



**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**  
**WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI**  
INSTYTUT ELEKTRONIKI

**Projekt zaliczeniowy MPiMS**

**Sonar z wykorzystaniem czujnika ultradźwiękowego**

Autor: **Hubert Sternalski**

Kierunek studiów: **Elektronika i Teleinformatyka**

Kraków, 2024

## Spis treści

1	Wprowadzenie .....	3
1.1	Cel projektu .....	3
1.2	Zastosowanie.....	3
1.3	Założenia projektowe .....	3
2	Budowa systemu mikroprocesorowego.....	4
2.1	Schemat blokowy systemu .....	4
2.2	Mikrokontroler STM32 .....	5
2.3	Czujnik odległości HC-SR04 .....	7
2.4	Sterownik silnika TMC2226 .....	8
2.5	Reszta podzespołów .....	9
2.6	Schemat podłączenia systemu .....	10
3	Opis projektu .....	11
3.1	Uproszczony opis działania .....	11
3.2	Struktura projektu .....	11
3.3	Wykorzystane biblioteki .....	11
3.4	Diagram projektu.....	12
4	Realizacja projektu .....	13
4.1	Realizacja krok po kroku.....	13
4.2	Działanie skończonego projektu .....	15
4.3	Napotkane problemy.....	15
5	Podsumowanie .....	16
6	Bibliografia.....	17

# **1 Wprowadzenie**

## **1.1 Cel projektu**

Celem projektu jest stworzenie systemu mikroprocesorowego, którego sposób działania ma przypominać profesjonalne sonary wykorzystujące fale ultradźwiękowe do wykrywania obiektów na ich drodze. Całość oprócz mierzenia na bieżąco odległości od czujnika ultradźwiękowego ma za zadanie w czasie rzeczywistym rysować na podłączonym do niego komputerze miejsce wykrycia danego obiektu.

## **1.2 Zastosowanie**

Czujniki ultradźwiękowe są szeroko stosowane między innymi w samochodowych czujnikach parkowania, w robotach odkurzających do wykrywania obiektów, ścian oraz rysowania mapy pomieszczenia.

Urządzenie zbudowane w ramach tego projektu będzie w założeniu spełniać wszystkie zastosowania dla zwykłego czujnika odległości z tym, że nie będzie ograniczone do jednego kierunku. Jego dodatkowa funkcja jaką jest obrazowanie obiektów pod znacznie większym kątem, co usprawni jego zastosowanie, a dodatkowo odkryje przed nim przyszłe nowe zastosowania.

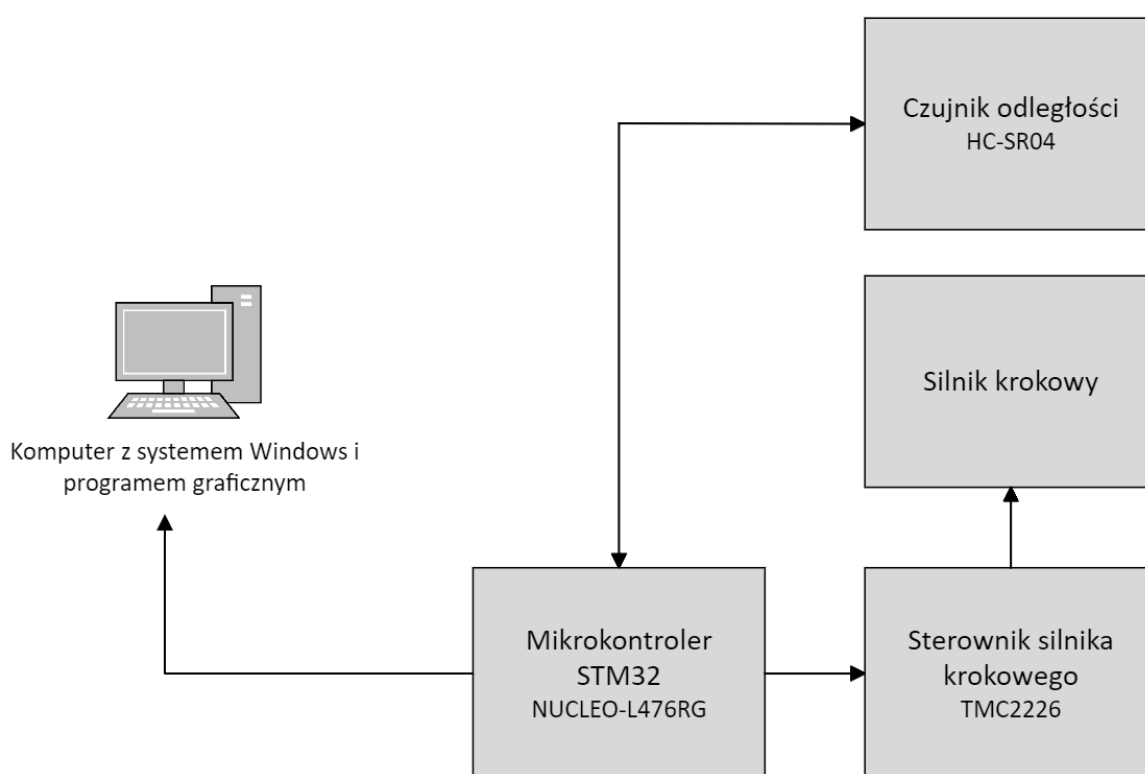
## **1.3 Założenia projektowe**

Elementem spinającym wszystkie podzespoły wykorzystane w zbudowaniu sonaru jest mikrokontroler STM32 znajdujący się na płycie rozszerzeniowej z programatorem NUCLEO-L476RG. Łączy on moduł z ultradźwiękowym czujnikiem odległości HC-SR04 z silnikiem krokowym sterowanym za pomocą sterownika TMC2226 oraz komputerem za pomocą portu USART. Na komputerze znajdować się będzie program, który w graficzny sposób w czasie rzeczywistym ma rysować wykryte obiekty wraz z kątem nachylenia do czujnika.

