Brajan Gąbka

POLITECHNIKA LUBELSKA

Wydział Elektrotechniki i Informatyki

Projekt Inżynierski

Zastosowanie mikrokontrolerów i sygnałów bezprzewodowych do zdalnego sterowania oświetleniem LED w sieci AC 230 V

Brajan Gąbka

Data: 16.11.2024

Spis treści

1 Opis ogólny....3
1.1 Cel projektu....3
1.2 Zarys problemów....3
1.3 Zakres projektu....3
2 Specyfikacja wymagań....4
2.1 Wymagania funkcjonalne....4
2.2 Wymaganie niefunkcjonalne....4
2.3 Założenia i ograniczenia....4
3 Projekt elektryczny....5
3.1 Schemat połączeń....5
3.2 Lista komponentów....7
3.3 Opis komponentów....7
4 Oprogramowanie....10
4.1 Zaimplementowane biblioteki....10
4.2 Projekt systemu sterowania....11
4.3 BLUETOOTH....12
4.4 Strona internetowa....15
4.5 SPIFFS....15
5 Projekt miniaturowego domu....16
6 Implementacja....17
7 Instrukcja użytkownika....18
7.1 Wprowadzenie....18
7.2 Ustawienia ESP Oprawy....18
7.3 Ściemnianie oprawy LED....18
7.4 Dodanie nowych opraw do programu....19
7.5 Nie wszystkie urządzenia zostały znalezione....20
7.6 Master komunikuje się z oprawą ale LED się nie świeci....20
8 Kosztorys....20
9 Podsumowanie i wnioski....21
10 Załączniki....22

1 Opis ogólny

Niniejsza dokumentacja techniczna stanowi opis projektu inżynierskiego pod tytułem: „*Zastosowanie mikrokontrolerów i sygnałów bezprzewodowych do zdalnego sterowania oświetleniem LED w sieci AC 230 V*”.

1.1 Cel projektu

Celem projektu jest stworzenie systemu sterowania oświetleniem LED, który można zastosować w budynku/mieszkaniu jednorodzinnym bez konieczności dodania przewodów do już istniejącej instalacji elektrycznej.

1.2 Zarys problemów

W związku z wzrastającymi wymaganiami mieszkańców i niepoprawnym oszacowaniu planu budynku/mieszkania przez inwestora, zastosowanie inteligentnego oświetlenia, w którym sygnały sterujące byłyby przenoszone przez przewód, powodowałoby wysokie koszty implementacji danego rozwiązania w już istniejącym budynku. [1]

1.3 Zakres projektu

Projekt obejmuje projektowanie, implementację, testowanie i przedstawienie systemu sterowania oświetleniem LED, w miniaturowym domu zasilanym z gniazdka elektrycznego 230 V 50 Hz. Do przesyłu informacji została zastosowana technologia Bluetooth. Jest stosowana do komunikacji między serwerem a sterownikami wykonawczymi, którymi są mikrokontrolery, oraz między telefonem a serwerem.

2 Specyfikacja wymagań

2.1 Wymagania funkcjonalne

- System powinien być w stanie komunikować się z smartfonem za pomocą Bluetooth oraz WiFi
- System powinien umożliwiać wyłączenie lub włączenie oświetlenia LED
- System powinien umożliwiać ściemnianie oświetlenia LED przez wielu użytkowników
- System powinien umożliwić sprawdzenie aktualnego poziomu oświetlenia
- System powinien kontrolować oświetlenie z poziomu pojedynczej oprawy jak i wszystkich dostępnych opraw

