

1. Jan Adamski (lider)	180299	ACiR, sem. 6, KSA
2. Agnieszka Klingiert (dok.)	180231	
3. Ewa Piórkowska	180392	
4. Karolina Szyndlarewicz	175892	
5. Marina Galanina	183053	
6. Jelizaveta Kurilcik	183022	

**Projekt z przedmiotu
Techniki Programowania w Systemach Wbudowanych**

Robot mobilny PG-8 inspirowany modelem BB-8

DOKUMENTACJA



Spis treści

WYKAZ NAJWAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW	4
1. Wstęp	5
1.1. Cel pracy	5
1.2. Zakres pracy	5
1.3. Przegląd pracy	6
2. Informacje wstępne o robocie	7
2.1. Idea robota	7
2.1.1. Rys historyczny	7
2.1.2. Wykonywane zadania	8
2.2. Zasady działania - balansowanie, sterowanie, algorytmy sterowania	9
2.2.1. Zasada działania pozwalająca na balansowanie robota	9
2.2.2. Zasada działania sterowania robotem	9
2.2.3. Algorytmy sterowania robotem	10
2.3. Elementy elektroniczne	10
2.3.1. Płytki z mikrokontrolerem	10
2.3.2. Silniki elektryczne oraz sterownik do silników	12
2.3.3. Akcelerometr	13
2.3.4. Moduł Bluetooth	13
2.3.5. Moduł MP3	14
3. Realizacja	16
3.1. Konstrukcja ciała i głowy robota	16
3.2. Wybór elementów elektronicznych	18
3.3. Projekt układu	21
3.4. Program sterujący	24
3.5. Aplikacja sterująca	27
4. Testy	33
4.1. Poprawki	33
4.2. Końcowe testy	35
5. Podsumowanie	37
5.1. Streszczenie pracy	37
5.2. Wnioski końcowe	37
BIBLIOGRAFIA	39
WYKAZ RYSUNKÓW	41
WYKAZ TABEL	42

WYKAZ NAJWAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

- Ø - średnica
- przetwornik A/C - przetwornik analogowo-cyfrowy
- CPU (ang. Central Processing Unit) - procesor
- DAC (ang. Digital-Analog Converter) - przetwornik analogowo-cyfrowy
- DC (ang. Direct Current) - prąd stały
- EDR (ang. Enhanced Data Rate) - opcjonalna technologia wprowadzona do standardy bluetooth zwiększająca przepływ danych
- Fat (ang. File Allocation Table) - system plików określający rozmieszczenie plików, katalogów i wolnej przestrzeni na nośnikach danych
- MEMS (ang. Microelectromechanical System) - mikroukład elektromechaniczny
- PWM (ang. Pulse Width Modulation) - metoda regulacji sygnału polegająca na zmianie wypełnienia reprezentującego go sygnału prostokątnego
- PVA (ang. polyvinyl alcohol) - poli (alkohol winylowy)
- RPM (ang. Rotations Per Minute) - obroty na minutę - jednostka prędkości obrotowej
- karta SD (ang. Secure Digital) - popularny standard kart pamięci
- UART (ang. Universal Asynchronous Receiver-Transmitter) - układ scalony służący do asynchronicznego przekazywania i odbierania informacji poprzez port szeregowy
- USB (ang. Universal Serial Bus) - popularne złącze komunikacyjne

1. Wstęp

W tym rozdziale zostanie dokładnie określony cel i przewidywany zakres pracy. Krótki wstęp przedstawia ogólną ideę projektu oraz informacje o tym, co można znaleźć w kolejnych rozdziałach. Aby ułatwić czytanie dokumentacji, w spisie treści zostały zamieszczone hiperłącza do każdego z podpunktów.

1.1. Cel pracy

Celem pracy jest zaplanowanie konstrukcji oraz zbudowanie robota mobilnego inspirowanego modelem BB-8. W tym celu wymagane jest zapoznanie się z konstrukcją mechaniczną oraz fizyką ruchu robota, aby następnie móc przyjąć odpowiednie założenia projektowe. Kolejnym krokiem jest przegląd dostępnych rozwiązań technicznych i wybranie najlepszych. Celem projektu jest nie tylko zaprojektowanie i zbudowanie fizycznego robota, ale również sporządzenie dokumentacji technicznej, jak i nauczenie się współpracy w większej grupie, rozdziału zadań i gospodarowania czasem.

1.2. Zakres pracy

Pracę nad projektem można podzielić na etapy: skonstruowanie ciała i głowy robota, wybranie elementów elektronicznych, zaprojektowanie połączeń układu, napisanie programu sterującego robotem, stworzenie aplikacji do zdalnego sterowania robotem, przetestowanie działania robota.

Konstrukcja ciała i głowy robota jest jednym z prostszych, lecz najistotniejszych działań do wykonania, ponieważ na nich będzie montowana cała elektronika. Dodatkowo wymagany jest określony kształt tych elementów, aby zapewnić płynny ruch robota. Ciało robota zostanie wykonane techniką paper mache, natomiast głowa z użyciem kulki styropianowej.

Elementy elektroniczne muszą być dobrane zarówno ze względu na ich parametry prądowo-napięciowe, jak i wymiary i wzajemną kompatybilność. Kolejną istotną rzeczą jest zaprojektowanie połączeń całego układu, który wykorzysta wybrane wcześniej elementy elektroniczne. Trzeba wziąć pod uwagę, że musi być to na tyle dobrze rozplanowane, aby w razie wszelkich usterek i/lub potrzeby wymiany komponentów było to proste do wykonania i bez zbędnej ingerencji w resztę układu.

Aby robot mógł się przemieszczać oraz korzystać z kamerki i wydawać dźwięki niezbędne jest napisanie programu wysyłającego odpowiednie sygnały sterujące. Aby umożliwić zdalne sterowanie robotem należy dodatkowo stworzyć aplikację w tym celu.

Ostatnim zakresem pracy są testy, które pozwolą określić poprawność działania robota. Wykryją one również wszelkie niedociągnięcia, które następnie będzie można poprawić.

1.3. Przegląd pracy

W kolejnych rozdziałach zostanie przedstawione: idea robota i jego rys historyczny, zasada działania umożliwiająca balansowanie tego typu robotów, zasady sterowania, opis elementów elektronicznych oraz realizacja (konstrukcja robota, wybór elementów elektronicznych, projekt układu, napisany program, aplikacja sterująca, testy).

Rozdział 2. zawiera opis stanu aktualnej wiedzy. Zostaną przedstawione ogólne informacje o modelu robota, jego idea, historia i zadania, jakie wykonuje. W tym samym rozdziale znaleźć można opis zasad pozwalającej na balansowanie robota oraz zasad umożliwiających sterowanie robotem. Pod koniec rozdziału zamieszczono przegląd elementów elektronicznych, jakie mogą być wykorzystywane przy budowie tego typu robotów.

Rozdział 3. w całości poświęcony jest działaniom, dzięki którym projekt został zrealizowany. Rozpoczyna się od opisu konstrukcji ciała i głowy robota, wykorzystanych do tego technik i rozwiązań oraz etapów tych prac. Kolejnym podpunktrem jest argumentacja wyboru użytych elementów elektronicznych oraz opis projektu układu. Ostatnią częścią tego rozdziału jest opis programu i aplikacji sterującej robotem.

Rozdział 4. to część pracy, w której zamieszczone są wszelkie informacje o testach, sposobie ich wykonywania, niedociągnięciach wykrytych w testach pośrednich oraz ich poprawkach, testach końcowych i ich wyników.

Rozdział 5. jest przeznaczony do podsumowania całej pracy oraz wniosków z niej wyciągniętych.

2. Informacje wstępne o robocie

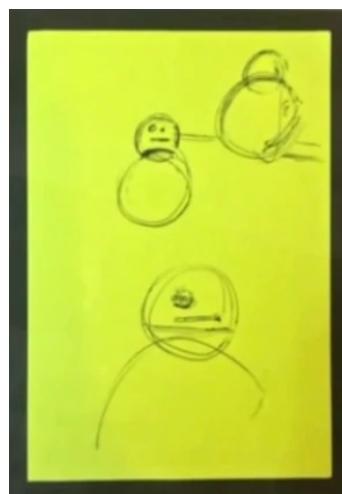
W tym rozdziale zostanie przedstawiona idea projektu robota mobilnego inspirowanego modelem BB-8, jego rys historyczny, opis wykonywanych działań. Kolejną rzeczą, którą zajęto się w tym rozdziale, to zaprezentowanie posiadanej wiedzy oraz nabyciej w trakcie pracy nad projektem. Wliczają się w to zasady sterowania oraz pozwalające na balansowanie robota, jak i przegląd elementów elektronicznych możliwych do wykorzystania w tym projekcie.

2.1. Idea robota

Podrozdział ten skupia się na ukazaniu historii robota BB-8 oraz jego przeznaczenia i wykonywanych działań.

2.1.1. Rys historyczny

BB-8 został zaprojektowany przez Neala Scanlana na potrzeby serii filmów *Gwiezdne Wojny*.



Rys.2.1. Pierwszy szkic BB-8, wykonany przez Jeffreya Abramsa [12]



Rys.2.2. Kadr z filmu Gwiezdne Wojny [13]

Po raz pierwszy został użyty w filmie *Gwiezdne Wojny: Przebudzenie Mocy* z 2015 roku. Na planie filmowym używano zdalnie sterowanego robota i tylko w najtrudniejszych scenach generowano go komputerowo. Jako bohater tej popularnej serii stał się równocześnie jednym z najbardziej rozpoznawalnych robotów na świecie - w serii filmów określanych mianem droidów.

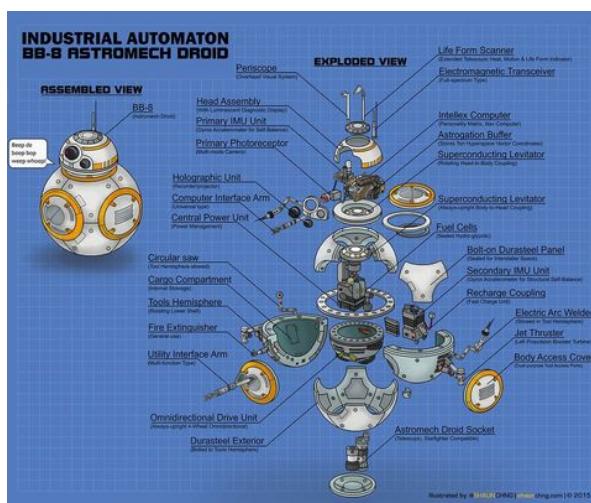
2.1.2. Wykonywane zadania

Główym zadaniem BB-8 jest poruszanie się ruchem obrotowo-postępowym. Robot ten ma kształt kuli, na której szczycie znajduje się jego głowa. Porusza się, tocząc się po ziemi, podczas gdy głowa zawsze pozostaje na górze.



Rys.2.3. Konstrukcja robota - tocząca się kula, utrzymująca głowę na górze [14]

Robot PG-8 inspirowany modelem BB-8 będzie sterowany za pomocą aplikacji. Poza poruszaniem się, robot będzie również wydawać z siebie charakterystyczne dla modelu dźwięki.



Rys.2.4. Elementy droida BB-8 [15]

2.2. Zasady działania - balansowanie, sterowanie, algorytmy sterowania

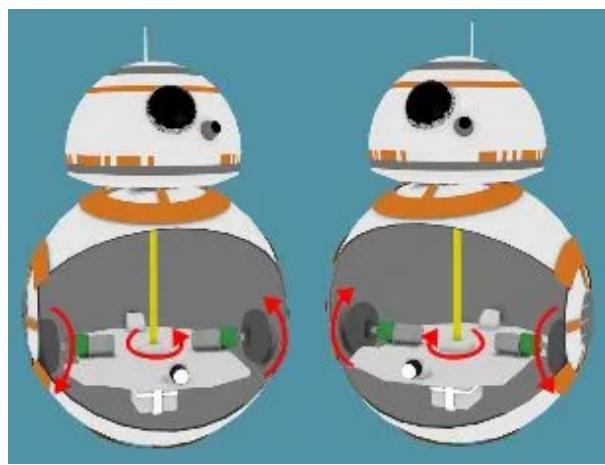
Podrozdział ten przedstawia zasadę działania pozwalającą na balansowanie robota. Podrozdział ten opisuje również zasady umożliwiające sterowanie robotem inspirowanym modelem BB8. Ostatnim elementem tego rozdziału jest opis algorytmów sterowania robotem.

