



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA



Projekt:

Zdalnie sterowany robot mobilny, wykorzystujący technologie Bluetooth.

Wykonał:

Dzieciuch Hubert

Spis treści

Wstęp	3
Opis użytych rozwiązań technicznych – główne użyte komponenty	4
Mikrokontroler – Bluepill	4
Moduł Bluetooth HC-06	5
Listewka HY-S301	6
Dwukanałowy sterownik silników – TB6612FNG	7
Silniki	8
Źródło zasilania:	8
Wykonanie robota	9
Schematy	9
Mikrokontroler	9
Moduł Bluetooth	9
Mostek H	10
Stabilizator napięcia	10
Źródło zasilania	11
Czujniki odbiciowe	11
Płytki PCB	11
Wykonany robot	13
Utworzona aplikacja	14
Główne elementy projektu	14
Sterowanie robotem	14
Odczyt czujników	15
Aplikacja	15
Programy Wykonawcze	17
Aplikacja Desktopowa	17
Program Mikrokontrolera	18
Wyniki testów	18

Wstęp

Wykonany projekt jest to wynikiem realizacji zadania projektowego z przedmiotu Technologie Bezprzewodowe. Wybór projektu był rozpatrywany na podstawie wielu kryteriów, jednak głównymi było jak największe wykorzystanie tematu przewodniego przedmiotu oraz wykonanie odpowiedniego modelu pozwalającego na implementację danych założeń.

Realizacja projektu polega na wykonaniu autorskiego robota mobilnego. Roboty mobilne tego typu są to bardzo popularne konstrukcje wśród studentów, gdyż dzięki nim mogą poszerzać swoją wiedzę w zakresie elektroniki oraz programowania. Wykonany robot w wielu aspektach przypomina robota typu „Line Follower”, który jest zarówno konkurencją na niemal każdych zawodach robotów. Wykonany robot jednak posiada inne funkcjonalności. Po połączeniu się z komputerem będzie sterowany przy użyciu klawiatury, dodatkowo będzie zbierał informacje z czujników ukazując odczyt na aplikacji.

Sercem robota jest płytki rozwojowa Bluepill. Robot zawiera moduł Bluetooth HC-06, dzięki któremu komunikuje się z komputerem. Całość została osadzona na płytce PCB.

Zakres prac jakie zostały realizowane w celu wykonania projektu:

- zaprojektowanie płytki PCB
- wytrawienie płytki
- wykonanie modelu robota
- napisanie programu dla mikrokontrolera
- napisanie programu aplikacji
- testowanie oraz dostrajanie programów w celu najlepszej kompatybilności

Opis użytych rozwiązań technicznych – główne użyte komponenty

Mikrokontroler – Bluepill



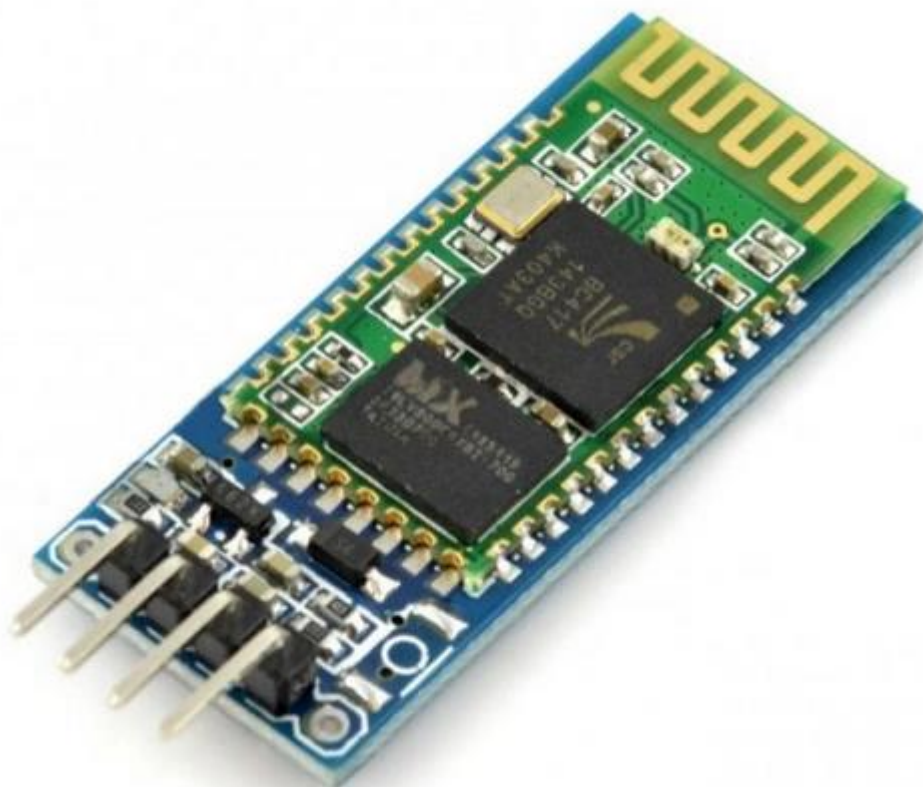
Bluepill jest to właściwie zestaw ewaluacyjny wyposażony w mikrokontroler STM32F103C8T6.

Specyfikacja mikrokontrolera STM32F103C8T6:

- Rdzeń: ARM Cortex-M3 CPU
- Częstotliwość 72Mhz stabilizowana kwarem 8MHz
- Pamięć Flash 64K
- Pamięć RAM 20K
- Zasilanie 3.3V

Jest to popularny zestaw z powodu małych gabarytów oraz stosunkowo do nich prędkości działania, gdyż częstotliwość taktowania 72MHz pozostawia z tyłu Arduino. Wszystko to pieczętuje możliwość programowania w środowisku ArduinoIDE, co znacznie ułatwia pracę z tym zestawem. Dodatkowy atut to bogate wyposażenie, zestaw daje możliwość użycia: 37 wyjść/wejść cyfrowych, 10 analogowych (o rozdzielczości 12 bitów), 15 PWM, oraz komunikacje I2C, SPI, UART, CAN oraz USB. Oczywiście nie możemy używać wszystkiego jednocześnie, gdyż zestaw posiada jedynie 40 pinów wyjściowych.