2.2 Wymaganie niefunkcjonalne

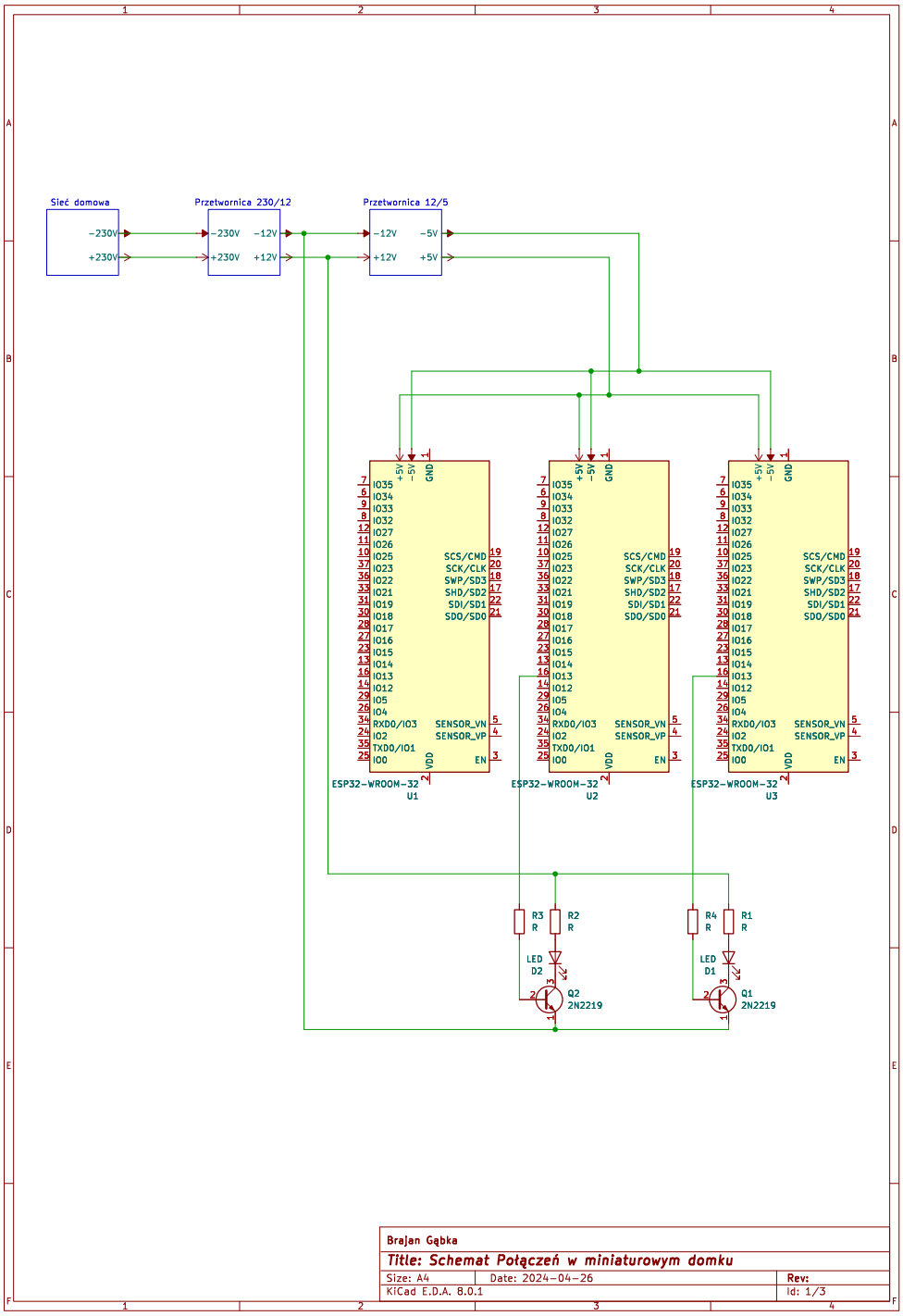
- System powinien nie wymagać remontu pokoju/budynku, w którym ma funkcjonować
- System powinien być łatwy w implementacji
- System powinien nie wymagać instalacji aplikacji na smartfon do sterowania oświetleniem

