



Thu Dau Mot University  
Journal of Science

ISSN 2615 - 9635

journal homepage: [ejs.tdmu.edu.vn](http://ejs.tdmu.edu.vn)



## The demonstration device for remote light control via the Internet by using MQTT protocol and Dual-Chip ESP32

by **Gia Thinh Vo** (Thu Dau Mot University)

**Article Info:** Received June 13th, 2022, Accepted Aug. 9th, 2022, Available online Sep. 15th, 2022

Corresponding author: [thinhvg@tdmu.edu.vn](mailto:thinhvg@tdmu.edu.vn)

<https://doi.org/10.37550/tdmu.EJS/2022.03.313>

### ABSTRACT

*In this article, the primary purpose is to design and implement a remote light control example that makes use of the Arduino IDE platform for programming, the MQTT protocol, and an ESP32 microcontroller, which has a dual-core CPU as well as Wi-Fi and Bluetooth capabilities. This aspect is related to the Internet of Things (IoT), which allows any fundamental device to be linked and become a smart device via the application of the Internet. As a result, these devices can be operated from a significantly long-distance, eliminating the need to physically interact with the switches to turn them on or off. The first part of the paper is a discussion of the fundamental theory that is provided to examine the potential of remote control methods, followed by a short overview of the Arduino integrated development environment (IDE), MQTT communication protocol, and PWM technique to regulate the brightness of lights. The second section is devoted to hands-on work, such as setting up an environment connection over the Internet, compiling and designing a printed circuit board (PCB).*

**Keywords:** *remote control, wireless communications, MQTT protocol, ESP 32 dual-core, PWM method*

### 1. Introduction

One of the primary arguments against wasting energy is the ability to access and operate equipment from several distant places. A good example of this is when individuals leave home and realize they forgot to turn off certain gadgets such as lights, televisions, or air

conditioners, they must come home and turn them off. This results in energy loss and wastes their time. Meanwhile, using a variety of communication methods such as Wi-Fi, Bluetooth, Radio-Frequency, or a sensor to connect to these devices and control them, or another way, with the continued growth of the Internet, a human being can remote control these devices using a web server or an android application. These technologies have significant drawbacks, in a typical residence, the Wi-Fi range is between dozen and hundred meters (Bhooanusas & Sou, 2021). They can transmit up to 30 meters (Błaszkiwicz et al., 2021) and are often used in long-range applications. In general, Bluetooth devices coupled to a chip or nodemcu have a range of around 10 meters ("Understanding Bluetooth Range | Bluetooth® Technology Website", 2022). Despite its low-power design, Bluetooth-enabled devices still use a great deal of energy. Bluetooth communication lines are often used to send data up to 100 meters (Finlay, 2016). Long-range techniques may suffer from signal loss or interference. Wi-Fi has the largest range of any of the Bluetooth, Wi-Fi and Radio-frequency technologies. A remote control of equipment from anywhere in the globe is crucial. Remotely administering devices through the Internet network addresses this problem.

Analyze possibilities using these approaches, which are wireless controllers and have the benefit of not needing physical switches to turn them on or off. ESP32 is Espressif Systems' upgraded version of ESP8266 featuring dual-chip Wi-Fi and Bluetooth wireless capabilities. It has antenna switches, RF balun, power amp, low-noise receive amp, filters, and power management modules. Fine resolution clock gating, multiple power modes, and dynamic power scaling make ESP32 appropriate for mobile devices, wearables, and IoT. ESP32 supports several programming languages. It can be programmed using Arduino IDE, Espressif IDF, Micropython, and JavaScript ("The Internet of Things with ESP32", 2022). This research will concentrate on programming the ESP32 using the Arduino IDE, which features a built-in editor, compiler, programmer, boot loader, library packages, and easy integration ("About Arduino", 2022).

Evidently, the HTTP protocol has been around for a long time and has been used for data transport in heterogeneous systems throughout the history of the Internet (Patel & Doshi, 2020). A detailed comparison of the protocols MQTT, CoAP, HTTP, XMPP, and others is covered in (Patel & Doshi, 2020), (Prada et al., 2016), (Naik, 2017). Message Queuing Telemetry Transport (MQTT) is a lightweight or simple messaging protocol that is used for Internet of Things (IoT) devices with little bandwidth. The MQTT protocol plays a critical part in the IoT because of its high stability and ability to be utilized in unreliable networks. Due to the limited bandwidth, this problem results in a high latency environment, which is why this protocol is used in a broad variety of Machine to Machine (M2M) communication, and it is also the advantage of MQTT compared with HTTP protocol (Baig et al., 2021).

