**SMART ELECTRICITY MONITORING**

**SYSTEM (SEMS)**

**A PROJECT REPORT**

*Submitted By*

**SHIVARANJANI.T**

Reg. No. 412513205069

**SUBATHRA.P**

Reg. No. 412513205079

**VAISHALI.V**

Reg. No. 412513205090

***In partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**SRI SAIRAM ENGINEERING COLLEGE, CHENNAI**

**ANNA UNIVERSITY: CHENNAI - 600 025**

APRIL 2017

**BONAFIDE CERTIFICATE**

Certified that this project report titled **“SMART ELECTRICITY MONITORING SYSTEM (SEMS)”** is the bonafide work of **“SHIVARANJANI.T (412513205069)”,** who carried out the project work under my supervision. Certified further, that to the best of my knowledge that the work reported here in does not form part of any other thesis on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

|  |  |
| --- | --- |
| **HEAD OF THE DEPARTMENT** | **INTERNAL GUIDE** |
| **Dr. T.SHEELA. B.E., M.S., PhD**  **DEAN [NETWORK] & HEAD**  Department of Information Technology  Sri Sairam Engineering College  Chennai- 600 044 | **MS. M.ANANTHI. B.E., M.E., PhD**  **ASSOCIATE PROFESSOR**  Department of Information Technology  Sri Sairam Engineering College  Chennai- 600 044 |

Submitted for University Project Examination held on ……………. at Sri Sairam Engineering College, Chennai-600 044.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We express our gratitude to our beloved chairman **Late** **Thiru. MJF. Ln. LEO MUTHU** for the help and advice he has shared upon us.

We express our gratitude to our CEO Mr**.J. SAI PRAKASH LEO MUTHU** and our trustee **Ms. J. SHARMILA RAJAA** for their constant encouragement in completing this project.

We express our solemn thanks to our Principal **Dr.C.V. JAYAKUMAR** for having given us spontaneous and whole hearted encouragement for completing this project.

We are indebted to our Head of the Department **Dr. T. SHEELA** for her constant support during the entire course of this project work.

We express our gratitude and sincere thanks to our guide **Ms. M.ANANTHI,** AssociateProfessor for her valuable suggestions and constant encouragement for successful completion of this project.

Our sincere thanks to our project coordinator **Mrs. S.GOMATHI** for her kind support in bringing out this project successfully.

Finally, we thank all the staff members of the **Department of Information Technology** and all others who contributed directly or indirectly for the successful completion of our project.

We thank our almighty for his blessings to make this project a success.

**ABSTRACT**

Electricity is taking lion’s share of everybody’s income in the present scenario, and this creates a greater necessity for controlling the consumption of electricity right from our homes to organizations. This system is primarily aimed at having greater control over the electricity consumed in household and consequently reducing the consumption. This system is intended to indicate the power consumption of different devices from time to time which overcomes the main difficulty faced in the existing system, where the indication about the consumption is given only at the end of a long tenure. The proposed system computes the power consumed by various devices using an integrated embedded kit and analyzes the cumulative consumption which will be sent as an alert SMS to the user if it overruns the threshold. This system also senses the natural light and temperature periodically and if the sensed parameters are beyond the threshold, or the threshold is the not met, the light/ fan is switched OFF/ON automatically and a SMS is sent to the user stating that the required light intensity / temperature is available/ not available for human, and hence the light/fan is switched OFF/ON. This lets the user extend control over the power consumption of his household and reduce the consumption in case of overrunning, which consequently saves electricity which can be used by the future generations, giving them an energy efficient environment. Besides it further improves the economy of individuals following the economic improvement of the society.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | **ACKNOWLEDGEMENT** | **iii** |
|  | **ABSTRACT** | **iv** |
|  | **TABLE OF CONTENTS** | **V** |
|  | **LIST OF FIGURES** | **Viii** |
|  | **LIST OF ABBREVIATIONS** | **Viii** |
| **I** | **INTRODUCTION** | **9** |
|  | 1.1 AIM AND OBJECTIVE | **9** |
|  | 1.1.1.ELECTRICITY MONITORING AND CONTROL  1.1.2 TEMPERATURE AND LIGHT SENSING AND CONTROL | **10** |
|  | 1.2 SCOPE OF THE PROJECT | **17** |
| **II** | **LITERATURE SURVEY** | **18** |
|  | 2.1 HOUSEHOLD ELECTRICITY CONSUMPTION ANALYSIS WITH DATA MINING TECHNIQUES-2016  2.2 ELECTRICITY CONSUMPTION ANALYSIS AND POWER QUALITY MONITORING IN COMMERCIAL BUILDINGS-2015  2.3 ELECTRICITY-METERING IN A CONNECTED WORLD:VIRTUAL SENSORS FOR ESTIMATING THE ELECTRICITY CONSUMPTION OF IoT APPLIANCES -2015  2.4 ANALYSIS OF THE RESIDENTIAL, COMMERCIAL AND INDUSTRIAL ELECTRICITY CONSUMPTION-2015  2.5 AN INTELLIGENT ENERGY MANAGEMENT SCHEME WITH MONITORING AND SCHEDULING APPROACH FOR IoT APPLICATIONS IN SMART HOME-2015 | **19** |
| **III** | **SYSTEM ANALYSIS** | **21** |
|  | 3.1 EXISTING SYSTEM | **21** |
|  | 3.1.1 TNEB METERS TARIFF | **21** |
|  | 3.2 PROPOSED SYSTEM | **21** |
|  | 3.2.1 LIGHT SENSOR | **22** |
|  | 3.2.2 TEMPERATURE SENSOR | **22** |
|  | 3.2.3 GSM MODEM FOR MESSAGE COMMUNICATION | **23** |
|  | 3.2.4 HALL EFFECT SENSORS FOR MEASURING CURRENT. | **24** |
| **IV** | **SYSTEM DESIGN & IMPLEMENTATION** | **26** |
|  | 4.1 SYSTEM ARCHITECTURE | **27** |
|  | 4.1.1 PRODUCT FEATURES | **27** |
|  | 4.1.2 OPERATING ENVIRONMENT | **27** |
|  | 4.1.3 SYSTEM CONSTRAINTS | **27** |
|  | 4.2 MODULE DESCRIPTION | **28** |
|  | 4.2.1 ENERGY CALCULATION | **28** |
|  | 4.2.2 TEMPERATURE AND LIGHT SENSING | **30** |
|  | 4.2.3 MONITORING AND CONTROL  4.2.4 MESSAGE INDICATION | **30** |
| **V** | **REQUIREMENT SPECIFICATION** | **33** |
|  | 5.1 HARDWARE SPECIFICATION | **33** |
|  | 5.2 SOFTWARE SPECIFICATION | **34** |
| **VI** | **CODING AND TESTING** | **37** |
|  | 6.1 INSTALLATION | **37** |
|  | 6.2 CODING | **40** |
|  | 6.3 CODING STANDARDS | **40** |
|  | 6.3.1 CODING STYLE GUIDELINES | **40** |
|  | 6.4 TEST PROCEDURE | **41** |
|  | 6.5 TEST DATA AND OUTPUT | **42** |
|  | 6.5.1 UNIT TESTING | **42** |
|  | 6.5.2 INTEGRATION TESTING | **43** |
|  | 6.5.3 SYSTEM TESTING | **43** |
| **VII** | **IMPLEMENTATION AND RESULTS** | **44** |
| **VIII** | **CONCLUSION & FUTURE WORKS** | **93** |
| **IX** | **REFERENCES** | **94** |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| 1.1 | Typical LDR Symbol |
| 1.2 | LDR Circuit |
| 1.3 | Typical LDR resistance vs. light intensity graph |
| 1.4 | LDR controlled transistor circuit |
| 1.5 | LM35 |
| 1.6 | LM35 Circuit |
| 1.7 | LM35 on board |
| 3.1 | TNEB Tariff 1 |
| 3.2 | TNEB Tariff 2 |
| 3.3 | Light Sensor LDR |
| 3.4 | Temperature Sensor LM35 |
| 3.5 | GSM for Mobile Communication |
| 3.6 | Hall Effect Sensor |
| 4.1 | SEMS Architecture Diagram |
| 4.2 | SEMS Microprocessor Kit |
| 4.3 | LED glows on capturing parameters. |
| 4.4 | Configuring Thresholds |
| 4.5 | Light and Fan Switched ON automatically |
| 4.6 | Light switched OFF automatically |
| 4.7 | Text Message indicating the above two changes in fig 4.5 and fig 4.6 |
| 6.1 | Tref file |
| 6.2 | Eref file |
| 6.3 | Lref file |
| 6.4 | Phone\_1,2,3 file |

**CHAPTER I**

**INTRODUCTION**

**1.1 AIM AND OBJECTIVE**

SEMS mainly focuses to monitor and conserve Electricity in the simplest ways possible. It also focuses to periodically sense the Light and Temperature values, and switch OFF the Light/Fan if the parameters are sufficiently available for Human use. It includes the usage of sensors for Light & Temperature. Additional feature is that SMS alerts are sent to the users giving them the exact estimation of the EB Tariff consumption rates every day. SMS alerts are also sent when the Light/Temperature is available in sufficient amount and they are hence switched off automatically.

**1.1.1 ELECTRICITY MONITORING AND CONTROL**

Energy savings can usually be achieved by optimizing the operation of existing control systems. This makes the best use of the capital already invested in those systems.

The settings and control actions of older stand-alone controllers could usually be viewed by looking at gauges and indicator lights on a switchboard. Modern control system settings and control actions are usually only visible through a user interface.

**Electricity Monitoring and Control** (M&C) is an energy efficiency technique based on the standard [management](https://en.wikipedia.org/wiki/Management) axiom stating that “you cannot manage what you cannot measure”. M&C techniques provide [energy managers](https://en.wikipedia.org/wiki/Energy_engineering) with feedback on operating practices, results of energy management projects, and guidance on the level of energy use that is expected in a certain period. Importantly, they also give early warning of unexpected excess consumption caused by equipment malfunctions, operator error, unwanted user behaviors, lack of effective maintenance, etc...

The foundation of M&C lies in determining the normal relationships of energy consumptions to relevant driving factors (HVAC equipment, production though puts, weather, occupancy available daylight, etc.) and the goal is to help every individual in the household to:

* Identify and explain excessive energy use
* Detect instances when consumption is unexpectedly higher or lower than would usually have been the case
* Visualize energy consumption trends (daily, weekly, seasonal, operational…)
* Determine future energy use and costs when planning changes in the business
* Diagnose specific areas of wasted energy
* Observe how changes to relevant driving factors impact energy efficiency
* Develop performance targets for energy management programs
* Manage energy consumption, rather than accept it as a fixed cost

The ultimate goal is to reduce energy costs through improved [energy efficiency](https://en.wikipedia.org/wiki/Efficient_energy_use) and energy management control. Other benefits generally include increased resource efficiency, improved [production](https://en.wikipedia.org/wiki/Production,_costs,_and_pricing) budgeting and reduction of [greenhouse gas](https://en.wikipedia.org/wiki/Greenhouse_gas) (GHG) emissions.

