

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Technologie Informacyjne
w Systemach Automatyki (ART)

**PRACA DYPLOMOWA
INŻYNIERSKA**

System lokalizacji samolotów z wykorzystaniem
ADS-B

Airplane tracking system using ADS-B

AUTOR:
Karol Szpila

PROWADZĄCY PRACĘ:

dr inż. Krzysztof Halawa

Katedra Informatyki Technicznej

OCENA PRACY:

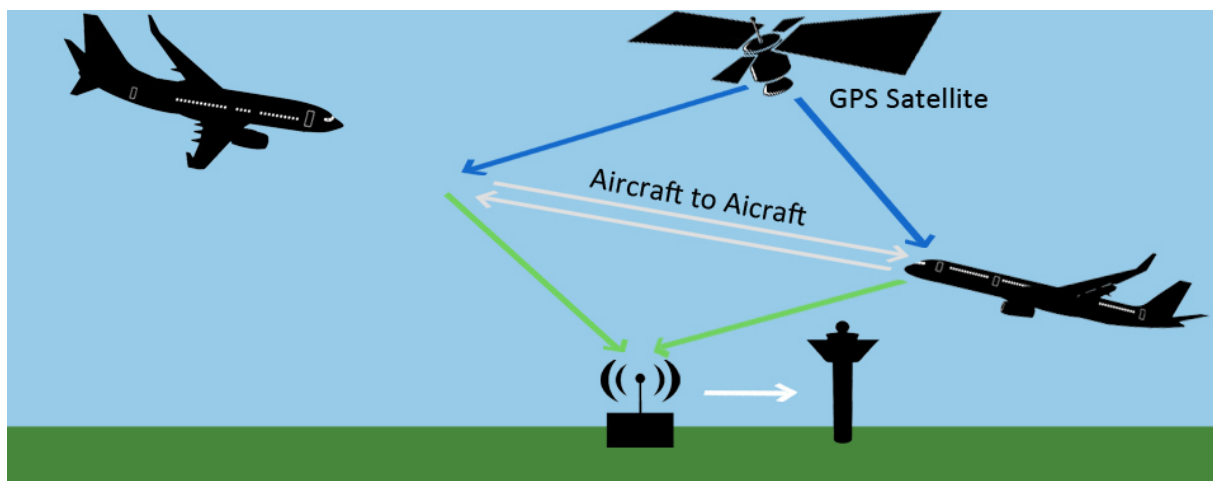
Spis treści

1	Wstęp	2
2	Cele i założenia projektowe	3
3	Realizacja projektu	4
3.1	Schemat Urządzenia	4
3.1.1	Zasilanie	4
3.1.2	Mikrokontroler	5
3.1.3	Interfejsy komunikacyjne	6
3.1.4	Wyświetlacz	6
3.1.5	SDRAM	7
3.1.6	Moduł GPS	8
3.2	Wykonanie PCB	9
3.2.1	Parametry PCB	9
3.2.2	GPS	9
3.2.3	USB	11
3.2.4	Interfejsy szybkie	13
3.2.5	Warstwa zasilania	15
3.2.6	Prezentacja PCB	16
3.3	Warstwa Programowa	20
3.3.1	Model UML	20
3.3.2	System Operacyjny	20
4	Podsumowanie	21
	Bibliografia	21

Rozdział 1

Wstęp

ADS-B jest systemem śledzenia położenia statków powietrznych mający uzupełniać pracę PSR (eng. Primary Surveillance Radar). PSR to radar aktywny bazujący na wysyłaniu fal elektromagnetycznych oraz pomiarze czasu powrotu fali odbitej od obiektu. Wadą takich systemów jest brak informacji o wykrytym obiekcie poza jego lokalizacją i rozmiarem. ADS-B polega na automatycznym wysyłaniu pozycji statku powietrznego określonego przy pomocy pokładowego systemu GPS w przestrzeń, tak że zarówno kontrola naziemna jak i pobliskie samoloty są w stanie odebrać tę wiadomość.



Rysunek 1.1 Schemat podłączenia mikrokontrolera STM32f767ZIT6

Rozdział 2

Cele i założenia projektowe

Celem niniejszej pracy było zbudowanie prototypu systemu pozwalającego na lokalizację oraz zbieranie informacji o statkach powietrznych wyposażonych w transpondery ADS-B. System zostanie zbudowany w oparciu o mikrokontroler z rodziny STM32F7 ze względu na parametry sprzętowe i peryferia. Jako radio do odbierania wiadomości ADS-B zostanie wykorzystany tuner DVB-T USB ze względu na niską cenę oraz gotowe biblioteki pozwalające łatwo przekształcić urządzenie w SDR (ang. Software Defined Radio). SDR to programowalne radio które można dostroić do odbierania zadanej częstotliwości bez ingerencji w sprzęt. Wykorzystane zostaną do tego biblioteki STM32 USB Host i rtl_sdr. Urządzenie zostanie wyposażone w wyświetlacz prezentując zebrane dane jako obraz radaru z pozycją samolotu względem systemu oraz tabelą z informacjami dodatkowymi. Wykorzystana zostanie do tego biblioteka STemWin będąca dedykowaną wersją emWin dla mikrokontrolerów ST.

Rozdział 3

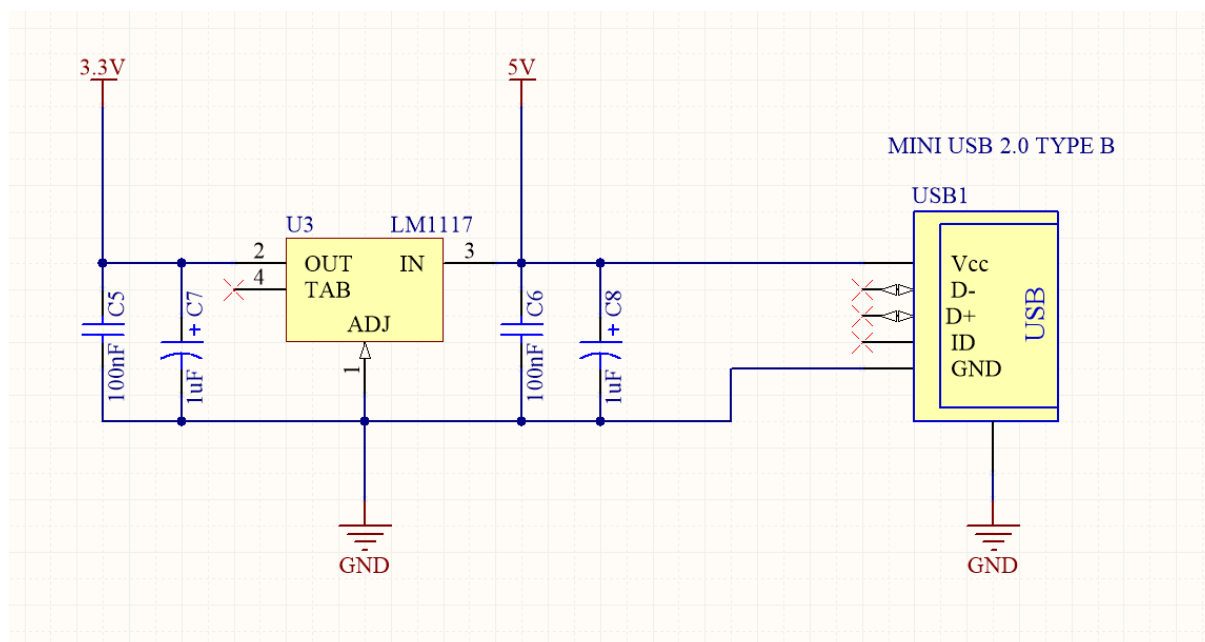
Realizacja projektu

3.1 Schemat Urządzenia

W tym rozdziale zostanie opisana część sprzętowa projektu, czyli schemat urządzenia oraz projekt PCB, wykorzystane elementy oraz zewnętrzne urządzenia.

