



Development of Arduino Based IoT Metering System for On-Demand Energy Monitoring

K. C. Okafor^{1*}, G. C. Ononiwu², U. Precious³, A.C Godis⁴

^{1&2}Dept. of Mechatronics Engineering, Federal University of Technology, Owerri, Nigeria

^{3&4}Dept. of Electrical and Electronic Engineering, Federal University of Technology, Owerri, Nigeria

*Corresponding Author's E-mail: kennedy.okafor@futo.edu.ng

Abstract

Intelligent energy management strategy for cost optimization offers a new paradigm for energy users in Nigeria. This research developed Arduino based energy consumption rate meter for residential homes using multidisciplinary concepts in Mechatronics. The system features real time demand side management using composite design methodology (CDM). It comprises the metering and cloud server cluster units. The work introduced ACS712 Hall Effect current sensor, Arduino Uno (with ATmega328 chipset), and SIM800L GSM modules to achieve the system functionalities. The design description on editor run-time environment enabled direct debugging in the open source Integrated Development Environment (IDE). The approach was evaluated through selected case studies and usability experiments. With the latter, the suitability of the system provided an efficient means of monitoring energy consumption with minimal errors. The results showed that when the meter was switched-ON with no load, an output of **0.00kWh** was read. By connecting i-core7 Dell Inspiron laptop to it, an output of **10.75kWh** was then observed on the cloud web application (real time). The output kept increasing whilst the load was still connected to the meter as expected. Utilities can smartly provide value-added services using the system, thereby increasing their revenues while ensuring customers satisfaction.

Keywords: Internet of Things, Cloud Computing, Smart Meter, Demand Side Management, Utility

1. Introduction

1.1. Background

Internet of Things and Everything (IoT/E) is a relatively new concept that explains the aggregation of objects, systems, processes etc, via an IPv6 backbone into the Internet environment [1], [2]. In the existing power utility setup, an IoT framework can be used to improve the problems faced by energy consumers on daily basis. For instance, consumers are presented with consumed energy details only once a month with zero control on the deployed smart meter (if in existence). The length of time between updates on power usage is far too long for a consumer to observe instantaneous changes in the power usage. In addition, utility bills can be convoluted on how they present usage information, but a consumer may not be able to decipher changes in their power usage from the last bill. In most cases, if any consumer fails to pay the bill, the utility field marketers goes to their houses to disconnect the power supply [3]. Conventional energy meters presently installed in households only display total real time usage of its power and the amount of electricity available. There is no way to access the details on what the day's, week's or month's consumption was on these meters. Often, these meters are placed in an inconvenient location which makes regular viewing somewhat difficult. Sadly, ubiquitous access to consumption rate and energy profile by

consumers is impracticable. By deploying a sensing/monitoring mechanism in real time (using IoT), the exact power consumed can be ascertained while deriving informed decisions on how to manage the connected loads. A system that can give users an estimate of how much energy is being used, that has been used, or that might be consumed will allow them to adjust their habits and lower the costs [4]. This is a form of Demand Side Management discussed in a previous study, i.e., R-SGEMS.

Using Arduino module to achieving real time high performance computing (RTHPC) for smart metering application is novel concept. This has benefits for transaction processing and data mining in cloud computing environment. Arduino based wireless meter gives more control to both utilities and consumers by providing them detailed information about power consumption on real time basis. The meter send readings such as power (kWh), voltage (v), current (I), cost billing (Naira/USD), etc. to a central cloud server. This then stores the information in its database for predictive analysis while allowing both power consumed and bill generated to be viewed by consumers remotely using any internet-ready device. Data can be collected at any desired time interval such as hourly, daily, weekly, or monthly basis. Since, there is no human intervention in the entire process, this eliminates human error and corruption. This type of remote meter can be used in residential apartments, especially for industrial consumers where bulk energy is consumed [5]. The novelty of this work is based on the realization and integration of a Composite Design methodology for smart metering and cloud interfacing.

This paper focused on smart system that wirelessly profiles energy consumption by calculating the power consumed by the individual consumers, transmits the calculated rate to a cloud web server so that any valid consumer can view individual consumption rate remotely and enforce demand side management. This system can be adapted in on-grid or off-grid utilities. A simple solar off-grid pure sine wave Inverter or utility can be used as Alternating Current (AC) source. The materials used include: Arduino Uno microcontroller for saving metering data and transmitting to the web server; a liquid crystal display (LCD), a matrix keypad, a GSM module, a current sensor and a designed website application with database. The integration concepts involve electronics, control, communication, computer and mechanical engineering. These are the elemental concepts of mechatronics engineering research project.

1.2. Research Contributions/Benefits

The contribution of this paper is to develop an IoT wireless energy meter prototype using open-source Arduino Microcontroller (ATmega328P) which facilitates sensing and actuation of controls from ubiquitous sources/devices. This work will benefit utilities, retail providers and customers. Such benefits will reflect on the increased efficiencies and reduced labor cost as a result of automatic readings, connections and disconnections. With timely usage information available to the customer, the benefits will be seen through the opportunities offered to manage their energy consumption. Some other relevance of the work include: accurate meter reading eliminating estimate billing, improved billing, less accrued expenditure, improved billing and tacking of usage. The rest of the paper is organized as follows. Section 2 discussed relevant studies associated with smart metering. Section 3 presents the materials and methods. It discussed the various components used and exposes the methods that were implemented in this work using block diagram, it also highlighted the design consideration, specification and the technology used. Section 4 discussed the results obtained after the final testing. Chapter 5 presented the validation discussions. Finally, a conclusion and summary of the work as well as recommendations for future research were presented in Section 6.

2. Related Research Efforts

2.1. Smart Meters/Advanced Metering Infrastructure

Currently, there are several products already in the market for power monitoring purposes [4]. For instance, Automatic Meter Reading System (AMR) has been developed in [6] which offer remote collection of consumption data from customers' utility and process the data to generate the bill. The

system proposed in [5] improves on the model in [6] because the technology sends energy consumption data from buildings, factories and houses to the utilities for load curve, power quality analysis and consumers' billing purposes. The work in [7] discussed existing smart meters such as Tendril Insight, Energy Controls -Three Phase Smart Meter, GEO – Minim Energy Monitor, Watson Smart Meter, Tendril Vision Smart Meter, and Alert Me Smart Energy.

The work in [8] discussed energy metering system which extends the capacity of previous Advanced Metering Infrastructure (AMI) or Automatic Meter Reading (AMR) [9], [10], [11], [12] and [13]. Several works on AMI or smart metering leads to Fig. 1 which a typical smart meter used in Nigeria. This captures the energy consumption in kWh only.



