

PROJEKT

WIZUALIZACJA DANYCH SENSORYCZNYCH

Wizualizacja danych sensorycznych z robota typu „LineFollower”

Kamil Winnicki, 263434



Prowadzący:
dr inż. Bogdan Kreczmer

Katedra Cybernetyki i Robotyki
Wydziału Elektroniki, Fotoniki i
Mikrosystemów
Politechniki Wrocławskiej

16 maja 2024

Spis treści

| | | |
|----------|--|----------|
| 1 | Charakterystyka tematu projektu | 1 |
| 2 | Podcele i etapy realizacji projektu | 1 |
| 3 | Specyfikacja finalnego produktu | 2 |
| 4 | Terminarz realizacji poszczególnych podcelów (z dokładnością do 1 tygodnia) | 2 |
| 5 | Wykres Gantta z harmonogramem projektu | 3 |
| 6 | Projekt graficznego interfejsu użytkownika | 3 |
| 6.1 | Wizualizacja robota | 4 |
| 6.2 | Kompas | 4 |
| 6.3 | Przyciski | 4 |
| 6.4 | Statystyki | 4 |
| 6.5 | Ustawianie parametrów | 4 |
| 7 | Opis ramki danych | 5 |
| 7.1 | Komunikacja z robota do aplikacji | 5 |
| 7.2 | Komunikacja z aplikacji do robota | 6 |
| 8 | Opis aktualnych postępów nad aplikacją | 7 |
| 8.1 | Aktualny wygląd aplikacji | 7 |

1 Charakterystyka tematu projektu

Projekt zakłada stworzenie aplikacji wizualizującej dane sensoryczne dla robota typu linefollower, czyli robota którego celem jest śledzenie linii. Aplikacja ta ma na celu monitorowanie i analizowanie zachowań robota w czasie rzeczywistym. Głównym celem jest spowodowanie że testowanie i analiza zachowań takiego robota będzie znacznie prostsza. Robot wyposażony będzie w:

- czujniki odbiciowe,
- enkodery na kołach,
- żyroskop i akcelerometr

Sterowanie robota oparte będzie o mikrokontroler ESP32 z anteną bluetooth co pozwoli na zdalny przesył danych sensorycznych do aplikacji.

2 Podcele i etapy realizacji projektu

Praca nad realizacją projektu podzielona będzie na poniższe etapy:

- utworzenie bazowej aplikacji,
- utworzenie zdalnej komunikacji aplikacji z mikrokontrolerem ESP32 poprzez Bluetooth,
- wyświetlenie danych z czujników użytych w robocie,
- ostateczne testy i dopracowanie aplikacji,

3 Specyfikacja finalnego produktu

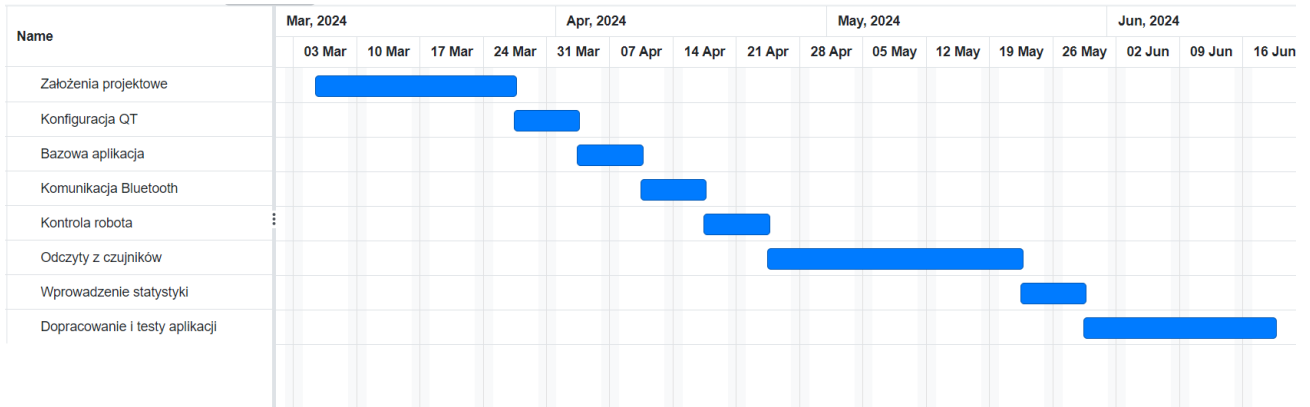
Aplikacja ma działać następująco:

- komunikować się z robotem w dwie strony poprzez protokół Bluetooth HCI UART,
- pozwalać na zdalne wystartowanie oraz zatrzymanie robota, a także na zmianę parametrów początkowych takich jak np: maksymalna prędkość, wzmocnienie parametrów regulatora PID,
- wyświetlać aktualną pozycję linii pod robotem na podstawie cyfrowych odczytów z czujników odbiciowych,
- wyświetlać poziom naładowania baterii (graficznie, procentowo oraz aktualne napięcie) i wyświetlać ostrzeżenie o jej rozładowaniu,
- graficznie wyświetlać aktualny obrót robota względem startu na podstawie odczytów z żyroskopu,
- wyświetlać prędkość kątową oraz procentowo wypełnienie PWM na danym kole, przy czym graficznie prezentować obroty,
- prowadzić statystykę przejazdu robota (przebyty dystans, prędkość, czas przejazdu),

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

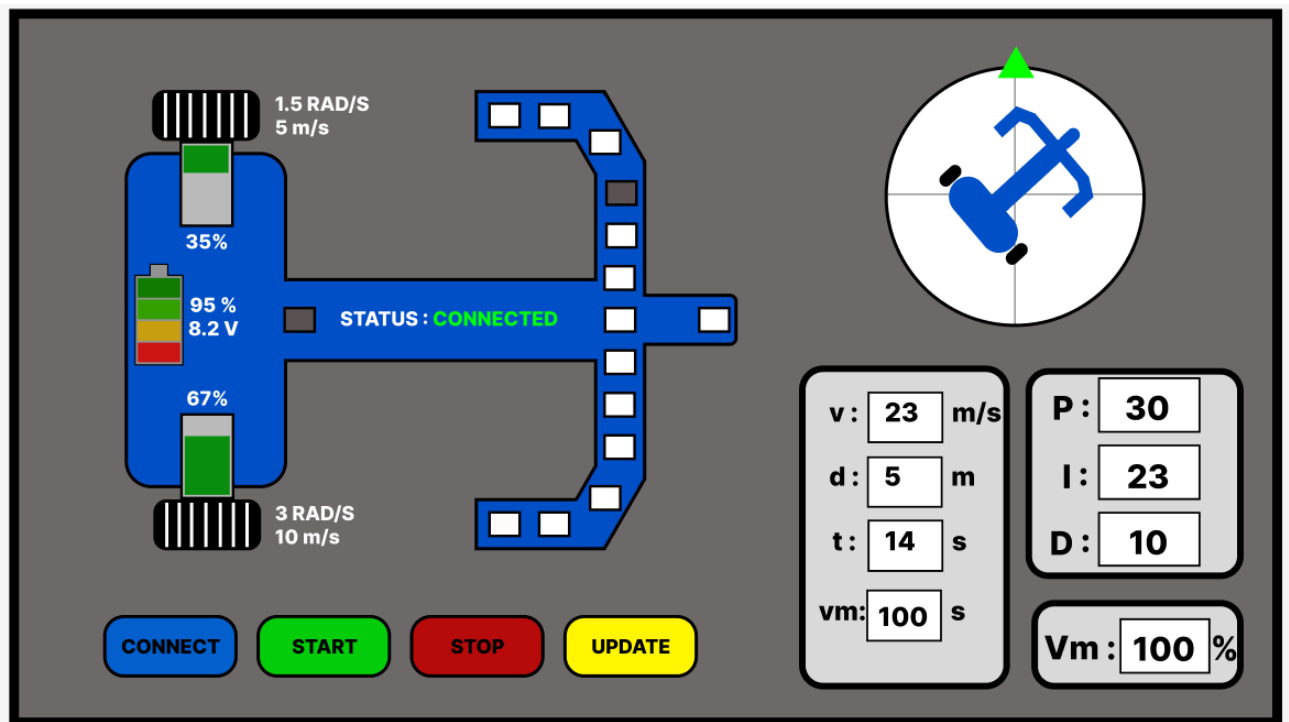
- 20 marca 2024 – Ustalenie założeń projektowych
- 27 marca 2024 – Utworzenie projektu graficznego aplikacji
- 3 kwietnia 2024 – Przeprowadzenie wstępnej konfiguracji Qt
- 10 kwietnia 2024 – Utworzenie bazowej aplikacji
- 17 kwietnia 2024 – Komunikacja mikrokontrolera z aplikacją poprzez Bluetooth
- 24 kwietnia 2024 – Implementacja elementów graficznych do kontroli robota
- 1 maja 2024 – Wyświetlanie poziomu naładowania baterii
- 8 maja 2024 – Wyświetlanie odczytów z czujników podłoża
- 15 maja 2024 – Wyświetlenie odczytów z enkoderów
- 22 maja 2024 – Wyświetlenie aktualnego obrotu robota
- 29 maja 2024 – Wprowadzenie statystyki przejazdów
- 5 czerwca 2024 – Implementacja finalnej wersji aplikacji
- 12 czerwca 2024 – Przeprowadzenie ostatecznych testów
- 19 czerwca 2024 – Oddanie gotowej aplikacji