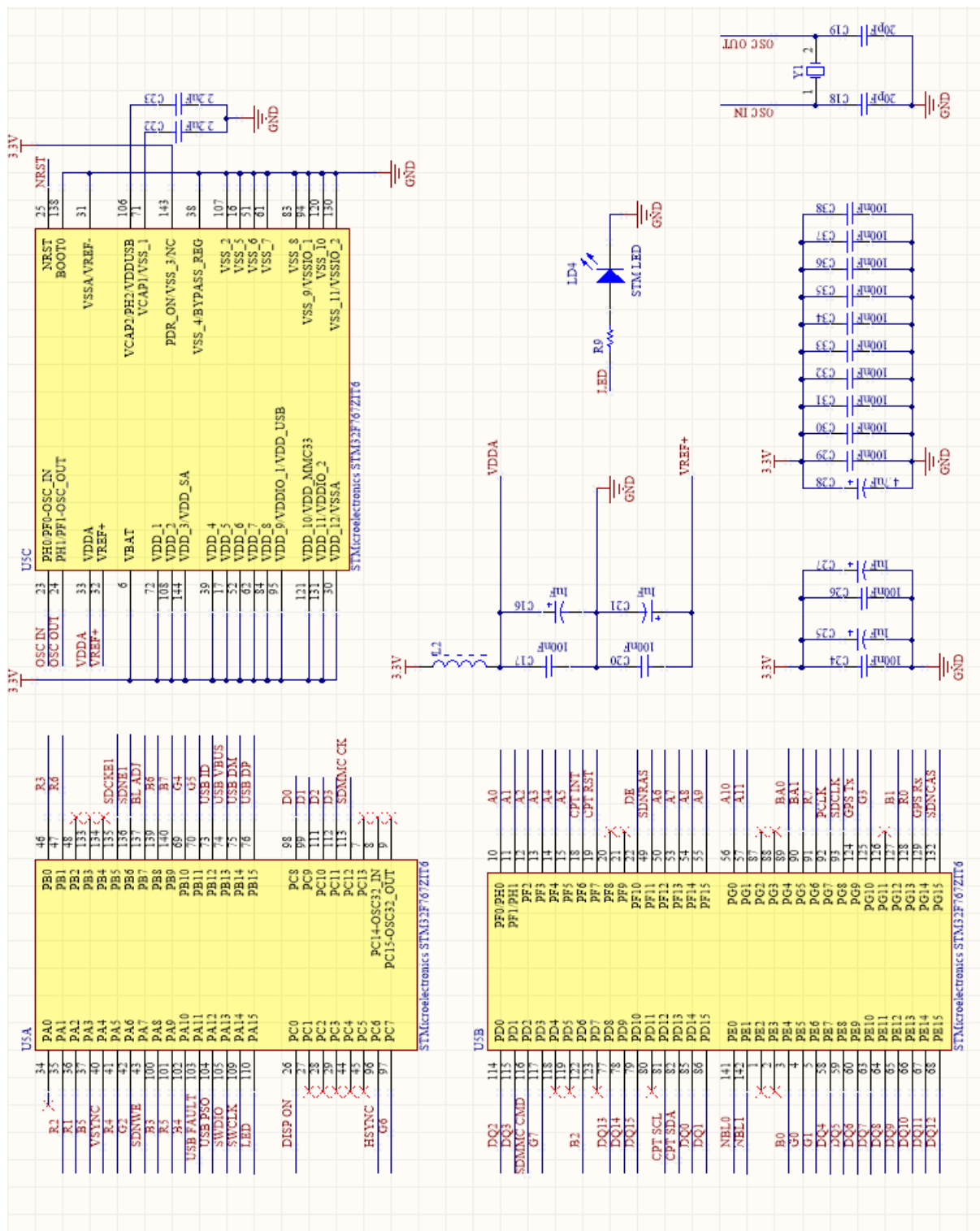
3.1.1 Zasilanie

Urządzenie jest zasilane z zewnętrznego źródła 5V poprzez port USB typu mini A. Pozwala to na podłączenie PCB zarówno do sieci przy pomocy ładowarki do telefonu jak i z komputera czy z przenośnego power banku. Ponieważ układy takie jak mikrokontroler czy pamięć SDRAM potrzebują napięcia 3.3V zastosowano stabilizator LM1117. Poniżej przedstawiono schemat podłączenia.



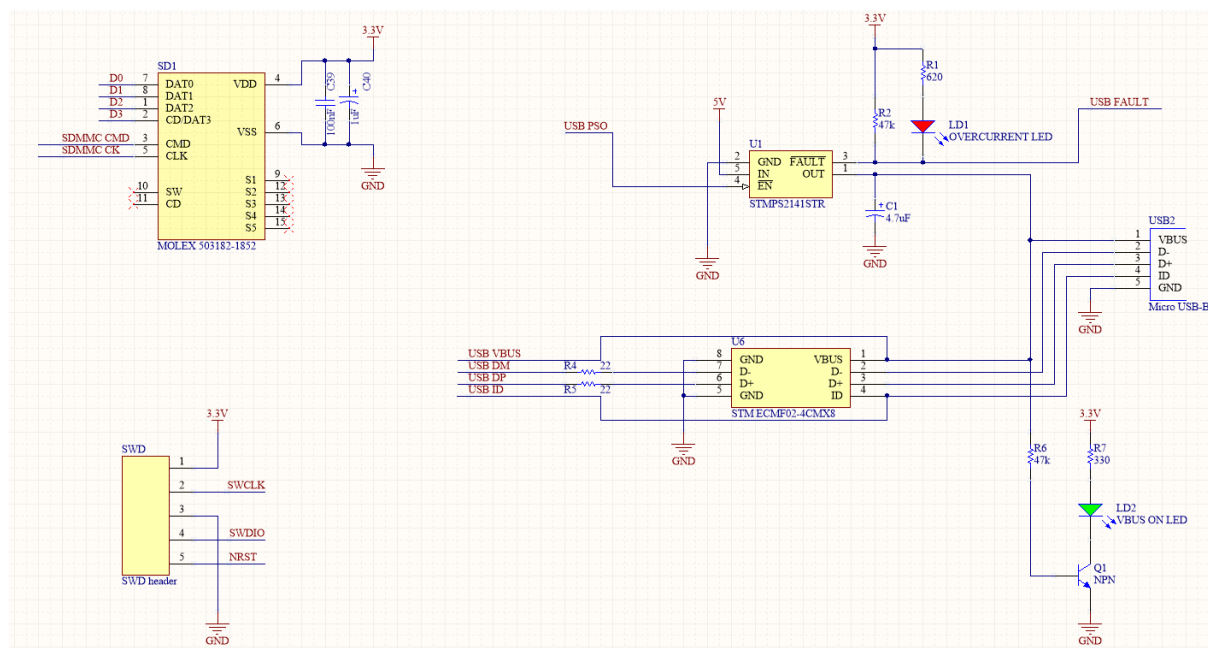
Rysunek 3.1 Schemat zasilania płytki PCB

Wykorzystany mikrokontroler to STM32F767ZIT6. Układ został wybrany ze względu na zmieszczenie w najmniejszej obudowie wszystkich wymaganych układów peryferiów takich jak: kontroler LDTC do sprzętowej obsługi wyświetlacza LCD, kontroler pozwalający obsłużyć zewnętrzną pamięć SDRAM i interfejsy komunikacyjne USB, I2C i UART. Poniżej przedstawiono schemat podłączenia.

Rysunek 3.2 *Schemat podłączenia mikrokontrolera STM32f767ZIT6*

3.1.3 Interfejsy komunikacyjne

System posiada gniazdo na kartę Micro SD, która może zostać wykorzystana do przechowywania skompresowanych obrazów do tła GUI lub co planowane jest w przyszłości map pozwalających lepiej orientować się w terenie na podstawie obrazu z radaru. Do programowania mikrokontrolera wykorzystany został dedykowany dwuprzewodowy interfejs SWD zgodny z ST-Link. Ostatni zostanie omówiony interfejs USB do komunikacji z SDR (ang. Software Defined Radio). Interfejs został wyprowadzony poprzez gniazdo Micro USB-B pozwalając w ten sposób zaoszczędzić miejsca na PCB. System jako Host USB będzie zasilac podłączony układy których pobór nie powinien przekroczyć, zgodnie ze standardem USB2.0 500mA, dlatego zastosowano power switch STMP2141STR. W przypadku podania stanu wysokiego na pin USB PSO switch zostanie włączony. Jeżeli nie występują żadne sytuacje nieporządne takie jak przetężenie prądowe czy zwarcie, zapali się zielona dioda sygnalizująca poprawnie działanie zasilania. W przypadku jakichkolwiek problemów prąd zostanie natychmiast odcięty a układ wystawi wysoki stan na pin FAULT informując mikrokontroler i zapalając czerwoną diodę. W celu zabezpieczenia interfejsu przed nieporzadanymi wyładowaniami elektrostatycznymi związanymi z dotykiem urządzenia czy wkładaniem urządzenia do gniazda zastosowano układ ochrony ESD (ang. Electro Static Discharge) STMECMF02-4CMX8 dedykowany dla USB2.0. Poniżej przedstawiono schemat połączenia na PCB.

