



**AGH**

**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**  
Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

## Projekt dyplomowy

*Autonomiczny rejestrator czasu rzeczywistego do identyfikacji  
orientacji przestrzennej*  
*Real-time autonomous recorder for identification of spatial  
orientation*

Autor:  
Tomasz Bednorz  
Kierunek studiów:  
Automatyka i robotyka  
Opiekun pracy:  
*dr hab. inż. Adam Piłat, prof. u.*

*Serdecznie dziękuję  
dr hab. inż. Adamowi Piłatowi  
oraz mgr Hubertowi Milanowskemu  
za inspirację oraz merytoryczną  
pomoc podczas realizacji pracy.*

## Spis treści

<b>1 Wstęp</b>	<b>5</b>
1.1 Cel i zakres pracy . . . . .	6
<b>2 Jednostka do nawigacji inercyjnej</b>	<b>7</b>
2.1 Akcelerometr . . . . .	8
2.2 Żyroskop . . . . .	8
2.3 Magnetometr . . . . .	9
2.4 Przykłady rozwiązań opatycznych o IMU na rynku . . . . .	10
2.4.1 NGIMU . . . . .	10
2.4.2 3DMCX5-AHRS . . . . .	11
<b>3 Założenia projektowe</b>	<b>13</b>
3.1 Dobór komponentów elektronicznych . . . . .	13
3.1.1 IMU . . . . .	13
3.1.2 Moduł sieciowy . . . . .	14
3.1.3 Mikrokontroler . . . . .	15
3.2 Oprogramowanie . . . . .	17
3.2.1 Altium Designer . . . . .	17
3.2.2 STM32CubeIDE . . . . .	17
3.2.3 Visual Studio Code . . . . .	18
3.2.4 MATLAB . . . . .	18
<b>4 Warstwa sprzętowa urządzenia</b>	<b>19</b>
4.1 Schemat obwodu elektronicznego . . . . .	19
4.1.1 Sekcja zasilania . . . . .	20
4.1.2 Sekcja mikrokontrolera . . . . .	22
4.1.3 Sekcja IMU . . . . .	24
4.1.4 Sekcja modułu sieciowego . . . . .	25
4.1.5 Sekcja karty μSD . . . . .	25
4.2 Schemat układu PCB . . . . .	27
4.3 Montaż PCB . . . . .	30
<b>5 Warstwa oprogramowania urządzenia</b>	<b>31</b>
5.1 Oprogramowanie na platformę wbudowaną . . . . .	31
5.1.1 Generowanie kodu . . . . .	31
5.1.2 Automatyzacja procesu budowania projektu . . . . .	32
5.1.3 Architektura oprogramowania . . . . .	33

5.1.4	Praca systemu . . . . .	35
5.2	Aplikacja do akwizycji danych . . . . .	36
5.2.1	Interfejs graficzny . . . . .	36
<b>6</b>	<b>Testy systemu</b>	<b>39</b>
6.1	Przebieg eksperymentu . . . . .	40
<b>7</b>	<b>Podsumowanie</b>	<b>45</b>
	<b>Bibliografia</b>	<b>47</b>
	<b>Załączniki</b>	<b>49</b>

## 1 Wstęp

Określenie orientacji przestrzennej obiektów jest istotnym zagadnieniem od setek lat. Już na przełomie XI i XII wieku chińczycy używali do nawigacji igły magnetycznej unoszącej się na wodzie w naczyniu, czyli czegoś na wzór kompasu [1]. Było to niezwykle istotne w celu określenia pozycji podczas wojny oraz podróży morskich. Wraz z upływem czasu oraz rozwoju technologii powstawały coraz to dokładniejsze urządzenia w tej dziedzinie. W 1852 roku został wynaleziony żyroskop przez Francuskiego fizyka Jeana Foucaulta. Użył on go do zademonstrowania obrotu ziemi. W kolejnych dziesięcioleciach rozwiązywanie to było rozwijane oraz wykorzystywane w coraz to większej ilości dziedzin. Podczas II wojny światowej wczesne niemieckie systemy naprowadzania V2 łączyły dwa żyroskopy i akcelerometr poprzeczny z prostym komputerem analogowym w celu regulacji azymutu rakiety w locie [2]. Było to bardzo innowacyjne rozwiązanie w tamtych czasach.

Połączenie żyroskopu i akcelerometru określa się mianem IMU (ang. *inertial measurement unit*). Często ten zestaw poszerzany jest o magnetometr, który pomaga określić orientację systemu poprzez pomiar pola magnetycznego Ziemi. Wraz z rozwojem technologii MEMS (ang. *microelectromechanical systems*) i związaną z tym miniaturyzacją komponentów elektronicznych, czujniki do pomiaru inercji stały się bardzo popularne. Mają wiele swoich zastosowań zarówno w rozwiązaniach naukowych, jak i życia codziennego. Przykładem może być aparat fotograficzny, który wykrywa drgania w celu stabilizacji obrazu lub samochód, który jest w stanie zidentyfikować precyzyjnie moment wypadku w celu uruchomienia poduszek powietrznych.

Bardziej rozbudowaną wersją IMU jest AHRS (ang. *attitude and heading reference system*). Taki system poza zestawem sensorów posiada również wbudowany procesor, który ma za zadanie przekształcenie zebranych pomiarów. Dzięki niemu dane odbierane przez użytkownika są wolne od dryftu oraz przekształcone do innej, bardziej czytelnej postaci (np. orientacja, kwaterniony, macierz rotacji). Często wewnątrz procesora wbudowany jest już algorytm odpowiedzialny za filtrację danych (np. rozszerzony filtr Kalmana).

W czasach współczesnych wśród najbardziej popularnych metod stosowanych do określania pozycji poza nawigacją inercyjną jest nawigacja satelitarna. Jej najpopularniejszym systemem jest GPS (ang. *Global Positioning System*). Został opracowany przez departament obrony Stanów Zjednoczonych i obejmuje zasięgiem całą kulę ziemską. Jego działanie polega na pomiarze czasu lotu sygnału radiowego pomiędzy odbiornikiem, a sztucznymi satelitami o znanym położeniu. Na tej podstawie urządzenie odbiorcze z wbudowanym procesorem jest w stanie obliczyć aktualną pozycję geograficzną obiektu.

## 1.1 Cel i zakres pracy

Obecnie na rynku można znaleźć wiele rozwiązań zawierających zbiór sensorów typu IMU. Najtańsze z nich kosztują kilkadziesiąt, a najdroższe kilkadziesiąt tysięcy złotych. Różnią się między innymi: rozmiarem, napięciem zasilania, interfejsami komunikacyjnymi, częstotliwością pracy oraz zakresem i dokładnością akwizycji danych. Pomimo sporej liczby dostępnych układów, trudno jest znaleźć w przystępnej cenie taki, który oferowałby komunikację opartą na standardzie Ethernet, gwarantującym bardzo szybką transmisję danych, a co się z tym wiąże wysoką przepustowość łącza.

Biorąc pod uwagę uprzednio przedstawione informacje, celem pracy jest zaprojektowanie platformy do poboru danych w czasie rzeczywistym z jednostki do nawigacji inercyjnej. Komunikacja z urządzeniem będzie odbywała się za pomocą komputera klasy PC z wbudowanym złączem Ethernet umożliwiającym transmisję danych przy użyciu protokołu TCP. Przekazywanie danych pomiędzy platformą, a komputerem będzie odbywało się przy użyciu przełącznika sieciowego. Powstaną również dedykowane narzędzia umożliwiające zapis zebranych danych do pliku tekstopowego oraz pozwolą na synchronizację pomiędzy urządzeniami w momencie, gdy będzie podpięta do przełącznika sieciowego (ang. *switch*) ich większa liczba.

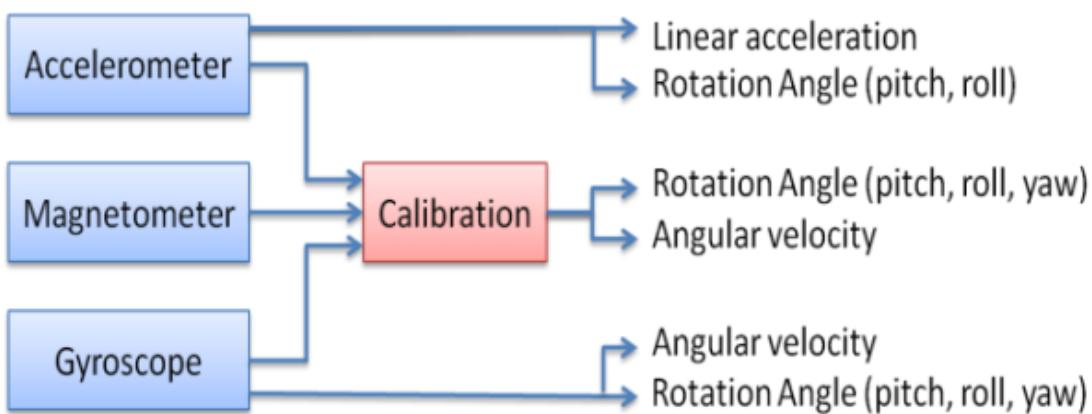
W drugim rozdziale zostaną przedstawione informacje na temat IMU oraz sensorów w nim zawartych, czyli akcelerometru, żyroskopu oraz magnetometru. Pokazane będą również obecnie dostępne na rynku rozwiązania oparte o jednostkę do nawigacji inercyjnej. Trzeci rozdział poinformuje o założeniach projektowych. Znajdą się w nim informacje dotyczące wyboru najważniejszych komponentów elektronicznych oraz wykorzystywane oprogramowanie komputerowe. Czwarty rozdział pokaże proces projektowania obwodu elektronicznego oraz płytki PCB wraz z montażem. Natomiast piąty przedstawi dedykowane oprogramowanie, które powstało podczas realizacji projektu. Kolejny rozdział zaprezentuje praktyczne wykorzystanie systemu, a ostatni przeznaczony jest na podsumowanie projektu.

## 2 Jednostka do nawigacji inercyjnej

IMU jest urządzeniem elektronicznym zawierającym kilka rodzajów czujników. Najczęściej zawartych w jednym układzie scalonym lub w obrębie niewielkiej płytki drukowanej. W jego skład wchodzą jeden lub więcej akcelerometrów oraz jeden lub więcej żyroskopów. Czasami posiada również magnetometr. Najpopularniejsza konfiguracja zawiera po jednym akcelerometrze, magnetometrze i żyroskopie na każdą z trzech osi orientacji kątowej - *pitch*, *roll*, *yaw*. Określa się ją mianem 9DoF (ang. Degrees of Freedom).

W obecnych czasach ciągle rośnie liczba dziedzin w których można zastosować czujniki inercyjne. Jest to związane z rozwojem technologii MEMS na której bazują. Umożliwiają określenie pozycji oraz orientacji każdego obiektu do którego są przyczepione. W związku z tym mogą służyć między innymi do śledzenia ludzkiego ruchu (np. opaski na nadgarstek zliczające liczbę wykonanych kroków) lub estymacji pozycji obiektów latających (np. zachowanie równowagi drona). Czujniki inercyjne w połączeniu z pomiarami GNSS (ang. *Global navigation satellite system*) są używane do określania pozycji samochodów podczas autonomicznej jazdy [3].

IMU jest głównie stosowane w urządzeniach do pomiaru prędkości, orientacji oraz siły grawitacji. W przeszłości do pomiarów wykorzystywano tylko dwa rodzaje czujników: akcelerometr oraz żyroskop. Wraz z rozwojem technologii zaczęto wykorzystywać kolejny typ sensora - magnetometr. Umożliwia on określenie kierunku oraz wartości pola magnetycznego, którego uwzględnienie może pozytywnie wpływać na odczyt pomiarów z żyroskopu [4]. Schemat przedstawiony na rysunku 5.2 prezentuje zasadę działania IMU opartego o trzy typy czujników.

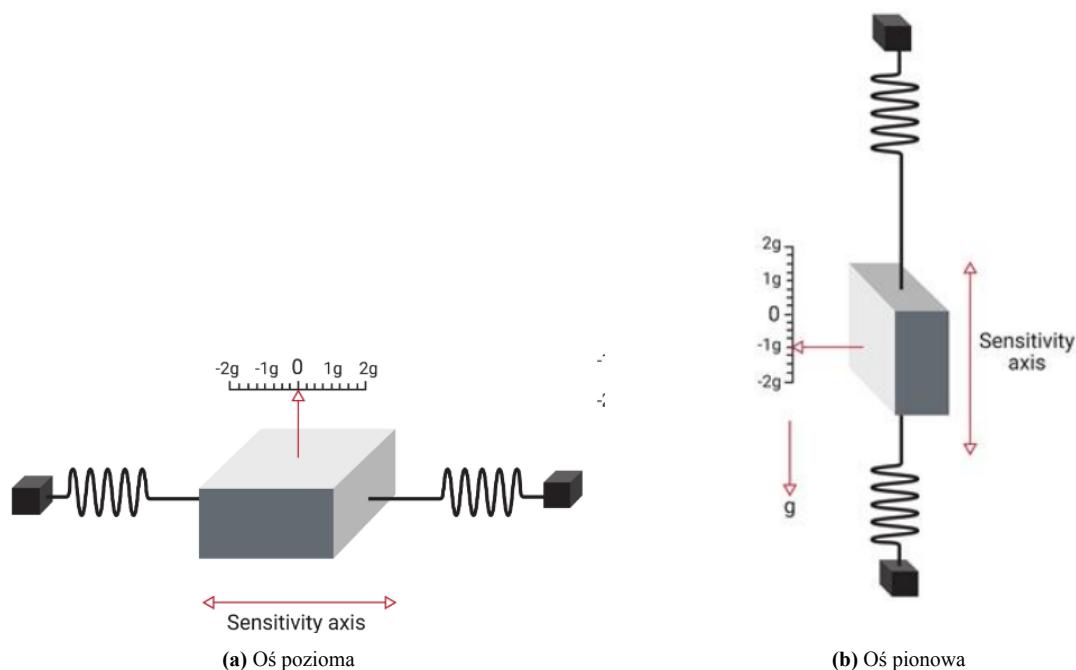


Rys 2.1. Zasada działania IMU [4]

W poniższych podrozdziałach zostaną omówione poszczególne sensory wchodzące w skład jednostki do nawigacji inercyjnej oraz przykłady rozwiązań bazujących na IMU aktualnie znajdujących się na rynku.

## 2.1 Akcelerometr

Akcelerometr jest czujnikiem umożliwiającym pomiar przyspieszenia, czyli wzrostu lub spadku prędkości w pewnym okresie czasu. Zasadę jego działania można zobrazować wykorzystując drugą zasadę dynamiki Newtona, gdzie elementem aktywnym jest masa zawieszona na sprężynie z tłumikiem znajdująca się w pewnym układzie odniesienia. Zostało to przedstawione na rysunku 2.2. Za pomiar wskazujący  $-1\text{ g}$  w osi pionowej odpowiedzialna jest grawitacja. Warto zauważyć, że podczas wolnego spadania wyniósłby  $0\text{ g}$  pomiomo, że przyspieszenie byłoby niezerowe.



Rys 2.2. Prosty model akcelerometru [5]

Można wyróżnić dwie główne metody pomiaru przyspieszenia przez akcelerometry oparte o technologię MEMS. Pierwsza z nich polega na badaniu zmiany pozycji masy w układzie odniesienia. Druga natomiast mierzy zmianę częstotliwości wibrującego elemntu aktywnego [6].

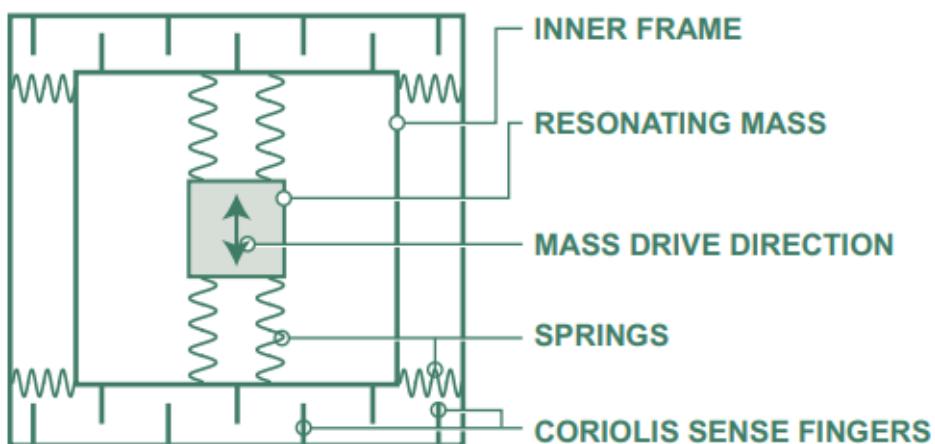
## 2.2 Żydroskop

Żydroskop to czujnik mierzący prędkość kątową obiektu dla własnego inercyjnego układu odniesienia. Na rynku jest dostępnych wiele różnych rodzajów żyroskopów o zróżnicowanym stopniu dokładności, między innymi:

- mechaniczne,
- optyczne,

- laserowe,
- MEMS.

