



People`s Democratic Republic of Algeria



Ministry of Higher Education and Scientific Research

University of Echahid Hamma Lakhdar - El Oued

Faculty of Technology

Department of Electrical Engineering

Dissertation

ACADEMIC MASTER

Domain: Science and Technology

Division: Telecommunications

Specialty: Telecommunication Systems

Entitled:

Smart Energy Monitoring and Management System

Dissertation Submitted in Partial Fulfillment of the Requirements for the Master

Degree in Telecommunication Systems

Presented by:

DEMAIDA HOUSSAM EDDINE

BELMABROUK REDOUANE

Supervisor

HIMA ABDELKADER

Academic Year: 2023/2024

Thanks

Above all, we thank ALLAH the Almighty for having us given the courage, will and patience to complete this present work in better conditions.

First of all, we would like to thank our supervisor Mr Hima AbdElkader, for agreeing to supervise us, and for the theme he proposed to us. As well as all these remarks constructive ideas that allowed us to deepen the scientific knowledge.

We would particularly like to thank our teachers for their precious help, their advice and for the time they devoted to us throughout this work.

Without forgetting we thank Ben Mabrouk Redouane for his contribution and help which was essential to us for the completion of this work.

Thank you to our parents for listening to us and motivating us.

Thank you to all of our friends, who were there during the periods of doubt and stress.

Thanks again everyone...

DEDICATION

I dedicate this work to:

*My very dear parents who guided me during the
most difficult moments of this long journey,
my mother who was by my side and supported me
throughout my life,*

*and my father who sacrificed
all His life in order to see me become what I am,*

My brothers and my sisters.

*To all my friends: Redouane, adnan, Assem,
To my dear fiancée All those who love me and
whom I love.*

Demaida Houssam Eddine

DEDICATION

*I dedicate this work to My beloved
father'-Mohammed-' who always guided
me on the path of life*

*My dear mother'-SAÏDA-' who
sacrificed herself for me and spared no
effort to make me happy My brothers
and sisters all my friends To whom my
heart loves.*

Belmabrouk redouane

Table of Content

Table of Figures	
ABSTRACT	
Résumé	
ملخص	
General Introduction	
CHAPTER 1	1
1-2.DIFINITION OF (SEMMS)	2
1-3. Objectives of SEMMS	2
1-3-1 The main objectives:.....	2
1-3-2 Specific Objectives:	3
1.4-1 Components	4
1.4-2 Architecture	5
1.4.3 Applications of (SEMMS)	6
1.5 Litarture review about (SEMMS)	7
1.5.1 (SEMMS) background	7
1.5.2 Previous work	8
1.5.3 Gaps in Current Research about (SEMMS)	11
1.6.1 Energy Efficiency	12
1.6.2 Cost Savings	12
1.6.3 Environmental Impact	12
1.6.4 Enhanced Control and Convenience	13
1.7 Conclusion	13
CHAPTER 2	14
2.1 Introduction	14
2.2 Hardware Design	14
2.2.1Connections:	14
2.2 Key components	15
2.2.1Node MCU ESP8266	15
2.2.2 ACS712 Current sensor	17
2.2.3Arduino Uno.....	20
2.3Software Programming	24

2.3.1 Arduino IDE and Node MCU	24
2.3.2 Application software	24
2.4 Conclusion	28
CHAPTER 3	29
3.1 Introduction	29
3.2 Block Diagram of the (SEMMS)	29
3.3 Calculate The Energy Consumed in kwh	31
3.4 The Communication Protocol	31
3.4.1 Setting up an AdaFruit account for communication.....	31
3.4.2 Creating Applet in IFTTT for Triggering SMS/Email for Energy Meter	35
3.4.3 AndroidAppforMonitoringElectricity Consumption	38
3.5 The Creation and Practical Results of The (SEMMS)	40
3.5.1 Practical Realization of (SEMMS)	40
3.5.2 Practical Results	40
3.6 Conclusion	41
General Conclusion	42
References	43
Annexe A	43

Table of Figures

Figure 1:General Plan of Smart Energy monitoring and management system.....	2
Figure 2:Circuit diagram for IoT based Energy Meter using Arduino andNodeMCU	5
Figure 3:Realistic diagram of smart energy meter and management system	6
Figure4 Application areas of (SEMMS)	7
Figure 5 Automatic meter Reading system.....	9
Figure 6 Block diagram of IR remote controlled home automation.....	10
Figure 7 SMS based system	11
Figure 8 Design of Node MCU	16
Figure 9 Pin out of Node MCU ESP8266	17
Figure 10 Diagram of ACS 712 mode 30A	18
Figure 11 Pin out of ACS712 Current sensor	19
Figure12 Principle of the hall effect.....	20
Figure 13 Arduino uno Atmega328	20
Figure 14 Pin out diagram of Arduino UNO	23
Figure 15 Programming the Arduino board on the computer.....	23
Figure 16 Programming Adafruit program with Arduino platform.....	24
Figure 17:MQTT-Broker Diagram.....	25
Figure 18:Adafruit IO logo	27
Figure 19 Components diagram of the project.....	29
Figure 20 Flowchart of the Project.....	30
Figure 21 Create an Account on the Adafruit Platform	31
Figure 22 Go to the dashboards section	32
Figure 23 Creating a new dashboard.....	32
Figure 24 Adding name and description to the dashboard.....	32
Figure 25 Note the AIO keys of the account.....	33
Figure 26 Create a bloc for display the energy	33
Figure 27 Giving name to the feed.....	34
Figure 28 Adjust feed settings.....	34
Figure 29 Create a new feed to display bill.....	34
Figure 30 Click on New applet	35
Figure 31 Search AdaFruit	35
Figure 32 Setting up a feed on adafruit.....	36
Figure 33 Determing a threshold to the feed.....	36

Figure 34 Adding G-mail credentials to our feed 37

Figure 35 Click on **send yourself an email**..... 37

Figure 36 Create an g-mail recipe with there options 37

Figure 37 Finiching the g-mail recipe 38

Figure 38 Click on "+" sign on the app and create dashboard with there options
..... 38

Figure 39 Taking the username and active key of our feed from Ada fruit
account..... 39

Figure 40 Make a subscription to both elctricity meter and bill feeds..... 39

Figure 41:Structure of smart energy meter 40

Figure 42:the data displayed in the smartphone application..... 40

Figure 43:the consumed power by the load 41

Table of Abbreviations

-SEMMS: Smart Energy Monitoring and Management System

-SEM: Smart Energy Meter

-IOT: Internet Of Things

-AC: Alternative Current

-DC: Continous Current

-IFTT: If This Than That Application

ABSTRACT

The energy meter that we use in our house to measure daily consumption is a device that is not connected to the Internet, so it has to be manually checked. There are power meters in every house. As the number of houses increases, the power meters will increase dramatically, but labor will be required. The main problem with old energy preparations is that customers do not know how much they use energy accurately, leading to higher electricity bills. Power outages in villages, causing financial problems for consumers. To avoid these problems, a smart energy meter does an important job. This service will make the power company's clients happy because they can track their use of energy over the Internet. In this project, we will establish an intelligent energy control system using the Arduino IDE program according to a set of algorithms that are programmed by the latter to measure energy consumed at the household or factory level. A short text message is sent to reload the meter. Given the use of the Internet, things can monitor the energy level on the Internet. It can also be accessed online if you have a mobile phone, you can only call home. An automatic notice is also sent as a text message to the user 's mobile phone according to the system's balance. Thus, energy loss is reduced, bills are rationalized through this project, energy-intensive devices are identified, and orders are given either to close or to regulate the distribution of energy from important devices to less important ones.

Résumé

La quantité d'énergie que nous utilisons dans notre maison pour mesurer la consommation quotidienne est un appareil hors connexion à Internet, il faut donc le calculer manuellement. Il y a un compteur d'énergie dans chaque maison. Avec l'augmentation du nombre de maisons, les coûts d'énergie augmenteront de manière exponentielle, mais les constructeurs auront besoin de main-d'œuvre. Le principal problème avec les calculs d'énergie anciens est que les clients ne savent pas exactement combien ils consomment d'électricité, ce qui entraîne des factures élevées. Les coupures d'électricité dans les villages causent des problèmes financiers pour les consommateurs. Pour éviter ces problèmes, les numéros d'énergie intelligente remplissent une fonction importante. Ce service rendra les clients de l'entreprise électrique heureux car ils peuvent suivre leur consommation d'énergie en ligne. Dans ce projet, nous allons créer un système intelligent de surveillance de l'énergie à l'aide du logiciel Arduino IDE en fonction d'un ensemble de graphismes programmés pour mesurer la consommation énergétique au niveau des maisons ou des usines. Un petit message texte est envoyé pour recharger les numéros. En raison de l'utilisation d'Internet des objets, il peut surveiller les niveaux d'énergie sur Internet. Vous pouvez également accéder à Internet si vous avez un téléphone portable, vous ne pouvez appeler à la maison. Une notification automatique est également envoyée sous forme de message texte au téléphone portable de l'utilisateur en fonction de la tarification du système. Ainsi, les pertes d'énergie sont réduites, les factures sont distribuées par ce projet, les appareils qui consomment beaucoup d'électricité sont identifiés et les ordres sont donnés soit pour les fermer, soit pour organiser la distribution de l'énergie des appareils importants aux appareils moins importants.

ملخص

عداد الطاقة الذي نستخدمه في منزلنا لقياس الاستهلاك اليومي هو جهاز غير متصل بالإنترنت، لذا يجب فحصه يدويًا. توجد عدادات الطاقة في كل منزل. ومع زيادة عدد المنازل، فإن عدادات الطاقة ستزيد بشكل طردي، ولكن سوف يتطلب الأمر توفر اليد العاملة. تكمن المشكلة الرئيسية لعدادات الطاقة القديمة في أن العملاء لا يعرفون مقدار استهلاكهم للطاقة بشكل دقيق، مما يؤدي إلى ارتفاع فواتير الكهرباء. انقطاع التيار الكهربائي في القرى، مما يسبب مشاكل مالية للمستهلكين. لتجنب هذه المشاكل، يقوم عداد الطاقة الذكي بوظيفة مهمة. ستجعل هذه الخدمة عملاء شركة الكهرباء سعداء لأنه يمكنهم تتبع استخدامهم للطاقة عبر الإنترنت. سنقوم في هذا المشروع بإنشاء نظام ذكي لمراقبة الطاقة باستخدام برنامج Arduino IDE وفق مجموعة من الخوارزميات التي تبرمج الأخير لقياس الطاقة المستهلكة على مستوى المنازل أو المصانع. يتم إرسال رسالة نصية قصيرة لإعادة شحن العداد. نظرًا لاستخدام إنترنت الأشياء، يمكنه مراقبة مستوى الطاقة على الإنترنت. ويمكن أيضًا أن يتم الوصول عبر الإنترنت إذا كان لديك هاتف محمول، يمكنك الاتصال بالمنزل فقط. ويتم أيضًا إرسال إشعار تلقائي كرسالة نصية إلى الهاتف المحمول الخاص بالمستخدم وفقًا للرصيد الموجود في النظام. وبالتالي يتم تقليل فقدان الطاقة، و ترشيد الفواتير من خلال هذا المشروع، وتحديد الأجهزة التي تستهلك الطاقة بشكل كبير، وإعطاء الأوامر إما بإغلاقها أو تنظيم توزيع الطاقة من الأجهزة المهمة إلى الأجهزة الأقل أهمية.

General Introduction

In recent years, the growing demand for energy and the need for environmental sustainability have led to significant progress in monitoring and energy management technology. With these innovations, intelligent energy monitoring and management systems have become an important solution for improving energy use, reducing costs and promoting sustainable practices.

These systems combine advanced technologies such as the Internet of Things (IoT), big data collection, artificial intelligence(AI) and the cloud to provide comprehensive energy management solutions.

The development and implementation of intelligent energy monitoring and management systems is an important step towards a safe and sustainable energy future. Using advanced technology, these systems allow users to take quick action to manage their energy consumption, ultimately saving money and protecting the environment. As the world continues to struggle with energy, the role of energy systems will become increasingly important in shaping a sustainable energy future.

CHAPTER 1

Generalities About Smart Energy Monitoring system Using Arduino

1-1.INTRODUCTION

According to a survey in 2017 about electricity about 13% of world population does not receive electricity. In countries like south Sudan about 268 days of black out of electricity. And there are several countries in African continent people live more than 100days without electricity in a year. But we have sufficient amount of electricity but we all of us waste without our knowledge. Where several peoples does not take care about wastage electricity. There is a lot of ways that electricity is wasted.In that household wastage of electricity stays in second position after the transportation wastage, we proposed this paper to control the wastage of electricity in home appliances. In our method we implement theArduino controller to reduce the wastage of electricity. This method is cost efficient and easy to implement in our day today life.

