ENERGAUGE –INTUITIVE ENERGY METER

A Minor Project work submitted in partial fulfillment of the requirement for the award ofthe degree of

BACHELOR OF TECHNOLOGY

In

ELECTRONICS & COMMUNICATION ENGINEERING

By

M.PRAVEEN (21211A04G6)

Under the esteemed guidance of

Mr. J. Kishore (Phd)

Assistant Professor

M.Tech

B.V. Raju Institute of Technology

UGC- AUTONOMOUS

Department of Electronics and Communication

Vishnupur, Narsapur, Medak.(Dt)
(Affiliated to JNTU, Hyderabad)
2022-2023

B.V. Raju Institute of Technology

Vishnupur, Narsapur, Medak. (DT) Pin: 502313

(Affiliated to JNTU, Hyderabad)

Ph.: 08458-222000, 222001 Fax: 08458-222002

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the Minor Project work entitled **EnerGauge – Intuitive Energy Meter** is being submitted by **Mr. M. Praveen (21211A04G6)** in partial fulfillment of the requirement for the award of the degree of **B.Tech in Electronics & Communication Engineering**, byJawaharlal Nehru Technological University Hyderabad is a record of bonafide work carried out by themunder my guidance and supervision from **2022** To **2023**

The results presented in this project have been verified and are found to be satisfactory.

Internal Guide	**Head of the Department**
Mr. J. Kishore	**Dr. B.R. Sanjeeva Reddy**
Assistance Professor	**M.E, MISTE, MIEEE, PhD.**
	Professor & HOD,
	Dept. Of ECE

EXTERNAL EXAMINER

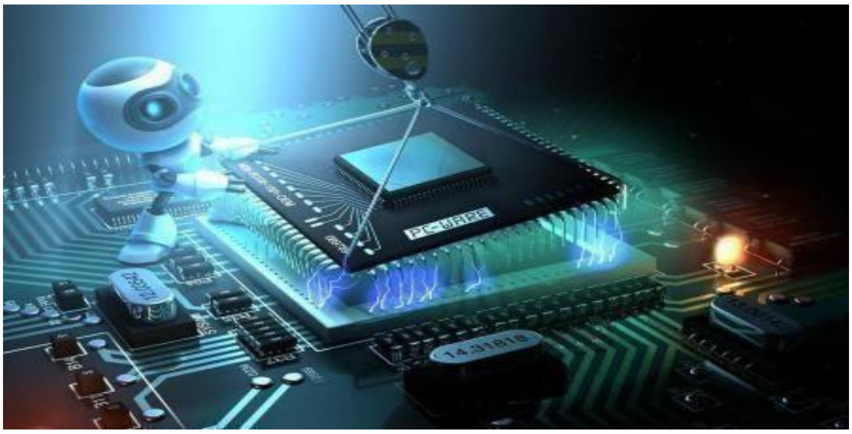
VISHNU
UNIVERSAL LEARNING

BVRIT ESTD 1997
BV Raju Institute of Technology
(UGC-Autonomous)

NBA&NAAC ACCREDITED
EAMCET CODE: **BVRI**

Department of Electronics and Communication Engineering

CENTRE FOR EMBEDDED SYSTEMS DESIGN



CERTIFICATE

This is to certify that **Mr. M. Praveen** bearing Roll No. 21211A04G6 have successfully completed his/her training on Embedded systems and Implemented a Project titled **"EnerGauge – Intuitive Energy Meter"** in Centre for Embedded Systems Laboratory, B.V. Raju Institute of Technology from **2022** to **2023**.

Coordinator

Mr. R. Anirudh Reddy, M. Tech
Assistant Professor (Phd)

Head of the Department
Dr. B.R. Sanjeeva Reddy
M.E, MISTE, MIEEE, PhD.
Professor & HOD, Dept. of ECE

ACKNOWLEDGEMENTS

We take opportunity to express our indebt gratitude to the persons who contributed for our work, for being our inspiration and guide which led to the successful completion of the project.

We are grateful towards our College Management and our beloved Principal **Dr. Sanjay Dubey** for providing us the necessary infrastructure and facilities that ensured smooth and satisfactory execution of the project. We would like to express our profound gratitude to our Head of the department, **Dr Sanjeev Reddy, Professor & HOD**, Dept. of ECE, for his encouragement inspiration and close monitoring and guidance he gave us during the execution of the project. We express our sincere thanks to our guide **Mr. J. Kishore, Assistant Professor**, Dept. of ECE, for his valuable suggestion and motivation in successful completion of project. We also wish to express our thanks to all the faculty members and laboratory staff who were helpful directly and indirectly for the completion of the project.

M. Praveen (21211A04G6)

ABSTRACT

"EnerGauge" is a pioneering energy monitoring system that offers real-time insights into current, voltage, and power consumption. Utilizing the ESP32 module and Blynk app integration, the project provides a seamless platform for users to track their energy usage dynamically. Notably, EnerGauge goes beyond monitoring by enabling users to remotely control appliances using relays. The system's unique feature lies in its ability to generate instant electricity bills based on real-time usage and customizable tariff rates. EnerGauge exemplifies the fusion of hardware and software to empower users with real-time energy data visualization, calculation, and remote management capabilities, ultimately fostering informed energy consumption decisions. The user gets an opportunity to monitor his energy consumption regularly and encourages him to reduce his usage.

DECLARATION

We hereby declare that the project titled "EnerGauge – Intuitive Energy Meter" submitted to **B.V Raju Institute of Technology, affiliated to Jawaharlal Nehru Technological University, Hyderabad** for the award of degree of **Bachelor of Technology in Electronics and Communication Engineering** is a result of original project work done by us under the guidance of **Mr. J. Kishore,** M-Tech, Assistant Professor Department of ECE

It is further declared that the project report on any part therefore has not been previously submitted to any university or institute for the award of degree.

M. Praveen (21211A04G6)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO ENERGAUGE -Intuitive Energy Meter

The world is witnessing an increasing demand for efficient energy management and sustainable practices to combat the challenges posed by rising energy consumption and environmental concerns. In this context, the development of Smart Energy Meters has emerged as a significant solution to monitor and optimize electricity usage.

This paper introduces an advanced Smart Energy Meter, which harnesses cutting-edge technology to accurately measure and monitor electrical parameters in a single-phase electrical system. The meter integrates essential components, including the ZMPT101B voltage sensor and SCT-103 000 current sensor, to provide precise readings of voltage and current levels, enabling real-time analysis of power consumption.

At the heart of this smart meter lies the ESP32 microcontroller**.** It acquires data from the voltage and current sensors and executes sophisticated algorithms to calculate crucial electrical parameters**.** The microcontroller also features a dual relay module, facilitating remote control of the connected load for enhanced energy efficiency and user convenience.

The LCD display is seamlessly integrated into the smart meter, providing users with real-time energy consumption data and essential electrical parameters. Furthermore, the Blynk platform is employed to enhance the smart meter's capabilities. By integrating wireless connectivity and cloud-based data storage, users can remotely access real-time energy consumption data and billing information on their smartphones through the Blynk app.

