

IOT BASED ELECTRICAL DEVICE AND SURVEILLANCE

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

by

A. RANJITH (37130343)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

SCHOOL OF ELECTRICAL AND ELECTRONICS

**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 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC 1 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF ELECTRICAL AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **RANJITH.A
(37130343)** who carried out the project entitled “**IOT BASED ELECTRICAL
DEVICE SURVEILLANCE** ” under our supervision from October 2020 to April
2021.

Internal Guide

Dr. E. ANNADEVI, M.E., Ph.D.

Assistant professor, Department of ECE

Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **A.RANJITH (37130343)** hereby declare that the Project Report entitled “**IOT BASED ELECTRICAL DEVICE SURVEILLANCE**” done by us under the guidance of **Dr. E. ANNADEVI, M.E., Ph.D. Assistant Professor, Department of ECE** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

I convey my thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D. Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, M.E., Ph.D. Head of the Department, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to our Project Guide **Dr. E. ANNADEVI, M.E., Ph.D. Assistant Professor, Department of Electronics and Communication Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of the project work.

I wish to express my thanks to all teaching and Non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

We express our gratitude to our parents for their constant encouragement and support for the completion of the project.

ABSTRACT

In this work, a new procedure is followed based on MICROCONTROLLER Node MCU to detect and monitor the power consumption, faults, and loss. A data will be sent automatically to the utility central server through IOT module. Here Automatic meter reading can be Proposed in the System along with Cloud data analysis and Money Management with Energy meter report and Billing Displayed in webpage.

TABLE OF CONTENTS

CHAPTER	DESCRIPTION	PAGE NO.
1	Introduction	1
1.1	Electronic	1
1.2	Existed system	1
1.3	Proposed System	2
1.4	Working of Block Diagram	3
1.5	Signal Condition Circuit	3
1.6	Drawbacks of the Regular energy meter	4
1.7	Modules	4
1.7.1	Potential transformer	5
1.7.2	Node MCU	5
1.7.3	cloud	6
2	Literature Review	7
2.1	Literature Survey	7
2.2	Literature summary	8
2.3	Problem Definition	9
3	Aim and Scope of Present Investigation	10
3.1	Project Methodologies	10
3.2	Area of project Work and Application	10
3.2.1	Overview of Alert Message System	12
3.3	Software and Hardware Requirement	12
3.3.1	Python overview	13
4	Experimental Material and Method	16
4.1	Hardware requirements	16
4.2	Hardware Explanation	16
4.3.	ESP-12E Module	21

4.4	Power Requirement	22
4.5	Peripherals and I/O	23
4.6	On-board switches & LED Indicator	24
4.7	Power Pins	27
4.8	ESP8266 Development Platform	29
4.9	Installing the ESP8266 core to windows	30
4.9.1	Application	35
4.9.2	Current transformer	36
4.10	Handheld current Transformer	44
4.11	Potential Transformer	45
4.11.1	Phasor diagram of a potential transformer	51
4.12	Software required	52
4.12.1	Software explanation	52
4.13	python	57
4.13.1	An introduction to python	57
5	Result	65
6	Conclusion	66
	Reference	67

LIST OF FIGURES

FIGURE (FIG)	NAME OF THE FIG	PAGE NO.
1.1	Block diagram	3
1.2	Current transformer	4
1.3	Potential transformer	5
1.4	Pin diagram	6
1.5	Cloud data	6
4.1	Pin diagram of ESP8266 Module	18
4.2	GPIO Pins	19
4.3	ESP-12E chip	21
4.4	Power Requirement	22
4.5	Multiplexed GPIO	23
4.6	Node MCU	24
4.7	Serial communication	25
4.8	Pin diagram of node MCU	26
4.9	Arduino	29
5	Real time monitoring	65
5.1	Energy meter monitoring voltage	66
5.2	Monitoring power	67
5.3	Monitoring unit	67
5.4	Monitoring current	68

LIST OF TABLES

TABLE	NAME OF THE TABLE	PAGE NO.
4.1	PIN NAMES ON NODE MCU	19

LIST OF ABBREVIATION

IOT- INTERNET OF THINGS

GPIO- GENERAL PURPOSE INPUT OUTPUT

IC- INTEGRATED CIRCUIT

MQTT- MQ TELEMETRY TRANSPORT

NODE MCU-NODE MICROCONTROLLER

CHAPTER 1

1.INTRODUCTION

1.1 ELECTRONIC

Electronic metering technology greatly reduces the man power and time and also made easy for remote area people for paying the bill. It also reduces the non-payment of the bill and avoids the mistake due to manual calculation. Power theft is one of the greatest problems that our country is facing and with the help of this AME power theft can be greatly reduced.

Consider a complicated power system network, it is very usual to get faults in every branches of the network. In this system all the loads will be monitored and examined through online using IoT.

In case of losses also can be determined by this system by comparing the past values or normal values with present value. If there is any variation occurred can be intimated by this system.

1.2 EXISTED SYSTEM

Conventional System Energy meters display kilowatt-hour by continuously measuring the instantaneous voltage (volts) and current (amperes) to give energy used in joules. The primary type of electricity meter is the electromechanical induction meter and an electronic meter. In electromechanical induction meter, the total number of rotations of the aluminium disc is directly proportional to the power consumed.

Electronic meters show the power consumed, power factor, the reactive power used digitally displayed on LCD or LED display, and also able to send the energy consumed readings to remote places through some communication network. It is a single way communication. In addition to measuring energy used, electronic meters

can also record other parameters of the load and supply such as instantaneous and maximum rate of usage demand, voltage, power factor and reactive power used etc. “Traditional, the electricity meter are installed on consumer’s premises and the consumption information is called by meter-reader on their fortnightly or monthly

visits to the premises

In case of loss cannot be detected by a conventional system. It only measures instantaneous values.

Fault can be occurred in a power system at any time, it can be due to overvoltage, short circuit etc. consider a network got affected by fault in a particular branch. In case of our normal system this cannot be identified easy

1.3. PROPOSED SYSTEM

A smart energy meter works on communication directly with wireless data protocol so there will be precise reading. Smart energy meter can operate in divergent ways with IOT Module. The proposed system consists of digital energy meter, a NODE MCU (microcontroller). The data from energy meter by the help of current and potential transformer will be transmitted to cloud by IOT modules. The stored data can recognize by authorized persons. Terms like power loss, over power usage, instantaneous power, total energy usage, faulty loads, can be recognized by this system. Every branch in the power system can be monitored in every instant of time

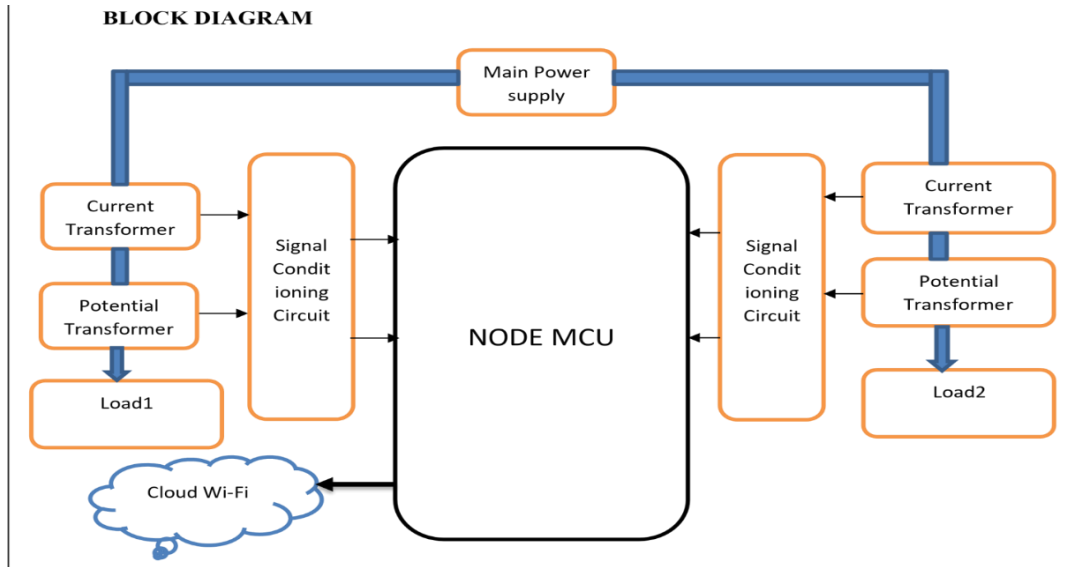


Fig : 1.1 : block diagram

1.4 WORKING OF BLOCK DIAGRAM

There is two branches of loads are there in the block diagram. For those branches separate voltage and current transformer pairs are added each will continuously monitor the instantaneous power. Whenever loss occurred the value of dataset will be compared with present values and the fault will be determined, fault in a particular Branch can also be determined so that it will be easy for regulating those faults.

If any fault occurred it can also be determined by regular check-up, also faulty load can be fixed by this system

1.5 SIGNAL CONDITIONING CIRCUIT

Voltage transformer is a normal transformer which convert high voltage to particular voltage comfortable for us. Here the microcontroller work in 5 voltage or less than that so by using potential transformer we have to convert (0-230) volt to (0-5) voltage, for that conversion we are using signal conditioning circuit

In case of current transformer also we are converting current value to a proportional voltage value comfortable to microcontroller.

1.6 DRAWBACKS OF THE REGULAR ENERGY METER

- 1) Highly depends on meter reader.
- 2) Human error cannot be avoided for the manual meter reading.
- 3) Always there is no cross checking or recheck of human readers for energy utilization.
- 4) High chance of stealing and bribery always high to misuse it especially during events.
- 5) Possibility to change the reading when taking photos of energy meter by using software tools.
- 6) More number of meter reading employees is extra expenses to the company for hiring them and their expense on traveling too expensive one.
- 7) Wherever energy meter installed inside the house, which may lead to non-checking of reading due to lock.
- 8) The consumer is not receiving updates of his regular usage of energy.
- 9) The consumer may not receive his energy bill as per regular interval of the due date

1.7 MODULES

Current Transformer (CT):

CT is used to measure or monitor the current in transmission lines and to isolate the metering equipment and relay connected to secondary side. The current level from energy meter by the help of current transformer will be transmitted to cloud by IoT module

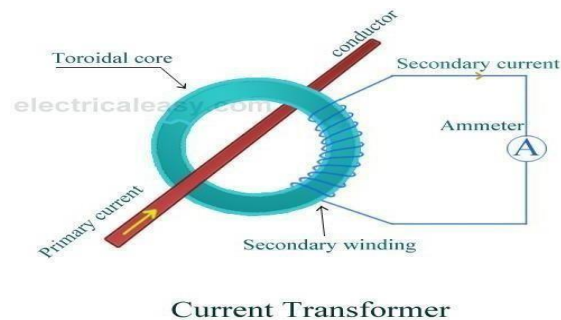


Fig : 1.2 : current Transformer

1.7.1. Voltage Transformer or Potential Transformer (PT):

PT is used to measure or monitor the voltage on transmission lines and to isolate the metering equipment from the lines. The voltage level from energy meter by the help of current transformer will be transmitted to cloud by IoT module.



Fig 1.3 : Potential Transformer

1.7.2 Node MCU:

This module is the controller with inbuilt Wi-Fi. With the help of this modules the data obtained to the transformer. These transformers are used to measure the current and voltage level and stored into the cloud

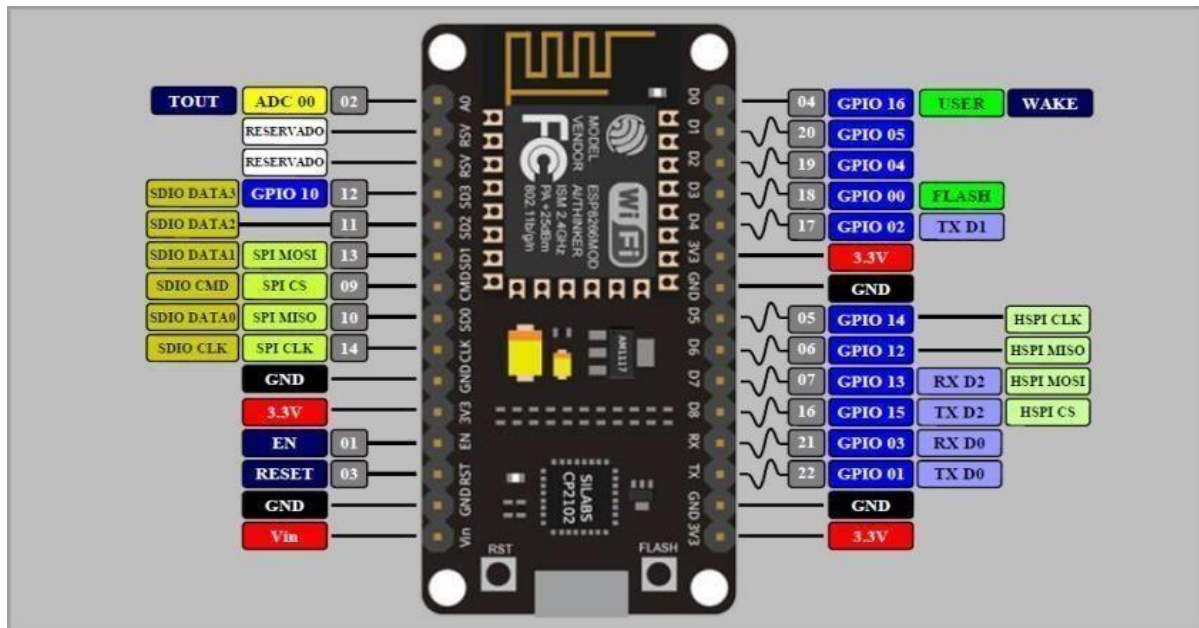


Fig 1.4 : Pin Diagram

1.7.3 Cloud

The entire data collections from transformer are stored in the cloud. These stored data are highlysecured. The stored data can recognize by authorized persons.

