



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI

PROJEKT

Układ sterowania roletą okienną typu dzień – noc.

Day-night roller blind control system.

Autor: **Jan Rudecki**

Kierunek studiów: Elektronika i Telekomunikacja II stopień

Kraków, 2024

1. Wstęp

1.1. Cel Projektu

Celem projektu jest zbudowanie systemu sterowania roletą D-N, który umożliwi użytkownikowi kontrolowanie położenia rolety, jak i odczytanie parametrów otoczenia za pomocą sensorów. W ramach pracy powstała makieta zamkniętego pomieszczenia z oknem, w którym zostały zintegrowane sensory, monitorujące warunki środowiskowe.

Kluczową rolę w projekcie odgrywa technologia IoT, umożliwia kontrolę nad roletą poprzez zdalny dostęp, który powstał dzięki wykorzystaniu komunikacji modułu mikrokontrolera ESP32S z domową siecią Wi-Fi. Dzięki integracji sensorów środowiskowych, możliwy jest odczyt wartości natężenia światła oraz temperatury, co sprawia, że użytkownik ma możliwość precyzyjnego dostosowania położenia rolety do panujących warunków środowiskowych.

1.2. Wstęp do projektu

Stosowanie przysłon okiennych już od starożytności stanowiło istotny element funkcjonowania ludzi, zarówno w kontekście poprawy estetyki przestrzeni mieszkalnych, jak i w celu poprawy funkcjonalności i komfortu życia.

Mechanizm regulacji światła umożliwia płynną regulację dopływu światła do wnętrza pomieszczenia. Sterowanie roletami, podobnie jak w zwykłych roletach, może być realizowane za pomocą dedykowanego sznurka, bądź też za pomocą silnika.

Roletę D-N cechuje to że składa się z dwóch rodzajów materiału, jeden jest gęstszy przepuszczający mniej światła a drugi jest prawie przeźroczysty. Dzięki budowie rolety dzień- noc, użytkownik ma wiele możliwości ustawienia jej.

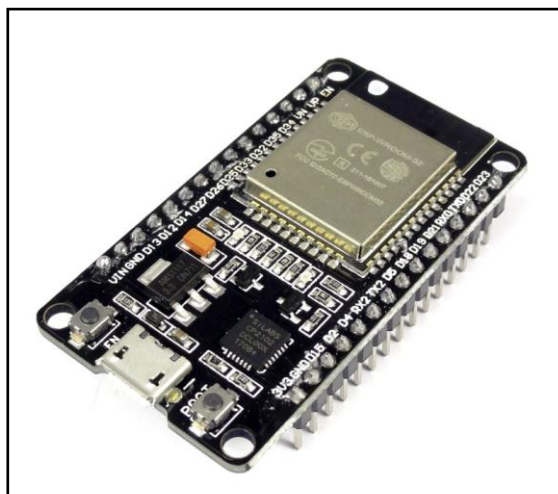


Rys. 1 Roleta Dzień-Noc

2. Dobór sprzętu i technologii

2.1. Mikrokontroler

W projekcie wykorzystano mikrokontroler ESP32S, który przedstawiono na Rys. 2. Posiada on wbudowany moduł ESP-WROOM-32, z którym tworzy układ posiadający 30 wyprowadzeń GPIO. Główną zaletą układu jest jego moc obliczeniowa, wystarczająca przy takim projekcie oraz niska cena – ok. 32 zł i jego możliwości komunikacji bezprzewodowej.



Rys. 2 ESP-32S

Istotną cechą układu ESP32 jest jego możliwość komunikacji bezprzewodowej za pomocą Bluetooth v4.2 lub Bluetooth BLE, oraz obsługa standardów Wi-Fi IEEE 802.11 b/g/n.

2.2. Komunikacja Wi-Fi

Jako sposób kontroli i odczytywania parametrów otoczenia z mikrokontrolera wybrano komunikację Wi-Fi. Główną zaletą takiej komunikacji jest to, że użytkownik ma możliwość połączenia się z mikrokontrolerem z dowolnego miejsca w obrębie domowej sieci Wi-Fi. Dzięki temu możliwa jest kontrola położenia rolety oraz odczytywanie warunków środowiskowych bez konieczności przebywania blisko rolety.

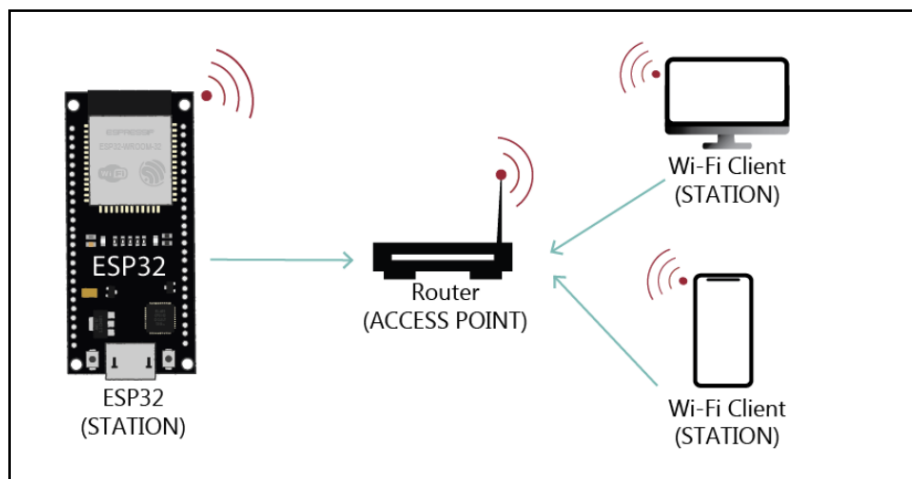
2.3. Sposób komunikacji bezprzewodowej za pomocą ESP3

ESP32 wykorzystując bibliotekę WiFi.h, jak mówi artykuł „ESP32 Useful Wi-Fi Library Functions”, ma możliwość nawiązywania komunikacji z siecią Wi-Fi na dwa podstawowe sposoby. Moduł może pracować w trybie punktu dostępu lub stacji. Pierwszy sposób polega na stworzeniu prywatnej sieci, do której użytkownik może się podłączyć, drugi opiera się na podłączeniu do istniejącej sieci domowej.

Dla domowego projektu lepszym rozwiązaniem jest zastosowanie trybu stacji, głównie ze względu na to, że projekt nie jest przenośny oraz w takim przypadku istnieje już domowa sieć Wi-Fi, więc nie ma potrzeby tworzyć kolejnej.

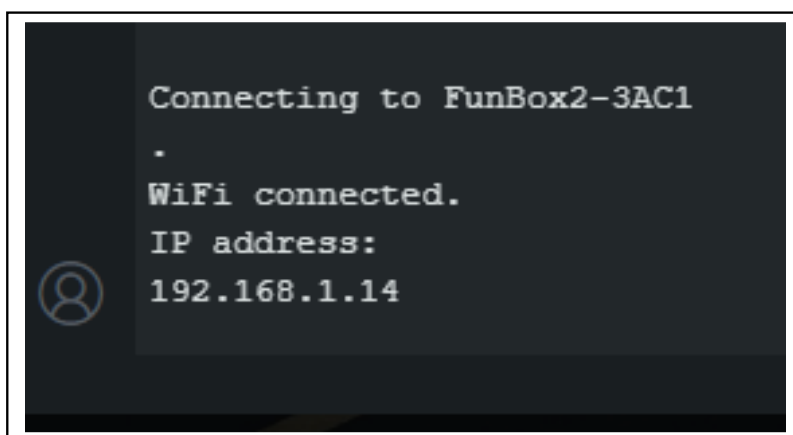
2.3.0. ESP32 jako Stacja Wi-Fi

Moduł ESP32 w trybie stacji łączy się z innymi sieciami Wi-Fi za pomocą takich urządzeń jak Router lub Access Point, co obrazuje Rys. 3.



