

Politechnika Warszawska

WYDZIAŁ MECHANICZNY
ENERGETYKI I LOTNICTWA



Instytut Techniki Lotniczej i Mechaniki Stosowanej

Praca dyplomowa magisterska

na kierunku Automatyka i Robotyka
w specjalności Robotyka

Opracowanie układu automatycznego
sterowania ruchem kolejowym na makiecie modelarskiej

Łukasz Wach

Numer albumu 252079

promotor
dr inż. Andrzej Chmielniak

Warszawa, 2019

Streszczenie

Tematem niniejszej pracy jest opracowanie układu automatycznego sterowania ruchem kolejowym na makiecie modelarskiej. Projekt obejmuje przygotowanie makiety z torowiskiem i peryferiami, skonstruowanie układu sterowania oraz zaimplementowanie algorytmu kontroli ruchu kolejowego. Praca powinna spełniać wyznaczone przez autora założenia, do których należy przede wszystkim zapewnienie bezkolizyjnej jazdy lokomotyw oraz kontrolowanie ustalonego wzorca zachowań w strefach postojowych i mijania.

W pierwszej części pracy przedstawiono konspekt teoretyczny uwzględniający systemy sterowania w rzeczywistym ruchu kolejowym (SRK) oraz w modelarstwie. Omówiono także dwa główne rodzaje sterowania stosowane w makietach kolejowych, czyli sterowanie analogowe oraz cyfrowe. Scharakteryzowano najczęściej wykorzystywane obwody torowe oraz wyjaśniono koncepcję protokołu komunikacyjnego DDC (*ang. Digital Command Control*).

Druga część pracy poświęcona jest opisowi opracowanego układu oraz analizie przeprowadzonych badań. Autor najpierw przedstawił zaprojektowany model makiety, uwzględniając ukształtowanie torowiska oraz planowane strefy postojowe. Następnie omówił budowę elektronicznego układu sterowania, który został podzielony na część decyzyjną oraz wykonawczą. Pierwszą z nich stanowi centrala nadzorująco-sterująca, której zadaniem jest monitorowanie stanu czujników oraz kontrola ruchu kolejowego. Druga część odpowiada natomiast za bezpośrednie sterowanie silnikiem lokomotywy. Cały układ komunikuje się ze sobą za pomocą modułów radiowych, pełniących rolę zarówno odbiornika jak i nadajnika.

W dalszej części pracy przedstawiono strukturę oraz metodę działania algorytmu automatycznego sterowania. Zaimplementowany program sporządzony został w formie obiektowej przy pomocy środowiska Arduino. Komunikacja oparta została na modelu master-slave, w którym centrala przesyła rozkazy oraz odpytuje jednostki wykonawcze znajdujące się bezpośrednio w składach pociągowych.

Na koniec przeprowadzono badania funkcjonalności i zachowania układu. Stwierdzono, że zaprojektowany system spełnia przyjęte założenia, jednakże nie jest wystarczająco rozbudowany, aby zostać wdrożonym do rzeczywistych układów sterowania ruchem kolejowym.

SŁOWA KLUCZOWE: sterowanie ruchem kolejowym, modelarstwo kolejowe, komunikacja radiowa, Arduino, sterowanie cyfrowe, DDC, enkodery magnetyczne

Abstract

The subject of the thesis is the development of an automatic railway traffic control system on a model mockup. The project includes the preparation of the mockup with the trackway and peripheries, the construction of the control system and the implementation of the railway traffic control algorithm. The thesis should meet the assumptions set by the author, to which belongs primarily ensuring collision-free locomotives driving and controlling the determined behavior pattern in the stopping and passing zones.

The first part of the thesis presents a theoretical outline that takes into account real railway traffic and modeling control systems. Two main types of control used in railway models, the analog and digital control, were also discussed. Furthermore, the most commonly used trackway circuits were characterized and the concept of the DDC (Digital Command Control) communication protocol was explained.

The second part of the thesis is devoted to the description of the developed system and the analysis of the conducted research. The author firstly introduced a model mockup, taking into account the shape of the trackway and the planned stopping zones. Then he discussed the construction of the electronic control system, which was divided into the decision-making and the executive parts. The first of these is the controlling and supervising unit, whose task is to monitor the state of sensors and control railway traffic. The second part is responsible for direct control of the locomotive engine. The whole system communicates with each other using radio modules, acting both as a receiver and a transmitter.

The further part of the thesis presents the structure and operation method of the automatic control algorithm. The implemented program was prepared in object form using the Arduino environment. The communication was based on a master-slave model, in which the control unit sends the orders and asks execution units located in the train sets.

At the end, functional and system behaviour tests were performed. It was found that the designed system meets the assumptions, however, it is not sufficiently developed to be implemented into real railway traffic control systems.

KEY WORDS: railway traffic control, railway modeling, radio communication, Arduino, digital control, DDC, magnetic encoders

Oświadczenie autora (autorów) pracy

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

Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płyce kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

.....
data

.....
podpis autora (autorów) pracy

Oświadczenie

Wyrażam zgodę / nie wyrażam zgody^{*1} na udostępnianie osobom zainteresowanym mojej pracy dyplomowej. Praca może być udostępniana w pomieszczeniach biblioteki wydziałowej. Zgoda na udostępnienie pracy dyplomowej nie oznacza wyrażenia zgody na jej kopiowanie w całości lub w części.

Brak zgody nie oznacza ograniczenia dostępu do pracy dyplomowej osób:

- reprezentujących władze Politechniki Warszawskiej,
- członków Komisji Akredytacyjnych,
- funkcjonariuszy służb państwowych i innych osób uprawnionych, na mocy odpowiednich przepisów prawnych obowiązujących na terenie Rzeczypospolitej Polskiej, do swobodnego dostępu do materiałów chronionych międzynarodowymi przepisami o prawach autorskich. Brak zgody nie wyklucza także kontroli tekstu pracy dyplomowej w systemie antyplagiatowym.

.....
data

.....
podpis autora (autorów) pracy

^{*1} - niepotrzebne skreślić

Spis treści

| | | |
|---------|--|----|
| 1. | Wprowadzenie | 7 |
| 2. | Wstęp teoretyczny | 9 |
| 2.1. | Systemy sterowania w rzeczywistym ruchu kolejowym | 9 |
| 2.2. | Systemy sterowania w modelarstwie kolejowym | 12 |
| 2.2.1. | Sterowanie analogowe | 12 |
| 2.2.2. | Sterowanie cyfrowe | 16 |
| 3. | Cel i zakres pracy | 23 |
| 4. | Opis projektu makiety kolejowej | 25 |
| 4.1. | Torowisko PIKO A-Gleis | 25 |
| 4.2. | Zasilacz i regulator napięcia | 26 |
| 4.3. | Lokomotywy PIKO BR 218 | 27 |
| 4.4. | Podsumowanie | 28 |
| 5. | Opis układu sterowania i komunikacji | 29 |
| 5.1. | Zestawienie elementów i urządzeń wchodzących w skład systemu automatycznego sterowania | 29 |
| 5.1.1. | Płytki modułowa Arduino Leonardo | 30 |
| 5.1.2. | Płytki modułowa Arduino Pro Mini 328 | 30 |
| 5.1.3. | Moduł radiowy nRF24L01 | 31 |
| 5.1.4. | Stabilizatory liniowe L7805CV oraz LM1117T | 31 |
| 5.1.5. | Dwukanałowy sterownik silników TB6612FNG | 32 |
| 5.1.6. | Czujnik odbiciowy KTIR0711S | 32 |
| 5.1.7. | Ekspander wyprowadzeń mikrokontrolera PCF8574 | 33 |
| 5.1.8. | Enkoder magnetyczny Pololu 3081 | 33 |
| 5.1.9. | Akumulator Li-Pol XP17002EX | 34 |
| 5.1.10. | Moduł przekaźnikowy SRD-05VDC-SL-C | 34 |

| | | |
|--------|---|----|
| 5.2. | Układ sterowania lokomotywą..... | 35 |
| 5.3. | Układ centrali nadzorująco-sterującej..... | 38 |
| 6. | Opis algorytmu automatycznego sterowania ruchem kolejowym. | 40 |
| 6.1. | Algorytm centrali nadzorująco-sterującej..... | 41 |
| 6.1.1. | Ogólna struktura programu..... | 42 |
| 6.1.2. | Komunikacja radiowa | 45 |
| 6.1.3. | Metoda auto-lokalizacji lokomotyw..... | 47 |
| 6.1.4. | Metoda kontroli i sterowania ruchem kolejowym..... | 47 |
| 6.2. | Algorytm sterowania lokomotywą | 51 |
| 7. | Analiza przeprowadzonych badań i eksperymentów..... | 53 |
| 7.1. | Badanie funkcjonalności układu pod kątem spełnienia założonych kryteriów..... | 53 |
| 7.2. | Badanie profilu prędkości lokomotyw..... | 59 |
| 7.3. | Badanie pomiarów przebytej drogi | 63 |
| 8. | Wnioski oraz propozycje dalszego rozwoju pracy..... | 66 |
| | Spis rysunków..... | 68 |
| | Bibliografia..... | 70 |

1. Wprowadzenie

Historia pojazdów szynowych sięga aż do czasów starożytności, o czym świadczą odnalezione przez archeologów ślady kamiennych torowisk. Znacznie lepiej zachowane, aczkolwiek nadal prymitywne tory odnaleziono także w pozostałościach kopalń powstałych w epoce średniowiecza. Główną siłą napędową w tamtych czasach była grawitacja, a także mięśnie zwierząt pociągowych i człowieka. Dopiero na przełomie XVIII i XIX skonstruowano pierwszy pojazd szynowy z napędem mechanicznym. Jednym z najbardziej znanych ówczesnych wynalazców był George Stephenson, który wybudował w pełni funkcjonalny dwuosiowy parowóz adhezyjny (Rysunek 1.1). Niedługo potem, w roku 1825 otworzył pierwszą publiczną linię kolejową łączącą Stockton i Darlington. Wydarzenie to zapoczątkowało wielkoskalową rozbudowę transportu kolejowego, a także miało ogromny wpływ na rozwój przemysłu [1].

Nieustanna modernizacja i postęp technologiczny w branży kolejowej doprowadziła do powstania nowych źródeł napędu, począwszy od silników spalinowych i elektrycznych, poprzez napędy aerodynamiczne odrzutowe aż po elektromagnetyczne. Zmianom uległy również materiały wykorzystywane do konstrukcji lokomotyw i wagonów oraz torowisk i trakcji. Stare, toporne i w pełni stalowe nadwozia zastąpiono nowoczesnymi i aerodynamicznymi, wykonanymi z aluminium lub kompozytu konstrukcjami. Wraz z nastaniem XX wieku oraz rozwojem automatyki i elektroniki zaczęto wyposażać pociągi w rozbudowane systemy bezpieczeństwa i komunikacji. Położono również większy nacisk na wygodę pasażerów projektując specjalnie amortyzowane podwozia oraz wytłumione, przechylne nadwozia [2].

W obecnych czasach dąży się do wprowadzenia i rozprzestrzenienia bezzałogowych składów kolejowych. W niektórych miastach świata takich jak Lyon, Paryż czy Hong Kong działają już w pełni autonomiczne linie metra lub kolejki podwieszane. Jednakże prawdziwe wyzwanie stanowi automatyzacja dalekosiężnych kolei naziemnych, które w odróżnieniu od metra są bardziej podatne na działanie czynników zewnętrznych takich jak warunki pogodowe lub niebezpieczeństwo kolizji.

Uproszczony model automatycznego sterowania koleją naziemną można zbudować wykorzystując szeroko dostępne na rynku produkty z branży modelarstwa kolejowego. Większość z nich korzysta ze standardu DCC, który pozwala na sterowanie lokomotywami i infrastrukturą za pomocą specjalnego sterownika lub komputera, bez konieczności prowadzenia indywidualnych przewodów sterujących do każdego urządzenia. Budowa takiego

modelu niesie ze sobą jednak duże koszty i przeważnie jest dość ograniczona w kwestii automatyzacji kontroli ruchu.

W niniejszej pracy autor przedstawi alternatywną i bardziej ekonomiczną metodę automatycznego sterowania makietą kolejową. W tym celu wykorzystany zostanie gotowy model z zasilaniem analogowym oraz odpowiednio zaprojektowany układ elektroniczny zapewniający bezprzewodową komunikację. Ponadto przeprowadzone zostaną badania nad jakością działania układu oraz analiza możliwości zastosowania opracowanego systemu w rzeczywistych układach sterowania ruchem kolejowym.



Źródło: modelnet.pl

Rysunek 1.1 Model pierwszej lokomotywy George'a Stephensona - „Rakiety”

2. Wstęp teoretyczny

2.1. Systemy sterowania w rzeczywistym ruchu kolejowym

Organizacja ruchu kolejowego w dzisiejszych czasach wymaga wydajnych oraz zapewniających najwyższe bezpieczeństwo systemów sterowania. Prawdłowo zaprojektowany układ SRK (Sterowania Ruchem Kolejowym) powinien spełniać następujące wymagania [3]:

- Ustawianie zwrotnic zgodnie z planowaną drogą przejazdu pociągu oraz kontrola i utrzymanie ich stanu do momentu zakończenia manewru.
- Sprawdzanie oraz utrzymywanie stanu niezajętości torów i zwrotnic, które znajdują się na drodze przejazdu po nadaniu sygnału zezwalającego.
- Wykluczenie realizacji manewrów dwóch lub większej liczby pojazdów, których drogi przejazdu krzyżują się lub pokrywają na pewnych fragmentach.
- Kontrola nad informacjami przesyłanymi do pojazdów trakcyjnych oraz nadzór nad ich odpowiednim wykorzystaniem.
- Zwolnienie urządzeń ochrony bocznej, uniemożliwiających wjazd innym pociągom, w momencie zajęcia przez pojazd pozycji bezpiecznej i niestwarzającej ryzyka kolizji.
- Zapewnienie określonego odcinka drogi ochronnej, w przypadku nieprzewidzianych trudności przy hamowaniu przed sygnalizatorem zabraniającym jazdy.

