



Instytut Informatyki Politechniki Śląskiej  
Zespół Mikroinformatyki i Teorii Automatów  
Cyfrowych



Rok akademicki:

Rodzaj studiów\*: SSI/NSI/NSM

Przedmiot (Języki  
Asemblerowe/SMiW):

Grupa

Sekcja

**2016/2017**

**SSI**

**SMiW**

**2**

**4**

Imię:

Mateusz

Prowadzący:

OA/JP/KT/GD/BSz/GB

Nazwisko:

Chudy

**GB**

## ***Raport końcowy***

Temat projektu:

**Przenośna konsola gier**

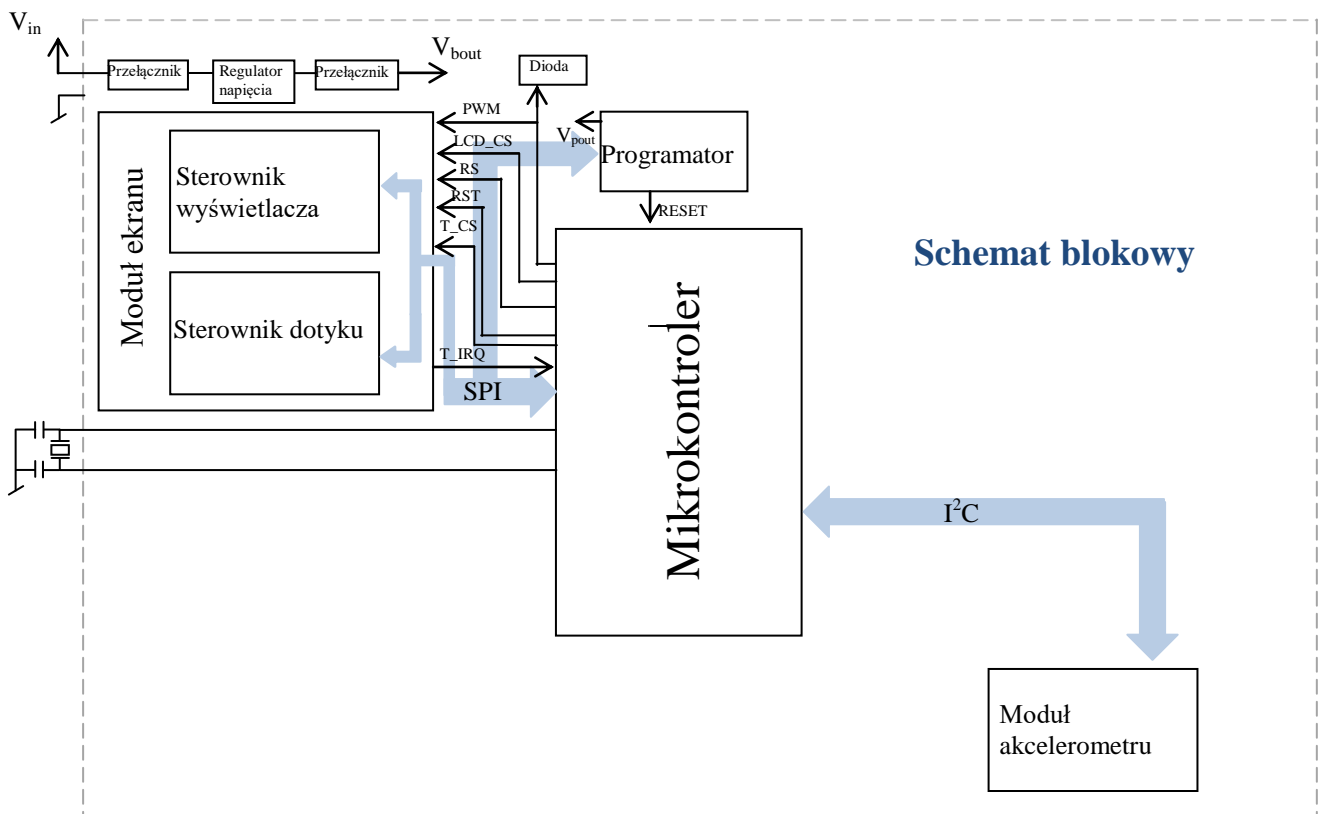
Data oddania:  
dd/mm/rrrr

**26.02.2017**

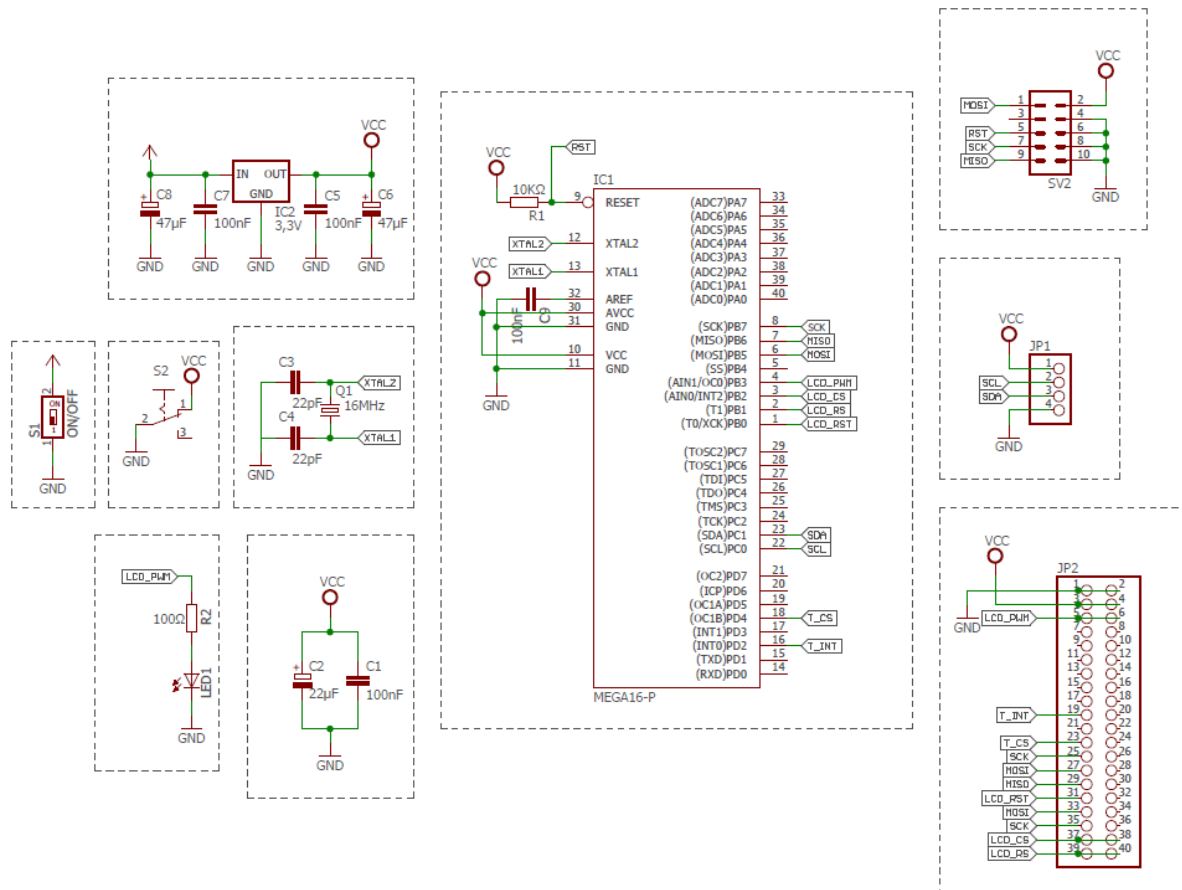
# Założenia

- Sterowanie:
  - Przełącznik: włączenie / wyłączenie urządzenia
  - Ekran dotykowy: sterowanie interfejsem użytkownika
  - Akcelerometr: sterowanie grą
- Wyświetlanie:
  - Kolorowy wyświetlacz LCD TFT: wyświetlanie interfejsu użytkownika i gry
  - Dioda LED: sygnalizacja zasilania urządzenia
- Zasilanie:
  - 3 baterie AA
- Programowanie:
  - Możliwość programowania na płytce za pomocą programatora USBasp

## Specyfikacja wewnętrzna

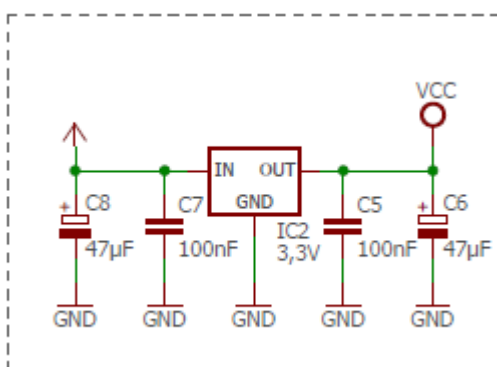


## Schemat ideowy



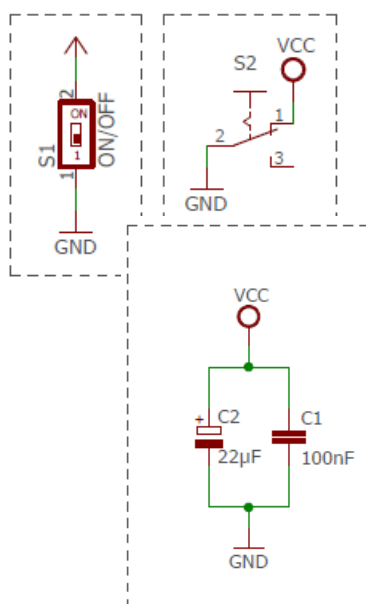
## Poszczególne bloki układu

### Zasilanie



W module zasilania został użyty regulator napięcia LIS35DE, pozwalający na mniejszy spadek napięcia w stosunku wejścia a wyjścia, tzn. jedyne 0,45V