Moduł Bluetooth HC-06

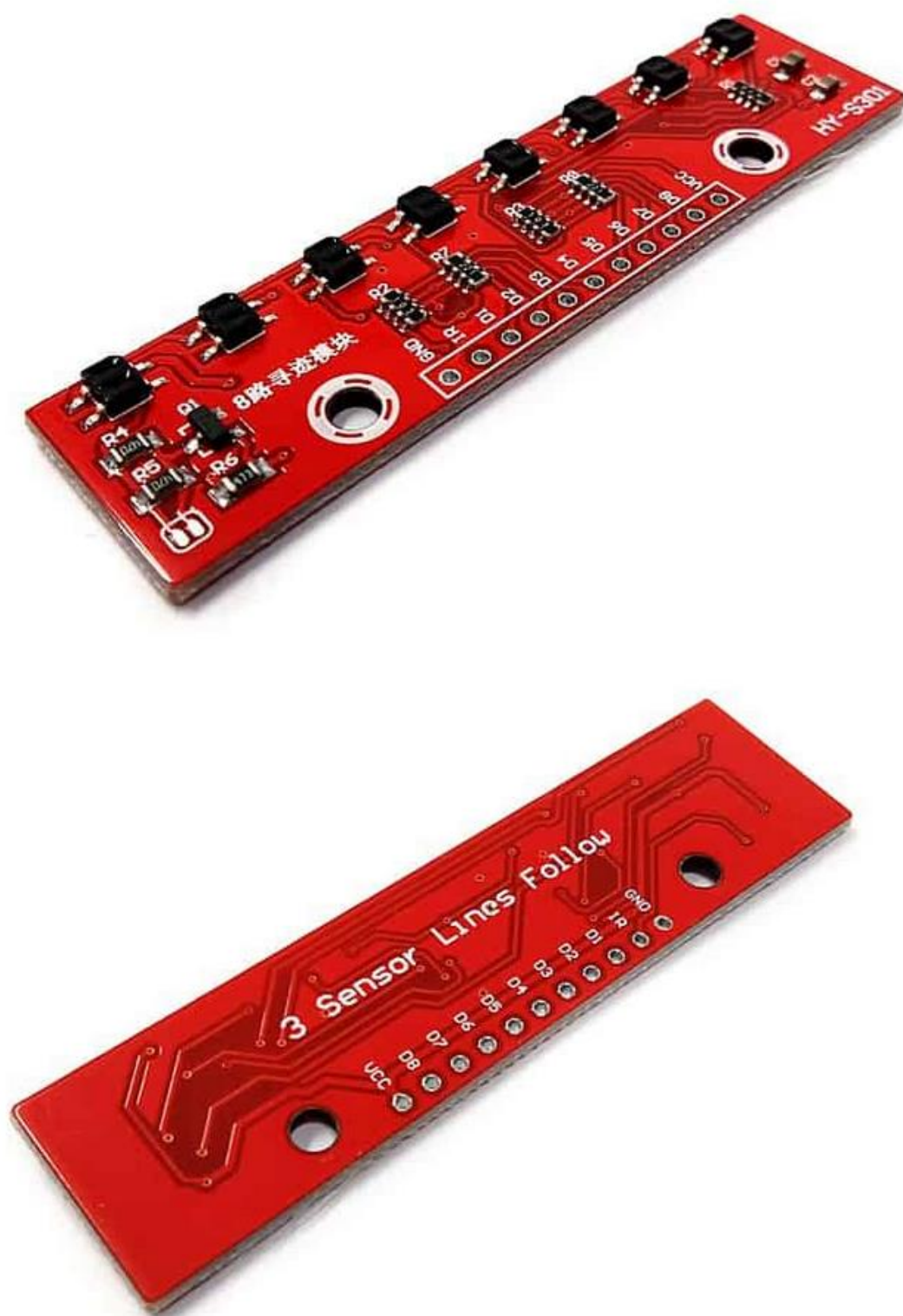


Specyfikacja modułu:

- Napięcie zasilania: 3,6 V do 6 V podłączane do pinu VCC
- Wyprowadzenia komunikacyjne pracują z napięciem 3,3 V (tolerują 5 V)
- Pobór prądu: ok. 50 mA
- Klasa 2 - moc nadajnika maks. + 4 dBm
- Zasięg: do 10 m
- Standard: Bluetooth 2.0 + EDR
- Profil SPP z możliwością ustawień poprzez komendy AT
- Wersja oprogramowania: Linvor v1.8
- Komunikacja: UART (RX, TX)
- Wymiary: 37 x 16 mm

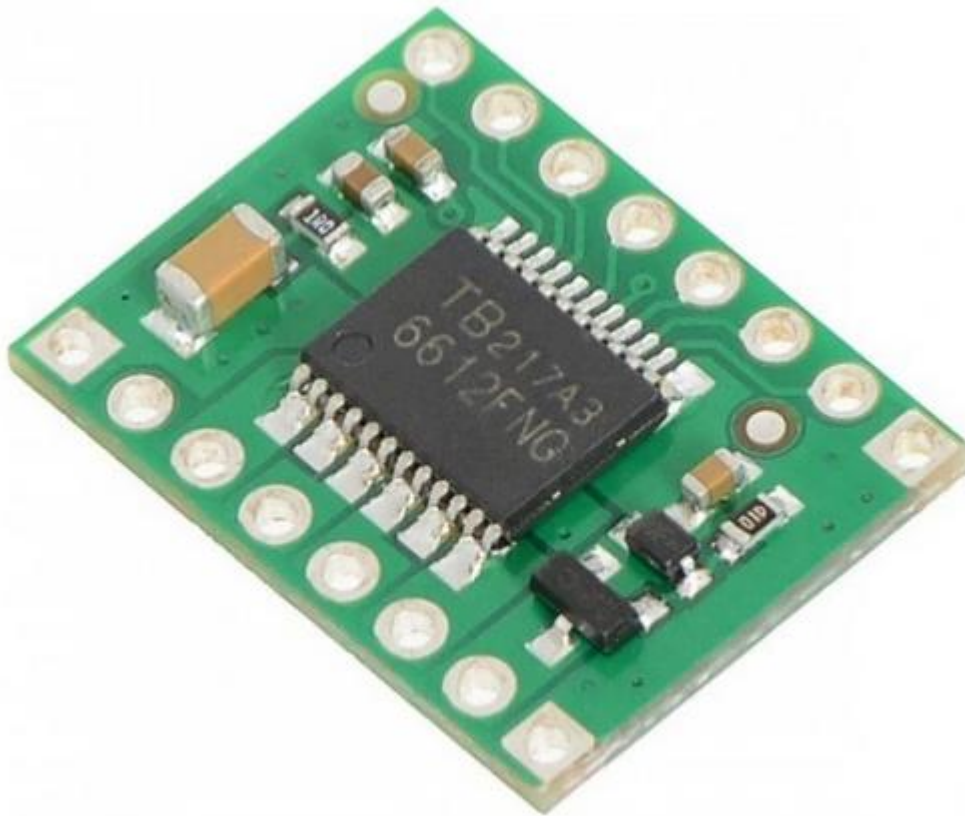
Moduł ten umożliwia połączenie dowolnego urządzenia z telefonem, smartfonem, tabletem lub laptopem. Przy użyciu komend AT możemy modyfikować jego ustawienia np. hasło lub prędkość przesyłania.

Listewka HY-S301

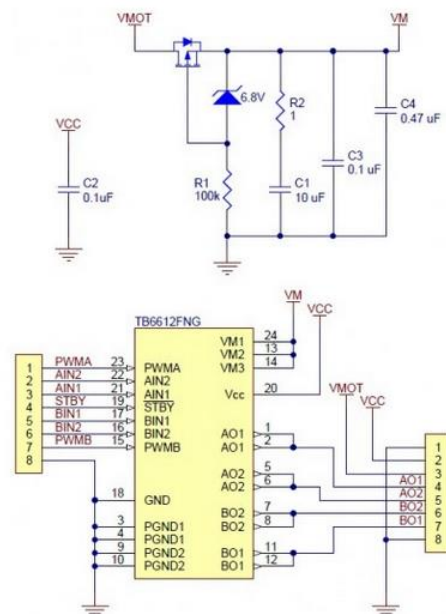
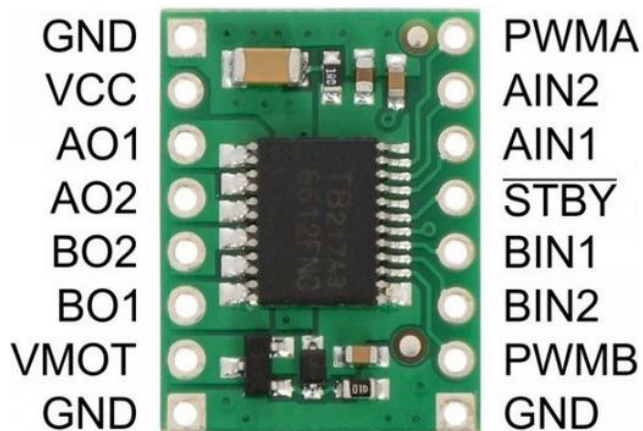


Moduł ten działa w oparciu o odbicie światła podczerwonego. Zasilany napięciem 3.3V-5V. Dzięki analizie odczytów 8 czujników, można rozczytać czy dany czujnik znajduje się właśnie nad czarną czy białą linią.

