Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2023. Vol. 2. No. 3; e-ISSN : 2808-4519

Temperature Measurement System Using Arduino

Yusfrizal[1]*, Mutiara Sovina[2], Faisal Amir Harahap[3], Ivi Lazuly[4]

[1, 2, 3, 4]*Universitas Potensi Utama*
yusfrizal80@gmail.com[1*], mutiarasovina@gmail.com[2], faisalamirharahap5@gmail.com[3], ivilazuli2@gmail.com[4]

Abstract

Living things need the right temperature to meet their life needs. It is necessary to do the measurement process. One process for measuring room temperature that is widely used at this time is the automatic temperature measurement and air conditioning system. The temperature measurement system and air conditioning automation is a system that is very useful for human life. With this system is expected to make it easier to know the room temperature. LM35 is a sensor used in this study as a sensor that is very sensitive to temperature. The sensor results are processed by Arduino Uno which is then displayed in a desktop application using C# .Net and stored in the database. The room temperature measurement system, especially the one made in this project, can measure room temperature with an unstored data tolerance value ± 4 times in one minute. When the temperature exceeds the set temperature, the fan will rotate automatically as an air conditioner. This system works every second and displays the results on the desktop application and stores them in the database as the final storage place which can be developed or made a decision in the future.

Keywords: *measurement; temperature; sensor; Arduino Uno; database*

1. Introduction

In this increasingly advanced era, many solutions have been born that can solve human problems. Problems that arise due to human limitations or from other factors, are now being overcome little by little. One solution that can solve human problems is to use a computer-based control system. By using a computer-based control system, it is hoped that it can help and lighten human work and be a solution for every human problem [1].

A closed room with dense visitors requires an air conditioning system that is able to stabilize room temperature with the density of visitors who come. The density of visitors without being balanced with an adequate cooling system creates an atmosphere that is not conducive, namely the temperature in the room becomes hotter and uncomfortable [2].

Ideally, if there is a surge in visitors, the cooling system will automatically increase the maximum coolant level, if the maximum level is not stable, the system will automatically add several cooling units until the room temperature stabilizes. Vice versa if there is a decrease in the number of visitors the system also automatically reduces several cooling units until the room temperature stabilizes [3].

Microcontroller is a micro-sized controller, which can be used together with other electronic devices. The advantage possessed by the microcontroller is as a control system. The use of microcontrollers is generally used in embedded systems, namely special microcomputer sub-systems as part of a system whose controllers, namely microcontrollers, are connected to a machine. The distinctive feature of embedded systems is that they do not transform data but directly interact with external devices such as sensors and actuators.

To prove that a microcontroller can be designed for a computer-based control system and performs human work, which in particular in this study is used as a system for measuring room temperature [4].

2. Research Methodology

To complete the writing of this study the authors conducted several stages of the research method as follows:

1. Literature Study, in this method the writer looks for materials for writing this final project obtained from books or journals that are specifically about making this research.
2. Experiment, with the experimental methodology the author makes a room temperature measuring device, where all data is taken based on the good results from the design process, the programming process to the tool testing process [5].

2.1. Tools and Materials

The tools and materials used in this research are hardware and software. Table 1. is the tools and materials used.

Table 1: Tools and Materials

Hardware	Software
Arduino UNO R3	Microsoft Visual C#
9 Volt Battery	Software Arduino 1.0.5
12 Volt DC Fan/Motor	Software fritzing
TIP 120 Transistors	Microsoft Access
LM35 Temperature Sensor	
1 Kilo Ohm Resistor	
Capacitor	
Protoboard	

2.2. Design of Room Temperature Measurement and Air Conditioning System

The temperature measurement system and air conditioning automation in this study generally work in the same way. That is, when the temperature exceeds the set temperature, the fan will turn on as a cooler, then the system will send data to the desktop application as information that is displayed and then stored in the database [6]. The temperature sensor medium used in this final project is of the IC type, namely LM35 as a temperature gauge that can provide an output of -55 to 150 ℃. Automatic cooling media uses a 12V dc fan that gets a voltage source from the battery.

