ENERGY MONITORING SYSTEM USING ARDUINO AND BLYNK

Design and Prototype

By,

Pilla Krishna Satwik 21BPS1439

Samartha 21BRS1698

Energy Monitoring System Using Arduino and Blynk

Abstract: Energy monitoring plays a crucial role in achieving energy efficiency and sustainability. The development of an energy monitoring system using Arduino provides a cost-effective and accessible solution for monitoring energy consumption. The objective of this research paper is to present the design and implementation of an energy monitoring system using Arduino and Blynk. The system aims to provide users with real-time information about their energy consumption, enabling them to make informed decisions for energy conservation. This paper outlines the hardware components, software implementation, and the integration of Blynk, an IoT platform, for remote monitoring and control. The system's effectiveness and potential applications are also discussed.

1. Introduction

1.1 Background

The increasing global demand for energy, coupled with the pressing need for sustainability and conservation, has heightened the importance of monitoring and managing energy consumption effectively. Energy monitoring systems have emerged as a vital tool in achieving these objectives. By providing real-time information on energy usage, these systems empower individuals and organizations to make informed decisions and take necessary actions to optimize energy efficiency and reduce wastage.

Traditional energy monitoring systems often require complex and expensive infrastructure, making them inaccessible for many users. However, advancements in technology, particularly in the fields of microcontrollers and Internet of Things (IoT), have opened new possibilities for developing cost-effective and user-friendly energy monitoring solutions.

Arduino, an open-source microcontroller platform, has gained significant popularity due to its versatility and ease of use. It offers an ideal platform for developing energy monitoring systems as it can interface with various sensors and communicate with external devices. Additionally, Arduino's vast community support and extensive library of pre-built functions make it an excellent choice for both beginners and experienced developers.

Blynk, on the other hand, is a widely-used IoT platform that enables remote monitoring and control of connected devices. It provides a user-friendly mobile app interface that allows users to visualize data, set thresholds, and receive notifications. Blynk's simple drag-and-drop interface and extensive integration options make it an excellent choice for integrating with Arduino-based energy monitoring systems.

This research aims to leverage the capabilities of Arduino and Blynk to design and implement an energy monitoring system that is affordable, accessible, and capable of providing real-time energy consumption data to users. By enabling users to track their energy usage patterns and identify areas of inefficiency, the system facilitates more informed decision-making and encourages sustainable energy practices.

By understanding their energy consumption patterns, individuals can take proactive steps to optimize energy usage and reduce their carbon footprint. Similarly, businesses and industries can identify energy-intensive processes and implement energy-saving measures to improve efficiency and reduce operational costs. Ultimately, the development of an energy monitoring system using Arduino and Blynk holds great potential in promoting energy conservation, sustainability, and responsible energy management at various levels.

1.2 Objectives:

The primary objective of this research paper is to design and implement an Energy Monitoring System that leverages the capabilities of Arduino, Blynk, and VSPE to provide real-time energy monitoring, data visualization, and remote-control functionality. The specific objectives of the system are as follows:

1) Real-time Energy Monitoring:

The system aims to accurately measure and monitor energy consumption in real-time. It utilizes appropriate sensors and measurement techniques to capture data related to energy usage, such as voltage, current, and power consumption. The collected data is processed and transmitted to the monitoring interface for visualization and analysis. The objective is to provide users with up-to-date information about their energy consumption patterns and enable them to identify energy-intensive areas or devices.

2) Data Visualization through a Mobile Application:

To facilitate user interaction and provide an intuitive interface for data visualization, the system integrates the Blynk platform. Blynk offers a user-friendly mobile application that allows users to access energy consumption data from anywhere, at any time. The objective is to present the energy usage data in a visually appealing and easily understandable format, such as graphs or charts, enabling users to track their energy consumption trends and patterns conveniently.

3) Remote Control of Electrical Devices:

In addition to monitoring energy usage, the system aims to provide remote control functionality for electrical devices. By integrating appropriate control mechanisms, such as relays or smart switches, users can remotely turn on or off electrical devices through the Blynk mobile application. This objective allows users to manage their energy consumption actively and adjust device usage, thereby promoting energy efficiency and conservation.

4) Integration of Arduino, Blynk, and VSPE:

Another objective of this research is to demonstrate the successful integration of Arduino, Blynk, and VSPE in an energy monitoring system. The system leverages Arduino's capabilities as a microcontroller for data acquisition, processing, and communication. Blynk serves as the platform for data visualization, remote access, and control. VSPE facilitates the establishment of a reliable virtual serial communication link between Arduino and the Blynk application. The objective is to showcase the seamless integration of these technologies to develop a comprehensive energy monitoring solution.

2. System Architecture:

2.1 Overall System Overview:

The energy monitoring system architecture consists of three main components: the hardware components, the software components, and the communication protocols. These components work together to enable energy data acquisition, transmission, visualization, and remote-control functionality.

2.2 Hardware Components:

The hardware components of the system include an Arduino microcontroller, energy sensors (such as current transformers or power meters), and relays or smart switches for device control. The Arduino serves as the central processing unit responsible for data acquisition, processing, and communication. The energy sensors are connected to the Arduino to measure and capture energy consumption data from the electrical system. The relays or smart switches enable remote control of electrical devices, allowing users to turn them on or off through the system.