Dwukanałowy sterownik silników – TB6612FNG



Mostek H TB6612FNG jest wydajnym sterownikiem silników marki Pololu. Umieszczony został w obudowie SSOP24, gdzie znajduje się mostek oraz potrzebne do jego poprawnego działania komponenty pasywne. Mostek ten działa w zakresie napięć 4V-13.5V, a jego wydajność prądowa wynosi 1A. Poniżej zobaczyć możemy opis jego wyjść oraz schemat.



Silniki



W budowie użyte zostały dwa micro silniki N20-BT03 10:1.

Specyfikacja:

- Napięcie zasilania: 3V-12V
- Prędkość bez obciążenia(12V): 3000obr/min
- Moment obrotowy (12V): 0.3kg*cm (0,028 Nm)
- Przełożenie: 10:1
- Średnica wału: 3mm
- Wymiary korpusu: 24x10x12 mm
- Masa: 10g

Źródło zasilania:



Źródło zasilania to litowo-polimerowy akumulator marki Dualsky. Prąd rozładowania: ciągły 30 C (6,6 A), chwilowy do 50 C (11 A). Wymiary 39 x 18 x 13 mm. Masa 16 g.

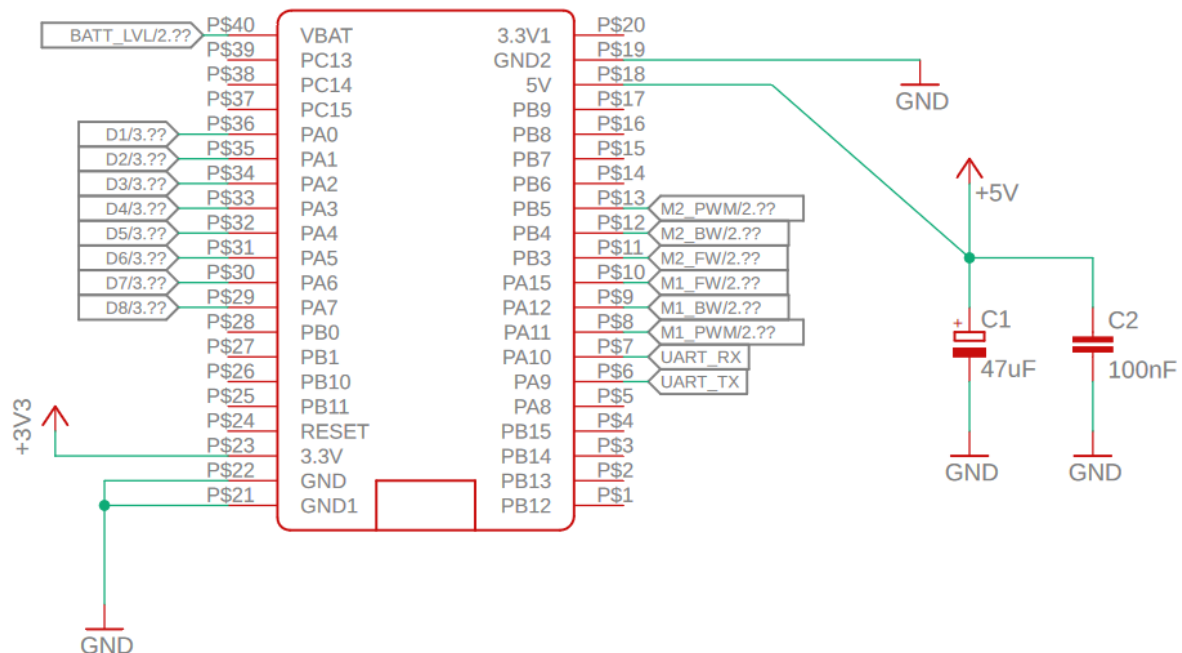
Wykonanie robota

Schematy

Poniżej zostaną ukazane kolejne fragmenty schematu połączeń na płytce PCB. Schematy zawierają opisy kolejnych połączeń oraz wyprowadzanych sygnałów z poszczególnych elementów. Użyte zostały również wymagane do odpowiedniego działania elementy pasywne: kondensatory oraz rezystory, których pojemności zostały zaznaczone na schemacie.

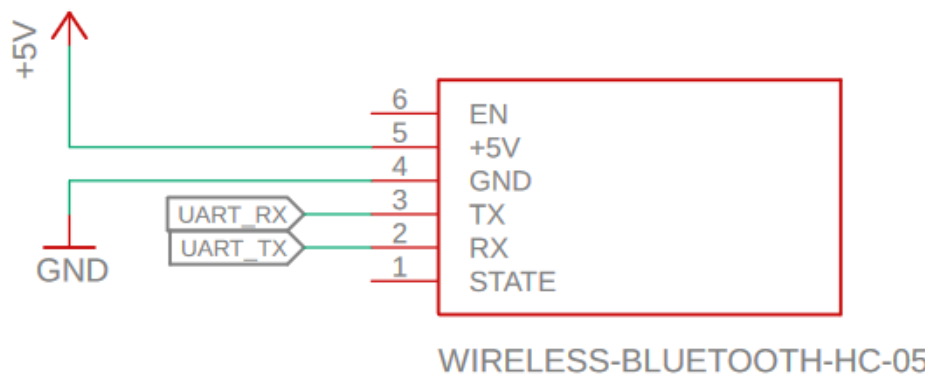
Mikrokontroler

- Do wejść analogowych PA0-PA7 zostały podpięte odczyty czujników z listewki.
- Do wyjść PA11-PB5 zostały podpięte dwa wyjścia PWM oraz 4 wyjścia cyfrowe.
- Mikrokontroler zasilany jest napięciem 5V.