Spełnienie powyższych wymogów może być zrealizowane w pełni za pomocą odpowiedniego systemu sterowania i podlegających mu urządzeń. W zależności od stopnia złożoności układu SRK można zakwalifikować go do jednej z czterech klas bezpieczeństwa ruchu, poczynwszy od pierwszej definiującej najbardziej podstawowe aspekty ochrony przed kolizjami, do czwartej obejmującej najbardziej kompleksowe układy sterowania, które zapewniają najwyższy stopień bezpieczeństwa. Klasy te nie odnoszą się natomiast do sytuacji, w których ruch kolejowy koliduje z ruchem pojazdów samochodowych.

Każdy system sterowania ruchem realizuje pewne zadania, które można uszeregować i przyporządkować do jednego z następujących etapów [4]:

a) Przygotowanie drogi przejazdu pociągu

System odbiera informację o potrzebie przygotowania drogi przejazdu dla pojazdu szynowego na pewnym odcinku, który znajduje się pod jego jurysdykcją. Następnie określa rodzaj zadania ruchowego np. przejazd bez zatrzymania, zatrzymanie na stacji lub wyprawienie pociągu. Po zdefiniowaniu zadania i wyznaczeniu warunków brzegowych, układ SRK przystępuje do realizacji poniższych zadań:

- Analiza istniejącej sytuacji ruchowej oraz zaplanowanie bezkolizyjnej drogi przejazdu.
- Sprawdzenie stanu i gotowości urządzeń do pracy.
- Zawiadomienie sąsiedniego okręgu o potrzebie przygotowania kolejnego zadania ruchowego.
- Przesłanie sygnałów sterujących do zwrotnic, nastawienie drogi przejazdu oraz odebranie informacji zwrotnych.
- Unieruchomienie urządzeń w położeniu zgodnym z ustaloną drogą przejazdu.
- Podanie do semafora informacji zezwalającej na jazdę.
- Określenie dozwolonej prędkości pociągu oraz przesłanie pozwolenia na start.

b) Dozorowanie jazdy pociągu

System na bieżąco kontroluje stan pojazdu, wliczając w to jego prędkość, pozycję, zesprzęglenie wagonów oraz stan techniczny poszczególnych części. Ponadto nadzoruje pozycję zwrotnic oraz utrzymuje blokadę zajętości torów.

c) Zwolnienie nastaw urządzeń

W ramach procesu zwalniania drogi przejazdu układ SRK wykonuje następujące czynności:

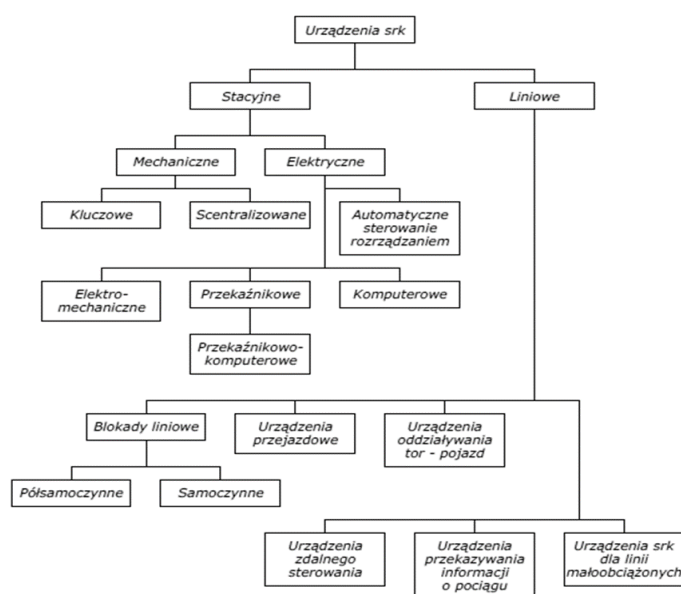
- Podawanie do semafora sygnału 'Stop' przez cały czas, kiedy droga przejazdu pozostaje zajęta przez pojazd.
- Określenie momentu, w którym pociąg opuścił miejsce kontrolne dla zwolnienia blokady przejazdu (opisana sytuacja ma miejsce zazwyczaj po zajęciu przez pojazd toru za ostatnią zwrotnicą planowanej trasy).
- Kontrolowanie właściwej kolejności zajmowanych przez pociąg odcinków trasy.
- Zwalnianie zwrotnic oraz innych urządzeń, które brały udział w wytyczeniu drogi przejazdu.

d) Rejestrowanie jazdy

Układ SRK rejestruje przebieg i czas przejazdu, a następnie przekazuje odpowiednie informacje do kolejnego okręgu i dyspozytury.

Systemy sterowania ruchem kolejowym podzielone są ze względu na różne obszary zastosowań i ograniczają się do sterowania urządzeniami tylko w obrębie wytyczonego okręgu. Najczęściej na pojedynczy okręg sterowania przypada jedna stacja, pomniejszy sektor torowiska należący do większej stacji lub prosty odcinek linii kolejowej z kilkoma posterunkami. Ze względu na odmienną charakterystykę sterowania takimi obiektami, można rozróżnić następujące typy urządzeń [5]:

- Urządzenia stacyjne – wykorzystywane są do zabezpieczenia i kontrolowania ruchu pociągów w obrębie posterunków zapowiadawczych (stacje oraz posterunki odgałęźne). Ich najważniejszą funkcją jest wzajemne uzależnienie sygnalizatorów, zwrotnic, wykołnic oraz innych urządzeń podczas ustawiania drogi przejazdu. Ponadto urządzenia stacyjne kontrolują stan i położenie elementów układu SRK, blokując sygnał zezwalający na jazdę w przypadku jakichkolwiek nieprawidłowości.
- Urządzenia liniowe – występują przede wszystkim na szlakach kolejowych oraz odcinkach łączących stacje i służą do regulacji ruchu pomiędzy posterunkami zapowiadawczymi i odstępowymi. Jedno z najważniejszych urządzeń liniowych stanowi blokada liniowa, która ma za zadanie regulować następstwo ruchu pociągów wyprawianych na szlak. Odstępy blokowe pomiędzy wyjeżdżającymi pociągami ustalane są na podstawie ich prędkości w taki sposób, aby w razie potrzeby drugi pociąg mógł się zatrzymać w bezpiecznej odległości za pierwszym.



Rysunek 2.1 Klasyfikacja urządzeń SRK pod względem funkcjonalności [5]

2.2. Systemy sterowania w modelarstwie kolejowym

2.2.1. Sterowanie analogowe

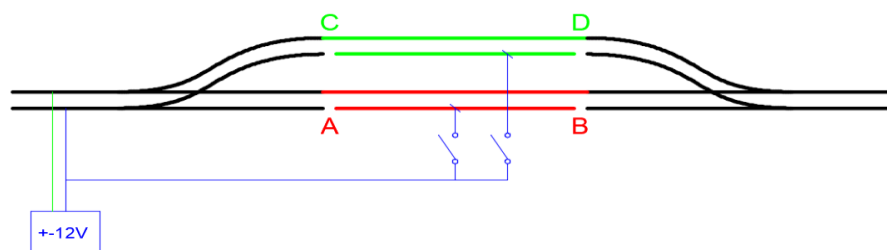
Pierwsze kolejki elektryczne, które pojawiły na rynku komercyjnym w 1890 roku, sterowane były poprzez zmianę napięcia lub natężenia prądu zasilającego. Wraz z rozwojem techniki układy zasilania wzbogacono o transformatory oraz mechanizmy pozwalające na zmianę kierunku jazdy lub przejście na bieg jałowy. Dzięki wprowadzeniu prostowników uzyskano także pierwsze modele zasilane na prąd stały, które w niedługim czasie zdominowały rynek modelarski.

Sterowanie analogowe kolejkami elektrycznymi można rozróżnić ze względu na konstrukcję torowiska, która może być jednolita lub blokowa. W pierwszym przypadku jedno źródło zasilania jest podłączone, poprzez odpowiedni układ prostowniczo-nastawny, do całego torowiska. Zmiana wartości napięcia zasilającego wpływa przez to jednakowo na cały układ. Niemożliwe jest wtedy osobne sterowanie prędkościami różnych lokomotyw. Inne elementy wchodzące w skład infrastruktury, takie jak natężenie światła sygnalizatorów, również zależą proporcjonalnie od wartości ustawionej w nastawniku.

W przypadku struktury blokowej, torowisko podzielone jest na kilka izolowanych odcinków, na które można dostarczać niezależne od siebie napięcie trakcyjne. Izolowany odcinek, zwany sekcją, otrzymuje się poprzez przerwanie fragmentu szyny w dwóch punktach. Operacji tej dokonuje się tylko na szynie z wyższym potencjałem, natomiast druga szyna stanowiąca masę jest wspólna dla wszystkich sekcji. w przypadku rozbudowanego torowiska o wielu pętlach i wielu źródłach zasilania konieczne może być izolowanie obydwu szyn. Dzięki podzieleniu układu na sekcje możliwe jest niezależne sterowanie prędkościami lokomotyw znajdujących się w różnych strefach torowiska.

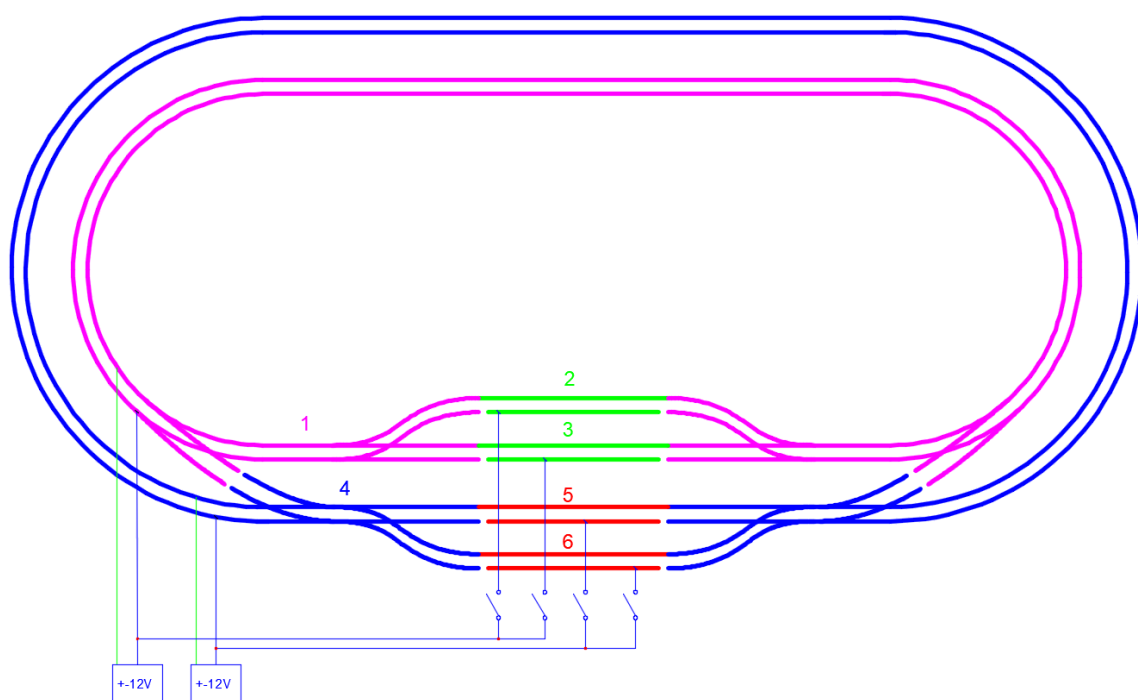
Obwody torowe bazujące na strukturze blokowej można podzielić na kilka kategorii w zależności od kształtu torowiska i wybranej metody sterowania [6]:

a) Obwody odcinkowe



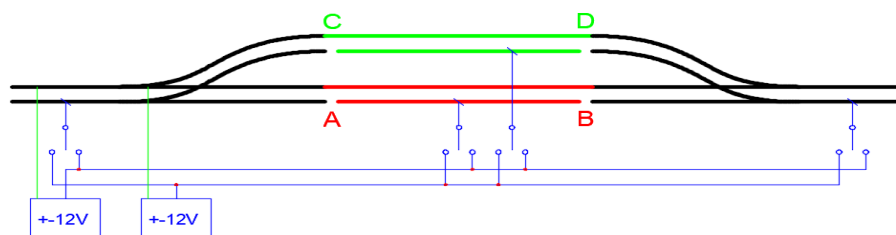
Rysunek 2.2 Obwód odcinkowy

b) Obwody grupowe



Stosuje się je w przypadku bardziej rozbudowanych torowisk, na które zazwyczaj składają się co najmniej dwie zamknięte pętle. Projektowanie tego rodzaju obwodów polega na połączeniu wybranych sekcji w minimum dwie grupy, przy czym każda z nich posiada oddzielne źródło zasilania. Na rysunku 2.3 przedstawiono przykładowy model obwodu grupowego, na który składa się linia dwutorowa tworząca jedną pętlę wewnątrz drugiej. Dopóki jedna z lokomotyw porusza się w grupie sekcji (1, 2, 3), a druga w grupie (4, 5, 6) możliwe jest jednocześnie i niezależne sterowanie ich prędkościami. W przypadku, kiedy ma nastąpić przejazd pociągu z jednej grupy do drugiej konieczne jest ustawienie takiej samej wartości napięcia w obu źródłach. Wadą takiego rozwiązania jest możliwość wystąpienia nagłych zrywów lokomotywy w trakcie przejazdu do innej grupy.