2.3. Application Design

In a system made to measure room temperature, the temperature range is between 20 ℃ to 50 ℃ where the normal room temperature is set at 28 ℃ to 30 ℃, so a temperature sensor is needed that is able to detect the temperature range in the room. In this system design, the LM35 sensor is used which can detect temperatures from 20 ℃ to 50 ℃. In addition, a signal condition is also needed as well as a means to display accurate room temperature processed results that can be easily monitored continuously.

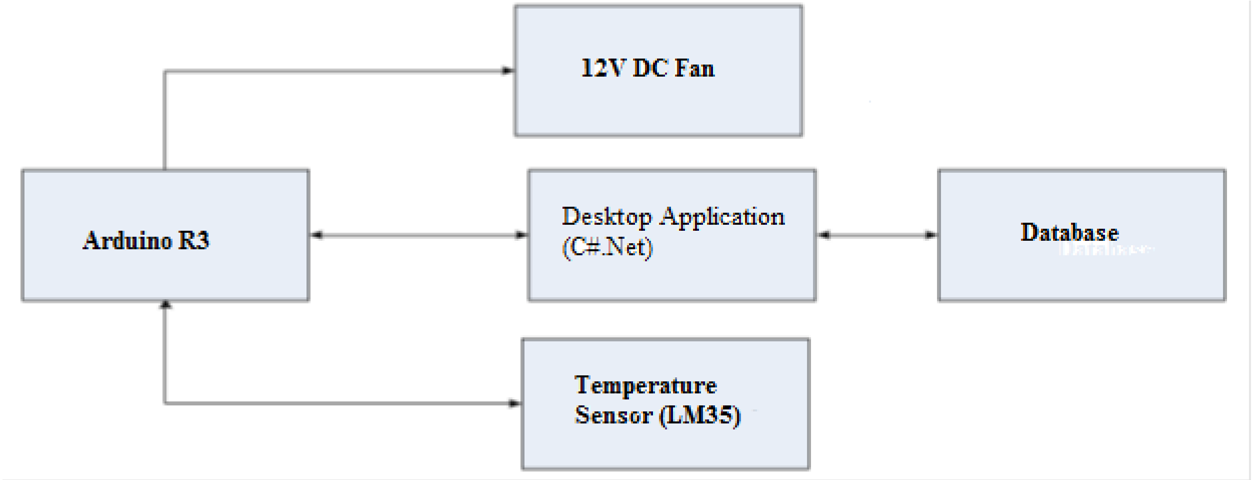


Fig. 1: System Block Diagram

From the room temperature control block diagram, it consists of several circuit blocks consisting of a Temperature Sensor (LM35), Arduino UNO R3, 12V DC Fan and the C# .Net desktop application. The sensor used is LM35 to input changes in voltage to the ADC system. Changes received are in the form of analog data and convert analog data into digital data so that it can be received by a microcontroller that only accepts digital data. The microcontroller used is Arduino UNO R3 with Atmega328. The microcontroller used is loaded with a program that will later be displayed on a computer screen using C# .Net and stored in the database.

The LM35 is used as a temperature sensor which has been calibrated directly in Celsius (℃). The output voltage (Vout) will change 10mV for every 1 ℃ temperature change. The fan used is a 12 volt DC type as a sign that if the indoor temperature exceeds the set temperature (32 ℃), the fan will turn on. The external voltage source for the fan uses a 9 volt battery.

2.3. Room Miniature Design

This room consists of one room made of cardboard. The room where the electronic circuits are located is in a cardboard room that can be opened, to make it easier to check the electronic circuits or repair them in the event of damage to the electronic circuits and to control the temperature in the room.

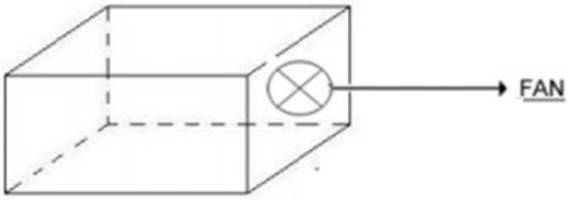


Fig. 2: Miniature Room

2.3. Arduino Circuit with LM 35 Sensor

The LM35 temperature sensor component has 3 pins, namely pin 1 to receive input 1 to 5 volts, pin 2 as analog output (Vout) and pin 3 is connected to ground [7]. At this stage the Arduino pin used is the Analog pin, namely A0 as the output of the LM35 Sensor. The picture below is an image of the Arduino circuit with the LM35 temperature sensor.

