



**Politechnika  
Śląska**

## **PROJEKT INŻYNIERSKI**

System wizualizacji danych webowego interfejsu Maszyny W z użyciem  
mikrokontrolera ESP32

**Bartosz FARUGA**

Nr albumu: 305985

**Kierunek:** Informatyka

**Specjalność:** Bazy danych i inżynieria systemów

**PROWADZĄCY PRACĘ**

**dr. inż. Tomasz Rudnicki**

**KATEDRA SYSTEMÓW CYFROWYCH**

**Wydział Automatyki, Elektroniki i Informatyki**

**OPIEKUN, PROMOTOR POMOCNICZY**

**⟨stopień naukowy imię i nazwisko⟩**

**Gliwice 2025**



**Tytuł pracy**

System wizualizacji danych webowego interfejsu Maszyny W z użyciem mikrokontrolera ESP32

**Streszczenie**

(Streszczenie pracy – odpowiednie pole w systemie APD powinno zawierać kopię tego streszczenia.)

**Słowa kluczowe**

(2-5 słów (fraz) kluczowych, oddzielonych przecinkami)

**Thesis title**

Thesis title in English

**Abstract**

(Thesis abstract – to be copied into an appropriate field during an electronic submission – in English.)

**Key words**

(2-5 keywords, separated by commas)



# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
1.1	Cel pracy . . . . .	1
1.2	Analiza rozwiązań literaturowych . . . . .	1
<b>2</b>	<b>Przykładowa realizacja systemu wizualizacji</b>	<b>3</b>
2.1	Założenia realizacyjne . . . . .	3
2.2	Schemat blokowy . . . . .	4
<b>3</b>	<b>Część projektowa</b>	<b>5</b>
3.1	System mikroprocesorowy . . . . .	5
3.2	Adresowalne diody LED . . . . .	7
3.3	Przyciski . . . . .	8
3.4	Podświetlenie LED . . . . .	10
3.5	Enkoder . . . . .	10
3.6	Przełącznik trybów . . . . .	11
3.7	Serwer . . . . .	12
3.8	Punkt dostępowy . . . . .	12
3.9	Strona internetowa . . . . .	12
3.10	Interfejs komunikacyjny . . . . .	12
3.11	Zasilanie . . . . .	12
<b>4</b>	<b>Schematy bloków systemu wizualizacji</b>	<b>13</b>
4.1	Płyta główna SWIM . . . . .	13
4.2	Wyświetlacz siedmiosegmentowy trzycyfrowy . . . . .	15
4.3	Wyświetlacz siedmiosegmentowy dwucyfrowy . . . . .	16
4.4	Połączenia pomiędzy blokami . . . . .	17
<b>5</b>	<b>Część konstrukcyjna</b>	<b>19</b>
5.1	Obwody drukowane . . . . .	19
5.1.1	Płyta główna . . . . .	19
5.1.2	Wyświetlacze siedmiosegmentowe . . . . .	21

5.2	Opis złącz . . . . .	22
5.2.1	Płyta główna . . . . .	22
5.2.2	Wyświetlacze segmentowe . . . . .	23
<b>6</b>	<b>Opis oprogramowania</b>	<b>25</b>
6.1	Program mikrokontrolera . . . . .	25
6.1.1	Obsługa linii adresowalnych diod LED . . . . .	26
6.1.2	Obsługa przycisków, enkodera oraz przełącznika trybów . . . . .	27
6.1.3	Obsługa pamięci nieulotnej/systemu plików . . . . .	27
6.1.4	Obsługa trybu sieciowego SWIM . . . . .	27
6.1.5	Obsługa trybu lokalnego SWIM . . . . .	27
<b>7</b>	<b>Obsługa</b>	<b>29</b>
	<b>Bibliografia</b>	<b>31</b>
	<b>Spis skrótów i symboli</b>	<b>33</b>
	<b>Źródła</b>	<b>35</b>
	<b>Lista dodatkowych plików, uzupełniających tekst pracy</b>	<b>37</b>
	<b>Spis rysunków</b>	<b>39</b>
	<b>Spis tabel</b>	<b>41</b>

# Rozdział 1

## Wstęp

### 1.1 Cel pracy

Celem niniejszej pracy jest opracowanie systemu wizualizacji danych pochodzących z webowego interfejsu Maszyny W z wykorzystaniem mikrokontrolera ESP32-S3.

Opracowane w ramach pracy dyplomowej urządzenie umożliwia odwzorowanie w czasie rzeczywistym wartości liczbowych oraz sygnałów sterujących symulatorem Maszyny W na adresowalnych diodach RGB. Ponadto system pozwala na zdalne sterowanie funkcjami symulatora zarówno za pomocą przycisków fizycznych, jak i interfejsu internetowego. System obsługuje hosting interfejsu webowego bezpośrednio na mikrokontrolerze oraz zapewnia komunikację pomiędzy ESP32 a interfejsem. Wykorzystany w tym celu protokół WebSocket umożliwia natychmiastową synchronizację zmian między stroną internetową a wyświetlaczem, co pozwala użytkownikowi na intuicyjny i interaktywny monitoring oraz kontrolę stanu Maszyny W.

### 1.2 Analiza rozwiązań literaturowych





# Rozdział 2

## Przykładowa realizacja systemu wizualizacji

### 2.1 Założenia realizacyjne

Pierwszym etapem budowy systemu wizualizacji danych jest projektowanie, które powinno być poprzedzone przemyśleniami dotyczącymi możliwych realizacji oraz wyborem tych, które są optymalne i spełniają przyjęte założenia. Założenia dodatkowe nie są bezwzględnie wymagane, ale mogą poszerzyć tworzone rozwiązanie o alternatywne sposoby działania.

Założenia podstawowe:

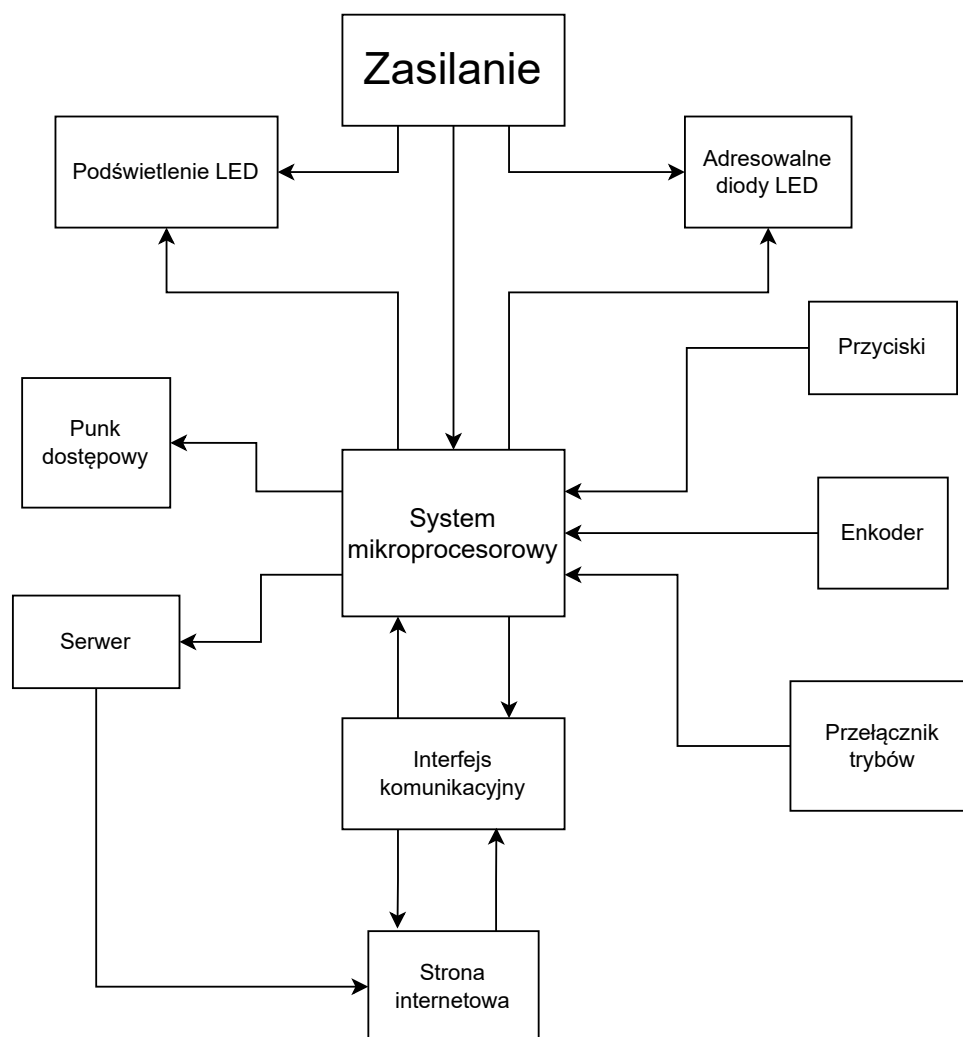
- dedykowany układ dla mikrokontrolera z odpowiednimi wyprowadzeniami,
- dedykowany układ dla LED-owych wyświetlaczy siedmiosegmentowych,
- konstrukcja kompletnego układu elektronicznego z układów mikrokontrolera, wyświetlaczy LED oraz przycisków,
- hosting strony internetowej bezpośrednio na urządzeniu wraz z przesyłem danych pomiędzy urządzeniem a stroną,
- wizualizacja informacji pobieranych ze strony internetowej na adresowalnych diodach LED,
- przesył informacji o wciśniętych przyciskach do strony internetowej,
- obsługa podświetlenia modelu Maszyny W diodami LED.

Założenia dodatkowe:

- lokalna symulacja Maszyny W na mikrokontrolerze,
- obsługa enkodera do wprowadzania wartości liczbowych.

## 2.2 Schemat blokowy

Na rysunku poniżej (Rys. 2.1) przedstawiono przykładowy schemat blokowy systemu wizualizacji interfejsu maszyny (SWIM). Schemat uwzględnia zarówno bloki spełniające założenia podstawowe, jak i dodatkowe.



Rysunek 2.1: Schemat blokowy SWIM

# Rozdział 3

## Część projektowa

### 3.1 System mikroprocesorowy

Ze względu na wymaganą komunikację sieciową projektu, wysoką złożoność rozwiązania oraz zapotrzebowanie na wiele portów wejścia/wyjścia należało wybrać system mikroprocesorowy zapewniający poniższe możliwości:

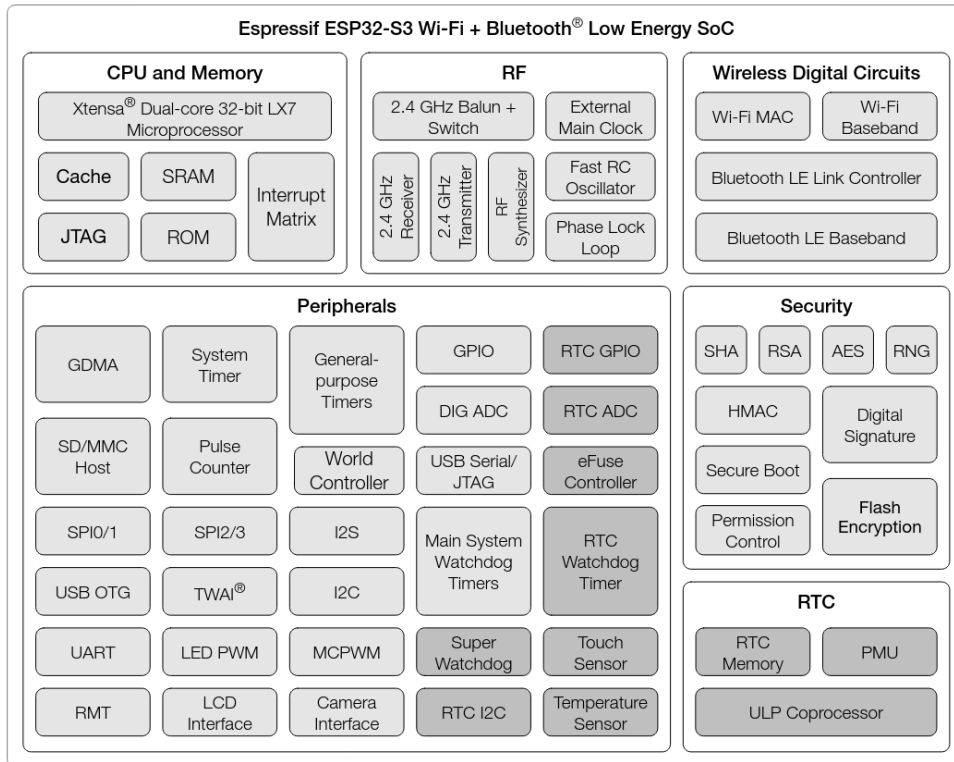
- obsługa dwóch kanałów dla adresowalnych diod LED RGB,
- obsługa 17 przycisków wraz z enkoderem,
- zapewnienie punktu dostępu internetu,
- przechowywanie danych w pamięci nieulotnej,
- serwowanie strony internetowej.

Zwracając uwagę na powyżej wymienione wymagania rozważono dwie możliwości: mikrokomputer z rodziny *Raspberry Pi* lub mikrokontroler z rodziny *ESP32*. Zarówno mikrokomputer jak i mikrokontroler spełniają wszystkie ww. założenia. Urządzenia z rodziny *Raspberry Pi* są proste w użytku, posiadają system operacyjny oparty na jądrze *Linux'a* oraz pozwalają na programowanie w językach *Python* oraz *C++*. System operacyjny pozwala również na łatwe przechowywanie danych w pamięci nieulotnej urządzenia z uwagi na obecny w nim system plików. Obecny na urządzeniach *Raspberry Pi* system operacyjny *Raspbian* będący pochodną popularnej dystrybucji *Debian* pozwala na tworzenie własnych *daemonów* (procesy działające w tle).

Mikrokontrolery z rodziny *ESP32* nie posiadają systemu operacyjnego, a ich programowanie jest możliwe przy użyciu języków takich jak *C* oraz *C++*. Służą do tego specjalnie przygotowane środowiska programistyczne takie jak *Visual Studio Code* z rozszerzeniem *PlatformIO* a także *Arduino IDE*. Ich przewagą nad mikrokomputerami są: brak systemu operacyjnego, niższa cena, mniejsza złożoność oraz mniejsze zużycie energii.

Po rozważeniu obu opcji zdecydowano się na użycie mikrokontrolera z rodziny ESP32. Wybór został podyktowany pozwalającym na pełną kontrolę urządzenia językiem *C++*, niższą ceną, brakiem systemu operacyjnego, dostępnością oraz bogatą dokumentacją. Z katalogu firmy *Espressif Systems* wybrano urządzenie *ESP32-S3-WROOM-1*.