2.2.1. Zasada działania pozwalająca na balansowanie robota

Balansowanie robota będzie się odbywało na zasadzie umieszczenia w kuli ruchomej platformy, co będzie skutkowało powracaniem kuli do punktu równowagi i pionowej orientacji wyjściowej. Będzie się to opierało na podobnej zasadzie co wańka-wstańka, która stabilność zawdzięcza nierównomiernemu rozkładowi masy - jej środek znajduje się blisko podłoża, na którym stoi. Pośrodku platformy będzie znajdował się stabilizator, służący do utrzymywania platformy i głowy robota w tej samej pozycji.

2.2.2. Zasada działania sterowania robotem

Wspomniana w poprzednim punkcie platforma będzie przemieszczana wewnątrz sfery za pomocą przymocowanych do niej silników z kołami oraz dodatkowych rolek. Zadanie robotowi sygnału jazdy do przodu spowoduje, że platforma podjadzie do „przedniej” jego połowy. W ten sposób środek masy zostanie przesunięty nad punkt kontaktu obudowy z ziemią w kierunku, w którym robot ma się przemieszczać. Spowoduje to pojawienie się niezerowego momentu siły na obudowie, który po pokonaniu momentu tarcia statycznego spowoduje obrót obudowy i w rezultacie liniowy ruch robota. Prędkość robota będzie kontrolowana poprzez ustalenie prędkości obrotu silników. Większa prędkość ruchu kółek spowoduje większe zdecentralizowanie punktu środka masy. Obrót platformy, wykonany, gdy robot znajduje się w stanie równowagi, pozwoli mu zkręcać.



Rys.2.5. Zasada balansowania i sterowania BB-8 [16]

2.2.3. Algorytmy sterowania robotem

Poniżej przedstawione zostaną algorytmy sterowania robotem. Omówione zostaną w krokach algorytmy ruchu oraz analizy kolorów otoczenia.

1. Algorytm ruchu:

- pobranie rozkazu ruchu z aplikacji, bądź funkcji szukającej kolorów,
- obrót (podanie napięcia na silniki, aby obracały się w przeciwnych kierunkach) lub ruch postępowy (podanie napięcia na silniki, aby obracały się w tym samym kierunku) zależnie od rozkazu ruchu,
- pobranie odczytu z akcelerometru,
- porównanie przewidywanej prędkości ruchu ze scałkowanym odczytem z akcelerometru,
- doprecyzowanie prędkości obrotów silników poprzez dobranie podanego na nie napięcia w celu uzyskania założonej prędkości (na podstawie uchybu).

2. Algorytm analizy kolorów otoczenia:

- wykonanie ruchu obrotowego (algorytm ruchu) platformy (razem z głową, zawierającą kamerkę),
- wysłanie odczytu z kamery rozpoznającej kolory na telefon z aplikacją/komputer poprzez wifi,
- przetworzenie uzyskanych przez telefon/komputer danych na odpowiednie polecenia wykonawcze i przesłanie ich do Arduino poprzez moduł Bluetooth,
- w przypadku przesłania na Arduino sygnału wynikającego z wykrycia koloru "wyzwalającego", rozpoczęcie ruchu robota do przodu (algorytm ruchu),
- w przypadku przesłania na Arduino sygnału wynikającego z wykrycia koloru należącego do katalogu "kolorów dźwiękowych" wysłanie odpowiadającego mu sygnału na moduł MP3, który dalej wysyła plik .mp3 na głośnik,
- włączenie timera dźwiękowego, aby robot nie próbował wydawać dźwięków ze zbyt dużą częstotliwością.

2.3. Elementy elektroniczne

Ten podrozdział będzie się skupiać na przeglądzie elementów elektronicznych możliwych do wykorzystania przy realizacji projektu robota inspirowanego modelem BB-8.

2.3.1. Płytki z mikrokontrolerem

Mikrokontroler można zdefiniować jako układ cyfrowy z wyspecjalizowanym mikroprocesorem i niezbędnym dla jego pracy urządzeniami, zawartymi w jednym układzie scalonym. Jest to system mikroprocesorowy, używany do pobierania, interpretowania i przesyłania informacji. Często jest nazywany "mózgiem", ponieważ przekazując wszystkie operacje arytmetyczne i/lub logiczne, umożliwia wykonanie istotnych programów.

Najczęściej wykorzystywanymi płytami są płytki z firm Arduino oraz Raspberry Pi. Przykładowe modele to Arduino Uno, Arduino Leonardo, Arduino TIAN, Raspberry Pi 3B+, Raspberry Pi 4 B, Raspberry Pi 3B.

Uwzględniając założenia projektowe oraz dostępność płyt drukowanych, dokonywany był wybór pomiędzy dwoma rozwiązaniami: Arduino Uno i Raspberry Pi 3.

Została stworzona tabela porównawcza, uwzględniająca wszystkie istotne parametry obu urządzeń. Tabela porównawcza Raspberry PI 3 z Arduino UNO przedstawiona jest poniżej.

Tabela 2.1. Porównanie płyt Raspberry PI 3 z Arduino UNO [22]

CECHA	RASPBERRY PI 3	ARDUINO UNO
Wymiary	85x56x17mm	75x53x15mm
Pamięć RAM	1024 MB	2 kB
Pamięć FLASH	do 64 GB (karta micro SD)	32 kB
Procesor/mikrokontroler	64-bit, 4-rdzeniowy Broadcom BCM2837	8-bit AVR Atmega328
Zasilanie	5V z micro USB	5V z USB typu B lub 7-14V z zasilacza
Częstotliwość taktowania	4 rdzenie, każdy 1,2 GHz	16 MHz
Interfejs sieciowy	10/100 Mbps	Brak
Interfejs Wi-Fi	802.11 b/g/n 150 Mbps	Brak
Bluetooth	BLE 4.1	Brak
Złącza uniwersalne	40 pinów	32 pinów
Przetwornik ADC	Brak	6 kanałów
Wyjścia audio/wideo	HDMI, 4-polowe jack 3,5 mm+RCA	Brak
Interfejs USB	4xUSB 2.0, 1xUSB micro do zasilania	1xUSB typu B do komunikacji i zasilania
Pozostałe interfejsy	I2C, SPI, UART, CSI, DSI	I2C, SPI, UART
Możliwość podł. wyświetlacza	monitor przez HDMI, wyświetlacz poprzez SPI lub DSI	wyświetlacze tekstowe i graficzne o niskiej rozdzielcości
Kamera	kamera HD (poprzez złącze CSI lub USB)	potrzebne układy zewnętrzne
System operacyjny	niezbędny	Brak
Cena	około 170 złotych	około 100 złotych



Rys.2.6. Porównane płytki (po lewej Arduino UNO, po prawej Raspberry PI 3)

2.3.2. Silniki elektryczne oraz sterownik do silników

Silniki elektryczne to urządzenia, które zamieniają energię elektryczną na energię mechaniczną. Podstawowe parametry silników elektrycznych to prędkość obrotowa i moment obrotowy. Są one zależne od dobranego uzwojenia, zastosowanych przekładni i sterowników. Moment obrotowy powstaje w takich silnikach w wyniku oddziaływania pola magnetycznego i prądu elektrycznego. Silnik elektryczny składa się ze stojana oraz wirnika. Wyróżniamy silniki elektryczne prądu stałego, zmiennego i uniwersalne.

Wymagania, które były uwzględnione przy wyborze silnika:

- musi być to silnik szczotkowy, z powodu wysokich cen silników bezszczotkowych,
- potrzebny jest silnik z przekładnią, ponieważ projekt wymaga dużego momentu obrotowego,
- zastosowany silnik musi być silnikiem DC,
- musi posiadać prosty kształt, aby pasował do obudowy robota (zwykły silnik prostej formy, nie kątowy, nieprzymocowany do innego elementu, np. koła),
- wymagana mała waga silnika.

Sterowanie silnikami nie jest możliwe bezpośrednio z płytki Arduino, więc konieczne jest zastosowanie sterowników. Jest to urządzenie, które umożliwia zmianę prędkości i kierunku obrotu silników. Sterowniki składają się najczęściej z mostków H zabezpieczonych diodami tłumiącymi przepięcia.

Wymagania stawiane sterownikom:

- możliwość zmiany napięcia na odpowiednie do zasilania części logicznej,
- koniecznie jest, aby moduł posiadał wszystkie potrzebne elementy pasywne,
- płytki musi mieć niewielkie wymiary i małą wagę,
- układ powinien posiadać możliwość łatwego podłączenia do Arduino,
- sterownik musi być przystosowany do silników prądu stałego.

2.3.3. Akcelerometr

Akcelerometry to przyrządy do pomiaru przyspieszenia liniowego i kątowego w trzech osiach: X, Y, Z (w starszych rozwiązaniach: w jednej lub dwóch osiach). W momencie, gdy występują odchylenia od punktu początkowego, wrażliwe struktury krystaliczne wewnętrz akcelerometru generują napięcie, za pomocą którego można precyjnie określić przyspieszenie oraz prędkość obiektu. Przetwornik ten również wykrywa wibracje.

Głównym elementem czujnika jest masa bezwładna zamocowana na sprężystych belkach, stanowiąca elektrodę w układzie kondensatora pomiarowego. W wyniku przemieszczenia masy zmienia się pojemność i tym samym napięcie wyjściowe, które dalej przetworzone (np. za pomocą przetwornika A/C). Ze względu na sposób zmiany przyspieszenia wyróżniają się przyspieszeniomierze piezoelektryczne, piezorezystancyjne, pojemnościowe i termiczne. Najmniejszymi i najtańszymi są akcelerometry pojemnościowe wykonane w technologii MEMS. Te sensory są często wykorzystywane w urządzeniach mobilnych oraz innej elektronice użytkowej.

Wybór czujnika był wykonywany wśród urządzeń 3-osiowych, ponieważ jest to najbardziej nowoczesne rozwiązanie. Wybrany moduł musiał również współpracować z Arduino Uno, tzn. mieć możliwość połączyć się z CPU poprzez magistralę I2C. Uwzględniony także był zakres pomiarowy: musiał być jak największy (minimum FS=+- 3g). FS=+-3g będzie oznaczało, że jest dozwolony pomiar przyspieszeń, 3-krotnie przewyższającym średnią wartość przyśpieszenia Ziemi. Maksymalny zakres możliwych wartości z tego musi wynosić więcej niż $2 \times FS = 6g$. Rozdzielcość pomiaru, która jest najmniejszą wielkością, na którą reaguje przyrząd, w takim razie powinna wynosić około 13-16 bitów, bo jest używana przy wyznaczaniu czułości. Wzór na czułość będzie następujący:

$$2^{\text{rozdzielcość}} / (2 * FS) \quad [2.1]$$

Także wymagane były stabilizatory napięcia, które byłyby zawarte w konstrukcji, aby utrzymywać stałe napięcie.

2.3.4. Moduł Bluetooth

Moduł Bluetooth to niewielki element (układ scalony na płytce drukowanej z odpowiednimi wyprowadzeniami) spełniający rolę nadajnika i odbiornika. Po odpowiednim połączeniu z urządzeniem, zapewnia on dwustronną łączność z innym modułem Bluetooth. Dzięki modułowi Bluetooth można bezprzewodowo łączyć się z różnymi komponentami, gdzie połączenie kablem nie wchodzi w grę, np. w przypadku sterowania robotem za pomocą aplikacji.

Zasięg Bluetooth zazwyczaj mieści się w zakresie od kilku do około 100 metrów (licząc w linii prostej bez przeszkód, które utrudniają odebranie sygnału). Zasięg może zmieniać się w zależności od mocy nadajnika oraz odbiornika, przeszkód napotkanych między nimi lub innych zakłóceń. Może się wydawać, że niewielki zasięg jest wadą danego standardu łączności, jednak w rzeczywistości dana cecha rozwiązania jest wykorzystywana z powodzeniem. Takie moduły stosuje się w urządzeniach, których funkcje obsługiwane przez Bluetooth nie wymagają stałej łączności bezprzewodowej na większych odległościach.

