



**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 slow (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

1 Wstęp	1
1.1 Cel pracy	1
1.2 Analiza rozwiązań literaturowych	1
2 Przykładowa realizacja systemu wizualizacji	3
2.1 Założenia realizacyjne	3
2.2 Schemat blokowy	4
3 Część projektowa	5
3.1 System mikroprocesorowy	5
3.2 Adresowalne diody LED	7
3.3 Przyciski	7
3.4 Podświetlenie LED	8
3.5 Enkoder	8
3.6 Przełącznik trybów	8
4 Schematy bloków systemu wizualizacji	9
4.1 System mikroprocesorowy – płyta główna	9
4.2 Wyświetlacz siedmiosegmentowy trzycyfrowy	10
4.3 Wyświetlacz siedmiosegmentowy dwucyfrowy	11
4.4 Połączenia pomiędzy blokami	11
5 Część konstrukcyjna	13
5.1 Obwody drukowane	13
5.1.1 Płyta główna	13
5.1.2 Wyświetlacze siedmiosegmentowe	15
5.2 Opis złącz	16
5.2.1 Płyta główna	16
5.2.2 Wyświetlacze segmentowe	17
6 Opis oprogramowania	19
6.1 Program mikrokontrolera	19

6.1.1	Obsługa linii adresowalnych diod LED	20
6.1.2	Obsługa przycisków, enkodera oraz przełącznika trybów	21
6.1.3	Obsługa pamięci nieulotnej/systemu plików	21
6.1.4	Obsługa trybu sieciowego SWIM	21
6.1.5	Obsługa trybu lokalnego SWIM	21
7	Obsługa	23
	Bibliografia	25
	Spis skrótów i symboli	27
	Źródła	29
	Lista dodatkowych plików, uzupełniających tekst pracy	31
	Spis rysunków	33
	Spis tabel	35

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 реализациj oraz wyborem tych, które są optymalne i spełniają przyjęte założenia. Założenia dodatkowe nie są bezwzględnie wymagane, ale mogą poszerzyć tworzone rozwiązań o alternatywne sposoby działania.

Założenia podstawowe:

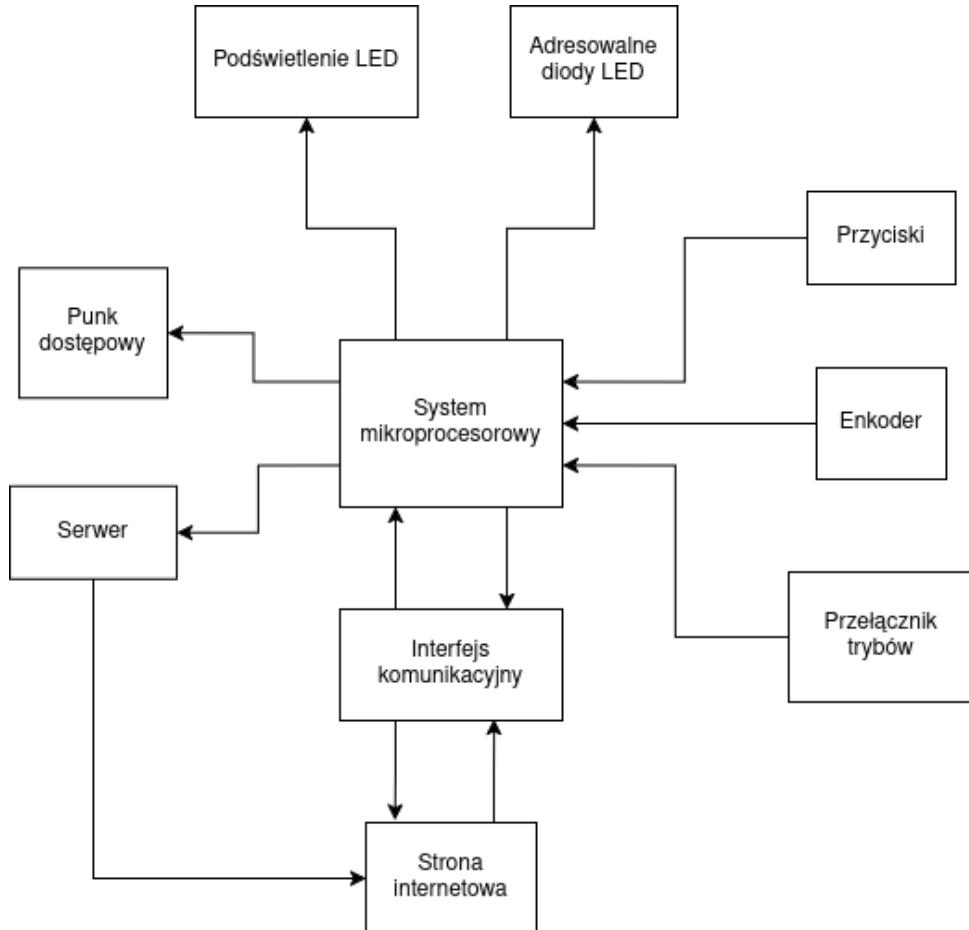
- dedykowany układ dla mikrokontrolera z odpowiednimi wyprowadzeniami,
- dedykowany układ dla LED-owych wyświetlaczy siedmiosegmentowych,
- konstrukcja kompletnego układu elektronicznego z układów mikrokontrolera, wyświetlacz LED oraz przycisków,
- hosting strony internetowej bezpośrednio na urządzeniu 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:

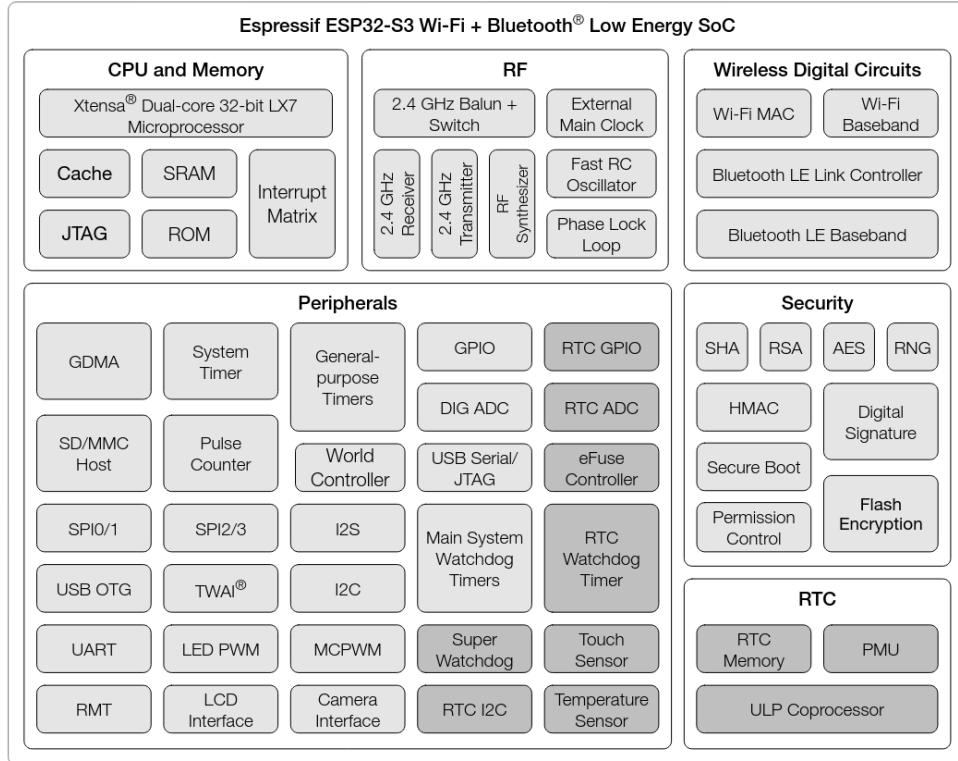
- obsługa dwóch kanałów dla adresowalnych diod LED RGB,
- obsługa 17 przycisków wraz z enkoderem,
- zapewnienie punktu dostępu internetu,
- przechowywanie danych w pamięci nieulotnej,
- serwowanie strony internetowej.

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*a 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 decydowano 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:

- 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,
- zasilanie napięciem 3.3V.

