

# Cost-effective solar water heater monitoring system using the internet of things

Pierre E Hertzog

*Department of Electrical, Electronics and Computer Engineering*

*Central University of Technology*

*Bloemfontein, South Africa*

[phertzog@cut.ac.za](mailto:phertzog@cut.ac.za)

**Abstract**—Monitoring of solar water heater systems can be a daunting task for residential home owners. The effective management of solar water heaters is dependent on accurate data regarding the usage and measurable parameters of the specific installed system. This research aimed to develop a non-complex and cost effective solar water heater monitoring system that is connected to the internet of things for data collection. Data collected by the system includes the power consumed by the system, the outside and inside temperatures, as well as the time of occurrence. The designed system is non-complex in the sense that it can easily be installed by the home owner without disturbing the existing electrical wiring of the residential house. Furthermore, data is collected via an internet server, called ThingSpeak, using an existing WiFi connection at the residence. The system is cost-effective regarding affordable hardware and free software that is used in the operation of the system. With data that is obtained from the system regarding warm water usage and power consumption, the home owner can make informed decisions to optimize the solar water installation.

**Keywords**—*Arduino; ThingSpeak; energy; solar water heater*

## I. INTRODUCTION

In South Africa, a properly working solar water heater (SWH) system can reduce the electricity demands of a household for hot water by up to 70% [1]. This can result in a significant monthly saving, as heating water takes up more than 30% of a households total electricity consumption [1, 2].

In a study on SWH systems, it was found that the payback period for a typical installation in South Africa (SA) is in the order of 8 years [3]. This payback period is highly dependent on the installed system, how effectively the system is configured as well as the level of maintenance on the system [4]. Typical system configuration settings may include water temperature, heater on/off times and circulation pump settings. All of these settings, as well as the pattern of hot water usage of the specific household, will influence the efficiency of the SWH system.

However, for the homeowner to manage and configure the SWH system efficiently, he or she must have access to accurate data that is specific to the installed system, as well as the hot water usage patterns of the household. This study aimed to develop a non-complex and cost-effective solar water heater monitoring system that is connected to the internet of things for data collection. From such a designed system data can then

easily be accessed by the home owner for monitoring, optimization, and maintenance of the SWH system.

The aim of this paper is thus to describe the designed system. In the designed system, current and temperature sensors are used as inputs to the analog to digital converters of an Arduino microprocessor. The microprocessor then communicates the sensor data to an ESP8266 Wi-Fi module. The Wi-Fi module connects to the Internet of Things (IoT) cloud server named ThingSpeak, where the data is stored and displayed. The structure of this paper is as follows. Firstly, the Wi-Fi Module, the ThingSpeak server, and the Microprocessor is introduced. After that, the specific SWH system that was monitored in this study is explained. The complete hardware and software design, as well as an example of collected data, follows. Succinct conclusion end the paper.

## II. ESP8266

The ESP8266 is a low-cost Wi-Fi chip with a full TCP/IP stack and an onboard microcontroller [5]. In August 2014, the ESP-01 module was launched by a third-party manufacturer with the name AI-Thinker [6]. The ESP-01 module incorporates the ESP8266 chip and allows a microcontroller to connect to a Wi-Fi network and make simple TCP/IP connections [7]. In this paper, the ESP-01 module is used to communicate current and temperature readings that are received via the analog to digital converters of 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 of the SWH as well as two thermistors that measure the inside and outside temperature of the SWH.

## III. THINGSPEAK

ThingSpeak was launched in 2010 with the aim of supporting IoT applications [8]. As part of the service, ThingSpeak has integrated support from MATLAB through Mathworks [9]. This gives ThingSpeak users access to MATLAB tools to analyze and visualize data without the need for a MATLAB software license [10].

The data captured on ThingSpeak is stored in user defined channels [11]. Each channel includes eight fields that can receive and store data. There is also three additional fields for location data and one field for status data. Once data is collected in a channel, the ThingSpeak web page can be used to analyze and visualize the data. Recorded data can also be exported in a

spreadsheet format for further analysis by the home owner [12]. Channel settings include the following:

- Channel Name: The user can assign a unique channel name [13].
- Description: Here the user can describe the ThingSpeak channel.
- Field#: UP to 8 data fields can be named and enabled by the user.
- Metadata: Any additional information concerning the channel can be entered as metadata.
- Tags: User identified tags.
- Latitude: The latitude where the installation is located.
- Longitude: The longitude where the installation is located.
- Elevation: Specify the height position of the installation.
- Make Public: A check box to make channel public.
- URL: If the user has a website that contains information about the channel, the URL can be specified here.
- Video ID: If the user has a video that displays the channel information, it can also be specified.