c) Obwody pociągowe

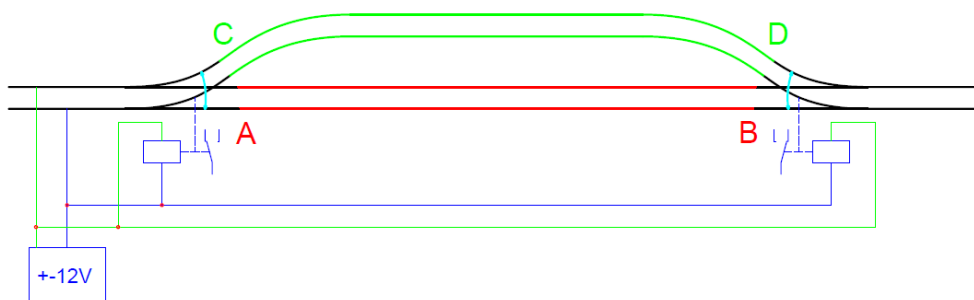


Rysunek 2.4 Obwody pociągowe

Obwody tego typu pozwalają na sterowanie danym pociągiem ciągle z tego samego źródła napięcia. Każda sekcja posiada wielopozycyjny przełącznik, na którego wejścia doprowadzone jest zasilanie z każdego źródła w układzie. Jeżeli lokomotywa przejeżdża z jednej sekcji do następnej to następuje przestawienie przełącznika na odbiór zasilania ze źródła odpowiadającego danej lokomotywie (Rysunek 2.4). Pozwala to sterować wieloma pociągami jednocześnie i niezależnie, eliminując przy tym ograniczenia występujące w obwodach grupowych. Obwody pociągowe są najczęściej stosowane w przypadku rozbudowanych makiet z wieloma lokomotywami.

d) Obwody zwrotnicowe

Sekcje torowe wydzielone są bezpośrednio za pomocą zwrotnic bez konieczności tworzenia specjalnie izolowanych odcinków. Zwrotnice posiadają wewnętrznie wbudowane przełączniki, które zmieniają ich ustawienie w zależności od polaryzacji źródła zasilania. Utworzenie nowej sekcji torowej wymaga zastosowania co najmniej dwóch zwrotnic albo w bardziej złożonych przypadkach, ich parzystej wielokrotności. Sterowanie przełącznikami powinno być skojarzone, tzn. odpowiednie pary powinny zmieniać swoje położenie w tym samym czasie, aby zachować ciągłość zasilania w torowisku. Na rysunku 2.5 przedstawiono najprostszy układ wykorzystujący obwody zwrotnicowe. Wykorzystano pojedyncze źródło zasilania, z którego doprowadzono napięcie zarówno na szyny torowiska, jak i przełączniki zwrotnic. W momencie zmiany polaryzacji źródła zasilania, następuje przestawienie widełek zwrotnicy i zasilenie odcinka |CD| przy jednoczesnym odcięciu sekcji |AB|. Zmiana biegunów źródła zasilania wpływa również na zmianę kierunku jazdy pociągów, dlatego obwody zwrotnicowe są najczęściej wykorzystywane w przypadku realizacji zadania mijania się pociągów jadących w różnych kierunkach. Wadą tego typu rozwiązania jest możliwość wystąpienia krótkookresowych braków zasilania w trakcie przełączania zwrotnic.



Rysunek 2.5 Obwody zwrotnicowe

e) Obwody sygnałowo-sprężone

Tego typu obwody bazują na odcinkach izolowanych, ale dodatkowo wyposażone są w elementy umożliwiające przetwarzanie sygnału oraz sterowanie. Najczęściej spotykane obwody sygnałowo-sprężone składają się z semaforów świetlnych, czujników zajętości toru oraz przekaźników. W momencie, kiedy jeden z pociągów wjeżdża na nowy, izolowany odcinek, zostaje aktywowany czujnik zajętości toru. Wykorzystywanych jest wiele rodzajów czujników, do których należą między innymi:

- Czujniki kontaktowe mechaniczne – sygnał wystawiany jest poprzez zamknięcie styku zwierne go za pomocą mechanicznej dźwigni. Przejeżdżający pociąg zahacza swoimi kołami o koniec dźwigni powodując jej zmianę położenia.
- Czujniki indukcyjne – zmiana pola elektromagnetycznego, wywołana pojawieniem się metalowego elementu przytwierdzonego do spodu kolejki, powoduje zwarcie styku i podanie sygnału zajętości
- Czujniki rezystancyjne – wyposażone są w wewnętrzny rezystor i mierzą wartość impedancji pomiędzy szynami. W momencie, kiedy do sekcji wjeżdża pociąg, wartość rezystancji wzrasta powodując zwarcie styku.
- Czujniki optyczne – emitują wiązkę światła podczerwonego, która w przypadku pojawienia się pociągu zostaje odbita od powierzchni podwozia i powraca do detektora.

Sygnał z czujnika zajętości toru trafia do semafora świetlnego oraz do przekaźników skojarzonych z sąsiednimi sekcjami. Jeżeli w danej sekcji znajduje się pociąg to semafor zapali się na czerwono, natomiast przekaźniki odetną zasilanie z odpowiednich sekcji, tak aby uniemożliwić wjazd innym pociągom i nie doprowadzić do kolizji. Wadą tego typu rozwiązania jest konieczność tworzenia sekcji o długości większej niż całkowita długość taboru kolejowego.

2.2.2. Sterowanie cyfrowe

Systemy analogowe sprawdzały się najlepiej w niewielkich i nieskomplikowanych aplikacjach modelarskich. Jednakże wraz ze wzrostem liczby pociągów oraz sekcji, układy sterowania stawały się coraz bardziej rozbudowane i trudniejsze w synchronizacji. W 1980 roku niemiecka firma Lenz Elektronik GmbH opracowała pierwszy cyfrowy system sterowania kolejek modelarskich. Przez kolejne lata projekt był rozwijany aż w 1993 roku z inicjatywy NMRA (*ang. National Model Railroad Association*) system został ustandaryzowany i przyjął nazwę DCC (*ang. Digital Command Control*). Standard ten definiuje podstawową strukturę komunikacji oraz magistralę wymiany danych pomiędzy nadajnikiem a dekodernem za pośrednictwem szyn kolejowych. Format danych przyjmuje formę impulsów w postaci fali prostokątnej i pozwala na współdziałanie sprzętu różnych firm.

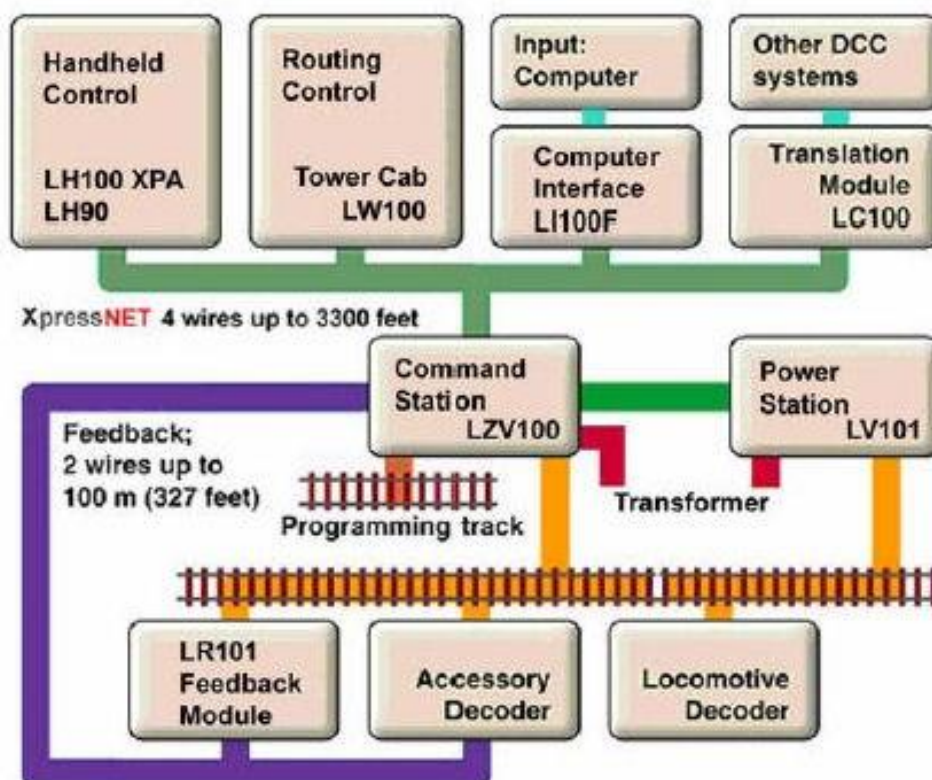
System cyfrowej komunikacji pozwala na indywidualne sterowanie wieloma pociągami bez konieczności dzielenia torów w izolowane sekcje. Ponadto wszelkie urządzenia peryferyjne takie jak semafony, zwrotnice czy rogatki mogą być zasilane i sterowane z tego samego źródła sygnału. W przeciwieństwie do systemów analogowych, układ sterowania DCC nie wymaga na ogół żadnych rozszerzeń w przypadku rozbudowy aplikacji modelarskiej.

Elementy wchodzące w skład systemu DCC [7]:

- Centrala DDC – urządzenie dedykowane lub komputer PC, które generuje pakiety danych, a następnie rozsyła do urządzeń wykonawczych zwanych dekodernami. Powstałe sygnały małej mocy są z reguły zgodne z poziomami TTL i CMOS, a ich kodowanie podlega standardom NMRA S-9.1 i S-9.2.
- Wzmacniacz (*ang. Booster*) – urządzenie, które wzmacnia sygnał generowany przez centralę i dopasowuje go do standardu DCC. Wzmacniacz powinien być dobrany odpowiednio do sumarycznej mocy wszystkich zainstalowanych odbiorników w układzie. W przypadku bardzo rozbudowanych aplikacji o dużej liczbie odbiorników możliwe jest podzielenie układu na kilka bloków i zasilanie każdego z nich poprzez osobny wzmacniacz. Sygnał generowany przez centralę powinien być doprowadzony równolegle do każdego boostera.
- Dekoder – odbiornik cyfrowy, który przetwarza otrzymany sygnał oraz przekierowuje zasilanie do elementu wykonawczego zgodnie z otrzymanymi instrukcjami. Każdy dekodek posiada osobny adres i wykonuje instrukcje przeznaczone tylko dla niego. Dekodery występujące w układach DCC możemy podzielić na trzy rodzaje:
 - a) dekodek lokomotywy – steruje kierunkiem i prędkością obrotów silnika lokomotywy oraz dodatkowym osprzętem takim jak oświetlenie, generatory

dymu i sygnałów dźwiękowych. Prędkość silnika regulowana jest za pomocą modulacji szerokości impulsu PWM, co umożliwia płynny start oraz stabilną pracę nawet przy niskich prędkościach jazdy.

- b) dekodery stacjonarne akcesoriów – jest odpowiedzialny za kontrolowanie urządzeń peryferyjnych takich jak przełączniki, zwrotnice, semafony, oświetlenie itp.
- c) dekodery funkcyjne – dołączany jest zazwyczaj do innego dekodera i pozwala rozszerzyć jego funkcje na przykład o obsługę wentylatorów, złożonych efektów świetlnych, automatycznych sprzęgów itp.
- Kontroler (*ang. Throttle*) – urządzenie wyposażone w interfejs pozwalający użytkownikowi regulować prędkość i kierunek jazdy lokomotyw oraz wywoływać funkcje dodatkowe. Rozbudowane kontrolery posiadają własny moduł przełączników i przełączników oraz wyświetlacz umożliwiający podgląd parametrów sterowanych urządzeń.
- Zasilacz – układ przekształcający napięcie sieciowe na napięcie potrzebne do zasilenia centrali i pozostałych elementów układu.

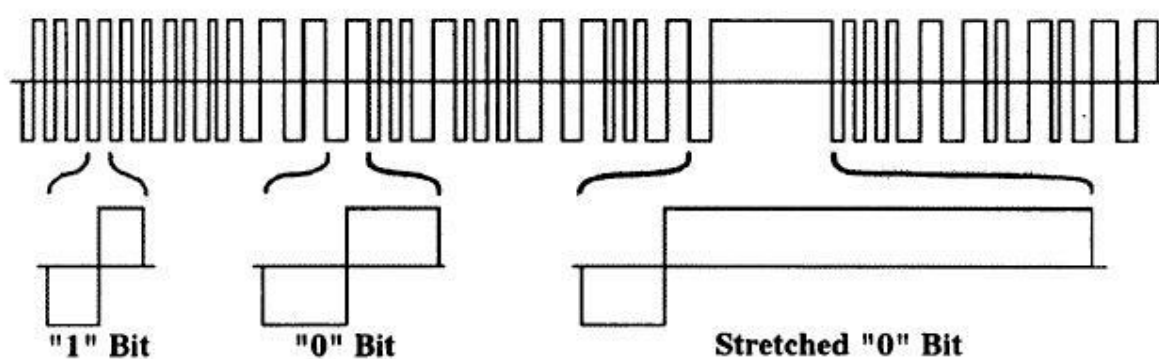


Rysunek 2.6 Przykładowy schemat systemu DCC [8]

Protokół komunikacyjny w systemach DCC:

W aplikacjach modelarskich zgodnych ze standardem NMRA S-9.1, odbiorniki zasilane są prądem przemiennym o stałej amplitudzie $\pm 16V$. Centrala sterująca przekształca sygnał o charakterze sinusoidalnym na falę prostokątną o różnych długościach impulsów. Zgodnie z powyższym standardem długości te powinny zawierać się w przedziale od 55 mikrosekund do 9900 mikrosekund. Na pojedynczy bit przypada kolejno następujący po sobie impuls ujemny oraz impuls dodatni. Można rozróżnić następujące wartości bitów [9]:

- Bit o wartości „1” – kodowany za pomocą dwóch impulsów, których długość mieści się w zakresie od 55 do 61 mikrosekund
- Bit o wartości „0” – kodowany za pomocą dwóch impulsów, których długość mieści się w zakresie od 95 do 120 mikrosekund
- Wydłużony bit o wartości „0” (*ang. Stretched „0” bit*) – składający się z dwóch impulsów, których długość mieści się w zakresie od 95 do 9900 mikrosekund. Jednakże długość całego bitu nie może przekroczyć 12000 mikrosekund.