.With the help of {Arduino IDE} and some embedded programming algorithms we designed the hardware and programmed the hardware in such a way that reduce wasting power with the help of remote control. Where the microcontroller receive the hexadecimal code from the remote according to the remote signal it will control the appliances in ourhome and reduce the wastage of power in the our day to day life.

1-2.DIFINITION OF (SEMMS)

Smart Energy monitoring and management system is a technology-based system that optimizes energy production, distribution and consumption. It uses advanced software, sensors and communications networks to monitor and control energy use in real time. Its main objectives are to increase energy efficiency, reduce costs and minimize environmental impact by optimizing energy consumption patterns. The most basic smart energy systems include energy-saving air conditioners and smart home appliances. There are also more complex smart energy systems, such as office and commercial building automation systems. For the purpose of centralized monitoring and management of the electricity, lighting, air conditioning, water supply and drainage, fire protection and other equipment of the entire building, a comprehensive system of computer control, management and monitoring is formed.

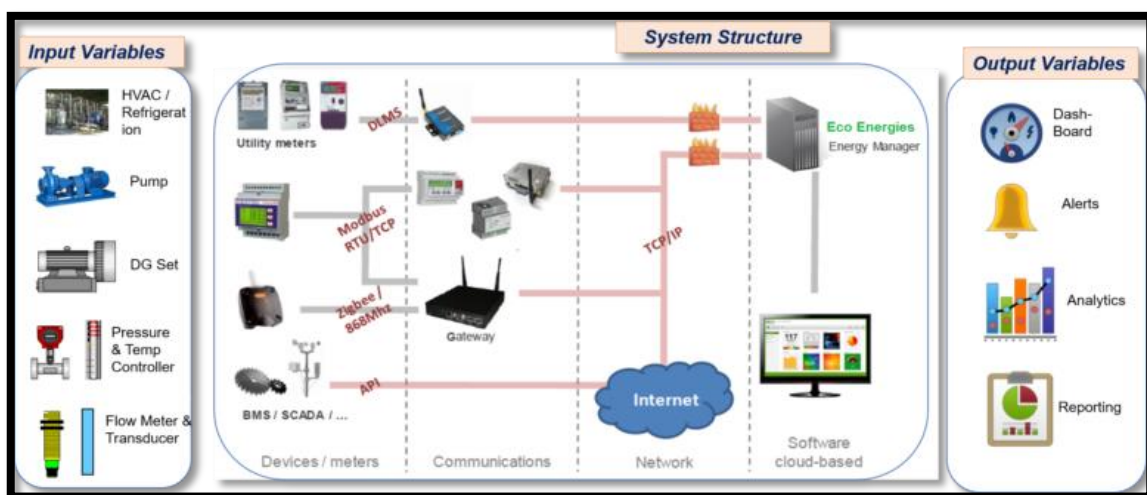


Figure 1:General Plan of Smart Energy monitoring and management system

1-3. Objectives of SEMMS

1-3-1 The main objectives:

The primary objective of an Smart Energy Monitoring and Management System (SEMMS) is to enable organizations to make more informed decisions, aiding in the achievement of their financial, infrastructural, and/or environmental goals.

An SEMMS, whether deliberately implemented or not, serves as an initial step towards decarbonization.

The main objectives of this project are:

- a. Increase public awareness about the impact of high electricity consumption
- b. Encourage the adoption of energy conservation, efficiency, and demand control measures
- c. Demonstrate smart energy metering and automated appliance control
- d. Develop a single solution for monitoring energy consumption and controlling appliances, contributing to the construction of smart homes.

1-3-2 Specific Objectives:

The project will involve the design of a smart energy system that will perform to the following target specifications:

- 1) Intended for devices with constant currents and low voltage (12V, 60Hz)
- 2) Digitally measure consumption of electricity by measuring current, voltage, power factor.
- 3) Digitally measure consumption of electricity by measuring voltage to load sensitivity and energy consumed.
- 4) A mobile application is developed in Android Platform to remotely communicate with the Main Control Energy system.
- 5) Voice recognition and controlling the device power supply
- 6) Wi-Fi enabled appliances/load control switching control system.
- 7) Apply control measures to reduce consumption.
- 8) Control all appliances remotely to save energy.
- 9) Compact design with a dimension of 10.5 in * 8.5 in * 2 in.
- 10) The system is portable.

1.4 Project Components and Architecture

1.4-1 Components

In this project, we used A list of components We mention the following:

I-Battery Source:

A DC source of 12V is utilized to power up the whole circuit.

II-Current Sensor:

A current transformer (CT) is utilized to measure AC and DC current passing through the circuit. The current flowing in the secondary coil is proportional to the current in its primary coil. In this project, we have selected ACS712 current sensor. This sensor comes with three different types(5mA,20mA and 30mA)

In this project we use the 30mA ACS712 type.

III Arduino Uno:

Arduino Uno is **an ATmega328 based microcontroller board**. It has 14 digital I/O pins (**six of** which can be used as PWM **inputs**), six analog inputs, a 16 MHz ceramic resonator, a power **connector**, a USB **connector** and a reset button.

IV. NodeMcu8266:

NodeMCU is an open source hardware platform that includes an open board design. The word "NodeMCU" is an abbreviation of the words "node" and "microcontroller". The term "NodeMCU" refers to the hardware, not the associated development kit. The firmware and configuration of the prototype board is open source , they use a circuit board designed as a two-in-line package (DIP) that connects the USB controller to a small surface board that contains the MCU and antenna. The DIP format allows easy printing on breadboards. The design is based on the ESP-12 module of the ESP8266, a Wi-Fi SoC widely embedded in IoT applications.

1.4-2 Architecture

This work is divided into several parts for easy management and analysis. The most important part of the project is the construction of a smart electric machine that measures and controls the consumption of electricity.

In the first part, the current sensor is used to measure the voltage and current, and then calculate the power that will be displayed on the Arduino microcontroller board. The second phase involves the analysis and implementation of the energy balance system.

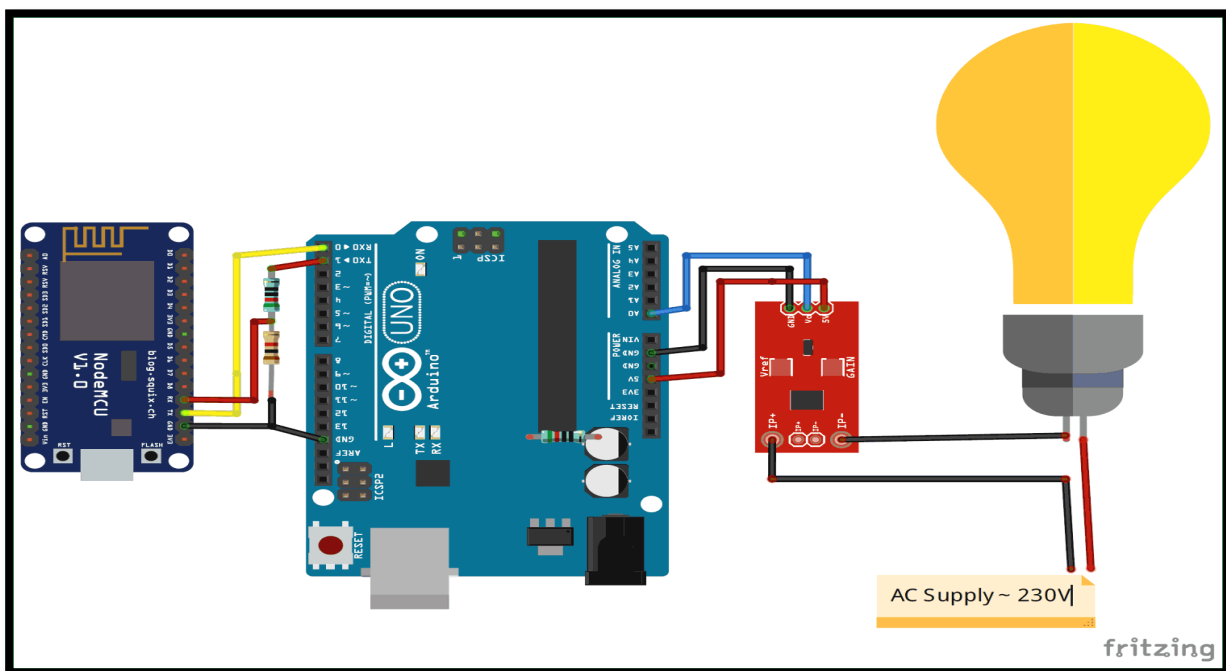


Figure 2: Circuit diagram for IoT based Energy Meter using Arduino and NodeMCU

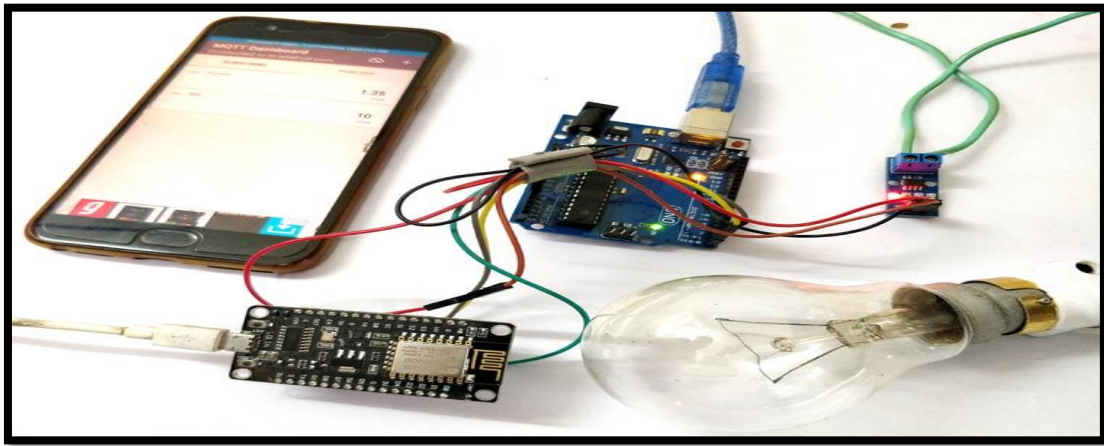


Figure 3:Realistic diagram of smart energy meter and management system

In the third phase, we will use two relays and connected our controller to the Internet using the Node MCU ESP8266,we will connect this system with MQTT broker as **AdaFruit IO platform** and follow the below process to make this IoT Energy Meter. The main task is to use IoT to power our home devices. A block diagram of the project is shown in Figure 2.

In this project, we developed a method to measure the consumed energy by using a current sensor. A block diagram of system is shown in Figure3. These tasks were carried out by creating a modern electric power system of . There are:

- Accurate power measurements
- Comparative study of current transformer and current sensor for accurate calculation
- Work on user interface to control the loads
- Test and analyze each subsystem and make necessary improvements.
- Integrate all subsystems and perform final testing.
- Write final report and presentation.

1.4.3 Applications of (SEMMS)

The (SEEMS) plays important role in many areas It has many uses among them:

- a) Monitor consumption of power and voltage to load sensitivity
- b) Enable smart control of electrical appliances
- c) Controlling Electric Vehicles Charging Rate.
- d) Controlling loads at via an app using hands or voice.
- e) Measure consumption of electricity at homes
- f) Optimal Battery Placement
- g) Controlling renewables Generation

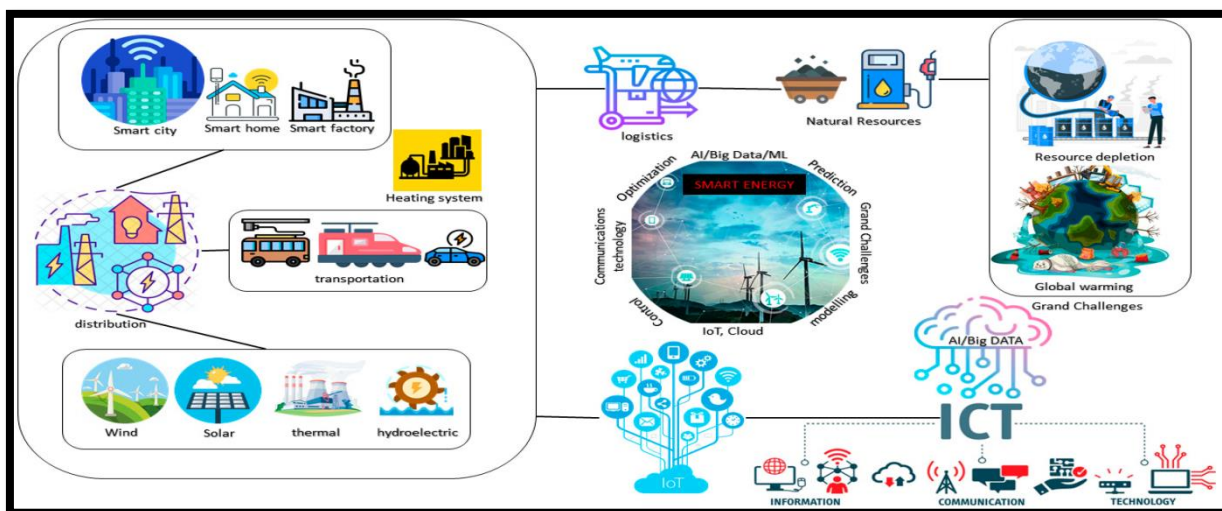


Figure4:Application areas of (SEMMS)

1.5 Litarture review about (SEMMS)

1.5.1 (SEMMS) background

In the smart meter system, power electronics are used to effectively control the devices, prevent interference or theft methods, detect, use and monitor this peak power, time tariffs, error signals and other possible conditions. . The use of "PowerLine Communication", known as PLC or wireless technology, with the appropriate cable to connect to the operator at meters allows the compatibility and effectiveness of the mentioned features on top of the future protocol of the smartgrid.

