Instytut Informatyki Politechniki Śląskiej

Zespół Mikroinformatyki i Teorii Automatów Cyfrowych

Rok akademicki:	Rodzaj studiów*: SSI/NSI/NSM	Przedmiot (Języki Asemblerowe/SMiW):	Grupa	Sekcja
2017/2018	**NSI**	**SMiW**	**BDIS**	**3**

Imię:	**Patryk**	**Prowadzący:** OA/JP/KT/GD/BSz/GB	**GD**
Nazwisko:	**Dusza**		

Raport końcowy

Temat projektu:

Czujnik odległości

Data oddania: dd/mm/rrrr	

Czujnik odległości

1. Założenia projektu:

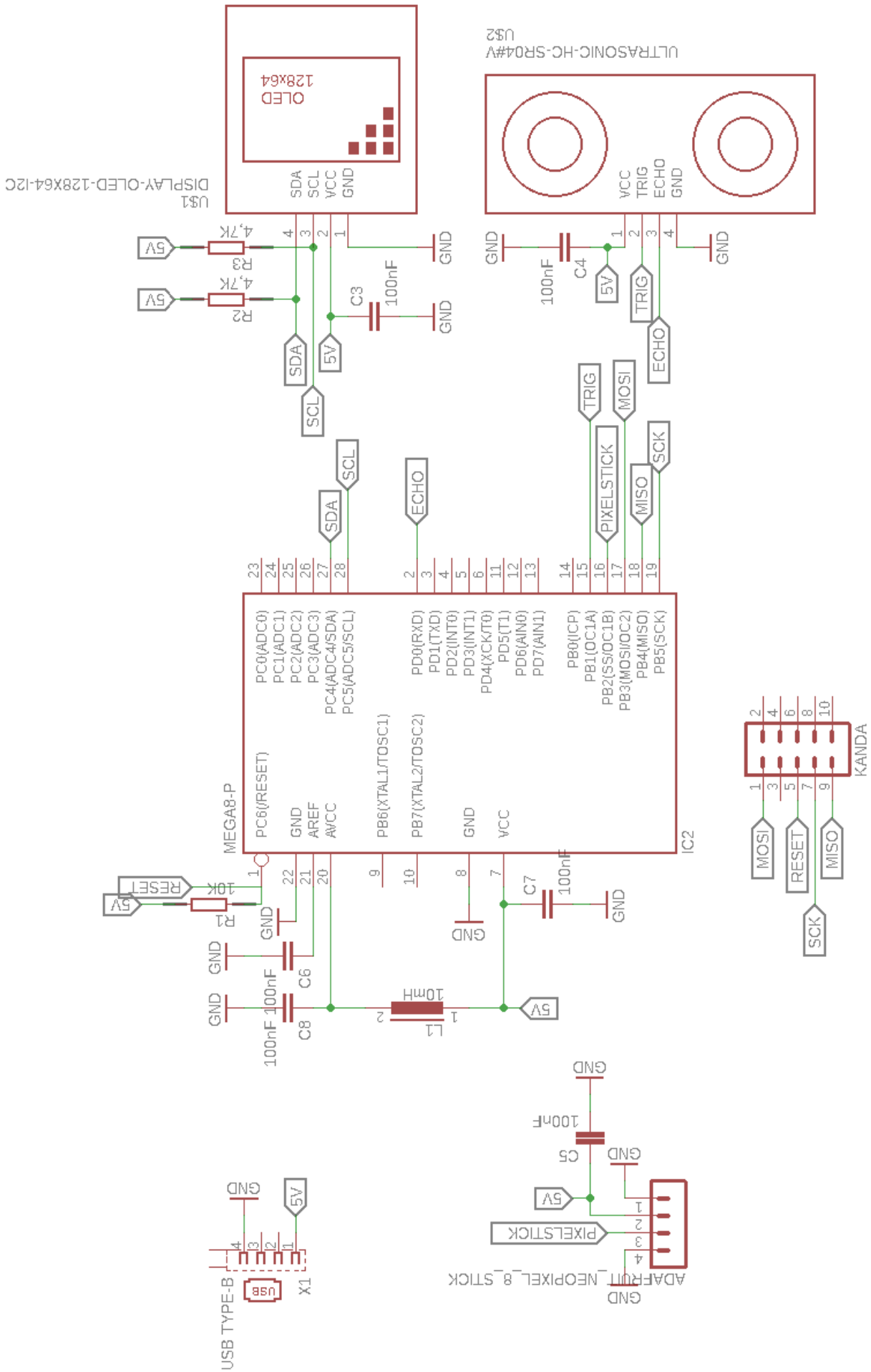
a. Pomiar odległości i wyświetlanie na wyświetlaczu OLED 0.96”
b. Wizualizacja odległości na programowalnej linijce LED
c. Wykorzystanie mikrokontrolera Atmega168