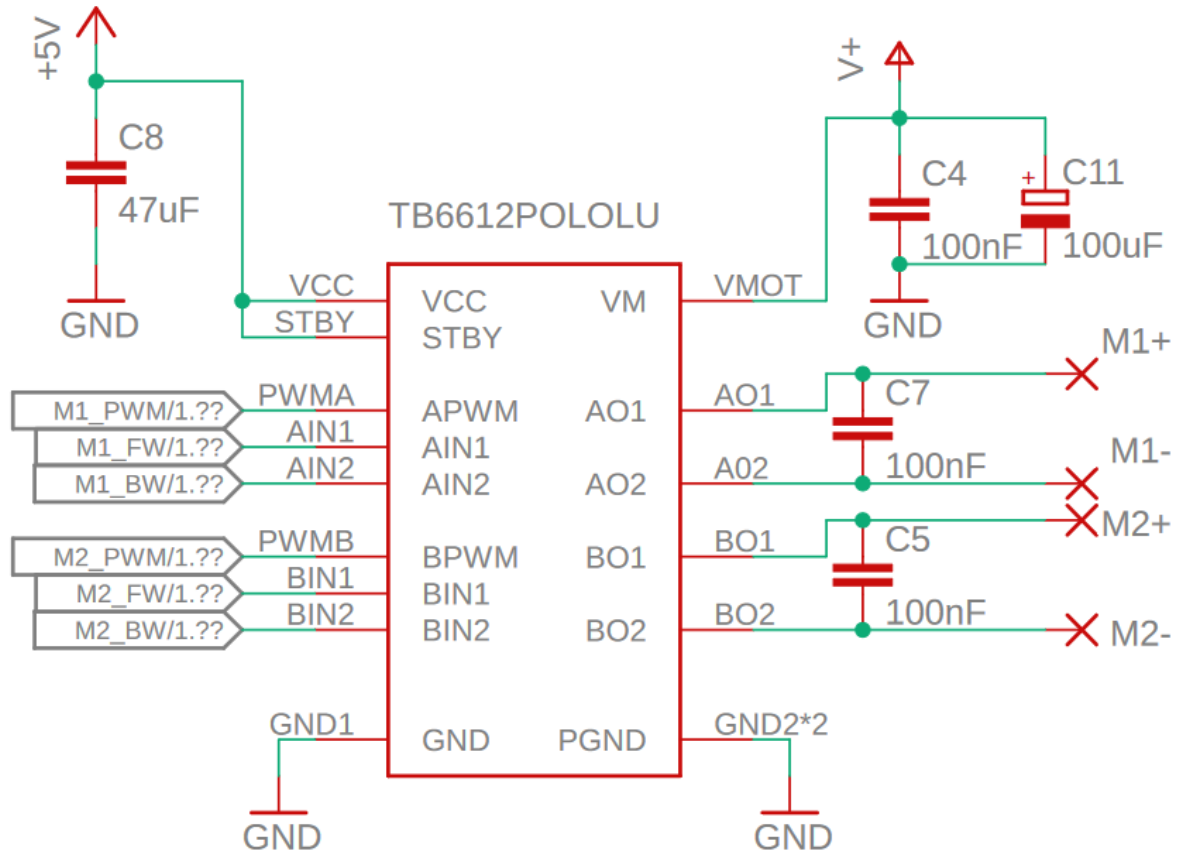
Moduł Bluetooth

Został on odpowiednio podpięty do zasilania 5V, GND oraz wyjść portu szeregowego w mikrokontrolerze.



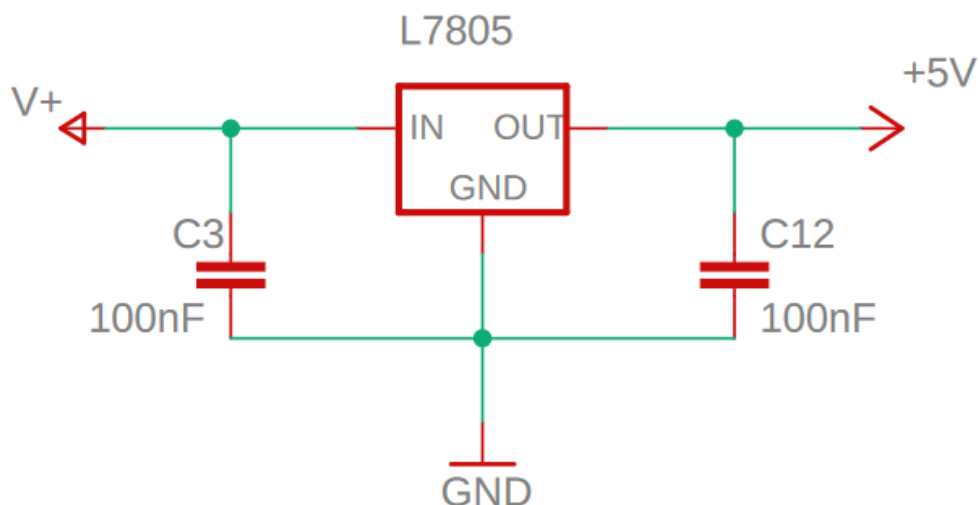
Mostek H

Do mostka zostało doprowadzone napięcie zasilające 7,4V bezpośrednio z baterii oraz 5V do sterowania logiką mostka. Zostały doprowadzone sygnały sterujące dla mostka z mikrokontrolera oraz sygnały wyjściowe dla silników.



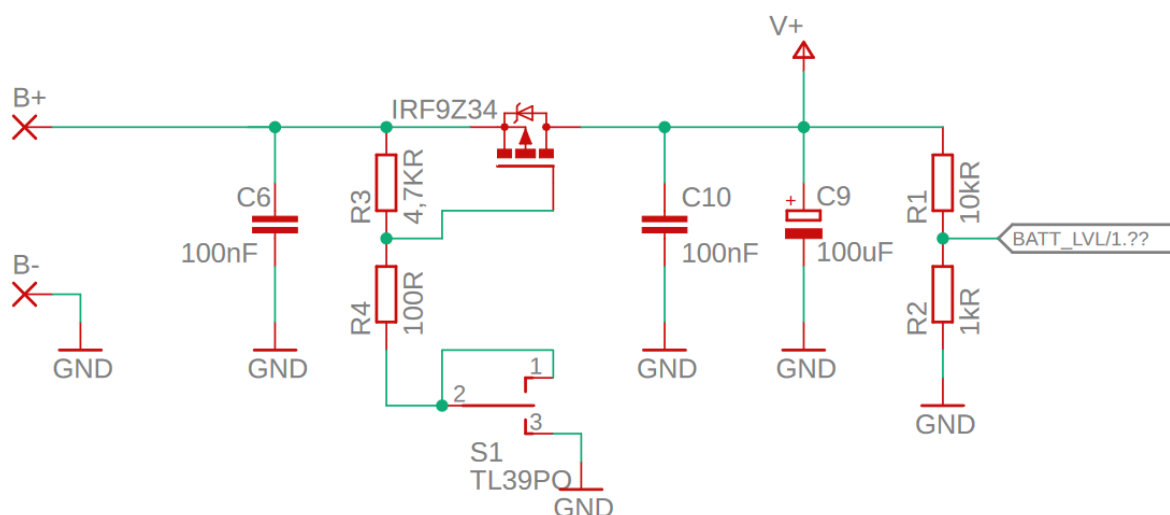
Stabilizator napięcia

Do ustabilizowania napięcia został użyty bardzo popularny stabilizator L7805. Jego zaletą jest oczywiście prostota oraz cena. Jednak zauważalną wadą jest osiągana wysoka temperatura podczas pracy robota.



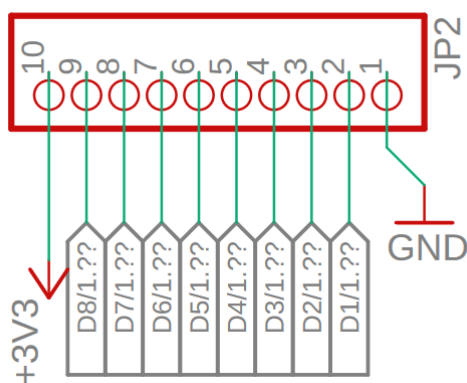
Źródło zasilania

Do zasilania standardowo służy dwa wyprowadzenia baterii + oraz -. Schemat zawiera też dzielnik napięcia, z którego wyprowadzony sygnał daje możliwość kontrolowania napięcia na baterii, jest to szczególnie przydatne przy dłuższej pracy robota. Kolejnym elementem na schemacie jest włącznik robota. Zastosowany został tutaj też tranzystor IRF9Z34 typu MOSFET, pozwala on nam na użycie włącznika o małej przepustowości prądowej. Dzięki takiemu zabiegowi możliwe jest sterowanie dużym prądem pochodzącym z baterii przy użyciu małego prądu wchodzącego na bramkę tranzystora.