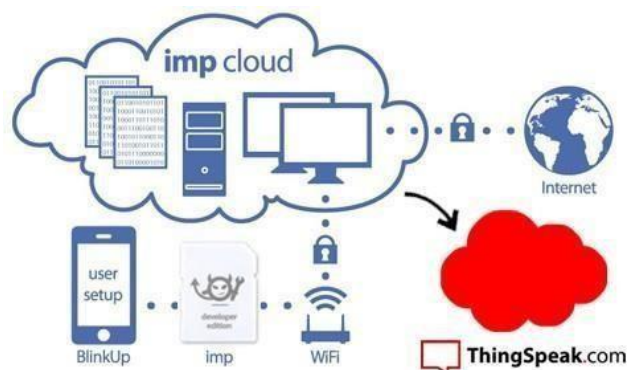


Fig : 1.5 : Cloud Data

CHAPTER 2

2.LITERATURE REVIEW

This chapter discusses we about the literature survey and summary. It also gives a detailed idea on the existing.

2.1. LITERATURE SURVEY

1. SMS-Based Reconfiguration Automatic Meter Reading System –2007-Ali Abdollahi, marajanDehghani, and Negar Zamanzadeh—In recent years many engineers and companies have been working on Automatic Meter Reading (AMR), and different media, such as radio, telephone lines, and power lines have been used for this purpose. Nowadays, GSM network, with its vast coverage in most countries and also its competitive ever-growing market, is becoming popular as the main medium for the machine applications, and AMR is not an exception.

2. Smart energy meter—brindas1 ,vishalkumarsah2,jaladi,harish3,akshayu4,vishal,deo matho5,swetha umapathy6 --- (2008) In the currently working system, electricity meter reading for electricity usage and building is done by human workers from home to home and buildings. The purpose of this project is to develop a smart electricity meter using GSM. This can reduce human error and helps to retrieve the real time meter value via GSM and sent it to customers mobile phone through GSM. This also allow the electricity board to modify the variable package price in specific duration. The administrator can analyse the customers power consumption data and generate the report from the data online. 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---GSM,Power Consupction billing ,smart electricity meter.

3. Data Collecting From smart meter in an advanced metering infrastructure—2011—M.popa-
- This paper describes an AMI based system. Smart meters are connected to a gateway through power lines and the gateway communicates with a central computer through GSM. The power line communication is implemented with an industrial bus, LonWork. The paper focuses the data collection solution.—plc,zigbee,gsm

4. Design of an automatic meter reading system[electricity metering]—2012—shi-weilee;cheng shong wu; meng shi chiou; kou-tan wu.-- A microprocessor-based automatic meter-- reading system is implemented, which provides a cost-effective, reliable, and interference free data transfer between remote meter reading units and the utility control center. The meter reading and management processes are free from human involvement. Based on the existing telephone networks, it is very flexible for the utility companies to access, service and maintain this meter reading system.—AutomaticMeterReading,Utility public Telephone.Network

2.2 LITERATURE SUMMARY

The Survey has been done for the IOT based electrical surveillance and automatic meter reading. Upon the analysing many papers , it can be concluded that using IOT surveillance we can produce or we can stop the power theft and monitor in the work space and it could be easy and convenient and it can be used to billing conserve our household energy and we send consumed energy via mobile sms.

2.3. PROBLEM DEFINITION

1. Highly depends on meter reader
2. Human error cannot be avoided for the manual meter reading.
3. Always there is no cross checking or recheck of human reader for energy utilization
4. possible to change the reading when taking photos of energy meter by using software tools.
more number of meter reading employees is extra expense to the company.

CHAPTER 3

3. AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1. PROPOSED METHODOLOGIES

Our proposed project integrates all the requirements to compensate the specified drawbacks

A smart energy meter works on communication directly with wireless data protocol so there will be precise reading.

Every branch in the power system can be monitored in every instant of time

3.2. AREA OF PROJECT WORK AND APPLICATION

Consider a complicated power system network, it is very usual to get faults in every branches if the network. In this system all the loads will be monitored and examined through online using IoT. In case of losses also can be determined by this system by comparing the past values or normal values with present value. If there is any variation occurred can be intimated by this system. Electronic metering technology greatly reduces the man power and time and also made easy for remote area people for paying the bill. It also reduces the non-payment of the bill and avoids the mistake due to manual calculation. Power theft is one of the greatest problems that our country is facing and with the help of this AME power theft can be greatly reduced. Electronic meters show the power consumed, power factor, the reactive power used digitally displayed on LCD or LED display, and also able to send the energy consumed readings

to remote places through some communication network. It is a single way communication Fault can be occurred in a power system at any time, it can be due to overvoltage, short circuit etc. consider a network got affected by fault in a particular branch. In case of our normal system this cannot be identified easy

3.2 ALERT MESSAGE SYSTEM

There are many alert message techniques available. Each technique has its own drawbacks, So necessary knowledge is required to utilize a technique balancing all the needs.



Fig : 3.1 : Alert Message

3.2.1 OVERVIEW OF ALERT MESSAGE SYSTEM

The message alert system is the process of manipulating pl data in order to give users the message according how they have consumed their units on specific terms.

3.3 SOFTWARE AND HARDWARE REQUIREMENTS

HARDWARE:

Potential transformer , CT sensor , NODE MCU.

SOFTWARE:

Python

3.3.1 PYTHON OVERVIEW

Flexible and powerful, Python was originally developed in the late 1980s at the National Research Institute for Mathematics and Computer Science by Guido van Rossum as a successor to the ABC language. Since its introduction, Python has grown in popularity thanks to what is seen as a clear and expressive syntax developed with a focus on ensuring that code is readable. Python is a high-level language. This means that Python code is written in largely recognizable English, providing the Pi with commands in a manner that is quick to learn and easy to follow. This is in marked contrast to low-level languages, like assembler, which are closer to how the computer “thinks” but almost impossible for a human to follow without experience. The high-level nature and clear syntax of Python make it a valuable tool for anyone who wants to learn to program. It is also the language that is recommended by the Raspberry Pi Foundation for those looking to progress from the simple Scratch. It’s good practice to start all Python programs with a line known as a shebang, which gets its name from the # and ! characters at the beginning of the line. This

line tells the operating system where it should look for the Python files. Although this is not entirely necessary for programs that will be run from within IDLE or will call Python explicitly at the terminal, it is required for programs that are run directly by calling the program's filename. To ensure the program runs regardless of where the Python executable is installed

This line tells the operating system to look at the \$PATH environment variable—which is where Linux stores the location of files that can be executed as programs—for the location of Python, which should work on any Linux distribution used on the Pi. The \$PATH variable contains a list of directories where executable files are stored, and is used to find programs when you type their name at the console or in a terminal window. To achieve the goal of printing out a message, you should use Python's print command. As its name suggests, this command prints text to an output device—by default, to the console or terminal window from which the program is being executed

Enter the following line in your new project:

```
print "Hello, World!"
```

The final program should look like this:#!/usr/bin/env python

```
print
```

```
"Hello, World!"
```

These were the following simple program to understand the workflow of the python

3.3.2 PYTHON AND NETWORKING

So far, you have learned how Python can be used to create standalone programs, but the language can also be used to create programs that communicate with the outside world over a computer's network connection. This next example, written by Tom Hudson, offers a brief glimpse of these possibilities with a tool for monitoring the users connected to an Internet Relay Chat (IRC) channel. As usual, create a new project in IDLE or a text editor and enter the shebang line along with a comment describing the purpose of the program:

```
#!/usr/bin/env python
```

There are some constants needed for this program to operate. Constants are like variables in that they can have values assigned to them—but unlike variables, the value in a constant shouldn't change. To help differentiate a constant from a variable, it's good practice to use all-capital letters for their names—that way it's easy to see at glance whether a particular section of the code is using a constant or a variable. Type the following two lines into the program:

```
RPL_NAMREPLY = '353'
```

```
RPL_ENDOFNAMES = '366'
```

These are IRC status codes, provided by the server to indicate when particular operations have completed. These are used by the program to know when it has received the required list of names from the IRC server.

```
irc = {  
    'host':  
        'chat.freenode.net'  
    , 'port': 6667,  
    'channel':  
        '#raspiuserguide',
```

```
'namesinterval' : 5 }
```

however: until the 366 response, which signals the end of the member names, is received, the list is incomplete. That is why the last line—`names += names_list.split('')`—is appending the newly-received names to the existing list, rather than blanking it out entirely: each time that section of the code runs, the program is only likely to have received a subsection of the entire member list. To tell Python what to do when the full list has been received, enter the following lines:

```
if response_code == RPL_ENDOFNAMES:
    # Display the names
    print '\nUsers in %(channel)s:'
    % irc forname in names:
        print name names
    = []
```

This tells Python that when the 366 response has been received, it should print out the now-complete list of names to the standard output before blanking the names list again. This last line—`names = []`—is important: without it, each time the loop runs it will add users' names to the list even though they already exist from an earlier run. Finally, finish the program by entering the following lines:

```
time.sleep(irc['namesinterval'])
s.send('NAMES
%(channel)s\r\n' % irc)
```

This tells Python to wait the names interval number of seconds before sending another request for user names and beginning the loop again. Be careful to set names interval to a reasonable value—if the IRC server receives too many requests in too short a space of time, it may forcibly disconnect you for flooding.

CHAPTER 4

MATERIAL AND METHODS USED

4.1 HARDWARE REQUIRED

- 1) NODE MCU
- 2) Current transformer
- 3) Potential transformer
- 4) Required load

4.2 HARDWARE EXPLANATION:

INTRODUCTION

General-purpose input/output (GPIO) is a pin on an IC (Integrated Circuit). It can be either input pin or output pin, whose behaviour can be controlled at the run time.

Where to use ESP8266-01

The ESP8266 is a very user friendly and low cost device to provide internet connectivity to your projects. The module can work both as a Access point (can create hotspot) and as a station (can connect to Wi-Fi), hence it can easily fetch data and upload it to the internet making Internet of Things as easy as possible. It can also fetch data from internet using API's hence your project could access any information that is available in the internet, thus making it smarter. Another exciting feature of this module is that it can be programmed using the ArduinoIDE which makes it a lot more user friendly. However this version of the module has only 2 GPIO pins (you can hack it to use upto 4) so you have to use it along with another microcontroller like Arduino, else you can look onto the more standalone ESP-12 or ESP-32 versions. So if you are looking for a module

to get started with IOT or to provide internet connectivity to your project then this module is the right choice for you.

How to use the ESP8266 Module

There are so many methods and IDEs available to with ESP modules, but the most commonly used on is the Arduino IDE. So let us discuss only about that further below. The ESP8266 module works with 3.3V only, anything more than 3.7V would kill the module hence be cautions with your circuits. The best way to program an ESP-01 is by using the FTDI board that supports 3.3V programming. If you don't have one it is recommended to buy one or for time being you can also use an Arduino board. One commonly problem that every one faces with ESP-01 is the powering up problem. The module is a bit power hungry while programming and hence you can power it with a 3.3V pin on Arduino or just use a potential divider.

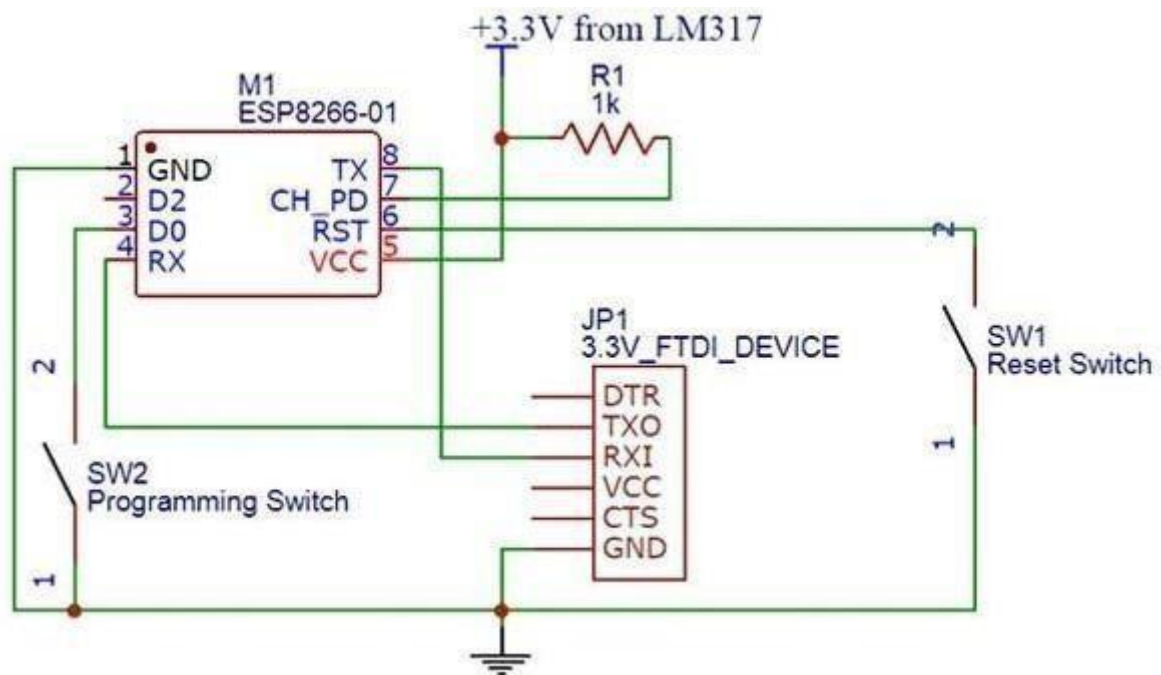


FIG : 4.1 : Pin Diagram

The switch SW2(Programming Switch) should be held pressed to hold theGPIO-0 pin to ground. the code is released the switch can be released.

GPIO PINS:

NodeMCU Development kit provides access to these GPIOs of ESP8266. The only thing to take care is that NodeMCU Dev kit pins are numbered differently than internal GPIO notations of ESP8266 as shown in below figure and table. For example, the D0 pin on the NodeMCU Dev kit is mapped to the internal GPIO pin 16 of ESP8266.

The Internet of Things (IoT) has been a trending field in the world of technology. It has changed the way we work. Physical objects and the digital world are connected now more than ever. Keeping this in mind, Expressive Systems (A Shanghai-based Semiconductor Company) has released an adorable, bite-sized Wi-Fi enabled microcontroller – ESP8266, it can monitor and control things from anywhere in the world.

