

# Systemy Komputerowe w Sterowaniu i Pomiarach

Projekt – „Lidar na bazie czujnika odległości ToF i serwomechanizmu”

Aleksander Kruk

Michał Sobiech

17 stycznia 2024

## Struktura projektu


Projekt składa się z serwera napisanego w C++ działającego na płycie rozgłaszającego dane na podanym porcie oraz z klienta zaimplementowanego w pythonie wizualizującego otrzymane dane z lidaru.

Po uruchomieniu serwer oczekuje na wiadomość początkową, którą w naszym przypadku jest znak `START OF TEXT`. Po otrzymaniu wiadomości początkowej serwer aktywuje lidar zaczyna przysyłać odczytane dane o odległości i kącie jako dwie liczby 16-bitowe. Lidar dokonuje pełnej rotacji, a następnie serwer informuje klienta o końcu danych, poprzez wysłanie wiadomości końcowej zawierającej maksymalne wartości obu integerów, czyli `0xFFFF|0xFFFF`.

Klient na bieżąco wizualizuje odczytane punkty. Po odebraniu wszystkich punktów klient kończy odbiór, a wykres pozostaje do analizy bądź zapisu.

Do interakcji z czujnikiem odległości wykrozystaliśmy API `VL53L0X_rasp` – [https://github.com/cassou/VL53L0X\\_rasp](https://github.com/cassou/VL53L0X_rasp). Do interakcji z serwowmotorem wykorzystaliśmy sterowanie `pwm`.

Zarówno praca domowa jak i projekt zostały przygotowane jako pakiety OWRT (część serwerowa).



communication.png

## Przygotowanie płytki

Aby móc prowadzić interakcje z urządzeniami potrzebowaliśmy, pobrać odpowiednie rozszerzenia płytki. Dla `i2c` musieliśmy zainstalować rozszerzenia `kmod-i2c-bcm2835`, natomiast dla `pwm` zainstalowaliśmy `kmod-pwm-bcm2835`. Dodatkowo potrzebowaliśmy zainstalować rozszerzenie `libstdcpp6` aby skompilować projekt.

Do pliku konfiguracyjnego `/boot/config.txt` dodaliśmy następujące linijki: `dtoverlay=i2c1` i `dtoverlay=pwm-2chan`.

## Efekty pracy


Na laboratorium udało nam się bez problemów połączyć z czujnikiem odległości, natomiast nie byliśmy w stanie skorzystać z wbudowanego `pwm-a` pinu raspberry pi.

Po analizie kodu w domu, uświadomiliśmy sobie, że chociaż napisaliśmy funkcje do eksportu i uneksportu kanału PWM, to nie wywołałyśmy jej w kodzie... Jesteśmy pewni, że gdybyśmy wywołały nasze funkcje, to program by zadziałał.

Ponieważ przeoczyliśmy brak eksportów, to postanowiliśmy zaimplementować własne PWM, ale nie udało nam się go przetestować przed końcem laboratorium.

Udało nam się wygenerować wykres dla czujnika, ale zapomnieliśmy o pobraniu jego screena z komputera laboratoryjnego... Ponieważ klient wizualizacyjny nie zmienił się względem pracy domowej, to powstający wykres różnił się jedynie tym, że pomiary były prawdziwe, a nie losowe.

## Schemat połączeń




schemat.png

## Dokumentacja połączeń


---


Niestety zdjęcie schematu połączeń zrobione zostało po koniec laboratorium w trakcie rozbiórki, więc zdążyliśmy już odłączyć czujnik odległości.


 images/PROJ\_cables.png


## Dokumentacja użytych pakietów

---

 images/PROJ\_package1.jpg

 images/PROJ\_package2.jpg

 images/PROJ\_package3.jpg

 images/PROJ\_package4.jpg