RESEARCH ARTICLE OPEN ACCESS

Smart Power Calculator Over IOT

Vennala Hemalatha [1], Guntupu Sowjanya [2], Md Shair Ali [3]
Sathaka Mounika [4], P. Kavya [5]
Department of Computer Science & Engineering
Nadimpalli Satyanarayana Raju Institute Of Technology
Andhra Pradesh –India

ABSTRACT

The project is to design a power calculator for calculating power using IOT. This can aid in cutting energy consumption as the owner is continuously notified about the no. of units is generate bill automatically checking the consumption of the house and way to reduce manual task. The calculations performed automatically, the invoice is updated on the net by using IOT. The proposed scheme is applied to compute the force. The info about the bill updated automatically and displayed on the LCD and phone. The microcontroller in the organization is responsible to hold the associated peripherals in it. If any power theft takes place, it automatically transmits a notification to the phone and turns off the force. It is cost efficient and compact, so installing is. The result is uploaded to the phone.

Keywords:- *IOT*

I. INTRODUCTION

Typically, IOT is expected to offer advanced connectivity of devices, systems and services that goes beyond machine-to-machine communication and embraces a variety of protocols, knowledge bases and applications. The interconnection of these implanted devices (including smart objects) is expected to usher in automation in virtually all areas, while also enabling advanced applications like a Smart Grid, and expanding to area such as smart cities.

The Internet of Things (IOT) refers to the ever-growing network of physical objects that feature an IP address for the internet connected, and the communication that takes place between these targets and the other internet-enabled devices and system.

If we had computer that know everything there was to know about things using the data they gathered without any assistance from us—we would be able to cut through and consider everything, and greatly reduce waste, loss and monetary value. We would experience when things needed replacing or returning, and whether they were fresh or past their best. We require to empower computer with their own way of collecting information and then they can see her and small the world for themselves, in all its random glory. Sensor technology enables computers to observe, identify and understand the Earth without the restriction of human entered data.From a broad perspective the confluence of several technologies and securities industry trends is making it possible to interconnect more and smaller devices cheaply and well

II. PHASE OF THE IOT LIFECYCLE

- **Five phases of IOT life cycle**

Firstly ,create phase where devices or sensors collect information from the physical environment around them .the data from smart connected devices can be used to generate insights that can help business, customers and partners.

Secondly, communication phase where the data and results brought forth are transmitted through the mesh to the desired address.

Thirdly, aggregation phase is where the data collected is integrated by the twist itself.

Fourthly, analyze phase where upon further sophisticated analytics the aggregated data can be utilized to generate basic patterns, control and optimize operation.

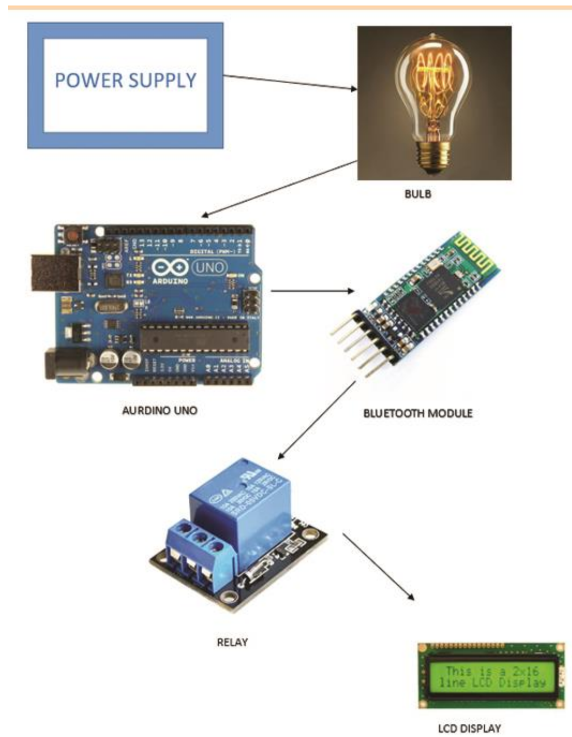
In conclusion, active phase, where suitable actions are done based on the information produced.

