IoT Based Smart Energy Consumption Monitoring System

This project report has been submitted to the Department of Electrical and Electronic Engineering, Sonargaon University of Bangladesh, in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronics Engineering.

Submitted By

Name: Azaharul Islam

ID: EEE1902017055

Name: Angkon Chandra Das

ID: EEE1902017080

Name: Md Mannan Miah

ID: EEE1902017020

Name: Md. Shahadat Hossain

ID: EEE1703012045

Name: Md Shahin Alam

ID: EEE1701010147

Supervised by

Mr. Md Rais Uddin Mollah

Asst. Co-Ordinator

Department of Electrical and Electronics Engineering

SONARGAON UNIVERSITY

January, 2023

DECLARATION

We do hereby solemnly declare that the work that has been presented in this project paper has been carried out by us and has not been previously submitted to any University/Organization for an academic qualification or for professional qualification. We hereby ensure that the work that has been presented does not breach any existing copyright.

We further undertake to indemnify the University against any loss or damage arising from breach of the forgoing obligation.

CERTIFICATION

This is to certify that the project paper on "**IoT Based Smart Energy Consumption Monitoring System**" is the bona fid record of project work done by Name: Name: Azaharul Islam ID: EEE1902017055, Name: Angkon Chandra Das ID: EEE1902017080,Name: Md Mannan Miah ID: EEE1902017020, Name: Md. Shahadat Hossain ID: EEE1703012045, Name: Md Shahin Alam ID: EEE1701010147 for their partial fulfillment of the requirements of B.Sc. Degree in Department of Electrical and Electronics Engineering from Sonargaon University

The project report has been carried out under my guidance and is a record of the successful work. I wish their every success in life.

Supervised By

…….....

Md. Rais Uddin Mollah

Asst. Co-Ordinator
Department of Electrical and Electronics Engineering
Sonargaon University

ACKNOWLEDGEMENT

We are very grateful to almighty Allah at first for finishing my project successfully. We would like to express our profound gratitude to our supervisor, **Md Rais Uddin Mollah, Asst. Co-Ordinator,** Department of Electrical and Electronics Engineering, Sonargaon University, for his guidance throughout all phases of our project. We would like to thankhim for his cordial suggestion which helped us to reach our goal. We would also like to thankall our faculty members, Lab instructors of the Department of Electrical and Electronics Engineering, Sonargaon University. The authors are also grateful to **Professor Dr. M. Bashir Uddin,** Head of the Department of EEE and all respected teachers of Electrical and Electronics Engineering Department for their co-operation and significant help for completing the project work successfully.

Thank you all

Authors

Name: Azaharul Islam
ID: EEE1902017055

Name: Angkon Chandra Das
ID: EEE1902017080

Name: Md Mannan Miah
ID: EEE1902017020

Name: Md. Shahadat Hossain
ID: EEE1703012045

Name: Md Shahin Alam
ID: EEE1701010147

Abstract

This project describes the digitization of energy meter usage readings over the internet. The growth of population and the rise of electrical appliance usage have been increasing the electricity consumption, leaving a lot of the industrial sectors desperate to optimize their energy consumption. The energy meter provided by the service provider for billing is an accumulated energy used type. This meter measures in kilowatt-hour (kWh) which is not suitable for detailed review and analysis. The lack of data measurement and recording for every energy load point lead to incorrect planning and target for managing energy performance. Thus, this project developed an energy monitoring system that aims to provide users with information about the consumption trend in their buildings. In this project, the power sampling in watts or kilowatts has been taken in the distribution board from two different consumption loads. The two load sampling data were transmitted using Node MCU ESP82266 as the controller to MySQL to store the data in the cloud. As part of this project, the system is provided with a website and a mobile application that allow users to get the real-time energy usage for every load for the entire building. Based on the experimental result using sampling data, the website displayed a live data page providing information on power usage, accumulated kWh, cost in Ringgit Malaysia and the peak power in watt. This project contributes to the existing knowledge of energy monitoring systems by providing detailed energy reviews from two or more load points as well as the capability to store raw data in the cloud which can be retrieved from the website for future planning and management. Now day consuming electricity is increasing as the evolution of new technology is emerging. The electrical power company measures the electrical power unit consumed by the users at the end of every month, but for this they need to visit each customer. It is getting difficult for the electricity board officials to take energy meter readings and calculate the bill of the consumed electricity. Also, they put extra bill amount even though if the electrical bills are paid. So to overcome this, it is necessary to make a monitoring system that can automatically detect the meter readings and can save human efforts of visiting every house and then there will be no fraud bill amount that consumers need to pay. By adding these features to the energy meter, it can be termed as Smart energy meter.

Contents

Name of the Content	Page No
Letter of Transmittal	1
Declaration	2
Certificate	3
Acknowledgement	4
Abstract	5
Contents	6
List of Figures	8
List of Tables	9
Chapter 1: Introduction	
1.1 Introduction	10
1.2 Objective	12
1.3 Background of the Project	12
1.4 Internet of things	12
1.5 IoT System	13
1.6 Methodology	13
Chapter 2: Literature Review	
2.1 Introduction	14
2.2 Literature Survey	15
Chapter 3: Hardware Implementation	
3.1 Introduction	17
3.2 Block Diagram of This Project	17
3.3 Circuit Diagram of this Project	18
3.4 The list of Devices used in the Project	19
3.4.1 Node MCU	19
3.4.2 SMPS	22
3.4.3 Voltage Sensor	23
3.4.4 LCD Display	24
3.4.5 Relay Module	25
3.4.6 Current Sensor	26
3.5 Project Implementation Result	29

Chapter 4: Software and simulation		
4.1	Introduction	30
4.2	Arduino IDE	30
4.3	Proteus 8 Professional	36
4.4	Blynk App	38

Chapter 5: Discussion and Conclusion		
5.1	Discussion	43
5.2	Conclusion	43
5.3	Advantages	44
5.4	Future Works	44

References	45
Appendix	47

List of Figures

Figure No	Figure Contain	Page No
Figure 1.1	Methodology	13
Figure 3.1	Block Diagram of our proposed system	17
Figure 3.2	Circuit Diagram of our proposed system	18
Figure 3.3	Node MCU ESP8266	19
Figure 3.4	SMPS	22
Figure 3.5	Voltage Sensor	23
Figure 3.6	LCD Display	25
Figure 3.7	Relay Module	25
Figure 3.8	schematic Of Relay Module	26
Figure 3.9	Pin out of Relay Module	26
Figure 3.10	Current Sensor	26
Figure 3.11	Project picture	29
Figure 4.1	Java Point	31
Figure 4.2	Toolbar Button	31
Figure 4.3	Page Setup	32
Figure 4.4	Edit	33
Figure 4.5	Added file	34
Figure 4.6	Tools	34
Figure 4.7	Help	36
Figure 4.8	Schematic Capture	37
Figure 4.9	Biynk Info	38
Figure 4.10	DataStream	39
Figure 4.11	Events	39
Figure 4.12	Automations	40
Figure 4.13	Web Dashboard	41
Figure 4.14	New Device Created	41

List of Table

Figure No	Figure Contain	Page No
Table 01	LCD display outline	15

CHAPTER 1
INTRODUCTION

1.1 Introduction

In this era, the world is facing more challenges related to the power and energy consumption as the population of the world is increasing gradually. One of the major crises is in the field of power distribution to different places. Burden on electricity offering divisions is also increasing because the number of users is increasing rapidly throughout worldwide [1]. The users are facing problem like getting poor electric supply as well as they also receiving due bills that they have already paid. Sometime Meter tampering i.e. illegal means of electricity bill payment is also done in some of the areas. This is why consumer has to suffer for this inconsistent billing even though they have used minimum amount of energy [4]. Therefore it is necessary to apply a technique of taking readings of the meter automatically, which can realize the energy consumption management to the customers to be controllable, adjustable to save the electrical energy. Also to take the reading of the meter a human operator has to go turn by turn to every resident & commercial building, this increases the work and efficiency and therefore the working hours also increases to achieve the complete area data reading [2]. So to achieve efficient meter reading, operation cost, reduce billing error we need a system that can read the meter reading automatically at each time interval i.e. IOT based energy meter. Electric energy meters is the direct billing interface between the power companies and consumers and it undergone several advancements in the last decade. This present project "IoT Based Smart Energy Meter" solve the problems faced by both the consumers and the distribution companies [5]. This paper mainly describes the smart energy meter, including Arduino UNO as the main controller which utilizes the combination of hardware and software both in order to implement desired functionality of IOT based energy meter. This paper demonstrates the IOT based system in which the consumer as well as the service provider will get the used energy reading with the respective amount at each time interval [2]. This proposed energy meter system replaces the traditional meter reading methods and enables remote access to existing energy meter by the authority. Also, it can monitor the meter readings regularly without the person visiting each premise [1].

Nowadays, there is a massive movement in installing energy monitoring systems in new smart homes and even in the industries. Automatic lighting systems, temperature control systems,

security control systems, and many other functions are developed not only for a stylish lifestyle but also for energy conservation [1]. Based on usual practice, most of the electricity providers have workers collect the data at the premises' energy meters [2]. The collection of data was done manually to provide the number of units consumed (NUC) and the consumption period. The data is then sent to the server for electricity bill generation. These practices are done regularly every month at all premises. Thus, the existing system requires human intervention and is time-consuming in interpreting the energy meter reading of the consumers [3], [4]. The use of cloud computing for the energy monitoring system help users to monitor the data online [5], [6]. The system gathered and analysed the data and then directly send the data through the client's device using the Internet of Things (IoT) [7], [8]. The IoT-based smart energy meter project enables the electricity provider to record the meter readings remotely without human involvement every month and the energy consumption data can be displayed on a webpage [9], [10]. The demand for information traffic has recently matured because of the variety of connected devices and good meters. To satisfy the explosive growth of big data, new technologies are needed to enhance the prevailing metering infrastructure [11]. The Automatic Meter Reading (AMR) system uses SMS as the medium to transfer data to the server where the system generates bills automatically. However, this approach is still limited and inefficient because the SMS will only display the overall energy consumption for the month. Moreover, there is still the possibility of accessibility bottlenecks to isolated places [12]. A system using the existing Electronic Energy Meters was introduced where GPRS is used as the communication medium to get the values of the meter. Since GPRS is more cost-effective than SMS, it is very useful for infrequent updates [13]. It makes it possible to monitor energy meters at a lower cost. Thus, reports of daily consumption can be created and monitored via an Android application and/or web portal [14]. However, this approach only records the total accumulated energy consumption for the whole building for the whole month and cannot be divided into every floor area or any load point. To address this issue, we introduce an energy monitoring system that could analyse the energy data from different consumption loads. In this project, two different consumption loads were used to record the sampling data. It records the data on average every 30 seconds using a microcontroller and transfers it to the secure repository of the cloud which is MySQL. Finally, in an easy-to-understand interface, a website and a mobile phone display the customer's energy consumption. Users may view their in-house data consumption in real time for the live reading, total usage and total cost. This feature allows users to identify at specific times, the use of excessive energy, and to address wasteful habits.

1.2 Objective of the Project

The main objective of this project is to “Design a smart energy meter using IoT ’’ project for the main purpose is to create a smart system so that we can take many benefits from one project.

