A Non-intrusive Approach for Smart Power Meter

Hugo Tavares, Bruno Prado, Kalil Bispo, Daniel Dantas
Department of Computing, Federal University of Sergipe
São Cristóvão, Brazil
{hugomt, bruno, kalil, ddantas}@dcomp.ufs.br

***Abstract*—This paper presents the development of a non intrusive smart power meter capable of giving its users methods of remote management of electric power consumption information. It has as objectives the real-time monitoring of electric energy consumption using non-intrusive, with reduced size and low cost sensors, and the wireless management of the data. This meter can also be used as a module in other projects, by using the MQTT protocol. The *emonLib* library was used, allowing power factor measurement and compatibility with non-invasive current sensors. Communication between the power meter and the outer environment, was made through Wi-Fi technology, allowing observation of the data in real-time, either through *Google Sheets*, the HTTP web server hosted in ESP8266 module or on *MyDevices Cayenne* platform via MQTT protocol. As a case study of information provided by the meter, we use the prediction of the electricity bill. The smart power meter was tested in actual situations and using different kinds of loads, yielding to highly accurate measurements.**

***Index Terms*—smart meter, real-time, non-intrusive, load monitoring, Arduino, ESP8266, MQTT**

I. Introduction

Global electricity consumption has increased significantly, especially in the residential and commercial sectors [1]. One of the factors that influence this increase is the inefficient use of electric energy, mainly in HVAC (Heating, Ventilation, and Air Conditioning) equipment, water heaters, light bulbs, and other electrical appliances [2]. The easy access to the information of energy consumption can reduce this promoting changes in user habits [3].

The advancement in electronic devices made possible to develop accurate, small and inexpensive sensors and the popularization of SoC (System-on-a-Chip). These, on the other hand, have made it possible to create inexpensive, efficient and easy-to-install energy meters, which is the proposal of our meter. Electricity load monitoring is essential to predict usage patterns, power consumption and service quality. In a context of growing demand for power and renewable distributed generation, real-time smart power monitoring is required to achieve a sustainable electricity production [4].

II. Related Works

Commercially available off-the-shelf power meters were considered as related work, such as Schneider PowerLogic PM1000 (panel mount), Eastron SDM530D and Deson D52-2048 (DIN rails). Only Deson D52-2048 measures the electric current in a non-intrusive manner and none of them support three-phase loads or provide a remote access interface for acquired data. Thus, some power-aware applications, such as real-time power consumption or energy bill prediction, can not be straightforwardly integrated to mobile or web based systems.

Academic solutions are also available. Vergara and Villaruz [5] propose an automated utility power monitoring system. It calculates the average consumed power along with its price in real-time and saves a data log to an SD card. The recorded information includes real-time date and time for data retention, in case of power failure recovery. Sabalza et al. [6] show a device able to measure the energy consumption of each device at home, identifying which devices contribute most to the cost of the electric bill. The proposed device is capable of monitoring a three-phase load using a non-intrusive current sensor, but remote access to data has not been implemented. Ashraf et al. [7] also measured electrical energy consumption of each home device, but the consumption information can only be accessed locally through Wi-Fi. Juhana and Irawan [3] built a monitoring system for some electrical parameters, such as RMS current and voltage, real power, apparent power and power factor, using non-intrusive current sensor and access to consumption information through Bluetooth Low Energy (BLE) technology. Similarly, Chandra et al. [8] also used non-intrusive current sensors, but it has a different approach than previous works by using a light sensor module. This module detects the pulses given out by the power meter through its LED, which may not be available or accessible in all power meter models. All the considered works use non-intrusive current sensors, but their lack of integration with IoT platforms, such as myDevices Cayenne [9], prevent a useful deployment of proposed approaches.

Sonandkar et al. [10] show a real-time graph of the power consumed during certain time intervals, encouraging users to use devices that consume a lot of energy at times when energy demand is lower and, consequently, the kWh rate be cheaper. Despa et al. [11] propose an electrical system information monitor (voltage, current, power factor, power and energy consumption) in real-time, recording the data in a database and using a web application to access it remotely. Similarly, Shree and Vadana [12] show a real-time power, energy and power factor (corrected if necessary). These approaches use intrusive current sensors, making the installation more difficult due to the need of power system interruption for serial connection of the sensor. Only Despa et al. [11] allow remote viewing of the acquired information over the web, making data easily accessible to applications and user visualization.

