

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

23 czerwca 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	Rezultaty Końcowe – Opis Działania Aplikacji	7
8.1	Główne Okno Aplikacji	8
8.2	Okno z Wykresami	9
9	Podsumowanie	10

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:

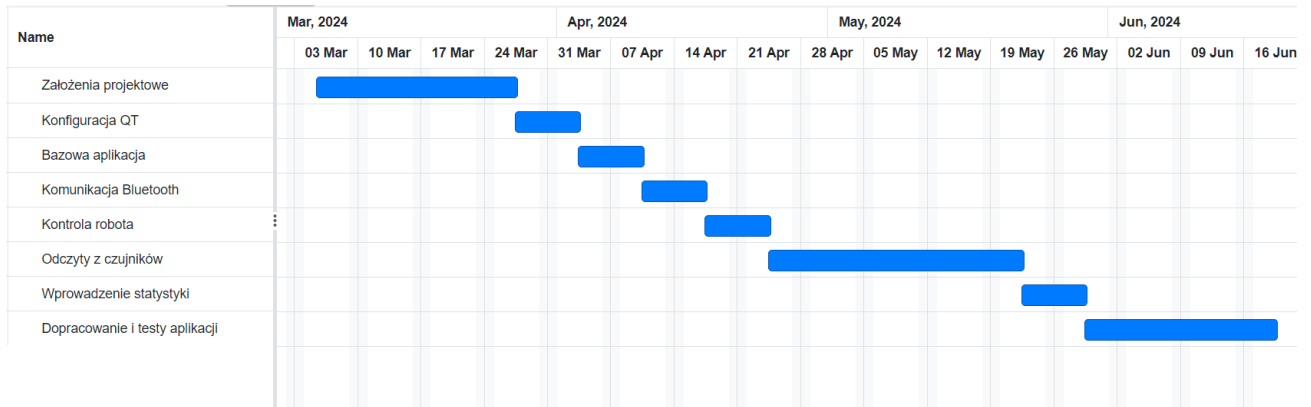
- komunikować się z robotem w dwie strony poprzez protokół Bluetooth HCI UART,
- pozwalać na zdalne wystartowanie oraz zatrzymanie robota, a także na zmianę parametrów początkowych takich jak np: maksymalna prędkość, wzmocnienie parametrów regulatora PID,
- wyświetlać aktualną pozycję linii pod robotem 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

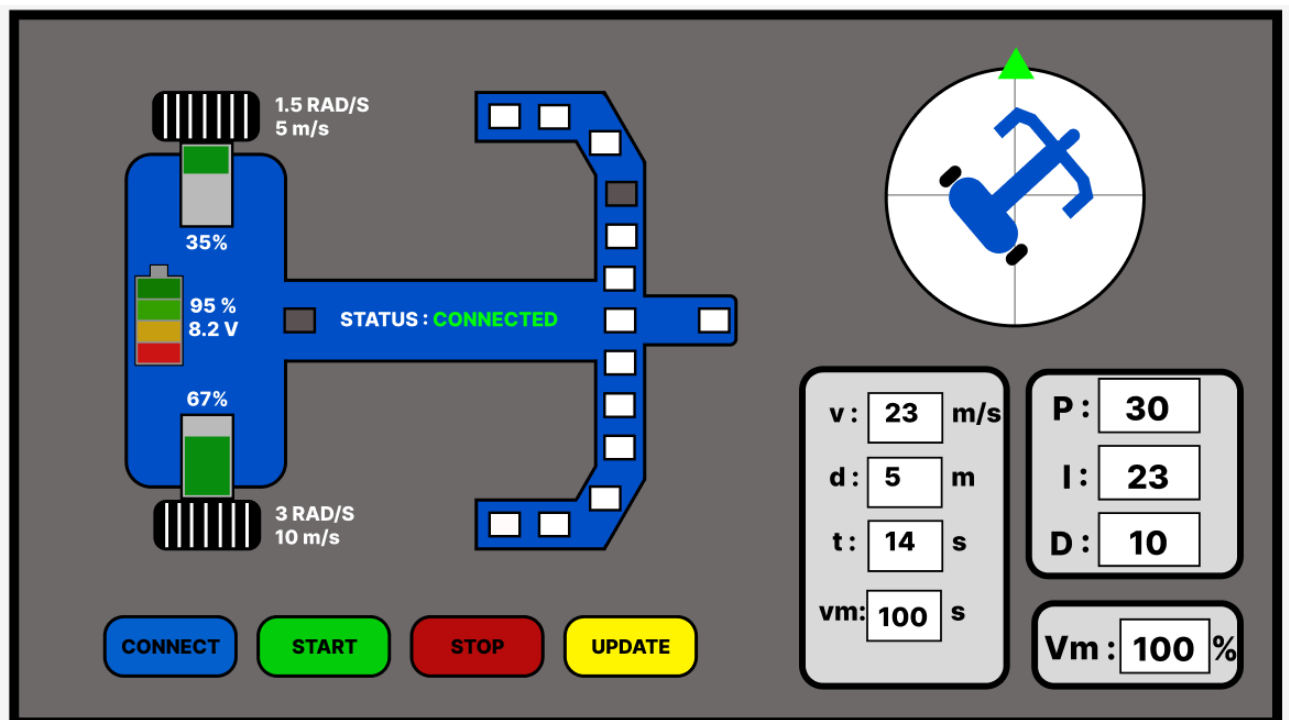
Wykres Gantta został przedstawiony na rys. 1.



