Development of Real-Time Energy Monitoring System and Data Log Using NodeMCU ESP 8266 and MYSQL Database

Ezmin Abdullah, Nurain Izzati Shuhaimi, Roslina Mohamad*
Wireless High Speed Network Research Group (WHiSNet),
Universiti Teknologi MARA, Selangor, Malaysia
**ezmin@uitm.edu.my.com*

Nur Amirah Kamaluddin, Nabil M Hidayat, Norhafizah Burham
School of Electrical Engineering, Universiti Teknologi MARA,
Selangor, Malaysia

ABSTRACT

The growth of population and the rise of electrical appliance usage have been increasing the electricity consumption, leaving a lot of the industrial sectors desperate to optimise their energy consumption. The energy meter provided by the service provider for billing is an accumulated energy used type. This meter measures in kilowatt-hour (kWh) which is not suitable for detailed review and analysis. The lack of data measurement and recording for every energy load point lead to incorrect planning and target for managing energy performance. Thus, this project developed an energy monitoring system that aims to provide users with information about the consumption trend in their buildings. In this project, the power sampling in watts or kilowatts has been taken in the distribution board from two different consumption loads. The two load sampling data were transmitted using NodeMCU ESP82266 as the controller to MySQL to store the data in the cloud. As part of this project, the system is provided with a website and a mobile application that allow users to get the real-time energy usage for every load for the entire building. Based on the experimental result using sampling data, the website displayed a live data page providing information on power usage, accumulated kWh, cost in Ringgit Malaysia and the peak power in watt. This project contributes to the existing knowledge of energy monitoring systems by providing detailed energy reviews from two or more load points as well as the capability to store raw data in the

ISSN 1823-5514, eISSN 2550-164X

Received for review: 2020-09-18
Accepted for publication: 2021-01-20
Published: 2021-11-15

cloud which can be retrieved from the website for future planning and management.

Keywords: *MySQL; Monitoring System; NodeMCU; Energy Consumption; Energy Metering*

Introduction

Nowadays, there is a massive movement in installing energy monitoring systems in new smart homes and even in the industries. Automatic lighting systems, temperature control systems, security control systems, and many other functions are developed not only for a stylish lifestyle but also for energy conservation [1].

Based on usual practice, most of the electricity providers have workers collect the data at the premises' energy meters [2]. The collection of data was done manually to provide the number of units consumed (NUC) and the consumption period. The data is then sent to the server for electricity bill generation. These practices are done regularly every month at all premises. Thus, the existing system requires human intervention and is time-consuming in interpreting the energy meter reading of the consumers [3], [4].

The use of cloud computing for the energy monitoring system help users to monitor the data online [5], [6]. The system gathered and analysed the data and then directly send the data through the client's device using the Internet of Things (IoT) [7], [8]. The IoT-based smart energy meter project enables the electricity provider to record the meter readings remotely without human involvement every month and the energy consumption data can be displayed on a webpage [9], [10]. The demand for information traffic has recently matured because of the variety of connected devices and good meters. To satisfy the explosive growth of big data, new technologies are needed to enhance the prevailing metering infrastructure [11].

The Automatic Meter Reading (AMR) system uses SMS as the medium to transfer data to the server where the system generates bills automatically. However, this approach is still limited and inefficient because the SMS will only display the overall energy consumption for the month. Moreover, there is still the possibility of accessibility bottlenecks to isolated places [12]. A system using the existing Electronic Energy Meters was introduced where GPRS is used as the communication medium to get the values of the meter. Since GPRS is more cost-effective than SMS, it is very useful for infrequent updates [13]. It makes it possible to monitor energy meters at a lower cost. Thus, reports of daily consumption can be created and monitored via an Android application and/or web portal [14]. However, this approach only records the total accumulated energy consumption for the whole building for the whole month and cannot be divided into every floor area or any load point.

To address this issue, we introduce an energy monitoring system that could analyse the energy data from different consumption loads. In this project, two different consumption loads were used to record the sampling data. It records the data on average every 30 seconds using a microcontroller and transfers it to the secure repository of the cloud which is MySQL. Finally, in an easy-to-understand interface, a website and a mobile phone display the customer's energy consumption. Users may view their in-house data consumption in real time for the live reading, total usage and total cost. This feature allows users to identify at specific times, the use of excessive energy, and to address wasteful habits.

Methodology

The development of this monitoring system is divided into three main parts. The first part is identifying the whole process of the system and the components being used. This part is explained in the system design and architecture subsection. The next part is the construction of the circuits which is explained in the circuit design subsection. Lastly, the third part is the cloud-based storage that links to the website and mobile app. It is explained in the MySQL database and connection subsection.

System Design and Architecture

Figure 1 shows the flowchart of the whole process while Figure 2 shows the block diagram of the system development. They start with a sensor reading using a CT sensor, then the reading data is managed by the NodeMCU ESP8266 microcontroller. In the microcontroller, the data is translated into several parameters such as power usage, accumulated kWh, cost in Ringgit Malaysia and summarised peak power in watt. Then, the data is transmitted to the cloud MySQL database every 30 seconds. Lastly, the data can be retrieved or displayed via the website or mobile app. This process is continuously repeated to acquire real-time data.

