Developing a Weather Monitoring System on Android Platform

Putri Elfa Masudia
Dept. of Electrical Engineering
State Polytechnic of Malang
Malang, Indonesia

Moechammad Sarosa
Dept. of Electrical Engineering
State Polytechnic of Malang
Malang, Indonesia

Azerino Yogananta Gatot
Dept. of Electrical Engineering
State Polytechnic of Malang
Malang, Indonesia

Putri Afriana
Dept. of Electrical Engineering
State Polytechnic of Malang
Malang, Indonesia

Amalia Eka Rakhmania
Dept. of Electrical Engineering
State Polytechnic of Malang
Malang, Indonesia

ABSTRACT

Weather and climate are important factors in human life since it is one of the main deciding factors for their activities to run smoothly. Furthermore, recent extreme weather changes affected a person's physical and mental health, thus interfering community activities. Temperature, humidity, rainfall rate, wind speed, and direction are the essential elements of weather that could be measured using specific tools or sensors. Rapid development of android technology helps people to access any information through the grasp, including information about weather monitoring in a particular place that could be obtained through android application. This paper designed a real-time weather monitoring tool that could be accessed publicly through Android devices. It uses three sensors, which are wind vane sensor to calculate the wind direction and its velocity; raindrop sensor to detect if it is rain; and DHT11 sensor to monitor the temperature and humidity. All three sensors are integrated with Arduino Uno system that will translate sensors' data. These data then transmitted to server using ESP 8266.

On the other hand, android based application is made so that user could access data in the server easily. Besides testing the functionality of the application, testing step is used to calibrate each sensor with conventional measurement tool to get the precise data. Result shows that there is 0.223m/s difference of wind vane sensor and anemometer measurement, while there is insignificant measurement difference as 1ºC in testing DHT11 sensor.

Keywords

Monitoring, Wind vane, Anemometer, DHT11, Raindrop, Arduino Uno, Android

1. INTRODUCTION

Climate change not only affects the environment but also influence a person's physical and mental condition. That is, during good weather, many benefits can be obtained in the agriculture, tourism, fisheries and many others. Recent extreme weather changes shown a direct impact on humans, for example, humans are susceptible to flu, cough, fever and many others. But the lack of public knowledge about the influence of weather conditions has disrupted community activities. Damage caused by wind speeds that exceed the limits is very detrimental; it can knock down trees and electric poles that are found around houses or public places.

Weather and climate are important factors for human life. Temperature, humidity, rainfall rate, wind speed, and direction are the essential elements of weather. Wind speed determined by the difference in air pressure between the origin and destination of the wind. A calm wind with its speed around the safe threshold can be useful.

Therefore, to reduce the negativity of the weather condition, a tool to continually monitor the weather is required. The monitoring system should be real-time and publicly accessible. This paper proposed an android based application for weather monitoring system. The application built with android programming language so that the public could access it freely. The monitoring system used Arduino Uno to translate data gathered from three sensors, wind vane, anemometer, and DHT11. Decoded data is then sent to MQTT protocol through ESP 8266 for public access in android application.

This paper is written as follows. Section I, Introduction explains the background of this research. The basic theory is described in Section II. Section III shows the system design, while implementation and testing clarify in Section IV. Conclusion is explained in Section V.

2. BASIC THEORY

Internet of things offers many facilities that convenience human work. IoT represents hybrid architecture, which means that it could have different subsystem architecture [2]. However, it needs a mechanism to associate data from sensors stored in big data storage [6].

Authors in [1], [3], [8] proposed an IoT based monitoring system. Authors in [1] suggested a web-based room monitoring system using temperature, humidity, and light intensity sensors. [3] proposed a web-based wind speed and temperature monitoring system. This paper focused on using proxy Reserve in Transmission Control Protocol. IoT based data logger system for weather monitoring using wireless sensor network was proposed in [8], in which it includes controlling when temperature has risen, the fan will be switched on, when light intensity is low, LED will be switched on automatically.

