2018 International Conference on Multidisciplinary Research
Track 1: STEM (Science, Technology, Engineering & Mathematics)
https://doi.org/10.26803/MyRes.2018.03

Non-Complex Cost-Effective Energy Monitoring using the Internet of Things

Pierre E Hertzog and Arthur J Swart
Central University of Technology
Bloemfontein, South Africa

Abstract
Energy monitoring is vital to identifying ways of reducing energy consumption. This may lead to a reduction in the use of non-renewable energy sources (such as coal), increasing the lifespan of current energy storage systems (such as batteries) and reducing financial costs (such as electrical utility costs). Current commercial monitoring systems may prove too complex or too expensive for the average homeowner to install and use. How can the internet of things, therefore, be used to identify sources of energy consumption? The purpose of this paper is to describe the design and setup of a non-complex cost-effective energy monitoring system for a residential home environment. This may enable homeowners to identify unwanted energy consumption and thereby reduce their electrical utility bills. The system incorporates an Arduino Uno Board, ESP8266 WiFi module and a cloud server called ThingSpeak. No physical wiring connections are needed, as only a current transformer sensor needs to be hooked around the main AC supply line from the electrical utility provider. No PC data storage is required, as the data is stored on the cloud server. The only key requirement is internet connectivity so that the WiFi module may connect directly to the cloud server where all data is captured and presented.

Keywords: Arduino, internet of things, energy.

Introduction
Energy monitoring is essential for understanding the sources of consumption inside a building and to take appropriate measures to save energy (Peng & Qian, 2014). The importance of understanding the sources of energy consumption may lead to identifying ways of reducing energy consumption. This is vital as a global drive exists in decreasing the use of non-renewable energy sources and in reducing carbon footprints. This is even more critical within the boundaries of South Africa (SA), that has felt the effects of load shedding and massive electricity price hikes over the past decade.

Load shedding is defined as coordinated sets of controls that decrease the electric loads in the system so as to restore the system to its normal operating condition (Mageshvaran & Jayabarathi, 2015). It is often manifested by switching off the electrical energy supply to pre-selected neighborhoods within a city or town for up to 2 hours per day. This may have a serious impact on many small medium and micro-sized enterprises (SMME) (such as food stores, laundry services and internet café's), causing them to lose revenue or valuable stock items. Reducing the occurrence of load shedding is of primary importance which each homeowner may contribute to by taking responsibility for the electrical energy usage in their homes. Helping homeowners to identify unwanted energy consumption in their homes may empower them to better fulfill their responsibility.

Massive electrical energy price increases have hit SA home owners over the past decade. The National Energy Regulator of South Africa (NERSA) approved an 8% average electrical energy price increase per annum for the period 2013 through 2017 (NERSA, 2013). Currently, homeowners pay around R1 000 for 555 kWh of electrical energy (or 0.12 $ per kWh) that is almost sufficient to cover the energy consumption of an average sized residential household (consists of 3 people) in SA for one month.

Many commercial energy monitoring systems exist that may be used to identify unwanted energy consumption (P.E. Hertzog & Swart, 2015; Swart, Pienaar, & Schoeman, 2013). Stand-alone systems may cost from $150 to $1000, often requiring specialised PC software to analyse the data. One innovative way of by-passing these systems involves the use of the internet of things (IoT). IoT is demarcated as a huge network that merge with the Internet through sensors to collect information such as temperature, electricity, light intensity and position to name a few (Fan & Liu, 2014). How can the IoT though be used to identify sources of energy consumption?

The purpose of this paper is to describe the design and setup of a non-complex cost-effective energy monitoring system for a residential home environment. Firstly, a short overview of the Arduino UNO board, the ESP8266 WiFi module and the ThingSpeak cloud server will be given. The design of the energy monitoring system will follow. Lastly, a few test results are provided to highlight the data presentation formats of the system.