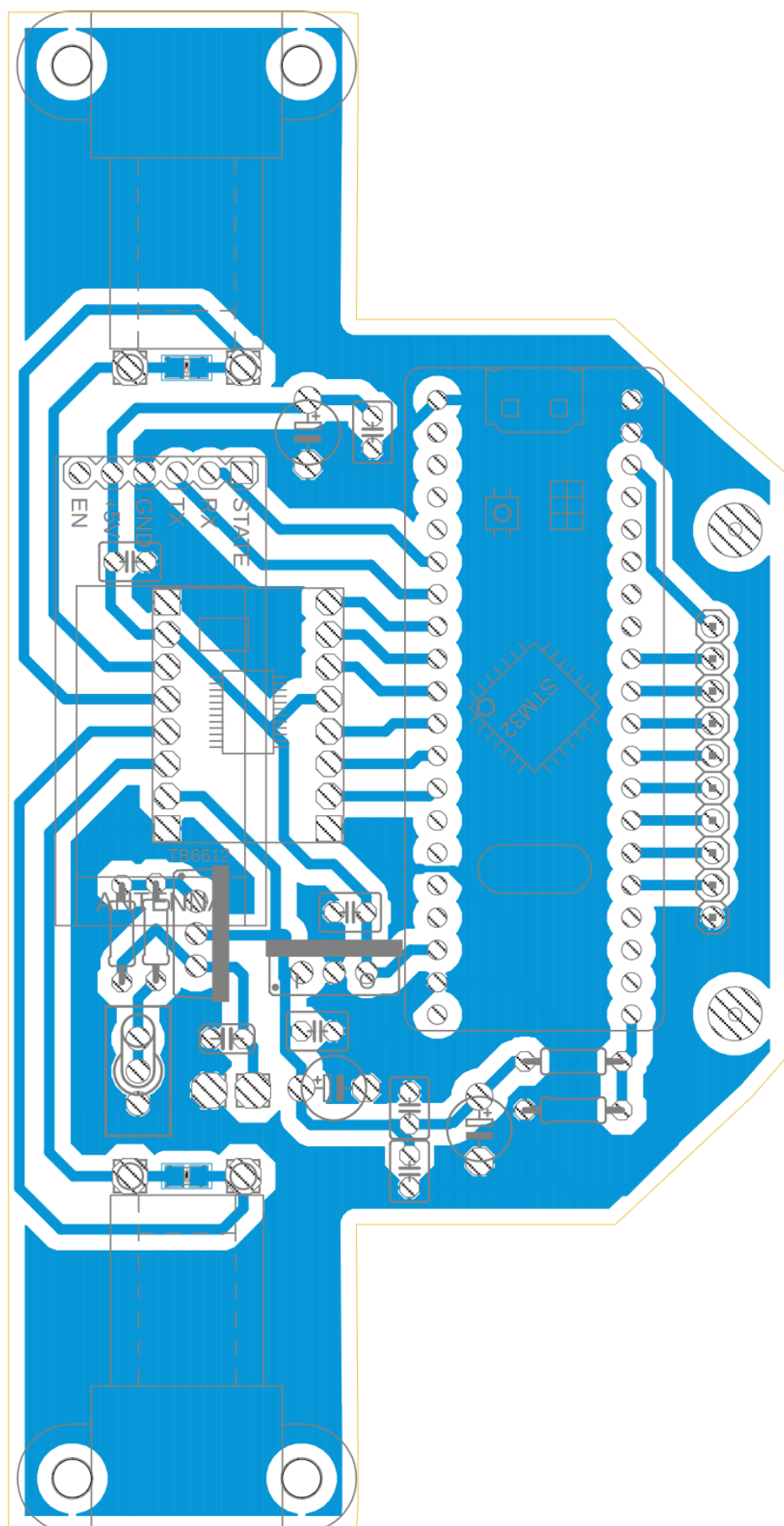
Czujniki odbiciowe

Listewka czujników odbiciowych posiada odpowiednie wyprowadzenia, więc pozostało jedynie doprowadzić do niej zasilanie oraz doprowadzić do mikrokontrolera odpowiednie sygnały z czujników.



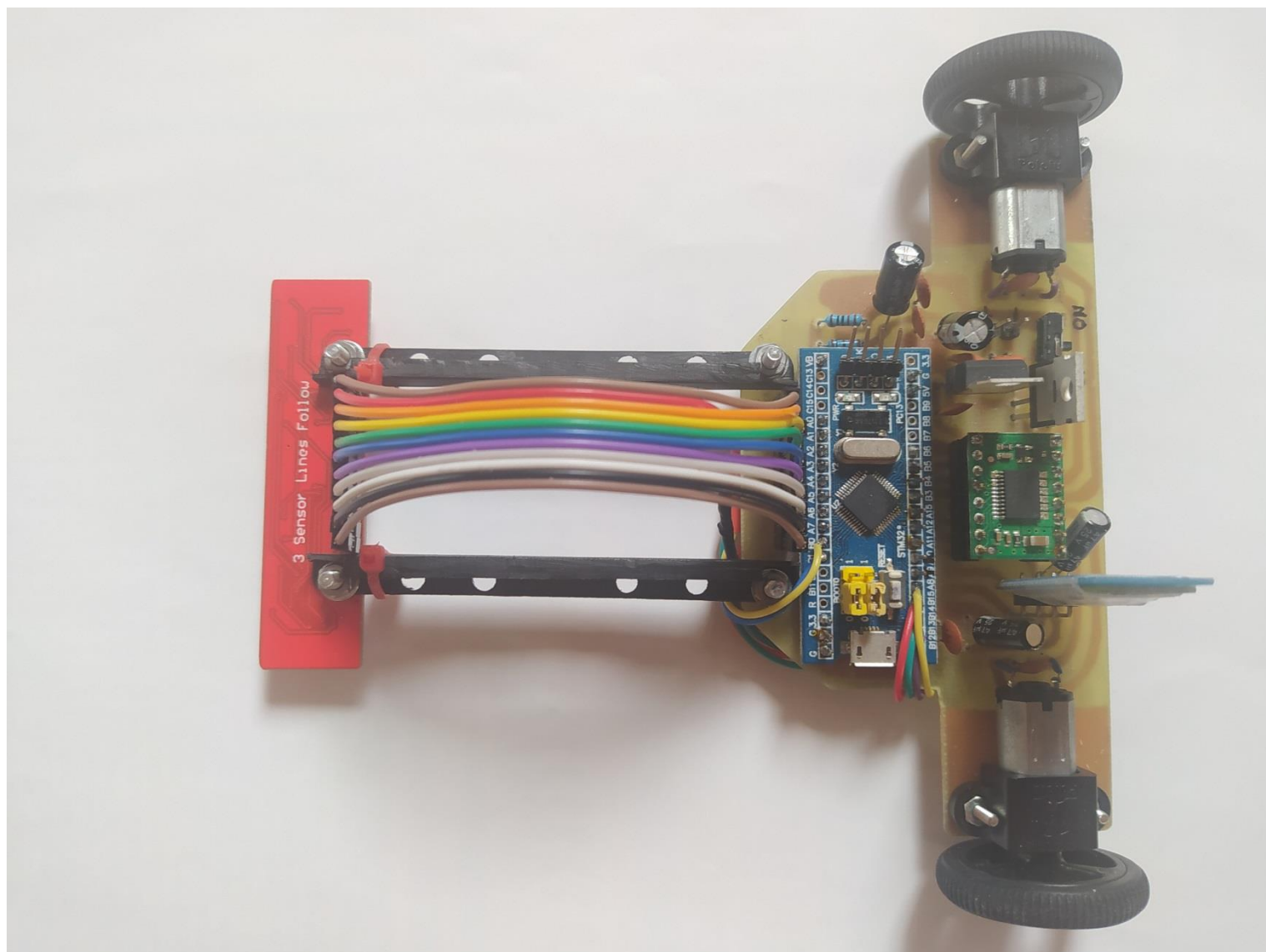
Płytką PCB

Zarówno płytkę jak i schemat zostały wykonane przy użyciu programu EAGLE. Powyższe schematy połączeń zostały przeniesione na model płytki drukowanej. Poniżej ukazany schemat pokazuje widok płytki wraz ze wszystkimi ścieżkami oraz użytymi elementami. Schemat został przeniesiony na rzeczywistą płytkę PCB za pomocą metody termotransferu.