A standout feature of this Smart Energy Meter is its ability to calculate the electrical bill, including GST and tariff details. Users can configure the meter with GST rates and tariff structures applicable to their location and energy provider. The meter then accurately computes the billing amount based on power consumption data, enabling users to have a comprehensive understanding of their electricity expenses

The successful implementation of this prototype highlights its potential to revolutionize energy management in residential and commercial settings. As this Smart Energy Meter evolves and becomes more widely adopted, it promises to play a pivotal role in shaping a greener and more sustainable future.

1.1.1 OBJECTIVE

The aim of an energy consumption meter is to measure and monitor the amount of electrical energy consumed by a specific device, appliance, or an entire house. It provides valuable information to users, helping them understand their energy usage patterns, identify areas for potential energy savings, and make informed decisions to optimize energy efficiency and reduce costs. As we can also track our electricity bill, the user can control the energy usage saving electricity.

1.1.2 MOTIVATION

Electricity plays a cardinal role in day-to-day life. The electrical energy consumption in India is the third biggest after China and USA with 5.5% global share in 2016. The per person energy use rate in India is closer to 0.7 KW. India's share with global energy demand will rise to 9% by 2035.

So we want our device to accurately monitor the energy utilization and display the analysis to the user and notify him so that he can utilize it in a proper manner.

Electricity theft incites huge losses for utilities globally. With the growing installation of smart meters, these conflicts are growing more and more discrete and challenging to identify. With the implementation of smart meter, the specific energy reading from any machine can easily be matched with the electrical meter data to conclude whether electricity theft is taking place or not. Thus, smart meters can give important insights and an answer to the problem of electricity theft.

1.1.3 SCOPE OF STUDY

The Smart energy meter market, owing to the rapid adoption of home automation technologies, changing consumer preferences inclining towards smart home applications, growing need for a convenient lifestyle, improved energy savings, advancement in IoT framework, high acceptance of technologically advanced products among customers and the average disposable income of the population is expected to grow at a greater pace.

The smart-home ecosystem is continuing its accelerated expansion, but market growth rates are directly connected to the speed of 5G implementation. Recent IoT investments by Apple, Amazon, Google, or Alibaba have transformed the landscape noticeably, giving opportunities for all sorts of companies, but have also forced market strengthening. Acknowledging the contemporary moderate user penetration, long product replacement cycles and rising device connectivity, the adoption of smart meters will improve in the future.

The most fundamental application of the smart meter is energy monitoring and Control in household use, for example in Smart HVAC Systems. Apart from that, smart plugs give real-time insights with more exceptional accuracy at equipment level.

A particular feature of smart meters such as standby power-saving and occupancy detection will assist to save energy without yearly intervention by the user. The interpretation of the energy savings tendered over a month or a year can then be included in the electricity bill to intensify energy awareness in the public domain. Also, the detection of defective devices using excessive energy proves to be an advantage to both the user as well as utility.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

Introducing an advanced smart energy meter crafted from the ZMPT 101B voltage sensor, SCT 013 000 current sensor, ESP32 microcontroller, relay module, LCD display, Blynk IoT platform, and an intricate billing algorithm. This smart meter transcends traditional monitoring, offering real-time voltage, current, power, and energy data. Through the ESP32 and Blynk, remote control and live data viewing from smartphones become effortless, enhancing user convenience. The relay module extends control by allowing remote device management, optimizing energy use. The LCD display ensures instant access to vital energy statistics, promoting informed choices.

What sets this smart meter apart is its capability to calculate electricity bills. By integrating the tariff details and applying a billing algorithm, it computes accurate costs. It even factors in GST, ensuring comprehensive billing accuracy. This innovation introduces profound financial transparency, enabling users to witness the direct correlation between energy consumption and costs. This project champions cost savings, energy efficiency, environmental responsibility, and technical ingenuity, while unveiling the intricacies of energy economics in a user-friendly manner.

2.2 RELATED WORKS

S.NO	**PAPER TITLE**	**AUTHORS**	**REFERED FROM**	**YEAR OF PUBLICATION**	**REMARKS**
1.	Design and implementation of smart energy meter	V. Preethi	Project Hub	Feb 7,2021	It just displays the energy readings.
2.	Smart Automated Home Application using IoT with Blynk App	Homera Durani Mitul Sheth Madhuri Vaghasia Shyam Kotec	Research Gate	October,2020	Using Blynk to display the monitored data on users phone.

3.	Electricity Bill Forecasting Application by Home Energy Monitoring Systemglasses for blind	Charnon Chupong	Project Hub	June 5,2019	Generating electricity bill along with displaying units of power consumed

Table 2.2.1 Previous works done on Energy meter

CHAPTER 3

ANALYSIS AND DESIGN

3. 1 SETTING UP ARDUINO IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.
Programs written using Arduino Software (IDE) are called sketches**.** These sketches are written in the text editor and are saved with the file extension. ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righth and corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Step 1 − First you must have your Arduino board (you can choose your favourite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable.

Fig 3.1.1: A to B connector

Step 2 − Download Arduino IDE Software.
You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

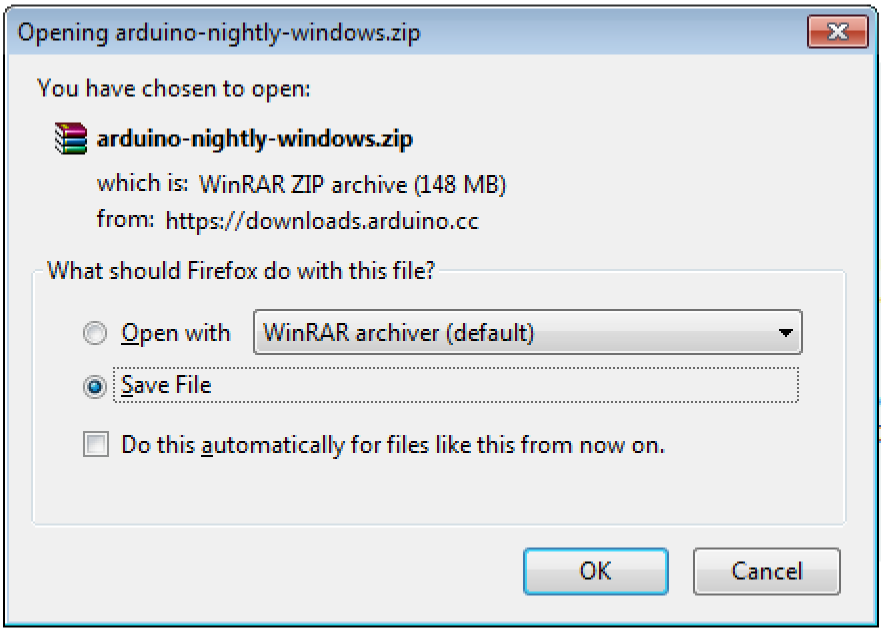


Fig 3.1.2: Downloading Arduino IDE

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power fromeither, the USB connection to the computer or an external power supply.