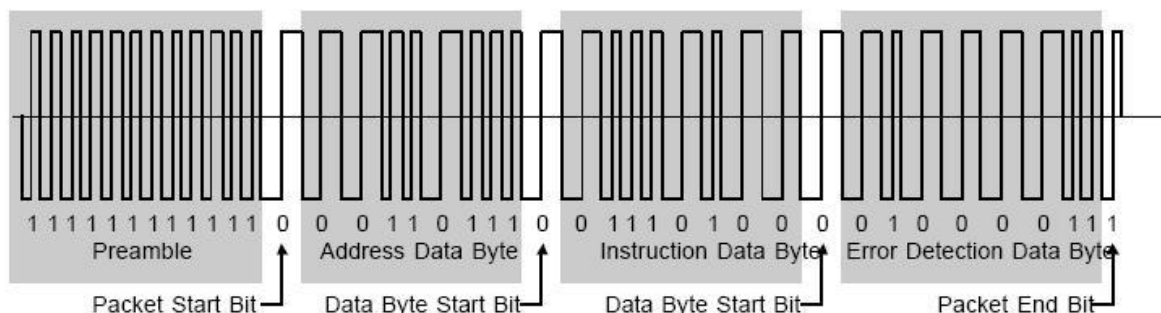


Rysunek 2.7 Kodowanie bitów z układach DCC [9]

Odpowiednia sekwencja bitów stanowi pakiet, którego struktura musi być zgodna z obowiązującym standardem, aby nawiązać połączenie pomiędzy urządzeniami. Wszelkiego rodzaju instrukcje, które nie są kodowane według zastosowanego protokołu, powinny być ignorowane przez urządzenia. Standard NMRA S-9.2 ściśle określa specyfikację pakietu DCC, który powinien składać się z od 3 do 6 bajtów. Każda porcja danych rozpoczyna się od preambuły, która wykorzystywana jest do synchronizacji nadajnika z odbiornikiem oraz niesie informację o nadchodzącym pakiecie. Kolejne bajty określają adres urządzenia, przesłane dane oraz sumę kontrolną. Poniżej przedstawiono elementy wchodzące w strukturę pakietu DCC:

| | |
|------------------------------|---|
| Preambuła | Preambuła składa się z szeregu bitów o wartości „1”. Centrala sygnałowa przesyła minimum 14 bitów w preambule, natomiast odbiornik (dekoder cyfrowy) wymaga do poprawnego odbioru więcej niż 12 bitów o wartości „1”, które poprzedzają bit startowy pakietu. |
| Bit startowy pakietu | Bit startowy pakietu to pierwszy bit o wartości „0” następujący po preambule. Stanowi informację, że kolejne odczytane bity będą należeć do bajtu adresu. |
| Bajt adresu | Bajt adresu składa się najczęściej z 8 bitów, przy czym pierwszy bit jest najstarszym bitem adresu. Adres nie może przyjmować samych wartości „0” lub samych wartości „1”, ponieważ są one zarezerwowane dla operacji specjalnych. |
| Bit startowy bajtu danych | Bit o wartości „0”, który poprzedza bajt danych. W pakiecie może wystąpić więcej niż jeden bajt danych i każdy z nich jest poprzedzony własnym bitem startowym. |
| Bajt danych | Bajt danych składa się z 8 bitów i może zawierać informację o adresie, instrukcję dla odbiornika lub innego rodzaju dane. |
| Bit startowy sumy kontrolnej | Bit poprzedzający bajt sumy kontrolnej i przyjmujący wartość „0” |
| Bajt sumy kontrolnej | Bajt sumy kontrolnej służy do detekcji błędów transmisji. Wartości bitów przypisywane są w momencie wysyłania pakietu przez centralę sygnałową, a obliczane są za pomocą operacji EX-OR zawartości bajtu danych oraz adresu. Dekoder cyfrowy odczytuje otrzymane dane, a następnie ponownie realizuje powyżej wymienioną operację. Otrzymany wynik porównywany jest z bajtem sumy kontrolnej. W przypadku niezgodności pakiet zostaje wysłany ponownie. |
| Bit końca pakietu | Ostatni bit, który przyjmuje wartość „1” i oznacza koniec pakietu. |

Na rysunku 2.8 przedstawiono przykład poprawnego pakietu zgodnego z semantyką Ogólnego Formatu Pakietu DCC, który określa standardową formę wymiany danych pomiędzy urządzeniami w cyfrowych układach modelarskich.



Rysunek 2.8 Przykładowy pakiet danych w systemach DCC [10]

W celu zapewnienia minimalnej funkcjonalności oraz współpracy pomiędzy różnymi systemami, NMRA zdecydowało się wprowadzić szablon instrukcji zwany Pakietami Standardowymi. Centrala sterująca odczytuje instrukcje operatora, a następnie koduje je zgodnie z semantyką Pakietów Podstawowych oraz przesyła do dekodera. Dekoder rozpoznając konkretne wartości bajtów danych i adresu wykonuje jedno z niżej wymienionych poleceń [10]:

a) Pakiet Prędkości i Kierunku Jazdy dla Dekoderów Lokomotyw

Zawiera instrukcje sterowania silnikiem lokomotywy, określając jego prędkość oraz kierunek obrotów. Kodowanie tego rodzaju pakietu odbywa się w następujący sposób:

| | | | | | | | |
|--------------|---|-------------|---|-------------|---|----------------|---|
| 111111111111 | 0 | 0AAAAAA | 0 | 01DCSSSS | 0 | EEEEEEEE | 1 |
| Preambuła | | Bajt adresu | | Bajt danych | | Suma kontrolna | |

Pierwszy bit adresu dekodera lokomotywy przyjmuje wartość „0”. Pozostałe bity oznaczone literami „A” mogą przyjmować wartości dowolne, natomiast dekodery powinny zapewniać możliwość ich prostej konfiguracji. W celu wysterowania każdej lokomotywy z osobna, ich adresy dekoderek powinny różnić się między sobą.

Bajt danych posiada zakodowaną instrukcję odnośnie prędkości i kierunku. Pierwsze dwa bity powinny przyjmować wartości kolejno „0” oraz „1”, tworząc sekwencję używaną do rozpoznania przez dekodery omawianego rodzaju Pakietu Podstawowego. Kolejny bit oznaczony literą „D” określa kierunek obrotów silnika lokomotywy. Jeżeli bit kierunku przyjmuje wartość „1” to lokomotywa powinna poruszać się do przodu, a jeżeli „0” to do tyłu. Bit opisany literą „C” domyślnie oznacza dodatkowy (i jednocześnie najbardziej znaczący) bit prędkości, ale może być również używany do sterowania oświetleniem czołowym. Ostatnie cztery bity określone literą „S” zawierają informację o prędkości, która zmienia się krokowo w zależności od konkretnych kombinacji przedstawionych w tabelce poniżej (Rysunek 2.9).

| CS ₃ S ₂ S ₁ S ₀ | Speed | CS ₃ S ₂ S ₁ S ₀ | Speed | CS ₃ S ₂ S ₁ S ₀ | Speed | CS ₃ S ₂ S ₁ S ₀ | Speed |
|--|-------------|--|---------|--|---------|--|---------|
| 00000 | Stop | 00100 | Step 5 | 01000 | Step 13 | 01100 | Step 21 |
| 10000 | Stop (I) | 10100 | Step 6 | 11000 | Step 14 | 11100 | Step 22 |
| 00001 | E-Stop* | 00101 | Step 7 | 01001 | Step 15 | 01101 | Step 23 |
| 10001 | E-Stop* (I) | 10101 | Step 8 | 11001 | Step 16 | 11101 | Step 24 |
| 00010 | Step 1 | 00110 | Step 9 | 01010 | Step 17 | 01110 | Step 25 |
| 10010 | Step 2 | 10110 | Step 10 | 11010 | Step 18 | 11110 | Step 26 |
| 00011 | Step 3 | 00111 | Step 11 | 01011 | Step 19 | 01111 | Step 27 |
| 10011 | Step 4 | 10111 | Step 12 | 11011 | Step 20 | 11111 | Step 28 |

Rysunek 2.9 Tabela kodowania prędkości lokomotywy [10]

* Dekoder cyfrowy natomiast odcina zasilanie silnika lokomotywy

(I) Bit kierunku może zostać zignorowany w przypadku funkcji wrażliwych na zmianę kierunku

b) Pakiet Reset do Wszystkich Dekoderów Cyfrowych

Zostaje odebrany przez wszystkie urządzenia podłączone do magistrali danych, niezależnie od przypisanych do nich adresów. Dekoder cyfrowy, który otrzyma omawiany pakiet kasuje swoją pamięć ulotną oraz powraca do stanu, w jakim się znajdował tuż po włączeniu zasilania. Centrala sterująca po przesłaniu Pakietu Reset nie powinna wysyłać żadnych innych pakietów o adresach z przedziału '01100100' do '01111111' przez 20 milisekund, chyba że planowane jest przejście w tryb serwisowy.

| | | | | | | | |
|--------------|---|-------------|---|-------------|---|----------------|---|
| 111111111111 | 0 | 00000000 | 0 | 00000000 | 0 | 00000000 | 1 |
| Preambuła | | Bajt adresu | | Bajt danych | | Suma kontrolna | |

c) Pakiet Pusty do Wszystkich Dekoderów Cyfrowych

Każdy dekodek cyfrowy, który otrzyma ten pakiet nie podejmie żadnych nowych działań, ale powróci do wcześniej wykonywanych zadań traktując ten pakiet tak, jakby został rozesłany do innych dekodek.

| | | | | | | | |
|--------------|---|-------------|---|-------------|---|----------------|---|
| 111111111111 | 0 | 11111111 | 0 | 00000000 | 0 | 11111111 | 1 |
| Preambuła | | Bajt adresu | | Bajt danych | | Suma kontrolna | |

d) Pakiet Ogłoszenie Stop do Wszystkich Dekoderów Cyfrowych

Po otrzymaniu tego rodzaju pakietu, w którym bit oznaczony literą „S” ma wartość „1”, wszystkie dekodery lokomotyw powinny odciąć zasilanie podawane do silników. Jeżeli bit opisany literą „C” przyjmuje wartość „1” to bit kierunku „D” może zostać zignorowany dla funkcji wrażliwych na zmianę kierunku.

| | | | | | | | |
|--------------|---|-------------|---|-------------|---|----------------|---|
| 111111111111 | 0 | 00000000 | 0 | 01DC000S | 0 | EEEEEEEE | 1 |
| Preambuła | | Bajt adresu | | Bajt danych | | Suma kontrolna | |

3. Cel i zakres pracy

Celem niniejszej pracy było opracowanie układu automatycznego sterowania ruchem kolejowym na makiecie modelarskiej. Budowa oraz zarządzanie makietą miały być wzorowane na standardach panujących w rzeczywistym ruchu kolejowym. Autor samodzielnie zaprojektował oraz wykonał układy elektroniczne, do których należała infrastruktura pomiarowo-sygnałowa, centrala nadzorująco-sterująca oraz urządzenia wykonawcze, odpowiedzialne między innymi za sterowanie prędkością lokomotyw. Następnie zbadano działanie i zachowanie systemu na podstawie opracowanej aplikacji w środowisku programistycznym Arduino. Na koniec przeanalizowano możliwość wprowadzenia zaprojektowanego systemu do rzeczywistych kolejowych układów SRK.

W toku studiów magisterskich autor niniejszej pracy poszerzył swoją wiedzę na temat programowania obiektowego oraz projektowania układów sterowania automatyki. Liczne zadania oraz wymagające projekty pozwoliły mu usystematyzować zebrane wiadomości, a następnie wykorzystać je w procesie tworzenia aplikacji. Autor również niejednokrotnie zetknął się z problematyką przetwarzania sygnałów w ruchu kolejowym, co w rezultacie zainspirowało go do napisania pracy związanej z tą dziedziną. Autonomiczne pojazdy kolejowe mogły zostać potraktowane w podobny sposób jak roboty mobilne, poruszające się po wcześniej wyznaczonych ścieżkach. Szeroki zakres rozwiązań technicznych oraz produktów z branży modelarstwa kolejowego pozwoliły mu skompletować zestaw, który następnie posłużył do zbudowania prostej makiety testowej. Autor zapoznał się z oferowanymi przez rynek układami sterowania, które są najczęściej stosowane w modelarstwie, jednakże zdecydował się na zaprojektowanie własnego systemu bazującego na komunikacji bezprzewodowej.

We wstępnym etapie przygotowań do rozpoczęcia pracy nad projektem autor dokonał przeglądu istniejących rozwiązań w dziedzinie komunikacji bezprzewodowej i zdecydował się na wybór modułów radiowych ze względu na ich niewielkie rozmiary oraz atrakcyjną cenę. Model wymiany informacji ostatecznie przyjął na zasadzie „master-slave”, gdzie główna jednostka nadzorująca przesyła zapytania do urządzeń wykonawczych, a następnie odczytuje wysłane przez nie odpowiedzi.

