IoT Based Power Grid Controlling and Monitoring System



SONARGAON UNIVERSITY (SU)

Supervised By

Md. Ashikur Rahman

Lecturer
Department of EEE
Sonargaon University (SU)

Submitted By

Name:	**ID No:**
Md. Tushar Imran	EEE1901016191
Md. Anis Ahmed	EEE1901016194
Md. Alamin	EEE1901016198
Md. Alamin Shaikh	EEE1901016208

Department of Electrical & Electronic Engineering (EEE)
Sonargaon University (SU)
147/I, Panthopath,
Dhaka-1215, Bangladesh.

Date of submission: September 2022

Candidate's 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.

…… ……..

Md. Tushar Imran **Md. Anis Ahmed**

…… …….

Md. Alamin **Md. Alamin Shaikh**

Under Supervision of

Md. Ashikur Rahman
Lecturer, EEE
Sonargaon University (SU)

Certificate

This is to certify that the project paper on “IoT Based Power Grid Controlling and Monitoring System” is the bona fid record of project work done by Name: Md. Tushar Imran ID: EEE1901016191 Section:16C2 Name: Md. Anis Ahmed ID: EEE1901016194 Section:16C2 Name: Md. Alamin ID: EEE1901016198 Section:16C2 Name: Md. Alamin Shaikh ID: EEE1901016208 Section :16C2 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.

Supervisor

……

Md. Ashikur Rahman
Lecturer, EEE
Sonargaon University (SU)
Department of Electrical and Electronic Engineering (EEE)

Acknowledgements

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. Ashikur Rahman, Lecturer,** Department of Electrical and Electronics Engineering, Sonargaon University, for his guidance throughout all phases of our project. We would like to thank him for his cordial suggestion which helped us to reach our goal.

We would also like to thank all 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

Abstract

This project describes the digitization of load energy usage readings over the internet. The proposed system design eliminates the involvement of human in electricity maintenance. The user can monitor energy consumption in watts from a apps by providing a channel id for the load. Wi-Fi unit performs IOT operation by sending energy data of the load to the apps which can be accessed through the channel id of the device.

In the proposed system, consumer can do power management by knowing energy usage time to time. This proposed system utilizes an Arduino microcontroller. The unit which is generated can be displayed on the apps through the Wi-Fi module. Smart grid is one of the features of smart city model. It is energy consumption monitoring and management system. Smart grids are based on communication between the provider and consumer. One of the main issues with today's outdated grid deal with efficiency. The grid becomes overloaded during peak times or seasons. It is also possible to hack the system, and basically, take free electricity. By using smart grid consumer and owner get daily electricity consumption reading and owner can cut electricity supply remotely through internet if bill is not paid.

One more thing, the data collected from the smart meters should not be accessed by any unauthorized entities. We can monitor from anywhere through the internet. We can also do this if we want to turn a system on or off. In case meter tempering is happened then owner and consumer get message and then owner take the action accordingly. Fitting the circuit on customer's energy meter, from that energy consumption data can be acquired. After acquiring of data, that data can be updated on cloud service, so that consumer and provider can access that data through internet.

Contents

Name of the Content	Page No
Candidate's Declaration	ii
Certificate	iii
Acknowledgement	iv
Abstract	v
Contents	vi
List of Figures	vii
List of Tables	ix
Chapter 1: Introduction	
1.1 Introduction	1
1.2 Objective	1
1.3 Background of the Project	2
1.4 Internet of things	2
1.5 IoT System	2
1.6 Methodology	4
1.7 Motivation for Project	4
Chapter 2: Literature Review	
2.1 Introduction	5
2.2 Literature Survey	6
Chapter 3: Hardware Implementation	
3.1 Introduction	7
3.2 Block Diagram of this Project	7
3.3 Circuit Diagram of this Project	8
3.4 The list of Devices used in the Project	8
3.5 Node MCU	9
3.6 SMPS	16

3.7 Optocoupler 16
3.8 Voltage Sensor 17
3.9 LCD Display 18
3.10 Relay Module 20
3.11 LM2596 Voltage Converter 23
3.12 Current Sensor 23
3.13 Project Implementation Result 24
3.14 Project Picture 25
3.15 List of Components with Price 26

Chapter 4: Requirement Analysis, Design, and Development

4.1 Introduction 27
4.2 Software Tools 27
4.3 Programming 28
4.4 Program Developments 29
4.5 Proteus 8.1 29
4.6 Android Apps 30
4.7 Project Result 31

Chapter 5: Discussion and Conclusion

5.1 Discussion 33
5.2 Conclusion 33
5.3 Advantages 33
5.4 Applications 33
5.5 Future Works 34

References 34
Appendix 34

List Of Figures

Figure No	Figure Contain	Page No
Figure 1.1	IoT System	3
Figure 1.2	Block Diagram of IoT	3
Figure 1.3	Methodology	4
Figure 3.1	Block Diagram of the Project	7
Figure 3.2	Circuit diagram of the Project	8
Figure 3.3	Node MCU	9
Figure 3.4	IDE Software	12
Figure 3.5	SMPS	16
Figure 3.6	Optocoupler	17
Figure 3.7	Voltage Sensor	17
Figure 3.8	LCD Display	19
Figure 3.9	Relay module	20
Figure 3.10	Stematic of Relay Module	21
Figure 3.11	PinOut of Relay Module	21
Figure 3.12	LM2596 Voltage	22
Figure 3.13	Current Sensor	23
Figure 3.14	Project Picture	24
Figure 4.1	Program installation process	27
Figure 4.2	Flowchart of the compiling process	27
Figure 4.3	User interface of proteus 8.1	29
Figure 4.4	Android Apps	30
Figure 4.5	project Result	31
Figure4.6	Project Result	31

List Of Table

Table No:	Table Name	Page No:
Table 01	LCD display outline	19
Table 02	List of Components with Price	25

Chapter 1: Introduction

1.1 Introduction

Energy generation companies supply electricity to all the households via intermediate controlled power transmission hubs known as Electricity Grid. Sometime problems arise due to failure of the electricity grid leading to black out of an entire area which was getting supply from that particular grid. The project aims to solve this problem using IOT as the means of communication and also tackling various other issues which a smart system can deal with to avoid unnecessary losses to the energy procedures IOT smart energy grid is based on AT mega family controller which controls the various activities of the system. The system communicates over internet by using Wi-Fi technology. The foremost thing that this project facilitates is reconnection of transmission line of active grid. If an energy grid becomes faulty and there is an another energy grid, the system switches the transmission lines towards this grid thus facilitating an interrupted electricity supply to that particular region whose energy grid went OFF. And this information of which grid is active updated over IOT smartphone apps where the authorities can login and can be the updates. Apart from monitoring the grid, this project has the advance capabilities of monitoring energy consumption and even detects theft of electricity.

1.2 Objective of the Project

The main objective of this project is to“ Controlling of Power Grid of 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 closed at dangerous levels.
- Monitoring the power line current.
- Reduce costs from conventional systems.

1.3 Background of the Project

- The title of project is IoT Based Power Grid Controlling and Monitoring System .

