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

Gliwice 2026

Tytuł pracy

System wizualizacji danych webowego interfejsu Maszyny W z użyciem mikrokontrolera ESP32

Streszczenie

Przedmiotem pracy jest zaprojektowanie, budowa oraz zaprogramowanie fizycznego systemu wizualizacji danych dla internetowego symulatora Maszyny W. Urządzenie oparte na mikrokontrolerze *ESP32* komunikuje się ze stroną poprzez protokół *WebSocket*, zapewniając synchronizację w czasie rzeczywistym. Warstwa sprzętowa obejmuje trzy autorskie projekty płytek drukowanych: płytę główną oraz moduły wyświetlaczy siedmiosegmentowych w wersji dwu- i trzycyfrowej, wykorzystujące adresowalne diody LED *WS2815B*. Interfejs sterujący z 17 przyciskami i enkoderem umożliwia pełną, dwukierunkową interakcję z modelem, który posiada także lokalny tryb autonomicznej symulacji. Gotowy system stanowi nowoczesne narzędzie dydaktyczne, łączące symulację cyfrową z modelem fizycznym.

Słowa kluczowe

Maszyna W, ESP32, Systemy wbudowane, Płytki drukowane, Diody LED RGB

Thesis title

Visualization System for the Maszyna W Web Interface Using an ESP32 Microcontroller

Abstract

The subject of this thesis is the design, construction, and programming of a physical data visualization system for the Maszyna W web-based simulator. Based on the *ESP32* microcontroller, the device utilizes the *WebSocket* protocol for real-time synchronization with the website. The hardware layer consists of three custom-designed PCBs: a main board and seven-segment display modules in three- and two-digit versions, all employing addressable *WS2815B* LEDs. A control interface with 17 buttons and an encoder enables full bi-directional interaction, while a local mode allows for autonomous simulation on the microcontroller. The completed system serves as a modern educational tool, bridging digital simulation with a physical model.

Key words

W Machine, ESP32, Embedded systems, Printed circuit boards, LED RGB diodes

Spis treści

Rozdział 1

Wstęp

1.1 Cel pracy

Celem niniejszej pracy jest opracowanie systemu wizualizacji danych pochodzących z sieciowego interfejsu Maszyny W z wykorzystaniem mikrokontrolera *ESP32*.

Opracowane w ramach pracy dyplomowej urządzenie umożliwia odwzorowanie, w czasie rzeczywistym, wartości liczbowych oraz sygnałów sterujących symulatorem Maszyny W na adresowalnych diodach LED (z ang. *Light-Emitting Diode*) typu RGB (z ang. *Red, Green, Blue*).

Urządzenie pozwala na zdalne sterowanie funkcjami symulatora zarówno za pomocą przycisków fizycznych, jak i interfejsu internetowego. System obsługuje udostępnianie strony internetowej bezpośrednio na mikrokontrolerze oraz zapewnia komunikację pomiędzy stroną a urządzeniem. Wykorzystane rozwiązania umożliwiają natychmiastową synchronizację zmian między stroną internetową a modelem Maszyny, co pozwala użytkownikowi na intuicyjny i interaktywny monitoring oraz kontrolę stanu Maszyny W.

1.2 Wprowadzenie do Maszyny W

Współczesne systemy komputerowe charakteryzują się wysokim stopniem złożoności zarówno pod względem architektury sprzętowej, jak i warstwy programowej. Pomimo tego, podstawowe zasady ich działania pozostają zgodne z klasycznymi założeniami architektury von Neumanna, sformułowanymi w połowie XX wieku [8].

Maszyna W została zaprojektowana w latach siedemdziesiątych XX wieku na Politechnice Śląskiej przez zespół kierowany przez prof. zw. dr inż. Stefana Węgrzyna jako uproszczony model komputera, przeznaczony do nauki podstaw architektury komputerów oraz zasad działania procesora [1].

Konstrukcja stanowi uproszczenie rzeczywistych jednostek obliczeniowych, przy jednoczesnym zachowaniu kluczowych cech architektury von Neumanna, takich jak wspólna

pamięć dla danych i rozkazów oraz sekwencyjne wykonywanie instrukcji programu.

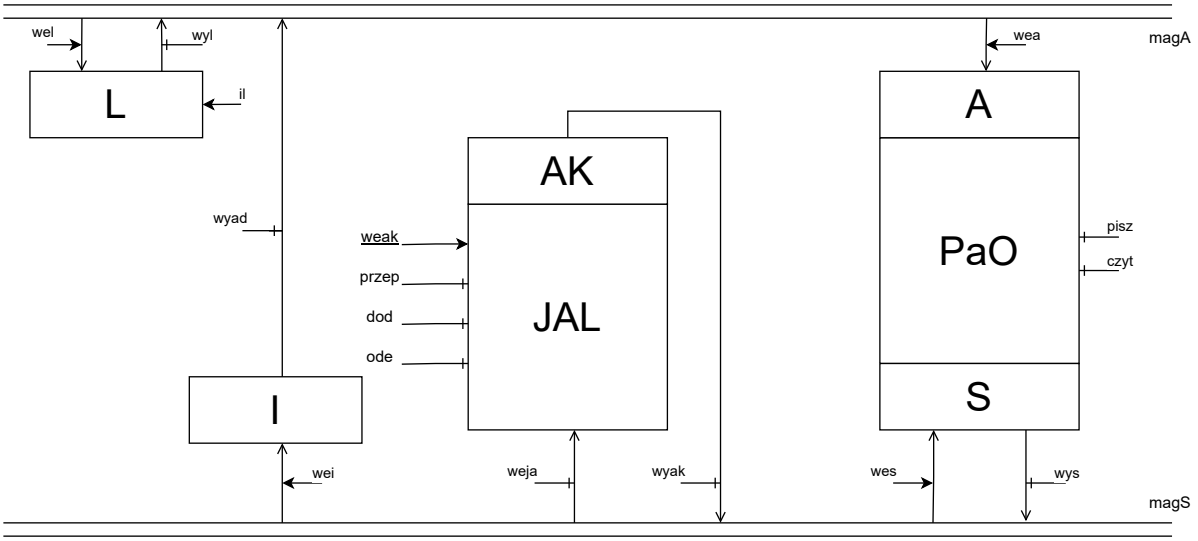
Zgodnie z przyjętymi założeniami, Maszyna W składa się z trzech podstawowych bloków funkcjonalnych: pamięci operacyjnej, jednostki arytmetyczno-logicznej oraz układu sterującego. W skład Maszyny W wchodzą następujące rejestry:

- **rejestr L** — 5-bitowy licznik rozkazów;
- **rejestr I** — 8-bitowy rejestr instrukcji. 5 mniej znaczących bitów stanowi rejestr AD, przechowujący adres argumentu rozkazu. 3 bardziej znaczące bity przechowują rejestr kodu rozkazu;
- **rejestr A** — 5-bitowy rejestr adresowy, służący do adresowania komórki pamięci, do której wykonywane są operacje zapisu i odczytu;
- **rejestr S** — 8-bitowy rejestr słowa, do którego zapisywane/odczytywane są wartości z pamięci;
- **rejestr AK** — 8-bitowy akumulator, do którego zapisywany jest wynik operacji arytmetycznej/logicznej.

Oraz następujące magistrale:

- **magistrala S** — 8-bitowa magistrala słowa,
- **magistrala A** — 5-bitowa magistrala adresowa.

Wymienione elementy znajdują się na poniższym schemacie urządzenia 1.1.



Rysunek 1.1: Schemat Maszyny W. Źródło: opracowanie własne na podstawie [7].

Pamięć operacyjna przechowuje zarówno dane, jak i rozkazy programu w formie 8-bitowego słowa. Jednostka arytmetyczno-logiczna (JAL) realizuje operacje arytmetyczne i logiczne (m.in. dodawanie, odejmowanie), wykorzystując akumulator jako główny rejestr roboczy, natomiast układ sterujący odpowiada za pobieranie, dekodowanie oraz wykonywanie rozkazów [7].

Istotnym elementem Maszyny W jest sposób sterowania przepływem danych pomiędzy jej komponentami. Komunikacja realizowana jest za pomocą magistrali adresowej i magistrali słowa, przebieg operacji kontrolowany jest przez zestaw sygnałów mikrosterujących (np. *czyt, wys, wei, il*). Sygnały te, umożliwiają precyzyjne sterowanie przesyłami międzypamięciowymi i międzypamięciowo-rejestrowymi w kolejnych taktach zegara [1]. Dzięki temu możliwe jest szczegółowe śledzenie procesu realizacji pojedynczego rozkazu na poziomie mikrooperacji.

Z perspektywy dydaktycznej szczególnie istotna jest możliwość analizy pełnego cyklu rozkazu, obejmującego fazy pobrania, dekodowania oraz wykonania instrukcji. W Maszynie W faza pobrania i dekodowania rozkazu jest identyczna dla wszystkich instrukcji, co pozwala na wyraźne oddzielenie części wspólnej cyklu rozkazu od operacji charakterystycznych dla konkretnego typu instrukcji. Takie podejście sprzyja lepszemu zrozumieniu mechanizmów sterowania procesorem oraz zależności pomiędzy rejestrami i pamięcią.

Maszyna W wykorzystywana jest obecnie głównie w postaci programowego symulatora, który umożliwia implementację i uruchamianie własnych rozkazów asemblerowych. W ramach zajęć laboratoryjnych studenci projektują rozkazy, definiując sekwencje mikrosygnałów realizujących zadane operacje [2]. Proces ten wymaga nie tylko znajomości struktury Maszyny W, lecz także umiejętności analizy algorytmicznej oraz świadomego zarządzania stanem rejestrów i pamięci.

Rozdział 2

Przykładowa realizacja systemu

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:

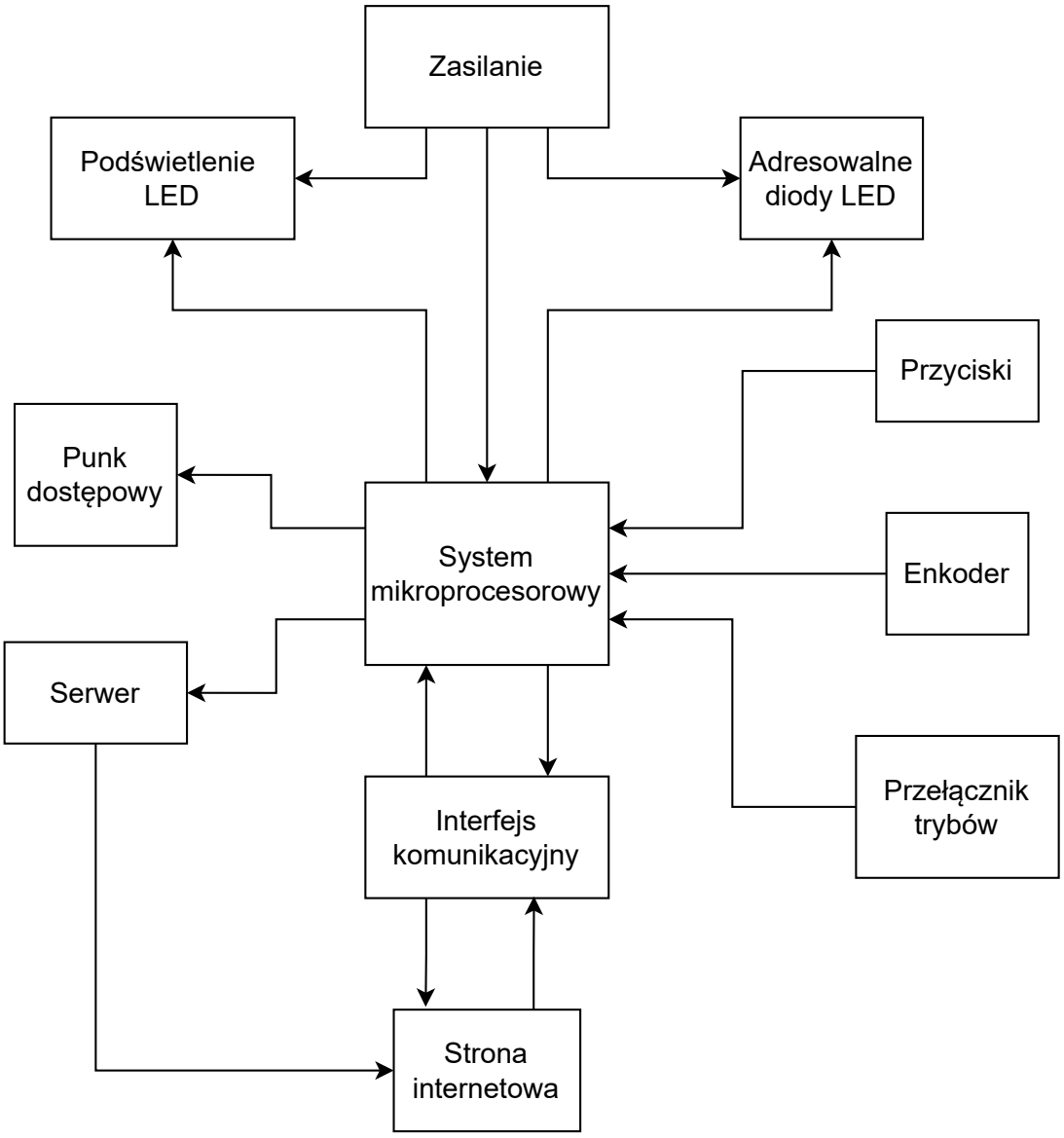
- dedykowany układ dla mikrokontrolera z odpowiednimi wyprowadzeniami,
- dedykowany układ dla LED-owych wyświetlaczy siedmiosegmentowych,
- konstrukcja kompletnego układu elektronicznego z układów mikrokontrolera, wyświetlaczy LED oraz przycisków,
- usługa udostępniania zasobów strony internetowej bezpośrednio 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. Źródło: opracowanie własne.