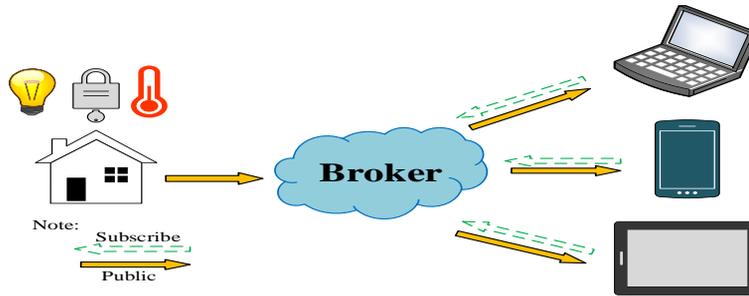
This control system takes some of the concerns discussed above into consideration when devices are controlled from a distance beyond the operational range of (Bhooanusas & Sou, 2021), (Błaszkiwicz et al., 2021), ("Understanding Bluetooth Range | Bluetooth® Technology Website", 2022), (Finlay, 2016) control methods, such as through Wi-Fi, Bluetooth, or radio-frequency. This study is inspired by the aforementioned investigations and focuses on remote lighting control from anywhere with an Internet connection. This remote control system uses ESP32 dual chips and MQTT protocol instead of HTTP, which is renowned in Internet data transfer history. It has grown simpler and less sophisticated than the HTTP protocol, making it easier to construct a control system without developing a new application. When adjusting the illumination time of LEDs or OLEDs by altering the voltage signal sequence, pulse width modulation (PWM) is used to regulate brightness (Akagi & Kondo, 2010), (Holtz, 1992), (Holmes & Lipo, 2003). In this scenario, the PWM approach is utilized to alter the light bulb's brightness by sending signals from the clients to the MQTT broker.

## **2. Materials and Methods**

### ***2.1. MQTT Protocol***

MQTT is used in a variety of fields to meet demand and benefit human life, including smart health-care systems: smart healthcare systems use sensors like EEG and ECG to sense and deliver data to subscribers via MQTT (Ding Yi et al., 2016), (Yang et al., 2016). Smart homes: temperature, light, motion, gas, and camera sensors are used in smart home systems; micro-programmers capture data and send it to the homeowner using MQTT protocol (Froiz et al., 2018), (Jamborsalamati et al., 2019). Smart industries: machine-to-machine communication is vital in smart industry. Material optimization, decision making delay reduction, and identification of required objects are some of today's industry's critical needs. Due to its tiny header size and straightforward communication mechanism, MQTT can be vital in the smart industrial sector (Katsikeas et al., 2017), (Patel & Doshi, 2019). Smart parking: RFID is an important part of a smart parking system. RFID does not require a scanner in front of it. RFID tags and readers are vital. RFID reader reads tag and publishes it to cloud via MQTT. The smart parking system user can simply recognize if the location is empty (Katsikeas et al., 2017), (Patel & Doshi, 2019). Smart weather stations: sensors used in smart weather systems include wind speed, solar radiation, air pressure, temperature, and humidity. The MQTT protocol can also be utilized in smart weather systems (Dhar & Gupta, 2016), (Pooja et al., 2017).

Figure 1 depicts the information on the development of the MQTT protocol, which is comprised of three basic elements: the MQTT server, also known as the Broker, the Publish and Subscribe functions. The Public and Subscribe channels are two types of channels via which a device may transmit (publish) a message on a certain subject or receive a message from a subscribed channel on a specific topic.



**Figure 1.** Block diagram of MQTT protocol structure

A system that employs the MQTT protocol contains a number of node stations (also known as MQTT clients or clients) that are linked to the MQTT server (Broker). This implies that the MQTT Clients do not communicate directly with one another; instead, all data packets are routed via the MQTT Broker. Subscribe is the term used to describe the process by which a client registers several topic channels, such as topic/control/light1 or topic/control/light2. When a user registers a channel, the client will receive data from any node station that sends data to that channel. This procedure is known as Publish.

### 2.2. MQTT – Broker

MQTT-Broker provides various functions, including receiving, filtering, assessing whether or not customers are interested in it, and publishing messages to client devices that have subscribed to it. As a result, it might be considered the most crucial feature of the MQTT protocol (CISCO, IOTWF, 2014). There are several additional free MQTT cloud brokers accessible on the Internet, but two of the most well-known are cloudmqtt.com and thingstud.io, both of which have a large number of customers throughout the globe. Figure 2 depicts extensive information about instances that are used to establish a broker from a free instance version with a restricted connection to a purchased instance version for industrial applications.

|   |   |  |
|---|---|--|
|    | <p><b>Keen Koala</b></p> <ul style="list-style-type: none"> <li>o Up to 100 connections</li> <li>o No artificial limitations</li> <li>o Support by e-mail</li> </ul>                    | <p><b>\$ 19</b><br/>PER MONTH</p> <p>Get Now</p> |
| <p><b>Shared instances</b><br/>For development or small hobby projects. Not recommended for production due to variable performance.</p> |   |  |
|    | <p><b>Humble Hedgehog</b></p> <ul style="list-style-type: none"> <li>o 25 users/acl rules/connections</li> <li>o 20 Kbit/s</li> <li>o 3 bridges</li> <li>o Support by e-mail</li> </ul> | <p><b>\$ 5</b><br/>PER MONTH</p> <p>Get Now</p>  |
|    | <p><b>Cute Cat</b></p> <ul style="list-style-type: none"> <li>o 5 users/acl rules/connections</li> <li>o 10 Kbit/s</li> </ul>   | <p><b>FREE</b></p> <p>Get Now</p>                |

