



Instytut Informatyki Politechniki Śląskiej

Zespół Mikroinformatyki i Teorii Automatów
Cyfrowych



Rok akademicki:	Rodzaj studiów*: SSI/NSI/NSM	Przedmiot Asemblerowe/SMiW): (Języki	Grupa	Sekcja
2020/2021	SSI	SMiW	5	1
Imię:	Justyna	Prowadzący:	HM	
Nazwisko:	Książek	OA/JP/KT/GD/BSz/GB		

Raport końcowy

Temat projektu:

Dalmierz Ultradźwiękowy

urządzenie do pomiaru odległości od obiektu za
pomocą fal dźwiękowych

Data oddania:

dd/mm/rrrr

08/02/2021

1 TEMAT PROJEKTU

Tematem projektu jest zaprojektowanie oraz zbudowanie układu dalmierza ultradźwiękowego, mierzącego odległość od miernika do obiektu.

1.1 OPIS ZAŁOŻEŃ I FUNKCJI URZĄDZENIA

Układ powinien działać z możliwie najmniejszym błędem pomiarowym w zakresie odległości od 2cm do 4m. Urządzenie powinno być wyposażone w złącze mikroUSB, umożliwiające zasilanie napięciem sieciowym jak i źródłem przenośnym (np. powerbank).

Urządzenie powinno umożliwiać włączanie i wyłączanie za pomocą przycisku umieszczonego na obudowie oraz wielokrotne pomiary bez potrzeby resetu czy odłączania zasilania.

Miernik powinien sygnalizować gotowość do pracy i ukończenie pomiaru za pomocą kolorowych diod LED oraz wyświetlać wyniki pomiaru na wyświetlaczu LCD.

2 ANALIZA ZADANIA

Projektowany układ do działania zgodnie z powyższymi założeniami potrzebuje mikrokontrolera sterującego pracą urządzenia, czujnika odległości oraz wyświetlacza.

Ponadto do wykonania projektu niezbędne będą elementy takie jak rezystory, przyciski, kondensatory oraz rezonator kwarcowy.

2.1 UZASADNIENIE WYBORU ELEMENTÓW ELEKTRONICZNYCH

Atmel 89S51 to tani i szeroko dostępny mikrokontroler z funkcją ISP, ułatwiającą wprowadzanie zmian w projekcie i programowanie mikrokontrolera bezpośrednio w układzie. Posiada 4kB pamięci – ilość wystarczającą do realizacji projektu.

LCD 16x2 to najpopularniejszy model wyświetlacza LCD, który jest bardzo tani oraz posiada wystarczającą przestrzeń do wyświetlania prostych komunikatów programowych. Większość wyświetlaczy tego typu jest stworzonych w standardzie zgodnym z HD44780.

HC-SR04 to bardzo popularny i niedrogi czujnik do pomiaru odległości. Urządzenie posiada tylko cztery wyprowadzenia i jego obsługa jest nietrudna, dzięki czemu proces programowania i testowania przebiega bez większych przeszkód.

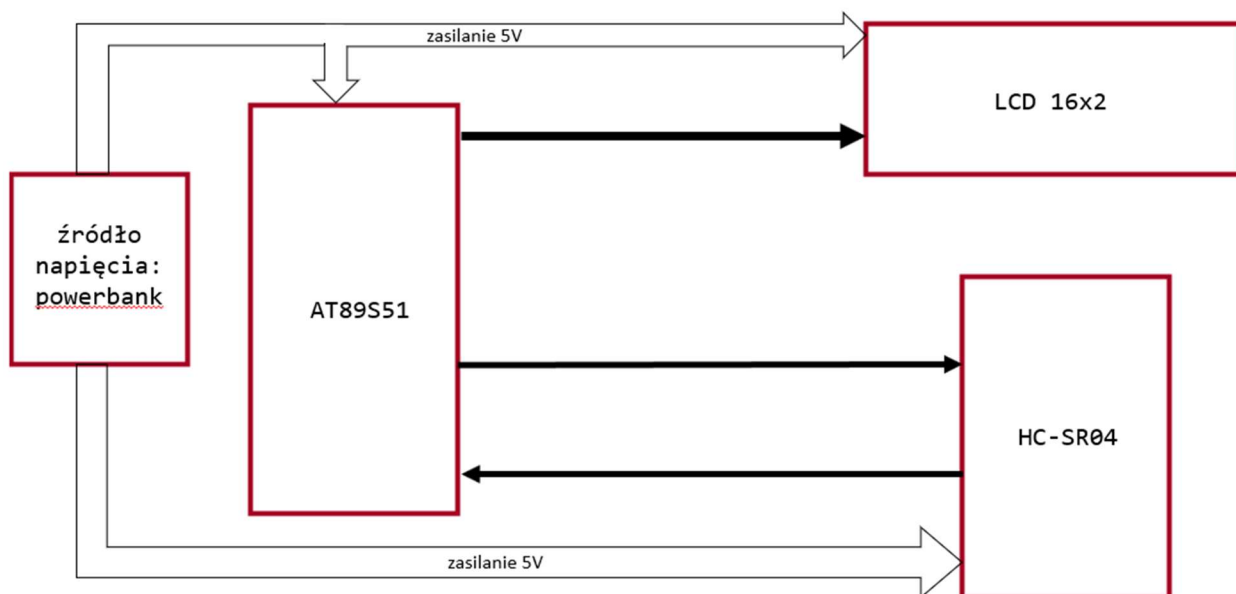
2.2 UZASADNIENIE WYBORU NARZĘDZI UŻYTYCH DO REALIZACJI PROJEKTU

Keil uVision 5 popularne i darmowe narzędzie do programowania, debugowania i kompilowania programów przeznaczonych do obsługi mikrokontrolerów. Baza urządzeń i bibliotek programu jest bardzo rozbudowana, dzięki czemu nie trzeba samemu definiować funkcji wyprowadzeń urządzenia. Ponadto na stronie producenta dostępna jest obszerna baza wiedzy wyjaśniająca zagadnienia programowania Cx51.

progISP 17.2 to łatwo dostępne i bezpłatne oprogramowanie do wypalania skompilowanych plików HEX na mikrokontrolerach przez złącze USBASP. Program zawiera bogatą bibliotekę obsługiwanych mikrokontrolerów, a jego obsługa jest prosta i przejrzysta.

