IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 14, Issue 4, Ser. I (Jul.-Aug. 2019), PP 39-50
www.iosrjournals.org

Internet of Things (IoT) based automatic electrical energy meter billing system

Shahajan Miah[1], G.M. Jahedul Islam[2], Sajib Kanti Das[2], Shahidul Islam[2], Mirajul Islam[2], Kazi Kamrul Islam[2]
[1, 2] Department of EEE, Bangladesh University of Business & Technology, Dhaka-1216
Corresponding Author: Shahajan Miah

Abstract: *We developed an automatic electrical energy meter billing system based on IoT (Internet of Things) for use in homes, offices, factories, etc. Electricity became one of the basic requirements of human life and it is widely used for domestic, industrial and agricultural purposes. In most cases, consumers are not satisfied with the maintenance of electrical networks. They have complaints about statistical errors in monthly bills. Therefore, we present ideas to minimize the technical mistakes and, at the same time, try to reduce dependence on a person. In addition, the important concept used in this work is IoT (Internet of Things). IoT is an emerging technology with unique identifiers and having the ability to transfer data over a network without human-to-human or human-to-computer interaction. Applying this technology in this work we have tried to send the total energy consumption date in the form of email from the remote location directly to the consumer and electricity board via internet. Additionally, we have created a separate web page and upload the meter readings to the webpage. In the proposed system, consumers can control the power, from time to time knowing the energy consumption. The generated invoice can be displayed on the web page via the Wi-Fi module. The customer can pay the bill, as planned. We can reduce the loss of revenue to the government and we can use electricity more efficiently.*

Keywords *- IOT, ESP8266 Module, Electrical Energy Meter, GPIO, I^2C*

Date of Submission: 22-07-2019 Date of acceptance: 07-08-2019

I. Introduction

The rapid development of information technology (IT) led to the creation of a hyperlink community in which objects are connected to mobile devices and the internet and communicate with each other. In the 21st century, I would like to lead something at any time and in any place that already happens in different places around the world. The central element of this super connected society is IoT, which is also called inter-machine communication technology. This technology has been improved with advanced technologies, such as Big Data and Hadoop. This is expected to be the next biggest thing that affects our life in different ways. Although, IoT is a new technology, it is believed to make significant changes in the history of computing. Sensors built into the car, implants for heart monitoring, biochip transponders and intelligent thermostat systems are examples of these. Such devices can be adapted to the needs of the business. As on real devices, it is expected that IoT devices will be able to communicate with people [**1**]

1. ESP8266 Module

ESP8266 Module is a set of high-performance highly integrated wireless SOC, designed for developers of mobile platforms with limited space and power. Wi-Fi functions can be included in other systems. It also functions as a stand-alone application, minimizing costs and minimizing space. ESP 8266 provides a complete and standalone Wi-Fi network solution. It can be used to host applications or to unload Wi-Fi network functions from another application processor. When the ESP8266EX starts the application, it starts directly from the external flash. In has built-in caching to improve system performance with these applications. You can also add wireless Internet access to a microcontroller-based project that functions as a Wi-Fi adapter and has convenient connectivity (SPI / SDIO or I2C / UART interface). ESP8266EX is among the most integrated Wi-Fi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

ESP8266EX also integrates an enhanced version of Ten silica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; sample codes for such applications are provided in the software development kit (SDK). [**2**]

Fig 1.1: ESP8266 SoC IC [**2**]

2. Ultra Low Power Technology

ESP8266EX is designed for mobile, wearable electronics, internet applications for soles aimed at minimizing energy consumption by combining several unique technologies. Energy-saving architecture mainly works in three modes: active mode, standby mode and deep sleep mode. The ESP 8266 EX consumes about 60 μA with extended standby (the RTC clock is still running) and uses methods and power management logic before it needs less than 1.0 mA (DTIM = 3) power consumption Turn off unnecessary functions) or leave it connected to Access point with less than 0.5 mA (DTIM = 10).

In the sleep mode, only the calibrated real-time clock and the watchdog remain. The real-time clock can be programmed to wake the ESP 8266EX at any desired intervals. The ESP 8266 EX can be programmed to wake up when a certain state is detected. This minimum wake-up time function of the ESP 8266 EX is available to the SOC of the mobile station, so that it can support low-power standby mode until Wi-Fi is required. In order to satisfy the power demand of mobile and wearable electronics, ESP8266EX can be programmed to reduce the output power of the P A to fit various application profiles, by trading off range for power consumption [**3**].

1.3 . Hardware Overview

3. 1. Pin Mapping

ESP8266 has 17 I/O pins, 2 power pin, 1 ground pin, reset pin and 2 clock pins shown in **Fig 1. 2.**

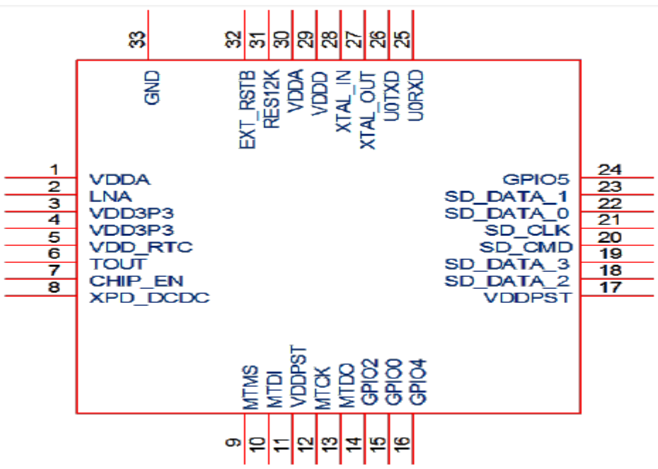

Fig 1.2: Pin Diagram of ESP8266 [**2**]