## 2 Budowa systemu mikroprocesorowego

### 2.1 Schemat blokowy systemu

Wszystkie elementy systemu mikroprocesorowego zostały przedstawione na schemacie blokowym widocznym poniżej. Mikrokontroler STM32 jest sercem całego układu, gdyż łączy ze sobą wszystkie poszczególne komponenty. Komunikuje się z czujnikiem odległości HC-SR04, Wysyła do sterownika wartość o jaki kąt ma zostać obrócony silnik krokowy, a także wysyła po USART-cie zmierzone wartości odległości wraz z danym w momencie pomiarów kątem do komputera. Tam w programie napisanym za pomocą języka programowania Python rysowane jest graficzne przedstawienie wykonanych pomiarów.



Rysunek 1 Schemat blokowy systemu mikroprocesorowego sonaru

Warto zauważyć, że komunikacja między mikrokontrolerem a czujnikiem odbywa się w dwie strony. Jest to spowodowane tym, że procesor zadaje do czujnika sygnał rozpoczynający pomiar odległości, a następnie czujnik zwraca do kontrolera wartości zmierzoną.

## 2.2 Mikrokontroler STM32

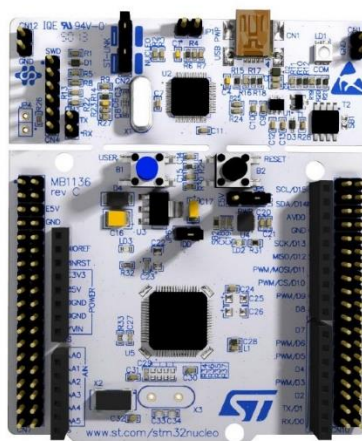
Mikrokontrolerem użytym w tym projekcie jest STM32 na płytce rozszerzeniowej NUCLEO-L476RG. Komponent ten został zakupiony wcześniej w celu naukowym, a jego parametry nie były wybierane dokładnie dla tego projektu.

Mikrokontroler STM32 na płytce rozszerzeniowej NUCLEO-L476RG to jednostka z rodziny STM32L4 firmy STMicroelectronics, zaprojektowana do zastosowań wymagających niskiego poboru mocy i wysokiej wydajności. Oto kilka kluczowych cech tej płytki:

- **Rdzeń** – ARM Cortex-M4 32-bitowy z jednostką FPU (Floating Point Unit), pracujący z maksymalną częstotliwością 80 MHz,
- **Pamięć** – 1 MB pamięci Flash, 128 KB SRAM,
- **Niskie zużycie energii** – różne tryby oszczędzania energii, umożliwiające długą pracę na zasilaniu bateryjnym,
- **Interfejsy komunikacyjne:**
  - USB 2.0 FS,
  - UART,
  - SPI,
  - I2C,
  - CAN,
  - I2S.
- **Peryferia:**
  - 12-bitowy przetwornik ADC,
  - DAC,
  - Komparatory,
  - Timery, w tym timery ogólnego przeznaczenia, timery zaawansowane, timery podstawowe i timery RTC (Real-Time Clock).

Płytką NUCLEO-L476RG to wszechstronne narzędzie do prototypowania, umożliwiające łatwe testowanie i rozwijanie aplikacji z użyciem mikrokontrolera STM32L476RG. Dzięki bogatemu zestawowi interfejsów i peryferiów, płytka ta jest idealna do szerokiej gamy projektów, od prostych aplikacji sterujących po zaawansowane systemy wbudowane. Dodatkowe możliwości oferowane nam przez płytkę rozszerzeniową:

- **Zgodność z Arduino Uno V3** – Umożliwia łatwe podłączanie tarczek (shields) Arduino,
- **Złącze ST morpho** – Dodatkowe wyprowadzenia pinów mikrokontrolera, umożliwiające dostęp do wszystkich pinów STM32,
- **Programator i debugger** – Wbudowany ST-LINK/V2-1, umożliwiający programowanie i debugowanie mikrokontrolera,
- **Zasilanie** – Możliwość zasilania przez USB, zewnętrzne źródło napięcia lub piny złącza,
- **Dioda LED i przyciski** – Wbudowane diody LED (np. dla zasilania, statusu) oraz przycisk RESET i przycisk użytkownika.