3 SPECYFIKACJA WEWNĘTRZNA URZĄDZENIA

3.1 SCHEMAT BLOKOWY UKŁADU



3.2 OPIS FUNKCJI POSZCZEGÓLNYCH BLOKÓW UKŁADU

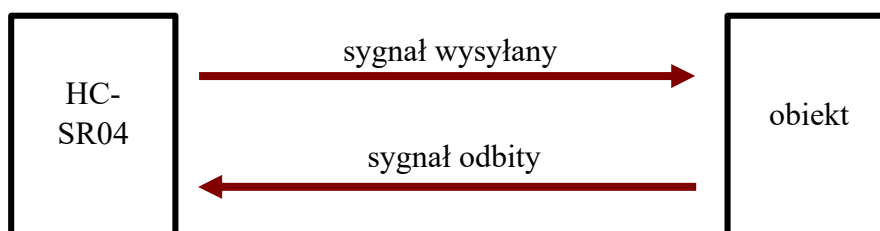
Urządzenie składa się z bloku mikrokontrolera kierującego pracą układu, bloku wyświetlacza LCD oraz bloku modułu pomiarowego nadającego i odbierającego sygnały ultradźwiękowe. Układ jest zasilany przez złącze USB w bloku źródła napięcia.

3.2.1 Szczegółowy opis działania ważniejszych elementów układu

Mikrokontroler wysyła sygnały sterujące do urządzeń oraz odbiera informacje od podłączonych modułów. Analizuje stan układu i reaguje zgodnie z programem sterującym, szczegółowo omówionym w dalszej części raportu.

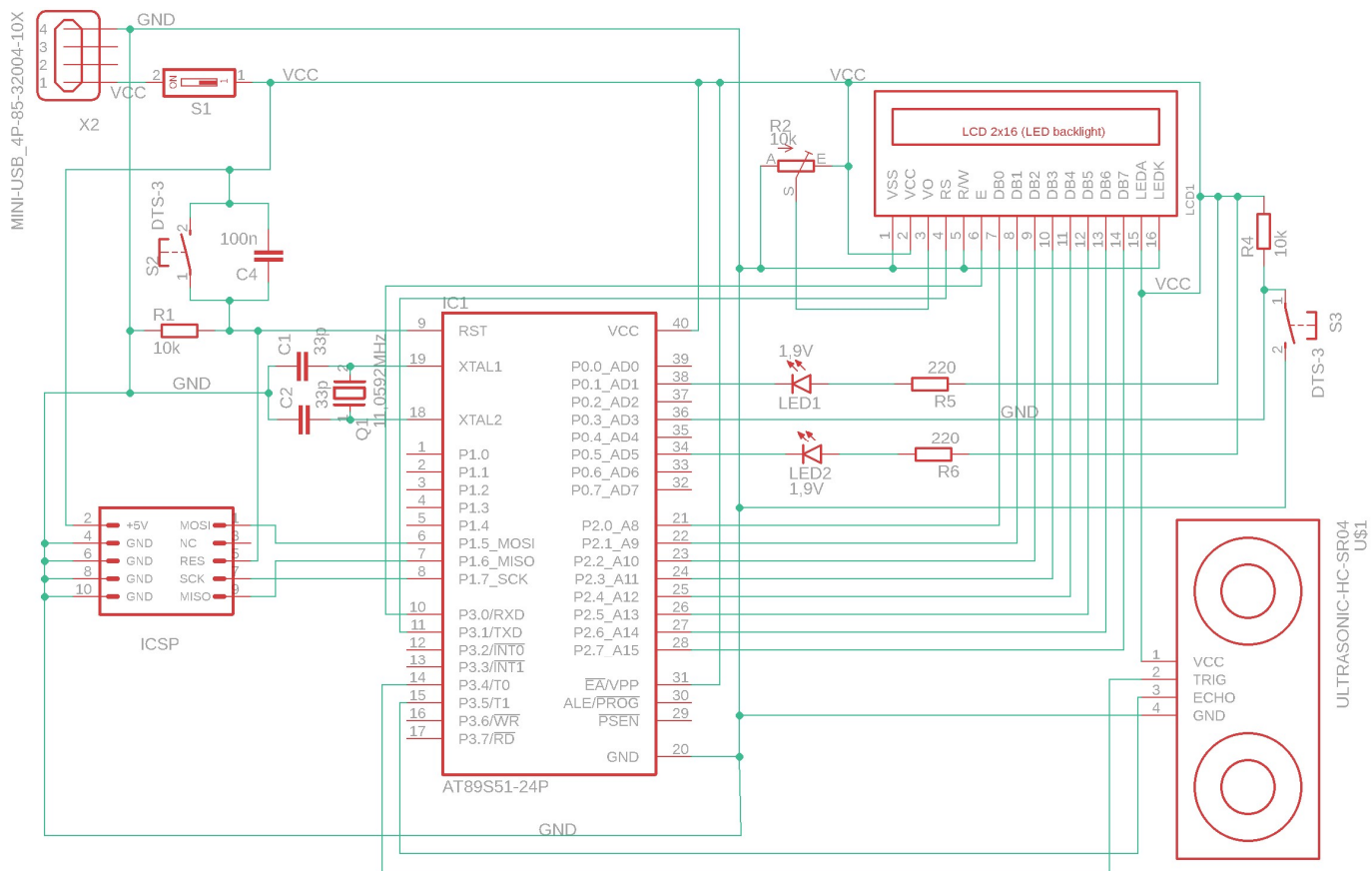
Ekran LCD jest podłączony do mikrokontrolera w konfiguracji do pracy w trybie 8-bitowym, gdyż jest to jedyny tryb obsługiwany przez wykorzystany model wyświetlacza. Reaguje na sygnały wysyłane liniami sterującymi i danych, wyświetlając na ekranie zbiory znaków.