**1.1.2 TEMPERATURE AND LIGHT SENSING AND CONTROL**

***Sensor***

Sensors are used in everyday objects such as touch-sensitive elevator buttons ([tactile sensor](https://en.wikipedia.org/wiki/Tactile_sensor)) and lamps which dim or brighten by touching the base, besides innumerable applications of which most people are never aware. With advances in [micro machinery](https://en.wikipedia.org/wiki/Micromachinery) and easy-to1-use [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) platforms, the uses of sensors have expanded beyond the traditional fields of temperature, pressure or flow measurement. A sensor's sensitivity indicates how much the sensor's output changes when the input quantity being measured changes. For instance, if the mercury in a thermometer moves 1 cm when the temperature changes by 1 °C, the sensitivity is 1 cm/°C (it is basically the slope Dy/Dx assuming a linear characteristic). Some sensors can also affect what they measure; for instance, a room temperature thermometer inserted into a hot cup of liquid cools the liquid while the liquid heats the thermometer. Sensors are usually designed to have a small effect on what is measured; making the sensor smaller often improves this and may introduce other advantages

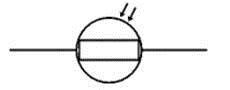
A good sensor obeys the following rules:

* It is sensitive to the measured property
* It is insensitive to any other property likely to be encountered in its application, and
* It does not influence the measured property.

The [sensitivity](https://en.wikipedia.org/wiki/Sensitivity_(electronics)) is then defined as the ratio between the output signal and measured property.

***Light Sensor***

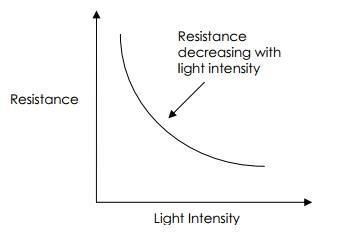
An [LDR](https://www.kitronik.co.uk/products/components/leds/light-sensors/) is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits.



**Fig 1.1: TYPICAL Fig 1.2: LDR CIRCUIT SYMBOL**

**LDR**

**Variation in resistance with changing light intensity**



**Fig 1.3: Typical LDR resistance vs light intensity graph**

The most common type of LDR has a resistance that falls with an increase in the light intensity falling upon the device (as shown in the image above).

The resistance of an LDR may typically have the following resistances:

Daylight

= 5000Ω

Dark

= 20000000Ω

You can therefore see that there is a large variation between these figures. If you plotted this variation on a graph you would get something similar to that shown by the graph shown above.

**Applications of LDRs**

There are many applications for Light Dependent Resistors. These include:

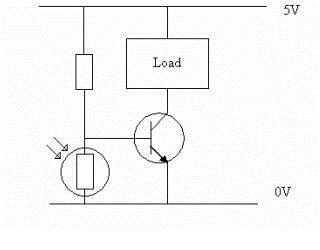
**Lighting switch**

The most obvious application for an LDR is to automatically turn on a light at a certain light level. An example of this could be a street light or a garden light.

**Camera shutter control**

LDRs can be used to control the shutter speed on a camera. The LDR would be used to measure the light intensity which then adjusts the camera shutter speed to the appropriate level.

**Example - LDR controlled Transistor circuit**

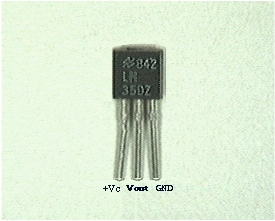


**Fig 1.4: LDR controlled transistor circuit**

***Temperature Sensor***

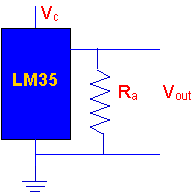
The LM35 is an integrated circuit sensor that can be used to measure temperature with an electrical output proportional to the temperature (in **o**C)

* **Why Use LM35’s To Measure Temperature?**
  + You can measure temperature more accurately than a using a thermistor.
  + The sensor circuitry is sealed and not subject to oxidation, etc.
  + The LM35 generates a higher output voltage than thermocouples and may not require that the output voltage be amplified.
* **What Does An LM35 Look Like?**



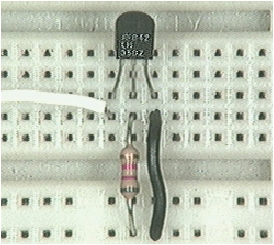
**Fig 1.5: LM35**

* **What Does an LM35 Do?  How does it work?**
  + It has an output voltage that is proportional to the Celsius temperature.
  + The scale factor is .01V/**o**C
  + The LM35 does not require any external calibration or trimming and maintains an accuracy of  +/-0.4 **o**C at room temperature and +/- 0.8 **o**C over a range of 0 **o**C to +100 **o**C.
  + Another important characteristic of the LM35DZ is that it draws only 60 micro amps from its supply and possesses a low self-heating capability. The sensor self-heating causes less than 0.1 oC temperature rise in still air.
* **How Do You Use An LM35?  (Electrical Connections)**
  + In this circuit, parameter values commonly used are:
    - Vc = 4 to 30v
    - 5v or 12 v are typical values used.
    - R**a** = V**c** /10**-6**
    - Actually, it can range from 80 K**W** to 600 K**W** , but most just use 80 K**W**.



**Fig 1.6: LM35 Circuit**

* + Here is a photo of the LM 35 wired on a circuit board.
    - The white wire in the photo goes to the power supply.
    - Both the resistor and the black wire go to ground.
    - The output voltage is measured from the middle pin to ground.



**Fig 1.7: LM35 on board**

* **What Can You Expect When You Use An LM35?**
  + You will need to use a voltmeter to sense Vout.
  + The output voltage is converted to temperature by a simple conversion factor.
  + The sensor has a sensitivity of 10mV / **o**C.
  + Use a conversion factor that is the reciprocal, which is 100 **o**C/V.
  + The general equation used to convert output voltage to temperature is:
    - Temperature ( **o**C) = Vout \* (100 **o**C/V)
    - So if  Vout  is  1V , then, Temperature = 100 **o**C
    - The output voltage varies linearly with temperature.

**1.2 SCOPE OF THE PROJECT**

Electricity is taking lion’s share of everybody’s income in the present scenario, and this creates a greater necessity for controlling the consumption of electricity right from our homes to organizations. This system is primarily aimed at having greater control over the electricity consumed in household and consequently reducing the consumption.

In this chapter, we provide information regarding the introduction of the project. This chapter includes the main aim and objectives, scope of the project, purpose of the project and an overall view of what all are implemented in this project. It also provides information about the basic project features, the benefits and constraints of the entire project.

**CHAPTER II**

**LITERATURE SURVEY**

**2.1 HOUSEHOLD ELECTRICITY CONSUMPTION ANALYSIS WITH DATA MINING TECHNIQUES**

**Authors** [Usman Ali](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Usman%20Ali.QT.&newsearch=true)., [Concettina Buccella](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Concettina%20Buccella.QT.&newsearch=true) and [Carlo Cecati](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Carlo%20Cecati.QT.&newsearch=true) - Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, Italy.

It was developed by [Usman Ali](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Usman%20Ali.QT.&newsearch=true)., [Concettina Buccella](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Concettina%20Buccella.QT.&newsearch=true) and [Carlo Cecati](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Carlo%20Cecati.QT.&newsearch=true) at the conference held onFebruary 2016. The Data mining techniques on the two different public data set are used and Electricity consumption patterns analyzed. Only analysis of electricity consumption is done and no control measures are taken.

**2.2 ELECTRICITY CONSUMPTION ANALYSIS AND POWER QUALITY MONITORING IN COMMERCIAL BUILDINGS**

**Authors** [Imre Drovtar](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Imre%20Drovtar.QT.&newsearch=true)., [Jaan Niitsoo](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Jaan%20Niitsoo.QT.&newsearch=true) and [Argo Rosin](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Argo%20Rosin.QT.&newsearch=true) - Department of Electrical Drives and Power Electronics, Tallinn University of Technology, 19086 Tallinn, Estoni.

It was developed by [Imre Drovtar](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Imre%20Drovtar.QT.&newsearch=true)., [Jaan Niitsoo](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Jaan%20Niitsoo.QT.&newsearch=true) and [Argo Rosin](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Argo%20Rosin.QT.&newsearch=true) at the conference held on December 2015. It is a Meta-heuristic algorithm which is used on power consumption data sets. Efficiency and quality are analyzed. In this paper, Efficiency and quality of power alone is measured.

**2.3** ELECTRICITY-METERING IN A CONNECTED WORLD:VIRTUAL SENSORS FOR ESTIMATING THE ELECTRICITY CONSUMPTION OF IoT APPLIANCES

**Authors** [Frank Englert](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Frank%20Englert.QT.&newsearch=true)., [Patrick Lieser](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Patrick%20Lieser.QT.&newsearch=true) and [Alaa Alhamoud](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Alaa%20Alhamoud.QT.&newsearch=true)

It was developed by [Frank Englert](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Frank%20Englert.QT.&newsearch=true)., [Patrick Lieser](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Patrick%20Lieser.QT.&newsearch=true) and [Alaa Alhamoud](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Alaa%20Alhamoud.QT.&newsearch=true) at the conference held onAugust 2015. This is a software based virtual energy sensors calculates energy consumed by IoT devices. There is no SMS alerts provided in this project.

**2.4 ANALYSIS OF THE RESIDENTIAL, COMMERCIAL AND INDUSTRIAL ELECTRICITY CONSUMPTION**

**Authors**  [John D. Hobby](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.John%20D.%20Hobby.QT.&newsearch=true) and  [Gabriel H. Tucci](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Gabriel%20H.%20Tucci.QT.&newsearch=true)

It was developed by [John D. Hobby](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.John%20D.%20Hobby.QT.&newsearch=true) and  [Gabriel H. Tucci](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Gabriel%20H.%20Tucci.QT.&newsearch=true) at the conference held on October 2015. Electricity consumption along with light and temperature parameters is gathered on daily basis. But no alerts are provided through SMS.

**2.5 AN INTELLIGENT ENERGY MANAGEMENT SCHEME** WITH MONITORING AND SCHEDULING APPROACH FOR IoT APPLICATIONS IN SMART HOME

**Author** [Tui-Yi Yang](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Tui-Yi%20Yang.QT.&newsearch=true)., [Chu-Sing Yang](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Chu-Sing%20Yang.QT.&newsearch=true) and [Tien-Wen Sung](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Tien-Wen%20Sung.QT.&newsearch=true)

It was developed by [Tui-Yi Yang](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Tui-Yi%20Yang.QT.&newsearch=true)., [Chu-Sing Yang](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Chu-Sing%20Yang.QT.&newsearch=true) and [Tien-Wen Sung](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Tien-Wen%20Sung.QT.&newsearch=true) at the conference held onOctober 2015. Electrical usage and appliance history is collected through statistical analysis and context-aware technologies in this paper. Light and Temperature sensors are not used.