Modern energy meters are usually digital meters that work in place of the old analog meters that most homes use to record electricity consumption. The details

of the energy consumption can be transmitted to the service provider through a digital consumption meter, unlike the traditional analog meter, which requires the information to be transmitted to the meter reader. home energy usage was recorded for one hour. With a smart meter, monitoring consumption is easy and accurate, so you can make the right decisions about energy or energy management. Some types of meters may receive a interrupt level or allow the equipment to turn on or off the electrical service.

1.5.2 Previous work

In this section, we outline some previous work on the proposed plan. A comparison of the proposed designs can be found in the next section of this chapter.

a)Design of the Automatic Meter Reading System

This project deals with the Automatic Meter Reading (AMR) system which is a programmable integrated with the signal system in the phone as expected automatic system in two ways system , which helps management and in visual reading of service meter, meter reading and management system without human assistance. Efficiency, effectiveness, speed, and accuracy are the distinct advantages of using the AMR system. The entire system depends on the existing telephone network. Therefore, wherever there is a telephone network, this service works. It receives these automatically and electronically to complete the installation and reading of semi-automatic meters (and manual), reading errors, floating charges and recalls [1].

detection and remote control on or off .In the near future, AMR will be popular. The AMR system is now open. The combination of benefits these digital power meters offer makes the modern AMR series more advanced and more flexible than the series of years past. The advent of the latest communication technologies, including continuous cost reduction and competitive market, makes the traditional meter reading system a necessity [4].

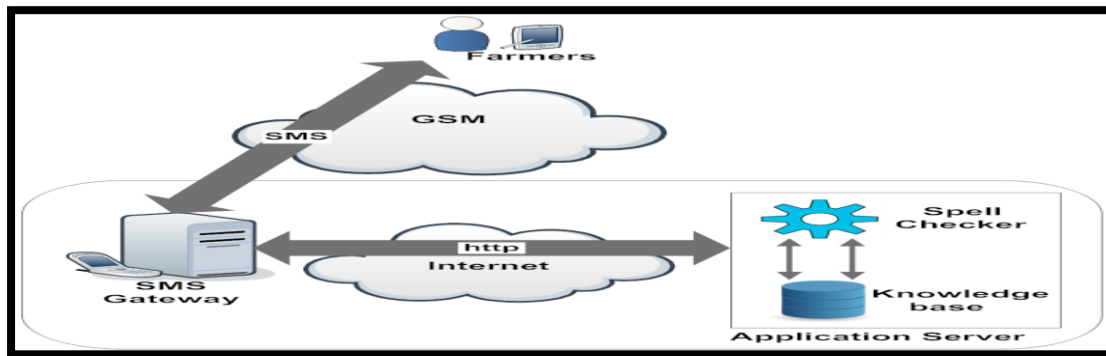


Figure 7:SMS based system

1.5.3 Gaps in Current Research about (SEMMS)

The increase in energy consumption is a concern for the development of energy security, especially in Algeria due to rising tariffs. It is imperative that we explore and implement ways to increase awareness and develop new ways to reduce use and improve implementation. Since the development of energy, the protection of houses and shops is increasing, providing a suitable method for the effective analysis of energy use. On the part of the books in Chapter , more efforts were made and discussed.

The chief concern of smart energy meters critics, besides the cost, is the potential privacy concerns that come with a household's data being transmitted to a supplier. Although smart meters send meter readings to your energy supplier, they do not store your name, address or bank details. Energy firms are adamant that only they can see your data and that information cannot be passed on to a third party without the customer's explicit permission.

During our work, we proposed a simple and cost-effective way to save energy, this system is used to accurately calculate line parameters, it can determine various values such as voltage, current, power, active and passive power, total power and energy consumption. In addition, the system is equipped with a WiFi

module for remote control of the device. This will help customers to monitor the power of their devices and allow them to control home appliances ,this affordable solution is often suitable for indoor application.

1.6 Benefits of (SEMMS)

Smart energy meters are an important part of the development of the smart grid. Promoting the use of smart energy meters is crucial for improving product quality, building electricity savings and creating smart grids.

1.6.1 Energy Efficiency

The (SEMM) system make users directly access the electricity consumption information from the smart energy meter, which increases the user's ability to know the electricity consumption information and makes it easier for users to change the electricity consumption pattern and control the electricity consumption; Also, remote meters reduce on-site readings. The weight of the meter also reduces readability for operators in the field, meaning that the high performance levels are becoming more difficult.

1.6.2 Cost Savings

The design of modern electric meters has changed the situation of different electric meters, which is convenient to reduce the number of standby meters, make storage easier, transportation and maintenance, and reduce the cost of operation, it is very economical and can be easily implemented in residential as well as commercial sites. The total cost of one smart energy meter is very cheap which make it more appropriate solution to be utilized. The electricity providing companies are in need to use such sophisticated system to implement for commercial and residential areas.

1.6.3 Environmental Impact

By optimizing energy usage and reducing wastage, automation and remote monitoring contribute to lower greenhouse gas emissions and a smaller carbon footprint.

The process of operation, type and technology requirements of electricity meters are good for improving the life cycle management system of electricity meters,

further developing the technology of electricity measurement and control system, and improving the control level of consumption meters and work meter.

1.6.4 Enhanced Control and Convenience

Users can control and monitor their energy systems remotely using smartphones, tablets, or computers, providing flexibility and convenience.

By monitoring and controlling energy usage, users can significantly reduce their energy consumption, leading to lower utility bills, automation systems can adjust energy consumption based on real-time data and predefined schedules, ensuring that energy is used efficiently and only when needed.

1.7 Conclusion

In this chapter, we gave generalities about smart energy monitoring and management system. We also gave an overview of the nature of the project intended to be completed by giving its block diagram, as well as explaining how to monitor the energy consumed via Wi-Fi.

In the next chapter, we will go deeper into the project, where we will give an identification card for each of the project subsystems and explain the role of each of them in the project system.

CHAPTER 2

Hardware and Software design of the (SEMMS)

2.1 Introduction

For a long time, it was difficult to create an electric circuit with components (resistance, capacitance, inductance, transistor ...), and its changes are also very powerful, because the creates its circuit to perform special functions. A small change is to go through many tasks such as (calculation, welding, graphic editing,etc.).

But with the development of electronic technology, creating a system based on electronic circuits has become easier and easier thanks to the development board and the programming module that allows the to perform many tasks at the same time.

In this chapter, we discuss the Node MCU8266,arduino uno and ACS712 current sensor and show its features and show a diagram of the project including the components used in the system.

2.2 Hardware Design

This is a circuit diagram for an IoT-based electricity energy meter using an Arduino Uno, a NodeMCU (ESP8266), and a current sensor (ACS712). Below is a detailed description of the circuit components and their connections:

2.2.1Connections:

Arduino Uno:

Power Connections:

The Arduino is powered via its USB port or through its power jack.

Sensor Connections:

A0 Pin: Connected to the output of the ACS712 current sensor (middle pin).

5V Pin: Connected to the VCC pin of the ACS712 current sensor.

GND Pin: Connected to the GND pin of the ACS712 current sensor.

Communication with NodeMCU:

TX (D1) Pin: Connected to the RX (D1) pin of the NodeMCU through a voltage divider made with two resistors to match the voltage levels.

GND Pin: Connected to the GND pin of the NodeMCU.

II/NodeMCU (ESP8266):

Power Connections: Powered via its USB port.

Communication with Arduino:

RX (D1) Pin: Connected to the TX (D1) pin of the Arduino through a voltage divider to ensure safe voltage levels for communication.

GND Pin: Connected to the GND pin of the Arduino to ensure a common ground.

III/ACS712 Current Sensor:

VCC Pin: Connected to the 5V pin of the Arduino.

GND Pin: Connected to the GND pin of the Arduino.

OUT Pin: Connected to the A0 pin of the Arduino.

IP+ and IP- Pins: Connected in series with the AC supply line and the load (bulb).

IV/AC Load:

AC Supply: A 230V AC supply is connected to the ACS712 current sensor. The live wire goes through the sensor, and then to the AC load (bulb).

2.2 Key components

2.2.1 Node MCU ESP8266

1/Definition

The NodeMCU ESP8266 is a microcontroller with an integrated WiFi module. It is very user-friendly, lightweight, and offers superior memory and processing capabilities compared to Arduino. The ESP8266 is ideal for your connected projects that are controlled via a web interface.

To create connected devices, the first step is to connect them to a network like WiFi. The built-in WiFi transmitter and antenna on the microcontroller enable Internet connectivity. This allows you to create a server that hosts a web page, which can be used to remotely control the microcontroller. This page can display values measured by the NodeMCU or control the microcontroller's inputs and outputs.

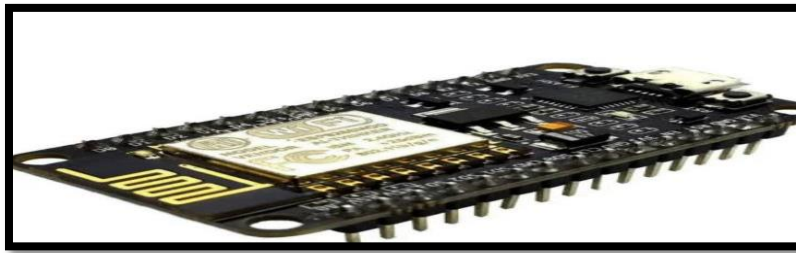


Figure 8:Design of Node MCU

2/Technical Specifications and features

The NodeMCU ESP8266 boasts 12 GPIO pins, with 10 of them supporting PWM. It also features I2C on D1 and D2.

Dimensions: 58 x 31 x 12 mm

Pin count: 15

I/Power Supply:

5 VDC via micro-USB

5 to 9 VDC via Vin pin (integrated regulator)

II/Technical Specifications:

-Microcontroller: ESP8266 ESP-12

-Microprocessor: Tensilica LX106

-Frequency: 80 MHz

-RAM: 64 kB

-Flash Memory: 96 kB

III/Interfaces:

I2C

SPI

UART

IV/Node MCU ESP8266 Features

- 802.11 b/g/n support
- 802.11 n support (2.4 GHz), up to 72.2 Mbps
- Defragmentation
- 2 x virtual Wi-Fi interface
- Automatic beacon monitoring (hardware TSF)
- Support Infrastructure BSS Station mode/SoftAP mode/Promiscuous mode

V/Wireless Connectivity:

Node Mcu ESP8266 uses the WiFi network under the interval of (802.11b/g/n) at 2.4 GHz with an integrated antenna.

VI/Pinout and NodeMCU functions

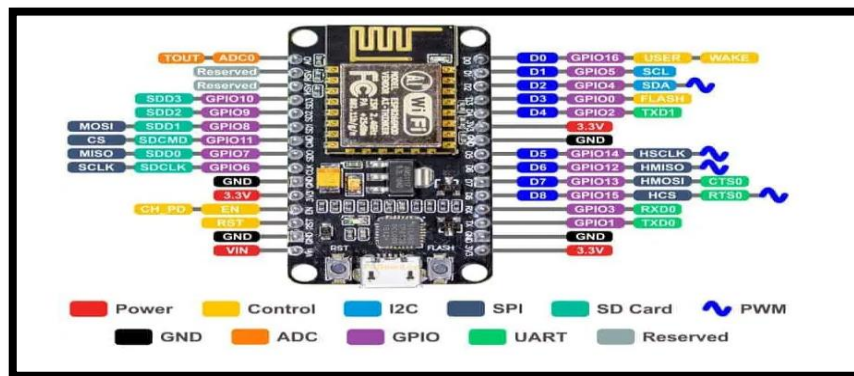


Figure 9:Pin out of Node MCU ESP8266

2.2.2 ACS712 Current sensor

I/ACS712 Definition:

This current sensor board is based on the Allegro ACS712ELCTR-30A bi-directional Hall-effect current sensor chip, which detects both positive and negative flowing currents in the range of -30 Amps to +30 Amps. The board operates at 5V DC, and the current flow through the sensor is converted to an output voltage that starts at half of Vcc (or 2.5V) for no current flow, and changes by 66mV per Amp for positive current or decreases by 66mV per Amp for negative current.