Do przesyłu danych technologia Bluetooth wykorzystuje jedną z odmian promieniowania elektromagnetycznego - fale radiowe. Takie fale elektromagnetyczne mają długość większą niż 10-4 m (0,1 mm) oraz częstotliwość z zakresu od 3 kHz do 3 THz. W przypadku Bluetooth częstotliwość działania obserwuje się w paśmie 2,4 GHz, gdyż na tak dobranej częstotliwości nie pracuje większość innych urządzeń.

Zasadę działania modułu Bluetooth można opisać jako prosty układ, składający się z zasilania oraz anteny. Dzięki przyłożeniu zasilania do anteny, zaczynają poruszać się w niej elektrony, w skutek czego w jej pobliżu powstaje pole elektromagnetyczne na określonym obszarze. Jeżeli w takim polu znajdzie się druga antena – pole elektromagnetyczne spowoduje, że w niej również zacznie płynąć taki prąd (sygnał). Dzięki odpowiedniej modulacji generowanej fali (zmienianiu amplitudy lub częstotliwości) można przesyłać informacje – sygnał z zapisaną informacją.

Dokonując wyboru modułu Bluetooth, były brane pod uwagę następujące czynniki:

- kompatybilność z wykorzystaną płytą elektroniczną,
- mały pobór mocy,
- możliwość pracy w dwóch trybach: MASTER i SLAVE,
- bezprzewodowe połączenie z komputerem, smartfonem, tabletem,
- dostępność na rynku i przystępna cena.

2.3.5. Moduł MP3

Moduły MP3 to urządzenia, które pozwalają na odtwarzanie dźwięków uruchamianych przez program na głównej płycie. Są to miniaturowe zestawy uruchamiające dźwięk zapisany na karcie SD lub z USB. Mogą być obsługiwane "standalone" jeśli dołączona zostanie bateria, głośnik i przyciski, lub w połączeniu z płytą rozwojową, np. Arduino. Sprzętowy odtwarzacz jest dekoderem cyfrowym plików muzycznych. Zasada jego działania polega na tym, że konwertuje analogowe sygnały audio na cyfrowe, które mogą być obrabiane za pomocą komputera. Na początku odbywa się próbkowanie (sampling), po którym otrzymujemy sygnał w postaci dyskretnej. Dalej plik podlega kwantyzacji, w trakcie której zmienia się precyzyja sygnałów. Ostatnim krokiem jest przekształcenie sygnału poddanego kwantyzacji i próbkowaniu w postać cyfrową.

Istotnym parametrem tego urządzenia jest częstotliwość dyskretyzacji (próbkowania). Ta wielkość określa liczbę próbek sygnału ciągłego na sekundę uwzględnionych w syntezie sygnału dyskretnego. Podsumowując, wartość ta pokazuje chwilę, gdy przetwornik przetwarza chwilową wartość analogową na cyfrową. Częstotliwość ta musi być dwa razy większa niż najwyższa składowa częstotliwościowa w badanym sygnale. Im wyższa jest częstotliwość dyskretyzacji, tym dokładniej jest odtwarzany sygnał przez moduł MP3. Rozdzielcość bitowa dźwięku (bit depth) musi być również uwzględniana przy pracy z modułem MP3. Wysoka rozdzielcość powiększa ilość kombinacji, które mogą występować w dynamice sygnału: w przedstawieniu głośnych, jak i cichych dźwięków. Istnieje więcej możliwości do przedstawiania amplitudy dźwięku, co

polepsza stosunek dźwięku do szumu i pomaga sprawić, że dźwięk będzie głośniejszy/cichszy, unikając zaszumienia.

Dokonując wyboru modułu MP3, były brane pod uwagę następujące czynniki:

- kompatybilność z wykorzystaną płytą elektroniczną,
- małe wymiary,
- obsługa kart microSD,
- dostępność na rynku i przystępna cena,
- częstotliwość próbkowania,
- rozdzielczość bitowa dźwięku,
- posiadanie specjalnego układu dźwiękowego dla odtwarzania plików dźwiękowych z lokalnej karty SD.

3. Realizacja

W tym rozdziale przedstawione są działania, dzięki którym projekt został zrealizowany. Opisana zostanie konstrukcja ciała i głowy robota, wybór elementów elektronicznych, projekt układu, program oraz aplikacja sterująca.

3.1. Konstrukcja ciała i głowy robota

Zarówno ciało i głowa robota zostały skonstruowane ręcznie. Sposób ten został wybrany z powodu niskiej ceny wykonania, szybkości wykonania oraz prostoty operowania taką konstrukcją, np. możliwości jej przecięcia, czy wycięcia fragmentów, w których będą umieszczane inne elementy.

Ciało robota wykonane zostało techniką paper mache. W tym celu napompowano maksymalnie piłkę plażową o średnicy $\varnothing=51\text{cm}$. Zaczęto obklejać ją paskami papierowych gazet za pomocą wody zmieszanej z klejem dyspersyjnym PVA w proporcji 2:1. Położono trzy warstwy gazet, następnie warstwę ręcznika papierowego. Warstwy suszono ciepłym powietrzem suszarek, a dodatkowo w celu suszenia został wykorzystany wentylator domowy.



Rys.3.1. Ciało robota po położeniu trzech warstw gazet

Warstwy te zostawiono do zupełnego wyschnięcia na 24h. Kolejna warstwa złożona była z pasków opaski gipsowej szybkowiążącej maczanej w wodzie o temperaturze około 20°C . Po wyschnięciu i utwardzeniu się tej warstwy, położono ostatnie dwie warstwy. Tym razem były to kawałki płotna przyklejane ponownie wodą zmieszaną z klejem. Konstrukcję zostawiono do całkowitego wyschnięcia. Po wyschnięciu zaczęto nakładać kit do drewna i szpachlować, aby zakryć wszystkie niedoskonałości i wygładzić powierzchnię. Konstrukcję raz jeszcze pozostawiono do wyschnięcia. Finalnym etapem tych prac było oszlifowanie powierzchni szlifierką, aby nadać konstrukcji jak najbardziej idealny kształt kuli, w celu jak najpłynniejszego ruchu.

Główę robota tworzy piłka styropianowa pusta w środku o średnicy $\varnothing=30\text{cm}$. Odcięto kawałek na wysokość około $\frac{1}{3}$ piłki. Kolejnym krokiem było nałożenie masy szpachlowej na powierzchnię oraz brzegi w celu wyrównania i utwardzania powierzchni.



Rys.3.2. Głowa robota po położeniu warstwy masy szpachlowej

Po wyschnięciu masy oszlifowano ją papierem ściernym. Nałożona została warstwa kitu do drewna i ponownie oszlifowana.



Rys.3.3. Ciało i głowa robota przed oszlifowaniem i położeniem końcowych warstw

Ostatnim krokiem przed pomalowaniem było nałożenie dwóch warstw masy szpachlowej samochodowej na ciało i głowę robota. Warstwy te finalnie utwardziły obie konstrukcje. Po wyschnięciu i oszlifowaniu można było przejść do malowania. Na początku użyto farby białej, która była bazą dla robota. Następnie pomalowano pomarańczowe i srebrne elementy na wzór modelu BB-8.



Rys.3.4. Ciało i głowa robota po pomalowaniu

3.2. Wybór elementów elektronicznych

Jednym z najważniejszych elementów elektronicznych w projekcie jest płytka drukowana Arduino Uno. Głównym powodem wybrania tego rozwiązania jest prostota użytkowania. Nie trzeba znać budowy SO Linux oraz jego poleceń. Drugą istotną zaletą Arduino Uno jest niezawodność. Układ może być włączony/wyłączony w każdym momencie bez żadnego ryzyka uszkodzenia. W dodatku Arduino pobiera około 100 mA (dla porównania Raspberry Pi potrzebuje około 500 mA). Ostatnią cechą, która może uzasadnić wybór Arduino Uno jest niska cena (około 70 zł) i powszechna dostępność.

Podsumowując, płytka Arduino Uno została wybrana, ponieważ:

- ma niski pobór mocy,
- cechuje się niezawodnością,
- przeznaczona jest do projektów sprzętowych,
- posiada dużo modułów, chipów, pozwalających poszerzyć możliwości,
- jest stosunkowo tania,
- obecnie jest powszechnie dostępna,
- jest prosta w użytkowaniu.

Kolejnym elementem elektronicznym jest cyfrowy akcelerometr ADXL345 od Analog Devices. Jego wybór nie był jednoznaczny, ponieważ nie jest połączony z żyroskopem. Żyroskop i akcelerometr najczęściej są stosowane razem, ponieważ się uzupełniają. Akcelerometr mierzy własny ruch, natomiast żyroskop wyznacza położenie konkretnej rzeczy w przestrzeni. W dużej ilości projektów potrzebne są informacje zarówno o przemieszczeniach liniowych jak i o obrotach wokół osi. Jednak w naszym projekcie żyroskop

nie był istotnym urządzeniem, dlatego odrzucone zostały takie przyrządy jak moduł serii Grove ICM 20600+AK09918-I2C oraz moduł MPU-6050 firmy DFRobot, ponieważ kosztują dość dużo (około 50 zł) i stanowiłyby sporą część budżetu projektu. W celu użycia w konstrukcji robota był również rozpatrywany moduł Gravity-LIS DW12, który ma podobne parametry do ADXL345, ale nie został wybrany, ponieważ ADXL345 był wcześniej kupiony przez jednego z uczestników zespołu i nie ustępowała parametrycznie drugiemu rozważanego akcelerometru.

ADXL345 jest 3-osiowym akcelerometrem, co spełnia założenia z rozdziału 2.3.3. Może być wykorzystany bezpośrednio z dowolną płytą Arduino poprzez interfejs I2C. Jest to akcelerometr wykonany w technologii MEMS, co daje możliwość dokonywania pomiaru przyspieszenia w zakresie od +-16g. Dzięki wbudowanemu stabilizatorowi napięcia może być zasilany napięciem od 3V do 5V. Dodatkowym atutem tego przetwornika jest posiadanie dwóch linii przerwań zewnętrznych, co umożliwia odciążenie mocy obliczeniowej procesora pomiędzy odczytami. Maksymalna rozdzielcość pomiarów wynosi 13 bitów dla każdej z 3 osi (X,Y,Z).

Moduł ADXL345 został wybrany dlatego, że:

- jest akcelerometrem 3-osiowym,
- ma niższą cenę, niż układy akcelerometr+Żyroskop,
- jest wykonany w technologii MEMS,
- komunikuje się przez magistralę I2C,
- posiada odpowiedni zakres pomiarowy: +-16g,
- ma wbudowany stabilizator,
- posiada wymaganą rozdzielcość: 13 bitów,
- ma dwie linie przerwań zewnętrznych.

Ważnym komponentem w projekcie jest moduł Bluetooth HC-05 (ZS-040). Jest to moduł v2.0 z EDR klasy 2. Pracuje z napięciem 3,3 V, komunikuje się poprzez interfejs szeregowy UART (piny RX, TX), wspiera komendy AT. Maksymalna moc nadajnika wynosi + 4 dBm, czułość odbiornika to - 85 dB. Oprogramowaniem układowym jest HC/Wavesen 2.0-20100601 mające różne tryby AT. Tryb poleceń AT pozwala na przesłuchanie modułu Bluetooth i zmianę niektórych ustawień, jak nazwa, szybkość transmisji, czy działa w trybie podrzędnym lub nadzędnym. W przypadku użycia jako urządzenie nadzędne polecenia AT umożliwiają łączenie się z innymi urządzeniami podrzędnymi Bluetooth.

Model został wybrany ze względu na mały pobór mocy i możliwość pracy w dwóch trybach MASTER i SLAVE. Dzięki temu nie będzie potrzebna bateria o dużych gabarytach, zajmująca miejsce w i tak ograniczonym wnętrzu robota, a jeśli zaistnieje potrzeba, będzie można połączyć bezprzewodowo nie tylko aplikację, ale także ewentualny drugi identyczny moduł po użyciu trybu MASTER. Na rynku dostępny jest model HC-06 o lepszych parametrach, jednak jest on równocześnie droższy, a przy wymienionych wymaganiach i określonym budżecie wyżej wymieniony jest wystarczający. Można natknąć się również na moduły zastępcze (tzw. podróbki), ale znacznie bezpieczniejsze jest użycie oryginału.