**Figure 1.** The instances of cloudmqtt.com

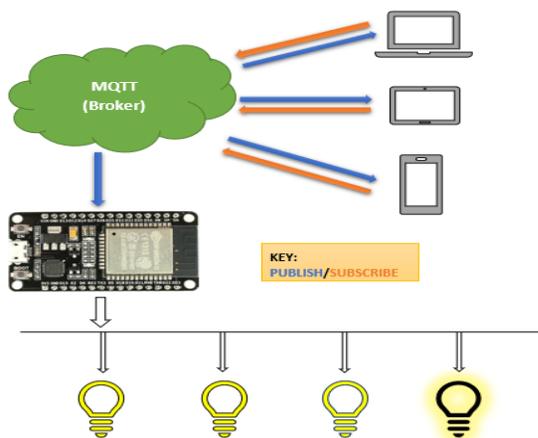
In this project, a free Cute Cat instance on cloudmqtt.com is used to create MQTT-Broker. The Send message in websocket describes the detail that the user can use the client devices to choose which topic and send out a message straightforwardly from the WebSocket on the broker of cloudmqtt.com to control the LED's switching states, and the Send message properly includes the topic's name generated to publish and the source code from Arduino IDE platform.

### 2.3. PWM method for light brightness control

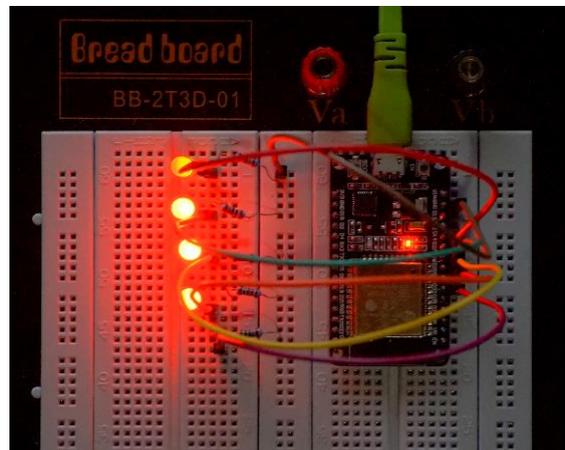
PWM is the most generally used approach for altering the duty cycle, which illustrates the amount of time during which a signal is in the state of being on (high) or being off (low), the percentage of the total time necessary to complete one period of operation (Akagi & Kondo, 2010), (Holtz, 1992), (Holmes & Lipo, 2003). The functioning of items, such as motor drives, may be regulated by applying the characteristics of the PWM approach. In this demonstration, it is used to regulate the brightness of the lights.

## 3. Results and Discussion

### 3.1. Operating model of the switching light states



**Figure 2.** The block diagram of remote lights control by using MQTT protocol and the ESP32 dual-chip



**Figure 3.** LEDs demonstration version

The essential function of the remote light control is shown in the block diagram displayed in Figure 3. Several lights are on in different states; lights 1, 2, and 3 are turned on and shining, while light 4 is regulated by the pulse width modulation technology. In the example version, this system consists of the following four topics to build four LEDs (or light bulbs in the actual version). One of them is Topic dimmer, which allows users to manage the brightness as well as use it as a dimmer. Topiclight1, Topiclight2, and Topiclight3 are the remaining topics, and they are used to control the light states. Devices such as a PC, laptop, tablet, or mobile phone that are subscribed to the same topic on MQTT-Broker send "1" and "0" messages to that topic and subsequently transfer the

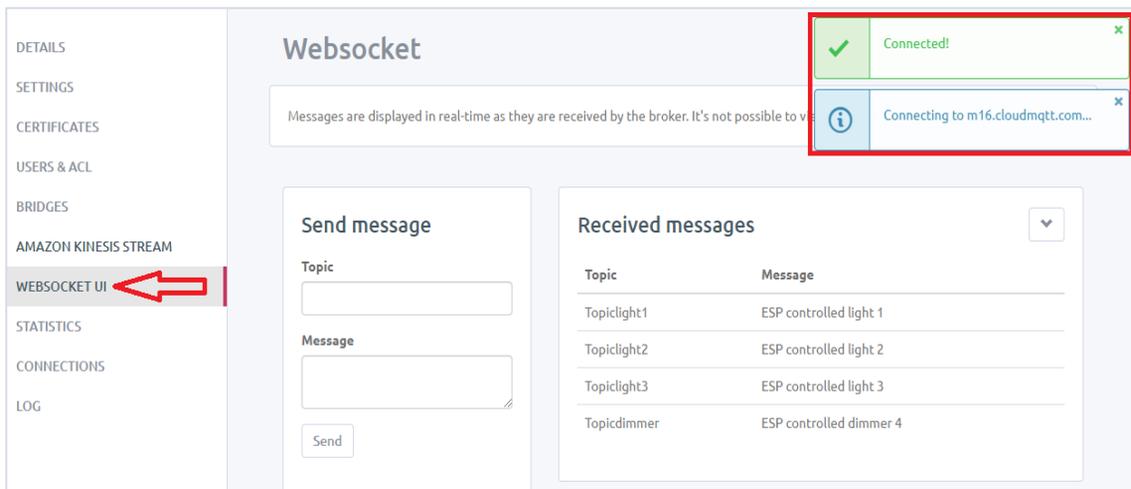
statement "on" or "off" signal to the ESP32, which controls the light states. Take a look at the guided presentation given in Figure 4 to see how well this argument is made. By sending a message 1 for each subject of light, all LEDs are switched on. It can be observed that there is one LED from the uppermost, whose brightness is controlled by the PWM approach, and that the other LEDs are changeable by turning them on or off.

