

# **IOT BASED PREPAID SMART ENERGY METER**

Submitted in partial fulfilment of the requirements for the award of  
Bachelor of Technology degree in BIOMEDICAL ENGINEERING

by

**KEERTHANA R (39240019)**

**NISHA S (39240027)**

**POOJAHSRI S (39240029)**



**INTERDISCIPLINARY PROJECT REPORT**

**DEPARTMENT OF BIOMEDICAL ENGINEERING**

**SCHOOL OF BIO AND CHEMICAL ENGINEERING**

# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**APRIL 2022**



# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE  
[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

## **DEPARTMENT OF BIOMEDICAL ENGINEERING**

### **BONAFIDE CERTIFICATE**

This is to certify that this Interdisciplinary Project Report is the bonafide work of **Keerthana R (39240019) Nisha S (39240027), Poojalsri S (39240029)** who carried out the Inter disciplinary project entitled **"IOT BASED PREPAID SMART ENERGY METER"** under our supervision from **December 2021 to April 2022**.

**Guide**  
**(Dr. J. PREM KUMAR)**

**Head of the Department**  
**(Dr. T. SUDHAKAR)**

**Submitted for Viva voce Examination held on 30.04.2022**

**Internal Examiner**

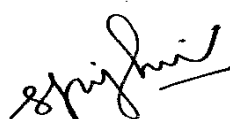
**External Examiner**

## DECLARATION

We **Keerthana R, Nisha S, Poojahsri S** hereby declare that the Interdisciplinary Project Report entitled “**IOT BASED PREPAID SMART ENERGY METER**” done by us under the guidance of **Dr. J. PremKumarat** Department of Biomedical Engineering, Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of **Bachelor of Technology** degree in **Biomedical Engineering**.

DATE: 28-04-2022

PLACE: Chennai



Keerthana.R

S.Nisha

SIGNATURE OF THE CANDIDATES

## ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this Interdisciplinary Project and for completing it successfully. We are grateful to them.

We would like to express our sincere and deep sense of gratitude to our Project Guide Dr. **J. Premkumar** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We convey our thanks to Dr. **T. Sudhakar**, Head of the Department, Department of Biomedical Engineering for providing us necessary support and details at the right time during the progressive reviews.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Biomedical Engineering who were helpful in many ways for the completion of the project.

## TABLE OF CONTENTS

| <b>TITLE</b>                                                  | <b>PAGE NO.</b> |
|---------------------------------------------------------------|-----------------|
| <b>List of Figures</b>                                        | <b>vi</b>       |
| <b>List of Abbreviations</b>                                  | <b>vii</b>      |
| <b>Abstract</b>                                               | <b>viii</b>     |
| <b>Chapter –1 Introduction</b>                                | <b>1</b>        |
| <b>Chapter –2 Literature survey</b>                           | <b>4</b>        |
| <b>Chapter – 3 Aim and Scope of the present Investigation</b> | <b>5</b>        |
| <b>3.1 Aim</b>                                                | <b>5</b>        |
| <b>3.2 Scope of the present Investigation</b>                 | <b>5</b>        |
| <b>Chapter –4 Materials and methods</b>                       | <b>6</b>        |
| <b>4.1 Components used</b>                                    | <b>6</b>        |
| <b>4.1.1 Arduino UNO</b>                                      | <b>6</b>        |
| <b>4.1.2 ESP 12 node MCU module</b>                           | <b>7</b>        |
| <b>4.1.3 16*2 LCD</b>                                         | <b>8</b>        |
| <b>4.1.4 Optocoupler</b>                                      | <b>9</b>        |
| <b>4.1.5 Single channel relay driver</b>                      | <b>10</b>       |
| <b>4.1.6 Analogue Energy meter</b>                            | <b>10</b>       |
| <b>4.2 Software used</b>                                      | <b>10</b>       |
| <b>4.2.1 Arduino IDE</b>                                      | <b>10</b>       |
| <b>4.2.2 MQTT Mobile application</b>                          | <b>11</b>       |
| <b>4.3 Methodology</b>                                        | <b>11</b>       |
| <b>4.3.1 Working principle</b>                                | <b>11</b>       |
| <b>4.3.2 Block diagram with explanation</b>                   | <b>12</b>       |
| <b>4.3.3 Work flow</b>                                        | <b>13</b>       |
| <b>Chapter –5 Results</b>                                     | <b>14</b>       |
| <b>Chapter –6 Summary and Conclusions</b>                     | <b>17</b>       |
| <b>References</b>                                             | <b>18</b>       |



## LIST OF FIGURES

| Sl. No | Name Of Figure                                    | Page Number |
|--------|---------------------------------------------------|-------------|
| 4.1    | Arduino UNO                                       | 7           |
| 4.2    | Esp12E nodemcu MODULE                             | 8           |
| 4.3    | LCD Output                                        | 9           |
| 4.4    | Optocoupler                                       | 9           |
| 4.5    | Single channel relay driver                       | 10          |
| 4.6    | Analog Energy Meter                               | 10          |
| 4.7    | Flow Diagram                                      | 11          |
| 4.8    | Block diagram                                     | 12          |
| 4.9    | Methodology Diagram                               | 13          |
| 5.1    | Initialization of Module                          | 14          |
| 5.2    | Initial balance and units consumed                | 14          |
| 5.3    | Low Balance displayed in LCD                      | 15          |
| 5.4    | Low balance alert in mobile                       | 15          |
| 5.5    | Load is switched OFF Due to zero balance          | 15          |
| 5.6    | Recharged amount is added to the previous balance | 16          |
| 5.7    | Sending message to module to recharge             | 16          |