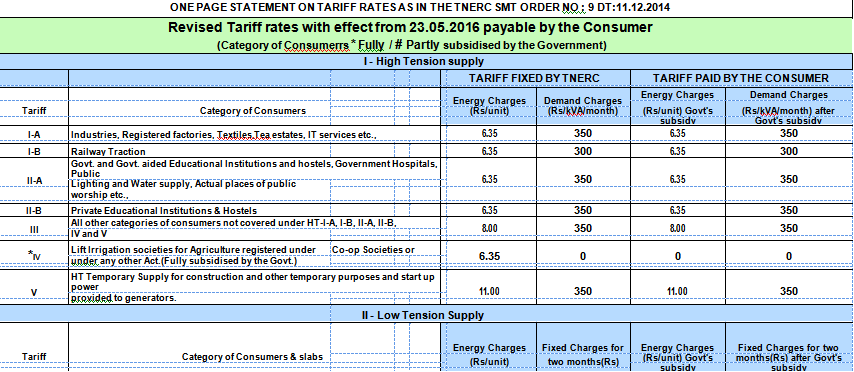
**CHAPTER III**

**SYSTEM ANALYSIS**

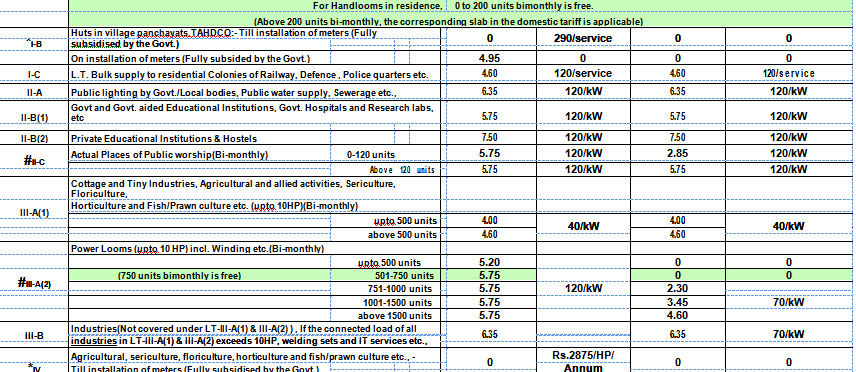
**3.1 EXISTING SYSTEM**

* Bimonthly TNEB Readings
* Manual and e-bill payment

**3.1.1 TNEB METERS TARIFF**



**Fig 3.1: TNEB Tariff 1**



**Fig 3.2: TNEB Tariff 2**

**DRAWBACKS**

Per device consumption is not known and proper SMS alerts for the consumption of the day and cumulative consumption of units and rate as per tariff is not given.

**3.2 PROPOSED SYSTEM**

The proposed system is an integrated Hardware with an application software with an User Interface, where the per device consumption is known and periodic alerts are given after the devices are switched OFF automatically by using Hall Effect Sensor if the parameters reach the threshold and also consumption rate for the day and cumulative consumptions are also given.

**3.2.1 LIGHT SENSOR**

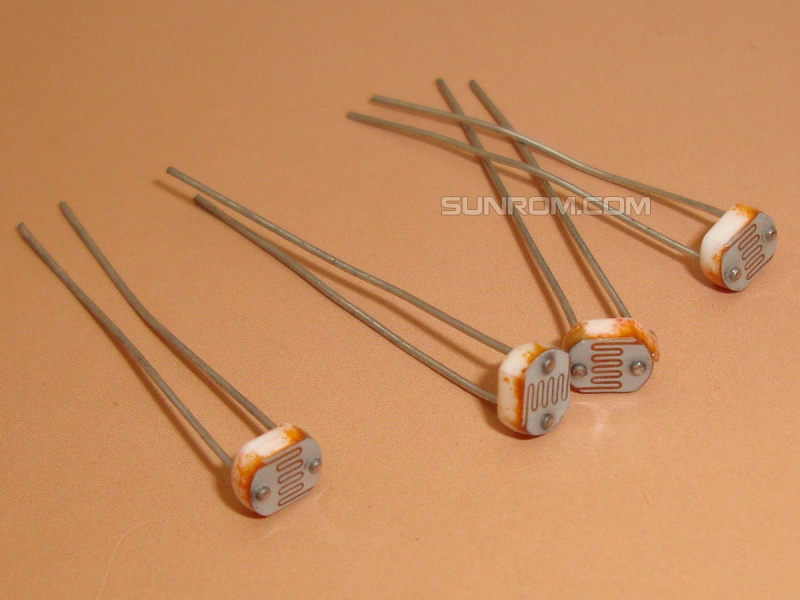
This is a **light dependent resistor** (LDR), suitable for use in projects which require a device or circuit to be automatically switched on or off in darkness or light.  
**As the amount of light falling on this LDR increases, its resistance decreases.**

Fig 3.3: Light sensor- LDR

**3.2.2 TEMPERATURE SENSOR**

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 μA from the supply, it has very low self-heating of less than 0.1°C in still air.

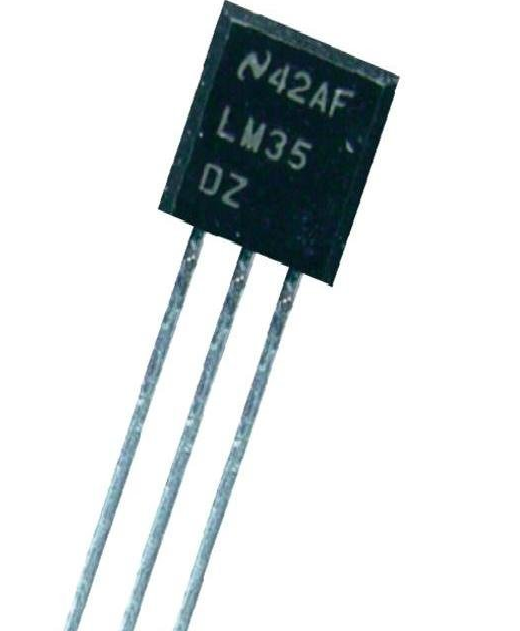


Fig 3.4: Temperature Sensor LM35

**3.2.3 GSM MODEM FOR MESSAGE COMMUNICATION**

GSM is a mobile communication modem; it is stands for global system for mobile communication (GSM). The idea of GSM was developed at Bell Laboratories in 1970.  It is widely used mobile communication system in the world. GSM is an open and digital cellular technology used for transmitting mobile voice and data services operates at the 850MHz, 900MHz, 1800MHz and 1900MHz frequency bands. GSM system was developed as a digital system using time division multiple access (TDMA) technique for communication purpose. A GSM digitizes and reduces the data, then sends it down through a channel with two different streams of client data, each in its own particular time slot. The digital system has an ability to carry 64 kbps to 120 Mbps of data rates.

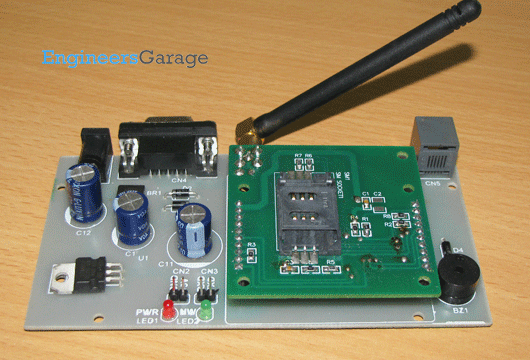


Fig 3.5: GSM for Mobile Communication

**3.2.4 HALL EFFECT SENSORS FOR MEASURING CURRENT**

A **Hall Effect sensor** is a [transducer](https://en.wikipedia.org/wiki/Transducer) that varies its output [voltage](https://en.wikipedia.org/wiki/Voltage) in response to a [magnetic field](https://en.wikipedia.org/wiki/Magnetic_field). [Hall Effect](https://en.wikipedia.org/wiki/Hall_effect) sensors are used for [proximity](https://en.wikipedia.org/wiki/Distance) switching, positioning, speed detection, and current sensing applications.

In its simplest form, the sensor operates as an [analog](https://en.wikipedia.org/wiki/Analog_(signal)) transducer, directly returning a voltage. With a known magnetic field, its distance from the Hall plate can be determined. Using groups of sensors, the relative position of the magnet can be deduced.

Frequently, a Hall sensor is combined with threshold detection so that it acts as and is called a [switch](https://en.wikipedia.org/wiki/Switch). Commonly seen in industrial applications such as the pictured [pneumatic cylinder](https://en.wikipedia.org/wiki/Pneumatic_cylinder), they are also used in consumer equipment; for example some [computer printers](https://en.wikipedia.org/wiki/Computer_printer) use them to detect missing paper and open covers. They can also be used in [computer keyboards](https://en.wikipedia.org/wiki/Keyboard_technology#Hall-effect_keyboard) applications that require ultra-high reliability.



Fig 3.6: Hall Effect Sensor

**CHAPTER IV**

**SYSTEM DESIGN AND IMPLEMENTATION**

**4.1 ARCHITECTURE DIAGRAM**

The overall architecture of Social Application For electricity consumption control, which is as follows, explains the functionality of the project, various modules and their respective functions. The overall architecture diagram is shown in the figure 4.1. There are four different modules in this project.

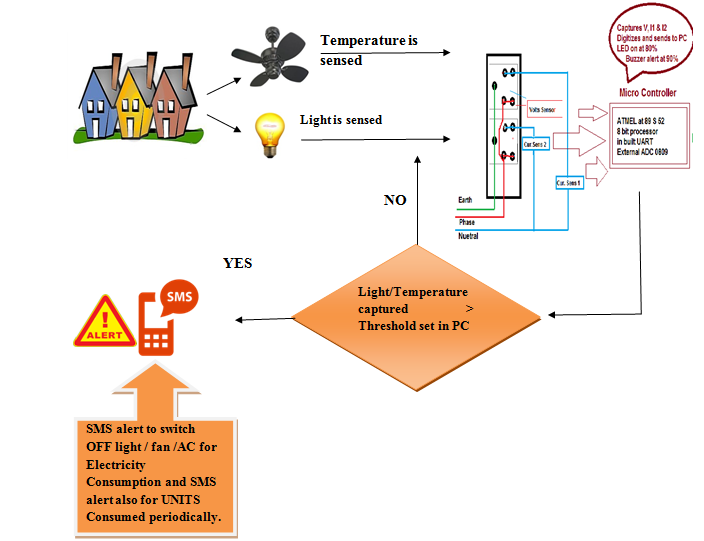
****

Fig 4.1 SEMS Architecture Diagram

**4.1.1 PRODUCT FEATURES**

The project proposed is a software application which can be easily made available everywhere. Any Windows OS supports this and above it can make use of this by installing the application in their laptops or desktop. SEMS provides the use the privilege to control and monitor 3 different parameters like current computation and control, temperature and light sensing. The basic system architecture diagram gives an overall view of all the processes in SEMS.

**4.1.2 OPERATING ENVIRONMENT**

The application proposed will run on all versions of Windows OS. All devices which support this version will be able to run the application. This platform is easily made reachable to a large number of users due to its easy user interface and the range of services it provides. The intention of this application is to provide greater control over the electricity consumed by us every day, and indicate when consumption overruns a pre-defined range to control the further consumption. It also aims at saving electricity by switching light and fan off when not required.

**4.1.3 SYSTEM CONSTRAINTS**

The project has its own design and implementation constraints. SEMS’s design can be viewed clearly as three separate processes. The first part is where the parameters like temperature, current and light are captured. Secondly, these captured parameters are computed by a microprocessor kit. The calculated data is finally compared with a set of pre defined thresholds and when it overruns the fixed threshold, corresponding operations take place and message indications are sent to the user. The till date consumption is also indicated to the user periodically, which makes it more useful and convenient.

**4.1.4 ASSUMPTIONS AND DEPENDENCIES**

The system is independent of the server and can be installed as a free application in any system which has a windows OS with any version. The laptop and the kit combines with a GSM modem to furnish the complete functionality, making it stand alone and ubiquitous.

**4.2 MODULE DESCRIPTION**

The SEMS application consists of four main modules each performing different operations.

* Parameters Capturing and Predefining the thresholds
* Computing energy consumed and Computing cost for energy consumed.
* Comparing the computed values with the threshold.
* Message indications stating the electricity consumption, cost and status of the appliances connected to the system.
  + 1. **PARAMETER CAPTURING AND PREDEFINING THRESHOLDS**

In order to capture all the above parameters like the current consumed by various equipments connected to the system, temperature parameter and the light parameter, different sensors are used such as haul effect sensor for current, LDR for sensing the LUX level of light and LM35 for sensing the temperature. The parameters are captured by circuiting all these components together along with other necessary components like step down transformer to step the incoming voltage down to the TT level, regulator to regulate and capacitor. The input is got in form of string and computed further. The initial threshold values are fixed for the temperature, light and the energy.

