International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I5P205
Volume 69 Issue 5, 25-31, May 2021

An Internet of Things Based Power Consumption Monitoring System

Batlang Mahatlane[1], Sajid M. Sheikh[2], Benjamin C. Molefhi[3]

[1,2,3]*Department of Electrical Engineering, Faculty of Engineering and Technology, University of Botswana, Private Bag UB0061, Gaborone, Botswana*

[1]201501864@ub.ac.bw [2]sheikhsm@ub.ac.bw [3]molefhib@ub.ac.bw

Abstract — *The difficulty of monitoring electricity usage is a hindrance that prevents people from saving power. This is because there is a lack of visual cues or a push for them to realize the rate of their consumption. In this work, we designed and built an external system to monitor power consumption using the power meters. The system uses a NodeMCU microprocessor, and the data is processed and stored in the ThingSpeak cloud service. The device attaches externally to a power meter and uses the LED which flashes, as input. The data is output from ThingSpeak to a web app that is designed to be user-friendly. The data is presented in graphs that plot the rate of usage against time and amount of power used in an equivalent Pula (Botswana Currency) unit. The system has been designed and tested in Botswana on the Botswana Power Corporation (BPC) residential meters.*

Keywords — *BPC, IoT, NodeMCU, ThingSpeak, Power*

I. INTRODUCTION

With the advancement in technology and an increase in population across the world, more people are getting connected to the grid as their lifestyles change [1]. This means more power is consumed, and governments of developing countries are faced with large costs to meet this demand. Electricity consumption has increased by more than 30% in developing countries between 2010 and 2014, and a significant portion of this is wasted [1],[2]. Figure 1 represents a typical modern home that uses electricity for all its needs. About 47% of the needs are for water and space heating. Appliances used for this are geysers and air conditioners which use high power. These are things that a user can easily regulate by turning off when or where not needed if they had an incentive to do so. To do this, people need visual cues, and it has been shown by [3] that they create an important cause-effect link between consumer behavior and power consumption. It is inevitable that measures must be put in place to mitigate electricity wastage. In the case of Botswana, various policies have been adopted to address power wastage. Highlighted by [4], through Botswana Power Corporation (BPC), the government implements various policies to reduce power loss; mass load shedding, educating the public on the importance of saving power, and targeted rationing of power to households. The latter policy was implemented in 2010 using smart meters deployed to major towns of Botswana [5]. However, these implementations lack an important aspect of user control. The smart meters rolled out in 2010 are controlled by BPC, but this does little help in helping the consumers to save power themselves [6]. This targeted load shedding is an intrusive method that does not consider customer comfort. In Botswana, it is difficult for an average user to monitor their rate of power consumption. Customers do not have a way of monitoring their own consumption to be able to make a conscious decision to turn off certain appliances. Thus, they need a

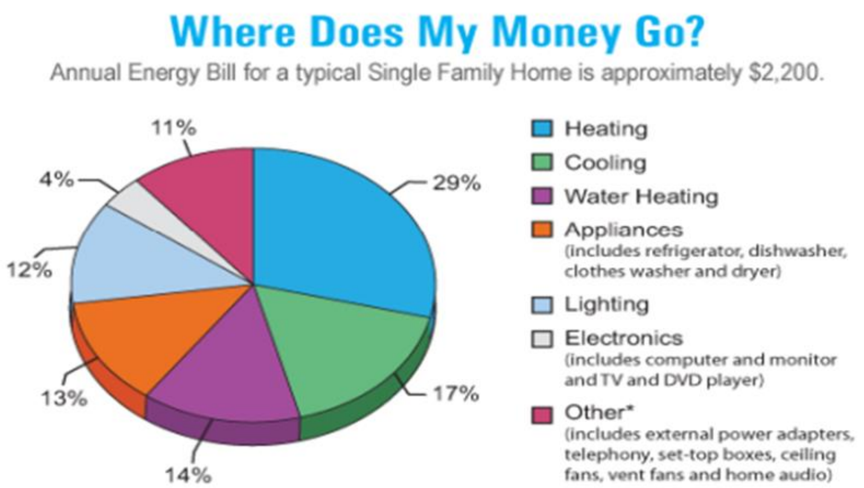

Figure 1 Electrical energy consumption in a typical home [6].

