

Politechnika Wrocławska

Wydział Elektroniki, Fotoniki i Mikrosystemów

KIERUNEK: Automatyka i Robotyka (AIR)

PRACA DYPLOMOWA INŻYNIERSKA

TYTUŁ PRACY:
Interakcja smartfona z otoczeniem
z wykorzystaniem platformy mobilnej

AUTOR:
Mateusz Zabolski

PROMOTOR:
Dr inż. Robert Muszyński,
Katedra Cybernetyki i Robotyki

Spis treści

1	Wstęp	3
2	Interakcje robota	5
2.1	Funkcjonalność robotów	5
2.2	Możliwości interakcji robota ze środowiskiem	6
2.3	Dostępne roboty interaktywne	7
2.4	Wybrane rozwiązanie	8
3	Konstrukcja mechaniczna robota	11
3.1	Specyfikacja korpusu	11
3.2	Układ napędowy	11
4	Układ elektroniki	15
4.1	Układ komunikacji zewnętrznej	15
4.2	Układ komunikacji wewnętrznej	15
4.3	Układ zasilania	16
5	Aplikacja mobilna	19
5.1	Narzędzia	19
5.1.1	Środowisko programistyczne	19
5.1.2	Biblioteki i technologie	21
5.2	Implementacja	22
5.2.1	Uprawnienia	23
5.2.2	Realizacja połączenia bluetooth	24
5.2.3	Analiza obrazu	24
5.2.4	Szukanie zadanego obiektu	25
5.2.5	Rozpoznawanie mowy	27
6	Testy	29
6.1	Testy platformy mobilnej	29
6.2	Testy aplikacji	30
6.3	Interakcja z otoczeniem	30
7	Zakończenie	33
	Bibliografia	35
	Spis rysunków	37

Rozdział 1

Wstęp

Wraz z postępem technologicznym rośnie znaczenie wykorzystania robotów w różnych dziedzinach życia. Coraz częściej zdarza się, że roboty przebywają w bezpośrednim otoczeniu człowieka, a nawet współpracują z nim. Tematem pracy jest budowa platformy mobilnej oraz integracja jej ze smartfonem, co umożliwi mu interaktywne reagowanie na otoczenie. W ramach projektu zostanie również stworzona dedykowana aplikacja, która wspomogę analizę otoczenia oraz dostosuje działanie platformy w zależności od kontekstu.

Interakcję robota rozumianą w bardzo szerokim kontekście definiujemy jako oddziaływanie robota na przedmioty lub ludzi, a w drugą stronę przedmiotów oraz ludzi na robota [Wike]. Każdy zmienny element otoczenia może wywołać odpowiednią reakcję robota, zgodnie z zaprogramowanymi wcześniej instrukcjami. Interakcja z otoczeniem jest dynamicznym procesem, który ulega modyfikacją w zależności od zmian w środowisku.

W przypadku interakcji z ludźmi, proces ten staje się znacznie bardziej skomplikowany, a próby osiągnięcia efektywnej komunikacji sięgają wielu lat wstecz. Pierwszymi robotami współpracującymi z człowiekiem były, tak naprawdę, maszyny przemysłowe wyprodukowane przez General Motors w 1961 roku. Nie reagowały one na polecenia człowieka, lecz wykonywały powtarzalne zadania. Przełomem w komunikacji robotów z ludźmi był robot WABOT-1 [Hum], zbudowany w Japonii w 1972 roku. Słynął głównie z poruszania się, chwytania przedmiotów i, co najważniejsze, z prostych form komunikacji z człowiekiem.

Obecnie większość fabryk korzysta z robotów, gdzie istotnym aspektem jest interakcja z człowiekiem. Przykładem mogą być roboty Baxter i Sawyer od firmy Rethink Robotics [Piq18], które zostały zaprojektowane z myślą o bezpieczeństwie interakcji z ludźmi, wyposażone w czujniki siły. Rejestrują one opór przy każdym kontakcie fizycznym, aby móc odpowiednio się dostosować i uniknąć kolizji dzięki kamerom wizyjnym.

Aktualnie smartfony stały się nieodłącznym elementem codziennego życia większości ludzi. Służą nam nie tylko do rozrywki, ale również do komunikacji z innymi ludźmi oraz do załatwiania codziennych spraw, takich jak opłacanie rachunków. Można również wymienić inne funkcjonalności, takie jak robienie zdjęć, notatek i tym podobne. Mimo tak wielu funkcjonalności, które oferuje smartfon, większość nie widzi w nim innego zastosowania, niż do codziennego użytku. Powodem jest fakt, że smartfon potrafi oddziaływać na otoczenie jedynie „dźwiękiem i obrazem”.

Wśród funkcjonalności smartfona brakuje metody przemieszczania się. Taką funkcjonalność mogłaby zapewnić mu platforma mobilna, na której byłby zamocowany. Przy takiej ilości sensorów, które posiada smartfon, pomysłów na wykorzystanie tak powstałego robota może być niezwykle dużo. Jedynym ograniczeniem pozostaje nasza kreatywność. Telefon można potraktować jako mikrokontroler, do którego nie możemy bezpośrednio fizycznie podłączyć dodatkowych efektorów czy sensorów. Jednak może on komunikować się z innymi mikrokontrolerami posiadającymi funkcjonalności, których smartfon nie posiada. W technologie takie jak Bluetooth, Wi-Fi czy NFC, które posiada smartfon można również wyposażyć mikrokontroler do wspólnej komunikacji.

Celem pracy jest skonstruowanie platformy mobilnej, która da możliwość przemieszczania się smartfonowi. Praca ta obejmuje również opracowanie aplikacji, która połączy te dwa urządzenia i umożliwi interaktywne reagowanie na otoczenie. Układ pracy jest następujący. W rozdziale drugim omówione zostały możliwości interakcji z otoczeniem. Kolejno, w rozdziałach trzecim i czwartym skoncentrowano się na konstrukcji platformy mobilnej, obejmującej aspekty mechaniczne i układ elektroniczny. W rozdziale piątym przedstawiono działanie aplikacji mobilnej oraz wymieniono użyte narzędzia i biblioteki do jej stworzenia. Rozdział szósty zawiera opis przeprowadzonych testów oraz napotkanych problemów w trakcie ich realizacji. Ostatecznie, w rozdziale siódmym przedstawione zostało podsumowanie całej pracy.

Rozdział 2

Interakcje robota

Stworzenie robota WABOT-1 zapoczątkowało różnorodne wyobrażenia o tym, jak może wyglądać interakcja robotów ze środowiskiem. Spektrum tych wyobrażeń jest szerokie – od najbardziej realistycznych, w których roboty pomagają nam w codziennych czynnościach, po te najbardziej skrajne scenariusze, gdzie roboty przejmują nad nami kontrolę. Poniżej zostaną omówione funkcjonalności, w jakie wyposażone są roboty oraz przykładowe konstrukcje robotów interaktywnych.

2.1 Funkcjonalność robotów

O funkcjonalności robota decydują peryferia w jakie został on wyposażony: sensory i efektory [Wikf]. Obecnie roboty wyposażane są w różnorodne sensory pozwalające im wychwytywać coraz więcej sygnałów zewnętrznych. A dzięki efektorom, takim jak chwytaki, koła, wyświetlacze czy głośniki, mogą one adekwatnie reagować na zmieniające się warunki środowiskowe. Często budowa robotów ma na celu symulację naszych zmysłów lub części ciała, inspirowane anatomią człowieka. Na przykład kamera wizyjna imituje ludzkie oko a mikrofon – ucho. Mimo że roboty mogą mieć zdolności w zakresie „widzenia i słyszenia” jak ludzie, to wyróżniają się również posiadaniem czujników, dla których nie znajdziemy ludzkiego odpowiednika. Przykładem są dalmierze ultradźwiękowe, które informują o obecności obiektów w określonej odległości – coś, czego człowiek nie potrafi stwierdzić. Innym przykładem jest kamera termowizyjna, pozwalająca na ocenę temperatury na odległość, w przeciwieństwie do ludzkiego zmysłu dotyku, który wymaga bezpośredniego kontaktu. Takie cechy sprawiają, że roboty mogą działać lepiej czy też szybciej oraz wykonywać zadania, których ludzie nie są w stanie zrealizować. Warto też wspomnieć, że roboty mogą być znacznie szybsze i silniejsze niż ludzie, zastępując pracę wielu osób i wykonując zadania dokładniej oraz efektywniej. Oczywiście na możliwości robota wpływają także efektory. Pozwalają one robotowi na przemieszczanie się w wyznaczonym kierunku. Dzięki nowoczesnej technologii często mogą przenosić obiekty cięższe niż człowiek i robić to znacznie szybciej. Roboty potrafią również reagować na swoje otoczenie przy użyciu manipulatorów, które w wielu przypadkach naśladują wygląd i funkcjonalność ludzkiej dłoni. Główną zaletą manipulatorów jest ich zdolność do działania z milimetrową precyzją i ustaloną prędkością. Bez odpowiedniego doświadczenia, ludzkie ręce nie są w stanie osią-

gnąć takiej samej dokładności w ruchach [Pul]. Innymi elementami, które pomagają robotowi w interakcji, są zdecydowanie ekran czy głośnik. Z ich pomocą może odpowiadać na zadane pytania oraz informować o zaistniałych sytuacjach. Niemniej jednak praca robota zawsze wiąże się z interakcją z człowiekiem. Obecnie prawie wszystkie fabryki zatrudniają pracowników, którzy współtworzą zespoły razem z robotami [zro20]. Podobnie można zaobserwować obecność robotów w szkołach czy domach opieki, gdzie pełnią one funkcje wspomagające w procesie edukacji lub oferują wsparcie osobom samotnym. Dlatego tak istotna jest sprawnie działająca interakcja między robotami a ludźmi, umożliwiającą osiągnięcie celów efektywniej i precyzyjniej.

2.2 Możliwości interakcji robota ze środowiskiem

Interakcja z robotem może przybierać różne formy, które pozwalają na przekazywanie informacji. Tak samo interakcja charakteryzuje się różnorodnymi sposobami jej inicjacji. Niektóre mogą być wywołane przez człowieka, takie jak: mówienie, naciskanie przycisków czy wskazywanie i dostarczają maszynom interaktywnym danych o różnym stopniu dokładności. Inne wynikają z zmian otoczenia, które robot obserwując, wykrywa.