2.3 Założenia i ograniczenia

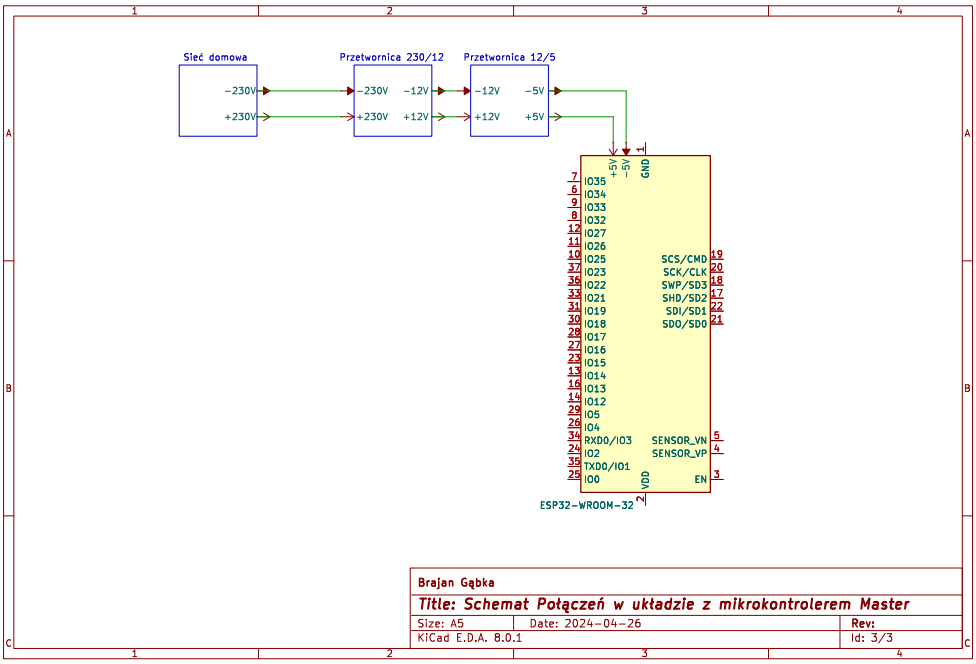
- System będzie wykorzystywał bezprzewodową transmisje danych WiFi do komunikacji między urządzeniami
- System będzie wykorzystywał bezprzewodową transmisje danych Bletooth do komunikacji między ESP oprawy a smartfonem
- Użytkownik będzie musiał zainstalować na telefon dowolną aplikację umożliwiającą komunikację Bluetooth z oprawami do implementacji systemu oświetlenia
- System będzie przystosowany do sieci niskiego napięcia 230 V
- LED zastosowany w oświetleniu będzie przystosowany do 12 V