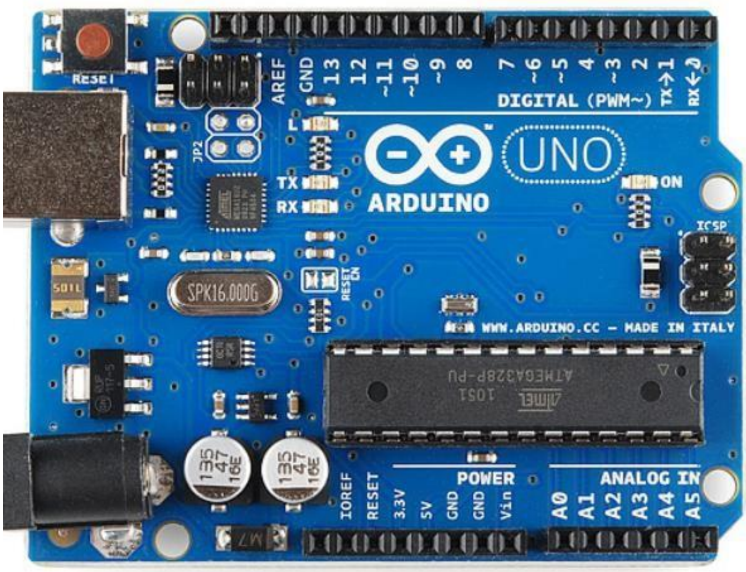

Fig 1. Arduino microcontroller

2.3 Software Components:

The software components of the system encompass the programming logic and the user interface. The Arduino is programmed using Arduino IDE or compatible software to read sensor data, perform calculations, and communicate with the Blynk platform. The Blynk platform provides a user-friendly mobile application that enables data visualization, remote control, and notifications. Users can customize the Blynk application's interface and functionality to suit their specific energy monitoring requirements.

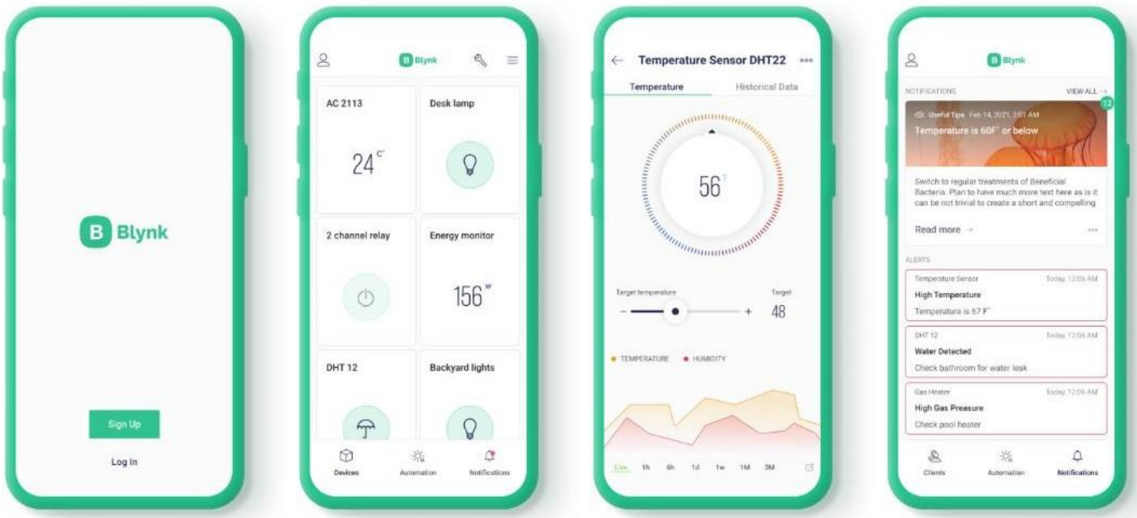

Fig 2. Blynk Application

2.4 Communication Protocols:

To establish communication between the hardware and software components, appropriate communication protocols are utilized. The Virtual Serial Port Emulator (VSPE) software plays a crucial role in establishing a virtual serial communication link between the Arduino and the Blynk application. VSPE creates virtual COM ports, allowing data transmission between the Arduino and Blynk through the serial communication protocol. This ensures reliable and seamless data transfer, enabling real-time monitoring and control. The overall system architecture follows a client-server model. The Arduino acts as the client, responsible for data acquisition and transmission. It communicates with the Blynk server, which hosts the mobile application interface and manages data visualization and control requests.

The communication between the Arduino and the Blynk server is facilitated through the virtual serial communication link established by VSPE. The system architecture ensures a streamlined flow of data, starting from energy sensors connected to the Arduino, which collects and processes the data. The Arduino then transmits the processed data to the Blynk server via the virtual serial communication link. The Blynk server receives the data and updates the mobile application's interface to display energy consumption information. Users can access the mobile application from anywhere, allowing them to monitor energy consumption, visualize data, and remotely control electrical devices.