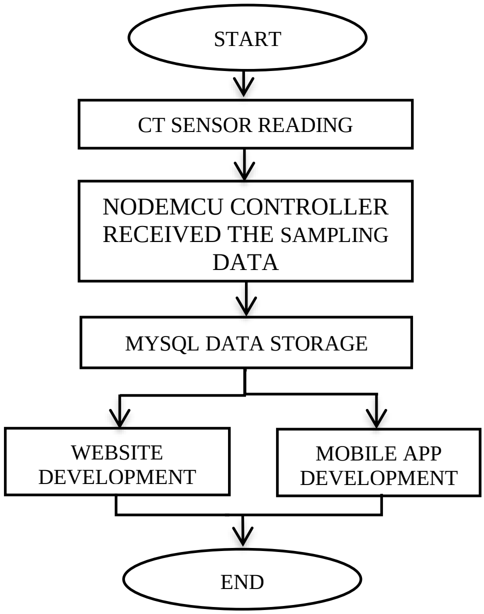


Figure 1: Flowchart of the monitoring system.

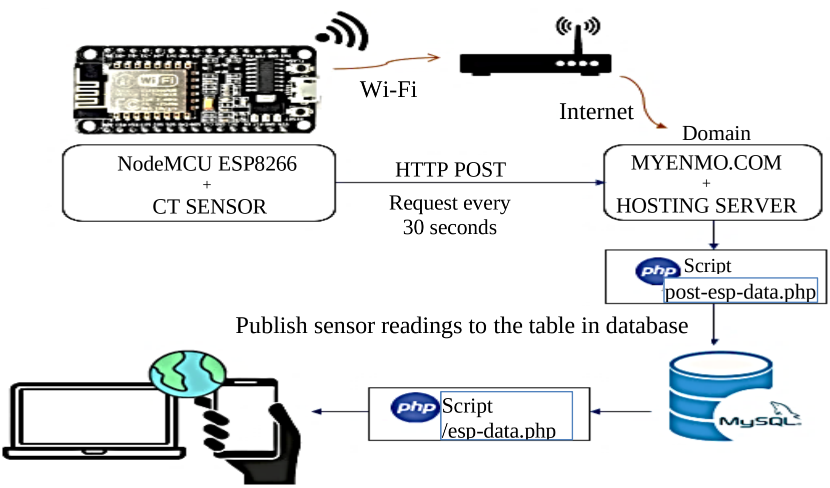


Figure 2: Block diagram of the monitoring system.

Circuit Design

Figure 3 shows the circuit design for the system. The circuit is powered up using 3.3 V and the current transformer is programmed inside the ESP8266. In this project, three CT sensors are used for experiments. However, only two CT sensors were connected to loads. Thus, the reading from two different consumption loads can be retrieved from the system. The NodeMCU ESP 8266 has only one analogue input, therefore, the 7HC 4051 Multiplexor were used to enable more than one CT sensor.

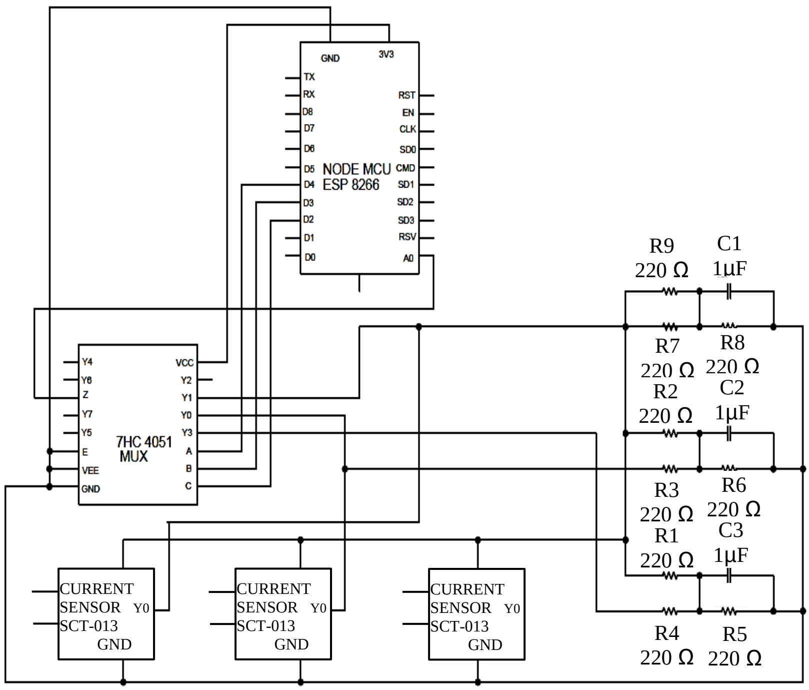
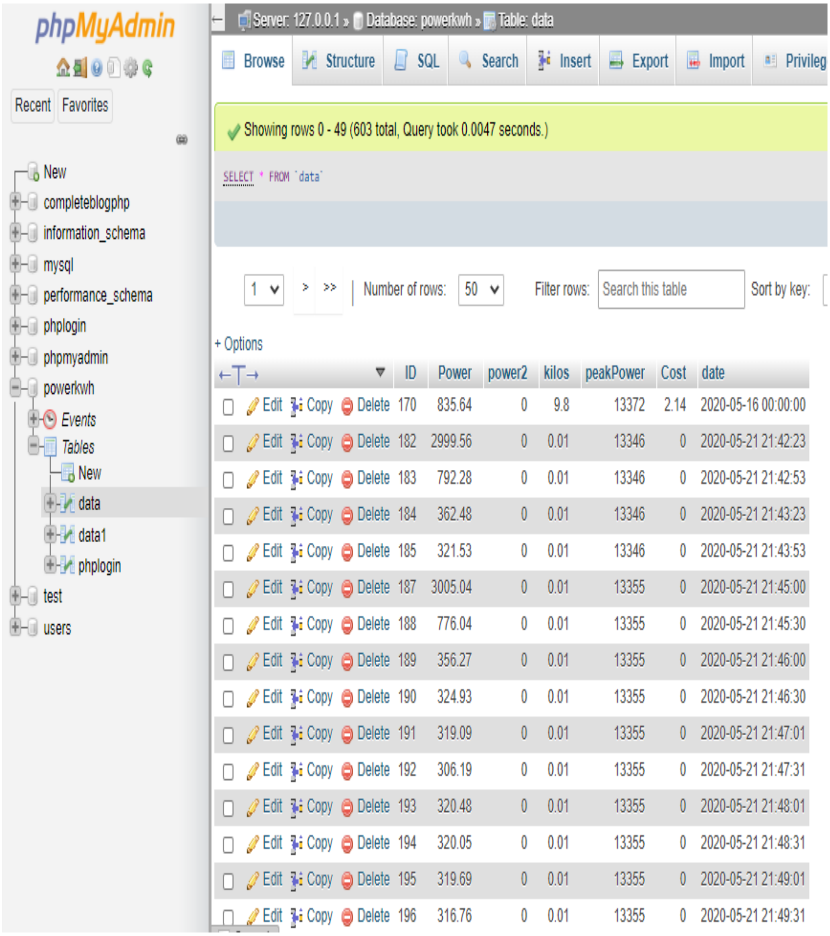