If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double click theicon to start the IDE.

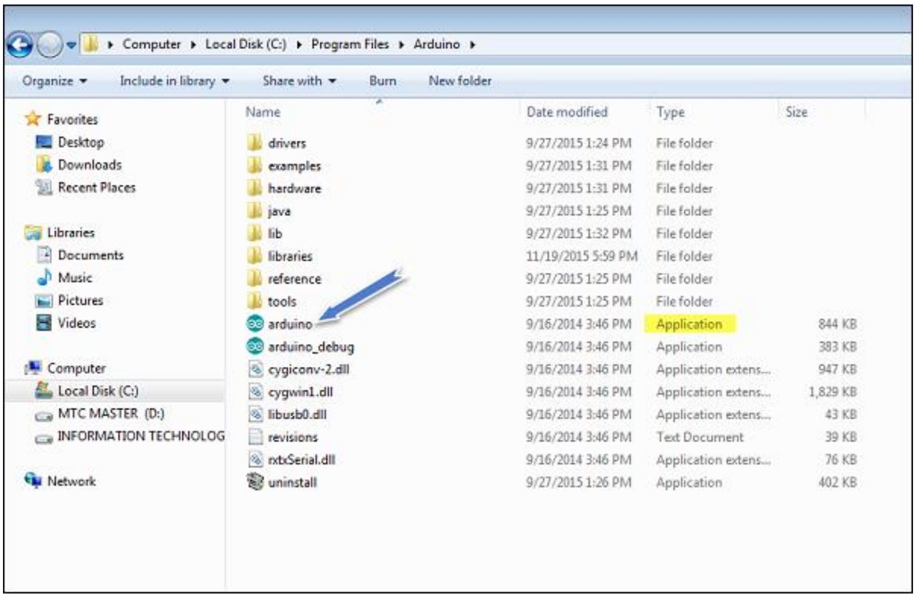


Fig 3.1.3: launch Arduino IDE

Step 5 - Open your first project. Once the software starts, you have two options.

Ï Create a new project.

Ï Open an existing project example.To create a new project, select File → New.

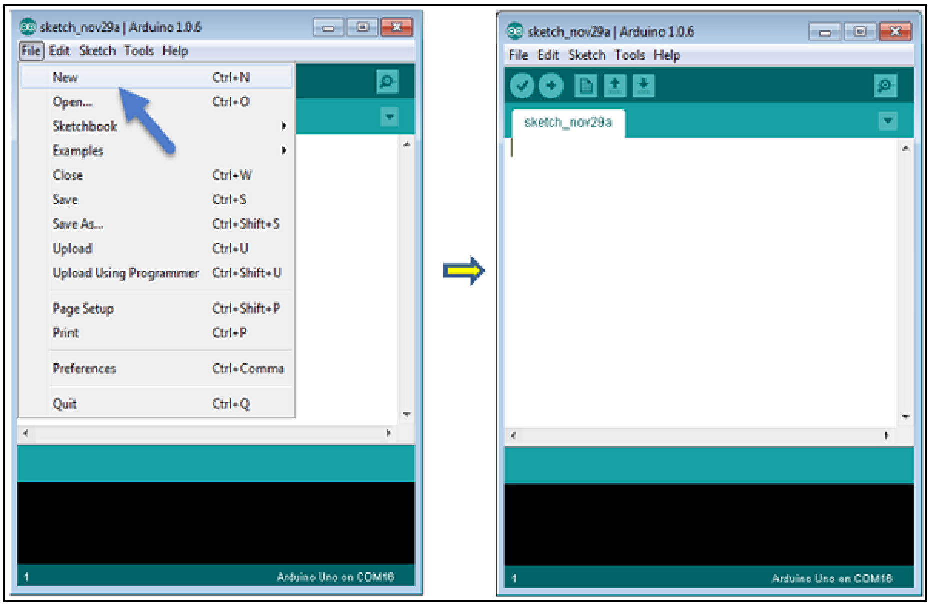


Fig 3.1.4: to create a new project

To open an existing project example, select File → Example → Basics → Blink.

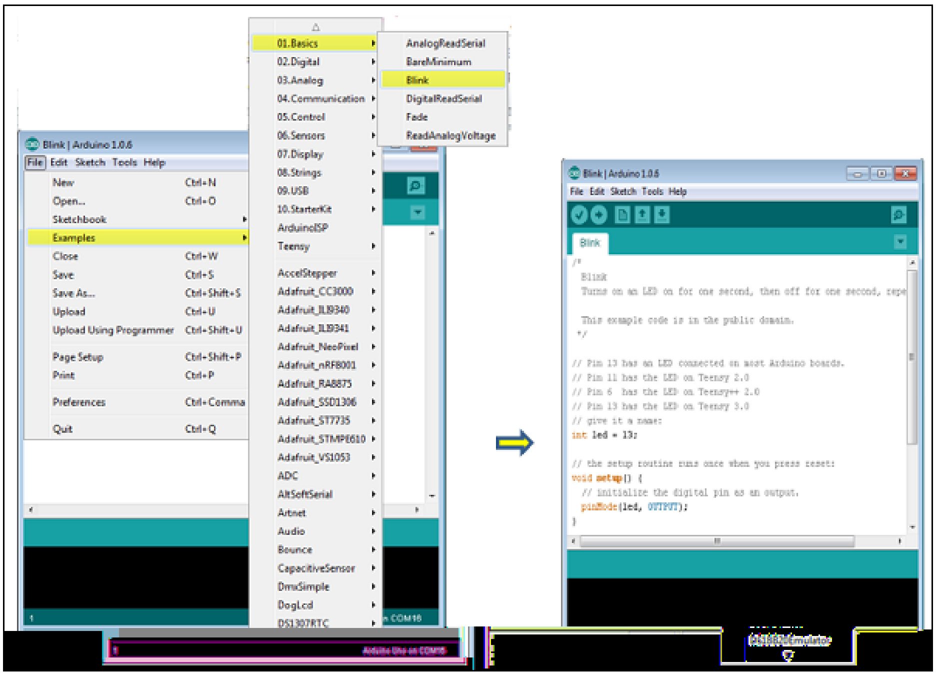


Fig 3.1.5: Opening project

Here, we are selecting just one of the examples with the name Blink. It turns the LEDon and off with some time delay. You can select any other example from the list.

Step 6 – Select your Arduino board

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer. Go to Tools → Board and select your board.