Pin Assignments

01 - VDDA P Analog Power 3.0 ~3.6V
02 - LNA I/O
03 - VDD3P3 P Amplifier Power 3.0~3.6V
04 - VDD3P3 P Amplifier Power 3.0~3.6V

05 - VDD_RTC P NC (1.1V)
06 - TOUT I & ADC Pin
07 - CHIP_EN I Chip Enable.
08 - XPD_DCDC I/O Deep-Sleep Wakeup；GPIO16
09 - MTMS I/O GPIO14; HSPI_CLK
10 - MTDI I/O GPIO12; HSPI_MISO
11 - VDDPST P Digital/IO Power Supply (1.8V~3.3V)
12 - MTCK I/O GPIO13; HSPI_MOSI; UART0_CTS
13 - MTDO I/O GPIO15; HSPI_CS; UART0_RTS
14 - GPIO2 I/O UART Tx during flash programming; GPIO2
15 - GPIO0 I/O GPIO0; SPI_CS2
16 - GPIO4 I/O GPIO4
17 - VDDPST P Digital/IO Power Supply (1.8V~3.3V)
18 - SDIO_DATA_2 I/O Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPIHD; GPIO9
19 - SDIO_DATA_3 I/O GPIO10
20 - SDIO_CMD I/O Connect to SD_CMD (Series R: 200Ω); SPI_CS0; GPIO11
21 - SDIO_CLK I/O Connect to SD_CLK (Series R: 200Ω); SPI_CLK; GPIO6
22 - SDIO_DATA_0 I/O Connect to SD_D0 (Series R: 200Ω); SPI_MSIO; GPIO7
23 - SDIO_DATA_1 I/O Connect to SD_D1 (Series R: 200Ω); SPI_MOSI; GPIO8
24 - GPIO5 I/O GPIO5
25 - U0RXD I/O UART Rx during flash programming; GPIO3
26 - U0TXD I/O UART Tx during flash programming; GPIO1; SPI_CS1
27 - XTAL_OUT I/O Crystal oscillator output
28 - XTAL_IN I/O Connect to crystal oscillator input
29 - VDDD P Analog Power 3.0V~3.6V
30 - VDDA P Analog Power 3.0V~3.6V
31 - RES12K I
32 - EXT_RSTB I External reset signal (Low voltage level: Active)

1. 3. 2. MCU

The ESP8266EX includes a 32-bit Ten silica L106 microcontroller (MCU) with low power consumption and a 16-bit RSIC. The CPU clock speed is 80MHz. It can also reach a maximum value of 160MHz. Real Time Operation System (RTOS) is enabled. Currently, only 20% of MIPS has been occupied by the Wi-Fi stack, the rest can all be used for user application programming and development [**3**].
The following interfaces can be used to connect to the MCU embedded in ESP8266EX:

Programmable RAM/ROM interfaces (iBus), which can be connected with memory controller, and can also be used to visit external flash;
Data RAM interface (dBus), which can connected with memory controller;
AHB interface can be used to visit the register.

1.4 . Memory Organization

1. 4.1. Internal SRAM and ROM

ESP8266EX Wi-Fi SoC is embedded with memory controller, including SRAM and ROM. MCU can visit the memory units through iBus, dBus, and AHB interfaces. All memory units can be visited upon request, while a memory arbiter will decide the running sequence according to the time when these requests are received by the processor.
According to our current version of SDK provided, SRAM space that is available to users is assigned as below:

- RAM size < 36kB, that is to say, when ESP8266EX is working under the station mode and is connected to the router, programmable space accessible to user in heap and data section is around 36kB.)
- There is no programmable ROM in the SoC, therefore, user program must be stored in an external SPI flash.

Block Diagram of ESP8266:

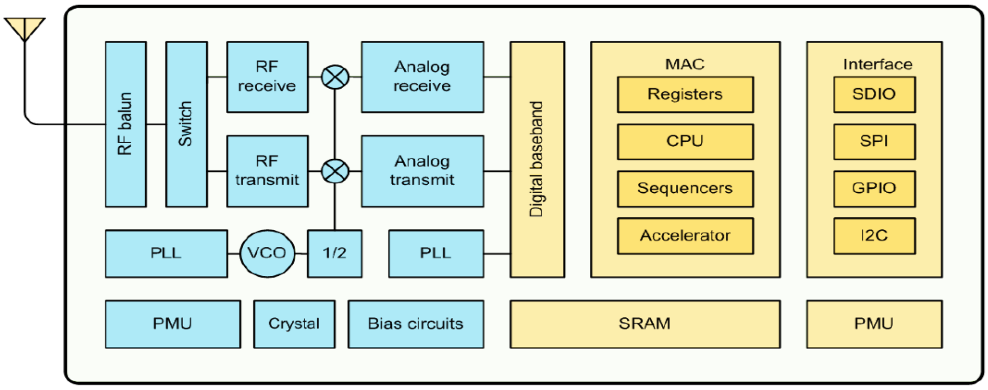

Fig 1.3: Block Diagram of ESP8266 [**2**]

1.4.2. External SPI Flash

An external SPI flash is used together with ESP8266EX to store user programs. Theoretically speaking, up to 16 Mbyte memory capacity can be supported. Suggested SPI Flash memory capacity:

- OTA is disabled: the minimum flash memory that can be supported is 512 kByte;
- OTA is enabled: the minimum flash memory that can be supported is 1 Mbyte.