Moduł HC-SR04 działa zgodnie z zasadą sonaru, zilustrowanej na poniższej grafice – nadajnik wysyła impuls ultradźwiękowy, który odbija się od obiektu i powraca do odbiornika.

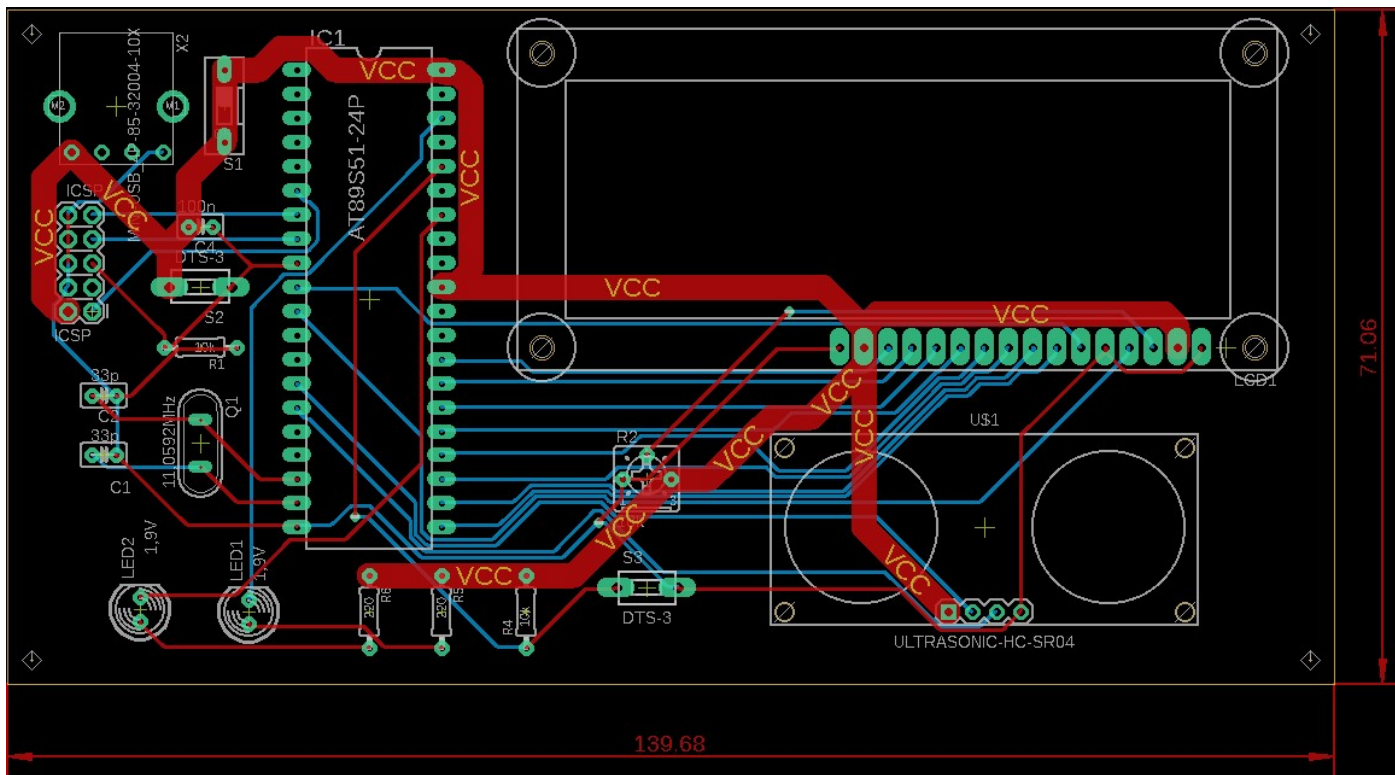


Czujnik jest podłączony do mikrokontrolera dwoma liniami sygnałowymi: TRIG i ECHO. Na linię TRIG wysyłany jest sygnał rozpoczynający pomiar, następnie urządzenie wysyła serię impulsów ultradźwiękowych i nasłuchuje w oczekiwaniu na ich powrót. Po odebraniu impulsów moduł odsyła sygnał linią ECHO.