Rys. 3 ESP32 jako stacja

Domowy Access Point przypisuje adres IP modułowi ESP32, umożliwiając mu komunikację z innymi urządzeniami (stacjami) podłączonymi do tej samej sieci. Po odpowiedniej Konfiguracji mikrokontrolera i wgraniu programu, możliwe jest sprawdzenie adresu IP, łącząc moduł z komputerem za pomocą USB, następnie, wystarczy uruchomić Serial Monitor lub inny program do komunikacji, naciskając jednocześnie przycisk EN na mikrokontrolerze. Po tych operacjach w programie Arduino IDE na Serial Monitorze powinien pojawić się komunikat jak na Rys. 4, przedstawiający nazwę sieci, do której łączy się mikrokontroler oraz adres, który otrzymał od domowego routera.



Rys. 4 Adres przydzielony dla ESP32

Możliwe jest również ustawienie statycznego adresu dla mikrokontrolera, jednak należy wtedy uważać by nie nadać już zajętego adresu.

W projekcie wykorzystano sposób komunikacji jako Stacja, ze względu na to, że cały projekt jest stacjonarny, przez co nie jest potrzebna komunikacja jako Access Point. Dodatkowo taki sposób komunikacji jest wygodny dla zastosowań typu Smart Home, umożliwiając kontrolę urządzeń bez konieczności łączenia się dodatkowej sieci.

2.4. Silnik Krokowy

2.4.1. Wybór silnika krokowego

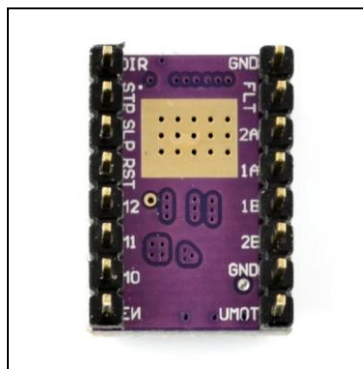
Do projektu wybrano silnik krokowy JK42HS40. Jego główne parametry to: rozdzielczość wynosząca 200 kroków na obrót, 4 wyprowadzenia (gdyż jest silnikiem bipolarnym) oraz możliwość obrotu w obu kierunkach, wymaga zasilania 12 V, a pobór prądu na cewkę wynosi 400 mA [15].



Rys. 5 Silnik Krokowy

2.4.2. Sterownik silnika krokowego StepStick DRV8825

Sterownik silnika krokowego oparty na układzie DRV8825, pozwala na sterowanie silnikiem krokowym o poborze 1,5 A, jednak, gdy użytkownik zastosuje radiator możliwy pobór wzrasta do 2,2 A na cewkę, przy zasilaniu 24 V. Napięcie zasilania wynosi od 12 do 24 V a sam moduł posiada kilka trybów pracy opierających się na wielkości kroku.



Rys. 6 Sterownik silnika krokowego StepStick DRV8825

Wyprowadzenia STP (STEP) oraz DIR (DIRECTION) służą do sterowania wielkością kroków i kierunkiem obrotów, w zależności od stanu podanego na pin DIR zmienia się kierunek na zgodny z obrotami zegara lub przeciwny, a podanie pojedynczego impulsu na pin STP powoduje obrót silnika o jeden krok.

Piny SLP (Sleep) oraz RST (Reset) służą do usypiania oraz resetowania układu, jeżeli podamy na nie stan niski system działa ciągle i piny nie wpływają na niego, bezpiecznym wyjściem jest zwarcie tych dwóch pinów, ze względu na zagrożenie zwarcia ze stanem wysokim i możliwe niekontrolowane uśpienie lub zresetowanie układu.

Wyprowadzenie FLT powinno być podpięte do 5 V ze względu na kompatybilność z układem A4988 oraz takie podłączenie informuje układ o braku błędów, a VMOT do napięcia zasilania między 12 a 24 V, sam silnik krokowy powinien być podpięty w wyprowadzenia między 1A-2B,

2.5. Sensory

2.5.1. Magistrala I²C

Magistrala I²C jest interfejsem służącym do nawiązywania komunikacji między urządzeniami. W magistrali występują dwa rodzaje urządzeń: urządzenia podrzędne zwane Slave oraz urządzenia nadrzędne nazywane Master. Komunikacja odbywa się za pomocą dwóch sygnałów:

- SDA (Serial Data), który odpowiada za przesyłanie danych

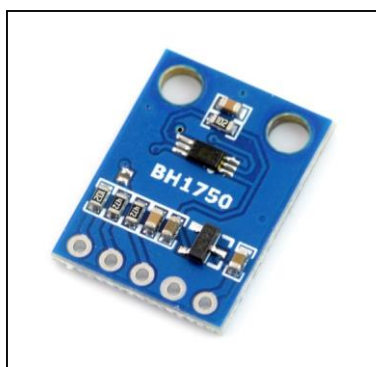
- SCL (Serial Clock) będący sygnałem zegara.

Mikrokontroler ESP32S posiada dwa interfejsy, które mogą obsłużyć magistralę I²C. Są to wyprowadzenia GPIO22 oraz GPIO23, będące pierwszą magistralą odpowiednio SDA oraz SCL i wyprowadzenia GPIO18 oraz GPIO19.

W przypadku takiej magistrali ważne jest by pamiętać o jej ograniczeniach, takich jak maksymalna liczba urządzeń na magistrali oraz maksymalna długość przewodów, wynikające z obciążenia magistrali i rezystancji przewodów. W projekcie sensory zostały rozłożone na dwie magistrale a długość przewodów jest ograniczona, przez co nie powinny wystąpić z tym problemy. Jednakże, gdyby projekt miał być rozszerzany, należy pamiętać o odpowiednich urządzeniach zwiększających zasięg magistrali.

2.5.2. Sensor Natężenia światła

Sensor BH1750 jest cyfrowym sensorem natężenia światła, na interfejs I²C. Zakres jego pracy obsługuje natężenia światła od 1 do 65535 lx. Czujnik posiada 5 wyprowadzeń, poza typowymi odpowiadającymi za magistralę I²C (SDA, SCL), zasilanie wynoszące 5 V lub 3,3 V, posiada wyprowadzenie odpowiedzialne za adresowanie. Podstawowy adres sensora to 0x23, jednak, gdy na wejście ADDR podamy stan wysoki adres sensora zmieni się na 0x5C, dzięki czemu możliwe jest proste podłączenie dwóch takich samych sensorów na jednej magistrali.