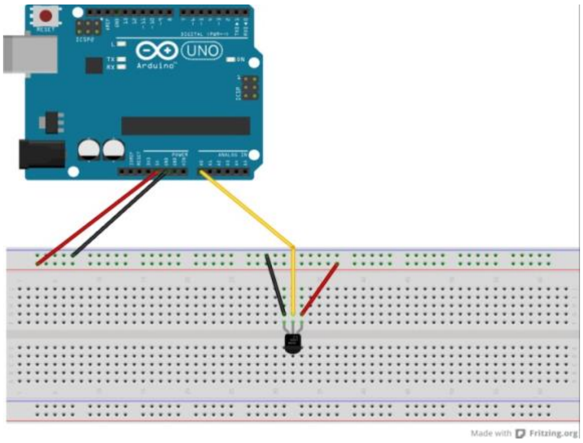


Fig. 3: Arduino Circuit and LM 35 Sensor

After assembling the Arduino circuit with the Temperature Sensor, this discussion will add a fan to the circuit. The output pin of Arduino that is used is pin 9 which is an analog pin. Pin 9 is connected to the base of the Darlington Transistor, namely TIP120, with a 1K ohm resistance. The emitter pin of the transistor is connected to ground and the collector pin is connected to a capacitor and the collector pin of this transistor is connected to a 12V DC fan which gets 9V from an external voltage using a 9V battery.

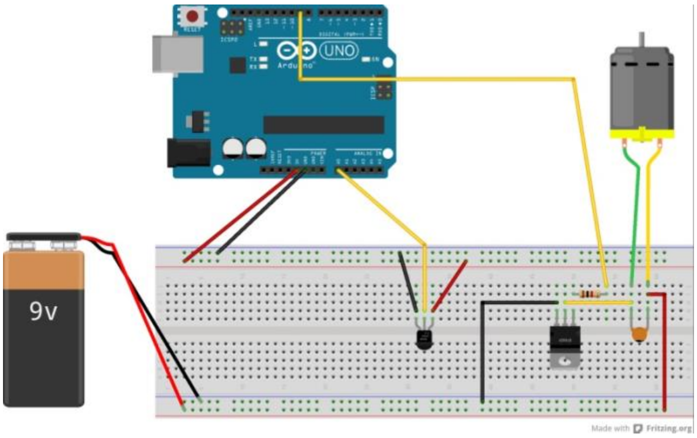


Fig. 4: Arduino Circuit and 12V DC Fan

The overall working description of the temperature measurement system and air conditioning automation in this final project, this system will start operating when a voltage source is supplied via the computer's USB port to the Arduino. This USB port is also a bridge for sending data from Arduino to a computer that is displayed on the Desktop Application and stores voltage history in the database every second using C#. Net.

The 12V DC fan installed in the system will only be ON to provide cooling when the inside temperature has reached a predetermined limit or more, and will be turned OFF again when the room temperature is below a predetermined minimum limit.

Because the temperature measurement and cold room automation system is a closed system (loop system), the temperature reading process will continue to repeat itself. This system will only stop operating if it is OFF or no voltage source is provided.

For a clearer description of the overall work description of the temperature measurement system and air conditioning automation in this study, see the figure in Figure 5., where the image is a flowchart or overall flow chart for the temperature measurement and air conditioning automation system and is displayed on the desktop application. using C# .Net as the measurement result display.

