THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY – VELLORE

IOT PHASE 4 PROJECT – SMART WATER MANAGEMNET

SMART WATER METER USING ARDUINO

-SUBMITTED BY

PUGAZHMANI P

SANJAY KRISHNAN N V

SERALATHAN N

SARAVANA PRABU G

SURIYA R

SMART WATER METER

Smart water meters are devices that measure and communicate water usage from consumer to provider to facilitate water management and proper billing.

WATER FLOW SENSOR

Water flow sensors are installed at the water source or pipes to measure the rate of flow of water and calculate the amount of water flowed through the pipe. Rate of flow of water is measured as liters per hour or cubic meters.

WORKING OF WATER FLOW SENSOR

The main working principle behind the working of this sensor is the Hall effect. According to this principle, in this sensor, a voltage difference is induced in the conductor due to the rotation of the rotor. This induced voltage difference is transverse to the electric current.

When the moving fan is rotated due to the flow of water, it rotates the rotor which induces the voltage. This induced voltage is measured by the hall effect sensor and displayed on the LCD display.

The water flow sensor can be used with hot waters, cold waters, warm waters, clean water, and dirty water also. These sensors are available in different diameters, with different flow rate ranges.

Creating a smart water meter using an Arduino and a water flow sensor is a great project to monitor water usage and promote water conservation. In this project, we'll outline the components you'll need and provide a general overview of how to build a basic smart water meter. Keep in mind that this is a simplified example, and you can expand upon it to suit your specific needs.

Components Required:

Arduino board (e.g., Arduino Uno)

Water flow sensor (e.g., YF-S201)

LCD display (optional, for real-time readings)

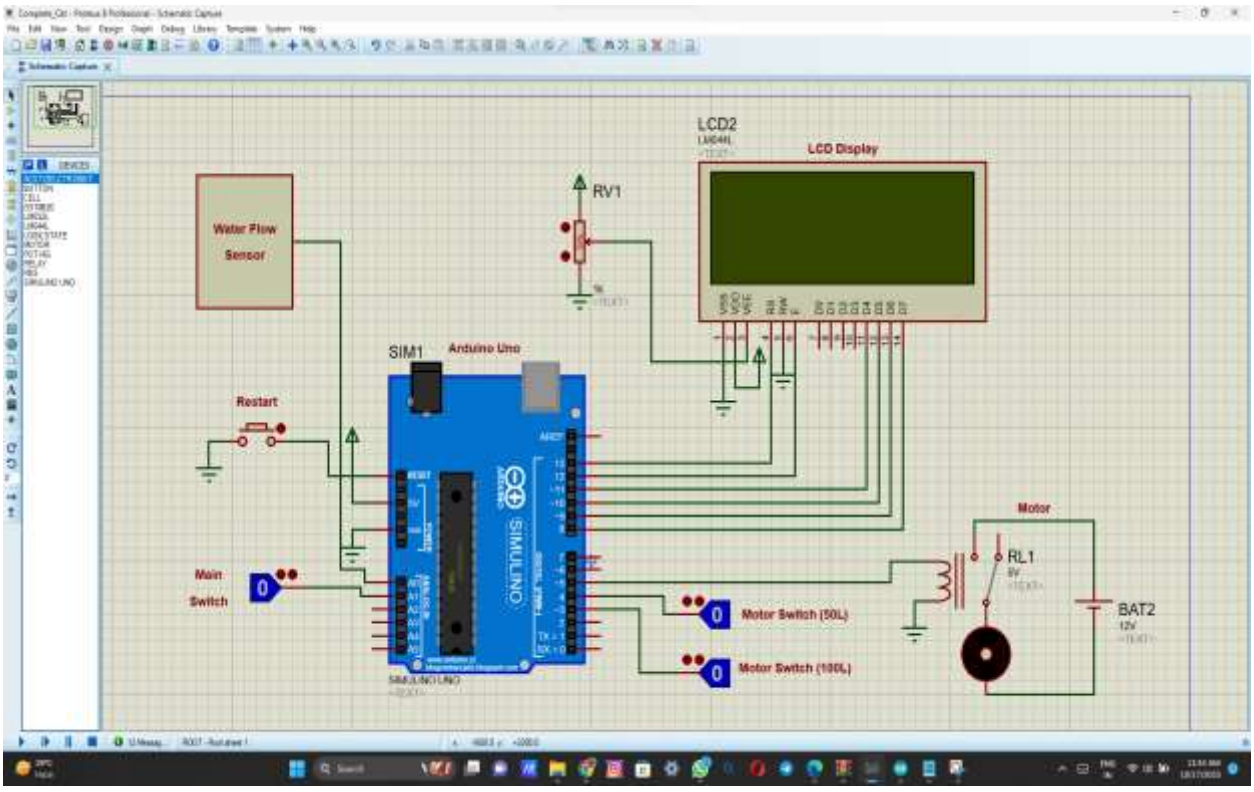
PROTEUS SIMULATOR

Assemble the Hardware:

a. Using PROTEUS simulator .

b. Connect the water flow sensor to the Arduino using wires. The sensor typically has three pins: VCC (power), GND (ground), and OUT (data). Connect VCC to 5V, GND to GND, and OUT to a digital pin on the Arduino (e.g., D2).

c. If you're using an LCD display, connect it to the Arduino. You'll need to connect power, ground, and data pins to the appropriate Arduino pins.