## LIST OF ABBREVIATIONS

| ABBREVIATIONS    | FULL FORM                                                 |
|------------------|-----------------------------------------------------------|
| <b>IOT</b>       | Internet Of Things                                        |
| <b>LCD</b>       | Liquid Crystal Display                                    |
| <b>EB</b>        | Electricity Board                                         |
| <b>LDR</b>       | Light Dependent Resistor                                  |
| <b>ROM</b>       | Read Only Memory                                          |
| <b>EEPROM</b>    | Electrically Erasable Programmable Read Only Memory       |
| <b>CPU</b>       | Central Processing Unit                                   |
| <b>GSM</b>       | Global Service for Mobile Communication                   |
| <b>IDE</b>       | Integrated Development Environment                        |
| <b>USB</b>       | Universal Serial Bus                                      |
| <b>SRAM</b>      | Static Random Access Memory                               |
| <b>TCP/IP</b>    | Transmission Control Protocol/Internet Protocol           |
| <b>GPIO</b>      | General Purpose Input Output                              |
| <b>UART</b>      | Universal Asynchronous Receiver Transmitter               |
| <b>LED</b>       | Light Emitting Diode                                      |
| <b>NPN</b>       | Negative Positive Negative transistor                     |
| <b>NC and NO</b> | Normally Closed and Normally Open                         |
| <b>ULN2003</b>   | High Voltage and High Current Darlington Transistor Array |

## **ABSTRACT**

An important research issue in smart metering is to correctly transfer the smart meter readings from consumers to the operator within the given time period by consuming minimum energy. The smart meter is recharged with the help of Esp12E nodemcumodule. The EB station receives the amount and it enables the smart meter to provide the electricity to the house. The recharge can be done on the daily basis and it sends the alert message whenever the consumer exceeds the recharged days. This method helps to reduce the wastage of electricity. This paper presents a smart energy meter for an automatic metering and billing system. In this meter, energy utilized and the corresponding amount will be displayed on the LCD continuously and communicated to the controlling base station. The feedback from the user helps in identifying the usages between authorized and unauthorized users which helps in controlling the power wastage. Esp module is used for sending messages to the local authorities regarding user power consumptions. This meter can work as either prepaid meter. The proposed system replaces traditional meter reading methods and enables remote access of existing energy meter by the energy provider. Also they can monitor the meter readings regularly without the person visiting each house. The prototype will be able to introduce the billing system to the customers, get the power consumption data from smart meter, keep the data in centralized database and generate the report.

Keywords: Esp module, smart meter, IoT, energy

# CHAPTER 1

## INTRODUCTION

Electricity is one of the vital requirements for sustainment of contents of life. It should be used very judiciously for its proper utilization. But in our country, we have lot of locality where we have surplus supply for the electricity while many areas do not even have access to it. Our policies of its distribution are also partially responsible for this because we are still not able to correctly estimate our exact requirement and still power theft is prevailing. On the other hand, consumers are also not satisfied with the services of power companies. Most of the time they have complaints regarding statistical errors in the monthly bills. With this we can monitor meter and track if any fault is there or not. In previous meter a circular metal strip rotates and according to that rotation we calculate the consumption. But our meter works on pulse which is obtained with the help of LDR sensor according to consumption and we previously connected Arduino board which monitor the pulse and according to pulse the bill is generated. With the help of this project we are aiming to receive the monthly energy consumption from a remote location directly to centralized office. In this way we can reduce human efforts needed to record the meter readings which are till now recorded by visiting every home individually. Smart energy meter is an electronic device that measures the most accurate amount of electricity consumed by a residence, business or any electrically-powered device. A smart meter is reliable source for most accurate information of consumed energy that reduces the chance of error in the existing billing system to minimal.

The work system adopts a totally new concept of "Prepaid Energy Meter". The WIFI MODULE is used so that the consumer would receive messages about the consumption of power (in watts) and if it reaches the minimum amount, it would automatically alert the consumer to recharge. This technology holds good for all electricity distribution companies, private communities, IT parks and self-containing housing projects. The implementation of this project will help in better energy management, conservation of energy and also in doing away with the unnecessary hassles over incorrect billing. The automated billing system will keep track of the real time consumption and will leave little scope for disagreement on consumption and billing. It is observed that one of the faulty subsystems contributing to the huge

revenue loss in Nigerian Power Sector is the metering and billing system. Errors get introduced at every stage of energy billing, like: errors with electro-mechanical meters, human errors while noting down the meter reading; and error while processing the paid bills and the due bills. The remedy for this drawback is a prepaid energy billing. There are clear results from many countries, where prepaid system has reduced the revenue loss by a large amount. A WIFImodule based Energy Recharge Interface which contains a prepaid card equivalent to a mobile SIM card. The prepaid setup communicates with the power utility using communication network. Once the prepaid setup is out of balance, the consumer load is disconnected from the utility supply by the latching Relay (contactor). The power utility can recharge the prepaid setup remotely through messages base on customer requests. The results obtained shows good system performance. A prior billing is bound to do away with the problems of unpaid bills and human error in meter readings, thereby ensuring justified revenue for the utility.

## **EXISTING PREPAID METERING TECHNOLOGIES:**

### **SMART CARD based prepaid Energy Meters:**

Smart card is a credit card sized plastic card embedded with an integrated circuit (IC) and usually it consists of a ROM, EEPROM and a CPU. A smart card provides both the memory capacity and the computational capability. Access to data stored on the card is under the control of the smart card operating system. In this method consumer have to have the smart card recharged for the amount he chooses and enter the card into the card reader of the energy meter. Then the meter store the number of units recharged and start to measures the energy consumption. When purchased units are used up the meter disconnect the power supply until the next recharge.

### **RFID (Radio-Frequency Identification) based Prepaid Energy Meters:**

Radio-frequency identification (RFID) is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders. The technology requires some extent of cooperation of an RFID reader and an RFID tag. An RFID tag is an object that can be applied to or

incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. Some tags can be read from several meters away and beyond the line of sight of the reader. In this method RFID cards which are issued by the electricity suppliers to individual consumers are used. This RFID card is unique with a code in it and consumers are free to make flexible recharge. When the consumer wants to use the system, he needs to show the card to the reader, then the unique code inside the card is recognized by the reader, and starts deducing the amount of the RFID card as per the quantized unit charge. When the usage completes the consumer has to recharge the RFID card again.

### **Our Project is based on IOT Based Prepaid Smart metering system:**

- Analog Energy Meter, which is used to measure the flow of electric power from input to the output terminal.
- LCD Display, which is used for displaying readings of the parameters that are being measured.
- Esp12E nodemcumodule is used for communication with the end user and it is responsible for updating the data on the distribution side.