**3.2. ESP32 network connection with the local wi-fi network**

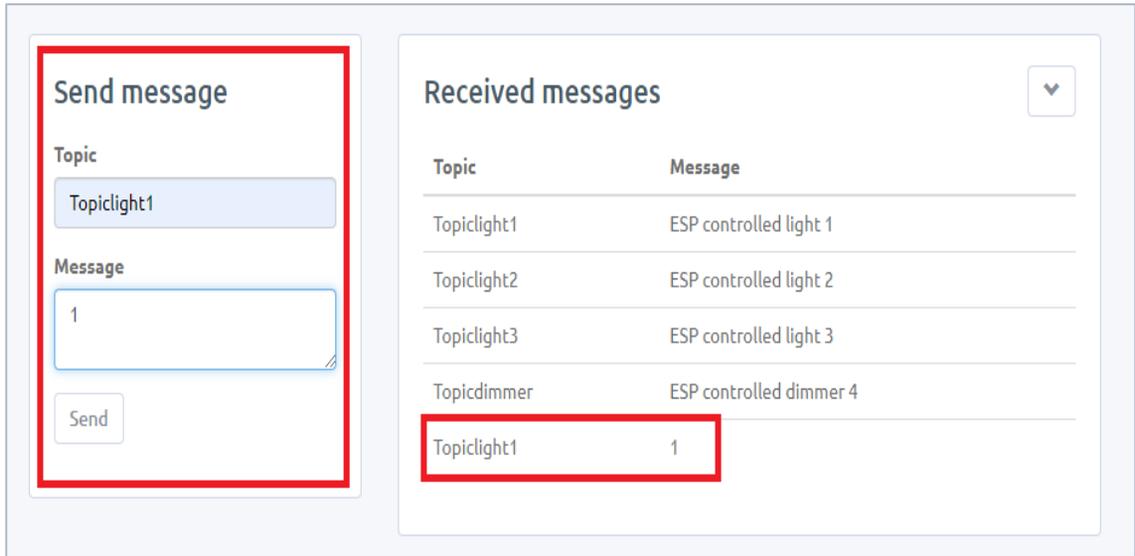
To communicate with the Internet, the ESP32 must be linked to a local Wi-Fi network and configured as a station mode, while also receiving the network SSID and Wi-Fi network name. The Wi-Fi Library must be included in order for the ESP32 to connect to the network and declare the SSID and password of this local Wi-Fi network. When the ESP32 is successfully connected to the Wi-Fi network, its access information is shown in the Arduino monitoring window. The ESP32 then acts as a client to receive broker statements on a previously published topic.

**3.3. Data Publishing/Subscribing to MQTT topic**

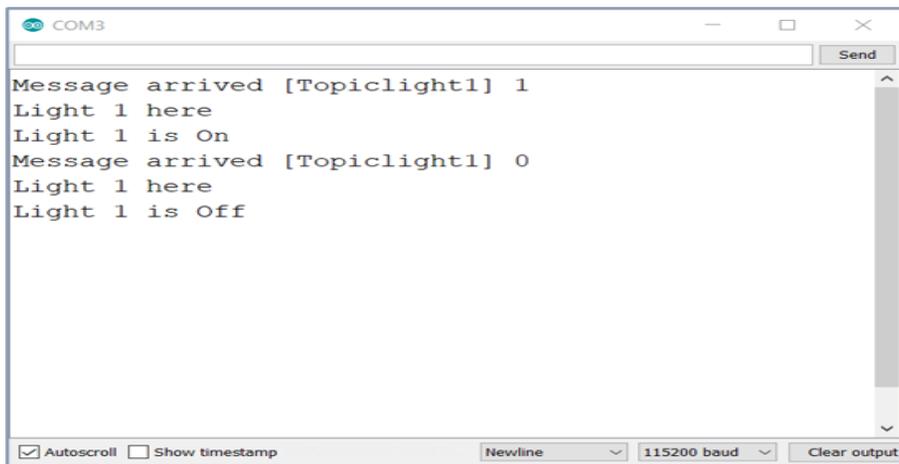
The system publishes a message to MQTT topics, it is needed to create an MQTT-Broker, which is established and hosted on CloudMQTT. Next, it is necessary to add PubSubClient from the Arduino library to use publishing on the ESP32. This library is used for Subscribing messages also. When the code is uploaded, the ESP32 subscribes to the same topic as the one published on the broker. Besides, the broker announces that ESP32 is connected and shows the topics to which ESP32 is subscribing, which is given in Figure 5. The instance project on the category of the cloudmqtt webpage chooses the WEBSOCKET UI will appear information about connecting to a broker (server cloudmqtt) marked on the red box and then upload the sketch from the computer. The information about publishing and subscribing to the project in the code compiled on the Arduino IDE will be informed in the Received messages box of the broker.



