

## STRESZCZENIE

Celem niniejszego projektu dyplomowego jest zbudowanie małego pojazdu autonomicznego zasilanego panelem fotowoltaicznym oraz poruszającego się po wyznaczonej trasie i reagującego na przeszkody. Sposobem autonomicznej pracy pojazdu jest śledzenie czarnej linii na białym tle. Panel fotowoltaiczny ma za zadanie przetwarzać energię słoneczną na energię elektryczną, która doładowuje akumulatory. Dzięki takiemu zastosowaniu wydłuży się czas pracy pojazdu. Projektowanie i budowa samochodu obejmują cztery etapy, z których każdy jest opisany w 4 głównych rozdziałach. Na początku konieczne jest dobranie odpowiedniego panelu fotowoltaicznego, co wiąże się z określeniem jego masy i mocy. Panel musi dawać jak najwięcej mocy, ale nie ważyć zbyt dużo, aby pojazd mógł go udźwignąć. Po dobraniu panelu należy zbudować odpowiedni system sterowania, czego istotną częścią jest dobranie elementów elektrycznych oraz ich połączenie. Ważniejszymi częściami są: sterownik L293D odpowiadający za możliwość zmiany prędkości i kierunku obrotu kół napędzanych dwoma silnikami, moduł Arduino Nano V3.0 z mikrokontrolerem Atmega328 na który wgrywany jest program oraz przetwornica odpowiedzialna za zmniejszanie napięcia otrzymywanego z panelu fotowoltaicznego do wartości 5,1 V (wartość napięcia akumulatorów waha się 4,0 do 5,6 V). Aby zapewnić poprawne działanie małego pojazdu koniecznym jest napisanie i wgranie na mikrokontroler odpowiedniego programu, który zapewni śledzenie czarnej linii i reagowanie na przeszkody. Dla opisanego projektu zostały napisane dwa kody programu. Po wykonaniu tych czynności pojazd może poruszać się autonomicznie. Ostatnim etapem jest wykonanie badań i określenie czy zastosowanie panelu fotowoltaicznego wydłuża pracę.

Projekt był realizowany przez dwóch studentów: Mikołaj Witkowski - udział w rozdziałach 1, 2, 3, 4 i 5 oraz indywidualny w podrozdziałach 2.1, 2.2, 3.2, 3.5, 3.6, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 i 4.9, Tobiasz Loos – udział w rozdziałach 1, 2, 3, 4, 5 oraz indywidualny w podrozdziałach 2.3, 2.4, 3.1, 3.3, 3.4, 3.8, 4.10, 4.11, 4.12.

**Słowa kluczowe:** Energia odnawialna, samochód autonomiczny, panel fotowoltaiczny, program wgrany na mikrokontroler modułu Arduino Nano, line follower.

## ABSTRACT

The aim of this project is to create small autonomous vehicle powered by photovoltaic panel and moving after the designated route, also responding to obstacles (stopping when something is In front of it). Option of the autonomous operation of the vehicle is following a black line on a white background. Photovoltaic panel is designed to process solar energy into electricity that recharges the batteries. Thanks to this time operation of the vehicle is extended. Design and construction of the car includes four stages, each of which is described in the 4 major chapters. At the beginning it is necessary to select the right photovoltaic panel, which stems from the determination of its weight and power. The Panel must give as much power, but do not weigh too much so the vehicle can carry it. After selecting the appropriate Panel, comes a time to build a control system, which an important part is the selection of electrical components and their connection. Major parts are; L293D driver responsible for the ability to change the speed and direction of rotation of the driven wheel two engines, the Arduino Nano V3.0 with microcontroller Atmega328 that can be uploaded with program and the inverter is responsible for reducing the voltage received with photovoltaic panel to the value 5.1 V (battery voltage ranges 4.0 to 5.6 V). To ensure proper operation of a small vehicle it is necessary to write and upload for microcontroller corresponding program that will keep vehicle following the black line and responding to obstacles. For the described project were written two codes. When these steps are completed the vehicle can move autonomously. The last step is to perform research and to determine whether the use of photovoltaic panel extends the work.

The project was carried out by two students: Mikołaj Witkowski – contribution to chapters 1, 2, 3, 4 and 5, and individual sections 2.1, 2.2, 3.2, 3.5, 3.6, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 and 4.9, Tobias Loos – contribution to chapters 1, 2, 3, 4, 5, and individual sections 2.3, 2.4, 3.1, 3.3, 3.4, 3.8, 4.10, 4.11, 4.12.

**Keywords:** renewable energy, autonomous car, photovoltaic panel, program uploaded to the microcontroller module Arduino Nano, line follower.

## SPIS TREŚCI

1. Wstęp i cel pracy (Mikołaj Witkowski, Tobiasz Loos).....	11
2. Doładowywanie akumulatorów pojazdu (Mikołaj Witkowski, Tobiasz Loos) .....	15
2.1. Zjawisko fotowoltaiczne (Mikołaj Witkowski) .....	15
2.2. Generowanie fotoprądu (Mikołaj Witkowski) .....	16
2.3. Ogniwa monokrystaliczne, a polikrystaliczne (Tobiasz Loos) .....	17
2.4. Wybrany panel fotowoltaiczny (Tobiasz Loos) .....	18
3. System sterowania pojazdem (Mikołaj Witkowski, Tobiasz Loos) .....	25
3.1. Schemat elektryczny (Tobiasz Loos) .....	25
3.2. Moduł Arduino Nano V3.0 (Mikołaj Witkowski) .....	27
3.3. Sterownik L293D (Tobiasz Loos) .....	28
3.4. Czujniki odbiciowe (Tobiasz Loos) .....	31
3.5. Zbliżeniowy czujnik ultradźwiękowy (Mikołaj Witkowski) .....	33
3.5.1.Kod programu do obsługi czujnika ultradźwiękowego (Mikołaj Witkowski)...	35
3.6. Zbliżeniowy czujnik podczerwieni (Mikołaj Witkowski) .....	36
3.6.1.Kod programu do obsługi czujnika podczerwieni (Mikołaj Witkowski) .....	38
3.7. Akumulatory niklowo-kadmowe (Ni-Cd) (Tobiasz Loos) .....	40
3.8. Przetwornica DC/DC (Tobiasz Loos) .....	41
4. Kody programów sterujących (Mikołaj Witkowski, Tobiasz Loos) .....	43
4.1. Opis programu pierwszego (Mikołaj Witkowski).....	44
4.2. Schemat blokowy algorytmu sterowania (Mikołaj Witkowski).....	44
4.3. Inicjalizacja (Mikołaj Witkowski) .....	45
4.4. Wnętrze funkcji setup (Mikołaj Witkowski) .....	47
4.5. Kod do obsługi czujników odbiciowych (Mikołaj Witkowski) .....	48
4.6. Kod do obsługi czujników ultradźwiękowych (Mikołaj Witkowski) .....	50
4.7. Kod do obsługi czujnika podczerwieni (Mikołaj Witkowski) .....	51
4.8. Kod do ustawiania prędkości (Mikołaj Witkowski) .....	52
4.9. Wnętrze funkcji loop (Mikołaj Witkowski) .....	53
4.10. Funkcja setup i inicjalizacja zmiennych programu drugiego (Tobiasz Loos).....	55
4.11. Wydzielone funkcję (Tobiasz Loos) .....	56
4.12. Funkcja loop (Tobiasz Loos) .....	58
5. Badania, wnioski i podsumowanie projektu (Mikołaj Witkowski, Tobiasz Loos).....	61
5.1. Poruszanie się pojazdu po zaprojektowanych trasach.....	62
5.2. Rozładowywanie się akumulatorów w czasie jazdy pojazdu.....	63
5.3. Absorbowanie natężenia słonecznego.....	64
5.4. Rozładowywanie się akumulatorów przy różnych ustawieniach pojazdu.....	65
5.5. Badania przeprowadzone przy użyciu halogenów.....	66
5.6. Wnioski, podsumowanie i przyszły rozwój projektu.....	69
Wykaz literatury.....	71

Wykaz rysunków.....	72
Wykaz tabel.....	74

## 1. WSTĘP I CEL PRACY

Rozwój samochodów elektrycznych oraz odnawialnych źródeł energii, na przestrzeni ostatnich lat, znacząco wzrósł. Trend ten będzie się utrzymywał, ponieważ zapotrzebowanie społeczeństwa na źródła energii odnawialnej nie zanieczyszczającej środowiska stale rośnie. Ludzie zmieniają swój punkt widzenia i w coraz większym stopniu zaczynają dbać o ekologię. Liczne kampanie promujące zieloną energię, samochody elektryczne, czy nawet segregacja odpadów przynoszą widoczne skutki. Wielkimi postaciami, którzy przyczynili się do tego są Elon Musk, Boyan Slat oraz wielu naukowców, którzy po dziś dzień rozwijają technologię paneli słonecznych, w tym Polka Olga Malinkiewicz. Istnieją liczne ustawy i zobowiązania podpisywane przez głowy państw na zgromadzeniach politycznych, które dotyczą minimalizacji emisji zanieczyszczeń, w tym dwutlenku węgla. Stworzony w tym celu został między innymi pakiet energetyczno-klimatyczny. Można zaobserwować coraz większe inwestycje w elektrownie słoneczne, wiatrowe czy w wodne.

Nawet motoryzacja zaczęła się rozwijać uwzględniając w dużym stopniu minimalizację emisji spalin. Jednym ze sposobów na redukcję zanieczyszczeń w miastach jest wprowadzenie samochodów elektrycznych. Kolejnym krokiem są samochody autonomiczne, czyli bezzałogowe pojazdy mogące poruszać się po drodze bez pomocy osób trzecich oraz podejmujące samodzielnie decyzje. Wprowadzenie takich udogodnień w miastach umożliwiłoby programowanie systemów sterowania, które zapewniłyby minimalne zużycie energii elektrycznej bądź paliwa.

W artykule „Autonomous Vehicle Implementation Predictions” [1.1] opublikowanym 8 sierpnia 2017 roku podano przypuszczalny kierunek rozwoju samochodów autonomicznych. Autorzy przewidują, że w latach 2020 – 2030 będą widoczne pierwsze korzyści z takiego rodzaju transportu, a w latach 2040 – 2060 będziemy mogli je z łatwością odczuć. Jednak bez widocznego wpływu wielkich koncernów samochodowych nie ma szans na szybki rozwój pojazdów tego rodzaju. Potencjalnymi korzyściami, jakie mogą odnieść ludzie, dzięki autom prowadzonym bez udziału kierowców są: wzrost bezpieczeństwa w ruchu drogowym, redukcja zanieczyszczeń w miastach oraz minimalizacja kosztów ponoszonych na zużycie paliwa bądź energii elektrycznej. Rolą jaką człowiek by pełnił, korzystając z samochodu autonomicznego mogłaby być możliwość nadzorowania pracy systemu. Tylko w nielicznych momentach człowiek miałby reagować na zaistniałe niebezpieczeństwo, które i tak będzie malało wraz z rozwojem tej technologii oraz zaprzestaniem wprowadzania do ruchu drogowego samochodów obsługiwanych przez człowieka. Pojazdy miałyby wybierać optymalną trasę, jeździć i parkować samodzielnie, reagować na znaki drogowe oraz oczywiście na zagrożenia pojawiające się na drodze.

W książce „An integrated perspective on the future of mobility” [1.2] autorzy przewidują, że w ciągu 15 lat zaludnienie miast wzrośnie o 30%. Skutkiem czego będą przeludnione miasta i pojawi się problem z utrzymaniem sprawnego ruchu drogowego. Rozwiązaniem, jakie sugerują twórcy teorii jest wdrożenie w miastach autonomicznego transportu drogowego zasilanego energią elektryczną.

Jest to stosunkowo nowa dziedzina , więc potrzeba czasu, aby ludzie przekonali się do takiej formy korzystania z samochodów. Patrząc jednak na liczne opinie konsumentów autonomicznych pojazdów elektrycznych, pod kątem dbania o środowisko, można postawić tezę, iż będą odgrywały kluczową rolę w codziennym życiu społeczeństwa. Wiele światowych koncernów takich jak Google, Tesla, czy Uber inwestują w badania i rozwój autonomicznych samochodów elektrycznych.

Zanim zostanie umożliwione wprowadzenie samosterujących się aut w miastach, dobrym pomysłem jest produkowanie samochodów ciężarowych poruszających się na długie dystanse międzymiastowe, bez udziału kierowców. Dużą zaletą takiego rozwiązania jest przyspieszenie i zwiększenie dystansu transportowanych produktów. Brak udziału człowieka w sterowaniu samochodem umożliwi eliminację dłuższych przerw, ponieważ samochód nie potrzebuje snu. Wraz z upływem czasu, ludzi chętnych do pracy w transporcie będzie coraz mniej, ze względu na nieregularne godziny pracy i częste dni spędzane poza domem.

Na chwilę obecną odbyła się jedna podróż z udziałem kierowcy, ale bez jego ingerowania w sterowanie przez prawie całą trasę. Autonomiczną pracę pojazdu, zwanego tirem, podczas poruszania się na autostradzie, opracowała firma Otto z Anthonym Lewandowskim na czele, który uzyskał dofinansowanie na swoje badania od wymienionej wcześniej korporacji, Ubera. Z artykułu [1.3] na temat działań spółki, wynika że udało się dokonać pierwszego w historii przetransportowania kilkudziesięciu tysięcy sztuk puszek piwa na dystansie 120 mil. Mimo, że kierowca znajdował się w pojeździe, to jego rola ograniczała się do obserwacji oraz sterowania na początku i końcu trasy. Firma zdradza, że opracowany przez nią system wbudowany do zaprojektowanego samochodu ciężarowego marki Volvo współpracował przy pomocy kamer, radaru oraz lidar. Przykład został zaprezentowany na rysunku 1.1.



*Rys 1.1. Ciężarówka Volvo z systemem sterowania OTTO (Za kierownicą nie znajduje się człowiek) [1.3]*

Oprócz wyżej wymienionego samochodu, Uber inwestuje jeszcze w projekty autonomicznych taksówek, ale z powodu nieścisłości prawnych nie są one na razie wielkim

osiągnięciem. Prawo w większości krajów mówi jasno, że za kierownicą pojazdu musi znajdować się osoba prawna posiadająca odpowiednie uprawnienia. Problem tym nie przejmują się wysoce już rozwinięta firma produkująca samochody marki Tesla. Na podstawie artykułu „Autopilot za kierownicą...” [1.4] w roku 2016 został zaprezentowany film z udziałem samochodu poruszającego się autonomicznie. Na rysunku 1.2 zostało przedstawione zdjęcie wykonane podczas badań pracy samochodu. System sterowania składa się z 8 kamer wizyjnych o zasięgu 250 m umożliwiających rejestrowanie obrazu w 360<sup>0</sup> wokół samochodu. Układ kamer wspierany jest przez 12 czujników ultradźwiękowych i radar, mających wspierać pracę systemu w niesprzyjających warunkach pogodowych.



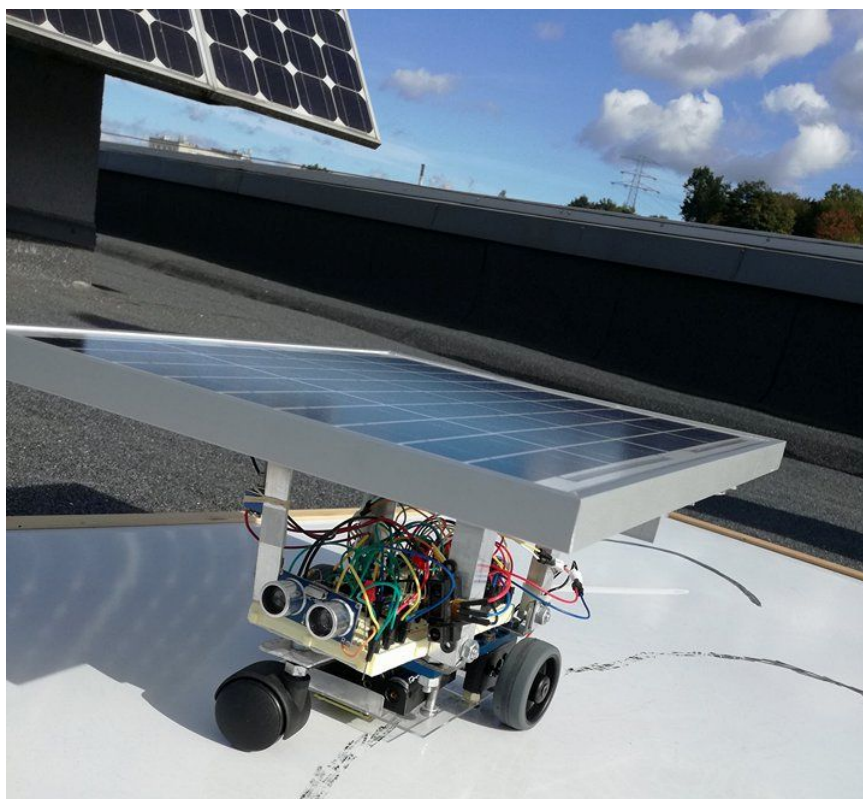
*Rys 1.2. Wnętrze pojazdu tesli podczas pracy autonomicznej [1.4]*

Wymienione wyżej artykuły i dokonania wielu firm zapoczątkowały projekt małego autonomicznego pojazdu zasilanego panelem fotowoltaicznym, który jest tematem tej pracy inżynierskiej. Celem przy projektowaniu było zbudowanie odpowiedniego systemu sterowania oraz zaimplementowanie programu, które zapewniłyby poprawne funkcjonowanie.

Pojazd został zaprojektowany jako samochód autonomiczny śledzący linię i reagujący na przeszkody. W tym celu zaopatrzono go w odpowiednie czujniki. Funkcje akumulatora energii elektrycznej pełnią ogniwa niklowo-kadmowe, doładowywane panelem fotowoltaicznym, o maksymalnej mocy 10 watów. W teorii rozwiązanie powinno umożliwić pojazdowi pracę, czerpiąc tylko energię uzyskaną z panelu fotowoltaicznego, ponieważ do napędzania zostały wykorzystane dwa niewielkie silniki wolnoobrotowe o mocy nie przekraczającej 2 watów każdy. Aby było to możliwe potrzebne jest odpowiednie nasłonecznienie. Zakładając że cały układ łącznie z całą elektroniką potrzebuje około 5 watów, to aby go zasilić bez udziału energii pobieranej z akumulatorów, nie jest konieczna optymalna praca panelu.

Na rysunku 1.3 pokazano pojazd laboratoryjny z zamontowanym panelem fotowoltaicznym. Pod podwoziem samochodu umieszczone są trzy czujniki optyczne.

Dodatkowymi czujnikami zastosowanymi w pojeździe są dwa czujniki ultradźwiękowe oraz jeden analogowy czujnik podczerwieni.



*Rys 1.3. Autonomiczny pojazd elektryczny zasilany panelem fotowoltaicznym*

Mikrokontroler odpowiadający za autonomiczną pracę samochodu jest wbudowany w wykorzystanym module Arduino Nano. Za poprawną pracę silników i możliwość zmiany kierunku jazdy odpowiada sterownik L293D. Skręcanie odbywa się poprzez wymuszenie obracania się tylko jednego z dwóch napędzanych kół, przy czym przednie porusza się swobodnie. Zastosowanie przełącznika trójstanowego, umożliwia operatorowi wybór kierunku jazdy. Nominalne napięcie układu wynosi około 5 V, a z panelu słonecznego można uzyskać około 17 V. W celu obniżenia napięcia została zastosowana przetwornica DC/DC dająca na wyjściu 5.1 V.

Panel fotowoltaiczny podtrzymywany jest przez aluminiową konstrukcję. Aluminium jest dość lekkim tworzywem, a jego wytrzymałość w zupełności wystarcza do utrzymania panelu słonecznego.



## 2. DOŁADOWYWANIE AKUMULATORÓW POJAZDU

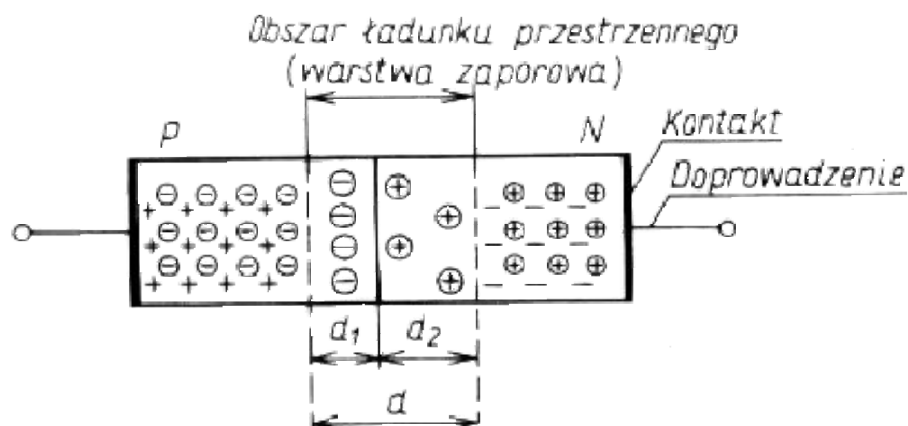
Wśród szybko rozwijającej się cywilizacji występuje zapotrzebowanie na zaopatrzenie w środki transportu, będący innym, niż zasilany ropą naftową lub źródłem nie tylko zagrażającym środowisku, ale i takim, którego zasoby skończą się w nieodległej przyszłości. Jednym ze sposobów zasilania samochodów przyszłości jest energia elektryczna. Istniejące w dzisiejszych czasach samochody elektryczne mają niewielki zasięg i niewiele stacji doładowujących, a czas odzyskiwania żądanej wartości napięcia akumulatora zabiera dużo czasu. Przykładem najbardziej prestiżowego oraz o imponujących parametrach samochodu jest Tesla Model S, który oferuje wbudowane akumulatory litowo-jonowe o pojemności 85 kWh i zasięg 480 km do przejechania na jednym doładowaniu, na specjalnych stacjach „SuperCharger” zajmującym około 45 minut. Innym przykładem, tym razem tańszej wersji samochodu elektrycznego jest Ford Focus Electric, z wbudowanymi bateriami litowo-jonowymi o pojemności 23 kWh, umożliwiającymi przejechanie 122 km na jednym doładowaniu, trwającym do 4 godzin, korzystając przy tym z gniazdka sieciowego i wbudowanego prostownika [2.1].

Na podstawie książki Bogumiła Fica „Samochody Elektryczne” można przyjąć, że 1 kWh pozwala na przejechanie 5 km, a z metra kwadratowego paneli fotowoltaicznych możemy otrzymać 100 W. Jednakże w polskich warunkach klimatycznych nie zawsze jest słoneczna pogoda, a możliwym do uzyskania jest tysiąc godzin nominalnej pracy, czyli 100 kWh rocznie dla panelu o danych wymiarach [2.1].

### 2.1. *Zjawisko fotowoltaiczne*

Zjawisko fotowoltaiczne powstaje w materiałach półprzewodnikowych i polega na przemieszczaniu się elektronów z pasma podstawowego (walencyjnego) do pasma przeniesienia. Dzięki absorpcji światła słonecznego uzyskuje się energię potrzebną do uwolnienia ładunku o potencjale ujemnym. Należy uzyskać odpowiednią ilość energii, aby elektron pokonał pasmo wzbronione – przerwę elektryczną, która dla krzemu, w temperaturze otoczenia  $\approx 27^{\circ}\text{C}$ , wynosi 1,12 eV. Krzem jest głównym materiałem, z którego zbudowane są ogniwa fotowoltaiczne.

Uwolnienie elektronu z pasma walencyjnego powoduje powstanie dziury o potencjale dodatnim. W wyniku czego pomiędzy obszarem typu p (pasma walencyjne) i obszarem typu n (pasma przewodzenia) powstaje różnica potencjałów. Dochodzi do przemieszczania się ładunków. Na granicy obu obszarów generuje się połączenia elektronów i dziur. Powoduje to dyfuzję elektronów z obszaru n do obszaru p oraz z obszaru p do obszaru n. W paśmie przewodzenia będzie przeważał ładunek dodatni, a w paśmie podstawowym ujemny. Dalszej dyfuzji, czyli przemieszczaniu się ładunków, zapobiega warstwa zaporowa, znajdująca się na środku obszaru złącza p-n. Warstwa zaporowa to bariera pola elektrycznego. Skierowane jest (umownie) od potencjału dodatniego, do ujemnego, czyli w tym przypadku od obszaru n do obszaru p.

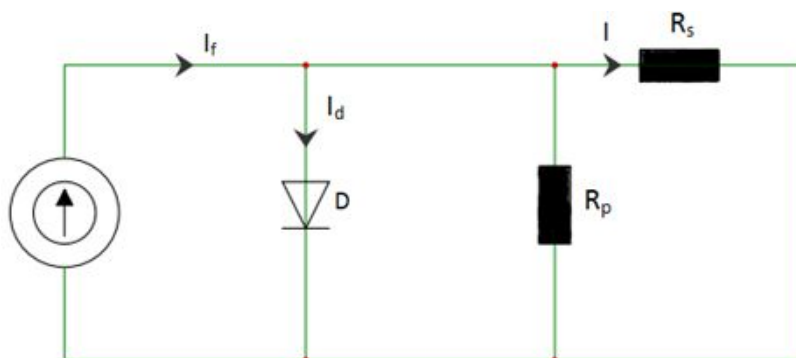


Rys. 2.1. Przedstawienie równowagi w obszarze złącza p-n [2.2.]

Po zakończeniu przemieszczania się ładunków, w obu pasmach półprzewodnika, znajdują się również nośniki mniejszościowe. Są to nośniki o przeciwnym ładunku elektrycznym. W każdym obszarze jeden ładunek jest przeważający i taki charakter elektryczny dany obszar posiada.

## 2.2. Generowanie fotoprądu

Zjawisko fotowoltaiczne, opisane powyżej, powoduje powstawanie fotoprądu. Generowanie fotoprądu  $I_f$  można przedstawić na schemacie zastępczym ogniwa słonecznego. Dioda D reprezentuje złącze p-n. Rezystor równoległy, o rezystancji  $R_p$ , modeluje upływność, a rezystor połączony szeregowo, o rezystancji  $R_s$ , modeluje straty w ogniwie słonecznym.