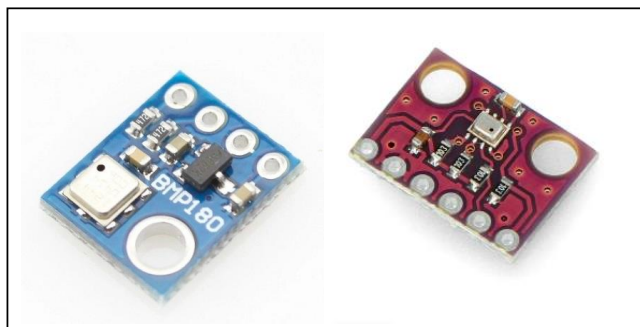
Rys. 7 Sensor natężenia światła BH1750

W pracy zdecydowano się na wybór tego sensora ze względu na dużą dokładność, szeroki zakres pomiaru oraz ze względu na pracę na magistrali I²C, dzięki czemu implementacja oraz rozkład okablowania jest mniej skomplikowany.

2.5.3. Sensory temperatury

Czujniki BMP180 oraz BMP280 są złożonymi sensorami o zbliżonych parametrach. Mają możliwość pomiaru ciśnienia, wysokości oraz temperatury. W projekcie wykorzystano tylko

pomiar temperatury, jednak dodatkowe możliwości czujników mogą być wykorzystane przy ewentualnych modyfikacjach układu. Oba sensory mogą pracować na magistrali I²C, a sensor BMP280 obsługuje również komunikację po magistrali SPI. Sensory powinny być zasilone napięciem 3.3 V, a zakres pomiaru temperatur jest od -40 do +85°C [24].



Rys. 8 Sensory natężenia światła BMP180 oraz BMP 280

Do realizacji projektu wybrano sensory temperatury BPM ze względu na duży zakres pomiarowy, przystępną cenę, dużą dokładność oraz wygodę obsługi magistrali I²C.

2.6. Sposób zasilania projektu

W projekcie do zasilania potrzebne są dwa różne napięcia zasilania, mikrokontroler ESP32 wymaga napięcia zasilania 5 V, a sterownik silnika wymaga napięcia zasilania 12 V. W tej sytuacji dobrym wyjściem jest zastosowanie jednego źródła zasilania do dwóch układów, z tego powodu zastosowano Powerbank.

Smart Powerbank ma dwa wyjścia, jedno z nich to złącze USB 5 V, za pomocą którego możliwe jest zasilanie mikrokontrolera ESP32 oraz wyjście 12 V, 1,5 A, poprzez wyjście RJ45.

Możliwe jest wykorzystanie wyjścia 12 V do zasilania sterownika silnika, poprzez technologię PoE (Power over Ethernet). Polega ona na przenoszeniu napięcia poprzez kabel typu skrętka, jest głównie wykorzystywana do zasilania urządzeń sieciowych, jednak możliwe jest wyprowadzenie z niej wymaganego napięcia, poprzez pary niebieską oraz brązową. Przewody niebieski oraz biało-niebieski są odpowiedzialne za napięcie +12 V, a przewody brązowy oraz biało-brązowy są masą.

Powerbank jest ładowany poprzez wejście 12 V, poprzez odpowiednią ładowarkę, pojemność akumulatora to 3400 mAh, a czas do pełnego naładowania to 3h.



Rys. 9 Smart Powerbank

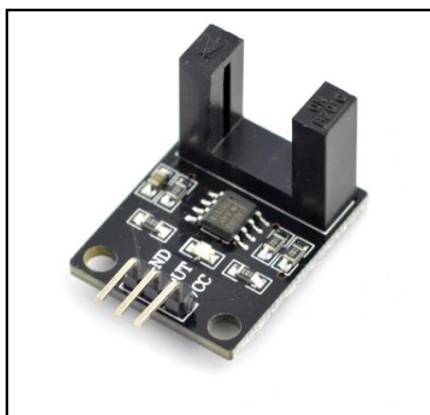
Zdecydowano się na Smart Powerbank ze względu na jego przenośność oraz obsługę PoE, co umożliwia łatwe zasilanie układów, nawet tych bardziej wymagających pod względem poziomu napięcia.

2.7. Czujnik szczelinowy z konwerterem LM393

W projekcie znaczącym aspektem jest kontrola położenia rolety, w tym celu zostały zastosowane dwa czujniki szczelinowe, jeden na górze a drugi na samym dole okna.

Moduł z czujnikiem szczelinowym wykrywa obiekty znajdujące się w szczelinie o szerokości 10 mm, wykorzystując transoptor oraz zwraca sygnał cyfrowy dzięki układowi komparatora LM393. Czulość czujnika zmienia się w zależności od napięcia zasilania.

Rola tych czujników opiera się na tym, że wykrywają, gdy roleta jest na samym dole albo u samej góry, za pomocą specjalnej wypustki przyczepionej do rolety. Wykrywanie położenia usprawnia mechanizm sterowania roletą. Jest to szczególnie istotne, gdy roleta znajdzie się w skrajnych położeniach - za nisko lub za wysoko, czujnik wykryje to w odpowiednim momencie i zatrzyma ją. Niezastosowanie takiego lub podobnego rozwiązania skutkowało tym, że użytkownik mógł przypadkowo za nisko opuścić roletę, albo sprawić, że nawinie się ona w drugą stronę, bądź też zablokuje się.



Rys. 10 Czujnik szczelinowy z konwerterem LM393 [26]

2.8. Makieta

Makieta testowego pomieszczenia została skonstruowana z płyt OSB, jej wymiary to 41x 50x 50 cm. Prostopadłościan mający imitować rzeczywiste pomieszczenie ma z samego przodu wycięte okno o wymiarze 34 na 40 cm, a w środku jest umieszczona roleta.

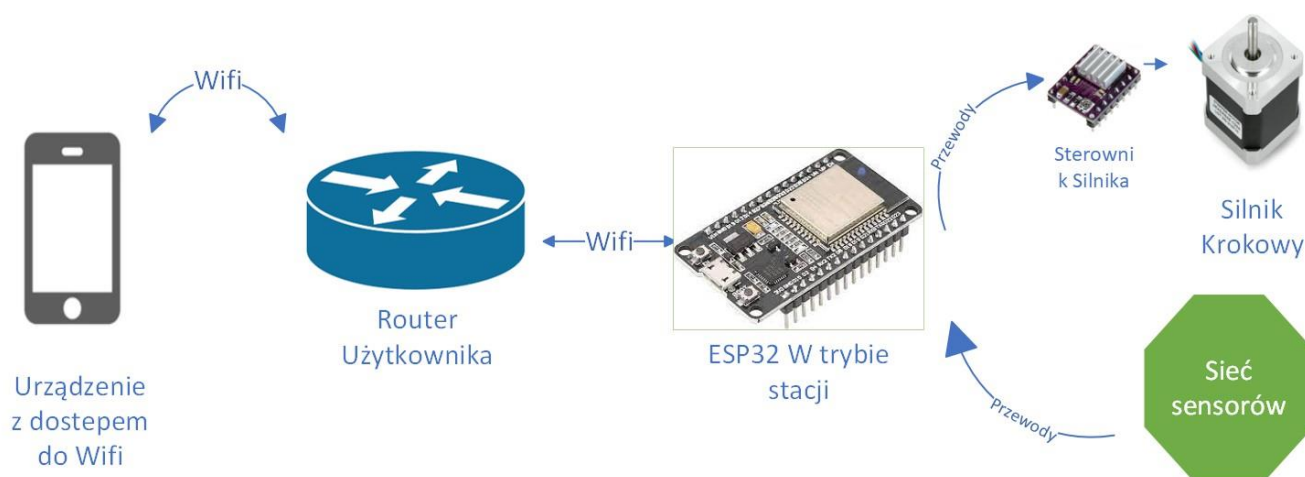
Wymiary makiety są stosunkowo duże, przez co w środku możliwe jest umieszczenie całego osprzętu, takiego jak: zasilanie, mikrokontroler, silnik z roletą i sensory.



