

ACM IOT TASK REPORT

SMART HOME ENERGY MANAGEMENT SYSTEM USING IOT

By,

Suyash Dabholkar

24BCE2189

1. Introduction

This report presents a real-time home energy monitoring and management system using IoT-based devices and sensors. The system enables users to track power consumption and remotely control household appliances through a central platform.

2. Hardware Requirements

1. **Microcontroller:** ESP8266/ESP32 (for network)
2. **Current Sensors:** ACS712 (for measuring current)
3. **Voltage Sensors:** ZMPT101B (for measuring voltage)
4. **Relay Modules:** To control appliance power state
5. **LCD Display/OLED:** To show real-time data
6. **Power Supply**
7. **Cables and PCB Board:**

3. Software Requirements

1. **Arduino IDE:** For coding
2. **Tinkercad:** For circuit simulation
3. **Blynk/Thingspeak/Node-RED:** For IoT-based remote monitoring
4. **Firebase/MQTT Protocol:** For real-time data communication

4. Implementation Steps

- Connect ACS712 and ZMPT101B sensors to ESP32.
- Use relays to control appliance power supply.
- Read sensor data (voltage & current).
- Calculate power consumption ($P = V * I$).
- Send data to cloud using MQTT/Firebase.
- Use Blynk or Thingspeak for real-time monitoring.
- Automate appliance control based on usage patterns.
- Implement scheduling and remote ON/OFF controls.
- Test using Tinkercad before real-world deployment.

5. Challenges and Considerations

- **Network Reliability:** Dependence on Wi-Fi may cause latency.
- **Power Sensor Accuracy:** Calibration required for precise readings.
- **Security:** Implement encryption for safe data transmission.
- **Scalability:** Future upgrades for multiple devices should be considered.

6. Sample Code To Program Microcontroller

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Initialize the LCD display with the corresponding pin connections  
const int currentPin = A0; // Analog input pin for current measurement  
const int voltagePin = A1; // Analog input pin for voltage measurement  
const int currentIndicatorPin = 8; // Digital output pin for current indicator LED  
const int voltageIndicatorPin = 9; // Digital output pin for voltage indicator LED  
const int powerIndicatorPin = 10; // Digital output pin for power indicator LED  
const float voltageMultiplier = 0.0048828125; // Calibration factor for voltage measurement  
const float currentMultiplier = 0.185; // Calibration factor for current
```

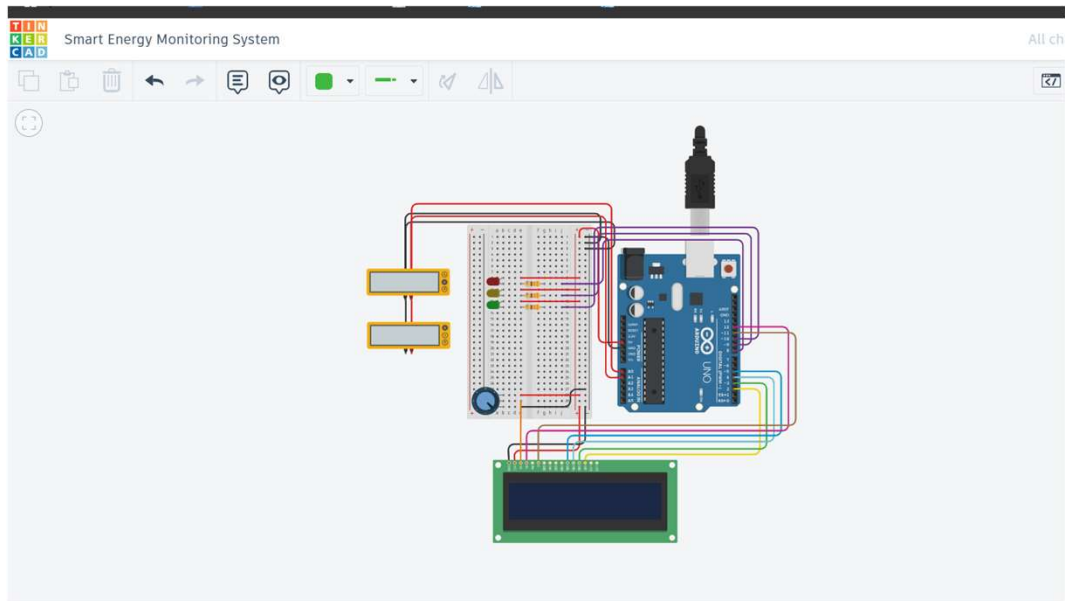
```
void setup() {  
    lcd.begin(16, 2); // Initialize the LCD display with 16 columns and 2 rows  
    pinMode(currentIndicatorPin, OUTPUT);  
    pinMode(voltageIndicatorPin, OUTPUT);  
    pinMode(powerIndicatorPin, OUTPUT);  
}  
void loop() {  
    // Read voltage and current values  
    float voltage = analogRead(voltagePin) * voltageMultiplier;  
    float current = analogRead(currentPin) * currentMultiplier;  
    float power = voltage * current;  
    // Update LCD display  
    lcd.clear(); lcd.setCursor(0, 0); lcd.print("Voltage: "); lcd.print(voltage); lcd.print(" V");  
    lcd.setCursor(0, 1); lcd.print("Current: "); lcd.print(current); lcd.print(" A");  
    // Update LED indicators based on thresholds  
    if (voltage > 4.5 && voltage < 5.5) {  
        digitalWrite(voltageIndicatorPin, HIGH);  
    } else {  
        digitalWrite(voltageIndicatorPin, LOW);  
    }  
    if (current > 0.1 && current < 0.3) {  
        digitalWrite(currentIndicatorPin, HIGH);  
    }  
}
```

