Jurnal Mantik
Volume 4, Number 1, May 2020, pp. 962-968 E-ISSN 2685-4236
https://iocscience.org/ejournal/index.php/mantik

Android-Based Monitoring and Control of Electricity Consumption

Rahmat Azyad Samallo[1*], Septi Andryana[2], Ratih Titi Komala Sari[3]

[123]Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional, Jakarta, Indonesia

Email: [1,*]rahmatsamallo@gmail.com, [2]septi.andryana@civitas.unas.ac.id, [3]ratih.titi@civitas.unas.ac.id
Email Penulis Korespondensi: rahmatsamallo@gmail.com

ARTICLE INFO

Article history:
Received: 04/04/2020
Revised: 20/04/2020
Accepted: 30/05/2020

Keywords:
Android,
Electricity,
NodeMCU ESP8266
Relay

ABSTRACT

The purpose of this study is to create a system of "Monitoring and Control of Electricity Based on Android" where electricity is now an important part of people's daily lives and the community cannot be far from electricity because daily needs require electricity. There are bad habits of the people in using unnecessary and excessive electricity such as always turning on lights, fans, TVs and others when not at home so that electricity bills become expensive and uncontrolled. Monitoring and controlling electricity is one way to help the efficiency of the budget incurred due to unnecessary electricity usage. By using the Android Electric Monitoring and Control can be done remotely that can move the relay that is in the NodeMCU module ESP8266 so that electricity can be controlled with the system to be created and users no longer need to worry about electricity costs and forgetfulness to turn off the electricity that lights up because it can monitored via Android.

1. Introduction

Electrical energy has become an important part of modern human life.As a resource that is always used in everyday life, electricity must be used optimally and adjusted because it is still a resource that has limits.[1][2].

The dependence of society on electricity creates bad habits. Many people sometimes let one living device either be at home or not at home[3]. An example is forgetting to turn off the fan, TV, AC, and others, of course this is a very wasteful and inefficient thing to do. NodeMcu ESP8266 is a versatile wifi module because it is equipped with GPIO, ADC, UART and PWM[4]. In this study, nodeMCU ESP8266 functions as a client and an Android-based electric controller. NodeMCU ESP8266 will receive input from sensors to control electricity according to sensor conditions sent via Android and Relay[5][6].

Android includes a Linux-based operating system developed specifically for mobile devices such as smartphones or PC tables, such as Symbian used by Nokia and Blackberry OS, for example, Microsoft Windows, which is very familiar to users of computer and laptop devices, if we analize, Android is the windows while the smartphone or tablet is the computer unit, besides that Android itself is distributed as open source[7][8].

Relay is an electronic component that is moved by an electric current[9]. The relay is a lever wound with a wire on a nearby iron rod (solenoid). When the solenoid is electrified, the lever will be pulled because of the magnetic force that occurs on the solenoid so that the switch contacts will close[10]. When the electric current has stopped, the switch will open again.

From some of the explanations above, the author's idea emerged to create a system "Monitoring and Control of Android-Based Electricity Usage" so that it not only turns off the lights but also turns off the power terminal so that if you go far away you can control via Android, you no longer need to go home to see all the equipment is already dead or not. The research objectives of the design of the Android-based electricity monitoring and control application system are:

a) The system can monitor and control electricity remotely.
b) The system can work in realtime and precisely.
c) The system can work as a whole both on the system and control android applications.

2. Research methods

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

2.1 Systems Development Method

The method used in making the "Android-Based Electricity Monitoring and Control" system is the waterfall method, where the work is carried out in stages, there are 5 stages, namely determining the concept, requirements specification, design and system design, hardware and software implementation and finally unit testing. system[11][12].

