



WYDZIAŁ ELEKTRONIKI,  
TELEKOMUNIKACJI  
I INFORMATYKI

Imię i nazwisko studenta: Kamil Dubilis

Nr albumu: 189021

Poziom kształcenia: Studia pierwszego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Elektronika i telekomunikacja

Profil: Komputerowe systemy elektroniczne

## PROJEKT DYPLOMOWY INŻYNIERSKI

Tytuł projektu w języku polskim: Nadajnik systemu zdalnego sterowania obsługiwany gestami

Tytuł projektu w języku angielskim: Remote control transmitter controlled by gestures

Opiekun pracy: dr hab. inż. Paweł Wierzba



### **Streszczenie**

Aliquam erat volutpat. Nunc eleifend leo vitae magna. In id erat non orci commodo lobortis. Proin neque massa, cursus ut, gravida ut, lobortis eget, lacus. Sed diam. Praesent fermentum tempor tellus. Nullam tempus. Mauris ac felis vel velit tristique imperdiet. Donec at pede. Etiam vel neque nec dui dignissim bibendum. Vivamus id enim. Phasellus neque orci, porta a, aliquet quis, semper a, massa. Phasellus purus. Pellentesque tristique imperdiet tortor. Nam euismod tellus id erat.

**Słowa kluczowe:** a, b, c

**Dziedzina nauki i techniki, zgodne z wymogami OECD:** nauki inżynieryjno–techniczne: automatyka, elektronika, elektrotechnika i technologie kosmiczne

### **Abstract**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec hendrerit tempor tellus. Donec pretium posuere tellus. Proin quam nisl, tincidunt et, mattis eget, convallis nec, purus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla posuere. Donec vitae dolor. Nullam tristique diam non turpis. Cras placerat accumsan nulla. Nullam rutrum. Nam vestibulum accumsan nisl.

**Keywords:** a, b, c

## Spis treści

## Wykaz ważniejszych oznaczeń i skrótów

# 1 Wstęp i cel pracy

Aliquam erat volutpat[1]. Nunc eleifend leo vitae magna. In id erat non orci commodo lobortis. Proin neque massa, cursus ut, gravida ut, lobortis eget, lacus. Sed diam. Praesent fermentum tempor tellus. Nullam tempus. Mauris ac felis vel velit tristique imperdiet. Donec at pede. Etiam vel neque nec dui dignissim bibendum. Vivamus id enim. Phasellus neque orci, porta a, aliquet quis, semper a, massa. Phasellus purus. Pellentesque tristique imperdiet tortor. Nam euismod tellus id erat.

## 2 Koncepcja układu

Koncepcją układu jest zaprojektowanie i wykonanie budzika opartego o lampy Nixie, który będzie synchronizował się dzięki odpytywaniu serwera NTP oraz z wykorzystaniem modułu RTC wbudowanego w mikrokontroler.

### 2.1 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 będzie można ustawić na nich animacje podczas alarmu.
- Alarm będzie sygnalizowany dźwiękiem oraz miganiem pasków LED.
- Dźwięk może być grany 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 projektowe będą skutkować bardzo wszechstronnym budzikiem, który będzie miał wygląd retro, ale z nowoczesnymi funkcjonalnościami.

### 2.2 Wstępne obliczenia i założenia

Pierwszą rzeczą, która została wykonana to zakup lamp Nixie, które będą wykorzystane w projekcie. Przez coraz bardziej ograniczony dostęp do tych lamp, zdecydowano się na zakup lamp Z570M, które zostały zakupione w ilości 6 sztuk. Lampy te mają 10 cyfr oraz kropkę dziesiętną, która nie jest nam potrzebna, ale została przewidziana możliwość jej sterowania w razie potencjalnego zastosowania w przyszłości.