Figure 3: Circuit design for the system development.

MySQL Database and Connections

Figure 4 shows the database which stored the data from the CT sensor. The MySQL database is connected to the webpage to visualise the data in the form of several parameters. By using this database platform, the data can be unlimitedly stored. In this project, the data stored in the database are the power usage, accumulated kWh, cost in Ringgit Malaysia and peak power in watt.



			ID	Power	power2	kilos	peakPower	Cost	date
Edit	Copy	Delete	170	835.64	0	9.8	13372	2.14	2020-05-16 00:00:00
Edit	Copy	Delete	182	2999.56	0	0.01	13346	0	2020-05-21 21:42:23
Edit	Copy	Delete	183	792.28	0	0.01	13346	0	2020-05-21 21:42:53
Edit	Copy	Delete	184	362.48	0	0.01	13346	0	2020-05-21 21:43:23
Edit	Copy	Delete	185	321.53	0	0.01	13346	0	2020-05-21 21:43:53
Edit	Copy	Delete	187	3005.04	0	0.01	13355	0	2020-05-21 21:45:00
Edit	Copy	Delete	188	776.04	0	0.01	13355	0	2020-05-21 21:45:30
Edit	Copy	Delete	189	356.27	0	0.01	13355	0	2020-05-21 21:46:00
Edit	Copy	Delete	190	324.93	0	0.01	13355	0	2020-05-21 21:46:30
Edit	Copy	Delete	191	319.09	0	0.01	13355	0	2020-05-21 21:47:01
Edit	Copy	Delete	192	306.19	0	0.01	13355	0	2020-05-21 21:47:31
Edit	Copy	Delete	193	320.48	0	0.01	13355	0	2020-05-21 21:48:01
Edit	Copy	Delete	194	320.05	0	0.01	13355	0	2020-05-21 21:48:31
Edit	Copy	Delete	195	319.69	0	0.01	13355	0	2020-05-21 21:49:01
Edit	Copy	Delete	196	316.76	0	0.01	13355	0	2020-05-21 21:49:31

Figure 4: MySQL Database.

Figure 5 shows the PHP Script that is used to set up the connection between the MySQL database, mobile application and webpage. First, it is important to make sure the connection is valid. The database is called by using MySQL connect and the connection can be checked at the mobile application whether it failed or succeed. Then, to retrieved the data, SELECT* from the data command is used as in Figure 5. Lastly, the data is fetched and displayed in the form of JavaScript Object Notation (JSON) as a file to be called.

```php
<?php

    //database constants
    define('DB_HOST', 'localhost');
    define('DB_USER', 'root');
    define('DB_PASS', '');
    define('DB_NAME', 'powerkwh');

    //connecting to database and getting the connection object
    $conn = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);

    //Checking if any error occured while connecting
    if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
        die();
    }

    //creating a query
    $stmt = $conn->prepare("SELECT id, kwh, date FROM data1;");

    //executing the query
    $stmt->execute();

    //binding results to the query
    $stmt->bind_result($id, $kwh, $date);

    $data1 = array();

    //traversing through all the result
    while($stmt->fetch()){
        $temp = array();
        $temp['id'] = $id;
        $temp['kwh'] = $kwh;
        $temp['date'] = $date;
        array_push($data1, $temp);
    }

    //displaying the result in json format
    echo json_encode($data1);

?>
```

Figure 5: PHP script for database connection.