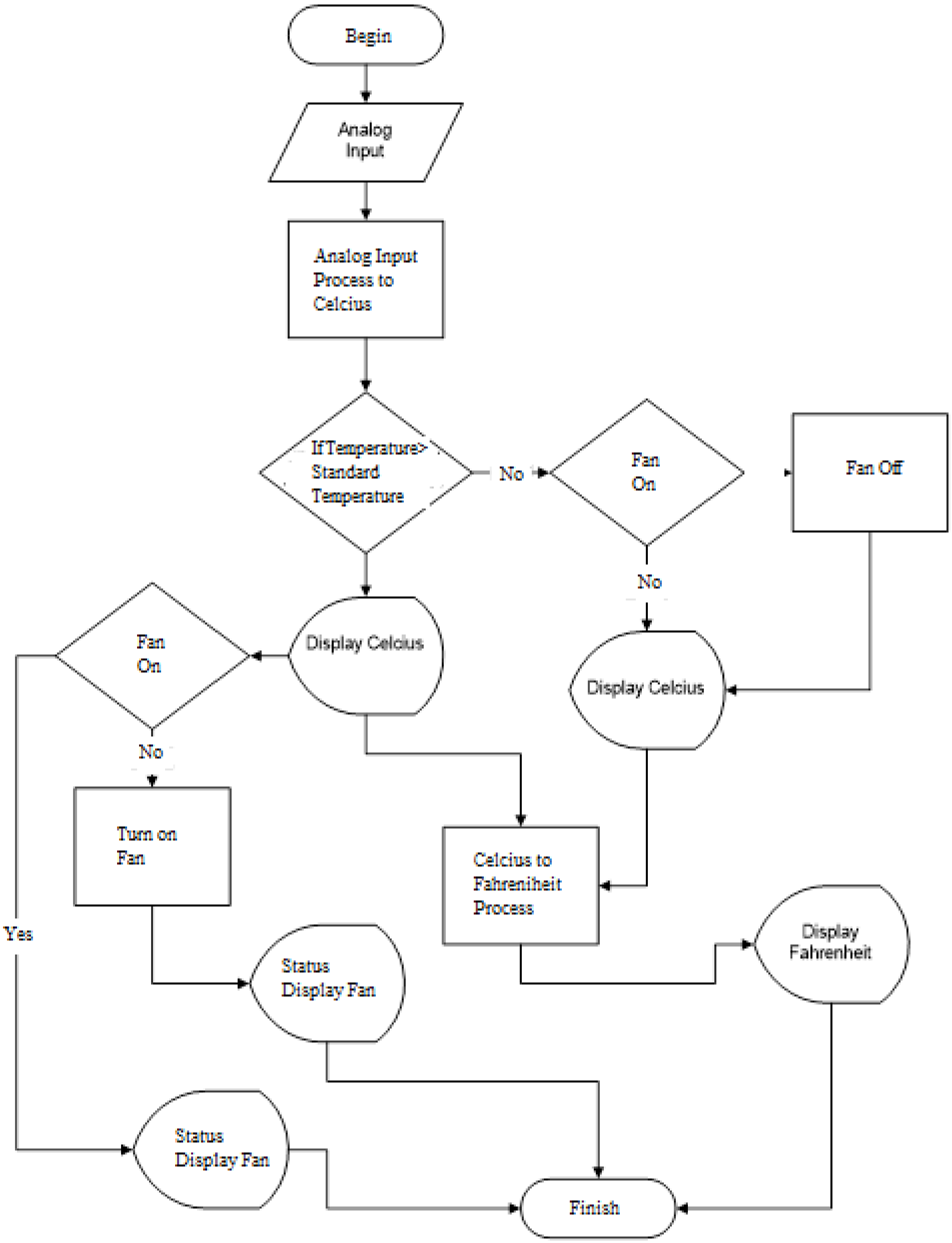


Fig. 5: Temperature Control System Flow Chart

2.3. Desktop Application Design

The main form of the desktop application is a form that functions as a display of output sent from Arduino. Information displayed is in the form of temperature in Celsius and Fahrenheit in the form of images, and the original form in a text block. In the main form there is a trigger to send data to the database and store it as historical which can later be known what the lowest temperature and highest temperature are.

