Aleksander Strzeboński

Tomasz Goryczka

Piotr Radecki

Sprawozdanie z wykonania projektu z przedmiotu Sensory w Aplikacjach Wbudowanych

Temat projektu: Wykonanie płytki PCB integrującej akcelerometr ICM-20948 w standardzie Mikro-Bus oraz zastosowanie akcelerometru jako myszka desktopowa

Opiekun: Jan Macheta

1. Opis projektu

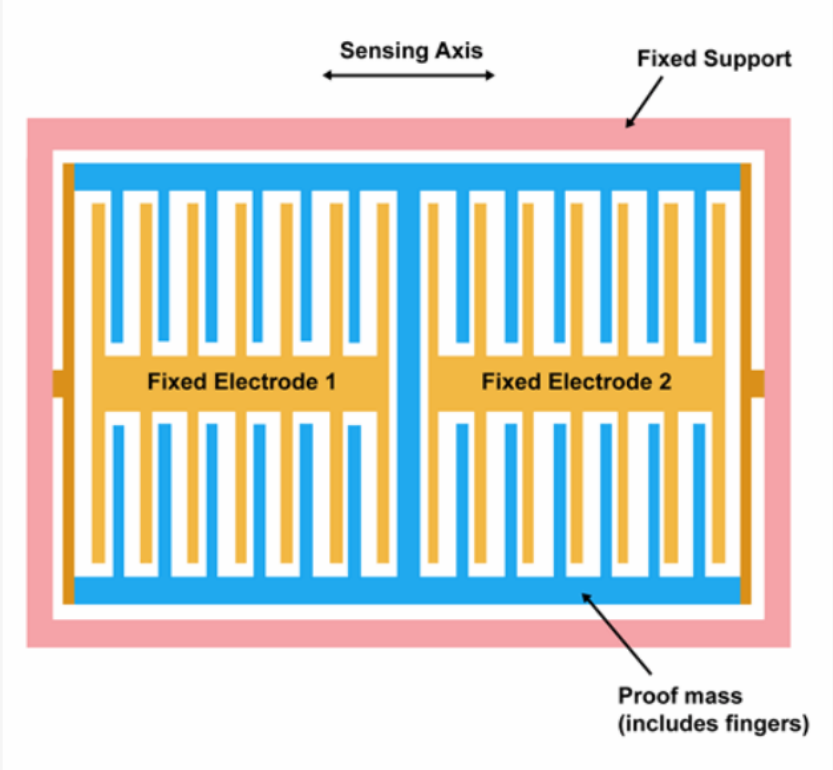
Celem projektu było wykonanie płytki PCB z akcelerometrem ICM-20948 w standardzie Mikro-Bus, oprogramowanie akcelerometru tak żeby możliwe było zebranie danych pomiarowych z akcelerometru i przesłanie ich przez interfejs UART do komputera PC oraz napisanie aplikacji na komputer, która odbierze dane po UART i użyje ich do sterowania myszką komputerową. Poniżej zamieszczono schemat, który symbolicznie przedstawia jak przekazywane są dane.

Sensor mierzy przyspieszenie i prędkość kątową i przesyła dane do mikrokontrolera z rodziny Nucleo, który opakowuje dane w pakiet i przesyła przez UART do komputera.

2. Zasada działania czujnika

Akcelerometr

Czujnik przyspieszenia składa się z masy zawieszonej na elementach sprężystych. W wyniku działania przyspieszenia na masę działa siła F = ma, która powoduje odkształcenie sprężyny. Odkształcenie to mierzy się wykrywając zmianę pojemności pomiędzy ruchomą masą a obudową, mierząc częstotliwość drgań obwodu rezonansowego wykorzystującego tą zmienną pojemność. Struktura akcelerometru MEMS może wyglądać następująco:

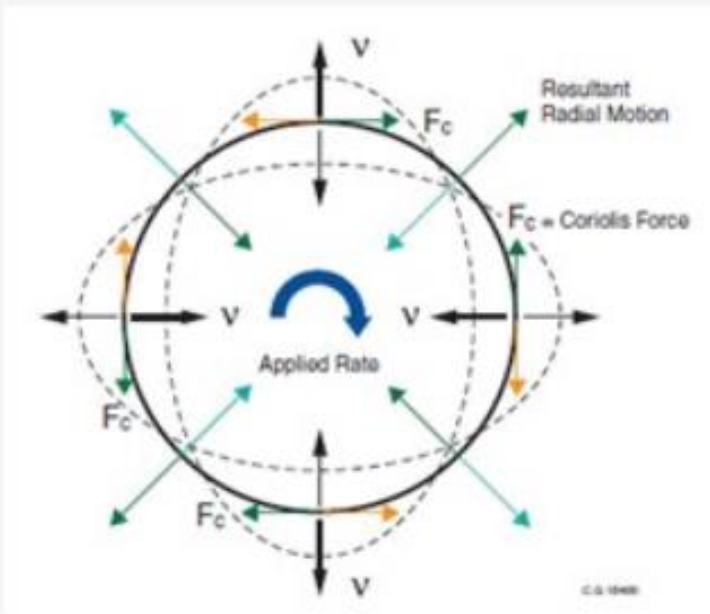


Źródło: https://www.siliconsensing.com/technology/mems-accelerometers/

W czujniku umieszczone są trzy takie struktury, które mierzą przyspieszenie w trzech wzajemnie prostopadłych osiach.