**Figure 4.** Publishing and Subscribing to the topic of MQTT cloud

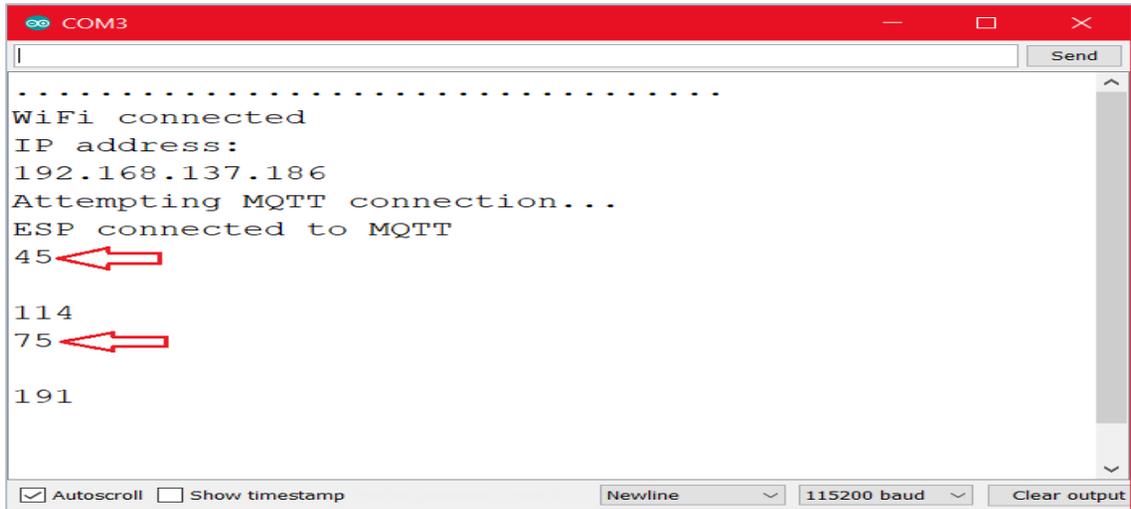


**Figure 5.** Sending and Receiving a message by MQTT-Broker



**Figure 6.** Switching information send back to Serial Monitor of Arduino IDE

Figure 6 illustrates the information about all lights that need control in the Received messages, which include four subjects, as indicated before, including Topiclight1, Topiclight2, Topiclight3, and Topicdimmer. Users might control the Topiclight1 by selecting it in the Send message box and then sending messages with the numbers 1 (high) to switch it on and 0 (low) to switch it off. Meanwhile, in the Arduino IDE's monitoring window, the connectivity information for light 1 will be shown as shown in Figure 7. Likewise, it is utilized for different topics. The user might regulate the brightness of the Led by choosing the Topicdimmer and providing values ranging from 1 to 100 directly from the clients. The PWM technique for controlling the brightness of an LED in the control unit offers information on the amount of the Led's brilliance, which is adjusted between 45 percent and 75 percent by varying the duty cycle width with an 8-bit resolution, as illustrated in Figure 8. As seen in Figure 9, the brightness of the led number 4 varies from low to high.

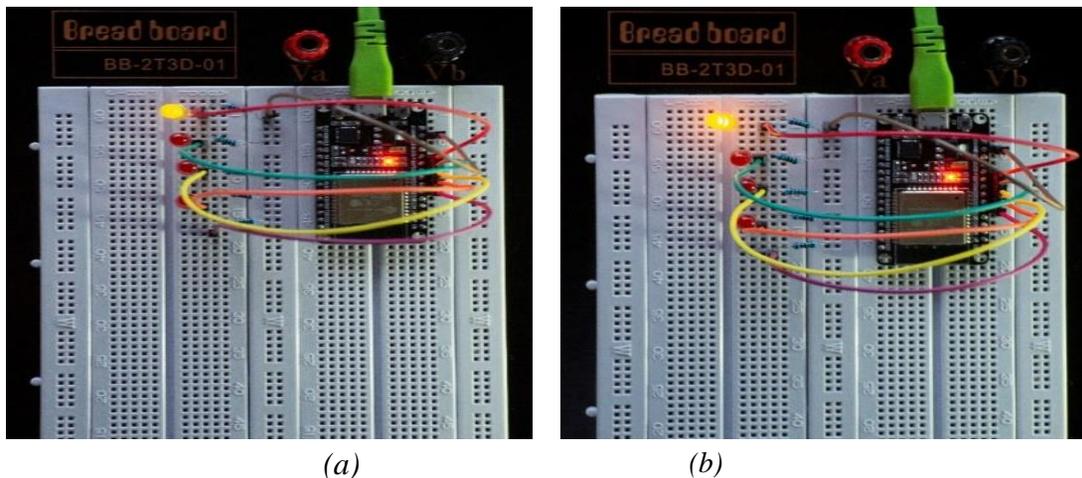


```

.....
WiFi connected
IP address:
192.168.137.186
Attempting MQTT connection...
ESP connected to MQTT
45
114
75
191

