



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Imię i nazwisko studenta: Wojciech Paderewski

Nr albumu: 184823

Poziom kształcenia: Studia pierwszego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Elektronika i telekomunikacja

Profil: Komputerowe systemy elektroniczne

PRACA DYPLOMOWA INŻYNIERSKA

Tytuł pracy w języku polskim: Budzik synchronizowany przez WiFi

Tytuł pracy w języku angielskim: Alarm clock synchronized via WiFi

Opiekun pracy: dr hab. inż. Paweł Wierzba

Streszczenie

Głównym celem jaki został postawiony jest zrealizowanie budzika który synchronizuje się z wyko- rzystaniem zewnętrznego serwera czasu. ctd.

Słowa kluczowe: Nixie, ESP32, Wi-Fi, Home Assistant, NTP **Dziedzina nauki i techniki,**
zgodne z wymogami OECD: nauki inżyniersko-techniczne: automatyka, elektronika, elektro-
technika i technologie kosmiczne

Abstract

The main goal was to create an alarm clock that synchronizes with the use of using an external time server. ctd.

Keywords: Nixie, ESP32, Wi-Fi, Home Assistant, NTP

Spis treści

Wykaz ważniejszych oznaczeń i skrótów	7
1 Wstęp i cel pracy	9
1.1 Wstęp	9
1.2 Cel pracy	9
2 Lampy nixie	10
2.1 Zasada działania	10
2.2 Problemy związane z wykorzystaniem lamp nixie	10
2.3 Rozwiązania problemów związanych z lampami nixie	11
2.4 Sterowanie lampami	12
2.5 Sytuacja na rynku	12
3 Mikrokontroler ESP32-S3	13
3.1 Moduł RTC	13
3.2 Kontroler USB/JTAG	13
3.3 GPIO	13
3.4 Zasilanie	14
4 Serwery czasu	15
4.1 Protokoły synchronizacji czasu	15
4.2 Struktura serwerów w protokole NTP	16
4.3 Zasada działania protokołu NTP	17
5 Koncepcja układu	19
5.1 Założenia projektowe	19
5.2 Ogólny schemat komunikacji z serwerem	20
6 Realizacja	21
6.1 Szczegółowa koncepcja układu	21
6.1.1 Sterowanie lampi nixie	21
6.1.2 Mikrokontroler	21
6.1.3 Źródło dźwięku	21
6.1.4 Pasek LED	22
6.1.5 Interfejs użytkownika	22
6.1.6 Zasilanie	22
6.1.7 Złącza	23
6.1.8 Obudowa	23
6.1.9 Szczegółowy schemat projektu	23
6.2 Test lamp	24
6.3 Obliczenia mocy	26
7 Realizacja modułów	27
7.1 Przetwornica 12V na HV	27
7.1.1 Wybór układu scalonego	27
7.1.2 Założenia projektowe	27

7.1.3	Dobór cewki	27
7.1.4	Dobór kondensatorów	28
7.1.5	Dobór diody	29
7.1.6	Dobór tranzystora	29
7.1.7	Ustawienie napięcia wyjściowego	30
7.1.8	Dobór rezystora ograniczającego prąd	31
7.1.9	Schemat	31
7.2	Przetwornica 12V na 5V	31
7.2.1	Wybór układu scalonego	32
7.2.2	Założenia projektowe	32
7.2.3	Dobór komponentów	32
7.2.4	Schemat	33
7.3	Złącze zasilania	34
7.3.1	Dobór złącza	34
7.3.2	Opis podłączenia	34
7.3.3	Zabezpieczenia ESD	34
7.3.4	Schemat	34
7.4	Złącze do programowania	35
7.4.1	Dobór złącza	35
7.4.2	Opis podłączenia	35
7.4.3	Zabezpieczenia ESD	35
7.4.4	Schemat	35
7.5	Buzzer	36
7.6	Encoder	38
7.7	LDO 5V na 3.3V	39
7.8	Złącze do debugowania	40
8	Oprogramowanie	41
9	Testowanie	41
10	Podsumowanie	41
	Bibliografia	42
	Spis rysunków	43
	Spis tabel	44

Wykaz ważniejszych oznaczeń i skrótów

RTC - Real Time Clock, zegar czasu rzeczywistego

NTP - Network Time Protocol, protokół czasu sieciowego

Wi-Fi - Wireless Fidelity, bezprzewodowa łączność

HV - High Voltage, wysokie napięcie

1 Wstęp i cel pracy

1.1 Wstęp

Głównym celem jaki został postawiony jest zrealizowanie budzika który synchronizuje się z wykorzystaniem zewnętrznego serwera czasu. ctd.

1.2 Cel pracy

Celem pracy jest zaprojektowanie i wykonanie budzika opartego o lampy Nixie, który będzie synchronizował się dzięki odpytywaniu serwera czasu oraz z wykorzystaniem modułu RTC wbudowanego w mikrokontroler. Budzik domyślnie ma działać ze stałym połączeniem do sieci Wi-Fi, ale ma również możliwość działania w trybie offline, z wykorzystaniem modułu RTC wbudowanego w mikrokontroler, chociaż wtedy jego dokładność będzie mniejsza i nie będzie możliwe ustawienie alarmu.

Urządzenie będzie również wykorzystywać serwer Home Assistant jako interfejs użytkownika, przy zachowaniu pewnych funkcji bezpośrednio na urządzeniu. Budzik ten ma na celu bycie bardzo wszechstronnym urządzeniem, które będzie można dostosować do własnych potrzeb, przy ponadczasowym wyglądzie, dzięki lampom Nixie.

2 Lampy nixie

Lampy nixie są to szklane lampy wyświetlające cyfry, litery lub inne symbole. Lampa znajduje się wewnątrz szklanej bańki. Składa się z katod w kształcie cyfr, liter lub innych symboli np. plus czy minus, oraz anoda w kształcie siatki. Wyświetlenia danej cyfry odbywa się poprzez podanie napięcia na katodę odpowiadającą danej cyfrze, oraz napięcie na anodę.

Nixie były używane w latach 50-70 XX wieku w różnych urządzeniach pomiarowych, licznikach, zegarach, itp. Obecnie są one nie praktyczne ze względu na konieczność zasilania wysokim napięciem, skomplikowanym układem sterującym oraz drogą produkcją. Mają one natomiast bardzo duże walory estetyczne, co było powodem ich wyboru.

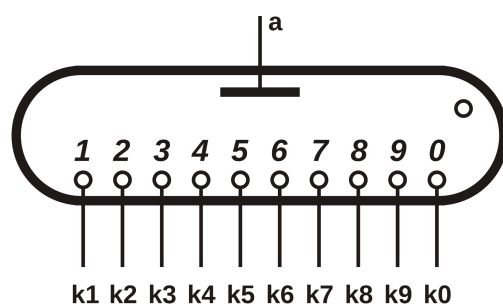
2.1 Zasada działania

Zjawisko zachodzące w lampach zwane jest jako wyładowanie gazowe.[1] Naładowane elektrycznie cząstki (elektrony), poprzez wysokie napięcie osiągają dużą energię kinetyczną. W momencie zderzenia z atomami gazu, elektrony w atomie gazu są wzbudzane do wyższych stanów energetycznych, a następnie wracają do stanu podstawowego emitując foton światła.

Barwa światła zależy od gazu:

- jony neonowe - czerwono-pomarańczowe,
- wodór - niebieskavo-fioletowy,
- azot - fiolet,
- krypton - biało-niebiesko.

Najczęściej stosowana jest mieszanka neonu i argonu pod małym ciśnieniem. Dodawana jest również rtęć, która ma za zadanie zwiększyć trwałość lampy, minimalizując tak zwane zatrucie katodowe. Efekt ten powoduje nie pełne pokrycie katody warstwą gazową, co powoduje zanikanie wyświetlania cyfr.



Rysunek 2.1: Schemat elektryczny lampy nixie ze wspólną anodą[2]

Najczęściej spotykaną konfiguracją jest wspólna anoda, gdzie katody są podłączone do napięcia sterującego, a anoda do napięcia zasilania.

2.2 Problemy związane z wykorzystaniem lamp nixie

Lampy są podatne na wiele problemów, które mogą wystąpić podczas użytkowania, takie jak:

- **Zatrucie katodowe** - zanikanie wyświetlania cyfr, spowodowane nie pełnym pokryciem katody warstwą gazową.
- **Zjawisko kropkowania** - zjawisko polegające na wyświetlaniu kropek w miejscach gdzie nie powinny się one znajdować.
- **Zjawisko spalania** - zjawisko polegające na spalaniu się katod, spowodowane zbyt dużym prądem płynącym przez katodę.
- **Zjawisko cienia** - zjawisko polegające na wyświetlaniu cienia cyfr, spowodowane zbyt dużym napięciem na anodzie.
- **Zjawisko zaniku** - zjawisko polegające na zanikaniu wyświetlania cyfr, spowodowane zbyt małym napięciem na anodzie.
- **Zjawisko migotania** - zjawisko polegające na migotaniu wyświetlania cyfr, spowodowane zbyt małym prądem płynącym przez katodę.

Największym problemem jest zatrucie katodowe w szczególności w kontekście lamp używanych w zegarach nixie. Tylko parzyste wyświetlacze działają w optymalny sposób, ponieważ używają wszystkich cyfr, a więc wszystkie cyfry są używane równomiernie. Problem pojawia się w przypadku nieparzystych wyświetlaczy, gdzie niektóre cyfry są używane znacznie rzadziej niż inne.

Na przykład, lampa po skrajnej lewej stronie używa cyfr 0, 1, 2 do wyświetlania części dziesiątek godziny. Trzecia lampa (pozycja dziesiątek minut) używa cyfr 0, 1, 2, 3, 4, 5. Gdy konkretna cyfra nie jest używana przez długi czas (np. cyfra 8 na najbardziej po lewej stronie lampy), cyfra ta jest pokryta osadem metalu uwolnionym z innych aktywowanych cyfr. Te konkretne cyfry ostatecznie będą miały defekty w świeceniu, jeżeli nie zostaną użyte przez długi czas.

2.3 Rozwiązania problemów związanych z lampami nixie

Rozwiązaniem problemu zatrucia katodowego jest sterowanie lampami w taki sposób, aby wszystkie cyfry były używane. Można to osiągnąć na przykład poprzez animacje na wszystkich cyfrach przy zmianie minuty w zegarze. Można również w godzinach nocnych przez jakiś czas wyświetlać cyklicznie wszystkie cyfry, aby zapobiec zatruciu katodowemu.

W celu zwiększenia trwałości lamp można również tak sterować lampami, aby zmniejszyć napięcie na anodzie, co spowoduje zmniejszenie prądu płynącego przez katody. Mniejszy prąd katodowy ogranicza zjawisko spalania katod, natomiast prąd musi być wystarczająco duży aby lampa świeciła jasno.

2.4 Sterowanie lampami

Sterowanie lampami nixie jest trudne ze względu na konieczność zastosowania wysokiego napięcia, które wynosi od 150 V do 200 V. Istnieje kilka sposobów sterowania lampami nixie[3]:

- Sterowanie bezpośrednie - każda lampka ma swoje wejście i jest sterowana osobno za pomocą tranzystora HV połączonego z mikrokontrolerem. Wadą jest konieczność posiadania wielu pinów GPIO, co jest nieoptymalne. W przypadku 66 pinów GPIO oraz 66 tranzystorów HV. Można by w tym porządku zastosować multiplexery co zmniejszyło by ilość wymaganych pinów GPIO do 16, ale zwiększa to skomplikowanie układu.
- Multiplexing - wszystkie lampy są podłączone do jednego drivera, który wybiera katodę i załączamy odpowiednią anodę. Wymaga to mniej pinów GPIO bo tylko 10, ale multipleksacja powoduje szybsze zużycie lamp i pojawia się efekt migotania(ghosting).
- Wykorzystanie dedykowanych driverów - istnieją specjalne układy scalone, które są przeznaczone do sterowania lampami nixie, niestety one również wymagają wielu pinów GPIO po 4 na każdą lampkę, co daje 24 wymaganych pinów GPIO.
- Rejestr przesuwany HV - najbardziej optymalne rozwiązanie, wymaga tylko 3 pinów GPIO. Rejestry HV są ciężko dostępne i dość drogie, wymagane jest też by były to rejestry z zatraskiem. Wymagane jest również by rejestry miały wyjścia typu Open Drain.
- Połączenie rejestrów przesuwanych z driverami - połączenie rejestrów przesuwanych HV z dedykowanymi driverami, pozwala na zastosowanie rejestru przesuwanego dla niskiego napięcia. Ta kombinacja również wymaga 3 pinów GPIO, przy rejestrze 32 bitowym i 6 driverach. Powoduje jednak to większe skomplikowanie układu oraz układ taki zajmuje więcej miejsca.