Kolejnym ważnym elementem do ustalenia był rodzaj czujników wykorzystywanych do określania obecności pociągu na danym odcinku torowiska. Autor również w tym aspekcie zdecydował się na niestandardowe rozwiązanie i wybrał czujniki odbiciowe ze względu na ich prosty i bezproblemowy montaż na makiecie. Pozostałe rodzaje, w tym mechaniczne dźwignie

lub przełączniki wymagałyby specjalnych przeróbek lub nawet ingerencji w strukturę torowiska.

Ostatnią i równie ważną decyzją był wybór układów logicznych, których zadaniem jest nadzorowanie ruchu pociągów oraz przetwarzanie i realizacja otrzymanych poleceń. Autor założył, że rolę centralnego systemu sterowania i dyspozytorni będzie pełnił moduł Arduino Leonardo wyposażony w mikrokontroler AVR Atmega32U4. Moduł ten jest najprostszym modelem, który posiada odpowiednią liczbę wejść i wyjść spełniających wymagania projektowe. Natomiast układy podwykonawcze, znajdujące się na poszczególnych pociągach, zarządzane będą poprzez osobne moduły Arduino Pro Mini. Ich najważniejszą zaletą są niewielkie rozmiary oraz obsługa interfejsu szeregowego UART.

Poniżej przedstawiona została szczegółowa lista założeń projektowych:

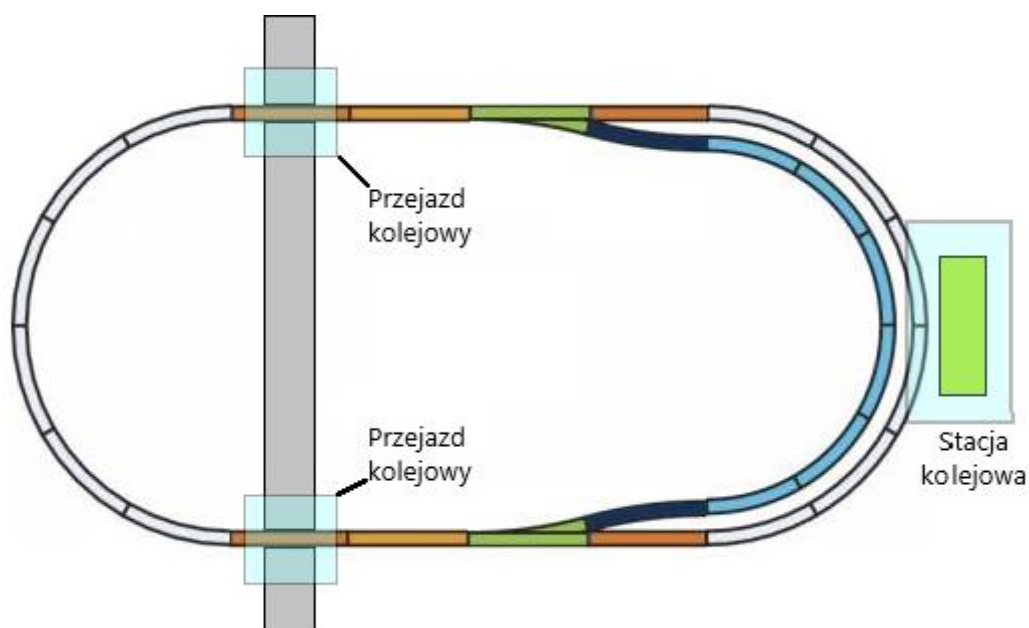
- Automatycznemu sterowaniu podlegać będą dwie lokomotywy poruszające się w przeciwnych kierunkach oraz zwrotnice umożliwiające wjazd na tor manewru wymijania.
- Ruch lokomotyw będzie odbywał się w sposób bezkolizyjny, tzn. niedopuszczalna jest sytuacja, kiedy dwa pociągi znajdą się w tej samej sekcji torowej.
- Lokomotywy będą wykorzystywać dodatkowy tor manewrowy, jeżeli zaistnieje konieczność wykonania operacji mijania.
- Pociągi będą zatrzymywać się w specjalnie wyznaczonych strefach, umownie przyjętych jako stacje oraz przejazdy kolejowe.
- Składy kolejowe będą rejestrować długość przejechanej drogi za pomocą enkoderów magnetycznych zamontowanych do osi kół.
- Algorytm zostanie opracowany w strukturze obiektowej, umożliwiając łatwe zmiany oraz możliwość rozszerzenia funkcjonalności.

4. Opis projektu makiety kolejowej

4.1. Torowisko PIKO A-Gleis







Do budowy torowiska, po którym poruszać się będą automatycznie sterowane lokomotywy, autor użył gotowych zestawów modelarskich PIKO A-Gleis Set A + C. Tory zostały wykonane w skali H0, co w modelarstwie kolejowym oznacza, że stosunek wymiarów repliki do rzeczywistych wymiarów danego elementu wynosi 1:87. Szyny zakończone są specjalnie wykończonymi łącznikami, które gwarantują ciasne pasowanie pomiędzy kolejnymi fragmentami torów. Wykonane zostały z niklowanego srebra, co zapewnia dobrą przewodność elektryczną, a w konsekwencji umożliwia wykorzystanie ich jako elementu pośredniczącego pomiędzy źródłem zasilania, a odbiornikiem. Podkłady torowe wyprodukowano z tworzywa sztucznego ABS, które cechuje się wysoką udarnością, dobrym tłumieniem dźwięku oraz odpornością na pęknięcia sprężynowe.

Przed właściwym montażem torowiska, autor pracy zdecydował się wykonać planszę o wymiarach 105 x 200 cm, która będzie pełnić rolę fundamentu makiety. Skonstruowana została z płyty MDF o grubości 3 mm, a następnie pokryta zieloną tkaniną imitującą trawę i roślinność. W kolejnym etapie zmontowane zostało torowisko PIKO o kształcie przedstawionym na rysunku 4.1.



Rysunek 4.1 Model torowiska PIKO

Kolorem błękitnym zostały wyróżnione umowne strefy przejazdowe i stacyjne, które mają odzwierciedlać rzeczywistą infrastrukturę kolejową. Poszczególne odcinki torowe w zależności od długości i kształtu zostały oznaczone różnymi kolorami według poniższej tabeli:

| Oznaczenie | Kolor i kształt | Długość | Opis |
|------------|---|---|----------------------|
| G231 |  | 231 mm | Tor prosty |
| G239 |  | 239 mm | Tor prosty |
| WLWR |  | 239 mm | Zwrotnica lewa/prawa |
| R9 |  | $\alpha = 15^\circ, R = 908 \text{ mm}$ | Tor łukowy |
| R2 |  | $\alpha = 30^\circ, R = 422 \text{ mm}$ | Tor łukowy |
| R1 |  | $\alpha = 30^\circ, R = 360 \text{ mm}$ | Tor łukowy |

Gotowe torowisko o wymiarach 85 x 180 cm zostało przytwierdzone do planszy za pomocą gumek modelarskich, przewleczonych przez specjalnie nawiercone otwory. Następnie poszczególne fragmenty torowe zostały dodatkowo powiązane, aby zagwarantować spójność linii szynowych oraz zabezpieczyć je przed ewentualnymi przesunięciami na skutek bodźców zewnętrznych.

4.2. Zasilacz i regulator napięcia

Autor zdecydował, że do zasilania silników lokomotyw oraz układów elektronicznych odpowiedzialnych za sterowanie ich prędkością, wykorzysta dołączony do zestawu zasilacz oraz regulator prędkości. Zasilacz wyposażony jest w wewnętrzny transformator, który obniża napięcie wejściowe 230 V AC do wartości 12 V AC pojawiającej się na złączu wyjściowym. Ponadto posiada układ zabezpieczeń, który określa górną granicę natężenia prądu na poziomie 450mA. W przypadku podłączenia zbyt dużego obciążenia lub wystąpienia zwarcia, następuje rozłączenie obwodu wyjściowego. Ponowne przejście do trybu pracy wymaga usunięcia źródła usterki oraz chwilowego odłączenia napięcia wejściowego.

Regulator prędkości PIKO składa się przede wszystkim z mostka prostowniczego oraz układu regulacji napięcia wyjściowego. Za pomocą zewnętrznego pokrętła możliwe jest ręczne sterowanie wartością napięcia w zakresie od 0 do 12 V DC. Autor założył, że regulator PIKO będzie służył jedynie jako element prostowniczy oraz wyłącznik główny ON/OFF, natomiast sterowaniem prędkością lokomotywy będzie się zajmował oddzielny układ elektroniczny [11].

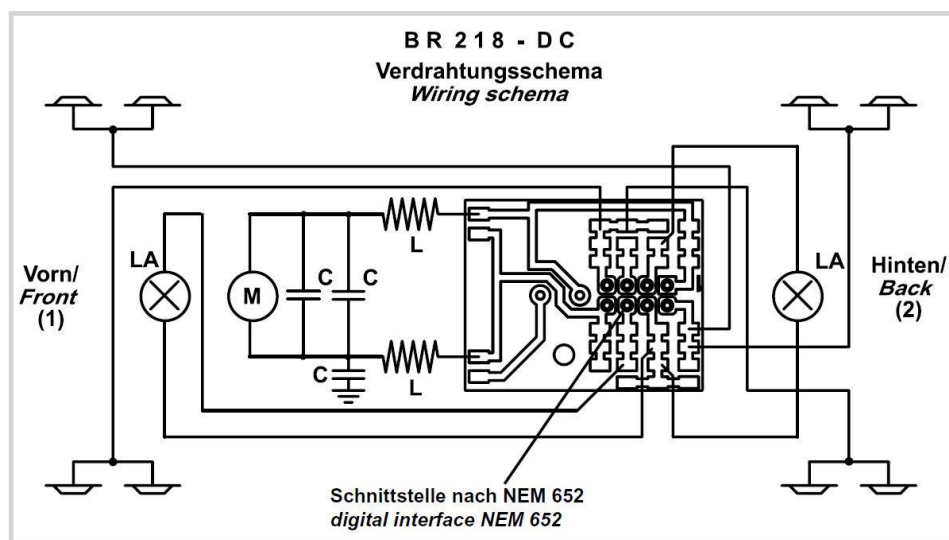
4.3. Lokomotywy PIKO BR 218

W założeniach projektu określono, że po torowisku poruszać się będą dwa składy pociągowe, jeżdżące w przeciwnych kierunkach. Zdecydowano się na zakup dwóch lokomotyw PIKO BR 218 DB IV, które oferują wymaganą funkcjonalność w atrakcyjnej cenie. Wspomniany model jest wierną kopią lokomotywy Baureihe 218, która produkowana była w latach 1963 – 1979 w zachodnich Niemczech. Napędzana była przez silnik diesla o mocy 1900kW i osiągała prędkość rzędu 140km/h. Powstało jedynie 398 egzemplarzy lokomotyw z serii 218, ponieważ wraz z rozwojem kolejnictwa i rosnącą popularnością systemów DMU (*ang. Diesel Muiltiple Unit*) spadło zapotrzebowanie na tego rodzaju lokomotywy.

Wykorzystane w projekcie modele PIKO BR 218 przystosowane zostały do poboru energii elektrycznej z szyn torowiska za pośrednictwem specjalnie zaprojektowanych osi kołowych. Zdecydowano, że wspomniany mechanizm zostanie zmodyfikowany oraz przystosowany do zasilania zarówno silnika lokomotywy jak i układów elektronicznych odpowiedzialnych za sterowanie. w tym celu podzielono istniejący obwód (Rysunek 4.2) na dwie części [12]:

- a) Układ doprowadzający prąd elektryczny, który składa się z mechanizmu łożyskowanych osi kołowych oraz ślizgających się po nich szczotek.
- b) Układ stabilizujący pracę silnika złożony z kondensatorów i cewek oraz obwody świetlne.

Wolne końce stanowiące wejście i wyjście każdego z nowopowstałych obwodów zostały wyprowadzone poza obrys lokomotywy, a następnie połączone z układem sterowania znajdującym się w wagonie.



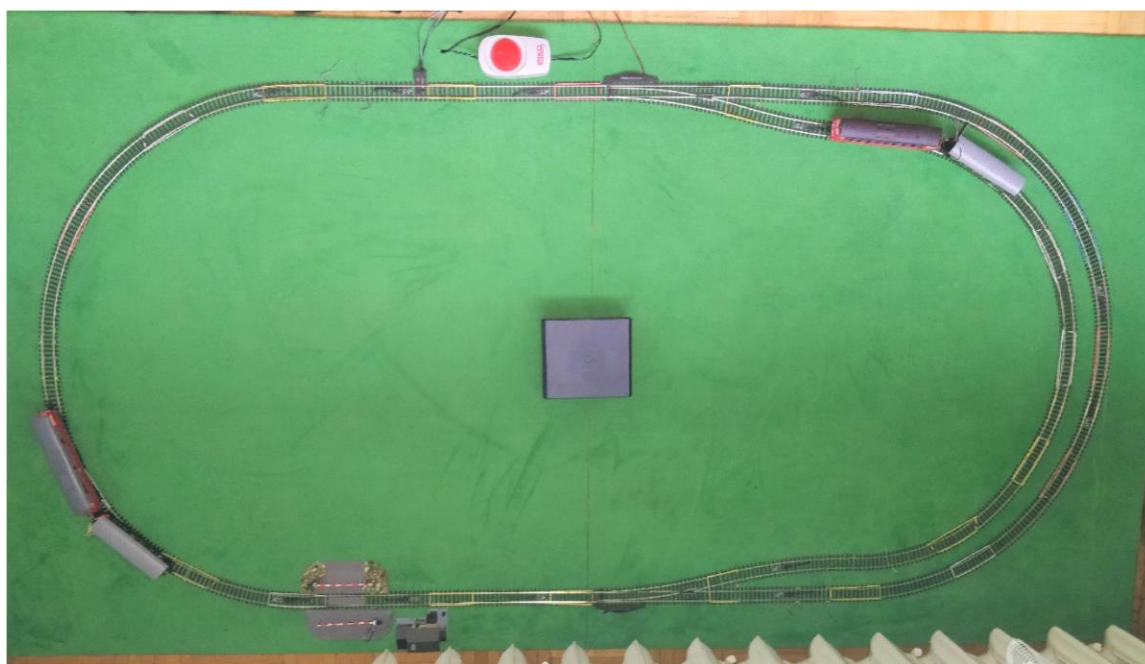
Rysunek 4.2 Schemat elektryczny lokomotywy PIKO BR 218 [12]

4.4. Podsumowanie

Skonstruowanie makiety było pierwszym etapem na drodze realizacji projektu automatycznego sterowania ruchem kolejowym. Autor pracy określił założenia oraz wymagania jakie powinien spełniać układ, a następnie skompletował odpowiednie części i urządzenia. W kolejnym kroku skonstruował planszę pełniącą rolę fundamentu, ułożył torowisko oraz odpowiednio je przymocował i zabezpieczył. Na gotowej makiecie wyznaczył strefy stacyjne i przejazdu, co w dalszej części projektu będzie niezbędne do określenia dokładnej liczby i umiejscowienia czujników zajętości toru.