- In this project we have done voltage and frequency monitoring. As well as monitoring the 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, Embedded Processing 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 vertical domain 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, commodity sensors, and embedded systems.

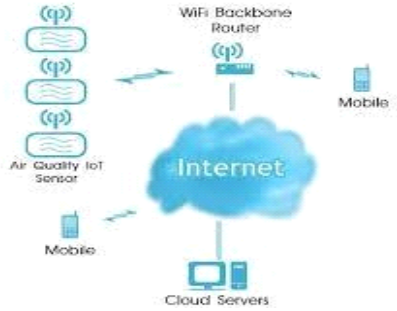


Fig-1.1: IoT System

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.

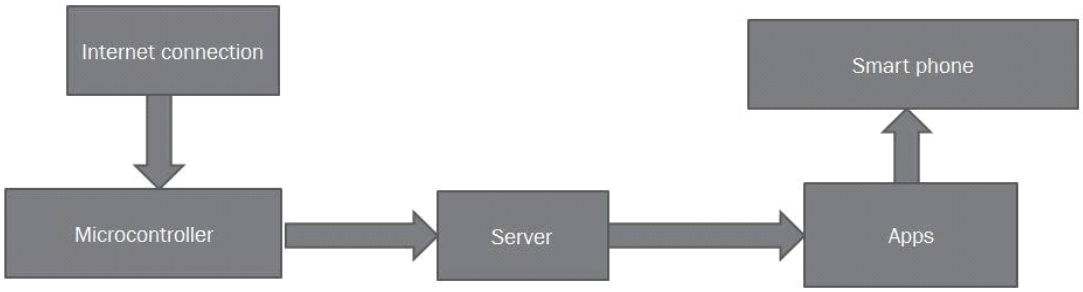


Fig-1.2: Block Diagram of IOT

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.

1.6 Methodology

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.

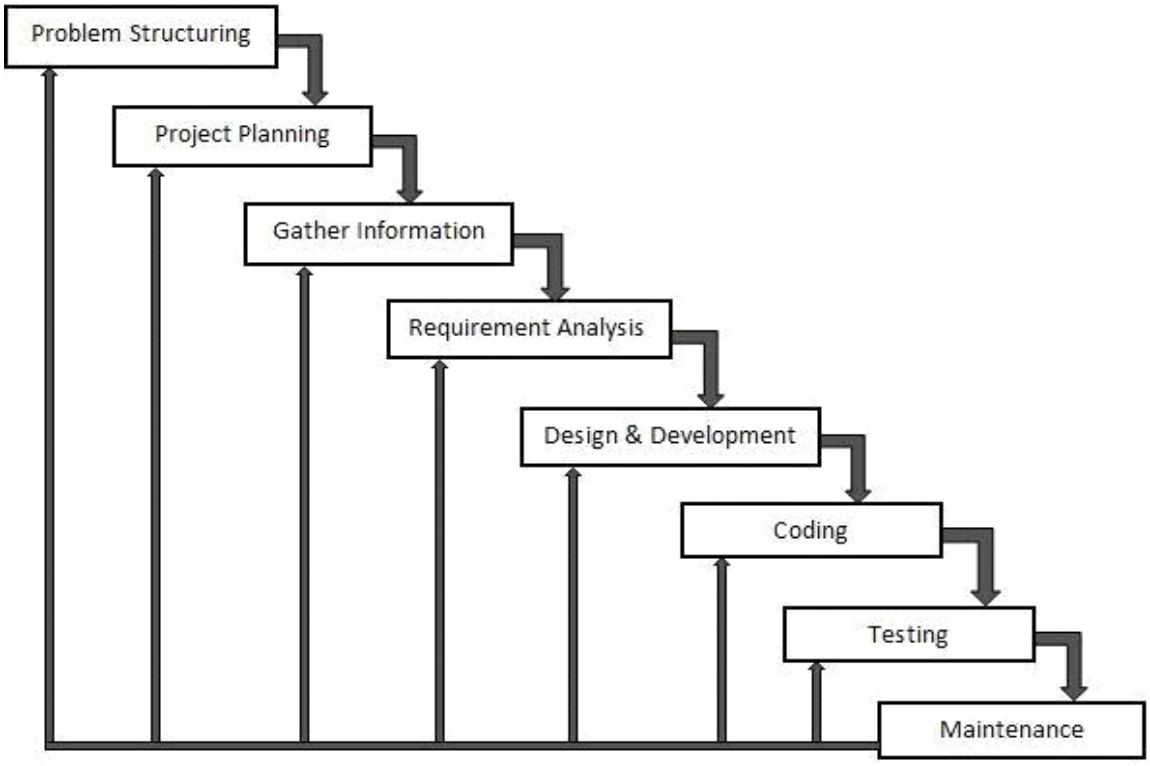


Fig-1.3: Methodology

1.7Motivation for Project

A poorly motivated team has been known to unravel even the best project plan. A good project manager needs to know how to harness the initial excitement that comes with starting a project and use it to maintain motivation – leading to success throughout the project's lifecycle. We now know that contemporary project managers need to be more than just schedulers and contract managers. They need excellent skills in managing those complex human elements that have the potential to bring any project down.

Chapter 2: Literature Review

2.1 Introduction

through internet and mobile applications, enabling improved grid management. In the future generation of power grids sensors, actuators, and transducers are expected to play a crucial role in providing real-time energy monitoring and controlling services. IoT has become an enabling technology to provide novel solutions to the challenges in the power grid system. IoT enabled sensors are used pervasively in the power grid system to share their useful information

The integration of Information and Communication Technologies (ICTs) and IoT in SG ensure reliability, cost effectiveness, and intelligent features with minimal human intervention. Two-way communication is the key requirement in the paradigm of IoT among smart devices and components.

Smart homes are developed by the integration of smart electric meters and IoT. For deployment in smart cities, an IoT assisted real-time Zigbee mesh WSN based Automatic Meter Reading (AMR) system is implemented in. The proposed system provides a reduction in peak loads with improved Demand Side Management (DSM). Sensors, communication, and control mechanisms will play a vital role in achieving reliable and secure power grids. A survey on potential communication technologies for the connectivity of devices in the SG is presented in.

Motivated by the above progress in IoT and their deployment in power grids, we have proposed an IoT assisted power monitoring and controlling system. It provides benefits to both consumers and utility companies to analyze and manage their resources. This paper provides an implementation of IoT based power monitoring using blynk software. A literature review is provided that highlights existing research in the area of SG, IoT, and IoT aided SG. In this paper, our contributions include the following:

- The implementation of IoT assisted power monitoring system is provided with the integration of an open-source IoT platform, which provides to analyze information.
- The hardware design enables to access of electrical parameters of loads including current, frequency, and voltage of connected loads.

An "IDMT over Current Relay" is a type of protective relay which operates when the load current exceeds a preset value. In a typical application the over current relay is used for over current protection, connected to a current transformer and calibrated to operate at or above a specific current level.

This project will attempt to design and fabricate over current protection relay using micro controller. The micro controller will cause the circuit breaker to trip when the current from load current reaches the setting value in the micro controller.

