UNIWERSYTET GDAŃSKI
Wydział Matematyki, Fizyki i Informatyki

Daniel Sienkiewicz
nr albumu: 206358

Projekt komputera samochodowego bazujący na systemie mikrokomputera Intel Galileo

Praca magisterska na kierunku:
INFORMATYKA
Promotor:
dr inż. Janusz Młodzianowski

Gdańsk 2015

Streszczenie

Celem pracy jest stworzenie komputera pokładowego do samochodu, w którego skład wchodzi:

1. Mikrokomputer Intel Galileo Gen 1,
2. Ekran dotykowy FTDI VM800,
3. Oprogramowanie,
4. Kamerka cofania.

Pierwsza część pracy przedstawia architekturę projektu wraz z jego opisem funkcjonalnym. Opisuję również mechanizmy komunikacji systemu mikroprocesorowego z otoczeniem.

Następnie przedstawiony jest pomysł implementacji oraz proces tworzenia niezbędnej do obsługi symulatora samochodu biblioteki pozwalającej na komunikację z I/O Expanderem PCF8574N.

Ostatnia część pracy przedstawia pomysły możliwych rozszerzeń projektu o dodatkowe moduły oraz funkcjonalności w zależności od potrzeb użytkownika.

Słowa kluczowe

Intel Galileo, I^2C, SPI, C, Arduino, GPIO, FTDI Chip, VM800

Spis treści

ROZDZIAŁ 1

Wprowadzenie

1.1. Cele

Celem pracy jest budowa oraz oprogramowanie komputera pokładowego do samochodu. Komputer powinien móc wczytać z czujników temperaturę panującą w silniku, na zewnątrz oraz w środku samochodu. Ponadto powinien on móc zapisać aktualną pozycję GPS na karcie microSD oraz umożliwić korzystanie z kamerki cofania lub inteligentnego lusterka wstecznego.

1.2. Założenia

Do wykonania komputera wykorzystano: Intel Galileo używane w trybie Arduino o oprogramowywane za pomocą Arduino IDE, lokalizator GPS służący do podawania aktualnej pozycji dzięki której obliczana zostaje droga przebyta przez samochód, kamerka internetowa służąca jako czujnik cofania oraz inteligentne lusterko wsteczne oraz symulator samochodu. Aktualnie komputer nie będzie zamontowany do fizycznego samochodu więc do tych celów zbudowany został symulator samochodu składający się z podstawowych czujników takich jak: guziki służące za czujnik zapięcia pasów/zamknięcia drzwi, potencjometry służące za czujniki temperatury oraz I/O expander PCF8574N pozwalający zmniejszyć ilość kabli wychodzących z symulatora do Intel Galileo do 2 zamiast 11. Na komputerze nie będzie wyświetlana aktualna prędkość ani przebieg ponieważ nawet w najnowszych samochodach nie jest to dostępna opcja. Dane te są dostępne na zegarach samochodowych więc nie ma potrzeby powtarzania tej informacji.

1.3. Plan pracy

TO DO

ROZDZIAŁ 2

Architektura

2.0.1. Opis funkcjonalny

2.0.2. Mechanizmy komunikacji systemu mikroprocesorowego z otoczeniem

2.0.2.1. Porty

Porty są jednym z najbardziej podstawowych interfejsów. Najczęściej dzieli się je na porty:

1. Cyfrowe
2. Analogowe

Porty cyfrowe charakteryzują się możliwością przyjęcia lub wysłania sygnału binarnego (1 - jest sygnał, 0 - sygnału nie ma). Z kolei porty analogowe mogą przesyłać sygnały nawet 10 bitowe. Każdy z portów może działać w jednym z dwóch trybów: wejścia - oczekiwać na przyjęcie danych od urządzenia zewnętrznego oraz wyjścia - wysyłać dane do urządzenia zewnętrznego.

W środowisku Arduino aby obsłużyć port analogowy wystarczy:

Listing 2.1. Obsługa portu analogowego w środowisku Arduino

```
int val = 0;
int analogPin = A1;
pinMode(analogPin, OUTPUT);
val = analogRead(analogPin);
pinMode(analogPin, INPUT);
analogWrite(ledPin, val);
```

i odpowiednio dla portu cyfrowego:

Listing 2.2. Obsługa portu cyfrowego w środowisku Arduino

```
int val = 0;
int digitalPin = 1;
pinMode(digitalgPin, OUTPUT);
val = digitalRead(analogPin);
pinMode(digitalPin, INPUT);
digitalWrite(digitalPin, HIGH);
```

2.0.2.2. Przerwania

Przerwania są to bezpośrednie funkcje systemu lub sprzętu ułatwiające komunikację ze światem zewnętrznym. Część z nich jest zarezerwowana przez system lecz część z nich jest wolna do wykorzystania dla programisty. Przerwania możemy podzielić na trzy podstawowe rodzaje:

1. Programowe
2. Sprzętowe
   (a) Maskowalne (NMI)
   (b) Niemaskowalne (INTR)
3. Wyjątek

Przerwania programowe wywołuje się za pomocą komendy INT XX gdzie XX oznacza numer przerwania zadeklarowanego w tablicy wektorów przerwań, która jest tworzona przy każdorazowym starcie systemu. Przerwanie to może przyjąć wartości do 255 i są one zarezerwowane przez procesor oraz użytkownika.

