

Using Android to Set Fan Spin

Imam Syafi'i¹, Dwi Hadidjaja¹, Jamaaluddin Jamaaluddin*¹

¹Department of Electrical Engineering, Faculty of Science and Technology
Universitas Muhammadiyah Sidoarjo

Jalan Raya Gelam 250, Candi, Sidoarjo, Telp (031) 318945444

*Corresponding author, e-mail: jamaaluddin@umsida.ac.id

Abstract – *The Android operating system on smartphones has developed rapidly and become a favorite among other operating systems. In addition to applications that can be obtained for free, making Android applications is also easy because the platform is open source. Hence, it allows users to create or develop applications already available. By utilizing convenience, Android smartphones can be used to control household appliances such as fans. Applications on Android can be made with the MIT App Inventor software, which can be used to adjust the fan spin. Android applications can ease users to increase or decrease the spin speed of the fan from a distance. Arduino Nano microcontroller and Relay were used to adjust the fan spin, while the Bluetooth Module was used to connect the microcontroller and Android smartphone. After the testing process, the fan spin could be set using Android with a maximum distance of 15.2 meters without barriers and 11 meters with a wall barrier. Fan spin has 3 speed, 1473 rpm for the high spin, 1462 rpm for the medium spin, and 1451 rpm for the low spin, while the timer used consists of 10 minutes, 20 minutes, 30 minutes, 60 minutes, and 120 minutes.*

Keywords: *Arduino, Bluetooth, Fan, Spin*

I. Introduction

Nowadays, many household appliances employ automatic systems to help users operate them, starting from the washing machine, rice cooker, air conditioning (AC), and others. They even use artificial intelligence technology [1][2]. The rapid development of technology makes humans competing to create equipment that can facilitate users in operating household appliances [3]. One of them is controlling household appliances through Android smartphones. Android smartphones are in demand by various groups of people, ranging from teenagers to adults, as the applications are easily made and obtained for free.

The simplicity of Android smartphones makes many smartphone maker vendors competing to create Android smartphones with the best technology and an affordable price. The number of smartphones available in the market and thousands of Android application developers leads to the creation of applications that can be used to regulate fan spin.

Fans today still use push or pull switches and a remote to change the spin. Aside from being suitable for all brands and types of fans, the advantage of using Android lies in the level of fan response faster after receiving commands. It is because of the program on Arduino using the millis function as the main program. In addition to a faster response, the millis function is also multi-tasking as it can operate more than one command simultaneously.

However, fans are still a favorite among people due to their easier maintenance than air conditioning (AC). The required maintenance is only cleaning the dirt around the blade and lubricating the rotating parts. In contrast, AC maintenance is more complicated and requires more expensive costs. The indoor AC unit must be cleaned every month to prevent blockages in the evaporator section. Freon contents checking must also be routinely carried out.

Based on the above problems, this study entitled USING ANDROID TO SET FAN SPIN was carried out to examine the problems in-depth and explain them in this paper.

II. Theoretical Review

II.1. Arduino Nano

The Arduino hardware module was created by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David A. Mellis, and Nicholas Zambetti in Ivrea, Italy in 2005. Arduino is a derivative of the wiring platform and processing language. Arduino uses open source licensing standards, including hardware (circuit schemes, PCB design), bootloader firmware, documents, and Integrated Development Environment (IDE) software as an application to program Arduino [4][5].

The boot loader in flash memory must be programmed on the microcontroller to allow the microcontroller to communicate with the Arduino IDE. Arduino modules that have been released since 2009 include Diecimila, Uno, Duemilanova, Mega, LilyPad, and Nano [6].

Arduino language is the development of a programming language wiring platform with a more straightforward language structure than C language. Sketch language can also be combined with the wiring language or native languages of the AVR family microcontrollers, such as CodevisionAVR or WinAVR.

Arduino Nano is one of the outputs of an inexpensive Arduino Module. It is small and has more ADC PINs than the Arduino Uno Module often used. Arduino Nano uses an Atmega168 microprocessor equipped with a 16MHz oscillator and a 5V supply voltage [7].