Żyroskop

Żyroskop, czyli czujnik prędkości obrotowej w technologii MEMS wykorzystuje efekt Coriolisa. Polega on na tym, iż kiedy obiekt porusza się po ruchu okrężnym ze stałą prędkością kątową, to w przypadku zmiany promienia występuje przyspieszenie związane ze zmianą prędkości liniowej. Zmiana promienia następuje w wyniku ruchu drgającego, a przyspieszenie Coriolisa wykrywa się na kierunku prostopadłym do kierunku drgań. Struktura układu może składać się z zawieszonego drgającego pierścienia, jak na rysunku poniżej:

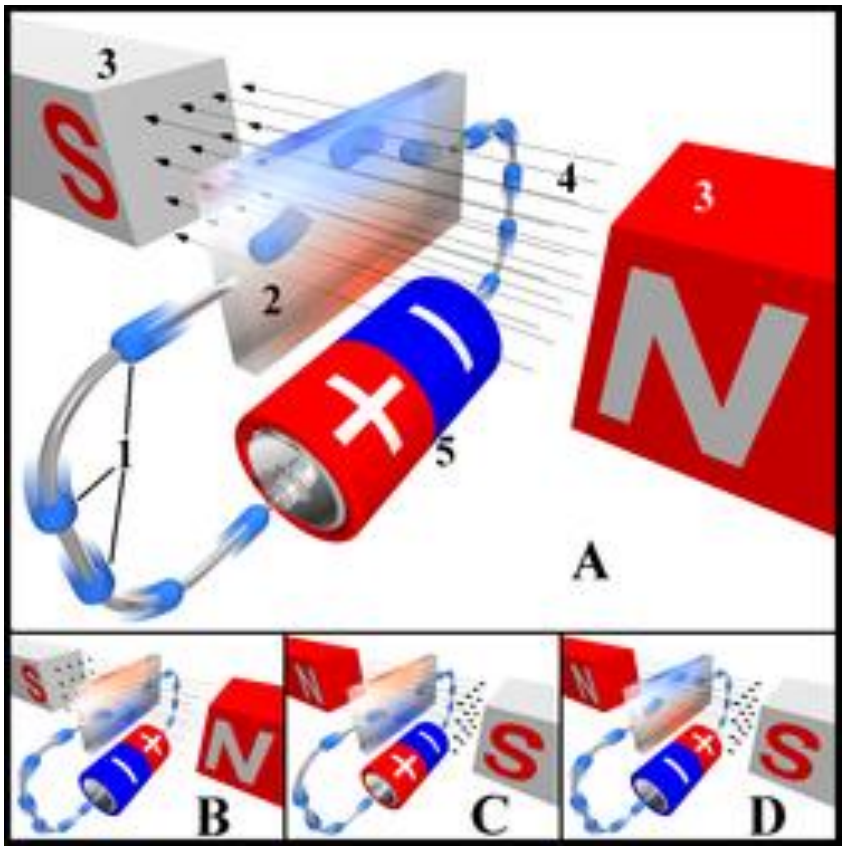


Źródło: https://www.siliconsensing.com/technology/mems-gyroscopes/

Czujnik mierzy prędkość obrotową w trzech wzajemnie prostopadłych płaszczyznach.

Magnetometr

Trzecią, lecz nieużywanym przez naszą aplikację, wielkością, którą mierzy czujnik jest natężenie pola magnetycznego. W celu jego pomiaru wykorzystuje się efekt Halla. Efekt ten jest wynikiem działania siły Lorentza. Gdy umieści się przewodnik przez który płynie prąd w polu magnetycznym, między ścianami przewodnika prostopadłymi do wektora natężenia pola magnetycznego pojawi się napięcie zwane napięciem Halla. Dzieje się tak, gdyż na płynące w przewodniku elektrony działa siła Lorentza, która popycha elektrony w kierunku jednej ze ścian. Zmierzone napięcie Halla można przeliczyć na wartość wektora natężenia pola magnetycznego.

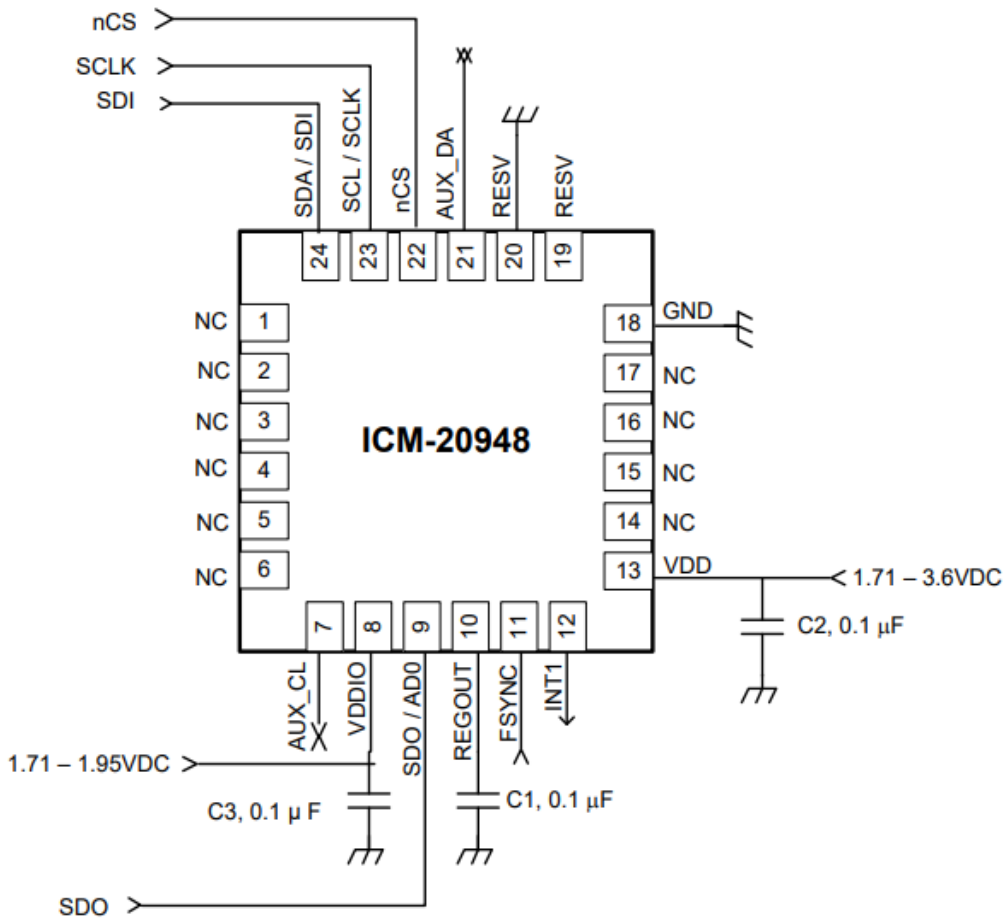


Źródło: https://pl.wikipedia.org/wiki/Zjawisko_Halla

Czujnik mierzy pole magnetyczne w trzech wzajemnie prostopadłych osiach.