Rozdział 3

Część projektowa

3.1 System mikrokontrolerowy

Ze względu na wymaganą komunikację sieciową projektu, wysoką złożoność rozwiązania oraz zapotrzebowanie na wiele portów wejścia/wyjścia należało wybrać system mikrokontrolerowy zapewniający poniższą funkcjonalność:

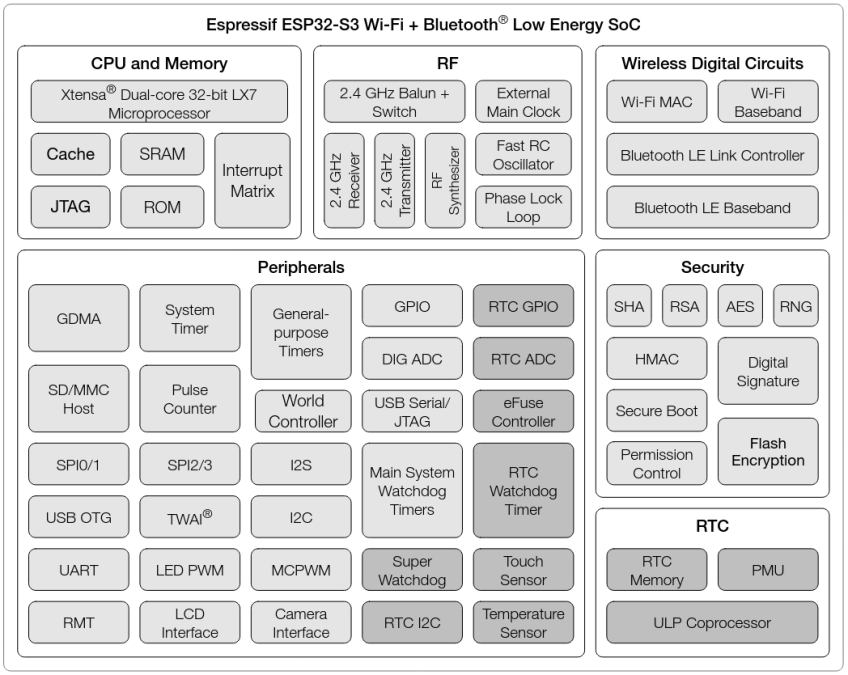
- obsługa dwóch kanałów dla adresowalnych diod LED RGB,
- obsługa 17 przycisków wraz z enkoderem,
- zapewnienie punktu dostępu do internetu,
- przechowywanie danych w pamięci nieulotnej,
- udostępnianie strony internetowej.

Biorąc pod uwagę powyższe 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* charakteryzują się prostotą obsługi oraz konfiguracji, są wyposażone w system operacyjny oparty na jądrze *Linuxa* 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 *Debiana* pozwala na tworzenie własnych procesów typu *daemon* (procesów działających 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*. Źródło: `https://www.circuitstate.com/wp-content/uploads/2023/08/Espressif-ESP32-S3-Functional-Block-Diagram-CIRCUITSTATE-Electronics-1.png`

Własności układu zapożyczone z dokumentacji firmy *Espressif Systems* [11]:

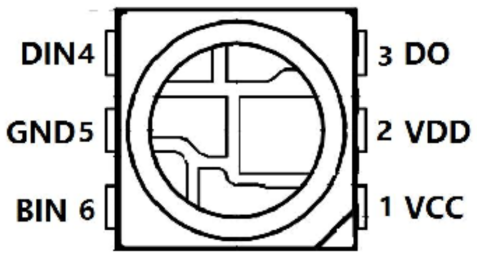
- dwurdzeniowy 32-bitowy mikrokontroler *Xtensa LX7*,
- 16 MB pamięci flash, 16MB pamięci PSRAM,
- 512 KB pamięci SRAM,
- 36 portów GPIO,
- komunikacja 2.4 GHz WiFi (802.11b/g/n) oraz Bluetooth 5,
- wbudowana antena,
- wsparcie dla: I2C, LED PWM, USB Serial,
- maksymalny pobór prądu 300-400 mA,
- zasilanie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

3.2 Adresowalne diody LED

Przyjęto założenie dotyczące wizualizacji 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 5 V wykazywały zauważalny spadek jakości odwzorowania kolorów przy liczbie elementów przekraczającej 100 (w ciągłej linii). Wykorzystanie ww. elementów niosłoby za sobą potrzebę dodatkowego wprowadzania napięcia do linii diod. Diody *WS2815B* zasilane napięciem 12 V 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 5 V odpowiednika.

Poniższy rysunek (Rys. 3.2) z dokumentacji producenta przedstawia schemat wyprowadzeń elementu [9].



Rysunek 3.2: Schemat wyprowadzeń adresowalnej diody LED RGB *WS2815B-V1*. Źródło: `https://www.alldatasheet.com/htmldatasheet2/1134588/WORLDSEMI/WS2815B/1139/2/WS2815B.png`

Własności elementu:

- napięcie pracy 12 V,
- pobór prądu 15 mA przy maksymalnej jasności każdej z 3 kolorowych diod,
- montaż SMT (z ang. *surface-mount technology*),
- częstotliwość odświeżania 4 kHz.

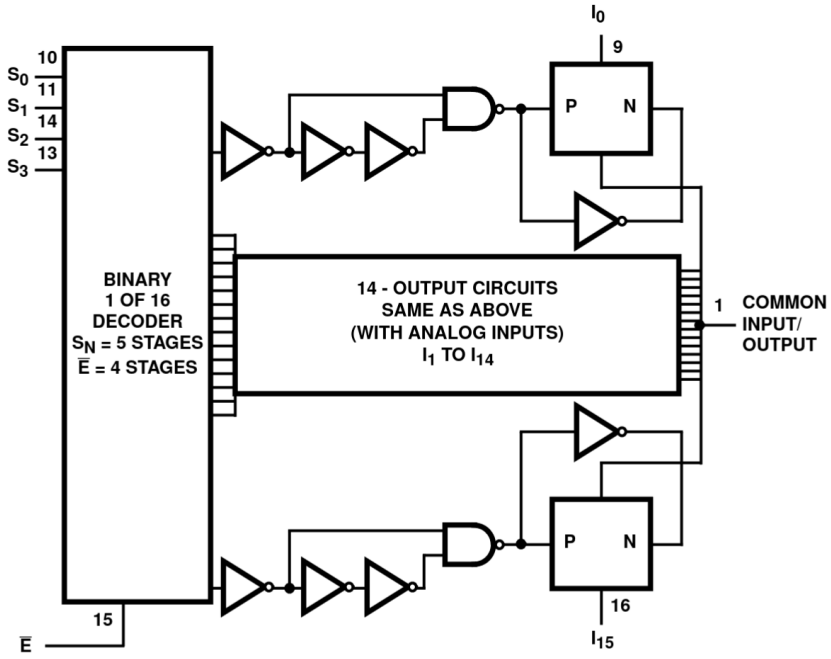
3.3 Przyciski

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* [7]. 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 [3]. 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 multiplekser. W tym celu wybrano 16 kanałowy multiplekser z wyprowadzeniem *COM*. Urządzenie to pozwala na odczyt stanu przycisku wprowadzając na wejście multipleksera 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 mikrokontrolera pozwala na odczyt aktualnego stanu przycisków. Z uwagi na niską cenę oraz niewielki rozmiar wybrano model *CD74HC4067M* firmy *Texas Instruments*.

Poniższy rysunek (Rys. 3.3) z dokumentacji producenta przedstawia diagram funkcjonalny układu [10].



Rysunek 3.3: Diagram funkcjonalny multipleksera *CD74HC4067M*. Źródło: `https://www.alldatasheet.com/html-pdf/517535/TI/CD74HC4067M/62/2/CD74HC4067M.html`

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

S0	**S1**	**S2**	**S3**	**COM**	**Kanał**
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

Tabela 3.1: Tablica prawdy multipleksera

3.4 Podświetlenie LED

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

- licznika rozkazów,
- akumulatora,
- rejestru instrukcji,
- rejestrów A oraz S,
- loga Politechniki Śląskiej.

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

3.5 Enkoder

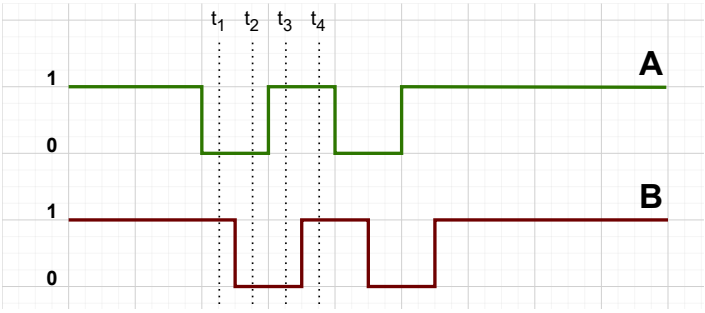
W celu zapewnienia możliwości wprowadzania wartości do: licznika rozkazów, akumulatora oraz rejestrów w lokalnym trybie działania urządzenia zastosowano enkoder obrotowy z przyciskiem.

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.

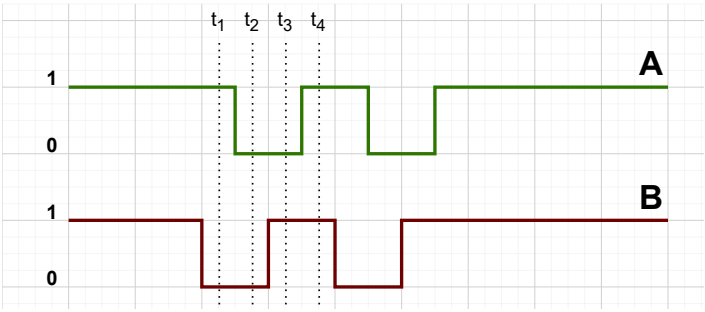
Aby określić kierunek obrotu elementu należy odczytać zmianę stanów na wyprowadzeniach urządzenia. Poniższe wykresy przedstawiają sposób odczytu obrotu w prawo (Rys. 3.4, Tab. 3.2) oraz w lewo (Rys. 3.5, Tab. 3.3).



Rysunek 3.4: Wykres stanów przy obrocie w prawo. Źródło: opracowanie własne.

Krok	A	B
1	0	1
2	0	0
3	1	0
4	1	1

Tabela 3.2: Tablica prawdy dla obrotu w prawo



Rysunek 3.5: Wykres stanów przy obrocie w lewo. Źródło: opracowanie własne.

