Politechnika Warszawska

W Y D Z I A Ł  E L E K T R Y C Z N Y

Instytut Sterowania i Elektroniki Przemysłowej

# Praca dyplomowa inżynierska

na kierunku Automatyka i Robotyka Stosowana

w specjalności Automatyka

## Projekt i budowa robota mobilnego wyposażonego w kamerę

Bartłomiej Przygodzki

numer albumu 299214

promotor

dr inż. Maciej Sławiński

WARSZAWA 2021

# Projekt i budowa robota mobilnego wyposażonego w kamerę


## Streszczenie

Praca przedstawia proces budowy oraz implementację oprogramowania robota mobilnego wyposażonego w kamerę. Jej założeniem jest skonstruowanie zdalnie sterowanej jednostki umożliwiającej płynne przekazywanie obrazu. Zakres pracy wiąże się również z uruchomieniem i przetestowaniem wykonanej konstrukcji.

Analizując tematykę zagadnienia, dokonano zestawienia trzech robotów z kamerą. Do tego celu wybrano konstrukcje o zróżnicowanej budowie oraz przeznaczeniu. Z ich opisów zaczerpnięto zastosowania później wykorzystane we własnym robocie.

Wykonany pojazd został zaprojektowany tak, aby sprawdzał się w warunkach domowych. Pod tym kątem dokonano wyboru silników oraz platformy. W czasie projektowania zdecydowano się na użycie dwóch mikrokontrolerów, z powodów ograniczeń jednego z nich oraz z chęci upłynnienia obrazu kamery. Dokonano również opisu sposobu doboru sterownika silników, sonarów, źródła zasilania oraz stabilizatora napięcia. Projektując część elektroniczną, zadbano o odpowiednią temperaturę układu oraz o bezpieczeństwo w przypadku wystąpienia zwarć. Przedstawiono też schematy ideowe oraz blokowe połączeń elektrycznych w układzie. Umieszczono również opis dwóch własnoręcznie zaprojektowanych płytek PCB.

W ramach pracy dokonano udanej komunikacji robota z użytkownikiem za pomocą wi-fi. Napisano również stronę internetową służącą jako panel kontroli jazdy. Witryna wyświetla obraz z kamery robota oraz wyniki pomiarów odległości i napięcia. Zastosowano również komunikacje UART w celu przekazywania poleceń między mikrokontrolerami.

Aby zrealizować jazdę robota po linii prostej, dokonano porównania trzech sposobów regulacji. Były one zrealizowane na postawie regulatorów PI, które synchronizowały prace silników lub tego nie robiły. Napisano również funkcje zabezpieczające przed kolizją w czasie jazdy na podstawie pomiarów odległości wykonywanych za pomocą sonarów.

W czasie testów dokonano oceny otrzymywanego obrazu. Okazał on się płynny i wyraźny na tyle, by zidentyfikować co, znajduje się w otoczeniu robota. Za pomocą wyników testu jazdy po linii prostej wykonano zestawienie osiągnięć różnych sposobów sterowania. Udało się dzięki temu wytypować najlepszy z nich. Dokonano również sprawdzenia bezkolizyjności robota. Zastosowane sonary okazały się wystarczające, jednak znaleziono też wady użytego zastosowania. Można uznać, że całościowy projekt sprawdził się w swoim przeznaczeniu.



**Słowa kluczowe:** robot, ESP32, kamera, mikrokontroler, wi-fi

# Design and construction of a mobile robot equipped with a camera


## Abstract

The work presents the construction process and implementation of a mobile robot software equipped with a camera. Its assumption is to construct a remotely controlled unit enabling smooth image transmission. The scope of work is also related to the commissioning and testing of the constructed structure.

When analysing the subject of the issue, a comparison of three robots with a camera was made. For this purpose, structures of various construction and purpose were selected. Their descriptions are used to draw applications later used in my own work.

The completed vehicle was designed to work well at home. It was with this in mind that the engines and the platform were selected. During the design stage, it was decided to use two microcontrollers, due to the limitations of one of them and the desire to liquefy the camera image. The method of selecting a driver for motors, sonars, power source and voltage stabilizer has also been described. When designing the electronic part, care was taken to ensure the appropriate temperature of the system and safety in the event of short circuits. Schematic diagrams and block diagrams of electrical connections in the system are also presented. There is also a description of two hand-designed PCBs.

As part of the work, the robot successfully communicated with the user via wi-fi. A website was also written to serve as a driving control panel. It contains the image from the robot's camera as well as the results of distance and voltage measurements. UART's communications were also used to pass commands between the microcontrollers.

In order to drive the robot in a straight line, three methods of regulation were compared. They were made on the basis of PI regulators, which synchronized the work of the engines or did not do it. There are also written anti-collision functions while driving based on distance measurements made with the use of sonars.

During the tests, the obtained image was assessed. It turned out to be fluid and clear enough to identify what is in the robot's surroundings. Using the results of the straight-line driving test, a comparison of the achievements of various control methods was made. Thanks to this, it was possible to select the best of them. The robot's collision-free condition was also checked. The sonars used turned out to be sufficient, but there were also found disadvantages of the application used. The overall project turned out to be successful in its application.



**Keywords:** robot, ESP32, camera, microcontroller, wi-fi

Załącznik nr 1 do zarządzenia nr 109 /2021
Rektora PW z dnia 9 listopada 2021 r.

„załącznik nr 5 do
zarządzenia nr 42 /2020 Rektora PW

Politechnika
Warszawska

Politechnika Warszawska

Warszawa,19.01.2022
miejscowość i data

Bartłomiej Przygodzki
imię i nazwisko studenta

299214
numer albumu

Automatyka i Robotyka Stosowana
kierunek studiów

**OŚWIADCZENIE**

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.
Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2021 r., poz. 1062) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Bartłomiej Przygodzki
.....
czytelny podpis studenta ".

Załącznik nr 3 do zarządzenia nr 109 /2021
Rektora PW z dnia 9 listopada 2021 r.

„załącznik nr 9 do
zarządzenia nr 42 /2020 Rektora PW

Politechnika
Warszawska

**Politechnika Warszawska**

Warszawa,19.01.2022
miejscowość i data

Bartłomiej Przygodzki
imię i nazwisko studenta
299214
numer albumu
Automatyka i Robotyka Stosowana
Wydział i kierunek studiów

Oświadczenie studenta w przedmiocie udzielenia licencji

Politechnice Warszawskiej

Oświadczam, że jako autor/~~współautor~~* pracy dyplomowej pt. „Projekt i budowa robota mobilnego wyposażonego w kamerę" udzielam/~~nie udzielam~~* Politechnice Warszawskiej nieodpłatnej licencji na niewyłączne, nieograniczone w czasie, umieszczenie pracy dyplomowej w elektronicznych bazach danych oraz udostępnianie pracy dyplomowej w zamkniętym systemie bibliotecznym Politechniki Warszawskiej osobom zainteresowanym.

Licencja na udostępnienie pracy dyplomowej nie obejmuje wyrażenia zgody na wykorzystywanie pracy dyplomowej na żadnym innym polu eksploatacji, w szczególności kopiowania pracy dyplomowej w całości lub w części, utrwalania w innej formie czy zwielokrotniania.

Bartłomiej Przygodzki
.....
czytelny podpis studenta

* niepotrzebne skreślić ".

# Spis treści


**1 Wstęp** **15**
1.1 Wprowadzenie . . . . . 15
1.2 Cel pracy . . . . . 16

**2 Wybrane przykłady robotów mobilnych wyposażonych w kamerę** **19**
2.1 Mobilny robot poszukiwawczo-transportowy . . . . . 19
2.2 Sześcionożna maszyna krocząca . . . . . 21
2.3 Rasberry Pi G1 Tank – robot edukacyjny . . . . . 23

**3 Projekt i budowa robota mobilnego, posiadającego kamerę** **25**
3.1 Wymagania projektowe . . . . . 25
3.2 Ogólna budowa robota . . . . . 26
3.3 Wybór silników . . . . . 27
3.4 Użyte koła . . . . . 28
3.5 Platforma oraz elementy montażowe . . . . . 28

**4 Projekt i realizacja mikroprocesorowego układu sterującego pracą maszyny** **31**
4.1 Wybór mikroprocesora . . . . . 31
4.2 Wybór dodatkowego mikroprocesora . . . . . 33
4.3 Wybór kamery . . . . . 35
4.4 Wybór sterownika silników . . . . . 35
4.5 Wybór sensorów odległości . . . . . 36
4.6 Wybór źródła zasilania . . . . . 37
4.7 Wybór sposobu stabilizacji napięcia . . . . . 38
4.8 Układ pomiaru napięcia . . . . . 39
4.9 Dodatkowe elementy układu . . . . . 40
4.10 Schemat połączeń . . . . . 41
4.11 Projekt płytek . . . . . 42

**5 Implementacja oprogramowania urządzenia** **45**

5.1 Sposób implementacji programów . . . 45
5.2 Podział zadań między mikrokontrolery . . . 46
5.3 Obsługa kamery oraz transmisji obrazu . . . 46
5.4 Komunikacja UART . . . 48
5.5 Komunikacja wi-fi . . . 49
5.6 Działanie interfejsu . . . 51
5.7 Sterowanie ruchem . . . 52
5.7.1 Sterowanie bez użycia regulatora . . . 52
5.7.2 Sterowanie z użyciem regulatora PI . . . 53
5.7.3 Sterowanie z użyciem regulatora PI z synchronizacją kół . . . 55
5.7.4 Sterowanie z użyciem podwójnego regulatora PI z synchronizacją kół . 57
5.8 Obsługa enkoderów . . . 58
5.9 Pomiar napięcia . . . 58
5.10 Funkcje zabezpieczające . . . 59
5.10.1 Bezpieczne działanie panelu kontroli ruchu . . . 59
5.10.2 Pomiar odległości . . . 59
5.10.3 Wykrywanie utraty połączenia wi-fi . . . 60

**6 Uruchomienie i testy 61**
6.1 Ocena pojedynczej klatki obrazu . . . 62
6.2 Działanie efektów kamery . . . 62
6.3 Test jazdy po linii prostej . . . 63
6.4 Test bezkolizyjności . . . 65
6.5 Bezpieczeństwo układu elektrycznego . . . 65
6.6 Funkcjonalność robota . . . 66

**7 Uwagi i wnioski 67**

**Bibliografia 69**

**Wykaz skrótów i symboli 71**

**Spis rysunków 73**

**Spis tabel 75**

**Załącznik I Ideowy schemat połączeń elektrycznych 79**

**Załącznik II Schemat płytki głównej 81**

**Załącznik III Projekt płytki głównej 83**

**Załącznik IV Zdjęcie płytki głównej** **85**

**Załącznik V Schemat płytki z kamerą** **87**

**Załącznik VI Projekt płytki z kamerą** **89**

**Załącznik VII Zdjęcie płytki z kamerą** **91**

# Rozdział 1

# Wstęp

## 1.1 Wprowadzenie

Robot może być maszyną autonomiczną lub sterowaną zdalnie zdolną do wspomagania bądź realizacji zadań. Niektórymi z jego możliwości jest poruszanie się, odbieranie sygnałów, analizowanie ich oraz reagowanie na nie [14].

Słowo robot powstało w 1920 roku jeszcze przed skonstruowaniem rzeczywistych urządzeń. Zostało użyte w sztuce Karela Čapka pod tytułem „Uniwersalne roboty Rossuma" [15]. Robotami nazwane zostały tam istoty skonstruowane z kopii żywej tkanki. Ich zadaniem było wykonywanie ludzkiej pracy. Mimo zmiany sposobu produkcji współczesne roboty często realizują ten sam cel.

W 1928 r. zbudowano robota na zwór wyglądu ludzkiego zdolnego do poruszania głową, rękami oraz transmisji mowy ludzkiej. Działał on za pomocą kilku silników oraz głośnika. Służył do przekazywania głosu użytkownika znajdującego się za kurtyną w czasie prezentacji. Konstrukcja ta nie była mobilna, czyli nie mogła poruszać się w przestrzeni [1].

Przykładem historycznych robotów mobilnych były Elmer i Elisie, zaprojektowane przez Graya Waltera. Poruszały się one, reagując na bodziec świetlny. Mimo prostej budowy roboty te były autonomiczne, oznaczało to, że same podejmowały decyzje. Składały się z podstawowych układów analogowych naśladujących rzeczywiste neurony [11]. Konstrukcje te uzmysławiały jakie możliwości roboty będą miały w przyszłości.

Szerszy rozwój robotyki ułatwiło zaprojektowanie mikroprocesorów, czyli układów zdolnych do wykonywania listy instrukcji. Znalazły one zastosowanie w większości budowanych współcześnie robotów. Większa moc obliczeniowa umożliwiła zaprojektowanie maszyn zdolnych do bardziej zaawansowanych zadań.

Współczesne roboty mobilne wykorzystujemy do rozbrajania bomb, ratowania ocalałych, czy wspomagania żołnierzy. Niektóre z nich można spotkać w domach np. autonomiczny robot

sprzątający Roomba. Ich budowa oraz oprogramowanie znacznie różnią się w zależności od powierzonej funkcji [12].

Zróżnicowanie w robotach wynika między innymi z wyboru sensorów, czyli czujników wykrywających zmiany w otoczeniu. Wybór ich ściśle definiuje możliwości zaprojektowanej maszyny. Przykładami sensorów są czujniki stykowe wykrywające dotyk, czujniki Halla wykrywające pole magnetyczne lub kamera. W przypadku urządzeń zdalnie sterowanych mogą one pomagać w rozeznaniu się w otoczeniu robota.

W niniejszej pracy podjęto się zbudowania trójkowego robota mobilnego z kamerą. Został on zaprojektowany jako zdalnie sterowana jednostka, z możliwością przekazywania obrazu kierującemu. Robot sprawdza się jako pojazd, którym sterować można poza zasięgiem wzroku. Motywacją do wytworzenia konstrukcji była chęć zaprojektowania urządzenia zdolnego do obserwacji wnętrza domu i zachowania zwierząt podczas nieobecności właściciela.

## 1.2 Cel pracy

Celem pracy jest zaprojektowanie robota mobilnego wyposażonego w kamerę. Głównym założeniem jest, aby obraz przekazywany był czytelny oraz płynny. Ma on służyć użytkownikowi do orientacji w przestrzeni robota. Na podstawie otrzymywanego obrazu kierujący powinien być w stanie, określić co znajduje się przed pojazdem. Sam robot nie ma podejmować decyzji o kierunku jazdy, a jedynie przekazywać informacje z otoczenia.

Dodatkowymi założeniami jest napisanie strony z panelem kontrolnym dla użytkownika dostępnej na komputerze. Z jej poziomu możliwe ma być ustawianie prędkości oraz dostęp do wyników pomiarów wykonywanych w układzie robota takich jak napięcie, oraz odległość od przeszkód. Robot ma być bezpieczny dla siebie oraz obiektów w swoim otoczeniu, dlatego powinien posiadać zabezpieczenia przed kolizją. Ponadto powinien poruszać się po linii prostej z możliwie jak najmniejszym błędem.

W pierwszym rozdziale pracy znajdują się podstawowe informacje dotyczące robotów mobilnych. Wymieniono w nim kilka projektów zawierających kamerę. Przytoczone przykłady zawierają zróżnicowany sposób realizacji systemu wizyjnego. W opisie robotów skupiono się na wyborze kamery oraz jej obsłudze. Ponadto opisano niektóre rozwiązania często wybierane podczas konstruowania robotów.

Drugi rozdział prezentuje mechaniczną konstrukcję zbudowanego układu. Przedstawiono w nim ogólną budowę robota. Opisano sposób wyboru układu napędowego oraz platformy. Mieści się w nim również krótki opis montażu.

W trzecim rozdziale omówiono projekt i realizacje mikroprocesorowego układu sterującego pracą maszyny. Uzasadniono w nim wybór wszystkich elementów elektronicznych robota oraz

zaprezentowano sposób ich połączenia. Za pomocą schematu blokowego przedstawiono również relacje między sensorami oraz układem sterującym.

Czwarty rozdział przedstawia implementacje oprogramowania urządzenia. Pokazano w nim sposób działania komunikacji robota z użytkownikiem. Człon ten zawiera opis realizacji przyjętych założeń użytkowych w kodzie programu. Widnieje w nim również porównanie oraz analiza różnych metod sterowania silnikami robota.

W piątym rozdziale pokazano testy oraz wyniki możliwości robota. Zawiera on opis uruchomienia oraz ocenę użytkowania. Porównano w nim zastosowane sterowania za pomocą testów na rzeczywistym układzie oraz zaprezentowano obraz uzyskiwany z robota. Poddano w nim próbie również systemy zapobiegające uszkodzeniu pojazdu.

Ostatni rozdział poświęcony został końcowym uwagom oraz wnioskom. Przedstawiono w nim plusy i minusy zaprojektowanego urządzenia oraz możliwości dalszego rozwoju. Podsumowano również realizację zamierzonych celów.

# Rozdział 2

# Wybrane przykłady robotów mobilnych wyposażonych w kamerę

Konstrukcje robotów mobilnych cechują się niezwykłą różnorodnością. Biorąc pod uwagę sposób poruszania, można podzielić je na [9]:

- jeżdżące,
- chodzące,
- latające,
- pływające.

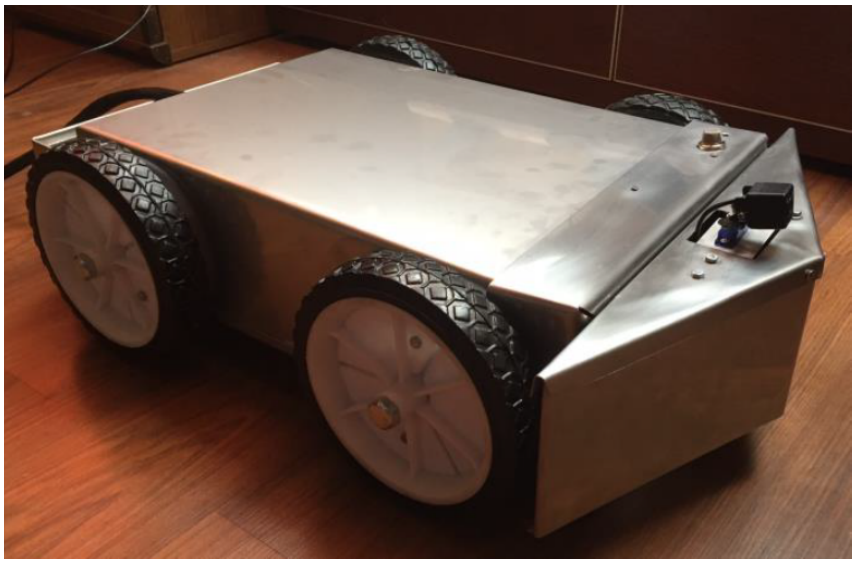
Grupy te dodatkowo dzielą się na podgrupy wynikające z ilości używanych kół czy nóg. Istnieją, tak więc roboty trzykołowe czy sześciokołowe albo jednonożne lub sześcionożne. Dostępnych konfiguracji jest wiele, dopuszczalne jest również ich łączenie, wyposażając robota w kończyny oraz koła.

Do wszystkich wymienionych konfiguracji możliwe jest dołączenie kamery. Jednak częściej wybierane są maszyny jedzące. Spowodowane jest to tym, że ich konstrukcja jest prostsza, a same są bardziej wydajne niż roboty chodzące. Niższa cena ich budowy dodatkowo zachęca projektantów do wyboru tego typu konfiguracji.

## 2.1 Mobilny robot poszukiwawczo-transportowy

Przykładem czterokołowego robota z kamerą jest projekt autorstwa dr inż. Przemysława Filipeka, dr inż. Tomasza Kamińskiego, inż. Kacpra Czuryło oraz Kamila Piwko. Zaprezentowany w artykule [10] robot został skonstruowany jako jednostka mobilna zdolna do pomocy w czasie misji ratunkowych. Zadaniem maszyny jest wspomaganie ratowników poprzez dostarczenie sprzętu poszkodowanym. Dodatkowo robot przekazuje informacje o otoczeniu dzięki czujnikom i kamerze.

Korpus robota widoczny na zdjęciu 1 został zaprojektowany z trwałych metali w celu zwiększenia wytrzymałości konstrukcji. Do tego celu projektanci wybrali stal kwasoodporną. Kształt przedniej części pojazdu umożliwia spychanie z drogi przeszkód. Waga robota osiąga wartość 12 kg. Dodatkowo może on przetransportować do 15 kg bagażu. Jego wymiary wynoszą 540x390x200 mm, zapewniając mu możliwość dostania się w miejsca ciężko dostępne dla ludzi.



**Rysunek 1.** Korpus robota poszukiwawczo-transportowego [10]

Z przodu konstrukcji znajduje się kamera umieszczona na ruchomej platformie. Platforma ta umożliwia obrót wokół osi pionowej oraz poziomej dając tym samym szeroki zakres widoczności bez potrzeby obracania pojazdem. Realizuje to dzięki dwóm serwonapędom. Jako kamerę projektanci wybrali model CMD-053IRN. Jest ona powszechnie stosowana w samochodach jako kamera cofania. Jej parametry zostały dobrane tak, aby umożliwić dobrą widoczność obrazu nawet przy słabym oświetleniu. Jest to możliwe dzięki czterem wbudowanym jasnym diodom. Kamera obsługiwana jest za pomocą kabla RCA. Maksymalny pobór prądu wynosi 150 mA. Twórcy robota zaznaczyli, że starali się zminimalizować zużycie energii w celu wydłużenia czasu pracy działania baterii.

Układ napędowy robota składa się z czterech oddzielnie napędzanych kół. Elementy zostały dobrane tak, aby robot miał możliwość przewiezienia dodatkowego ładunku. Do tego celu zastosowano silniki z przekładnią o napięciu znamionowym 12 V. Do ich obsługi użyto dwa podwójne sterowniki L298N. Sterowniki te wysyłają sygnał o modulowanej szerokości impulsu. Zapewnia to możliwość dostosowania wartości napięcia podawanego na silniki. Moduł L298N sterowany jest za pomocą sygnału z Arduino Nano.

Zastosowane rozwiązanie jest bardzo powszechne w dziedzinie robotyki. Sterowanie pracą silników wymaga wysokiej mocy, która mogłaby zniszczyć mikroprocesor. Z tego powodu wysyła się z niego sygnał o niewielkiej wartości prądu. Jest on podawany na oddzielny moduł sterownika silników, który otwiera i zamyka styki w mostku H. Czas trwania poszczególnych konfiguracji otwarcia i zamknięcia styków powoduje podanie napięcia o określonej wartości na silnik. Oczywiście do sterowania robotem nie jest wymagany mikrokontroler, jednak w bardziej złożonych projektach użycie go jest koniecznie. Przykładem robota bez mikrokontrolera może być line follower opisany w podanej książce [2].