Nazwa	Ilość pinów	Cena	Trudność implementacji	Objętość
Sterowanie bezpośrednie	66	niska-średnia	niska-średnia	duża
Multiplexing	10	niska	wysoka	mała
Dedykowane drivery	24	średnia	niska	średnia
Rejestr przesuwany HV	3	wysoka	niska	średnia
Rejestr przesuwany + drivery	3	wysoka	średnia	duża

Tablica 1: Tabela opłacalności sposobów sterowania lampami nixie

2.5 Sytuacja na rynku

Obecnie dostępność części do lamp nixie jest ograniczona, gdyż nie są one już produkowane masowo. Istnieją małe firmy, które zajmują się produkcją, ale są to bardzo duże lampy w małych ilościach, co powoduje ich wysoką cenę. Dostępne są również lampy używane, ale w tym przypadku również duże lampy są drogie, a małe lampy są trudne do znalezienia. Problem jest również nie wiadomy czas ich działania, ponieważ są to używane lampy, które mogą być w różnym stanie.

3 Mikrokontroler ESP32-S3

ESP32-S3 to mikrokontroler firmy Espressif Systems, który jest następcą popularnego ESP32. Jednego z najpopularniejszych mikrokontrolerów z modulem WiFi, wykorzystanego w wielu projektach IoT[4]. Układ ten posiada następujące cechy, odczytane z karty katalogowej[5]:

- 2 rdzenie Xtensa LX7 o taktowaniu 240 MHz
- 2,4 GHz WiFi 4 (802.11 b/g/n)
- Bluetooth 5.0 LE
- dwa 12-bitowe przetworniki ADC do 20 kanałów
- 14 pinów do obsługi dotykowego ekranu
- 45 programowalnych GPIO - część z nich ma specjalne funkcje
- USB/JTAG kontroler
- ROM: 384 KB
- SRAM: 512 KB
- Wbudowany moduł RTC

3.1 Moduł RTC

RTC (Real Time Clock) to moduł czasu rzeczywistego, który pozwala na śledzenie aktualnego czasu, daty oraz dnia tygodnia. ESP32-S3 posiada wbudowany taki moduł, charakteryzuje się on 16kB pamięci SRAM, wynika z tego nie może on przechowywać daty i czasu w przypadku braku zasilania.

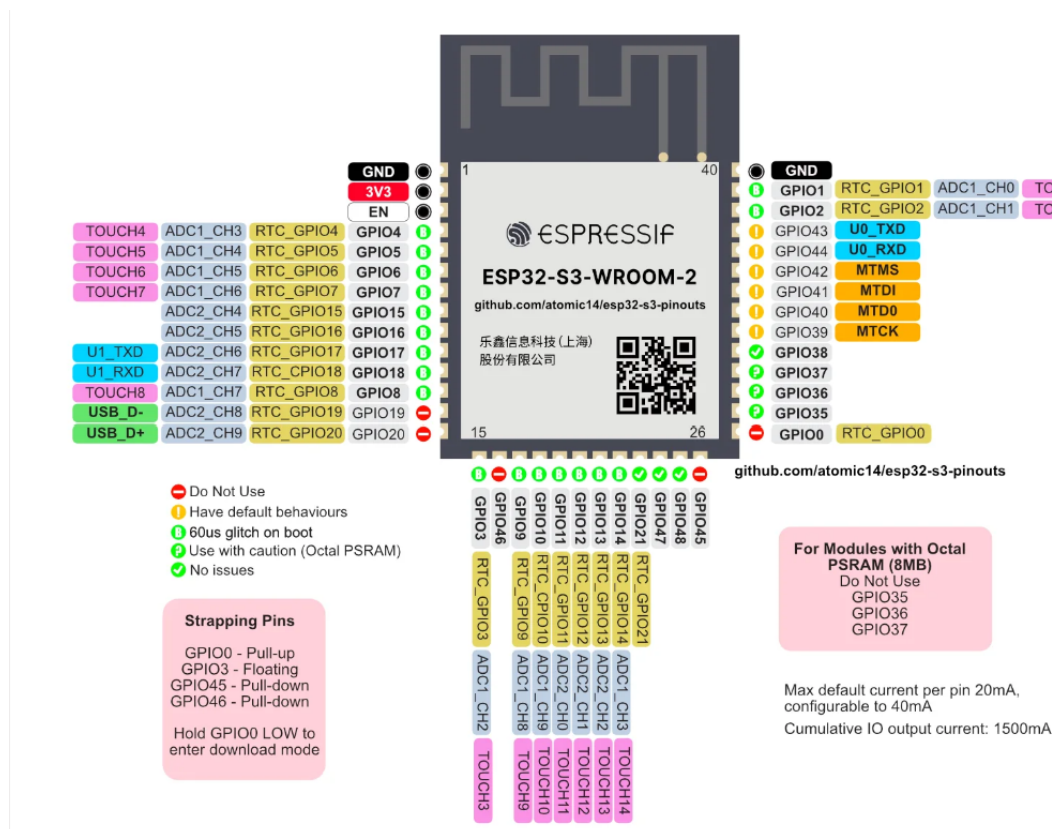
Sam moduł RTC nie jest bardzo dokładny, dlatego zaleca się synchronizację czasu z zewnętrznym serwerem czasu, takim jak NTP. Istnieje dużo bibliotek do realizacji tego zadania z wykorzystaniem Arduino Framework.

3.2 Kontroler USB/JTAG

ESP32-S3 posiada wbudowany kontroler USB/JTAG, który pozwala na programowanie układu oraz debugowanie go, co jest bardzo przydatne podczas tworzenia projektów, gdyż nie wymusza użycia dodatkowego programatora. PIN do programowania układu to GPIO19(D-) oraz GPIO20(D+).

3.3 GPIO

Układ posiada wiele GPIO ogólnego przeznaczenia, nie posiada on dedykowanych pinów do obsługi interfejsów takich jak I2C, można je skonfigurować na dowolnych pinach GPIO. Generacja sygnałów PWM jest również możliwa na dowolnych pinach GPIO. Posiada on również 20 pinów które mogą obsługiwać wejście analogowe.



Rysunek 3.1: Rozkład pinów mikrokontrolera ESP32-S3[6]

3.4 Zasilanie

Według dokumentacji producenta, ESP32-S3 może być zasilany napięciem od 3V do 3.6V, zalecane napięcie zasilania to 3.3V. Jego maksymalny pobór prądu wynosi 340mA, jednak w praktyce jest on znacznie mniejszy, zależy to od wykorzystywanych funkcji.

Układ można wprowadzić w dwa tryby uśpienia:

- Light Sleep - pobór prądu wynosi około 240uA, w tym odłączany jest moduł WiFi a wszystkie piny GPIO są w stanie High-Z.
- Deep Sleep - pobór prądu wynosi około 8uA, jedynie zasilany jest moduł RTC, wszystkie inne funkcje są wyłączone.

4 Serwery czasu

Serwerem czasu nazywany serwer komputerowy, pobierający czas z zewnętrznych źródeł i dystrybuje go do innych urządzeń w sieci[7]. Udostępniają bardzo precyzyjne dane czasowe, dokładność zależy od źródła czasu, z którego serwer korzysta. Serwer czasu może być używany jako lokalny lub internetowy.

Serwery wykorzystują różne źródła zewnętrzne do synchronizacji czasu, takie jak:

- zegary atomowe,
- odbiorniki czasu GNSS (Global Navigation Satellite System),
- oscylatory rubinowe,
- oscylatory cezowe.
- zegary wodorowe

Są to zegary o bardzo dużej precyzji, rzędu nanosekund, co pozwala na synchronizację czasu w sieciach komputerowych, telekomunikacyjnych, itp.

4.1 Protokoły synchronizacji czasu

Serwery te Wykorzystują różne protokoły sieciowe do synchronizacji czasu, takie jak:

- NTP (Network Time Protocol) - Wysyła okresowo pakiety synchronizacji czasu do serwerów w sieci i odpowiednim dostosowywaniu zegarów lokalnych. Jest to najpopularniejszy protokół synchronizacji czasu w sieciach komputerowych, jest on wspierany przez większość systemów operacyjnych.
- PTP (Precision Time Protocol) - jest bardziej precyzyjną alternatywą NTP i jest używany w systemach o wysokiej precyzji. Najczęściej stosowany w sieciach przemysłowych oraz przy badaniach naukowych. Jest w stanie osiągnąć dokładność synchronizacji zegarów do poniżej mikrosekundy.
- Algorytm Berkeley - to algorytm synchronizacji czasu opracowany na Uniwersytecie Kalifornijskim w Berkeley. Jego działanie polega na pomiarze szybkości dryfowania zegara między serwerami, często jest łączony z protokołem NTP.
- GPS - wykorzystuje odbiorniki GPS do synchronizacji zegarów na różnych serwerach. Zapewnia bardzo dokładne sygnały czasu. Czas ten można wykorzystać do synchronizacji zegarów serwerów podłączonych do tego samego odbiornika GPS.

Każdy z tych protokołów ma swoje następujące wady i zalety:

- NTP - Główną zaletą jest niezawodność i dokładność, co sprawia, że nadaje się do szerokiego zakresu zastosowań. Jednak NTP nie jest tak dokładny jak PTP i może synchronizować zegary z dokładnością do kilku milisekund. W związku z tym, że jest to leciwy protokół, nie jest najbardziej bezpiecznym rozwiązaniem, może być podatny na niektóre rodzaje ataków, takie jak ataki typu man-in-the-middle. Protokół istnieje już bardzo długo, więc dobrze znany i jest bardzo łatwy do obsługi.
- PTP - porównując do NTP, PTP jest bardziej precyzyjny i może synchronizować zegary z dokładnością do kilku mikrosekund. Jednak ma zdecydowanie większe wymagania sprzętowe (specjalistyczny sprzęt) i konfiguracyjne, co sprawia, że jest bardziej skomplikowany w użyciu.
- Algorytm Berkeley - można być używać w połączeniu z NTP. Jedną z głównych zalet tego algorytmu jest to, że może synchronizować zegary z dokładnością do kilku mikrosekund, dzięki czemu nadaje się do wielu zastosowań. Podobnie jak w PTP wymaga on specjalistycznego sprzętu, co sprawia, że jest bardziej skomplikowany w użyciu i droższy.
- GPS - najbardziej precyzyjny z wymienionych protokołów, może synchronizować zegary z dokładnością do kilku nanosekund. Jest jednak nie zalecany do zastosowań wewnątrz pomieszczeń, ze względu na konieczność widoczności satelitów GPS i wymaga odbiornika GPS.

Z wyżej wymienionych protokołów, NTP jest najczęściej stosowany w sieciach komputerowych, dlatego też wydaje się być najlepszym wyborem do synchronizacji zegara nixie. Alternatywnym rozwiązaniem może być wykorzystanie własnego serwera który by zwracał czas wykorzystując REST API, ale wymaga to posiadania własnego serwera i jest zależne od jego działania.

4.2 Struktura serwerów w protokole NTP

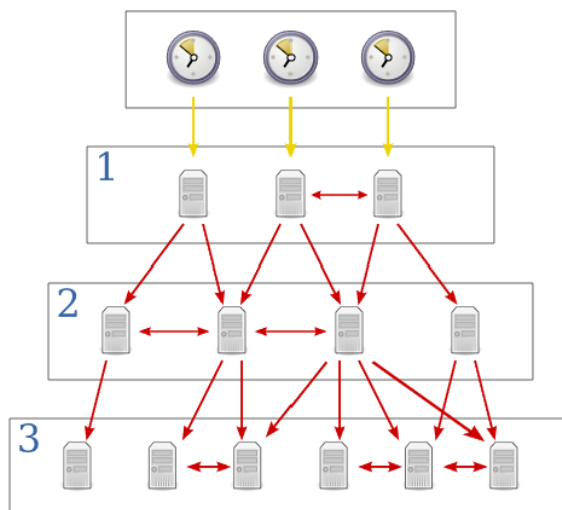
Synchronizacji NTP wykorzystuje uporządkowaną strukturę gałęziową STRATUM[8]. Zasada hierarchii wygląda następująco: urządzenia warstwy STRATUM N mogą być serwerami czasu dla warstwy STRATUM N+1, ale nie na odwrót. Komputery STRATUM N mogą być również klientami urządzeń warstwy STRATUM N-1 itd.

Struktura ta ma na celu uporządkowanie i wprowadzenie hierarchii priorytetów urządzeń, zgodnie z ich rzeczywistym przeznaczeniem i funkcją. Aby nie nadmierne skomplikowania systemu i związanych z tym opóźnień, ilość warstw została ograniczona do 16 (STRATUM 0 - STRATUM 15).