After capturing all these parameters it is sent to the next stage serially in the next subsequent steps.

******

Fig 4.2 SEMS Microprocessor kit



Fig. 4.3 LED glows on capturing parameters.

4.2.2. Energy computation and Computing cost for energy consumed

As described above the parameters are fetched and then computed in the micro processor kit. It manipulates all the data in form of voltage and computes the power consumed by the appliances. Along with the consumed power it also calculates the cost till that particular day based on the Electricity Board’s tarrrif and norms. These calculated values are then used for comparison with the threshold which is pre defined and the performs operations accordingly, providing indications to the end user.

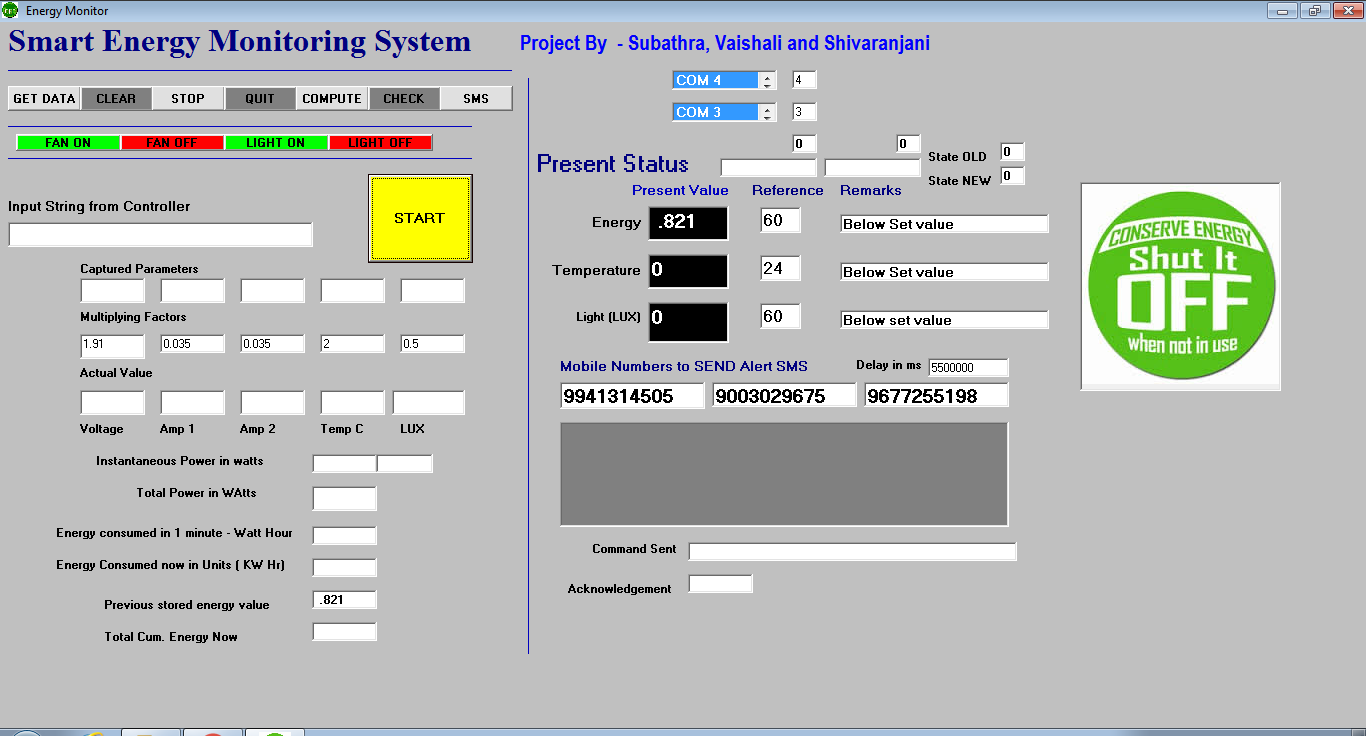


Fig 4.4.Configuring Thresholds

* + 1. Comparing the computed values with the threshold.

The computed values in the microprocessor kit is sent to the PC or laptop using the RS-32 cable, Serial communication is used for this transfer, and the following are done.

* If the temperature value is lesser than the set threshold, the fan gets automatically switched off and switched on in the vice versa case.
* If the sensed LUX level is greater than the set threshold, The light gets switched off and switched on in the vice versa case.
* If the consumed power overruns the threshold all the appliances are switched off.

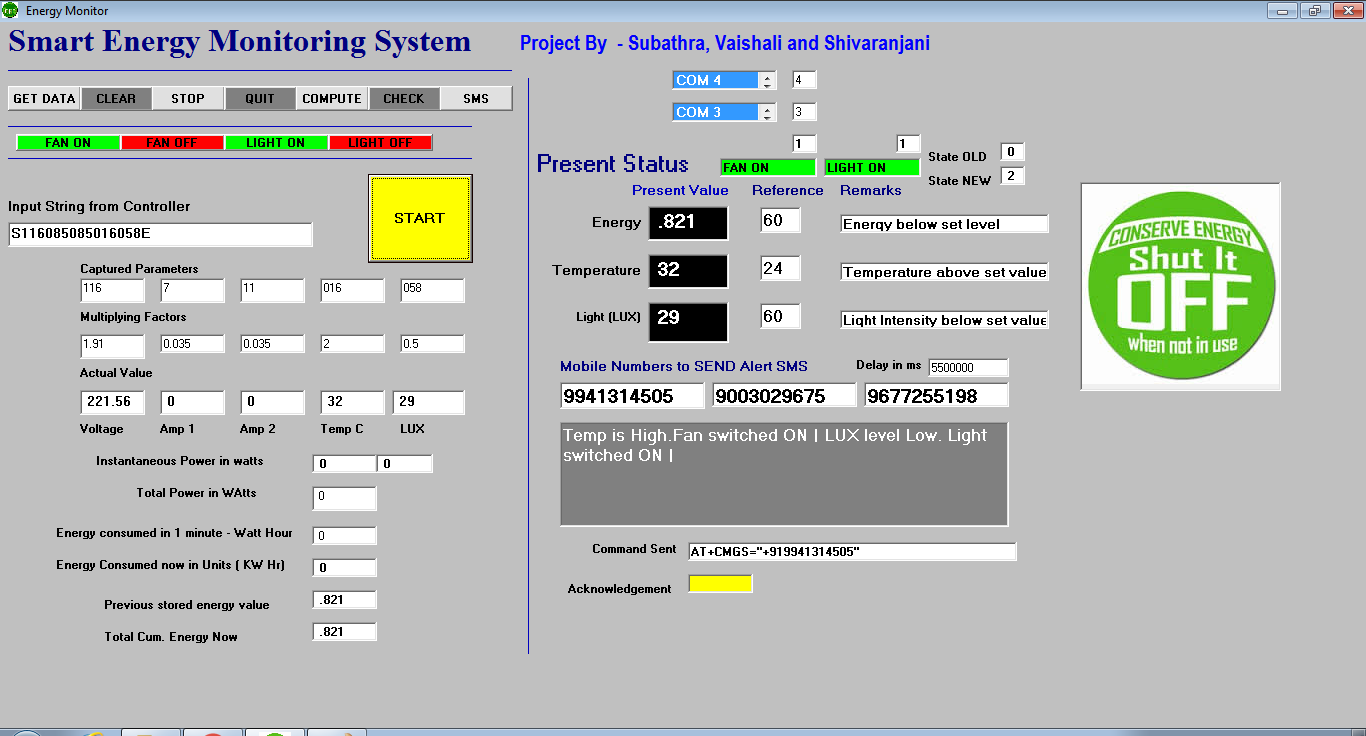


Fig 4.5 Fan and Light automatically switched ON

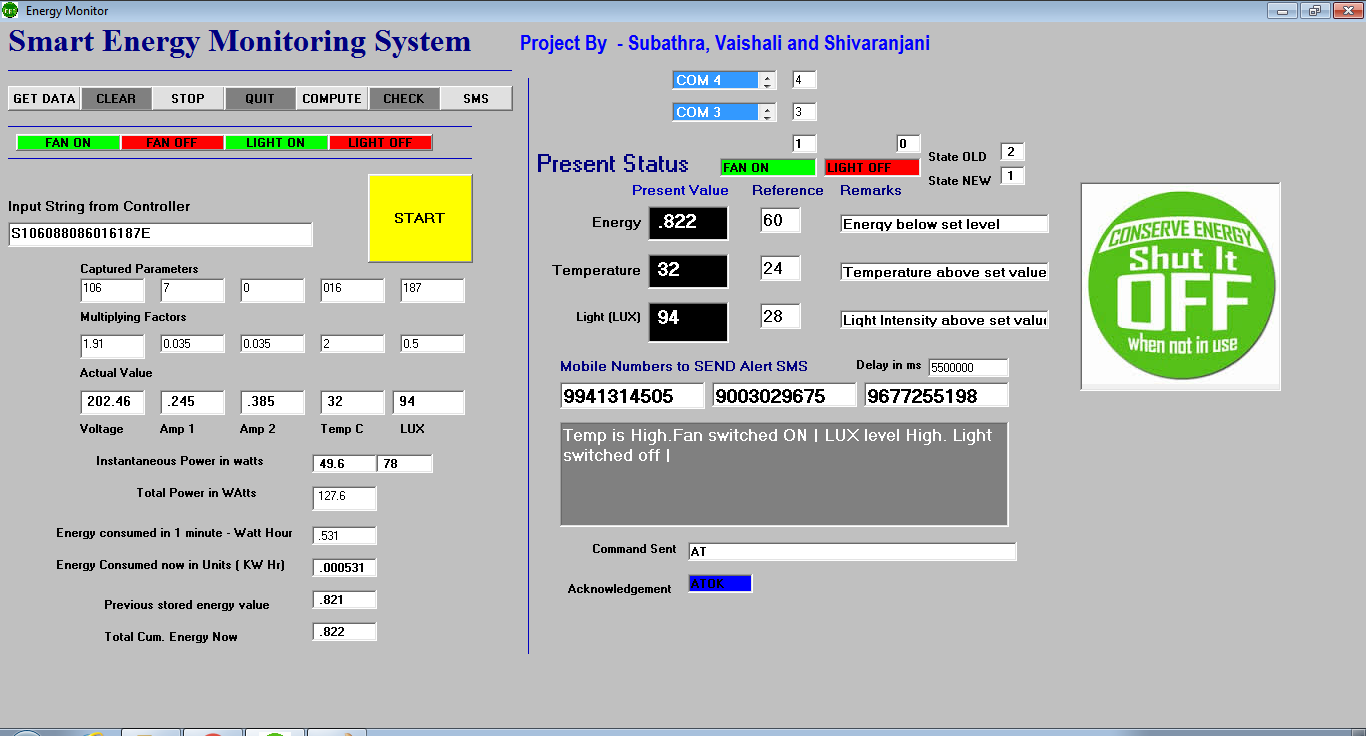


Fig 4.6. Light switched off automatically.

**4.2.4. Message indications stating the electricity consumption, cost and status of the appliances connected to the system.**

Message indications are given to the user stating the current status.