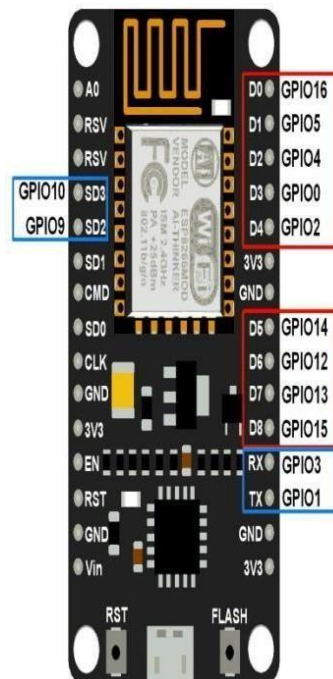


Fig : 4.2 : pin diagram

Pin Names on NodeMCU Development Kit	ESP8266 Internal GPIO Pin number
D0	GPIO16
D1	GPIO5
D2	GPIO4
D3	GPIO0
D4	GPIO2
D5	GPIO14
D6	GPIO12
D7	GPIO13
D8	GPIO15
D9/RX	GPIO3

The GPIO's shown in blue box (1, 3, 9, 10) are mostly not used for GPIO purpose on Dev Kit. ESP8266 is a system on a chip (SoC) design with components like the processor chip. The processor has around 16 GPIO lines, some of which are used internally to interface with other components of the SoC, like flash memory.

Since several lines are used internally within the ESP8266 SoC, we have about 11 GPIO pins remaining for GPIO purpose.

Now again 2 pins out of 11 are generally reserved for RX and TX in order to communicate with a host PC from which compiled object code is downloaded.

Hence finally, this leaves just 9 general purpose I/O pins i.e. D0 to D8.

As shown in above figure of NodeMCU Dev Kit. We can see RX, TX, SD2, SD3 pins are not mostly used as GPIOs since they are used for other internal process. But we can try with SD3 (D12) pin which mostly like to respond for GPIO/PWM/interrupt like functions.

Note that D0/GPIO16 pin can be only used as GPIO read/write, no special functions are supported on it.

4.3 ESP-12E Module

The development board equips the ESP-12E module containing ESP8266 chip having Tensilica Xtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency and supports RTOS.

ESP-12E Chip

- Tensilica Xtensa® 32-bit LX106
- 80 to 160 MHz Clock Freq.
- 128kB internal RAM
- 4MB external flash
- 802.11b/g/n Wi-Fi transceiver

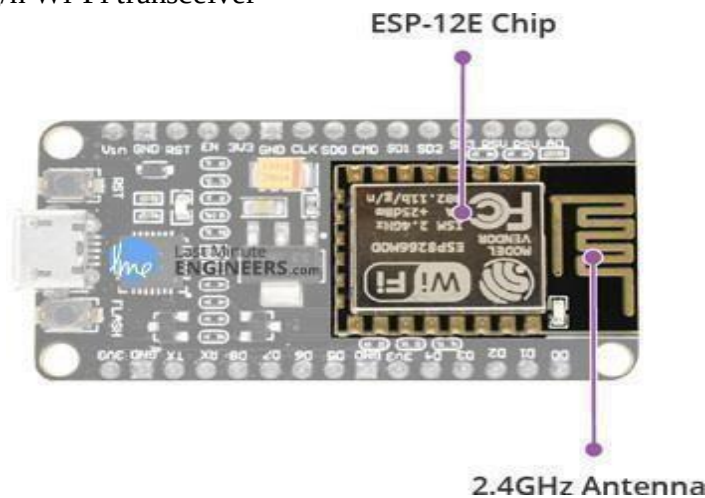


Fig : 4.3 : ESP-12E CHIP

There's also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays.

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 NodeMCU even more versatile

4.4 Power Requirement

As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as 80mA during RF transmissions. The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.

- Operating Voltage: 2.5V to 3.6V
- On-board 3.3V 600mA regulator
- 80mA Operating Current
- 20 μ A during Sleep Mode

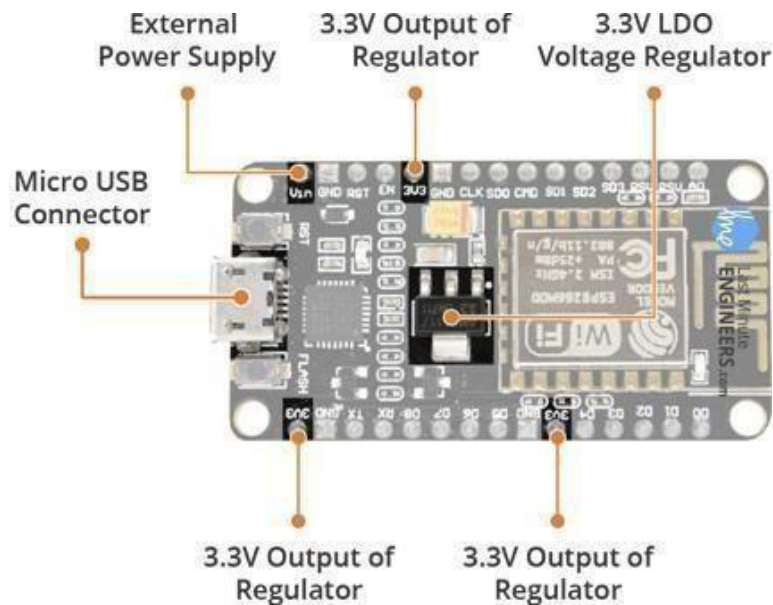


Fig 4.4 : pin diagram power requirement

Power to the ESP8266 NodeMCU is supplied via the on-board MicroB USB connector. Alternatively, if you have a regulated 5V voltage source, the VIN pin can be used to directly supply the ESP8266 and its peripherals

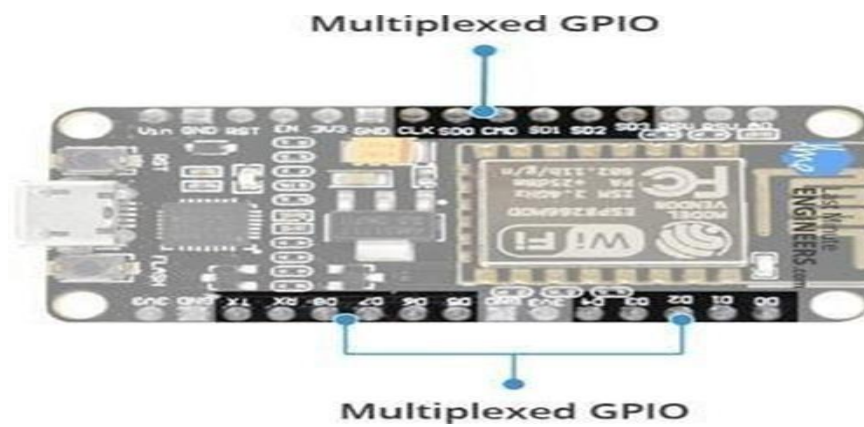


FIG : 4.5 : pin diagram GPIO

Warning

The ESP8266 requires a 3.3V power supply and 3.3V logic levels for communication. The GPIO pins are not 5V-tolerant! If you want to interface the board with 5V (or higher) components, you'll need to do some level shifting.

4.5 Peripherals and I/O

The ESP8266 NodeMCU has total 17 GPIO pins broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:

- ADC channel – A 10-bit ADC channel.
- UART interface – UART interface is used to load code serially.
- PWM outputs – PWM pins for dimming LEDs or controlling motors.
- SPI, I2C & I2S interface – SPI and I2C interface to hook up all sorts of sensors and peripherals.
- I2S interface – I2S interface if you want to add sound to your project.

Multiplexed I/Os

- 1 ADC channels
- 2 UART interfaces
- 4 PWM outputs
- SPI, I2C & I2S interface

The ESP8266's pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as PWM/UART/SPI.

4.6 On-board switches & LED indicator

The ESP8266 NodeMCU features two buttons. One marked as RST located on the topleft corner is the Reset button, used of course to reset the ESP8266 chip. The other FLASH button on the bottom left corner is the download button used while upgrading firmware. **Switches & Indicators**

- RST – Reset the ESP8266 chip
- FLASH – Download new programs
- Blue LED – User Programmable

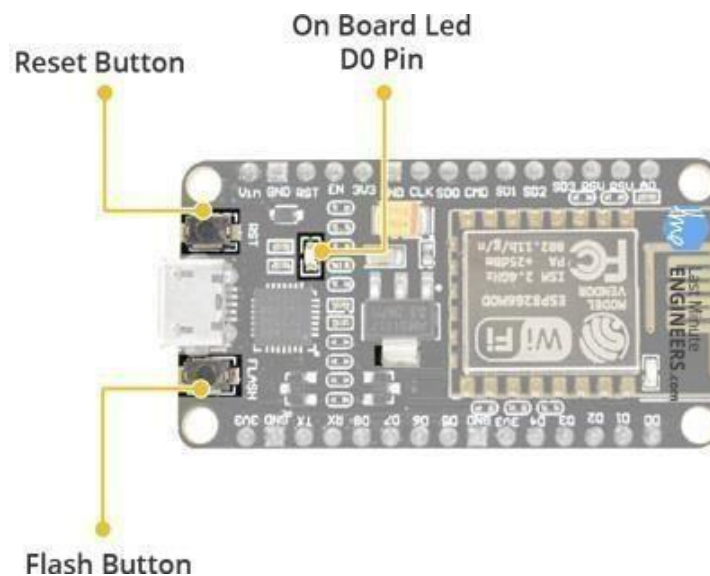


FIG : 4.6 : Pin Diagram ESP8266 NODE MCU

The board also has a LED indicator which is user programmable and is connected to the D0 pin of the

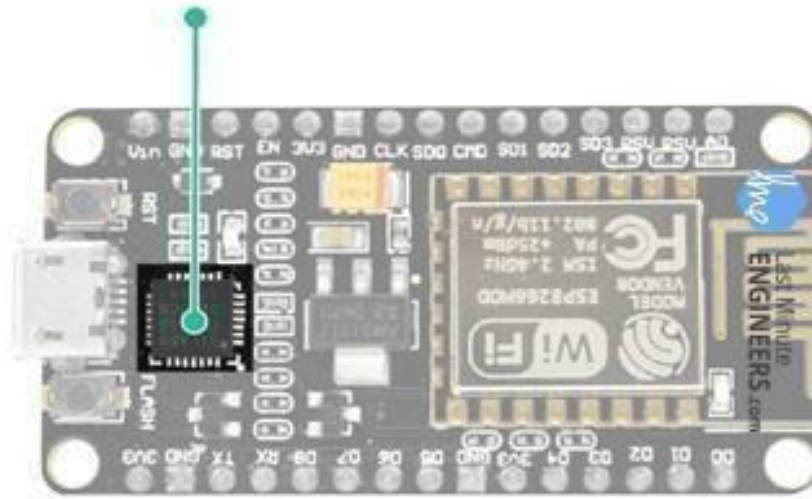
Board.

Serial communication

The board includes CP2102 USB-to-UART Bridge Controller from Silicon

Labs, which converts USB signal to serial and allows your computer to program and communicate with the ESP8266 chip.

USB To TTL Converter CP2102



- CP2102 USB-to-UART converter
- 4.5 Mbps communication speed
- Flow Control support

FIG : 4.7 : Pin Diagram SERIAL COMMUNICATION

If you have an older version of CP2102 driver installed on your PC, we recommend upgrading now.

ESP8266 Node MCU pinout

The **ESP8266** Node MCU has 30 pins that interface it to the outside world. The connections are follows:

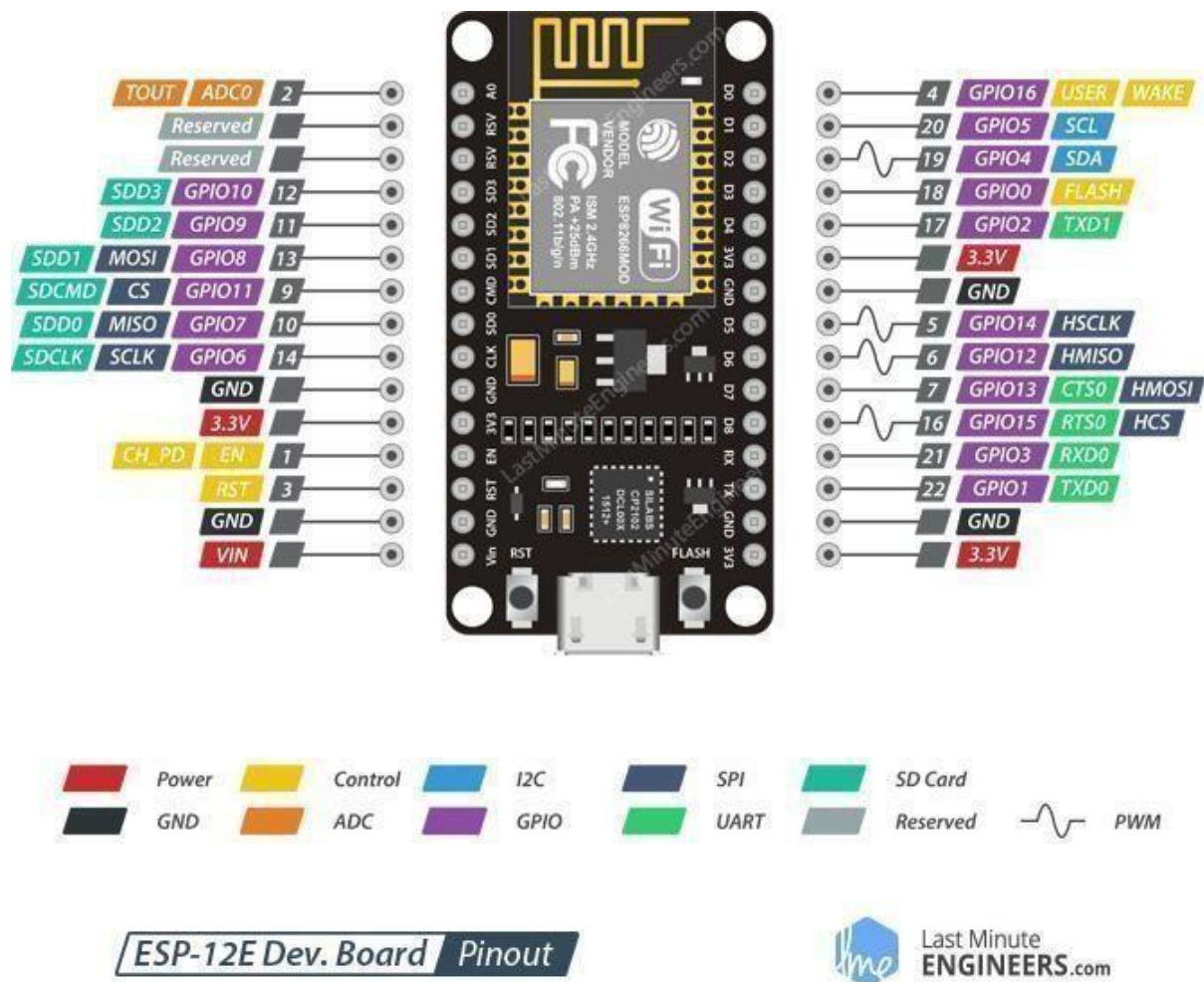


FIG : 4.8 : Pin Diagram

For the sake of simplicity, we will make groups of pins with similar functionalities

4.7 Power Pins

There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin can be used to directly supply the ESP8266 and its peripherals, if you have a regulated 5V

voltage source. The 3.3V pins are the output of an on-board voltage regulator. These pins can be used to supply power to external components.