Arduino

Arduino originated from Italy and was designed in 2005 to teach Interaction Design. This design discipline puts prototyping at the heart of its process (Junior et al., 2013). The Arduino users consist of scientists and people from all walks of life. These may include, artists, DIY hobbyists and students from almost all disciplines (Jamieson, 2010). The concept of sharing designs on the web gives these users entree to a vast knowledge base (Jamieson, 2010).

The Arduino platform is very adaptable, and it is also an open-source platform which makes it very accessible (Aguilar-Acevedo & Alejo, 2013). The system offers a variety of digital and analog inputs and outputs, serial interface and Pulse-width-modulation (PWM) outputs (Pierre E Hertzog & Swart, 2016). The user code can easily be changed and updated via the USB port on a PC. The communication to the USB occurs by the use of standard serial protocol. The hardware is inexpensive, and the software is freely available for students (Kumar, Bindu, Sneha, & Sravani, 2013).

ESP8266

The ESP8266 is a very cheap WiFi chip with a full TCP/IP stack and an onboard microcontroller (Di Nisio, Di Noia, Carducci, & Spadavecchia, 2015). In August 2014, the ESP-01 module was launched by a manufacturer with the name AI-Thinker (Bohora, Maharjan, & Shrestha, 2016). The ESP-01 module (shown in Figure 1) incorporates the ESP8266 chip and allows a microcontroller to connect to a WiFi network and make simple TCP/IP connections. In this paper, the ESP-01 module is used to communicate data from an Arduino UNO board to the ThingSpeak cloud server. The Arduino board serves to interface a current transformer (CT) sensor that is hooked around the main AC supply line from the electrical energy utility.

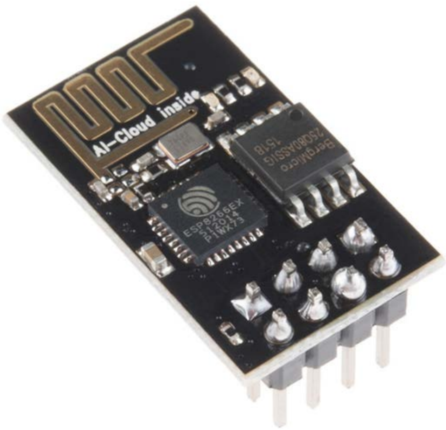

Figure 1- ESP-01 Wi-Fi module

ThingSpeak

ThingSpeak was launched in 2010 with the aim of supporting IoT applications (Montori, Bedogni, & Bononi, 2016). As part of the service, ThingSpeak has integrated support from MATLAB through Mathworks (Malagi, 2017). This gives ThingSpeak users access to tools to analyse and visualize uploaded data without the need for a software license for MATLAB.

The data captured on ThingSpeak is stored in channels (Acharya & Kuzhalvaimozhi, 2015). Each channel includes eight fields that can hold measured data, three fields for location data and one field for status data. Once data is collected in a channel, ThingSpeak can be used to analyse and visualize it. Recorded data can also be exported in a spreadsheet format (Abo-Zahhad, Ahmed, Farrag, Ahmed, & Ali, 2015). Channel settings include the following:

1. Channel Name: The user can enter a unique channel name.
2. Description: The user can describe the ThingSpeak channel.
3. Field#: Here the user can name and enable up to 8 data fields per channel.
4. Metadata: The user can enter information about channel data.
5. Tags: User identified tags.
6. Latitude: The latitude where the sensor is located.
7. Longitude: The longitude where the sensor is located.
8. Elevation: Specify the height position of the sensor.
9. Make Public: A checkbox to make channel public or to keep it private.
10. URL: If the user has a website that contains information about the channel, it can be specified here.

11. Video ID: If the user has a YouTube™ or Vimeo® video that displays the channel information, it can also be specified.

Design

In this section, the design of the energy monitoring system is presented. Firstly, an overview of the whole system will be given, where after the CT sensor and signal conditioning circuit will be described. The next step will be to explain the software and calculations performed by the Arduino microcontroller. The WiFi module will be described next, where after the setup of the cloud server (ThingSpeak) will be discussed. Lastly, the collection of data by the researcher from the cloud server will be outlined.