Rys. 2.2. Model elektryczny ogniwa fotowoltaicznego

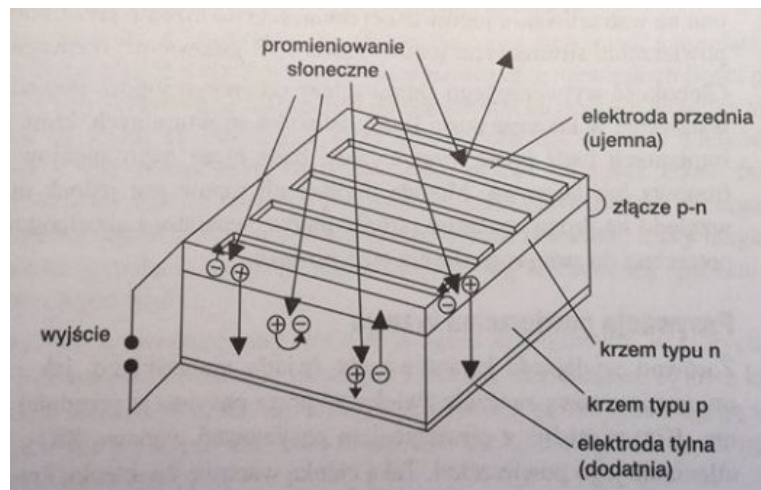
Panel słoneczny jest źródłem prądu stałego. Poprzez absorpcję promieni słonecznych, które są falami elektromagnetycznymi, generowany jest prąd w ogniwach fotowoltaicznych. Jest to prąd elektryczny wywołany zjawiskiem fotoelektrycznym, opisanym w rozdziale [2.3]. Zgodnie z prawem Ohma prąd na wyjściu ogniwa jest różnicą między prądem źródła, a prądem diody. Na natężenie fotoprądu  $I_f$  ma wpływ gęstość strumienia fotonów  $N_f$ , powierzchnia ogniwa  $S$ , współczynnik wydajności kwantowej  $\alpha_f$  oraz współczynnik strat  $\beta$ .

$$I_f(\lambda) = e * \alpha_f * N_f(\lambda) * S * (1 - \beta) \quad (2.1)$$

$I_f(\lambda)$  – Natężenie fotoprądu na jednostkę długości fali słonecznej

$e$  – ładunek elementarny, elektrony [C]

Ogniwo fotowoltaiczne nie absorbuje całego strumienia fotonów, część z nich zostaje odbita. Stosuje się warstwę antyodblaskową, którą pokrywa się przednią część ogniwa słonecznego. Ogniwo fotowoltaiczne składa się przede wszystkim z płytki krzemowej, w której wytworzono złącze p-n.



Rys. 2.3. Schemat budowy ogniwa fotowoltaicznego [2.3]

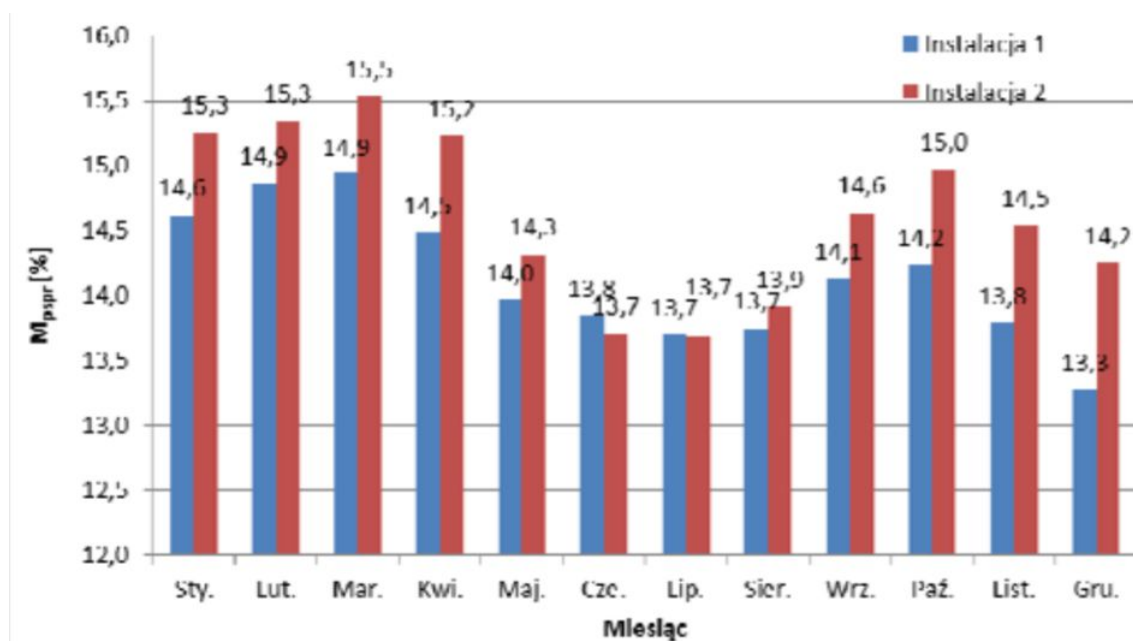
### 2.3. Ogniwa monokrystaliczne, a polikrystaliczne

Dobranym do małego pojazdu źródłem energii odnawialnej został polikrystaliczny panel fotowoltaiczny, między innymi ze względu na łatwą dostępność i niewielką cenę. W obecnych czasach jest najbardziej popularnym rodzajem z pierwszego typu. Typ drugi różni się od pierwszego pierwiastkami, jakie są wykorzystywane do produkcji ogniw fotowoltaicznych (krzem jest wykorzystywany tylko do tworzenia paneli pierwszego typu). Mimo tego, że moduł monokrystaliczny jest w stanie wyprodukować więcej energii, ponieważ posiada wyższą sprawność, panel polikrystaliczny jest mniej narażony na utratę sprawności spowodowaną wzrostem temperatury.

Według badań wykonanych na Uniwersytecie Warszawskim do projektu „Badanie ogniw fotowoltaicznych” [2.4] stwierdzono, że ogniwo monokrystaliczne posiada o około 80% większą sprawność od ogniwa polikrystalicznego. Z informacji jakie zostały umieszczone w projekcie wynika, że badania były wykonane w zamkniętym pomieszczeniu, przez co ciężko było stwierdzić wpływ temperatury na sprawność przetwarzania energii. Mimo tego z artykułu naukowego Uniwersytetu Warmińsko-Mazurskiego [2.5] opisującego badanie różnicy rocznej energii elektrycznej wytworzonej przez panele monokrystaliczne, ułożone pod kątem  $0^\circ$  do podłoża i panele polikrystaliczne, nachylone pod kątem  $15^\circ$  wynika, że sprawność w warunkach STC (ang. Standard Test Conditions, czyli takich dla których temperatura wynosi  $25^\circ\text{C}$ , a promieniowanie  $1000\text{ W/m}^2$ ) jest nieco wyższa dla rodzaju polikrystalicznego, ponieważ wynosi 15,3%, a dla monokrystalicznego 14,91%. Z opisanych badań i rysunku 2.4 wynika, że monokrystaliczne panele fotowoltaiczne znacząco tracą sprawność w okresach, kiedy jest najcieplej, a promieniowanie słoneczne jest największe. Jest to niekorzystne, ponieważ w czasie wakacyjnym można uzyskać najwięcej energii elektrycznej oraz znaczenie mają nawet

najmniejsze zmiany skuteczności w przetwarzaniu energii świetlnej na elektryczną. Dla polikrystalicznego rodzaju panelu widać, że wahania sprawności nie są takie duże.

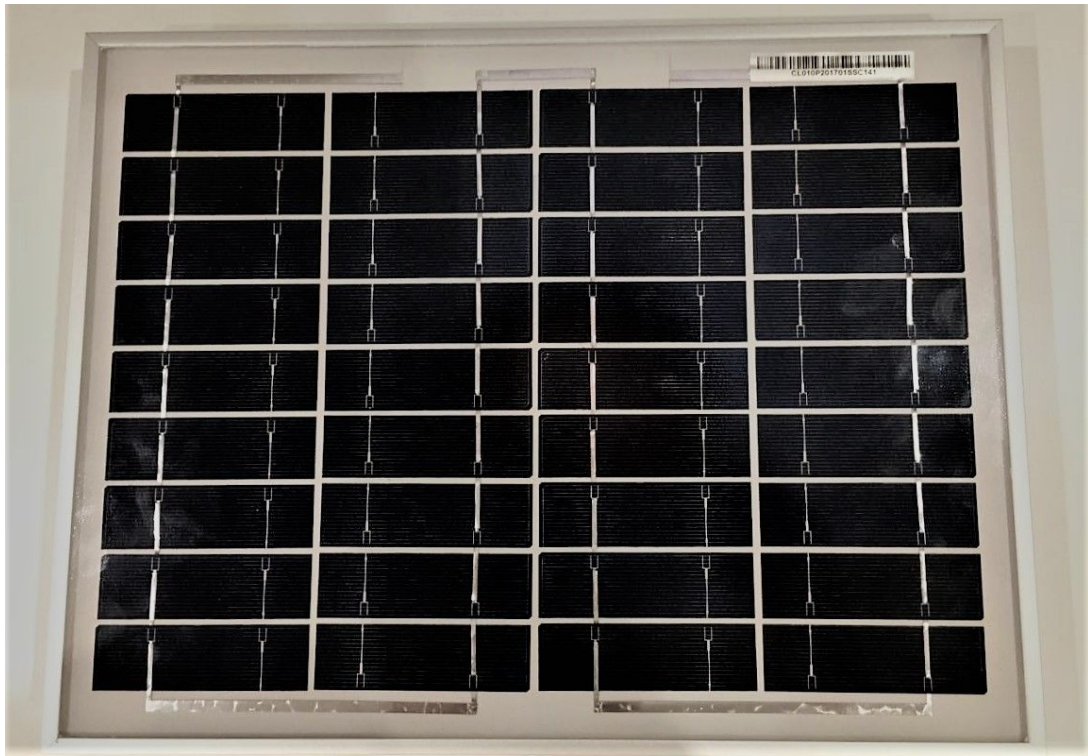
Na podstawie wyżej wymienionych argumentów nasuwają się wnioski, że w klimacie, w którym mogą występować wysokie temperatury bardziej opłaca się stosować polikrystaliczne ogniwa fotowoltaiczne, będące mniej wrażliwe na duże wzrosty temperatur. Jest to bardzo ważną i istotną cechą, ponieważ panele słoneczne potrafią się bardzo szybko nagrzewać poprzez padające na nie promienie słoneczne.



Rys. 2.4. Sprawność uzyskanej energii elektrycznej z paneli fotowoltaicznych monokrystalicznych (kolor czerwony) i polikrystalicznych (kolor niebieski) [2.6]

## 2.4. Wybrany panel fotowoltaiczny

Do badań wybrano panel o mocy 10 W i masie 1.1 kg, który przedstawiono na rysunku 2.5. Panel służy do doładowywania czterech ogniw akumulatorów niklowo-kadmowych o napięciu 1.2 V każdy, zasilających dwa oddzielnie działające silniki oraz mikrokontroler z czujnikami. Przed dokonaniem zakupu zostały przeprowadzone testy na małym pojeździe za pomocą prostych obciążników odpowiadających przybliżonej wadze panelu w celu ustalenia czy silniczki będą w stanie ruszyć z danym ciężarem. Na rysunku 2.6 zostały przedstawione dane techniczne podane przez producenta wraz z optymalnym prądem i napięciem. Dane panelu w tabelce odpowiadają wcześniej wspomnianym określonym warunkom pracy baterii słonecznej, tak zwanym STC.



*Rys. 2.5. Panel Fotowoltaiczny wykorzystany do badań*

Badania wartości prądu i napięcia zależnie od manipulowanej rezystancji obciążenia mają na celu określenie jaką moc ma panel fotowoltaiczny w różnych porach roku oraz warunkach klimatycznych, czyli temperaturze i nasłonecznieniu. Charakterystyki zostały wyznaczone dla różnych miesięcy. Pierwsze podejście odbyło się 19 czerwca w godzinach popołudniowych. Z informacji uzyskanych tamtego dnia na temat warunków pogodowych, wynikało, iż temperatura wynosiła  $28^{\circ}\text{C}$ , a natężenie promieniowania słonecznego  $850 \text{ W/m}^2$ . Drugie podejście przeprowadzono na przełomie lata i jesieni, 18 października. Temperatura powietrza wynosiła  $16^{\circ}\text{C}$ , a nasłonecznienie  $500 \text{ W/m}^2$ . Wartość optymalna uzyskanej mocy elektrycznej wyniosła dla danych warunków  $8,76 \text{ W}$ , czyli około 88% wartości mocy maksymalnej podanej przez producenta. Na podstawie wykonanych pomiarów, można potwierdzić, że moc maleje wraz ze spadkiem nasłonecznienia. Zaprezentowany w dalszej części pracy rysunek 2.9 przedstawiający charakterystykę prądowo-napięciową potwierdza, że spadek temperatury powoduje wzrost maksymalnej wartości napięcia. Dzięki czemu w niższej temperaturze i przy dużym nasłonecznieniu, można uzyskać większą moc elektryczną. Punkty pomiarowe z badań zostały zaprezentowane w tabelach 2.1 i 2.2.

W celu dokładnego wyznaczenia charakterystyk prądowo-napięciowych został zbudowany prosty układ elektryczny zbudowany z amperomierza, woltomierza i 3 rezystorów suwakowych: dwóch o maksymalnej rezystancji równej  $10 \Omega$  oraz jednego o rezystancji  $40 \Omega$ , przy pomocy którego zostały wykonane badania w czerwcu. W powtórnych badaniach wykonanych w październiku już dla innych warunków atmosferycznych rezystor o zakresie  $40 \Omega$  został zamieniony na  $50 \Omega$ . Schemat elektryczny, wykorzystywany przy wykonaniu pierwszych badań został pokazany na rysunku 2.7.



Model type		CL010-12
Rated power	Pmax	10W
Rated current	I <sub>mp</sub>	0.57A
Rated voltage	V <sub>mp</sub>	17.5V
Short circuit current	I <sub>sc</sub>	0.63A
Open circuit voltage	V <sub>oc</sub>	22.0V
Module dimensions		340X280X17mm
Weight		1.1kg

Standard test condition : AM1.5; 1000W/m<sup>2</sup> ; 25 °C



Rys. 2.6. Dane techniczne panelu fotowoltaicznego

Aby odpowiednio dobrać rezystory, należało skorzystać z informacji podanych przez producenta, takich jak optymalny prąd i napięcie. Dzięki temu można było w przybliżeniu wyznaczyć, ze wzoru na prawo Ohma (2.2), jaki zakres rezystancji będzie potrzebny do manipulacji i zmiany wartości napięcia, aby wykreślić charakterystyki prądowo-napięciowe oraz przetestować działanie panelu.

W przypadku pierwszych badań dla obwodu z zerową rezystancją, prąd okazał się nieznacznie większy niż zakładany w danych technicznych, a napięcie dla obwodu rozwartego mniejsze. Poza dwoma wymienionymi wyżej pomiarami wykonano ich jeszcze szesnaście, a dokładne wartości zostały przedstawione w tabeli 2.1.

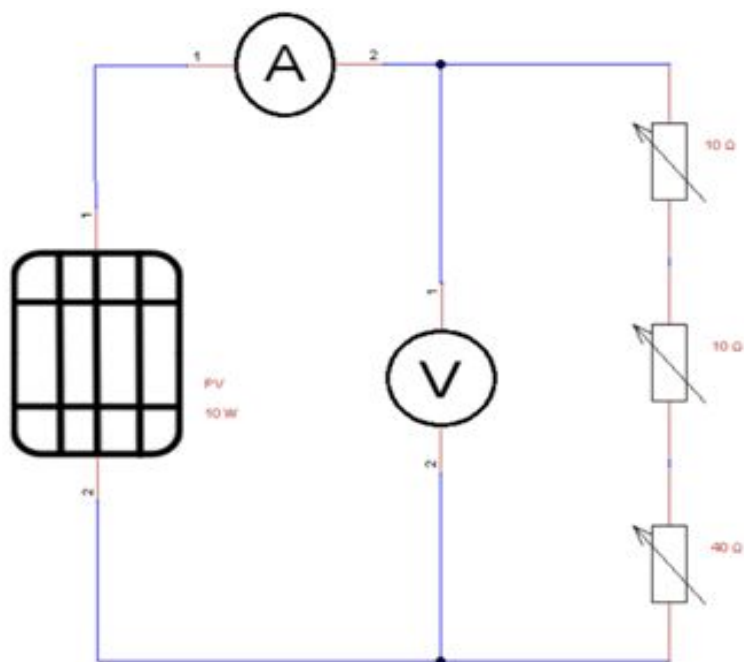
Przekształcone prawo Ohma do wyznaczania rezystancji:

$$R = \frac{U}{I} \quad (2.2)$$

gdzie:

R – rezystancja, U – napięcie, I – prąd.

Dla danych podanych w tabeli 2.1 korzystając ze wzoru na moc elektryczną (2.3) można określić, że moc optymalna jest osiągana dla napięcia o wartości 16 V. Umożliwiło to uzyskanie z panelu mocy optymalnej o wartości 9,28 W, czyli około 93% wartości mocy maksymalnej podanej przez producenta. Prąd zwarcia jest równy 0,6675 A, a napięcie obwodu otwartego 21,25 V. Dane w przybliżeniu zgadzają się z tymi podanymi przez producenta, a nieznaczne różnice wynikają z innych warunków, w jakich wykonywane były badania. Na rysunku 2.8 i 2.9 zostały wykreślone dwie charakterystyki (wspomnianą wyżej oznaczono kolorem czerwonym).



Rys. 2.7. Schemat elektryczny układu pomiarowego prądu i napięcia

**Tabela 2.1.** Pomiary wartości napięcia i prądu otrzymane w warunkach zbliżonych do STC –  $850 \text{ W/m}^2$ ,  $28^\circ\text{C}$

Numer pomiaru	Napięcie [V]	Prąd [A]
1	0,5	0,6675
2	2	0,6625
3	3	0,6625
4	5	0,6625
5	7,5	0,661
6	10	0,66
7	12	0,65
8	13,5	0,64
9	15	0,61
10	16	0,58
11	17	0,54
12	18	0,46
13	18,2	0,44
14	18,5	0,4
15	18,8	0,37
16	19	0,34
17	19,2	0,32
18	21,25	0

W celu zaprezentowania wyraźnych różnic, na rysunku 2.9 przedstawiono przebiegi mocy od napięcia dla wcześniej wyznaczonych danych. Na podstawie tabeli 2.2 można wywnioskować, że dla mniejszej temperatury i nasłonecznienia wartość prądu zwarcia zmalała i wyniosła  $0.65 \text{ A}$ , a napięcie rozwarcia wzrosło do wartości  $21,75 \text{ V}$ , czyli wciąż mniejszej, niż

podana przez producenta. Ponownie korzystając ze wzoru na moc (2.3), można stwierdzić, że moc optymalna jest osiągalna dla napięcia o wartości 17 V.

Wzór na moc w obwodzie prądu stałego:

$$P = U \cdot I \quad (2.3)$$

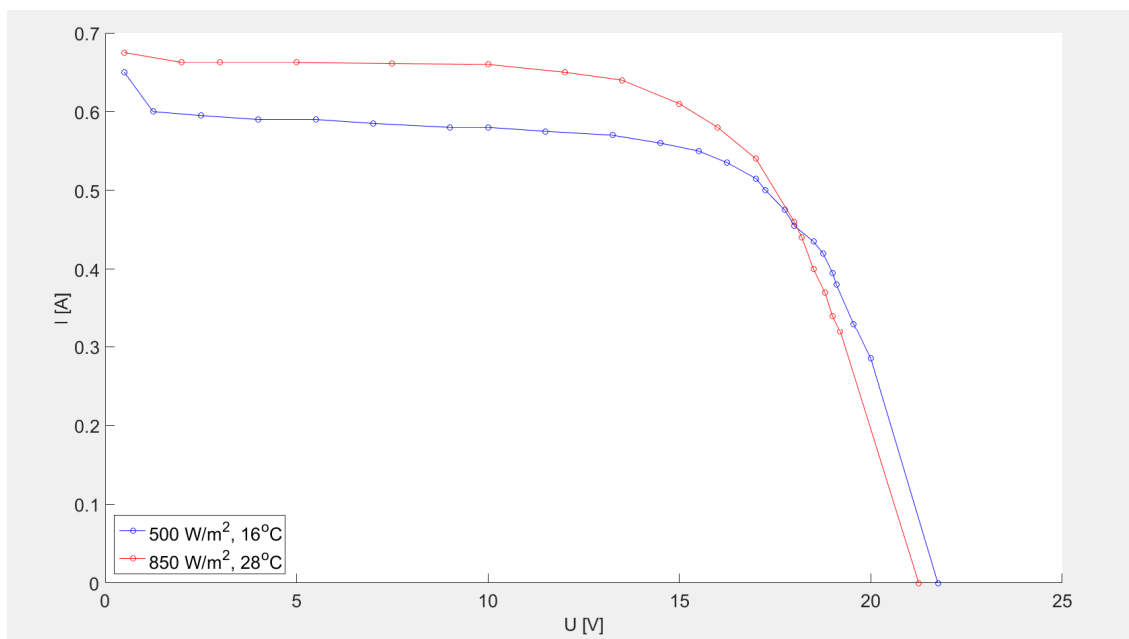
gdzie:

P – moc elektryczna, U – napięcie prądu stałego, I – prąd stały.

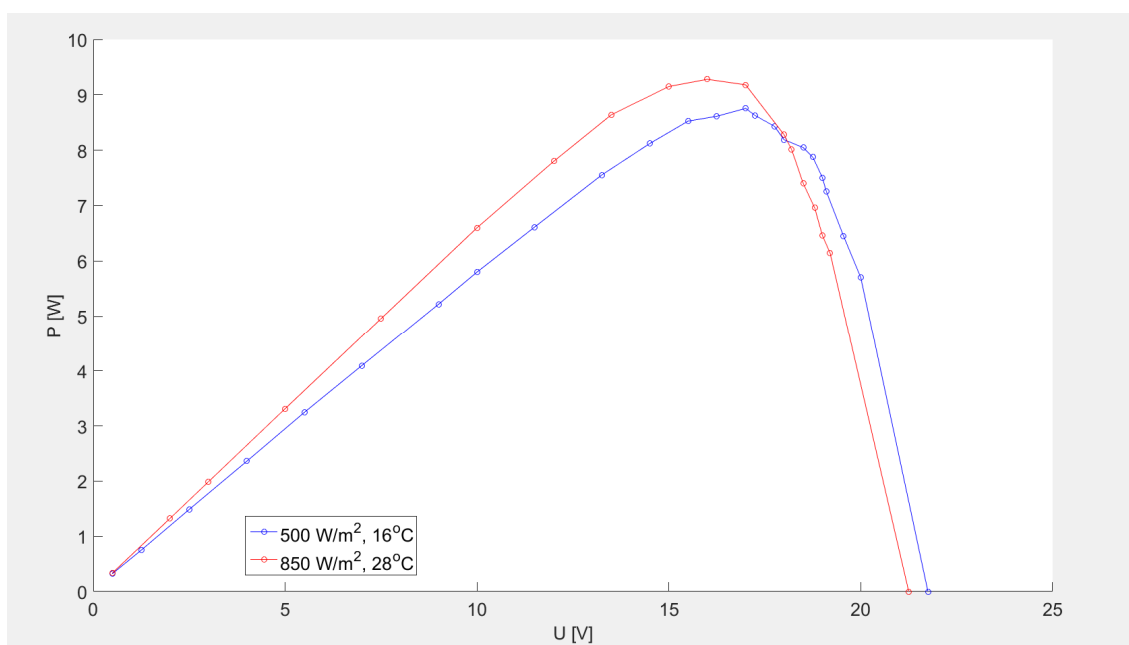
**Tabela 2.2.** Pomiary wartości napięcia i prądu otrzymane w warunkach: 500 W/m<sup>2</sup>, 16°C

Numer Pomiaru	Napięcie [V]	Prąd [A]
1	0,5	0,65
2	1,25	0,6
3	2,5	0,595
4	4	0,59
5	6	0,59
6	7	0,585
7	9	0,58
8	10	0,58
9	11,5	0,575
10	13,25	0,57
11	14,5	0,56
12	15,5	0,55
13	16,25	0,535
14	17	0,515
15	17,25	0,5
16	17,75	0,475
17	18	0,455
18	18,5	0,435
19	18,75	0,42
20	19	0,395
21	19,1	0,38
22	19,55	0,33
23	20	0,285
24	21,75	0



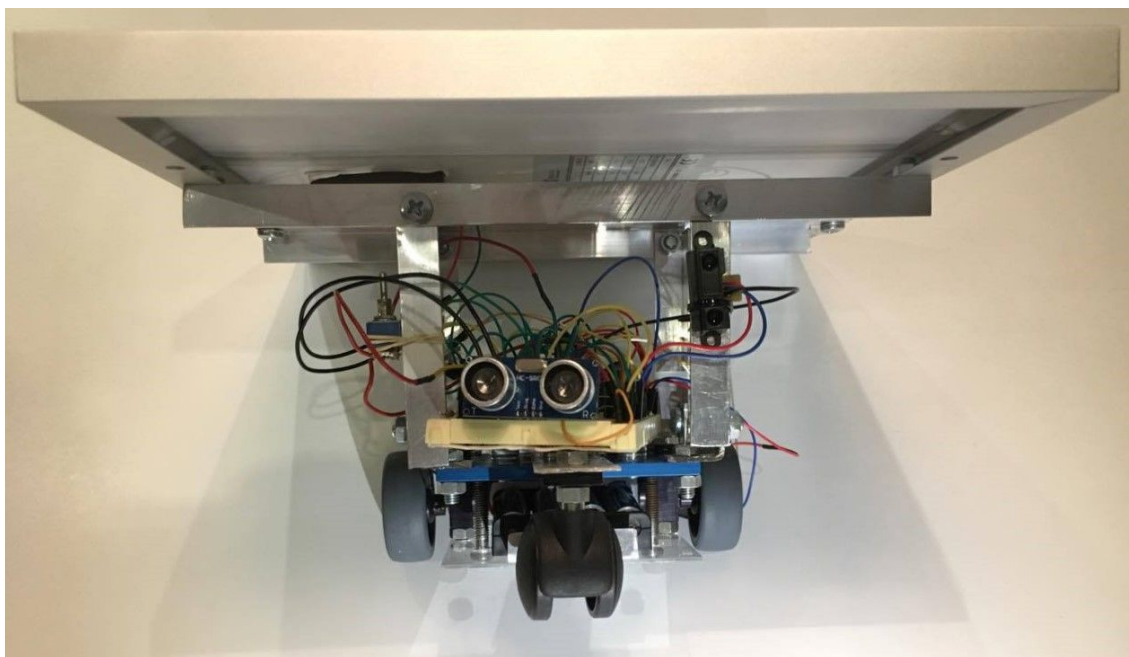


Rys. 2.8. Charakterystyki prądowo-napięciowe panelu fotowoltaicznego



Rys. 2.9. Przebiegi mocy od napięcia panelu fotowoltaicznego

Ważnym aspektem poprawnego działania układu doładowywania akumulatorów przez panel fotowoltaiczny, w trakcie jazdy, było połączenie go z pojazdem, za pomocą jak najłżejszej konstrukcji. W tym celu zastosowano aluminiowe kątowniki, które przykręcono zarówno do panelu jak i do pojazdu, a następnie połączono ze sobą za pomocą śrub oraz nakrętek. Zbudowany szkielet do podtrzymywania konstrukcji został zaprezentowany na rysunku 2.10. Panel został umiejscowiony 20 cm nad pojazdem, aby zredukować wagę wykorzystanych kątowników (krótszy kątownik waży mniej), ale jednocześnie zostawić wystarczająco dużo miejsca dla mikrokontrolera i przewodów. Panel został tak zamontowany, aby w prosty sposób zachować położenie środka ciężkości w centralnej części pojazdu.