Poniższy rysunek (Rys. 3.1) z dokumentacji producenta przedstawia schemat funkcjonalny wybranego układu



Rysunek 3.1: Schemat blokowy funkcjonalny układu ESP32-S3

Własności wybranego układu zapożyczone z dokumentacji firmy *Espressif Systems*:

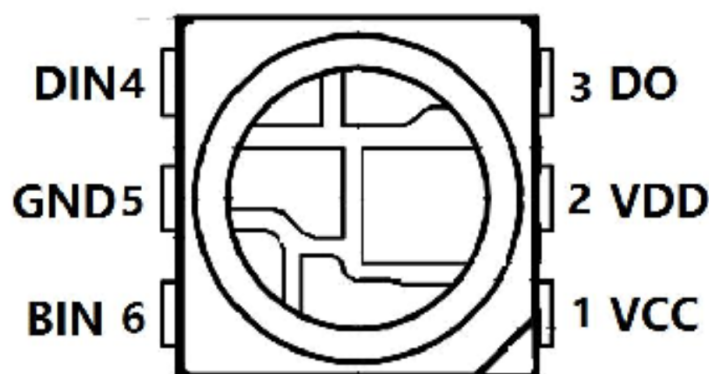
- dwurdzeniowy 32-bitowy mikroprocesor *Xtensa LX7*,
- 16MB pamięci flash, 16MB pamięci PSRAM,
- 512MB pamięci SRAM,
- 36 portów GPIO,
- komunikacja 2.4GHz WiFi (802.11b/g/n) oraz Bluetooth 5,
- wbudowana antena,
- wsparcie dla: I2C, LED PWM, USB Serial,
- maksymalny pobór prądu 300-400mA,
- zasilanie napięciem 3.3V.

## 3.2 Adresowalne diody LED

Zakładano wizualizację interfejsu Maszyny W na tablicy złożonej z diod LED RGB. Adresowalne diody LED RGB pozwalają na sekwencyjne zadanie kolorów każdej z 3 diod (czerwonej, zielonej oraz niebieskiej) elementu, połączonych w ciągłą linię. Przewidywana liczba pojedynczych diod LED podłączonych do SWIM przekracza 600. Znaczna liczba elementów może doprowadzać do wpływających na poprawne działanie układu spadków napięcia.

Spośród aktualnie dostępnych na rynku wersji elementu wyróżniły się dwa modele: *WS2812B* oraz *WS2815B* firmy *WORLDSEMI*. Podczas wstępnych testów obu elementów zaobserwowano następujące cechy. Diody *WS2812B* zasilane napięciem 5V wykazywały zauważalny spadek jakości odwzorowania kolorów przy liczbie elementów przekraczającej 100 (w ciągłej linii). Wykorzystanie ww. elementów niesłoby za sobą potrzebę dodatkowego wprowadzania napięcia do linii diod. Diody *WS2815B* zasilane napięciem 12V nie wykazały zauważalnych spadków napięcia przy tej samej liczbie elementów. Cecha ta zadecydowała o ostatecznym wyborze modelu *WS2815B-V1*, jego jedyną wadą jest nieznacznie większa cena w porównaniu do 5V odpowiednika.

Poniższy rysunek (Rys. 3.2) z dokumentacji producenta przedstawia schemat wyprowadzeń elementu.



Rysunek 3.2: Schemat wyprowadzeń adresowalnej diody LED RGB *WS2815B-V1*

Własności elementu:

- napięcie pracy 12V,
- pobór prądu 15mA przy maksymalnej jasności każdej z 3 kolorowych diod,
- montaż SMD,
- częstotliwość odświeżania 4kHz.

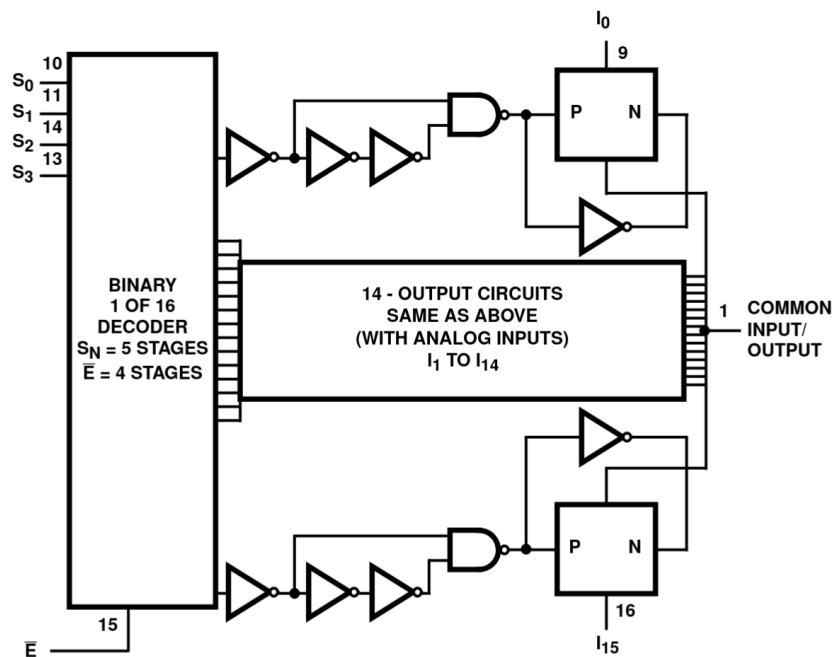
### 3.3 Przyciski

W celu zapewnienia pełnej obsługi Maszyny W potrzebne jest wykorzystanie 17 przycisków. Symulator w wersji podstawowej udostępnia użytkownikowi 16 możliwych sygnałów: *czyt, wys, wei, il, wyad, weak, pisz, przep, wel, wyl, dod, ode, wes, weja, wyak, wea*. Do wymienionych sygnałów należy również wliczyć przycisk *takt*, który zapewni możliwość wykonania zadanego rozkazu.

Zdecydowano się na wykorzystanie przełączników *MX Blue* firmy *Cherry*. Według dokumentacji producenta wyżej wymieniony przełącznik klawiaturowy pozwala na 50 milionów aktywacji. Pozwoli to na długoletnią, niezawodną działalność. Wykorzystano przyciski typu *Blue* oferujące słyszalny odgłos kliknięcia przy wciskaniu. Daje to użytkownikowi informację zwrotną o stanie przełącznika.

Aby ograniczyć liczbę potrzebnych wyprowadzeń dla przycisków zdecydowano wykorzystać multiplexer. W tym celu wybrano 16 kanałowy multiplexer z wyprowadzeniem *COM*. Urządzenie to pozwala na odczyt stanu przycisku wprowadzając na wejście multiplexera 4-bitowego sygnału. Cykliczny odczyt stanu wyprowadzenia *COM* wraz z odpowiednią kombinacją 4-bitowego wejścia pozwala na odczyt aktualnego stanu przycisków. Zmiana kombinacji 4-bitów wejścia oraz odczyt stanu wyprowadzenia *COM* co cykl mikroprocesora pozwala na odczyt aktualnego stanu przycisków. Z uwagi na niską cenę oraz niewielki wymiar wybrano model *CD74HC4067M* firmy *Texas Instruments*.

Poniższy rysunek (Rys. 3.3) z dokumentacji producenta przedstawia diagram funkcjonalny układu.