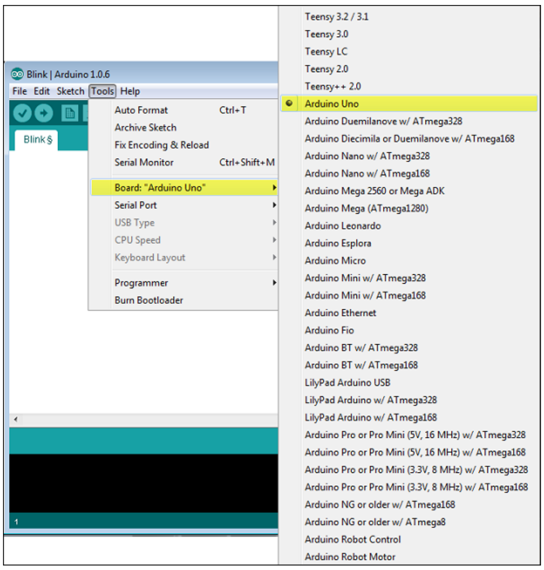


Fig 3.1.6: Selecting Arduino board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This islikely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports).

To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

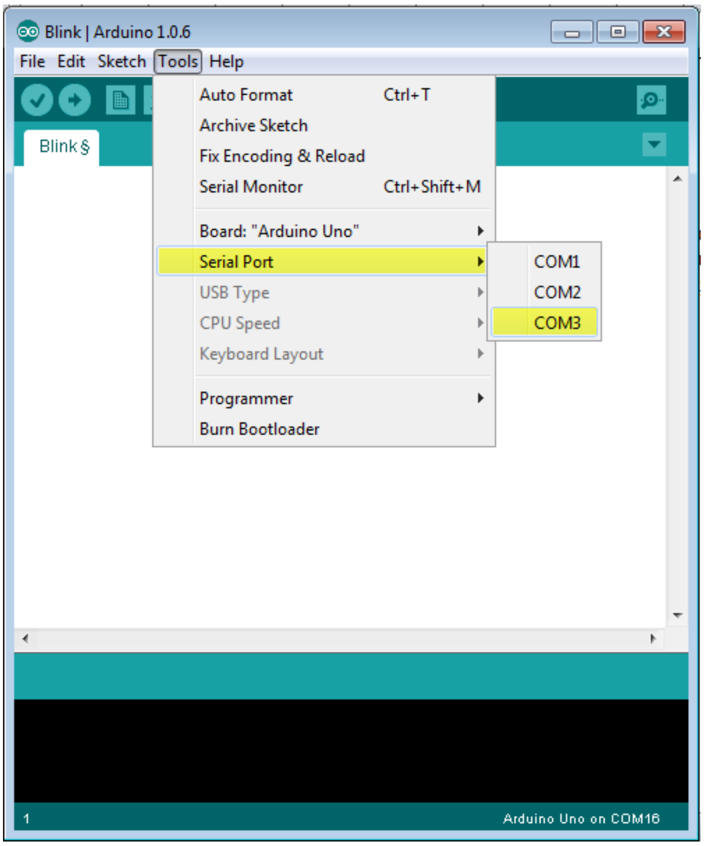


Fig 3.1.7: serial port selection.

Step 8 −Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstratethe function of each symbol appearing in the Arduino IDE toolbar.

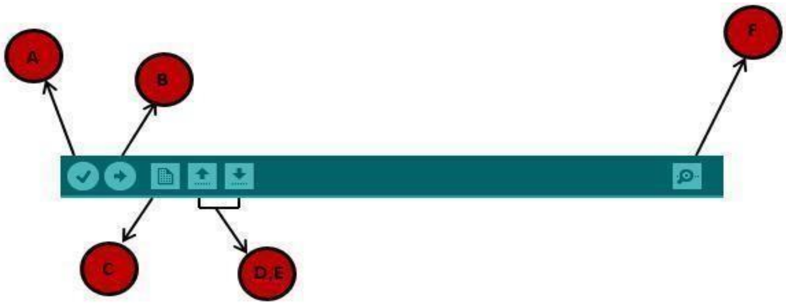


Fig 3.1.8 Uploading program

3.2 SETTING UP BLYNK ACCOUNT:

Blynk is a comprehensive software suite that enables the prototyping, deployment, and remote management of connected electronic devices at any scale. Whether it's personal IoT projects or commercial connected products in the millions, Blynk empowers users to connect their hardware to the cloud and create iOS, Android, and web applications, analyze real-time and historical data from devices, remotely control them from anywhere, receive important notifications, and much more.

To get started with the Blink app, follow the instructions below:

With Blink, you have one user account with one email and one phone number. On log in and certain other actions, a verification number is sent to the mobile number or email address of the account.
Note: If you want to share your account with other users, learn more about setting up a second phone or device.

Launch the Blink App and tap **Create Account** at the bottom of the screen
1. Select your country from the drop-down menu and tap **Next**.
2. Your region is pre-selected based on your IP address. Confirm your region and tap **OK**.
3. Enter a valid email address, and tap **Next**.
4. Create a password and tap **Create Account**. The password requirements are listed at the bottom of the screen. Tap the eyeball icon if you want to view what you are typing.

5. Complete the multi-factor authentication (MFA) process. Learn more about multifactor authentication.

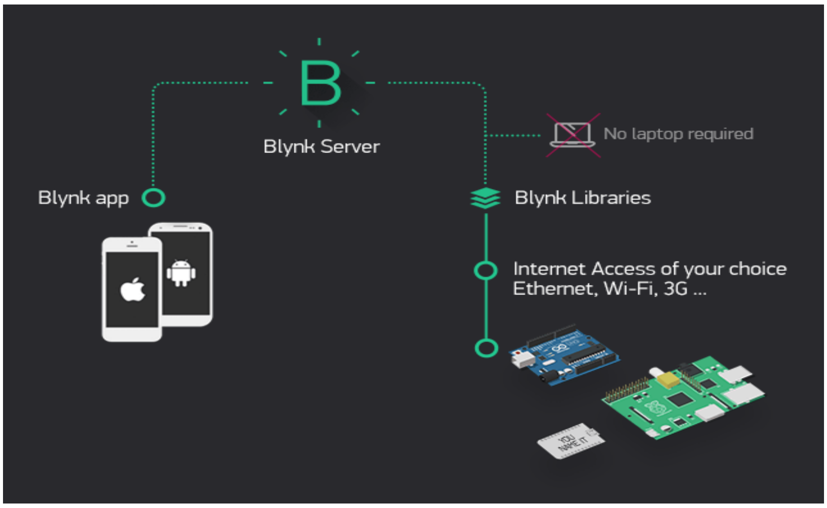


Fig 3.2.1 Blynk application

3.3 HARDWARE TOOLS USED:

3.3.1 ESP32 Microcontroller:

The ESP32 is a versatile and widely used microcontroller developed by Espressif Systems. It is known for its integrated Wi-Fi and Bluetooth capabilities, making it suitable for a wide range of IoT and embedded applications. The pinout of the ESP32 module can vary based on the specific development board or module.

Fig: 3.3.1(a): ESP32 Microcontroller

PIN Description:

1. GPIO Pins (General Purpose Input/Output): These pins can be configured for various functions, such as digital input/output, PWM, SPI, I2C, UART, etc. They are labeled GPIO0 to GPIO39.

2. Power Supply Pins:

3V3: 3.3V power supply output.

GND: Ground connection.