In order to design it, first the load current need to measure in order to monitor it using current sensor including testing the fault (over current) and when such condition arise, it will isolate in the shortest time possible without harming the any other electrical devices. It will also including in developing the algorithm for instantaneous over current relay and IDMT (Inverse Definite Minimum Time) relay for the circuit breaker to trip. In this project, arduino microcontroller will be used to control and operate the tripping coil in circuit breaker.

2.2 Literature Survey

One year after the past edition of the Cluster book 2012 it can be clearly stated that the Internet of Things (IOT) has reached many different players and gained further recognition.
Out of the potential Internet of Things application areas, Smart Cities (and regions), Smart Car and mobility, Smart Home and assisted living, Smart Industries, Public safety, Energy environmental protection, Agriculture and Tourism as part of a future IoT Ecosystem have acquired high attention. In line with this development, the majority of the governments in Europe, in Asia, and in the Americas consider now the Internet of Things as an area of innovation and growth. Although larger players in some application areas still do not recognized the potential, many of them pay high attention or even accelerate the pace by coining new terms for the IoT and adding additional components to it. Moreover, end-users in the private and business domain have nowadays acquired a significant competence in dealing with smart devices and networked applications. As the Internet of Things continues to develop, further potential is estimated by a combination with related technology approaches and concepts such as Cloud computing, Future Internet, Big Data, robotics and Semantic technologies. The idea is of course not new as such but becomes now evident as those related concepts have started to reveal synergies by combining them. However, the Internet of Things is still maturing, in particular due to a number of factors, which limit the full exploitation of the IOT. Among those factors the following appear to be most relevant

- No clear approach for the utilization of unique identifiers and numbering spaces for various kinds of persistent and volatile objects at a global scale.
- No accelerated use and further development of IOT reference architectures.
- Less rapid advance in semantic interoperability for exchanging sensor information in heterogeneous environments.
- Difficulties in developing a clear approach for enabling innovation, trust and owner ship of data n the IOT while at the same time respecting security and privacy in a complex environment.
- Difficulties in developing business which embraces the full potential of the Internet of Things.
- Missing large-scale testing and learning environments, which both facilitate the experimentation with complex sensor networks and stimulate innovation through reflection and experience.

Chapter 3: Hardware Implementation

3.1 Introduction

Internet of Things (IoT) is widely used in monitoring, industrial automation, and a variety of applications. At various stages of power Grid(PG), IoT devices are deployed to monitor and control grid statistics for reliable and efficient delivery of power. Although IoT integration in the PG domain provides manifold benefits, the challenges in IoT-PG integration needs to be solved for the efficient operation of the grid. In this paper, firstly an overview of PG and IoT based PG system is provided. This project describes the IoT based power monitoring system that is capable to measure and analyze the electrical parameters such as voltage, current, and frequency consumption of loads. IoT based software application is used to obtain the real-time electrical data of consumers. Based on this data, the consumer and electric power companies in the PG paradigm can better manage their consumption to reduce billing costs.

The main aim of Micro-controller based Controlling of Power Grid of IOT. In above block diagram there are current sensor, voltage sensor, relay module,Wi-Fi module, optocoupler, LCD display and arduino which we have used for sensing when is detected the parameter Microcontroller continuously check the output of sensor and gives signal to the system circuit which drives the automatic.

3.2 Block Diagram of This Project

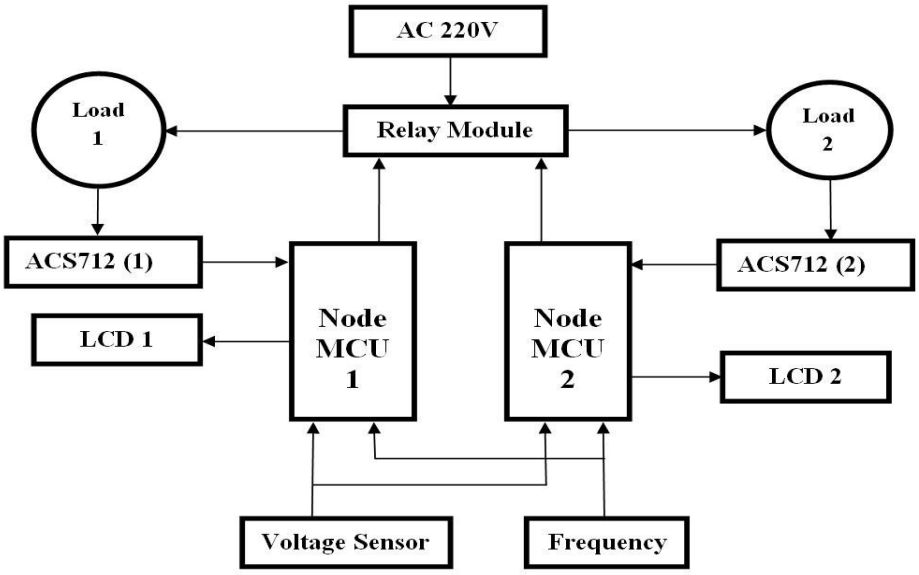


Fig: 3.1 Block Diagram

3.3 Circuit Diagram of This Project

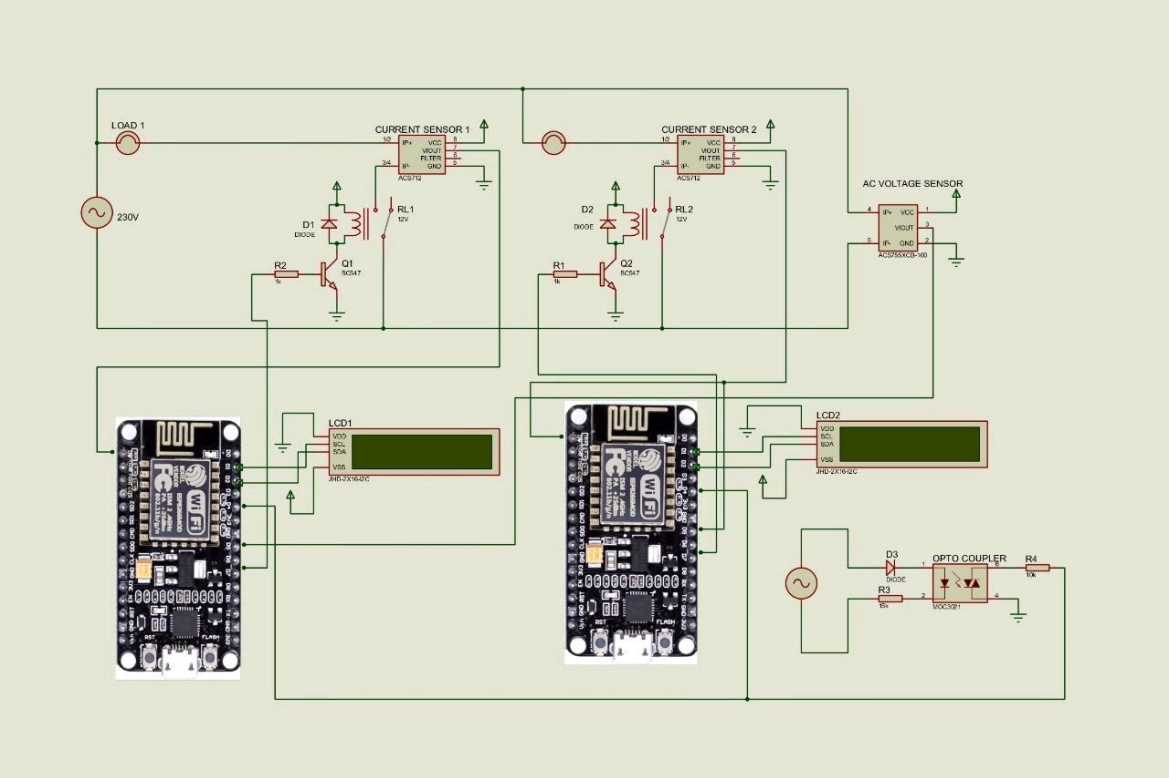


Fig: 3.2 Circuit Diagram

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

- Node MCU.
- Adaptor.
- Opto coupler.
- Voltage Sensor.
- LCD Display.
- Relay Module.
- LM2596 Voltage Converter ▪Current Sensor.
- Wifi Module.