### Rysunek 2 STM32 NUCLEO-L476RG

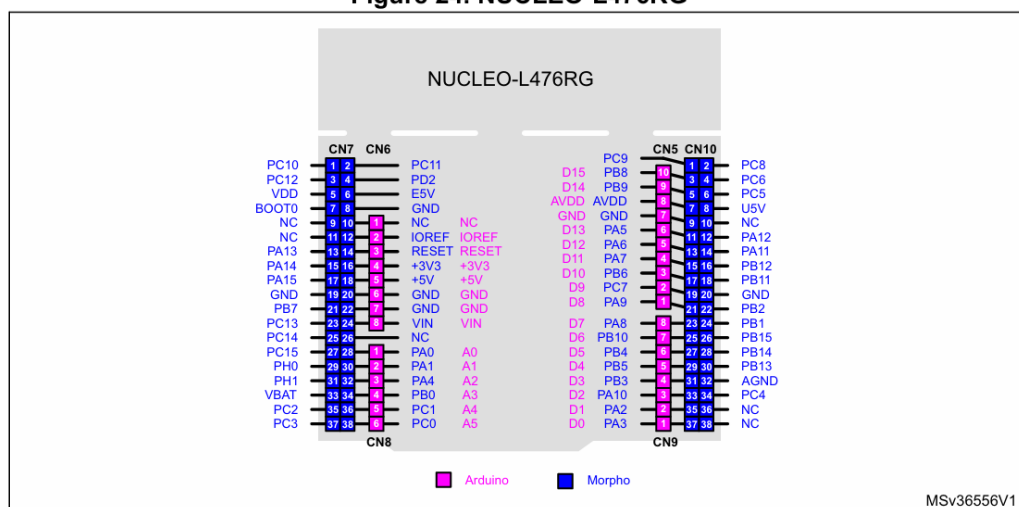
W projekcie tym wykorzystywane będą następujące porty mikrokontrolera:

- **PA2** – USART\_TX,
- **PA3** – USART\_TX,
- **PA9** – Tigger HC-SR04,
- **PA8** – Echo HC-SR04,
- **PB5** – DIR TMC2226,
- **PB10** – STER TMC2226.

Warto zaznaczyć, że sygnał oddawany przez ultradźwiękowy czujnik odległości HC-SR04 posiada stan wysoki 5V, gdzie cała logika mikrokontrolera STM32 jest 3.3V. Nie stanowi to problemu, gdyż użyte do tego wejścia tolerują logikę 5V i nie zostaną przez to uszkodzone.

Poniżej widać jeszcze rozmieszczenie poszczególnych portów na użytej płycie rozszerzeniowej.

**Figure 24. NUCLEO-L476RG**



*Rysunek 3 Rozkład portów STM32 NUCLEO-L476RG*

## 2.3 Czujnik odległości HC-SR04

Głównym komponentem pomiarowym sonaru jest moduł czujnika ultradźwiękowego HC-SR04. Jest on zasilany napięciem 5V, pobiera średnio 15mA w czasie pracy i jak podaje producent potrafi mierzyć odległość od 2 do 200cm.

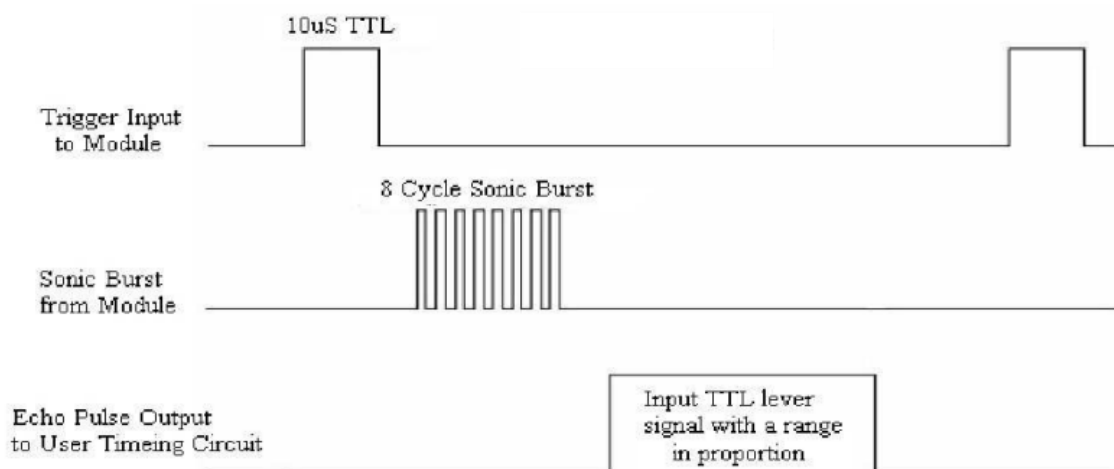
Posiada on 4 porty:

- **VCC** - +5V,
- **GND** – masa,
- **Echo** – Wyjściowy sygnał odległości,
- **Trig** – Wejściowy sygnał sterowania.



Rysunek 4 Moduł czujnika odległości HC-SR04

Pomiar odległości odbywa się po podaniu na wejście sygnałowe Trig pulsu logicznej jedynki trwającego dokładnie 10uS. Następnie moduł czujnika wysyła kilka impulsów do czujnika odległości, a następnie mierzy czas z jakim wraca on z powrotem. Po zmierzeniu odległości przez moduł na wyjściu Echo pojawia się puls, którego czas trwania odpowiada zmierzonej przez czujnik odległości.