3 Projekt elektryczny

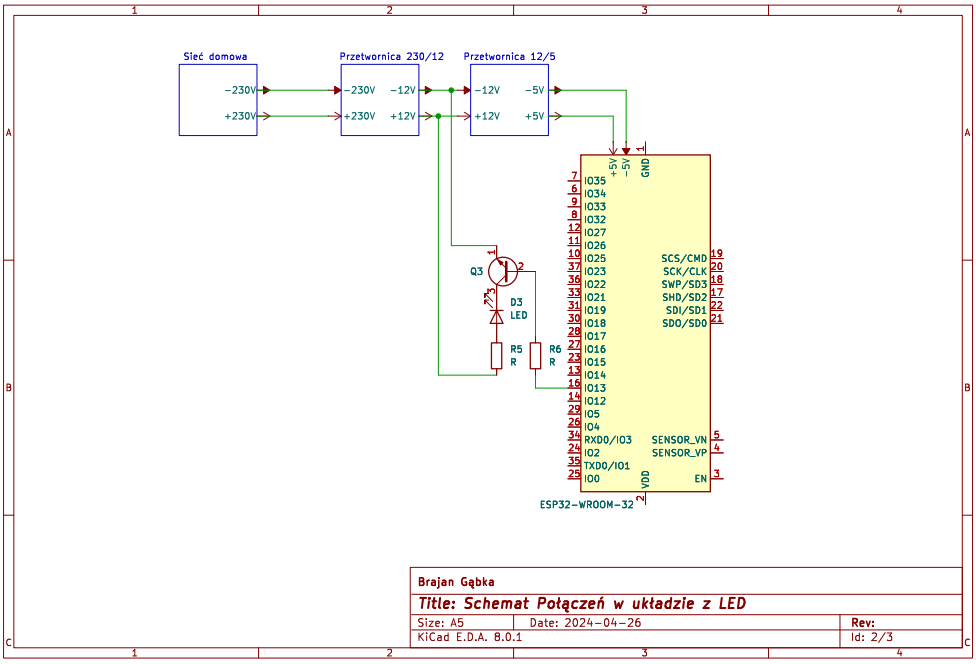
3.1 Schemat połączeń



Rys 1: Schemat Połączeń w miniaturowym domu



Rys 2: Schemat Połączeń w układzie z ESP Master

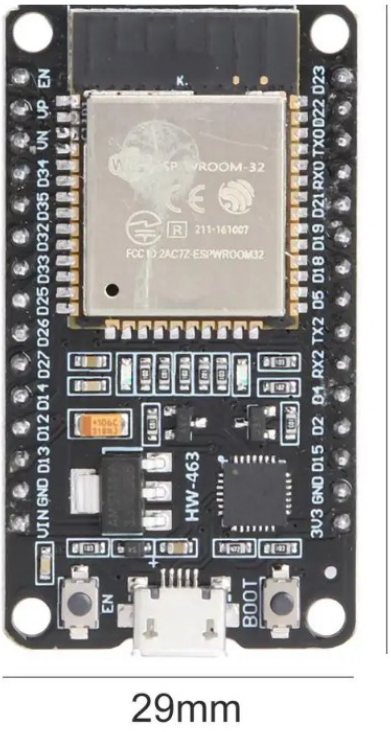


Rys 3: Schemat Połączeń w układzie z ESP Oprawa

3.2 Lista komponentów

- ESP-WROOM-32 [2]
- Przetwornica 240/12 V AC/DC [3]
- Przetwornica 12/5 V DC/DC [4]
- Tranzystor BC547B [5]

3.3 Opis komponentów



ESP-WROOM-32

ID FCC: 2AC7Z-espwroom32

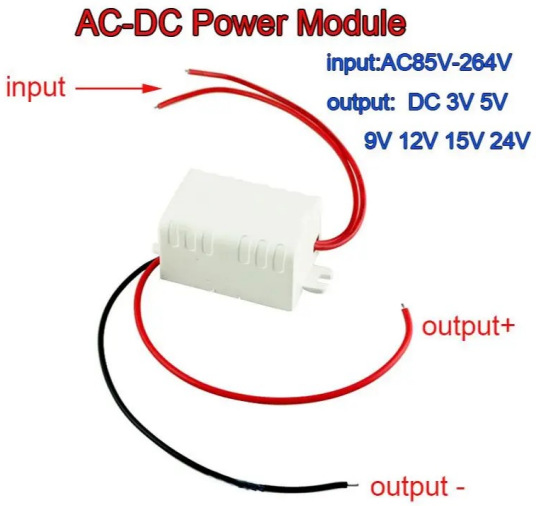
Moduł 802.11b/g/WiFi + BT

Bluetooth: Zgodne ze standardami Bluetooth 4.2 BR/EDR i BLE

Szybkość portu szeregowego: 115200

Dwurdzeniowy 32-bitowy procesor

Pobór mocy: 300 mA 3,3 V



Przetwornica

Model: LS-3S-WA

Napięcie wejściowe: AC 100-240 V 50/60 Hz

Napięcie wyjściowe: DC 12 V

Prąd: 0,3 A

Tętnienie: <150mV

Dokładność napięcia wyjściowego: ± 3%

Przetwornica

Napięcie wejściowe: DC 12 V

Napięcie wyjściowe: DC 5 V

Prąd: 3 A

Wydajność: 96%

Rozmiar: 46 mm x 27 mm x 14 mm

Producent: Diymore

Tranzystor

Model: BC547B

Napięcie maksymalne kolektor-emiter: 50 V

Prąd maksymalny kolektora: 0,1 A

Konfiguracja wyprowadzeń: CBE

Obudowa: TO92 (THT)

4 Oprogramowanie

4.1 Zaimplementowane biblioteki

WiFi.h

Umożliwia urządzeniom ESP32 łączenie się z sieciami Wi-Fi, pełniąc rolę klienta lub tworząc punkt dostępowy.

WebServer.h

Pozwala na tworzenie serwera HTTP na urządzeniach ESP, umożliwiając obsługę żądań HTTP i generowanie stron internetowych.

WebSocketsServer.h

Umożliwia tworzenie serwera WebSocket na ESP, co pozwala na dwukierunkową, bezpośrednią komunikację w czasie rzeczywistym z przeglądarką lub innymi urządzeniami.

WebSocketsClient.h

Umożliwia urządzeniu ESP działanie jako klient WebSocket, co pozwala na komunikację w czasie rzeczywistym z serwerami WebSocket.