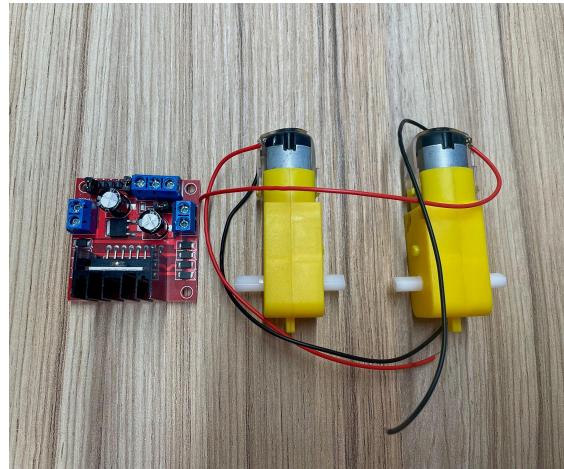
Rys.3.5. Elektronika wymieniona wyżej - Arduino Uno, akcelerometr, moduł Bluetooth

Kolejnym komponentem jest moduł MP3. Wybór padł na model WTV020M01. Jest on niewielkich rozmiarów i można wykorzystać go jako komunikator dźwiękowy w układzie zarządzanym przez uC lub może pracować jako zwykły odtwarzacz. Wyposażony jest w układ (WTV020SD-20SS) do odtwarzania plików dźwiękowych z kart microSD. Główne właściwości wybranego modułu MP3 to wsparcie mikroprocesora, zapisywanie danych podczas wyłączenia zasilania, napięcie zasilania DC 2,6 do 3,6V, prąd spoczynkowy 16 uA.

Mimo, że moduł potrafi wykorzystywać niewygodnie duże napięcie, wybór padł na ten model, ze względu na dużą pojemność pamięci potrzebną do przechowywania dźwięków wykonywanych przez oryginalnego robota, by jak najwierniej go odtworzyć. Również użycie w nim karty SD znacznie ułatwi pracę przy zgrywaniu potrzebnych dźwięków.

W projekcie zostały wykorzystane dwa silniki krokowe, w których impulsowe zasilanie prądem elektrycznym powoduje, że jego wirnik nie obraca się ruchem ciągłym, lecz wykonuje za każdym razem ruch obrotowy o ścisłe określony kąt. Został wybrany silnik TT z przekładnią 1:48 typu prostego. Jest to silnik elektryczny prądu stałego ze znamionowym napięciem zasilania 6V. Prędkość obrotowa danego urządzenia wynosi 100 RPM. Pobór mocy stosowanego silnika to 350 mA. Silnik ma małą masę (jest to spowodowane plastikową obudową) oraz małe wymiary: długość około 70 mm.

Do sterowania silników użyto modułów L298N - dwukanałowych sterowników silników dużej mocy. Sterownik zasilany jest napięciem 12V, a maksymalny prąd wynosi 2A. Układ ma możliwość regulowania napięcia w celu zasilania silników napięciem 5V. Moduł pozwala na sterowanie dwoma silnikami DC lub jednym silnikiem krokowym. Układ do sterowania silnikiem wykorzystuje mostek H. Sterownik L298N wybrano głównie ze względu na jego niską cenę, niewielkie wymiary i dobrą współpracę z Arduino.



Rys.3.6. Silniki oraz sterownik silników

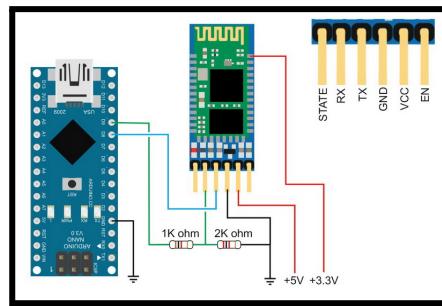
3.3. Projekt układu

Po wybraniu najlepszych do tego projektu elementów elektronicznych przystąpiono do zaprojektowania układu. Większość elektroniki umieszczona jest na platformie wewnętrznej sfery - ciała robota. Platforma przemieszcza się na dwóch kołach oraz dodatkowych rolkach.

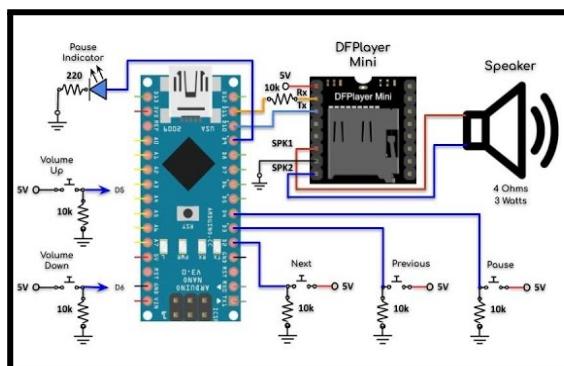


Rys.3.7. Koła wykorzystywane do platformy z elektroniką

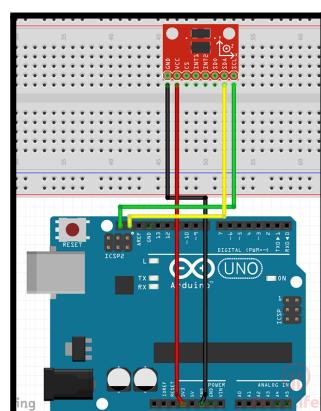
W pierwszej kolejności po zamówieniu komponentów przeprowadzono próbne połączenie układu. Poniżej przedstawiono schematy elektroniczne poszczególnych elementów oraz całego układu.