Robot poszukiwawczo-transportowy z racji swojego przeznaczenia wyposażony jest w czujnik gazu oraz temperatury. Zostały one połączone do wcześniej wspomnianego mikrokontrolera. Czujniki wysyłają co jedną sekundę informacje o swoim stanie.

Oprogramowanie Arduino Nano zostało napisane w środowisku ArduinoIDE. Jest to aplikacja wykorzystywana do programowania płytek z rodziny Arduino oraz kilku innych kompatybilnych modeli. Język w niej wykorzystywany to najczęściej C i C++. W przedstawionym programie zaprogramowano sterowanie modułem L298N oraz czujnikami. Wartości z sensorów przekazane są do użytkownika za pomocą wyświetlacza LCD również połączonego z Arduino. Niestety robot nie posiada modułu komunikacji bezprzewodowej.

Panel sterujący z wyświetlaczem podłączony jest za pośrednictwem kabla. Na panelu możliwe jest sterowanie ruchem robota z pomocą joysticka. Położenie kamery regulowane jest poprzez potencjometry. Elementy elektryczne umieszczone zostały na uniwersalnej płytce drukowanej. Jej otwory są równomiernie ustawione w odległości 2,54 mm. Umożliwia to przylutowanie do niej większości typów elementów elektronicznych.

Przestawiony robot posiada raczej prostą konstrukcję, ale nadaje się jako podstawa do rozbudowy projektu. Jego napęd realizuje założenia przyjęte przez konstruktorów. Kamera na robocie dzięki wyborze odpowiednich parametrów oraz ruchomej platformie zapewniania dobrą widoczność. Niestety niemożliwe jest w aktualnym stanie wykonanie analizy lub przetwarzania obrazów w tak wyposażonym robocie. Zastosowana płytka Arduino Nano nie nadaje się do tego typu zadań z powodu małej mocy obliczeniowej. Dostęp do kamery odbywa się jedynie za pośrednictwem kabla RCA. Największym minusem wydaje się brak zdalnego sterowania oraz konieczność podążania za robotem. Chociaż takie rozwiązanie ma też swoje zalety, gdyż zapewnia niezawodność komunikacji.

## 2.2 Sześcionożna maszyna krocząca

Kolejnym przykładem robota z kamerą jest konstrukcja Jakuba Raka. W pracy jego autorstwa [13] przedstawiono platformę sterującą nadającą się do użycia w budowie sześcionożnej

maszyny kroczącej. Za zadania jej powierzone przyjęto pełną kontrolę nad napędem, obsługę obrazu z kamery oraz bezprzewodową komunikację z użytkownikiem.

W pierwszej kolejności projektant wybrał sposób komunikacji. Wybór padł na transmisje za pośrednictwem wi-fi. Takie rozwiązanie jest często spotykane w projektach z kamerą. Komunikacja ta cechuję się wysoką prędkością oraz szerokim dostępem modułów i oprogramowania. Alternatywny sposób przekazu informacji poprzez Bluetooth jest bardzo rzadko stosowany w projektach z przesyłaniem obrazu z powodu niższej prędkości oraz zasięgu sygnału. Zastosowany w maszynie moduł komunikacji wi-fi to Murata SN8000.

Kamera wybrana w przytoczonym projekcie to MT9D111. Posiada ona wbudowaną kompresje obrazu, co odciąża mikrokontroler. Komunikuje się ona za pośrednictwem interfejsu I2C w celu konfiguracji. Za to sam obraz jest przesyłany za pomocą 8-bitowej komunikacji szeregowej.

Przytoczona platforma robota umożliwia dołączenie osiemnastu serwonapędów oraz inercyjnej jednostki pomiarowej. Z tego powodu przy wyborze mikrokontrolera projektant kierował się spełnieniem potrzeby komunikacji ze wszystkimi wybranymi wcześniej elementami. Użyto tu mikrokontrolera STM32F4, który zapewnia odpowiednią ilość portów wejścia-wyjścia oraz interfejsy I2C, SDIO.

Na tym etapie warto zaznaczyć różnice między mikrokontrolerem, a mikroprocesorem. Mikrokontrolery są w stanie działać bez dołączania do nich dodatkowych modułów. Posiadają one wbudowaną pamięć, porty i tym podobne. Mikroprocesor zaś jest wyspecjalizowaną jednostką obliczeniową, która do działania wymaga podłączenia dodatkowych układów. Wewnątrz mikrokontrolera znajduje się mikroprocesor.

W ostatnim kroku dobrany został układ zasilania. Twórca zdecydował się w tym przypadku na akumulatory litowo-polimerowe. Jest to bardzo popularny wybór w mobilnych robotach. Jednak przy ich zastosowaniu należy zapoznać się z zasadami użytkowania baterii. W szczególności podczas ładowania, ponieważ pakiety te mogą eksplodować. Do źródła zasilania została podłączona podwójna przetwornica impulsowa dostarczająca napięcie stabilizowane do układu sterowania oraz silników.

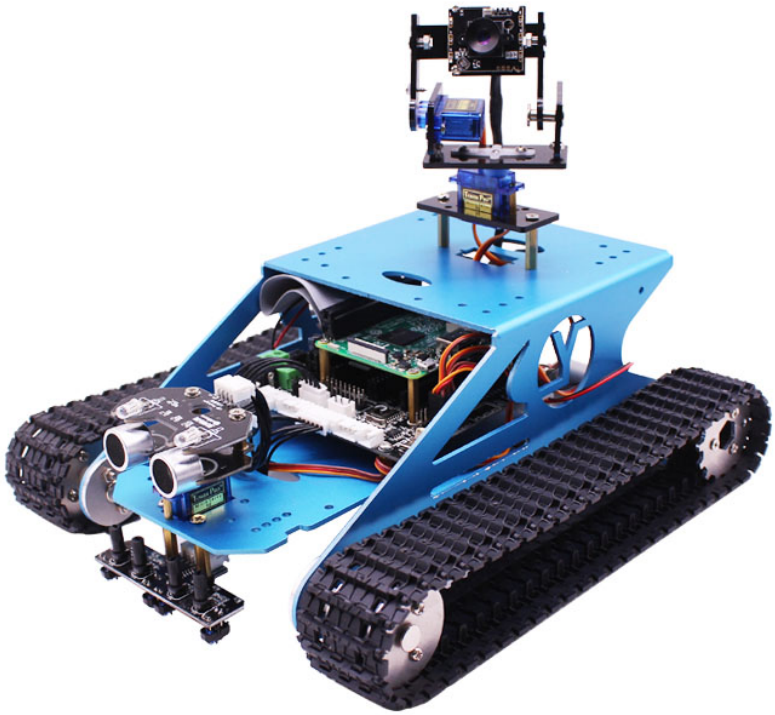
Komunikacja kamery z mikrokontroler odbywa się za pomocą wbudowanego w STM32F4 kontrolera DMA. Jest to urządzenie zdolne do bezpośredniego zapisu w pamięci z ominięciem procesora. Pozwala to na odciążenie głównej jednostki obliczeniowej i zapewnienie odpowiedniej prędkości transmisji. Kamera dodatkowo wymagała podłączenia do zewnętrznego zegara.

Robot posiada funkcje monitorowania stanu baterii zrealizowaną za pomocą pomiaru napięcia na pinie z przetwornikiem analogowo-cyfrowym w procesorze. Pomiar odbywa się na rezystorze w układzie dzielnika napięcia. Układ ten obniża napięcie, zabezpieczając w ten sposób procesor przed zniszczeniem.

Przytoczona maszyna pokazuje jedno z możliwych rozwiązań problemu komunikacji kamery z układem robota. Dzięki zastosowanemu rozwiązaniu możliwy jest dostęp do obrazu poprzez wi-fi. Projekt nie posiada niestety opisu części mechanicznej robota oraz bezprzewodowej transmisji obrazu.

## 2.3 Rasberry Pi G1 Tank – robot edukacyjny

G1 Tank to programowalny robot gąsienicowy firmy Yahboom. Został on przedstawiony na zdjęciu 2. Jako model edukacyjny posiada szereg możliwości w tym również te związane z obsługą kamery. Z tego powodu został on wybrany do opisu w tej pracy.



**Rysunek 2.** Robot G1 Tank
Źródło: http://www.yahboom.net/study/G1-T-PI

Korpus robota wykonany jest ze stopu aluminium. Pojazd napędzany jest przez dwa silniki umieszczone z tyłu. Obracają one układem gąsienic, pozwalając na jazdę maksymalnie pod

kątem 60 stopni. Konstrukcja ta sprawdza się w warunkach domowych nawet przy trudniejszych powierzchniach.

Główną częścią robota jest mikrokomputer Raspberry Pi 4B odpowiedzialny za sterowanie. Aby można było z niego skorzystać, należy wgrać w płytkę system operacyjny. Następnie zaprogramować polecenia we właściwie dowolnym języku. Twórcy robota proponują już gotowe rozwiązania w Pythonie. Mikrokomputer połączony jest z płytką modułową, która dostarcza mu energii oraz przekazuje informacje z sensorów. Na płytce tej znajduje się większość elektronicznych części robota.

Kamera umieszczona została na ruchomej platformie z dwoma serwomotorami. Wysyła ona obraz o rozdzielczości 480 pikseli. Podłączona jest do Raspberry Pi poprzez USB. Dodatkowo robota wyposażono w kolorowe diody, sonar do pomiaru odległości oraz moduł do jazdy wzdłuż wyznaczonej czarnej linii. Obsługiwane są one za pomocą wcześniej wspomnianej płytki.

Robot sterowany jest przez aplikacje w telefonie. Łączy się ona z Rasberry za pośrednictwem Bluetooth 4.0. W takiej konfiguracji możliwe jest sterowanie kierunkiem jazdy urządzenia, jego prędkością oraz kolorem diod. Na panelu istnieje również informacja o aktualnym położeniu robota, odległości od przeszkód oraz stanie baterii.

Podgląd z kamery możliwy jest po konfiguracji mikrokomputera tak, aby wysyłał obraz za pośrednictwem wi-fi. Po połączeniu telefonu z nową powstałą siecią udostępniony jest drugi tryb sterowania. Za jego pomocą obserwując obraz z kamery, możliwa jest kontrola jazdy robota oraz obrót serwomechanizmów.

W drugim trybie sterowania wykorzystującym kamerę możliwe jest wykonanie przetwarzania oraz analizy obrazu. W tym celu używany jest program Open Source Computer Vision Library znany również pod nazwą OpenCV. Do funkcji programu zaliczyć można wykonanie oraz zmianę histogramu, binaryzacje, dylatacje, erozje czy wiele innych. Prace szczególnie ułatwia wykorzystanie gotowych bibliotek. Za ich pomocą możliwe jest rozpoznawanie kolorów, twarzy czy kodów QR. Tak zaprogramowany robot może służyć na przykład jako pojazd śledzący człowieka.

Przedstawiony robot wykorzystuje mikroprocesor o dużej wydajności w celu obsługi kamery. Jest to rozwiązanie efektywne, lecz również kosztowne. Na rynku dostępne są procesory o mniejszej mocy obliczeniowej nadal zdolne do tego działania. G1 Tank jest produktem najbardziej zbliżonym przeznaczeniem do robota, który jest tematem tej pracy. Niektóre z jego funkcji zostało również zaimplementowane w zbudowanym robocie. Sposób transmisji obrazu poprzez wi-fi powtórzył się w stosunku do poprzednio zaprezentowanego przykładu. W tym pojeździe został on użyty, mimo że sterowanie nim odbywa się już za pomocą Bluetooth.

# Rozdział 3

# Projekt i budowa robota mobilnego, posiadającego kamerę

## 3.1 Wymagania projektowe

Wymagania projektowe robota wynikają ze środowiska oraz zadań, jakie przed nim postawiono. Budowany w ramach tej pracy pojazd ma poruszać się we wnętrzu domu. Z tego powodu nie musi on być przystosowany do pokonywania wysokich przeszkód ani trudnych terenów.

Jednym z przyjętych wymagań konstrukcyjnych jest nieduża waga. Robot ma być lekki, dzięki czemu możliwe będzie zastosowanie silników o mniejszym momencie obrotowym. Z tego powodu należy użyć platformy wykonanej z lekkich materiałów. Nie powinien to być metal. Robot nie musi też posiadać zamkniętej obudowy, jednak jego elementy nie powinny wystawać poza podstawę.

Konfiguracja robota powinna umożliwić jazdę z użyciem niedużej ilości kół napędowych. Zapewni to mniejszą cenę całego projektu. Zdecydowano się na konfigurację trzykołową, w której skład wchodzą dwa koła napędowe oraz jedno podporowe. Koszt budowy robota powinien być niewysoki. Dlatego jego elementy konstrukcyjne powinny składać się z łatwo dostępnych uniwersalnych części oraz modułów.

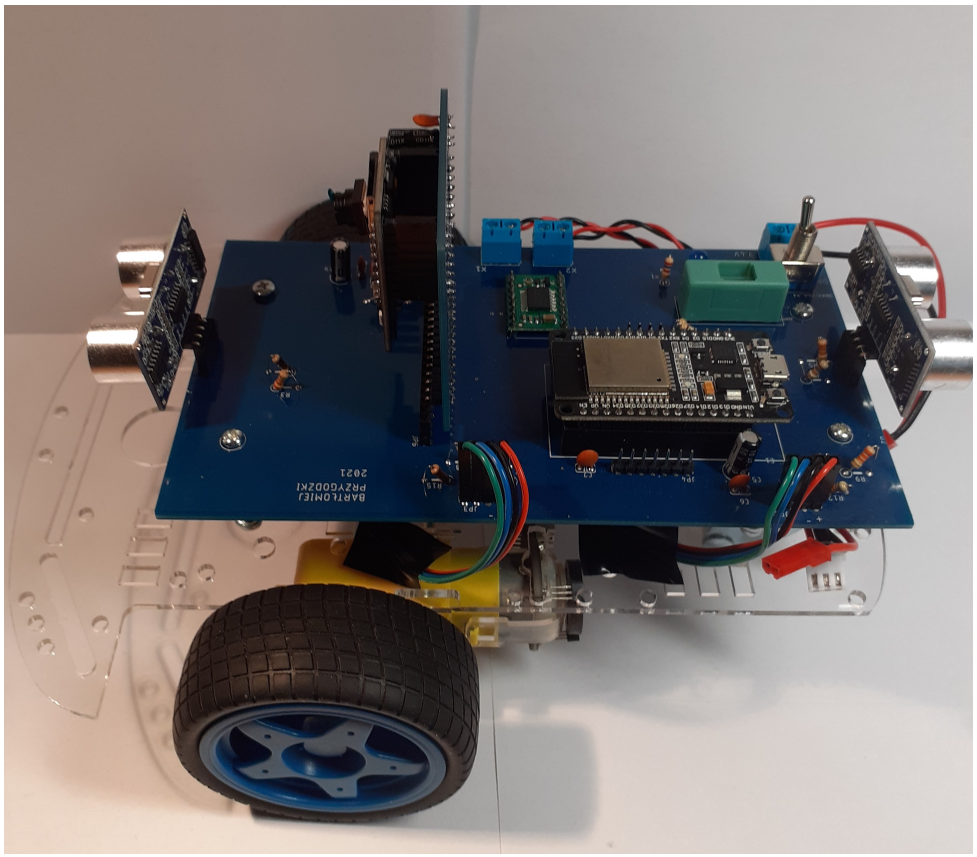
Cała konstrukcja musi być stabilna. Nie może się ona przewracać w czasie nagłego zatrzymania. W tym celu należałoby zadbać o odpowiednie wyważenie pojazdu.

Głównym elementem robota jest kamera. Powinna się ona znajdować w miejscu umożliwiającym dobry podgląd na przestrzeń przed pojazdem. Ponadto należałoby zaprojektować własne płytki PCB w celu zmniejszenia liczby przewodów użytych przy budowie. Na samych płytka powinny znajdować się elementy odpowiedzialne za sterowanie robotem (mikrokontrolery, sterownik silników), jego uruchomienie (włącznik) oraz bezpieczeństwo (sonary, bezpiecznik).

## 3.2 Ogólna budowa robota

Pierwszym etapem budowy robota był wybór podstawy. Analizując możliwości, podjęto decyzje o kupnie uniwersalnej platformy dla robotów trójkołowych. Zakupiony element wykonany jest z plexiglasu. Posiada on wiele gotowych już otworów montażowych. Jednak w celu dołączenia wybranych później silników w platformie wywiercono dodatkowe otwory.

Następnie dokonano wyboru napędu. Zdecydowano się na moduł z przekładnią oraz enkoderem. W późniejszym kroku, wybrano i przetestowaniu wszystkie elementy elektroniczne. Na ich podstawie stworzono projekty płytek pcb. Po złożeniu wszystkich elementów otrzymano robota przedstawionego na rysunku 3.




**Rysunek 3.** Zaprojektowany robot z kamerą

Zbudowana maszyna posiada dwa koła napędowe z przodu oraz jedno podporowe z tyłu. Tylne koło dzięki łożysku może obracać się w dowolnym kierunku. W skład pojazdu wchodzą dwie płytki pcb. Pionowa posiadająca kamerę i pozioma z układem, sterowania ruchem. Robot wyposażony jest w dwa sonary mierzące odległość od przeszkód. Całkowite wymiary konstrukcji wynoszą 21x16x12 cm.

Możliwe jest odłączenia drugiej płytki, aby otrzymać robota sprawnego, lecz bez funkcji przekazywania obrazu. Płytki posiadają elementy sprzyjające obsłudze takie jak wyłącznik czy dioda sygnalizująca zasilanie robota. Elementy na płytce oraz bateria ustawione są tak, aby dociążyć tył pojazdu. Pozwala to na przesunięcie środka ciężkości, co zapewnia większą stabilność w czasie hamowania.

## 3.3 Wybór silników

Wybierając układ napędowy, można natrafić przede wszystkim na silniki prądu stałego. Dzielą się one na modele z przekładnią i bez. Przekładnia to zestaw zębatek dołączonych do wału silnika, których działanie zmniejsza prędkość obrotową, dodając przy tym moment. Do budowy robota poszukiwano modeli wyposażonych w przekładnie, ponieważ niewymagana jest praca silników z dużą prędkością obrotową.

Dodatkowo wybierając napęd, poszukiwano modułu z dołączonym enkoderem, ma on posłużyć do obliczania prędkości obrotowej kół i jej regulacji. Układ napędowy stanowi znaczną część ceny robota. Dlatego brano pod uwagę elementy o możliwie najniższej cenie w celu zmniejszenia kosztu budowy.

Ostatecznie wybrano silnik FIT0450 w module z przekładnią oraz enkoderem. Spełnia on wszystkie wymienione kryteria. Opis jego parametrów znajduje się w tabeli 1.

| Parametr | Wartość |
|---|---|
| Napięcie pracy | Od 3 V do 7,5 V |
| Przekładnia | 120:1 |
| Prędkość przy 6 V | 160 obr/min |
| Prędkość przy 3 V | 60 obr/min |
| Maksymalny prąd | 2,8 A |
| Maksymalny moment obrotowy | 0,07 Nm |

**Tabela 1.** Parametry silnika SJ01

Jednak przed zastosowaniem wytypowanych silników upewniono się, że będą one w stanie zapewnić odpowiednie przyśpieszenie. Z tego powodu obliczono przewidywany potrzebny moment obrotowy dla robota. W tym celu przyjęto wagę pojazdu na 2 kg, promień kół na 3 cm, a planowane przyśpieszenie na 1 $\mathrm{m/s^2}$.

$$\mathrm{M} = \frac{\mathrm{F} * \mathrm{r}}{2}$$

$$\mathrm{M} = \frac{\mathrm{a} * \mathrm{m} * \mathrm{r}}{2} = \frac{1\frac{\mathrm{m}}{\mathrm{s}^2} * 2\mathrm{kg} * 0,03\mathrm{m}}{2} \approx 0,03\mathrm{Nm}$$

Obliczenia dotyczą potrzebnego momentu dla tylko jednego silnika, dlatego wartość podzielona jest we wzorze przez dwa. Wynik pokazuje, że wybrany silnik posiada moment wystarczający dla projektowanego robota. Prędkość obrotowa jest również zadowalająca do tego projektu.

Wybrany moduł silnika posiada enkoder magnetyczny dołączony do wału przed przekładnią, reaguje on więc na zwiększoną prędkość obrotową. Enkoder wysyła 8 impulsów w trakcie pełnego obrotu wału. Oznacza to, że jeden obrót koła powoduję przesłanie 960 impulsów, co wynika z przełożenia. Stanowi to bardzo dobrą podstawę do obliczenia prędkości obrotowej.

Maksymalny prąd silnika osiąga dużą wartość, jednak występuje on tylko w przypadku zatrzymania napędu i podaniu na niego wysokiego napięcia. W czasie rozruchu czas trwania zwiększonej wartości prądu jest bardzo krótki. Jednak w przypadku zablokowania kół robota może dojść do długotrwałego przeciążenia. Nieobracający się silnik posiada rezystancje jedynie kilku omów, na której odkłada się całość podanego napięcia. Powoduję to znaczny wzrost temperatury a w ostateczności uszkodzenie termiczne. W późniejszym etapie projektowania zadbano o zabezpieczenie przed takim zdarzeniem.

Kolejną zaletą wybranego modułu jest jego obudowa. SJ01 jest opakowany w plastikową osłonę zapewniającą łatwy montaż do platformy robota. Mieści ona również w sobie zębatki z przekładni.

## 3.4 Użyte koła

Wybrane koła zostały dobrze pokazane na zdjęciu 3. Są one półpneumatyczne, oznacza to, że zostały wypełnione powietrzem, lecz nie pod ciśnieniem. Posiadają bieżnik rowkowy bardzo popularny wśród kół samochodowych. Zapewnia on odprowadzanie zanieczyszczeń z miejsca styku koła z podłożem [2]. Ich szerokość wynosi 26 mm, a obwód 204 mm.

Znając ten parametr, możliwe jest obliczenie prędkość pojazdu. Zakładając, że silnik obraca się z prędkością 160 obr/min, to sam robot poruszać się będzie w tempie około 2 km/h. Pojazd mógłby jechać szybciej przy większych kołach, jednak nie jest to potrzebne.

## 3.5 Platforma oraz elementy montażowe

Platforma robota została wykonana z lekkiego oraz łatwego w obróbce plexiglasu. Materiał ten nie przewodzi prądu, co zmniejsza ryzyko powstania uszkodzeń elektrycznych w układzie. Platforma została zakupiona gotowa, jedynymi modyfikacjami było dodanie otworów dających miejsce na enkodery silnika.

Elementy robota zostały skręcone śrubami M3. Dodatkowo użyte zostały podkładki sprężyste, które nie pozwalają na odkręcanie się łączeń pod wpływem drgań. Płytka pcb oraz tylne koło zostały dołączone do podstawy z użyciem dystansów.