The major contribution of this work, while compared to

978-1-5386-4829-2/18/$31.00 ©2018 IEEE

commercial products and related works, is a multi-purpose smart power meter with scalable number of phases, low cost and MQTT compatible communication.

III. Power Measurement System Overview

The system uses as sensors in each of the three phases, a current transformer (CT), as a non-intrusive current sensor, and a power transformer (PT), as a voltage sensor. These sensors are installed closer to the meter of the electric company. The data is processed on Arduino platform, which is free and consists of very simple and easy-to-use hardware and software elements [13]. The System-On-Chip (SoC) ESP8266 Wi-Fi module allows the microprocessor to connect to the Internet. Electrical information, such as voltage, current, and energy consumption are saved to an SD card, preventing loss of data in electrical failures. Data is periodically sent to platforms in the cloud such as myDevices Cayenne and Google Sheets, and shown on a webpage hosted on ESP8266. The block diagram of the system is presented in Fig. 1 and the prototype is presented in Fig.2.

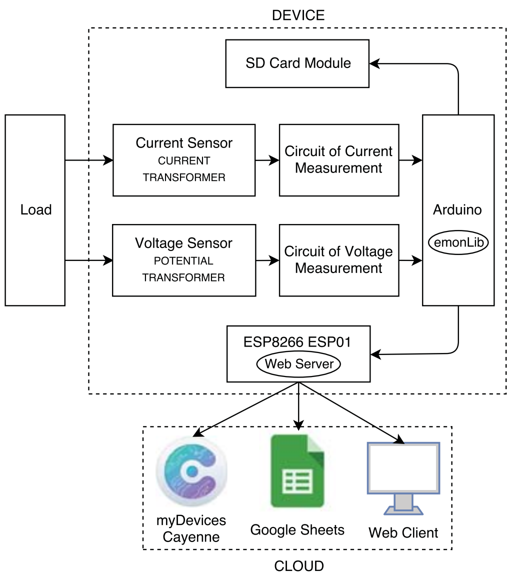


Fig. 1: Block diagram of the system

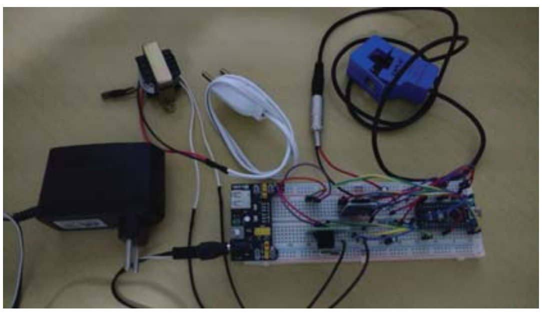


Fig. 2: System's prototype

A. emonLib

The emonLib library calculates real power, apparent power, power factor, voltage and current regardless the number of phases the circuit has. This library considers the use of non-intrusive clamp type current sensor, making our design easier. Operation of the library consists of implementing in software a zero crossing detector for calculating the power factor, reading samples of voltage and current in the ADC, applying a digital filter and removing the bias voltage from them, and finally, calculating the effective values of these quantities using (1). As the samples of instantaneous power are taken every second, to transform those into consumed energy we must divide the sampled value by k $\times$ h, which corresponds to 1000 $\times$ 3600s, totaling 3600000s.

$$X_{RMS} = \sqrt{\frac{x_1^2 + x_2^2 + \cdots + x_n^2}{n}} \tag{1}$$

B. Calibration

Some factors that contribute to uncertainty in the value that we read are the internal reference voltage of Arduino, nominally 1.1V, within the range of 1.0V and 1.2V, representing an approximately 9.0% uncertainty rate; the current transformer accuracy, that represents an approximately 1.0% of uncertainty [14]; the potential transformer accuracy, that represents an approximately 2.0% of uncertainty and the voltage divider uncertainty, which depends on the values of the resistors and the applied voltage, is 0.85%, obtained by the propagation of uncertainties of (2).

$$V_p = \frac{R_4}{R_3 + R_4} \cdot V_{RMS} \tag{2}$$

For calibration, a Minipa ET-1649 digital multimeter and a Minipa ET-3111 digital clamp meter [15] were used as measuring instruments. The load used, purely resistive, was an 40w incandescent lamp. The process consisted in adjusting the current and voltage measurement calibration coefficients of the emonLib library in the Arduino sketch until the measured values converged to the values presented in these measuring instruments. To calibrate the phase angle, the phase coefficient must be adjusted until the power factor become 1, because the chosen load was purely resistive, so that real power and apparent power have the same value.