way of monitoring power consumption in their own homes. The current stand-alone meters in most homes only have a pulsing LED as a way of telling the rate of power consumption. The meters also show the remaining power to be used in KWH (kilowatt-hour) - a unit that an average user does not understand. Most people do not understand these features and simply ignore them. Therefore, to make a conscious decision to save power, users need to know their consumption rate in real-time and in units and visual graphics that they can understand. They must receive this information as push notifications/emails on their phones and view it on a device, wherever they are. Integrating a power monitoring device on existing infrastructure will provide many benefits. Users will have more control over their electricity usage. This is because they will be able to tell how much power they are currently using and will decide on what to turn off and when. If the user is unaware of their usage, they will get notifications that prompt them to be aware of their usage. The user will save money for their homesteads, and in turn, the country will also save. The user's data will also be secure and free from outside access or tampering with.

II. LITERATURE REVIEW

A couple of solutions have been provided by various people and institutions before, each targeting a specific need. In the Botswana scene, the deployed smart meters were designed to be controlled by the power utility [1]. Users are not involved in the monitoring of their own power consumption. Furthermore, their data is uploaded to the utility servers, which brings a question of privacy and data security. As [2] explained, power usage data can be used to determine whether a user is at home or not. The proposed device will tackle security and privacy concerns issues by sending encrypted data to the cloud for processing and requiring authentication to access it. The data will also only be accessible to the user. Installing smart meters costs governments, a lot of money. This is because old standalone meters are replaced with newer connected smart meters. The Botswana government installed these in 2010 at a very high cost in some homes [1]. The proposed device will save a lot of money as it does not require replacing old meters (or newer ones). The device will attach to the existing meter externally, requiring no technical skills. The only cost will be of purchasing the device itself. The data generated will be stored and processed on remote servers and will be viewed on a computer or smartphone application. Other innovators have made the following implementations. The current sensor/transformer used by [3] and [7] to [10] introduces a slight error in computing the consumed power. This method requires an assumption of the operational voltage $Vrms$, which varies around 240V (Botswana). The exact power consumed may not be that far from the actual, and this may be accepted since tariffs are calculated based on a range of values. However, it is worth noting that the final calculations are based off an assumed voltage. References [3] and [7] to [10] installed their sensors within the power plugs and meter, respectively. Their methods require tampering with existing infrastructure to get the amount of current passing through. The proposed device should be designed to be easy to install and low cost. Therefore, a different method of sensing power consumption should be considered in Botswana.

III. METHODOLOGY

To monitor the flashing LED, a Light Dependent Resistor (LDR) was used to detect the flashing light. A developer on the Arduino forum [3] noted that an LDR is the best sensor for detecting the red light at a high sampling rate. The limitation of using LDR is that it detects unwanted ambient light. This can be solved, though, by shielding the light collection area to allow only the red flashing light. Other developers had used an InfraRed (IR) sensor, but they found that signal was noisy and required more circuitry to filter.

TABLE 1: WAYS IN WHICH OTHER DEVELOPERS HAVE DESIGNED A POWER MONITORING DEVICE

Paper	Type of input (sensor)	Microprocessor used	Technology used	Cloud service	Output
Savjee [6]	Current transformer	Esp82 Atmega16	Wifi	AWS, dynamo DB	Smartphone app Graph QL PowerBi
Wisam Nader [7]	Current transformer	Atmega16	Wifi	N/A	LCD and computer
Santos [8]	Current transformer	Arduinos	LoRa	Rasberry Pi as a server	Web app
Digiteum [9]	Current transformer	(undisclosed)	(undisclosed)	Digiteum cloud based on AWS	Digiteum smartphone app
Pocero [10]	Current transformer	XBee Arduino pro	LoRa	N/A	LCD

To realize the implementation of this project, a product was designed as a solution to the problem statement. Therefore, some specifications must be met. The device should have the following attributes:

- Low Cost

All components used should be such that the total cost of the device is affordable to most people. Any online services used should be either free or affordable.

- Small Footprint

The device should be as small as possible to be able to fit on the face of most power meters.

- Low Power Consumption

The device will have no physical connections to the meter; hence it will need its own power source. Batteries will, therefore, be used as a source of power.

- Easy to Use.

The device should have as few inputs as possible and require very few setup requirements. It should be plug-and-play with intuitive functions both online and physically.