Kluczowe parametry zastosowanych lamp nixie odczytane z noty katalogowej:

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

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:



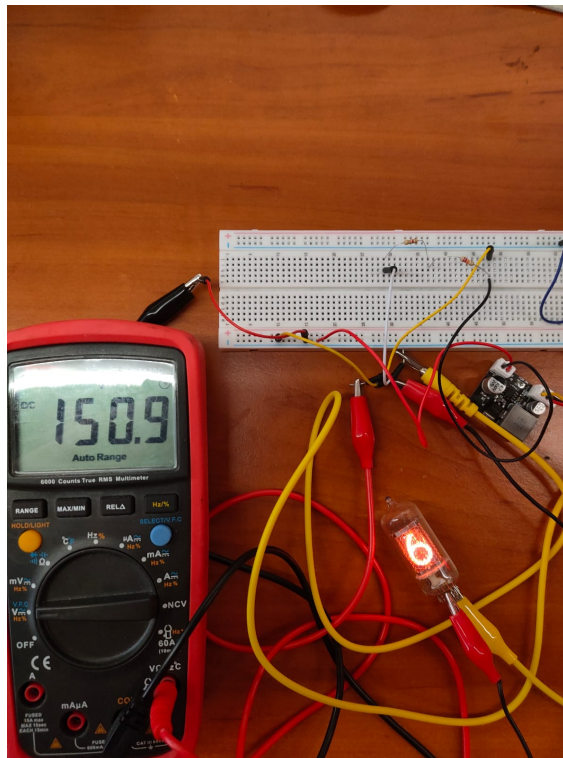
- Napięcie wejściowe: 5-12
- Napięcie wyjściowe: 150-220
- Prąd wyjściowy: 20

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

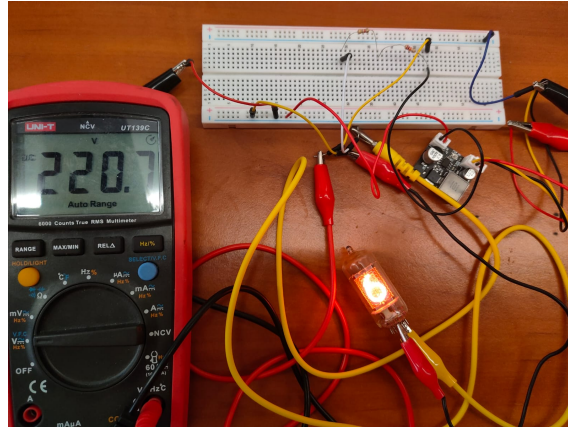
$$R = \frac{U_{max} - U_{zap}}{I_{kat}} = \frac{220 - 150}{2} = 35 \quad (1)$$

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

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



Rysunek 2.1: Prototyp układu z lampą nixie przy napięciu zasilania 150



Rysunek 2.2: Prototyp układu z lampą nixie przy napięciu zasilania 220

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

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

Z testów można wyciągnąć następujące wnioski:

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

## 2.3 Dobór mikrokontrolera

Następnie wybrano mikrokontroler, który będzie użyty w projekcie. Wybrano mikrokontroler ESP32-S3, który jest nowym mikrokontrolerem z wbudowanym modułem Wi-Fi oraz Bluetooth. Mikrokontroler ten ma wbudowany programator co pozwoli na łatwe wgrywanie oprogramowania, bez zewnętrznego programatora, a także ma wbudowany moduł RTC, co pozwoli na zachowanie czasu, pomiędzy zapytaniami do serwera NTP.

Wybrany układ ma również wystarczająco dużo pinów GPIO, by móc obsłużyć wszystkie lampy nixie, paski LED, głośnik oraz przyciski. Posiada on układ dwu rdzeniowy, co w razie potrzeby pozwoli na obsługę wielowątkowości. Zegar procesora ma częstotliwość 240, co w zupełności wystarczy do obsługi wszystkich elementów układu.

## 2.4 Dobór paska LED

Do budzika wybrano adresowalne paski LED WS2812B, które są bardzo popularne w projektach DIY, ponieważ są łatwe w obsłudze, mają wbudowany kontroler, który pozwala na sterowanie każdym diodą z osobna oraz są dostępne w różnych długościach i kolorach. Wybrano pasek z diodami RGB, co pozwoli na wyświetlanie wielu kolorów. Moc na metr wynosi 18. Pasek posiada też stopień ochrony IP65, czyli powinien być odporny na kurz oraz wilgoć.