Te ostatnie wykorzystują mechaniczny element wibrujący jako człon sensoryczny do pomiaru prędkości kątowej. Nie posiadają części obrotowych wymagających łożyskowania co pozwala na łatwą miniaturyzację [7]. Działanie żyroskopów MEMS oparte jest na efekcie Coriolisa, czyli siły bezwładności, która działa na poruszające się ciało w obracającym układzie odniesienia. Schemat mechanicznej struktury żyroskopu został przedstawiony na rysunku 2.3.



Rys 2.3. Schemat struktury mechanicznej żyroskopu [8]

Powszechnie znaną wadą żyroskopu jest dryft, czyli błąd pomiaru wynikający z systematycznego całkowania błędu odchylenia oraz zakłóceń [9]. Częstą metodą stosowaną w celu kompensacji dryftu jest zaprojektowanie modelu uwzględniającego wskazania z innych czujników, takich jak akcelerometr lub magnetometr.

## 2.3 Magnetometr

Magnetometr to rodzaj czujnika umożliwiający pomiar siły oraz kierunku pola magnetycznego. Działanie większości magnetometrów MEMS opiera się na sprawdzeniu wartości magnetooporu (ang. *magnetoresistance*), czyli zmiany rezystacji materiału pod wpływem zewnętrznego pola magnetycznego [5]. Zazwyczaj czujniki tego typu są używane do badania lokalnego pola magnetycznego, którego składowymi są pole magnetyczne Ziemi oraz pola magnetyczne utworzone przez pobliskie obiekty.

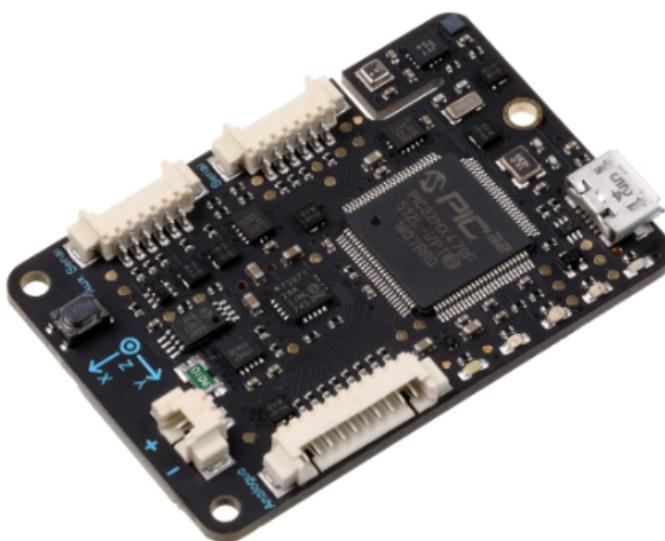
Zastosowanie magnetometru w jednostce do nawigacji inercyjnej ma swoje wady i zalety. Mocną stroną tego rozwiązania jest możliwość otrzymywania kąta odchylenia (ang. *yaw*), co w połączeniu z danymi z żyroskopu daje szansę na redukcję błędu związanego z dryfem. Natomiast mankamentem są pomiary przeprowadzone w środowisku zawierającym metale o właściwościach feromagnetycznych. Ich wyniki mogą być zakłócone.

## 2.4 Przykłady rozwiązań opartych o IMU na rynku

Na rynku można znaleźć wiele komercyjnych rozwiązań przeznaczonych do akiwzycji danych z użyciem jednostki do nawigacji inercyjnej. Niektóre z nich zawierają dedykowane oprogramowanie umożliwiające szybkie rozpoczęcie pracy z użyciem danej platformy oraz umożliwiające przetworzenie zebranych danych. Poniżej zaprezentowane zostaną dwa rozwiązania wraz z ich praktycznym wykorzystaniem.

### 2.4.1 NGIMU

Pierwsza platforma to NGIMU (Next Generation Inertial measurement unit) firmy x-io Technologies z Wielkiej Brytanii zajmującej się rozwiązaniami czujnikowymi w systemach wbudowanych (rys. 2.4). Na jej pokładzie znajduje się trójosiowy żyroskop, akcelerometr oraz magnetometr, a także czujniki ciśnienia oraz wilgotności. Posiada wbudowany algorytm fuzji danych, który zapewnia pomiary orientacji jednostki względem Ziemi bez dryftu. Ma wyprowadzone złącze z dziesięcioma wejściami analogowymi umożliwiające podłączenie zewnętrznych czujników i sygnałów. Cena NGIMU wraz z obudową wynosi 310 funtów brytyjskich nie wliczając podatku VAT.



Rys 2.4. NGIMU firmy x-io Technologies [10]

Rozwiązanie firmy x-io Technologies jest wykorzystywane w wielu pracach badawczych. Przykładem mogą być naukowcy z *School of Information Engineering Xijing University* w Chinach, którzy użyli NGIMU przyczepione do stopy oraz mapę rozkładu budynku do utworzenia systemu nawigacji wewnętrzobudynkowej [11]. Zastowali filtr Kalmana do aktualizacji prędkości zerowej w celu korekty błędu nawigacji inercyjnej oraz nierekurencyjny filtr Bayesowski, który wykorzystuje informacje z mapy pomieszczeń w celu kalibracji pozycji i kierunku pieszego.

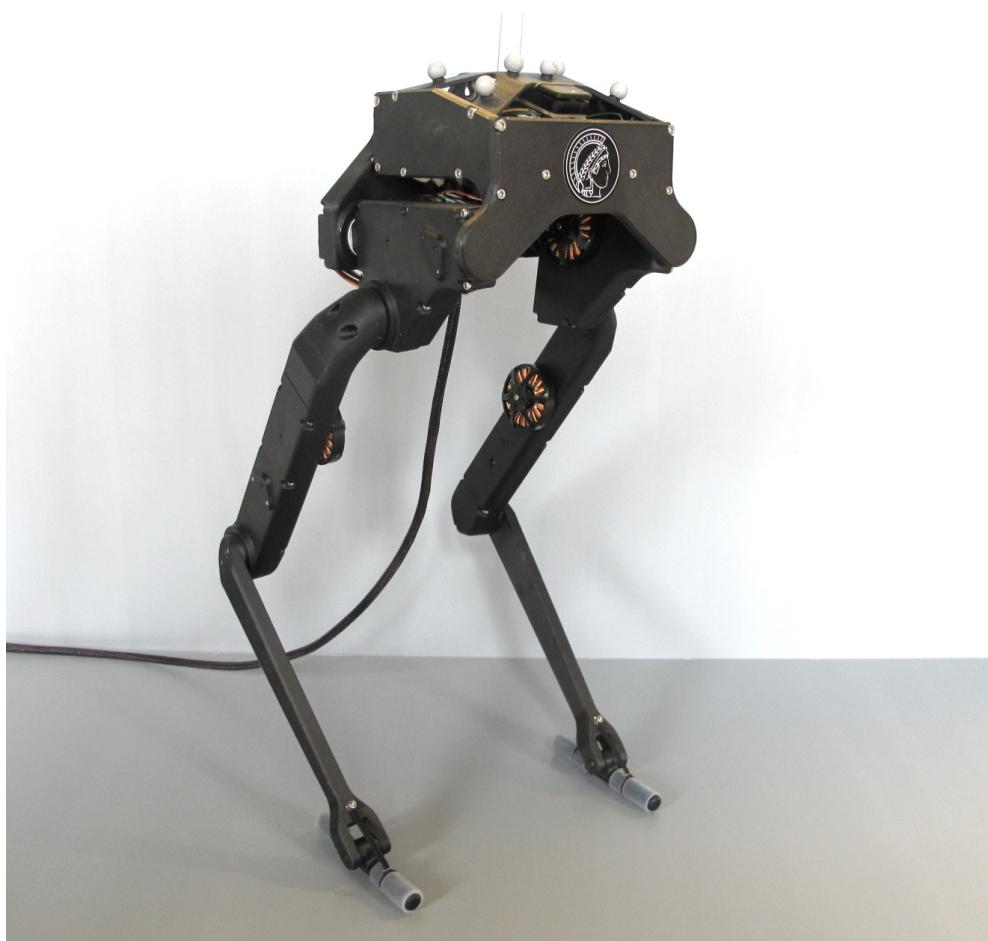
#### 2.4.2 3DMCX5-AHRS

Druga platforma to czujnik odniesienia położenia o wysokiej wydajności 3DMCX5-AHRS amerykańskiej firmy Parker LORD (rys. 2.5). Zawiera dwurdzeniowy procesor na którym zrealizowany jest autoadaptacyjny rozszerzony filtr Kalmana. Ma niewielki rozmiar i waży zaledwie 12 g. Umożliwia pobór danych z IMU o częstotliwości do 1000 Hz. Cena pojedyńczego układu wynosi około 6000 złotych bez podatku VAT.



Rys 2.5. 3DMCX5-AHRS firmy Parker LORD [12]

*Open Dynamic Robot Initiative* to projekt o otwartym kodzie źródłowym [13]. W jego ramach udostępnione są schematy elektroniczne, mechaniczne i oprogramowanie dla robotów różnego typu o stosunkowo niskim koszcie budowy oraz poziomie skomplikowania. Wśród przedstawionych rozwiązań jest robot dwunożny o sześciu stopniach swobody (rys. 2.6). Zarówno ten jak i inne konstrukcje możliwe do realizacji w ramach projektu wykorzystują czujnik 3DMCX5-AHRS.



Rys 2.6. Robot dwunożny zrealizowany w ramach *Open Dynamic Robot Initiative* [14]

### 3 Założenia projektowe

Główym celem przyświecającym powstaniu niniejszej pracy był brak urządzenia w zasobach koła naukowego Focus działającego przy Katedrze Automatyki i Robotyki pozwalającego na identyfikację dynamiki wszelkiego rodzaju robotów. Przede wszystkim robotów przemysłowych oraz humanoidalnych, które są głównym tematem prac odbywających się we wspomnianym kole naukowym. Rozwiązań dostępne na rynku w przystępnej cenie nie spełniały oczekiwanych wymagań, takich jak: możliwość synchronizacji zegarów wielu urządzeń oraz transmisja danych oparta o standard Ethernet. Poza wcześniej wspomnianymi wymaganiami, realizowana platforma w postaci płytki PCB (ang. *printed circuit board*) powinna mieć możliwość transmisji danych zebranych z jednostki do nawigacji inercyjnej w czasie rzeczywistym wraz z uwzględnieniem znacznika czasu (ang. *timestamp*). Dodatkowo użytkownik urządzenia ma mieć możliwość łatwego połączania się z nim korzystając z komputera klasy PC oraz zapisu zebranych danych do pliku tekstowego.

#### 3.1 Dobór komponentów elektronicznych

Bardzo ważnym etapem podczas projektowania płyt drukowanych jest dobór odpowiednich elementów elektronicznych. Układy scalone oraz drobniejsze komponenty (np. rezystory, kondensatory, cewki) należy wybrać z odpowiednią starannością. Powinny spełniać postawione przez nas wymagania oraz umożliwiać odpowiednie funkcjonowanie systemu. W poniższych podrozdziałach zostanie przedstawiony tok myślowy doboru najistotniejszych podzespołów elektronicznych będących częścią projektowanej płytki PCB.

##### 3.1.1 IMU

Dobór jednostki do nawigacji inercyjnej, czyli układu odpowiedzialnego za jakość pomiarów przeprowadzony był na podstawie kilku kryteriów. Pierwsze z nich to rozmiar układu. Miał być niewielki. Jego pole powierzchni nie mogło przekraczać kilku centymetrów kwadratowych. Kolejne kryterium to jakość i dokładność pomiarów wykonywanych przez czujniki wchodzące w skład IMU. Nie powinny one odbiegać negatywnie w porównaniu z urządzeniami w podobnej półce cenowej. Wyniki przyśpieszenia, prędkości kątowej oraz pola magnetycznego miały być dostępne dla wszystkich trzech osi. Urządzenie powinno posiadać conajmniej jeden z standardowych interfejsów komunikacyjnych stosowanych w systemach wbudowanych, co jest niezbędne przy komunikacji z mikrokontrolerem. Pozostałe uwzględnione kryteria to: napięcie zasilania, zużycie energii oraz maksymalna częstotliwość akwizycji danych przez układ.

Urządzenie spełniające powyższe kryteria na które padł wybór to moduł MEMS firmy Xsens o oznaczeniu MTi-1(rys. 3.1). Jest to niewielka płytko o rozmiarze 12.1 x 12.1 x 2.55 mm posi-



Rys 3.1. Jednostka do nawigacji inercyjnej Xsens MTi-1 [15]

adająca na sobie kilka układów scalonych oraz drobniejsze komponenty elektroniczne. W jej skład wchodzą trójosiowy akcelerometr, żyroskop oraz magnetometr. Poza tym posiada czujnik temperatury oraz procesor taktowany zewnętrznym rezonatorem kwarcowym. Standardowy zakres pomiaru przyśpieszenia wynosi  $\pm 16$  g, prędkości kątowej  $\pm 2000$  °/s, a pola magnetycznego 8 G. Komunikacja z urządzeniem może się odbywać za pośrednictwem jednego z trzech interfejsów: SPI (ang. *Serial Peripheral Interface*), I2C (ang. *Inter-Integrated Circuit*) oraz UART (ang. *Universal Asynchronous Receiver-Transmitter*). Dozwolone napięcie wejściowe mieści się w zakresie pomiędzy 2.19 V, a 3.6 V. Maksymalne zużycie energii określane przez producenta wynosi 100 mW. Jednostka potrafi wygenerować synchroniczne dane z wszystkich sensorów z częstotliwością 100 Hz.

### 3.1.2 Moduł sieciowy

Jednym z podstawowych założeń powstającej płytki PCB jest możliwość przekazywania dalej danych pobranych z IMU w czasie rzeczywistym. Postanowiono zastosować rozwiązanie oparte na standardzie Ethernet. Idealnym rozwiązaniem byłoby zastosowanie mikrokontrolera, który posiada sprzętową obsługę drugiej warstwy modelu OSI (ang. *Open Systems Interconnection model*), czyli łącza danych. Umożliwiłoby to nadanie urządzeniu fizycznego adresu MAC (ang. *Media Access Control*), a następnie połączenie z zewnętrznym nadajnikodemodnikiem (ang. *transceiver*) Ethernet odpowiedzialnym za pierwszą warstwę modelu OSI, czyli warstwę fizyczną. Niestety z powodu kryzysu na rynku półprzewodników nie udało się znaleźć interesujących układów scalonych umożliwiających zastosowanie tego rozwiązania.

Po przeanalizowaniu dostępnych na rynku komponentów dokonano wyboru modułu sieciowego WIZ820io (rys. 3.2). W jego skład wchodzi chip W5200 [18] będący kontrolerem Ethernet. Posiada również złącze RJ45 wraz z transformatorem dzięki czemu jest niezwykle łatwy w obsłudze od strony sprzętowej.

Komunikacja z chipem W5200 znajdującym się na pokładzie modułu odbywa się za pomocą protokołu komunikacyjnego SPI o maksymalnej prędkości transmisji 80 MHz. Jednak zgodnie z



Rys 3.2. Moduł sieciowy WIZ820io [16]

notą katalogową producent informuje o stabilnej komunikacji przy minimalnej gwarantowanej prędkości na poziomie 33.3 MHz. Posiada 32 kB pamięci przeznaczonej na wewnętrzne bufory do transmisji oraz odbioru danych. Obsługuje dwa rodzaje standardu Ethernet: 10BASE-T (maksymalna przepustowość 10 Mb/s) i 100BASE-TX (maksymalna przepustowość 100 Mb/s). Umożliwia sprzętową obsługę dwóch pierwszych warstw modelu OSI oraz stosu TCP/IP ze wsparciem kilku protokołów wyższych warstw przedstawionych w tabeli [1]. Zalecane napięcie zasilania wynosi 2.97÷3.63 V, a maksymalny pobór prądu to 175 mA.

