

PROJEKT

STEROWNIKI ROBOTÓW

Raport

Analizator chodu

AnCh

Skład grupy (4):

Sebastian JAMROZIŃSKI, 218546

Hubert GRZEGORCZYK, 218596

Termin: srTP11

Prowadzący:

mgr inż. Wojciech DOMSKI

12 czerwca 2017

Spis treści

1	Opis projektu	2
2	Konfiguracja mikrokontrolera	2
3	Konfiguracja peryferiów	4
3.1	SPI2	4
3.2	LCD	4
3.3	QUADSPI	4
3.4	USART	4
3.5	TIMER6	4
3.6	Podsumowanie peryferiów	4
3.7	GPIO	4
4	Opis wykorzystywanych układów	4
4.1	Akcelerometr	5
4.2	Żyroskop	5
5	Opis działania programu	5
5.1	Algorytm rozpoznawania chodu	5
5.2	Opis algorytmu	5
5.3	Właściwy algorytm	7
6	Możliwości aplikacji	8
7	Podsumowanie	8

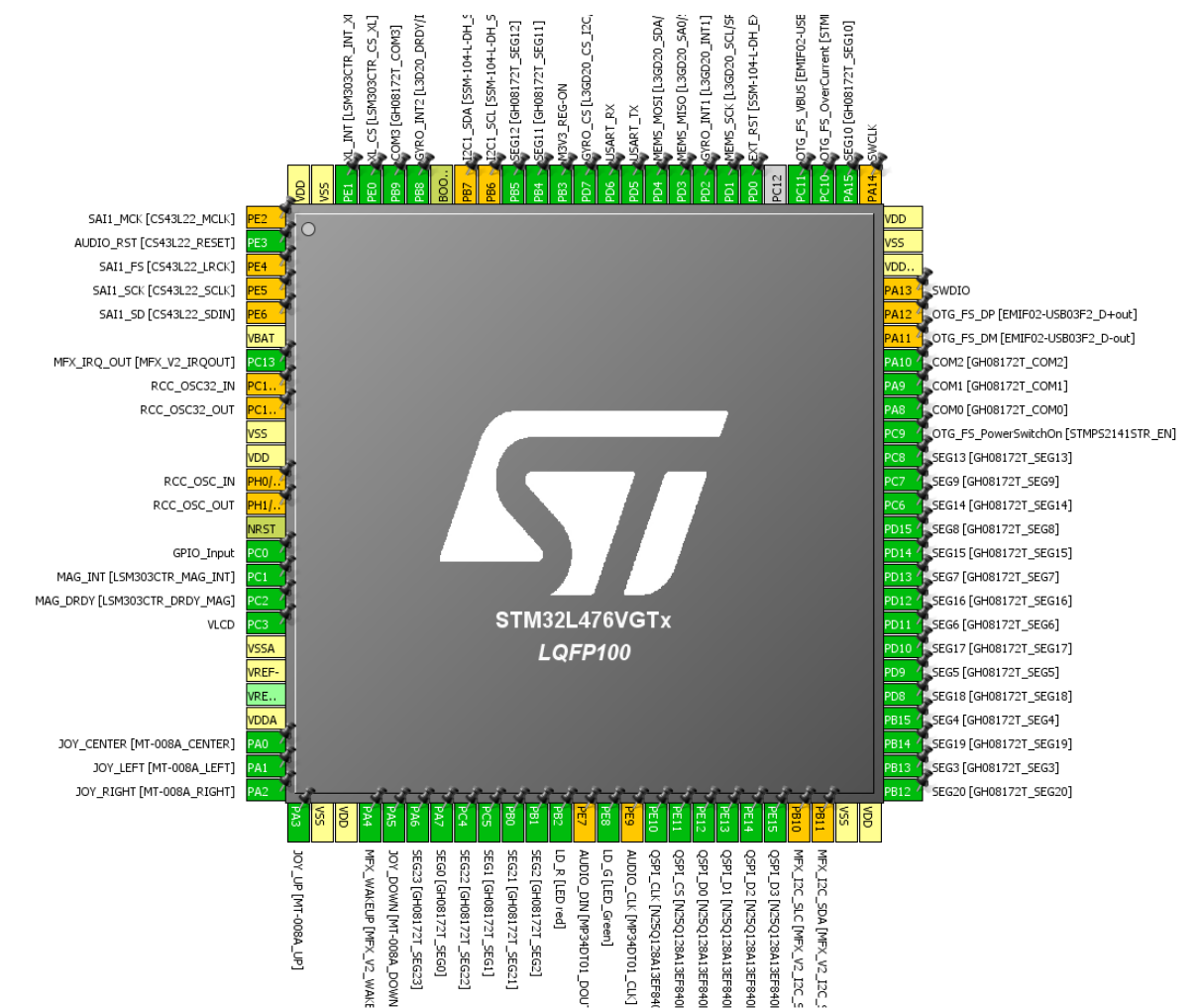
1 Opis projektu

