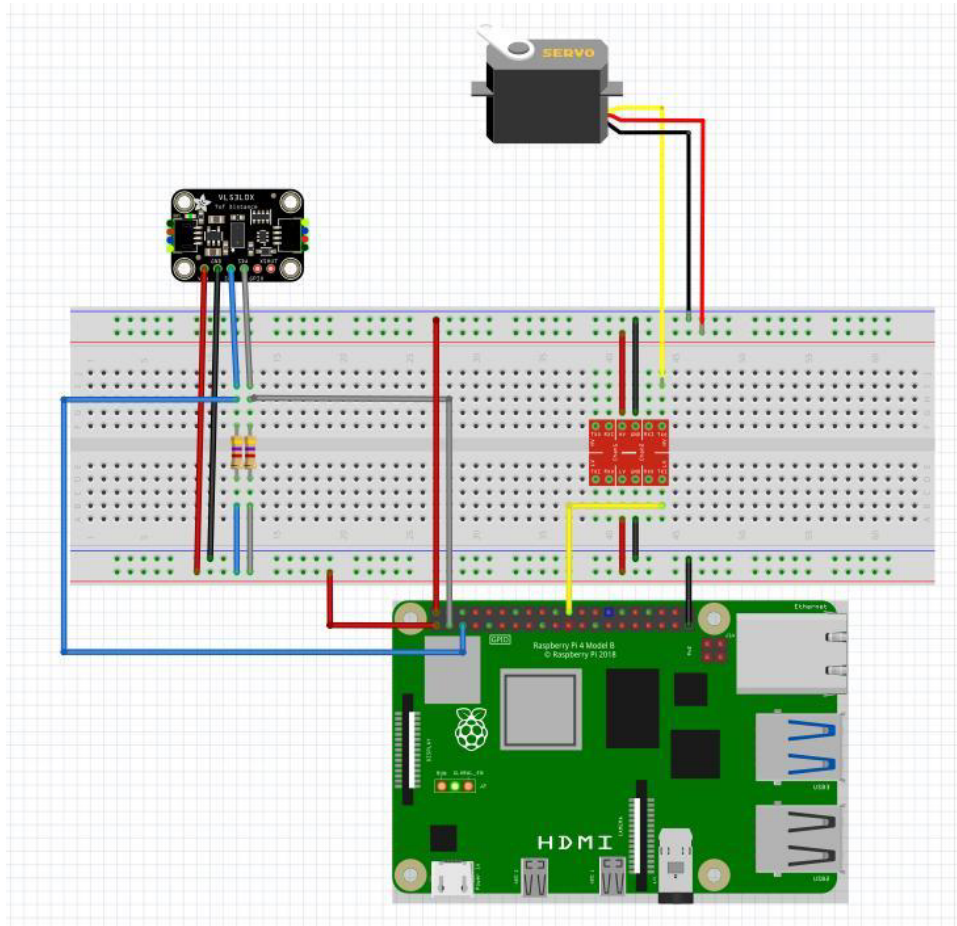


SKPS – Projekt

1. Temat projektu

Lidar na bazie czujnika odległości ToF i serwomechanizmu

2. Schemat zaprojektowanego układu

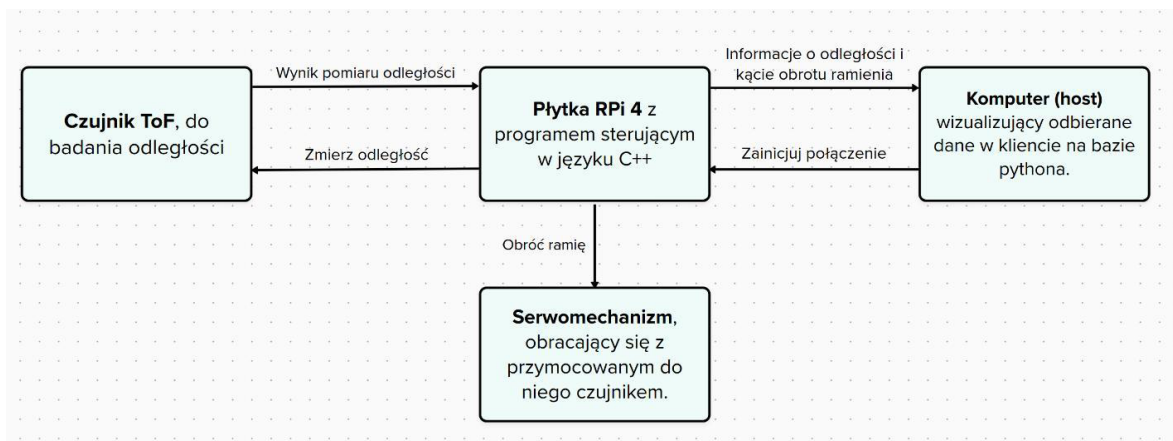


Zaprojektowany układ składa się z:

- Płytki RaspberryPi4
- Serwomechanizmu,
- Czujnika Odległości: Adafruit VL53L0X ToF Distance Sensor
- Konwertera poziomów logicznych 3,3V/5V

Oporniki 4.7k Ω , które widoczne są na powyższej grafice nie były w praktyce potrzebne przy czujniku, z którego korzystaliśmy, bo ten miał już wbudowane pull-upy.

3. Sposób działania



Powyżej widoczny jest uproszczony schemat działania naszego układu. Za sterowanie serwomechanizmem i czujnikiem odległości odpowiedzialny jest program napisany w C++ i zainstalowany na OpenWRT znajdującym się na płytce. Po wykonaniu pomiaru i obróceniu serwomechanizmu o zadany kąt, program wysłał poprzez gniazdo TCP (w tym przypadku na porcie 8888) informacje o zmierzonej odległości i kącie obrotu ramienia. Program klienta znajdujący się na komputerze odbierał dane z gniazda TCP i na ich podstawie w czasie rzeczywistym wizualizował pomiary wykonywane przez układ. Został on napisany w pythonie z wykorzystaniem biblioteki matplotlib.

