

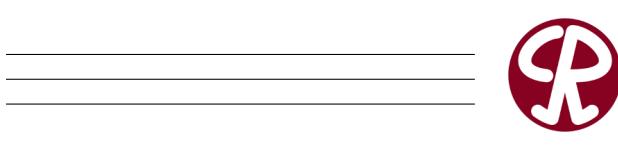
PROJEKT  
WIZUALIZACJA DANYCH SENSORYCZNYCH

---

**Wizualizacja czujników parkowania**

---

Dawid Jarząbek, 248991



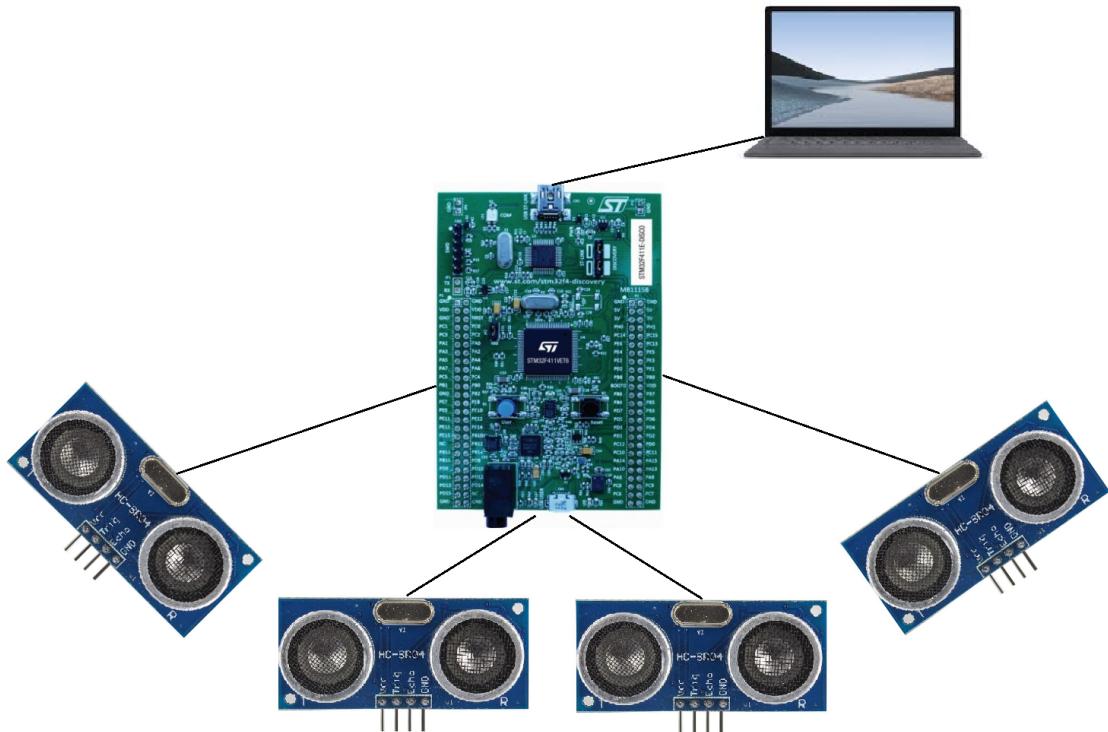
*Prowadzący:*  
dr inż. Bogdan Kreczmer  
  
Katedra Cybernetyki i Robotyki  
Wydziału Elektroniki  
Politechniki Wrocławskiej

# **Spis treści**

<b>1</b>	<b>Charakterystyka tematu projektu</b>	<b>1</b>
<b>2</b>	<b>Etapy realizacji projektu</b>	<b>1</b>
<b>3</b>	<b>Specyfikacja finalnego produktu</b>	<b>2</b>
<b>4</b>	<b>Harmonogram</b>	<b>2</b>
<b>5</b>	<b>Projekt graficznego interfejsu użytkownika</b>	<b>3</b>
5.1	Podstawowe funkcjonalności . . . . .	3
5.2	Wygląd interfejsu . . . . .	3
5.3	Mögliwe scenariusze użycia . . . . .	4
<b>6</b>	<b>Obsługa czujników</b>	<b>4</b>
<b>7</b>	<b>Komunikacja</b>	<b>7</b>
<b>8</b>	<b>Rezultaty wstępne</b>	<b>7</b>
<b>9</b>	<b>Rezultaty zaawansowane</b>	<b>8</b>
<b>10</b>	<b>Rezultaty prawie końcowe</b>	<b>10</b>
<b>11</b>	<b>Opis działania aplikacji</b>	<b>11</b>
<b>12</b>	<b>Podsumowanie</b>	<b>12</b>
	<b>Bibliografia</b>	<b>14</b>

# 1 Charakterystyka tematu projektu

Celem projektu jest stworzenie aplikacji desktopowej, która będzie wizualizowała dane sensoryczne odczytane z czujników ultradźwiękowych, mającej symulować działanie czujników parkowania w samochodzie. Ideowa struktura systemu została pokazana na Rysunku 1.



Rysunek 1: Ideowy schemat połączeń.

# 2 Etapy realizacji projektu

Kamienie milowe:

## I. Projekt interfejsu graficznego

W pierwszym etapie skupię się analizie potrzebnych funkcjonalności i zaprojektowaniu szczegółów aplikacji.

## II. Komunikacja płytki z aplikacją

Istotą tego etapu będzie właściwa komunikacja na linii czujniki–płytki–aplikacja.

## III. Wyświetlanie danych z czujników

Prawidłowe wyświetlanie odległości każdego z czujników od przeszkody oraz przebieg pomiarów w funkcji czasu naniesiony na wykresy.

## IV. Wizualizacja samochodu z systemem parkowania

Wizualizacja 2D samochodu z czujnikami parkowania oraz rozchodzące się fale ultradźwiękowe, których długość zależy od napotkanych przeszkód.

## V. Integracja wszystkich elementów – gotowy projekt

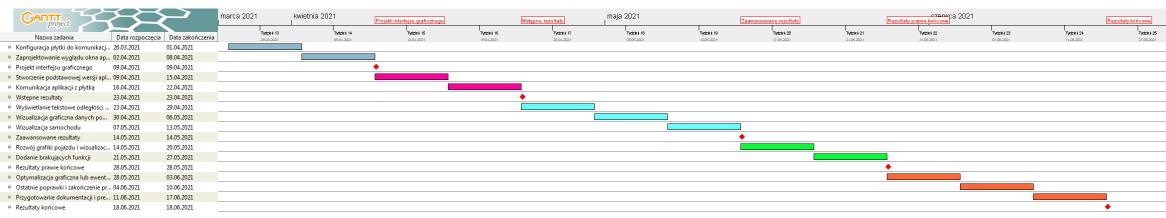
Zadbanie o przejrzystości aplikacji i poprawki graficzne.

