Knock Knock

Systemy Mikroprocesorowe II

Projekt końcowy

Agnieszka Lupierz, Adrian Barnaś Elektronika III rok

1. Założenia projektowe

1.1. Wstęp

Projekt realizowany jest w ramach laboratorium projektowego przedmiotu Systemy Mikroprocesorowe II.

1.2. Opis urządzenia

Urządzenie będzie wzorowane na urządzeniu KNOCKI. Zadaniem systemu będzie detekcja stuknięć przez akcelerometry, interpretowanie ich i w zależności od wykrytej sekwencji wykonanie jednej z zaprogramowanych akcji. Do detekcji stuknięć wykorzystamy układ MPU6050, komunikujący się z mikrokontrolerem poprzez magistralę I^2C. Wykorzystanym przez nas mikrokontrolerem będzie Kinetis L na płytce NXP Freedom KL46Z.

1.3. Logika działania

Przygotowane mamy dwie koncepcje logiki działania urządzenia. Preferowanym sposobem wykrywania stuknięć będzie generowanie przerwań, kiedy zostanie wykryte przyspieszenie przez układ MPU6050, następnie odczytanie danych z akcelerometrów i sprawdzenie czy wykryty ruch to rzeczywiście stuknięcie. Zaletą takiego podejścia do problemu byłoby ograniczenie zużycia energii przez procesor, który mógłby być uśpiony, jeśli urządzenie pozostawało by bez ruchu. Trudne może być określenie progu zadziałania programu. Drugim podejściem byłaby ciągła analiza pobieranych z czujnika informacji i rozpoznawanie, kiedy wykrytym przez urządzenie ruchem jest wstrząs podłoża spowodowany stuknięciem.

1.4. Faza prototypu

Docelowo zamierzamy stworzyć prototyp urządzenia, który wykryte sekwencje będzie sygnalizował zmianą stanu na wyjściu kontrolera(np. różne diody dla różnych sekwencji.

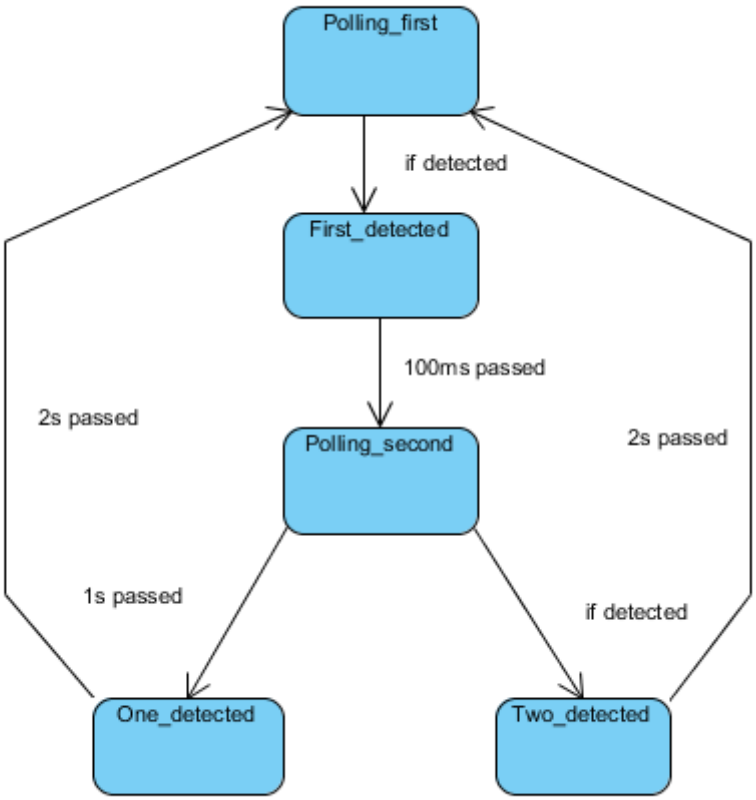
2. Realizacja projektu

2.1. Wykorzystane narzędzia

Do realizacji projektu wykorzystaliśmy środowisko programistyczne Keil uVision. Program był napisany w całości w C++, a kontrola wersji była wykonywana z użyciem narzędzia GIT. Repozytorium git było zdalnie umieszczone na serwerze serwisu GitHub, gdzie wspólnie umieszczaliśmy wprowadzane zmiany. Do stworzenia dokumentacji wykorzystaliśmy program WORD z pakietu MS Office, do stworzenia diagramu stanów narzędzie Visual Paradigm, do wizualizacji danych środowiska MATLAB.