```

**Figure 7.** PWM method to control the brightness of Led



**Figure 8.** The brightness of Led is controlled by 10% (a) and 75% (b)

### 3.4. Actual Light Control Demonstration

It is evident that the experimental and actual versions vary in specific ways. The objective of the experimental version is to regulate the on and off states of the LED or its brightness. Some components are required to make a light control device a reality, including a power source for the whole circuit board, a relay component, a diver, and an ESP32. The primary components are relays, the integrated circuit LM317T TO220 (voltage regulator), the driver L298N, and the ESP32-DEVKIT V1, which has the kit with the chip that controls the whole board. The ESP32 DEVKIT V1 part requires a power supply of 7 to 12 volts. The ESP32 DEVKIT DOIT can operate on a power supply of around 6 to 20 volts, but the device can overheat and damage if it goes beyond the threshold of 12 volts ("Zerynth Documentation", 2022). Both sides of ESP32 have added two lines of extended pinouts according to the connection pins of ESP32 DEVKIT V1 and two lines of extended pins for power supply and ground used for the measurement in the other potential development projects.

The ESP32 DEVKIT V1 component needs a power source with a voltage range of 7 to 12 volts. It can work on a power source ranging from 6 to 20 volts, but if the voltage is exceeded by more than 12 volts, the device will overheat and become damaged ("Zerynth Documentation", 2022). The ESP32 has introduced two lines of expanded pinouts in accordance with the connection pins of the ESP32 DEVKIT V1, as well as two lines of extended pins for power supply and ground, which will be utilized for measurement in other prospective development projects on both sides of the board. The relay operates at 3 to 48 volts. Electromechanical relays control devices coupled with LEDs to indicate which relay is on or off when the lights switch. When the MQTT-Broker sends a statement to modify a light's status, the ESP32 must receive the signal and switch on or off the relays. The lights turn on or reverse. It can control lights, electric fans, motors, and other electrical equipment.

