



POLITECHNIKA POZNAŃSKA
WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI
KIERUNEK: ELEKTRONIKA I TELEKOMUNIKACJA
**Specjalność: MULTIMEDIA I ELEKTRONIKA
POWSZECHNEGO UŻYTKU**

PRACA DYPLOMOWA INŻYNIERSKA

PŁYTKA STERUJĄCA EKSPRESEM DO HERBATY

Promotor pracy:

dr hab. inż. Maciej Krasicki

Autor:

Łukasz Nowarkiewicz

Poznań, 2023

Streszczenie [pl]

Głównym celem całego projektu było zaprojektowanie działającej płytki sterującej ekspresem do herbaty, realizacja oraz testy układu. Do jego zaprojektowania użyto programu Kicad. Płytki PCB jest czterowarstwowa, wykonana na laminacie FR-4. Zawiera komponenty wykonane w technologii montażu powierzchniowego (SMT) oraz przewlekłanego (THT). Po wykonaniu projektu oraz skompletowaniu dokumentacji produkcyjnej, złożono zamówienie na płytę w firmie JLPCB. Montaż komponentów został wykonany za pomocą maszyny *pick and place (PnP)* oraz ręcznie. Uruchomiano poszczególne fragmenty układu, aż upewniono się, że układ jest pełni sprawny. W tym celu zostało napisane oprogramowanie, służące do sekwencyjnego sterowania poszczególnymi wyjściami. Przeprowadzono testy elektryczne jak i termiczne za pomocą kamery termowizyjnej.

Streszczenie [ang]

The main goal of the whole project was to design a working control board for a tea express, implement and test the system. The Kicad program was used to design it. The PCB board is four-layered, made of FR-4 laminate. It contains components made in both surface mount technology (SMT) and through-hole technology (THT). After completing the design and assembling the production documentation, an order for the board was made from the company JLPCB. The components were assembled using a pick and place (PnP) machine and by hand. Individual parts of the system were started one by one until the complete device was fully operational. To this end, a piece of software was written to sequentially control individual outputs. Electrical and thermal tests were conducted using a thermal imaging camera.

Wstęp

Ekspres do herbaty, działający na podobnej zasadzie co dobrze znane i powszechnie używane ekspresy kawowe, zdaje się być unikatowym pomysłem. Aktualnie na rynku nie są dostępne urządzenia, które zaparzałyby herbatę wykorzystując do tego liściastą formę produktu i dobierały odpowiedni proces zaparzania. Taki sposób parzenia gwarantowałby najwyższą jakość finalnego napoju i smak porównywalny z herbatą zaparzaną w sposób tradycyjny. Urządzenie jednak umożliwiaoby użytkownikom zaoszczędzić czas. Przeznaczone byłoby dla użytkowników domowych.

W niniejszej pracy skupiono się na elektronicznej części urządzenia, jaką jest płytka sterująca ekspresem do herbaty. Główną tematyką pracy jest przedstawienie koncepcji projektu płytki sterującym ekspresem, przedstawienie poszczególnych etapów projektowania: analiza potrzeb użytkownika końcowego, analiza procesu, definicja funkcji urządzenia, projekt schematu płytki, projekt płytki, generacja plików fabrykacyjnych, zamówienie PCB i komponentów, montaż komponentów na maszynie *pick and place* (PnP), montaż ręczny elementów przewlekanych (THT), uruchomienie i testy. Wymienione etapy są po kolejno przedstawione i opisane szczegółowo w kolejnych rozdziałach.

Sam układ ma za zadanie sterowanie urządzeniami wykonawczymi ekspresu, takimi jak podajniki śrubowe, na przykład do suszonych owoców (ogólnie do produktów w formie granulatu o frakcji od około 2 mm do 8 mm), podajników opartych na pompkach perystaltycznych (do płynów, jak miód, syrop klonowy, soki owocowe, itp.), grzałkami do wody, serwomechanizmami, siłownikami, jak i zbieranie oraz przetwarzanie informacji z czujników, takich jak poziomu cieczy, temperatury, podczerwonych, odbiciowych oraz wszystkich innych, jakie w przyszłości zostaną jeszcze zaprojektowane. W ekspresie wybór herbaty oraz ustawienia mają odbywać się docelowo na komputerze jednopłytkowym Raspberry PI 2 z zainstalowanym systemem Windows 10 iOT w połączeniu z ekranem dotykowym — także konieczne będzie zapewnienie komunikacji pomiędzy płytą sterującą a głównym komputerem. Prace nad samym ekspresem cały czas trwają, także nie jest znana finalna wersja urządzenia.

Niniejsza praca oparta jest głównie na zdobytej wiedzy i własnym doświadczeniu autora, podparty literaturą i badaniami z zakresu: procesu parzenia herbaty, elektroniki, elektrotechniki, automatyki, informatyki, projektowania urządzeń AGD, projektowania płyt PCB oraz programowania mikrokontrolerów.

Słowa kluczowe:

Raspberry Pi Pico, Ekspander IO 16bit I2C, Przetwornica *buck* XL4015, Stabilizator liniowy LM7805, NCP1117, Konwerter stanów logicznych, Arduino IDE, UART, PWM, I2C.

Spis treści:

1. Słownik akronimów i skrótów użytych w pracy	5
2. Nomenklatura.....	5
3. Cel i założenia projektu	6
3.1 Rozwiązania dostępne na rynku.....	6
3.2 Założenia funkcjonalne, analiza potrzeb	8
4. Środowisko sprzętowe	9
4.1 Schemat blokowy	9
4.2 Mikrokontroler RP2040 oraz RPI Pico	11
4.3 Ekspandery wejść i wyjść oraz magistrala I2C.....	15
4.4 Zasilanie sieciowe.....	18
4.5 Przetwornica BUCK oraz stabilizatory liniowe	18
4.6 Zasilanie komputera oraz ekranu	22
4.7 Obciążenia załączane przekaźnikami.....	24
4.8 Uniwersalne bloki wejściowo-wyjściowe.....	25
4.9 Wskazywanie stanów na poszczególnych wejściach/wyjściach.....	28
4.10 Kompletny schemat układu.....	29
4.11 Płytką drukowaną	34
4.12 Zestawienie materiałów	43
4.13 Montaż.....	47
4.14 Uruchamianie układu.....	53
4.15. Testy	58
5. Oprogramowanie układowe.....	65
6. Wnioski i dalsze prace	68
7. Bibliografia.....	69

1. Słownik akronimów i skrótów użytych w pracy

PCB — Printed Circuit Board — Obwód drukowany

RPI - Raspberry PI — komputer jednoplatformowy

DI - ang. Digital in — wejście cyfrowe

DO — ang. Digital Out — wyjście cyfrowe

AI - ang. Analog in — wejście analogowe

SMD/SMT — surface mounted device / technology — urządzenie / technologia montażu powierzchniowego

THD/THT — through hole device / technology — urządzenie / technologia montażu przewlekaneego

TI - Texas instruments — producent układów scalonych

DRC (ang. Design rule check) — sprawdzenie elektrycznej poprawności projektowej

PLC — programmable logic controller — sterownik programowany

2. Nomenklatura

Kicad — darmowe oprogramowanie do projektowania obwodów drukowanych

JLCPCB — firma zajmująca się produkcja PCB

PCB stackup — kolejne ułożenie warstw miedzi i izolatorów oraz ich grubości

Pliki Gerber — pliki fabrykacyjne — wektorowy zapis poszczególnych warstw

Pliki Drill — pliki fabrykacyjne — zaznaczenie miejsc wiercenia otworów

Bootloader — pierwszy program uruchamiany na urządzeniu — umożliwiający rozruch konkretnego programu

Footprint — ślad komponentu — pola lutownicze do którego komponent ma zostać przyłutowany

Pick&place — maszyna do automatycznego układania komponentów elektronicznych

3. Cel i założenia projektu

3.1 Rozwiązania dostępne na rynku

Kiedy potrzebą jest zbieranie stanów z czujników, sterowanie urządzeniami wykonawczymi, przetwarzanie informacji oraz wykonywanie algorytmów sterujących dostępnych na rynku jest szereg potencjalnych produktów do wymienionych celów. Począwszy od przemysłowych sterowników PLC- zarówno tych najprostszych jak przedstawiony na rysunku 3.1.1. Siemens LOGO! 8, poprzez średniej wielkości jak przedstawiony na rysunku 3.1.2 Siemens Simatic S7-1200, skończywszy na systemach złożonych systemach SCADA (ang. *Supervisory Control and data Acquisition*).



Rys. 3.1.1 Siemens LOGO! 8



Rys. 3.1.2 Siemens S7-1200

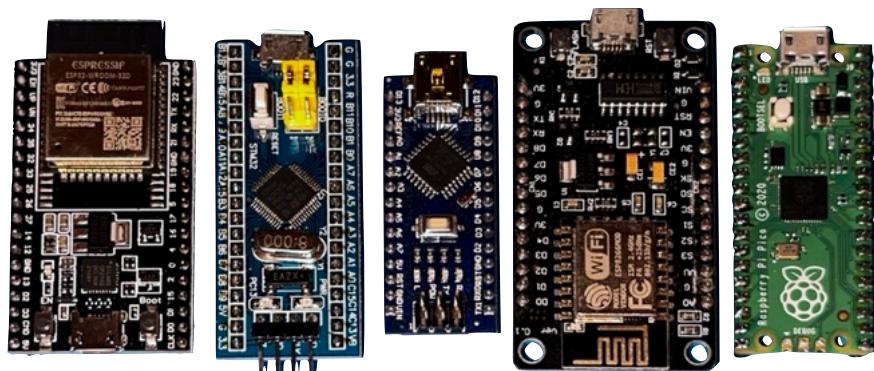
Źródło: opracowanie własne

Źródło: opracowanie własne

Niestety sterowniki PLC są przeważnie dosyć drogie, same zajmują dużo miejsca, a każde dodatkowe moduły wyjścia i wejścia bardzo szybko potęgują rozmiary. Zwykle programowanie odbywa się w IDE (ang. *Integrated Development Environment*) stworzonym przez producenta, w kilku udostępnionych przez niego językach. Jest to rozwiązanie bardzo pewne, trwałe oraz niezawodne. Czas potrzebny na stworzenie układu sterującego jest bardzo krótki, bowiem użytkownik otrzymuje praktycznie gotowe rozwiązanie, które trzeba tylko zaprogramować. Niestety zastosowanie do sterowania ekspresem sterownika PLC bardzo ogranicza późniejszy rozwój, brak możliwości przeniesienia kodu właściwie uniemożliwia realizację z wykorzystaniem własnego mikrokontrolera.

Z wyżej wymienionych powodów zdecydowanie odrzucono możliwość zbudowania układu sterującego opartego na gotowych sterownikach programowalnych.

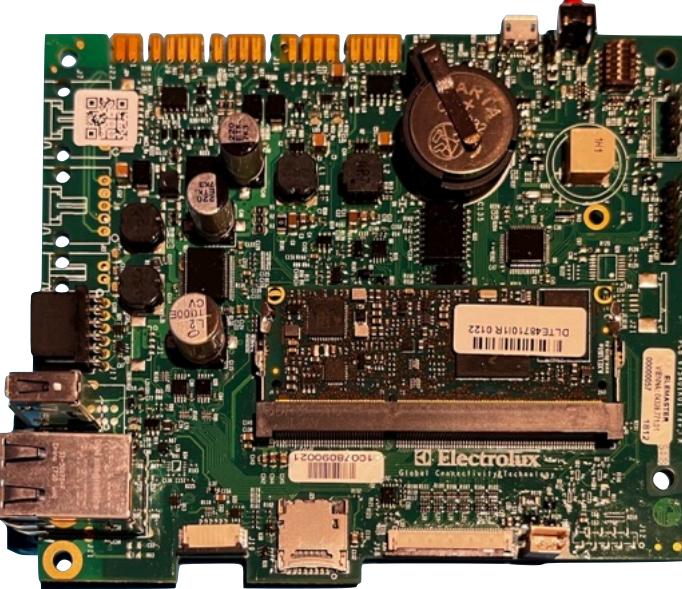
Kolejnym rozwiązaniem jest gotowy moduł zawierający mikrokontroler, pamięć oraz podstawowe komponenty wymagane do uruchomienia układu, jak wewnętrzny stabilizator napięcia, oscylator kwarcowy, itp. Na rys. 3.1.3 przedstawiono kilka przykładowych modułów dostępnych na rynku. Takie moduły są tanie, łatwo dostępne, umożliwiają przyłączenie do zaprojektowanej płytki, zajmują bardzo mało miejsca.



Rys. 3.1.3 Różnego rodzaju mikrokontrolery

Źródło: opracowanie własne

Obecnie najszerzej stosowanym przez producentów sprzętu AGD rozwiązaniami jest całkowicie własnoręcznie zaprojektowany, specjalizowany układ sterujący, jak na przykład ten z rys. 3.1.4. Pośród zalet takiego rozwiązania należy wymienić: całkowitą kontrolę nad tym jakie funkcjonalności zostaną zaimplementowane, płytki może mieć dowolny kształt, a przy tym niewielkie rozmiary. Niestety takie rozwiązanie jest dużo bardziej kosztowne. Kupno każdego komponentu oddziennie jest problematyczne mając na uwadze liczbę wymaganych komponentów. Ponadto, przy projektowaniu każdego bloku "od zera", znacznie rośnie ryzyko popełnienia błędu oraz wydłuża się czas potrzebny na zdiagnozowanie, naprawienie i uruchomienie układu.



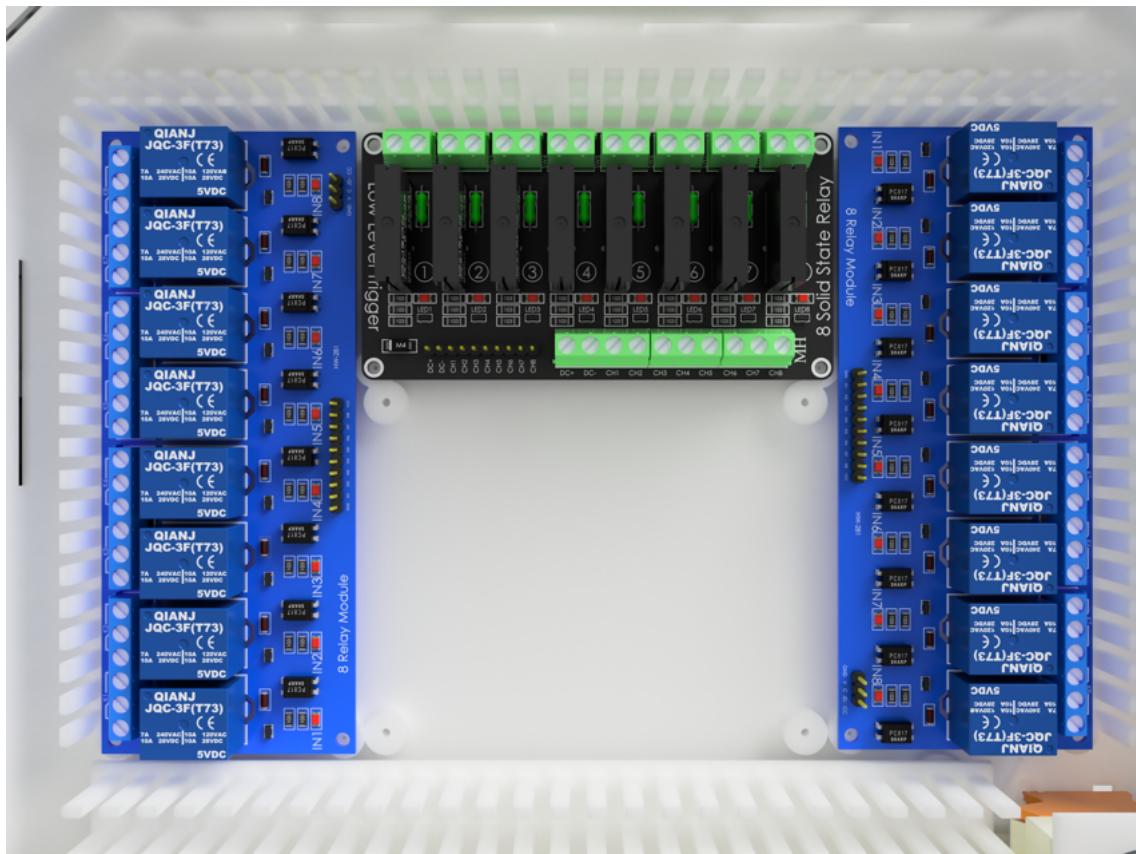
Rys. 3.1.4 Płytki sterująca z komputerem jednoplatformowym

Źródło: opracowanie własne

Mając na uwadze wszystkie powyższe rozważania — zdecydowano się na moduł zawierający mikrokontroler na własnoręcznie przygotowanej płytce głównej.

3.2 Założenia funkcjonalne, analiza potrzeb

Ze względu na dostępną przestrzeń w maszynie — ograniczoną samą obudową, a następnie modułami przekaźników i zasilaczem — na układ zaplanowano miejsce o wymiarach 80 mm na 100 mm, widoczne na rysunku 3.1.1 pomiędzy modułami przekaźników.



Rys. 3.1.1 Miejsce w ekspresie gdzie ma zmieścić się płytka sterująca

Źródło: opracowanie własne

Całe ekspres ma być zasilany z sieci elektrycznej 230 V. Ze względu na dużą liczbę urządzeń końcowych sterowanych z napięcia 12 V, zdecydowano się na zastosowanie zasilacza impulsowego zasilanego napięciem sieciowym. Dlatego też napięciem zasilającym układ również będzie 12 V. Wiele czujników lub mniejszych urządzeń wykonawczych jak serwomechanizmy, wymagają zasilania 5 V. Również główny komputer, czy też ekran dotykowy wymagają tego samego napięcia. Część czujników pracuje w logice 3,3 V, a część 5 V. Sam mikrokontroler zasilany jest napięciem 3,3 V. Uzyskanie napięć 3,3 V oraz 5 V wymaga stworzenia układów przetwarzających napięcie. Liczba potrzebnych sygnałów wejściowo-wyjściowych na ten moment przekracza 50. Mało który mikrokontroler posiada tak dużą liczbę wyprowadzeń. Z kolei modele wyposażone w odpowiednią liczbę wyprowadzeń mają jednocześnie wiele bloków do komunikacji z różnymi magistralami, przetworników cyfrowo-analogowych czy analogowo-cyfrowych, które w rozważanym zastosowaniu są zbędne, a podnoszą cenę sprzętu. Dlatego zdecydowano się na mały, tani i łatwo dostępny

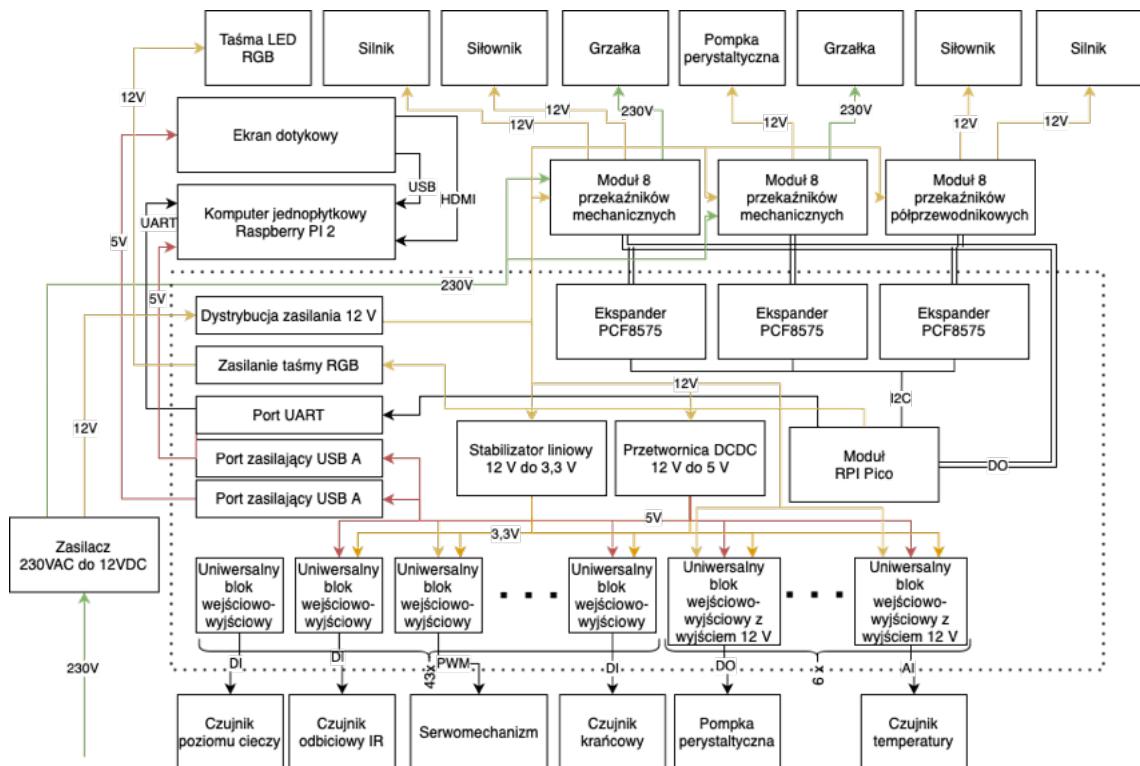
mikrokontroler RP2040 — posiadający zaledwie 28 portów GPIO oraz użycie trzech 16-bitowych ekspanderów komunikujących się poprzez magistralę I2C. Płytki ma umożliwiać dalszy rozwój maszyny, także musi zostać dodany zapas wejść i wyjść. Ma pomagać w rozwiązywaniu problemów z podłączonymi do niej modułami. Dlatego potrzebne są diody elektroluminescencyjne celem sygnalizowania wizualnie stanów poszczególnych wejść oraz wyjść. Układ ma się komunikować z głównym komputerem poprzez port UART i umożliwia sterowanie poszczególnymi elementami, także odczyt stanów czujników właśnie przez ten protokół. Wszystkie układy użyte muszą być dostępne w sprzedaży, biorąc pod uwagę obecną sytuację na rynku półprzewodników.

4. Środowisko sprzętowe

Jak napisano we wcześniejszych rozdziałach zdecydowano o realizacji układu w formie płytki PCB. Projekt schematu oraz płytki wykonano w darmowym programie Kicad. Ze względu na braki w asortymencie ważnym kryterium doboru komponentów jest możliwość zakupu poszczególnych części. Starano się wybierać popularne komponenty z dobrze napisaną dokumentacją, ze względu na późniejsze ułatwione uruchamianie układu oraz minimalizowanie błędów projektowych we wczesnym stadium projektu.

4.1 Schemat blokowy

Na rysunku 4.1.1. przedstawiony jest schemat blokowy projektowanego układu — za przerywaną linią — przykładowe urządzenia podłączone do poszczególnych wyjść.



Rys. 4.1.1. Schemat blokowy projektowanego układu

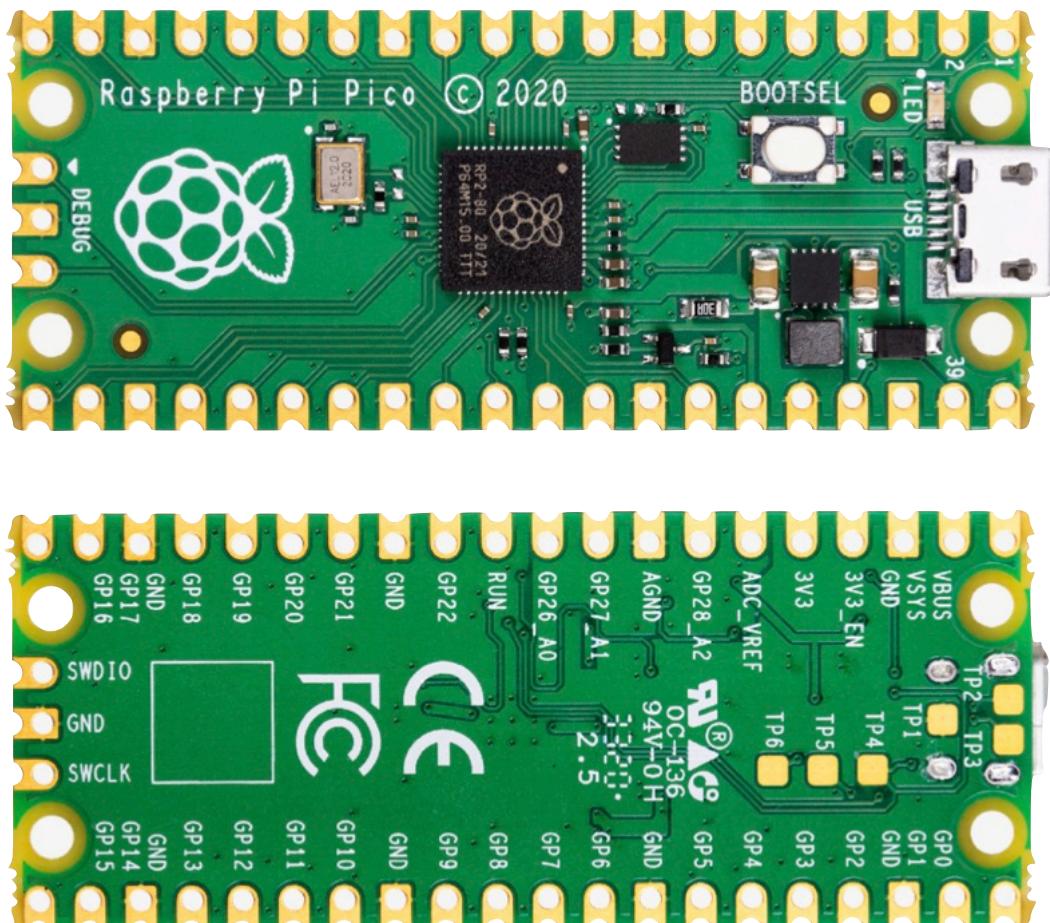
Źródło: opracowania własne

Płytkę zasilana jest napięciem stałym 12 V pochodzącym z zasilacza sieciowego. Pełni również funkcję dystrybucji zasilania — służy do zasilania modułów przekaźników.

W zależności od dostępności komponentów można użyć jednego z dwóch układów do obniżania napięcia: przetwornicy typu *buck* (ang. *Buck converter*) lub trzech stabilizatorów liniowych. Na wyjściu z nich otrzymuje się napięcie stałe 5 V. Jest ono wykorzystywane do urządzeń wejściowo-wyjściowych oraz — poprzez porty USB — głównego komputera w ekspresie (czyli Raspberry Pi 2) oraz do ekranu dotykowego. Do zasilania mikrokontrolera oraz do wyjść potrzebne jest również napięcie 3,3 V. Ze względu na relatywnie nieduże oczekiwane natężenie prądu odbiorników użyto stabilizatora liniowego o stałym napięciu wyjściowym 3,3 V. Jako główną jednostkę, przetwarzającą informacje z wejść i sterującą wyjściami, wykorzystano mikrokontroler RP 2040 w zestawie deweloperskim Raspberry Pi Pico. Część urządzeń, m.in. wyjścia PWM do serwomechanizmów korzysta bezpośrednio z wyprowadzeń mikrokontrolera RPI Pico. Ze względu na zbyt małą liczbę dostępnych pinów, konieczne było użycie ekspanderów wejściowo-wyjściowych. Wybrano takie, które pracują na magistrali I2C aby ograniczyć liczbę wykorzystanych pinów na mikrokontrolerze. Poszczególne urządzenia wyjściowe mogą pracować w logice 3,3 V lub 5 V odpowiednio, przy zasilaniu 3,3 V lub 5 V. Dlatego powstały bloki umożliwiające wybór pracy. Sześć bloków może również funkcjonować jako wyjścia pod napięciem 12 V o dużym natężeniu prądu (do urządzeń wykonawczych).