Figure 1: A Typical Smart Meter [13]

The only related work on IoT smart metering was found in [14]. The work focused literal discussions on concept of smart meters, different types of communication, communication employing PLC, security and frauds detection, cost optimization and data sets. Considering technology imperatives of IoT, it is perceived that as the electrical grid migrates from analog to digital, the existing smart metering systems will require built-in, secure, interconnected intelligence. On a large scale, a supporting cloud and network infrastructure will help protect data, manage devices and perform real time data analytics.

2.1. Research Gaps

From the literature review, similar work by Rastogi IoT Smart meter [14] have major setback. The system have no provision for cloud based web application integration optimized for consumers to access their consumption rate and enforce DSM. In most works, the SMS application was a substitute, but this is unreliable particularly when the network operator's Key Performance Index (KPI) is below its expected threshold/index. A real time IoT based metering system that does not depend on SMS communication, but uses a data driven based web application is novel. This is more efficient and offers a very cost effective model that satisfies the aspirations of energy users in Nigeria.

3. Research Methodology

Composite Design Methodology (CDM) was used to achieve the system by modularizing the system into front-end metering and back end cloud interfacing. This is a time to market approach of bringing two extreme systems into a unified model. This comprises two subsystems (the smart meter and the cloud server) linked with the IoT framework. Using the CDM, the following system design phases as found in [19], [20] were considered: design objectives, consideration, specification, technologies, programming techniques, and hardware design. The materials and methods used in CDM are briefly presented below.

3.1. Materials

There are different materials used in this research. This section briefly highlights the different materials used to achieve the project.

i. IoT Arduino Chipset Target Device

Arduino UNO shown in Fig. 2 was used as an open-source platform in the CDM. This consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE that runs on the testbed computer used to write and upload design codes to the physical board. This single board microcontroller was easily adapted to IoT smart metering system by using the IDE to generate a simplified version of C++ code for implementation. The Arduino Uno adopted is based on the ATmega328 chipset in Fig. 2. It is made up of 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. The Uno differs from all preceding boards in that it is programmed as a USB-to-serial converter.

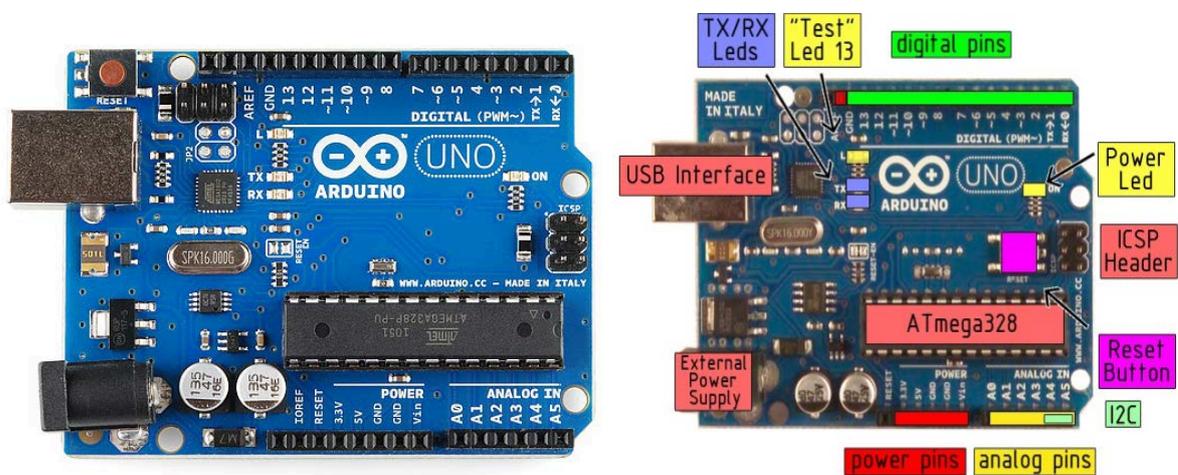


Figure 2: Arduino UNO Chipset (Source: Data sheet.pdf, 2016) [15]

Arduino Uno was used as the heart of the system controller unit. Once actuated, it receives signals from all sensors, processes these signals and saves the value while sending the processed information or command out when necessary.

ii. Current Sensor (ACS 712)

This is the transducer used for sensing and monitoring of current flow. In essence, Allegro ACS712 current sensor was used owing to its high accuracy. It senses AC and DC currents based on Hall-effect principle illustrated in Fig. 3. The ACS712 Current Sensor is a Hall Effect current sensor that accurately measure current when induced. The magnetic field around the AC wire is detected which gives the equivalent analog output voltage. Since, the actual current reading is needed, the analog voltage is sent to the Arduino for processing.

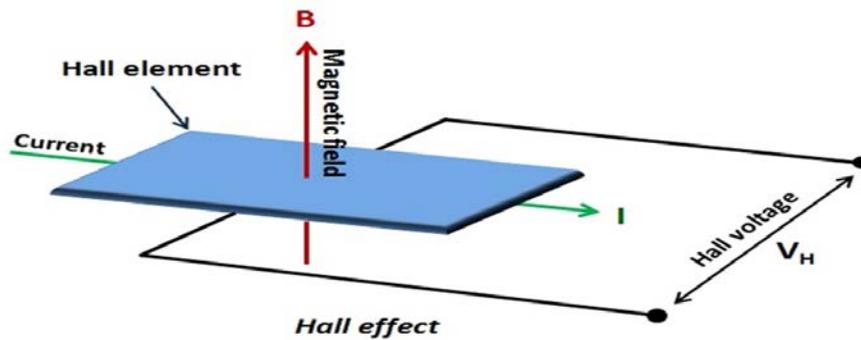


Figure 3: Principle of Hall-effect [16]

iii. Communication GSM Module Device

Communication with Arduino board was established using SIM800L GSM module (as shown in Fig. 4) via the AT command. In context, the module has a large library with object oriented supports for communication with the module. It is a complete Quad-band GSM/GPRS solution in a Surface Mount Technology (SMT) type which is embedded in the customer meter device. The SIM800 support Quad-band 850/900/1800/1900MHz, making it adaptable for transmitting voice, SMS and data information with lower power consumption. Its tiny size (24*24*3mm) makes it very portable for the design packaging. Besides, it also has Bluetooth and embedded AT which allows for total cost savings and fast time-to-market while supporting TCP/UDP and HTTP protocols through a GPRS connection. In order to interface it with the cellular network, the board uses a SIM card provided by a network operator.

