Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 304−311
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5908

IoT-based Intelligent System for Laboratory Facility Control with Blynk

Puji Utami Rakhmawati*, Rizdania, Muhammad Nasikhul Muladi

Faculty of Science and Technology, Computer Science, University of PGRI Wiranegara, Pasuruan
Jl. Ki Hajar Dewantara No.27-29, Tembokrejo, Kec. Purworejo, Kota Pasuruan, Jawa Timur, Indonesia
Email: [1,*]tammyglory@gmail.com, [2]rizdania.uniwara@gmail.com, [3]nasikhul@gmail.com
Correspondence Author Email: tammyglory@gmail.com
Submitted: **17/09/2024;** Accepted: **09/10/2024;** Published: **19/10/2024**

Abstract−This study presents the development of a smart laboratory using the Internet of Things (IoT) intelligent system technology to control electronic devices, especially air conditioners (AC). The AC in the laboratory is a device that consumes very high electrical power. Human resources personnel who use the laboratory often forget to turn off the devices after use. The problem is high electricity costs due to forgetting to switch off devices. This study solves this problem with an IoT-based intelligent system to control laboratory facilities by utilizing the Blynk App. Blynk is an Android-based open-source software that can connect with ESP32, so control can be done easily anywhere and anytime. In this study, a prototype IoT control system was developed that allows users to control and monitor equipment via the internet automatically. This study uses ESP32 technology, where an Arduino device is used as the receiver connected to Wi-Fi, and a microcontroller is used as an automatic control. ON/OFF control using 220v 30A relays as many as four pieces. Test results from this study showed that the device provides a response time of 1 second to respond to commands, and the four relays installed successfully controlled the devices.

Keywords: Internet of Things; Intelligent System; Laboratory; Blynk

1. INTRODUCTION

The term Internet of Things (IoT) was first created by Kevin Ashton in 1999, a man with a supply management working background. IoT is expected to affect everything, from industry optimization to new product opportunities to escalating efficiency in human resources to support the performance of management and workers [1][2]. IoT represents the ability of network devices to sense and collect knowledge from the environment, and this knowledge is then shared via the internet, which can be processed and used for various applications such as smart homes, healthcare, storage warehouses, etc. [3]. IoT is also a communication paradigm that connects everyday objects to the internet. These objects are connected to microcontrollers and receivers to communicate and are configured with protocol procedures for communication interactions to occur without human intervention. This pattern is suitable for interactions with various devices such as AC, drones, alarms, robots, computer devices, network devices, and other electronic devices [4]. This concept is called Smart House, which connects electronic devices to the internet so that communication interactions are controlled through applications or software[5].

As universities grow yearly, issues such as energy efficiency and new management emerge. Managing resources in institutions poses a significant challenge. The inability to monitor and control unused devices that continue to consume power in the absence of people is also an important issue. Furthermore, the coordination problem with users is tedious because of the underestimation of resources. Another problem occurs in power management. It is hard to monitor all electronic devices, but energy will be wasted if this is left alone. Furthermore, there is no monitoring of the status of the device. To overcome these problems, IoT technology is one of the appropriate methods [6].

This research initiated the building of an IoT Smart System implemented in a computer laboratory. This system aims to activate and deactivate devices with system control and IoT cloud application tools anywhere and anytime in the palm of your hand. This system also monitors the device's status to see whether it is active. So that the expected resource management is achieved [7]. IoT is believed to increase remote monitoring, energy efficiency, and control of productivity and physical assets through various applications, such as home security to monitoring conditions on the industrial floor. The general description of IoT is the involvement of RFID, WIFI, and Bluetooth technologies that have helped develop IoT [8].