- Remote monitoring system
- The line can be Control loads without any physical interfare.
- Monitoring the power line current.
- Monitoring the value of consumed power with running month cost.

1.3 Background of the Project

- The title of project is “IoT Based Smart Energy meter System”.
- In this project we have done voltage, consumed power and Cost monitoring. As well as monitoringthe current of the system.
- We can monitor from anywhere through the internet.
- We can also do this if we want to turn a system on or off.

1.4 Internet of Things (IoT)

The Internet of Things (IoT) is a new revolution for Data Transfer and Storage. Objects that make themselves recognizable and they obtain intelligence by making or enabling context related decisions to the situations. They can transfer information about themselves. They can access information that has been used by other things, or they can be components of other services. The three factors that makes IoT look forward are Sensing Nodes, EmbeddedProcessing and Communication. This transformation is accompanied by the emergence of cloud computing capabilities supported by an increased storage capacity and high-end data processing and the Machine-to-Machine communication for data transport with complete security for data. By introducing cloud computing, we can make a full call to the storage resource pool and computing resource pool in the cloud computing architecture, and provide high reliability for IoT cloud storage service and efficient cloud computing services to users. This Machine-to-Machine service layer will provide the needed services like data transport, security, devices, management and device discovery in a harmonized manner across a verticaldomain to the application layer.

1.5 IoT System

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human- to-computer interaction. The definition of the Internet of Things has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commoditysensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of Things. In the consumer market, IoT technology is most synonymous with

products pertaining to the concept of the " Raspberry Based Weather Station ", covering devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. There are a number of serious concerns about dangers in the growth of IoT, especially in the areas of privacy and security; and consequently industry and governmental moves to begin to address these.

1.6Methodology

Basically, the design and development of this project are divided into two main parts which are hardware architecture and software details. In the hardware architecture, the design of the circuit was constructed and the prototype of the project was built. While in the software development, the whole complete prototype was operated via programming codes.

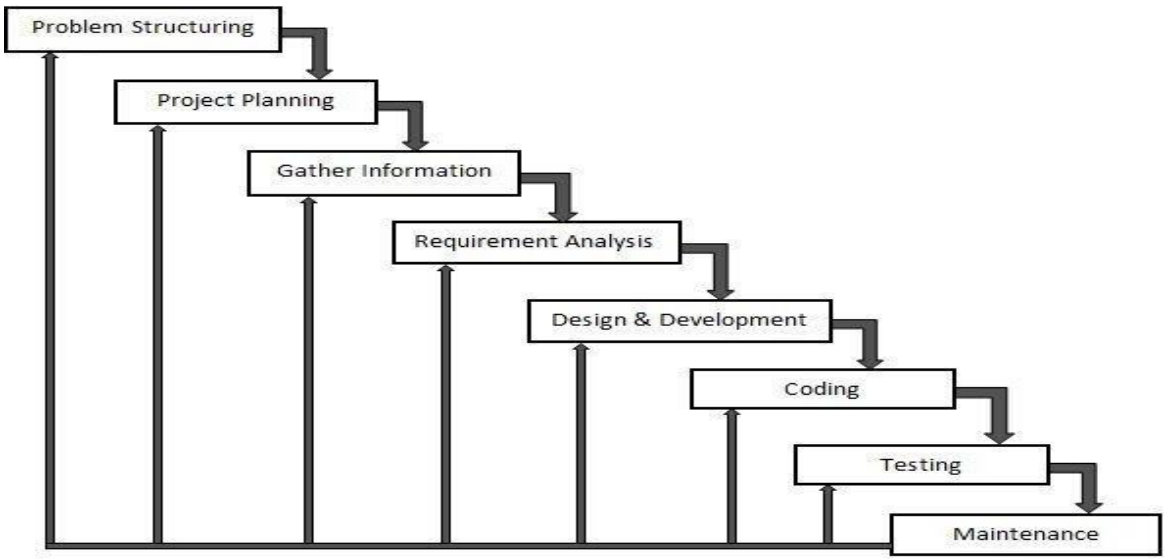


Fig-1.1: Methodology

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In The Internet of things concept enables us to connect the normal day to day devices with each other over the internet. The devices connected through IOT concept can be controlled and analysed remotely. The IOT concept provides the basic infrastructure and opportunities to form a connection between the physical world and computer based systems [8]. The concept has been gaining importance with more and more wireless devices that are increasing rapidly in the market. It connects the hardware devices with each other over the internet. The ESP 8266 Wi-Fi module used in the system provides the connectivity with the internet in the system. Now-a-days the demand for electricity is increasing at a constant rate throughout the population and is being utilized for various purposes wiz, agriculture, industries, household purposes, hospitals etc.,. So, it is becoming more and more complicated to handle the electricity maintenance and requirements. Therefore there is an immediate requisite to save as much electricity as possible. As the demand from the newer generations of population for electricity is increasing so in accordance with it the technology improvement is needed. The proposed system provides a technical twist to the normal energy meters using the IOT technology. Also there are other issues that we have to address such as power theft and meter tampering which in turn generate economic loss to the nation. Monitoring, Optimized power usage and reduction of power wastage are the major objectives that lie ahead for a better system. The present system vastly depends on human involvement for billing. Billing requires a human individual to visit each and every customer's energy meter and generate the bill by taking the unit readings from the energy meter. This is a time consuming process.

To address all the mentioned constraints we developed a system on the basis of IOT technology. Smart energy meter using Wi-Fi system is designed based on three major objectives. They are

- To provide automated meter reading over an immediate basis.
- To use the electricity in an optimized manner
- In order to reduce the power wastage.

Similarly the system should also be useful to the service end. So the system basically can be classified on the basis of service ends in two ways.

- Consumer end
- Service end.

The data from the system is displayed on a webpage which can be accessed by both the consumer and service provider. The system is designed on an Arduino micro controller [2]. It can be structurally differentiated into three parts controller, theft detection circuit and a WiFi unit. The controller performs the basic calculations and processes the information. Theft detection circuit provides information about any meter tampering and the most important role is played by the Wi-Fi unit to send the information from the controller over the Internet. The service end can remind the consumer about the bill generated for the usage over a common messaging platform. The Arduino controller is programmed on the Arduino software IDE which is a prerequisite to operate on the Arduino board. Its code is derivative of the c language. The block diagram consists of an Arduino UNO board, an ESP 8266 Wi-Fi module and a 16*2 LCD display. The WiFi module is the main component used in the IOT operation. The centre piece being the Arduino board provides the connection between the different components of the proposed system .The Arduino UNO board is based on the ATmega 328p processor. It is the core of the system which is necessary for the principle operations that are necessary to be carried out such as the automatic electricity billing and tampering detection inputs from the tampering circuit. The load represents the devices that require the electricity to operate. The ac supply is connected to the system through the transformers to power the system. The Meter is also connected to the system to automate the power usage of the household. The readings from the energy meter are then processed and are updated over the Wi-Fi through the ESP 8266 Wi-Fi Module.

2.2 Literature Survey

One Internet of things (IOT) based and highly desirable in field of energy, in this framework customer can do control administration by Knowing vitality utilization time to time, the buyer needs to pay the bill on plan. On the off chance that couldn't the electric power availability can be killed self-ruling from the far off host. [1] Explained the modelling and working of different units of the system and also discussed the components and their functions such that IOT and its working microcontroller and its architecture. Reducing energy consumption and monitors the units consumed. To make the electrical apparatuses insightful and give solace to devoured and to lessen control utilization in web applications. [2] Suggested it in light of ARDUINO UNO controller and

IOT innovation. On the off chance that any altering happens the controller will send to information to the server and in addition it is chopped down the vitality supply naturally. At the point when most extreme request of vitality expends will be shown in the meter utilized by the customer. [3] Clarified in the wake of surpassing the greatest request, the meter and subsequently the association will be consequently disengaged by an installed framework embedded in the meter sensor. The LDR (Light Dependent Resistor) sensor placed on energy meter which sense LED blinking pulse. At that time microcontroller sending this reading via GSM module and its send this message to electricity board. In this framework a keen vitality meter is introduced in each customer unit a server is kept up at the specialist co-op side. [4] Implemented both the meter a server furnished with GSM module which encourages bidirectional correspondence between the two closures utilizing the current GSM foundation. Shopper can without much of a stretch energize their vitality meter by sending a stick number covered up in a scratch card to the server utilizing SMS. In order to avoid all these drawbacks we have intended to construct an IOT based energy meter so that proposed energy meter measures the amount of power consumed and uploads it to cloud from which the concerned person can view the reading .The power reading send to cloud using ESP8266, a Wi-Fi module. [5] Explained the power reading from digital wattmeter is read using the coupler and transmitted digitally to the Arduino. So it automates the process of measuring the power consumption at homes using IOT.

CHAPTER 3

HARDWARE IMPLEMENTATION

3.1 Introduction

Therefore it is necessary to apply a technique of taking readings of the meter automatically, which can realize the energy consumption management to the customers to be controllable, adjustable to save the electrical energy. Also to take the reading of the meter a human operator has to go turn by turn to every resident & commercial building, this increases the work and efficiency and therefore the working hours also increases to achieve the complete area data reading. So to achieve efficient meter reading, operation cost, reduce billing error we need a system that can read the meter reading automatically at each time interval i.e. IOT based energy meter. Electric energy meters is the direct billing interface between the power companies and consumers and it undergone several advancements in the last decade. This present project "IoT Based Smart Energy Meter" solve the problems faced by both the consumers and the distribution companies

3.2 Block Diagram of This Project

In this part of this book we attached here Block diagram of this proposed system. Here we are going to describe about our hardware part by showing part by part in this block diagram.

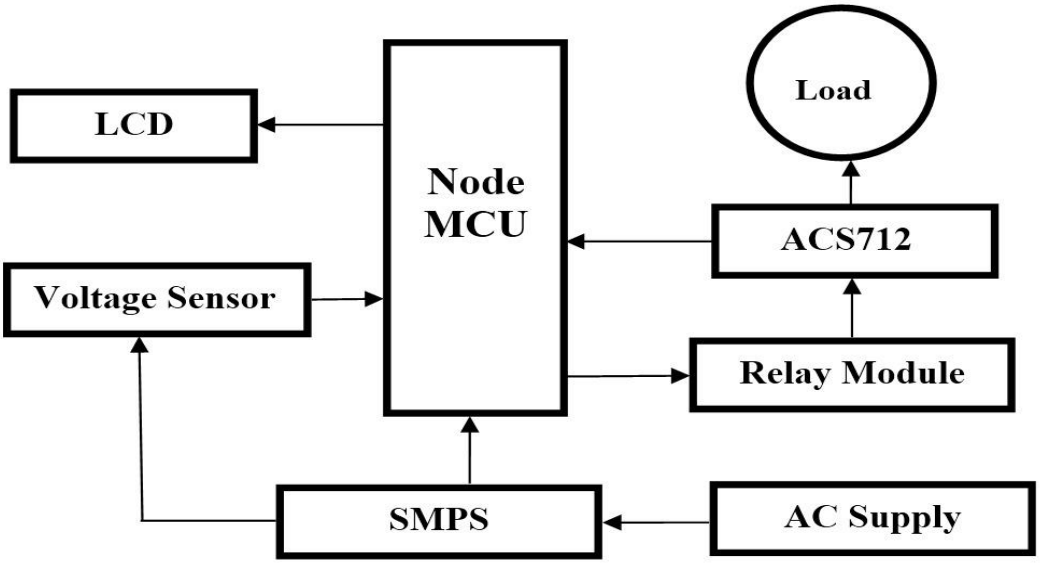


Fig: 3.1 Block Diagram of our proposed system

We use here a Node MCU to control this total system which is shown here in the middle point of this diagram. We need to make sure power supply to operate our total system. For that reason we attached here AC Supply and then we convert this AC voltage to DC 5V DC. All components like Voltage sensor, Current Sensor, Relay, LCD and Node MCU are powered ON by using 5V DC voltage. Voltage sensor will measure the voltage value of main AC line. ACS712 or Current sensor will measure the consumed current value. Relay module is used here.