Several SPI modes can be supported, including Standard SPI, Dual SPI, DIO SPI, QIO SPI, and Quad SPI.

1.5. GPIO

There are up to 17 GPIO pins. The firmware is assigned to various functions. Each GPIO has an internal pull-up (excluding XPD_DCDC configured with internal ejection), an input that can be used for sampling by software registers, an input that triggers an interrupt of a processor or a CPU level, an input that causes a wake-up interrupt, an open Drain or push output driver, or the source of the software output signal, or the sigma-delta PWM DAC. These pins are multiplexed with other functions such as I2C, I2S, UART, PWM, IR Remote Control, etc. Data I/O soldering pad is bidirectional and tri-state that include data input and output controlling buffer. Besides, I/O can be set as a specific state and remains like this. For example, if you intend to lower the power consumption of the chip, all data input and output enable signals can be set as remaining low power state. You can transport some specific state into the I/O. When the I/O is not powered by external circuits, the I/O will remain to be the state that it was used the last time.

Therefore, positive feedback is generated by the residual state function of the pin state, if the external drive power must be stronger than the positive feedback. Nevertheless, the required driving force is within 5 µA. All digital IO pins are protected against overvoltage by connecting the instantaneous switching circuit between the pad and the ground. The wrapping voltage is typically around 6 V, and the holding voltage is 5.8 V. This provides surge protection and ESD protection. In addition, the output device is protected by a diode against reverse voltage. [**3**]

1.6. Inter-Integrated Circuit Interface (I^2C)

One I^2C, which is mainly used to connect with micro controller and other peripheral equipment such as sensors, is defined by ESP8266EX. The present pin definition of I^2C is as defined : Both I^2C-Master and I^2C-Slave are supported. I^2C interface functionality can be realized via software programming, the clock frequency can be up to around 100 KHz at most. It should be noted that I^2C clock frequency should be higher than the slowest clock frequency of the slave device [**4**].

1.6. 1. I^2C – Inter-Integrated Circuit Communication Protocol

Features

- Full-duplex, Three-wire Synchronous Data Transfer
- Master or Slave Operation
- Master node – node that generates the clock and initiates communication with slaves.
- Slave node – node that receives the clock and responds when addressed by the master.
- The bus is a multi-master bus, which means that any number of master nodes can be present. Additionally, master and slave roles may be changed between messages (after a STOP is sent).
- There may be four potential modes of operation for a given bus device, although most devices only use a single role and its two modes:

- ✓ Master transmit – master node is sending data to a slave,
- ✓ Master receive – master node is receiving data from a slave,
- ✓ Slave transmit – slave node is sending data to the master,
- ✓ Slave receive – slave node is receiving data from the master.

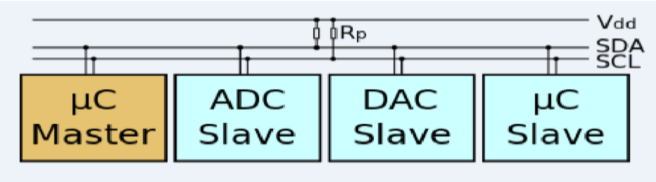

Fig 1.4: I^2C Connection Diagram

1.7. Analog to Digital Converter

The device features a 10-bit successive approximation ADC. The ADC is connected to an 8-channelAnalog Multiplexer which allows eight single-ended voltage inputs constructed from the pins of Port A. The single-ended voltage inputs refer to 0V (GND).The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown below. The ADC has a separate analog supply voltage pin, AV_{CC}. AV_{CC} must not differ more than ±0.3V from V_{CC}. The Power Reduction ADC bit in the Power Reduction Register (PRR &PRADC) must be written to '0' in order to be enable the ADC. The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin minus 1LSB. Optionally, AV_{CC} or an internal 1.1V reference voltage may be connected to the AREF pin by writing to the REFSn bits in the ADMUX Register. The internal voltage reference must be decoupled by an external capacitor at the AREF pin to improve noise immunity [**4**].

1.8. Power Management

The chip can be put into the following states:

- OFF: CHIP_PD pin is low. The RTC is disabled. All registers are cleared.
- DEEP_SLEEP: Only RTC is powered on – the rest of the chip is powered off. Recovery memory of RTC can keep basic Wi-Fi connecting information.
- SLEEP: Only the RTC is operating. The crystal oscillator is disabled. Any wakeup events (MAC, host, RTC timer, external interrupts) will put the chip into the WAKEUP state.
- WAKEUP: In this state, the system goes from the sleep states to the PWR state. The crystal oscillator and PLLs are enabled.
- ON: the high speed clock is operational and sent to each block enabled by the clock control register. Lower level clock gating is implemented at the block level, including the CPU, which can be gated off using the WAITI instruction, while the system is on.