Rysunek 3.3: Diagram funkcjonalny multiplexera *CD74HC4067M*

Poniższa tabela (Tab. 3.1) prawdy przedstawia odczyt stanu przycisków

Tabela 3.1: Tablica prawdy multiplexera

<b>S0</b>	<b>S1</b>	<b>S2</b>	<b>S3</b>	<b>COM</b>	<b>Kanał</b>
0	0	0	0	1	0
1	0	0	0	1	1
0	1	0	0	1	2
1	1	0	0	1	3
0	0	1	0	1	4
1	0	1	0	1	5
0	1	1	0	1	6
1	1	1	0	1	7
0	0	0	1	1	8
1	0	0	1	1	9
0	1	0	1	1	10
1	1	0	1	1	11
0	0	1	1	1	12
1	0	1	1	1	13
0	1	1	1	1	14
1	1	1	1	1	15

## 3.4 Podświetlenie LED

Chcąc zapewnić przyjazny wizualnie wygląd modelu SWIM zdecydowano się na podświetlenie następujących elementów:

- licznik rozkazów,
- akumulator,
- rejestr instrukcji,
- rejestry A oraz S,
- logo Politechniki Śląskiej.

Wykorzystano do tego 12V linie diod LED. Aby uniknąć zauważalny dla użytkowników spadków jasności podświetlenia zdecydowano na podłączenie bezpośrednio do zasilacza 12V.

## 3.5 Enkoder

W celu zapewnienia możliwości wprowadzania wartości do: licznika rozkazów, akumulatora oraz rejestrów w lokalnym trybie działania urządzenia zastosowano enkoder obrotowy z przyciskiem.

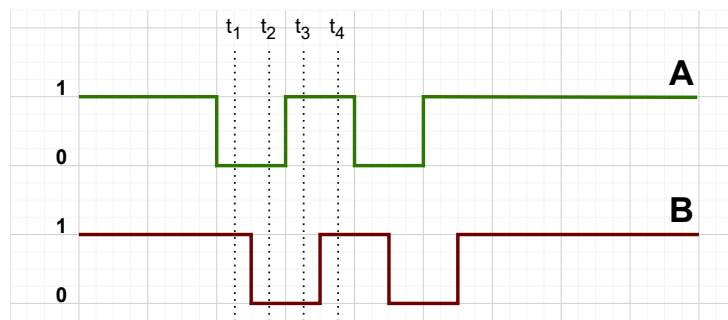
Zamierzone działanie enkodera:

- zwiększanie/zmniejszanie wartości liczbowych w zależności od kierunku obrotu,
- aktywacja trybu wprowadzania danych poprzez przytrzymanie przycisku,
- przełączanie między polami wprowadzania danych przy użyciu szybkich kliknięć.

Enkoder pozwala użytkownikowi w prosty i intuicyjny sposób sterować parametrami urządzenia bez korzystania z interfejsu sieciowego. Cykliczny odczyt stanów dwóch kanałów enkodera umożliwiał stwierdzenie strony rotacji.



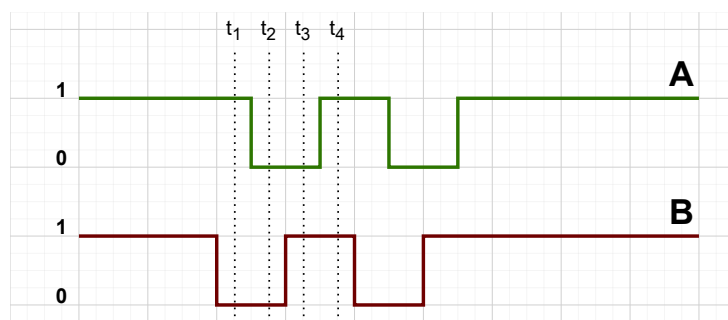
Aby określić kierunek obrotu elementu należy odczytać zmianę stanów na wyprowadzeniach urządzenia. Poniższe wykresy przedstawiają sposób odczytu obrotu w prawo (Rys. 3.4, Tab. 3.2) oraz w lewo (Rys. 3.5, Tab. 3.3).



Krok (t)	A	B
1	0	1
2	0	0
3	1	0
4	1	1

Tabela 3.2: Tablica prawdy dla obrotu w prawo

Rysunek 3.4: Wykres stanów przy obrocie w prawo



Krok (t)	A	B
1	1	0
2	0	0
3	0	1
4	1	1

Tabela 3.3: Tablica prawdy dla obrotu w lewo

Rysunek 3.5: Wykres stanów przy obrocie w lewo

## 3.6 Przełącznik trybów

Jednym z dodatkowych założeń projektu jest możliwość przełączania SWIM między trybami lokalnym oraz sieciowym. Aby zapewnić taką możliwość bezpośrednio z poziomu modelu wykorzystano przełącznik bistabilny. Podłączenie przełącznika bezpośrednio do systemu mikroprocesorowego pozwala na wykorzystanie przerwań w celu natychmiastowej zmiany trybu pracy urządzenia. Korekta drgań styków urządzenia została zapewniona z poziomu oprogramowania.