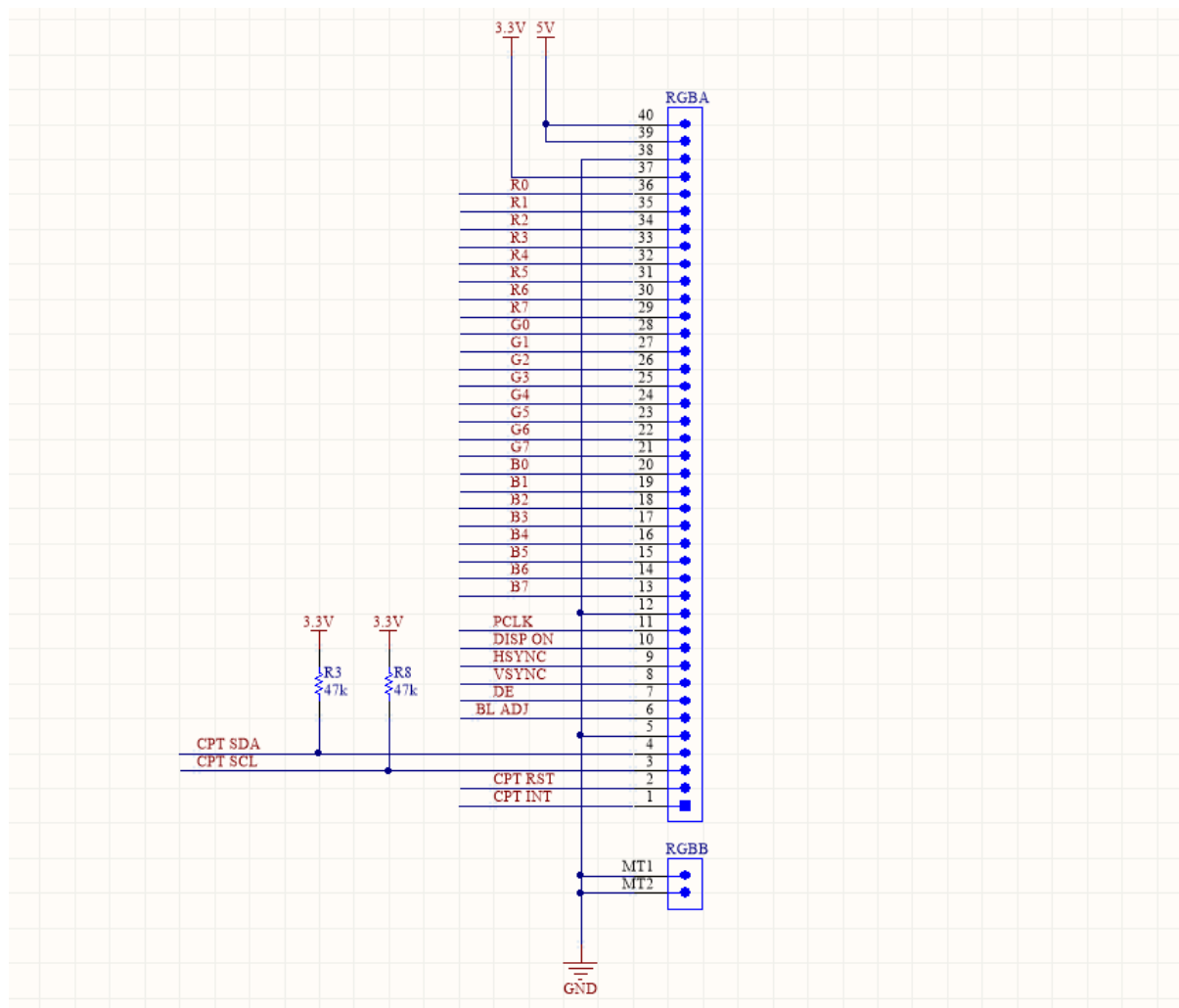


Rysunek 3.3 Schemat podłączenia zewnętrznych interfejsów

3.1.4 Wyświetlacz

Zgodnie z założeniem, system ma posiadać interfejs graficzny dla użytkownika. Do tego celu wybrano wyświetlacz HY101CTP o przekątnej 10,1" i rozdzielczości 1024 na 600 pikseli. Matryca jest sterowana poprzez MIPI-DPI (ang. Mobile Industry Processor Interface - Display Parallel Interface) 24bity na piksel przy użyciu wbudowanego w układ STM32F7 kontrolera LTDC. Panel dotykowy komunikuje się z mikrokontrolerem poprzez interfejs I2C. Rezystory R3 i R8 mają za zadanie wymusić stan wysoki na liniach, ponieważ są

sterownie w trybie otwartego drenu. W tej konfiguracji można tylko zewrzeć linie do masy zmieniając stan wysoki na niski. Pozwoliło to wyeliminować sytuacje w której linia na jednym końcu jest zwarta do masy a na drugim do zasilania co doprowadziło by do zwarcia i gwałtownego wzrostu mocy mogącego uszkodzić układ.

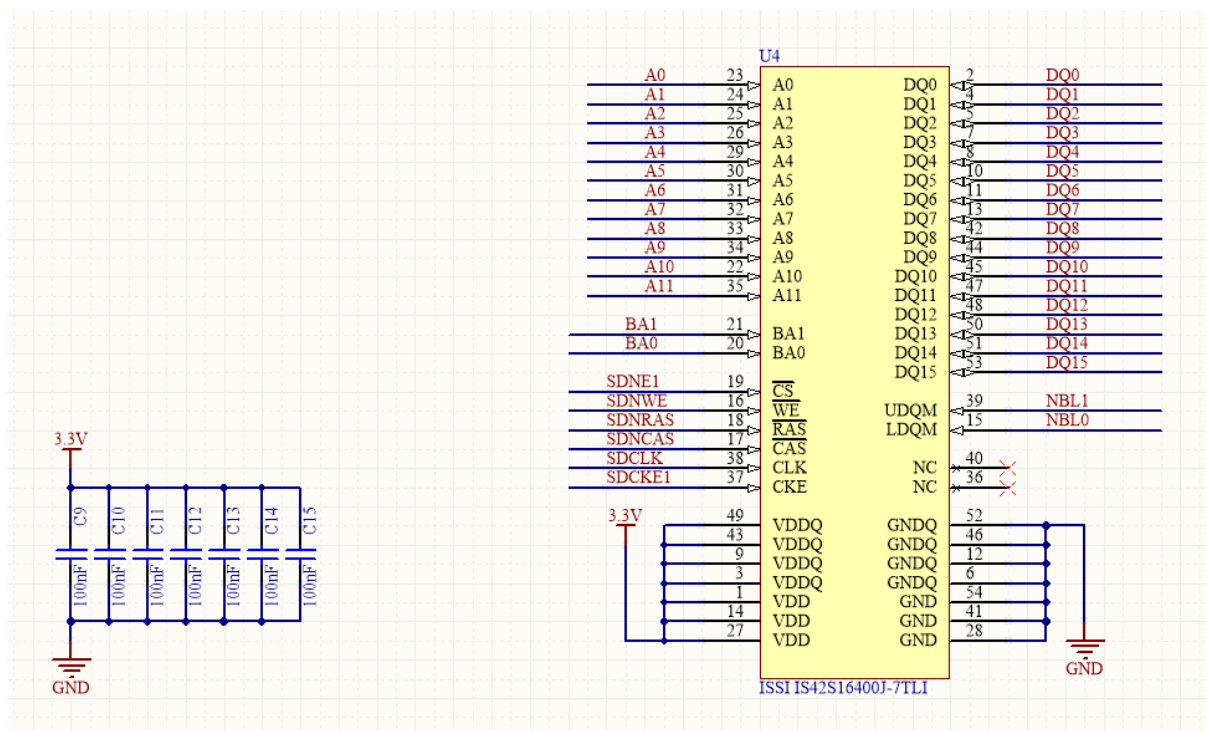


Rysunek 3.4 Schemat podłączenia wyświetlacza

3.1.5 SDRAM

Do działania interfejsu graficznego potrzebna jest pamięć do przechowywania ramek wysyłanych do wyświetlacza poprzez interfejs MIPI-DPI. Założono iż warstwy będą przechowywane w pamięci w formacie ARGB8888 (po bajcie na każdy kolor i kanał alfa). Kontroler LTDC jest w stanie sprzętowo mieszać dwie warstwy w wynikową która jest wysyłana do wyświetlacza. Zdecydowano, że potrzeba pamięci wystarczająco dużej, by zmieścić trzy bufor. Pierwsza warstwa byłaby przeznaczony na tło, które jest niezmiennie podczas działania urządzenia. Dwie pozostałe warstwy służyłyby do naprzemiennej prezentacji zmiennych danych takich jak położenie statku powietrznego. Będzie to implementacja mechanizmu podwójnego buforowania pozwalająca wyeliminować migotanie matrycy podczas modyfikacji bufora aktualnie wyświetlanego. Trzy warstwa o rozmiarze 1024 na 600 pikseli w formacie ARGB8888 zajmą 57600 Kb. W pracy zdecydowano się wyko-

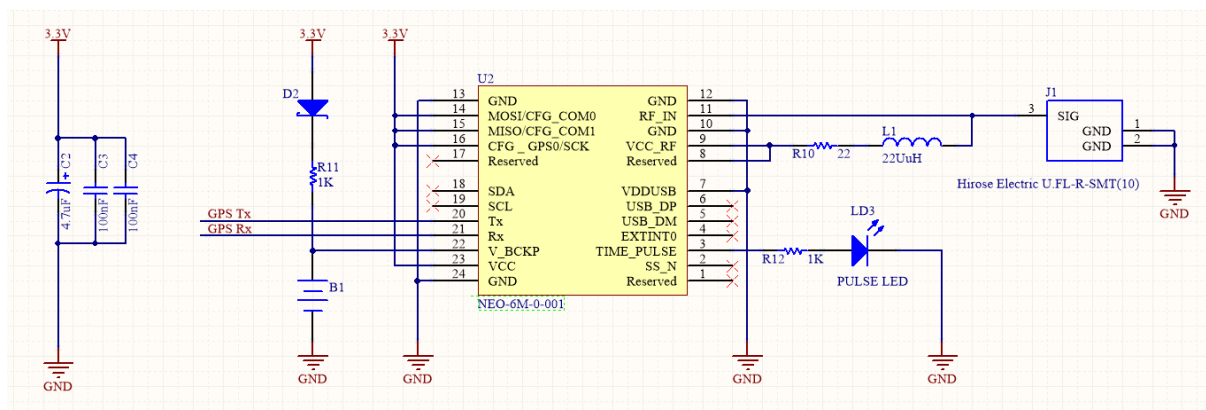
rzystać układ IS42S16400J-7TLI posiadający 65536 Kb pamięci. Poniżej przedstawiono schemat połączenia z mikrokontrolerem.