1.9. Clock Management

The high frequency clock on ESP8266EX is used to drive both transmit and receive mixers. This clock is generated from the internal crystal oscillator and an external crystal. The crystal frequency can range from 26MHz to 52MHz. While internal calibration of the crystal oscillator ensures that a wide range of crystals can be used, in general, the quality of the crystal is still a factor to consider, to have reasonable phase noise that is required for good performance. When the crystal selected is sub-optimal due to large frequency drifts or poor Q-factor, the maximum throughput and sensitivity of the Wi-Fi system is degraded. Please refer to the application notes on how the frequency offset can be measured [**5**]

1. 10. Radio

The ESP8266EX radio consists of the following main blocks:

- 2.4GHz receiver
- 2.4GHz transmitter
- High speed clock generators and crystal oscillator
- Real time clock
- Bias and regulators
- Power management

1. 11. NodeMCU 12E (ESP8266) Development Board
The NodeMCU is an open-source firmware and development kit that helps you toPrototype your IOT product within a few Lua script lines.

Pin Configuration
10 GPIO, every GPIO can be PWM, I^2C, 1-wire communication interface.

1. 12. Arduino-like hardware IO
Advanced API for hardware I/O, which can dramatically reduce the redundant work for configuring and manipulating hardware.

- Code like Arduino, but interactively in Lua script.
- Nodejs style network API
- Event-driven API for network applications, which facilitates developers writing code running on a 5mm x 5mm sized MCU inNodejs style.
- Greatly speed up your IOT application developing process.

Specification:
The Development Kit based on ESP8266, integrates GPIO, PWM, I^2C, 1-Wire and ADC all intone board.Power your development in the fastest way combination with NodeMCU Firmware!

- USB-TTL included, plug&play
- 10 GPIO, every GPIO can be PWM, I^2C, 1-wire
- FCC CERTIFIED WI-FI module
- PCB antenna

1. 13. Electrical Energy Meter
Electricity meters are devices that measure the amount of electricity consumed by homes, businesses or electrical devices. Electric networks use electric meters installed in the premises of the customer, which are usually calibrated in billing devices, the most frequent are kilowatt-hours [kWh]. They are usually downloaded once during each billing cycle. If energy conservation is desired for a certain period of time, some meters can measure demand, which is the maximum use of electricity at certain intervals [**6**].

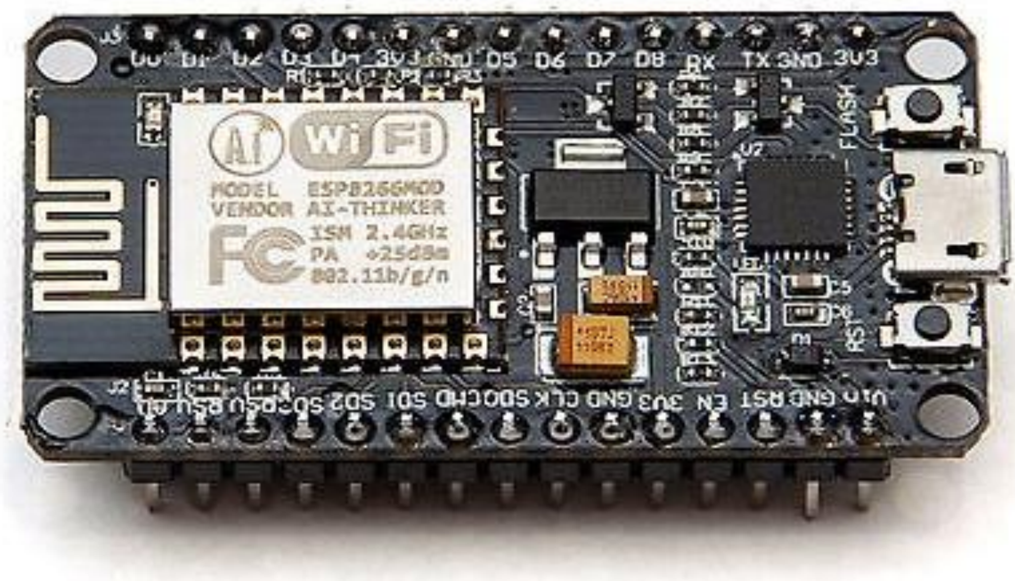

Fig 1.5: NodeMCU 12E ESP8266 Module

1. 14. Power Supply
The +5 volt power supply is based on the commercial 7805 voltage regulator IC. This IC contains all the circuitry needed to accept any input voltage from 8 to 18 volts and produce a steady +5 volt output, accurate to within 5% (0.25 volt). It also contains current-limiting circuitry and thermal overload protection, so that the IC won't be damaged in case of excessive load current; it will reduce its output voltage instead.

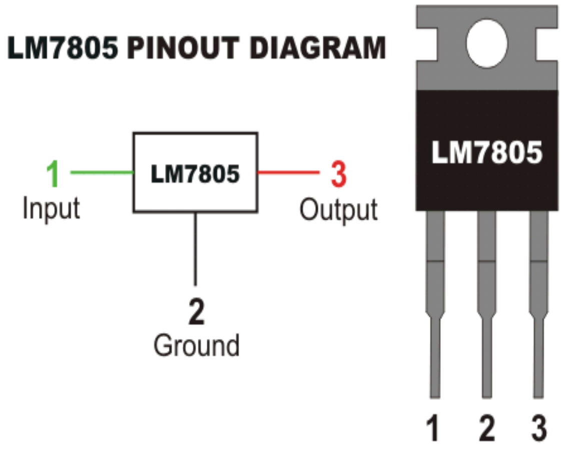

Fig 1.6: Voltage Regulator PIN Diagram

1. 15. Vero Board

A thin layer of copper foil is laminated on one or both sides of the FR-4 glass epoxy panel. They are usually called laminated copper layers. A clad copper sheet is created by an interconnected bond etched into a copper layer to make a printed circuit board. More complex and complex circuit boards are manufactured with several layers, also known as multilayer circuits. In this project, printed circuit boards are manufactured using a one-sided copper clad laminate.