#### IV. ARDUINO

The Arduino microcontroller, that is used in this paper, is an affordable open source platform that is widely used for the development of electronic projects [14]. Some of the advantages that contributed to the success of the Arduino platform are the following:

- Arduino boards are relatively inexpensive compared to other microcontroller platforms[15];
- The Arduino Software runs on Windows, Macintosh OSX and Linux operating systems[16];
- The programming environment is easy to use but also flexible to benefit advanced users[17];
- The software is open source and can be extended by advanced users to suit their specific needs[18]; and
- The hardware is also open source, so experienced users can modify, extend or improve the board [19].

#### V. SWH INSTALLATION

In this section, the SWH hardware installation that is monitored by the designed system is explained. A 150 W SWH was monitored regarding power consumption, as well as inside water and outside air temperature. The solar collectors of this installation can be seen in Fig. 1. Water is circulated between the SWH and the solar collectors with a circulation pump. The SWH is also equipped with an electronic control system where the temperature of the SWH, as well as the on/off times of the electrical element and circulation pump timing, can be set. The timer of this SWH was set to be on for three times during a 24h cycle. The on times were from 02h00 to 04h00, from 07h00 to 08h00 and lastly from 15h00 to 16h00. The set temperature for the SWH was 50°C. The electronic controller was set to allow water to circulate thru the solar collectors as soon as the collector temperature was 5°C higher than the inside

temperature of the SWH. These settings were chosen by the home owner and were not controlled by the researcher.



Fig. 1: Solar heat collectors of monitored SWH system

#### VI. DESIGN OF MONITORING SYSTEM

In this section, the design of the SWH energy and temperature monitoring system is presented. Firstly, an overview of the system will be given, where after the CT, temperature sensors and associated components will be described. The next step is to explain the designed software and calculations that were performed in the Arduino microcontroller. The ESP 8266 Wi-Fi module is then described, where after the setup of the cloud server is discussed. Lastly, the collection of data from the cloud server is presented.

The complete block diagram of the SWH energy monitoring system is shown in Fig. 2. Data that is collected by the system includes the SWH element power, the SWH water temperature and the ambient air temperature next to the SWH. As the system was designed as a non-complex and cost-effective system, only the mains current was measured and not the 230V AC mains voltage, which is standard in SA. Data from the CT (first block in Fig. 2) is passed to the Arduino microcontroller. The RMS current is measured in one-second intervals. The average current for ten measurements is multiplied by a fixed voltage value of 230V to calculate the power value. The temperature inside the SWH as well as the ambient temperature is sampled every 10 seconds and is sensed by two thermistors.

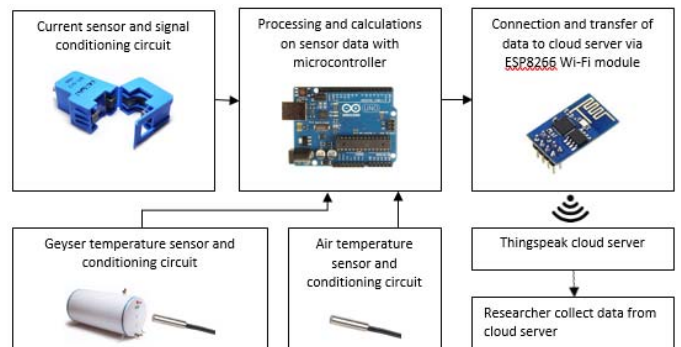


Fig. 2: Block diagram of SWH energy monitoring system

The data from the current and the two temperature sensors is then passed to the ESP8266 Wi-Fi module. The module is programmed via the Arduino microprocessor to connect to a specified Wi-Fi network to enable it to upload data to a cloud server, named ThingSpeak. The researcher can access the cloud server to view or download the data for further analysis.

#### A. The current transformer and temperature sensors

The split core CT sensor that was chosen is rated for 100 A input, giving a corresponding 50 mA output on its secondary side (SCT013). This enables easy installation of the CT sensor around the AC supply line to the SWH element as shown in Fig. 3 (within the dotted circle). Subsequently, the user does not need to disconnect any wires or modify any connections to the SWH or to the electrical distribution board in the home.