3.3 Circuit Diagram of This Project

Here we attached here full circuit diagram of this proposed system. Here we use proteous simulation software to design this circuit diagram which version is Proteous 8.0. Here we use a Node MCU to control our total system by using the arduino program.

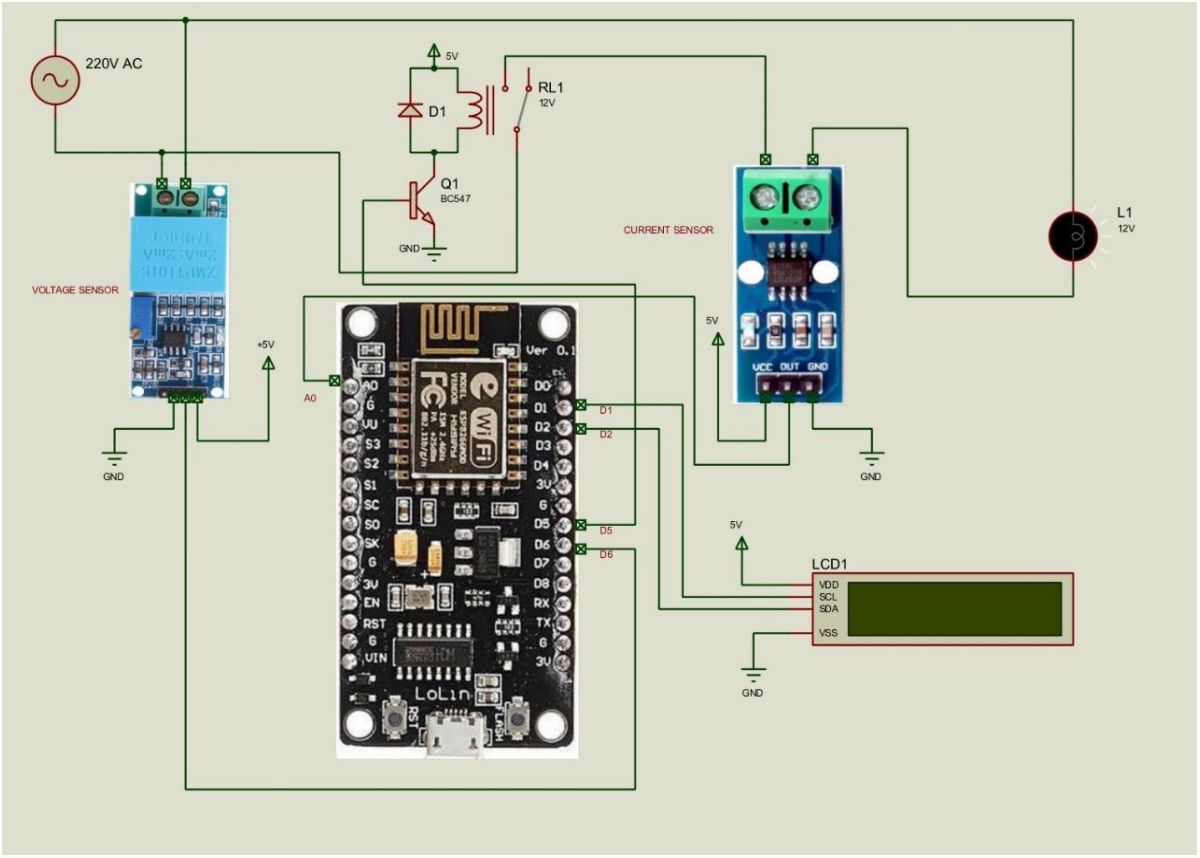


Fig: 3.2 Circuit Diagram of our proposed system

You see here that we use here Voltage sensor to measure the voltage value of our supply line. Current Sensor is works for measure the consumed current value by using any load, Node MCU is the main controller of this system, LCD will show all the data which is measured here from different sensor.

3.4 The list of devices used in the project is given below

- Node MCU.
- Voltage Sensor.
- LCD Display.
- Relay Module.
- Current Sensor.
- SMPS

3.4.1 Node MCU

The Here we use a Node MCU or ESP8266 development Board which is transformation of ESP12E wifi module and containing the ESP8266 chip which capacity is 32-bit LX106 RISE. It supports RTOS and it can operate 80MHz to 160MHz with adjustable clock frequency. It contain 1280 KB RAM and also 4 MB of flash memory to store data and programs. In this module we find wi-fi and Bluetooth and deep sleep operating features that make it ideal for IoT Projects. This ESP8266 is a family of Wi-Fi modules developed by the Chinese company Expressive. It is a System on a Chip, abbreviated simply as Sock with integrated TCP/IP protocol stack. It consists of a Ten silica L106 32-bit micro controller and a Wi-Fi transceiver. It has 11 input/output (I/O) pins. And it has a dedicated analog input for the programming purpose. It is programmable and it can be modified in several ways and that is the reason why this chip is the most prevalent IoT device in the market.

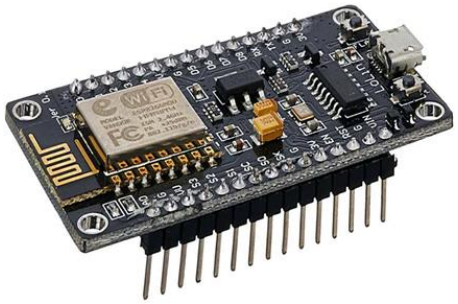


Figure 3.3: Node MCU ESP8266

In our system, we are using this Wi-Fi module because of its higher reliability and easy programming. (Peter, 2007) Wi-Fi module acts as a micro web server and hence eliminates the need for the wired connection between the Arduino board and computer which has many

advantages. Main advantages are the reduction in size, cost, and increased convenience.

The Wi-Fi module will require an active internet connection and it will act as the gateway for the Arduino board to communicate with the internet. This proposed system is flexible. I.e. apart from the above-listed components, our system can be interfaced with other sensors according to the user requirement. As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU used in the Arduino Due, they needed to modify the Arduino IDE so it would be relatively easy to change the IDE to support alternate tool chains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WiFi SoC, popularly called the "ESP8266 Core for the Arduino IDE".[18] This has become a leading software development platform for the various ESP8266-based modules and development boards, including Node MCUs. The Node MCU (Node Micro Controller Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds.

The NodeMCU (Node MicroController Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds. However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the “computer” on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware. This level of integration is not a problem using the ESP8266 as an embedded controller chip in mass-produced electronics. It is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects.

But, what about Arduino? The Arduino project created an open-source hardware design and

software SDK for their versatile IoT controller. Similar to NodeMCU, the Arduino hardware is a microcontroller board with a USB connector, LED lights, and standard data pins. It also defines standard interfaces to interact with sensors or other boards. But unlike NodeMCU, the Arduino board can have different types of CPU chips (typically an ARM or Intel x86 chip) with memory chips, and a variety of programming environments. There is an Arduino reference design for the ESP8266 chip as well. However, the flexibility of Arduino also means significant variations across different vendors. For example, most Arduino boards do not have WiFi capabilities, and some even have a serial data port instead of a USB port.

However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the "computer" on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware. This level of integration is not a problem using the ESP8266 as an embedded controller chip in mass-produced electronics. It is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects. But, what about Arduino? The Arduino project created an open-source hardware design and software SDK for their versatile IoT controller. Similar to Node MCU, the Arduino hardware is a microcontroller board with a USB connector, LED lights, and standard data pins. It also defines standard interfaces to interact with sensors or other boards. But unlike Node MCU, the Arduino board can have different types of CPU chips (typically an ARM or Intel x86 chip) with memory chips, and a variety of programming environments. There is an Arduino reference design for the ESP8266 chip as well. However, the flexibility of Arduino also means significant variations across different vendors. For example, most Arduino boards do not have WiFi capabilities, and some even have a serial data port instead of a USB port. The Node MCU Development Board can be easily programmed with Arduino IDE since it is easy to use. Programming Node MCU with the Arduino IDE will hardly take 5-10 minutes. All you need is the Arduino IDE, a USB cable and the Node MCU board itself. You can check this Getting Started Tutorial for Node MCU to prepare your Arduino IDE for Node MCU. Besides adding WiFi capability, the main claim to fame for the ESP8266 processor over the AVR processor of the standard Arduino is that it has a larger 4 MB of Flash memory and runs at clock speeds of 80 MHz and can sometimes optionally be overclocked to 160 MHz and therefore has a very fast processing speed. The Digital I/O all support PWM and interrupts. In addition they can be configured to have pull-up or pull-down resistors. Though there are 11 digital I/O pins, 2 are typically reserved for use as the TX/RX lines if serial communications are used which leaves 9 digital I/O.

On the down-side, it has only 1 analog input which is probably the most significant limitation for some sensor type applications. That can always be overcome by using an external Analog Mux module like our 16-channel 74HC4067 or our ADS1115 4-channel 16-bit ADC module below if more analog I/O is desired. The module can be powered via the USB port or by using an external 7-12V power supply connected to the VIN pin. The module runs at 3.3V, so keep that in mind when working with I/O. The digital I/O is stated as being 5V tolerant, but the analog input needs to be limited to < 3.3V.The blue on-board LED is connected to D0 (GPIO16) and can be accessed using the LED_BUILTIN constant. The module has a typical 'Reset' button as well as a 'Flash' button. The Flash button is used when programming using the original Node MCU firmware. If the module is being used with the Arduino IDE, the Flash button does not need to be used to program the board and it will program just as any Arduino board would.

The module comes loaded with the Node MCU software that accepts the standard AT command set. The module was initially designed primarily to be programmed using Lua which is an interpreted language, but Lua has been mostly abandoned because it takes up a lot of memory and is slow because it is interpreted and more importantly it is generally buggy. It can also be programmed in C using the Arduino IDE and is how the modules are most often used. An example program is shown down below. If a program is download via the IDE, it will overwrite the Node MCU software or whatever else was loaded before. If that is a problem for what you want to do, the Node MCU software can always be reloaded. here are many instructions for installing and using ESP8266 based boards with the Arduino IDE, but here is a short-hand version. Note that once the ESP8266 board type is added to the IDE, there will be many more items added to the Tools drop down menu.

3.4.2 SMPS

The S.M.P.S operated at input regular voltage AC 100V – 240V.
The S.M.P.S should be capable of a total continuous DC power output of 40 Watts.
The S.M.P.S should be capable of a total peak 50 Watts.



Fig-3.4: SMPS

3.4.3 Voltage Sensor

A voltage sensor is a sensor is used to calculate and monitor the amount of voltage in an object. Voltage sensors can determine both the AC voltage and DC voltage level. The input of this sensor can be the voltage whereas the output is the switches, analog voltage signal, a current signal, an audible signal, etc.