Krok	A	B
1	1	0
2	0	0
3	0	1
4	1	1

Tabela 3.3: Tablica prawdy dla obrotu 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 mikrokontrolerowego pozwala na wykorzystanie przerwań w celu natychmiastowej zmiany trybu pracy urządzenia. Korekta drgań styków urządzenia została zapewniona z poziomu oprogramowania.

3.7 Serwer

Na potrzebę modernizacji narzędzia dydaktycznego, jakim jest Maszyna W, została stworzona strona internetowa symulująca działanie modelu komputera [12]. Projekt oraz wykonanie ww. strony internetowej nie jest częścią bieżącej pracy dyplomowej. Zadaniem SWIM jest udostępnianie ww. strony internetowej na lokalnej sieci utworzonej przez mikrokontroler.

Mikrokontroler wybrany w rozdziale 3.1 umożliwia udostępnianie złożonych stron internetowych z poziomu urządzenia. Aby było to możliwe planowano wykorzystać biblioteki takie jak:

- *WiFi* - obsługa połączeń i konfiguracji interfejsu sieci bezprzewodowej,
- *AsyncTCP* - asynchroniczna, nieblokująca komunikacja TCP (z ang. *Transmission Control Protocol*),
- *AsyncWebServer* - asynchroniczna obsługa serwera HTTP (z ang. *Hypertext Transfer Protocol*),
- *LittleFS* - system plików dla pamięci flash mikrokontrolera,
- *DNSServer* - lokalna obsługa i przekierowywanie zapytań DNS (z ang. *Domain Name Server*).

Wykorzystanie owych bibliotek pozwala na:

- kontrolę połączeń z użytkownikami oraz ograniczenie ich liczby,
- komunikację przy wykorzystaniu protokołu sieciowego TCP,
- przechowywanie złożonej strony internetowej Maszyny W bezpośrednio w pamięci flash mikrokontrolera,
- wykorzystanie portalu przechwytującego zapytania DNS w celu przekierowania użytkowników na wybraną stronę.

3.8 Punkt dostępowy

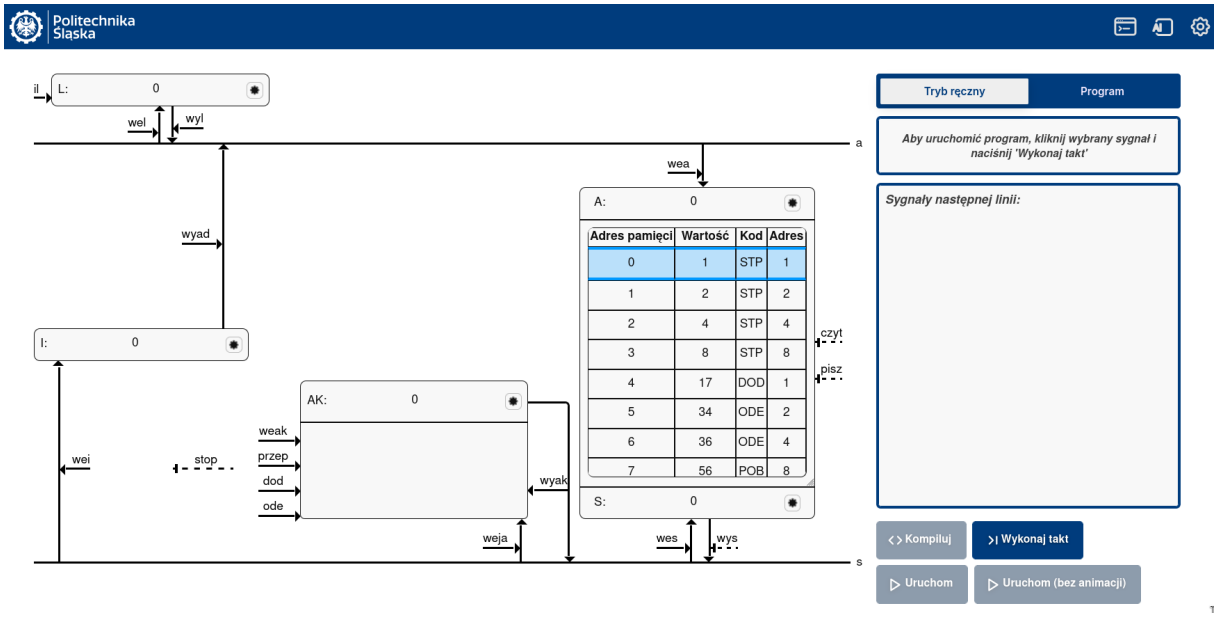
Wykorzystanie urządzenia sieciowego wymaga zabezpieczenia rozwiązania przed niechcianym dostępem oraz atakami. Spośród dwóch możliwych rozwiązań — udostępniania strony internetowej w sieci globalnej bądź lokalnej — wybrano rozwiązanie w pełni lokalne. Aby było to możliwe mikrokontroler tworzy własną sieć internetową na której następnie serwuje stronę internetową.

Tworzenie własnego punktu dostępu oraz izolacja go od sieci globalnej zapewnia bezpieczeństwo oraz kontrolę dostępu dla urządzenia. Bibioteki wymienione w rozdziale 3.7 umożliwiają zadanie hasła punktowi dostępu oraz ograniczenie liczby połączonych na raz użytkowników.

3.9 Strona internetowa

Bieżąca praca dyplomowa wykorzystuje wykonaną na potrzeby projektu PBL (z ang. *Project-Based Learning*), którego SWIM również był częścią, stronę modernizującą symulator Maszyny W. Niezwykle ważnym jest zaznaczyć, iż projekt oraz wykonanie ww. strony internetowej nie są częścią bieżącej pracy dyplomowej. Została ona stworzona przez członków projektu naukowego zajmującego się modernizacją Maszyny W [12]. Mając na uwadze autorów tej części projektu należy zrozumieć jej działanie, aby w odpowiedni sposób wdrożyć ją do projektu.

Strona umożliwia naukę działania Maszyny W w intuicyjny oraz łatwo dostępny sposób. Na Rys. 3.6 widnieje strona główna symulatora.



Rysunek 3.6: Strona główna sieciowego symulatora Maszyny W. Źródło: `https://maszynaw.aei.polsl.pl`

Zakłada się wizualizację obecnych na powyższym rysunku elementów symulatora, takich jak: licznik, akumulator, czy pamięć operacyjna przy pomocy wymienionych w rodziale 3.2 wyświetlaczy. Kontrolę sygnałów takich jak: czyt, wys, wei, itd. umożliwią opisane w rozdziale 3.3 przyciski.

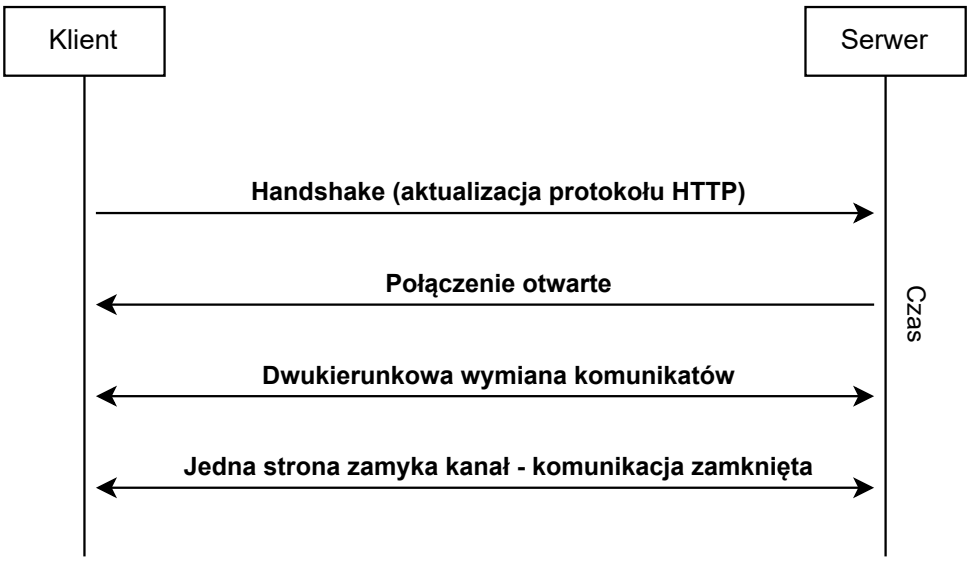
3.10 Interfejs komunikacyjny

Niezbędnym elementem bieżącego projektu jest możliwość komunikacji pomiędzy mikrokontrolerem, a udostępnianą na nim stroną w czasie rzeczywistym. Wyjściowo, nie jest możliwe przesyłanie informacji o aktualnie znajdujących się wartościach w polach tekstowych strony internetowej lub o aktualnie wybranych sygnałach.

Standardowa implementacja serwera posiada jedynie możliwość komunikacji za pomocą protokołu HTTP. Wykorzystanie takiego rozwiązania wymagałoby wysyłania cyklicznych żądań do serwera co powodowałoby wysokie zużycie zasobów oraz opóźnienia w działaniu. W celu zapewnienia wydajnej oraz szybkiej komunikacji pomiędzy stroną internetową a serwerem wykorzystano technologię *WebSocket*.

Websocket jest dwukierunkowym protokołem komunikacyjnym. Działa zarówno w warstwie aplikacji (strona internetowa) oraz na serwerze (mikrokontroler). Umożliwia wymianę danych w czasie rzeczywistym, bez konieczności każdorazowego inicjowania żądania przez użytkownika. Wykorzystuje asynchroniczną komunikację, co pozwala na nieblokujące odbieranie i nadawanie złożonych informacji [5].

Poniższy schemat ilustruje przebieg komunikacji *WebSocket*, pomiędzy klientem (stroną internetową), a serwerem.



Rysunek 3.7: Schemat komunikacji przy wykorzystaniu protokołu *WebSocket*. Źródło: opracowanie własne.

3.11 Zasilanie

Mając na uwadze wszystkie wymienione w powyższym rozdziale elementy SWIM, dokonano analizy wymagań energetycznych układu w celu dobrania odpowiedniego zasilacza. Adresowalne diody LED oraz podświetlenie wymagały zasilania napięciem 12 V, natomiast mikrokontroler *ESP32* oraz multiplekser korzystały z napięcia 3,3 V.

Przeprowadzono szacunkowe obliczenia maksymalnego poboru prądu:

- dla linii diod LED *WS2815B-V1*, przy założeniu pełnej jasności każdej z trzech diod RGB oraz wykorzystaniu 600 elementów, przewidziano maksymalny pobór prądu na poziomie około 9 A,
- dla podświetlenia LED przewidziano pobór prądu rzędu 0,5 A, uwzględniając wszystkie elementy podświetlane jednocześnie,
- dla mikrokontrolera *ESP32-S3* oraz układów towarzyszących przewidziano pobór prądu około 600 mA przy pełnym obciążeniu interfejsów peryferyjnych i aktywnym WiFi.

Na podstawie powyższych obliczeń dobrano zasilacz impulsowy o napięciu wyjściowym 12 V i mocy minimalnej 330 W, co zapewniało bezpieczny margines względem przewidywanego maksymalnego poboru prądu. Zastosowanie zasilacza o większej mocy umożliwiało również stabilne działanie układu przy chwilowych skokach poboru prądu przez linie LED.

Zasilanie mikrokontrolera *ESP32* oraz innych układów niskonapięciowych realizowano przy użyciu przetwornicy DC-DC obniżającej napięcie z 12 V do 3,3 V. Rozwiązanie to zapewniało izolację układów niskonapięciowych od wysokoprądowych linii LED oraz stabilne napięcie zasilania, eliminując spadki i zakłócenia w pracy mikrokontrolera. Dodatkowo, w celu ochrony układu przed odwrotną polaryzacją, należy zapewnić odpowiednią ochronę. Takie podejście zapewniło bezpieczne i niezawodne działanie całego systemu SWIM.