Wykonany robot

Wykonany model robota prezentuje się następująco.



Utworzona aplikacja

Temat przewodni, czyli Technologia Bezprzewodowa została w danym projekcie wykorzystana do komunikacji pomiędzy robotem a laptopem. Komunikacja jest w tym przypadku dwustronna.

Główne elementy projektu

Sterowanie robotem

Użytkownik steruje robotem poprzez aplikację desktopową napisaną w języku Python. Sterowanie odbywa się poprzez naciśnięcie odpowiednich przycisków na klawiaturze w sposób ciągły, nie jest konieczne potwierdzenie komunikatu. Jest to sposób porównywalny do tego jaki występuje w grach komputerowych. Po naciśnięciu odpowiednich przycisków laptop wysyła do robota odpowiedni komunikat z informacją jak maysterować silniki.

Sterowanie odbywa się następująco:

- Silnik lewy:

Wciśnięcie przycisku „t” – silnik przód

Wciśnięcie przycisku „g” – silnik tył

Brak wciśnięcia jakiegokolwiek przycisku, lub wciśnięcie ich jednocześnie skutkuje łagodnym hamowaniem silnika.

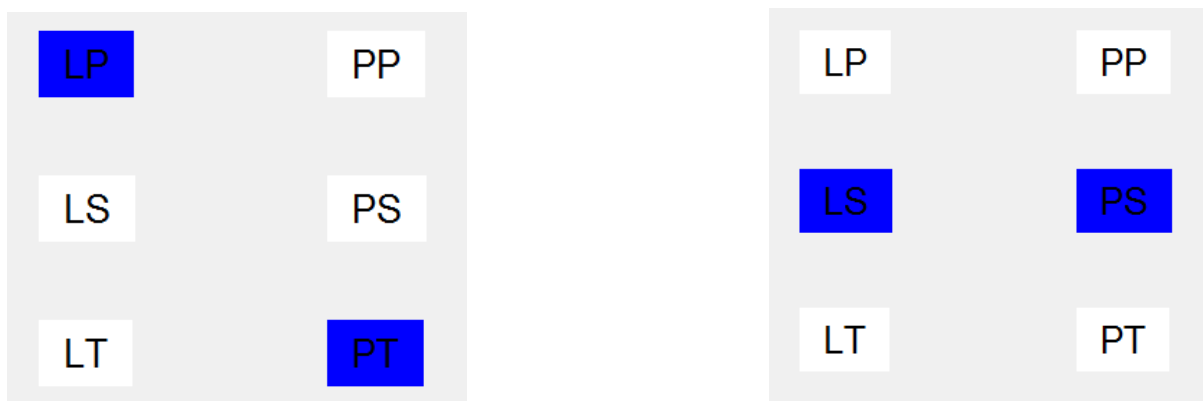
- Silnik prawy:

Wciśnięcie przycisku „o” – silnik przód

Wciśnięcie przycisku „k” – silnik tył

Brak wciśnięcia jakiegokolwiek przycisku, lub wciśnięcie ich jednocześnie skutkuje łagodnym hamowaniem silnika.

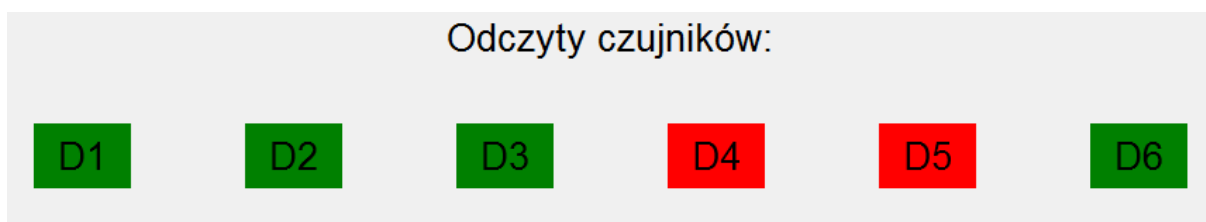
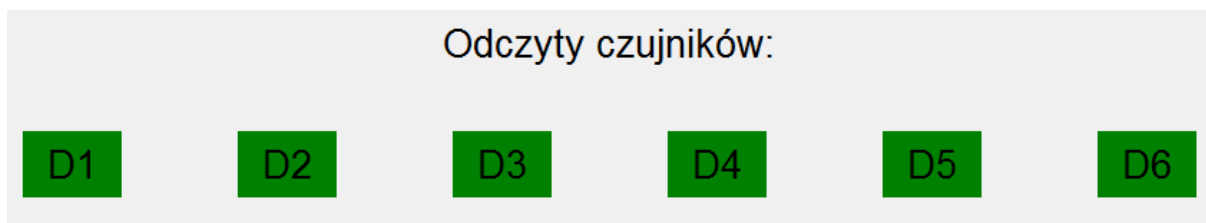
Dodatkowo na aplikacji zostaje zasygnalizowane w jaki sposób dany silnik jest sterowany:



Są to odpowiednio informacje: LP-Lewy Przód, LS-Lewy Stop, LT-Lewy Tył, PP-Prawy Przód, PS-Prawy Stop, PT-Prawy Tył.