Niektóre warstwy mają specjalne właściwości. Warstwa STRATUM 0 służy wyłącznie dla wzorców czasu, czyli zegarów atomowych, satelitarnych, itp. będących faktycznym źródłem czasu. Połączenie ze źródłem nie jest sieciowe, a zazwyczaj odbywa się za pomocą specjalnych interfejsów sprzętowych.

STRATUM 1 oraz STRATUM 2 stanowią najwyższe warstwy NTP i powinny być wykorzystane w przypadku dużych serwerów wysokiej jakości, superkomputerów lub sprzętowych serwerów czasu. Pozostałe warstwy są przeznaczone dla urządzeń lokalnych, takich jak komputery, routery, itp.

Numer STRATUM mówi jak daleko od wzorca czasu znajduje się dany serwer. Im niższy numer, tym bliżej źródła czasu. W rozbudowanych sieciach poziom STRATUM nie ma znaczącego wpływu na jakość synchronizacji i precyzję uzyskiwanego czasu.



Rysunek 4.1: Struktura serwerów czasu w protokole NTP[9]

W przypadku zegara nixie poziom STRATUM nie ma większego znaczenia, ponieważ zegar nie wymaga bardzo precyzyjnego czasu, chociaż oczywiście zależy, jak precyzyjny czas będzie wyświetlany, ale w przypadku zegara na 6 cyfrach, różnica w czasie rzędu kilku milisekund nie będzie zauważalna.

4.3 Zasada działania protokołu NTP

NTP różni się od typowego protokołu komunikacyjnego. Nie transmituje on bowiem absolutnej wartości czasu, lecz przekazuje informacje o opóźnieniach i korelacjach czasowych w regularnych odstępach czasu, jakie zachodzą w sieci TCP/IP. Protokół wyróżnia się dopiero przy stosowaniu wielu źródeł czasu jednocześnie, wykorzystuje od wtedy algorytm analizy statystycznej czasu oparty na metodzie DTS (Dynamic Time Scales).

NTP wykorzystuje pakiety UDP o długości 72 bajtów na porcie 123, które są okresowo wymieniane co 2^τ sekund, gdzie τ wynosi od 4 (16s) do 17 (36h). Pozwala to klientom serwera, wyliczyć opóźnienie względem idealnego czasu UTC. Znając aktualne opóźnienie w odniesieniu do czasu UTC, klient NTP sam kalibruje swój zegar lokalny, która polega na płynnym przyspieszaniu lub spowalnianiu pracy lokalnego zegara programowego. Przy różnicach czasu przekraczających 128ms, stosowana jest metoda step, która polega na skokowym przesunięciu zegara o określoną wartość. Dzięki temu każdy z klientów, asymptotycznie zmierza do czasu pochodzącego z wzorcowego zegara czasu UTC.

Sam pakiet NTP opisany jest w następujący sposób:

LI	VN	Mode	Stratum	Poll	Precision
Root Delay					
Root Dispersion					
Reference Identifier					
Reference Timestamp					
Originate Timestamp					
Receive Timestamp					
Transmit Timestamp					
Authenticator					

Tablica 2: NTP – format komunikatu

- LI – wskaźnik sekund przestępnych
- VN – (Version Number) numer wersji protokołu
- Mode – tryb pracy
- Stratum – warstwa, w której funkcjonuje komputer będący nadawcą komunikatu
- Poll interval – okres pomiędzy kolejnymi aktualizacjami czasu
- Precision – określenie dokładności zegara komputera wysyłającego dany komunikat
- Root Delay – opóźnienie pomiędzy nadawcą a serwerem warstwy 1
- Root Dispersion – maksymalny błąd pomiędzy zegarem lokalnym a serwera warstwy 1
- Reference Identifier – identyfikator źródła czasu, względem którego następuje synchronizacja
- Reference Timestamp – pole zawierające pomocnicze informacje o czasie poprzedniej synchronizacji
- Originate Timestamp – pole zawierające czas wysłania żądania przez klienta
- Receive Timestamp – czas odebrania komunikatu od klienta
- Transmit Timestamp – czas wysłania odpowiedzi do klienta
- Authenticator – informacje uwierzytelniające zarówno klienta, jak i serwer czasu
- Root Dispersion – maksymalny błąd pomiędzy zegarem lokalnym a serwera warstwy 1
- Reference Identifier – identyfikator źródła czasu, względem którego następuje synchronizacja
- Reference Timestamp – pole zawierające pomocnicze informacje o czasie poprzedniej synchronizacji
- Originate Timestamp – pole zawierające czas wysłania żądania przez klienta
- Receive Timestamp – czas odebrania komunikatu od klienta
- Transmit Timestamp – czas wysłania odpowiedzi do klienta
- Authenticator – informacje uwierzytelniające zarówno klienta, jak i serwer czasu

5 Koncepcja układu

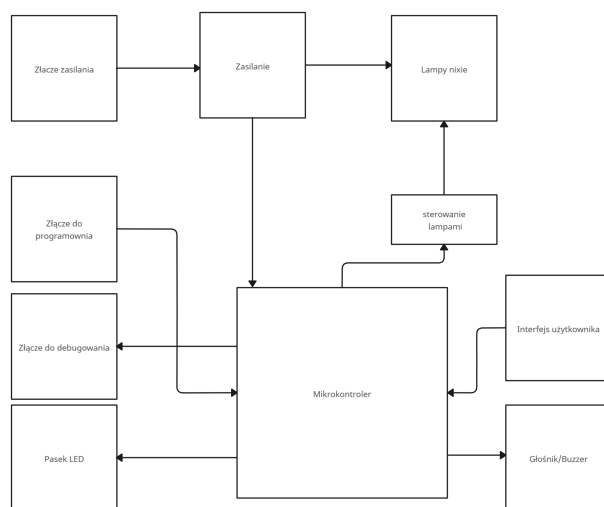
Etap koncepcyjny został podzielony na dwa etapy: koncepcję układu oraz realizację. W tym rozdziale znajduje się ogólna koncepcja działania układu, która nie jest związana z konkretnymi elementami sprzętowymi, a jedynie z funkcjonalnościami, które mają być zrealizowane.

5.1 Założenia projektowe

Zgodnie z celem pracy, określono następujące założenia projektowe:

- Funkcjonalność ustawiania godziny budzika będzie realizowana przez zewnętrzny serwer dla wygody użytkownika, który nie musi dostosowywać czasu ręcznie.
- Na wyświetlaczu Nixie będą wyświetlane godziny, minuty, sekundy.
- Od spodu obudowy będą umieszczone paski LED, które będą podświetlały obudowę i będą wyświetlane animacje podczas alarmu.
- Alarm będzie sygnalizowany dźwiękiem oraz miganiem pasków LED.
- Dźwięk może być odtwarzany z głośnika wbudowanego w obudowę lub z zewnętrznego głośnika komunikującego się z serwerem przez Wi-Fi.
- Wyłączanie alarmu będzie możliwe poprzez przycisk na obudowie, aplikację mobilną lub zewnętrzny przycisk połączony z serwerem.
- W przypadku braku połączenia z serwerem, czas będzie mierzony przez RTC wbudowany w mikrokontroler.
- Programowa oraz manualna regulacja jasności wyświetlacza Nixie oraz pasków LED.

Powyższe założenia powodują podzielenie projektu na poszczególne moduły realizujące poszczególne funkcje, które będą opisane w dalszej części dokumentu. Ogólna koncepcja układu jest przedstawiona na rysunku 5.1.

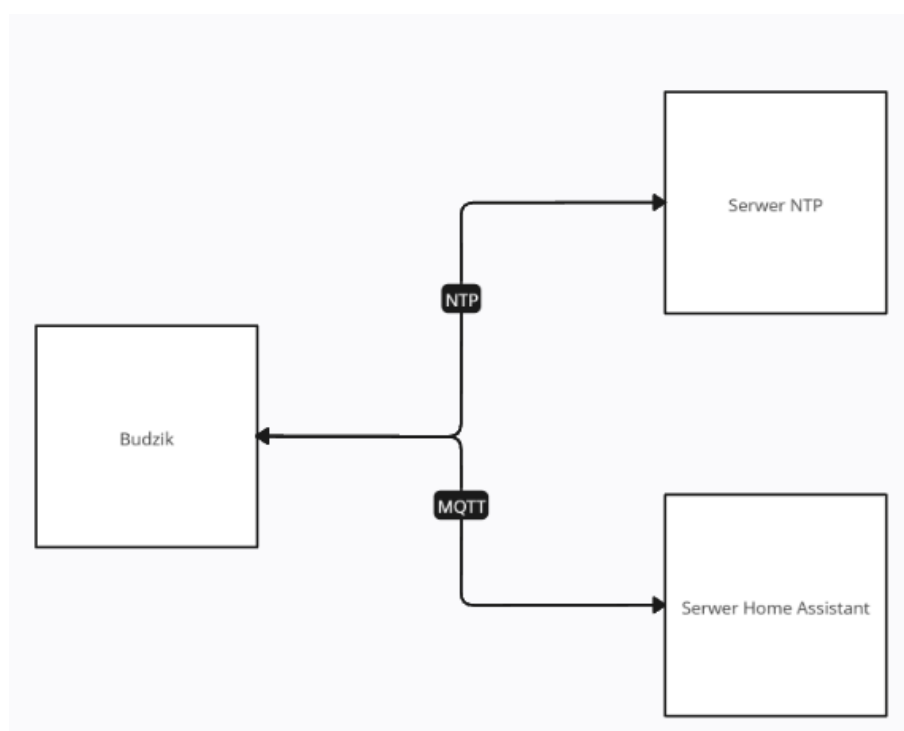


Rysunek 5.1: Ogólna koncepcja układu

Sekcja opisana jako *zasilanie* będzie odpowiedzialna za zasilanie wszystkich elementów układu, w tym lamp Nixie, pasków LED, mikrokontrolera oraz głośnika, więc będzie wymagane rozbić jej na kilka podsekcji, ponieważ będą potrzebne różne napięcia. Lampy Nixie potrzebują zasilania wysokim napięciem, natomiast pozostałe elementy potrzebują zdecydowanie niższych napięć. Blok *sterowanie lampami Nixie* będzie odpowiedzialny za wyświetlanie odpowiednich cyfr na lampach. Sekcja *Interfejs użytkownika* będzie odpowiedzialna za interakcję z użytkownikiem, w tym regulacja jasności oraz wyłączania alarmu, ważne by interfejs był intuicyjny i jak najbardziej rozwojowy na potencjalne przyszłe funkcje.

5.2 Ogólny schemat komunikacji z serwerem

Urządzenie będzie musiało komunikować się z serwerem czasu, który będzie dostarczał aktualny czas, a także z serwerem Home Assistant, który będzie interfejsem użytkownika. Komunikacja z serwerem czasu będzie odbywała się poprzez protokół NTP, ponieważ jest to najbardziej optymalne rozwiązanie, ponieważ inne protokoły opisane w rozdziale 4 służą do zapewnienia większej dokładności czasu, co nie jest wymagane w tym projekcie. Inne protokoły wymagają też większej ilości zasobów lub specjalistycznego sprzętu. Kolejną zaletą wyboru NTP jest to, że jest to najbardziej popularny protokół do synchronizacji czasu w sieciach komputerowych, co sprawia, że jest on najbardziej przetestowany i stabilny. Posiada on wiele implementacji, które są dostępne na wielu platformach, w tym na platformę ESP32. Komunikacja z serwerem Home Assistant będzie odbywała się poprzez protokół MQTT, który jest bardzo popularnym protokołem w IoT, co pozwoli na łatwe rozbudowanie funkcjonalności. Połączenia te przedstawione są na rysunku 5.2.



Rysunek 5.2: Ogólny schemat komunikacji z serwerem

6 Realizacja

W tym rozdziale, opisano szczegółową koncepcję układu, która została opracowana na podstawie analizy przeprowadzonej w rozdziałach 2 i 3 oraz po zapoznaniu się z ofertą sklepów elektronicznych. Zostały określone konkretne rozwiązania projektowe, które będą wykorzystane oraz wykonano niezbędne obliczenia, które pozwoliły na wybór odpowiednich komponentów na dalszym etapie projektowania.

6.1 Szczegółowa koncepcja układu

Szczegółowa koncepcja układu wymaga rozbicia na poszczególne sekcje układu, ponieważ wybory jednego elementu wpływają na wybór kolejnych elementów.

6.1.1 Sterowanie lampi nixie

Kluczowym jest wybór sterownia lampami nixie, ponieważ na podstawie tego wyboru zostanie zaprojektowany reszta układu. Zgodnie z analizą przeprowadzoną w podrozdziale 2.4, zdecydowano się na sterowanie lampami za pomocą rejestrów przesuwnych HV. Zastosowanie tego rozwiązania pozwala na zredukowanie ilości potrzebnych pinów mikrokontrolera do sterowania lampami oraz jest to rozwiązanie proste w implementacji.