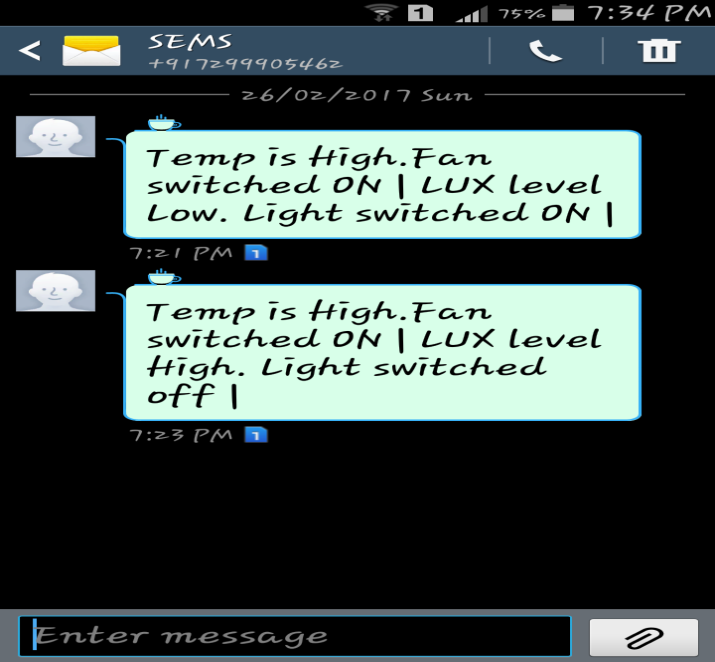


Fig 4.7. Text Message indicating the above two changes in fig 4.5 and fig 4.6

**CHAPTER V**

**REQUIREMENT SPECIFICATION**

**5.1 HARDWARE SPECIFICATION**

1. **Plug Sockets**

Two 5A plug sockets are used to allow electrically operated equipment to be connected to the primary alternating current or power supply.

1. **ADC 0809**

It is used to digitize the captured currents, voltage, temperature and light parameters. It requires clock signals to convert the analog inputs to digital.

1. **Microcontroller 89S52**

It commands the frequency timing interval to ADC and also what input to be chosen by ADC to digitize.

1. **Temperature sensor LM35**

The LM35 is an integrated circuit sensor that can be used to measure temperature with an electrical output proportional to the temperature (in **o**C).

1. **Light sensor LDR**

An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it.

1. **Max 232 chip**

The chip is used to convert Transistor-Transistor Logic (TTL) level to RS 232 level during serial communication of microcontroller with personal computer.

1. **Transistor Amplifier 2003**

It is used to set 2 lights to glow when data is being captured and sent to system.

1. **Rectifier 1N4007**

It is an electrical device that converts Alternating Current (AC), which periodically reverses direction, to direct current (DC), which flows in only one direction.

1. **Capacitor**

Capacitors are used to smooth a current in a circuit as they can prevent false triggering of other components such a relays.

1. **Regulator 7805**

It maintains the voltage of a power source within acceptable limits.

1. **Global System for Mobile communications (GSM)**

GSM is used to send the collected readings or data to the system which may be a mobile phone through SMS.

1. Bulb & Fan
2. Laptop –

***CPU* i5 CORE processer**

**RAM 1 GB**

**Hard Disk 80 GB**

1. Mobile Phone

**5.2 SOFTWARE SPECIFICATION**

A software requirements definition is an abstract description of the services which a system should provide and the constraints under which the system must operate.

1. Operating System - Windows 7/8
2. Assembly language 89s52 kit Software

An **assembly** (or **assembler**) **language**, often abbreviated **ASM**, is a [low-level programming language](https://en.wikipedia.org/wiki/Low-level_programming_language) for a [computer](https://en.wikipedia.org/wiki/Computer), or other programmable device, in which there is a very strong (generally [one-to-one](https://en.wikipedia.org/wiki/One-to-one_correspondence)) correspondence between the language and the [architecture's](https://en.wikipedia.org/wiki/Computer_architecture) [machine code](https://en.wikipedia.org/wiki/Machine_code) [instructions](https://en.wikipedia.org/wiki/Instruction_(computer_science)). Each assembly language is specific to particular computer architecture. In contrast, most [high-level programming languages](https://en.wikipedia.org/wiki/High-level_programming_language) are generally [portable](https://en.wikipedia.org/wiki/Porting) across multiple architectures but require [interpreting](https://en.wikipedia.org/wiki/Interpreter_(computing)) or [compiling](https://en.wikipedia.org/wiki/Compiler). Assembly language may also be called *symbolic machine code*.

Assembly language is converted into executable machine code by a [utility program](https://en.wikipedia.org/wiki/Utility_program) referred to as an [*assembler*](https://en.wikipedia.org/wiki/Assembly_language#Assembler). The conversion process is referred to as *assembly*, or *assembling* the [source code](https://en.wikipedia.org/wiki/Source_code). *Assembly time* is the computational step where an assembler is run.

Assembly language uses a [mnemonic](https://en.wikipedia.org/wiki/Mnemonic) to represent each low-level [machine instruction](https://en.wikipedia.org/wiki/Machine_instruction) or [opcode](https://en.wikipedia.org/wiki/Opcode), typically also each [architectural register](https://en.wikipedia.org/wiki/Register_(computing)#ARCHITECTURAL), [flag](https://en.wikipedia.org/wiki/Flag_(computing)), etc. Many operations require one or more [operands](https://en.wikipedia.org/wiki/Operand#Computer_science) in order to form a complete instruction and most assemblers can take [expressions](https://en.wikipedia.org/wiki/Expression_(computer_science)) of numbers and named constants as well as registers and [labels](https://en.wikipedia.org/wiki/Label_(computing)) as operands, freeing the programmer from tedious repetitive calculations. Depending on the architecture, these elements may also be combined for specific instructions or [addressing modes](https://en.wikipedia.org/wiki/Addressing_mode) using [offsets](https://en.wikipedia.org/wiki/Offset_(computer_science)) or other data as well as fixed addresses. Many assemblers offer additional mechanisms to facilitate program development, to control the assembly process, and to aid [debugging](https://en.wikipedia.org/wiki/Debugging).

1. Microsoft Visual Studio

**Microsoft Visual Studio** is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program) for [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), as well as [web sites](https://en.wikipedia.org/wiki/Web_site), [web apps](https://en.wikipedia.org/wiki/Web_app), [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). Visual Studio uses Microsoft software development platforms such as [Windows API](https://en.wikipedia.org/wiki/Windows_API), [Windows Forms](https://en.wikipedia.org/wiki/Windows_Forms), [Windows Presentation Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](https://en.wikipedia.org/wiki/Windows_Store) and [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight). It can produce both [native code](https://en.wikipedia.org/wiki/Native_code) and [managed code](https://en.wikipedia.org/wiki/Managed_code).

Visual Studio includes a [code editor](https://en.wikipedia.org/wiki/Code_editor) supporting [IntelliSense](https://en.wikipedia.org/wiki/IntelliSense) (the [code completion](https://en.wikipedia.org/wiki/Code_completion) component) as well as [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring). [The integrated debugger](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building [GUI](https://en.wikipedia.org/wiki/GUI) applications, [web designer](https://en.wikipedia.org/wiki/Web_designer), [class](https://en.wikipedia.org/wiki/Class_(computing)) designer, and [database schema](https://en.wikipedia.org/wiki/Database_schema) designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for [source control](https://en.wikipedia.org/wiki/Source_control) systems (like [Subversion](https://en.wikipedia.org/wiki/Subversion_(software))) and adding new toolsets like editors and visual designers for [domain-specific languages](https://en.wikipedia.org/wiki/Domain-specific_language) or toolsets for other aspects of the [software development lifecycle](https://en.wikipedia.org/wiki/Software_development_lifecycle) (like the [Team Foundation Server](https://en.wikipedia.org/wiki/Team_Foundation_Server) client: Team Explorer).

Visual Studio supports different [programming languages](https://en.wikipedia.org/wiki/Programming_language) and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B) and [C++/CLI](https://en.wikipedia.org/wiki/C%2B%2B/CLI) (via [Visual C++](https://en.wikipedia.org/wiki/Visual_C%2B%2B)), [VB.NET](https://en.wikipedia.org/wiki/VB.NET) (via [Visual Basic .NET](https://en.wikipedia.org/wiki/Visual_Basic_.NET)), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)) (via [Visual C#](https://en.wikipedia.org/wiki/Visual_C_Sharp)), and [F#](https://en.wikipedia.org/wiki/F_Sharp_(programming_language)) (as of Visual Studio 2010). Support for other languages such as [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)), [Node.js](https://en.wikipedia.org/wiki/Node.js), and [M](https://en.wikipedia.org/wiki/MUMPS) among others is available via language services installed separately. It also supports [XML](https://en.wikipedia.org/wiki/XML)/[XSLT](https://en.wikipedia.org/wiki/XSLT), [HTML](https://en.wikipedia.org/wiki/HTML)/[XHTML](https://en.wikipedia.org/wiki/XHTML), [JavaScript](https://en.wikipedia.org/wiki/JavaScript) and [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). Java (and J#) was supported in the past.