Odczyt czujników

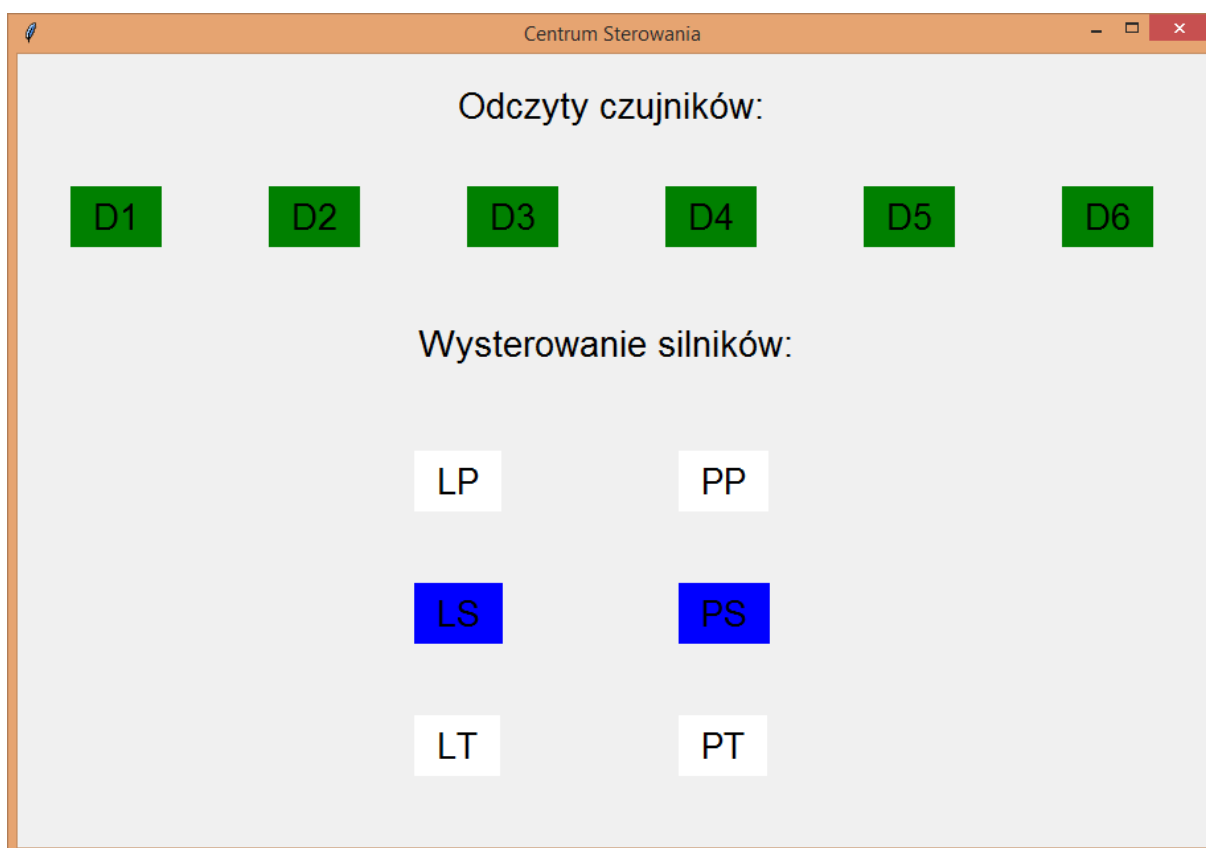
Odczyt czujników jest wysyłany przez mikrokontroler do laptopa, następnie odczyt jest zasygnalizowany na aplikacji desktopowej odpowiednim kolorem.

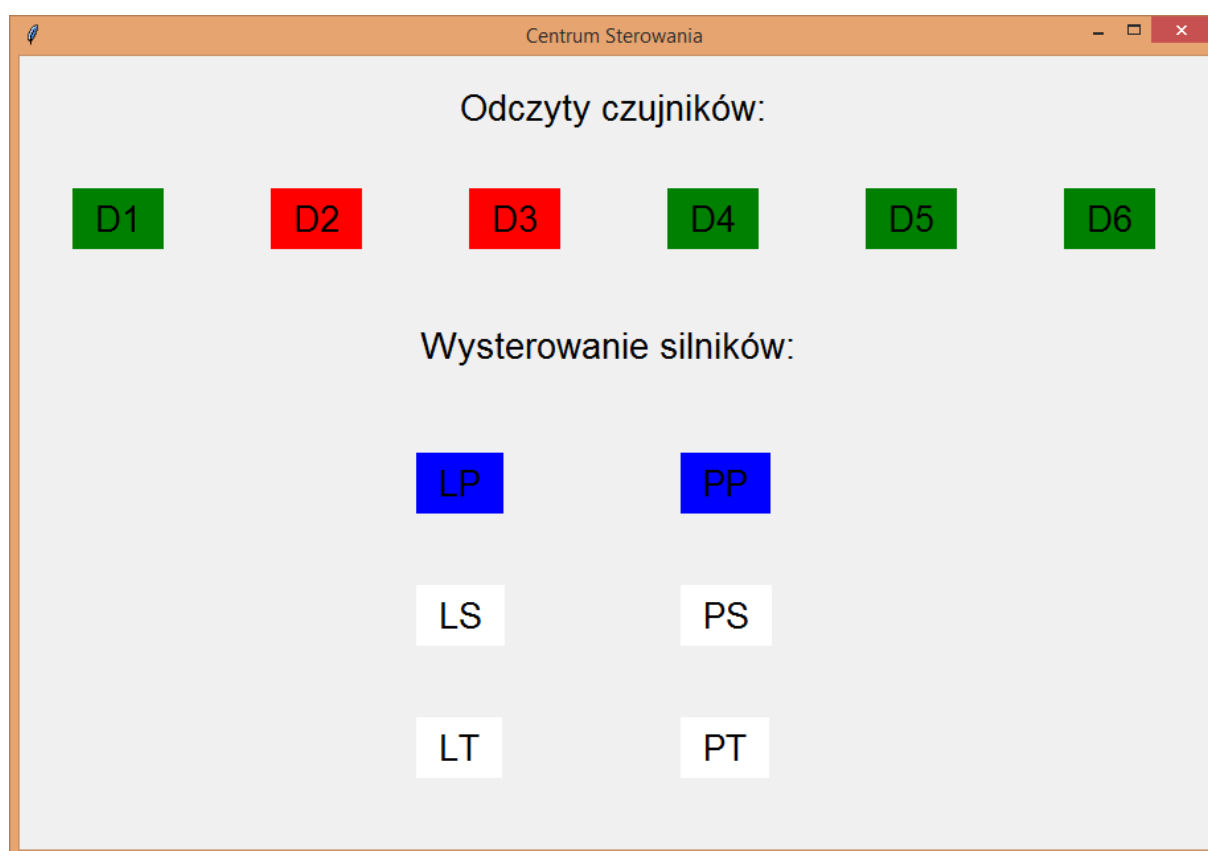
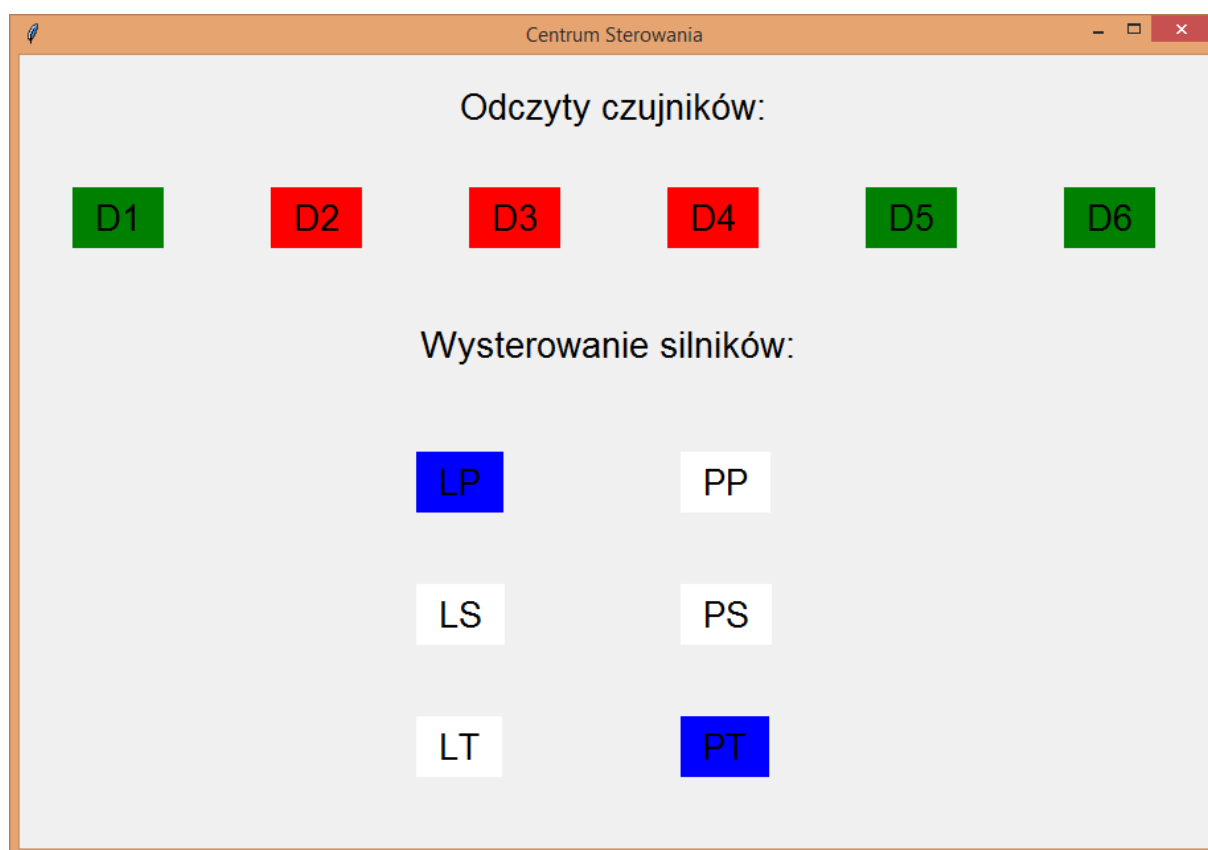