The complete block diagram of the energy monitoring system is shown in Figure 2. In SA, the single-phase AC mains supply from the electrical energy utility to residential houses is 230V. As a non-complex cost-effective system was envisioned, only the mains current was measured and not the mains voltage, which is known. Data from the CT in block one (see Figure 2) is passed to the Arduino microcontroller in the next block. The RMS current is measured every one second, with a one-second delay between samples, where the average current for ten measurements is used to calculate the power. This data is then passed to the ESP8266 WiFi module that is set to connect to a specified WiFi network to enable it to upload data to a cloud server, named ThingSpeak. The researcher can access the cloud server at any time and from anywhere on the planet, to simply view the data or download it for further analysis.

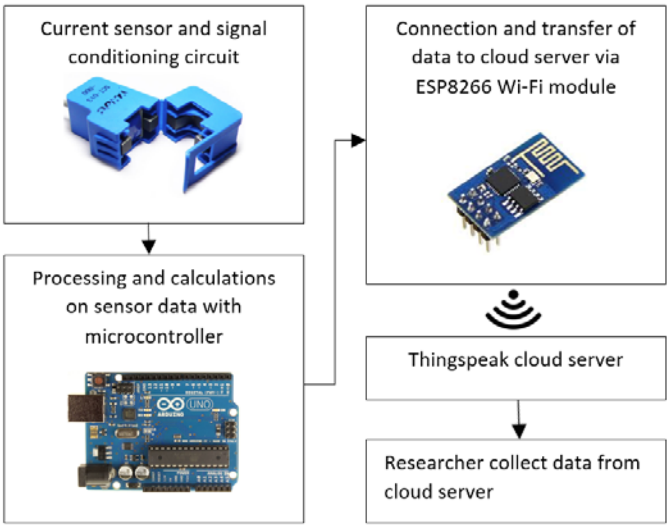

Figure 2- Block diagram of non-complex energy monitoring system

The current transformer sensor

The CT sensor that was chosen is rated for 100 A input, giving a corresponding 50 mA output, having a split core (SCT013). This enables easy installation of the CT sensor around the main AC supply line as shown in Figure 3 (within the dotted circle). Subsequently, the user does not need to disconnect any wires or modify any connections within the distribution board of the house. A multimeter is also visible in Figure 3 (AC-DC MS2600) that was used to validate the captured data on the

cloud server. The circuit that was used for signal conditioning is shown in Figure 4, which is connected between the CT sensor and the Arduino microcontroller.
The purpose of the signal conditioning circuit is to convert the current reading from the CT to a voltage below 2.5 V that can be sampled by the analog to digital converter of the Arduino microcontroller. Calculations concerning component values and calibration will follow.