Rysunek 1: Harmonogram Gantta

6 Projekt graficznego interfejsu użytkownika

Projekt interfejsu graficznego przedstawia początkowo założony sposób wizualizacji danych sensorycznych z robota. Projekt ten zobaczyć można na rys. 2.



Rysunek 2: Projekt graficzny

Na podstawie rys. 2 w aplikacji wyodrębnić można 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,

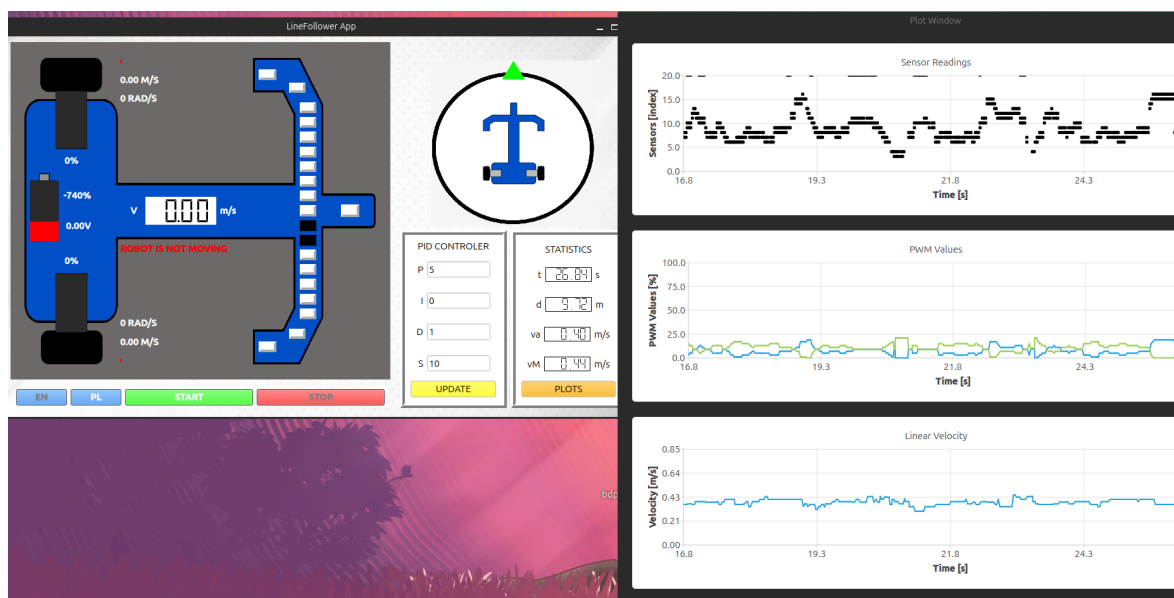
W przypadku otrzymania przez robota instrukcji startu lub stopu, pozostałe dane z ramki zostaną pominięte.

8 Rezultaty Końcowe – Opis Działania Aplikacji

Aplikacja która została utworzona spełnia wszystkie założone wymagania o których mowa w sekcji 3. Wprowadzone jednak zostało kilka zmian które spowodowały ulepszenie korzystania z aplikacji. Dodano tłumaczenia na język Polski i Angielski a także prezentacje wykresów czasowych. Ze względu na procesor użyty w urządzeniu zmieniono protokół komunikacyjny z Bluetooth na WiFi. Znacząco zmieniony został również interfejs graficzny co zaskutkowało wygodniejszym i przyjemniejszym korzystaniem z aplikacji. Jednak sama idea i schemat pozostał taki sam.