## 3 Specyfikacja finalnego produktu

Planuje wykorzystać cztery czujniki odległości HC-SR04 [2], które będą podłączone do płytki STM32F411E-DISCOVERY[1], płytka natomiast wstępnie komunikować się będzie przewodowo poprzez interfejs UART z aplikacją (możliwe przejście na komunikację Bluetooth). Odległość będzie mierzona w zakresie 2 cm – 2 m z dokładnością do centymetra. Aplikacja powstanie przy użyciu biblioteki Qt. Projekt będzie zorientowany na wizualizację 2D, jednak z możliwością przejścia na 3D. W aplikacji będzie wyświetlana tekstowo odległość każdego z czujników od przeszkode, a także wykres historii pomiarów wybranego z czujników. Centralnym punktem aplikacji będzie jednak wizualizacja samochodu i czujników parkowania.

## 4 Harmonogram

- 26 marca 2021 – Zakończenie przeglądu materiałów i rozpoczęcie pracy
- 2 kwietnia 2021 – Konfiguracja płytki do komunikacji z czujnikami
- 9 kwietnia 2021 – Zaprojektowanie wyglądu okna aplikacji
- **9 kwietnia 2021 – Projekt interfejsu graficznego**
- 16 kwietnia 2021 – Stworzenie podstawowej wersji aplikacji okienkowej
- 23 kwietnia 2021 – Komunikacja aplikacji z płytą
- **23 kwietnia 2021 – Wstępne rezultaty**
- 30 kwietnia 2021 – Wyświetlanie tekstowe odległości z czujników w aplikacji
- 7 maja 2021 – Wizualizacja graficzna danych pomiarowych
- 14 maja 2021 – Wizualizacja samochodu
- **14 maja 2021 – Zaawansowane rezultaty**
- 21 maja 2021 – Rozwój grafiki pojazdu i wizualizacja fal ultradźwiękowych
- 28 maja 2021 – Dodanie brakujących funkcji
- **28 maja 2021 – Rezultaty prawie końcowe**
- 4 czerwca 2021 – Optymalizacja graficzna lub ewentualne rozszerzenia (np. komunikacja Bluetooth albo obracanie samochodem )
- 11 czerwca 2021 – Ostatnie poprawki i zakończenie pracy nad projektem
- 18 czerwca 2021 – Przygotowanie dokumentacji i prezentacji projektu
- **18 czerwca 2021 – Rezultaty końcowe**



Rysunek 2: Diagram Gantta.

## 5 Projekt graficznego interfejsu użytkownika

### 5.1 Podstawowe funkcjonalności

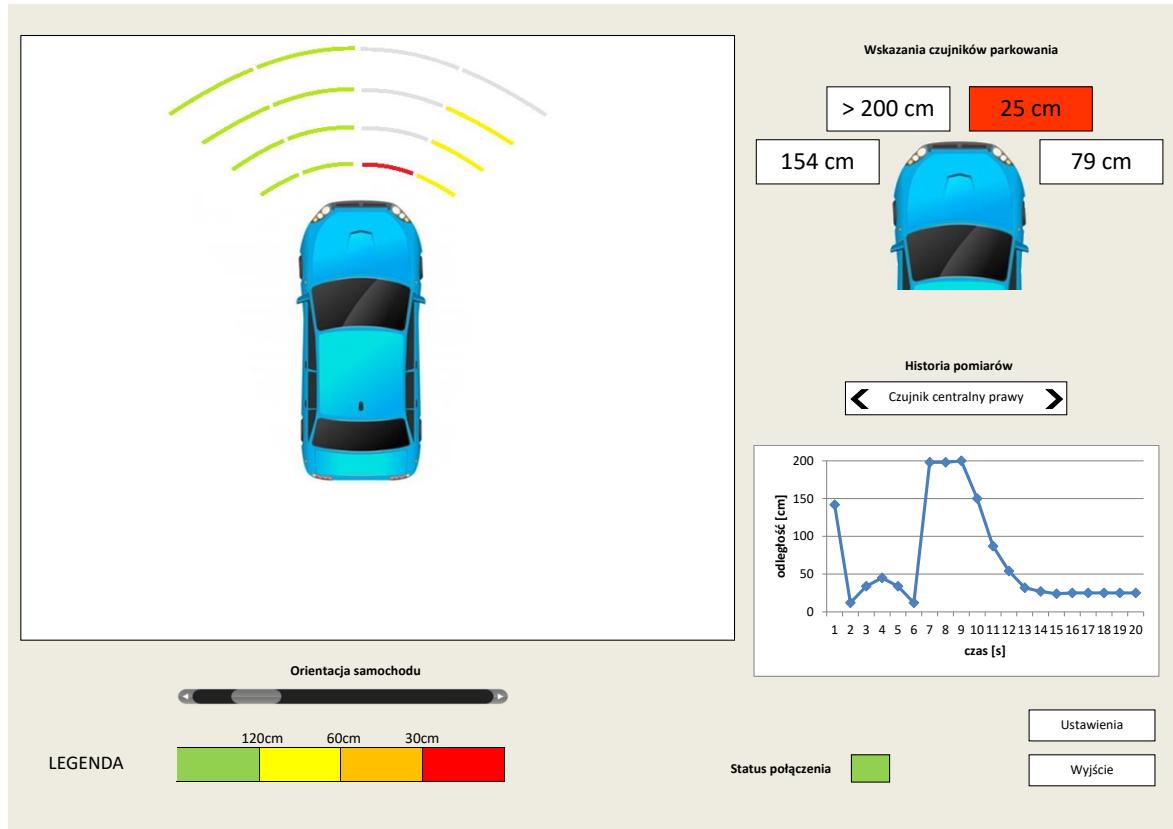
- **Wizualizacja danych z czujników odległości**

Aplikacja będzie wyświetlać odległość każdego z czujników od przeszkody, można też będzie śledzić historię pomiarów (wybranego czujnika) przedstawioną na wykresie. Odległość wizualizowana będzie także w postaci fal ultradźwiękowych wraz z modelem samochodu.

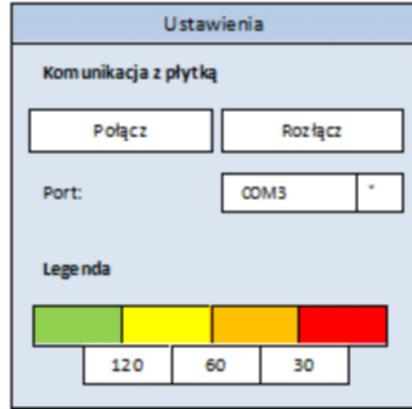
- **Rotacja samochodu**

Będziemy mogli obracać samochodem w poziomie przesuwając odpowiednio pasek obrotu, w rozszerzeniu możliwe dodanie czujników orientacji platformy

### 5.2 Wygląd interfejsu



Rysunek 3: Projekt grafiki aplikacji



Rysunek 4: Okno ustawień

### 5.3 Możliwe scenariusze użycia

- **Historia pomiarów**

Domyślnie wyświetlany będzie wykres z czujnika skrajnie lewego, natomiast klikając przycisk można będzie zmieniać czujnik, którego historia pomiarów jest wyświetlana.