Previous research about the Internet of Things, such as the Prototype of Remote Control of Lights and AC with an Internet Network Using the Telegram Application Based on Nodemcu Esp8266 using NodeMCU ESP8266 and a control system using Telegram. The test results show that all devices in the Prototype function properly. Although there is a response delay of about 1-3 seconds on devices controlled via the Website due to unstable internet connection and electric current, the results of device sensor detection remain accurate [9]. A prototype is created by utilizing Blynk Cloud Technology in a Home Light Socket Control System Based on an Android Application using NodeMCU ESP8266 and the Blynk Android application as a control or monitoring tool. The test results of this prototype of the whole tool work well. The tools controlled via the website experience delays caused by unstable internet and electricity networks.

Meanwhile, the detection results of the temperature sensor device can be sent to the database server accordingly [10]—implementation of IoT (Internet of Things) technology on smart homes based on the ESP8266 microcontroller. This study concludes that IoT (Internet Of Think) technology is very suitable because it can cover

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 304−311
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5908

a wide area. Even though the homeowner is not in place, this can be accessed anytime and anywhere, making it easier for users [11]. The research designs and develops an intelligent and multipurpose IoT embedded system device using an ESP32 microcontroller [12]. With the increasing popularity of IoT devices, the ESP32 microcontroller offers an efficient and cost-effective solution for a wide range of embedded system applications [13].

2. RESEARCH METHODOLOGY

2.1 ESP32

The ESP32 microcontroller is an integrated SoC (System on Chip) microcontroller with Bluetooth version 4.2, Wi-Fi 802.11 b/g/n, and various peripherals. ESP32 is a reasonably complete chip with a processor, storage, and access to GPIO (General Purpose Input Output). ESP32 can be used to replace Arduino circuits. ESP32 can support direct WI-FI connection (Agus Wagyana, 2019). The specifications of the ESP32 are as follows: This board has two versions, namely 30 GPIO and 36 GPIO. Both have the same function, but the 30 GPIO version was chosen because it has two GND pins. All pins are labelled on the top of the board so they are easy to recognize. This board has a USB to UART interface that is easy to program with application development programs such as Arduino IDE. Board resources can be provided via a micro USB connector [14].

This study used a four-channel relay board type 220v 30A, also called an IoT kit. This research installation is located in the Smart Laboratory of PGRI Wiranegara University; in terms of infrastructure, this laboratory already has a Wi-Fi connection to support this development. Before using these four 30A relays, researchers used ESP8266 and eight 10A relays. However, in reality, the 10A relay is not recommended by practitioners because the power of the controlled device is also very high. Likewise with ESP8266, by considering along the way, a more sophisticated microcontroller was found, namely ESP32.

These IoT kits will be made into four packages, namely for two AC units with a power of 1900 watts, 2 PC Power Sources, and a network of around 1800 watts. These four packages will later be connected to ESP32 using a power cable. The circuit design can be seen in Figure 1. In Figure 1, it is explained that four relays are connected to the ESP. The four relays are Relay 1, connecting to the PC1 terminal; Relay 2, connecting to the PC2 terminal; Relay 3, connecting to Air Conditioner (AC) 1; and Relay 4, connecting to Air Conditioner (AC) 2. At the same time, the AC Power pin from each ESP connects to each power provision. The load is used as an electrical load breaker according to its instructions. Keempat relay tersebut divirtualkan dengan tombol, sehingga pada rangkaian ini juga terdapat 4 tombol yang mana tombol ini akan dijelaskan pada Aplikasi Blynk.

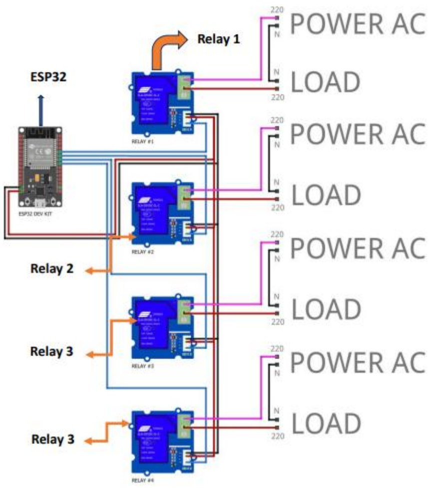

Figure 1. ESP32 4 Relay Circuit

2.2 Arduino IDE

Arduino IDE (Integrated Development Environment) software is a built-in Arduino application that can control open-source single-board micros built to facilitate users. Arduino software can create, open, compile, and upload programs to the Arduino board. The Arduino IDE application is designed to make it easier for users to develop various applications. Arduino IDE uses the C/C++ language for programming with complete functions so that it can be reviewed by new users [15].