3. Opis czujnika

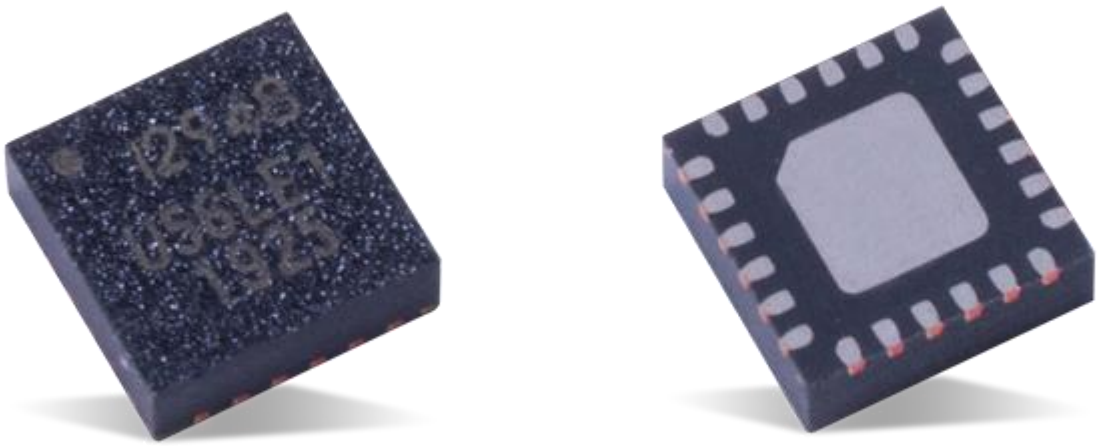
W projekcie użyto czujnika ICM-20948. Jest to czujnik, który mierzy przyspieszenie, prędkość obrotową oraz pole magnetyczne – każde w trzech osiach. Stąd nazywany jest 9-osiowym czujnikiem do śledzenia ruchu. Poniżej umieszczono pinout czujnika z noty katalogowej oraz zdjęcie czujnika:



PIN NUMBER	PIN NAME	PIN DESCRIPTION
7	AUX_CL	I²C Master serial clock, for connecting to external sensors
8	VDDIO	Digital I/O supply voltage
9	AD0 / SDO	I²C Slave Address LSB (AD0); SPI serial data output (SDO)
10	REGOUT	Regulator filter capacitor connection
11	FSYNC	Frame synchronization digital input. Connect to GND if unused
12	INT1	Interrupt 1
13	VDD	Power supply voltage
18	GND	Power supply ground
19	RESV	Reserved. Do not connect.
20	RESV	Reserved. Connect to GND.
21	AUX_DA	I²C master serial data, for connecting to external sensors
22	nCS	Chip select (SPI mode only)
23	SCL / SCLK	I²C serial clock (SCL); SPI serial clock (SCLK)
24	SDA / SDI	I²C serial data (SDA); SPI serial data input (SDI)
1 – 6, 14 - 17	NC	Do not connect

Table 10. Signal Descriptions

Źródło: DS-000189-ICM-20948-v1.3.pdf (tdk.com)

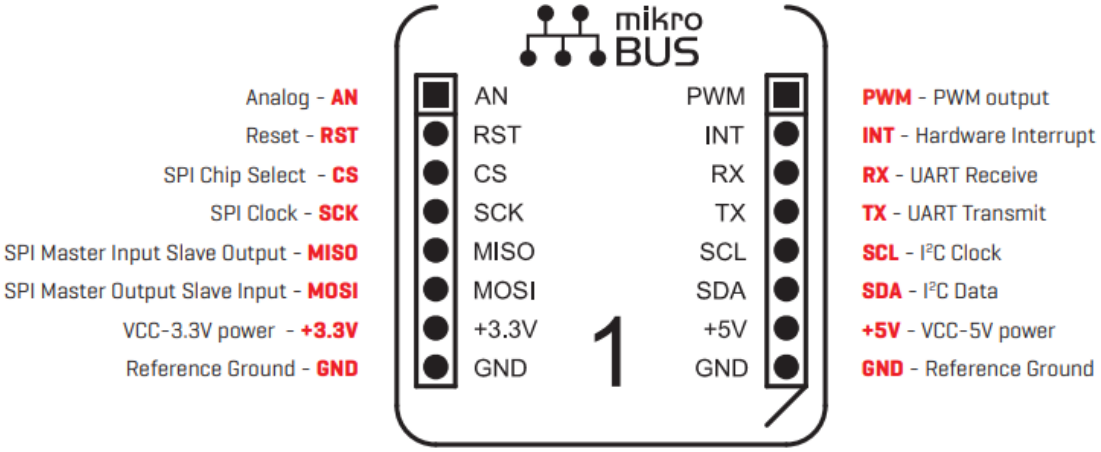


Źródło: https://www.mouser.pl/new/tdk/invensense-icm-20948/

Czujnik pozwala na komunikację po interfejsie SPI lub I2C. Komunikacja z czujnikiem oparta jest na pisaniu do oraz czytaniu z rejestrów czujnika. W fazie wymiany danych czujnik funkcjonuje jako slave. W pierwszym etapie wpisywany jest adres rejestru oraz informacja czy master chce przeczytać dany rejestr czy do niego pisać następnie w zależności od wybranego trybu dane spod wybranego rejestru są przesyłane do mastera lub master przesyła dane, które następnie są zapisywane do wybranego rejestru. Dodatkowo, gdy korzysta się z interfejsu I2C możliwe jest ciągłe pisanie lub czytanie z kolejnych rejestrów. Po każdej operacji odczytu/zapisu adresu, adres jest inkrementowany o 1 i następuje kolejna operacja odczytu/zapisu do nowego adresu aż do momentu kiedy master wyśle warunek przerwania operacji.