For example:

-The output voltage at +2 Amps would be :

$$2.5V + (0.066 \times 2) = 2.632V$$

$$2.5V + (0.066 \times 2) = 2.632V$$

-The output voltage at -5 Amps would be :

$$2.5V - (0.066 \times 5) = 2.170V$$

$$2.5V - (0.066 \times 5) = 2.170V$$

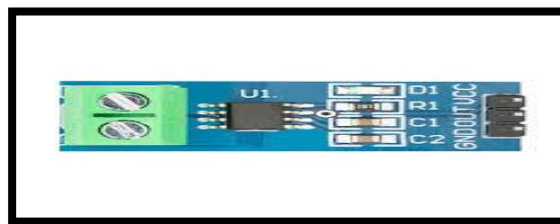


Figure 10:Diagram of ACS 712
mode 30A

II/Specifications:

VCC: 5V DC Nominal

Measurement Range: -30 to +30 A

Voltage at 0A: VCC/2 (Nominally 2.5V DC)

Scale Factor: 66 mV per A

III/Dimensions:

Length: 31.73 mm (1.24")

Width: 13.25 mm (0.52")

Height: 14 mm (0.55")

Weight: 2.72 g (0.096 oz)

IV/Rôle ACS712

The ACS712 module is an economical solution for measuring current in your DIY circuits. The sensor detects the amount of current and provides a voltage output proportional to the current consumed.

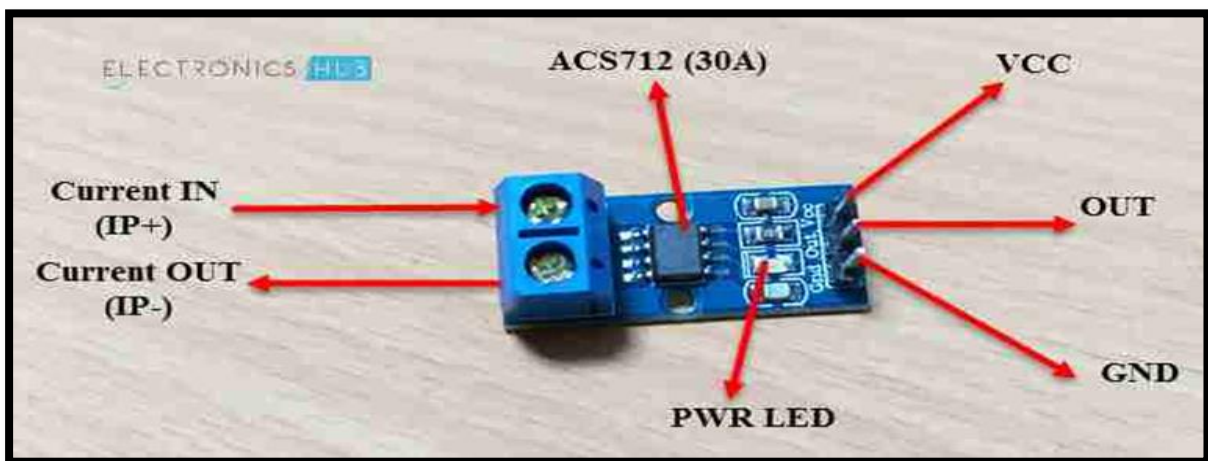


Figure 11:Pin out of ACS712 Current sensor

V/ Method of work:

ACS712 is sensor works with hall effect, this effect depends on producing a voltage exact image of the current to be measured. The appearance of an electric field transverse and a potential difference in a metal or semiconductor carried by an electric current when introduced into a field of magnetic induction perpendicular to the direction of the current, is called the hall effect.

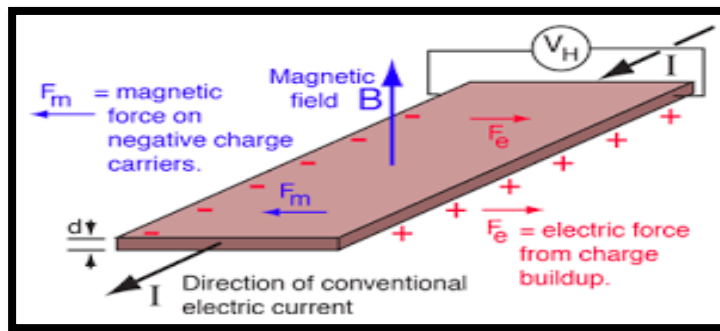


Figure12 :Principle of the hall effect

2.2.3 Arduino Uno

I/Definition:

Arduino Uno is a small control board based on ATmega328 (database). It has 14 digital inputs/outputs (6 of which can be used as PWM inputs), 6 analog inputs, 16 MHz ceramic resonator, USB connection, power connection, ICSP key and reset button. It includes everything needed to support a microcontroller; Simply connect it to your computer with a USB cable or connect it to an AC-DC adapter or battery to get started.[12]

The Uno differs from all previous boards in that it does not use an FTDI USB driver chip. Instead, it has an Atmega16U2 (type Atmega8U2-R2) designed as a USB-to-serial converter

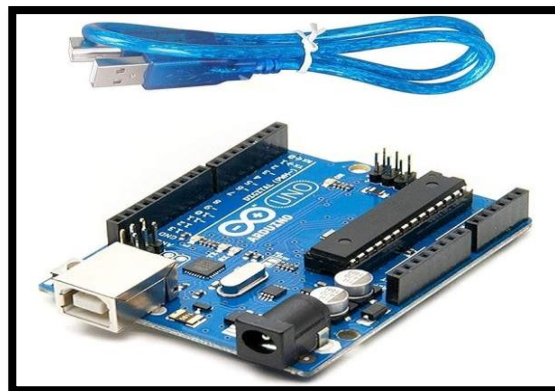


Figure 13:Arduino uno Atmega328

II/Summary

Microcontroller Atmega328

Operating Voltage 5V

Input voltage (Recomended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14(of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40mA

DC Current for 3.3V Pin 50mA

Flash Memory 32Kb(Atmega328) of which 0.56Kb used by bootloader

SRAM 2 Kb (Atmega328)

EEPROM 1 Kb (Atmega328)

Clock Speed 16Mhz

III/Power and Input and Output:

-Power:

Arduino Uno can be powered by USB connection or external power supply. The power source is selected automatically.

External power can come from an AC-DC adapter (not USB) or a battery. The adapter can be connected by inserting a 2.1mm positive value into the card's power connector. The battery wire can be connected to the Gnd and Vin pins of the POWER connector. The board can work with an external power supply between 6 and 20 V. However, if the provides less than 7V, pin5V can provide less than 5V and the board has the power to become negative. If you use more than 12V, the voltage regulator may overheat and damage the wood. Recommended range is(7-12)V. The light source is:

]VIN. Input voltage of the Arduino board when using an external power source (as opposed to 5 volts from a USB connection or other fixed power source). You can supply voltage to this pin, or if you supply voltage from the power supply, you can access this pin.

]5V. This 5V pin outputs the fixed 5V from the motherboard. The card can be powered from a DC power supply (7-12V), a USB connection (5V), or the VIN pin on the card (7-12V). Charging from the 5V or 3.3V pin will bypass the regulator and may damage the board. We don't support it.

] **3V3**. 3.3 volt electricity is produced by the ship's regulator. The maximum current draw is 50mA.

] **GND**. pin on the ground.

-Input and Output:

Any of the Uno's 14 digital pins can be used as input or output using `pinMode()`, `digitalWrite()`, and `digitalRead()`. They work from 5 volts. Each pin can supply or receive a maximum of 40 mA and has an internal (connected) resistance of (20-50) kOhm. In addition, some pins have special functions:

- **Serial: 0 (RX) and 1 (TX)**. It is used to receive (RX) and transmit (TX) TTL serial data. These pins connect to the corresponding pins on the ATmega8U2 USB-TTL chip.

- External Indicators: 2 and 3**. These pins can be configured to trigger a stop or price change at a low, high or low price. For details, see the attach `Interrupt()`

- PWM function: 3, 5, 6, 9, 10 and 11**. Specify an 8-bit PWM output with the `analogWrite()` function.

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

- **LED: 13**. The built-in power supply is connected to digital pin 13. If the pin is high, the LED is on, if the pin is low, it is off. The Uno has 6 analog inputs, labeled A0 to A5, each providing 10-bit resolution (ie 1024 different values).

They are usually measured from low to 5V, although it is possible to change the upper limit of the range using the AREF pin in `analogReference()`. In addition, some pins have special functions:

- **TWI: A4 or SDA pin and A5 or SCL pin**. Supports TWI communication using the `Wire` library.

There are **several** other pins on the board:

- **AREF**. reference voltage for analog input. Used in `analogReference()`.

- **Reset**. this line to the bottom to reset the microcontroller. A shielded reset button is often added to block the board. Also see the separation between Arduino pins and ATmega328 ports. The map of Atmega8, 168 and 328 is the same.

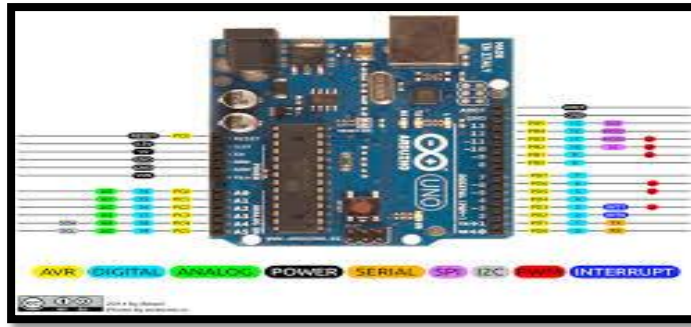


Figure 14:Pin out diagram of Arduino UNO

IV/ Programming and Role of Arduino UNO:

-Programming:

Although Arduino boards are typically programmed using Arduino C, you can also use other programming languages such as Java, Matlab, VB.NET, and more. These languages offer a range of libraries for interacting with Arduino, providing greater flexibility for your projects.

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board).



Figure 15 Programming the Arduino board on the computer

2.3 Software Programming

2.3.1 Arduino IDE and Node MCU

In our project, we will connect arduino uno with node mcu with the help of (AdaFruit MQTT library) for this we will download this library from this Link (https://github.com/adafruit/Adafruit_MQTT_Library)

Now, we open Arduino IDE. We will go to examples -> AdaFruit MQTT library -> mqtt_esp8266

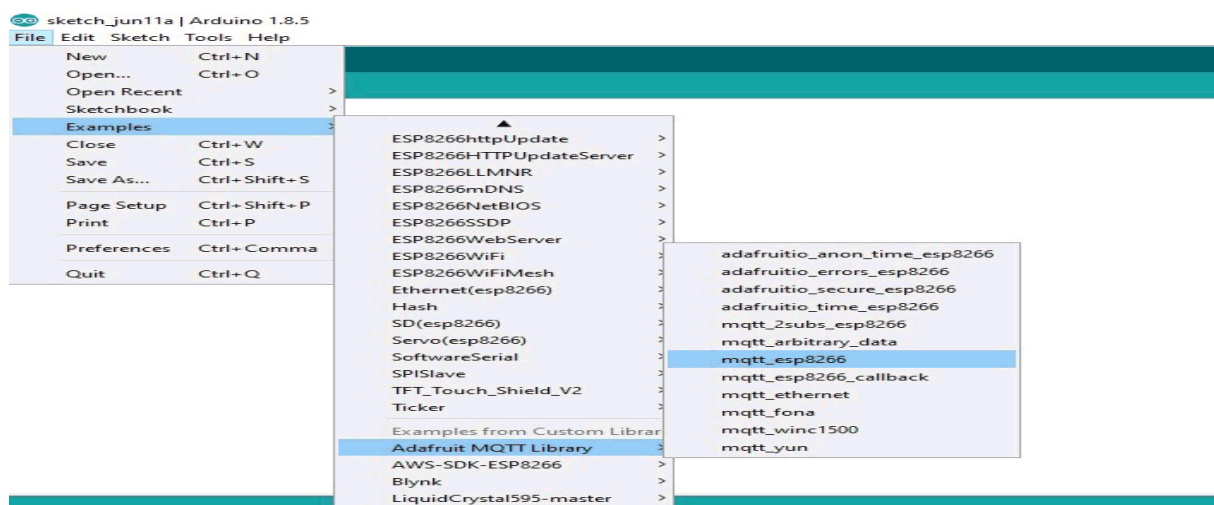


Figure 16 Programming Adafruit program with Arduino platform

2.3.2 Application software

2.3.2.1 MQTT Protocole

MQTT stands for (Message Transport Queuing Telemetry). It is a simple messaging system used when clients need a small code space and connect to unreliable networks or networks with limited resources. It is primarily used for machine-to-machine (M2M) communication or Internet of Things types of connections.