Najprostszym i najbardziej oczywistym sposobem komunikacji z robotem jest użycie panelu sterowania, który umożliwia programowanie ruchów, określanie zadań czy monitorowanie działania maszyny [Wike]. To podejście, choć zrozumiałe dla robota, wymaga od człowieka sporej wiedzy na temat programowania. Takie interfejsy najczęściej znajdują zastosowanie w robotach przemysłowych, gdzie priorytetem jest precyzja wykonania zadania, więc inne formy interakcji mogłyby okazać się niewystarczające. Na przykład komunikacja głosowa nie dostarczy maszynie tak sprecyzowanych informacji, jakie może otrzymać za pośrednictwem kodu programu. Jednakże taka komunikacja ma istotną zaletę – prostotę w obsłudze. Doskonając komunikację międzyludzką doprowadziliśmy ją niemal do perfekcji. Pozostaje, więc tak zaprogramować robota, aby postępował podobnie jak człowiek.

Interakcja jednak może być zainicjowana przez robota i nie wymagać od użytkownika żadnych działań. Robot, analizując środowisko, może samodzielnie interpretować, co ma zrobić. W takim przypadku oprogramowanie musi być zdolne radzić sobie z różnymi scenariuszami i wiedzieć, jak reagować jedynie na podstawie obserwacji zmian w otoczeniu [Fra23]. Dzięki temu robot interaktywnie monitoruje środowisko, lecz nie komunikuje się z nim bezpośrednio.

Największym wyzwaniem wydaje się być interakcja polegająca na reagowaniu na ludzkie emocje. Robot, który jest w stanie zareagować na smutek człowieka, może nawiązywać interakcje mające na celu pocieszenie go w trudnych chwilach. Wydedukowanie ludzkich emocji z dostępnych danych sensorycznych jest wyjątkowo skomplikowane. Sam obraz często nie wystarcza, konieczne jest wykorzystanie dodatkowych czujników, aby prawidłowo ocenić emocje człowieka.

Wymienione sposoby komunikacji, takie jak interpretacja obrazu, głosu czy posługiwanie się panelem sterowania, można łączyć, modyfikować i wykorzystywać na różne sposoby, by maksymalnie ułatwić ludziom życie.



Rysunek 2.1 Robot Paro [Med17]

2.3 Dostępne roboty interaktywne

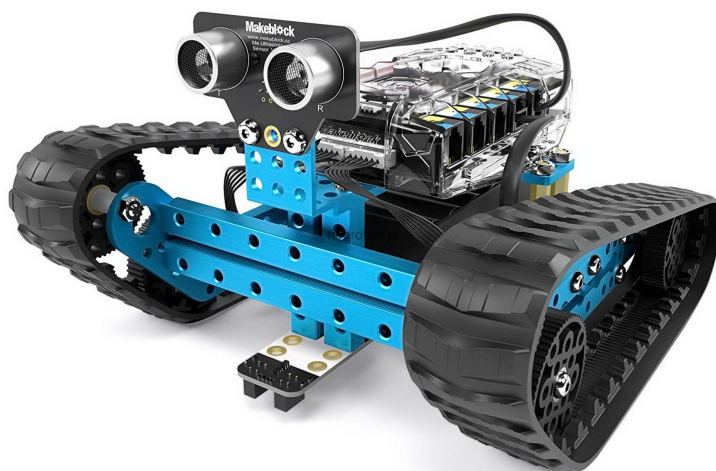
Aktualnie w licznych miejscach można znaleźć różnorodne typy robotów, z których wiele jest interaktywne i reaguje na różne bodźce z otoczenia. Choć przemysł pozostaje głównym obszarem zastosowania interaktywnych maszyn, ich obecność jest coraz bardziej zauważalna także poza fabrykami — w domach, biurach czy szkołach. Tam roboty znajdują zastosowanie w edukacji, terapii oraz wsparciu w realizacji codziennych zadań.

Przykładem mogą być roboty zabawkowe, takie jak Cozmo i Vector [Ank], interaktywnie spędzają czas z dziećmi rozpoznając ich twarze oraz emocje. Podstawowym sposobem interakcji z tymi robotami jest projektowanie gier i zadań, tworząc je w ich oprogramowaniu. Ponadto reagują one na komendy głosowe oraz emocje człowieka, które rozpoznają za pomocą analizy obrazu z kamery lub dźwięku z mikrofonu, wyrażając swoje reakcje poprzez uśmiech, odgrywania wesołych dźwięków lub podskoki.

Innym przykładem są roboty asystujące w opiece, takie jak Paro [PAR], przedstawiony na rysunku 2.1, który specjalizuje się w pomocy osobom starszym, szczególnie tym z demencją. Paro reaguje, nie tylko na głos, ale także na dotyk. Zaprojektowano go tak, aby przypominał małą fokę i interaktywnie odpowiadał wsparciem emocjonalnym pacjentów, uspokajając ich i wyrażając zadowolenie, kiedy jest głaskany lub zachęcając do dalszych pieszczot.

W obszarze edukacji również znajdziemy wyzwania związane z reakcją na otoczenie, zwłaszcza ze względu na różnorodność emocji u dzieci. Podjęli się ich pracownicy firmy NeoRobot tworząc robota edukacyjnego Makeblock [Neob], który emituje dźwięki przy poprawnym zaprogramowaniu, uśmiecha się lub świeci na zielono, aby odzwierciedlić pozytywną reakcję na udaną realizację zadania. Robot został przedstawiony na rysunku 2.2. Jako panel sterowania do tego robota została użyta strona internetowa lub aplikacja na smartfona. Dzieci układają tam bloczki, wyznaczając kroki postępowania robota, gdzie od razu dostają informację zwrotną o pozytywnej lub negatywnej kompilacji robota.

Bee-Bot [TTS] to kolejny przykład robota edukacyjnego, zaprojektowanego specjalnie dla młodszych dzieci. Interakcja z nim odbywa się poprzez programowanie



Rysunek 2.2 Robot Makeblock [Neoa]

trajektorii ruchu za pomocą przycisków na jego grzbiecie. Robot sygnalizuje wykonanie zadań zmianą kolorów i emisją dźwięków. Dziecko metodą prób i błędów uczy się rozróżniać kierunki oraz rozwija logiczne myślenie dzięki zabawie z robotem.

Wymienione przykłady podkreślają, jak ważna jest interakcja robotów z otoczeniem. Dzięki niej możemy unikać niepotrzebnych zdarzeń, ułatwiamy sobie pracę i poprawiamy zdrowie psychiczne. Można stwierdzić, że to właśnie interakcja sprawia, że robot staje się więcej niż zwykłą maszyną.

2.4 Wybrane rozwiązanie

Naszym celem jest wyposażenie smartfona w dodatkową funkcjonalność umożliwiającą mu poruszanie się. Zapewni mu ją zaprojektowana w ramach tej pracy platforma mobilna. Smartfon wraz z tą platformą mobilną stanowi swego rodzaju robot mogący wchodzić w interakcję z otoczeniem.

Podczas projektowania robota została zwrócona uwaga na aspekty dotyczące robotów edukacyjnych i terapeutycznych, przywiązując wagę do prostoty mechanizmów interaktywnych. Robot został zaprojektowany tak, by jego obsługa była intuicyjna i nie wymagała zbyt wiele od użytkownika. Skoncentrowano się na podstawowych formach komunikacji, takich jak mowa czy zmiana otaczającego środowiska. Wykorzystując dostępne funkcjonalności robota zaprogramowano komunikację głosową z użytkownikiem oraz przemieszczanie się w reakcji na poczynione obserwacje otoczenia. Poniżej zostały przedstawione wszystkie interakcje, które udało się zrealizować.

Do interakcji z otoczeniem zdecydowano wykorzystać kamerę, głośnik, mikrofon oraz panel dotykowy. Opracowano trzy scenariusze, z których dwa: „śledzenie” i „szukanie” wykorzystują kamerę, głośnik i panel dotykowy do realizacji zadania, w którym robot porusza się w kierunku zadanego obiektu. Trzeci scenariusz, „roz-

mowa” używający głośnika i mikrofonu, służy do głosowej komunikacji z użytkownikiem.

W pierwszej i drugiej interakcji wykorzystano kamerę do lokalizacji zadanego obiektu oraz głośnika do informowania użytkownika o postępie realizacji zadania. Robot obraca się, skanując otoczenie kamerą w poszukiwaniu zadanego obiektu i wypowiadając komunikat o rozpoczęciu procesu. Po zlokalizowaniu przedmiotu robot przemieszcza się w jego kierunku, dopóki nie zbliży się na wystarczająco małą odległość, jednocześnie informując o znalezieniu obiektu. W scenariuszu „śledzenie” po przesunięciu zadanego obiektu robot kontynuuje jego śledzenie. Jeśli poszukiwany przedmiot znajdzie się poza zasięg widzenia, robot ponownie rozpocznie obracanie się i poszukiwanie go po pomieszczeniu.

Podobnie przebiega scenariusz „szukania” kilku przedmiotów różniących się kolorem. Różnica między obecnym scenariuszem a poprzednim polega na tym, że po znalezieniu przez robota jednego zadanego obiektu, przemieszcza się do innego, losowo wybranego miejsca. Następnie rozpoczyna poszukiwania kolejnego przedmiotu o innym kolorze.

Kolejnym zadaniem interakcji robota jest komunikowanie się z użytkownikiem za pomocą mikrofonu i głośnika, zrealizowane w scenariuszu „rozmowy”. Robot oczekuje na wypowiedź użytkownika i, gdy wykryje określoną frazę, odpowiada mu komunikatem, a następnie wykonuje powiązane z nią zadanie. Robot jest również zaprogramowany do reagowania na inne komendy, które inicjują krótką, werbalną interakcję. Po zakończeniu nasłuchiwanie i otrzymaniu odpowiedniej komendy, robot odtwarza plik z nagrany pożegnaniem.

Rozdział 3

Konstrukcja mechaniczna robota

Ważnym aspektem pracy jest element, na którym będzie się poruszał smartfon, przedstawiony na rysunku 3.1 – platforma mobilna. Celem jej skonstruowania jest stworzenie smartfonowi możliwości przemieszczania się, w tym interakcji z użytkownikiem. Przy projektowaniu platformy kluczową rolę odgrywała równomierna dystrybucja wagi i stabilność konstrukcji. Wszystkie elementy zostały rozmieszczone tak, aby zapewnić łatwe łączenie się z innymi komponentami bez ryzyka kolizji.