IV. Device Architecture

The meter hardware consists mainly of the current transformer (subsection IV-A), voltage transformer (subsection IV-B), the SD card module (subsection IV-C) and the SoC ESP8266 (subsection IV-D). Besides these items, there are some more discrete components, which are shown in Table I.

The cost of the meter, from the items in Table I, was $60.00 at the distributor of electronic components Digikey [16].

TABLE I: Meter Components

Item	Quantity
YHDC SCT-013-050 Current Transformer	1
100-240VAC to 9VAC 150mAh Power Transformer	1
100-240VAC to 9VDC 1A Power Supply	1
Arduino Nano	1
ESP8266 ESP01 SoC	1
1GB MicroSD Card	1
MicroSD Card Module For Arduino	1
10KΩ (±5%) 1/4W Resistor	4
18KΩ (±5%) 1/4W Resistor	1
3K9Ω (±5%) 1/4W Resistor	1
10μF 50V Electrolytic Capacitor	2
1N4734 Zener Diode	1

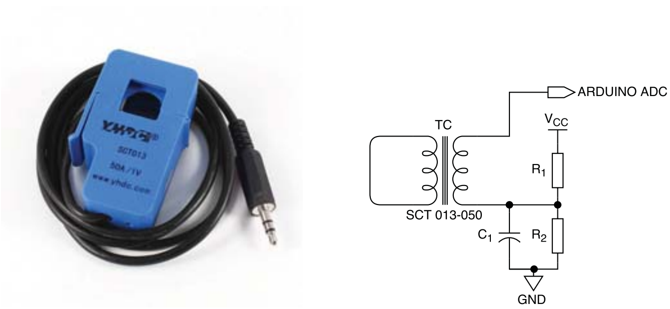
A. Non-intrusive Current Sensor Using a Current Transformer

The electrical current transformer reproduces in its secondary circuit a current proportional to that which circulates in its primary circuit. In our system, current measurement is done by CT SCT-013-050, presented in Fig. 3a and manufactured by YHDC, measuring a maximum of 50A. This specific model has a burden resistor coupled to the secondary circuit, based on Ohm's Law, will generate a voltage proportional to the current flow, which is 50A / 1V.

The sensor response is an almost sine wave with vertical axis centered at zero. Arduino Analog-Digital Converter (ADC) reads only positive voltages in the range of 0V to 5V, so it is necessary to add an DC bias voltage of 2.5V, keeping the signal on the positive side maximum (3.5V) and minimum (1.5V) excursions. The circuit schematic can be seen in Fig. 3b.

VCC voltage, from Arduino, is 5V. Equation (3) gives the relation $R_1 = R_2$ to obtain the desired bias voltage. The higher the resistance chosen, the lower the consumption of the meter. Since our system was not powered by battery, we chose $R_1 = R_2 = 10k\Omega$ (1/4W). The bypass capacitor C_1 must have a low reactance, being used in our system 10μF 50V.

$$V_{bias} = V_{CC} \cdot \frac{R_2}{R_1 + R_2} \tag{3}$$



(a) Current transformer [17] (b) Schematic circuit

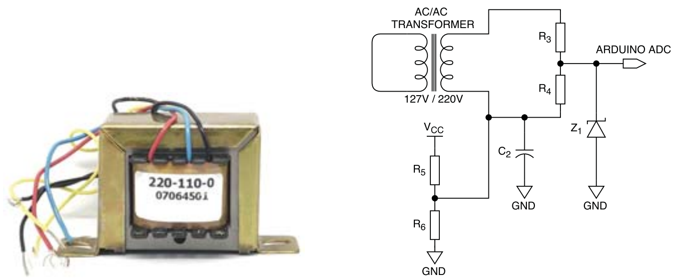
Fig. 3: Current measurement

B. Voltage Sensor Using a Potential Transformer

This transformer generates an induced voltage in its secondary winding proportional to the voltage of the primary winding. In our system, we used a voltage transformer from 127/220VAC to 9VAC, presented in Fig. 4a.