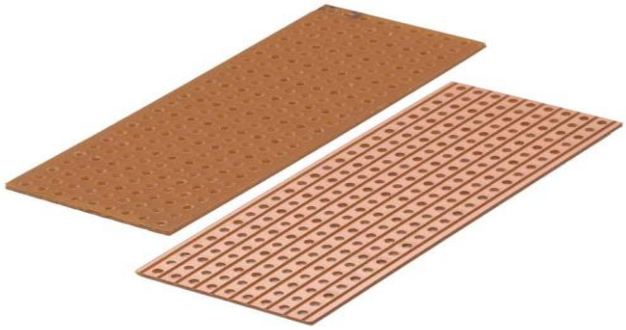

Fig 1.7: Vero Board

II. Design & Fabrication

A block diagram and a circuit diagram have been designed. Then we implemented according to diagrams.

2. 1. Block Diagram of the System

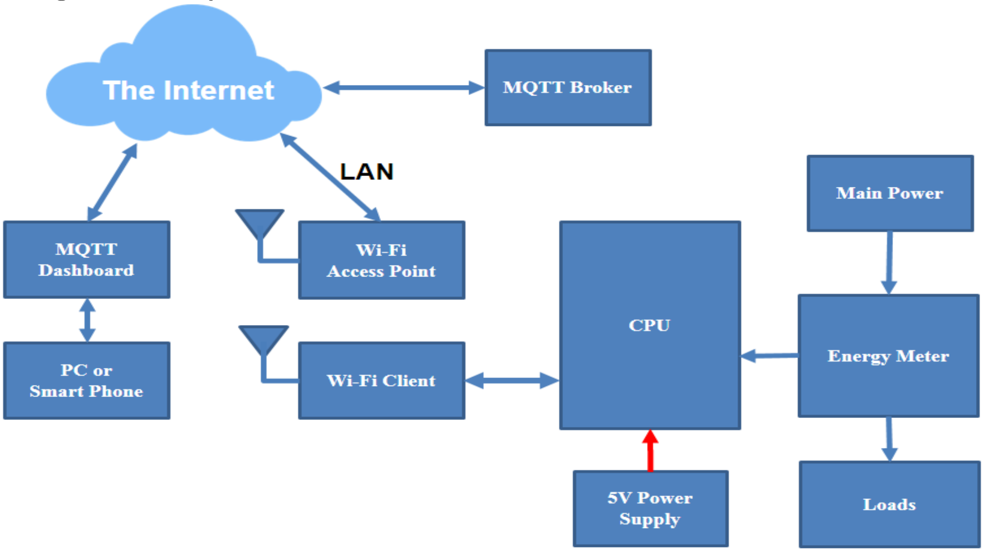

Fig 2.1: Block diagram of the system.

1.3. Designing Circuit

The circuit diagram of the project is designed by Easy EDA online circuit design tool.

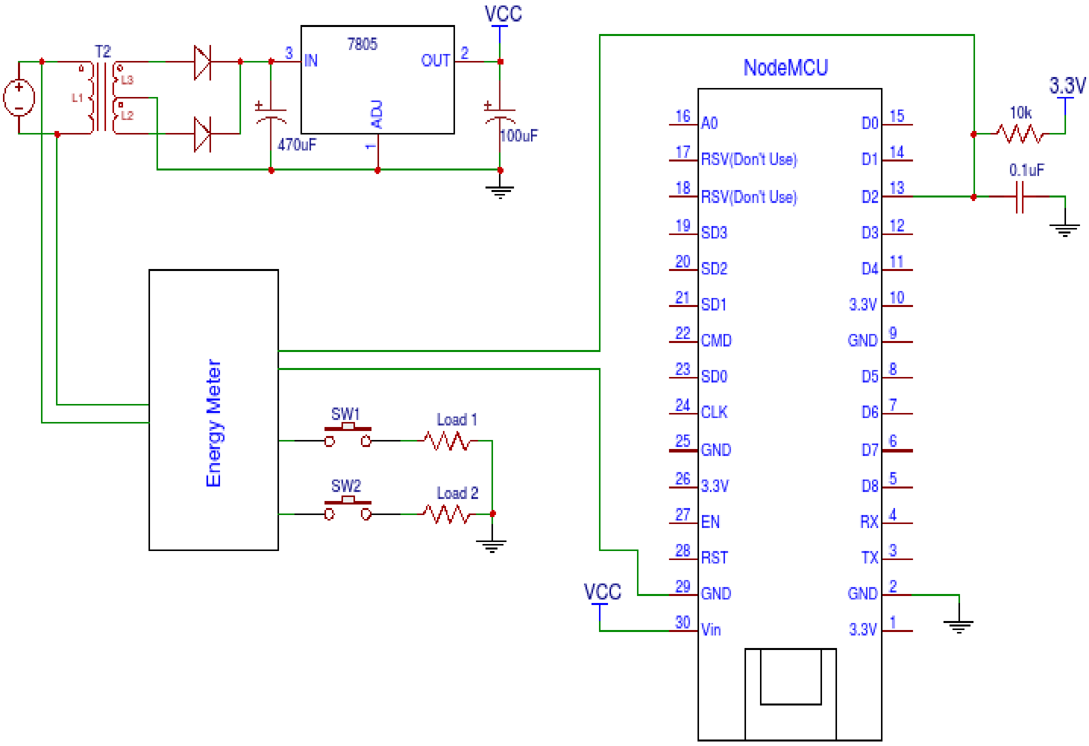

Fig 2.2: Complete Circuit diagram of the system.