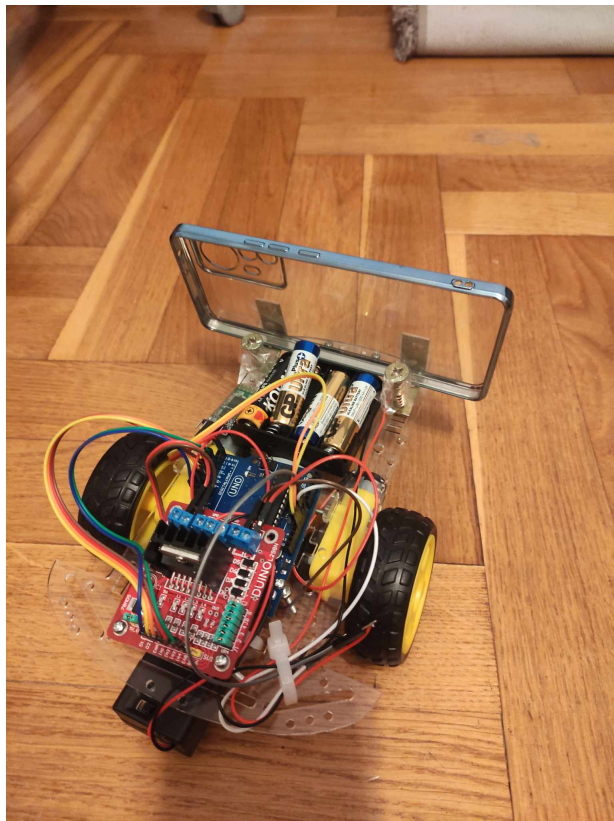
Podstawą robota jest przezroczysty korpus, do którego przymocowane są pozostałe elementy. W przedniej części robota podporę stanowi koło typu kastor. Bezpośrednio nad nim umieszczono mocowanie smartfona. Zastosowanie tego rozwiązania ma na celu zapobieganie ślizganiu się urządzenia oraz utrzymanie go w centralnej pozycji na robocie. Na środku platformy zlokalizowane są baterie bezpośrednio przykręcone do korpusu, a obok nich moduł Bluetooth. W części tylnej podporę stanowią dwa koła napędowe, nad którymi zamocowany jest mikrokontroler. Sterownik silników został zamontowany powyżej jednostki sterującej mikroprocesorem za pomocą dwóch długich śrub przykręconych do korpusu. Drugi zestaw baterii został przytwierdzony do spodu w części tylnej. Dla poprawy trakcji został również przytwierdzony dodatkowy balast przy kołach napędowych.

3.1 Specyfikacja korpusu

Wymiary podstawy mobilnej to 200 x 140 x 65 mm, co można zobaczyć na rysunku 3.2. Korpus wykonany jest z akrylu. Na rysunku 3.1 przedstawiono platformę mobilną z smartfonem, a na rysunku 3.3 znajduje się platforma mobilna z smartfonem.

3.2 Układ napędowy

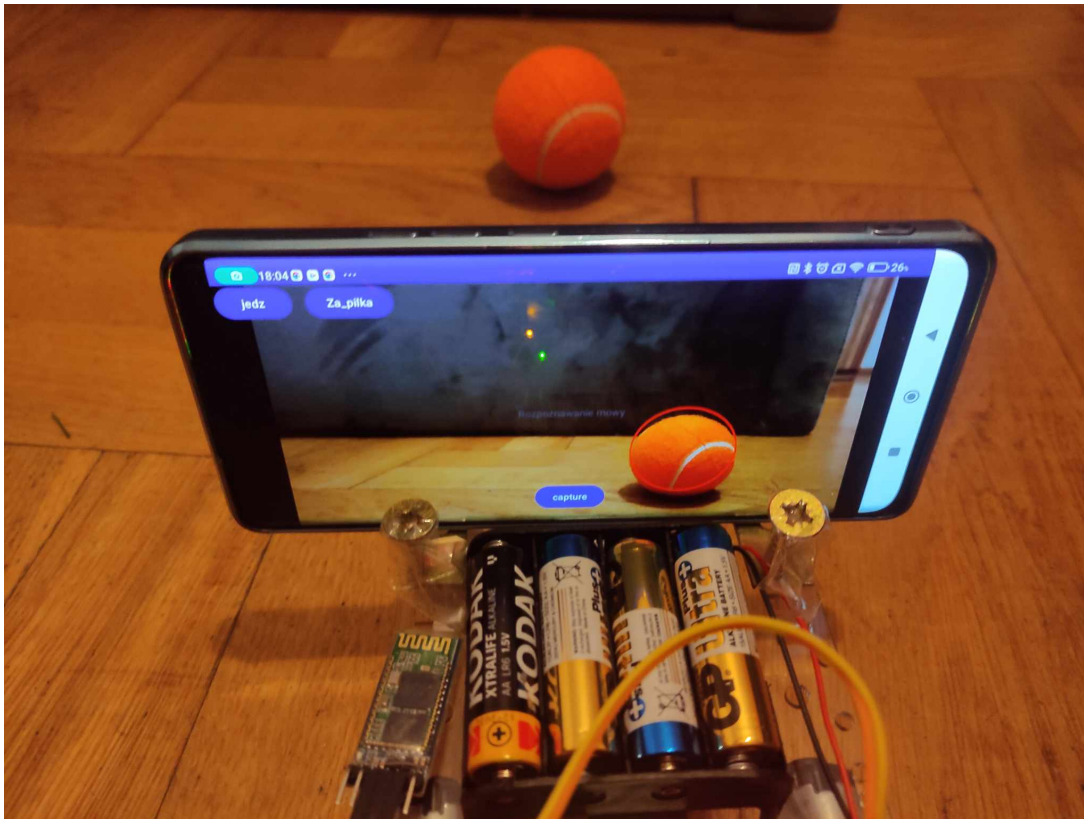
Prezentowana platforma mobilna ze smartfonem jest robotem klasy (2,0), stając klasycznym przykładem monocykla [TM18]. Układ napędowy składa się z dwóch silników i trzech kół. Tylne koła mają średnicę 65 mm i stanowią wyznacznik wysokości całego podwozia. Koło kastora ma średnicę 25 mm. Silniki są zamontowane



Rysunek 3.1 Kompletny robot z układem sterowania



Rysunek 3.2 Korpus robota z układem napędowym



Rysunek 3.3 Sposób montowania smartfona

pod główną płaszczyzną, a koła przyczepione tak, aby wystawały poza nią i nie blokowały się o nią. Po drugiej stronie platformy znajduje się koło kastera umożliwiające robotowi swobodne poruszanie się w różnych kierunkach. Działanie robota opisuje model kinematyki robota mobilnego klasy (2,0) w postaci ogólnej

$$\dot{q}_r = \begin{pmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{pmatrix} = \begin{bmatrix} \frac{\cos(\theta)}{2} & \frac{\cos(\theta)}{2} \\ \frac{\sin(\theta)}{2} & \frac{\sin(\theta)}{2} \\ \frac{-1}{L} & \frac{1}{L} \end{bmatrix} \begin{pmatrix} \eta_L \\ \eta_R \end{pmatrix},$$

gdzie q_r – współrzędne robota (\dot{x}_r, \dot{y}_r - położenie, $\dot{\theta}_r$ - orientacja), η_L, η_R – prędkości postępowe lewego i prawego koła, zaś L rozstaw kół.

Do platformy mobilnej zostały zamontowane dwa silniki DC o następujących parametrach [Botb]:

- moment obrotowy $0,8 \text{ kg} \times \text{cm}$,
- przekładnia 1:48,
- napięcie zasilania od 3 V do 6 V,
- prędkość obrotowa do ok. 200 obr/min,
- pobór prądu średnio 150 mA,
- wymiary 70 x 22 x 19 mm.

Rozdział 4

Układ elektroniki

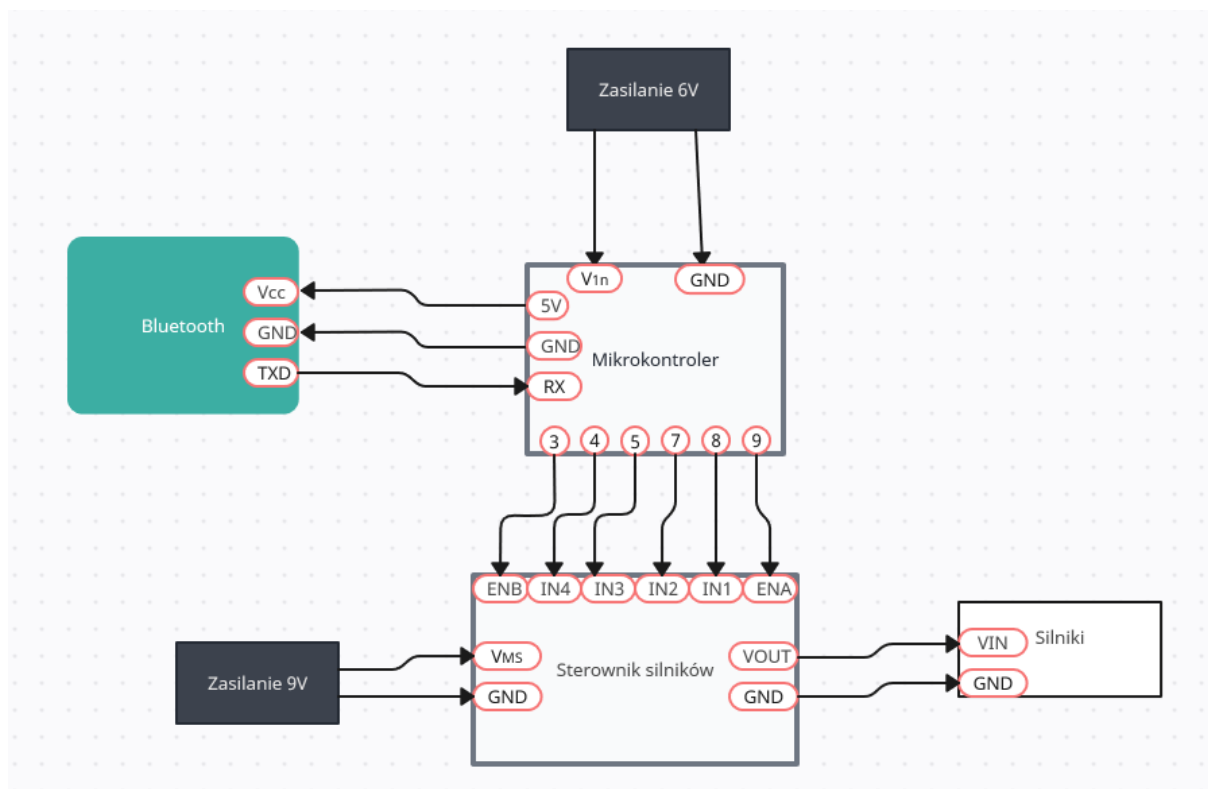
Układ elektroniczny platformy mobilnej składa się z klona Arduino UNO R3 [Net], sterownika silników, źródła zasilania oraz modułu Bluetooth HC-05 [Bota]. Do programowania mikrokontrolera wykorzystano środowisko Arduino [Wikb]. Schemat połączeń urządzeń elektronicznych został przedstawiony na rysunku 5.1. Mikrokontroler jest skonfigurowany do odbioru danych z modułu Bluetooth. Gdy dane są otrzymywane, mikroprocesor analizuje otrzymany sygnał, aby zidentyfikować odpowiednie komendy sterujące. Piny mikroprocesora są połączone ze sterownikami silników. Gdy stan pinów zostaje zmieniony na wysoki, sterowniki silników reagują na tę zmianę, dostosowując napięcie dostarczane do silników.