4.2 Mikrokontroler RP2040 oraz RPI Pico

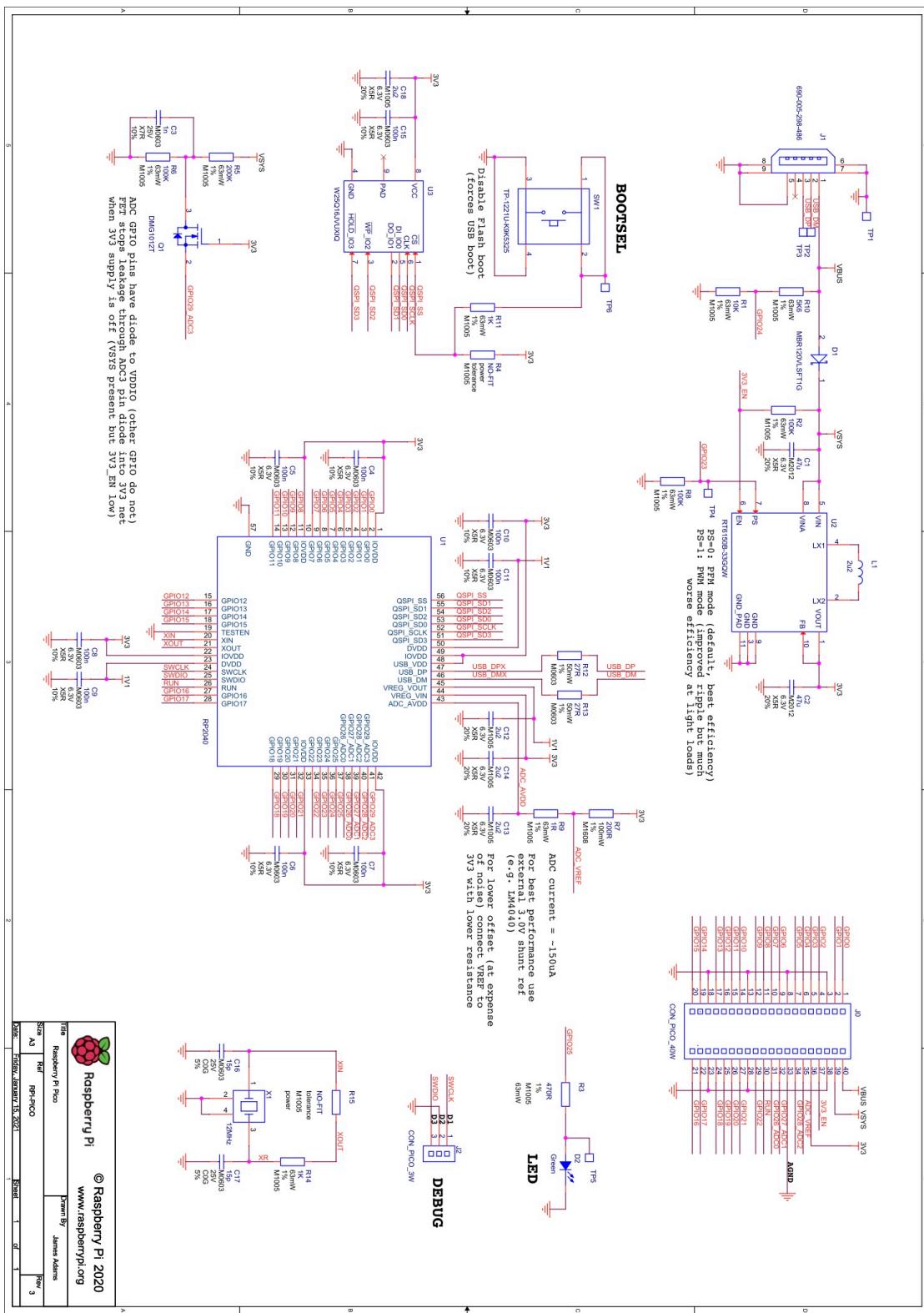
Ze względu na utrudnioną dostępność, zwłaszcza układów średniej i małej skali integracji wybrano mikrokontroler RP2040 z Raspberry PI Foundation. Ten konkretny model nie zdobył jak do tej pory popularności w przemyśle, co zapewne skutkuje jego wysoką dostępnością. W tej pracy wykorzystany został mikrokontroler w zestawie deweloperskim RPI Pico, pokazany na rysunku 4.2.1. Sam mikrokontroler jak i cały moduł mają bardzo obszerną i rzetelnie prowadzoną dokumentację.



Rys. 4.2.1 — Zestaw deweloperski Raspberry PI Pico

Źródło: <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>

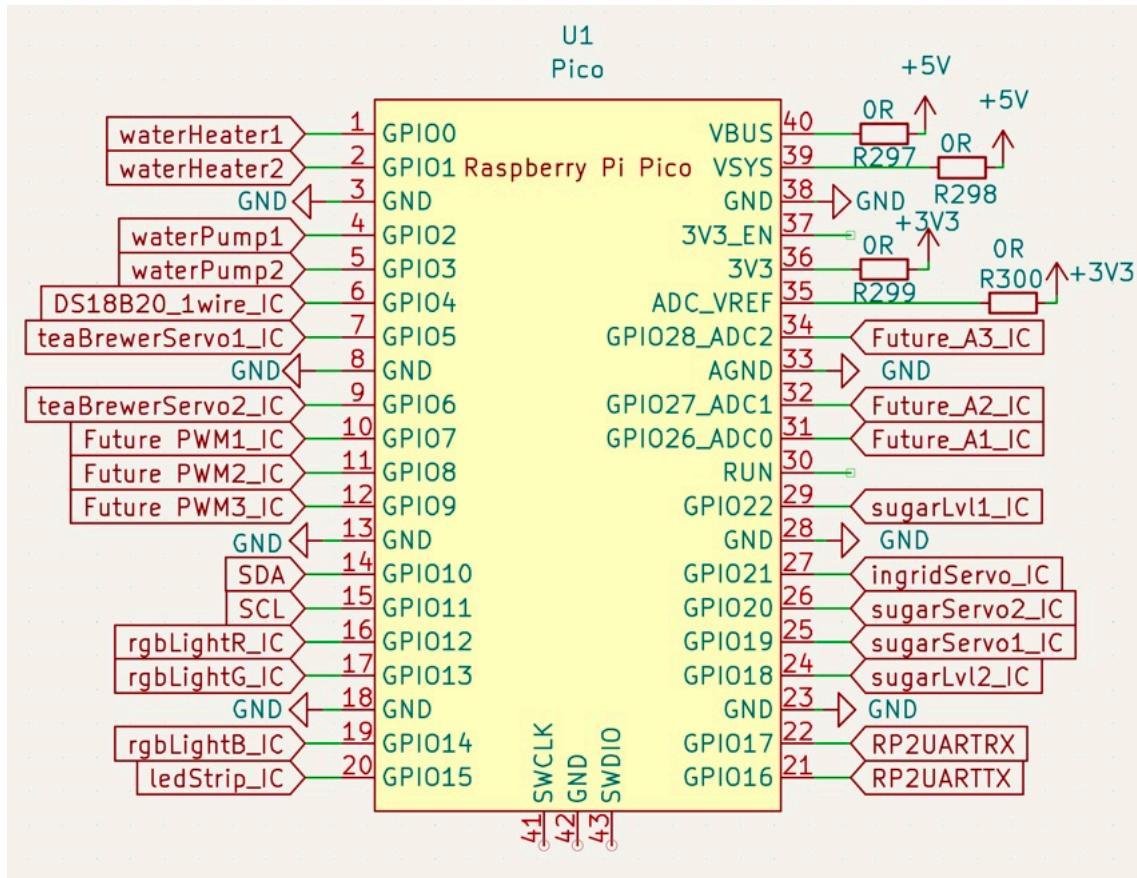
Moduł RPI Pico posiada przetwornicę zamieniającą napięcie 5 V (z zasilania USB) na 3,3 V, pod jakim działa mikrokontroler, pamięć QSPI, oscylator, układ zapewniający odpowiednią filtrację napięcia oraz przycisk do wyboru *bootloadera* — widoczne na rys. 4.2.2. Ze względu na mały *footprint* układu i zastosowaną obudowę QFN-64, która jest dosyć trudna w poprawnym przylutowaniu, zdecydowano się na zastosowanie gotowego modułu.



Rys. 4.2.2 Schemat modułu Raspberry PI Pico

Źródło: <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>

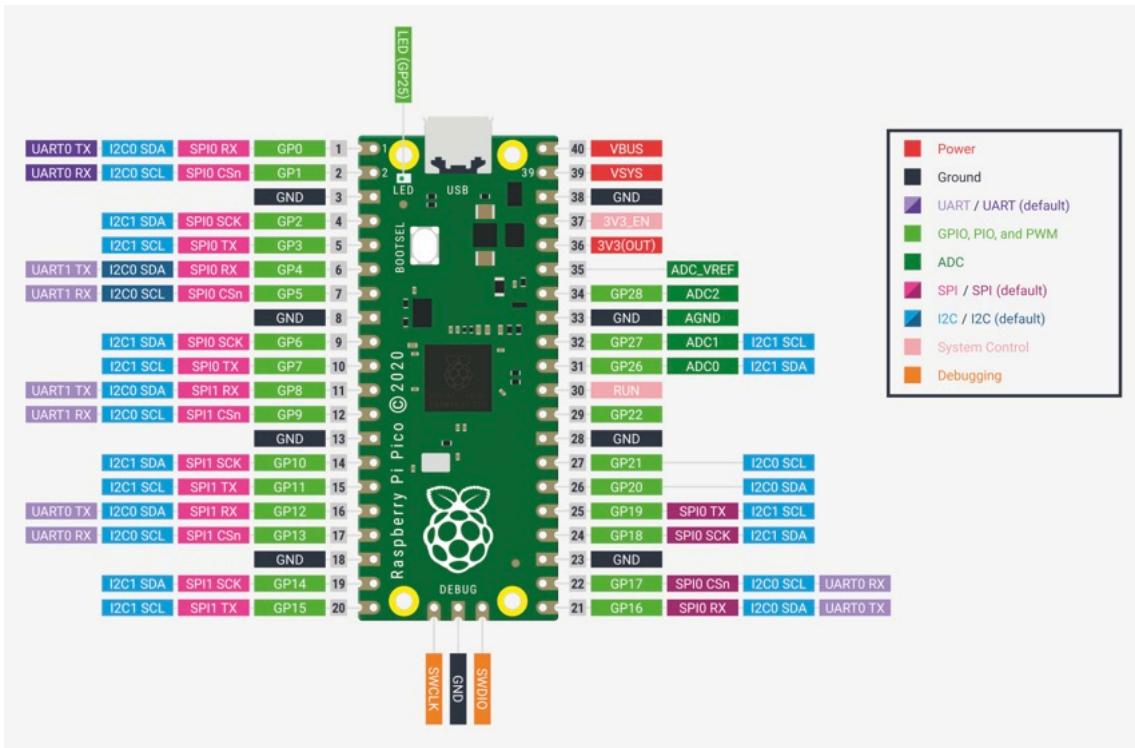
Sygnały z RPI Pico wyprowadzone są na dwie listwy typu *goldpin*, dzięki czemu możliwe jest stworzenie połączenia rozłączalnego i w przypadku uszkodzenia modułu możliwa jest jego bezproblemowa wymiana. Ma to również drugą zaletę — pod modułem jest dużo miejsca, które można wykorzystać do rozlokowania pozostałych komponentów.



Rys. 4.2.3 Sygnały podłączone do modułu RPI Pico

Źródło: opracowania własne

Sygnały podłączone do modułu RPI Pico zostały pokazane na rys. 4.2.3. Zaczynając od zasilania. VBUS oraz VSYS są podłączone do linii 5 V, 3V3 do linii 3,3 V w obu przypadkach poprzez rezystory 0Ω (które w celu rozwiązywania problemów można wylutować i rozłączyć połączenia). Sygnał 3V3_EN, aktywujący układ zasilający modułu RPI Pico jest podcięgnięty do linii 3,3 V już w module (rys. 4.2.2), także nie jest wymagane podłączanie go do żadnego z sygnałów. Podobnie sygnał RUN, który zgodnie z dokumentacją [11] jest podcięgnięty do linii 3,3 V już wewnątrz układu.



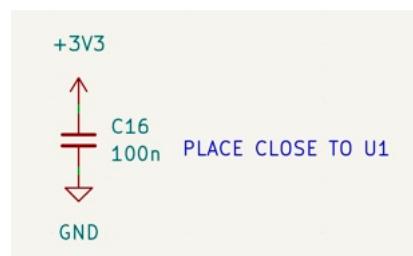
Rys. 4.2.4 Wyprowadzenia RPI Pico oraz role, jakie mogą pełnić

Źródło: <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>

Sygnały na schemacie — SDA oraz SCL, podłączone odpowiednio do GPIO10, oraz GPIO12 pełnią rolę magistrali protokołu I2C. Podobnie RP2UARTRX i RP2UARTTX, podłączone do GPIO16 oraz GPIO17, odpowiadają za komunikację poprzez port UART z komputerem RPI 2. Wybrano te konkretne piny ze względu na możliwość wyprowadzenia wewnętrznych sygnałów z bloków do komunikacji z I2C oraz UART w mikroprocesorze RP2040 na nie (zgodnie z rysunkiem 4.2.4).

Pozostałe sygnały to wejścia lub wyjścia cyfrowe pracujące w logice 3,3 V oraz trzy wejścia analogowe GPIO26_ADC0, GPIO27_ADC1 i GPIO28_ADC2.

Dodano również kondensator 100nF, umieszczony na płytce blisko modułu RPI Pico, pokazany na rys. 4.2.5. Ma on na celu filtrować napięcie zasilające moduł.

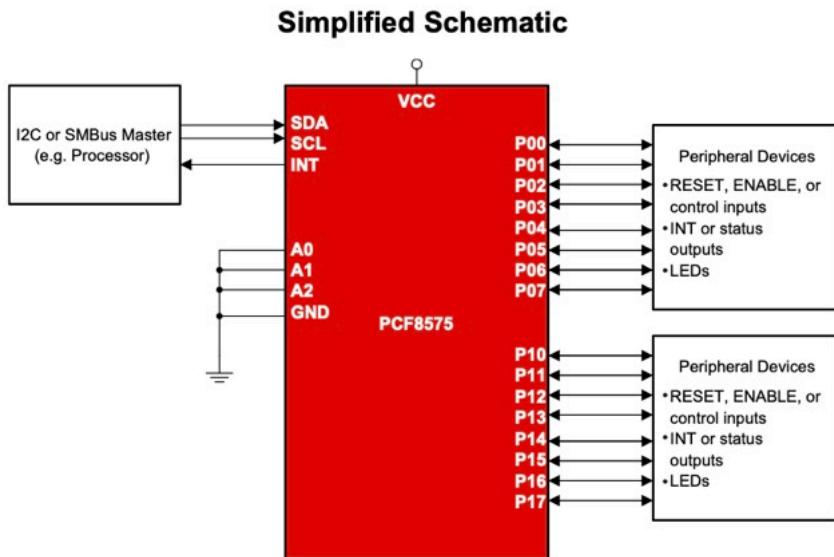


Rys. 4.2.5 Kondensator filtrujący napięcie

Źródło: opracowania własne

4.3 Ekspandery wejść i wyjść oraz magistrala I2C

Z względu na zbyt małą liczbę pinów mikrokontrolera RP2040, konieczne było zastosowanie ekspanderów wejść i wyjść. W tym celu zastosowano trzy układy PCF8575. Są to 16-bitowe ekspandery, współpracujące z magistralą I2C, posiadające wyjście przerwań.. Uproszczony schemat wejść i wyjść takiego układu pokazano na rys. 4.3.1.



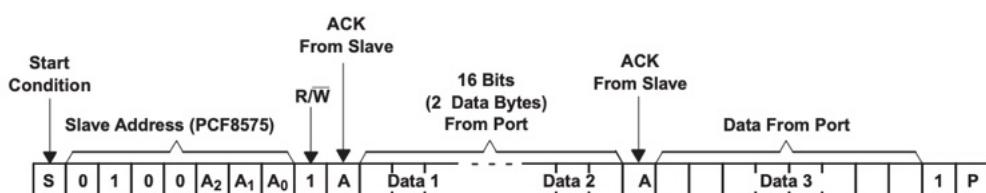
Rys 4.3.1 Wyjścia układu PCF8575

Źródło: https://www.ti.com/lit/ds/symlink/pcf8575.pdf?ts=1673710446867&ref_url=https%3A%2F%2Fwww.google.com%2F

SDA (ang. *Serial data line*) to linia danych szeregowych. Wraz z linią zegarową SCL (ang. *Serial Clock Line*), odpowiada ona za komunikację na magistrali I2C.
Wyjście INT (ang. *interrupt*) służy do zgłaszania przerwań, gdy zmieni się stan na którymś z wejść, jednak w tej pracy nie jest wykorzystane.

Sygnały A0, A1, A2 (ang. *adress*) odpowiadają za zaadresowanie układu na magistrali I2C. Przykładowo, podłączając A0 do lini 3,3 V, A1 a A2 do GND otrzymamy adres 0x21 — zgodnie z schematem adresacjii na rys 4.3.2.

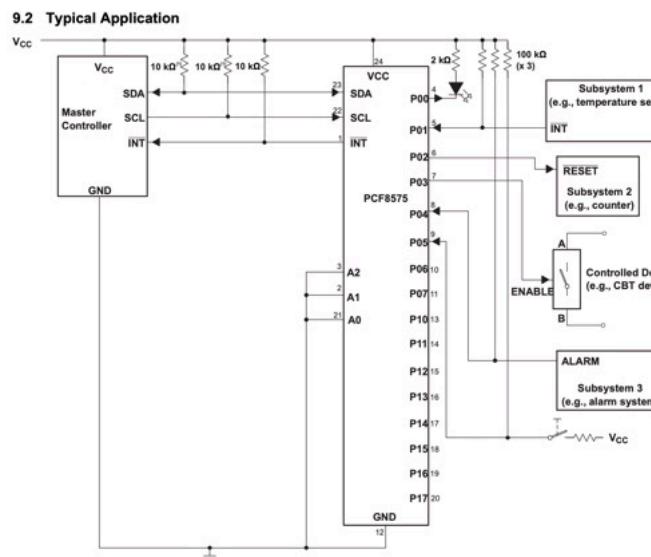
Rys. 4.3.2 Adresacja na linii I2C



Źródło: https://www.ti.com/lit/ds/symlink/pcf8575.pdf?ts=1674129274894&ref_url=https%253A%252F%252Fwww.google.com%252F

Linia VCC (ang. Voltage Common Collector) służy do zasilania ekspandera dodatnim potencjałem. W przypadku tego układu jest to 3,3 V.

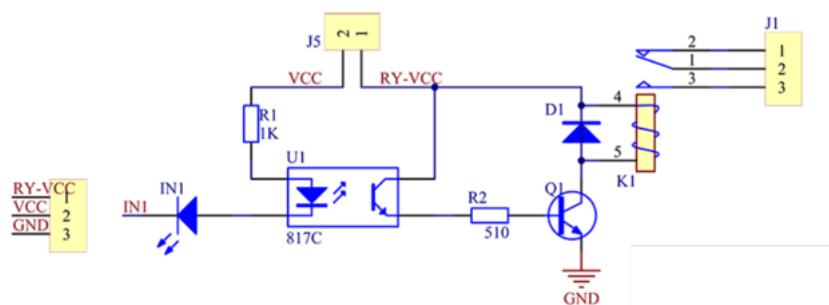
Linie P00-P17 mogą być używane do podłączania peryferiów, urządzeń wejściowych lub wyjściowych w zależności od konfiguracji danego pinu. W przypadku użycia pinu jako wejścia, zgodnie z dokumentacją na rys. 4.3.3, należy go podciągnąć do linii VCC za pomocą rezystora $100\text{k}\Omega$.



Rys. 4.3.3 Aplikacja układu PCF8575

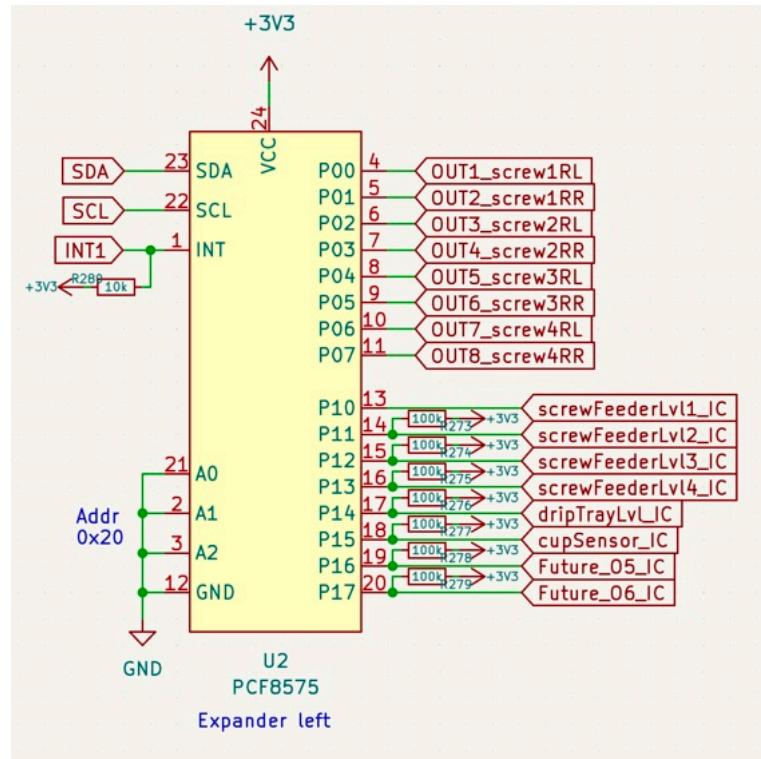
Źródło: www.ti.com

Wymaga się tego ze względu na charakter wyjścia (z otwartym drenem). Dlatego sygnały, które będą podłączone do wejść czujników zostały podciągnięte do potencjału 3,3 V. Natomiast dla wyjść przekaźnikowych nie było to wymagane ze względu na obecną optoizolację — aby układ został załączony trzeba ściągnąć go do masy (rys.4.3.4).



Rys. 4.3.4 Schemat jednego z przekaźników

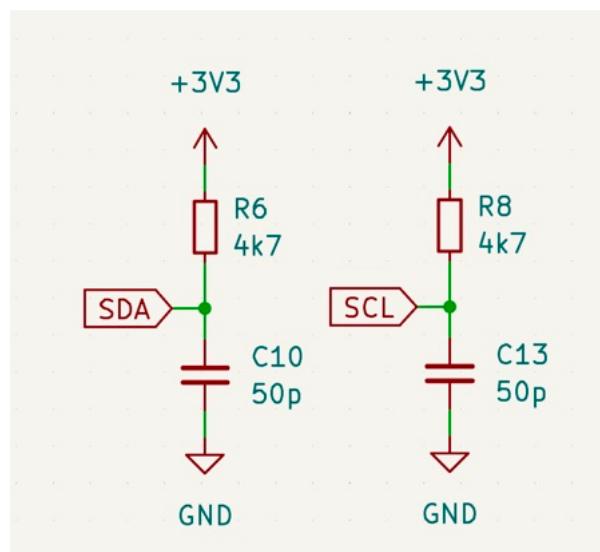
Źródło: www.wiki.sunfounder.cc



Rys. 4.3.5 Schemat podłączenia jednego z ekspanderów PCF8575

Źródło: opracowanie własne

Zgodnie z wymaganiami standardu I2C [12], linie SDA i SCL należy podpiąć przez rezystancje do zasilania oraz dołożyć kondensator o bardzo małej pojemności między liniami a masą układu — tak jak pokazano na rys. 4.3.6.



Rys. 4.3.6 Podciąganie linii I2C do 3,3 V

Źródło: opracowanie własne

4.4 Zasilanie sieciowe

Z przyczyn bezpieczeństwa, układ zasilany jest napięciem stałym 12 V z zewnętrznego zasilacza, zamiast zasilania sieciowego 230V. W układzie wykorzystywane są dodatkowo napięcia 3,3 V oraz 5 V. Aby je uzyskać wykorzystano układ obniżający napięcie. Powinny one cechować się niewielkimi rozmiarami, aby mogły zmieścić się na płytce PCB na ograniczonym obszarze pozostałym po umieszczeniu pozostałych komponentów.

4.5 Przetwornica BUCK oraz stabilizatory liniowe

Ze względu na wymagany duży spadek napięcia zasilania — z 12 V na 5 V, wraz z poborem mocy sięgającym 15W, zdecydowanie najlepszą opcją jest przetwornica typu *buck*. Początkowo wybór padł na bardzo popularny układ firmy Texas Instruments — LM2575. Niestety jego wydajność przy pracy ciągłej sięga zaledwie 5W. Ostatecznie użyto układu XL4015 chińskiej firmy XLSEMI (Shanghai Xinlong Semiconductor Technology Co., Ltd.). Łatwy do kupienia, w tej samej obudowie co LM2575, a mimo tego oferuje aż 5A na wyjściu — co w tej aplikacji pozwoli na uzyskanie teoretycznie 25W mocy (efektywne maksymalne obciążenie będzie możliwe do określenia po zmontowaniu układu i przetestowaniu sprawności zastosowanego padu odprowadzającego ciepło). W odróżnieniu od produktu firmy TI, który w obudowie pad termiczny ma wyprowadzony jako potencjał masy, XL4015 ma w tym miejscu pin SW — łączący się z cewką. Oznacza to że należy stworzyć pad, który elektrycznie łączy tylko dwa sygnały, ale musi mieć całkiem dużą powierzchnię. Trzeba więc poświęcić trochę użytecznej przestrzeni płytki na pad termiczny, zamiast (jak można by zrobić w układzie LM2575) odprowadzić ciepło w płaszczyznę masy, która rozciąga się na całą powierzchnię płytki. Bazując na module z układem XL4015 (rys. 4.5.1., rys. 4.5.2) starano się odtworzyć pad termiczny o powierzchni nie mniejszej niż obecnej w tym wykonaniu. Opracowany przez autora pad termiczny pokazano na rysunku 4.5.3- 4.5.4, a kompletny schemat przetwornicy na rysunku 4.5.5.



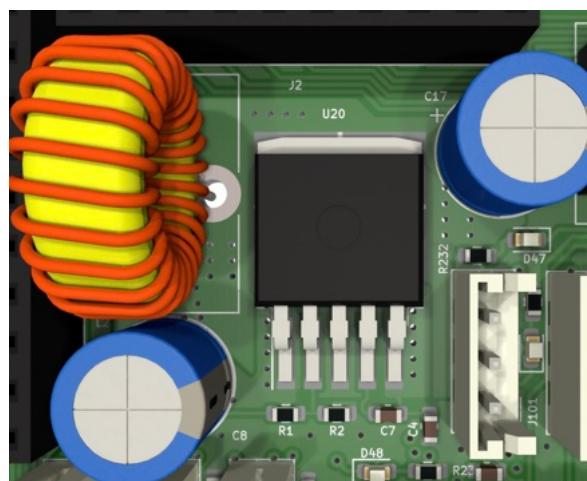
Rys. 4.5.1 Moduł z układem XL4015 — dół

Źródło: opracowanie własne



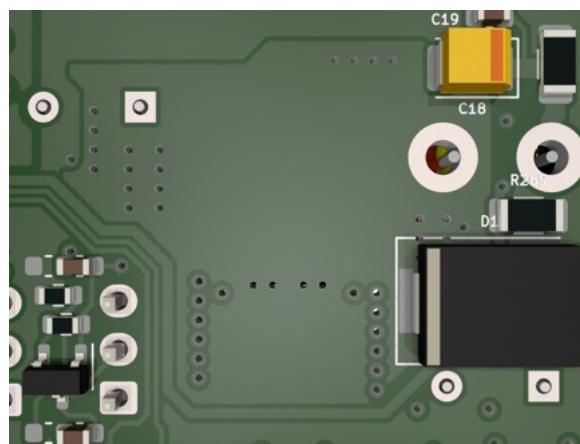
Rys. 4.5.2 Moduł z układem XL4015 – góra

Źródło: opracowanie własne



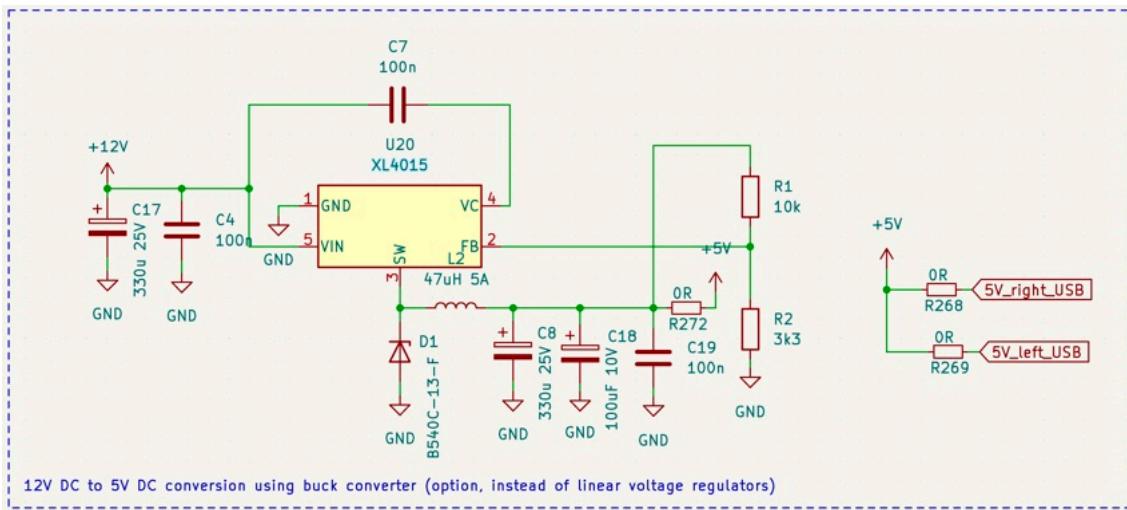
Rys. 4.5.3 Pad termiczny- góra

Źródło: opracowanie własne



Rys. 4.5.4 Pad termiczny- dół

Źródło: opracowanie własne



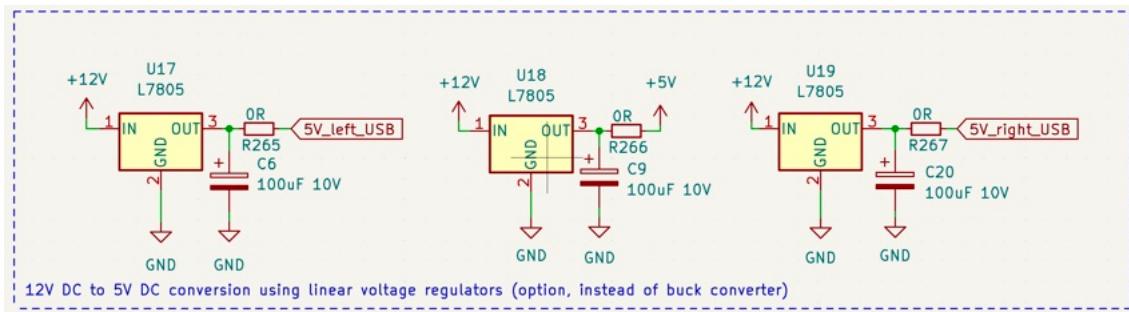
Rys. 4.5.5 Schemat przetwornicy opartej o układ XL4015

Źródło: opracowanie własne

Działanie przetwornicy typu *buck* polega na kontrolowaniu częstotliwości przełączania tranzystora MOSFET (znajdującego się wewnątrz układu XL4015). Istnieją dwa zasadnicze stany tego typu przetwornicy. [5] Gdy tranzystor wewnątrz układu XL4015 jest załączony, przez cewkę L2 zaczyna płynąć prąd, część napięcia odkłada się na niej, a ona sama magazynuje energię w polu magnetycznym. Następuje również ładowanie kondensatorów C8, C18 oraz C19. Gdy napięcie wyjściowe przekroczy zadaną wartość — przekazywaną w pętli sprzężenia zwrotnego z rezystorów R1 oraz R2 na wejście FB (ang. *feedback*), następuje przełączenie stanu pracy, tj. rozłączenie tranzystora MOSFET. W tym momencie spadające natężenie prądu spowoduje wyindukowanie się napięcia na cewce i w konsekwencji przepływ prądu w oczku: cewka L2 — obciążenie wyjściowe — dioda D1. Kondensatory zapewniają stabilizację napięcia wyjściowego. Gdy ono spadnie, nastąpi ponowne przejście do stanu pierwszego (ang. *Ripple voltage*).

