Myszka akcelerometryczna

Paweł Jedynak 148597
Michał Mielnicki 148835

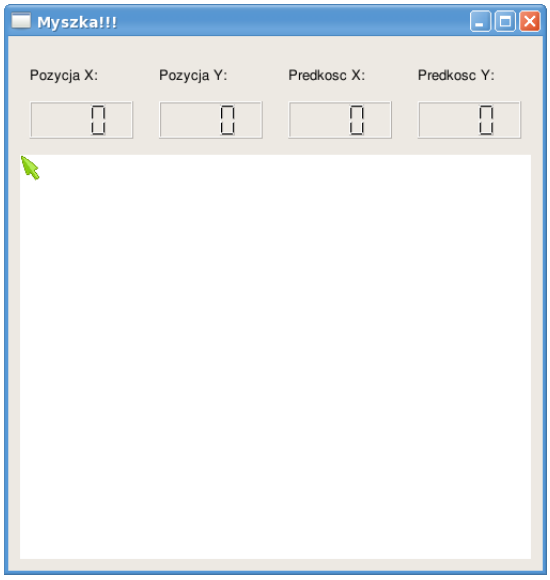
18 czerwca 2009

1 Opis rozwiązywanego problemu

Celem projektu było stworzeniem układu elektronicznego działającego na zasadzie myszki komputerowej. W celu sprawdzania działania stworzonego układu został napisany program do wizualizacji obrazującej ruch myszki. Oprogramowanie zostało stworzone przy użyciu biblioteki Qt oraz funkcji systemowych wykorzystanych do komunikacji z myszką. Na podstawie przyspieszeń w danych kierunkach odczytanych z dwóch akcelerometrów jest określana pozycja myszki i jej prędkości przemieszczania się. Dzięki temu, że zastosowane akcelerometry są trój-osiowe będzie można również korzystać z dwóch przycisków jak w prawdziwej myszce komputerowej.

2 Część programowa

Został napisany program wizualizujący ruch myszki Rysunek 1. Myszka jest sterowana prędkościowo. Program wizualizujący oblicza położenie na podstawie prędkości przesłanych z mikrokontrolera Freescale MC68HCS12A64 poprzez interfejs szeregowy w standardzie RS232. Program oprócz wizualizacji ruchu myszki pokazuje jej prędkość względem osi X oraz osi Y, jak również położenie (x,y). Dodatkową możliwością programu jest obrazowanie punktami, w których miejscach nastąpiło kliknięcie prawym bądź lewym przyciskiem.



Rysunek 1: Screen programu do wizualizacji

2.1 Interfejs graficzny

Program do wizualizacji ruchu myszki jest zbudowany z jednego okienka. Zawiera w sobie pole, w którym obrazowane są działania myszki, czy to ruch w dowolną stronę czy też kliknięcia. W celu pokazania ruchu myszki został wczytany obraz strzałki, który potrafi się przemieszczać w obrębie tego pola. Przy pomocy wyżej wymienionej strzałki można zaobserwować położenie

myszki na ekranie bądź też z jaką szybkością porusza się w danym kierunku. Zostały nałożone ograniczenia, które nie pozwalają kursorowi na przemieszczenie się za obręb pola przeznaczonego do symulacji, tak jak ma to miejsce w rzeczywistości, że kursor myszki nie może znaleźć się poza ekranem monitora. Po uzyskaniu skrajnych wartości położeń prędkości są zerowane, ponieważ z wcześniej wymienionych ograniczeń wynika, że kursor nie może sie dalej przemieszczać.
Program ma również możliwość obrazowania zaistniałych kliknięć. W przypadku, gdy w danym momencie nastąpi kliknięcie, w miejscu, w którym aktualnie znajduje się myszka pojawia się punkt odpowiedniego koloru. Dla kliknięcia prawym przyciskiem jest to kolor zielony natomiast dla lewego kolor czerwony.
Oprócz samego pola obrazującego poczynania myszki znajdują się również cztery pola wyświetlające położenie myszki względem osi X oraz osi Y, a także z jakimi prędkościami myszka się porusza względem osi X i osi Y. Prędkości są przeskalowane od -10 do 10 pikseli podczas każdego uaktualnienia pola wyświetlającego myszkę.