- **Rotacja samochodu**

Samochód domyślnie skierowany będzie ku górze ekranu, przesuwając pasek obrotu będziemy mogli go obrócić w zakresie od 0 do 360 stopni. W ramach rozszerzenia można dodać do platformy czujniki orientacji, które automatycznie pozycjonowałyby samochód.

- **Wizualizacja fal ultradźwiękowych**

Długość fali oraz kolory zależeć będą od odległości przeszkody od czujnika. Zakresy poszczególnych odległości widoczne na legendzie będzie można zmienić w ustawieniach – w nowym oknie, które pojawi się po wcisnięciu przycisku „Ustawienia”. Fale będą się obracały wraz z samochodem.

- **Komunikacja z płytką**

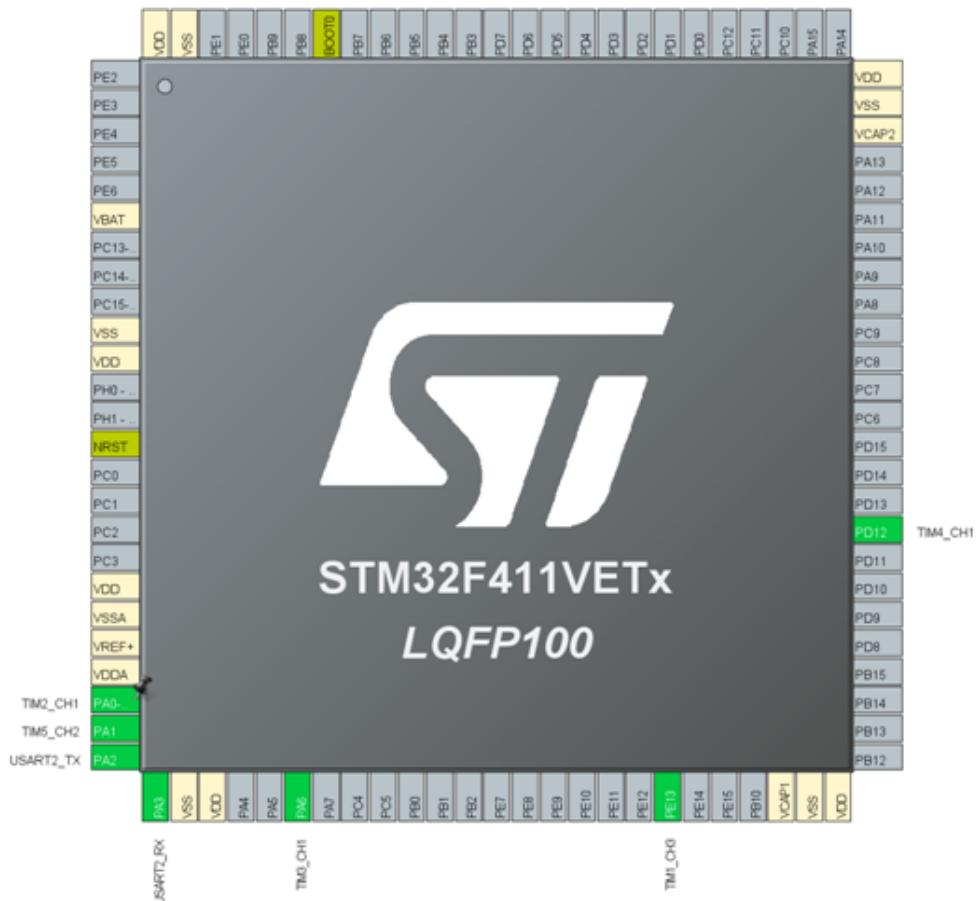
Żeby rozpoczęć odczyt danych z czujników należy najpierw ustawić port, do którego podłączona jest płytka. Można to zrobić po wcisnięciu przycisku „Ustawienia”. Po wybraniu portu przyciskiem „Połącz” łączymy się z płytą i do aplikacji zaczną spływać dane. W głównym oknie aplikacji widnieje dioda, która będzie informować o statusie połączenia – zielony kolor oznacza poprawne połączenie, czerwony przeciwnie. Do zakończenia przesyłu informacji służy przycisk „Rozłącz”, natomiast zamknięcie aplikacji umożliwia przycisk „Wyjście”.

## 6 Obsługa czujników

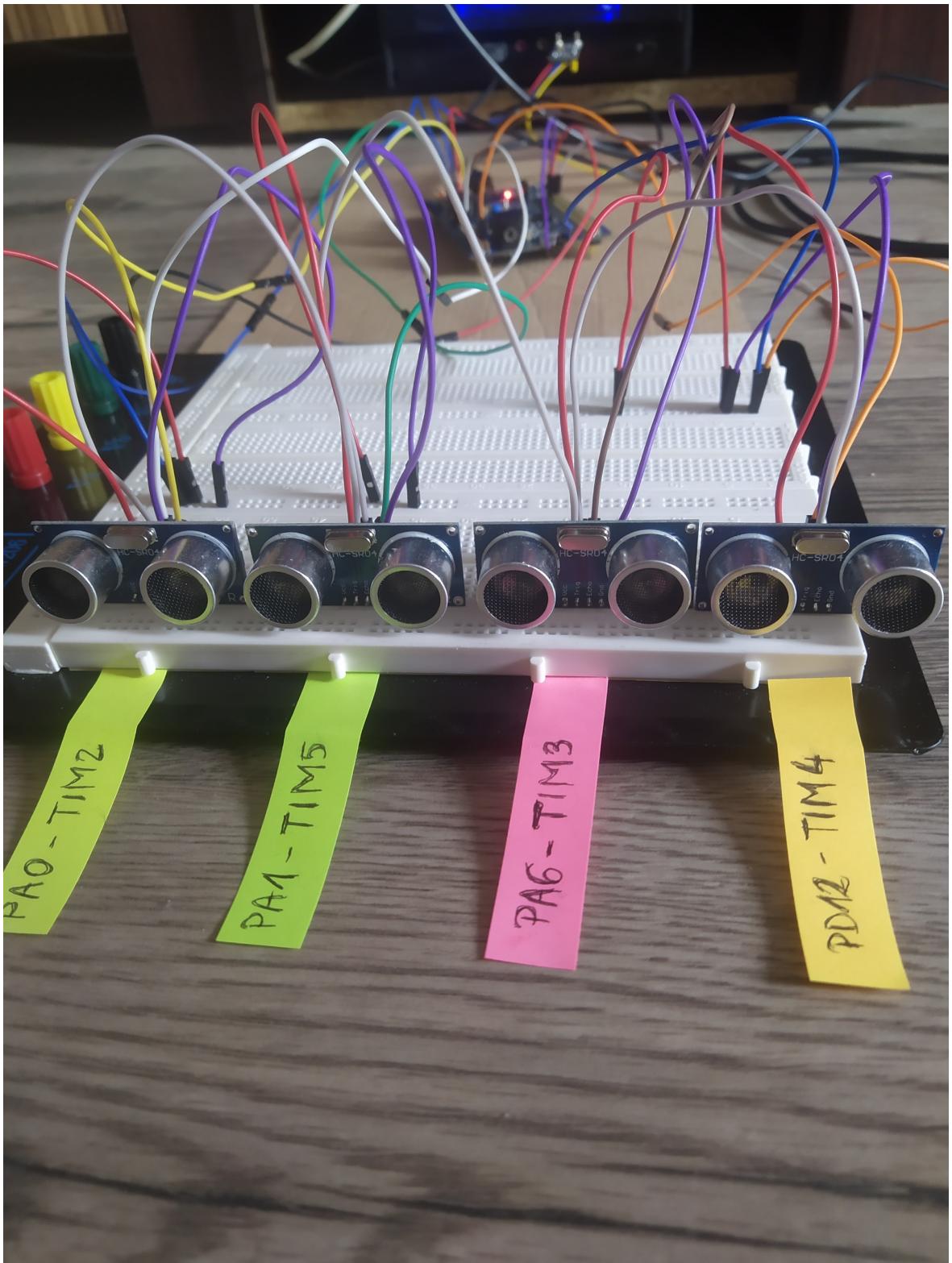
Do pomiarów odległości wykorzystuję 4 czujniki ultradźwiękowe HC-SR04 [2]. Wstępnie zostały one zamontowane na płytce stykowej (rys. 6), możliwe jest przeniesienie ich na osobne stojaki, dzięki czemu zwiększy się rozstaw, czujniki będą na siebie oddziaływać w mniejszym stopniu, obszar śledzenia będzie bardziej zróżnicowany. Po