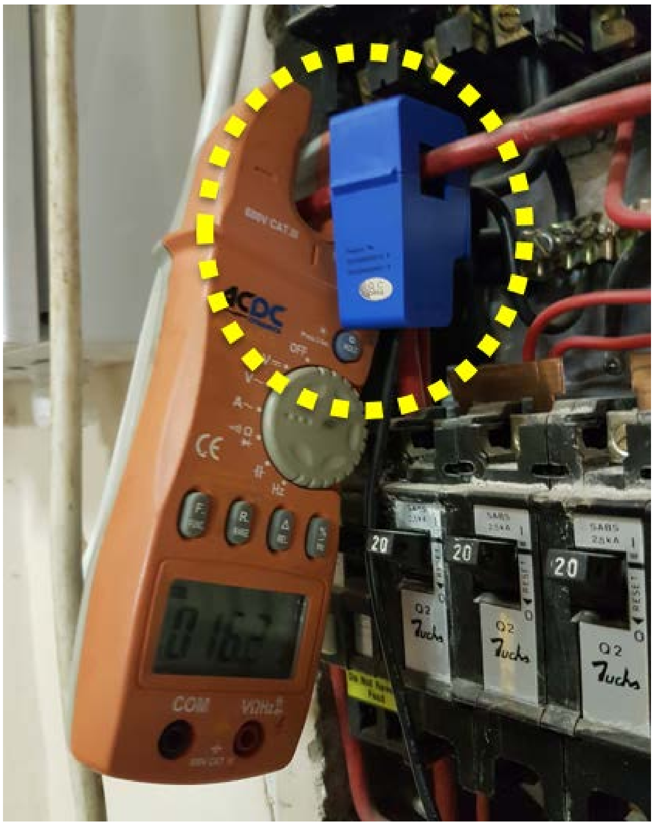

Figure 3- 100 A CT coupled in a distribution board of a residential home

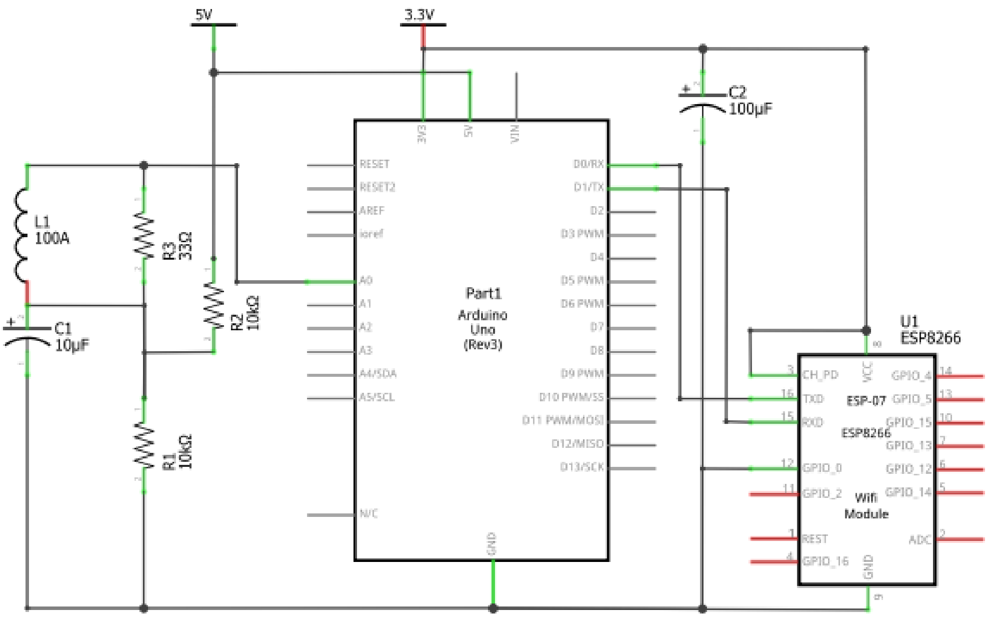

Figure 4- Circuit diagram of noncomplex energy monitor system

The CT's primary to secondary winding ratio can be calculated from the specified primary and secondary current ratio, as shown in Equation (1). The secondary current must be calculated next in order to determine a suitable burden resistor which is required by the CT sensor. The burden resistor is dependent on the maximum system current. Although this is a 100 A CT sensor, it may be that the maximum possible system current is less than 100 A. In this paper, the burden resistor is calculated for a maximum current of 100 A RMS. Current will flow through the burden resistor that will result in a specific voltage that can be read by the analog to digital converter of the microcontroller. The 100 A specified rating of the CT is an RMS value and needs to be converted to a maximum value (See Equation (2). The maximum value is needed to complete the calculation of the calibration factor that is discussed later in this section. The two 10 kΩ resistors (R1 and R2 in Figure 4) form a voltage divider circuit limiting the maximum input voltage to the Arduino to 2.5 V, which is developed across the burden resistor, R3. The burden resistor may then be calculated using Equation (4). The closest E12 value for the calculated burden resistor is then chosen. The voltage generated across the burden resistor by the current flowing in the secondary of the CT sensor is used as an input voltage to the analog to digital converter in the Arduino microprocessor (pin A0 in Figure 4). The last calculation (Equation (5)) is performed to determine a calibration factor. The calibration factor is dependent on the burden resistor and is used in the software to convert the measured voltage from the Arduino microcontroller (less than 2.5 V) into a real-time current value (maximum of 100 A).