2.2 Komunikacja

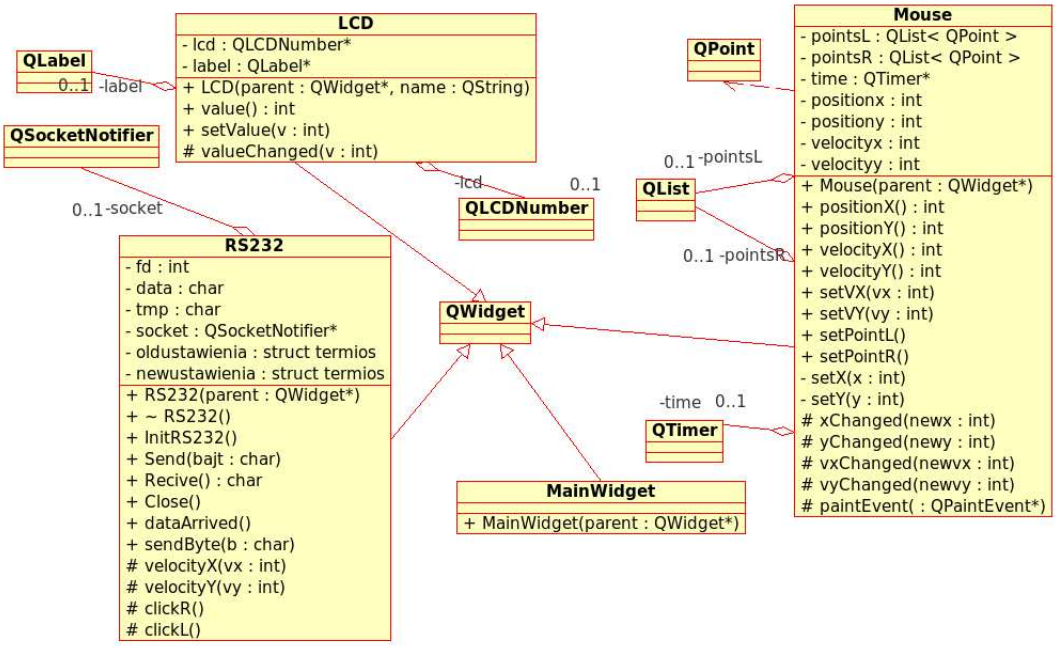
Komunikacja z układem działającym jako myszka jest zrealizowana za pomocą połączenia szeregowego asynchronicznego. Prędkość transmisji wynosi 19200b/s, a format otrzymywania danych ma postać 8N1, czyli 8 bitów danych, brak bitu parzystości oraz jeden bit stopu.
Konfiguracja łącza odbywa się przy wykorzystaniu biblioteki systemowej `termios`. Natomiast dane są odbiera przy wykorzystaniu funkcji systemowej `read`. Dokładny opis konfiguracji oraz odczytu bajtu znajduje się w [8].

Funkcja	Wartość	Ilość	Opis reakcji
'x'=0x78	data	2	Prędkość o wartości `data` względem osi X
'y'=0x79	data	2	Prędkość o wartości `data` względem osi Y
'r'=0x72	-	1	Informacja o kliknięciu prawym klawiszem
'l'=0x6C	-	1	Informacja o kliknięciu lewym klawiszem

Tabela 1: Opis otrzymywanych danych przez program

Tabela 1 przedstawia zestawienie danych jakie może otrzymać program od myszki. Pierwsza kolumna określa funkcję do wykonania przez program, natomiast druga kolumna wartość prędkości, ale tylko w wypadku 1 i 2 funkcji. Ilość oznacza ile bajtów program musi odebrać, aby wykonanie danej funkcji było możliwe. Ostatnia kolumna zawiera krótki opis wymienionych funkcji.
W celu sygnalizacji nadejścia nowej informacji przez port szeregowy została użyta klasa `QSocketNotifier` wysyłająca sygnał mówiący, że w buferze znajduje się bajt do odczytania.