II.2. The HC-05 Bluetooth Module

Bluetooth is a wireless communication technology operating in the 2.4 GHz unlicensed Industrial, Scientific, and Medical (ISM) frequency band using a frequency hopping transceiver that can provide real-time data and voice communication services between Bluetooth hosts with limited service range (about 10 meters). Bluetooth can be used to exchange information between devices [8].

HC series Bluetooth Modules have many types or variants, broadly divided into two types, industrial series (HC-03 and HC-04) and civil series (HC-05 and HC-06). The serial Bluetooth Module referred to as the BT Module, is used to transmit TTL serial data via Bluetooth. This BT Module consists of two types, master and slave.

The serial number can recognize the BT HC Module series. If the serial number is even (HC-06 and HC-04), then the BT Module works as a slave

and cannot be changed into a master. While the BT HC Module uses odd serial numbers (HC-05 and HC-03), and the default conditions are usually set as slave mode, but users can change them to master mode with specific AT Command [9].

II.3. Android Smartphones

Android is a Linux based operating system used on touch screen phones, smartphones, and tablets. Android provides an open platform for developers to create their applications. There has been much available software used to create applications on Android, for example, MIT App Inventor. Android is a new generation of mobile platforms allowing developers to carry out development as they want.

Android is the most popular platform for its open-source nature and providing millions of free applications that can be downloaded through the Google Play Store. The massive number of Android developers causes the emergence of many new vendors. Previously, Android devices were dominated by Samsung, but now many new smartphone vendors have sprung up, such as Lenovo, Asus, HTC, Smartfren, Cross, Dell, Nexus, and many more.

Android is distributed under the Apache Software License (ASL/Apache2), allowing for the other distribution. Commercialization of developers (handset manufacturers in particular) can improve the platform without providing improvements to the public. Android has two distributors, Google Mail Service (GMS) and Open Handset Distributor (OHD). GMS is an Android distributor receiving full support from Google, while OHD is an Android distributor without direct support from Google [5].

II.4. Motor Capacitors

Motor Capacitors are widely used in household appliances such as water pump motors, washing machine motors, and fan motors. The simple form of construction, able to work on 220V voltage supply, and low power usage, makes it widely used in household appliances. The stator winding consists of the primary and auxiliary winding. The number of the primary winding is less than the auxiliary winding, and the primary winding wire diameter is higher than that of the auxiliary wire winding [6].



Fig. 1. Fan capacitor motor

Two types of 1 phase induction motors widely used in fans in Indonesia are capacitor-run motors and shaded-pole motors. The capacitor-run motors have 2 coils on their stator, the primary and auxiliary coils. The capacitors on the capacitor-run motors are installed in series with the auxiliary coil and parallel with the primary coil.

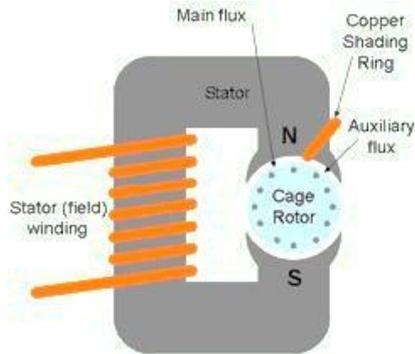


Fig. 2. Shaded-pole motor

Shaded-pole motor construction is different from other 1 phase induction motors. The difference is in the stator section split into two and given a ring at the end. Moreover, the type of rotor used is the same as other 1 phase induction motors, namely cage rotors. The rotating field produced by the shaded-pole motor is caused by the induction of the ring short-circuited at the end of the pole with the influence of magnetic induction from the other side originated from the coil placed between the stator halves used for the rotor. Shaded-pole motors are used in fans with small power capacities [7].