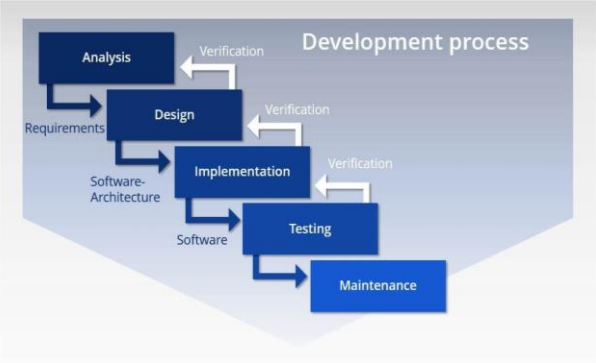

Fig 1. Waterfall method

At the concept stage, it is necessary to determine objectives including requirements specifications. Making this system aims to create a system that can control and view the electrical status of a building. In addition, the system that is made must also be easy to use by users, where Android will be the protocol for turning off and on electricity and seeing the status of electricity.

2.1.1 Analysis Phase

The specification stage is the stage for analyzing and describing what is needed to conduct research. The developer collects the data needed for research by analyzing and comparing several existing remote control systems, looking for data for the development of remote control systems via Android. Furthermore, the developer collects equipment to create a system such as Relay, ESP8266 NodeMCU Module, and ACS712 Module, so that they can build an electrical appliance control control system using Android[11].

2.1.2 Design Stage

The design and design stage must be made immediately after the concept and specification stages have been met. The design and system design stages are based on what equipment is needed to realize the system to be made, there are several stages of electrical equipment control, namely:

a) The user sends the command format from Android to the system.
b) Format of commands from Android that enter the system and will be processed if the command is correct by the system.
c) After the activity is successful (turning on, turning off, and knowing the status of the electricity), the system will provide a report according to what the user needs[11].

2.1.3 Implementation Stage

The next stage is the implementation of hardware and software, in hardware implementation, from the equipment that has been designed then implemented into a series of systems that can later control electrical equipment and after the hardware implementation is completed, then the implementation of the software aims to form a system. by providing a function which then becomes an internal command on the system, so that the system processes the input data command making it output data and can be implemented on electrical devices so that the protocal can execute user commands[11].

2.1.4 System Testing and Maintenance Phase

The last stage is the testing and maintenance stage of the system, this stage is carried out after all the previous stages have been completed. Testing is needed so that the system is not just an abstract program that cannot run at the time of system use. Tests are carried out whether the entire system is working properly and correctly as desired. The following are the test stages that will be carried out on the system:

a) Unit Testing
b) Testing System Work
c) ACS712 Current Sensor Testing
d) Testing Turning On and Off the Terminal

NodeMCU is an IoT platform which is opensource. The hardware consists of the ESP8266 System On Chip from the Espressif System ESP8266, as well as the firmware used, which uses the Lua scripting

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

programming language. The term NodeMCU by default actually refers to the firmware being used rather than the development kit hardware[13].

Fig2. NodeMCU ESP8266

3. Results and Discussion

3.1. Source Code

Below is the program code compiled using the Arduino IDE software and uploaded to NodeMCU ESP8266.

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>

#define relay1 D3
#define relay2 D4
#define relay3 D5
#define relay4 D6
#define relay5 D7

#define STASSID "Hasyim"
#define STAPSK "nasiuduk"

#define sensor_arus A0

const char * host = "192.168.56.1/script/cek_perangkat.php";
const int httpsPort = 443;


int ON = LOW;
int OFF = HIGH;
char command;
void setup () {
Serial.begin (9600);
pinMode (relay1, OUTPUT);
pinMode (relay2, OUTPUT);
pinMode (relay3, OUTPUT);
pinMode (relay4, OUTPUT);
pinMode (relay5, OUTPUT);
digitalWrite (relay1, OFF);
digitalWrite (relay2, OFF);
digitalWrite (relay3, OFF);
digitalWrite (relay4, OFF);
digitalWrite (relay5, OFF);

Serial.begin (115200);
 Serial.println ();
 Serial.print ("connecting to");
 Serial.println (ssid);
```


Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

```
 WiFi.mode (WIFI_STA);
 WiFi.begin (ssid, password);
 while (WiFi.status ()! = WL_CONNECTED) {
 delay (500);
 Serial.print (".");
 }
 Serial.println ("");
 Serial.println ("WiFi connected");
 Serial.println ("IP address:");
 Serial.println (WiFi.localIP ());