III. SCOPE OF IOT

Net of Things can connect devices embedded in several systems to the cyberspace. When devices/objects can represent themselves digitally, they can be controlled from anywhere. The connectivity, then helps us capture more information from more places, ensuring more ways of increasing efficiency and improving safety and IOT security. IOT is transformational forces that can help campaniles improve performance through IOT analytics and IOT Security to deliver better outcomes. Lines in the utilities, oil & gas, insurance, manufacturing, transportation, infrastructure and retail sectors can reap the benefits of IOT by making more informed decisions, aided by the torrent of instructional and transaction information at their disposition. It is expected that IOT products with interoperable capability will dominate the marketplace. Awareness of IOT products is too vital for market penetration along with protection features. Even very few Americans are cognizant of the use of these wares. As per a study of the Consumer Electronics Association and Parks Association found only 10% of the household in automatic door locks, Wi-Fi connected ceiling fans, light switches, LED bulbs, smart watches, 3-D printers and smart clothes will be popular among consumers.

IOT is proved to be an emerging technology innovation. In the current context, it is now possible that a helmet of a two wheeler can interact with a car for avoiding collision. Connected toothbrush can now supervise and make one’s experience pleasurable. A three dimensional sensor of the electric brush can connect with Smartphone apps and offer real time feedback to the individual.

IV. SYSTEM ARCHITECTURE



V. SRS (SOFTWARE REQUIREMENT SPECIFICATION)

A software requirements specification (SRS) is a description of a software system to be built up. It lays out functional and non-functional requirements, and may include a set of usage cases that describe user interactions that the software must provide. In parliamentary law to fully understand one's project, it is really important that they come up with an SRS listing out their requirements, how are they proceeding to play it and how will they finish the task. It assists the team to save upon their time as they are capable to comprehend how are going to go about the task. Doing this also enables the team to find out about the limitations and risks early on.

- **FUNTIONAL REQUIREMENTS**

A functional requirements document defines the functionality of a system or one of its subsystems. It likewise depends upon the character of software, expected users and the type of system where the software is applied. Functional user requirements may be high-level statements of what the scheme should do, but functional system requirements should also describe clearly about the system services in particular.

- **NON-FUNCTIONAL REQUIREMENTS**

A non-functional requirement is a requirement that defines standards that can be employed to judge the performance of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behaviour or functions. The design for implementing functional requirements is detailed in the scheme design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

Generally, functional requirements determine what a system is supposed to perform and non-functional requirements determine how a system is said to be. Functional requirements are usually in the form of "system shall do <requirement>", an individual action or part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of "system shall be <requirement>", an overall property of the system as a whole or of a particular aspect and not a specific function. The system's overall properties commonly mark the difference between whether the development project has succeeded or failed. Non-functional requirements are often called "quality attributes" of a system.

Other terms for non-functional requirements are "qualities", "quality goals", "quality of service requirements", "constraints", "non-behavioural requirements", or"technicalrequirements". Informally these are sometimes called the "ilities", from attributes like stability and portability. Qualities—that is non-functional requirements—can be divided into two main categories

- Execution qualities, such as safety, security and usability, which are observable during operation (at run time).
- Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the system.

VI. SYSTEM REQUIREMENTS

System requirements refer to the hardware and software components of a computer system that are required to install and use software efficiently. The software manufacturer will list the system requirements on the software bundle. If your computer system does not match the system requirements, then the software may not function correctly after installation. Scheme requirements for operating systems will be hardware components, while other application software will list both hardware and operating system essentials. System requirements are most commonly seen listed as *minimum* and *recommended* requirements. The minimum system requirements need to be taken on for the software to work at all on your system, and the recommended system requirements, if satisfied, will offer better software usability. Both hardware and software tools have been employed in this task.