- Security

User data should be secure. The cloud service used should support some form of password authentication to deter unwanted access.

To help decide on which microprocessors and databases are ideal for this work, a decision matrix was used. A decision matrix is defined as a series of values in columns and rows that allow for a visual comparison of possible solutions by weighing their variables based on the order of importance [4],[5]. It is an ideal decision-making tool if one is debating between a few comparable solutions that each have multiple quantitative criteria. For this purpose, different factors were weighed and scored according to their relative importance to the project. The factors chosen for this work are those outlined above in the specifications. For **Error! Not a valid bookmark self-reference.** and **Error! Reference source not found.**, the specifications are weighed on a score from 1 to 5. A desirable specification is given a maximum of 5, and the least desirable is given a score of 1. The microprocessor/database with the highest total score will be chosen. There is a wide variety of microprocessors available that can be used to implement this project. A decision matrix is used to help choose from the most common cloud services available. Hence the NodeMCU, as shown in **Error! Not a valid bookmark self-reference.**, was chosen as it received the highest score. Cloud services connect edge devices to the end-user application. A decision matrix was used again to determine the best cloud service to use for the implementation of this project. The cloud service should meet the specifications of this project. The highest scoring service/database was chosen. ThingSpeak was chosen as shown in **Error! Reference source not found.**.

TABLE 2: A DECISION MATRIX TABLE TO HELP DECIDE ON WHICH DEVICES TO USE [11]

	Microprocessors					
Specification	**Raspberry Pi**	**Arduino Uno**	**Arduino Nano**	**Arduino Pro Mini**	**NodeMCU**	**Photon**
Size	1	1	3	3	3	3
Price	1	2	4	3	5	3
Friendly IDE	2	3	3	3	3	3
Power consumption	2	2	2	3	3	3
Memory	3	3	2	2	2	3
Total	**9**	**11**	**14**	**14**	**16**	**15**

TABLE 3: A DECISION MATRIX TABLE TO HELP DECIDE ON WHICH CLOUD SERVICE TO USE [12]–[14]

	Cloud Services and Databases			
Specification	**Cloudino**	**VPS**	**ThingSpeak**	**DynamoDB (AWS)**
Security	4	5	5	5
Friendly IDE	3	3	4	4
Price	3	1	5	3
Graphic interface	3	3	5	4
Total	**13**	**12**	***19***	**16**

IV. IMPLEMENTATION

The system will monitor power consumption in homes and report to the customers via a monitoring dashboard on a smartphone or web app. This device is an embedded system with an LDR and a microcontroller, and it stores data on the ThingSpeak cloud. This data will be processed in the cloud and reported back to the user on the app. It attaches to the exterior of an existing meter with the LDR aligned to the blinking LED without interfering with the existing power circuitry within the meter.

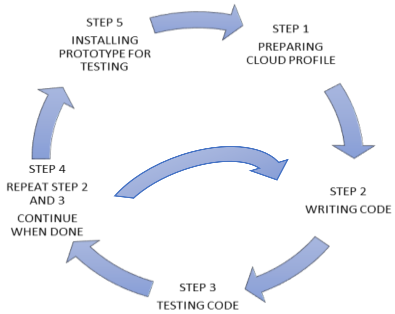

Figure 3 Steps followed in the design of the prototype.

The design process in Figure 2 was followed to design the prototype. The cloud account for uploading data was created first, followed by writing the Arduino code. Finally, the code was tested to see if the data recorded was able to be uploaded. Steps 1 to Step 5 are explained by sections A to D.

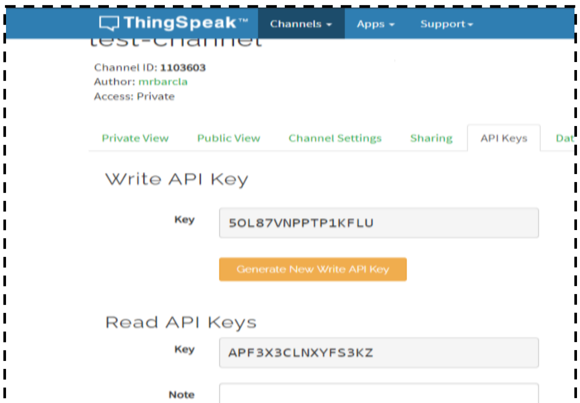

Figure 4 The read and write APIs used to communicate between ThingSpeak and the device.