# Rozdział 4

# Projekt i realizacja mikroprocesorowego układu sterującego pracą maszyny

## 4.1 Wybór mikroprocesora

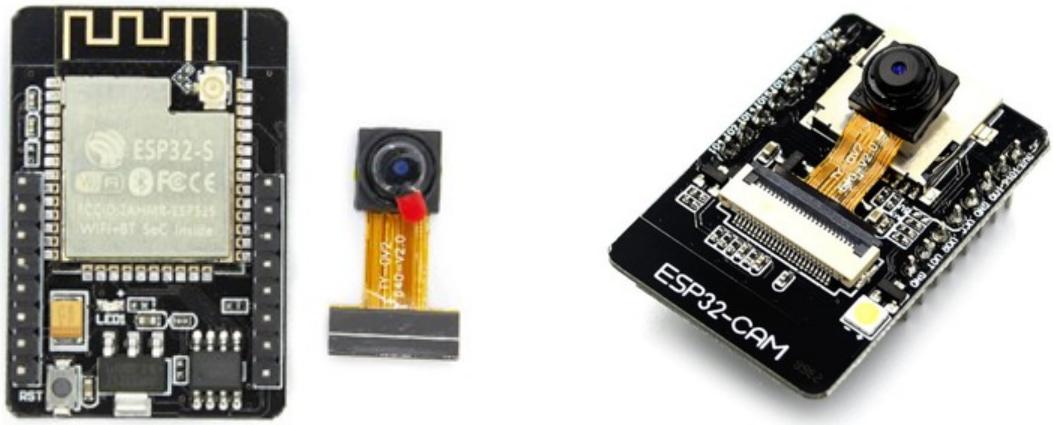
Wybór centralnej jednostki sterującej został ściśle podyktowany możliwością obsługi kamery, która jest głównym elementem robota. Możliwe rozwiązania były podobne jak w przypadku przykładów przytoczonych w rozdziale 2:

- użycie mikrokomputera takiego jak Rasberry Pi,
- użycie mikrokontrolera z wbudowaną pamięcią i modułami komunikacji.

Zdecydowano się na drugą opcję. W tym celu poszukiwano mikrokontrolera o wydajności wystarczającej, aby mógł obsłużyć kamerę oraz wysłać obraz użytkownikowi. Przykładowo popularna płytka Arduino uno z 8-bitowym mikrokontrolerem ATmega328P niestety jest niewystarczająca do tego zadania.

Jako alternatywę wybrano moduł ESP 32 CAM. Jest to płytka deweloperska z mikrokontrolerem ESP-32s oraz wbudowaną kamerą i modułem wi-fi. Została ona zaprojektowana przez firmę AI-Thinker. O jej wyborze zdecydowała wysoka wydajność, niska cena, łatwość programowania oraz duża popularność. Zakupiony moduł został przedstawiony na rysunku 4.

Wybrany mikrokontroler posiada 32-bitowy procesor. Oznacza to, że może on obsługiwać 32 bity, czyli 4 bajty danych jednocześnie. Już na tym etapie czyni go to bardziej wydajnym niż wspomniana wcześniej ATmega328P. ESP-32s wyposażony jest w zegar taktujący o częstotliwości od 80 MHz do 240 MHz z możliwością zmiany tej wartości. Definiuje ona czas pomiędzy kolejnymi instrukcjami. Domyślnie jest ona ustawiona na maksymalną wartość. W przypadku występowania wewnętrznego zegara taktującego nie jest konieczne dołączanie do mikrokontrolera zewnętrznego oscylatora. Wybrane urządzenie posiada również dwa rdzenie.




**(a)** ESP 32 CAM z tyłu **(b)** ESP 32 CAM z przodu

**Rysunek 4.** Zastosowany moduł z kamerą
Źródło: `sklep.msalamon.pl` oraz `allegro.pl`

Jest to element, w którym zachodzą procesy obliczeniowe. Większa ilość oznacza również większą prędkość.

Powyższa analiza upewniła, że procesor będzie wystarczająco wydajny. Dodatkowo przeprowadzono test za pomocą dołączonej kamery. Użyty wtedy został przykładowy program dostarczony przez producenta w pakiecie bibliotek. Mikrokontroler zaprogramowano przy pomocy Arduino IDE. Test pokazał, że wydajność kamery mieści się pomiędzy 30, a 60 klatkami na sekundę w zależności od rozdzielczości.

Mikrokontroler posiada interfejsy komunikacyjne Bluetooth oraz wi-fi. Korzystając z wcześniej przytoczonych przykładów projektów, wybrano komunikację internetową. Wysyłanie sygnału wi-fi jest możliwe dzięki wbudowanej w płytkę antenie. Mikrokontroler obsługuje standardy 802.11 b/g/n. Definiują one sposób przesyłania informacji oraz parametry sieci takie jak przepustowość. Najbardziej zaawansowany 802.11 n umożliwia przesyłanie danych z prędkością do 150 Mbps. Standard ten zapewnia działanie na częstotliwości 2,4 GHz, umożliwiając transmisje bez zakłóceń. 802.11 n jest również często wykorzystywany w ruterach domowych. Płytka posiada również możliwość dołączenia zewnętrznej anteny w celu zwiększenia zasięgu.

Połączenie internetowe z kontrolerem może być zrealizowane w dwóch trybach pracy. Jako punkt dostępu oraz stacja. W pierwszej opcji układ zachowuje się jak ruter, do którego można podłączyć inne urządzenie. Umożliwia to działanie komunikacji bez udziału dodatkowych przyrządów. W drugiej konfiguracji ESP-32 CAM łączy się z istniejącą już siecią. Dane wysyłane

w ten sposób są przekazywane do użytkownika za pomocą rutera. Umożliwia to zwiększenie zasięgu odbierania danych. W ramach tego projektu zdecydowano się na użycie drugiej opcji.

Mikrokontroler posiada 520 KB wbudowanej pamięci SRAM. Jest to pamięć o swobodnym dostępie wykorzystywana w czasie działania programu. Informacje w niej nie zapisują się po wyłączeniu zasilania. Dodatkowo na płytce ESP-32 CAM znajduje się 4 MB zewnętrzej pamięci PSRAM o szybkim dostępie. Zaprogramowany kod mikrokontrolera zostaje zapisany w pamięci nieulotnej SPI flash, której urządzenie posiada 32 Mbity.

Rekomendowane jest zasilanie mikrokontrolera napięciem o wartości 3,3 V. Jest to napięcie pracy tego urządzenia. Podanie większej wartości na zasilanie lub jeden z pinów spowodowałoby uszkodzenie układu. Jednak na samej płytce mieści się wyprowadzenie na 5 V. Jest to spowodowane tym, że producent dołączył zewnętrzny stabilizator napięcia. Dzięki temu płytkę zasilano 5 V bez uszkodzenia układu, jednak nadal portom IO dostarczano napięcie maksymalne o wartości 3,3 V.

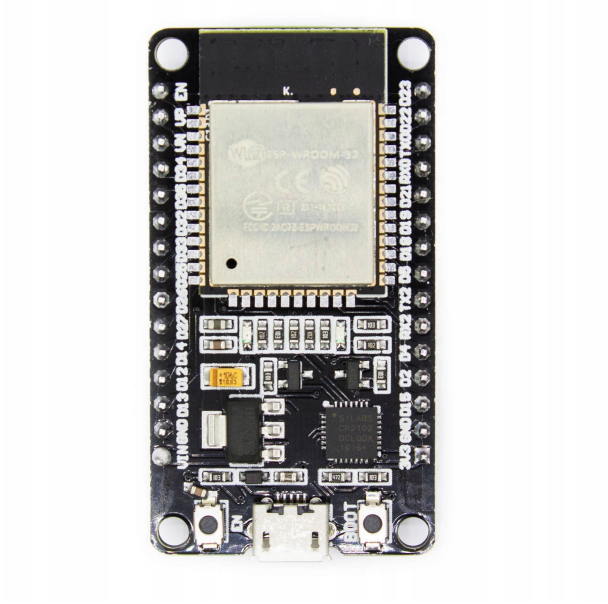
Kolejnym elementem płytki dołączonym obok mikrokontrolera oprócz stabilizatora i pamięci PSRAM jest dioda LED. Z powodu swojej dużej jasności sprawdza się przy oświetlaniu drogi robota. Niestety została na stałe podłączona do jednego z wyprowadzeń kontrolera, co wyklucza jego użycie w inny sposób. Płytka rozwojowa posiada również wejście na kartę pamięci, jednak nie jest ono używane w tym projekcie. Najważniejszą kwestią jest połączenie z kamerą. Zostało ono zrealizowane przez producenta płytki, który zaprojektował ścieżki tak, aby kamerę można było w łatwy sposób dołączyć za pomocą złącza w sposób pokazany na rysunku 4b.

Płytka ESP 32 CAM posiada 16 wyprowadzeń. Sześć z nich to piny zasilające albo uziemienie. Reszta jest dostępna dla użytkownika do programowania. Niestety niektóre z nich są wykorzystywane podczas używania kamery, diody LED lub portu UART. Ostatecznie użytkownik podczas używania kamery dostępne ma jedynie 6 wyprowadzeń. Z tego powodu należało znaleźć sposób na rozszerzenie ilości możliwych konfigurowalny wejść i wyjść. W tym celu przetestowano działanie ekspandera portów oraz komunikacji I2C. Niestety ekspander nie był w stanie obsłużyć sygnału PWM wysyłanego do silników oraz szybko zmiennego sygnału z enkoderów. Ostatecznie zdecydowano się na użycie drugiego mikrokontrolera.

## 4.2 Wybór dodatkowego mikroprocesora

Drugim mikrokontrolerem użytym w układzie został bliźniaczo podobny do wcześniej opisywanego ESP32-WROOM-32. Jest on umieszczony na płytce ESP 32 DevKit v1, która już nie posiada wbudowanej kamery. Jego wybór został spowodowany chęcią użycia podobnego urządzenia, co uprościło implementacje oprogramowania. Dodatkowo fakt, że moduł ten

również posiada interfejs wi-fi, umożliwiło użycie go jako centralnej jednostki do sterowania samym robotem, a poprzedni mikrokontroler użyto jedynie do wysyłania obrazu. Zapewniło to większą wydajność kamery oraz jej modułowy charakter. Użyta płytka została przedstawiona na grafice 5.



**Rysunek 5.** Płytka ESP 32 DevKit v1
Źródło: allegro.pl

Mikrokontroler ESP 32 posiada taki sam mikroprocesor oraz układy dodatkowe jak wcześniej wybrany moduł. Główne różnice znajdują się na samej płytce uruchomieniowej. Tak jak w poprzednim przypadku posiada ona stabilizator liniowy na 3,3 V umożliwiający zasilanie układu napięciem o wartości 5 V. Jednak na tym podobieństwa się kończą. Płytka DevKit zawiera konwerter USB/UART umożliwiający programowanie mikrokontrolera bezpośrednio z komputera. Poza tym posiada diody oraz przycisk resetu. Dzięki mniejszej liczbie dodatków płytka udostępnia więcej wyjść użytkownikowi.

Wybrany mikrokontroler posiada jeszcze szereg przydatnych funkcji. Nie zostały one wymienione w poprzednim rozdziale dlatego, że nie są wykorzystywane przez wcześniej wspomniany model. Jednak oba układy posiadają te same możliwości.

Jedną z nich jest wysyłanie sygnału PWM, czyli sygnału o modulowanej szerokości impulsu. ESP 32 w przeciwieństwie do Arduino nie posiada konkretnych pinów PWM. Za to ma on możliwość do wygenerowania 16 niezależnych kanałów i dołączenia do nich dowolnych programowalnych wyprowadzeń. Rozdzielczość sygnału wynosi maksymalnie 16 bitów, a jego częstotliwość może być ustawiona na wartość do 80 MHz. Sygnał PWM jest wykorzystywany przez robota do sterowania silnikami.

Mikrokontroler posiada również port UART. Jest to moduł asynchroniczny do przekazywania i odbierania danych. Został on użyty do zaprogramowania mikrokontrolera oraz komunikacji dwóch płytek między sobą.

Oprócz wymienionych funkcji mikrokontroler oferuje też szereg możliwości, które nie zostały wykorzystane. Są to między innymi sensor dotyku oraz czujnik Halla.

## 4.3 Wybór kamery

Kamera wybrana do mikrokontrolera ESP32s musiała być kompatybilna z połączeniami zaprojektowanymi przez producenta płytki. Zdecydowano się na użycie OV2640 już dołączonej do zestawu. Była ona przedstawiona na rysunku 4.

Jest to miniaturowa kamera o niewielkim poborze mocy rzędu 140 mW. Najwyższe potrzebne jej napięcie zasilające wynosi 3,3 V. Oferuje ona obsługę takich rozdzielczości jak UXGA (1600 $\times$ 1200), SVGA (800 $\times$ 600) oraz CIF (352 $\times$ 288). Jej maksymalna wydajność wynosi 60 klatek na sekundę [6]. Dodatkowo zapewnia ona możliwości zmiany jakości, saturacji kolorów oraz wiele innych. Jej funkcje zostały później wykorzystane w celu poprawienia czytelności lub płynności widzianego obrazu.

Sama kamera do poprawnego funkcjonowania potrzebuje zewnętrznego sygnału zegarowego podanego z mikroprocesora. Komunikacja z kamerą odbywa się za pomocą szeregowej magistrali I2C. Służy ona do przekazywania otrzymanego obrazu oraz otrzymywania polecań. Dzięki swojej szybkości komunikacja ta bardzo dobrze sprawdza się w przesyłaniu danych.

## 4.4 Wybór sterownika silników

Po wybraniu elementów umożliwiających komunikacje z użytkownikiem oraz przekazywanie obrazu zajęto się kwestią ruchu robota. W tym celu poszukiwano mostka sterowniczego umożliwiającego kontrolowanie pracy silników. Wymagania mu postawione to:

- prąd wyjściowy impulsowy wyższy od maksymalnego prądu silnika,
- prąd ciągły wyjściowy wyższy od prądu pracy silników,
- zabezpieczenia termiczne,
- możliwie szeroki zakres napięcia wejściowego,
- zapewnienie różnych trybów pracy silników,
- możliwość obsługi dwóch silników jednocześnie.

Wybierając sterownik, nie kierowano się możliwością stałego dostarczania napędom maksymalnej wartości prądu, czyli 2,8 A. W przypadku dopuszczenia do takiej sytuacji najprawdopodobniej uszkodziłby się silniki. Chociaż wybierając słabszy moduł, to go narażamy

na zniszczenie. Zamiast kupowania mocniejszego elementu zdecydowano się na realizacje zabezpieczeń przed zatrzymaniem silników oraz wysokim prądem.

Sterownik, który został wybrany do realizacji projektu to TB6612FNG firmy TOSHIBA. Spełnia on wszystkie wymienione wymagania. Jest to moduł w obudowie SMD, zdecydowano się więc na zakup płytki z wyprowadzeniami do montażu THT. Dzięki temu jest ona łatwiejsza w lutowaniu. Płytka wyprodukowana jest przez firmę Pololu. Posiada ona również potrzebne kondensatory filtrujące, dzięki temu jest zgodna z propozycją aplikacji zaproponowaną przez firmę TOSHIBA [8]. Na rysunku 6 przedstawiono zdjęcie modułu.

**Rysunek 6.** TB6612FNG umieszczony na płytce Pololu 713
Źródło: botland.com.pl

Wybrany sterownik opary jest na podwójnym mostku H złożonym z tranzystorów MOSFET oraz diod. Konfiguracja ta zapewnia wszystkie możliwe tryby pracy, czyli jazdę w obu kierunkach zatrzymanie oraz swobodne obroty dla obu silników. Możliwe jest połączenie obu kanałów w celu uzyskania większej wydajności prądowej.

## 4.5 Wybór sensorów odległości

Sensory odległości w projektowanym robocie pełną funkcję informacyjną oraz zabezpieczającą. Znajdują się ich dwie sztuki po jednej na przód oraz tył robota. Ich zadaniem jest pomiar odległość od przeszkód. Na podstawie otrzymanych wyników mikroprocesor może zatrzymać robota. Jest to pierwszy stopień zabezpieczenie przed uszkodzeniem układu.

Jako wspomniany sensor wybrano ultradźwiękowy układ HC-SR04. Jego działanie jest podobne do lokalizowania przeszkód przez nietoperze. Sensor po otrzymaniu sygnału wyzwalającego wysyła osiem impulsów ultradźwiękowej fali. Odbiją się one od przeszkód i wracają do odbiornika. Na podstawie zmierzonego czasu powrotu wysyłany jest impuls do mikrokontrolera o szerokości proporcjonalnej do zmierzonej odległości [5].

Na pomiar nie wpływa światło oni kolor powierzchni w przeciwieństwie do alternatywnego dalekomierza laserowego. Niestety sonar nie wykrywa miękkich materiałów takich jak zasłonki, ponieważ słabo odbijają falę dźwiękową. Na szczęście zderzenie się z nimi nie wywołuje poważnych skutków.

Układ zasilany jest napięciem 5 V. Taką samą wartość wysyła ze swojego wyjścia. Pamiętając o tym, że użyte mikrokontrolery działają na niższe poziomy, dodano do wyprowadzeń sensorów dzielniki napięcia. Sprowadzają one sygnał do wartości 3,3 V bez zniekształcenia odpowiedzi układu. Ten sam zabieg został również wykonany w przypadku sygnałów z enkoderów.

## 4.6 Wybór źródła zasilania

Wybierając źródło zasilania, zdecydowano się na użycie akumulatorów litowo-polimerowych. Zapewniają one duży zapas energii przy niewielkiej wadze. Akumulatory złożone są z pakietów ogniw połączonych szeregowo. Wartość nominalna napięcia pojedynczego ogniwa wynosi 3,7 V.

Aby zapewnić wystarczające napięcie wszystkim układom, wybrano pakiet złożony z dwóch ogniw o łącznym napięciu 7,4 V. Jednak wartość ta nie jest stała w trakcje użytkowania, spada ona wraz z rozładowywanie się baterii. Maksymalnie może wynosić 8,4 V, a minimalnie 6 V. Dopuszczenie do pełnego rozładowania powoduje jeszcze większe spadki napięć. Może to doprowadzić do nieprawidłowego działania układów oraz skrócenia życia baterii. Z tego powodu w późniejszym etapie projektu zdecydowano się na pomiar poziomu napięcia.

Maksymalny prąd rozładowywania wynosi 32 A. Jest to wartość znacznie przekraczająca potrzeby układu. Może ona wystąpić w trakcie zwarcia i gwałtownie uszkodzić elektronikę. Pojemność akumulatora wynosi 800 mAh, oznacza to, że może on dostarczać 0,8 A nieprzerwanie przez godzinę. Zużycie prądu przez robota nie jest stałe w czasie, więc trudno jest określić czas działania akumulatora. Bliskie prawdy może być stwierdzenie, że będzie on działać przy pełnym naładowaniu przez około jedną godzinę.

W trakcje korzystania z pakietów Li-Pol postępowano zgodnie z zasadami użytkowania. Ładując je, zachowano szczególną ostrożność oraz użyto dostosowanej ładowarki z balanserem.

## 4.7 Wybór sposobu stabilizacji napięcia

Niemożliwe jest użycie napięcia z akumulatorów bezpośrednio do zasalania układów cyfrowych z dwóch powodów. Jest ono za wysokie oraz niestabilne. Aby robot mógł działać poprawnie, należało użyć odpowiedniego układu zmniejszającego napięcie do 5 V. Można tego dokonać za pomocą stabilizatora napięcia albo przetwornicy step-down.

Przetwornica jest układem impulsowo zmieniającym poziom napięcia. Cechuje się wysoką sprawnością, dzięki czemu traci mniejszą ilość energii oraz mniej się nagrzewa. Jednak jej charakter impulsowy wymaga użycia większej ilości kondensatorów, aby zapewnić odpowiednie działanie układów mikroprocesorowych. Jest ona układem droższym oraz zajmującym więcej miejsca.

W tym projekcie zdecydowano się na użycie stabilizatora liniowego. Jest on bardzo popularnym układem. Jego działanie opiera się głównie na wytracaniu nadmiaru napięcia w postaci ciepła. Posiada niedużą sprawność, jednak jest układem tańszym, który generuje mniej zakłóceń.

Wybierając stabilizator liniowy, zwrócono głównie uwagę na maksymalny prąd, który ma dostarczyć do układu. W celu wyboru odpowiedniego modelu sporządzono tabele 2 opisującą maksymalny pobór prądu elementów zasilanych przez stabilizator.

| | Maksymalne zużycie prądu [mA] |
|---|---|
| ESP 32 CAM | 310 |
| ESP 32 DevKit v1 | 100 |
| HC-SR04 x2 | 30 |
| TB6612FNG | 2,2 |
| Enkodery x2 | 11 |
| Suma | 453,2 |

**Tabela 2.** Zestawienie maksymalnego poboru prądu przez układ robota

Wysokie zużycie prądu przez ESP 32 CAM wynika z użycia mocnej diody LED. Oba mikrokontrolery pobierają znaczną część energii na komunikacje wi-fi. W każdym przypadku do obliczeń używano maksymalnego poboru prądu odczytanego z dokumentacji technicznej użytych urządzeń.

Do spełnienia oczekiwanych wymagań prądowych wybrano popularny układ LM7805 w obudowie TO-220. Stabilizuje on napięcie do poziomu 5 V, umożliwiając zasilenie wszystkich układów logicznych w robocie. Stabilizator ten jest w stanie dostarczyć do 1.5 A oraz działać przy minimalnym napięciu 7 V. Znając napięcie zasilania i maksymalny prąd możliwe jest

obliczenie szczytowych strat mocy.

$$(\mathrm{V_{max}} - 5\mathrm{V}) * \mathrm{I_{max}} = \mathrm{P_{strat}}$$

$$(8,4\mathrm{V} - 5\mathrm{V}) * 0,453\mathrm{A} \approx 1,54\mathrm{W}$$

Gdzie $\mathrm{V_{max}}$ maksymalne napięcie baterii, $\mathrm{I_{max}}$ to maksymalny prąd stabilizatora.

Następnie możliwe jest obliczenie temperatury LM7805, znając parametry z dokumentacji [7].

$$\mathrm{R_{thJA}} * \mathrm{P_{strat}} + \mathrm{T_{amb}} = \mathrm{T}$$

$$50\frac{\mathrm{K}}{\mathrm{W}} * 1,54\mathrm{W} + 25°\mathrm{C} \approx 102°\mathrm{C}$$

Gdzie $\mathrm{R_{thJA}}$ to to rezystancja miedzy złączem a otoczeniem odczytana z noty katalogowej, $\mathrm{T_{amb}}$ to temperatura otoczenia, a T to maksymalna temperatura układu.