Rezystor R272 pełni rolę zworki, umożliwiając odłączenie układu od linii zasilania 5 V, rozprowadzonej po całej płytce. Dzięki takiej konfiguracji w przypadku nieprawidłowego działania przetwornicy istnieje możliwość jej odłączenia od całego układu dla ochrony. Co więcej, możliwy jest wybór zasilania pomiędzy przetwornicą, a stabilizatorami liniowymi.

Drugą opcję konwersji napięcia z 12 V do 5 V są trzy stabilizatory liniowe LM7805 w obudowach TO-220. Dodano tę alternatywę na wypadek braku układu XL4015 w sprzedaży. Ze względu na pobór mocy — do 15W, gdy pojedynczy stabilizator może zapewnić jedynie 5W — wybrano trzy takie same stabilizatory, przy czym każdy zasila inny fragment układu. Nie podłączono ich w konfiguracji równoległej, ponieważ może to skutkować niestabilnością pracy, a nawet uszkodzenia układu. Brak równowagi termicznej pomiędzy stabilizatorami, czy różnica napięć wyjściowych może przysporzyć wiele trudności. Dlatego użyto jednego do zasilania płytki, drugiego do komputera, a trzeciego do ekranu dotykowego.



Rys. 4.5.6 Schemat zasilania w oparciu o układ LM7805

Źródło: opracowanie własne

Każdy stabilizator ma niewielki kondensator na wyjściu, którego zadaniem jest niwelowanie tętnień oraz rezistor, który służy jako zworka. Należy ją wylutować w razie użycia przetwornicy XL4015 zamiast stabilizatorów liniowych.

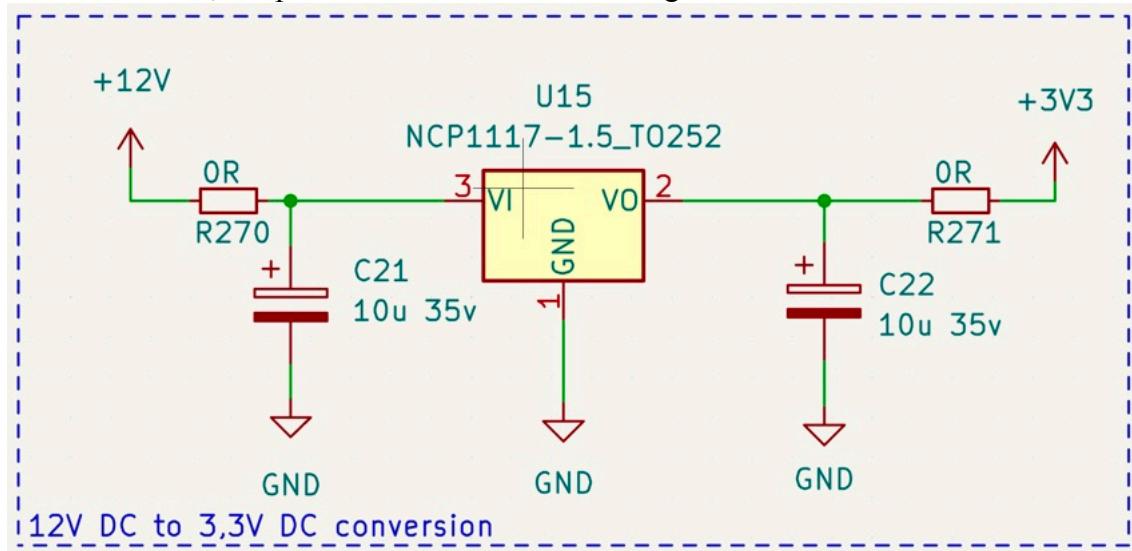
W zależności od potrzeb — użycia układu XL4015 lub stabilizatorów LM7805 — układ należy skonfigurować według tabeli 4.5.1.

Tab. 4.5.1 Wybór układu zasilającego

Elementy:	Przetwornica buck	Stabilizatory liniowe
R272	obecny	wylutowany
R268	obecny	wylutowany
R269	obecny	wylutowany
R265	wylutowany	obecny
R266	wylutowany	obecny
R267	wylutowany	obecny

Źródło: opracowanie własne

Zasilanie linii 3,3 V pochodzi stabilizatora liniowego NCP1117 w obudowie TO-252-3.



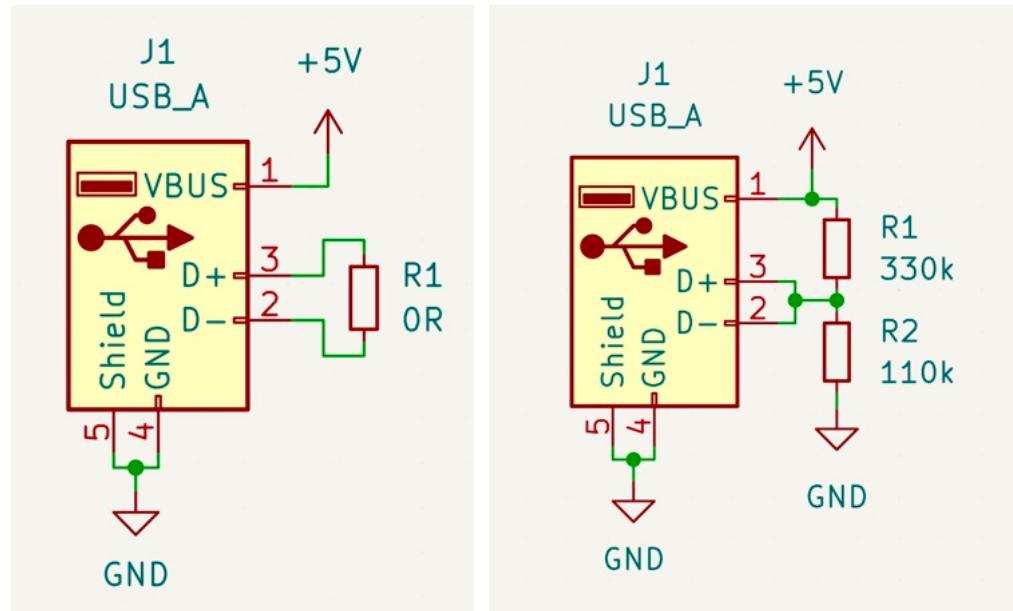
Rys. 4.5.7 Schemat zasilania 3,3 V w oparciu o układ NCP1117

Źródło: opracowanie własne

Rezystory R270 oraz R271 służą do odłączania układu od linii zasilania 12 V oraz 3,3 V. Są one używane podczas pierwszego uruchamiania układu po zlutowaniu. Kondensatory C21 oraz C22 mają za zadanie filtrację napięć.

4.6 Zasilanie komputera oraz ekranu

Zarówno komputer RPI 2, jak i ekran dotykowy wymagają zasilania 5 V z portu USB. RPI 2, bez dodatkowych urządzeń zasilanych z jego portów USB, nie pobiera więcej niż 5W. Jednak dla poprawnego działania na port USB zasilający nie wystarczy podać 5 V, ale również informację o wydajności prądowej zasilacza. Informacja o tym że urządzeniem jest ładowarka w podstawowej wersji podawana jest poprzez zwarcie linii danych D+ oraz D- portu USB (rys. 4.6.1). Jednak obecnie większość ładowarek podaje informacje o swojej wydajności prądowej, co jest wymagane do poprawnego działania RPI 2. Dlatego — poza zwarciem — oba sygnały zostają dołączone do potencjału, który ustalany jest przez dzielnik rezystancyjny. Firma Samsung ustaliła 1,2V jako napięcie wyjściowe we własnych dzielnicach we własnych produktach (rys. 4.6.2). Pomimo tego, że wartość nie jest objęta żadnym standardem, została przyjęta przez większość producentów ładowarek USB.



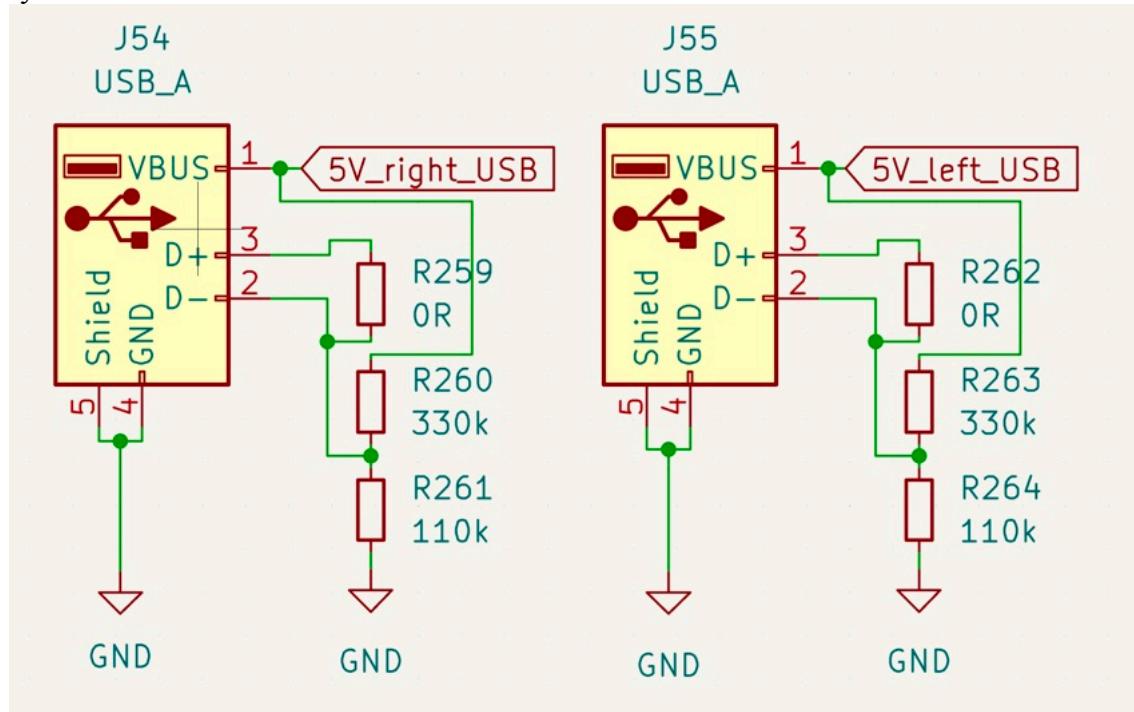
Rys. 4.6.1 Podstawowa konfiguracja

Źródło: opracowanie własne

Rys. 4.6.1 Konfiguracja w realizacji firmy Samsung

Źródło: opracowanie własne

Dlatego do zasilania ekranu i komputera zdecydowano na połączenie obu rozwiązań — zostawienie miejsca na płytce na ewentualne eksperymenty i testy, co przedstawiono na rys. 4.6.3.

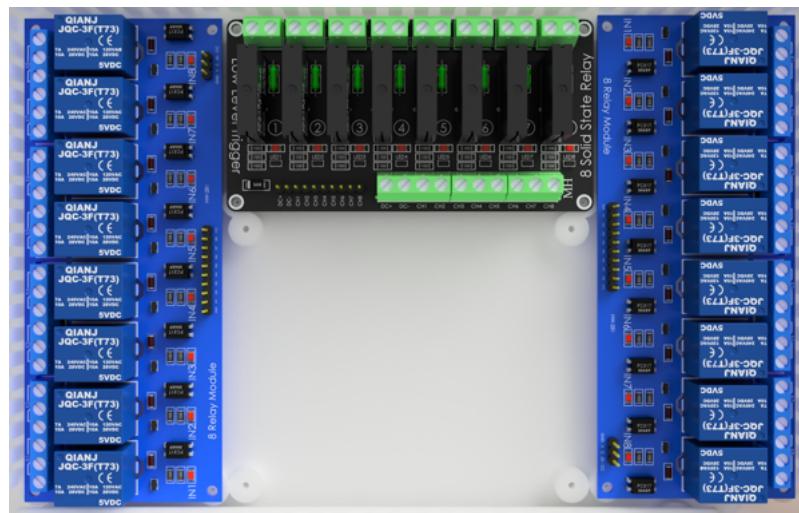


Rys. 4.6.3 Schemat wyjść USB

Źródło: opracowanie własne

4.7 Obciążenia załączane przekaźnikami

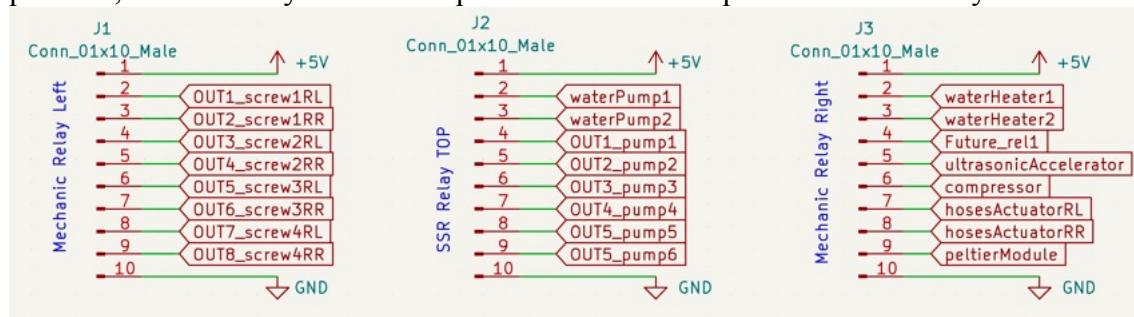
Większość urządzeń wyjściowych wymaga dla poprawnego działania prądu o znacznym natężeniu albo wysokim napięciu. Ze względu na zwiększone ryzyko związane z przełączaniem takich obciążzeń, zwłaszcza gdy układ ma charakter rozwojowy i urządzenia podłączane mogą powodować ekstremalne stany, takie jak przepięcia i zwarcia, zdecydowano o użyciu gotowych modułów przekaźników, które w przypadku uszkodzenia można bez problemu i tanio wymienić. Na rysunku 4.7.1 pokazano moduły przekaźników.



Rys. 4.7.1 Moduły przekaźników

Źródło: opracowanie własne

Same moduły mogą być podłączone bezpośrednio do mikrokontrolera, gdy — poza przekaźnikami — mają układy optoisolacji, diody zabezpieczające przy odłączaniu oraz tranzystory, które sterują prądem wymaganym do zasilania cewek w przekaźnikach. Dlatego po podłączeniu zasilania 5 V do całego modułu, sterowanie poszczególnymi przekaźnikami może odbywać się napięciowo. Co więcej wystarczającym napięciem jest 3,3 V, dzięki czemu nie są wymagane bufore napięciowe na wyjściach RPI Pico. Sygnały wyprowadzone są na listwę kołkową z rozstawem pinów 2,54 mm. Na rysunku 4.7.2 pokazano schemat wprowadzeń na listwy kołkowe.



Rys. 4.7.2 Schemat wyprowadzeń na listwy kołkowe

Źródło: opracowanie własne

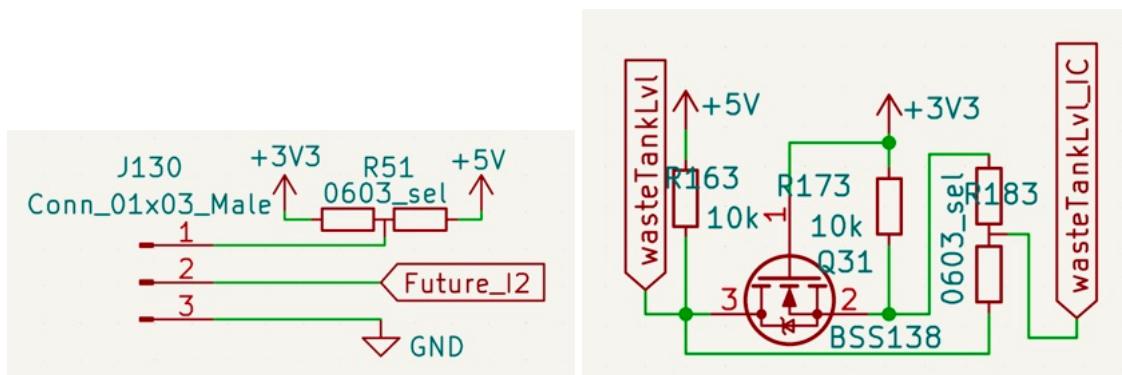
4.8 Uniwersalne bloki wejściowo-wyjściowe

Zważając na rozwojowy charakter układu, fakt że urządzenia podłączone do ekspresu do herbaty nie są jeszcze znane w swojej finalnej wersji, wymagane było wykonanie układu w taki sposób, aby łatwo można zmieniać tryby pracy peryferiów jako wejście lub wyjście, zasilanie napięciem 3,3 V lub 5 V. Dodatkowo potrzebne są napięcia zasilające peryfery 3,3 V lub 5 V. Aby to umożliwić zastosowano wiele rezystorów 0 Ω jako zworki. We wszystkich przypadkach są to rezystory w obudowie 0603, ale na 3 padach — jak na rys 4.8.1 — co pozwala na wylutowanie ich w jednej z dwóch pozycji.



Rys. 4.8.1 Zworka z wyborem

Źródło: opracowanie własne



Rys. 4.8.2 Schemat złącza do podłączania peryferiów

Źródło: opracowanie własne

Rys. 4.8.3 Schemat konwertera poziomów logicznych

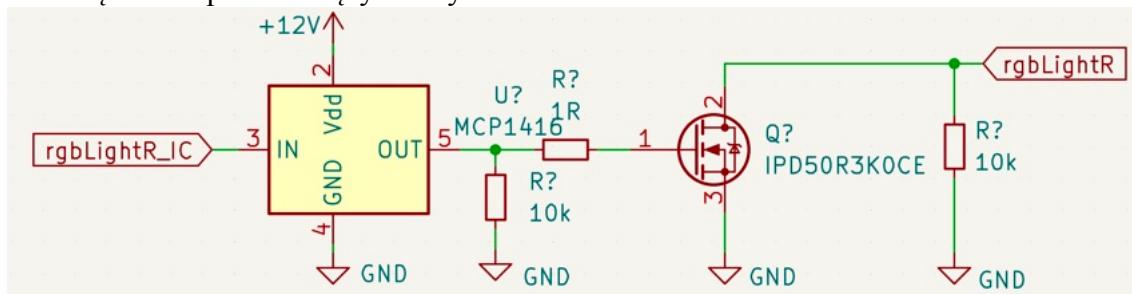
Źródło: opracowanie własne

Na rysunku 4.8.2 przedstawione jest złącze wyjściowe wraz ze zworką z możliwością wyboru napięcia wyjściowego do zasilania czujników lub urządzeń peryferyjnych.

Na rysunku 4.8.3 przedstawiono schemat bufora dwukierunkowego, umożliwiającego translację napięciową. Aby na urządzenie peryferyjne pracowało w logice napięciowej 3,3 V, rezystor R183 należy ustawić w pozycji dolnej oraz uruchomić wewnętrzny rezystor pull-up do napięcia 3,3 V w układzie RPI Pico (optymalnie byłoby wylutować R163, ale przeprowadzone testy pokazały że nie jest to konieczne). Natomiast jeżeli R183 będzie w pozycji górnej, napięcie 3,3 V z układu RPI Pico, w tym przypadku sygnał wasteTankLvl_IC — trafi na tranzystor MOSFET, który przejdzie w stan nieprzewodzenia, a napięcie przez rezystor R163 zostanie podciągnięte do napięcia 5 V.

Z kolei gdy pin mikrokontrolera pełni rolę wejścia, a zworka R183 przyjmie pozycję górną, to w przypadku podania napięcia 5 V na linię wasteTankLvl tranzystor przejdzie w stan nieprzewodzenia i na wyjściu sygnał zostanie podcięgnięty do napięcia 3,3 V przez rezystor R173. Jeżeli na wasteTankLvl zostanie podane 0V — tranzystor będzie przewodził i na pinie wasteTankLvl_IC również pojawi się wartość zbliżona do 0V (nie idealnie zero, ale logicznie wciąż interpretowana jako stan niski).

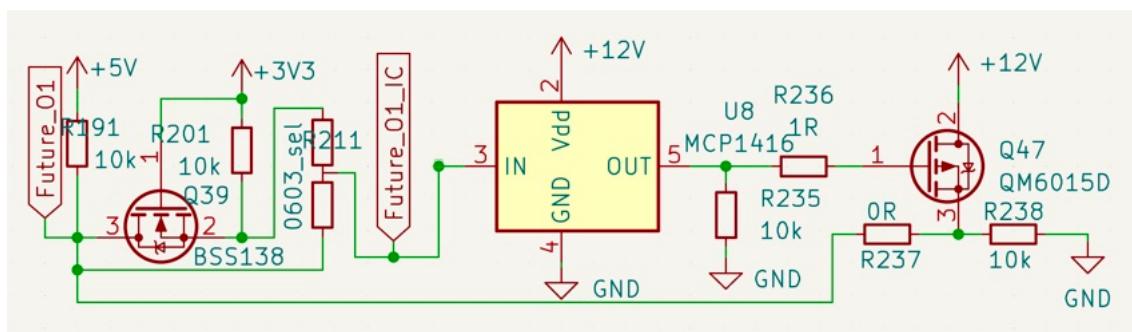
Na rys. 4.8.4. przedstawiono schemat pojedynczego kanału wyjściowego do taśmy RGB. Na wejście układu MCP1416 zostaje podany sygnał logiczny do załączania i wyłączania tranzystora — praktycznie czysto napięciowy. Układ MCP1416 odpowiada za wysterowanie większego tranzystora MOSFET jakim jest IPD50R3K0CE — dostarczenie odpowiednio dużego prądu, by w jak najkrótszym czasie naładować pojemność bramki tranzystora i jak najszybciej przełączyć tranzystor w stan pełnego przewodzenia oraz w drugą stronę — jak najszybciej rozładować pojemność i w pełni zamknąć kanał przewodzący tranzystora.



Rys. 4.8.4 Wyjście pojedynczego kanału taśmy RGB

Źródło: opracowania własne

W układzie jest również sześć wyjść, które mogą funkcjonować jako dwukierunkowe translatory logiczne oraz jako wyjścia wysokoprądowe 12 V. Jest to połączenie wcześniejszych omawianych układów (jedyną zmianą jest zamiana tranzystora na MOSFET z kanałem typu P, także załączanie będzie się tu odbywało w odwróconej logice napięciowej). Układ przedstawiono na rys. 4.8.5.



Rys. 4.8.5 Wyjście pojedynczego kanału uniwersalnego bloku wejściowo-wyjściowego z dodatkowym wyjściem 12 V

Źródło: opracowania własne

W tabeli 4.8.1. przedstawiono pozycje elementów w zależności od wymaganej funkcji danego bloku.

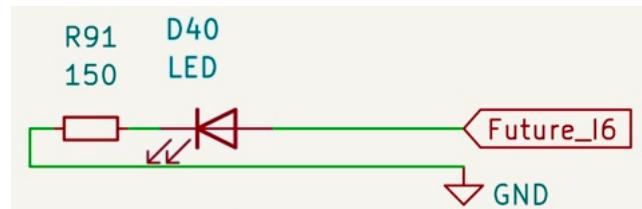
Tab. 4.8.1 Ustawienia układu w zależności od wymaganej funkcji

Elementy:	Translator 3,3 – 5 V	Brak translacji	Wyjście 12 V
R211	Pozycja górnna	Pozycja dolna	Wylutowany
R237	Wylutowany	Wylutowany	Obecny

Źródło: opracowania własne

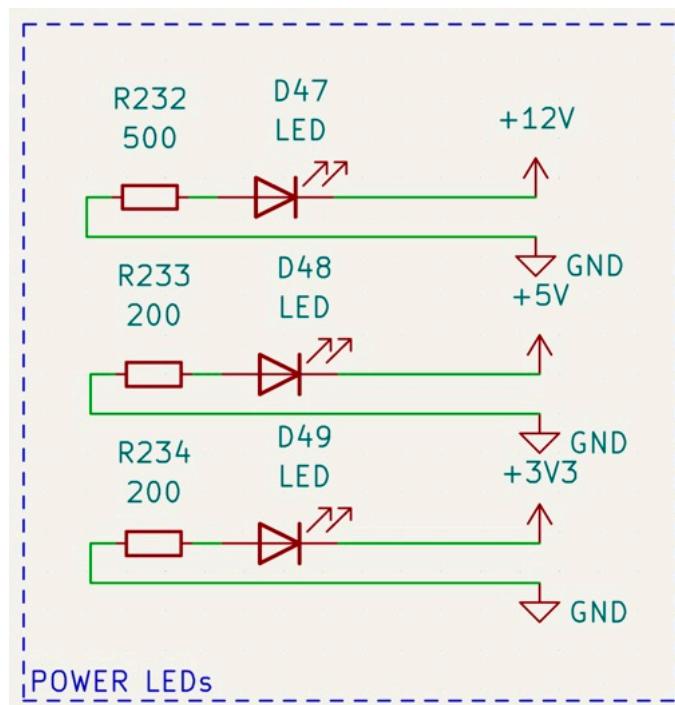
4.9 Wskazywanie stanów na poszczególnych wejściach/wyjściach

Diody elektroluminescencyjne LED (ang. *Light emitting diode*) są zdecydowanie najprostszym sposobem na wskazywanie stanów logicznych na poszczególnych wyjściach. Aby ułatwić późniejszą lokalizację usterek (na płytce czy w urządzeniu wykonawczym lub czujniku) zdecydowano o dodaniu diod do wszystkich wyjść sygnałowych na płytce (rys. 4.9.1), a także tych, które wskazują na obecność napięć 3,3 V, 5 V oraz 12 V (rys. 4.9.2).