3.2 Adresowalne diody LED

Zakładano wizualizację wielu informacji na znaczących rozmiarów modelu Maszyny W. Przewidywana liczba pojedynczych diod LED przekracza 600. Oznaczać to może znaczące spadki napięcia na długiej linii LED.

Podczas wstępnych testów linii 5V zaobserwowano zauważalny spadek jakości odzwierciedlania kolorów.

Biorąc pod uwagę wyżej wymienione wymagania zdecydowano się na użycie adresowalnych diod *WS2815B-V1* firmy *WORLDSEMI*. Wybrany model diod pozwala na indywidualne zadawanie wartości Red, Green oraz Blue każdej z nich. Wyższe napięcie zasilania eliminuje powstałe w wyniku spadków napięcia zmiany barwy.

Własności elementu

- napięcie pracy 12V,
- obudowa 5050,
- montaż SMD,
- częstotliwość odświeżania 4kHz.

3.3 Przyciski

W celu zapewnienia pełnej obsługi Maszyny W potrzebne jest wykorzystanie 17 monostabilnych 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 kliknięć. Pozwoli to na długofalową, niezawodną działalność. Wykorzystanie przycisków typu *Blue* oferujących słyszalny odgłos kliknięcia przy wciskaniu. Daje to użytkownikowi informację zwrotną o stanie przełącznika.

Wykorzystanie tak dużej liczby przycisków wymaga użycia multipleksera 16:4. Z uwagi na niską cenę, łatwą dostępność oraz niewielki wymiar wybrano model *CD74HC4067M* firmy *Texas Instruments*.

3.4 Podświetlenie LED

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

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

Wykorzystano do tego 12V linie diod LED.

TODO skończyć tekst PODŚWIETLENIA LED

3.5 Enkoder

Do wprowadzania danych liczbowych w lokalnym trybie działania urządzenia zastosowano enkoder obrotowy z przyciskiem. Wykorzystanie 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.

3.6 Przełącznik trybów

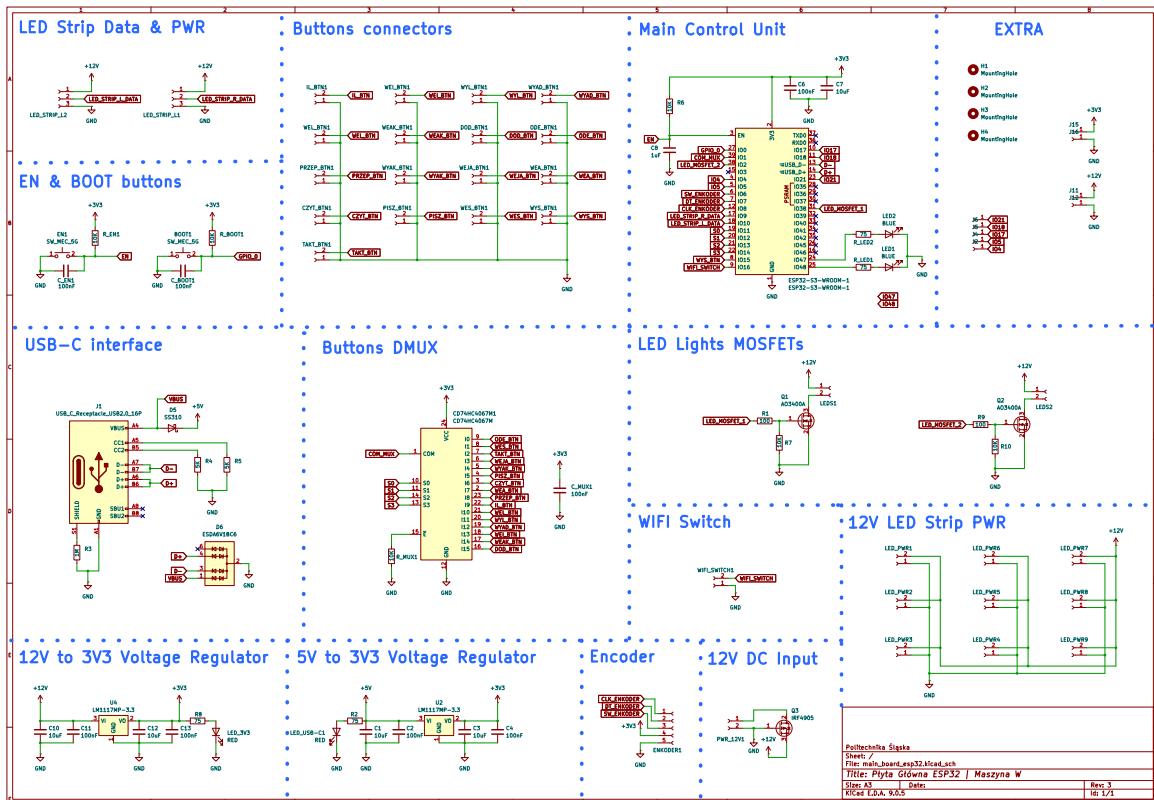
Jednym z założeń dodatkowych projektu jest możliwość przełączania SWIM między trybami lokalnym oraz sieciowym. Aby zapewnić taką możliwość bezpośrednio z poziomu modelu wykorzystano przełącznik bistabilny.