Projekt ma na celu takie oprogramowanie zestawu rozwojowego, by na podstawie odczytów układów sensorycznych (MEMS) móc stwierdzić kto nosi płytkę w danym momencie. Projekt zakłada, że osoba nosząca płytkę ma ją przymocowaną do jednej z nóg, imitując noszenie telefonu w kieszeni.

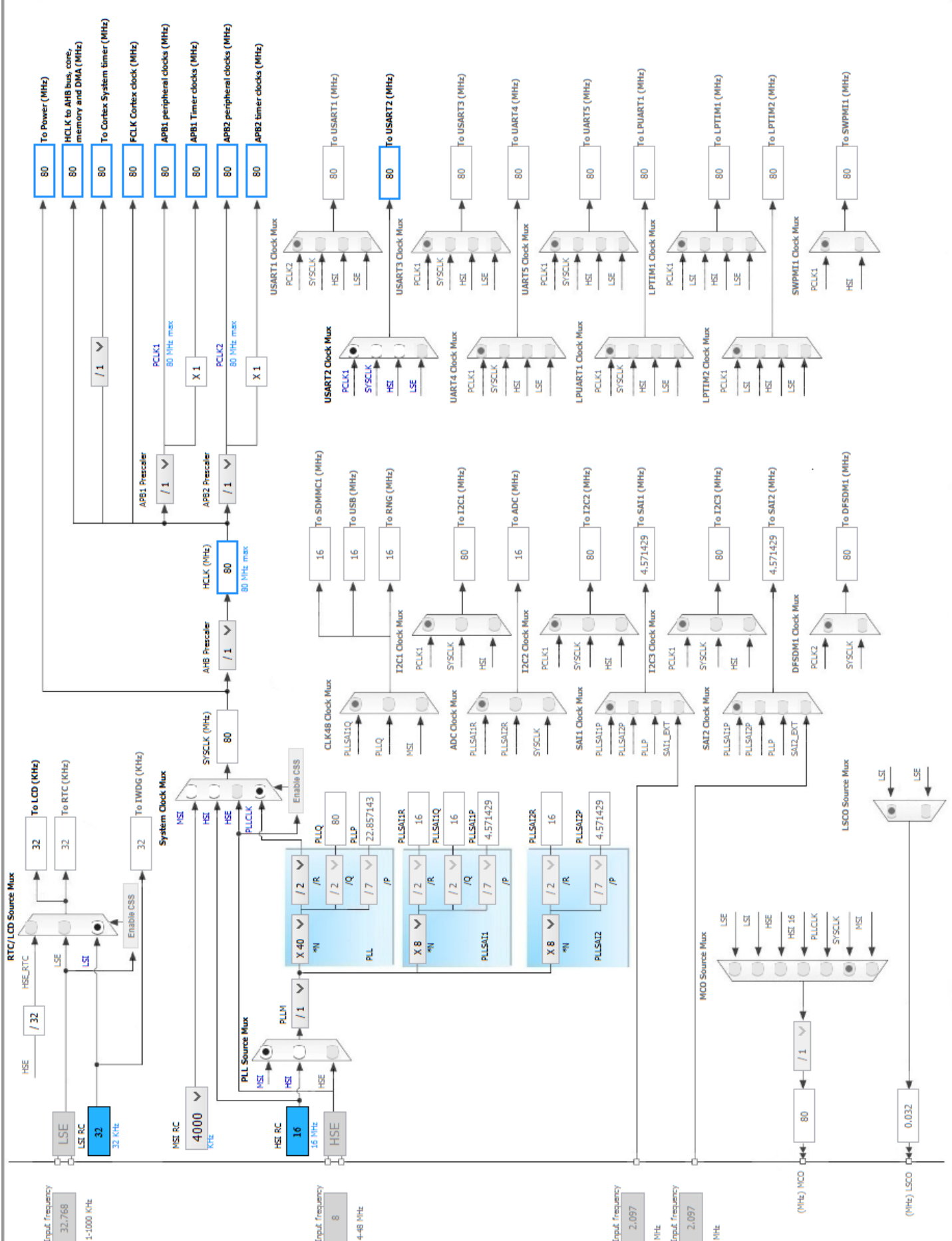
W tej części projektu udało się uruchomić wszystkie potrzebne peryferia oraz układy znajdujące się na płytce ewaluacyjnej. Udało się także nawiązać komunikację niezbędną podczas debugowania urządzenia. Skutecznie wykorzystano procesor zmiennoprzecinkowy do obliczania transformaty fouriera czy innych związanych z tym obliczeń. Zaimplementowano na mikrokontrolerze algorytm pozwalający na klasyfikację użytkowników, który wcześniej był rozwijany na komputerze.

