

Home Switching using IoT System via Telegram and Web User Interface

RidzaAzri Ramlee, Eric Law Chee Yong, Siva Kumar Subramaniam, AsemKhmag, Ahmad ShukriFazil Rahman

Abstract: *Nowadays, the Internet of Things (IoT) becomes ubiquitous in engineering field and is perceived as paradigms for applications such as a home automation system where data can be exchanged and shared easily across the Internet. This project focuses on the implementation of the home automation system with Telegram using Wireless-Fidelity (Wi-Fi) network access on any smart-phone and laptop devices. It is highly beneficial for elderly, physically-challenged people and travellers to control home appliances effortlessly. This system aims to solve the limitation faced by the technology used in the existing home automation system such as Bluetooth -distance issue, Global Standard for Mobile Communication (GSM) – cost issue and Zigbee – power and bandwidth constraint. In fact, this system provides an interactive graphical user interface (GUI) on both web and Telegram platform where data can be interchanged and synchronized between both GUI in which sensor's readings and status of home appliances can be monitored and manipulated consistently. The analytical findings based on temperature sensor's accuracy and system latency showed that the performance of the system is viable, and the system proved to be more prominent than other home automation systems.*

Keywords: *NodeMCU, Telegram, DHT 22, Home Automation system, database, IoT.*

I. INTRODUCTION

Generally, the home appliances are either being controlled manually or with the aid of existing communication technology such as Bluetooth[1]–[3], GSM [4]–[6] and Zigbee[7]–[10]. These methods, however, have certain shortcomings such as the range of distance, low data transfer

Revised Manuscript Received on August 29, 2019.

RidzaAzri Ramlee, Electronic, Faculty of Electronics and Computer Engineering (FKEKK), Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia. Email: ridza@utem.edu.my.

Eric Law Chee Yong, Electronic, Faculty of Electronics and Computer Engineering (FKEKK), Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia. Email: ericlaw95@gmail.com

Siva Kumar Subramaniam, Electronic, Faculty of Electronics and Computer Engineering (FKEKK), Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia. Email: siva@utem.edu.my

Asem Khmag, Department of Computer Systems Engineering, University of Zawia, Libya. Email: khmag2002@gmail.com

Ahmad Shukri Fazil Rahman, Pusat Pengajian Kejuruteraan Sistem Elektrik, Universiti Malaysia Perlis, Kampus Pauh Putra, Perlis

rate and call tariff. For instance, Bluetooth's network is restricted to a specified range of distance [11][12] while GSM inflicted short messaging service (SMS) charge which can be costly over time. Regardless, Zigbee has lower transfer rate and not ideal for remote monitoring application. Moreover, the limitation in Bluetooth and Zigbee technology pose explicit disadvantages especially for elderly or physically challenged people to access home appliances and travellers to monitor and control home appliances remotely.

Thus, one of the solution is by using the IoT based home switching system [13], [14]. In this paper the IoT based home switching system via Telegram is proposed, analyzed and implemented. [15] Equipped with data encryption and ability to create Telegram Bot, Telegram messenger can be integrated with Arduino compatible Wi-Fi board (NodeMCU) to allow people to machine communication, which can certainly manipulate and monitor the status of home appliances with Wi-Fi access. Besides, Telegram Bot API also made the platform more dynamic and user friendly due to its versatile supports in various programming languages such as Python, C, and Java.

The project is comprised of both hardware and software parts. The software part for integration of Telegram and NodeMCU is conducted with C language on Arduino IDE, whereas web user interface (web UI) is built with HTML 5, Bootstrap, JQuery and php programming language[12]–[15]. For hardware development, the components are constructed on breadboard and interfaced with NodeMCU to test the feasibility of the system.

II. METHODOLOGY

The general block diagram of the IoT based home switching system using the Telegram, shown in Fig.1. Main control panel will monitor and parse incoming requests from Telegram user via Wi-Fi network to decide corresponding action (measuring ambient temperature/ humidity or controlling home appliances). Ambient temperature and humidity will be measured by DHT 22 regularly and the data will be pushed to mysql database for data-logging.