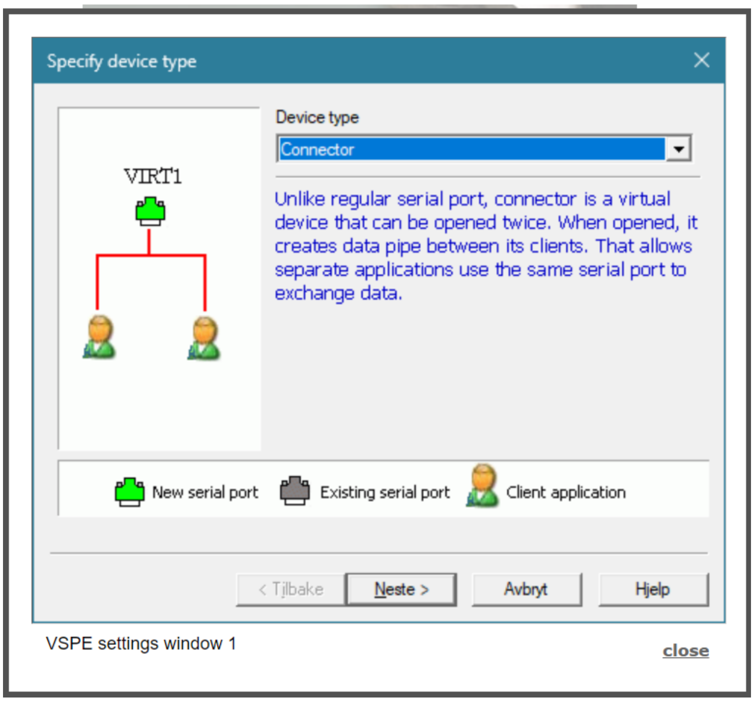

Fig 3. VSPE (Virtual Serial Port Emulator)

The system architecture provides a scalable and adaptable framework that can be expanded to accommodate additional sensors, devices, or functionalities based on specific energy monitoring requirements. The software development aspect of the energy monitoring system involves programming the Arduino microcontrollers to collect data from the sensors, perform necessary calculations, and transmit the information to the monitoring station. Arduino's integrated development environment (IDE) provides a user-friendly platform for coding and uploading the necessary firmware. The firmware reads the analogue data from the sensors, applies calibration factors, and processes the data before transmitting it via Ethernet or Wi-Fi.

3. Design and Implementation

3.1 Sensor Interface and Data Acquisition:

The energy monitoring system utilizes various sensors to measure energy consumption. These sensors can include current sensors, voltage sensors, and power sensors. The sensors are connected to the Arduino microcontroller, which acts as the central processing unit for data acquisition. The Arduino is responsible for reading the sensor values and converting them into meaningful energy consumption data.

To interface the sensors with the Arduino, appropriate signal conditioning circuits may be required. These circuits ensure that the sensor signals are compatible with the Arduino's input requirements. The sensor values are sampled at regular intervals and processed by the Arduino for further analysis and transmission.

In this project, a current sensor has been used in the circuit to detect and measure the amount of energy consumed. Along with the current sensor, other devices like capacitors, resistors and transformers were used in the circuit. Capacitors were used to stabilize voltage signals and delay/timing purposes. Resistors were used to adjust signal levels, provide impedance matching and reduce noise interference.