3. Analog Input Pins:

A0: Analog input pin that can be used for ADC (Analog-to-Digital Conversion).

4. Serial Communication Pins (UART):

TX0/RX0: UART transmit (TX0) and receive (RX0) pins for serial communication

TX2/RX2: Additional UART transmit (TX2) and receive (RX2) pins.

5. I2C Pins:

SDA: I2C data line.

SCL: I2C clock line.

6. SPI Pins:

MOSI: Master Out Slave In (SPI data input).

MISO: Master In Slave Out (SPI data output).

SCK: SPI clock signal.

CS/SS: Chip Select/Slave Select for SPI communication.

7. PWM Pins: These pins can generate Pulse Width Modulation signals for various applications like controlling motors, LED brightness, etc.

8. Boot Mode Selection Pins:

GPIO0: Used to determine the boot mode of the ESP32 during startup.

EN (Enable): Enable pin to reset the module or put it into deep sleep mode.

9. Other Functionality Pins:

VBAT: Battery voltage input pin (for powering ESP32 from a battery).

IOREF: Voltage reference for input/output pins.

3.3.2 VOLTAGE SENSOR-ZMPT 101B:

Fig:3.3.2(a) Voltage Sensor-ZMPT 101B

The ZMPT101B is a voltage sensor module designed to measure AC voltage. It is commonly used in various applications, including energy monitoring, home automation, and power quality analysis. Here's a description of the ZMPT101B voltage sensor:

1. Voltage Sensing Range: The ZMPT101B can measure AC voltage in the range of 0 to 250V. It is suitable for standard household voltages found in most countries.

2. Non-Contact Measurement: The sensor provides non-contact measurement, meaning it does not require direct electrical connections to the measured voltage source. Instead, it uses a transformer to induce a voltage proportional to the input voltage, ensuring electrical isolation and safety.

3. Built-in Voltage Transformer: The module incorporates a voltage transformer, which steps down the measured voltage to a lower and safer level for the output.

4. Output Signal: The output signal of the ZMPT101B is analog and linearly proportional to the measured voltage. The output voltage range is typically 0 to 1V, corresponding to the input voltage range of 0 to 250V.

5. Calibration: The module may require calibration to ensure accurate voltage measurements. Calibration involves mapping the output voltage to the actual input voltage using calibration constants.

6. Low Power Consumption: The ZMPT101B is designed with low power consumption, making it suitable for battery-powered applications.

7. Application Compatibility: The sensor can be used with various microcontrollers, such as Arduino and other development boards, to process the analog output and display or transmit the voltage readings.

Pin Description

1.VCC: This pin is used to supply power to the sensor module. It is usually connected to a 5V power source.

2. GND: The GND pin is the ground connection of the sensor module, and it should be connected to the ground of the power supply and other components.

3. VOUT: The VOUT pin is the output of the sensor module. It provides an analog voltage signal that is linearly proportional to the measured AC voltage. The output voltage typically ranges from 0 to 1V, corresponding to the input voltage range of 0 to 250V.

3.3.3 CURRENT SENSOR-SCT-013-000

Fig:3.3.3(a) Current Sensor-SCT-013-000

The SCT-013-000 is a type of current sensor designed to measure alternating current (AC) in electrical systems. It is a non-invasive sensor, which means it can measure current without the need for direct electrical contact with the current-carrying conductor. Here are some key details about the SCT-013-000 current sensor:

1. Current Sensing Range: The SCT-013-000 is commonly available in different models, each with a specific current sensing range. The most common variants have a sensing range from 0 to 100A or 0 to 200A, but other ranges may also be available.

2. Non-Invasive Measurement: The sensor operates based on the principle of magnetic induction. It is designed to be clamped around a current-carrying conductor (e.g., a wire) to measure the magnetic field generated by the current passing through the conductor.

3.Output Signal: The SCT-013-000 generates an alternating current (AC) output signal that is proportional to the measured current. The output signal has a magnitude and frequency corresponding to the current being measured.

4. Application Compatibility: The SCT-013-000 current sensor can be used with various microcontrollers, such as Arduino or Raspberry Pi, to process the output signal and obtain current measurements.

5. Calibration: For accurate measurements, calibration may be required. Calibration involves mapping the output signal to the actual current being measured using calibration constants.

Overall, the SCT-013-000 current sensor provides a convenient and safe method to measure AC currents in electrical systems, making it suitable for energy monitoring, power consumption analysis, and other applications that involve current measurement.

3.3.4 RELAY MODULE

Fig: 3.3.4(a) Relay Module

- A relay module is an electronic device that allows you to control high-voltage and high current circuits using low-voltage microcontrollers or digital signals.
- It acts as an interface between the low-power control circuit and the high-power load, providing electrical isolation and safety.
- Relay modules are widely used in home automation projects, industrial control systems, robotics, and other applications where electrical isolation between control and load circuits is crucial.
- They are often employed to control appliances, lights, motors, and other electrical devices that require higher currents or voltages than what microcontrollers or digital signals can handle directly.

3.3.5 16*2 LCD DISPLAY

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

Pin decription:

- **Pin1 (Ground/Source Pin):** This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- **Pin2 (VCC/Source Pin):** This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- **Pin3 (V0/VEE/Control Pin):** This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- **Pin4 (Register Select/Control Pin):** This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- **Pin5 (Read/Write/Control Pin):** This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- **Pin 6 (Enable/Control Pin):** This pin should be held high to execute Read/Write

process, and it is connected to the microcontroller unit & constantly held high.

- **Pins 7-14 (Data Pins):** These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- **Pin15 (+ve pin of the LED):** This pin is connected to +5V
- **Pin 16 (-ve pin of the LED):** This pin is connected to GND.

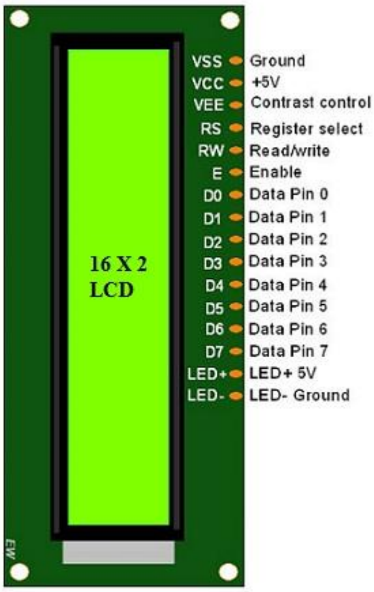


Fig 3.3.5(a) 16*2 LCD Display

CHAPTER 4
IMPLEMENTATION

4.1 BLOCK DIAGRAM:

In the below block diagram the inputs are voltage sensor, current sensor, power supply and the outputs are cloud, 16*2 LCD display, relay module.