Fig. 3: 100 A CT coupled on the live AC wire to the SWH element

The circuit that was used for signal conditioning is shown in Fig. 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 that can be sampled by the analog to digital converter of the Arduino microcontroller. The 10k thermistors are coupled to the analog inputs of the Arduino via a simple potential divider circuit (see Fig. 4). Calculations concerning component values and calibration will follow.

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 to determine a suitable burden resistor which is required by the CT sensor. The burden resistor is dependent on the maximum system current ( $R_3$  in Fig. 4). Although this is a

100 A CT sensor, it may be that the maximum possible system current is less than 100 A.

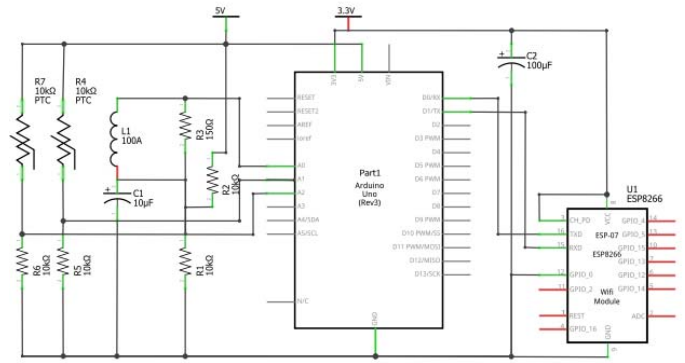


Fig. 4: Circuit diagram of SWH monitoring system

In this paper, the burden resistor is calculated for a maximum current of 25A RMS. The selection of maximum current were done so that it is higher than the 20A breaker connected to the supply of the SWH circuit and therefore ensuring that the primary current to the CT will never exceed 25A. Current will flow through the burden resistor that will result in voltage on the input of the analog to digital converter of the microcontroller. The 25A selected maximum current on the primary 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 ( $R_1$  and  $R_2$  in Fig. 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,  $R_3$ . 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 Fig. 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 25 A).

$$n = \frac{I_s}{I_p} \quad (1)$$

$$= \frac{100A}{50mA}$$

$$= 2000$$

$$I_p(\text{measured}) = \sqrt{2} \times I_{rms} \quad (2)$$

$$= \sqrt{2} \times 25A$$

$$= 35.35A$$

$$\begin{aligned}
 I_s &= \frac{I_p}{n} & (3) \\
 &= \frac{35.35A}{2000} \\
 &= 0.0177A
 \end{aligned}$$

$$\begin{aligned}
 R(\text{burden}) &= \frac{V(\text{sensor})}{I_s} & (4) \\
 &= \frac{2.5V}{0.0177A} \\
 &= 141.2\Omega
 \end{aligned}$$

$$\begin{aligned}
 \text{Calibration value} &= \frac{n}{R(\text{burden}(E12))} & (5) \\
 &= \frac{2000}{150\Omega} \\
 &= 13.3
 \end{aligned}$$

Where:

$n$  = Transformer ratio

$I_s$  = Secondary current

$I_p$  = Primary current

$R(\text{burden})$  = Burden resistor on secondary of C

$V(\text{sensor})$  = Maximum voltage across the CT

#### B. Processing of sensor data

In this section, important aspects of the Arduino's microprocessor code is discussed in the form of a flow diagram.

The Wi-Fi name and password must be defined, as well as the IP address of the cloud server. This is done in the setup part of the flow diagram that is presented in Fig. 5. The key for the specific channel within the cloud server where the sensor data must be uploaded must be specified. This key is obtained by the home owner from his or her account on the ThingSpeak cloud server. The setup also contains the library for the energy calculations as well as the standard mathematical library. The energy calculations done in the library is to convert the measured voltage across the burden resistor to a real time current value.

The calibration factor that was calculated by using Equation (5), as well as the analog ports that are used is also defined in the setup. The three defined analog ports are used on the Arduino microprocessor to connect to the CT sensor (port A0) for the current measurement, as well as the two thermistors that are used to measure the ambient air temperature (port A1) and the SWH water temperature (port A2). The voltage across the burden resistor of the CT is sampled via an analog to digital converter in the Arduino microcontroller. This is done in the third block of the flow diagram in Fig. 5 where a library function is called and the RMS current is calculated from the voltage that is read from the analog port (A0).