II.5. MIT App Inventor

App Inventor is an application originally provided by Google and is now held by the Massachusetts Institute of Technology (MIT). It was completed on July 12, 2010, and released to the public on December 15, 2010. Google stopped the App Inventor project on December 31, 2011. It allows anyone to create applications that can run on the Android operating system. Users can create graphical GUI displays and object visual drag and drop features in making the application.

The language used by the App Inventor is a block language easy to understand even though the user is not a professional programmer. The user only has to specify the application to be made and select the program block and drag it to a worksheet that has been determined by the App Inventor and then arrange it like a puzzle game [8].

III. Research Methodology

III.1. Block Diagram System

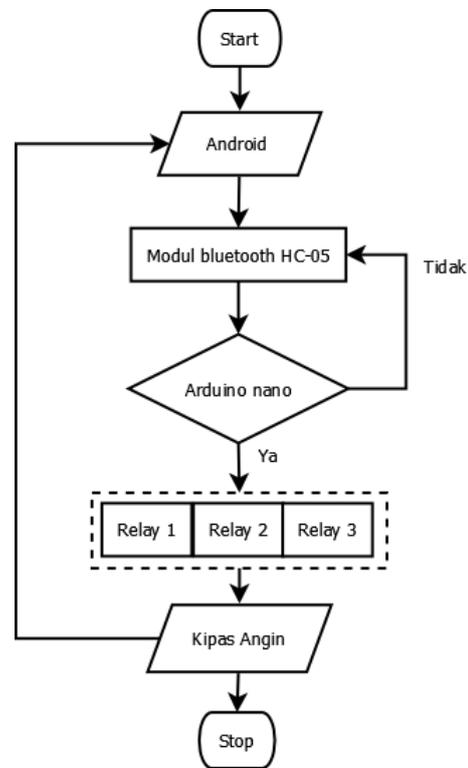


Fig. 3. Block diagram system

In the application that has been installed on Android, by touching the Bluetooth image, the Android smartphone will look for an active HC-05 Bluetooth address. When connected, the HC-05 Bluetooth Module led will blink slowly. When the high button is touched, the Android smartphone will send data in the form of number 1 to the Arduino Nano Module, which will then send commands to Relay 1 to work, so that the fan will work with high spin. If the medium button is touched, Android will send data in the form of number 2 to the Arduino Nano, and it will send commands to Relay 2, and the fan will spin with a medium spin. The low button has data in the form of number 3, which, when touched, the Relay 3 will work, and the fan spins with low rotation. The fan will turn off when the button off is touched because Android will send data in the form of number 4 to the Arduino Nano, meaning that it will turn off the entire Relay working. In the timer menu on Android, the fan can turn off automatically according to the input. Five timers used are 10 minutes, 20 minutes, 30 minutes, 60 minutes, and 120 minutes.