GND

GND is a ground pin of ESP8266 NodeMCU development board.

I2C Pins

I2C Pins are used to hook up all sorts of I2C sensors and peripherals in your project. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

GPIO Pins

ESP8266 NodeMCU has 17 GPIO pins which can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pulldown, or set to high impedance. When configured as an input, it can also be set to edge trigger or level-trigger to generate CPU interrupts.

ADC Channel

The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC viz. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART Pins

ESP8266 NodeMCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It supports flow control. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

SPI Pins

ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

- 4 timing modes of the SPI format transfer
- Up to 80 MHz and the divided clocks of 80 MHz
- Up to 64-Byte FIFO

SDIO Pins

ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins

The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μ s to 10000 μ s, i.e., between 100 Hz and 1 kHz.

Control Pins

Control Pins are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

- EN pin – The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
- RST pin – RST pin is used to reset the ESP8266 chip.
- WAKE pin – Wake pin is used to wake the chip from deep-sleep.

4.8 ESP8266 Development Platforms

Now, let's move on to the interesting stuff!

There are a variety of development platforms that can be equipped to program the ESP8266. You can go with Espruino – JavaScript SDK and firmware closely emulating Node.js, or use Mongoose OS – An operating system for IoT devices (recommended platform by Espressif Systems and Google Cloud IoT) or use a software development kit (SDK) provided by Espressif or one of the platforms listed on Wikipedia.

Fortunately, the amazing ESP8266 community took the IDE selection a step further by creating an Arduino add-on. If you're just getting started programming the ESP8266, this is the environment we recommend beginning with, and the one we'll document in this tutorial.

This ESP8266 add-on for Arduino is based on the amazing work by Ivan Grokhotkov and the rest of the ESP8266 community. Check out the ESP8266 Arduino GitHub repository for more information.

4.9 Installing the ESP8266 Core on Windows OS

Let's proceed with installing ESP8266 Arduino core.

The first thing is having latest Arduino IDE (Arduino 1.6.4 or higher) installed on your PC. If don't have it, we recommend upgrading now.

Latest Arduino IDE

To begin, we'll need to update the board manager with a custom URL. Open up Arduino IDE and go to File > Preferences. Then, copy below URL into the Additional Board Manager URLs text box situated on the bottom of the window:
http://arduino.esp8266.com/stable/package_esp8266com_index.json

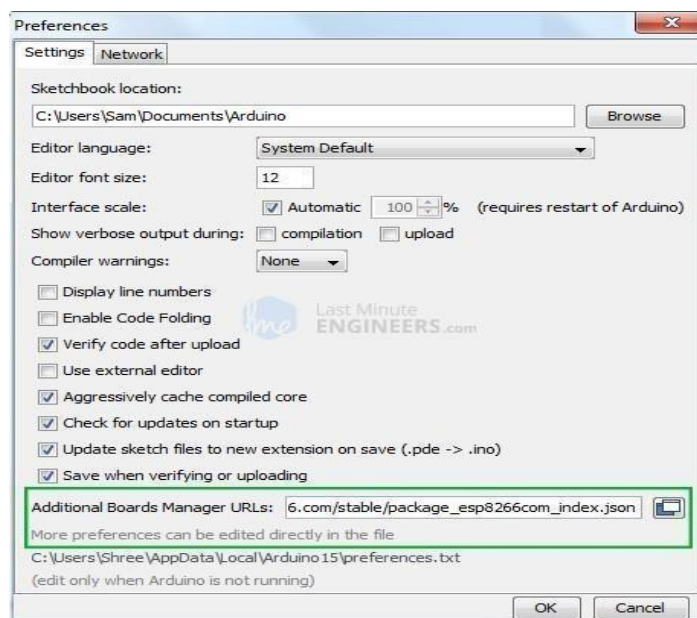


FIG : 4.9 : ARDUINO

Hit OK. Then navigate to the Board Manager by going to Tools > Boards > Boards Manager. There should be a couple new entries in addition to the standard Arduino boards. Filter your search by typing esp8266. Click on that entry and select Install.

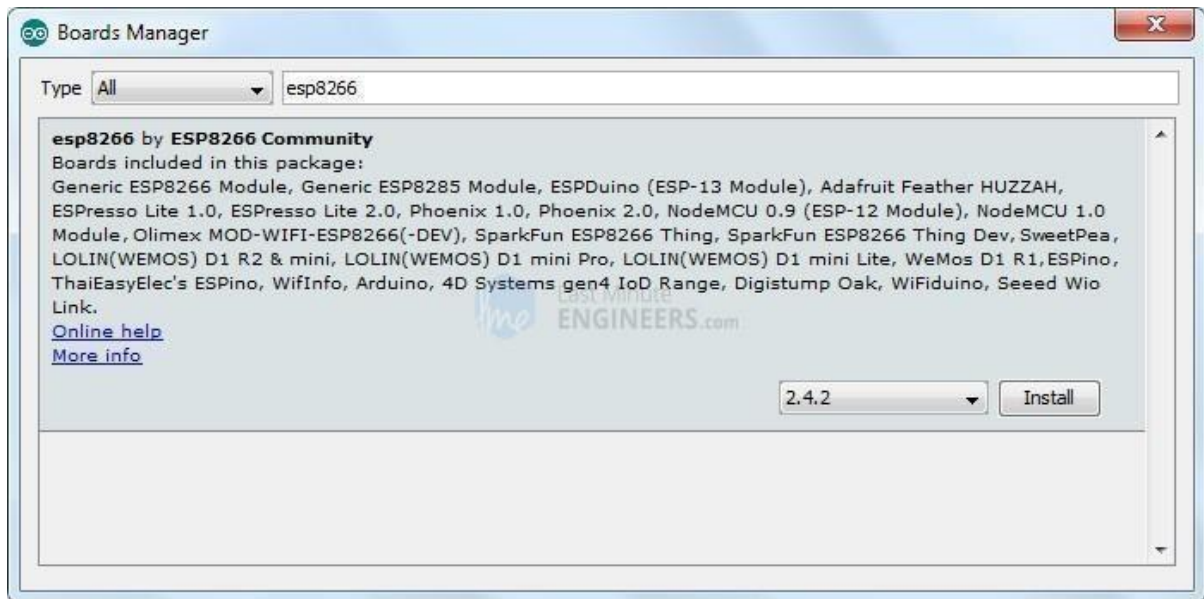


Fig : 4.10 : Download Window

The board definitions and tools for the ESP8266 include a whole new set of gcc, g++, and other reasonably large, compiled binaries, so it may take a few minutes to download and install (the archived file is ~110MB). Once the installation has completed, a small INSTALLED text will appear next to the entry. You can now close the Board Manager.

Arduino Example: Blink

To make sure ESP8266 Arduino core and the NodeMCU are properly set up, we'll upload the simplest sketch of all – The Blink!

We will use the on-board LED for this test. As mentioned earlier in this tutorial, D0 pin of the board is connected to on-board Blue LED & is user programmable. Perfect!

Before we get to uploading sketch & playing with LED, we need to make sure that the board is selected properly in Arduino IDE. Open Arduino IDE and select NodeMCU 0.9 (ESP-12Module) option under your Arduino IDE > Tools > Board menu.

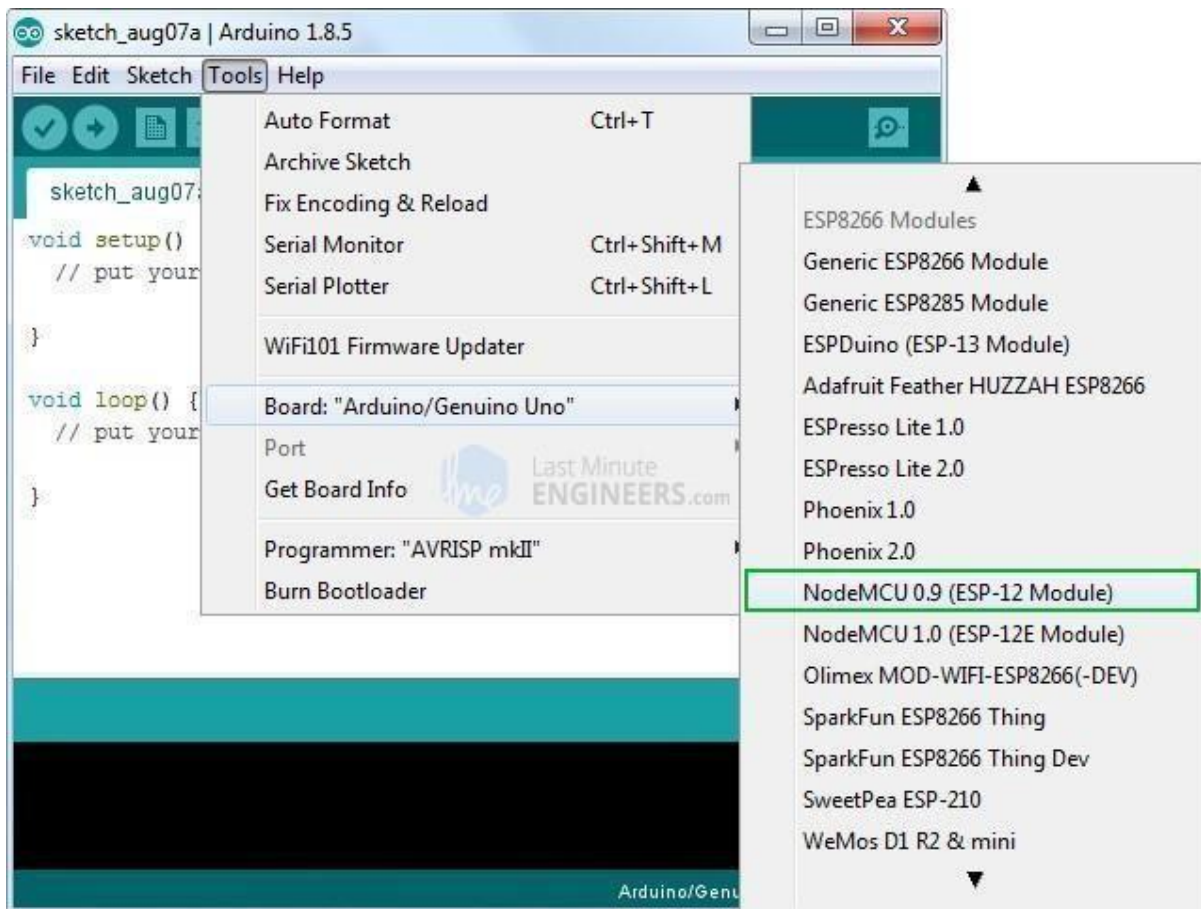


FIG : 4.11 : ARDUNIO

Now, plug your ESP8266 NodeMCU into your computer via micro-B USB cable. Once the board is plugged in, it should be assigned a unique COM port. On Windows machines, this will be something like COM#, and on Mac/Linux computers it will come in the form of /dev/tty.usbserial-XXXXXX. Select this serial port under the Arduino IDE > Tools > Port menu. Also select the Upload Speed : 115200

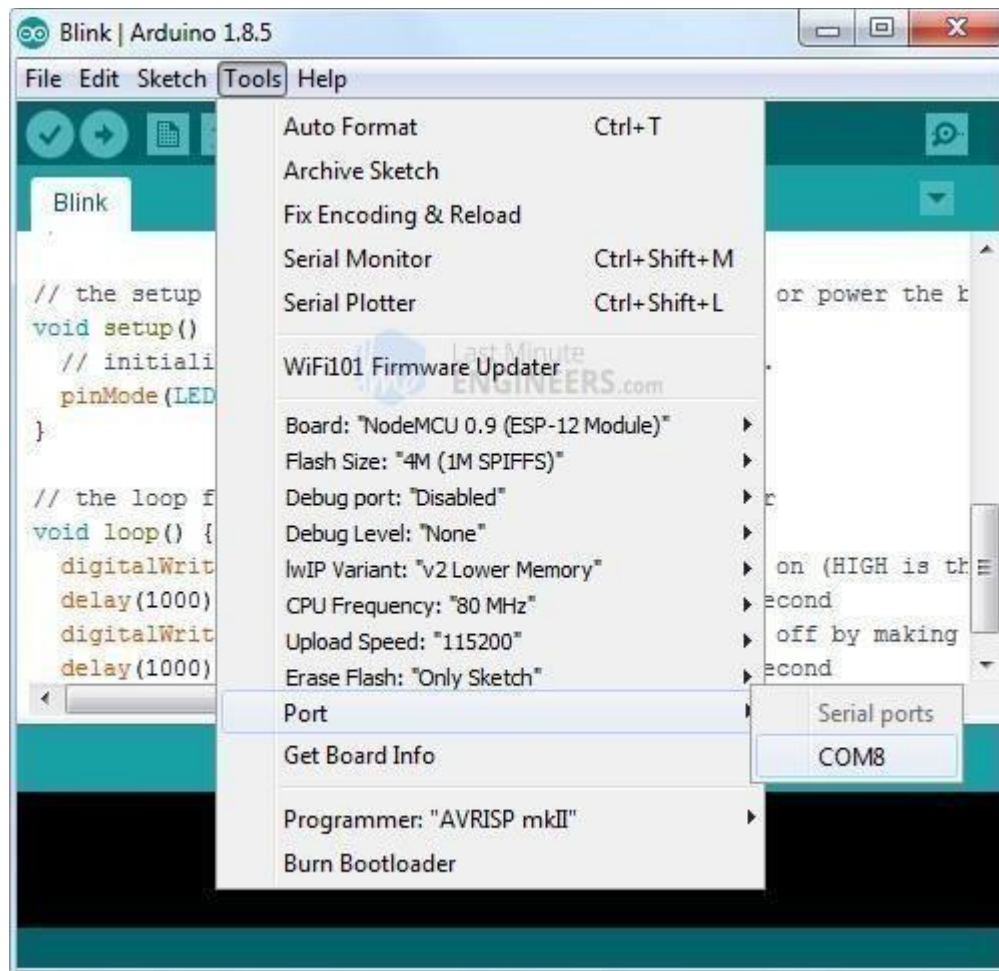


FIG : 4.12 : Port Option

WARNING

More attention needs to be given to selecting board, choosing COM port and selecting Upload speed. You may get espcomm_upload_mem error while uploading new sketches, if failed to do so.