2.2. Struktura programu

Program został zrealizowany jako maszyna stanów. Maszyna przyjmowała jeden z pięciu stanów zgodnie z logiką zamieszczoną na diagramie:

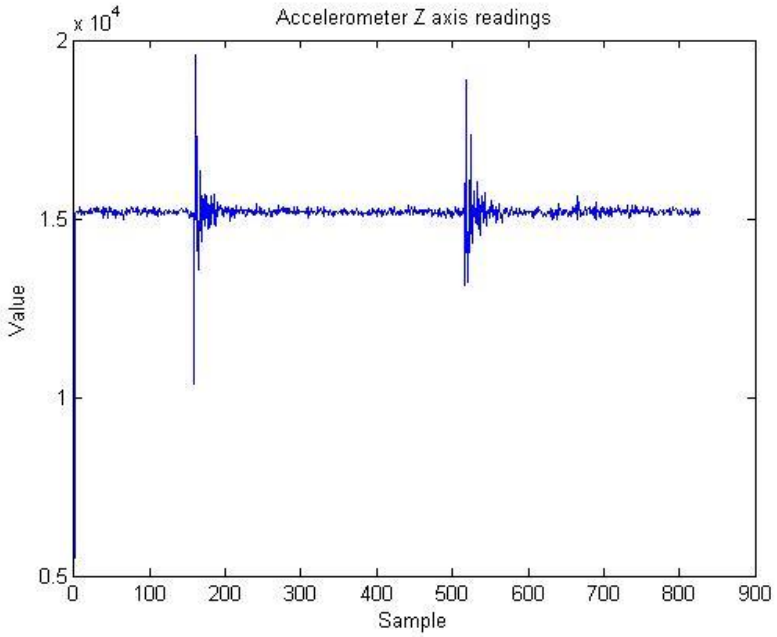


- W stanie Polling_first odpytujemy czujnik o surowe dane przyspieszenia w osi Z czujnika i porównujemy z zadanym poziomem odniesienia aż do przekroczenia tego poziomu (detected)
- W stanie First_detected sygnalizujemy wykrycie pierwszego stuknięcia oraz odczekujemy na ustanie drgań.
- W stanie Poling_second znów odpytujemy czujnik przez maksimum 1000ms i jeśli w tym czasie znów wykryjemy uderzenie to następuje „detected”
- W stanach One_detected oraz Two_detected programujemy akcje odpowiednie dla każdej z sekwencji. Stan trwa 2000ms.

Do realizacji obsługi czasu w programie użyliśmy licznika systemowego SysTick, do komunikacji z akcelerometrem użyliśmy sprzętowego I^2C, do wyświetlania wyników używaliśmy diod i wyświetlacza slcd, używaliśmy również UART w celu analizy danych wyjściowych z akcelerometru.

2.3. Wykrywanie stuknięć

Wykrywanie stuknięć odbywa się przez porównanie odpytywanych wartości przyspieszenia w osi ’z’ z ustalonym progiem zadziałania. Na potrzeby wyznaczenia progu zadziałania zebraliśmy surowe dane akcelerometru, wysłaliśmy na komputer z pomocą UART i wykreśliliśmy ich wykres w programie MATLAB:



Wykres przedstawia dwa stuknięcia mocniejsze i słabsze. Oba dały się wyraźnie oddzielić od normalnych odczytów akcelerometru, a próg zadziałania ustaliliśmy na połowę piku wywołanego słabszym stuknięciem.