by na wyjściu osiągnąć 3,3V. Układ jest zasilany z 3 baterii AA. Zgodnie z założeniami dokumentacji pomiędzy masą a napięciem wejściowym został dołączony kondensator ceramiczny 100nF. Pomiedzy masą a napięciem wyjściowym również dołączono kondensator ceramiczny 100nF ze względu na zwiększenie stabilności napięcia i wcześniejsze testy z regulatorem

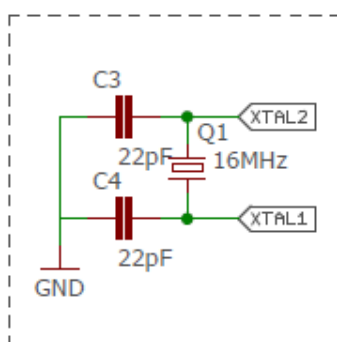
napięcia L8733. Dodatkowo pomiędzy masą i napięciem wejściowym, oraz masą i napięciem wyjściowym zostały dołączone równolegle kondensatory elektrolityczne o pojemności 47µF, magazynujące energię i zapewniające polaryzację.



Na kablu doprowadzającym zasilanie wejściowe do regulatora zastosowano przełącznik kołyskowy, pozwalający włączyć lub wyłączyć dopływ prądu do regulatora. Na ścieżce zasilania wyjściowego zastosowano przełącznik przesuwny, który pozwala na włączenie lub wyłączenie urządzenia.

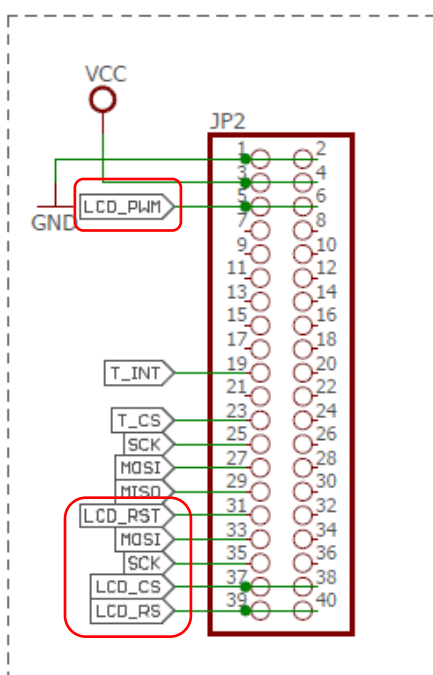
Do filtracji zasilania zostały użyte dwa kondensatory połączone równolegle, umieszczone tuż przy wejściach zasilania i masy mikrokontrolera. Jeden elektrolityczny 22µF oraz drugi ceramiczny 100nF pozwalających na zmniejszenie szumów napięcia dochodzącego zarówno z regulatora napięcia lub programatora w zależności od trybu pracy urządzenia.