Once you are done, try the example

sketch below.

```
void setup()
```

```

{
    pinMode(D0, OUTPUT);
}
void loop()
{
    digitalWrite(D0,
HIGH);
    delay(500);
    digitalWrite(D0,
LOW);
    delay(500);

}

```

Once the code is uploaded, LED will start blinking. You may need to tap the RST button to get your ESP8266 to begin running the sketch.



FIG : 4.13 : Wifi Chip

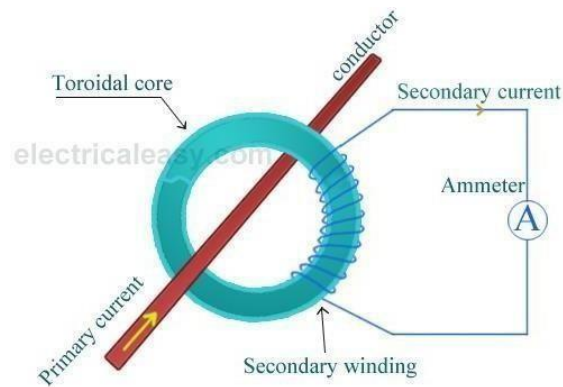
ESP8266-01 Features

- Low cost, compact and powerful Wi-Fi Module
- Power Supply: +3.3V only
- Current Consumption: 100mA
- I/O Voltage: 3.6V (max)
- I/O source current: 12mA (max)
- Built-in low power 32-bit MCU @ 80MHz
- 512kB Flash Memory
- Can be used as Station or Access Point or both combined
- Supports Deep sleep (<10uA)
- Supports serial communication hence compatible with many development platform like Arduino
- Can be programmed using Arduino IDE or AT-commands or Lua Script.

4.9.1 Applications

- 4.7.1 IOT Projects
- 4.7.2 Access Point Portals
- 4.7.3 Wireless Data logging
- 4.7.4 Smart Home Automation
- 4.7.5 Learn basics of networking
- 4.7.6 Portable Electronics
- 4.7.7 Smart bulbs and Sockets

4.9.2 Current transformer



Current Transformer

FIG : 4.14 : current transformer

The **Current Transformer (C.T.)**, is a type of “instrument transformer” that is designed to produce an alternating current in its secondary winding which is proportional to the current being measured in its primary. *Current transformers* reduce high voltage currents to a much lower value and provide a convenient way of safely monitoring the actual electrical current flowing in an AC transmission line using a standard ammeter. The principal of operation of a basic current transformer is slightly different from that of an ordinary voltage transformer.

Unlike the voltage or power transformer looked at previously, the current transformer consists of only one or very few turns as its primary winding. This primary winding can be of either a single flat turn, a coil of heavy duty wire wrapped around the core or just a conductor or bus bar placed through a central hole as shown.



FIG 4.15 : Transformer

Due to this type of arrangement, the current transformer is often referred too as a “series transformer” as the primary winding, which never has more than a very few turns, is in series with the current carrying conductor supplying a load. The secondary winding however, may have a large number of coil turns wound on a laminated core of low-loss magnetic material. This core has a large cross-sectional area so that the magnetic flux density created is low using much smaller cross-sectional area wire, depending upon how much the current must be stepped down as it tries to output a constant current, independent of the connected load.

The secondary winding will supply a current into either a short circuit, in the form of an ammeter, or into a resistive load until the voltage induced in the secondary is big enough to saturate the core or cause failure from excessive voltage breakdown.

Unlike a voltage transformer, the primary current of a current transformer is not dependent of the secondary load current but instead is controlled by an external load. The secondary current is usually rated at a standard 1 Ampere or 5 Amperes for larger primary current ratings.

There are three basic types of current transformers: **wound**, **toroidal** and **bar**.

- Wound Current Transformer – The transformers primary winding is physically connected in series with the conductor that carries the measured

current flowing in the circuit. The magnitude of the secondary current is dependent on the turns ratio of the transformer.

- Toroidal Current Transformer – These do not contain a primary winding. Instead, the line that carries the current flowing in the network is threaded through a window or hole in the toroidal transformer. Some current transformers have a “split core” which allows it to be opened, installed, and closed, without disconnecting the circuit to which they are attached.
- Bar-type Current Transformer – This type of current transformer uses the actual cable or bus-bar of the main circuit as the primary winding, which is equivalent to a single turn. They are fully insulated from the high operating voltage of the system and are usually bolted to the current carrying device.

Current transformers can reduce or “step-down” current levels from thousands of amperes down to a standard output of a known ratio to either 5 Amps or 1 Amp for normal operation. Thus, small and accurate instruments and control devices can be used with CT's because they

are insulated away from any high-voltage power lines. There are a variety of metering applications and uses for current transformers such as with Wattmeter's, power factor meters, watt-hour meters, protective relays, or as trip coils in magnetic circuit breakers, or MCB's. 4.2

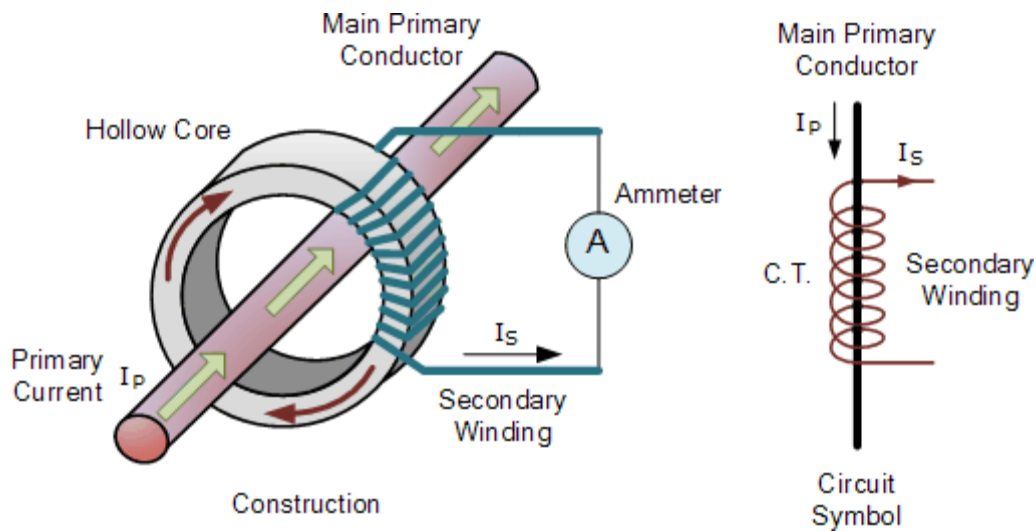


FIG : 4.16 : Construction

Generally current transformers and ammeters are used together as a matched pair in which the design of the current transformer is such as to provide a maximum secondary current corresponding to a full-scale deflection on the ammeter. In most current transformers an approximate inverse turns ratio exists between the two currents in the primary and secondary windings. This is why calibration of the CT is generally for a specific type of ammeter.

Most current transformers have a the standard secondary rating of 5 amps with the primary and secondary currents being expressed as a ratio such as 100/5. This means that the primary current is 20 times greater than the secondary current so when 100 amps is flowing in the primary conductor it will result in 5 amps flowing in the secondary winding. A current transformer of say 500/5, will produce 5 amps in the secondary for 500 amps in the primary conductor, 100 times greater.

By increasing the number of secondary windings, N_s , the secondary current can be made much smaller than the current in the primary circuit being measured because as N_s increases, I_s goes down by a proportional amount. In other words, the number of turns and the current in the primary and secondary windings are related by an inverse

proportion.

A current transformer, like any other transformer, must satisfy the amp-turn equation and we know from our tutorial on double wound voltage transformers that this turns ratio is equal to:

$$\text{T.R.} = n = \frac{N_P}{N_S} = \frac{I_S}{I_P}$$

from which we get:

$$\text{secondary current, } I_S = I_P \left(\frac{N_P}{N_S} \right)$$

The current ratio will sets the turns ratio and as the primary usually consists of one or two turns whilst the secondary can have several hundred turns, the ratio between the primary and secondary can be quite large. For example, assume that the current rating of the primary winding is 100A. The secondary winding has the standard rating of 5A. Then the ratio between the primary and the secondary currents is 100A-to-5A, or 20:1. In other words, the primary current is 20 times greater than the secondary current.

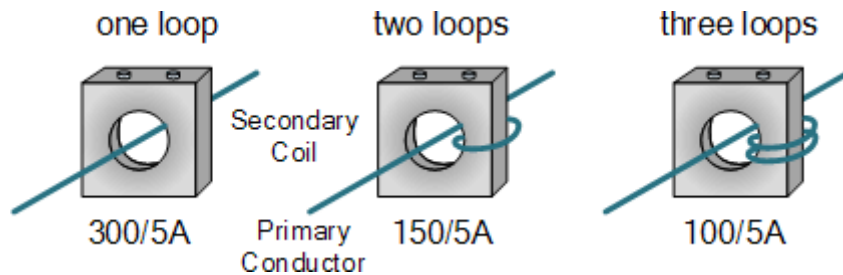
It should be noted however, that a current transformer rated as 100/5 is not the same as operated as 20/1 or subdivisions of 100/5. This is because the ratio of 100/5 expresses the “input/output current rating” and not the actual ratio of the primary to the secondary currents. Also note that the number of turns and the current in the primary and secondary windings are related by an inverse proportion.

But relatively large changes in a current transformers turns ratio can be achieved by modifying the primary turns through the CT’s window where one primary turn is equal to one pass and more than one pass through the window results in the electrical ratio being modified.

So for example, a current transformer with a relationship of say, 300/5A can be

converted to another of 150/5A or even 100/5A by passing the main primary conductor through its interior window two or three times as shown. This allows a higher value current transformer to provide the maximum output current for the ammeter when used on smaller primary current lines.

4.9.3 Current Transformer Primary Turns Ratio



Current Transformer Example No1

A bar-type current transformer which has 1 turn on its primary and 160 turns on its secondary is to be used with a standard range of ammeters that have an internal resistance of 0.2Ω . The ammeter is required to give a full scale deflection when the primary current is 800 Amps.

Calculate the maximum secondary current and secondary voltage across the ammeter.

Secondary Current:

$$I_S = I_P \left(\frac{N_P}{N_S} \right) = 800 \left(\frac{1}{160} \right) = 5A$$

Voltage across Ammeter:

$$V_S = I_S \times R_A = 5 \times 0.2 = 1.0 \text{ Volts}$$

We can see above that since the secondary of the current transformer is connected across the ammeter, which has a very small resistance, the voltage drop across the secondary winding is only 1.0 volts at full primary current.

However, if the ammeter was removed, the secondary winding effectively becomes open-circuited, and thus the transformer acts as a step-up transformer. This is due in part to the very large increase in magnetising flux in the secondary core as the secondary leakage reactance influences the secondary induced voltage because there is no opposing current in the secondary winding to prevent this.

The result is a very high voltage induced in the secondary winding equal to the ratio of: $V_p(N_s/N_p)$ being developed across the secondary winding. So for example, assume our current transformer from above is used on a 480 volt to earth three-phase power line.

Therefore:

$$T.R. = n = \frac{V_P}{V_S} = \frac{N_P}{N_S}$$

$$\therefore V_S = V_P \left(\frac{N_S}{N_P} \right) = 480 \left(\frac{160}{1} \right) = 76,800V \text{ or } 76.8kV$$

This high voltage is because the volts per turns ratio is almost constant in the primary and secondary windings and as $V_s = N_s \cdot V_p$ the values of N_s and V_p are high values, so V_s is extremely high.

For this reason a current transformer should never be left open-circuited or operated with no load attached when the main primary current is flowing through it just as a voltage transformer should never operate into a short circuit. If the ammeter (or load) is to be removed, a short-circuit should be placed across the secondary terminals first to eliminate the risk of shock.

This high voltage is because when the secondary is open-circuited the iron core of the transformer operates at a high degree of saturation and with nothing to stop it, it produces an abnormally large secondary voltage, and in our simple example above, this was calculated at 76.8kV!. This high secondary voltage could damage the insulation or cause electric shock if the CT's terminals are accidentally touched.

4.10 Handheld Current Transformers



FIG : 4.18 : Meter

There are many specialized types of current transformers now available. A popular and portable type which can be used to measure circuit loading are called “clamp meters” as shown.

Clamp meters open and close around a current carrying conductor and measure its current by determining the magnetic field around it, providing a quick measurement reading usually on a digital display without disconnecting or opening the circuit.

As well as the handheld clamp type CT, split core current transformers are available which have one end removable so that the load conductor or bus bar does not have to be disconnected to install it. These are available for measuring currents from 100 up to 5000 amps, with square window sizes from 1" to over 12" (25-to-300mm).

Then to summarise, the **Current Transformer, (CT)** is a type of instrument transformer used to convert a primary current into a secondary current through a magnetic medium. Its secondary winding then provides a much reduced current which can be used for detecting overcurrent, undercurrent, peak current, or average current conditions.