Utworzona aplikacja została podzielona na dwa osobne okna:

- Główne okno aplikacji, zawierające wizualizację danych ze wszystkich czujników oraz elementy służące do sterowania robotem. Okno to jest przedstawione na rysunkach: 3 oraz 4
- Okno z wykresami prezentujące najważniejsze odczyty dotyczące przejazdu robota w czasie. Okno to jest przedstawione na rysunkach: 3 oraz 5

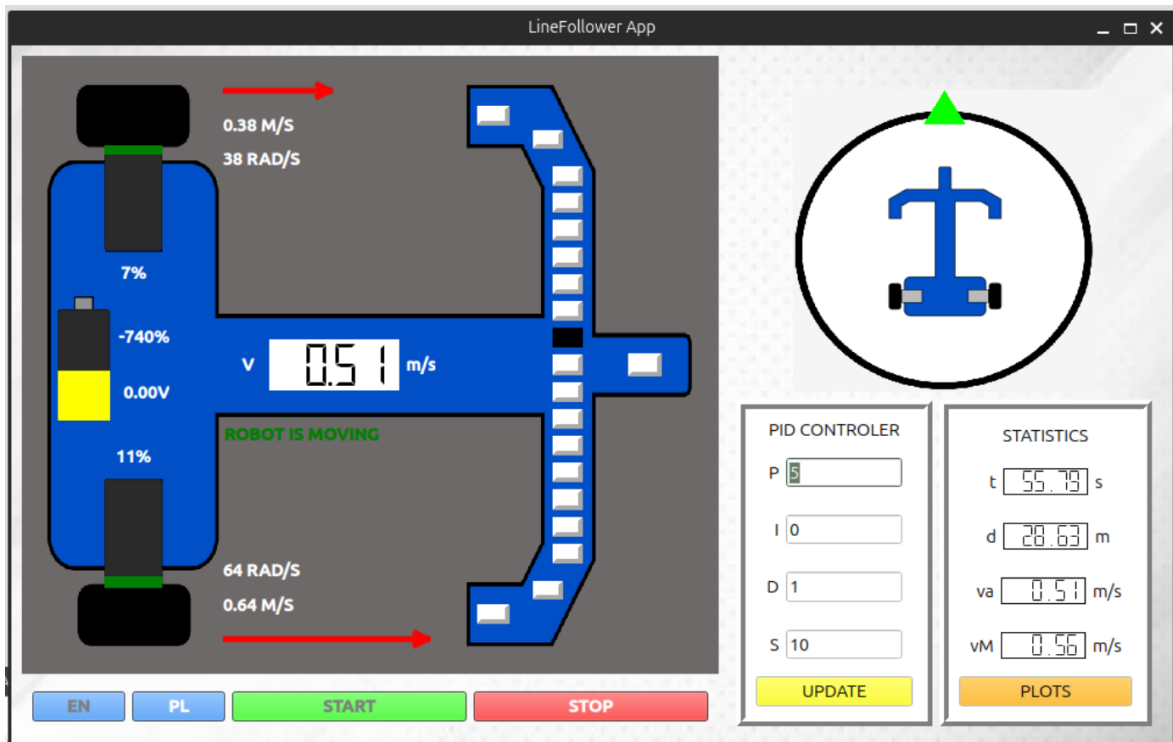


Rysunek 3: Interfejs graficzny utworzonej aplikacji

W dalszej części tej sekcji zostały opisane poszczególne funkcjonalności aplikacji z podziałem na osobne okna.

8.1 Główne Okno Aplikacji

Główne okno aplikacji zostało przedstawione na rys. 4. Zawiera ono wszystkie najważniejsze funkcjonalności aplikacji.