Tabela 1. Protokół wspierane sprzętowo przez moduł WIZ820io

Skrót	Nazwa	Warstwa modelu OSI
TCP	<i>Transmission Control Protocol</i>	transportowa (4)
UDP	<i>User Datagram Protocol</i>	transportowa (4)
IPv4	<i>Internet Protocol version 4</i>	sieciowa (3)
ICMP	<i>Internet Control Message Protocol</i>	sieciowa (3)
ARP	<i>Address Resolution Protocol</i>	sieciowa (3)
IGMP	<i>Internet Group Management Protocol</i>	sieciowa (3)
PPPoE	<i>Point-to-Point Protocol over Ethernet</i>	sieciowa (3)

### 3.1.3 Mikrokontroler

Jednym z najważniejszych elementów podczas projektowania płytki drukowanej jest układ mikroprocesorowy. Powinien udostępniać odpowiednią liczbę konkretnych układów peryferijnych, umożliwiających realizację funkcjonalności wymaganych w projekcie. Należy również zadbać, aby układ posiadał odpowiedni rozmiar pamięci wystarczający do realizacji zamierzanej aplikacji. Poniżej zostanie przedstawionych kilka czynników wpływających na wydajność mikrokontrolera:

- Rodzaj architektury procesora - obecnie na rynku układy mikroprocesorowe najczęściej

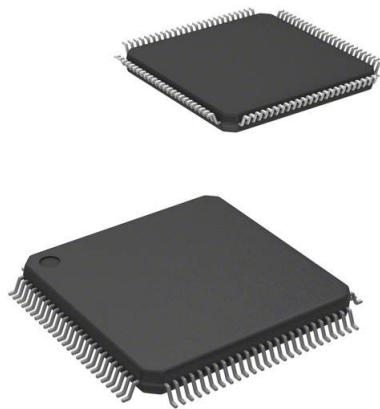
posiadają architekturę 8, 16 lub 32-bitową. Ilość bitów określa długość szyny danych w układzie. Operacje na typach danych o większej ilości bitów niż architektura systemu będą negatywnie wpływać na wydajność, a więc procesor będzie musiał wykonać większą liczbę operacji, a co się z tym wiąże, będzie potrzebował większej liczby cykli zegarowych do wykonania operacji.

- Maksymalna możliwa częstotliwość taktowania zegara - im jest większa tym procesor jest w stanie wykonać więcej operacji w określonym przedziale czasowym. Wiąże się to jednak z zwiększoną poborem prądu.
- Układy koprocesorowe - coraz to częściej występują w mikrokontrolerach i wspomagają CPU (ang. *central processing unit*). Najpopularniejszymi układami tego typu są:
  - FPU (ang. *Floating-Point Unit*) - wspomaga obliczenia na liczbach zmiennoprzecinkowych.
  - DMA (ang. *Direct Memory Access*) - umożliwia przesył danych znajdujących się zarówno w pamięci jak i tych wygenerowanych przez układy peryferyjne bez udziału CPU.

Obecnie na rynku prosperuje spora ilość producentów układów mikroprocesorowych. Niektórzy z nich rozwijają układy przeznaczone do wykorzystania w konkretnych dziedzinach przemysłu. Przykładem mogą być firmy Renesas oraz NXP posiadające portfolio układów dedykowanych do zastosowania w branży samochodowej. Charakteryzują się one obecnością peryferiów wykorzystywanych w branży, takich jak CAN (ang. *Controller Area Network*) i LIN (ang. *Local Interconnect Network*), a także certyfikacją ASIL (ang. *Automotive Safety Integrity Level*) bardzo często wymaganą w automotive. Nordic Semiconductors jest firmą specjalizującą się w produkcji układów przeznaczonych do transmisji bezprzewodowej, a Microchip i STMicroelectronics posiadają układy zarówno specjalistyczne jak i ogólnego przeznaczenia. Poza wyżej wymienionymi na rynku jest jeszcze wielu innych producentów, między innymi: Texas Instruments, Espressif Systems, Infineon.

Mikrokontrolerem wybranym do realizacji projektu został STM32L433VCT6 [17] firmy STMicroelectronics w obudowie LQFP100 (rys. 3.3). Spełnia on wszystkie potrzebne wymagania, a producent oferuje dedykowane narzędzia programistyczne ułatwiające pracę z układem. Istotnym czynnikiem wpływającym na dobór mikrokontrolera była sytuacja na rynku półprzewodników. Wiele procesorów nie było możliwych do zakupu. Rdzeń układu stanowi 32-bitowy Arm Cortex-M4 z jednostką FPU. Jego maksymalna częstotliwość pracy wynosi 80Mhz. Posiada 256 kB pamięci Flash oraz 64 kB pamięci SRAM (ang. *Static Random Access Memory*), a więc nie będzie obawy o brak miejsca podczas rozwoju oprogramowania. W realizowanym projekcie odpowiada za takie kluczowe funkcjonalności jak:

- komunikacja z jednostką do nawigacji inercyjnej poprzez protokół komunikacyjny SPI,
- komunikacja z modułem sieciowym poprzez protokół komunikacyjny SPI,



Rys 3.3. Obudowa LQFP100 (ang. *low-profile quad flat package*) [19]

- komunikacja z kartą uSD poprzez interfejs SDMMC (ang. *secure digital multimedia card*),
- udostępnienie interfejsu umożliwiającego wgrywanie nowego oprogramowania oraz debugowanie,
- udostępnienie interfejsu UART (ang. *universal asynchronous receiver-transmitter*) na potrzeby użytkownika,

## 3.2 Oprogramowanie

Prace związane z projektowaniem płytki drukowanej oraz rozwojem oprogramowania będą prowadzone przy użyciu komputera klasy PC. W tym podrozdziale zostaną przedstawione aplikacje służące dedykowanym czynnościom.

### 3.2.1 Altium Designer

Altium Designer to oprogramowanie umożliwiające projektowanie schematów obwodów elektronicznych oraz wielowarstwowych płyt drukowanych. Posiada wiele gotowych elementów (komponentów elektronicznych i footprintów) gotowych do wykorzystania w projekcie oraz pozwala na tworzenie własnych bibliotek. Sprawdza poprawność realizowanego projektu oraz umożliwia wygenerowanie pliku w formacie gerber, czyli standardowym formacie podczas zamawiania PCB.

### 3.2.2 STM32CubeIDE

STM32CubeIDE to zintegrowane środowisko programistyczne dedykowane do pracy z mikrokontrolerami STM32 firmy STMicroelectronics. Posiada wiele standardowych funkcji-

alonalności, takich jak: budowanie projektu, debugowanie, podgląd aktualnych wartości rejestrów peryferiów oraz rdzenia procesora, podgląd pamięci mikrokontrolera. Jego częścią jest narzędzie STM32CubeMX umożliwiające konfigurację układów peryferyjnych mikrokontrolera, a następnie wygenerowanie kodu z użyciem wysokopoziomowej biblioteki HAL (ang. *Hardware Abstraction Layer*).

### 3.2.3 Visual Studio Code

Visual Studio Code to rozwinięty edytor kodu źródłowego. Posiada wiele dodatków możliwych do zainstalowania wewnątrz oprogramowania. Tabela [2] prezentuje rozszerzenia wykorzystane w projekcie, które ułatwiają pracę podczas rozwijania kodu oraz umożliwiają budowanie oraz debugowanie projektu.

**Tabela 2.** Rozszerzenia oprogramowania Visual Studio Code wykorzystane w projekcie

Nazwa	Opis
Python	Wsparcie języka programowania Python
C/C++	Wsparcie języków programowania C oraz C++
Cmake Tools	Wsparcie narzędzia Cmake
Cmake	Kolorowanie składni skryptów Cmake
LinkerScript	Kolorowanie składni skryptów linkera
Arm Assembly	Kolorowanie składni assemblera
Cortex-Debug	Wsparcie debugowania dla mikrokontrolerów z rdzeniem ARM Cortex-M

### 3.2.4 MATLAB

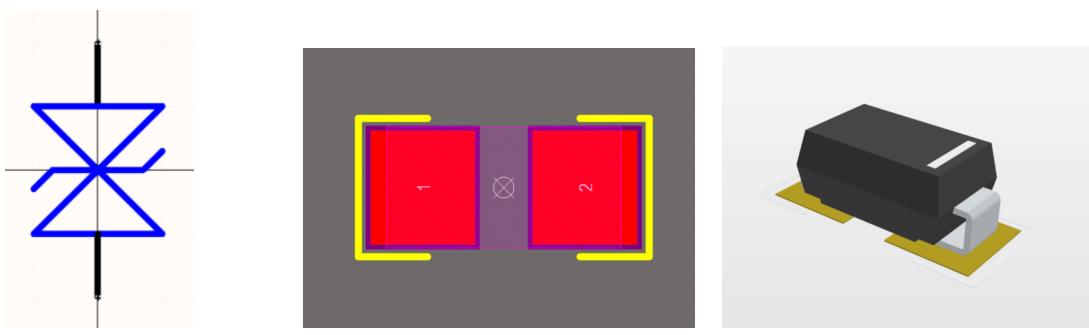
MATLAB to zarówno środowisko służące do obliczeń numerycznych jak i język programowania rozwijane przez firmę MathWorks. Posiada dużą ilość przyborników (ang. *toolbox*) rozszerzających jego funkcjonalności, jak na przykład "System Identification Toolbox" służący do modelowania systemów dynamicznych.

## 4 Warstwa sprzętowa urządzenia

W niniejszym rozdziale nastąpi omówienie najważniejszych działań podczas realizacji warstwy sprzętowej projektowanej płytki drukowanej. Będą one podzielone na trzy główne etapy: projektowanie schmatu obwodu elektronicznego, projektowanie płytki PCB oraz montaż urządzenia. Każdy z nich zostanie przedstawiony jako osobny podrozdział. Realizacja dwóch pierwszych etapów dokonana będzie przy pomocy oprogramowania Altium Designer.

### 4.1 Schemat obwodu elektronicznego

Pierwszym etapem wykonania płytki drukowanej jest zaprojektowanie schematu obwodu elektronicznego. W jego skład wchodzą wszystkie elementy elektroniczne wraz z połączeniami pomiędzy nimi. Oprogramowanie Altium Designer udostępnia zbiór bibliotek posiadających symbole konkretnych komponentów elektronicznych z dedykowanymi footprintami, czyli układem graficznym zawierającym przestrzeń zajmonaną przez komponent wraz z padami lutowniczymi oraz ich numeracją. Umożliwia również samodzielne zaprojektowanie symbolu elementu wraz z dedykowanym footprintem oraz wizualizację 3D (rys. 4.1).



(a) Symbol komponentu elektrycznego

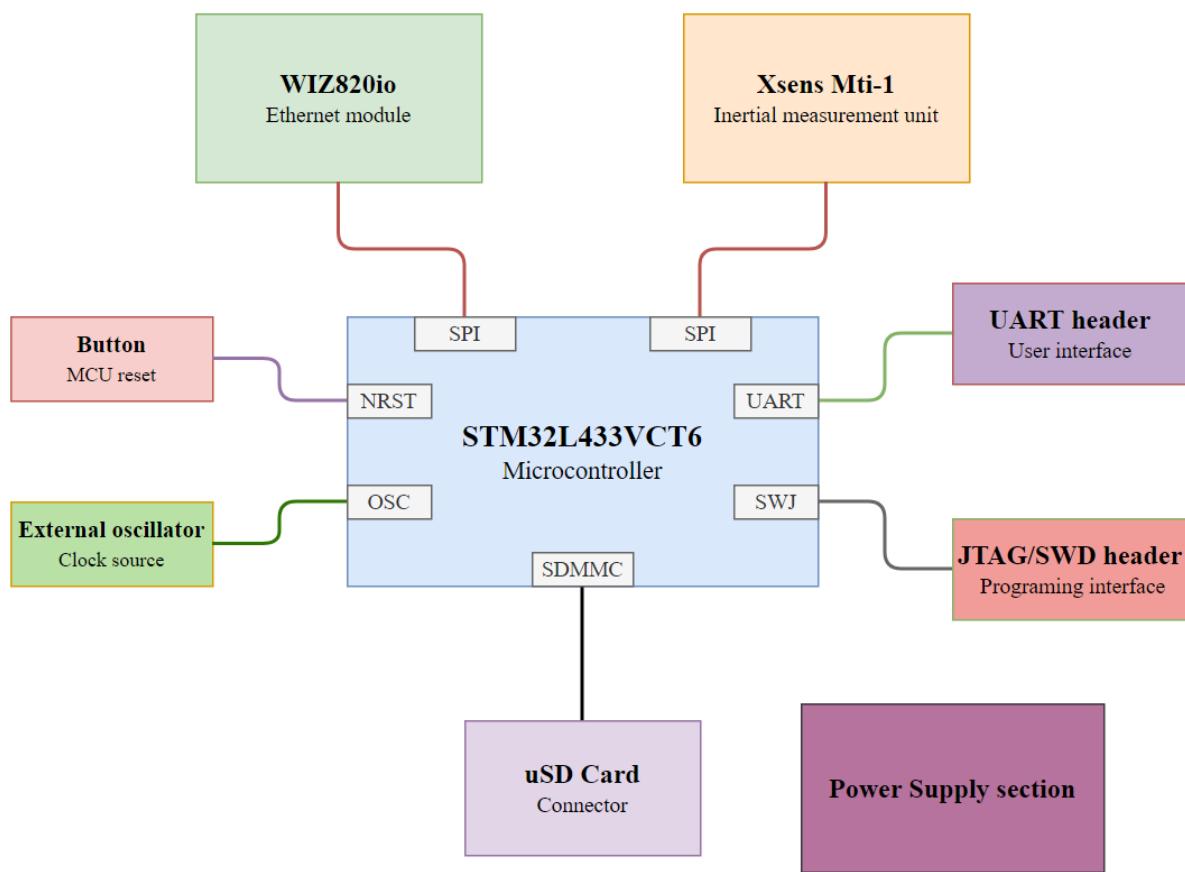
(b) Footprint 2D

(c) Wizualizacja 3D

Rys 4.1. Zaprojektowana dioda transil w oprogramowaniu Altium Designer

Ważną czynnością podczas projektowania obwodu elektronicznego jest przeczytanie not katalogowych układów scalonych wybranych podczas fazy koncepcyjnej. Zapoznanie się z nimi jest konieczne podczas doboru sąsiadujących elementów pasywnych takich jak rezystory, kondensatory czy cewki, bez których w wielu przypadkach działanie układu scalonego byłoby niemożliwe. Częstą praktyką producentów jest zamieszczenie w dokumentacji przykładowego obwodu elektronicznego dla konkretnego układu scalonego, co może okazać się bardzo pomocne podczas realizacji projektu.

Z uwagi na złożoność zagadnienia w celu zachowania czytelności cały obwód elektroniczny podczas projektowania podzielono na 4 arkusze dostępne w załączniku nr 1 do pracy. Ogólny schemat urządzenia pokazany jest na rysunku 4.2. W celu przejrzystego przedstawienia został podzielony na 5 sekcji, które będą zaprezentowane w poniższych podrozdziałach.



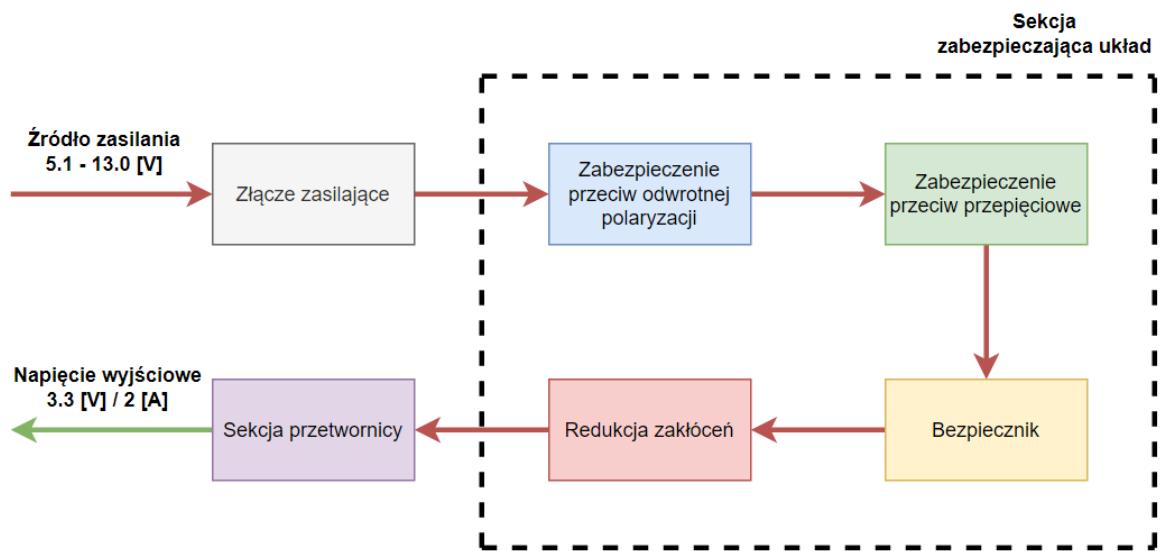
Rys 4.2. Ogólny schemat obwodu elektronicznego

#### 4.1.1 Sekcja zasilania

Elementy elektroniczne znajdujące się na płytce drukowanej mogą pracować w zakresie różnych napięć oraz pobieranych prądów. Zadaniem sekcji zasilania jest odpowiednie przetworzenie napięcia wyjściowego, aby zapewnić odpowiednią wartość oraz jakość napięcia doprowadzonego do komponentów. Odpowiada również za zabezpieczenie układu.