3.5 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. 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.

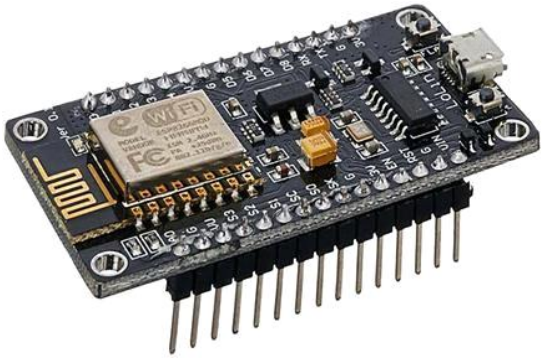


Figure 3.3 Node MCU ESP8266

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 WiFiSoC, 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-aChip (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 16bit 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.

Software:

A program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit AVR and 32-bit ARM Cortex-M based microcontrollers: AVR Studio (older) and Atmel Studio (newer).

Integrated Development Environment:

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.

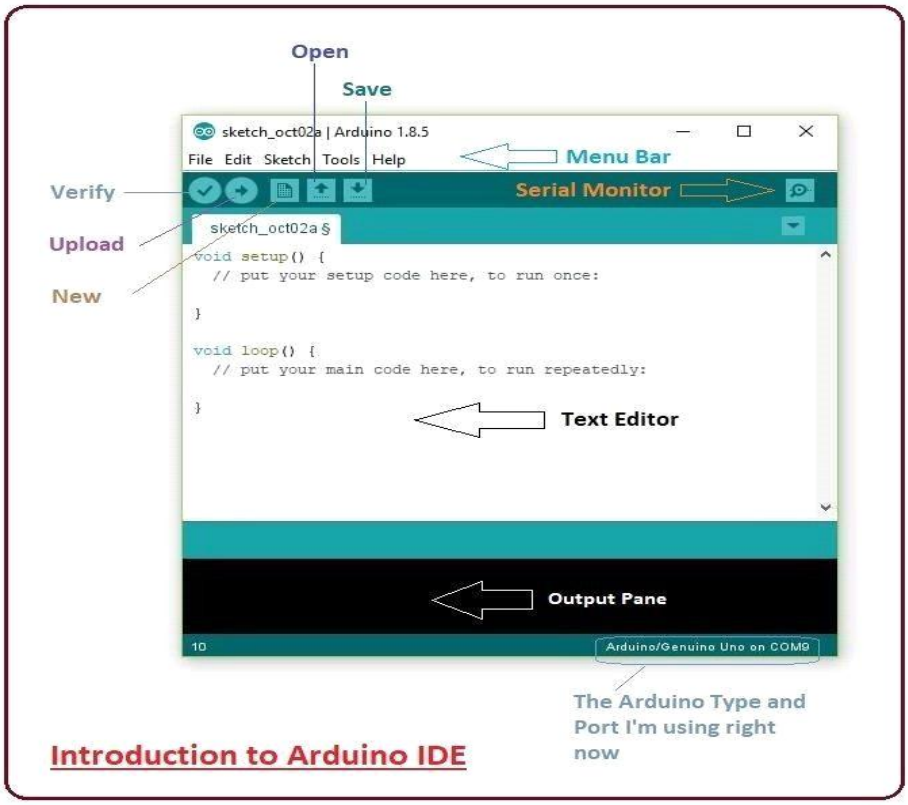


Fig-3.4 IDE software

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain,

also included with the IDE distribution. The Arduino IDE employs the program AVR dude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Sketch:

A sketch is a program written with the Arduino IDE. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension pde.

A minimal Arduino C/C++ program consists of only two functions:
setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. It is analogous to the function main().

loop(): After setup() function exits (ends), the loop() function is executed repeatedly in the main program. It controls the board until the board is powered off or is reset. It is analogous to the function while(1).

Blink example:

Most Arduino boards contain a light-emitting diode (LED) and a current limiting resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions. A typical program used by beginners, akin to Hello, World!, is "blink", which repeatedly blinks the on-board LED integrated into the Arduino board. This program uses the functions pinMode(), digitalWrite(), and delay(), which are provided by the internal libraries included in the IDE environment. This program is usually loaded into a new Arduino board by the manufacturer.

Libraries:

The open-source nature of the Arduino project has facilitated the publication of many free software libraries that other developers use to augment their projects.

- 3.4.9 Technical specifications
- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6

- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm ▪Width: 53.4 mm
- Weight: 25 g

Pins:

General pin functions:

LED: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it's off.

VIN: The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.

3V3: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND: Ground pins.

IOREF: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

Reset: Typically used to add a reset button to shields which block the one on the board.

Special pin functions

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal

pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.

In addition, some pins have specialized functions:

Serial / UART: pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.

External interrupts: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

PWM (pulse-width modulation): 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite() function.

SPI (Serial Peripheral Interface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

TWI (two-wire interface) / I²C: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
AREF (analog reference): Reference voltage for the analog inputs.

Communication:

The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A software serial library allows serial communication on any of the Uno's digital pins.

Applications:

- Arduboy, a handheld game console based on Arduino.
- Arduinome, a MIDI controller device that mimics the Monome.
- Ardupilot, drone software and hardware.
- ArduSat, a cubesat based on Arduino.
- C-STEM Studio, a platform for hands-on integrated learning of computing, science, technology, engineering, and mathematics (C-STEM) with robotics.
- Data loggers for scientific research.
- OBDuino, a trip computer that uses the on-board diagnostics interface found in most modern cars.
- OpenEVSE an open-source electric vehicle charger.
- XOD, a visual programming language for Arduino.

3.6 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-3.5 SMPS

3.7 Optocoupler

These are High Density Mounting Type Photo couplers used in Computer terminals, System appliances, measuring instruments, Registers, copiers, automatic vending machines, Electric home appliances, such as fan heaters etc.

Features:

Current transfer ratio (CTR:MIN. 50% at IF=5mA Vce=5V)

High isolation voltage between input and output (Viso: 5300Vrms).