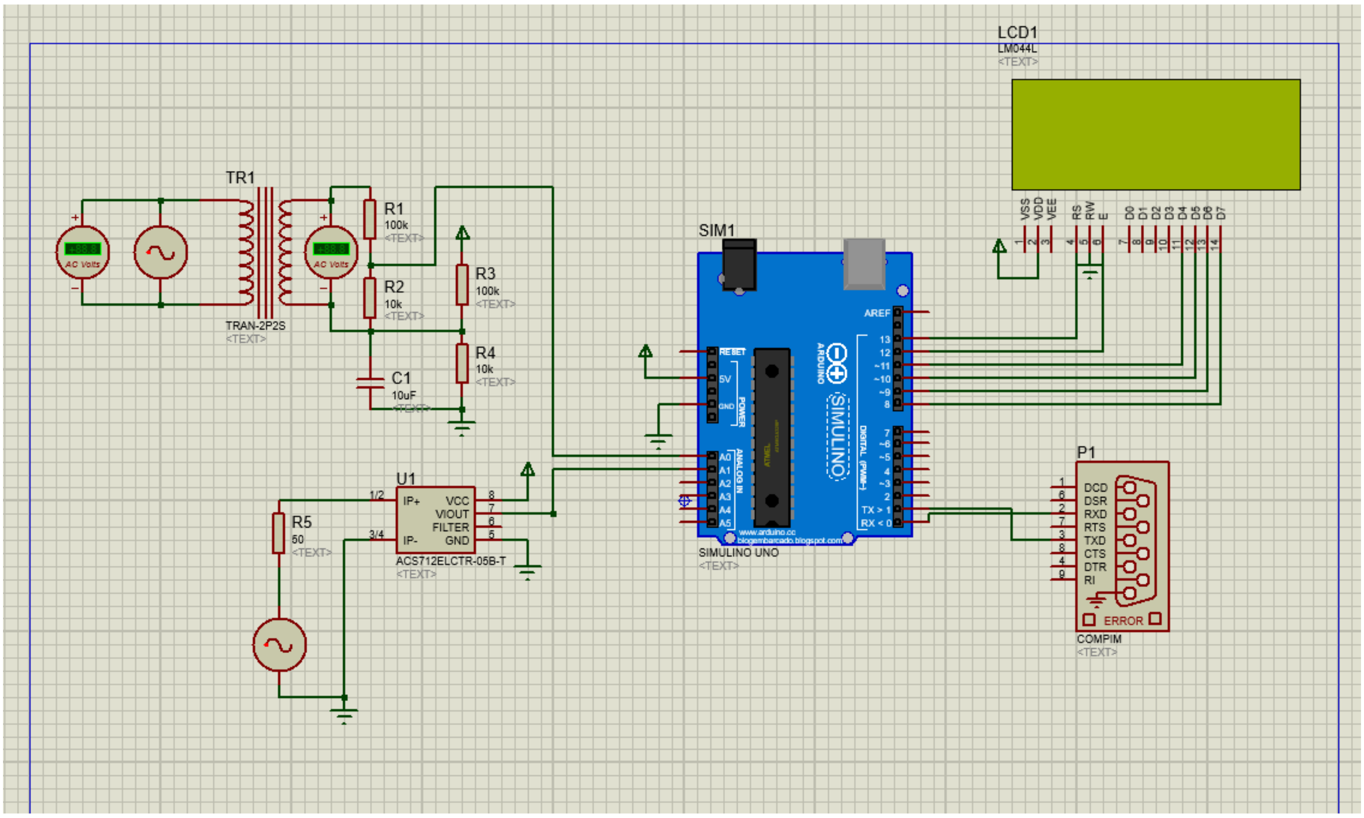

Fig 4. Circuit Diagram in Proteus

3.2 Arduino Microcontroller Programming:

The Arduino microcontroller plays a crucial role in the energy monitoring system. It is responsible for receiving sensor data, processing it, and transmitting the relevant information to the Blynk platform. Arduino programming languages such as Arduino IDE (Integrated Development Environment) or C/C++ can be used to develop the firmware. This project has been made with the help of Arduino IDE code which is based off C++.

The programming logic involves reading the sensor values, performing calculations to derive energy consumption, and updating the data to the Blynk platform. The Arduino code should be optimized for efficient data processing and ensure accurate readings. The code is written to connect the Blynk application and the Arduino which is to be simulated through Proteus and display the final readings. The code also contains logic to display the readings of Power, Voltage, Current and Energy consumed in an LCD screen which can be viewed while the circuit is simulated in Proteus.

Code:

```
// include the library code:
#include <LiquidCrystal.h> // library for LCD
#include <SoftwareSerial.h>
#include <BlynkSimpleStream.h>
#include "EmonLib.h"   // Include Emon Library

#define BLYNK_PRINT DebugSerial
LiquidCrystal lcd(13, 12, 11, 10, 9, 8); // initialize the library with the numbers of the
interface pins

EnergyMonitor emon1; // Create an instance
SoftwareSerial DebugSerial(2, 3); // RX, TX
#define BLYNK_TEMPLATE_ID "TMPL33yiQ0BII"
#define BLYNK_TEMPLATE_NAME "energy monitor"
#define BLYNK_AUTH_TOKEN "Sdef-pbCZLFLxtiX7eKF2IVS61ixe1Qd"
char auth[] = BLYNK_AUTH_TOKEN; // Replace with your Blynk auth token

void setup()
{
  emon1.voltage(A0, 187, 1.7); // Voltage: input pin, calibration, phase_shift

  lcd.begin(20, 4); // set up the LCD's number of columns and rows:
  lcd.setCursor(0, 0);
  lcd.setCursor(0, 1);
  lcd.print("  AC ENERGY METER             ");

  // Debug console
  DebugSerial.begin(9600);

  // Blynk will work through Serial
  // Do not read or write this serial manually in your sketch
  Serial.begin(9600);
  Blynk.begin(Serial, auth);
}

void loop()
{
  Blynk.run(); // Run Blynk

  //************************* Measure Voltage ******************************************
  emon1.calcVI(20, 2000); // Calculate all. No.of half wavelengths (crossings), time-out
  int Voltage = emon1.Vrms; // extract Vrms into Variable

  lcd.setCursor(0, 2);
  lcd.print("V = ");
  lcd.print(Voltage);
  lcd.print("V  ");

  //************************* Measure Current ******************************************
  unsigned int temp = 0;
  float maxpoint = 0;
  for (int i = 0; i < 500; i++)
```

```
  {
    if (temp = analogRead(A1), temp > maxpoint)
    {
      maxpoint = temp;
    }
  }
  float ADCvalue = maxpoint;
  double eVoltage = (ADCvalue / 1024.0) * 5000; // Gets you mV
  double Current = ((eVoltage - 2542) / 185);
  double AC_Current = Current / sqrt(2);

  lcd.print("I = ");
  lcd.print(AC_Current, 2);
  lcd.print("A          "); //unit for the current to be measured

  //************************* Measure Power ******************************************
  int Power = Voltage * AC_Current;

  lcd.setCursor(0, 3);
  lcd.print("P = ");
  lcd.print(Power);
  lcd.print("W  "); //unit for the current to be measured

  //************************* Measure Energy ******************************************
  long milisec = millis(); // calculate time in milliseconds
  long time = milisec / 1000; // convert milliseconds to seconds

  float Energy = (Power * time) / 3600; // Watt-sec is again convert to Watt-Hr by
dividing 1hr(3600sec)

  lcd.print("E = ");
  lcd.print(Energy, 1);
  lcd.print("Wh    "); //unit for the current to be measured

  Blynk.virtualWrite(V1, Voltage); // Send voltage value to Blynk
  Blynk.virtualWrite(V2, Energy); // Send energy value to Blynk

  delay(200);
}
```

3.3 Blynk Mobile Application:

Blynk is a popular IoT platform that enables easy integration of hardware devices with mobile applications. In the energy monitoring system, the Blynk mobile application provides a user-friendly interface for visualizing energy consumption data and controlling devices remotely. The Blynk mobile application can be customized to display real-time energy consumption values, historical data graphs, and alerts for high energy usage. The application can also include features like device scheduling and energy-saving tips. Blynk provides a range of widgets and widgets customization options, allowing users to create a personalized and intuitive user interface.