 // Use WiFiClientSecure class to create TLS connection
 WiFiClientSecure client;
 Serial.print ("connecting to");
 Serial.println (host);
 Serial.printf ("Application Connection '% s' \ n", fingerprint);
 if (! client.connect (host, httpsPort)) {
 Serial.println ("connection failed");
 return;
 }
}

void loop () {

HTTPClient http;

// Send request
http.useHTTP10 (true);
http.begin (host);
http.GET ();
// Parse response
DynamicJsonDocument doc (2048);
deserializeJson (doc, http.getStream ());
// Read values
Serial.println (doc ["data"]. As <char *> ());
if (doc ["data"]. as <char *> ()> 0) {
command = doc ["data"]. as <char *> ()
if (command == ('A')) {
digitalWrite (relay1, ON);
Serial.print ("A -");
Serial.println ("Lights on");
} else if (command == ('B')) {
digitalWrite (relay1, OFF);
Serial.print ("B -");
Serial.println ("Lights Off");
} else if (command == ('C')) {
digitalWrite (relay2, ON);
Serial.print ("C -");
Serial.println ("Fan On");
} else if (command == ('D')) {
digitalWrite (relay2, OFF);
Serial.print ("D -");
Serial.println ("Fan Dead");
} else if (command == ('E')) {
digitalWrite (relay3, ON);
Serial.print ("E -");
Serial.println ("Iron On");
} else if (command == ('F')) {
digitalWrite (relay3, OFF);
Serial.print ("F -");
Serial.println ("Iron Off");
} else if (command == ('G')) {
digitalWrite (relay4, ON);
Serial.print ("G -");
```

