

Implementation of Wireless Temperature and Humidity Monitoring on an Embedded Device

Muhamad Fazril Afif Samsudin, Roslina Mohamad,
Saiful Izwan Suliman
Faculty of Electrical Engineering
Universiti Teknologi MARA
Shah Alam, Selangor
roslina780@salam.uitm.edu.my

Nuzli Mohamad Anas, Hafizal Mohamad
Wireless Innovation
MIMOS Berhad
Bukit Jalil, Kuala Lumpur

Abstract—This paper presents an implementation of wireless temperature and humidity monitoring on a Raspberry Pi. The objective of this project is to design a monitoring temperature and humidity reading kit which implemented on Raspberry Pi. The study focuses on the embedded project for sensing the temperature and humidity in room and library area. The method of analysis is performed on Raspberry Pi implementation with interfacing of a web server, Thingspeak that has an API and posting and reviewing the data to the channel. Python language is encoded in Raspberry Pi Linux OS with DHT11 temperature and humidity sensor for retrieving the data. Hence, the temperature and humidity reading implemented on Raspberry Pi has been designed and it can be concluded that the increasing in temperature indicates to decreasing of humidity levels based on the result collected.

Keywords— temperature and humidity reading; wireless; Raspberry Pi; web server.

I. INTRODUCTION

Temperature and humidity are two main factors that most contribute to the global climate patterns. Their changes shall affect the environment worldwide where life cycles of plants and animals become different. For example, mushroom nursery was built by farmers in order to optimize the temperature and humidity level since both factors ensure the product contains high nutritional value [1]. In civil structure, parameters of temperature and humidity were very important where they can be a reason of deterioration of structure and a significant job for the concrete to have a long term consequences [2].

The use of manual method for temperature monitoring using analog thermometer has made a work becomes slower with more work to do. It maybe create several errors such as gross error by taking the reading of thermometer, the eyes are not perpendicular with the indicator. It is also not easy to access that it cannot be a remote monitoring. Besides that, analog Hygrometers employment for measuring the humidity reading is less accurate as they are individually hand calibrated and slow to react to abrupt changes of humidity.

This paper describes the implementation of a wireless temperature and humidity for autonomous monitoring on a Raspberry Pi. Hence, the objective of this paper is two-fold, first is to emphasize the development of wireless temperature

as well as humidity for autonomous monitoring on an embedded Raspberry Pi. And secondly explained the development of an alert system for monitoring the readings when it reached a certain condition. Finally perform the analysis on the result obtained from temperature and humidity readings in the experimental areas.

There are several things that are focused in order to achieve the objectives, which is to implement a wireless temperature and humidity monitoring on the Raspberry Pi. First of all, the project covering for Raspberry Pi, running with Linux operating system, OS coded with Python language for retrieving temperature and humidity readings where the values are sensed through DHT11 sensor and sent to the internet. Besides that, it is only covering for experimental area, room and library area for the readings to be taken.

The relevancy of this framework is to ensure that the monitoring system for temperature and humidity wirelessly is developed in IoT world. Hence, it can be an easy access to remote monitoring an area for temperature and humidity when the reaction is being created. It is also shall reduce the uses of manual monitoring such as analog thermometer that contribute to a lot of errors for the reading to be taken. It is good to be notified when a certain temperature condition is met for the next step to be taken. The rest of this paper is organized as follows: Section II explains about literature review. Section III explains about methodology while Section IV discusses about result and its discussion for analysis part. For the conclusion part it is discussed in the Section V.

II. LITERATURE REVIEW

With the advancement of wireless technology, life has become simpler and easier in almost all aspects. The preferred of world automation systems shows the rapid increase in the number of users of internet usage over the past decade. The development of Internet of Things, IoT ensures the monitoring method becomes more advance in the uses that can be access from remote area.

One of the previous research papers used the monitoring system for temperature and humidity to identify the chances of grape diseases. This method shall detect the diseases in its early stages via SMS provided by Hidden Markov Model to the

farmer [3]. Another paper used monitoring system of temperature and humidity in civil structure through the concept of Structural Health Monitoring (SHM). This method was to ensure the structure is monitored on the real time and determine the structure damages as well as evaluate the performance under many loads to the structure [2].