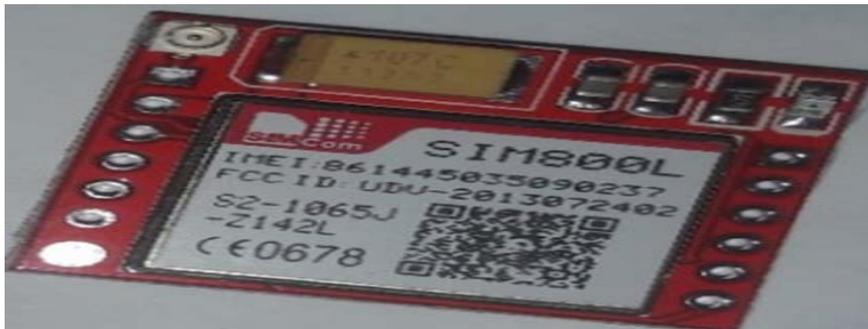


Figure 4: SIM 800 L GSM Module

For active power supply, AMS1117 voltage regulator was used to supply +12V and +5V respectively in the subsystems of the IoT (Arduino board) metering system. A crystal of 12MHz was used in the design. The transistors BJT and MOSFET, C1815-Y3G used for switching the disconnection relay ON or OFF since the relays cannot be switched directly from the microcontroller. Other components introduced in the design include: electrolytic Capacitor, Diodes, LEDs, relays and Bridge Rectifiers, Liquid Crystal Display (LM016L) based on 7-segment display which serves as a digital clock. Resistors (Fixed and Potentiometer), Switch Mode Power Supply are used. The Arduino communicates to the LCDs via the four data lines using the digital to display what is required of it.

Apart from these lines, there is also an enable pin, RS pin and a RW pin. The backlight on the LCD is activated when 5V is supplied from the Arduino to a pin on the LCD and by grounding another pin. A potentiometer was used to adjust the contrast of the LCD.

Fig. 5 depicts the system architecture showing the smart meter and the cloud server communicating with the IPv6 on HTTP POST and GET REQUEST procedures. The method is presented later in this section.

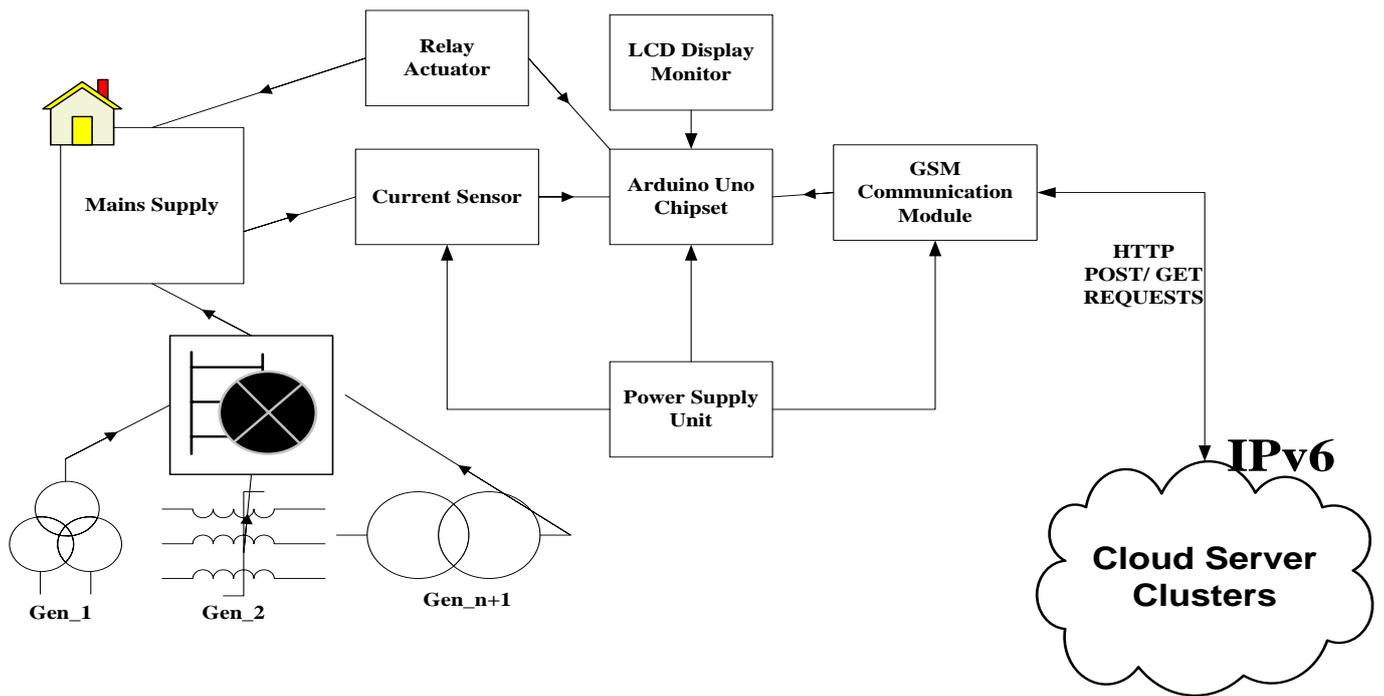


Figure 5: IoT Global System Architecture for Smart Metering

3.2. Method

In the CDM, a functional description of the IoT based energy meter as well as the cloud integration (as illustrated in Fig. 5) was achieved through a high level integration. Since the proposed system is to be used in the consumer power system end, the following were put into consideration in realizing the system, viz: cost, method of sensing and signal conditioning, reliability and performance efficiency of components. Also, the selection of system sensors were based on factors viz: sensitivity, accuracy, precision and accuracy. Table 1 show the specification used in this paper. The choice of the design module was informed by the availability of materials at the time of design.

Table 1: Design modules and specifications of the IoT metering system

DESIGN MODULE	DESIGN SPECIFICATION
Controller board	Arduino Uno Atmega328P
Display	LM016L LCD
Keypad	4 x 4 Keypad
Sensors	Current sensor: ACS712 hall effect sensor
Actuators	SPDT 12V coil relay
Timer	12MHz crystal oscillator
Communication	SIM 800L GSM module
Input Voltage (recommended)	7-12V
DC Current for 3.3V Pin	50 mA
DC Current per I/O Pin	20 mA
PWM Digital I/O Pins	6

The focused method in this section is on the software programming approach and hardware design integration discussed below.