Rys. 11 Makieta testowego pomieszczenia

3. Opis działania

Schemat na Rys. 12 przedstawia działanie zaprojektowanego układu. Użytkownik wykorzystując urządzenie, które ma dostęp do domowej sieci, wchodząc na odpowiednią stronę, ma możliwość sterowania roletą oraz odczytania wartości z sensorów. Komunikacja między urządzeniem użytkownika, Routerem oraz ESP32 odbywa się bezprzewodowo, a samo ESP32 steruje silnikiem poprzez sterownik oraz pobiera wartości z sensorów za pomocą przewodów. Mikrokontroler oraz sterownik silnika są zasilane z jednego urządzenia, które posiada włącznik zasilania. Możliwe jest również podłączenie się do sieci przewodowo łącząc się z routerem.



Rys. 12 Obrazowe przedstawienie działania projektu

Procedura rozpoczęcia pracy układu zaczyna się od uruchomienia urządzenia zasilającego cały projekt. W tym celu należy włączyć Powerbank, za pomocą przycisku umieszczonego na nim, wtedy zostaje zasilony układ ESP32 oraz sterownik silnika. Następnie należy odczekać aż mikrokontroler nawiąże połączenie z siecią domową, gdy to nastąpi, należy wejść na odpowiedni adres IPV4, który został nadany mikrokontrolerowi, poprzez domowy Router lub urządzenie Access Point. Po wejściu na stronę internetową wyświetli się odpowiedni interfejs użytkownika, można na nim odczytać wartości temperatury, natężenia światła oraz sterować silnikiem, który jest przy roletcie. Znając parametry otoczenia, użytkownik jest w stanie precyzyjnie zdecydować, jakie ustawienie rolet go interesuje. Sterowanie roletami D-N może odbywać się na dwa sposoby: manualny lub automatyczny.

Tryb manualny polega na tym, że użytkownik wybiera kierunek oraz prędkość, a rolety opuszczają się lub podnoszą aż do momentu zatrzymania ich poprzez naciśnięcie przycisku stop, bądź też po wykryciu przez czujnik szczelinowy skrajnego położenia rolety. Drugi tryb

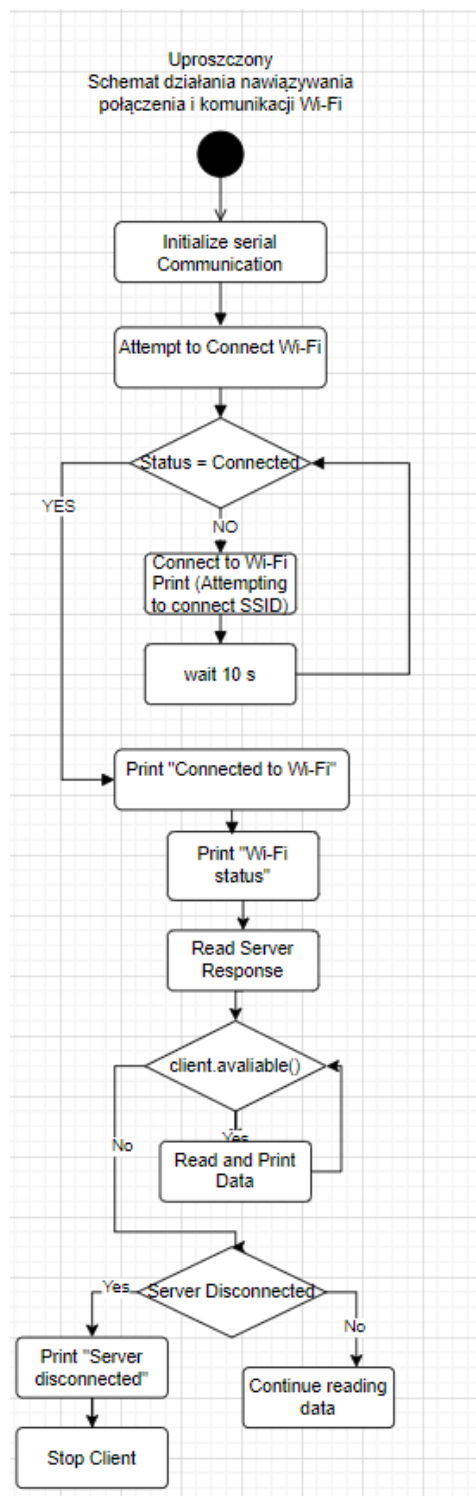
jest automatyczny, opiera się na czasowym ustawianiu rolet. Użytkownik ma do wyboru podniesienie lub opuszczenie rolety o daną wartość. W roletach dzień - noc dużą zaletą jest wiele możliwości ustawień, co również zostało uwzględnione w pracy. Dzięki temu użytkownik jest w stanie dodatkowo ustawić różne położenia pośrednie.

3.11 Skrócony opis działania komunikacji Wi-Fi

Poniższy schemat przedstawia kroki jakie wykonuje mikrokontroler by nawiązać połączenie, oraz jak wygląda sama komunikacja.

Ważne jest to że mikrokontroler powinien być początkowo podpięty do komputera, dzięki czemu po nawiązaniu połączenia otrzymamy informacje zwrotną z adresem, za pomocą którego możemy się połączyć z aplikacją webową.

Sam proces rozpoczyna się od inicjalizacji komunikacji szeregowej, przechodzi przez próbę połączenia z siecią Wi-Fi, sprawdzenie statusu połączenia, łączenie z Wi-Fi, drukowanie statusu, a w przypadku niepowodzenia, czeka i podejmuje ponowną próbę.



Rys. 13 Schemat działania komunikacji Wi-Fi

4. Realizacja

4.1. Struktura projektu i repozytorium.

Całość projektu znajduje się w repozytorium pod linkiem:

<https://github.com/janek252/Day-Night-Roller>

Główne foldery w projekcie to:

- src – folder źródłowy, w nim znajdują się biblioteki, wykorzystywane w projekcie, sam kod źródłowy.
- utilities – folder zawierający pliki pomocnicze, w przypadku tego projektu program doszukania urządzeń podłączonych do danej magistrali i2c.
- documentation – folder zawierający plik Word z dokumentacją, oraz plik video prezentujący działanie projektu.