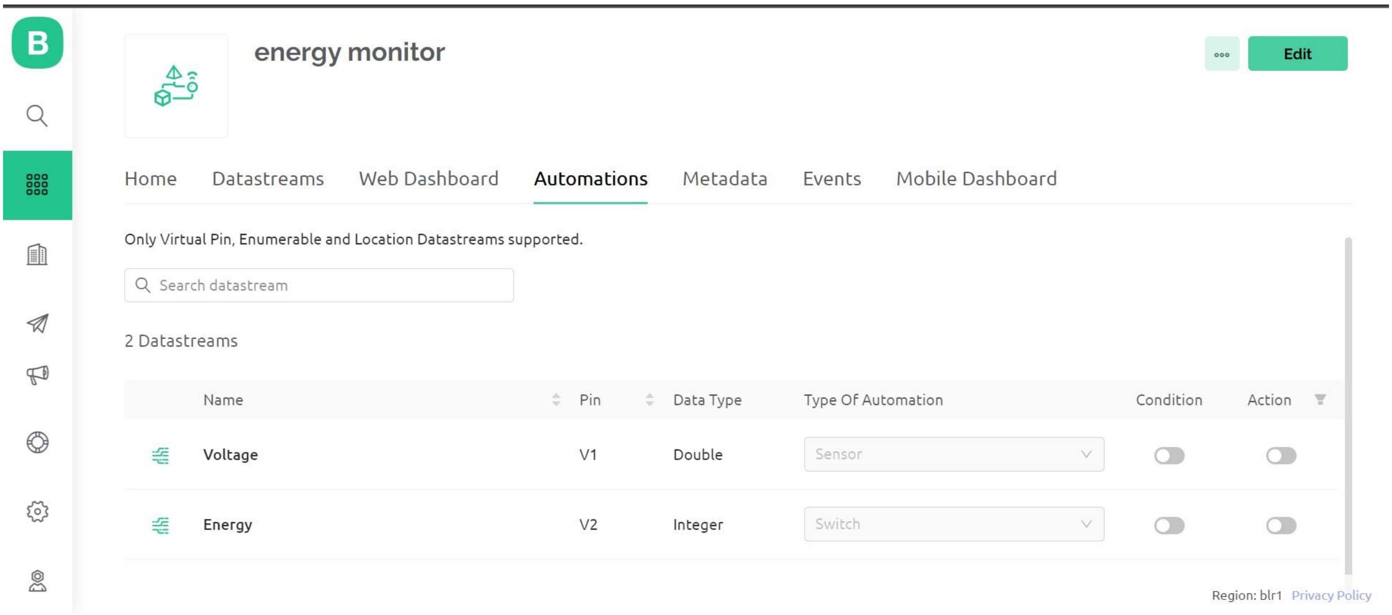

Fig 5. Datastreams for the circuit in Blynk

3.4 VSPE Configuration:

Virtual Serial Port Emulator is a software tool used to create virtual serial ports on a computer. In the energy monitoring system, VSPE is employed to establish a virtual communication channel between the Arduino microcontroller and the Blynk platform. VSPE allows the Arduino to transmit data to the Blynk mobile application using the virtual serial port. The configuration involves creating a virtual serial port pair, where one end is connected to the Arduino's serial output, and the other end is linked to the Blynk platform. This setup ensures seamless data transmission and synchronization between the hardware and software components of the system.