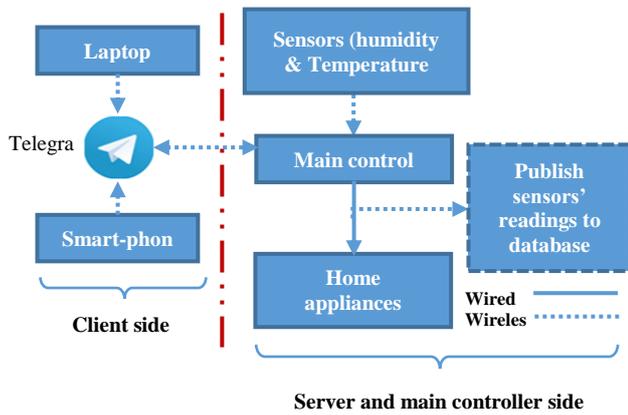


Fig.1.General block diagram of the system

This system is designed and installed at housing area, specifically to facilitate physically challenged people or elderly' welfare and concerned travellers about their houses' conditions. Due to user-friendly and dynamic features of IoT platform across the Internet, data can be retrieved and shared easily and therefore, the designed system is more flexible and suitable for people of all age.

This system aims to create a border-less communication between people and machines and thus, emphasizes on carrying out wireless remote control access. In this case, two methods are approached to interface with the main control panel such as Telegram and Website interface.

The Telegram messenger is used as the main framework to initiate HTTP communication on the client side and connect it with server and controller side. It can be accessed from both laptop and smart-phon wirelessly. Therefore, users can simply enter a predefined command to forward to main control panel where it will decide for the further actions such as to trigger the state of the actuator (home appliances).

The second method to mention here is to interface via the web UI. The designed web UI is supported and accessible on browsers of both laptop and smart-phon. Moreover, web UI application is run and hosted on a local server (Wampserver) or on free domain (000webhost.com). This provide a platform to make the system more user-friendly and dynamic to use. Thus, redirect users to the designed page and provide access on the widgets on the page to control home appliances. It also as alternative to access the home switching system if either one fails to function properly.

A. Hardware Design

Hardware design of the system will focus on the construction of the main control panel along with its peripherals. Fig. 2 demonstrated the layout of the hardware block diagram of the system. The main control panel consisted of NodeMCU, DHT 22 sensor, relay module, voltage regulators and actuators.

The main reason that NodeMCU is used in this system is due to its capability to connect to the Internet and its power saving feature. ESP8266 draws only $60\mu A$ in deep sleep mode and it can stay in low power standby mode unless Wi-Fi network is required for communication[16]. Apart from that, NodeMCU is also inexpensive, Arduino compatible (can be programmed in Lua or C) and has vast supports in Arduino community. Compared with Arduino UNO without Wi-Fi module, a

Wi-Fi shield or Ethernet shield is needed to access the Internet.

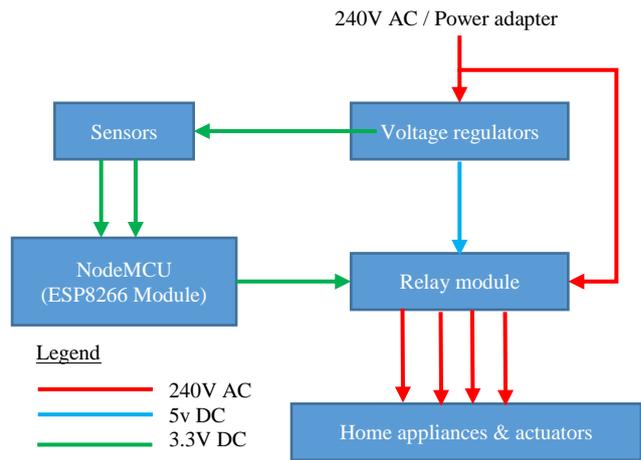


Fig.2.Hardware block diagram of system

There is, however, a noticeable limitation on NodeMCU, where only one analogue pin ($A0$) is manufactured on the board.