As they are today, it is predicted that for the future where there are no PCs in the industry when there will be only smart embedded devices that are connected via a wide wireless network to each other [4]. This has led the development of Internet of Things (IoT). Essentially, IoT is placed for doing objects representation in the digital realm through giving them a specific ID and connecting them into a network [5].

A. Raspberry Pi GPIO Pin

The Raspberry Pi is an economic budget, super pocket sized card computer that connects into a monitor, sometimes laptop and uses a particular mouse and keyboard. Even with small size, it could experience with several big things like internet browsing, word processing and video games playing.

GPIO represents for General Purpose Input/Output and attached to the Raspberry Pi, they are in physical interface among software part and outside world. GPIO pins are bidirectional way that they can send the signals out and receive the signals in from external circuits at the same times. Different versions of the Raspberry Pi interface with different amount of GPIO pins. Fig.1 shows the Raspberry Pi 3 Model B+ with total of 40 pins attached to it.

Physical Pin Number		
3.3V	1	2
GPIO2	3	4
GPIO3	5	6 GND
GPIO4	7	8 GPIO14
GND	9	10 GPIO15
GPIO17	11	12 GPIO18
GPIO27	13	14 GND
GPIO22	15	16 GPIO23
3.3V	17	18 GPIO24
GPIO10	19	20 GND
GPIO9	21	22 GPIO25
GPIO11	23	24 GPIO8
GND	25	26 GPIO7
DNC	27	28 DNC
GPIO5	29	30 GND
GPIO6	31	32 GPIO12
GPIO13	33	34 GND
GPIO19	35	36 GPIO16
GPIO26	37	38 GPIO20
GND	39	40 GPIO21

Key
Power+
GND
GPIO

Fig. 1. Raspberry Pi3 Model B+ GPIO Pin

B. Wireless Sensor Networks

As stated by to Jason Lester Hill, Wireless Sensor Network (WSN) is Sensing + CPU + Radio = Thousands of potential applications [3]. It is embedded with independent devices such as sensors, do task for monitoring physical conditions like temperature and humidity, sound, pressure, etc. The choosing of a suitable temperature sensor for connected hardware becomes the very first priority although it may seem like a straightforward decision [6]. The experience of monitoring temperature by sensing of hot and cold, analog thermometers and many digital temperature displays in many industries such

as agriculture and manufacturing with different settings and needs make the decision for choosing the right temperature becomes a critical part of taking into account of cost, scale, location and intended use. Fig. 2 shows the working component which is DHT11 Temperature and Humidity Sensor.

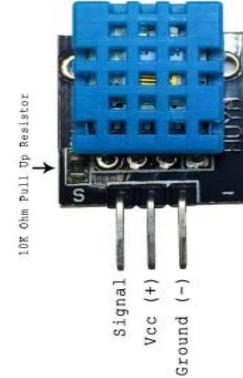


Fig. 2. DHT11 Temperature and Humidity Sensor

C. Android Apps

The introduction of apps where the development of software programs to execute on mobile devices for accomplishing a specific goal [7] is increasing as well as the technology. For free downloading or at nominal value, there are more than 100000 apps available for five following stores: Google Play, Apple iTunes, Amazon App, Windows and Blackberry [8]. For the goal of connecting an object to the IoT, the interphase is needed that give services for simple communication capabilities to the objects within the IoT environment, whereas it is a web server or database platform. The data is stored which provides with a list of features.

D. Influence of Temperature and Humidity

Temperature and air humidity conditions are very important in today activities since they can contribute to a positive product in industrial, agriculture, biological, etc. In plant response, they enroll the most significant and direct influence on transpiration and other gases exchange by increasing the transpiration rate for a normal growth. Generally, the humidity level was influenced by the temperature itself where the increase in temperature leads the humidity to be increase. However, it is also depends on other factors such as geographical location where high humidity level in the coastal areas, an area with a lot of surface water.

III. METHODOLOGY

The methodology of this works is manifest through the flowchart for overall project from the beginning for hardware design, continues with software design and ending with analysis of the result.

