**Cyber Physical System**

**for Industrial Applications**

**Contents:-**

|  |  |
| --- | --- |
| **Sl no** | **Description** |
| 1 | Aim of the project |
| 2 | Problem Statement and Solution |
| 3 | Project Design Specification |
| 4 | Project Architecture |
| 5 | Flow chart Explanation |
| 6 | Wiring Diagram |
| 7 | Components working and functionality |
| |  |  | | --- | --- | | 8 | Execution and Results | | 9 | Bill of Materials | | 10 | Project Outcomes | | |

**1)Aim of the Project: -**Measurement of water flow rate and volume using flow sensor.

**2)Problem Statement and Solution: -**We have to develop and build a digital water meter which will do the task of measuring water flow rate and volume using flow sensor.

Not only water but we can measure the flow rate and volume of any other liquid also. This type of device are used in various industrial applications and in domestic purposes also. For example, we find it’s usage in water meters used at homes to measure water consumption to generate the bill accordingly. In industrial aspects, it is used in beverage industries as dispensers, which automatically fills the empty bottles with water or any other soft drinks.

**3)Project Design Specifications: -**Following are the key parameters of the digital water meter that are to be monitored continuously. To achieve this we have deployed an Arduino Uno, a microcontroller that will measure the parameters.

* ***Flow rate: -***The water flowing is to be measured by the flow sensor. But the question is what is the unit of measurement of the flow rate. So this flow sensor YF-S201 measures the water flow rate in L/min.
* ***Volume of the liquid: -***Once the flow rate is measured, the data is transferred to the microcontroller and then the volume is calculated in liters.
* ***Water Bill (If included could be a parameter but for our case is not mentioned): -***The data of volume consumed can be used to generate the bill also by giving the micro-controller the respective instructions.

**4)Project Architecture: -**

* Water Flow Sensor: This is the primary component responsible for measuring the flow rate of water. There are various types of flow sensors available, such as turbine, paddlewheel, ultrasonic, or electromagnetic sensors. The selection depends on factors like accuracy requirements, flow rate range, and the properties of the fluid being measured.
* Microcontroller or Microprocessor: The flow sensor's output needs to be processed to derive meaningful flow rate data. A microcontroller or microprocessor is used for this purpose. It reads the sensor output, performs necessary calculations, and interfaces with other system components.
* Signal Conditioning Circuitry: Depending on the type of flow sensor used, the signal may need conditioning to improve accuracy and reliability. For example, amplification, filtering, or voltage level conversion might be necessary to ensure compatibility with the microcontroller.
* Data Processing and Storage: The microcontroller processes the sensor data to calculate flow rates, total volume, and other relevant parameters. It may store this data temporarily in onboard memory or transmit it to an external device for further processing and storage.
* User Interface: A user interface allows users to interact with the system, view real-time data, configure settings, and retrieve historical data if applicable. This interface can be implemented using a display screen, LEDs, buttons, or a graphical user interface (GUI) if the system is connected to a computer or mobile device.
* Power Supply: The system requires a stable power supply to operate reliably. This can be achieved using batteries, AC power adapters, or power over Ethernet (PoE) depending on the application's power requirements and environmental constraints.