A. Preparing Cloud Profile:

The first thing was to use an existing MathWorks account to sign in to the ThingSpeak cloud. Then a new ThingSpeak channel named "*test-channel*" was created for data to stream into. ThingSpeak then generated a new ChannelID, a Read API, and a Write API, as shown in **Error! Reference source not found.** in the Arduino project source code. To allow the channel to send email alerts, a MATLAB analysis app was created. It uses the TimeControl app to trigger it at daily intervals. The MATLAB Analysis app in Figure 5 analyses the data to calculate the power used in a day, and the appropriate email message is generated based on power consumption

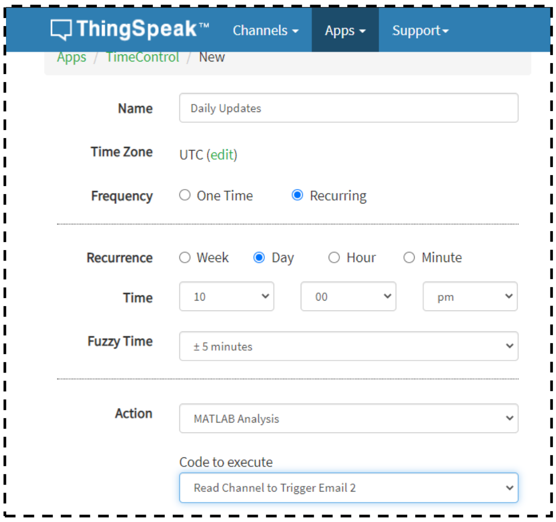

Figure 2 The Time Control app.

data. Channel 1103603 logs power usage data recorded by the device. The code uses this function to make a sum of data entries in the kwh field:

meter_reading=thingSpeakRead(readChannelID,'Fields',kwhField,'NumDays',1,'ReadKey',readAPIKey);

and

Billing_cost = sum(meter_reading);

This code is activated using a TimeControl app. The parameters of this app are shown in Figure 5.

B. Writing Arduino Code:

```
readChannelID = 1103603;
kwhField = 2;
readAPIKey = 'APF3X3CLNXYFS3KZ';
meter_reading =
thingSpeakRead(readChannelID,'Fields',kwhFi
eld,'NumDays',1,'ReadKey',readAPIKey);
% Calculate the Cost
Billing_cost = sum(meter_reading);
display(Billing_cost,'Total Billing Cost
(INR)');

formatSpec = "The Power consumption is:
%d,%d";
A1 = sum(meter_reading);
A2 = meter_reading
apiKey = 'TAKDE7CE5VUP1VZM3RHQG';
alertURL =
"https://api.thingspeak.com/alerts/send";
options = weboptions("HeaderFields",
["ThingSpeak-Alerts-API-Key", apiKey ]);
alertBody = sprintf(formatSpec,A1,A2)
alertSubject = sprintf(" Power consumption
exceeded 5 l!");

if meter_reading >= 5
    webwrite(alertURL, "body", alertBody,
"subject", alertSubject, options);
end
% 5 is used as a random suggestion, to be
edited when there is enough data
```

Figure 5 The MATLAB Analysis app.

The logic of the program is to detect the LED pulse followed by counting these pulses/blinks. From this point

Figure 7 LED power rate indicator on the face of a power meter.

on in this paper, "blinks" will be used to indicate "LED pulses." Once the counter reaches 1000 blinks, the program will start counting the amount of power used in kilowatt-hours. It uses the ratio, 1000 blinks/kwh, as shown in Figure 6.

```
int sensorValue = analogRead(A0);
   if (sensorValue >100){
        blinks++;
        }
        else {
        blinks = blinks;
        }
        if (blinks==1000){
        kwh++;
        //reset blinks
        blinks = 0;
        }
        else {
        blinks = blinks;
        }
```

Error! Reference source not found. is the flowchart that shows the main parts of the Arduino code. There is more going on in between these steps, and the explanations will be shown in subsequent sections. The first loop of connecting to the internet is explained in section C.