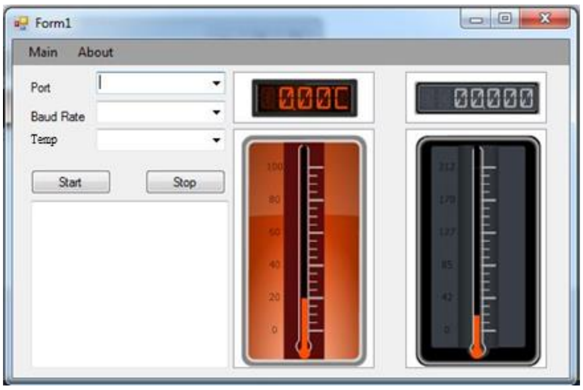


Fig. 6: Desktop Application Main Form

In the form of Figure 6 above, it can be explained Port, which is the Arduino port installed on the computer, Bolt Rate, which is the number of times per second the signal changes in analog communication data, and what is commonly used is 9600 baud, Temperature, which is a temperature variable that functions as the default temperature is set, the brown temperature graph is the temperature in Celsius and the gray one is the Fahrenheit temperature, the text block is output information from Arduino when the start button is clicked, and the stop button is to end Arduino communication with the Desktop Application.

3. Results and Discussion

Hardware testing is carried out to find out how the hardware performance that has been designed, this test includes:

1. Testing the Arduino Circuit with LM35
2. Circuit Testing with Fan

3.1. Arduino Circuit Testing with LM35

The LM35 sensor is a sensor that can measure temperatures from 0 to 100 C. This sensor accepts inputs ranging from 1V to 5V and has an output of 10mV per 1 C. The output of this sensor is input for Arduino on the analog pin which will later be calculated to display the real temperature.

$$T = (Va \times Vb \times 100.0\ ) / 1024.0 \quad (1)$$

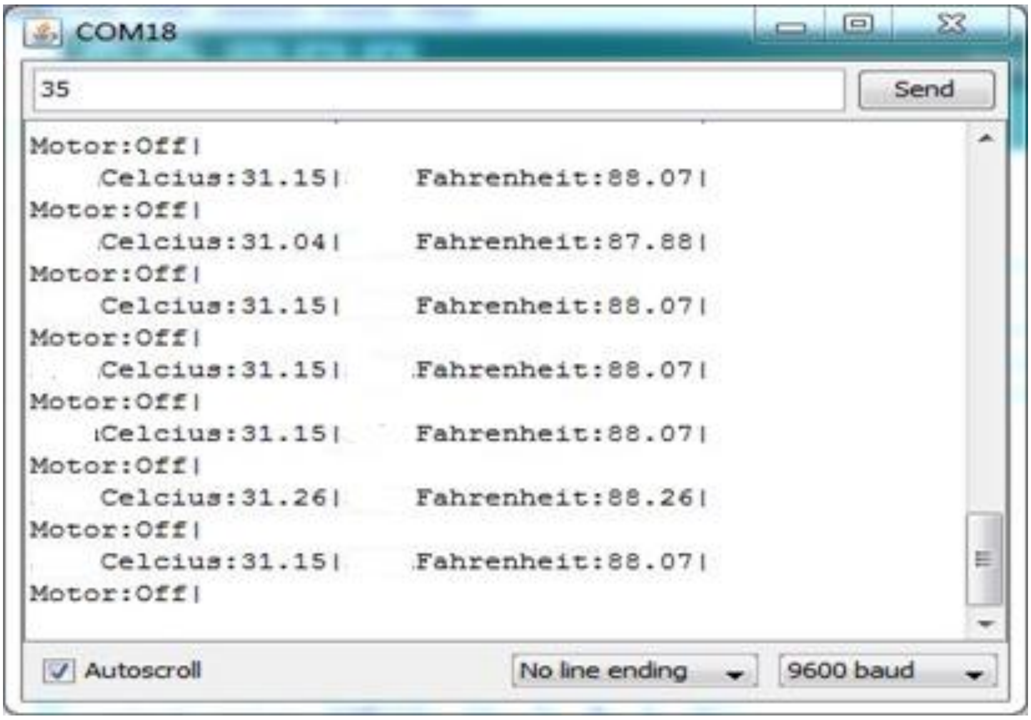
In this study, the circuit was tested at normal temperature and heated with a lighter. The results of this test are in Table 2.

