



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Wiedza i doświadczenie projektowe wizytówką absolwenta kierunku automatyka i robotyka na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej

POKL.04.01.02-00-020/10

Program Operacyjny Kapitał Ludzki współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego

Gliwice, 18.02.2014r.

Międzywydziałowe Koło Naukowe High Flyers
Wydział Automatyki Elektroniki i Informatyki
Kierunek Automatyka i Robotyka

Raport z realizacji projektu:

Mobilcam 3. System sterowania.

Zespół projektowy:

Marek Nocoń(Lider projektu)	
Paweł Kędzior	
Wojciech Brychcy	
Jakub Piasecki	

Zespół konsultantów:

Maksymilian Byliński	
Tomasz Korzyniec	
Mateusz Drost	

Opiekun projektu:

mgr inż. Grzegorz Szafrąński	
------------------------------	--

1. Opis projektu.

1.1. Cel projektu:

Wykonanie jednostki GCS pozwalającej na sterowanie platformą mobilną w trybie manualnym oraz umożliwiającej zbieranie danych i wyświetlenie obrazu w trybie autonomicznym.

Poruszanie się platformy mobilnej w trybie autonomicznym na bazie odczytów z radaru oraz kamery.

1.2. Założenia projektu:

- Rozwinięcie transportu wizji oraz fonii na jednostkę GCS za pomocą WiFi.
- Sterowanie platformą mobilną z jednostki GCS.
- Wykonanie algorytmów sztucznej inteligencji

1.3. Oczekiwane wyniki: Uzyskanie sztucznej inteligencji pozwalającej na omijanie przeszkód, dokładne wyznaczanie mapy terenu, poruszanie się z punktu do punktu oraz rozpoznawanie potencjalnych obiektów.

1.4. Ocena ryzyka projektu: Poziom trudności wykonania projektu jest wysoki. Nie istnieją problemy związane z dostępnością podzespołów do realizacji projektu. Koszty projektu są jasno określone. Harmonogram został dobrany w sposób umożliwiający stopniowy rozwój projektu oraz został wyszczególniony okres, który jest przeznaczony na nadrobienie ewentualnych zaległości w projekcie. Ponadto wszyscy członkowie zespołu wyrazili chęć podjęcia się wykonania projektu. Mając na uwadze interdyscyplinarny kierunek kształcenia studentów na Wydziale Automatyki, Elektroniki i Informatyki, ich zainteresowania tematyką autonomicznych pojazdów mobilnych istnieje duże prawdopodobieństwo pozytywnego ukończenia projektu.

2. Podział projektu na zadania.

1. Transport danych:
 - 1.1. Określenie wymagań sprzętowych.
 - 1.2. Wybór protokołu komunikacji.
 - 1.3. Wykonanie algorytmu transportu danych oraz błędów.
 - 1.4. Transport obrazu oraz dźwięku drogą radiową na jednostkę GCS.
2. Sterowanie platformą mobilną z jednostki GCS.
 - 2.1. Określenie środowiska programistycznego.
 - 2.2. Projekt panelu sterowania.
 - 2.3. Testowanie sterowania.
3. Sztuczna inteligencja:
 - 3.1. Określenie wymagań sprzętowych.
 - 3.2. Wykonanie algorytmu omijania przeszkód na bazie kamery oraz radaru.
 - 3.3. Tworzenie mapy obszaru.
4. Wykonanie raportu.

3. Harmonogram prac projektowych.

Osadzając wyszczególnione zadania na osi czasu, sporządzono harmonogram realizacji projektu.

Miesiąc		Listopad				Grudzień				Styczeń					Luty			
Projekt		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Transport danych	1	█	█	█	█													
	2					█	█	█										
	3									█	█	█	█	█				
	4														█	█	█	█
Sterowanie platform	1	█	█															
	2			█	█	█	█	█	█	█	█	█						
	3												█	█	█	█	█	
Sztuczna inteligencja	1	█	█	█														
	2				█	█	█	█	█	█	█							
	3												█	█	█	█	█	
Raport																		

4. Kamienie milowe.

- Transport danych z platformy mobilnej do jednostki GCS.
- Sterowanie platformą mobilną z poziomu GCS.
- Omijanie przeszkód oraz poruszanie się z punktu A do punktu B.

5. Określenie zasad odpowiedzialności członków zespołu.

- Marek Nocoń – Odpowiedzialny za projekt oraz algorytmy.
- Paweł Kędzior – Odpowiedzialny za transport danych drogą radiową.
- Wojciech Brychcy – Odpowiedzialny za wykonanie GCS'a.
- Jakub Piasecki – Odpowiedzialny za wykonanie algorytmów.
- Maksymilian Byliński – konsultant do spraw związanych z transportem danych.
- Tomasz Korzyniec – konsultant do spraw związanych z algorytmami.
- Mateusz Drost – konsultant do spraw związanych z GCS'em.

6. Wykonanie

1. Transport danych:

1.1. Określenie wymagań sprzętowych.

W celu wykonania komunikacji między jednostką nadrzędną rozumianą tutaj jako GCS, a jednostką podrzędną czyli pojazdem mobilnym z szerokim pasmem przepustowym oraz o dużym zasięgu komunikacji, należało dobrać odpowiedni sposób komunikacji. Z pośród dostępnych metod komunikacji tj. XBEE, Bluetooth, RFM itd. Żaden nie spełniał założeń projektowych tj:

- Małe ograniczenie pasma przepustowego,
- możliwość osiągnięcia komunikacji na dużych odległościach,
- wysoki poziom szyfrowania danych między urządzeniami komunikującymi się,
- możliwość łączenia większej liczby urządzeń.

Jedynym rozwiązaniem była komunikacja w oparciu o WLAN – bezprzewodowe sieci komputerowe. W celu wykonania takiej komunikacji konieczne było zastosowanie urządzenia posiadającego dużą moc obliczeniową, która umożliwiła by pracę urządzenia jako Acces Point – czyli punkt dostępu do sieci WLAN, oraz przetwarzanie obrazu włącznie z jego przesyłaniem. Poniżej zamieszczono podstawowe założenia jakie powinno spełnić urządzenie, aby można było w łatwy i szybki sposób wykonać z niego urządzenie dostępne i przesyłające dane między pojazdem a jednostką GCS:

- Duża moc obliczeniowa,
- możliwość pracy pod systemem operacyjnym Linux,
- wejście USB pod które można podpiąć nadajnik/odbiorcę sieci bezprzewodowej oraz kamerę,
- możliwością instalacji OpenCV oraz gstreamera potrzebnego do obróbki oraz transportu obrazu.