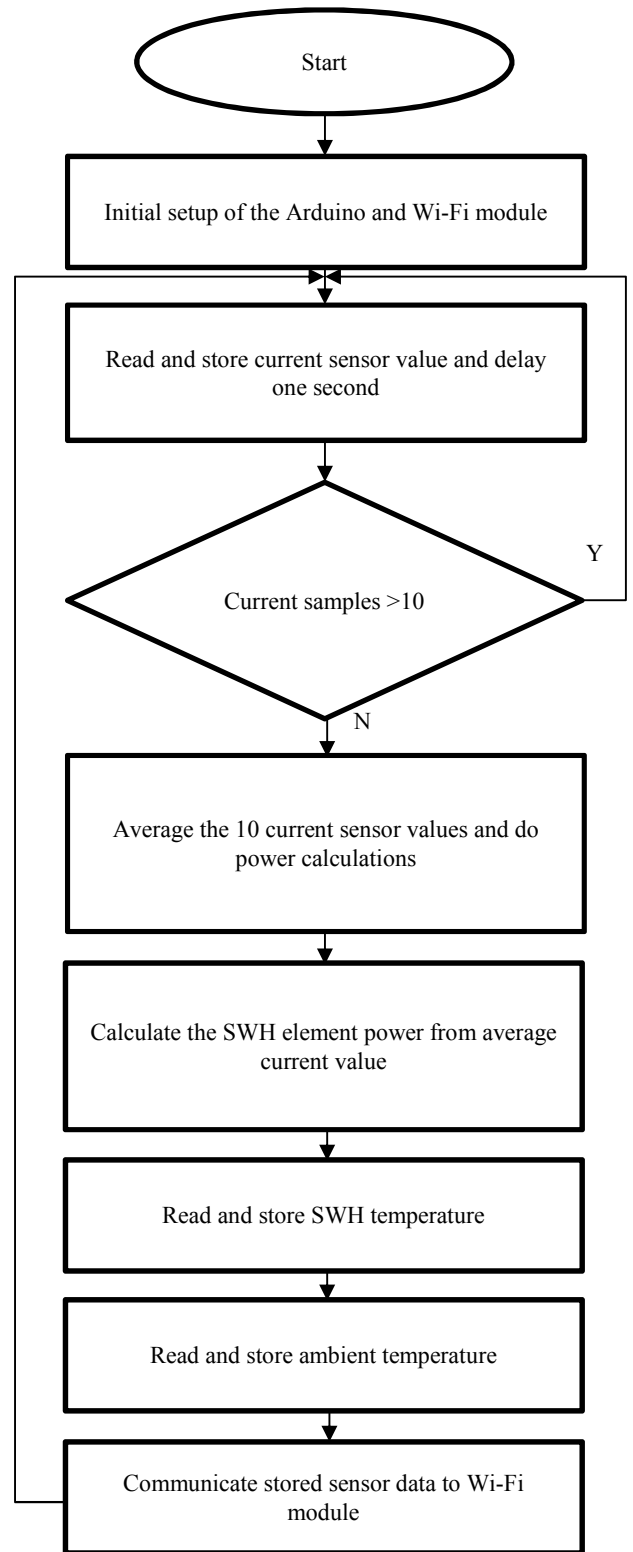


Fig. 5: Flow diagram of Arduino software

Ten readings are taken at one-second intervals (block 3 and 4), and then averaged in block 5 of the flow diagram in Fig. 5. The power calculation is done in block 6 where the average



RMS current and the 230V standard AC voltage are used to calculate the average power over the 10-second interval.

In Block 7 and 8, voltage readings from the thermistor circuits, are taken, and the ambient temperature, as well as the SWH temperatures, are calculated.

### C. Wi-Fi communication and cloud storage

The average power as well as the ambient and the SWH temperature that was obtained in the previous section must now be communicated to the cloud server, ThingSpeak. This is accomplished using the ESP8266 Wi-Fi module. The communication and transmission of the data by the ESP8266 module is done in block 9 of the flow diagram (see Fig. 5). AT commands are used to communicate with the ThingSpeak cloud server via the ESP8266 module. In the ThingSpeak cloud server, the data is captured and recorded and can be retrieved by the home owner.

### D. Example of recorded data

As an example, the recorded data for 4 August 2017 is displayed in Fig. 6. The SWH power is presented on the primary X axis and the outside and inside temperatures of the SWH on the secondary Y axis.