2. 3. 1. Circuit Description

The complete project is built with NodeMCU 12E (ESP8266) module, electrical energy meter, connectors, loads & power supply. The microcontroller ESP8266 acts as a CPU of the system. It needs 5v to power-up the NodeMCU board. Constant 5V from power supply is connected to V_{in} pin of NodeMCU board. Loads are connected directly to output of energy meter via switches. The supply voltage of the loads is 220V AC.

2. 3. 2. Circuit Board Design

The circuit board is designed on Vero Board.

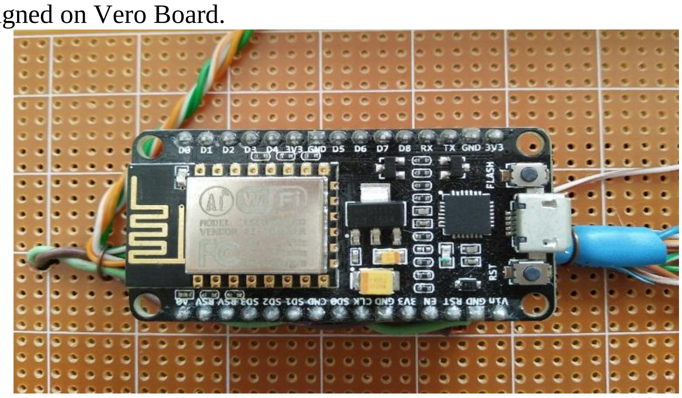

Fig 2.3: Circuit on Vero Board.

1.4. Working Principle

The energy meter measures the energy consumption of an electrical load. The energy meter used in this particular project has a pulse output pin. It gives 1600 pulses per kWh. One of the CPU pins is connected to the pulse output of the energy meter. When the pulse arrives at the CPU's input pin, it counts the number of pulses, saves them in the EEPROM and starts the timer. When another impulse happens, stops timer and calculate amount of load. This system connects to Wi-Fi and sends electricity bills to electricity suppliers and users. The system also connects to the broker MQTT and sends invoices and loads for monitoring in real time. MQTT is an easy publishing and subscription system that allows anyone to publish and receive messages as customers. It is very easy to establish communication between several devices. The ESP8266 Wi-Fi modules and the embedded microcontroller connect to the MQTT broker via the Internet. If the system is in manual mode, the smartphone or PC browser connects to the MQTT broker via the Internet using the MQTT toolbar.

1.5. Microcontroller

The microcontroller executes the program loaded into the flash memory. This is the so-called executable code, which is a seemingly meaningless sequence of 0 and 1. It consists of 12, 14 or 16 bits of wide words depending on the architecture of the microcontroller. All words are treated by the CPU as commands that must be executed while the microcontroller is running [**6**].

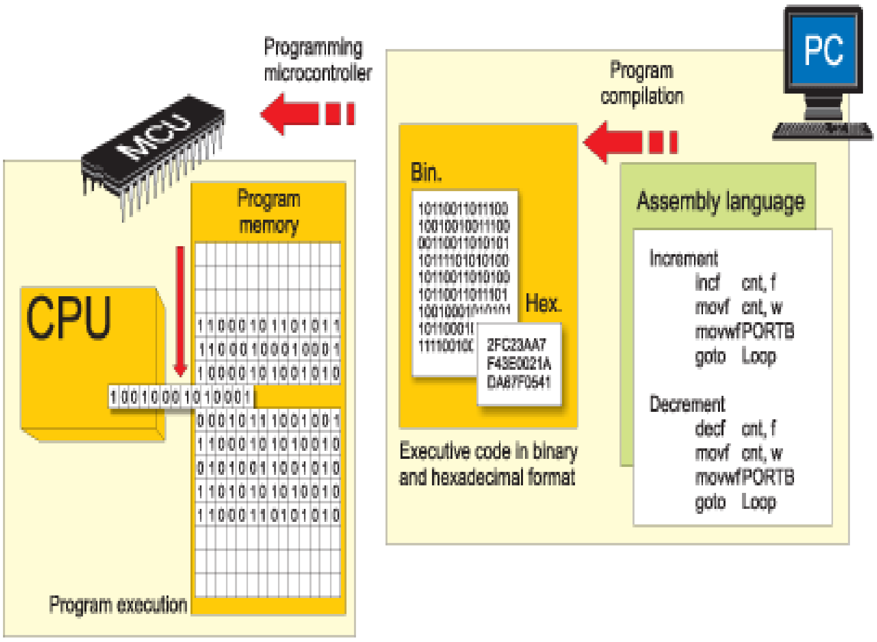

Fig 2.4: Compiling Program [**6**]

1.6. Programming using IDE

To write our code, use the Arduino IDE (Integrated Development Environment). Using the Arduino open source software (IDE), you can easily write the code and upload it to the whiteboard. This software can be used on the Arduino board. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born in the Ivrea Interaction Design Institute as a simple tool for rapid prototyping for students who do not have experience in electronics and programming. All Arduino boards are completely open source, users can build themselves and eventually can adapt to specific needs. The software is also open source and grows thanks to the contribution of users around the world. [**7**]