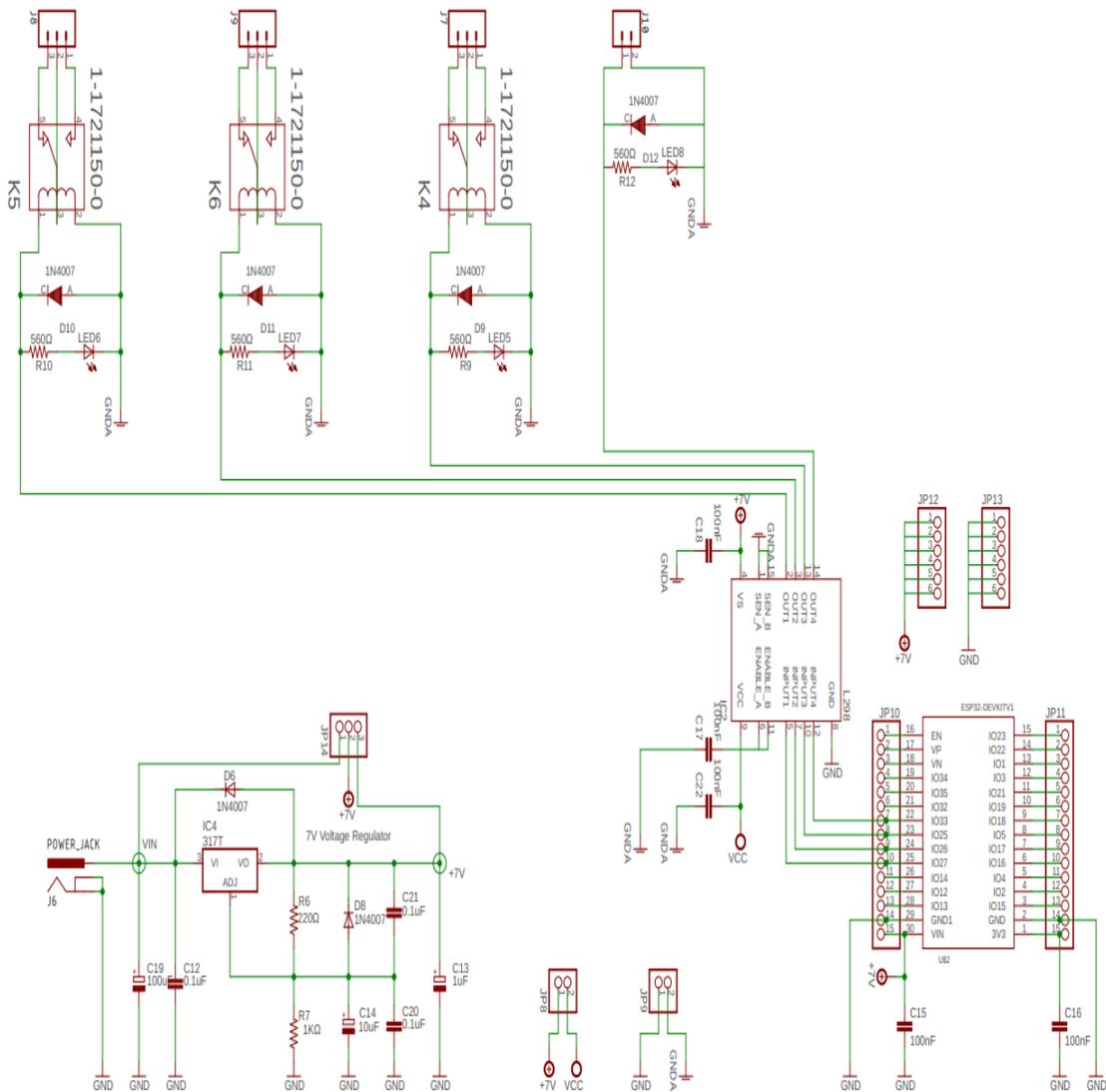
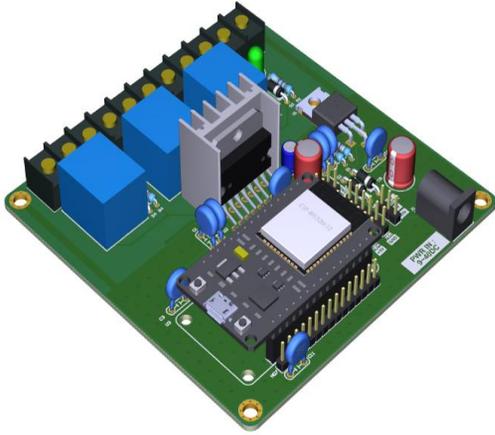
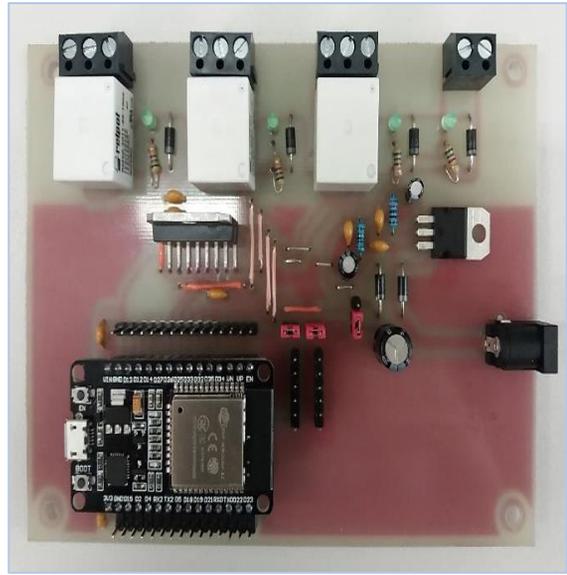


Figure 9. The schematic of remote control with ESP32



**Figure 10.** 3D layout of ESP32 control lights.



**Figure 11.** The Printed Circuit Board Assembly.

#### 4. Conclusion

The control devices may be linked from anywhere with an Internet connection over the Internet. IoT's effect in various industries and multi-function as an ultimate technique through the Internet to link control devices and lead them to become intelligent things were considered throughout the investigation. Sending data from a mobile phone browser or PC activates the ESP32 as a client to operate three bulbs remotely. Another bulb is controlled by PWM through MQTT-Broker without a smartphone app. This saves time and storage on control devices like mobile phones, laptops, and tablets since they connect to the Internet as clients.

MQTT Protocol's lightweight machine-to-machine message transfer is favored over HTTP in IoT. The user may use MQTT-Broker to research or learn on many websites. The ESP32 is a dual-core Wi-Fi and Bluetooth processor that helps manage electronic devices and create Internet connection, allowing data to be delivered to the cloud, a server for long distances, and support for certain devices that must be operated from hundreds of meters away.

This demonstration is not limited to smart remote lighting; it might incorporate a smart home system capable of controlling motors, air conditioners, and electric fans and altering heater level and motor speed through PWM. The weather station may also display the house's temperature. In business and agriculture, it's employed in automated irrigation systems. In this article, the example does not use all the sophisticated capabilities of the dual-chip ESP32 when the device loses Internet connection.