Obliczona temperatura jest dopuszczalna przez producenta. Jednak może ona być niebezpieczna dla użytkownika oraz obiektów w otoczeniu, dlatego zdecydowano się na użycie radiatora. Do obliczenia temperatury po dołączeniu radiatora została użyta rezystancji termicznej pomiędzy złączem i obudową oraz radiatorem a otoczeniem.

$$(\mathrm{R_{thJC}} + \mathrm{R_{thCA}}) * \mathrm{P_{strat}} + \mathrm{T_{amb}} = \mathrm{T}$$

$$(5\frac{\mathrm{K}}{\mathrm{W}} + 21\frac{\mathrm{K}}{\mathrm{W}}) * 1,54\mathrm{W} + 25°\mathrm{C} \approx 65°\mathrm{C}$$

Taka wartość maksymalna temperatury jest już dopuszczalna w układzie.

Do stabilizatora zostały dołączone kondensatory filtrujące. Zmniejszają one zakłócenia o wysokiej częstotliwości. Dodatkowo umieszczono kondensatory elektrolityczne, które działają jako tymczasowe źródło energii w czasie dużego obciążenia zasilania. Zostały one dodane, ponieważ robot w czasie rozruchu silników pobierał tyle energii, że powodowało to czasem wyłączenie mikrokontrolerów.

## 4.8 Układ pomiaru napięcia

Pomiar napięcia baterii odbywa się za pomocą przetwornika analogowo-cyfrowego w mikrokontrolerze ESP32-WROOM-32. Aby skorzystać z tej funkcji poprawnie, przeanalizowano informacje zamieszczone w nocie katalogowej [4]. Wynika z nich, że dokładna konwersja sygnałów na cyfrowy jest możliwa tylko przy wartościach napięcia poniżej 2,45 V.

Z tego powodu zdecydowano się na użycie dzielnika napięcia zmniejszającego maksymalną wartość napięcia baterii do poziomu 1,71 V. Otrzymano w ten sposób pomiar z dokładnością +-

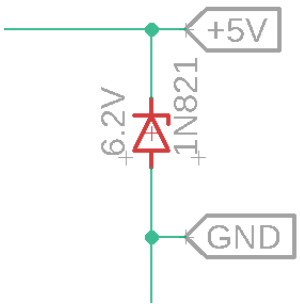
60 mV, według noty katalogowej. Dzielnik zrealizowano za pomocą rezystorów o wartości 39k oraz 10k oma. Aby otrzymać prawdziwą wartość napięcia baterii, pomiar został programowo wymnożony przez 4,9. Co ostatecznie sprowadziło błąd do wartości +-294 mV.

Dodatkowo producent podaje, że w przypadku użycia transmisji wi-fi błąd może okazać się jeszcze większy. W celu jego eliminacji dodano kondensator filtrujący przed pinem ADC oraz zastosowano wielokrotne pobieranie próbek. Dokładność pomiaru zwiększyła się, gdy układ został przeniesiony z płytki prototypowej na PCB.

## 4.9 Dodatkowe elementy układu

Oprócz wymienionych wcześniej modułów dodano do układu jeszcze kilka elementów ułatwiających jego obsługę oraz zapewniających bezpieczeństwo. Znajduje się w nim przełącznik dwupozycyjny umożliwiający wyłączenie zasilania całego robota. Za nim umieszczony został bezpiecznik topikowy. Bezpiecznik składa się z cienkiego przewodu, który topi się przy wartości prądu wynoszącej 5 A. Jest on ostatecznym zabezpieczeniem przed zwarciem układu lub krytycznym zatrzymaniem pracy silników. Umieszczony jest w obudowie ułatwiającej jego wymianę. Za bezpiecznikiem umiejscowiona została dioda sygnalizująca podłączenie zasilania do układu. Rezystor ograniczający prąd został do niej dobrany tak, aby świeciła z niedużą jasnością. Prąd przez nią płynący wynosi około 2 mA. Zapewnia to dobrą widoczność diody, przy tym nie oślepiając użytkownika.

Dodatkowo do części układu zasilanej z 5 V została dołączona dioda Zenera. Sposób jej umieszczenia został pokazany na rysunku 7. Jej napięcie blokowania wynosi 6,2 V. W czasie bezawaryjnego funkcjonowania uniemożliwia ona przepływ prądu, więc działa jak przerwa. W przypadku awarii i dostania się do układu 5 V napięcia wyższego zaczyna ona przewodzić. Dioda ochrania układy scalone przez zbyt dużym napięciem. W przypadku zaistnienia takiej sytuacji to ona zostanie zniszczona, a najprawdopodobniej układ wyłączy bezpiecznik.



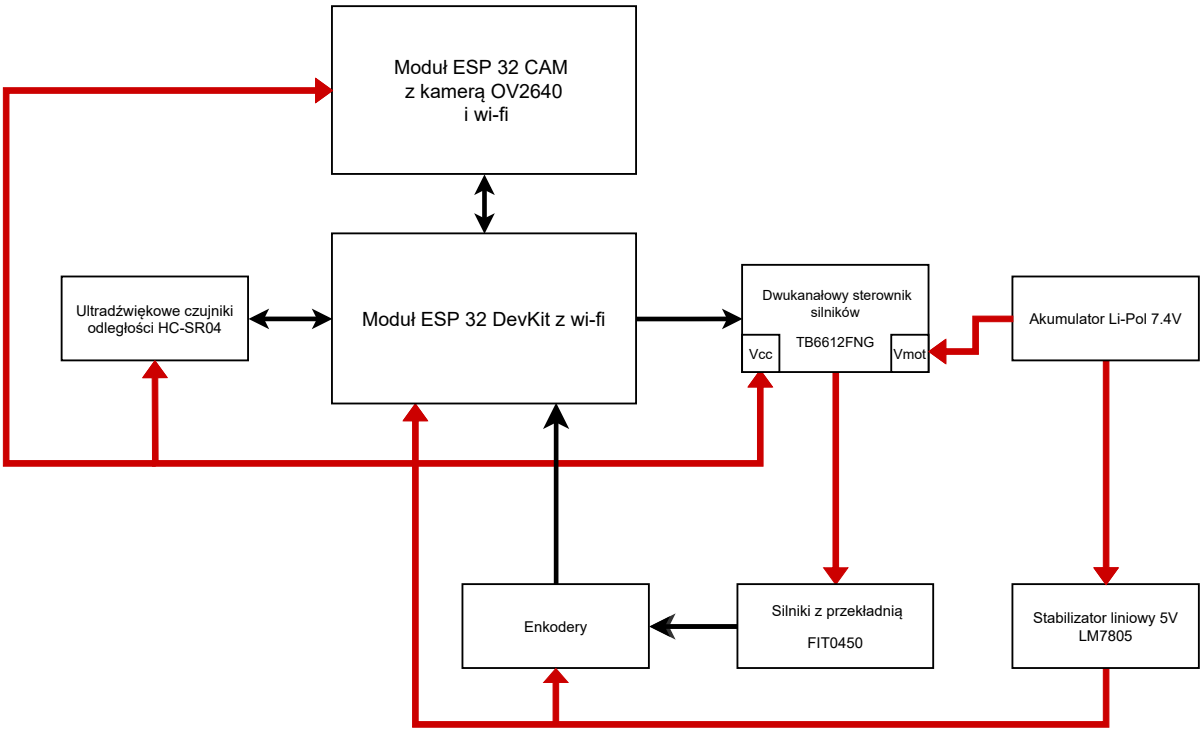

**Rysunek 7.** Sposób podłączenia diody Zenera

Do układów mikroprocesorowych dołączone zostały po dwa kondensatory filtrujące. Elektrolityczny o wartości 10 uF oraz ceramiczny 0.1 uF. Ich różnorodność zapewnia lepszą wydajność. Użycie bezpiecznika oraz diody Zenera tak jak i wiele innych informacji przydatnych do konstrukcji robota zostało zaczerpniętych z książki [3].

## 4.10 Schemat połączeń

Blokowy schemat połączeń został pokazany na rysunku 8. Przedstawia on relacje między modułami w układzie robota. Schemat zaprojektowano w programie Draw.io. Czerwonymi strzałkami zaprezentowano połączenia zasilania. Większość elektroniki podłączona jest przez stabilizator liniowy. Wyjątek stanowi wejście zasilania Vmot znajdujące się na sterowniku. Służy ono do przekazywania napięcia na silniki. Wejście te jest bezpośrednio połączone z baterii. Mimo to logika układu sterownika silników zasilana jest z 5 V.

Czarnymi strzałkami ukazano połączenia logiczne między układami. Ze schematu można wywnioskować, że moduł ESP 32 DevKit stanowi centralną jednostkę układu. Kierunek strzałek informuje o tym, czy mikrokontroler wysyła, czy odbiera dane sygnały.

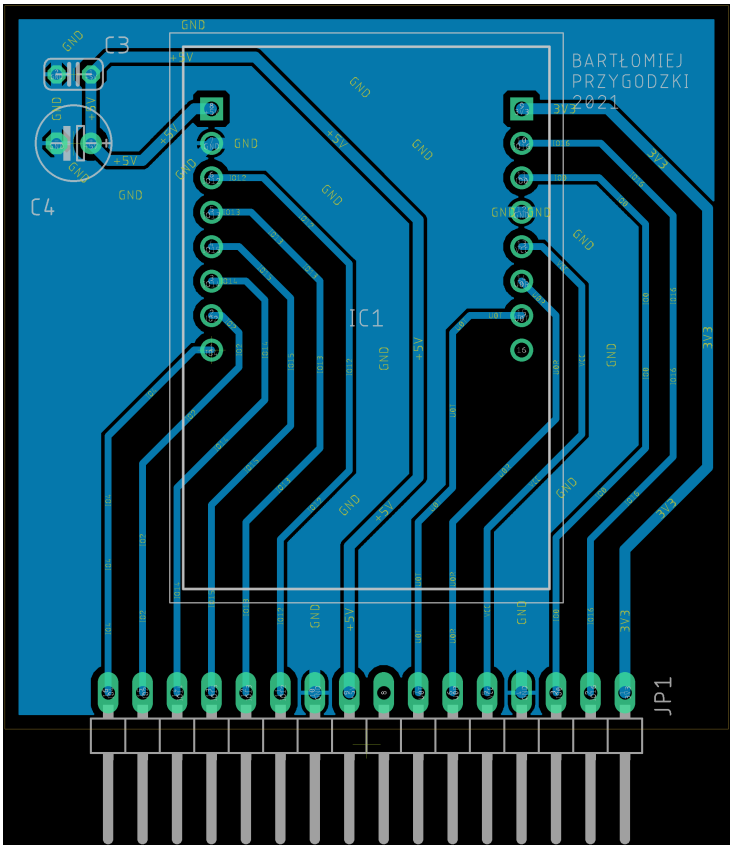



**Rysunek 8.** Schemat blokowy połączeń elektrycznych

Ideowy schemat elektryczny połączeń robota wykonany w programie Eagle znajduje się w załączniku. Zawiera on wszystkie elementy układu, jednak nie odnosi się bezpośrednio do projektu płytek.

## 4.11 Projekt płytek

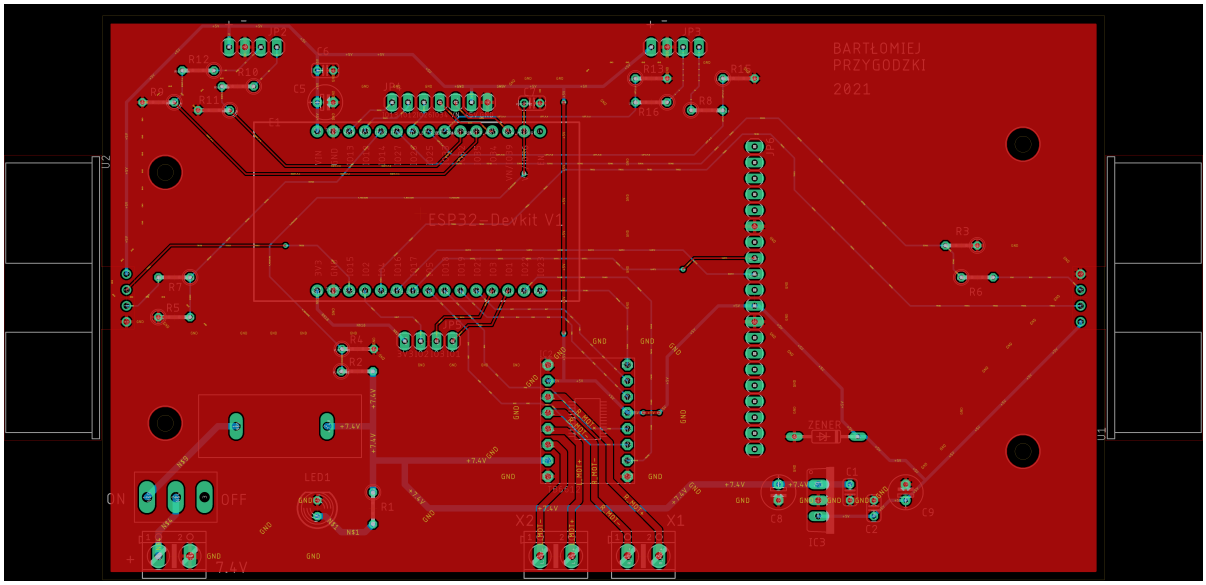
W ramach budowy robota wykonane zostały dwie płytki pcb. Pierwsza z nich posiada moduł ESP 32 CAM oraz kondensatory filtrujące. Jej wymiary wynoszą 53 × 53 mm. Została stworzona jako oddzielny element, aby umożliwić pionowe ustawienie kamery, przez szesnaście kątowych wtyków goldpin umieszonych w dolnej części płytki. Wtyki oprócz pozycjonowania kamery zapewniają połączenie elektryczne z główną płytka. Wszystkie ścieżki zostały poprowadzone w warstwie dolnej. Dodatkowo wyznaczono pole masy na całej powierzani płytki, aby zapewnić lepsze odprowadzanie ciepła. Wykonany projekt został pokazany na rysunku 9 a jego pełen schemat jest umieszczony w załączniku.




**Rysunek 9.** Projekt płytki z kamerą wykonany w programie Eagle

Druga zaprojektowana płytka zawierała już wszystkie pozostałe układy. Znajduje się na niej dwudziestu pinowe złącze kołkowe, w którym zostanie umieszczony poprzedni obwód

drukowany. Elementy na płytce głównej umieszczono tak, aby zgromadzić wszystkie układy zasilane z napięcia 7,4 V blisko siebie. Dodatkowo umieszczono pionowe wtyki goldpin połączone z nieużywanymi wyprowadzeniami mikrokontrolera. W warstwie opisowej została zawarta informacja o nazwie wykonanych przy nich połączeń. Wymiary płytki wynoszą 160 $\times$ 90 mm. Pokrywają one znaczną część powierzchni robota. Rozmiary ścieżek zostały dobrane pod kątem przepływającego przez nie prądu. Maksymalny rozmiar wynosi 1 mm, co umożliwia dostarczenie najwyższej wartości prądu do układu bez wysokiego wzrostu temperatury. Projekt płytki został pokazany na rysunku 10 a jej pełen schemat również jest umieszczony w załączniku.




**Rysunek 10.** Projekt płytki głównej wykonany w programie Eagle

Obie płytki zostały wydrukowane na zlecenie w firmie JLCPCB. Wszystkie ograniczenia produkcyjne zostały uwzględnione w trakcje projektowania płytek. Każda ze ścieżek została poprowadzona ręcznie, unikając ostrych katów. W miejscach mikrokontrolerów i sonarów na płytce zostały wlutowane złącza kołkowe. Umożliwia to łatwe wyjmowanie urządzeń i ewentualne użycie w innych projektach. W ramach projektowania zostały napisane dwie własne biblioteki dla obudowy bezpiecznika oraz włącznika. Płytki zostały własnoręcznie zlutowane. Ich zdjęcia są zamieszczone w załączniku.

# Rozdział 5

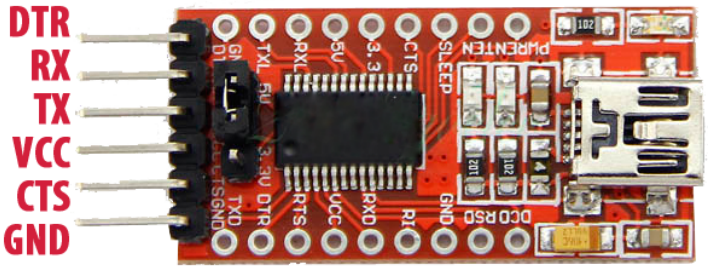
# Implementacja oprogramowania urządzenia

## 5.1 Sposób implementacji programów

W projekcie wykorzystano dwa mikrokontrolery, z tego powodu napisano oddzielny program na każdy z nich. Sposób ich implementacji jest bardzo zbliżony. W tym celu wykorzystano program Arduino IDE. Jest to zintegrowane środowisko programistyczne służące przede wszystkim do programowania płytek Arduino. Zawiera ono kompilator oraz uploader umożliwiający wgrywanie kodu do mikrokontrolera poprzez wybrany port komputera. Aby użycie programu było możliwe, zainstalowano rozszerzenie ESP 32 do menadżera płytek. Jest ono stworzone przez firmę Espressif Systems, czyli producenta oryginalnych mikrokontrolerów. Działa w przypadku obu płytek.

Wgrywanie programu do ESP 32 DevKit jest niezwykle proste. Wystarczy w tym celu podłączyć płytkę do komputera poprzez kabel micro usb. Jest to możliwe dzięki wbudowanemu konwerterowi, który zamienia komunikacje USB na bardziej zrozumiałą dla mikrokontrolerów transmisje UART.

Niestety zapisywanie programu w ESP 32 CAM jest trudniejsze. Płytka nie posiada wspomnianego wcześniej konwertera. Z tego powodu zakupiono zewnętrzny moduł FTDI232, przedstawiony na rysunku 11. Zamienia on komunikacje USB na UART, jednocześnie przy tym umożliwiając zasilanie układu. Łącząc piny RX oraz TX modułu odpowiednio z TX oraz RX mikrokontrolera możliwa jest dwustronna komunikacja. W celu wgrania programu należało za każdym razem połączyć port GPIO 0 z uziemieniem, aby wprowadzić mikrokontroler w tryb programowania. Po wgraniu kodu połączenie usuwano i resetowano urządzenie.




**Rysunek 11.** Moduł FTDI232 użyty do programowania ESP 32 CAM
Źródło: iot-guider.com

## 5.2 Podział zadań między mikrokontrolery

Zadaniami mikrokontrolera znajdującego się na płytce ESP 32 CAM są:

- pobieranie obrazu z kamery,
- obsługa przesyłania zdjęć przez wi-fi,
- zmiana parametrów kamery na podstawie poleceń wysyłanych przez port UART.

Funkcjami mikrokontrolera na płytce ESP 32 DevKit są:

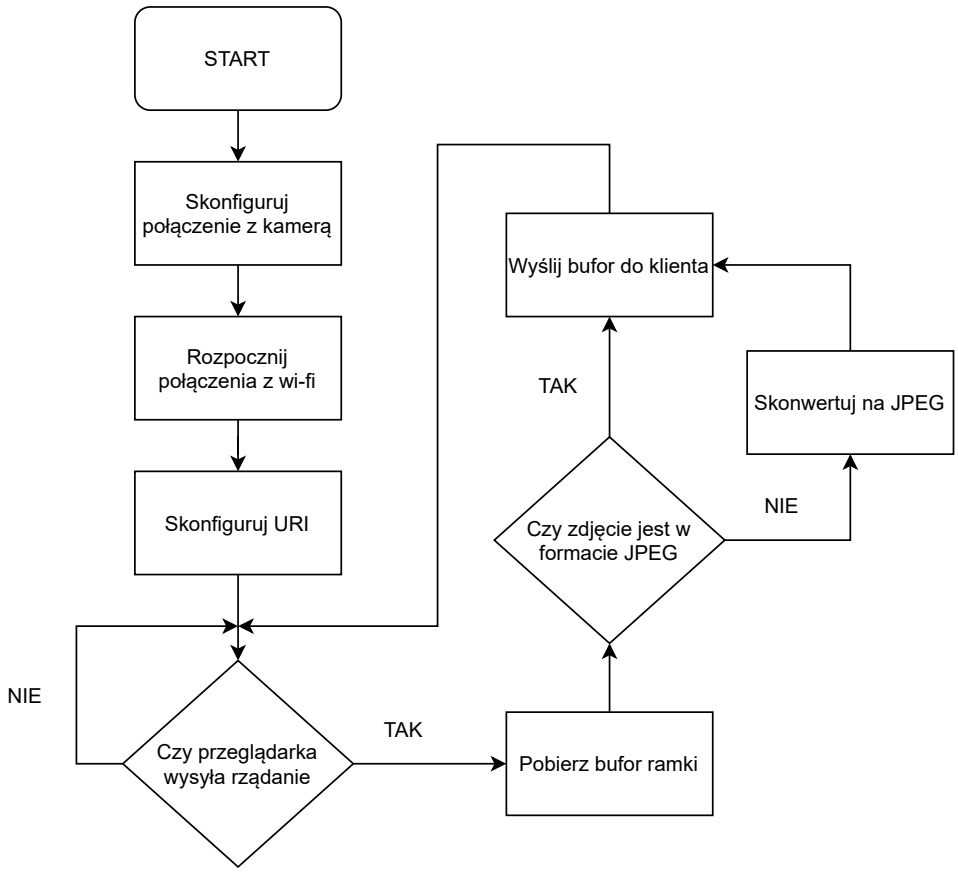
- komunikacja z użytkownikiem poprzez panel sterujący w przeglądarce,
- obliczanie odległości na podstawie sygnału z sensorów,
- pomiar napięcia baterii,
- wysyłanie poleceń o zmianie parametrów kamery za pomocą portu UART,
- obsługa przerwań zewnętrznych z enkoderów,
- sterowanie prędkością silników.

## 5.3 Obsługa kamery oraz transmisji obrazu

Kod obsługujący kamerę znajduje się całkowicie na mikrokontrolerze ESP 32 CAM w oddzielnym pliku. Jest to kod zaczerpnięty z przykładu CameraWebServer firmy Espressif. W ramach tej pracy użyto jedynie fragmentu o streamowaniu obrazu. Zasada działania kodu przedstawiona jest na rysunku 12.

Aby wysłać obraz z mikrokontrolera na początku następuje konfiguracja połączenia z kamerą. Odbywa się ona poprzez stworzenie struktury. Przechowywane w niej są dane o połączeniach z pinami, częstotliwości zegara kamery oraz formacie obrazu. Następnie znacznik do stworzonej struktury jest wysyłany do funkcji konfiguracji z biblioteki esp_camera.

Obsługa połączenia z wi-fi odbywa się dzięki bibliotece Arduino. Połączenie wymaga jedynie podania nazwy oraz hasła lokalnej sieci internetowej. Następnie za pomocą funkcji WiFi.begin




**Rysunek 12.** Schemat blokowy kodu obsługi kamery

jest ono rozpoczynane. Adres IP urządzenia jest wybierany automatycznie przez ruter. Zostaje on później wyświetlany w konsoli portu szeregowego. Pod tym przydzielonym adresem dostępny będzie obraz z kamery.