2.3 Diagram klas w języku UML



Rysunek 2: Diagram UML

2.4 Diagram przepływu sterowania

Rysunek 3: Diagram przepływu sterowania

Rysunek 3 przedstawia diagram przepływu danych w aplikacji służącej wizualizacji ruchu myszki. Najpierw zostaje zainicjowane połączenie z myszką, następnie tworzy się okienko wraz z polem do obrazowania działania

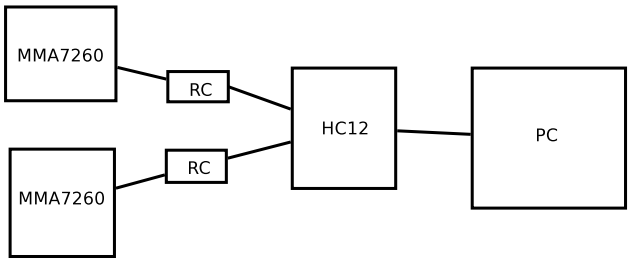
myszki. Nad odświeżaniem pola czuwa timer. Jeśli licznik timer'a nie jest przekroczony to w tym czasie odbywa się komunikacja z myszką. Jeśli otrzymane dane są poprawne następuje akutalizacja parametrów myszki, tzn czy został naciśnięty któryś z klawiszy bądź nastąpiła zamiana prędkości ruchu w dowolnym kierunku.

3 Część sprzętowa

Przedstawia opis budowy urządzenia działającego jako myszka komputerowa. Zawiera również schematy elektroniczne podzespołów, z których zbudowany jest moduł.

3.1 Opis urządzenia

Podstawowym elementem zbudowanego urządzenia jest układ scalony mikrokontrolera `MC9S12A64` firmy `Freescale`. Do pomiarów zostały wykorzystane dwa akceleroemtry trój-osiowe `MMA7260` również firmy `Freescale`. Każdy z akcelerometrów posiada 3 wyjścia analogowe mówiące o przyspieszeniach w kierunkach X,Y oraz Z. Zaletą akcelerometrów `MMA7260` jest ich mały rozmiar oraz mały pobór prądu ok. 500μ A. Każdy z akcelerometrów jest zasilany napięciem 3.3V oraz ma możiwość ustawienia czułości w czterech poziomach (1,5g/2g/4g/6g). Mikrokontroler `MC9S12A64` jest rozbudowanym kontrolerem posiadającym wiele możliwości. W projekcie została wykorzystana komunikacja szeregowa asynchroniczna (`SCI`), przerwanie cykliczne (`RTI`) oraz jeden przetwronik analog-cyfra, w którym zostało wykorzystanych 6 kanałów. Odczyt wartości z przetwornika jest dokonywany co 10ms w przerwaniu cyklicznym.