ArduinoJson.h

Biblioteka do obsługi formatu JSON, umożliwia łatwe przetwarzanie i generowanie danych JSON, co jest przydatne w aplikacjach IoT.

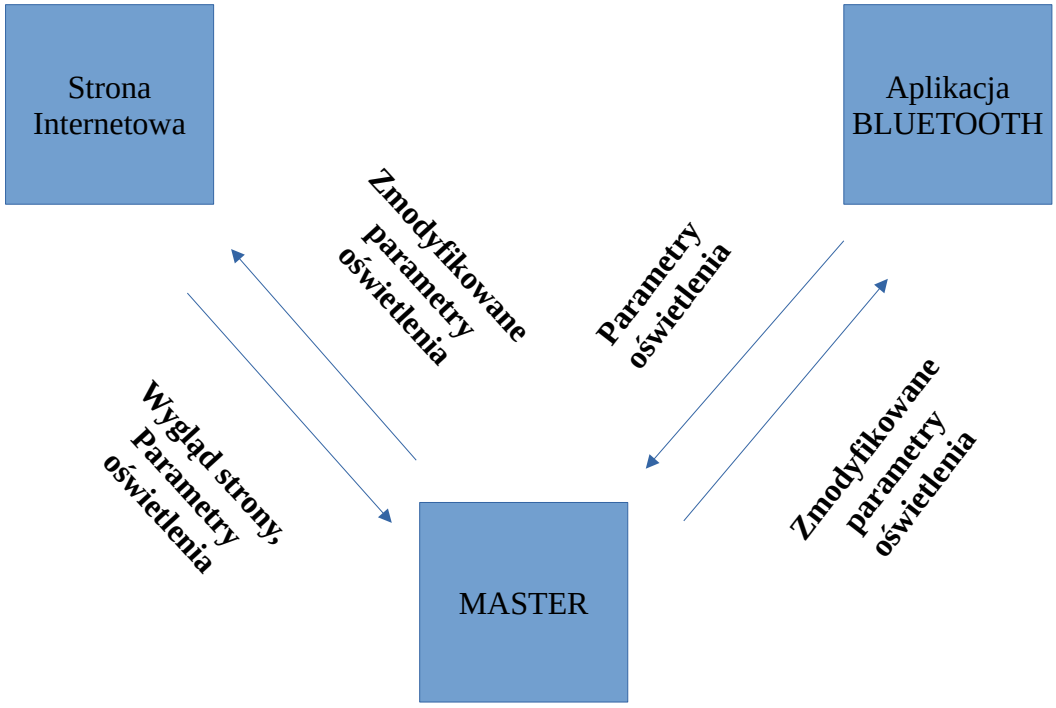
SPIFFS.h

Obsługuje system plików SPIFFS (Serial Peripheral Interface Flash File System) na ESP, pozwalając na przechowywanie i zarządzanie plikami na pamięci Flash.

BluetoothSerial.h

Biblioteka Bluetooth dla ESP32, umożliwiająca komunikację z innymi urządzeniami za pomocą Bluetooth w trybie seryjnym.

4.2 Projekt systemu sterowania



Rys 4: Schemat Systemu Sterowania za pomocą ESP32 jako Master dla Aplikacji Webowej

Użytkownik ma możliwość modyfikowania wartości oświetlenia przez stronę internetową. Nie jest wymagane wysłanie zapytania o aktualne parametry oświetlenia. Aktualizacja parametrów oświetlenia dla wszystkich użytkowników następuje automatycznie po modyfikacji.

4.3 BLUETOOTH

```
void bluetoothActive() {
  flashing();
  Serial.println("Próba połączenia z Bluetooth");
  if (btON == false) {
    SerialBT.begin(lightFixture);
    btON = true;
  }
  delay(1000);
  if (SerialBT.connected()) {
    menu();
    messageBTMenu = "x";
    while (messageBTMenu != "0") {
      flashing();
      messageBT = "x";
      messageBTMenu = readBT();
      if (messageBTMenu == "1") {  // Obsługa menu ...
      } else if (messageBTMenu == "2") { ...
      } else if (messageBTMenu == "3") { ...
      } else if (messageBTMenu == "4") { ...
      }
      if (!SerialBT.connected()) {
        break;
      }
        }
      }

      if (!SerialBT.connected()) { //Zerwane połącenie Bluetooth
        break;
      }
    }
    saveFile();
    resetFixture = false;
    SerialBT.end();
    WiFi.disconnect(true);
    btON = false;
  }
}
```

Tekst 1: Funkcja bluetoothActive()

Jest to główna funkcja odpowiedzialna za komunikację Bluetooth. Występują w niej dwie pętle, pierwsza pętla jest odpowiedzialna za utworzenie menu, w którym użytkownik będzie miał możliwość wybrania zmiany:

Nazwy,

Serwera,

Hasła,

Hosta.