5 Wykres Gantta z harmonogramem projektu



Rysunek 1: Harmonogram Gantta

6 Projekt graficznego interfejsu użytkownika



Rysunek 2: Projekt graficzny

W aplikacji wyodrębnione zostały poniższe sekcje:

6.1 Wizualizacja robota

Wizualizacja została przedstawiona jako rysunek rzeczywistego robota. W trakcie przejazdu robota na trasie wyświetlane będą w czasie rzeczywistym odczyty z czujników podłoża, wypełnienie PWM na silnikach, prędkość kątowna i liniowa kół, napięcie na baterii oraz status połączenia.

6.2 Kompas

W trakcie przejazdu wyświetlony będzie aktualny kąt obrotu robota względem pozycji startowej.

6.3 Przyciski

Aplikacja posiadać będzie 4 przyciski:

- „CONNECT” - do połączenia aplikacji z robotem,
- „START” - do rozpoczęcia przejazdu na trasie,
- „STOP” - do zatrzymania robota,
- „UPDATE” - do zaktualizowania parametrów,

6.4 Statystyki

Sekcja ta zawierać będzie statystyki aktualnego przejazdu takie jak: aktualna prędkość, przejechany dystans, czas przejazdu oraz prędkość maksymalna.

6.5 Ustawianie parametrów

Sekcja ta pozwalać będzie użytkownikowi na zaaktualizowanie parametrów regulatora PID oraz maksymalnego wypełnienia PWM.

7 Opis ramki danych

Komunikacja odbywać się będzie poprzez Bluetooth za pośrednictwem biblioteki BluetoothSerial, która pozwala na używanie bluetooth jako portu szeregowego. Protokół ten ma już wbudowane mechanizmy typu sprawdzenie sumy kontrolnej, więc samodzielne implementowanie tego typu zabezpieczeń jest zbędne.

Dane pomiędzy aplikacją a robotem przesyłane będą jako ciąg znaków (zmienna String), o określonym ułożeniu, tworzącym ramkę danych.

7.1 Komunikacja z robota do aplikacji

W przypadku komunikacji z robota do aplikacji, ramka danych będzie miała poniższy schemat:

| nazwa | START | ST | S | SNS | S | PWM_L | S | PWM_R | S | w_L | S | w_R | S | Z_ROT | S | BATT | STOP |
|----------|-------|-----|---|-----|---|-------|---|-------|---|--------|---|--------|---|-------|---|--------|------|
| n znaków | 1 | 1 | 1 | 20 | 1 | 3 | 1 | 3 | 1 | 4 | 1 | 4 | 1 | 3 | 1 | 4 | 1 |
| znak | \$ | 0/1 | x | 0-1 | x | 0-100 | x | 0-100 | x | 0-9999 | x | 0-9999 | x | 0-359 | x | 0-4096 | # |

Ramka danych rozpoczynać się będzie od znaku "\$", a kończyć znakiem "#". Wszystkie dane oddzielone od siebie będą poprzez znak – separator "x". opis przesyłanych danych:

- ST – stan w jakim robot aktualnie się znajduje: "0" gdy stoi, "1" gdy się porusza,
- SNS – odczyty z czujników linii w postaci ciągu liczb: "0" lub "1", gdzie 1 oznacza wykrycie linii na określonym czujniku,
- PWM_L – procentowe wypełnienie PWM na lewym silniku,
- PWM_R – procentowe wypełnienie PWM na prawym silniku,
- w_L – prędkość kątowna na lewym kole w rad/s,
- w_R – prędkość kątowna na lewym kole w rad/s,
- Z_ROT – rotacja robota względem osi Z w kątach,
- BATT – napięcie na baterii jako bezpośredni odczyt z 12-bitowego ADC,

7.2 Komunikacja z aplikacji do robota

W przypadku komunikacji z aplikacji do robota, ramka danych będzie miała poniższy schemat:

| nazwa | START | INS | S | PID_P | S | PID_I | S | PID_D | S | V_MAX | STOP |
|----------|-------|-------|---|-------|---|-------|---|-------|---|-------|------|
| n znaków | 1 | 1 | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 | 1 |
| znak | \$ | S,M,U | x | 0-999 | x | 0-999 | x | 0-999 | x | 0-100 | # |

Podobnie jak w poprzednim punkcie, ramka danych rozpoczynać się będzie od znaku "\$", a kończyć znakiem "#". Wszystkie dane oddzielone od siebie będą poprzez znak – separator "x". opis przesyłanych danych:

- INS - instrukcja do wykonania przez robota:
 - S – start jazdy,
 - M – stop jazdy,
 - U – aktualizacja parametrów,
- PID_P – wartość wzmocnienia członu proporcjonalnego,
- PID_I – wartość wzmocnienia członu całkującego,
- PID_D – wartość wzmocnienia członu różniczkującego,
- V_MAX – współczynnik maksymalnej prędkości w procentach,

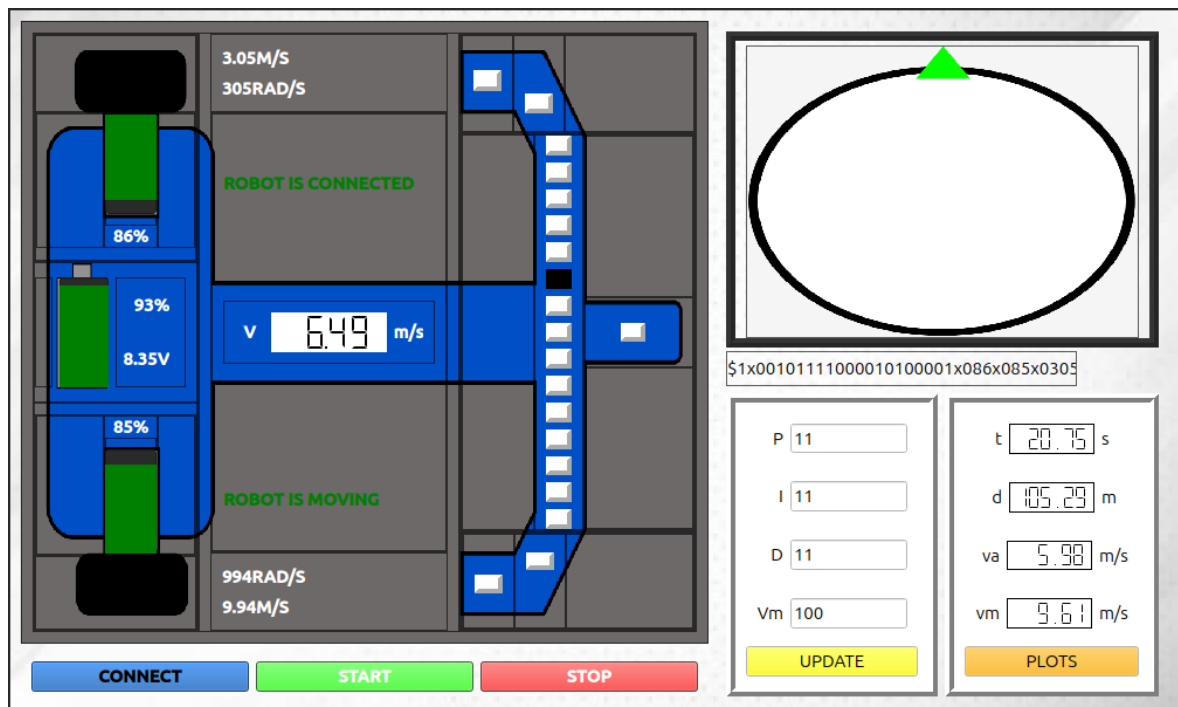
W przypadku otrzymania przez robota instrukcji startu lub stopu, pozostałe dane z ramki zostaną pominięte.

8 Opis aktualnych postępów nad aplikacją

Na obecnym etapie prac nad aplikacją udało się zrealizować zgodnie z harmonogramem następujące funkcjonalności:

- komunikacja w dwie strony za poprzez port szeregowy oraz bluetooth, wyświetlenie statusu połączenia,
- przyciski do zdalnego startowania robota, zatrzymywania oraz pola do wpisania wartości parametrów służących do kontroli robota (wzmocnienia regulatora PID oraz maksymalne wypełnienie PWM) wraz z przyciskiem do ich zaaktualizowania,
- wyświetlanie poziomu naładowania baterii w postaci graficznej oraz tekstowo napięcie wraz z procentowym naładowaniem,
- wyświetlenie odczytów z czujników podłoża,
- wyświetlenie wypełnienia PWM dla każdego z kół w postaci graficznej oraz liczbowej,
- wyświetlenie odczytów z enkoderów w postaci prędkości kątowej oraz liniowej dla każdego z kół, oraz przeliczenie tych pomiarów na prędkość liniową robota i jej wyświetlenie.
- zaimplementowanie statystyki aktualnego przejazdu: liczenie czasu, pokonanego dystansu średniej oraz maksymalnej prędkości.

8.1 Aktualny wygląd aplikacji



Rysunek 3: Aktualny wygląd aplikacji