DHT22 is chosen because of its capability to measure ambient humidity and temperature. The reasons that it is used instead of the thermistor are because it is a dual-functional component and easy to set up in NodeMCU by just importing Adafruit library. The only limitation of this component is that it is not water-resistant[17].

B. Wiring connection

The electric current of the whole system is supplied from the power outlet of $240V_{AC}$. As the voltage specification of each device is different, it is necessary to regulate the voltage accordingly to the several voltage ratings ($3.3 V_{DC}$, $5 V_{DC}$ and $240 V_{AC}$). Each of the components presented in the wiring connection in Fig. 2, they are classified and denoted by red, blue and green color respectively.

Table-I: Voltage rating of each component

Voltage rating	Component
$5 V_{DC} / 240 V_{AC}$	4 channel relay module
$3.3 V_{DC}$	NodeMCU, DHT22, servo motor

Based on Table 1, since there are various voltage ratings in our system, voltage regulator circuit is required to regulate voltage from power source ($240 V_{AC}$) to $3.3 V_{DC}$ and $5 V_{DC}$. The voltage regulator circuit is constructed with step-down transformer, rectifier, filter, and IC regulator.

C.Algorithm's design

The Algorithm's design will illustrate the main system interaction with the both GUI (Telegram and web UI) in flowchart. As depicted in Fig. 3, the system will connect to Wi-Fi network first and then will check for routines with its built in timer such as interaction of users and Telegram, publishing of sensors' readings and synchronization of Telegram and web UI.

Basically, the system will monitor whether there is request from Telegram user via its getUpdates() function and will parse the messages to carry out the intended work. After that, sensor's parameters (humidity and temperature) are measured within every hour and passed as query string parameters along the http get request to be stored in the mysql database.

In addition, the sensor readings are also retrieved from database and displayed on the web UI for the purposes of illustration and monitoring. Last but not least, both Telegram and web UI's routines are synchronized to each other so that the current output states are up to date.

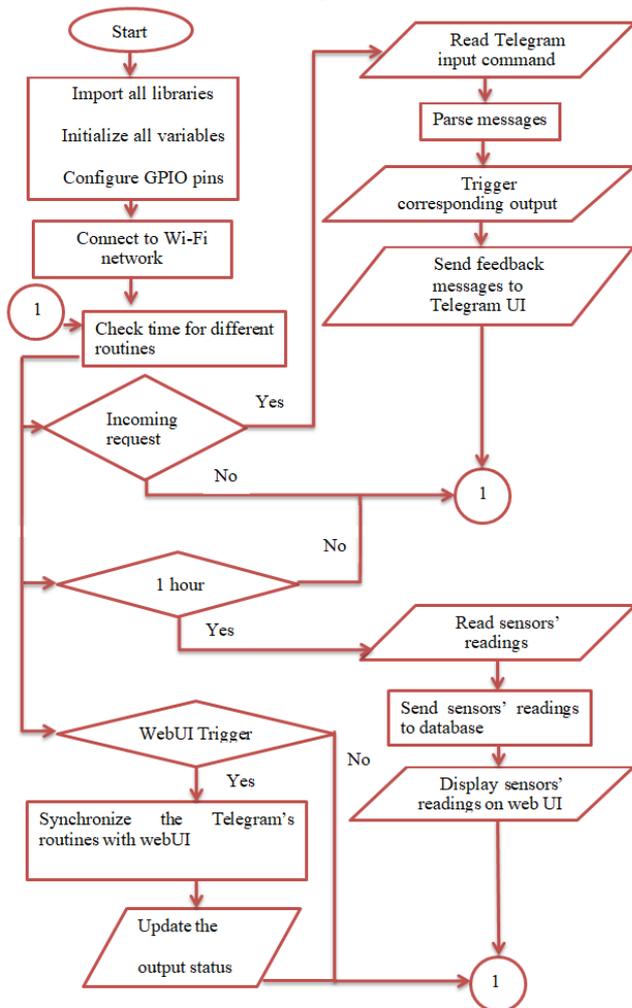


Fig.3. System Flowchart

III. RESULT AND DISCUSSION

A. Results

NodeMCU monitors and triggers the action based on the code parsed from Telegram user. The meaning of each code is depicted as in Table 2 below. The code will be parsed using multiple if statement as in Fig. 4 to carry out task accordingly.