Meanwhile, inside the Android Studio for mobile apps, as shown in Figure 6, the t_test. php file is requested using string URL to parse the data, while Figure 7 shows the command to fetch the data from the database to the webpage.

```java
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    pd = new ProgressDialog( context: MainActivity.this);
    pd.setMessage("loading");

    chart = (BarChart) findViewById(R.id.chart);
    load_data_from_server();
}

public void load_data_from_server() {
    pd.show();
    String url = "http://192.168.43.248/apps/t_test.php";
    xAxis1 = new ArrayList<>();
    yAxis = null;
    yValues = new ArrayList<>();
```

Figure 6: Mobile app: Retrieved data from the database.

```php
<?php
include ("koneksi.php");
?>



<?php
  $x_tanggal  = mysqli_query($konek, 'SELECT date FROM ( SELECT * FROM data ORDER BY id DESC LIMIT 50) Var1 ORDER BY ID ASC');
  $y_metana   = mysqli_query($konek, 'SELECT Power FROM ( SELECT * FROM data ORDER BY id DESC LIMIT 50) Var1 ORDER BY ID ASC');
  $y_power   = mysqli_query($konek, 'SELECT power2 FROM ( SELECT * FROM data ORDER BY id DESC LIMIT 50) Var1 ORDER BY ID ASC');
  $y_cost   = mysqli_query($konek, 'SELECT Cost FROM ( SELECT * FROM data ORDER BY id DESC LIMIT 50) Var1 ORDER BY ID ASC');

  ?>

  <div class="panel panel-primary">
    <div class="panel-heading">
      <h3 class="panel-title"><center>Reading for Today</h3>
    </div>
```

Figure 7: Webpage: Retrieved data from the database.

Results and Discussion

In this project, the experiment focused on the development of the system rather than the analysis of the data. Therefore, in this section, the results are about the webpage interface, mobile app interface and sampling data, each explained in their own subsection.

Webpage Interface

This section demonstrates some pages of the developed webpage interface. Figure 8 shows the landing page of the website. Users need to log in to their respective accounts. Thus, the username and password must be registered using the mobile application to request the meter data from the cloud database.

Figure 8: Landing page of the monitoring system.

After login, the HOME page displays the energy consumption graph as shown in Figure 9. The data is provided in real time as it is refreshed every 30 seconds. On this page, the first graph shows the 24 hours live energy reading. In this experiment, a 120-watt refrigerator and an 80-watt lamp were used to provide the sample reading. The second graph shows the last 30 days of accumulated energy consumption together with a table showing the summary of 24 hours energy consumption in terms of peak power, power usage, cost and accumulated kWh.