MQTT is making by Dr. Andy Stanford-Clark and Arlen Nipper started in 1999. The original purpose of the communication system was to allow oil and gas monitoring equipment to send their data to remote servers. In many cases, such surveillance devices are used in remote areas where any type of building,

cable or radio would be difficult or impossible to establish. At that time, the only option in such cases was satellite communication, which was expensive and paid according to the amount of data used. With thousands of sensors in the field, the industry needs a form of communication that provides enough reliable data for use at minimum speed.

MQTT became open source under the Organization for Development of Structured Information Standards (OASIS) in 2013. OASIS continues to manage the MQTT standard.

MQTT runs over TCP/IP using a PUSH/SUBSCRIBE topology. There are two types of processes in the MQTT architecture: clients and vendors. An intermediary is a server that a client communicates with. A marketer receives information from customers and sends information to other customers. Customers do not communicate directly with each other, but you communicate with the seller. Each client can be a publisher, a buyer or both.

MQTT is a program driven protocol. No long-term or continuous data transmission. This reduces transmission. The client only publishes when there is a message to send, and the broker sends a message to the client when a new message arrives.

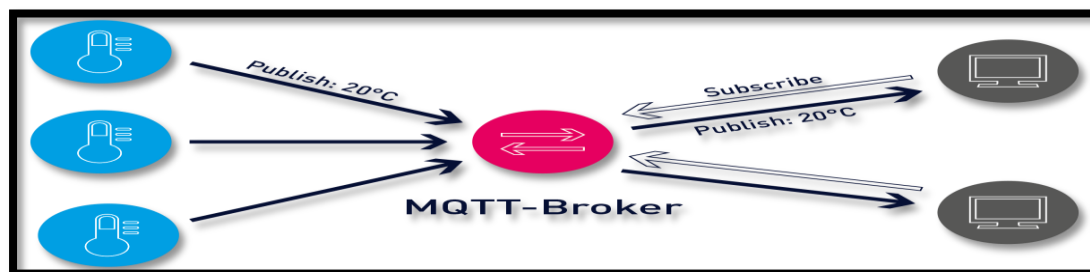


Figure 17:MQTT-Broker Diagram

2.3.2.2 Adafruit IO

I/Defintion and components:

Developed by Adafruit Company, Adafruit IO is a cloud-based service designed for IoT applications. It works as a bridge between devices connected to the cloud, enabling real-time data exchange. At its core, Adafruit IO has two main components: feeds and dashboards.

Feed: A feed is a stream of data generated by an IoT device. Each listing is associated with a unique name and can store various types of data, including sensor readings, status updates, and user input. Food acts as a database, allowing developers to access and manage information collected on their devices.

Dashboards: Dashboards provide a visual interface for monitoring and controlling IoT devices. Developers can create customized dashboards that include widgets such as charts, gauges, and buttons to display real-time data interactively on connected devices. Dashboards serve as a central hub for IoT project management, providing a user-friendly way to visualize and analyze data.

II/Applications:

The flexibility and flexibility of Adafruit IO makes it suitable for many IoT applications in different industries. Some common applications are:

1. Environmental Monitoring:

Adafruit IO can be used to collect and analyze environmental data such as temperature, humidity and air quality in real time. This data can be used for purposes such as weather forecasting, pollution monitoring and agriculture.

2. Home Automation:

Adafruit IO allows homeowners to automate and control a variety of smart devices, including lights, lights, and security cameras. By connecting IoT devices with Adafruit IO, homeowners can create automated systems that monitor the health of their home from anywhere in the world.

3. Industrial IoT (IIoT):

In industrial environments, Adafruit IO can help monitor and optimize production processes, equipment performance, and energy consumption. By collecting and analyzing data from sensors at the factory level, manufacturers can improve operational efficiency, reduce downtime and increase overall productivity.

4. Healthcare Monitoring:

By integrating wearable sensors with Adafruit IO medical devices, healthcare providers can monitor a patient's health status in real time and react to emergencies.

5. Modern agriculture:

In agriculture, Adafruit IO can help farmers schedule irrigation, monitor soil moisture levels, and analyze weather conditions to improve crop production and conserve water resources. . By placing IoT sensors in the field and connecting them to Adafruit IO, farmers can make data-driven decisions to improve farming practices.

III/Create account:

Getting started with Adafruit IO is straightforward, thanks to its extensive documentation and community support.

- 1. Create an Account:** To begin using Adafruit IO, sign up for a free account on the Adafruit IO website.
- 2. Set Up Feeds:** Once logged in, create feeds to represent the data streams from your IoT devices. Assign unique names to your feeds to distinguish between different types of data.
- 3. Configure Devices:** Configure your IoT devices to send data to **Adafruit IO** using supported protocols and APIs. Most popular hardware platforms, such as Arduino and Raspberry Pi, have libraries or modules available for integrating with Adafruit IO.



Figure 18:Adafruit IO logo

2.4 Conclusion

When designing the ESP8266 board, one of the goals of designing is to make it easy for everyone to use, and to make anyone capable to perform various electronic tasks. it's important to know a lot of technology. In this section, we found this card and some sensors to identify the smart electricity meter during the remote data reading, which we consider a useful part of the project.

CHAPTER 3

Practical Realization of (SEMMS)

3.1 Introduction

The objective of this last chapter is to use the technology of communication in order to make the energy meter intelligent.

In the first part, the synoptic diagram of the smart meter energy is proposed. The second part is devoted to the protocol of communication used.

The creation and practical results of the communicating energy meter are presented at the end of this chapter.

3.2 Block Diagram of the (SEMMS)

Our project has the main components of the (SEMMS), and the NodeMCU, which is the microprocessor of the project, and the card required for communication between traditional meters and consumers.

A diagram of the proposed meter is shown below:

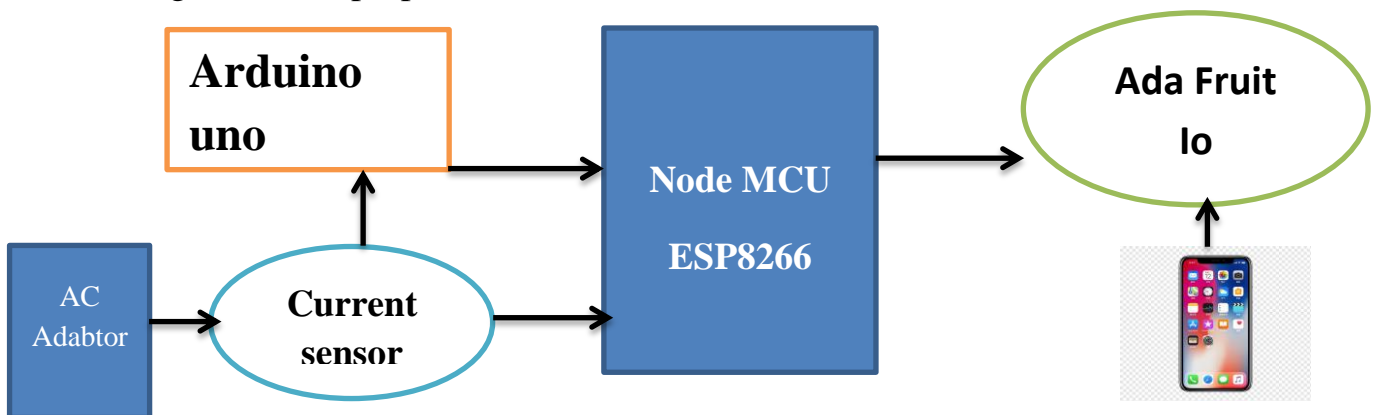


Figure 19 Components diagram of the project

The flowchart below explains the steps in the operation of our project in order to synchronize data with the NodeMCU on one side and the smart phone on the other hand.

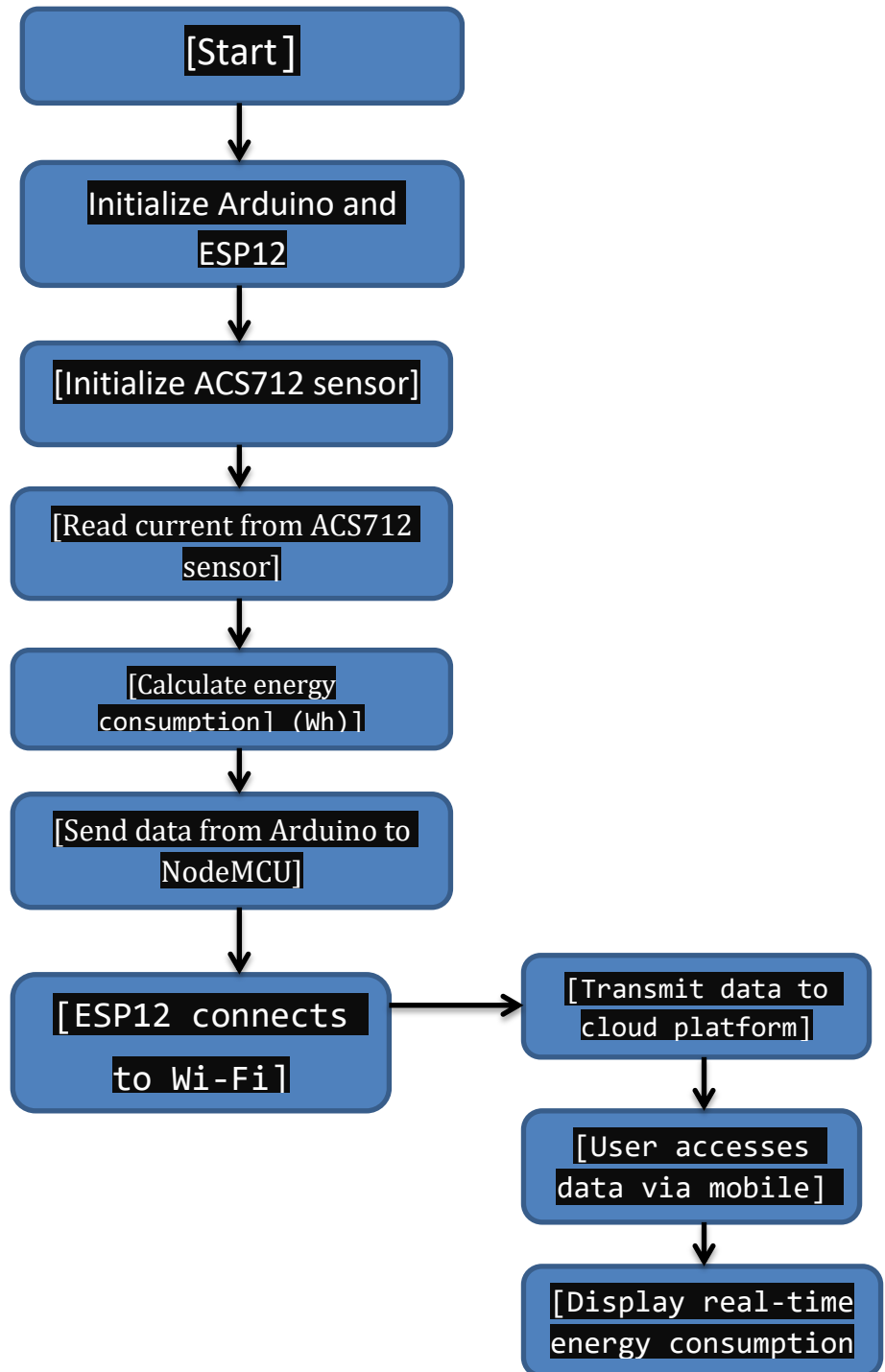


Figure 20 Flowchart of the Project

3.3 Calculate The Energy Consumed in kwh

The ESP32 uses an Emonlib Library which reads data from voltage and current sensors and calculates the following values:

- Effective tension. • Active power.
- The effective current. • Reactive power.
- The power factor. • Apparent power.

And to calculate the energy consumed:

$$E = Pa.t/3600000000$$

E: Energy consumed (KWh).

Pa: Active power (W).

t: Time (ms).

3.4 The Communication Protocol

3.4.1 Setting up an AdaFruit account for communication

First, we will make a feed in **AdaFruit IO**. Feed stores the data sent by IFTTT. To make feed follow these steps:

Step 1:- Login to [Adafruit IO](https://adafruit.io) with your credentials or Sign up if you don't have an account.