4.2. Schemat połączeń

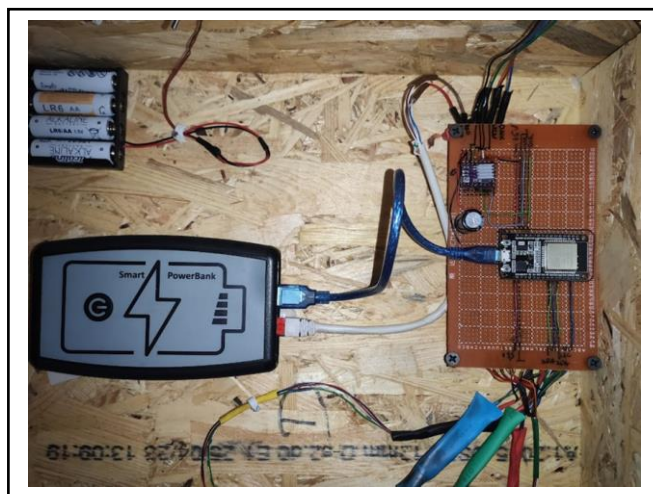
Projekt opiera się na mikrokontrolerze ESP32S. Jak widać na schemacie połączeń Rys. 14, mikrokontroler jest w centrum i wszystkie elementy są do niego podpięte. Sam układ odpowiada, za komunikację Wi-Fi z użytkownikiem, sterowanie silnikiem, zasilanie mniejszych elementów oraz odczyt z sensorów. Dokładniejsze opisy połączeń będą opisane przy poszczególnych elementach.

transoptorów szczelinowych oraz jest podane na pin FLT na sterowniku silnika, ponieważ stan niski (logiczne zero) sygnalizuje możliwe nieprawidłowości.

4.2.2. Sposób podłączenia sieci sensorów

Wszystkie wykorzystywane sensory w projekcie zostały podpięte do magistrali I²C, linia SCL jest za pomocą niebieskich przewodów, a linia SDA zielonych. W projekcie wykorzystywane jest 5 sensorów: 2 odpowiadające za pomiar temperatury oraz 3 służące do pomiaru natężenia światła. Sensory BMP120 oraz BMP280 posiadają zbliżone parametry, jednak mają różne adresy, dzięki czemu możliwe jest umieszczenie ich na jednej magistrali. Natomiast sensory natężenia światła BH1750, mają możliwość zmiany adresu, dzięki czemu możliwe jest podłączenie dwóch takich samych sensorów na jednej magistrali.

Jak można zaobserwować na zdjęciu Rys. 15, w celu uniknięcia pomyłek oraz zwiększenia dokładności pomiaru, zastosowane zostały odpowiednie kolory przewodów oraz połączenia lutowane za pomocą uniwersalnej płytki PCB wraz z miedzianymi punktami lutowniczymi. Możliwe było również wyprowadzenie przewodów wykorzystując zwieranie punktów lutowniczych.



Rys. 15 Zdjęcie połączeń elementów w zaprojektowanym układzie

4.2.3. Sposób podłączenia silnika krokowego

Silnik Krokowy został podłączony do mikroprocesora ESP32S za pomocą sterownika silników krokowych DRV8825, silnik ma wyprowadzone 4 przewody, odpowiadające za uzwojenia, są to kolejno czarny, zielony, czerwony oraz niebieski przewód, są one podpięte do sterownika do wejść idących po kolei 2A, 1A, 1B, 2B. Sterownik, jak wcześniej zostało to opisane, jest zasilony z SmartPowerbanka oraz ma doprowadzone napięcie 5 V z mikrokontrolera. Układ sterownika posiada również dwa wejścia DIR oraz STP. Są one

podpięte do portów D13 oraz D12. Wyprowadzenia SLP oraz RST, są ze sobą zwarte, dzięki czemu mają stan niski, by sterownik sam się nie resetował lub wprowadzał w tryb uśpienia. Wyprowadzenia M0-M2 służą do wykorzystania półkroków, które w projekcie nie są potrzebne.

4.3. Podłączenie silnika krokowego do rolety

Silnik krokowy JK42HS40 został bezpośrednio podłączony do wału rolety D-N. Taka roleta wewnątrz miała mechanizmy zabezpieczające przed cofaniem się oraz umożliwiające pracę za pomocą sznurka. Zostały one zablokowane, a w środku zamontowano silnik. Sposób montażu silnika polegał na tym, że oś silnika mająca średnicę 22 mm, została odpowiednio sfrezowana z jednej strony do wartości 16 mm, co umożliwiło mocne zamocowanie go wewnątrz wału rolety, w taki sposób by obracał cały wał.

Otwór o odpowiednim wymiarze w plastikowym wnętrzu wału rolety został zrobiony za pomocą pręta, o takim samym wymiarze jak oś silnika. Został on nagrany do wysokiej temperatury oraz wtopiony w plastik, który po wyciągnięciu nabrał odpowiedniego kształtu, w który wpasował się silnik.

Roleta została przyczepiona do ścianki za pomocą zdejmowalnych zaczepów, dzięki czemu jest łatwo demontowalna.



Rys. 16 Silnik krokowy zamontowany na końcu rolety

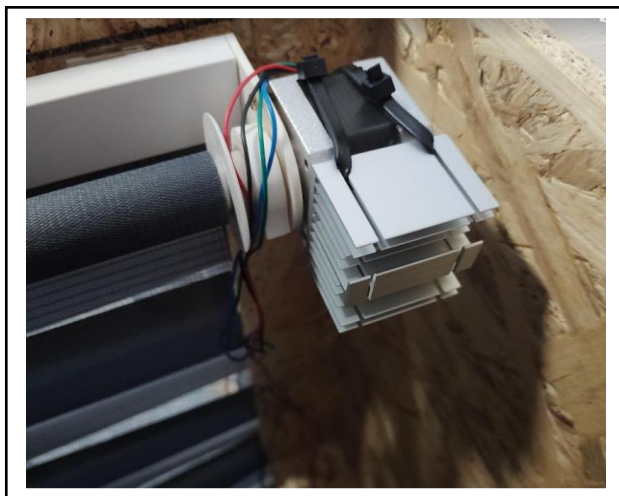
Silnik ma wystarczający moment obrotowy, który umożliwia sterownie roletą bez potrzeby zastosowania przekładni. Dodatkowo, gdy silnik jest podpięty do zasilania, jest w trybie blokującym obrót, dzięki czemu roleta sama się nie obróci.

4.4. Problemy termiczne w projekcie

Zarówno silnik, jak i sterownik silnika wytwarzają ciepło w trakcie pracy. Jak mówi producent sterownik silnika może pozwolić na sterowanie silnikiem o poborze prądu do 1,5 A, jednak, gdy założy się odpowiednie chłodzenie, sterownik może pozwolić na pobór prądu do 2,2 A na cewkę. Silnik użyty w projekcie ma pobór 0,4 A na cewkę, więc mógłby być obsługiwany, nawet bez użycia radiatora na sterowniku.

Silniki krokowe, a w szczególności bipolarne czteroparowe, będąc nawet w stanie nieobciążonym, pobierają prąd znamionowy, przez co się nagrzewają, nawet do temperatur rzędu 70°C. W celu zminimalizowania nagrzewania można zastosować ograniczenie prądowe, umożliwiające redukcję prądu, gdy silnik jest w stanie spoczynku lub dobrym sposobem odprowadzenia ciepła może być radiator.

W projekcie do chłodzenia grzejących się elementów zastosowano dwa radiatory: jeden mniejszy o wymiarach 0,8x0,8x0,5 cm do sterownika silnika, a drugi większy o wymiarach 4,4x4,3 cm do silnika. Pomiędzy radiator a element, do którego jest podpięty, umieszczono termopady i pastę termoprzewodzącą.