Jeżeli na danym czujniku świeci się kolor zielony to oznacza, że dany czujnik znajduje się nad białym podłożem. Czerwony kolor sygnalizuje, że dany czujnik znajduje się nad podłożem o kolorze czarnym.

Aplikacja

Całościowy widok aplikacji prezentuje się następująco:





Programy Wykonawcze

Wykonane zostały dwa programy. Pierwszy w języku Python, odpowiadający za działanie aplikacji. Drugi został napisany w języku C,C++.

Aplikacja Desktopowa

Aplikacja ta została napisana w języku Python. Do jej wykonania użyte zostały następujące biblioteki:

- Threading – która umożliwia utworzenie osobnych wątków w programie
- Tkinter – jest to jedna z najpopularniejszych bibliotek GUI
- Serial – biblioteka ta umożliwia połączenie się z portem szeregowym komputera
- Keyboard – dzięki użyciu tej biblioteki, możliwe było łatwe sprawdzanie czy klawisz odpowiadający za sterowanie jest wciśnięty.
- Time – ta biblioteka umożliwiła uśpienie danego wątku na zadany okres czasu

Poniżej opisane są najważniejsze aspekty działania programu:

- Pierwszym jest fragment generujący GUI, czyli graficzny interfejs użytkownika na którym wyświetlane są odczyty czujników oraz sygnały sterujące silnikami. Do wytworzenia GUI użyta została biblioteka tkinter.
- Drugi element to funkcja wysyłająca sygnały do sterowania silnikami. Funkcja za pomocą biblioteki keyboard czytuje czy dany przycisk na klawiaturze jest wciśnięty, a następnie wysyła odpowiedni komunikat do mikrokontrolera.
- Trzeci element to funkcja odbierająca dane o czujnikach z mikrokontrolera oraz wywołanie odpowiedniej funkcji aktualizującej dane na interfejsie graficznym.
- Dodatkowym elementem wysyłanym przez mikrokontroler jest informacja o wysterowaniu czujników. Jest to jeden znak wcześniej zakodowany w mikrokontrolerze. Dzięki temu aplikacja sterująca sprawdza jak są aktualnie wysterowane silniki robota. Aplikacja wysyła więc do robota komunikat tylko w razie gdy będzie dokonywana zmiana wysterowania, w czasie gdy komunikat się powtarza aplikacja wstrzymuje się z ponownym jego wysłaniem.
- Ostatni element to wywołanie dwóch poprzednich funkcji do pracy na osobnych wątkach programu. Do tego wykorzystana została biblioteka Threading. Dzięki temu wysyłanie, odbieranie oraz pętla GUI działają na osobnych wątkach. Komunikacja pomiędzy wątkami odbywa się za pomocą aktualizacji zmiennych globalnych występujących równocześnie w dwóch wątkach. Wątki odpowiadające za komunikację po wysłaniu lub odebraniu wiadomości wywołują funkcje, która aktualizuje wartości w interfejsie.

Program Mikokontrolera

Został on napisany w środowisku ArduinoIDE. Środowisko zostało dodatkowo zmodyfikowane tak, aby była możliwość wgrywania przez niego program na używany w projekcie mikrokontroler.

Główne elementy programu mikrokontrolera:

- Inicjalizacja zmiennych początkowych. Są to zmienne używane w programie, tablica przeznaczona do wysyłania, tablica spełniająca rolę bufora oraz zmienne potrzebne doysterowania mostka.
- Część odbierająca komunikaty z aplikacji. Komunikat szczytywany jest z portu szeregowego, do którego podpięty jest moduł bluetooth. Komunikat zapisywany jest do bufora, jeżeli suma kontrolna zawarta w komunikacie się zgadza to odpowiednie elementy tablicy są zapisywane do zmiennych które odpowiadają za sterowanie silnikiem.
- Część wysyłająca odczyty czujników wraz z sumą kontrolną. Dodatkowo wysyłana jest zakodowana informacja, która ukazuje jaki ostatni komunikat został wysłany do sterowania silnikami.

Dodatkowo pliki wykonawcze zawierające kody programów zawierają liczne komentarze ułatwiające zrozumienie działania programu.

Wyniki testów

Wykonane zostało kilkanaście testowych przejazdów podczas tworzenia projektu. Po każdym z nich można było zauważyć pewnie niedokładności i dokonać odpowiednich modyfikacji w kodzie programów.

Pierwszym większym błędem były problemy nawiązania komunikacji pomiędzy aplikacją a modułem Bluetooth. Zostało to rozwiązane wprowadzeniem czasu oczekiwania(500ms) po wykonaniu otwarcia prostu szeregowego.

Drugim większym błędem było konwertowanie odczytów czujników pochodzących z komunikatów jako typ „int”. Przez co, po wystąpieniu zakłóceń w komunikacji program aplikacji przerywał swoje działanie. Ta część programu została zmodyfikowana dodając do niej pętlę warunkową try: except w programie aplikacji.

Trzecim i największym problemem okazała się awaria jednego z czujników na płytce. Szczęśliwym trafem jest to czujnik skrajny, więc wyeliminowane zostało szczytywanie odczytów z dwóch skrajnych czujników zmniejszając tym ilość używanych czujników do 6.

Po zaimplementowaniu ostatecznego programu zostało wykonane kilka przejazdów. Wszystkie zakończyły się powodzeniem. Robot odpowiednio reaguje na sygnały sterujące z klawiatury, chociaż zdarza się że robot nie zdąży zareagować na szybkie impulsowe naciśnięcie przycisku. Pomimo to ogólne sterowanie robotem jest całkiem przyjemne. Odczyty czujników są również prawidłowo wyświetlane na interfejsie graficznym. Jednoczesne działanie czujników oraz sterowania też przebiegło pomyślnie. W finalnej wersji robot ani razu nie zerwał komunikacji z aplikacją i nie wystąpiły żadne błędy w działaniu programu, więc poprawienie kodu zadziałało odpowiednio.