A current transformer's primary coil is always connected in series with the main conductor giving rise to it also being referred to as a series transformer. The nominal secondary current is rated at 1A or 5A for ease of measurement. Construction can be one single primary turn as in Toroidal, Doughnut, or Bar types, or a few wound primary turns, usually for low current ratios. Current transformers are intended to be used as proportional current devices. Therefore a current transformer's secondary winding should never be operated into an open circuit, just as a voltage transformer should never be operated into a short circuit.

Very high voltages will result from open circuiting the secondary circuit of an energized current transformer so their terminals must be short-circuited if the ammeter is to be removed or when a CT is not in use before powering up the system. In the next tutorial about Transformers we will look at what happens when we connect together three individual transformers in a star or delta configuration to produce a larger power transformer called a *Three Phase Transformer* used to supply 3-phase supplies.

4.11 Potential Transformer (PT)



FIG : 4.19 : Transformer

Definition

The potential transformer may be defined as an instrument transformer used for the transformation of voltage from a higher value to the lower value. This transformer step down the voltage to a safe limit value which can be easily measured by the ordinary low voltage instrument like a voltmeter, wattmeter and watt-hour meters, etc.

Construction of Potential Transformer

The potential transformer is made with high-quality core operating at low flux density so that the magnetizing current is small. The terminal of the transformer should be designed so that the variation of the voltage ratio with load is minimum and the phase shift between the input and output voltage is also minimum.

The primary winding has a large number of turns, and the secondary winding has a much small number of turns. For reducing the leakage reactance, the co-axial winding is used in the potential transformer. The insulation cost is also reduced by dividing the primary winding into the sections which reduced the insulation between the layers.

Connection of Potential Transformer

The potential transformer is connected in parallel with the circuit. The primary windings of the potential transformer are directly connected to the power circuit whose voltage is to be measured. The secondary terminals of the potential transformer are connected to the measuring instrument like the voltmeter, wattmeter, etc. The secondary windings of the potential transformer are magnetically coupled through the magnetic circuit of the primary windings.

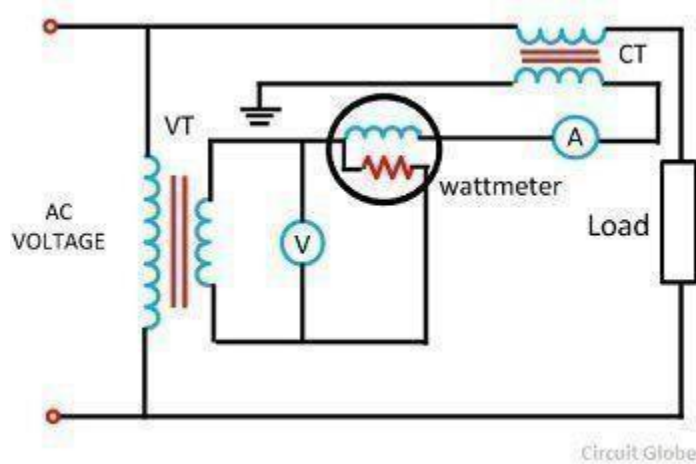


FIG : 4.20 : Circuit Diagram

The primary terminal of the transformer is rated for 400V to several thousand volts, and the secondary terminal is always rated for 400V. The ratio of the primary voltage to the secondary voltage is termed as transformation ratio or turn ratio.

Types of Potential Transformer

The potential transformer is mainly classified into two types, i.e., the conventional wound types (electromagnetic types) and the capacitor voltage potential transformers. Conventional wound type transformer is very expensive because of the requirement of the insulations. Capacitor potential transformer is a combination of capacitor potential divider and a magnetic potential transformer of relatively small ratio.

The circuit diagram of the capacitor potential transformer is shown in the figure below. The stack of high voltage capacitor from the potential divider, the capacitors of two sections becomes C_1 and C_2 , and the Z is the burden.

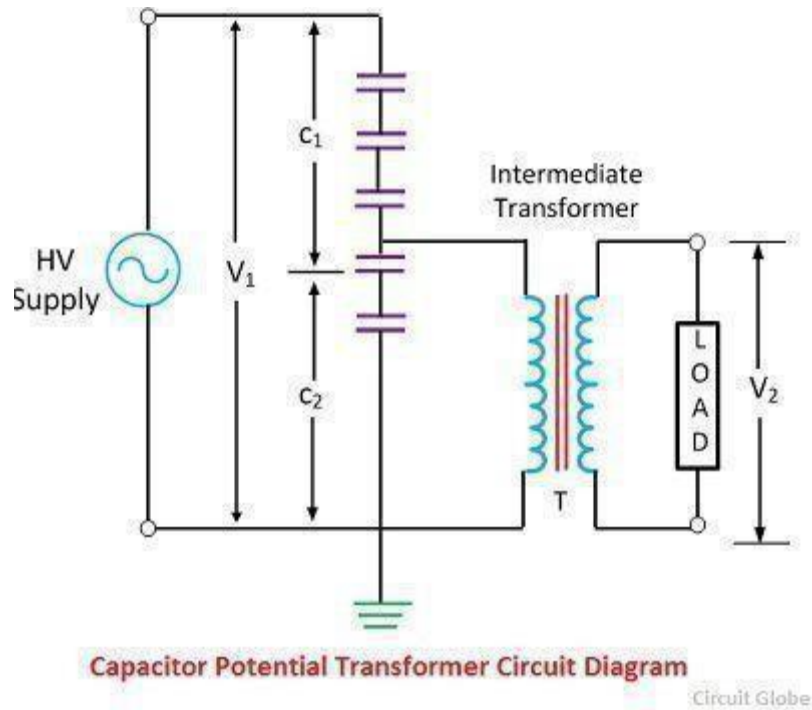


FIG : 4.21 : Capacitor Potential Transformer

The voltage applied to the primary of the intermediate transformer is usually of the order 10kV. Both the potential divider and the intermediate transformer have the ratio and insulation requirement which are suitable for economical construction.

The intermediate transformer must be of very small ratio error, and phase angle gives the satisfactory performance of the complete unit. The secondary terminal voltage is given by the formula shown below.

$$V_2 = V_1 \times \frac{C_1}{C_1 + C_2}$$

Ratio and Phase Angle Errors of Potential Transformer

In an ideal potential transformer, the primary and the secondary voltage is

exactly proportional to the primary voltage and exactly in phase opposition. But this cannot be achieved practically due to the primary and secondary voltage drops. Thus, both the primary and secondary voltage is introduced in the system.

Voltage Ratio Error – The voltage ratio error is expressed in regarding measured voltage, and it is given by the formula as shown below.

$$\text{Ratio Error} = \frac{K_t I_s - I_p}{I_p}$$

Where K_n is the nominal ratio, i.e., the ratio of the rated primary voltage and the rated secondary voltage.

Phase Angle Error – The phase angle error is the error between the secondary terminal voltage which is exactly in phase opposition with the primary terminal voltage.

The increases in the number of instruments in the relay connected to the secondary of the potential transformer will increase the errors in the potential transformers.

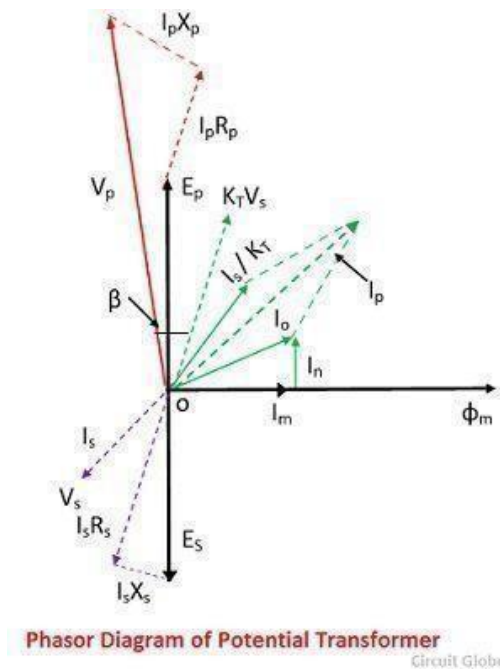
Burden of a Potential Transformer

The burden is the total external volt-amp load on the secondary at rated secondary voltage. The rated burden of a PT is a VA burden which must not be exceeded if the transformer is to operate with its rated accuracy. The rated burden is indicated on the nameplate.

The limiting or maximum burden is the greatest VA load at which the potential transformer will operate continuously without overheating its windings beyond the permissible limits.

This burden is several times greater than the rated burden.

4.11.1 Phasor Diagram of a Potential Transformer



The phasor diagram of the potential transformer is shown in the figure below

Where,

I_s – secondary current

E_s – secondary induced emf

V_s – secondary terminal voltage R_s – secondary winding resistance X_s – secondary winding reactance I_p – Primary current

E_p – primarily induced emf V_p – primary terminal voltage

R_p – primary winding resistance X_p – primary winding reactance K_t – turn ratio

I_o – excitation current

I_m – magnetising component of I_o I_w – core loss component of I_o Φ_m – main flux

β – phase angle error

The main flux is taken as a reference. In instrument transformer, the primary current is the vector sum of the excitation current I_o and the current equal to the reversal secondary current I_s multiplied

by the ratio of $1/k$. The V_p is the voltage applied to the primary terminal of the potential transformer. The voltage drops due to resistance and reactance of primary winding due to primary current is given by $I_p X_p$ and $I_p R_p$. When the voltage drop subtracts from the primary voltage of the potential transformer, the primarily induced emf will appear across the terminals. This primary emf of the transformer will transform into secondary winding by mutual induction and converted into secondary induced emf E_s . This emf will drop by the secondary winding resistance and reactance, and the resultant voltage will appear across the secondary terminal voltage, and it is denoted by V_s .

Applications of Potential Transformer

1. It is used for a metering purpose.
2. For the protection of the feeders.
3. For protecting the impedance of the generators.
4. For synchronising the generators and feeders.

The potential transformers are used in the protecting relaying scheme because the potential coils of the protective device are not directly connected to the system in case of the high voltage. Therefore, it is necessary to step down the voltage and also to insulate the protective equipment from the primary circuit.

4.12 SOFTWARE REQUIRED

- 1) Thing speak cloud web page
- 2) Python platform for NODE MCU

4.12.1 SOFTWARE EXPLANATION:

Cloud

The entire data collected by sensors and camera is stored in cloud. Later on these data can retrieved whenever is needed. These data are highly secured and also these are authenticated by authorised person.

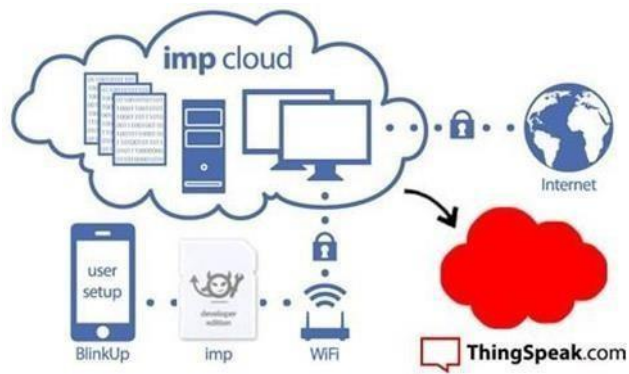


FIG : 4.23 : Cloud Services

A cloud service has three distinct characteristics that differentiate it from traditional web hosting. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access).

Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet, have accelerated interest in cloud computing.

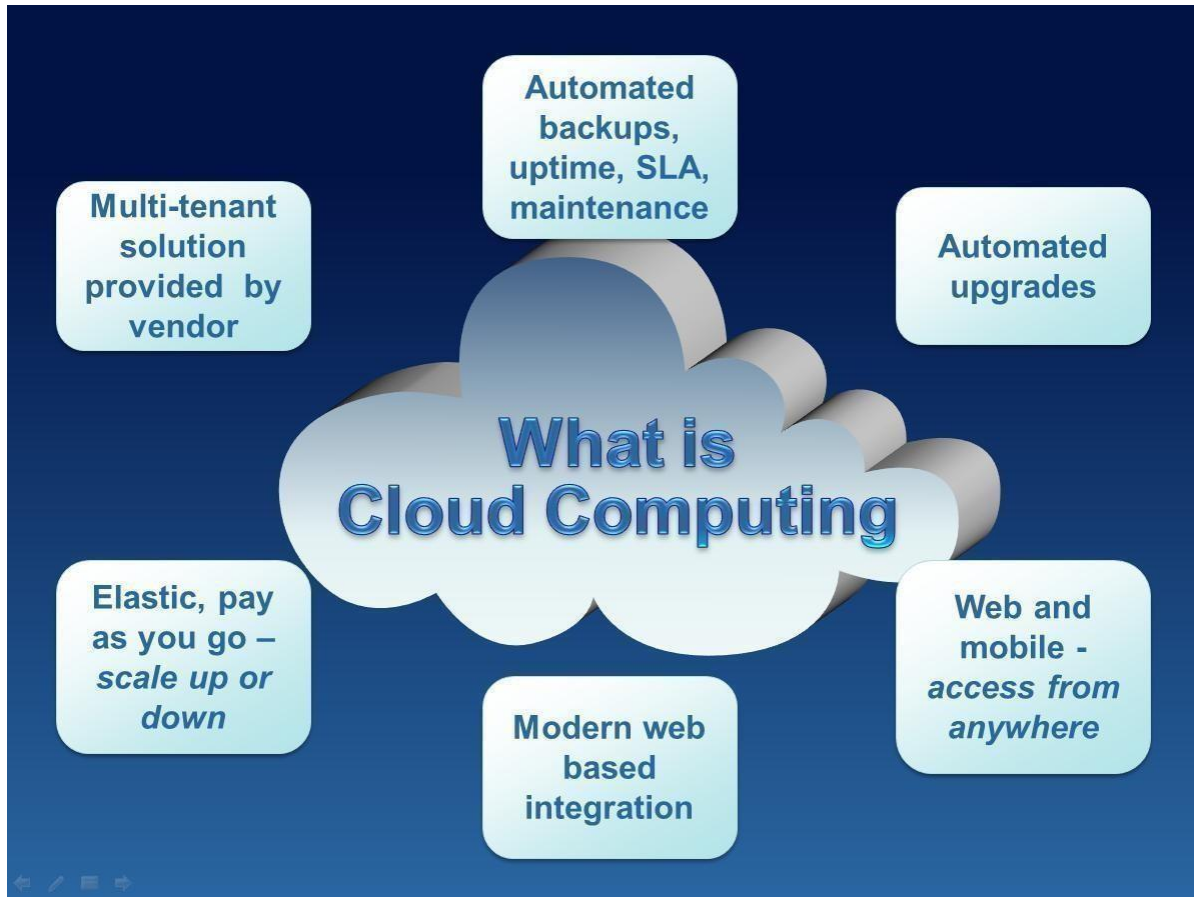


FIG : 4.24 : Cloud Computing

A cloud can be private or public. A public cloud sells services to anyone on the Internet. (Currently, Amazon Web Services is the largest public cloud provider.) A private cloud is a proprietary network or a data center that supplies hosted services to a limited number of people. Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services.