Rozdział 4

Schematy bloków systemu wizualizacji

4.1 System mikroprocesorowy – płyta główna

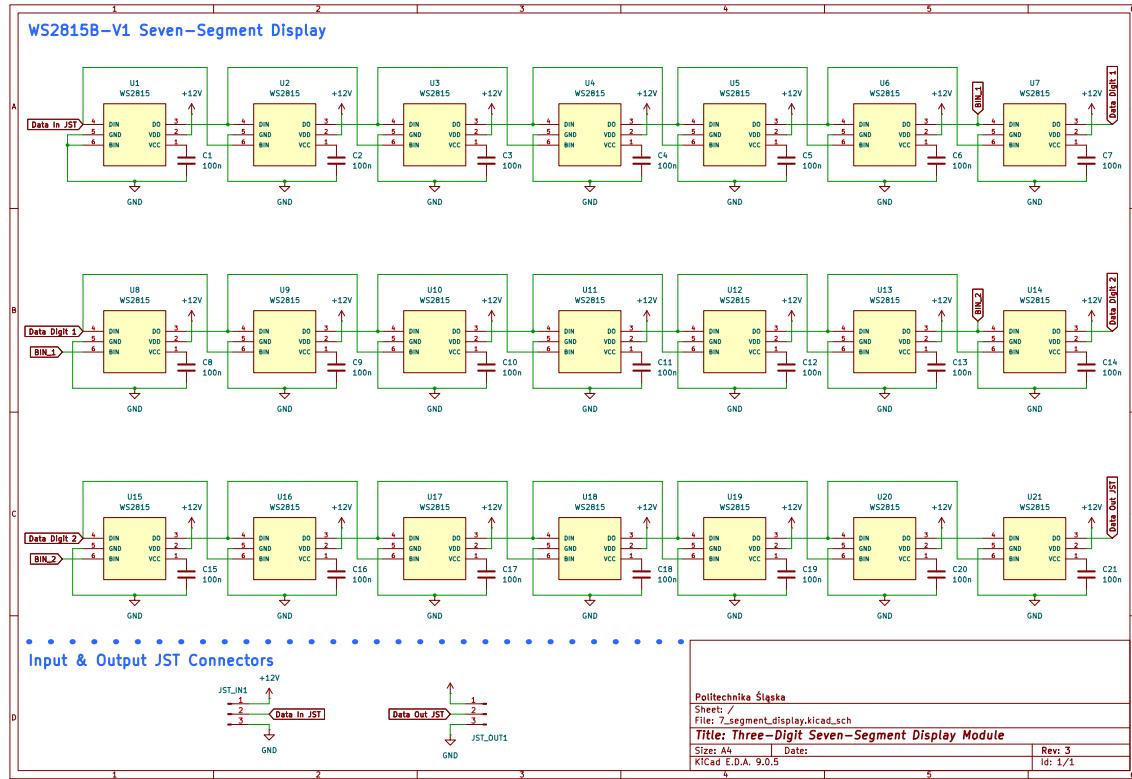
Na rysunku (Rys. 4.1) przedstawiono schemat systemu mikroprocesorowego.