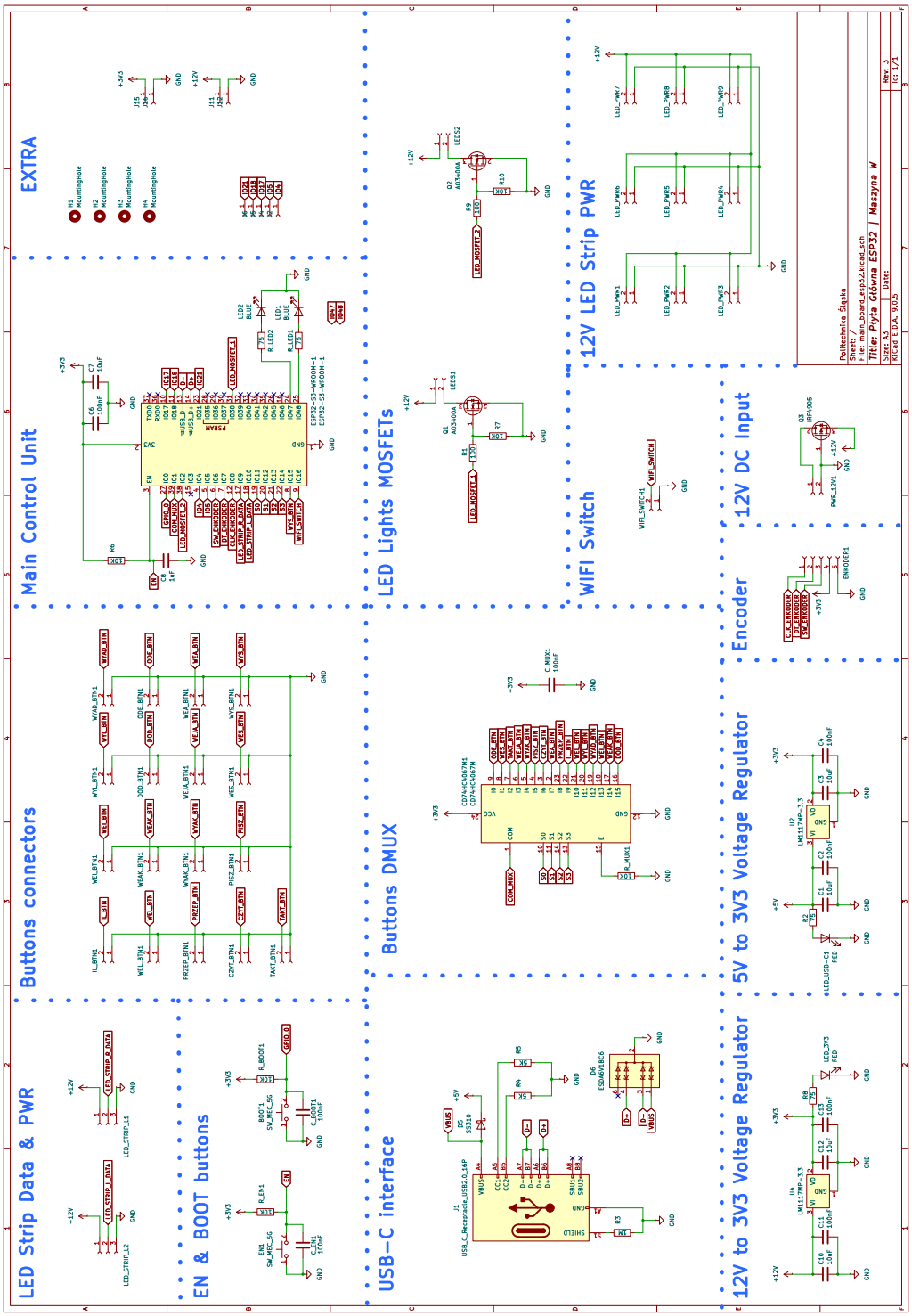
Rozdział 4

Schematy bloków systemu

Poniższy rozdział ma na celu prezentację schematów elektronicznych dla systemu wizualizacji stanu interfejsu Maszyny W. Schematy zostały wykonane przy użyciu otwartego oraz darmowego programu *KiCAD*. Rysunki zostały sformatowane w widoczny sposób z uwagi na ich znaczący rozmiar. Rysunki zostały przygotowane w formacie papieru A3.

4.1 Płyta główna SWIM

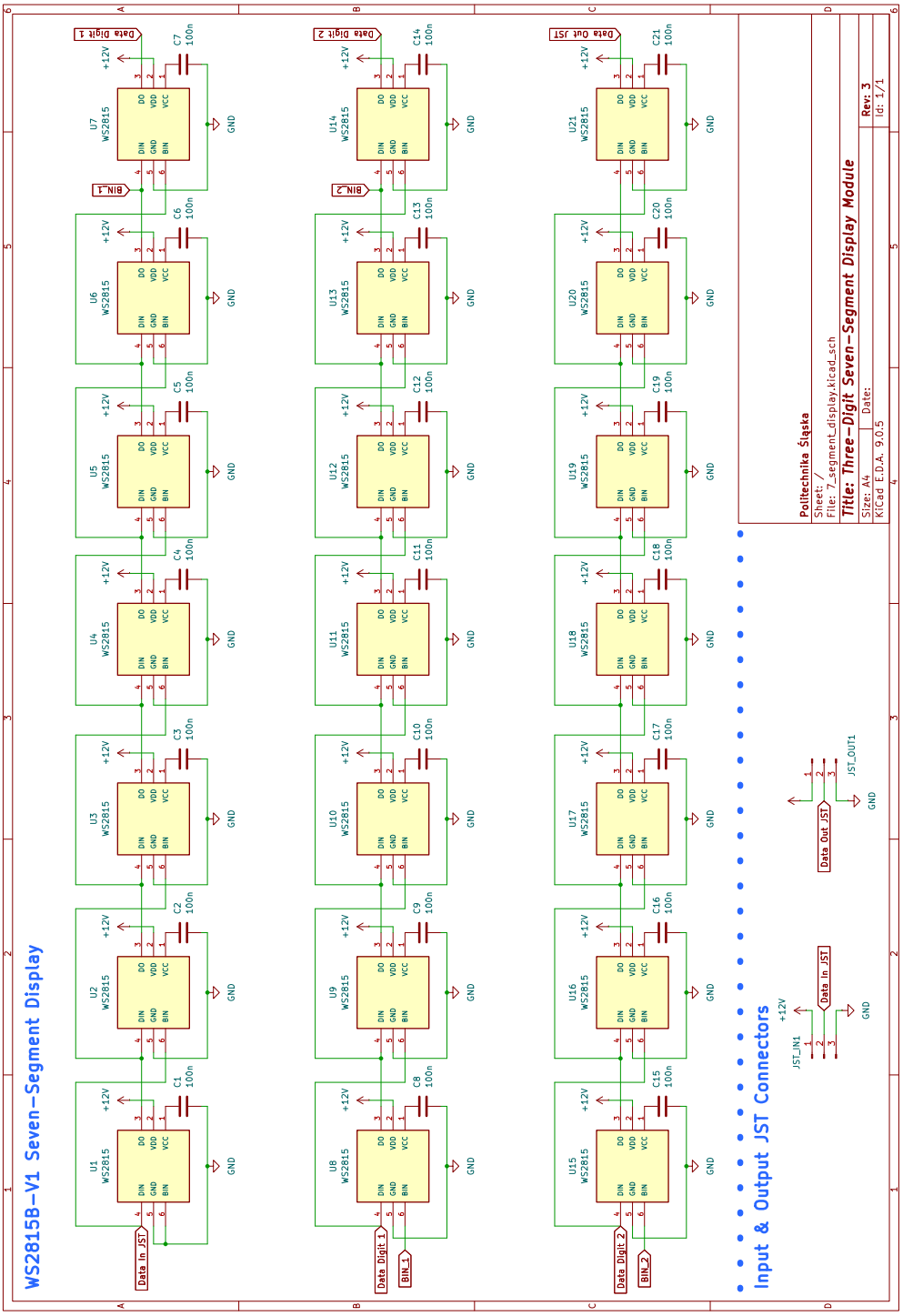
Na rysunku (Rys. 4.1) przedstawiono schemat płyty głównej systemu wizualizacji interfejsu Maszyny W.



Rysunek 4.1: Schemat płyty głównej z mikrokontrolerem *ESP32-S3*. Źródło: opracowanie własne.

4.2 Wyświetlacz siedmiosegmentowy trzycyfrowy

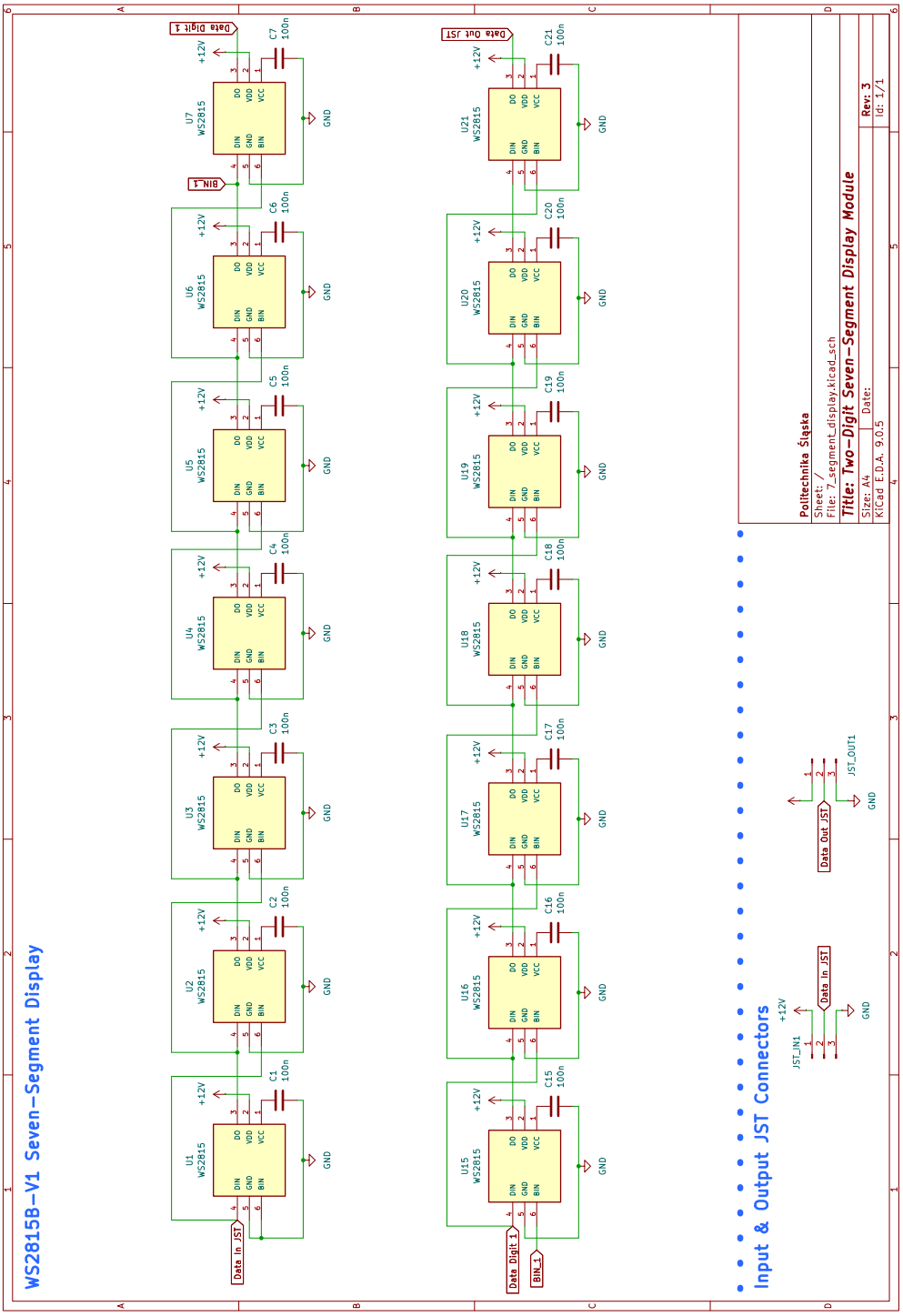
Na rysunku (Rys. 4.2) przedstawiono schemat wyświetlacza trzycyfrowego skonstruowanego z połączonych ze sobą w linii ciągłej diod LED.



Rysunek 4.2: Wyświetlacz siedmiosegmentowy trzycyfrowy. Źródło: opracowanie własne.