*Rys. 2.10. Mały pojazd z aluminiową konstrukcją podtrzymującą panel fotowoltaiczny*

### 3. SYSTEM STEROWANIA POJAZDEM.

W celu zapewnienia autonomicznej pracy małego pojazdu potrzebne jest odpowiednie dobranie metody sterowania, zapewniającej jazdę bez ingerencji człowieka. Zadaniem opisywanego w pracy robota, zwanego „line followerem” jest śledzenie czarnej linii. Aby uzyskać taką możliwość został zbudowany układ sterowania składający się z modułu Arduino Nano, z wbudowanym mikrokontrolerem Atmega328, do którego podłączone zostały trzy czujniki optyczne (odbiciowe).

Zastosowanie dwóch silników prądu stałego podłączonych do sterownika L293D, zbudowanego z dwóch mostków H, umożliwiło obracanie się kół w dwóch kierunkach. W taki sposób można uzyskać możliwość poruszania się samochodziku elektrycznego do przodu i do tyłu. O konkretnym kierunku obrotu kół informują diody LED podłączone do odpowiednich przewodów zasilających silnik, które zostały podpisane po angielski jako „RED LED” wraz z opisem kierunku obrotu konkretnego koła.

Oprócz wyżej wymienionych części zostały również dodane dwa ultradźwiękowe czujniki odległościowe, jeden z przodu, a drugi z tyłu, które umożliwiają wykrycie przeszkody na drodze i zatrzymanie się w odpowiedniej odległości unikając zderzenia. Z lewej strony został zamontowany analogowy czujnik odległościowy podczerwieni, dzięki czemu uzyskano większy poziom bezpieczeństwa. W razie pojawienia się obiektu naprzeciwko pojazdu, uruchamiany jest brzęczyk informujący o zagrożeniu.

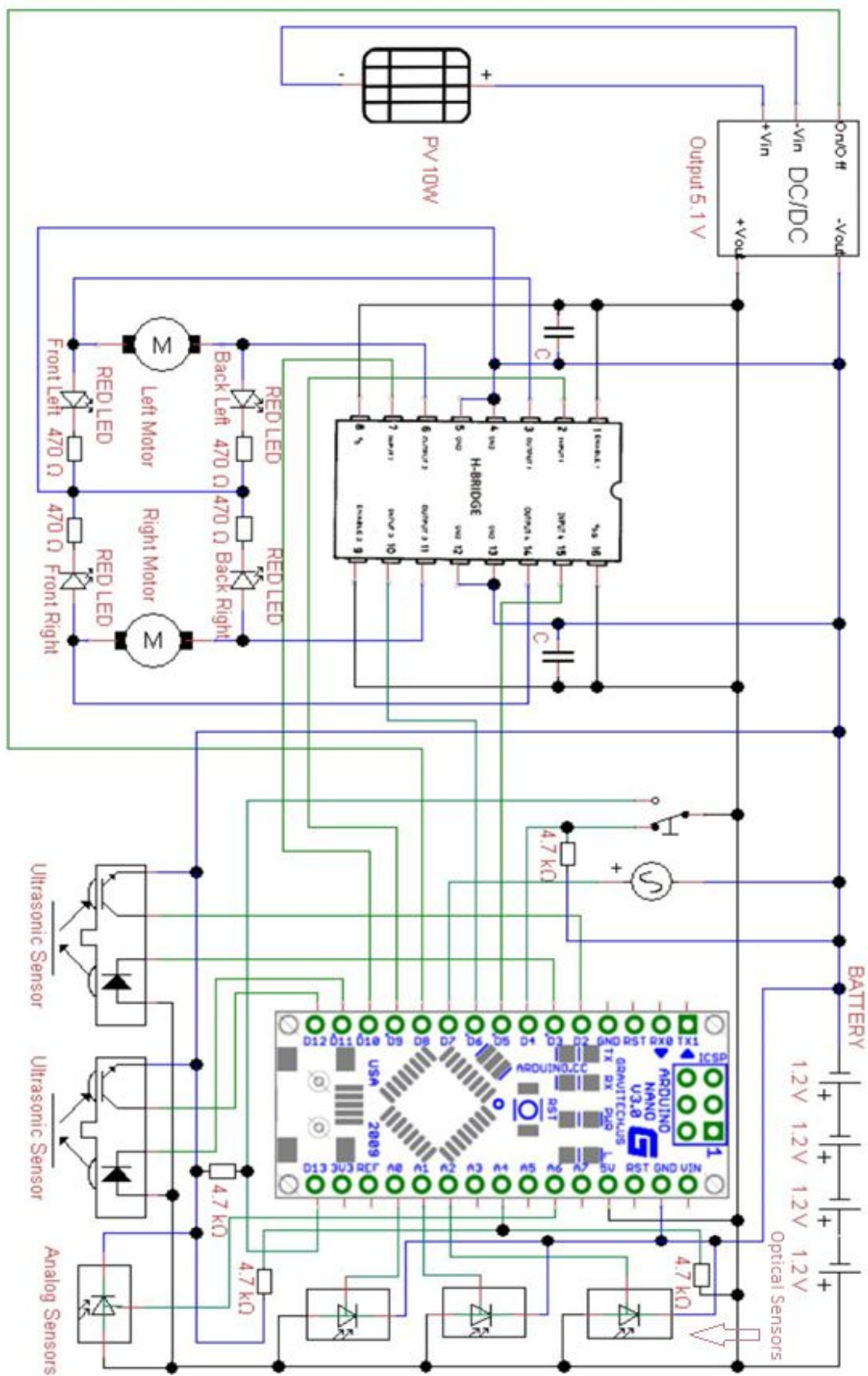
Kierunek jazdy samochodem może być zmieniony za pomocą zamontowanego trójstanowego przełącznika, który został przymocowany do aluminiowego kątownika z boku pojazdu.

Układ zasilany jest akumulatorami niklowo-kadmowymi, podłączonymi do panelu fotowoltaicznego za pomocą przetwornika DC/DC TRACO POWER THL 10-2411WI o mocy 10 W. Aby móc uzyskać informacje o stanie naładowania akumulatorów został zbudowany dzielnik napięcia z dwóch rezystorów 4.7kΩ.

#### 3.1. Schemat elektryczny

Wymienione we wstępie części układu sterowania wraz z elementami elektronicznymi zostały połączone ze sobą za pomocą płytki stykowej i przewodów, zakończonych tulejkowymi końcówkami. Dzięki temu rozwiązaniu z łatwością można modyfikować oraz rozwijać stworzony układ elektryczny wpinając i wypinając elementy. Schemat ostatecznej wersji został przedstawiona na rysunku 3.1. Elementami elektronicznymi niewspomnianymi wcześniej w pracy, a przedstawionymi na schemacie są kondensatory, zmniejszające wahania napięć oraz rezystory zmniejszające wartość przepływającego prądu przez diody. Dwa odprowadzone od wejść D13 i D4 do uziemienia pełnią tak zwaną funkcję „pull down”, czyli obniżają napięcie do zera.

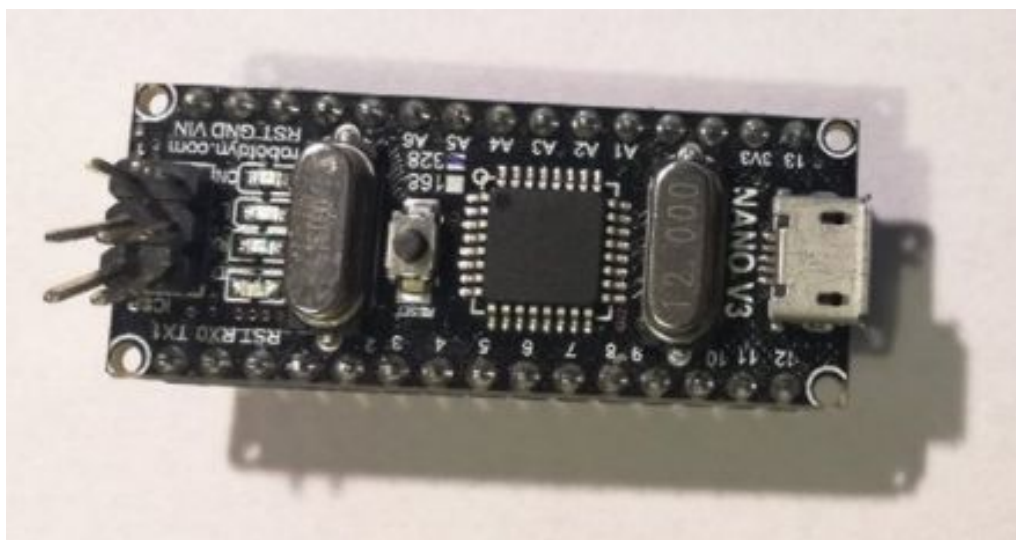
Pojedyncze elementy układu zostały opisane w kolejnych podrozdziałach.



Rys. 3.1. Schemat elektryczny zbudowanego układu sterowania pojazdem [3.1] [3.2]

### 3.2. **Moduł Arduino Nano V3.0**

Arduino przedstawione na rysunku 3.2 jest to platforma programistyczna, która umożliwia sterowanie układem elektronicznym. Posiada ona czternaście pinów cyfrowych oraz osiem analogowych. Programowo należy określić które z nich są wyjściami, a które wejściami do płytki Arduino. Piny analogowe różnią się od wejść/wyjść cyfrowych tym, że przed numerem jest litera A. Sześć wyprowadzeń cyfrowych są kanałami PWM, dzięki którym można czytywać oraz zadawać wartości w zakresie od 0 do 255. Daje to możliwość regulacji pracy silników. W autonomicznym pojeździe są wykorzystane dwa silniki prądu stałego. Modułacja szerokości impulsów, z angielskiego Pulse-Width Modulation, to metoda regulacji sygnału o stałej amplitudzie i częstotliwości. Polega na zmianie wypełnienia sygnału, poprzez zmianę czasu długości impulsu. Zastosowanie zwykłych wyprowadzeń cyfrowych jest prostsze – napięcie 0 V reprezentuje wartość 0 bitów, a napięcie 5 V jest związane z wartością 1 bitu. Wejścia i wyjścia analogowe operują na wartościach od 0 do 1023. Napięciu 5 V odpowiada liczba 1023 bitów, a napięcie 0 reprezentuje wartość 0 bitów. Przedział ten jest związany z rozdzielczością mikrokontrolera wbudowanego na platformie programistycznej Arduino. Centrum platformy jest mikrokontroler Atmega328, o częstotliwości taktowania 16 MHz. Posiada wbudowany przetwornik analogowo-cyfrowy o rozdzielczości 10 bitów. Dzięki temu mikrokontroler posiada 1024 kombinacji możliwych do wykorzystania, przy użyciu pinów analogowych.



*Rys. 3.2. Płytki Arduino Nano wykorzystana w projekcie*

Platforma Arduino Nano ma wymiary 45x19 mm, waży 5 g i posiada między innymi przycisk reset, złącze 27pinowe oraz czip, w którym zbudowany jest mikrokontroler Atmega328. Arduino Nano wyposażony jest w programator CH340. Wgranie programu z komputera na procesor wykonuje się poprzez port mini USB, ale istnieje możliwość zaprogramowania Arduino przez złącze ICSP. Mały rozmiar płytki nie pozwolił producentowi na wypisanie wszystkich oznaczeń wyprowadzeń, dlatego braki zostały one uzupełnione na zdjęciu 3.1. Napięcie zasilania płytki Arduino to 5 V. W przypadku gdy doprowadzone napięcie jest większe, to Arduino posiada stabilizator napięcia obniżający je do znamionowej wartości. W czasie pracy

zapala się zielona dioda oznajmiająca, że platforma została załączona. W przypadku resetu, na moment zapala się niebieska dioda. Zaletami Arduino Nano jest niewielki rozmiar oraz mała waga. Dla porównania wymiary Arduino Leonardo to 80 x 60 mm oraz 28 gram. Nano posiada mniej wyprowadzeń możliwych do wykorzystania, ale 22 wejścia/wyjścia w zupełności wystarczyły do sterowania pojazdem. Inną ważną cechą jest możliwość wpięcia platformy Arduino Nano do płytki stykowej, do której zostały podłączone również inne komponenty potrzebne do stworzenia projektu.

**Tabela 3.1.** Wykaz wyprowadzeń na platformie programistycznej Arduino Nano

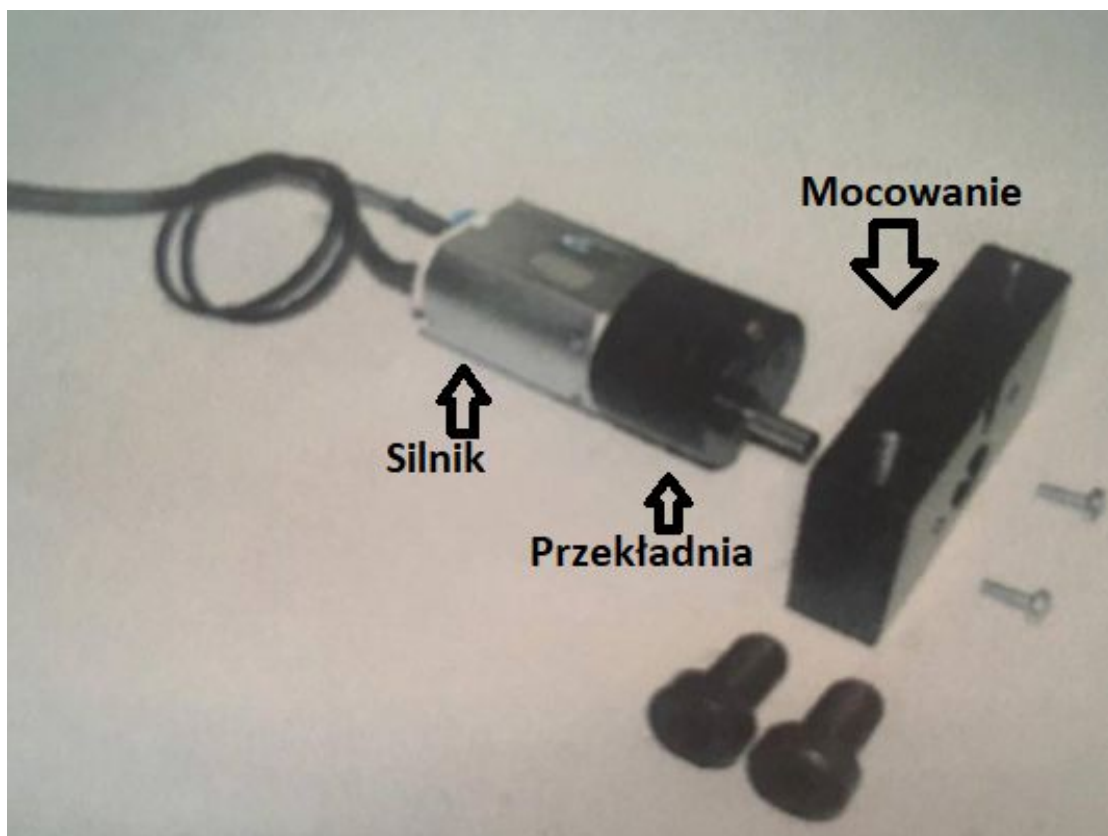
Oznaczenia pinów	Opis
TX1	Wejście cyfrowe, związane z przesyłaniem danych szeregowo, asynchronicznie z komputera do procesora Arduino. Wyjście danych.
RX0	Wejście cyfrowe, związane z przesyłaniem danych szeregowo, asynchronicznie z komputera do procesora Arduino. Wejście danych.
RST	Reset. Sprawia, że zapisany program uruchamia się od nowa.
GND	Masa układu
3,5,6,9,10,11	Piny cyfrowe, kanały PWM
2,4,7,8,12,13	Piny cyfrowe, zwykłe
3v3	Wyjście o napięciu 3,3 V
REF	Dostęp do napięcia z jakim pracują wyprowadzenia wejścia/wyjścia
A0 – A7	Piny analogowe
5V	Wyjście o napięciu 5 V
VIN	Napięcie wejściowe

Arduino oprócz platformy sprzętowej, to również zintegrowane środowisko programistyczne IDE dedykowane do programowania mikrokontrolerów wbudowanych na płytkach Arduino. Oprogramowanie pozwala na tworzenie, modyfikowanie i testowanie programów. Aplikacja cechuje się również łatwym przeniesieniem programu do procesora platformy Arduino. Pozwala również weryfikować poprawność działania programu poprzez funkcję monitorowania danych oraz obserwowanie zmieniających się wykresów w czasie pracy układu. Środowisko posiada wbudowane biblioteki ułatwiające tworzenie kodu. Każdy program Arduino wyróżnia się dwoma podstawowymi funkcjami – *setup* oraz *loop*. Kod, który został zaimplementowany w funkcji *setup* zostanie wykonany tylko raz, ale za każdym razem, po uruchomieniu lub resetowaniu programu. Funkcja *loop* będzie się wykonywała w pętli, cały czas, po zrealizowaniu funkcji *setup*.

### 3.3. Sterownik L293D

Wymienione we wstępie silniki prądu stałego mają moc 3 W i napięcie zasilania od 3,6 V do 7,2 V oraz pracują z przekładnią mechaniczną 1:50. Oznacza to, że po podłączeniu kół prędkość obrotowa, jaką można uzyskać jest 50 razy mniejsza, niż w wypadku, gdyby nie było przekładni. Wadą takiego zastosowania jest mała i ograniczona możliwość zmiany prędkości

poruszania się pojazdu, natomiast zaletą duży moment obrotowy, gdyż jest to niezbędne do poruszania się z dużym obciążeniem. Zostały one wybrane, ponieważ po przymocowaniu panelu o masie 1.1 kilograma układ dużo ważył, a im większa masa tym większe opory ruchu na podstawie wzoru  $T = \mu N$ , gdzie  $T$  – siła tarcia,  $\mu$  - współczynnik tarcia,  $N$  – siła nacisku [N], w przypadku poruszania się po płaskiej powierzchni  $N=G$ , gdzie  $G$  – siła grawitacji [N]. Pojedynczy silnik został przedstawiony na rysunku 3.3.



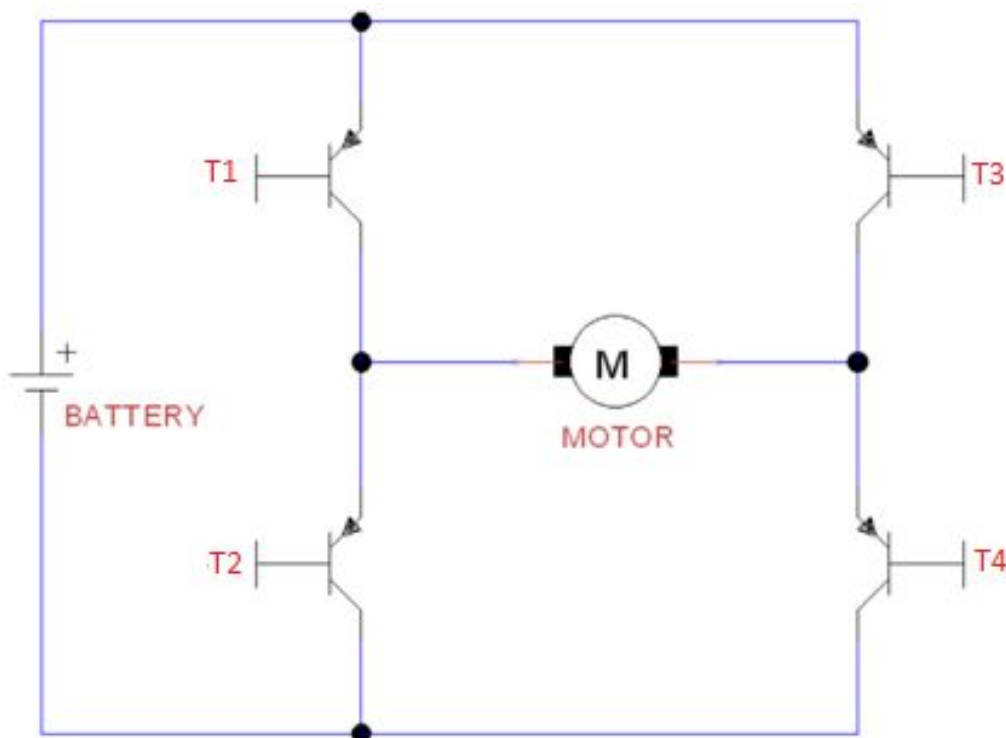
Rys. 3.3. Silnik wraz z przekładnią i mocowaniem

Wykorzystany sterownik L293D, nie tylko zapewnił możliwość skręcania, ale również możliwość zmiany kierunku jazdy do przodu lub do tyłu. Zwrot realizowany jest za pomocą trzystanowego przełącznika, który podłączony jest do dwóch pinów modułu arduino (D13 i D4). Przełącznik, jak sama nazwa wskazuje może znaleźć się w trzech stanach, jeden to przerwa, a dwa pozostałe to zwarcie konkretnego pinu z zasilaniem. Po otrzymaniu napięcia o wartości około 5 V na pinie D13 pojazd porusza się do tyłu, a w przypadku wysokiego stanu logicznego na pinie D4 do przodu. Projekt samochodziku składa się z trzech kół, dwóch napędzanych wymienionymi wyżej silnikami i jednym z możliwością bezwładnego obracania się wokół dwóch osi. Dzięki takiemu zastosowaniu zmiana kierunku jazdy może odbywać się poprzez obracanie jednego koła wolniej od drugiego, całkowite zatrzymanie, albo obrót dwóch kół w przeciwną stronę. Podczas wykonywanych manewrów przednie koło obraca się i zostaje skierowane przodem do kierunku jazdy.

Budowa sterownika oparta jest o dwa układy elektryczne nazywane mostkami H. Podstawowy mostek jest zbudowany z czterech tranzystorów, połączonych ze sobą w kształcie



liter H (stąd pochodzi nazwa). W zależności od wysyłanych sygnałów na konkretne tranzystory, można zapewnić zmianę kierunku przepływu prądu przez silnik, dzięki czemu zmienia się kierunek obrotu. Przykładowo, jeżeli w układzie elektrycznym przedstawionym na rysunku 3.4 tranzystory T1 i T4 będą przewodzić prąd, a T2 i T3 nie, to prąd wymusi pracę silnika i obrót. Po aktywowaniu tranzystorów T2 i T3 oraz dezaktywowaniu T1 i T4 zmieni się kierunek obrotu.