The sensor response is an almost sine wave with the vertical axis centered at zero and the effective voltage (RMS) at the transformer output is 12.73V ($9\sqrt{2}$) for an input voltage of 220VAC. The first step is to reduce this peak-to-peak voltage to an acceptable level by the ADC. We chose V_{pp} = 4.5V, leaving a voltage safety margin. The ratio $R_4 = 0.215 \times R_3$ was defined by (2) and we chose $R_3 = 18k\Omega$ (1/4W) and $R_4 = 3.9k\Omega$ (1/4W), following the Electronic Industries Association (EIA) E12 standard resistor series.

Analogously to current measurement, a bias voltage and a bypass capacitor are required, so $R_1 = R_2 = R_5 = R_6 = 10k\Omega$ (1/4W) and $C_1 = C_2 = 10\mu F$ X 50V. The zener diode was used to protect the ADC in case of overvoltage, where Z_1 = 1N4734, whose Zener voltage is 5.6V.



(a) Potential transformer [18] (b) Schematic circuit

Fig. 4: Voltage measurement

C. SD Card Module

Arduino EEPROM has a lifetime of over 100.000 write/read cycles [19] and it has very low storage capacity, so the energy consumption value for each phase and the data sent to the ESP01, regardless the success of sending, have been saved to a microSD card. Saving the data locally avoids data loss due to power failure and registers it even if there is no Internet connection, which is required to sending data to the cloud. The module wiring diagram to the Arduino is presented in Fig. 5.

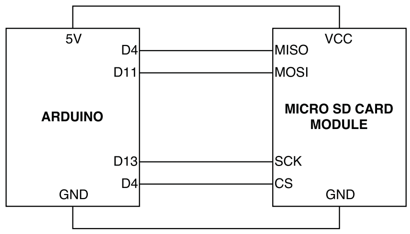


Fig. 5: Wiring diagram

D. ESP8266 ESP01 SoC

The SoC ESP8266 ESP01 was chosen for meeting the objectives of this system, being of low-cost and popular. Communication between the SoC and the Arduino is made through serial port, but that can not be direct because the SoC operates at 3.3V and the Arduino at 5V, making it necessary the use of a bi-directional logic level converter. To supply the SoC, a 3.3V voltage for at least 200mA [20] is also required, which can be supplied by a voltage regulator, such as the LD33, for example, which provides 800mA. The wiring diagram is presented in Fig. 6, where C_1 is a 100nF ceramic disc capacitor and C_2 is a 10μF X 50V electrolytic capacitor.

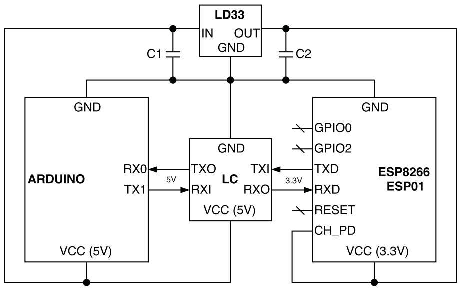


Fig. 6: Wiring diagram

V. Cloud Components

The Arduino sends to the ESP8266 via serial communication the string with the following pattern: "CM+V1;V2;V3;I1;I2;I3;CE1;CE2;CE3;TIP", where V1, V2 and V3 are the voltages of each phase; I1, I2 and I3 are the currents of each phase; CE1, CE2, CE3 the consumed energy of each phase and TIP the total instantaneous power. If the data matches the defined pattern, it will be decoded and sent by ESP01 to the myDevices Cayenne platform and Google Sheets at a given time interval.

A. HTTP Web Server

The page loaded in ESP8266, which can be seen in Figure 7, is capable of informing the user of accumulated power, instantaneous power and voltages and currents of each phases. As a case study, we use this data to also calculate the energy bill prediction. This data is sent from the Arduino to the ESP8266 via serial communication and is made available to the webpage via JSON file. The information is read by the page via HTTP GET method.

To make data visualization more dynamic, the page also displays the daily, monthly and annual consumption charts, presented in Fig. 7, allowing more interactivity. These charts are built using the Google Charts tool, which allows to create customizable graphs from a defined data selection. The monthly consumption chart, for example, allows the daily comparison of consumption for each week of the month, making the user know, for example, on which Sunday of the month the consumption was greater.