### Zewnętrzny oscylator



W urządzeniu został użyty zewnętrzny oscylator RC 16MHz, pozwalający na szybszą pracę procesora. Również dzięki zwiększeniu taktowania procesora uzyskujemy znaczny wzrost prędkości na magistrali SPI, bez używania trybu double speed, z którym sterownik ekranu oraz sterownik dotyku nie współpracują. Dzięki takiemu rozwiązaniu, sterowniki te pracują w częstotliwości  $f_{osc}/4$  czyli 4MHz przy mniejszym procentowo czasie obsługi przez procesor samego interfejsu SPI, w stosunku do ilości czasu poświęconego na wszystkie operację w dowolnej jednostce czasu.

### Wyświetlanie



Do komunikacji z użytkownikiem zastosowano moduł ekranu dotykowego. Moduł ten dysponuje sterownikiem wyświetlacza BD663474 działającego w oparciu o interfejs SPI. Posiada on wewnętrzną pamięć GRAM, rejestr indeksowy, rejestr sterujący (statusu) oraz rejestr danych. Komunikacja odbywa się za pomocą magistrali SPI z wykluczeniem linii MISO, ze względu na możliwy jedynie zapis danych do pamięci wyświetlacza. BD663474 został zrealizowany w oparciu o programowalny układ CPLD pracujący na magistrali 8-bitowej (80-series). Po odebraniu szeregowo wysłanego bajtu danych przez sterownik, zostaje on wysłany równolegle w trybie big-endien wewnętrzna magistrale.

Wyróżnia się 4 rodzaje wewnętrznych operacji:

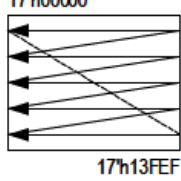
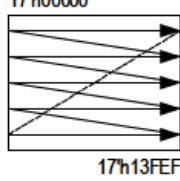
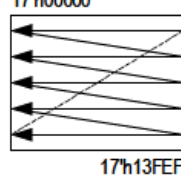
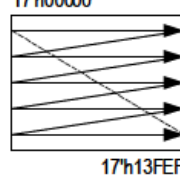
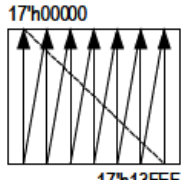
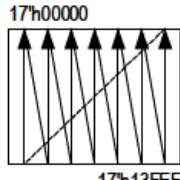
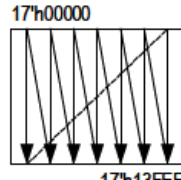
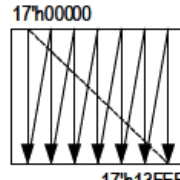
- 1) zapis danych do rejestru indeksowego,
- 2) odczyt wewnętrznego statusu,
- 3) zapis danych do rejestru sterującego lub do GRAM za pośrednictwem rejestru danych,
- 4) odczyt danych z GRAM za pośrednictwem rejestru danych.

Sterowanie wyświetlaczem z zewnątrz odbywa się w następujący sposób:

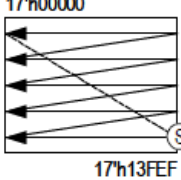
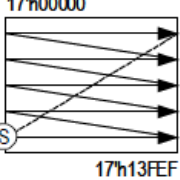
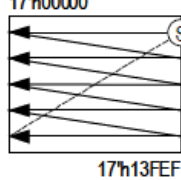
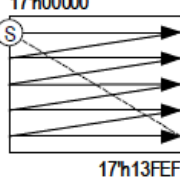
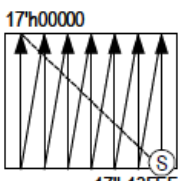
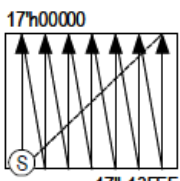
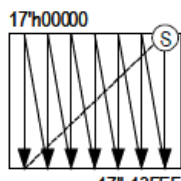
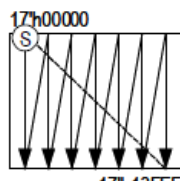
- 1) Inicjalizacja: start oscylacji, ustawienia zasilania, ustawienia wyświetlania, ustawienia LTPS, ustawienia dostępu dla okna, ustawienia skali szarości, ustawienia osi.
- 2) Wysyłanie danych do wyświetlenia: ustawienie punktu początkowego dostępu do GRAMu (współrzędne punktu początku obszaru), ustawienie punktu końcowego dostępu do GRAMu (współrzędne punktu końca obszaru), ustawienie indeksu wpisu do GRAMu, wysłanie koloru dla kolejnego punktu.
- 3) Rysowanie: po punktach w obszarze rysowania iterujemy w ustawiony przez bity ORG, I/D1, I/D2 i AM znajdujące się w rejestrze pod indeksem 003h (entry mode):

#### Entry Mode 1 (R003h)

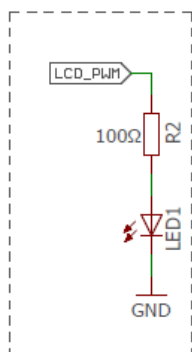
R/W	RS	IB15	IB14	IB13	IB12	IB11	IB10	IB9	IB8	IB7	IB6	IB5	IB4	IB3	IB2	IB1	IB0
W	1	TRI	DFM	0	BGR	0	0	HWM	0	ORG	0	I/D [1]	I/D [0]	AM	0	0	0

ORG=0	I/D[1:0] = "00" H-direction: Decrement V-direction: Decrement	I/D[1:0] = "01" H-direction: Increment V-direction: Decrement	I/D[1:0] = "10" H-direction: Decrement V-direction: Increment	I/D[1:0] = "11" H-direction: Increment V-direction: Increment
AM = "0" Horizontal direction				
AM = "1" Vertical direction				

Note: Writing is available only to the GRAM within the window address when setting the window address.