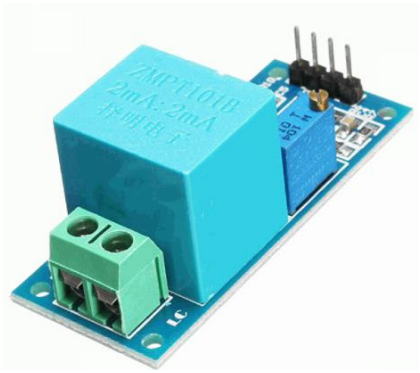


Figure 3.5: Voltage Sensor

Sensors are basically a device which can sense or identify and react to certain types of electrical or some optical signals. Implementation of voltage sensor and current sensor techniques have become an excellent choice to the conventional current and voltage measurement methods.

Specification:

This module is ideally suited to measuring an AC voltage. When connected to the screw terminals the sensor will sample the input and output an analog voltage, in the region of 0 - 5V. This makes it easy to use in power monitoring applications and checking supplies

*Drive Voltage: 5VCharacteristics: Analog output Up to 250V AC detection range Calibration potentiometer.

*Chipset: ZMPT101B.

*Mounting holes: M2 46/16mmPin pitch: 2.54Connectors: Screw Terminals.

*Operating Temperature: -40°C to 85°C (unconfirmed)

3.4.4 LCD Display

LCD20*04, or 20*04 character-type liquid crystal display, is a kind of dot matrix module to show letters, numbers, and characters and so on. It's composed of 5x7 or 5x11 dot matrix positions; each position can display one character. There's a dot pitch between two charactersand a space between lines, thus separating characters and lines. The model 20*04 means it displays 4 lines of 20characters.

Generally, LCD20*04 has parallel ports, that is, it would control several pins at the same time. LCD20*04 can be categorized into eight-port and four-port connections. If the eight- port connection is used, then all the digital ports of the Sun Founder Uno board are almost completely occupied. If you want to connect more sensors, there will be no ports available. Therefore, the four-port connection is used here for better application.

Pin	Function
VSS	connected to ground
VDD	connected to a +5V power supply
VO	to adjust the contrast
RS	A register select pin that controls where in the LCD's memory you are writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD's controller looks for instructions on what to do next.
R/W	A Read/Write pin to select between reading and writing mode
E	An enabling pin that reads the information when High level (1) is received. The instructions are run when the signal changes from High level to Low level.
D0-D7	to read and write data
A	Pins that control the LCD backlight. Connect A to 3.3v.
K	Pins that control the LCD backlight. Connect K to GND.

Tabel.01: LCD display outline

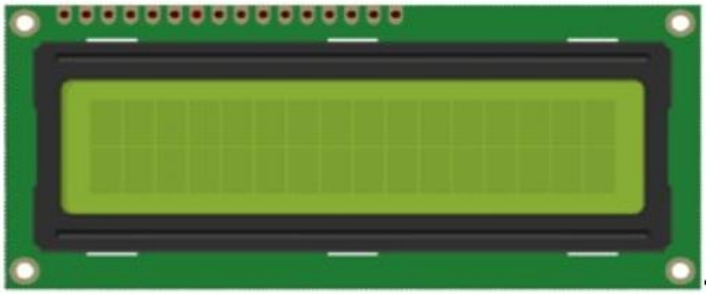


Fig- 3.6: LCD Display

3.4.5 Relay Module

A relay is an electrically operated device. It has a control system and (also called input circuitor input contactor) and controlled system (also called output circuit or output cont. actor). Itis frequently used in automatic control circuit. To put it simply, it is an automatic switch to controlling a high-current circuit with a low-current signal.



Fig-3.7: Relay Module

The advantages of a relay lie in its lower inertia of the moving, stability, long-term reliability and small volume. It is widely adopted in devices of power protection, automation technology, sport, remote control, reconnaissance and communication, as well as in devicesof electro mechanics and power electronics. Generally speaking, a relay contains an inductionpart which can reflect input variable like current, voltage, power, resistance, frequency, temperature, pressure, speed and light etc. It also contains an actuator module (output) which can energize or de-energize the connection of controlled circuit. There is an intermediary partbetween input part and output part that is used to coupling and isolate input current, as well as actuate the output. When the rated value of input (voltage, current and temperature etc.) is above the critical value, the controlled output circuit of relay will be energized or de-energized.

NB: input into a relay can be divided into two categories: electrical quantities (including current, voltage, frequency, power etc.) and non- electrical quantities (including temperature, pressure, speed, etc.

Features:

The features of 1-Channel Relay module are as follow:

1) Good in safety. In power system and high voltage system, the lower current can control the higher one.

2) 1-channel high voltage system output, meeting the needs of single channel control

3) Wide range of controllable voltage.

4) Being able to control high load current, which can reach 240V, 10A

5) With a normally-open (NO) contact and a normally-closed (NC) contacts

Interface specifications:

The output contacts of a relay (including NO, NC, and the common port) works as a SPDT – Single Pole Double Throw switch. Its operating principle is as follow: VCC----5V,

GND--- for ground

IN1 connects to the control valve which output 3V-5V

Output contacts: connect to applications

Interface Connecting and Setting:

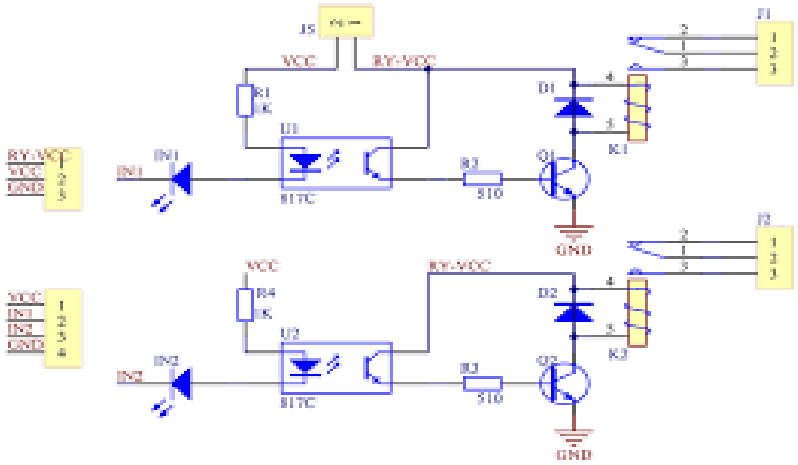


Fig-3.8: schematic Of Relay Module

Get Started:

Firmware resources: Arduino board (any versions), wires, LED, 5v power supply. Software resource: Arduino IDE The one-channel relay can be programmed to realize the open and close automatically. NB: customers can use any software or firmware to control the module as long as the IN1 of which can input a voltage of 3V-5V.

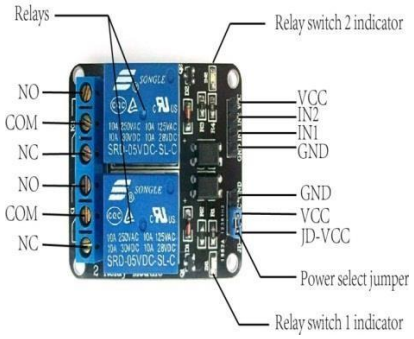


Fig-3.9: Pin out of Relay Module

you can do further development with the development tool you like as well as test it in the way of testing firmware. Firmware test: after the connection as in picture1-4, pay attention to the blink of LED, listen to the flicker of relay when it is working. Software test code:

Note: notice that the module has a jumper cap connecting the VCC and JD-VCC pins; the one shown here is blue, but yours may be a different color. The jumper cap allows you to choose whether the circuit is physically connected to the Arduino circuit or not, and you can choose to have it on or not. With the jumper cap on, the VCC and JD-VCC pins are connected. That means the relay electromagnet is directly powered from the Arduino's power pin, so the relaymodule and the Arduino circuits are not physically isolated from each other (this is the configuration we'll use). Without the jumper cap, you need to provide an independent power source to power up the relay's electromagnet through the JD-VCC pin. That configuration physically isolates the relays from the Arduino with the module's built-in opt coupler.

3.4.6 Current Sensor

This current sensor board is based on the Allegro ACS712ELCTR-30A bi-directional hall- effect current sensor chip that detects positive and negative flowing currents in the range of minus 30 Amps to positive 30 Amps. The board operates at 5V DC and the current flow through the sensor is converted to an output voltage starting at 1/2Vcc (or 2.5V) for no current flow and moves up 66mV per amp for positive current or down -66mV per amp for negative current.



Figure3.10: Current Sensor

Description:

- The current sensor chips: acs712elc-20a;
- Pin 5 V power supply on board power indicator;
- The module can measure the positive and negative 20 amps, corresponding to the analog output of 100mv / a;
- No test current through the output voltage is VCC / 2;
- PCB board size: 31 (mm) x13 (mm);

3.5 Project Implementation Result

In this system we decorate here an IoT based smart energy meter system to monitoring our meter readings from remote area. Here we can control our loads from any area of world and we can monitor the consumed ampere.