```
else {  
  digitalWrite(currentIndicatorPin, LOW);  
}  
if (power > 0.3 && power < 1.5) {  
  digitalWrite(powerIndicatorPin, HIGH);  
}  
else {  
  digitalWrite(powerIndicatorPin, LOW);  
}  
delay(1000);  
}
```

7. Screenshots of Tinkercad

Parts Used in Tinkercad:

1. Arduino UNO
2. ACS712 Current Sensor
3. ZMPT101B Voltage Sensor
4. Relay Module
5. LCD Display with I2C Module
6. Wi-Fi Module (ESP8266-01)
7. LED Bulb
8. Power Supply (9V Battery) and Jumper Wires

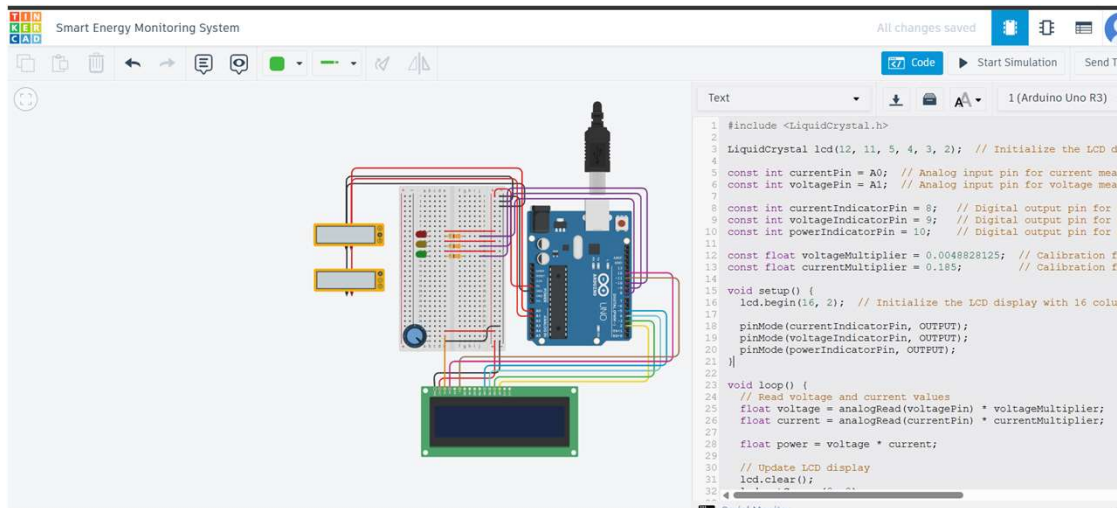


```

Text
1 (Arduino Uno R3)

32 lcd.setCursor(0, 0);
33 lcd.print("Voltage: ");
34 lcd.print(voltage);
35 lcd.print(" V");
36
37 lcd.setCursor(0, 1);
38 lcd.print("Current: ");
39 lcd.print(current);
40 lcd.print(" A");
41
42 // Update LED indicators based on thresholds
43 if (voltage > 4.5 && voltage < 5.5) {
44   digitalWrite(voltageIndicatorPin, HIGH);
45 } else {
46   digitalWrite(voltageIndicatorPin, LOW);
47 }
48
49 if (current > 0.1 && current < 0.3) {
50   digitalWrite(currentIndicatorPin, HIGH);
51 } else {
52   digitalWrite(currentIndicatorPin, LOW);
53 }
54
55 if (power > 0.3 && power < 1.5) {
56   digitalWrite(powerIndicatorPin, HIGH);
57 } else {
58   digitalWrite(powerIndicatorPin, LOW);
59 }
60
61 delay(1000);
62
Serial Monitor

```



Link of project :-

<https://www.tinkercad.com/things/gWPP02EKq8M-smart-energy-monitoring-system/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard%2Fdesigns%2Fcircuits>