2 Konfiguracja mikrokontrolera

Na rysunku 1 i 2 przedstawiono konfigurację mikrokontrolera. Użyto największego możliwego taktowania w celu jak najszybszego liczenia operacji zmiennoprzecinkowym takich jak transformata Fouriera.



Rysunek 1: Konfiguracja pinów mikrokontrolera



Rysunek 2: Ustawienia taktowania mikrokontrolera

3 Konfiguracja peryferiów

3.1 SPI2

Użyty został protokół komunikacji SPI w trybie Full-Duplex Master na pinach PD4, PD3 oraz PD1 (kolejno MOSI, MISO i SCK). Przy taktowaniu procesora 80MHz baudrate wynosi 10MHz. Jako chip select wykorzystywane są piny PD7 dla żyroskopu oraz PE0 dla akcelerometru.

3.2 LCD

W projekcie wykorzystywany jest interfejs do wyświetlacza LCD. Jak widać na rysunku 1 wykorzystuje on bardzo dużą ilość pinów. Wykorzystano standardowe ustawienia zaimplementowane w bibliotekach do STM32L476Discovery, między innymi parametr Duty Selection ustawiony został na 1/4.

3.3 QUADSPI

Pamięć flash podłączona jest do mikrokontrolera za pomocą szybkiego protokołu QUADSPI wykorzystującego 4 linie danych. Prędkość transmisji została ustawiona na 80MHz.

3.4 USART

Do debugowania użyto portu USART2 w trybie asynchronicznym na pinach PD6 i PD5 z baudratem 115200 bitów na sekundę.

3.5 TIMER6

Jako podstawy czasu użyto timera 6 generującego cyklicznie co 60ms przerwanie. Aby to osiągnąć ustawiono prescaler na 1000 oraz counter period na 4799.