By utilizing the IoT cloud, researchers still need to configure Arduino. In other words, the Blynk IoT cloud application uses a front-end configuration. So that the back-end configuration is still carried out so that Blynk and ESP32 can communicate successfully.

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 304−311
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5908

Figure 2 shows some source code from the ESP32 program in the Arduino IDE. The most important requirements for building the program are the code from Blynk and the registered Wi-Fi. In contrast, the relay and button pins correspond to the relays needed. The four relays are virtualised with buttons, so in this circuit there are also 4 buttons which will be explained in the Blynk Application.

Figure 2. Application Source Code on Arduino IDE

2.3 Relay

Relay is an electrical component. It works based on the electromagnetic field induction principle. If a conductor is flowed by an electric current, then a magnetic field arises around the conductor. The magnetic field produced by the electric current is then induced into the metal of ferromagnetic. The first inventor of the Relay was Joseph Henry in 1835 [16].

The cooperation principle with the magnetic contractor is the same as the magnetism produced by the coil when a power source is given to the coil. Based on the incoming power source, the Relay is split into two types: DC and AC. The amount of DC voltage entering the relay coil varies according to the size listed on the relay body, including relays with a voltage of 6 Volts, 12 Volts, 24 Volts, and 48 Volts, while for AC voltage of 220 Volts.

An example of the shape of the relay is shown in Figure 3. The relay is a 30A relay used by researchers. The Relay contains a coil and a contact; the coil is a wire reel that receives electric current, while the contact is a type of switch whose movement depends on the presence or absence of electric current in the coil. There are two types of contacts. The first is Normally Open (initial condition before being activated is Open), and the second is Normally Closed (initial condition before being activated is Closed). In simple terms, the following is the working principle of a relay. When the coil receives electricity (energized), an electromagnetic force will arise, which will pull the spring-loaded armature, and the contact will close.

Figure 3. Relay 30A

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 304−311
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5908

2.3 Blynk

Blynk is an Android and iOS platform operating system that can power ESP8266 modules, Raspberry Pi, Arduino modules, and other devices via the internet [17]. Blynk is connected to Arduino with a token and Wi-Fi configuration to run communication. Blynk can communicate if the ESP32 or Arduino is given the required configuration or source code. If the source code is not provided, then Blynk cannot be used. Arduino is configured with C language to translate communication commands with the receiver.

At the beginning of its appearance, Blynk came with an application called Blynk Legacy. Then, in May 2021, the developer officially announced that the application was no longer being developed. Instead, the application developer launched Blynk 2.0 or Blynk IOT [18].

An example of a Blynk App is shown in Figure 6. The Blynk application allows the developer to make a project interface with various components of input-output that support sending and receiving data and presenting data according to the selected components. Data representation can be in the form of graphs or visual numbers. Blynk is a cloud-based back-end service facility responsible for managing communication between smartphone applications and the hardware environment. Blynk Library can be used to help develop source code [19].

2.4 Prototype Model Development

In general, the methodology used in this study is the method of prototype. The prototype in this study is applied because the technique for describing a structured system has several processes that must be passed in its creation. The prototyping model is one of the software development methods, and it is the initial version of the system. Implementing this method allows users to interact at every stage of information system development [20]. The study utilizes the IoT cloud application, namely the Blink IoT application, to develop this research. Blynk was chosen because the facilities from Blynk meet the needs of this research.