III.2. Flowchart Application

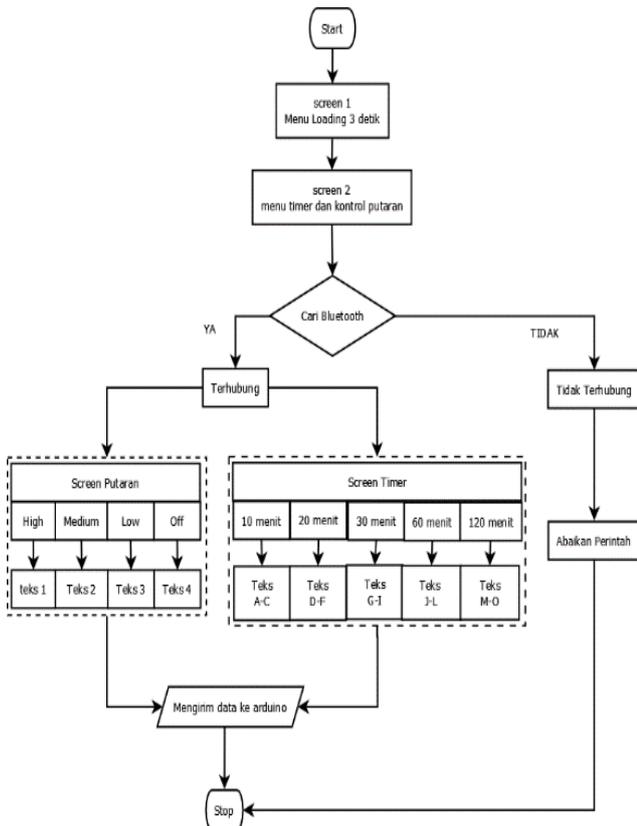


Fig. 4. Flowchart application

App Inventor is used to create applications on Android. App Inventor uses block language to fill programs for each selected component. To use the App Inventor software, users must use an internet connection (online) and log in with an email account. App Inventor software can be accessed at <http://appinventor.mit.edu>, and then a new project window will appear. The results of the programming carried out in the App Inventor software are saved with the .apk file format with the name of fan control. Thereby, it can be installed on Android.



Fig. 5. Application initial display



Fig. 6. Application main display

The picture above explains that the application display has 4 screens. According to the selected time, the fan control application consists of 4 main menus, timer, manual, info, and wind quit. The manual menu for adjusting fan spin consists of high, medium, and low. The info menu contains HC-05 Bluetooth wiring to Arduino Nano, Relay wiring to Arduino Nano, and a code table for each button. Moreover, the quit menu is used to exit the application.

Feedback sent by Arduino will be displayed in the application, as shown in Figure 6. Each button has a different code as an identifier in the Arduino program. The code for each button can be seen in Table 1.

TABLE I
CODE ON EACH APPLICATION BUTTON

No	Main Button	Speed Button	Text Sent to the Arduino
1	10 minutes	HIGH	A
		MEDIUM	B
		LOW	C
2	20 minutes	HIGH	D
		MEDIUM	E
		LOW	F
3	30 minutes	HIGH	G
		MEDIUM	H
		LOW	I
4	60 minutes	HIGH	J
		MEDIUM	K
		LOW	L
5	120 minutes	HIGH	M
		MEDIUM	N
		LOW	O
6	High	-	1
7	Medium	-	2
8	Low	-	3
9	Off	-	4

TABLE II
APPLICATION INSTALLATION ON ANDROID SMARTPHONES

No.	Smartphone Type	Android Version	Testing Type			
			Screen (inch)	Application Installation	Application Display	Application Performance
1	Lenovo A7000	Marshmallow 6.0	5.5	Succeed	Full	Fast
2	Samsung Galaxy J1 mini	Lollipop 5.1	5	Succeed	Cut off	Fast
3	Andromax A	Lollipop 5.1	5.5	Succeed	Full	Slow
4	Oppo f1s	Lollipop 5.1	5.5	Succeed	Full	Fast
5	Asus Zenfone	Jelly Bean 4.1	4	Succeed	Cut off	Slow

Table 2 explains that the application can be installed on smartphones of different types and brands. The difference is in the display and performance of each type of smartphone. For maximum results, the application display is on the type of smartphone with 5.5-inch screen size, such as the Lenovo A7000, Andromax A, and Oppo F1s. For a screen size of 4 to 5 inches, such as Asus Zenfone and Samsung Galaxy J1 Mini, the screen display is cut off slightly in the spin status menu, but it can be resolved by activating the scrollable menu on the App Inventor design.

Android brands and versions also affect application performance. Maximum performance can be felt on the Lenovo A7000, Samsung Galaxy J1 Mini, and Oppo F1s, while the performance is slightly slowed on Andromax A and Asus Zenfone. The Samsung and Oppo brands are better than Andromax in terms of performance even though they have the Android version of Lollipop 5.1.

Testing the application program shows that the codes read to set the fan spin timer are the capital letters A to O, while the manual setting codes are numbers 1 to 4. Moreover, the status in the application does not appear, and all indicator lights are off.