### **3.7 Serwer**

### **3.8 Punkt dostępowy**

### **3.9 Strona internetowa**

### **3.10 Interfejs komunikacyjny**

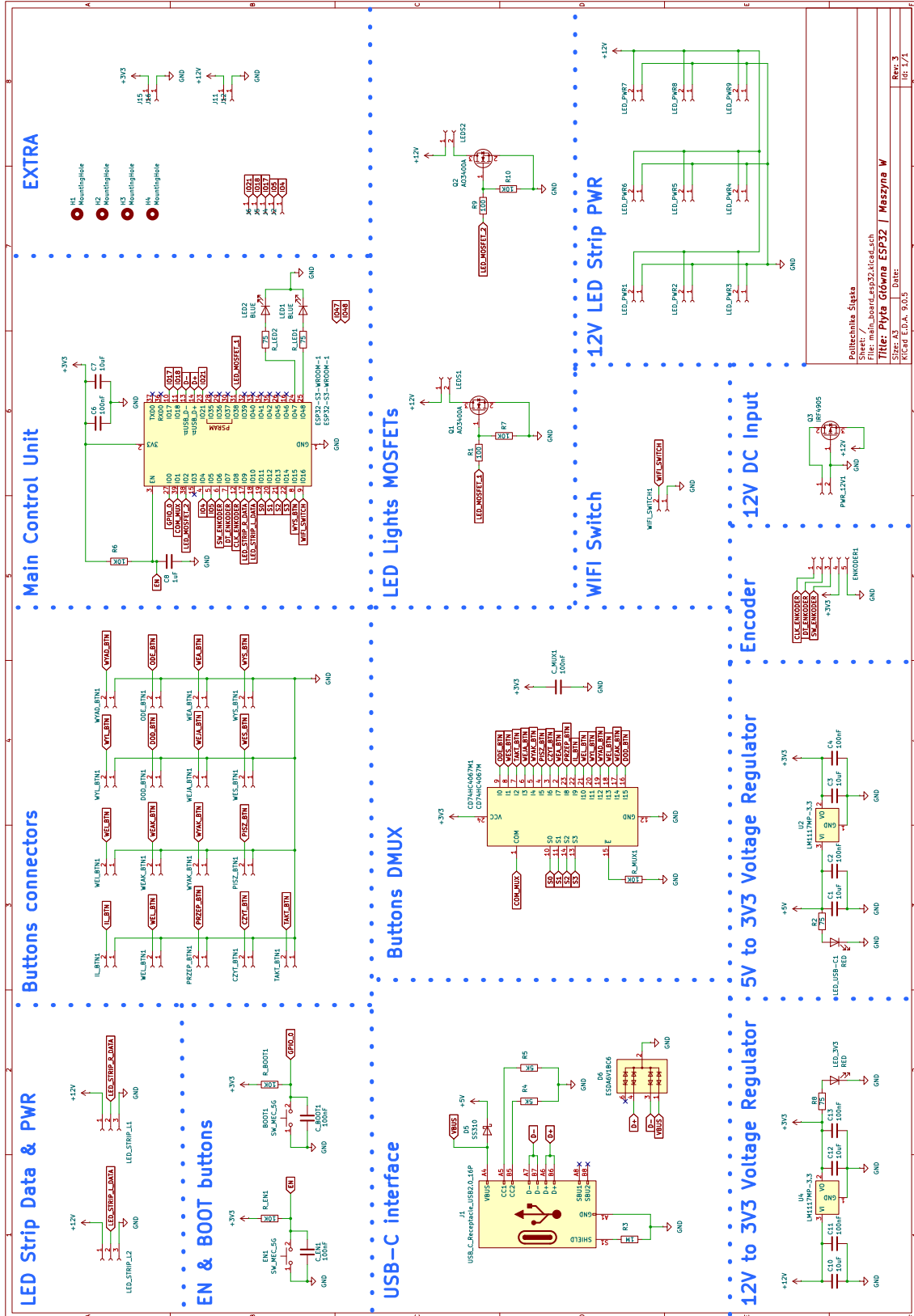
### **3.11 Zasilanie**

## Rozdział 4

# Schematy bloków systemu wizualizacji

### 4.1 Płyta główna SWIM

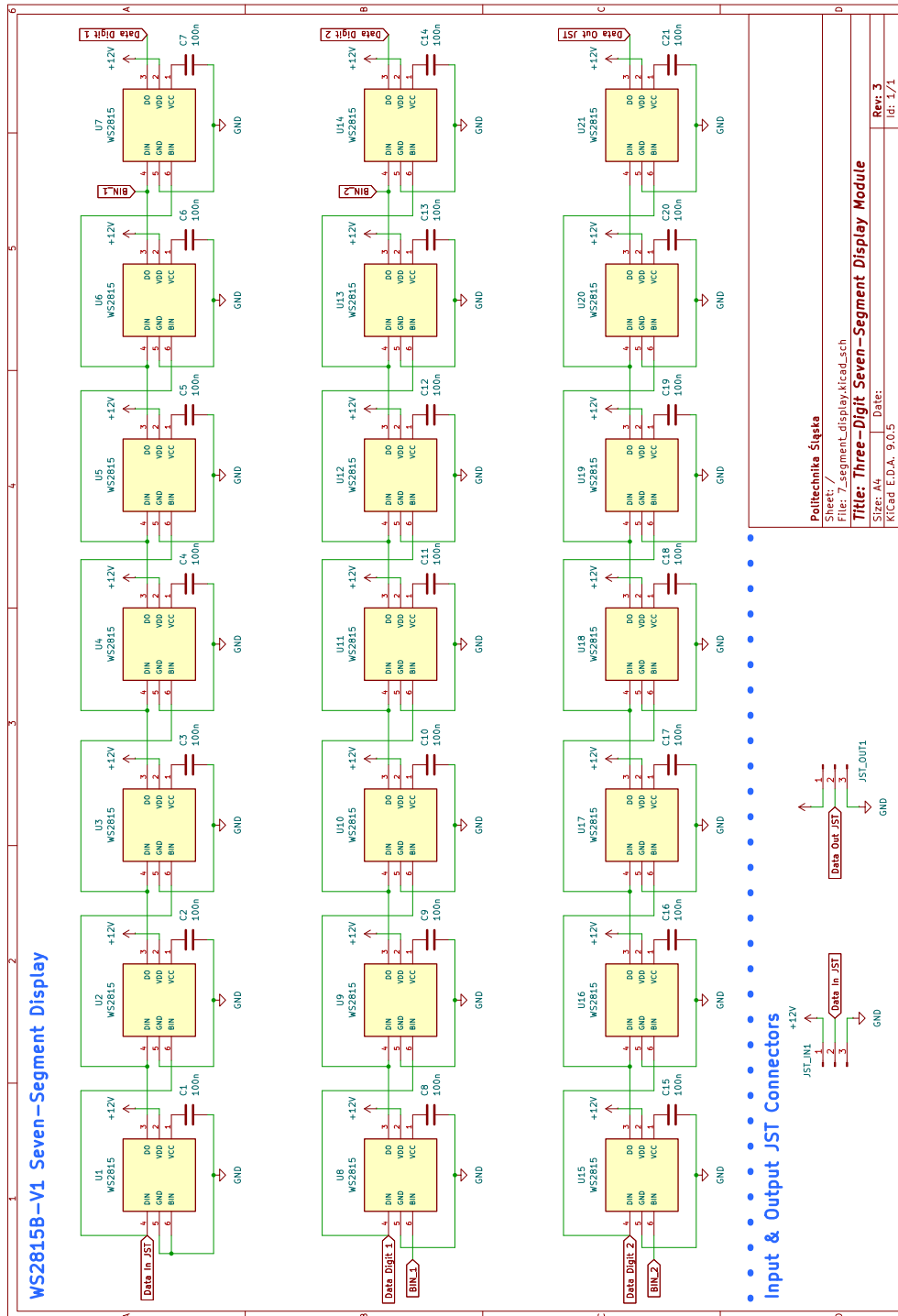
Na rysunku (Rys. 4.1) przedstawiono schemat płyty głównej systemu wizualizacji interfejsu Maszyny W.