Urządzenie spełniające powyższe wymagania zostało zamówione w sklepie Kamami. Charakteryzuje się ono 4 rdzeniami o częstotliwości taktowania 1GHz, oraz przygotowanym systemem Linux pod ten procesor co w dużym stopniu ułatwia późniejszą pracę.

1.2. Wybór protokołu komunikacji.

Ze względu na założenia sprzętowe, w których do komunikacji wybrano WLAN, wybór protokołu komunikacyjnego ograniczył się do UDP oraz TCP/IP. Pierwszy protokół charakteryzuje się szybszym działaniem, gdyż nie wymaga potwierdzania ramki odebranej. W drugim przypadku komunikacja odbywa się poprzez negocjację połączenia. Po krótko maszyna, która rozpoczyna komunikację wysyła pakiet TCP z znacznikiem SYN do konkretnej maszyny na konkretny port. Następnie maszyna odbierająca pakiet odpowiada znacznikiem SYN ACK jeśli zgadza się na połączenie lub RST w sytuacji kiedy nie zgadza się na połączenie. W sytuacji nawiązania połączenia każdy poprawnie odebrany pakiet musi zostać potwierdzony. Wprowadza to pewne ograniczenia w prędkości transmisji. Dodatkowo wymusza nawiązywanie połączenia z jednostką podrzędną w celu odebrania danych. Lepszym rozwiązaniem jest Zastosowanie protokołu UDP i przesyłaniem pakietów do wszystkich połączonych maszyn w sieci – czyli pracującego jako multicast. Pozwala to zaoszczędzić moc obliczeniową procesora na liczbie nawiązanych połączeń. Minusem tego trybu pracy jest brak informacji o komunikacji z jednostką nadrzędną w przypadku przesyłania rozkazów lub błędów. W tym przypadku zastosowano się o komunikację na poziomie warstwy aplikacji, gdzie powstał osobny protokół komunikacji wykorzystywany w komunikacji między urządzeniami zamontowanymi na platformie. Plusami zastosowania komunikacji UDP między GCS a platformą jest możliwość przesyłania obrazu do większej liczby urządzeń dzięki zastosowaniu multicast.

1.3. Wykonanie algorytmu transportu danych oraz błędów.

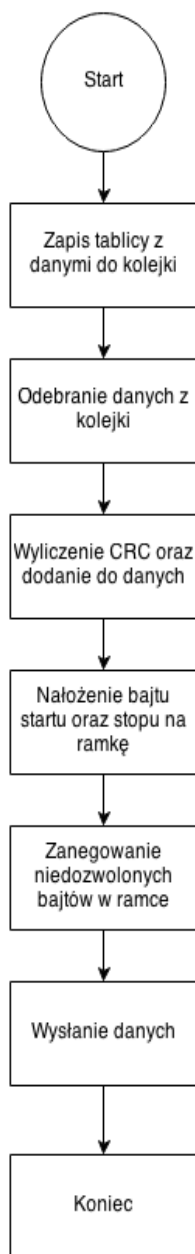
Transport danych i błędów na jednostkę GCS może odbywać się poprzez WLAN, oraz Bluetooth, XBee, RFM itd. Dzięki zastosowaniu uniwersalnego protokołu komunikacyjnego przesyłanego za pomocą dowolnego medium jest możliwa szybka zmiana urządzenia komunikacyjnego w sytuacji straty połączenia za pomocą aktualnie wykorzystywanego medium.

Protokół ten opiera się o kapsułkowanie danych tzn. dane przygotowane do wysłania są dodatkowo zamykane między:

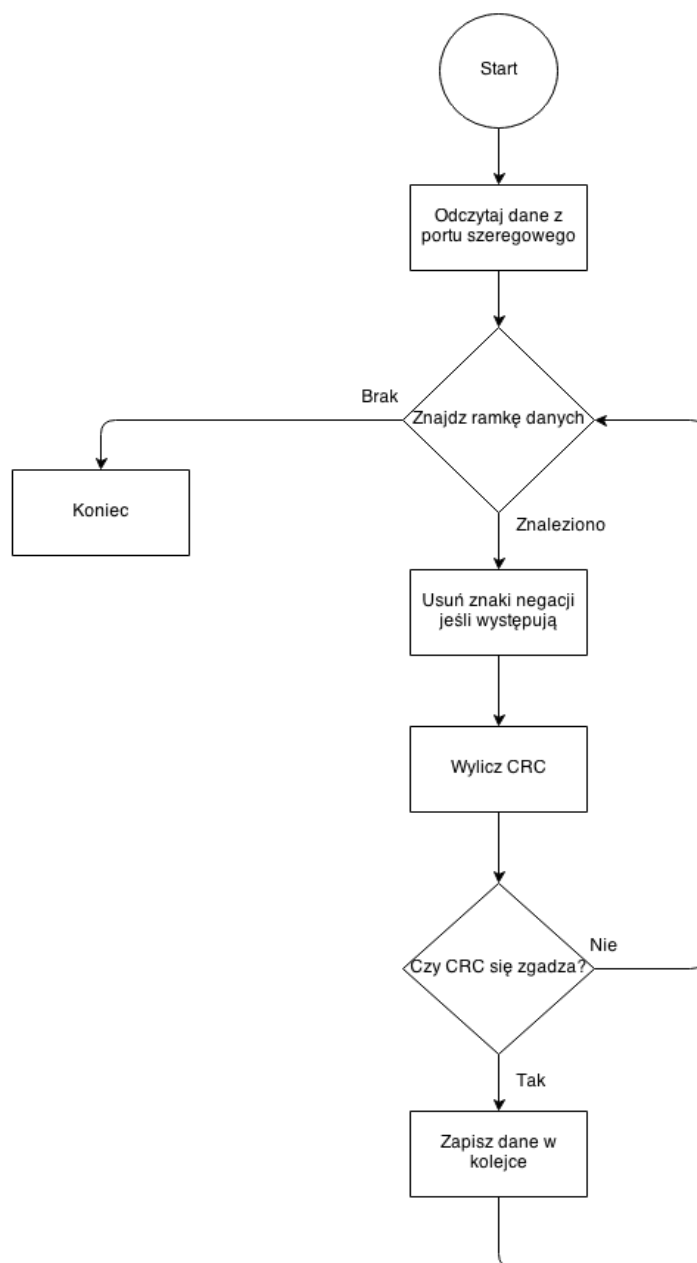
- Bajt startu – 0x10,
- Bajt stopu – 0xFF.