A. Flowchart of Project

Fig. 3 shows the flowchart for overall project. The flow of the study is begun with the Raspberry Pi setup where as a

Raspbian, the NOOBS has to be installed in SD card. After that, the python had to be configured, means that the Adafruit GPIO Python library and Adafruit DHT11 library have to load in Raspberry Pi. Then, the Python script is prepared and is saved with .py at the back since it will be program in Python language. Next, the circuit is built where DHT 11 temperature and humidity sensor is connected to the Raspberry Pi through GPIO pins. Later, the Thingspeak is setup for three fields, temperature in degree Celsius, Fahrenheit and humidity in percentage. The API key in the channel of Thingspeak was synchronized with the Python script. Then, the program is executed through LX terminal of Linux OS. If the execution was not successful, the script had to be reconfigured. When the program run is successful, the temperature met a condition whether it exceed 35°C or not. If condition is matched, a notification was send to Twitter account. Lastly, the data were analyzed according to the area of research.

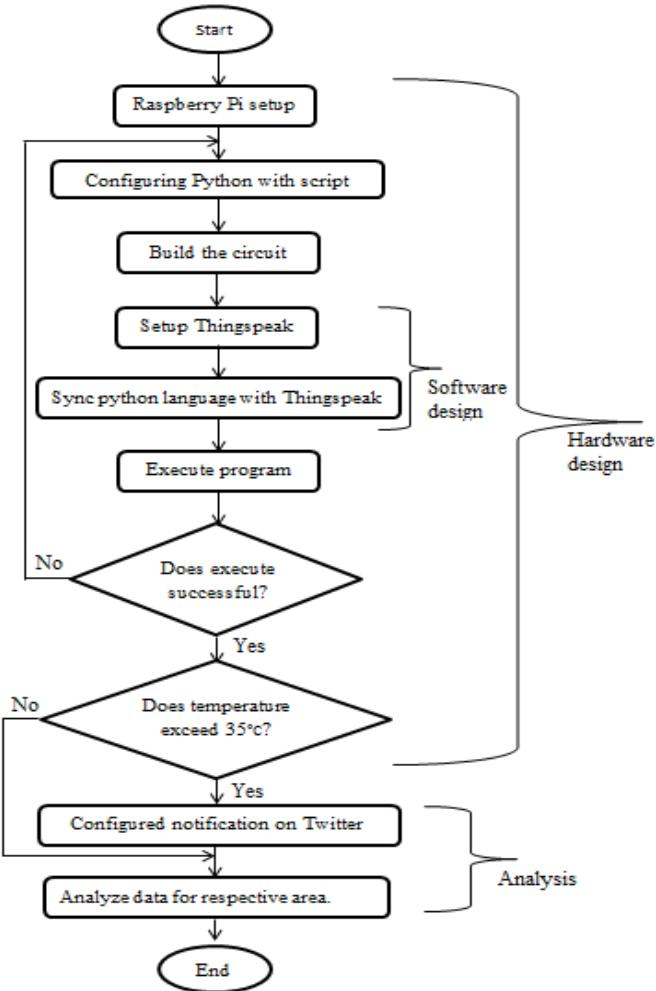


Fig. 3. Flowchart of Project

B. Hardware Design

The hardware design in the Fig. 4 and Fig. 5 as shown below is schematic diagram of Raspberry Pi 3 connected to DHT11 sensor and connection diagram respectively. The raspberry Pi was interfaced with a temperature and humidity

sensor and reads data from it through its GPIO pin whereas GPIO 2 (pin 3) was used. The sensor was connected to the Raspberry Pi as such:

- Connect the Ground Pin of the sensor to physical pin 4 of the GPIO board.
- Connect the Data Pin to physical pin 3 (GPIO 2) of the GPIO board.
- Connect Vcc to physical pin 1 (3.3V) of the GPIO board.

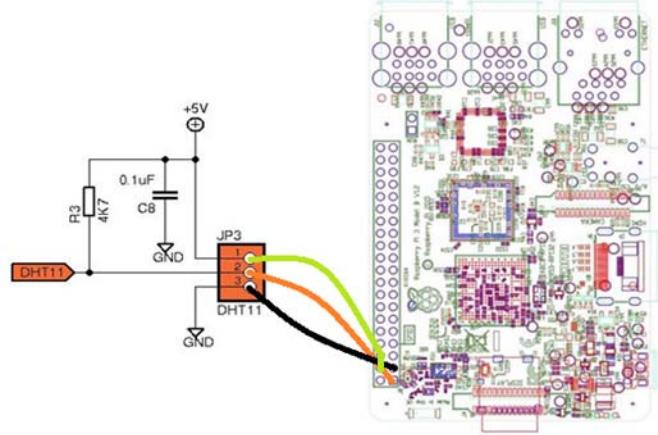


Fig. 4. Schematic diagram of Raspberry Pi 3 connected to DHT11 sensor.