Rysunek 3.5 Schemat podłączenia zewnętrznej pamięci SDRAM

3.1.6 Moduł GPS

Aby system był w stanie poprawnie obliczyć odległość od namierzonego statku powietrznego i poprawnie zaznaczyć jego pozycję na radarze, potrzebna znać pozycję urządzenia. Do tego zadania wybrano układ NEO-6M-0-001 z zewnętrzną aktywną anteną. Do układu została podłączona zewnętrzna bateria podtrzymująca zasilanie. Dzięki temu urządzenie może uruchomić poprzez ciepły start, co pozwala zaoszczędzić czas potrzebny na znalezienie odpowiedniej liczby satelit. Do komunikacji z mikrokontrolerem wybrano interfejs UART.



Rysunek 3.6 Schemat podłączenia modułu GPS

3.2 Wykonanie PCB

W tym rozdziale szczegółowo opisano projekt oraz parametry wykonanego PCB. Pokazano obliczenia uzasadniające decyzje projektowe oraz ograniczenia wynikające z technologii produkcji i zastosowanych układów.

3.2.1 Parametry PCB

W projekcie PCB zdecydowano się na wykonanie technologia czterowarstwową. Pozwoliło to zmniejszyć rozmiary urządzenia oraz zapewnić dobre ekranowanie dla pomiędzy warstwami sygnałowymi. Ograniczyło to przesłuchy pomiędzy warstwami. Ponadto ciągłość warstw referencyjnych (masy i zasilania) pozwala by prądy powrotne przepływało możliwe najkrótszą drogą o najmniejszej impedancji zmniejszając emisję EMC. Poniżej przedstawiono tabele z układem warstw PCB.

Tabela 3.1 *Kolejność i grubość warst PCB*

Warstwa	grubość [mm]	grubość [mil]
sygnały	0.03556	1,4
dielektryk	0,17018	6,7
masa	0,01778	0,7
rdzeń	1,1938	47
zasilanie	0,01778	0,7
dielektryk	0,17018	6,7
sygnały	0.03556	1,4

Warstwy przewodzące wykonane są z miedzi natomiast jako dielektryk wykorzystano materiał FR-408 o stałej dielektrycznej $\varepsilon_r = 3.66$

3.2.2 GPS

Zgodnie z notą katalogową układu NEO-6M-0-001, ścieka antenowa powinna mieć impedancję dopasowaną do $Z_0 = 50\Omega$, którą obliczono z następującego wzoru :

$$Z_0 = \frac{87}{\sqrt{\varepsilon_r + 1.41}} \ln \left(\frac{5.98H}{0.8W + T} \right) \quad (3.1)$$

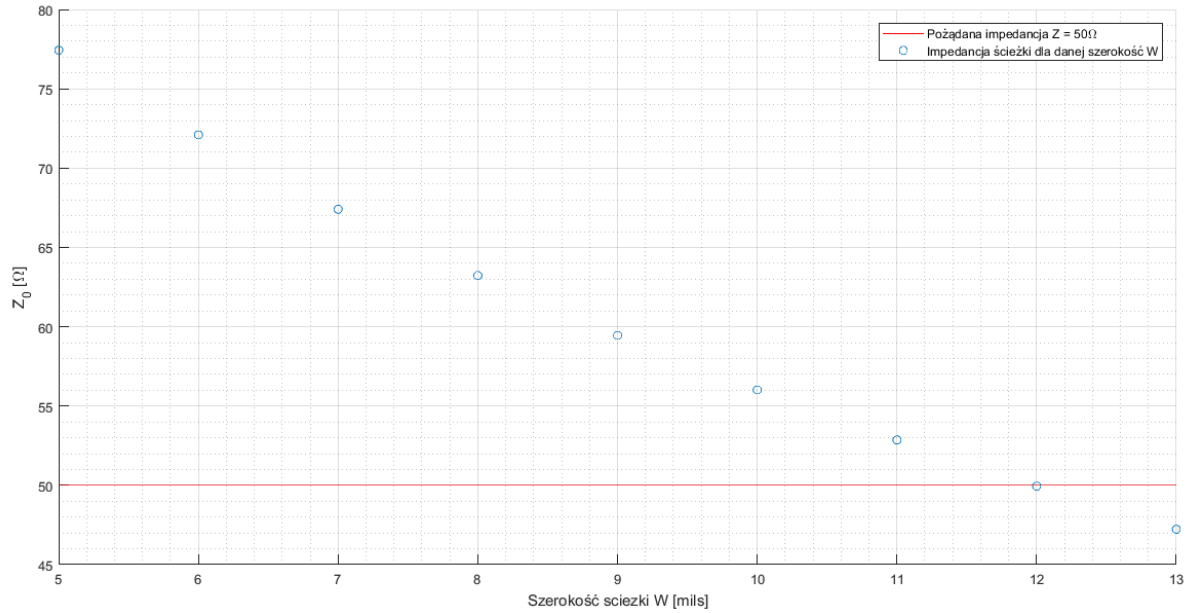
dla

$$0,1 < W/H < 2,0$$

$$1 < \varepsilon_r < 15$$

gdzie

- ε_r - stała dielektryczna
- H - grubość dielektryka
- W - szerokość ścieżki
- T - grubość ścieżki



Rysunek 3.7 Impedancja ścieżki w zależności od szerokości ścieżki

Powyższy wykres przedstawia impedancję ścieżki w zależności od szerokości dla anteny modułu GPS obliczoną ze wzoru 3.1 z parametrami $T = 1,4\text{mils}$ i $H = 6,7\text{mils}$ wziętymi z tabeli 3.1. Czerwoną linią zaznaczono pożądanę dopasowanie $Z_0 = 50\Omega$. Założono że W zostanie obliczone z dokładnością do 1 mils. Obliczenia wykonano zaczynając od najmniejszej szerokości gwarantowanej przez producenta $W = 5\text{mils}$, aż do punktu pod prostą z optymalnym dopasowaniem. Najlepsze dopasowanie otrzymano dla $W = 12\text{mils}$, gdzie $Z_0 \approx 49.95$. Dla takiego Z_0 obliczono współczynnik odbicia, który reprezentuje jaką część sygnału została stracona w wyniku niedopasowania impedancji przy przejściu pomiędzy liniami transmisyjnymi.

$$\Gamma = \frac{Z_l - Z_s}{Z_l + Z_s} \quad (3.2)$$

gdzie

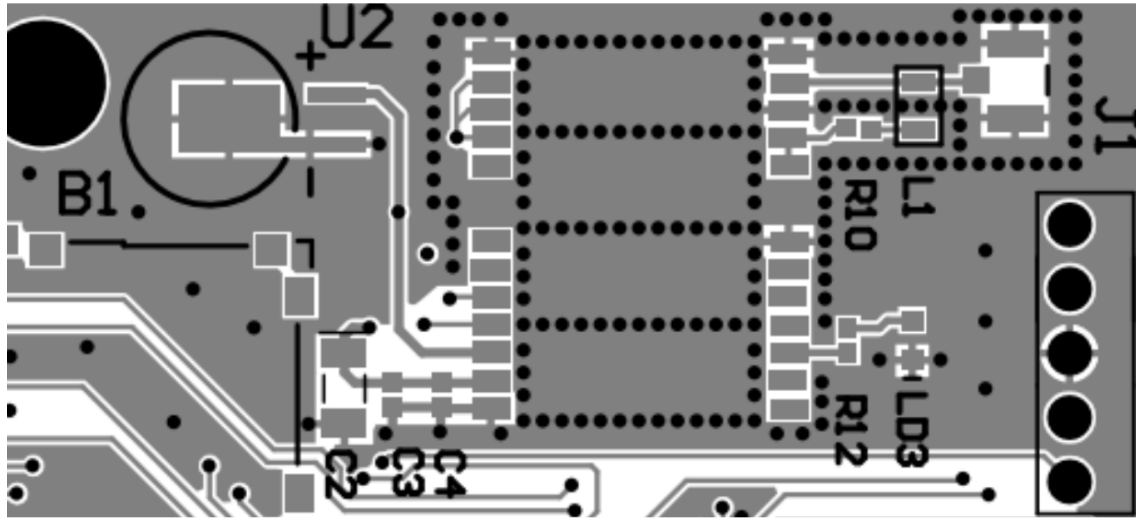
- Z_l - impedancja obciążenia, w tym przypadku impedancja linii transmisyjnej $Z_0 = 49,95\Omega$
- Z_s - impedancja źródła, czyli wejścia antenowego modułu GPS, zgodnie z notą katalogową $Z_s = 50\Omega$

Korzystając ze wzoru 3.2 obliczono:

$$\Gamma = \frac{Z_0 - Z_s}{Z_0 + Z_s} = \frac{49,95\Omega - 50\Omega}{49,95\Omega + 50\Omega} \approx 0.0005 \quad (3.3)$$

W przeliczeniu na procenty daje to $\Gamma \cdot 100\% = 0.0005 \cdot 100\% = 0.05\%$ sygnału odbitego przez linie transmisyjną. Jest to wartość akceptowalna zatem pozostano przy szerokości $W = 12\text{mils}$.