## 2.5 Koncepcja zasilania

Projekt sekcji zasilania został rozpoczęty od oszacowania mocy potrzebnej do zasilania całego układu. 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. Z deklaracji producenta paska LED wynika, że moc na metr wynosi 18, co daje:

$$P_{LED} = 18 \cdot 0.3 = 5.4 \quad (2)$$

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

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

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

$$P_{ESP32} = 340 \cdot 3.3 = 1.122 \quad (4)$$

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

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

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

$$P_{Nixie} = \frac{U_{Nixie} \cdot I_{Nixie}}{Sprawno} = \frac{220 \cdot 12}{0.7} = 3.43 \quad (6)$$

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

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

Szacunkowa moc potrzebna do zasilania całego układu wynosi około 10, co powoduje problem z zasilaniem z gniazd USB w komputerze, ponieważ maksymalna moc, jaką można pobrać z gniazda USB wynosi 2.5, co jest zdecydowanie za mało.

Problem jest też projekt przetwornicy HV zasilanej z 5V, ponieważ przetwornica boost przy takiej aplikacji miała by bardzo wysokie wypełnienie rzędu 95 %, co prowadziło by do dużych strat mocy.

Alternatywą jest przetwornica flyback, która ma mniejsze wypełnienie i była by bardziej wydajna, ale jest bardziej skomplikowana w projektowaniu i montażu. Podczas prób projektowania takiej przetwornicy, problem okazało się znaleźć odpowiedni transformator, który miał by odpowiednie parametry, a także byłby dostępny w sklepach elektronicznych.

W związku z tymi problemami, zdecydowano się na zasilanie z zewnętrznego zasilacza 12V, który będzie zasilany z gniazda sieciowego, podłączonego do budzika za pomocą złącza DC-Plug. Do programowania mikrokontrolera będzie służyć złącze USB-C.

Ostatecznie sekcja zasilania została podzielona na kilka podsekcji:

- Przetwornica 12V na 5V do zasilania pasków LED.
- Regulowana przetwornica 12V na HV do zasilania lamp nixie.
- LDO 5V na 3.3V do zasilania mikrokontrolera.

## 2.6 Koncepcja sterowania lampami nixie

Kolejnym krokiem było zaprojektowanie sekcji sterowania lampami nixie, która będzie odpowiedzialna za wyświetlanie odpowiednich cyfr na lampach. Jest kilka sposobów na sterowanie lampami:

- Sterowanie bezpośrednie - każda lampka ma swoje wejście i jest sterowana osobno za pomocą tranzystora HV połączonego z mikrokontrolerem. Niestety wymagane jest wiele pinów GPIO również trzeba użyć ponad 60 tranzystorów HV, co jest bardzo kosztowne.
- 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, aby uniknąć efektu ghostingu.

W projekcie zdecydowano się na wykorzystanie rejestrów przesuwanych HV. Znalaziono 32 kanałowy rejestr HV5530PG-G firmy Microchip mogące pracować z napięciem do 300 oraz posiadające zatrask i wyjście Open Drain.

Rodzaj wyjścia open drain jest bardzo istotny, ponieważ na wyjściu znajduje się stan wysokiej impedancji, co nie powoduje zwierania wszystkich wyjść do masy. W przypadku wyjścia push-pull, stan wysoki jak i niski powodował by zapalenie się wszystkich lamp, ponieważ spadek napięcia na lampach nawet w stanie wysokim byłby na tyle duży, że lampy były by w stanie zapłonąć.

Do sterowania 6 lampami potrzebne są 2 rejestry HV5530PG-G, co daje 64 kanały, co pozwoli na sterowanie 6 lampami nixie oraz 2 neonówkami, które służą jako separator między godzinami, minutami i sekundami. Do sterowania kropkami dziesiętnymi, wykorzystano tranzystory HV połączone z mikrokontrolerem.

Sterowanie lampami z wykorzystaniem rejestru jest proste, wymaga wgrania do rejestru 64 bitów, które odpowiadają za stan lamp oraz przełączanie zatrasku co sekundę, aby zmienić ich stan.

## 2.7 Dobór głośnika

Jako głośnik do budzika wybrano głośnik piezoelektryczny (buzzer) bez generatora. Wybór był podyktowany małym rozmiarem, ponieważ wymaganiem jest by budzik był jak najmniejszy, a także niską ceną oraz prostotą w obsłudze. Buzzer bez generatora pozwala na generowanie dźwięku za pomocą prostego sygnału prostokątnego, co jest bardzo proste do zaimplementowania, istnieje wiele bibliotek do generowania dźwięku na mikrokontrolerze.