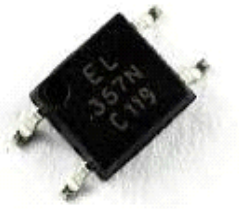


Fig-3.6 Optocoupler

3.8 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 or 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.

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.



Fig-3.7 Voltage Sensor

Specification:

*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)

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.

3.9 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 characters and 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.

LCD display outline

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

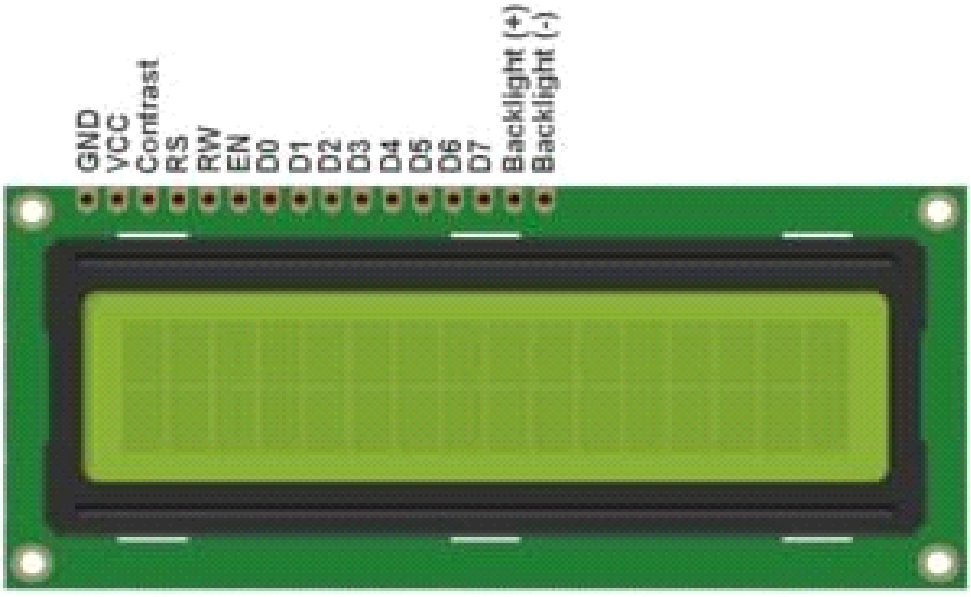


Fig- 3.8 LCD Display

3.10 Relay Module

A relay is an electrically operated device. It has a control system and (also called input circuit or input contactor) and controlled system (also called output circuit or output cont. actor). It is 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-3.9 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 devices of electro mechanics and power electronics. Generally speaking, a relay contains an induction part 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 part between 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:

- Good in safety. In power system and high voltage system, the lower current can control the higher one.
- 1-channel high voltage system output, meeting the needs of single channel control

3)Wide range of controllable voltage.

- Being able to control high load current, which can reach 240V, 10A
- 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:

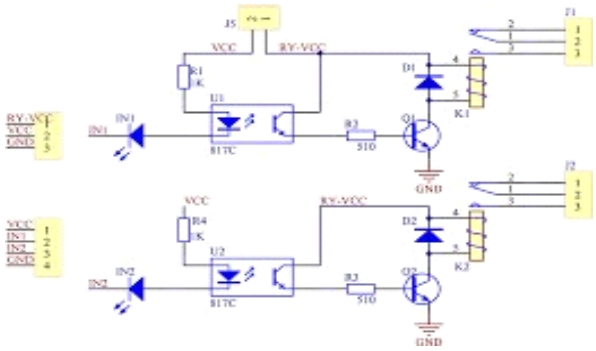


Fig-3.10 Stematic Of Relay Module

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.

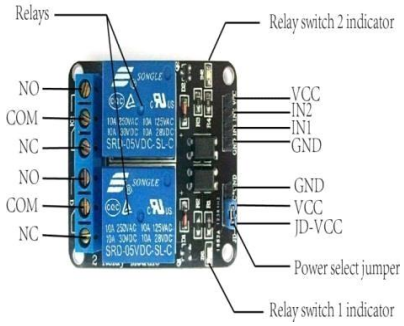


Fig-3.11 Pin out of Relay Module

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: void

```
    setup(){ pinMode(7,OUTPUT);
}         void
loop(){
```

```
pinMode(7,OUTPUT);
delay(2000);
digitalWrite(7,LOW);
delay(2000);
}
```

Feedback of test:

After the upload of code, connect the IN1 pins to the 7th pin of Arduino board separately, you can hear the ticktack of the relay (it keeps opening and closing), in addition, the LED-G is blinking every 2s.

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 relay module 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.

3.11 LM2596 Voltage Converter

The LM2596 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3-A load with



Fig: 3.12 LM2596 Voltage Converter

Excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version. Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation, and a fixed frequency oscillator. The LM2596 series operates at a switching frequency of 150 kHz, thus allowing smaller sized filter components than what would be required with lower frequency switching regulators. Available in a standard 7-pin TO-220 package with several different lead bend options, and a 7-pin TO-263 surface mount package.

3.12 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.



Fig-3.13 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);

3.13 Project Implementation Result

Power grid represents one of the most promising and prominent internet of things applications and More efficient transmission of electricity and Quicker restoration of electricity after power disturbances. Reduced operations and management costs for utilities, and ultimately lower power costs for consume. A sensor is set at the heap to ascertain current, a circuit is utilized to figure voltage and with these two, power can be computed. Control qualities are put away in cloud database. A apps facilitating and space is made to get the orders from android application and send them to arduino board at load, which triggers an electromagnetic transfer to change the condition of the heap. This project permit to get the power values and control gadgets from anyplace on the planet.