Rysunek 4: Schemat blokowy urządzenia

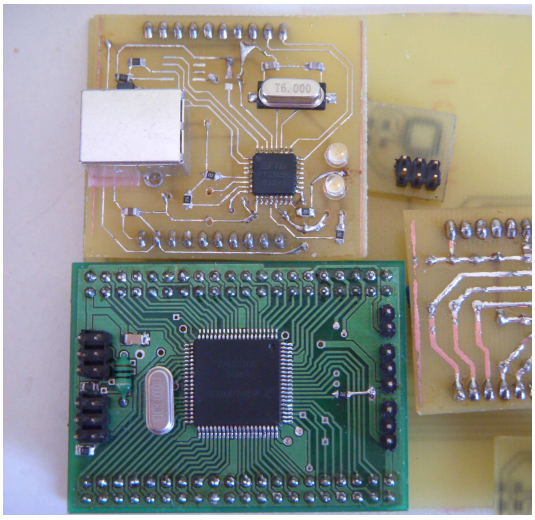
3.2 Opis funkcjonalności urządzenia

Podstawowym zadaniem urządzenia jest możliwość sterowania kursorem myszki. Sterowanie odbywa się prędkościowo. Z wartości odczytanych z akcelerometrów uzyskujemy prędkość wyskalowaną od -10 do 10, która jest wysyłana do komputera szeregowo w sposób asynchroniczny. Sygnał wysyłany przez akcelerometry jest analogowy, który zostaje podany bezpośrednio na przetwornik ADC w mikrokontrolerze `MC9S12A64`. Sygnał cyfrowy jest odpowiednio przetwarzany w mikrokontrolerze. Urządzenie oprócz możwliwości poruszania kursorem myszki może również symulować kliknięcia zarówno prawym jak

i lewym przyciskiem. Zostało to umożliwione dzięki temu, że wykorzystane akcelerometry są trój-osiowe. W celu wykrycia kliknięcia zostają poddane analizie przyspieszenia w kierunku pionowym do ziemi.

3.3 Hardware

W celu wykonania naszego projektu niezbędne było wykonanie funkcjonującego urządzenia. Wychodząc z założenia zaprojektowania układu w oparciu o mikrokontroler z rodziny HC12 skonstruowaliśmy poniższe urządzenie:



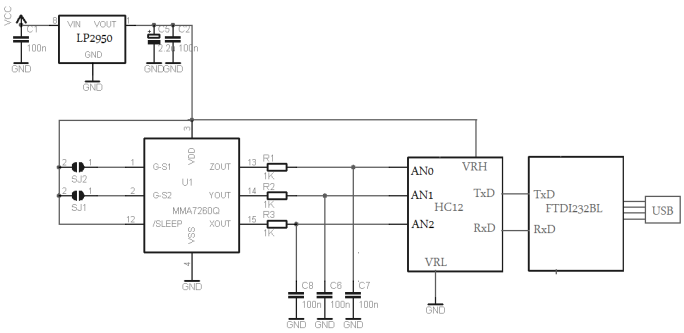
Rysunek 5: Myszka akcelerometryczna.

Na przedstawionym rysunku (Rysunek 5) widać całość skonstruowanego układu. Składa się on z 5 modułów:

- moduł mikrokontrolera MC9S12a64
- układ konwertera $RS232 <=> USB$
- filtr dolno-przepustowy
- dwie płytki z akcelerometrami MMA7260

Całość po połączeniu kablami oraz podłączeniu do portu USB komputera działa jak myszka komputerowa. Schemat blokowy skonstruowanego urządzenia przedstawia 6.

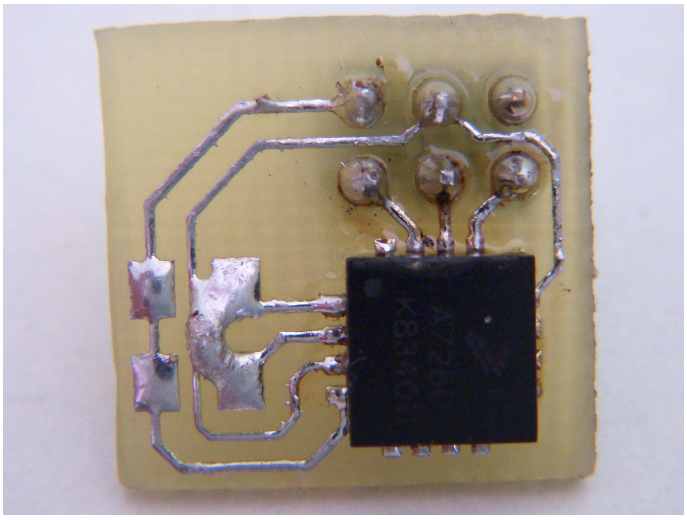
Jak widać na przedstawionym rysunku (6) akcelerometr jest podłączony do trzech pinów mikrokontrolera poprzed dolno-przepustowy filtr RC ($R = 1k\Omega$, $C = 100nF$). Mikrokontroler mierzy napięcie na wyjściach akcelerometru, wykonuje pojedyńcze całkowanie po czym wysyła uzyskane dane poprzez konwerter $RS232 <=> USB$ do komputera, gdzie zostają przetworzone na pozycje kursora myszki. Jest to wizualizowane w napisanym do tego programie. Szczegółowy opis modułów znajduje się w następnych podrozdziałach.