III.3. Arduino Flowchart

Figure 7 explains that Arduino Nano will read data sent via Bluetooth by an application on Android. In addition to receiving data, Arduino also sends data to Arduino in the form of fan spin status. Thereby, if there are changes in spin, it can be seen on Android. Each data sent by Android will be processed by the Arduino Nano to move the Relay installed on the Arduino Nano pins. The Relay output is connected to each motor speed coil.

The Arduino ports connected to Relay are D7, D8, and D9 ports, while the indicator led ports are connected to D2, D3, and D4 ports, respectively. The RX Bluetooth pin is connected to the TX Arduino, and the TX Bluetooth pin is connected to the RX

Arduino. The 5V power supply is connected to Arduino and Coil Relay.

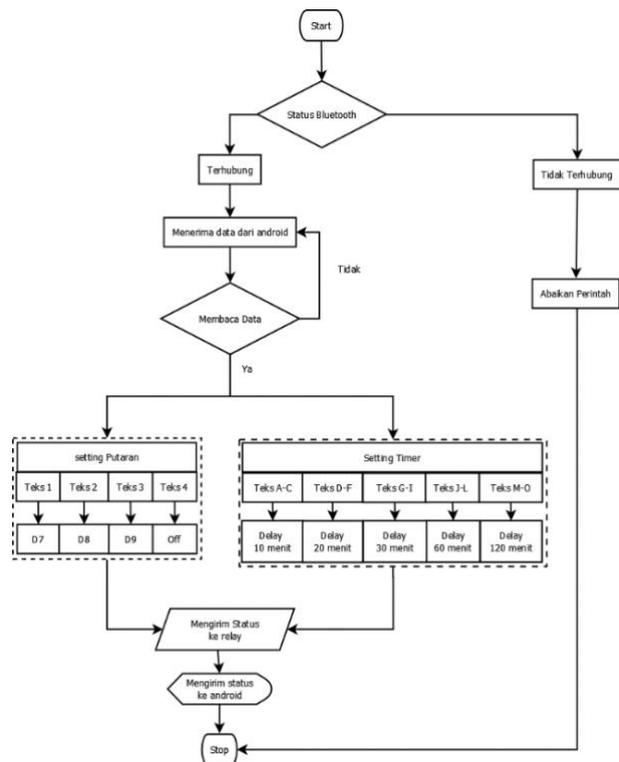


Fig. 7. Arduino Flowchart

TABLE III
ARDUINO PORT USAGE

No	Arduino Port	Usage
1	TX	RX HC-05 Bluetooth
2	RX	TX HC-05 Bluetooth
3	D2	High led indicator
4	D3	Medium led indicator
5	D4	Low led indicator
6	D5	Timer led indicator
7	D7	Driver Relay 1
8	D8	Driver Relay 2
9	D9	Driver Relay 3
10	5V power supply	Coil Relay and Arduino

IV. Results and Discussion

Testing was carried out to discover how this tool worked. Moreover, the test data gathering was carried out in every part of the system and the whole system. The tests were carried out as follows:

IV.1. Testing the Android Application toward HC-05 Bluetooth

Testing the Android application toward Bluetooth aims to determine the maximum distance of Bluetooth in sending or receiving data to Arduino.

TABLE IV

TESTING ANDROID APPLICATION DATA TRANSMISSION TOWARD HC-05 BLUETOOTH MODULE WITHOUT BARRIERS

No.	Distance (cm)	Data Sent to Arduino Port						Error (%)
		D2	D3	D4	D7	D8	D9	
1	50	1	1	1	1	1	1	0
2	250	1	1	1	1	1	1	0
3	450	1	1	1	1	1	1	0
4	650	1	1	1	1	1	1	0
5	850	1	1	1	1	1	1	0
6	1050	1	1	1	1	1	1	0
7	1100	1	1	1	1	1	1	0
8	1350	1	1	1	1	1	1	0
9	1520	1	1	1	1	1	1	0
10	1570	0	0	0	0	0	0	100

Description: 0 = Unsent data
1 = Sent data

Where:

$$\mu = \frac{x_1+x_2+x_3+\dots+x_n}{n} \quad (1)$$

$$\text{Error Percentage} = (1 - \mu) \times 100\% \quad (2)$$

Description: μ = Average Value

X_1 = First Data

n = Data Amount

After testing the distance of sending Android application data to the HC-05 Bluetooth Module without a barrier with a range of 50 cm to 1570 cm, it can be concluded that the farthest distance to send data was 1520 cm with an error percentage of 0%. If the distance exceeds 1520 cm, then the data cannot be sent with an error percentage of 100%.