Rys. 4.9.1 Schemat zasilania pojedynczej diody

Źródło: opracowanie własne

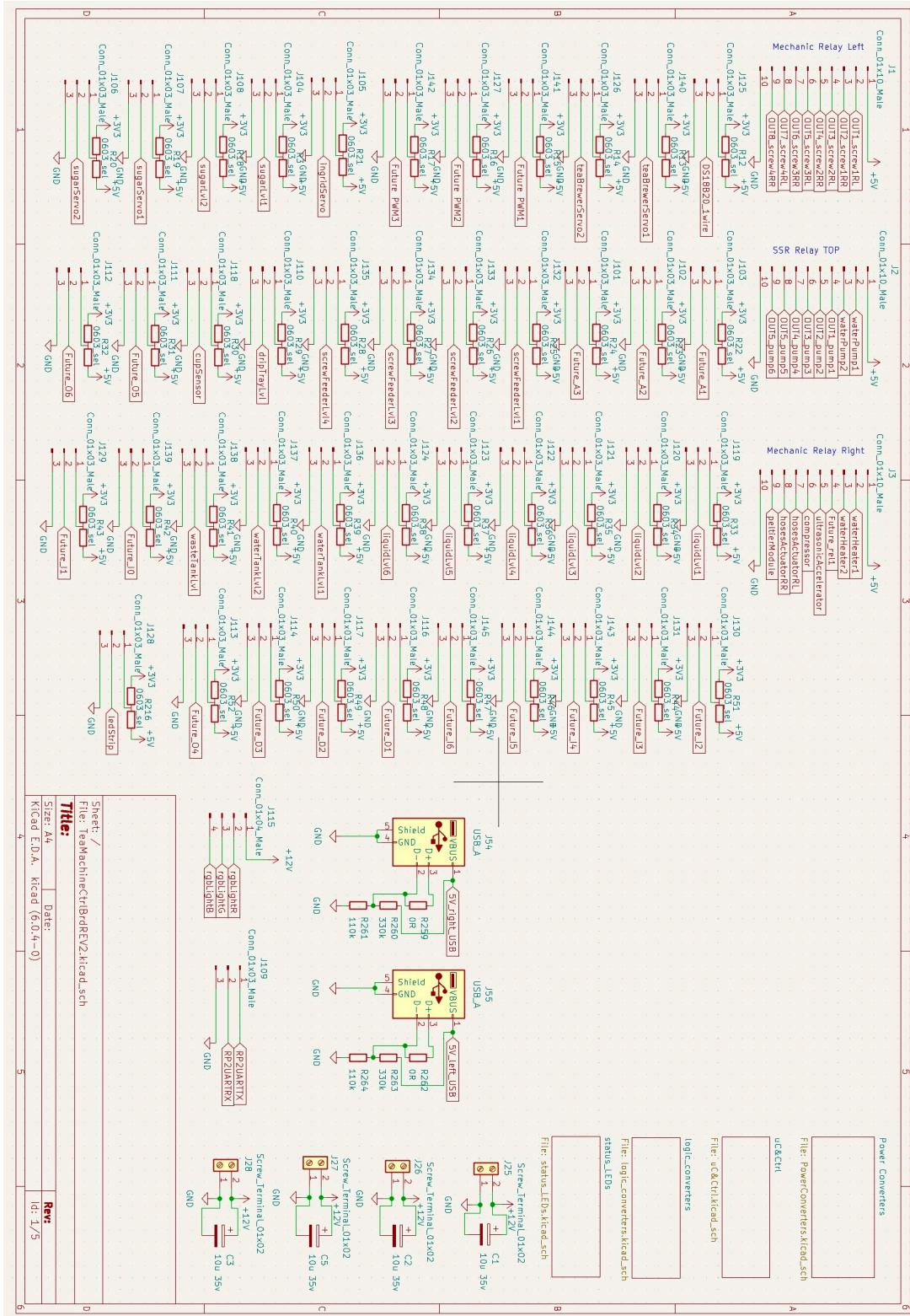


Rys. 4.9.2 Zasilanie diod informujących o obecności poszczególnych napięć

Źródło: opracowanie własne

4.10 Kompletny schemat układu

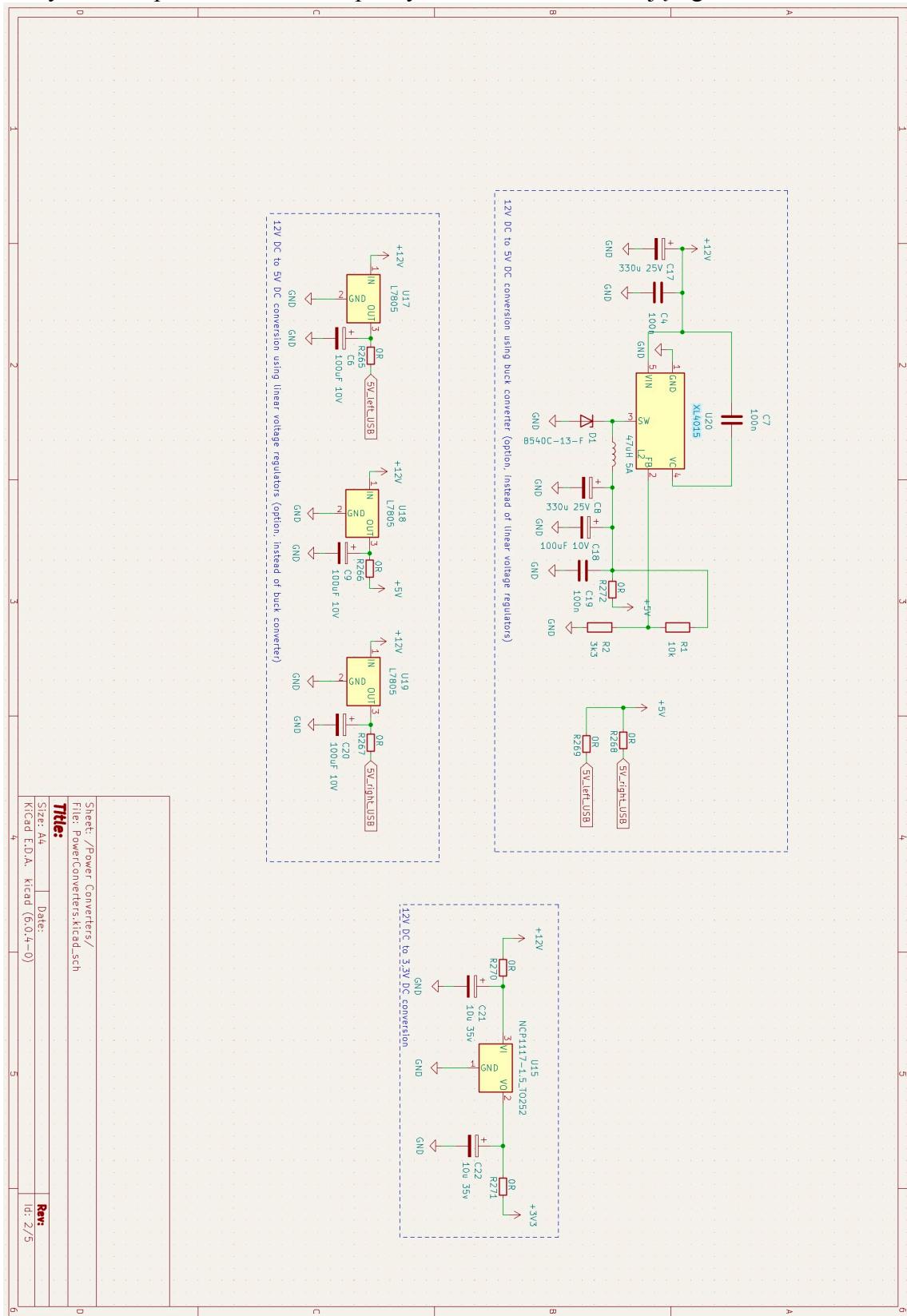
Na rys. 4.10.1 przedstawiono główny arkusz schematu — z wszystkimi portami wejściowymi i wyjściowymi oraz odnośnikami po pozostałych arkuszy.



Rys. 4.10.1 Schemat arkusz 1

Źródło: opracowanie własne

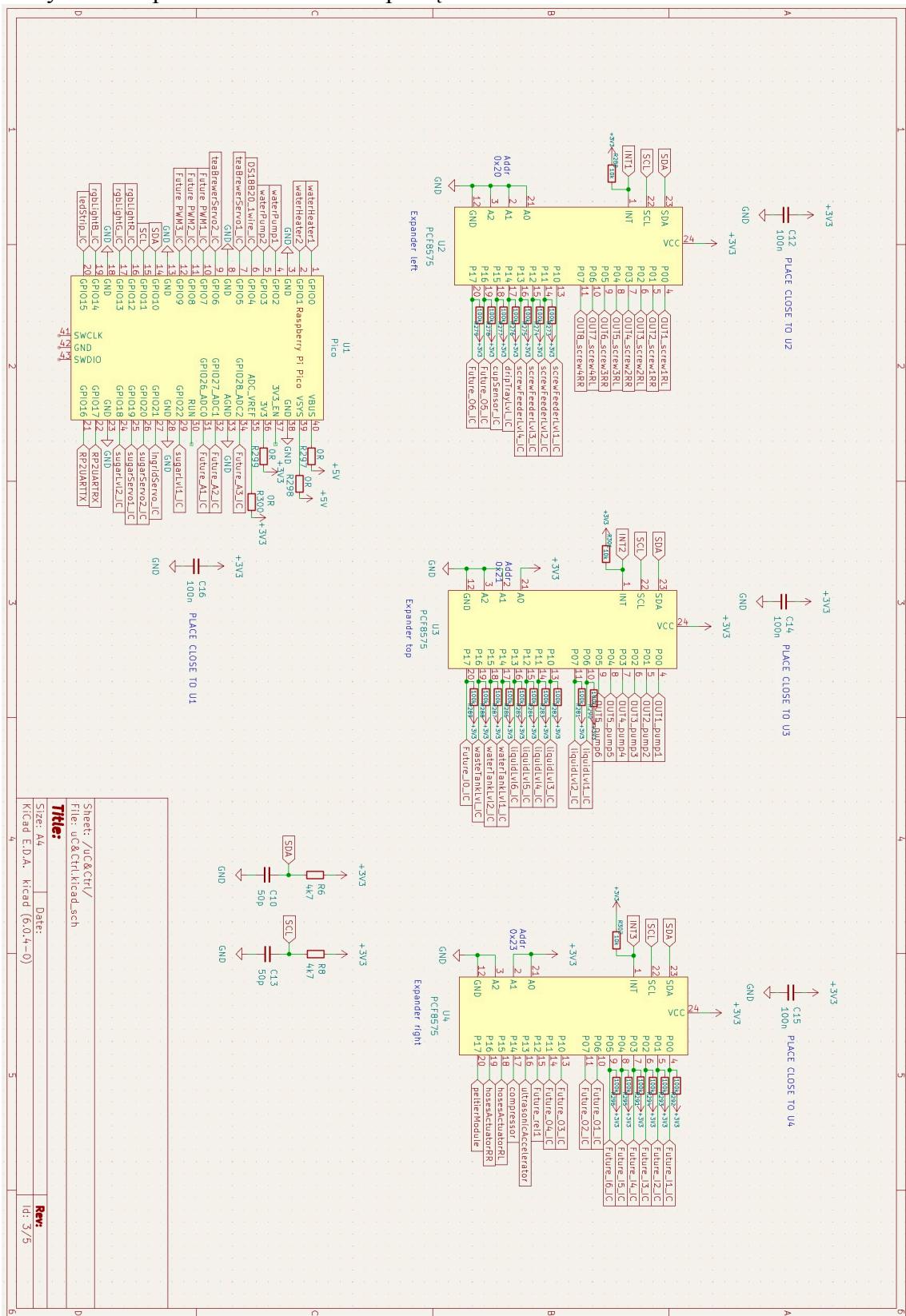
Na rys. 4.10.2 przedstawiono kompletny schemat układu zasilającego.



Rys. 4.10.2 Schemat arkusz 2

Źródło: opracowanie własne

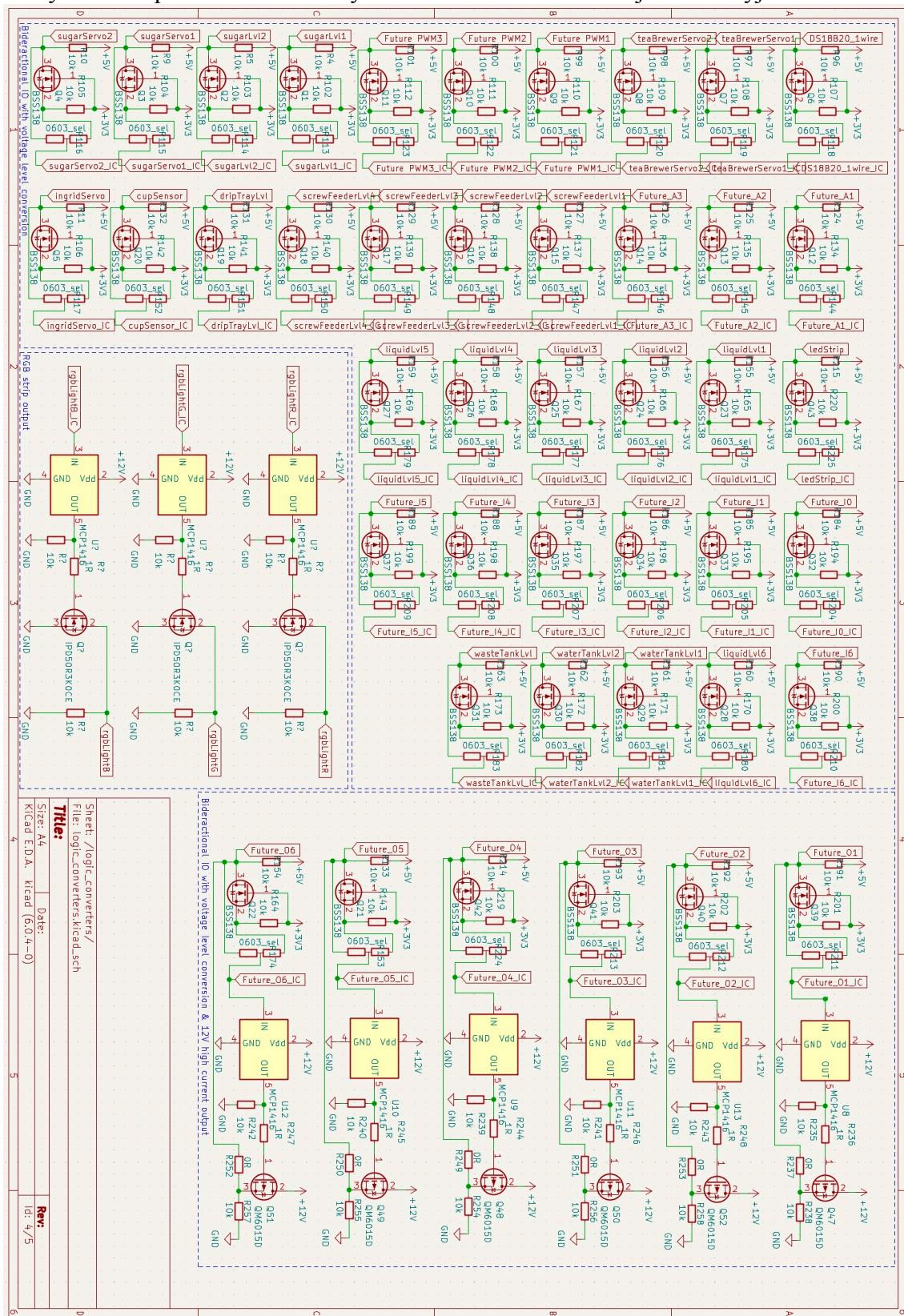
Na rys. 4.10.3 przedstawiono schemat podłączeń RPI Pico oraz układów PCF8575.



Rys. 4.10.3 Schemat arkusz 3

Źródło: opracowanie własne

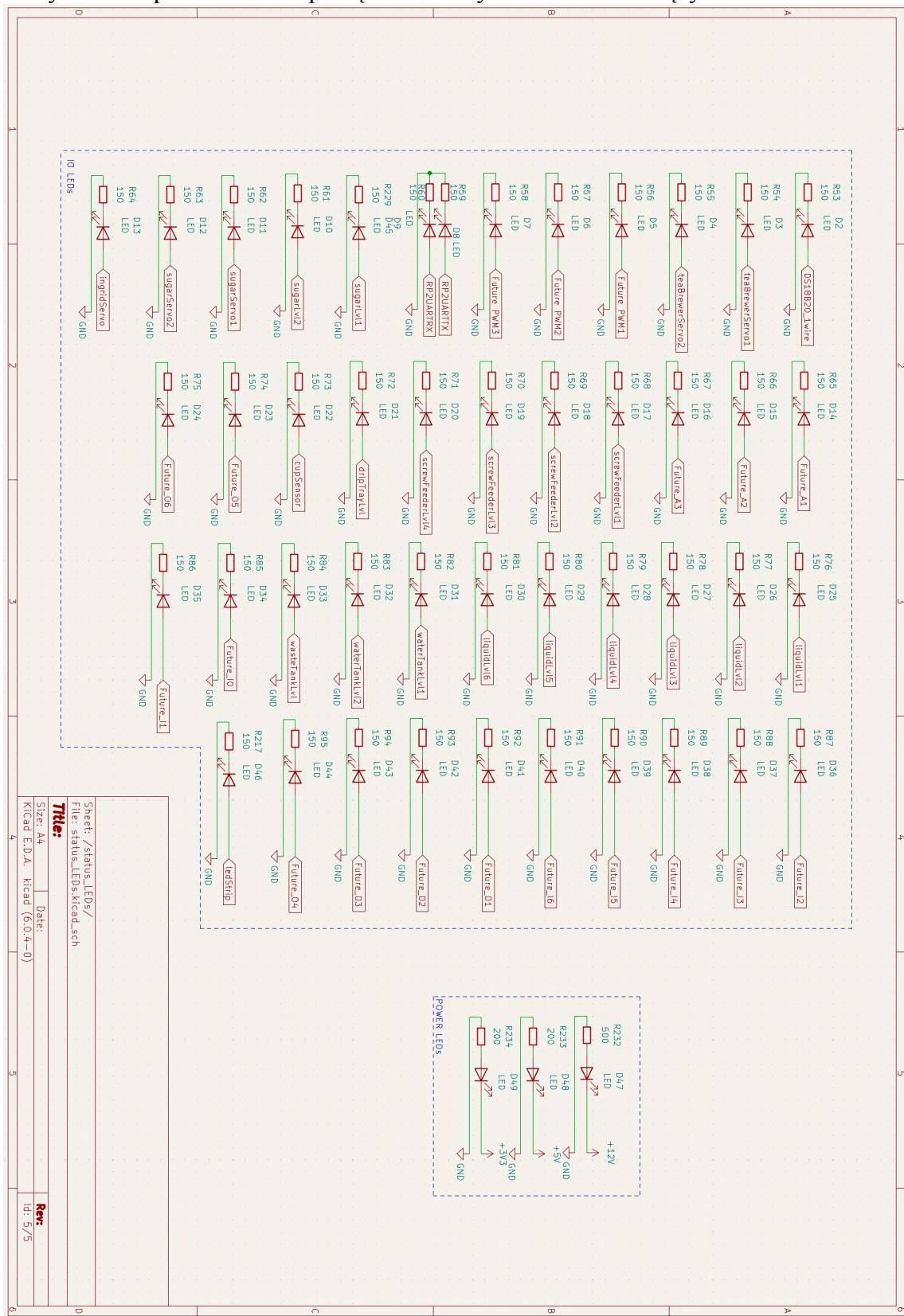
Na rys. 4.10.4 przedstawiono wszystkie uniwersalne bloki wejściowo-wyjściowe.



Rys. 4.10.4 Schemat arkusz 4

Źródło: opracowanie własne

Na rys. 4.10.5 przedstawiono podłączenia wszystkich diod świecących.

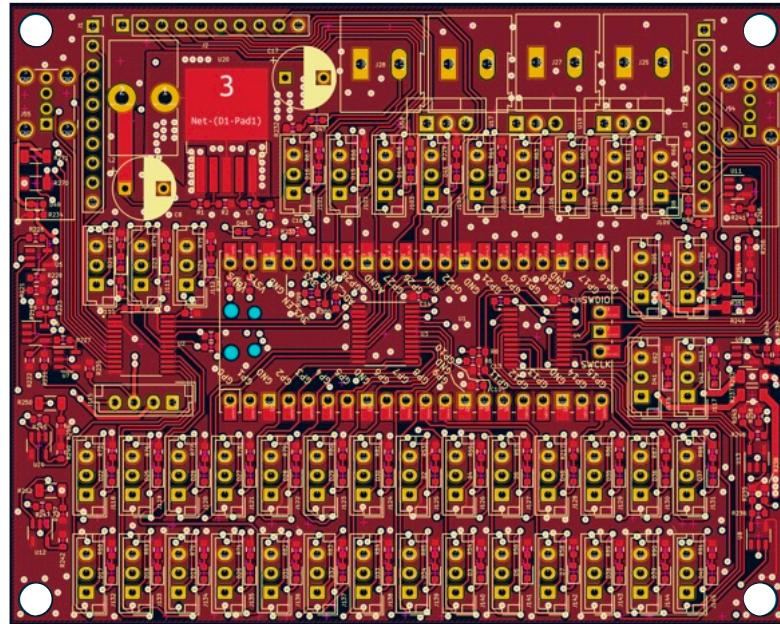


Rys. 4.10.5 Schemat arkusz 5

Źródło: opracowanie własne

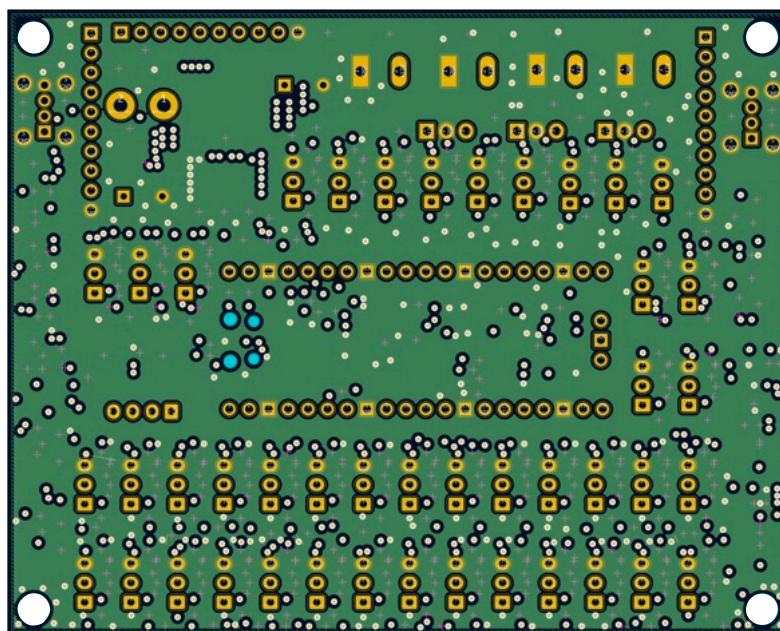
4.11 Płytki drukowane

Na rysunkach 4.11.1-4.11.4 przedstawiono widoki poszczególnych warstw zaprojektowanej płytki w programie Kicad.



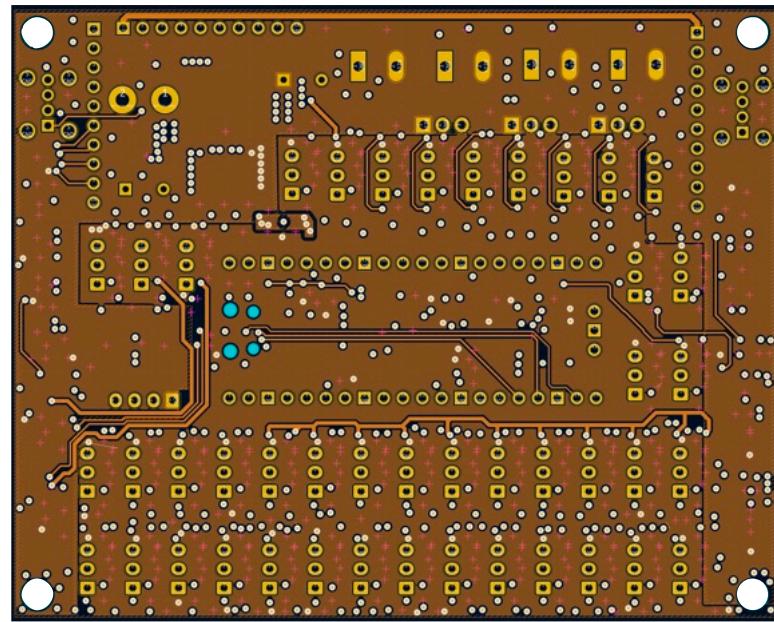
Rys. 4.11.1 Projekt PCB — widok z góry

Źródło: opracowanie własne



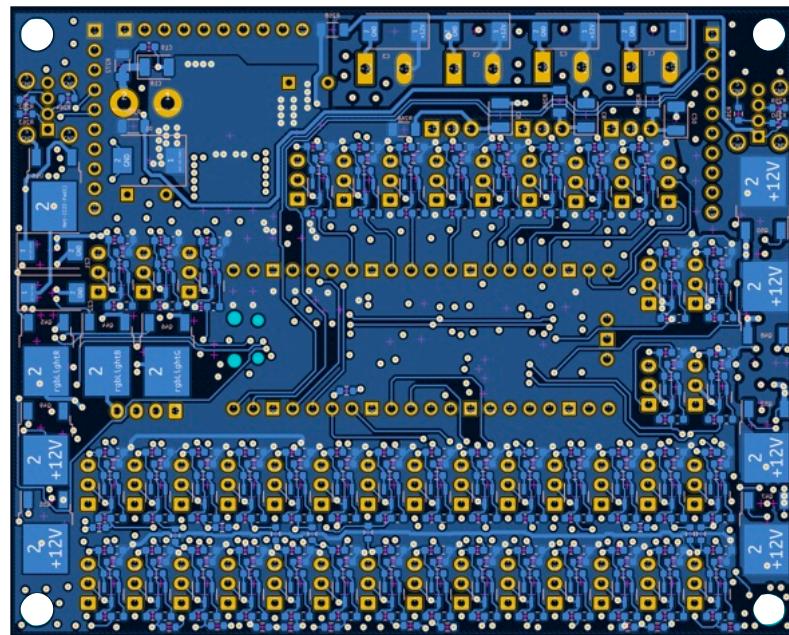
Rys. 4.11.2 Projekt PCB — warstwa wewnętrzna 1

Źródło: opracowanie własne



Rys. 4.11.3 Projekt PCB — warstwa wewnętrzna 2

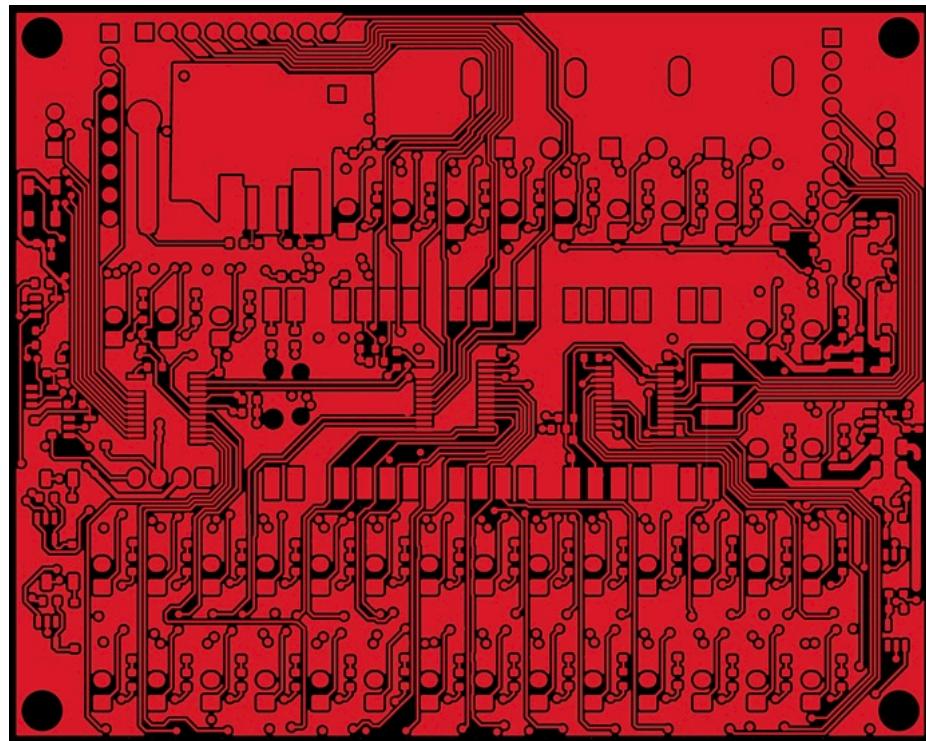
Źródło: opracowanie własne



Rys. 4.11.4 Projekt PCB — widok z dołu

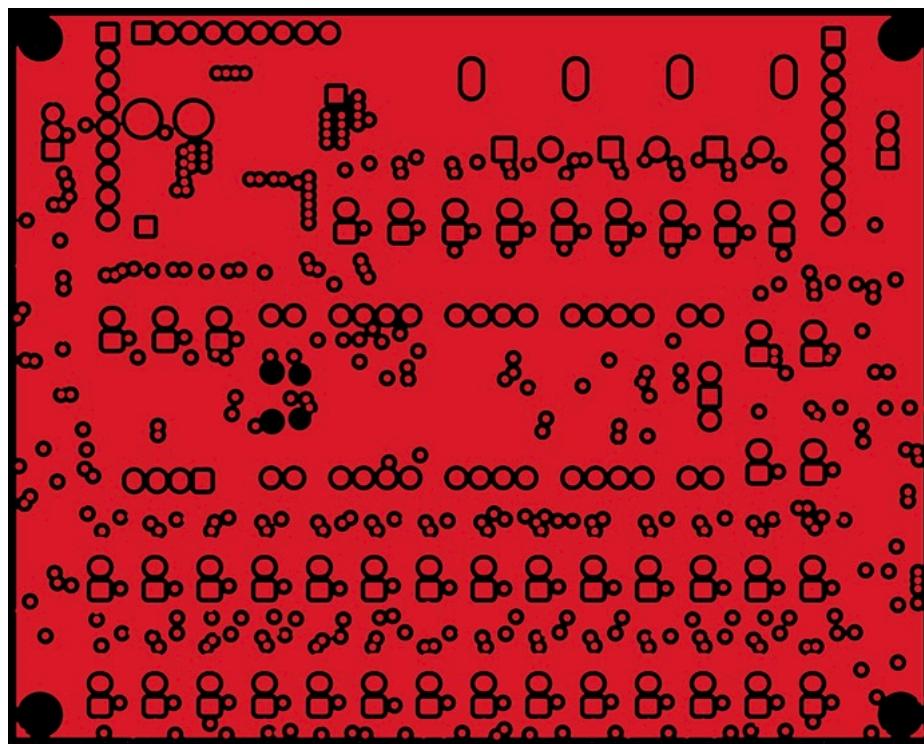
Źródło: opracowanie własne

Na rysunkach 4.11.5-4.11.17 przedstawiono pliki fabrykacyjne. Obejmują one kolejno maski poszczególnych warstw, miejsca do aplikacji pasty lutowniczej, adnotacje, miejsca do wykluczenia z nakładania maski lutowniczej, obrys płytki, miejsca odwierturnych otworów do późniejszej metalizacji oraz otwory nie metalizowane.