### **Problem Statement**

As we mentioned above several advantages of digital energy meters, but always there are chances of innovation or modification in different instruments for ease of consumer and supplier. Following are some problems observed in those energy meters which should be rectified:

- Meter reading and other related tasks like bill payment are performed by a large number of staff i.e., large number of employees are required.
- An expensive number of staff is utilized for meter reading and other related assignments like bill payment.
- Billing errors due to carelessness of meter readers during meter reading and sometime billing estimation.
- Consumer has to stand in queue for hours for bill payment.
- Careless usage of electricity by consumer who is unaware of its cost.

## CHAPTER - 2

### LITERATURE SURVEY

*Ashna, et.al., (2013)* has proposed the design of a simple low-cost wireless GSM energy meter and its associated wave interface, for automating billing and managing the collected data globally.

*Vivek Kumar Sehgal et.al.,(2020)* has introduced the concept of Post paid Energy Meter which automatically sense the energy used in the home and when it reaches value which is initially fed in the hardware it will disconnect the power line. A user interface given in the hardware for user which will interact with the hardware, through user interface user can set a value.

*Prajakta et al., (2020)* proposed the design and implementation of a GSM based remote operation of an energy meter which gives the solution to power theft, consumption control, auto billing and payment, data logging and Manpower reduction in power distribution and management.

The major disadvantage of the present systems is these systems are post paid base service, Electricity board gives the service before collecting the payment from the customers and it is difficult for Electricity board to collect the payable amount from the customers and at the same Electricity board cannot stop to providing the service. So there is need of a system service after payment.

## **CHAPTER 3**

### **AIM AND SCOPE OF PRESENT INVESTIGATION**

#### **3.1 AIM:**

The aim of the project is to minimize the queue at the energy meter billing counters and to restrict the usage of energy meter automatically, if the bill is not paid. The project also aims at proposing a system that will reduce the excess usage of power in household and reduce the man power in measuring the electricity consumed.

#### **3.2 SCOPE OF PRESENT INVESTIGATION:**

- This is 21st century and there is no space for errors or faults either in any technical system or in general applications. Prepaid energy meter is a beneficial concept for the future.
- It's facilitates the remission from electricity bills. Electricity vouchers will be available at nearby shops. The word prepaid means "pay before use" one of the beneficial feature of this concept prepaid energy meter is used to prepaid the current supply of electricity to homes, offices etc.
- Now Future Aspects is one of the main aspect regarding any projects. Now here "Prepaid Electricity Meter" provides you smooth as well as better capital processing with the management of energy utilities. It simplifies the problems approaching in billings & electricity by lowing the obstacles or difficulties faced by user in the respective living areas. Also it reduces the big amount of time along with manpower while taking or noting the readings.
- The consumer having "Prepaid Electricity Meter" can recharge to desirable amount like Rs20, Rs30. It lowers the transportation cost as it is possible to recharge the account of user all over the India. Also if needed we can set a mini printer to "prepaid electricity meter" to get printed bills.
- This project can be executed properly in remote areas is well. Considering the future magnifications or improvements the system will easily acceptable for 3 phase electricity distribution administration.

## **CHAPTER 4**

### **MATERIALS AND METHODS**

#### **4.1 COMPONENTS USED**

1. Arduino UNO
2. Optocoupler 4N35
3. Single channel Relay driver
4. Esp12E nodemcu module
5. Analogue Energy Meter
6. Potentiometers
7. Load (Bulb)
8. Bulb holder
9. 12V Adapter
10. Resistors, Connecting wires

##### **4.1.1 ARDUINO UNO:**

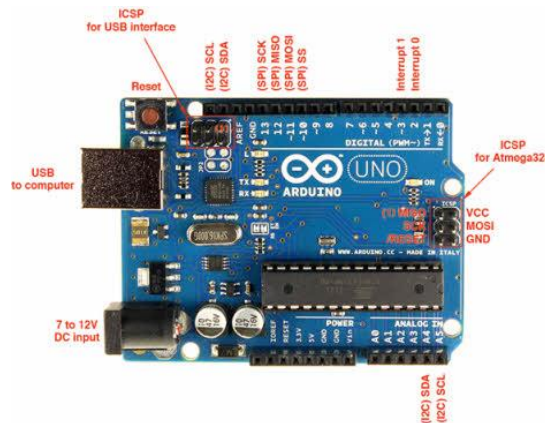
The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, that may be interfaced to various circuits and is programmable with the Arduino IDE, via a type B USB cable. It can be powered by the USB cable or by an external 9- volt battery, though it accepts voltages between 7 and 20 volts.

##### **Features of Arduino Uno Board**

The features of Arduino Uno ATmega328 includes the following.

- The operating voltage is 5V
- The recommended input voltage will range from 7v to 12V
- The input voltage ranges from 6v to 20V
- Digital input/output pins are 14
- Analog i/p pins are 6
- DC Current for each input/output pin is 40 mA
- DC Current for 3.3V Pin is 50 mA

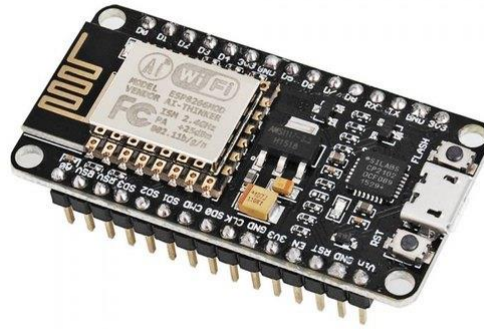
- Flash Memory is 32 KB
- SRAM is 2 KB
- EEPROM is 1 KB
- CLK Speed is 16 MHz



**Fig:4.1- Arduino UNO**

#### **4.1.2 Esp12E nodemcu MODULE**

NodeMCU is an open source development board and firmware based in the widely used ESP8266 -12E WiFi module. It allows you to program the ESP8266 WiFi module with the simple and powerful LUA programming language or Arduino IDE. Ai-Thinker's ESP-12E is a Wi-Fi Module based on ESP8266EX SoC. The ESP8266EX SoC is a Wi-Fi chip based on Tensilica's L106 Diamond 32-bit Processor and an integrated Wi-Fi MAC, with support for full TCP/IP Stack. Since it has a Microcontroller (in the form of Tensilica's L106 Diamond), the ESP-12E can be used as either a stand-alone device with its Wi-Fi connectivity and GPIO Pins or it can be used as a Wi-Fi adapter for other microcontrollers like Arduino, for example, through UART interface.