W następnym kroku konfiguracja adresu URI definiuje, gdzie dokładnie będzie znajdować się obraz z kamery. Przykładowo może to być adres http://192.168.1.27/stream. W tym miejscu następuje również konfiguracja handlera. Jest to funkcja, która obsługuje zdarzenia z przeglądarki. W tym przypadku będzie ona wysyłała obraz z kamery podczas wywołania. Kamera wykonuje zdjęcie jedynie, kiedy użytkownik korzysta z podanego adresu. Kiedy nie ma klienta, obraz nie jest nagrywany. Jednocześnie może istnieć tylko jeden stream obrazu.

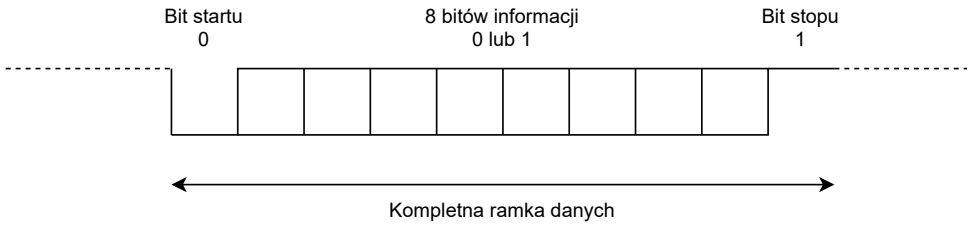
Gdy klient korzysta ze strony internetowej, przeglądarka wysyła żądanie obsługi zdarzenia do ESP 32 CAM. Powoduje to uruchomienie funkcji sczytywania obrazu. Polega ona na pobraniu bufora ramki z kamery. Jest to część pamięci przeznaczonej do przechowywania obrazu. Jeżeli obraz nie jest w formacie JPEG, jest on konwertowany przez mikrokontroler. Jednak sama kamera OV2640 jest w stanie wykonać taką konwersję, chyba że ustawiono inaczej.

Ostatecznie bufor ramki wraz z jej długością jest wysyłany do klienta. Wynikiem tego jest wyświetlenie obrazu w formie JPEG w przeglądarce. Częste powtarzanie żądania przez stronę skutkuje płynny przekazem wideo.

## 5.4 Komunikacja UART

W celu przekazywania poleceń między mikrokontrolerami wybrano komunikacje poprzez UART. Jest to sprzęt służący do asynchronicznej komunikacji szeregowej. Oznacza to, że bity danych są wysyłane po kolei. Asynchroniczność sygnalizuje zaś, że urządzenia nie posiadają jednego zsynchronizowanego zegara, a same ramki danych zawierają informacje synchronizujące. Są nimi bity startu. Dodatkowo w ramce mogą znajdować się dwa lub jeden bit stopu. Opcjonalnym elementem jest bit parzystości, służy on do potwierdzenia poprawności wysłanej informacji. Ilość bitów informacji to najczęściej od pięciu do ośmiu.

Ustawiając komunikacje UART, istnieje możliwość użycia wielu kombinacji podanych wyżej parametrów. Ważne jest, aby w obu mikrokontrolerach wybrać takie same ustawienia. Zdecydowano się na komunikacje zawierającą 8 bitów informacji oraz jeden bit stopu. Wygląd ramki bitów z wybranymi parametrami przedstawiono na rysunku 13. Prędkość wysyłania informacji ustawiono na 115200 bit/s.




**Rysunek 13.** Wizualizacja ramki danych użytej do komunikacji szeregowej

Aby komunikacja była możliwa, połączono pin nadajnika TX wysyłający dane z bitem RX odbiornika oraz odwrotnie. Zastosowano też wspólne zasilanie. Jako piny TX oraz RX na płytce DevKit wybrano te należące do UART2. Nie jest on wykorzystywany do wysyłania informacji systemowych, a jedynie do komunikacji z drugim mikrokontrolerem. Za to w ESP 32 CAM wybrano port domyślny. Umożliwia to ewentualne przesyłanie informacji o błędach.

Komunikacja została zaimplementowana, aby mikrokontroler obsługujący główny panel sterujący mógł wysłać polecenia do ESP 32 CAM. Możliwe komendy zostały przedstawione w tabeli 3. Kod obsługujący komunikacje znajduje się na ESP 32 CAM. Jego zadaniem jest w przypadku dostępności portu szeregowego, odczytać oraz zapisać odebrane polecenia.

| Treść polecenia | Opis |
|---|---|
| On | Włączenie wbudowanej diody LED |
| Off | Wyłączenie wbudowanej diody LED |
| Gray | Ustawienie obrazu w skali szarości |
| No Effect | Przywrócenie obrazu kolorowego |
| VGA | Ustawienie rozdzielczości (640 × 480) |
| HVGA | Ustawienie rozdzielczości (480 × 320) |
| QualityXX | Ustawienie jakości. |

**Tabela 3.** Polecenia wysyłane przez UART

Następnie komendy są rozpoznawane i wykonywane. Dodatkowo w przypadku jakość obrazu wydzielania jest liczba następująca po poleceniu.

## 5.5 Komunikacja wi-fi

Komunikacja z użytkownikiem poprzez panel sterujący została zrealizowana w całości na mikrokontrolerze ESP 32 DevKit. Znajduje się na nim kod tworzący główną stronę internetową. Jest on zrealizowany z pomocą biblioteki ESPAsyncWebServer na podstawie podanych przez twórcę przykładów.

Utworzony serwer jest asynchroniczny, oznacza to, że w przypadku żądania informacji nie czeka on na odpowiedź, ale kontynuuje przetwarzanie strony. Sprawdza się to w przypadku stron posiadających elementy, których aktualizacja zabiera dużo czasu. Dzięki temu strona może być w pełni funkcjonalna mimo to, że wysłała żądanie o długim czasie realizacji.

Utworzenie komunikacji rozpoczyna się od połączenia z lokalną siecią wi-fi. Odbywa się to w taki sam sposób, jak opisano przy poprzednim mikrokontrolerze. Założenie serwera wykonane zostało za pomocą kodu przedstawionego na listingu 1.

```
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html, processor);
  });
```

**Listing 1.** Funkcja założenia serwera głównego

Kod otwiera serwer o adresie wybranym przez ruter. Wykonuje to za pomocą funkcji HTTP_GET. Następnie definiuje sposób odpowiedzi na zapytanie serwera. W tym przypadku jest to wysłanie tekstu zawierającego kod strony z funkcji index_html. Funkcja ta jest surowym łańcuchem znaków znanym z języka C++. Znajduję się tam kod html, css oraz javascript, który w całości przekazywany jest do serwera. To on odpowiada za wygląd strony. Funkcja procesor zapewnia aktualizacje danych strony w przypadku jej odświeżenia.

Oprócz głównego serwera utworzone zostały podstrony obsługujące zdarzenia z głównego panelu. Przykład umieszczony na listingu 2.

```
server.on("/distance", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", Front.c_str());
  });
```

**Listing 2.** Funkcja obsługi serwera do przesyłania pomiarów

W ten sposób zrealizowana jest podstrona o ścieżce /distance. W przypadku żądania jest na niej umieszczony tekst ze zmiennej Front, w której przechowywana jest odległość zmierzona za pomocą przedniego sensora.

Możliwe jest również pobranie danej z serwera za pomocą parametru, czyli zapytania stworzonego przez stronę. Przekazuje ono informacje za pomocą adresu URL. Przykład takiej funkcji jest zamieszczony na listingu 3.

```
server.on("/action", HTTP_GET, [] (AsyncWebServerRequest *request) {
  String inputMessage;
  if (request->hasParam("button")) {
    inputMessage = request->getParam("button")->value();
    button = inputMessage;
    move(button);
  }
  else {
    inputMessage = "No message sent";
  }
  request->send(200, "text/plain", "OK");
});
```

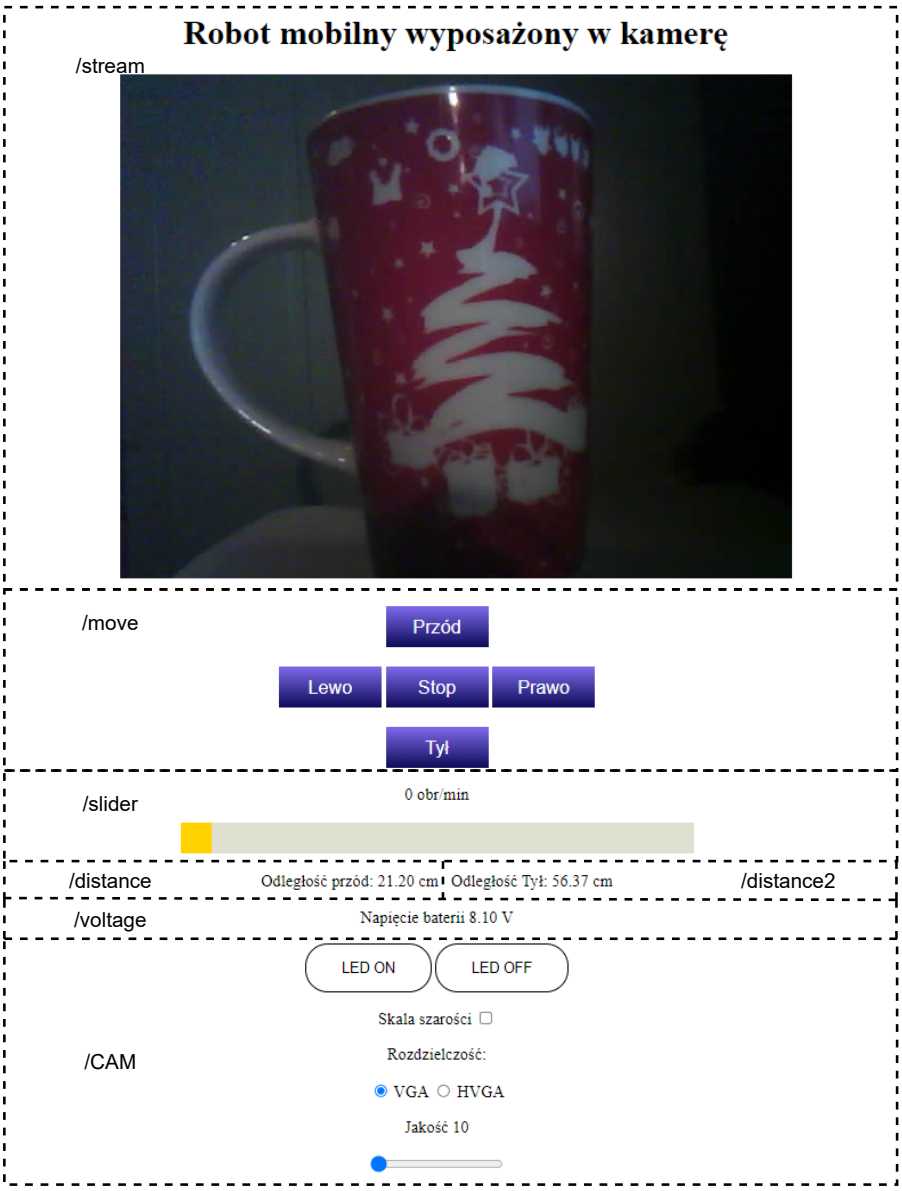
**Listing 3.** Funkcja obsługi serwera do sczytywania zachowań

W podanym przykładzie parametr button posiada wartość, która jest przekazywana do funkcji move. Odpowiada ona za wybór kierunku ruchu robota. W przypadku powodzenia funkcji do serwera zwracana jest kod wartość 200 OK, oznaczający bezbłędne wykonanie polecenia.

W oprogramowaniu zostały również utworzone podstrony /CAM obsługujące polecenia dotyczące kamery, /slider obsługujący suwak zadający prędkość pojazdu oraz /voltage odpowiadające za wyświetlanie zmierzonego napięcia. Obraz z kamery jest pozyskiwany z serwera założonego na mikrokontrolerze ESP 32 CAM.

## 5.6 Działanie interfejsu

Utworzony interfejs sterujący znajduje się na rysunku 14. Rysunek zawiera również informacje, jaka podstrona obsługuje dany element. Obszary te są zaznaczone przy pomocy przerywanej linii. Pierwszą podstroną, która widnieje na rysunku, jest /stream. Jej pełen adres zależy od IP




**Rysunek 14.** Interfejs komunikacyjny z dodanymi informacjami o podstronach

wybranego przez ruter dla ESP 32 CAM. Jest ona wyświetlana na stronie dzięki językowi html za pomocą polecenia <img>. Jako referencja do obrazu została podana ścieżka do streama.

Wszystkie pozostałe adresy są już stworzone przez ESP 32 DevKit. Sposób ich ogólnej obsługi został przedstawiony w poprzednim podrozdziale. Podstrony /move, /slider oraz /CAM

wysyłają do mikrokontrolera dane wybrane przez użytkownika. Mogą nimi być komunikaty o kliknięciu przycisku lub wartości ustawione za pomocą suwaków. Przykładem takiego zapytania może być /CAM?button=quality12, obsługujący ustawienie jakości obrazu na drugim suwaku. Podstrony /distance, /distance2 oraz /voltage służą do odbierania oraz przechowywania informacji pomiarowych z mikrokontrolera.

Strona główna za pomocą funkcji w języku JavaScript kontaktuje się z podanymi adresami wysyłając do nich lub odczytując wartości. W przypadku przycisków oraz suwaków są to funkcje reagujące na zdarzenie. Za to w przypadku wyświetlania pomiarów używane są funkcje cykliczne.

## 5.7 Sterowanie ruchem

Aby wprawić w ruch robota należy wysłać odpowiednie sygnały do sterownika silników. Dla każdego z napędów muszą to być:

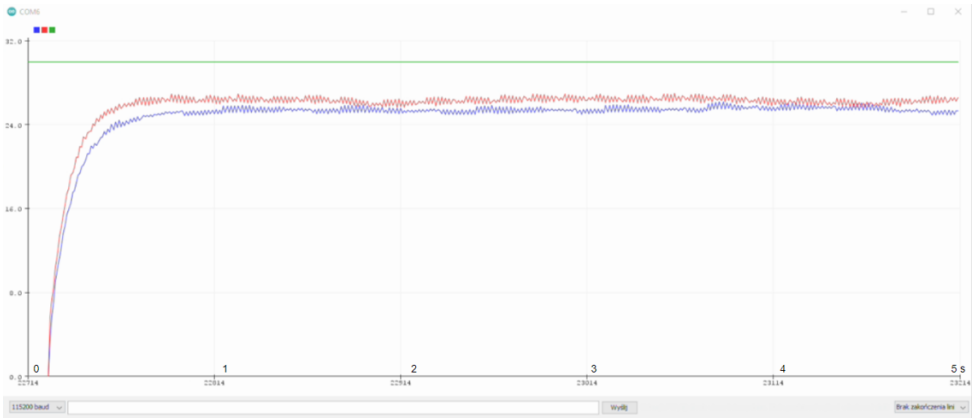
- dwa sygnały definiujące kierunek ruchu,
- sygnał PWM zadający wartość napięcia dostarczanego do silników.

PWM jest generowany sprzętowo za pomocą zegarów na mikrokontrolerze. W jego działaniu istotne jest wypełnienie sygnału. Wpływa ono bezpośrednio na podawane napięcie do silników. Na przykład wysyłając sygnał o wypełnieniu 75% na sterownik dołączony do baterii o napięciu 8 V, przekazuję on napięcie o wartości 6 V na zaciski silnika. Jest to zależność liniowa.

Ustalanie prędkości docelowej silnika oraz kierunku jazdy zrealizowane jest przez obsługę polecenia z panelu w przeglądarce. Sterowanie prędkością odbywa się poprzez regulator. Może on zmieniać napięcie dostarczane do silników, jednak nie ma on wpływu na kierunek obrotu. Za maksymalną wartość wypełnienia PWM wybrano 90%. Jest ona dobrana tak, aby przy pełnym naładowaniu baterii napięcie podawane na silniki nie przekroczyło maksymalnego napięcia pracy.

### 5.7.1 Sterowanie bez użycia regulatora

Sterowanie bez regulatora jest niezwykle uproszczone. Nie wymaga ono analizy ruchu robota, przez co dobrze nadaje się do prostych projektów. Jednak posiada ono wiele wad, a przede wszystkim powoduje duży błąd jazdy po linii prostej. Spowodowane jest to tym, że mimo podawania jednakowych wartości napięcia, silniki nie obracają się w taki sam sposób. Można to zauważyć na rysunku 15. Przedstawia on wykres prędkości obu użytych silników przy podaniu takiej samej wartości wypełnienia sygnału PWM. Przykład ten pokazuje, że prędkości obu silników nie równoważą się.

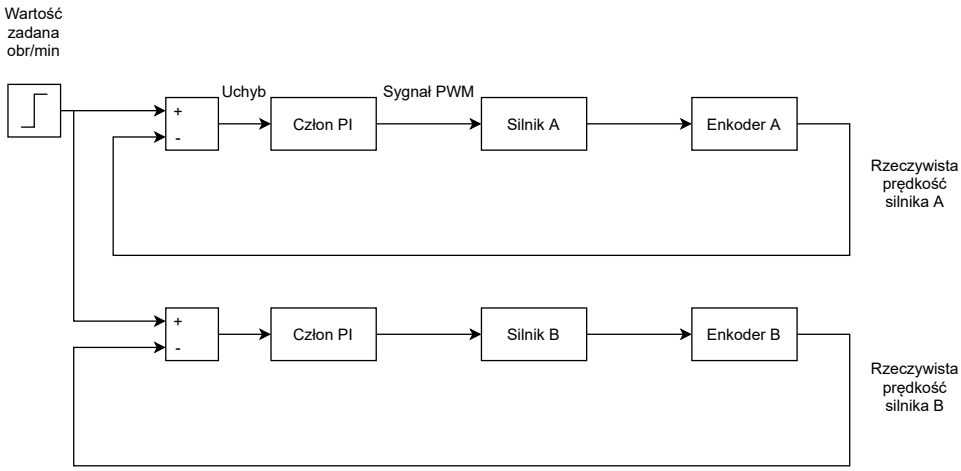



**Rysunek 15.** Przebieg przy sterowaniu bez użycia regulatora

Przyczyną tego zjawiska może być nierówne rozłożenie masy oraz nie idealność w produkcji silników. Dodatkowo ustalenie jednej wartości zadanej sygnału PWM nie powodowałoby jazdy zawsze z taką samą prędkością. Jest to spowodowane faktem, że napięcie baterii spada wraz z jej rozładowaniem.

### 5.7.2 Sterowanie z użyciem regulatora PI

Regulator proporcjonalnie całkujący jest sposobem regulacji z użyciem pętli sprzężenia zwrotnego. Odpowiada on za zmniejszenie różnicy pomiędzy wartością zadaną a rzeczywistą. Sprzężenie realizowane jest w robocie za pomocą pomiaru prędkości z użyciem enkoderów. W celu realizacji sterowania zaimplementowano układ regulacji, którego schemat pokazany jest na rysunku 16. W przypadku tej metody sterownia oba silniki otrzymują prędkość zadaną




**Rysunek 16.** Podstawowy układ sterowania silników

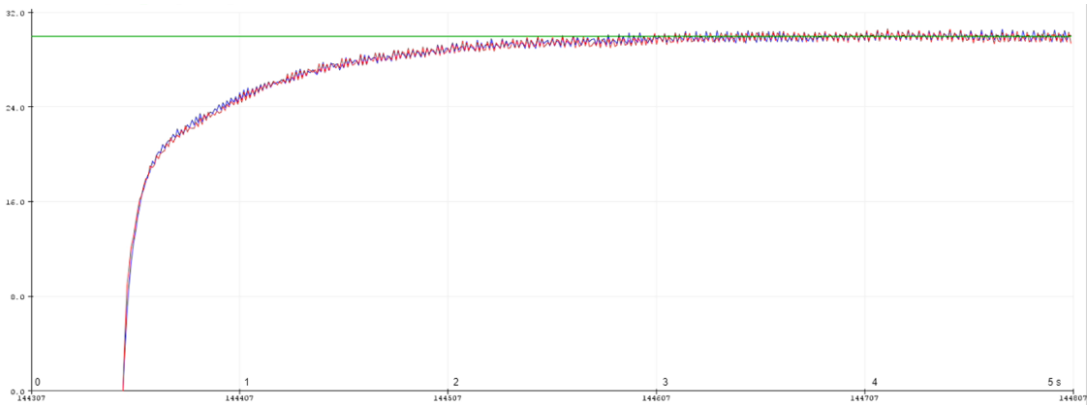
jako tą wybraną przez użytkownika. Wartość wyjścia członu PI w obu pętlach sterowania obliczana jest na podstawie wzoru.

$$\mathrm{u} = \mathrm{Kp} * \mathrm{e} + \mathrm{Ki} \int \mathrm{edt}$$

Gdzie e jest wartością uchybu równą różnicy wartości zadanej oraz rzeczywistej prędkości w obrotach na minutę. Kp oraz Ki są wzmocnieniami członów odpowiednio proporcjonalnego, oraz całkującego, a dt jest czasem pomiędzy kolejnymi iteracjami regulatora.

Wywoływanie poszczególnych okresów działania regulatora jest zrealizowane za pomocą zainicjowanego w kodzie timera, który wywołuje przerwania o częstotliwości 1000 Hz. Jednak sama obsługa regulatora wykonana jest w pętli głównej. Takie zastosowanie jest zrealizowane, aby możliwie maksymalnie skrócić czas działania przerwań.

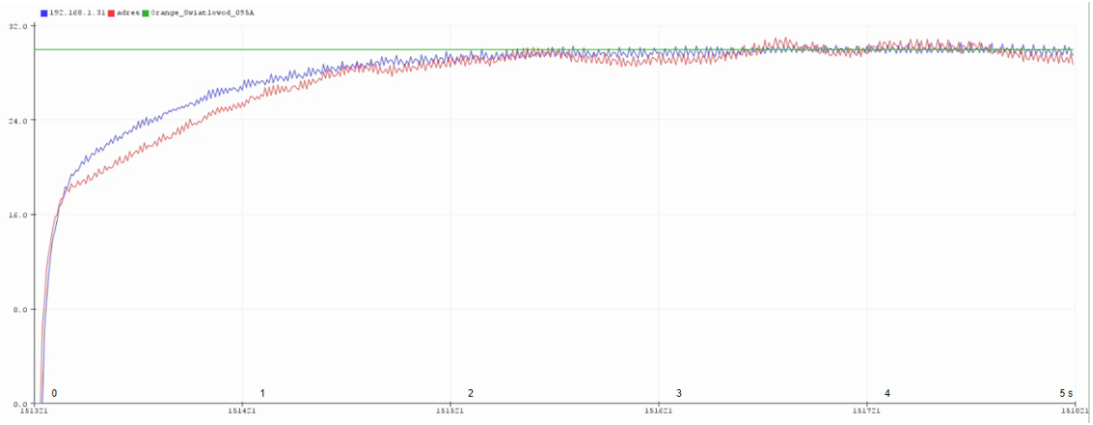
Po implementacji regulatora zostały wykonane testy dla różnych nastaw na nieobciążonych silnikach. Przykładowy przebieg znajduje się na rysunku 17. Przedstawia on zależność prędkości od czasu dla zaprojektowanego układu. Wartość zadana prędkości wynosiła 30 obr/min.