MQTT Protocol:

Multiple clients connect to a broker and subscribe to topics that they are interested in. Clients connect to the broker and publish messages to topics.

Topics are treated as a hierarchy, using a slash (/) as a separator.

Example: multiple computers may all publish their hard drive temperature information on the following topic, with their own computer and hard drive name being replaced as appropriate:sensors/COMPUTER_NAME/temperature/HARDDRIVE_NAME

Clients can receive messages by creating subscriptions. A subscription may be to an explicit topic, in which case only messages to that topic will be received, or it may include wildcards. Two wildcards are available, + or #. MQTT clients can register a custom 'last will testament' message to be sent by the broker if they disconnect. This message can be used to signal to subscribers when a device disconnects.

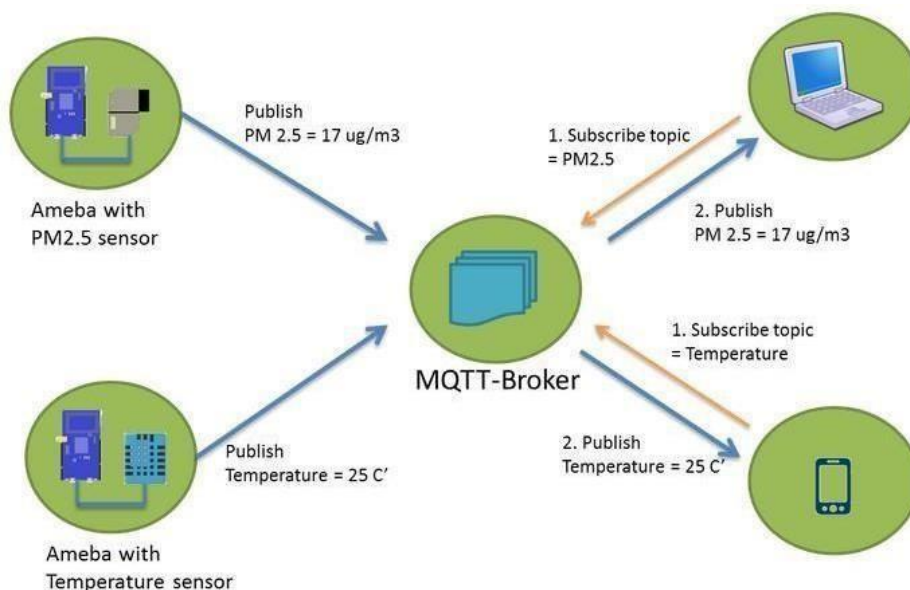


FIG : 4.25 : MQTT-Broker

Response:

A request/response message exchange pattern can be implemented on top of MQTT. The following example shows how:

1) Client

- subscribe topic "function-xyz/response/<id>" //note: <id> is a client unique ID

2) Server

- subscribe topic "function-xyz/request/+" //note: "+" is a wildcard

3) Client

- publish topic "function-xyz/request/<id>" payload <input parameter>

4) Server

- receive notification "function-xyz/request/<id>" payload <input parameter>

- retrieve <id> from string

- process function-xyz(<input parameter>)

- publish topic "function-xyz/response/<id>" payload "<response>"

5) Client

- receive notification "function-xyz/response/<id>" payload "<response>"

4.13 Python

4.13.1 An Introduction to Python

The Raspberry Pi gets the first half of its name from a long-standing tradition of using fruit to name new computing systems— from classic microcomputers like the Acorn, Apricot and Tangerine to more recognizably modern brands including Apple and BlackBerry—but the second half comes courtesy of the Python programming language.

Introducing Python

Flexible and powerful, Python was originally developed in the late 1980s at the National Research Institute for Mathematics and Computer Science by Guido van Rossum as a successor to the ABC language. Since its introduction, Python has grown in popularity thanks to what is seen as a clear and expressive syntax developed with a focus on ensuring that code is readable. Python is a high-level language. This means that Python code is written in largely recognizable English, providing the Pi with commands in a manner that is quick to learn and easy to follow. This is in marked contrast to low-level languages, like assembler, which are closer to how the computer “thinks” but almost impossible for a human to follow without experience. The high-level nature and clear syntax of Python make it a valuable tool for anyone who wants to learn to program. It is also the language that is recommended by the Raspberry Pi Foundation for those looking to progress from the simple Scratch .

Example : Hello World

As you learned in Chapter 10, “An Introduction to Scratch”, the easiest way to learn a new programming language is to create a project that prints “Hello World!” on the

screen. InScratch, you just had to drag and drop bricks of prewritten code, but in Python, you need to write this program entirely by hand. A Python project is, at heart, nothing more than a text file containing written instructions for the computer to follow. This file can be created using any text editor. For example, if you enjoy working at the console or in a terminal window, you can use nano; or if you prefer a graphical user interface (GUI), you can use Leaf pad. Another alternative is to use an integrated development environment (IDE) such as IDLE, which provides Python-specific functionality that's missing from a standard text editor, including syntax checking, debugging facilities and the ability to run your program without having to leave the editor. This chapter gives you instructions on how to create Python files using IDLE, but of course, the IDE program that you choose to use for programming is up to you. The chapter also includes instructions for running your created files directly from the terminal, which can be used in conjunction with any text editor or other IDE. To begin the Hello World project, open IDLE from the Programming menu in the Debian distribution's desktop environment. If you're not using IDLE, create a blank document in your favourite text editor and skip the rest of this paragraph. By default, IDLE opens up in Python shell mode (see Figure 11-1), so anything you type in the initial window will be immediately executed. To open a new Python project which can be executed later, click on the File menu and choose New Window to open a blank file. The IDLE Python Shell window.

It's good practice to start all Python programs with a line known as a shebang, which gets its name from the # and ! characters at the beginning of the line. This line tells the operating system where it should look for the Python files. Although this is not entirely necessary for programs that will be run from within IDLE or will call Python explicitly at the terminal, it is required for programs that are run directly by calling the program's filename. To ensure the program runs regardless of where the

Python executable is installed, the first line of your program should read as follows:

```
#!/usr/bin/env python.
```

This line tells the operating system to look at the `$PATH` environment variable—which is where Linux stores the location of files that can be executed as programs—for the location of Python, which should work on any Linux distribution used on the Pi. The `$PATH` variable contains a list of directories where executable files are stored, and is used to find programs when you type their name at the console or in a terminal window. To achieve the goal of printing out a message, you should use Python's `print` command. As its name suggests, this command prints text to an output device—by default, to the console or terminal window from which the program is being executed. Its usage is simple: any text following the word `print` and placed between quotation marks will be printed to the standard output device.

Enter the following line in your new project:

```
print "Hello, World!"
```

The final program should look like this:

```
#!/usr/bin/env python
```

```
print
```

```
"Hello, World!"
```

If you're creating the example program in IDLE rather than a plain text editor, you'll notice that the text is multi colored, (where colours are represented as differing shades of grey in the print edition). This is a feature known as syntax highlighting, and is a feature of IDEs and the more-advanced text editing tools. Syntax highlighting changes the colour of sections of the text according to their function, in order to make the program easier to understand at a glance. It also makes it easy to spot so-called syntax errors caused by forgetting to put an end-quote in a `print` command or forgetting

to comment out a remark. For this short example, syntaxhighlighting isn't necessary—but in larger programs, it can be an invaluable tool for finding errors. Syntax highlighting in IDLE

Before you run your program, save it as `helloworld.py` using the File menu. If you're using IDLE, the file will be given the extension `.py` automatically. If you're using a text editor, be sure to type `.py` at the end of the filename (not `.txt`) when you save it. This extension indicates that the file contains Python code—although Python is clever enough to run the program even if it's saved with a different file extension. How you run the file will depend on whether you're using IDLE or a text editor. In IDLE, simply choose Run Module from the Run menu, or press the F5 key on the keyboard. This will switch IDLE back to the Python shell window and run the program. You should then see the message `Hello, World!` appear onscreen in blue. If not, check your syntax—in particular, check that you have quotation marks at both the beginning and end of the message on the print line.

Running `helloworld.py` in IDLE

If you created the `helloworld.py` program in a text editor, you'll need to open a terminal window from the Accessories menu on the desktop. If you saved the file anywhere except your home directory, you'll also have to use the `cd` command to change to that directory (see Chapter 2, “Linux System Administration”). Once you're in the right directory, you can run your program by typing the following: `python helloworld.py` This tells the operating system to run Python and then load the `helloworld.py` file for execution. Unlike the Python shell in IDLE, Python will quit when it reaches the end of the file and return you to the terminal. The result, however, is the same: the message `Hello, World!` is printed to the standard output .

Running `helloworld.py` at the terminal

Making Python Programs Executable Normally, the only way to run a Python program is to tell the Python software to open the file. With the shebang line at the top of the file, however, it's possible to execute the file directly without having to call Python first. This can be a useful way of making your own tools that can be executed at the terminal: once copied into a location in the system's \$PATH environment variable, the Python program can be called simply by typing its name. First, you need to tell Linux that the Python file should be marked as executable—an attribute that means the file is a program. To protect the system from malware being downloaded from the Internet this attribute isn't automatically set, since only files that are marked as executable will run. To make the helloworld.py file executable, use the `chmod` command (described in detail in Chapter 2, "Linux System Administration") by typing the following: `chmod +x helloworld.py` Now try running the program directly by typing the following: `./helloworld.py`

Despite the fact that you didn't call the Python program, the helloworld.py program should run just the same as if you'd typed `python helloworld.py`. The program can only be run by calling it with its full location—`/home/pi/helloworld.py`—or from the current directory by using `./` as the location. To make the file accessible in the same way as any other terminal command, it needs to be copied to `/usr/local/bin` with the following command: `sudo cp helloworld.py /usr/local/bin/`

The `sudo` prefix is required because, for security reasons, non-privileged users cannot write to the `/usr/local/bin` directory. With the helloworld.py file located in `/usr/local/bin`, which is included in the \$PATH variable, it can be executed from any directory by simply typing its name. Try changing to a different directory, and then run the program by typing the following: `helloworld.py` To make your custom-made programs seem more like native utilities, you can rename them to remove the .py file extension. To change the helloworld.py program in this way, just type the following

line at the terminal as a single line: `sudo mv`
`/usr/local/bin/helloworld.py`
`/usr/local/bin/helloworld` Once renamed, the program can be run simply by typing
`helloworld` at the terminal or console.

Python and Networking

So far, you have learned how Python can be used to create standalone programs, but the language can also be used to create programs that communicate with the outside world over a computer's network connection. This next example, written by Tom Hudson, offers a brief glimpse of these possibilities with a tool for monitoring the users connected to an Internet Relay Chat (IRC) channel. As usual, create a new project in IDLE or a text editor and enter the shebang line along with a comment describing the purpose of the program:

```
#!/usr/bin/env python
# IRC Channel Checker, written for the Raspberry Pi User Guide by Tom Hudson
```

Next, import the modules required by the program—`sys`, `socket` and `time`—with the following line: `import sys, socket, time` You used the `sys` and `time` modules previously in the Raspberry Snake program, but you have not yet used `socket`. The `socket` module provides Python with the ability to open, close, read from and write to network sockets—giving Python programs rudimentary networking capabilities. It's the `socket` module that provides this example with its ability to connect to a remote IRC server.

There are some constants needed for this program to operate. Constants are like variables in that they can have values assigned to them—but unlike variables, the value in a constant shouldn't change. To help differentiate a constant from a variable, it's good practice to use all-capital letters for their names—that way it's easy to see at glance whether a particular section of the code is using a constant or a variable. Type the following two lines into the program:

```
RPL_NAMREPLY = '353'
```

```
RPL_ENDOFNAMES = '366'
```

These are IRC status codes, provided by the server to indicate when particular operations have completed. These are used by the program to know when it has received the required list of names from the IRC server. Next, set up the variables for the server connection by entering the following lines:

```
irc = {  
    'host':  
    'chat.freenode.net'  
, 'port': 6667,  
    'channel':  
    '#raspiuserguide',  
    'namesinterval': 5  
}
```

The first line tells Python to create a dict data type. Short for dictionary, this allows multiple variables to be stored in a single master variable—in this case, `irc`. These individual variables can then be recalled later in the program. Although you could write this program without using dicts to store variables, it would make the program significantly more difficult to read. The dict begins with the opening curly brace, and ends with the closing curly brace on the final line. The host variable should be set to the fully-qualified domain name (FQDN) of the IRC server to which the program will connect. In this example, `chat.freenode.net` is used, but if you want to customise the program to use a different server, change the domain name here. The port variable tells the program which network port IRC

CHAPTER 5

RESULT AND DISCUSSION

SHORT NOTE OF ELECTRICAL SURVEILLANCE

- The combination of machine learning, IoT, and cloud computing offers a scalable nonintrusive real -
- Time energy monitoring solution that can accommodate a large number of users. The prototype
- Developed for this study showed promising results that would warrant further research. Potential
- Directions to explore would be to test a similar system on actual users over a period of time and to
- Evaluate the eventual effect on energy use behavior. Other machine learning algorithms can also be
- Used to evaluate appliance detection accuracy. Further research can also be conducted on the
- Interface presented to users such as the kind of information to be displayed to users and the kind of
- Features desired by users to assist them in saving energy. Data collect...