$$n = \frac{Is}{Ip} = \frac{100A}{50mA} = 2000 \quad (1)$$

$$Ip(mesured) = \sqrt{2} \times Irms = \sqrt{2} \times 100 = 141.4A \quad (2)$$

$$Is = \frac{Ip}{n} = \frac{141.4A}{2000} = 0.0707A \quad (3)$$

$$R(burden) = \frac{V(sensor)}{Is} = \frac{2.5\text{V}}{0.0707A} = 35.4\Omega \quad (4)$$

$$Calibration\ value = \frac{n}{R(burden(E12))} = \frac{2000}{33\Omega} = 60.6 \quad (5)$$

Where:
$n = Transformer\ ratio$
$Is = Secondary\ current$
$Ip = Primary\ current$
$R(burden) = Burden\ resistor\ on\ secondary\ of\ C$
$V(sensor) = M$aximum voltage across the CT

Processing of sensor data

In this section, important aspects of the Arduino's microprocessor code is discussed. The entire code cannot be shown due to space constraints. Please note that the line numbers are only used to assist in the description of the required code.

The WiFi name (line 1) and password (line 2) must be defined, as well as the IP address (line 3) of the cloud server. In line 4 the key for the specific channel within the cloud server where the power reading must be uploaded is specified. This key can be obtained by the researcher from his or her account on the ThingSpeak cloud server. Line 5 includes the library for the energy calculations and line 6 the mathematical library. The energy calculations done in the library is to convert the measured voltage across the burden resistor, that was read by the analog to digital converter on the Arduino microcontroller, to a real-time current value.

```
1. #define SSID "SSID-WiFiname"
2. #define PASS "password"
3. #define IP "184.106.153.149"
4. String msg = "GET /update?key=KEY"
5. #include "EmonLib.h"
6. #include <math.h>
```

The calibration factor (60) that was calculated by using Equation (5), as well as the analog port (0) is stipulated in command line 7. This port is one of four analog input ports that may be used on the Arduino microprocessor to connect to the CT sensor.

```
7. emon1.current(0,60);
```

The voltage across the burden resistor of the CT is sampled via an analog to digital converter in the Arduino microcontroller. In line 14, a library function is called where the RMS current is calculated from the voltage that is read from analog port (A0). Ten readings are taken in a second, after which a delay of 1000 ms is set (line 13). This means that readings are obtained for every odd second (1, 3, 5 etc.) and then averaged according to line 18. The power calculation is done in line 19. This power value is the average power over a ten cycle period and is passed back to the main routine in the code (line 20).

```
8. void updateReadings(){
9. double Total=0;
10. int var = 0;
11. while(var < 10)
12. {
13. delay(1000);
14. Irms = emon1.calcIrms(1480);
15. Total=Irms+Total;
16. var++;
17. }
18. Irms=Tot/10;
19. Power=230*Irms;
20. }
```

WiFi communication and cloud storage

The average power that was obtained in the previous section must now be communicated to the cloud server, ThingSpeak. This is accomplished by the use of the ESP8266 WiFi module. The code for communication and transmission of the data by the ESP8266 module is shown in the following subroutine.

```
21. void WrightThingSpeak (){
22. String cmd = "AT+CIPSTART=\"TCP\",\"";
23. cmd += IP;
24. cmd += "\",80";
25. cmd = msg ;
26. cmd += "&field1=";
27. cmd += Power;
28. cmd += "\r\n";
29. }
```

In the previous subroutine, AT commands are used to communicate with the ThingSpeak cloud server via the ESP8266 module. The IP address of ThingSpeak (line 23), the port used (line 24), and the position of the calculated power value (line 27) is set. In the ThingSpeak cloud server, the data is captured and recorded and can be retrieved by the researcher at any time and anywhere. As an example, the recorded data for 24 April 2017 is displayed in Figure 5.