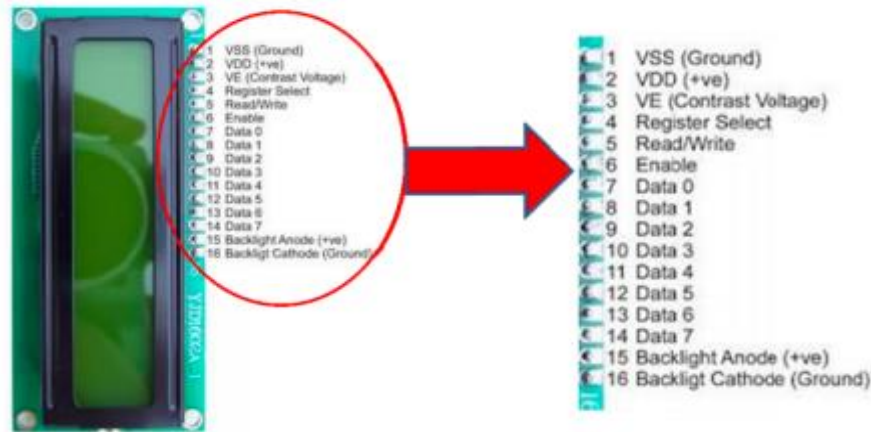
**Fig:4.2- Esp12E nodemcu MODULE**

### **4.1.3 16X2 LCD**

An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data.

Command register stores various commands given to the display. Data register stores data to be displayed. The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register. In your Arduino project Liquid Crystal Library simplifies this for you so you don't need to know the low-level instructions. Contrast of the display can be adjusted by adjusting the potentiometer to be connected across VEE pin.

### **LCD PIN OUT**



**Fig:4.3- LCD Output**

#### 4.1.4 OPTOCOUPLER

- The 4N35 is an optocoupler for general purpose application. It consists of gallium arsenide infrared LED and a silicon NPN phototransistor.
- What an optocoupler does is to break the connection between signal source and signal receiver, so as to stop electrical interference.
- An opto-isolator (also called an optocoupler, photocoupler, or optical isolator) is an electronic component that transfers electrical signals between two isolated circuits by using light.
- Opto-isolators prevent high voltages from affecting the system receiving the signal.



**Fig:4.4- Optocoupler**

#### 4.1.5 SINGLE CHANNEL RELAY DRIVER

The Single Channel Relay Module is a convenient board which can be used to **control high voltage, high current load such as motor, solenoid valves, lamps and AC load**. It is designed to interface with microcontroller such as Arduino, PIC and etc..



**Fig:4.5 Single channel relay driver**

#### **4.1.6 ANALOGUE ENERGY METER**

- An analog power meter is a device that features a printed display to indicate any electrical parameter.
- An example could be the energy consumed by a typical business, or electrical device. Also called an electromechanical meter, these offer a simple to read display.
- The energy meter is an electrical measuring device, which is used to record Electrical Energy Consumed over a specified period of time in terms of units.



**Fig:4.6- Analog Energy Meter**

## **4.2 SOFTWARES USED**

### **4.2.1 ARDUINO IDE**

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with

buttons for common functions and a series of menus. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

### 4.2.2 MQTT Mobile Application

MQTT is a flexible and easy-to-use technology that **provides effective communication within an IoT system**. So, no wonder there are many popular IoT applications which use MQTT protocol. For example, IBM Watson IoT Platform uses MQTT as the main communication protocol.