Niezależnie od wyboru lamp każda ma 10 katod z cyframi i jedną katodę od kropki dziesiętnej, więc potrzebujemy 11 wyjść na każdą lampę. Dostępne w sprzedaży są rejestry 32 bitowe, co pozwala na sterowanie 3 lampami nixie bez kropek i jedną neonówką która będzie służyć jako separator między godzinami a minutami oraz między minutami a sekundami. Do sterownia kropkami dziesiętnymi zostaną użyte tranzystory HV podpięte do wyjść mikrokontrolera, ponieważ nie opłacalnym jest dodawanie kolejnego rejestru przesuwne HV tylko do sterowania kropkami dziesiętnymi.

Wynika z tego, że potrzebne są 2 rejestry przesuwne HV do sterownia lampi i neonówkami oraz 6 tranzystorów HV do sterowania kropkami dziesiętnymi. Do sterowania rejestrami prawdopodobnie będzie potrzebny konwerter poziomów logicznych, ponieważ mikrokontroler ESP32-S3 pracuje na 3.3V, a rejestry prawdopodobnie będą operować na wyższym napięciu.

6.1.2 Mikrokontroler

Wybór sposobu sterowania lampami nixie wpłynął na wybór mikrokontrolera, ponieważ musi on posiadać odpowiednią ilość pinów GPIO oraz musi być w stanie generować sygnał zegarowy. Potrzebne jest 9 pinów GPIO do pełnego sterownia lampami oraz kropkami dziesiętnym, do tego trzeba pamiętać o zapasie pinów na pozostałe funkcje. W związku z tym wybrano mikrokontroler ESP32-S3, który posiada 45 programowalnych GPIO, co pozwala na swobodne zaprojektowanie reszty układu. Ma też on dużą zaletę w postaci kontrolera USB/JTAG, dzięki czemu nie potrzebujemy dodatkowego programatora do programowania układu. Jest to też popularny mikrokontroler dla którego istnieje dużo bibliotek i przykładów.

6.1.3 Źródło dźwięku

Jako źródło dźwięku wybrano głośnik piezoelektryczny, który jest prosty w implementacji i nie wymaga dodatkowego wzmacniacza, do tego jest mały i tani. Wystarczy jedynie podłączyć go do pinu GPIO mikrokontrolera i za pomocą PWM można generować proste melodie. Głośnik

piezoelektryczny jest wystarczająco głośny aby być słyszalnym w pomieszczeniu, w którym będzie znajdował się budzik.

6.1.4 Pasek LED

Pasek LED będzie służył jako dodatkowe źródło światła, które będzie sygnalizować alarm i jako element estetyczny. Pasek ten musi zawierać w sobie adresowane diody LED, które pozwolą na wyświetlanie różnych kolorów(RGB). Rozwiązanie to jest proste w implementacji, wystarczy podłączyć go do pinu GPIO mikrokontrolera i za pomocą PWM można sterować jasnością.

6.1.5 Interfejs użytkownika

Interfejs użytkownika będzie składał się z enkodera z przyciskiem, który będzie służył do regulacji jasności, a przycisk do wyłączania alarmu. Enkoder jest też na tyle uniwersalny, że można na nim dowolne funkcje ustawić manualnych, ale wygodniejsze jest korzystanie z aplikacji mobilnej.

6.1.6 Zasilanie

Kluczowe jest zaprojektowanie przetwornicy wysokiego napięcia do zasilania lamp nixie, ponieważ jest to najbardziej wymagający element układu.

Możliwe są dwa rozwiązania:

- Przetwornica typu flyback
- Przetwornica typu boost

Przetwornica typu flyback ma zaletę w postaci izolacji galwanicznej między wejściem a wyjściem oraz jest możliwe zaprojektowanie przetwornicy z napięciem zasilania 5V co by pozwoliło na użycie zasilacza USB. Wadą jest to, że jest potrzebny transformator który jest drogi i trudno dostępny, do tego jest to bardziej skomplikowane rozwiązanie na etapie projektowania.

Ze względu na duży problem ze znalezieniem transformatora, zdecydowano się na przetwornicę typu boost, która jest prostsza w implementacji i tańsza. Natomiast wymagało to zastosowania zasilania 12V, co uniemożliwia użycie tylko złącza USB do zasilania układu, natomiast znacząco upraszcza projektowanie układu.

Wybrano więc przetwornicę typu boost, która będzie zasilana z zasilacza 12V, a wyjście będzie podłączone do anod lamp nixie.

Do tego będzie potrzebne zasilanie 5V dla paska LED oraz 3.3V dla mikrokontrolera. Zasilanie 5V w związku z tym, że będzie zasilał pasek LED, który potrafi pobrać większy prąd, to ze względu na zachowanie wysokiej efektywności, zdecydowano się na przetwornicę typu buck. Zasilanie 3.3V będzie zasilaniem mikrokontrolera, więc wystarczy zastosować stabilizator liniowy.

Można, więc podzielić zasilanie na 3 pod moduły:

- Przetwornica typu boost z zasilacza 12V na HV
- Przetwornica typu buck z zasilacza 12V na 5V
- Stabilizator liniowy z zasilacza 5V na 3.3V

6.1.7 Złącza

W związku z tym, że będzie potrzebne zasilanie z zasilacza 12V, zdecydowano się na zastosowanie złącza DC jack, które jest powszechnie stosowane w zasilaczach. Do zasilania mikrokontrolera oraz programowania, zdecydowano się na złącze USB-C, które jest oceniane najbardziej uniwersalnym rozwiązaniem. Zostanie również dodane złącze Goldpin, które będzie służyło jako złącze debugowe, co pozwoli na łatwe debugowanie układu, na etapie tworzenia oprogramowania.

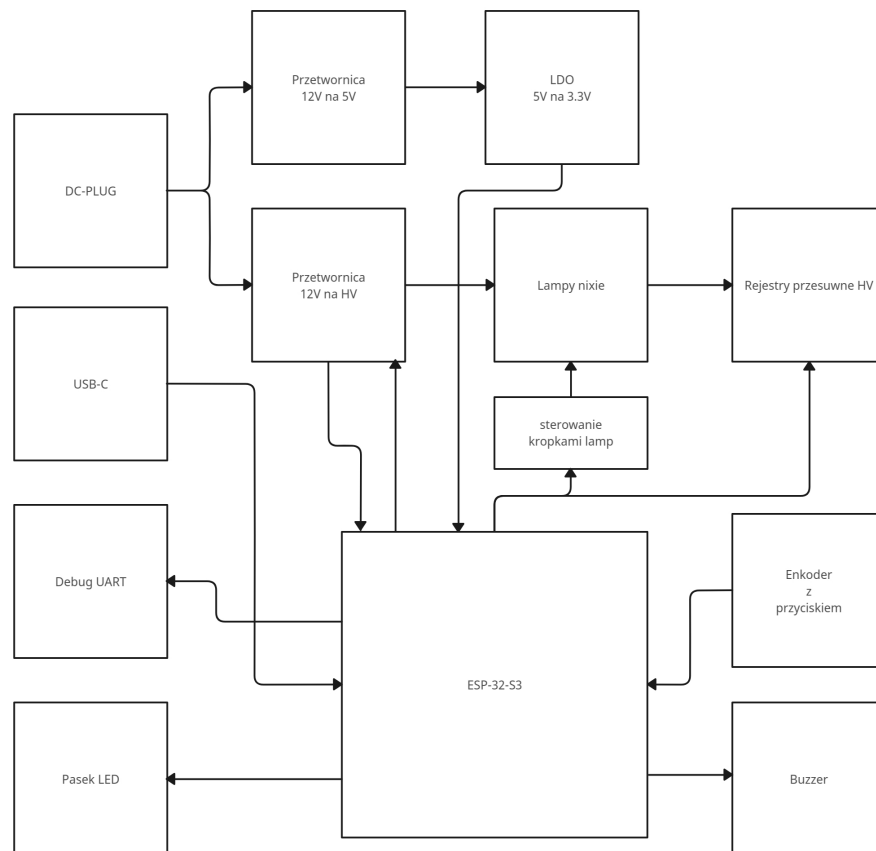
Urządzenie będzie posiadać 3 złącza:

- Złącze USB-C do programowania mikrokontrolera
- Złącze DC do zasilania układu
- Złącze Goldpin jako złącze debugowe

6.1.8 Obudowa

By zachować wygląd retro, zdecydowano się na zastosowanie obudowy drewnianej. Od góry będzie znajdowało się szkło akrylowe, które będzie służyło jako osłona przed kurzem i jednocześnie będą widoczne elementy elektryczne.

6.1.9 Szczegółowy schemat projektu



Rysunek 6.1: Schemat blokowy projektu

6.2 Test lamp

Pierwszą rzeczą, która została wykonana to zakup lamp Nixie, które będą wykorzystane w projekcie. Przez coraz mniejszą dostępność lamp Nixie, zdecydowano się na zakup małych lamp Z570M, które zostały zakupione w ilości 6 sztuk. Lampy te mają 10 cyfr oraz kropkę dziesiętną. Lampy są używane, ale wszystkie lampy zostały sprawdzone i działają poprawnie. Kluczowe parametry zastosowanych lamp nixie odczytane z karty katalogowej[10] to:

- Napięcie zapłonu: 170 V
- Napięcie wygaszania: 120 V
- Napięcie pracy: 150 V
- Prąd katodowy średni: 2 mA

W celu zweryfikowania działania lamp nixie i sprawdzenia parametrów zasilania, został wykonany prototyp układu z jedną lampą nixie, wykorzystujący zasilacz impulsowy HV z regulowanym napięciem wyjściowym zakupiony w sklepie internetowym.

Zakupiona przetwornica HV ma następujące parametry:

- Napięcie wejściowe: 5 – 12 V
- Napięcie wyjściowe: 150 – 220 V
- Prąd wyjściowy: 20 mA

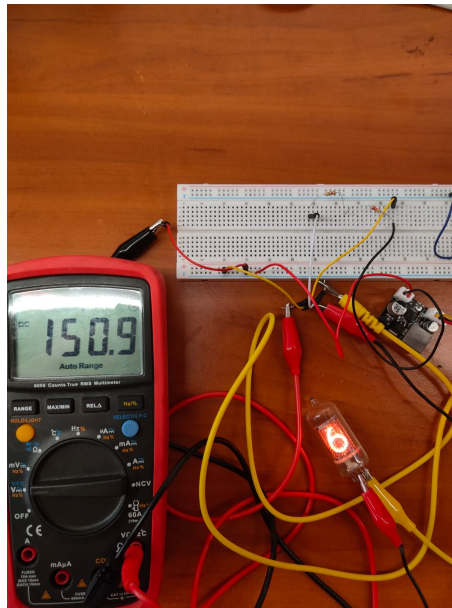
W celu sprawdzenia działania lampy musimy najpierw dobrać rezystor ograniczający prąd katodowy. Zakładając, że napięcie zasilania wynosi maksymalnie $U_{\max} = 220 \text{ V}$, a napięcie zapłonu lampy $U_{\text{zap}} = 170 \text{ V}$, przy prądzie katodowym $I_{\text{kat}} = 2 \text{ mA}$, rezystor ograniczający prąd katodowy można obliczyć ze wzoru:

$$R = \frac{U_{\max} - U_{\text{zap}}}{I_{\text{kat}}} = \frac{220 \text{ V} - 150 \text{ V}}{2 \text{ mA}} = 35 \text{ k}\Omega \quad (1)$$

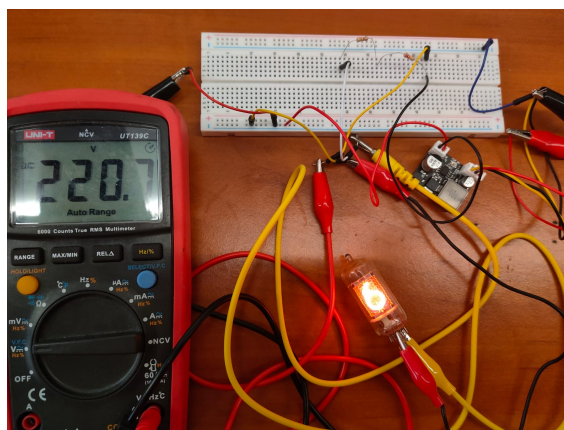
W nocie katalogowej lampy nixie Z570M producent podaje, że zalecany rezystor ograniczający prąd katodowy powinien mieć wartość $33 \text{ k}\Omega$ dla napięcie zasilania 200 V, więc wartość $35 \text{ k}\Omega$ dla napięcia 220 V wydaje się obliczona prawidłowo, taki też rezystor ma zostać użyty w faktycznym układzie.

Zatem rezystor ograniczający prąd katodowy powinien mieć wartość około $35 \text{ k}\Omega$. Do testu użyto rezystora o wartości $22 \text{ k}\Omega$ oraz rezystora o wartości $10 \text{ k}\Omega$, połączonych szeregowo, co daje wartość $32 \text{ k}\Omega$, co jest wartością zbliżoną do obliczonej.