W czasie trwania pierwszej pętli wykonuje się funkcja flashing(), jest ona odpowiedzialna za umożliwienie rozpoznania modyfikowanej oprawy poprzez rozpoczęcie migania diodą.

```
void flashing() {
  currentMillis = millis();

  // Sprawdź, czy minął czas na zmianę jasności
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    // Zmiana jasności diody LED
    analogWrite(pinLED, dutyCycle);

    if (increasing) {
      dutyCycle++;
      if (dutyCycle >= 255) {
        increasing = false;  // Zmień kierunek na zmniejszanie
      }
    } else {
      dutyCycle--;
      if (dutyCycle <= 0) {
        increasing = true;  // Zmień kierunek na zwiększanie
      }
    }
  }
}
```

Rys 5: Funkcja flashing()

Druga pętla jest odpowiedzialna za oczekiwanie na wpisanie przez użytkownika modyfikacji.

```
if (messageBTMenu == "1") {  // Obsługa menu
  SerialBT.println("1. Nazwa: " + String(lightFixture));
  SerialBT.println("Wybrano zmianę nazwy, 0 cofnij");
  while (messageBT != "0") {
    messageBT = readBT();
    if (messageBT.length() > 1) {
      lightFixture = messageBT;
      SerialBT.println("Zmieniono Nazwę na: " + String(lightFixture));
      menu();
      break;
    } else if (messageBT == "0") {
      menu();
      break;
    }
    if (!SerialBT.connected()) {
      break;
    }
  }
}
```

Rys 6: Pętla oczekująca na wpisanie wartości przez użytkownika

Po wybraniu opcji rozłączenia się, ESP zapisuje aktualne dane do pliku SPIFFS, w przypadku przerwania połączenia, ESP nie aktualizuje danych. SPIFFS pozwala na odczytanie danych w przypadku zaniku napięcia.

```
void saveFile() {
  File file = SPIFFS.open("/config.txt", "w");
  file.println(ssid);
  file.println(password);
  file.println(serverHost);
  file.println(lightFixture);
  file.close();
}
```

Rys 7: Funkcja saveFile()

4.4 Strona internetowa

Do prawidłowego działania strony internetowej, został zastosowany JavaScript, który jest szeroko stosowany w HTML do dodawania interaktywności i logiki na stronach internetowych. JavaScript działa po stronie klienta, umożliwiając modyfikowanie wartości suwaka i przekazywanie aktualnych danych do serwera, oraz odbieranie aktualnych danych z serwera. Kod strony internetowej jest dostępny na GitHub.

4.5 SPIFFS

SPIFFS jest stosowany w ESP Oprawa i ESP Master, umożliwia zapisanie i odczytanie danych potrzebnych do łączenia się z siecią oraz zapisanie strony internetowej serwera.

5 Projekt miniaturowego domu

Do zasymulowania sterowania oświetleniem, wykonano projekt domku miniaturowego.

Rys 8: Domek z wyłączonym oświetleniem

Rys 9: Domek z włączonym oświetleniem

6 Implementacja

Układ sterujący LED (Rys 3) powinien zostać umieszczony w oprawie oświetleniowej. Wszystkie elementy z wyjątkiem LED należy umieścić nad oprawą oświetleniową w celu ukrycia elementów elektronicznych.

Układ odpowiedzialny za komunikację (Rys 2) wymaga dostępu do sieci zasilającej 230 V AC lub 5 V DC. Zastosowanie zasilania 5 V pozwala na pominięcie stosowania przetwornicy.

Układ Rys 1: Schemat Połączeń w miniaturowym domu jest stosowany wyłącznie w pokazowym przedstawieniu niniejszego projektu i zawiera się w konstrukcji, która do zasilenia wymaga dostępu do gniazdka elektrycznego.