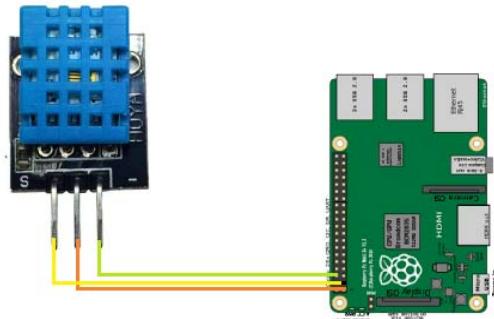


Fig. 5. The Connection Diagram

C. Software Design

The setup of server side was needed in order to receive data. In this study, the use of <https://thingspeak.com> has been developed. Thingspeak provides server-side infrastructure to collect data and process it. Fig. 6 shows the flow for creating the server side where the first step is a new account was created on <https://thingspeak.com> and MathWorks account was signed up. Then, a new channel was created and fields for device to collect and send data were defined. The Write API key is an important key for data to be served to the right field and the key was saved and noted. In order for additional channel, the step had to go back for creating a new channel and follow the rest. Lastly, the Thing speak setup was done after the channels to store all data were created. After that, an alert was created based on an event that was provides in apps from Thingspeak. The alert was linked with the Twitter account for sending the notification.

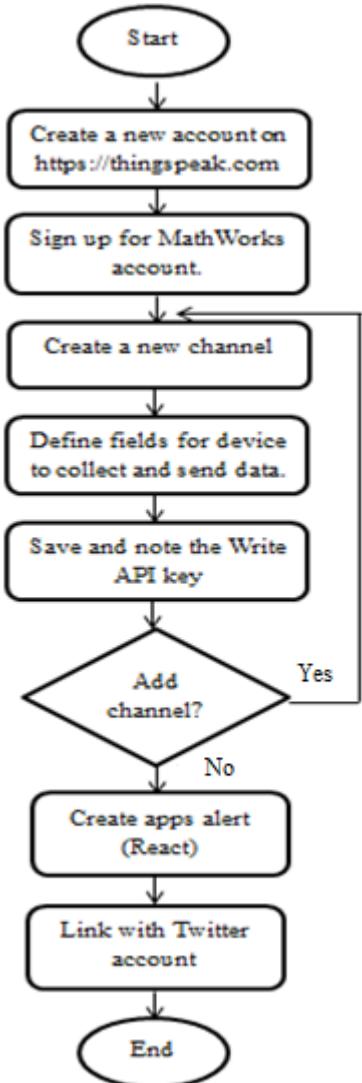


Fig. 6. Flow of creating server side

IV. RESULTS AND DISCUSSIONS

The results were obtained by collecting the data for two areas which is temperature in room and at the lobby of Perpustakaan Tun Abdul Razak 3 (PTAR3) in UiTM Shah Alam. They also were collected for day and night. In daytime, the data were collected between 2p.m. until 4p.m. while at night, they were collected between 12a.m. until 2a.m.. All results obtained were tabulated in Thingspeak.com as a web server and all the data is sent in here.

Table I shows the data collection for room and lobby PTAR3 temperatures in days and at night. The data were collected for 20 entries with 15 seconds delay for each entry. The readings were taken for temperature in degree Celsius and Fahrenheit as well as humidity.

From the Table I, the comparison had been made for each condition. In Fig. 7, the temperature in the room in daytime is higher than at night. The average temperature for room in day is 31°C while at night, its average temperature is 30°C by 1°C difference.

However, from data collected in lobby PTAR3, the temperature at night is higher than in the daytime as shown in Fig. 8. The average temperature stated that in day is 28°C while at night, 29°C with 1°C difference.