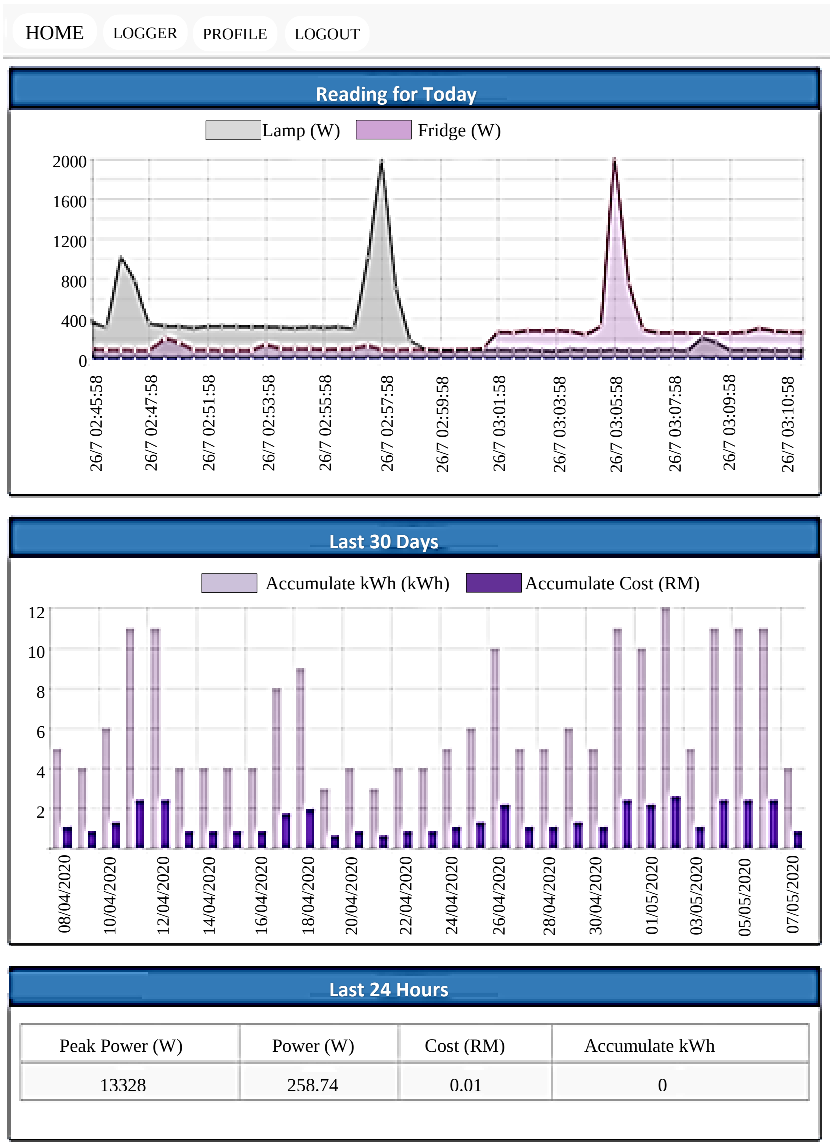
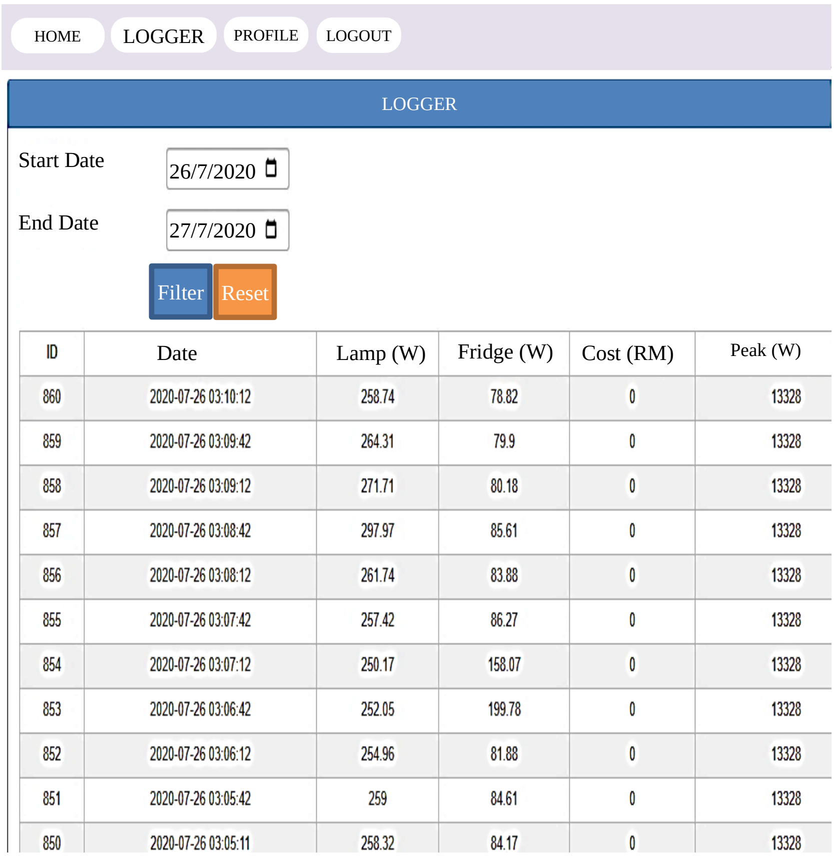


Figure 9: Energy Monitoring Screen in the webpage.

On the LOGGER page, the user can retrieve the data by choosing the start and end dates. As for validation, the data log for this project is taken from two different consumption loads. Thus, the display provided two consumption load readings of the lamp and refrigerator, the peak power, and the total cost as shown in Figure 10.

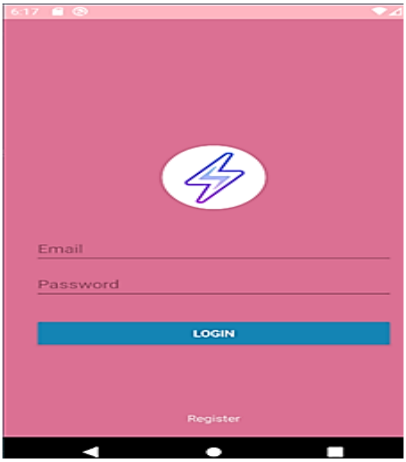


ID	Date	Lamp (W)	Fridge (W)	Cost (RM)	Peak (W)
860	2020-07-26 03:10:12	258.74	78.82	0	13328
859	2020-07-26 03:09:42	264.31	79.9	0	13328
858	2020-07-26 03:09:12	271.71	80.18	0	13328
857	2020-07-26 03:08:42	297.97	85.61	0	13328
856	2020-07-26 03:08:12	261.74	83.88	0	13328
855	2020-07-26 03:07:42	257.42	86.27	0	13328
854	2020-07-26 03:07:12	250.17	158.07	0	13328
853	2020-07-26 03:06:42	252.05	199.78	0	13328
852	2020-07-26 03:06:12	254.96	81.88	0	13328
851	2020-07-26 03:05:42	259	84.61	0	13328
850	2020-07-26 03:05:11	258.32	84.17	0	13328

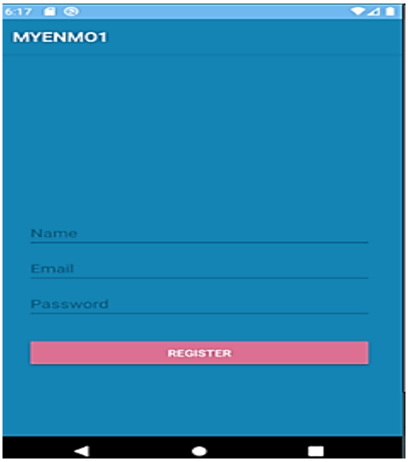
Figure 10: Logger data consumption in the webpage.

