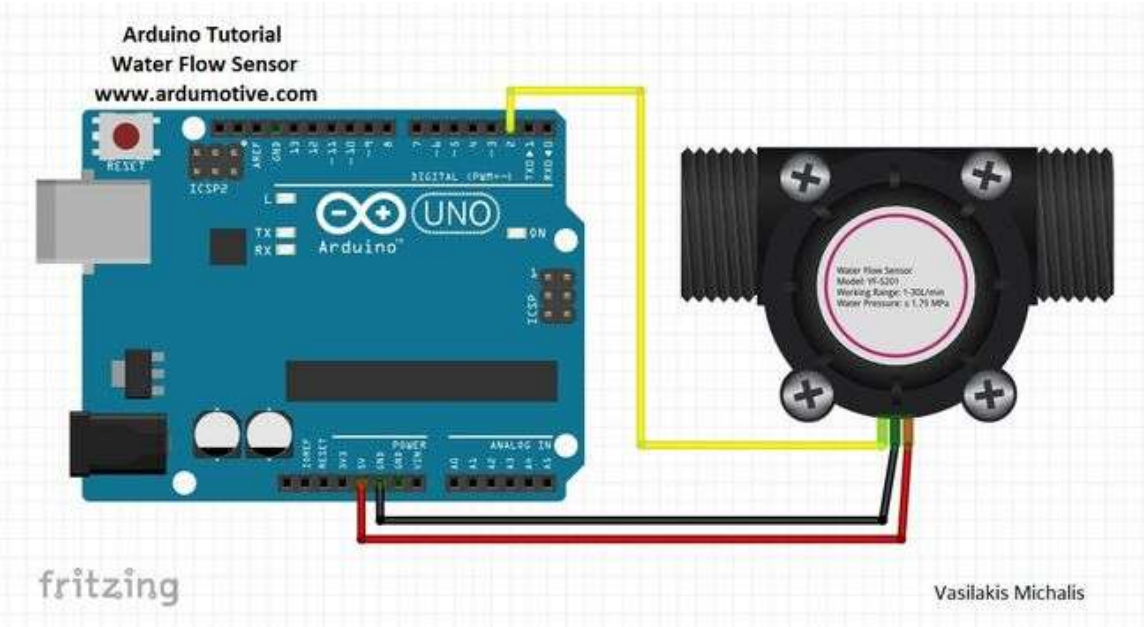
What you will need - Hardware

For this tutorial you will need:

- Arduino uno
- Water flow sensor

3 long breadboard cables

The Circuit



The connections are pretty easy, see the image above with the breadboard circuit schematic.

The code

```
/*
Liquid flow rate sensor -DIYhacking.com Arvind Sanjeev

Measure the liquid/water flow rate using this code.
Connect Vcc and Gnd of sensor to arduino, and the
signal line to arduino digital pin 2.

 */

byte statusLed    = 13;

byte sensorInterrupt = 0;  // 0 = digital pin 2
byte sensorPin       = 2;

// The hall-effect flow sensor outputs approximately 4.5 pulses per second per
// litre/minute of flow.
float calibrationFactor = 4.5;

volatile byte pulseCount;

float flowRate;
unsigned int flowMilliLitres;
unsigned long totalMilliLitres;

unsigned long oldTime;

void setup()
{

  // Initialize a serial connection for reporting values to the host
  Serial.begin(9600);
```

```
  // Set up the status LED line as an output
  pinMode(statusLed, OUTPUT);
  digitalWrite(statusLed, HIGH);  // We have an active-low LED attached

  pinMode(sensorPin, INPUT);
  digitalWrite(sensorPin, HIGH);

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

  // The Hall-effect sensor is connected to pin 2 which uses interrupt 0.
  // Configured to trigger on a FALLING state change (transition from HIGH state to LOW state)
  attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
}

/* Main program loop  */
void loop()
{

   if((millis() - oldTime) > 1000)    // Only process counters once per second
  {
    // Disable the interrupt while calculating flow rate and sending the value to
    // the host
    detachInterrupt(sensorInterrupt);

    // 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() - oldTime)) * pulseCount) / calibrationFactor;

    // Note the time this processing pass was executed. Note that because we've
    // disabled interrupts the millis() function won't actually be incrementing right
    // at this point, but it will still return the value it was set to just before
    // interrupts went away.
    oldTime = 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;

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

    unsigned int frac;

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

```
    Serial.print("\t");                 // Print tab space

    // Print the cumulative total of litres flowed since starting
    Serial.print("Output Liquid Quantity: ");
    Serial.print(totalMilliLitres);
    Serial.println("mL");
    Serial.print("\t");                 // Print tab space
          Serial.print(totalMilliLitres/1000);
          Serial.print("L");


    // Reset the pulse counter so we can start incrementing again
    pulseCount = 0;

    // Enable the interrupt again now that we've finished sending output
    attachInterrupt(sensorInterrupt, pulseCounter, FALLING);
  }
}