Autor zaopatrzył się również w dwie lokomotywy z wagonami, które według założeń będą poruszać się po torowisku w przeciwnych kierunkach. Planując ich przystosowanie do późniejszej współpracy z układem sterowania, zmodernizował istniejące obwody zasilania silników. Podłączył również zasilacz i regulator, oraz sprawdził poprawność działania całej makiety.

W ostatnim etapie autor wyznaczył punkt, w którym znajdować się będzie centralny system sterowania i dyspozytornia. Optymalnym miejscem okazał się środek makiety, ze względu na możliwie najkrótsze trasy kablowe oraz odległości pomiędzy nadajnikiem i odbiornikiem radiowym. Autor zainstalował w tym miejscu czarną obudowę na elektronikę o wymiarach 129 x 159 x 50 mm, a następnie przygotował otwory i przepusty na przewody. Ostateczna wersja makiety wraz z wprowadzonymi później poprawkami przedstawiona została na rysunku 4.3.



Rysunek 4.3 Zdjęcie makiety wykorzystanej w projekcie

5. Opis układu sterowania i komunikacji

Przed przystąpieniem do projektowania układu sterowania, zweryfikowano ponownie założenia projektowe. Na podstawie wyznaczonych stref postojowych określono, że wymaganych jest minimum dwanaście czujników zajętości toru do uzyskania pożądanej funkcjonalności. Ponadto wynikła konieczność pozyskiwania informacji o tym, w którym momencie nastąpiła zmiana stanu w odczytach z czujników. Stwierdzono, że optymalnym rozwiązaniem problemu będzie zastosowanie modułów rozszerzających liczbę wejść i posiadających możliwość obsługi przerwań.

Napotkano również trudności w sterowaniu zwrotnicami, których napięcie znamionowe cewek według dokumentacji wynosiło 12 – 16V DC. Napięcie na wyjściach mikrokontrolera osiągało natomiast jedynie 5V DC i było niewystarczające do poprawnej pracy zwrotnic. Zdecydowano, że w tym przypadku zastosowane zostaną przekaźniki NO/NC, na których styki podane zostanie napięcie bezpośrednio z regulatora PIKO.

Po zakończonej weryfikacji oraz wprowadzeniu poprawek w schematach ideowych, zamówiono odpowiednie części i rozpoczęto budowę układu sterowania.

5.1. Zestawienie elementów i urządzeń wchodzących w skład systemu automatycznego sterowania

Poniżej przedstawiono tabelę zawierającą dokładne zestawienie elementów użytych w projekcie:

| Nr | Opis | Model | Producent | Ilość szt. |
|----|---------------------------|------------------|----------------------|------------|
| 1 | Mikrokontroler | Leonardo A000057 | Arduino | 1 |
| 2 | Mikrokontroler | Pro Mini 328 5V | Arduino | 2 |
| 3 | Moduł radiowy 3,3V | nRF24L01 | Nordic Semiconductor | 3 |
| 4 | Stabilizator liniowy 5V | L7805CV | STMicroelectronics | 2 |
| 5 | Stabilizator liniowy 3,3V | LM1117T | STMicroelectronics | 2 |
| 6 | Mostek H 13,5V | TB6612FNG | Pololu | 2 |
| 7 | Czujnik odbiciowy | KTIR0711S | Kingbright | 12 |

| | | | | |
|----|---------------------|----------------|-----------|---|
| 8 | Ekspander wejść | PCF8574 | Waveshare | 2 |
| 9 | Enkoder 5V | 3081 | Pololu | 2 |
| 10 | Akumulator Li-Pol | XP17002EX | Dualsky | 1 |
| 11 | Moduł przekaźnikowy | SRD-05VDC-SL-C | Songle | 1 |

5.1.1. Płytki modułowa Arduino Leonardo

Główną jednostkę obliczeniowo-sterującą w projekcie stanowi Arduino Leonardo z mikrokontrolerem AVR Atmega320U. Płytki posiada zainstalowany program rozruchowy (*ang. bootloader*), który ułatwia nawiązanie komunikacji z komputerem oraz programowanie mikrokontrolera. Wspomniana cecha znacząco ułatwiła i przyspieszyła tworzenie algorytmu, który mógł być na bieżąco testowany i poprawiany. Arduino Leonardo posiada łącznie 20 cyfrowych wejść/wyjść, z których 7 posiada możliwość modulacji sygnału PWM, a kolejnych 12 można wykorzystać jako wejścia analogowe.

W ramach projektu wykorzystano jedynie siedem portów I/O, jednakże pierwotne zapotrzebowanie było większe. Redukcja aktywnych portów wejść i wyjść wynika natomiast z konieczności zastosowania modułów rozszerzających. Płytki obsługuje ponadto interfejsy komunikacyjne I2C, SPI i USB, które wykorzystane zostały kolejno w celu nawiązania łączności z modułami ekspanderów, modułem radiowym oraz komputerem.



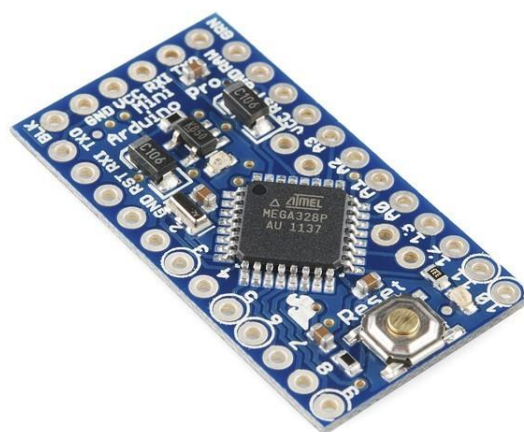
Źródło: botland.pl

Rysunek 5.1 Arduino Leonardo

5.1.2. Płytki modułowa Arduino Pro Mini 328

Do przetwarzania odebranych informacji oraz sterowania lokomotywą wykorzystano moduły Arduino Pro Mini (Rysunek 5.2) wyposażone w mikrokontroler Atmega328. Płytki podobnie jak Arduino Leonardo posiada zainstalowany program rozruchowy, jednakże ze względu na oszczędność miejsca nie została wzbogacona w interfejs USB. W celu zaprogramowania płytki autor musiał zatem skorzystać z interfejsu UART i zaopatrzył się w konwerter USB-UART wykorzystujący układ FT232RL.

Moduł Arduino Pro Mini cechuje się przede wszystkim niewielkimi wymiarami (33 x 18 mm) oraz niskim profilem, dzięki czemu mógł zostać bezproblemowo zamontowany w wagonie kolejowym o wewnętrznej szerokości 25 mm. Ponadto posiada 14 cyfrowych wejść/wyjść, z których 6 można wykorzystać jako wyjścia PWM oraz 6 wejść analogowych. W ramach projektu wykorzystanych zostało 11 portów I/O, które posłużyły do komunikacji z modułem radiowym, mostkiem H oraz enkoderem.

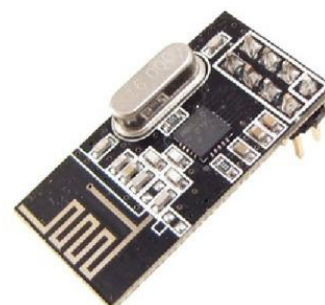


Źródło: botland.pl

Rysunek 5.2 Arduino Pro Mini

5.1.3. Moduł radiowy nRF24L01

Podstawowy element w projekcie zapewniający komunikację pomiędzy centralą dyspozytorni, a lokomotywami stanowi moduł radiowy nRF24L01. Każdy z nich może działać zarówno jako nadajnik jak i odbiornik, co umożliwia pełną wymianę informacji, na którą składa się dodatkowo funkcja wysyłania odpowiedzi zwrotnej, raportowania błędów lub ponownego wysyłania w przypadku niepowodzenia. Moduł nRF24L01 działa w paśmie fal radiowych o częstotliwości 2,4 – 2,525 GHz, które z kolei dzieli się na 125 kanałów i pozwala stworzyć właśnie tyle osobnych sieci tych urządzeń. Maksymalna prędkość wymiany danych z jaką może działać moduł to 2 Mb/s. Modem posiada ponadto możliwość obsługi przerwań, które mogą być użyte np. do wybudzania Arduino w momencie otrzymania danych.



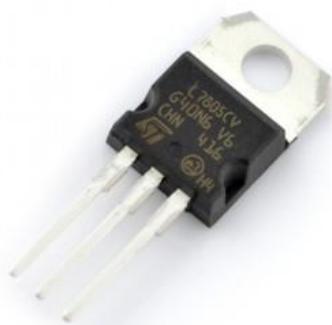
Źródło: botland.pl

Rysunek 5.3 Moduł radiowy

5.1.4. Stabilizatory liniowe L7805CV oraz LM1117T

Napięcie zasilania otrzymywane z regulatora PIKO wynosi 12 V DC i jest zbyt wysokie, aby zasilać układy elektroniczne odpowiedzialne za komunikację i sterowanie lokomotywą. z tego powodu autor zdecydował się wykorzystać stabilizatory liniowe L7805CV oraz LM1117T., których zadaniem jest obniżenie napięcia wejściowego do zadanej, stabilnej wartości otrzymywanej na wyjściu. Moduł Arduino Pro Mini, mostek TB6612FNG oraz enkoder

3081 zasilane są napięciem 5V DC za pośrednictwem stabilizatora L7805CV, natomiast moduł radiowy nRF24L01 zasilany jest napięciem 3,3V DC ze stabilizatora LM1117T.

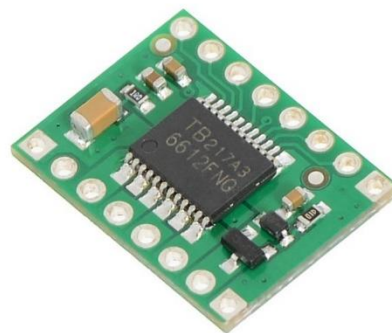


Źródło: botland.pl

Rysunek 5.4 Stabilizator liniowy

5.1.5. Dwukanałowy sterownik silników TB6612FNG

Sterowniki silników producenta Pololu wykorzystują układy mostka H o oznaczeniu TB6612FNG. Urządzenie to składa się z czterech tranzystorów unipolarnych LD MOS, które zmieniają kierunek przepływu prądu przez odbiornik. Częstotliwość przełączania tranzystorów pomiędzy stanem zatkania a przewodzenia regulowana jest za pomocą modulacji PWM. W zależności od stopnia wypełnienia sygnału sterującego możliwe jest uzyskiwanie różnych prędkości obrotu wirnika. Moduł TB6612FNG został zaprojektowany w celu sterowania dwoma silnikami, jednakże odpowiednie połączenie kanałów pozwala zwiększyć wydajność prądową i wykorzystać układ do pracy z mocniejszym silnikiem. Autor zastosował wspomnianą funkcjonalność i wykorzystał pojedynczy moduł mostkowy do sterowania tylko jednym silnikiem znajdującym się w każdej z lokomotyw.



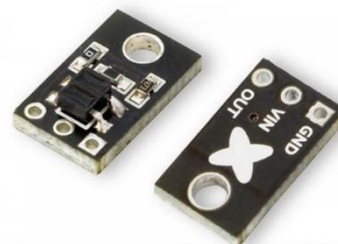
Źródło: botland.pl

Rysunek 5.5 Sterownik silnikowy

5.1.6. Czujnik odbiciowy KTIR0711S

Moduł czujnika odbiciowego zbudowany jest z nadajnika promieni podczerwonych oraz fototranzystora. Wiązka promieniowania wysyłana przez nadajnik odbija się od powierzchni roboczej, a następnie wraca z powrotem do czujnika i mierzona jest za pomocą detektora. Natężenie światła odbitego zależy w bardzo dużym stopniu od rodzaju, barwy oraz odległości w jakiej znajduje się powierzchnia odbijająca. Według założeń projektowych, wiązka promieniowania emitowana przez czujnik w normalnych warunkach pracy będzie całkowicie

rozpraszana, ponieważ na jej drodze nie będzie znajdować się żadna powierzchnia odbijająca. Czujnik powinien wystawić sygnał obecności dopiero w momencie, kiedy nad detektorem znajdzie się podwozie lokomotywy. W celu zapewnienia jak najlepszej sprawności powierzchni odbijającej, spód lokomotywy został pomalowany białą farbą o niskim stopniu pochłaniania światła.