C. Connecting To The Internet:

NodeMCU is based on the Expressif ESP8266 architecture, and therefore, to connect to the internet, it will use Wi-Fi. This means that it uses the *ESP8266WiFi* library from the library manager. The code snippet for connecting to the Wi-Fi is shown in the appendix highlighted in blue.

Figure 8 is further explained by the code snippet below: One of the main objectives of the project is to present the consumption rate data in Pula. Hence, the following formula will be used to calculate this rate. Pula = kwh * BPC_Rate;

A. Testing The Code:

Before uploading the code into the NodeMCU, the circuit was connected as shown in Figure 10. After

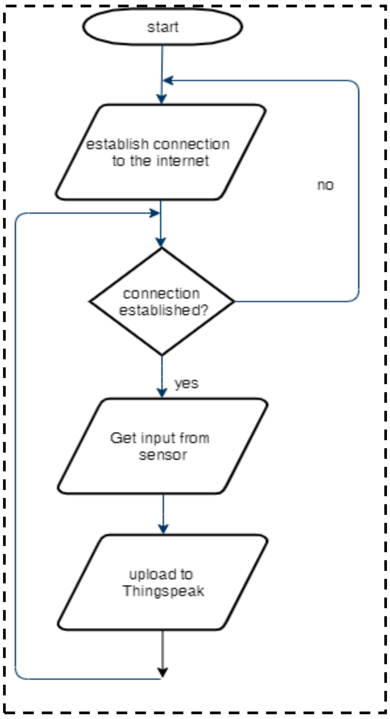

Figure 6 Flowchart logic for the Arduino code.

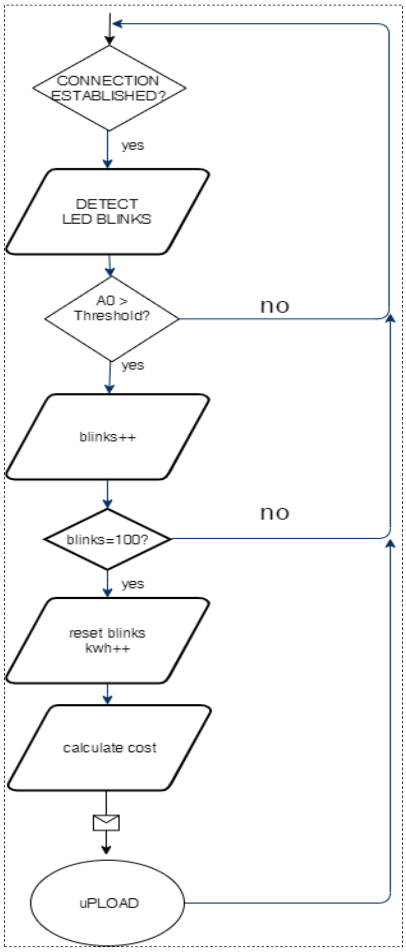

Figure 8 Flowchart showing how to calculate the amount of power used.

uploading, another Arduino board running a blink sketch, with an LED connected to one of its GPIO pins was used to simulate the power meter LED. The two boards were aligned such that their LED and LDR were aligned. During testing, it was noted that for every blink, the NodeMCU would record multiple counts for the duration of the ON time of the LED. This presented a problem because 1 blink would be counted as 15 or 20 blinks, within the short period that the LED is on. Therefore, the solution was to slow down the refresh rate of the loop by introducing a delay. Trial and error were used to determine the optimal delay time to allow the loop to wait for the LED to go off. This was found to be about 500 milliseconds. In the main code, the variable used to represent this time is *sensitivity*. The results of successful runs are shown in Figure 11.