The webpage also provides the possibility to change the electricity rate ($/kWh), which is passed from the webpage to the ESP8266 through an HTTP POST method. Thinking about the scalability of the project, this option could be used to change other values, such as calibration constants, the MQTT parameters for communication to myDevices Cayenne [9], the sheet identifier parameter saved in Google Sheets, among others.

Fig. 7: Power meter home page

For the generation of the graphs previously seen (subsection A), Google Sheets was the easiest and fastest tool that we found for storing and connecting the data, since both services are from the same company. Although it seems unlimited, Google Sheets has a restriction of two million cells written [21]. Because of that, we adopted a rate of 7 writes every 90 seconds for the data is saved for a year.

To receiving information from ESP8266, it was necessary to create a script associated with the spreadsheet using the Google Apps Script editor. This script will be a middle agent between the ESP01 and the spreadsheet, receiving information via the HTTPS protocol and inserting it directly into the cells.

The ESP8266, by using the ESP8266WiFi core [22], performs communication with HTTP protocol, which does not have the security layer. Since the script receives information only via HTTPS protocol, we had two possibilities: To use third-party services (such as PushingBox, among others) to receive module information via HTTPS and pass them to the script via HTTP with a daily request limit; or to use a library (HTTPSRedirect [23]) that provided the HTTPS protocol to the module, which was the chosen option.

B. myDevices Cayenne

The myDevices Cayenne platform was the first one with drag & drop capability, facilitating the creation, prototyping and production systems for IoT, in addition to allowing the sharing of projects [9]. The platform stores a history of data and displays it in graphs, making it easy to read, and provides a mobile application for Android and iOS, making it possible to remotely access to data. We have chosen to display voltage and current for each of the three phases, the energy consumption and the quality of the Wi-Fi signal, presented in Fig. 8.

Integration with this platform allows the user to use the information obtained according to their needs, being able to use them to monitoring a residence's voltage and current values as well as to calculating the energy consumed, estimating the value of the electricity bill, alerting or powering off loads in cases of undervoltages and overvoltages, giving warnings of absence of voltage in any of the phases, powering on or power off loads, among other creative functionalities.

Communication between the ESP8266 and the platform is through the MQTT (Message Queue Telemetry Transport) protocol, which was invented by IBM in 1999. The MQTT is an extremely simple messaging protocol designed for small devices with low network bandwidth consumption or unreliable, making it ideal for IoT devices. One of its main features is a high rate of message delivery guarantee [24].

The protocol is based on the principle of publish/subscribe model, that is, if a device on the network wants to receive information, it can subscribe to a topic to which it belongs and receive any messages published in it. Similarly, it occurs for devices wishing to publish information, making it available in a topic for all devices that are subscribed in it [25]. The main point of the publish/subscribe protocol is the broker. He is responsible for receiving all the messages, filtering them and deciding who are interested in it and then sending the message to the subscribed clients [26].

The registration of device must first be done at myDevices Cayenne site. In order to creating the project, the selected device to be connected to the platform is the Generic ESP8266, and after the project's creation, it will be provided by the platform the parameters: MQTT Username, MQTT Password, Client ID, MQTT Server and MQTT Port. After downloading the Cayenne-MQTT-ESP8266 library, the firmware is loaded into ESP01 and the MQTT parameters mentioned above is placed, promoting the connection between the ESP8266 and the platform. After these steps, the information that will be sent can be configured in the firmware.

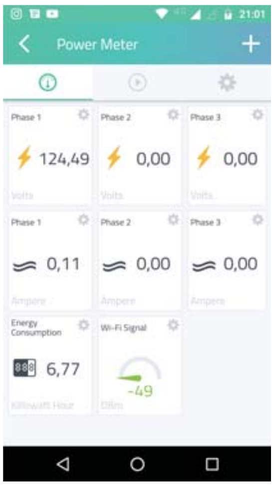


Fig. 8: myDevices Cayenne Mobile

VI. Results

The experiments were designed to evaluate the meter created by us for accuracy. As a comparative meter we used the Peacefair PZEM-002, which accuracy is 1% [27]. The first evaluation consisted of comparing the instantaneous power between our meter and the Peacefair's oneand the second evaluation was to record the energy consumption of both meters after the end of each experiment. Figure 9 shows both power meters.