VII. SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements and requirements that need to be set up on a computer to provide optimal functionality of an application. These requirements or prerequisites are generally not included in the software installation paCkage and need to be set up separately before the software is set up

- **Arduino IDE**

This is the Arduino IDE once it's been spread out. It unfolds into a blank sketch where you can begin programming right away. Foremost, we should configure the circuit card and port settings to grant us to upload code. Plug into your Arduino board to the PC via the USB cable

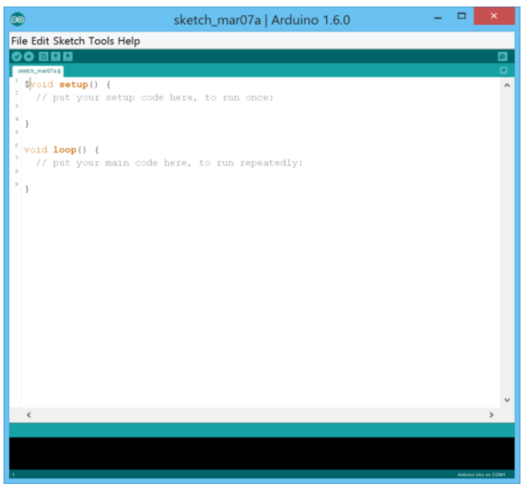


Fig 1: Arduino IDE Default Window

- **IDE: Board Setup**

You hold to tell the Arduino IDE what board you are uploading to. Choose the Tools pull down menu and go to the Board. This list is populated by default with the currently available Arduino Boards that are prepared by the Arduino. If you are using an UNO or an UNO-Compatible Clone (ex. Funduino, SainSmart, IEIK, etc.), select Arduino Uno. If you are using another board/clone, select that board.

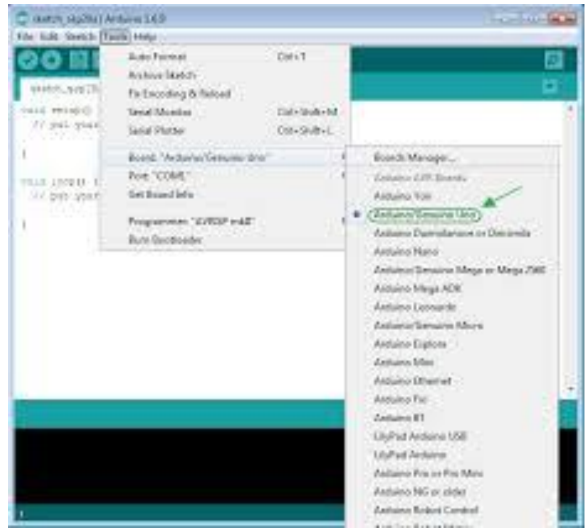


Fig 2: Board Setup Procedure

- **IDE: COM Port Setup**

If you downloaded the Arduino IDE before plugging in your Arduino board, when you plugged into the board, the USB drivers should have installed automatically. The most recent Arduino IDE should recognize connected boards and label them with which COM port they are using. Select the Tools pull down menu and then Port. Here it should list all open COM ports, and if there is a recognized Arduino Board, it will also give its name. Select the Arduino board that you have connected to the PC. If the setup was successful, in the bottom right of the Arduino IDE, you should see the board type and COM number on the board you plan to program. Note: the Arduino UNO occupies the next available COM port; it will not always be COM3.

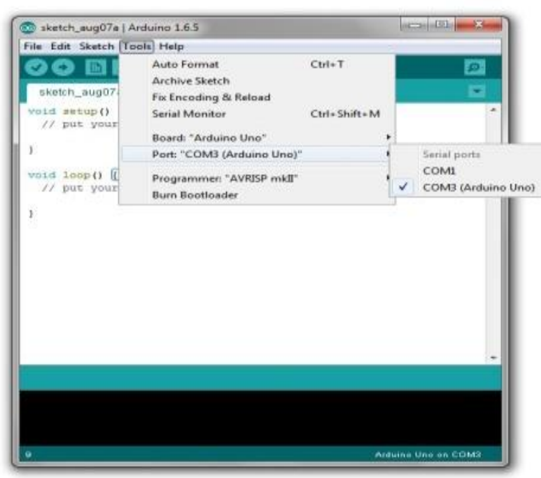


Fig 3: COM Port Setup

At this point, your board should set up for programming, and you can begin writing and uploading code.

- **Testing Your Settings: Uploading Blink**

One common procedure to test whether the board you are using is properly set up is to upload the "Blink" sketch. This sketch is included with all Arduino IDE releases and can be accessed from the File pull-down menu and going to Examples, 01. Basics, and then select Blink. Standard Arduino Boards include a surface-mounted LED labelled "L" or "LED" next to the "RX" and "TX" LEDs that is connected to digital pin 13. This sketch will blink the LED at a regular interval, and is an easy way to confirm if your board is set up properly and you were successful in uploading code. Open the "Blink" sketch and press the "Upload" button in the upper-left corner to upload "Blink" to the board.