Rysunek 6: Schemat blokowy.

3.3.1 Płytka z akcelerometrem

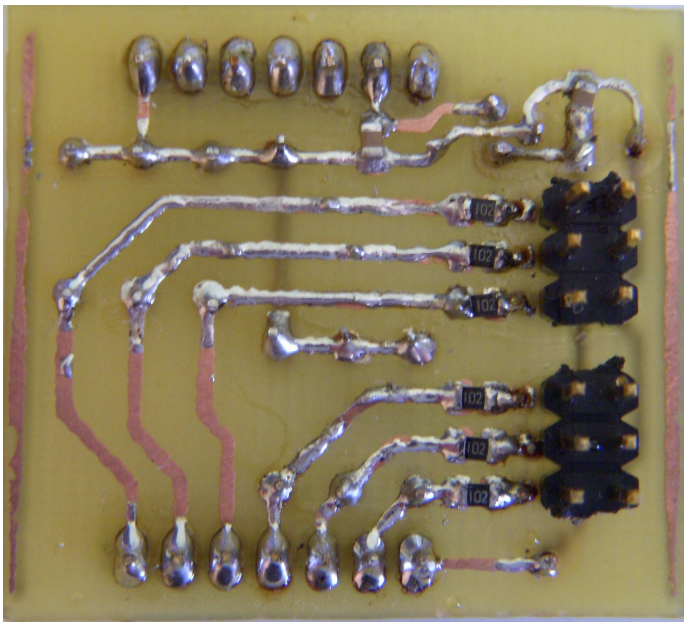
Jest najmniejszym i najmniej skomplikowanym elementem urządzenia. Płytka jest niewielkich wymiarów, aby mogła się zmieścić na palcu (jest niewiększa niż paznokieć). Na płytce (Rys. 7) umieściliśmy jedynie akcelerometr, 6 pinów sygnałowych oraz miejsce do zmiany czułości akcelerometru. Akcelerometr $MMA7260QT$ [3] dostępny jest jako darmowe próbki na stronie producenta ($http://freescale.com$). Jest on zasilany napięciem $3.3V$ dlatego niezbędne było zastosowania dodatkowego stabilizatora napięcia, który znajduje się na module z filtrem RC.
Akcelerometr mierzy napięcie w trzech osiach. Posiada zatem 3 wyjścia analogowe na których pojawia się napięcie zależne od sił działających na czujnik. W stanie spoczynku, tzn. gdy nie działają żadne siły na wyjściach akcelerometru powinno się pojawić napięcie równe połowie napięcia zasilania ($1,65V$). W rzeczywistości tak nie jest dlatego niezbędna jest kalibracja czujnika. Polega ona na zsumowaniu odpowiedniej ilości próbek (dobranej doświadczalnie) a następnie podzielenia uzyskanej sumy przez liczbę próbek. Dzięki temu uzyskujemy wartość średnią, którą będziemy nazywać wartością referencyjną i przyjmujemy ją jak wartość odniesienia do dalszych pomiarów.



Rysunek 7: Płytka z akcelerometrem.