Table- II: Parsing of Telegram's command

Code-to-parsed	Function
/ledon	Turn the LED on.
/ledoff	Turn the LED off.
/status	Check the status of LED.
/temp	Measure the ambient temperature and humidity reading.

```

//message = bot.getUpdates();
// if( message == "/ledon"){
// Turn LED on
}

// if( message == "/ledoff"){
// Turn LED off
}

// if( message == "/status"){
// get status
}

// if( message == "/temp"){
// get temperature and humidity reading
}
    
```

Fig. 4. Telegram sub-routine

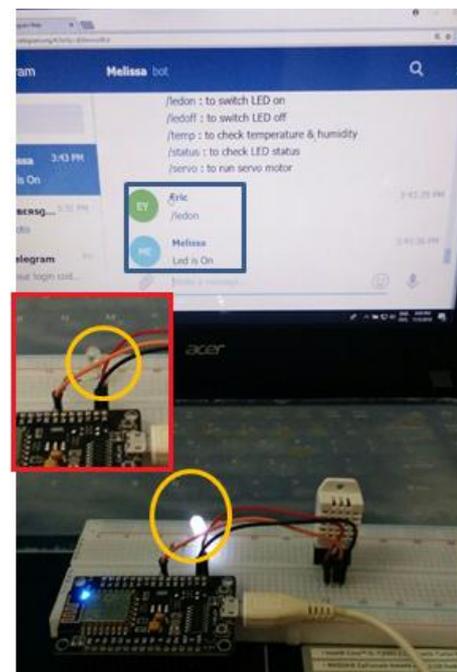


Fig. 5. LED ON and OFF

As shown in Fig. 5, the blue rectangular box indicated the requests(/ledon and /ledoff) sent by Telegram user and the responses from the Telegram Bot (Led is On and Led is off). Meanwhile, yellow circle implied the output status corresponding to the request. It showed that the LED functioned accurately with the code.

Likewise, the status of LED and the measuring of temperature and humidity readings are illustrated as in Fig. 6. The ambient temperature and humidity reading process as in Fig. 6(b) is simplified since the import of DHT.h library in the code will calculate the readings automatically.

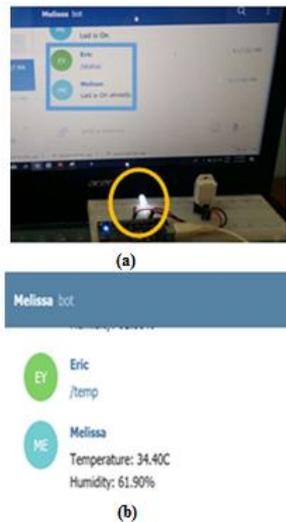


Fig. 6. System status, (a) LED’s status, (b) temperature and humidity readings

Subsequently, in Fig. 7, the second method of manipulating the LED remotely is demonstrated via the implementation of web UI. The red rectangular box indicated the URL of the website to access the LED. By typing or bookmarking the URL (<https://happyeric.000webhostapp.com>), the user can view the webpage as shown in Fig. 7 below.

By toggling the UI button, the user can switch ON or OFF the LED, which is a bit different from Telegram that is only controlled by sending messages. The light indicator created on website will change colors when the UI button is toggled. This meant that the light indicator will turn yellow when toggled and vice versa. The light indicator is synchronized with the LED on the breadboard and therefore will be changed state concurrently once the toggle button on website is pushed.