Zgodnie z zaleceniami noty katalogowej wykonano serię przelotek pod i do okola układu GPS (U2). Ten sam zabieg zastosowano również dla ścieżki antenowej (pomiędzy J1 i U2). Zapewniło to lepsze odprowadzanie ciepła przez moduł, a zatem niższą temperaturę pracy i mniejsza emisję EMC. Poniżej zaprezentowano zdjęcie z projektem.



Rysunek 3.8 Projekt PCB dla GPS

3.2.3 USB

W projekcie wykorzystano wbudowany w mikrokontroler kontroler interfejsu USB2.0 High-Speed o maksymalnej przepustowości 12Mb/s. Zgodnie ze standardem, należy dopasować impedancję różnicową pomiędzy liniami transmisyjnymi sygnału nieodwróconego i odwróconego do $90\ \Omega$. Poniżej przedstawiono zastosowany wzór oraz obliczenia.

$$Z_0 = \frac{174}{\sqrt{\varepsilon_r + 1.41}} \ln \left(\frac{5.98H}{0.8W + T} \right) \left(1 - 0.48e^{(-0.96\frac{D}{H})} \right) \quad (3.4)$$

dla

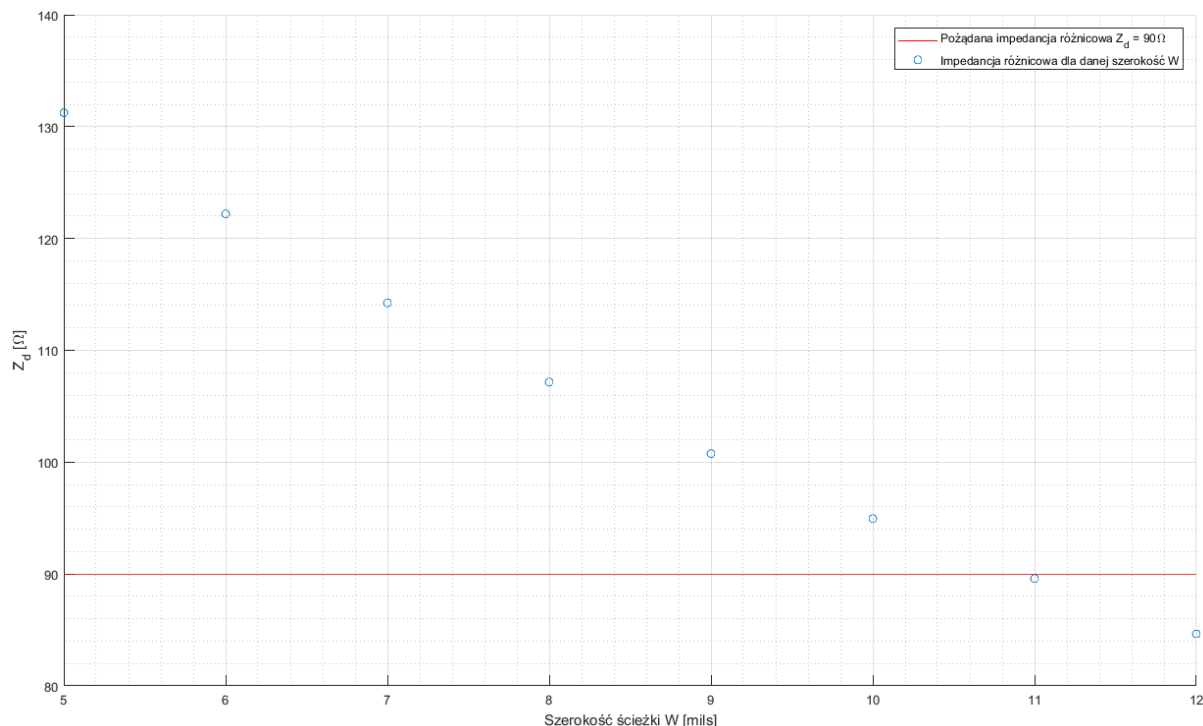
$$0,1 < W/H < 2,0$$

$$1 < \varepsilon_r < 15$$

gdzie

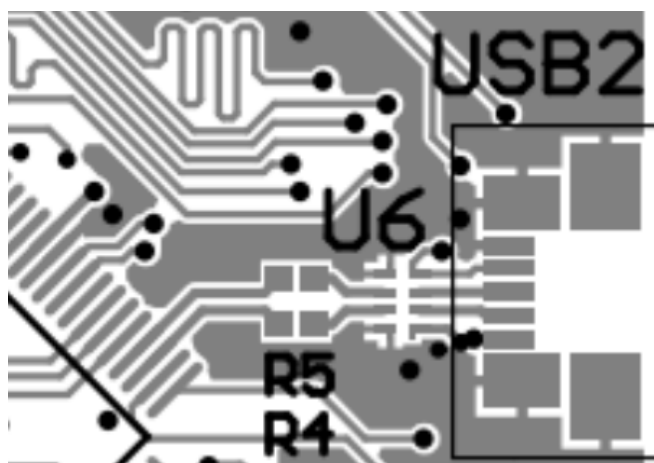
- ε_r - stała dielektryczna
- H - grubość dielektryka
- W - szerokość ścieżki
- T - grubość ścieżki
- D - odległość pomiędzy ścieżkami

Poniższy wykres przedstawia impedancję różnicową linii transmisyjnych interfejsu USB w zależności od szerokości obliczoną ze wzoru 3.4 z parametrami $T = 1,4\text{mils}$ i $H = 6,7\text{mils}$ wziętymi z tabeli 3.1. Założono odległość pomiędzy ścieżkami $D = 8\text{mils}$ równą rozstawowi padów w mikrokontrolerze. Czerwoną linią zaznaczono pożądane dopasowanie $Z_d = 90\Omega$. Założono że W zostanie obliczone z dokładnością do 1 mils. Obliczenia wykonano zaczynając od najmniejszej szerokości gwarantowanej przez producenta $W = 5\text{mils}$, aż do punktu pod prostą z optymalnym dopasowaniem. Najlepsze dopasowanie otrzymano dla $W = 11\text{mils}$, gdzie $Z_d \approx 89.56$. Zgodnie ze specyfikacją interfejsu USB2.0 Full-speed (12 Mb/s) impedancja różnicowa musi wynosić $Z_d = 90 \pm 15\%$. Zatem $76,5\Omega \leq Z_d \leq 103,5\Omega$. Z_d obliczone dla $W = 11\text{mils}$ spełnia to wymaganie.



Rysunek 3.9 Impedancja różnicowa w zależności od szerokości ścieżki

Wszystkie elementy zostały umieszczone na górnej warstwie sygnałowej, aby nie stosować przelotek które wprowadzając niepożądane pojemności i indukcyjności linii transmisyjnym. Ścieżki są zaginane po kącie nie większym niż 45° . Warstwą referencyjną dla sygnałów jest warstwa masy co zapewnia lepsze ekranowanie dla sygnałów szybkich. Linie różnicowe są oddzielone polem masy od innych sygnałów o co najmniej 50mils (1.27mm). Zastosowanie powyższych reguł pozwoliło ograniczyć zjawiska utrudniające dopasowania impedancji. Wspominanie w rozdziale schematu rezystory R5 i R4 służą jako szeregową terminację sygnału. Dzięki temu zabiegowi czasy narastania zboczy się większe co zmniejsza generowanie zakłócenia EMC. Układ U6 działa jako zabezpieczenie przeciwko wyładowaniom elektrostatycznym mogącym uszkodzić urządzenie oraz jako dodatkowy filtr EMC. Zgodnie z notą katalogową katalogową układu STM32F767xx ich rezystancja wynosi $22\ \Omega$. Poniżej przedstawiono część projektu PCB z interfejsem USB.