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 dwucyfrowy. Źródło: opracowanie własne.

Rozdział 5

Część konstrukcyjna

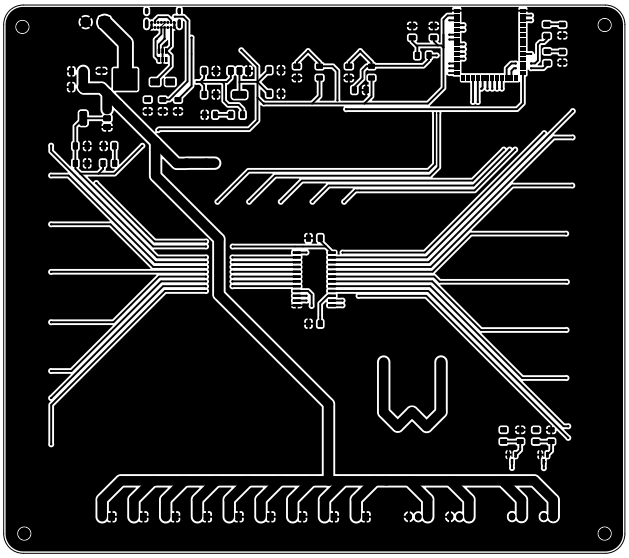
5.1 Obwody drukowane

Obwody drukowane zostały wykonane na specjalne zamówienie przez zewnętrzną firmę *JLCPCB*. Mozaikę ścieżek płytek drukowanych zaprojektowano na bazie schematów z rozdziału 4. Wykorzystano w tym celu 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

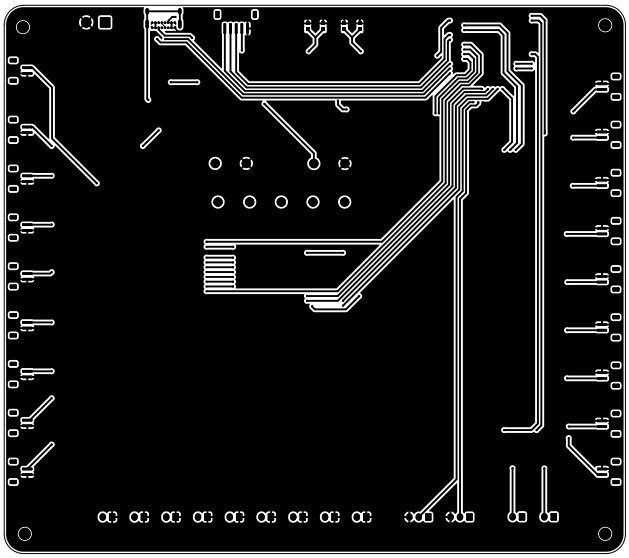
Na rysunkach 5.1 oraz 5.2 przedstawiono mozaiki obwodów drukowanych płyty głównej SWIM.

Front płytki zapewnia możliwość przymocowania elementów urządzenia wyłączając wyprowadzenia oraz porty. Rysunek 5.1 przedstawia frontową mozaikę ścieżek płytki pomniejszoną dwukrotnie względem wymiarów rzeczywistych.



Rysunek 5.1: Widok warstwy górnej płyty głównej. Źródło: opracowanie własne.

Rewers płytki zapewnia możliwość przymocowania wyprowadzeń oraz portów urządzenia. Umożliwia również podłączenie dodatkowych elementów przy wykorzystaniu dostępnych wyprowadzeń dodatkowych. Rysunek 5.2 przedstawia tylną mozaikę ścieżek płytki pomniejszoną dwukrotnie względem wymiarów rzeczywistych.

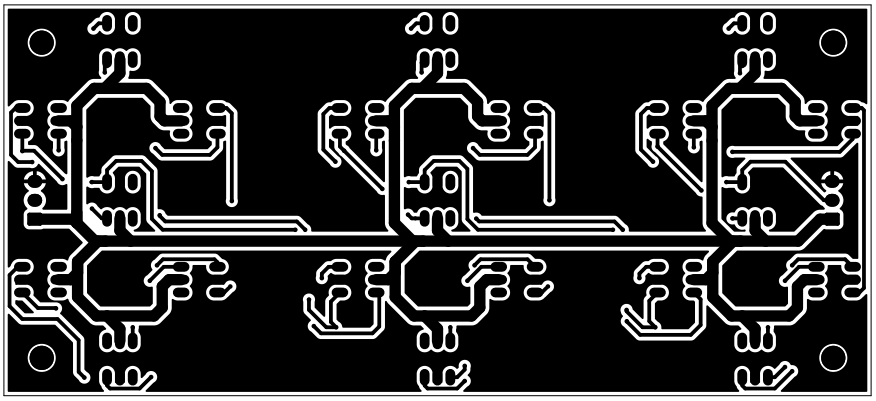


Rysunek 5.2: Widok warstwy dolnej płyty głównej. Źródło: opracowanie własne.

5.1.2 Wyświetlacze siedmiosegmentowe

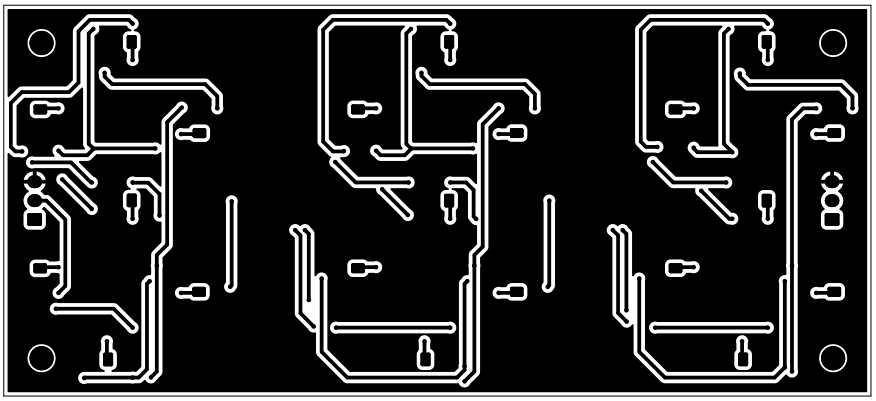
Na rysunkach 5.3 oraz 5.4 przedstawiono mozaiki obwodów drukowanych wyświetlacza trzycyfrowego.

Front płytki zapewnia możliwość przymocowania diod LED. Rysunek 5.3 przedstawia płytkę w rzeczywistych wymiarach.



Rysunek 5.3: Widok warstwy górnej wyświetlacza trzycyfrowego. Źródło: opracowanie własne.

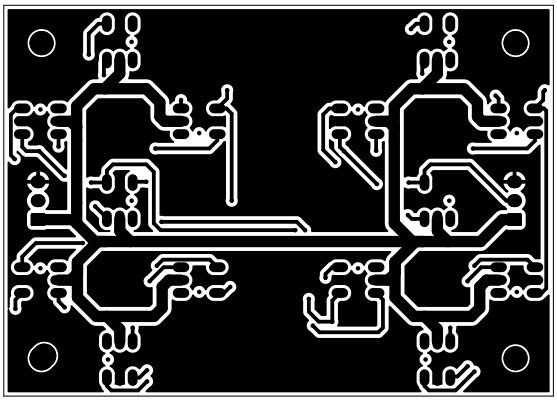
Rewers płytki zapewnia możliwość przymocowania kondensatorów filtrujących. Rysunek 5.4 przedstawia płytkę w rzeczywistych wymiarach.



Rysunek 5.4: Widok warstwy dolnej wyświetlacza trzycyfrowego. Źródło: opracowanie własne.

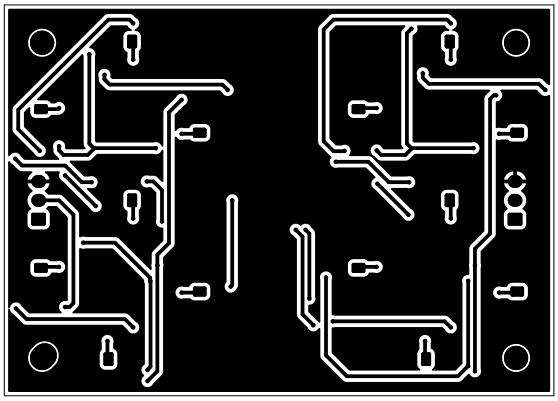
Na rysunkach 5.5 oraz 5.6 przedstawiono mozaiki obwodów drukowanych wyświetlacza dwucyfrowego.

Front płytki zapewnia możliwość przymocowania diod LED. Rysunek 5.5 przedstawia płytkę w rzeczywistych wymiarach.



Rysunek 5.5: Widok warstwy górnej wyświetlacza dwucyfrowego. Źródło: opracowanie własne.

Rewers płytki zapewnia możliwość przymocowania kondensatorów filtrujących. Rysunek 5.6 przedstawia płytkę w rzeczywistych wymiarach.



Rysunek 5.6: Widok warstwy dolnej wyświetlacza dwucyfrowego. Źródło: opracowanie własne.

5.2 Opis złącz

5.2.1 Płyta główna

Złącze zasilania

Urządzenie jest wyposażone w 2-wyprowadzeniową 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.

Złącze USB-C

W celu umożliwienia łatwego programowania i komunikacji z mikrokontrolerem, płytkę wyposażono w złącze USB (z ang. *Universal Media Bus*) typu C (tryb USB 2.0). Na złącze wyprowadzono jedynie linie $D+$ oraz $D-$. Linie zasilania *VBUS* są wykorzystywane do zasilania sekcji USB oraz wykrywania podłączenia komputera. Aby zapewnić zabezpieczenie przed prądem wstecznym wykorzystano diodę *Schottky'ego* połączoną w kierunku zaporowym na linii zasilania. Pozostałe wyprowadzenia USB-C niewykorzystywane przez układ zostały pozostawione niepodłączone, zgodnie ze specyfikacją [4].

Złącze enkodera

Do podłączenia impulsatora zastosowano 5-wyprowadzeniowe 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 mikrokontrolera. Redukcję drgań styków zapewniono z poziomu oprogramowania.

Złącza przycisków

Każdy przycisk posiada dedykowane 2-wyprowadzeniowe złącze typu JST-ZH o rastrze 1.5 mm. Jedno wyprowadzenie każdego złącza podłączone jest do masy *GND*, drugie — do wejścia cyfrowego mikrokontrolera. Linie sygnałowe wyposażono w rezystor polaryzujący do zasilania z poziomu wbudowanych w mikrokontroler rezystorów. Filtrację sygnałów zapewniono od strony oprogramowania.

Złącze przełącznika trybów

Przełącznik trybów posiada dedykowane 2-wyprowadzeniowe złącze typu JST-ZH o rastrze 1.5 mm. Jedno wyprowadzenie złącza podłączone jest do masy *GND*, drugie — do wejścia cyfrowego mikrokontrolera wykorzystującego wbudowany w mikrokontroler rezystor polaryzujący do zasilania. Filtrację sygnałów zrealizowano programowo.

Złącza linii LED

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

- *+12 V* — zasilanie diod,
- *DATA* — linia danych,
- *GND* — masa układu.

Dodatkowo zastosowano kondensator filtrujący do masy na każdej z adresowalnych diod obecnych na płytkach drukowanych, zgodnie z dokumentacją producenta [9].

Złącza zasilania podświetlenia modelu

Płytka zawiera dodatkowe dwa 2-wyprowadzeniowe złącza typu JST-XH o rastrze 2.5 mm. Każde złącze wspiera kontrolę zasilania 12 V z poziomu mikrokontrolera. 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. Wyprowadzenia:

- *+12 V* — zasilanie diod,
- *DATA_IN/DATA_OUT* — wejście/wyjście linii danych,
- *GND* — masa układu.