Platform Used : PROTUES 8 PROFESSIONAL

Proteus is electronic design automation software used by engineers and hobbyists. It enables circuit design and simulation, including microcontroller-based systems. Users can design PCB layouts, simulate circuit behavior, and utilize virtual instruments for testing.

It supports custom component creation and advanced analysis. Proteus is popular in education for electronics instruction. The Professional version offers 3D visualization for PCB designs. It runs on Windows and is a versatile tool for electronic design and testing. For the latest features and updates, check the Labcenter Electronics website.

Proteus is compatible with Windows operating systems and is known for its versatility and robust simulation capabilities.

Upload the Arduino Code:

Write and upload the Arduino code to read data from the water flow sensor. The code should include instructions to calculate and display water flow rate and total water usage. If you're using an LCD display, make sure the code displays the data in a user-friendly format.

Code:

```
#include <FlowMeter.h>

#include "Wire.h"

#include "OakOLED.h"

OakOLED oled;

FlowMeter Meter1 = FlowMeter(2);

FlowMeter Meter2 = FlowMeter(3);


void Meter1ISR() {

   Meter1.count();

}

void Meter2ISR() {

Meter2.count();

}
```

```
void setup() {
   oled.begin();
   Serial.begin(9600);
   attachInterrupt(INT0, Meter1ISR, RISING);
   attachInterrupt(INT1, Meter2ISR, RISING);
   Meter1.reset();
   Meter2.reset();
}

void loop() {
data();
}

void data(){
Meter1.tick(3000);
 Meter2.tick(3000);
Serial.print(String(Meter1.getCurrentFlowrate()));
Serial.print("l/m");
Serial.print(":");
Serial.print( String(Meter1.getTotalVolume()));
Serial.print("L total");
Serial.print(":");
Serial.print(String(Meter2.getCurrentFlowrate()));
Serial.print("l/m");
Serial.print(":");
Serial.print( String(Meter2.getTotalVolume()));
Serial.print("L total:");
```

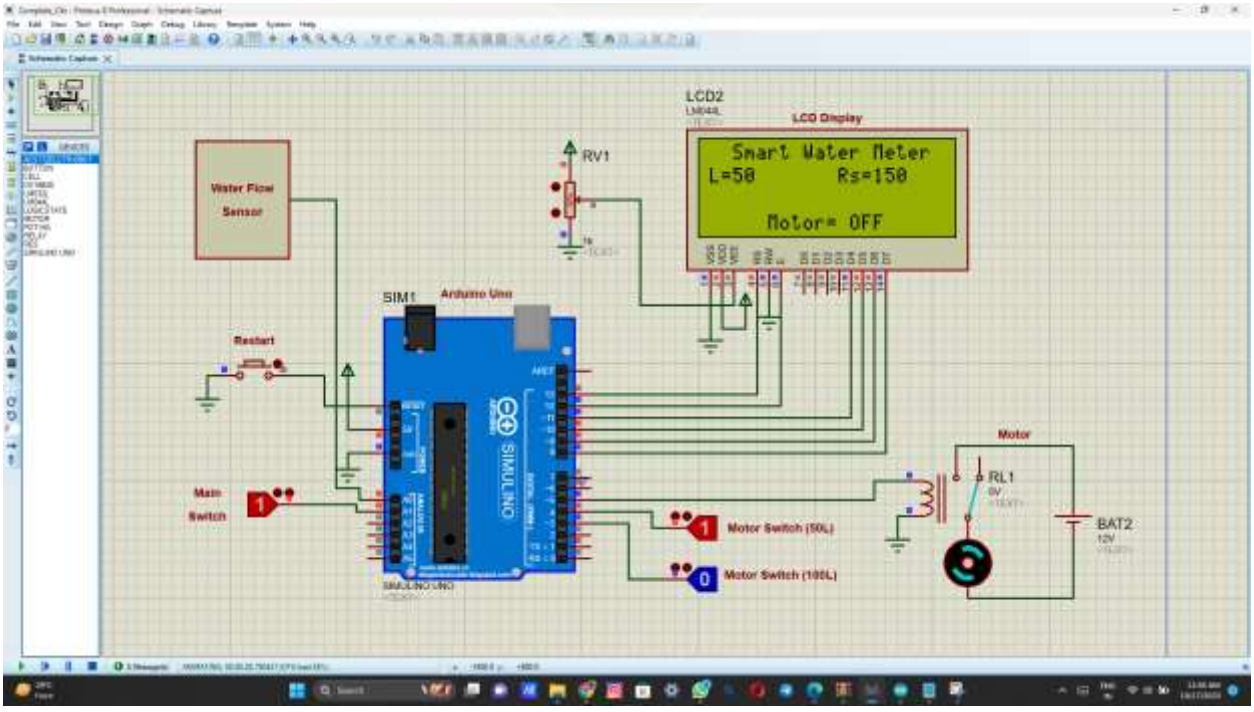
```
Serial.println(" ");


oled.clearDisplay();
oled.setTextSize(1);
oled.setTextColor(1);
oled.setCursor(0, 0);
oled.println("Flow rate");
oled.setTextSize(2);
oled.setCursor(10, 12);
oled.println(Meter2.getCurrentFlowrate());
oled.setTextSize(2);
oled.setCursor(80, 12);
oled.println("l/m");


oled.setTextSize(1);
oled.setTextColor(1);
oled.setCursor(3, 29);
oled.println("Total Volume");
oled.setTextSize(2);
oled.setTextColor(1);
oled.setCursor(10,40 );
oled.println(Meter2.getTotalVolume());
oled.setCursor(80, 40);
oled.println("lit");
oled.display();
delay (3000);
}
```