4.1 Układ komunikacji zewnętrznej

Do komunikacji platformy mobilnej ze smartfonem wykorzystywany jest moduł Bluetooth HC-05. Komunikacja odbywa się w sposób jednokierunkowy, gdzie moduł HC-05 oczekuje na dane przesyłane ze smartfona. Dioda, która miga co sekundę, sygnalizuje, że moduł nie jest połączony z żadnym urządzeniem. Po nawiązaniu poprawnego połączenia stałego, częstotliwość mignięć diody zmniejsza się do jednego mignięcia na 4 sekundy. Moduł jest podłączony do zasilania i uziemienia mikrokontrolera. Ponadto, pin TX (Transmitter) modułu HC-05, służący do przesyłania danych, jest podłączony do portu RX (Receiver) mikrokontrolera, który odbiera dane przesyłane ze smartfona.

4.2 Układ komunikacji wewnętrznej

Mikrokontroler nieustannie nasłuchuje sygnału na pinie RX. Oczekiwane wiadomości to pojedyncze znaki, które zmieniają wartość lub stan portów mikrokontrolera, bezpośrednio połączonych ze sterownikiem silników [Botc]. Dwa z tych portów kontrolują wypełnienie sygnału PWM w zakresie od 0 do 255, co umożliwia regulację prędkości silników. Cztery pozostałe porty są wykorzystywane do ustawienia stanu wysokiego lub niskiego na odpowiednich pinach sterownika silnika. W poniższej tabeli 4.1 przedstawiono wszystkie możliwe kombinacje ustawień portów. Za każdym razem, gdy mikrokontroler ustawia jeden z pinów sterującymi silnikami na stan wysoki, inicjuje równocześnie odliczanie zadanego czasu. Po jego upływie



Rysunek 4.1 Diagram połączeń elektronicznych platformy mobilnej

IN1/IN3	IN2/IN4	Działanie Silnika
HIGH	LOW	Obraca się zgodnie z wskazówkami zegara
LOW	HIGH	Obraca się przeciwnie do wskazówek zegara
LOW	LOW	Szybkie hamowanie
HIGH	HIGH	Swobodne dryfowanie

Tabela 4.1 Zachowanie silników w zależności od stanów wejściowych

mikrokontroler automatycznie zmienia stan tego pinu z powrotem na niski. Mikroprocesor stale wysyła sygnał o wartości 240 na piny ENA oraz ENB, które są odpowiedzialne za ustawienie stanu na wejściu PWM. Przy 8-bitowym systemie, to sprawia, że wypełnienie sygnału PWM wynosi około 94%.

4.3 Układ zasilania

Zasilanie platformy zostało zrealizowane przy użyciu dwóch źródeł. Pierwsze z nich to cztery baterie alkaliczne typu AA/AAA, z których każda dostarcza 1,5 V. Połączone szeregowo zapewnia napięcie 6 V. To napięcie dostarczane jest bezpośrednio do mikrokontrolera, który następnie przekazuje część tego napięcia, na poziomie 5V, do modułu Bluetooth. Drugie źródło zasilania stanowi pojedyncza ba-

teria alkaliczna 9V, która dostarcza napięcie do sterownika silników. Sterownik ten, zależnie od otrzymanego sygnału PWM, przekazuje prąd bezpośrednio do silników.

Rozdział 5

Aplikacja mobilna

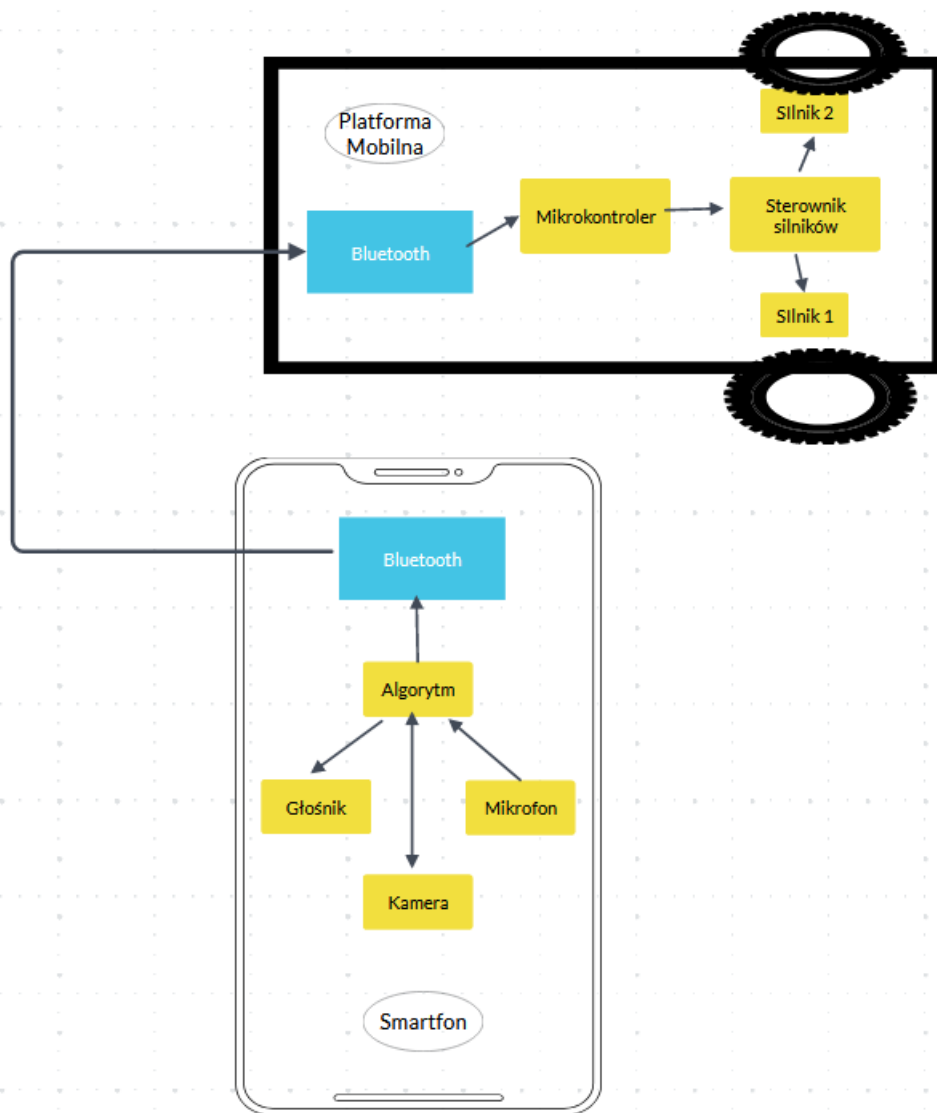
Aplikacja mobilna została zaprojektowana tak, aby wykorzystać funkcjonalność smartfona i zapewnić jego integrację z platformą mobilną, zgodnie ze schematem przedstawionym na rysunku 5.1. Oprogramowanie składa się z dwóch głównych modułów. Pierwszy moduł obsługuje detekcję i analizę sygnałów z otoczenia, do czego wykorzystuje kamerę do rozpoznawania określonych elementów oraz mikrofon do odbioru poleceń głosowych użytkownika. Drugi moduł realizuje odpowiedzi robota na te zewnętrzne bodźce, które obejmują ruch oraz generowanie odpowiedzi głosowych przez głośnik. Aplikacja mobilna została zbudowana przy użyciu narzędzi dostarczonych przez wybrane środowisko programistyczne oraz z wykorzystaniem dostępnych w nim bibliotek. Poniżej zostaną opisane narzędzia oraz biblioteki użyte przy tworzeniu aplikacji mobilnej i jego struktura.

5.1 Narzędzia

5.1.1 Środowisko programistyczne

Tworzenie aplikacji mobilnej to skomplikowany proces, wymagający dużej wiedzy o podzespołach programowanego urządzenia. Bez odpowiednich praw dostępu do każdej oddzielnej funkcjonalności w telefonie takich jak kamera, mikrofon czy dostęp do internetu, niemożliwe jest stworzenie aplikacji. Obecnie już mało kto sobie wyobraża programowanie aplikacji bez użycia dedykowanych zintegrowanych środowisk programistycznych. Jednym z nich jest Android Studio [[Wika](#)]. Oferuje ono szereg funkcji i narzędzi ułatwiających programowanie aplikacji mobilnych, takich jak: emulatory do testowania, edytor kodu, edytor widoku, dostęp do zestawu bibliotek.

Emulatory w Android Studio pozwalają na testowanie aplikacji bez potrzeby posiadania fizycznego smartfona. Pozwala to zdecydowanie szybciej, bez konieczności wgrywania programu na telefon, sprawdzić działanie poszczególnych funkcjonalności aplikacji. Można również przeprowadzać testy oprogramowania na różnych modelach smartfonów i wersjach systemu Android. Dzięki temu przetestowanie oprogramowania w różnorodnych środowiskach jest możliwe w każdej chwili. Pomijane jest również zagrożenie uszkodzenia urządzenia w wyniku testowania wadliwego oprogramowania. Do przetestowania wszystkich funkcjonalności sam emulator może być niewystarczający i wtedy będziemy potrzebowali fizycznego urządzenia.



Rysunek 5.1 Schemat komunikacji smartfona i platformy mobilnej

Samo IDE daje nam wiele możliwości do przetestowania aplikacji mobilnych, ale najprostszym sposobem wydaje się połączenie przez Wi-Fi czy podłączenie smartfona kablem do komputera. W tym wypadku również jest to proste i wygodne dla użytkownika.

Edytor widoków to narzędzie, które pozwala na wizualne projektowanie interfejsów użytkownika. Znacząco usprawnia proces tworzenia kodu XML. Przedstawia rozmieszczenie wszystkich elementów, przez co programista w każdej chwili może je przemieścić, usunąć lub dodać nowe, używając interfejsu graficznego. Stosuje się go również do skalowania i precyzyjnego umieszczania elementów interfejsu użytkownika.