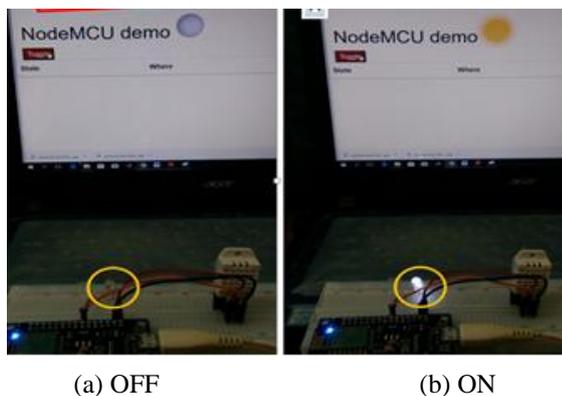


Fig. 7. Web UI demonstration

```

1 <?php
2 if (isset($_GET['current'])){
3 //open and read current state from data.txt
4 fopen() -> fgets() ->fclose();
5 }
6 if(isset($_GET['state'])){
7 //open and write next state to data.txt
8 fopen() -> fwrite()-> fclose()
9 }
10 ?>

```

Fig. 8. Simplified version of toggle.php

Based on Fig. 8, server-side programming language (php) is used to create the contents of web-server. Basic syntax used began with (<?php) and ended with (?>) while the filename must be ended with .php extension. Methods used in our case

are such as fopen(), fgets(), fwrite() and fclose() with the former two methods are for open and read file while the latter ones are for write and close file. Another important thing to mention here is \$_GET in php[18].

```

1 http.begin("http://happyeric.000webhostapp.com/toggle.php?state=0");
2 int httpCode = http.GET(); // Send the request.
3 if (httpCode > 0) { //Check the returning code
4 String payload = http.getString(); // Get the text from the
destination (1 or 0).
5 digitalWrite(ledPin,payload.toInt());
6 }
7 http.end();

```

Fig. 9. HTTP get request from NodeMCU

From Fig. 9, HTTP get request from NodeMCU is to synchronize both web and Telegram user interface. Once the Telegram user turns ON or OFF the LED, NodeMCU will send HTTP get request by calling http.begin() and http.GET() to toggle.php where the new value is written to data.txt. As a result of that, the returned value, http.getString() is converted into payload.toInt() method in NodeMCU to be used in digitalWrite() method to control the LED.

```

1 <script>
2 $(document).ready(function(){
3 current_state();
4 setInterval(current_state,1000);
5 $('#toggle').click(function(){
6 var current = $(this).data('state');
7 $.ajax({
8 url: 'toggle.php?state='+current,
9 method: 'get'
10 });
11 });
12 });
13 </script>

```

Fig. 10. Simplified version jQueryajax script in index.php

In Fig. 10, asynchronous javascript and XML (ajax) is used together with jQuery to not only simplify the work of javascript, but also to update the parts of webpage periodically, without having to refresh the whole page when the button is toggled[19].

B. DHT sensor temperature reading analysis

Table 3. Indoor temperature readings (DHT 22 and digital thermometer), readings taken in 13 hours period.

Table- III: Indoor temperature readings

Sampling time	T_{DHT22}	T_{DHT11}	$T_{digital}$	$e_{22}/\%$	$e_{11}/\%$
12:09 PM	28.2	28.0	28.4	0.704	1.408
1:09 PM	28.1	28.0	28.7	2.091	2.439
2:09 PM	28.6	28.0	28.8	0.694	2.777
3:09 PM	28.6	28.0	29.0	1.379	3.448
4:09 PM	28.8	28.0	29.1	1.031	3.780
5:09 PM	28.9	28.0	29.2	1.027	4.110
6:09 PM	29.10	29.0	29.4	1.020	1.361
7:09 PM	29.2	29.0	29.4	0.680	1.361
8:09 PM	29.2	29.0	29.6	1.351	2.027
9:09 PM	29.3	29.0	29.4	0.340	1.361
10:09 PM	29.3	29.0	29.3	0	1.024
11:09 PM	29.2	29.0	29.2	0	0.685
12:09 AM	29.2	29.0	29.2	0	0.685
1:09 AM	29.1	29.0	29.1	0	0.344

This experiment in this section is conducted in order to verify the accuracy of DHT sensor with respect to conventional digital thermometer. As shown in **Table 3**, parameters such as temperature (in Celsius) for each DHT 22, DHT 11 and digital thermometers are denoted as T_{DHT22} , T_{DHT11} and $T_{digital}$.