Table 2: Table Of Temperature Measurement Test Results

Condition	Temperature	
	Celcius	Fahrenheit
Normal	28-31	82-84
Heated with a match	40-42	107-109

3.2. Circuit Testing with Fan

Testing the fan circuit is intended to check whether the fan is working properly when the temperature exceeds the actual temperature. After testing the heated temperature, the fan should rotate as an automatic cooler.



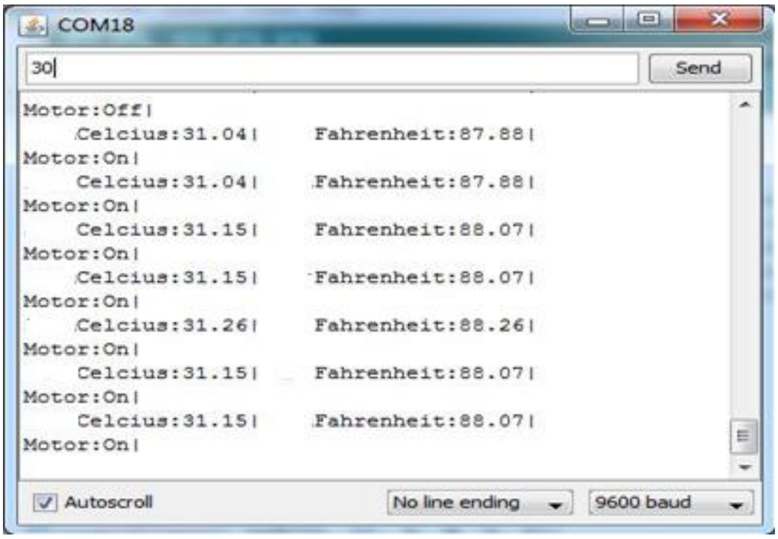
```
35

Motor:Off|
    Celcius:31.15|    Fahrenheit:88.07|
Motor:Off|
    Celcius:31.04|    Fahrenheit:87.88|
Motor:Off|
    Celcius:31.15|    Fahrenheit:88.07|
Motor:Off|
    Celcius:31.15|    Fahrenheit:88.07|
Motor:Off|
    Celcius:31.15|    Fahrenheit:88.07|
Motor:Off|
    Celcius:31.26|    Fahrenheit:88.26|
Motor:Off|
    Celcius:31.15|    Fahrenheit:88.07|
Motor:Off|
```

☑ Autoscroll   No line ending   9600 baud

Fig. 7: The fan does not rotate when the actual temperature is less than the variable temperature

In figure 7 the fan does not rotate because the actual temperature is 31 0C which is not greater than the variable temperature which is 35 0C, from the image above the information obtained is "Motor: Off" which indicates the fan is not rotating. When the temperature variable as a parameter is given a smaller input, namely 30 0C, the fan will rotate with the "Motor: On" sign as shown in Figure 8.

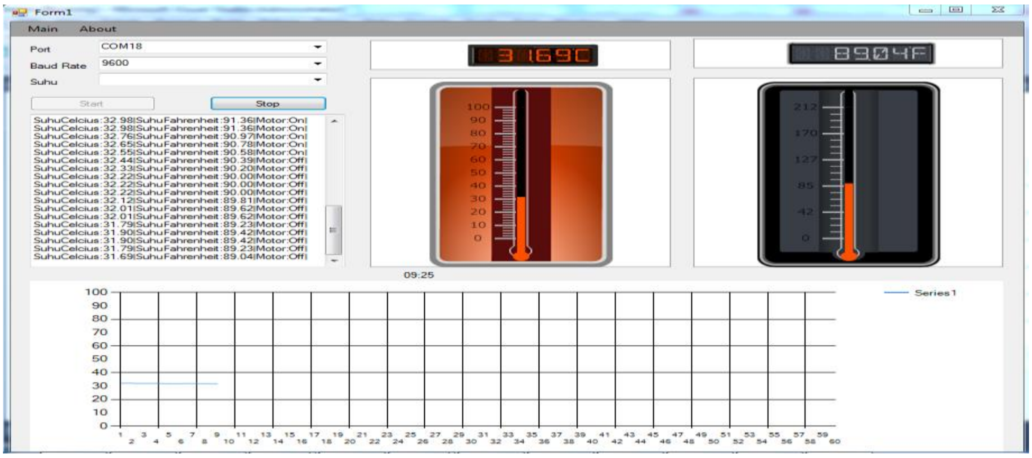


```
Motor:Off|
    Celcius:31.04|    Fahrenheit:87.88|
Motor:On|
    Celcius:31.04|    Fahrenheit:87.88|
Motor:On|
    Celcius:31.15|    Fahrenheit:88.07|
Motor:On|
    Celcius:31.15|    Fahrenheit:88.07|
Motor:On|
    Celcius:31.26|    Fahrenheit:88.26|
Motor:On|
    Celcius:31.15|    Fahrenheit:88.07|
Motor:On|
    Celcius:31.15|    Fahrenheit:88.07|
Motor:On|
```