Dla ułatwienia pisania kodu programiście, do Android Studio dodano edytor kodu. Narzędzie to posiada wiele użytecznych funkcjonalności umożliwiając szybsze pisanie kodu przy jednoczesnym minimalizowaniu prostych błędów. Inteligentne uzupełnianie kodu to mechanizm, który podsuwa przykładowe uzupełnienia pisanego przez użytkownika kodu. Sugeruje tym samym programiście listę dostępnych obecnie funkcji, które może on użyć. Jeśli na liście nie znajduje się pewna funkcja, edytor automatycznie sugeruje import odpowiednich bibliotek, które umożliwią jej implementację. Korzysta również z wbudowanego debugera, który znajduje błąd od razu po jego zrobieniu, umożliwiając szybką reakcję na źle sformułowany kod.

5.1.2 Biblioteki i technologie

Dla ułatwienia pisania kodu użyte zostały gotowe rozwiązania zawarte w bibliotekach. Przede wszystkim pozwalają one zrealizować procesy związane z podzespołami telefonu takie jak użycie kamery, mikrofonu, głośnika czy modułu bluetooth. Oprócz tego dostarczają gotowe rozwiązania w zakresie przetwarzania obrazów oraz rozpoznawania mowy. Te najbardziej podstawowe biblioteki jak Widget [Deva] czy Manifest [Devb] używane są w większości aplikacji. Bardziej zaawansowane biblioteki takie jak openCV [Wikd], bluetooth, Speech czy hardware.camera2, choć stosowane rzadziej niż standardowe biblioteki, znajdują swoje zastosowanie i są efektywnie wykorzystywane w aplikacjach wymagających specjalizowanych funkcji i złożonych zadań, takich jak przetwarzanie obrazu, komunikacja bezprzewodowa, rozpoznawanie mowy czy zaawansowane zarządzanie kamerą.

Podstawową biblioteką, której nie da się pominąć przy tworzeniu aplikacji na Androida, jest zdecydowanie biblioteka Widget. Pozwala ona zrealizować najprostsze funkcjonalności w zaawansowanych i prostych programach. Wykonuje proste funkcjonalności związane z obsługą reakcji na kliknięcie przycisków lub do wyświetlenia obrazków i tekstu na samym ekranie. Również świetnie się sprawdza przy zadawaniu tzw. „toastów”, czyli wyświetlaniu prostych i szybkich komunikatów, nieblokujących interakcji z użytkownikiem.

Kolejną biblioteką, bez której nie da się stworzyć aplikacji, jest Manifest. Dokonuje przetwarzania pliku AndroidManifest.xml, który zawiera szczegółowy opis uprawnień niezbędnych do prawidłowego uruchomienia aplikacji. Bez tej biblioteki nie zapiszemy danych w pamięci urządzenia czy nie skorzystamy z mikrofonu oraz aparatu. Kolejnym zadaniem jest zapobieganie nieautoryzowanemu użyciu podzespołów przez aplikację.

Biblioteka Bluetooth, jak sama nazwa wskazuje, realizuje komunikację między

urządzeniami w technologii Bluetooth. Jednymi z licznych możliwości są na przykład: przeszukiwanie listy sparowanych urządzeń lub uruchamianie odkrywania nowych urządzeń. Po uzyskaniu takich informacji są w stanie zainicjować połączenie oraz otworzyć stałą komunikację z drugim urządzeniem.

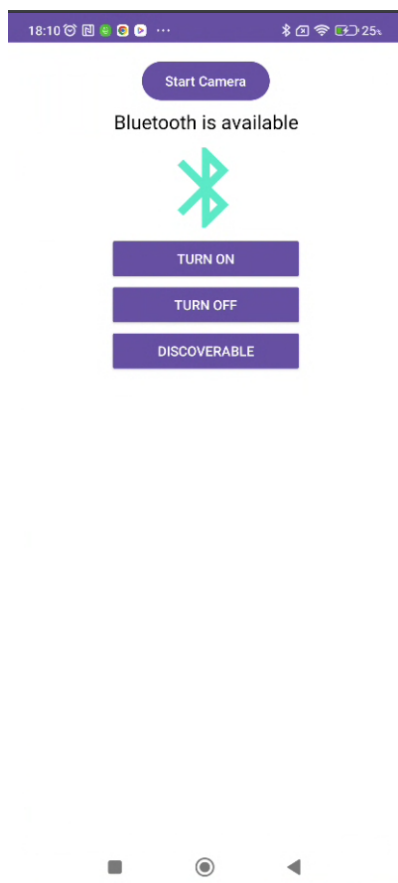
Korzystanie z kamery w smartfonie umożliwia biblioteka `hardware.camera2`. Oferują kontrolę nad parametrami kamery oraz przetwarzaniem obrazu do aplikacji. Biblioteka również steruje strumieniami z różnych kamer smartfonu oraz potrafi nimi zarządzać. Kolejnymi przydatnymi funkcjonalnościami są na pewno uzyskiwanie informacji o charakterystyce poszczególnych kamer oraz definiowanie właściwości przechwytywania obrazu. Wykorzystać można tę zaletę do ustawienia trybów takich jak: autofokus czy balans bieli.

Biblioteka `camera2` nie jest, jednakże w stanie zrealizować bardziej zaawansowanego przetwarzania obrazów, takiego jak na przykład wyszukiwanie obiektów na obrazie, czym zajmuje się `OpenCV`. Ta biblioteka jest szeroko stosowana w robotyce i medycynie do analizy wizji komputerowej. Potrafi przetwarzać obrazy, konwertując przestrzeń kolorów, filtrując je, czy realizując detekcję krawędzi. Wymienione rozwiązania zostały stworzone, aby ułatwić detekcję obiektów, twarzy czy innych części ciała. Takie technologie umożliwiają rysowanie znalezionych konturów na obrazie, co może zobrazować użytkownikowi analizę obrazu przez aplikację. `OpenCV` charakteryzuje się również zaawansowanymi algorytmami do kalibracji kamery oraz rekonstrukcji scen 3D. Rozwój biblioteki oraz jej popularność wynikają głównie z otwartej licencji, dostępnej dla wszystkich.

Dokładna analiza obrazu nie byłaby możliwa bez stałego rozwoju biblioteki `OpenCV`. To samo dotyczy rozpoznawania mowy. Biblioteka `Speech` jest w stanie interpretować i przetwarzać mowę człowieka. Umożliwia nasłuchiwanie użytkownika i rozpoznawanie tego, co mówi. Dodatkowo posiada mechanizmy do konwersji rozpoznanej mowy na tekst, co może mieć wiele zastosowań, takich jak ułatwienie pisania wiadomości czy komentarzy. Dzięki wsparciu Google, obsługiwanych jest wiele języków, jednak wymaga to połączenia z internetem. Biblioteka `Speech` również umożliwia aplikacjom konwertować zapisane słowa na syntetyzowane wypowiedzi głosowe. Dzięki temu aplikację mogą komunikować się z użytkownikiem, do przekazywania informacji lub instrukcji.

5.2 Implementacja

Główną funkcjonalnością aplikacji jest interakcja z otoczeniem, zgodnie z opisem w podrozdziale 2.4. W celu zapewnienia funkcjonalności opracowano aplikację, która będzie łączyć się z platformą mobilną za pomocą technologii Bluetooth oraz interaktywnie komunikować się z użytkownikiem. Początkowy widok aplikacji, głównie realizujący połączenie Bluetooth, został przedstawiony na rysunku 5.2. Zasadniczy widok użytkownika jest zaprojektowany do interakcji z otoczeniem. Posiada główną funkcjonalność programu, jaka jest ciągła analiza obrazu w celu wyszukania zadanych elementów i dojeżdżania do nich. To działanie jest wykorzystywane w dwóch różnych scenariuszach: szukaniu i podążaniu za wyznaczonym obiektem. Aktywacja tych scenariuszy odbywa się poprzez naciśnięcie przycisków na panelu dotykowym. Dodatkowo aplikacja nasłuchuje poleceń głosowych użytkownika



Rysunek 5.2 Początkowy widok aplikacji – ekran ustawień bluetooth

i generuje odpowiednie reakcje, a także przesyła sygnały do platformy mobilnej. Umożliwia on również inicjowanie wcześniej wspomnianych scenariuszy. Przykład zasadniczego widoku aplikacji przedstawiono na rysunku 5.3. Aby aplikacja mogła efektywnie funkcjonować, konieczne jest nadanie jej odpowiednich uprawnień umożliwiających korzystanie z wymaganych funkcjonalności, takich jak analiza obrazu, łączność Bluetooth oraz rozpoznawanie mowy. Poniżej przedstawiona zostanie realizacja całej aplikacji mobilnej, począwszy od omówienia nadanych uprawnień i obsługi połączenia Bluetooth, aż po opis funkcjonowania algorytmów wyszukiwania piłki i nasłuchiwanie poleceń użytkowników.

5.2.1 Uprawnienia

Wszystkie uprawnienia potrzebne do używania aplikacji zostały zawarte w pliku `AndroidManifest.xml`. Do zarządzania tymi uprawnieniami wykorzystywana jest biblioteka `Manifest` [Devb], dzięki której aplikacja sprawdza, czy uprawnienia zostały wcześniej zaakceptowane przez użytkownika. Jeśli nie, to dzięki funkcji `ActivityCompat` z biblioteki systemowej `Core`, aplikacja prosi o dostęp do poszczególnych podzespołów. W początkowym widoku pyta ona jedynie o dostęp do modułu Bluetooth. Natomiast w zasadniczym widoku następuje prośba o dostęp do mikrofonu, internetu oraz kamery.