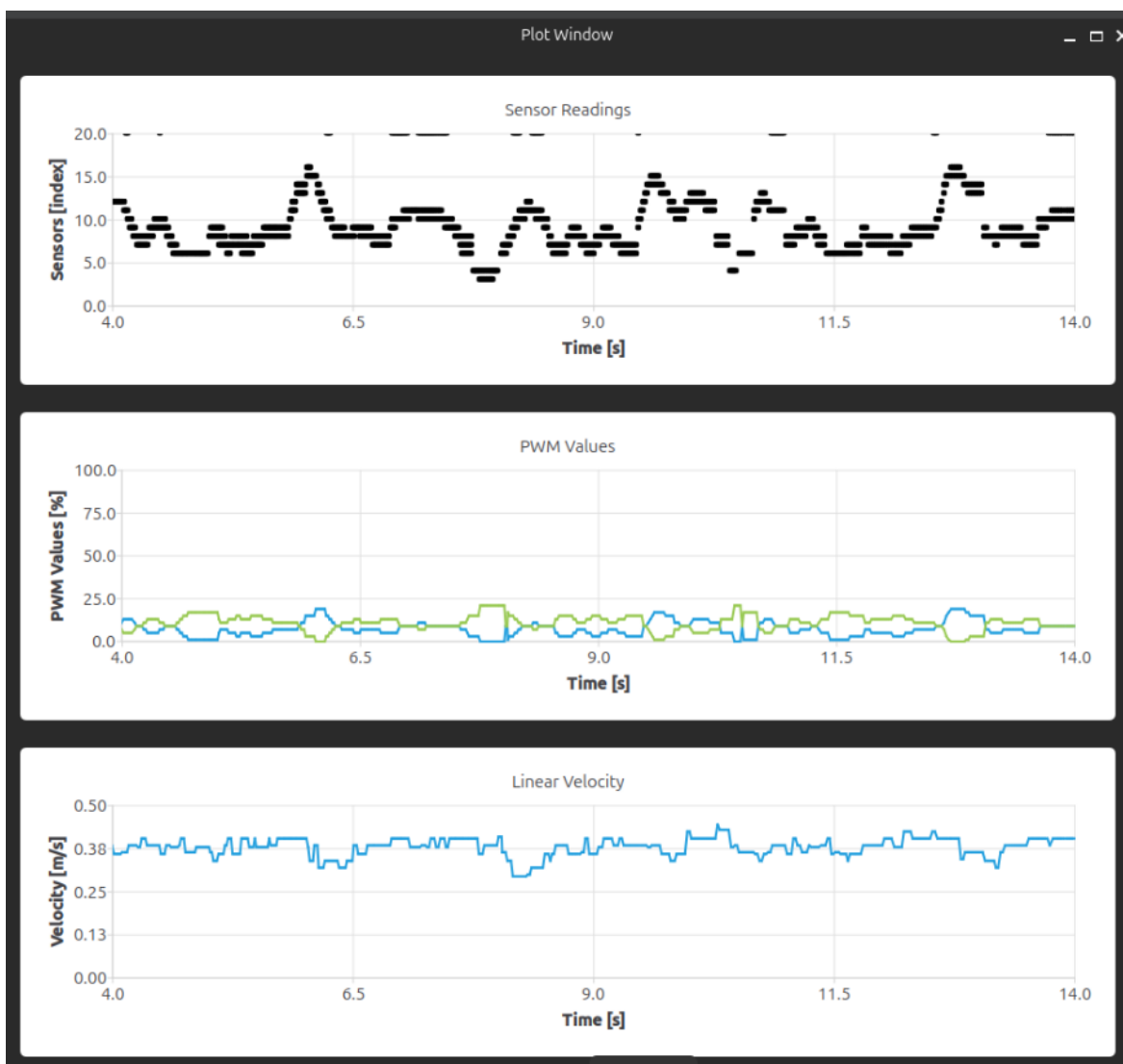


Rysunek 4: Główne okno aplikacji

- wizualizacja robota stanowi główny element okna, zawarte na nim zostały:
 - wizualizacja poziomu naładowania baterii, graficznie oraz tekstowo,
 - procentowe wypełnienie PWM na silnikach,
 - parametry odczytane z enkoderów czyli prędkość kątową kół oraz prędkość liniową kół prezentowaną również graficznie za pomocą strzałek,
 - odczyty z czujników podłoża, gdzie czarny prostokąt oznacza wykrycie linii na danych czujnikach,
 - aktualną prędkość liniową robota,
 - stan w jakim aktualnie znajduje się robot,
- kompas prezentuje aktualny obrót robota w osi "yaw" względem startu,
- przyciski zmiany języka służące do tłumaczenia całej aplikacji na wybrany język,
- przyciski "START" oraz "STOP" służące do sterowania robotem,
- sekcja regulatora PID służąca do wpisania parametrów regulatora wraz z przyciskiem do ich aktualizacji w robocie,
- statystyki danego przejazdu które prezentują takie parametry jak: czas przejazdu, przejechany dystans, średnią prędkość liniową oraz prędkość maksymalną, w tym miejscu umieszczony jest również przycisk do otworzenia okna wykresów (patrz rys. 5)

8.2 Okno z Wykresami

Przedstawione na rys. 5, okno z wykresami prezentuje odczyty z w trakcie aktualnego lub ostatniego przejazdu robota po trasie. Okno to otwierane jest po wciśnięciu przycisku "PLOTS" na głównym oknie aplikacji w sekcji "STATISTICS" (zobacz rys. 4).



