

Szczegółowa koncepcja układu została podzielona na poszczególne sekcje układu.

0.0.1 Sterowanie lampami nixie

Kluczowym jest wybór sterownia lampami nixie, ponieważ na podstawie tego wyboru zostanie zaprojektowany pozostała część układu. Zgodnie z analizą przeprowadzoną w podrozdziale ??, zdecydowano się na sterowanie lampami za pomocą rejestrów przesuwnych wysokiego napięcia. Zastosowanie tego rozwiązania pozwala na zredukowanie ilości potrzebnych wyprowadzeń mikrokontrolera do sterowania lampami. Ten typ sterowania jest łatwy w implementacji, wymaga jedynie wgrania odpowiednich danych do rejestrów przesuwnych, a następnie zatrzaśnięcie wyjścia, co pozwala na wyświetlenie odpowiedniej cyfry na lampie.

Niezależnie od wyboru lamp każda ma 10 katod z cyframi i jedną katodę od kropki dziesiętnej, więc wymagane jest 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 wysokiego napięcia podłączone do wyjść mikrokontrolera, ponieważ nie opłacalnym jest dodawanie kolejnego rejestru przesuwne wysokiego napięcia tylko do sterowania kropkami dziesiętnymi.

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

0.0.2 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 sterowania lampami oraz kropkami dziesiętnymi, 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 ważną zaletę w postaci kontrolera USB/JTAG, dzięki czemu nie jest potrzebny dodatkowy programator do programowania układu. Jest to też popularny mikrokontroler, dla którego istnieje wiele bibliotek i przykładów.

0.0.3 Źródło dźwięku

Jako źródło dźwięku wybrano głośnik piezoelektryczny, który jest prostym elementem i nie wymaga dodatkowego wzmacniacza. Jego zaletami są niski pobór prądu, małe rozmiary i niska cena. Sterowanie odbywa się za pomocą sygnału PWM, który pozwala generować proste melodie. Głośnik piezoelektryczny jest wystarczająco głośny, aby być słyszalnym w pomieszczeniu, w którym będzie znajdował się budzik.

0.0.4 Pasek LED

Pasek LED będzie służył jako dodatkowe źródło światła, które będzie sygnalizować alarm, oraz jako element estetyczny. Pasek ten musi zawierać w sobie adresowane diody LED, które poz-

wolą na wyświetlanie różnych kolorów (RGB). Rozwiązanie jest proste w implementacji, wystarczy podłączenie do pinu GPIO mikrokontrolera i za pomocą sygnału PWM można sterować jasnością.

0.0.5 Interfejs użytkownika

Interfejs użytkownika będzie składał się z enkodera obrotowego z przyciskiem, który będzie służył do regulacji jasności, a przycisk do wyłączania alarmu. Enkoder obrotowy jest też na tyle uniwersalnym rozwiązaniem, które pozwala na mnogość kombinacji sterowania, ale wygodniejsze jest korzystanie z aplikacji mobilnej.

0.0.6 Zasilanie

Elementem wymagającym najwięcej pracy jest zaprojektowanie przetwornicy wysokiego napięcia do zasilania lamp Nixie.

Możliwe są dwa rozwiązania:

- Przetwornica typu flyback
- Przetwornica typu boost

Przetwornica typu flyback ma zaletę w postaci izolacji galwanicznej między wejściem a wyjściem oraz umożliwia zaprojektowanie przetwornicy z napięciem zasilania 5 V. 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 problem ze znalezieniem transformatora, zdecydowano się na przetwornicę typu boost, która jest prostsza w implementacji i tańsza. Wadą jest to, że nie ma izolacji galwanicznej między wejściem a wyjściem, ale w tym przypadku nie jest to wymagane. Następnym krokiem było wybranie jednej z dwóch konfiguracji układu:

- USB-C jako zasilanie zewnętrzne, przetwornica typu boost z zasilacza 5 V na 12 V i przetwornica typu boost z zasilacza 12 V na wysokie napięcie
- Złącze DC jako zasilanie zewnętrzne, przetwornica typu boost z zasilacza 12 V na wysokie napięcie oraz przetwornica typu buck z zasilacza 12 V na 5 V

Wadą pierwszej konfiguracji jest to, że niemożliwym będzie jednoczesne programowanie i zasilanie układu, ponieważ złącze USB-C podłączone do komputera ma niską wydajność prądową. Dla USB 2.0 wynosi ona 500mA, a dla USB 3.0 wynosi 900mA, co jest niewystarczające dla zasilania układu. Brak możliwości jednoczesnego zasilania i programowania było by poważnym problemem na etapie pisania oprogramowania, więc zdecydowano się na drugą konfigurację.