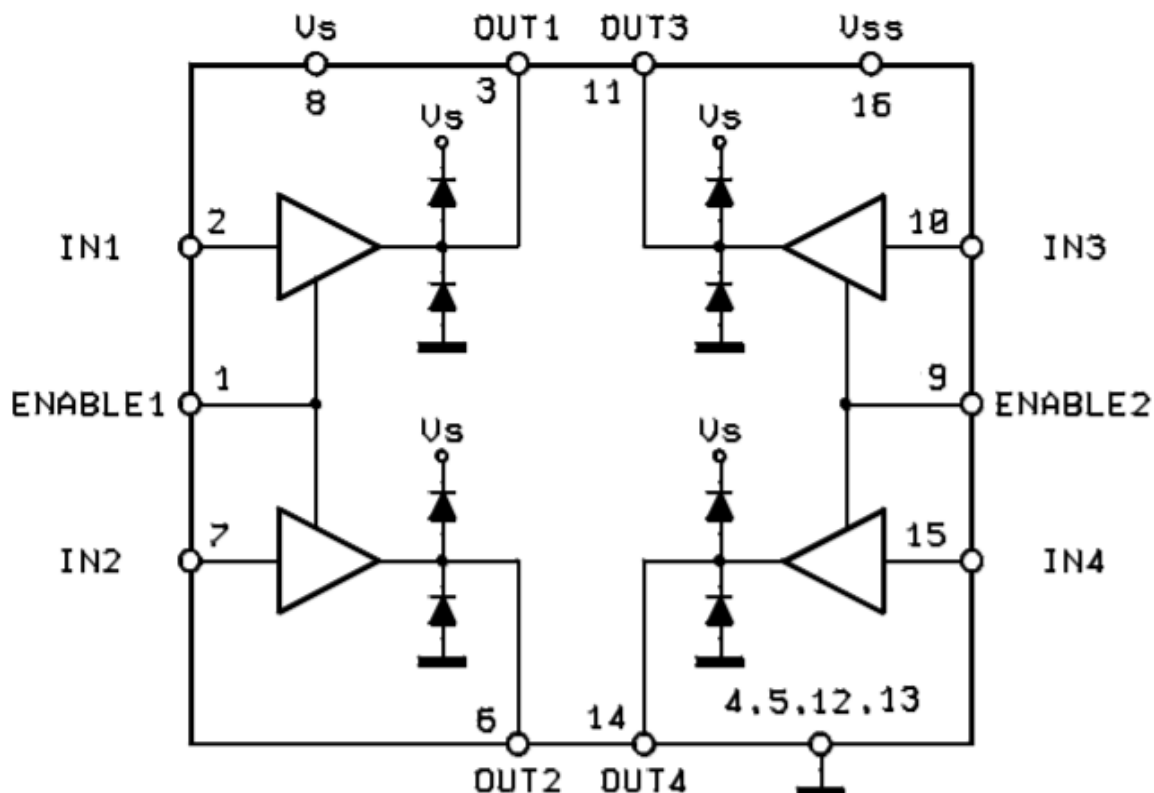
Rys. 3.4. Schemat przykładowego mostka H

Powyższy rysunek został zaprezentowany, aby ułatwić zrozumienie działania układu elektrycznego jaki tworzy sterownik L293D, którego schemat przedstawiono na rysunku 3.5. Jego budowa jest bardziej skomplikowana, ale zasada działania identyczna jak dwóch mostków H. Sterownik jest tak zaprojektowany, aby zajmował mało miejsca oraz umożliwiał sterowanie dwoma silnikami jednocześnie. Numerami od 1 do 16 oznaczono piny, do których podłączone zostały przewody na płytce stykowej. Wejścia ENABLE1 i ENABLE2 podpisane numerami 1 i 9 sterują (za pomocą funkcji PWM) wartością napięcia, które później jest przekazywane na silniki przez wyjścia; dla Enable1 jest to OUT1 (3) lub OUT2 (5), a dla Enable2 OUT3 (11) lub OUT4 (14). Im większe jest napięcie na konkretnym wejściu tym większe napięcie zostanie dostarczone na silnik, co spowoduje wzrost prędkości obrotowej. W przypadku opisywanego pojazdu wartości te są na stałe podłączone do zasilania z baterii. Prędkość obrotu jest zmieniana za pomocą wyjść cyfrowych PWM modułu arduino, z których sygnał jest wysyłany na wejścia IN1(2) lub IN2 (7) dla jednego silnika oraz IN3 (10) lub IN4 (14) dla drugiego. Głównym ich zastosowaniem jest zmiana kierunku poprzez wysyłanie napięcia z arduino, albo na jeden albo na drugi pin. W pojeździe użyto czterech wejść zmiany kierunku, również do zmiany prędkości, co umożliwiło zaoszczędzenie dwóch wyjść cyfrowych mikrokontrolera.



Miejsce podłączenia zasilania układu logicznego  $V_s$  zostało oznaczone numerem 8, a zasilania silników  $V_{ss}$  numerem 16. Pozostałe numery 4, 5, 12, 13 posłużyły do oznaczenia pinów uziemienia.

Przedstawiony poniżej rysunek powstał po przekształceniu schematu elektrycznego sterownika L293DD zaprezentowanego w dokumentacji. L293DD różni się od L293D, tym że ma o cztery więcej miejsc podłączeń uziemienia.



Rys. 3.5. Schemat elektryczny sterownika L293D [3.2]

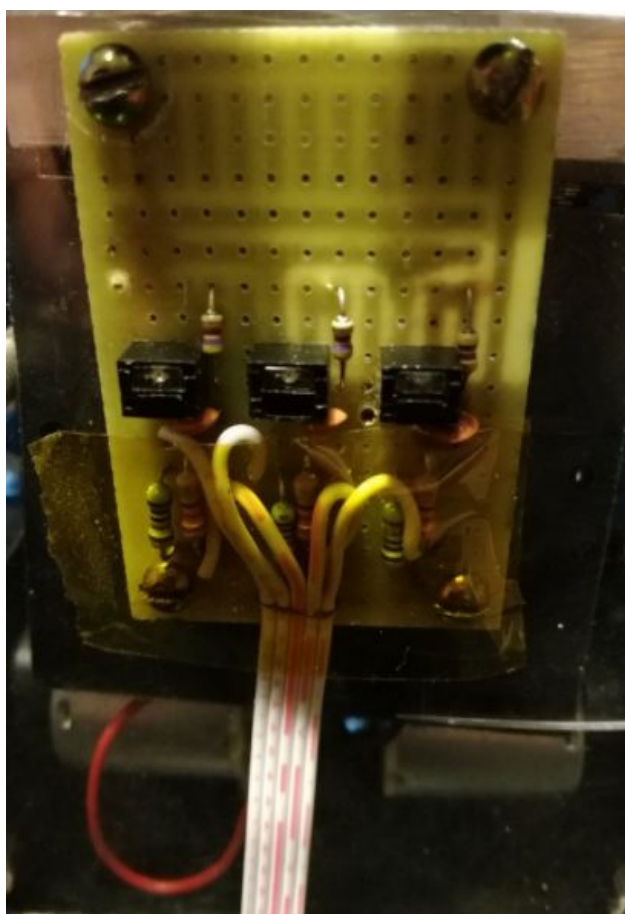
### 3.4. Czujniki odbiciowe

Oznaczone na rysunku 3.1 podpisem „optical sensors” tak zwane czujniki odbiciowe, zostały umieszczone pod spodem pojazdu na oddzielnej płycie stykowej. Aby zminimalizować ilość przewodów schematu elektrycznego opisanego w podrozdziale 3.1 czujniki zostały przedstawione jako trzy oddzielne elementy (prostokąt z namalowanym w nim symbolem fotodiody półprzewodnikowej) z doprowadzonymi do każdego trzema przewodami. W rzeczywistości połączenie pomiędzy całym układem, a płytą stykową na której zamontowano czujniki odbiciowe odbyło się za pomocą pięciu przewodów, co można zaobserwować na rysunku 3.5.

Budowa czujnika odbiciowego charakteryzuje się umieszczeniem emitera oraz detektora w tej samej obudowie. Działanie polega na emisji wiązki promieniowania oraz odebranie odbitego od określonej powierzchni promienia. Można je stosować w celu wykrywania obiektów na mniejsze odległości niż czujniki ultradźwiękowe.

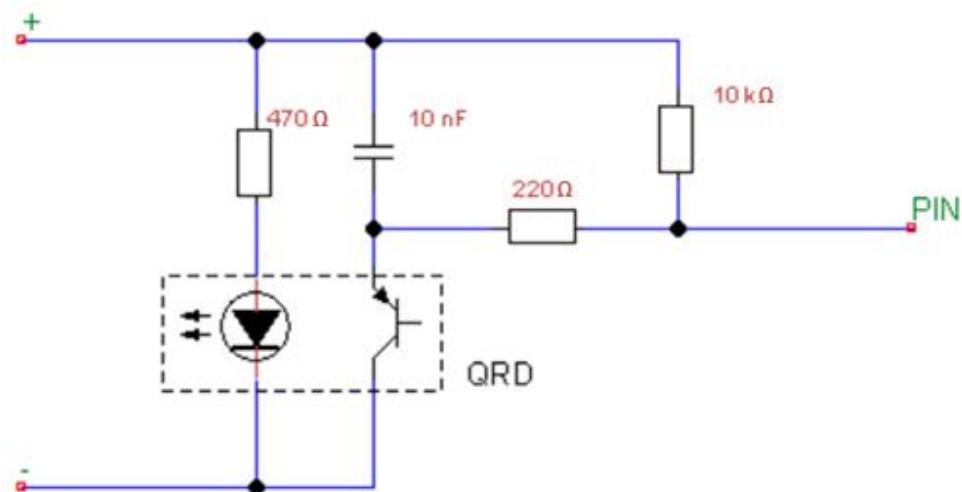
W pojeździe funkcjonującym jako „line follower”, który jest przedmiotem danej pracy inżynierskiej zadaniem jest wykrywanie linii. Rodzajem czujnika odbiciowego jaki został wykorzystany w tym celu jest QRD1114. Role emitera pełni dioda wytwarzająca promieniowanie podczerwone, a detektora fototranzystor, są one umieszczone obok siebie [3.3].

Dzięki cesze koloru czarnego jaką jest pochłanianie promieniowania podczerwonego można wykorzystać opisane czujniki odbiciowe do wykrywania koloru czarnego i białego. Ułożenie trzech obok siebie jak przedstawiono na rysunku 3.6 oraz stworzenie odpowiedniego układu sterowania wraz z wgranym programem na mikrokontroler może zapewnić śledzenie czarnej lub białej linii. Kolor biały znajdujący się pod czujnikiem spowoduje odbicie wiązki promieniowania i wykrycie jej przez detektor. Padając na fototranzystor zwiększa przepływ prądu przez niego co skutkuje zmniejszeniem napięcia na pinie modułu arduino.



*Rys. 3.6. Płytkę stykową z układem elektrycznym obsługującym trzy czujniki optyczne w kolorze czarnym umieszczone na środku płytki.*

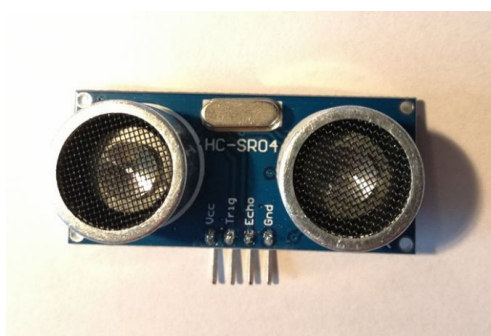
Do zbudowania układu elektrycznego, który został przedstawiony na rysunku powyżej posłużyły 3 rezystory o wartościach 470  $\Omega$ , 220  $\Omega$ , 10 k $\Omega$  i jeden kondensator 10 nF. Zaprezentowanie połączeń elementów elektronicznych biernych wraz z fototranzystorem (symbol tranzystora) i diodą (symbol emitującej diody) zaprezentowano na rysunku 3.7. Czujnik odbiciowy został otoczony przerywaną linią tworzącą prostokąt i podpisany jako „QRD”



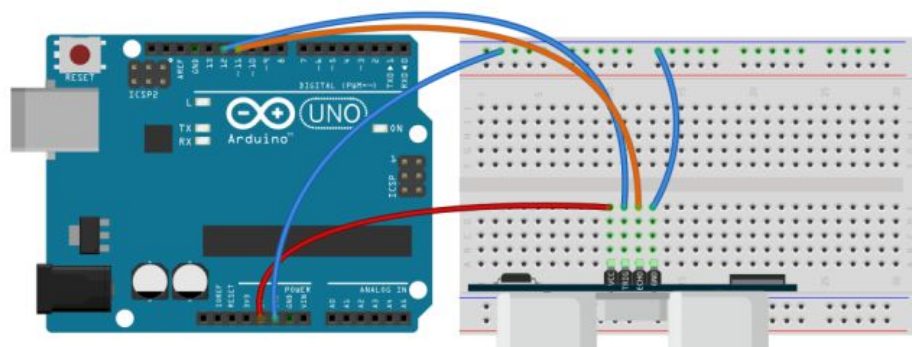
Rys. 3.7. Schemat elektryczny połączeń czujnika odbiciowego QRD1114

### 3.5. Zbliżeniowy czujnik ultradźwiękowy

Jednym z rodzajów czujników, który został wykorzystany w projekcie, jest czujnik ultradźwiękowy przedstawiony na rysunku 3.8. Popularnym modelem jest HC-SR04 i właśnie ten został użyty do wykrywania odległości. Niewielkich rozmiarów czujnik, 45 x 20 x 15 mm, pozwala wykryć obiekt znajdujący się nawet 200 cm od robota. Zasilany jest napięciem 5V, czyli tym samym napięciem, które zasila cały układ. Zużywa mało prądu, jedynie 15 mA w momencie wykonywanego pomiaru. Na rysunku 3.9 zaprezentowano przykładowy sposób podłączenia czujnika do arduino.



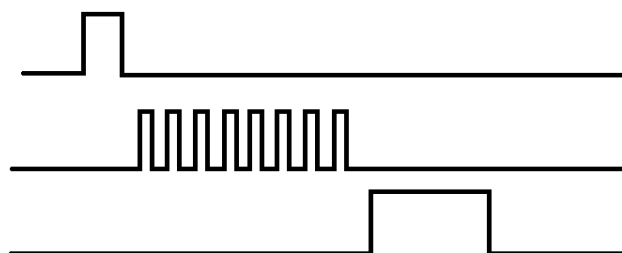
Rys.3.8. Czujnik ultradźwiękowy HC-SR04, wykorzystany w projekcie



Rys. 3.9. Przykładowe połączenie czujnika HC-SR04 z Arduino Leonardo [3.4]

Czujnik posiada 4 piny: VCC, do którego należy podpiąć zasilanie 5V, Trig –wejście, Echo – wyjście, Gnd – masa. Zasada działania polega na wysyłaniu sygnału wysokiego o czasie 10  $\mu$ s na pin Trig, czyli trigger, z angielskiego wyzwalacz. Nadajnik wysyła wówczas wiązkę ultradźwiękową złożoną z 8 impulsów o częstotliwości 40 kHz. Fala dźwiękowa po napotkaniu obiektu na swojej drodze wraca do odbiornika. Pin echo jest tak długo w stanie wysokim, dopóki sygnał nie powróci do odbiornika.

Zasada działania czujnika ultradźwiękowego przedstawiona na rysunku 3.10. opiera się na zjawisku echa. Fala dźwięku rozprzestrzenia się z prędkością około 340  $\left[\frac{m}{s}\right]$ . Prędkość ta jest zależna od takich warunków atmosferycznych jak temperatura, wilgotność oraz ciśnienie.



Rys. 3.10. Przedstawienie działania czujnika HC-SR04

Pierwszy sygnał, to załączenie czujnika poprzez Trigger. Drugi sygnał to maksymalnie 8 impulsów o częstotliwości 40 kHz. Ostatni przebieg przedstawia sygnał wyjściowy z czujnika. Długość tej odpowiedzi jest zależna od odległości między czujnikiem, a badanym obiektem. Poprzez odpowiednią funkcję można określić jak daleko znajduje się dany przedmiot. Odległość obliczana jest zgodnie z algorytmem (3.1).

$$L = T[\mu s] * \frac{340\left[\frac{m}{s}\right]}{2} \quad (3.1)$$

gdzie:

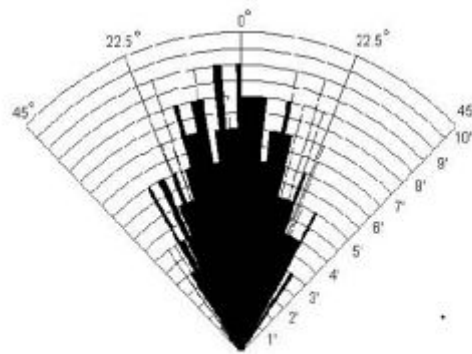
L – odległość przedmiotu od czujnika [cm]

T – czas, w którym pin echo jest w stanie wysokim [ $\mu$ s]

W liczniku znajduje się prędkość dźwięku wyrażona w  $\left[\frac{m}{s}\right]$ . Czas jest zliczany od momentu wysłania przez nadajnik fali ultradźwiękowej, do momentu gdy fala ta powróci do odbiornika. Aby otrzymać informację jak daleko znajduje się obiekt, potrzebna jest jedynie połowa tego czasu. Wzór (3.2) na wyrażenie odległości w centymetrach znajduje się poniżej.

$$L = 170 * T[\mu s] * \frac{100[cm]}{10^6[\mu s]} = T * \frac{17}{10^3} [cm] = T * 0.017 [cm] \quad (3.2)$$

Największą precyzję określania odległości pomiędzy czujnikiem, a badanym przedmiotem otrzymuje się jeżeli dany obiekt znajduje się naprzeciwko czujnika. Czujnik HC-SR04 posiada zdolności wykrywania przeszkody znajdującej się maksymalnie pod kątem 30 stopni względem wysyłanej wiązki ultradźwiękowej. Jeżeli przedmiot nie znajduje się w tym obszarze, to czujnik go nie wykryje. Na rysunku 3.11 zaprezentowano detekcję fali ultradźwiękowej.



Practical test of performance,  
Best in 30 degree angle

Rys 3.11. Katalogowa charakterystyka promieniowania czujnika HC-SR04

### 3.5.1 Kod programu do obsługi czujnika ultradźwiękowego.

Poniżej na rysunku 3.12 został zaprezentowany przykładowy program do obsługi czujnika. Podobny kod, lecz po pewnych konfiguracjach, został wykorzystany w głównym programie projektu.

```
int trig = 2;
int echo = 3;

void setup() {
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  digitalWrite(trig, LOW);
  Serial.begin(9600);
  Serial.println("odległość czujnika od obiektu wyrażona w centymetrach");
}

void loop() {
  int T;
  int L;
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  T = pulseIn(echo, HIGH);
  L = T * 0.017;
  Serial.println(L);
  delay(500);
}
```

Rys. 3.12. Przykładowy program do czujnika ultradźwiękowego [3.5]

Na początku programu zostały zainicjalizowane zmienne globalne trig, echo i przypisane do odpowiednich pinów. Numery pinów w programie muszą być zgodne z połączeniem elektronicznym. Nadajnik trig może zostać podłączony do pinu cyfrowego, ale odbiornik musi zostać połączony z pinem umożliwiającym odczytywanie sygnału przez dłuższy czas. Dlatego został on podłączony z pinem 3, który obsługuje funkcje PWM (Pulse Width Modulation). Zmienna trig została potraktowana jako wejście do programu, a zmienna echo jako wyjście. Przy każdym uruchomieniu programu pin 2 będzie w stanie niskim.

Zasada działania programu polega na wysłaniu sygnału wysokiego, o czasie 10 mikrosekund, na pin 2.

Dzięki funkcji `pulseIn` program szczytuje czas, w którym zmienna `echo` miała wartość wysoką. Funkcja `pulseIn` jest dostępna w bibliotece środowiska Arduino. Po odpowiednim przekształceniu otrzymujemy poszukiwaną odległość pomiędzy czujnikiem, a przedmiotem. Wyrażona jest w wartości od 0 do 1023, gdzie im większa liczba tym bliżej znajduje się obiekt. Komenda `Serial.begin(9600)` pozwala na komunikację szeregową z komputerem. Dzięki temu na monitorze można obserwować jak daleko znajduje się dany obiekt, przykład przedstawiono na rysunku 3.31.

```
odległość czujnika od obiektu
94
95
95
95
95
95
95
97
107
19
13
9
18
5
4
3
3
3
5
7
10
15
107
96
57
96
96
96
96
```

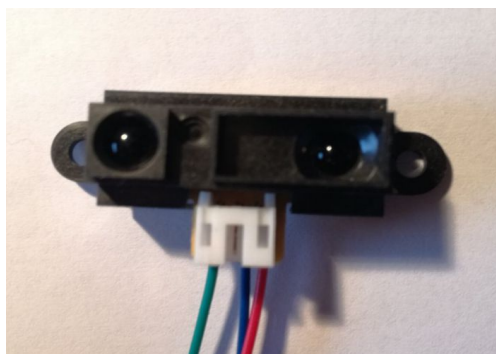
Rys. 3.13. Prezentacja pomiarów otrzymanych przy badaniu czujnika

### 3.6. Zbliżeniowy czujnik podczerwieni

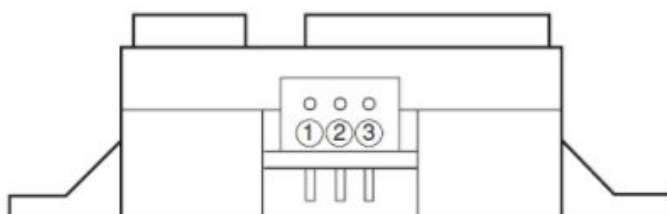
Kolejnym czujnikiem wykorzystanym w projekcie jest analogowy czujnik podczerwieni marki Sharp. Model GP2Y0A02YK0F jest popularnym czujnikiem wykorzystywanym do wykrywania obiektów.

Zasięg to od 10 do 80 centymetrów. Zasilany jest napięciem od 4,5V do 5,5V, więc tym samym jak cały układ. Średni pobór prądu to 30mA, a czas odpowiedzi to 5ms. Czujnik ten jest niewielkich rozmiarów – 29,5 x 13 x 13,5mm.

Posiada 3 wyprowadzenia, – zasilanie `Vcc`, masę `GND` oraz `pin`, dzięki któremu można odbierać sygnał wyjściowy, czyli odpowiedź jak daleko dany obiekt się znajduje, ich opis przedstawiono na rysunku 3.15.



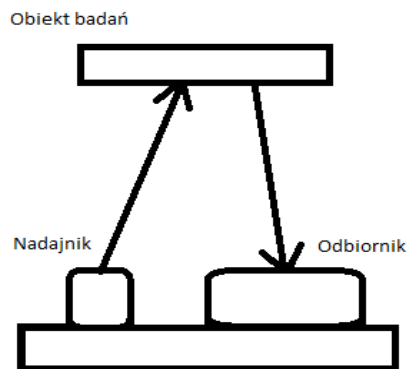
Rys.3.14. Czujnik podczerwieni GP2Y0A02YK0F, wykorzystany w projekcie



PIN	SIGNAL NAME
①	$V_O$
②	GND
③	$V_{CC}$

Rys 3.15. Opis wyprowadzeń czujnika GP2Y0A02YK0F [3.6]

Zasada działania czujnika polega na wysłaniu wiązki światła podczerwieni. Nadajnikiem jest dioda LED, która wysyła wiązkę fali elektromagnetycznej, które mieści się w paśmie światła podczerwieni. W momencie gdy fala wykryje obiekt, to odbija się od jego powierzchni i powraca do odbiornika. Odbiornikiem jest czujnik światła reagujący na zmieniające się napięcie. Im obiekt jest bliżej, tym wyższe napięcie generuje się na czujniku. Na rysunku 3.16 zaprezentowano uproszczone działanie.

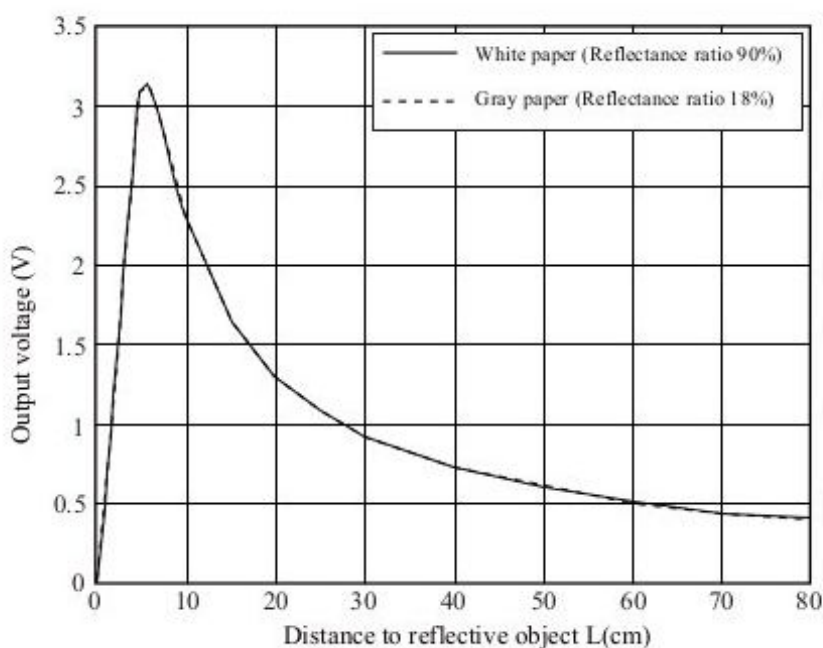


Rys 3.16. Uproszczony system detekcji zbliżeniowej

Czujnik zasilany jest napięciem 5 V. Maksymalne napięcie, które może zostać wygenerowane na odbiorniku wynosi, w przybliżeniu, 3,2 V. Charakterystyka napięcia, w

zależności od odległości przedmiotu od czujnika, zmienia się nieliniowo i została zaprezentowana na rysunku 3.17. Współczynnik odbicia światła podczerwonego od obiektu o kolorze białym wynosi 90%, a dla przedmiotu szarego 18%. Wynika to ze zjawiska pochłaniania światła przez ciała czarne. Wiązka światła padająca na daną powierzchnię jest absorbowana przez nie. Ciało doskonale czarne, które w rzeczywistości nie istnieje, całkowicie pochłania padające na nie światło.

W przeciwieństwie do obiektów czarnych, przedmioty białe w większym stopniu odbijają niż pochłaniają światło. Sygnał odpowiedzi czujnika analogowego podczerwieni GP2Y0A02YK0F jest w przybliżeniu identyczny niezależnie od koloru badanego przedmiotu.



Rys 3.17. Katalogowy wykres napięcia zależnego od odległości [3.6].

### 3.6.1. Kod programu do obsługi czujnika podczerwieni

Do obsługi analogowego czujnika podczerwieni potrzebne jest podłączenie pinu wyjściowego  $V_o$  do wejścia analogowego na płytce Arduino. W przykładowym programie został on podłączony do analogowego pinu numer 6. Po każdym pomiarze odczekujemy 250 milisekund.