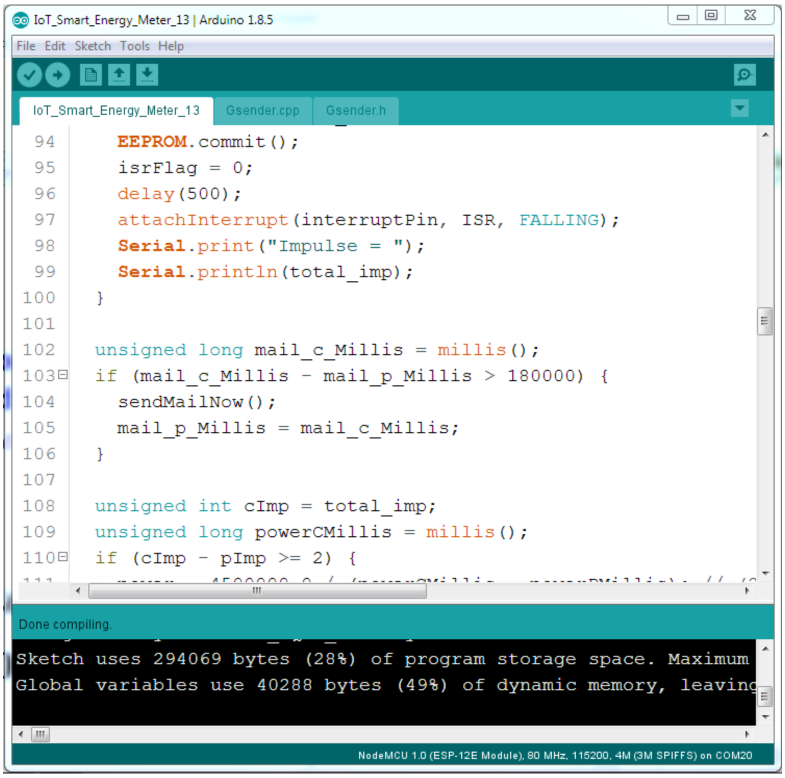

Fig 2.5: Compilation of Program Using Arduino IDE

2.7. Microcontroller Programmer / Program Burner

A microcontroller programmer or microcontroller burner is a hardware device accompanied with software which is used to transfer the machine language code to the microcontroller/EEPROM from the PC.

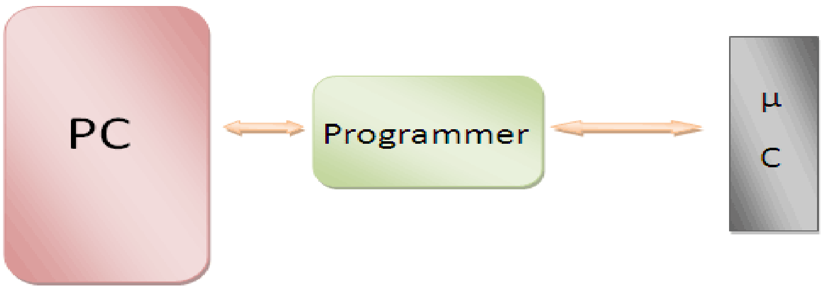

Fig 2.6: Download program to MCU

The compiler converts codes written in languages such as assembly language, C language and Java language, into a machine language that the computer / microcontroller can understand and store in a hex file. The microcontroller programmer functions as an interface between the PC and the target controller. The programmer's API / program reads the data from the hex file stored on the PC and sends it to the controller memory. The target controller, which must write the program, is placed in the programmer using a ZIP socket. The software transfers data from the PC to the hardware using a serial, parallel or USB port [9]

Depending on the way it interacts with PC, there are two types of microcontroller programmers:

- Parallel Programmer uses the parallel port of the PC. They are low cost programmer but not widely used. Serial Programmers uses the serial port to interact with PC via RS232 protocols. They are more popular among hobbyist working on PC. However both the serial and parallel programmers will become obsolete in near future. The major reason being unavailability of parallel and serial ports in the PCs & Laptops in the coming years.
- USB Programmer uses the USB interface to transfer the data from PC. The main advantage of the USB burner is that they are powered from the PC itself and there is no need of any additional supply. The USB programmers have already become popular and will soon replace the serial and parallel programmer.

Fig **2.7**: USB to TTL Converter

The conventional method to burn a controller is to take it out the circuit, place it on burner and then dump the hex file into the controller using the API. In order to remove this problem of removing the controller from the circuit every time it needs to be programmed, the controllers have now been upgraded with In System Programmer (ISP) feature. This allows burning/programming a controller without removing the controller from the circuit it is used in. The latest controllers are coming with the feature like bootloader memory which allows self-burning capabilities, i.e. such microcontroller controller does not need any additional programmer hardware. They need only an API to transfer the program to the target controller. This API can also be incorporated in the compiler and hence the compiler can directly burn the target controllers. [**8**]

2.8 Image of the system

The complete is built with NodeMCU 12E (ESP8266) module, Electrical Energy Meter, switches, loads & power supply. The microcontroller ESP8266 is the CPU of the system. It needs 5v to power-up. Constant 5V from power supply is connected to VIN pin of NodeMCU board. Loads are connected to relay module. The supply voltage of the loads is 220V AC. Image of the project is shown on Fig **2.7** below.