Rysunek 5 Sposób działania czujnika odległości HC-SR04

Czas trwania pulsu możemy przekształcić na odległość od czujnika za pomocą podanego przez producenta wzoru. Odległość w metrach wynosi czas trwania pulsu w sekundach pomnożony przez 340 i podzielony na 2, ponieważ sygnał ultradźwiękowy musi wyjść z czujnika i do niego wrócić. Całość wygląda następująco:

$$L[m] = T[s] \cdot \frac{340}{2}$$

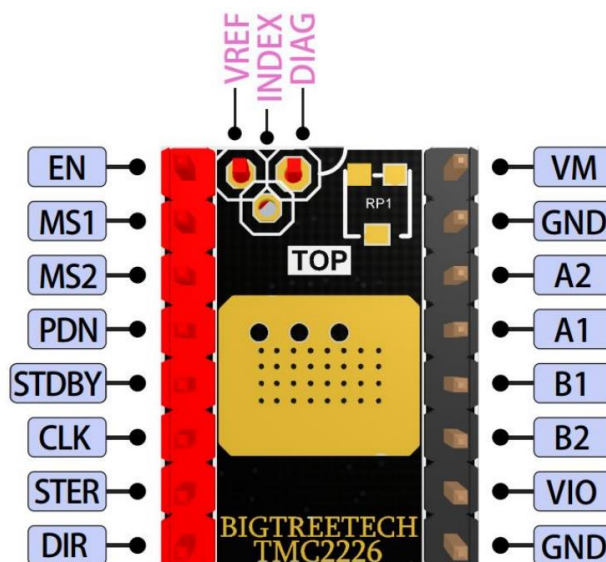
Alternatywnym rozwiązaniem jest podzielenie otrzymanego wyniku w mikrosekundach przez 58, co da nam od razu wynik w cm:

$$L[cm] = \frac{T[\mu s]}{58}$$

## 2.4 Sterownik silnika TMC2226

Sterownik TMC2226 służy do precyzyjnego sterowania silnikami krokowymi, stosowany jest w drukarkach 3D, maszynach CNC, robotyce, automatyce domowej i laboratoriach. Jest ceniony za precyzję, efektywne zarządzanie prądem oraz zaawansowane funkcje diagnostyczne.

W tym projekcie jego zadaniem będzie sterowanie silnikiem krokowym obracając go o zadany kąt po każdorazowym pomiarze odległości przez czujnik HC-SR04.



Rysunek 6 Sterownik silnika krokowego TMC2226

Sterownik ten daje dużo możliwości, które nie będą wykorzystywane w tym projekcie, więc skupię się tylko na tym do czego był potrzebny. Wszystkie porty wejściowe, które zostały wykorzystane do stworzenia tego systemu mikroprocesorowego to:



- **STER** – zadawanie kroków silnikowi,
- **DIR** – zmiana kierunku obrotu,
- **VM** – zasilanie silnika,
- **GND** – masa,
- **VIO** – zasilanie układu sterownika,
- **A2** - }
- **A1** - }
- **B1** - } Wyprowadzenia do silnika krokowego.
- **B2** - }

Sterowanie silnikiem odbywa się poprzez podawanie na port STER sygnału PWM. Modulując czas trwania i długość pulsów zmienia szybkość z jaką wykonywane są kolejne kroki silnika. Portem DIR zmieniamy kierunek obrotu silnika w zależności czy podany został stan niski, czy wysoki. VM to zasilanie silnika. Wymagane jest żeby było dosyć wydajne, więc w tym przypadku został zastosowany powerbank o napięciu wyjściowym 5V. Osobno został zasilony sam sterownik na wejściu VIO, tak żeby ewentualne zakłócenia powodowane przez silnik w czasie zatrzymywania nie wpływały na sam sterownik oraz mikrokontroler. Kolejno porty A1, A2, B1, B2 to porty do których podłączamy nasz silnik krokowy.

## 2.5 Reszta podzespołów

W tym podpunkcie zostaną opisane na szybko reszta użytych podzespołów, ponieważ ich opis nie jest istotny z punktu widzenia projektu.

### Silnik krokowy

Użyty w tym projekcie silnik krokowy został znaleziony w elektrośmieciach, a same jego parametry nie są bardzo ważne z punktu widzenia systemu, gdyż sterowanie nim nie będzie zmieniane i zostanie ustawione za pomocą prób i błędów do stanu jaki jest wymagany do działania sonaru.