2. Funkcja urządzenia – urządzenie dokonuje pomiaru odległości co ~100ms, następnie wyświetla wynik na wyświetlaczu i linijce LED. Odległość 2cm-2m sygnalizuje kolorem czerwonym (2cm – mocno czerwony, 2m – słabszy), a od 2m do 4m kolorem zielonym(2m – słaby zielony, 4m – mocny).

3. Elementy zostały dobrane według ceny oraz napięcia zasilania. Udało mi się dobrać wszystkie podzespoły tak, aby pracowały one na napięciu 5V. Cały projekt został polutowany na uniwersalnej płytce.

4. Specyfikacja wewnętrzna:

a. Schemat blokowy:



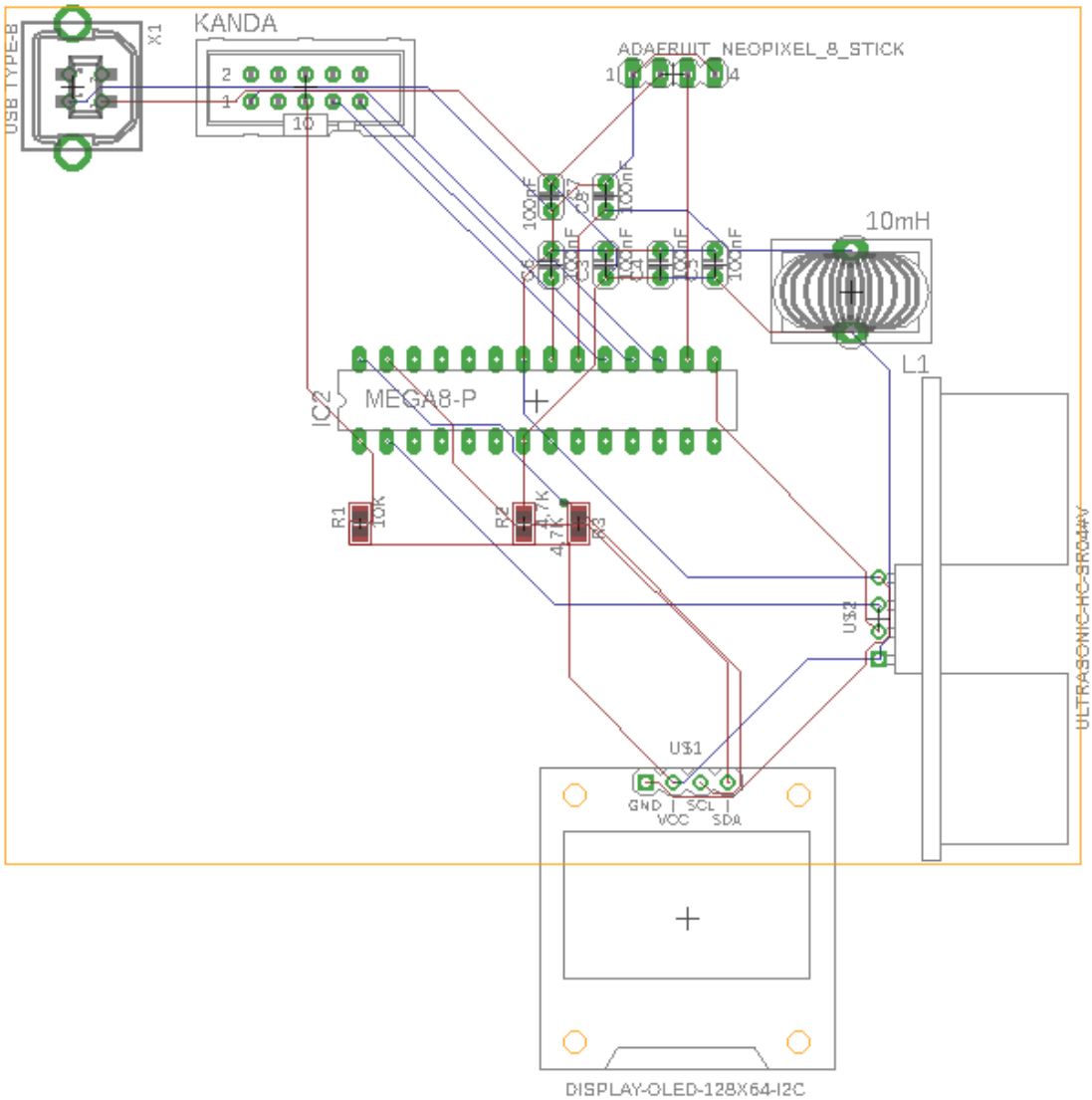
b. Opis funkcji poszczególnych bloków układu:

- Czujnik odległości HC-SR04 – na pin TRIG należy ustawić stan wysoki na 10us, a następnie mierzy się czas, na którym odbieramy sygnał wysoki z czujnika na wyjściu ECHO. Pomiar czasu jest wykonywany przez TIMER.

- Wyświetlacz SH1106 – komunikacja poprzez protokół I^2C

- Adafruit Neopixel WS2812 – programowalna linijka LED. Posiada tylko jeden PIN do komunikacji z mikrokontrolerem.

c. Schemat montażowy:



d. Lista wykorzystanych elementów

- Mikrokontroler AVR – Atmega168A-PU DIP
- Ultradźwiękowy czujnik odległości HC-SR04 2-200cm
- Adafruit NeoPixel Stick - pasek LED RGB 8 x WS2812 5050
- Wyświetlacz OLED niebieski graficzny 0,96'' 128x64px I2C
- Podstawka do układów DIP 28 pin
- 5x Kondensator ceramiczny 100nF/50V THT
- Płytka uniwersalna duża 830 pól - SparkFun
- USB typ B Proto - złącze do płytki stykowej
- 1x Gniazdo kątowe 1x4pin raster 2,54mm
- 2x Gniazdo proste 1x4pin raster 2,54mm - pionowe
- 1x Rezystor THT 3W 10kΩ
- 2x Rezystor THT 3W 4,7kΩ
- Dławik pionowy 10mH 10% 9x12mm 120mA 13R

e. Algorytm oprogramowania urządzenia:

Inicjalizacja

WHILE 1

Dokonaj pomiaru odległości

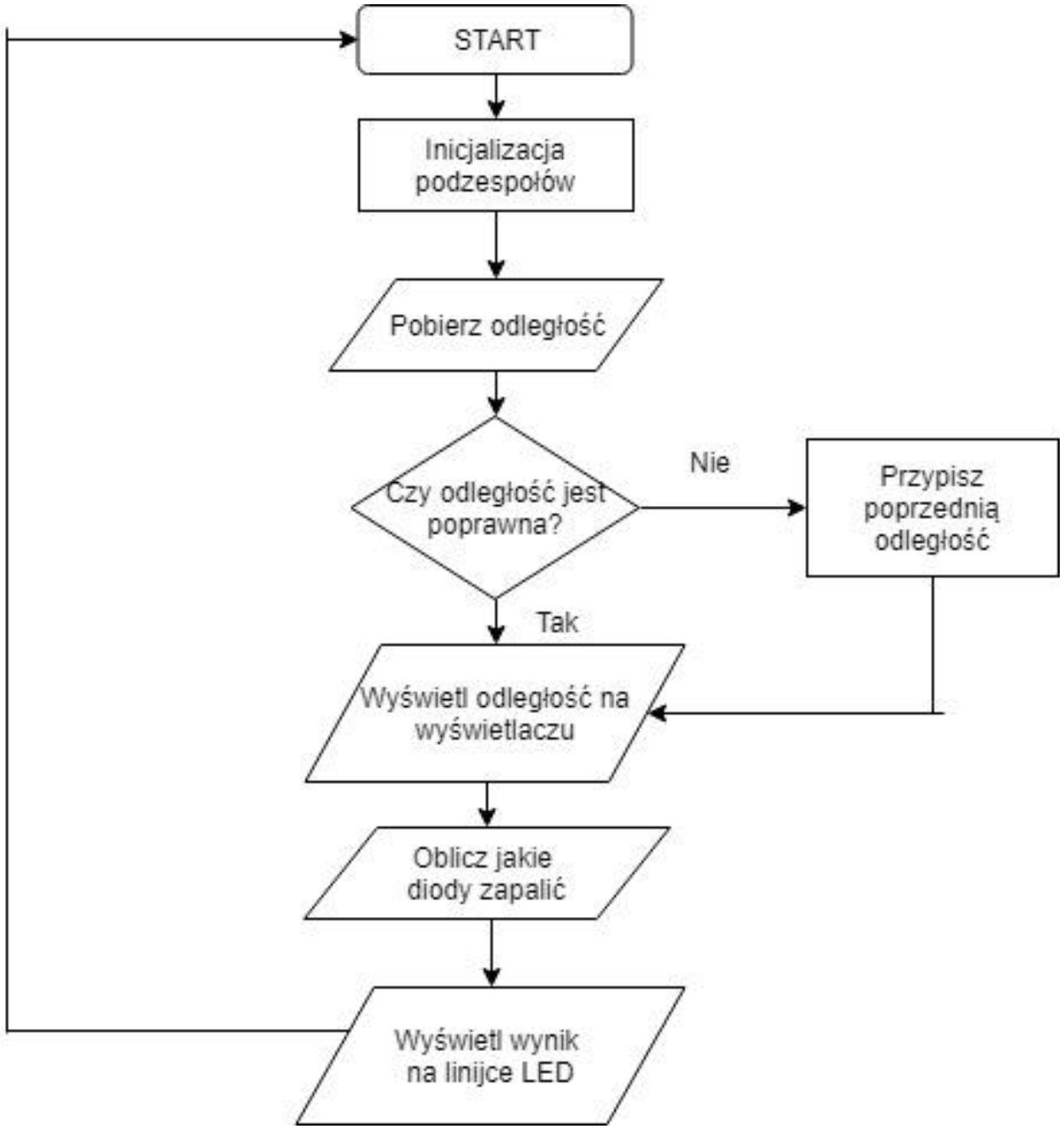
IF wartość odległości nie jest poprawna

    THEN przypisz poprzednią odległość

Wyświetl wynik na ekranie

Oblicz kolor dla odległości

Wyświetl wynik na linijce LED


START
Inicjalizacja podzespołów
Pobierz odległość
Czy odległość jest poprawna?
Nie
Przypisz poprzednią odległość
Tak
Wyświetl odległość na wyświetlaczu
Oblicz jakie diody zapalić
Wyświetl wynik na linijce LED

f. Opis zmiennych

- int pulse – zmienna przechowująca czas na wyjściu z czujnika odległości.
- int flag – zmienna przechowująca flagę uruchamiającą Timer lub odczytującą wartość z Timera i zerowanie go.
- int prev_dist – zmienna przechowująca poprzednią odległość na wypadek złego pomiaru.