**Rysunek 17.** Przykładowy przebieg prędkości silników z regulatorem

Przebieg został wykonany dla parametrów Kp=10.0, Ki=24.0, Kp2=10.0, Ki2=24.0 Gdzie Kp oraz Ki to wzmocnienia członów regulatora dla silnika A, a Kp2 oraz Ki2 to wzmocnienia dla silnika B.

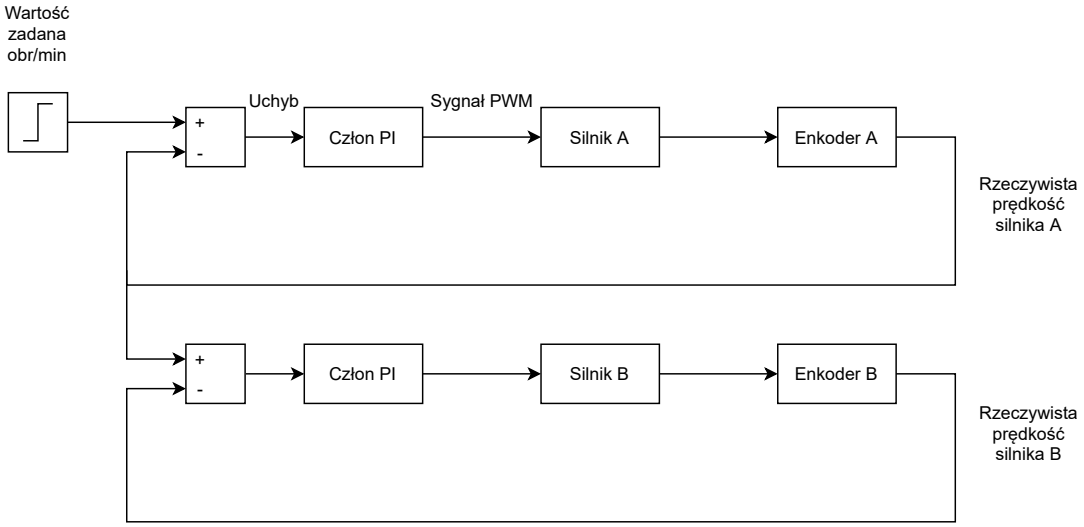
Oglądając wykres, można wnioskować, że silniki są idealnie synchronizowane. Jednak w przypadku dodania na jeden z napędów niewielkiego obciążenia przestają one obracać się w ten sam sposób. Jest to pokazane na rysunku 18. W czasie ruchu silnik oznaczony kolorem czerwonym był lekko spowalniany palcem. Z chęci wyeliminowania pokazanego problemu zdecydowano się na utworzenie regulatora synchronizującego oba koła.




**Rysunek 18.** Przykład wady pierwszego regulatora

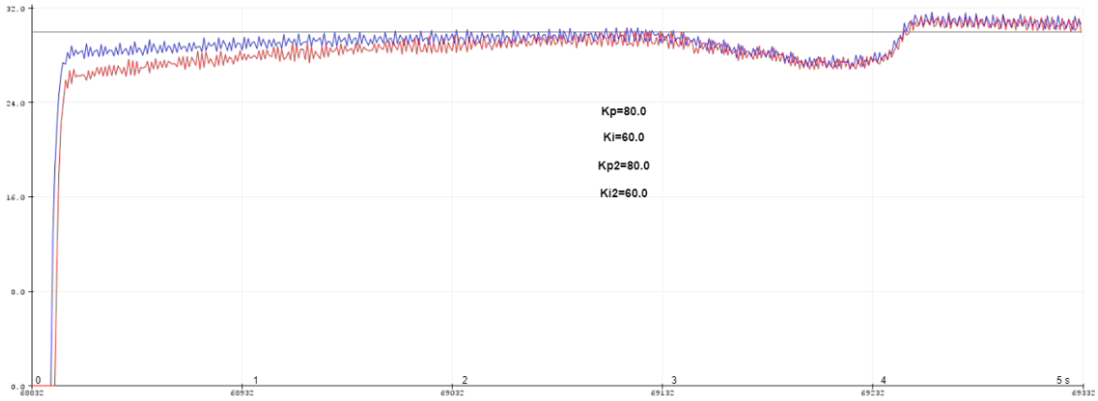
### 5.7.3 Sterowanie z użyciem regulatora PI z synchronizacją kół

W celu utworzenia alternatywnego sposobu sterowania zrealizowano w kodzie robota pętle sterowania ukazaną na rysunku 19. Od razu można zauważyć, że sposób sterowania obu



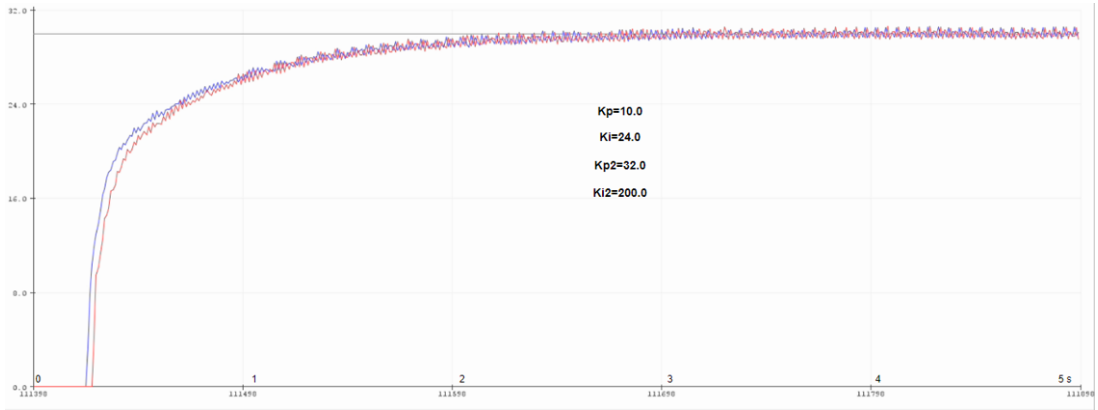

**Rysunek 19.** Układ sterowania silników z synchronizacją kół

silników jest inny. Silnik A jako wartość zadaną otrzymuje prędkość wybraną przez użytkownika. Za to drugi silnik ma za zadanie zbliżyć się jak najbardziej do aktualnej prędkości silnika pierwszego. Przykładowy wynik oraz wybrane parametry wzmocnień ukazane są na rysunku 20. Analizując przebieg można zauważyć, że oba silniki nie startują jednocześnie. Dodatkowo silnik oznaczony kolorem czerwonym osiąga mniejsze prędkości przez pierwsze 2 sekundy ruchu. Jest to spowodowane tym, że druga pętla regulacji jest opóźniona względem pierwszej. W 3 sekundzie ruchu na silnik pierwszy zostało dodane niewielkie obciążenie. Nie skutkowało to




**Rysunek 20.** Przebieg prędkości silników z regulatorem PI z synchronizacją kół

jednak desynchronizacja ruchu. W celu minimalizacji początkowego błędu zmieniono nastawy regulatorów. Dzięki temu otrzymano przebieg ukazany na rysunku 21.

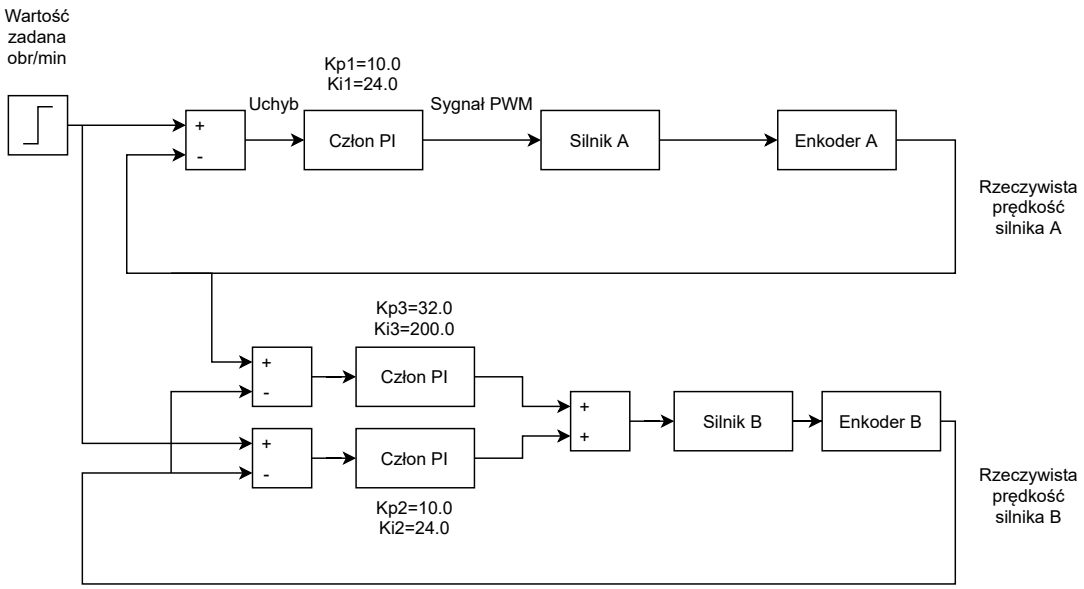



**Rysunek 21.** Przebieg prędkości silników z regulatorem PI z synchronizacją kół ze zmienionymi parametrami

Zmniejszając parametry nastaw regulatora pierwszego oraz zwiększając drugiego, udało otrzymać się bardziej zbliżony sposób rozruchu. Czas ustalania się prędkości zadanej jest o wiele dłuższy niż poprzednio, skutkuje to wolniejszym ustaleniem prędkości silników. Z powodu niższego przyśpieszenia silnika pierwszego, drugi silnik był w stanie wyrównać prędkość w czasie już około 1 sekundy. Wysoka wartość Ki2 dotyczącego części całkującej drugiego silnika dodatkowo przyspiesza synchronizacje. Zapewnia ona również mniejsze odchylenia prędkości przy dodaniu obciążenia. Jednak nie jest to przebieg idealny. Nadal występuje opóźnienie przy starcie pojazdu. Z tego powodu zdecydowano się na inne rozwiązanie problemu.

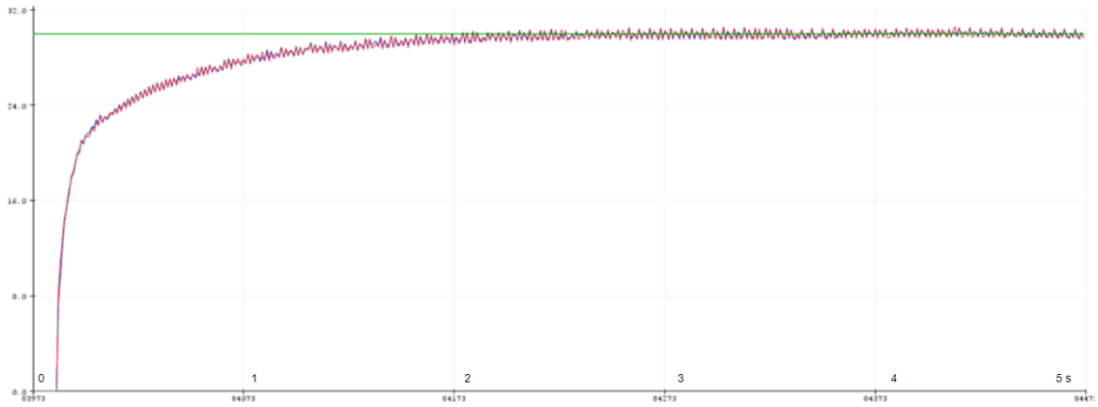
### 5.7.4 Sterowanie z użyciem podwójnego regulatora PI z synchronizacją kół

Łącząc budowę dwóch poprzednich sposobów regulacji, otrzymano schemat pokazany na rysunku 22. W podanej pętli sterowania zmianie ulega sposób regulacji silnika B. Wartość



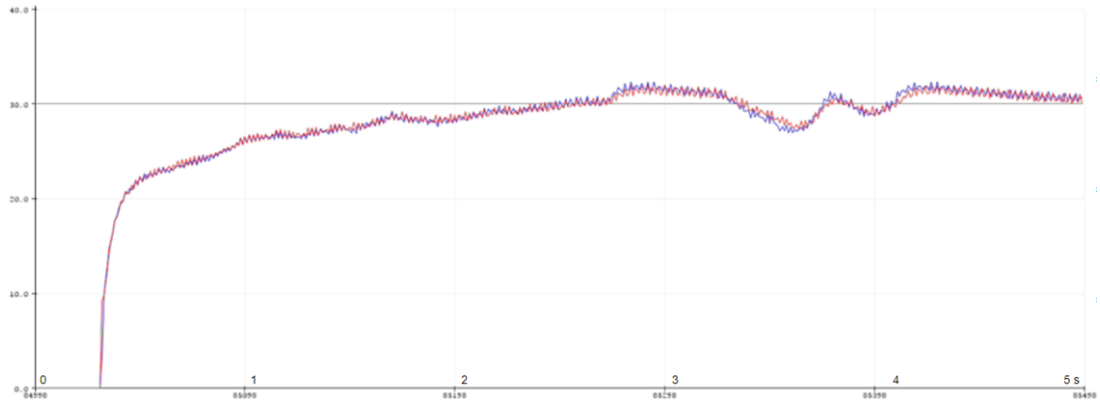

**Rysunek 22.** Podwójny układ sterowania silników

napięcia podawanego zależy od sumy wyjścia dwóch regulatorów. Jednego ze stałą wartością zadaną oraz drugiego synchronizującego. Istotne są również proporcje nastaw układu. Zostały one zrealizowane tak, aby największy wpływ miała całka członu synchronizującego prędkość. Używając opisanego regulatora, otrzymano wyniki przedstawione na rysunkach 23 i 24.




**Rysunek 23.** Przebieg dla podwójnego regulatora bez obciążenia silników

Jak widać na obu przebiegach udało się pozbyć opóźnienia w starcie silników. Dodatkowo




**Rysunek 24.** Przebieg dla podwójnego regulatora z obciążeniem silników

zachowano przy tym ich synchronizacje. Przebieg z obciążeniem jest nierównomierny, ponieważ silnik był spowalniany w sposób losowy.

## 5.8 Obsługa enkoderów

Enkodery wybrane do robota są modelami magnetycznymi. Oznacza to, że wykrywają obracający się magnes na dodanej do wału tarczy. Wysyłają one sygnał prostokątny w zależności od jego położenia. Każdy z enkoderów posiada dwie fazy. Wspólnie pozwalają one określić kierunek obrotu. Czas pomiędzy kolejnymi iteracjami sygnałów umożliwia obliczenie prędkości obrotowej. W zrealizowanym robocie wykorzystano tylko jedną fazę. Jest to możliwe, ponieważ kierunek obrotu jest z góry nałożony i nie trzeba go wyznaczać. Obsługa sygnału odbywa się za pośrednictwem zewnętrznych przerwań wykrywających zbocze narastające. W samych przerwaniach zapisywany jest czas między kolejnymi zboczami. W pętli głównej obliczana jest prędkość silników za pomocą wzoru.

$$\frac{1}{\mathrm{deltaT[min]}} * (\text{impulsy na obrót} * \text{ przełożenie}) = \mathrm{V}[\frac{\mathrm{obr}}{\mathrm{min}}]$$

Gdzie deltaT to czas pomiędzy kolejnymi impulsami. Przełożenie jest uwzględnione we wzorze, ponieważ enkodery znajdują się na silniku przed przekładnią.

## 5.9 Pomiar napięcia

Pomiar napięcia na mikrokontrolerze zrealizowany jest za pomocą przetwornika analogowo-cyfrowego. Jest on 12-bitowy, co oznacza, że jego maksymalny wynik pomiaru może wynosić 4095. W celu obliczenia wartości napięcia baterii pomiar wykonany przez przetwornik zostaje

ze skalowany do maksymalnego napięcia wynoszącego 3,3 V. Potem dodawany jest stały błąd pomiaru wynikający z nieliniowości przetwornika. Aby ostatecznie otrzymać napięcie baterii, wynik wymnażany jest przez proporcje dzielnika napięcie. Zrealizowane jest to w kodzie za pomocą wzoru 4.Ostatecznie zgodnie z propozycją producenta obliczana jest średnia z trzech ostatnich pomiarów [4].

```
v_in = ((analogRead(v_baterry)/4095.0)*3.3+0.25)*4.9;
```

**Listing 4.** Sposób otrzymtwania napięcia baterii

Pomiar napięcia wywoływany jest przez funkcje cykliczne obsługujące serwer /voltage. Powoduje to obliczanie tej wartości jedynie w przypadku użycia panelu kontrolnego przez użytkownika.

## 5.10 Funkcje zabezpieczające

Programując robota, zamieszczono szereg funkcji zabezpieczających przed jego kolizją z przeszkodami. Są to funkcje zatrzymywania z panelu kontrolnego, pomiar odległości od przeszkód oraz wykrywanie utraty połączenia wi-fi.

### 5.10.1 Bezpieczne działanie panelu kontroli ruchu

Panel kontroli kierunku ruchu jest zrealizowany tak, aby w przypadku puszczenia przycisku odpowiadającego za kierunek jazdy robot od razu stanął. Umożliwia to lepszą kontrolę nad jazdą. W zrealizowanym przypadku przycisk stop jest zbędny i pełni tylko funkcję zabezpieczającą w przypadku awarii.

### 5.10.2 Pomiar odległości

Pomiar odległości zrealizowany jest przez dwa sonary HC-04 mierzące dystans do przeszkód z tyłu i przodu robota. Pomiar na nich wykonany jest zgodnie z propozycją podaną przez producenta. Zdecydowano się na mierzenie odległości co 0,5 s. W celu zapewnienia niezawodności działania postanowiono zrealizować odmierzanie czasu za pomocą przerwań timerów, a nie funkcji cyklicznych panelu kontrolnego. Dzięki temu czas realizacji pomiaru jest niezależny od obciążenia strony internetowej.

Na podstawie zmierzonej odległości robot decyduje o ewentualnym zatrzymaniu. Odległość krytyczna została wyznaczona na podstawie drogi, którą pojazd może przejechać w 0,5 s z maksymalną prędkością, wynoszącą 70 obr/min. Wynosi ona 11,9 cm. Zdecydowano się

więc na ustawienie wartości krytycznej na 20 cm. Przy przekroczeniu tej odległości robot jest zatrzymywany a ruch w danym kierunku blokowany.

### 5.10.3 Wykrywanie utraty połączenia wi-fi

Funkcja wykrywania utraty połączenia została zrealizowana na podstawie biblioteki WiFi.h. W przypadku zerwania połączenia robot jest od razu zatrzymywany. Stara się on następnie wznowić utraconą łączność.

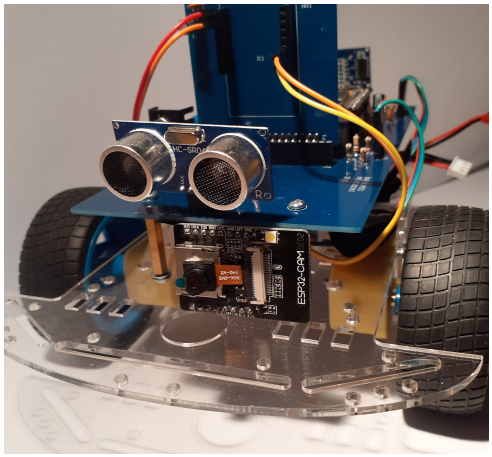
# Rozdział 6

# Uruchomienie i testy

Po zaprogramowaniu mikrokontrolerów oraz zlutowaniu całego układu robot został uruchomiony. Niestety nie wypadł on w pierwszych testach dobrze. Problemem była bardzo niewielka wydajność kamery. Obraz z niej otrzymywany był wyraźny, lecz mało płynny.

W celu eliminacji problemu doszukiwano się błędów. Nie znaleziono ich w użytej metodzie zasilania, ponieważ reszta elementów działała poprawnie, a sam mikrokontroler posiadał kondensatory filtrujące. Wgrany kod również nie stanowił problemów, ponieważ ESP32 CAM umieszczony na płytce stykowej działał poprawnie.

Udało się rozwiązać zaistniały problem poprzez połączenie przewodami tylko niektórych wyprowadzeń płytki oraz przeniesienie jej do innej części robota. Sposób tego wykonania został pokazany na rysunku 25.



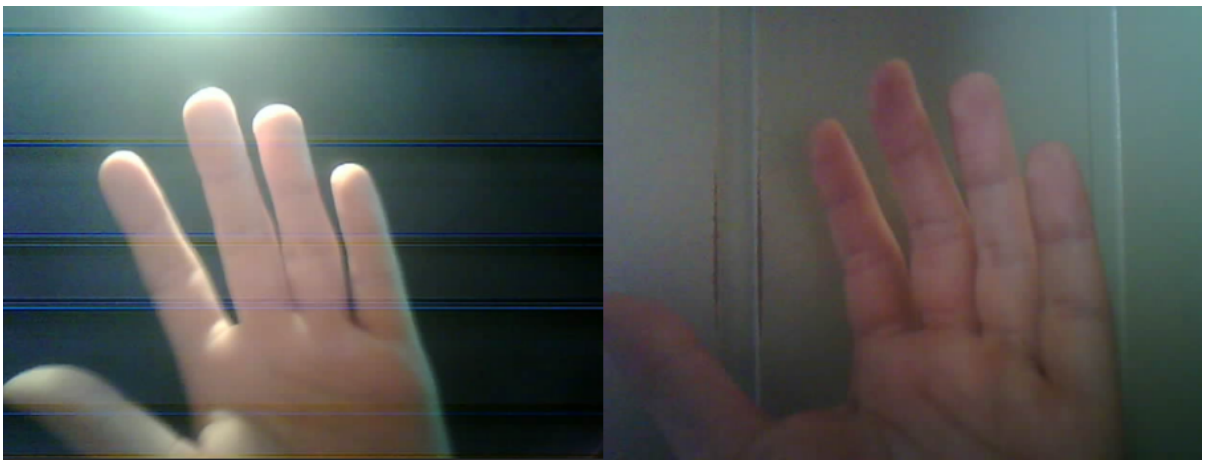

**Rysunek 25.** Dokonana modyfikacja

Najprawdopodobniej problem wynikał z konstrukcji płytki PCB służącej do umieszczenia kamery. Wyprowadzono w niej wszystkie dostępne połączenia elektryczne z mikrokontrolera do

złącza umieszczonego na drugiej płytce PCB. Mimo tego, że płytka pomyślnie przeszłą test na brak zwarć, nieużywane wyprowadzenia nadal mogły generować niepożądane zakłócenia.