### Komputer z systemem Windows

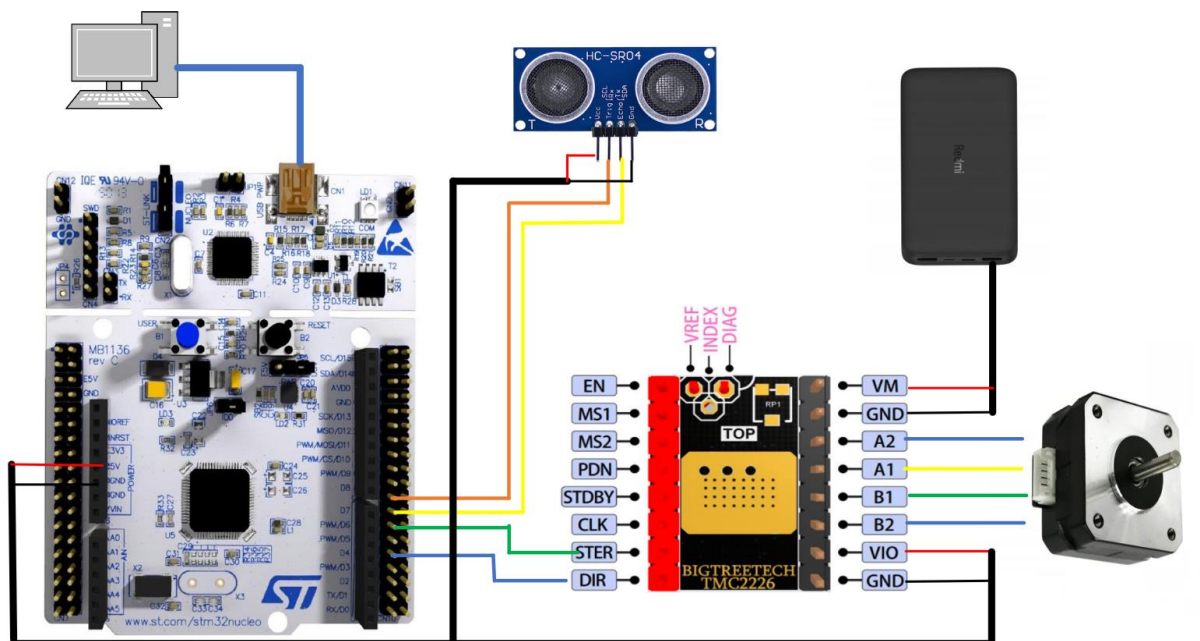
Komputer służy jedynie graficznemu przedstawieniu działania sonaru, i nie są wymagane specyficzne parametry jego podzespołów. System Windows nie jest również wymagany ze względu, że program napisany został w języku Python i działa na różnych platformach. Komunikacja z mikrokontrolerem odbywać się będzie za pomocą kabla USB po interfejsie komunikacyjnym USART.

### Powerbank

Powerbank służy jedynie do zasilania silnika krokowego, jak podaje jego producent posiada pojemność 20Ah. Na wyjściu jest stabilne 5V, które jest podane na opisany wyżej sterownik silnika.

## 2.6 Schemat podłączenia systemu

Poniżej widać sposób podłączenia całego układu systemu mikroprocesorowego. Podane wcześniej porty PA2 i PA3 do komunikacji z komputerem są podłączone w płytce rozszerzeniowej do portu mini USB, po którym odbywa się także zasilanie logicznej części układu.



Rysunek 7 Schemat podłączenia poszczególnych elementów układu

## 3 Opis projektu

### 3.1 Uproszczony opis działania

Mikrokontroler w celu wykrycia obiektu i zmierzenia jego odległości od sonaru wysyła puls trwający 10us do modułu czujnika odległości HC-SR04. Następnie ustawiony jest tryb czekania na porcie podłączonym do wyjścia czujnika. Jeżeli na porcie tym pojawi się zbocze nastające uruchamiany jest licznik czasu trwania pulsu. Po wykryciu zbocza opadającego zapisywana jest czas trwania pulsu, a następnie przeliczana na centymetry. Po wykonaniu pomiaru wysyłany jest z mikrokontrolera sygnał do sterownika silnika krokowego TMC2226, o przesunięciu o 5 stopni w jedną stronę. Następnie informacja o danym kącie i zmierzonej dla niego odległości wysyłane jest po USART-cie do komputera, gdzie na wykresie rysowane są kolejne wykryte obiekty. Jeżeli silnik krokowy dojdzie do końca zadanego zakresu to na wejście sterownika podawany jest stan przeciwny, co spowoduje zmianę kierunku obrotu silnika krokowego.

### 3.2 Struktura projektu

Całość projektu znajduje się w repozytorium pod linkiem:  
[https://github.com/HubertStern/Sonar\\_projekt\\_MPiMS.git](https://github.com/HubertStern/Sonar_projekt_MPiMS.git)

Główne jego foldery to:

- **graphics** – folder zawierający pliki kodu Python służące do generowania graficznego przedstawienia działania sonaru,
- **src** – folder z bibliotekami, źródłami oraz kodem obsługującym działanie sonaru,
- **documentation** – folder z dokumentacją projektu.

### 3.3 Wykorzystane biblioteki

W projekcie do obsługi mikrokontrolera użyte zostały następujące biblioteki w języku programowania C:

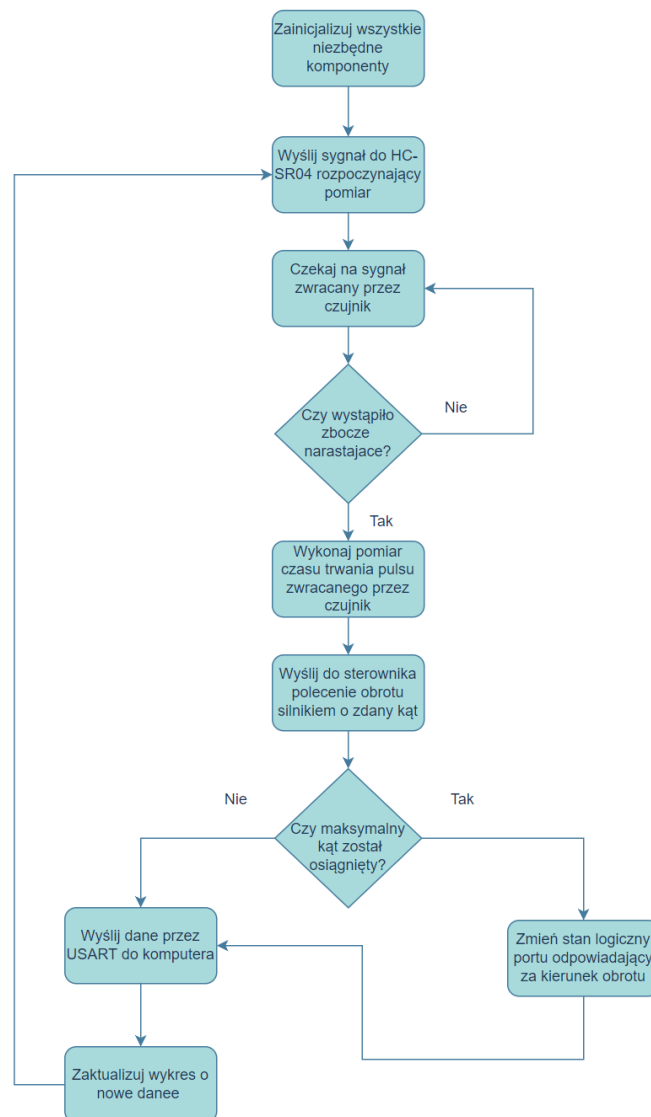
- **gpio** – Zarządza konfiguracją i sterowaniem portami wejścia/wyjścia (GPIO) mikrokontrolera,
- **HC-SR04** – Autorska biblioteka do obsługi czujnika odległości HC-SR04,
- **Stepper** – Autorska biblioteka do obsługi sterownika silnika krokowego,
- **stm32l4xx\_hal\_msp** – Implementuje funkcje inicjalizacyjne dla obsługi sprzętu specyficznego dla platformy STM32L476,
- **stm32l4xx\_it** – Zawiera obsługę przerwań dla mikrokontrolera STM32L476,
- **syscalls** – Realizuje funkcje systemowe, takie jak operacje wejścia/wyjścia, używane przez system wbudowany,

- **sysmem** – Zarządza przydziałem pamięci dynamicznej w systemie wbudowanym,
- **system\_stm32l4xx** – Konfiguruje system zegarów i inne podstawowe ustawienia mikrokontrolera STM32L476,
- **tim** – Obsługuje konfigurację i zarządzanie timerami w mikrokontrolerze,
- **usart** – Zarządza konfiguracją i komunikacją przez interfejs szeregowy USART.

Program do rysowania graficznego wykresu za pomocą zbieranych danych posiada następujące biblioteki w języku Python:

- **numpy** – Biblioteka do obliczeń numerycznych w Pythonie, oferująca wsparcie dla tablic wielowymiarowych oraz różnorodne funkcje matematyczne,
- **serial** – Biblioteka do komunikacji szeregowej, umożliwiającą wymianę danych między komputerem a urządzeniami z interfejsem szeregowym,
- **matplotlib** – Biblioteka do tworzenia wykresów i wizualizacji danych w Pythonie.