## 4.3 Methodology

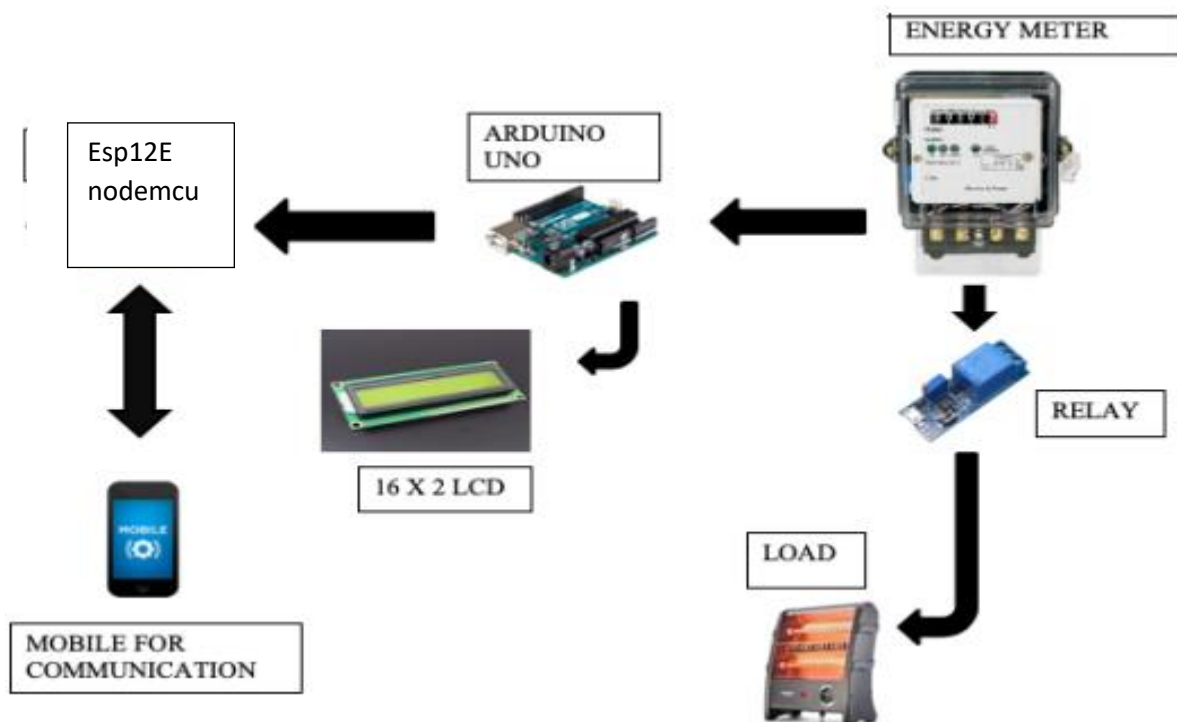


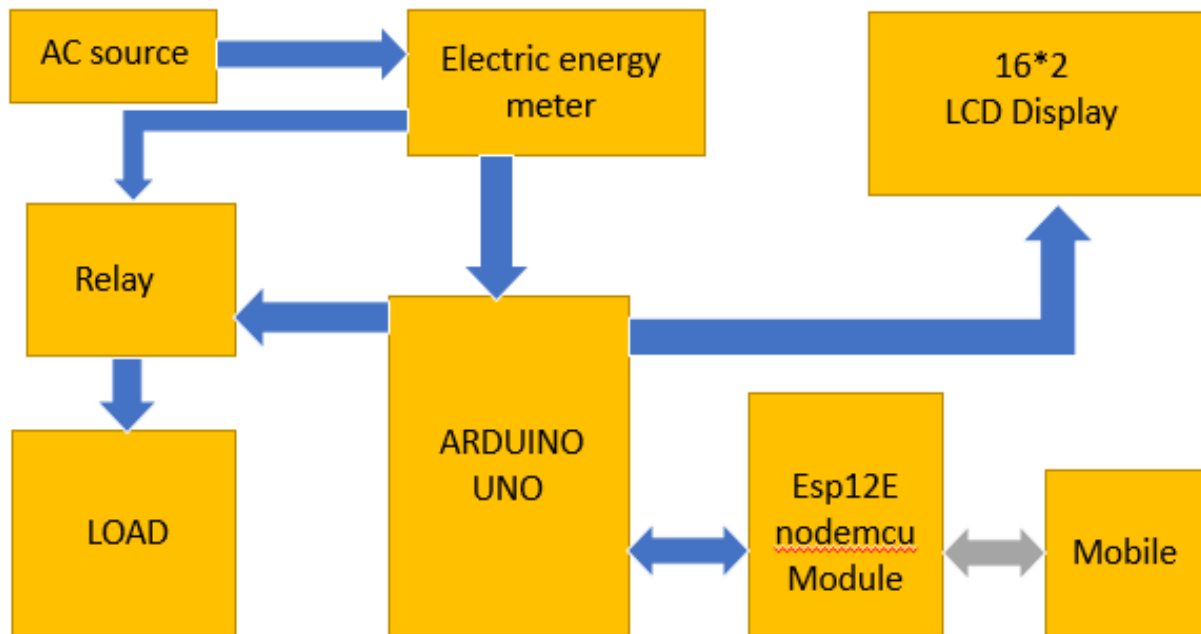
Fig:4.7- Flow Diagram

### 4.3.1 WORKING PRINCIPLE

This model has the Arduino UNO as CPU. The entire system is interfaced with Arduino UNO. The Esp12E nodemcu module is serially connected with the controller which is used as communication module between User and provider. The module uses its own network for the transfer of data. Special coding in Arduino is used for programming AT89S51 microcontroller. The relay is used as switching device to cut off and restore power supply. The LCD is interfaced to microcontroller using parallel

connection. In this project the Microcontroller based system continuously measures the readings and the current meter reading can be sent to the Electricity department on request. This system also can be used to cut off the power supply to the house in case of non-payment of electricity bills. This modem with SIM card is essential for each energy meter.

#### 4.3.2 BLOCK DIAGRAM:



**Fig:4.8-Block diagram**

#### Block Diagram Explanation:

The AC source is the mainline which is connected directly to the Analogue energy meter. From the Energy meter the load line is connected to the load which is nothing but any electrical appliance which we use. The energy pulse is generated which is obtained with the help of LDR sensor is given to the Arduino UNO which is connected to the relay and LCD which has 16\*2 Display. The Arduino is also connected to the ESP12E nodemcu module. This module is responsible for the wireless transmission between the setup and mobile.