On the other hand, this paper proposed real-time weather monitoring application that could perform automatically, continuously, and accessible for all users. It uses Arduino Uno to convert sensor data into digital data then transmit the data to MQTT, a protocol that implements a publish-subscribe mechanism using TCP/IP as the communication basis [7] through ESP 8266. There are three sensors used in this paper, they are:

Wind Direction Sensor

Wind direction could be measured using a wind vane. It uses Hall effect A3144 sensor consists of IC 3144 which is Hall effect non-latching IC sensor. When the IC is exposed to a magnetic field, the sensor will change the voltage output. It also generates Hall voltage to detect the magnetic field and comparator to amplifies Hall voltage. This sensor could also detect wind speed. Beaufort is a standard scale to measure wind speed. It usually uses to compare wind speed measurement with anemometer [5].

DHT 11

DHT11, a temperature and humidity sensor that has analog output signal calibrated to the more complex sensor. Although it is a simple sensor, it has good quality, fast response, and high performance [4].

Raindrop Sensor

Raindrop sensor is used to detect rain in the environment. Its analog output converted to logic 1 when there is no rain, and 0 when there is rain [9]. It could be used as the switch when there is water dropped through the raining board on the sensor. It could also be used to measure rain intensity.

ESP 8266

ESP8266, an electronic module that can connect with WiFi. It has firmware and setATCommand that could be programmed using Arduino ODE. Its General port input-output (GPIO) could access sensor and Arduino so that it enhances Arduino capability to connect to WiFi.

3. DESIGN

3.1 System Design

This system uses three sensors. First, wind vane sensor is used to measure wind speed and direction. Second, the DHT11 sensor for measuring temperature and humidity. Last is the raindrop sensor to detect rainfall. All three sensors are connected to Arduino, which its primary role is converting analog data obtained by the sensor into digital data. Data is then sent to MQTT protocol through ESP 8266. A function called " readArd" is created to connect the application with thingspeak. ThingSpeak is an open source Internet of Things applications and API to store and acquire data from MQTT via internet [10].

Then, the address is set with the global arduinoStatus variable to call Web1. Both steps are done to make sensor data on thingspeak server could be accessed via Android. System block diagram is shown in Fig. 1.

Macintosh, use the font named Times. Right margins should be justified, not ragged.

3.2 Flowchart Design

The first step to do is defining ports and each sensor's variable. This variable will be used as data referral for the sensor. Fig. 2 shows the system flowchart.

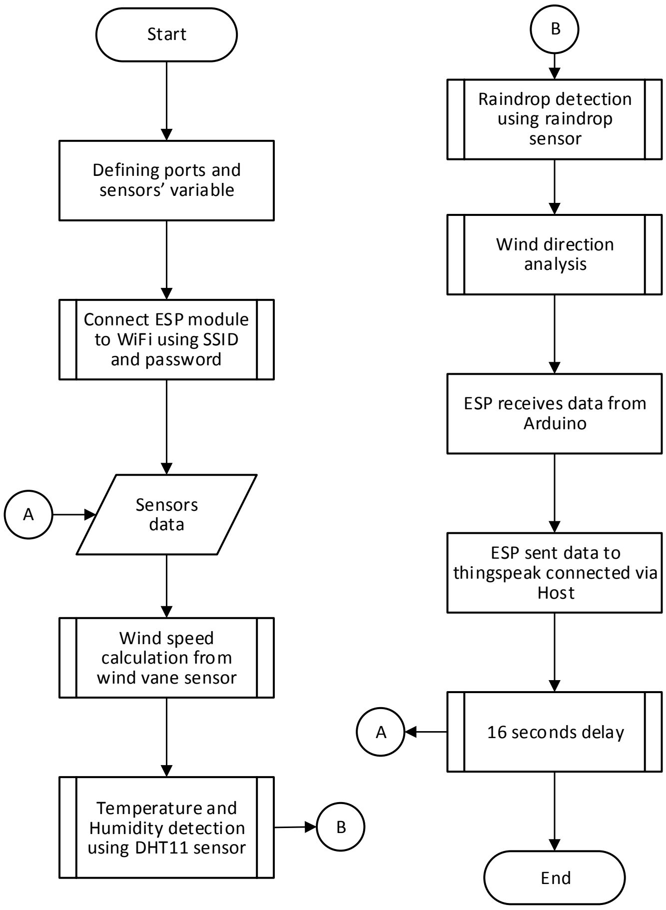

Fig. 2 Weather Monitoring System Flowchart

The first step to do is defining ports and each sensor's variable. This variable will be used as data referral for the sensor. The connection between ESP and WiFi network is then established using SSID and password. Fig. 2 shows the system flowchart.