3.2.1. Programming Technique

To achieve the IoT metering depicted in Fig.5, C++ was used as the programming language. This was used to realize the communication between the Arduino Uno, GSM Module and the Cloud server. For communication between the Arduino and GSM module, the Arduino accesses the preinstalled GSM library using the 'AT commands' so as to effectively communicate with the GSM module. The microcontroller was programmed using Arduino C++. Essentially, the Arduino software consists mainly of two parts: the Integrated Development Environment and a core library. The development environment is a complete source code editor, serving as a cross platform application. Code management, editing, compilation as well as code script uploading is carried out via the IDE which has a serial monitor. This enables a user to send and also receive data from the board. Programs are written in the development environment of the Arduino and then loaded to the Arduino board via a USB in-circuit reprogrammer. Also, the AT commands were used to control the interface modem. In this paper, the AT commands were used for GPRS communication using the SIM800.

3.2.2 Hardware Design

As depicted in Fig. 5, the hardware design comprises of all the necessary design procedure, block diagram(s), theory, and design calculations of the basic subsystems/components that make up the IoT based smart meter. As shown in Fig.5, the system architecture has supports for distributed generation systems denoted by Gen_1, Gen_2 and Gen_{N+1}. On the cloud, IPv6 is used for the HTTP POST and GET REQUEST routines. For experimental purposes, the main supply consists of a normal wire connected to a wall socket in a building which passes through the actuators and current sensor in the meter to a socket outlet. The initial load was a 60watt bulb and an electronic device which was connected to the socket outlet.

During the implementation, a DPDT AC relay was interfaced between the Arduino and the main supply. Because, the 12V relay cannot be directly connected to the 5V Arduino; a switching transistors (described above) was used to interface the both of them. Pin 12 on the Arduino was connected to the base of the transistor, the collector was connected to the 12V relay and the emitter was connected to GND. The COM pin is supposed to be connected to the main supply while the NO (Normally open) is supposed to be connected to the output terminal e.g. the distribution box, of which in this project it is the socket outlet where a device would be connected. The relay was basically used to switch the meter off and on depending on if a customer/user has exceeded the budget or not.

- Power Supply Unit Interfacing

In Nigeria context, the Alternating Current (AC) and voltages supplied to the consumer end by the main supply was stepped down to a minimum voltage via the power supply unit which in this case is in a form of an adapter. The adapter steps down the 220V supplied from the mains to a minimum of 12V. The voltage regulator was used to further step down the 12V to a minimum of 5V.

- Current Sensor Interfacing

To determine the amount of current that is drawn by the load or heating element, a 30A ACS712 Hall Effect current sensor was used. This Hall Effect sensor isolates the current that is supplied by the main supply from damaging the Arduino, since it can allow only 5V input voltage. The ACS was interfaced with the Arduino Uno and the main supply. For the ACS712; the pin 5 or GND, pin 7 or Vout, and pin 8 or V_{cc} were considered in developing of the system. Therefore for the connection configuration, the ACS712 output pin was connected to the analog pin A₀ of the Arduino, the GND pin also was connected to the GND of the Arduino, and then the V_{cc} to the 5V pin on the Arduino board.

The sensor outputs an analog output voltage that corresponds to the current flow which ranges from 0-5V. Fig. 6a and 6b illustrates the pin connection configuration of the ACS712 connected to the Arduino and an image of what the connection looks like respectively.

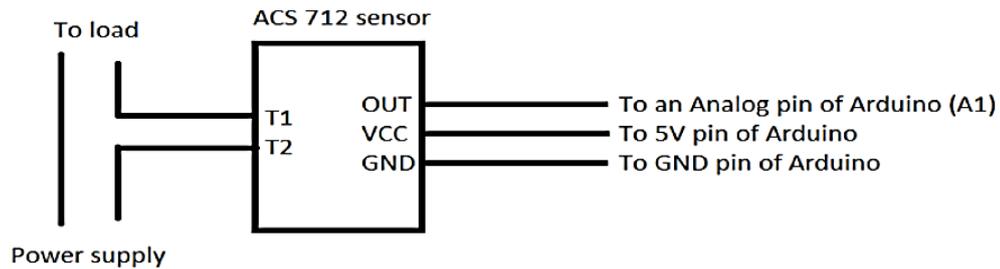


Figure 6a: Pin Connection Configuration [15]

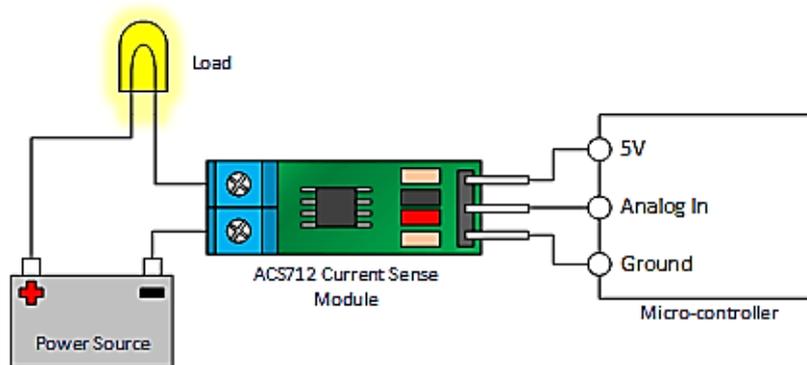


Figure 6b: Connection Configuration [15]

- Arduino Interfacing

The Arduino module was interfaced with the following components : ACS712, SIM800L, the LCD, and relay as depicted in Fig. 5. It receives inputs from these components and then generates the desired output to the connected components. It performs the following functions:

- Converts the Analog output voltage from the ACS712 to an equivalent dc current value ranging from 0-30A with the aid of an on board Analog to digital converter.
- Scans the keypad for input. This input is the pin code for subscription specific to a customer. For payments made online, it gets the input via the GSM module discussed in this section.
- Controls the relay via a transistor.
- Sends and receives messages from the GSM module while monitoring the output displayed by the LCD.

3.2.3. Load Consumption Rate Calculation

In this work, few selected loads were considered in the experimentation, while using ohms-law to generate the exact power demand and cost details. Now, assuming a cloth iron plugged gave an output analog voltage = $1.8181V$, the equivalent current output = $10.9090A$.