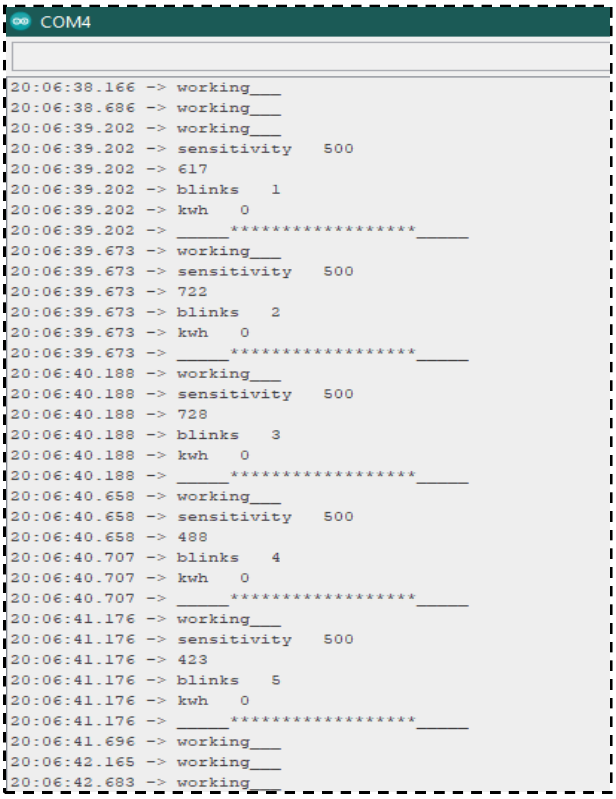

Figure 9 Results when testing the code on the serial monitor.

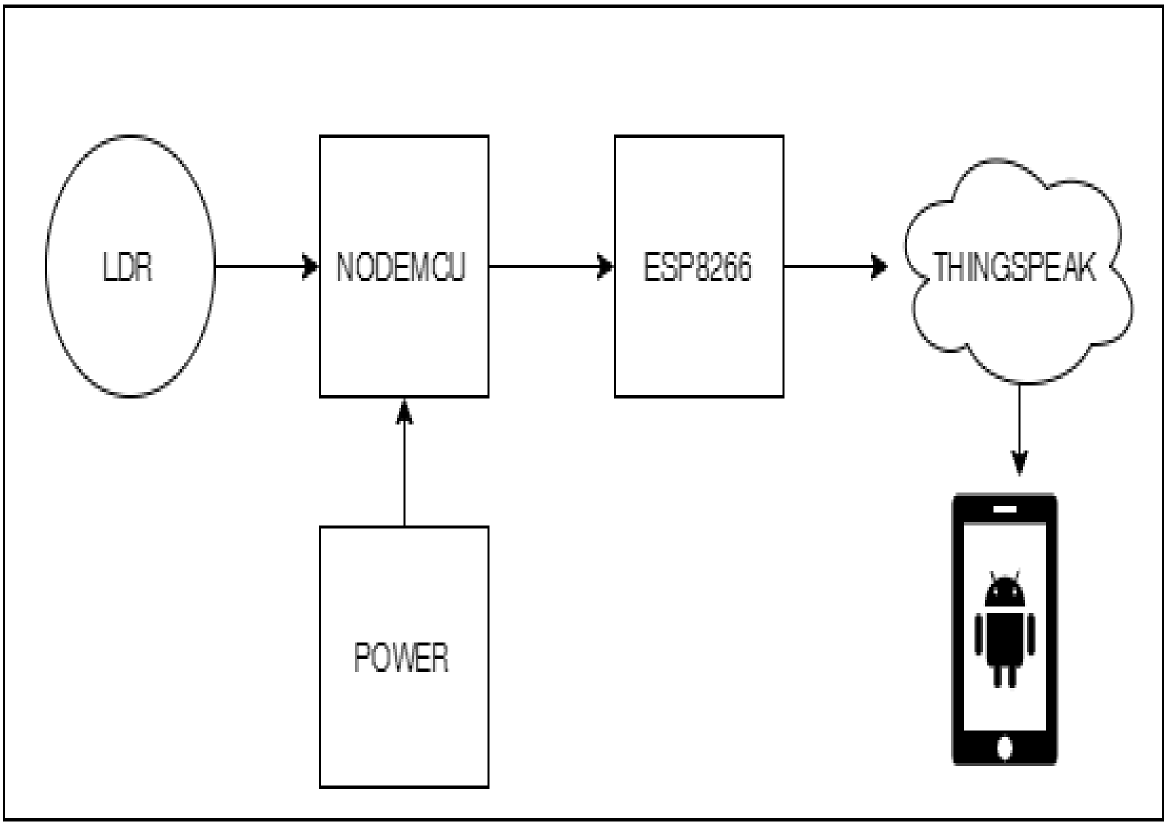

Figure 10 Schematics showing circuit block diagram.