### 4.3.3 Work flow

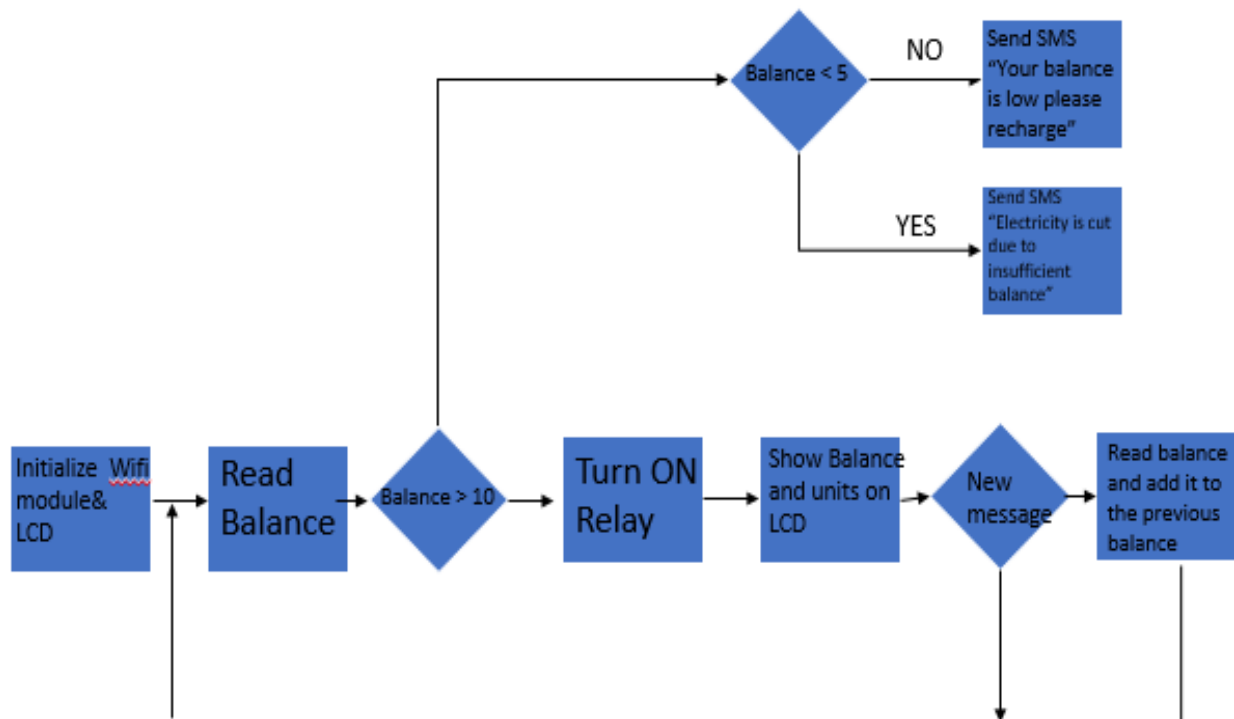


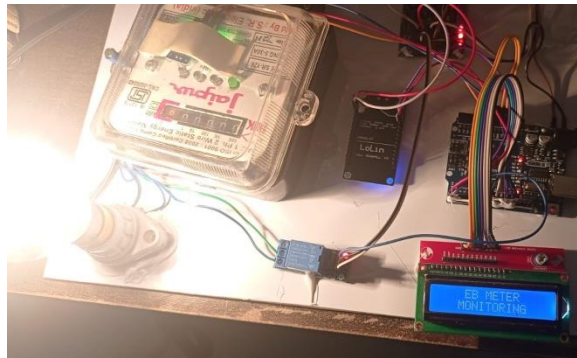
Fig:4.9Methodology Diagram

### Methodology Explanation

When the module is started with the initialization of the Wifi module (Esp12E nodemcu) and the LCD takes place first. Then it will read the amount of balance available in the module and check for the balance. A threshold value is set in the module here for example we have taken a balance greater than 10. If balance is not greater than 10 it will check another condition if balance is less than 5 or not. If the balance is less than 5 then the power will shut automatically. If the balance is less than 10 and greater than 5 the setup will send an alert message to the user via the Esp12E nodemcu module. If balance is greater than 10 the relay will turn ON and units of electricity consumed so far and the balance will be shown on the LCD screen. If want the user can send a new message to the setup and add the amount recharged to the balance, if not the process will continue.

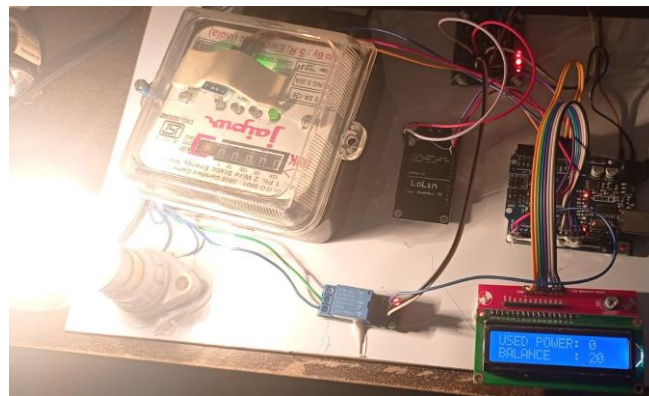
## CHAPTER –5

### RESULTS



**Fig : 5.1 Initialization of Module**

Initialization of the setup –The LCD displays the Monitoring setup has started.

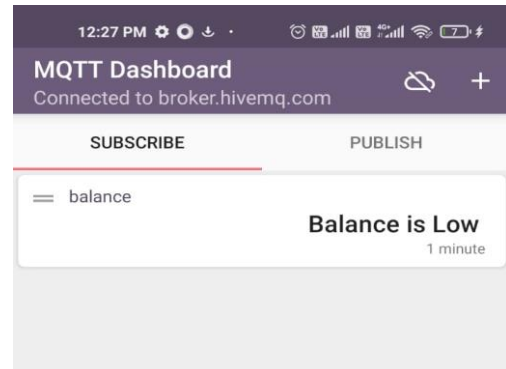


**Fig : 5.2 Initial balance and units consumed**

Initially the Energy units consumed is zero is displayed on the LCD and the initial balance is taken as 20. Here we have considered one unit is equal to one blink on energy meter and one unit costs 2 which is displayed.



**Fig : 5.3 Low Balance displayed in LCD**



**Fig : 5.4 Low balance alert in mobile**

When the Balance is below 5 it will send an alert message to the user through a cloud based application called MQTT. This will let the user to decide whether to recharge or not.

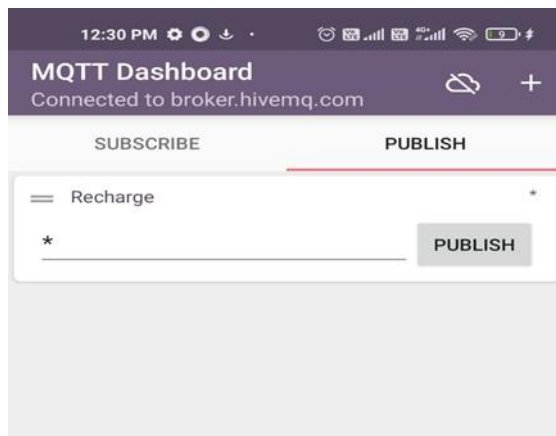


**Fig : 5.5 Load is switched OFF Due to zero balance**

If the user decides not to recharge the power will automatically shut down here leaving the bulb OFF and showing the user the so far consumed power.



**Fig : 5.6 Recharged amount is added**



**Fig : 5.7 Sending a message to the previous balance module to recharged**

Since this is a prototype we have send a character ( \* ) to the module instead of exact money to the setup so that it will get recharged before the power is getting cut down. we can recharge with publish option for number of times.

## **CHAPTER 6**

### **SUMMARY AND CONCLUSIONS**

- By using this project, we can reduce the manual effort to take the reading from the energy meter which is cost effective.
- It is user friendly and we can enhance this project, in which an electricity department can send message to the consumer about the billing information.
- It reduces man power. The recharge can be done on the daily basis and it sends the alert message whenever the consumer exceeds the recharged days which the consumer can easily pay their bills on time.
- The prototype will be able to introduce the billing system to the customers, get the power consumption data from smart meter, keep the data in centralized database and generate the report.