W ramach realizowanego projektu: mikrokontroler, IMU, moduł sieciowy oraz karta  $\mu$ SD wymagają dostarczenia napięcia o wartości 3.3 V. W związku z tym w układzie wystąpi tylko jedna przetwornica imulsowa odpowiedzialna za konwersję napięcia przy niewielkiej stracie mocy. Sekcja zasilania powinna być tak dostosowana, aby zapewniała możliwość zasilenia płytki PCB za pomocą zasilacza impulsowego 12V oraz trójogniwowego akumulatora litowo-polimerowego, którego napięcie wyjściowe przy pełnym naładowaniu wynosi 12.6 V.

Na rysunku 4.3 został zaprezentowany schemat blokowy sekcji zasilania. Tranzystor P-MOSFET odpowiedzialny jest za protekcję układu w momencie nieprawidłowego podłączenia zasilania do płytki PCB przez użytkownika. Chroni w przypadku odwrotnego wpięcia przewodów do złącza zasilającego. Dioda TVS (ang. *Transient Voltage Suppressor*) zabezpiecza



Rys 4.3. Schemat sekcji zasilania układu

przed przepięciami, w szczególności wyłodowaniami elektrostatycznymi. Jej maksymalne napięcie wsteczne zostało dobrane na poziomie 13 V, co definiuje maksymalne napięcie zasilania układu. Ważnym elementem podczas projektowania sekcji zasilania jest bezpiecznik chroniący w sytuacji wystąpienia zbyt wysokiej wartości natężenia prądu. Został dobrany w oparciu o maksymalną jego konsumpcję poszczególnych elementów deklarowaną w notach katalogowych. Tabela [3] przedstawia uwzględnione części składowe podczas obliczeń. Dodano również około 35 procentowy zapas, co spowodowało wybór bezpiecznika o prądzie znamionowym 500 mA.

Tabela 3. Maksymalny pobór prądu poszczególnych elementów układu

Element	Pobór prądu
Moduł sieciowy	175 mA
IMU	50 mA
Karta μSD	100 mA
Mikrokontroler	35 mA
Dioda LED x 2	7 mA
<b>Łącznie</b>	<b>367 mA</b>

Aby dodatkowo zabezpieczyć układ podczas stosowania zasilacza impulsowego jako źródło zasilania dodano dławik CMC (ang. *Common Mode Choke*), którego zadaniem jest redukcja zakłóceń. Każdy z wyżej opisanych elementów zabezpieczających został dobrany z należytą starannością w oparciu o jego poszczególne charakterystyczne parametry.

Za obwodem zabezpieczającym układ występuje przetwornica impulsowa typu *buck*, czyli

obniżająca napięcie. Wybrano układ scalony firmy Texas Instruments o numerze TPS54228D. Kluczowymi parametrami decydującymi o jej wyborze są: napięcie wejściowe zasilania, możliwe napięcie wyjściowe oraz maksymalny prąd pracy układu (rys. 4.4).

### 6.3 Recommended Operating Conditions

over operating free-air temperature range and  $V_{IN} = 12$  V (unless otherwise noted)

		MIN	NOM	MAX	UNIT
$V_{IN}$	Supply input voltage	4.5		18	V
$V_I$	VBST	-0.1		24	V
	VBST (transient, 10 ns)	-0.1		27	
	VBST(vs SW)	-0.1		5.7	
	SS	-0.1		5.7	
	EN	-0.1		18	
	VFB	-0.1		5.5	
	SW	-1.8		18	
	SW (transient, 10 ns)	-3		21	
	GND	-0.1		0.1	
$V_O$	Output voltage, VREG5	-0.1		5.7	V
$I_O$	Output current, $I_{VREG5}$	0		10	mA
$I_{OUT}$	Operating output current <sup>(1)</sup>			2	A
$T_A$	Operating free-air temperature	-40		85	°C

Rys 4.4. Fragment noty katalogowej układu TPS54228D [20]

Warunkiem do prawidłowego działania przetwornicy jest odpowiedni dobór elementów pasywnych ją otaczających, między innymi: kondensatorów filtrujących znajdujących się na wejściu przetwornicy, rezystorów definiujących napięcie wyjściowe (wzór 1), filtra LC znajdującego się na wyjściu przetwornicy.

$$V_{OUT} = 0.765 * \left(1 + \frac{R_1}{R_2}\right) \quad (1)$$

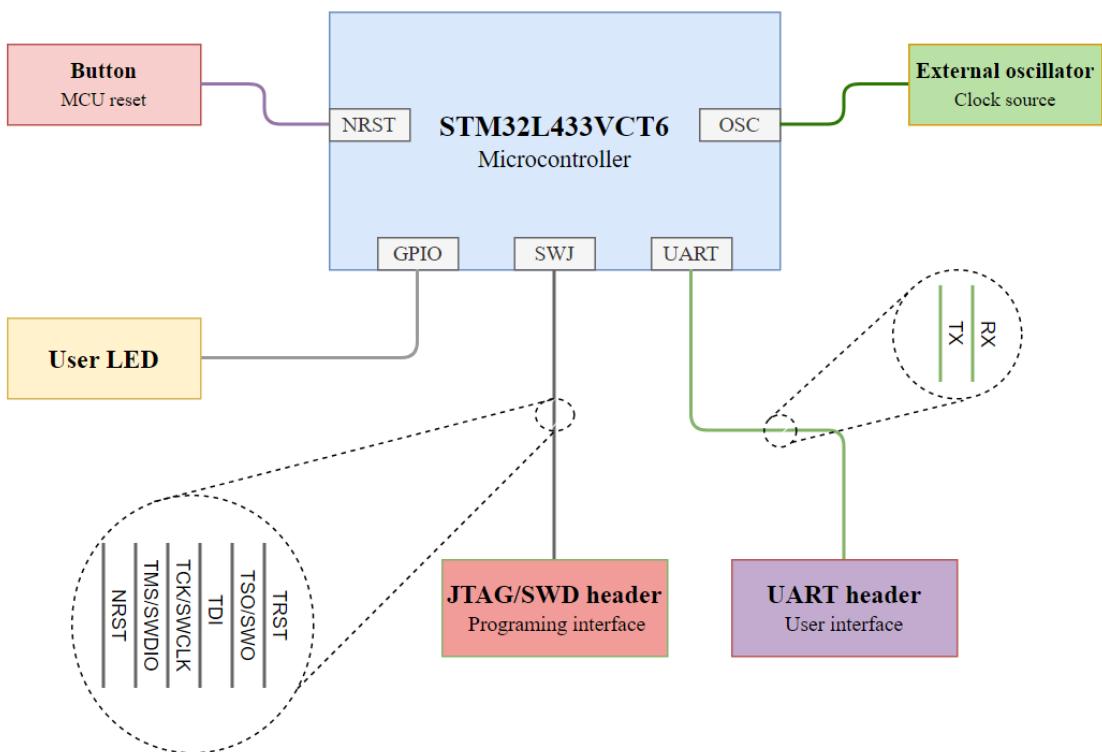
Zgodnie z dokumentacją przetwornicy (wzór 2) zostało dobrane minimalne napięcie wejściowe układu na poziomie 5.1 V.

$$V_{IN} = V_O / 0.65 \quad (2)$$

Za przetwornicą została umieszczona dioda led o kolorze zielonym informująca, że układ jest zasilony.

#### 4.1.2 Sekcja mikrokontrolera

Sercem płytki drukowanej jest mikrokontroler STM32L433VCT6, którego najważniejszym zadaniem jest komunikacja z poszczególnymi układami za pomocą interfejsów komunikacyjnych. Jako, że jednostce do nawigacji inercyjnej, modułowi sieciowemu oraz karcie μSD będą przeznaczone kolejne podrozdziały, w tym nie zostanie przedstawiony ich szczegółowy opis. Schemat sekcji mikrokontrolera przedstawiony jest na rysunku 4.5.



Rys 4.5. Schmat sekcji mikrokontrolera

Wgrywanie oprogramowania na mikrokontroler jest możliwe dzięki wyprowadzeniu złącza interfejsu SWJ (ang. *Serial Wire / JTAG*). Pozwala on na komunikację przy pomocy dwóch popularnych interfejsów. Pierwszym z nich jest JTAG (ang. *Joint Test Action Group*) [21]. Poza wgrywaniem oraz debugowaniem oprogramowania umożliwia testowanie wielu płyt drukowanych oraz układów scalonych jednocześnie poprzez zastosowanie technologii *boundary-scan*. Komunikacja odbywa się za pomocą co najmniej czterech linii sygnałowych:

- TDI (ang. *Test Data Input*) - wejście danych testowych,
- TDO (ang. *Test Data Output*) - wyjście danych testowych,
- TCK (ang. *Test Clock*) - sygnał zegarowy,
- TMS (ang. *Test Mode Select*) - wybór trybu testowania,
- TRST (ang. *Test Reset*) - asynchroniczny reset (opcjonalnie).

Drugi udostępniony interfejs to SWD (ang. *Serial Wire Debug*). Jest on dedykowany do wgrywania oraz debugowania oprogramowania na platformy oparte o architekturę ARM. Jego zaletą w porównaniu do JTAG-a jest możliwość komunikacji przy zastosowaniu mniejszej ilości sygnałów (minimum dwóch), przedstawionych poniżej:

- SWDIO (ang. *Serial Wire Data Input Output*) - wejście oraz wyjście danych,
- SWCLK (ang. *Serial Wire Clock*) - sygnał zegarowy,
- SWO (ang. *Serial Wire Output*) - wyjście debugowe (opcjonalnie).

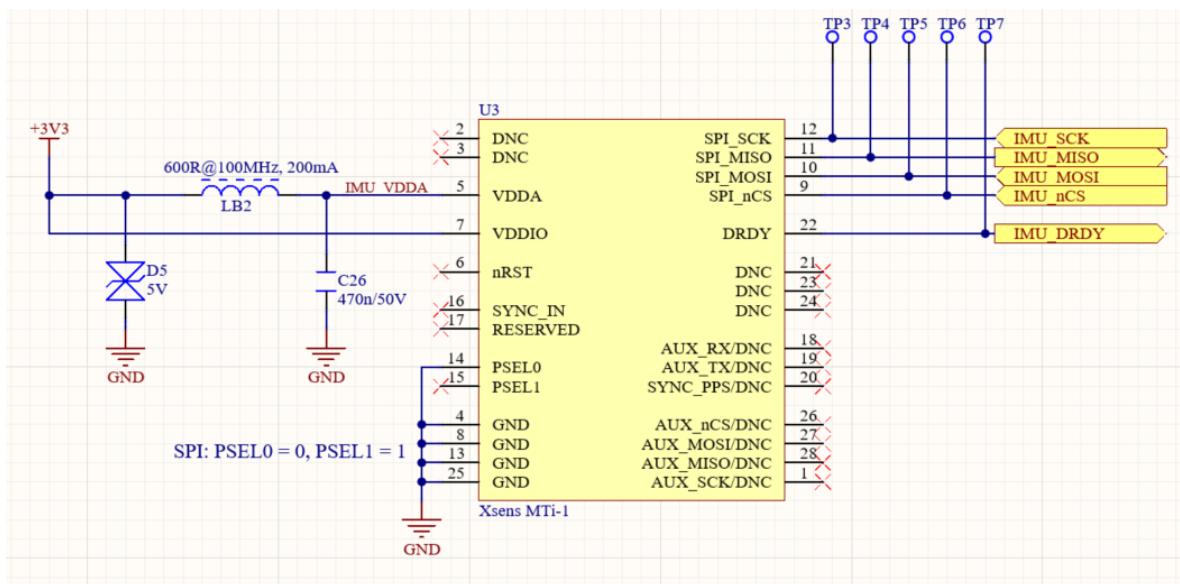
Z mikrokontrolera zostały wyprowadzone linie sygnałowe interfejsu komunikacyjnego UART. Nie ma on dedykowanego przeznaczenia w projekcie, ale może być wykorzystany w celach debugowych przez użytkownika lub służyć do komunikacji z zewnętrznym urządzeniem.

Do dedykowanych pinów mikrokontrolera został podłączony rezonator kwarcowy o częstotliwości taktowania 8 MHz. Zasila on układ HSE (ang. *high-speed external*), czyli zewnętrzny układ zegarowy wysokiej częstotliwości. Jest to o wiele precyzyjniejsze rozwiązanie niż zastosowanie wewnętrznego oscylatora znajdującego się w układzie scalonym mikrokontrolera.

Do jednego z wejść GPIO (ang. *general-purpose input/output*) układu mikroprocesorowego została podłączona żółta dioda led. Przeznaczona jest na potrzeby użytkownika. Mikrokontroler można zresetować za pomocą dedykowanego przycisku.

#### 4.1.3 Sekcja IMU

Urządzeniem odpowiedzialnym za pomiar danych umożliwiających identyfikację orientacji przestrzennej jest jednostka do nawigacji inercyjnej Xsens MTi-1. Fragment schematu elektronicznego zawierający jego sekcję przedstawiony został na rysunku 4.6.



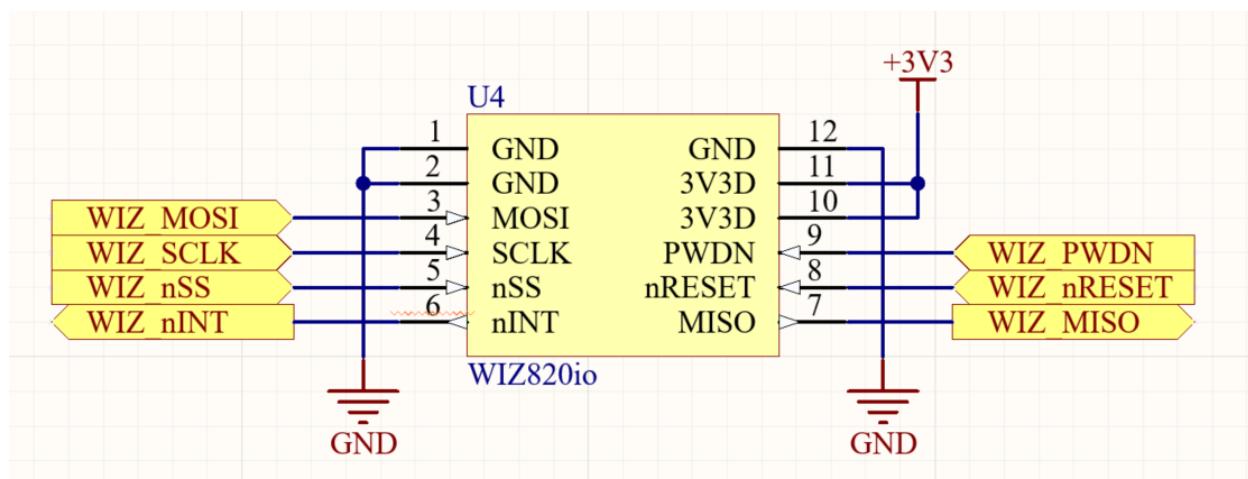
Rys 4.6. Sekcja IMU - fragment schematu elektronicznego

Wybór protokołu do komunikacji IMU z procesorem odbywa się poprzez odpowiednie ustawienie stanu logicznego na dwóch pinach PSEL urządzenia. Posiadają one wewnętrzne podciagnięcie do stanu wysokiego (ang. *pull-up*) poprzez rezystor ( $30\text{ k}\Omega - 50\text{ k}\Omega$ ). Wejście PSEL0 zostało zwarte do masy, co umożliwia transmisję danych poprzez protokół SPI. Dodatkowym sygnałem wychodzącym z Xsens-a jest DRDY (ang. *data ready*), którego domyślnie stan wysoki świadczy, że najnowsze dane są gotowe do odczytania. Zgodnie z notą katalogową odpowiedzialną za integrację sprzętową [22] została zastonowana filtracja zasilania urządzenia.

W jej skład wchodzi kondensator odsprzęglający oraz koralik ferrytowy odpowiedzialny za tłumienie sygnałów o wysokiej częstotliwości na linii zasilania. Dodatkowym zabezpieczeniem jest transil o maksymalnym napięciu wstecznym 5 V.

#### 4.1.4 Sekcja modułu sieciowego

Moduł WIZ820io zawierający chip W5200 firmy WIZnet odpowiada za komunikację z urządzeniem sprzętowym stosu TCP/IP. Fragment obwodu elektronicznego zawierający jego sekcję znajduje się na rysunku 4.7.