3.6 Podsumowanie peryferiów

Poniżej przedstawiono podsumowanie najważniejszych pinów używanych przez peryferia mikrokontrolera. Nie uwzględniono pinów używanych przez wyświetlacz LCD.

```
PERIPHERALS MODES FUNCTIONS PINS
QUADSPI Quad SPI Line QUADSPI_BK1_I00 PE12
QUADSPI Quad SPI Line QUADSPI_BK1_I01 PE13
QUADSPI Quad SPI Line QUADSPI_BK1_I02 PE14
QUADSPI Quad SPI Line QUADSPI_BK1_I03 PE15
QUADSPI Quad SPI Line QUADSPI_NCS PE11
QUADSPI Quad SPI Line QUADSPI_CLK PE10
SPI2 Full-Duplex Master SPI2_MISO PD3
SPI2 Full-Duplex Master SPI2_MOSI PD4
SPI2 Full-Duplex Master SPI2_SCK PD1
USART2 Asynchronous USART2_RX PD6
USART2 Asynchronous USART2_TX PD5
```

3.7 GPIO

W projekcie wykorzystywane są także cyfrowe wyjścia oraz wejścia, między innymi dla 2 diod LED czy do obsługi joysticka. Ich podłączenie można znaleźć w dokumentacji płytki ewaluacyjnej [3].

4 Opis wykorzystywanych układów

Sposób podłączenia układów na płytce STM32L476DISCOVERY można znaleźć w jej opisie [3]. Sposób inicjalizacji akcelerometru oraz żyroskopu opisany jest w ich dokumentacjach technicznych [1] i [2].

4.1 Akcelerometr

Akcelerometr został zainicjalizowany w trybie ciągłego pomiaru z częstotliwością 50Hz oraz skali pomiaru $\pm 4g$. Aby to osiągnąć należy ustawić odpowiednie wartości w rejestrach opisane w poniższej tabeli:

Rejestr (adres rejestru)	wartość
CTRL_REG1_A (0x20)	0x27
CTRL_REG4_A (0x23)	0x21

4.2 Żyroskop

Żyroskop został zainicjalizowany w trybie ciągłego pomiaru z częstotliwością 95Hz oraz skali pomiaru ± 500 stopni na sekundę. Ustawiono także tryb zapisu pomiaru w notacji „little-endian”. Aby to osiągnąć należy ustawić odpowiednie wartości w rejestrach opisane w poniższej tabeli:

Rejestr (adres rejestru)	wartość
CTRL_REG1 (0x20)	0x23F
CTRL_REG4 (0x23)	0x10

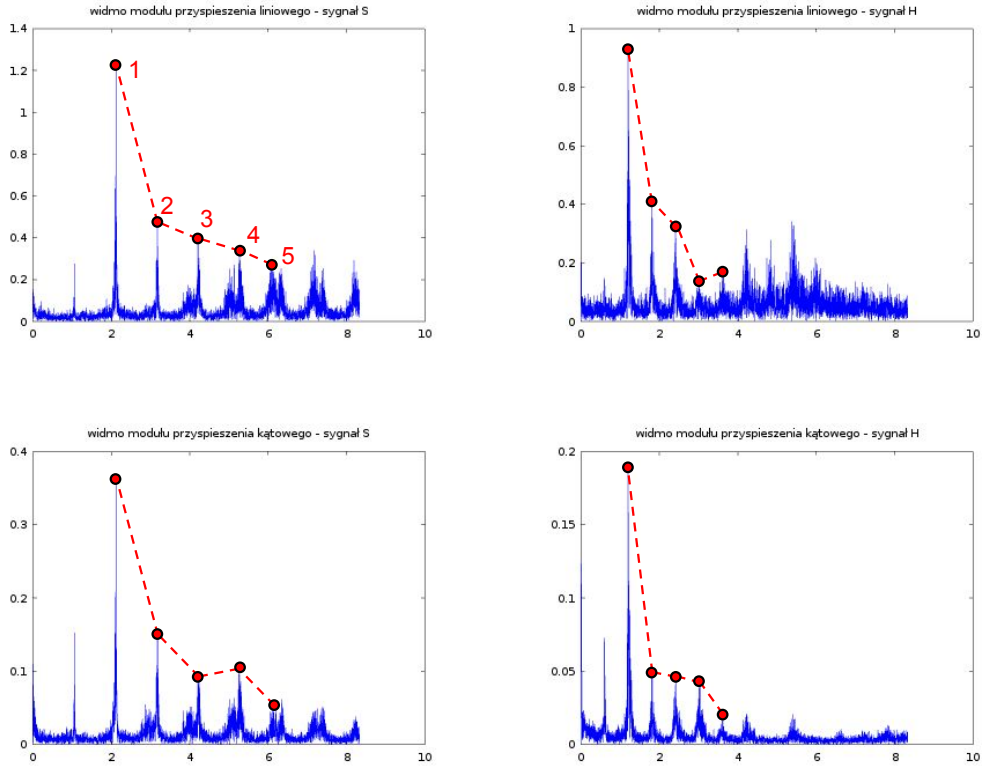
5 Opis działania programu

5.1 Algorytm rozpoznawania chodu

W projekcie zastosowano autorski algorytm rozpoznawania chodu klasyfikujący użytkowników na podstawie modułów sygnału z akcelerometru oraz żyroskopu. W opisie działania algorytmu sygnałem nazywany będzie moduł przyspieszenia liniowego o długości okna 128 próbek zbieranych co 60ms, co przekłada się na sygnał o długości około 7,7s. Nazwijmy użytkowników H oraz S.

5.2 Opis algorytmu

W trakcie badania sygnału zauważono, że harmoniczne sygnałów chodu użytkowników H oraz S zachowują swoje proporcje względem siebie niezależnie od momentu rozpoczęcia badania sygnału, a co istotniejsze, są inne dla chodu H i S jak pokazano na rysunku 3.

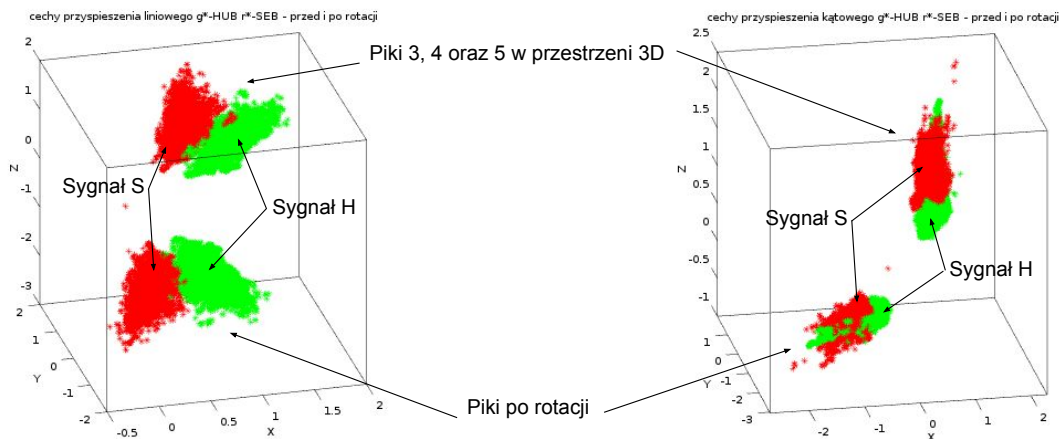


Rysunek 3: Charakterystyczny kształt składowych widma chodu.

Zauważmy, że jesteśmy w stanie niezależnie analizę od prędkości chodu obserwując jedynie amplitudę lub moc kolejnych pików widma, pomijając tym samym częstotliwość na której się one pojawiają. Ponadto, skalując wysokość pików tak, by pierwszy z nich zawsze przyjmował wartość 1, niezależniamy estymację od całkowitej mocy sygnału. Otrzymujemy w ten sposób ciąg wartości charakteryzujący danego użytkownika.