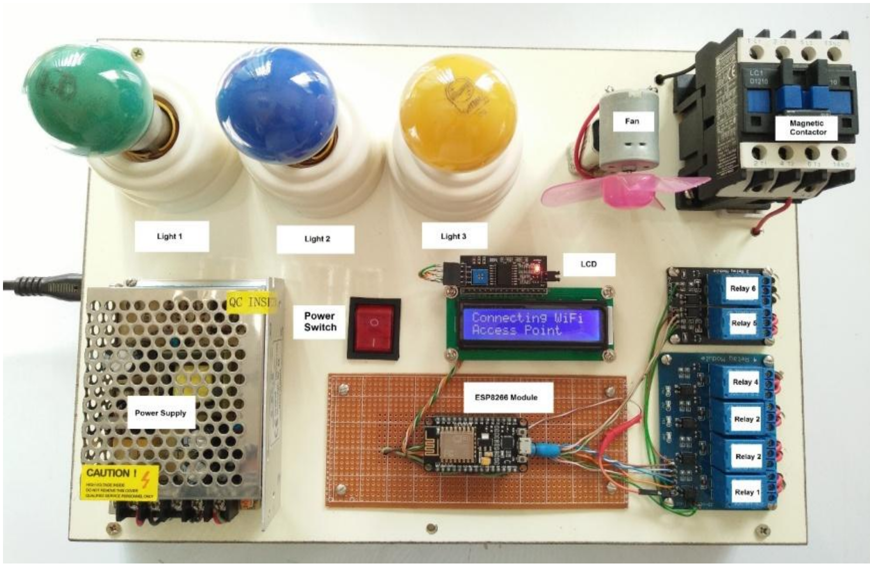

Fig 2.7: Image of the Project

III. Result

The system is implemented using required parts in a wooden box and an Avonite board as shown in **Fig 2.7.** To test the system, a smart phone is used as Wi-Fi hotspot and 220V AC supplied to the system. After turning ON, the system connected itself to Wi-Fi hotspot quickly. Then it connects itself to MQTT broker server and sends calculated bill and load. It also sends email to the user and to electric supply company. On the user side, a smart phone is connected to MQTT broker server via internet connection. An android application named “MQTT Dashboard” is used in order to connect to MQTT broker server. On MQTT Dashboard, user can view bill and load in real time. All over, the system has been run as desired and it is showed satisfactory result [shown in the **Fig 2.8 (a), Fig 2.8 (b)**].

Fig 2.8 (a): Image of the electricity Consumption bill via email

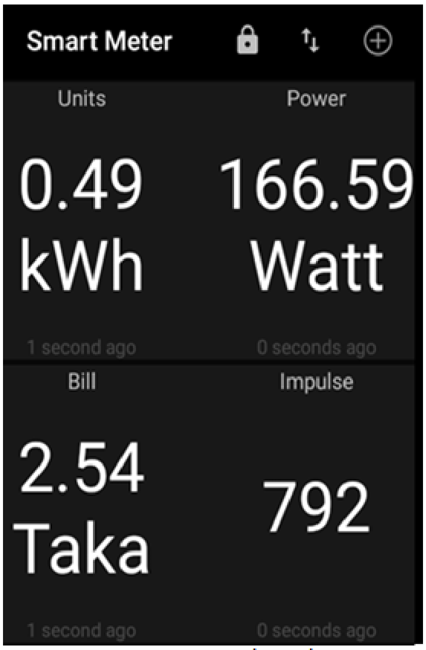

Fig 2.8 (b): Image shows how can energy consume by load

IV. Conclusion

The objective to build an IoT Based Automatic Electrical Energy Meter Billing System was successfully achieved. Embedded systems are emerging as a technology with high potential. In the past few decades, microprocessor embedded systems have dominated the market. Over the past ten years, I've seen a revolution in embedded systems based on microcontrollers. This system basically accurately measures the electricity bills and sends e-mails using a microcontroller and an internet connection. Regarding the collected requirements, the complexity of management and counting can be achieved with the help of electronic devices. This system provides an effective watt-hour recording system in comparison with traditional methods in terms of productivity and efficiency.

Acknowledgements

All praise to “Almighty Allah” the most merciful, the most gracious, the source of knowledge and wisdom endowed to mankind, who conferred us with the power, mind and capability to take this study of exciting ocean of knowledge.

We are grateful to Engr. M. Azharul Haque, Chairman, Department of EEE, Bangladesh University of Business and Technology for giving us the opportunity to use the facilities of EEE Department to made our system and for all his advices and suggestions.

References

[1]. International Journal of Innovative Research in Computer and Communication Engineering (Vol. 5, Issue 4, April 2017)
[2]. https://espressif.com/en/products/hardware/esp8266ex/overview
[3]. ESP 8266 Datasheet
[4]. https://en.wikipedia.org/wiki/Internet_of_things
[5]. https://www.postscapes.com/internet-of-things-history
[6]. http://www.microcontroller-project.com/16x2-lcd-working.html
[7]. https://www.engineersgarage.com/tutorials/microcontroller-programmer-burner
[8]. A T89Sxx ISP PROGRAMMER by Frontline Electronics, Pvt. Ltd., Salem.

Shahajan Miah" Internet of Things (IoT) based automatic electrical energy meter billing system" IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) 14.4 (2019): 39-50.