ewentualnym zaimplementowaniu komunikacji bezprzewodowej czujniki będzie można przenieść także na rzeczywistego robota.

Każdy z czujników posiada cztery wyprowadzenia: dwa do zasilania (VCC i GND) i dwa do wykonywania pomiarów (TRIGGER oraz ECHO). HC-SR04 powinien być zasilany napięciem 5 V. Podanie stanu wysokiego przez co najmniej  $10 \mu\text{s}$  na wejście TRIGGER powoduje wyzwolenie pomiaru odległości. Czujnik generuje wtedy 8 impulsów o częstotliwości 40 kHz, a następnie oczekuje na powrót odbitej fali ultradźwiękowej. Wynik tego pomiaru jest podawany na wyjście ECHO, które generuje impuls o czasie trwania proporcjonalnym do zmierzonej odległości. W praktyce 1 cm to impuls o długości  $58 \mu\text{s}$ , wynika to z prędkości dźwięku ( 340 m/s ). Do wyzwalania pomiarów wykorzystałem TIMER1 skonfigurowany w trybie PWM, który co pół sekundy przez 20 mikrosekund ustawia stan wysoki na pinie PE13, który jest podłączony do wejść TRIGGER czujników. Do odczytu pomiarów wykorzystuje timery: TIM2, TIM5, TIM3 i TIM4, skonfigurowane w trybie Input PWM (osobny dla każdego czujnika). Mierzą one czas trwania impulsów pochodzących od czujników. Następnie dane przeliczane są na odległość i wysyłane za pomocą interfejsu UART do komputera. Ponieważ zdecydowałem się mierzyć odległość w zakresie do 2 m, wszystkie większe odległości będą oznaczone jako 2 m. Natomiast zbyt bliskie odległości lub inne zakłócenia będą traktowane jako 0 cm. Konfiguracja pinów mikrokontrolera została przedstawiona na Rysunku 5.



Rysunek 5: Konfiguracja mikrokontrolera



Rysunek 6: Montaż czujników

## 7 Komunikacja

Mikrokonroler przesyła dane do komputera poprzez konwerter UART-USB w trybie ciągłym co 1s . Ramka danych składa się z 14 bajtów (tab.1). Nagłówkiem, który informuje o początku ramki jest znak '#'. W dalszej kolejności wysyłane są odległości (uint8\_t) poszczególnych czujników od przeszkoły podane w centymetrach, a na koniec zapisywana jest suma kontrolna w postaci heksadecymalnej z pierwszych dziewięciu bajtów i znaki CR LF. Jako separator wykorzystano znak spacji. Po odebraniu ramki, jeśli została ona poprawnie przesłana ( nagłówkiem jest znak '#' i suma kontrolna się zgadza ), odległości z czujników wyświetlane są w aplikacji.

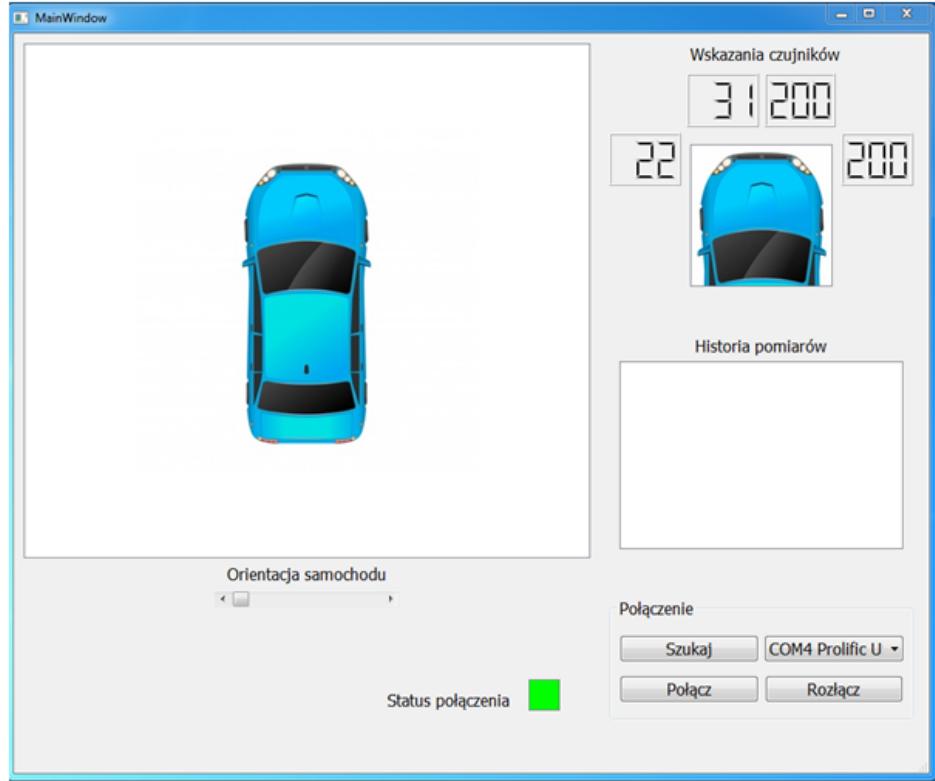
1	2	3	4	5	6	7	8	9	10	11	12	13	14
'#'	,	cz1	,	cz2	,	cz3	,	cz4	,	CRC	,	CR	LF

Tabela 1: Ramka danych

O statusie połączenia informuje „dioda”, które świeci się na zielono, gdy połączymy się z portem szeregowym, natomiast na czerwono, gdy połączenie nie jest nawiązane.