## 6.1 Ocena pojedynczej klatki obrazu

Na wygląd obrazu mocno wpływa jakość dostarczonego zasilania. Na rysunku 26 widnieje porównanie dwóch klatek z kamery. Jedno ze zdjęć zostało wykonane przy zasilaniu z USB, a drugie w przypadku podłączenia do wykonanej płytki PCB. Test zrealizowano dla rozdzielczości VGA z najwyższą dostępną jakością.



**Rysunek 26.** Porównania dwóch sposobów zasilania. USB (lewo) zbudowany układ (prawo)

W przypadku zasilania z USB na obrazie widnieją błędy w postaci poziomych pasków. Korzystając z własnego sposobu zasilania, udało się wyeliminować te defekty. Ponadto obraz jest czytelny, mimo że widać na nim pojedyncze piksele.

## 6.2 Działanie efektów kamery

Modyfikacje obrazu kamery z panelu kontrolnego działają bez zarzutów. Oznacza to, że zastosowana komunikacja między dwoma kontrolerami jest poprawna. Przykłady niektórych modyfikacji ukazane są na rysunku 27.

Ustawienie obrazu czarno-białego lub w niższej jakości w niewielkim stopniu zwiększa ilość klatek widzianych na streamie. Jest to efekt zamierzony przy tworzeniu dostępu do tych opcji. Jednak poprawa jest na tyle niewielka, że raczej nie zachodzi potrzeba jej używania.

**Rysunek 27.** Ukazanie możliwych modyfikacji obrazu. Bez efektów (lewo), obraz czarno biały (środek), obraz o zmniejszonej jakości (prawo)

## 6.3 Test jazdy po linii prostej

Z racji, że robot nie posiada funkcji obrotu o dany kąt, niemożliwe było wykonanie testu jazdy po kwadracie. Z tego powodu przygotowano tor do jazdy po linii prostej. W czasie testu robot ustawiany był na początku trasy i miał przejechać dystans 385 cm. Po pokonaniu drogi mierzona była odległość, na której robot skończył jazdę w stosunku do końca wyznaczonej prostej linii.

Pierwsze testy zostały wykonane dla podstawowego sterowania opisanego w rozdziale 5.7.2. Wykazały one duży błąd, który średnio wynosił 43 cm. W dodatku wstępowały duże odchylenia pomiarów od uzyskanej średniej wyników. Taki sam efekt przyniosły również inne regulatory.

Spodziewano się, że jest to spowodowane błędem w konstrukcji robota. W celu jego eliminacji postarano się o stabilniejsze ustawienie silników, jednak nie przyniosło to skutków. Ostatecznie wymieniono koła na inny dostępny model. Skutkowało to znacznym zmniejszeniem błędów i większą powtarzalnością przeprowadzonych testów. Najprawdopodobniej błędy wynikały z ich niedokładnej konstrukcji. Wyniki zostały przedstawione w tabeli 4.

Pomiary zostały podzielone ze względu na zastosowane sterowanie silnika B. Sposób regulacji silnika A zawsze pozostaje bez zmian. Użyte metody są przedstawione w rozdziale 5.7. Parametry regulatorów powtarzają się w każdej z prób.

Wyniki mogą być obarczone błędem w postaci nieidealnego ustawienia robota na starcie trasy. Z tego powodu w każdym przypadku zastosowano średnią z 5 przejazdów. Każdy otrzymany wynik był odchyleniem od toru jazdy w prawą stronę. Działo się to nawet w przypadku zamiany sterowania silnika A i B ze sobą. Odchylenie to nie wynika więc z błędów programistycznych. Najprawdopodobniej było spowodowane nie idealnością w konstrukcji robota.

Dodatkowo w ramach porównania wykonano test jazdy bez żadnego regulatora ze stałą wartością sygnały PWM. W każdym przypadku tak zaprogramowany robota ostro skręcał

| **Zastosowany regulator dla silnika B** | **Odchylenie od prostej dla prędkości 30 obr/min [cm]** | **Odchylenie od prostej dla prędkości 60 obr/min [cm]** | **Średnia dla 30 obr/min [cm]** | **Średnia dla 60 obr/min [cm]** |
|---|---|---|---|---|
| **Podstawowy regulator PI** | 25 | 22 | 27,8 | 28,6 |
| | 22 | 33 | | |
| | 26 | 27 | | |
| | 34 | 35 | | |
| | 32 | 26 | | |
| **Regulator tylko z synchronizacją z silnikiem A** | 31 | 29 | 34,2 | 38,6 |
| | 47 | 40 | | |
| | 38 | 39 | | |
| | 32 | 36 | | |
| | 23 | 49 | | |
| **Podwójny regulatora PI z synchronizacją kół** | 21 | 11 | 19 | 14 |
| | 15 | 18 | | |
| | 25 | 19 | | |
| | 15 | 8 | | |
| | 19 | 14 | | |

**Tabela 4.** Tabela przedstawiająca wyniki testu jazdy robota

w prawą stronę. Działo się to do tego stopnia, że na dystansie 2 metrów pojazd zatrzymywał się na ścianie oddalonej od wyznaczonej linii o 1,5 metra.

Biorąc pod uwagę test bez użycia regulatora, każdy spośród użytych regulatorów jest znacznie lepszy niż jego brak. Bez wątpienia najgorszym sposobem regulacji jest regulator z samą synchronizacją. Jest to spowodowany tym, że oba silniki w tym przypadku nie uruchamiają się jednocześnie. Widać to było za każdym razem przy starcie robota. Powodowało to lekkie odchylenie kierunku jazdy już na samym początku. Był to efekt spodziewany oraz widziany już na przebiegu przedstawionym w rozdziale 5.7.3.

Podstawowy regulator PI taki sam dla obu kół robota sprawdził się lepiej niż przewidywano. Najprawdopodobniej podobne warunki jazdy oraz brak zmiennego obciążenia wpłynął pozytywnie na tor jazdy. Najwidoczniej sama stabilizacja prędkości oraz podobne parametry nastaw regulatorów umożliwiają względnie dobrą jazdę.

Najlepiej sprawdził się podwójny regulatora PI z synchronizacją kół, chociaż nie jest to wynik bezsporny. Mimo najmniejszej średniej błędów wiele prób okazywało podobny skutek jak w poprzedniej opcji. Istnieje możliwość, że dzięki zastosowaniu tego regulatora udało się najbardziej zminimalizować zmiany prędkości wynikające z niedokładności działania silników.

Nie ma jednoznacznych dowodów, aby wartość prędkości obrotowej miała jednoznaczny wpływ na kierunek toru jazdy zbudowanego robota. Przy prędkościach 30 oraz 60 obrotów na minutę obserwowane było zmniejszenie, jak i zwiększenie błędu.

Mimo wszelkich starań nie udało się zaprojektować sposobu sterowanie całkowicie niwelującego błąd jazdy. Jest to spowodowane między innymi nie idealnością pomiaru prędkości oraz budowy kół. Nawet przy idealnej synchronizacji prędkości koła mogą powodować nierówny ruch robota przez występujący poślizg.

Pojazd ma stałą wadę, która powoduje jego skręcanie w prawą stronę. Aby całkowicie ją zniwelować należałoby zastosować inne sterowanie, takie jak jazdę na podstawie obrazu z kamery lub kompasu. Mimo wszystko zastosowane sterowania umożliwiają komfortową jazdę bez potrzeby częstej korekty kierunku.

## 6.4 Test bezkolizyjności

W celu sprawdzenia funkcji awaryjnego zatrzymywania ustawiono naprzeciw robota w odległości półtora metra przeszkodę. Dystans ten wybrano, aby pozwolić pojazdowi maksymalnie się rozpędzić. W celu zwiększenia szansy na wykrycie jako przeszkodę wybrano duże pudło. Prędkość testowa zawsze była wybierana na maksymalną dostępną przez użytkownika w panelu kontrolnym, czyli 0,85 km/h.

W przypadku, kiedy pudło było ustawione pod kątem 90 stopni do robota, był on w stanie zatrzymać się bez zderzenia. Jednak, gdy przestawiono przeszkodę pod kątem 45 stopni, robot w nią uderzał. Dzieje się tak dlatego, że sonary odczytują błędną odległość, gdy przedmiot przed nimi znajduje się pod dużym kątem. W celu eliminacji tego błędu należałoby na przykład zastosować większą ilość sonarów ustawionych pod różnymi kątami w robocie.

## 6.5 Bezpieczeństwo układu elektrycznego

W czasie obsługi robota nie doszło do ani jednego zwarcia więc działanie diody Zenera oraz bezpiecznika nie zostało sprawdzone. Temperatura pracy całego układu jest na tyle niska, że możliwe jest jego dotknięcie bez oparzenia. Dotyczy to też stabilizatora napięcia, mimo że układ ten często mocno się grzeje. Rozległe pole masy oraz radiator wystarczająco obniżają temperaturę układu.

## 6.6 Funkcjonalność robota

Zaprojektowany robot może być bez problemu sterowany z komputera podłączonego do tej samej sieci. Robot ma możliwość działania z kamerą, jak i bez dzięki modułowemu charakterowi zaprojektowanych płytek. Sterowanie nim odbywa się bez większych opóźnień. Dzięki zastosowanym regulatorom PI może on obracać się z niewielką prędkością nawet 4 obr/min co ułatwia jego obracanie. Zasięg sterowania ogranicza się do obszaru pokoju z ruterem. W przypadku przeniesienia robota w inne miejsce występują duże zakłócenia. W celu eliminacji problemu należałoby dołączyć do robota antenę.

Poprawność pomiarowa napięcia baterii waha się w granicach +-200 mV. Mimo to jest to wystarczające, aby określić, kiedy należałoby naładować akumulatory. Napięcie baterii spada w trakcie ruchu robota, co jest normalne z powodu obciążenia zasilania. Wygląd strony jest dość prosty, lecz wykazuje ona pełną funkcjonalność.

# Rozdział 7

# Uwagi i wnioski

Podsumowując prace, udało się zbudować robota wyposażonego w kamerę. Spełnia on główne założenie projektu, czyli płynną transmisje obrazu. Zgodnie z przyjętymi celami jest on sterowany z przeglądarki oraz posiada zabezpieczenia z poziomu elektroniki, jak i oprogramowania. Niestety ochrona przed zderzeniami nie jest w pełni efektywna z powodów ograniczeń sonaru. W pracy udało porównać się różne metody sterowania silnikami. Bynajmniej ich zastosowanie nie za skutkowało pełną eliminacją błędów. Wnioskując z problemów z pierwszą parą kół, duże znaczenie przy wybranym sposobie sterowania ma poprawność konstrukcji mechanicznej pojazdu. Przeprowadzony eksperyment udowodnił wady sterowania bez aktualizacji położenia. W przyszłości, aby lepiej kierować robotem możliwe będzie zastosowanie analizy obrazu z użytej kamery.

W ramach pracy udało się zaprojektować oraz zmontować dwie płytki PCB. Główna z nich działa bez zarzutów, niestety płytka z kamerą okazała się na ten moment wadliwa i konieczna była modyfikacja połączeń. Konstrukcja mechaniczna całego robota okazała się stabilna oraz wytrzymała. Jako dodatkowy aspekt pracy udało się stworzyć komunikacje miedzy mikrokontrolerami za pomocą portu UART.

W przyszłości jako rozwój projektu możliwe będzie założenie własnej strony internetowej za pomocą któregoś z serwisów hostingowych. W ten sposób robot będzie mógł być sterowany z dowolnego miejsca na ziemi. Dawałoby to możliwość obserwacji domu w sposób zdalny. Umożliwiałoby to na przykład wejście w interakcje ze zwierzętami lub sprawdzenie czy żelazko zostało odłączone z kontaktu.

Jako inną drogę rozwoju projektu można przyjąć użycie kamery do autonomicznego ruchu pojazdu. Dzięki obserwacji przekazywanego przez robota obrazu możliwa by była analiza zachowań układu w stosunku do danych, którymi dysponuje. Działoby się to w czasie rzeczywistym, a podgląd zachowania mogłaby ułatwić zaprogramowanie systemu.

# Bibliografia


[1] *1928 - Eric Robot - Capt. Richards A.H. Reffell (English)*, sierp. 2015. adr.: http://cyberneticzoo.com/robots/1928-eric-robot-capt-richards-english/ (term. wiz. 09. 11. 2021).

[2] Cook, D. A., *Budowa robotów dla początkujących*, pol, Wyd. 3. Gliwice: Wydawnictwo Helion, 2016, ISBN: 9788328324695.

[3] Cook, D. A., *Budowa robotów dla średnio zaawansowanych*, pol. Gliwice: Wydawnictwo Helion, 2013, ISBN: 9788324655298.

[4] *Dokumentacja ESP32 Datasheet V3.0*, Espressif Systems, 2019.

[5] *Dokumentacja HC-SR04 Ultrasonic Sensor*, Cytron Technologies, wrz. 2012.

[6] *Dokumentacja OV2640 Color CMOS UXGA*, Omnivision, lut. 2006.

[7] *Dokumentacja Positive voltage regulator ICs*, STMicroelectronics, mar. 2014.

[8] *Dokumentacja Toshiba Bi-CD Integrated Circuit Silicon Monolithic TB6612FNG*, Toshiba, maj 2008.

[9] *Elementy robotyki mobilnej*, pol, ser. Podręczniki Akademickie - Polsko-Japońska Wyższa Szkoła Technik Komputerowych t. 9. Warszawa: Wydaw. PJWSTK, 2004, ISBN: 8389244160.

[10] Filipek, P., Kamiński, T., Czuryło, K. i Piwko, K., „Mobilny robot poszukiwawczo-transportowy", *Autobusy : technika, eksploatacja, systemy transportowe*, t. R. 18, nr 6, s. 658–662, CD, 2017, ISSN: 1509-5878.

[11] Inglis-Arkell, E., *The very first robot "brains" were made of old alarm clocks*, grud. 2015. adr.: https://gizmodo.com/the-very-first-robot-brains-were-made-of-old-alarm-cl-5890771 (term. wiz. 09. 11. 2021).

[12] Pacelt, O., *Roboty wśród nas, roboty W życiu codziennym*, lut. 2021. adr.: https://botland.com.pl/blog/roboty-wsrod-nas-roboty-w-zyciu-codziennym/ (term. wiz. 09. 11. 2021).

[13] Rak, J., *Robot mobilny z zaawansowanym sterowaniem ruchu*, 2014.

[14] SA, W. P., *Definicja pojecia robot*. adr.: https://encyklopedia.pwn.pl/haslo/robot;3968150.html.

[15] Zunt, D., *Who did actually invent the word "robot" and what does it mean?* Adr.: https://web.archive.org/web/20130123023343/http://capek.misto.cz/english/robot.html (term. wiz. 09. 11. 2021).

# Wykaz skrótów i symboli

| | |
|---|---|
| I | Natężenie prądu elektrycznego |
| V | Napięcie prądu elektrycznego, prędkość |
| T | Temperatura |
| P | Moc strat |
| $\mathrm{R_{th}}$ | Rezystancja termiczna |
| M | Moment obrotowy |
| F | Siła |
| r | Promień |
| a | Przyśpieszenie |
| m | Masa |

# Spis rysunków


1 Korpus robota poszukiwawczo-transportowego [10] . . . . . 20
2 Robot G1 Tank Żródło: http://www.yahboom.net/study/G1-T-PI . . . . . 23
3 Zaprojektowany robot z kamerą . . . . . 26
4 Zastosowany moduł z kamerą Żródło: sklep.msalamon.pl oraz allegro.pl 32
5 Płytka ESP 32 DevKit v1 Żródło: allegro.pl . . . . . 34
6 TB6612FNG umieszczony na płytce Pololu 713 Żródło: botland.com.pl . . 36
7 Sposób podłączenia diody Zenera . . . . . 40
8 Schemat blokowy połączeń elektrycznych . . . . . 41
9 Projekt płytki z kamerą wykonany w programie Eagle . . . . . 42
10 Projekt płytki głównej wykonany w programie Eagle . . . . . 43
11 Moduł FTDI232 użyty do programowania ESP 32 CAM Żródło: iot-guider.com . . . . . 46
12 Schemat blokowy kodu obsługi kamery . . . . . 47
13 Wizualizacja ramki danych użytej do komunikacji szeregowej . . . . . 48
14 Interfejs komunikacyjny z dodanymi informacjami o podstronach . . . . . 51
15 Przebieg przy sterowaniu bez użycia regulatora . . . . . 53
16 Podstawowy układ sterowania silników . . . . . 53
17 Przykładowy przebieg prędkości silników z regulatorem . . . . . 54
18 Przykład wady pierwszego regulatora . . . . . 55
19 Układ sterowania silników z synchronizacją kół . . . . . 55
20 Przebieg prędkości silników z regulatorem PI z synchronizacją kół . . . . . 56
21 Przebieg prędkości silników z regulatorem PI z synchronizacją kół ze zmienionymi parametrami . . . . . 56
22 Podwójny układ sterowania silników . . . . . 57
23 Przebieg dla podwójnego regulatora bez obciążenia silników . . . . . 57
24 Przebieg dla podwójnego regulatora z obciążeniem silników . . . . . 58
25 Dokonana modyfikacja . . . . . 61

26 Porównania dwóch sposobów zasilania. USB (lewo) zbudowany układ (prawo) 62
27 Ukazanie możliwych modyfikacji obrazu. Bez efektów (lewo), obraz czarno biały (środek), obraz o zmniejszonej jakości (prawo) . . . . . 63