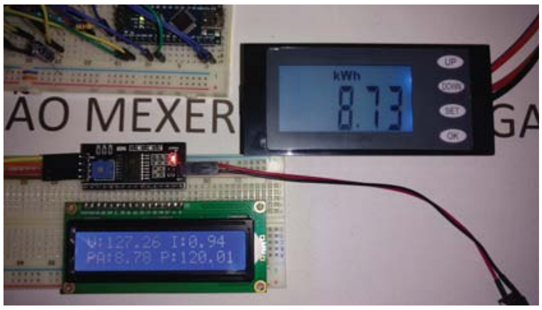


Fig. 9: Our meter and the PZEM-002 off-the-shelf power meter

To do the evaluation, 60 samples of each meter were obtained, each sample of both meters being taken at the same time and at intervals of 1 minute. After sampling, the average and standard deviation values were calculated for each meter. Those values served as basis for calculating the relative error of both meters compared to the value of the load declared power. Finally, the relative error between the two meters was calculated. To capture the data of our meter a timer was used that sent instantaneous data every minute, transmitting it via serial communication. For the PZEM-002, which has no serial communication, the Auto Webcam Capture application was used, capturing images every minute and saving them on the computer. The results of the sampling of both meters were imported by a spreadsheet editor and the calculations were performed.

TABLE II: Loads Description

Load	Description
1	25W incandescent light bulb
2	40W incandescent light bulb
3	100W halogen light bulb
4	13W fluorescent light bulb
5	25W fluorescent light bulb
6	32W fluorescent light bulb
7	34W welding iron
8	50W welding iron
9	190W mixed load (loads 1, 2, 3 and 5)

The loads used in the experiment can be seen in Table II and the accuracy of our meter compared to the off-the-shelf meter can be seen in Table III for the instantaneous power. The experiment with the 13W fluorescent light bulb ended to be less precise one, due to be the one with the lowest load, so noises in the ADC might have had a greater magnitude than the signal itself, compromising the reading. However, the meter is designed to measuring the consumption of the entire residence, so a similar situation will hardly occur.

TABLE III: Instantaneous Power Accuracy

Load	Our Meter	PZEM-002	Accuracy
1	24.73	24.36	98.50%
2	40.96	38.63	94.31%
3	94.48	99.29	95.15%
4	13.94	11.75	84.28%
5	22.90	21.40	93,44%
6	22.65	21.87	96.55%
7	28.45	30.20	94.20%
8	43.09	44.58	96.65%
9	177.03	187.56	94.38%

Both meters display the values in two decimal places. As the loads tested were of low power, the order of magnitude of the values that were measured was also of two decimal places, thus, there was no notable difference between the values of energy consumption registered by both meters.

VII. Conclusion And Future Works

The proposed real-time smart power meter uses non-intrusive, small and low-cost current sensors, making the meter installation easy, safe and feasible. It also provides friendly remote data access and makes the prediction of the electricity bill as a study case. Communication is performed by the MQTT protocol and the meter was tested by measuring instantaneous power and energy consumption in ten loads, having their values compared to the Peacefair PZEM-002 off-the-shelf meter. The results obtained from instantaneous power measurement was satisfactory, presenting an 5.8% error rate. From the energy consumption view, the values measured by both meters were not different, making our meter feasible for being inexpensive, easy to install and of low cost.

Unlike conventional meter models, our meter sends the information to myDevices Cayenne platform, to Google Sheets and also to a webpage hosted on the ESP8266, displaying currents, voltages and instantaneous power for each phase, as well as the energy consumption and prediction of the cost of the electricity bill. The system is modifiable and not restricted only to this information, because the user is free to choose what information to send.The main contribution of this work is the development of a real-time smart power meter that allows the remote monitoring of information and that can be used as a module for other works.

Aiming at sustainable development and the growth of the use of renewable energy, we must add another pair of sensors to the meter developed for monitoring a solar energy generation system, developing techniques that facilitate the visualization of the energy generated and consumed and that make our meter even more customizable for the user. In addition, we intend to facilitate the calibration process of the meter, using software instead of hardware. For this, we will add trimmer potentiometers (trimpots) for the user to regulate the calibration constant without having to re-record the firmware in Arduino.

References

[1] I. E. Agency, "Key world energy statistics," Available at: https://goo.gl/SPqQH7, 2017, [Accessed 24 May 2018].

[2] Connect4Climate, "What uses the most energy in your home?" Available at: https://goo.gl/V1qyww, 2017, [Accessed 24 May 2018].