LIMITATION

Consider a complicated power system network, it is very usual to get faults in every branches of the network. In this system all the loads will be monitored and examined through online using IoT. In case of losses also can be determined by this system by comparing the past values or normal values with present value. If there is any variation occurred can be intimated by this system. Electronic metering technology greatly reduces the man power and time and also made easy for remote area people for paying the bill. It also reduces the non-payment of the bill and avoids the mistake due to manual calculation. Power theft is one of the greatest problems that our country is facing and with the help of this AME power theft can be greatly reduced. Electronic meters show the power consumed, power factor, the reactive power used digitally displayed on LCD or LED display, and also able to send the energy consumed readings

to remote places through some communication network. It is a single way communication, Fault can be occurred in a power system at any time, it can be due to overvoltage, short circuit etc. consider a network got affected by fault in a particular branch. In case of our normal system this cannot be identified easy

5.3 ALERT MESSAGE SYSTEM

There are many alert message techniques available. Each technique has its own drawbacks, So necessary knowledge is required to utilize a technique balancing all the needs and energy consume and alert message are shown in a below figure.

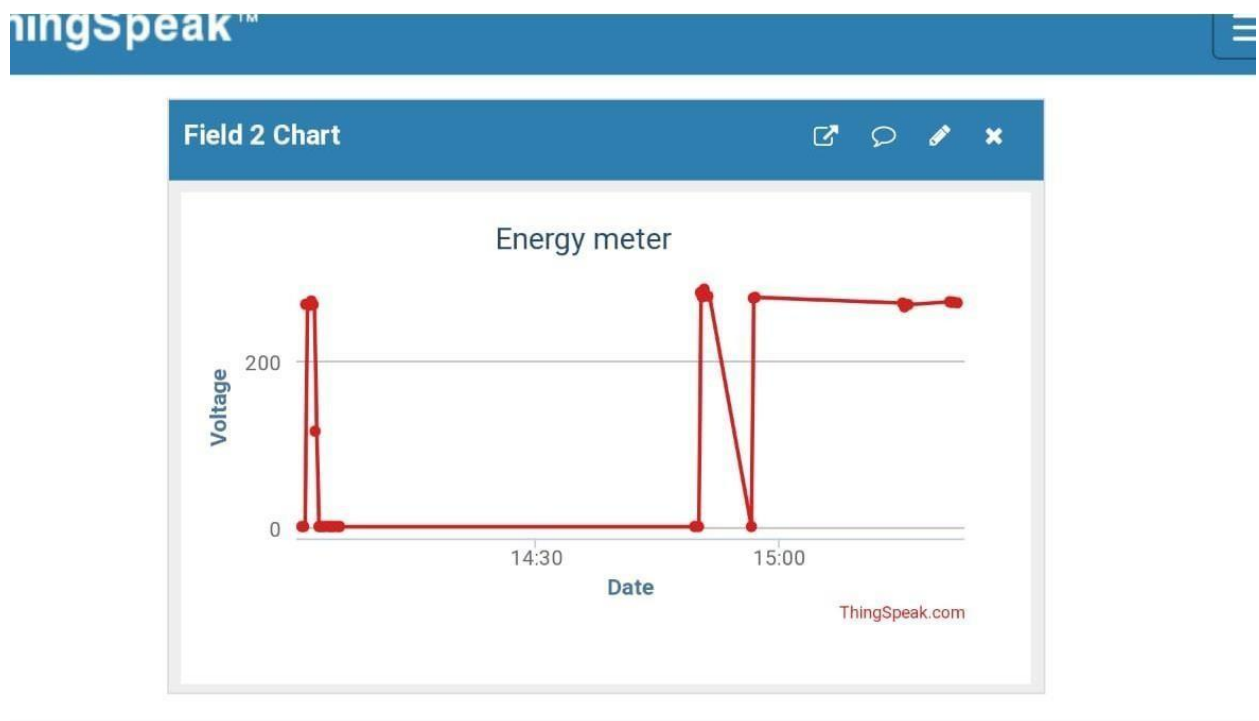


FIG : 5.1 : Energy Meter voltage

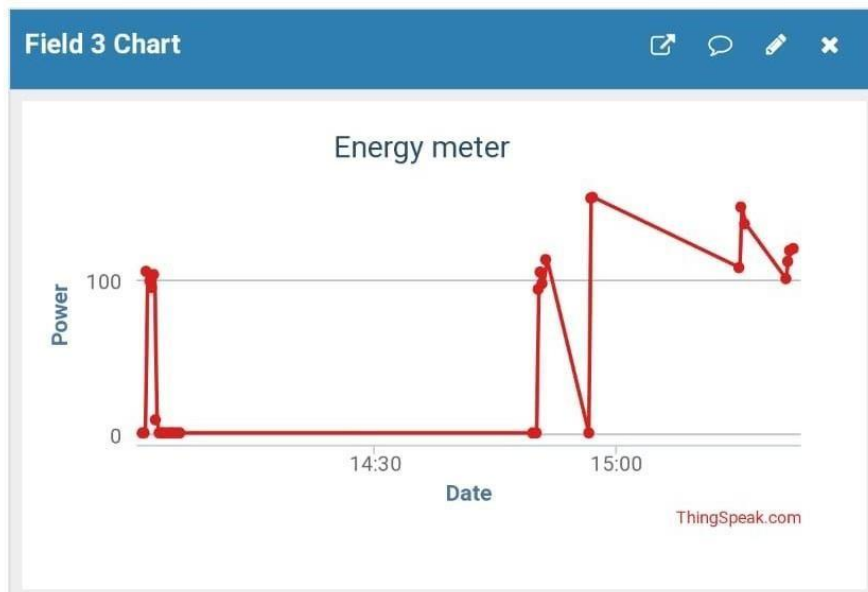


FIG : 5.2 : power



FIG: 5.3 : UNIT

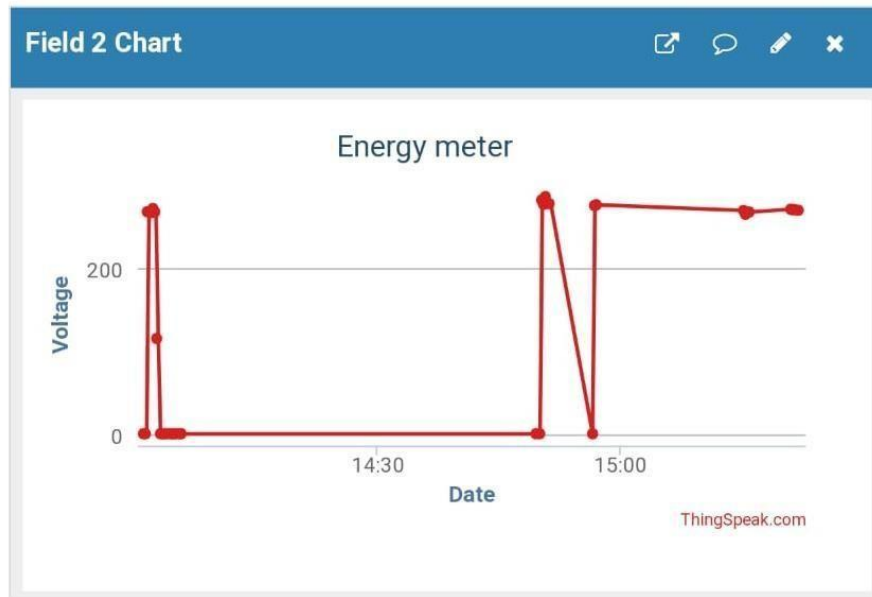


FIG: 5.4 : VOLTAGE



FIG :5.5: CURRENT



FIG : 5.6 : ENERGY

CHAPTER 6

CONCLUSION AND FUTURE WORKS

This paper is the combined hardware advantage for both utility and the customer. NODE MCU, SSR, and IoT stationed Energy Meter for smart metering, is built which is able to read and send data via wireless protocol using IoT technology through IoT module, capable of managing and controlling the supply. In the case of faults, losses, and faulty loads, Power consumption, power quality, and its accuracy can be monitored by the consumers directly.

- System can be enhanced for power theft detection using IOT.
- The android application can append options for online payment of bill.
- In the proposed method load analysis of single entity is performed; in future it can be boosted for area wise load analysis which will help for load forecasting.

REFERENCES

- [1] Aryo H. Primicanta, MohdYunusNayan and MohdAwan, “ZigBee-GSM based AutomaticMeter Reading System”, IEEE 2008
- [2] B.S Koay, S.S Cheah, Y.H Sng, P.H.J Chong, P Shum, Y.C Tong , X.Y Wang, Y.X Zuoand H.W Kuek, “Design and Implementation of Bluetooth Energy Meter”, IEEE 2003
- [3] G. Thavasi Raja, T. D. Sudhakar “Electricity consumption and automatic billing throughpower line”, IEEE 2007
- [4] Karen Rose, Scott Eldridge and Lyman Chapin, “The Internet of things An OverviewUnderstanding the Issues and Challenges of a More Connected World”, Internet Society,October 2015
- [5] Manisha V Shinde&Pradip W Kulkarni, “Automation of Electricity Billing Process”, IEEE 2014
- [6] RozitaTeymourzadeh , S Mahmud Iwan & Ahmed J. A. Abueida , “RFID-based prepaidpower meter”, IEEE 2013



Plagiarism Checker X Originality Report

Similarity Found: 13%

Date: Thursday, April 15, 2021

Statistics: 499 words Plagiarized / 12217 Total words

Remarks: Low Plagiarism Detected - Your Document Improved.

9) The consumer may not receive his energy bill as per regular interval of the due date
1.7 MODULES Current Transformer (CT): CT is used to measure or monitor the current in transmission lines and to isolate the metering equipment and relay connected to secondary side. The current level from energy meter by the help of current transformer will be transmitted to cloud by IoT module Fig : 1.2 : current Transformer 5 1.7.1.

Voltage Transformer or Potential Transformer (PT): PT is used to measure or monitor the voltage on transmission lines and to isolate the metering equipment from the lines. The voltage level from energy meter by the help of current transformer will be transmitted to cloud by IoT module. Fig 1.3 : Potential Transformer 1.7.2 Node MCU: This module is the controller with inbuilt Wi-Fi. With the help of this modules the data obtained to the transformer. These transformers are used to measure the current and voltage level and stored into the cloud 6 Fig 1.4 : Pin Diagram 1.7.3

Cloud The entire data collections from transformer are stored in the cloud. These stored data are highly secured. The stored data can recognize by authorized persons. Fig : 1.5 : Cloud Data 7 CHAPTER 2 2.LITERATURE REVIEW This chapter discusses we about the literature survey and summary. It also gives a detailed idea on the existing. 2.1.

LITERATURE SURVEY 1.SMS-Based Reconfiguration Automatic Meter Reading System 2007-Ali Abdollahi, marajanDehghani, and Negar Zamanzadeh In recent years many engineers and companies have been working on Automatic Meter Reading (AMR), and different media, such as radio, telephone lines, and power lines have been used for this purpose. Nowadays, GSM network, with its vast coverage in most countries and also its competitive ever-growing market, is becoming popular as the main medium for the machine applications, and AMR is not an exception.

2.Smart energy meter brindas1 ,vishalkumarsah2,jaladi,harish3,akshayu4,vishal,deomatho5,swetha umapathy6 ---(2008) In the currently working system, electricity meter reading for electricity usage and building is done by human workers from home to home and buildings. The purpose of this project is to develop a smart electricity meter using GSM. This can reduce human error and helps to retrieve the real time meter value via GSM and sent it to customers mobile phone through GSM.

This also allow the electricity board to modify the variable package price in specific duration. The administrator can analyse the customers power consumption data and generate the report from the data online. 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---GSM,Power Consumption billing ,smart electricity meter. 3.Data Collecting From smart meter in an advanced metering infrastructure 2011 M.popa- - This paper describes an AMI based system. Smart meters are connected to a 8 gateway through power lines and the gateway communicates with a central computer through GSM.

The power line communication is implemented with an industrial bus, LonWork. The paper focuses the data collection solution. plc,zigbee,gsm 4. Design of an automatic meter reading system[electricity metering] 2012 shi- weilee;cheng shong wu; meng shi chiou; kou-tan wu.-- A microprocessor-based automatic meter-- reading system is implemented, which provides a cost-effective , reliable, and interference free data transfer between remote meter reading units and the utility control center.

The meter reading and management processes are free from human involvement. Based on the existing telephone networks, it is very flexible for the utility companies to access,



service and maintain this meter reading system. — AutomaticMeterReading,Utility public Telephone.Network

2.2 LITERATURE SUMMARY

The Survey has been done for the IOT based electrical surveillance and automatic meter reading. Upon the analysing many papers , it can be concluded that using IOT surveillance we can produce or we can stop the power theft and monitor in the work space and it could be easy and convenient and it can be used to billing conserve our household energy and we send consumed energy via mobile sms.

2.3. PROBLEM DEFINITION

1. Highly depends on meter reader
2. Human error cannot be avoided for the manual meter reading.
- 3.

secondary winding reactance I_p Primary current E_p primarily induced emf V_p primary terminal voltage R_p primary winding resistance X_p primary winding reactance K_t turn ratio I_o excitation current I_m magnetising component of I_o I_w main flux - phase angle error The main flux is taken as a reference. In instrument transformer, the primary current is the vector sum of the excitation current I_o and the current equal to the reversal secondary current I_s multiplied by the ratio of $1/k_t$.

The V_p is the voltage applied to the primary terminal of the potential transformer. The voltage drops due to resistance and reactance of primary winding due to primary current is given by $I_p X_p$ and $I_p R_p$. When the voltage drop subtracts from the primary voltage of the potential transformer, the primarily induced emf will appear across the terminals. This primary emf of the transformer will transform into secondary winding by mutual induction and converted into secondary induced emf E_s .

This emf will drop by the secondary winding resistance and reactance, and the resultant voltage will appear across the secondary terminal voltage, and it is denoted by V_s .

Applications of Potential Transformer

1. It is used for a metering purpose.
2. For the protection of the feeders.
3. For protecting the impedance of the generators.
4. For synchronising the generators and feeders.

The potential transformers are used in the protecting relaying scheme because the potential coils of the protective device are not directly connected to the system in case of the high voltage.