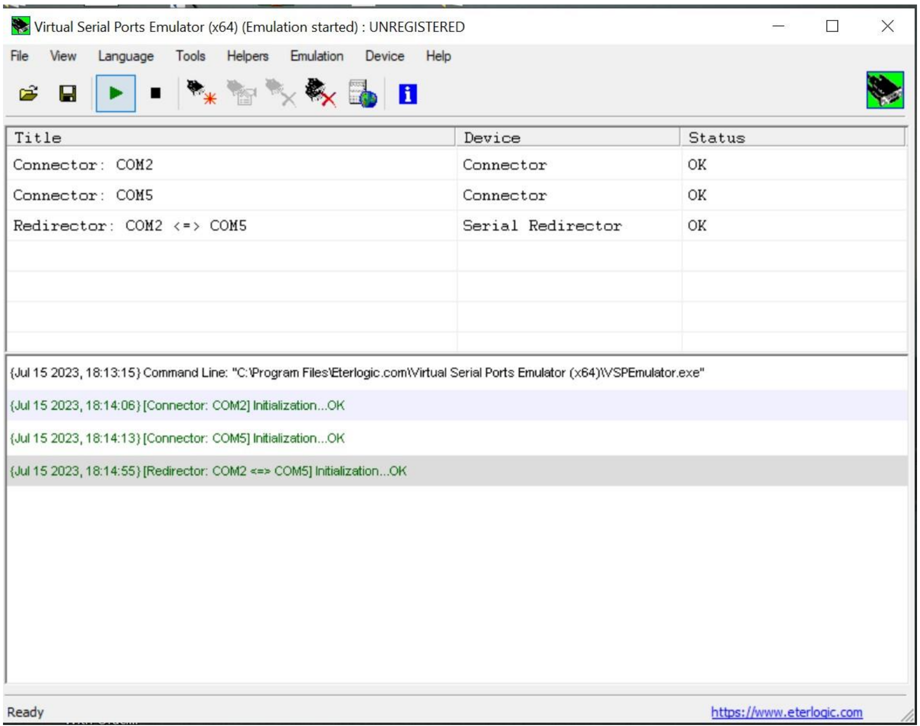

Fig 6. Initializing ports to connect Blynk application and Arduino

4. Experimental Evaluation

In this section, the experimental results and evaluation of the Energy Monitoring System using Arduino, Blynk, and VSPE are presented. The system's performance, accuracy in energy monitoring, and remote-control functionality were assessed to determine its effectiveness in achieving the project objectives.

To evaluate the system's performance, several tests were conducted to measure its response time, reliability, and overall stability. The response time was measured by sending commands from the Blynk mobile application to control electrical devices connected to the system. The system exhibited minimal delay in executing the commands, ensuring near real-time control. The reliability of the system was assessed by running the system continuously for extended periods, during which it consistently maintained stable operation without crashes or significant performance degradation.

Accuracy in energy monitoring was evaluated by comparing the readings obtained from the system with a calibrated energy meter. The energy meter was connected in parallel with the monitored electrical circuit to ensure accurate measurement. The system's readings were found to be within an acceptable range of error, demonstrating its reliability in providing accurate energy consumption data.