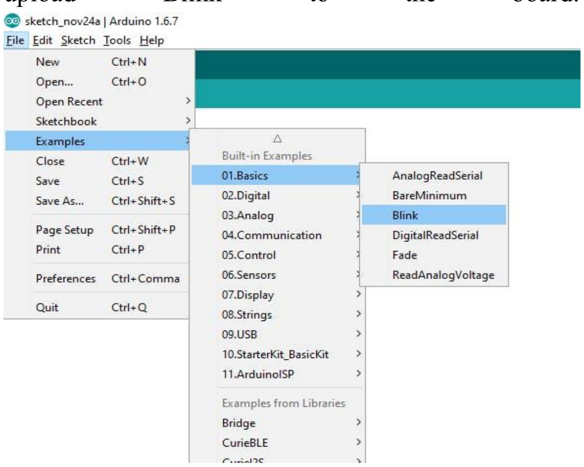


Fig 4: Loading Blink Sketch

Fig 5: Uploading Blink

VIII. HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

- **Arduino UNO**

The Arduino UNO is a microcontroller board based on the ATmega328 (datasheet). It holds 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connector, a power jack, an ICSP header, and a reset button. It comprises everything needed to hold up the microcontroller; simply tie it to a data processor with a USB cable or power it with an AC-to-DC adapter or battery to obtain gone. The UNO differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial convertor.

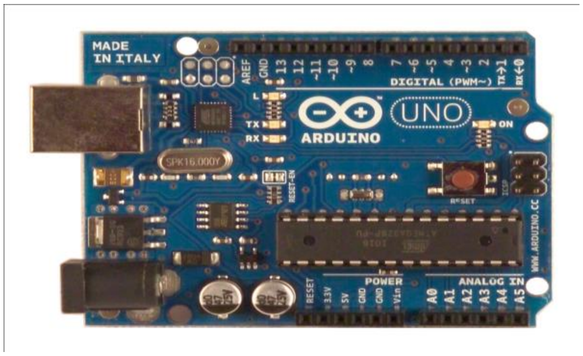


Fig 6: Arduino UNO

- **Relay**

A relay is an electrically operated switch. It consists of a lot of input terminals for a single or multiple control signals, and a set of operating contact terminals. The shift may have whatever number of contacts in multiple contact forms, such as make contacts, break contacts, or combinations thereof.

Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal. Relays were first used in long-distance telegraph circuits as signal repeaters: they refresh the signal coming in from one circuit by transmitting it on another lap. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

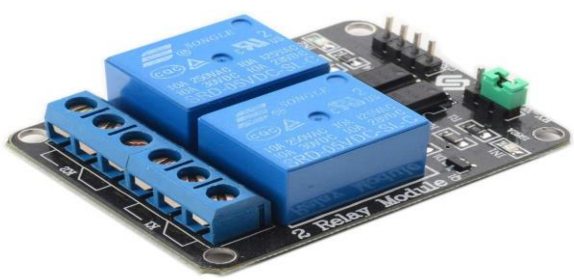


Fig 7:5V Relay module

The traditional form of a relay uses an electromagnet to close or open the contacts, but other operating principles have been invented, such as in solid-state relays which use semiconductor properties for control without relying on moving parts. Relays with calibrated operating characteristics and sometimes multiple operating coils are applied to protect electrical circuits from overload or faults; in innovative electric power systems these functions are executed by digital instruments still called protective relays.

Latching relays require only a single pulse of control power to operate the switch persistently. Another pulse applied to a second set of control terminals, or a pulse with opposite polarity, resets the switch, while repeating pulses of the same sort have no effects. Magnetic latching relays are useful in applications when interrupted power should not bear upon the circuits that the relay is controlling.