3.14 Project picture

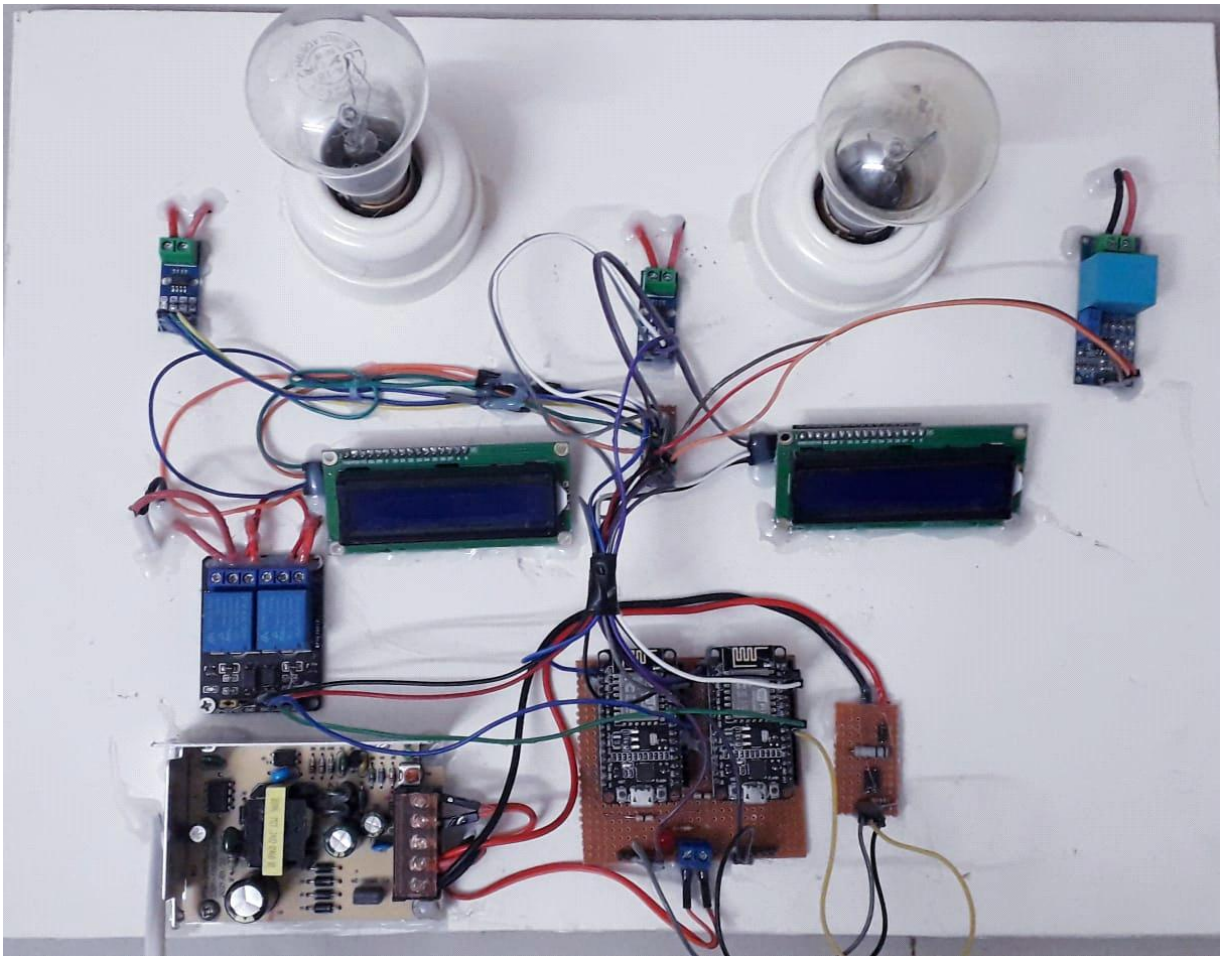


Fig-3.14 Project picture

3.15 List of Components With Price

SL NO	Components Name	Quantity	Unite Price	Total Price (BDT)
1	Node MCU	2	550	1100
2	SMPS	1	380	380
3	LM2596 Buck Module	1	100	100
4	Relay Module	1	150	150
5	Current Sensor	2	220	440
6	LCD Display	1	450	450
7	2 pin	1	20	20
8	Voltage Sensor	1	350	350
9	Optocoupler	1	45	45
10	Load	2	70	140
11	Resistor			20
12	Others	1	500	500
	Total Cost			3,395tk

Table02: List of Components With Price

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 the software. The code is written on C.

4.2 Software Tools

The software that is used to program the microcontroller is open-source-software and can be downloaded for free on www.arduino.cc.With this "Arduino software" we can write little programs with the microcontroller. These programs are called "Sketch".

In the end the sketches are transferred to the microcontroller by USB cable and More on that later on the subject "programming".

Installation

Now one after another the Arduino software and the USB driver for the board have to be installed.

Installation and setup of the Arduino software

1. We have downloaded the Arduino software from www.arduino.cc and installed it on the computer (This was NOT connected to the PC). After that we opened the software file and installed the program named arduino.exe.

Two set ups on the program are important and should be considered.

- The board that we want to connect has to be selected on the arduino software. The "Arduino Uno" is here known as "Arduino / Genuino Uno".

- We have to choose the right "Serial-Port", to let the Computer know to which port the board has been connected. That is only possible if the USB driver has been installed correctly. It can be checked this way:

At the moment the Arduino wasn't connected to the PC. If we now choose "Port", under the field "Tool", we will already see one or more ports here (COM1/ COM2/ COM3…). The quantity of the shown ports doesn't depend on the quantity of the USB ports on the computer. When the board gets connected to the computer, we will find one more port.

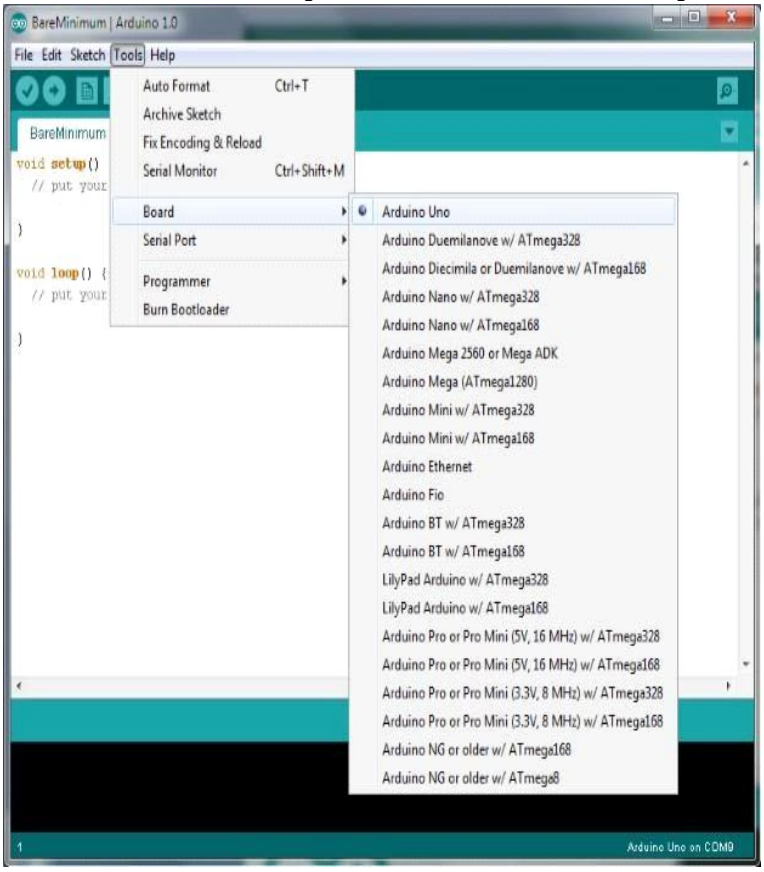


Fig. 4.1: Program installation process

4.3 Programming

The development cycle is divided into 4 phases:

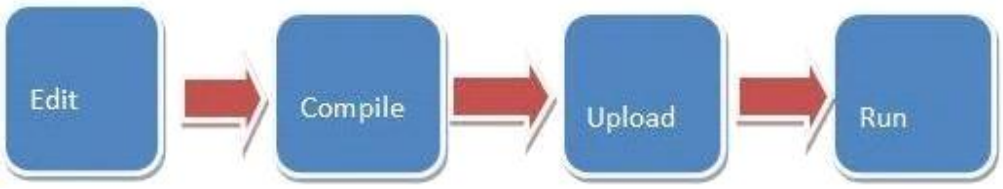


Fig-4.2: Flowchart of the compiling process

Compile: Compile means to translate the sketch into machine language, also known as object.