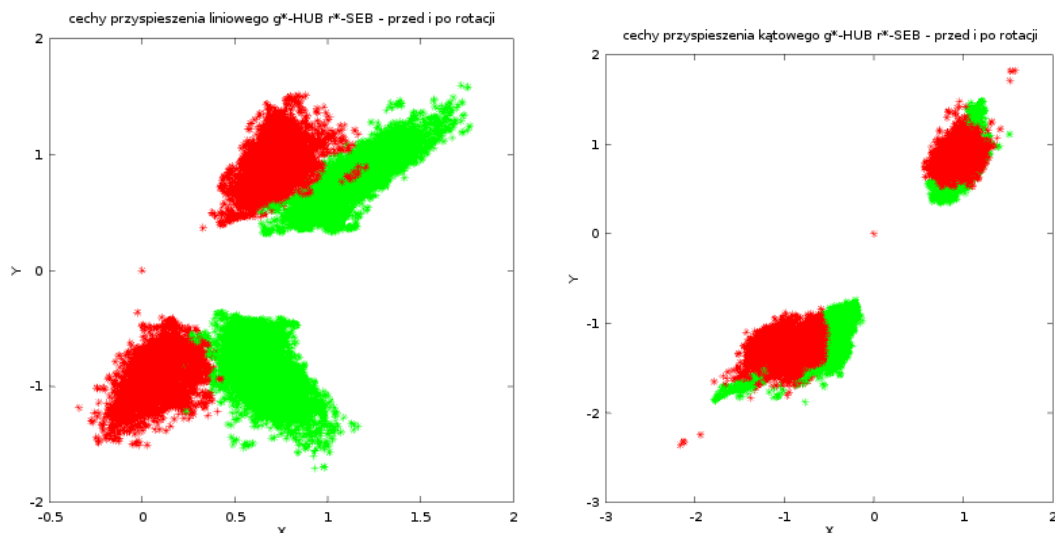
Kolejną eksperymentalnie wyznaczoną własnością jest to, że im wyższe harmoniczne badamy, tym bardziej się one między sobą różnią pomiędzy sygnałami¹.

Posiadając te informacje opracowano algorytm czerpiący dane z przeskalowanych pików 3, 4 oraz 5 otrzymanych z estymacji na oknie 128 próbek z ponad 10 minut pomiarów dla sygnałów H oraz S. Narzucając na wyznaczone w ten sposób piki interpretację geometryczną otrzymujemy łatwo separowalne obszary punktów widoczne na rysunku 4.



Rysunek 4: Charakterystyczny kształt składowych widma chodu

¹ Nie należy zapominać, że nie posiadamy pełnej informacji o sygnale z powodu ignorowania fazy sygnału jak i ograniczonego pasma obserwacji (w przypadku próbkowania 60ms jest to 16,5Hz).



Rysunek 5: Charakterystyczny kształt składowych widma chodu - rzut na płaszczyznę XY

Dokonując odpowiedniej rotacji owych chmur punktów, możliwe staje się rozróżnianie przynależności punktu do użytkownika na podstawie pojedynczej zmiennej (w przypadku ukazanym na rysunku 5 są to zmienne x).

$$Estymator = [(1 \ 0 \ 0) Rot(\vec{v}, \alpha) \vec{P}ik_{345} = (w_1 \ w_2 \ w_3) \vec{P}ik_{345}, \quad (1)$$

gdzie $\vec{P}ik_{345}$ oznacza wektor pionowy składający się z przeskalowanych pików 3, 4 oraz 5.