Rysunek 4.1: Schemat płyty głównej z mikroprocesorem ESP32-S3

## 4.2 Wyświetlacz siedmiosegmentowy trzycyfrowy

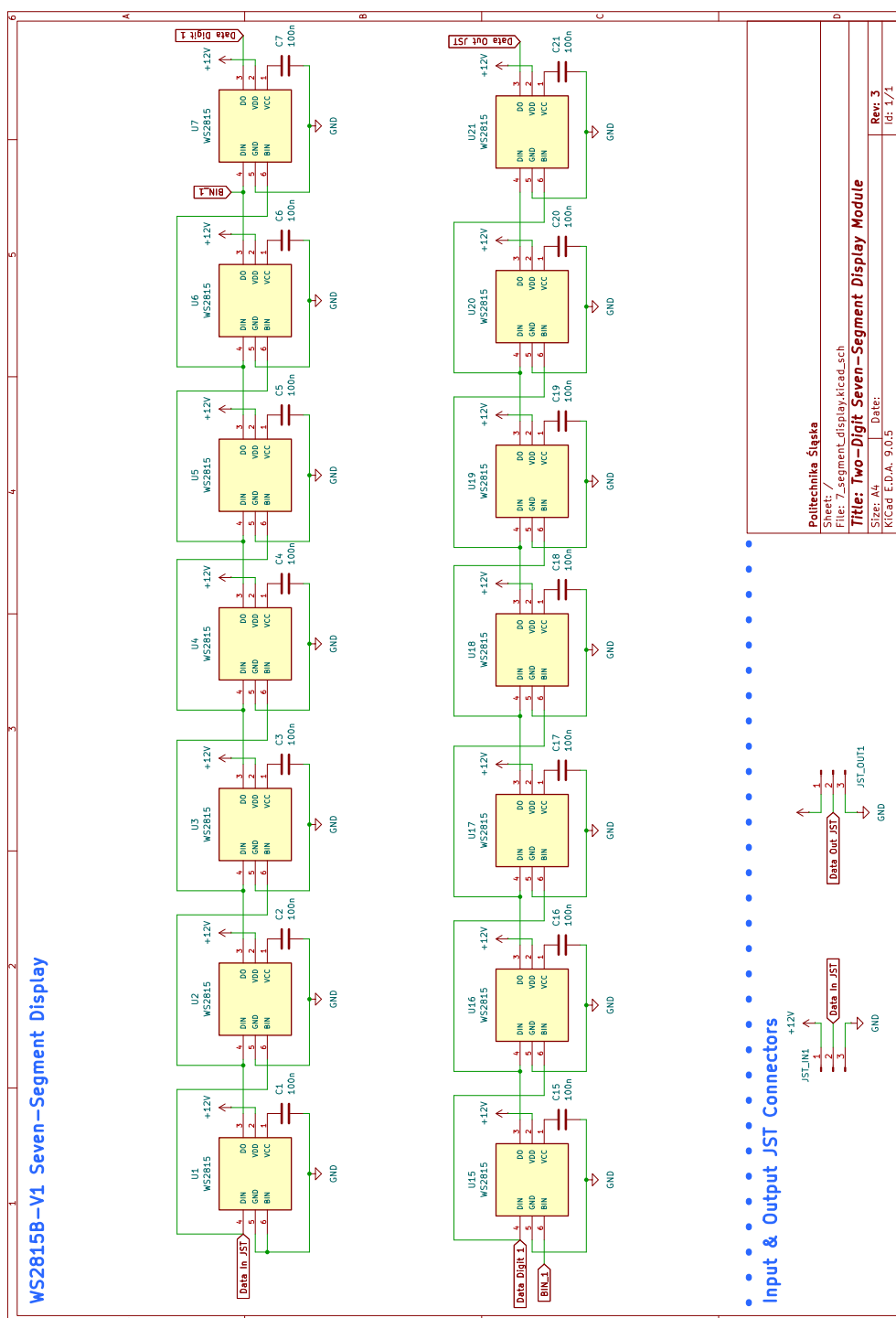
Na rysunku (Rys. 4.2) przedstawiono schemat schemat wyświetlacza trzycyfrowego skonstruowanego z połączonych ze sobą w linii ciągłej diod LED.



Rysunek 4.2: Wyświetlacz siedmiosegmentowy trzycyfrowy

## 4.3 Wyświetlacz siedmiosegmentowy dwucyfrowy

Na rysunku (Rys. 4.3) przedstawiono schemat wyświetlacza dwucyfrowego skonstruowanego z połączonych ze sobą w linii ciągłej diod LED.



Rysunek 4.3: Wyświetlacz siedmiosegmentowy trzycyfrowy

## 4.4 Połączenia pomiędzy blokami

TODO totalnie nie wiem co tutaj napisać do prowadzącego





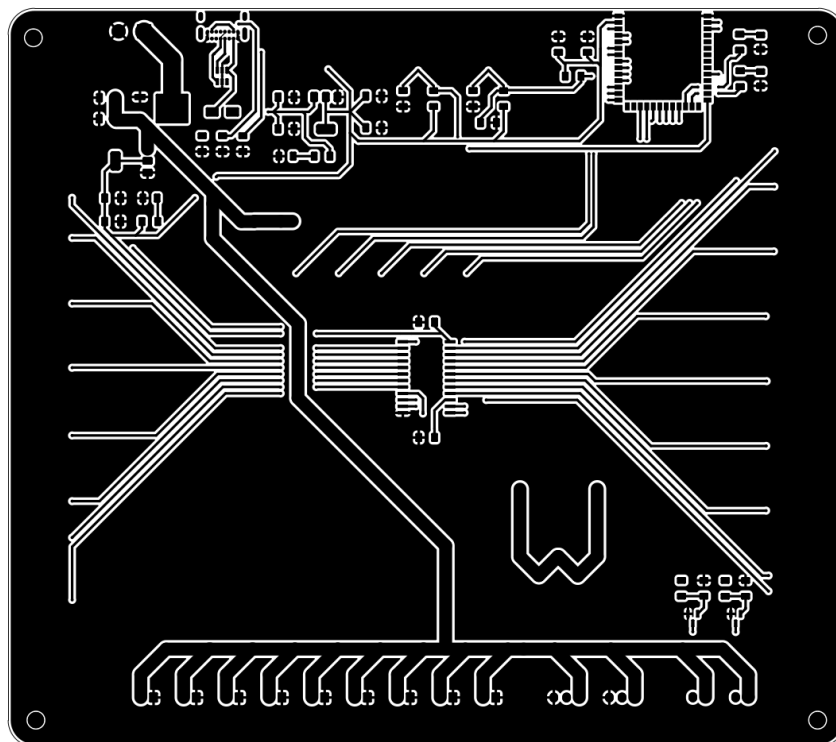
# Rozdział 5

## Część konstrukcyjna

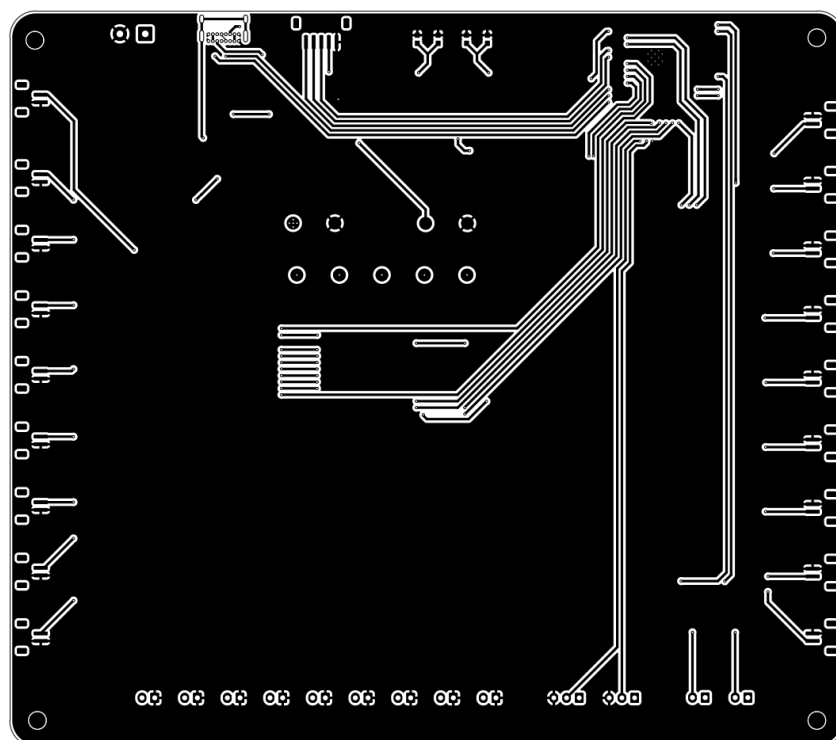
### 5.1 Obwody drukowane

Obwody drukowane zostały wykonane na specjalne zamówienie przez zewnętrzną firmę *JLCPCB*. Mozaikę ścieżek płytek drukowanych zaprojektowano wykorzystując program *KiCAD*. Odpowiednie rysunki przedstawiające warstwę górną oraz dolną płytek drukowanych przedstawiono w podrozdziałach 5.1.1 oraz 5.1.2.