These parameters are measured indoor hourly from 12:09 PM to 1:09 AM on 23rd, September 2018 and are recorded in the three left columns of **Table 3**. In addition, the percentage of error of each sensor, denoted as $e_{22}/\%$ and $e_{11}/\%$ are calculated by comparing with the measurement of digital thermometer as standard value and later tabulated in the two right columns of **Table 3**. As a result of the experiment, DHT 22 showcased an expected higher accuracy than DHT 11 as its percentage of error ranged from $\pm 1\%$, while the latter one reached approximately $\pm 4\%$.

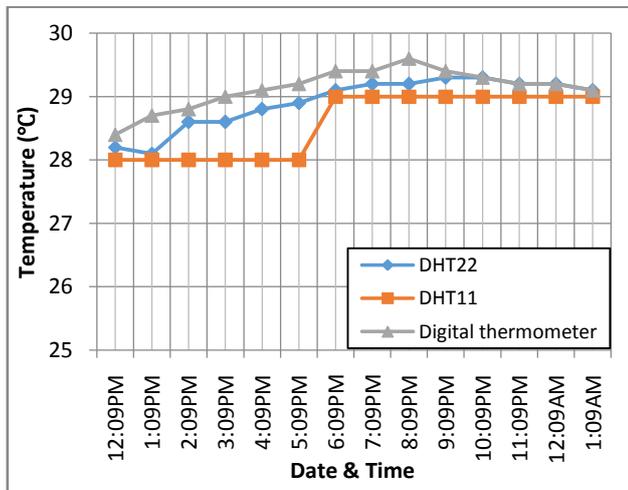


Fig. 11: Graphical illustration of DHT 22 and digital thermometer’s temperature readings.

As for **Fig. 11** and **Fig. 12**, the graphical illustration of the temperature readings from DHT sensors, digital thermometer and percentage of error are illustrated in line graph. Overall, it showed that DHT 22 produced less deviation readings to digital thermometer, thus demonstrated the lowest percentage of error as in **Fig. 12**. Hence, DHT 22 produced more feasible results than DHT 11.

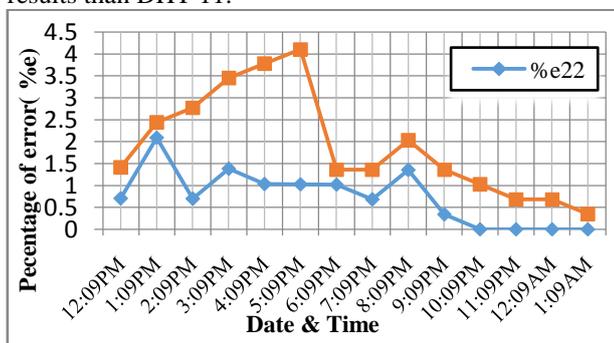


Fig. 12. Graphical illustration of temperature readings in terms of percentage of error exhibited by DHT 22 and DHT 11.

C. System latency analysis

▪ Web UI’s system latency

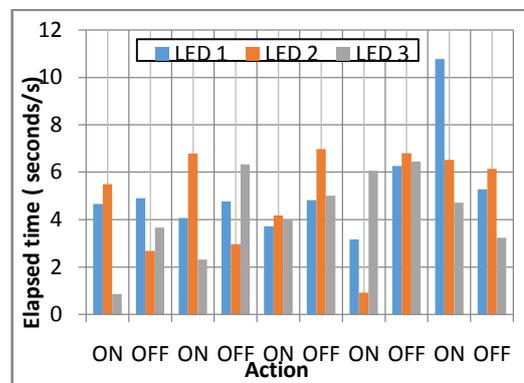
This experiment is carried out to determine the range of elapsed time using web and Telegram UI. **Table 4** below showed the tabulated results where the left first column

depicted the action conducted. The second until fourth columns displayed the elapsed time taken by three actuators (LED 1, LED 2 and LED 3) with web UI.