Rysunek 3.10 Projekt PCB dla interfejsu USB

3.2.4 Interfejsy szybkie

Częstotliwość sygnału zegarowego FMC wynosi 108MHz . W przypadku komunikacji z wyświetlaczem szybkość interfejsu zależy od oczekiwanej liczby klatek na sekundę. Częstotliwość sygnału zegarowego dla interfejsu MIPI-DPI wyraża się wzorem:

$$CLK = W \cdot H \cdot fps \quad (3.5)$$

gdzie:

- W - szerokość matrycy w pikselach
- H - wysokość matrycy w pikselach
- fps - częstotliwość odświeżania ekranu

Dla użytego w projekcie wyświetlacza przy założeniu odświeżania matrycy 60Hz korzystając z równania 3.5 otrzymujemy:

$$CLK = 1024 \cdot 600 \cdot 60\text{Hz} = 614400 \cdot 60\text{Hz} = 36864000\text{Hz} \approx 37\text{MHz}$$

Interfejsy o takich częstotliwościach zostały uznane za szybkie, co za tym idzie podjęto dodatkowe działania podczas projektowania ich linii transmisyjnych. Zadbano by warstwy referencyjne pod ścieżkami sygnałowymi były ciągłe, aby zapewnić ekranowanie i ograniczyć przesłuchy od linii po przeciwnej stronie płytki. Ponadto dopasowano długości wszystkich ścieżek w interfejsie, by sygnały przychodziły w tym możliwie podobnym czasie. Wzorując się na projekcie referencyjnym płytki ewaluacyjnej STM32F746G-DISCO zadbano, by różnica długości pomiędzy ścieżkami dla obu interfejsów nie była większa niż 100mils . Opóźnienie linii transmisyjnej, czyli czas jaki sygnał potrzebuje na pokonanie określonej drogi wyraża się wzorem:

$$t = \frac{l\sqrt{\epsilon_r}}{c} \quad (3.6)$$

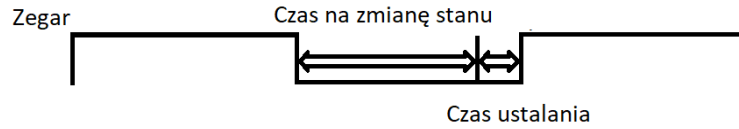
gdzie

- l - długość linii transmisyjnej
- ϵ_r - stała dielektryczna
- c - szybkość rozchodzenia się fali w próżni

Korzystając ze wzoru 3.6 dla parametrów $l = 0.00254\text{m}$ (100mils) i $c = 299\,792\,458\frac{\text{m}}{\text{s}}$ oraz $\epsilon_r = 3,66$ obliczono opóźnienie sygnału czyli maksymalny odstęp czasu w jakim dzieli sygnały w liniach o długości różniące się o 100mils .

$$t = \frac{0.00254\text{m}\sqrt{3,66}}{299792458\frac{\text{m}}{\text{s}}} = \frac{0.00485931\text{m}}{299792458\frac{\text{m}}{\text{s}}} = 1.621 \cdot 10^{-11} = 16,21\text{ps}$$

Wyświetlacz jest sterowany bezpośrednio poprzez interfejs LVDS (ang. Low Voltage Differential Signaling), jednak posiada również wbudowany konwerter THC63LVDM83D pozwalający na komunikację ze pomocą MIPI-DPI. Oznacza to, że wszelkie ograniczenia czasowe powinny być rozpatrywane względem wspomnianego wcześniej układu. Wyjścia mikrokontrolera zmieniają się przy zboczu opadającym sygnału zegarowego, natomiast



Rysunek 3.11 Diagram ograniczeń czasowych

są zatrzymywane przez konwerter przy zboczu narastającym. Wszystkie sygnały powinny dojść do wyświetlacza od momentu wystąpienia zbocza opadającego do narastającego z uwzględnieniem czasu ustalania się sygnału (ang. Setup Time). Jest to najpóźniejszy moment w którym muszą ustalić się stany na wszystkich liniach przed przyjściem sygnału taktującego. Poniżej przedstawiono diagram ilustrujący opisane ograniczenia czasowe.

Zatem maksymalna rozbieżność czasowa pomiędzy sygnałami wyraża się wzorem.

$$t_{max} = t_{low} - t_{setup} \quad (3.7)$$

gdzie

- t_{low} - czas trwania stanu niskiego dla sygnału zegarowego
- t_{setup} - czas ustalania dla sygnału

Z noty katalogowej mikrokontrolera odczytano że stan niski dla sygnału zegarowego LTDC wynosi minimalnie 45% okresu sygnału zegarowego obliczonego z równania 3.5 zatem:

$$t_{low} = \frac{1}{36864000 MHz} \cdot 45\% = 27127 ps \cdot 45\% = 12207 ps$$

W nocie THC63LVDM83D znaleziono $t_{setup} = 2000 ps$. Korzystając z 3.7 obliczono

$$t_{max} = 4167 ps - 2000 ps = 2167 ps$$

Ograniczenia czasowe nie zostały przekroczone, ponieważ $t_{max} = 12207 ps \geq t = 16, 21$.

W przypadku FMC wykorzystywanego do obsługi SDRAM, komunikacja odbywa się w obie strony. Ponownie czas trwania stanu niskiego sygnału zegarowego wynosi 45% okresu. Zatem:

$$t_{low} = \frac{1}{108 MHz} \cdot 45\% = 9259 ps \cdot 45\% = 4167 ps$$

Poniżej przedstawiono tabelę z czasami ustalania dla wszystkich sygnałów interfejsu odczytanych z noty układu IS42S16400J-7TLI.

Tabela 3.2 Czasy ustalania dla wszystkich sygnałów układu SDRAM

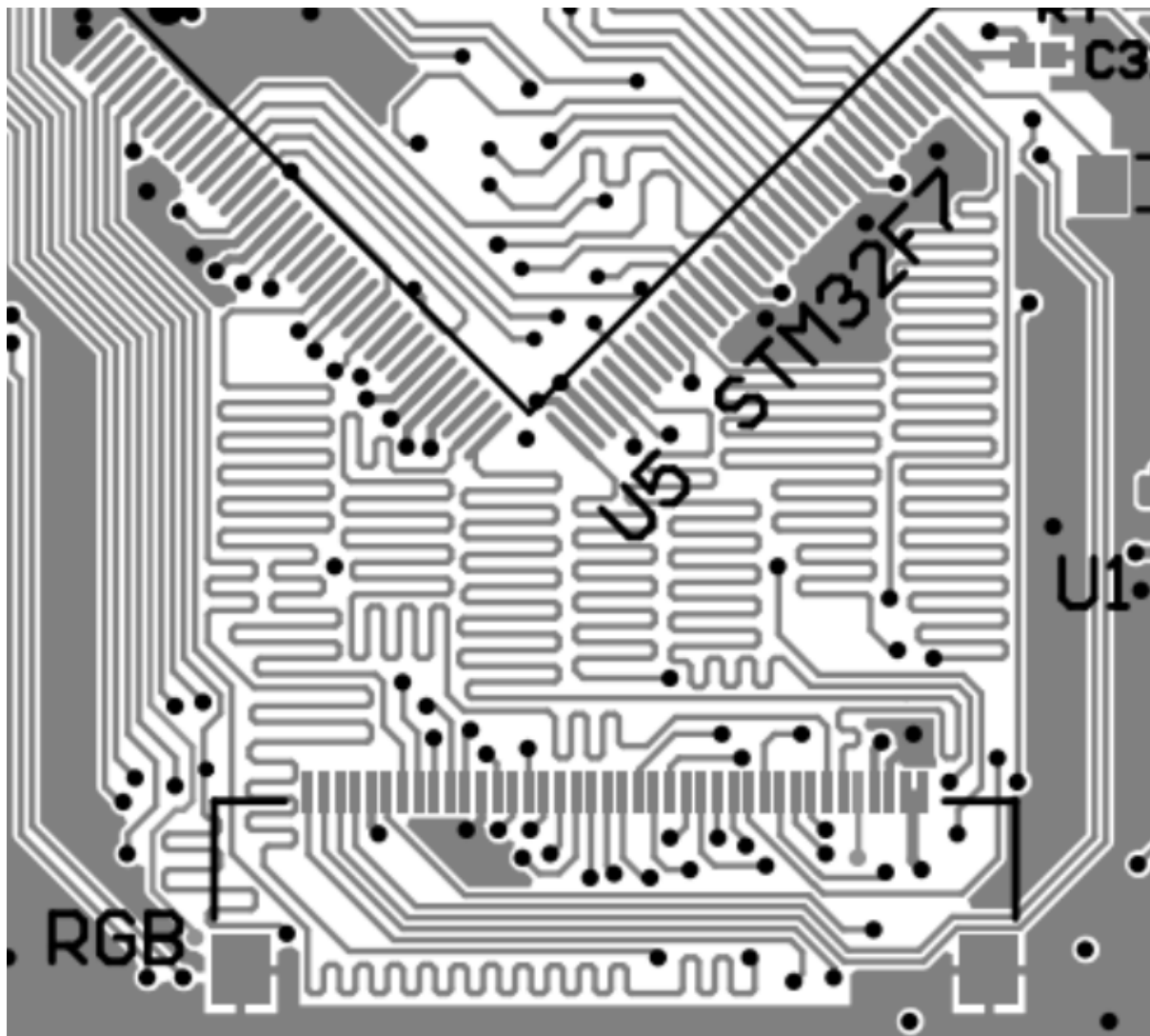
Parametr	czas [ps]
Input Data Setup Time	1500
Address Setup Time	1500
CKE Setup Time	1500
Command Setup Time	1500

Oznacza to, że można przyjąć $t_{setup} = 1500 ps$. Korzystając z równania 3.7 obliczono:

$$t_{max} = 4167 ps - 1500 ps = 2667 ps$$

Ponownie $t_{max} = 2667ps \geq t = 16, 21$, zatem ograniczenia czasowe nie zostały przekroczone.