## 8 Rezultaty wstępne

Wygląd aplikacji na tym etapie jest widoczny na Rysunku 7.



Rysunek 7: Widok aplikacji na etapie „Rezultaty wstępne”

## 9 Rezultaty zaawansowane

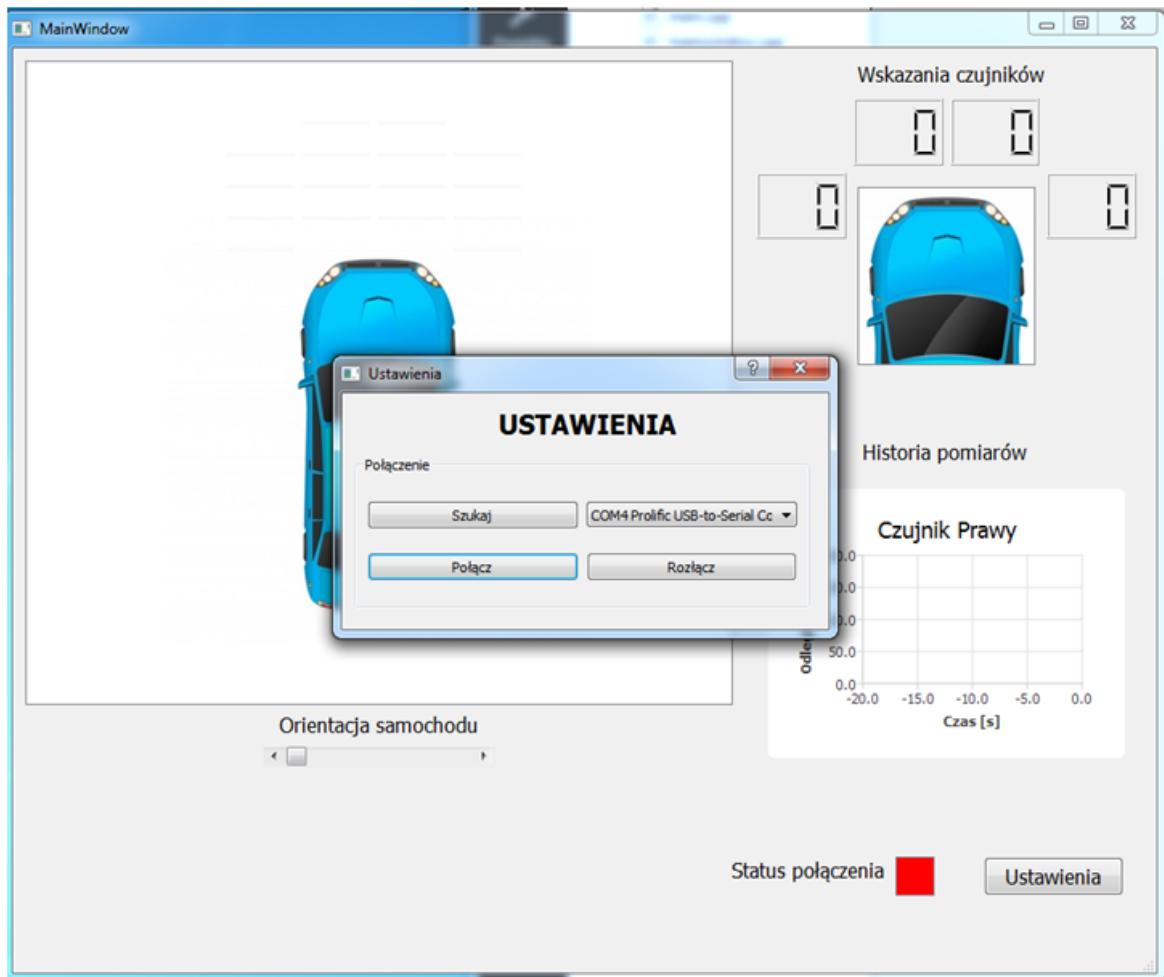
Do aplikacji dodano wizualizację odległości w formie emitowanych fal ultradźwiękowych (w postaci linii prostych) o zmiennym kolorze:

- zielony – odległości większe niż 120 cm
- żółty – odległości pomiędzy 60 cm a 120 cm
- pomarańczowy – odległości pomiędzy 30 cm a 60 cm
- czerwony – odległości do 30 cm.

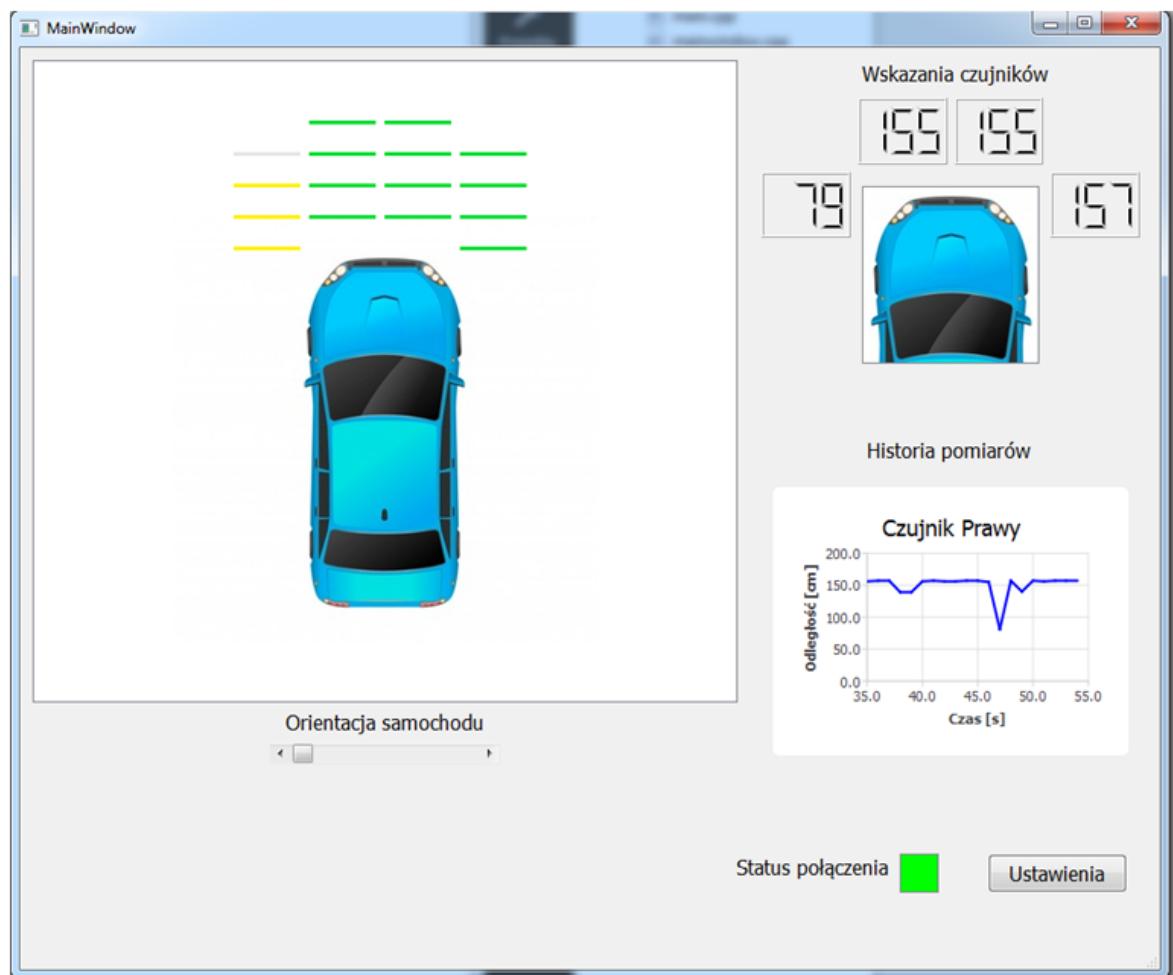
Do aplikacji dodano też wykresy pokazujące historię pomiarów prawego czujnika. W późniejszym etapie dodane zostaną, także wykresy dla pozostałych czujników.