/*
Insterrupt Service Routine
 */
void pulseCounter()
{
  // Increment the pulse counter
  pulseCount++;
}
```

Serial Monitor - Testing

Press the connect button below to start the serial communication.
Connect your sensor with your water tap, or just blow on it.

Note: The back side of the sensor show with one arrow the correct flow side.

Water flow measurement with Arduino

The **flow sensor** used here works on the principle of “Hall Effect”.

According to which, a voltage difference is induced in a conductor transverse to the electric current and the magnetic field perpendicular to it. Here, Hall Effect is utilized in the flow meter using a small fan/propeller shaped rotor which is placed in the path of the liquid flowing.

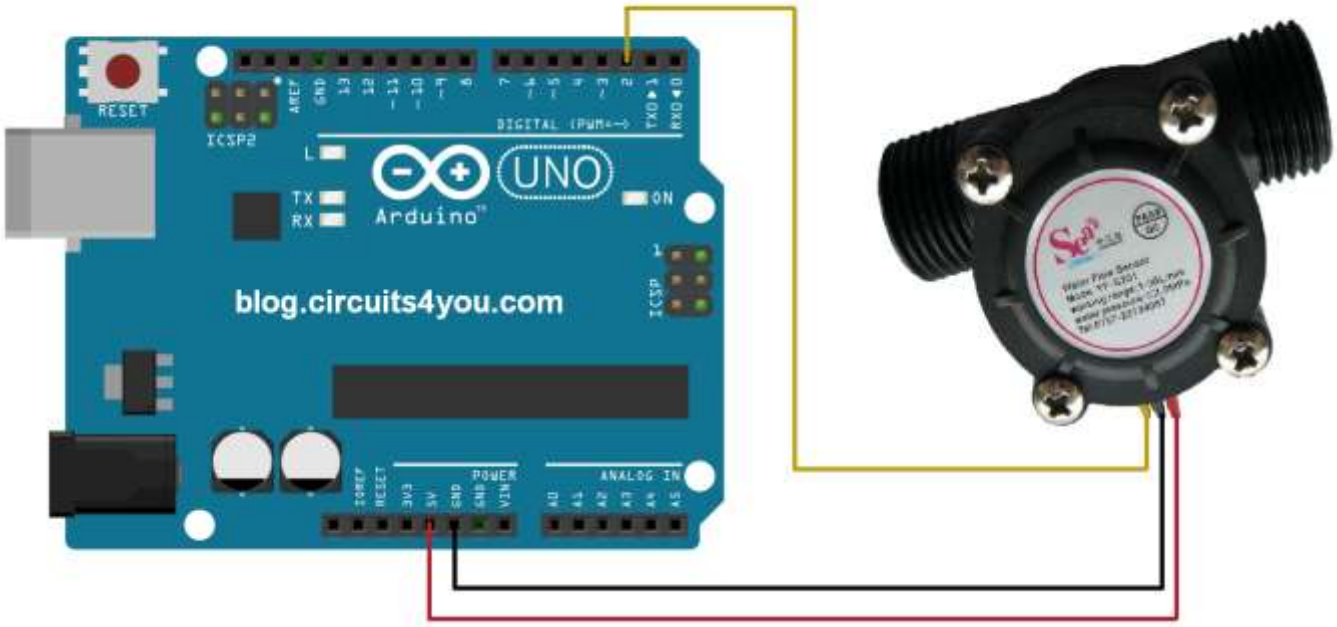
The liquid thus pushes against the fins of the rotor, causing it to rotate.

The shaft of the rotor is connected to a hall effect sensor. It is an arrangement of a current flowing coil and a magnet connected to the shaft of the rotor. Thus a voltage/pulse is induced as this rotor rotates. In this flow meter, for every liter of liquid passing through it per minute it outputs about 4.5 pulses. This is due to the changing magnetic field caused by the magnet attached to the rotor shaft. Arduino measure the number of pulses and then calculate the **flow rate in L/hr** using a simple conversion formula.

The sensor comes with three wires: red (5-24VDC power), black (ground) and yellow (Hall effect pulse output).

Water Flow Measurement Circuit

Use only interrupt pin 2 for sensor connections



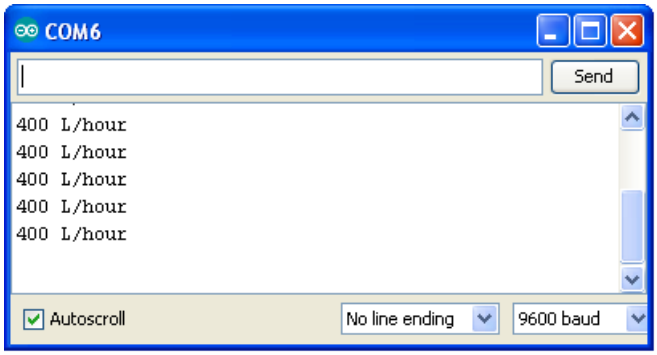
Water Flow Sensor Arduino Circuit

Water Flow Sensor Arduino Code

```
//Water Flow Measurement


volatile int FlowPulse; //measuring the rising edges of the signal

int Calc;

int flowsensor = 2;    //The pin-2 location of the sensor Always use this pin
as we are using interrupt 0


void setup() {

   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)

}


void 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

 Serial.print (Calc, DEC); //Prints the number calculated above

 Serial.println (" L/hour"); //Prints "L/hour"

}


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

}
```

Results of Water Flow Measurement

Open Serial monitor and allow flow of some water from sensor or blow air in flow sensor to rotate the rotor.



Arduino Flow Measurement Result