TABLE V

TESTING ANDROID APPLICATION DATA TRANSMISSION TOWARD HC-05 BLUETOOTH MODULE WITH A WALL BARRIER

No.	Distance (cm)	Data Sent to Arduino Port						Error (%)
		D2	D3	D4	D7	D8	D9	
1	50	1	1	1	1	1	1	0
2	250	1	1	1	1	1	1	0
3	450	1	1	1	1	1	1	0
4	650	1	1	1	1	1	1	0
5	850	1	1	1	1	1	1	0
6	1050	1	1	1	1	1	1	0
7	1100	1	1	1	1	1	1	0
8	1350	0	0	0	0	0	0	100
9	1520	0	0	0	0	0	0	100
10	1570	0	0	0	0	0	0	100

Description: 0 = Unsent data
1 = Sent data

Where:

$$\mu = \frac{x_1+x_2+x_3+\dots+x_n}{n} \quad (1)$$

$$\text{Error Percentage} = (1 - \mu) \times 100\% \quad (2)$$

Description: μ = Average value

X_1 = First data

n = Data amount

Testing the distance of sending Android application data to the HC-05 Bluetooth Module with a wall barrier at a distance of 50 cm to 1570 cm revealed that the farthest distance of data transmission was 1100 cm with an error percentage of 0%. If the distance exceeded 1100 cm, then the data could not be sent with a percentage error of 100%. It indicates that the wall barrier can reduce the distance of data transmission.

TABLE VI

TESTING ANDROID APPLICATION DATA TRANSMISSION TOWARD HC-05 BLUETOOTH MODULE WRAPPED IN PLASTIC

No.	Distance (cm)	Data Sent to Arduino Port						Error (%)
		D2	D3	D4	D7	D8	D9	
1	50	1	1	1	1	1	1	0
2	250	1	1	1	1	1	1	0
3	450	1	1	1	1	1	1	0
4	650	1	1	1	1	1	1	0
5	850	1	1	1	1	1	1	0
6	1050	1	1	1	1	1	1	0
7	1100	1	1	1	1	1	1	0
8	1350	1	1	1	1	1	1	0
9	1520	1	1	1	1	1	1	0
10	1570	0	0	0	0	0	0	100

Description: 0 = Unsent Data
1 = Sent Data

Where:

$$\mu = \frac{x_1+x_2+x_3+\dots+x_n}{n} \quad (1)$$

$$\text{Error Percentage} = (1 - \mu) \times 100\% \quad (2)$$

Description: μ = Average Value
 X_1 = First Data
 n = Data Amount

The result of testing the distance of sending Android application data to the HC-05 Bluetooth Module wrapped in plastic at a distance of 50 cm to 1570 cm showed that the farthest distance of data transmission was 1520 cm with an error percentage of 0%. If the distance exceeded 1520 cm, the data could not be sent with an error percentage of 100%. It implies that the test results of data transmission using Bluetooth wrapped in plastic and without barriers are the same.

IV.2. Arduino Nano Testing Toward Motor Capacitors

Arduino Nano testing toward motor capacitors aimed to determine whether the Arduino Nano could change the spin of the capacitor-run motor. Capacitor motor-run had 2 coils on the stator, the primary and auxiliary coils. The speed coil was placed in the auxiliary coil because the number of the auxiliary coil was less than the primary coil.