Rozdział 6

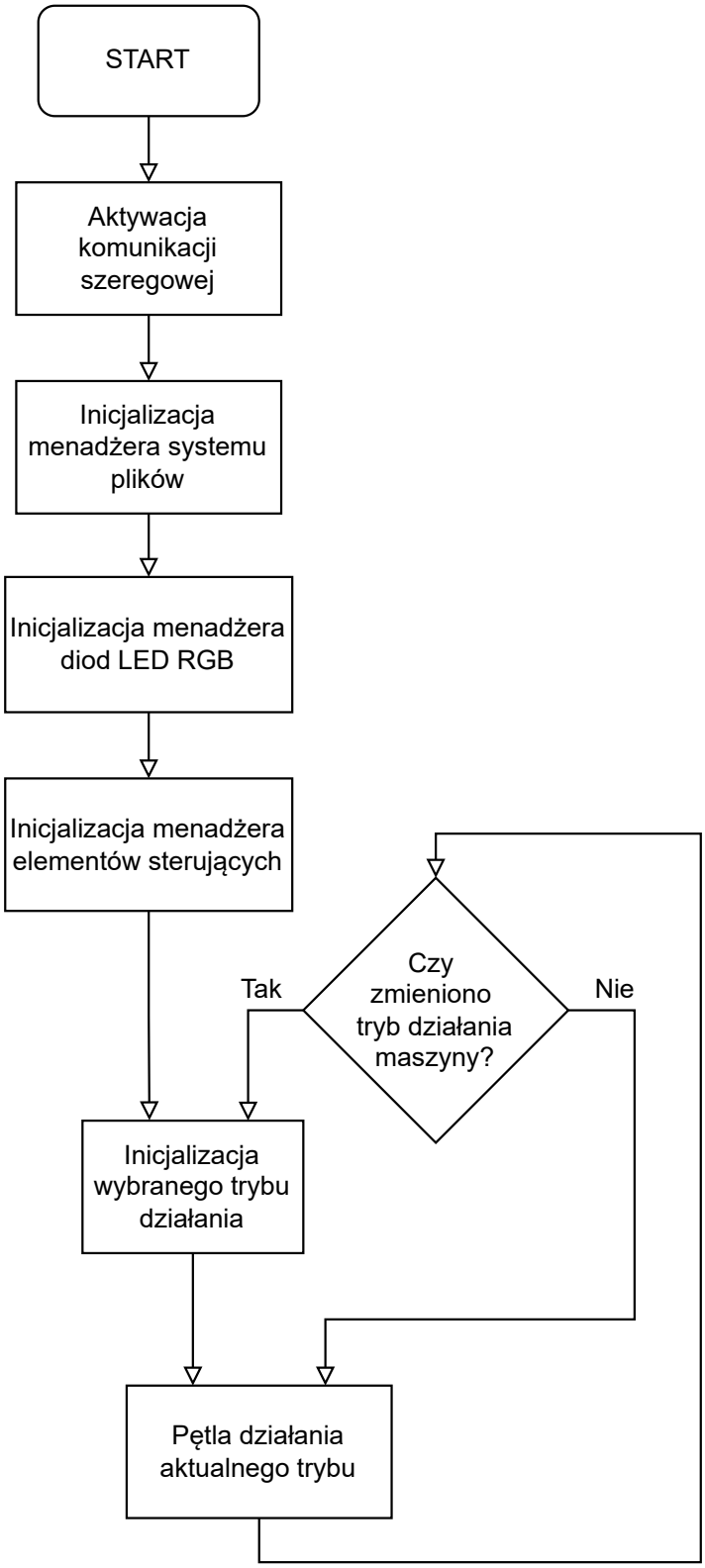
Opis oprogramowania

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

- obsługa 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.

Poniższe rozdziały opisują w szczegółowy sposób działanie oprogramowania urządzenia. Rozdziały 6.1, 6.3 oraz 6.4 opisują działanie elementów oprogramowania funkcjonujących zarówno w trybie sieciowym, jak i lokalnym urządzenia. Opis działania owych elementów jest kluczowy do zrozumienia funkcjonowania obu trybów. Metody zarządzania ww. elementami oraz sposób działania obu trybów SWIM zostały omówione w rozdziałach 6.5 oraz 6.6.

Poniższy schemat (Rys. 6.1) przedstawia punkt wejściowy programu urządzenia. Rozszerzony opis działania widocznych na nim elementów znajduje się w ww. rozdziałach.



Rysunek 6.1: Ogólny diagram sekwencji działania oprogramowania. Źródło: opracowanie własne.

6.1 Obsługa adresowalnych diod LED

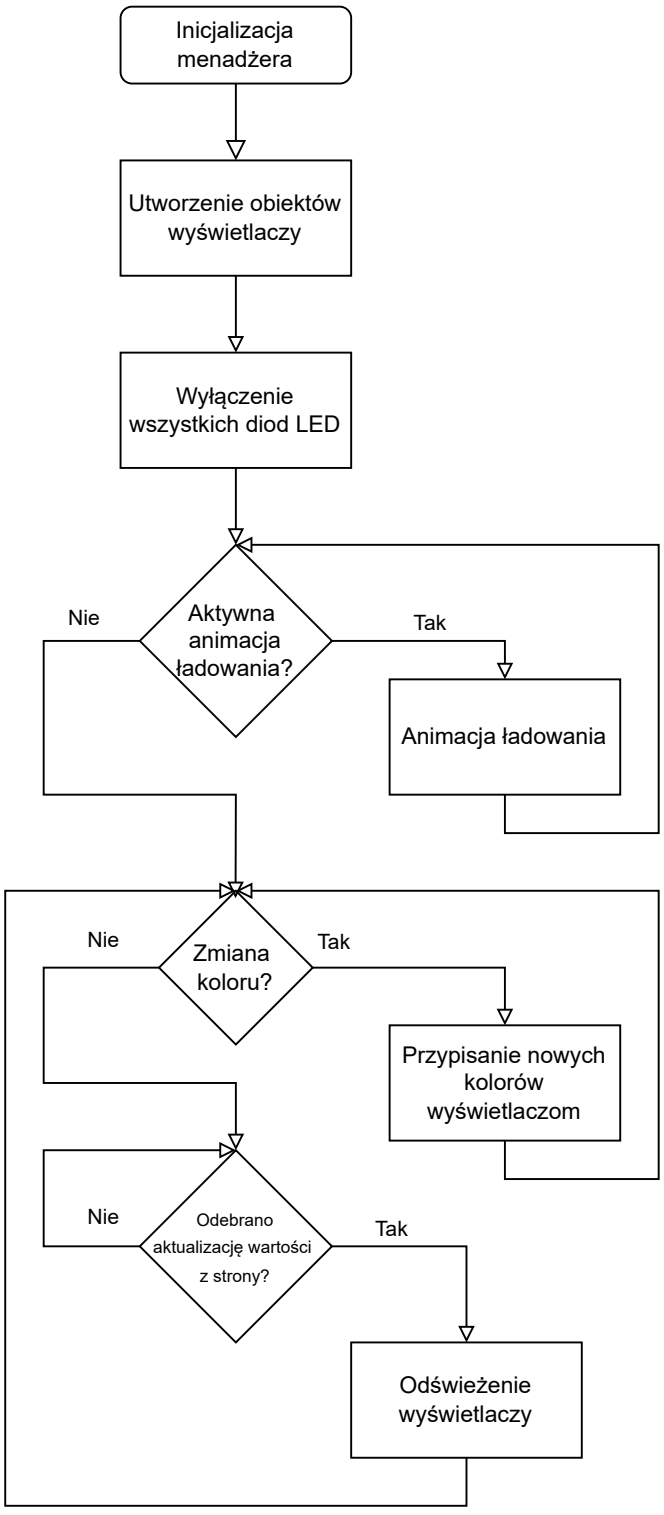
Stworzenie wydajnego oraz łatwego w rozbudowie oprogramowania wymagało zaprojektowania odpowiedniej architektury kodu. W tym celu stworzono obiekt *DisplayManager*. Jest on odpowiedzialny za zarządzanie wszystkimi adresowalnymi diodami LED urządzenia. Implementacja obejmuje następującą funkcjonalność:

- tworzenie/usuwanie wyświetlaczy,
- zmianę koloru poszczególnych elementów,
- zadawanie wartości do wyświetlenia elementom,
- odświeżanie koloru wszystkich elementów naraz,
- wyświetlanie animacji testowych/ładowanie.

Podczas tworzenia implementacji dla obsługi diod zdecydowano się na abstrakcyjne rozwiązanie problemu wielu różnych typów wyświetlanych wartości. Powstała w tym celu klasa wirtualna *LedElement* pozwala na proste odnalezienie elementu w linii diod LED, dynamiczne zarządzanie pamięcią urządzenia oraz tworzenie klas pochodnych dla specyficznych wartości takich jak: sygnał, magistrala, czy pamięć operacyjna. Sposób wyświetlania wartości liczbowych przy użyciu diod LED opisano w rodziale 6.2.

Wykorzystanie abstrakcyjnego podejścia w architekturze oprogramowania pozwoliło w łatwy sposób odizolować i wykryć błędy. Pozwala na łatwą rozbudowę lub modyfikację układu wyświetlaczy.

Implementacja menadżera w ostatecznej wersji oprogramowania wywołuje metodę odświeżającą wszystkie diody LED zgodnie z poniższym schematem blokowym (Rys. 6.2). Widoczny na rysunku schemat ma na celu prezentację podstawowych funkcji zarządzania diodami LED.

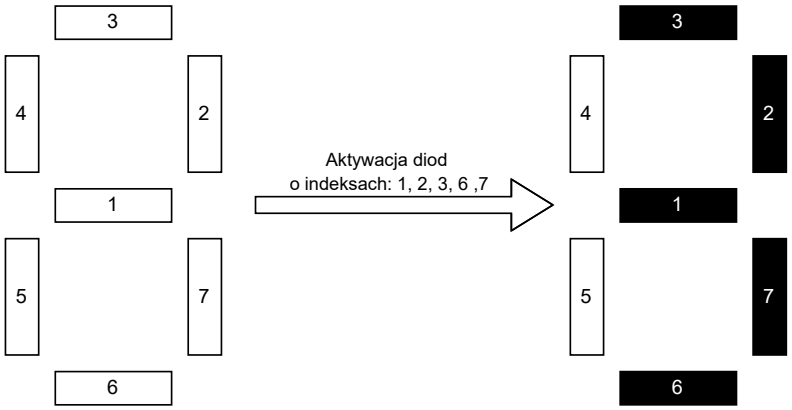


Rysunek 6.2: Diagram sekwencji działania menadżera diod LED. Źródło: opracowanie własne.

6.2 Obsługa wyświetlaczy segmentowych

Stworzenie czytelnego oraz łatwego w obsłudze oprogramowania wymagało specjalnego obiektu zarządzającego diodami LED wyświetlającymi wartości liczbowe. Model Maszyny W został skonstruowany z 9 wyświetlaczy 3-cyfrowych oraz 8 wyświetlaczy 2-cyfrowych. Zarządzanie 43 cyframi wymagało stworzenia obiektu przechowującego indeksy diod odpowiedzialnych za daną wartość liczbową.

Powyżej opisany problem należało rozwiązać na etapie konstrukcji schematów elektrycznych wyświetlaczy (rozdziały 4.2 oraz 4.2). Rysunek 6.3 przedstawia kolejność połączenia diod w wyświetlaczu segmentowym. Wyjście ostatniej diody segmentu jest podłączone do wejścia pierwszej diody kolejnego segmentu lub elementu wyświetlacza.



Rysunek 6.3: Wizualizacja schematu działania wyświetlaczy segmentowych. Źródło: opracowanie własne.

Obiekt *Segment* obecny w oprogramowaniu SWIM jest odpowiedzialny za zarządzanie diodami LED wchodzącymi w skład segmentu. Przechowuje indeksy diod, oraz zawiera implementację metody wyświetlania cyfr.

6.3 Obsługa przycisków, enkodera, diod płyty głównej oraz przełącznika trybów

Podobnie jak w przypadku obsługi diod LED (rozdział 6.1), przygotowano rozwiązanie umożliwiające zarządzanie wszystkimi elementami sterującymi. W tym celu utworzono obiekt *HumanInterface* nakładający warstwę abstrakcji na niskopoziomowe operacje związane z obsługą wejść urządzenia.

Obiekt *HumanInterface* jest odpowiedzialny za ustawienie wyprowadzeń mikrokontrolera. Przy inizjalizacji ustawia wyprowadzenia jako wyjście dla multipleksera, przycisku *takt*, przełącznika trybów oraz enkodera. Inicjalizowane również są wyprowadzenia dla dwóch diod LED obecnych na płycie głównej.