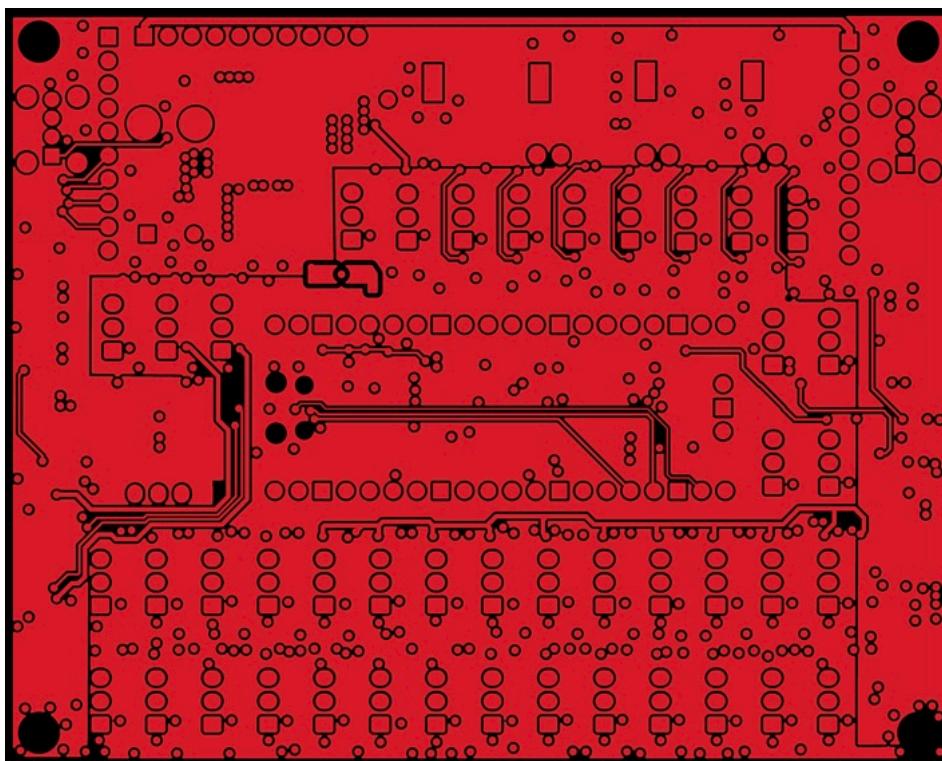
Rys. 4.11.5 Pliki fabrykacyjne — wierzchnia warstwa miedzi

Źródło: opracowanie własne



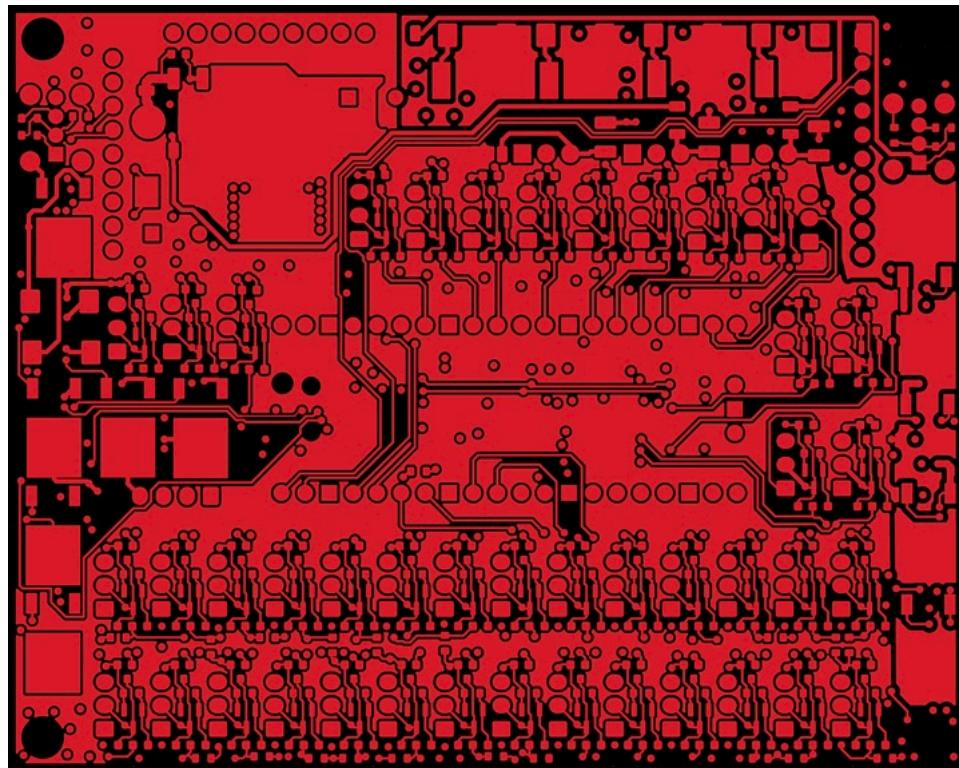
Rys. 4.11.6 Pliki fabrykacyjne — wewnętrzna warstwa miedzi 1

Źródło: opracowanie własne



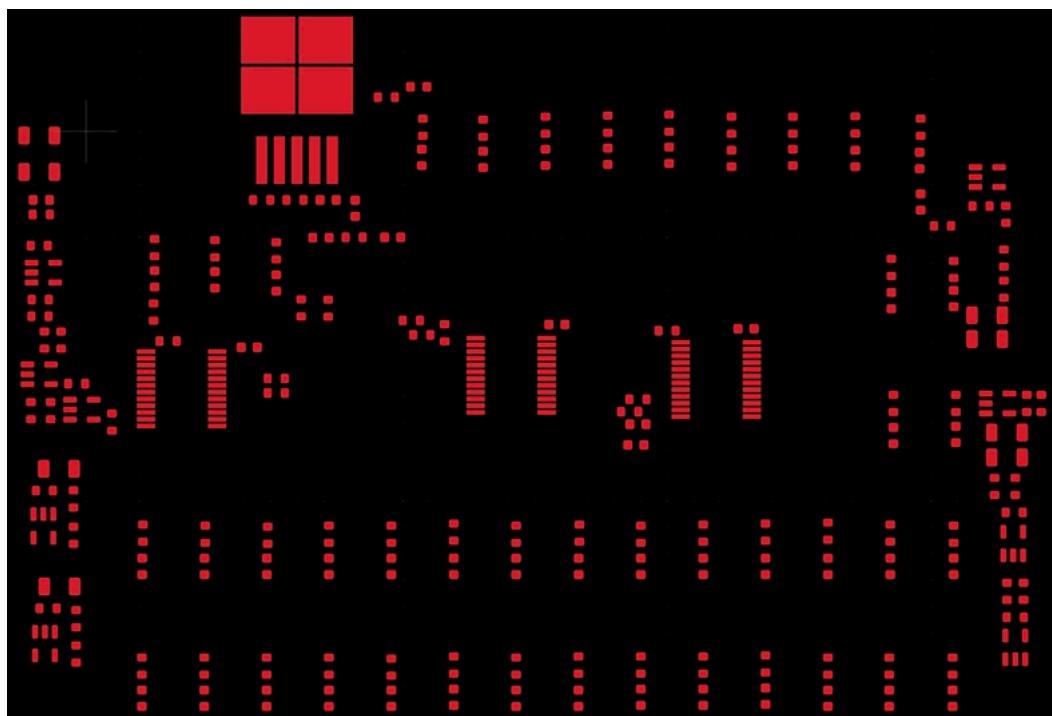
Rys. 4.11.7 Pliki fabrykacyjne — wewnętrzna warstwa miedzi 2

Źródło: opracowanie własne



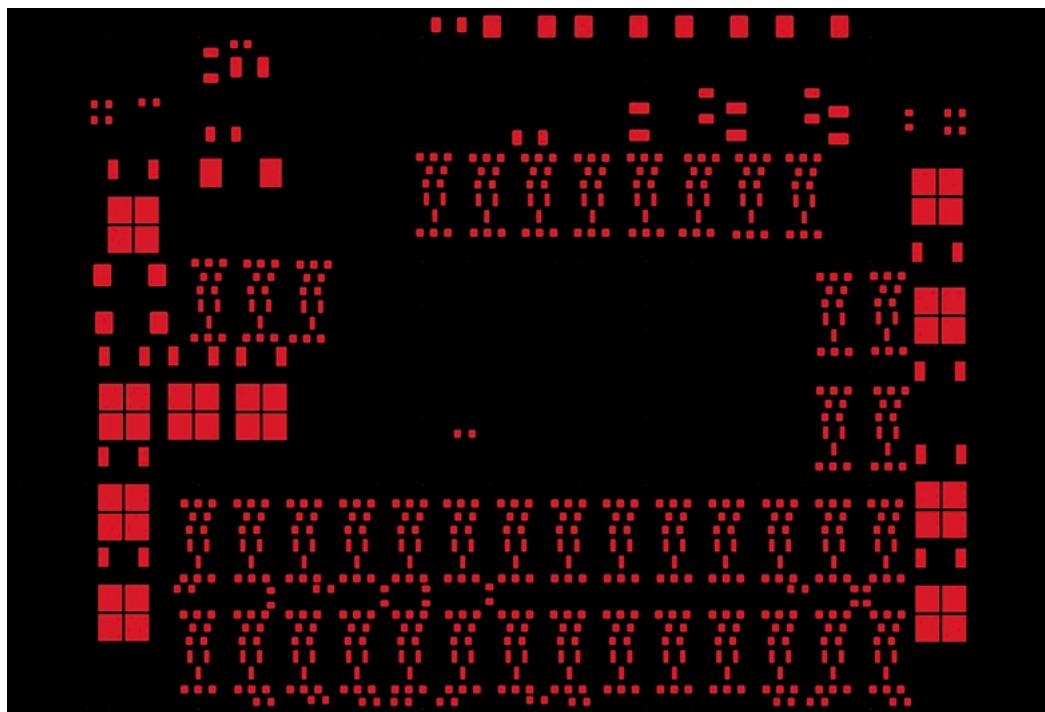
Rys. 4.11.8 Pliki fabrykacyjne — dolna warstwa miedzi

Źródło: opracowanie własne



Rys. 4.11.9 Pliki fabrykacyjne — miejsca do aplikacji pasty lutowniczej na warstwie wierzchniej

Źródło: opracowanie własne



Rys. 4.11.10 Pliki fabrykacyjne — miejsca do aplikacji pasty lutowniczej na warstwie dolnej

Źródło: opracowanie własne



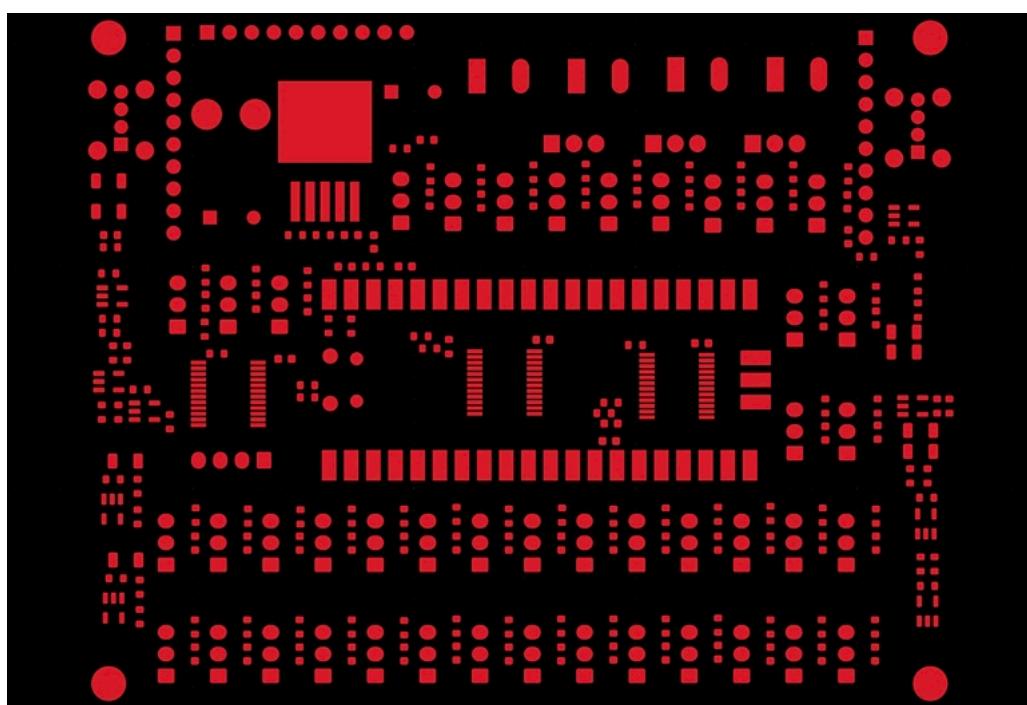
Rys. 4.11.11 Pliki fabrykacyjne — adnotacje na warstwie wierzchniej

Źródło: opracowanie własne



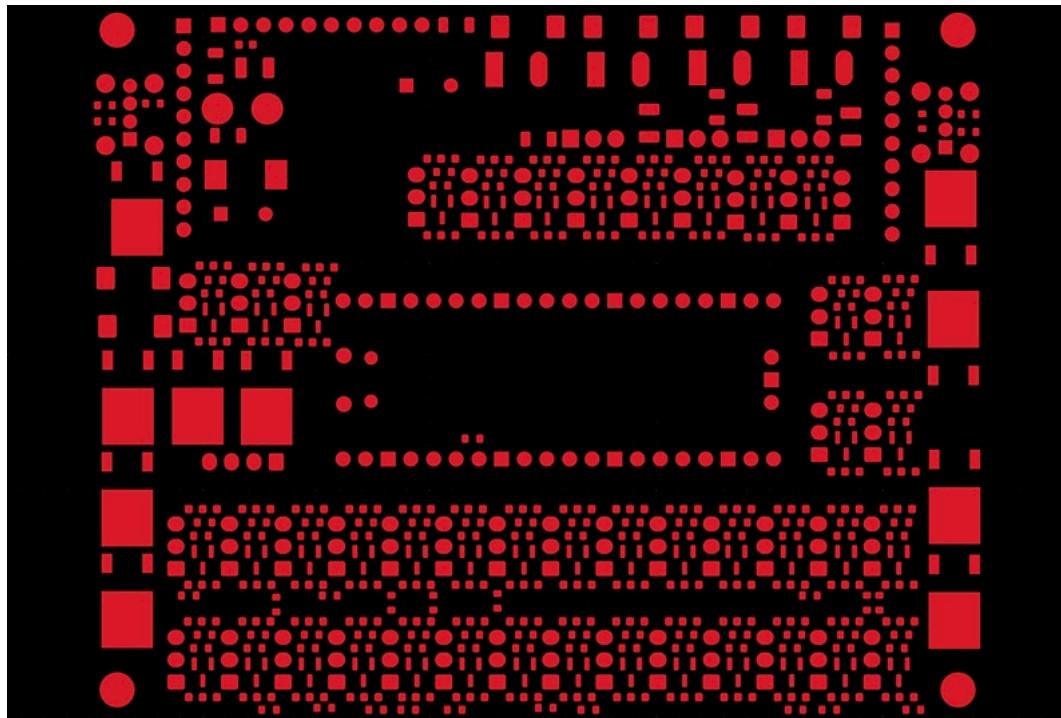
Rys. 4.11.12 Pliki fabrykacyjne — adnotacje na warstwie dolnej

Źródło: opracowanie własne



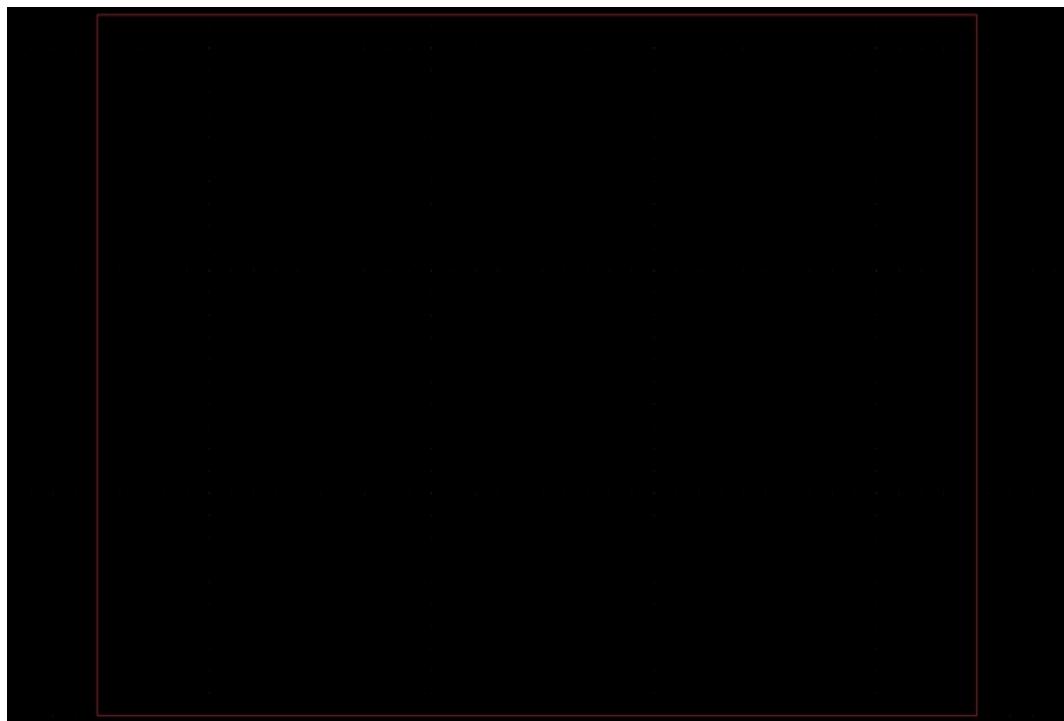
Rys. 4.11.13 Pliki fabrykacyjne — miejsca do wykluczenia z nakładania maski lutowniczej na warstwie wierzchniej

Źródło: opracowanie własne



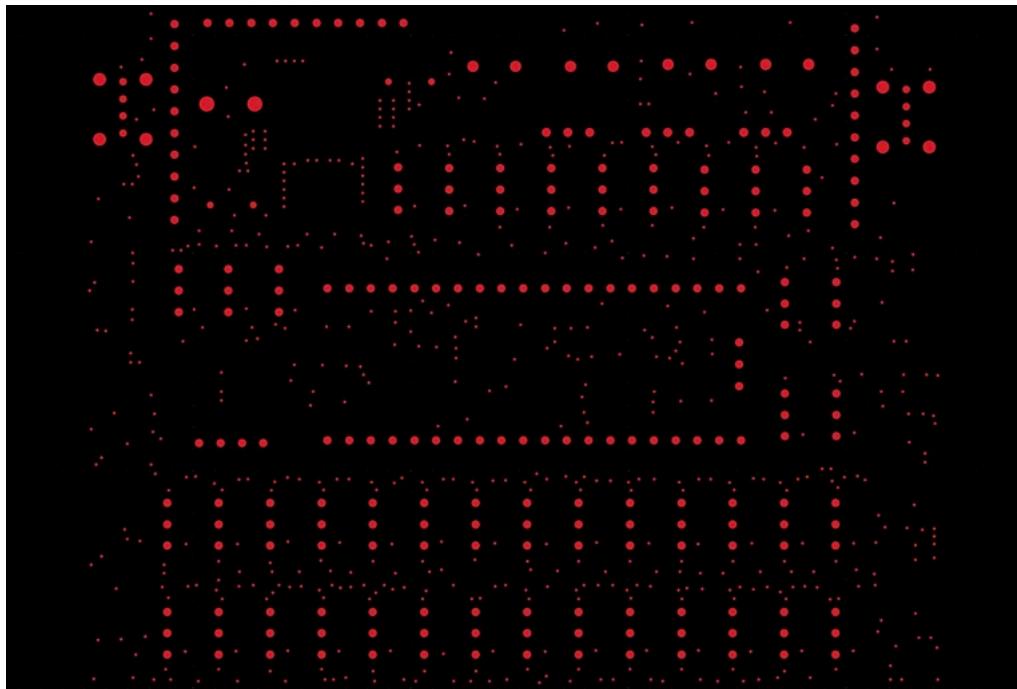
Rys. 4.11.14 Pliki fabrykacyjne — miejsca do wykluczenia z nakładania maski lutowniczej na warstwie dolnej

Źródło: opracowanie własne



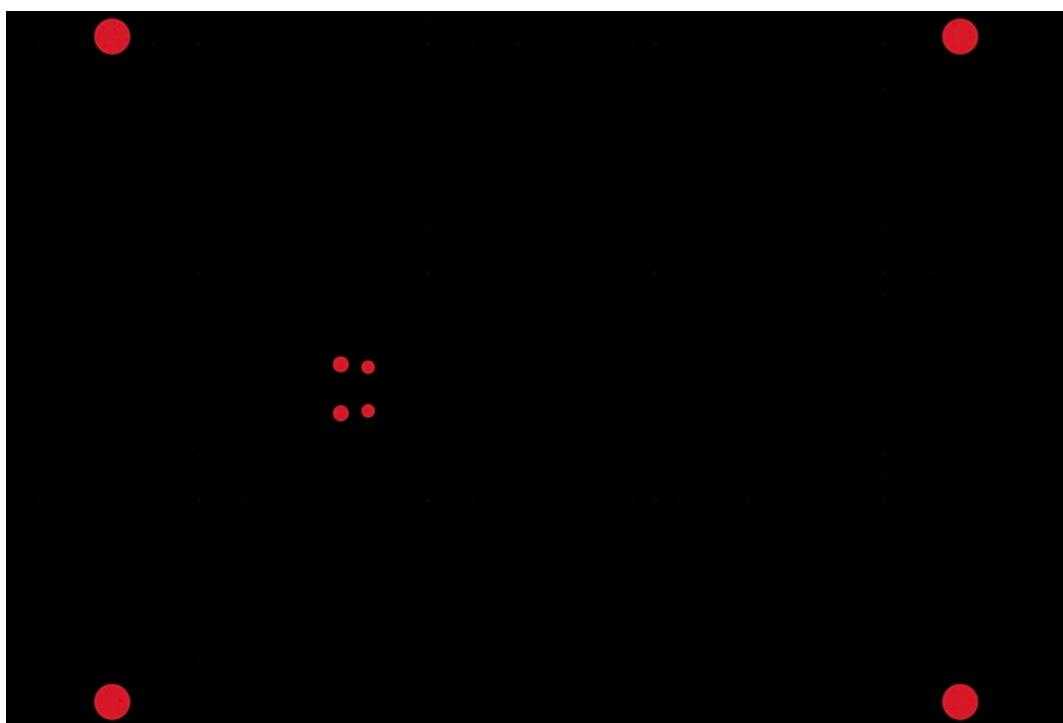
Rys. 4.11.15 Pliki fabrykacyjne — miejsca cięć płytka

Źródło: opracowanie własne



Rys. 4.11.16 Pliki fabrykacyjne — miejsca wiercenia otworów do późniejszej metalizacji

Źródło: opracowanie własne



Rys. 4.11.17 Pliki fabrykacyjne — miejsca wiercenia otworów bez metalizacji

Źródło: opracowanie własne

4.12 Zestawienie materiałów

Tab. 4.12.1 Zestawienie materiałów

Nr	Ilość	Desygnotory	Wartość	Footprint
1	6	C1, C2, C3, C5, C21, C22	10u 35 V	Capacitor_Tantalum_SMD:CP_EI_A-7343-31_Kemet-D
2	3	C4, C7, C19	100n	Capacitor_SMD:C_0603_1608Metric
3	4	C6, C9, C18, C20	100uF 10 V	Capacitor_Tantalum_SMD:CP_EI_A-3528-21_Kemet-B
4	2	C8, C17	330u 25 V	Capacitor_THT:CP_Radial_D8.0mm_P5.00mm
5	2	C10, C13	50p	Capacitor_SMD:C_0603_1608Metric
6	4	C12, C14, C15, C16	100n	Capacitor_SMD:C_0603_1608Metric
7	1	D1	B540C-13-F	Diode_SMD:D_SMC
8	48	D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, D16, D17, D18, D19, D20, D21, D22, D23, D24, D25, D26, D27, D28, D29, D30, D31, D32, D33, D34, D35, D36, D37, D38, D39, D40, D41, D42, D43, D44, D45, D46, D47, D48, D49	LED	LED_SMD:LED_0603_1608Metric
9	3	J1, J2, J3	Conn_01x10_Male	Connector_PinSocket_2.54mm:PinSocket_1x10_P2.54mm_Vertical
10	4	J25, J26, J27, J28	Screw_Terminal_01x02	TerminalBlock:TerminalBlock_Altech_AK300-2_P5.00mm
11	2	J54, J55	USB_A	Connector_USB:USB_A_Wuerth_614004134726_Horizontal
12	44	J101, J102, J103, J104, J105, J106, J107, J108, J109, J110, J111, J112, J113, J114, J116, J117, J118, J119, J120, J121, J122, J123, J124, J125, J126, J127, J128, J129, J130, J131, J132, J133, J134, J135, J136, J137, J138, J139, J140, J141, J142, J143, J144, J145	Conn_01x03_Male	Connector_JST:JST_EH_B3B-EH-A_1x03_P2.50mm_Vertical

13	1	J115	Conn_01x04_Male	Connector_JST:JST_EH_B4B-EH-A_1x04_P2.50mm_Vertical
14	1	L2	47uH 5A	Inductor_THT:L_Toroid_Vertical_L_14.7mm_W8.6mm_P5.58mm_Pulse_KM-1
15	43	Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, Q16, Q17, Q18, Q19, Q20, Q21, Q22, Q23, Q24, Q25, Q26, Q27, Q28, Q29, Q30, Q31, Q32, Q33, Q34, Q35, Q36, Q37, Q38, Q39, Q40, Q41, Q42, Q43	BSS138	Package_TO_SOT_SMD:SOT-23
16	3	Q44, Q45, Q46	IPD50R3K0CE	Package_TO_SOT_SMD:TO-252-2
17	6	Q47, Q48, Q49, Q50, Q51, Q52	QM6015D	Package_TO_SOT_SMD:TO-252-2
18	1	R1	10k	Resistor_SMD:R_0603_1608Metric
19	1	R2	3k3	Resistor_SMD:R_0603_1608Metric
20	86	R3, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R23, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34, R35, R36, R37, R38, R39, R40, R41, R42, R43, R44, R45, R46, R47, R48, R49, R50, R51, R52, R113, R114, R115, R116, R117, R118, R119, R120, R121, R122, R123, R144, R145, R146, R147, R148, R149, R150, R151, R152, R153, R174, R175, R176, R177, R178, R179, R180, R181, R182, R183, R204, R205, R206, R207, R208, R209, R210, R211, R212, R213, R216, R224, R225	0603_sel	test:0R_0603_selector