Calibrate the Flow Sensor:

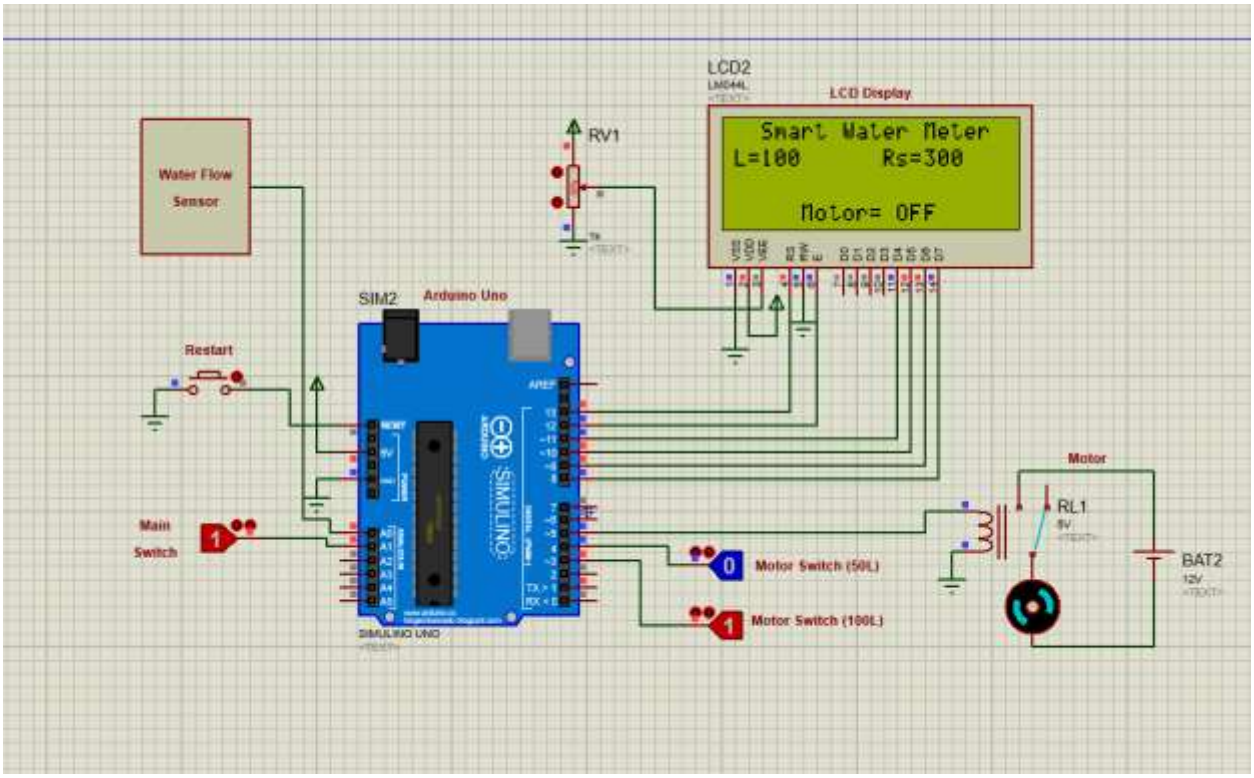
To obtain accurate readings, you may need to calibrate the water flow sensor. This involves adjusting the flow rate conversion factor in the code based on your sensor's specifications.

Monitor and Control Water Usage:

Once the system is set up, it will continuously monitor water flow and display real-time data. You can expand the project to include features like sending data to a cloud platform for remote monitoring or even controlling a solenoid valve to regulate water flow based on specific conditions.



Above, the 50L motor switch is ON so the water is had flown till the sensor reaches 50L ,then as per our program after that the motor had switched OFF and the price for the same been calculated Rs.150.



In second case, the 100L motor switch is ON and the respective amount for the water flow has been calculated and displayed.When it reaches 100L then the Motor has turned OFF automatically.

THANK YOU