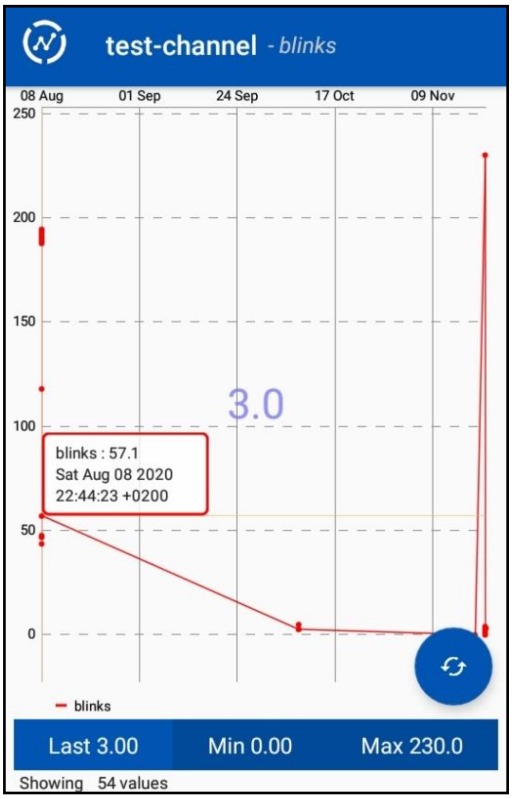

Figure 11 ThingSpeak app receiving data.

V. CONCLUSION

This project solves one of the biggest problems of the twenty-first century of high energy consumption by individuals. It does this by presenting the data using visual aids such as graphs and charts in an understandable unit to a regular person in Botswana. The presentation is in the local currency, "Pula." The system creates consumption awareness to users to stimulate the urge to save electricity. This will help the government to save power as individuals using the system will be more aware of their consumption. The system is inexpensive and easy to install. It requires internet connectivity. The system has been tested for residential use. Business and Industrial meters still need to be tested for technical compatibility.

REFERENCES

[1] B. P. Corporation., smart_meters_brochure.pdf., BPC, Gaborone, (2010) 1–2, [Online]. Available: https://www.bpc.bw/media-site/Brochures/smart_meters_brochure.pdf.
[2] E. Keelson, K. Boateng, and I. Ghansah, A Smart Retrofitted Meter for Developing Countries., Int. J. Comput. Appl., 90(2014), doi: 10.5120/15573-4203.
[3] Home Energy Monitoring via flashing LED on Meter,(2020). https://forum.arduino.cc/index.php?topic=114202.0 (accessed Mar. 09, 2020).
[4] N. Munier, A Strategy for Using Multicriteria Analysis in Decision-Making. Valencia, Spain, (2011).
[5] How to Use a Decision Matrix for Your Business - Business News Daily., https://www.businessnewsdaily.com/6146-decision-matrix.html (accessed Jun. 12, 2020).
[6] X. D. (Savjee)., Home Energy Monitor: ESP32 + CT Sensors + Emonlib., https://savjee.be/2019/07/Home-Energy-Monitor-ESP32-CT-Sensor-Emonlib/ (accessed Mar. 10, 2020).
[7] W. Nader., REAL-TIME POWER MONITORING, HOME AUTOMATION, AND SUSTAINABILITY., University of Nebraska, (2011).
[8] D. Santos and J. C. Ferreira., IoT power monitoring system for smart environments., Sustain., 11(19)(2019), doi: 10.3390/su11195355.
[9] IoT for Energy Monitoring: Power Consumption Monitoring and Control | Digiteum.com., https://www.digiteum.com/portfolio/electricity-consumption-monitoring-remote-control (accessed Jul. 22, 2020).
[10] L. Pocero, D. Amaxilatis, G. Mylonas, and I. Chatzigiannakis, "Open source IoT meter devices for smart and energy-efficient school buildings., HardwareX, 1(2017) 54–67, Apr., doi: 10.1016/j.ohx.2017.02.002.
[11] Farnell., Arduino Datasheets, (2020). [Online]. Available: http://www.farnell.com/grh/?mpn=A000005 ARDUINO&CMP=os_pdf-datasheet.
[12] (Cloudino)., Innovation, cloudino, (2020). cloudino.io.
[13] IoT Analytics - ThingSpeak Internet of Things., https://thingspeak.com/ (accessed Mar. 09, 2020).
[14] ThingSpeak., Apps, www.thingspeak.com., (2020).