Rys. 17 Silnik z przymocowanym radiatorem

4.5. Algorytm sterowania silnikiem

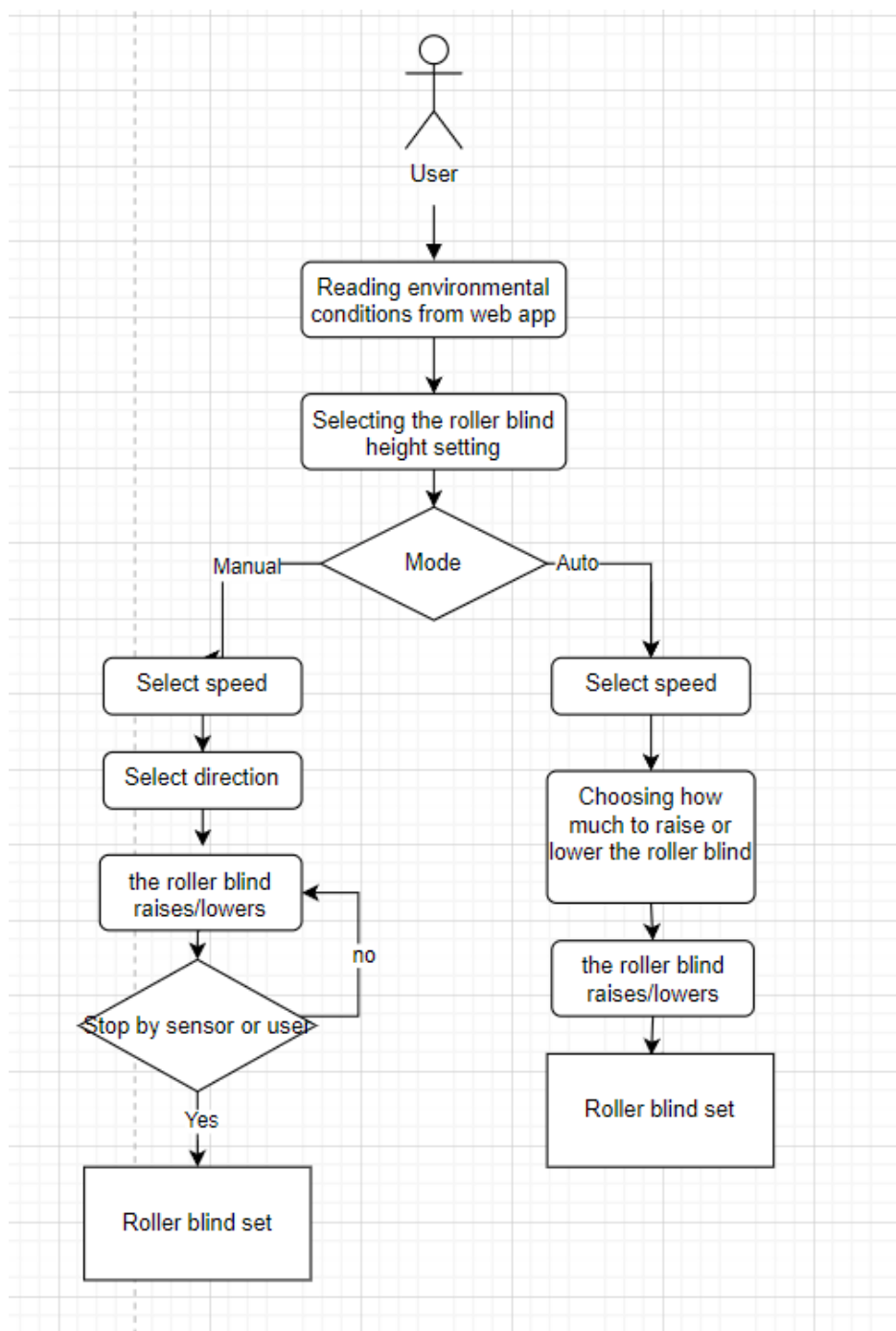
Algorytm sterowania silnikiem krokowym zaczyna się od zdefiniowania dwóch istotnych pinów: DIR (Direction - Kierunek) i STP (Step - Krok). Odpowiadają one odpowiednio za ustalanie kierunku obrotu silnika oraz generowanie impulsów krokowych, co skutkuje jego ruchem. Te piny są wcześniej zdefiniowane w pętli setup jako wyjścia, co umożliwia ich użycie do sterowania silnikiem.

Dodatkowo, w algorytmie ustalono parametr MAX_SPEED, określający maksymalną prędkość silnika. Wartość ta została ustawiona na 750, co zostało dobrane tak, aby silnik utrzymywał precyzyjny ruch bez utraty kroków. Parametry kontrolujące obrót silnika są domyślnie ustawione na 0, co uniemożliwia samoczynne uruchamianie silnika.

Roletę kontrolują dwa sensory szczelinowe, jeden u góry, drugi u dołu. Gdy roleta jest zbyt nisko lub zbyt wysoko, czujniki informują mikrokontroler, który zatrzymuje silnik, a następnie wysyła sygnał do obrócenia go krótko w przeciwnym kierunku. Działa to jako zabezpieczenie przed uszkodzeniem silnika lub nawinięciem rolety w niewłaściwą stronę.

Kontrola kierunku obrotu odbywa się poprzez ustawianie pinu DIR w stan wysoki (HIGH) lub niski (LOW). Gdy pin DIR jest ustawiony na HIGH, silnik obraca się w jednym kierunku, natomiast ustawienie na LOW powoduje obrót w przeciwnym kierunku. W ten sposób algorytm zapewnia podstawową funkcjonalność sterowania silnikiem krokowym, umożliwiając precyzyjną kontrolę nad kierunkiem ruchu i prędkością.

Jak można zaobserwować na rysunku 18, użytkownik ma do wyboru dwa tryby sterowania silnikiem, automatyczny i manualny. Tryb manualny opiera się na 3 funkcjach, odpowiadających za obrót silnika zgodnie ze wskazówkami zegara, przeciwnie oraz zatrzymanie silnika.



Rys. 18 Graficzne przedstawienie algorytmu sterowania roletami

Użytkownik w trybie manualnym może wybrać wartość prędkości poprzez przyciski od 1 do 4 oraz ustawić kierunek, gdy roleta ustawi się w odpowiedniej pozycji, może zatrzymać ją poprzez przycisk stop lub poczekać, aż zostanie ona zatrzymana poprzez interwencję czujnika szczelinowego.