Figure 21 Create an Account on the Adafruit Platform

Step 2:-Click on *My account* -> *Dashboard*

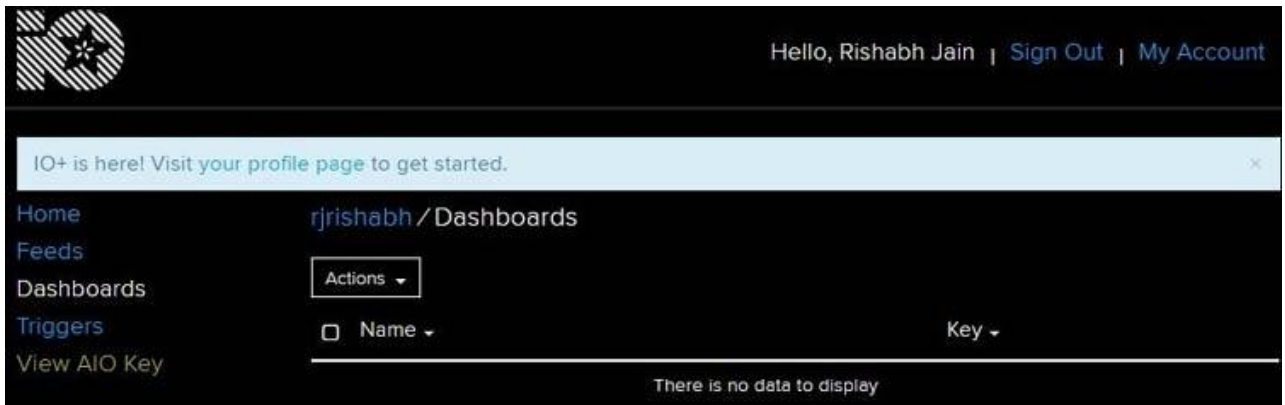


Figure 22 Go to the dashboards section

Step 3:-Click on *Actions* and *Create a New Dashboard*.

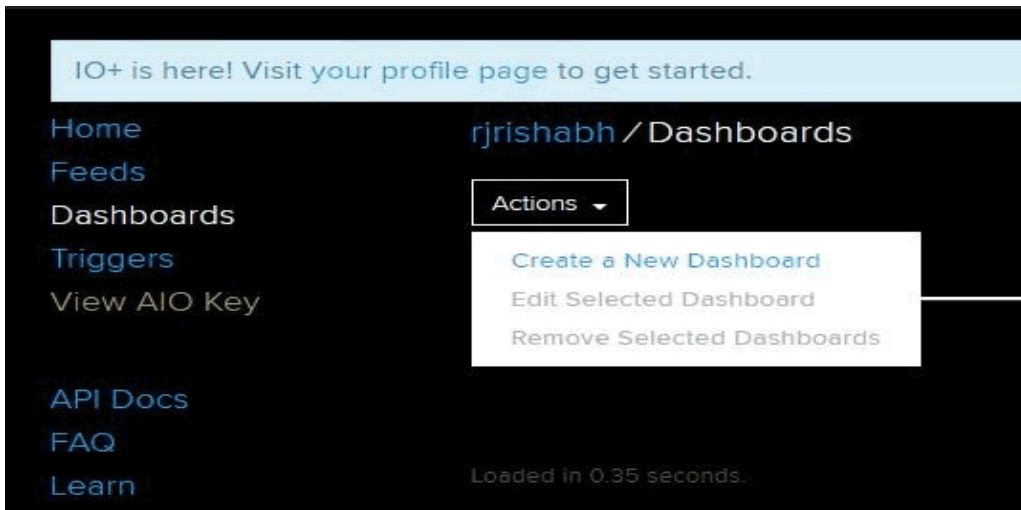


Figure 23 Creating a new dashboard

Step 4: Give name and description to your feed and click on *Create*.

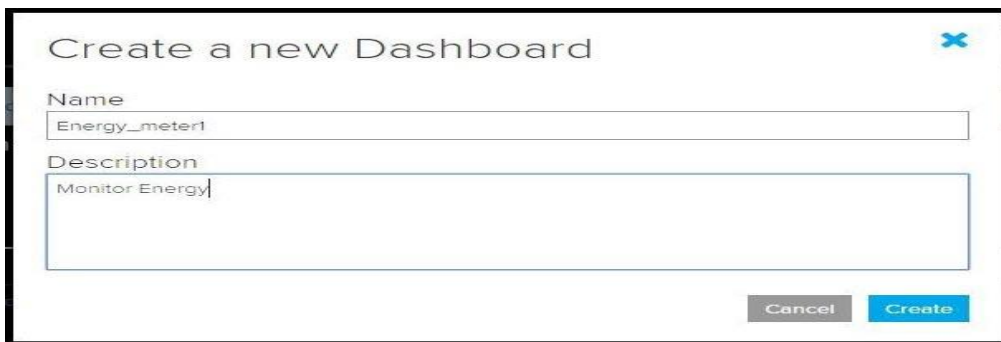


Figure 24 Adding name and description to the dashboard

Step 5: Click on *Key* button and note down the AIO Keys, we will use this key in our code.

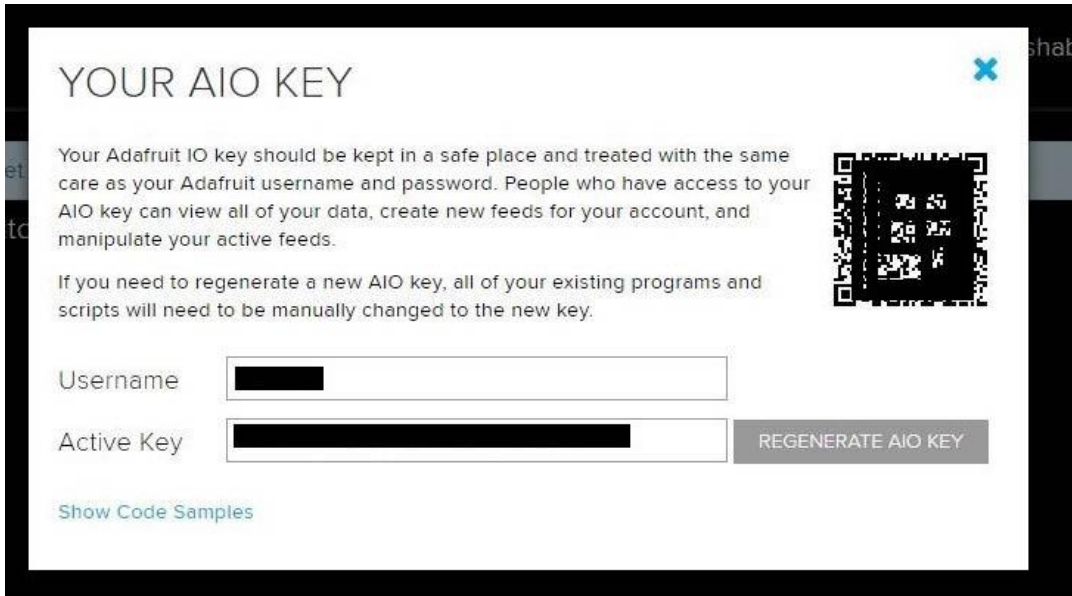


Figure 25 Note the AIO keys of the account

Step 6: Click on '+' button to create a new block and click on *Gauge* to display Energy uses level. You can also use simple text box to display energy.

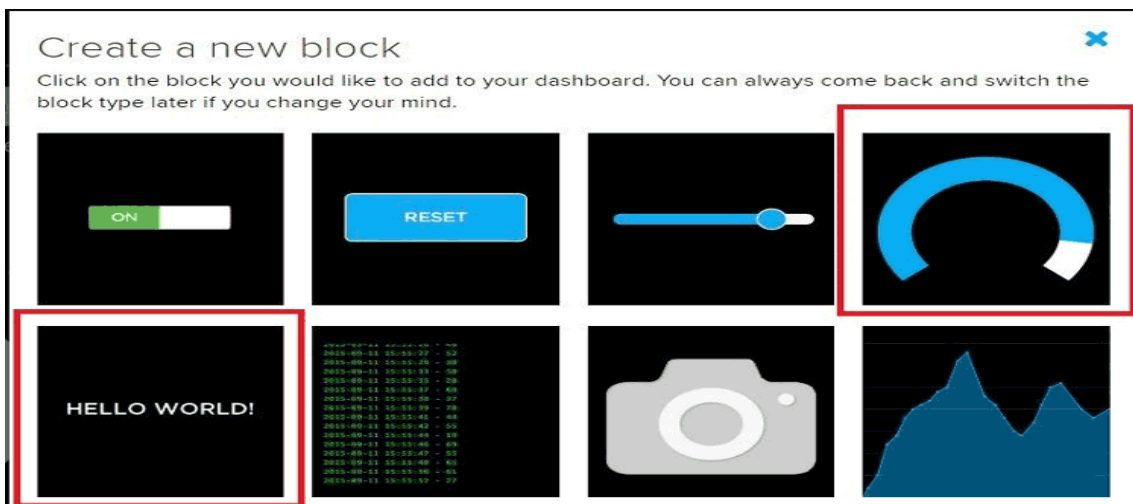


Figure 26 Create a bloc for display the energy

Step 7: Now, Enter Name of Feed and click on *Create*. Then Select the feed and click on Next step.

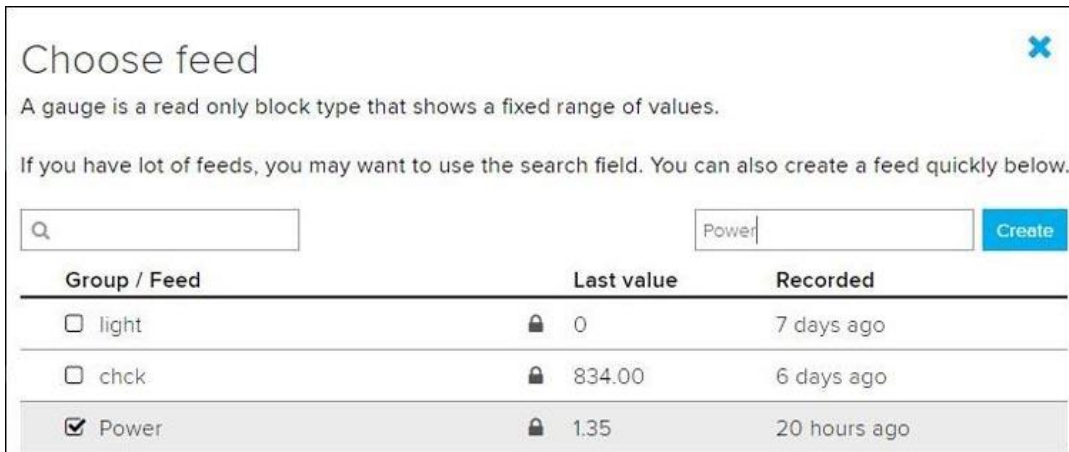


Figure 27 Giving name to the feed

Step 8: In block settings, fill the min. and max values as 0 and 100 respectively or you can modify as you want.

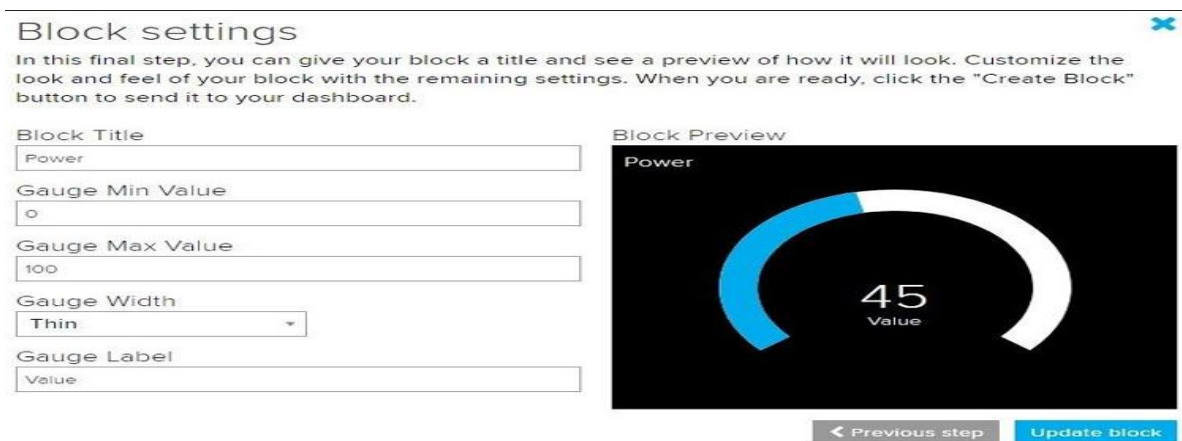


Figure 28 Adjust feed settings

Step 9: Your Power feed is successfully created. Now, we create feed to display Bill by clicking on “+” sign.



Figure 29 Create a new feed to display bill

Now, we have to link AdaFruit IO to SMS/E-mail using IFTTT.

3.4.2 Creating Applet in IFTTT for Triggering SMS/Email for Energy Meter

Step 1: Login to [IFTTT](#) with your credentials.