4. Projekt płytki PCB:

Pierwszą częścią projektu było zrobienie płytki PCB. Schemat oraz layout płytki robione były przy użyciu programu Altium Designer. Płytka zrobiona została w standardzie Mikro-Bus. Mikro-Bus to standard określający kształt, rozmiar oraz pinout płytek PCB, które obsługują dowolne sensory komunikujące się po interfejsach SPI lub I2C. Dzięki temu możliwe jest swobodne podmienianie sensorów przyłączonych do mikrokontrolerów. Poniżej zamieszczono pinout określony przez standard Mikro-Bus:



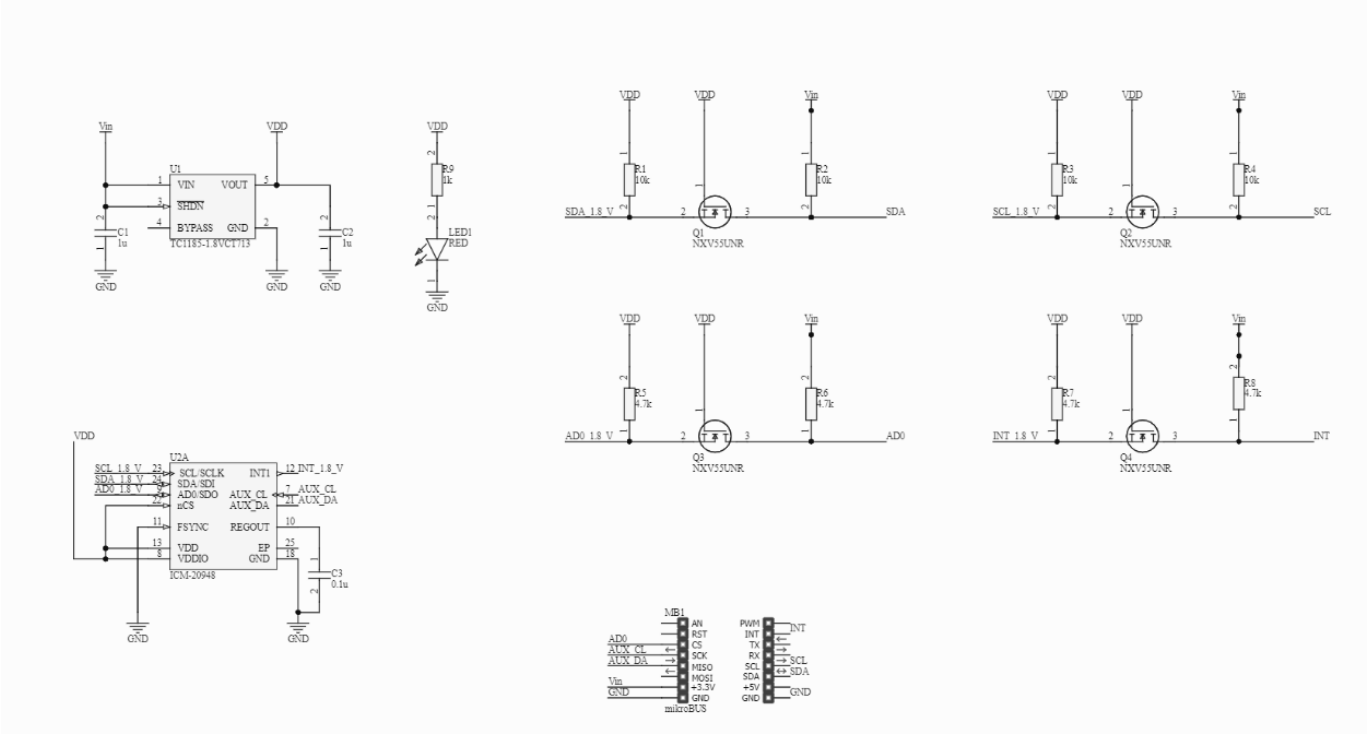
Źródło: [mikrobus-standard-specification-v200.pdf (mikroe.com)]

Schemat płytki PCB

Pierwszym etapem projektowania płytki było stworzenie schematu elektronicznego. Na schemacie możemy wyróżnić takie elementy jak:

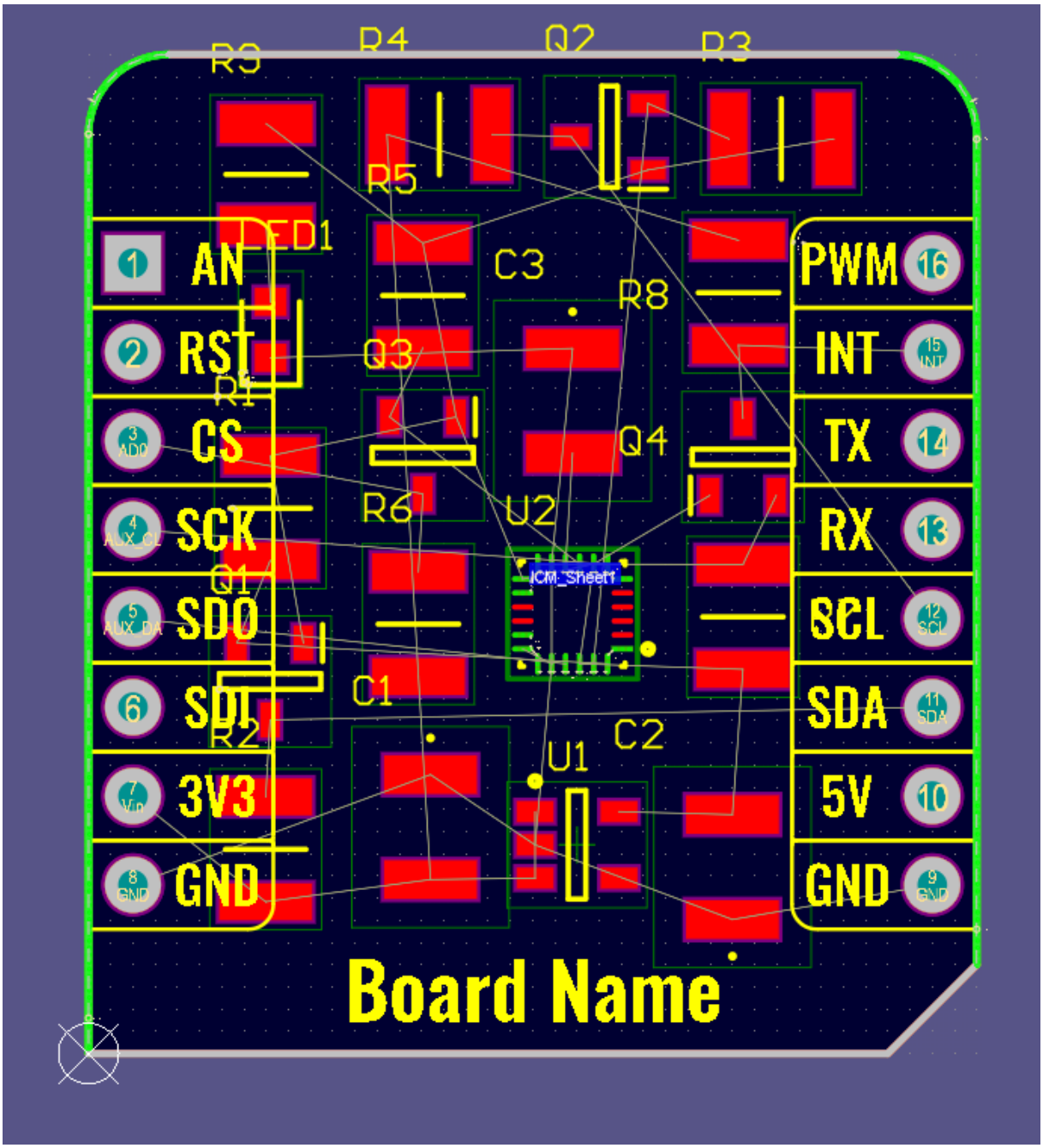
- Stabilizator konwertujący napięcie 3.3V pochodzące od mikrokontrolera Nucleo na napięcie 1.8V, którym zasilany jest czujnik ICM-20948,
- Cztery układy stworzone z dwóch rezystorów podciągających i tranzystorów MOS, które odpowiedzialne są za konwersję stanów logicznych z napięcia 2.2V na 1.8V,
- Diodę LED sygnalizującą, że płytka jest zasilana.