3.3 SCHEMAT IDEOWY URZĄDZENIA



3.4 SCHEMAT MONTAŻOWY



3.5 LISTA ELEMENTÓW

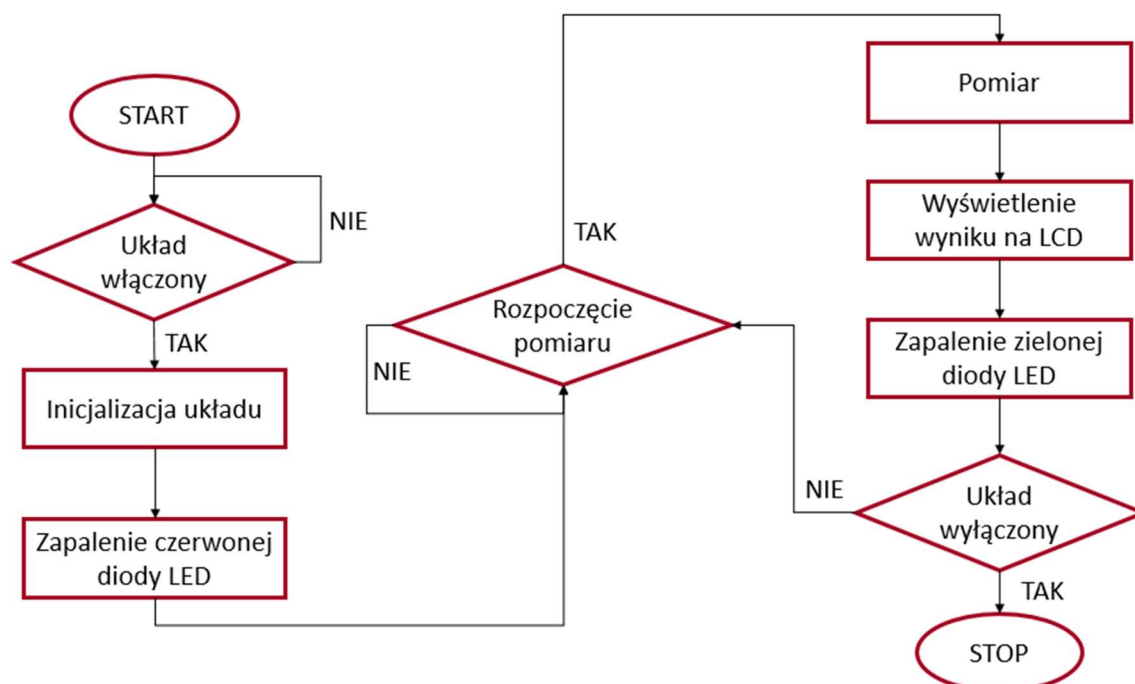
- » Mikrokontroler AT89S51
- » Wyświetlacz LCD 16x2 WC1602A
- » Moduł dalmierza ultradźwiękowego HC-SR04
- » Moduł zasilania mikroUSB typu B
- » Rezonator kwarcowy 11,0592MHz
- » Dioda LED czerwona 2,8V
- » Dioda LED zielona 2,8V
- » Przełącznik bistabilny ON/OFF
- » Przełącznik monostabilny RESET (x2)
- » Potencjometr 10k Ω
- » Rezystor 220 Ω (x2)
- » Rezystor 10k Ω (x2)
- » Kondensator 33pF (x2)
- » Kondensator 100nF
- » Gniazdo IDC-10

3.6 ALGORYTM OPROGRAMOWANIA URZĄDZENIA

Urządzenie po włączeniu inicjalizuje wyświetlacz LCD i przygotowuje się do działania. Następnie oczekuje na akcję ze strony użytkownika. Po wciśnięciu przycisku START inicjowany jest pomiar i obliczana odległość od obiektu. Wynik wyświetlany jest na ekranie LCD. Ponowne wciśnięcie przycisku Start skutkuje wykonaniem kolejnego pomiaru.

Pełen kod źródłowy został załączony na końcu raportu jako „Dodatek – kod źródłowy”.

3.6.1 Schemat blokowy



3.6.2 Opis ważniejszych zmiennych

W kodzie głównym używana jest zmienna przechowująca pełen wynik pomiaru, będąca liczbą zmiennoprzecinkową. Dodatkowo w funkcjach wykorzystywana jest zmienna służąca do wyznaczania cyfry do wyświetlenia oraz iteratory pętli.

3.6.3 Opis funkcji procedur

Funkcje zwarte w programie można podzielić na trzy główne kategorie: obsługa układu i czujnika, obsługa wyświetlacza LCD oraz obsługa timerów.

Układ i czujnik są obsługiwane wewnątrz funkcji main, która odpowiada za inicjalizację urządzenia, odbieranie sygnałów wysyłanych przyciskami oraz wysyłanie i odbieranie sygnałów modułu HC-SR04. Analizowana jest tam także mierzona odległość oraz generowane komunikaty do wyświetlenia na ekranie LCD.

Funkcje obsługujące LCD są odpowiedzialne za wysyłanie komend, znaków i napisów jak i inicjalizację wyświetlacza w sposób podawany przez producenta w dokumentacji.

Układ korzysta także z timerów wbudowanych w AT89S51 do generacji przerwań, mierzenia czasu i obsługi pracy układu. Timer0 jest wykorzystywany w funkcji delay generującej opóźnienia, a Timer1 użyto do mierzenia czasu propagacji impulsu ultradźwiękowego.

3.6.4 Szczegółowy opis działania ważniejszych procedur

Funkcja obsługi układu inicjalizuje pracę układu i wysyła sygnał zapalający czerwoną diodę LED. Następnie oczekuje aż użytkownik wciśnie przycisk rozpoczęcia pomiaru.