Dodano także przycisk „Ustawienia”, który otwiera okno ustawień połączenia z portem szeregowym.

Wygląd aplikacji po uruchomieniu został przedstawiony na Rysunku 8. Natomiast Rysunek 9 przedstawia aplikację już po połączeniu się z mikrokontrolerem.



Rysunek 8: Widok aplikacji na etapie „Rezultaty zaawansowane”



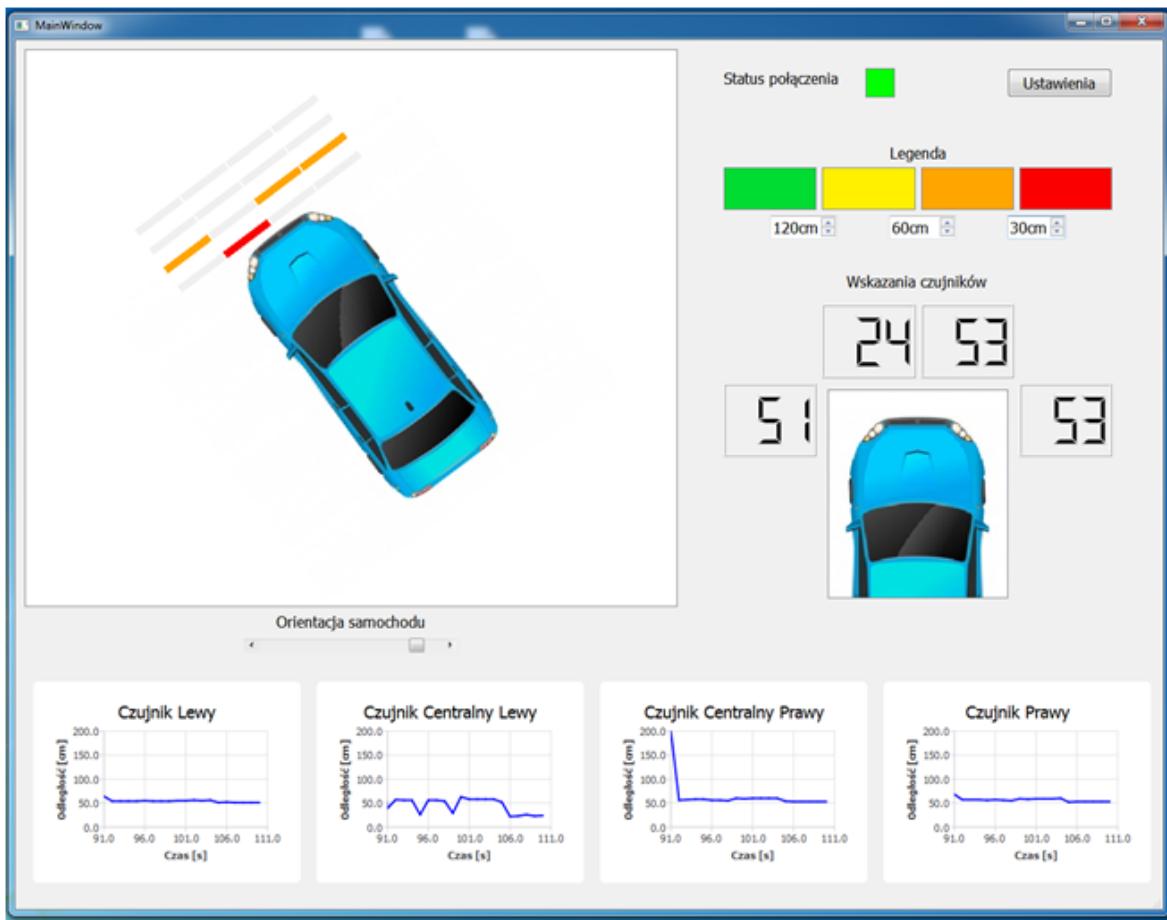
Rysunek 9: Widok aplikacji na etapie „Rezultaty zaawansowane”

## 10 Rezultaty prawie końcowe

W tym etapie dodano możliwość zmiany orientacji samochodu za pomocą suwaka (rys. 10). Obrót w jest wykonywany w zakresie od 0 do  $360^\circ$ .

Zmieniono koncepcję wizualizacji fal ultradźwiękowych. Teraz kolorowane są miejsca, w których wykryto przeszkodę – kolor zależy od odległości. Zakresy odległości, dla których przypisywany jest dany kolor, można zmieniać pod legendą umieszczoną w prawej części aplikacji. Gdy nie wykryto żadnej przeszkody fale są koloru szarego.