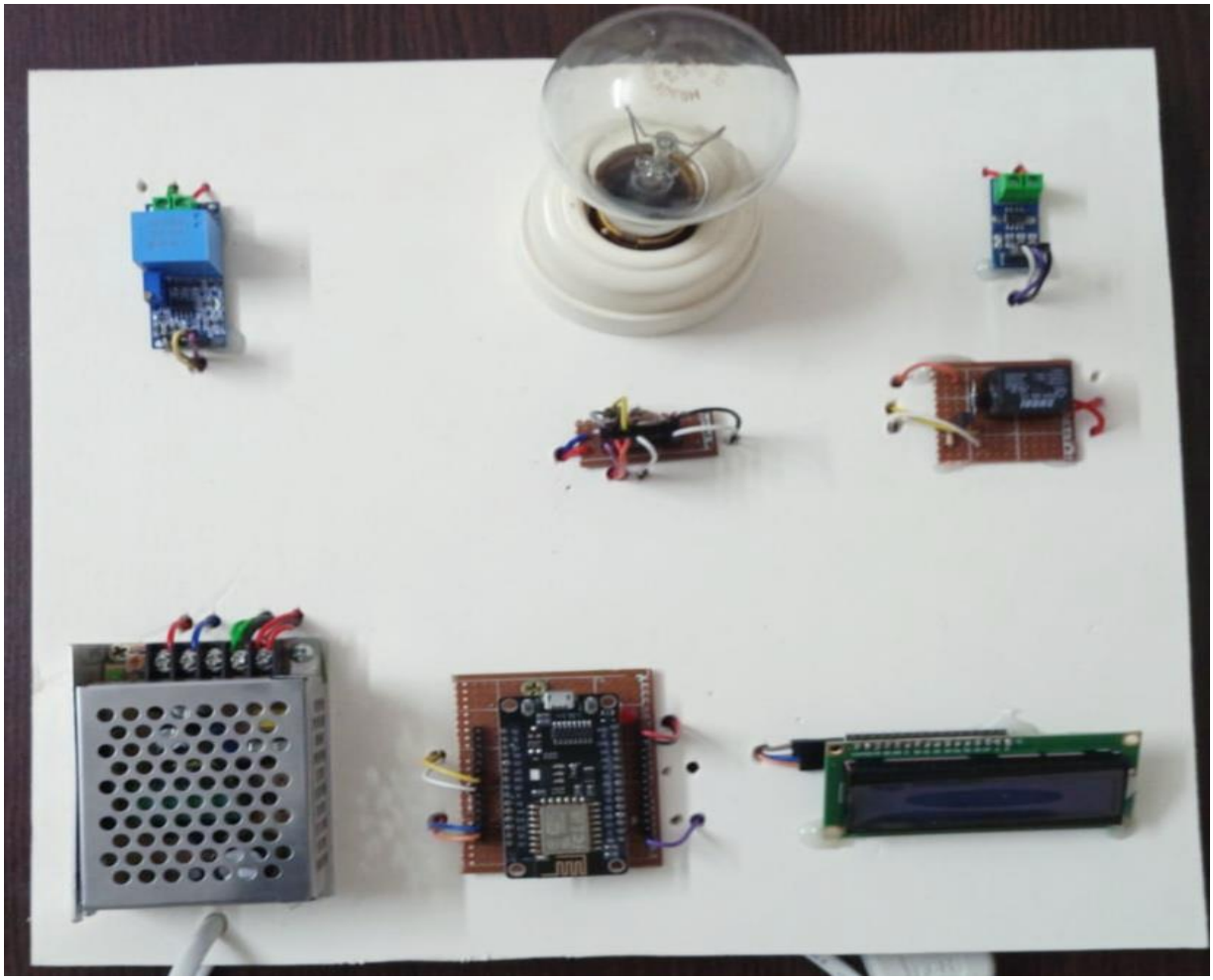


Fig-3.11: Project picture

Here we use a Node MCU to control this total system by using given program. In this system we use here blynk app to monitoring our total system using IoT. To decorate the base of our project we use here sun board. And then all the components are added here on the board. By drilling the board we connect all components each with others using wires. One SMPS is used here to make sure the power supply for our all components. In this system we measure here Current value, Voltage value, monthly used power value, and total cost based on used power as per rating of per unit electricity.

CHAPTER 4

SOFTWARE AND SIMULATION

4.1 Introduction

Software part is one of the main part of the Project. The Algorithm is based on different conditions & measuring parameters. Total function of the system is controlled by thesoftware. The code is written on C.

4.2 Arduino IDE

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as **Windows, Mac OS X, and Linux**. It supports the programming languages C and C++. Here, IDE stands for **Integrated Development Environment**.

The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'

The Arduino IDE will appear as:

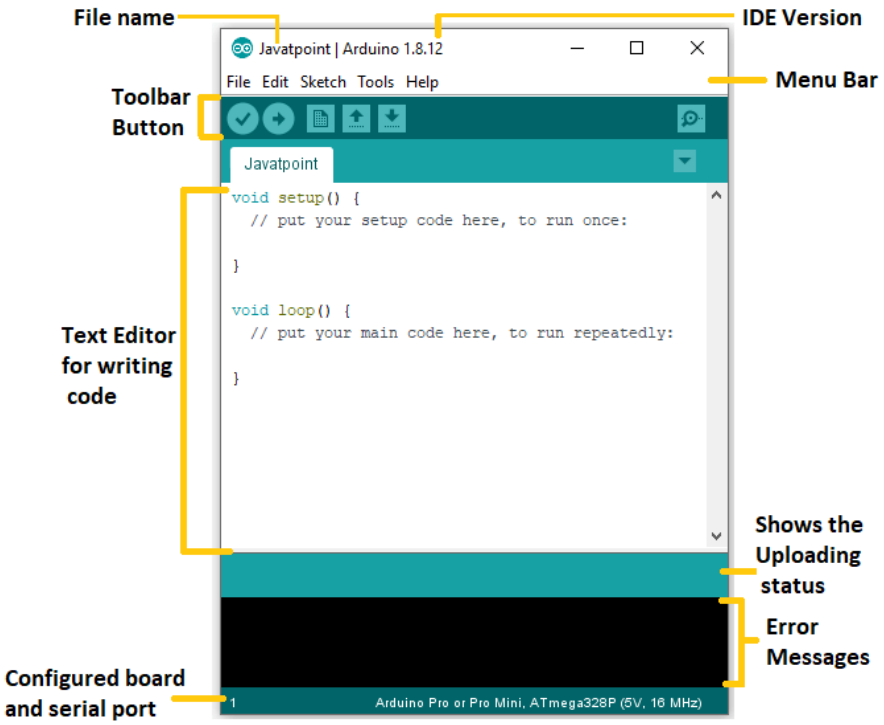


Figure 4.1: Java Point

Toolbar Button

The icons displayed on the toolbar are **New, Open, Save, Upload,** and **Verify**.

It is shown below:

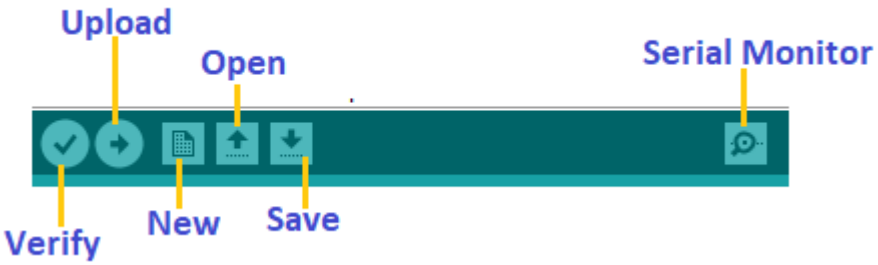


Figure 4.2: Toolbar Button

Upload

The Upload button compiles and runs our code written on the screen. It further uploads the code to the connected board. Before uploading the sketch, we need to make sure that the correct board and ports are selected.

We also need a USB connection to connect the board and the computer. Once all the above measures are done, click on the Upload button present on the toolbar.

The latest Arduino boards can be reset automatically before beginning with Upload. In the older boards, we need to press the Reset button present on it. As soon as the uploading is done successfully, we can notice the blink of the Tx and Rx LED.

Page Setup

It allows setting the page margins, orientation, and size for printing. The '**Page Setup**' window will appear as:

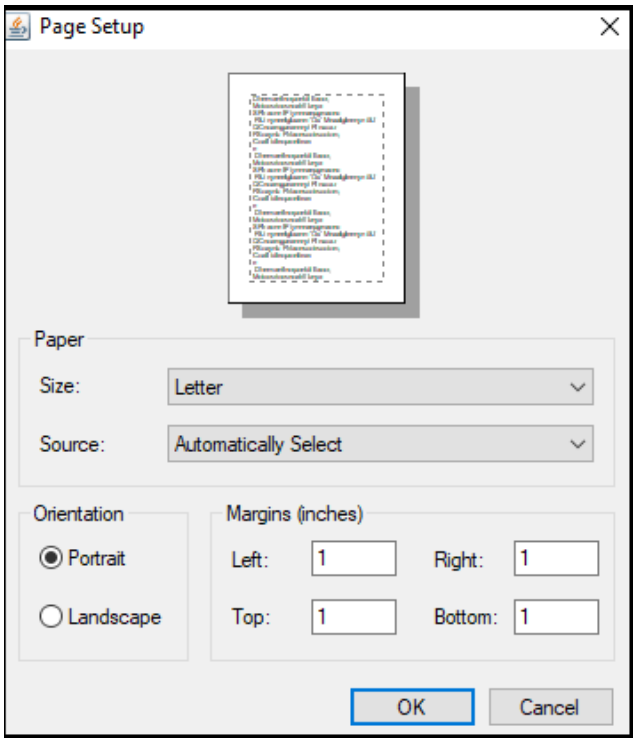


Figure 4.3: Page Setup

Print

According to the settings specified in the 'Page Setup', it prepares the current sketch for printing.

Preferences

It allows the customization settings of the Arduino IDE.

Quit

The Quit button is used to close all the IDE windows. The same closed sketch will be reopened when we will open the Arduino IDE.

- **Edit**

When we click on the Edit button on the Menu bar, a drop-down list appears. It is shown below:

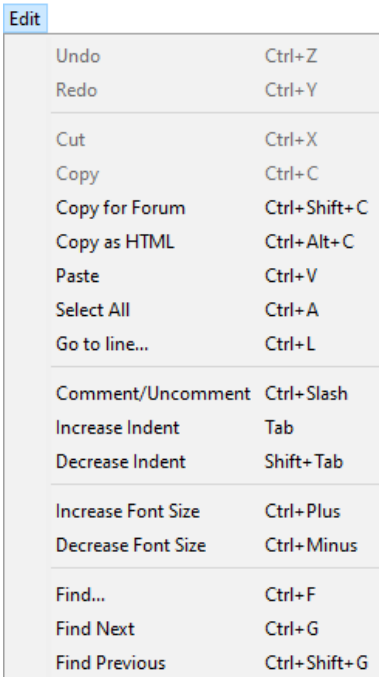


Figure 4.4: Edit

Let's discuss each option in detail.

Undo

The Undo button is used to reverse the last modification done to the sketch while editing.

Redo

The Redo button is used to repeat the last modification done to the sketch while editing.

Cut

It allows us to remove the selected text from the written code. The text is further placed to the clipboard. We can also paste that text anywhere in our sketch.

Copy

It creates a duplicate copy of the selected text. The text is further placed on the clipboard.

Upload

The Upload button is used to configure the code to the specified board through the port.

Upload Using Programmer

It is used to override the Boot loader that is present on the board. We can utilize the full capacity of the Flash memory using the '**Upload Using Programmer**' option. To implement this, we need to restore the Boot loader using the **Tools**-> **Burn Boot loader** option to upload it to the USB serial port.

Export compiled Binary

It allows saving a .**hex** file and can be kept archived. Using other tools, .hex file can also be sent to the board.

Show Sketch Folder

It opens the folder of the current code written or sketch.

Include Library

Include Library includes various Arduino libraries. The libraries are inserted into our code at the beginning of the code starting with the #. We can also import the libraries from .zip file.

Add File...

The Add File... button is used to add the created file in a new tab on the existing file.

For example, let's add '**Blink**' file to the '**Java point**' file. The tab will now appear as:

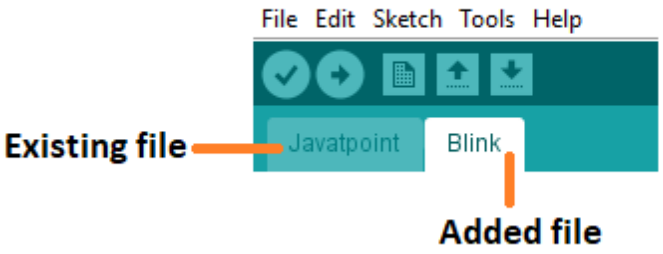


Figure 4.5: Added file

We can also delete the corresponding file from the tab by clicking on the **small triangle** -> **Delete** option.

Tools

When we click on the Tools button on the Menu bar, a drop-down list appears. It is shown below:

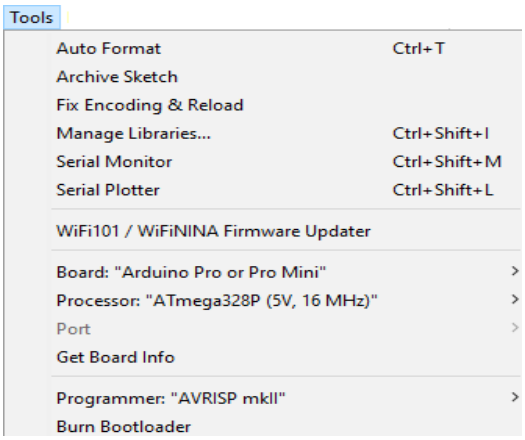


Figure 4.6: Tools

Let’s discuss each option in detail.