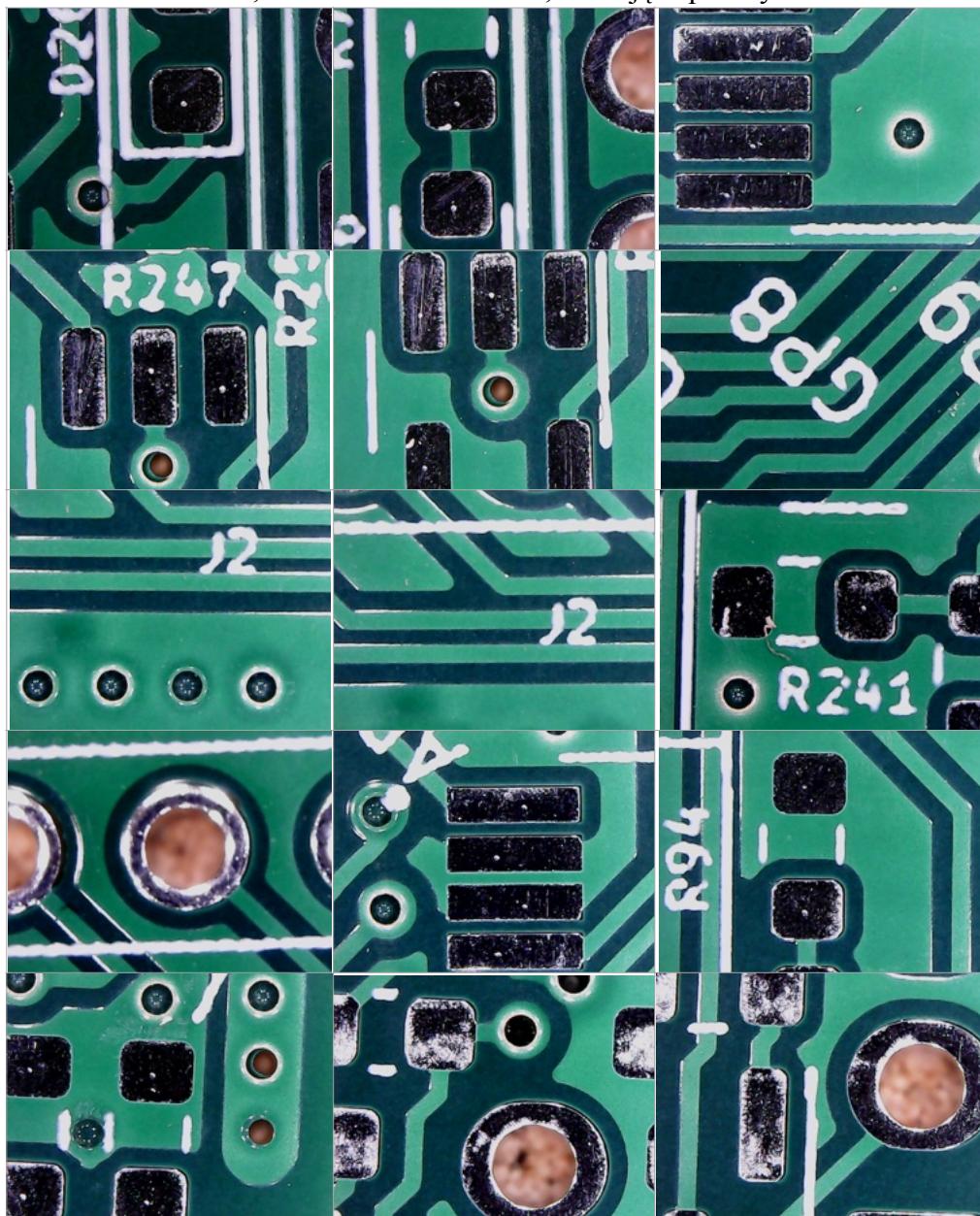
21	107	R4, R5, R7, R9, R10, R11, R96, R97, R98, R99, R100, R101, R102, R103, R104, R105, R106, R107, R108, R109, R110, R111, R112, R124, R125, R126, R127, R128, R129, R130, R131, R132, R133, R134, R135, R136, R137, R138, R139, R140, R141, R142, R143, R154, R155, R156, R157, R158, R159, R160, R161, R162, R163, R164, R165, R166, R167, R168, R169, R170, R171, R172, R173, R184, R185, R186, R187, R188, R189, R190, R191, R192, R193, R194, R195, R196, R197, R198, R199, R200, R201, R202, R203, R214, R215, R218, R219, R220, R221, R227, R228, R230, R235, R238, R239, R240, R241, R242, R243, R254, R255, R256, R257, R258, R280, R301, R302	10k	Resistor_SMD:R_0603_1608Metric
22	2	R6, R8	4k7	Resistor_SMD:R_0603_1608Metric
23	45	R53, R54, R55, R56, R57, R58, R59, R60, R61, R62, R63, R64, R65, R66, R67, R68, R69, R70, R71, R72, R73, R74, R75, R76, R77, R78, R79, R80, R81, R82, R83, R84, R85, R86, R87, R88, R89, R90, R91, R92, R93, R94, R95, R217, R229	150	Resistor_SMD:R_0603_1608Metric
24	9	R222, R223, R226, R236, R244, R245, R246, R247, R248	1R	Resistor_SMD:R_0603_1608Metric
25	1	R232	500	Resistor_SMD:R_0603_1608Metric

26	2	R233, R234	200	Resistor_SMD:R_0603_1608Metric
27	14	R237, R249, R250, R251, R252, R253, R265, R266, R267, R268, R269, R270, R271, R272	0R	Resistor_SMD:R_1206_3216Metric
28	6	R259, R262, R297, R298, R299, R300	0R	Resistor_SMD:R_0603_1608Metric
29	2	R260, R263	330k	Resistor_SMD:R_0603_1608Metric
30	2	R261, R264	110k	Resistor_SMD:R_0603_1608Metric
31	23	R273, R274, R275, R276, R277, R278, R279, R281, R282, R283, R284, R285, R286, R287, R288, R289, R290, R291, R292, R293, R294, R295, R296	100k	Resistor_SMD:R_0603_1608Metric
32	1	U1	Pico	pico:RPi_Pico_SMD_TH
33	3	U2, U3, U4	PCF8575	test:SOP65P775X200-24N
34	9	U5, U6, U7, U8, U9, U10, U11, U12, U13	MCP1416	Package_TO_SOT_SMD:SOT-23-5
35	1	U15	NCP1117-1.5_TO252	Package_TO_SOT_SMD:TO-252-2
36	3	U17, U18, U19	L7805	Package_TO_SOT_THT:TO-220-3_Vertical
37	1	U20	XL4015	Package_TO_SOT_SMD:TO-263-5_TabPin3

4.13 Montaż

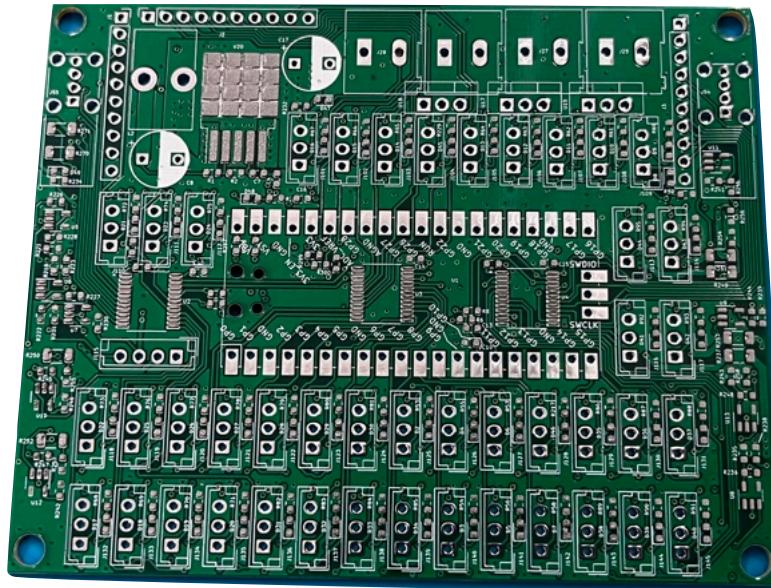
Pierwszy etap montażu polega na sprawdzeniu, czy płytka została wykonana zgodnie z projektem oraz czy kluczowe sygnały, jak płaszczyzny zasilania nie są zwarte w wyniku ewentualnych błędów produkcyjnych. Następnie należy zweryfikować ciągłość pozostałych sygnałów, takich jak linie magistrali I2C.

Wykonano całosciowy przegląd płytki za pomocą mikroskopu (rys. 4.13.1). Sprawdzono, czy nie występują wizualne zmiany, które potencjalnie stanowią zagrożenie dla prawidłowego funkcjonowania układu takie jak: przebarwienia, rozwarstwienia laminatu, niedotrawienia miedzi, brakujące punkty lutownicze.



Rys. 4.13.1. Inspekcja PCB pod mikroskopem
Źródło: opracowania własne

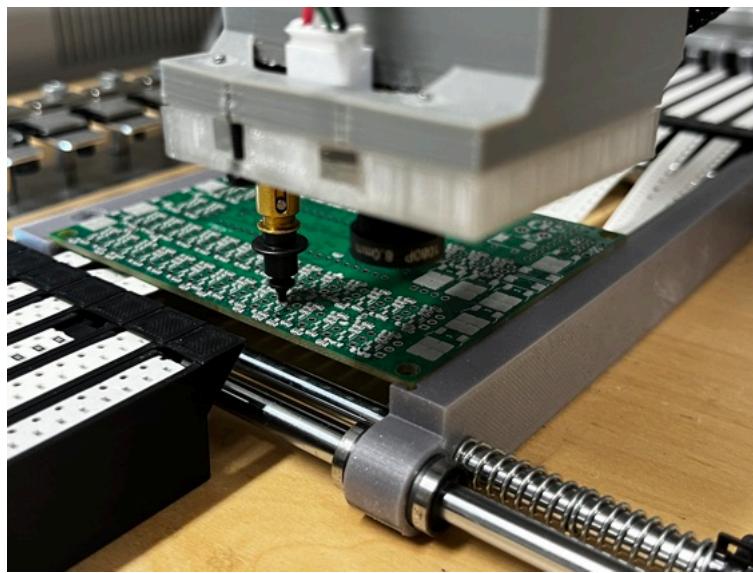
Uzbrojono maszynę do sitodruku, montując szablon przygotowany specjalnie do tej płytki. Nałożono pastę do lutowania na górną warstwę, czego wynik widoczny jest na rys. 4.13.2.



Rys. 4.13.2 PCB z nałożoną pastą

Źródło: opracowania własne

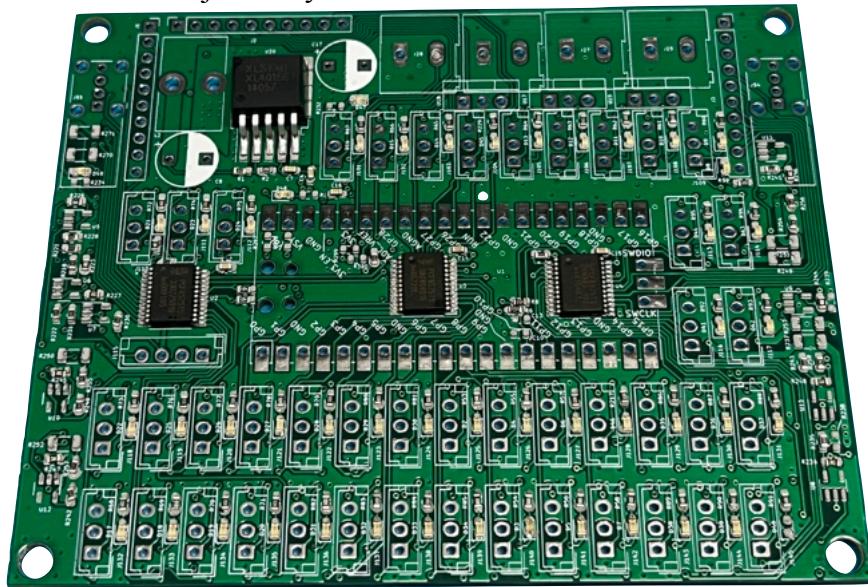
Następnie przygotowano maszynę typu *pick and place*. Załadowano podajniki z komponentami oraz rozpoczęto montaż (rys. 4.13.3).



Rys. 4.13.3 Płytki w trakcie montażu

Źródło: opracowania własne

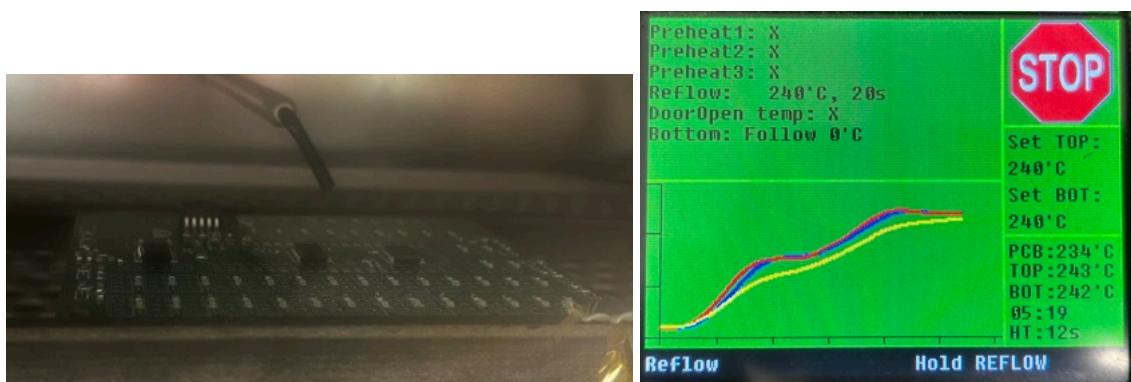
Następnie pozostałe komponenty, takie jak układ XL4015, czy ekspandery położono ręcznie. Płytki z obsadzonymi wszystkimi (posiadanymi na dzień montażu) komponentami widoczna jest na rys. 4.13.4.



Rys. 4.13.4 płytka z obsadzonymi komponentami SMD

Źródło: opracowania własne

Po zakończonym montażu warstwy górnej, płytka trafiła do specjalnego pieca do lutowania rozpływowego — rys 4.13.5, gdzie lutowanie odbyło się według nastaw temperatury widocznych na rys. 4.13.6. (widoczne dwa etapy — wygrzewanie oraz lutowanie rozpływowe).



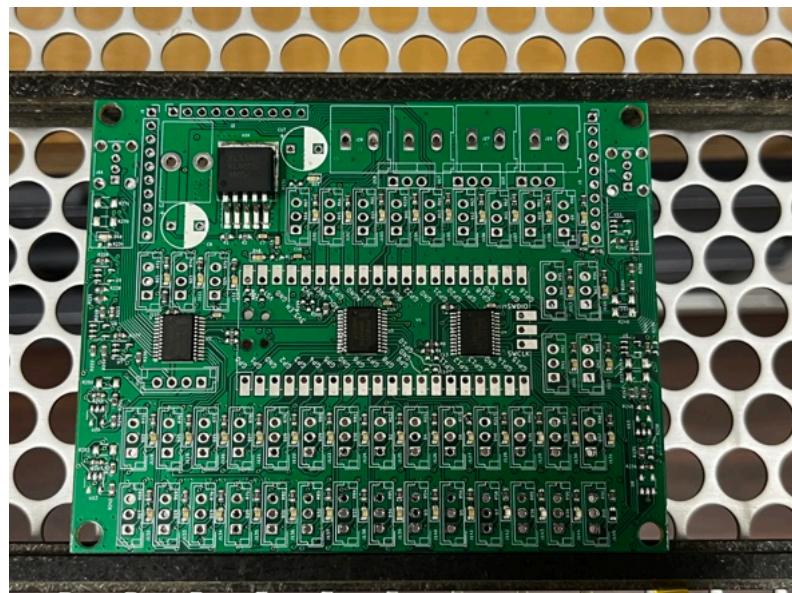
Rys. 4.13.5 Płytki w piecu

Źródło: opracowania własne

Rys. 4.13.6 Profil temperaturowy

Źródło: opracowania własne

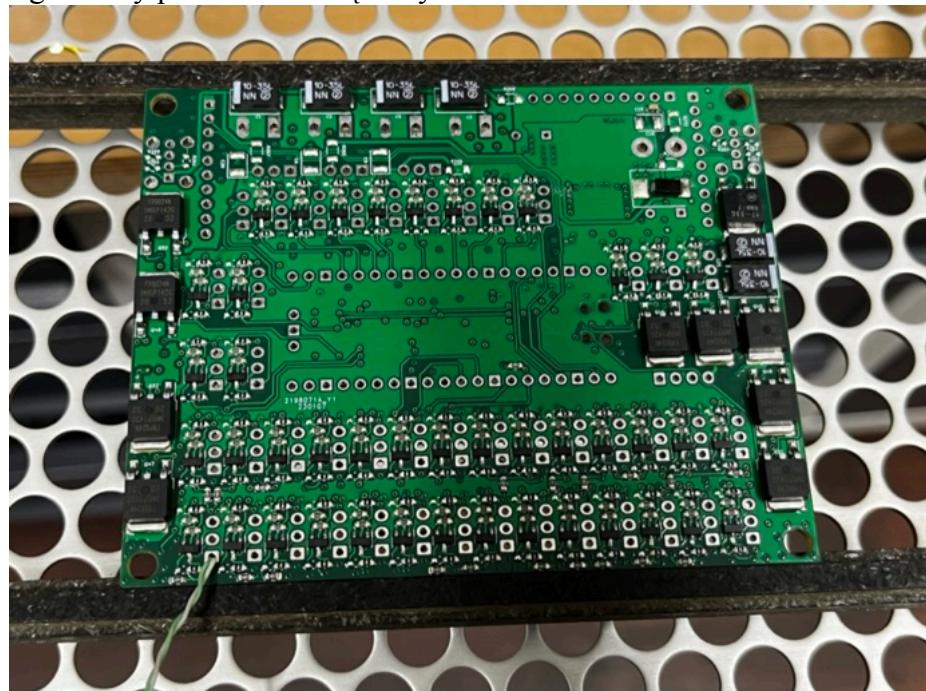
Stan płytki po zlutowaniu widoczny jest na rys. 4.13.7.



Rys. 4.13.7 Płyta po lutowaniu rozpływowym

Źródło: opracowania własne

Po zakończeniu lutowania i ostygnięciu płytki, ponownie zostały powtórzone powyższe kroki dla warstwy dolnej, tj: nałożenie pasty, ułożenie komponentów, zlutowanie w piecu, czego efekty przedstawione są na rys. 4.13.8.

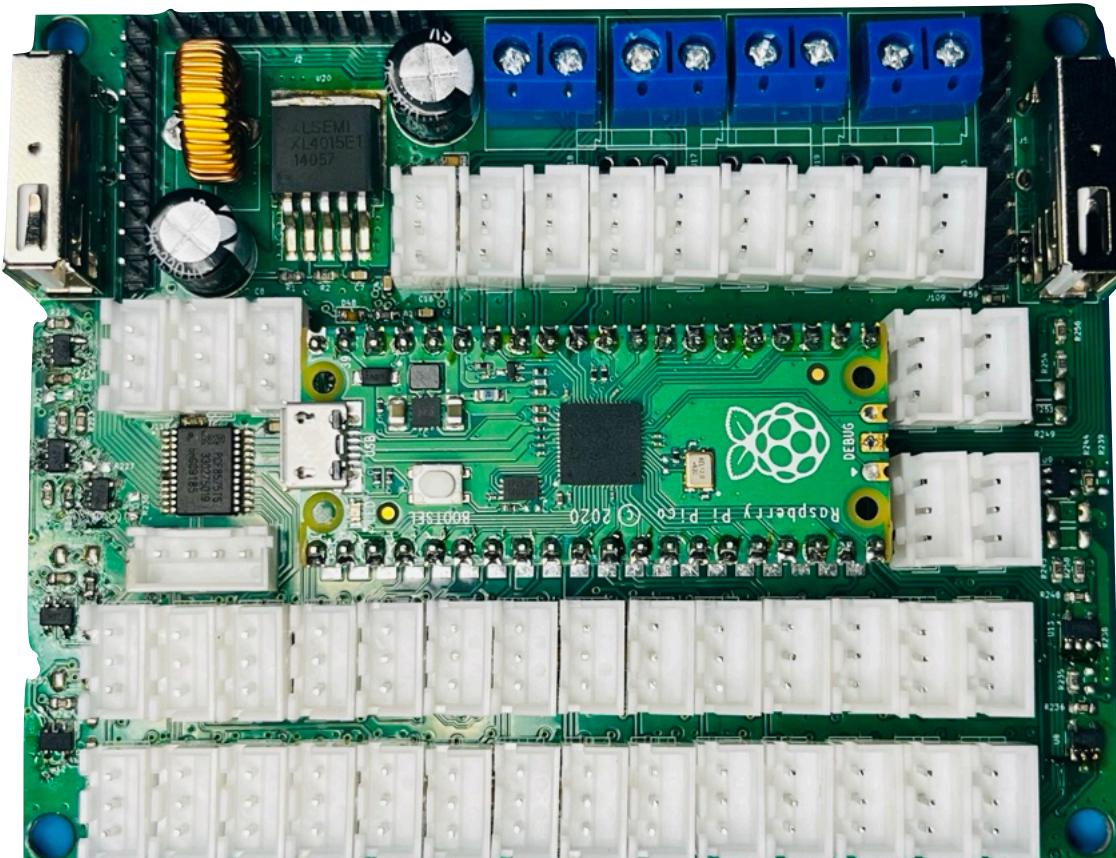


Rys. 4.13.8 Zlutowana warstwa dolna

Źródło: opracowania własne

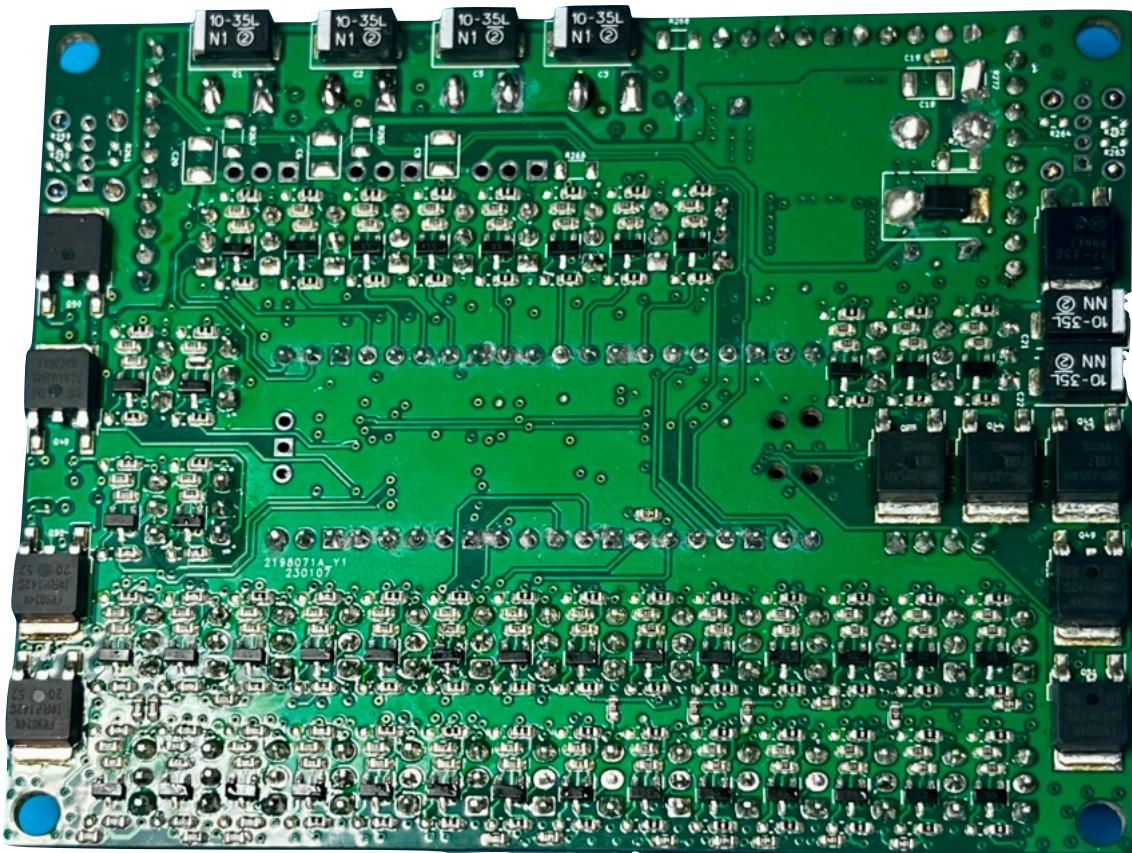
Ze względu na problemy z dostępnością elementów w chwili montażu, na płytce brakuje (rys. 4.13.7) sterowników do dużych tranzystorów MOSFET (od Q44 do Q52) oraz kilku rezystorów.

Ze względu na utrudniony dostęp do wielu komponentów SMD, po zakończonym etapie montażu, przystąpiono do uruchamiania płytki. Następnie należało upewnić się, że wszystkie peryferia funkcjonują. Także w tym momencie wykonano wszystkie czynności opisane w rozdziałach 4.14 oraz 4.15. Dopiero na koniec, po pełnej weryfikacji działania układu, przylutowano ostatnie komponenty THT, czego efekt jest przedstawiony na rys. 4.13.9 oraz rys. 4.13.10.



4.13.9 W pełni zmontowana płytka — widok górnny

Źródło: opracowania własne

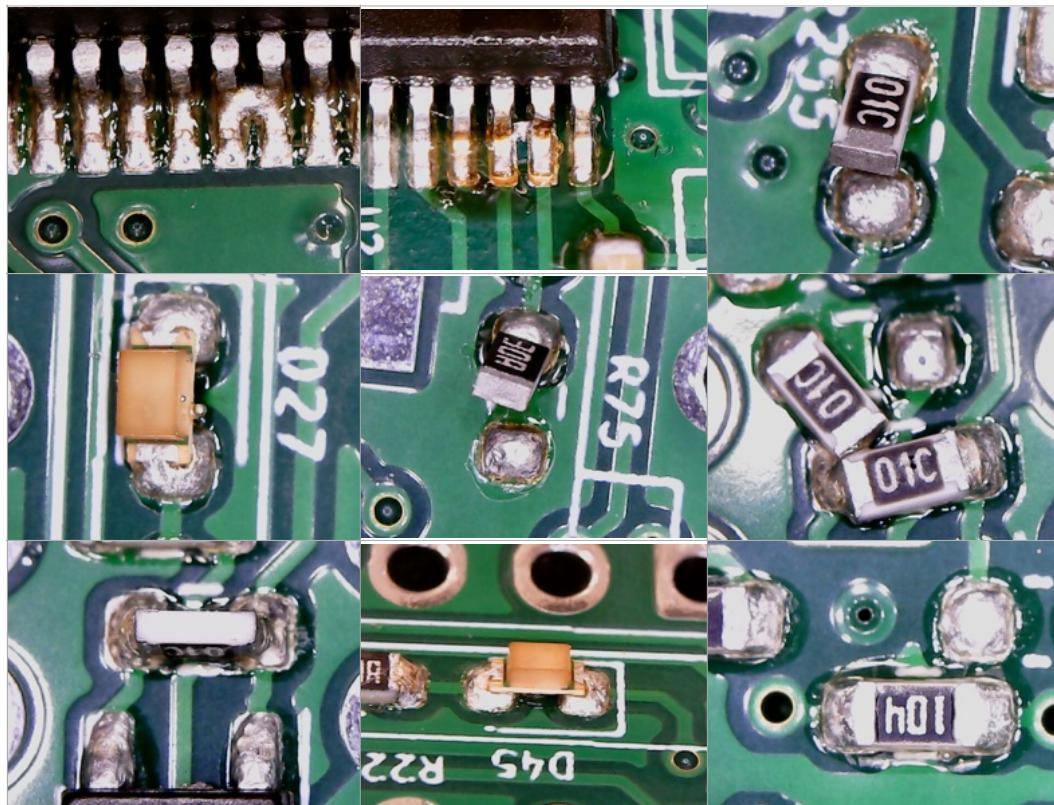


4.13.10 W pełni zmontowana płytka — widok dolny

Źródło: opracowania własne

4.14 Uruchamianie układu

Uruchamianie płytka rozpoczęto jeszcze raz od weryfikacji wizualnej, czyli sprawdzeniu pod mikroskopem wszystkich komponentów w poszukiwaniu ewentualnych niedoskonałości.