#### 5.1.1 Płyta główna

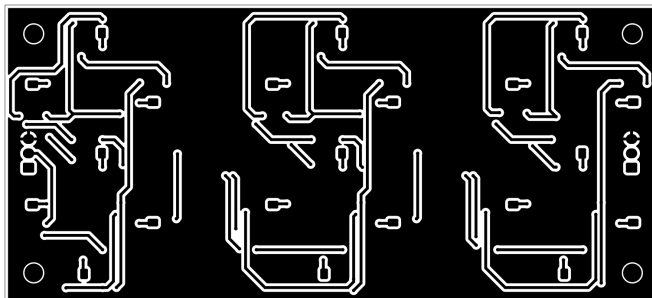


Rysunek 5.1: Widok warstwy górnej płyty głównej

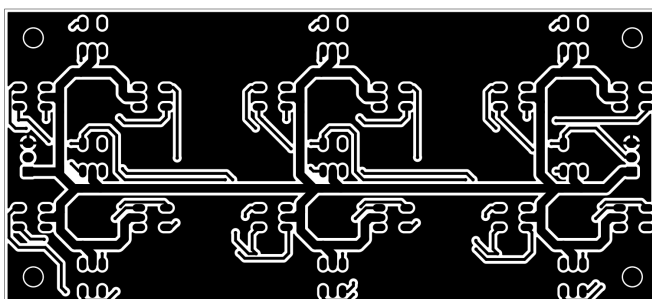


Rysunek 5.2: Widok warstwy dolnej płyty głównej

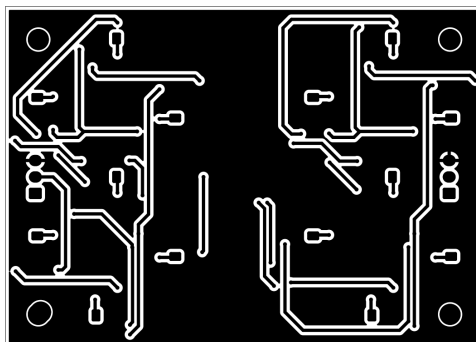
### 5.1.2 Wyświetlacze siedmiosegmentowe



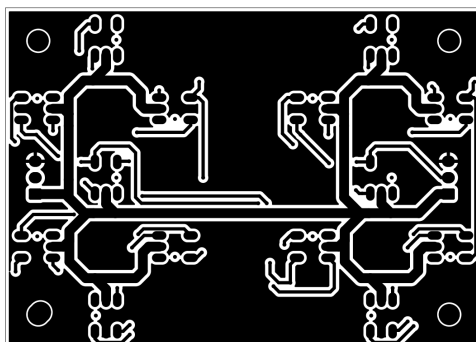
Rysunek 5.3: Widok warstwy górnej wyświetlacza trzycyfrowego



Rysunek 5.4: Widok warstwy dolnej wyświetlacza trzycyfrowego



Rysunek 5.5: Widok warstwy górnej wyświetlacza dwucyfrowego



Rysunek 5.6: Widok warstwy dolnej wyświetlacza dwucyfrowego

## 5.2 Opis złącz

### 5.2.1 Płyta główna

#### Złącze zasilania

Urządzenie jest wyposażone w 2-pinową listwę zaciskową typu *ARK*. Złącze pełni funkcję głównego wejścia zasilania i przyjmuje napięcie 12 V. Układ elektryczny wyposażono w zabezpieczenie przed odwrotną polaryzacją, uniemożliwiające uszkodzenie płytki w przypadku błędnego podłączenia przewodów. Wybrany model złącza wspiera napięcia do 250 V oraz prąd znamionowy do 18 A.

#### Złącze USB-C

W celu umożliwienia łatwego programowania i komunikacji z mikroprocesorem, płytkę wyposażono w złącze USB typu C (tryb USB 2.0 Device). Wyprowadzone są jedynie linie  $D+$  oraz  $D-$ . Linie zasilania *VBUS* są wykorzystywane do zasilania sekcji USB oraz wykrywania podłączenia komputera. Pozostałe piny USB-C niewykorzystywane przez układ zostały pozostawione niepodłączone, zgodnie ze specyfikacją.

#### Złącze enkodera

Do podłączenia impulsatora zastosowano 5-pinowe złącze typu *JST-ZH* o rastrze 1.5 mm. Wyprowadzone sygnały:

- $+3V3$  – zasilanie elementu,
- *GND* – masa układu,
- *CLK* – pierwszy kanał kwadraturowy enkodera,
- *DT* – drugi kanał kwadraturowy enkodera,
- *SW* – złącze przycisku enkodera

Sygnały prowadzone są do wejść cyfrowych mikroprocesora. Redukcja drgań styków zapewniono z poziomu oprogramowania.

#### Złącza przycisków

Każdy przycisk posiada dedykowane 2-pinowe złącze typu *JST-ZH* o rastrze 1.5 mm. Jeden pin każdego złącza podłączony jest do masy *GND*, drugi – do wejścia cyfrowego mikroprocesora. Linie sygnałowe wyposażono w podciągnięcie napięcia z poziomu wbudowanych w mikroprocesor rezystorów podciągających. Filtrację sygnałów zapewniono od strony oprogramowania.

### Złącze przełącznika trybów

Przełącznik trybów posiada dedykowane 2-pinowe złącze typu *JST-ZH* o rastrze 1.5 mm. Jeden pin złącza podłączony jest do masy *GND*, drugi – do wejścia cyfrowego mikroprocesora wykorzystującego wbudowany w mikroprocesor rezystor podciągający. Filtrację sygnałów zapewniono od strony oprogramowania.

### Złącza linii LED

Dwa wyprowadzenia linii LED wykorzystują 3-pinowe złącza typu *JST-XH* o rastrze 2.5 mm, prowadzące sygnały do linii diod LED typu *WS2815B-V1*. Piny:

- *+5V* – zasilanie diod,
- *DATA* – linia danych,
- *GND* – masa układu.

Dodatkowo zastosowano kondensator filtrujący do uziemienia na każdej z adresowalnych diod obecnych na płytkach drukowanych, zgodnie z dokumentacją producenta.

### Złącza zasilania podświetlenia modelu

Płytką zawiera dodatkowe dwa 2-pinowe złącza typu *JST-XH* o rastrze 2.5 mm. Każde złącze wspiera kontrolę zasilania 12 V z poziomu mikroprocesora. W tym celu zostały wykorzystane tranzystory typu *MOSFET*.

### Złącza dodatkowego wprowadzania zasilania

Projekt układu przewiduje potrzebę dodatkowego wprowadzania zasilania na linie adresowalnych diod LED. W tym celu płytkę wyposażono w 9 złącz typu *JST-XH* o rastrze 2.5 mm. Każde złącze zapewnia zasilanie 12 V.

## 5.2.2 Wyświetlacze segmentowe

### Złącza wejścia/wyjścia

Każdy z wyświetlaczy segmentowych (zarówno trzycyfrowe, jak i dwucyfrowe) posiadają dwa złącza *JST\_IN* oraz *JST\_OUT* typu *JST-XH* o rastrze 2.5 mm. Złącza umożliwiają łączenie wyświetlaczy w ciągłe linie. Piny:

- *+5V* – zasilanie diod,
- *DATA\_IN/DATA\_OUT* – wejście/wyjście linii danych,
- *GND* – masa układu.