## References

- Bhooanas, N., & Sou, S. (2021). Performance modeling of multipath mobile data offloading in cellular/WiFi networks with bandwidth uncertainty. *Computer Networks*, 197, 108351. <https://doi.org/10.1016/j.comnet.2021.108351>
- Błaszkiwicz, L., Pazderski, E., Wolak, P., Marecki, A., Rudawy, P., & Olech, M. et al. (2021). Comparative measurements of radio frequency interference in the 10–250 MHz. *Measurement*, 184, 109916. <https://doi.org/10.1016/j.measurement.2021.109916>
- Understanding Bluetooth Range | Bluetooth® Technology Website. Bluetooth® Technology Website. (2022), from <https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/>.
- Finlay, D. (2016). Connected Health Approaches to Wound Monitoring. *Smart Bandage Technologies*, 229-244. <https://doi.org/10.1016/b978-0-12-803762-1.00007-2>
- The Internet of Things with ESP32. *Esp32.net*. (2022), from <http://esp32.net/>.
- About Arduino. *Arduino.cc*. (2022), from <https://www.arduino.cc/en/about#what-is-arduino>.
- Patel, C., & Doshi, N. (2020). A Novel MQTT Security framework In Generic IoT Model. *Procedia Computer Science*, 171, 1399-1408. <https://doi.org/10.1016/j.procs.2020.04.150>
- Prada, M., Reguera, P., Alonso, S., Morán, A., Fuertes, J., & Domínguez, M. (2016). Communication with resource-constrained devices through MQTT for control education. *IFAC-Papersonline*, 49(6), 150-155. <https://doi.org/10.1016/j.ifacol.2016.07.169>
- Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. 2017 IEEE International Systems Engineering Symposium (ISSE). <https://doi.org/10.1109/syseng.2017.8088251>
- Baig, M., Iqbal, M., Jamil, M., & Khan, J. (2021). Design and implementation of an open-Source IoT and blockchain-based peer-to-peer energy trading platform using ESP32-S2, Node-Red and, MQTT protocol. *Energy Reports*, 7, 5733-5746. <https://doi.org/10.1016/j.egyr.2021.08.190>
- Akagi, H., & Kondo, R. (2010). A Transformerless Hybrid Active Filter Using a Three-Level Pulsewidth Modulation (PWM) Converter for a Medium-Voltage Motor Drive. *IEEE Transactions On Power Electronics*, 25(6), 1365-1374. <https://doi.org/10.1109/tpel.2009.2040002>
- Holtz, J. (1992). Pulsewidth modulation-a survey. *IEEE Transactions On Industrial Electronics*, 39(5), 410-420. <https://doi.org/10.1109/41.161472>
- Holmes, D., & Lipo, T. (2003). *Pulse width modulation for power converters*. Hoboken, NJ: John Wiley.
- Ding Yi, Fan Binwen, Kong Xiaoming, & Ma Qianqian. (2016). Design and implementation of mobile health monitoring system based on MQTT protocol. 2016 IEEE Advanced Information Management, Communicates, Electronic And Automation Control Conference (IMCEC). <https://doi.org/10.1109/imcec.2016.7867503>
- Yang, Z., Zhou, Q., Lei, L., Zheng, K., & Xiang, W. (2016). An IoT-cloud Based Wearable ECG Monitoring System for Smart Healthcare. *Journal Of Medical Systems*, 40(12). <https://doi.org/10.1007/s10916-016-0644-9>
- Froiz-M´iguez, I., Fern´andez-Caram´es, T., Fraga-Lamas, P., Castedo, L., 2018. Design, implementation and practical evaluation of an iot home automation system for fog

computing applications based on mqtt and zigbee-wifi sensor nodes. *Sensors* 18, 2660.

- Jamborsalamati, P., Fernandez, E., Moghimi, M., Hossain, M., Heidari, A., & Lu, J. (2019). MQTT-Based Resource Allocation of Smart Buildings for Grid Demand Reduction Considering Unreliable Communication Links. *IEEE Systems Journal*, 13(3), 3304-3315. <https://doi.org/10.1109/jsyst.2018.2875537>
- Katsikeas, S., Fysarakis, K., Miaoudakis, A., Van Bemten, A., Askoxylakis, I., Papaefstathiou, I., & Plemenos, A. (2017). Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol. 2017 IEEE Symposium On Computers And Communications (ISCC). <https://doi.org/10.1109/iscc.2017.8024687>
- Patel, C., Doshi, N. (2019). Security Challenges in IoT Cyber World. Springer International Publishing, Cham. pp. 171–191. URL: <https://doi.org/10.1007/978-3-030-01560-2-8>
- Dhar, P., & Gupta, P. (2016). Intelligent parking Cloud services based on IoT using MQTT protocol. 2016 International Conference On Automatic Control And Dynamic Optimization Techniques (ICACDOT). <https://doi.org/10.1109/icacdot.2016.7877546>
- Pooja, S., Uday, D., Nagesh, U., & Talekar, S. (2017). Application of MQTT protocol for real time weather monitoring and precision farming. 2017 International Conference On Electrical, Electronics, Communication, Computer, And Optimization Techniques (ICECCOT). <https://doi.org/10.1109/iceccot.2017.8284616>
- CISCO, 2014. The internet of things reference model. URL: [http://cdn.iotwf.com/resources/71/IoT\\_Reference\\_Model\\_White\\_Paper\\_June\\_4\\_2014.pdf](http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf)
- Zerynth Documentation. Docs.zerynth.com. (2022), from [https://docs.zerynth.com/latest/official/board.zerynth.doit\\_esp32/docs/index.html](https://docs.zerynth.com/latest/official/board.zerynth.doit_esp32/docs/index.html).