Dodatkowo w ramce tej zamieszczany jest 8 bitowy CRC w celu możliwości sprawdzenia czy odebrane dane są poprawne. Z wiadomych przyczyn w ramce danych mogą się znajdować bajty początku lub końca ramki dlatego zastosowano dodatkowy znak do ich negowania – 0x92 czyli „/”. Pozwala to na wyeliminowanie podstawowej przyczyny związanej z źle odebraną ramką. Ramka zawierająca dane nie zawiera informacji o ilości przesyłanych danych. Jedynie zawiera identyfikator danej. Nie powoduje to problemów w komunikacji, gdyż algorytm przetwarzania danych jest odpowiednio przygotowany do przetworzenia takich ramek.

Ze względu na duże prędkości transmisji nie stosowano w tym przypadku priorytetów dla danych wysyłanych tzn. dane zawierające informację o błędach są wysyłane w takiej kolejności jakiej trafiły do bufora danych. Do kolejkowania danych zastosowano kolejkę FIFO. Poniżej zamieszczono wykonane algorytmy odbioru i wysyłania danych. Algorytmy zostały przetestowane przy pomocy medium Bluetooth. Zapewniono poprawną komunikację.



Rysunek 2.
Kompresja danych



Rysunek 1. Dekompresja danych

1.4. Transport obrazu oraz dźwięku drogą radiową na jednostkę GCS.

Ze względu na duże zapotrzebowanie pasma przepustowego w celu przesyłania obrazu oraz dźwięku drogą radiową oparto się na WLAN`ie oraz protokole UDP. Połączenie to umożliwi w znacznym stopniu zwiększyć prędkość transmisji obrazu a małe straty pakietów nie powodują znacznych zauważalnych błędów. Dopiero w sytuacji utraty połączenia między GCS a platformą można zauważyć brak przesyłanego obrazu. Należy wspomnieć, iż zastosowanie komunikacji za pomocą wifi obarczone jest zastosowaniem anten kierunkowych oraz ciągłym śledzeniem pojazdu w przypadku komunikacji w terenie otwartym. W przypadku terenu zamkniętego przesyłanie obrazu jest uzależnione od mocy nadajnika oraz ilości metalowych elementów w konstrukcji budynku.

Zastosowanie protokołu UDP dodatkowo było podyktowane zastosowaniem pakietu gstreamer do przesyłania obrazu. Umożliwia on w łatwy sposób przesyłanie danych po tym protokole. Poniżej pokazano przykład przesyłania obrazu z kamery po UDP oraz odbiór danych w urządzeniu i wyświetleniu ich na monitorze komputera:

Źródło video: `gst-launch v4l2src device=/dev/video0 ! 'video/x-raw-yuv,width=640,height=480' ! x264enc pass=qual quantizer=20 tune=zerolatency ! rtpH264pay ! udpsink host=224.1.1.1 port=1234 auto-multicast=true`

Odbiorca video: `gst-launch udpsrc multicast-group=224.1.1.1 auto-multicast=true port=1234 ! "application/x-rtp, payload=127" ! rtpH264depay ! ffdec_h264 ! xvimagesink sync=false`

Powyższy zapis można rozumieć jako wysłanie ramki z obrazem na multicast czyli ramka ta trafi do wszystkich urządzeń w sieci na port 1234. Urządzenie, które będzie miało otwartą na tym porcie aplikację gst-launch z włączonym odbiorem multicast'a to odbierze ramkę wideo, która zostanie wyświetlona na ekranie monitora.

Implementację odbioru danych w GCS aktualnie jest w fazie projektowej. Etap ten jest znacznie skomplikowany.