Z testów wynika, że lampy nixie działają poprawnie, a dobrany rezystor ograniczający prąd katodowy jest odpowiedni. Przy napięciu zasilania 150 V lampa świeci słabiej, ale jest to zgodne z oczekiwaniami, natomiast przy napięciu 220 V lampa świeci jasno i pojawiają się lekko niebieskie refleksy wewnątrz lampy, co jest zgodne z oczekiwaniami.



Rysunek 6.2: Prototyp układu z lampą nixie przy napięciu zasilania 150 V



Rysunek 6.3: Prototyp układu z lampą nixie przy napięciu zasilania 220 V

Nie sprawdzono napięcia wygaszania, ponieważ zasilacz nie pozwalał na takie napięcie, ale ustalono, że lampa nawet przy napięciu 150 V była w stanie zapłonąć i świecić poprawnie.

Z testów można wyciągnąć następujące wnioski:

- Lampy nixie działają poprawnie przy napięciu zasilania 150 V oraz 220 V.
- Dobrany rezystor ograniczający prąd katodowy jest odpowiedni.
- Lampa nixie Z570M jest w stanie zapłonąć i świecić przy napięciu wygaszania 150 V.
- Zakres regulacji napięcia na zasilaczu HV powinien być większy np. 130 – 250 V, by lampa mogła być jeszcze słabiej podświetlona, może się to okazać przydatne w nocy.

6.3 Obliczenia mocy

By móc zaprojektować odpowiednie zasilanie dla całego układu, należy obliczyć szacunkową moc potrzebną do zasilania wszystkich komponentów. Poza lampami nixie, najbardziej obciążającym elementem będzie pasek LED oraz mikrokontroler, pozostałe elementy będą pobierały znikome ilości prądu.

Założono maksymalną długość paska LED na 30 cm. Z deklaracji producenta paska LED wynika, że moc na metr wynosi 18 W, co daje:

$$P_{\text{LED}} = 18 \text{ W m}^{-1} \cdot 0.3 \text{ m} = 5.4 \text{ W} \quad (2)$$

Następnie obliczono prąd potrzebny do zasilania paska LED przy napięciu 5 V:

$$I_{\text{LED}} = \frac{P_{\text{LED}}}{U_{\text{LED}}} = \frac{5.4 \text{ W}}{5 \text{ V}} = 1.08 \text{ A} \quad (3)$$

Następnie obliczono moc potrzebną do zasilania mikrokontrolera ESP32-S3, według producenta maksymalny pobór prądu wynosi 340 mA, co przy napięciu zasilania 3.3 V daje:

$$P_{\text{ESP32}} = 340 \text{ mA} \cdot 3.3 \text{ V} = 1.122 \text{ W} \quad (4)$$

Następnie policzono prąd pobierany przez wszystkie lampy, których jest 6 sztuk, przy prądzie katodowym 2 mA każda, co daje:

$$I_{\text{Nixie}} = 6 \cdot 2 \text{ mA} = 12 \text{ mA} \quad (5)$$

Następnie obliczono moc potrzebną do zasilania lampy nixie, przy napięciu 220 V oraz prądzie wszystkich lamp 12 mA, zakładając sprawność przetwornicy na poziomie 70 %:

$$P_{\text{Nixie}} = \frac{U_{\text{Nixie}} \cdot I_{\text{Nixie}}}{\text{Sprawność}} = \frac{220 \text{ V} \cdot 12 \text{ mA}}{0.7} = 3.43 \text{ W} \quad (6)$$

Pozostałe komponenty będą pobierały znikome ilości prądu, więc nie będą brane pod uwagę w obliczeniach. Szacunkowa moc potrzebna do zasilania całego układu wynosi:

$$P_{\text{całkowita}} = P_{\text{LED}} + P_{\text{ESP32}} + P_{\text{Nixie}} = 5.4 \text{ W} + 1.122 \text{ W} + 3.43 \text{ W} = 9.952 \text{ W} \quad (7)$$

Szacunkowa moc potrzebna do zasilania całego układu wynosi około 10 W,

7 Realizacja modułów

Po określeniu szczegółowej koncepcji układu, przystąpiono do realizacji poszczególnych modułów, które zostały opisane w rozdziale 6. Każdy z modułów ma opisany dobór komponentów do realizacji funkcjonalności modułu oraz sposób podłączenia.

7.1 Przetwornica 12V na HV

7.1.1 Wybór układu scalonego

Wybrano układ LM3488 produkcji Texas Instruments, który jest układem przeznaczonym do budowy przetwornic typu Boost oraz Flyback. Jest to układ high efficiency, co jest powodem dla którego został wybrany.

7.1.2 Założenia projektowe

- Napięcie wejściowe: 12V
- Napięcie wyjściowe: 130-220V
- Prąd wyjściowy: 20mA
- 0.1V tętnienia napięcia wyjściowego
- Częstotliwość przełączania: 400kHz

Jako początkowe założenie przyjęto częstotliwość przełączania 400kHz zgodnie z domyślną wartością w nocie katalogowej układu, ale możliwe jest zwiększanie częstotliwości do 1MHz, co pozwala na zmniejszenie rozmiarów cewki oraz kondensatorów, natomiast może to pogorszyć sprawność układu, ponieważ według noty katalogowej wraz ze wzrostem częstotliwości spada wzmocnienie układu, co przekłada się na mniejszą sprawność.

7.1.3 Dobór cewki

Oszacowano wartość prądu cewki na podstawie założonego prądu wyjściowego:

$$I_l = \frac{V_{out} \cdot I_{out}}{V_{in}} = \frac{220 \cdot 0.02}{12} \approx 0.36A \quad (8)$$

Dodatkowe 30% prądu wyjściowego zostało dodane jako tętnienia prądu, co pozwala oszacować wartość maksymalnego prądu cewki:

$$I_{peak} = (1 + 0.3) \cdot I_l = 1.3 \cdot 0.36 \approx 0.468A \quad (9)$$

Wynika z tego, że potrzebna jest cewka o prądzie przewodzenia większym niż 0.5A.

Obliczono wypełnienie PWM, na podstawie wzoru:

$$D_{220} = \frac{V_{out} - V_{in}}{V_{out}} = \frac{220 - 12}{220} \approx 0.945 \quad (10)$$

$$D_{130} = \frac{V_{out} - V_{in}}{V_{out}} = \frac{130 - 12}{130} \approx 0.908 \quad (11)$$

Zgodnie z notą katalogową układu LM3488, cewka powinna mieć wartość określaną wzorem:

$$L > \frac{D(1 - D)V_{in}}{2f_{sw}I_{out}} \quad (12)$$

Według dokumentacji I_{out} podczas obliczeń powinno stanowić 30% minimalnej wartości prądu wyjściowego:

$$I_{out} = 0.3 \cdot 20\text{mA} = 6\text{mA} \quad (13)$$

Dla napięcia wyjściowego 220V oraz napięcia wejściowego 12V oraz częstotliwości 400kHz otrzymano wartość cewki:

$$L_{220} > \frac{0.945 \cdot (1 - 0.945) \cdot 12}{2 \cdot 400000 \cdot 0.006} \approx 128.9\mu\text{H} \quad (14)$$

Natomiast dla napięcia wyjściowego 130V oraz napięcia wejściowego 12V oraz częstotliwości 400kHz otrzymano wartość cewki:

$$L_{130} > \frac{0.908 \cdot (1 - 0.908) \cdot 12}{2 \cdot 400000 \cdot 0.006} \approx 209.5\mu\text{H} \quad (15)$$

Napotkano problem z doбором cewki, ponieważ nie udało się znaleźć w sklepie cewki o wartości powyżej 200 μH w rozsądnej cenie i odpowiednich rozmiarach. Dlatego zdecydowano się na zastosowanie cewki o wartości 180 μH , która jest najbliższą wartością dostępną w sklepie. Wartość graniczna prądu cewki wynosi 0.9A, co jest wystarczającym zapasem prądowym.

W celu osiągnięcia założonego zakresu napięcia wyjściowego z użyciem wybranej cewki, zdecydowano się na zwiększanie częstotliwości przełączania do 500kHz. Obliczono wartość cewki dla napięcia wyjściowego 130V oraz napięcia wejściowego 12V oraz częstotliwości 500kHz:

$$L_{220} > \frac{0.945 \cdot (1 - 0.945) \cdot 12}{2 \cdot 500000 \cdot 0.006} \approx 103.1\mu\text{H} \quad (16)$$

$$L_{130} > \frac{0.908 \cdot (1 - 0.908) \cdot 12}{2 \cdot 500000 \cdot 0.006} \approx 167.6\mu\text{H} \quad (17)$$

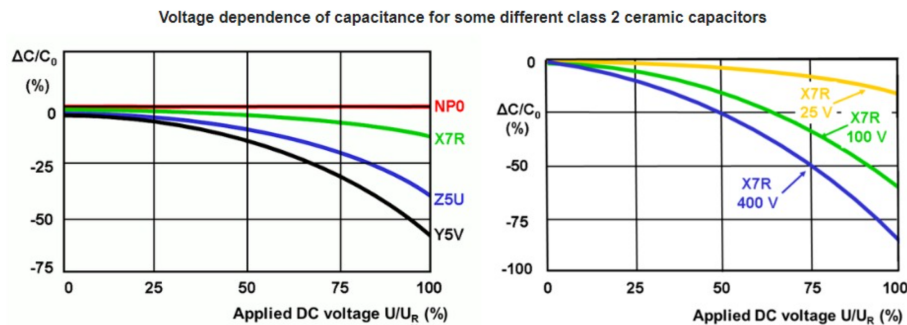
Z obliczeń wynika, że zwiększanie częstotliwości pozwala na zmniejszenie wartości cewki, co pozwala na zastosowanie cewki o wartości 180 μH .

7.1.4 Dobór kondensatorów

Według zaleceń z noty katalogowej w przetwornicy powinny być zastosowane kondensatory o jak najniższym ESR, dlatego odrzucono kondensatory elektrolityczne na rzecz kondensatorów ceramicznych, które mają bardzo niski ESR, tak mały że producent nie podaje tej wartości w notach katalogowych, gdyż jest ona zbyt mała by miała znaczenie.

Jednak problem jest znalezienie kondensatorów ceramicznych dla wysokich napięć, mimo tego został znaleziony kondensator ceramiczny o wartości 2.2 μF i napięciu pracy do 250V.

W obliczeniach należy uwzględnić spadek pojemności kondensatora wraz ze wzrostem napięcia. Wartości spadku pojemności dla kondensatora ceramicznego odczytano z następującego wykresu:



Rysunek 7.1: Spadek pojemności kondensatora ceramicznego wraz ze wzrostem napięcia

Wybrany kondensator jest klasy 2, wykonany z materiału X7R, co oznacza, że jego pojemność spadnie w przybliżeniu o 30% dla napięcia 220V. Zdecydowano się na zastosowanie 2 kondensatorów o wartości 2.2μF połączonych równolegle w celu zminimalizowania tętnień napięcia wyjściowego. Ostatecznie pojemność oszacowana na:

$$C_{220} = 2 \cdot (1 - 0.3) \cdot 2.2\mu\text{F} = 3.08\mu\text{F} \quad (18)$$

Obliczono tętnienia dla dobranych wartości kondensatorów:

$$\Delta V_{220} = \frac{V_{out}}{2 \cdot \frac{V_{out}}{I_{out}} \cdot C} \cdot \frac{D}{f_{sw}} = \frac{220}{2 \cdot \frac{220}{0.02} \cdot 3.08\mu\text{F}} \cdot \frac{0.945}{500000} \approx 6.1\text{mV} \quad (19)$$

$$\Delta V_{130} = \frac{V_{out}}{2 \cdot \frac{V_{out}}{I_{out}} \cdot C} \cdot \frac{D}{f_{sw}} = \frac{130}{2 \cdot \frac{130}{0.02} \cdot 3.08\mu\text{F}} \cdot \frac{0.908}{500000} \approx 5.89\text{mV} \quad (20)$$

Wartości tętnień jest mniejsza niż założone 0.1V, co oznacza, że dobrano odpowiednie wartości kondensatorów.

7.1.5 Dobór diody

Oszacowano prąd diody na podstawie wzoru z noty katalogowej:

$$I_d = \frac{I_{out}}{1 - D} + \Delta I_{out} = \frac{0.02}{1 - 0.945} + 0.006 \approx 0.37\text{A} \quad (21)$$

Dioda powinna mieć prąd przewodzenia większy niż 0.4A oraz być szybką diodą shottky'ego, by zminimalizować straty w układzie.