Poniżej umieszczono schemat płytki

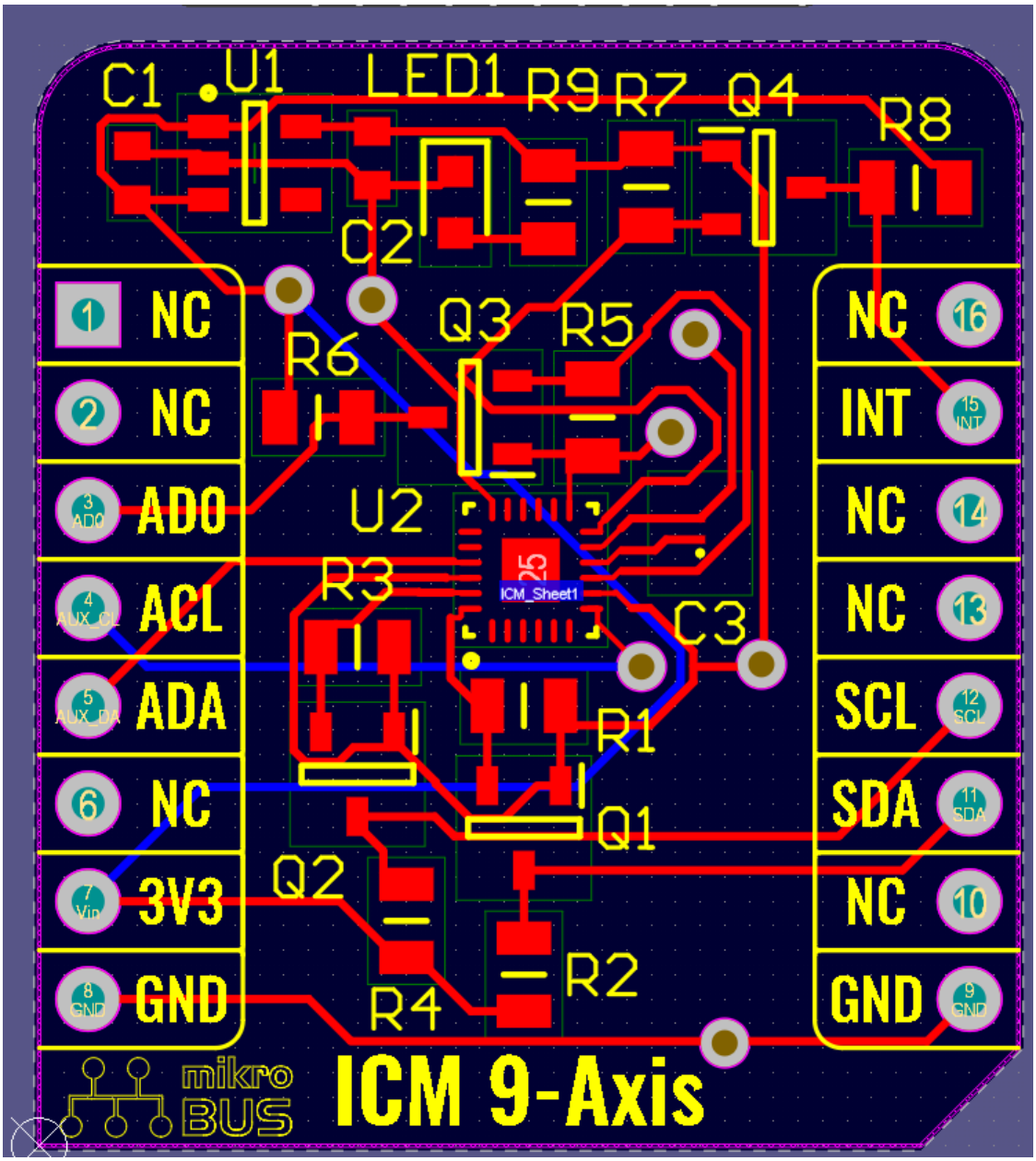


Layout płytki

Kolejnym etapem było ułożenie layout płytki zgodnie ze standardem Mikro-Bus. Tutaj pojawiły się z początku pewne problemy z ułożeniem elementów. Jako, że nigdy wcześniej nie lutowaliśmy, to zdecydowaliśmy się na elementy o dużym rozmiarze, które łatwiej by się lutowało. Okazało się jednak, że przesadziliśmy z rozmiarem elementów i nie dało się ich ułożyć. Dodatkowo layout czujnika ICM-20948 był tak zrobiony, że narożne pady były bardzo blisko siebie i nie spełniały reguł DRC. Poniżej zamieszczono ułożone zbyt duże elementy i niespełniający reguł model.



W celu uporania się z problemami wybraliśmy mniejsze elementy oraz inny model layoutu sensora. Dodatkową zaletą nowego modelu był pad po środku sensora służący do odprowadzania ciepła. Czego nie zauważyliśmy, a co dało się później we znaki przy lutowaniu, to fakt że wybrano niezwykle mały model kondensatora C3 (0.2mm x 0.3mm). W nazwie było popularne 0203 ale okazało się, że odnosiło się to do rozmiaru w milimetrach co z resztą widać na layoucie. Poniżej przedstawiono już skończony layout:

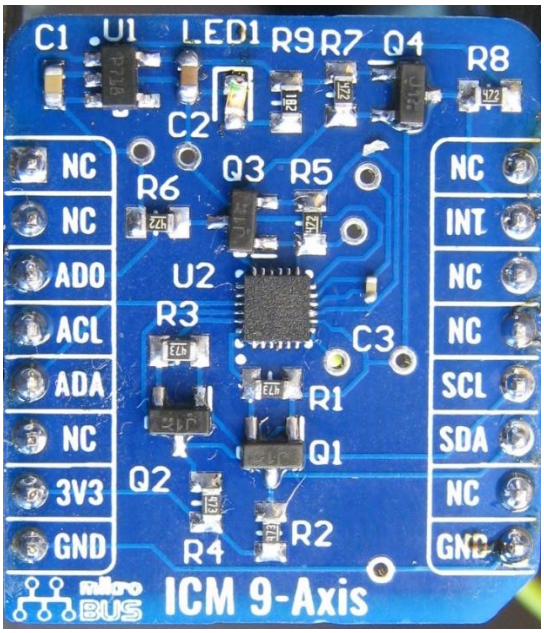


Płytka składa się z czterech warstw – dwóch warstw sygnałowych, jednej warstwy z zasilaniem i jednej warstwy z masą. Wyprowadzono następujące piny:

- AD0 - pin wyznaczający ostatni bit adresu I2C akcelerometru. Dzięki niemu można podłączyć dwa akcelerometry do jednej szyny.
- ACL – zegar modułu I2C używany do podłączania urządzeń po magistrali I2C w łańcuch
- ADA – szyna danych modułu I2C przy podłączaniu urządzeń w łańcuch
- 3V3 – zasilanie 3.3V
- GND – masa
- INT – port przerwań generowanych przez akcelerometr
- SCL – zegar I2C
- SDA – szyna danych I2C.

Lutowanie

Jako, że nigdy wcześniej nie lutowaliśmy, umówiliśmy się na lutowanie z opiekunem projektu Panem mgr inż. Janem Machetą, który pomógł nam przy lutowaniu. Przy lutowaniu też wyszedł problem z za małym kondensatorem C3, który przy użyciu odpowiednio dużej ilości cyny zastąpiony został większym kondensatorem niż miało to miejsce na layoucie. Z powodu braku zasobów oraz tego, że ta zmiana nie była zbytnio istotna zmieniono wartości rezystorów podciągających i rezystora ograniczającego prąd diody na takie, które akurat były dostępne. Poniżej umieszczono zdjęcie płytki po lutowaniu:



5. Aplikacja na mikrokontroler

Aplikacja działająca na mikrokontrolerze odbiera dane z akcelerometru (przyspieszenie i rotację) po I2C, opakowuje w nagłówek, oblicza sumę kontrolną i przesyła skonstruowany pakiet do komputera po UART-cie.

Użyty został kod napisany na laboratoriach z tego przedmiotu. Kod wymagał pewnych modyfikacji – należało przerobić projekt CubeMX, gdyż użyta została inna płytka – Nucleo-F767ZI. Dodatkowo należało przerobić komunikację z akcelerometrem z protokołu SPI na komunikację po protokole I2C.