[3] T. Juhana and A. I. Irawan, "Non-intrusive load monitoring using Bluetooth low energy," in *2015 9th International Conference on Telecommunication Systems Services and Applications (TSSA)*. IEEE, Nov. 2015, pp. 1 – 4.

[4] F. D. Garcia, F. P. Marafo, W. A. d. Souza, and L. C. P. d. Silva, "Power metering: History and future trends," in *2017 Ninth Annual IEEE Green Technologies Conference (GreenTech)*, March 2017, pp. 26–33.

[5] A. L. Vergara and H. M. Villaruz, "Development of an Arduino-based automated household utility power monitoring system," in *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. IEEE, Nov. 2014, pp. 1 – 6.

[6] M. M. Sabalza, J. D. B. O., and J. C. M. Santos, "Design and construction of a power meter to optimize usage of the electric power," in *2014 III International Congress of Engineering Mechatronics and Automation (CIIMA)*. IEEE, oct 2014, pp. 1 – 5.

[7] Q. M. Ashraf, M. I. M. Yusoff, A. A. Azman, N. M. Nor, N. A. A. Fuzi, M. S. Saharedan, and N. A. Omar, "Energy monitoring prototype for Internet of Things: Preliminary results," *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, pp. 1 – 5, 2016.

[8] P. A. Chandra, G. M. Vamsi, Y. S. Manoj, and G. I. Mary, "Automated energy meter using WiFi enabled Raspberry Pi," in *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, may 2016, pp. 1992 – 1994.

[9] myDevices, "Features," 2017. [Online]. Available: https://goo.gl/nMtSVc

[10] V. Sonandkar, A. Bhati, D. Gupta, S. Chouhan, N. Kinhekar, and N. P. Padhy, "Power measurement using Arduino for effective demand response," in *2016 IEEE 6th International Conference on Power Systems (ICPS)*. IEEE, Mar. 2016, pp. 1 – 5.

[11] D. Despa, A. Kurniawan, M. Komarudin, Mardiana, and G. F. Nama, "Smart monitoring of electrical quantities based on single board computer BCM2835," in *2015 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. IEEE, oct 2015, pp. 315 – 320.

[12] M. A. Shree and D. P. Vadana, "Smart meter for power factor enhancement in real-time," *International Conference on Embedded Systems, ICES 2014*, pp. 177 – 181, 2014.

[13] A. Javed, *Building Arduino Projects for the Internet of Things: Experiments with Real-World Applications*, 1st ed. Lake Zurich, Illinois, USA: Apress, 2016.

[14] YHDC, "Product specification," Available at: https://goo.gl/dGJmuC, 2015, [Accessed 17 Oct. 2017].

[15] "Minipa," 2018. [Online]. Available: https://goo.gl/3i6Mdm

[16] "Digikey," 2018. [Online]. Available: https://goo.gl/BCXhJZ

[17] Aliexpress, "1PCS NEW SCT-013-050 Non-invasive AC Current Sensor Clamp Sensor 50A," 2018. [Online]. Available: https://goo.gl/VUdsYE

[18] EletronicaGPL, "Transformador Hayama 12/800 12+12V, 800mAh, Bivolt," 2018. [Online]. Available: https://goo.gl/YXFqVg

[19] Arduino, "EEPROM write," 2017. [Online]. Available: https://goo.gl/X4Hz78

[20] E. W. Weisstein, "Root-mean-square," Available at: https://goo.gl/78mTKW, 2017, [Accessed 26 May. 2017].

[21] Google, "Arquivos que você pode armazenar no Google Drive," 2017. [Online]. Available: https://goo.gl/ts6m2d

[22] "Arduino core for ESP8266 WiFi chip," 2018. [Online]. Available: https://goo.gl/5As718

[23] "HTTPS redirect," 2018. [Online]. Available: https://goo.gl/WKnNGq

[24] MQTT, "Frequently asked questions," 2018. [Online]. Available: https://goo.gl/cjVxFd

[25] V. Lampkin, W. T. Leong, L. Olivera, S. Rawat, N. Subrahmanyam, and R. Xiang, *Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry*. IBM Redbooks, 2012.

[26] HiveMQ, "MQTT essentials part 3: Client, broker and connection establishment," 2018. [Online]. Available: https://goo.gl/FxrvhF

[27] M. P. Jones, "Multifunction AC panel meter," 2017. [Online]. Available: https://goo.gl/oJWbjc