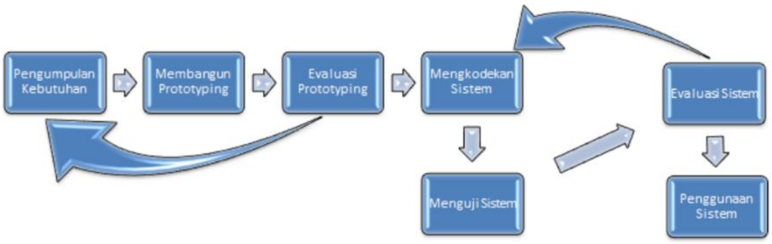

Figure 4. Prototyping Method [21]

Figure 4 explains the sequence of the prototype development model used in this study. This prototype development model is used by researchers because of the advantages of the prototype model which is easy to adjust to users if there is an inappropriate development. According to Figure 4 [22], the stages in the prototyping process are as follows:

1. Requirement Gathering
   In this stage, customers and developers will collaborate to formulate the overall format of the software, determine all the requirements, and determine the system to be developed. The application of this research is to determine the materials, tools, and software needed in this development and the design of the system to be developed.
2. Prototype Development
   Prototype development involves creating a temporary design that is focused on the initial design. At this stage, what is input, process, and output is declared. In this activity, the researcher assembles esp32 and relays and connects it to the power source cable, then integrates it with the Blynk and connects it to the device.
3. Prototyping Evaluation
   At this stage, the researcher evaluates the prototype; the obstacle found is the use of Wi-Fi, which must be stable. So, the researcher created a special SSID for this study.
4. System Codification
   The approved prototyping will be implemented into the appropriate programming code with the Arduino IDE platform.
5. System Testing
   After the software is developed, testing must be carried out before use. This research uses a black box function test.
6. System Evaluation
   The researcher will evaluate the system to ensure it meets their expectations. The following process will continue if the system has met expectations. However, if deficiencies exist, the system will be tested and accepted again for repair and retesting.
7. System Use
   After the software has been tested and is successful, it is ready to be used.

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 304−311
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5908

3. RESULT AND DISCUSSION

3.1 Designing Prototype

At this stage, the design and making of prototypes are carried out based on user information. Based on the results of the analysis, a block diagram design is made; this block diagram is made to plan the hardware according to the specifications and working methods of the system to be made so that it is expected to be able to save time, costs and energy [23].

Figure 5 is a block diagram of the development of this research. The input of this development is Blynk Trigger. This Blynk trigger has 4 buttons shown in Figure 7. The Blynk trigger button will forward to the ESP32. Then ESP32 forwards the command according to the controlled button to the relay. When the relay gets the command, the relay will forward to the terminal which is the output. So that these four terminals will be able to ON/OFF according to the command from Blynk Trigger.

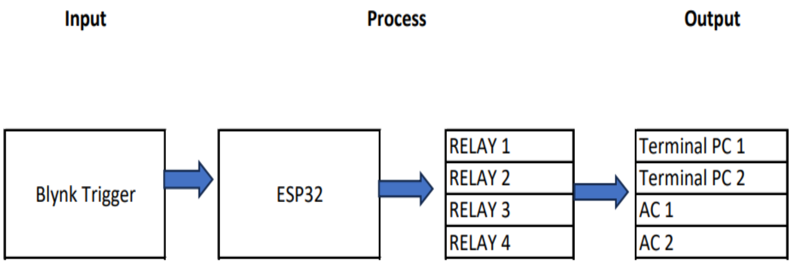

Figure 5. Block Diagram of Monitoring Control System

The steps in development using this prototype model are:

1. Requirement Gathering
   The needs in this study are:
   a. ESP32 Development Board
   b. Relay Module 5V 1-Way 30A
   c. Black Box 18×11×6
   d. Adapter 5V 3A Switching Adapter 15W
   e. Toggle Switch MTS-102 125V/3A 250V ON ON Switch 3-Feet
   f. Power Cable 2×1.55mm
2. Prototype Development
   At this stage, researchers assembled the ESP32 by connecting relay components, toggle switches, and power cables to become an ESP Dev Kit. Figure 5 shows these stages become inputs, processes, and outputs. ESP32 will detect input in the form of commands from Blynk as input. ESP32, as a receiver, is then sent to the relays, in this case as a process, and the output of this research is the on/off command, which is carried out.
3. Prototyping Evaluation
   After becoming an ESP Dev Kit, the next step is to test the power on-off circuit with the indicator on the Relay.
4. System Codification
   At this stage, programming is done on the Arduino IDE to connect ESP, Wi-Fi, and Blynk. The needs at this stage are stable Wi-Fi and code from the Blynk system, where Blynk is open source so that the code can be found easily.
5. System Testing
   The prototype is tested using a black box function test at this stage. That is, the four buttons have functioned as expected.
6. System Evaluation
   At this stage, the researcher improved the Wi-Fi because it was found that the Wi-Fi was unstable, so a special SSID was made for this research.
7. System Use
   At this stage, the prototype can be used following expectations.

This ESP32 requires stable Wi-Fi to establish communication between Blynk and ESP32. In this study, the ideal specification was found to be 30V, which previously used 10V; it turned out that there was heat on the installed cables due to the device's reasonably high load. This study also found that the appropriate NODEMCU or the best technology today is ESP32, which researchers also tried with ESP8266, but communication was not as fast as ESP32. When ESP32 receives a command, it will instruct the Relay that is ordered to turn on the device. This research has four types of outputs: Terminal PC 1, which covers the power source from 10 computers, and Terminal PC 2 and Air Conditioner; AC1 and AC2, which have a power of 1900 watts.

Figure 6 shows the circuits worked on by researchers, namely the ESP32 circuit with 4 relays with each relay with a capacity of 30A and the ESP8266 circuit with 8 channel relays each with a capacity of 10A.

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 304−311
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5908

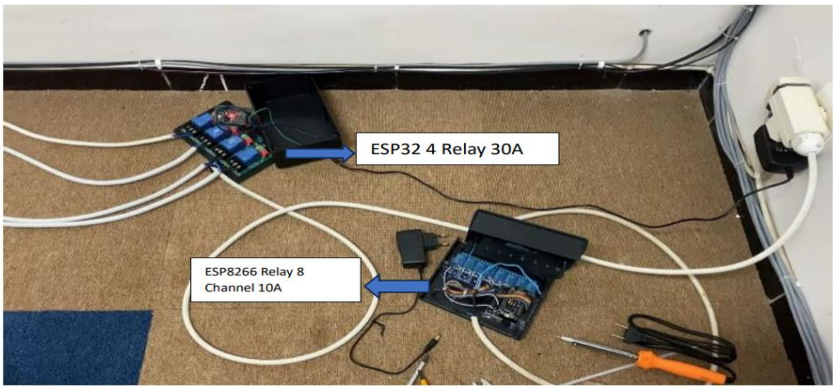

Figure 6. ESP32 4 relay circuit and ESP8266 8 relay circuit

3.2 Implementation

From the components above, the circuit shown in Figure 7 is produced. In Figure 7, it explains the ESP32 circuit with 4 relays with each relay having a capacity of 30A. ESP32 is a microcontroller with a Wi-Fi feature utilized in this study. ESP32 is installed with Arduino IDE, configuring with the source code shown in Figure 2. The most essential thing in this configuration is Wi-Fi and token, where both features play a role in communication on the ESP32. Parallel to the configuration, Blynk is also configured with its user interface; using Blynk is very easy, but paying attention to the naming of variables is also necessary. The suitability of the naming of variables and source code is also no less important in the success of this device.