- **LCD 16×2**

LCD modules are very commonly practiced in most embedded projects, the reason being its cheap cost, availability and programmer friendly. Most of us would have come across these displays in our day to day life, either at PCO's or calculators. The appearance and the spots have already been visualized above now let us acquire a bit technical. **16×2 LCD is made so because; it has 16 Columns and 2 courses.** On that point are a set of combinations available like, 8×1, 8×2, 10×2, 16×1, etc., but the most used one is the 16×2 LCD. Thus, it will have (16×2=32) 32 characters in total and each case will be made of 5×8 Pixel Dots.

Today, we recognize that each character has (5×8=40) 40 Pixels and for 32 Characters we will have (32×40) 1280 Pixels. Farther, the LCD should also be taught about the Position of the picture elements. Hence it will be a hectic task to cover everything with the help of MCU, hence an Interface IC like HD44780is used, which is waxed on the rear of the LCD Module itself. The role of this IC is to get the Commands and Data from the MCU and process them to display meaningful information onto our LCD Screen. You can determine how to interface an LCD using the above mentioned links. If you are an innovative programmer and would like to make your own library for interfacing your Microcontroller with this LCD module then you accept to understand the HD44780 IC is working and commands which can be found its datasheet.

- 2D model of 16×2 LCD module

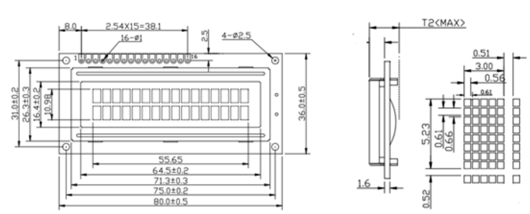


Fig 8:16x2 LCD Datasheet

- **Features of 16×2 LCD module**
  - Operating Voltage is 4.7V to 5.3V
  - Current consumption is 1mA without backlight
  - Alphanumeric LCD display module, meaning can display alphabets and numbers
  - Consists of two rows and each course can print 16 characters.
  - Each part is established by a 5×8 pixel box
  - Can work on both 8-bit and 4-bit mode
  - It can also display any custom generated characters
  - Available in Green and Blue Backlight

- **HC-05 Bluetooth Module**
  - It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications.
  - It accepts a range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.
  - It is IEEE 802.15.1 standardized protocol, through which one can make a wireless Personal Area Network (PAN). It uses frequency-hopping spread spectrum (FHSS) radio technology to send data over the air.
  - It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).
  - HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.

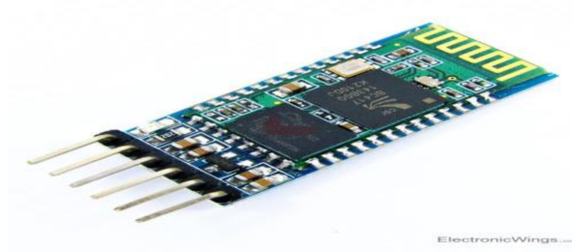


Fig 9:HC-05 Bluetooth Module

Fig 10: Pin Description

- **MAX 232**

The MAX232 is an integrated circuit first created in 1987 by Maxim Integrated Products that converts signals from a TIA-232 (RS-232) serial port to signals suitable for use in TTL-compatible digital logic circuits. The MAX232 is a dual transmitter / dual receiver that typically are used to convert the RX, TX, CTS, and RTS signals. The drivers provide TIA-232 voltage level outputs (about ±7.5 volts) from a single 5-volt supply by on-chip charge pumps and external capacitors. This makes it useful for implementing TIA-232 in devices that otherwise does not need any other voltages. The receivers reduce TIA-232 inputs, which may be as high as ±25 volts, to standard 5 volt TTL levels. These receivers have a typical threshold of 1.3 volts and a typical hysteresis of 0.5 volts. The MAX232 replaced an older pair of chips MC1488 and MC1489 that performed similar RS-232 translation. The MC1488 quad transmitter chip required 12 volt and -12 volt power,[1] and MC1489 quad receiver chip required 5 volt power.[2] The main disadvantages of this older solution was the +/- 12 volt power requirement, only supported 5 volt digital logic, and two chips instead of one.