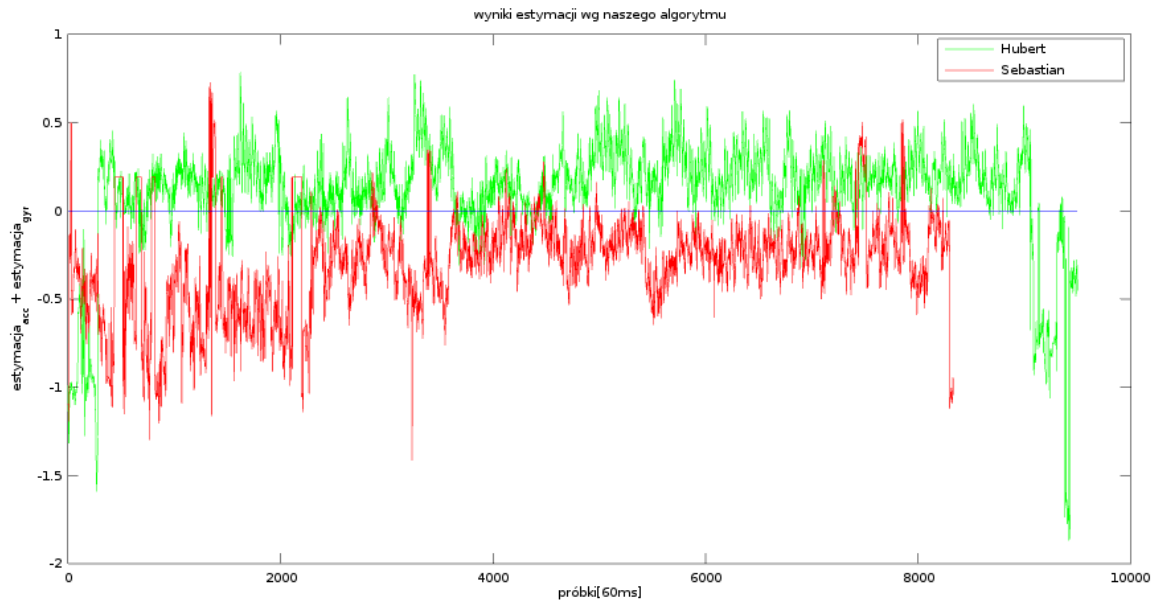
5.3 Właściwy algorytm

Zaimplementowany algorytm oblicza FFT modułu sygnału z akcelerometru oraz żyroskopu, poszukuje głównego pików znajdując maksimum wartości w całym widmie, po czym korzysta z wielokrotności jego indeksu do znalezienia kolejnych pików. By uniknąć drobnych fluktuacji w położeniu kolejnych pików sumowane są okolice danego indeksu i suma ta zwracana jest jako wartość pików, dzieląc przy tym każdy z nich przez pierwszy, z założenia największy, pik². Tak wyznaczonych jest 5 pierwszych pików. Pierwszy z nich jest ignorowany, gdyż zawsze wynosi 1, a drugi z nich jest pomijany ze względu na niewielką ilość niesionych informacji. Do kolejnych trzech wartości stosowana jest transformacja (2) wywodząca się z obrotu chmur punktów (rysunki 4 i 5) z odjęciem pewnej stałej, tak by rozróżnić osoby testując czy estymator jest dodatni bądź nie.

$$Estymator = (w_x \ w_y \ w_z) \vec{P}ik_{345} - const, \quad (2)$$

Wartość *Estymator* obliczana jest odpowiednio z innymi stałymi $(w_x \ w_y \ w_z)$ i *const* tak, by dodatnia wartość oznaczała sygnał H, a ujemna sygnał S zarówno dla sygnału z akcelerometru oraz sygnału żyroskopu. W ostatnim kroku dodawane są do siebie wartości *Estymator* pochodzące od akcelerometru oraz żyroskopu w celu zwiększenia redundancji danych. Na rysunku 6 przedstawiono wykres estymacji względem czasu dla sygnału zebranego podczas pracy nad algorytmem.

²w ten sposób realizowane jest skalowanie



Rysunek 6: Charakterystyczny kształt składowych widma chodu - rzut na płaszczyznę XY

6 Możliwości aplikacji

Aplikacja pozwala na 2 tryby działania:

- zapis danych do pamięci flash z możliwością ich późniejszego wysłania przez UART,
- analizę chodu.