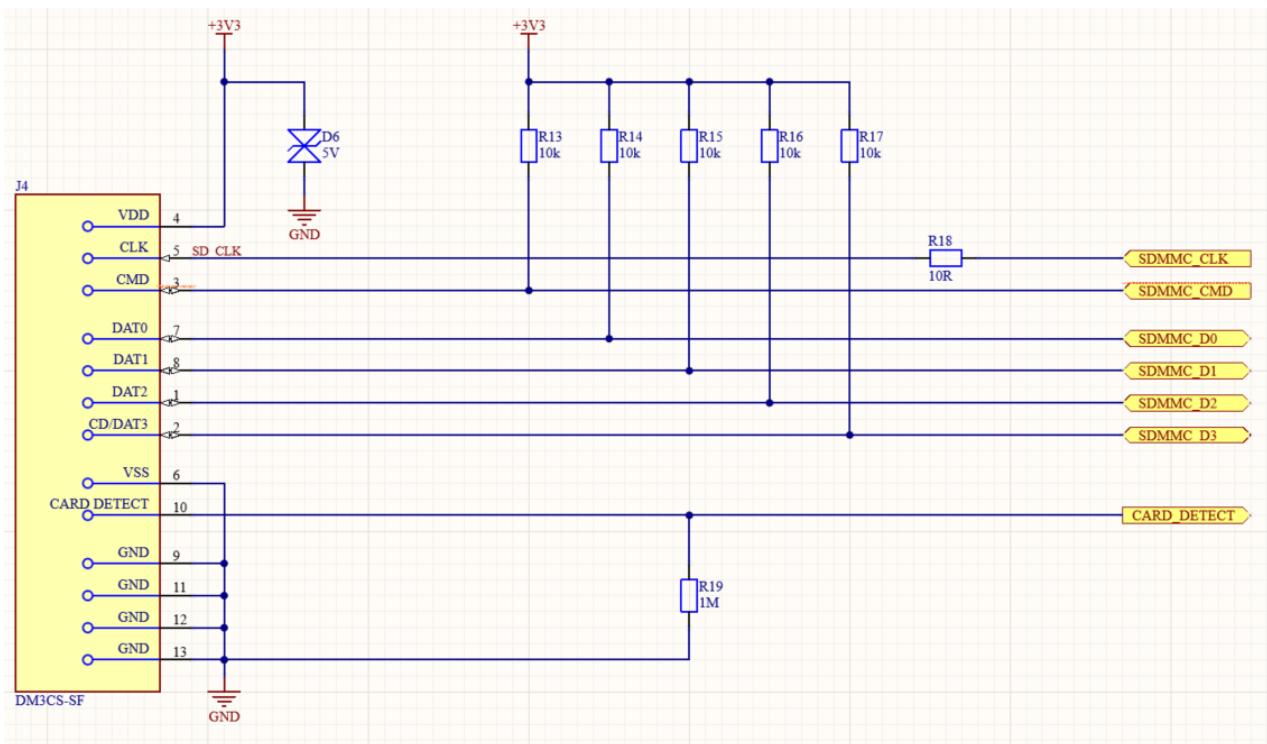
Rys 4.7. Sekcja modułu sieciowego - fragment schematu elektronicznego

Tak jak w przypadku jednostki do nawigacji inercyjnej komunikacja z modułem sieciowym odbywa się za pomocą protokołu komunikacyjnego SPI. Poza sygnałami transmitującymi dane WIZ820io posiada kilka dodatkowych połączeń z mikrokontrolerem:

- nINT - sygnał wyjściowy, stan niski informuje o przerwaniu.
- PWDN - sygnał wejściowy, stan wysoki wprowadza urządzenie w stan wyłączenia.
- nRESET - sygnał wejściowy, stan niski inicializuje/reinicializuje moduł.

#### 4.1.5 Sekcja karty μSD

Umieszczenie złącza na kartę μSD na płytce PCB pozwala na przechowanie ogromnej ilości danych zebranych za pomocą IMU. Mikrokontroler zastosowany w projekcie posiada 256 kB pamięci Flash oraz 64 kB pamięci SRAM co w porównaniu z wielogigabajtowymi pojemnościami kart pamięci dostępnych na rynku jest niewielką ilością. Rysunek 4.8 przedstawia zaprojektowaną sekcję karty μSD.



Rys 4.8. Sekcja karty µSD - fragment schematu elektronicznego

Komunikacja układu mikroprocesorowego z kartą µSD odbywa się za pomocą protokołu komunikacyjnego SDMMC, w którego skład wchodzą poniższe linie sygnałowe:

- D0/D1/D2/D3 - sygnały transmisji danych,
- CLK - sygnał zegarowy,
- CMD - sygnał komand.

Każdy z sygnałów poza zegarowym jest podcięgnięty poprzez rezystor  $10\text{k}\Omega$  do stanu wysokiego. Definiują one poziom logiczny w stanie bezczynności, gdyż wtedy wyjścia mikrokontrolera są w stanie wysokiej impedancji. Rezystor o niewielkiej wartości włączony szeregowo do sygnału zegarowego zapobiega tzw. efektowi dzwonienia (ang. *ringing*). Dodatkowym sygnałem pomiędzy złączem karty, a procesorem jest CARD\_DETECT, który przyjmuje stan wysoki w momencie wsunięcia karty do złącza. Spowodowane jest to rezystorem podciągającym o wartości około  $50\text{k}\Omega$  zawartym w karcie uSD przyłączonym do odpowiedniej lini sygnałowej. Aby nie doprowadzić do stanu pływającego (ang. *floating*) na lini zastosowano podcięgnięcie do masy z użyciem rezystora  $1\text{M}\Omega$ . Jako zabezpieczenie na wejściu zasilania znajduje się transil o maksymalnym napięciu wstecznym 5 V.

## 4.2 Schemat układu PCB

Kompletna realizacja schematu obwodu elektronicznego umożliwia przejście do kolejnej fazy realizacji płytki drukowanej, czyli projektowania układu (ang. *layout*) PCB. W jego skład wchodzą między innymi: pola lutownicze na elementy SMD, otwory na elementy THT i montażowe oraz połączenia elektryczne pomiędzy komponentami. Rozmieszczenie komponentów oraz ścieżek powinno być dostosowane do zdolności produkcyjnych udostępnionych przez firmę wykonującą płytę. Tabela [4] przedstawia kilka wybranych kluczowych ograniczeń, które należy mieć na uwadze podczas zamawiania czterowarstwowych płyt drukowanych za pośrednictwem chińskiej firmy JLCPCB. Jej to zostało zlecone wykonane PCB podczas realizacji projektu. Ograniczenia wynikają ze zdolności oraz dokładności urządzeń znajdujących się w parku maszynowym firmy.

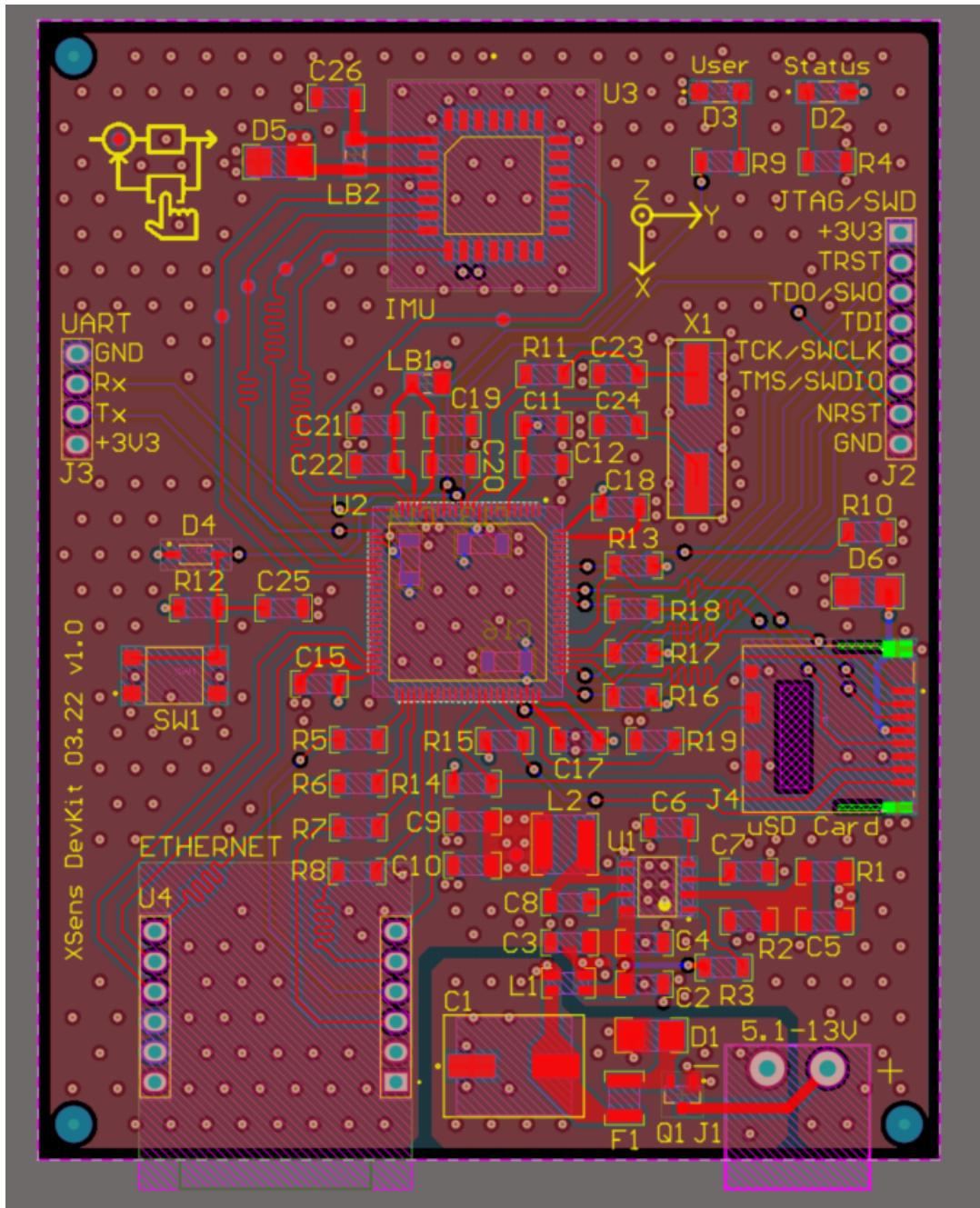
**Tabela 4.** Zdolności produkcyjne płyt drukowanych przez firmę JLCPCB

Parametr	Zdolność produkcyjna
Rozmiar otworu	0.20 - 6.30 mm
Minimalna odległość pomiędzy otworami	0.5 mm
Minimalna odległość pomiędzy polami lutowniczymi	0.127 mm
Minimalna odległość pomiędzy ścieżkami	0.09 mm
Minimalna szerokość ścieżki	0.09 mm

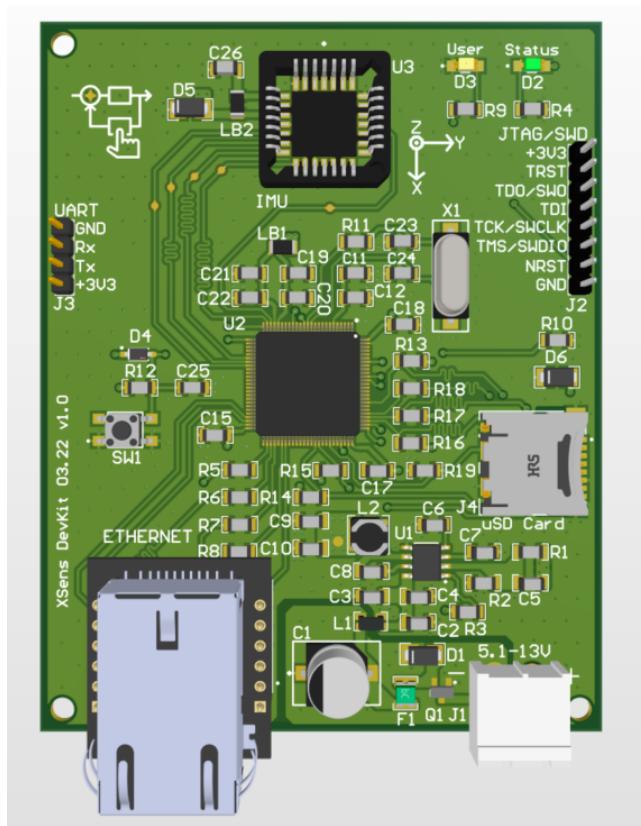
Rysunki 4.9 - 4.11 prezentują dwu oraz trójwymiarowy model PCB z oprogramowaniem Altium Designer. Pierwszy pokazuje widok roboczy podczas projektowania *layoutu*. Rozmieszczenie komponentów oraz połączeń pomiędzy nimi zostało zrealizowane w oparciu o dokumentacje techniczne kluczowych komponentów oraz dobre praktyki projektowania PCB.

Ilość komponentów na płytce drukowanej i związana z tym ilość połączeń pomiędzy nimi wpływała na realizację projektu w oparciu o płytę składającą się z czterech warstw miedzi, gdzie każda z nich oddzielona jest materiałem izolacyjnym. Poniżej przedstawione jest przeznaczenie każdej z warstw:

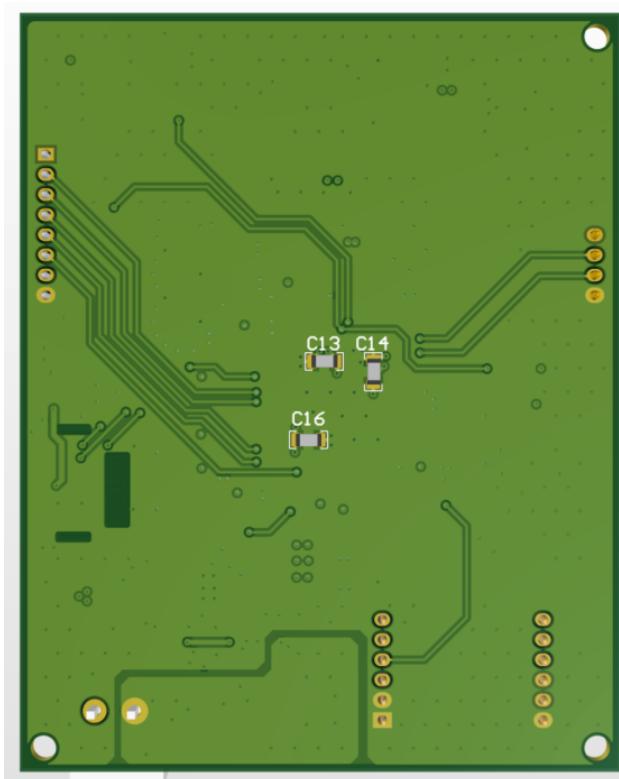
- warstwa 1: pierwsza warstwa sygnałowa,
- warstwa 2: warstwa masy (GND),
- warstwa 3: warstwa zasilania (+3V3),
- warstwa 4: druga warstwa sygnałowa.