As explained in section V of this paper, the SWH element is switched on by an electronic control system. These switch on times can be observed in Fig. 6 (02h00-04h00, 07h00-08h00 and 15h00-16h00). As the SWH temperature is set to 50°C it can be observed in Fig. 6 that the water temperature will rise to 50°C and the element will then switch off. A rapid fall in water temperature can be observed whenever water is drawn from the SWH (see for instance at 05h28, 11h11 and 18h20). As this system is also a solar system, a steep increase in water temperature from 26.5°C to 39.5°C can be observed from 11h27 to 14h21 (no external electrical power applied). The ambient temperature that was measured next to the SWH inside the tile roof can also be seen in Fig. 6.

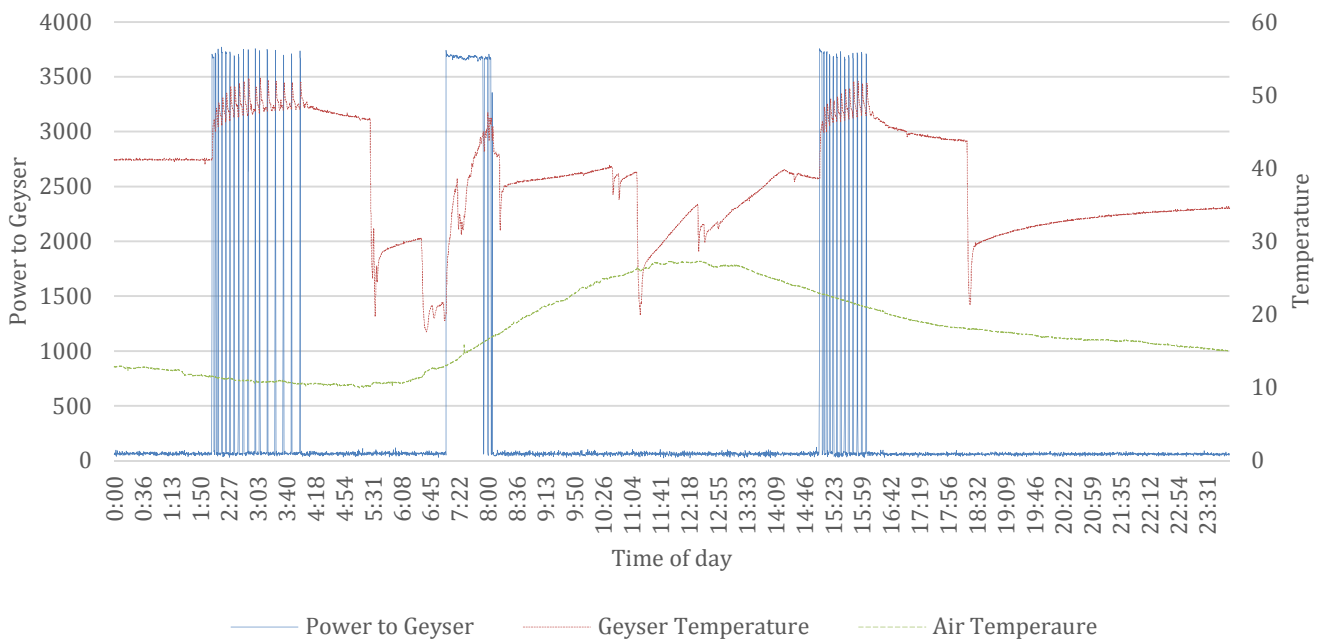


Fig. 6 Example of SWH energy and temperature data that was collected on 4 August 2017

The validity of the recorded RMS current values was confirmed in previous research, and the maximum deviation between the measured and recorded values was 1.6%.

## VII. CONCLUSIONS

The purpose of this paper was to describe the design and development of a non-complex and cost-effective solar SWH monitoring system that is connected to the IoT for data collection. This non-complex system only requires a CT sensor, two temperature sensors, Arduino microprocessor, ESP8266 Wi-Fi module and internet connectivity. The system is also cost-effective, in the sense that it costs less than R1000 to build.

The designed system collected data regarding electrical energy consumption as well as water and air temperatures. The collected data were recorded on a cloud server, called

ThingSpeak, where the user can access the data from any internet connection. Valuable information regarding hot water usage, power consumption and water and air temperature is now accessible and can be used for SWH system optimization and maintenance.

As indicated in literature an optimised SWH can reduce the hot water requirements of a residential home by UP to 70%. With the use of this cost-effective SWH monitoring system, this target is within reach of the residential home owner. Future research may entail the use of the collected data to optimize and to verify optimisation of the installed SWH system.