Mobile Application Interface

This section demonstrates some pages of the developed mobile application interface. Figure 11(a) shows the landing page for the mobile app. For the first time, users need to register before using the application as in Figure 11(b). For the registered users; their account can be accessed after login. After login, the next page shows four tabs to be selected which are live, statistics, reading and sign out tabs as shown in Figure 11(c). Figure 11(d) shows the table summary of 24-hour energy consumption information such as peak power, power usage, cost and accumulated kWh.

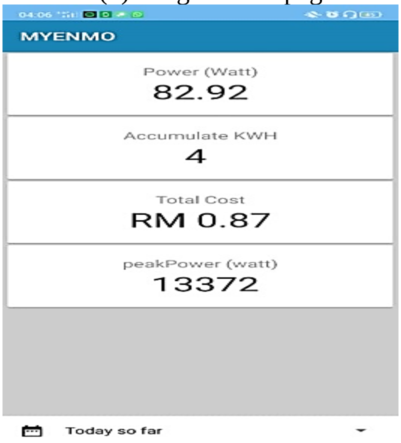


(a) Landing page



(b) Registration page

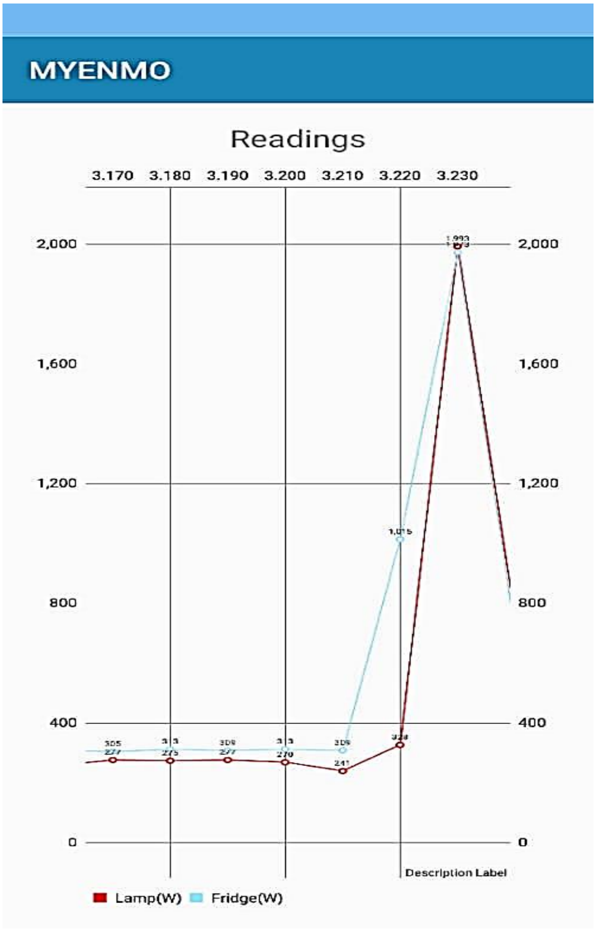
(c) Menu tabs



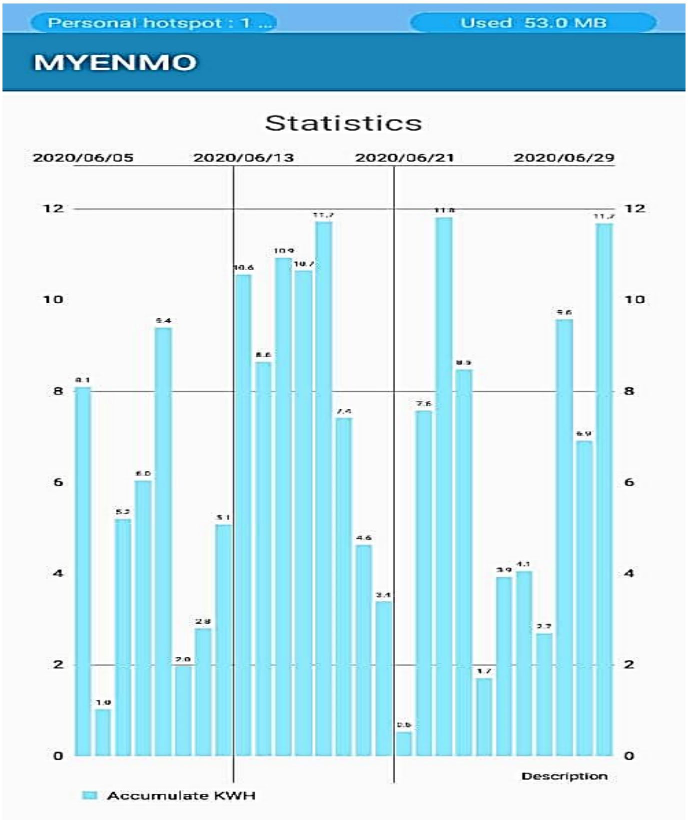
(d) Live tab

Figure 11: Mobile app interfaces.

Figure 12 shows the mobile app's display data. Figure 12 (a) shows the reading tab page where the power is displayed in real time as a line graph for users to observe the consumption pattern. The two lines present in the figure are the reading from two consumption loads labelled as lamp and fridge. Lastly, Figure 12(b) shows the statistics tab page which displays the bar chart of the last 30 days of accumulated energy consumption.