2.4. Sposób drugi – działanie na przerwaniach

Sposobem na wykrywania stuknięć jest generowanie przerwań, następuje wykrycie przyspieszenia przez układ MPU6050, następnie odczytanie danych z akcelerometrów i sprawdzenie czy wykryto ruch. Zaletą takiego podejścia jest ograniczenie zużycia energii przez procesor, który jest uśpiony, jeśli urządzenie pozostawało by bez ruchu i czeka na sygnał.

W tym podejściu procesor nie musi pytać po I2C, tylko rejestruje impulsy pojawiające się na pinie INT po stuknięciu.

Poniższa tabela przedstawia rejestry, które należy ustawić w pamięci układu MPU6050, aby uruchomić możliwość działania na przerwaniach:

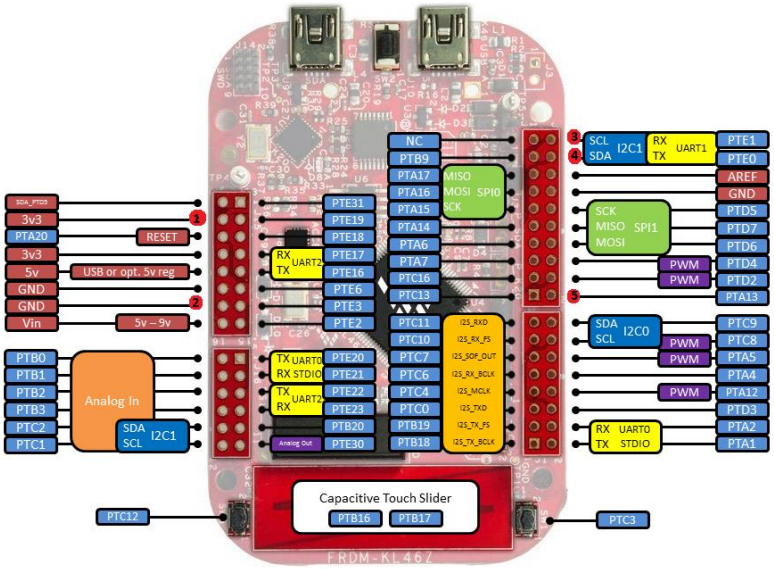
Adres rejestru	Wartość	Rejestr
0x1c	1	Accelerometer Configuration
0x37	0	INT Pin / Bypass Enable Configuration
0x1f	10	Motion Detection Threshold
0x20	1	Motion Detection Duration
0x69	0x13	Motion Detection Control
0x38	64	INT_ENABLE

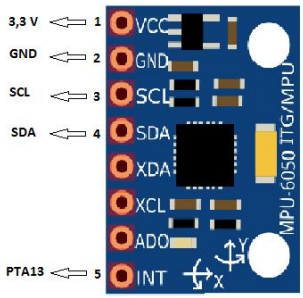
2.5. Sygnalizacja wyników

Na potrzeby projektu przyjęliśmy dwie sekwencje. Pierwsza to pojedyncze uderzenie w powierzchnię na której leży czujnik, druga to podwójne uderzenie w powierzchnie w przeciągu 1,1s. Sekwencje w podanej kolejności sygnalizowane są na wyświetlaczu slcd jako 0001 i 0002. Stan Polling_first sygnalizowany jest na wyświetlaczu jako AAAA. Jednocześnie wykrycie pierwszego uderzenia sygnalizowane jest diodą zieloną, natomiast wykrycie drugiego stuknięcia diodą czerwoną.

2.6. Schemat podłączenia

Aby sprawdzić działanie programu należy podłączyć wybrane piny NXP Freedom KL46Z z układem MPU6050 zgodnie z poniższym schematem:





3. Podsumowanie

Efektem naszej pracy jest działające urządzenie (prototyp) oraz aplikacja, w trakcie pisania wersja V2.0

Do obsługi każdego z peryferii napisaliśmy osobną klasę będącą uniwersalnym i łatwym do rozwinięcia „biblioteką”. Kod sam w sobie jest podstawą, na której może zostać zbudowany większy projekt rozwijający stworzoną przez nas funkcjonalność. Efekty naszej pracy można obejrzeć i wykorzystać w ramach otwartego kodu na serwerze github pod adresem:

https://github.com/designLaboratory/KnockKnock