**5)Flow chart Explanation: -**

**Signal Conditioning**

**Arduino**

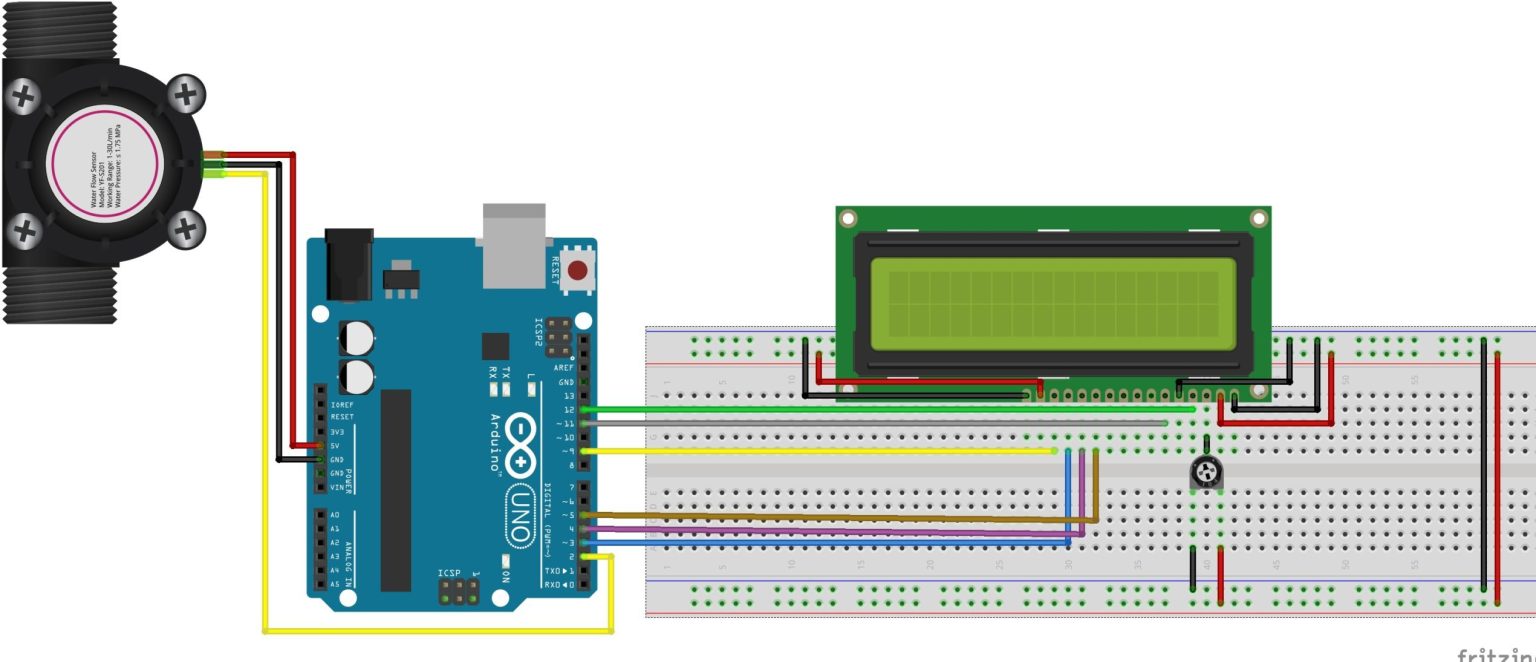
**Water** Flow

**Flow** **Sensor**

**Display**

**Data Processing & Calculation**

**6)Wiring diagram: -**

****

**7)Components working and functionality: -**

* ***Display unit: -***A 16\*2 LCD (liquid crystalline display is used which willdisplay us the information like flow rate and the volume.
* ***Flow sensor: -***The sensor is used to measure the liquid flow rate. This sensor sits in line with the water line and contains a pinwheel sensor to measure how much liquid has moved through it. There’s an integrated magnetic hall effect sensor that outputs an electrical pulse with every revolution. The hall effect sensor is sealed from the water pipe and allows the sensor to stay safe and dry. The sensor comes with three wires: red (5-24VDC power), black (ground), and yellow (Hall effect pulse output). By counting the pulses from the output of the sensor, we can easily calculate the water flo**w rate**. The water flow sensor for flow rate and volume measurement using Arduino works on the principle of the Hall effect. According to the Hall effect, a voltage difference is induced in a conductor transverse to the electric current and the magnetic field perpendicular to it. Here, the 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 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, its outputs about 4.5 pulses. This is due to the changing magnetic field caused by the magnet attached to the rotor shaft. We measure the number of pulses using an Arduino and then calculate the flow rate in liters per hour (L/Hr) and total volume in liters.
* ***Computing and controlling unit: -***The controlling component is an Arduino Uno board. It is based on a ATmega-328p MCU. It has 14 digital input-output pins or I/O pins and 6 analog pins I/O pins. We in this project used only digital I/O pins. The ATmega-328p is a programmable chip which is given instructions using code written in embedded C or in Arduino IDE.

**8)Execution and results:-**

***Code: -*** #include <LiquidCrystal\_I2C.h>

int flowPin = 2; //This is the input pin on the Arduino

double flowRate; //This is the value we intend to calculate.

volatile int count;

LiquidCrystal\_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16 column and 2 rows

void setup()

{

pinMode(flowPin, INPUT); //Sets the pin as an input

attachInterrupt(0, Flow, RISING);

Serial.begin(9600);

lcd.init(); // initialize the lcd

lcd.backlight();

}

void loop()

{

count = 0; // Reset the counter so we start counting from 0 again

interrupts(); //Enables interrupts on the Arduino

delay (1000); //Wait 1 second

noInterrupts();

flowRate = (count \* 2.25); //Take counted pulses in the last second and multiply by 2.25mL

flowRate = flowRate \* 60; //Convert seconds to minutes, giving you mL / Minute

flowRate = flowRate / 1000; //Convert mL to Liters, giving you Liters / Minute

flowMilliLitres = (flowRate / 60) \* 1000;

totalMilliLitres += flowMilliLitres;

Serial.println(flowRate);

lcd.backlight();

lcd.setCursor(0,0);

lcd.print("VOL.:");

lcd.setCursor(1,0);

lcd.print(flowRate);

lcd.setCursor(2,0);

lcd.print("L/M");

lcd.setCursor(0,1);

lcd.print("TOTAL:");

lcd.setCursor(1,1);

lcd.print(totalMilliLitres/1000);

lcd.setCursor(2,1);

lcd.print("L");