Rysunek 4.1: Schemat płyty głównej z mikroprocesorem ESP32-S3

4.2 Wyświetlacz siedmiosegmentowy trzycyfrowy

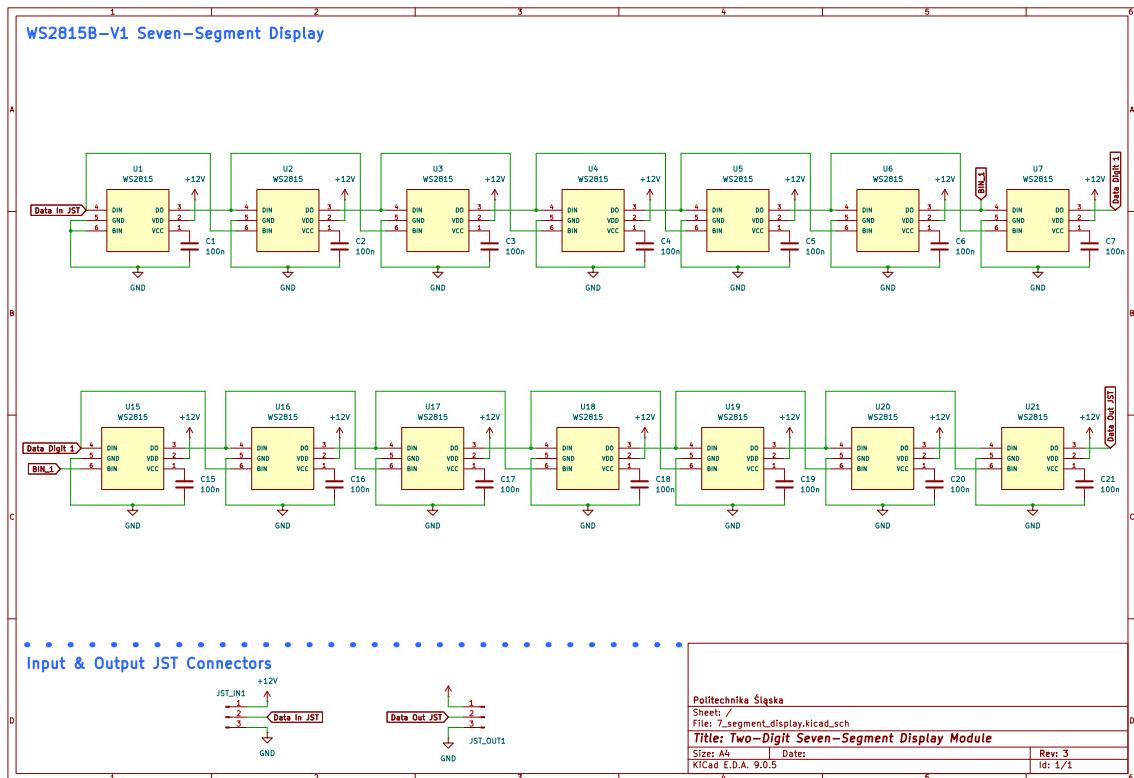
Na rysunku (Rys. 4.2) przedstawiono schemat wyświetlacza trzycyfrowego skonstruowanego z połączonych ze sobą w linii ciągowej diod LED.