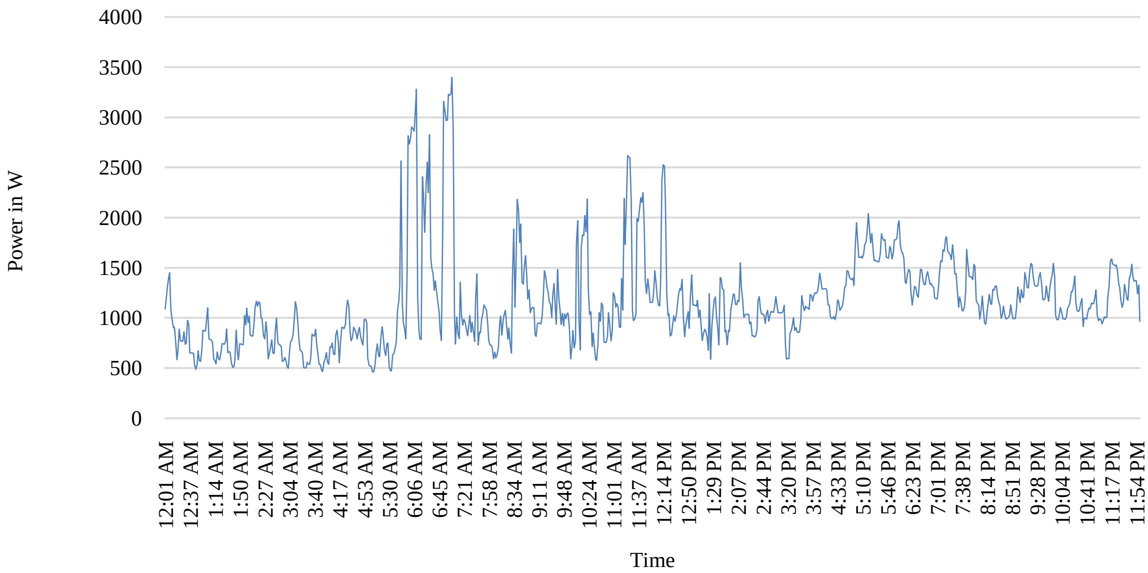

Figure 5- Example of energy data that was collected for 24 April 2017

It is evident that there are many individual peaks as appliances are switched on and off. General higher consumption can be noticed between 06h00 and 07h00 where the household is getting ready for work. In this period they have used hot water to shower, that will trigger the geyser thermostat, and have prepared food using a microwave oven and kettle. Another higher consumption area is evident between 11h00 and 12h00 that can be contributed to the preparation of lunch, which is the main meal of the family. A third and smaller peak can be noticed between 17h00 and 19h30 when various technological devices are being used by the family that has returned from work.

To confirm the validity of the recorded values, the Root-mean-square (RMS) current was also measured (using the multimeter shown in Fig. 4.) and compared to the recorded values. As can be seen from Figure 6 the values were very close, and the maximum deviation between the measured and recorded values were 1.6%.

Figure 6- Comparison between multimeter and ThingSpeak readings

Conclusions

The purpose of this paper was to describe the design and setup of a non-complex cost-effective energy monitoring system that an average residential homeowner may use to identify sources of energy consumption. This system is non-complex in that it only requires a CT sensor, Arduino microprocessor, ESP8266 WiFi module and internet connectivity. These components are furthermore cost-effective, as they may be purchased in SA for around $100. No dedicated PC is required to capture and record the data, and no expensive software is required to use the components. The data may be viewed at any time and anywhere, making it very convenient for the homeowner who may want to keep an eye on his home's energy consumption while away on vacation. This enables the continuous monitoring of heavy intensive electrical energy guzzlers, such as water pumps. These are required to supply adequate water from bore holes in drought-stricken areas or to maintain the circulation of water in swimming pools.