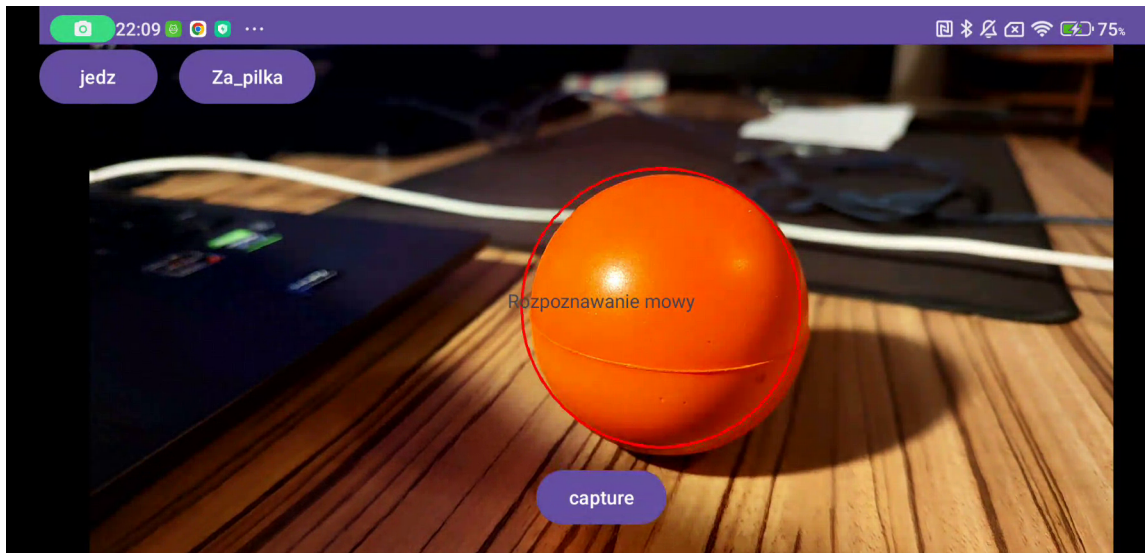
5.2.2 Realizacja połączenia bluetooth

Po wejściu do aplikacji użytkownikowi prezentowany jest interfejs, który został przedstawiony na rysunku 5.2. Ten widok pozwala na konfigurację interfejsu Bluetooth na smartfonie. Skonstruowany jest z czterech przycisków, pola tekstowego i jednego obrazka. Przycisk oznaczony jako „Start Camera” umożliwia przejście do zasadniczego widoku aplikacji. Dwa kolejne przyciski, „Turn On” i „Turn Off”, zaimplementowane przy użyciu biblioteki Bluetooth, konkretnie jej komponentu „BluetoothAdapter”, służą do włączania i wyłączenia modułu Bluetooth na smartfonie. Po ich użyciu pole tekstowe informuje nas czy Bluetooth jest obecnie uruchomiony lub wyłączony. Ostatni przycisk, korzystając z tej samej biblioteki, pozwala na wysłanie żądania aktywacji trybu odkrywania Bluetooth w systemie. Obrazek w tym widoku ma funkcję wskaźnika stanu modułu Bluetooth: wyświetla logo Bluetooth, gdy moduł jest aktywny, natomiast logo przekreślone sygnalizuje jego wyłączenie, informując użytkownika o gotowości systemu do przejścia do kolejnego ekranu.

Z gotowym modułem bluetooth, po przejściu do zasadniczego widoku, oprogramowanie od razu realizuje stałe połączenie Bluetooth, zgodnie ze schematem blokowym na rysunku 5.1. W tym celu wykorzystana jest biblioteka Bluetooth, która używa wcześniej zainicjowanego gniazda z podaną nazwą modułu Bluetooth i identyfikatorem UUID.

5.2.3 Analiza obrazu

Zadaniem zasadniczego widoku jest wyświetlanie obrazu z kamery przedniej i analiza obrazu w celu poszukiwaniu zadanego obiektu. Aplikacja identyfikuje przedmiot na podstawie zadanego koloru i określa jego położenie na obrazie w postaci dwóch zmiennych: środka i promienia okręgu. Realizacja tej operacji jest możliwa dzięki wykorzystaniu biblioteki OpenCV. Przechwytyjąc obraz i używając funkcji visibility oraz setCvCameraViewListener, aplikacja przetwarza obraz, który wyświetlany jest na całym ekranie, jak na rysunku 5.3. Za pomocą funkcji onCameraFrame, oprogramowanie analizuje obraz, dostarczając więcej informacji o obrazie z kamery. Aplikacja ma za zadanie rozpoznać, jak daleko i w którym miejscu znajduje się zadany obiekt. Do realizacji tego celu obraz jest przetwarzany z formatu RGB do HSV za pomocą funkcji cvtColor. Dla potrzeb identyfikacji obiektu zastosowano definicję jego koloru w obrębie wyznaczonego zakresu w formacie HSV. Aplikacja nakłada na obraz maskę, używając funkcji inRange, która rozróżnia piksele danego koloru od reszty obrazu. Funkcja findContours jest wykorzystywana do znalezienia wszystkich konturów na obrazie, które są następnie umieszczane na liście. Oprogramowanie identyfikuje największy kontur, zakładając, że jest to zadany obiekt, i rysuje wokół niego koło obejmujące cały kontur. Elementy na obrazie są zaznaczane za pomocą funkcji circle, a obliczenie najmniejszego okręgu, obejmującego cały kontur, realizowane jest przez komendę minEnclosingCircle. Przed wykonaniem tej operacji, dwie zmienne: środek i promień okręgu, przyjmują wartość null. Po operacji ich wartości zmieniają się zgodnie z położeniem i wielkością obiektu na obrazie.



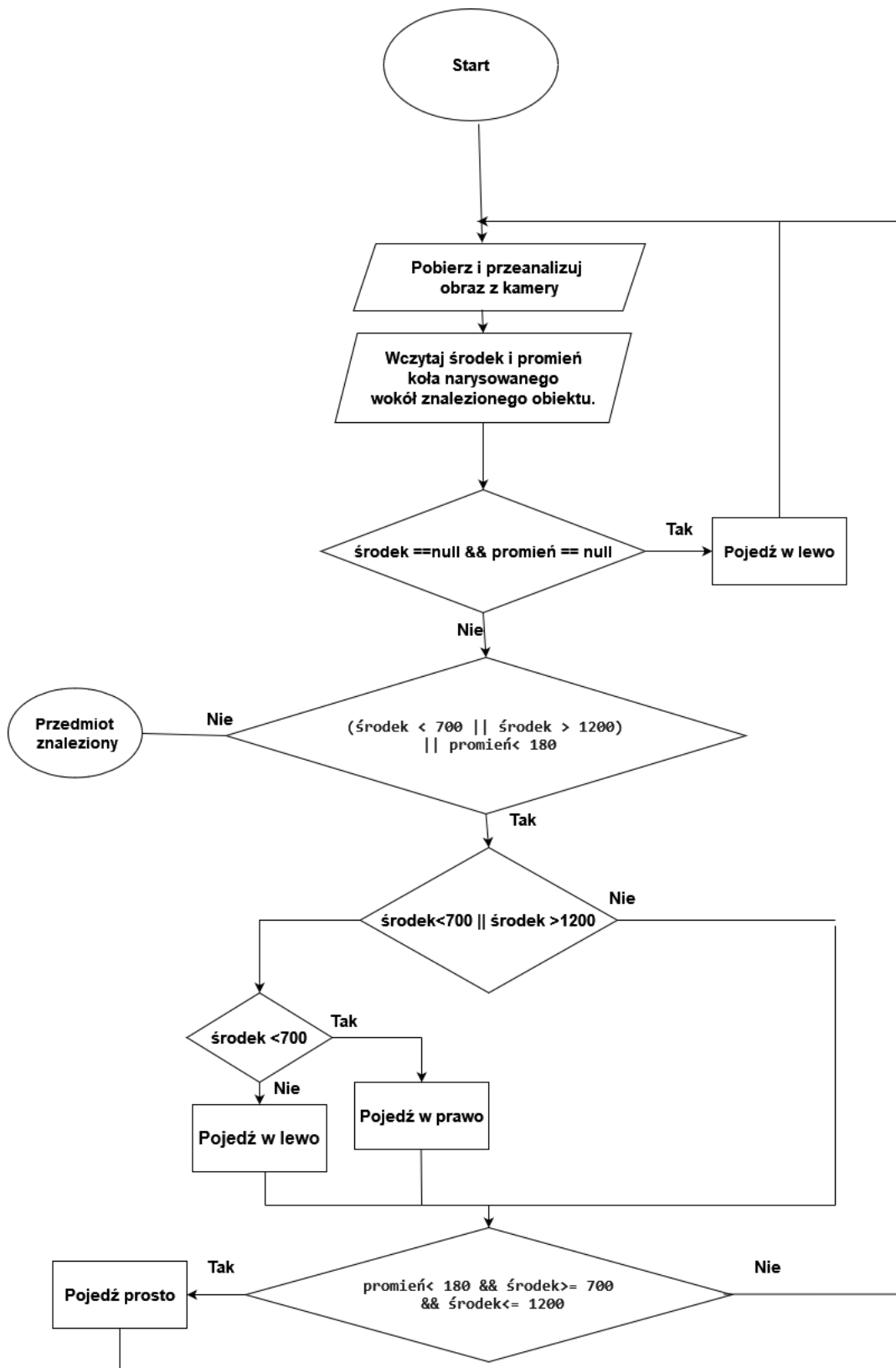
Rysunek 5.3 Przykładowa analiza obrazu

5.2.4 Szukanie zadanego obiektu

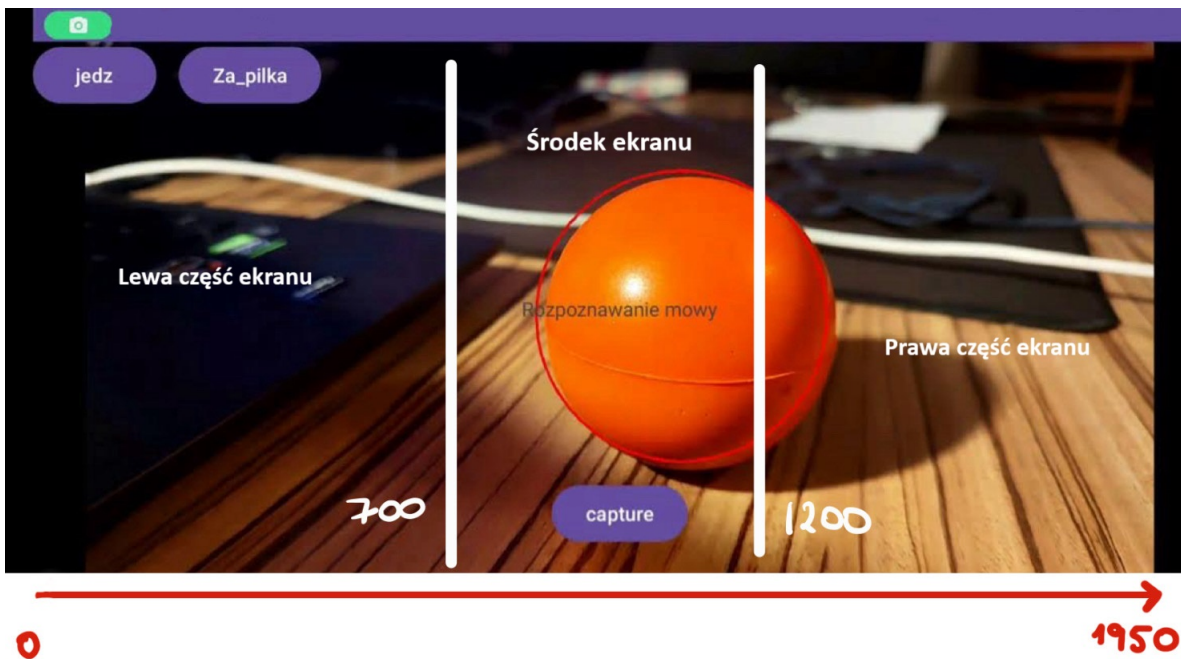
Proces wyszukiwania określonego obiektu przez robota obejmuje analizę obrazu i przesyłanie odpowiednich sygnałów, które umożliwiają robotowi przemieszczanie się. Algorytm analizuje położenie obiektu i na tej podstawie decyduje o kierunku ruchu. Zastosowany jest do dwóch trybów pracy, jeden związany z podążaniem za zadanym obiektem, a drugi z szukaniem paru obiektów różniących się kolorem. Tryb podążania za danym przedmiotem aktywowany jest poprzez naciśnięcie przycisku „jedz”, natomiast tryb wyszukiwania obiektów uruchamiany jest za pomocą przycisku „Za_pilka”. Oba te zadania używają funkcji dojeżdżania platformy do obiektu.