Przetwornik mikroprocesora Arduino jest 10 bitowy. Najmniejszą analogową wartością jest 0, a największą 1023. W programie do obsługi czujnika podczerwieni zastosowano funkcję woltomierza, aby sprawdzić jak napięcie generowane na odbiorniku będzie się zmieniało w sytuacji gdy przeszkoda będzie się zbliżała i oddalała od pojazdu. Zgodnie z wykresem 3.17, wiemy że maksymalne napięcie jakie może zostać wygenerowane na czujniku wynosi 3,2 V. Napięcie 5 V doprowadzone do mikrokontrolera, odpowiada liczbie 1023 bitów. Zgodnie z tymi informacjami można zastosować przekształcenie umożliwiające sczytywanie zmieniającego się



napięcia. Im bliżej badany przedmiot się znajduje, tym większe napięcie zostanie podane na mikrokontroler Arduino Nano. Napisany kod został przedstawiony na rysunku 3.18.

```
int analogSensor = A6;
void setup() {
  pinMode(analogSensor, INPUT);
  Serial.begin(9600);
  Serial.println("Test czujnika odleglosci na podczerwień");
}
void loop() {
  float object;
  float voltage;
  object = analogRead(analogSensor);
  voltage = object * (5.0/1023.0);
  Serial.println(voltage);
  delay(250);
}
```

Rys. 3.18. Przykładowy program obsługi czujnika podczerwieni GP2Y0A02YK0F

$$U = X [b] * 5 [V] \div 1023 [b] \quad (3.3)$$

gdzie:

$U$  – napięcie generowane na czujniku (w programie odpowiada zmienna *voltage*)

$X$  - wartość wyrażona w bitach, zmieniająca się w zależności od odległości między czujnikiem, a przedmiotem (w programie odpowiada zmienna *object*)

Wzór (3.3) pozwala obliczyć napięcie będące na pinie mikrokontrolera, podłączonego do czujnika, a następnie na przykład wyświetlić je na komputerze, dzięki funkcji „monitora portu szeregowego”. Przykład przedstawiono na rysunku 3.19.

```
Test czujnika odleglosci na podczerwień
0.05
0.06
0.05
0.05
0.47
0.04
0.55
1.04
1.63
2.38
2.96
2.48
2.15
1.92
2.44
2.84
1.95
1.23
0.73
0.25
0.11
0.09
0.05
0.04
0.13
0.05
0.06
```

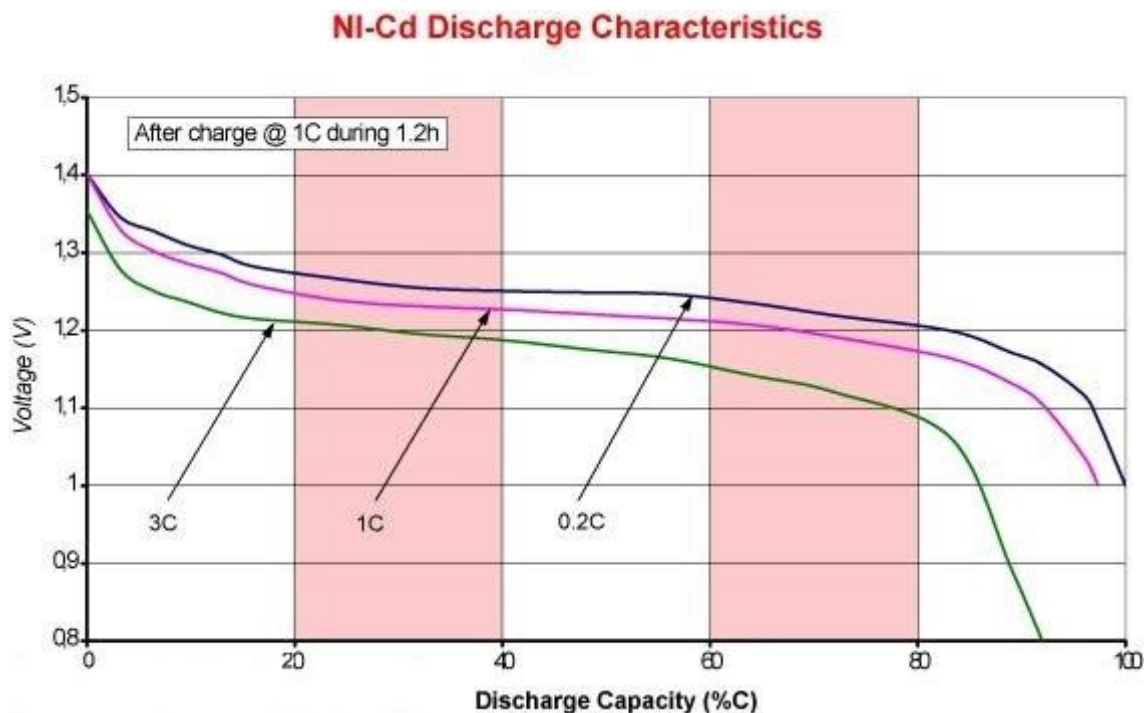
Rys.3.19. Tablica pomiarów z badania działania czujnika podczerwieni

### 3.7. Akumulatory niklowo-kadmowe (Ni-Cd)

Do magazynowania energii uzyskanej z panelu fotowoltaicznego wykorzystano akumulatory niklowo-kadmowe. Pojedyncze ogniwo akumulator ma napięcie nominalne o wartości 1,2 V. Według artykułu „Sprawdzone sposoby na mierzenie stanu baterii” zakres zmiany napięcia pojedynczego ogniwa mieści się w zakresie od 1V do 1,4V [3.7]. Dla górnej wartości jest w pełni naładowane, a dla dolnej rozładowane.

Charakterystyka rozładowywania się pojedynczego ogniwa została zaprezentowana na rysunku 3.20, na którym można zaobserwować trzy różniące się od siebie wykresy. Każdy opisano za pomocą tak zwanego wskaźnika „C-rate”, określającego czas, w jakim ładowany/rozładowywany jest akumulator, przez konkretną wartość prądu. Im większy jest wskaźnik C, tym większy jest prąd ładowania/rozładowania oraz krótszy czas pracy ogniwa. Wskaźnik został opisany w artykule „What Is C-rate” [3.8]. Według legendy charakterystyk 1C odpowiada około 1 godzinie i 12 minutom pracy akumulatora, czyli 0.2C będzie odpowiadać 6 godzinom. Z wykresu wynika, że im mniejszy prąd będzie pobierany w dłuższym okresie czasu tym wolniej będzie spadać napięcie [3.7].

Wykorzystane w projekcie akumulatorki mają pojemność 2300 mAh. Oznacza to, że przy pobieranym ciągłym prądzie o wartości 2,3 A, zasilania wystarczy na godzinę. Sytuacja odpowiadałaby wykresowi oznaczonemu jako 1C. Z badań wykonanych za pomocą multimetru wynika, że w trakcie jazdy z panelem fotowoltaicznym pojazd pobiera z akumulatorów prąd o wartości 0,4 A. W takim stanie pracy pojemność wystarczyłaby na około 6h. Najbardziej zbliżona do tych warunków pracy akumulatorów charakterystyka została oznaczona jako 0.2C.



Rys. 3.20. Charakterystyki rozładowania ogniwa niklowo-kadmowego [3.7].

Na schemacie elektrycznym pokazanym w podrozdziale 3.1 na rysunku 3.1 akumulatorki przedstawiono jako 4 oddzielne ogniwa, a przy każdym podano wartość 1.2 V. Całość podpisano jako „BATTERY”.

### 3.8. Przetwornica DC/DC

Zgodnie z książką autorstwa Ödöna Ferenczi [3.8] konwertery, czyli przetwornice i przekształtniki napięcia stałego składają się z falownika oraz prostownika. Napięcie wyjściowe zależne jest od sposobu budowy układu elektronicznego, od napięcia wejściowego, obciążenia oraz temperatury otoczenia. Do budowy pojazdu została wykorzystana izolowana przetwornica prądu stałego obniżająca napięcie. Przetwornica zabezpiecza układ elektroniczny przed zbyt dużym napięciem. Zasada działania przetwornicy impulsowej DC – DC polega na zmianie napięcia stałego na napięcie stałe o innych parametrach. Impulsowe działanie przetwornicy pozwala na mniejszy pobór prądu, dzięki czemu uzyskuje się lepszą sprawność niż przy użyciu zwykłego stabilizatora.

Maksymalne napięcie wygenerowane przez panel fotowoltaiczny, jakie może zostać dostarczone do akumulatorów wynosi 22 V. Praca znamionowa akumulatorów niklowo-kadmowych to 4,8 V. Należało więc, zabezpieczyć układ elektryczny poprzez dołączenie przetwornicy prądu stałego obniżającej napięcie, przedstawionej na rysunku 3.21. Zakres napięcia wejściowego to  $9 \div 36$  V, a napięcie wyjściowe wynosi 5,1 V. Oznacza to, że jeżeli na panelu fotowoltaicznym zostanie wygenerowane napięcie wyższe niż 9 V i mniejsze niż 22 V, to zostanie ono obniżone do 5,1 V. Prąd wyjściowy z przetwornicy to 2A. Sprawność urządzenia wynosi 84%. Wyprowadzenia przetwornicy to zasilanie (lewa część urządzenia) oraz napięcie dostarczane do układu (prawa część). Istnieje możliwość wyłączania przetwornicy programowo, przyłączono więc wyprowadzenie „On/Off” do pinu numer 7 na platformie Arduino.



Rys. 3.21 Przetwornica izolowana DC/ DC podłączona do układu elektronicznego

Do regulacji napięcia zasilania pojazdu elektrycznego została zastosowana izolowana przetwornica DC – DC. Separacja galwanicznej wyjścia do wejścia daje większe bezpieczeństwo układu, poprzez zastosowanie transformatora, który po stronie elektroniki posiada mniej zwojów, dzięki czemu napięcie zostanie obniżone do napięcia znamionowego. W sytuacji gdy część przetwornicy, po stronie wyższego napięcia, zostanie uszkodzona, to dalsza część układu elektronicznego jest bezpieczna – prąd zwarcia nie przedostanie się dalej.

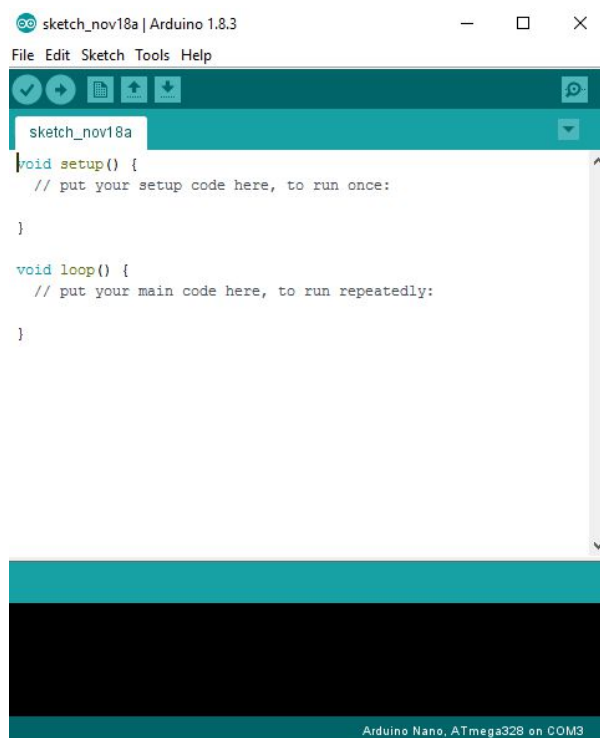
## 4. KODY PROGRAMÓW STERUJĄCYCH

Kolejny rozdział pracy został poświęcony opisowi dwóch kodów napisanych dla różnych warunków pracy. Podrozdziały od 4.1 do 4.9 zostały napisane w celu objaśnienia działania pierwszej wersji programu, która została przystosowana do pracy samochodu poruszającego się po grubszej linii (około 3cm). Drugą wersję programu, opisaną w podrozdziałach od 4.10 do 4.12, przystosowano do funkcji śledzenia cieńszej linii (około 1 cm), skutkiem czego wersja jest mniej skomplikowana, a samochód nie jeździ tak płynnie jak w przypadku poprzednim.

Kody programu zostały napisane przez dwóch różnych autorów, z czego wynikają znaczące różnice w stylu pisania. Mimo tego, zasada działania jest bardzo podobna. Zadaniem w obu wersjach jest śledzenie czarnej linii i reagowanie na przeszkody. Głównym sposobem, jaki został zastosowany, w celu uniknięcia kolizji, jest zatrzymanie pojazdu w przypadku małego dystansu od obiektu, z którym mogłoby nastąpić zderzenie. Dodatkowo w razie zagrożenia zbliżającego się z naprzeciwka aktywowany jest brzęczyk, symulując ostrzeganie pasażerów pojazdu o niebezpieczeństwie, jak mogłoby to się odbywać w rzeczywistym samochodzie osobowym.

Główną różnicą w programach jest funkcja zawierająca kod programu przetwarzający dane, otrzymane z czujników odbiciowych, umieszczonych pod pojazdem. W przypadku pierwszej wersji programu opisano 6 przypadków położenia samochodu względem czarnej linii, a w przypadku drugiej 3. Dokładniej zostało to opisane w dalszej części rozdziału.

Pisanie kodu programu odbywało się w środowisku „Arduino IDE”, którego interfejs został zaprezentowany na rysunku 4.1. Wgrywanie programu na mikrokontroler, umożliwił przewód USB, łączący moduł Arduino z komputerem za pomocą portu.



Rys 4.1. Aplikacja środowiska Arduino IDE

#### 4.1. Opis programu pierwszego

Kod programu zaprezentowany poniżej służy do obsługi pojazdu automatycznego typu line follower. Jego głównym zadaniem jest podążanie za czarną linią. Kod obsługuje również czujnik ultradźwiękowy oraz podczerwieni. Zastosowano przełącznik trójstanowy, którego zaprogramowanie daje możliwość poruszania się robota w przód, w tył oraz postój. Przy inicjalizacji programu, pinom przypisano nazwy zmiennych. Wykaz zmiennych został zaprezentowany w tabeli 2.1 oraz 2.2.

**Tabela 4.1.** Wykaz nazw zmiennych oraz połączeń z pinami sterownika Arduino

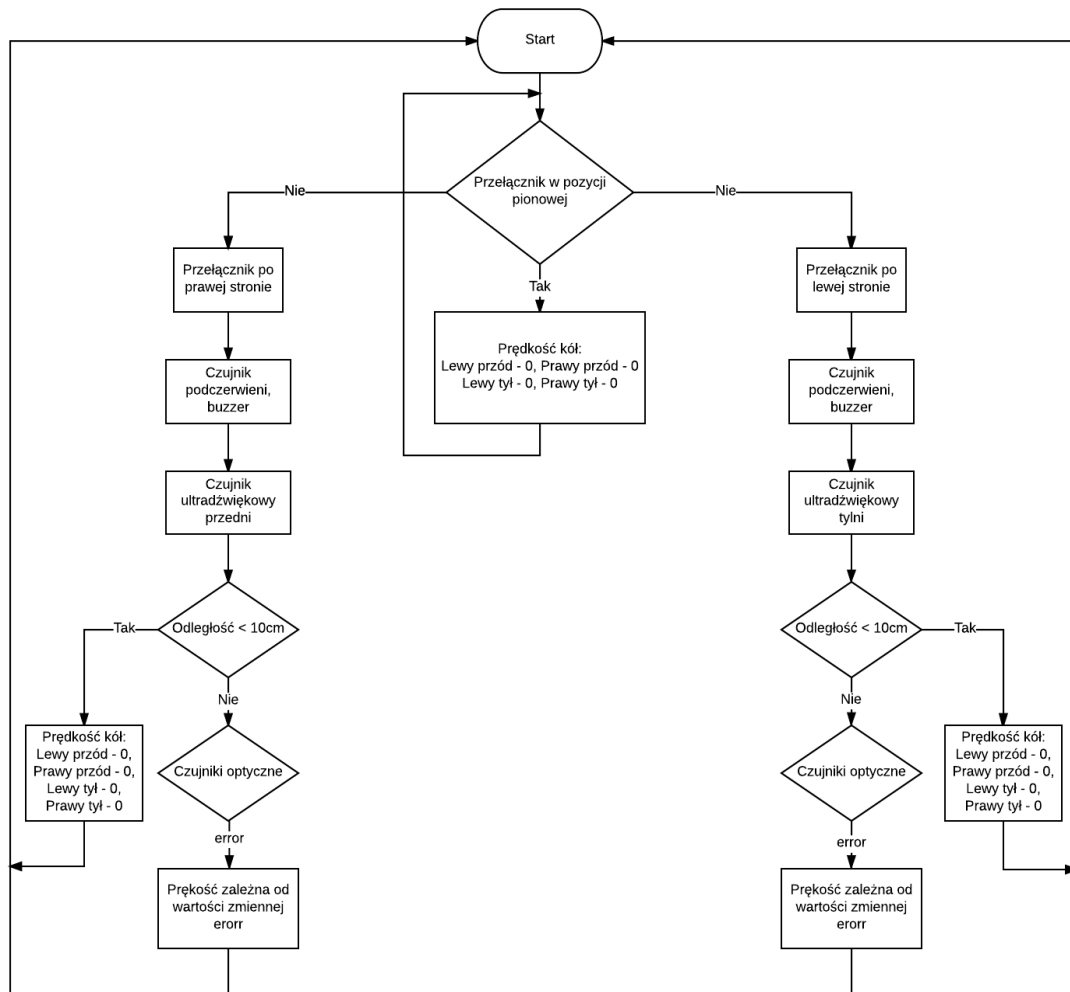
Nazwa zmiennej	Numer pinu	Podsumowanie
<i>Lfmotor</i>	5 (PWM)	Ruch lewego koła do przodu
<i>Lbmotor</i>	6 (PWM)	Ruch lewego koła do tyłu
<i>Rfmotor</i>	9 (PWM)	Ruch prawego koła do przodu
<i>Rbmotor</i>	10 (PWM)	Ruch prawego koła do tyłu
<i>Rsensor</i>	A0	Prawy czujnik optyczny
<i>Csensor</i>	A1	Centralny czujnik optyczny
<i>Lsensor</i>	A2	Lewy czujnik optyczny
<i>echo_front</i>	3 (PWM)	Wyjście przedniego czujnika ultradźwiękowego
<i>trig_front</i>	2	Wejście przedniego czujnika ultradźwiękowego
<i>echo_back</i>	11 (PWM)	Wyjście przedniego czujnika ultradźwiękowego
<i>trig_back</i>	12	Wejście przedniego czujnika ultradźwiękowego
<i>buzzer</i>	7	Wyjście z głośnika
<i>analogSensor</i>	A6	Wejście analogowego czujnika podczerwieni
<i>go_forward</i>	4	Stan przełącznika nakazujący ruch do przodu
<i>go_backward</i>	13	Stan przełącznika nakazujący ruch do tyłu

**Tabela 4.2.** Wykaz nazw zmiennych.

Nazwa zmiennej	Podsumowanie
<i>error</i>	Błąd, zależny od pozycji pojazdu względem osi jazdy
<i>time1</i>	Czas, zależny jak długo <i>echo_front</i> jest w stanie wysokim
<i>time2</i>	Czas, zależny jak długo <i>echo_back</i> jest w stanie wysokim
<i>distance1</i>	Odległość pomiędzy przodem pojazdu, a danym przedmiotem
<i>distance2</i>	Odległość pomiędzy tyłem pojazdu, a danym przedmiotem
<i>Lfspeed</i>	Prędkość z jaką lewe koło ma się poruszać do przodu
<i>Lbspeed</i>	Prędkość z jaką lewe koło ma się poruszać do tyłu
<i>Rfspeed</i>	Prędkość z jaką prawe koło ma się poruszać do przodu
<i>Rbspeed</i>	Prędkość z jaką prawe koło ma się poruszać do tyłu
<i>Sensor0</i>	Czujnik odbiciowy lewy, przechowuje wartość z <i>Lsensor</i>
<i>Sensor1</i>	Czujnik odbiciowy środkowy, przechowuje wartość z <i>Csensor</i>
<i>Sensor2</i>	Czujnik odbiciowy prawy, przechowuje wartość z <i>Rsensor</i>
<i>loud</i>	Zmienna przechowująca wartość z <i>analogSensor</i>

#### 4.2. Schemat blokowy algorytmu sterowania

Poniżej na rysunku 4.2 został zaprezentowany schemat blokowy, w jaki sposób program jest realizowany krok po kroku. Następne rozdziały są poświęcone opisowi kodu. W tym rozdziale wytłumaczono jak działa program i jaka jest jego struktura. Na poniższym schemacie zostało pokazane, że w zależności, w którą stronę przełącznik zostanie przesunięty to program realizuje inną funkcję.



Rys. 4.2. Działanie kodu programu zaprezentowane w formie schematu blokowego

#### 4.3. Inicjalizacja

Na początku programu została przedstawiona deklaracja i inicjalizacja zmiennych, w celu łatwiejszego pisania kodu. Jeżeli na pin numer 5 zostanie doprowadzony sygnał wysoki to lewy silnik zacznie się kręcić w jedną stronę powodując ruch pojazdu do przodu. Zmienna *Lbmotor* [Tabela 2.1] odpowiada za ruch w przeciwną stronę, czyli w tył. Analogicznie jest z kołem prawym.

*Rsensor* [Tabela 2.1] to zmienna reprezentująca prawy czujnik optyczny, który umieszczony jest na nadwoziu samochodu. *Csensor* to centralny, środkowy sensor, a *Lsensor* to czujnik optyczny po lewej stronie. Zostały one przypisane do wyjść/wejść analogowych, aby uzyskać oszczędność miejsca dla urządzeń wymagających podłączenia do pinów cyfrowych. Jednak innym, ważniejszym argumentem jest to, że sterowanie nie jest zero jedynkowe, ale analogowe. Otrzymuje się, więc większą swobodę i możliwości w regulacji ruchu nadążnego za linią czarną.

*Echo* i *Trig* są to zmienne wykorzystywane przy obsłudze czujników ultradźwiękowych. Są one opisane szczegółowo w podrozdziale 3.5. Pin *Echo*, który reprezentuje zmienną *Echo*, został doprowadzony do wyjść PWM. Do robota zostały podłączone dwa czujniki HC-SR04.

Jeden z nich z przodu pojazdu, a drugi po przeciwnej stronie. Przedrostki „front” dotyczą się czujnika umieszczonego z przodu, a „back” z tyłu.

Wyjście głośnika (buzzera) jest podłączone do pinu 7 i odpowiada zmiennej `46error`. `AnalogSensor` reprezentuje analogowy czujnik podczerwieni opisany szerzej w podrozdziale numer 3.6. Obie zmienne będą wykorzystywane w jednym obiekcie programowania.

Przełącznik należy podłączyć do dwóch wejść cyfrowych sterownika Arduino, oraz do zasilania. Do obsługi zastosowano dwie zmienne – `go_forward`, `go_backward`. Po przełączeniu mechanicznym, w środku przycisku zwierają się przewody, dzięki czemu płynie prąd w odpowiednią stronę. Gdy przełączymy do przodu, to na pinie numer 4 otrzymuje się wartość wysoką – pojazd będzie poruszał się do przodu. Jeżeli do tyłu, to na pinie numer 13 zostanie wygenerowana wartość wysoka, a pojazd zacznie się cofać. Jeżeli przełącznik będzie w pozycji pionowej, to samochód będzie stał w miejscu. Kod przypisania zmiennych pokazano na rysunku 4.3.

```
//przypisanie pinow
int LFMotor = 5;
int LBMotor = 6;
int RFMotor = 9;
int RBMotor = 10;
int Rsensor = A0;
int Csensor = A1;
int Lsensor = A2;
int echo_front = 3;
int trig_front = 2;
int echo_back = 11;
int trig_back = 12;
int buzzer = 7;
int analogSensor = A6;
int go_forward = 4;
int go_backward = 13;
//inicjalizacja zmiennych globalnych
int error;
float time1,time2;
float distance1, distance2;
int LFspped;
int RFspped;
int LBSpped;
int RBSpped;
```

Rys. 4.3. Początek programu związany z inicjalizacją zmiennych

Niektóre zmienne musiały zostać zainicjalizowane globalnie, ponieważ wymagane jest, aby były widoczne nie tylko w jednym obiekcie, ale w więcej niż jednej funkcji, lub nawet w całym programie. Zmienna `46error` odpowiada błędowi. Błąd oznacza, że 3 czujniki optyczne



nie czytują czarnej linii w tym samym momencie. Zmienne *time1*, *time2*, *distance1*, *distance2* są wykorzystywane w funkcji obsługującej czujnik ultradźwiękowy. Jest również wyróżnionych 5 zmiennych związanych z prędkością silników. Każda z nich zostało podłączone do wyjść PWM.

#### 4.4. Wnętrze funkcji *setup*

Każdy program napisany w aplikacji Arduino posiada funkcję *setup*. Funkcja ta jest wywoływana, tylko raz przy każdym uruchomieniu programu. W tym miejscu programu określa się co w programie jest wejściem, a co wyjściem. Komenda „OUTPUT” określa wyjście z programu, a „INPUT” wejście. W tej części programu zmiennym nadaje się wartości początkowe. Jeśli operujemy na wartościach analogowych to należy użyć komendy „analogWrite”, jeśli na cyfrowych to „digitalWrite”. Jeżeli są to obiekty podłączone do wyjść PWM to przypisujemy im wartości od 0 do 255. Jeśli są one połączone z wyjściem cyfrowym to używamy komendy „LOW” dla sygnału niskiego 0V, lub „HIGH” dla sygnału wysokiego 5V. Ostatnia linijka kodu funkcji to połączenie z portem szeregowym między komputerem a płytką Arduino. Dzięki temu można odczytywać na ekranie komputera, w aplikacji Arduino, wartości zmiennej, która nas interesuje. Napisany kod w funkcji *setup* przedstawiono na rysunku 4.4.

```
void setup() {  
    pinMode(LFmotor, OUTPUT);  
    pinMode(LBmotor, OUTPUT);  
    pinMode(RFmotor, OUTPUT);  
    pinMode(RBmotor, OUTPUT);  
    pinMode(Lsensor, INPUT);  
    pinMode(Csensor, INPUT);  
    pinMode(Rsensor, INPUT);  
    analogWrite(LFmotor, 0);  
    analogWrite(LBmotor, 0);  
    analogWrite(RFmotor, 0);  
    analogWrite(RBmotor, 0);  
    pinMode(trig_front, OUTPUT);  
    pinMode(echo_front, INPUT);  
    pinMode(trig_back, OUTPUT);  
    pinMode(echo_back, INPUT);  
    pinMode(buzzer, OUTPUT);  
    pinMode(analogSensor, INPUT);  
    digitalWrite(buzzer, LOW);  
    pinMode(go_forward, INPUT);  
    pinMode(go_backward, INPUT);  
    Serial.begin(9600);  
}
```

Rys. 4.4. Funkcja *setup*

#### 4.5. Kod do obsługi czujników odbiciowych

Głównym zadaniem pojazdu jest śledzenie czarnej linii, która wyznacza mu tor jazdy. Na dole pojazdu umocowane są 3 czujniki optyczne QRD1114, są one opisane szczegółowo w podrozdziale 3.4. Informacja o linii, otrzymana jest z czujników odbiciowych (Sensor0, Sensor1, Sensor2) w postaci analogowej. Aby prześledzić działanie programu, sczytujemy wartość czujnika środkowego czyli *Csensor*, przypisanego do zmiennej *Sensor1*. Jeżeli środkowy czujnik optyczny będzie widział czarną linię, to oznacza, że robot porusza się prawidłowo. Kod funkcji nazwanej *read\_sensor*, w której przypisywane są wartości do zmiennych przedstawiono na rysunku 4.5.

```
int read_sensor(){
    int Sensor0;
    int Sensor1; //optyczny
    int Sensor2;
    Sensor0 = analogRead(Lsensor);
    Sensor1 = analogRead(Csensor);
    Sensor2 = analogRead(Rsensor);
    Serial.println(Sensor1);
}
```

Rys. 4.5. Początek funkcji, odpowiedzialnej za sczytywanie czarnej linii

Następnym krokiem jest określenie sytuacji, w których pojazd będzie się znajdował w czasie swojej pracy oraz przypisanie tym sytuacjom odpowiedniej wartości. Zostało wydzielonych 6 sytuacji – czujniki nie wykrywają czarnej linii (jeden przypadek), tylko jeden, skrajny wykrywa czarną linię (dwa przypadki), dwa czujniki są nad czarną linią (dwa przypadki) oraz wszystkie 3 czujniki są nad osią jazdy (jeden przypadek). Odpowiednio do podanych sytuacji, zmienna *48error* przyjmuje różne wartości. Liczby ujemne reprezentują sytuację, gdy pojazd będzie odchylony w lewo od toru jazdy. Im większa liczba przypisana do *48error*, tym większy błąd, czyli większe odchylenie od osi jazdy.