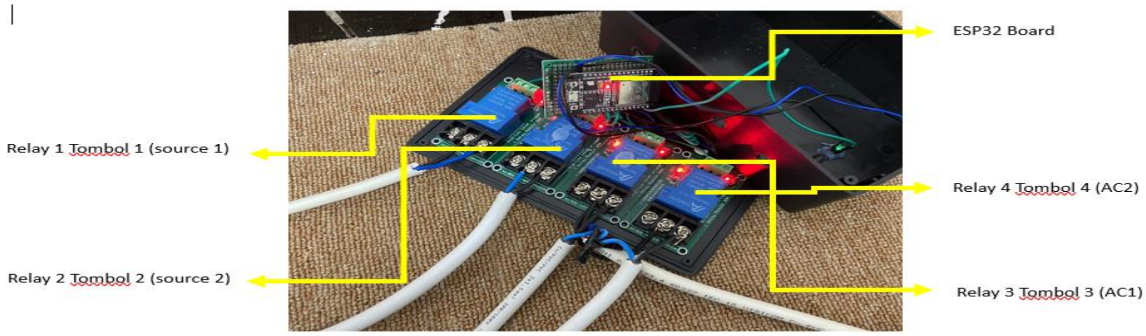

Figure 7. The Results of IoT Tool Design

This circuit works because Relay 1 is connected to the power source for PC laboratory 1. The Relay is used to connect or disconnect the flow of electricity to the load (PC laboratory1). The cable from the 220v source is connected to the relay leg "COM", and the cable to the Load (PC laboratory1) is connected to the "NO" leg. When the application is selected "ON", the ESP will send a signal to the Relay to connect the "COM" and "NO" legs; finally, the electric current flows from the source to the load. And vice versa. This procedure is also done on Relay 2, Relay 3 and Relay 4.

The design of Blynk can be seen in Figure 8. Making a user interface on Blynk is easy with drag and drop, so it can be easily learned. Nevertheless, how Blynk can communicate with this ESP32 requires a programming configuration, as shown in Figure 2. Stable Wi-Fi must also support Blynk's communication needs; previous researchers have experienced Wi-Fi problems, so if stable Wi-Fi is found, Blynk communicates quickly and responds quickly. In Figure 6, the buttons are designed based on the relays. The ON/OFF button is then instructed to signal the device.

Figure 8. Blynk Layout Design

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 304−311
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5908

How the Blynk App works is :

1. Hardware Setup : Firstly, hardware such as ESP32 is configured with Blynk code by Arduino IDE that connects it to the Blynk server.
2. Creating Dashboard : Users then create a dashboard in the Blynk app by adding widgets as per their needs.
3. Connecting : After that, the hardware and mobile app are connected through the Blynk server, and users can monitor and control their hardware remotely.

3.3 Hardware of IoT and Blynk Test Result

From the design of the IoT hardware, we can communicate smoothly, stably, and responsively. This test uses a black box function test. This test examines the response to the ESP32 and Relays when Blynk is instructed, as shown in Table 1. Table 1 describes the 4 relays with 4 buttons functioning as expected with an average repson speed at 1 second. The feature test results in table 1 briefly explain that :

1. Button 1 successfully connected ON/OFF to Relay 1 to control computer terminal 1 with a response time from Blynk command to ESP-Relay of 1 second.
2. Button 2 successfully connected ON/OFF to Relay 2 to control computer terminal 2 with a response time from Blynk command to ESP-Relay of 1 second.
3. Button 3 successfully connected ON/OFF to Relay 3 to control AC 1 terminal 3 with a response time from Blynk command to ESP-Relay of 1 second.
4. Button 4 successfully connected ON/OFF to Relay 4 to control AC 2terminal 4 with a response time from Blynk command to ESP-Relay of 1 second.

Table 1. Hardware and Blynk Test Result

Name Of Relay	ON	OFF	Time (second)
Relay 1 (Terminal PC 1)	success	success	1
Relay 2 (Terminal PC 2)	success	success	1
Relay 3 (AC 1)	success	success	1
Relay 4 (AC 2)	success	success	1

Figure 8 shows the log of the successful connection of the Blynk App with ESP32. This makes it easy to monitor whether the ESP32 and Blynk App are connected or not. When the ESP32 is successfully connected to the Blynk cloud, the app logs this as "Online," indicating that the device is communicating properly. If it's "Offline," it means there's no connection between the ESP32 and the app, possibly due to network issues, power problems, or a disconnection from the Blynk server.