Tryb automatyczny, jak można zaobserwować na zdjęciu Rys. 19, oparty jest na czasowych funkcjach, polegających na tym, że silnik obraca się w danym kierunku przez określony czas, takim sposobem jednym przyciskiem, użytkownik może zjechać roletą na sam

dół, zatrzymać ją w połowie lub ustawić roletę w ustawienie $\frac{3}{4}$. W tym przypadku zalecana prędkość to jest 1 oraz w programie jest zabezpieczenie, by użytkownik nie kliknął dwa razy przycisku maksymalnego podniesienia. Polega ono na tym, że po naciśnięciu przycisk podświetla się na czerwono, informując użytkownika by nie naciskał go drugi raz.

```
327 //////////////////////////////////////////////////
328 // Funkcje czasowe
329 //////////////////////////////////////////////////
330 void fulllowerRollerBlind() {
331     digitalWrite(DIR, LOW);
332     motorRunning = true;
333     tone(STP, motorSpeed);
334     delay(20000);
335     noTone(STP);
336     motorRunning = false;
337 }
338
339 void fullraiseRollerBlind() {
340     digitalWrite(DIR, HIGH);
341     motorRunning = true;
342     tone(STP, motorSpeed);
343     delay(20000);
344     noTone(STP);
345     motorRunning = false;
346 }
```

Rys. 119 Fragment kodu odpowiedzialnego za automatyczne czasowe ustawienie rolet

4.6. Aplikacja WEB

Aplikacja webowa tworzy serwer umożliwiający odczytanie wartości z sensorów oraz sterowanie silnikiem. Została napisana przy użyciu biblioteki WiFi.h. Mikrokontroler generuje stronę o adresie zgodnym z przyznanym mu adresem przez domowy router. Strona składa się z dwóch głównych sekcji.

Górna część strony odpowiada za prezentację danych z pięciu sensorów, z których każdy podłączony jest do jednej z dwóch magistral I²C. Sama strona działa tak, że użytkownik musi ręcznie odświeżyć stronę, aby uzyskać aktualne dane z sensorów. Taka konfiguracja pozwala na łatwe pobranie bieżących danych w celu ich zapisu.

Dolna część strony pełni funkcję sterowania silnikiem. Na dole znajduje się sterowanie manualne, które opiera się na wyborze prędkości i kierunku obrotu. Wyżej znajduje się sterowanie automatyczne z przyciskami do ośmiu różnych ustawień, co pozwala użytkownikowi wybrać pożądaną pozycję rolety. Po naciśnięciu przycisków zmieniają one kolor na czerwony, informując użytkownika, aby nie naciskał ich przypadkowo więcej niż raz, w tym celu wykorzystywany jest specjalny skrypt.

Light Sensor Data

Sensor BH1750 Inside

Light Level: 193.33 lx

Sensor BMP280

Temperature: 23.25 °C

Sensor BH1750 Inside

Light Level: 190.00 lx

Sensor BH1750 outside

Light Level: 405.00 lx

Sensor BMP180

Temperature: 22.20 °C

Roller Blind Control

Full Lower	Full Raise	Half Lower	Half Raise	1/4 Lower	3/4 Lower	1/4 Raise	3/4 Raise
------------	------------	------------	------------	-----------	-----------	-----------	-----------

Motor Control

Rotate Clockwise	Rotate Counterclockwise	Stop Motor
Speed 1	Speed 2	Speed 3
Speed 4		

Current Motor Speed: 0

Rys. 20 Widok aplikacji

4.7. Wykorzystane biblioteki

W projekcie wykorzystano biblioteki z ArduinoIDE, są to:

- Adafruit_BMP085_Library – biblioteka do obsługi sensora BMP-85.
- Adafruit_BMP280_Library — biblioteka do obsługi sensora BMP-280.
- Stepper – biblioteka do obsługi silnika krokowego.
- WiFi-h – biblioteka dla ESP32 do obsługi komunikacji Wi-Fi.
- BH1750 - – biblioteka do obsługi sensora BH-1750.

5. Przeprowadzone Testy

5.1. Testy dla aplikacji

Test został przeprowadzony w celu sprawdzenia aplikacji webowej pod kątem łatwości sterowania przez użytkownika oraz przetestowania poprawności działania projektu.

Przykładowy użytkownik otrzymywał dostęp do makiety oraz urządzenia umożliwiającego przetestowanie możliwości całego projektu. Test został przeprowadzony w temperaturze pokojowej, a makieta była ustawiona blisko okna.

Najpierw przetestowano warunki połączenia się z aplikacją, użytkownik po włączeniu Powerbanka służącego do zasilenia projektu, połączył się z aplikacją webową poprzez znany wcześniej adres IP. Użytkownikowi udało się bez większych problemów w prawidłowy sposób odczytać warunki środowiskowe za pomocą aplikacji i sieci sensorów, oraz przetestować działanie ustawiania rolety. Przetestowano zarówno tryb automatyczny, jak i manualny dla kilku różnych ustawień.

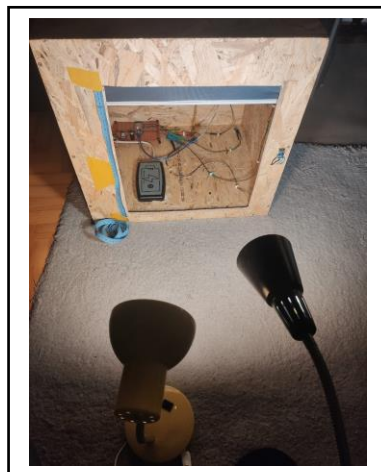
Zrealizowane testy pozwalają stwierdzić, że aplikacja działa w prawidłowy sposób, ma interfejs przyjazny dla użytkownika, dzięki czemu bez większych problemów udało się odczytać wartości z sensorów oraz przetestować różne ustawienia rolety.

5.2. Pomiary natężenia światła

Celem testu jest sprawdzenie zależności między odsunięciem rolety, a zmianą natężenia światła w pomieszczeniu.

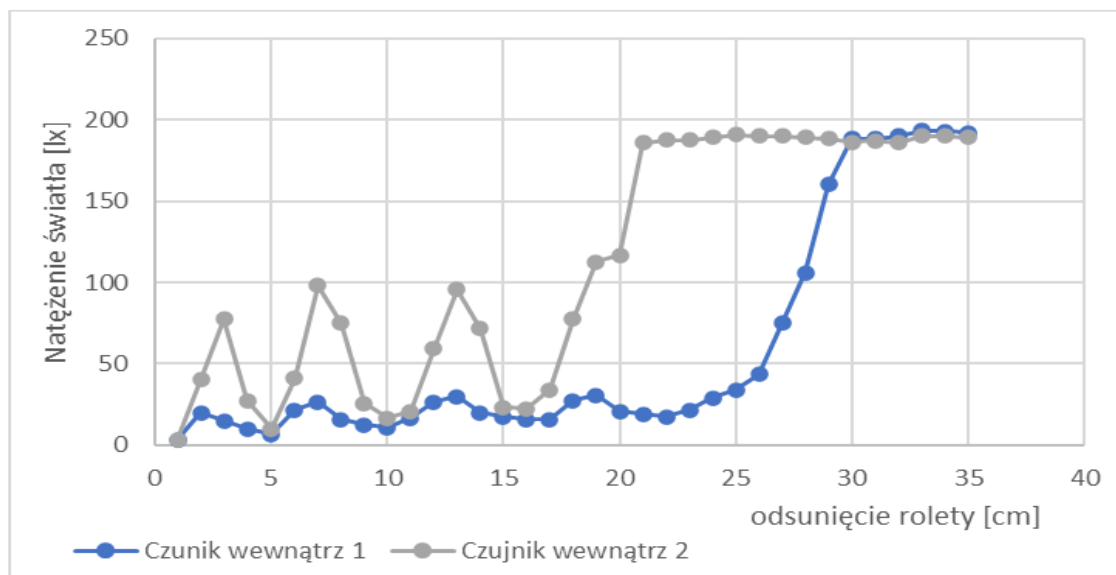
Pomiar został przeprowadzony w temperaturze pokojowej, zgodnie z warunkami przedstawionymi na Rys. 29. Do oświetlenia stanowiska użyto dwóch lamp: jednej emitującej ciepłe światło, a drugiej z zimnym światłem. Dodatkowe oświetlenie, poza lampami, było wyłączone.

