

PROJEKT

WIZUALIZACJA DANYCH SENSORYCZNYCH

Wizualizacja Sensorów Line Follower'a

Michał Maćkowiak, 249464



Prowadzący:
dr inż. Bogdan Kreczmer

Katedra Cybernetyki i Robotyki
Wydziału Elektroniki
Politechniki Wrocławskiej

23 marca 2021

Spis treści

1	Cel projektu	1
2	Przewidywane efekty końcowe	1
3	Podcele i etapy realizacji projektu	2
4	Terminarz realizacji poszczególnych podcelów (z dokładnością do 1 tygodnia)	2
5	Efekty przeglądu materiałów związanych z projektem	4

1 Cel projektu

Celem projektu jest stworzenie aplikacji za pomocą biblioteki *Qt* w języku *C++*, która będzie wizualizować dane robota mobilnego klasy *line follower*. Robot wyposażony w enkodery i czujniki odbiciowe będzie przysyłał dane, które posłużą wyświetlaniu trasy przejechanej przez robota, przedstawieniu prędkości oraz odczytu czujników odbiciowych *line follower*'a.

Trasa przejechana będzie zaznaczana punktowo na podstawie położenia robota w przestrzeni dwuwymiarowej, mając informacje z enkoderów zawartych na kołach robota. Dodatkowo enkodery pozwolą przedstawić, z jaką prędkością się porusza.

Czujniki odbiciowe służą detekcji linii, którą ten *line follower*, jak nazwa wskazuje, śledzi. Przedstawiony zostanie stan tych czujników, a zatem informacja czy czujnik wykrywa linię, czy nie.

Dodatkowe informacje o konstrukcji robota mobilnego:

- Wszystkie komponenty robota będą przylutowane lub przymocowane do zaprojektowanej płytki drukowanej.
- Zastosowany będzie mikrokontroler *STM*.
- Będzie posiadać co najmniej 5 czujników odbiciowych. Czujniki odbiciowe wykorzystują do działania diodę emitującą promieniowanie oraz fototranzystor odbierający światło odbite. W ten sposób w zależności od właściwości refleksyjnych i pochłaniających materiału oświetlanego na fototranzystorze emitowane jest napięcie proporcjonalne do ilości otrzymanego światła. Stąd też czarna linia (która pochłania światło) jest odbierana przez mikrokontroler jako sygnał niski natomiast biała przestrzeń toru (odbijająca światło) jako sygnał wysoki.

2 Przewidywane efekty końcowe

- Aplikacja będzie posiadać prosty interfejs użytkownika.
- Komunikacja z robotem, a aplikacją będzie odbywać się poprzez połączenie *bluetooth* 4.0.
- Położenie robota w przestrzeni będzie odwzorowane z dokładnością do 1 *cm*. Okresowo, co 30 *sekund*, rysowane będą punkty reprezentujące aktualne położenie robota w przestrzeni 2D.
- Będzie wyświetlana aktualna prędkość kątowna kół robota z dokładnością do 1 *rpm*. Graficznie prędkość będzie przedstawiona przez słupki, które będą rosły wraz ze zwiększaniem prędkości oraz malały ze zmniejszaniem jej. Dodatkowo będą zawierać informacje o kierunku jazdy robota.
- Sygnały z czujników będą posiadać interfejs graficzny przypisany do każdego z 5 czujników. Interfejs będzie wyświetlać kolor biały lub czarny w zależności czy wykrywana jest linia, czy też nie.

3 Podcele i etapy realizacji projektu

W tym rozdziale znajdują się wyszczególnione podcele oraz etapy tworzenia projektu. Każdy etap uznawany jest za zakończony po wykonaniu serii testów oraz ewentualnych poprawek.

Lista podcelów:

1. Zapoznanie się z dokumentacją rozwiązań sprzętowych zastosowanych w projekcie. Pozyskanie dodatkowej wiedzy śledząc artykuły oraz inne zasoby Internetu
2. Projekt interfejsu graficznego
3. Postęp bez gotowego robota:
 - (a) Wykonanie pierwszej wersji aplikacji posiadającej pojedynczy widżet
 - (b) Zaimplementowanie komunikacji przez *UART*, by otrzymywać dane z mikrokontrolera
 - (c) Ręczne symulowanie poruszania silników oraz odczytywanie „surowego” stanu enkoderów dzięki podłączeniu silniki z enkoderami do zewnętrznej płytki deweloperskiej *STM32F429I-DISCO1*
 - (d) Symulacyjne pobieranie danych z czujnika/ów odbiciowych
 - (e) Przetworzenie danych, by otrzymać prędkość kątową kół oraz współrzędne przestrzeni *2D* znając prędkość i kierunek obrotu kół
 - (f) Stworzenie graficznej reprezentacji danych w aplikacji - wyświetlanie trasy przejechanej, aktualnej prędkości oraz stanu czujników odbiciowych
4. Zmontowanie robota (zlutowanie i podłączenie) oraz wstępne testy poprawności jego działania. W tym podłączenie czujników
5. Pełna integracja robota mobilnego z aplikacją. Testy oraz zmiany konstrukcyjne/programowe.
6. Ostateczne sprawdzenie poprawności działania aplikacji przy współpracy z robotem. Skończenie dokumentacji projektu.