For the comparison in humidity, in room condition shows in Fig. 9, that the average humidity at night is higher than in daytime. The average room humidity in day was 41% while at night was 53%. In Fig. 10 for condition in lobby PTAR3, the temperature in daytime is higher than at night. The average lobby PTAR3 humidity in day was 47% and at night was 42%. In humidity, the higher the percentage leads to rise of air moisture.

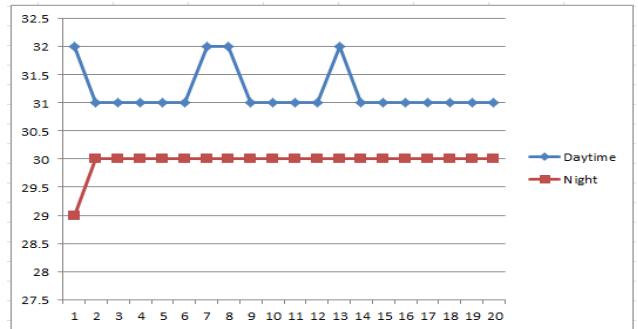


Fig. 7. Comparison of temperature (°C) vs time for daytime and night in room condition

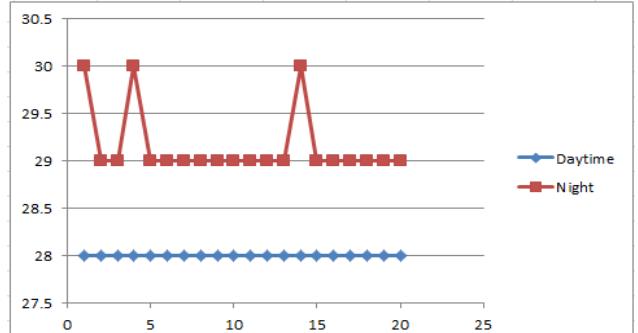


Fig. 8. Comparison of temperature (°C) vs time for daytime and night in lobby PTAR3

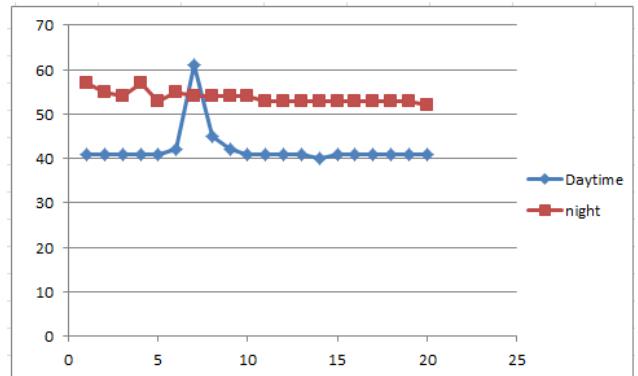


Fig. 9. Comparison of humidity vs time for daytime and night in room condition

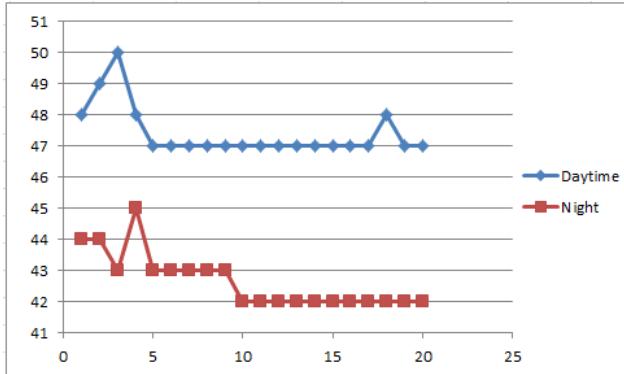


Fig. 10. Comparison of humidity vs time for daytime and night in lobby PTAR3

Fig. 11 shows the platform in Twitter notification. It stated that the temperature is greater than 35°C. The notification was configured as the temperature monitored was higher than 35°C. From this notification, the next step could be taken as the alert action has been twitted using configured account.