Auto Format

The Auto Format button is used to format the written code. For example, lining the open and closed curly brackets in the code.

Archive Sketch

The copy of the current sketch or code is archived in the .zip format. The directory of the archived is same as the sketch.

Fix Encoding and Reload

This button is used to fix the inconsistency between the operating system char maps and editor char map encoding.

Manage Libraries…

It shows the updated list of all the installed libraries. We can also use this option to install a new library into the Arduino IDE.

Serial Monitor

It allows the exchange of data with the connected board on the port.

Serial Plotter

The Serial Plotter button is used to display the serial data in a plot. It comes preinstalled in the Arduino IDE.

WiFi101/WiFi NINA Firmware Updater

It is used to check and update the Wi-Fi Firmware of the connected board.

Board

We are required to select the board from the list of boards. The selected board must be similar to the board connected to the computer.

Processor

It displays the processor according to the selected board. It refreshes every time during the selection of the board.

Port

It consists of the virtual and real serial devices present on our machine.

Get Board Info

It gives the information about the selected board. We need to select the appropriate port before getting information about the board.

Programmer

We need to select the hardware programmer while programming the board. It is required when we are not using the onboard USB serial connection. It is also required during the burning of the Boot loader.

Burn Boot loader

The Boot loader is present on the board onto the microcontroller. The option is useful when we have purchased the microcontroller without the boot loader. Before burning the boot loader, we need to make sure about the correct selected board and port.

Help

When we click on the Help button on the Menu bar, a drop-down list will appear. It is shown below:

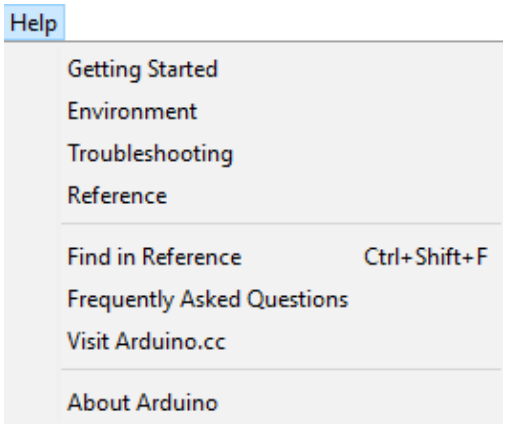


Figure 4.7: Help

The Help section includes several documents that are easy to access, which comes along with the Arduino IDE. It consists of the number of options such as Getting Started, Environment, Troubleshooting, Reference, etc. We can also consider the image shown above, which includes all the options under the Help section. Some documents like Getting started, Reference, etc., can be accessed without the internet connection as well. It will directly link us to the official website of Arduino.

4.3 Proteus 8 Professional

This is a brief introduction to Proteus application. The special features and advantages of this application will be immediately below. Hopefully this article will give you the most extensive and useful view on how to use Proteus 8 Professional.

Want to check the circuit diagram? Seeing circuit layout seems like a good choice but it is not easy to do it for large circuit boards. Or you can try creating a circuit diagram (PCB) printout and testing but it takes a lot of time-consuming operations. It seems, using Proteus 8 Professional is the most ideal. Proteus 8 Professional is a software used to plot, sketch PCB and simulate diagrams.

This software was developed by **Lab center Electronic Ltd**.

Features of Proteus 8 Professional

Very easy to draw diagrams on Proteus 8 Professional. Click "select device" and select the desired detail, you can draw the wire by clicking on the output of electricity or Vcc, Ground, etc.
Is the circuit completed? The simulation feature of Proteus 8 Professional can display most details in the circuit. There are two options for this feature: run the simulation and run each table. "Running simulation" shows circuit board at normal speed (if the circuit is not too complicated). "Running each table" will run each circuit board at every click. This customization is useful for fixing digital circuits. You can simulate microcontrollers. Depending on each microcontroller will be outlined based on PIC24, dsPIC33, 8051, Arduino, or ARM7. You can download the compiler and dump hex file into the microcontroller contained in Proteus. Moreover, real-time integration with simulation using switches, resistors, photoelectric resistors, etc.

Is an easy-to-use feature in Proteus. You can create your own designs or make Proteus work for you. Creating your own design is easy, you just need to put the details into the diagram and draw the electrical circuit through.

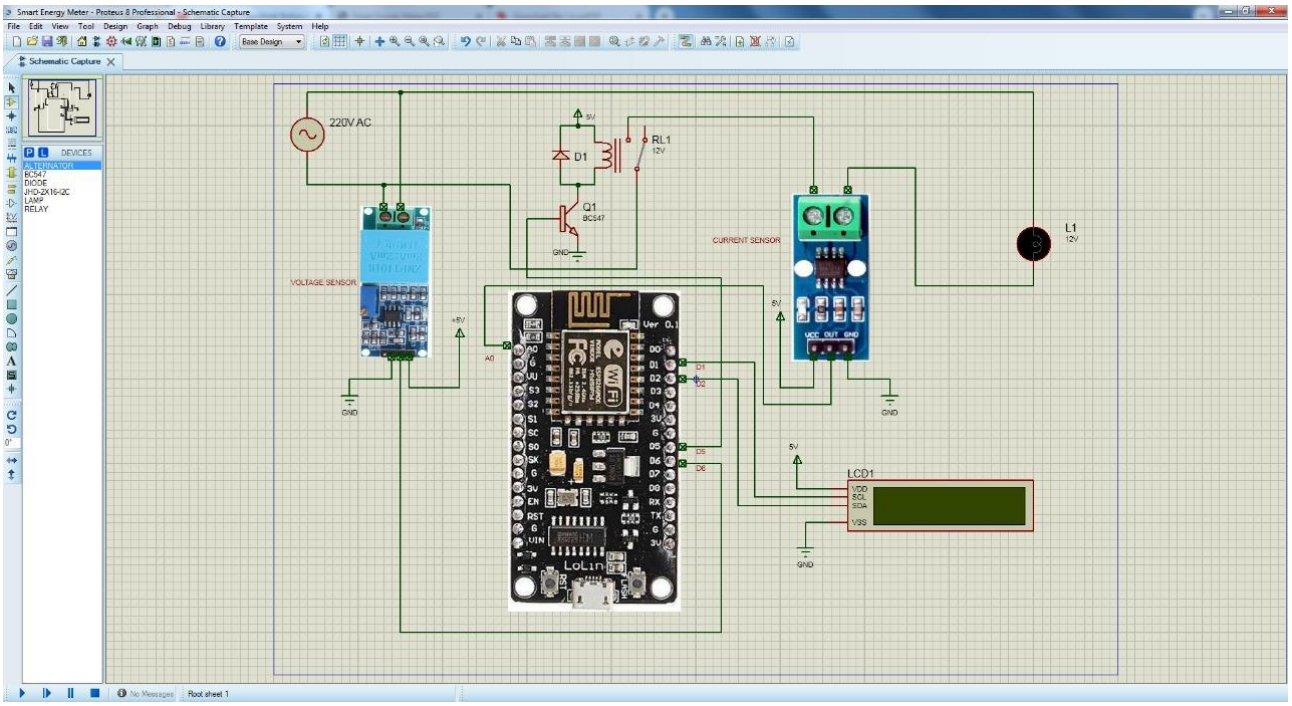


Figure 4.8: Schematic Capture

Don't worry about breaking any design rules because it will automatically detect errors. If you want Proteus to do it for you, just put the details in the corresponding position and run it automatically. It will draw the way to set the circuit and choose the best version. And now there's another custom "Auto placer", which requires you to set the board size by drawing the shape and size of the board. After that, it automatically puts the details into the mold. After that, all you have to do is make a circuit diagram.

Pros and cons Proteus 8 Professional

Advantages:

- You can interact with test run simulations, use switches, etc.
- There are virtual instruments available for measuring electricity.

Defect:

- User interface should be better.

4.4 Blynk App

With Blynk, you can create smartphone applications that allow you to easily interact with microcontrollers or even full computers such as the Raspberry Pi. The main focus of the Blynk platform is to make it super-easy to develop the mobile phone application. As you will see in this course, developing a mobile app that can talk to your Arduino is as easy as dragging a widget and configuring a pin. With Blynk, you can control an LED or a motor from your mobile phone with literally zero programming. This is actually the first experiment that I will demonstrate in this course. But don't let this simplicity make you think that Blynk is only useful for trivial applications. Blynk is a robust and scalable tool that is used by hobbyists and the industry alike.

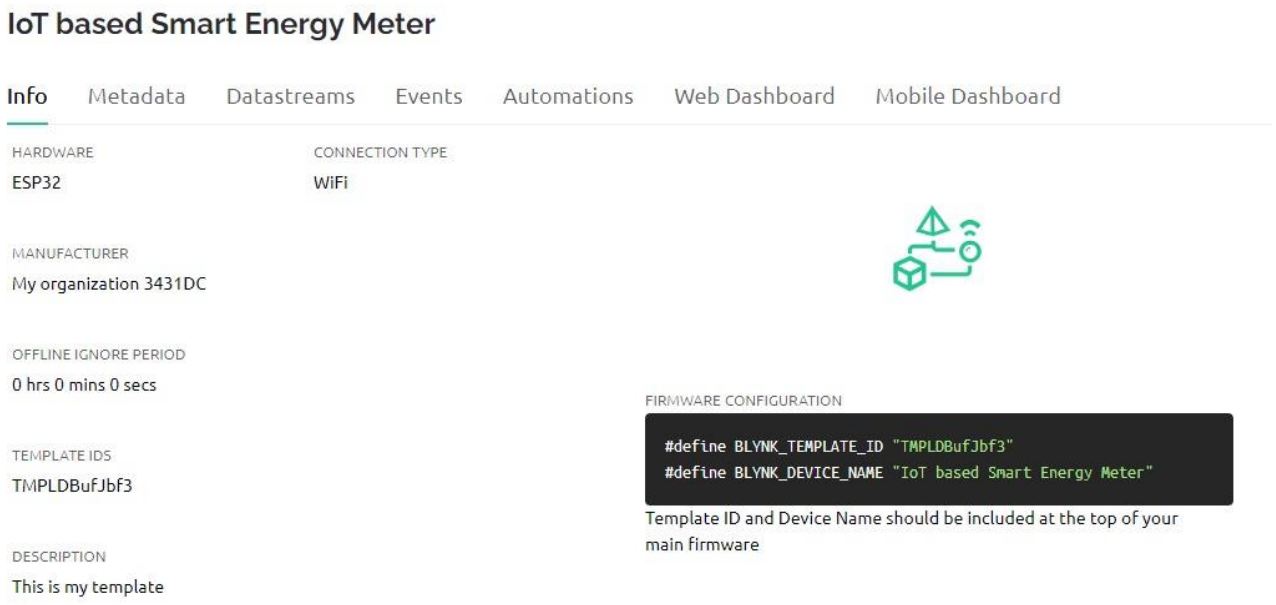


Figure 4.9: Biynk Info

You can use it to monitor the soil humidity of your vegetable garden and turn on the water, or open up your garage door, with your phone. You can also use it to control smart furniture that can learn

from your routines, or embed IoT and AI to traditional industrial products such as a boiler, or for improving the integrity and safety of oilfields. Blynk is free to use for personal use and prototyping. Their business model generates profits by selling subscriptions to businesses that want to publish Blynk-powered apps for their hardware products or services.