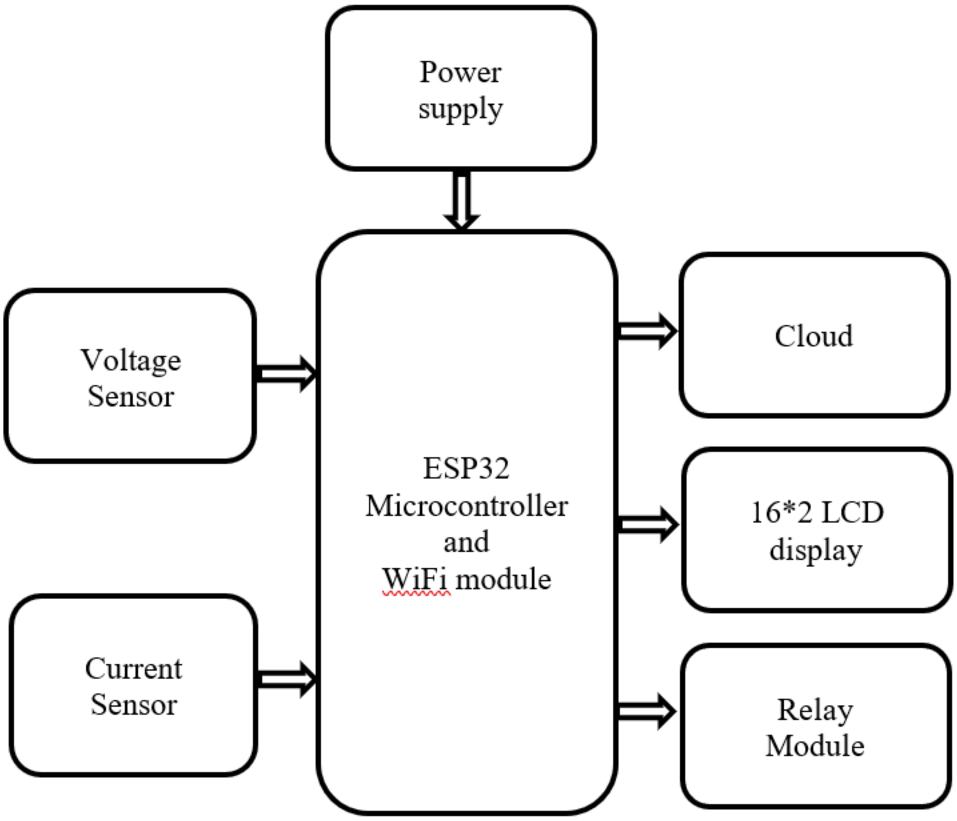


Fig 4.1.1 Block Diagram

4.2 SCHEMATIC:

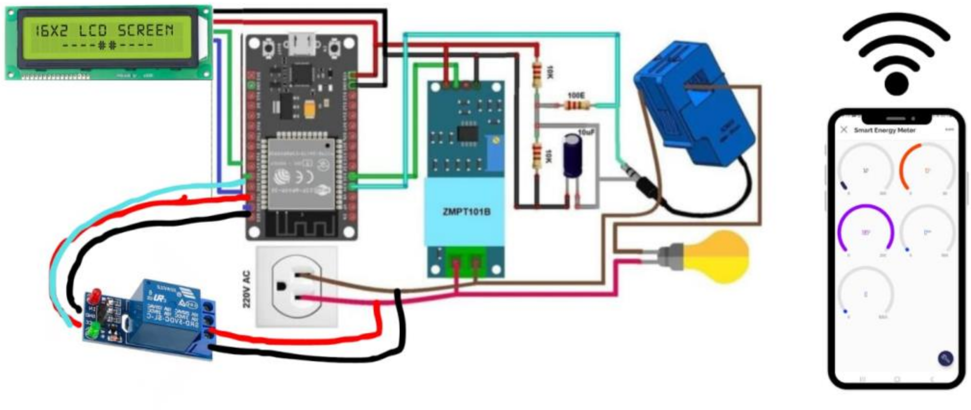


Fig 4.2.1 Schematic

DESIGN:

The components used are one ESP 32 board, a Current sensor, Voltage sensor, LCD display, Relay modules, Power supply, 10W bulb, breadboard, USB cable and connecting wires. The step-by-step process of hardware connections are as follows:

Step 1. Fix all the components on the breadboard or PCB, as shown in figure.

Step 2. Make a common ground and a common positive on breadboard/PCB.

Step 3. Connect the ESP 32 to Current sensor using following resistors and capacitors circuit.

Step 4. Connect the input of Voltage sensor parallel to the load and connect the pins of voltage sensor to the ESP 32 module.

Step 5. Connect the Current sensor as given in the circuit and pass the load line of the connected load from the current sensor.

Step 6. Connect the relay modules in series to the required appliances and connect the pins of relay module to the ESP 32 module.

Step 7. Connect the LCD Display I2C pins to the ESP 32 module.

Step 8. Now upload the designed code into ESP 32 module.

Step 9. Design web dashboard and mobile dashboard on Blynk app to display the readings captured from the sensors.

Step 10. Upload the prepared code into ESP 32 module and connect it to internet. As the device comes online we can start monitoring the readings on dashboard prepared.

4.3 FUNCTIONING:

In the described setup, the sensors initiate the process by collecting voltage and current data, which then converges to compute real-time power consumption. This system also includes a monitoring mechanism to detect sudden energy spikes beyond defined limits. The gathered data is seamlessly transmitted to the cloud through IoT technology, ensuring both data security and accessibility. This integrated approach enables users to effortlessly access a comprehensive visual representation of their energy usage patterns. Through graphical insights, users can better understand their consumption trends, alongside the provision of an electricity bill derived from the recorded data. This holistic system amalgamates cutting-edge technology with practicality, empowering users to effectively manage and comprehend their energy consumption patterns.