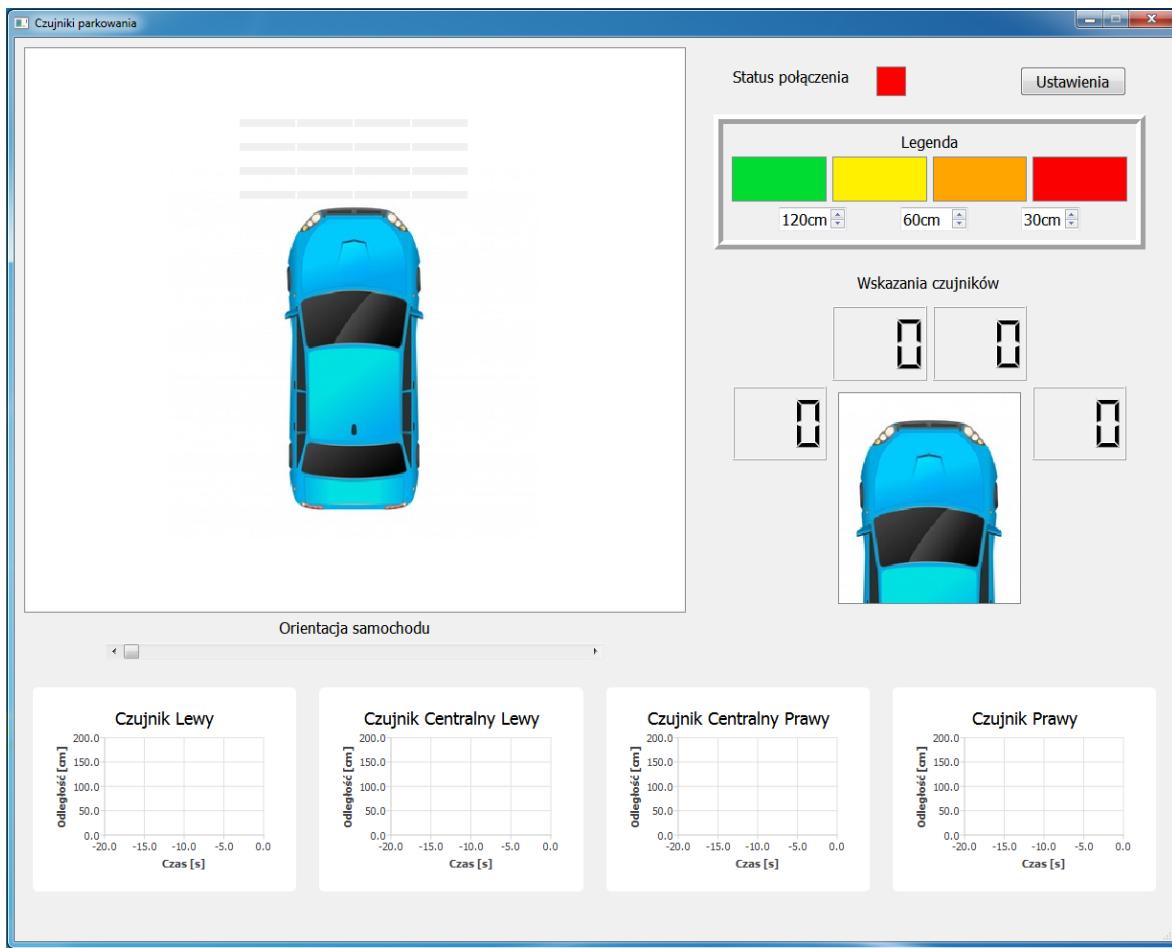
Dodano także wykresy historii pomiarów dla pozostałych czujników, przez co zmienił się trochę rozkład elementów w aplikacji.



Rysunek 10: Widok aplikacji na etapie „Rezultaty prawie końcowe”

## 11 Opis działania aplikacji

Po uruchomieniu aplikacji pojawia się okno główne aplikacji (rys.11). Wykresy historii pomiarów dla poszczególnych czujników nie zawierają linii danych, wskazania odległości od przeszkody wynoszą 0, samochód nie jest obrócony o żaden kąt, a jego orientacja jest pionowa, fale ultradźwiękowe są koloru szarego, co jest równoznaczne z brakiem wykrycia przeszkody, dioda sygnalizująca status połączenia świeci na czerwono, a zakresy dla legendy są domyślnie ustawione (120 cm, 60 cm i 30 cm). Żeby zacząć odczytywać należy kliknąć w przycisk Ustawienia i w nowym oknie wyszukać port podłączony do płytki, a następnie się z nim połączyć. Jeśli udało się poprawnie połączyć dioda informująca o statusie połączenia zmieni kolor na zielony, oraz pojawią się wskazania odległości od przeszkoda, wizualizacja umiejscowienia przeszkód oraz zaczyna się wyrysowywać wykresy historii pomiarów. Orientację samochodu można zmieniać za pomocą suwaka umieszczonego pod obrazkiem. Natomiast pod legendą można zmieniać odległości w jakich są wyświetlane przeszkody odpowiedniego koloru. Czujniki w miarę dokładnie oddają rzeczywistą odległość od przeszkód (rys. 12, rys. 13). Żeby zakończyć przesył danych w Ustawieniach należy wcisnąć przycisk Rozłącz.