Fig. 8: The fan turns on when the variable temperature is less than the actual temperature

3.3. Temperature Measurement Results and Application Interaction with Arduino

After hardware testing and desktop application installation is complete. So to connect between the application and Arduino use a USB Serial Port with a USB cable. When the desktop application is run and selects the Arduino port that is read on the computer, then selects the Baud Rate, which is 9600, it directly interacts with the desktop application and the desktop application immediately responds to output from Arduino visually as shown in Figure 9.



Port: COM18
Baud Rate: 9600
Suhu:

```
SuhuCelcius:32.98|SuhuFahrenheit:91.36|Motor:On|
SuhuCelcius:32.98|SuhuFahrenheit:91.36|Motor:On|
SuhuCelcius:32.76|SuhuFahrenheit:90.97|Motor:On|
SuhuCelcius:32.65|SuhuFahrenheit:90.78|Motor:On|
SuhuCelcius:32.55|SuhuFahrenheit:90.58|Motor:On|
SuhuCelcius:32.44|SuhuFahrenheit:90.39|Motor:Off|
SuhuCelcius:32.33|SuhuFahrenheit:90.20|Motor:Off|
SuhuCelcius:32.22|SuhuFahrenheit:90.00|Motor:Off|
SuhuCelcius:32.22|SuhuFahrenheit:90.00|Motor:Off|
SuhuCelcius:32.22|SuhuFahrenheit:90.00|Motor:Off|
SuhuCelcius:32.12|SuhuFahrenheit:89.81|Motor:Off|
SuhuCelcius:32.01|SuhuFahrenheit:89.62|Motor:Off|
SuhuCelcius:32.01|SuhuFahrenheit:89.62|Motor:Off|
SuhuCelcius:31.79|SuhuFahrenheit:89.23|Motor:Off|
SuhuCelcius:31.90|SuhuFahrenheit:89.42|Motor:Off|
SuhuCelcius:31.90|SuhuFahrenheit:89.42|Motor:Off|
SuhuCelcius:31.79|SuhuFahrenheit:89.23|Motor:Off|
SuhuCelcius:31.69|SuhuFahrenheit:89.04|Motor:Off|
```

Fig. 9: Arduino interaction with desktop applications

Room temperature measurement results can change very easily. Therefore, the results of temperature and air conditioning measurements displayed on desktop applications, especially in this final project, often show unstable measurement results and measurement errors occur. This can be caused by several factors, including:

1. Unstable input voltage value. Thus, changing the obtained time constant value results in measurement errors because the conversion value becomes inappropriate.
2. Delay in sending data from Arduino to Desktop Application.

Table 3: Time Temperature Measurement Results Per Second

	6/3/2023 7:29	6/3/2023 7:30	6/3/2023 9:06
0	29.54	29.43	30.5
1	29.54	29.43	30.4
2	29.54	29.43	30.4
3	29.43		30.4
4	29.43	29.43	
5	29.43	29.54	30.5
6	29.43	29.54	30.61
7	29.54	29.54	30.4
8	29.43	29.54	30.4