The primary coil had an email wire cross-section of 0.12 mm while the cross-section of the auxiliary coil, sp1 coil, and sp2 coil had an email wire of 0.10 mm. High spin speeds were obtained from auxiliary coils. Meanwhile, auxiliary coils were connected in series with sp1 coils for medium spin speed, and for low spin speed, auxiliary coils were connected in series with sp1 and sp2 coils.

Table 7 indicates that the higher the resistance of the motor coil is, the slower the spin of the motor will be, due to the stronger the magnetic field produced. As the capacitor-run motor increased its speed, the flow passing through it would also increase.

IV.3. Tool Testing Results

The testing tool was carried out to test the whole system. The setting fan spin using Android could work as desired. The steps in testing the tool were as follows:

1. Connecting the whole series of fan spin speed controls using Android

2. Connecting the Arduino Nano to the laptop with a USB cable
3. Running the Arduino program and uploading it to the Arduino Nano
4. Inserting the plug into the wall socket to make the HC-05 Bluetooth work, and pairing the Bluetooth with a smartphone
5. Opening the fan control application on Android and then opening the manual or timer menu
6. Connecting the HC-05 Bluetooth to allow the application to connect to the fan spin setting circuit

The working principle of this tool was the fan spin was regulated by the Arduino Nano through ports D7, D8, and D9 connected to the Relay. Every port used was set via the Android application using Bluetooth communication.

The application could read the HC-05 Bluetooth if it was paired first. To connect the Android application and the fan spin setting, the HC-05 Bluetooth should be selected. The HC-05 Bluetooth indicator led would blink slowly when it was connected to the Android application.



Fig. 8. Test result of 10-minutes timer

Figure 8 shows the results of the fan timer testing. When manual control of the high button was pressed, the fan would spin with a speed of 1473 rpm, and the speed status of the application was high. If the medium button was pressed, the fan would spin with a speed of 1462 rpm with the medium speed status in the application. Moreover, pressing the low button would spin the fan with a speed of 1451 rpm, changing the application's status to low.

TABLE VII
 ARDUINO NANO TESTING ON MOTOR CAPACITOR

No.	Monitor Serial Code	Spin Speed Status	Flow (A)	Spin Speed (rpm)	Coil Resistance (Ω)	Motor Condition
1	1	HIGH	0.16	1473	730	Spinning
2	2	MEDIUM	0.14	1462	842	Spinning
3	3	LOW	0.11	1451	954	Spinning
4	4	STOP	0	0	0	Stay
5	a	Cannot be read	0	0	0	Stay

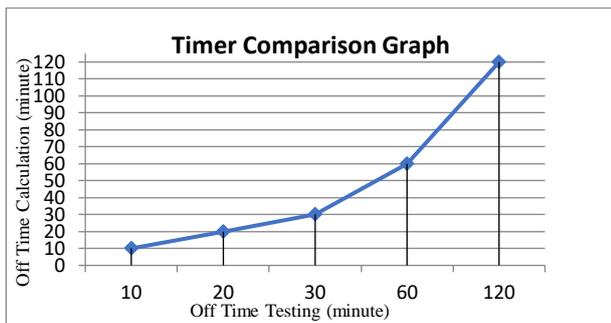


Fig. 9. Timer comparison graph

The fan spin setting test automatically began from 10 minutes to 120 minutes at a distance of 2 meters to 15 meters. The timer calculation results with the testing timer were the same, even though it was carried out at different distances with an error percentage of 0%. It was because the distance only affected the sent and unsent data, and did not affect the calculation of the programmed timer.

V. Conclusion

After testing the fan spin setting using Android, the following conclusions can be drawn:

1. Application created through the App Inventor and installed on an Android smartphone was able to connect to the fan spin setting circuit via the HC-05 Bluetooth and successfully adjusted fan spin with 3 types of speed, high, medium, and low. The values obtained were 1473 rpm for high speed, 1462 rpm for medium speed, and 1451 rpm for low speed.
2. The fan spin setting circuit, in which each speed coil connected to the Relay, could change the fan spin controlled using Android with a maximum distance of 15.2 meters without a barrier and 11 meters with a wall barrier.
3. The Android application was equipped with a timer menu to turn off the fan according to the selected time. There were several timer options, 10 minutes, 20 minutes, 30 minutes, 60 minutes, and 120 minutes. The timer test results showed that the manual calculation value was the same as the test result value. Timer programmed in the Arduino using the millis function was able to change the fan spin when the timer was working and made program response faster.