Zdecydowano się na zastosowanie diody ES1G firmy Onsemi, która jest diodą super szybką, o prądzie przewodzenia 1A, co jest wystarczające dla tego zastosowania. Dioda ta ma maksymalne napięcie wsteczne 400V, co jest wystarczające.

7.1.6 Dobór tranzystora

Można założyć że prąd tranzystora to prąd cewki, czyli 0.468A. Zgodnie z noty katalogowej tranzystor powinien mieć następujące parametry:

- Napięcie minimalnie drain-source: 250V
- Jak najmniejszy $R_{DS(on)}$
- Prąd przewodzenia większy niż 0.5A
- Niskie napięcie progowe V_{TH}
- Jak najmniejszy ładunek bramki
- Wymaganego prąd bramki mniejszego niż 1A

Zdecydowano się na zastosowanie tranzystora N-Channel MOSFET TPH5200FNH firmy Toshiba, który ma następujące parametry:

- Napięcie drain-source: 250V
- $R_{DS(on)}$: 44m Ω
- Prąd przewodzenia: 26A
- Napięcie progowe: 2V
- Ładunek bramki: 22nC
- Rozpraszana moc: 2.5W

Prąd potrzebny na załączenie tranzystora można obliczyć na podstawie wzoru:

$$I_g = Q_g \cdot f_{sw} = 22\text{nC} \cdot 500000 = 11\text{mA} \quad (22)$$

Tranzystor ten spełnia wszystkie założenia, a także ma bardzo niskie $R_{DS(on)}$, co pozwala na zminimalizowanie strat w układzie.

Moc wydzielana na tranzystorze można obliczyć na podstawie wzoru:

$$P_{mos} = I_l^2 \cdot R_{DS(on)} = 0.468^2 \cdot 0.044 \approx 0.01\text{W} \quad (23)$$

Moc jest bardzo niska, co oznacza, że tranzystor nie będzie się nagrzewał, a także nie będzie wymagał radiatora.

7.1.7 Ustawienie napięcia wyjściowego

Napięcie wyjściowe będzie regulowane, dlatego zdecydowano się na zastosowanie potencjometru cyfrowego włączonego w obwód sprzężenia zwrotnego. Potencjometr cyfrowy będzie się komunikował z mikrokontrolerem za pomocą magistrali I2C, co pozwoli na zdalne ustawianie napięcia wyjściowego, by regulować jasność lamp.

Wybrano potencjometr cyfrowy MCP4018T-103E/LT firmy Microchip, który ma 128 poziomów ustawień, co pozwala na dokładne ustawienie napięcia wyjściowego, wybrano wartość 10k Ω , co pozwala na uzyskanie odpowiedniego zakresu ustawień napięcia wyjściowego.

Zgodnie z notą katalogową napięcie na pinie FB powinno wynosić 1.26V, napięcie to oznacza, że napięcie wyjściowe jest odpowiednie. Po przetestowaniu kilku kombinacji zdecydowano się na zastosowanie następujących rezystorów:

- $R_{fb1} = 2.49\text{M}\Omega$
- $R_{fb2} = 14.39\text{k}\Omega$

Obliczone napięcie na wyjściu dla potencjometru z nastawą 10k Ω :

$$V_{out} = 1.26 \cdot \left(1 + \frac{R_{fb1}}{R_{fb2} + R_{pot}}\right) = 1.26 \cdot \left(1 + \frac{2.49\text{M}\Omega}{14.39\text{k}\Omega + 10\text{k}\Omega}\right) \approx 130.4\text{V} \quad (24)$$

Obliczone napięcie na wyjściu dla potencjometru z nastawą 0 Ω :

$$V_{out} = 1.26 \cdot \left(1 + \frac{R_{fb1}}{R_{fb2} + R_{pot}}\right) = 1.26 \cdot \left(1 + \frac{2.49\text{M}\Omega}{14.39\text{k}\Omega}\right) \approx 220.6\text{V} \quad (25)$$

Uzyskano zakres napięcia wyjściowego od 130.4V do 220.6V, co jest zgodne z założeniami projektowymi.

7.1.8 Dobór rezystora ograniczającego prąd

Układ ma możliwość ustawienia limitu prądu jaki będzie płynąć przez tranzystor, co jest dodatkowym zabezpieczeniem przed uszkodzeniem tranzystora.

Najpierw obliczono wartość limitu szczytowego prądu przełączania zgodnie z notą katalogową:

$$ISW_{limit} = \left(\frac{I_{out}}{1-D} + \frac{D \cdot V_{in}}{2 \cdot f_{sw} \cdot L} \right) = \left(\frac{0.02}{1-0.945} + \frac{0.945 \cdot 12}{2 \cdot 500000 \cdot 180\mu H} \right) \approx 0.426A \quad (26)$$

Następnie obliczono wartość rezystora ograniczającego prąd zgodnie z notą katalogową:

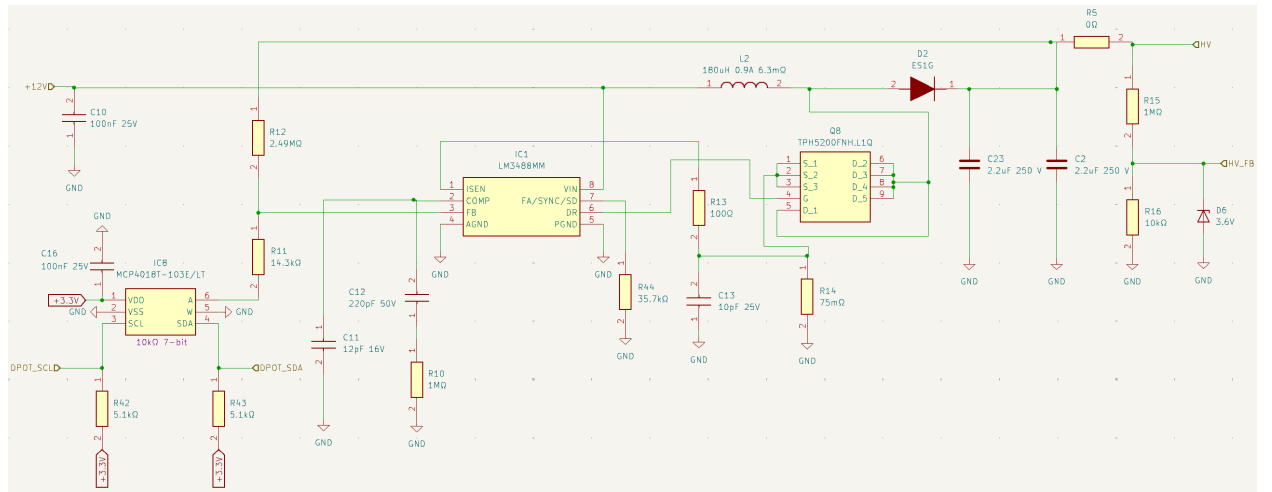
$$R_{sense} = \frac{V_{SENSE} - (D \cdot V_{SENSE} \cdot V_{SL-ratio})}{ISW_{limit}} = \frac{156mV - (0.945 \cdot 156mV \cdot 0.49)}{0.426} \approx 74m\Omega \quad (27)$$

Następnie sprawdzono warunek na maksymalną wartość rezystora ograniczającego prąd:

$$R_{sense} < \frac{2 \cdot V_{SL} \cdot f_{sw} \cdot L}{V_{out} - (2 \cdot V_{IN})} = \frac{2 \cdot 92mV \cdot 500000 \cdot 180\mu H}{220 - (2 \cdot 12)} \approx 84m\Omega \quad (28)$$

Zdecydowano się na zastosowanie rezystora o wartości 75mΩ, który spełnia wszystkie założenia. Został również dodany kondensator o wartości 10pF w celu zminimalizowania tętnień napięcia na rezystorze, oraz rezystor kompensujący 100Ω.

7.1.9 Schemat



Rysunek 7.2: Schemat przetwornicy 12V na 5V

7.2 Przetwornica 12V na 5V

7.2.1 Wybór układu scalonego

Zdecydowano się na układ TPS563219ADDFR produkcji Texas Instruments, który jest przetwornicą impulsową z wbudowanym tranzystorem mocy oraz zapewniającym prąd wyjściowy do 3A przy napięciu wyjściowym do 7V. Układ jest też w obudowie na tyle dużej, by móc go polutować ręcznie.

Układ posiada soft-start oraz wyjście power good (potwierdzające start przetwornicy), co nie jest potrzebne w tym zastosowaniu, tak samo nie jest to najmniejszy układ, ale zapewnia to łatwość montażu co jest ważne w tym przypadku.

7.2.2 Założenia projektowe

- Napięcie wejściowe: 12V
- Napięcie wyjściowe: 5V
- Prąd wyjściowy: 2A
- 50mV tętnienia napięcia wyjściowego

7.2.3 Dobór komponentów

Dobór komponentów wykonano na podstawie sugerowanych wartości z noty katalogowej układu TPS563219ADDFR.

Table 4. TPS563219A Recommended Component Values

Output Voltage (V)	R2 (k Ω)	R3 (k Ω)	L1 (μ H)			C6 + C7 + C8 (μ F)
			MIN	TYP	MAX	
1	3.09	10.0	1.0	1.5	4.7	20 - 68
1.05	3.74	10.0	1.0	1.5	4.7	20 - 68
1.2	5.76	10.0	1.0	1.5	4.7	20 - 68
1.5	9.53	10.0	1.0	1.5	4.7	20 - 68
1.8	13.7	10.0	1.5	2.2	4.7	20 - 68
2.5	22.6	10.0	1.5	2.2	4.7	20 - 68
3.3	33.2	10.0	1.5	2.2	4.7	20 - 68
5	54.9	10.0	2.2	3.3	4.7	20 - 68
6.5	75	10.0	2.2	3.3	4.7	20 - 68

Rysunek 7.3: Tabela doboru komponentów z noty katalogowej

Na podstawie tabeli dobrano następujące wartości komponentów:

- C1: 22 μ F 10V
- C29: 22 μ F 10V
- L1: 2.2 μ H 9.2A 14.5m Ω
- R1: 56k Ω
- R2: 10k Ω

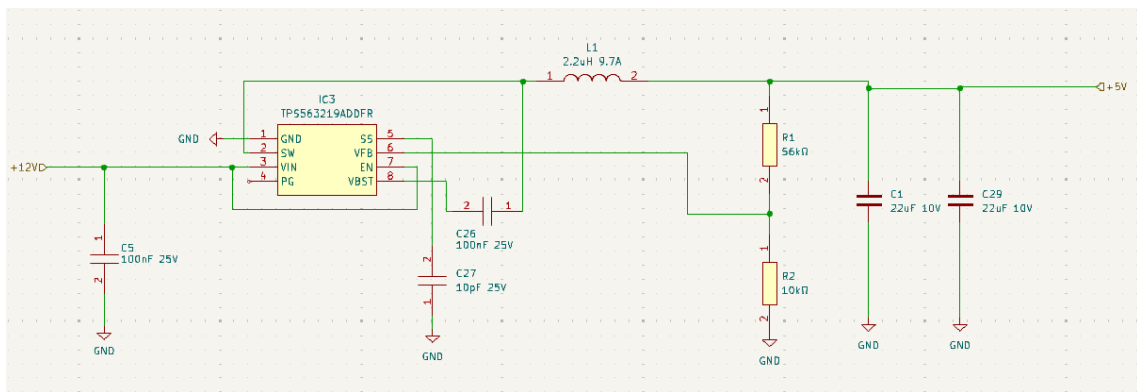
Kondensator podpięty pod pin SS(soft start) oraz kondensator podpięty pod pin VBST, został skopiowany z układu z noty katalogowej, gdyż nie jest to krytyczny element i nie ma potrzeby doboru wartości pod kątem zastosowania w zegarze.

Zdecydowano się na użycie kondensatorów ceramicznych, gdyż są one mniejsze i mają lepszy ESR niż elektrolityczne, co ma znaczenie przy przetwornicach impulsowych, gdzie mamy wyższe częstotliwości przełączania, więc ESR kondensatora ma większe znaczenie, przy zbyt dużym ESR kondensatora, może on się nagrzewać, co prowadzi do jego uszkodzenia. Kondensatory ceramiczne natomiast cechują się tak małym ESR, że producent nie podaje tej wartości w notach katalogowych, gdyż jest ona zbyt mała by miała znaczenie.

Cewka dobrano biorąc pod uwagę jest stosunek oporu do ceny, zdecydowano się na cewkę o oporze $14.5\text{m}\Omega$, gdyż jest to najniższa wartość jako udało się znaleźć w sklepach elektronicznych w sensownej cenie i dość małej obudowie.

Dodano również kondensator filtrujący 100nF na pinie VCC, by zredukować szumy z linii zasilania.