**CHAPTER VI**

**CODING AND TESTING**

**6.1 INSTALLATION**

**Files to be configured:**

------------------------

* tref - Temperature limit

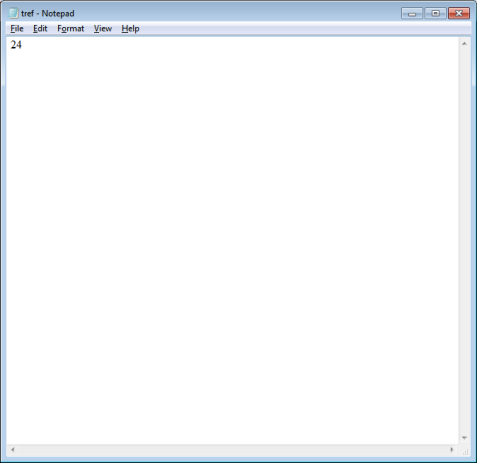


Fig 6.1: tref file

* eref - Energy limit

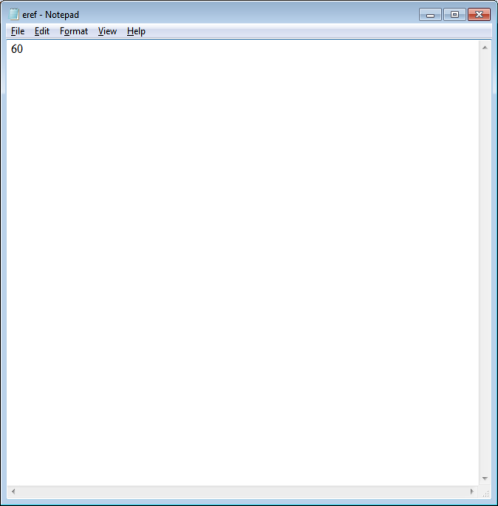


Fig 6.2: eref file

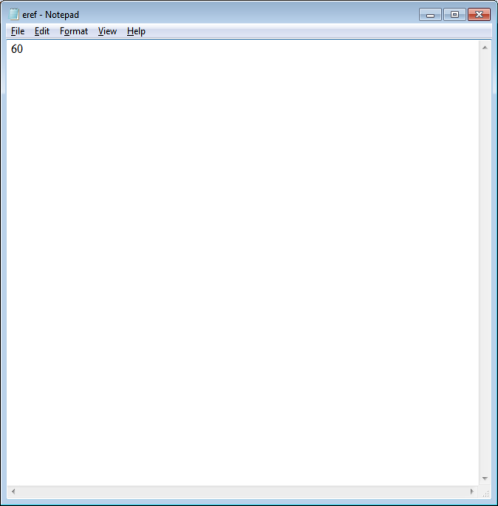
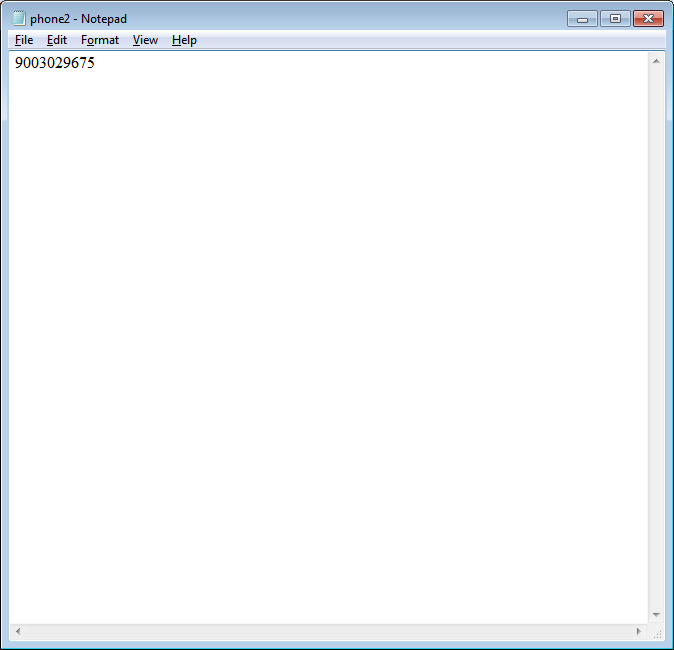
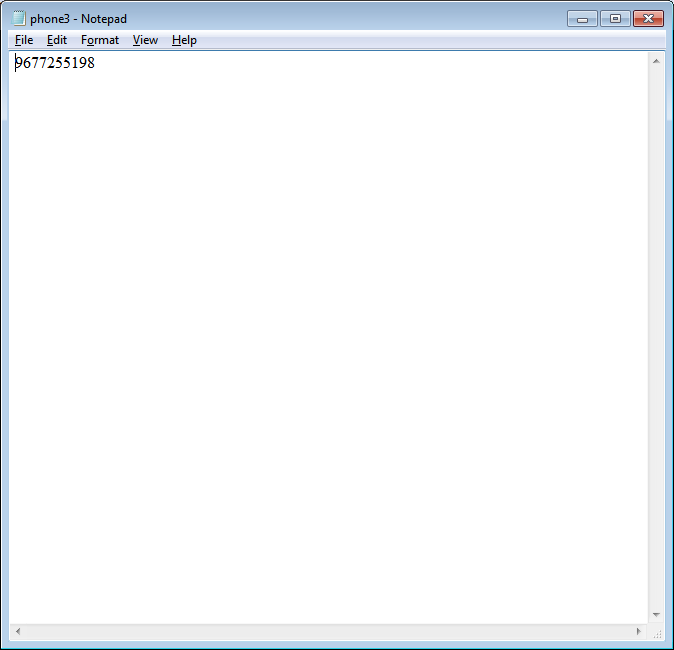
* lref - Light level limit

Fig 6.3: lref file

* Phone1
* Phone2
* Phone3

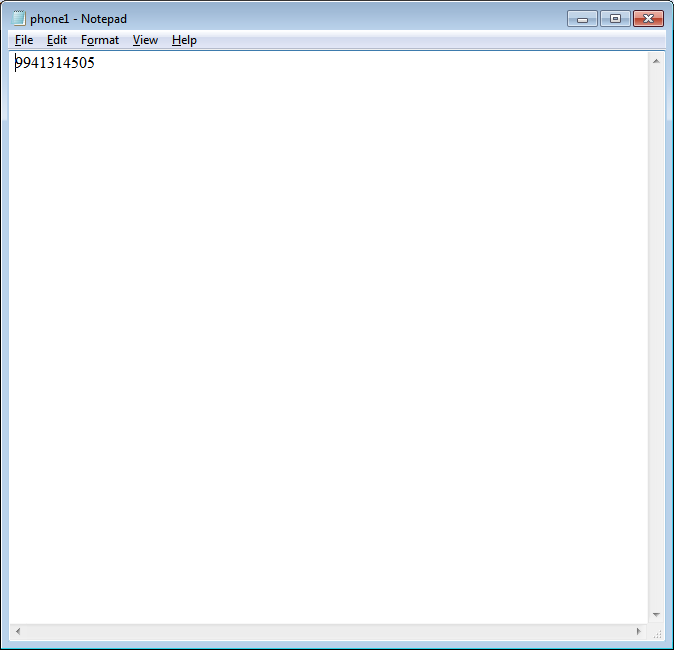


Fig 6.4: phone\_1,2,3 file

**Setting the Communication ports.**

--------------------------------

1. Connect the USB cable to PC/laptop

2. Right click on my computer

3. Click on manage

4. Click on device manager

5. Find the newly added communication port number

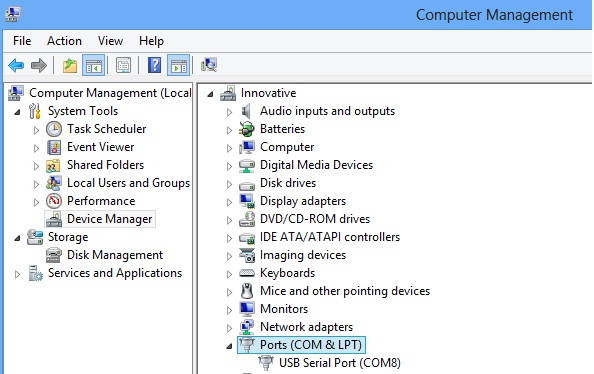


Fig 6.5: Communication port

Based on the port details, the following files to be configured

**Unitport** – USB port connecting the control unit

**Smsport** – USB port connecting the modem

**6.2 CODING**

Once the design aspect of the application is finalized, the application enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the application into the code in a given programming language. Therefore, a good coding style has to be taken and whenever changes are required, it should be easily screwed into the system.

**6.3 CODING STANDARDS**

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary.

**6.3.1 CODING STYLE GUIDELINES**

The rules below are not guidelines or recommendations, but they are strict rules. All the existing codes as well as the new codes should follow these rules.

* Java Language rules:

There are few java coding conventions that are to be followed.

* Do not ignore the exceptions
* Do not catch the generic exceptions

There are rare exceptions to this rule: There are certain test codes and top-level codes where there might be a need to catch all kinds of errors. In that case the generic Exception is caught and the error is handled appropriately.

Alternatives to catching generic Exception:

* Catch each exception separately as separate catch blocks after a single try. This can be awkward but is still preferable to catching all Exceptions. Beware repeating too much code in the catch blocks.
* Refactor the code to have more fine-grained error handling, with multiple try blocks. Split up the IO from the parsing, handle errors separately in each case.
* Rethrow the exception. Many times there isn’t a need to catch the exception at this level.
* Finalizers should not be used

Finalizers are a way to have a chunk of code executed when an object is garbage collected.

* Java style rules:
  + Methods should be short.
  + Fields are defined in standard places.
  + Limit the variable scope.
  + Indentation should be used for spaces.
  + Follow the Naming Conventions.
  + Standard Brace style should be used.
* Java tests style rules:
* Naming Conventions:

Naming conventions of classes, data member, member functions, procedures etc., should be self-descriptive. One should even get the meaning and scope of the variable by its name. The conventions are adopted for easy understanding of the intended message by the user. So it is customary to follow the conventions.

**6.4 TEST PROCEDURE**

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in it. If it is not detected before coding commences, the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough. Testing is one of the important steps in the software development phase.

**6.5 TEST DATA AND OUTPUT**

It explains the various testing methods incorporated in the application to overcome the errors.

**6.5.1 UNIT TESTING**

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module.

***Black-Box Testing***

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing.

Testing the SEMS kit by just switching ON the power supply and the software where the data string is captured is the Black Box Testing.

***White-Box Testing***

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code.

Sending SMS alerts using the logic of EB Tariff rates for each and every day’s consumption and the cumulative consumption is the white Box Testing.

**6.5.2 INTEGRATION TESTING**

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

**6.5.3 SYSTEM TESTING**

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

**CHAPTER VII**

**IMPLEMENTATION AND RESULTS**

Private Sub Command1\_Click()

'collect data from unit

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

MSComm1.CommPort = List1.ListIndex + 1

MSComm1.Settings = "9600,N,8,1"

MSComm1.InBufferSize = 4096

MSComm1.Handshaking = comNone

MSComm1.PortOpen = True

Text1.Text = ""

Dim i As Integer

j = 0

Dim inval As String

Dim volt, amp1, amp2, temp, lux As Integer

For i = 1 To 50

MSComm1.InputLen = 0

Do

DoEvents

Loop Until MSComm1.InBufferCount >= 1

inval = MSComm1.Input

Text1.Text = Trim(Text1.Text + inval)

If inval = "E" Then

Call Command5\_Click

Call Command6\_Click

End If

If i = 40 Then

i = 1

End If

Next i

End Sub

Private Sub Command10\_Click()

'light off

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

MSComm1.CommPort = List1.ListIndex + 1

MSComm1.Settings = "9600,N,8,1"

MSComm1.InBufferSize = 4096

MSComm1.Handshaking = comNone

MSComm1.PortOpen = True

MSComm1.Output = "4"

Text38.Text = "LIGHT OFF"

Text38.BackColor = vbRed

Text4.Text = "0"

Text45.Text = "0"

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

End Sub

Private Sub Command11\_Click()

'send sms

If Len(Text36.Text) > 140 Then

MsgBox " Maximum allowed length is only 140 Chrs.", vbCritical

Exit Sub

End If

Dim inval, instring As String

Dim i, j, k As Integer

Dim del As Double

del = Val(Text39.Text)

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

MSComm1.Settings = "9600,N,8,1"

MSComm1.CommPort = List2.ListIndex + 1

MSComm1.InBufferSize = 4096

MSComm1.Handshaking = comNone

MSComm1.PortOpen = True

Text40.BackColor = vbWhite

Text40.Text = ""

For i = 1 To del

DoEvents

Next i

AT1:

Text41.Text = "AT" + Chr(13)

Text40.Text = ""

MSComm1.Output = Text41.Text

Text40.BackColor = vbBlue

MSComm1.InputLen = 0

Do

DoEvents

Loop Until MSComm1.InBufferCount >= 1

inval = MSComm1.Input

Text40.Text = inval

For i = 1 To del

DoEvents

Next i

If Text40.Text = "" Then

GoTo AT1

End If

ATCMGF1:

Text40.Text = ""

Text41.Text = "AT+CMGF=1" + Chr(13)

MSComm1.Output = Text41.Text

Text40.BackColor = vbGreen

MSComm1.InputLen = 0

Do

DoEvents

Loop Until MSComm1.InBufferCount >= 1

inval = MSComm1.Input

Text40.Text = inval

For i = 1 To del

DoEvents

Next i

If Text40.Text = "" Then

GoTo ATCMGF1

End If

ATCMGS1:

Text41.Text = Trim("AT+CMGS=" + """" + "+91" + Text33.Text + """")

Text40.Text = ""

MSComm1.Output = Text41.Text + Chr(13)

Text40.BackColor = vbYellow

For i = 1 To (del \* 5)

DoEvents

Next i

SENDMES1:

MSComm1.Output = Text36.Text + Chr(13)

Text40.Text = ""

MSComm1.Output = Chr(26)

MSComm1.InputLen = 0

Do

DoEvents

Loop Until MSComm1.InBufferCount >= 1

inval = MSComm1.Input

Text40.Text = inval

If Text40.Text <> "" Then

Text41.Text = "The Message has been Sent"

End If

For i = 1 To del

DoEvents

Next i

Text40.Text = ""