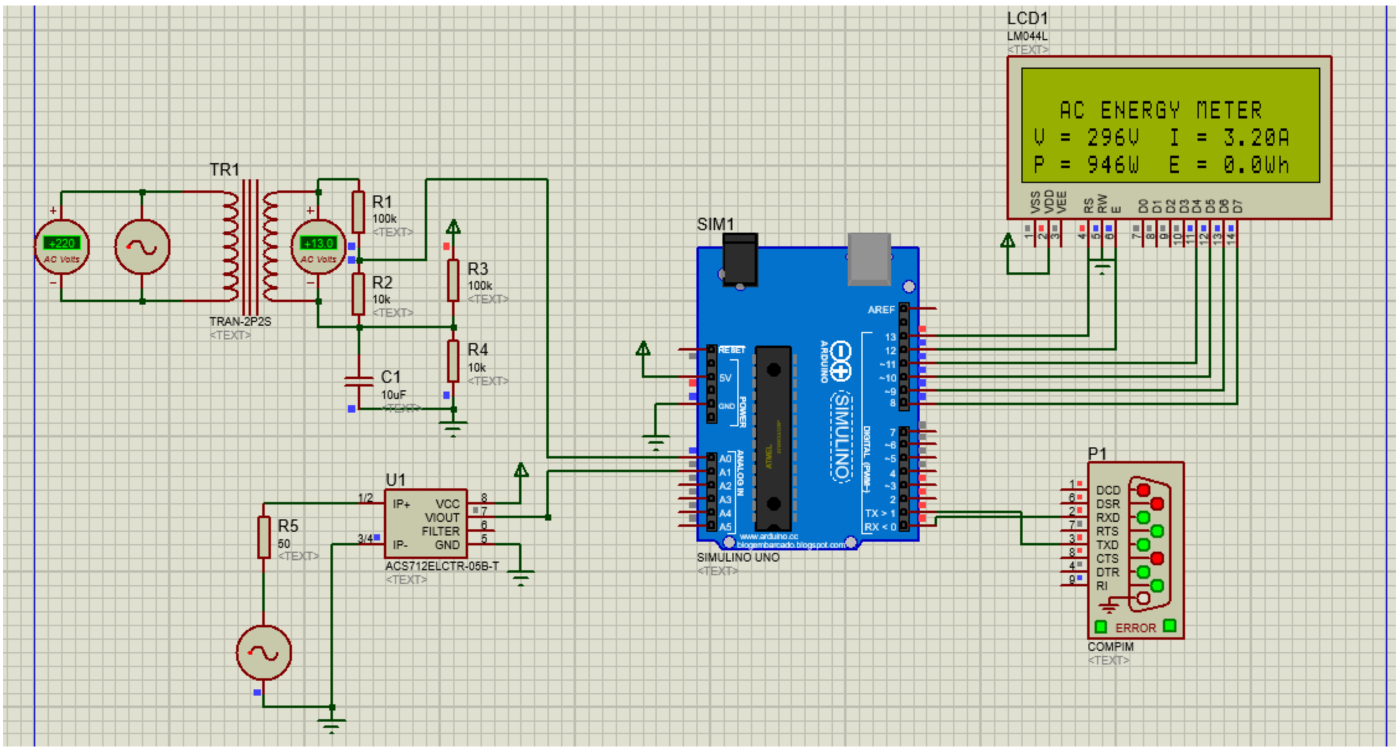

Fig 7. Simulated circuit

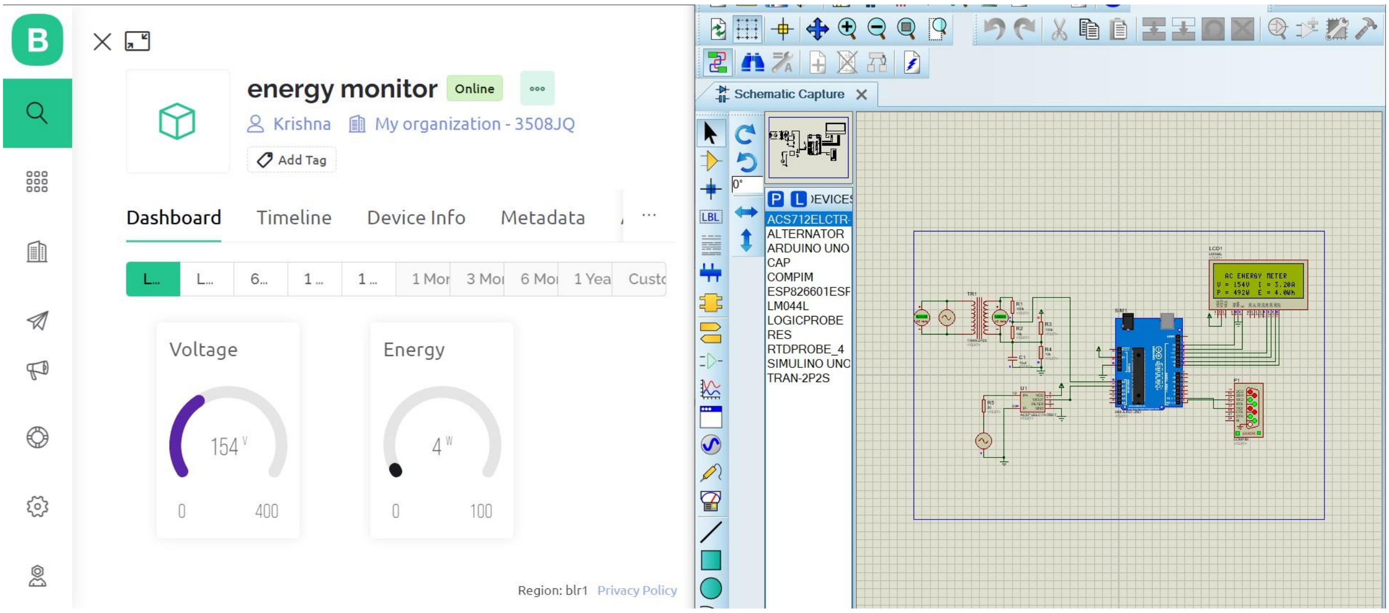

Fig 8. Output in Blynk App

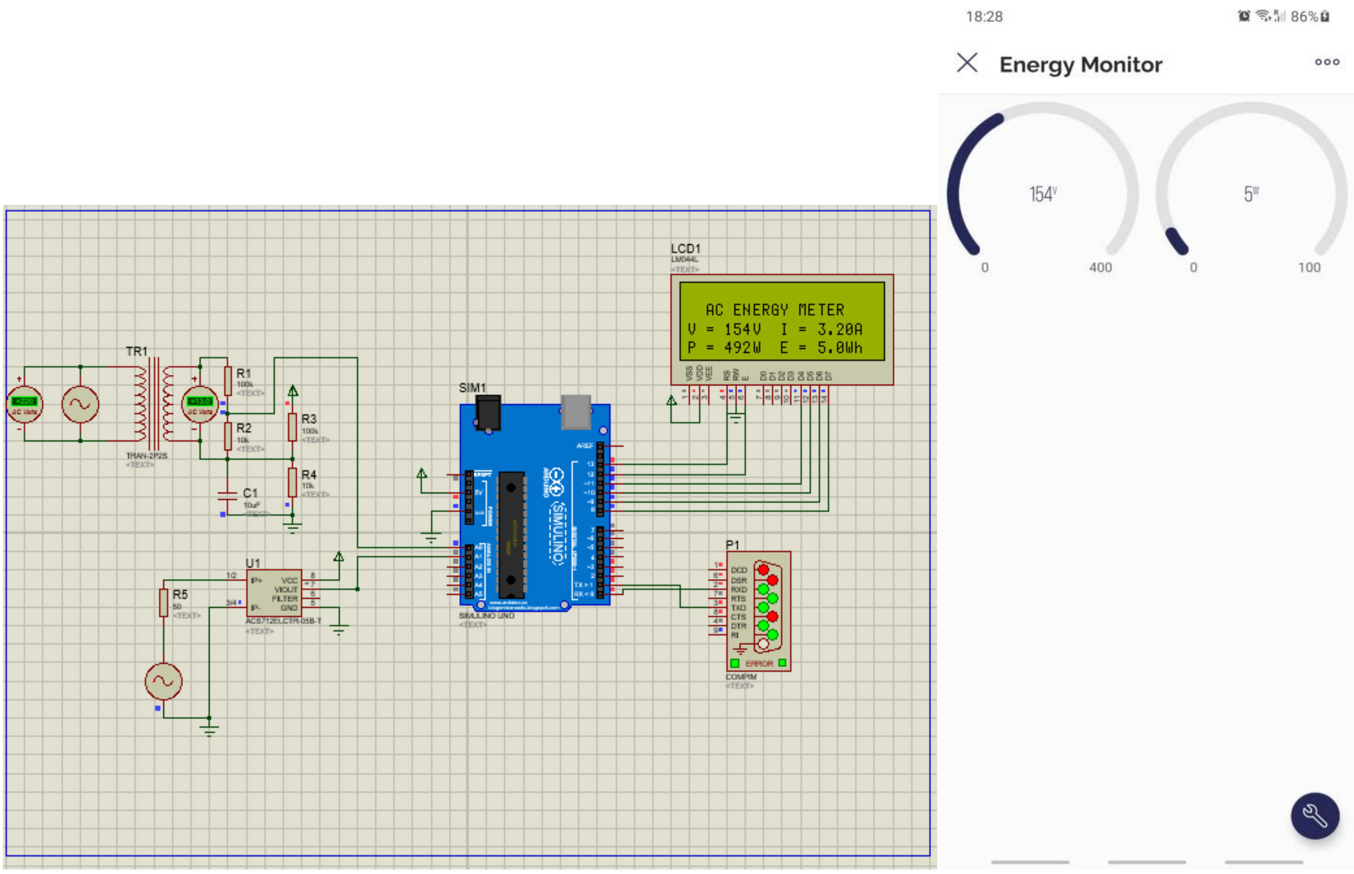

Fig 9. Mobile Interface

5. Applications

Smart Homes: IoT energy monitoring systems can be used in smart homes to track and optimize energy consumption. They provide real-time data on energy usage, allowing homeowners to identify energy-intensive appliances or behaviours and make adjustments to reduce energy waste.

Industrial Energy Management: In industrial settings, IoT energy monitoring systems help monitor and manage energy consumption in factories, warehouses, and other facilities. By collecting data from sensors installed on equipment and machinery, these systems enable businesses to identify energy inefficiencies, optimize operations, and reduce costs.

Building Automation: IoT energy monitoring systems play a crucial role in building automation and energy management. They collect data on energy usage, occupancy patterns, and environmental conditions to optimize heating, ventilation, and air conditioning (HVAC) systems, lighting, and other building systems for improved energy efficiency and occupant comfort.

Renewable Energy Integration: IoT energy monitoring systems are used to integrate and manage renewable energy sources such as solar panels and wind turbines. By monitoring the energy production and consumption in real-time, these systems ensure efficient utilization of renewable energy and facilitate grid integration.

Energy Grid Management: IoT energy monitoring systems enable utility companies to monitor and manage energy distribution across the grid. By collecting data on energy demand, supply, and distribution, these systems help utilities optimize energy generation, detect faults or outages, and improve overall grid reliability and efficiency.

Electric Vehicle Charging: With the growing adoption of electric vehicles (EVs), IoT energy monitoring systems are used to manage and optimize EV charging. These systems provide real-time data on charging station availability, energy consumption, and billing information, ensuring efficient use of resources and seamless charging experiences for EV owners.

Agriculture: IoT energy monitoring systems find applications in agricultural settings for optimizing energy usage in irrigation systems, greenhouse operations, and livestock management. By monitoring energy consumption and environmental conditions, these systems help farmers make informed decisions to reduce energy waste and improve productivity.