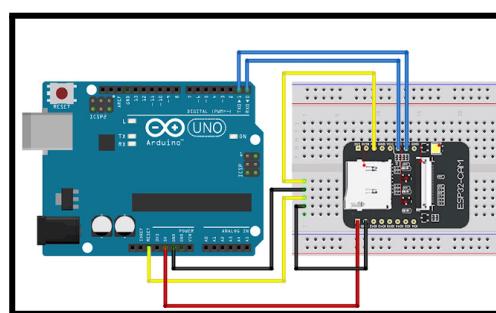
Rys.3.8. Schemat modułu Bluetooth



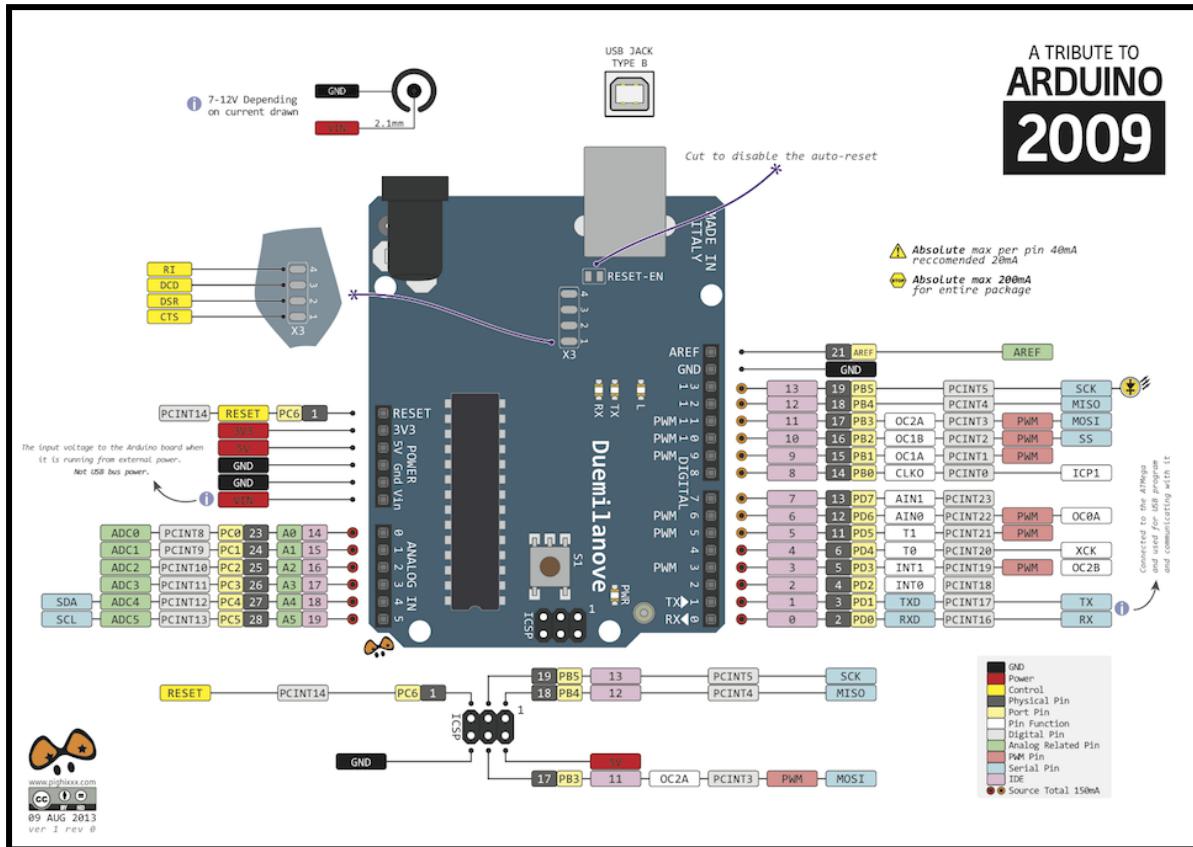
Rys.3.9. Schemat modułu MP3



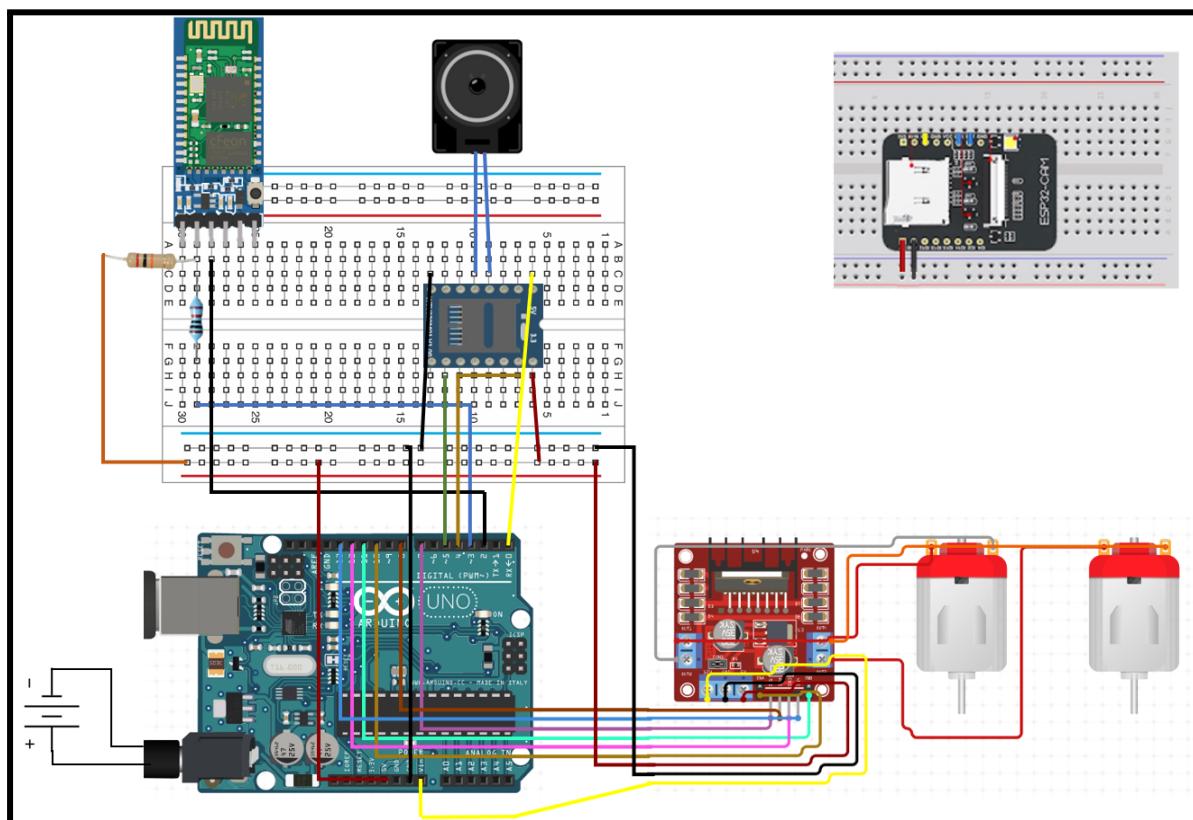
Rys.3.10. Schemat akcelerometru



Rys.3.11. Schemat modułu z kamerą

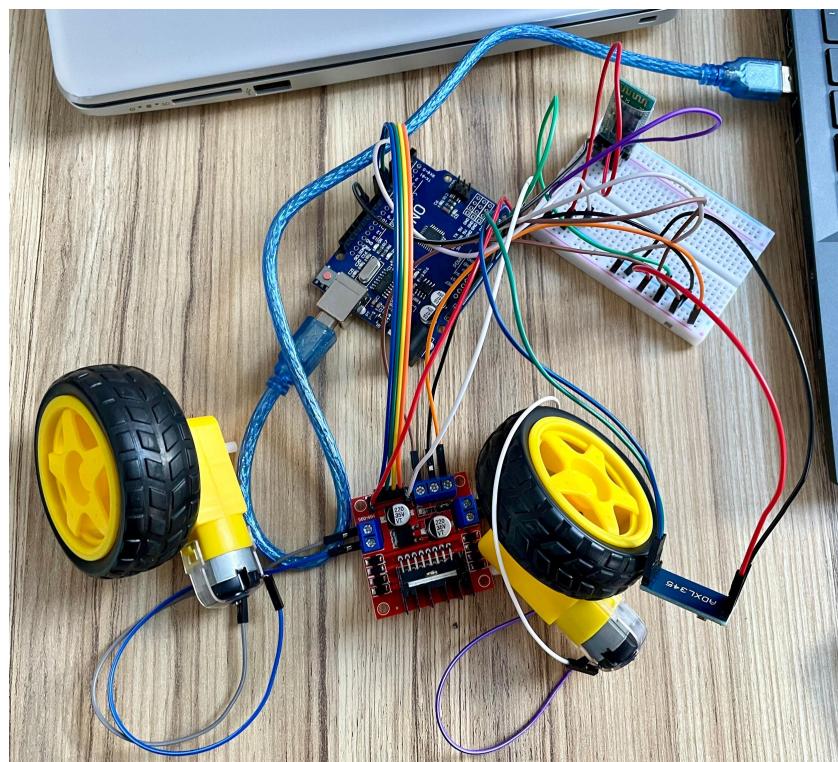


Rys.3.12. Schemat płytka Arduino UNO



Rys.3.13. Schemat próbnego połączenia układu

W połączeniu próbny płytka Arduino Uno połączona została ze wszystkimi elementami układu za pomocą przewodów, oprócz kamery, z którą kontakt docelowo ma się odbywać pośrednio poprzez PC z Wi-Fi oraz moduły Bluetooth. Zasilanie na wszystkie elementy połączone z Arduino pobierane jest z samej płytki sterującej pośrednio przez płytę stykową. Połączenia ze wszystkimi elementami poza sterownikiem silników również wykonane są poprzez płytę stykową. Do sterownika silników podłączone są 4 wyjścia cyfrowe (po 2 na silnik), odpowiadające za ich uruchomienie z możliwością zmiany kierunku poprzez zmianę polaryzacji, oraz dwa wyjścia PWM odpowiedzialne za określenie mocy wykorzystywanej przez silniki. Sterownik jest dalej podłączony przewodami do silników. Arduino jest również połączone z wejściem oraz wyjściem komunikacji szeregowej płytki Bluetooth, z pinami niezbędnymi do kontroli modułu MP3, połączonego dalej z głośnikiem, oraz z wyjściami akcelerometru, przekazującymi kontrolerowi wyniki pomiarów.



Rys.3.14. Rzeczywiste próbne połączenie układu

3.4. Program sterujący

Jednym z ostatnich etapów pracy nad realizacją projektu było stworzenie programu sterującego robotem. Ponieważ robot sterowany będzie głównie płytą Arduino Uno, większa część programu sporządzona została w softwarze dedykowanym do programowania produktów Arduino. Nazwy i przeznaczenia pinów zdefiniowane zostały na początku kodu oraz w funkcji `setup()` zgodnie ze schematem połączeń. Działania arytmetyczne i logiczne, wykorzystania dla sygnałów wejściowych oraz sposoby nadawania wartości i czasy wysyłania sygnałów wyjściowych określone zostały w funkcji `loop()` zgodnie z algorytmem sterowania. Kod źródłowy programu sterującego przedstawia się następująco:

1) Deklaracja zmiennych.

```
#include<SoftwareSerial.h>

SoftwareSerial bt(2, 3);

char inputByte;

#define mot11 7
#define mot12 8
#define mot21 12
#define mot22 13
#define m1s 10
#define m2s 11
```

W tej części odbywa się dołączenie biblioteki Software Serial, która umożliwia komunikację szeregową. Następnie, z wykorzystaniem załaczanej biblioteki, zostają zdefiniowane piny 2 i 3 mikrokontrolera jako kolejno odbiornik i nadajnik modułu Bluetooth. Zmienna inputByte służy do odczytu znaków przekazywanych z modułu Bluetooth do programu. Na końcu tej części zostały zdefiniowane piny odpowiadające za sterowanie silnikiem. Piny 7, 8, 12 i 13 odpowiadają za podawanie sygnału wysokiego i niskiego na wejścia sterownika i w konsekwencji polaryzację silników. Piny 10 i 11 wysyłają sygnał PWM i zostały użyte do ustawienia prędkości silników.

2) Ustawienia początkowe.

```
void setup() {
    // put your setup code here, to run once:
    bt.begin(38400);
    Serial.begin(38400);
    Serial.write(bt.read());
    pinMode(mot11, OUTPUT);
    pinMode(mot12, OUTPUT);
    pinMode(mot21, OUTPUT);
    pinMode(mot22, OUTPUT);
    pinMode(m1s, OUTPUT);
    pinMode(m2s, OUTPUT);

    analogWrite(m1s, 0);
    analogWrite(m2s, 0);
}
```

Funkcja setup odpowiada za zdefiniowanie w programie ustawień początkowych. Przepływ danych został ustalony na 38400 bitów na sekundę. Piny służące do sterowania silnikiem zostały ustawione jako wyjścia a prędkość początkowa silników została ustaliona na 0.

3) Główna funkcja sterująca.

```
void loop() {
```

```

// put your main code here, to run repeatedly:

if (bt.available()) /* If data is available on serial port */
{
    inputByte = bt.read();
    Serial.println(inputByte);

    if (inputByte == 'w')
    {
        analogWrite(m1s, 255);
        analogWrite(m2s, 255);
        digitalWrite(mot11, LOW);
        digitalWrite(mot12, HIGH);
        digitalWrite(mot21, LOW);
        digitalWrite(mot22, HIGH);
    }
    .
    .
    .

    else if (inputByte == 'q')
    {
        analogWrite(m1s, 0);
        analogWrite(m2s, 0);
        digitalWrite(mot11, LOW);
        digitalWrite(mot12, LOW);
        digitalWrite(mot21, LOW);
        digitalWrite(mot22, LOW);
    }
}
if (Serial.available())
{
    bt.write(Serial.read());
}
}

```

Funkcja loop wykonyuje powtarzalnie ten sam kod. Na początku sprawdzane jest czy zostały przesłane dane z modułu Bluetooth. Jeżeli pojawiły się dane, są one odczytywane i na ich podstawie zostaje wybrane działanie. Wysłanie "w" powoduje obrót silników w tą samą stronę i w konsekwencji ruch do przodu. Analogicznie, "s" powoduje ruch w tył, "a"- w lewo, a "d"- w prawo. Możliwe jest również zatrzymanie silników za pomocą "q". Po wybraniu odpowiedniego ruchu dane przesyłane są do mikrokontrolera.

W miarę postępu projektu, do pliku zaczęto dodawać kod odpowiedzialny za obsługę modułu mp3 z głośnikiem. Na początku dodano parę linijek, które przeznaczają konkretne piny do kontroli modułu mp3:

```

int resetPin = 2; // The pin number of the reset pin.
int clockPin = 3; // The pin number of the clock pin.
int dataPin = 4; // The pin number of the data pin.
int busyPin = 5;
WTW020SD16P wtv020sd16p(resetPin,clockPin,dataPin,busyPin);

```

Następnie w sekcji *setup()* go zresetowano i odciszyono:

```

wtv020sd16p.reset();
wtv020sd16p.unmute();

```

Na koniec w pętli programu umieszczono funkcje odpowiedzialne za załączanie wybranego pliku dźwiękowego spośród zawartych na karcie micro sd znajdującej się w module mp3 (w tym przykładzie pliku oznaczonego numerem 0001):

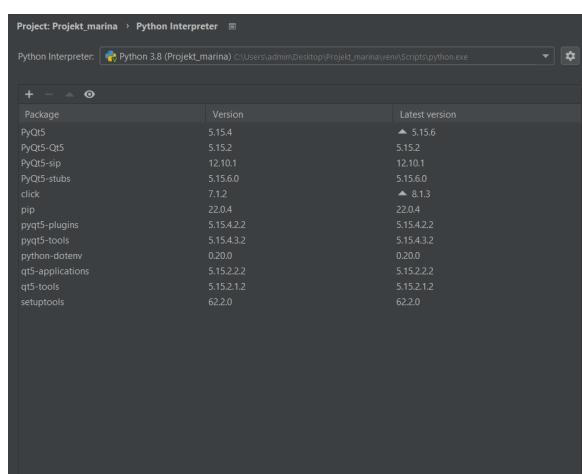
```
wtv020sd16p.playVoice(1);  
wtv020sd16p.stopVoice();
```

3.5. Aplikacja sterująca

W celu stworzenia aplikacji, pozwalającej na połączenie pomiędzy panelem operatorskim (komputerem) oraz systemem wbudowanym (mikrokomputerem) będzie stosowany Python. Wybór narzędzia do programowania dokonywany był pomiędzy Pythonem z biblioteką qt5, a C++ w Qt Creator. Została podjęta decyzja o wyborze Python 3.8 ponieważ okazało się, że Qt w wersji 7.0.0 posiada pewne wady w obsłudze połączeń szeregowych Bluetooth.

Na początku zostały zainportowane biblioteki PyQt5 oraz PyQt5-tools za pomocą Settings. Zainstalowane pakiety są przedstawione na rys.3.15. W kolejnym kroku została stworzona klasa, odpowiadająca za połączenie z płytą Bluetooth, przedstawiona poniżej.

```
import serial  
  
class RobotSerial:  
    UP = bytearray([119])  
    DOWN = bytearray([115])  
    LEFT = bytearray([97])  
    RIGHT = bytearray([100])  
    NOTHING = bytearray([113])  
  
    def __init__(self, port='COM5', baudrate=38400):  
        self.serialPort = serial.Serial(port=port, baudrate=baudrate)  
        print(f"port={port}")  
        self.size = 1024  
  
    def send_byte(self, byte: bytearray):  
        self.serialPort.write(byte)
```

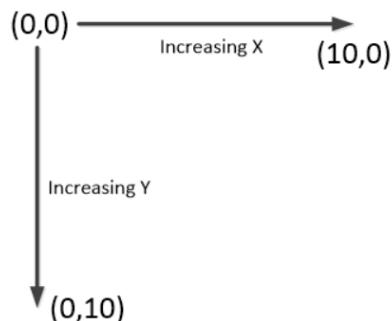


Rys.3.15. Zainportowane biblioteki

Ta klasa będzie obsługiwać przyciski, niezbędne do ruchu robota: Up, Down, Left, Right. Takie przyciski zostaną utworzone w pliku głównym projektu. Najbardziej skomplikowaną częścią było tworzenie GUI, ponieważ było przeprowadzone bez żadnych dodatkowych narzędzi (programów). Przy rozmieszczaniu obiektów: przycisków, menu oraz pictureboxów był uwzględniony fakt, że Python ma takie koordynaty jak na rys.3.16. Punkt (0,0) jest punktem początkowym. Dla podania położenia używany jest setGeometry(x,y).