Obok rolety przyczepiono metr, za pomocą którego istniała możliwość oceny rozwinięcia rolety. Do pomiaru temperatury wykorzystano 3 sensory natężenia światła BH1750, dwa z nich zamontowane wewnątrz testowego pomieszczenia, sensor numer 2 był zamontowany trochę niżej na przeciwnej do rolety ścianie, a sensor 1 wyżej oraz jeden sensor został zamontowany na zewnątrz w celach porównawczych.



Rys. 21 Stanowisko pomiarowe natężenia światła

Na Rys. 30 widoczne jest, że unoszenie rolety D-N powoduje sinusoidalny wzrost natężenia światła. Gdy roleta się podnosi, zwiększa się dostęp światła, ale jednocześnie odsłaniają się i zasłaniają kolejne paski, co powoduje cykliczne zmiany jasności w pomieszczeniu. Po przekroczeniu poziomego czujnika, natężenie światła gwałtownie rośnie do maksimum. Wartości natężenia światła na zewnątrz wynoszą około 400-410 lx, ale mogą być obarczone błędami pomiarowymi i niedokładnościami sensorów.



Rys. 22 Wykres zależności natężenia światła [lx] od odsunięcia rolety [cm]

Przeprowadzone testy wykazały, że natężenie światła wewnątrz pomieszczenia, w miarę podnoszenia rolet, początkowo wzrastało w sposób zbliżony do sinusoidalnego. Zjawisko to wynika z faktu, że w miarę unoszenia rolety, kolejne przeźroczyste i gęste paski odkrywały się i zasłaniały, co skutkowało zmianami w ilości wpadającego światła. Nierówności na

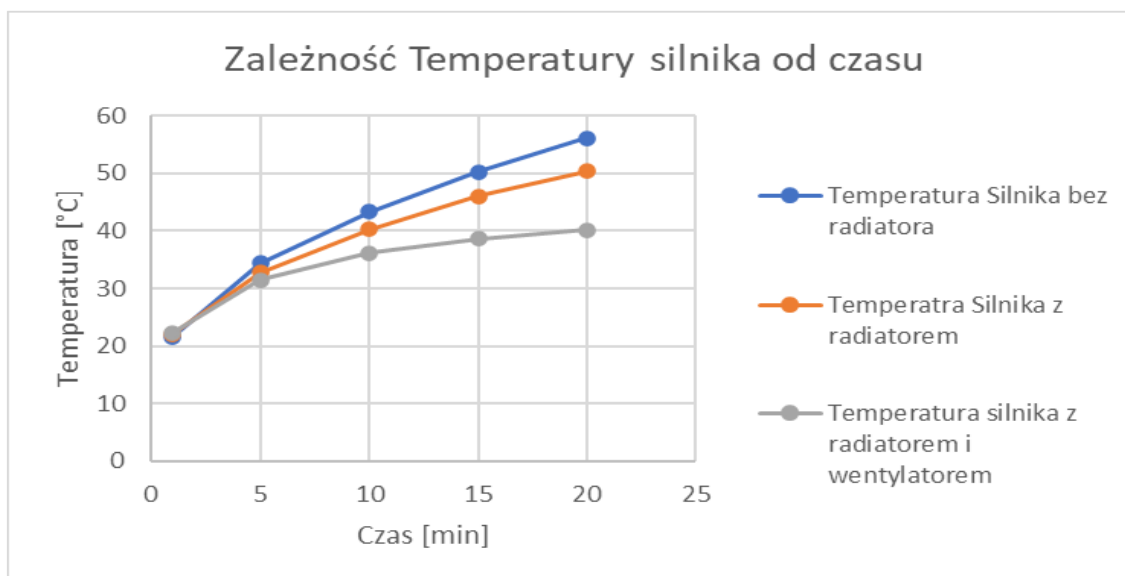
wykresach mogą być spowodowane charakterystyką czujnika BH1750 oraz niedokładnościami pomiarowymi podczas oceny stopnia podniesienia rolety. Po odkryciu rolety, czujnik wewnątrz pomieszczenia wskazywał wartość maksymalną.

5.3. Pomiary temperatur silnika krokowego

Celem pomiaru było ocenienie potrzeby zastosowania radiatora w celu kontrolowania temperatury silnika krokowego podczas postoju. Wysoka temperatura silnika może prowadzić do przegrzania, co z kolei może skutkować uszkodzeniem układu sterowania.

Temperatura silnika krokowego była badana w temperaturze pokojowej (21°C) w 4 pomiarach co 5 minut za pomocą termometru bezdotykowego Microlife NC-150, działającego na zasadzie pomiaru podczerwieni (dokładność $\pm 2^{\circ}\text{C}$). Bez radiatora silnik osiągał nawet 56°C po 20 minutach postoju, natomiast z zastosowaniem radiatora temperatura była niższa o $5,7^{\circ}\text{C}$, a nawet o 16°C przy dodatkowym wentylatorze.

Wykres na Rys. 31 ukazuje, że nagrzewanie silnika jest bardziej liniowe bez radiatora, a zastosowanie radiatora i wentylatora skutecznie ogranicza wzrost temperatury.



Rys. 23 Wykres zależności zmian temperatury silnika od czasu

6. Podsumowanie

Celem projektu było zaprojektowanie i wykonanie układu sterowania roletą okienną typu dzień-noc, z siecią sensorów informującą użytkownika o warunkach środowiskowych. Umożliwiono zdalną kontrolę stopnia rozwinięcia rolety. Przeprowadzono testy aplikacji webowej, sensorów i temperatury silnika, potwierdzając skuteczność interfejsu i działania radiatora z wentylatorem.

Całkowity koszt projektu był umiarkowany dzięki ekonomicznemu mikrokontrolerowi. Można było użyć tańszych komponentów, ale kosztem funkcjonalności. Projekt zużywa średnio dużo energii, głównie przez silnik, który można wyłączać w okresach bezczynności.

Główną zaletą projektu jest prostota obsługi i uniwersalność, z łatwą możliwością rozszerzeń.

Propozycje modyfikacji:

1. Usprawnienie algorytmu sterowania silnikiem: Dodanie większej liczby czujników położenia pozwoli na dokładniejsze sterowanie roletą, lepiej dostosowując jej pozycję do oczekiwań użytkownika.
2. Rozbudowa wyświetlania parametrów: Dodanie wykresów zmian temperatury i natężenia światła w aplikacji webowej zapewni pełniejszą kontrolę nad monitorowanym środowiskiem.
3. Poprawa komunikacji: Zastąpienie obecnego mechanizmu komunikacji dedykowaną aplikacją ułatwi kontrolę i integrację z systemami smart home, centralizując wszystkie funkcje w jednym miejscu.
4. Poprawa bezpieczeństwa: Wprowadzenie dodatkowej weryfikacji użytkowników zwiększy bezpieczeństwo, chroniąc system przed nieautoryzowanym dostępem w przypadku włamania do sieci domowej.