Fig. 11. Twitter Notification platform

TABLE I. DATA COLLECTION FOR TEMPERATURE IN ROOM AND LOBBY PTAR3, DAYTIME AND NIGHT

Entry	Room						PTAR3 Lobby					
	Temperature (°C)		Temperature (F)		Humidity (%)		Temperature (°C)		Temperature (F)		Humidity (%)	
	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night
1	32	29	89.6	84.2	41	57	28	30	82.4	86.0	48	44
2	31	30	87.8	86	41	55	28	29	82.4	84.2	49	44
3	31	30	87.8	86	41	54	28	29	82.4	84.2	50	43
4	31	30	87.8	86	41	57	28	30	82.4	86.0	48	45
5	31	30	87.8	86	41	53	28	29	82.4	84.2	47	43
6	31	30	87.8	86	42	55	28	29	82.4	84.2	47	43
7	32	30	89.6	86	61	54	28	29	82.4	84.2	47	43
8	32	30	89.6	86	45	54	28	29	82.4	84.2	47	43
9	31	30	87.8	86	42	54	28	29	82.4	84.2	47	43
10	31	30	87.8	86	41	54	28	29	82.4	84.2	47	42
11	31	30	87.8	86	41	53	28	29	82.4	84.2	47	42
12	31	30	87.8	86	41	53	28	29	82.4	84.2	47	42
13	32	30	89.6	86	41	53	28	29	82.4	84.2	47	42
14	31	30	87.8	86	40	53	28	30	82.4	86.0	47	42
15	31	30	87.8	86	41	53	28	29	82.4	84.2	47	42
16	31	30	87.8	86	41	53	28	29	82.4	84.2	47	42
17	31	30	87.8	86	41	53	28	29	82.4	84.2	47	42
18	31	30	87.8	86	41	53	28	29	82.4	84.2	48	42
19	31	30	87.8	86	41	53	28	29	82.4	84.2	47	42
20	31	30	87.8	86	41	52	28	29	82.4	84.2	47	42

V. CONCLUSION

This paper has presented a way for monitoring temperature and humidity using a Raspberry Pi wirelessly. It can be concluded that the monitoring method for temperature and humidity had been developed. From the result obtained, it can be said that the factor of day and night only does not influence the temperature and humidity level. However, the analysis also indicates that the level of humidity is depend on the temperature level where, the higher the temperature, the lower the humidity and vice versa.

For the future work, it can be recommended that this project should include on the controlling of the temperature and humidity of an area. This opinion will make the area maintain a normal or necessary temperature and humidity.

ACKNOWLEDGMENT

The authors would like to thank Universiti Teknologi MARA (UiTM) and Minister of Higher Education (MOHE) for supporting this research through FRGS grants number FRGS/1/2017/ICT05/UITM/02/2 respectively.

REFERENCES

- [1] T. Kaewwiset and P. Yodkhad , “Automatic temperature and humidity control system by using Fuzzy Logic algorithm for mushroom nursery,” 2017 International Conference on Digital Arts, Media and Technology (ICDAMT), Chiang Mai, 2017, pp. 396-399.
- [2] M. Bouyah and T. Ezzedine, “Design of Smart Bridge based on WSN for efficient measuring of Temperature , Strain and Humidity,” 2016 4th International Conference on Control Engineering & Information Technology (CEIT), Hammamet, Tunisia, 2016, pp. 1-5.
- [3] S. S. Patil and S. A. Thorat, “Early Detection of Grapes Diseases Using Machine Learning and IoT,” 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP), Mysore, 2016, pp. 1-5.
- [4] N. Ahuja, U.S. Banerjee, V.A. Darbhe, T.N. Mapara, A.D. Matkar, R.K. Nirmal, S. Balagopalan, “Computer controlled robotic arm,” 16th IEEE Symposium on Computer-Based Medical Systems, New York, 2003, pp. 361-366.
- [5] Ashton, K. 2009. That “internet of things” thing. RFID Journal. 22, (2009), 97–114.
- [6] M. Salas, M. Koerdt, M. Hübner, M. Kahali and W. Lang, "Material integrated sensors for an optimal baseline selection on a wireless SHM network," 2016 IEEE SENSORS, Orlando, FL, 2016, pp. 1-3.
- [7] Wallace S, Clark M, White J. ‘It’s on my iPhone’: attitudes to the use of mobile computing devices in medical education, a mixed-methods study. BMJ Open 2012; 2: e001099.
- [8] European Commision. Healthcare in your pocket: unlocking the potential of mHealth. 2014. Available at http://europa.eu/rapid/press-release_IP-14-394_en.htm (accessed on 23 December 2015)..