W projekcie wykorzystano narzędzia i metody przedstawione poniżej.

1. setGeometry (x,y,szerokość,wysokość) - położenie i parametry geometryczne
2. setBrush - wypełnienie figur
3. setStyle,setPalette - wizualna część
4. setObjectName - dodanie nazwy obiektu
5. setText - wyświetlenie tekstu
6. setWindowTitle - nazwa okna
7. QPushButton - przycisk
8. connect() - łączenie funkcji z wykonaniem akcji
9. menubar - menu wyboru
10. label - tekst do wyświetlenia
11. picturebox - zdjęcie
12. setShortcut - narzędzie do nazwy skrótów



Rys.3.16. Sposób opisu punktów w Python

Wymienione komponenty są najważniejszymi w kodzie programu. Poniżej zostały umieszczone najważniejsze części kodu z komentarzami.

a) Import bibliotek:

```
import sys
from PyQt5 import QtCore, QtGui, QtWidgets
from RobotSerial import *
```

b) Tworzenie klasy głównego okna:

```
class Ui_MainWindow(object):
```

c) Definicja wyglądu wizualnego:

```
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Dark, brush)
```

d) Przycisk Up:

```
self.Up = QtWidgets.QPushButton(self.centralwidget)
```

e) Dodanie geometrii dla Up2:

```
self.Up_2.setGeometry(QtCore.QRect(210, 520, 131, 61))
```

f) Dodanie zdjęcia:

```
elf.photo.setPixmap(QtGui.QPixmap("C:\\\\Users\\\\admin\\\\Desktop\\\\..\\Downloads/bb-8-star-wars.webp"))
```

g) Dodanie menu:

```
self.statusbar = QtWidgets.QStatusBar(MainWindow)
```

h) Dodanie menu file:

```
self.menuFile = QtWidgets.QMenu(self.menubar)
```

i) Dodanie menu porty:

```
self.menuPorty.setObjectName("menuPorty")
```

j) Akcja - usuń plik:

```
self.actionDelete_file.setObjectName("actionDelete_file")
self.menuFILE.addAction(self.actionDelete_File)
```

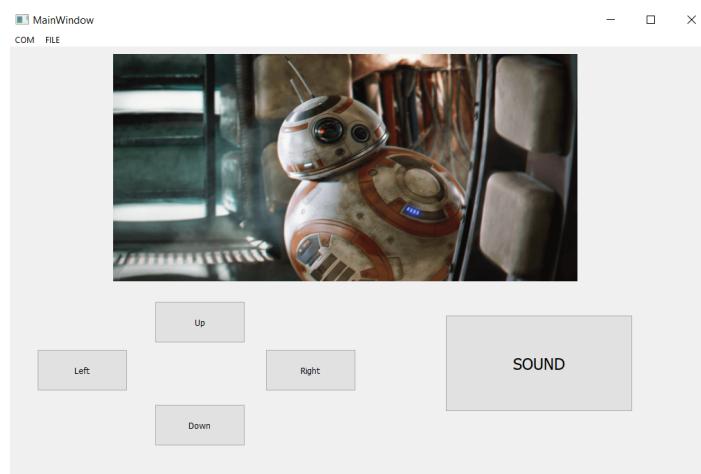
k) Dodanie tekstu na Up:

```
self.Up.setText(_translate("MainWindow", "Up"))
```

l) Ruch w dół przy naciśnięciu Down:

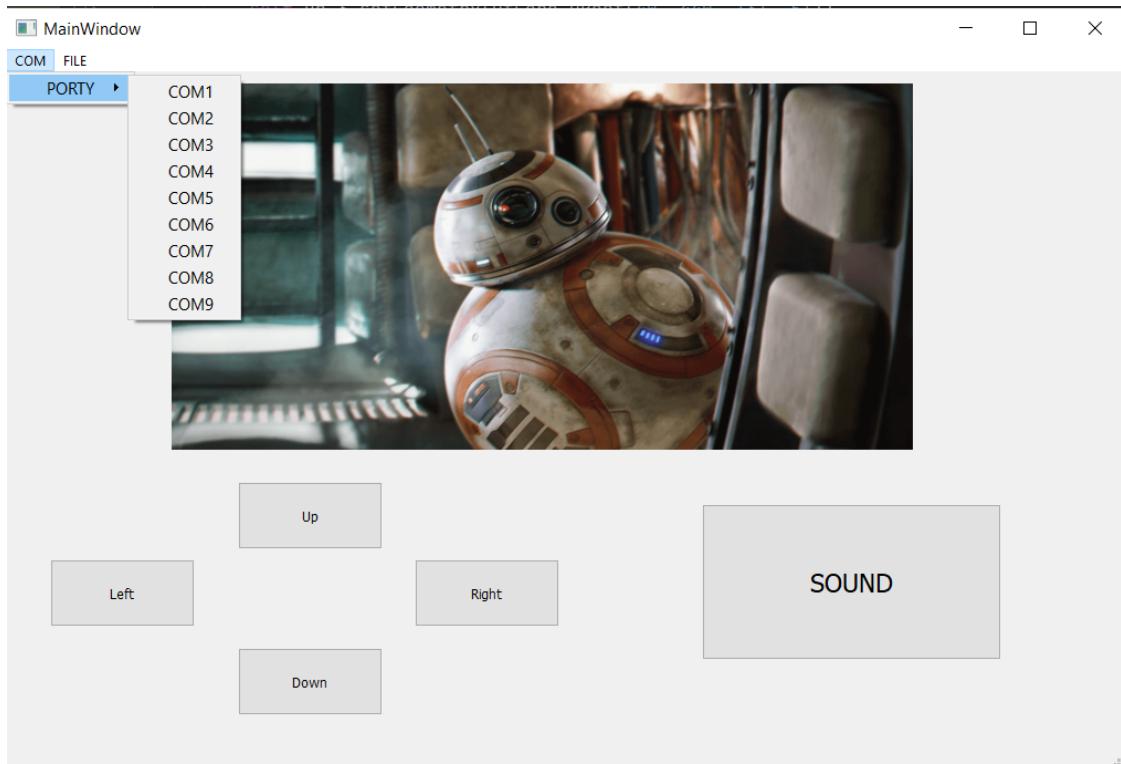
```
def clicked_down(self):
    print("DOWN")
    self.robot.send_byte(RobotSerial.DOWN)
```

Stworzony GUI będzie wyglądać w sposób przedstawiony poniżej.

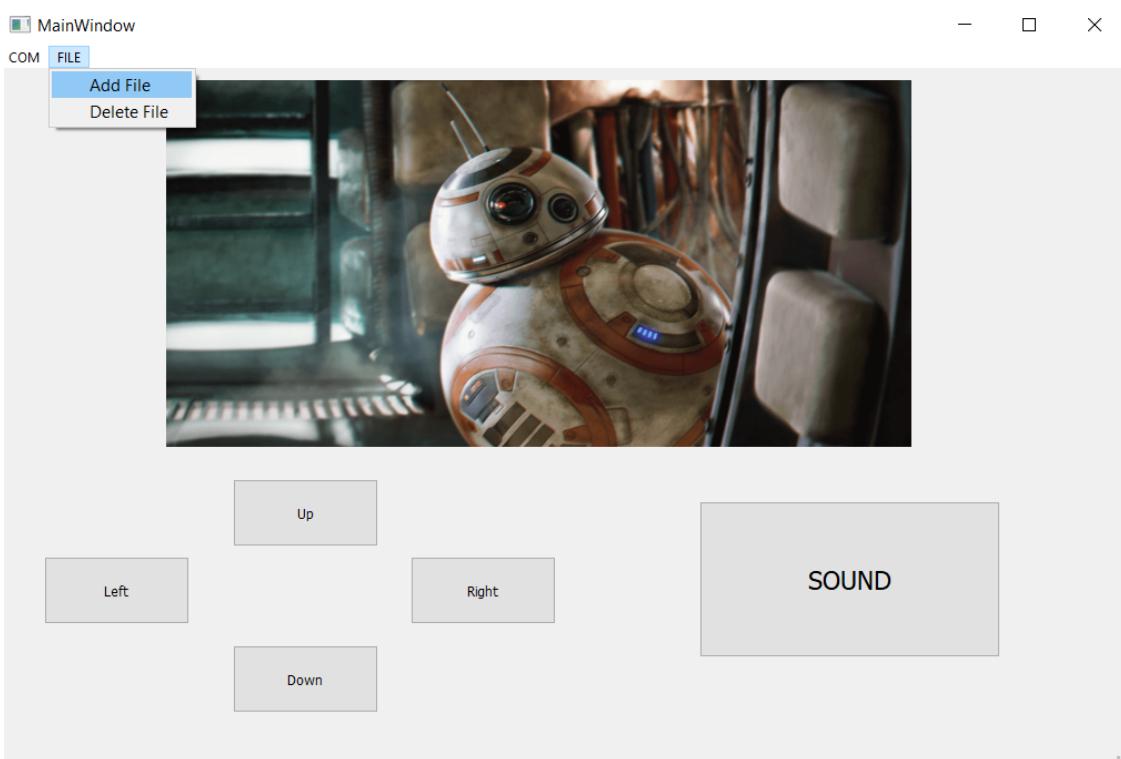


Rys.3.17. Wygląd GUI

Menu na górze można rozwinąć, żeby dokonać wyboru portu oraz plików z kamery robota w okienku, gdzie obecnie wstawiony jest obraz demonstracyjny.

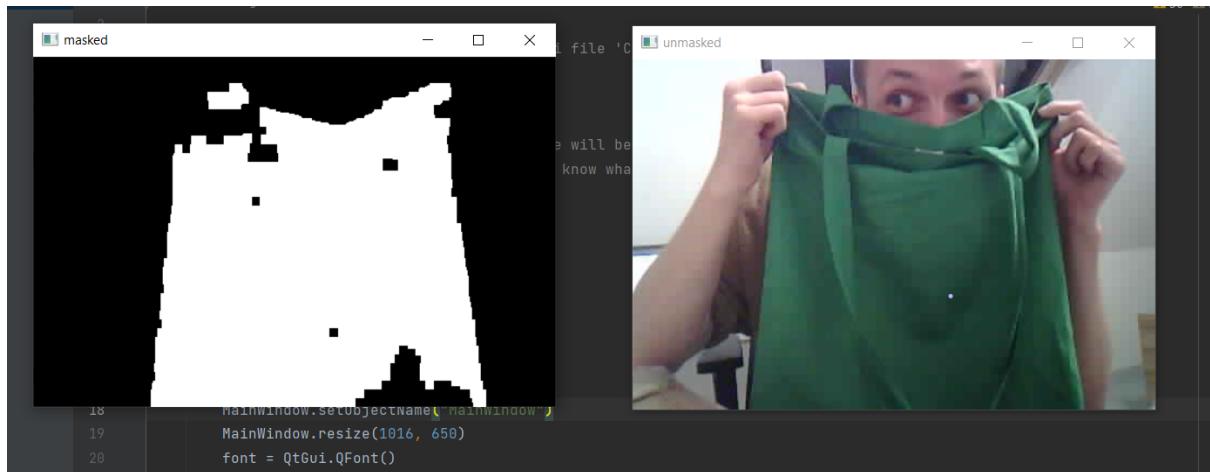


Rys.3.18. Menu COM



Rys.3.19. Menu File

Aby zapewnić wyświetlanie programu dla kamery, która wykrywa kolor zielony, w aplikacji użyta została wielowątkowość. Każdy wątek to osobna jednostka wykonawcza działająca w obrębie tego samego procesu. Jednocześnie obsługiwany jest plik główny, który odpowiada za łączenie Bluetooth oraz plik zapewniający przetwarzanie obrazu. Obraz z kamery oraz maska są pokazywane w osobnym oknie, ponieważ prowadzi to do zniekształceń obrazu w wyniku rozciągania/zmniejszenia. Na rys. 3.20 jest zaprezentowany wynik działania programu, czyli wykrycie zielonego koloru torby. Jak widać, program nie wykrywa człowieka tylko torbę.