### 3.4 Diagram projektu

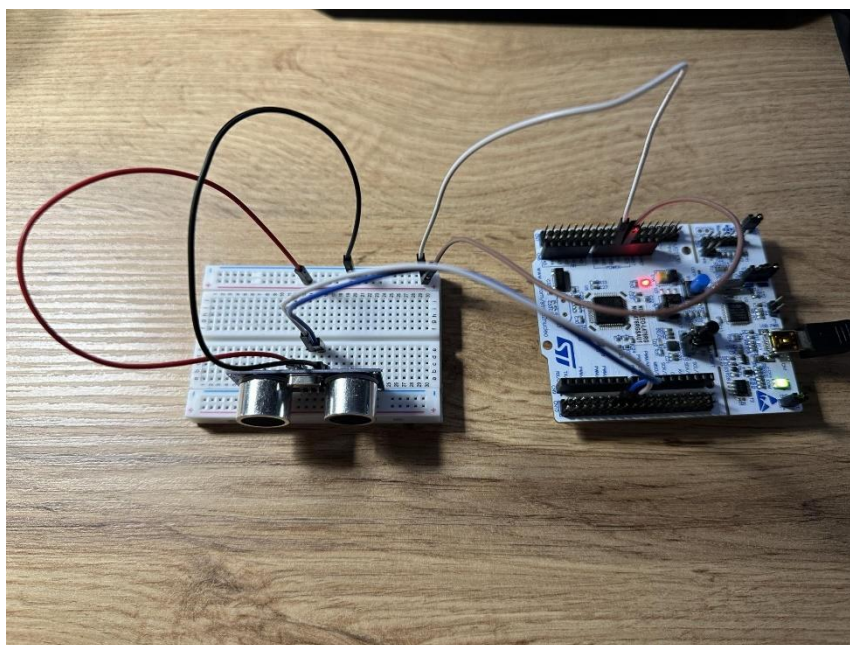


Rysunek 8 UML diagram projektu

## 4 Realizacja projektu

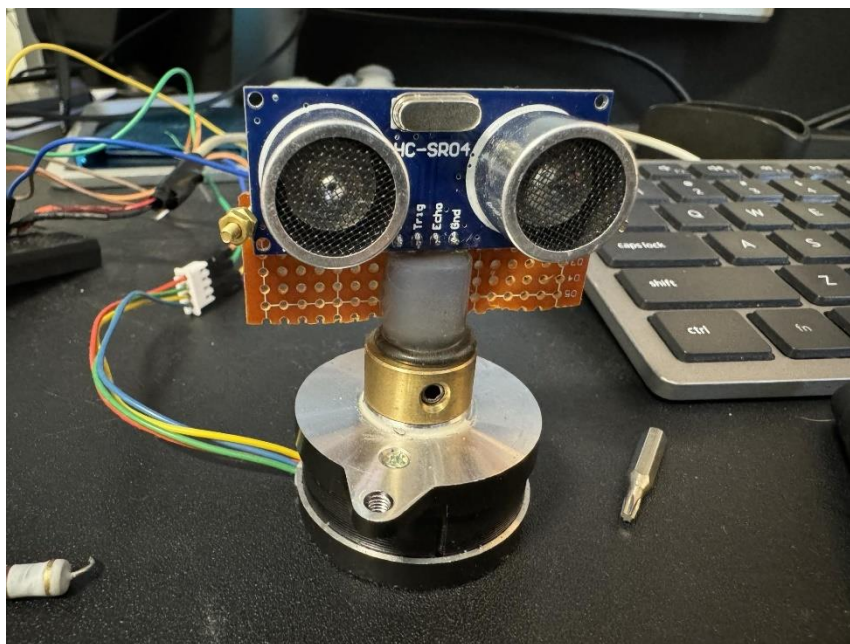
### 4.1 Realizacja krok po kroku

Pierwszym elementem, który został skonfigurowany był czujnik odległości HC-SR04. Głównym założeniem tego kroku było osiągnięcie odczytu odległości z czujnika w nieskończonej pętli. W tym celu skonfigurowany został timer, tak aby generował puls na wyjściu trwający dokładnie 10 $\mu$ s, co miało zainicjalizować pomiar w module czujnika. Następnie ten sam timer został ustawiony w tryb czuwania, tak żeby po wykryciu czasu zbocza na sygnale wyjściowym czujnika zmierzyć czas jego trwania. Ważnym aspektem tego kroku było ustawienie czasu pomiędzy kolejnymi pomiarami, tak żeby moduł zwracał poprawne nie zakłócone dane. W tym kroku pojawił się także aspekt odbierania przez mikrokontroler STM32 sygnału o napięciu 5V, ale nie było z tym żadnego problemu. Zasilanie modułu zostało także dostarczone z odpowiedniego 5V portu na płytce rozszerzeniowej NUCLEO.



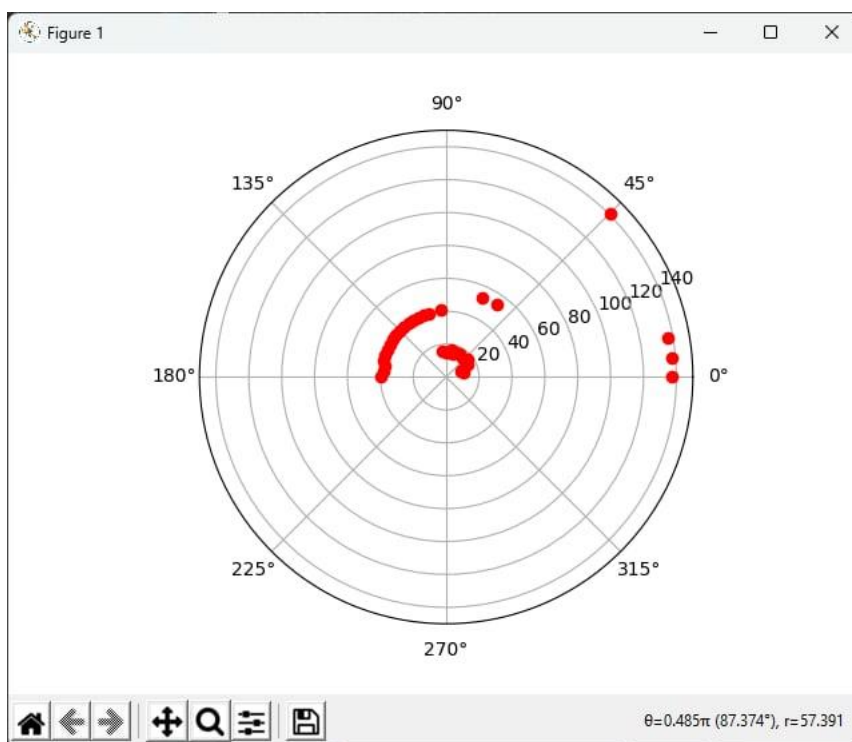
*Rysunek 9 Układ testowania czujnika odległości*

Następnym krokiem było skonfigurowanie silnika krokowego, oraz fizyczne połączenie do niego czujnika odległości. W tym celu skonfigurowany został sterownik TMC2226 tak, żeby w pętli działania czujnika co każdy pomiar wykonywać obrót o zadany kąt 5°. Nie było sensu ustawiania obrotu na mniejszy kąt, ponieważ czujnik sam w sobie jak podaje producent ma kąt z jakim mierzona jest odległość wynoszący ok 15°. Jeżeli kąt obrotu silnika dojdzie do maksymalnego odchylenia, następuje zmiana stanu na porcie odpowiadającym za zmianę kierunku obrotu. Całość została ze sobą złączona za pomocą kawałka płytki prototypowej. Ważnym aspektem było ustawienie czujnika prostopadle do podłoża, tak żeby pomiar wykonywany był pod stałym kątem i w jednej osi.