ORG=1	I/D[1:0] = "00" H-direction: Decrement V-direction: Decrement	I/D[1:0] = "01" H-direction: Increment V-direction: Decrement	I/D[1:0] = "10" H-direction: Decrement V-direction: Increment	I/D[1:0] = "11" H-direction: Increment V-direction: Increment
AM = "0" Horizontal direction				
AM = "1" Vertical direction				

Note: When ORG = 1, writing to the RAM starts from the address at the corner (S) in the area.

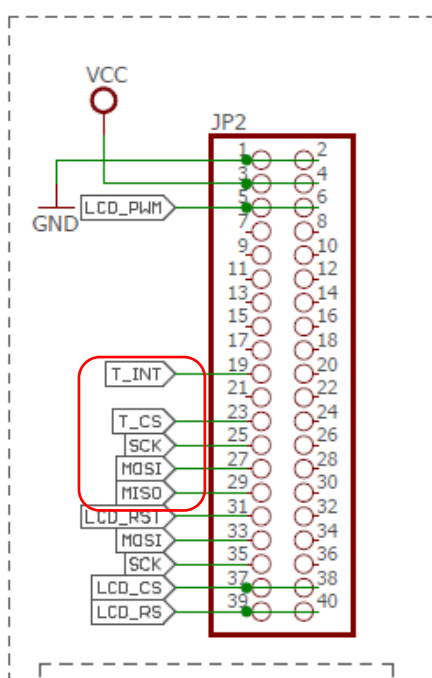


Podświetlanie ekranu jest realizowane przez pin LCD\_PWM. Ekran reaguje na jego stan „0” podświetleniem, a „1” zgaszeniem podświetlania. Ze względu na oszczędność energii na pinie PB3 (OC0) generowany jest sygnał PWM w trybie Fast PWM, o stałej częstotliwości:

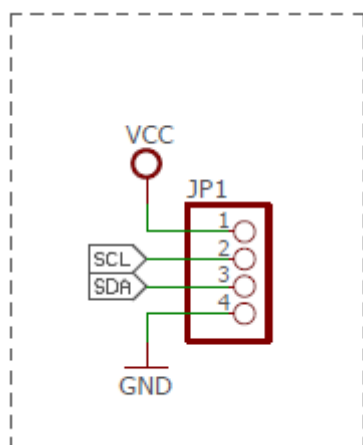
$$f_{OCnPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

, gdzie N to współczynnik preskalera. Współczynnik wypełnienia został ustawiony na 50%. Poprzez sygnał z pinu PB3 poprowadzono również przez opornik do masy diodę czerwoną LED, sygnalizującą uruchomienie urządzenia.

## Sterowanie

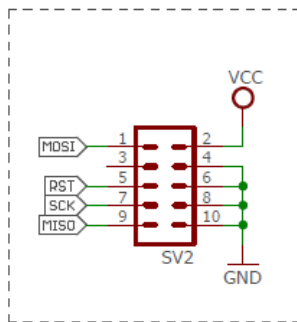


Moduł ekranu dotykowego posiada również sterownik dotyku XPT2046, także pracujący w oparciu o interfejs SPI. Działa on na zasadzie badania zmian rezystancji, wywołanych przez wywieranie ciśnienia na powierzchnię ekranu, poprzez przetwornik analogowo-cyfrowy z kompensacją wagową (SAR) z 12-bitowym próbkowaniem 125 kHz. By wykryć współrzędne dotyku, wykonują się dwie konwersje A/D odpowiednio dla osi x i y. Dla wysłanej 8-bitowej komendy, ekran zwraca 16-bitów reprezentujących analogową wartość zmierzona dla jednej osi, której dotyczyła komenda. Po wykonaniu konwersji i korekcy błędów wynoszącej -20 pikseli na osi x i -2 piksele na osi y, otrzymujemy przybliżone współrzędne punktu dotknięcia w ekran. Sterownik posiada również funkcję pomiaru temperatury i poboru baterii.



Kolejnym modulem poświęconym sterowaniu jest moduł akcelerometru wyposażony w 3-osiowy akcelerometr LIS35DE. Moduł jest wyposażony w interfejs I<sup>2</sup>C z wbudowanymi rezystorami podciągającymi. Komunikacja odbywa się poprzez wysłanie komendy wpisu, następnie podanie indeksu rejestru statusu, zawierającego liczbę proporcjonalną do kąta nachylecha do danej osi, oraz wysłania komendy odczytu. Po odpowiednim skorygowaniu w stosunku do punktu (0, 0) otrzymujemy współczynnik kąta nachylenia do poszczególnych osi.