Po otrzymaniu sygnału startu pomiaru układ wysyła sygnał na wyprowadzenie TRIG modułu HS-SR04 trwający minimum 10us, uruchamia Timer1 i nasłuchuje odpowiedzi na wyprowadzeniu ECHO. Jeżeli odpowiedź nie jest otrzymana do czasu przepełnienia Timer1, wywoływana jest procedura obsługi przerwania, która sygnalizuje błąd miganiem czerwonej diody. W przeciwnym przypadku obliczana jest odległość za pomocą następującego wzoru:

$$s = \frac{\text{prędkość dźwięku} \cdot \text{licznik timera} \cdot \text{czas wykonywania instrukcji}}{2}$$

Wynik równania dzielony jest na pół, gdyż impuls przebywa obliczany dystans dwa razy - podróżując od modułu do obiektu i po odbiciu od obiektu do modułu. Czas podróży obliczany jest poprzez pomnożenie zawartości rejestrów licznika Timer1 przez czas wykonywania jednej instrukcji w mikrokontrolerze.

$$\text{częstotliwość timera} = \frac{\text{częstotliwość rezonatora}}{12} = \frac{11,0592 \text{ MHz}}{12} = 921,6 \text{ kHz}$$

$$\text{czas wykonywania instrukcji} = \frac{1}{921,6 \text{ kHz}} = 1,085 \mu s$$

Stąd wzór na mierzoną odległość po podstawieniach ma postać:

$$s = \frac{34000 \frac{\text{cm}}{\text{s}} \cdot \text{licznik timera} \cdot 1,085 \cdot 10^{-6} \text{ s}}{2}$$

Dla oszczędności pamięci zajmowanej przez kod wszystkie stałe wartości zostały ze sobą wymnożone, a operacje zestawione w krótkim równaniu.

Funkcje obsługujące Timer1 służą do mierzenia czasu propagacji sygnału wysłanego przez czujnik. Rejestry Timer1 są zerowane, a on sam włączany. Jeżeli czujnik odbierze sygnał zegar jest zatrzymywany, a zawartość jego rejestrów jest podstawą obliczania odległości.

Wyświetlanie napisów na LCD jest realizowane przez funkcję LCD_string. W pętli odczytywane są kolejne znaki, aż do natrafienia na znak końca napisu. Każdy znak jest wysyłany do wyświetlacza poprzez funkcję LCD_char. Wyświetlanie wyniku jest realizowane w funkcji LCD_number, która rozkłada liczbę na czynniki i wyświetla je kolejno od setek do części setnych.

3.6.5 Opis interakcji oprogramowania z układem elektronicznym

Oprogramowanie mikrokontrolera komunikuje się z układem za pomocą zadeklarowanych wyprowadzeń i portu.

Port 2 wykorzystywany jest do obsługi LCD, więc jego adres zadeklarowano jako rejestr specjalnej funkcji i wystawiając na niego znaki przesyła się je do ekranu. Dodatkowo do obsługi wyświetlacza wykorzystano bity RS oraz E zadeklarowane jako wyprowadzenia 1 i 0 portu 3.

Bitów portu 3 użyto także do komunikacji z modułem HC-SR04 – pod pin 5 podpięto wyprowadzenie ECHO, a pod pin 4 – TRIG. Chcąc zainicjować pomiar wysyłany jest sygnał na wyprowadzenie TRIG i nasłuchuje się zmian na wyprowadzeniu ECHO.

Mikrokontroler komunikuje się również z diodami LED oraz z przyciskiem rozpoczynającym pomiar. Diody włączane są poprzez wystawienie logicznego 0 na wyprowadzeniach 1 lub 5 portu 0. Obsługę przycisku inicjuje się poprzez ustawienie wyprowadzenia 3 portu 0 jako urządzenia wprowadzania, a następnie poprzez oczekiwanie na zmianę sygnału z 1 na 0.

4 SPECYFIKACJA ZEWNĘTRZNA URZĄDZENIA

4.1.1 Opis funkcji elementów sterujących urządzeniem

Urządzenie jest sterowane przyciskiem rozpoczęcia pomiaru obsługiwanym programowo oraz przyciskiem resetu obsługiwanym sprzętowo.

4.1.2 Opis funkcji elementów wykonawczych

W układzie wykorzystano dwie diody LED do sygnalizowania użytkownikowi stanu, w jakim znajduje się układ. Czerwona dioda zapala się w momencie kiedy urządzenie jest gotowe do wykonania pomiaru lub miga w przypadku wystąpienia błędu pomiaru. Zielona dioda zapala się gdy wynik pomiaru jest gotowy i wyświetlony na ekranie.

Wyświetlacz w układzie służy do zakomunikowania użytkownikowi zmierzonej odległości. Jego kontrast można wygodnie regulować za pomocą potencjometru.

Dodatkowo układ korzysta z modułu HC-SR04 do pomiarów odległości. Czujnik jest aktywowany sygnałem z mikrokontrolera i wysyła sygnał zwrotny po odebraniu impulsu ultradźwiękowego.

4.1.3 Opis reakcji oprogramowania na zdarzenia zewnętrzne

Po przyciśnięciu przycisku rozpoczynającego pomiar oprogramowanie wysyła sygnał na wyprowadzenie HC-SR04. W momencie wykrycia sygnału zwrotnego od modułu dalmierza program rozpoczyna kalkulację odległości.

4.2 SKRÓCONA INSTRUKCJA OBSŁUGI URZĄDZENIA

- » Układ podłączyć do zasilania i włączyć przyciskiem ON/OFF, odczekać aż zaświeci się czerwona dioda LED oznaczająca, że układ został poprawnie zainicjowany.
- » Ustawić czujnik na wprost obiektu i wcisnąć zielony przycisk START.
- » Gdy pomiar zostanie wykonany zaświeci się zielona dioda LED, a jego wynik wyświetlony na ekranie LCD w centymetrach.
- » W razie potrzeby układ można zresetować czerwonym przyciskiem RESET.
- » Po zakończeniu pracy wyłączyć urządzenie przyciskiem ON/OFF.

4.3 OPIS ZŁĄCZY I/LUB SCHEMATU OKABLOWANIA

Elementy układu są połączone izolowanymi przewodami miedzianymi o przekroju 0.5mm². Przesyłane są nimi sygnały, napięcie oraz realizuje się nimi podłączenie do masy.