- Case A: Electric Iron

If current (I) = $10.9090A$ and Voltage (V) = $220V$
 Then Power $P = I * V = 10.9090 \times 220 = 2399.98 \approx 2400W$
 Therefore, Power generated = $P = 2400W$

To get the energy consumed, this it will be

$$P = P * T / 1000 = kWh$$

Using 1hr, this will give

$$P = 2400 * 1 / 1000 = 2.4kWh \quad (1)$$

The microcontroller uses the energy consumption information to calculate the cost. This was achieved by uploading the code scripts on the microprocessor. Assuming different appliances are generating power, the cost can be calculated. Thus, using a clothing iron, electric kettle, fan, refrigerator and washing machine, energy consumed for an hour and the cost incurred can be calculated based on assumption and using the formulas stated above as follows;

- **Case B: Electric kettle**

Analog output voltage $V = 1.5151V$

Equivalent current output $I = 9.0909A$

Voltage $V = 220V$

Power = $P = 9.0909 * 220 = 1999.98 \approx 2000W$

$$\text{Energy Consumed for 1hr} = 2000 * 1 / 1000 = 2kWh \quad (2)$$

- **Case C: Fan**

Analog output voltage $V = 0.0530V$

Equivalent current output $I = 0.3181A$

Voltage $V = 220V$

Power = $P = 0.3181 * 220 = 69.982 \approx 70W$

$$\text{Energy Consumed for 1hr} = 70 * 1 / 1000 = 0.07kWh \quad (3)$$

- **Case D: Refrigerator**

Analog output voltage $V = 0.1515V$

Equivalent current output $I = 0.9090A$

Voltage $V = 220V$

Power = $P = 0.9090 * 220 = 199.98 \approx 200W$

$$\text{Energy Consumed for 1hr} = 200 * 1 / 1000 = 0.02kWh \quad (4)$$

- **Case E: Washing Machine**

This was computed in a similar way with electric kettle,

therefore energy consumed (for 1hr) = $2kWh$ (5)

Therefore, total energy consumed (for 1hr) from Equ 1-5,

$$T_e = 2.4 + 2 + 0.07 + 0.2 + 2 = 6.67kWh \quad (6)$$

To ascertain the cost implication using the Nigeria context, an assumption made. The current tariff rate for residential areas according to the Nigerian Electricity Regulatory Commission is N23.60 [17].

Assuming $1kWh = \text{N}23.60K$.

Therefore, for $6.67kWh$ the cost incurred will be = $6.67 * 23.60 = \text{N}157.412$.

These computation scenarios have been validated in the prototype tasetbed. Regardless of the case, the energy consumption is sent to the display LCD while the energy consumption and the amount incurred is sent the cloud server via a GSM module. This information is sent from the coded microcontroller. When a consumer exceeds his payment, a signal is sent from the same coded microcontroller to the relay to shut off the meter from the main supply automatically. It remains so until the consumer subscribes by inputting a pin code. This pin code is scanned by the Arduino, and

then a signal is sent to the relay in other for the relay to turn on the meter. C++ programming language was used in writing the codes that were used in programming the microcontroller. It was used because of its simplicity. The diagram in Fig. 7 shows how all the components were interfaced to the Arduino.

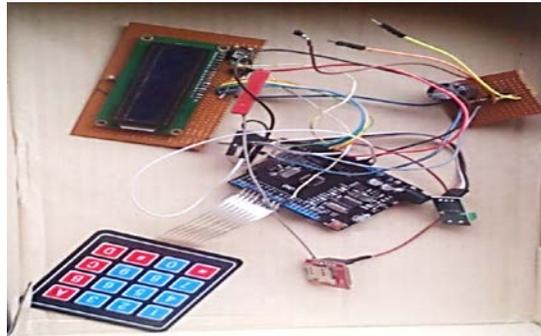


Figure 7: Testbed Components Interfacing With the Arduino UNO

As described previously, the SIM800 was used to provide a physical link between the Arduino and the cloud server for communication. Basically, it was configured to carry out the following functions:

- Sends information about the consumption rate from the Arduino to the cloud server for a consumer to view via a web application
- Sends pin code for subscription form the cloud server to the Arduino for processing, peradventure a customer subscribes online.
- Sends information to the Arduino if a consumer exceeds the budget for the month.

Out the 7pins for SIM800, only 4 pins were used to achieve Fig. 8. These include: Vcc, the GND, the TXD, and the RXD. The SIM800 was interfaced with the Arduino and the pin map between SIM800 and Arduino is summarized below;

- SIM800 5v/4v ↔ Arduino 5v
- SIM800 GND ↔ Arduino GND
- SIM800 SIM TXD ↔ Arduino D8
- SIM800 SIM RXD ↔ Arduino D7

The SIM800 module itself uses input voltage of 3.7V-4.2V. Therefore, a direct connection of the Vcc pin to Arduino 5V pin might damage the module as well. Thus the Arduino was connected to a diode before connecting it to the SIM800. This was used to step down the input voltage by 0.7V (i.e. from 5.0V to 4.3V) thus powering the GSM module with an input voltage of 4.3V (tolerance of 0.1). The Arduino transmits using 5V, while the SIM800 receiver port works with 3V, therefore a potential divider is used to slash the voltage from 5V-3.3V (tolerance of 0.3). Fig. 8 illustrates the connection of the Arduino to the SIM800.

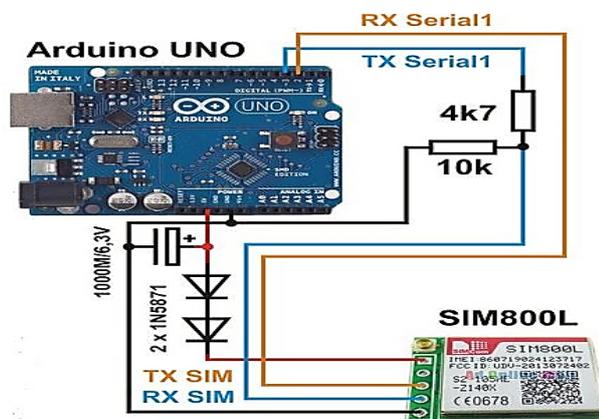


Figure 8: Connection of the Arduino Uno to the SIM800L GSM Module.

The SIM800 was able to communicate with the cloud server via AT commands. The Code script was uploaded into the microprocessor chip alongside the AT commands to establish communication. Information from the cloud server is gotten by sending a HTTP GET request. This was realized using Algorithm I shown below.