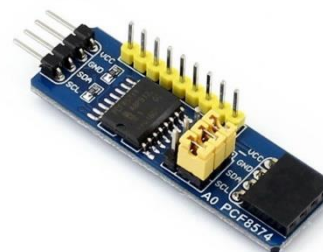


Źródło: botland.pl

Rysunek 5.6 Czujnik odbiciowy

5.1.7. Ekspander wyprowadzeń mikrokontrolera PCF8574

Do obsługi czujników odbiciowych użytych w projekcie autor wykorzystał moduły rozszerzające liczbę wejść płytki Arduino Leonardo. Każdy ekspander pozwala powiększyć liczbę wyprowadzeń mikrokontrolera o dodatkowych 8 portów. Jednakże ich najważniejszą cechą z punktu widzenia założeń projektu jest możliwość wykrywania zmiany stanu na dowolnym cyfrowym wejściu. W przypadku, kiedy którakolwiek z lokomotyw uaktywni czujnik zajętości toru, ekspander wysyła sygnał przerwania do centrali sterującej. Pozwala to uniknąć konieczności ciągłego monitorowania stanów wszystkich portów. Ekspandery komunikują się z mikrokontrolerem za pomocą magistrali I2C, a możliwość ręcznej zmiany ich adresu w zakresie od 0x20 do 0x27 pozwala na jednoczesne podłączenie do 8 urządzeń.

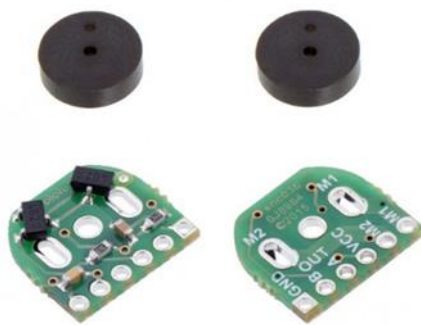


Źródło: botland.pl

Rysunek 5.7 Ekspander wyprowadzeń

5.1.8. Enkoder magnetyczny Pololu 3081

Na osi kołowej wagonów pociągowych autor zamontował magnetyczne czujniki obrotu, czyli tzw. enkodery. Wykorzystany w projekcie model Pololu 3081 składa się z dwóch czujników Halla zamontowanych na płytce PCB oraz wirnika z magnesami. Wirnik przypomina kształtem gruby dysk, na którym znajduje się sześć równomiernie rozstawionych biegunów magnetycznych, co pozwala uzyskać rozdzielczość 12 impulsów na obrót. Wprawienie wirnika w ruch obrotowy powoduje przemieszczanie się biegunów magnetycznych, wzbudzenie czujników Halla oraz generowanie sygnału o przebiegu prostokątnym na wyjściach kwadraturowych. W zależności od wzajemnego przesunięcia kąтового sygnałów można rozróżnić aktualny kierunek obrotu wirnika.



Źródło: botland.pl

Rysunek 5.8 Enkodery magnetyczne Pololu

5.1.9. Akumulator Li-Pol XP17002EX

Autor zdecydował, że centrala sterująca, której rolę pełni Arduino Leonardo, będzie zasilana z niezależnego źródła. W tym celu zaopatrzył się w akumulator litowo-polimerowy XP17002EX o napięciu wyjściowym 7,4V i pojemności 1700mAh. W akumulatorach tego typu elektrolit występuje w formie stałej lub żelowej, co pozwala na produkcję ogniw w dowolnym kształcie. Charakteryzuje się również większą gęstością, niż w przypadku baterii litowo-jonowych, dzięki czemu uzyskiwana jest większa pojemność przy takich samych rozmiarach.

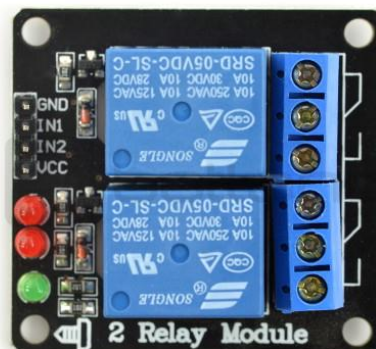


Źródło: botland.pl

Rysunek 5.9 Akumulator Li-Pol

5.1.10. Moduł przekaźnikowy SRD-05VDC-SL-C

Moduł wyposażony jest w dwie cewki przekaźnikowe o napięciu sterowania 5V, które pozwalają sterować stykami NO/NC za pomocą sygnałów nadawanych bezpośrednio z mikrokontrolera. Przekaźniki wykorzystywane są w projekcie do sterowania i przełączania obwodów zwrotnicowych, które wymagają zasilania minimum 12V DC.



Źródło: botland.pl

Rysunek 5.10 Moduł przekaźnikowy

5.2. Układ sterowania lokomotywą

Po skompletowaniu wszystkich niezbędnych części i elementów, autor rozpoczął budowę układów odpowiedzialnych za sterowanie i komunikację ze składami pociągowymi. Wcześniej wyprowadzone z lokomotywy przewody zasilania 12V DC zostały podłączone do stabilizatora L7805CV, który utrzymuje na swoim wyjściu stałe napięcie 5V. Maksymalny prąd wyjściowy jaki jest w stanie zapewnić stabilizator wynosi 1,5A, co stanowi wystarczający zapas zgodnie z poniższymi obliczeniami:

$$I_{s1_max} = I_{apm} + I_{enk} + I_{lss} + I_{s2_max} = 150mA + 50mA + 50mA + 800mA = 1,05A < 1,5A,$$

gdzie: I_{s1_max} – maksymalny prąd wyjściowy stabilizatora L7805CV,

I_{apm} – maksymalny prąd zasilania Arduino Pro Mini,

I_{enk} – maksymalny prąd zasilania enkodera,

I_{lss} – maksymalny prąd zasilania logiki sterownika silnika,

I_{s2_max} – maksymalny prąd stabilizatora LM1117T

Stabilizator LM1117T został włączony szeregowo zaraz za wyjściem ze stabilizatora L7805CV, aby zredukować liczbę kondensatorów, jaka byłaby wymagana w przypadku połączenia równoległego. Zapewnia on na wyjściu stabilne napięcie 3,3V DC, które służy jedynie do zasilania modułu radiowego o bardzo małym poborze prądu.

Porty komunikacyjne modułu radiowego nRF24L01 zostały połączone poprzez magistralę SPI z portami D11 (MOSI), D12 (MISO) i D13 (SCK) płytki Arduino Pro Mini. Odpowiadają one za synchroniczną wymianę danych pomiędzy urządzeniami i pełnią następujące funkcje:

- MOSI (*ang. Master Output Slave Input*) – przesyłanie danych z mikrokontrolera do urządzenia odbiorczego.
- MISO (*ang. Master Input Slave Output*) – przesyłanie danych z odbiornika do mikrokontrolera.
- SCK (*ang. Serial Clock*) – sygnał taktujący odpowiedzialny za synchronizację.

Ponadto do pinów D7, D8 i D9 zostały podpięte pozostałe porty modułu radiowego zgodnie z następującą kolejnością:

- CE (*ang. Chip Enable*) – wejście określające stan nadawania (logiczne „0”) lub stan odbierania (logiczne „1”).

- CSN (*ang. Chip Select NOT*) – wejście pełniące rolę adresu, potwierdzające czy aktualnie przesyłane dane mają zostać odebrane przez urządzenie.
- IRQ (*ang. Interrupt*) – wyjście generujące sygnał przerwania w odpowiedzi na pewne zdarzenia.

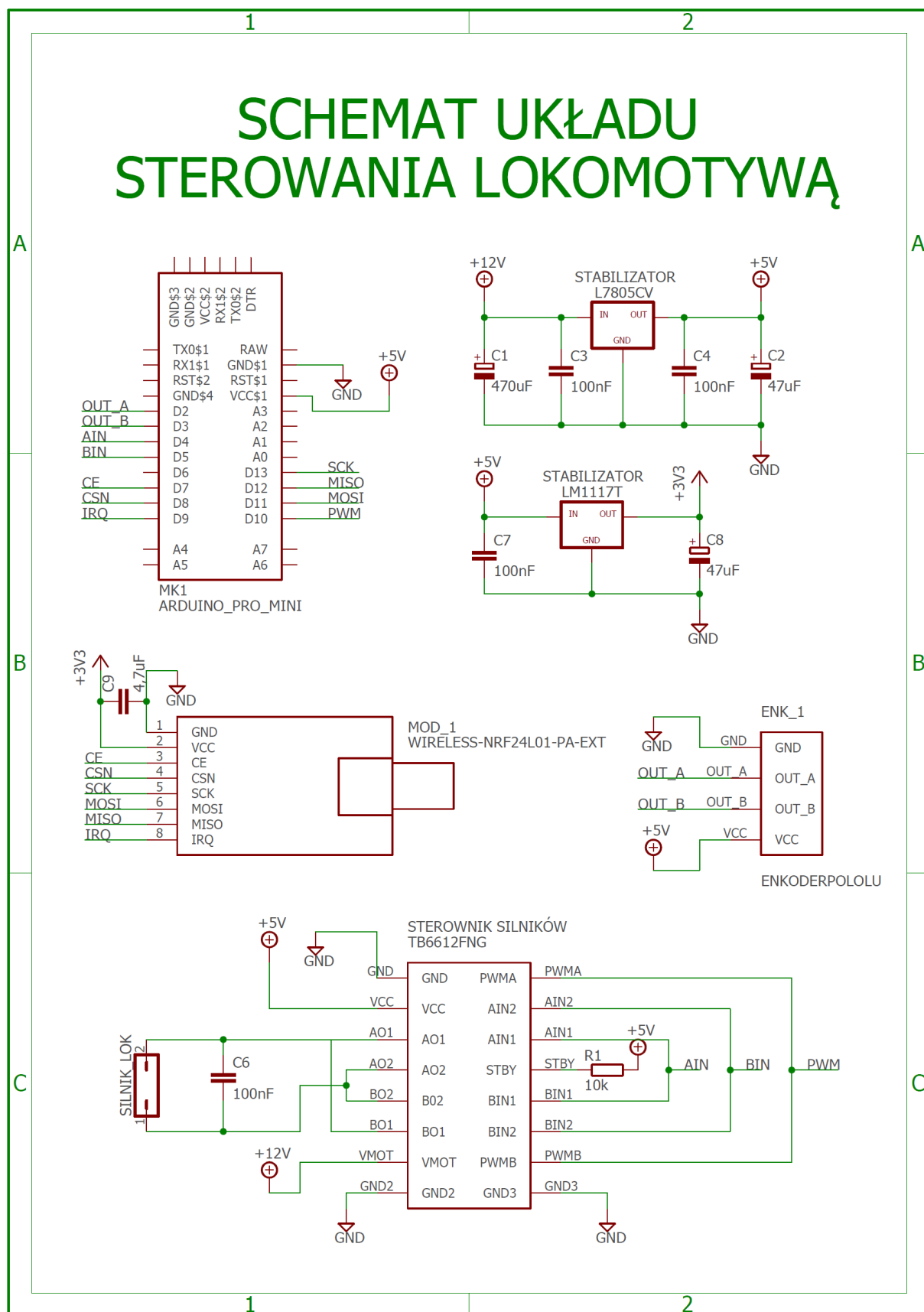
W następnym etapie podłączony został dwukanałowy sterownik silnikowy TB6612FNG. Jednakże w celu zwiększenia jego wydajności prądowej autor zdecydował się na sparowanie kanałów, a końcowy diagram połączeń przedstawił w poniższej tabeli:

| Nr poł. | Porty sterownika | Opis | Port mikrokontrolera |
|---------|------------------|--|----------------------|
| 1 | AIN1 + BIN1 | Wejście służące do sterowania kierunkiem obrotów (kanał A+B) | D4 |
| 2 | AIN2 + BIN2 | Wejście służące do sterowania kierunkiem obrotów (kanał A+B) | D5 |
| 3 | AO1 + BO1 | Wyjście sterowania silnikiem | - |
| 4 | AO2 + BO2 | Wyjście sterowania silnikiem | - |
| 5 | PWMA + PWMB | Sygnał modulacji PWM | D10 |

Na wejście STBY mostka TB6612FNG, odpowiadające za przełączanie układu pomiędzy trybem pracy a uśpieniem, zostało podciągnięte stałe napięcie 5V DC. Ponadto do portu VMOT, będącym wejściem zasilania silników, doprowadzone zostało napięcie 12V DC.

Na koniec do układu podłączono jeszcze enkoder magnetyczny Pololu, którego porty wyjść kwadraturowych OUTA oraz OUTB zwarto z wejściami D2 i D3 Arduino Pro Mini.

Wszystkie połączenia w zbudowanym układzie zostały wykonane przewodami o polu przekroju $0,35\text{mm}^2$ (AWG 22). Miejsca, w których wykonano łączenia lutowane lub które były w inny sposób narażone na ryzyko zwarcia, zostały zabezpieczone rurkami termokurczliwymi oraz taśmą izolacyjną. Gotowe układy zostały ulokowane wewnątrz wagonów w każdym z dwóch składów pociągowych. Dokładny schemat elektryczny zaprojektowanego systemu sterowania i komunikacji przedstawiony został na rysunku 5.11.