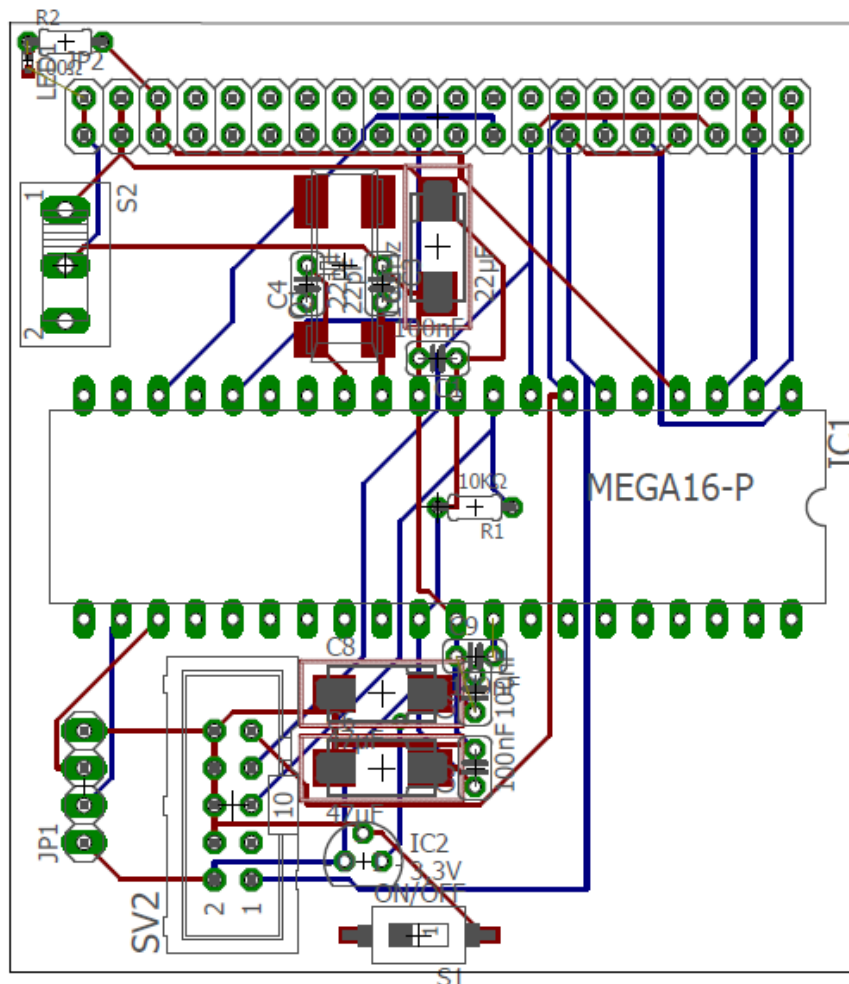
# Programowanie



Na płytce zamontowano złącze KANDA pozwalające na podłączenie programatora zgodnego z USBasp ISP. Przed podłączeniem programatora należy wyłączyć oba przełączniki, by nie uszkodzić programatora i regulatora napięcia, ponieważ przy włączonych może dojść do sytuacji, że do układu podłączone są 2 różne źródła zasilania.

Wszystkie zastosowane elementy elektroniczne zostały wybrane ze względu na zakres postawionych wymagań funkcjonalnych urządzenia, stosunek ceny do funkcjonalności elementu oraz dostępności w domowym warsztacie.

## Schemat montażowy



Ze względu na plany rozbudowy urządzenia nie zastosowano obudowy.

## Lista elementów

- Mikrokontroler Atmega16A PU firmy Atmel.
- 2,2 calowy ekran dotykowy o rozdzielczości 320x240 firmy Waveshare.
  - Sterownik ekranu: BD663474 Hitachi Displays (240RGB × 320-Dot 1-Chip Driver IC with Built-in RAM for 262,144-Color TFT-LCD Panel).
  - Sterownik dotyku: XPT2046 Shenzhen Xptek (Touch Screen Controller).
- Moduł 3-osiowego akcelerometru z wyjściem cyfrowym KAmoMEMS2 firmy Kamami.
  - Czujnik ruchu: LIS35DE STMicroelectronics (3-axis -  $\pm 2g/\pm 8g$  smart digital output “piccolo” accelerometer).
- Regulator napięcia 3,3V LF33CV STMicroelectronics (Very low drop voltage regulator with inhibit function).
- Złącza goldpin:
  - 1x8 pin
  - 1x4 pin
  - 2x20 pin
- Gniazdo KANDA 10 pin
- Podstawka do układów DIP 40 pin
- Przełączniki:
  - Kołyskowy
  - Przesuwany
- Czerwona dioda LED
- Rezonator kwarcowy 16MHz
- 3 baterie AA
- Kondensatory ceramiczne:
  - 4 x 100nF
  - 2 x 22pF
- Kondensatory elektrolityczne:
  - 2 x 47 $\mu$ F
  - 1 x 22 $\mu$ F
- Oporniki:
  - 1 x 10K $\Omega$
  - 1 x 100 $\Omega$

## Algorytm oprogramowania urządzenia

Po włączeniu zasilania:

- Inicjalizacja kolejno: interfejs I<sup>2</sup>C, akcelerometr, sterownik dotyku, interfejs SPI, sterownik wyświetlacza
- Wyświetlenie ekranu powitalnego
- Wyświetlenie pulpitu z przyciskiem menu
- Oczekiwanie w pętli nieskończonej na zmianę flagi przez przerwanie ze sterownika dotyku.
- Po wybraniu najniższej położonych opcji (liści menu) przerwania w kontekście menu są ignorowane, a wykonywana jest gra bądź aplikacja.



## Opis zmiennych

<code>TouchPoint touchPoint</code>	Globalny punkt dotknięcia
------------------------------------	---------------------------

<code>volatile uint8_t interruptFlag</code>	Globalna flaga wskazująca na dotknięcie w ekran
---	---

## Opis funkcji

<code>ISR(INT0_vect)</code>
Procedura obsługi przerwania dotknięcia w ekran

<code>void TouchInit(void)</code>
Inicjalizuje sterownik dotyku

<code>uint16_t TouchGetADC(uint8_t cmdCode)</code>
Pobiera ze sterownika pojedynczy wynik ADC

<code>uint16_t TouchGetADCEx(uint8_t cmdCode)</code>
Pobiera ze sterownika kilka razy wynik ADC i filtruje zwracaną wartość

<code>ExecuteResult TouchReadADS(uint16_t* xad, uint16_t* yad)</code>
Pobiera ze sterownika przefiltrowane wyniki konwersji AD dla osi x i y

<code>ExecuteResult TouchReadADS2(uint32_t* xad, uint32_t* yad)</code>
Dwukrotnie pobiera ze sterownika przefiltrowane wyniki konwersji AD dla osi x i y

```
void TouchConvertAdToXy(void)
```

Konwertuje wartości odczytanych ADC na wartości współrzędnych

```
ExecuteResult TouchReadOnce(void)
```

Pobiera jednokrotnie współrzędne dotknięcia

```
ExecuteResult TouchReadContinue(void)
```