g. Opis funkcji

- void InitInterrupt() – funkcja inicjalizująca przerwania
- void FullInit() – funkcja inicjalizująca porty, wyświetlacz i dokańczająca inicjalizację przerwań
- void draw() – funkcja wypisująca na wyświetlacz aktualnej odległości
- void set_red(int n) – funkcja ustawiająca w bibliotece wartość n czerwonego koloru diód na linijce LED
- void set_green(int n) – funkcja ustawiająca w bibliotece wartość n zielonego koloru diód na linijce LED
- void isDistanceCorrect() – sprawdza czy pomiar został wykonany poprawnie
- void ShowDistanceOnLed() – funkcja obliczająca jakie diody zapalić i wyświetlająca odległość na linijce LED
- ISR(INT0_vect) – funkcja do obsługi przerwania, uruchamiająca TIMER lub odczytująca wartość z TIMERa i zerująca go

5. Specyfikacja zewnętrzna:

a. Opis funkcji elementów wykonawczych

- Wyświetlacz OLED 0.96'' SH1106 – Informuje o aktualnej odległości pobranej z czujnika
- Adafruit Neopixel WS2812 – Na diodach pokazuje aktualną odległość

b. Instrukcja obsługi

Aby uruchomić urządzenie należy podpiąć zasilania przez kabel USB typu B przykładowo z powerbanka, który na wyjściu ma 5V oraz ~2A. Urządzenie samo się uruchamia i wykonuje pomiary. Dokładność zależy od powierzchni od jakiej odbije się ultradźwięk z czujnika, dlatego czasami mogą wystąpić przekłamania.

6. Testowanie

Na początku cały układ podłączyłem na płytce stykowej, aby wszystko po kolei podpinać i sprawdzać działanie na bieżąco każdego podzespołu. Jako pierwszy podłączyłem wyświetlacz i oprogramowałem mikrokontroler, aby móc wyświetlać zmienne na ekranie, co zastąpiło debugowanie urządzenia. Następnie podłączyłem czujnik i wyświetlałem wynik na wyświetlaczu w celi sprawdzenia poprawności algorytmu. Jako ostatnie podłączyłem linijkę LED, której wiedziałem, że będę musiał poświęcić najwięcej czasu, ponieważ był problem z biblioteką, ale po której próbie udało się wszystko ze sobą zgrać i cały projekt zaczął działać pomyślnie.

7. Wnioski

a. Problemu, które wystąpiły podczas montażu i uruchamiania

Jeden z większych problemów na początku sprawiała płytka stykowa, ponieważ nigdy nie byłem do końca pewny czy kable stykają (czasem trzeba było mocniej docisnąć), czy mam jakiś problem w programie. Największy jednak problem narodził się, gdy musiałem podłączyć linijkę LED. Próbowałem kilku bibliotek, jednak linijka świeciła tylko intensywnym białym kolorem. Myślałem, że coś zwarłem jak ją lutowałem (nie miała żadnych pinów), dlatego kupiłem dwie z innych firm, jednak na nich był taki sam problem. Na końcu okazało się, że w ATmedze mam ustawiony fusebit CKDIV8, który dzielił taktowanie procesora przez 8, czyli symbol, którym definiowałem prędkość nie był do końca poprawny, a przy okazji wyszło, że musiałem także przerobić obsługę czujnika.

b. Testy poprawności działania urządzenia

Projekt jeszcze w wersji prototypowej testowałem mierząc odległość od ściany, lub jakiegoś przedmiotu i sprawdzając czy wyświetla poprawny wynik. Dzięki tym testom zauważyłem, że duże znaczenie ma powierzchnia od której odbijają się ultradźwięki z czujnika, ponieważ czasem otrzymywałem niepoprawny wynik. Po lutowaniu projektu wykonałem te same testy, które wykonałem dla prototypu.

Projekt jest dostępny pod adresem: https://github.com/vSoulz/SMiW_Projekt