## 3 Budowa układu

### 3.1 Przetwornica 12V na 5V

#### 3.1.1 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.

#### 3.1.2 Założenia projektowe

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

#### 3.1.3 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 $\Omega$ )	R3 (k $\Omega$ )	L1 ( $\mu$ H)			C6 + C7 + C8 ( $\mu$ 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 3.1: Tabela doboru komponentów z noty katalogowej

Na podstawie tabeli dobrano następujące wartości komponentów:

- C1: 22 $\mu$ F 10V
- C29: 22 $\mu$ F 10V
- L1: 2.2 $\mu$ H 9.2A 14.5m $\Omega$
- R1: 56k $\Omega$
- R2: 10k $\Omega$

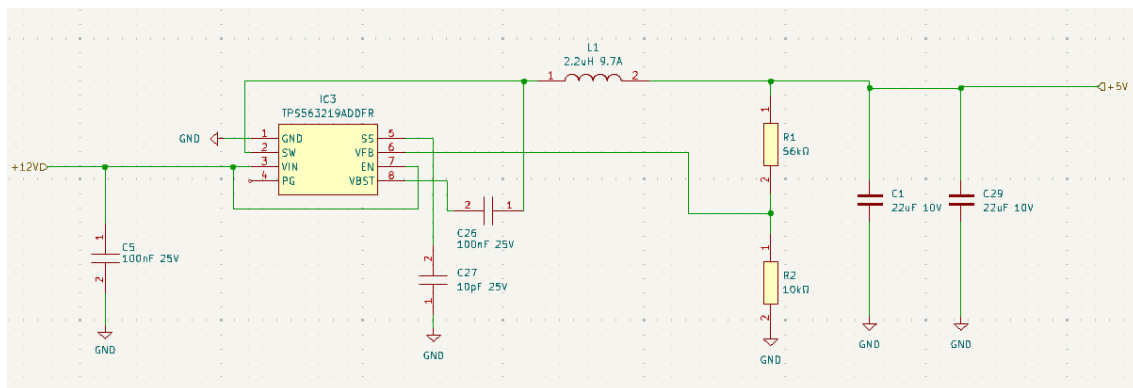
Kondensator podpięty pod pin SS(soft start) oraz kondensator podpięty pod pin VBST, został skopiowany z układu z noty katalogowej, gdyż nie jest to krytyczny element i nie ma potrzeby doboru wartości pod kątem zastosowania w zegarze.

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  $100\text{nF}$  na pinie VCC, by zredukować szумы z linii zasilania.

### 3.1.4 Schemat



Rysunek 3.2: Schemat złącza DC-Plug

## 3.2 Złącze DC-Plug

### 3.2.1 Dlaczego złącze DC-Plug?

Złącze DC jest złączem, które jest powszechnie stosowane w zasilaczach, co czyni je uniwersalnym rozwiązaniem, odpowiednim do tej aplikacji. Wybrano złącze o jak najmniejszych rozmiarach, by zegar był jak najcieńszy. Złącze to zostało wybrane również ze względu na to, że łatwo znaleźć zasilacz 12V z takim złączem.

### 3.2.2 Opis podłączenia

Do złącza dołączono 3 kondensatory filtrujące, które mają za zadanie zminimalizować zakłócenia zasilania. 2 kondensatory o pojemności 100uF w celu zminimalizowania zakłóceń niskich częstotliwości, oraz 1 kondensator o pojemności 100nF w celu zminimalizowania zakłóceń wysokich częstotliwości.

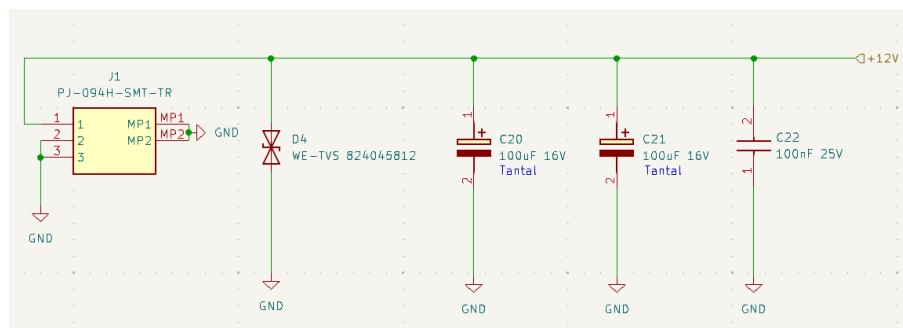
kondensatory 100uF są kondensatorami tantalowymi, a kondensator 100nF jest kondensatorem ceramicznym. Wykorzystano te rodzaje kondensatorów, ponieważ są to kondensatory o długiej żywotności, a także są to kondensatory o małych rozmiarach w przeciwieństwie do kondensatorów elektrolitycznych.