Przerwanie sprzętowe jest to rodzaj przerwań wywoływanych przez urządzenia wejścia/wyjścia lub zgłaszane przez procesor. Zostają one wywołane

niezależnie w określonych przypadkach. Przerwania te dzielimy na maskowalne oraz niemaskowalne. Główna różnica między nimi polega na możliwości zablokowania przerwań maskowalnych podczas gdy przerwania niemaskowalne muszą zostać obsłużone. Przykładem przerwania niemaskowalnego INT2 czyli popularny blue screen of death.

Ostatnim rodzajem przerwań są wyjątki. Wywoływane sa podczas napotkania przez procesor błędów oraz niepowodzeń. Arduino oczywiście obsługuje przerwania. Ich obsłga jets bardzo prosta: W środowisku Arduino aby obsłużyć port analogowy wystarczy:

Listing 2.3. Obsługa przerwań sprzętowych w środowisku Arduino

```
attachInterrupt(pinInt, funcName, mode);
```

gdzie pinInt jest to pin na którym Arduino będzie nasłuchiwało na przerwanie, funcName jest to nazwa funkcji który zostanie wykonana gdy przerwanie zostanie zgłoszone, mode - jest to określenie kiedy sygnał może być uznany za przerwanie.

2.0.2.3. Odpytywanie w pętli

Jednym z najprostszych metod pozyskania danych z mikro kontrolera jest jego odpytywanie w nieskończonej pętli.Jest to najmniej efektywny sposób ponieważ cały czas zajmuje niepotrzebnie zasoby sprzętu niepotrzebnymi zapytaniami.

Listing 2.4. Odpytywanie w nieskończonej pętli w środowisku Arduino

```
void loop() {
        funcName();
        delay(1000);
}
```

2.0.2.4. Timer

Listing 2.5. Użycie timer w środowisku Arduino

```
#include <TimerOne.h>
Timer1.initialize(500000); // ustawienie śdugoci timera
Timer1.attachInterrupt(funcName, 500000);
```

2.0.2.5. Protokół komunikacyjny

TO DO

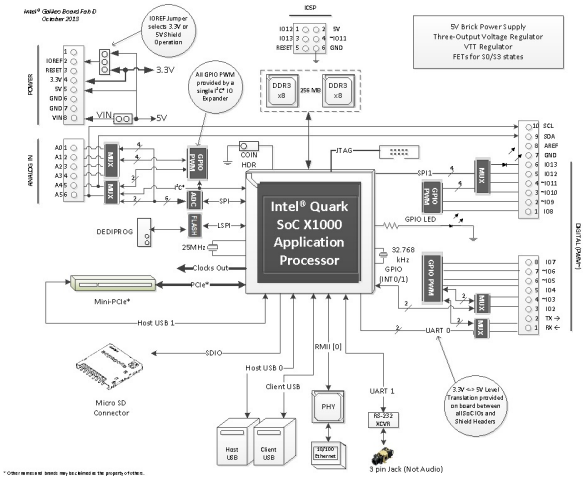
2.0.2.6. Złączki oraz kable

TO DO

2.0.3. Intel Galileo

Intel Galileo jest to mikro kontroler oparty na 32-bitowym procesorze Intel® Quark SoC X1000 i taktowaniu 400MHz. Został on wyposażony w 14 pinów cyfrowych (w tym 6 pinów mogących pełnić funkcję PWM) oraz 6 pinów cyfrowych.Każdy z tych pinów jest w stanie operować napięciem max 5V. Bardzo dużym atutem Galileo jest wbudowana karta sieciowa, port RS-232 oraz port USB oraz slot karty microSD. Galileo może być używane w dwóch trybach - trybie w pełni kompatybilnym z Arduino oraz w trybie z zainstalowanym systemem operacyjnym (np. Linux).

ŹRÓDŁO: https://www.arduino.cc/en/ArduinoCertified/IntelGalileo



Rysunek 2.1. Schemat logiczny układu Intel Galileo

Źródło: https://www.arduino.cc/en/ArduinoCertified/IntelGalileo

ROZDZIAŁ 3

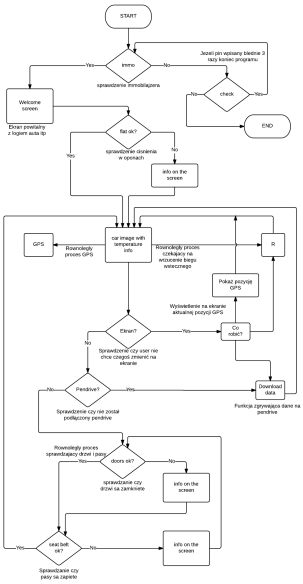
Implementacja

3.1. I^2C

3.1.1. Problemy z bibliotekami

3.1.2. moja implenatacja I^2C (read)