3.4 Filtr dolno-przepustowy

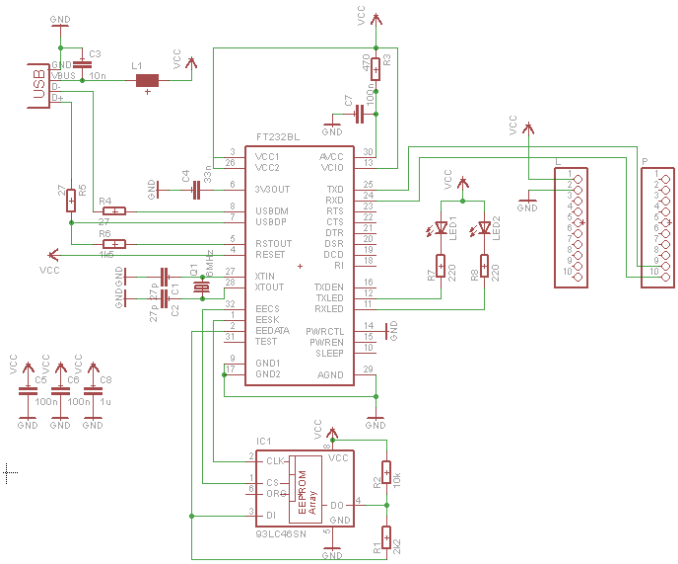
Zalecanym połączeniem czujnika do przetwornika analogowo cyfrowego [5]jest połączenie poprzez dolno-przepustowy filtr złożony z elementów RC. Taka konfiguracja czujnika zapewnia tłumienie częstotliwości generowanych przez czujnik. Na module z filtrem (8) umiszczono także stabilizator napięcie $Lp2950-3.3ACZ$. Jest to stabilizator typu Low-Drop (o niskim spadku napięcia) pozwalający na konwersję napięcia z 5V na 3.3V. Na płytce z filtrem umieszczono dwa porty (6 pinów) wyjściowe do płytek z akcelerometrami. W jednorzedowych pinach znajdują się wyjścia prowadzące do 6 pinów przetwornika A/C mikrokontrolera oraz wejścia zasilania, masy oraz napięcia odniesienia (3.3V) dla przetwornika A/C. Płyta główna całego urządzenie jest tak skonstruowana, że zamiast naszego modułu z filtrem dolno-przepustowym można włożyć moduł z przetwornikiem A/C komunikującym się z mikrokontrolerem za pomocą równoległego interfejsu SPI.



Rysunek 8: Płytka z filtrem dolno-przepustowym.

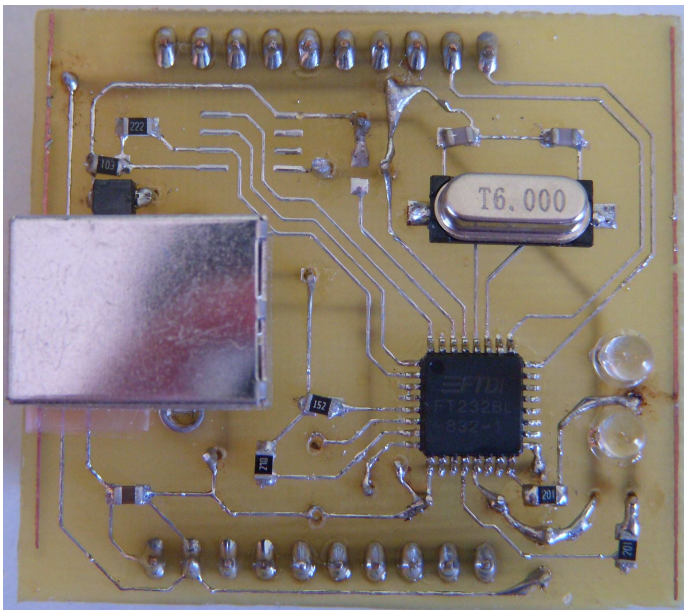
3.4.1 Układ konwertera $RS232 <=> USB$

Moduł ten jest skonstruowany na popularnym układzie $FT232BL$ [7] najczęściej wykorzystywanym do konwersji transmisji interfejsu szeregowego [6] (w naszym przypadku UART) do standardu USB. Układ ten od strony komputera widoczny jest jako wirtualny port `ttyUSB0`. Układ ten zapenia nie tylko konwersję przesyłanego sygnału, ale również translację napięć ze standardu $RS232$ do TTL dzięki czemu nie ma potrzeby używania dodatkowego translatora napięć (np. $MAX232$). Mikrokontroler komunikuje się z układem za pomocą dwóch linii sygnałowych (TxD oraz RxD). Całe urządzenie zasilane jest z portu USB, dlatego do pinów modułu doprowadziliśmy napięcie i masę. Na przedstawionym schemacie (9) można się bliżej zapoznać z zaprojektowanym modułem oraz jego wyprowadzeniami.