Rys.3.20. Wykrycie koloru zielonego przez kamerę

Po nawiązaniu połączenia między kamerą, a komputerem poprzez sygnał wifi, w pętli while true dochodzi do pobrania klatki obrazu w standardzie BGR (Blue, Green, Red).

```

img_resp = urllib.request.urlopen(url)
imgnp = np.array(bytarray(img_resp.read()), dtype=np.uint8)
frame = cv2.imdecode(imgnp, -1)

```

Następnie konwertowany jest on na standard HSV (Hue, Saturation, Value), w celu łatwiejszego wyciągania konkretnego koloru z mniejszą zależnością od oświetlenia.

```
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

Po konwersji tworzona jest maska, która sprawdza, czy wartości poszczególnych pikseli znajdują się pomiędzy ustalonymi granicami wartości zdefiniowanymi w standardzie HSV. W ten sposób piksele odpowiadające założeniom zostają podmienione na białe, a nieodpowiadające na czarne.

```

lower_bound = np.array([55, 50, 50])
upper_bound = np.array([90, 255, 255])
mask = cv2.inRange(hsv, lower_bound, upper_bound)

```

Dodano również funkcję redukującą szumy w postaci białych kropek w czarnych polach i na odwrót.

```

kernel = np.ones((7, 7), np.uint8)
mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)

```

Na sam koniec sprawdzane jest, jaka część pola widzenia robota po przetworzeniu przez maskę jest biała. Jeśli znajduje się ona powyżej ustalonej granicy, uznaje się, że robot wykrył kolor wyzwalający i zostaje ustalona wartość odpowiedniej zmiennej. W przypadku, gdy zawartość białych pikseli spadnie poniżej granicy, wartość zmiennej wraca do stanu sygnalizującego brak wykrycia koloru.

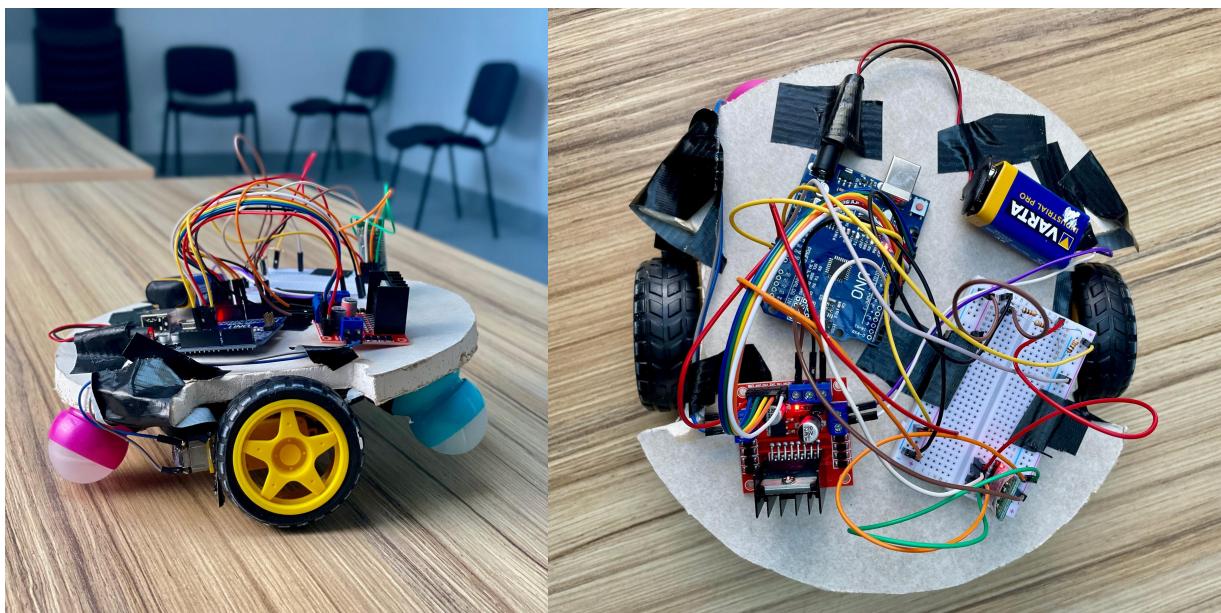
```
posmask = mask[np.where(mask != 0)]
if len(posmask) / (200*len(mask)) >= 0.4:
    out = 1
else:
    out = 0
```

4. Testy

W tym rozdziale opisane zostaną testy dotyczące działania robota. Omówione zostaną pierwsze testy i poprawki, jakich dokonano oraz testy końcowe. Przedstawione zostaną wyniki testów i wykryte niedociągnięcia.

4.1. Poprawki

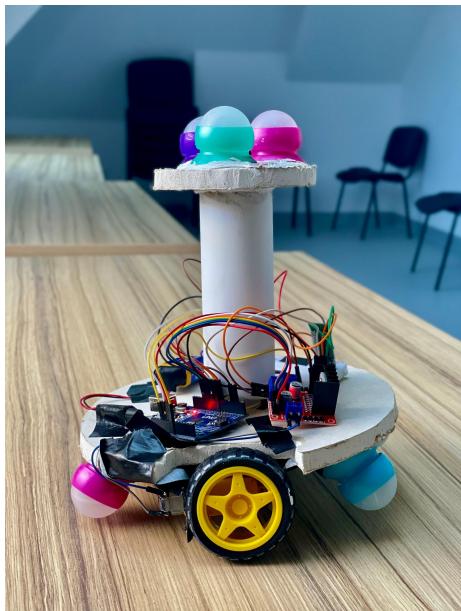
Pierwsze testy rozpoczęto od wycięcia z karton-gipsu platformy z wcięciami na koła. Platforma ta porusza się wewnątrz kuli-ciała robota. Następnie przyklejono rolki oraz koła wraz z ich silnikami do jej dolnej części i sprawdzono stabilność konstrukcji oraz jej wytrzymałość. Po pomyślnym wykonaniu tych działań, do górnej części platformy przyczepiona została elektronika wraz z zasilającą baterią. Połączono się bezprzewodowo z układem za pomocą aplikacji Bluetooth Terminal i wykonano testy działania obrotu i zatrzymania silników. Gdy działanie silników i kół było zadowalające - konstrukcja przemieszczała się we właściwych kierunkach - postanowiono umieścić całą platformę w ciele robota. Kulę-ciało robota przecięto uprzednio na pół i wyjęto ze środka piłkę, na której tworzono całą konstrukcję. Do obu połówek przyklejono rzep, aby łatwo je można było połączyć i rozłączyć, gdy to potrzebne.



Rys.4.1. Platforma przemieszczająca się w środku ciała robota

Po umieszczeniu platformy w kuli szybko okazało się, że koła przytwierdzone do platformy są za małe (zamówiono większe - rys. 3.7), gdyż nie stykały się z wewnętrzną powierzchnią kuli. Aby wciąż móc wykonać testy na zastosowanych kołach, doklejonych zostało kilka warstw tektury do silników, aby wraz z kołami je obniżyć. Poprawka ta, chociaż prosta, przyniosła oczekiwany efekt - koła dotykały powierzchni kuli i zaczęły po niej jeździć. Domknięto obie połówki kuli i ponownie wykonano testy. Platforma z kołami zaczęła przesuwać się po wewnętrznej powierzchni kuli, dzięki czemu kula toczyła się po ziemi.

Ostatecznie wykonano to samo działanie z dedykowaną do tego projektu aplikacją, zamiast z gotowym terminaliem Bluetooth, co również okazało się udane. Do takiego następstwa wszystkich działań próbowało doprowadzić w pierwszej fazie testów.



Rys.4.2. Konstrukcja platform

Do kolejnej fazy testów przyklejono do głównej platformy stabilizator w postaci rury PVC wraz z zamontowaną mniejszą platformą na jej drugim końcu (rys. 4.2). Mniejsza platforma będzie wykorzystywana nie tylko do stabilizacji dolnej, ale również zatwierdzone będą na niej magnesy, które przeznaczone będą do utrzymywania głowy robota cały czas w tej samej pozycji. Mniejsza, górnna platforma, tak samo jak dolna, posiada przyklejone do niej rolki, które będą przemieszczały się po wewnętrznej powierzchni kuli.

Do górnej platformy przytwierdzone zostały 3 neodymowe magnesy walcowe o średnicy 20 mm i wysokości 5 mm. Są to stosunkowo niewielkie wymiary, jednak magnesy okazały się bardzo mocne i dobrze trzymają głowę robota. Różnica pomiędzy magnesami neodymowymi, a typowymi, ferrytowymi, polega na tym, że magnesy neodymowe są silniejsze, więc mogą zapewnić odpowiednią wartość strumienia magnetycznego przy mniejszych wymiarach. Jest to dużą zaletą, gdyż w projekcie potrzebne są magnesy o dużym strumieniu magnetycznym, żeby trzymać głowę cały czas w jednym miejscu, a specyfikacja konstrukcji robota nie pozwoliła na wykorzystanie magnesów o dużych wymiarach. Drugą, niemniej ważną zaletą magnesów neodymowych jest fakt, że wytwarzają one jednorodne pole w szczelinie, stąd zniekształcenia wobec innych urządzeń elektronicznych są niższe (pod warunkiem, że neodym nie zostanie narażony na działanie zbyt wysokich temperatur, ale ze specyfikacji konstrukcji robota nie grozi to magnesom). Jest to istotne, ponieważ w okolicy magnesów znajduje się elektronika, która nie może zostać zakłócona przez pole magnetyczne.

Aby głowa robota mogła trzymać się na ciele, potrzebne jest stworzenie bliźniaczej konstrukcji magnetycznej po stronie zewnętrznej kuli. Do tego zostały wykorzystane kolejne

3 magnesy, drut metalowy w obudowie izolacyjnej oraz rolki. Z metalowego drutu wygięto figurę o kształcie trójkąta, na końcach którego przyklejono magnesy. Bardzo ważną kwestią było poprawne przeklejenie magnesów, aby nie zmienić ich polaryzacji - wtedy magnesy, odpychałyby się nawzajem, co jest działaniem przeciwnym, które chciano uzyskać. Pomiędzy magnesami umieszczono rolki, które również przyklejono do trójkątnej konstrukcji. Będą one przemieszczały się po zewnętrznej powierzchni kuli, co spowoduje następstwo ruchu głową robota.



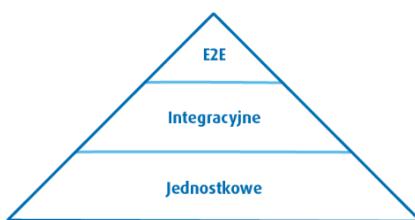
Rys.4.3. Konstrukcja górnej platformy wraz z magnesami

4.2. Końcowe testy

Testowanie było podzielone na dwa etapy:

- testowanie kodów,
- testowanie fizycznych właściwości robota.

Testy kodów oparte były na wykorzystaniu niżej przedstawionej piramidy. Piramida testów to graficzny model, który reprezentuje rodzaje testów oraz ich stosunek (proporcje). W podstawie piramidy znajdują się testy szybko wykonywalne, na wyższych poziomach - wolniej wykonywalne.



Rys.4.4.Piramida testów

Pierwszy poziom to testy jednostkowe. Testuje się w nich małe elementy programów

(jednostki). Na tym etapie sprawdzane były działania kodów pod względem logicznym. Ich wykonanie trwało kilka milisekund, dlatego były wykonywane w dużej ilości.

W drugim etapie zostały dokonane testy integracyjne, które polegały na sprawdzeniu współdziałania modułów. Na tym poziomie zbadana została interakcja pomiędzy nimi. Wymagało to więcej czasu, ponieważ kilka rzeczy wykonywane było jednocześnie.

Ostatni, trzeci etap (poziom) przedstawia testy wykonane na końcu: testy akceptacyjne (E2E). Do wykonania danego etapu została zaproszona osoba, która nigdy nie miała kontaktu z robotem. Za pomocą wskazówek przechodziła wszystkie etapy uruchomienia robota/aplikacji metodą end-to-end (od początku do końca). Celem tych testów była obserwacja zachowania użytkownika końcowego.

W trakcie testów właściwości fizycznych uwzględnione zostały takie rzeczy jak:

- wytrzymałość,
- opóźnienia w reakcjach,
- poprawność toczenia.

Wytrzymałość badana była poprzez puszczanie robota na różnych powierzchniach (gładki parkiet, płytki ceramiczne) przy różnych prędkościach sterowania. Nie wykryto żadnych niekorzystnych zjawisk: wszystkie mocowania, złożone struktury pozostały na swoich miejscach.

Opóźnienia w reakcjach były liczone za pomocą zegara wbudowanego oraz diody. Liczony był czas pomiędzy naciśnięciem klawiszy w aplikacji i rzeczywistym ruchem robota (sygnał na diodzie). Opóźnienia zostały uznane za tolerowalne.

Poprawność toczenia była ważną częścią testów, ponieważ robot jest kulą i cały mechanizm jego działania opiera się na wykorzystaniu tego kształtu. Po małym wyrównaniu robota i korekcji nierówności (nie 100%, ponieważ jest to praca manualna) robot zaczął poruszać się w sposób kontrolowany.

5. Podsumowanie