Code Run: Arduino sketch is executed as soon as terminates the step of uploading on the board.

4.4 Arduino Program Developments

- Based on C++ without 80% of the instructions.
- A handful of new commands.
- Programs are called 'sketches'.
- Sketches need two functions:
- void setup ( )
- Void loop ( )
- Setup ( ) runs first and once.
- loop ( ) runs over and over, until power is lost or a new sketch is loaded.

4.5 Proteus8.1

Proteus 8 is best simulation software for various designs with microcontroller. It is mainly popular because of availability of almost all microcontrollers in it. So it is a handy tool to test programs and embedded designs for electronics hobbyist. You can simulate your programming of microcontroller in Proteus 8 Simulation Software. After simulating your circuit in Proteus 8 Software you can directly make PCB design with it so it could be a all in one package for students and hobbyists. So I think now you have a little bit idea about what is proteus software.

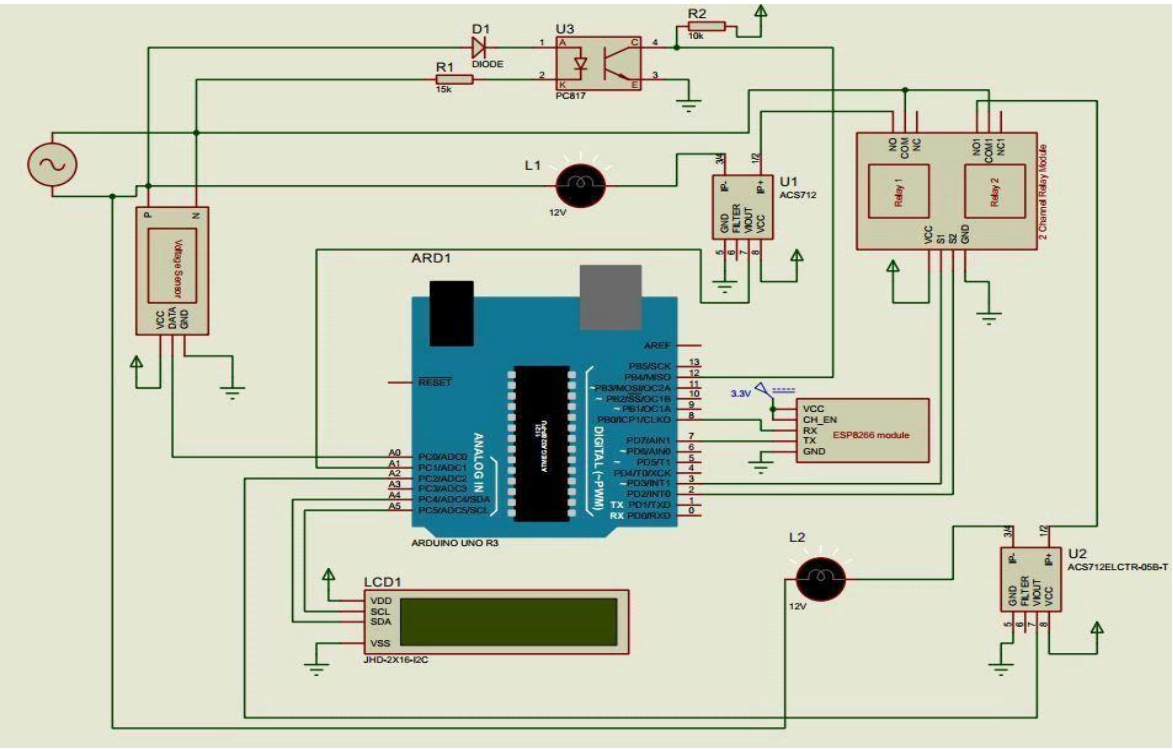


Fig- 4.3: User interface of proteus 8.1

Proteus 7.0 is a Virtual System Modeling (VSM) that combines circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller based designs. This is the perfect tool for engineers to test their microcontroller designs before constructing a physical prototype in real time. This program allows users to interact with the design using on-screen indicators and/or LED and LCD displays and, if attached to the PC, switches and buttons. One of the main components of Proteus 7.0 is the Circuit Simulation -- a product that uses a SPICE3f5 analogue simulator kernel combined with an event-driven digital simulator that allow users to utilize any SPICE model by any manufacturer. Proteus VSM comes with extensive debugging features, including breakpoints, single stepping and variable display for a neat design prior to hardware prototyping. In summary, Proteus 7.0 is the program to use when you want to simulate the interaction between software running on a microcontroller and any analog or digital electronic device connected to it.

4.6 Android Apps

For visualization and authentication, in this research work, the team has an Android App. The Android Application does not restrict itself to Thing Speak but also has a Firebase login system. Firebase is a Google-provided API to create a database and fetch from it in real time. It also provides enhanced security for the developed App. Also, it is used for backend support and other functionalities like data storage, user authentication and hosting. Real-time

data variations are recorded automatically and updates are sent to clients. The HTTP protocol works on a simple request and response system but firebase is different as its real-time database uses data synchronization. It can be used for user authentication purposes as the usercredentials are securely stored using crypt. Firebase authentication to the android studio is added using Firebase Android Studio tool. Whenever someone authenticates during the sign- in process either using an Email or Google, it handovers a firebase user object that represents that the authentication is successfully done by the user.

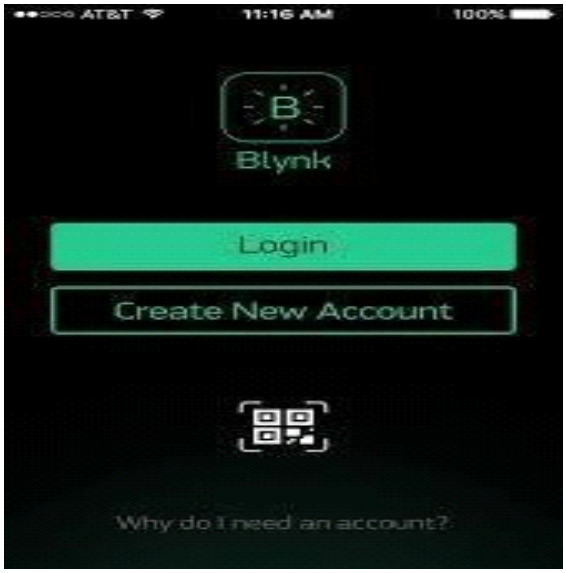


Fig- 4.4: Android Apps.

This object contains the basic information about the user. Once the firebase API is included in Android or iOS App, firebase features like Analytics, Authentication, Storage, Messaging, Hosting, Crash reporting, Real-time Database etc.