Pobiera współrzędne dotknięcia w sposób ciągły

```
void AccelerometerInit(void)
```

Inicjalizuje akcelerometr

```
int8_t AccelerometerGetAngle(Axis axis)
```

Pobiera wartość proporcjonalna do kąta nachylenia do danej osi

```
Direction AccelerometerGetDirection(void)
```

Pobiera kierunek nachylenia

```
void LCDPWMInit(void)
```

Inicjalizuje i włącza sygnał PWM

```
void LCDWriteIndex(uint16_t index)
```

Wpisuje do sterownika indeks rejestru

```
void LCDWriteReg(uint16_t index, uint16_t data)
```

Wpisuje dane do rejestru

<code>void LCDWriteData(uint16_t data)</code>
Wpisuje dane

<code>void LCDInit(void)</code>
Inicjalizuje sterownik wyświetlacza

## Interakcja z oprogramowaniem

Oprogramowanie posiada 3 rodzaje dostępnych funkcjonalności:

- **Menu**

W przypadku menu, interakcja z układem elektronicznym zachodzi na zasadzie przerwań generowanych przez sterownik dotyku, co powoduje ustawienie zmiennej globalnej wskazującej, że ekran został dotknięty. Sterownik ten współpracuje z mikrokontrolerem przy użyciu interfejsu SPI, co pozwala na wysyłanie komend i odbiór danych aktualizujących globalną strukturę z współrzędnymi dotyku. W głównej pętli programu znajdującej się w funkcji main, taka zmiana zostaje wykryta i następuje przejście do procedury sprawdzającej czy nowe współrzędne punktu, zawierają się obszarze na ekranie przedstawiającym przycisk z opcją menu. Gdy dotknęliśmy przycisk następuje wyświetlenie nowego menu, uruchomienie gry lub aplikacji.

- **Gry**

Sterowanie grami odbywa się za pośrednictwem akcelerometru wykorzystującego interfejs I<sup>2</sup>C. Podczas rozgrywki zostają wysyłane komendy żądania udostępnienia przez akcelerometr wartości, przechowywanych w jego wewnętrznych rejestrach. Wartości te są proporcjonalne do kąta nachylenia do poszczególnych osi w zależności od rejestru. Po otrzymaniu ich oprogramowanie interpretuje je i powoduje zmianę kierunku ruchu.

- **Aplikacje**

Podobnie jak w przypadku menu sterowanie aplikacjami odbywa się za pośrednictwem sterownika dotyku, lecz bez wykorzystania przerwań. Badanie zmian współrzędnych odbywa się w sposób ciągły w pętli while kończącej się w przypadku wybraniu opcji wyjścia.

Kompletna dokumentacja oprogramowania została zamieszczona w pliku Dokumentacja\_oprogramowania.pdf, znajdującym się na serwerze GIT ZMiTAC.

# Specyfikacja zewnętrzna

## Opis funkcji elementów sterujących

- **Akcelerometr**

W zaimplementowanych grach poruszając urządzeniem można sterować pewnymi elementami gry. W przypadku gry snake badane jest odchylenie urządzenia dla osi x oraz y od poziomu, powodując zmianę kierunku poruszania się węża. Natomiast w przypadku gry ping pong badane jest tylko odchylenie na osi x.

- **Ekran dotykowy**

Podczas poruszania się po menu lub pulpicie, można wybierać poszczególne opcje za pomocą przycisków wybieranych dotknięciem w nie. Po wybraniu opcji, następuje przejście do kolejnego menu, gry bądź aplikacji. Podczas działania aplikacji, dzięki ekranowi dotykowemu istnieje również możliwość wybierania opcji związanych z uruchomioną aplikacją. W przypadku aplikacji paint są to zmiana koloru rysowania oraz wyjście z niej.

## Opis funkcji elementów wykonawczych

- **Kolorowy ekran o rozdzielczości 240x320 pikseli**

Wszystkie aplikacje i gry, jak również pulpit i menu są wyświetlane na kolorowym wyświetlaczu dotykowym. Dzięki rozdzielczości ekranu posługiwanie się funkcjonalnością urządzenia jest intuicyjne i proste.

- **Czerwona dioda LED**

Dioda pełni zadanie sygnalizacji włączonego zasilania, oraz kontroli nad częstotliwością odświeżania ekranu. Dzięki jej jasności można w sposób szacunkowy określić, czy ekran odświeżany jest większą, bądź mniejszą częstotliwością, co przekłada się z kolei na pobór prądu przez urządzenie.

## Opis reakcji oprogramowania na zdarzenia zewnętrzne

- **Pulpit**

Podczas wyświetlania pulpitu, wyświetlany jest również komponent w postaci przycisku menu. Gdy użytkownik dotknie ekranu w miejscu gdzie znajduje się przycisk, nastąpi przejście do menu głównego.

- **Menu**

W menu głównym wyświetlają się przyciski z opcjami przejścia do kolejnych menu związanych z aplikacjami lub grami.

- **Aplikacje**

W menu aplikacji dostępna jest aplikacja paint, uruchamiana po naciśnięciu przycisku.

- **Gry**

W menu gier dostępne są do wyboru gry takie jak snake, czy ping pong, również uruchamiane po naciśnięciu wybranego przycisku.

- **Paint**

Podczas działania aplikacji paint, można rysować na ekranie dotykowym, oraz zmieniać kolor rysowania wybierając jeden z ośmiu kwadratów w danym kolorze na górnym pasku aplikacji. Wyjście z aplikacji odbywa się poprzez wybranie przycisku exit w lewym dolnym rogu ekranu.