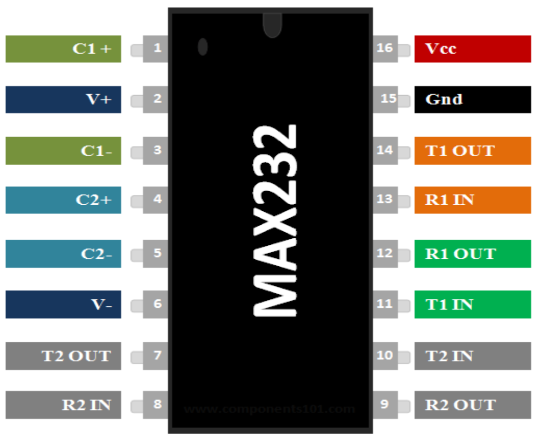


Fig 11: MAX 232

- **Micro-Controller**

A microcontroller (MCU for microcontroller unit) is a small computer on a single metal-oxide-semiconductor (MOS) integrated circuit chip. In modern nomenclature, it is similar to, but less sophisticated than, a system on a chip (SoC); a SoC may include a microcontroller as one of its ingredients. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. Program memory in the form of ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications consisting of various discrete chips.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. In the context of the internet of things, microcontrollers are an economical and popular means of data collection, sensing and actuating the physical world as edge devices.

Some microcontrollers may use four-bit words and operate at frequencies as low as 4 kHz, for low power consumption (single-digit mill-watts or microwatts). They generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nano-watts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