3.1.3. Schemat blokowy programu



Rysunek 3.1. Schemat blokowy głównego programu

Źródło: Opracowanie własne

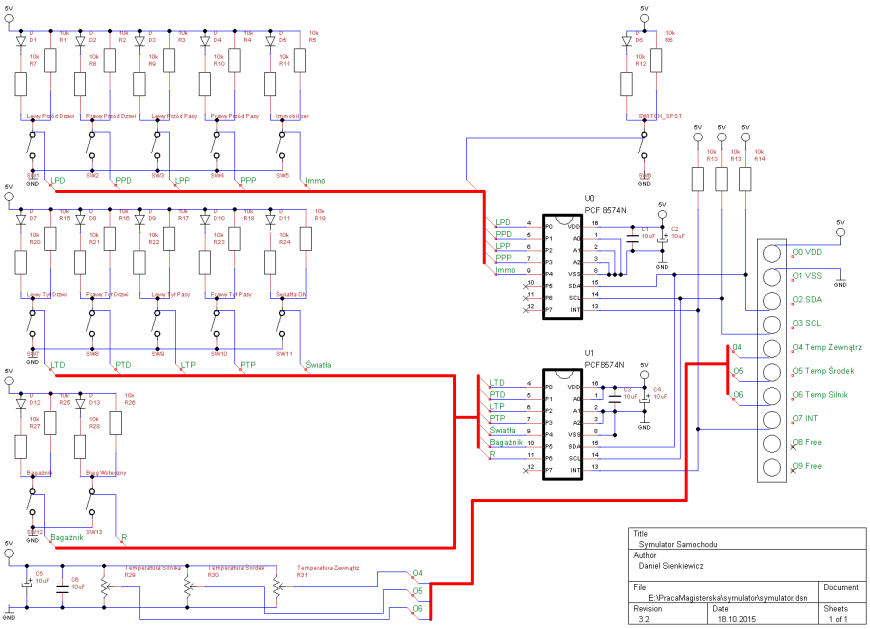
3.1.4. Moja biblioteka do R/W Arduino dla Intel Galileo

3.2. Założenia funkcjonalne

- czytanie z czujników, pisanie do ekranu, czytanie z ekranu - włączanie i wyłączanie systemu

3.3. Integracja z samochodem

- podpięcie pod auto - włączanie i wyłączanie systemu - można brutalnie wyłączyć



Rysunek 3.2. Schemat symulatora samochodu

Źródło: Opracowanie własne

3.4. VM800

- na poczatku emulacja na PC - potem przepisanie na niski poziom - ostatecznie podpiecie do Galielo (poszukac czy juz jest?)

programers manual reference vm800 ftdi POSZUKAĆ!!!!

3.5. Dalsze kroki oraz propozycje

- schemat blokowy z BAJERAMI i wybrane to co zrobię

Zakończenie

TO DO

DODATEK A

Karty Katalogowe

Katalog *datasheets* zawiera karty katalogowe użytych podzespołów

DODATEK B

Porównanie dostępnych na rynku mikro kontrolerów

	Intel Galileo	Raspberry Pi (Model B)	Arduino Uno
Wymiary	10cm x 7cm	85.60mm x 56mm x 21mm	5.59cm x 16.5cm
Procesor	Intel Quark X1000	Broadcom BCM2835	ATmega328
Taktowanie	400MHz	700MHziv	16 MHz
Cache	16 KB	32KB L1 cache, 128KB L2 cache	-
RAM	512 SRAm	512 SRAM	2 kB
Analog I/O	6	17	6
Digital I/O	14	8	14
PWM	6	1	6

Tabela B.1. Specyfikacja dostępnych na rynku mikro kontrolerów

Źródło: http://eu.mouser.com/applications/open-source-hardware-galileo-pi/http://botland.com.pl/arduino-moduly-glowne/1060-arduino-uno-r3.html

DODATEK C

Programy

Katalogi *Galileo, PCF8574N* zawierają kod źródłowy oprogramowania stworzonego na potrzeby pracy.
Katalog *Galileo* zawiera oprogramowanie mikrokomputera Intel[1] Galileo.
Katalog *PCF8574N* zawiera oprogramowanie I/O Expander PCF8574N.

Bibliografia

[1] Agjffgjgdjfgflbert fffffEinstein. hkljklbkln. *Annalen der Physik*, 322(10):891–921, 1905.

Spis tabel

Spis rysunków

Oświadczenie

Ja, niżej podpisany(a) oświadczam, iż przedłożona praca dyplomowa została wykonana przeze mnie samodzielnie, nie narusza praw autorskich, interesów prawnych i materialnych innych osób.

.....	.....
data	podpis