## REFERENCES

- [1] Bharathy D, DhivyaC, Monisha A, Rathipriya S, Sikkandhar Batcha J- Smart Energy Meter Monitoring over IOT- **International Journal for Modern Trends in Science and Technology (IJMTST)**- Vol.06, Issue 06 -June 2020 - page no.[520-523]
- [2] Manisha Tejwani, Prachi Rane, Moin Syed, Oyesh Patel, Pragyan Pandey - lot Based Smart Energy Meter Monitoring and Billing System- **International Journal of Innovative Technology and Exploring Engineering (IJITEE)**- Vol.9 ,Issue 06 - April 2020 -page no.[1480-1483]
- [3] NaziyaSulthana, Rashmi, Prakyaathi N, Bhavana S - Smart Energy Meter and Monitoring System using IoT-**International Journal of Engineering Research and Technology (IJERT)**- Vol. 8, Issue 14, 2020- page no.[50-53]
- [4] Nitesh D.Bhavsar, Mansi D.Mahajan, Omkar V.Joshi, D.Y. Thorat- IoT Based Prepaid Energy Meter-**International journal of research and technology(IJRT)**- Vol.4 ,Issue 3, 2019- page no.[16-18]
- [5] Ranganayaki V, Nithish Kumar S , Subash, Mahaprabhu- IOT BASED PREPAID ENERGY METER- **International Journal of Creative Research Thoughts (IJCRT)**-Volume 9, Issue 7 July 2021-page no[219-224]
- [6] Rishabh Jain ,Sharvi Gupta , Chirag Mahajan , Ashish Chauhan- IOT based Smart Energy Meter Monitoring and Controlling System- **International Journal of Research in Electronics and Computer Engineering (IJRECE)** - Vol. 7 ,Issue 2 , 2019- page no.[1600-1604]
- [7] SACHIN TYAGI, ABHINAV SINGH, HARSHITA SINGH, KANIKA RAWAT, SANDEEP BHATIA - IOT Based Smart Energy Meter- **International Journal of Innovative Research in Technology. (IJIRT)** - Vol. 8.Issue 3, August 2021- page no.[280-282]
- [8] Sunil kumar.C, Chandra Mohan Reddy.P, Venkat Krishna.K, Dr.Shelejkhera- IoT Based Smart Energy Meter Monitoring and Theft Detection- **International Journal of Creative Research Thoughts (IJCRT)**-Volume 9, Issue 4 April 2021-page no[3598-3603]
- [9] USMAN SALEEM.M ,REHAN USMAN.M, and MUSTAFA SHAKIR - Design, Implementation, and Deployment of an IoT Based Smart Energy Management System- **Institute of Electrical and Electronics Engineers(IEEE)**- Vol. 9 ,Issue 4, 2021- page no.[59649-59664]

[10] Vishal Kumar, Tanishq Sharma, Abu Farhan - IoT Based Smart Energy Meter  
- IJERECE -Vol 8, Issue 9, September 2021 - page no.[48-52]

[11] <https://github.com>

[12] <https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/>

[13] <https://www.ieee.org>

[14] <https://www.instructables.com/id/GSM-SIM900A-With-Arduino/>

[15] <https://circuitdigest.com>

[16] <https://www.arduino.cc/en/guide/environment>

[17] <https://iot-analytics.com/10-internet-of-things-applications/>

## APPENDIX

### Code for Arduino

/\*

Analog input, analog output, serial output

Reads an analog input pin, maps the result to a range from 0 to 255 and uses the result to set the pulse width modulation (PWM) of an output pin. Also prints the results to the Serial Monitor.

The circuit:

- potentiometer connected to analog pin 0.

Center pin of the potentiometer goes to the analog pin.

- side pins of the potentiometer go to +5V and ground

- LED connected from digital pin 9 to ground

created 29 Dec. 2008

modified 9 Apr 2012

by Tom Igoe

This example code is in the public domain.

<http://www.arduino.cc/en/Tutorial/AnalogInOutSerial>

\*/

// These constants won't change. They're used to give names to the pins used:

#include <LiquidCrystal.h>

LiquidCrystallcd(13, 12, 11, 10, 9, 8);

const int analogInPin = A0; // Analog input pin that the potentiometer is attached to

unsigned int used\_power=0, remaining\_balance = 20;

char input[5];

int i = 0, flag=0, balance\_flag=0, send\_flag=0;

int sensorValue = 0, sensorValue\_previous = 0, timer = 0; // value read from the pot

```

void setup() {
    // initialize serial communications at 9600 bps:
    Serial.begin(9600);
    pinMode(5, OUTPUT);
    digitalWrite(5, HIGH);
    lcd.begin(16, 2);

    delay(1000);

    lcd.setCursor(0, 0);
    lcd.print("  EB METER  ");
    lcd.setCursor(0, 1);
    lcd.print("  MONITORING  ");

    delay(5000);

    lcd.setCursor(0, 0);
    lcd.print("          ");
    lcd.setCursor(0, 1);
    lcd.print("          ");

    lcd.setCursor(0, 0);
    lcd.print("USED POWER:");
    lcd.setCursor(0, 1);
    lcd.print("BALANCE  :");

    delay(5000);

    TCCR1A = 0; // set entire TCCR1A register to 0
    TCCR1B = 0; // same for TCCR1B
    TCNT1 = 0; // initialize counter value to 0
    // set compare match register for 1hz increments
    OCR1A = 15624; // = (16*10^6) / (1*1024) - 1 (must be <65536) // 100ms timer (15624 for 1 second)
    // turn on CTC mode
    TCCR1B |= (1 << WGM12);
    // Set CS10 and CS12 bits for 1024 prescaler