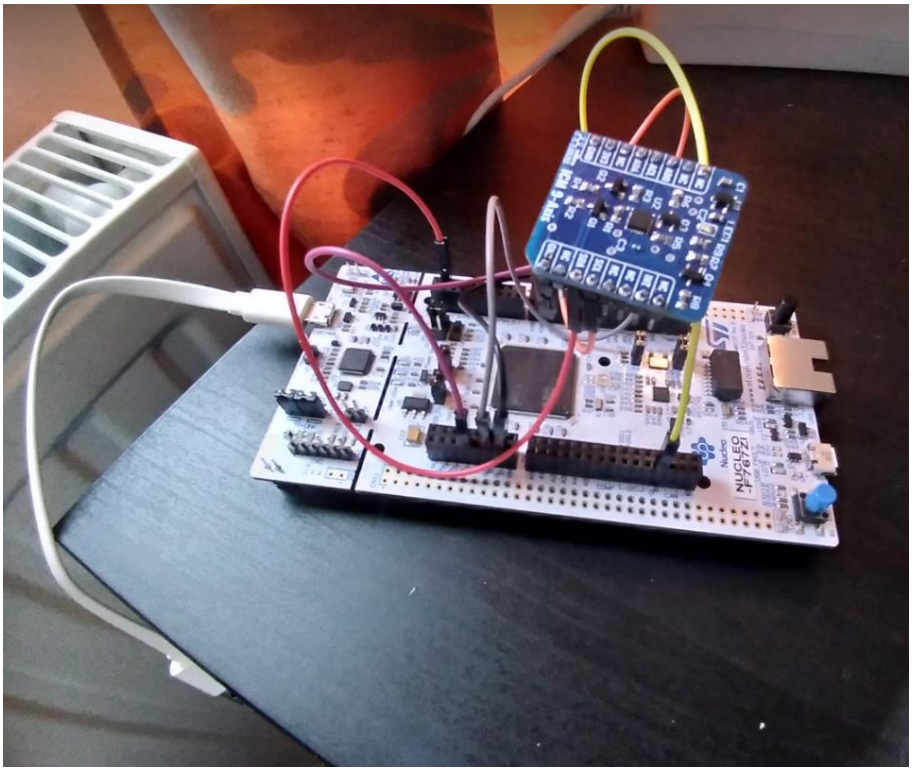
Poniżej prezentuję format zastosowanego pakietu, w którym umieszczane były dane z czujnika.

Byte:	1-2	3	4	5-6	(variable-size)
Field:	Preamble (0xFF, 0xFF)	SOP (0xAB)	Type (1 Byte)	checksum	payload

Źródło: [lab4-connecting_the_dots · main · Jan Macheta / SwAW Lab · GitLab]

Pierwszym polem pakietu jest charakterystyczna preambuła. Następnie umieszczone jest pole SOP (Start Of Packet). Kolejno przesyłany jest typ danych. Wartość 0x01 oznacza dane z żyroskopu, a wartość 0x02 oznacza dane z akcelerometru. Kolejnym polem jest 2-bajtowa suma kontrolna obliczana z pola typu oraz pola danych (payload). Ostatnim polem jest pole danych. Pole to składało się z 12 bajtów – po 4 bajty na każdą daną (pomiar przyspieszenia w jednym z trzech kierunków lub pomiar prędkości kątowej w jednej z trzech płaszczyzn). Z tego z czterech bajtów tylko dwa były wykorzystane na rzeczywiste dane, a reszta na padding.

W trakcie wyliczania sumy kontrolnej okazało się, że kod udostępniony na repozytorium zawiera błąd i suma jest niewłaściwie obliczana (program czyta payload z niezapisanego obszaru bufora). Błąd ten został naprawiony. Dane z żyroskopu i akcelerometru wysyłane są co 20ms każda, z czego ograniczenie prędkości związane było z wymaganiami postawionymi przez aplikację komputerową i prędkością z jaką była ona w stanie przetwarzać dane. Poniżej umieszczam zdjęcie płytki podłączonej do mikrokontrolera:



Zweryfikowano, że dane przesyłane z czujnika do komputera są poprawne.

Wersja testowa – sprzed posiadania płytki PCB

W celu zweryfikowania aplikacji na mikrokontroler oraz ułatwienia rozwoju aplikacji na komputer w pierwszej fazie – zanim zrealizowano zamówienie na płytkę PCB, do testów została użyta bliźniacza płytka od firmy MIKROE – 9DOF 2 Click, która również integruje akcelerometr ICM-20948, ale komunikuje się z mikrokontrolerem przez interfejs SPI. Poniżej zamieszczam zdjęcie płytki. Była ona również używana na laboratoriach, ale podpinaliśmy ją do innego mikrokontrolera.

Źródło: [9DOF 2 Click (mikroe.com)]

6. Użycie akcelerometru jako mysz desktopowa

Ostatnim elementem projektu jest skrypt Pythonowy, który odbiera pakiety przesłane od płytki przez UART z wykorzystaniem biblioteki PySerial, a następnie wykorzystując filtr Kalmana przetwarza otrzymane dane i na ich podstawie steruje myszką.