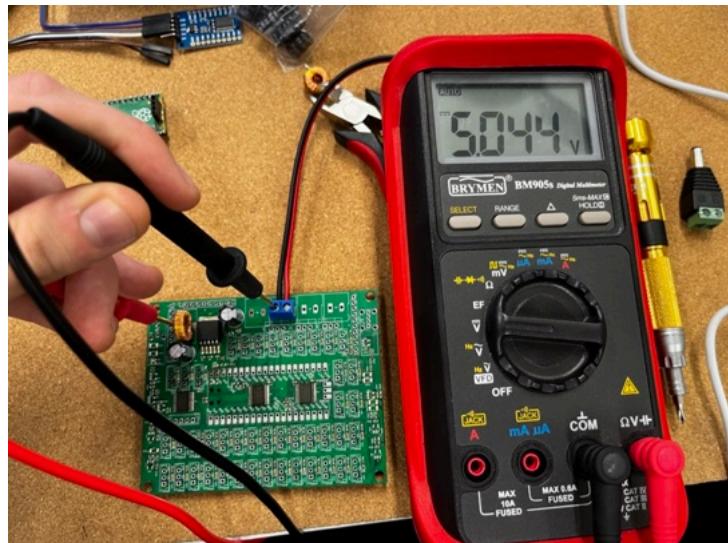


Rys.4.14.1 — Znalezione niedoskonałości
Źródło: opracowania własne

Na rysunku 4.14.1 przedstawiono przykładowe błędy, wykryte podczas badania za pomocą mikroskopu. Na rysunku 4.14.1 widoczne są: zwarcie między wyprowadzeniami układu scalonego, nieprzyłutowane punkty, przemieszczone elementy, przekręcone lub nie przylegające prawidłowo komponenty. Błędy poprawiono za pomocą lutownicy klasycznej i stacji na gorące powietrze.

Następnie sprawdzono zwarcia pomiędzy sygnałami zasilającymi oraz zweryfikowano ciągłość innych ważnych sygnałów.

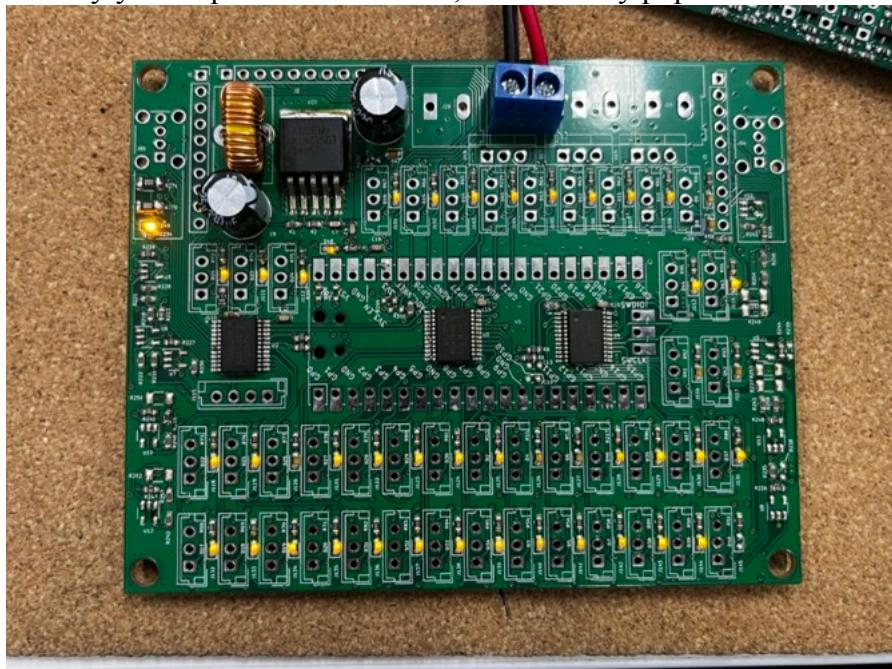
Układ celowo nie miał zamontowanych rezystorów łączących przetwornice napięcia z układem aby sprawdzić czy dostarczają one w stanie jałowym oczekiwane napięcia. Podłączono zasilanie 12 V z zasilacza laboratoryjnego. Ustawiono maksymalny prąd wynoszący 100mA. Na początek sprawdzono przetwornicę XL4015, a następnie stabilizator napięcia serii NCP1117.



Rys. 4.14.2 Weryfikacja napięć zasilających

Źródło: opracowania własne

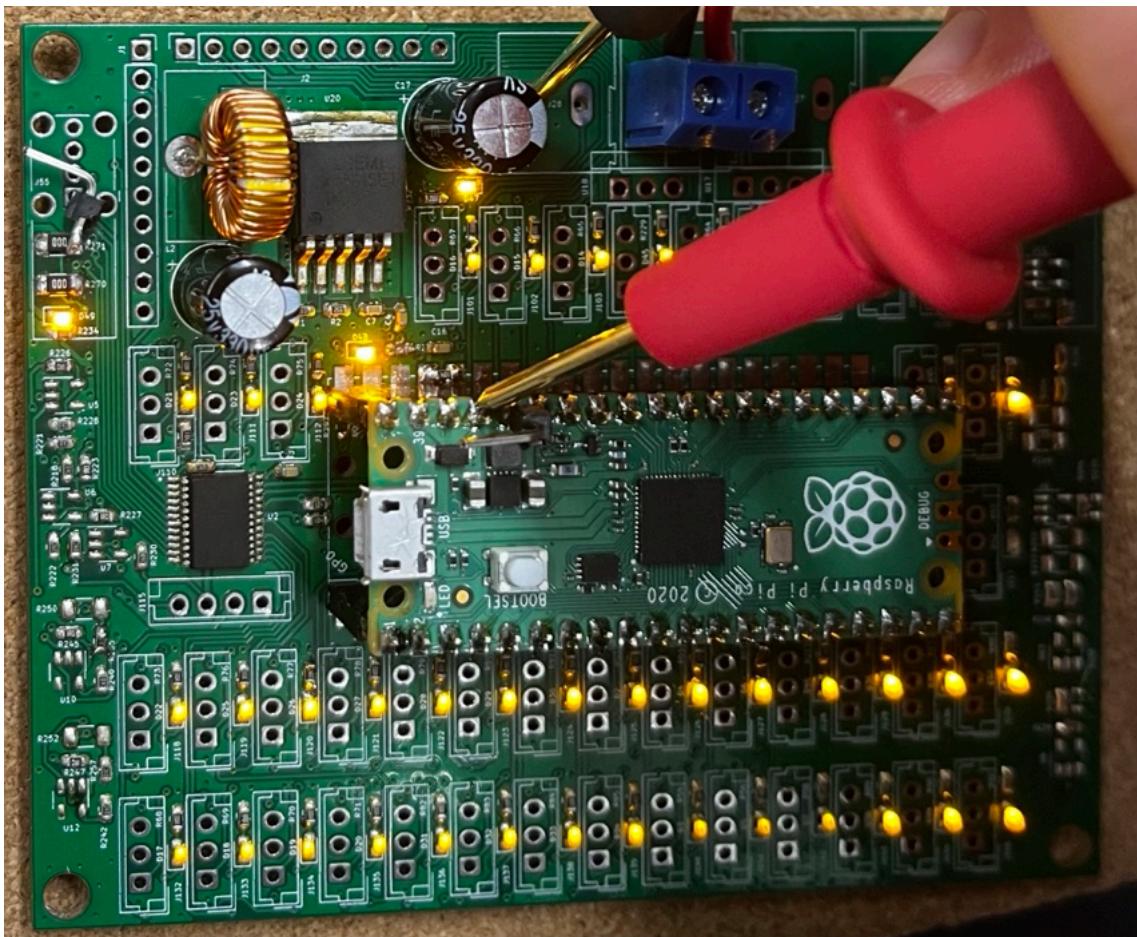
Następnie przylutowano rezystory R272, R268, R269 oraz R271, łączące przetwornicę z liniami zasilania. Włączono zasilacz ponownie i zweryfikowano, czy żadne z komponentów na płytce nie powodują zwarć i nie pobierają zbyt dużego prądu. Żadnych zwarć nie zaobserwowano, jednak diody, które się nie zaświeciły, wskazywały na problemy z poszczególnymi kanałami wyjściowymi (rysunek 4.14.3). Błędy spowodowane były nieodpowiednimi lutami, które zostały poprawione.



Rys. 4.14.3 Sprawdzenie zachowania układu po podłączeniu zasilania

Źródło: opracowania własne

Następnie wyłączono i odłączono zasilacz laboratoryjny. Przylutowano listwę kołkową i osadzono na niej moduł RPI Pino. Tymczasowo podłączono sygnał 3V3 z modułu do zasilania 3,3 V na całej płytce (poprzez zworkę w formie rezystora R299). Podłączono do komputera i wgrano program testowy, który konfiguruje wszystkie piny RPI Pico, jak i wyjścia. Ustawia je kolejno w stan wysoki, co w efekcie powoduje sekwencyjne świecenie diod. Skonfigurowano również ekspandery PCF8575 jako wyjścia, które okresowo były ustawiane w stan niski. Rysunek 4.14.4 przedstawia zachowanie układu po zamontowaniu mikrokontrolera.



Rys. 4.14.4 Zachowanie układu po zamontowaniu mikrokontrolera

Źródło: opracowania własne

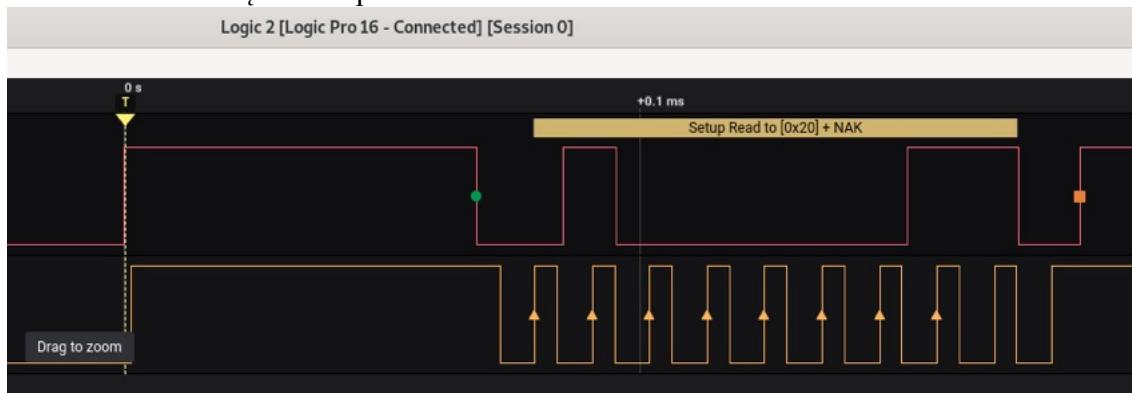
Niestety zachowanie układu nie spełniło wymagań, określonych programem, użytym w celu wykonania testów. Część diod nie zapalała się w ogóle, część migała w innej kolejności niż zakładano. Najgorsze było to, że niekiedy układ po krótkiej pracy (a nie od razu po podłączeniu zasilania) przestawały działać w ogóle.

Proces poszukiwania problemów był dosyć czasochłonny i złożony. Nie był to pojedynczy problem a kilka niezależnych w części z winy z twórcy, w części z przyczyn od niego nie zależnych.

Pierwszym odnalezionym problemem był mostek z cyny pomiędzy wyprowadzeniami jednego z ekspanderów (już po wcześniejszej poprawie lutów), akurat pomiędzy sygnałami SDA oraz SCL, tworzącymi magistralę I2C. Dało się go dostrzec dopiero po oględzinach mikroskopem pod bardzo dużym kątem. Przez to komunikacja między urządzeniami nie była możliwa.

Niestety to nie był jedyny problem. W wyniku dokładnych oględzin wszystkich komponentów podłączonych do magistrali nie znaleziono żadnych zwarć. Przeanalizowano dokładnie poprowadzenie sygnałów na płytce — wszystko było zaprojektowane poprawnie. Ostatecznie usunięto wszystkie układy scalone oraz wszystkie komponenty podłączone do płytki. Okazało się, że linie SDA i SCL nadal były zwarte. Przy zamawianiu płytka była wybrana dodatkowa opcja sprawdzenia ciągłości połączeń, ale zaiste nie oznacza to sprawdzenia braku ciągłości tam gdzie nie powinno jej być. Dokonano dokładnej inspekcji obu sygnałów pod mikroskopem w obu warstwach płytki. Niczego podejrzane nie znaleziono. W tym momencie było wiadome, że nie jest to oczywiste uszkodzenie, co więcej zwarcie może być zlokalizowane na wewnętrznych warstwach płytki. Zdecydowano się na radykalne kroki — podłączenie obu ścieżek do zasilacza w trybie stałego prądu i obserwację układu pod kamerą termowizyjną celem zlokalizowania zwarcia. Po włączeniu zasilania pod wpływem przyłożonego napięcia ok. 2V, natężenie płynącego prądu sięgało 3A. Po kilku sekundach, zanim zaobserwowano jakiekolwiek zmiany w kamerze termowizyjnej, prąd spadł do 0A. Sytuację zweryfikowano jeszcze omówieniem — rezystancja była większa od zakresu multimetru. W ten oto sposób udało się naprawić płytę poprzez przepalenie zwierającego elementu. Niestety nie udało się ustalić, w którym miejscu dokładnie było uszkodzenie. Przypuszcza się, że mógł to być opilek miedzi pochodzący z etapu wiercenia otworów pod przelotki między warstwami, co więcej znajdujący się któryś z wewnętrznych warstw. Na tym etapie przylutowano z powrotem usunięte wcześniej komponenty.

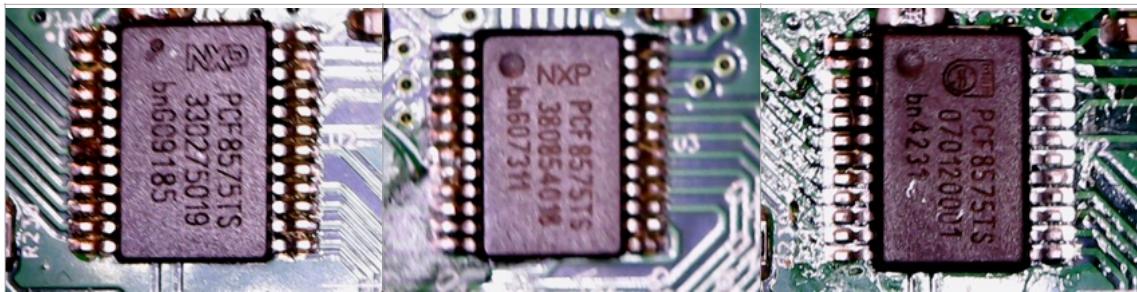
Następnie ponownie próbowało uruchamiać komunikację na linii I2C, tym razem zdecydowano się na próby wykrycia układów małymi krokami, używając programu I2C detect z biblioteki firmy Adafruit [22]. Program wymagał jedynie podania adresu układu i uruchomienia monitora portu szeregowego w Arduino iDE. Niestety układ pod adresem 0x20 wciąż nie odpowiadał.



Rys. 4.14.5 Przechwycona ramka za pomocą analizatora stanów logicznych

Źródło: opracowania własne

Przy użyciu multimetru stwierdzono poprawne napięcie 3,3 V na liniach SDA i SDL. Użyto więc następnie analizatora sygnałów logicznych celem sprawdzenia przesyłanej wiadomości. Przechwyconą ramkę przedstawiono na rysunku 4.14.5 — sygnał czerwony to linia SDA, a pomarańczowa to SCL. Widać poprawnie przeslaną wiadomość SETUP oraz brak potwierdzenia poprawnego odebrania od układu o adresie 0x20 w ostatnim takcie zegarowym. Sprawdzono postać sygnałów w trybie analogowym (oscyloskopowym) -- nie były one zniekształcone, ani nie występowały żadne problemy z ich integralnością. Ten fakt pozwolił wykluczyć wiele błędnych hipotez. Kluczowe w tym momencie okazało się pochodzenie układów.



Rys. 4.14.6 Układy PCF8575 pod mikroskopem
Źródło: opracowania własne

Na rysunku 4.14.6 przedstawiono trzy różne wersje układu PCF8575. Wszystkie trzy sprzedane jako układ PCF8575. Ale jak się później okazało, tylko jeden z nich jest prawdziwym układem PCF8575. Na pewnym etapie sprawdzono możliwości sterowania poszczególnymi wyjściami, pomimo braku odpowiedzi od układu (co sugerowałyby że nie działa). Jak się okazało wszystkie trzy poprawnie sterują wyjściami i odpowiadają na ramki sterujące. Ale tylko jeden z nich odpowiada na wiadomość SETUP. Jest to układ środkowy z rysunku 4.14.6. Pozostałe dwa są podróbkami. Po dokładnym obejrzeniu modułów, można stwierdzić różny sposób naniesienia oznaczeń. W układzie oryginalnym są one wypalone na obudowie laserem. W układzie nieoryginalnym, pokazanym po lewej stronie rysunku, zastosowano technikę mikrofrezowania. W dodatku inaczej zaprojektowano logo oraz inaczej wykonano wgłębienie wskazujące na pin nr 1.

W przypadku układu z prawej strony rysunku czujność powinna wzbiudzić dostępność modułu opatrzonego logiem Philipsa w 2023 r. Mianowicie firma Philips sprzedała swój dział projektujący półprzewodniki w 2006 roku [22], co oznacza że od tego czasu nie produkuje układów scalonych ze swoim logo. Szanse że po 17 latach od tego wydarzenia ktoś odnalazł potężne zasoby zaledgających w magazynach układów PCF8575 są raczej bliskie zera. Prawdopodobnie jest to więc również układ podrobiony. Niestety, podrabianie układów scalonych jest oębnie szeroko rozpowszechnionym procederem. Wiele układów TTL, prostych bramek logicznych, przerzutników, czy jak w tym przypadku eksplanderów jest wykonywanych na mikrokontrolerach, zaprogramowanych tak by wiernie pełnić funkcję danego układu. Niestety często nie możliwe jest odwzorowanie funkcjonowania układu w stu procentach, co jak w przypadku niniejszej pracy objawia się brakiem obsługi nie których wiadomości. Zważywszy na fakt że układy spełniają swoją zasadniczą funkcję — sterowanie wyjściami lub odczytywanie wejść, pozostały podrobione układy na płytce.

Kolejnym błędem, który uniemożliwiał uruchomienie układu był, jak się okazało błąd w oprogramowaniu. Mianowicie w pętli programu każdy port GPIO był ustawiany jako wyjście i następnie naprzemiennie ustawiano na nim stan niski albo wysoki. Dotyczy to również portów do obsługi magistrali I2C. Powodowało to prawdopodobnie wewnętrzne zwarcie w układzie między blokiem do obsługi I2C a tym obsługującym wyjście cyfrowe. Konsekwencją były obserwowane pętle resetowania programu, występujące niekiedy zaraz po zaraz po włączeniu zasilania, a czasami po dłuższej pracy. Powodem tego błędu był programista piszący program testowy. Niestety był jeszcze jeden program programistyczny, ale tym razem spowodowany błędem programisty piszącego bibliotekę do obsługi płytka RPI Pico w Arduino iDE. Gdy GPIO nr 22 ustawia się jako wyjście układ wpada w pętle resetu, znaleziono jeszcze jeden błąd programistyczny. Niestety tego problemu w ostateczności nie udało się rozwiązać. Podejrzenie pada na osobę tworzącą bibliotekę do programu Arduino iDE — jedyną poszlaką jest fakt, że port GPIO22 może być ustawione jako pin do obsługi magistrali I2C z tego samego bloku co GPIO11 (I2C1). Potencjalnie dochodzi do zwarcia pinów GPIO11 i GPIO22, a w konsekwencji do resetu układu.

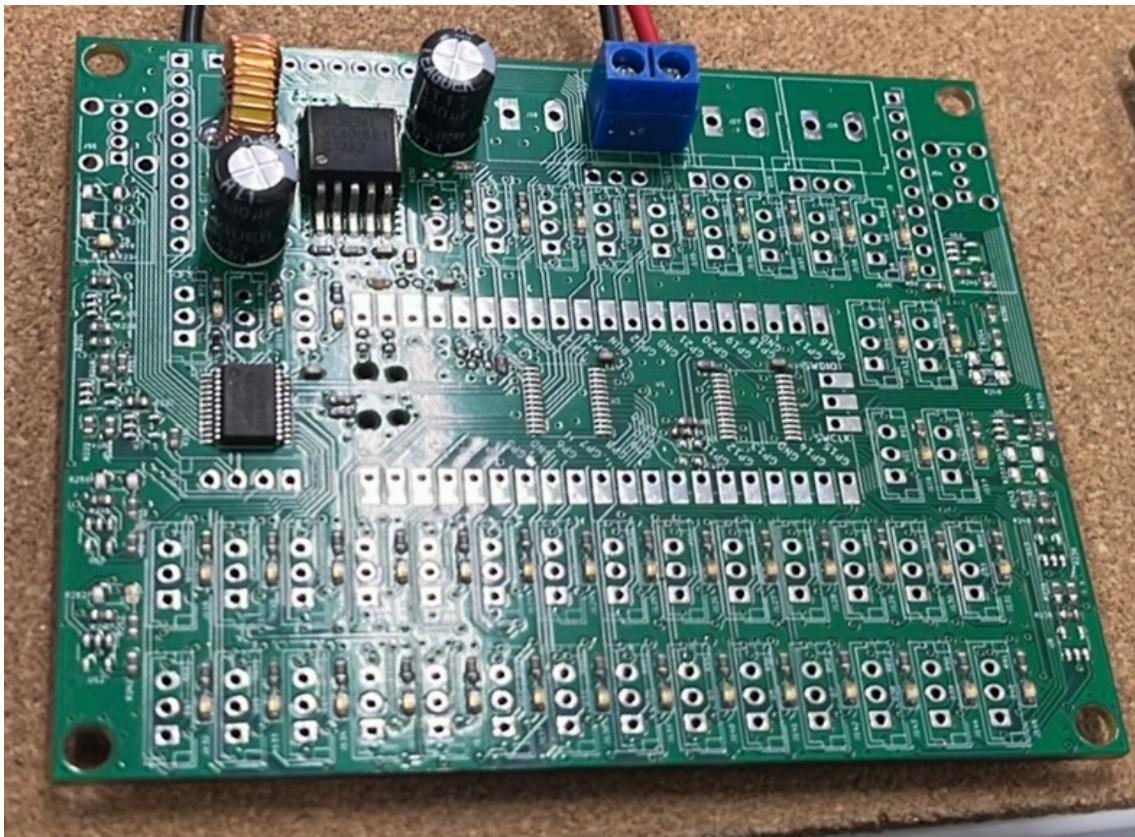
4.15. Testy

Testy obejmują pracę w dłuższym okresie obserwacji oraz analizę termiki poszczególnych komponentów, jak i całego układu.

Jak opisano we wcześniejszych rozdziałach, występowały problemy ze stabilnością układu — zawieszanie po losowym czasie pracy, spowodowane błędami w oprogramowaniu. Po ich usunięciu, układ pozostawiono na ponad 24 godziny pracy -- w stanie jałowym, bez podłączonych żadnych urządzeń peryferyjnych -- po tym czasie dalej funkcjonował. Wielokrotnie odłączano i podłączano zasilanie w dowolnym momencie pracy — w trakcie trwania testów układ zawsze się uruchamiał.

W następnej kolejności przebadano warunki pracy przetwornic napięcia. Już na wstępny etapie uruchamiania sprawdzono poprawność napięć za pomocą multimetru oraz sprawdzono jaki kształt ma sygnał wyjściowy na oscyloskopie. Następne testy wykonano pod obciążeniem, za pomocą aktywnego obciążenia, obserwując układ poprzez kamerę termowizyjną. Przebadano dwa układy: przetwarzania napięcia od 12 V do 5 V oraz stabilizacji od 12 V do 3,3 V.

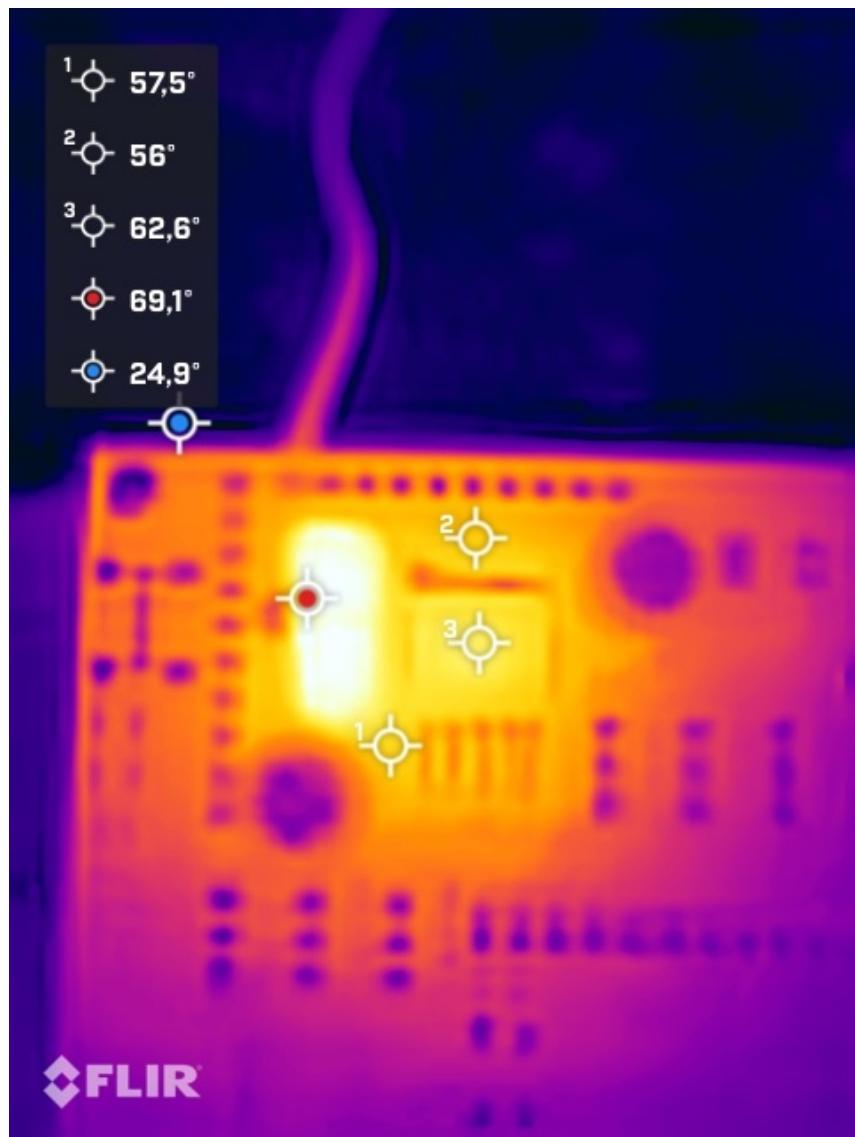
W pierwszym teście termicznym obciążono układ XL4015 obciążeniem wynoszącym 15W. Układ był badany po zakończeniu montażu SMT, w stanie przedstawionym na rys. 4.15.1.



Rys. 4.15.1 Stan układu w trakcie testów

Źródło: opracowania własne

Obraz zarejestrowany przez kamerę termowizyjną przedstawiony jest na rys. 4.15.2.