Algorytmu dojeżdżania do obiektu został przedstawiony na rysunku 5.4. Jak widać, algorytm oczekuje na podanie wartości promienia i punktu centralnego okręgu od operacji analizy obrazu, który zostanie narysowany wokół zidentyfikowanego przedmiotu. Do momentu znalezienia obiektu przez robota, porusza się on w lewo. Po uzyskaniu danych dotyczących promienia i środka okręgu algorytm analizuje dostarczone wartości. Jeżeli środek okręgu nie mieści się w centralnej części obrazu, określonej w zakresie od 700 do 1200 pikseli, tak jak przedstawiono na rysunku 5.5, pojazd skręca w lewo lub w prawo, aby platforma znalazła się naprzeciwko zadanego obiektu. Gdy środek przedmiotu znajduje się w podanym zakresie, pojazd porusza się prosto. Promień określa odległość przedmiotu od smartfona, przy czym większa wartość oznacza, że obiekt znajduje się bliżej. W momencie, gdy promień narysowanego okręgu przekracza 180 pikseli i środek znajduje się w określonym zakresie oznacza to, że robot zbliżył się na zadaną odległość i powinien zakończyć poszukiwanie przedmiotu. Znalezienie obiektu sygnalizowane jest użytkownikowi poprzez wypowiedzenie komunikatu za pomocą biblioteki Speech.

Proces podążania za obiektem polega na stałym zmniejszaniu odległości między robotem a obiektem, który co jakiś czas się od niego oddala. W momencie stracenia obiektu z pola widzenia, robot stale będzie poruszał się w lewo, aż do ponownego znalezienia przedmiotu. Proces podążania można zakończyć tylko poprzez wypowiedzenie odpowiedniej komendy. Robot nie kończy od razu działania po jej odebraniu,



Rysunek 5.4 Diagram algorytmu poruszania się do obiektu



Rysunek 5.5 Wymiary ekranu oraz wartości graniczne centrum

a dopiero po dojechaniu do zadanego obiektu.

W procesie wyszukiwania trzech obiektów o różnych kolorach robot dojeżdża do każdego obiektu po kolei. Przed rozpoczęciem szukania przedmiotu wypowiadany jest komunikat, jaki przedmiot jest szukany oraz o jakim kolorze. Po dojechaniu do pierwszego obiektu wypowiadany jest komunikat informujący o tym i robot przemieszcza się do losowo wybranej pozycji. Robot powtarza to działanie, aż do znalezienia wszystkich trzech obiektów, po czym kończy swoje działanie.

5.2.5 Rozpoznawanie mowy

Do wywołania zaimplementowanych funkcji używane są komendy wypowiadane przez użytkownika. System rozpoznaje polecenia użytkownika i odpowiada, generując syntetyczną mowę za pomocą biblioteki Speech. Cała implementacja jest uruchamiana po kliknięciu przycisku „Capture”, który aktywuje funkcję nasłuchiwanie użytkownika przy użyciu komendy `createSpeechRecognizer`. Oprogramowanie dodatkowo korzysta z funkcji `putExtra` z wartością „pl-PL”, aby interpretować polecenia w języku polskim. Gdy aplikacja zrozumie wypowiedź użytkownika, wywoływana jest funkcja `onResults`. Na początku oprogramowanie wyświetla na ekranie usłyszaną wypowiedź, używając polecenia `getStringArrayList`. Następnie aplikacja sprawdza za pomocą funkcji `contains`, czy wypowiedziana fraza zawiera jedną z komend:

- „w lewo” – robot przemieszcza się w lewo na zadaną odległość,
- „w prawo” – robot przemieszcza się w prawo na zadaną odległość,
- „do przodu” – robot przemieszcza się do przodu na zadaną odległość,
- „losowa” – robot przemieszcza się w losowym kierunku na losową odległość,

- „szukaj” – robot zaczyna szukać trzech zadanych obiektów,
- „za piłką” – robot podąża za zadany obiekt,
- „przestań” – robot przestaje podążać za zadany obiekt,
- „co tam” – robot „mówi” co u niego i pyta użytkownika co u niego,
- „koniec” – smartfon kończy nasłuchiwanie,

Realizacja komend następuje poprzez wysłanie do platformy odpowiednich sygnałów Bluetooth. Wyjątkiem jest komenda „przestań”, „koniec” i „co tam”, które wykonują się bezpośrednio na smartfonie. Do każdego zrealizowanego polecenia przy użyciu biblioteki Speech dołączana jest odpowiedź, która polega na generowaniu syntetycznej mowy odtwarzanej przez głośnik. Po usłyszeniu komendy od użytkownika i wykonaniu związanego z nim polecenia, smartfon kontynuuje nasłuchiwanie. Informuje użytkownika, jeśli nie otrzyma żadnego polecenia lub go nie zrozumie. Po odebraniu słów użytkownika „za piłką” realizowany jest proces podążania, a polecenie „przestań” kończy jego działanie. Komendy „w lewo”, „w prawo”, „do przodu” przemieszczają robota w określonym kierunku, a „losowa” w nieokreślonym kierunku. Instrukcja „co tam” inicjuje prostą wymianę uprzejmości, w której robot odpowiada, że wszystko u niego w porządku, a następnie pyta użytkownika o jego samopoczucie. Komenda „szukaj” rozpoczyna proces szukania trzech obiektów o zadanych kolorach. Z kolei komenda „koniec” uruchamia odtwarzanie pliku z pożegnaniem i kończy nasłuchiwanie poleceń użytkownika.

Rozdział 6

Testy

Zaimplementowane algorytmy umożliwiły interakcję robota ze środowiskiem. Przeprowadzono testy każdej części robota, aby sprawdzić i zapewnić jego prawidłowe działanie. Początkowo skoncentrowano się na działaniu platformy mobilnej. Testy te obejmowały sprawdzanie możliwości poruszania się platformy we wszystkich kierunkach. Następnie przetestowano aplikację na smartfonie, oceniając jej poprawność działania. W końcowej fazie połączono smartfon z platformą mobilną, by zweryfikować reakcje robota na otoczenie.

Do testów wybrano pomarańczową, zieloną i niebieską piłkę oraz różne komendy, które użytkownik mógł wydać robotowi. Robot został zaprogramowany do szukania piłek i reagowania na ich znalezienie. W zależności od wydanego polecenia robot miał za zadanie śledzić piłkę do momentu otrzymania komendy „przestań” albo szukać trzech piłek różnego koloru. Po znalezieniu piłki zawsze oznajmiał ten fakt wypowiadając komunikat i przemieszczał się na losowo wybraną pozycję, aby kontynuować swoje poszukiwania. Robot miał również wykonywać inne polecenia niezwiązane z poruszaniem się, na które odpowiadał werbalnie.

6.1 Testy platformy mobilnej

Testowanie działania platformy rozpoczęto od sprawdzenia kilku kluczowych aspektów: czy odpowiednie napięcie jest dostarczane do każdego elementu, jak działają silniki oraz czy przemieszczanie pojazdu odbywa się prawidłowo. Początkowo skoncentrowano się na działaniu samych silników zasilanych napięciem 6V. Został wgrany program mający poruszać platformę kolejno w każdym kierunku. Testy potwierdziły prawidłowe działanie obu silników. Jednakże, po podłączeniu modułu Bluetooth, okazało się, że napięcie jest zbyt niskie dla wszystkich komponentów platformy. Problem został rozwiązany przez dodanie dodatkowego zasilania 9V, które było dostarczane bezpośrednio do sterownika silników i samych silników, eliminując tym samym problemy z zasilaniem.

W trakcie testów przemieszczania platformy początkowo wszystko wydawało się działać poprawnie. Jednak pierwszy problem pojawił się, gdy platforma miała jechać prosto: z powodu nierównomiernego rozmieszczenia masy, pojazd nieznacznie zboczył w prawo. Kolejny problem, wynikający również z rozkładu masy, dotyczył koła napędowego, które traciło przyczepność. Oba te problemy zostały skutecznie roz-

wiązane poprzez dodanie obciążenia bezpośrednio przy kołach.

6.2 Testy aplikacji

Testy aplikacji koncentrowały się na dwóch kluczowych funkcjach: analizie obrazu oraz rozpoznawaniu mowy. Analiza obrazu miała na celu wyodrębnienie poszukiwanego elementu na podstawie największego konturu w określonym przedziale kolorów. Podczas testów najwięcej problemów sprawiało znalezienie niebieskiej piłki – zakres kolorów był zbyt szeroki, co powodowało, że aplikacja błędnie identyfikowała obiekty z otoczenia jako szukaną piłkę. Problem rozwiązano przez zastosowanie bardziej precyzyjnego zakresu kolorów dla niebieskiej piłki.

Kolejnym wyzwaniem były warunki oświetleniowe. Przy sztucznym świetle, w dolnej części piłki padał cień, co powodowało, że kamera widziała tylko jej część. Skutkowało to błędnym odbiorem rozmiaru piłki przez aplikację, przez co robot oceniał, że piłka jest dalej niż w rzeczywistości. Problem ten rozwiązano, ustawiając w aplikacji mniejszy promień piłki, który miał być wykrywany.