Figure 8. Blynk and ESP32 Connection

4. CONCLUSION

This IoT-based intelligent system for laboratory facility control can be implemented into various electronic devices that need to be controlled. The weakness in the Internet of Things is Wi-Fi; if the Wi-Fi is stable, the device can respond quickly and communicate appropriately; if it is unstable, it is slow to respond and not connected. Trials were conducted on 4 Relays that successfully controlled the specified device. Control of the device can be done

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 304−311
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.5908

anytime and anywhere by a smartphone installed with Blynk with a condition account ESP32 to get stable Wi-Fi and the specified Wi-Fi. So, the speed at which ESP32 responds to triggers is very fast, and there is no delay.

ACKNOWLEDGMENT

We would like to thank DP2M Ristekdikti and all those who have supported the implementation of this research. The authors' research was funded by Hibah Penelitian Dosen Pemula (PDP) 2024 from Higher Education DP2M Ristekdikti.

REFERENCES

[1] B. K. Tripathy and J. Anuradha, Internet of Things (IoT): Technologies , Applications , Challenges and Solutions, no. October 2017. CRC Press, Taylor & Francis Group, 2018.
[2] A. Rayes and S. Salam, Internet of Things From Hype to Reality - The Road to Digitization. Springer, 2019. doi: 10.1007/978-3-319-99516-8.
[3] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the Internet of Things (IoT)," Italia, 2015.
[4] M. Poongothai, P. M. Subramanian, and A. Rajeswari, "Design and Implementation of IoT Based Smart Laboratory," in 2018 5th International Conference on Industrial Engineering and Applications, ICIEA 2018, IEEE, 2018, pp. 169–173. doi: 10.1109/IEA.2018.8387090.
[5] A. A. Da Conceic'ao et al., "Internet of Things Environment Automation: A Smart Lab Practical Approach," in Proceedings - 2022 2nd International Conference on Information Technology and Education, ICIT and E 2022, IEEE, 2022, pp. 322–327. doi: 10.1109/ICITE54466.2022.9759899.
[6] M. Cherian and H. K. P, "Implementation of a Secure and Smart Lab with Wireless Sensor Network," Int. J. Sci. Res., vol. 3, no. 6, pp. 2012–2015, 2014.
[7] V. S. K. N and K. S. M, "Home Automation Using Internet of Things," Lect. Notes Networks Syst., vol. 39, pp. 293–301, 2019, doi: 10.1007/978-981-13-0277-0_24.
[8] H. Tahir, A. Kanwer, and M. Junaid, "Internet of Things (IoT): An Overview of Applications and Security Issues Regarding Implementation," Int. J. Multidiscip. Sci. Eng., vol. 7, no. 1, pp. 14–22, 2016, [Online]. Available: http://www.ijmse.org/Volume7/Issue1/paper3.pdf
[9] A. M. Ibrahim and D. Setiyadi, "Prototype Pengendalian Lampu Dan Ac Jarak Jauh Dengan Jaringan Internet Menggunakan Aplikasi Telegram Berbasis Nodemcu Esp8266," Infotech J. Technol. Inf., vol. 7, no. 1, pp. 27–34, 2021, doi: 10.37365/jti.v7i1.103.
[10] F. S. Permana, M. N. S. Putro, and R. Suwartika, "Jurnal Teknik Informatika Atmaluhur," J. Tek. Inform. Atmaluhur, vol. 6, no. 1, p. 40, 2022.
