THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY – VELLORE

IOT PHASE 3 PROJECT – SMART WATER MANAGEMNET

SMART WATER METER

-SUBMITTED BY

PUGAZHMANI P

SANJAY KRISHNAN N V

SERALATHAN N

SARAVANA PRABU G

SURIYA R

Introduction:

The Smart Water Meter Project is an ambitious initiative aimed at revolutionizing the way we measure, monitor, and conserve our water resources. This project leverages cutting-edge technology and data-driven insights to ensure efficient water usage, reduce wastage, and enhance the overall sustainability of our communities.

Traditional water meters, with their limitations in accuracy and data collection, have long been due for an upgrade. The Smart Water Meter Project seeks to replace these outdated systems with state-of-the-art smart water meters, capable of providing real-time, detailed information on water consumption. By deploying these innovative meters, we can proactively detect leaks, optimize water distribution, and empower consumers to make informed decisions about their water usage.

DEPLOYMENT:

Components Required:

Arduino board (e.g., Arduino Uno)

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

LCD display (optional, for real-time readings)

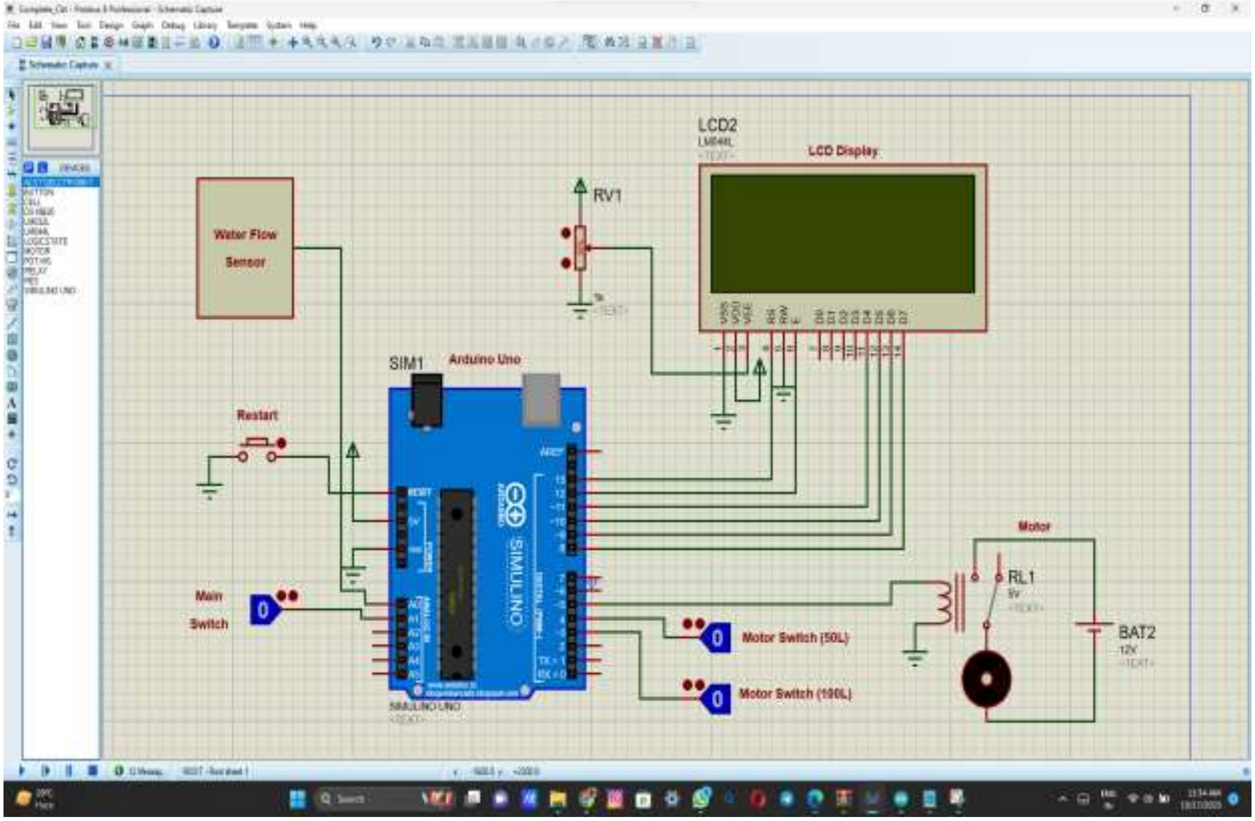
PROTEUS SIMULATOR

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.



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.

Script :

```
#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);
}
```

CONCLUSION:

In the implementation of a smart water meter project offers a myriad of benefits for both utility providers and consumers. By enabling remote monitoring and real-time data collection, smart water meters significantly enhance accuracy and efficiency in water management. The capacity to detect leaks, promote water conservation, and streamline billing_processes ensures cost savings and environmental sustainability.

THANK YOU