### 3.2.3 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.

### 3.2.4 Schemat



Rysunek 3.3: Schemat złącza DC-Plug

### 3.3 Złącze USB-C do programowania

#### 3.3.1 Dlaczego USB-C?

Złącze zostało wybrane tylko jako narzędzie do programowania, nie będzie ono zasilać zegara, ponieważ przez zastosowanie paska LED, zegar będzie pobierał więcej prądu niż jest w stanie dostarczyć złącze USB-C, bez wykorzystania power delivery.

Nie chcemy korzystać z power delivery, ponieważ nie jest on kompatybilny ze wszystkimi zasilaczami USB-C. Przez co zegar nie mógłby być zasilany np z złącza usb w komputerze, a wypadku takiego podłączenia, zegar mógłby przestać działać, dlatego zasilanie zegara będzie zrealizowane przez złącze DC-plug 12V.

Standard usb-c jest obecnie najbardziej popularnym złączem w urządzeniach mobilnych, więc jest to najwygodniejsze złącze do programowania. Złącze USB-C jest również rozwiązanie przyszłościowe.

#### 3.3.2 Opis podłączenia

Złącze USB-C będzie podłączone bezpośrednio do ESP32-S3, ponieważ ESP32-S3 posiada wbudowany kontroler USB-C, który obsługuje protokół USB 2.0. ESP32-S3 posiada również wbudowany programator, więc nie potrzebujemy dodatkowego programatora.

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

#### 3.3.3 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.

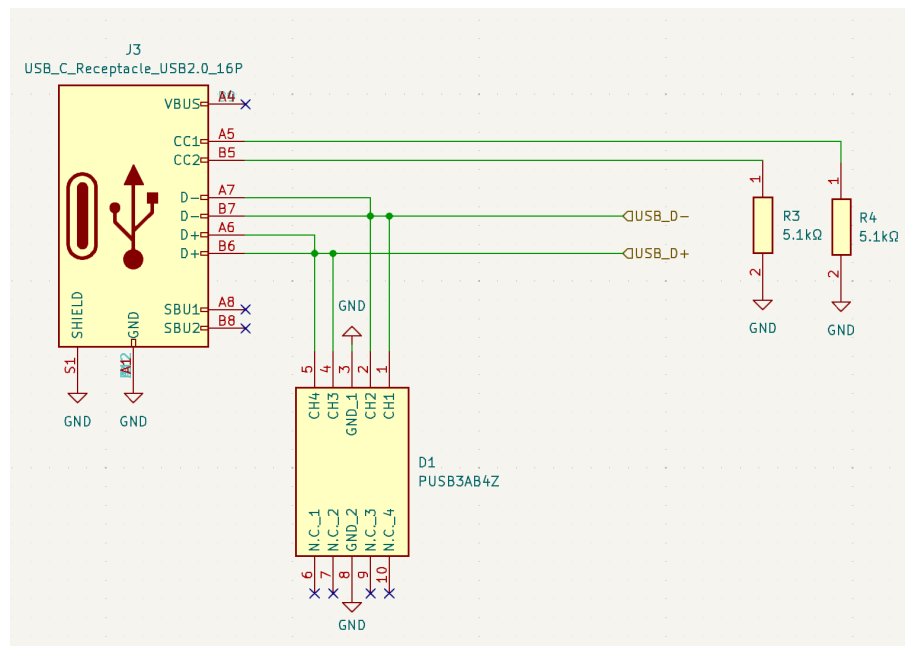
Wewnętrzne zabezpieczenia według noty katalogowej ESP32-S3:

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

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



### 3.3.4 Schemat



Rysunek 3.4: Schemat złącza USB-C do programowania

## Bibliografia

- [1] *ISM330DHCX*. DS13012. Rev. 7. STMicroelectronics. List. 2020. URL: <https://www.st.com/resource/en/datasheet/ism330dhcx.pdf>.

## Spis rysunków

## Spis tablic