The three sensors detect different variables. The DHT11 sensor will measure environment temperature and humidity, while wind wave will detect wind speed and its direction. On the other hand, raindrop sensor will measure rainfall occurrence.

Analog data gathered from sensors are converted into digital in Arduino Uno. These data are sent to a cloud server called thingspeak through ESP8266 that connected to WiFi. After this step, data will be shown in the thingspeak page. Our proposed android application will retrieve data from thingspeak and display it in the application

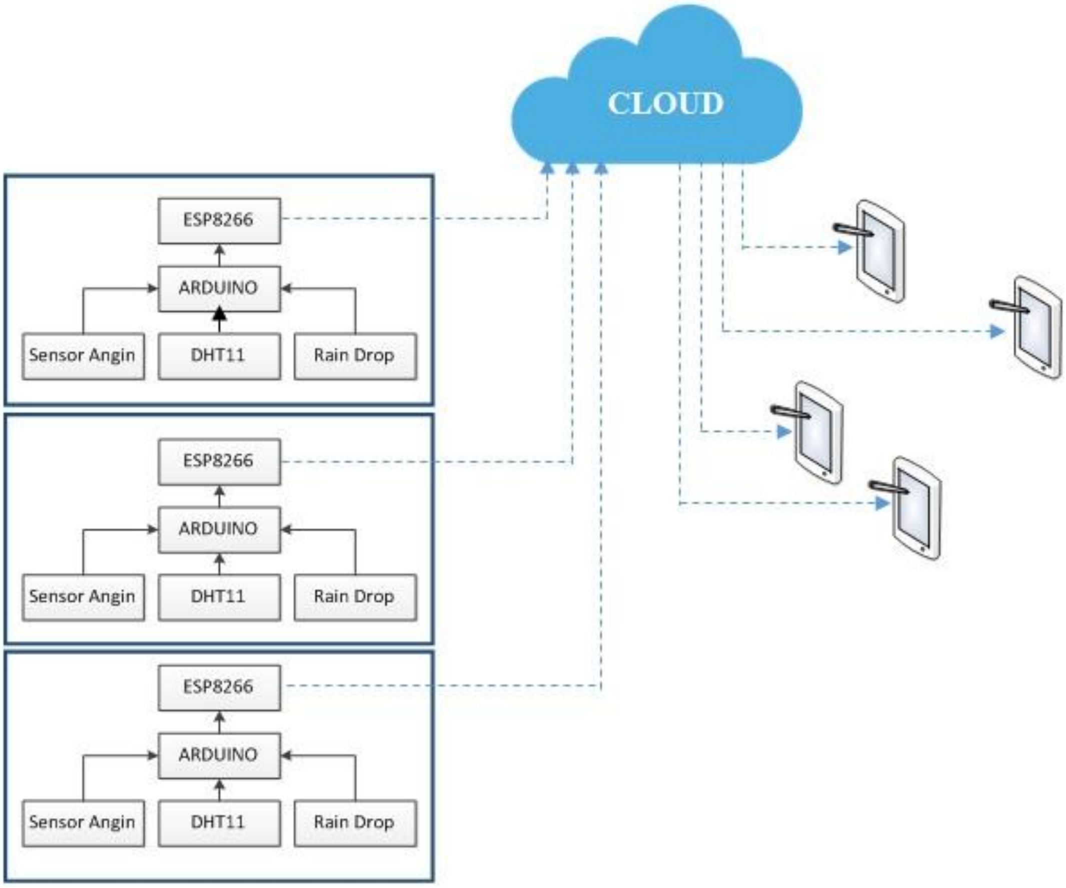

Fig.1 System Block Diagram

3.3 Hardware Design

Each pin on all three sensors is connected to each pin on Arduino Uno. Data pin will be connected to the input pin on Arduino Uno, whereas the VCC pin is connected to 5V DC supply, and so as the GND pin. Hardware design is shown in Fig. 3