9		29.54	30.5
10	29.43	29.54	30.4
11	29.65	29.43	30.4
12	29.54	29.54	30.4
13	29.54	29.54	30.5
14	29.43	29.54	30.4
15	29.43	29.54	30.4
16	29.54	29.65	30.4
17	29.43	29.54	30.4
18	29.43	29.65	30.29
19	29.43	29.54	30.4
20	29.43	29.54	30.4
21	29.43		30.4
22	29.43	29.54	
23	29.43	29.65	30.4
24	29.32	29.65	30.29
25	29.43	29.65	30.29
26	29.43	29.65	30.29
27		29.65	30.29
28	29.43	29.43	30.29
29	29.32	29.54	30.4
30	29.32	29.54	30.4
31	29.32	29.54	30.29
32	29.32	29.54	30.4
33	29.43	29.54	30.4
34	29.43	29.54	30.29
35	29.32	29.65	30.29
36	29.32	29.54	30.29
37	29.32	29.54	30.29
38	29.32		30.29
39	29.32	29.54	
40	29.22	29.54	30.29
41	29.32	29.54	30.29
42	29.32	29.43	30.4
43	29.32	29.54	30.29
44	29.43	29.54	30.4
45		29.43	30.4
46	29.43	29.54	30.29
47	29.43	29.54	30.4
48	29.43	29.54	30.4
49	29.43	29.43	30.4
50	29.43	29.54	30.4
51	29.32	29.43	30.5
52	29.43	29.54	30.4
53	29.43	29.43	30.4
54	29.22	29.43	30.4
55	29.43	29.43	30.4
56	29.43		30.5

57	29.43	29.54	
58	29.43	29.54	30.29
59	29.43	29.54	30.4

Table 3. is the result of room temperature measurements stored in the database. And because the measurement results often change over time, while the measurement results do not exceed the default temperature, which is 32 C, the fan will remain off, and if on the contrary the temperature is greater than the default temperature, the fan will be on. From Table 3. especially for columns that have no contents/blanks, it can be seen that the room temperature measurement system in this task has an error in each column of ± 4 times the data is not stored. This is due to the delay and arduino conversion as well as other factors that require a slightly longer conversion time.

4. Conclusion

From this research on room temperature measuring devices and the results of experiments that have been carried out, the authors can draw the following conclusions:

1. With a room temperature gauge, you can monitor temperature and act as a DC fan automatically if the temperature has passed a predefined temperature.
2. All room temperature data is stored in the database every second and data results can be seen with applications created using C# .Net / Microsoft Access.
3. The results of temperature and air conditioning measurements used in this final report are very dependent on the value / input of the temperature sensor, the value of the input voltage, and the relationship between hardware and software.
4. From the room temperature measurement data, the measurement tolerance value is obtained as much as ± 4 times the data is not stored against measurements made by Arduino and desktop applications.

Acknowledgement

Thank you to all those who have helped complete this research.

References

[1] P. Szolovits, *Artificial intelligence in medicine*. Routledge, 2019.

[2] K.-N. Rhee, B. W. Olesen, and K. W. Kim, “Ten questions about radiant heating and cooling systems,” *Build. Environ.*, vol. 112, pp. 367–381, 2017.

[3] E. Jiaqiang *et al.*, “Orthogonal experimental design of liquid-cooling structure on the cooling effect of a liquid-cooled battery thermal management system,” *Appl. Therm. Eng.*, vol. 132, pp. 508–520, 2018.

[4] F. A. Hassani *et al.*, “Smart materials for smart healthcare–moving from sensors and actuators to self-sustained nanoenergy nanosystems,” *Smart Mater. Med.*, vol. 1, pp. 92–124, 2020.

[5] A. H. A. Al-Waeli *et al.*, “Evaluation of the nanofluid and nano-PCM based photovoltaic thermal (PVT) system: An experimental study,” *Energy Convers. Manag.*, vol. 151, pp. 693–708, 2017.

[6] E. Png, S. Srinivasan, K. Bekiroglu, J. Chaoyang, R. Su, and K. Poolla, “An internet of things upgrade for smart and scalable heating, ventilation and air-conditioning control in commercial buildings,” *Appl. Energy*, vol. 239, pp. 408–424, 2019.

[7] P. D. P. Adi and A. Kitagawa, “Performance evaluation WPAN of RN-42 bluetooth based (802.15. 1) for sending the multi-sensor LM35 data temperature and raspBerry pi 3 Model B for the database and internet gateway,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 12, pp. 612–620, 2018.