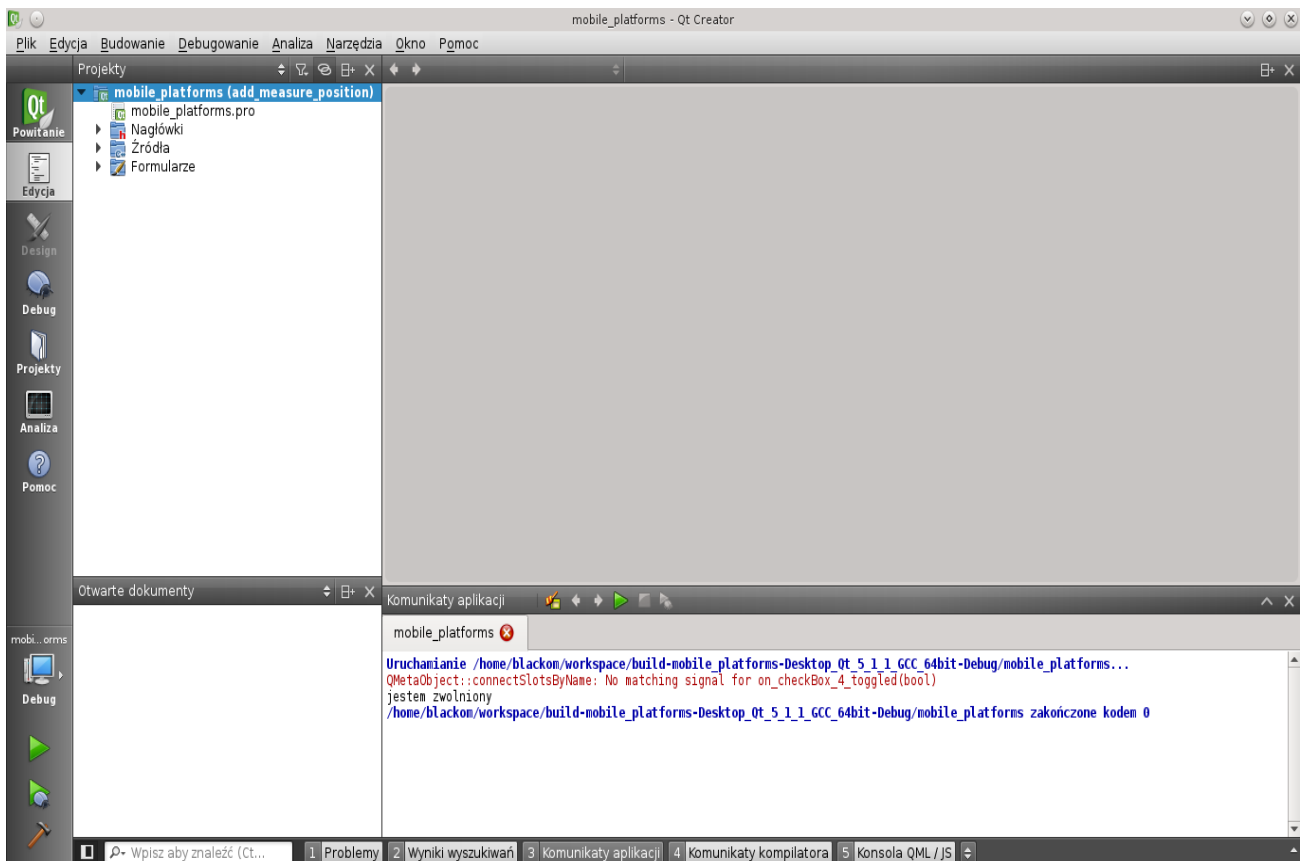
2. Sterowanie platformą mobilną z jednostki GCS.

2.1. Określenie środowiska programistycznego.

Głównym celem wyboru środowiska programistycznego było uzyskanie aplikacji wieloplatformowej. Jednym z ciekawych środowisk, które umożliwiają spełnienie wyżej wymienionego założenia jest Qt. Dostarcza ono bardzo przyjaznego w użyciu Środowiska developerskiego pokazanego na rysunku 3. Umożliwia ono pisanie kodu w C++ oraz tworzenie Aplikacji graficznej. Dodatkowo dostarcza w pełni zautomatyzowany proces kompilacji projektu. Należy wspomnieć, iż Qt posiada masę gotowych bibliotek ułatwiających pracę z urządzeniami wyjścia/wejścia oraz posiada bardzo rozbudowaną pomoc i opis poszczególnych funkcji. Minusem środowiska jest jego instalacja, która wymaga od użytkownika zmian w plikach konfiguracyjnych systemu aby dołączyć dodatkowe moduły lub biblioteki.

Qtcreator – jest to wspomniane wyżej narzędzie developerskie umożliwiające w łatwy sposób projektować aplikacje graficzne. Charakteryzuje się ono podziałem na kilka zakładek, które widać z lewej strony. Posiada możliwość debuggowania aplikacji w czasie rzeczywistym, co bardzo się przydaje podczas pisania skomplikowanych algorytmów związanych z przetwarzaniem danych.

Do kompilacji korzysta z qmake'a. Tworzy on plik Makefile na podstawie zapisanych w pliku *.pro danych. Plik ten jest głównym plikiem projektu. Zawiera informacje o plikach wchodzących w skład projektu oraz bibliotekach wykorzystywanych w projekcie np. QSerialPort – biblioteka umożliwiająca dostęp do portu szeregowego komputera.



Rysunek 3. Środowisko developerskie Qt

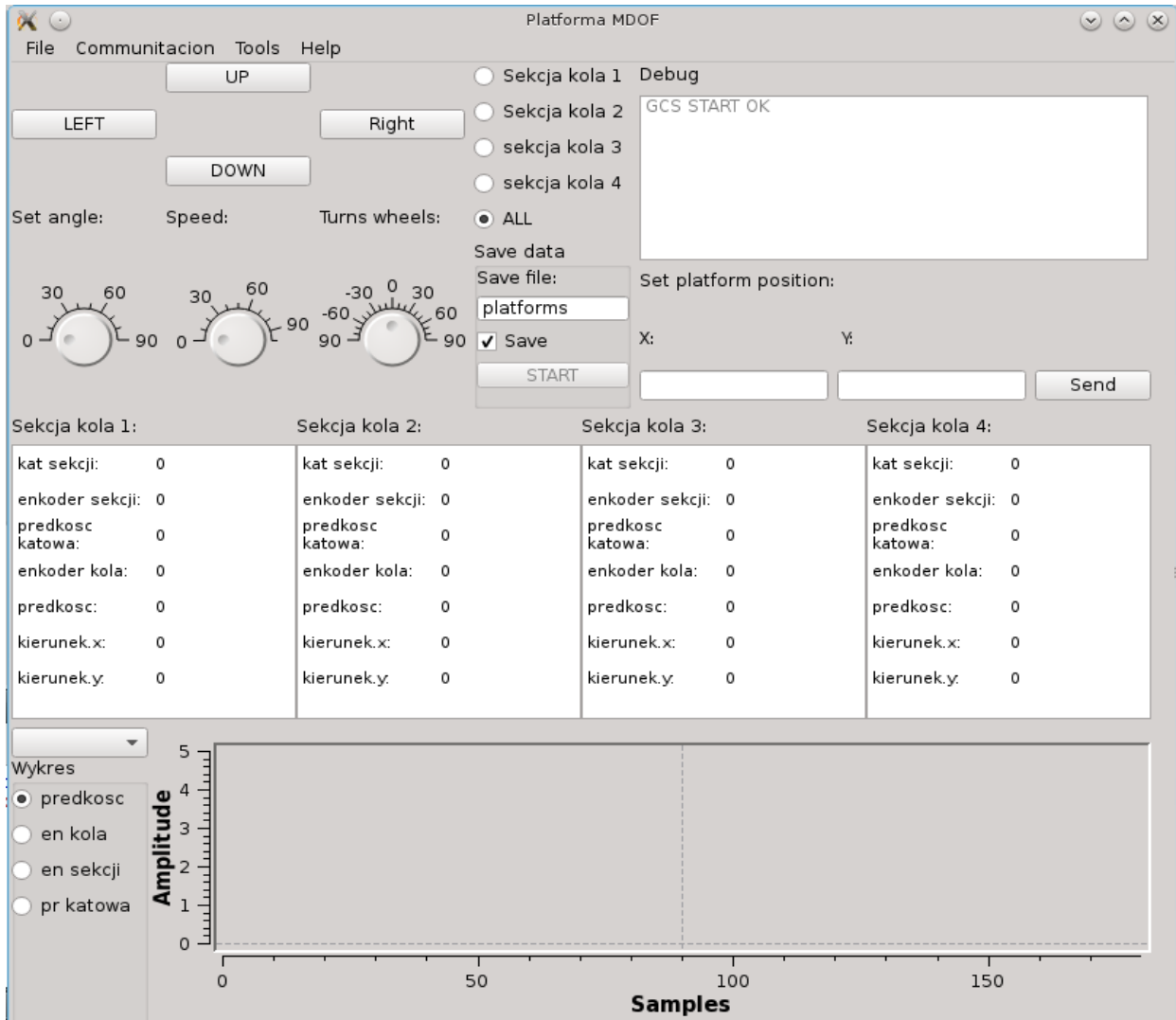
2.2. Projekt panelu sterowania.