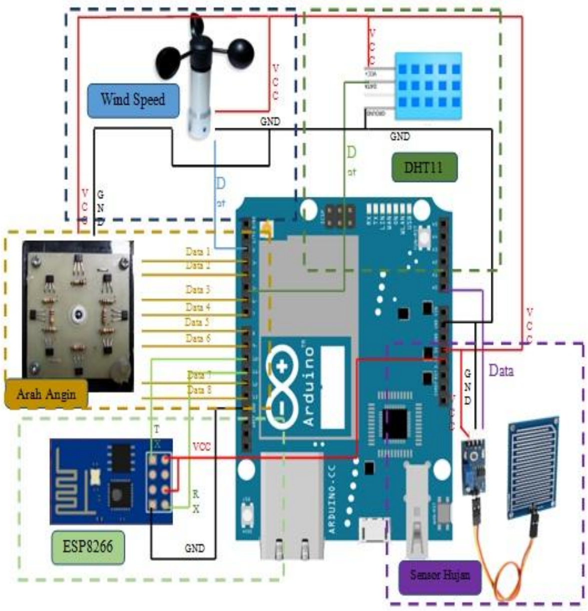

Fig. 3 Hardware Design

4. IMPLEMENTATION, TESTING, AND ANALYSIS

4.1 Implementation

There are three menus appear in application's main page. Data menu which displays the temperature, humidity, wind speed, and direction data, also rain occurrence in the desired location. User menu shows the user's information. Last is the Home menu that will bring the user back to the main page.

During the implementation stage, weather monitoring systems were placed in three different locations. Temperature, humidity, wind speed and direction data could be monitored in real time on android smartphone. These data will be updated automatically every 30 seconds. Weather data displayed in smartphone is shown in Fig. 4

Fig. 4 Weather Data Display

4.2 Testing

4.2.1 Anemometer Testing

Anemometer testing was done by calibrating anemometer used in the research with standard anemometer LTUtron AM4200. Calibration result shown in Table 1

Table1. Anemometer Calibration Result

Anemometer LTUtron AM4200	Used Anemometer	Differences
2,8 m/s	2,87 m/s	0,07 m/s
2,8 m/s	2,71 m/s	0,9 m/s
2,2 m/s	2,31 m/s	0,11 m/s
2,6 m/s	2,66 m/s	0,06 m/s
2,7 m/s	2,56 m/s	0,04 m/s
2,4 m/s	2,56 m/s	0,16 m/s
Average		0,223 m/s

The 0,223 m/s difference was caused by rounding factors of data. LTUtron anemometer shown one digit to the right of decimal point, while the used anemometer in this paper had two digits. It also means that the accuracy of our anemometer is better than LTUtron AM4200.

4.2.2 Rain detection Sensor Testing

Rain detection sensor calibration was done by dropping water to the sensor then measuring sensor's resistance value. The water dropping rate was differentiated into no rain, means there is no water fell to the sensor, moderate rain, water fell with the slow interval, and heavy rain, in which the water dropped in the faster range. During these three occurrences, resulted resistance were different. Thus, the sensor will send the different value of voltage based on the resulted strength. Results show that the output voltage during testing was the same as the specified amount during calibration. The output voltage and resulted resistance is shown in Table 2.

Table 2. Rain Detection Sensor Testing Result

Rain level	R (KΩ)	V (volt)
No Rain	90	4.6
Moderate Rain	22	2.1
Heavy Rain	16	1.2

4.2.3 Wind vane Sensor Testing

Wind vane sensor testing was done by comparing sensor data with compass. Sensor and compass were put on a line during testing. The same results shown in both wind vane and compass data.

Table 3. Comparison between compass and wind wave sensor data

Compass	Wind vane sensor
South	South
West	West
East	East
Southeast	Southeast
Northeast	Northeast

4.2.4 Wind vane Sensor Testing

Comparing sensor data with real thermometer is the way to test DHT11 sensor. Table 4 shows DHT11 sensor result. The sensor works well based on the difference result which only had 1º Celsius deviation.

Table 4. DHT11 Sensor Test Result

Thermometer	DHT11	Difference
27 ^{0}C	26^{0}C	1^{0}C
28 ^{0}C	28^{0}C	0^{0}C
31^{0}C	30^{0}C	1^{0}C
20^{0}C	22 ^{0}C	2^{0}C
Average		**1^{0}C**

4.2.5 Delay to Distance Testing

Measuring the reading of data from certain distance using stopwatch was done to test the relation of delay and sensor distance. Testing results shown in Table 5. It appears that there is no exact pattern between distance and delay due to the different quality of smartphone's internet connection.