7.2.4 Schemat



Rysunek 7.4: Schemat złącza DC-Plug

7.3 Złącze zasilania

7.3.1 Dobór złącza

W doborze złącza kluczowe była jego wielkość oraz prąd, który jest w stanie przewodzić. Teoretyczna moc układu to 10W, co przy napięciu 12V daje prąd 0.83A. Złącze musi być w stanie przewodzić prąd 1A, by zapewnić bezpieczeństwo.

Wybrano złącze firmy same sky o symbolu PJ-094H-SMT-TR o styku 0.65x2.35mm, które jest w stanie przewodzić prąd 2.5A, czyli więcej niż wystarczająco. Złącze jest bardzo niskie, jego wysokość to 3.5mm, co pozwala na zminimalizowanie wysokości zegara.

7.3.2 Opis podłączenia

Jako kondensatory filtrujące wykorzystano, 2 kondensatory o pojemności 100uF w celu zminimalizowania zakłóceń niskich częstotliwości, oraz 1 kondensator o pojemności 100nF w celu zminimalizowania zakłóceń wysokich częstotliwości.

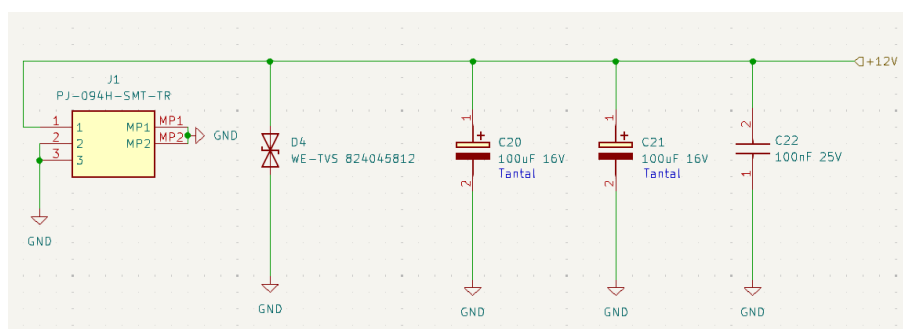
Kondensatory 100uF są kondensatorami tantalowymi, a kondensator 100nF jest kondensatorem ceramicznym. Wykorzystano te rodzaje kondensatorów, ponieważ są to kondensatory o długiej żywotności, a także są to kondensatory o małych rozmiarach w przeciwieństwie do kondensatorów elektrolitycznych.

7.3.3 Zabezpieczenia ESD

W celu zabezpieczenia linii przed przepięciami, wykorzystano diodę TVS firmy Wurth Elektronik o symbolu 824045812. Dioda ta jest diodą TVS o napięciu przebicia 13.3V, oraz napięciu stabilizacji 15V. Dioda musi mieć jak najmniejsze napięcie przebicia, by skok napięcia nie uszkodził rejestrów przesuwanych HV, które są wrażliwe na napięcia powyżej 13.2V.

Dioda ta została wybrana gdyż nie udało się znaleźć diody TVS o napięciu stabilizacji 13V. Transil ma również dużą pojemność, co powoduje, że nie jest ona zalecana do zastosowań z wysokimi częstotliwościami, jednak w tym przypadku nie jest to problemem, gdyż jest to tylko złącze zasilania o stałym napięciu.

7.3.4 Schemat



Rysunek 7.5: Schemat złącza DC-Plug

7.4 Złącze do programowania

7.4.1 Dobór złącza

Złącze usb musi posiadać przynajmniej 12 pinów, ponieważ dopiero w takim układzie jest na złączu są linie D+ i D-, czyli linie danych. Wybrano złącze 16 pinowe, ponieważ takie było dostępne w sklepie.

7.4.2 Opis podłączenia

By ustawić napięcie komunikacji USB-C na 3.3V, zastosowano rezystory podciągające R1 i R2 o wartości 5.1kΩ. Do podłączenia wykorzystano parę różnicową by połączyć linie D+ i D- z ESP32-S3, w celu zminimalizowania zakłóceń CMN (Common Mode Noise).

7.4.3 Zabezpieczenia ESD

W celu zabezpieczenia linii przed przepięciami, zastosowano diody TVS PUSB3AB4Z firmy Nexperia. Diody te mają wystarczająco duże opakowanie by dało się je zlutować ręcznie, napięciem roboczym jest 3.3V, a napięcie stabilizacji wynosi 5V.

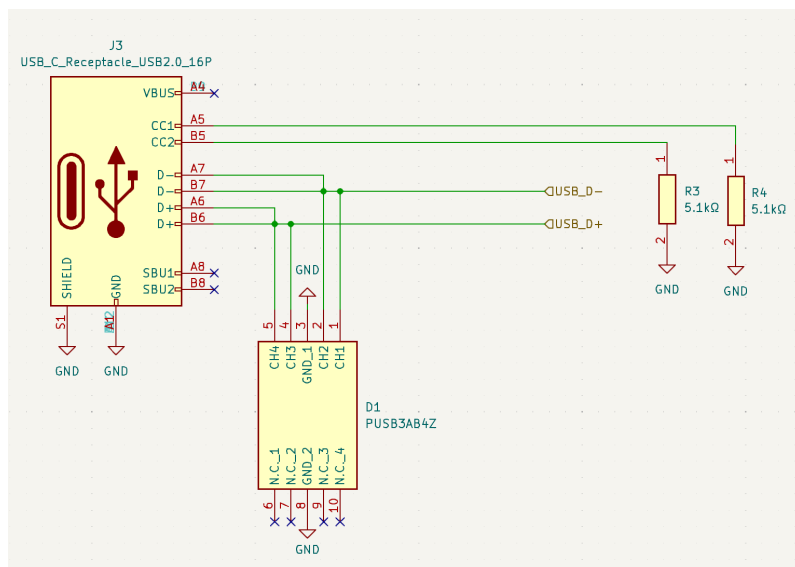
Mimo że jest to napięcie wyższe niż napięcie zasilania ESP32-S3, to nie powinno to stanowić problemu, ponieważ napięcie to pojawi się na krótki czas, a sam esp32-s3 ma również wbudowane zabezpieczenia przed przepięciami.

Wewnętrzne zabezpieczenia według noty katalogowej ESP32-S3:

- Test Standard JS-001; HBM (Human Body Mode) ± 2000 V
- Test Standard JS-002; CDM (Charged Device Model) ± 1000 V

Wynika z tego, że złącze USB-C w dość dobry sposób jest zabezpieczone przed przepięciami.

7.4.4 Schemat



Rysunek 7.6: Schemat złącza USB-C do programowania

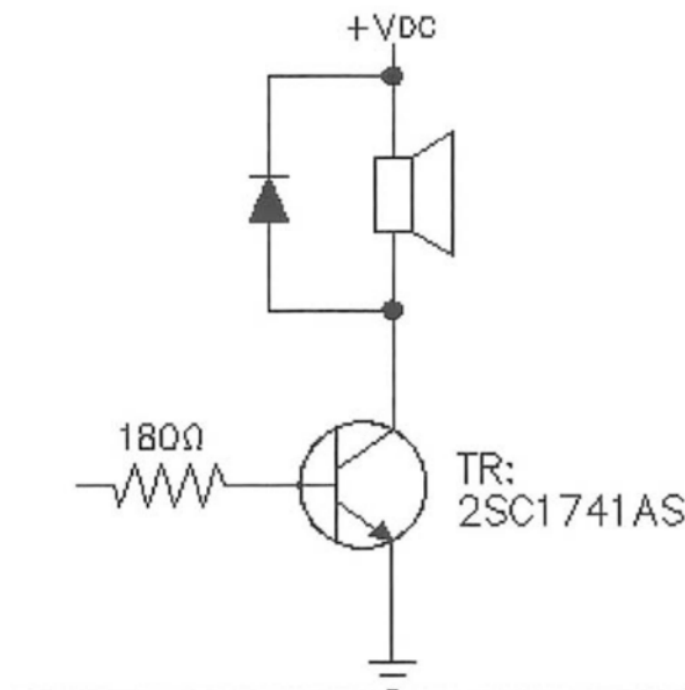
7.5 Buzzer

Kryterium wyboru buzzera było jego napięcie zasilania, głośność oraz jak najmniejszy rozmiar. Wybrano buzzer CEM120342, jest to buzzer piezoelektryczny. Z noty katalogowej[11] wypisano użyteczne parametry na etapie projektowania:

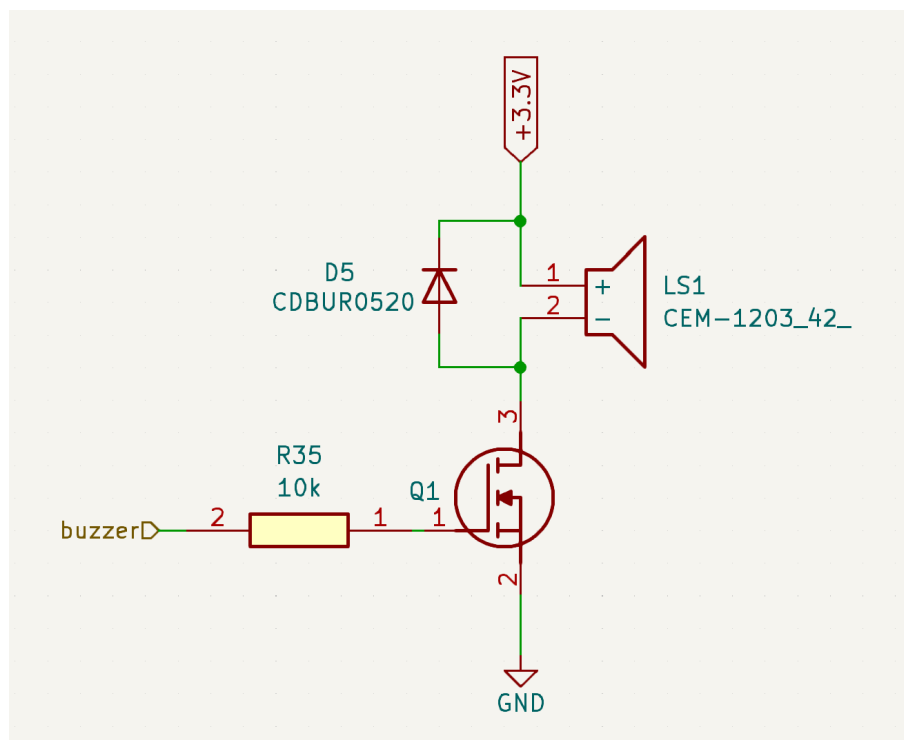
- Napięcie zasilania: 3-5V
- Max prąd zasilania: 35mA
- Częstotliwość: 2,048kHz
- Głośność: 85dB - 95dB
- Średnica: 12mm, wysokość: 8mm
- Rezystancja: 42 Ω
- Typ montażu: THT

Buzzer został podłączony według schematu zalecanego przez producenta. Do sterowania wykorzystano inny tranzystor, użyto tranzystora który służy do sterowania katodami kropek w lampach nixie, by nie dodawać kolejnego elementu do projektu. Tranzystor ten ma prąd kolektora 80mA, co jest wystarczające do sterowania buzzerem.

Buzzer jest dostatecznie mały, jedyną wadą jest montaż THT, który ogranicza miniaturyzację płytki drukowanej.



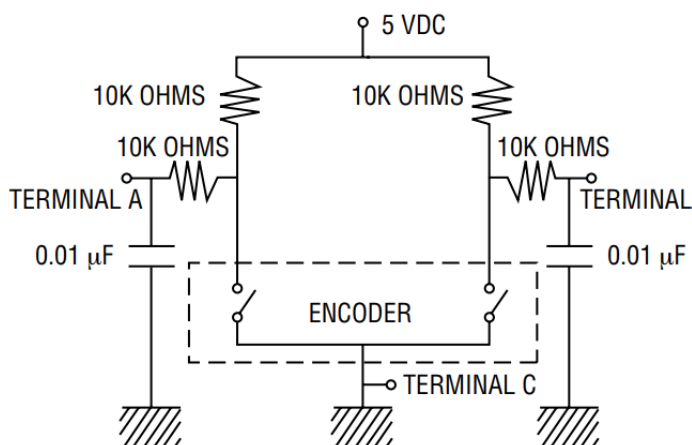
Rysunek 7.7: Schemat zalecany przez producenta[11]



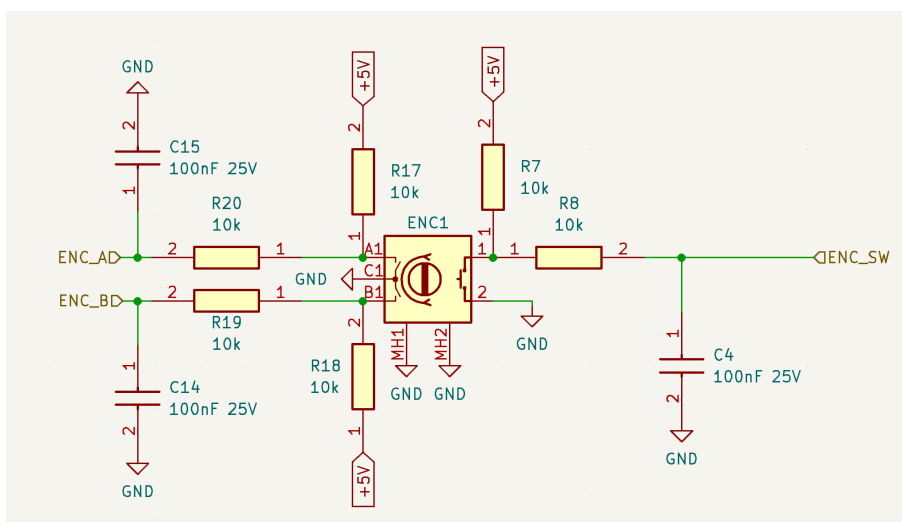
Rysunek 7.8: Gotowy układ z buzzerem

7.6 Encoder

Jedynymi kryteriami wyboru encodera były napięcie zasilania oraz by posiadał on przycisk. Wybrano encoder PEC12R41BBFS0012 firmy Bourns[12], jest to 2 kanałowy encoder o rozdzielczości 12 impulsów na obrót. Jego napięcie robocze wynosi 5V. Producent w karcie katalogowej informuje również, jakie filtry zastosować, by zapobiec zakłóceniom. Dodatkowo dodano filtr dolno-przepustowy, przy przycisku encodera, by zapobiec drganiom styków i nie musieć implementować debouncingu w oprogramowaniu.