Step 2: On My Applets, Click on **New Applet**

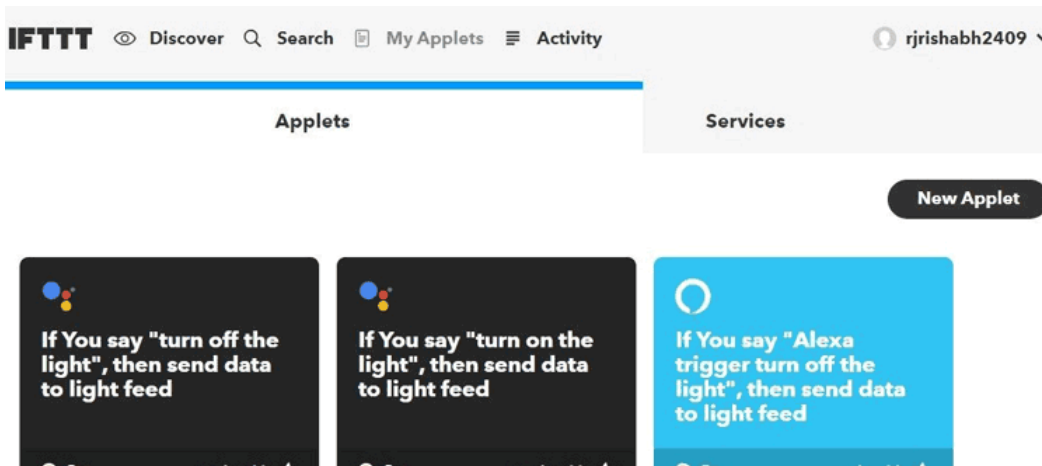


Figure 30 Click on New applet

Step 3: Click on **+this**

Step 4: Search *AdaFruit* and click on it.

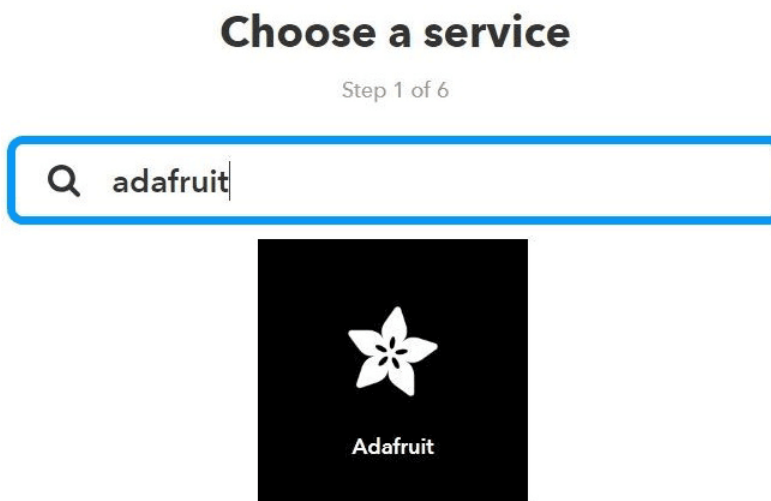


Figure 31 Search AdaFruit

Step 5: Click on *Monitor a feed* on AdaFruit IO.

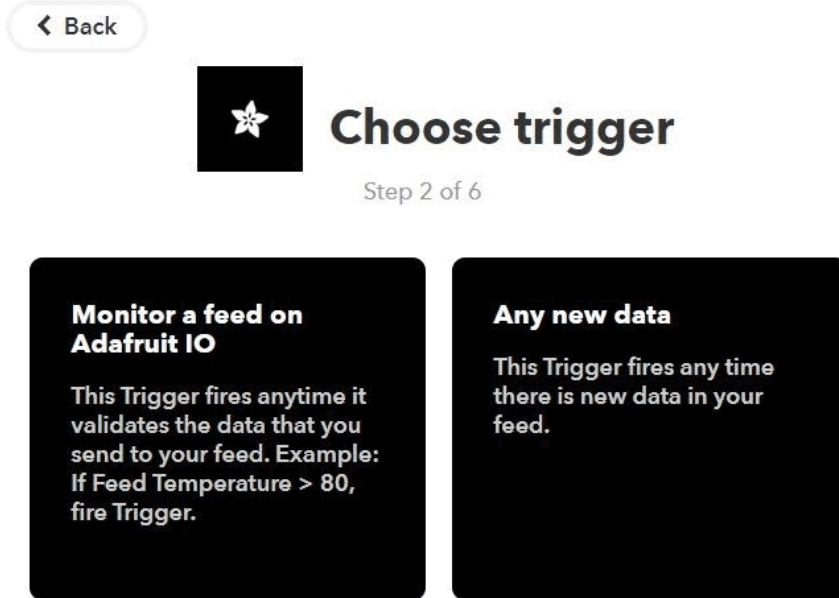


Figure 32 Setting up a feed on adafruit

Step 6: Choose *Feed* as *bill*, *Relationship* as *'equal to'* and the *threshold value* at which you want an E-mail. Click on *Create action*. I have used 4 as my threshold trigger value.



Figure 33 Determing a threshold to the feed

Step 7: Click on *+that*. Search for G-mail and click on it and Login with your g-mail credentials.

Choose action service

Step 3 of 6

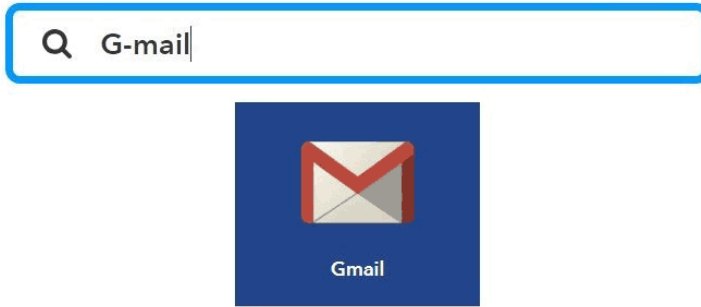


Figure 34 Adding G-mail credentials to our feed

Step 8: Click on **send yourself an email**.

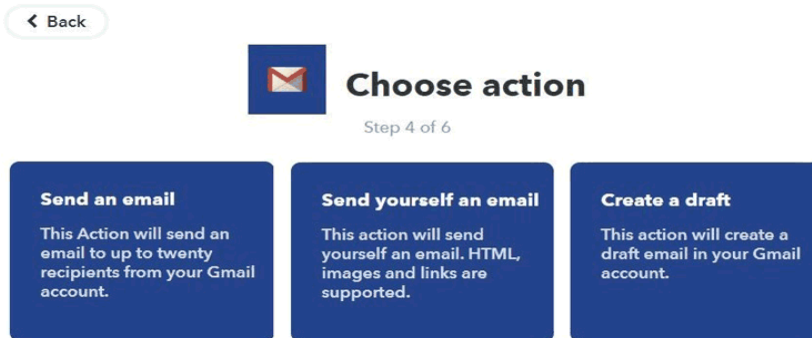


Figure 35 Click on **send yourself an email**

Step 9: Write your subject and body as shown and click to create.

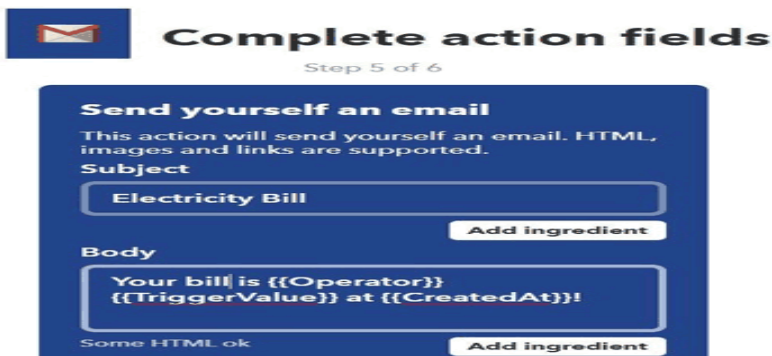


Figure 36 Create an g-mail recipe with there options

Step 10: Your 'recipe' is ready. Review it and click on finish.



Figure 37 Finishing the g-mail recipe

Now, we are done with web integration. Let's move on coding part..

3.4.3 Android App for Monitoring Electricity Consumption

You can use Android App for monitoring the values. For this we download the **MQTT Dashboard android app** from the Play store.

To setup connection with the io.adafruit.com follow these steps:

Step 1: Open the App and click on “+” sign. Fill Client Id anything you want. Server and port remain same as shown in the screenshot. You will get Username and password (Active key) from the AdaFruit IO dashboard as shown below.

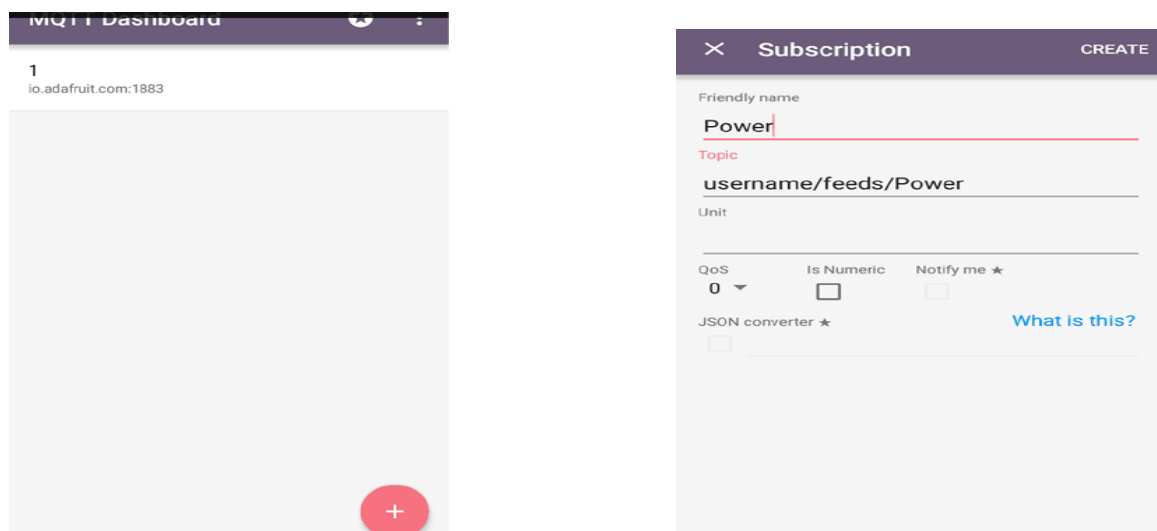


Figure 38 Click on "+" sign on the app and create dashboard with there options

Active Key is your password.

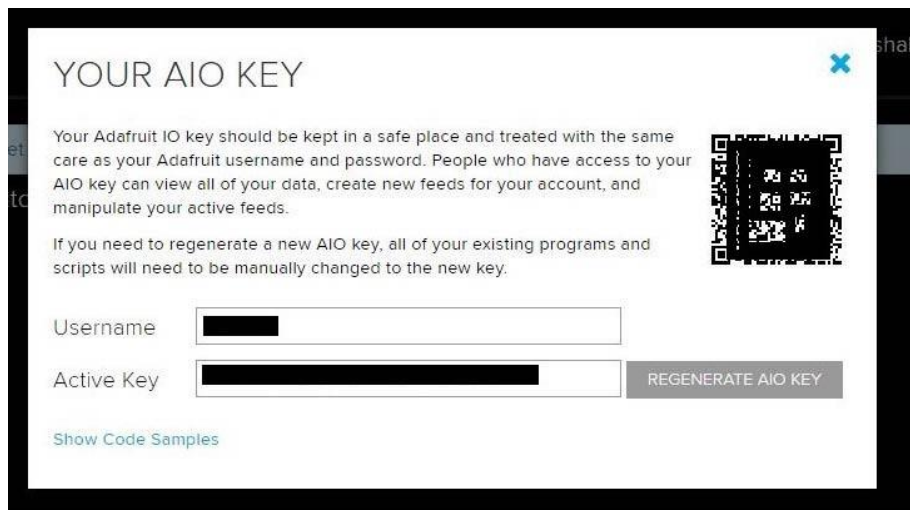


Figure 39 Taking the username and active key of our feed from Ada fruit account

Step 2: Select Electricity Meter and select **Subscribe**. In subscription, give friendly name and topic. Topic format is *'yourusername'/feeds/'feedname'* and click on create.