## REFERENCES

- [1] C. Curry, J. A. Cherni, and M. Mapako, "The potential and reality of the solar water heater programme in South African townships:

- Lessons from the City of Tshwane," *Energy Policy*, vol. 106, pp. 75-84, 2017.
- [2] A. Mostafaeipour, M. Zarezade, H. Goudarzi, M. Rezaei-Shouroki, and M. Qolipour, "Investigating the factors on using the solar water heaters for dry arid regions: A case study," *Renewable and Sustainable Energy Reviews*, vol. 78, pp. 157-166, 2017.
- [3] P. A. F. Ferrer, "Average economic performance of solar water heaters for low density dwellings across South Africa," *Renewable and Sustainable Energy Reviews*, vol. 76, pp. 507-515, 2017.
- [4] J. Gong and K. Sumathy, "Active solar water heating systems," *Advances in Solar Heating and Cooling*, vol. 15, p. 203, 2016.
- [5] A. Di Nisio, T. Di Noia, C. G. C. Carducci, and M. Spadavecchia, "Design of a low cost multipurpose wireless sensor network," in *Measurements & Networking (M&N), 2015 IEEE International Workshop on*, 2015, pp. 1-6.
- [6] B. Bohora, S. Maharjan, and B. R. Shrestha, "IoT Based Smart Home Using Blynk Framework," *Zerone Scholar*, vol. 1, pp. 26-30, 2016.
- [7] P. Sharmila, S. Shobhana, M. Abirami, and U. ESWARAN, "REALIZING INTERNET OF THINGS USING ARDUINO, ESP8266 & IIS SERVER AND MYSQL DB FOR REAL-TIME MONITORING & CONTROLLING MULTIPLE FIRE ALARM SYSTEMS OVER A WIRELESS TCP/IP NETWORK," *Journal on Software Engineering*, vol. 11, 2016.
- [8] F. Montori, L. Bedogni, and L. Bononi, "On the integration of heterogeneous data sources for the collaborative Internet of Things," in *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum on*, 2016, pp. 1-6.
- [9] M. Malagi, "Heath monitoring system based on IoT," 2017.
- [10] A. Carrasquilla-Batista, A. Chacón-Rodríguez, and M. Solórzano-Quintana, "Using IoT resources to enhance the accuracy of overdrain measurements in greenhouse horticulture," in *Central American and Panama Convention (CONCAPAN XXXVI), 2016 IEEE 36th*, 2016, pp. 1-5.
- [11] C. Acharya and S. Kuzhalvaimozhi, "Irrigation and Internet of Things Platform," 2015.
- [12] M. Abo-Zahhad, S. M. Ahmed, M. Farrag, M. F. Ahmed, and A. Ali, "Design and implementation of building energy monitoring and management system based on wireless sensor networks," in *Computer Engineering & Systems (ICCES), 2015 Tenth International Conference on*, 2015, pp. 230-233.
- [13] A. Atabekov, E. Mwangi, and D. Toler, "Smart Chair: An Internet of Things Case Study for a Capstone Research Project," in *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)*, 2015, p. 155.
- [14] C. A. Petry, F. S. Pacheco, D. Lohmann, G. A. Correa, and P. Moura, "Project teaching beyond Physics: Integrating Arduino to the laboratory," in *Technologies Applied to Electronics Teaching (TAEET), 2016*, 2016, pp. 1-6.
- [15] J. P. Grinias, J. T. Whitfield, E. D. Guetschow, and R. T. Kennedy, "An inexpensive, open-source USB Arduino data acquisition device for chemical instrumentation," ed: ACS Publications, 2016.
- [16] R. F. Siregar, R. Syahputra, and M. Y. Mustar, "Human-Robot Interaction Based GUI," *Journal of Electrical Technology UMY*, vol. 1, pp. 10-19, 2017.
- [17] W.-H. Kuan, C.-H. Tseng, S. Chen, and C.-C. Wong, "Development of a computer-assisted instrumentation curriculum for physics students: Using LabVIEW and Arduino platform," *Journal of Science Education and Technology*, vol. 25, pp. 427-438, 2016.
- [18] S. S. More, A. A. Gai, V. S. Sardar, C. S. Rupareliya, and P. T. Talole, "Home Automation on Android Using Arduino," *Journal of Android and IOS Applications and Testing*, vol. 2, 2017.
- [19] F. Mesas-Carrascosa, D. V. Santano, J. Meroño, M. S. de la Orden, and A. García-Ferrer, "Open source hardware to monitor environmental parameters in precision agriculture," *Biosystems Engineering*, vol. 137, pp. 73-83, 2015.