Algorithm I: Cloud Interfacing

Begin ()

- i. *Initiate the HTTP service ()*
- ii. *Set the HTTP session ()*
- iii. *Set the HTTP URL ()*
- iv. *Start the session ()*

End;

The HTTP POST request was used to send the current and power reading, as well as the energy consumption rate to the cloud server. The HTTP requests make it possible for the SIM800 to communicate with the cloud server. A valid subscriber identification module (SIM) card was inserted into SIM slot on the GSM module. The SIM card was used to purchase airtime and obtain data bundle for internet connection between the module and the cloud. This also interfaces with the cellular network. On the cloud Server, HTML/PHP scripts was used to design the web application for API communication. Figure 4.3b shows the realized customer's web page. With the API, this could display energy consumption rate online, in order for the consumers to view from any location. It also enables a customer to turn the meter OFF/ON from anywhere thereby achieving DSM. It can also be used to load the cloud server. Table 2 and Fig. 9 illustrates the schematic diagram of how the ACS712, the SIM800, the LCD and the keypad were interfaced with the Arduino.

Table 2: Sensing and Actuation Algorithm of the IoT Arduino Smart Meter

Algorithm II: IoT Sensing and Actuation for Smart Metering Demand Side Management

Input: Readings (current, I; voltage, v; power, p, power factor pf)

Output: LCD, Http Post, Request, Cloud_Driver

Begin ();

Initialize Arduino setup j == 0;

Initialize GSM Module k == 0;

Call Sensor_Current, I ();

Call Sensor_Voltage, v ();

Calculate_Power, p ();

Write_power p = LCD ();

Send_Readings (HTTP POST) = Cloud_Driver;

Check Status ();

If (Readings) > (Preset condition (payment)) **then** Trigger_Relay ();

Switch_off main_supply;

Else

Return;

Else if;

End

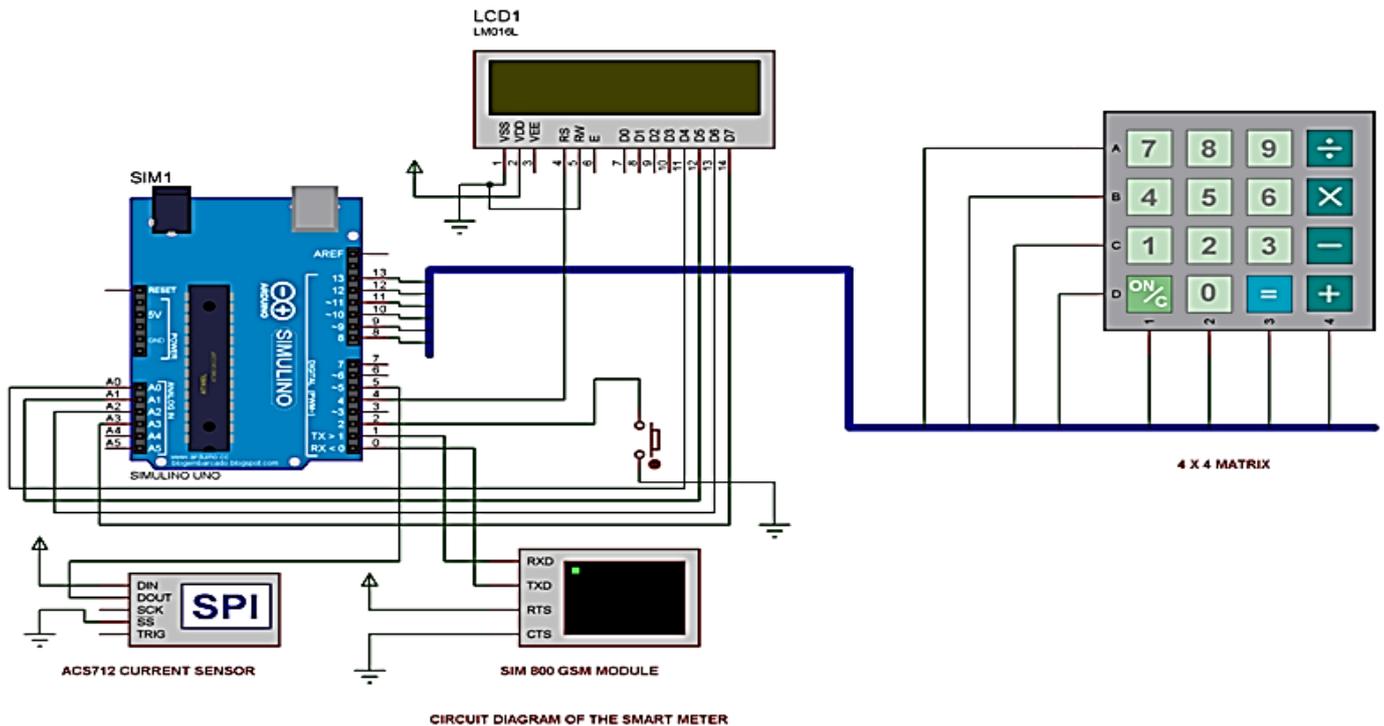


Figure 9: Schematics Diagram of the IoT Smart Meter

4. RESULTS AND ANALYSIS

This section presents the actual results obtained from the completed design. The results from design has been summarized under four stages:

- Programming of the Microcontroller
 - The Arduino Uno which is the heart of the system i.e. the meter controller was programmed using C++ on the Arduino IDE .
 - The program was properly tested and debugged in order to ensure that the communication between the Arduino and other components was achieved.
- Component Testing
 - The components that were used were individually tested to ensure that they were working as required.
 - Each major component that was supposed to be interfaced with the microcontroller was connected and tested to ensure proper communication
- Cloud Script/Web Application Design

The web application was designed using HTML/PHP script. It was then tested to ensure that there was communication between the smart meter and the web server.

- Component Coupling and System Testing

After the programming, configuration of the component, and testing the cloud web application design, the entire system was coupled and cased. The system was finally tested to satisfy requirement specifications. Fig. 10 shows the tested system on a smaller scale, using a bulb as the load and a socket outlet as the main supply unit. This test was done before coupling to ensure that the system is working properly. Before the heating element was plugged to the meter, the reading that was displayed on the LCD was **0.00kwh**. This reading was the default and it simply showed that no current was sensed because the heating element had not been connected to it.