Table 5. Delay to distance testing results

Time	Location	Distance from sensor	Delay
07:16	Pakis	15 Cm	0:0:01:24
08:12	Sulfate	9,7 Km	0:0:01:54
10:36	Polinema	13,2 Km	0:0:02:16
13:11	Malang city	12 Km	0:0:02:44

4.2.6 System Testing

The system testing performed at Sulfat Blimbing, Malang. The testing procedure as follows:

1. Tools preparation (anemometer, wind vane, DHT11, raindrop, power bank, ESP8266, and Arduino)
2. Connecting sensor with other instruments based on the Figure 3
3. Deploying the hardware on the testing area
4. Starting the device followed by running the Android application
5. Monitoring the data gathered from the equipment through Android application

Fig. 5 Weather Monitoring System on Smartphone

After the hardware installed in the selected area, it started to gather the data. It contains temperature, humidity, raindrop and wind speed. Table 6 shows the testing result.

Table 6. Testing Result

Hour	Temp	humidity	Speed Wind	Direction	Rain
6:01	22	83	0	Northeast	No
7:12	24	77	0	Northeast	No
8:13	26	71	0.10053	East	No
9:09	27	65	0	East	No
10:10	29	51	0	East	No
11:10	30	52	0	East	No
12:10	30	50	0	East	No
13:12	32	47	0	East	No
14:10	32	45	0.61627	Southeast	No
15:11	34	42	0.10053	West	No
16:10	30	52	0	West	No
17:10	28	58	0	North	No
18:08	27	65	0	South	No
19:10	25	72	0	Northwest	No
20:08	24	75	0	Southeast	No
21:09	24	77	0	South	No

5. CONCLUSION

This paper proposed an android based weather monitoring system. It used three sensors to monitor temperature, rain occurrence, also wind direction and speed. Testing shows that although the orders were placed in three different locations, each of them transmitted real-time data updated every 30 seconds. The data were compared to the standard measurement tools and had a slight difference that could be ignored since caused by rounding factors. Data sent to android application from *thingspeak* placed in the systems to be read by users. There is no change in those data, although for some variable like wind direction and rain occurrence were translated into daily language so as users could understand it easily.

6. REFERENCES

[1] Aleksandrovich, Vladislavs Filicevs, Eduards. Kampars, Janis. 2016. Internet of Things Structure, Features, and management. Riga Technical University, Latvia

[2] Pham, Linh Manh. 2015. A Big Data Analytics Framework for IoT Application in the Cloud. University of Grenoble Alpes, Prancis

[3] Prabowo, B., Muttaqin, A, Zainuri, A. 2017.) "Monitoring ruangan berbasis IoT." Jurnal Mahasiswa TEUB Vol.5. No.1

[4] D. Albert Sanada Yapri, 2017 "Rancang Bangun Aplikasi Mobile IniAtauItu sebagai Media Tanya Jawab Berbasis Komunitas," vol. 03, no. 02, pp. 47-56,

[5] Ram. Kondamudi, A.N.P.S Gupta, 2016. "IoT based data logger system for weather monitoring using wireless sensor network" International Journal of Engineering Trends and Technology (IJETT) –ISSN: 2231-5381. Volume 32 Number 2- February

[6] Stanford-Clark, A. 2013. MQTT For Sensor Networks (MQTT-SN) Protocol Specification. International Business Machines Corporation (IBM)

[7] E. R. I. S. Dewi Wijayanti. 2015. "Jurnal Inovasi Fisika Indonesia Volume 04 Nomor 03 Tahun 2015, Hal 150 - 156," *rancang bangun Alat ukur kecepatan dan arah angin berbasis Arduino Uno atmega 328p,* vol. 04, no. 03, pp. 150-156

[8] Arduino Project Hub, "sensor suhu dan kelembaban DHT11," [Online]. Available: https://www.create.arduino.cc/projecthub/Sherlly/sensor-suhu-dan-kelembaban-dht11-402fb9.

[9] Katyal, A., Yadav, R., & Pandey, M. 2016. Wireless Arduino Based Weather Station. International Journal of Advanced Research in Computer and Communication Engineering, 5(4).

[10] R. Santai, "Platform IoT berbasis Cloud," 2017. [Online]. Available: https://www.mathworks.com/help/thingspeak/