Poniżej przedstawiono zbliżenie na projekt PCB, gdzie można zauważyć meandry na ścieżkach, których celem jest wyrównanie różnicy długości pomiędzy ścieżkami sygnałowymi.



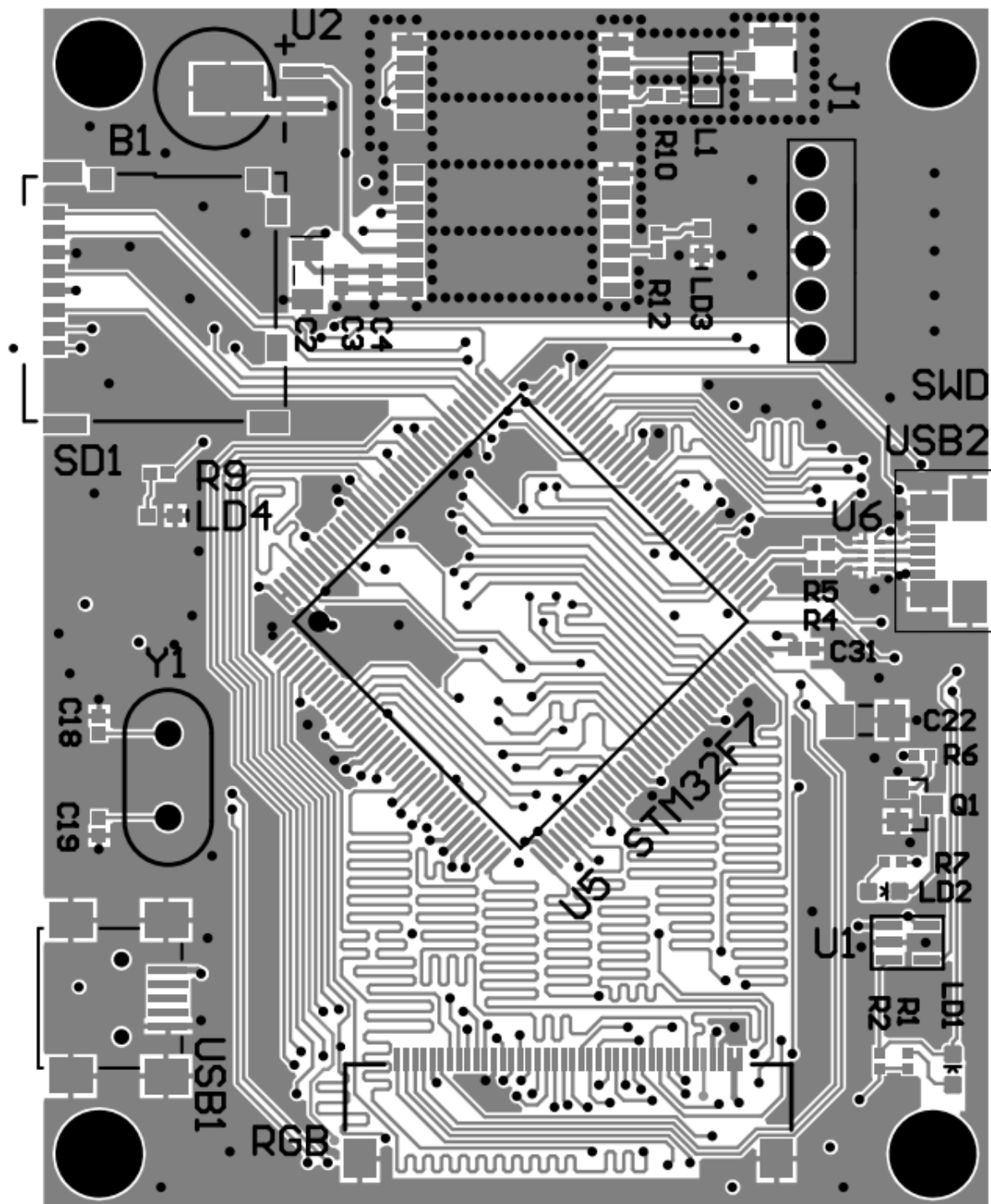
Rysunek 3.12 Przykład dopasowania długości ścieżek na PCB

3.2.5 Warstwa zasilania

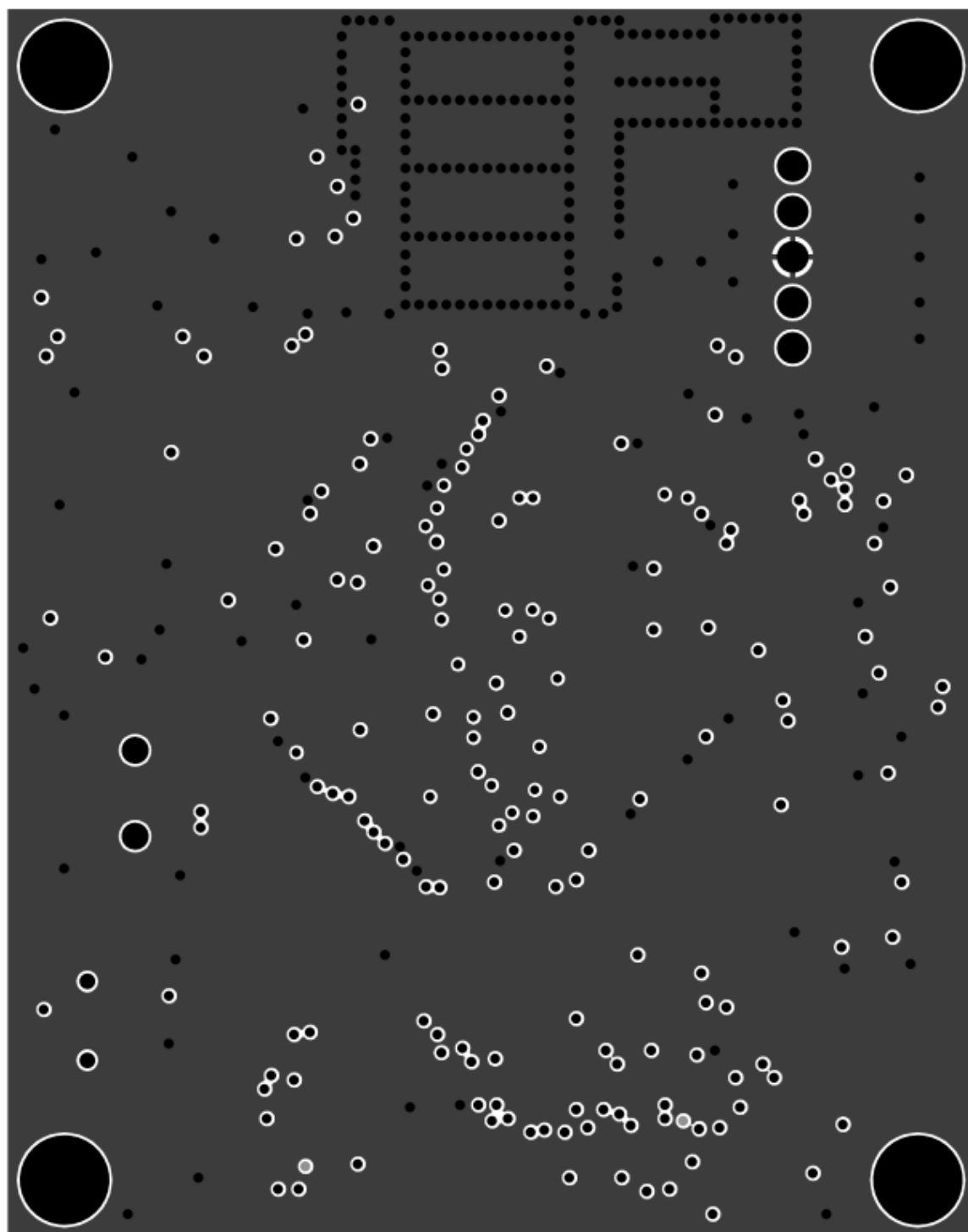
Większość układów na PCB jest zasilana napięciem 3,3V. Jednak interfejs USB i podświetlenie wyświetlacza wymaga napięciem 5V. Wymagało to odpowiedniego podzielenia warstwy trzeciej. Zadbano o to żeby żadne ścieżki sygnałów wrażliwych na zakłócenia nie przechodziły nad przerwą pomiędzy polem zasilanie 3,3V i 5V. Nieciągłość warstwy referencyjne powoduje powstawanie pętli prądowych, ponieważ prąd powrotny szuka innej dostępnej ścieżki powrotu. Zjawisko to może powodować wzrost zakłóceń elektromagnetycznych emitowanych przez urządzenie. Warstwę przedstawia obraz 3.2.6 w podrozdziale 3.2.6.