# Spis tabel


1 Parametry silnika SJ01 . . . . . 27

2 Zestawienie maksymalnego poboru prądu przez układ robota . . . . . 38

3 Polecenia wysyłane przez UART . . . . . 49

4 Tabela przedstawiająca wyniki testu jazdy robota . . . . . 64

# Spis załączników

Załącznik I Ideowy schemat połączeń elektrycznych 79

Załącznik II Schemat płytki głównej 81

Załącznik III Projekt płytki głównej 83

Załącznik IV Zdjęcie płytki głównej 85

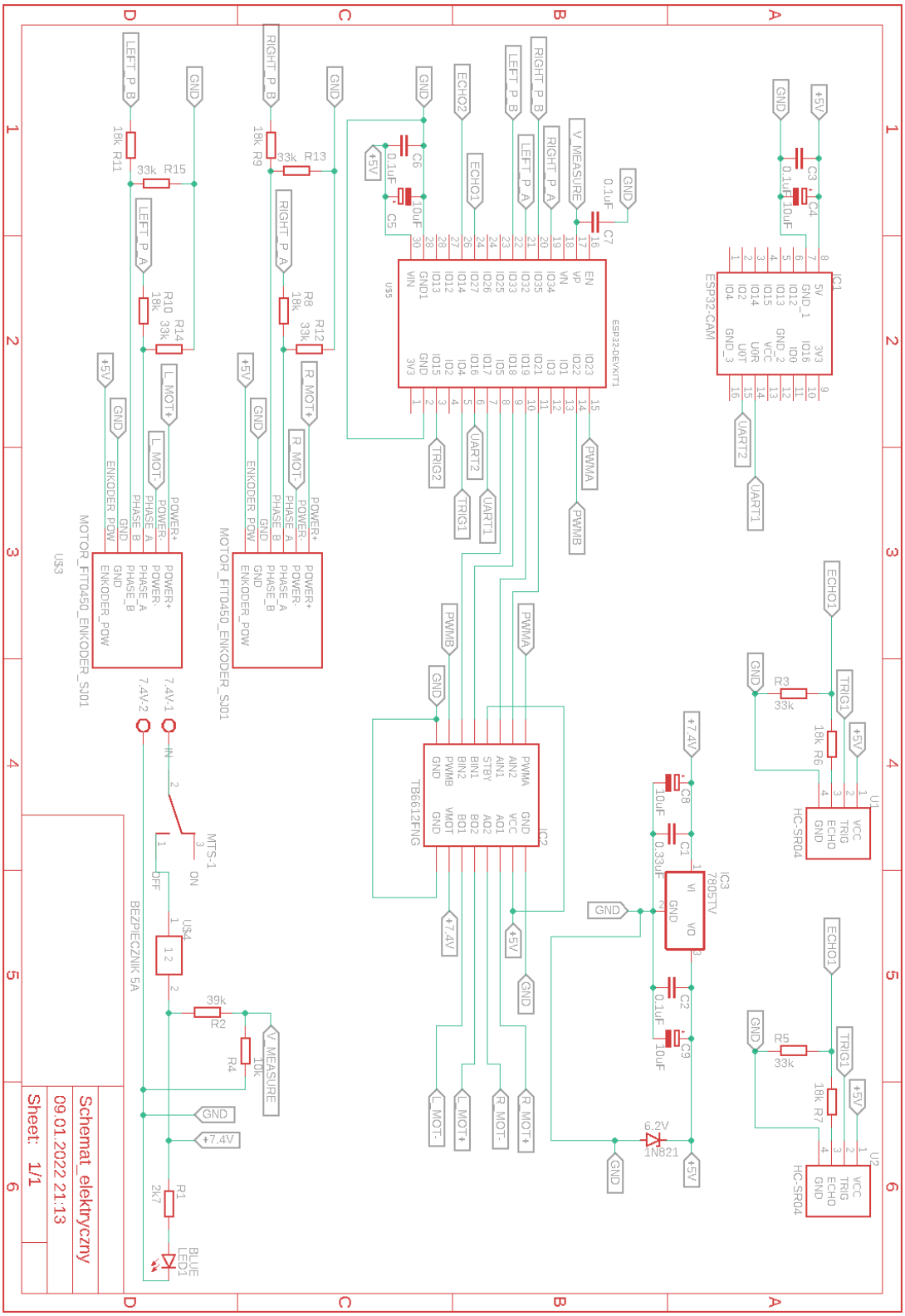
Załącznik V Schemat płytki z kamerą 87

Załącznik VI Projekt płytki z kamerą 89

Załącznik VII Zdjęcie płytki z kamerą 91

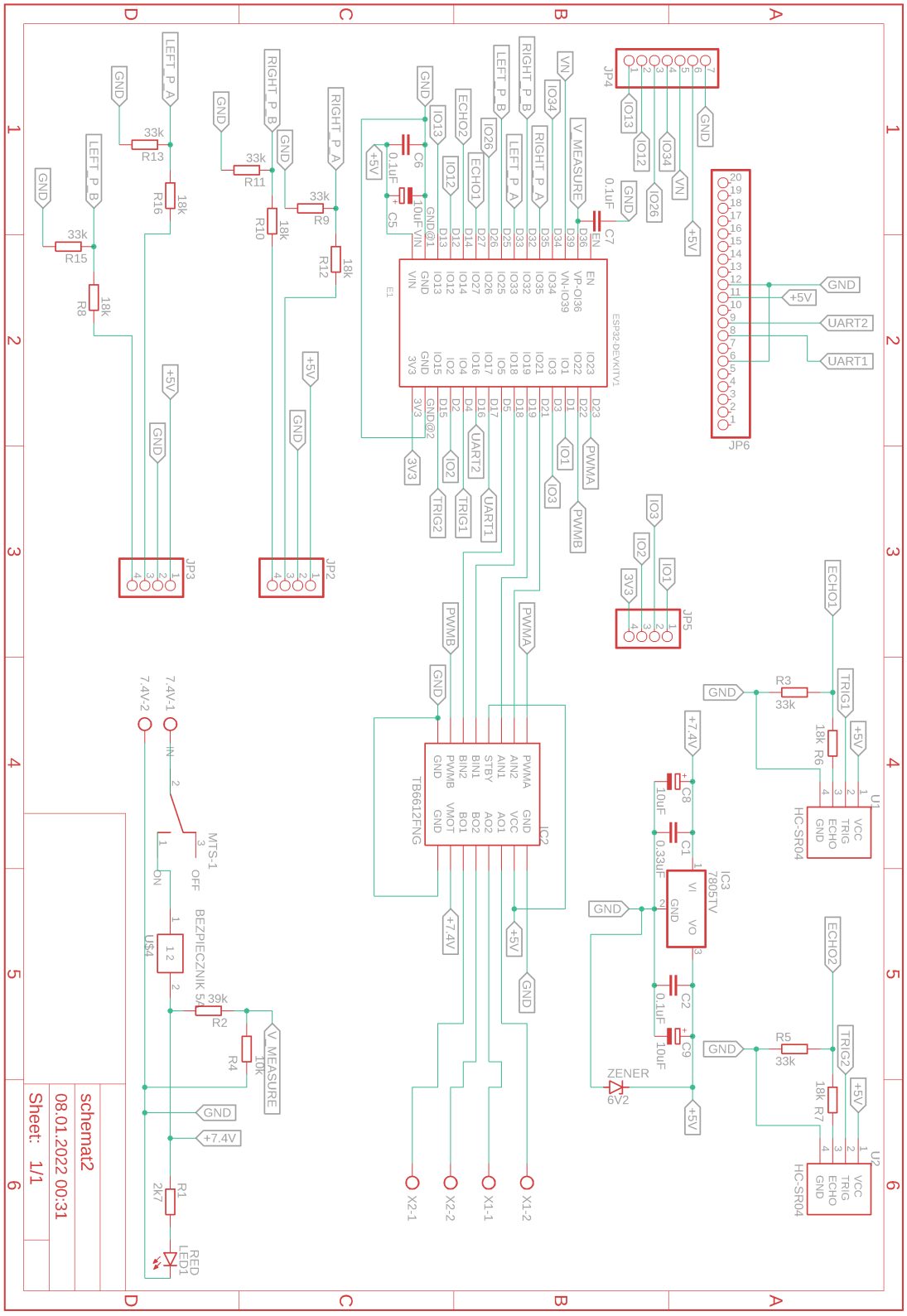
# Załącznik I

# Ideowy schemat połączeń elektrycznych


Schemat_elektryczny
09.01.2022 21:13
Sheet: 1/1
ESP32-CAM
IC1
ESP32-DEVKIT1
U35
HC-SR04
U1
U2
TB6612FNG
IC2
7805TV
IC3
MOTOR_FIT0450_ENKODER_SJ01
U53
U54
BEZPIECZNIK 5A
MTS-1
1N821
BLUE
LED1
R1
2k7
R2
39k
R3
33k
R4
10k
R5
33k
R6
18k
R7
18k
R8
18k
R9
18k
R10
18k
R11
18k
R12
33k
R13
33k
R14
33k
R15
33k
C1
0.33uF
C2
0.1uF
C3
0.1uF
C4
10uF
C5
10uF
C6
0.1uF
C7
0.1uF
C8
10uF
C9
10uF
+5V
+7.4V
GND
ECHO1
ECHO2
TRIG1
TRIG2
UART1
UART2
PWMA
PWMB
L_MOT+
L_MOT-
R_MOT+
R_MOT-
LEFT_P_A
LEFT_P_B
RIGHT_P_A
RIGHT_P_B
V_MEASURE
7.4V-1
7.4V-2

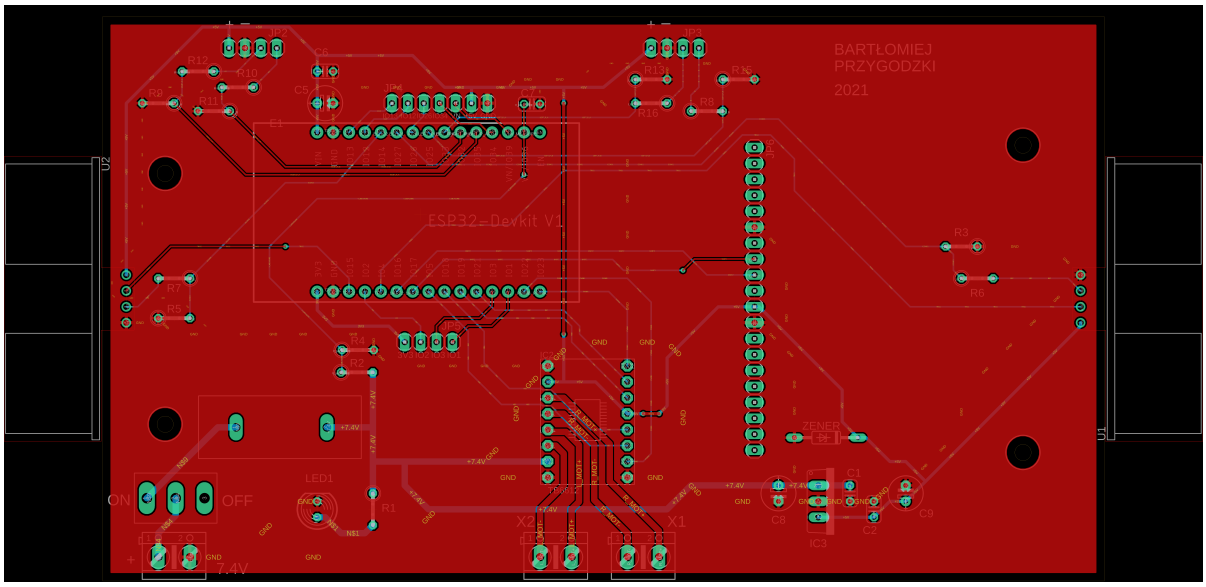
# Załącznik II

# Schemat płytki głównej


schemat2
08.01.2022 00:31
Sheet: 1/1
ESP32-DEVKITV4
TB6612FNG
HC-SR04
7805TV
BEZPIECZNIK 5A
MTS-1
ZENER
6V2
RED
LED1
JP6
JP4
JP5
JP3
JP2
U1
U2
IC2
IC3
X1-1
X1-2
X2-1
X2-2
7.4V-1
7.4V-2
V_MEASURE
UART1
UART2
TRIG1
TRIG2
ECHO1
ECHO2
PWMA
PWMB
LEFT_P_A
LEFT_P_B
RIGHT_P_A
RIGHT_P_B
GND
+5V
+7.4V
3V3

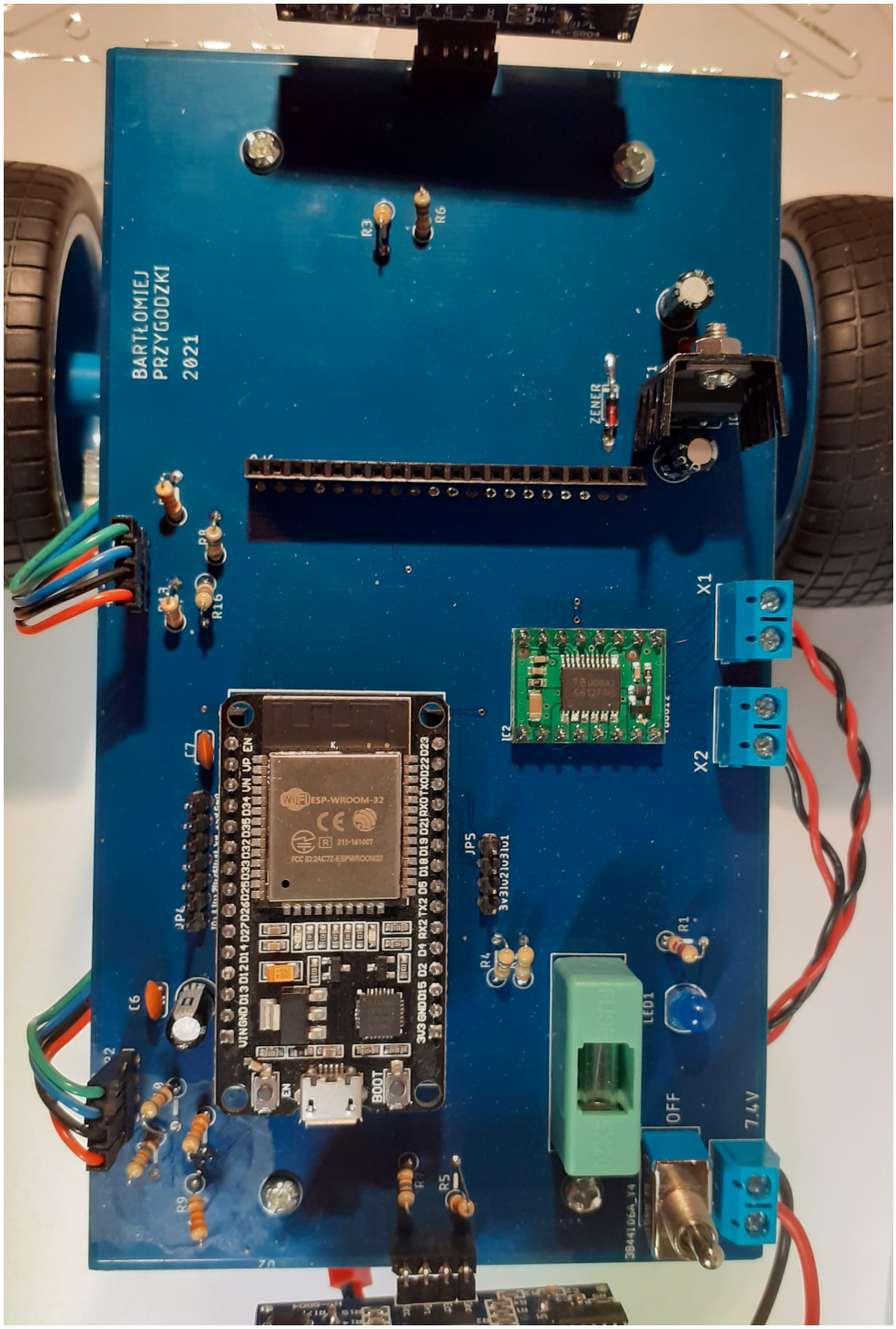
# Załącznik III

# Projekt płytki głównej



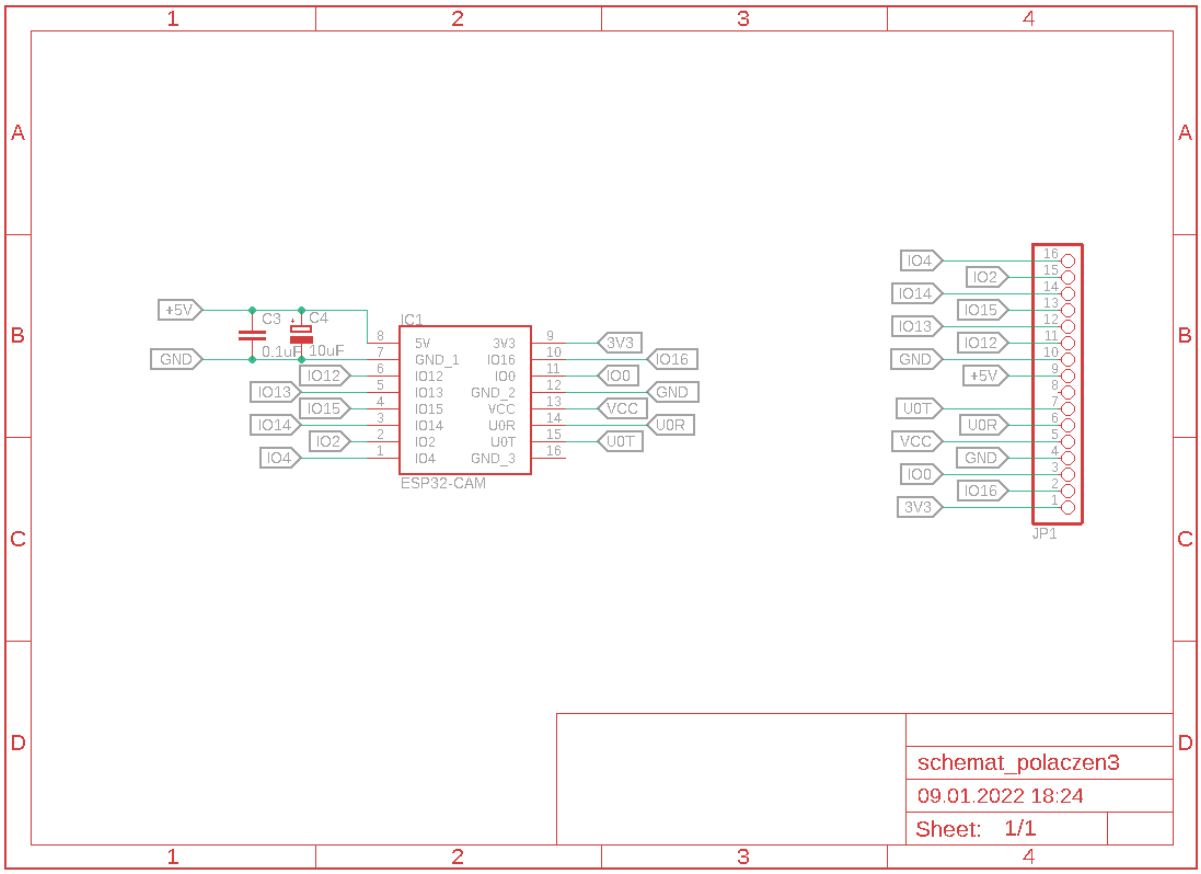
# Załącznik IV

# Zdjęcie płytki głównej


BARTŁOMIEJ
PRZYGODZKI
2021
ESP-WROOM-32
X1
X2
7.4V
OFF
ZENER
BOOT
EN

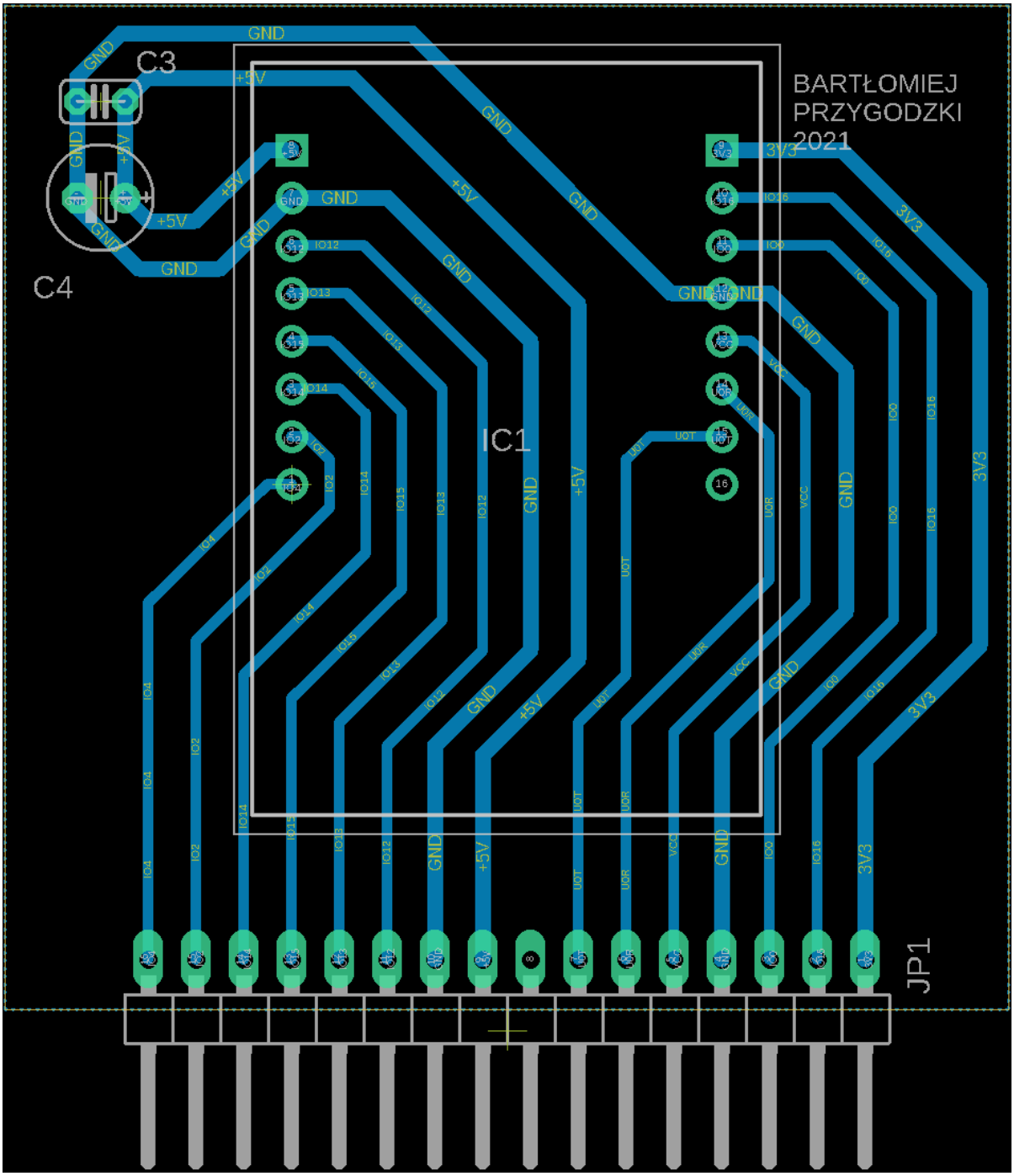
# Załącznik V

# Schemat płytki z kamerą



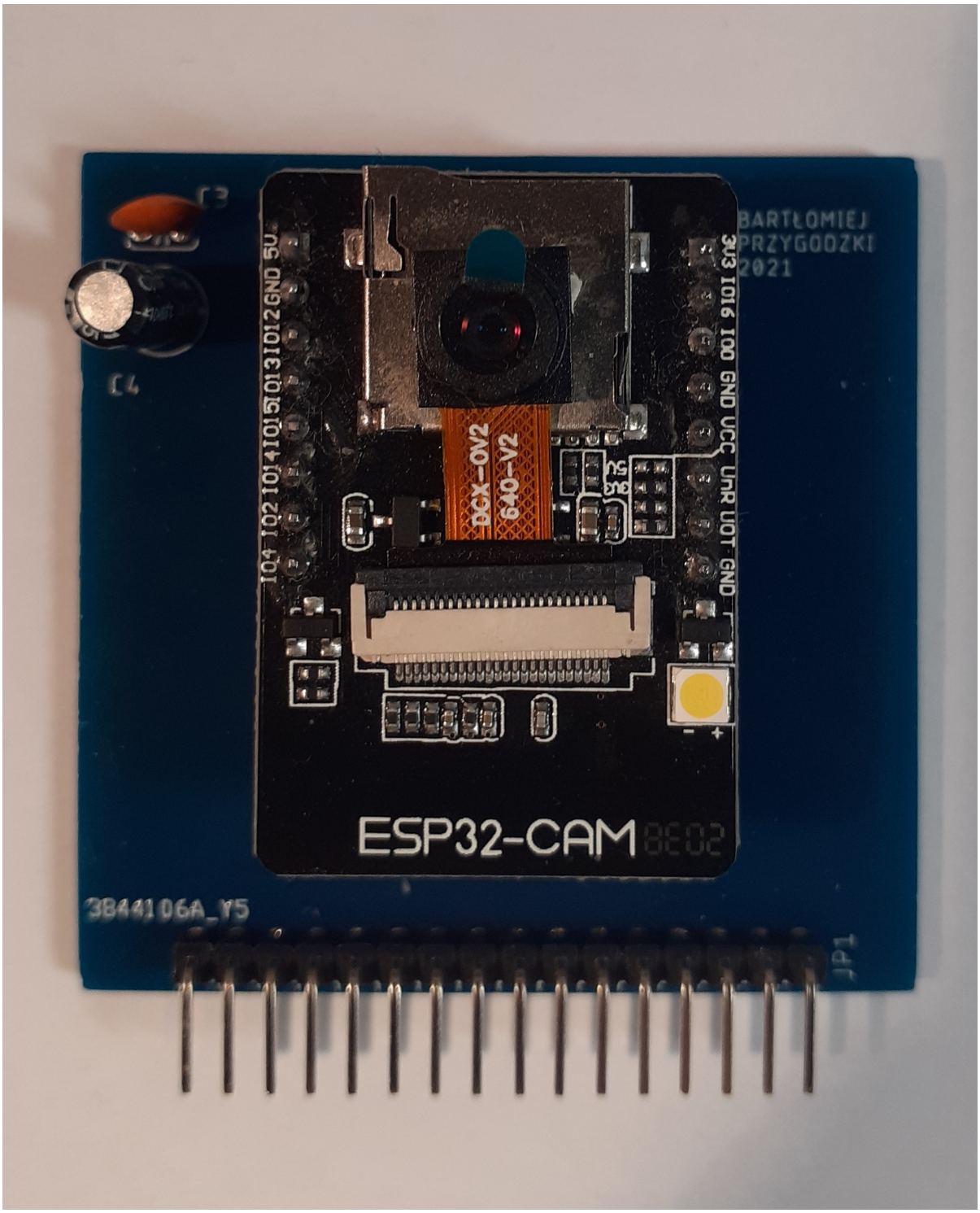
# Załącznik VI

# Projekt płytki z kamerą


BARTŁOMIEJ
PRZYGODZKI
2021
C3
C4
IC1
JP1
GND
+5V
3V3

# Załącznik VII

# Zdjęcie płytki z kamerą


BARTŁOMIEJ
PRZYGODZKI
2021
3V3 IO16 IO0 GND VCC U0R U0T GND
IO4 IO2 IO14 IO15 IO13 IO12 GND 5V
OGX-OV2
640-V2
ESP32-CAM
3B441 06A_Y5
JP1