Program estymuje orientację czujnika względem płaszczyzny poziomej względem ziemi obliczając kierunek i długość wektora grawitacji na podstawie danych z akcelerometru i scałkowanej prędkości obrotowej na podstawie danych z żyroskopu. Znając 2 kąty przechylenia względem ziemi odejmuje wektor grawitacji od wejściowych danych przyspieszenia. Otrzymana różnica będąca rzeczywistym przyspieszeniem obiektu jest zamieniana na prędkość poprzez całkowanie. Wykorzystano łącznie 4 kanały z filtrem Kalmana: 2 z nich służą do usuwania szumu akcelerometru i dryftu żyroskopu przy obliczaniu kątów przechylenia, kolejne 2 służą do usuwania szumu z danych przyspieszenia ruchu w osiach x i y czujnika. Pozostały błąd dryftu położenia rozwiązano zerując prędkość w przypadku niskiej wariancji przyspieszeń oznaczającej, iż myszka w rzeczywistości najprawdopodobniej nie porusza się. Obliczona prędkość w dwóch osiach w płaszczyźnie poziomej względem czujnika jest wykorzystywana do sterowania myszką za pomocą biblioteki Mouse. Funkcja sterująca kursorem przyjmuje względne przesunięcie w zadanym czasie (prędkość).

Parametry przyjętych filtrów Kalmana:

Obliczanie kąta obrotu:

Macierz stanu $A = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix}$, dt = 0,02

Macierz wejścia $B = \begin{bmatrix} dt \\ 0 \end{bmatrix}$

Macierz wyjścia $C = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

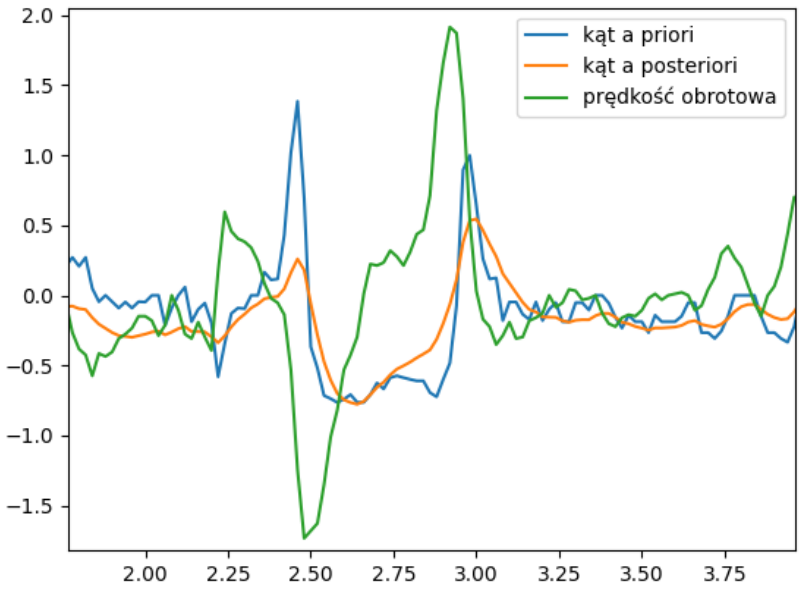
Szum przetwarzania: 0,6dt

Szum pomiarowy: 0,6dt

Prędkość kątową pomnożono razy 6 w celu przyspieszenia reakcji na obrót przy powolnej reakcji na zmiany danych wejściowych kąta.

Zmienna przechowuje wartość kąta i dryftu żyroskopu.

Poniżej zaprezentowano działanie filtru Kalmana na danych wejściowych kąta pochylenia prędkości obrotowej:



Obliczanie prędkości liniowej:

Macierz stanu $A = [1]$

Macierz wejścia $B = [dt]$

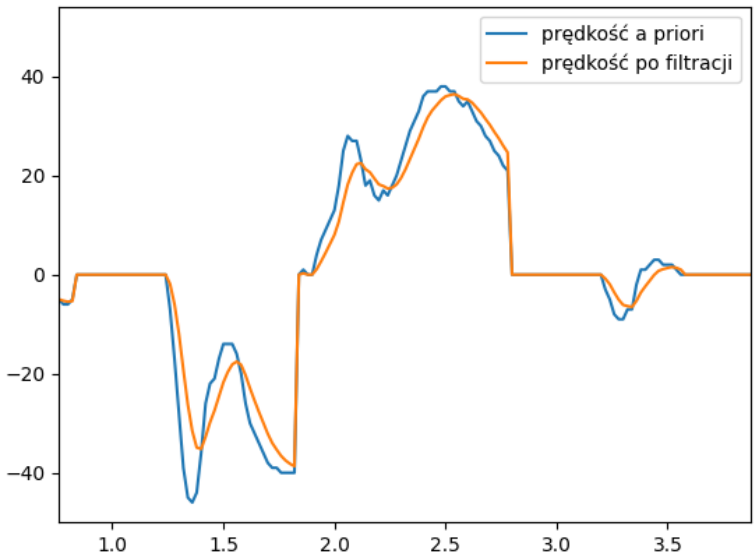
Macierz wyjścia $C = [1]$

Szum przetwarzania: 0,01dt

Szum pomiarowy: 0,02*dt*

Zmienna przechowuje wartość prędkości liniowej.

Poniżej zaprezentowano działanie filtru Kalmana na danych wejściowych przyspieszenia (przedstawione po scałkowaniu) wraz z zerowaniem prędkości za pomocą wariancji przyspieszenia:



Podsumowanie:

W efekcie program jest w stanie sterować myszą, jednak sterowanie to nie jest dokładne. Problematyczne okazało się dostrojenie filtru Kalmana tak, aby program poprawnie odejmował przyspieszenie pochodzące od grawitacji od danych akcelerometru. W efekcie nawet chwilowy błąd powoduje nadanie kursorowi prędkości niewynikającej z ruchu czujnika.

7. Kod źródłowy

Kod źródłowy skryptu Pythonowego oraz projektu na mikrokontroler dostępne są w poniższym repozytorium: AStrzebonski/SAW-ICM20948 (github.com).

8. Podział obowiązków

Zastosowano następujący podział obowiązków:

- **Piotr Radecki:**
  - Schemat płytki
  - Aplikacja na mikrokontroler
- **Aleksander Strzeboński:**
  - Layout płytki
  - Aplikacja na mikrokontroler
- **Tomasz Goryczka:**
  - Aplikacja na PC