4.4 FLOWCHART:

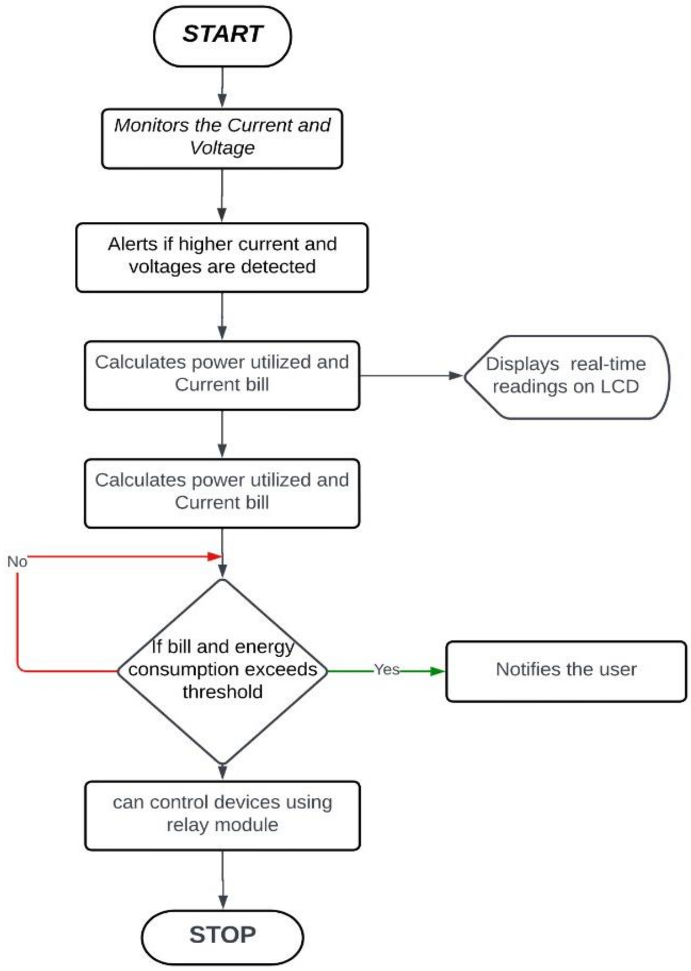


Fig 4.4.1 Flowchart

CHAPTER 5
RESULTS

5.1 KIT PHOTO:

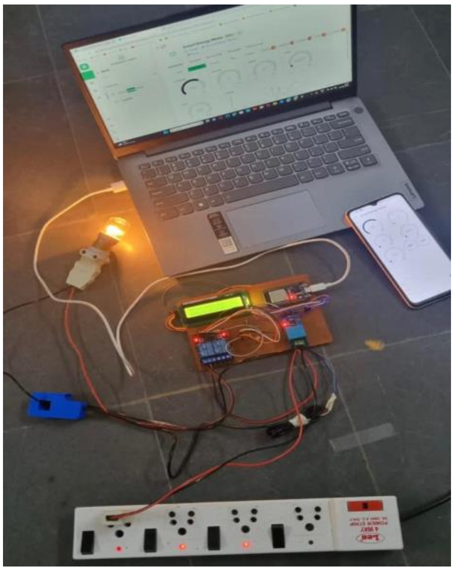


Fig 5.1.1 Kit Photo

5.2 RESULTS:

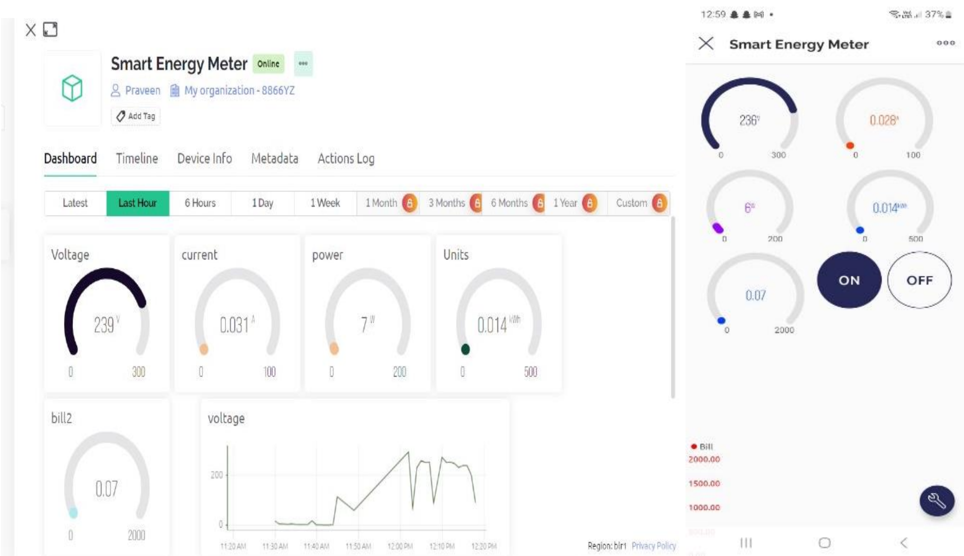


Fig 5.2.1 Results

CHAPTER 6

APPLICATIONS, ADVANTAGES AND LIMITATIONS

6.1 APPLICATIONS:

- Individuals and families can track their real-time energy consumption and make informed decisions to reduce usage.
- Accurate billing calculations help households manage their electricity expenses and understand how different devices impact their bills.
- Businesses can monitor their energy usage patterns and identify areas for improvement to enhance efficiency and reduce operational costs.
- Implement demand response strategies by adjusting energy usage during peak times to lower electricity bills.
- Monitor machinery and equipment to distribute power efficiently, preventing overloads and reducing downtime.
- Implement dynamic pricing based on real-time energy demand, encouraging consumers to reduce consumption during peak hours.

6.2 ADVANTAGES:

- **User-Friendly Convenience:** The system prioritizes user ease, ensuring that its operation is straightforward and uncomplicated.
- **Dependable Performance:** A robust architecture underscores the system's reliability, guaranteeing consistent and trustworthy functionality.
- **Real-Time Insight:** Users gain immediate access to their energy consumption, allowing them to monitor their usage in the moment.
- **In-Depth Usage Analysis:** The system delivers a comprehensive breakdown of power consumption, offering users a thorough understanding of their energy habits.
- **Clear Billing Information:** Current billing details are presented clearly, providing users with a concise overview of their accrued charges.
- **Mobile Integration:** Seamlessly integrating with mobile devices, the system enables users to effortlessly track their energy usage while on the move.
- **Remote Appliance Control:** The system's remote control capability, powered by the

relay module, permits users to manage high-load appliances from a distance.

6.3 LIMITATIONS:

- **Installation and Maintenance Complexity:** Setting up the system requires technical knowledge, potentially posing challenges for users unfamiliar with electronics.

- **Data Accuracy Dependency:** The accuracy of data collected relies on the precision of the sensors and calibration, which might lead to occasional inaccuracies.

- **Internet Connectivity Reliance:** The Blynk app and remote control functionality are dependent on a stable internet connection, which might cause disruptions in cases of network issues.

- **Regulatory and Legal Hurdles:** Depending on the region, there might be regulations or legal considerations regarding energy trading, sharing, and data privacy.

CHAPTER 7

CONCLUSION, FUTURE WORK AND FUTURE SCOPE

7.1 CONCLUSION:

The developed prototype offers a user-friendly, cost-effective, and accessible solution for monitoring daily energy usage. While the paper primarily focuses on the prototype's design rather than extensive testing, the initial test outcomes are promising. The prototype effectively identified electricity consumption patterns within the installed area.

The primary objective of this project was to establish an effortless monitoring system that motivates users to adopt energy conservation practices. Although the current scope emphasizes the prototype's design, its performance during preliminary testing showcases its potential.

Future enhancements could involve streamlining the components to reduce size and weight, facilitating potential commercialization. This innovation aligns with the corporate pursuit of efficient resource management and sustainability.

7.2 FUTURE WORK:

- **Advanced Analytics:** Incorporate machine learning for deeper insights into energy usage, predicting peak consumption times, and offering personalized energy-saving recommendations.
- **Enhanced UI:** Improve user interfaces on both the LCD display and Blynk app, with better data visualizations and interactive features for managing devices.
- **Smart Appliances:** Expand compatibility to control and monitor a wider range of smart appliances, allowing remote management and energy optimization.
- **Real-time Alerts:** Implement instant alerts for abnormal energy patterns, faults, or usage spikes, encouraging prompt action.
- **Energy Auditing:** Provide detailed breakdowns of energy usage by specific appliances to promote efficiency.
- **API Integration:** Develop APIs for third-party apps, expanding functionality and compatibility.