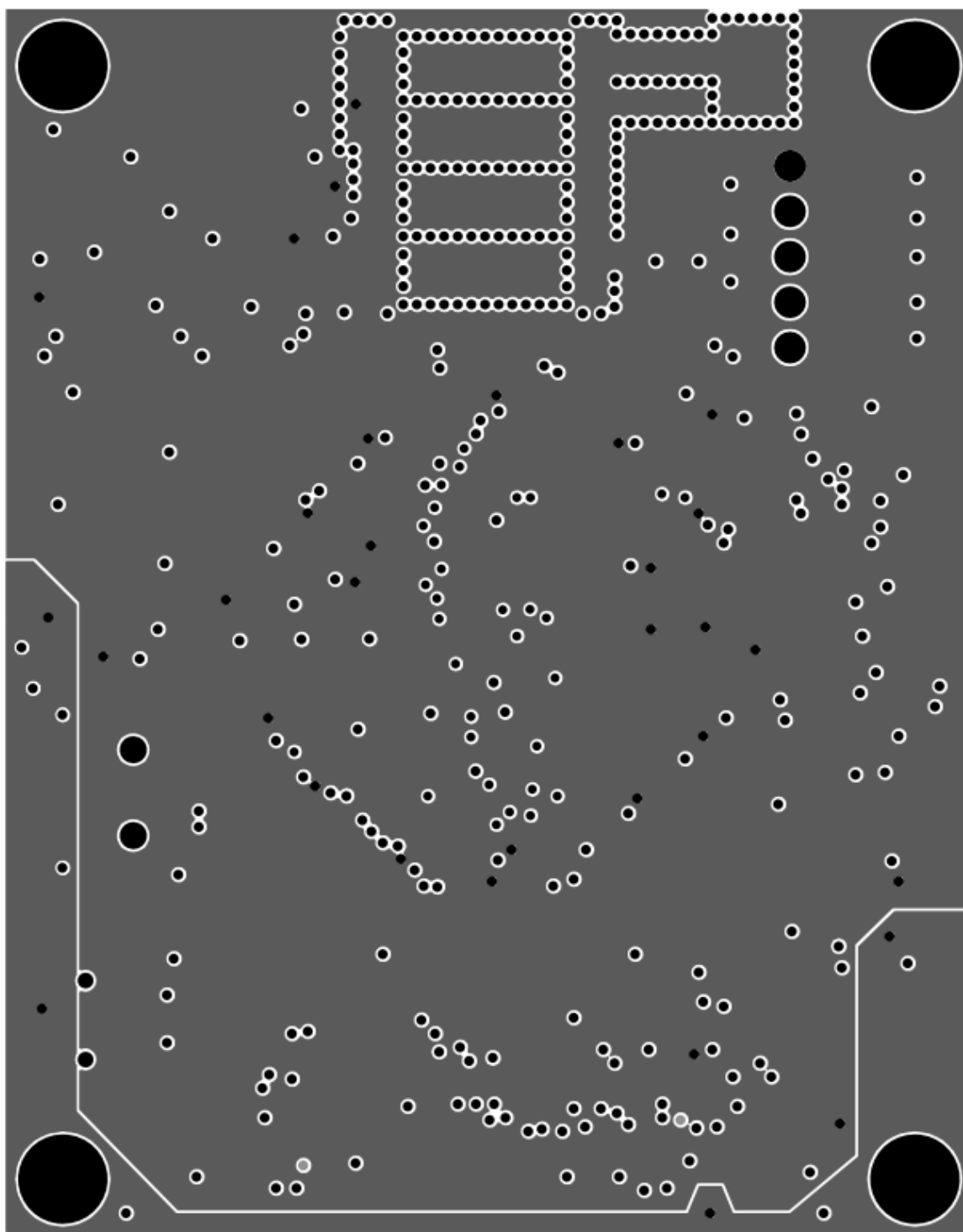
3.2.6 Prezentacja PCB

Poniżej przedstawiano obrazy wszystkich warstw z programu wykorzystanego do projektu oraz zdjęcie gotowego PCB.

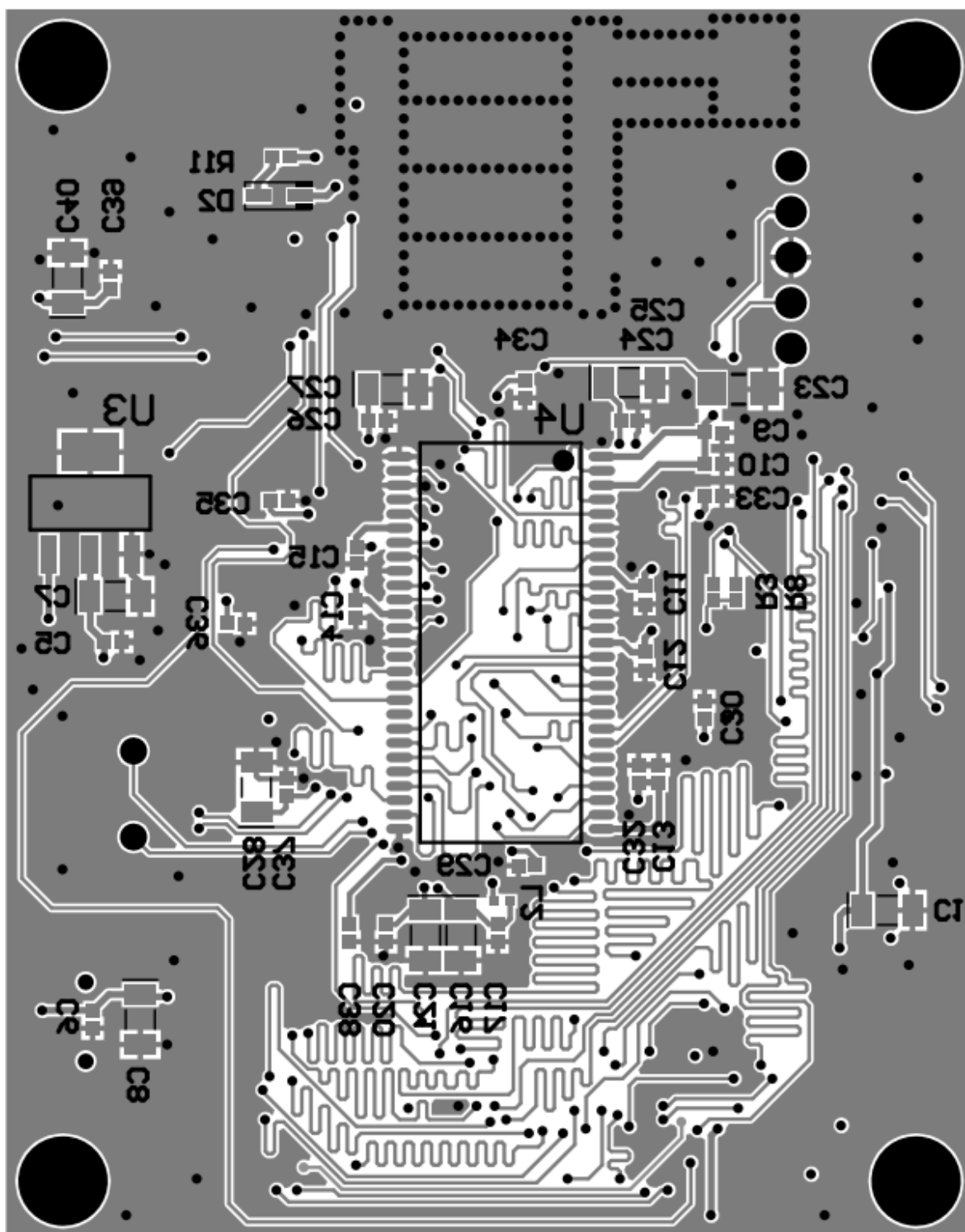


Rysunek 3.13 Układ warstwy 1.

Rysunek 3.14 *Układ warstwy 2.*



Rysunek 3.15 *Układ warstwy 3.*



Rysunek 3.16 Układ warstwy 4.

3.3 Warstwa Programowa

3.3.1 Model UML

3.3.2 System Operacyjny

Rozdział 4

Podsumowanie