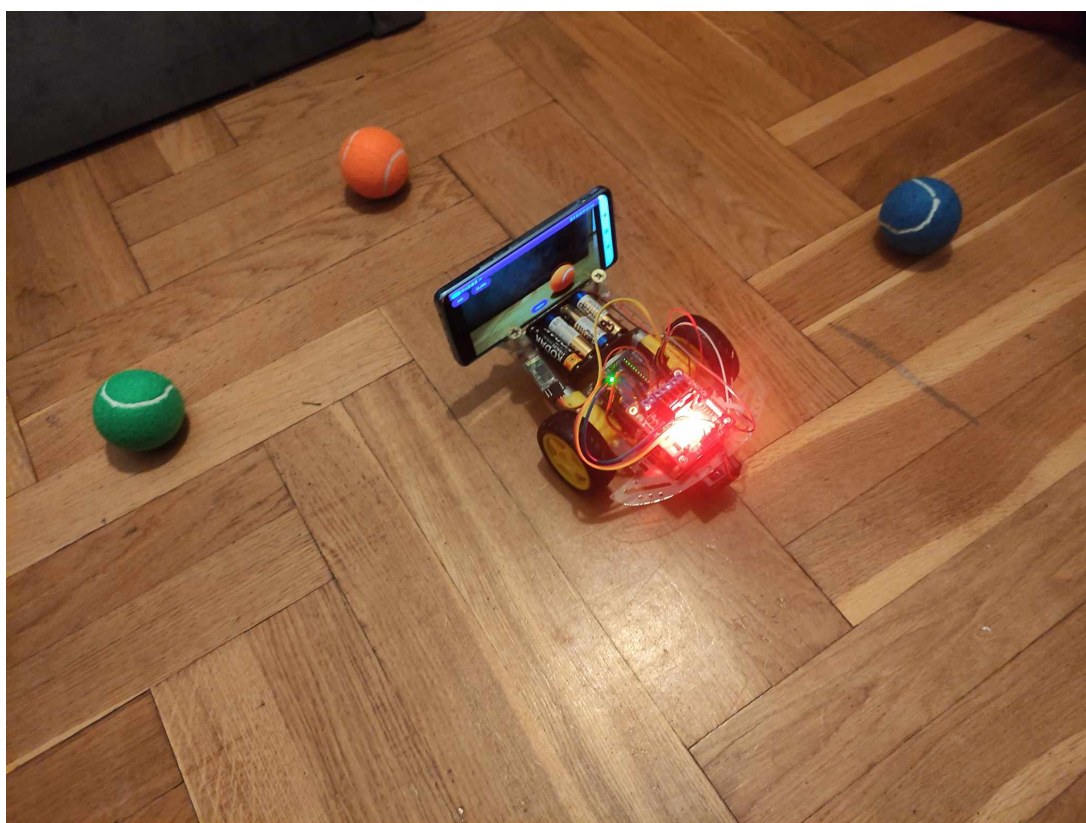
Przy testach rozpoznawania mowy robot często błędnie interpretował końcówki słów, zwłaszcza gdy użytkownik wypowiadał pełne zdania. Aby poprawić skuteczność rozpoznawania mowy, zmodyfikowano sposób komunikacji z robotem – zamiast całych zdań, zaczęto używać pojedynczych fraz. Dzięki temu robot zaczął lepiej rozumieć wydawane mu polecenia.

6.3 Interakcja z otoczeniem

Na rysunku 6.1 widać robota otoczonego trzema piłkami. Po otrzymaniu polecenia „szukaj”, robot rozpoczyna poszukiwania. Najpierw informuje użytkownika o poszukiwaniu pomarańczowej piłki. Po jej znalezieniu robot wykorzystuje algorytm do poinformowania użytkownika o sukcesie, następnie przemieszcza się w losowe miejsce i rozpoczyna poszukiwania niebieskiej piłki. Po jej znalezieniu proces jest powtarzany dla zielonej piłki, po czym robot kończy swoją aktywność.

Algorytm śledzenia piłki działa podobnie do algorytmu poszukiwania. Główna różnica polega na tym, że robot ciągle namierza pomarańczową piłkę i stara się do niej dojechać. Po dojechaniu robot czeka, aż piłka oddali się od niego, by móc znów do niej podjechać. Proces ten jest kontynuowany do momentu otrzymania odpowiedniego polecenia od użytkownika, po czym robot zatrzymuje śledzenie piłki.

Robot jest także zaprogramowany do zadawania prostych pytań, takich jak „Co u Ciebie?”, co ma na celu uczynienie interakcji z użytkownikiem bardziej przyjemną. Na zakończenie interakcji, robot żegna się z użytkownikiem.



Rysunek 6.1 Otoczenie robota podczas testów

Rozdział 7

Zakończenie

Praca miała na celu stworzenie platformy mobilnej, która da możliwość przemieszczania się smartfonowi. Zaprojektowano i zrealizowano konstrukcję platformy mobilnej. Stworzono także aplikację mobilną, która odbiera sygnały z otoczenia i potrafi na nie reagować. Zrealizowano komunikację między mikrokontrolerem a smartfonem.

Można stwierdzić, że cel pracy został osiągnięty. Smartfon wykorzystuje platformę mobilną do poruszania się oraz swoje wewnętrzne podzespoły do interakcji z otoczeniem. Zostały zaimplementowane metody pozwalające na znalezienie i przemieszczanie robota do określonych obiektów z wykorzystaniem obrazu z kamery. Opracowano również prostą interakcję z użytkownikiem. Polega na wydawaniu prostych komend robotowi, który reaguje na nie, przemieszczając się lub wypowiadając komunikaty. Niniejsza praca może przyczynić się do rozwoju różnego typu funkcjonalności do smartfonów takich jak poruszanie się czy chwytanie.

Możliwy jest dalszy rozwój projektu, na przykład poprzez wykorzystanie dodatkowych podzespołów smartfona, takich jak wyświetlacz czy moduł lokalizacji GPS. Ponadto istnieje potencjał do ciągłego doskonalenia aktualnie działających funkcji, takich jak rozpoznawanie obrazu lub mowy. Przykładowo, można by dodawać nowe komendy użytkownika lub prowadzić bardziej precyzyjną analizę obrazu, umożliwiając robotowi skanowanie pomieszczenia i zapamiętywanie swojej wcześniejszej pozycji.

Bibliografia

- [Ank] Anki Cozmo Robot. *Cozmo vs Vector – 5 Differences Between Them, and Which One is Better*. <https://ankicozmorobot.com/cozmo-vs-vector/>.
- [Bota] Botland. *Moduł HC-05*. <https://botland.com.pl/moduly-bluetooth/2570-modul-bluetooth-hc-05-5903351241311.html>.
- [Botb] Botland. *Silnik*. <https://botland.com.pl/silniki-dc-katowe-z-przekladnia/16016-silnik-dc-148-3-6v-z-podwojnym-walem-200rpm-5904422344078.html>.
- [Botc] Botland. *Sterownik silnika*. <https://botland.com.pl/sterowniki-silnikow-moduly/8227-dwukanalowy-sterownik-silnikow-l298n-modul-wb291111-iduino-st1112-5903351241236.html>.
- [Deva] Developers. *android.widget*. <https://developer.android.com/reference/android/widget/package-summary>.
- [Devb] Google for Developers. *App manifest overview*. <https://developer.android.com/guide/topics/manifest/manifest-intro>.
- [Fra23] Marcin Frąckiewicz. *Zalety robotyki roju w monitorowaniu i ochronie środowiska*. <https://ts2.space/pl/zalety-robotyki-roju-w-monitorowaniu-i-ochronie-srodowiska/#gsc.tab=0>. 2023.
- [Hum] Humanoid Robotics Institute, Waseda University. *WABOT WAseda roBOT*. https://www.humanoid.waseda.ac.jp/booklet/kato_2.html.
- [Med17] Medycyna Praktyczna. *Paro foka*. <https://adst.mp.pl/img/articles/neurologia/aktualnosci/paro-foca-700x550.jpg>. 2017.
- [Neoa] Neo robot. *Makeblock*. https://neorobot.pl/environment/cache/images/0_0_productGfx_bc9aa0a7b2d11dea49ac1d9142e12607.jpg.
- [Neob] Neorobot. *Makeblock*. <https://neorobot.pl/pl/Makeblock-mBot-S-Explorer-Kit.html>.
- [Net] Nettigo. *Arduino*. <https://nettigo.pl/products/klon-arduino-uno-r3-atmega328p-mega16u2>.
- [PAR] PARO Robots U.S. *Paro*. <http://www.parorobots.com/>.
- [Pią18] Zbigniew Piątek. *Roboty współpracujące Rethink Robotics*. <https://przemysl-40.pl/index.php/2018/06/03/rethink-robotics-rosnie-liczba-dostepnych-w-polsce-marek-robotow-wspolpracujacych/>. 2018.
- [Pul] Mariusz Pultyn. *Dlaczego robot nie równa się człowiek?* <https://dtmates.com/automatyzacja/dlaczego-robot-nie-rowna-sie-czlowiek/>.

- [TM18] K. Tchoń i R. Muszyński. *Mechanika Analityczna. Notatki do wykładów z dziedziny automatyki i robotyki*. Katedra Cybernetyki i Robotyki, Politechnika Wrocławska, https://kcir.pwr.edu.pl/~mucha/Skrypty/KTRM_Mechanika_analityczna.pdf, 2018.
- [TTS] TTS. *Bee-Bot - the story behind*. <https://www.tts-group.co.uk/blog/2019/01/25/bee-bot-the-story-behind-our-award-winning-programmable-robot.html>.
- [Wika] Wikipedia. *Android Studio*. https://pl.wikipedia.org/wiki/Android_Studio.
- [Wikb] Wikipedia. *Arduino*. <https://pl.wikipedia.org/wiki/Arduino>.
- [Wikc] Wikipedia. *Interakcja*. <https://pl.wikipedia.org/wiki/Interakcja>.
- [Wikd] Wikipedia. *OpenCV*. <https://pl.wikipedia.org/wiki/OpenCV>.
- [Wike] Wikipedia. *Panel sterowniczy*. https://pl.wikipedia.org/wiki/Panel_sterowniczy.
- [Wikf] Wikipedia. *Robot*. <https://pl.wikipedia.org/wiki/Robot>.
- [zro20] zrobotyzowany.pl. *Współpraca człowieka z robotem*. <https://zrobotyzowany.pl/informacje/publikacje/3730/bezpieczna-wspolpraca-czlowieka-z-robotem>. 2020.

Spis rysunków

2.1	Robot Paro [Med17]	7
2.2	Robot Makeblock [Neoa]	8
3.1	Kompletny robot z układem sterowania	12
3.2	Korpus robota z układem napędowym	12
3.3	Sposób montowania smartfona	13
4.1	Diagram połączeń elektronicznych platformy mobilnej	16
5.1	Schmat komunikacji smartfona i platformy mobilnej	20
5.2	Początkowy widok aplikacji – ekran ustawień bluetooth	23
5.3	Przykładowa analiza obrazu	25
5.4	Diagram algorytmu poruszania się do obiektu	26
5.5	Wymiary ekranu oraz wartości graniczne centrum	27
6.1	Otoczenie robota podczas testów	31