Rysunek 9: Schemat elektryczny modułu z układem $FTDI232BL$.

Widok zlutowanej plytki przedstawia rysunek 10.



Rysunek 10: Moduł z układem $FTDI232BL$.

Na płytce znajduje się miejsce na wlutowanie pamięci EEPROM, która pozwala na konfigurację urządzenia. Można dzięki niej ustawić dopuszczalny prąd urządzenia lub zmienić nazwę widoczną w systemie operacyjnym. Bardzo przydatnymi okazały się diody sygnalizujące transmisję w obie strony.

3.4.2 Mikrokontroler oraz płyta główna

Mikrokontroler MC9S12a64 [4] jest umieszony na module posiadającym 80 pinów dwurzędowych [1]. Do mikrokontrolera doprowadzone są sygnały (6

pinów przetwornika A/C) z akcelerometrów, napięcie odniesienia dla pomiaru napięcia, dwie linie danych interfejsu szeregowe, dodatkowo linie danych interfejsu SPI do modułu z filtrem, oraz napięcie oraz masa.
Na płycie głównej znajdują się wszystkie moduły oprócz akcelerometrów, które są przeznaczone do zamontowania na palcach użytkownika. Jedynie moduł z mikrokontrolerem nie byl zaprojektowany przez nas.

3.4.3 Podsumowanie hardware'u

Skonstruowane urządzenie działa jak myszka komputerowa. Jako przycisków użylismy dwa czujniki akcelerometryczne. Każdy z nich odpowiada jednemu przyciskowi myszki. Mierzone przetwornikiem A/C napięcia na wyjsciach akcelerometru są jednorazowo całkowane metodą trapezową. W ten sposób uzyskana prędkość czujników jest wysylana interfejsem szeregowym (z punktu widzenia mikrokontrolera) do komputera.
Czujniki akcelerometryczne warto ustawić na czulość 2g. Po przprowadzonych badaniach dla najmniejszej czułości (6g) urządzenie była praktycznie niestrerowalne, gdyż należalo zadziałać bardzo dużą siła aby ruszyć kursor.

3.5 Protokół komunikacji

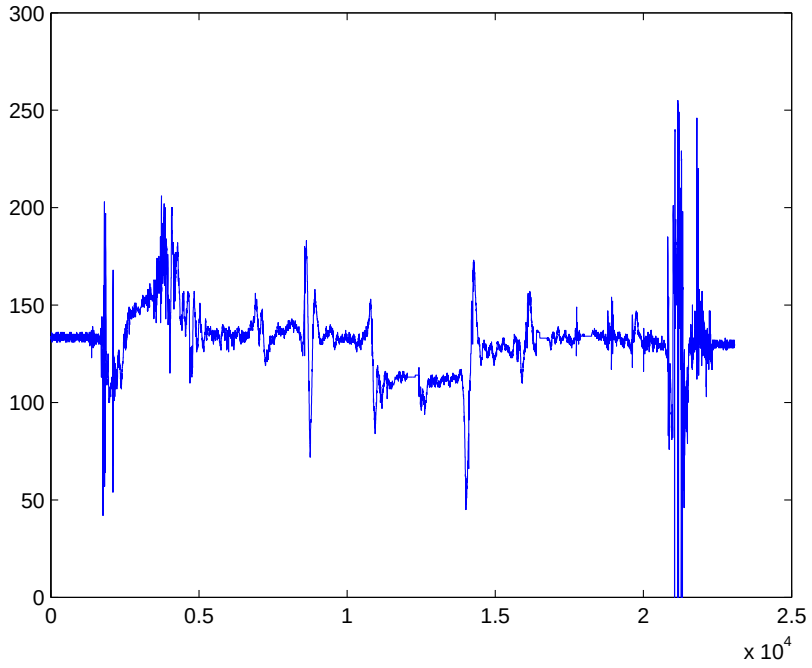
Do transmisji danych w stronę komputera został wyrzkosytany asynchroniczny szeregowy protokół komumnikacji. W mikrokontrolerze do tego celu zostało użyte SCI odpowiadające za transmisję szeregową. W celu konfiguracji komunikacji jest potrzeba ustawienia czterech rejestrów mikrokontrolera: `SCIBDH`, `SCIBDL`, `SCICR1` oraz `SCICR2`. Podstawowym parametrem konfiguracji łącza jest ustawienie prędkości transmisji. Szybkość przesyłania danych określa się wzorem:

$$SCIbaudrate = \frac{SCImoduleclock}{16 * SBR[12:0]} \tag{1}$$

gdzie SBR[12:0] to 12 najmłodszych bitów rejestru SCIBD. Prędkość ustawiona dla myszki wynosi 19200b/s. Natomiast połączenie jest w trybie 8N1.
Do połączenia się z komputerem potrzebujemy konwertera napięć na standard TTL przy użzyciu układu scalonego `MAX232` lub przez konwerter
$SCI <-> USB$ czyli układ `FT232BM` firmy `FTDI`.
Mikrokontrol po odczytaniu informacji z akcelerometrów analizuje dane oraz wysyła informacje do komputera klasy PC. Dane do wysłania prędkości są 2 bajtowe, natomiast informacje mówiące o kliknięciu jednym z przycisków są wysyłane jako jeden bajt. W celu inforamcji o prędkości jest najpierw wysyłany znak 'x' lub 'y' mówiący o kierunku prędkości a następinie bajt zawierający wartość danej prędkości. Natomiast jeśli nastąpiło kliknięcie jednym z przycisków to zostaje wysyłany bajt 'r' lub 'l' mówiący odpowiednio o prawym bądź lewym kliknięciu.

3.6 Eksperymenty

Badanie akcelerometrów były prowadzone z użyciem środowiska matematycznego `Matlab`. Odczyt wartości odbywał się przez mikrokotroler, który



Rysunek 11: Wykres przedstawiający pomiary z akcelerometru

wysyłał odczytaną wartość do komputera, gdzie napisany przez nas program dodatkowy odczytywał jej wartość i zapisywał do pliku. Zgromadzone dane zostały wykesportowane z pliku do środowiska `Matlab`, gdzie odbywała się analiza pomiarów.

4 Wnioski

Projekt myszki zbudowanej na akcelerometrach nie był łatwym zadaniem. Stworzył wiele trudności, szczególnie jeśli chodzi o analizę wartości odczytanych z akcelerometrów. Jednak w jakiś sposób poradziliśmy sobie z tymi problemami i można sterować prędkościami kursora w dowolnym z kierunków. Natomiast są jeszcze małe problemy z poprawnym działaniem przycisków. Problem związany z analizą jest spowodowany szumami występującymi w sygnale nadanym przez akcelerometry. Zastosowany filtr dolnoprzepustowy jedynie tłumi szumy wewnętrzne akclelerometru związane z kondensatorem różnicowym i generatorem wewnętrznym.

Literatura

[1] Marek Wnuk, Marek Kabała, Moduł z mikrokontrolerem MC9S12A64 (lub MC9S12C32), Wrocław 2005

[2] Jan Kedzierski, Edgar Ostrowski, Proste metody obsługi akcelerometrów - zastosowanie w robotach mobilnych na przykładzie ADXL202 oraz MMA7260, Wrocław 2007

[3] Freescale, MMA7260Q: 1.5g - 6g Three Axis Low-g Micromachined Accelerometer

[4] Freescale, MC9S12DJ64 Device User Guide

[5] Freescale, ATD_10B8C Block User Guide

[6] Freescale, HCS12 Serial Communications Interface (SCI) Block Guide

[7] FTDI, FT232BL USB UART (USB - Serial) I.C.

[8] Linux Programmer's Manual