(a) Reading tab



(b) Statistic tab

Figure 12: Mobile App Display.

Sampling Data

The first experiment was carried out by recording two types of consumption loads for about 24 minutes for live data to see the trend of energy consumption for the lamp and refrigerator. Figure 13 shows the power consumption of a 120-watt refrigerator. The monitoring system was able to sample the consumption details up to 24 minutes 14 seconds, hence the arbitrary consumption trend [15] shown in the figure. High consumption spikes were recorded for certain equipment in the system (i.e. fridge compressor) to turn on to reach the target temperature and then stay low when the target temperature was achieved [16], [17]. While for the lamp, due to the condition of the test rig in which other protection devices such as LPF filtering was not installed during the experiment, instantaneous readings were recorded resulting in current spikes and swells [18], [19] that were also recorded and shown in the figure.

Figure 14 shows the day-wise energy consumption graph for one month reading. The time interval of each collected data is 24 hours. This proved that the proposed device is able to capture the intended energy consumption. This approach will prove useful in expanding our understanding of how energy consumption will affect cost [20]. The time iteration to collect the consumption data was set at the optimum interval of 30 seconds [21].

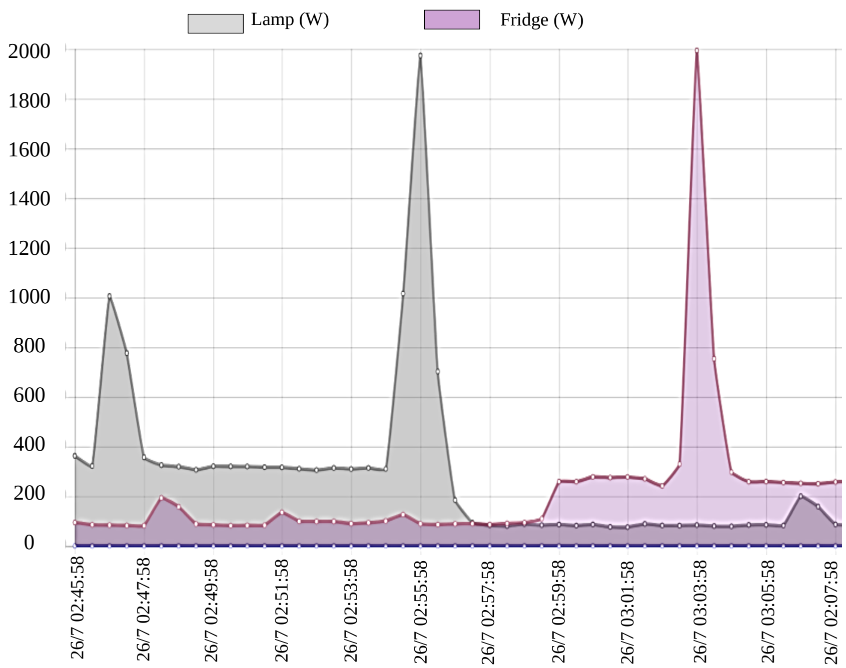


Figure 13: Live energy consumption graph.

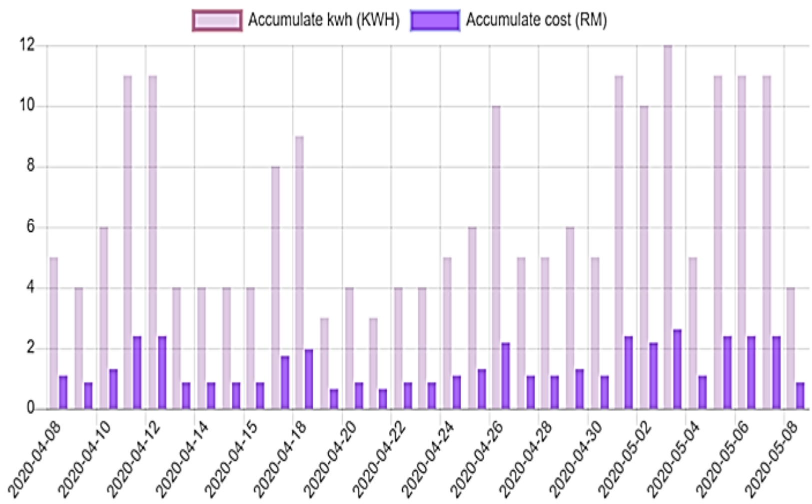


Figure 14: Day wise energy consumption graph.

Conclusion

This paper presents the design and programming for an energy monitoring system. The system was able to display the energy consumption in real time. The objective to design a system that can monitor energy consumption and store energy data in the cloud was successfully tested and achieved. Moreover, users would be able to monitor all the energy consumption using a mobile application or through a webpage, since the system could check on both applications. For future recommendations, we would recommend using ESP32 or the other microcontrollers that have a Wi-Fi connection and extra analogue inputs. This would allow the system to add more current sensors to get the data from multiple consumption loads.

Acknowledgement

This work was supported in part by 600-RMC/DANA 5/3/BESTARI (TD) (001/2019).