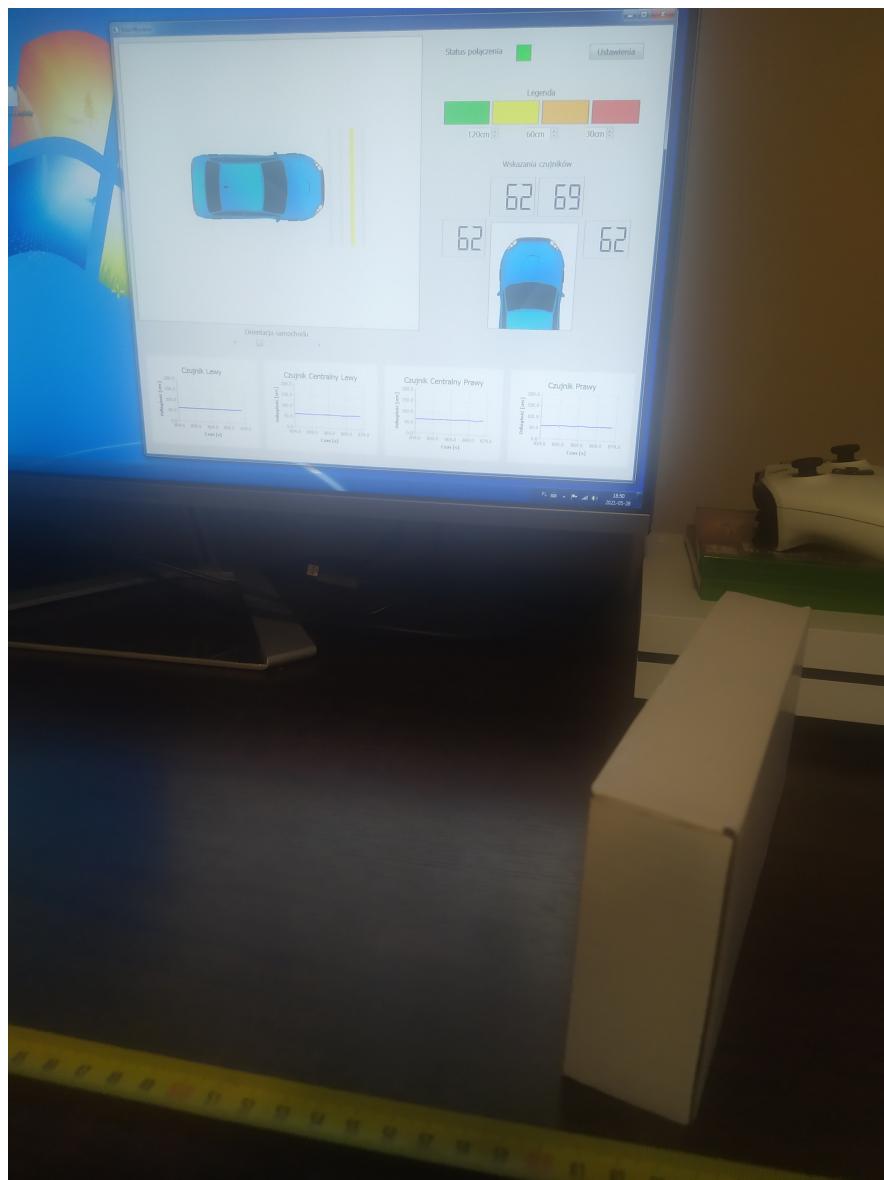
Rysunek 11: Rezultaty końcowe

Nagranie prezentujące wykonywanie pomiarów i działanie aplikacji:  
<https://www.youtube.com/watch?v=qMRQl52bFTw>

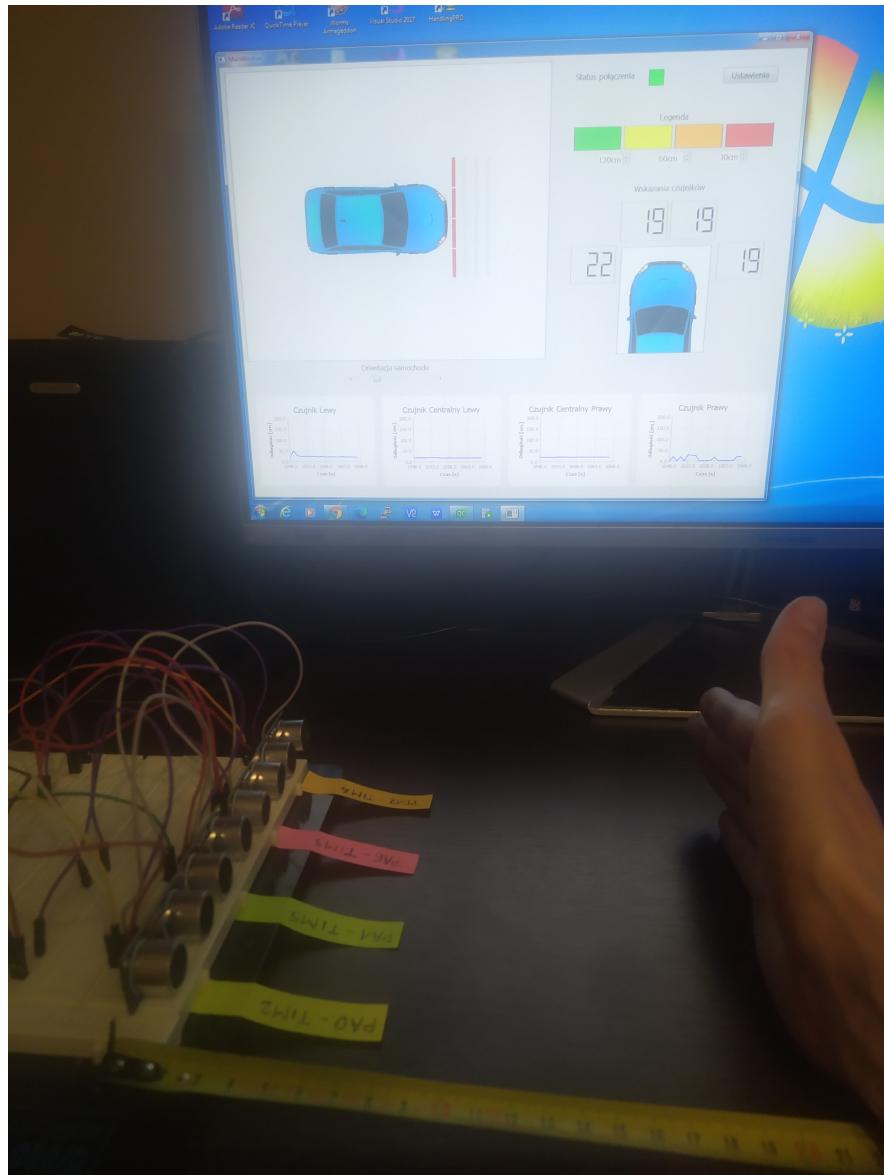
## 12 Podsumowanie

Podsumowując, w ramach projektu udało się zrealizować wszystkie wstępnie zadeklarowane cele. Zrezygnowano z ewentualnych rozszerzeń jak na przykład komunikacja przez Bluetooth czy dołączenie czujników orientacji, jednakże aplikacje można w tym kierunku rozwijać w przyszłości. Projekt pozwolił nauczyć się obsługi czujników ultradźwiękowych przez mikontrolery STM32 oraz tworzenia prostych aplikacji desktopowych i wizualizacji danych z czujników.

**Link do repozytorium:**  
<https://github.com/djairw4/WDS>



Rysunek 12: Przedmiot postawiony ponad 60cm od czujników



Rysunek 13: Ręka postawiona mniej więcej 20cm od czujników

## **Spis rysunków**

1	Ideowy schemat połączeń . . . . .	1
2	Diagram Gantta. . . . .	3
3	Projekt grafiki aplikacji . . . . .	3
4	Okno ustawień . . . . .	4
5	Konfiguracja mikrokontrolera . . . . .	5
6	Montaż czujników . . . . .	6
7	Widok aplikacji na etapie „Rezultaty wstępne” . . . . .	7
8	Widok aplikacji na etapie „Rezultaty zaawansowane” . . . . .	8
9	Widok aplikacji na etapie „Rezultaty zaawansowane” . . . . .	9
10	Widok aplikacji na etapie „Rezultaty prawie końcowe” . . . . .	10
11	Rezultaty końcowe . . . . .	11
12	Przedmiot postawiony ponad 60cm od czujników . . . . .	12
13	Ręka postawiona mniej więcej 20cm od czujników . . . . .	13

## **Spis tabel**

1	Ramka danych . . . . .	7
---	------------------------	---

## **Literatura**

- [1] Płytyka stm32f411e - discovery. [https://www.st.com/resource/en/user\\_manual/dm00148985-discovery-kit-with-stm32f411ve-mcu-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/dm00148985-discovery-kit-with-stm32f411ve-mcu-stmicroelectronics.pdf).
- [2] Ultradźwiękowy czujnik odległości hc-sr04. <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>.