[11] B. Setiawan and Windarto, "Prototype Internet of Things for Smart Home berbasis Web Menggunakan Modul ESP8266," in Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI), 2022, pp. 1103–1111. [Online]. Available: https://senafti.budiluhur.ac.id/index.php/senafti/article/view/284
[12] A. B. Lasera and I. H. Wahyudi, "Pengembangan Prototipe Sistem Pengontrolan Daya Listrik berbasis IoT ESP32 pada Smart Home System," Elinvo (Electronics, Informatics, Vocat. Educ., vol. 5, no. 2, pp. 112–120, 2020, doi: 10.21831/elinvo.v5i2.34261.
[13] S. Dey and T. Bera, "Design and Development of a Smart and Multipurpose IoT Embedded System Device Using ESP32 Microcontroller," in IEEE International Conference on Electrical, Electronics, Communication and Computers, ELEXCOM 2023, IEEE, 2023, pp. 1–6. doi: 10.1109/ELEXCOM58812.2023.10370327.
[14] A. Wagyana and Rahmat, "Prototipe Modul Praktik untuk Pengembangan Aplikasi Internet of Things (IoT)," Setrum Sist. Kendali-Tenaga-elektronika-telekomunikasi-komputer, vol. 8, no. 2, pp. 238–247, 2019, doi: 10.36055/setrum.v8i2.6561.
[15] Junaidi and Y. D. Prabowo, Project Sistem Kendali Elektronik Berbasis Arduino. Bandar Lampung: Aura, 2018.
[16] H. Santoso, Pengertian, Fungsi, dan Cara Kerja Relay. Situbondo, Accessed: Sep. 01, 2024. [Online]. Available: http://www.elangsakti.com/2013/03/pengertian-fungsi-prinsip-dan-cara.html
[17] P. Bayborodin, I. Liashchuk, and D. Dumanskiy, "Blynk." Accessed: Sep. 01, 2024. [Online]. Available: https://blynk.io/
[18] A. Herlina, M. I. Syahbana, M. A. Gunawan, and M. M. Rizqi, "Sistem Kendali Lampu Berbasis Iot Menggunakan Aplikasi Blynk 2.0 Dengan Modul Nodemcu Esp8266," INSANtek, vol. 3, no. 2, pp. 61–66, 2022, doi: 10.31294/instk.v3i2.1532.
[19] T. Juwariyah, S. Prayitno, and A. Mardhiyya, "Perancangan Sistem Deteksi Dini Pencegah Kebakaran Rumah Brbasis Esp8266 dan Blynk," J. Transistor Elektro dan Inform. (TRANSISTOR EI), vol. 3, no. 2, pp. 120–126, 2018.
[20] D. Purnomo, "Model Prototyping Pada Pengembangan Sistem Informasi," J I M P - J. Inform. Merdeka Pasuruan, vol. 2, no. 2, pp. 54–61, 2017, doi: 10.37438/jimp.v2i2.67.
[21] P. Yoko, R. Adwiya, and W. Nugraha, "Penerapan Metode Prototype dalam Perancangan Aplikasi SIPINJAM Berbasis Website pada Credit Union Canaga Antutn," J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi), vol. 7, no. 3, p. 212, 2019, doi: 10.24843/jim.2019.v07.i03.p05.
[22] U. R. Khotimah, Y. Yudhistira, and F. Nabyla, "Implementasi Sistem Informasi Pengelolaan Wakaf Menggunakan Metode Prototyping Pada Yayasan," J. Sist. Inf. dan Teknol. Perad., vol. 3, no. 2, pp. 38–45, 2022.
[23] A. Maulana Ibrahim, A. Solikhin, M. Karya Mandiri, and P. Studi Manajemen Informatika Politeknik Mitra Karya Mandiri, "Sistem Kontrol dan Monitoring Berbasis IoT pada Lampu dan AC di Laboratorium Komputer Politeknik Mitra Karya Mandiri," J. Univ. Muhammadiyah Jakarta, vol. 13, no. 2, pp. 87–91, 2023, [Online]. Available: https://jurnal.umj.ac.id/index.php/just-it/index