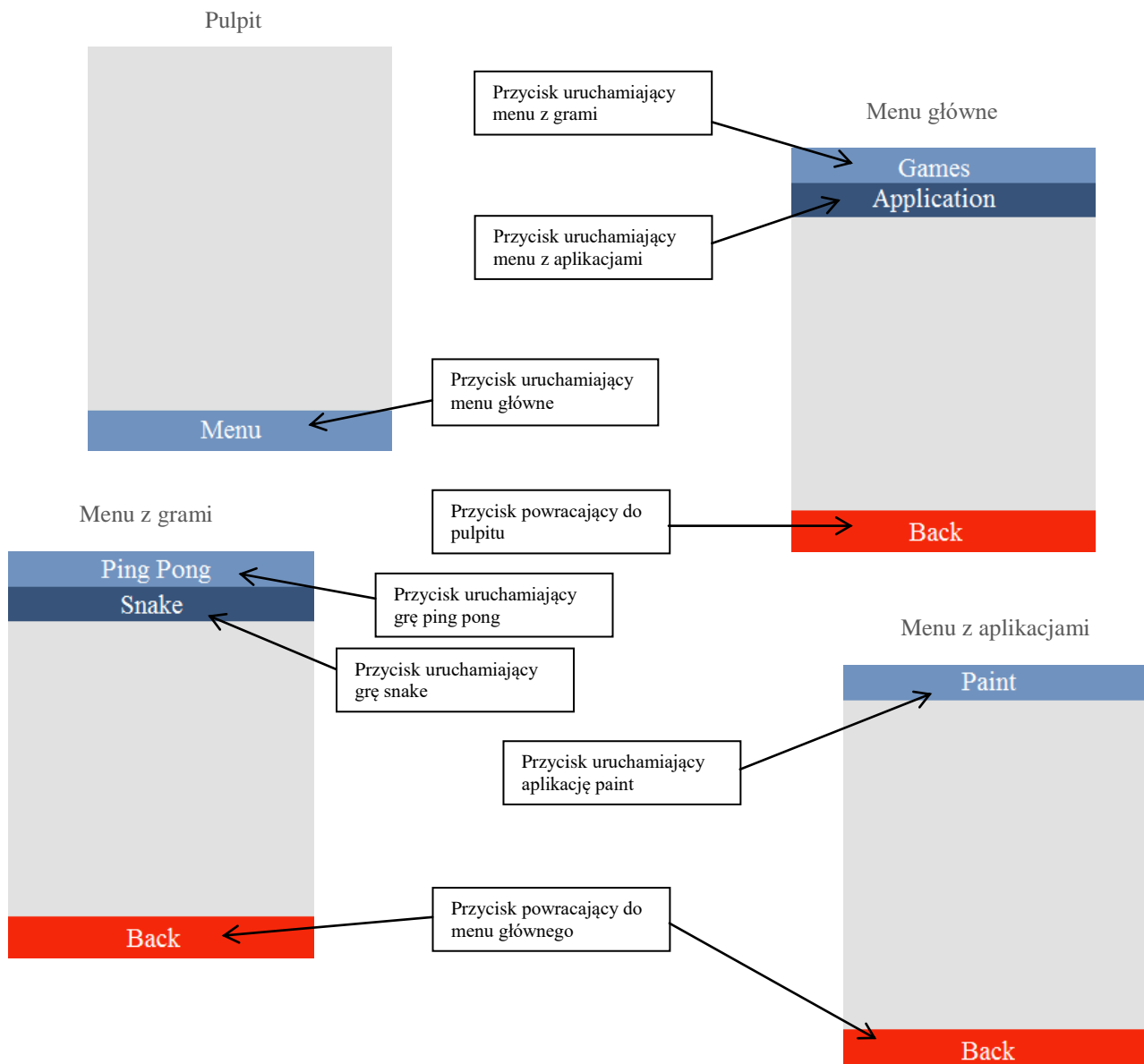
- **Snake**

Sterowanie rozgrywką odbywa się poprzez nachylenie urządzenia na osiach x i y w stosunku do poziomu. Powoduje to zmianę kierunku węża w tym samym kierunku w którym urządzenie zostało pochylone.

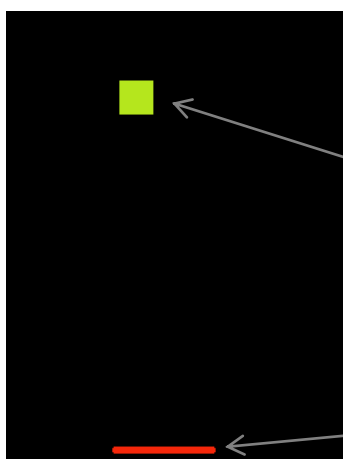
- **Ping Pong**

Sterowanie grą, również opiera się o mechanizm pochylania urządzenia, lecz w przypadku tej gry, reaguje ona tylko na przechylenia na osi x w stosunku do poziomu. Powoduje to przesunięcie kreski będącej paletką, odbijającej kwadrat reprezentujący piłkę.

## Instrukcja obsługi urządzenia



Gra ping pong



Gra ping pong polega na odbijaniu piłki paletką, poruszaną ruchem urządzenia. Gdy piłka upadnie poniżej linii paletki, nie trafiając w nią, rozgrywka kończy się i następuje przejście do pytania o ponowną grę.

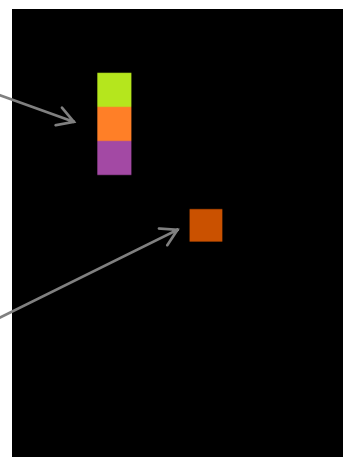
Samoczynnie poruszający się kwadrat reprezentujący piłkę

Ruchoma kreska reprezentująca paletkę

Poruszające się kwadraty będące

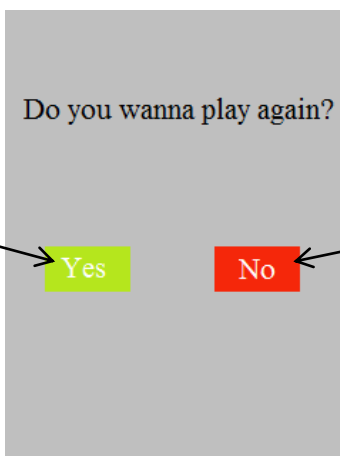
Kwadrat reprezentujący pokarm

Gra snake



Gra snake polega na zbieraniu pokarmu węzem, poruszonym ruchem urządzenia. Po zebraniu pokarmu, element zostaje dołączony na początek węża, co powoduje zwiększenie jego rozmiaru o jeden. Rozgrywka kończy się w przypadku uderzenia pierwszym elementem węża w któryś z jego pozostałych elementów, powodując uruchomienie ekranu pytania o ponowną grę.