```

```

TCCR1B |= (1 << CS12) | (1 << CS10);
// enable timer compare interrupt
TIMSK1 |= (1 << OCIE1A);

sei();//allow interrupts
}

void loop() {
    // read the analog in value:
    sensorValue = analogRead(analogInPin);
    // map it to the range of the analog out:

    // print the results to the Serial Monitor:
    // Serial.print("sensor = ");
    // Serial.println(sensorValue);
    timer++;
    if(timer >= 2)
    {
        timer = 2;
        flag = 1;
    }
    if(flag == 1)
    {
        if(sensorValue< (sensorValue_previous - 30))
        {
            used_power++;
            remaining_balance = remaining_balance - 2;
            // Serial.println("*");
        }
        sensorValue_previous = sensorValue;
    }

    // wait 2 milliseconds before the next loop for the analog-to-digital
    // converter to settle after the last reading:
    delay(100);
}

```

```

ISR(TIMER1_COMPA_vect)
{
  if(Serial.available() > 0 )
  {
    //    Serial.println("success");

    char a = Serial.read();
    if(a == '*')
    {
      balance_flag = 1;
      send_flag = 0;
    }
  }
  if(balance_flag == 1)
  {
    //  Serial.println("success");
    remaining_balance = remaining_balance + 20;
    balance_flag = 0;
  }
  if((remaining_balance<= 5) && (send_flag == 0))
  {
    send_flag = 1;
    if(send_flag == 1)
    {
      Serial.println('1');
      //  Serial.println('#');
    }
  }
  if(remaining_balance<= 1)
  {
    digitalWrite(5, LOW);
  }
  else if(remaining_balance> 1)
  {
    digitalWrite(5, HIGH);
  }
}

```

```

lcd.setCursor(12, 0);
lcd.print("  ");
lcd.setCursor(12, 0);
lcd.print(used_power);
lcd.setCursor(12, 1);
lcd.print("  ");
lcd.setCursor(12, 1);
lcd.print(remaining_balance);
}

```

### Code for ESP Module

```

#include <ESP8266WiFi.h> // Enables the ESP8266 to connect to the local network (via WiFi)

#include <PubSubClient.h> // Allows us to connect to, and publish to the MQTT broker

const int ledPin = 2; // This code uses the built-in led for visual feedback that the button has been pressed

//const int buttonPin = 13; // Connect your button to pin #13


// WiFi

// Make sure to update this for your own WiFi network!

const char* ssid = "sathya";

const char* wifi_password = "sathya@123";


// MQTT

// Make sure to update this for your own MQTT Broker!

const char* mqtt_server = "broker.hivemq.com";

const char* mqtt_topic = "sathya/recharge";

const char* mqtt_topic1 = "sathya/balance";

//const char* mqtt_topic3 = "stjoseph/alert";

//const char* mqtt_topic4 = "stjoseph/Date";

//const char* mqtt_topic5 = "stjoseph/live_usage";

// The client id identifies the ESP8266 device. Think of it a bit like a hostname (Or just a name, like Greg).

const char* clientID = "sathya";


// Initialise the Pushbutton Bouncer object

//Bounce bouncer = Bounce();


String no_of_sensor, data, sensor, sensor1, sensor2, sensor3, sensor4, sensor5, sensor6;

```

```

char char_data[50], string[50], lpm1[10], con1[10], alert[10], Date[10], live_usage[10];
char automation;
char time_value[50];
String time_data;
int i = 0, flag = 0, data_sent = 0, data_received = 0;
unsigned int get_data = 0, x = 0, y = 0, z = 0, x1 = 0, y2 = 0, z1 = 0, a = 0, b = 0, c = 0, d = 0, e = 0, f = 0;

// Initialise the WiFi and MQTT Client objects
WiFiClient wifiClient;
PubSubClient client(mqtt_server, 1883, wifiClient); // 1883 is the listener port for the Broker

void ReceivedMessage(char* topic, byte* payload, unsigned int length) {
    // Output the first character of the message to serial (debug)
    Serial.print((char)payload[0]);
    // flag = 0;
}

bool Connect() {
    // Connect to MQTT Server and subscribe to the topic
    if (client.connect(clientID)) {
        client.subscribe(mqtt_topic);
        return true;
    }
    else {
        return false;
    }
}

void setup() {
    pinMode(ledPin, OUTPUT);
    // pinMode(buttonPin, INPUT);
    // Switch the on-board LED off to start with
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
}

```

```

digitalWrite(ledPin, HIGH);

// Setup pushbutton Bouncer object
//  bouncer.attach(buttonPin);
//  bouncer.interval(5);

// Begin Serial on 115200
// Remember to choose the correct Baudrate on the Serial monitor!
// This is just for debugging purposes
Serial.begin(9600);
Serial.print("Connecting to ");
Serial.println(ssid);
// Connect to the WiFi
WiFi.begin(ssid, wifi_password);
// Wait until the connection has been confirmed before continuing
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
// Debugging - Output the IP Address of the ESP8266
Serial.println("WiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
// Connect to MQTT Broker
// client.connect returns a boolean value to let us know if the connection was successful.
// If the connection is failing, make sure you are using the correct MQTT Username and Password (Setup Earlier
in the Instructable)
client.setCallback(ReceivedMessage);
if (client.connect(clientID)) {
Serial.println("Connected to MQTT Broker!");
}
else {
Serial.println("Connection to MQTT Broker failed...");
}
if (Connect()) {
Serial.println("Connected Successfully to MQTT Broker!");
}
}

```

```

    else {
Serial.println("Connection Failed!");
    }

}

String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = {0, -1};
    int maxIndex = data.length() - 1;

    for (int i = 0; i<= maxIndex&& found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
strIndex[0] = strIndex[1] + 1;
strIndex[1] = (i == maxIndex) ? i + 1 : i;
        }
    }

    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

void loop()
{
    if (!client.connected()) {
Connect();
    }

    // client.loop() just tells the MQTT client code to do what it needs to do itself (i.e. check for messages, etc.)
client.loop();

    if (Serial.available() > 0)
    {
        char a = Serial.read();
if(a == '1')
        {

```

```
//    flag = 1;
client.connect(clientID);
delay(100); // This          delay ensures that client.publish doesn't clash with the client.connect call
client.publish("balan", "Balance is Low");
    //    Serial.println("published data");
//    flag = 2;
    }
    }
delay(100);
    }
```