Energy Conservation Programs: IoT energy monitoring systems are utilized in energy conservation programs implemented by governments or utility companies. These programs aim to raise awareness about energy consumption patterns, encourage energy-saving behaviours, and provide incentives for reducing energy usage.

6. Conclusion

6.1 Summary of Contributions:

The design and implementation of the Energy Monitoring System using Arduino, Blynk, and VSPE presented in this research paper has successfully addressed the need for efficient energy management and monitoring. The system incorporates various hardware and software components to provide real-time energy consumption data, data visualization through a mobile application, and remote-control capabilities. By utilizing Arduino microcontroller, Blynk platform, and VSPE, the system offers an accessible and user-friendly solution for energy monitoring and management.

6.2 Implications and Future Work:

The Energy Monitoring System presented in this paper has significant implications for energy conservation, as it enables users to monitor and control their energy usage effectively. The system can be implemented in various settings, including residential, commercial, and industrial sectors, to promote energy efficiency and reduce wastage.

Additionally, improvements can be made in terms of system scalability and security. Scaling the system to handle a larger number of devices and data points would enable its deployment in larger buildings or even in smart cities. Strengthening the security measures to protect the system from potential cyber threats and unauthorized access is also an important aspect to consider.

REFERENCES:

- Home Electric Energy Monitoring System: Design and Prototyping João Gil Josué, João Murta Pina, and Mário Ventim Neves - https://www.researchgate.net/publication/220832504_Home_Electric_Energy_Monitoring_System_Design_and_Prototyping
- Leverage IoT for Energy Efficiency in Building Management - https://www.buildingsiot.com/blog/leverage-iot-for-energy-efficiency-in-building-management-bd#:~:text=Using%20IoT%2C%20energy%20efficiency%20is,are%20exposed%20to%20direct%20sunlight.
- Interfacing ACS712 Current Sensor with Arduino - https://robu.in/interfacing-acs712-current-sensor-with-arduino-step-by-step-guide-to-measure-current/
- Blynk documentation - https://docs.blynk.io/en/
- VSPE documentation - https://eterlogic.com/Products.VSPE.html