The Blynk innovation is the smartphone app. The Blynk app is is really an app editor. It allows you to create one or more projects. Each project can contain graphical widgets, like virtual LEDs, buttons, value displays and even a text terminal, and can interact with one or more devices.

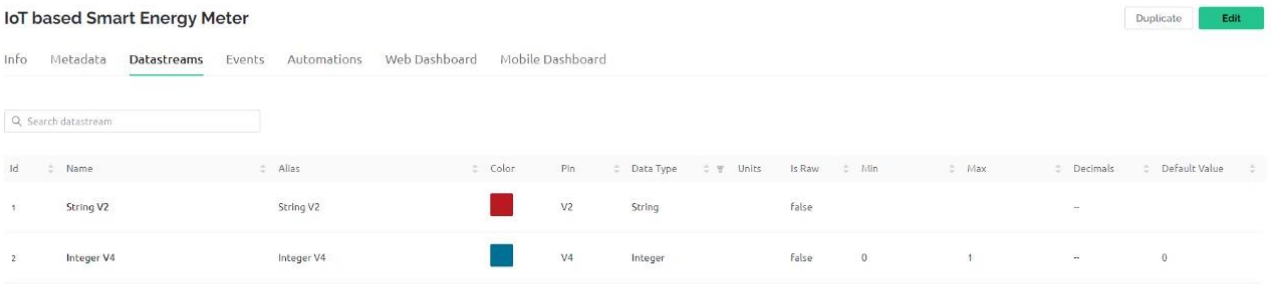


Figure 4.10: DataStream

With the help of the Blynk library, it is possible to control Arduino or ESP32 pins directly from your phone, without having to write any code at all. It is also possible to share a project with friends and even customers so that they can access the connected devices but not be able to modify the project. Imagine a scenario where you build a smartphone application where you can control lights, window blinds and room temperature from your phone. You can share the project with other family members so that they can also access the functionality.

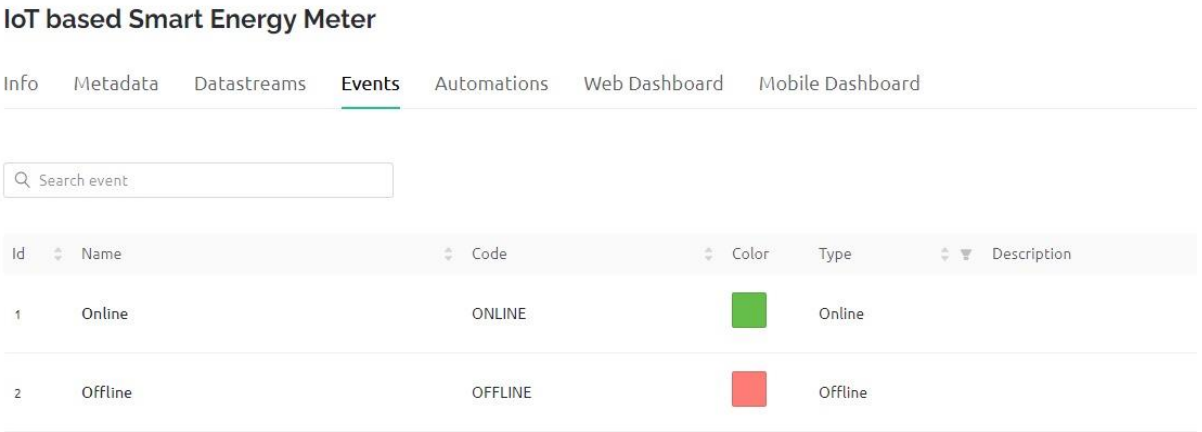


Figure 4.11: Events

I am amazed by the range of devices and connectivity types that the Blynk platform supports. The support is implemented by means of a Blynk library that targets a device and connectivity type combination. For example, if you want to use your Arduino Uno with an Ethernet shield, you would use the library "BlynkSimpleEthernet" which contains the Blynk firmware plus the required connectivity support.

Both sketches would use the same Blynk infrastructure, such as the physical pins that are part of the Arduino's hardware, as well as the virtual pins which are implemented in software by the Blynk Platform. This means that with a bit of planning, you can write sketches that can be easily shared among different target devices. You can potentially write a sketch for an Arduino Uno and and with minimal modification run it on an Arduino MKR1010. Blynk also supports clients that are not a microcontroller. You can write client code is JavaScript, Python or Lua thanks to the available Blynk libraries for these languages. This means that you can have a Blynk project that interacts with an Arduino and with Python code running on a Linux virtual machine somewhere on the Cloud. The possibilities are really very exciting.

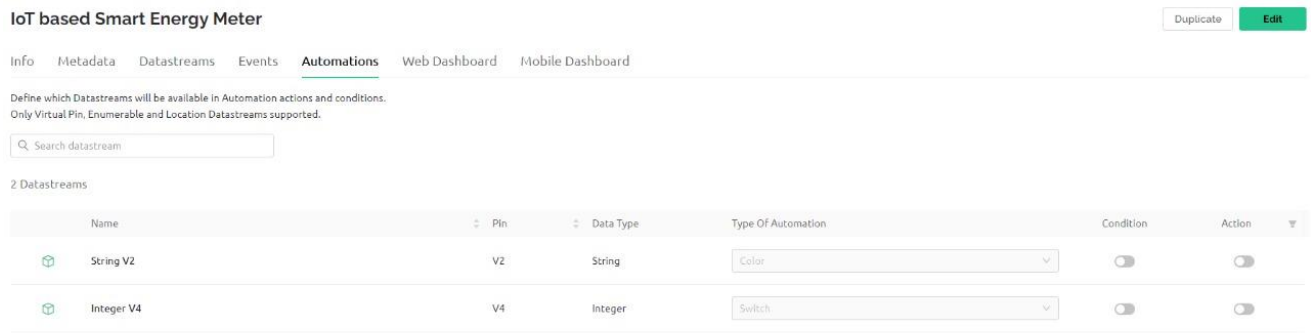


Figure 4.12: Automations

Unlike IoT platform such as IFTTT, Twillio, and even Adafruit IO, you can host a private instance of the full Blynk server and connect your smartphone Blynk app to it. The Blynk Cloud server is an excellent choice for most projects, as it is always there, ready to use. We will use the Cloud server in the first few experiments in this course to help you get started with minimal effort. However, as you will see, the Cloud Blynk server has imposed limitations. Some limitations are due to the topology of the server: depending on your geographical location, the server may be in a different continent, which makes communications between the app, the devices and the server slow due to the amount of time it takes packets to travel across the Internet.

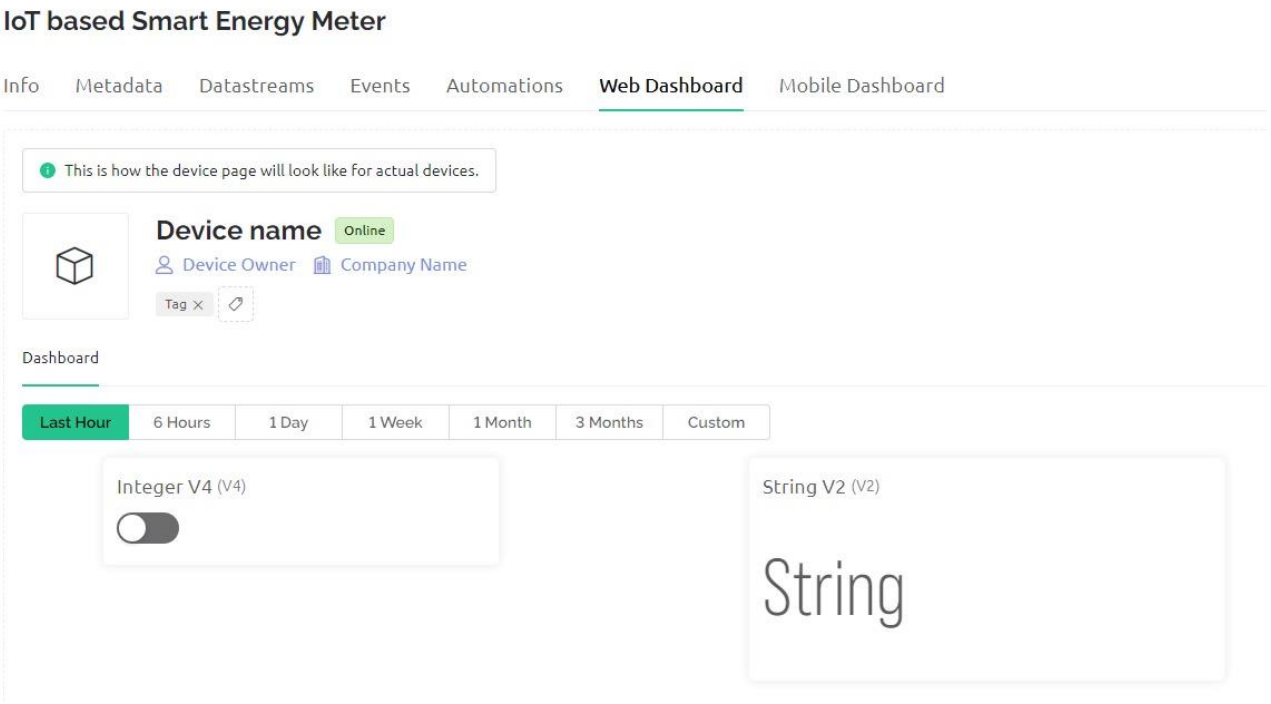


Figure 4.13: Web Dashboard

Another imposed limitation is that in the Cloud server, you can only use a small number of widgets. Blynk is using the concept of "energy" to implement a pricing system for its widgets. In the Cloud server you may start a new project with 1000 energy units.



Figure 4.14: New Device Created

An LED widget may cost you 200 units, leaving 800 units for other widgets. On a private server, you can set your own energy limits. You can configure your server to allocate 100,000 energy units to new users. It's totally up to you.

Of course, you can purchase additional energy units to spent on the Cloud Blynk server, and this is a legitimate consideration especially if the users of your Blynk project are distributed around the world (therefore latency is not a big issue).

However, a private Blynk server gives you additional benefits:

- Essentially unlimited energy units, so that you can build any Blynk application you can think of.
- Minimal latency, which is useful when your application is used in a limited geographic area and responsiveness is important.
- Total control of your data. You can keep your own backups of your Private server, migrate your server to a new host, implement whichever security mechanisms you wish, and finely control your users.

In this course, I will show you how to install and configure a private Blynk server so you can see the related benefits and costs. You can run an instance of the Blynk server on any host that has a Java runtime environment. In this course, I use a $15 Raspberry Pi Zero W, running the Raspberry Pi operating system.

This is a computer with just half a gigabyte of RAM, and still runs the Blynk server without any issues at all. You can use any Raspberry Pi with WiFi, it does not need to be a Zero W. In fact, you can use any Linux or Windows host you want, including virtual machines.