Rysunek 5: Okno z wykresami

Okno to zawiera trzy wykresy prezentujące:

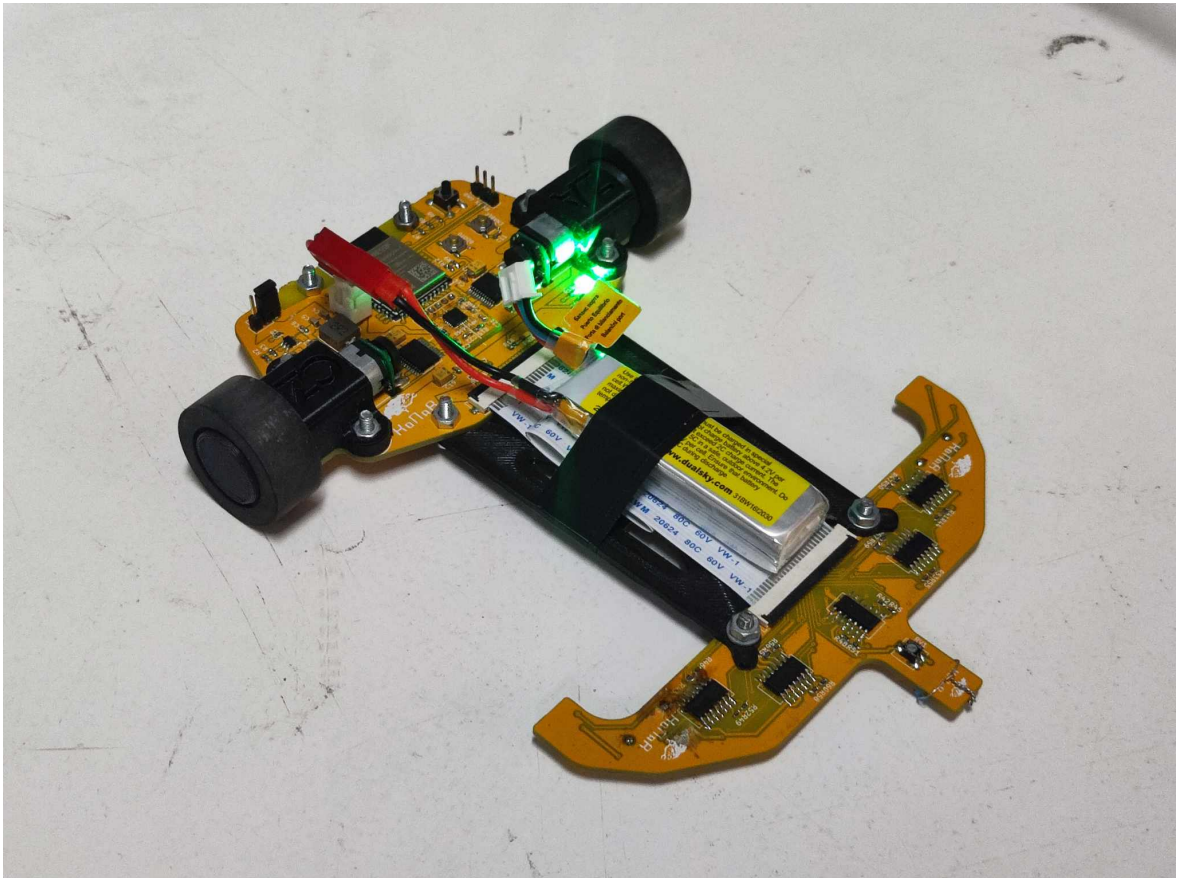
- Odczyty z czujników podłoża gdzie czarna linia oznacza na którym z czujników została znaleziona czarna linia na torze,
- Procentowe wartości wypełnienia PWM na silnikach,
- Prędkość liniową robota w metrach na sekundę,

9 Podsumowanie

Utworzona aplikacja spełnia wszystkie założone na początku funkcjonalności. Problematyczne okazało się jedynie wyświetlenie prawidłowego napięcia baterii ze względu na błąd przy projektowaniu elektroniki samego robota.

W trakcie prac nad aplikacją zmieniony został protokół komunikacyjny z Bluetooth na WiFi ze względu na użyty procesor w robocie posiadający niekompatybilny moduł, jednak nie wymusiło to znaczących zmian w sposobie komunikacji.

Ze względu na możliwość przedstawienia dużej ilości danych sensorycznych, stanowi ona bardzo dobry interfejs do sterowania takim robotem jak LineFollower czy Mini-Sumo, gdzie bez takiej aplikacji, w celu odczytania danych wymagane jest sprzętowe ulepszenie urządzenia lub wyświetlanie tych danych w nieintuicyjny, tekstowy sposób. Robot dla którego aplikacja ta została utworzona widoczny jest na rys. 6



Rysunek 6: Robot Klasy LineFollower