Further research may include the analysis of data over longer periods of time for specific households, the use of Long Term Evolution (LTE) technology to transfer the data to the ThingSpeak cloud server and to monitor more sensors with the system. Identifying the power signature of different sources of energy consumption is also planned.

References

Abo-Zahhad, M., Ahmed, S. M., Farrag, M., Ahmed, M. F., & Ali, A. (2015). *Design and implementation of building energy monitoring and management system based on wireless sensor networks.* Paper presented at the Computer Engineering & Systems (ICCES), 2015 Tenth International Conference on.

Acharya, C., & Kuzhalvaimozhi, S. (2015). Irrigation and Internet of Things Platform.

Aguilar-Acevedo, F., & Alejo, V. G. (2013). *Using open-source platform for trajectory control of DC motors.* Paper presented at the Power, Electronics and Computing (ROPEC), 2013 IEEE International Autumn Meeting on.

Bohora, B., Maharjan, S., & Shrestha, B. R. (2016). IoT Based Smart Home Using Blynk Framework. *Zerone Scholar, 1*(1), 26-30.

Di Nisio, A., Di Noia, T., Carducci, C. G. C., & Spadavecchia, M. (2015). *Design of a low cost multipurpose wireless sensor network.* Paper presented at the Measurements & Networking (M&N), 2015 IEEE International Workshop on.

Fan, Y. B., & Liu, H. (2014). Study on Application Model of Internet of Things for Green Manufacturing. *Applied Mechanics and Materials, 484-485*, 187-190.

Hertzog, P. E., & Swart, A. J. (2015). *A customizable energy monitoring system for renewable energy systems*. Paper presented at the SAUPEC 2015, University of Johannesburg.

Hertzog, P. E., & Swart, A. J. (2016). *Arduino – Enabling engineering students to obtain academic success in a design-based module.* Paper presented at the Global Engineering Education Conference (EDUCON), 2016 IEEE.

Jamieson, P. (2010). Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat? *Proc. FECS*, 289-294.

Junior, L. A., Neto, O. T., Hernandez, M. F., Martins, P. S., Roger, L. L., & Guerra, F. A. (2013). A low-cost and simple arduino-based educational robotics kit. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Robotics and Control (JSRC), December edition, 3*(12), 1-7.

Kumar, Y. N., Bindu, P. H., Sneha, A. D., & Sravani, A. (2013). A Novel Implementation of Phase Control Technique for Speed Control of Induction Motor Using ARDUINO. *International Journal of Emerging Technology and Advanced Engineering, 3*(4), 469-473.

Mageshvaran, R., & Jayabarathi, T. (2015). GSO based optimization of steady state load shedding in power systems to mitigate blackout during generation contingencies. *Ain Shams Engineering Journal, 6*(1), 145-160.

Malagi, M. (2017). Heath monitoring system based on IoT.

Montori, F., Bedogni, L., & Bononi, L. (2016). *On the integration of heterogeneous data sources for the collaborative Internet of Things.* Paper presented at the Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum on.

NERSA. (2013). NERSA's decision on Eskom's Revenue Application for the Third Multi-Year Price Determination period 2013/14 to 2017/18. Retrieved from http://www.nersa.org.za/

Peng, C., & Qian, K. (2014). Development and application of a Zigbee-based building energy monitoring and control system. *The Scientific World Journal, 2014*.

Swart, A. J., Pienaar, H. C., & Schoeman, R. M. (2013). Cost-effective energy monitoring of domestic off-grid PV systems. *JEPE, Journal of Energy and Power Engineering, 5*, 182-188. doi:10.4236/epe.2013.54B035.