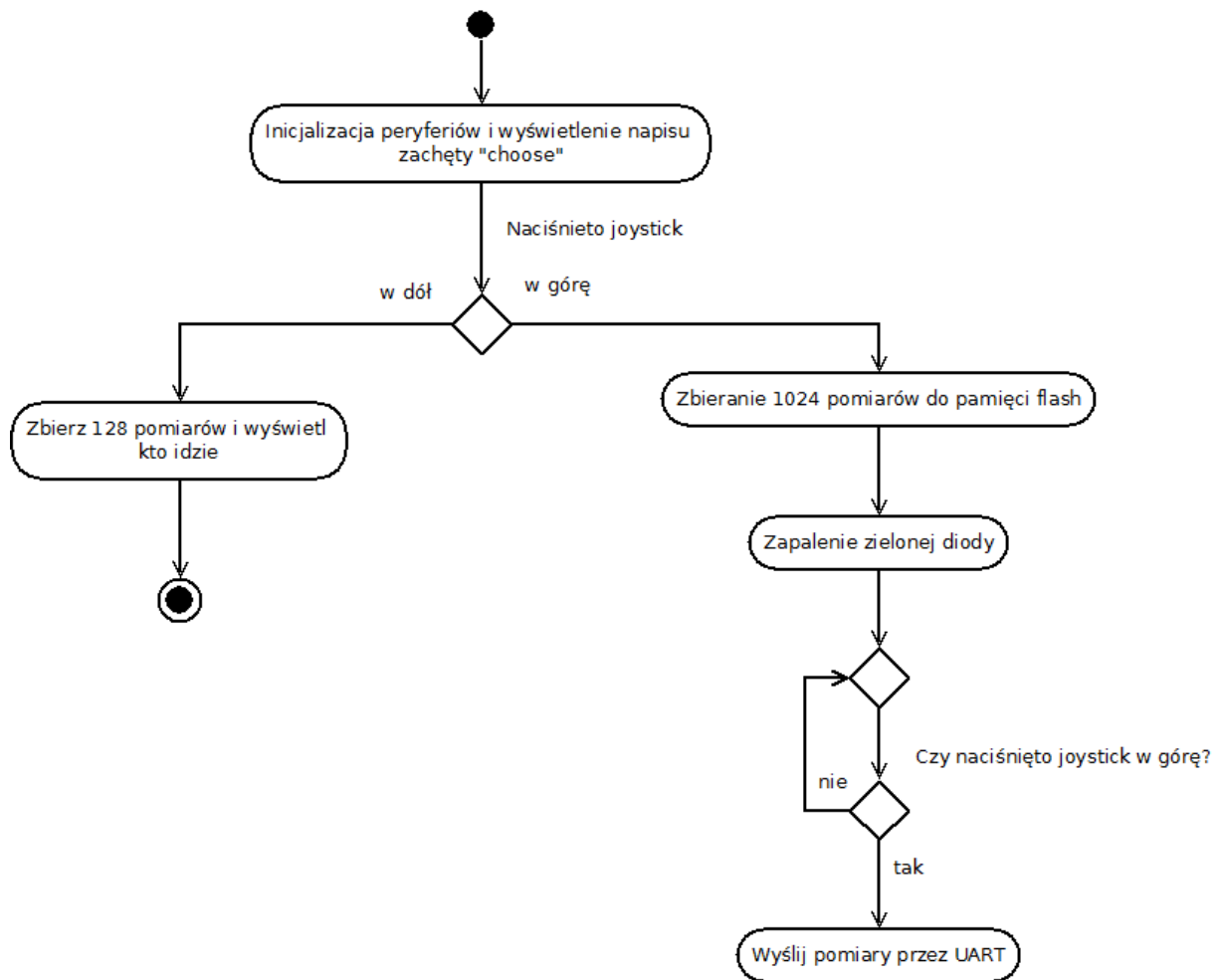
Na rysunku 7 przedstawiono diagram przepływu stworzonej aplikacji.

7 Podsumowanie

Udało się zrealizować wszystkie postawione zadania, a analiza chodu działa z zaskakująco dobrą skutecznością.

Literatura

- [1] ST, LSM303C, czerwiec 2014
- [2] ST, L3GD20, luty 2013
- [3] ST, UM1879 User manual Discovery kit with STM32L476VG MCU, marzec 2016



Rysunek 7: Diagram przepływu aplikacji