References

- [1] Jamaaluddin, I. Robandi, and I. Anshory, "A very short-term load forecasting in time of peak loads using interval type-2 fuzzy inference system: A case

study on java bali electrical system," *J. Eng. Sci. Technol.*, vol. 14, no. 1, pp. 464–478, 2019.

- [2] Jamaaluddin, I. Robandi, I. Anshory, Mahfudz, and R. Rahim, "Application of interval type-2 fuzzy inference system and big bang big crunch algorithm in short term load forecasting new year holiday," *J. Adv. Res. Dyn. Control Syst.*, 2020.
- [3] A. Solih and J. Jamaaluddin, "Rancang Bangun Pengaman Panel Distribusi Tenaga Listrik Di Lippo Plaza Sidoarjo Dari Kebakaran Berbasis Arduino Nano," *JEEE-U (Journal Electr. Electron. Eng.*, 2017.
- [4] R. H. Sudhan, M. G. Kumar, A. U. Prakash, S. A. R. Devi, and S. P., "ARDUINO ATMEGA-328 MICROCONTROLLER," *IJIREEICE*, 2015.
- [5] J. Ozer and H. Blemings, *Practical arduino: Cool projects for open source hardware*. 2010.
- [6] E. J. Istiyanto, *PENGANTAR ELEKTRONIKA & INSTRUMENTASI*. Yogyakarta: Andi, 2014.
- [7] A. Kadir, *APLIKASI MIKROKONTROLLER DAN PEMROGRAMAN ARDUINO*. Yogyakarta: Andi, 2013.
- [8] H. Andrianto, *PEMROGRAMAN MIKROKONTROLLER AVR ATMEGA 16*, 2nd ed. Bandung: Informatika, 2015.
- [9] A. Yulianto, "Sistem Pengontrol Lampu dan Kipas Angin Dalam Ruangan Dengan Menggunakan Operating System Android Media Bluetooth (Software)," *Politek. Negeri Sriwij.*, no. Tugas Akhir, pp. 1–6, 2015.
- [10] A. Yulianto, "SISTEM PENGONTROL LAMPU DAN KIPAS ANGIN DALAM RUANGAN DENGAN MENGGUNAKAN OPERATING SYSTEM ANDROID MEDIA BLUETOOTH (SOFTWARE)," *Politek. Negeri Sriwij.*, 2015.
- [11] Siswoyo, *Teknik Listrik Industri*, vol. 1. Direktora Pembinaan Sekolah Menengah Kejuruan, 2008.
- [12] P. Sumardjati, *TEKNIK PEMANFAATAN TENAGA LISTRIK*, Jilid 1. Jakarta: Direktora Pembinaan Sekolah Menengah Kejuruan, 2008.
- [13] K. Wahana, *PEMROGRAMAN ANDROID DENGAN APP INVENTOR*. Yogyakarta: Andi, 2013.

Authors' information

Imam Syafi'I is a student of Electrical Engineering,, Department of Electrical Engineering, Faculty of Science and Technology, Universitas Muhammadiyah Sidoarjo.



Dwi Hadidjaja is a lecturer at the Department of Electrical Engineering, Faculty of Science and Technology, Universitas Muhammadiyah Sidoarjo.



Jamaaluddin obtained his B.Eng. in Electrical Engineering from Universitas Brawijaya, Malang, Indonesia, in 1993. His master study was done at the Master of Management from Universitas Muhammadiyah Yogyakarta. He currently is a lecture in Department of Electrical Engineering, Universitas Muhammadiyah

Sidoarjo. His main research interest is in power system analysis, energy audit, energy conversion, power electronics, power system control, and load forecasting.