Fig 12: Micro-Controller

IX. SCREEN SHORTS

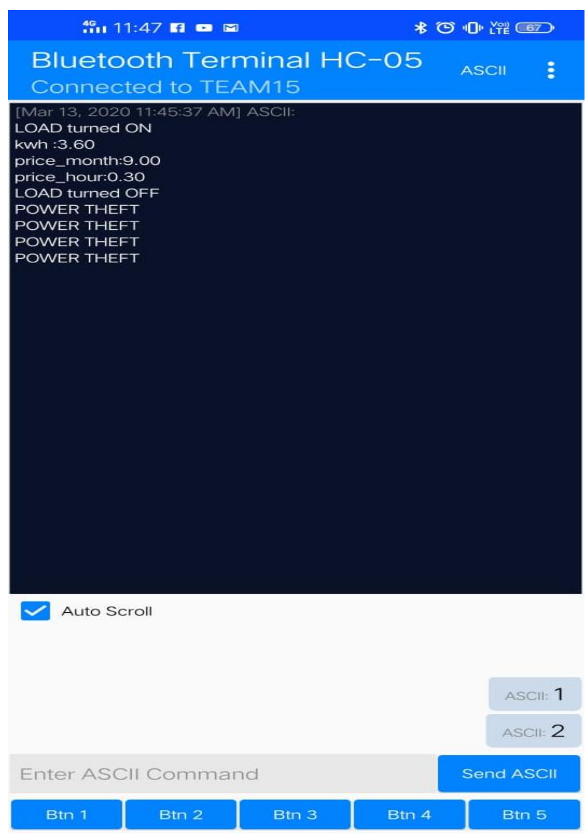


Fig 13: Power Theft

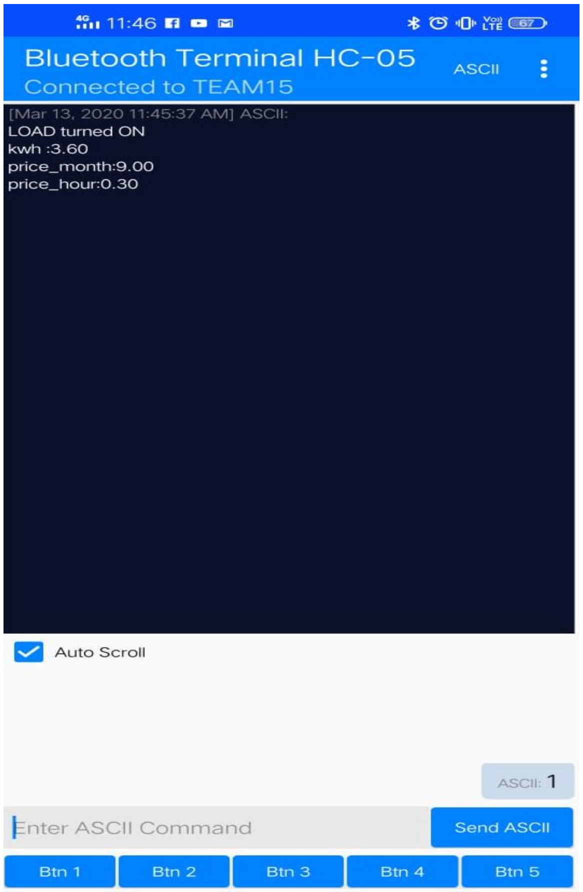


Fig 13: Load Turned ON

X. CONCLUSION

This smart power calculation ising IOT uses Aurdino UNO for micro-controllers and Bluetooth module for the purpose of flexibility of customer to moniter their bill they consumed for their purpose is notified using mobile phone.The results showed that system works successfully.Further research for controlling meter instead of just monitoring usage of power,controlling them will be one step further.So that users can ever control their bill, usage power consumption by themselves remotely from mobile phones.

REFERENCES

[1] Kamal Nayan Reddy, Challa, Venkata Sasank Pagolu and Ganapati Panda, “An Improved Approach for Prediction of Parkinson’s Disease using Machine Learning Techniques”, in Procedings of the International conference on Signal Processing, Communication, Power and Embedded System (SCOPES)-2016, pp. 1446-145, 2016.

[2] Geeta Yadav, Yugal Kumar and G. Sahoo, “Predication of Parkinson’s disease using Data Mining Methods: a comparative analysis of tree, statistical and support vector machine classifiers”, in Procedings of the National Conference on Computing and Communication Systems (NCCCS), pp. 1-4, 2012.

[3] Paolo Bonato, Delsey M. Sherrill, David G. Standaert, Sara S. Salles and Metin Akay, “Data Mining Techniques to Detect Motor Fluctuations in Parkinson's Disease”, in Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4766-4769, 2004.

[4] Sonu S. R., Vivek Prakash and Ravi Ranjan, “Prediction of Parkinson’s Disease using Data Mining”, in Proceedings of the International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), pp. 1082-1085, 2017.

[5] Aarushi Agarwal, Spriha Chandrayan and Sitanshu S Sahu, “Prediction of Parkinson’s Disease using Speech Signal with Extreme Learning Machine”, in Proceedings of the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 1-4, 2016.

[6] Akshaya Dinesh and Jennifer He, “Using Machine Learning to Diagnose Parkinson’s Disease from Voice Recording”, in Proceedings of the IEEE MIT Undergraduate Research Technology Conference (URTC), pp. 1-4, 2017.

[7] Giulia Fiscon, Emanuel Weitschek, Giovanni Felici and Paola Bertolazzi, “Alzheimer’s disease patients classification through EEG signals processing”, in Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM). pp 1-4, 2014.

[8] Pedro Miguel Rodrigues, Diamantino Freitas and Joao Paulo Teixeirab, “Alzheimer electroencephalogram temporal events detection by K-means”, in Proceedings of the International

Conference on Health and Social Care Information Systems and Technologies HCIST. pp. 859 – 864, 2012.

[9] Daniel Johnstone1, Elizabeth A. Milward1, Regina Berretta1 and Pablo Moscato1, "Multivariate Protein Signatures of Pre-Clinical Alzheimer's Disease in the Alzheimer's Disease Neuroimaging Initiative (ADNI) Plasma Proteome Dataset", in Proceedings of the Disease Neuroimaging Initiative, vol-7, pp. 1-17, 2017.

AUTHOR DETAILS

Vennala Hemalatha is presently pursuing B.Tech (CSE) Department of Computer Science Engineering from Nadimpalli satyanarayana raju institute of technology.

Guntupu Sowjanya is presently pursuing B.Tech (CSE) Department of Computer Science Engineering from Nadimpalli satyanarayana raju institute of technology.

Md Shair Ali is presently pursuing B.Tech (CSE) Department of Computer Science Engineering from Nadimpalli satyanarayana raju institute of technology.

Sathaka Mounika is presently pursuing B.Tech (CSE) Department of Computer Science Engineering from Nadimpalli satyanarayana raju institute of technology.

Mrs. P. Kavya M.Tech is working as an Assistant Professor in the Department Of Computer Science & Engineering in NS Raju Institute Of Technology.