Program, znajdujący się na płytce, steruje serwomechanizmem przy użyciu sygnału PWM, przy obecnie dobranych wartościach częstotliwości oraz minimalnego i maksymalnego wypełnienia, serwomechanizm obraca się w zakresie 0 - 180 stopni.

Do sterowania pomiarem odległości z użyciem czujnika ToF wykorzystaliśmy systemową magistralę I2C, a do inicjalizacji czujnika posłużył nam zmodyfikowany kod udostępniony w bibliotece obsługującej czujnik. Wynik pomiaru był zwracany w milimetrach.

4. Wymagania i sposób uruchomienia

Do pakietu OpenWRT niezbędne było dodanie bibliotek odpowiedzialnych za obsługę czujnika ToF:

VL53LOX_RPI

VL53LOX_API

Do systemu OpenWRT na RaspberryPi należy zainstalować pakiety:

kmod-i2c-bcm2835

kmod-pwm-bcm2835

A w pliku boot/config.txt na RaspberryPi dodać linie:

dtoverlay=i2c1

dtoverlay=pwm-2chan

Do odpalenia programu klienta niezbędne są:

Python w wersji 3.9+

matplotlib w wersji 3.6+

Uruchamianie programu sterującego - powinien zostać odpalony przed klientem:

lidar [krok_obrotu] [okres próbkowania (us)]