# Rozdział 6

## Opis oprogramowania

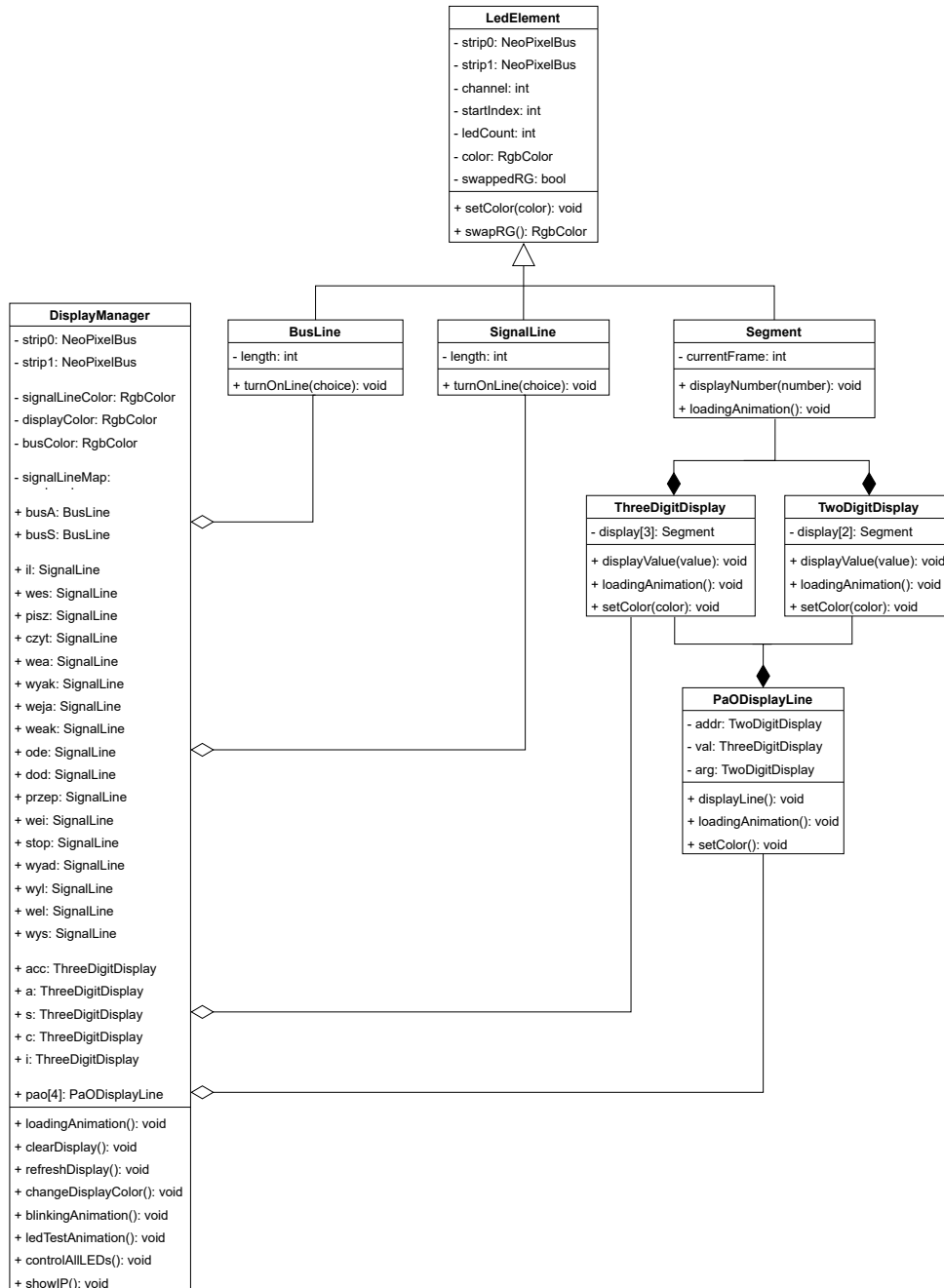
### 6.1 Program mikrokontrolera

Program mikrokontrolera ESP32 został napisany w języku *C++* za pomocą aplikacji *Visual Studio Code*. Kod wynikowy zajmuje około 900kB. Mikrokontroler ma do zrealizowania następujące zadania:

- obsługa linii adresowalnych diod LED,
- obsługa przycisków, enkodera oraz przełącznika trybów,
- obsługa pamięci nieulotnej/systemu plików,
- obsługa trybu sieciowego SWIM,
- obsługa trybu lokalnego SWIM,
- kontrola podświetlenia LED SWIM,
- obsługa enkodera.

Kompletny kod źródłowy mikrokontrolera wraz z instrukcją instalacji został zamieszczony w publicznym repozytorium pod adresem: *TODO adres repo*.

### 6.1.1 Obsługa linii adresowalnych diod LED



Rysunek 6.1: Diagram zależności klas obsługujących diody LED

TODO opis części kodu z przykładowymi funkcjami itp



- 6.1.2 Obsługa przycisków, enkodera oraz przełącznika trybów
- 6.1.3 Obsługa pamięci nieulotnej/systemu plików
- 6.1.4 Obsługa trybu sieciowego SWIM
- 6.1.5 Obsługa trybu lokalnego SWIM



# Rozdział 7

## Obsługa

Tutaj trzeba będzie opisać jak działa maszyna i jak ją można obsługiwać



# Dodatki



# Spis skrótów i symboli

SWIM system wizualizacji interfejsu maszyny

RGB red green blue

LED light emitting diode

USB universal serial bus

ARK Anschluss Reihenklemme Klemme





# Źródła

Jeżeli w pracy konieczne jest umieszczenie długich fragmentów kodu źródłowego, należy je przenieść w to miejsce.

---

```
1 if (_nClusters < 1)
2     throw std::string ("unknown_number_of_clusters");
3 if (_nIterations < 1 and _epsilon < 0)
4     throw std::string ("You_should_set_a_maximal_number_of_
        iteration_or_minimal_difference_epsilon.");
5 if (_nIterations > 0 and _epsilon > 0)
6     throw std::string ("Both_number_of_iterations_and_minimal_
        epsilon_set—you_should_set_either_number_of_iterations
        or_minimal_epsilon.");
```

---



# Lista dodatkowych plików, uzupełniających tekst pracy

W systemie do pracy dołączono dodatkowe pliki zawierające:

- źródła programu,
- dane testowe,
- film pokazujący działanie opracowanego oprogramowania lub zaprojektowanego i wykonanego urządzenia,
- itp.



# Spis rysunków

2.1	Schemat blokowy SWIM . . . . .	4
3.1	Schemat blokowy funkcjonalny układu ESP32-S3 . . . . .	6
3.2	Schemat wyprowadzeń adresowalnej diody LED RGB <i>WS2815B-V1</i> . . . . .	7
3.3	Diagram funkcjonalny multipleksera <i>CD74HC4067M</i> . . . . .	8
3.4	Wykres stanów przy obrocie w prawo . . . . .	11
3.5	Wykres stanów przy obrocie w lewo . . . . .	11
4.1	Schemat płyty głównej z mikroprocesorem ESP32-S3 . . . . .	14
4.2	Wyświetlacz siedmiosegmentowy trzycyfrowy . . . . .	15
4.3	Wyświetlacz siedmiosegmentowy trzycyfrowy . . . . .	16
5.1	Widok warstwy górnej płyty głównej . . . . .	19
5.2	Widok warstwy dolnej płyty głównej . . . . .	20
5.3	Widok warstwy górnej wyświetlacza trzycyfrowego . . . . .	21
5.4	Widok warstwy dolnej wyświetlacza trzycyfrowego . . . . .	21
5.5	Widok warstwy górnej wyświetlacza dwucyfrowego . . . . .	21
5.6	Widok warstwy dolnej wyświetlacza dwucyfrowego . . . . .	21
6.1	Diagram zależności klas obsługujących diody LED . . . . .	26



# Spis tabel

3.1	Tablica prawdy multipleksera . . . . .	9
3.2	Tablica prawdy dla obrotu w prawo . . . . .	11
3.3	Tablica prawdy dla obrotu w lewo . . . . .	11