Rys 4.9. Wizualizacja płytki drukowanej - widok 2D



Rys 4.10. Wizualizacja płytki drukowanej - widok 3D (przód)

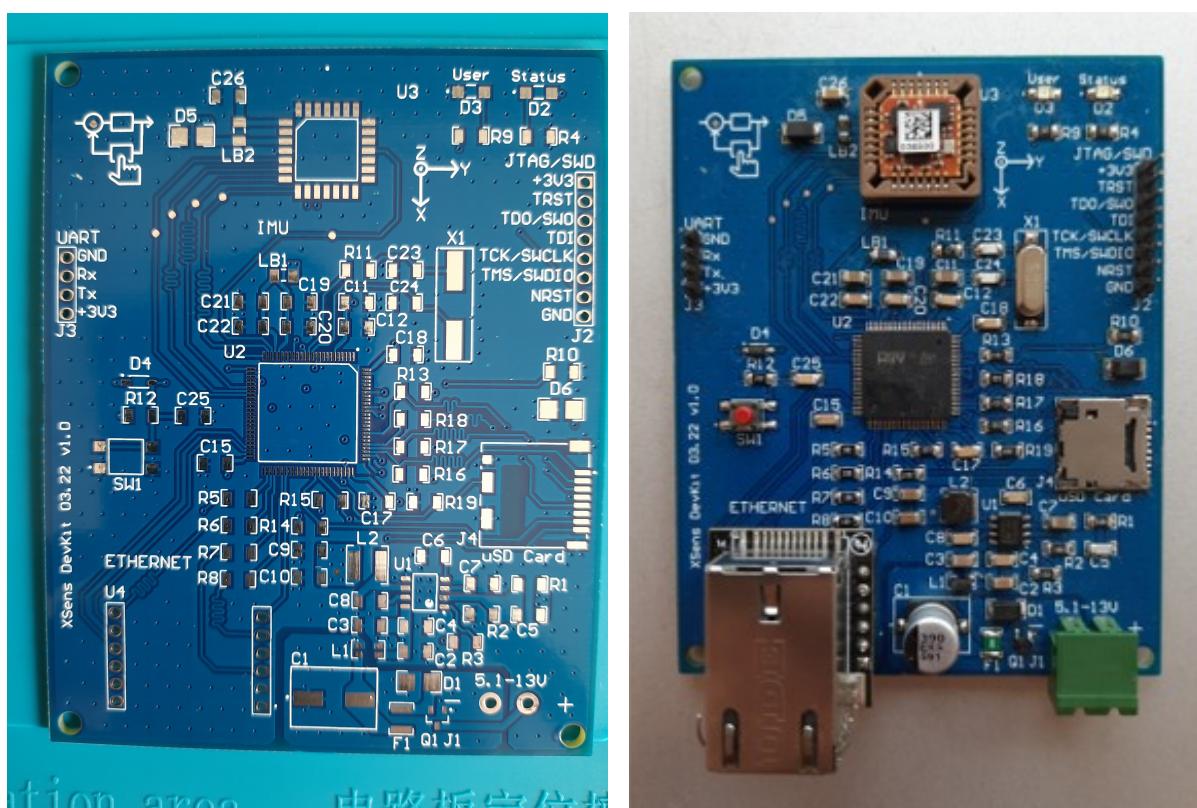


Rys 4.11. Wizualizacja płytki drukowanej - widok 3D (tył)

### 4.3 Montaż PCB

Kolejnym etapem po zakończeniu projektowania było zamówienie wykonania PCB w chińskiej firmie JLCPCB. Oprogramowanie Altium Designer umożliwia wyeksportowanie danych o zaprojektowanej płytce do plików z rozszerzeniem gerber (.GBR). Przy ich pomocy firma wykonująca zlecenie może w łatwy sposób odtworzyć wszystkie informacje potrzebne do jej wyprodukowania. Zestawienie materiałowe do wykonania PCB znajduje się w załączniku nr 2 do pracy.

Rysunek 4.12 prezentuje zrealizowaną płytę drukowaną przed oraz po procesie montażu komponentów elektronicznych. Zdecydowana większość elementów została przylutowana za pomocą standardowej lutownicy kolbowej z użyciem drutu lutowniczego o 60-procentowej zawartości cyny oraz 40-procentowej zawartości ołowiu. Podczas montażu kilku komponentów została zastosowana technika lutowania gorącym powietrzem z użyciem pasty lutowniczej. W ramach pracy dokonano pełnego montażu na trzech płytach drukowanych.



(a) PCB przed montażem komponentów

(b) PCB po zamontowaniu komponentów

Rys 4.12. Zrealizowana płytka drukowana

## 5 Warstwa oprogramowania urządzenia

Oprogramowanie zrealizowane w ramach projektu można podzielić na dwie części. Pierwsza z nich to oprogramowanie dedykowane na platformę wbudowaną, czyli mikrokontroler STM32L433VCT6. Głównym zadaniem układu mikroprocesorowego jest konfiguracja systemu oraz odpowiednia komunikacja z jednostką do nawigacji inercyjnej i modułem sieciowym. Druga część to aplikacja na komputer klasy PC, która realizuje zadanie komunikacji z zaprojektowaną płytą drukowaną. Poniższe rozdziały szczegółowo opiszą zrealizowane oprogramowanie.

### 5.1 Oprogramowanie na platformę wbudowaną

Cykl powstawania oprogramowania realizowanego w języku C dedykowanego na platformę wbudowaną można podzielić na kilka poniżej przedstawionych etapów:

- wygenerowanie kodu z oprogramowania STM32CubeIDE,
- konfiguracja narzędzia do automatyzacji procesu budowania i komplikacji projektu,
- rozwój oprogramowania do komunikacji z IMU,
- konfiguracja modułu sieciowego oraz RTOS-a (ang. *real-time operating system*),
- rozwój wysokopoziomowej warstwy aplikacyjnej oprogramowania.

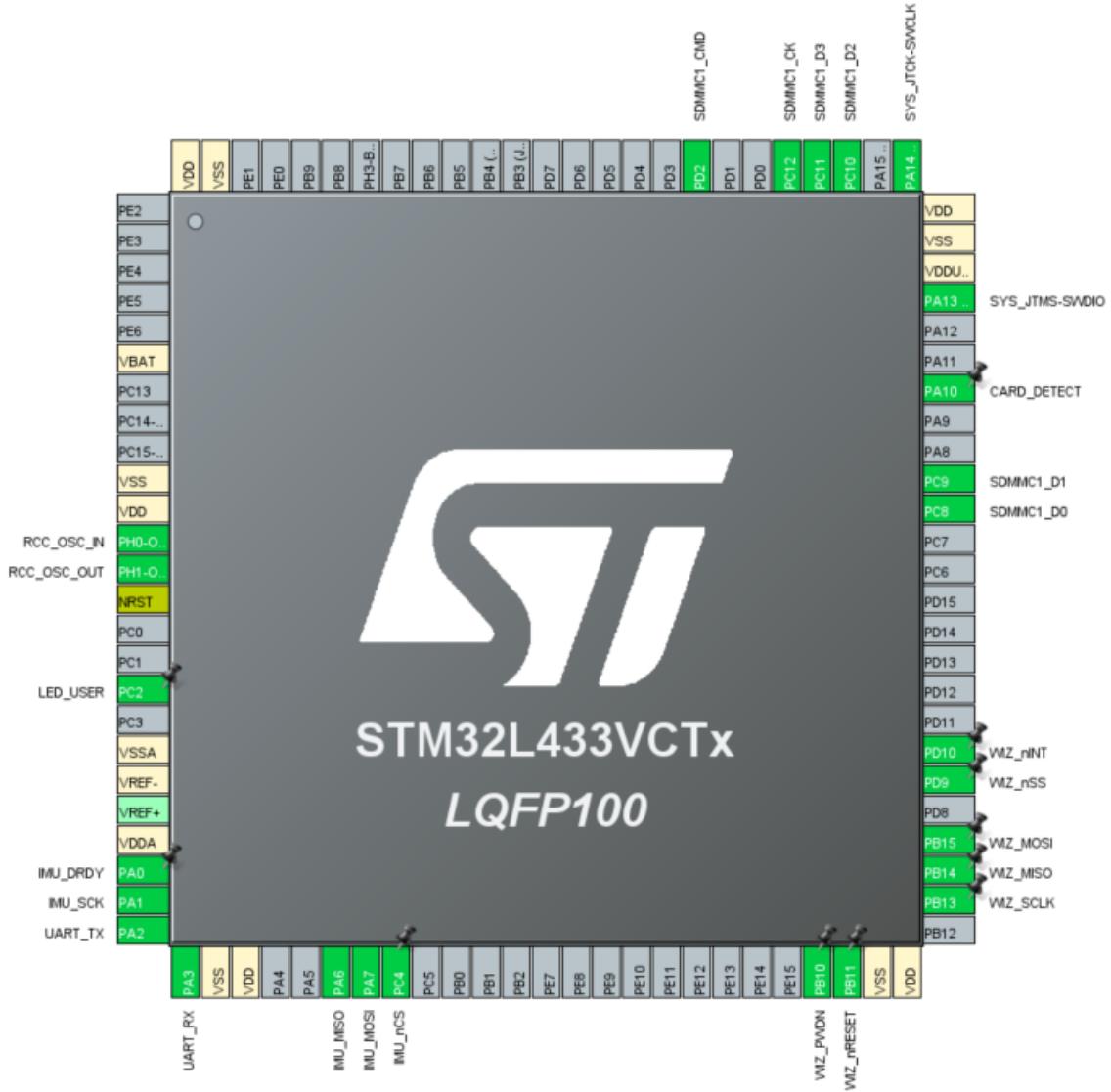
W dalszej części pracy niektóre z nich zostaną rozłożone na czynniki pierwsze. Ażeby uniknąć opisywania każdej linijki kodu powstałego oprogramowania, zaprezentowana zostanie architektura oraz działanie zrealizowanego systemu.

#### 5.1.1 Generowanie kodu

Oprogramowanie STM32CubeIDE zostało wykorzystane do wygenerowania części kodu zastosowanego w ramach projektu:

- Biblioteka CMSIS (ang. *Common Microcontroller Software Interface Standard*) udostępniająca pliki nagłówkowe o rozszerzeniu ".h". Opisują adresy rejestrów rdzenia oraz peryferiów mikrokontrolera. Umożliwiają one odwoływanie się do konkretnych peryferiów bez znajomości ich konkretnego adresu w pamięci układu.
- Biblioteka HAL dla mikrokontrolerów STM32 rodziny L4 udostępnia interfejsy stanowiące warstwę abstrakcji, umożliwiające konfigurację oraz komunikację z peryferiami układu mikroprocesorowego. Oparta jest na bibliotece CMSIS.
- Pliki konfigurujące wykorzystywane układy peryferyjne mikrokontrolera oparte na bibliotece HAL.
- Skrypt linkera oraz plik asemblerowy "startup" rozpoczynający pracę procesora po jego restarcie.

Przy pomocy wbudowanego narzędza STM32CubeMX odbyła się konfiguracja mikrokontrolera. Wraz z nią do poszczególnych układów peryferyjnych zostały przypisane piny GPIO przedstawione na rysunku 5.1.

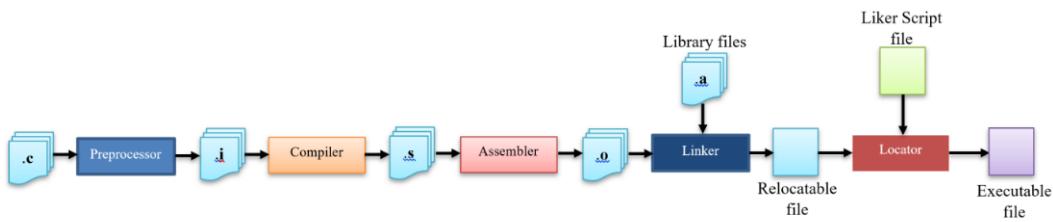


Rys 5.1. Konfiguracja pinów mikrokontrolera w oprogramowaniu STM32CubeIDE

### 5.1.2 Automatyzacja procesu budowania projektu

Aby napisany kod mógł znaleźć się w mikrokontrolerze jako wsad, należy zbudować projekt w celu otrzymania pliku wykonawczego, a następnie wgrać go na platformę wbudowaną za pomocą programatora sprzętowego. Schemat procesu budowania projektu w języku C został przedstawiony na rysunku 5.2.

Nowoczesne środowiska programistyczne (np. STM32CubeIDE, S32 Design Studio IDE, MPLAB X IDE) posiadają wewnątrz wbudowane narzędzia typu make, które automatyzują



Rys 5.2. Schemat procesu budowania projektu w języku programowania C [23]

i przyspieszają proces komilacji projektu. Ich zaletą jest fakt, że użytkownik po napisaniu oprogramowania może bezpośrednio przejść do procesu wgrywania wsadu na mikrokontroler, a budowanie najczęściej odwala się poprzez wcisnięcie jednego przycisku w środowisku graficznym. Procesem budowania projektu zajmuje się IDE, które automatyczne generuje wymagane skrypty, podpina właściwy kompilator oraz plik linkera. Wadą takiego rozwiązania jest uniezależnienie się użytkownika od środowiska programistycznego oraz brak świadomości czynności wykonywanych przez IDE.

W ramach realizowanego projektu część oprogramowania została wygenerowana, co zostało opisane w poprzednim podrozdziale. Następnie nastąpiło uniezależnienie się od środowiska programistycznego poprzez zastosowanie narzędzia *Cmake*, służącego do automatycznego zarządzania procesem komplikacji programu. Został napisany skrypt *CMakeList.txt*, w którym zdefiniowane zostały:

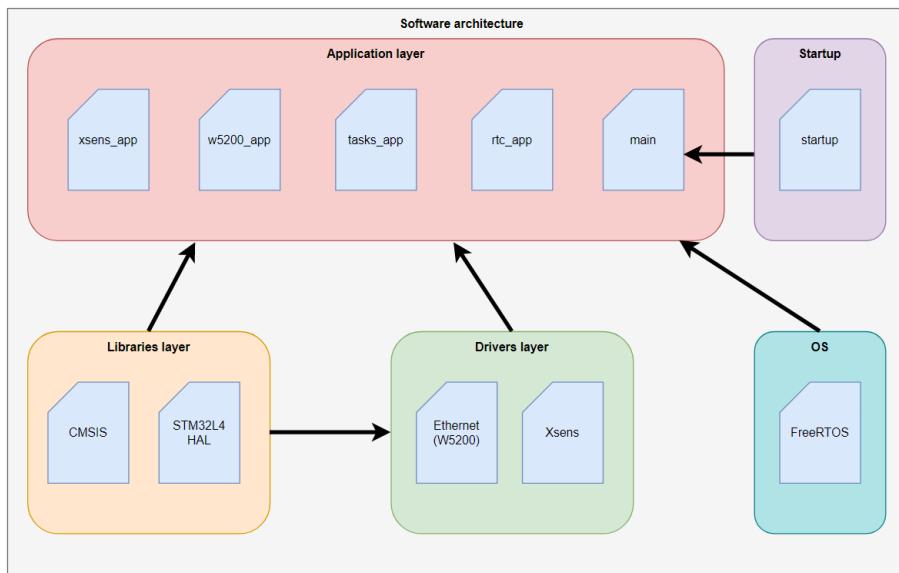
- parametry budowania projektu, między innymi: standard języka programowania, obecność oraz parametry FPU,
- lokalizacja skryptu linkera,
- lokalizacje plików źródłowych do komplikacji,
- lokalizacje plików nagłówkowych,
- symbole dyrektywy preprocesora,
- opcje kompilatora oraz linkera.

W trakcie rozwijania projektu skrypt był poszerzany poprzez dodawanie lokalizacji nowo powstających plików źródłowych oraz nagłówkowych. Dalszy rozwój oprogramowania prowadzono przy pomocy edytora kodu Visual Studio Code.

### 5.1.3 Architektura oprogramowania

Oprogramowanie zrealizowane w ramach projektu można podzielić na kilka części. Ich wizualizacja została przedstawiona na rysunku 5.3.

Warstwa bibliotek odpowiedzialna jest za ułatwienie rozwoju oprogramowania. W jej skład wchodzą biblioteki CMSIS oraz HAL dla mikrokontrolerów STM32 rodziny L4



Rys 5.3. Architekura oprogramowania

System operacyjny czasu rzeczywistego FreeRTOS umożliwia podzielenie realizowanej aplikacji na zadania, które wywoływanie są cyklicznie w reżimie czasu rzeczywistego. Udostępnia również wiele innych funkcjonalności przydatnych podczas działania na wielu wątkach (niekoniecznie wykonujących się równocześnie), takich jak: kolejki, notyfikacje, samafory, muteksy.

Zadaniem warstwy sterowników jest komunikacja z zewnętrznymi urządzeniami oraz udostępnienie wysokopoziomowych interfejsów warstwie aplikacyjnej. Każdy z modułów posiada dedykowane pliki konfiguracyjne umożliwiające szybką zmianę parametrów urządzeń. W jej skład wchodzą sterownik modułu sieciowego oraz jednostki do nawigacji inercyjnej.

Plik startup napisany w języku Asembler odpowiedzialny jest za kilka zadań realizowanych po restarcie mikrokontrolera, czyli na samym początku wykonywanego kodu:

- inicializacja adresów początkowych dla regionów pamięci określonych w skrypcie linkera,
- zdefiniowanie domyślej podrutyny występującej w momencie przyjścia przerwania odpowiedzialnego za reset mikrokontrolera,
- zdefiniowanie domyślej podrutyny występującej w momencie przyjścia przerwania, którego obsługa nie jest zdefiniowana przez użytkownika,
- inicializacja systemu zegarowego,
- ustawienie wskaźnika stosu,
- wywołanie punktu wejścia do aplikacji - funkcji main,
- zdefiniowanie prototypów funkcji z atrybutem *weak* dla całej tablicy wektorów przerwań.