7.3 FUTURE SCOPE:

The future scope of the smart energy meter system, encompassing components like the ZMPT 101B voltage sensor, SCT-013-000 current sensor, ESP32 microcontroller, relay module, LCD display, Blynk app, and advanced billing with GST and tariff details, holds immense potential for advancement. This includes refining user interfaces for intuitive data access, integrating AI-driven insights for personalized energy recommendations, and enabling alerts for real-time anomaly detection. Exploring energy trading and sharing options, predictive maintenance, and blockchain integration could further revolutionize energy management. Bidirectional grid communication, seamless home automation, and API integration for smart cities promise more comprehensive energy optimization. Personalized carbon footprint tracking, integration with energy storage, dynamic tariff adjustments, and global standardization contribute to a holistic approach toward sustainability and innovative energy management.

CHAPTER 8
REFERENCE

REFERENCES

[1] (2018) Smart Automated Home Application using IoT with Blynk App
Homera Durani ; Mitul Sheth ; Madhuri Vaghasia ; Shyam Kotech

[2] (2018) IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid
Bibek Kanti Barman ; Shiv Nath Yadav ; Shivam Kumar ; Sadhan Gope

[3] (2021) A Novel IoT based Smart Energy Meter for Residential Energy management
Gitanjali Mehta ; Ruqaiya Khanam ; Vinod Kumar

[4] (2021) Real-time Monitoring of Energy Meters Using Cloud Storage – IEEE International Power and Renewable Energy Conference (IPRECON).

[5] https://www.tutorialspoint.com/arduino/images/mini_cable.jpg

[6]https://www.tutorialspoint.com/arduino/images/opening_arduino_nightly_windows.jpg

[7] https://www.tutorialspoint.com/arduino/images/launch_arduino_ide.jpg

[8] https://www.tutorialspoint.com/arduino/images/edit_file.jpg

[9] https://www.tutorialspoint.com/arduino/images/open_project.jpg

[10] https://www.tutorialspoint.com/arduino/images/select_board.jpg

[11] https://www.tutorialspoint.com/arduino/images/serial_port.jpg

[12] https://www.tutorialspoint.com/arduino/images/upload_program.jpg

[13] https://osoyoo.com/driver/blynk/blynk-iot.png

[14]http://www.etteam.com/prodESP/ESP32-DEV-KIT/ESP-WROOM-32%20WIFI_2.jpg

[15]https://4.bp.blogspot.com/IQb0PVvRndM/WYfT15UM4JI/AAAAAAAAGTk/o1L1W2_Tk7YMgikIaHcrBJsXwi3Vrhm7ACLcBGAs/s320/Capture.JPG

[16]https://www.tinytronics.nl/shop/image/cache/data/product-2556/sct013-000v-1200x1200h.jpg

[17] https://robu.in/wp-content/uploads/2019/11/5V-2-Channel-Relay-Module-1.jpg

[18] https://images.app.goo.gl/iE57UVxFhCg4MZ6K9

[19] Flowchart made from https://lucid.app/documents#/documents?folder_id=recent

[20] Schematic designed from https://www.canva.com/design/DAFrDogoyaY/pU_HB45NfF9oc49AtUMfTg/edit

CHAPTER 10
APPENDIX

10.1 CODE:

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
#include "EmonLib.h"
#include <EEPROM.h>
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

EnergyMonitor emon;
#define vCalibration 83.3
#define currCalibration 0.50

float tariffRate = 5.0; // Cost per unit (kWh)
float gstRate = 18.0;

#define RELAY_PIN_1 4 // GPIO 4 for controlling the first relay
#define RELAY_PIN_2 5

BlynkTimer timer;
char auth[] = "gDwIZZqPC4fT5AYZxlIq4PHwWFH-LGoV";
char ssid[] = "Samsung";
char pass[] = "8106971560se";

float kWh = 0;
unsigned long lastmillis = millis();

void myTimerEvent()
{
  emon.calcVI(20, 2000);
  kWh = kWh + emon.apparentPower * (millis() - lastmillis) / 3600000000.0;
  float ratePerKWh = tariffRate; // Tariff rate in Indian Rupees per unit (kWh)
  float bill = kWh * ratePerKWh;
```

```
yield();
Serial.print("Vrms: ");
Serial.print(emon.Vrms, 2);
Serial.print("V");
EEPROM.put(0, emon.Vrms);
delay(100);

Serial.print("\tIrms: ");
Serial.print(emon.Irms, 4);
Serial.print("A");
EEPROM.put(4, emon.Irms);
delay(100);

Serial.print("\tPower: ");
Serial.print(emon.apparentPower, 4);
Serial.print("W");

EEPROM.put(8, emon.apparentPower);
delay(100);

Serial.print("\tkWh: ");
Serial.print(kWh, 5);
Serial.println("kWh");
EEPROM.put(12, kWh);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Vrms:");
lcd.print(emon.Vrms, 2);
lcd.print("V");
lcd.setCursor(0, 1);
lcd.print("Irms:");
lcd.print(emon.Irms, 4);
lcd.print("A");
```

```
  lcd.setCursor(0, 2);
  lcd.print("Power:");
  lcd.print(emon.apparentPower, 4);
  lcd.print("W");
  lcd.setCursor(0, 3);
  lcd.print("kWh:");
  lcd.print(kWh, 4);
  lcd.print("W");

  lastmillis = millis();

  Blynk.virtualWrite(V0, emon.Vrms);
  if (emon.Vrms > 300) {
    Blynk.logEvent("voltage", "HIGH VOLTAGE IS DETECTED");
  }
  Blynk.virtualWrite(V1, emon.Irms);
  if (emon.Irms > 100) {
    Blynk.logEvent("current", "HIGH CURRENT IS DETECTED");
  }
  Blynk.virtualWrite(V2, emon.apparentPower);
  Blynk.virtualWrite(V3, kWh);
  if (kWh > 130) {
    Blynk.logEvent("kWh", "high kwh is detected");
  }
  Blynk.virtualWrite(V6, bill);
  if (bill > 2000) {
    Blynk.logEvent("bill", "high bill is detected");
  }
}

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
```

```
  lcd.init();
  lcd.backlight();
  emon.voltage(35, vCalibration, 1.7);  // Voltage: input pin, calibration, phase_shift
  emon.current(34, currCalibration);    // Current: input pin, calibration.

  timer.setInterval(5000L, myTimerEvent);
  lcd.setCursor(3, 0);
  lcd.print("IoT Energy");
  lcd.setCursor(5, 1);
  lcd.print("Meter");
  delay(3000);
  lcd.clear();
}

void loop()
{
  Blynk.run();
  timer.run();
}
```