Projekt panelu rozpoczęto od określenia wymagań co do sterowania platformą. Najważniejsze z nich to:

- Możliwość zmian nastaw regulatorów PID,
- streaming obrazu,
- wyświetlanie podstawowych danych: położenie, prędkość itd.,
- wyświetlanie informacji o błędach oraz działaniu platformy,
- zobrazowanie położenia radaru oraz odległości od obiektów,
- wyświetlanie w postaci graficznej podstawowych danych z platformy,
- panel sterowania platformą,
- możliwość określania położenia platformy w płaszczyźnie xy.

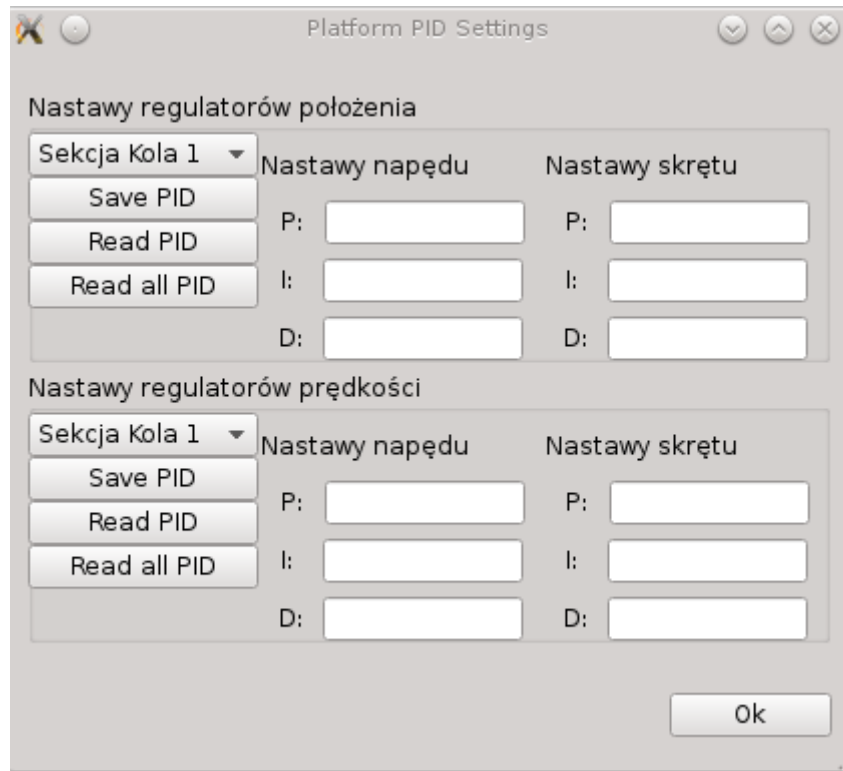
Następnie w projekcie panelu założono konstrukcję nowej platformy posiadającej większą liczbę napędów kół. Proces ten był konieczny w celu zmniejszenia ilości pracy w późniejszym procesie budowania nowej platformy.

Na rysunku 4 widać główne okno GCS. Zawarto tam większą część założeń projektu. W lewym górnym rogu znajduje się panel ręcznego sterowania platformą. Możliwe jest tutaj nastawienie odpowiedniej prędkości oraz maksymalnego skrętu platformą. Po prawej stronie widnieje lista wyboru do której sekcji skrętnej dane zostaną wysłane. Aktualnie wykorzystywana jest opcja ALL, gdyż platforma dysponuje jedną osią skrętną i napędową. Dalej jest okienko, w którym wyświetlane są komunikaty. Poniżej możliwość jest autonomicznego nastawienia pozycji platformy, w której platforma ma się znaleźć. Po środku okienka głównego wyświetlane są główne dane pochodzące z platformy tj. prędkość napędu koła, skręt, dane z enkoderów i pozycja napędu względem punktu początkowego platformy. Na samym dole jest możliwość wyświetlenia wykresu zmian podanych wyżej parametrów. Co w dużym stopniu ułatwia analizę działania np. nastaw regulatorów.



Rysunek 4 Główne okno GCS

Na rysunku 5 zamieszczono wygląd okienka umożliwiającego zmianę nastaw regulatorów PID w platformie z poziomu GCS. Jest to bardzo użyteczna opcja podczas fazy dostrajania platformy. Okienko to zostało podzielone na dwie części związane z nastawą regulatora PID dla położenia oraz prędkości. Jest to spowodowane tym, iż platforma ma za cel poruszać się między wyznaczonymi punktami lub dojeżdżać do określonego punktu ustalonego w GCS. Gdyby nie było jednej albo drugiej regulacji platforma pozbawiona byłaby kontroli swojej pozycji lub prędkości. Co wiąże się z utratą orientacji w otoczeniu.