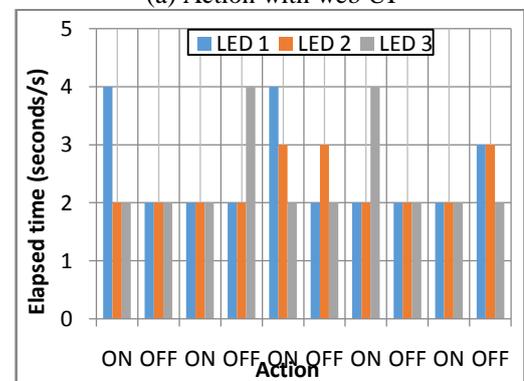
Table- IV: System delay of the web UI

Action (ON /OFF)	Elapsed time (seconds /s), using web UI			Elapsed time (seconds /s), using Telegram UI		
	LED 1	LED 2	LED 3	LED 1	LED 2	LED 3
ON	4.654	5.498	0.857	4.00	2.00	2.00
OFF	4.905	2.683	3.665	2.00	2.00	2.00
ON	4.068	6.783	2.322	2.00	2.00	2.00
OFF	4.769	2.967	6.327	2.00	2.00	4.00
ON	3.720	4.176	3.968	4.00	3.00	2.00
OFF	4.823	6.982	5.006	2.00	3.00	2.00
ON	3.171	0.912	6.066	2.00	2.00	4.00
OFF	6.262	6.802	6.448	2.00	2.00	2.00
ON	10.771	6.522	4.715	2.00	2.00	2.00
OFF	5.279	6.135	3.237	3.00	3.00	2.00

For Telegram UI the elapsed time is shown in fifth column until seventh. Meanwhile, **Fig. 13** illustrated the data in **Table 4** in a more explicit and interactive manner, for both UI. **Fig. 13(a)**, showed the action of LED ON/OFF using web UI and the **Fig. 13(b)** is for Telegram UI. The experimental results indicated that maximum and minimum elapsed time were 10.771 seconds and 0.857seconds respectively. Based on the results, the elapsed time for Telegram UI are consistent where the maximum and minimum values are approximately 4 seconds and 2 seconds.



(a) Action with web UI



(b) Action with Telegram interface

Fig. 13. Graphical illustration of elapsed time of LEDs’

▪ Comparison of system latency between two UI

The performance of this system can be further examined by interpreting the comparison of elapsed time between the two types of user interface. As shown in Fig. 14, the data somehow implied that web UI had the longest maximum and also shortest minimum system delay. On the contrary, Telegram UI suggested the otherwise. Hence, it is ambiguous to say that which user interface has the upper hand in terms of the system delay because the performance of the system can also be affected by the factors such as network bandwidth and throughput.

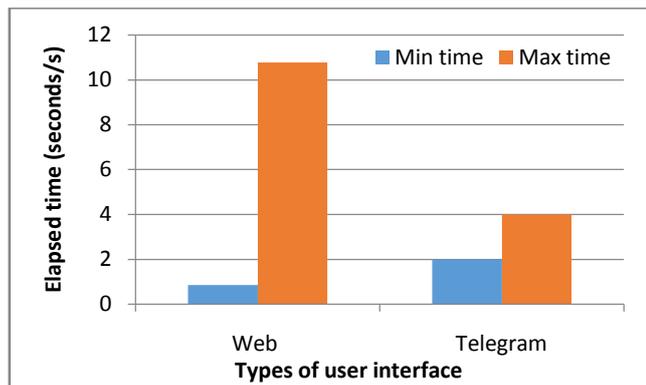


Fig. 14. Comparison of elapsed time between two types of interface

IV. CONCLUSION

This project has been developed in both hardware and software parts where the developed source codes are tested on the hardware to test its feasibility. The achieved outcomes were promising as the data can be sent react by the controller accurately via wireless Internet environment. This project is only tested with direct current (DC) application such as LED. The success achieved from the operating ON / OFF LED is capable of extending to the AC electric circuit. Where in this experiment, there is no test on the AC circuit. However, it is possible to use a DC to AC relay to connect the AC circuit load.

ACKNOWLEDGEMENT

I would like to thank my supervisor and UniversitiTeknikal Malaysia Melaka (UTeM) for the guidance and support in accomplishing this project.