Obiekt posiada dedykowane metody do obsługi następujących funkcji urządzenia:

- obrót enkodera,
- zwracanie stanu przycisku enkodera,
- multipleksowanie przycisków,
- zwracanie stanu przycisków,
- zwracanie stanu przełącznika trybów,
- eliminacja drgań styków,
- kontrola diod LED płyty głównej,
- kontrola podświetlenia LED modelu.

6.4 Obsługa pamięci nieulotnej/systemu plików

Rozmiar plików strony internetowej udostępnianej przez urządzenie oraz potrzeba aktualizacji strony wymagała stworzenia sposobu na zapis plików w pamięci nieulotnej urządzenia. Dodatkowo wykorzystanie diod LED RGB pozwala na personalizację kolorów poszczególnych elementów SWIM. Strona internetowa posiada opcję ustawienia koloru dla: wyświetlaczy segmentowych, sygnałów oraz magistrali. Aby zapewnić możliwość przechowywania wybranej konfiguracji również należało zapisać wartości w pamięci nieulotnej urządzenia.

Wyżej wymienione powody wymagały wykorzystania zewnętrznej biblioteki *LittleFS*. Pozwala ona stworzyć osobną partycję w pamięci flash urządzenia dla systemu plików [6]. Zapisane w ten sposób dane nie ulegają usunięciu po odłączeniu zasilania urządzenia.

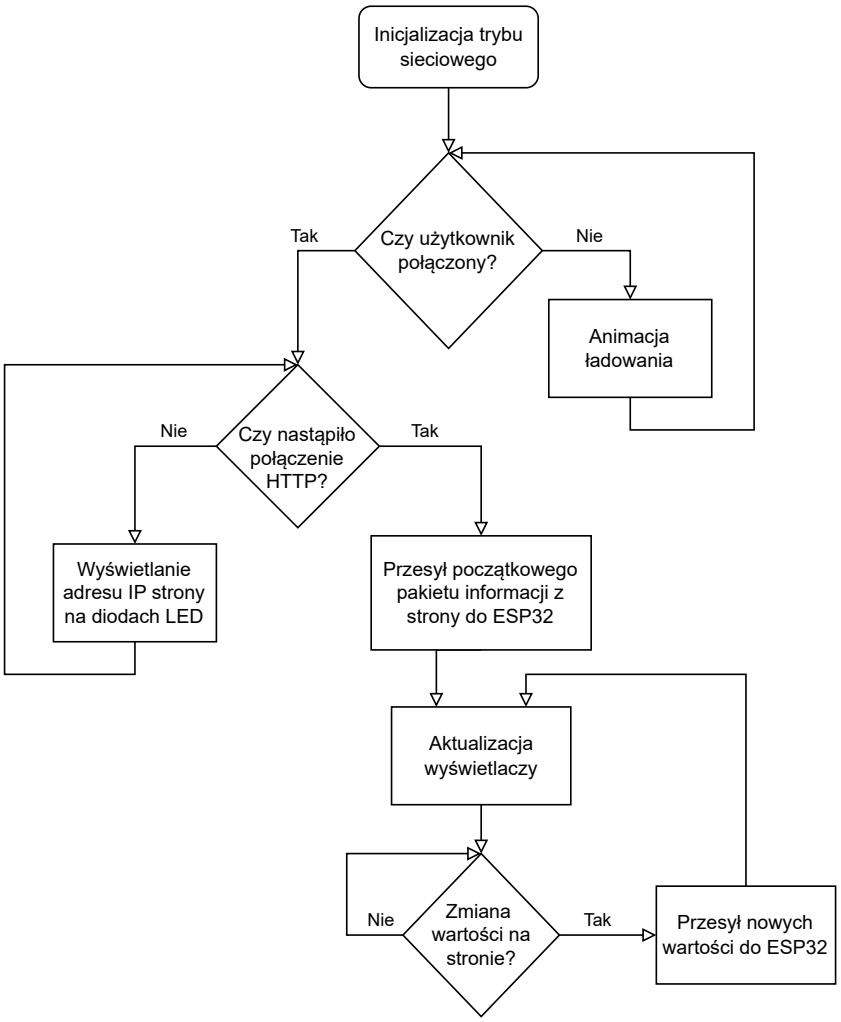
Urządzenie przechowuje w pamięci plik z konfiguracją kolorów w formacie *json*. Podczas uruchamiania urządzenia konfiguracja jest odczytywana, a kolory zostają odpowiednio ustawione. Użytkownik ma możliwość nadpisania pliku z konfiguracją wykorzystując odpowiednią opcję w ustawieniach strony internetowej udostępnianej przez urządzenie.

6.5 Obsługa trybu sieciowego SWIM

Głównym trybem działania urządzenia jest wizualizacja interfejsu sieciowego symulatora Maszyny W. Powstały w tym celu obiekt *W_Server* jest odpowiedzialny za zarządzanie następującymi funkcjami:

- tworzenie punktu dostępowego,
- tworzenie serwera DNS,
- udostępnianie strony internetowej,
- obsługa komunikacji pomiędzy stroną internetową, a serwerem.

Poniższy schemat blokowy (Rys. 6.4) przedstawia uproszczony sposób działania trybu sieciowego.



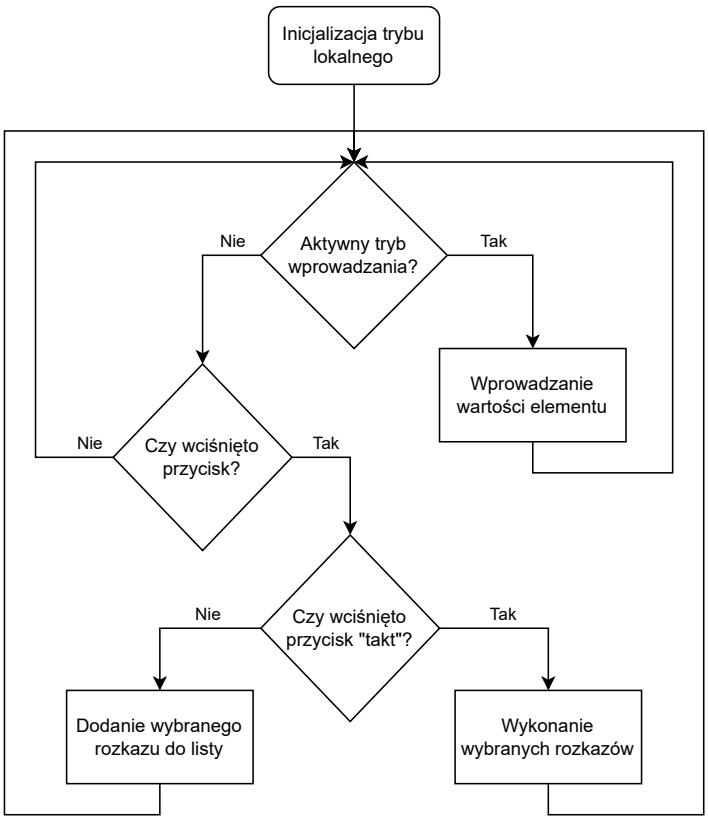
Rysunek 6.4: Uproszczony diagram sekwencji działania trybu sieciowego. Źródło: opracowanie własne.

Sposób działania pomija szczegóły zwiazane z inicjalizacją poszczególnych komponentów takich jak: tworzenie serwera DNS, czy udostępnianie strony internetowej.

6.6 Obsługa trybu lokalnego SWIM

Dodatkowym trybem działania urządzenia jest symulacja działania Maszyny W bezpośrednio na urządzeniu. Powstały w tym celu obiekt *W_Local* działa analogicznie do obiektu *W_Server*. Podobnie jak menadżer trybu sieciowego wykorzystuje obiekty *DisplayManager*, *HumanInterface* oraz *FileSystem*.

Poniższy schemat blokowy (Rys. 6.5) prezentuje uproszczony sposób działania trybu sieciowego.



Rysunek 6.5: Uproszczony diagram sekwencji działania trybu lokalnego. Źródło: opracowanie własne.

Sygnały wybierane poprzez naciśnięcie odpowiednich przycisków zostają dodane do listy sygnałów. Obiekt posiada funkcję wykrywającą, czy Maszyna W pozwala na daną kombinację rozkazów. Przykładem wykluczających się wzajemnie rozkazów są: *czyt/pisz*, *wyak/wys* lub *wyl/wyad*. W przypadku wykrycia wciśnięcia niedozwolonego przycisku urządzenie nie doda rozkazu do listy. Po wciśnięciu przycisku *takt* obiekt wykona rozkazy z listy w kolejności dodania.

W celu zapewnienia poprawnego działania trybu lokalnego wprowadzono zabezpieczenie przed przekroczeniem wartości możliwej do wyświetlenia na wyświetlaczach dwu oraz trzycyfrowych. W przypadku przekroczenia zadanego limitu liczba zostanie wyrównana do największej/najmniejszej możliwej do wyświetlenia.

Rozdział 7

Obsługa

Obsługa SWIM jest zależna od wybranego trybu działania urządzenia. W poniższych dwóch rozdziałach opiszę sposób obsługi urządzenia w obu trybach.

Po podłączeniu urządzenia do zasilacza 12 V urządzenie uruchomi się w jednym z trybów. Wybór jest zależny od ustawienia przełącznika trybów (Rozdział 3.6). Przy zmianie stanu przełącznika urządzenie automatycznie zainicjuje oraz zmieni swój tryb działania.

7.1 Tryb sieciowy

W przypadku wybrania trybu sieciowego:

1. Urządzenie udostępnia użytkownikowi dedykowaną sieć lokalną o nazwie *Maszyna W(eb)*. W czasie oczekiwania na połączenie użytkownika do sieci na wyświetlaczach segmentowych modelu wyświetlana będzie animacja ładowania w postaci obracających się okręgów.

2. Na wybranym przez użytkownika urządzeniu (komputer, telefon, tablet, itp.) należy połączyć się z udostępnianą siecią poprzez wpisanie hasła *czytwysweiil*. Poprawne połączenie do sieci skutkuje wyłączeniem animacji ładowania oraz wyświetleniem na wyświetlaczach *PaO* adresu IP strony internetowej.

3. *(OPCJONALNE)* Jeżeli urządzenie wybrane przez użytkownika wspiera funkcjonalność portalu uwierzytelniającego urządzenie użytkownika powinno wyświetlić powiadomienie o potrzebie zalogowania się do sieci. Wybierając tę funkcję użytkownik zostaje przekierowany na stronę internetową symulatora Maszyny W.

4. Jeżeli urządzenie użytkownika nie wspiera wymienionej w powyższym punkcie funkcjonalności, należy wpisać w polu adresowym przeglądarki wyświetlany na *PaO* adres IP.

5. Poprawne połączenie ze stroną internetową poskutkuje wyświetleniem na wyświetlaczu wartości obecnych na interfejsie strony. W prawym górnym rogu strony znajduje się przycisk informujący o poprawnym połączeniu z SWIM.

6. Na tym etapie użytkownik ma możliwość sterowania urządzeniem w sposób dowjaki. Wykorzystując:

   - **Interfejs strony internetowej** — Zmieniając wartości rejestrów, bądź rozkazów ich wartości będą wyświetlane na urządzeniu w czasie rzeczywistym,
   - **Przyciski SWIM** — Wciśnięte przyciski rozkazów będą odwzorowane na interfejsie strony internetowej oraz wyświetlaczach modelu.

7. *(OPCJONALNE)* Użytkownik ma możliwość zmiany koloru wybranych elementów SWIM poprzez stronę internetową w następujący sposób:

   (a) Wybór menu ustawień.

   (b) W sekcji *Kolory LED* ustawień. Zadanie wybranym wyświetlaczom wartości koloru.

   (c) Wciśnięcie przycisku *Wyślij wszystkie kolory do ESP32.*

   Po przesłaniu wartości kolorów wyświetlacze SWIM zmienią kolor na wybrany przez użytkownika. Konfiguracja kolorów zostaje zapisana w pamięci urządzenia. W przypadku ponownego uruchomienia SWIM zostanie odczytana wcześniej zapisana konfiguracja.