7 Instrukcja użytkownika

7.1 Wprowadzenie

Inteligentne oświetlenie zostało zaprojektowane w celu zwiększenia komfortu użytkowania oświetlenia i umożliwienia zdalnego sterowania ściemnianiem LED z dowolnego urządzenia zdolnego do zalogowania się na stronę internetową dostępną na Wi-Fi domowym i utworzonym przez ESP Master.

7.2 Ustawienia ESP Oprawy

Bluetooth

Aby móc korzystać z produktu, należy zainstalować na smartfonie z dostępem do Bluetooth aplikację pozwalającą na terminalową komunikację Bluetooth. Sprawdzoną i zalecaną aplikacją jest Serial Bluetooth Terminal.

Następnie należy połączyć się z ESP Oprawa, miganie diodą informuje o połączeniu się z daną oprawą za pomocą Bluetooth. W terminalu wyświetli się menu z opcjami zmiany: Nazwy, Hosta,Serwera i Hasła.

Wybranie opcji pozwala na modyfikację, w przypadku rezygnacji należy wpisać cyfrę 0. Aby wyjść i zapisać, należy w menu wpisać znak 0, rozłączenie się w inny sposób spowoduje cofnięcie zmian.

WiFi

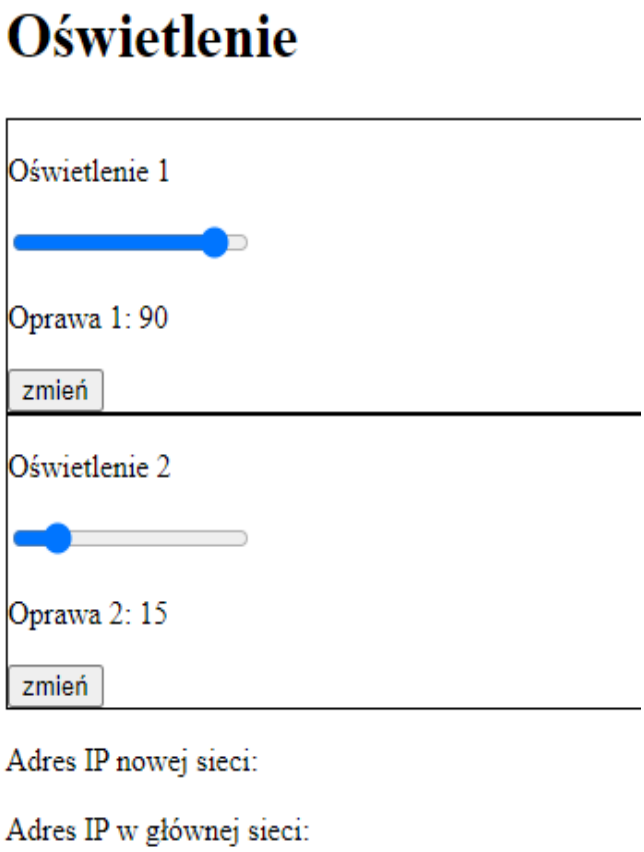
Aby móc korzystać z produktu należy sprawdzić do jakiego adresu został przypisany mikrokontroler Master lub połączyć się z domyślnym adresem 192.168.4.1. Po wyświetleniu się strony dostępne są suwaki, które pozwalają na ustawienie oświetlenia, oraz przyciski umożliwiające ESP Oprawy na dostęp do transmisji Bluetooth.

7.3 Ściemnianie oprawy LED

Aby ściemnić oświetlenie, należy połączyć się ze stroną internetową za pomocą adresu IP nowej sieci lub głównej sieci, a następnie wybrać suwakiem natężenie oświetlenia wyrażonego w procentach.

Przykład:

Chcę, aby oprawa „1” osiągnęła docelowy poziom oświetlenia „90” %, a oprawa „2” poziom „15” %.
W tym celu ustawię pierwszy suwak na 90, a drugi suwak na 15.