Rysunek 10 Połączenie czujnika odległości z silnikiem krokowym

Ostatnim krokiem w projekcie było stworzenie programu w języku Python, do graficznego przedstawienia działania programu. W tym punkcie także skonfigurowana została magistrala USART w mikrokontrolerze. Na samym początku program komunikuje się z mikrokontrolerem i odbiera od niego dane o aktualnym kącie i zmierzonej dla niego odległości obiektu od modułu czujnika. Za pomocą danych i biblioteki matplotlib rysowany jest animowany wykres przedstawiający działanie sonaru za pomocą wektora i zaznaczającym napotkane na drodze czujnika odległości obiekty.



Rysunek 11 Wykres przedstawiający działanie sonaru

## 4.2 Działanie skończonego projektu

Skończony projekt działa tak jak było to planowane od samego początku. Sonar skanuje po zadanym kącie obiekty w około i rysuje je na wykresie. Zmierzone wartości jak i sam kąt obrotu całego układu jest lekko niedokładny, ale zważając na koszt całego systemu, jest to jak najbardziej adekwatne.

## 4.3 Napotkane problemy

Podczas realizacji projektu zostało napotkane kilka problemów, które udało się rozwiązać w większym lub mniejszym stopniu. Podstawowym problemem od samego początku było uwzględnienie kabli wychodzących z czujnika, co uniemożliwiało obrót silnika w jednym kierunku cały czas. Spowodowałoby to uszkodzenie przewodów lub zaplątanie się całego układu. Rozwiązaniem tego problemu była zmiana kierunku obrotu po osiągnięciu wymaganego kąta.

Problem pojawił się przy czujniku odległości. W większości przypadków pomiarów zgadzają się one z rzeczywistą odległością z jaką został postawiony obiekt na przeciwko czujnika. Niestety zdarzają się miejscami odchyłki o  $\pm 2\text{cm}$ , co mogło być spowodowane nie równym kątem nachylenia czujnika względem podłoża, co mogło wpłynąć na rozchodzenie się fal dźwiękowych i w ostateczności wpłynąć na wynik końcowy. Jest to przypadłość dosyć taniego czujnika, ale nie sprawia większych problemów w kontekście działania tego projektu.

Kolejnym z nich był problem ze skonfigurowaniem silnika krokowego. Pierwsza zakładana opcja, czyli przesuwanie go o daną ilość kroków nie działała zgodnie z oczekiwaniami. Rozwiązaniem było ustawienie sterownika tak, żeby obracał silnik o zadaną ilość stopni. W praktyce silnik się obraca, ale kąt nie zgadza się dokładnie z rzeczywistym, co będzie można uwzględnić przy rysowaniu wykresu.

Z punktu widzenia mechaniki jednym z napotkanych także problemów jest ruszanie się całego silnika przy wykonywaniu kolejnych kroków. Zostało to rozwiązane za pomocą solidniejszego przytwierdzenia całego układu do podłoża, a w przyszłości jest możliwość stworzenia specjalnego mocowania.



## 5 Podsumowanie

Mimo problemów jakie pojawiły się podczas realizacji, projekt został pomyślnie ukończony. Jego działanie nie jest idealne, co może być spowodowane użyciem prostych i niedrogich komponentów, ale jest wystarczające w kontekście naukowym i eksperymentalnym. Na przyszłość zostaje spore zapas manewrów na ewentualne ulepszenie tego projektu, lub na stworzenie go od podstaw, z bardziej zaawansowanymi elementami. Projekt może być z powodzeniem wykorzystywany w różnego rodzaju pojazdach autonomicznych lub służyć do wyznaczania kształtu np. pomieszczeń.



## 6 Bibliografia

1. Cyganek B. *Programowanie w języku C++*. Wprowadzenie dla inżynierów  
Wydawnictwo Naukowe PWN, 2023,
2. Moduł HC-SR04 instrukcja obsługi: <https://web.eece.maine.edu/zhu/book/lab/HC-SR04%20User%20Manual.pdf>,
3. Sterownik silnika TMC2226 instrukcja obsługi: [https://robu.in/wp-content/uploads/2023/05/TMC2226\\_V1.0\\_Instruction\\_Manual.pdf](https://robu.in/wp-content/uploads/2023/05/TMC2226_V1.0_Instruction_Manual.pdf)
4. <https://www.stm32wrobotyce.pl/2021/03/24/silnik-krokowy/>,
5. <https://www.youtube.com/watch?v=xRnO4ktnMMY>,
6. <https://www.youtube.com/watch?v=dZ4Bjmm7fLk>,