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

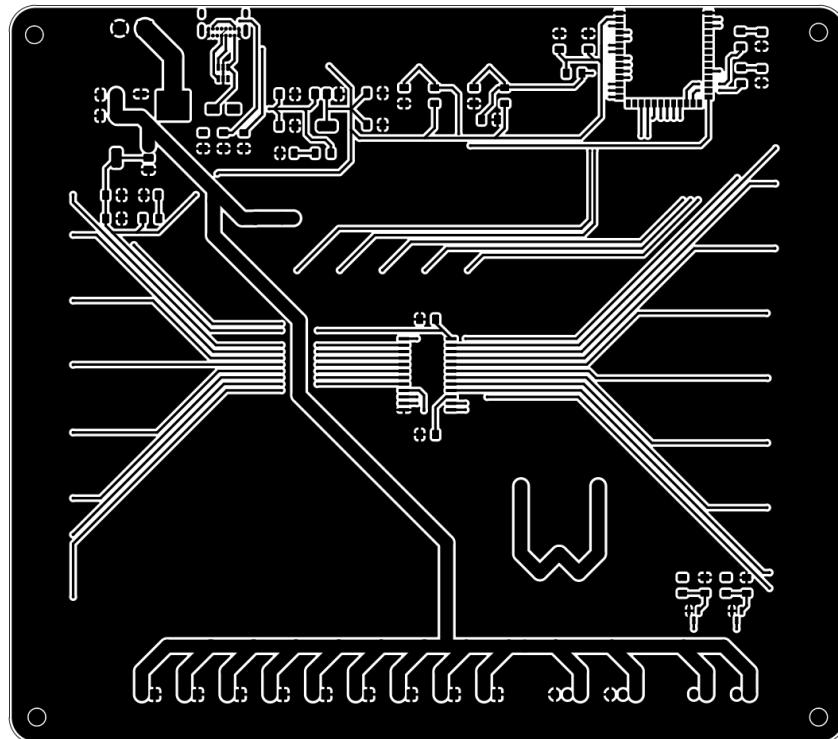
Rozdział 5

Część konstrukcyjna

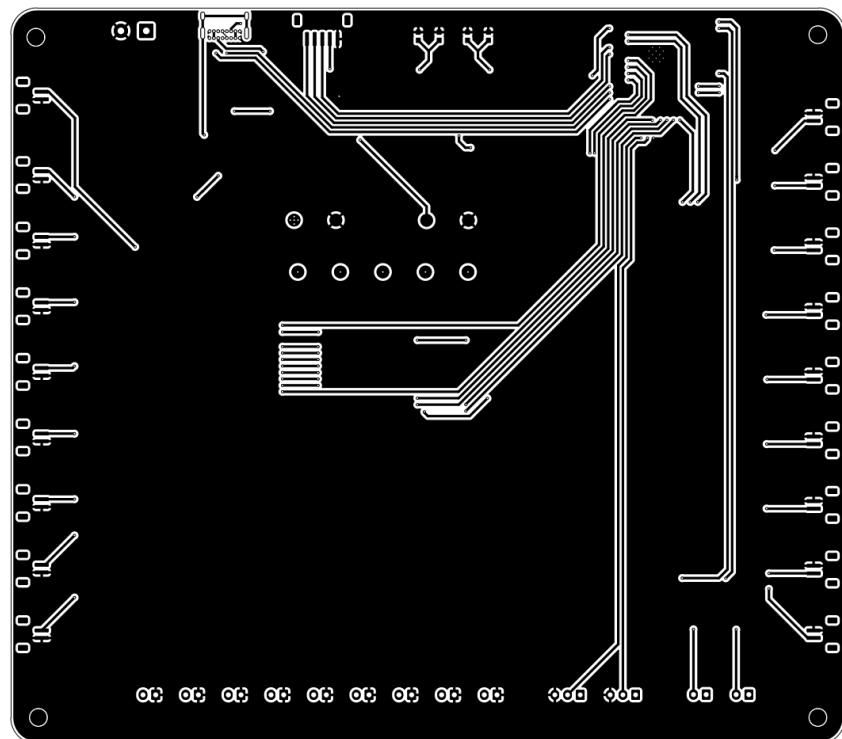
5.1 Obwody drukowane

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

5.1.1 Płyta główna

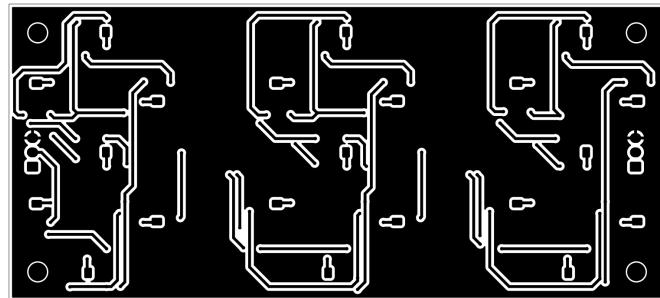


Rysunek 5.1: Widok warstwy górnej płyty głównej

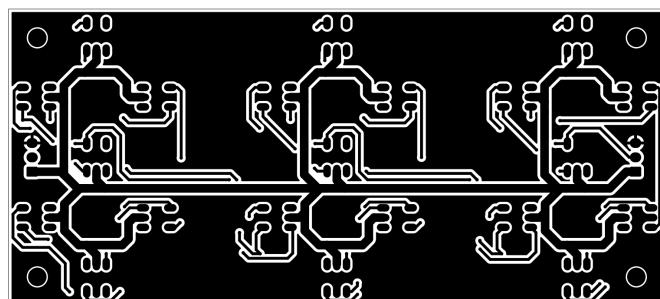


Rysunek 5.2: Widok warstwy dolnej płyty głównej

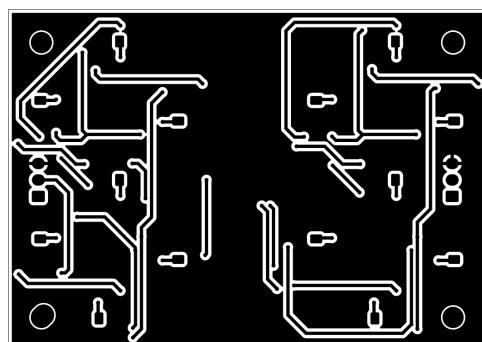
5.1.2 Wyświetlacze siedmiosegmentowe



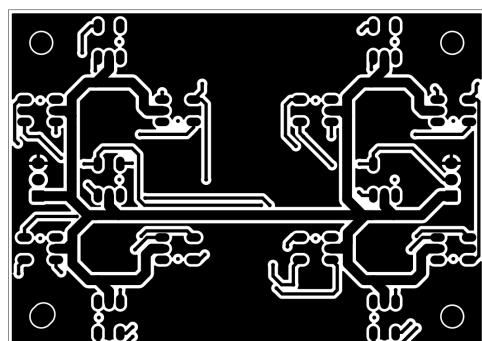
Rysunek 5.3: Widok warstwy górnej wyświetlacza trzycyfrowego



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 płytka 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, płytę 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 podciagnięcie 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 płytach drukowanych, zgodnie z dokumentacją producenta.

Złącza zasilania podświetlenia modelu

Płytki zawierają 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 płytę wyposażono w 9 złączy typu *JST-XH* o rastrze 2.5 mm. Każde złącze zapewnia zasilanie 12 V.

5.2.2 Wyświetlacze segmentowe

Złącza wejścia/wyjścia

Każdy z wyświetlaczów segmentowych (zarówno trzycyfrowe, jak i dwucyfrowe) posiada dwa złącza *JST_IN* oraz *JST_OUT* typu *JST-XH* o rastrze 2.5 mm. Złącza umożliwiają łączenie wyświetlaczów w ciągłe linie. Piny:

- $+5V$ – zasilanie diod,
- *DATA_IN/DATA_OUT* – wejście/wyjście linii danych,
- *GND* – masa układu.