Na ich zastosowanie zdecydowano się, gdyż efektywnie realizują swoje zadania, a przy planowaniu układu połączeń nie trzeba się martwić krzyżowaniem się ścieżek. Ponadto łatwo je rozróżnić dzięki różnym kolorom izolacji i proces ich lutowania dla osoby z niewielkim doświadczeniem jest znacznie łatwiejszy niż tworzenie cynowych ścieżek.

5 OPIS MONTAŻU UKŁADU

5.1.1 Sposobu uruchamiania

W początkowych etapach projektowania i testowania układ zawsze uruchamiano bezpośrednio podłączając go do zasilania przez USBASP. Rozwiązanie to umożliwiało zaoszczędzenie pól na płytce prototypowej, szybkie przeprogramowywanie mikrokontrolera po wykryciu błędu.

Po finalizacji prototypu urządzenie uruchamiano poprzez bezpośrednie zapewnienie zasilania, za pomocą złącza mikroUSB. Do gotowego urządzenia dołączono przycisk ON/OFF umożliwiający szybkie uruchamianie po podłączeniu do zasilania.

5.1.2 Testowania

Układ projektowano i testowano na płytce prototypowej. Po zlutowaniu urządzenia po raz kolejny przeprowadzono wszystkie testy aby wykryć ewentualne błędy i zoptymalizować oprogramowanie. Wykonano następujące testy:

- » miganie diody – na pin, do którego podpięta jest dioda cyklicznie ustawiane są stany 0 i 1, opóźnienie jest realizowane funkcją wykorzystywaną w kodzie projektu

```
while(1)
{
    LEDG = 0;    //led on
    LEDR = 1;    //led off
    delay(1000);
    LEDG = 1;    //led off
    LEDR = 0;    //led on
    delay(1000);
}
```


- » włączanie diody po naciśnięciu przycisku – na pin, do którego podpięto przycisk monostabilny nasłuchuje się zmiany stanu i po jej nastąpieniu załączana jest dioda LED, która gaśnie po krótkim czasie

```
button = 1; //switch = input
LED = 1;    //LED off
while(1)
{
    if(button == 0)
    {
        LED = 0; //LED on
        delay(2000);
        LED = 1; //LED off
    }
}
```

- » wyświetlenie prostego napisu na LCD - prosty program korzystający z funkcji użytych w programie głównym, po podłączeniu do zasilania na ekranie wyświetlany jest napis i zapalana dioda

```
signal = 1; //led off
LCD_init(); //initilize lcd
LCD_string("Test LCD... \n projekt SMIW <3");
signal = 0; //led on at the end
while(1);
```

Pełen układ testowano na płytce prototypowej ustawiając urządzenie w różnych odległościach od ściany i mierząc dystans miarką. Wszystkie pomiary kilkukrotnie powtórzono i obliczono błąd pomiarowy. Z powodu ograniczenia dostępnej przestrzeni roboczej maksymalna zmierzona odległość na tym etapie testów wyniosła 70 cm.

Po zlutowaniu układu sprawdzono ciągłość wszystkich połączeń za pomocą woltomierza i przeprowadzono pomiary różnych odległości w seriach. Sprawdzono najmniejszą i największą możliwą do zmierzenia odległość oraz obliczono błąd pomiarowy.

5.2 WNIOSKI I UWAGI Z PRZEBIEGU PRACY

- » Dobór elementów do układów wymaga dokładnego przemyślenia aspektów takich jak: możliwość programowania elementu, ilość dostępnej pamięci oraz wymagane napięcie zasilania.
- » Zarówno podczas projektowania jak i programowania układów warto pracować z dokumentacjami elementów, których zamierzamy użyć.
- » Budowane układy warto prototypować i badać ich działanie za pomocą dostępnych elementów (diod LED, wyświetlaczy) oraz multimetra przed przystąpieniem do montażu w celu wyeliminowania błędów programistycznych i usterek sprzętowych.

5.3 PROBLEMY, KTÓRE WYSTĄPIŁY W TRAKCIE MONTAŻU I URUCHAMIANIA I JAK ZOSTAŁY ROZWIĄZANE

Na samym początku pracy natrafiono na problem z programowaniem mikrokontrolera. Model AT89S51 zawiera ISP oraz możliwe jest programowanie go za pomocą USBASP, jednakże jako osoba początkująca miałam problem ze znalezieniem oprogramowania do wypalania, obsługującym moje urządzenie i niekorzystającym z podłączenia mikrokontrolera przez gniazdo LPT. Dodatkowym problemem było podłączenie programatora do mikrokontrolera na płytce prototypowej, z powodu braku styku. Zaradzono temu poprzez sklejenie i zlutowanie własnego gniazda IDC-10 z pojedynczych goldpin'ów.

Kolejnym problemem na który się natknęłam było podłączanie wyświetlacza LCD. Początkowo próbowano podłączyć i skonfigurować ekran do pracy w trybie 4-bitowym za pomocą popularnych bibliotek do obsługi wyświetlaczy zgodnych z HD44780. Po wielu nieudanych próbach wyświetlacz próbowano podłączać i konfigurować do pracy w trybie 8-bitowym, jednakże nadal napotymano problemy. Ekran udało się uruchomić i skonfigurować po przylutowaniu złącza typu goldpin i napisaniu własnego kodu obsługi na podstawie dokumentacji.

Ostatnimi problemami na jakie natrafiłam było obliczanie zmierzonego dystansu. Moje problemy wynikały zarówno ze złej interpretacji dokumentacji modułu HC-SR04 jak i wspieranie się niezaufanymi źródłami z internetu podczas programowania Timera. Po odszukaniu kompletnej dokumentacji mikrokontrolera byłam w stanie poprawić błędy w programie takie jak zły dzielnik podczas przeliczania częstotliwości kwarcu na częstotliwość Timera.