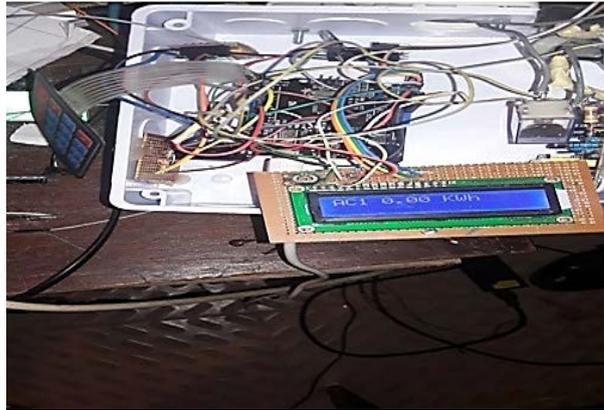


Figure 10: Initialization of Prototype IoT Meter Reading for device AC-1

In Figure 11, a heating element (200W bulb) was later connected to the meter to see if the power consumed by the heating element can be read by the meter and transmitted. When this was done, a reading of **9.35kWh** was observed and the value kept increasing as the heating element remained on the metering system..



Figure 11: testbed Meter reading with 200W bulb

With time, the value of the energy consumed increased to **10.75kWh** and the value kept increasing as shown in Fig 11.



Figure 12 : Testbed Incremental IoT meter reading.

Fig. 13 shows the coupled system after carrying out the final testing. The coupled system in Fig.13 captured with a laptop (1562 Dell inspiron) as the load, while Fig. 14 shows the web page display for read data, cost and pin.



Figure 13: Prototype/coupled System

The prototype in Fig. 13 involves mechanical casing design. The system addressed the problems of the present smart meters (prone to real time errors and inflexibility issues). As a cloud based meter, this offers little or no human and processing errors. A cloud web application in Fig. 14 is designed so that the consumer can easily subscribe and monitor the consumption profile ubiquitously.

The application offers shutdown automation procedure for the consumer such that the meter can be easily shutdown during a period of peak time. This is to minimize the bill that may be incurred. As illustrated in Fig. 5, the Arduino served as the heart of the system and relayed its output to the GSM module which aided communication between the metering section and the cloud server.

Now, the ACS712 Hall Effect current sensor senses current drawn by the load and isolates the current from damaging the Arduino Uno chipset. Signal conditioning was ensured with the Arduino ADC. This output is passed to the GSM module which communicates to the cloud server while in turn receives inputs from the cloud server using the HTTP Post/Get protocol. The consumption rate is displayed both on the web application interface as well as on the LCD of the meter.

From Fig.14, the results obtained after the system was coupled. It showed an increase in the value of the energy consumed with time. From the cloud server application, it was observed that the more load connected to the IoT based meter, the greater the amount of power consumed. These were obtained from remote locations independent of the metering system.

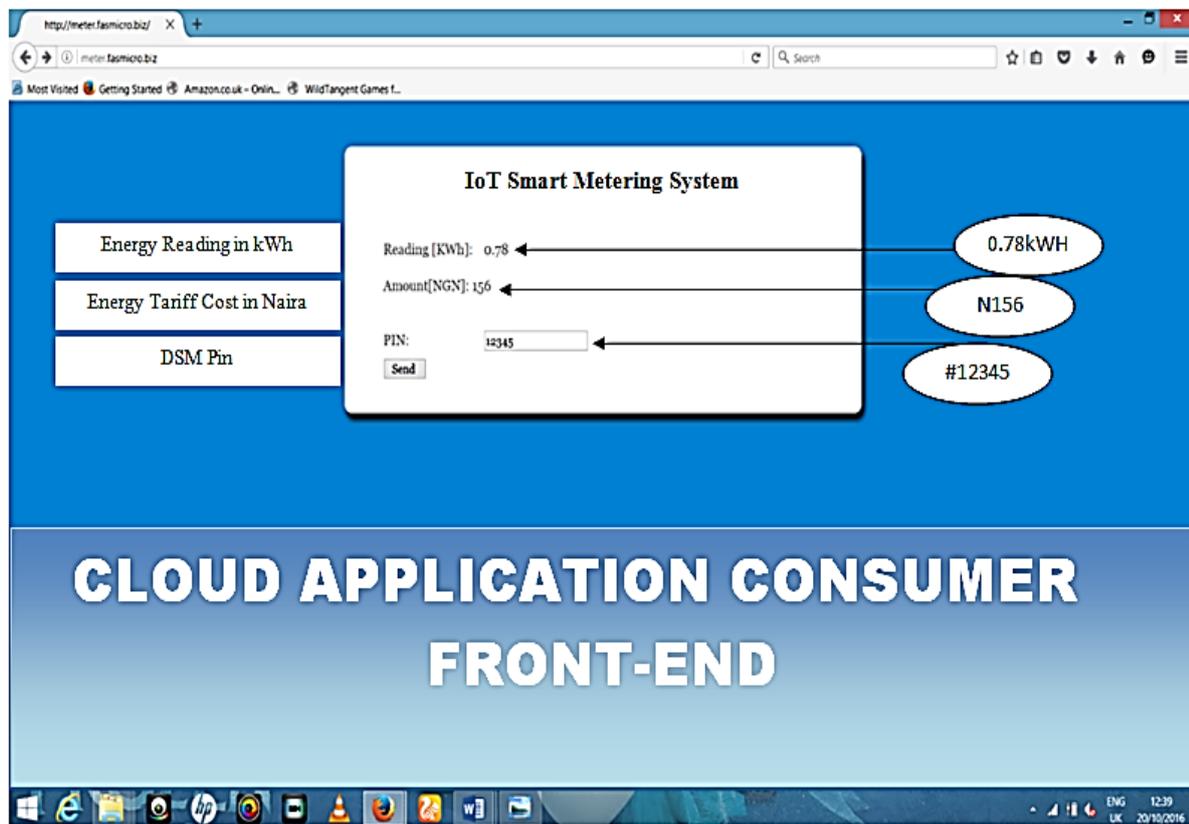


Figure 14: Test Snapshot of user End Cloud Server Web Application.

5. Validation Discussion

From the system architecture shown in Fig. 5, the results of the system tests were obtained. This was obtained through systematic checks and observations using laboratory test tools and equipment. The tests carried out on the design met the expected specifications with negligible deviation or tolerance. From the test results in Fig.11 and 12, the value that was sensed and displayed on the LCD showed that the Arduino and the current sensor were communicating in real time mode. The initial expected value of **0.00kwh** was observed as well as the other values. These were displayed on the LCD for remote transfers. This value was gotten by connecting the meter to a wall socket outlet (which served as the main supply). The value on the LCD kept increasing with time whilst the heating element was plugged to it as shown in Fig. 13. In order to effectively carry out this testing, an hour was represented by a second, and as such all readings in kilowatt-hour, was actually in kilowatt seconds.