Rys 10: Wygląd Strony internetowej i zmiana wartości oświetlenia

7.4 Dodanie nowych opraw do programu

Jeśli oprawa nie jest połączona z siecią Wi-Fi, to dostępny jest do oprawy dostęp przez Bluetooth. Za pomocą terminala Bluetooth należy wpisać: Nazwę oprawy, ID Hosta, nazwę serwera i Hasło do serwera. Jeśli wszystkie dane będą poprawne, to Oprawa automatycznie połączy się z serwerem i będzie dostępna na stronie internetowej. Jeśli oprawa nie może się połączyć z serwerem, to Oprawa umożliwia dostęp do niej przez Bluetooth.

7.5 Nie wszystkie urządzenia zostały znalezione

Jeżeli dodanie nowego urządzenia nie powiodło się, należy sprawdzić czy:

ID hosta serwera jest taki sam jak **Adres IP nowej sieci** dostępny na stronie internetowej,

Nazwa serwera jest taka sama jak nazwa Wi-Fi utworzonej przez ESP Master,

Hasło jest poprawne.

7.6 Master komunikuje się z oprawą ale LED się nie świeci

Jeżeli komunikacja zachodzi prawidłowo, to należy sprawdzić czy oświetlenie LED nie uległo spaleniu.

8 Kosztorys

Tabela 1: Kosztorys miniaturowego domku

Nazwa	Ilość	Cena/opakowanie, zł
ESP-WROOM-32	3	18,63
Przetwornica 240/12 V AC/DC	1	17,19
Przetwornica 12/5 V DC/DC	1	10,27
Tranzystor BC547B x200	1	16,46
LED 12 V x10	1	14,10
Domek	1	10,00
Total		123,91

9 Podsumowanie i wnioski

Upload stron internetowych do pamięci SPIFFS jest możliwy wyłącznie na starszej wersji oprogramowania „Arduino IDE 1.8.19”. Aktualna wersja „Arduino IDE 2.3.2” nie jest kompatybilna z biblioteką „SPIFFS.h”.[6]

Domyślny schemat wielkości partycji pamięci na mikrokontrolerze „ESP-WROOM-32” jest za mały, aby zapisać wszystkie biblioteki i kod do transmisji Bluetooth i WiFi. Aby rozwiązać ten problem, należy zmienić schemat partycji na „Huge APP (3MB No OTA/1MB SPIFFS)”.[7]

Zastosowanie WebSocket sprawdza się w sterowaniu oświetleniem, jednak wymaga to sporych nakładów pracy nad programem obsługujących WebSocket serwera i klienta. Wybranie protokołu HTTP również spełniłoby początkowe założenia projektu i byłoby szybsze w implementacji.[8]

Projekt inżynierski jest dostępny pod linkiem GitHub.[9]

10 Załączniki

Bibliografia

1: SmartSpace, Instalacja tradycyjna vs inteligentna - którą wybrać?, https://www.smartspace.pl/blog/blog-smartspace-1/instalacja-tradycyjna-vs-inteligentna-ktora-wybrac-50
2: fcc.report, 2AC7Z-ESPWROOM32, https://fcc.report/FCC-ID/2AC7Z-ESPWROOM32
3: AliExpress, Przetwornica LS-3S-WA (240/12 V), https://pl.aliexpress.com/item/1005005778264197.html?spm=a2g0o.order_list.order_list_main.79.21ef1c24FQTKjH&gatewayAdapt=glo2pol
4: Diymore, Przetwornica (12/5 V), https://www.diymore.cc/products/12v-to-5v-3a-15w-dc-dc-buck-converter-step-down-module-car-monitor-power-supply-car-power-converter-regulator-adapter-for-car
5: Botland, Tranzystor BC547B, https://botland.com.pl/tranzystory-npn/254-tranzystor-bipolarny-npn-bc547b-50v-01a-5szt-5904422308025.html
6: randomnerdtutorials, Install ESP32 Filesystem Uploader in Arduino IDE, https://randomnerdtutorials.com/install-esp32-filesystem-uploader-arduino-ide/
7: robotzero.one, Partition Schemes in the Arduino IDE, https://robotzero.one/arduino-ide-partitions/
8: bulldogjob.pl, Prosto o WebSocket, https://bulldogjob.pl/readme/prosto-o-websocket
9: Brajan Gąbka, Zastosowanie mikrokontrolerów i sygnałów bezprzewodowych do zdalnego sterowania oświetleniem LED w sieci AC 230 V, https://github.com/Duke-Axer/LED-lighting_NKN