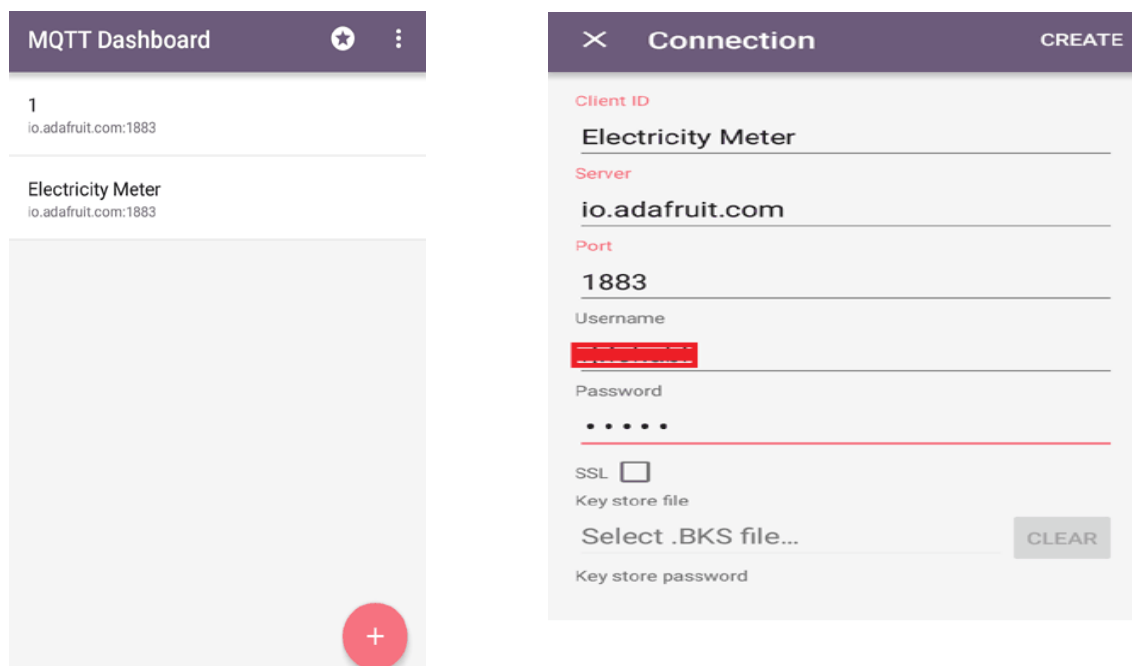


Figure 40 Make a subscription to both elctricity meter and bill feeds

Step 3: In the same way, make subscription for **bill** feed.

Step 4: As your appliances consuming energy, updated values will be displayed under the *Power* and *Bill*.

This is how you can create a **Smart Electricity Energy Meter**, which can be not only monitored from anywhere in the world but also trigger Email when you have high Electricity consumption.

3.5 The Creation and Practical Results of The (SEMMS)

3.5.1 Practical Realization of (SEMMS)

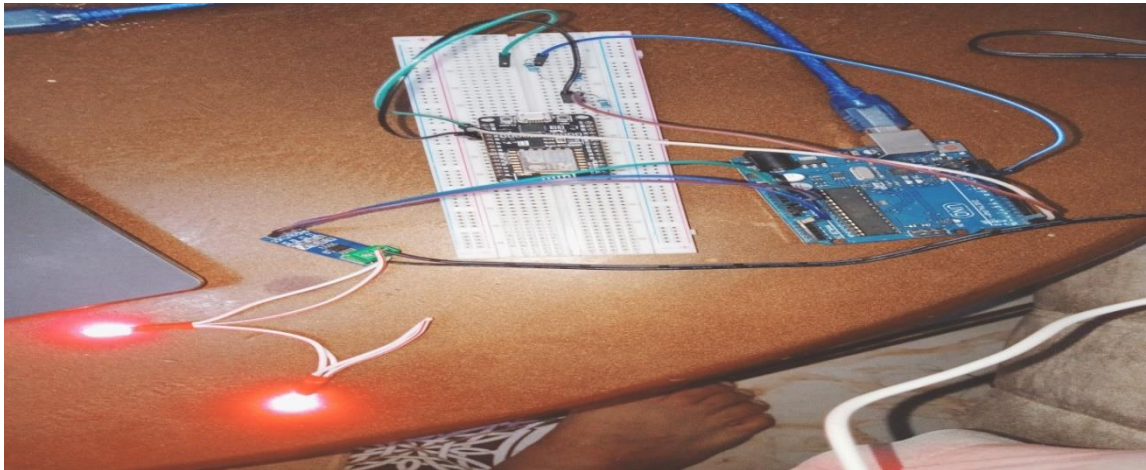


Figure 41: Structure of smart energy meter

3.5.2 Practical Results

Figure 42 represents the data displayed in the smartphone application.

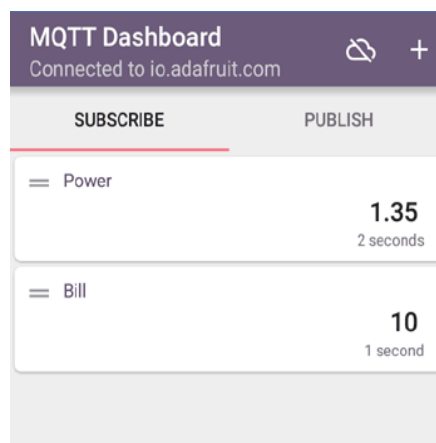


Figure 42: the data displayed in the smartphone application.

The figure43 shows the consumed power by the load while using ACS712 current sensor on arduino ide.

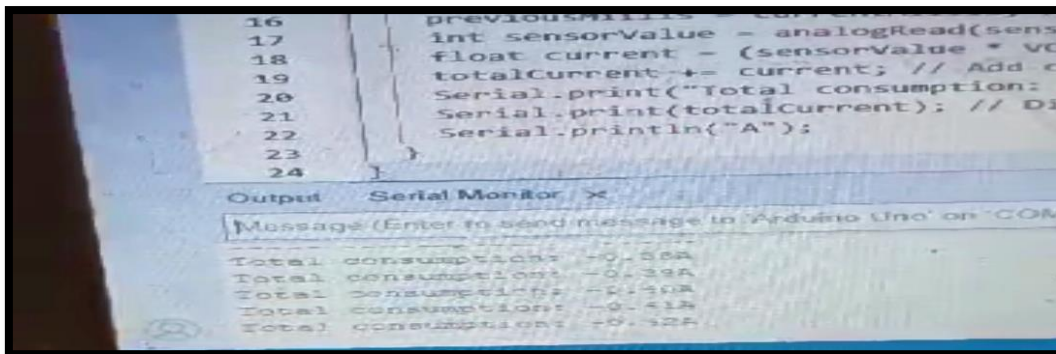


Figure 43:the consumed power by the load

3.6Conclusion

In the last chapter, we presented the construction of the circuit and the communication used to make the energy meter intelligent. By designing "AdaFruit IO" architecture, using "MQTT Dashboard android application", we have implemented an Android mobile application that can integrate data from the NodeMCU board to execute well-defined commands. At the end of the chapter, actual controls are demonstrated and tests confirm the correct operation of (SEMMS).

General Conclusion

In this research, we delved into the SEMMS system and learned about its components in detail. We embodied this system practically using the Arduino program. In the first chapter, we touched on the broad outlines of the SEMMS, which are its definition, features, areas of application, etc. Then, in the second chapter, we separated the components of the project and defined the role. Each of its elements. Finally, in the third chapter, we highlighted the theoretical and practical aspects of the project by presenting the steps for creating the project and testing it realistically, where we obtained good results.

We aspire to expand this project by employing artificial intelligence in order to save the financial cost of the project, as well as facilitate its use and generalize it to include factories and offices of major companies.

References

- [1]. Lee, Shi-Wei, et al. "Design of an automatic meter reading system [electricity metering]." Proceedings of the 1996 IEEE IECON. 22nd International Conference on Industrial Electronics, Control, and Instrumentation. Vol. 1. IEEE, 1996.
- [2]. Popa, M. "Data collecting from smart meters in an Advanced Metering Infrastructure." 2011 15th IEEE International Conference on Intelligent Engineering Systems. IEEE, 2011.
- [3]. Rashdi, Adnan, et al. "Remote energy monitoring, profiling and control through GSM network." Arabian Journal for Science and Engineering 38.11 (2013): 3249-3257.
- [4]. Abdollahi, Ali, Marjan Dehghani, and Negar Zamanzadeh. "SMS-based reconfigurable automatic meter reading system." 2007 IEEE International Conference on Control Applications. IEEE, 2007.
- [5]. Sehgal, Vivek Kumar, et al. "Electronic Energy Meter with instant billing." 2010 Fourth UKSim European Symposium on Computer Modeling and Simulation. IEEE, 2010.
- [6] Anon: "Electricity Tariff Fundamentals", University of Pretoria Electricity and Energy group, 2008. [Online]. Available e-mail: Werner.Badenhorst@up.ac.za Message: get Electrical Tariff Fundamentals paper
- [7]IR Remote Controlled Home Automation using PIC Microcontroller Published in May 11, 2017 by Mr.Saddam
- [8]article about (Why you should say no to getting a smart meter)/puplihed by Mr.tom haynes in 8march2023/ <https://www.telegraph.co.uk/>

[9]MBARKI Imen. (Juillet 2018). Etude et réalisation d'un compteur d'énergie intelligent. Université de Gabès. Tunisie.

[10]<https://www.paessler.com/it-explained/mqtt>

[11]"IoT Based Electricity Energy Meter using ESP12 and Arduino",Published July 10, 2018 by Rishabh Jain

[12] farnell.com,1682209

[13] Khalid Al-Otaibi, Ahmed Ananzi ,Abdullah Abaoud ,Mohammed Dhameri" Smart Energy Metering and Control System" ,pupliched in 2019 Prince Mohammad BIN Fahad university.

Annex A:

Code For Arduino

```
#include <ACS712.h>

// Constants and variables
const float VOLTAGE = 230.0; // Voltage in volts
const int BUFFER_SIZE = 10;
char watt[BUFFER_SIZE]; // Increased buffer size for safety

ACS712 sensor(A0, 5, 1023, 66);

unsigned long last_time = 0;
unsigned long current_time = 0;
float Wh = 0;

void setup() {
  Serial.begin(115200);

  // The library does not have a calibrate function
  // sensor.calibrate();

  last_time = millis(); // Initialize last_time to the current time
}

void loop() {
  float I = sensor.getCurrentAC(); // Assuming getCurrentAC() is correct

  float P = VOLTAGE * I; // Power calculation (P = V * I)
```

```
current_time = millis();

Wh += P * ((current_time - last_time) / 3600000.0); // Energy calculation

last_time = current_time; // Update last_time after the computation

dtostrf(Wh, 4, 2, watt); // Convert float to string

Serial.print("Energy (Wh): ");

Serial.println(watt);

delay(10000); // Delay for 10 seconds

}
```

Code For NodeMCU

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define WLAN_SSID      "a*****"
#define WLAN_PASS      "*****"

char watt[5];

#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    "rjrishabh"
#define AIO_KEY          "*****"

WiFiClient client;

int bill_amount = 0;

unsigned int energyTariff = 8.0;

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

Adafruit_MQTT_Publish Power = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/Power");
```

```
Adafruit_MQTT_Publish bill = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/bill");

void MQTT_connect();

void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println(F("Adafruit MQTT demo"));
  // Connect to WiFi access point.
  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);
  WiFi.begin(WLAN_SSID, WLAN_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
}

void loop() {
  // Ensure the connection to the MQTT server is alive (this will make the first
  // connection and automatically reconnect when disconnected).  See the MQTT_connect
  // function definition further below.
  MQTT_connect();

  int i=0;
  float watt1;
  if(Serial.available() > 0 ){
    delay(100); //allows all serial sent to be received together
```

```
while(Serial.available() && i<5) {
    watt[i++] = Serial.read();
}
watt[i++]='\0';
}
watt1 = atof(watt);
bill_amount = watt1 * (energyTariff/1000);    // 1unit = 1kWh
Serial.print(F("\nSending Power val "));
Serial.println(watt1);
Serial.print("...");
if (! Power.publish(watt1)) {
    Serial.println(F("Failed"));
} else {
    Serial.println(F("OK!"));
}
if (! bill.publish(bill_amount)) {
    Serial.println(F("Failed"));
} else {
    Serial.println(F("OK!"));
}
if (bill_amount==4){
for (int i =0; i<=2; i++)
{
    bill.publish(bill_amount);
delay(5000);
}
bill_amount =6;
}
```

```
delay(5000);
}
// Function to connect and reconnect as necessary to the MQTT server.
// Should be called in the loop function and it will take care if connecting.
void MQTT_connect() {
    int8_t ret;
    // Stop if already connected.
    if (mqtt.connected()) {
        return;
    }
    Serial.print("Connecting to MQTT... ");
    uint8_t retries = 3;
    while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
        Serial.println(mqtt.connectErrorString(ret));
        Serial.println("Retrying MQTT connection in 5 seconds...");
        mqtt.disconnect();
        delay(5000); // wait 5 seconds
        retries--;
        if (retries == 0) {
            // basically die and wait for WDT to reset me
            while (1);
        }
    }
    Serial.println("MQTT Connected!");
}
```