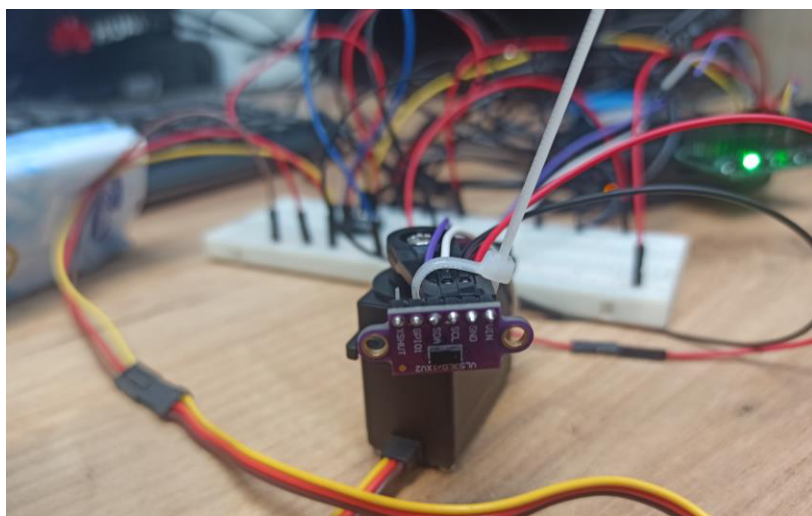
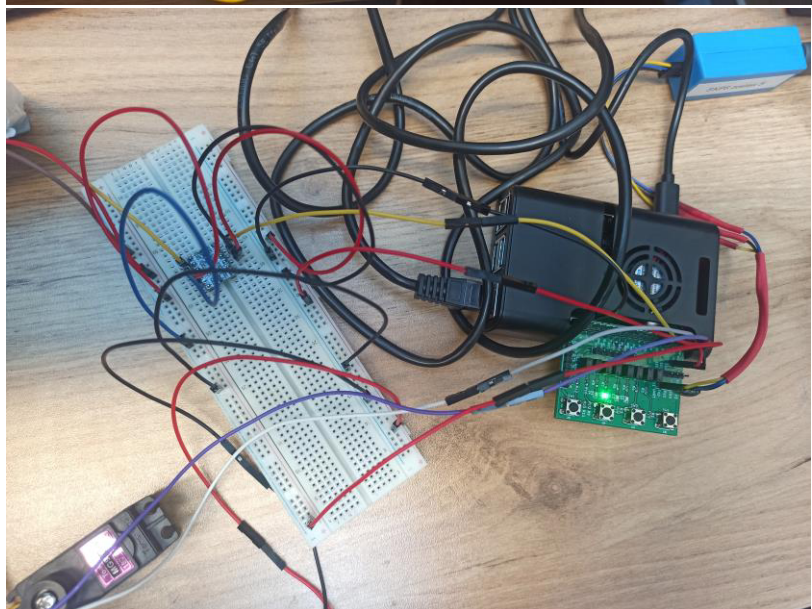
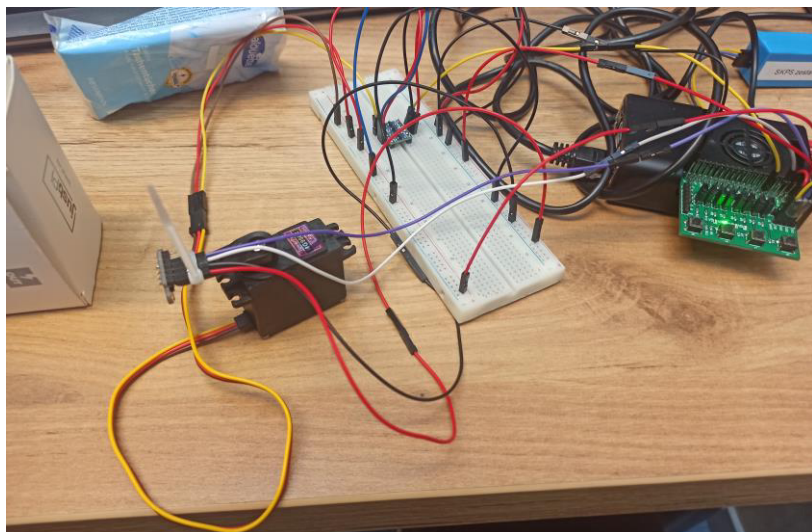
Uruchamianie programu klienta:

python3 client.py

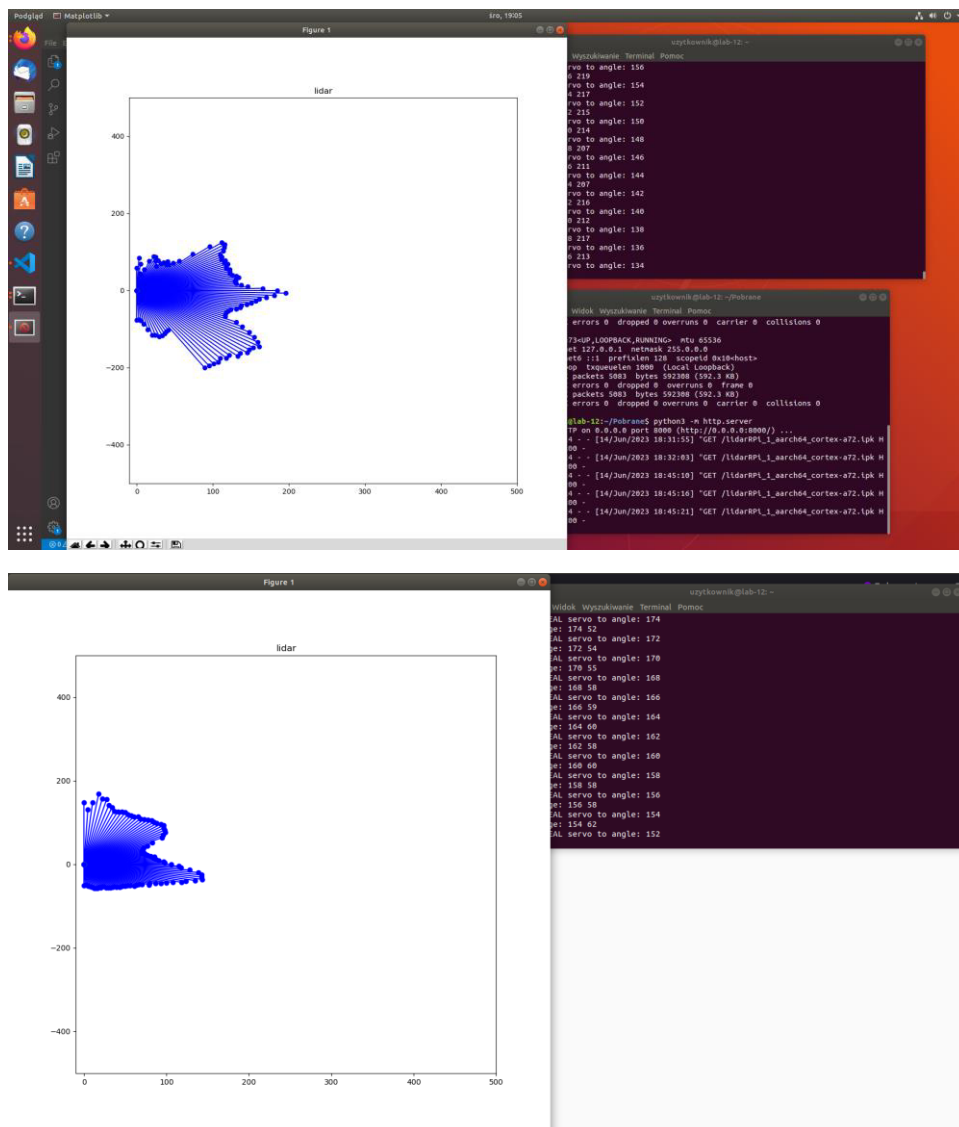
W przypadku chęci skorzystania z wersji symulacyjnej na Qemu, należy zmienić adres (domyślnie ustawiony na adres wykorzystywanej RPi4 - 10.42.0.124) na 127.0.0.1 oraz port (z domyślnego 8888) na 9888.

5. Realizacja i wyniki

Podłączony układ prezentował się następująco:



Program klienta wizualizuje dane w następujący sposób:



Udało się zbudować lidar, który działał w zadowalający sposób mierząc odległości obiektów znajdujących się w obrębie swojego otoczenia, a dane, które z niego uzyskujemy wizualizowane są w przejrzysty sposób na ekranie komputera. Czujnik odległości radzi sobie dość dobrze z obiektami w bliskiej odległości, ale z tymi znajdującymi się dalej (co jest obserwowane już mniej więcej powyżej 1 metra) ma problemy i zdarza się, że nie udaje się wykonać poprawnego pomiaru. Serwomechanizm na ogół też działa poprawnie, chociaż zdarzyło mu się zablokować w jednej pozycji na kilka sekund.