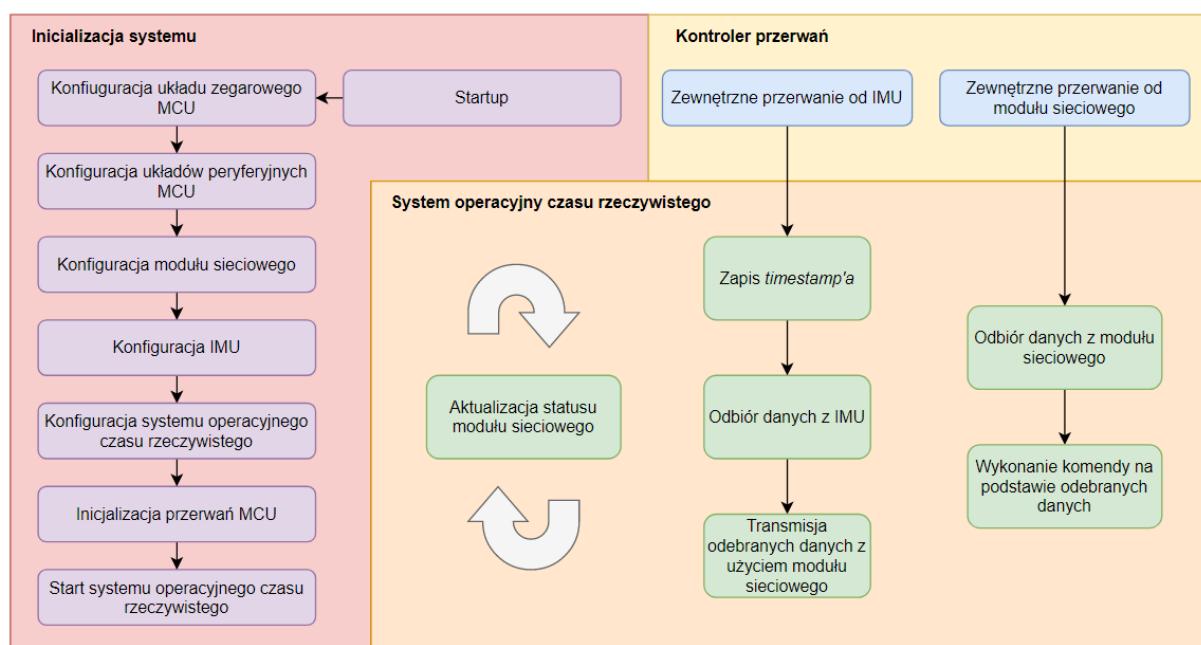
Użytkownik ma możliwość ich nadpisania poprzez zdefiniowanie serwisów obsługi przerwań.

Zadaniem warstwy aplikacyjnej jest wysokopoziomowe zarządzanie całym systemem. Poniżej przedstawione są zadania poszczególnych modułów:

- xsens\_app - inicializacja, resetowanie oraz aktualizacja konfiguracji IMU,
- w5200\_app - inicializowanie, resetowanie oraz komunikacja z modułem sieciowym,
- rtc\_app - aktualizacja czasu układu RTC zawartego w mikrokontrolerze (ang. *real-time clock*),
- tasks\_app - inicializacja oraz definicja zadań dla systemu operacyjnego czasu rzeczywistego,
- main - inicializacja poszczególnych peryferiów mikrokontrolera, inicialicacja przerwań, inicializacja systemu oraz załącznie dyspozytora (ang. *scheduler*) RTOS-a.

#### 5.1.4 Praca systemu

Uproszczony schemat pracy systemu przedstawiony jest na rysunku 5.4. Początkowym zadaniem układu mikroprocesorowego po restarcie jest inicializacja oraz konfiguracja wszystkich podrzędnych układów peryferyjnych oraz modułów oprogramowania. Następnie mikrokontroler przechodzi w tryb pracy oparty o system operacyjny czasu rzeczywistego. Wystąpienie zewnętrznego przerwania od IMU natychmiastowo rozpoczyna procedurę odbioru danych, po czym ich transmisję za pomocą modułu sieciowego z użyciem protokołu TCP. Zewnętrzne przerwanie od modułu sieciowego informuje o odbiorze danych. Kolejną czynnością jest ich interpretacja, która umożliwia na odpowiednią reakcję systemu (np. aktualizacja konfiguracji jednostki do nawigacji inercyjnej, reset czasu układu RTC).



Rys 5.4. Schemat działania systemu

## 5.2 Aplikacja do akwizycji danych

W celu łatwego nawiązania komunikacji z zaprojektowaną płytą drukowaną powstała dedykowana aplikacja napisana w języku Python. W ramach pracy wykorzystano poniższe biblioteki:

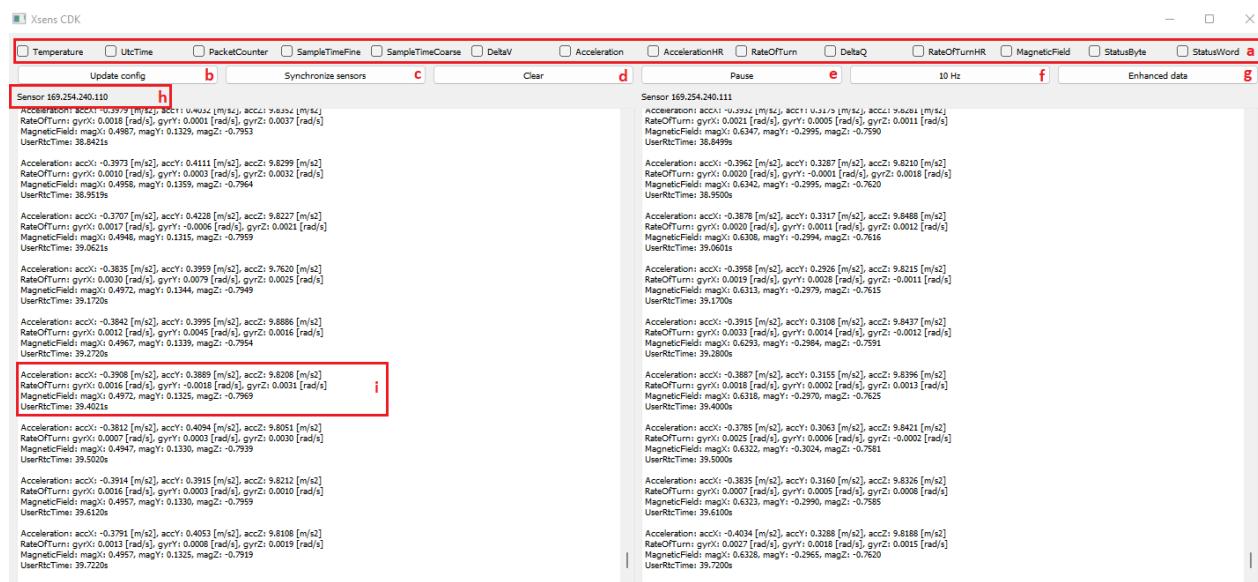
- Socket - interfejs sieciowy wykorzystany do komunikacji przy użyciu protokołu TCP,
- \_thread - moduł umożliwiający pracę z wykorzystaniem wielowątkowości,
- PyQt5 - nakładka na bibliotekę Qt napisaną w języku C++. Umożliwia zaprojektowanie interfejsu graficznego.

Przed rozpoczęciem pracy z aplikacją należy utworzyć sieć lokalną z wykorzystaniem przełącznika sieciowego do której podłączony będzie komputer klasy PC oraz jedna lub więcej zaprojektowanych płyt drukowanych. Powstałe oprogramowanie posiada poniższe funkcjonalności:

- wyświetlanie otrzymanych danych,
- synchronizacja układów RTC gdy podłączona jest większa ilość PCB,
- zmiana konfiguracji jednostki do nawigacji inercyjnej,
- automatyczny zapis wszystkich odebranych danych do pliku tekstowego.

### 5.2.1 Interfejs graficzny

Rysunek 5.4 prezentuje zrealizowany interfejs graficzny w momencie podłączenia do sieci dwóch PCB.



Rys 5.5. Interfejs graficzny aplikacji do komunikacji zrealizowaną płytą drukowaną

Opis poszczególnych elementów oznaczonych na rysunku:

- (a) pola wyboru konfiguracji IMU,
- (b) przycisk aktualizujący konfigurację IMU,
- (c) przycisk synchronizujący układy RTC będące peryferiami mikrokontrolerów znajdujących się na zrealizowanych płytach drukowanych,
- (d) przycisk czyszczący pola wizualizujące zebrane dane,
- (e) przycisk pauzujący pola wizualizujące zebrane dane,
- (f) przycisk aktualizujący częstotliwość wyświetlania danych (1/10/100 Hz),
- (g) przycisk zmieniający format wyświetlanych danych na CSV,
- (h) adres IP podpiętej płytki drukowanej zapisany w module sieciowym,
- (i) wizualizacja odebranych danych.



## 6 Testy systemu

Ostatnim etapem projektu są testy zrealizowanej platformy. Pozwolą one w pełni określić czy system działa poprawnie. Pierwszą oznaką właściwego funkcjonowania płytki drukowanej jest poprawna komunikacja ze zrealizowaną aplikacją do akwizycji danych. Zmiana orientacji położenia płytki wpływa w odpowiedni sposób na wartości odbieranych pomiarów z IMU zwizualizowanych w postaci tekstu na interfejsie graficznym. Pozostałe funkcjonalności oprogramowania, takie jak: synchronizacja zegarów urządzeń wielu PCB oraz zmiana konfiguracji jednostki do nawigacji inercyjnej również działają zgodnie z założeniami.

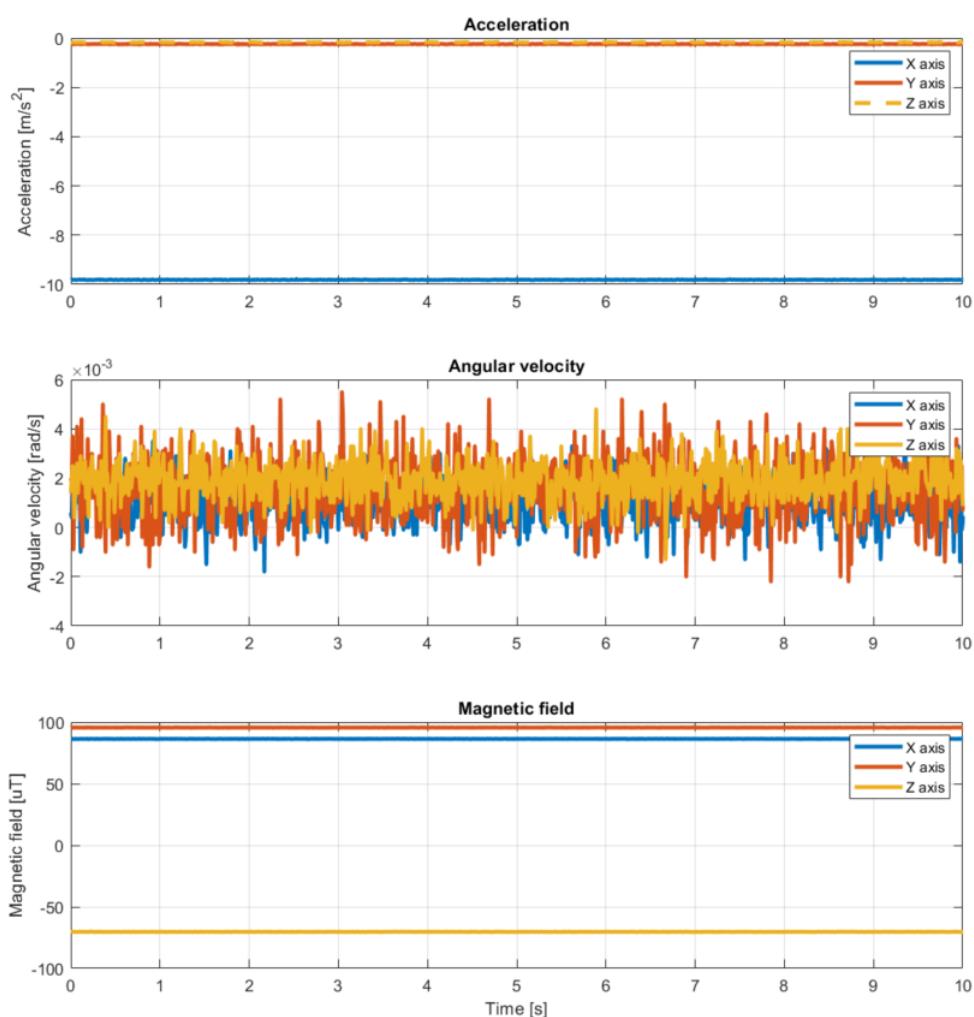
Kolejnym testem sprawdzającym funkcjonalność systemu jest przeprowadzenie akwizycji danych umożliwiających identyfikację orientacji przestrzennej kiści robota przemysłowego. Zostało to zrealizowane z wykorzystaniem robota FANUC znajdującego się w laboratorium robotyki w budynku C3 (rys. 6.1). Do przeprowadzenia eksperymentu wykorzystano jedną płytke drukowaną przymocowaną do efektora końcowego robota, którym był dedykowany uchwyt wykonany w technologii druku 3D. PCB oraz komputer odbierający dane podłączono do przełącznika sieciowego. Płytkę została zasilona przy pomocy zasilacza impulsowego 12 V.



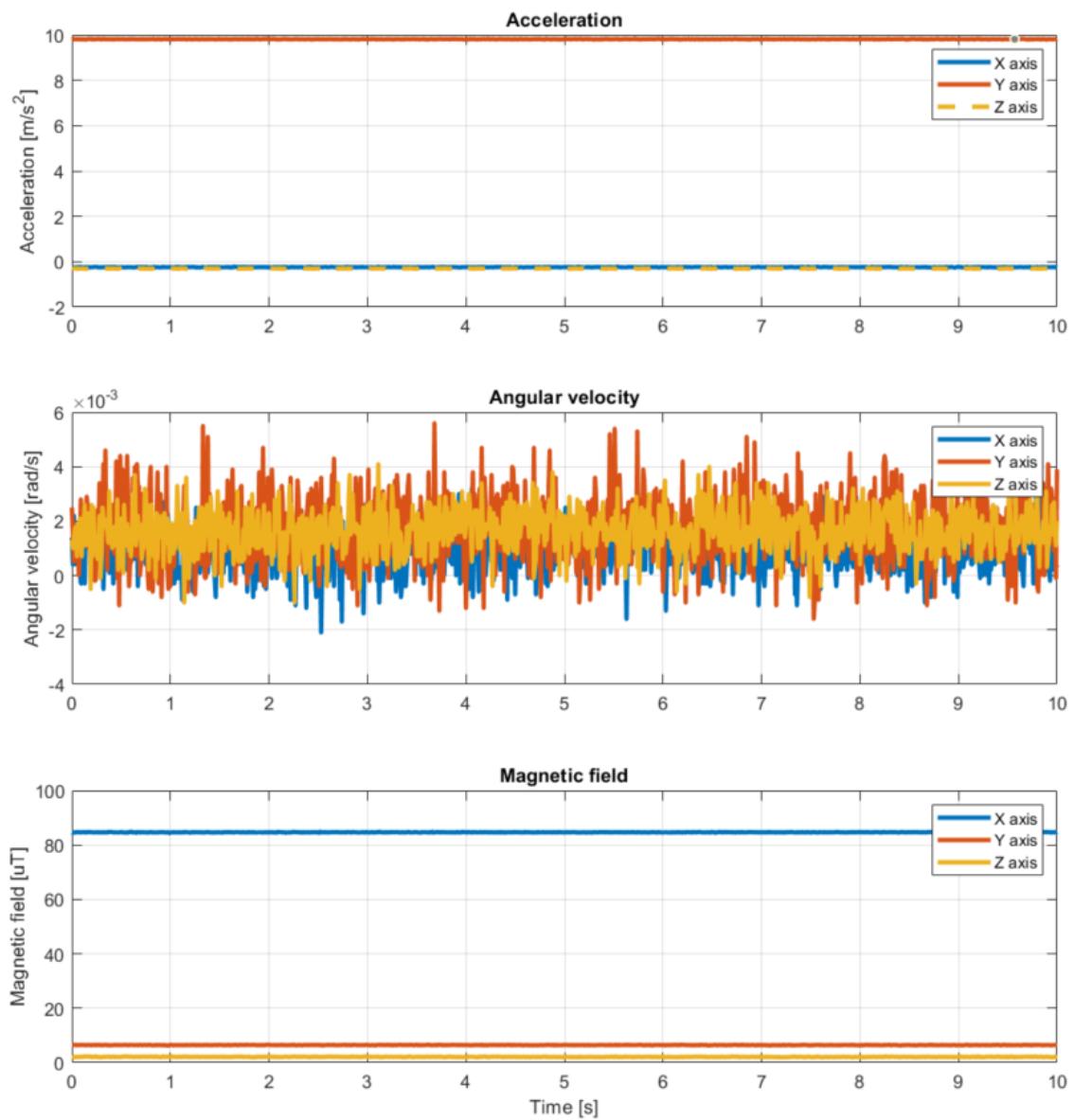
Rys 6.1. Zrealizowana platforma zamontowana do flanszy robota FANUC