To start up the meter, the “#” key on the keypad was used (where both key *A* and *B* were in setting of the energy budget and recharging or subscribing with pins respectively). The microcontroller first processed the information sent to the cloud server as shown in Fig. 5 by calculating the power consumed by the heating element and converting it into an equivalent energy consumption value as shown in Fig. 14. The microcontroller then uses this information to generate the cost incurred at each time interval, which is sent to the cloud server via HTTP Get/Post Request. The meter can also be shut down by sending information to the microcontroller through the GSM module; this then triggers the relay to shutdown the meter. The major limitation observed was that there was a time lag when sending information from the GSM module to the cloud server. This was not expected but it was unavoidable due to poor network. The significance of the obtained results is the IoT smart metering alternative provides a flexible real time DSM support. This offers an alternative way for customers, including utility companies to keep track of energy consumption rate and billing anyhow and anywhere.

Conclusion

This research developed an IoT based smart metering system while analyzing its integration procedure. Using a Composite Design methodology, the work provided a simple and useful solution in the form of an energy consumption rate wireless meter. The system was shown to be successful in measuring current, power consumption and also processing the cost incurred by a customer. These metrics are communicating the energy consumption and cost to the cloud server. This makes it easier for customers to view consumption rate ubiquitously. A comparison with [18] revealed that the developed IoT smart metering satisfied these requirements viz: Quantitative measurement, control and calibration, communication (sending and receiving of data effectively); ability to receive upgrades from firmware, effective power management, display as well as timing synchronization. These are essential between the meter and the utility provider's system. In the work, Demand Side Management concept has been satisfactorily achieved. However, the present research currently does not cover concepts for the validation of the system specification roadmap as presented in [20]. Hence, future research will be carried out to investigate possibilities for an automated validation of the system specification regarding the requirements for extending GSM wireless communication, with WiMax (4G LTE) and optic fiber for long distance communication. Also, introducing an open tampering technique in the IoT smart meter to monitor any form of tampering.

Conflict of Interest: The authors declare that there is no conflict of interest in this publication.

References

- [1] A.V. Dastjerdi, H.Gupta, S.K. Ghoshy, and R.Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments, 2016; 00:1–22. arXiv:1606.02007v1 [cs.DC] 7 Jun 2016.
- [2] A.G Chukwudebe K.C. Okafor, I.E Achumba, O.Gordon, "Leveraging Fog Computing For Scalable IoT Datacenter Using Spine-Leaf Network Topology", *Accepted in Journal of Electrical and Computer Engineering*, ISSN: 2090-0147. Hindawi Publishing Corporation, USA, 2016.
- [3] Christopher McNally, "Arduino Based Wireless Power Meter", p.4, May 2010.
- [4] Chao Chen, Carlos Pomalaza-Raez, Edwin Chobot, Daniel B Newby Renee Kathleen Chandler, Ms. Nusaybah Abu-Mulaweh, " Design of a Wireless Sensor and Actuator Network for Energy Management at Home, In International Journal of Embedded Systems and Applications (IJESA) Vol.3, No.1, March 2013.Pp.1-5. DOI : 10.5121/ijesa.2013.3101
- [5] Kesav H.O, Rahim B.A, "Automated Wireless Meter Reading System for Monitoring and Controlling Power Consumption", *International Journal of Recent Technology and Engineering (IJRTE)*, Vol-1, Issue-2, p.66, June 2012.
- [6] White paper "Smart Meters and Smart Meter Systems: A Metering Industry Perspective" published by Edison Electric Institute, Washington D.C, p.7, March 2011.
- [7] Ruhul .A, Mahmudul .H, Rajib B.R, —Roadmap To Smart Grid Technology: A Review of Smart Information and Communication System||, *International Journal of Control and Automation*, Vol. 7, No.8, Pp. 407-418, 2014
- [8] O.U. Oparaku, Ugwoke F.N, K.C. Okafor, Udeze C.C, Okeke I.O, "A Novel Framework for Cloud Energy Metering System (CEMS) in Solar PV Plant", *African Journal of Computing & ICT*, Vol 7. No. 2, ISSN2006-1781, June 2014.
- [9] Kersting W.H, Philips W.H, "Load Allocation Based on Automatic Meter Reading and control System", *International Journal of Advanced Research in Computer Engineering and Technology*, vol-1, issue-4, June 2012.
- [10] Ahmed. T., Miah. S., Manirus .I., Uddin .R., "Automatic Electric Meter Reading System: A Cost Alternative Report in Meter Reading for Bangladesh Perspective Using Low Cost Digital Watt Meter and Wimax Technology".
- [11] K.C Okafor, Oparaku O.U, Achumba I.E, Ezeh G.N, Chilakpu K.O, "R-SGEMS: A Novel Green Energy Management System for Renewable Energy Utility" 2014.
- [12] M.C. Ndinechi, O.A. Ogungbenro, K.C. Okafor, "Digital Metering System: A Better Alternative for Electromechanical Energy Meter in Nigeria", *International Journal of Academic Research* Vol. 3. No. 5, Pp.189-194, September, 2011.
- [13] Zahid I.R, Muhammed .W, "Automatic Energy Meter Reading Using Smart Energy Meter", *Conference Paper, Research Gate*, Pp. 2-6, March 2014.
- [14] S. Rastogi, M. Sharma, P. Varshney, " Internet of Things based Smart Electricity Meters", In *International Journal of Computer Applications* (0975 – 8887) Vol. 133 – No.8, January 2016.Pp.13-16.
- [15] <http://www.farnell.com/datasheets/1682209.pdf>- ATmega328P (datasheet), Oct.2016
- [16] Hall effect. www.embedded-lab.com, October 2016.
- [17] www.nerc.org, October 2016.
- [18] Silicon Laboratories, Inc. Smart metering brings intelligence and connectivity to utilities, green energy and natural resource management. Rev.1.0. [accessed August, 2013].

- [19] Frank T, Eckert K, Hadlich T, Fay A, Diedrich C, Vogel-Heuser B. Workflow and decision support for the design of distributed automation systems. In industrial informatics (INDIN); 2013. p. 293–9.
- [20] B.Vogel-Heuser, D. Schütz, T.Frank, C. Legat, “Model-driven engineering of Manufacturing Automation Software Projects – A SysML-based approach”, In Mechatronics (Elsevier-ScienceDirect) 24 (2014) 883–897.