4.7 Project Result

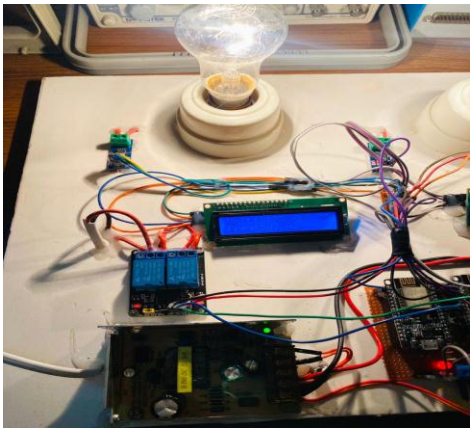


Fig-4.5 Line 01 Picture

System 01

1. Voltage=220V
2. Current = 0.50A
3. Frequency = 50 Hz

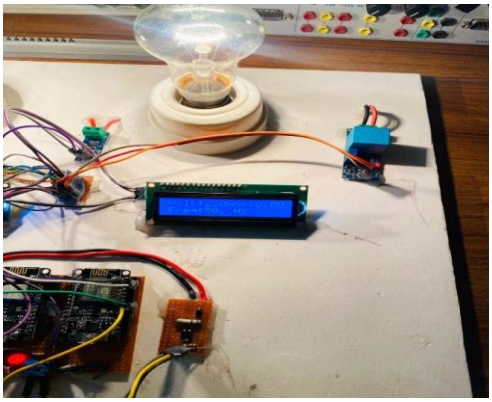


Fig-4.6 line 02 picture

System 02

1. Voltage = 220v
2. Current = 0.30A
3. Frequency = 50

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 Smart power Grid. Smart power Grid is owner as well as user friendly technology. Authority can check current per system from any location using internet. Authority can control system power line from internet. Smart power grid represents one of the most promising and prominent internet of things applications. More efficient transmission of electricity. Quick errestoration of electricity after power disturbances. Reduced operations and management costs for utilities, and ultimately lower power costs for consumers. Time saving technology.

5.3 Advantages

- Remote monitoring system
- The line can be closed at dangerous levels.
- Voltage and current can be monitored. ▪Low cost and very easy to implement.

5.4 Applications

- Power Grid.

- Industry. ▪Substation.

5.5 Future works

▪Sending notifications via message as soon as a problem is created on the line.

References:

- Madhuri G.Hiremath1, Veeresh Pujari2, Dr. BaswarajGadgay. “IOT Based Energy Monitoring & Control Devices”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 5 Issue VI, June 2017
- Rajiv .K. Bhatia and VarshaBodade,"Smart Grid Security and Privacy: Challenges, Literature Survey and Issues", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 1, January 2014.
- MitaliMahadevRaut, 2Ruchira Rajesh Sable, 3Shrutika RajendraToraskar. “Internet of Things(IOT) Based Smart Grid”, International Journal of Engineering Trends and Technology (IJETT) , Volume 34 Number 1- April 2016
- StamatisKarnouskos,"The cooperative Internet ofThings enabled Smart Grid", SAP Research,Vincenz-PriessnitzStrasse 1, D-76131, Karlsruhe, Germany,2013.
- https://en.wikipedia.org/wiki/Liquid-crystal_display
- https://www.eeweb.com/companyblog/allegro_microsystems/hall-effect-linear-currentsensor-ic
- http://esp8266.net

Appendix:

Program:

```
#define BLYNK_PRINT Serial


#include <ESP8266_Lib.h>
```

```
#include <BlynkSimpleShieldEsp8266.h>
#include <SoftwareSerial.h>
#include "EmonLib.h"             // Include Emon Library
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// for esp8266
SoftwareSerialEspSerial(7, 8); // RX, TX

char auth[] = "keN0TNT4BgDsSMyJYhqxqu65EskLTKyn";
char  ssid[]  =  "student";
char pass[] = "iotstudent";

#define ESP8266_BAUD 9600
ESP8266 wifi(&EspSerial);

EnergyMonitor emon1;             // Create an instance

float involt;


LiquidCrystal_I2C lcd(0x27, 20, 4);



// define for
Frequency int input =
12; inthigh_time;
intlow_time; float
time_period; float
freq;
float frequency;


float power;
float supplyVoltage;

//variable for current sensor
#define acs712 A1
intacspower = 5; float
amperage = 0.00;
float mean = 0.0;
```

```
//current sample count
long lastsample = 0;
long samplesum = 0;
intsampleCount = 0;
float vpc =
4.8828125;
float value;

//variable for current
sensor2 #define acs7121 A2
int acspower1 = 5; float
amperage1 = 0.00; float
mean1 = 0.0;


//current sample count
long lastsample1 = 0;
long samplesum1 = 0;
int sampleCount1 = 0;
float vpc1 = 4.8828125;
float value1;


//flag     int
flag = 0;
int flag1 = 0;

void setup() {

  pinMode(input, INPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);

  Serial.begin(9600); lcd.begin(); lcd.backlight(); emon1.voltage(0, 640.25, 1.7); //
  Voltage: input pin, calibration 234.26, phase_shift

  EspSerial.begin(ESP8266_BAUD);
  delay(10);


 Blynk.begin(auth, wifi, ssid, pass);
}
```

```
void loop() {

  emon1.calcVI(20, 1000);        // Calculate all 2000. No.of half wavelengths (crossings),
time-out
  //emon1.serialprint(); // Print out all variables (realpower, apparent power, Vrms, Irms,
power factor)



   float supplyVoltage = emon1.Vrms;             //extract Vrms into Variable


  high_time = pulseIn(input, HIGH);
  low_time = pulseIn(input, LOW);

  time_period    =    high_time    +    low_time;
  frequency = 1000 / time_period;



  if (supplyVoltage< 100.00)
  {    supplyVoltage    =
   0.0;
  }



  if (frequency >= 60.0)
  {    frequency    =
   0.0;
  }



  // current measure1

  for (sampleCount = 0; sampleCount<= 1000; sampleCount ++)
  {   samplesum   +=   (analogRead(acs712)   -
   508);
   // sampleCount++; delay(1);
  }


  mean = samplesum / 1000;
  // Serial.println(mean);
```

```
amperage = (mv / 16);
samplesum = 0;

  Serial.println(" amp: " + String(amperage));


// current measure2

for (sampleCount1 = 0; sampleCount1 <= 1000; sampleCount1 ++)
{   samplesum1   +=   (analogRead(acs7121)   -
  508);
  // sampleCount1++; delay(1);
}


mean1 = samplesum1 / 1000;

}


amperage1 = (mv1 / 16);



samplesum1 = 0;

Serial.println(" amp1: " + String(amperage1));




Serial.println(supplyVoltage);
Serial.println(frequency);
Serial.println("");

Blynk.virtualWrite(V0, supplyVoltage);
Blynk.virtualWrite(V1, frequency);
Blynk.virtualWrite(V2, amperage);
Blynk.virtualWrite(V3, amperage1);


delay(1000); lcd.clear();
lcd.setCursor(0, 0);
```

```
  lcd.print("Voltage:");
  lcd.print(supplyVoltage)
  ; lcd.print("V");

  lcd.setCursor(0, 1);
  lcd.print("Frequency:");
  lcd.print(frequency, 1);
  lcd.print("Hz");

  lcd.setCursor(0,     2);
  lcd.print("System
  1:");
  lcd.print(amperage);
  lcd.print("A");

  lcd.setCursor(0,     3);
  lcd.print("System
  2:");
  lcd.print(amperage1);
  lcd.print("A");


}
```