Wartością graniczną zostało 500 bitów (max to 1023). Wartość ta została dobrana eksperymentalnie. Jeżeli napięcie na wejściu analogowym podłączonym do któregoś z czujników odbiciowych wzrośnie, lub zmaleje do 2,44[V] (wartość graniczna napięcia wyliczona na podstawie wzoru (4.1)), to program przeskoczy do następnej funkcji if, else if.

$$X = 500 [b] * 5[V] \div 1023[b] \approx 2,44[V] \quad (4.1)$$

gdzie:

*X* – napięcie na czujniku optycznym odpowiadające 500 bitom na mikroprocesorze Arduino

Jeżeli wartość przypisana do sensora będzie większa od 500 bitów, to oznacza to, że samochód jest nad czarną linią (logiczna wartość 1). Jeżeli jest mniejsza od 500 bitów to mamy sytuację, że dany czujnik nie jest na czarnej linii. Została wybrana wartość 500 bitów, aby były łagodne przejścia i istniał bufor pomiędzy zmianą położenia. Jeżeli wartość graniczna zostanie zmniejszona do zbyt małej wartości, to robot zbyt szybko reaguje i nie jedzie płynnie. Jeżeli

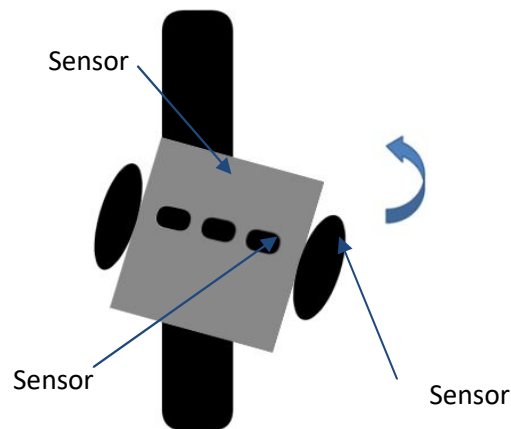
wartość będzie zbyt mocno zwiększona, to robot nie śledzi linii tak jak powinien i wypada z toru jazdy.

Jeżeli żaden czujnik nie wykrywa czarnej linii, to do zmiennej *49error* będzie przypisana liczba 10. Oznacza to, że uchyb jest duży, czyli pojazd wypadł z osi jazdy. Nie został w tym programie uwzględniony algorytm zawracania, ponieważ została przyjęta sytuacja, że pojazd nie zgubi toru jazdy. Jeżeli jednak tak się stanie to odpowiedni algorytm, który czytuje wartości z czujników ultradźwiękowych, zatrzyma pojazd w momencie, gdy natrafi on na przeszkodę. Wartości przypisywane do *49error* są umowne. Sposób ich wyznaczania przedstawiono na rysunku 4.6.

```
if( (Sensor0>=500) && (Sensor1>=500) && (Sensor2>=500) ) {  
    return error = 0;  
}  
else if( (Sensor0<500) && (Sensor1>500) && (Sensor2>=500) ) {  
    return error = 1;  
}  
  
else if( (Sensor0>=500) && (Sensor1>500) && (Sensor2<500) ) {  
    return error = -1;  
}  
else if( (Sensor0<500) && (Sensor1<500) && (Sensor2>=500) ) {  
    return error = 2;  
}  
else if( (Sensor0>=500) && (Sensor1<500) && (Sensor2<500) ) {  
    return error = -2;  
}  
else if( (Sensor0<500) && (Sensor1<500) && (Sensor2<500) ) {  
    return error = 10;  
}  
}
```

*Rys.4.6. Dalsza część funkcji do sczytywania osi jazdy. Instrukcje warunkowe*

Czujniki rozstawione są na szerokości 3 centymetrów. Aby program działał idealnie, to linia czarna powinna mieć szerokość minimum 3cm. Na rysunku 4.7 została przedstawiona sytuacja, w której pojazd zgubił oś jazdy. Dzięki zaimplementowanemu programowi, robot koryguje swoją pozycję. Czujnik *Sensor0* i *Sensor1* jest nad linią czarną, więc samochód wykonuje ruch w lewą stronę. Do zmiennej *49error* została przypisana wartość „-1”.



Rys. 4.7. Zobrazowanie ruchu pojazdu, w momencie gdy zgubi linię

#### 4.6. Kod do obsługi czujników ultradźwiękowych

Zasada działania czujników ultradźwiękowych HC-SR04 oraz przykładowy kod został opisany w podrozdziale 3.7. Poniższy kod zaprezentowany na rysunku 4.8 jest bardzo podobny do kodu przykładowego. Do budowy robota zostały wykorzystane dwa czujniki ultradźwiękowe. Jeden z nich umieszczony jest z przodu pojazdu, a drugi z tyłu. Inspiracją była budowa nowoczesnych samochodów, w których są dwa czujniki odległości wykrywające jak daleko przeszkoda znajduje się od pojazdu.

```
float ultra_front() {
    digitalWrite(trig_front, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig_front, LOW);
    time1 = pulseIn(echo_front, HIGH);
    distance1 = time1 * 0.017;
    return distance1;
}

float ultra_back() {
    digitalWrite(trig_back, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig_back, LOW);
    time2 = pulseIn(echo_back, HIGH);
    distance2 = time2 * 0.017;
    return distance2;
}
```

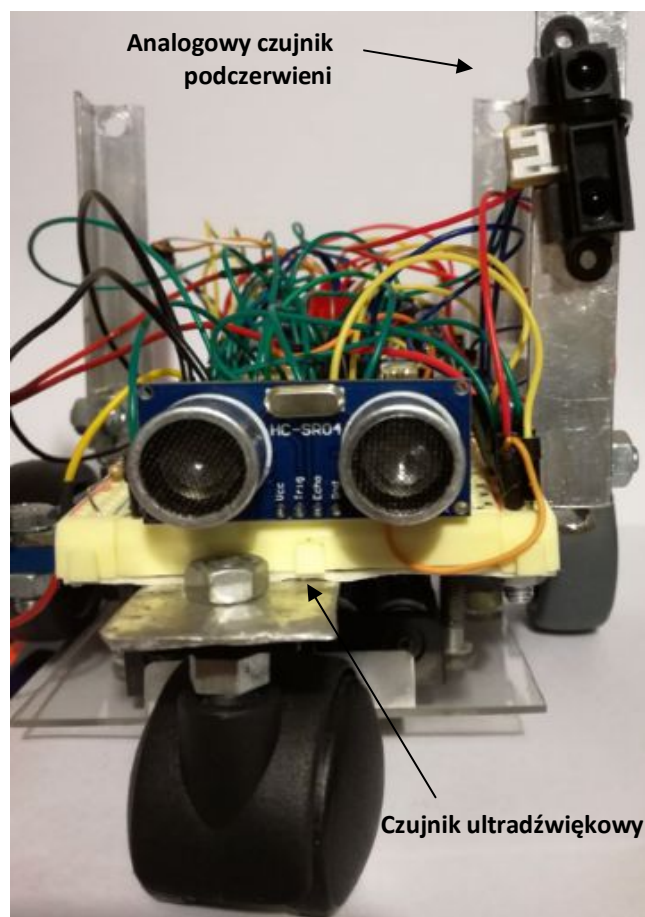
Rys. 4.8. Funkcja, która steruje dwoma czujnikami ultradźwiękowymi

#### 4.7. **Kod do obsługi czujnika podczerwieni**

Analogowy czujnik podczerwieni opisany został w podrozdziale 3.6. Czujnik ustawiony jest na słupku po prawej stronie. Jeżeli jakiś obiekt, np. wymijający samochód, będzie zbyt blisko naszego pojazdu to głośnik pokazany na rysunku 4.9 wyda donośny dźwięk.



Rys. 4.9. Zdjęcie głośnika używanego w projekcie



Rys. 4.10. Zdjęcie przodu robota bez panelu fotowoltaicznego

Wartość z czujnika jest odczytywana analogowo. Za sprawą kodu przedstawionego na rysunku 4.11 zostaje ona potem przypisana do zmiennej *loud*. Jeżeli zmienna *loud* będzie miała wartość powyżej 300 bitów to głośnik zacznie wydawać donośny dźwięk. Na rysunku 4.10 zaprezentowano odpowiednio podpisane czujniki.

```

int check(){
    int loud;
    loud = analogRead(analogSensor);
    if(loud<300){
        digitalWrite(buzzer, LOW);
    }
    else{
        digitalWrite(buzzer, HIGH);
    }
}

```

Rys. 4.11. Funkcja obsługująca analogowy czujnik podczerwieni oraz 52error52

#### 4.8. Kod do ustawiania prędkości

Aby ustawić odpowiednią prędkość, z jaką robot powinien się poruszać, należy odebrać informację z funkcji *read\_sensor*, opisanej w podrozdziale 4.5. Funkcja ta zwraca wartość błędu, związanego z położeniem w jakim pojazd się aktualnie znajduje. Zależnie od zmiennej *52error* należy podać na koła odpowiednie wartości. Wartości te są podawane na wyjścia PWM, także zakres to od 0 do 255 jednostek w programie Arduino.

Pierwsza funkcja to *motors\_speed*, której kod zaprezentowano na rysunku 4.12. W tej funkcji podaje się wartości PWM na poszczególne mostki H sterujące silnikami kół. Została ona stworzona po to, aby kod był krótszy oraz bardziej czytelny.

```

void motors_speed(int LFspped,int RFspped,int LBspped,int RBspped){
    analogWrite(LFmotor,LFspped);
    analogWrite(RFmotor,RFspped);
    analogWrite(LBmotor,LBspped);
    analogWrite(RBmotor,RBspped);
}

```

Rys. 4.12. Funkcja do podawania prędkości na odpowiednie koła

Następna funkcja, również związana z ustawieniem prędkości pojazdu, to funkcja, w której zastosowano instrukcje wielokrotnego wyboru „switch case”. Funkcja *switch* na wejściu otrzymuje wartość, która została przypisana do zmiennej *52error* z funkcji *read\_sensor*.

W zależności od danej wartości, funkcja *switch* przełączy się do odpowiedniego przypadku. W konsekwencji na koła zostanie przypisana inna wartość prędkości. Wyborów jest 6, tyle samo ile funkcja *read\_sensor* może zwrócić wartości *52error*. Samochód może jechać do przodu lub do tyłu, zależne od pozycji przełącznika. Dlatego istnieją dwie funkcję *forward*, *backward*. Każda z nich działa bardzo podobnie, jedynie z taką różnicą, że jedna funkcja podaje napięcie PWM na piny sterownika odpowiedzialne za obracanie się kół w jednym kierunku, a druga w przeciwnym.



Prędkość jest zależna od tego, jak bardzo pojazd jest wychylony od czarnej linii, za którą powinien prosto podążać. Jeżeli wszystkie czujniki sczytują linię, to samochód będzie jechał z maksymalną prędkością (255 wartości PWM). Jeżeli robot zgubi linię to pojazd ma zwolnić (245 wartości PWM). Dzięki temu zabiegowi, będzie łatwiej pojazdowi odnaleźć na nowo linię, a nie wyjechać za daleko poza obszar pracy.

W przypadku gdy jeden czujnik optyczny przestanie widzieć czarny kolor to samochód będzie się poruszał z prędkością proporcjonalną do wartości PWM równej 245.

Jeżeli pojazd będzie bardzo wychylony, to znaczy tylko jeden, skrajny czujnik będzie generował napięcie, czyli będzie na obszarze linii czarnej to wartość PWM na mikroprocesorze Arduino będzie wynosić 250. Wartość PWM w przypadku, gdy tylko jeden czujnik widzi czarną linię jest większa od wartości PWM, gdy pojazd jest dwoma czujnikami na linii, ponieważ samochód musi wykonać bardziej ostry skręt. Do tego manewru wymagana jest większa prędkość. Funkcje *forward* i *backward* opisywane wyżej zaprezentowano na rysunku 4.13.

```
void forward (int error){
    switch (error){
        case 10:
            motors_speed(240, 240, 0, 0);
            break;
        case 0:
            motors_speed(255, 255, 0, 0);
            break;
        case 1:
            motors_speed(250, 0, 0, 0);
            break;
        case -1:
            motors_speed(0, 250, 0, 0);
            break;
        case 2:
            motors_speed(255, 0, 0, 0);
            break;
        case -2:
            motors_speed(0, 255, 0, 0);
            break;
    }
}

void backward (int error){
    switch (error){
        case 10:
            motors_speed(0, 0, 240, 240);
            break;
        case 0:
            motors_speed(0, 0, 255, 255);
            break;
        case 1:
            motors_speed(0, 0, 245, 0);
            break;
        case -1:
            motors_speed(0, 0, 0, 245);
            break;
        case 2:
            motors_speed(0, 0, 250, 0);
            break;
        case -2:
            motors_speed(0, 0, 0, 250);
            break;
    }
}
```

Rys. 4.13. Funkcje odpowiedzialna za ruch pojazdu do przodu (po lewej) i do tyłu (po prawej)

#### 4.9. Wnętrze funkcji loop

Funkcja *loop*, obok funkcji *setup*, jest podstawowym członem każdego programu tworzonego w środowisku Arduino. W przeciwieństwie do *setup*, *loop* wykonuje się cyklicznie, bez przerwy. Wszystkie obiekty, opisane w powyższych podrozdziałach, są wykonywane w funkcji *loop*. Dzięki temu kod staje się bardziej czytelny.

Na samym początku funkcji, jest sprawdzany stan przełącznika. Jeżeli ustawiony jest w pozycji wertykalnej, to samochód stoi bez ruchu. W zależności od tego, w którą stronę użytkownik przestawi przełącznik, to samochód pojedzie do przodu (stan wysoki w zmiennej *go\_forward*), lub do tyłu (stan wysoki na zmiennej *go\_backward*).

W dalszej części kodu następuje sprawdzenie, czy jakiś obiekt nie jest za blisko lewej strony pojazdu (kod na obsługę analogowego czujnika podczerwieni, rozdział 4.7) i ewentualne wydanie dźwięku przez głośnik.

Następnie zostaje sprawdzone, czy dany przedmiot nie stoi naprzeciwko, lub z tyłu pojazdu. W zależności jaki jest stan przełącznika pojazd jedzie do przodu, lub do tyłu. Jeżeli jedzie do przodu to tylko przedni czujnik jest aktywny, a jeśli do tyłu to tylko tylny czujnik ultradźwiękowy jest w gotowości do działania. Jeśli dystans pomiędzy robotem, a obiektem jest mniejszy od 10, to samochód się zatrzyma. Liczba 10 odpowiada 10 centymetrom. Zastosowanie czujnika ultradźwiękowego jest opisane szerzej w podrozdziale 4.7 oraz w podrozdziale 3.5.

Jeżeli wszystkie te etapy bezpieczeństwa przeszły pomyślnie, to program przechodzi do miejsca, w którym zaczyna zaczytywać dane z czujników optycznych. Funkcja *read\_sensor* wystawia zmienną *54error*, która trafia do funkcji *forward*, lub *backward*, w zależności jaki kierunek ruchu samochodu został wybrany. Wnętrze funkcji *loop* zaprezentowano na rysunku 4.14.

```
void loop() {
  if(digitalRead(go_forward)==0 && digitalRead(go_backward)==0){
    motors_speed(0, 0, 0, 0);
  }
  else if(digitalRead(go_forward)==1){
    check();
    ultra_front();
    if (distance1 < 10){
      motors_speed(0, 0, 0, 0);
    }
    else{
      read_sensor();
      forward(error);
    }
  }
  else if(digitalRead(go_backward)==1){
    check();
    ultra_back();
    if (distance2 < 10){
      motors_speed(0, 0, 0, 0);
    }
    else{
      read_sensor();
      backward(error);
    }
  }
}
```

Rys. 4.14. Funkcja *loop*



#### 4.10. Funkcja *setup* i inicjalizacja zmiennych programu drugiego

Na początku programu zostały zainicjalizowane zmienne globalne, które są używane w więcej niż jednej funkcji. Zastosowanie to umożliwiło oszczędzenie miejsca w kodzie i sprawiło, że jest czytelniejszy. Wszystkie cztery zmienne są przywoływane w dwóch funkcjach: *front()* i *back()*, które zostaną opisane w następnych podrozdziałach. Funkcja *setup* tak samo jak w poprzednio opisanej wersji programu odpowiada za określenie czy dany pin jest wyjściem (OUTPUT), czy wejściem (INPUT) za pomocą funkcji *pinMode*. Na rysunku 4.15 przedstawiono kod programu, wraz z komentarzami opisującymi konkretne zmienne, których zastosowanie zostanie wytłumaczone w kolejnych podrozdziałach.

Oprócz przypisywania pinom roli wejścia lub wyjścia, funkcja *setup* została wykorzystana do określenia wartości początkowej konkretnych wyjść za pomocą funkcji *analogWrite* dla wejść analogowych i *digitalWrite* dla wejść cyfrowych. Jak można zobaczyć na rysunku 4.15. pinom 5, 6, 9 i 10, odpowiedzialnym za obracanie się kół przypisano 0. Wartość LOW, czyli niskie napięcie przypisano do pinu 7, uruchamiającego brzęczyk oraz pinu 8, odpowiedzialnego za uruchamianie przetwornicy DC/DC opisanej podrozdziale 3.8.

```
int flag;
int rightsensor = A0;
int centralsensor = A1;
int leftsensor = A2;

void setup()
{
    pinMode(2, OUTPUT); // TRIG front uf
    pinMode(3, INPUT);  // ECHO front uf
    pinMode(12, OUTPUT); // TRIG back uf
    pinMode(11, INPUT);  // ECHO back uf
    pinMode(5, OUTPUT);  // Forward Direction Left Motor
    pinMode(6, OUTPUT);  // Backward Direction Left Motor
    pinMode(9, OUTPUT);  // Forward Direction Right Motor
    pinMode(10, OUTPUT); // Backward Direction Right Motor
    pinMode(A2, INPUT);  // Left sensor
    pinMode(A1, INPUT);  // Central sensor
    pinMode(A0, INPUT);  // Right sensor
    pinMode(A4, INPUT);  // Volts
    pinMode(A6, INPUT);  // Analog sensor
    pinMode(4, INPUT);   // Front
    pinMode(13, INPUT);  // Back
    pinMode(7, OUTPUT);  // Voice
    pinMode(8, OUTPUT);  // Charging ON/OFF
    analogWrite(9, 0);
    analogWrite(5, 0);
    analogWrite(10, 0);
    analogWrite(6, 0);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    Serial.begin(9600);
}
```

Rys. 4.15. Kod inicjalizacji zmiennych i funkcji *setup*

#### 4.11. Wydzielone funkcje

Podstawowymi funkcjami każdego programu napisanego w środowisku Arduino IDE, jak można zauważyć we wstępie na rysunku 4.1, są *setup* i *loop*. Oprócz nich można tworzyć oddzielne funkcje, które potem można wywoływać w programie głównym, którym jest funkcja *loop*. Takie rozwiązanie umożliwia łatwiejsze programowanie i sprawię, że kod staje się czytelniejszy.

W programie zapewniającym autonomiczną pracę pojazdu, oprócz podstawowych funkcji, wydzielono jeszcze 5 dodatkowych:

- *direct*, odpowiedzialna za wyznaczanie kierunku obrotu kół,
- *us*, obsługująca czujnik ultradźwiękowy,
- *front*, odpowiadająca za jazdę do przodu,
- *back*, odpowiadająca za jazdę do tyłu,
- *stay*, postój samochodu.

Zastosowanie funkcji *direct* przedstawionej na rysunku 4.16 ułatwiło zapis funkcji *front*, *back* i *stay*. Zbudowana z czterech funkcji *analogWrite*, za pomocą których przypisywane są wartości napięcia z zakresu od 0 do 5V (manipulowanego za pomocą PWM, który w programie zakres napięcia mieści się w zakresie od 0 do 255) na konkretne piny. Zmienna *FR* przypisana numerowi 5, jest skrótem od „front left” (kierunek obrotu lewego koła „do przodu”), po zadaniu napięcia na dany pin, prawe koło zacznie obracać się w stronę, która zapewnia jazdę do przodu. Podobnie dla pozostałych, gdzie *FR* oznacza „front right” (kierunek obrotu prawego koła „do przód”), *BL* „back left” (kierunek obrotu lewego koła „do tyłu”), *BR* „back right” (kierunek obrotu prawego koła „do tyłu”). Dzięki takiemu zastosowaniu można zadać kierunek i prędkość poruszania się pojazdu, używając tylko jednej funkcji.

Funkcja *us*, zaprezentowana na rysunku powyżej, zwraca w programie przekształconą wartość odległości pojazdu od przeszkody, dokładny sposób przekształcania impulsów otrzymywanych z pinów przypisanych do zmiennej *echo* opisano w podrozdziale 3.5. W skrócie, naprzemiennie zadawane jest napięcie niskie i wysokie na odpowiednim pinie czujnika ultradźwiękowego za pomocą przypisywania konkretnej wartości do zmiennej *trig*. Aby określić długość opóźnienia pomiędzy impulsami skorzystano z funkcji *delayMicroseconds*. Dzięki czemu wysyłane są fale ultradźwiękowe w określonych odstępach czasu, które odbijają się od przeszkody i wracają na czujnik, gdzie są odbierane i konwertowane na impulsy napięcia. Przesyłane impulsy na odpowiednie piny mikrokontrolera opisano w programie za pomocą zmiennej *echo*. Do określenia odległości służy funkcja *PulseIn*, konwertująca czas otrzymania impulsu na wartość liczbową, której wykorzystanie opisano w następnym podrozdziale 4.12.

Funkcja *stay* jest wykorzystywana w celu zatrzymania pojazdu, wewnątrz jej znajduje się tylko jedna funkcja *direct* przypisująca wartości 0 do wszystkich zmiennych odpowiedzialnych, za obracanie się kół silnika.

```

void direct(int FL, int FR, int BL, int BR)
{
    analogWrite(9, FR);
    analogWrite(5, FL);
    analogWrite(10, BR);
    analogWrite(6, BL);
}

int us(int trig, int echo)
{
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    return pulseIn(echo, HIGH);
}

void stay()
{
    direct(0, 0, 0, 0);
}

```

Rys. 4.16. Funkcje użyte w programie, od góry: direct, us, stay

<pre> void front() {     if (analogRead(centralsensor) &gt;= 600){         direct(255, 255, 0, 0);         flag = 0;     }     else if(analogRead(leftsensor) &gt;= 600){         direct(0, 255, 255, 0);         flag = 1;     }     else if (analogRead(rightsensor) &gt;= 600){         direct(255, 0, 0, 255);         flag = 2;     }     else if(flag == 0){         direct(255, 255, 0, 0);     }     else if(flag == 1){         direct(0, 255, 255, 0);     }     else if(flag == 2){         direct(255, 0, 0, 255);     } } </pre>	<pre> void back() {     if (analogRead(centralsensor) &gt;= 600){         direct(0, 0, 255, 255);         flag = 0;     }     else if(analogRead(leftsensor) &gt;= 600){         direct(255, 0, 0, 255);         flag = 1;     }     else if (analogRead(rightsensor) &gt;= 600){         direct(0, 255, 255, 0);         flag = 2;     }     else if(flag == 0){         direct(0, 0, 255, 255);     }     else if(flag == 1){         direct(255, 0, 0, 255);     }     else if(flag == 2){         direct(0, 255, 255, 0);     } } </pre>
---	--

Rys 4.17. Funkcję front i back odpowiedzialne za wyznaczanie kierunku jazdy

Przedstawione na rysunku 4.17 kody programu umożliwiły skręcanie pojazdu w zależności od zwrotu jazdy. Obie części odpowiedzialne są prawidłowe funkcjonowanie detekcji czarnej linii i podążania za nią.

Pierwszym założeniem było stworzenie programu na trzech prostych warunkach „*if... else*”, każdy odpowiedzialny był za sprawdzanie wartości zmiennej przypisanej pinom czujnika. Jako, że w układzie zastosowano trzy czujniki odbiciowe umożliwiło to bardzo proste zaimplementowanie kodu programu do zmiany kierunku poruszania się pojazdu. Jeżeli czarna linia znajduje się pod czujnikiem, wysyła on sygnał napięciowy na podłączony odpowiedni analogowy pin modułu arduino. Zakres wartości napięcia wynosi od 0 do 5 V. W kodzie programu otrzymywany sygnał reprezentuje odpowiednia zmienna, w zależności od tego czy jest to czujnik z lewej strony (*leftsensor*), prawej strony (*rightsensor*), czy znajdujący się pomiędzy (*centralsensor*). Wartość reprezentująca i dopowiadająca sygnałowi napięciowemu zmienia się od 0 do 1023. Konkretny warunek z trzech przedstawionych na rysunku 4.17 jest spełniony, gdy wartości zmiennej reprezentującej sygnał napięciowy wysyłany z podłączonego czujnika przekroczy 600. W przypadku pojawienia się czarnej linii pod środkowym czujnikiem, w programie za pomocą funkcji *direct*, zostanie wysłany wysoki sygnał napięciowy na odpowiednie porty sterownika L293D, co spowoduje obrót kół w przód. Gdy czarna linia znajdzie się pod prawym czujnikiem, za pomocą przypisania w programie wartości 255 do odpowiednich zmiennych i wysłaniu sygnałów na sterownik koła prawe zaczną obracać się do tyłu a lewe do przodu. Odwrotnie dzieje się w przypadku pojawienia się czarnej linii pod lewym czujnikiem.