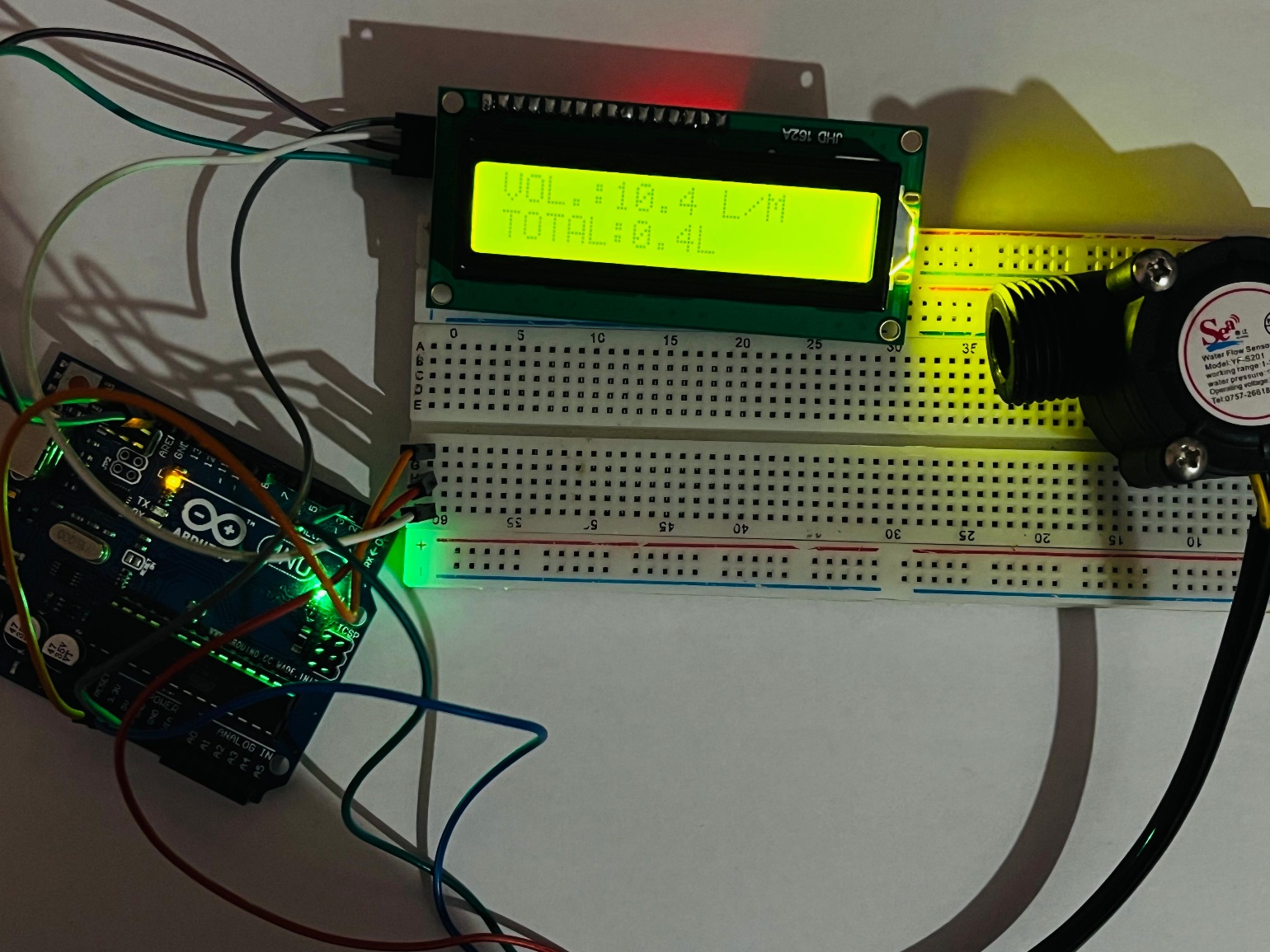
}

void Flow()

{

count++; //Every time this function is called, increment "count" by 1 is done}

***Circuit: -***



**9)Bill of Materials:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Instrument** | **Model** | **Quantity** | **Price** |
| 1 | Flow sensor | YFS201  Hall effect | 1 | Rs 210 |
| 2 | 16\*2 LCD | JHD162a | 1 | RS 130 |
| 3 | Arduino (MCU) | UNO | 1 | Rs 580 |
| 4 | Breadboard | 380 tie-points | 1 | Rs 350 |
| 5 | Jumper wires | Male to female | A whole set | Rs 50 |

**10)Project Outcomes:-**

* We implemented the project which highlights industrial aspects of CPS.
* We designed the product using components of CPS.
* Our calculations and understanding were right in accordance with the output.
* What we have done here is the data is to be monitored by physically reaching out to the meter. But it can be remotely observed using the IoT.