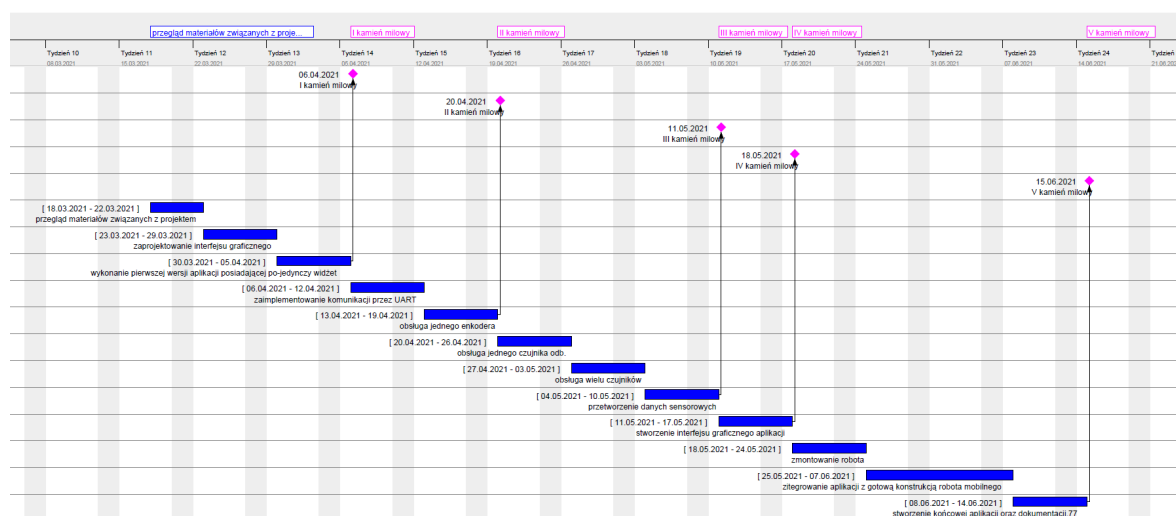
4 Terminarz realizacji poszczególnych podcelów (z dokładnością do 1 tygodnia)

Poniżej znajduje się harmonogram realizacji projektu. Flagą **KM** zaznaczone są kamienie milowe postępu wykonywania projektu. Data przy, każdym z zadań oznacza datę ukończenia podcelu.

- 22 marca 2021 – zakończenie przeglądu materiałów związanych z projektem
- 29 marca 2021 – zaprojektowanie interfejsu graficznego
- **KM 5 kwietnia 2021** – wykonanie pierwszej wersji aplikacji posiadającej pojedynczy widżet
- 12 kwietnia 2021 – zaimplementowanie komunikacji przez *UART*

- **KM 19 kwietnia 2021** – stworzenie interfejsu tekstowego do obsługi „suro-
wych” danych z jednego enkodera
- 26 kwietnia 2021 – zaimplementowanie obsługi jednego czujnika odbiciowego.
Testy na interfejsie tekstowym
- 4 maja 2021 – obsługa pięciu czujników odbiciowych oraz dwóch enkoderów
- **KM 10 maja 2021** – przetworzenie danych, by otrzymać prędkość kątową kół
oraz współrzędne przestrzeni 2D znając prędkość i kierunek obrotu kół
- **KM 17 maja 2021** – stworzenie interfejsu graficznego aplikacji
- 24 maja 2021 – zmontowanie robota
- 31 maja 2021 – zintegrowanie aplikacji z gotową konstrukcją robota mobilnego
- 7 czerwca 2021 – dokończenie poprzedniego punktu
- **KM 14 czerwca 2021** – stworzenie końcowej aplikacji oraz dokumentacji.

Poniżej na rysunku 1 znajduje się diagram Gantta przedstawiający harmonogram pracy.



Rysunek 1: Diagram Gantta

5 Efekty przeglądu materiałów związanych z projektem

Zapoznanie się z biblioteką oraz obsługą *Qt* w języku *C++*. [1] [2] [3] [4] [5] [6]

Zapoznanie się z programowaniem mikrokontrolerów STM. [7] [8] [9]

Pozyskanie wiedzy na temat portu szeregowego *UART*, jego implementacji i działania. [10] [11]

Literatura

- [1] Mateusz Patyk. Kurs qt. *forbot.pl*.
- [2] Jasmin Blanchette and Mark Summerfield. *C++ GUI Programming with Qt 4*. <http://www.qtrac.eu/C++-GUI-Programming-with-Qt-4-1st-ed.zip>, 2006.
- [3] Dan Munteanu. Robust qt c++ gui programming 2d graphics app tutorial. *udemy.com*, 2020.
- [4] Bryan Cairns. Qt 5 core for beginners with c++. *udemy.com*, 2020.
- [5] Bryan Cairns. Qt 5 core intermediate with c++. *udemy.com*, 2020.
- [6] Bryan Cairns. Qt 5 core advanced with c++. *udemy.com*, 2020.
- [7] Marek Galewski. *STM32. Aplikacje i ćwiczenia w języku C z biblioteką HAL*. BTC, 2019.
- [8] C. Noviello. *Mastering the STM32 Microcontroller*. LeanPub, May 2016.
- [9] Bartek Kurosz. Kurs stm32 f4. *forbot.pl*.
- [10] Kamami. Stm32cube w przykładach (usart). *stm32.com*.
- [11] Bartek Kurosz. Komunikacja przez uart. *forbot.pl*.