REFERENCES

- [1] M. A. E.-L. Mowad, A. Fathy, and A. Hafez, "Smart Home Automated Control System Using Android Application and Microcontroller," *Int. J. Sci. & Eng. Res.*, vol. 5, no. 5, pp. 935–939, 2014.
- [2] S. Kumar and S. R. Lee, "Android based smart home system with control via Bluetooth and internet connectivity," *Proc. Int. Symp. Consum. Electron. ISCE*, no. 2011, 2014.
- [3] C. L. Hsu, S. Y. Yang, and W. Bin Wu, "Constructing intelligent home-security system design with combining phone-net and bluetooth mechanism," *Proc. 2009 Int. Conf. Mach. Learn. Cybern.*, vol. 6, no. July, pp. 3316–3323, 2009.
- [4] R. Teymourzadeh, S. A. Ahmed, K. W. Chan, and M. V. Hoong, "Smart GSM based home automation system," *Proc. - 2013 IEEE Conf. Syst. Process Control. ICSPC 2013*, no. December, pp. 306–309, 2013.
- [5] R. Acker and M. Massoth, "Secure ubiquitous house and facility control solution," *5th Int. Conf. Internet Web Appl. Serv. ICIW 2010*, pp. 262–267, 2010.
- [6] M. H. A. Wahab, N. Abdullah, A. Johari, and Herdawatie Abdul Kadir, "GSM Based Electrical Control System for Smart Home Application," *J. Conver. Inf. Technol.*, vol. 5, no. 1, pp. 33–39, 2010.
- [7] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, and C. H. Lung, "Smart home: Integrating internet of things with web services and cloud computing," *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, vol. 2, pp. 317–320, 2013.
- [8] I. Alam, S. Khusro, and M. Naeem, "A review of smart TV: Past, present, and future," *ICOSST 2017 - 2017 Int. Conf. Open Source Syst. Technol. Proc.*, vol. 2018–Janua, no. 6, pp. 35–41, 2018.
- [9] D. Yan and Z. Dan, "ZigBee-based Smart Home system design," *ICACTE 2010 - 2010 3rd Int. Conf. Adv. Comput. Theory Eng. Proc.*, vol. 2, pp. 650–653, 2010.
- [10] D.-M. Han and J.-H. Lim, "Design and Implementation of Smart Home Energy Management Systems based on Zigbee," *IEEE Trans. Consum. Electron.* 56(3), pp. 1417–1425, 2011.
- [11] R. A. Ramlee, "Mobile Phone Controlling Home Appliances," *J. Telecommun. Electron. Comput. Eng.*, vol. 5, no. 1, pp. 37–46, 2013.
- [12] R. A. Ramlee *et al.*, "Bluetooth Remote Home Automation System Using Android Application," *Int. J. Engineering Sci.*, vol. 2, no. 1, pp. 149–153, 2013.
- [13] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, "Network-Level Security and Privacy Control for Smart-Home IoT Devices," no. October, 2015.
- [14] R. K. Kodali, V. Jain, S. Bose, and L. Boppana, "IoT Based Smart Security and Home Automation System," pp. 1286–1289, 2016.
- [15] J. C. De Oliveira, D. H. Santos, and M. P. Neto, "Chatting with Arduino platform through Telegram Bot," *Proc. Int. Symp. Consum. Electron. ISCE*, pp. 131–132, 2016.
- [16] ESP8266 Datasheet, "ESP8266EX Datasheet," *Espr. Syst. Datasheet*, pp. 1–31, 2015.
- [17] T. Liu and B. Manager, "Aosong Electronics Co., Ltd Aosong Electronics Co., Ltd," vol. 22, pp. 1–10.
- [18] "PHP 5 Form Handling." [Online]. Available: https://www.w3schools.com/php/php_forms.asp. [Accessed: 12-May-2018].
- [19] "jQuery AJAX Methods." [Online]. Available: https://www.w3schools.com/jquery/jquery_ref_ajax.asp. [Accessed: 05-Nov-2018].