CHAPTER 5

DISCUSSIONS AND CONCLUSIONS

5.1 Discussions

Intelligent operations in power system infrastructure are crucial needs of modern grid systems. PG is a new and improved grid that solves various problems of efficiency and reliability in the traditional grid. In this paper, the implementation of an IoT based power monitoring and controlling prototype is presented. Blink apps is used as a software tool to access consumer load data at the remote end. The large scale installation of proposed design requires to developed cost-effective power sensing and monitoring devices that can be easily integrated into the consumer premises. In the future, authors are interested to develop a cloud-based smart metering system for the deployment in smart cities.

5.2 Conclusions

A revolution in energy domain is underway, namely the IoT Based Smart Energy Meter System. In this system every person can monitor there meter reading using smart phone at any time and from any location. This paper presents the design and programming for an energy monitoring system. The system was able to display the energy consumption in real time. The objective to design a system that can monitor energy consumption and store energy data in the cloud was successfully tested and achieved. Moreover, users would be able to monitor all the energy consumption using a mobile application or through a webpage, since the system could check on both applications. For future recommendations, we would recommend using ESP32 or the other microcontrollers that have a Wi-Fi connection and extra analogue inputs. This would allow the system to add more current sensors to get the data from multiple consumption loads.

5.3 Advantages

Benefits of smart meter over traditional energy meters:

- Smart meters are having less chances to have error as compared to the traditional energy meter
- The electricity board officials can identify easily about the tampering of the electricity within a premise.
- Smart meters when programmed with home appliances can be used to control the electricity consumption.
- There is no need of electricity board officials to be physically present at the site as this work would be done by the smart energy meter by using the IOT technique.

This system enables the electricity department to read the meter readings without a person visiting each house. This can be achieved by the use of Arduino unit that continuously monitor and records the energy meter reading in its permanent memory location and display the reading onto the server using the Wi-Fi module. This system continuously records the reading and the live meter reading can be displayed on webpage to the consumer on request.

5.4 Future works

The project is focused on the government's plan to turn the major cities of the country into smart cities. The project provides the entire energy readings at one's finger tips. The project can be further extended to detect the energy meter tampering. A smart app can be designed to provide various alerts based on the readings from the device. A unified can be provided to the customers for both viewing the energy usage and a platform to pay the bill online following the digital India initiative. In one case the service provider can evaluate the bills which are not paid and can disconnect the energy connection remotely.

References:

[1] Sasanenikita N, “IOT based energy meter billing and monitoring system”, International research journal of advanced engineering and science, (2017).

[2] Pandit S, “Smart energy meter using internet of things (IOT)”, VJER vishwakarma Journal of engineering Research, Vol.1, No.2, (2017).

[3] Maitra S, “Embedded Energy Meter-A New concept to measure the energy consumed by a consumer and to pay the bill”, Power system technology and IEEE power india conference, (2008).

[4] Stamatis Depuru SSSR, Wang L & Devabhaktuni V, “Electricity theft: Overview, issues, prevention and a smart meter based approach to control theft”, Energy Policy, Vol.39,No.2, (2011), pp.1007-1015.

[5] Giri Prasad S, “IOT based energy meter”, International Journal of recent trends in engineering& research (IJRTER), (2017).

[6] Sehgal VK, Panda N, Handa NR, Naval S & Goel V, “Electronic Energy Meter with instant billing”, Fourth UKSim European Symposium on Computer Modeling and Simulation (EMS), (2010), pp.27-31.

[7] Ashna K & George SN, “GSM based automatic energy meter reading system with instant billing”, Proceeding of international multi conference on automation, computing, communication, control and compressed sensing (Imac4S),Vol.65,No.72, (2013), pp.22-23.

[8] Malik NS, kupzog F & Sonntag M, “An approach to secure mobile agents in automatic meter reading”, IEEE International Conference on Cyber worlds, computer society, (2010), pp.187-193.

[9] Praveen MP, “KSEB to introduce SMS-based fault maintenance system”, The hindu news, (2011).

[10] G Ainabekova, Z Bayanbayeva, B Joldasbekova, A Zhaksylykov (2018). The author in esthetic activity and the functional text (on the basis of V. Mikhaylov’s narrative (“The chronicle of the great jute”). Año 33. 63-80.

[11] Z Yesembayeva (2018). Determination of the pedagogical conditions for forming the readiness of future primary school teachers, Opción, Año 33. 475-499

[12] J. Widmer, Landis," Billing metering using sampled values according lEe 61850-9-2 for substations",IEEE 2014.

[13] Cheng Pang,Valierry Vyatkin,Yinbai Deng, Majidi Sorouri, "Virtual smart metering in automation and simulation of energy efficient lightning system" IEEE 2013.

[14] Amit Bhimte, Rohit K.Mathew, Kumaravel S, "Development of smart energy meter in labview for power distribution systems", "IEEE INDICON 2015 1570186881", 2015.

[15] H. Arasteh, V. Hosseinnezhad, V.Loia, A.Tommasetti, O.Troisi, M.Shafie Khan, P.Siano, "IoT Based Smart Cities: A survey"IEEE 978-1-5090-2320-2/1631.00,2016.

[16] Clement N. NYIRENDRE, Irvine NYANDOWE, Linda SHITUMBAPO, "A comparison of the collection tree protocol (CTP) and AODV routing protocol for a smart water metering.", PP NO. 1-8,2016.

[17] Yaacoub E & Abu-Dayya A, "Automatic meter reading in the smart grid using contention based random access over the free cellular spectrum", computer networks, (2014).

APPENDIX

Program:

```
#define BLYNK_TEMPLATE_ID "TMPLDBufJbf3"
#define BLYNK_DEVICE_NAME "IoT based Smart Energy Meter"
#define BLYNK_AUTH_TOKEN "qhDM76hmwJwQrxVLDIfN2jPx2I_cNkud"

#include <ESP8266WiFi.h>
#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27 ,16 , 2);

char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "Engineers Butty Lab";
char pass[] = "25802580";

int load1 = D5;

const int sensorIn = A0;
int mVperAmp = 10; // use 100 for 20A Module and 66 for 30A Module
double Voltage = 0;
double VRMS = 0;
double AmpsRMS = 0;
int watt=0;
int volt = 220;
double kwh=0;
int bill = 0;
float countt =0;
int count = 0 ;
```

```
BLYNK_WRITE(V4)
{
  int pinValue1 = param.asInt(); // assigning incoming value from pin V1 to a variable
  digitalWrite(D5, pinValue1);
Serial.println(pinValue1);

}


WidgetLCD lcd1(V2);

void setup()
{
  Serial.begin(115200);
 pinMode(D5 , OUTPUT);

lcd.backlight();
 lcd.init();
 lcd.clear(); //Use it to clear the LCD Widget
  lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("IoT based Smart");
lcd.setCursor(0,1);
lcd.print("Energy Meter");
delay(2000);
 lcd.clear(); //Use it to clear the LCD Widget

pinMode(sensorIn , INPUT);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

}

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

```
iot_all();

  float result;
  int readValue;             //value read from the sensor
  int maxValue = 0;          // store max value here
  int minValue = 1024;          // store min value here

   uint32_t start_time = millis();
   while((millis()-start_time) < 1000) //sample for 1 Sec
   {
      readValue = analogRead(sensorIn);
      // see if you have a new maxValue
      if (readValue > maxValue)
      {
         /*record the maximum sensor value*/
         maxValue = readValue;
      }
      if (readValue < minValue)
      {
         /*record the maximum sensor value*/
         minValue = readValue;
      }
   }

    // Subtract min from max
    result = ((maxValue - minValue) * 5.0)/1024.0;

  Voltage = result;
  VRMS =(Voltage/2.0) *0.707;
  AmpsRMS = ((VRMS * 1000)/mVperAmp)/20;
watt=AmpsRMS*volt;



if(AmpsRMS > .17){
lcd.setCursor(8,0);
lcd.print("Amp:");
```

```
lcd.setCursor(12,0);
lcd.print(AmpsRMS,8);
////////////////// lcd display print uporer tuku/////////////////
//////////////////lcd print nicher tuku online a ///////////
lcd1.print(8,0,"Amp:");
lcd1.print(12,0,AmpsRMS);
}

if(AmpsRMS < .17){
lcd.setCursor(8,0);
lcd.print("Amp:");
lcd.setCursor(12,0);
lcd.print("0.000");
////////////////// lcd display print uporer tuku/////////////////
//////////////////lcd print nicher tuku online a ///////////
lcd1.print(8,0,"Amp:");
lcd1.print(12,0,"0.000");

}
delay(50);
}
void iot_all()
{
  lcd.setCursor(0,1);
  lcd.print("KWH:");
  lcd.setCursor(5,1);
  lcd.print("254");
  lcd1.print(0,1,"KWH: ");
  lcd1.print(5,1,"254");

    lcd.setCursor(9,1);
  lcd.print("TK:");
  lcd.setCursor(12,1);
  lcd.print("1705");
  lcd1.print(9,1,"TK: ");
  lcd1.print(12,1,"1705");
```

```
  lcd1.print(0,0,"V:");
  lcd1.print(3,0,"220");
  lcd.setCursor(0,0);
  lcd.print("V:");
  lcd.setCursor(3,0);
  lcd.print("220");
delay(500);

  lcd.setCursor(0,1);
  lcd.print("KWH:");
  lcd.setCursor(5,1);
  lcd.print("254");
  lcd1.print(0,1,"KWH: ");
  lcd1.print(5,1,"254");

    lcd.setCursor(9,1);
  lcd.print("TK:");
  lcd.setCursor(12,1);
  lcd.print("1705");
  lcd1.print(9,1,"TK: ");
  lcd1.print(12,1,"1705");

  lcd1.print(0,0,"V:");
  lcd1.print(3,0,"221");
  lcd.setCursor(0,0);
  lcd.print("V:");
  lcd.setCursor(3,0);
  lcd.print("221");
delay(500);

  lcd.setCursor(0,1);
  lcd.print("KWH:");
  lcd.setCursor(5,1);
  lcd.print("254");
  lcd1.print(0,1,"KWH: ");
  lcd1.print(5,1,"254");
```

```
    lcd.setCursor(9,1);
  lcd.print("TK:");
  lcd.setCursor(12,1);
  lcd.print("1705");
  lcd1.print(9,1,"TK: ");
  lcd1.print(12,1,"1705");

  lcd1.print(0,0,"V:");
  lcd1.print(3,0,"220");
  lcd.setCursor(0,0);
  lcd.print("V:");
  lcd.setCursor(3,0);
  lcd.print("220");
delay(500);

  lcd.setCursor(0,1);
  lcd.print("KWH:");
  lcd.setCursor(5,1);
  lcd.print("254");
  lcd1.print(0,1,"KWH: ");
  lcd1.print(5,1,"254");

    lcd.setCursor(9,1);
  lcd.print("TK:");
  lcd.setCursor(12,1);
  lcd.print("1705");
  lcd1.print(9,1,"TK: ");
  lcd1.print(12,1,"1705");

  lcd1.print(0,0,"V:");
  lcd1.print(3,0,"219");
  lcd.setCursor(0,0);
  lcd.print("V:");
  lcd.setCursor(3,0);
  lcd.print("219");
delay(500);
```

```
}
```