5.4 JAKIE PRZEPROWADZONO TESTY POPRAWNOŚCI DZIAŁANIA URZĄDZENIA

Sporządzono tabelkę z zestawieniem wyników wielokrotnie powtórzonych pomiarów dla różnych odległości, obliczono błąd pomiarowy, zbadano odległość minimalną i maksymalną. Sprawdzone też czy poziom zakłóceń wynikający z prowadzenia rozmowy i innych hałasów do 80dB w trakcie pomiaru powoduje zakłócenia i przekłamuje wyniki.

mierzona odległość	420	410	400	390	380	350	320
wynik pomiaru	419,5	410,25	399,94	389,23	380,05	349,78	319,33

mierzona odległość	290	270	250	220	200	160	130
wynik pomiaru	290,47	268,5	245,67	218,3	198,2	157,95	128,5

mierzona odległość	100	85	70	50	35	20	10
wynik pomiaru	98,8	83,93	69,2	50,35	34,54	20,5	10,6

Z testów wynika, iż urządzenie mierzy odległości od 2cm do 420cm ze średnim błędem pomiarowym 1cm. Pomiary przeprowadzone przez urządzenie były bardzo dokładne dla niewielkich odległości (2cm do 45cm) oraz dystansów bliskich maksymalnemu zasięgowi miernika, gdzie błąd pomiarowy wynosił do 0.5cm. Natomiast dla pomiarów z zakresu od 50cm do 2,80m błąd pomiarowy przyjmował znacznie bardziej nieregularne wartości.

6 WNIOSKI KOŃCOWE

Z powodu znacznych błędów pomiarowych oraz trudności z dokładnym nakierowaniem urządzenia na obiekt, do którego mierzono dystans, wykorzystanie dalmierza ultradźwiękowego jako miernika stanowi nieoptymalne rozwiązanie. Trudność w pomiarach dystansów większych niż 50 cm stwarza także konieczność zapewniania braku innych obiektów mogących odbić fale dźwiękowe w promieniu ok 1m od celu. Wielokrotnie w czasie pomiarów urządzenie zwracało nieregularne i znacznie zaniżone wyniki, przez co wynikom pomiarów nie można całkowicie ufać.

Znacznie bardziej odpowiednim wykorzystaniem modułu pomiarów ultradźwiękowych jest zastosowanie go w prostych systemach alarmowych i czujnikach odległości, gdzie wymagane jest jedynie wykrywanie czy obiekt znajduje się w określonym zasięgu. Natomiast jako miernik znacznie lepiej sprawdzi się urządzenie wykorzystujące laser.

Mimo problemów z dokładnymi pomiarami wynikającymi z zastosowanej technologii wiele nauczyłam się podczas realizacji projektu i niewiele bym w nim zmieniła. Jedyną znaczącą modyfikacją, którą bym wprowadziła byłoby zastosowanie innego modelu mikrokontrolera, z większą ilością pamięci programowalnej.

W trakcie realizacji nauczyłam się korzystania z dokumentacji elementów w czasie projektowania i programowania układów, dzięki nim mogłam napisać proste i zwarte kody obsługi elementów. Ponadto poznałam różne sposoby testowania układów i połączeń między elementami oraz opanowałam sztukę lutowania oraz planowania układów płytek i okablowania.

Realizacja projektu była bardzo edukacyjnym doświadczeniem, które pozwoliło mi uzyskać praktyczne zdolności w dziedzinie projektowania i budowania układów elektronicznych. Dzięki wiedzy zdobytej w trakcie wykładów, ćwiczeń i laboratoriów z Systemów Mikroprocesorowych i Wbudowanych proces projektowania układu i pisanie programu sterującego jego pracą był znacznie łatwiejszy.

Udział w projekcie pozwolił mi zainteresować się projektowaniem, programowaniem i budowaniem układów elektronicznych i podjąć samodzielną naukę w tym kierunku. Doświadczenia wyniesione z projektu z pewnością okażą się wartościowe w trakcie mojej dalszej edukacji oraz w przyszłych prywatnych projektach.

7 LITERATURA

- » dokumentacja mikrokontrolera
<http://ww1.microchip.com/downloads/en/DeviceDoc/doc4316.pdf>
- » dokumentacja wyświetlacza
<http://www.wincomlcd.com/pdf/WC1602A-SFYLYHTC06.pdf>
- » dokumentacja modułu hs-sr04
<https://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf>
- » dokumentacja Cx51 Keil
https://www.keil.com/support/man/docs/c51/c51_intro.htm