Wybrano więc przetwornicę typu boost, która będzie zasilana z zasilacza 12 V, a wyjście będzie podłączone do anod lamp nixie. Do tego będzie potrzebne zasilanie 5 V dla paska LED oraz 3.3 V dla mikrokontrolera. Moduł zasilania 5 V będzie zasilał pasek LED, który potrafi pobrać większy prąd, to ze względu na zachowanie wysokiej efektywności, zdecydowano się na przetwornicę typu buck. Moduł zasilania 3.3 V będzie zasilaniem mikrokontrolera, więc wystarczy zastosować stabilizator liniowy.

Można, więc podzielić zasilanie na 3 pod moduły:

- Przetwornica typu boost z zasilacza 12 V na wysokie napięcie
- Przetwornica typu buck z zasilacza 12 V na 5 V

- Stabilizator liniowy z zasilacza 5 V na 3.3 V

Wynika z tego, że urządzenie będzie posiadać 3 złącza:

- Złącze USB-C do programowania mikrokontrolera
- Złącze DC do zasilania układu
- Złącze typu goldpin jako złącze do komunikacji szeregowej wykorzystywane w procesie uruchamiania układu

0.0.7 Szczegółowy schemat projektu

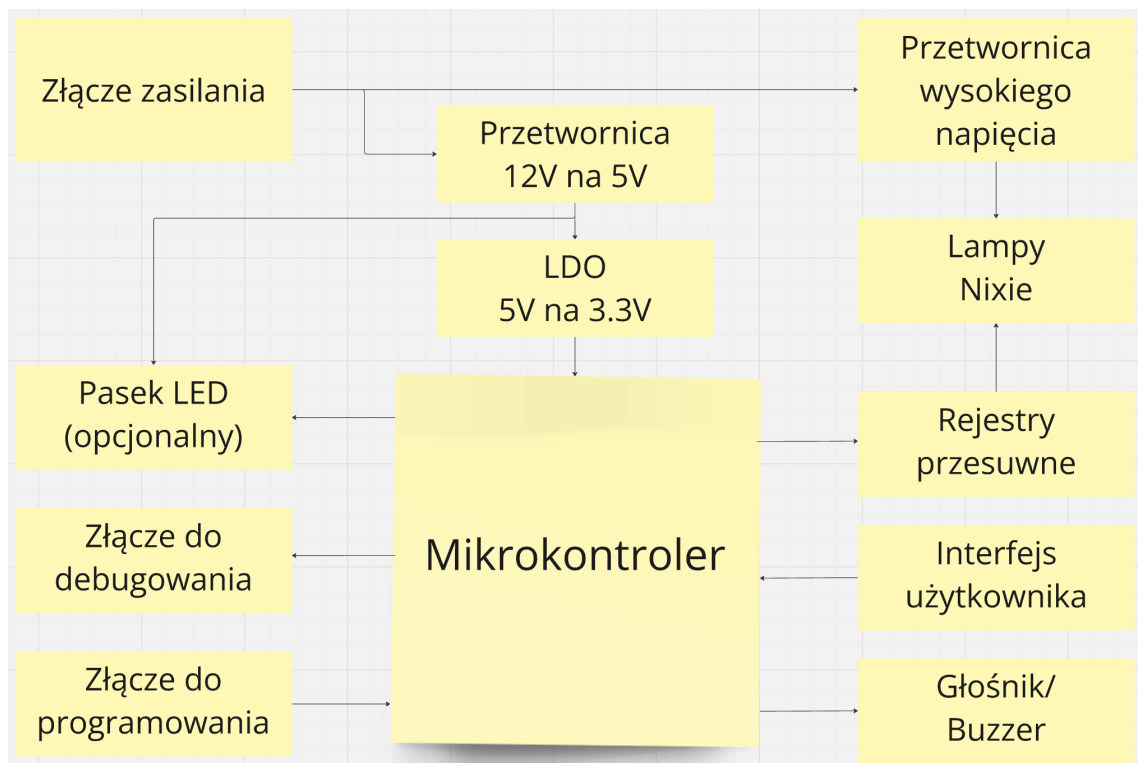


Figure 0.1: Schemat blokowy projektu