Wraz z testowaniem niezawodności działania programu, okazało się że przy ostrym skręceniu pojazd gubił tor jazdy. Spowodowane to było znikaniem czarnej linii z pola widzenia czujników odbiciowych. W celu rozwiązania problemu zastosowano nową zmienną globalną, *flag* (jedna z czterech wspomnianych we wstępie). W trakcie spełnienia warunków potrzebnych do zmiany kierunku do zmiennej jest przypisywana nowa wartość. Podczas jazdy prosto do zmiennej przypisywana jest wartość 0, podczas jazdy w lewo 1, a w prawo 2. Dzięki takiemu zastosowaniu, gdy czarna linia zniknie z pola widzenia czujników, w zmiennej *flag* jest zapisana odpowiednia wartość. Aby zapewnić poruszanie się pojazdu w opisanym przypadku i umożliwienie ponownego znalezienia linii należy odpowiednio zaprogramować mikrokontroler. Na rysunku 4.17 przedstawiono warunki spełniane w przypadku, gdy czujnik gubi linię, ale flaga uzyskała określoną wartość. Jeżeli zmienna *flag* jest równa 0 samochód kontynuuje jazdę do przodu, gdy 1 jazdę w lewo, gdy 2 w prawo.

#### **4.12. Funkcja loop**

Główny kod programu znajduje się w funkcji nazwanej z góry przez środowisko arduino IDE *loop*. Działanie odbywa się jak sama nazwa wskazuje w nieskończonej pętli, dzięki czemu napisany w niej kod będzie zapewniał ciągłą pracę pojazdu.

Na rysunku 4.18 zaprezentowano kod, w pierwszej jego części widać przypisanie zmiennych, które są wykorzystywane w danej funkcji. Zastosowanie to ułatwia zrozumienie kodu i sprawia, że jest czytelniejszy.

Pierwszy opisany w kodzie warunek odpowiada za sprawdzanie wartości przypisanej do zmiennej *IRsensor*, zmieniającej się od 0 do 600. Wartości te odpowiadają sygnałowi napięciowemu wysłanemu z odległościowego czujnika podczerwieni, opisanego w podrozdziale 3.6 na pin analogowy A6 mikrokontrolera. W przypadku przypisania wartości większej niż 200 do zmiennej *Voice* jest przypisywany stan wysoki, czego skutkiem uruchomienie jest brzęczyka ostrzegającego o zbliżającej się przeszkodzie.

Wartość sygnału napięciowego wysłanego na port A4 odpowiada napięciu jakie jest na układzie podzielonemu przez dwa. Jako że wartość 5.0 v otrzymana na pinie odpowiada wartości 1023 w programie zastosowano odpowiednie obliczenia (4.2), aby uzyskać wartość napięcia jakie jest na układzie. W przypadku przekroczenia wartości 5.1 wysyłany jest wysoki sygnał napięciowy na pin *on/off* przetwornicy DC/DC, opisanej w podrozdziale 3.8 w celu odłączenia panelu od układu i zaprzestania doładowywania akumulatorów.

```
void loop()
{
    int IRsensor = analogRead(A6);
    int gofront = digitalRead(4);
    int goback = digitalRead(13);
    int U = 0.009775*analogRead(A4);
    int Voice = 7;
    int Charge = 8;
    int UStrigfront = 2;
    int USechofront = 3;
    int UStrigback = 12;
    int USechoback = 11;

    if(IRsensor >= 200 && gofront == HIGH) digitalWrite(Voice, HIGH);
    else digitalWrite(Voice, LOW);
    if(U <= 5.1) digitalWrite(Charge, HIGH);

    if (goback == HIGH && us(UStrigback, USechoback) > 600) back();
    else if (gofront == HIGH && us(UStrigfront, USechofront) > 600) front();
    else stay();

    // Serial.println(analogRead(rightsensor));
    // Serial.println(analogRead(centralsensor));
    // Serial.println(analogRead(leftsensor));
    // Serial.println("napięcie");
    // Serial.println(U);
    // Serial.println(IRsensor);
    // delay(500);
}
```

Rys 4.18. Główny kod programu, czyli funkcja *loop*

$$U = \left( \frac{2}{1023} \right) * A6 \quad (4.2)$$

gdzie:

$U$  – napięcie układu,  $A6$  – wartość przetworzona przez mikrokontroler.

Po sprawdzeniu opisanych warunków, bez względu na to czy zostały one spełnione kolejnym zadaniem programu jest sprawdzenie, czy otrzymano wysoki sygnał napięciowy na wejścia cyfrowe 13 lub 4, odpowiednio przypisanych zmiennym *goback* i *gofront*. W przypadku, gdy na pinie 13 pojawi się wysoki sygnał napięciowy, a wartość zwrócona z funkcji *uf* będzie większa od 600 zostanie wykonywana funkcja *back* co spowoduje poruszanie się samochodu do tyłu. Dla wysokiego sygnału napięciowego na pinie 4 i wartości zwróconej funkcji *uf* większej od 600 samochód porusza się do przodu. Do funkcji *uf* można przypisać różne zmienne, w zależności od tego, czy sprawdzam napięcie na czujniku znajdującym się z przodu, czy z tyłu samochodu. W innym przypadku samochód się nie porusza.

Wykomentowane na rysunku za pomocą „//” linijki służyły obserwowania stanu pracy samochodu przy użyciu monitora portu szeregowego.

## 5. Badania, wnioski i podsumowanie projektu inżynierskiego

Po wykonaniu projektu zostały przeprowadzone badania wykazujące przydatność wykorzystania panelu fotowoltaicznego w pojeździe elektrycznym. Celem testów było udowodnienie, że dzięki zastosowaniu panelu słonecznego czas pracy akumulatorów się wydłuża, dzięki czemu samochód przejedzie dłuższą trasę. Ważnym aspektem badań było wykazanie, że akumulatory ładują się podczas jazdy.

Zostało sprawdzone ile energii słonecznej zostanie pobranej przez panel fotowoltaiczny, jeżeli zostanie on ustawiony pod kątem do promieni słonecznych, a ile jeżeli panel będzie ustawiony równolegle do podłoża. Aby testy były przeprowadzone właściwie, to wymagane jest wysokie promieniowanie słoneczne. Ogniwa fotowoltaiczne wykazują większą sprawność w niskich temperaturach. Odpowiednie warunki atmosferyczne to brak zachmurzenia, natężenie słoneczne powyżej  $500 \text{ W/m}^2$  oraz temperatura około  $20^\circ\text{C}$ .

Oprócz badanych zalet wykorzystania panelu do wydłużenia pracy działania pojazdu, przeprowadzono również testy pojazdu na zaprojektowanych i wydrukowanych na białym papierze formatu A1 trasach, w celu sprawdzenia skuteczności działania opracowanych programów sterujących pojazdem, a zaimplementowanych na mikrokontroler.

Duża waga panelu, w porównaniu do mocy silników, ogranicza sprawność ruchową samochodu. Mimo to autonomiczny pojazd wraz z panelem fotowoltaicznym działa bez zarzutów, a energia słoneczna, pobierana poprzez panel słoneczny dostarcza dodatkową energię elektryczną do akumulatorów.

Dodatkowymi badaniami, było sprawdzenie jak pojazd będzie pracował, gdy panel będzie zasilala lampa halogenowa. Pasma światła pochodzące z lamp halogenowych jest, w przybliżeniu, w tym samym paśmie co światło słoneczne. Badania miały na celu wykazanie, czy możliwe jest ładowanie akumulatora pojazdu zasilanego panelem fotowoltaicznym tylko przez lampy halogenowe. Podczas badań został wykorzystywany multimetr do sczytywania napięcia oraz pyranometr. Aby uzyskać wartość nasłonecznienia korzystano ze wzoru (5.1).



Rys 5.1. Pyranometr, używany podczas badań

$$P = \frac{U}{0,068} \quad (5.1)$$

Gdzie:

P - Natężenie moc promieniowania [ $\text{W/m}^2$ ], U – Napięcie na pyranometrze.



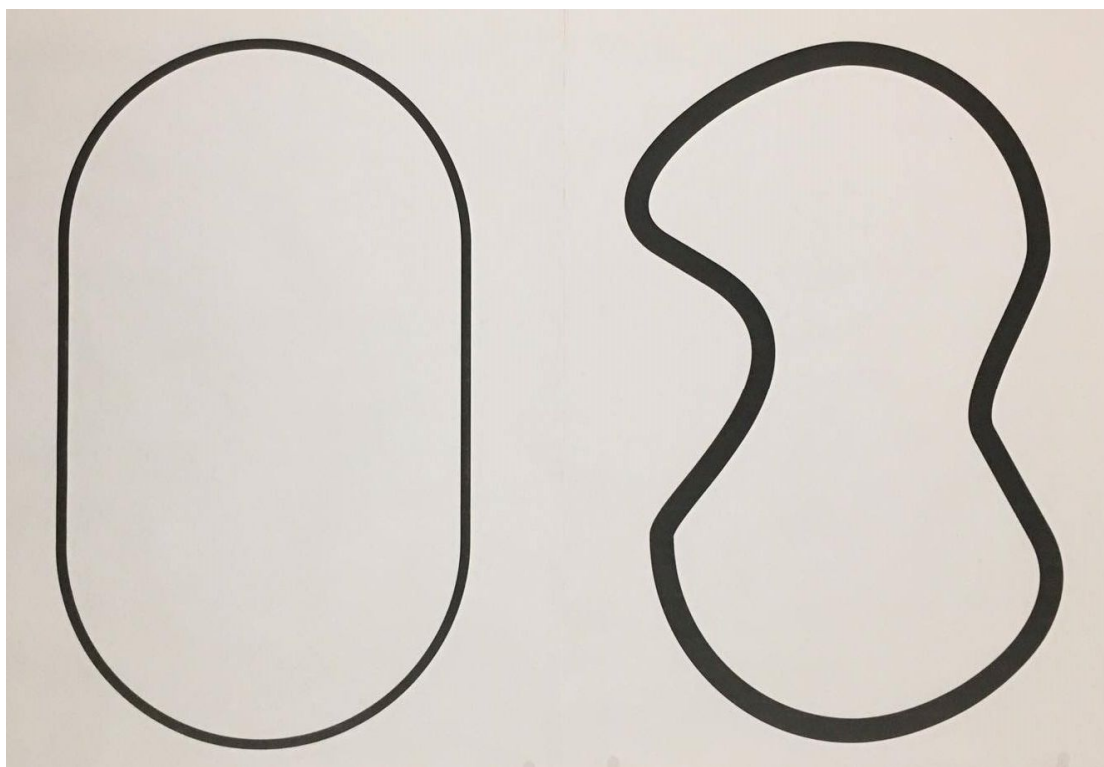
### 5.1. Praca pojazdu na zaprojektowanych trasach

W celu testowania działania układu regulacji pojazdu zwanego „line followerem” zostały wydrukowane dwie trasy, których wizerunek zaprezentowano na rysunku 5.2.

Grubość linii trasy pierwszej wynosi przybliżoną szerokość pola widzenia pojedynczego czujnika, czyli 1 cm. Zastosowany program, opisany w podrozdziałach od 4.10 do 4.12 idealnie spełniał swoje zadanie, a samochód poruszał się po trasie nie gubiąc toru jazdy, dzięki wykorzystanej do odnajdywania linii zmiennej „flag”. Przejechanie jednego okrążenia zajmuje małemu pojazdowi około 30 sekund. Film pod tytułem „pierwsza trasa” demonstrujący poruszanie się pojazdu został zamieszczony na dołączonej do pracy płytce CD.

Trasa druga (plik „druga trasa”) została zaprojektowana z myślą o badaniach, wykonanych z zaimplementowanym programem, opisanym w podrozdziałach od 4.1 do 4.9. Jak można zauważyć na rysunku linia trasy jest grubsza, ponieważ wynosi około 3 cm. Dzięki czemu możliwym jest znajdowanie się nad nią trzech czujników w tym samym czasie. Samochód po wydrukowanej trasie porusza się płynniej niż w poprzedni przypadku, ze względu na stopniowe zmniejszanie wartości obrotu koła po stronie, w którą skręca samochód.

Wykonując badania stwierdzono, że samochód porusza się sprawniej w kierunku przednim. Dzieje się tak, ponieważ czujniki odbiciowe sczytujące czarną linię znajdują się, w tym przypadku przed kołami napędzanymi przez silniki, a w przypadku poruszania się do tyłu za wymienionymi kołami. Skutkiem czego jest opóźniona reakcja na zakręcenie linii. „Bieg tylni” może być wykorzystywany do jazdy prosto.



Rys. 5.2. Trasy wykorzystane do badań sprawności działania „line followera” (po lewej trasa pierwsza, a po prawej druga)



## 5.2. Rozładowywanie się akumulatorów w czasie jazdy pojazdu

Zanim przystąpiono do badań wpływu panelu fotowoltaicznego na długość pracy akumulatorów mini pojazdu elektrycznego dokonano testów przy odłączonych przewodach łączących panel z ogniwami. Dzięki czemu, akumulatory nie doładowywały się w trakcie jazdy i można było sprawdzić w jakim czasie się rozładowują. Panel był przymocowany do pojazdu.

Do czterech wykonanych badań wykorzystano pierwszą trasę przedstawioną na rysunku 5.2. Łącznie samochód podążał za wyznaczoną linią przez około 3 h. Dnia 8 i 9 listopada bieżącego roku wykonano 10 pomiarów napięcia akumulatorów z przerwami pomiędzy którymi samochód się poruszał, bądź stał w miejscu z odłączonymi akumulatorami. Uzyskane pomiary zostały zaprezentowane i opisane w tabeli 5.1.

**Tabela 5.1.** Wyniki pomiarów napięcia akumulatorów podczas rozładowywania z dnia 8 i 9 listopada wykonane bez udziału panelu

Data wykonywanych pomiarów	Godzina	Otrzymane napięcie [V]	Stan samochodziku przed wykonanym pomiarem
08/11/2017	23:27	4,98	Spoczynek
08/11/2017	23:57	4,88	30 min jazdy
09/11/2017	24:05	4,9	8 min spoczynku
09/11/2017	24:35	4,85	30 min jazdy
09/11/2017	12:51	4,97	12 h 16 min spoczynku
09/11/2017	13:21	4,8	30 min jazdy
09/11/2017	14:06	4,86	45 min spoczynku
09/11/2017	14:36	4,74	30 min jazdy
09/11/2017	21:58	4,84	7 h 22 min spoczynku
09/11/2017	22:58	4,57	1 h jazdy

Na podstawie danych podanych przez producenta nominalne napięcie ogniwa niklowo-kadmowego wynosi 1.2 V, a zakres pracy mieści się w wartości od 1 do 1,4 V. Wynika z tego, że napięcie nominalne uzyskane z akumulatorów wynosi 4,8 V, minimalne 4,0 V (rozładowane baterie), a maksymalne 5,6 V (naładowane baterie).

Z wykonanych pomiarów wynika, że podczas około 3 h jazdy samochodu wartość jego napięcia spadła o  $4,98 - 4,57 = 0,41$  V. Jak można zauważyć pomiary mogą być niedokładne, ponieważ po dłuższym spoczynku samochodu, gdy akumulatory nie są rozładowywane napięcie mierzone multimetrem wzrasta. Dla najdłuższego spoczynku napięcie wzrosło o wartość  $4,97 - 4,85 = 0,12$  V.

Próbowano wykonać badania z podłączonym panelem słonecznym na zewnątrz przy temperaturze 5°C oraz nasłonecznieniu 400 W/m<sup>2</sup>, a otrzymane pomiary zaprezentowane w tabeli 5.2 wykazały, że napięcie spada szybciej, co mogło być spowodowane niską temperaturą. Wraz ze spadkiem temperatury sprawność i pojemność akumulatorów znacząco maleje.

Podczas wykonywania badań z halogenami samochód nie śledził poprawnie czarnej linii, ponieważ światło padające na czujniki odbiciowe i odbijające się od kartki papieru (na której była trasa) widocznie zakłócało ich pracę. Detektory czujnika odbierają wiązki

podczerwieni, które również wytwarzają halogeny. Rozwiązaniem jakie zastosowano, aby zbadać skuteczność doładowywania akumulatorów przez panel było wgranie na mikrokontroler programu, który sprawiał, że samochód jeździł w kółko. Dzięki czemu nie trzeba było ingerować oprogramowaniu czujników odbiciowych.

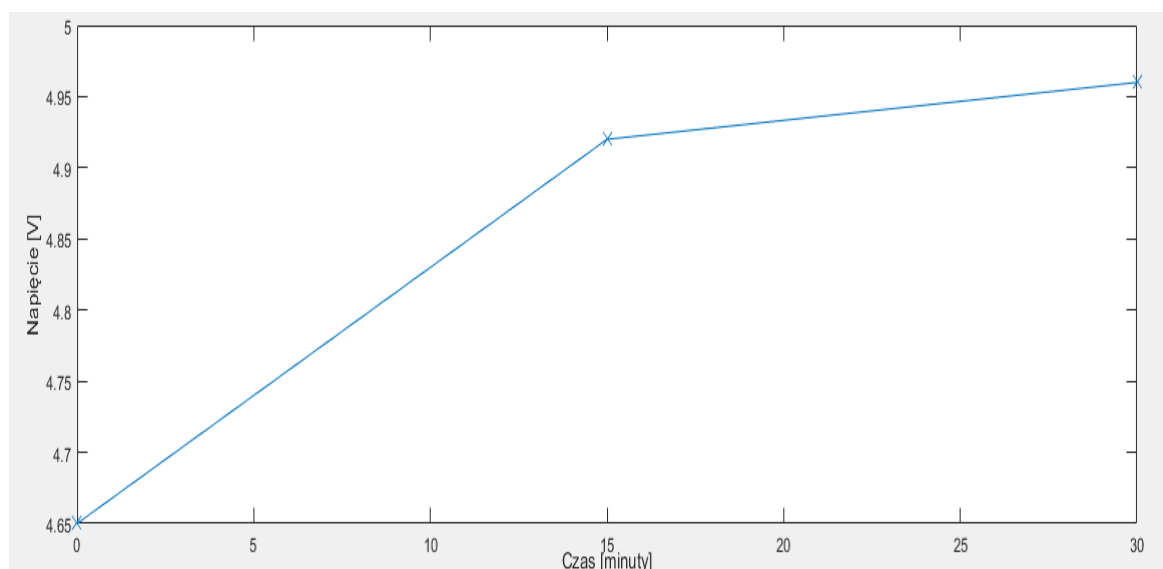
**Tabela 5.2.** Wyniki pomiarów napięcia akumulatorów wykonane podczas rozładowywania na zewnątrz przy temperaturze 5°C i nasłonecznieniu 400 W/m<sup>2</sup> (z udziałem panelu)

Zmierzone napięcie [V]	Godzina wykonania pomiaru
4,8	10:25
4,54	10:45
4,12	10:55
3,98	11:00

Według wykonanych pomiarów doładowywanie akumulatorów przez panel fotowoltaiczny nie odbywa się w niskiej temperaturze (w tym przypadku 5°C).

### 5.3. Absorbowanie natężenia słonecznego

Kolejne badania miały na celu sprawdzenie jak będą ładować się akumulatory, gdy panel fotowoltaiczny zostanie skierowany w stronę padających promieni słonecznych. Przez 30 minut panel skierowany był pod kątem około 45° do podłoża. Ważnym aspektem tych testów było to, że przez cały czas wykonywania badań silniki pojazdu napędzały koła, które nie stykały się z powierzchnią ziemi. Dzięki temu zabiegowi, mniej energii jest wytrącane na pokonanie siły tarcia i łatwiej można zaobserwować zmiany napięcia generowanego na akumulatorach. Celem badania jest wykazanie, że w czasie pracy silników akumulator pojazdu jest w stanie doładowywać się. Ustawienie panelu fotowoltaicznego w stronę słońca daje możliwość większego poboru energii słonecznej, a w konsekwencji możliwość dostarczenia większej ilości energii elektrycznej do akumulatorów samochodu.



Rys 5.3. Charakterystyka doładowywania akumulatorów przez panel ustawiony pod kątem 45° do podłoża

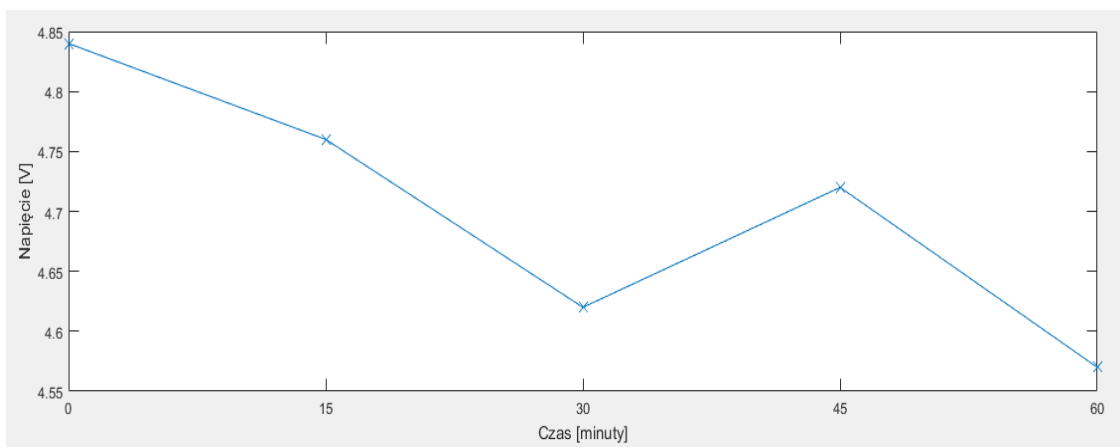
**Tabela 5.3.** Wyniki badań otrzymanych w sytuacjach gdy samochód jest ustawiony równolegle do powierzchni ziemi oraz momencie kiedy ustawiony jest w stronę słońca

Numer pomiaru	Godzina	Napięcie [V]	Uzyskane natężenie słoneczne [ $W/m^2$ ] dla ustawienia pojazdu równolegle do ziemi	Uzyskane natężenie słoneczne [ $W/m^2$ ] dla ustawienia pojazdu w kierunku słońca
1	12:25	4,65	87	265
2	12:40	4,92	122	367,5
3	12:55	4,496	44	191

Samochód elektryczny powinien zostać tak skonstruowany, aby panel fotowoltaiczny miał możliwość śledzenia położenia słońca. Dzięki temu ogniwa fotowoltaiczne będą miały możliwość absorpcji większej ilości energii słonecznej. W tabeli 5.3 zostało wykazane porównanie poboru energii słonecznej przy położeniu panelu równolegle do podłoża oraz pod kątem  $\approx 45^\circ$ , dla którego charakterystyka doładowywania została zaprezentowana na rysunku 5.3. Pożądanym efektem zastosowania panelu słonecznego jest doładowywanie się akumulatorów w czasie jazdy, a nie tylko mniejsze ich rozładowywanie. Aby uzyskać taki efekt to dobór mocy panelu musi być odpowiedni to ciężaru samochodu, ponieważ potrzeba więcej energii elektrycznej gdy pojazd musi pokonać większą siłę tarcia. Przeszkodą było pojawiające się zachmurzenie, jednak trzeba zdać sobie sprawę, że autonomiczne samochody elektryczne zasilane panelem słonecznym będą wykorzystywane w różnych warunkach atmosferycznych, nie tylko w najbardziej słoneczne dni.

#### **5.4. Rozładowywanie się akumulatorów pojazdu przy różnych ustawieniach pojazdu**

Trzecie badania były hybrydą wyżej wymienionych testów. We wtorek 14 listopada, zostały przeprowadzone godzinne badania sprawdzające jak szybko będą się rozładowywać akumulatory podczas jazdy samochodu. Przez pierwsze pół godziny pojazd poruszał się po powierzchni. Następnie samochód został tak skierowany w stronę słońca, aby panel fotowoltaiczny był w stanie absorbować jak najwięcej promieniowania słonecznego – panel był ustawiony pod kątem  $\approx 45^\circ$  do powierzchni, a koła nie stykały się z podłożem. Samochód był ciągle w czasie pracy – silniki napędzały koła, a panel fotowoltaiczny dostarczał do układu dodatkową energię słoneczną. Natężenie słoneczne zostało mierzone przy użyciu pyranometru. Za każdym razem przy badaniu mocy energii słonecznej, przyrząd był ustawiany równolegle do panelu fotowoltaicznego. W dniu 14 listopada było  $5^\circ C$ , w tym okresie były to najbardziej słoneczne dni.



Rys 5.4. Wykres napięcia od czasu, badanie z dnia 14 listopada 2017 roku

Duża siła nacisku, w przybliżeniu 110 N, wywołana przez ciężar panelu fotowoltaicznego oraz nieoptymalne ustawienie panelu sprawiły, że więcej energii elektrycznej zostało zużyte do napędzania pojazdu. W ciągu 30 minut akumulatory rozładowały się o 0,22 V. Charakterystykę zaprezentowano na rysunku 5.4. Mimo dobrego nasłonecznienia akumulatory nie naładowały się. Następnie samochód nie poruszał się po planszy, ale został ustawiony pod takim kątem, aby pobierać więcej energii słonecznej. Wyniki zostały zaprezentowane w tabeli 5.4 od punktu 3. Można zauważyć, że w ciągu 15 minut akumulatory zostały doładowane o 0,1, a w ciągu 30 minut napięcie na akumulatorach zmniejszyło się jedynie o 0,05 V. Jak widać w tabeli 5.4 nasłonecznienie zmniejszyło się przy drugim rodzaju testów.