Rozdział 6

Opis oprogramowania

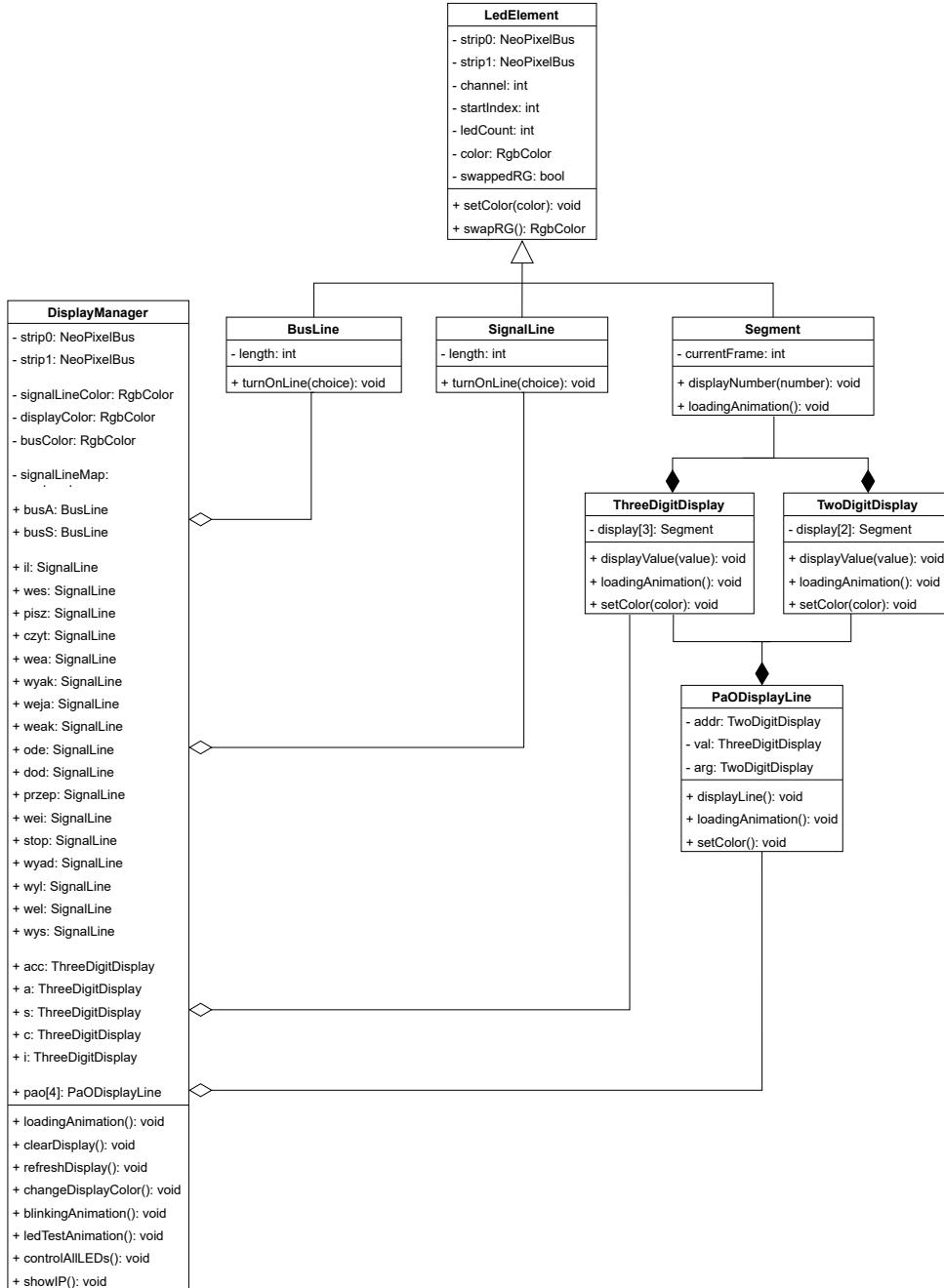
6.1 Program mikrokontrolera

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

- 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 Anschlus 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\u00b7number\u00b7of\u00b7clusters");
3 if (_nIterations < 1 and _epsilon < 0)
4     throw std::string ("You\u00b7should\u00b7set\u00b7a\u00b7maximal\u00b7number\u00b7of\u00b7
5         iteration\u00b7or\u00b7minimal\u00b7difference\u00b7\u00b7epsilon.");
6 if (_nIterations > 0 and _epsilon > 0)
7     throw std::string ("Both\u00b7number\u00b7of\u00b7iterations\u00b7and\u00b7minimal\u00b7
8         epsilon\u00b7set\u00b7\u00b7you\u00b7should\u00b7set\u00b7either\u00b7number\u00b7of\u00b7iterations
9         \u00b7or\u00b7minimal\u00b7epsilon.");
```

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
4.1	Schemat płyty głównej z mikroprocesorem ESP32-S3	9
4.2	Wyświetlacz siedmiosegmentowy trzycyfrowy	10
4.3	Wyświetlacz siedmiosegmentowy trzycyfrowy	11
5.1	Widok warstwy górnej płyty głównej	13
5.2	Widok warstwy dolnej płyty głównej	14
5.3	Widok warstwy górnej wyświetlacza trzycyfrowego	15
5.4	Widok warstwy dolnej wyświetlacza trzycyfrowego	15
5.5	Widok warstwy górnej wyświetlacza dwucyfrowego	15
5.6	Widok warstwy dolnej wyświetlacza dwucyfrowego	15
6.1	Diagram zależności klas obsługujących diody LED	20

Spis tabel