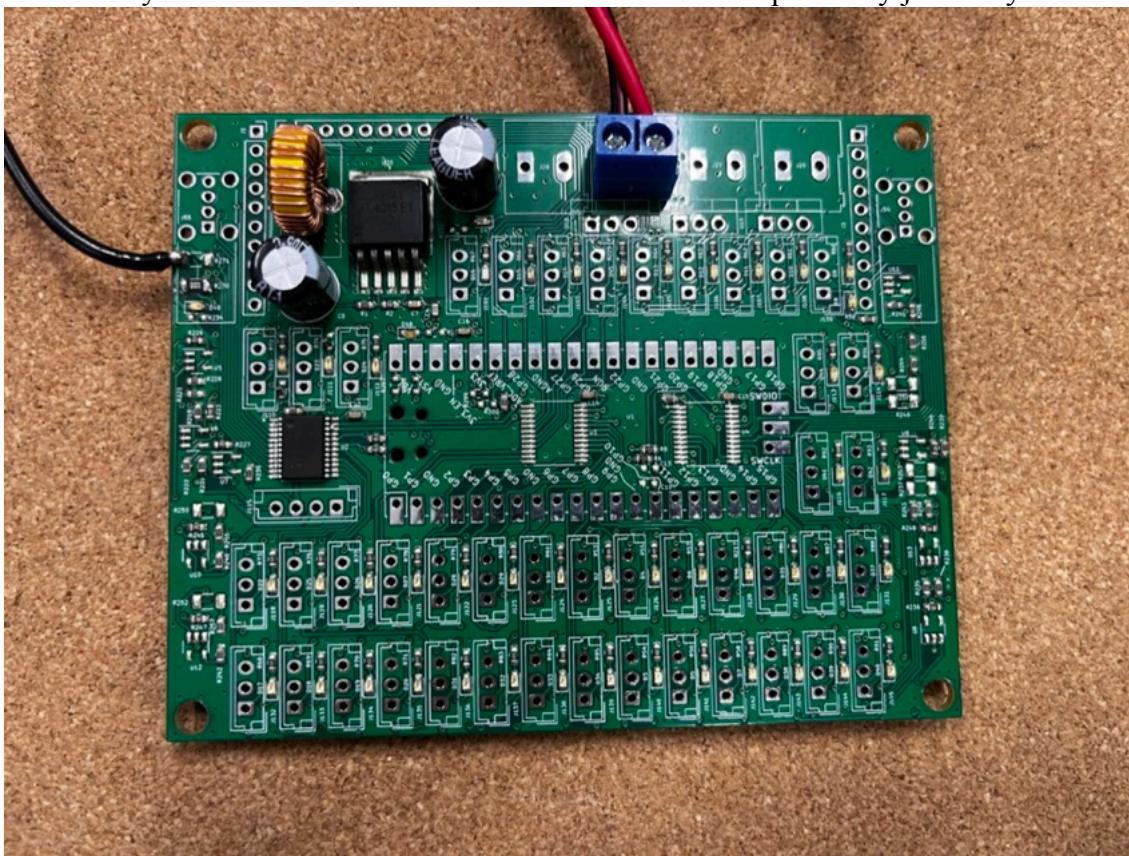


Rys. 4.15.2 Obraz termalny układu pod 15 watowym obciążeniem — front płytki

Źródło: opracowania własne

Dopuszczalna temperatura pracy dla tego typu układu wynosi 125 stopni Celsjusza [16]. Można zaobserwować, że żaden z komponentów nie nagrzewa się do niebezpiecznych temperatur. Widać jedynie, że cewka grzeje się zaskakująco mocno. Zwykle źródłem wyższej temperatury jest główny układ przetwornicy DCDC, a nie element pasywny, jakim jest cewka. W przyszłości należałoby rozważyć użycie modelu w technologii montażu SMD, który mógłby oddawać ciepło w powierzchnię laminatu, lub należałoby znaleźć cewkę o niższej rezystancji. Modyfikacja komponentów nie jest konieczna, ponieważ parametry pracy układu mieszczą się w wyznaczonych limitach.

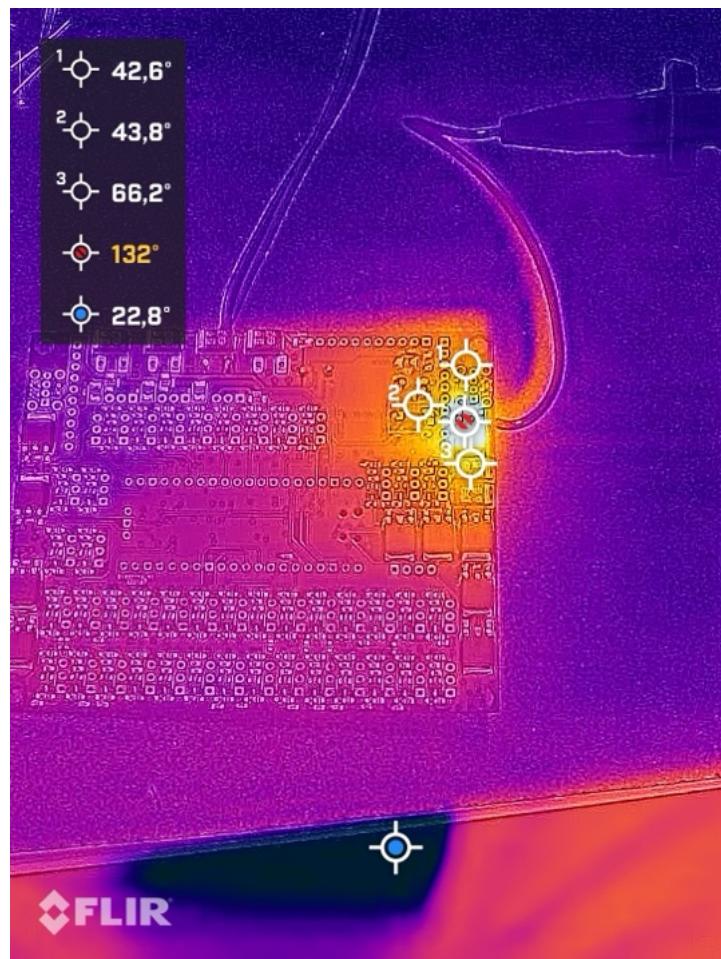
W kolejnym teście przebadano stabilizator liniowy. Układ również został zbadany po zakończonym montażu SMT. Stan układu w trakcie testów pokazany jest na rys. 4.15.3.



Rys. 4.15.3 Stan układu w trakcie testów

Źródło: opracowania własne

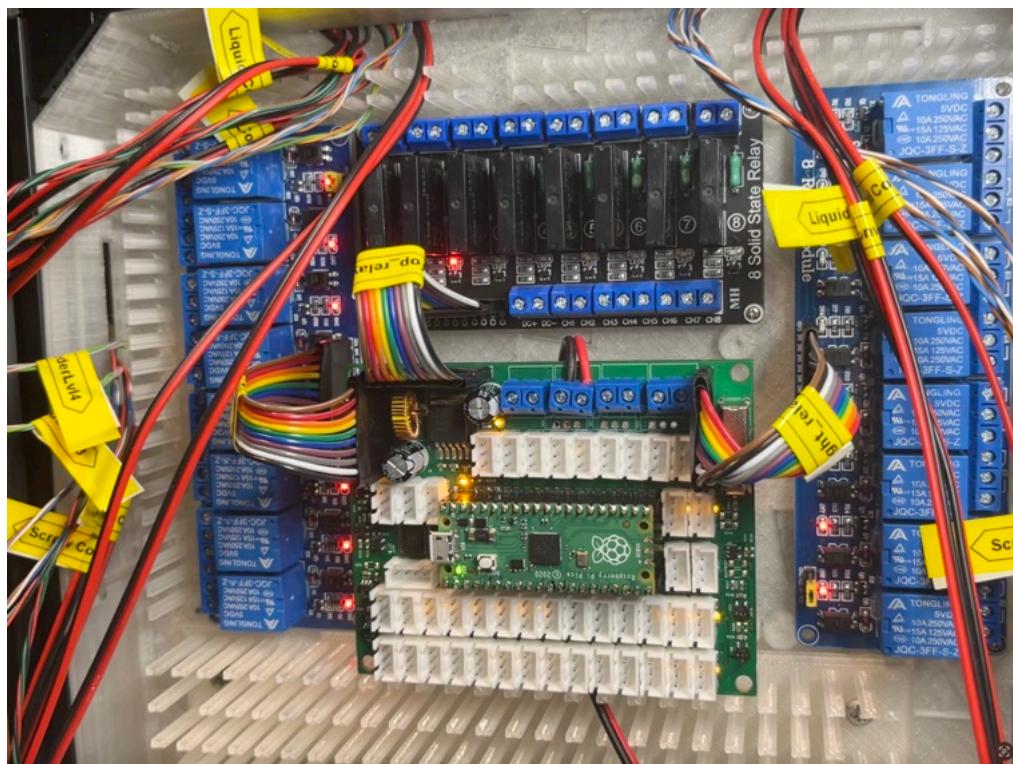
Jak można zaobserwować na rys. 4.15.4, układ znacznie się grzeje. Jest to normalne dla stabilizatorów liniowych. Różnica napięć pomiędzy wejściem a wyjściem to aż 8,7 V. Należy zaznaczyć, że został przebadany ekstremalny przypadek, kiedy obciążenie jest znacznie większe niż to, które faktycznie będzie występować w trakcie pracy układu. Obciążenie w trakcie pracy, jest znaczco mniejsze. RPI Pico ma wbudowany stabilizator od 5 V do 3,3 V, więc układ NCP1117, będzie jedynie zasilał czujniki, których pobór prądu jest niewielki.



Rys. 4.15.4 Obraz termalny układu pod 5 watowym obciążeniem — tył płytki

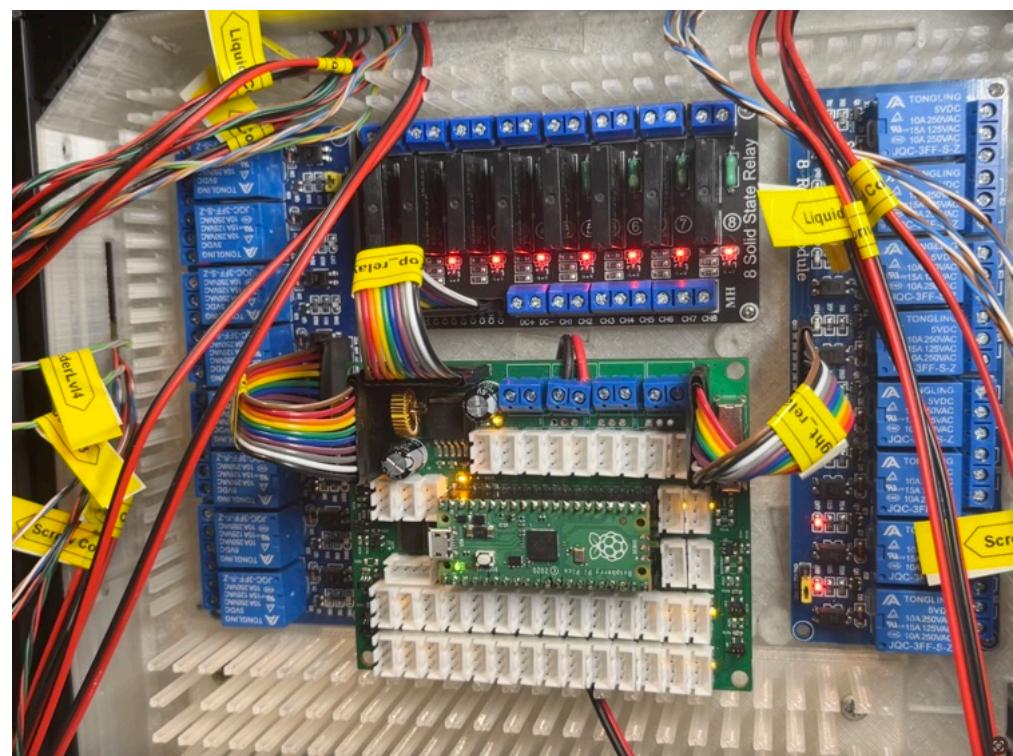
Źródło: opracowania własne

Po zakończeniu montażu SMT i THT oraz uruchamiania płytki, sprawdzono zachowanie całego układu w docelowych warunkach pracy, tj. w ekspresie do herbaty. Podłączono listwę LED oraz wszystkie moduły przekaźników. Uruchomiono program testowy. Zweryfikowano poprawność działania, tj. wszystkie przekaźniki naprzemiennie łączą się i rozłączają, co zarejestrowano na rys. 4.15.5 oraz 4.15.6.



Rys. 4.15.5 Układ w docelowym zastosowaniu w trakcie pracy 1

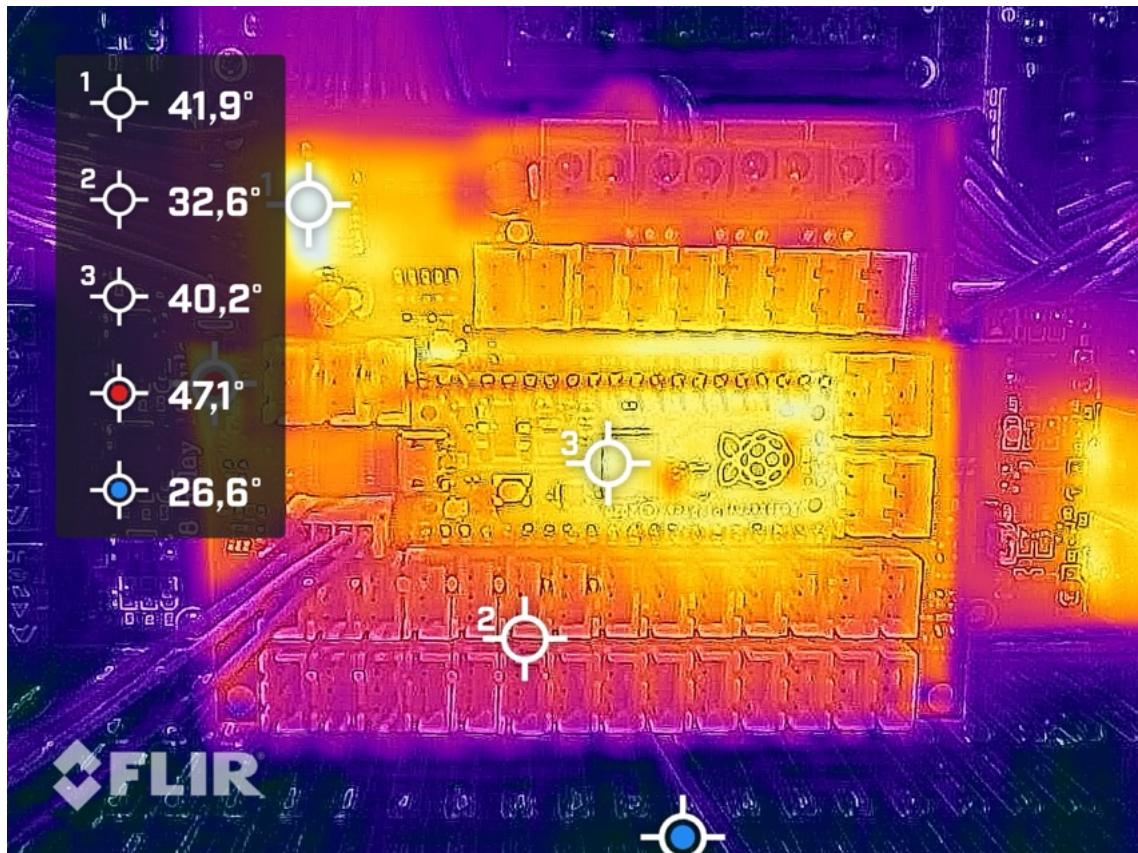
Źródło: opracowania własne



Rys. 4.15.6 Układ w docelowym zastosowaniu w trakcie pracy 2

Źródło: opracowania własne

Następnie za pomocą kamery termowizyjnej zbadano termikę układu, czego wyniki można zaobserwować na rys. 4.15.7



Rys. 4.15.7 Układ w docelowym zastosowaniu w trakcie pracy — obraz termowizyjny

Źródło: opracowania własne

Jak można zauważyć głównymi źródłami ciepła są przetwornice napięcia w lewym górnym rogu oraz przy lewej krawędzi płytki, układ RP2040 oraz ekspandery. W żadnym miejscu nie zaobserwowano skrajnie wysokich temperatur, które mogłyby zagrażać bezpiecznemu funkcjonowaniu układu.

5. Oprogramowanie układowe

Celem przetestowania układu przygotowano program, którego celem jest skonfigurowanie wszystkich pinów RPI Pico (poza GPIO10 i GPIO11 odpowiedzialnymi za komunikację I2C) jako wyjścia i ustawianie różnych sekwencji wyjściowych, które w połączeniu z zamontowanymi diodami LED umożliwiają weryfikację działania każdego z wyjść. Podobnie wszystkie sygnały na ekspanderach PCF8575 wszystkie piny ustawiane są jako wyjścia i chwilowo ściągane są do poziomu masy.

Program został napisany w darmowym środowisku Arduino IDE, przedstawionym na rys. 5.1. Na rys. 5.2 przedstawiony jest kod użyty do testów płytki, a na rys. 5.3 i 5.4 przykładowe momenty w trakcie pracy programu.

The screenshot shows the Arduino IDE interface with the title bar "TeaMachineRev1BRD_SelfTest_V1.2.0 | Arduino 1.8.19". The code editor contains the following C++ code:

```
TeaMachineRev1BRD_SelfTest_V1.2.0
Serial.println("PCFs low");
delay(100);
for (uint8_t p=0; p<16; p++) {
    pcf.digitalWrite(p, HIGH); // turn LED off by turning off sinking transistor
}
Serial.println("PCFs high");
delay(100);

pcf.begin(0x21, &Wire);
    for (uint8_t p=0; p<16; p++) {
        pcf.digitalWrite(p, LOW); // turn LED on by sinking current to ground
    }
    Serial.println("PCFs low");
delay(100);
for (uint8_t p=0; p<16; p++) {
    pcf.digitalWrite(p, HIGH); // turn LED off by turning off sinking transistor
}
Serial.println("PCFs high");

pcf.begin(0x23, &Wire);
    for (uint8_t p=0; p<16; p++) {
        pcf.digitalWrite(p, LOW); // turn LED on by sinking current to ground
    }
    Serial.println("PCFs low");
delay(100);
for (uint8_t p=0; p<16; p++) {
    pcf.digitalWrite(p, HIGH); // turn LED off by turning off sinking transistor
}
Serial.println("PCFs high");

Done Saving.

Sketch uses 97302 bytes (4%) of program storage space. Maximum is 2097152 bytes.
Global variables use 43760 bytes (16%) of dynamic memory, leaving 226576 bytes for local variables. Maximum is 270336
Forcing reset using 1200bps open/close on port /dev/cu.usbmodem1101
/Users/LukaszNowarkiewicz/Library/Arduino15/packages/arduino/tools/rp2040tools/1.0.6/rp2040load -v -D /var/folders/_k
rp2040load 1.0.6 - compiled with go1.16.2
Loading into Flash: [=====] 100%
```

The status bar at the bottom right indicates "Raspberry Pi Pico on /dev/cu.usbmodem1101".

Rys. 5.1 Środowisko Arduino iDE z otwartym programem testowym

Źródło: opracowanie własne

```
#include "Arduino.h"
#include "Adafruit_PCF8575.h"
Adafruit_PCF8575 pcf;
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, HIGH);
    for ( uint8_t i = 0; i<=28 ; i++) {
        if(i!=10 && i!=11 && i!=22){
            pinMode(i, OUTPUT);
        }
    }
    pinMode(22, INPUT);
    delay(2000);
    Serial.begin(115200);
    Serial.println("Adafruit PCF8575 LED blink test");

    if (!pcf.begin(0x20, &Wire)) {
        Serial.println("Couldn't find PCF8575");
    }
    for (uint8_t p=0; p<16; p++) {
        pcf.pinMode(p, OUTPUT);
    }
    pcf.begin(0x21, &Wire);
    for (uint8_t p=0; p<16; p++) {
        pcf.pinMode(p, OUTPUT);
    }
    pcf.begin(0x23, &Wire);
    for (uint8_t p=0; p<16; p++) {
        pcf.pinMode(p, OUTPUT);
    }

    Serial.println("Config done");
    delay(100);
}

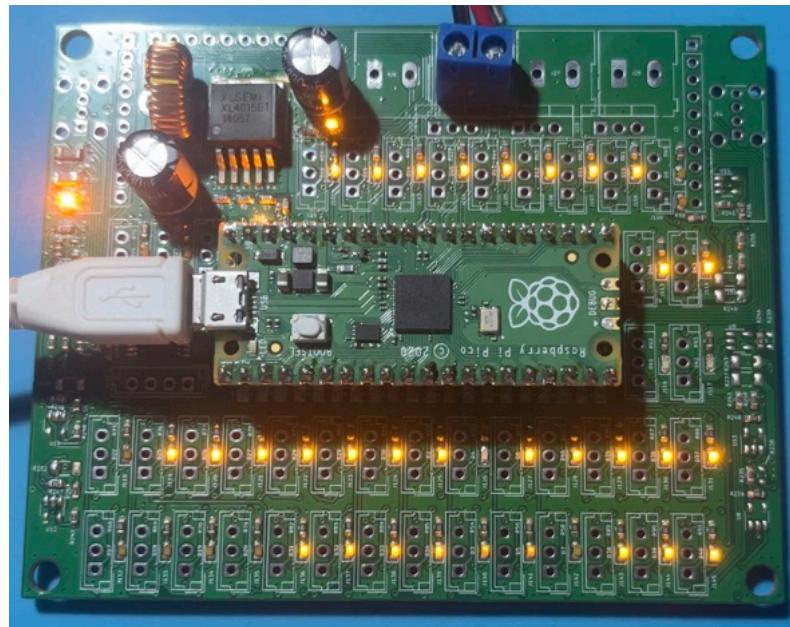
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
    digitalWrite(LED_BUILTIN, HIGH);
    Serial.println("LED_BUILTIN, HIGH");
    delay(100);
    pcf.begin(0x20, &Wire);
    for (uint8_t p=0; p<16; p++) {
        pcf.digitalWrite(p, LOW);
    }
    Serial.println("PCFs low");
    delay(100);
    for (uint8_t p=0; p<16; p++) {
        pcf.digitalWrite(p, HIGH);
    }
    Serial.println("PCFs high");
    delay(100);

    pcf.begin(0x21, &Wire);
    for (uint8_t p=0; p<16; p++) {
        pcf.digitalWrite(p, LOW);
    }
    Serial.println("PCFs low");
    delay(100);
    for (uint8_t p=0; p<16; p++) {
        pcf.digitalWrite(p, HIGH);
    }
    Serial.println("PCFs high");

    digitalWrite(LED_BUILTIN, LOW);
    for(uint8_t i=0; i<28 ; i++){
        if(i!=10 && i!=11 && i!=22){
            digitalWrite(i, HIGH);
        }
    }
    delay(100);
    if(i!=10 && i!=11 && i!=22){
        digitalWrite(i, LOW);
    }
}
```

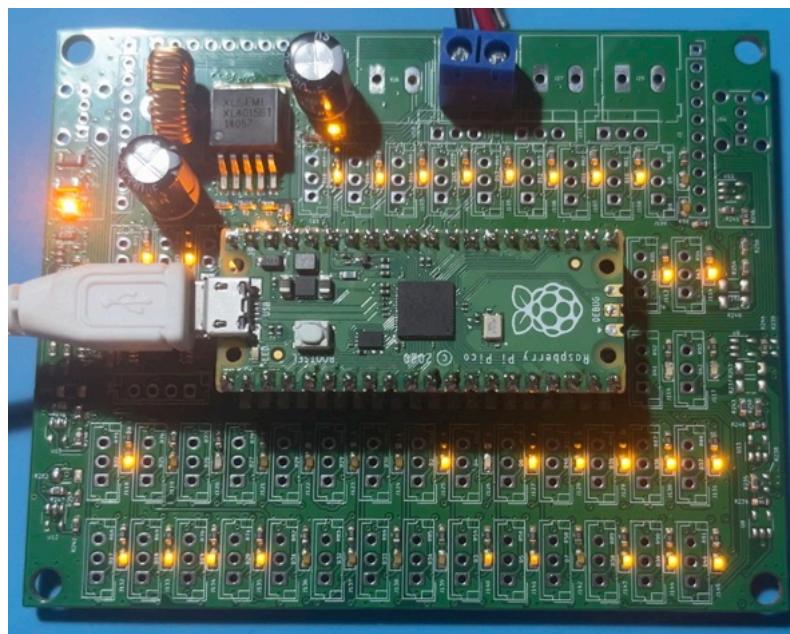
Rys. 5.2 — Kod programu sterującego mikrokontrolerem RPI Pico

Źródło: opracowanie własne



Rys. 5.3 Program w trakcie działania w fazie 1

Źródło: opracowanie własne



Rys. 5.4 Program w trakcie działania w fazie 2

Źródło: opracowanie własne

6. Wnioski i dalsze prace

Podsumowując kolejne etapy projektu takie jak: zaprojektowanie płytki sterującej ekspresem do herbaty, jej wykonanie, uruchomienie oraz testy, zakończony projekt można określić jako bardzo udany. Wszystkie elementy układu działają. Nie napotkano żadnych problemów, które dyskwalifikowałyby układ w jego użyciu. Nie ujawniły się błędy, które determinowałyby konieczność wykonania kolejnej wersji układu.

Na podstawie własnych doświadczeń, autor rekomenduje ostrożnie wybierać dostawców komponentów, ponieważ podrabiane układy mogą znacznie wydłużyć i utrudnić proces uruchamiania płytki oraz wykonania całego projektu. Układ należy uruchamiać etapami. Należałoby zweryfikować poprawność działania poszczególnych bloków w inny sposób i — jeżeli to nie jest konieczne — nie łączyć ich między sobą (np. przetwornicy z zasilaniem całej płytki, mikrokontrolera z jego wyjściami). Taki proces postępowania pozwala na etapową weryfikację poprawności działania i uniknięcia potencjalnych uszkodzeń bloków przez jeden błędnie funkcjonujący blok. Warto obserwować układ pod kamerą termowizyjną w celu poszukiwania potencjalnych błędów. Ułatwia ona w znacznym stopniu analizę termowizyjną i pozwala na wykrycie komponentów, które nie spełniają norm związanych z wysokością temperatur podczas pracy.

Rozwiązaniem, które mogłoby ulepszyć działanie układu, to między innymi umieszczenie przetwornic w środkowej części płytki. Pozwoliłoby to na równomierne rozpraszanie ciepła na całej powierzchni.

Należałoby wybrać układy, które (tak jak pad termiczny), mają wyprowadzone sygnały masy. Taki wariant wykonania znacznie poprawiłby odprowadzanie ciepła. W przyszłości należałoby zmniejszyć ilość pasty do lutowania, skrócić czas lutowania rozpływowego.

7. Bibliografia

- [1] Horowitz Paul , Hill Winfield: Sztuka elektroniki
- [2] Shawn Wallace: Płytki drukowane (PCB)
- [3] Joseph J. Carr: Zasilacze do urządzeń elektronicznych
- [4] Praca zbiorowa, wydawnictwo REA: Poradnik mechatronika ISBN: 978-83-7993-537-6
- [5] Ödön Ferenczi: Zasilanie układów elektronicznych. Zasilacze impulsowe. Warszawa: WNT, 1989, s. 77–95
- [6] Jay Geier: Jak naprawić sprzęt elektroniczny
- [7] Sebastian Kiek: Diagnostyka i naprawa płyt głównych laptopów
- [8] Dokumentacja programu Kicad (Dostęp zdalny: 01.01.2023) — <https://docs.kicad.org/>
- [9] Kurs wideo do programu Kicad (Dostęp zdalny: 01.12.2022)- <https://www.youtube.com/user/contextualelectronic/playlists>
- [10] Diagnozowanie usterek w sprzęcie elektronicznym (Dostęp zdalny: 01.12.2022) — <https://www.youtube.com/@rossmanngroup/playlists>
- [11] Raspberry Pi Pico dokumentacja (Dostęp zdalny: 19.01.2023) — <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>
- [12] Programowanie mikrokontrolerów rodziny Raspberry Pi w językach C/C++ (Dostęp zdalny: 19.01.2023) — <https://datasheets.raspberrypi.com/pico/getting-started-with-pico.pdf>
- [13] Projektowanie płytki pod układy RP2040 (Dostęp zdalny: 19.01.2023) — <https://datasheets.raspberrypi.com/rp2040/hardware-design-with-rp2040.pdf>
- [14] Dokumentacja układu RP2040 (Dostęp zdalny: 19.01.2023) — <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>
- [15] Dokumentacja ekspandera I2C PCF8575 (Dostęp zdalny: 21.01.2023) — <https://www.nxp.com/docs/en/data-sheet/PCF8575.pdf>
- [16] Dokumentacja układu przetwornicy buck XL4015 (Dostęp zdalny: 19.01.2023) — https://datasheet.lcsc.com/szlcsc/1811081616_XLSEMI-XL4015E1_C51661.pdf
- [17] Dokumentacja stabilizatorów liniowych serii 78** (Dostęp zdalny: 22.01.2023) — <https://www.sparkfun.com/datasheets/Components/LM7805.pdf>
- [18] Dokumentacja modułu przekaźników (Dostęp zdalny: 26.01.2023) — http://wiki.sunfounder.cc/index.php?title=2_Channel_5_V_Relay_Module
- [19] Dokumentacja standardu I2C (Dostęp zdalny: 14.01.2023) — <https://www.pololu.com/file/0J435/UM10204.pdf>
- [20] Ładowanie urządzeń USB (Dostęp zdalny: 11.01.2023) — <https://obddiag.net/usb-power.html>
- [21] Opis biblioteki do obsługi ekspanderów PCF8575 (Dostęp zdalny: 06.01.2023) — https://github.com/adafruit/Adafruit_PCF8574
- [22] informacja o sprzedaniu firmy Philips (Dostęp zdalny: 19.01.2023) — <https://www.eetimes.com/phillips-sells-semiconductor-unit-to-equity-firms>