References

[1] B. Mubdir, A. Al-Hindawi, S. Hussein, and H. Al-Rizzo, “Smart Energy

Monitoring and Control System Based on Wireless Communication," *Int. J. Eng. Adv. Technol.*, vol. 3, no. October, pp. 1–6, 2013.

[2] D. S. A. Joshi, S. Kolvekar, Y. R. Raj, and S. S. Singh, "IoT Based Smart Energy Meter," *Bonfring Int. J. Res. Commun. Eng.*, vol. 6, no. Special Issue, pp. 89–91, 2016.

[3] A. J. Abid and A. Hussein Ali, "Smart Monitoring of the Consumption of Home Electrical Energy," *Int. J. Comput. Trends Technol.*, vol. 47, no. 2, pp. 142–148, 2017.

[4] L. Pocero, D. Amaxilatis, G. Mylonas, and I. Chatzigiannakis, "Open Source IoT Meter Devices for Smart and Energy-Efficient School Buildings," *HardwareX*, vol. 1, pp. 54–67, 2017.

[5] R. Govindarajan, S. Meikandasivam, and D. Vijayakumar, "Cloud Computing Based Smart Energy Monitoring System," *Int. J. Sci. Technol. Res.*, vol. 8, no. 10, pp. 886–890, 2019.

[6] R. L. R. L. Babu, D. Rambabu, A. Rajesh Naidu, R. D. Prasad, and P. Gopi Krishna, "IoT Enabled Solar Power Monitoring System," *Int. J. Eng. Technol.*, vol. 7, no. 3.12 Special Issue 12, pp. 526–530, 2018.

[7] J. G. Josué, J. M. Pina, and M. V. Neves, "Home Electric Energy Monitoring System: Design And Prototyping," *IFIP Adv. Inf. Commun. Technol.*, vol. 349 AICT, no. September, pp. 437–444, 2011.

[8] M. R. Islam, S. Sarker, M. S. Mazumder, and M. R. Ranim, "An IoT based Real-time Low Cost Smart Energy Meter Monitoring System using Android Application," vol. 4, pp. 1–7, 2020, [Online]. Available: http://arxiv.org/abs/2001.10350.

[9] D. I. N and K. A. R. Rao, "IoT Based Electricity Energy Meter Reading , Theft Detection and Disconnection using PLC modem and Power optimization," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 4, no. 7, pp. 6482–6491, 2015.

[10] B. R., M. M., and P. Kumari, "Power Consumption Monitoring System using IOT," *Int. J. Comput. Appl.*, vol. 173, no. 5, pp. 23–25, 2017.

[11] K. Zaouali, M. L. Ammari, A. Choueib, and R. Bouallegue, "IoT Digital Enterprise Platform based on Cloud Metering and Big Data Services," in *The Twelfth International Conference on Digital Society and eGovernments (ICDS 2018)*, 2018, pp. 17–23.

[12] N. T. Surajudeen-Bakinde, S. O. Ayodele, T. David Oloruntoba, A. O. Otuoze, and N. Faruk, "Development of an Internet based Prepaid Energy Meter," *2017 IEEE AFRICON Sci. Technol. Innov. Africa, AFRICON 2017*, no. September, pp. 1370–1373, 2017.

[13] Dinesh Prasanth M K, "Live Energy Meter Reading and Billing System through GPRS," *Int. J. Eng. Res.*, vol. V4, no. 05, pp. 225–229, 2015.

[14] L. Charaabi, "Open Monitoring System For Photovoltaic Solar Installations," *6th IEEE Int. Energy Conf. ENERGYCon 2020*, pp. 1068–1071, 2020.

[15] M. Gotz, S. Khriji, R. Cheour, W. Arief, and O. Kanoun, "Benchmarking-

Based Investigation on Energy Efficiency of Low-Power Microcontrollers," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 7505–7512, 2020.

[16] T. Xia, T. Wu, J. Li, S. Zhu, Y. Zhang, and B. Ruan, "A distributed Coordination Scheme For Periodic Loads For Demand Side Management," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration: Connecting the Grids Towards a Low-Carbon High-Efficiency Energy System, EI2 2020*, 2020, pp. 2439–2444.

[17] A. Biglia, A. J. Gemmell, H. J. Foster, and J. A. Evans, "Energy Performance of Domestic Cold Appliances In Laboratory And Home Environments," *Energy*, vol. 204, p. 117932, 2020.

[18] U. M. Qureshi, Z. Umair, and G. P. Hancke, "Evaluating the Implications of Varying Bluetooth Low Energy (BLE) Transmission Power Levels on Wireless Indoor Localization Accuracy And Precision," *Sensors (Switzerland)*, vol. 19, no. 15, pp. 1–28, 2019.

[19] W. Liu, Y. Wang, Z. Yang, H. Ma, and W. Wen, "A Novel Two Stage LED Driver Compatible With Electronic Transformers for MR16 Lamp," in *IEEE International Symposium on Industrial Electronics*, 2014, pp. 380–385.

[20] M. Del Mar Solà, A. De Ayala, and I. Galarraga, "The Effect of Providing Monetary Information on Energy Savings for Household Appliances: A Field Trial in Spain," *J. Consum. Policy*, vol. 44, no. 2, pp. 279–310, 2021.

[21] Albert Thumann, T. Niehus, and W. J. Younger, *Handbook of Energy Audits*, 9th Editio. River Publishers, 2013.