7.2 Tryb lokalny

W przypadku wybrania trybu lokalnego:

1. Urządzenie przechodzi bezpośrednio do symulacji Maszyny W. Wszystkie linie rozkazów zostają wyłączone, a wartości w rejestrach oraz *PaO* wyzerowane.

2. Użytkownik ma możliwość wykonania następujących operacji:

   - **Przewijanie wyświetlanego fragmentu *PaO*** — wykonując obrót enkoderem w lewo/prawo zmieni wyświetlany na 4 wierszach *PaO* fragment pamięci operacyjnej.
   - **Skok do aktywnego adresu w *PaO*** — wciskając przycisk enkodera wyświetlacz *PaO* pokaże fragment pamięci operacyjnej zawierający adres zapiany w rejestrze *A*.
   - **Tryb wprowadzania** — aktywacja/dezaktywacja trybu następuje poprzez przytrzymanie przez 1 s przycisku enkodera. Aktywny tryb wprowadzania jest sygnalizowany miganiem jednego z rejestrów. Obrót enkodera w lewo/prawo zmniejsza/zwiększa wartość w wybranym (migającym) rejestrze. Wciśnięcie przycisku enkodera spowoduje zmianę wybranego wyświetlacza.
   - **Aktywacja rozkazów** — aktywacja/dezaktywacja rozkazów następuje poprzez wciśnięcie odpowiedniego przyciska rozkazu. Aktywacja jest sygnalizowana podświetleniem linii rozkazu. Wybrany rozkaz zostaje dodany do listy rozkazów do wykonania w takcie. Wybór niedozwolonej kombinacji rozkazów skutkuje brakiem aktywacji rozkazu.
   - **Wykonanie taktu** — wciśnięcie przycisku *TAKT* skutkuje wykonaniem wybranych przez użytkownika rozkazów. Przypisane im operacje (*czyt, wys, wei*, itp.) zostają wykonane w kolejności ich wyboru.

Rozdział 8

Podsumowanie

8.1 Uruchamianie układu

Proces uruchamiania urządzenia przebiegał wieloetapowo, obejmując zarówno warstwę sprzętową, jak i programową:

- **Programowanie mikrokontrolera:** Pierwszym krokiem było przygotowanie mikrokontrolera *ESP32* do pracy. Program został wgrany z poziomu środowiska *PlatformIO* po uprzednim wprowadzeniu układu w tryb *bootloadera* i podłączeniu go do komputera za pomocą przewodu USB-C.

- **Konfiguracja systemu plików:** Ważnym elementem konfiguracji programowej było przygotowanie struktury partycji pamięci flash pod system plików *LittleFS*. Było to niezbędne do poprawnego umieszczenia na urządzeniu plików źródłowych strony internetowej, która stanowi kluczowy element działania urządzenia.

- **Rozwiązywanie problemów sprzętowych:** Podczas testów części sprzętowej modelu wystąpiły trudności z poprawnym wyświetlaniem barw przez diody LED wyświetlaczy segmentowych. Część wyświetlaczy *PaO* wyświetlała błędne kolory, bądź nie uruchamiała się wcale. Po przeprowadzonej analizie stwierdzono, że przyczyną były błędy w przygotowaniu przewodów łączących poszczególne wyświetlacze, co skutkowało przekłamaniami w przesyłaniu sygnałów sterujących kolorami.

  Problem rozwiązano poprzez wymianę okablowania na nowe, wykonane z dużą precyzją przy użyciu dedykowanej zaciskarki do złącz JST. Zastosowanie profesjonalnego narzędzia zapewniło pewność połączeń elektrycznych i wyeliminowało błędy w odwzorowaniu barw.

- **Korekta oprogramowania:** Podczas testów oprogramowania modelu wystąpiły kolejne przekłamania kolorów, tym razem na wyświetlaczach sygnałów sterujących

oraz magistral. Test zakładał zadanie każdej pojedynczej diodzie LED koloru czerwonego. Wszystkie wyświetlacze segmentowe wyświetlały kolor czerwony, ale pozostałe elementy świeciły w kolorze zielonym. Po przeprowadzeniu analizy oprogramowania stwierdzono, że przyczyną przekłamań są paski LED, które wykorzystują inną wersję diod *WS2815* w układzie GRB w przeciwieństwie do tych wykorzystywanych w wyświetlaczach.

- **Weryfikacja komuniakcji:** Ostatnim etapem było sprawdzenie nawiązywania połączenia przez protokół *WebSocket*, co pozwoliło na synchronizację modelu fizycznego z symulatorem internetowym w czasie rzeczywistym.

8.2 Wnioski

Niniejsza praca inżynierska poświęcona była zaprojektowaniu i budowie systemu wizualizacji danych dla webowego interfejsu Maszyny W z wykorzystaniem mikrokontrolera *ESP32*. Głównym celem projektu było stworzenie fizycznego modelu, który w czasie rzeczywistym odwzorowuje stan rejestrów i sygnałów sterujących symulatora Maszyny W za pomocą adresowalnych diod LED oraz umożliwia interakcję z nim poprzez przyciski fizyczne.

8.2.1 Realizacja celów i wymagań

Wszystkie kluczowe cele pracy zostały zrealizowane, a system jest w pełni funkcjonalny:

- **Wizualizacja danych** — system skutecznie wyświetla zarówno wartości liczbowe oraz stany sygnałów sterujących na tablicy złożonej z ponad 600 adresowalnych diod LED RGB;
- **Interakcja fizyczna** — zaimplementowano obsługę 17 przycisków oraz enkodera obrotowego;
- **Komunikacja sieciowa** — *ESP32* pełni rolę punktu dostępowego i serwera HTTP, zapewniając dwukierunkową komunikację przez protokół *WebSocket*;
- **Tryb lokalny** — zrealizowano założenie pozwalające na pracę urządzenia w trybie autonomicznym, bez potrzeby połączenia z zewnętrznym komputerem.

8.2.2 Problemy napotkane w trakcie pracy

Podczas realizacji projektu napotkano kilka istotnych wyzwań technicznych, które wymagały modyfikacji pierwotnych założeń projektowych:

- **Problem z spadkami napięcia podświetlenia LED** — Podczas testów prototypu stwierdzono, że tranzystory MOSFET odpowiedzialne za sterowanie napięciem podświetlenia led zastosowane na płycie głównej nie działały zgodnie z oczekiwaniami (np. problemy z pełnym otwarciem kanału przy napięciu sterującym z *ESP32* lub nieprawidłowa charakterystyka przełączania). Aby zapewnić stabilne podświetlenie urządzenia podjęto decyzję o rezygnacji ze sterowania nimi z poziomu mikrokontrolera na rzecz stałego podłączenia diod LED do zasilania. Rozwiązanie to zapewnia stabilne podświetlenie, pozbywając użytkownika możliwości sterowania nim.
- **Spadki napięcia na linii diod LED RGB** — Spadki napięcia w linii diod LED. Duża liczba diod wymusiła przejście z zasilania 5 V na 12 V (model *WS2815B*), co wyeliminowało błędy w odwzorowaniu barw na końcach linii LED.
- **Ograniczona liczba wyprowadzeń *ESP32*** — Problem ten rozwiązano poprzez zastosowanie 16-kanałowego multipleksera analogowo-cyfrowego, co pozwoliło na oszczędność wyprowadzeń mikrokontrolera.

8.2.3 Kierunek dalszego rozwoju

Projekt SWIM był tworzony z zamysłem dalszego rozwoju jako narzędzie dydaktyczne. Bieżąca praca dyplomowa służy za dokumentację, a zarazem instrukcję obsługi urządzenia. Podczas pracy zanotowano następujące możliwe dalsze kierunki rozwoju projektu:

- **Poprawa projektu płyty głównej** — kolejna wersja urządzenia powinna uwzględnić problemy z zasilaniem podświetlenia LED.
- **Poprawa projektu wyświetlaczy** — wybrany model LED posiada dodatkową pomocniczą linię danych, która nie jest wykorzystana na wyprowadzeniach wyświetlaczy. Rewizja powinna wykorzystywać tę funkcjonalność.
- **Wsparcie dla rozszerzeń modelu dydaktycznego** — Maszyna W w swojej pełnej wersji posiada obsługę przerwań, dodatkowe sygnały sterujące oraz dodatkowe rejestry poszerzające jej funkcjonalność [7]. Kolejna wersja urządzenia mogłaby wspierać rozszerzenia Maszyny W.

Bibliografia

[1] Robert Tutajewicz Alina Momot. „Maszyna W – jak zaprojektować prosty rozkaz". W: *MINUT* (2019), s. 24–35.

[2] Robert Brzeski. „Prawidłowe tworzenie rozkazów asemblerowych dla Maszyny W cz.2". W: *MINUT* (2020), s. 136–148.

[3] *CHERRY MX BLUE Keyswitch Datasheet*. Cherry GmbH. 2025. URL: https://cherry.saas.contentserv.com/admin/rest/document/56?ContextIDs=22889&Language=1 (term. wiz. 12. 01. 2026).

[4] USB Implementers Forum. *USB Type-C® Cable and Connector Specification Revision 2.1*. 2024. URL: https://www.usb.org/document-library/usb-type-cr-cable-and-connector-specification-revision-21 (term. wiz. 12. 01. 2026).

[5] Alexey Melnikov Ian Fette. *The WebSocket Protocol*. Standard RFC 6455. 2011. URL: https://datatracker.ietf.org/doc/html/rfc6455 (term. wiz. 09. 01. 2026).

[6] ARM Ltd. *littlefs: A little fail-safe filesystem designed for microcontrollers*. 2023. URL: https://github.com/littlefs-project/littlefs (term. wiz. 09. 01. 2026).

[7] Robert Tutajewicz Mirosław Chłopek. *Ćwiczenia laboratoryjne z podstaw informatyki - Maszyna W*. Gliwice: Wydawnictwo Politechniki Śląskiej, 2001. ISBN: 83-7335-021-7.

[8] John von Neumann. *First Draft Report on the EDVAC*. Spraw. tech. Edited and corrected by Michael D. Godfrey (1992). Philadelphia: Moore School of Electrical Engineering, University of Pennsylvania, 1945. URL: https://people.csail.mit.edu/brooks/idocs/VonNeumann_EDVAC.pdf (term. wiz. 10. 01. 2026).

[9] Electronic Components Datasheet Search. *Alldatasheet*. 2026. URL: https://www.alldatasheet.com/datasheet-pdf/pdf/1134588/WORLDSEMI/WS2815B.html (term. wiz. 06. 01. 2026).

[10] Electronic Components Datasheet Search. *Alldatasheet*. 2026. URL: https://www.alldatasheet.com/datasheet-pdf/pdf/517535/TI/CD74HC4067M.html (term. wiz. 06. 01. 2026).

[11] Espressif Systems. *ESP32-S3 Datasheet | Espressif Documentation*. 2026. URL: https://documentation.espressif.com/esp32-s3_datasheet_en.html (term. wiz. 04. 01. 2026).

[12] Kacper Sikorski Sławomir Put Paweł Linek. *Maszyna W*. 2025. URL: https://maszynaw.aei.polsl.pl/ (term. wiz. 02. 01. 2026).

Dodatki

Spis skrótów i symboli

SWIM system wizualizacji interfejsu Maszyny W

JAL jednostka arytmetyczno-logiczna

RGB czerwony, zielony, niebieski (ang. *red, green, blue*)

LED dioda elektroluminescencyjna (ang. *light-emitting diode*)

SMT montaż powierzchniowy (ang. *surface-mount technology*)

HTTP protokół przesyłania dokumentów hipertekstowych (ang. *Hypertext Transfer Protocol*)

DNS system nazw domen (ang. *Domain Name Server*)

TCP protokół sterowania transmisją (ang. *Transmission Control Protocol*)

PBL metoda projektów (ang. *Project-Based Learning*)

USB uniwersalna magistrala szeregowa (ang. *universal serial bus*)

GND masa (ang. *ground*)

Lista dodatkowych plików, uzupełniających tekst pracy

W systemie do pracy dołączono dodatkowe pliki zawierające:

- kod źródłowy mikrokontrolera w pliku ZIP,
- pliki projektowe płytek drukowanych w pliku ZIP,

Spis rysunków

Spis tabel