## 6.1 Przebieg eksperymentu

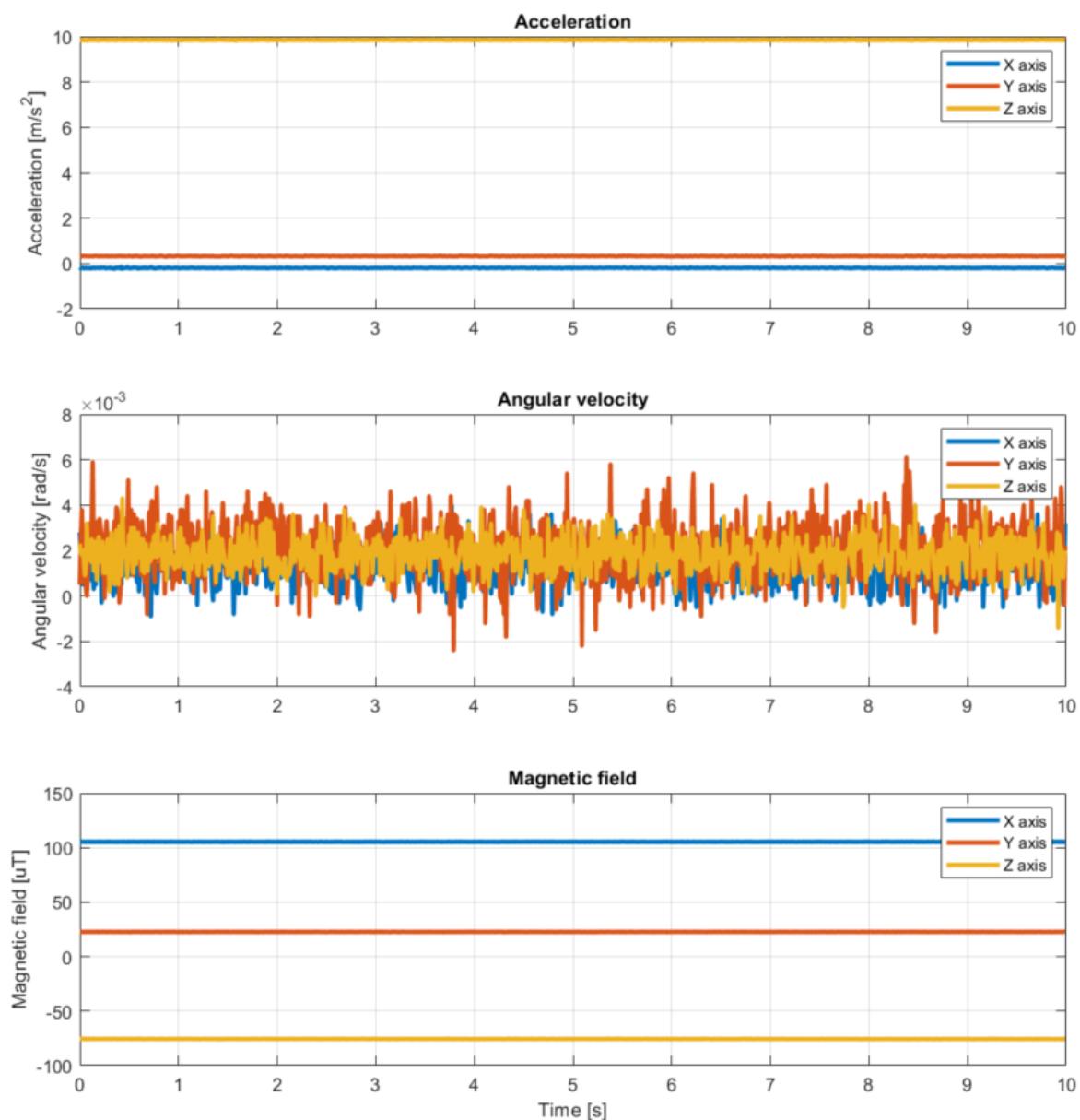
Po przygotowaniu systemu do pracy nastąpiło uruchomienie robota przemysłowego. Aby jednostka do nawigacji inercyjnej wykonywała jak najprecyzyjniejsze pomiary należy dokonać jej kalibracji. W dużym uproszczeniu polega ona na zorientowaniu platformy w każdej z trzech osi układu współrzędnych na pewien czas. Podczas testów systemu dokonano prezycyjnego pozycjonowania urządzenia za pomocą robota FANUC. Wykorzystano aplikację do akwizycji danych w celu zapisania wykonanych pomiarów do pliku tekstowego zgodnie z formatem CSV. Przy pomocy oprogramowania MATLAB dokonano wizualizacji wyników. Zarejestrowane dane zostały zaprezentowane na wykresach 6.2 - 6.4. Oś X przedstawia czas, a osie Y kolejno: przyśpieszenie ziemskie, prędkość kątową oraz pole magnetyczne. Wartości odebrane z akcelerometru na osi kalibracyjnej wynoszą 1 g lub -1 g. Wynik o tej samej wartości, ale przeciwnym znaku można otrzymać poprzez odwrócenie platformy o 180 stopni względem osi kalibracyjnej.



Rys 6.2. Kalibracja IMU względem osi X

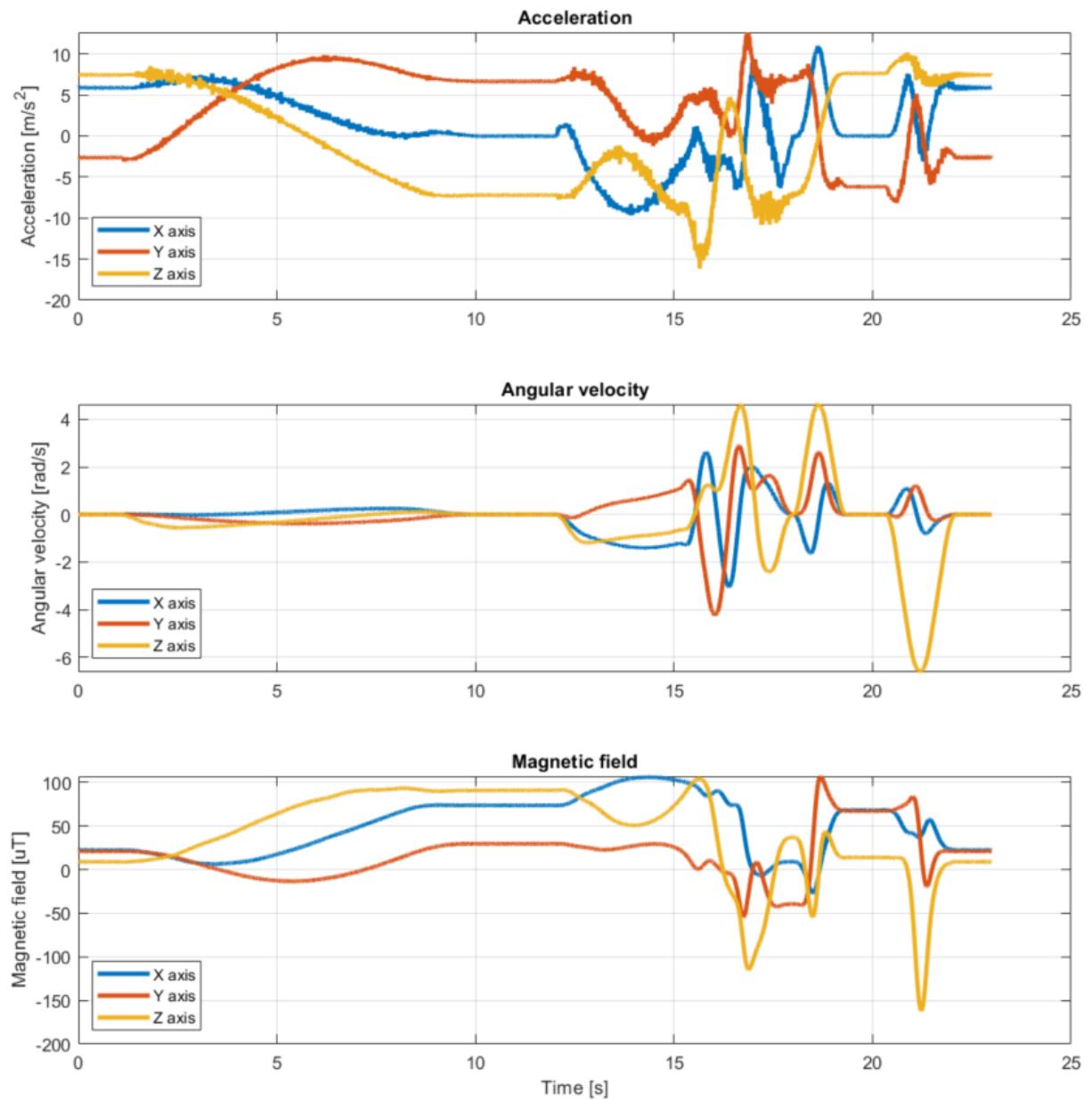


Rys 6.3. Kalibracja IMU względem osi Y



Rys 6.4. Kalibracja IMU względem osi Z

Po przeprowadzonej kalibracji robot przemysłowy FANUC został zaprogramowany, aby wykonywał zadany tor ruchu w zapętleniu. Rysunek 6.5 prezentuje dane zebrane podczas jednej trajektorii.



Rys 6.5. Wykresy wizualizujące dane zebrane podczas identyfikacji robota FANUC

Na samym początku oraz końcu wykresu wizualizującego pomiary z akcelerometru można zauważać pewne stałe wartości przyspieszenia ziemskiego w każdej z trzech osi. Wynikają one z faktu, że pomimo unieruchomienia układu, platforma nie była zorientowana równolegle do podłoża.

Tabela [5] zawiera część zebranych danych. Jako, że IMU działało z częstotliwością około 100 Hz zaprezentowany został co 50 pomiar z zakresu 1500-2000. Wyniki zaokrąglono do dwóch miejsc po przecinku.

**Tabela 5.** Wybrane dane otrzymane w przeprowadzonym eksperymencie

Znacznik czasu [s]	Akcelerometr [ $\text{m/s}^2$ ]			Żyroskop [rad/s]			Magnetometr [ $\mu\text{T}$ ]		
	oś X	oś Y	oś Z	oś X	oś Y	oś Z	oś X	oś Y	oś Z
14.989	-4.66	2.27	-7.69	-1.26	0.99	-0.72	103.03	26.28	72.54
15.489	0.24	4.55	-12.25	-0.40	1.04	-0.13	97.38	4.73	101.01
15.989	-4.56	4.57	-7.89	1.29	-4.16	1.07	88.11	8.50	64.00
16.489	-4.25	0.12	3.56	-2.47	1.69	3.73	73.92	-7.82	-29.54
16.989	7.96	7.34	-8.45	1.99	1.08	1.32	-0.50	1.08	-104.25
17.489	-2.99	5.93	-10.14	0.86	1.57	-2.31	3.02	-40.17	-14.01
17.989	-0.29	6.88	-7.12	0.00	0.00	0.00	9.08	-39.36	37.07
18.489	7.98	6.40	-2.95	-1.55	1.74	4.04	-26.71	27.56	-54.36
18.989	2.14	-5.90	6.36	1.06	0.45	2.29	53.65	77.72	28.74
19.488	0.02	-6.22	7.65	0.00	0.00	0.00	67.96	67.37	14.01
19.989	0.01	-6.19	7.63	0.00	0.00	0.00	68.25	67.29	13.96

## 7 Podsumowanie

Celem niniejszej pracy było zaprojektowanie, zaprogramowanie oraz praktyczne przetestowanie urządzenia do poboru danych umożliwiających identyfikację orientacji przestrzennej. Ważnym aspektem projektu była realizacja transmisji danych w reżimie czasu rzeczywistego. Może się to okazać niezwykle przydatne podczas zastosowania zrealizowanej płytki w systemie wymagającym jak najszybszej informacji o aktualnym położeniu (np. roboty kroczące, drony).

Praca dyplomowa zawiera szczegółowy opis kolejno wykonywanych czynności podczas powstawania urządzenia. Rozpoczęto od wybrania odpowiednich komponentów elektronicznych wraz z uzasadnieniem. Następnie dokonano opisu projektowania schematu obwodu elektronicznego, układu płytki PCB oraz procesu montażu komponentów elektronicznych. Przedstawiono architekturę oraz działanie systemu w oparciu o powstałe dedykowane oprogramowanie na układ mikroprocesorowy, który jest częścią zrealizowanego PCB. Pracę zakończono opisem testów powstałej platformy.

Podczas działań związanych z realizacją projektu zauważono kilka drobnych niedoskonałości, które nie miały większego wpływu na ostateczną formę urządzenia. Przykładową rzeczą, którą można było zrobić lepiej jest zastosowanie większej ilości punktów testowych na płytce drukowanej. W szczególności na liniach sygnałowych. Ułatwiłoby to proces sprawdzania poprawności komunikacji mikrokontrolera z pozostałymi układami znajdującymi się na płytce.

Praca posiada wiele możliwości rozwoju, w szczególności od strony programistycznej. Jedną z nich jest obsługa przez urządzenie większej ilości komend rozszerzających jego konfigurowalność. Można również zastanowić się nad realizacją wewnątrz procesora algorytmu filtrującego zebrane dane. Płytkę drukowaną posiada możliwość miniaturyzacji poprzez zastosowanie elementów pasywnych o mniejszym rozmiarze.



## Bibliografia

- [1] Barbara M. Kreutz, *Mediterranean Contributions to the Medieval Mariner's Compass Technology and Culture*, 1973
- [2] [https://en.wikipedia.org/wiki/Inertial\\_navigation\\_system](https://en.wikipedia.org/wiki/Inertial_navigation_system),  
dostęp: 15.10.2022
- [3] Manon Kok, Jeroen D. Hol and Thomas B. Schön, *Using Inertial Sensors for Position and Orientation Estimation*, 2017
- [4] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, and Nazirah M. Khairi, *Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications*, 2013
- [5] <https://www.vectornav.com/resources/inertial-navigation-articles/what-is-an-inertial-measurement-unit-imu>, dostęp: 29.11.2022
- [6] Anton Siswo, Roharjo Ansori, Journal of Engineering and Applied Sciences, *A Review of Accelerometer Sensor and Gyroscope Sensor in IMU Sensors on Motion Capture*, 2020
- [7] Vittorio M. N. Passaro, Antonello Cuccovillo, Lorenzo Vaiani, Martino De Carlo and Carlo Edoardo Campanella, *Gyroscope Technology and Applications: A Review in the Industrial Perspective*, 2017
- [8] John Geen, David Krakauer, *New iMEMS® AngularRate-Sensing Gyroscope*, 2003
- [9] Nonnarit O-larnnithipong, Armando Barreto, *Gyroscope Drift Correction Algorithm for Inertial Measurement Unit used in Hand Motion Tracking*, 2016
- [10] <https://x-io.co.uk/ngimu/>, dostęp: 01.12.2022
- [11] Zhang Yushuai, Guo Jianxin, Ji Xiang, Zhu Rui *Indoor Navigation System Based on Foot-Mounted IMU and Map Information Fusion*, 2021
- [12] <https://www.microstrain.com/inertial-sensors/3dm-cx5-25>,  
dostęp: 01.12.2022
- [13] <https://open-dynamic-robot-initiative.github.io/> dostęp: 01.12.2022
- [14] [https://github.com/open-dynamic-robot-initiative/open\\_robot\\_actuator\\_hardware/blob/master/mechanics/biped\\_6dof\\_v1/images/biped\\_3.jpg](https://github.com/open-dynamic-robot-initiative/open_robot_actuator_hardware/blob/master/mechanics/biped_6dof_v1/images/biped_3.jpg)  
dostęp: 01.12.2022
- [15] Xsens, *MTi 1-series Datasheet*, Revision 2019.A, Dec 2019

- [16] <https://www.wiznet.io/product-item/wiz820io/>, dostęp: 26.10.2022
- [17] STMicroelectronics, *STM32L433xx datasheet*, May 2021
- [18] WIZnet, *iEthernet W5200 Datasheet*, Version 1.4.0
- [19] [https://media.digikey.com/Renders/STMicro\%20Renders/497;-1L\\_ME;-T;-100.jpg](https://media.digikey.com/Renders/STMicro\%20Renders/497;-1L_ME;-T;-100.jpg), dostęp: 29.10.2022
- [20] Texas Instruments, *TPS54228 4.5-V to 18-V Input, 2-A Synchronous Step-Down Converter With Eco-Mode*, Revised august 2016
- [21] Be Van Ngo, Law P. and Sparks A. *Use of JTAG Boundary-Scan for Testing Electronic Circuit Boards and Systems* , 2008
- [22] <https://mtidocs.xsens.com/integration-manual>, dostęp: 11.11.2022
- [23] <https://www.linkedin.com/pulse/c-build-process-details-abdelaziz-moustafa/>,  
dostęp: 20.11.2022

## **Załączniki**

1. Załącznik 1 - Schemat elektroniczny urządzenia
2. Załącznik 2 - Zestawienie materialowe