Text40.BackColor = vbWhite

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

MSComm1.Settings = "9600,N,8,1"

MSComm1.CommPort = List2.ListIndex + 1

MSComm1.InBufferSize = 4096

MSComm1.Handshaking = comNone

MSComm1.PortOpen = True

Text40.BackColor = vbWhite

Text40.Text = ""

For i = 1 To del

DoEvents

Next i

AT2:

Text41.Text = "AT" + Chr(13)

Text40.Text = ""

MSComm1.Output = Text41.Text

Text40.BackColor = vbBlue

MSComm1.InputLen = 0

Do

DoEvents

Loop Until MSComm1.InBufferCount >= 1

inval = MSComm1.Input

Text40.Text = inval

For i = 1 To del

DoEvents

Next i

If Text40.Text = "" Then

GoTo AT2

End If

ATCMGF2:

Text40.Text = ""

Text41.Text = "AT+CMGF=1" + Chr(13)

MSComm1.Output = Text41.Text

Text40.BackColor = vbGreen

MSComm1.InputLen = 0

Do

DoEvents

Loop Until MSComm1.InBufferCount >= 1

inval = MSComm1.Input

Text40.Text = inval

For i = 1 To del

DoEvents

Next i

If Text40.Text = "" Then

GoTo ATCMGF2

End If

ATCMGS2:

Text41.Text = Trim("AT+CMGS=" + """" + "+91" + Text34.Text + """")

Text40.Text = ""

MSComm1.Output = Text41.Text + Chr(13)

Text40.BackColor = vbYellow

For i = 1 To (del \* 5)

DoEvents

Next i

SENDMES2:

MSComm1.Output = Text36.Text + Chr(13)

Text40.Text = ""

MSComm1.Output = Chr(26)

MSComm1.InputLen = 0

Do

DoEvents

Loop Until MSComm1.InBufferCount >= 1

inval = MSComm1.Input

Text40.Text = inval

If Text40.Text <> "" Then

Text41.Text = "The Message has been Sent"

End If

For i = 1 To del

DoEvents

Next i

Text40.Text = ""

Text40.BackColor = vbWhite

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

MSComm1.Settings = "9600,N,8,1"

MSComm1.CommPort = List2.ListIndex + 1

MSComm1.InBufferSize = 4096

MSComm1.Handshaking = comNone

MSComm1.PortOpen = True

Text40.BackColor = vbWhite

Text40.Text = ""

For i = 1 To del

DoEvents

Next i

AT3:

Text41.Text = "AT" + Chr(13)

Text40.Text = ""

MSComm1.Output = Text41.Text

Text40.BackColor = vbBlue

MSComm1.InputLen = 0

Do

DoEvents

Loop Until MSComm1.InBufferCount >= 1

inval = MSComm1.Input

Text40.Text = inval

For i = 1 To del

DoEvents

Next i

If Text40.Text = "" Then

GoTo AT3

End If

ATCMGF3:

Text40.Text = ""

Text41.Text = "AT+CMGF=1" + Chr(13)

MSComm1.Output = Text41.Text

Text40.BackColor = vbGreen

MSComm1.InputLen = 0

Do

DoEvents

Loop Until MSComm1.InBufferCount >= 1

inval = MSComm1.Input

Text40.Text = inval

For i = 1 To del

DoEvents

Next i

If Text40.Text = "" Then

GoTo ATCMGF3

End If

ATCMGS3:

Text41.Text = Trim("AT+CMGS=" + """" + "+91" + Text35.Text + """")

Text40.Text = ""

MSComm1.Output = Text41.Text + Chr(13)

Text40.BackColor = vbYellow

For i = 1 To (del \* 5)

DoEvents

Next i

SENDMES3:

MSComm1.Output = Text36.Text + Chr(13)

Text40.Text = ""

MSComm1.Output = Chr(26)

MSComm1.InputLen = 0

Do

DoEvents

Loop Until MSComm1.InBufferCount >= 1

inval = MSComm1.Input

Text40.Text = inval

If Text40.Text <> "" Then

Text41.Text = "The Message has been Sent"

End If

For i = 1 To del

DoEvents

Next i

Text40.Text = ""

Text40.BackColor = vbWhite

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

End Sub

Private Sub Command12\_Click()

' start function

Dim del, i As Long

del = Val(Text39.Text)

For i = 1 To del

DoEvents

Next i

Call Command1\_Click

End Sub

Private Sub Command3\_Click()

'stop

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

End Sub

Private Sub Command4\_Click()

'quit

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

End

End Sub

Private Sub Command5\_Click()

'compute

Dim vfact, afact, tfact, ifact, oldenergy, energyold, energynow, energynew, power1, power2 As Long

vfact = Val(Text7.Text)

afact = Val(Text8.Text)

tfact = Val(Text10.Text)

ifact = Val(Text11.Text)

If Left(Text1.Text, 1) = "S" Then

If Len(Text1.Text) = 17 Then

Text2.Text = Mid(Text1.Text, 2, 3)

'Text3.Text = Mid(Text1.Text, 5, 3)

'Text4.Text = Mid(Text1.Text, 8, 3)

Text5.Text = Mid(Text1.Text, 11, 3)

Text6.Text = Mid(Text1.Text, 14, 3)

End If

Open App.Path + "\energy.txt" For Input As #1

Do While Not EOF(1)

Line Input #1, oldenergy

Text31.Text = oldenergy

Loop

Close #1

Text12.Text = Str(Round((Val(Text2.Text) \* vfact), 2))

Text13.Text = Str(Round((Val(Text3.Text) \* afact), 3))

Text14.Text = Str(Round((Val(Text4.Text) \* afact), 3))

Text15.Text = Str(Round((Val(Text5.Text) \* tfact), 0))

Text16.Text = Str(Round((Val(Text6.Text) \* Val(Text11.Text)), 0))

power1 = Round((Val(Text12.Text) \* Val(Text13.Text)), 1)

power2 = Round((Val(Text12.Text) \* Val(Text14.Text)), 1)

Text26.Text = Str(power1)

Text27.Text = Str(power2)

Text28.Text = Str(power1 + power2)

energynow = Round(((Val(Text12.Text) \* ((Val(Text13.Text)) + (Val(Text14.Text))) / 60)), 3)

Text29.Text = Str(energynow)

Text30.Text = Str(energynow / 1000)

energyold = Val(Text31.Text)

energynew = Round((energyold + (energynow / 1000)), 3)

Text32.Text = Str(energynew)

Open App.Path + "\energy.txt" For Output As #2

Print #2, Text32.Text

Close #2

Text17.Text = Text32.Text

Text20.Text = Text15.Text

Text23.Text = Text16.Text

End If

End Sub

Private Sub Command6\_Click()

'check

Text36.Text = ""

Dim alert, statold, statnew As Integer

alert = 0

Text19.Text = "Energy below set level"

Text25.Text = "Light Intensity below set value"

If Val(Text20.Text) >= Val(Text21.Text) Then

Text22.Text = "Temperature above set value."

Text36.Text = Text36.Text + "Temp is High.Fan switched ON | "

Call Command7\_Click

Text3.Text = "7"

alert = alert + 1

End If

If Val(Text20.Text) < Val(Text21.Text) Then

Text22.Text = "Temperature below set value."

Text36.Text = Text36.Text + "Temp is low.Fan switched oFF | "

alert = alert + 1

Call Command8\_Click

Text3.Text = "0"

End If

If Val(Text23.Text) <= Val(Text24.Text) Then

Text25.Text = "Light Intensity below set value."

Text36.Text = Text36.Text + "LUX level Low. Light switched ON | "

alert = alert + 1

Call Command9\_Click

Text4.Text = "11"

End If

If Val(Text23.Text) > Val(Text24.Text) Then

Text25.Text = "Light Intensity above set value."

Text36.Text = Text36.Text + "LUX level High. Light switched off | "

alert = alert + 1

Call Command10\_Click

Text4.Text = "0"

End If

If Val(Text17.Text) > Val(Text18.Text) Then

Text19.Text = "Energy exceeded set value."

Text36.Text = "Energy above Limit.Fan and light switched OFF | "

Call Command8\_Click

Call Command10\_Click

alert = alert + 1

End If

statold = Val(Text47.Text)

statnew = Str(Val(Text44.Text) + Val(Text45.Text))

Text46.Text = Str(statold)

Text47.Text = Str(statnew)

If statold<>statnew Then

Call Command11\_Click

End If

Call Command12\_Click

End Sub

Private Sub Command7\_Click()

'fan on

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

MSComm1.CommPort = List1.ListIndex + 1

MSComm1.Settings = "9600,N,8,1"

MSComm1.InBufferSize = 4096

MSComm1.Handshaking = comNone

MSComm1.PortOpen = True

MSComm1.Output = "1"

Text37.Text = "FAN ON"

Text37.BackColor = vbGreen

Text3.Text = "7"

Text44.Text = "1"

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

End Sub

Private Sub Command8\_Click()

'fan off

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

MSComm1.CommPort = List1.ListIndex + 1

MSComm1.Settings = "9600,N,8,1"

MSComm1.InBufferSize = 4096

MSComm1.Handshaking = comNone

MSComm1.PortOpen = True

MSComm1.Output = "2"

Text37.Text = "FAN OFF"

Text37.BackColor = vbRed

Text3.Text = "0"

Text44.Text = "0"

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

End Sub

Private Sub Command9\_Click()

'light on

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

MSComm1.CommPort = List1.ListIndex + 1

MSComm1.Settings = "9600,N,8,1"

MSComm1.InBufferSize = 4096

MSComm1.Handshaking = comNone

MSComm1.PortOpen = True

MSComm1.Output = "3"

Text38.Text = "LIGHT ON"

Text38.BackColor = vbGreen

Text4.Text = "11"

Text45.Text = "1"

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

End Sub

Private Sub Form\_Load()

Dim energyref, tempref, luxref, unitport, smsport, delay, mob1, mob2, mob3 As String

Open App.Path + "\eref.txt" For Input As #5

Do While Not EOF(5)

Line Input #5, energyref

Text18.Text = energyref

Loop

Close #5

Open App.Path + "\tref.txt" For Input As #6

Do While Not EOF(6)

Line Input #6, tempref

Text21.Text = tempref

Loop

Close #6

Open App.Path + "\lref.txt" For Input As #7

Do While Not EOF(7)

Line Input #7, luxref

Text24.Text = luxref

Loop

Close #7

Open App.Path + "\energy.txt" For Input As #8

Do While Not EOF(8)

Line Input #8, oldenergy

Text17.Text = oldenergy

Text31.Text = oldenergy

Loop

Close #8

Open App.Path + "\unitport.txt" For Input As #12

Do While Not EOF(12)

Line Input #12, unitport

Text42.Text = unitport

Loop

Close #12

Open App.Path + "\smsport.txt" For Input As #13

Do While Not EOF(13)

Line Input #13, smsport

Text43.Text = smsport

Loop

Close #13

Open App.Path + "\delay.txt" For Input As #14

Do While Not EOF(14)

Line Input #14, delay

Text39.Text = delay

Loop

Close #14

Open App.Path + "\phone1.txt" For Input As #15

Do While Not EOF(15)

Line Input #15, mob1

Text33.Text = mob1

Loop

Close #15

Open App.Path + "\phone2.txt" For Input As #16

Do While Not EOF(16)

Line Input #16, mob2

Text34.Text = mob2

Loop

Close #16

Open App.Path + "\phone3.txt" For Input As #17

Do While Not EOF(17)

Line Input #17, mob3

Text35.Text = mob3

Loop

Close #14

List1.ListIndex = Val((Text42.Text) - 1)

List2.ListIndex = Val((Text43.Text) - 1)

End Sub

Private Sub Form\_Unload(Cancel As Integer)

Open App.Path + "\eref.txt" For Output As #11

Print #11, Text18.Text

Close #11

Open App.Path + "\tref.txt" For Output As #9

Print #9, Text21.Text

Close #9

Open App.Path + "\eref.txt" For Output As #10

Print #10, Text24.Text

Close #10

End Sub

**ASSEMBLY LANGUAGE CODE FOR MICROPROCESSOR 89s52**

;=================================================

; Assembly code for Energy Monitoring programme

Using 89S52 processor.