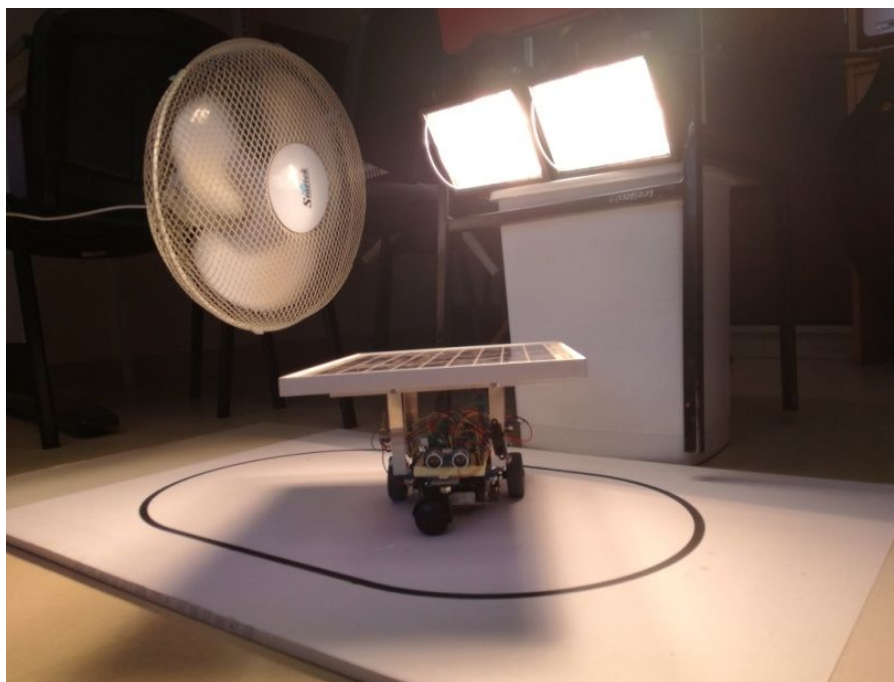
Tabela 5.4. Wykaz pomiarów otrzymanych podczas badania wykonanego dnia 14 listopada 2017 roku

Numer pomiaru	Godzina	Napięcie [V]	Natężenie słoneczne [ $\text{W/m}^2$ ]. Panel ustawiony równolegle do powierzchni	Natężenie słoneczne [ $\text{W/m}^2$ ]. Panel ustawiony pod kątem $\approx 45^\circ$ do powierzchni
1	11:30	4,84	235	440
2	11:45	4,76	147	340
3	12:00	4,62	88	162
4	12:15	4,72	60	107
5	12:30	4,57	62	124

### 5.5. Badania przeprowadzone przy użyciu halogenów.

Ze względu na nieodpowiednie warunki klimatyczne zdecydowano, aby przeprowadzić badania w pomieszczeniu. W przypadku, gdy „rolę słońca” pełnią halogeny. Przy wykonywanych badaniach istotna jest odpowiednia odległość panelu od halogenów. Zbyt mała może spowodować szybkie nagrzewanie się panelu słonecznego, co skutkowałoby utratą sprawności (halogeny oddają dużo energii w postaci ciepła). Im większy jest dystans pomiędzy panelem, a halogenami tym mniejsze jest natężenie mocy [ $\text{W/m}^2$ ].

Podczas wykonywanych badań panel ustawiono w odległości około 30 centymetrów od halogenów, a samochód kręcił się w kółko. Natężenie mocy światła widzialnego jakie uzyskano na wysokości panelu wyniosło w przybliżeniu  $400 \text{ W/m}^2$ . Stanowisko badań zostało przedstawione na rysunku 5.5.



Rys 5.5. Stanowisko badań doładowywania akumulatorów przy wykorzystaniu halogenów

Dnia 05.12.2017 wykonano dwie próby doładowywania akumulatorów. Każda trwała 1 godzinę, a pomiary odbywały się co 15 minut. Pierwszą rozpoczęto od wartości napięcia 5,02 V wyniki przedstawiono w tabeli 5.5, a wykreślone charakterystyki na rysunku 5.6.

**Tabela 5.5.** Pomiary napięcia rozpoczęte od wartości 5,02 V (z udziału panelu)

Zmierzone napięcie [V]	Godzina wykonania pomiaru
5,02	11:50
5,02	12:10
5,01	12:25
5,01	12:40
5,00	12:55

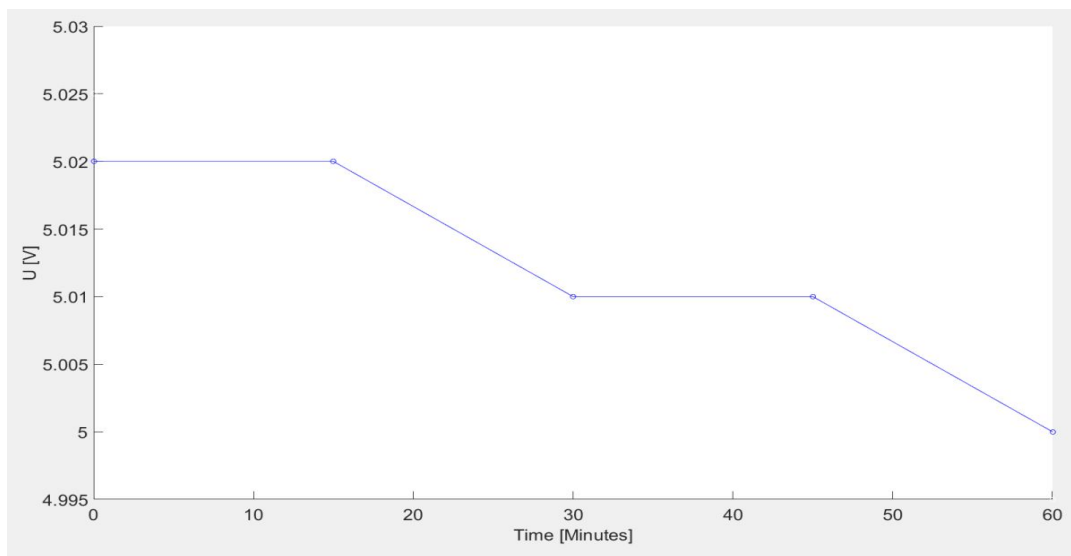
Próba druga odbyła się po ówczesnym rozładowaniu akumulatorów do wartości 4,95 V. Jako, że przetwornica do której podłączony jest panel daje napięcie na wyjściu o wartości 5.1 V, podczas pracy optymalnej. Dlatego dla wartości niższej możliwym jest zaobserwowanie doładowywania akumulatorów. Wyniki pomiarów zaprezentowano w tabeli 5.6, a wykreślone charakterystyki na rysunku 5.7.

**Tabela 5.6.** Pomiary napięcia rozpoczęte od wartości 4,95 V (z udziału panelu)

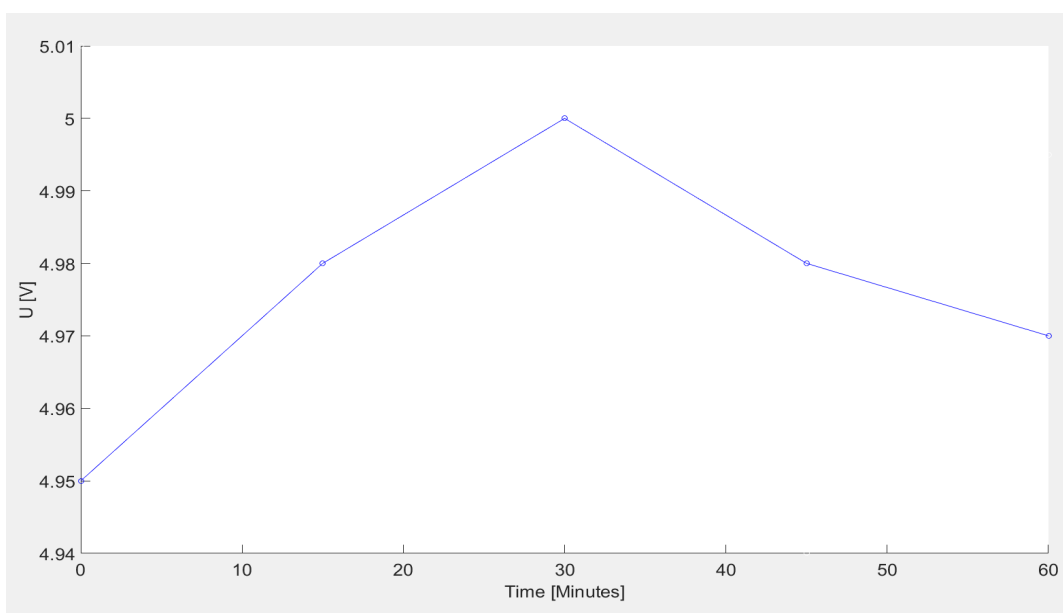
Zmierzone napięcie [V]	Godzina wykonania pomiaru
4,95	11:50
4,98	12:10
5,00	12:25
4,98	12:40
4,97	12:55

Aby porównać czas rozładowywania się akumulatorów wykonano dodatkowe badania dnia 06.12.2017, również jednogodzinne. Próby nie trzeba było powtarzać, ponieważ po pierwszej wartości napięcia akumulatorów spadła poniżej 4,95, dzięki czemu udowodniono

wydłużenie pracy akumulatorów. Pomiary zaprezentowano w tabeli 5.6, a charakterystykę na rysunku 5.8.



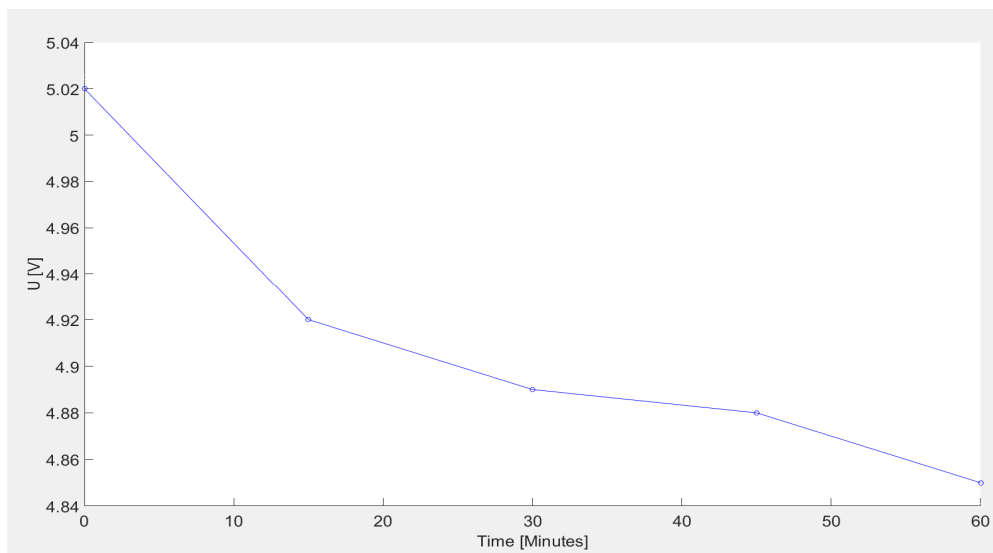
Rys. 5.6. Charakterystyki zmiany napięcia akumulatora od wartości 5.02 V (z udziałem panelu)



Rys. 5.7. Charakterystyki zmiany napięcia akumulatora od wartości 4.95 V (z udziałem panelu)

**Tabela 5.7.** Pomiary napięcia rozpoczęte od wartości 5.02 V (bez udziału panelu)

Zmierzone napięcie [V]	Godzina wykonania pomiaru
5,02	17:50
4,92	18:05
4,89	18:20
4,88	18:35
4,85	18:50



Rys.5.8. Charakterystyki zmiany napięcia akumulatora od wartości 5.02 V (bez udziału panelu)

Z wykonanych wyżej pomiarów wynika, że zastosowanie panelu fotowoltaicznego jako źródła dodatkowej energii elektrycznej nie tylko umożliwia wydłużenie pracy małego pojazdu, ale również może zapewnić doładowywanie się akumulatorów w trakcie jazdy szybciej niż są one rozładowywane.

### 5.6. Wnioski, podsumowanie i przyszły rozwój projektu

Autonomiczny pojazd elektryczny napędzany panelem fotowoltaicznym jest projektem innowacyjnego samochodu, który w przyszłości może być codziennym środkiem komunikacji. Pytanie tylko jak szybko do tego dojdzie. Bez odpowiednich badań, testów i inwestycji nie ma szans, aby rozwój tego rodzaju pojazdów wzrastał w znaczącym tempie. Jest to zaawansowana technologia, która jest przyjazna środowisku, ponieważ wykorzystywane zostało odnawialne źródło energii – promieniowanie słoneczne.

Panel fotowoltaiczny ma za zadanie doładowywać akumulatory podczas jazdy samochodu. W momencie postoju wymagany jest efekt szybszego wzrostu napięcia na bateriach pojazdu, ponieważ układ elektroniczny nie pobiera tak dużo mocy jak w przypadku poruszania się po drodze. Zostało to zaobserwowane podczas wykonywanych badań, że przy odpowiednim oświetleniu pojazd jest w stanie jeździć jedynie za pomocą energii słonecznej dostarczanej przez panel fotowoltaiczny. Przy słabszym oświetleniu należy zastosować rozruch silników, poprzez chwilowe włączenie zasilania, a pojazd również będzie jeździł dzięki zjawisku fotowoltaicznemu. Dobre warunki atmosferyczne – wysokie promieniowanie słoneczne, brak zachmurzenia oraz odpowiednia temperatura pozwalają na to, aby energia pochodząca z panelu słonecznego zasilila układu elektryczny. W wysokiej temperaturze zwiększają się drgania sieci krystalicznej ogniw fotowoltaicznych, co powoduje zaburzony przepływ elektronów. Konsekwencją tego zjawiska jest generowanie niższego napięcia, przez co zmniejsza się wartość dostarczanej mocy do układu. Z drugiej jednak strony elementy elektroniczne w niskiej temperaturze nie działają tak jak powinny i nie mogą spełnić swoich

funkcji w taki sposób jaki zostały zaprojektowane. Zostało to zaobserwowane podczas badań, gdy temperatura oscylowała wokół 0 °C. Informacje te zostały również zawarte w książce Włodzimierza Janke „Impulsowe przetwornice napięcia stałego” [5.1].

W dalszym rozwoju naszego projektu inżynierskiego proponujemy, aby elektronika została odpowiednio zabezpieczona np. po przez umiejscowienie jej w przystosowanej do tego obudowie. Płytką stykowa może zostać zastąpiona w odpowiedni sposób wytrawioną płytką drukowaną. Dzięki temu elektronika będzie pracować i wypełniać swoje funkcje w należyty sposób mimo niskiej temperatury otoczenia. Innym zjawiskiem, które miało wpływ na pracę pojazdu jest nierównomierne rozkładanie się światła pochodzącego od lamp halogenowych. Nierównomierne naświetlenie, oraz wysyłane światło nieprostopadłego do powierzchni toru jazdy spowodowało, że czujniki odbiciowe odbierały dodatkowe, zakłóceniewe sygnały, przez co pojazd nie jeździł zgodnie z torem jazdy.



*Rys. 5.9. Zdjęcie robota wykonane podczas badań na dworze*



## Wykaz Literatury

- 1.1. Litman T.: *Autonomous Vehicle Implementation Predictions: Implications for Transport Planning*, Victoria Transport Policy Institute 2017.
- 1.2. McKerracher C., Tryggstad C. i inni: *An integrated perspective on the future of mobility*, McKinsey & Company and Bloomberg New Energy Finance 2017.
- 1.3. *Poznaj Otto, autonomiczną ciężarówkę. Nie uwierzysz, co wiozła w trakcie swojego dziewiczego rejsu*, <http://innpoland.pl/130687.poznaj-otto-autonomiczna-ciezarowke-nie-uwierzysz-co-wiozla-w-trakcie-swojego-dziewiczego-rejsu> (data dostępu 07.12.2017 r.).
- 1.4. *Autopilot za kierownicą. Tesla pokazała, jak jeździ autonomiczny samochód*, <http://www.polsatnews.pl/wiadomosc/2016-11-22/autopilot-za-kierownica-tesla-pokazala-jak-jezdzi-autonomiczny-samochod/> (data dostępu 07.12.2017 r.).
- 2.1. Fic B.: *Samochody elektryczne*, Wydawnictwo i Handel Książkami „KaBe”, Krosno 2015.
- 2.2. Złącze p-n, <http://instalacje2004.republika.pl/Polprze/diody.html> (data dostępu 09.12.2017 r.).
- 2.3. Klugmann-Radziemska E.: *Fotowoltaika w teorii i w praktyce*, Wydawnictwo BTC.
- 2.4. Kordowski M., Jabłoński M., Bartosiewicz K., Rybusiński J.: *Badanie ogniw fotowoltaicznych*, Uniwersytet młodych wynalazców, Uniwersytet Warszawski, Wydział Fizyki.
- 2.5. Sołowiej P., Neugebauer M., Wesołowski M., Tejszowski M.: *Porównanie uzysków energii elektrycznej z poziomych polikrystalicznych i pionowych monokrystalicznych instalacji fotowoltaicznych w warunkach północno-wschodniej Polski*, Uniwersytet Warmińsko-Mazurski, Katedra Elektrotechniki Energetyki Elektroniki i Automatyki, 2017.
- 3.1. Dokumentacja Arduino Nano (V3.0), [http://www.mouser.com/pdfdocs/Gravitech\\_Arduino\\_Nano3\\_0.pdf](http://www.mouser.com/pdfdocs/Gravitech_Arduino_Nano3_0.pdf) (data dostępu 07.12.2017 r.).
- 3.2. Dokumentacja sterownika L293D, [https://www.arduino.cc/documents/datasheets/H-bridge\\_motor\\_driver.PDF](https://www.arduino.cc/documents/datasheets/H-bridge_motor_driver.PDF) (data dostępu 07.12.2017 r.).
- 3.3. Dokumentacja czujnika odbicowego QRD1113/QRD1114, [http://www.farnell.com/datasheets/2281687.pdf?\\_ga=2.150037809.600619424.15108%2041481-%201909192528.1508937344](http://www.farnell.com/datasheets/2281687.pdf?_ga=2.150037809.600619424.15108%2041481-%201909192528.1508937344) (data dostępu 07.12.2017 r.).
- 3.4. Dokumentacja czujnika ultradźwiękowego HC-SR04, <https://forbot.pl/blog/kurs-arduino-czujnik-odleglosci-hc-sr04-funkcje-id4290> (data dostępu 09.12.2017 r.).
- 3.5. Charakterystyka promieniowania czujnika ultradźwiękowego HC-SR04, <http://www.buildcircuit.com/simple-ultrasonic-range-finder-using-hc-sr04> (data dostępu 09.12.2017 r.).
- 3.6. Opis wprowadzeń i wykres napięcia analogowego czujnika odległości GP2Y0A21YK0F, <http://electropark.pl/czujniki-optyczne/41-sharp-gp2y0a21yk0f-czujnik-odleglosci-10-80cm.html> (data dostępu 09.12.2017 r.).
- 3.7. *Sprawdzone sposoby na mierzenie stanu baterii*, <https://forbot.pl/blog/sprawdzone-sposoby-na-mierzenie-stanu-baterii-id7306> (data dostępu 07.12.2017 r.).
- 3.8. *BU-402: What Is C-rate?*, [http://batteryuniversity.com/learn/article/what\\_is\\_the\\_c\\_rate](http://batteryuniversity.com/learn/article/what_is_the_c_rate) (data dostępu 07.12.2017 r.).
- 3.9. Ödön F.: *Zasilanie układów elektronicznych: Zasilacze ze stabilizatorami o pracy ciągłej: Przetwornice DC-DC*, Wydawnictwa Naukowo-Techniczne, Warszawa 1998.
- 5.1. Janke W.: *Impulsowe przetwornice napięcia stałego*, Politechnika Koszalińska.

## Wykaz Rysunków

1.1. Ciężarówka Volvo z systemem sterowania OTTO (Za kierownicą nie znajduje się człowiek) .....	12
1.2. Wnętrze pojazdu tesli podczas pracy autonomicznej [1.3] .....	13
1.3. Autonomiczny pojazd elektryczny zasilany panelem fotowoltaicznym [1.4] .....	14
2.1. Przedstawienie równowagi w obszarze złącza p-n [2.2.] .....	16
2.2. Model elektryczny ogniwa fotowoltaicznego .....	16
2.3. Schemat budowy ogniwa fotowoltaicznego [2.4] .....	17
2.4. Sprawność uzyskanej energii elektrycznej z paneli fotowoltaicznych monokrystalicznych (kolor czerwony) i polikrystalicznych (kolor niebieski) [2.6] .....	18
2.5. Panel Fotowoltaiczny wykorzystany do badań .....	19
2.6. Dane techniczne panelu fotowoltaicznego .....	20
2.7. Schemat elektryczny układu pomiarowego prądu i napięcia .....	21
2.8. Charakterystyki prądowo-napięciowe panelu fotowoltaicznego .....	23
2.9. Przebiegi mocy od napięcia panelu fotowoltaicznego .....	23
2.10. Mały pojazd z aluminiową konstrukcją podtrzymującą panel fotowoltaiczny .....	24
3.1. Schemat elektryczny zbudowanego układu sterowania pojazdem [3.1] [3.2] .....	26
3.2. Płyta Arduino Nano wykorzystana w projekcie .....	27
3.3. Silnik wraz z przekładnią i mocowaniem .....	29
3.4. Schemat przykładowego mostka H .....	30
3.5. Schemat elektryczny sterownika L293D [3.2] .....	31
3.6. Płyta stykowa z układem elektrycznym obsługującym trzy czujniki optyczne w kolorze czarnym umieszczone na środku płytki .....	32
3.8. Czujnik ultradźwiękowy HC-SR04, wykorzystany w projekcie .....	33
3.9. Przykładowe połączenie czujnika HC-SR04 z Arduino Leonardo [3.4] .....	33
3.10. Przedstawienie działania czujnika HC-SR04 .....	34
3.11. Katalogowa charakterystyka promieniowania czujnika HC-SR04 .....	35
3.12. Przykładowy program do czujnika ultradźwiękowego [3.5] .....	35
3.13. Prezentacja pomiarów otrzymanych przy badaniu czujnika .....	36
3.14. Czujnik podczerwieni GP2Y0A02YK0F, wykorzystany w projekcie .....	37
3.15. Opis wyprowadzeń czujnika GP2Y0A02YK0F [3.6] .....	37
3.16. Uproszczony system detekcji zbliżeniowej .....	37
3.17. Katalogowy wykres napięcia zależnego od odległości [3.6] .....	38
3.18. Przykładowy program obsługi czujnika podczerwieni GP2Y0A02YK0F .....	39
3.19. Tablica pomiarów z badania działania czujnika podczerwieni .....	39
3.20. Charakterystyki rozładowania ogniwa niklowo-kadmowego [3.7] .....	40
3.21. Przetwornica izolowana DC/ DC podłączona do układu elektronicznego .....	41
4.1. Aplikacja środowiska Arduino IDE .....	43
4.2. Działanie kodu programu zaprezentowane w formie schematu blokowego .....	45
4.3. Początek programu związany z inicjalizacją zmiennych .....	46

4.4. Funkcja setup .....	47
4.5. Początek funkcji, odpowiedzialnej za sczytywanie czarnej linii .....	48
4.6. Dalsza część funkcji do sczytywania osi jazdy. Instrukcje warunkowe .....	49
4.7. Zobrazowanie ruchu pojazdu, w momencie gdy zgubi linię .....	50
4.8. Funkcja, która steruje dwoma czujnikami ultradźwiękowymi .....	50
4.9. Zdjęcie głośnika używanego w projekcie .....	51
4.10. Zdjęcie przodu robota bez panelu fotowoltaicznego .....	51
4.11. Funkcja obsługująca analogowy czujnik podczerwieni oraz error .....	52
4.12. Funkcja do podawania prędkości na odpowiednie koła .....	52
4.13. Funkcje odpowiedzialna za ruch pojazdu do przodu (po lewej) i do tyłu (po prawej)	53
4.14. Funkcja loop .....	54
4.15. Kod inicjalizacji zmiennych i funkcji setup .....	55
4.16. Funkcje użyte w programie, od góry: direct, us, stay .....	57
4.17. Funkcję front i back odpowiedzialne za wyznaczanie kierunku jazdy .....	57
4.18. Główny kod programu, czyli funkcja loop .....	59
5.1. Pyranometr, używany podczas badań .....	61
5.2. Trasy wykorzystane do badań sprawności działania „line followera” (po lewej trasa pierwsza, a po prawej druga) .....	62
5.3. Charakterystyka doładowywania akumulatorów przez panel ustawiony pod kątem 45° do podłoża .....	64
5.4. Wykres napięcia od czasu, badanie z dnia 14 listopada 2017 roku .....	66
5.5. Stanowisko badań doładowywania akumulatorów przy wykorzystaniu halogenów ...	67
5.6. Charakterystyki zmiany napięcia akumulatora od wartości 5.02 V (z udziałem panelu) .....	68
5.7. Charakterystyki zmiany napięcia akumulatora od wartości 4.95 V (z udziałem panelu) .....	68
5.8. Charakterystyki zmiany napięcia akumulatora od wartości 5.02 V (bez udziału panelu) .....	69
5.9. Zdjęcie robota wykonane podczas badań na dworze .....	70

## Wykaz tabel

2.1. Pomiary wartości napięcia i prądu otrzymane w warunkach zbliżonych do STC – 850 W/m <sup>2</sup> , 28°C .....	21
2.2. Pomiary wartości napięcia i prądu otrzymane w warunkach: 500 W/m <sup>2</sup> , 16°C .....	22
3.1. Wykaz wyprowadzeń na platformie programistycznej Arduino Nano .....	28
4.1. Wykaz nazw zmiennych oraz połączeń z pinami sterownika Arduino .....	44
4.2. Wykaz nazw zmiennych .....	44
5.1. Wyniki pomiarów napięcia akumulatorów podczas rozładowywania z dnia 8 i 9 listopada wykonane bez udziału panelu .....	63
5.2. Wyniki pomiarów napięcia akumulatorów wykonane podczas rozładowywania na zewnątrz przy temperaturze 5oC i nasłonecznieniu 400 W/m <sup>2</sup> (z udziałem panelu) .....	64
5.3. Wyniki badań otrzymanych w sytuacjach gdy samochód jest ustawiony równolegle do powierzchni ziemi oraz momencie kiedy ustawiony jest w stronę słońca .....	65
5.4. Wykaz pomiarów otrzymanych podczas badania wykonanego dnia 14 listopada 2017 roku .....	66
5.5. Pomiary napięcia rozpoczęte od wartości 5,02 V (z udziału panelu) .....	67
5.6. Pomiary napięcia rozpoczęte od wartości 4,95 V (z udziału panelu) .....	67
5.7. Pomiary napięcia rozpoczęte od wartości 5.02 V (bez udziału panelu) .....	68