Rysunek 5 Nastawy regulatorów PID

3. Sztuczna inteligencja:

3.1. Określenie wymagań sprzętowych.

Aby móc określić wymagania sprzętowe, musieliśmy najpierw rozważyć możliwe alternatywy obliczania algorytmów i związane z nimi ograniczenia. Wybór sprowadził się do dwóch opcji:

- przetwarzanie danych i obliczanie sterowania na platformie

Zaletą tej opcji jest brak wymaganej komunikacji z platformą, nie występują więc błędy związane z transmisją danych. Wadą jest konieczność obciążenia procesora dodatkowymi, złożonymi obliczeniami.

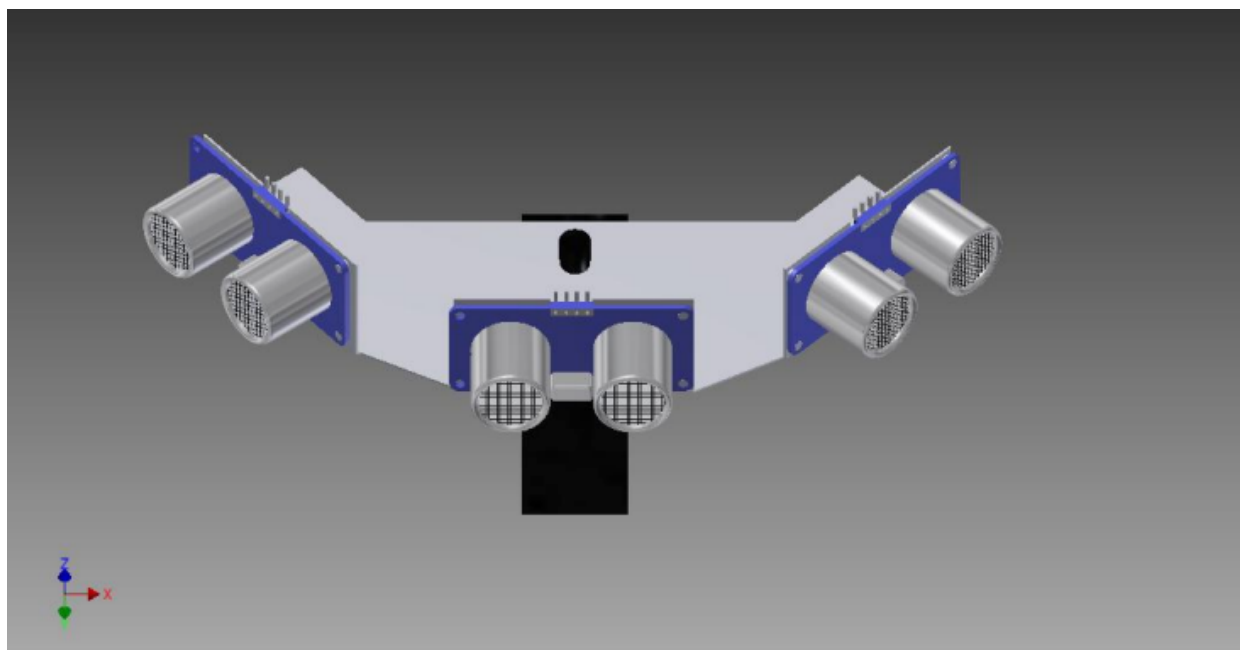
- przesył danych do jednostki GCS, wykonanie obliczeń na niej i przesłanie sterowania na platformę

Metoda obciążona błędami transmisji danych, jednak pozwalająca nam na wykonywanie obliczeń na oddzielnej jednostce, której parametry nie są aż tak ograniczone.

Zdecydowaliśmy się na drugą opcję, ponieważ dzięki niej możemy skupić się w większym stopniu na algorytmach nie myśląc aż tak o parametrach sprzętowych. Działanie platformy sprowadza się w tym wypadku do przesyłania pomiarów, odbioru rozkazów od jednostki sterującej i wykonania ich. W związku z tym wymagania sprzętowe związane z algorytmami nie są ograniczeniem, ponieważ możliwości sprzętowe GCSa są dużo lepsze niż platformy.

3.2. Wykonanie algorytmu omijania przeszkód na bazie kamery oraz radaru.

Nasza platforma mobilna jest wyposażona w radar, który został zwizualizowany poniżej:



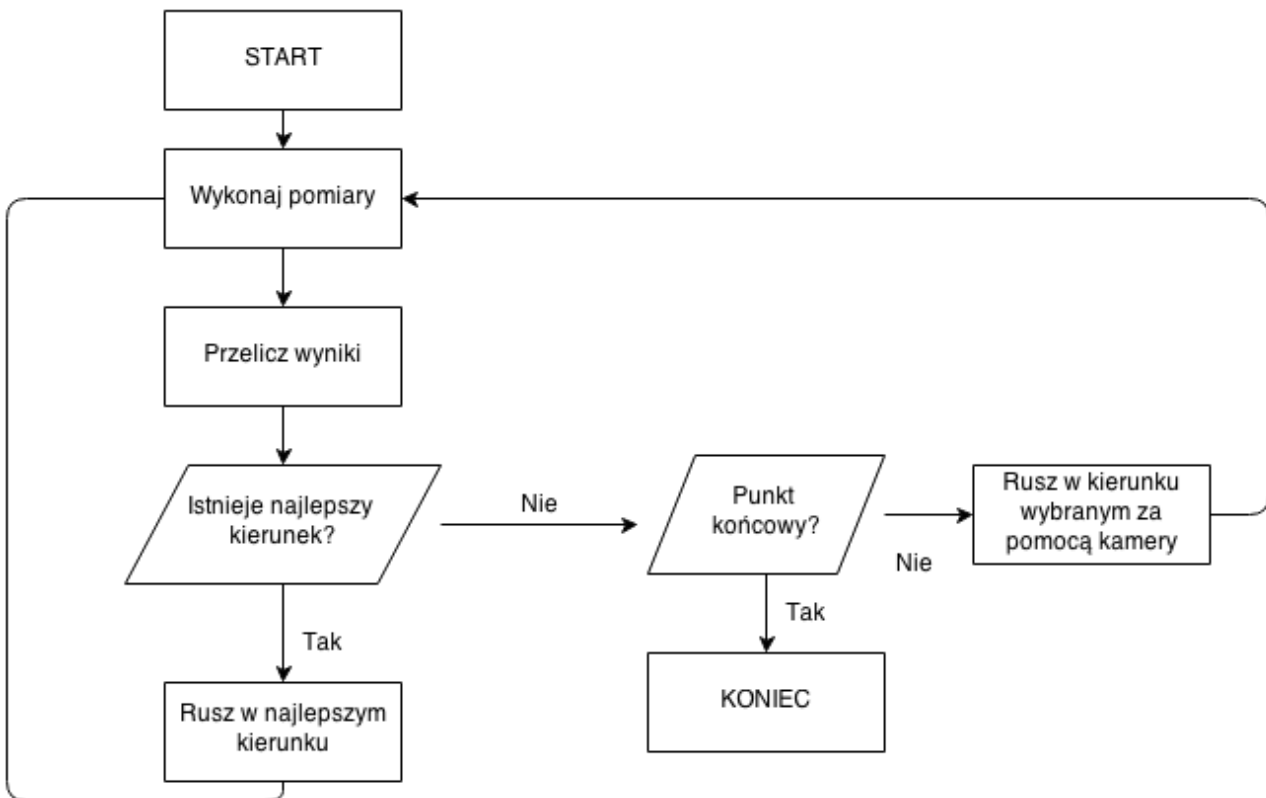
Rysunek 6 Radar platformy mobilnej

Do płytki pleksyglasowej przymocowano trzy czujniki, każdy pod innym kątem. Całość znajduje się na serwomechanizmie, który umożliwia obrót radaru i badanie dzięki temu większego obszaru wokół platformy. Czujniki zostały tak umieszczone, by nie interferowały ze sobą, ich maksymalny zasięg to 150 centymetrów.

Na podstawie tych pomiarów jesteśmy w stanie wykryć przeszkody i podjąć odpowiednie decyzje.

Omijanie przeszkód i sterowanie na płaszczyźnie dwuwymiarowej to znane już od dawna problemy, a algorytmy służące ich rozwiązaniu istnieją i są dobrze opisane. Aktualna platforma nie jest w stanie zmieniać kierunku w miejscu, co powinno nas zmusić do wybrania algorytmów dla robotów nieholonomicznych, jednak wybrany przez nas na początek algorytm korzysta z metody pól potencjałowych. W metodzie tej na pojazd działają tylko dwa rodzaje sił – przyciągająca, wytwarzana przez cel podróży, i odpychająca, wytwarzana przez każdą wykrytą przeszkodę. Platforma można więc porównać do elektronu, który porusza się w polu elektrycznym, i stąd też nazwa tej metody. Metoda ta nie wymaga dokładnej mapy całego terenu, co jest dużym ułatwieniem na początku. Jest ona jednak podatna na lokalne minima pola potencjalnego, przez które pojazd mógłby stanąć w miejscu przed osiągnięciem celu. W tym momencie zostanie wykorzystana kamera, która w przypadku braku ruchu pojazdu pomoże nam wybrać kierunek, w którym powinna ruszyć platforma by wydostać się z tej studni potencjału i ruszyć dalej.

Uproszczony schemat blokowy:



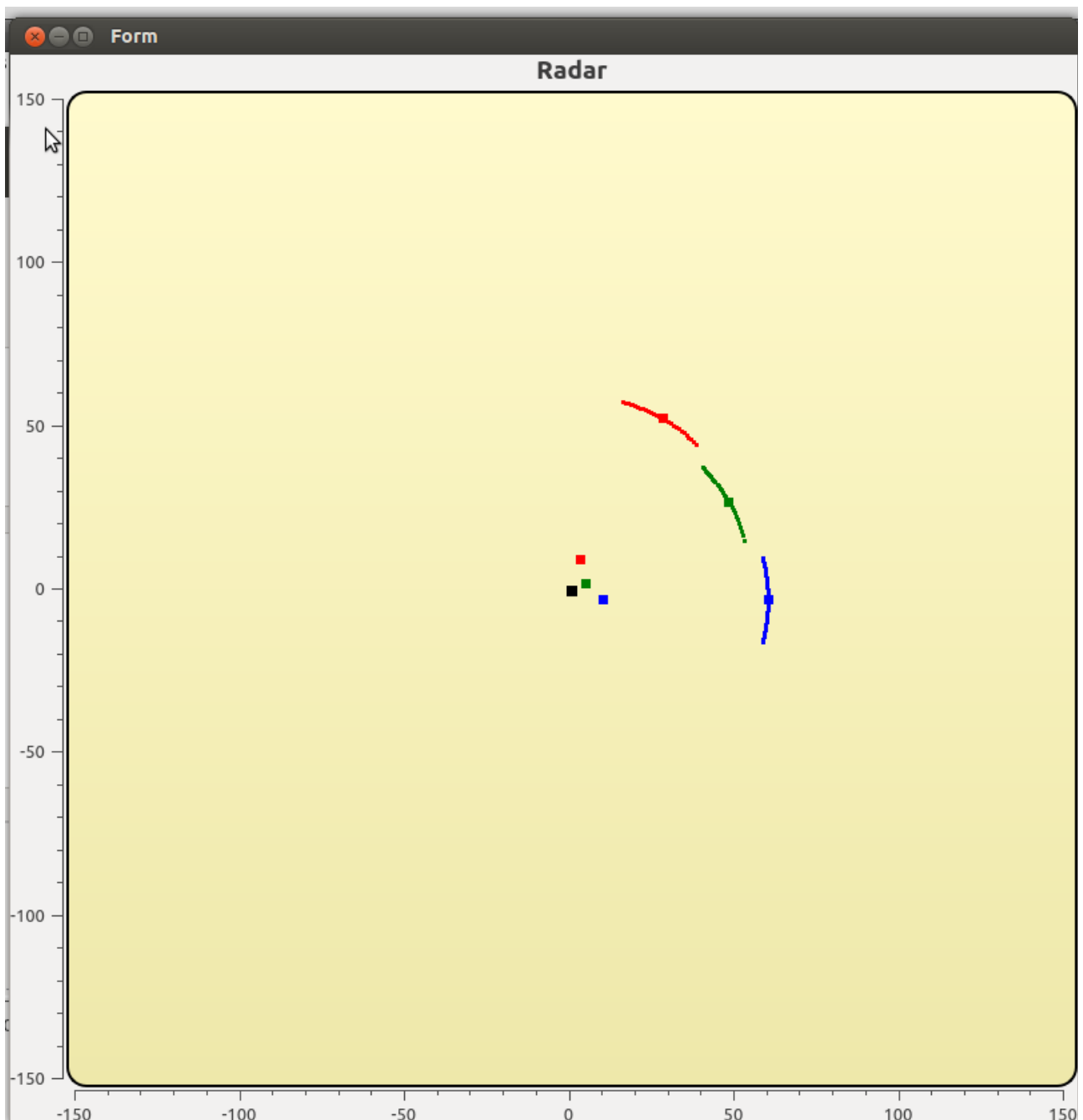
Rysunek 7 Schemat blokowy algorytmu

„Przelicz wyniki” oznacza obliczenie wartości „siły pola” działającego na pojazd w miejscach sąsiednich do jego aktualnego położenia, a najlepszy kierunek prowadzi do miejsca w którym ta wartość jest najmniejsza. Gdy wszystkie z nich są większe od aktualnej pojazd znajduje się w minimum, jednak trzeba określić czy jest to minimum lokalne czy globalne. Gdy nie jest to punkt docelowy należy to miejsce opuścić wbrew temu co sugeruje algorytm – po przejechaniu pewnej odległości istnieje szansa, że pojazd nie powróci do niego ponownie. Gdyby tak się jednak stało należy spróbować ponownie wyjechać, tym razem w innym kierunku.

Prace nad tą fazą projektu ciągle trwają, testowanie działania tego algorytmu i jego ewentualną zmianę na inny nastąpi w momencie, gdy mapowanie terenu będzie działało bezbłędnie.

3.3. Tworzenie mapy obszaru.

Na podstawie odczytu czujników należało stworzyć i narysować mapę terenu, wykrytą przez pojazd. Została napisana aplikacja, która pobiera odległości odczytane przez czujniki i stopień skrętu radaru a następnie rysuje je na płaszczyźnie. Poniżej znajduje się zrzut ekranu prezentujący działanie tego programu rysującego:



Rysunek 8 Zobrazowanie działania radaru

Czarna kropka oznacza oś obrotu, 3 kolorowe kropki tuż przy niej symbolizują nasze trzy czujniki. Na zrzucie ekranu widać przykładowe dane, oznaczone za pomocą kolorów odpowiadających kolorom czujnika który je zarejestrował. Ponieważ „kąt widzenia” każdego czujnika wynosi 30 stopni nie jesteśmy w stanie określić dokładnego położenia wykrytego obiektu. Program rysuje łuk, na którym może znajdować się przeszkoda, i zaznacza jego środek dla zwiększenia czytelności wizualizacji.

Aplikacja jest tak napisana, że prezentowane dane można dowolnie przesuwać, powiększać dany fragment i obliczać odległość między dwoma dowolnie wybranymi punktami. Przesyłany wektor danych może zawierać nie tylko odczyty radaru i jego kąt skrętu, lecz również dane przeliczone z uwzględnieniem aktualnego położenia i orientacji platformy w płaszczyźnie. Dzięki temu będziemy w stanie rysować wykryte przez pojazd przeszkody i ich położenie nie tylko względem aktualnej pozycji lecz punktu startowego. Aplikacja ta została już zintegrowana z GCS, jej dalszy rozwój będzie ciągle trwał.

7. Budżet oraz harmonogram zakupów.

Przewidziano zakup specjalistycznego sprzętu, który pozwoli na wykonanie projektu i późniejsze zastosowanie zakupionej aparatury elektronicznej w innowacyjnych projektach Międzywydziałowego Koła Naukowego High Flyers.

Nazwa	miesiąc zakupu	Cena
Zestaw Komponentów elektronicznych	styczeń	1920,60 zł

Aktualnie zamówienie jest w fazie realizacji. W skład zamówienia wchodzi:

- NGX Technologies LPC4357-Xplorer++,
- mX-LPC1768-S - zestaw uruchomieniowy,
- mX-LPC1343-S - zestaw uruchomieniowy,
- Embest SABRE Lite Development Kit.

8. Wnioski.

- Zrealizowano cel projektu w 50 procentach. Udało się zrealizować protokół transportu danych i błędów, sterowanie platformą oraz połączenie poprzez wifi. Aktualnie w fazie projektowej jest przesyłanie obrazu i wyświetlanie go w jednostce GCS.
- Dzięki pracy nad połączeniem radaru oraz kamery udało się uzyskać algorytm sterowania pomiarem odległości od obiektów za pomocą radaru, aktualnie w fazie projektowej jest wdrażanie określania obiektów za pomocą kamery oraz synchronizacja danych z kamery z danymi z radaru.
- Realizacja poszczególnych etapów projektu odbywała się bez opóźnień, poza transportem obrazu i dźwięku oraz algorytmu omijania przeszkód.
- Wyniki uzyskane w projekcie pozwoliły udoskonalić skuteczność prac w kolejnych projektach Koła Naukowego.
- Praca w projekcie nauczyła jego uczestników umiejętności pracy zespołowej oraz udoskonaliła wiedzę z zakresu budowy pojazdów mobilnych, sposobu sterowania, tworzenia algorytmów do autonomicznej jazdy platformy, działania czujników oraz innych urządzeń pomiarowych.
- Większość członków projektu sumiennie i rzetelnie wykonali swoje zadania, dzięki temu wyniki projektu są wiarygodne i można się na nich opierać w dalszych projektach.

9. Publikacja wyników.

Adres miejsca publikacji wyników projektu w Internecie: www.uav.polsl.pl