Port 1 - ADC

Pins 3.2,3.3,3.4 - ADC address registers Clear - 0 Set 1

Used address - 000 or 001 or 010 or 011 or 100

Volt Amp1 Amp2 Temp Light

Pin 3.5 - ADC address latch pin

Pin 2.0 - LED yellow - Clear - OFF | Set- ON

Pin 2.1 - LED Green - Clear - OFF | Set- ON

Pin 2.2 - Fan - Clear - OFF | Set- ON

Pin 2.3 - LIGHT - Clear - OFF | Set- ON

Pin 3.0 - Rx - Data In

Pin 3.1 - Tx - Data out

;==================================================

;

;

;

ORG 00000H

;

;

L0000: LJMP L0030

;

RETI

;

NOP

NOP

NOP

NOP

NOP

NOP

NOP

RETI

;

NOP

NOP

NOP

NOP

NOP

NOP

NOP

RETI

;

NOP

NOP

NOP

NOP

NOP

NOP

NOP

RETI

;

NOP

NOP

NOP

NOP

NOP

NOP

NOP

JNB RI,L002B

NOP

LJMP L0250

;

NOP

;

L002B: RETI

;

NOP

NOP

NOP

NOP

;

L0030: NOP

NOP

NOP

NOP

MOV P1,#0FFH

MOV P0,#000H

CLR P2.0

CLR P2.1

CLR P2.2

CLR P2.3

NOP

NOP

CLR A

MOV R0,A

MOV R1,A

MOV R2,A

MOV R3,A

MOV R4,A

MOV R5,A

MOV R6,A

MOV R7,A

MOV A,PCON

SETB ACC.7

MOV PCON,A

NOP

MOV TMOD,#025H

MOV SCON,#050H

MOV TCON,#050H

MOV TH1,#0FAH

CLR RI

CLR TI

SETB P2.0

SETB P2.1

MOV R6,#001H

LCALL L0150

NOP

CLR P2.0

CLR P2.1

MOV R6,#001H

LCALL L0150

NOP

NOP

NOP

NOP

NOP

NOP

NOP

SETB P2.0

SETB P2.1

MOV R6,#001H

LCALL L0150

NOP

NOP

CLR P2.0

CLR P2.1

MOV R6,#001H

LCALL L0150

NOP

SETB P2.0

SETB P2.1

MOV R6,#001H

LCALL L0150

NOP

CLR P2.0

CLR P2.1

MOV R6,#001H

LCALL L0150

NOP

SETB P2.0

SETB P2.1

MOV R6,#001H

LCALL L0150

NOP

CLR P2.0

CLR P2.1

MOV R6,#001H

LCALL L0150

NOP

NOP

MOV IE,#090H

NOP

LJMP L0300

;

;

L0100: CLR RI

CLR TI

MOV SBUF,ACC

;

L0107: DJNZ B,L0107

;

L010A: DJNZ B,L010A

;

L010D: DJNZ B,L010D

;

L0110: DJNZ B,L0110

;

L0113: DJNZ B,L0113

;

L0116: DJNZ B,L0116

NOP

RET

;

NOP

;

L0150: MOV A,R6

CJNE A,#000H,L0159

NOP

NOP

RET

;

NOP

NOP

;

L0159: NOP

NOP

CLR A

MOV R7,A

CLR A

;

L0163: MOV A,R7

CJNE A,#0FFH,L016E

NOP

DEC R6

LJMP L0150

;

NOP

NOP

;

L016E: NOP

NOP

MOV B,#000H

;

L0173: DJNZ B,L0173

;

L0176: DJNZ B,L0176

;

L0179: DJNZ B,L0179

;

L017C: DJNZ B,L017C

NOP

;

L0180: DJNZ B,L0180

;

L0183: DJNZ B,L0183

;

L0186: DJNZ B,L0186

;

L0189: DJNZ B,L0189

INC R7

LJMP L0163

;

NOP

NOP

NOP

;

L0250: CLR RI

CLR TI

MOV ACC,SBUF

MOV R1,A

NOP

NOP

CJNE R1,#031H,L0268

NOP

SETB P2.2

NOP

NOP

;

L0268: CJNE R1,#032H,L0270

NOP

CLR P2.2

NOP

NOP

;

L0270: CJNE R1,#033H,L0278

NOP

SETB P2.3

NOP

NOP

;

L0278: CJNE R1,#034H,L0280

NOP

CLR P2.3

NOP

NOP

;

L0280: RETI

;

NOP

;

L0300: CLR P3.2

CLR P3.3

CLR P3.4

SETB P2.0

CLR P2.1

MOV IE,#090H

NOP

NOP

NOP

SETB P3.5

MOV R6,#001H

LCALL L0150

CLR P3.5

MOV R6,#001H

LCALL L0150

NOP

NOP

MOV A,P1

MOV R0,A

MOV DPTR,#L0600

MOV A,R0

MOVC A,@A+DPTR

MOV 020H,A

NOP

MOV DPTR,#L0700

MOV A,R0

MOVC A,@A+DPTR

MOV 021H,A

NOP

MOV DPTR,#L0800

MOV A,R0

MOVC A,@A+DPTR

MOV 022H,A

NOP

CLR P3.2

CLR P3.3

SETB P3.4

CLR P2.0

SETB P2.1

NOP

SETB P3.5

MOV R6,#001H

LCALL L0150

CLR P3.5

MOV R6,#001H

LCALL L0150

NOP

NOP

MOV A,P1

MOV R0,A

MOV DPTR,#L0600

MOV A,R0

MOVC A,@A+DPTR

MOV 023H,A

NOP

NOP

NOP

NOP

NOP

MOV DPTR,#L0700

MOV A,R0

MOVC A,@A+DPTR

MOV 024H,A

NOP

MOV DPTR,#L0800

MOV A,R0

MOVC A,@A+DPTR

MOV 025H,A

NOP

CLR P3.2

SETB P3.3

CLR P3.4

SETB P2.0

CLR P2.1

NOP

SETB P3.5

MOV R6,#001H

LCALL L0150

CLR P3.5

MOV R6,#001H

LCALL L0150

NOP

NOP

MOV A,P1

MOV R0,A

MOV DPTR,#L0600

MOV A,R0

MOVC A,@A+DPTR

MOV 026H,A

NOP

MOV DPTR,#L0700

MOV A,R0

MOVC A,@A+DPTR

MOV 027H,A

NOP

MOV DPTR,#L0800

MOV A,R0

MOVC A,@A+DPTR

MOV 028H,A

NOP

CLR P3.2

SETB P3.3

SETB P3.4

CLR P2.0

SETB P2.1

SETB P3.5

MOV R6,#001H

LCALL L0150

CLR P3.5

MOV R6,#001H

LCALL L0150

NOP

NOP

MOV A,P1

MOV R0,A

MOV DPTR,#L0600

MOV A,R0

MOVC A,@A+DPTR

MOV 029H,A

NOP

MOV DPTR,#L0700

MOV A,R0

MOVC A,@A+DPTR

MOV 02AH,A

NOP

MOV DPTR,#L0800

MOV A,R0

MOVC A,@A+DPTR

MOV 02BH,A

NOP

SETB P3.2

CLR P3.3

CLR P3.4

SETB P2.0

CLR P2.1

NOP

SETB P3.5

MOV R6,#001H

LCALL L0150

CLR P3.5

MOV R6,#001H

LCALL L0150

NOP

NOP

MOV A,P1

MOV R0,A

MOV DPTR,#L0600

MOV A,R0

MOVC A,@A+DPTR

MOV 02CH,A

NOP

MOV DPTR,#L0700

MOV A,R0

MOVC A,@A+DPTR

MOV 02DH,A

NOP

MOV DPTR,#L0800

MOV A,R0

MOVC A,@A+DPTR

MOV 02EH,A

NOP

SETB P2.0

SETB P2.1

NOP

MOV A,#053H

LCALL L0100

NOP

MOV A,020H

LCALL L0100

MOV A,021H

LCALL L0100

MOV A,022H

LCALL L0100

NOP

MOV A,023H

LCALL L0100

MOV A,024H

LCALL L0100

MOV A,025H

LCALL L0100

NOP

MOV A,026H

LCALL L0100

MOV A,027H

LCALL L0100

MOV A,028H

LCALL L0100

NOP

MOV A,029H

LCALL L0100

MOV A,02AH

LCALL L0100

MOV A,02BH

LCALL L0100

NOP

MOV A,02CH

LCALL L0100

MOV A,02DH

LCALL L0100

MOV A,02EH

LCALL L0100

NOP

MOV A,#045H

LCALL L0100

MOV R6,#001H

LCALL L0150

NOP

NOP

CLR P2.0

CLR P2.1

MOV R6,#001H

LCALL L0150

NOP

LJMP L0300

;

NOP

LJMP L0000

;

;

; Unresolved Address Reference list

; Data table for HEX values

; 0600 - 30303030303030.(100 times)...3131313131..(100 times)....32323232...(55 times)

; 0700 - 303030..(10 times)31313131....(10 times) 3232323232...(10 times)....till 255 counts

; 0800 - 30313233343536373839..repeat till 255 counts.

;

L0600: EQU 00600H

L0700: EQU 00700H

L0800: EQU 00800H

;

;

END

**CHAPTER VIII**

**CONCLUSION**

Since electricity bills are being the biggest nightmare due to the revised slabs, the Smart Electricity Analysis system helps us to have a greater control over electricity consumption thereby saving energy and money. It aims to provide an energy efficient future to the following generations.

**FUTURE WORKS**

The above said implementation can be extended to various other home appliances with MMS in our mobile phones giving the per device images. The entire project can be implemented and all switch boards can be replaced by this SEMS kit which is cost effective and can also be implemented wireless.

**IX. REFERENCES**

[1] **Households electricity consumption analysis with data mining techniques**

Usman Ali Concettina BuccellaA Carlo Cecati

IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society year: 2016

[2] **Electricity consumption analysis and power quality monitoring in commercial buildings**

Imre Drovar; Jaan Niitsoo; Argo Rosin; Jako Kilter; Ivo Palu

2012 Electric Power Qualiy and Supply Reliability year: 2016

[3]**Electricity-Metering in a Connected World: Virtual Sensors for Estimating the Electricity Consumption of IoT Appliances**

Frank Englert; Patrick Lieser; Alaa Alhamoud; Doreen Boehnstedt;Ralf Steinmetz

2015 3rd International Conference on Future Internet of Things and Cloud year: 2015

[4]**Analysis ofthe residential, commercial and industrial electricity consumption**

John D. Hobby; Gabriel H. Tucci

2011 IEEE PES Innovative Smart Grid Technologies year: 2015

[5]**An Intelligent Energy Management Scheme with Monitoring and Scheduling Approach for IoT Applications in Smart Home**

Tui-Yi Yang Chu-Sing Yang Tien-Wen Sung

2015 Third International Conference on Robot, Vision and Signal Processing (RVSP) year: 2015