W tym rozdziale zostanie przedstawione podsumowanie całego projektu i związanej z nim pracy oraz fazy testów. Opisane zostaną również krótko informacje, jakie można znaleźć w poszczególnych rozdziałach. We wnioskach końcowych zawarty będzie też przegląd wiedzy nabytej w trakcie realizacji projektu.

5.1. Streszczenie pracy

Celem pracy było zaplanowanie konstrukcji oraz zbudowanie robota mobilnego PG-8 inspirowanego modelem BB-8. Zaplanowanie konstrukcji było działaniem znacznie bardziej efektywnym niż samo jej wykonanie. Podczas wykonywania rzeczywistej konstrukcji (w tym układu elektronicznego) napotkano wiele problemów, z których niektóre trzeba było rozwiązać w inny sposób niż początkowo zaplanowany. Faza testów początkowo wymagała wprowadzenia nowych konstrukcji i kilku poprawek, ale testy końcowe przebiegły pomyślnie.

W pracy można znaleźć pięć rozdziałów, kolejno: *Wstęp, Informacje wstępne o robocie, Realizacja, Testy oraz Podsumowanie*. W rozdziale *Wstęp* przedstawiony został cel i zakres niniejszej pracy. Rozdział *Informacje wstępne o robocie* opisuje takie rzeczy, jak idea i rys historyczny robota, jego zasada działania i wykorzystywane algorytmy oraz opis elementów elektronicznych możliwych do użycia w projekcie. W *Realizacji* omówiony został sposób konstrukcji ciała i głowy robota, dobór konkretnych elementów elektronicznych i projekt układu, wykonany program sterujący oraz aplikacja do zdalnego sterowania. W *Testach* udokumentowano zaistniałe problemy i niedociągnięcia oraz wprowadzone poprawki, pierwsze oraz końcowe - najważniejsze testy. W ostatnim rozdziale, czyli w *Podsumowaniu* zapoznać się można z podsumowaniem całego projektu, streszczeniem dokumentacji i wykonanej pracy oraz wnioskami końcowymi i przeglądem wiedzy nabytej podczas realizacji projektu.

5.2. Wnioski końcowe

Projekt robota mobilnego PG-8 w momencie oddania niniejszej dokumentacji, jest wciąż w trakcie testów. Z pewnością wiele rzeczy można w nim poprawić i usprawnić, jednak robot działa zgodnie z założeniami. Jest robotem mobilnym, więc realizuje swoje podstawowe przeznaczenie - porusza się. Pełni również dodatkowe funkcje, jak utrzymywanie głowy na szczycie konstrukcji, wydawanie dźwięków oraz rozpoznawanie kolorów.

Podczas całego okresu wykonywania projektu nabyto wiele nowych umiejętności i zapoznano się z wieloma zagadnieniami z różnych dziedzin. Na samym początku należało zrobić dobre rozeznanie na temat samego robota, jego konstrukcji i zasad działania. Podczas konstruowania ciała i głowy dowiedziano się wiele z zakresu materiałoznawstwa (między innymi o tym, jak dobierać odpowiednie materiały do siebie), nauczono się też korzystania z urządzeń, takich jak szlifierki. Aby stworzyć układ elektroniczny wymagane było dobrze zapoznanie się ze wszystkimi komponentami. Okazało się, że wiele urządzeń nie jest ze sobą

kompatybilnych i konieczna była wymiana kilku z nich. Zrozumiano, jak ważne jest precyzyjne dopasowywanie elementów oraz tworzenie schematów połączeń. Odkryto, że duże znaczenie dla niektórych elementów ma natężenie światła (przede wszystkim dla kamery), a także wpływ warunków pogodowych. Do tego należało pozostać uważnym przy wprowadzaniu magnesów do konstrukcji, aby nie zakłócić działania innych elementów. Do stworzenia aplikacji sterującej konieczne było podjęcie decyzji dotyczącej użytego narzędzia do programowania. Po próbach stworzenia aplikacji za pomocą jednego narzędzia, zmieniono zamysł i samo narzędzie w celach ułatwienia pracy. Oprócz tego typu prac, nauczono się też zarządzania czasem, rozdzielania obowiązków, pracy zespołowej i dobrej komunikacji między członkami zespołu.

Finalnym wnioskiem jest to, że założenia i modele matematyczne często są niezgodne z rzeczywistością. Konieczne było wprowadzenie wielu poprawek i zmian, aby robot działał zgodnie z przyjętymi założeniami. Ważną rzeczą okazało się sprawdzanie działania robota po wprowadzeniu każdej, nawet najmniejszej zmiany.

BIBLIOGRAFIA

- [1] R.Siegwart, *Autonomous Mobile Robots*, EPFL
- [2] Ceccarelli Marco, *Design and Prototypes of Mobile Robots*, ASME Press 2015
- [3] Jones, Joseph L.; Seiger, Bruce A.; Flynn, Anita M., *Mobile Robots, Inspiration to Implementation*, CRC Press, 2019
- [4] Informacje ogólne na temat modelu BB-8, <https://en.wikipedia.org/wiki/BB-8> (data dostępu 13.04.2022)
- [5] Instrukcja pomocnicza budowy przykładowego robota inspirowanego modelem BB-8, <https://www.instructables.com/DIY-Life-Size-Phone-Controlled-BB8-Droid/> (data dostępu 13.04.2022)
- [6] Instrukcja dodatkowa budowy przykładowego robota inspirowanego modelem BB-8, <https://www.diymachines.co.uk/star-wars-bb8-remote-control-printed-robot> (data dostępu 13.04.2022)
- [7] Przydatne wzory, <http://mech.math.msu.su/~fpm/ps/k05/k05802.pdf> (data dostępu 13.04.2022)
- [8] Instrukcja budowy transparentnego robota inspirowanego modelem BB-8, https://www-instructables-com.cdn.ampproject.org/c/s/www.instructables.com/Life-Size-Remote-Control-transparent-BB-8/?amp_page=true&fbclid=IwAR2wdd_3KyN98bSmwe-lxALPdCbBFv_wfwAX98VMkTQhUO_L5jYmN76Yv5Y (data dostępu 13.04.2022)
- [9] Opis działania robota BB-8, <https://howbb8works.com/> (data dostępu 13.04.2022)
- [10] Realizacja budowy robota BB-8 z użyciem BeagleBone, <https://makezine.com/projects/build-full-size-working-bb-8-using-beaglebone/?fbclid=IwAR0wCOHNds6lCe4h3vqN09c5PifEmPoBJ17woTK6wb2xwl6pM-ODMGlhhYM> (data dostępu 13.04.2022)
- [11] Sklep z częściami, <https://botland.com.pl/> (data dostępu 13.04.2022)
- [12] Rys.2.1,
https://media.cnn.com/api/v1/images/stellar/prod/151207112137-bb-8-sketch-jj-abrams.jpg?q=h_2056,w_1577,x_0,y_64,c_crop/h_2056,w_3654,c_ipad,b_rgb:061015/h_270,w_480 (data dostępu 13.04.2022)
- [13] Rys.2.2,
https://static.wikia.nocookie.net/disney/images/c/cf/SW_TFA_-_BB_escapes.PNG/revision/latest/top-crop/width/360/height/450?cb=20160405204941 (data dostępu 13.04.2022)
- [14] Rys.2.3,
https://static.wikia.nocookie.net/star-wars-canonical/images/5/51/Star-wars-bb-8-remote-controlled-robot-white_sku-header.png/revision/latest?cb=20160519234057 (data dostępu 13.04.2022)
- [15] Rys.2.4,
<https://www.shaunchng.com/wp-content/uploads/2015/11/bb8-e1498363302840.jpg> (data dostępu 13.04.2022)

- [16] Rys.2.5,
https://i0.wp.com/hackaday.com/wp-content/uploads/2016/06/bb8_hamster_type_turning.jpg?w=414&h=314&ssl=1 (data dostępu 13.04.2022)
- [17] Framework QT, <https://www.qt.io/offline-installers> (data dostępu 13.04.2022)
- [18] Schemat WTV020M01, https://electrongoobs.com/eng_arduino_tut8.php (data dostępu 13.04.2022)
- [19] Schemat ESP32Cam,
<https://create.arduino.cc/projecthub/CiferTech/setting-up-esp-cam-with-arduino-no-more-usb-ttl-35467a> (data dostępu 13.04.2022)
- [20] Schemat MP3, <https://www.youtube.com/watch?v=jyQQi-ay8cs> (data dostępu 13.04.2022)
- [21] Schemat Arduino UNO, <https://riptutorial.com/arduino/example/15314/arduino-uno-r3> (data dostępu 13.04.2022)
- [22] Źródła danych do tabeli 2.1,
https://forbot.pl/blog/raspberry-pi-vs-arduino-ktora-plata-forme-wybrac-id10439?fbclid=IwA1DdzJVpvpuRD3tbnPHESdyjakzWOzfp_pjJMQep3qMt2CebHEaWnQGPM (data dostępu 13.03.2022)

WYKAZ RYSUNKÓW

- Rys.2.1 - Pierwszy szkic BB-8, wykonany przez Jeffreya Abramsa
Rys.2.2 - Kadr z filmu Gwiezdne Wojny
Rys.2.3 - Konstrukcja robota - tocząca się kula, utrzymująca głowę na górze
Rys.2.4 - Elementy droida BB-8
Rys.2.5 - Zasada balansowania i sterowania BB-8
Rys.2.6 - Porównane płytki (po lewej Arduino UNO, po prawej Raspberry PI 3)
Rys.3.1 - Ciało robota po położeniu trzech warstw gazet
Rys.3.2 - Głowa robota po położeniu warstwy masy szpachlowej
Rys.3.3 - Ciało i głowa robota przed oszlifowaniem i położeniem końcowych warstw
Rys.3.4 - Ciało i głowa robota po pomalowaniu
Rys.3.5 - Elektronika wymieniona wyżej - Arduino Uno, akcelerometr, moduł Bluetooth
Rys.3.6 - Silniki oraz sterownik silników
Rys.3.7 - Koła wykorzystywane do platformy z elektroniką
Rys.3.8 - Schemat modułu Bluetooth
Rys.3.9 - Schemat modułu MP3
Rys.3.10 - Schemat akcelerometru
Rys.3.11 - Schemat modułu z kamerą
Rys.3.12 - Schemat płytki Arduino UNO
Rys.3.13 - Schemat próbnego połączenia układu
Rys.3.14 - Rzeczywiste próbne połączenie układu
Rys.3.15 - Zainportowane biblioteki
Rys.3.16 - Sposób opisu punktów w Python
Rys.3.17 - Wygląd GUI
Rys.3.18 - Menu COM
Rys.3.19 - Menu File
Rys.3.20 - Wykrycie koloru zielonego przez kamerę
Rys.4.1 - Platforma przemieszczająca się w środku ciała robota
Rys.4.2 - Konstrukcja platform
Rys.4.3 - Konstrukcja górnej platformy wraz z magnesami
Rys.4.4 - Piramida testów

WYKAZ TABEL

Tabela 2.1 - Porównanie płyt Raspberry PI 3 z Arduino UNO