8 DODATEK – KOD ŹRÓDŁOWY

Kod źródłowy został w całości napisany przeze mnie na podstawie dokumentacji elementów i mikrokontrolera oraz materiałów dostarczonych przez producenta mikrokontrolera.

```
//definicje wyprowadzeń i portów AT89S51 (biblioteka programu Keil uVision)
#include <regx51.h>

//diody LED
sbit LEDG = P0^5;
sbit LEDR = P0^1;
//przycisk START
sbit button = P0^3;
//wyświetlacz LCD
sfr display_port = 0xA0;
sbit rs = P3^1;
sbit e = P3^0;
//moduł HC-SR04
sbit echo = P3^5;
sbit trig = P3^4;

//prototypy funkcji opóźnienia i mierzenia czasu
void delay(unsigned int);
void timer_start();
void timer_ISR();
void timer_stop();
//prototypy funkcji obsługi LCD
void LCD_cmd(unsigned char);
void LCD_init();
void LCD_char(unsigned char);
void LCD_string(unsigned char*);
void LCD_number(float);

void main(void)
{
    float res = 0; //zmierzona i obliczona odległość
    button = 1;    //ustawienie wyprowadzenia jako input (przycisk START)
    LEDG = 1;      //LED red off
    LEDR = 0;      //LED green on
    echo = 1;      //ustawienie pinu jako input (odpowiedź HC-SR04)
    trig = 0;      //ustawienie niskiego stanu na pinie aktywującym pomiar
    LCD_init();
    TMOD = 0x11;   //przygotowanie Timer0 i Timer1
    while(1)
    {
        LCD_string("Gotowe do \n pomiaru");
        while(button); //oczekiwanie na wciśnięcie przycisku
        LEDR = 1;      //LED red off
        trig = 1;      //wysłanie długiego sygnału inicjującego pomiar
```

```

    delay(1);          //opóźnienie min 10us
    trig = 0;          //zakończenie impulsu inicjującego
    while(!echo);      //oczekiwane na stan wysoki na ECHO
    timer_start();      //początek pomiaru czasu
    while(echo);        //oczekiwanie na koniec stanu wysokiego na ECHO
    timer_stop();       //zakończenie pomiaru czasu
    LEDG = 0;          //LED green on
    res = (TL1 | (TH1<<8)) * 0.01860775; //obliczenie odległości
    LCD_string("wynik pomiaru: \n");
    LCD_number(res);

    while(button);     //oczekiwanie na wciśnięcie przycisku
    LEDG = 1;          //LED green off
    delay(2000);
    LEDR = 0;          //LED red on
}
}

//funkcje opóźniająca układ o podaną ilość ms - Timer0
void delay(unsigned int time)
{
    while(time!=0) //timer liczy po 1 ms
    {
        TL0 = 0x66;    //ustawienie wartości LOW timera
        TH0 = 0xFC;    //ustawienie wartości HIGH timera
        TR0 = 1;        //włączenie Timer0
        while(!TF0);    //oczekiwanie na overflow
        time--;          //zaktualizowanie czasu
        TF0 = 0;        //wyczyszczenie flagi overflow
    }
    TR0 = 0;            //wyłączenie Timera0
}

//wysyłanie komend do LCD
void LCD_cmd(unsigned char command)
{
    display_port = command; //ustawienie komendy na porcie LCD
    rs = 0;                //wysłanie sygnału komendy
    e = 1;                  //włączenie sygnału przetwarzania
    delay(2);
    e = 0;                  //wyłączenie sygnału przetwarzania
}

//inicjalizacja LCD
void LCD_init()
{
    delay(50);              //odczekanie min 30ms od włączenia
    LCD_cmd(0x3C);          //ustawienie funkcji wyświetlacza na pracę na 2 liniach
    delay(1);               //odczekanie >39us
}

```

```

    LCD_cmd(0x0F); //włączenie kursora i migania kursora
    delay(1);      //odczekanie >39us
    LCD_cmd(0x01); //wyczyszczenie wyświetlacza
    delay(2);      //odczekanie >1,53ms
    LCD_cmd(0x06); //ustawienie trybu wprowadzania
    delay(1);      //odczekanie >39us
}

//wysłanie znaku do wyświetlenia na LCD
void LCD_char(unsigned char letter)
{
    display_port = letter; //ustawienie znaku na porcie LCD
    rs = 1;               //wysłanie sygnału danych
    e = 1;                //włączenie sygnału przetwarzania
    delay(2);
    e = 0;                //wyłączenie sygnału przetwarzania
}

//wysłanie tekstu do wyświetlenia na LCD
void LCD_string(unsigned char* text)
{
    int i = 0;
    LCD_cmd(0x01); //wyczyszczenie wyświetlacza
    delay(2);      //odczekanie >1,53ms
    while(text[i] != '\0') //czytanie do znaku końca ciągu
    {
        if(text[i] == '\n') //przejdźcie do kolejnej linii po wykryciu znaku
        {
            LCD_cmd(0xC0);
            delay(1);
        }
        else //wysłanie znaku do wysłania
        {
            LCD_char(text[i]);
        }
        i++; //odczytanie kolejnego znaku w łańcuchu
    }
}

//wysłanie obliczonej odległości do wyświetlenia na LCD
//wykorzystanie rozkładu liczby na czynniki
void LCD_number(float number)
{
    int digit = 0; //cyfra do wyświetlenia
    int div = 100; //dzielnik liczby do rozłożenia wyniku na cyfry
    int i;
    for(i=0; i<5; i++) //wyświetlenie pięciu cyfr
    {
        if (div == 0) //wartości do obliczenia części dziesiętnej wyniku

```

```

    {
        LCD_char(',');
        div = 10;
        number = number * 100;
    }
    digit = number / div; //wyznaczenie cyfry - kolejnego czynnika
    LCD_char(digit + 48); //wysłanie kodu ASCII znaku do wyświetlania
    //obliczenie liczby do wyświetlania (po odjęciu współczynnika)
    number = number - digit * div;
    div = div / 10; //zaktualizowanie dzielnika
}
//wyświetlenie jednostki
LCD_char(' ');
LCD_char('c');
LCD_char('m');
}

//włączenie Timera1
void timer_start()
{
    EA = 1; //włączenie przerwań globalnych
    ET1 = 1; //włączenie przerwań Timera1
    //timer może liczyć do 65535
    TL1 = 0x00;
    TH1 = 0x00;
    TR1 = 1; //włączenie Timera1
}

//procedura obsługi przerwania Timera1
void timer_ISR() interrupt 3
{
    while(1) //miganie czerwoną diodą
    {
        LEDR = 1; //LED off
        delay(500);
        LEDR = 0; //LED on
        delay(500);
    }
}

//wyłączenie Timera1
void timer_stop()
{
    TR1 = 0; //wyłączenie Timera1
    EA = 0; //włączenie przerwań globalnych
    ET1 = 0; //włączenie przerwań Timera1
}

```