Rysunek 7.9: Schemat filtrów zalecany przez producenta[12]

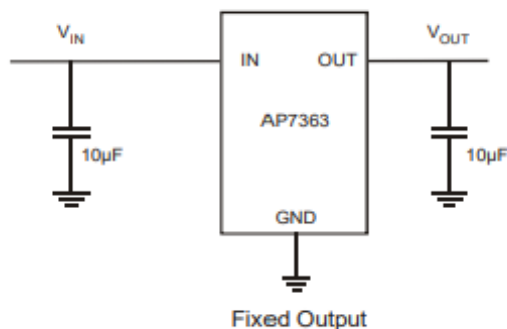


Rysunek 7.10: Gotowy układ encodera

7.7 LDO 5V na 3.3V

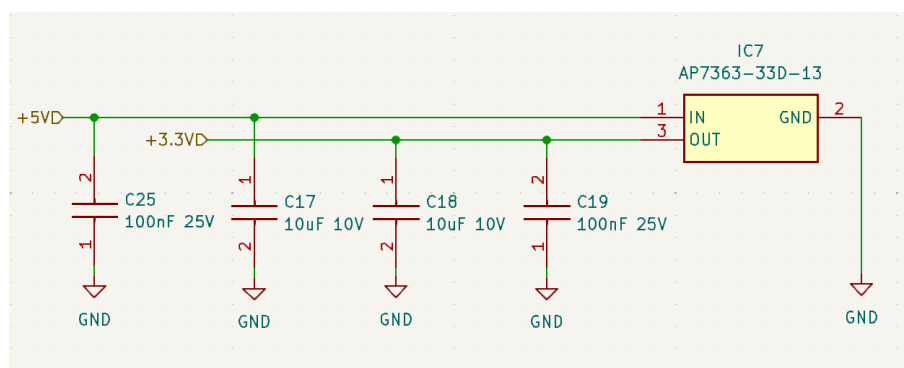
Wymagania jakie musi spełnić LDO to zapewne zasilania mikrokontrolera ESP32 oraz potencjometru cyfrowego. Zgodnie z dokumentacją ESP32[5] pobiera on maksymalnie 340mA prądu, a potencjometr cyfrowy[13] 100mA, co daje łącznie 440mA. Układ LDO ma być zasilany 5V, a na wyjściu ma być 3.3V.

Wybrano układ LDO AP7363 firmy Diodes Incorporated[14], jest to układ LDO o napięciu wyjściowym 3.3V i prądzie wyjściowym 1.5A, co daje duży zapas wydajności prądowej. Układ ten jest dostępny w obudowie TO252, co pozwala na łatwy montaż na płytce drukowanej.



Rysunek 7.11: Schemat podłączenia LDO zalecany przez producenta[14]

Względem schematu zaproponowanego przez producenta, zdecydowano się na zastosowanie dodatkowo kondensator ceramiczny o pojemności 100nF, by zminimalizować zakłócenia. Dodano po kondensatorze na wejście i wyjście układu równolegle do kondensatorów 10uF zaproponowanych przez producenta.



Rysunek 7.12: Gotowy układ LDO

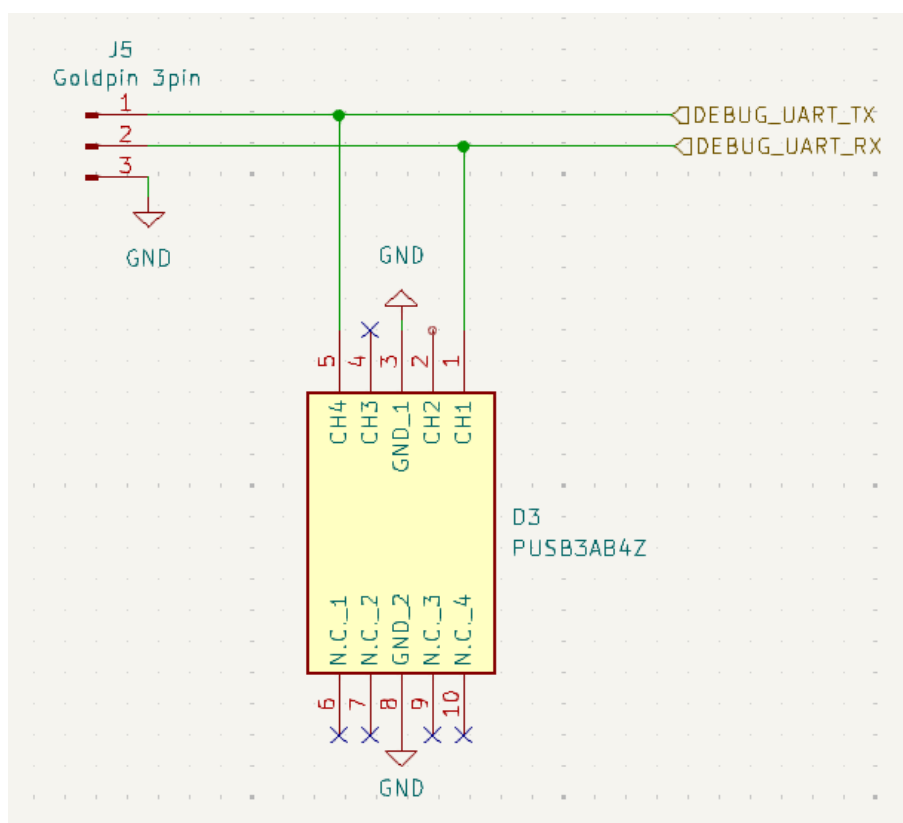
7.8 Złącze do debugowania

Wybrano horyzontalne złącze gold pin, ponieważ łatwo jest podłączyć do niego uniwersalne przewody żeńskie często wykorzystywane w prototypowaniu. Horyzontalna wersja złącza została wybrana by nie płytka była jak najniższa. Dodatkową zaletą tego rozwiązania jest ewentualna możliwość programowania mikrokontrolera w przypadku gdyby nie udało się tego zrobić przez USB.

Złącze będzie wystawia następujące sygnały:

- RX - odbiór danych
- TX - wysyłanie danych
- GND - masa

W celu zabezpieczenia złącza przed ESD, zastosowano diodę TVS, analogiczną co w podrozdziale 7.4.



Rysunek 7.13: Gotowy układ do debugowania przez UART

8 Oprogramowanie

Rozdział zawiera opis oprogramowania, które zostało napisane do mikrokontrolera ESP32-S3 oraz krótki opis integracji z serwerem czasu oraz serwerem Home Assistant.

9 Testowanie

W tym rozdziale opisano testowanie poszczególnych modułów oraz całego układu, które pozwoliło na sprawdzenie poprawności działania oraz zidentyfikowanie błędów.

10 Podsumowanie

Bibliografia

- [1] *Zegar Nixie - nostalgiczna elegancja*. 4/2024 Kwiecień (16). Zrozumieć Elektronikę z Piotrem Góreckim. Kw. 2024. URL: <https://piotr-gorecki.pl/wp-content/uploads/2024/03/ZE2404.pdf>.
- [2] *Lampa cyfrowa. Schemat elektryczny lampy NIXIE*. Wikipedia. 2023. URL: https://pl.wikipedia.org/wiki/Lampa_cyfrowa.
- [3] *Kompendium wiedzy o lampach Nixie*. Rev. 2. rafalbartoszak. Kw. 2019. URL: <https://rafalbartoszak.pl/kompendium-wiedzy-o-lampach-nixie/>.
- [4] *Internet of Things Projects with ESP32. Build exciting and powerful IoT projects using the all-new Espressif ESP32*. Agus Kurniawan. Mar. 2019.
- [5] *ESP32-S3 Series Datasheet*. v1.9. Espressif Systems. URL: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf.
- [6] *ESP32-S3: Which Pins Should I Use?* Chris Greening. Paź. 2023. URL: <https://www.atomic14.com/2023/11/21/esp32-s3-pins>.
- [7] *Expert Network Time Protocol: An Experience in Time with NTP (Expert's Voice)*. 9781590594841. Peter Rybaczek. 2005.
- [8] *eSezam 1.0/Synchronizacja w telekomunikacji/Podręcznik 2. Synchronizacja czasu*. Ośrodek Kształcenia na Odległość Politechniki Warszawskiej. URL: <https://esezam.okno.pw.edu.pl/mod/book/view.php?id=76&chapterid=1632>.
- [9] *Network Time Protocol. Struktura warstw STRATUM 0-15*. Wikipedia. 2024. URL: https://pl.wikipedia.org/wiki/Network_Time_Protocol.
- [10] *RFT tube data book and translation*. Z570M/Z5700M. RFT electronic. URL: <https://www.tube-tester.com/sites/nixie/data/z5730m/z5730m.html>.
- [11] *Part No: CEM-1203(42) Description: magnetic buzzer*. CUI INC. List. 2006. URL: [https://componentsearchengine.com/Datasheets/1/CEM-1203\(42\).pdf](https://componentsearchengine.com/Datasheets/1/CEM-1203(42).pdf).
- [12] *PEC12R - 12 mm Incremental Encoder datasheet*. Bourns. Maj 2018. URL: <https://www.mouser.pl/datasheet/2/54/PEC12R-777795.pdf>.
- [13] *7-Bit Single I2C™ Digital POT with Volatile Memory in SC70*. Microchip. Mar. 2009. URL: <https://datasheet.datasheetarchive.com/originals/distributors/Datasheets-DGA14/624353.pdf>.
- [14] *1.5A LOW QUIESCENT CURRENT, FAST TRANSIENT ULTRA-LOW DROPOUT LINEAR REGULATOR*. DS35059. Rev. 10 - 2. Diodes Incorporated. Paź. 2021. URL: <https://www.diodes.com/assets/Datasheets/AP7363.pdf>.

Spis rysunków

2.1	Schemat elektryczny lampy nixie ze wspólną anodą[2]	10
3.1	Rozkład pinów mikrokontrolera ESP32-S3[6]	14
4.1	Struktura serwerów czasu w protokole NTP[9]	17
5.1	Ogólna koncepcja układu	19
5.2	Ogólny schemat komunikacji z serwerem	20
6.1	Schemat blokowy projektu	23
6.2	Prototyp układu z lampą nixie przy napięciu zasilania 150 V	25
6.3	Prototyp układu z lampą nixie przy napięciu zasilania 220 V	25
7.1	Spadek pojemności kondensatora ceramicznego wraz ze wzrostem napięcia	28
7.2	Schemat przetwornicy 12V na HV	31
7.3	Tabela doboru komponentów z noty katalogowej	32
7.4	Schemat złącza DC-Plug	33
7.5	Schemat złącza DC-Plug	34
7.6	Schemat złącza USB-C do programowania	35
7.7	Schemat zalecany przez producenta[11]	36
7.8	Gotowy układ z buzzerem	37
7.9	Schemat filtrów zalecany przez producenta[12]	38
7.10	Gotowy układ encodera	38
7.11	Schemat podłączenia LDO zalecany przez producenta[14]	39
7.12	Gotowy układ LDO	39
7.13	Gotowy układ do debugowania przez UART	40

Spis tablic

1	Tabela opłacalności sposobów sterowania lampami nixie	12
2	NTP – format komunikatu	18