Ekran pytania o ponowną grę



Odpowiedź pozytywna, uruchamiająca ponowną rozgrywkę

Odpowiedź negatywna, powodująca powrót do menu z grami

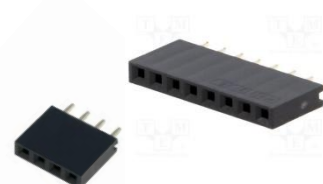
## Opis złączy



Złącze ekranu dotykowego. Ekran powinien zostać podłączony w taki sposób, że dioda na płytce powinna zostać odsłonięta, a ekran powinien zachodzić na mikrokontroler.



Złącze programatora w standardzie KANDA. By podłączyć programator, należy wcześniej wyciągnąć z płytki akcelerometr, by odsłonić dostęp do złączy.



Złącze akcelerometru składa się z dwóch złączy goldpin. Zgodnie z wyjściami na płytce modułu akcelerometru, powinny zostać one wsadzone odpowiednio w złącze 4 i 8 pinowe.

# Opis montażu układu, sposobu uruchamiania oraz testowania

Układ w pierwotnej fazie projektowania został zmontowany na płytce stykowej, z wyłączeniem modułu zasilania. Gdy zaimplementowano i przetestowano wszystkie sterowniki urządzeń zewnętrznych, układ został przeniesiony fragmentami na płytkę uniwersalną. Na początku rozplanowano ułożenie podkładki pod mikrokontroler, złączy akcelerometru, wyświetlacza dotykowego, oraz programatora. Po przylutowaniu złączy, nastąpiło połączenie odpowiednich pinów dzięki przylutowaniu kabli kynarowych. Następnie przylutowano wszystkie kondensatory, oporniki, diody oraz rezonator kwarcowy. Również na płytce uniwersalnej w pierwszych fazach realizacji projektu pominięto zasilanie zewnętrzne i korzystano z zasilania z programatora. Gdy wszystko zostało przetestowane z udziałem zewnętrznego zasilania, na płytce uniwersalnej zrealizowano moduł zasilania bateryjnego z przełącznikiem kołyskowym.

## Wnioski i uwagi z przebiegu pracy

Montaż układu nastąpił bez żadnych problemów, lecz podczas uruchamiania z zasilania bateryjnego ekran dotykowy nie chciał się zainicjalizować. Również sam akcelerometr nie działał poprawnie. Rozwiązaniem problemu okazała się wymiana stabilizatora napięcia z użytego pierwotnie L78L33 firmy STMicroelectronics na LF33CV tej samej firmy. Problem polegał na zbyt małym napięciu na wejściu stabilizatora, które dla 3 baterii AA wynosiło około 4,5V. Stabilizator ten wymagał o 2V większego napięcia na wejściu by na wyjściu otrzymać stabilne 3,3V. Zamiennie zastosowany stabilizator wymaga jedynie 0,4V więcej, co pozwala na prawidłowe zasilanie całego układu. Dodatkowo do układu dołączono przełącznik przesuwany, rozwierający wyjście zasilania stabilizatora i szynę zasilania układu, zapobiegając przy tym możliwości przyłożenia napięcia na wyjście stabilizatora w przypadku podłączenia programatora.

Testowanie urządzenia odbywało się stopniowo, testując funkcjonalności poszczególnych modułów zewnętrznych. Testując dany moduł uruchamiano coraz to bardziej rozbudowane funkcje, obserwując czy wszystko zachowuje się poprawnie. Przetestowano: Odczytywanie wartości proporcjonalnej do kąta nachylenia odczytywanej z akcelerometru, wyświetlanie prymitywów graficznych na ekranie, generowanie zewnętrznych przerwań przez sterownik dotyku, oraz odczyt i konwersję ADC współrzędnych dotknięcia ekranu. Po przeprowadzonych testach moduły zewnętrzne zostały skalibrowane na potrzeby działających gier i aplikacji.

## Wnioski końcowe

Dzięki realizacji projektu zostały przeciwiczone w praktyce umiejętności nabyte w czasie wykładów, ćwiczeń i laboratoriów systemów mikroprocesorowych i wbudowanych. Również nabyto cenną wiedzę z poza zakresu zajęć prowadzonych w ramach przedmiotu, min. działania sterowników wyświetlaczy kolorowych z pamięcią GRAM, zasad działania akcelerometrów elektronicznych, oraz zasad działania sterowania dotykowego, opartego o wykrywanie nacisku. Podsumowując realizacja projektu pozwala zdobyć bardzo cenne, praktyczne umiejętności konstruowania układów elektronicznych.

# Literatura

- XPT2046 Data Sheet, 2007.6, SHENZHEN XPTEK TECHNOLOGY
- BD663474240RGB × 320-Dot 1-Chip Driver IC with Built-in RAM for 262,144-Color TFT-LCD Panel, Rev. 0.032005.12.26, Hitachi Displays
- 2.2inch 320x240 Touch LCD (A) User Manual, Waveshare
- 8-bit Microcontroller with 16K Bytes In-System Programmable Flash ATmega16 ATmega16L, Rev. 2466T-AVR-07/10, ATMEL
- KAmoMEMS2 Moduł 3-osiowego akcelerometru z wyjściem cyfrowym, ver. 1.0, KAMAMI
- LIS35DE, MEMS motion sensor 3-axis -  $\pm 2g/\pm 8g$  smart digital output “piccolo” accelerometer, April 2009 Doc ID 15594 Rev 1, STMicroelectronics
- LFX19, Very low drop voltage regulator with inhibit function, January 2017 Doc ID 2574 Rev 30, STMicroelectronics