```
Serial.println ("TV On");
} else if (command == ('H')) {
digitalWrite (relay4, OFF);
Serial.print ("H -");
Serial.println ("TV Off");
} else {
Serial.println ("Please Enter Correct Keyword");
}
}
}
```

3.2. Testing of System Work Results on Android

This test is carried out on 4 switches contained in the monitoring and controlling tool of electricity consumption through the Android application. Testing is done on 4 switch buttons as an example of testing.

Table 1.
Test Results of Electrical Device Control Through Android

No.	Switch Button On Android	Device Condition	Status On Android System	Information
1	Button 1	Life	Life	It works
2	Button 2	Die	Die	It works
3	Button 3	Life	Life	It works
4	Button 4	Die	Die	It works

Information: In a working test this system has 4 switches or buttons that can be turned on and off. Through monitoring using Android, it can be seen that the switch status and status on Android are the same so that the user can clearly monitor what switches are on and off.

3.3. ACS712 Current Sensor Testing

This test is done to find out whether the sensor can detect current from one room to another. The terminals from one room to another are connected with copper cables from the relay with the ACS712 module which is connected to IP + and IP-, then the current flowing will be read by the system.

The test results from reading the current through the room to another room have been successfully carried out, namely the sensor can display the value of current usage in accordance with the parameters tested in Ampere (second) units. But if the sensor does not detect a current, the sensor will not read 0 Ampere but 0.10 Ampere because the ACS712 module readings are not linear when the reading is 0 Ampere, but when the module reads a current above 0.25 Ampere, the current rating will return to normal.

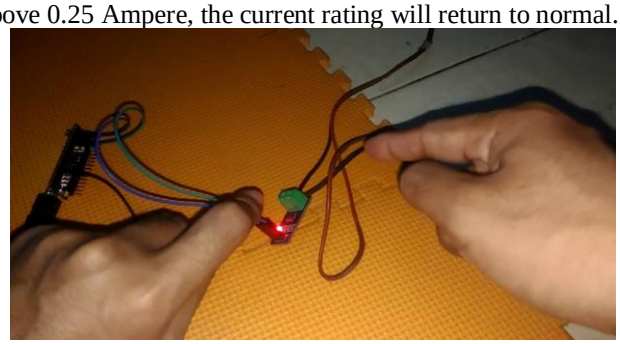

Fig 3. ACS712 Current Sensor Testing

3.4. Testing of On and Off Electrical Terminals

This test is done to find out whether the Electrical Terminal can be turned on and off via Android commands properly according to the design. The terminal is connected to a relay, so that the relay will be controlled by NodeMCU ESP8266 to disconnect or turn on the terminal, when the system asks to turn on or turn off the terminal the system must be able to run these commands properly.

The results of tests that have been carried out when the sensor value is 1 (HIGH), the terminal will turn on and when the sensor value is 0 (LOW) the terminal will turn off and when the led light on the relay turns on, then the terminal is off as well as when the relay led is off, then the terminal turns on .

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Fig 4. Testing of Turning On and Off Electrical Terminals Through Android Application Control

4. Conclusion

The conclusions that can be drawn from research on the Android-based Electricity Consumption Monitoring and Control system are as follows:
From the results of testing and analysis of the design of an Android-based electrical control system, the tool can turn on and off terminals connected to other electricity via Android commands. In addition to turning on and off the terminal, the tool can also monitor what switches are on and off so that the tool can help do remote monitoring and can operate the switch remotely. For the security system, if another Android device tries to enter the system, the system will notify you whether there is another device you want to register or not.

Thank-you note
The writers say especially to the lecturers who have been educating, guiding and providing knowledge and advice for me so that my journey to study at the National University has come this far. Do not forget the prayers of my parents and close friends who have supported me all this time to immediately complete this research.

5. References

[1] D. Risqiwati, "Rancang Bangun Sistem Monitoring Listrik Prabayar dengan Menggunakan Arduino Uno," *Kinetik*, vol. 1, no. 2, 2016.
[2] M. R. Fachri, I. D. Sara, and Y. Away, "Pemantauan Parameter Panel Surya Berbasis Arduino Secara Real Time," *J. Rekayasa Elektr.*, vol. 11, no. 4, p. 123, 2015.
[3] A. S. Romoadhon and D. R. Anamisa, "Sistem Kontrol Peralatan Listrik pada Smart Home Menggunakan Android," *Rekayasa*, vol. 10, no. 2, p. 116, 2017.
[4] S. Samsugi, A. Ardiansyah, and D. Kastutara, "Arduino dan Modul Wifi ESP8266 sebagai Media Kendali Jarak Jauh dengan antarmuka Berbasis Android," *J. Teknoinfo*, vol. 12, no. 1, p. 23, 2018.
[5] M. F. Wicaksono, "Implementasi Modul Wifi NodeMCU ESP8266 Untuk Smart Home," *J. Tek. Komput. Unikom*, vol. 6, no. 1, pp. 1–6, 2017.
[6] H. F. Imron, R. R. Isnanto, and E. D. Widianto, "Perancangan Sistem Kendali pada Alat Listrik Rumah Tangga Menggunakan Media Pesan Singkat (SMS)," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 3, p. 454, 2016.
[7] A. Giyartono and E. Kresnha, "Aplikasi Android Pengendali Lampu Rumah Berbasis Mikrokontroler Atmega328,"

Semin. Nas. Sains dan Teknol., no. November, pp. 1–9, 2015.
[8] D. E. Kurniawan, "Push Notification System Pada Prototype Kendali Listrik Rumah," *J. Comput. Eng. Syst. Sci.*, vol. 2, no. 1, pp. 89–92, 2017.
[9] A. M. Syafar, "Kendali Perangkat Listrik Dan Monitoring Daya Pada MCB Berbasis TCP/IP," *J. Instek*, vol. 1, no. 1, 2016.
[10] R. T. Hudan, Ivan Safril, "Rancang Bangun Sistem Monitoring Daya Listrik Pada Kamar Kos Berbasis Internet of Things (Iot)," *J. Tek. ELEKTRO*, vol. 08, no. 01, pp. 91–99, 2019.
[11] H. Nur, "Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan," *Gener. J.*, vol. 3, no. 1, p. 1, 2019.
[12] D. Handarly and J. Lianda, "Sistem Monitoring Daya Listrik Berbasis IoT (Internet of Thing)," *JEECAE (Journal Electr. Electron. Control. Automot. Eng.*, vol. 3, no. 2, pp. 205–208, 2018.
[13] M. Hayaty and A. R. Mutmainah, "IoT-Based Electricity Usage Monitoring and Controlling System Using Wemos and Blynk Application," *J. Teknol. dan Sist. Komput.*, vol. 7, no. 4, pp. 161–165, 2019.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.