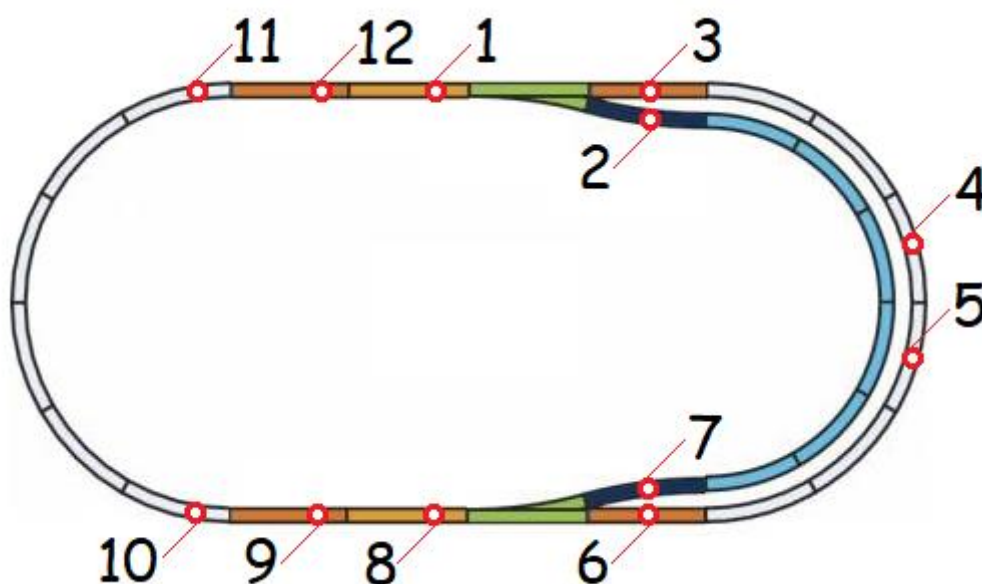


Rysunek 5.11 Schemat elektryczny układu sterowania lokomotywą

5.3. Układ centrali nadzorująco-sterującej

Zgodnie z wcześniej ustalonymi założeniami centrala sterująca, którą stanowi jednostka Arduino Leonardo, została umieszczona w czarnej obudowie znajdującej się w centrum makiety. Do portów SDA (*ang. Serial Data Line*) i SCL (*ang. Serial Clock Line*), obsługujących magistralę szeregową I2C, zostały podłączone odpowiednie porty ekspanderów wyprowadzeń PCF8574. Dodatkowo do wejść cyfrowych D0 i D1 zostały doprowadzone sygnały obsługujące przerwania.

W następnej kolejności autor rozpoczął montaż czujników odbiciowych KTIR0711S do podkładów torowiska. Przewody zasilająco-sterujące od każdego z czujników zostały przeciągnięte przez otwory w makiecie, a następnie poprowadzone spodnią warstwą do obudowy centrali. Tam zostały z powrotem wyciągnięte na wierzch i podłączone do odpowiednich portów I/O ekspanderów. Szczegółowe rozmieszczenie oraz numeracja modułów odbiciowych przedstawione zostały na poniższym rysunku.



Rysunek 5.12 Rozmieszczenie czujników odbiciowych na makiecie

Obwody sterowania cewkami mechanizmów zwrotnicowych podłączone zostały do portów NO/NC modułu przekaźnikowego SRD-05VDC-SL-C. Na wejścia COM przekaźników podane zostało napięcie bezpośrednio z regulatora PIKO o wartości 12V DC. Porty IN1 i IN2, odpowiedzialne za przełączanie styków NO/NC, podpięte zostały do pinów D5 i D6 płytki Arduino Leonardo.

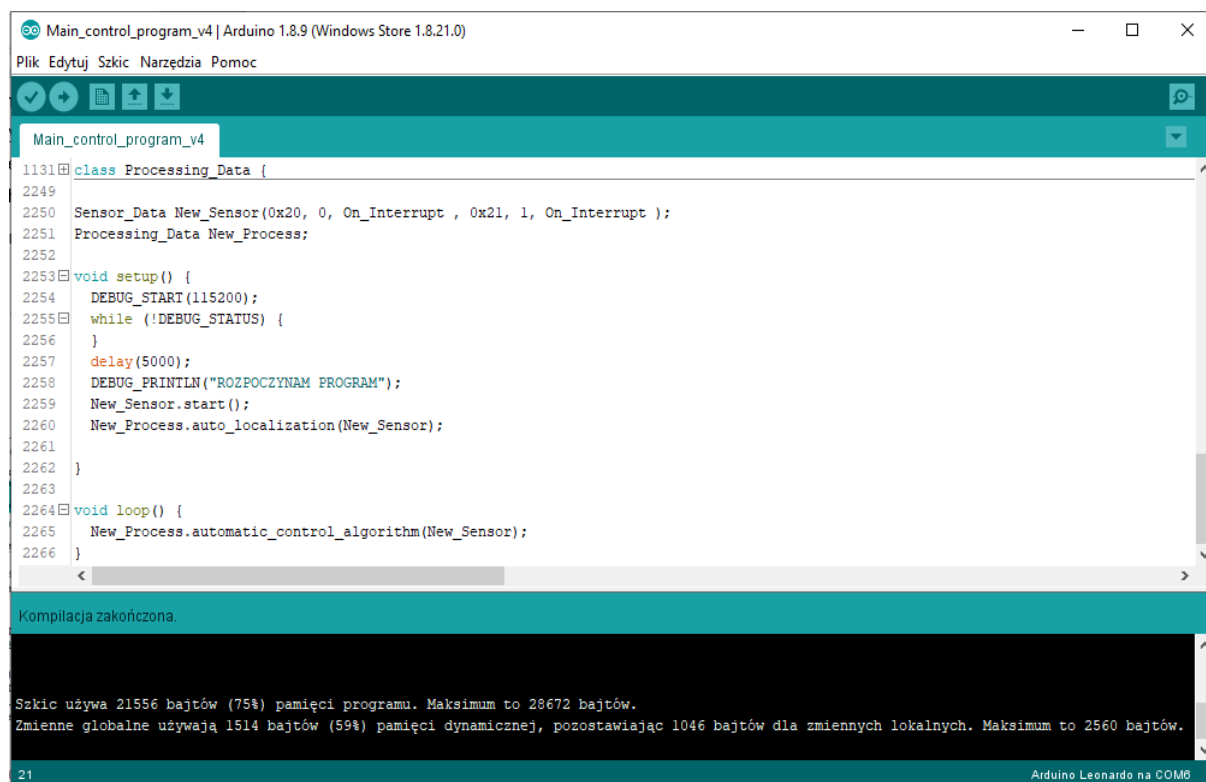
Na koniec do układu centrali sterującej dołączono moduł radiowy w sposób analogiczny, jak w przypadku układu sterowania lokomotywą (5.2). Dokładny schemat elektryczny zbudowanego systemu przedstawiony został na rysunku 5.13.

6. Opis algorytmu automatycznego sterowania ruchem kolejowym.

Programowanie modułów Arduino odbywa się za pośrednictwem specjalnie przygotowanej, wieloplatformowej aplikacji Arduino IDE. Napisana ona została w języku Java, natomiast samo środowisko programistyczne używa języka Processing oraz Wiring. Aplikacja korzysta z szeregu gotowych bibliotek, które między innymi znacząco ułatwiają programowanie wejść i wyjść oraz umożliwiają szybką implementację popularnych i kompatybilnych z Arduino urządzeń.

Strukturę każdego programu można podzielić się na dwie zasadnicze części:

- Funkcję *setup()* – wykonywana jest ona jednorazowo, jedynie na początku działania programu.
- Funkcję *loop()* – wykonywana jest wielokrotnie przez cały czas działania programu.



Rysunek 6.1 Środowisko programistyczne Arduino

Na rysunku 6.1 przedstawiona została składnia funkcji *setup()* oraz *loop()*, wykorzystanej w programie napisanym dla centrali sterującej. Przy pierwszym uruchomieniu programu następuje parametryzacja i nawiązanie komunikacji pomiędzy Arduino Leonardo a komputerem, konfiguracja modułów ekspanderów oraz wywołanie funkcji auto-lokalizacji. Następnie mikrokontroler zaczyna cyklicznie wykonywać polecenia składające się na właściwy algorytm automatycznego sterowania. Jedynym sposobem na przerwanie działania programu jest odcięcie zasilania lub zresetowanie modułu.

Chociaż funkcja *loop()* jest wykonywana w niekończącej się pętli, istnieje możliwość chwilowego wstrzymania jej działania i wywołania innej funkcji za pomocą tak zwanego przerwania. Tego rodzaju sytuacja występuje w momencie, kiedy na odpowiednio zadeklarowanym porcie opisanym w dokumentacji producenta jako *INT*, pojawia się sygnał o zboczu narastającym. Wówczas mikrokontroler natychmiast przerywa aktualnie wykonywane polecenie, wypełnia instrukcje zawarte w funkcji przerwania, a następnie wraca do normalnego działania w tym samym punkcie, w którym ostatnio skończył.

W ramach projektu autor przygotował trzy odrębne programy, które zostały wgrane na moduły Arduino Pro Mini znajdujące się w składach pociągowych oraz na Arduino Leonardo, które pełni rolę centrali. Algorytmy sterowania lokomotywami posiadają identyczny schemat działania, a różnią się między sobą jedynie adresowaniem i konfiguracją ustawień komunikacji. Ich zadaniem jest przede wszystkim odbiór poleceń z centrali,ysterowanie silników oraz przesłanie aktualnego statusu. Główny algorytm sterujący jest natomiast znacznie obszerniejszy, a jego rolą jest monitorowanie wskazań czujników zajętości toru, planowanie drogi lokomotyw oraz kontrola ruchu. Programy przygotowane zostały w języku obiektowym C++, a ich struktura umożliwia łatwe dodawanie nowych rozszerzeń oraz modyfikacji.

6.1. Algorytm centrali nadzorująco-sterującej

Autor pracy przygotowując algorytm zapewnił możliwość nawiązania komunikacji pomiędzy centralą nadzorującą a komputerem, za pomocą magistrali szeregowej. Ze względu na fakt, że moduł Arduino Leonardo posiada wbudowany port MicroUSB, do ustanowienia transmisji danych wystarczy zwykły kabel w standardzie USB. Program posiada możliwość raportowania błędów oraz monitorowania wybranych parametrów pracy układu. Stosowne komunikaty wyświetlone zostaną w oknie „*Monitor portu szeregowego*” kompilatora Arduino. W celu wyłączenia komunikacji z komputerem oraz zapewnienia samodzielnej pracy układu na zasilaniu baterijnym należy zakomentować 14 linijkę kodu „*#define PROGRAM_DEBUG*”.

6.1.1. Ogólna struktura programu

Algorytm centrali nadzorująco-sterującej korzysta z szeregu ogólnie dostępnych bibliotek, które w ramach projektu zostały w niektórych miejscach zmodyfikowane, aby zapewnić odpowiednią funkcjonalność:

- 1) `#include "RF24.h"` – biblioteka wprowadzająca gotowe funkcje obsługi protokołu komunikacyjnego dla modułów radiowych nRF24L01,
- 2) `#include <SPI.h>` - biblioteka zapewniająca możliwość obsługi magistrali szeregowej SPI, która jest z kolei wykorzystywana przez wyżej wspomnianą bibliotekę `RF24.h`,
- 3) `#include <PCF8574.h>` - biblioteka, która zawiera gotowe funkcje obsługujące moduł ekspandera wyprowadzeń PCF8574,
- 4) `#include <Wire.h>` - biblioteka wprowadzająca możliwość korzystania z magistrali szeregowej (min. USB i RS232) przy komunikacji z komputerem.

Struktura programu ma formę obiektową, tzn. składa się z klas, które stanowią całkowitą lub częściową definicję obiektów, a także samych obiektów, będących fizyczną instancją przechowującą konkretne wartości zmiennych. W poniżej tabeli przedstawiono rodzaje klas, metod oraz obiektów, które występują w algorytmie centrali.

| Klasa | Metody | Obiekt | Opis |
|------------------|---|-------------|--|
| Control_Data | void Set_PWM void Set_direction void Set_acceleration void Set_encoder void Set_encoder_priority void Set_fast_stop void Print_Data | New_Control | Klasa służąca do przechowywania zmiennych, które określają parametry jazdy lokomotywy. Obiekt <i>New_Control</i> jest w całości przesyłany do układu sterowania lokomotywy, niosąc informację o prędkości, przyspieszeniu itp. |
| Transmitted_Data | byte Get_adress byte Get_current_speed byte Get_error byte Get_encoder | New_Data | Klasa, w której przechowywane są dane odebrane z lokomotywy. Obiekt <i>New_Data</i> jest nadpisywany za każdym razem gdy z bufora zostają odczytane nowe dane. |
| Loc_State | byte Get_loc_speed byte Get_loc_priority byte Get_loc_section byte Get_loc_waiting_status | New_State | Klasa, w której zapisywane i odczytywane są parametry stanu obydwu lokomotyw. Przechowywane dane, do których |

| | | | |
|------------------|--|------------------|--|
| | byte Get_loc_mode void Set_loc_speed void Set_loc_priority void Set_loc_section void Set_loc_waiting_status void Set_loc_mode | | należą min. prędkość, tryb jazdy czy zajmowana sekcja torowa, przeznaczone są jedynie do użytku wewnętrznego i nie są przesyłane za pomocą modułu radiowego. |
| RF_Communication | void start void transmit bool receive void reconnect byte Check_error_status void Reset_error_status | New_Com | Klasa, która odpowiedzialna jest za wysyłanie i odbieranie danych za pomocą transmisji radiowej. |
| Sensor_Data | void start void Read_and_Save byte *Get_Sensor_Data | New_Sensor | Klasa, która służy do zapisywania i przechowywania danych z czujników zajętości toru. |
| Steering_control | void steering_begin void steering_control | New_Steering | Klasa odpowiedzialna za przełączanie pozycji zwrotnic |
| Localization | void searching_localization bool find_proper_loc byte check_sensor_order void Reset_error void Print_position byte Get_Poz_Loc byte Get_Error_status byte Get_dir_loc byte Get_change_status | New_Localization | Klasa, która definiuje algorytmy poszukiwania lokomotywy na podstawie aktualnych danych z czujników oraz rejestru poprzednich pozycji. |
| Processing_Data | void auto_localization void section_autopilot void automatic_control_algorithm bool check_free_section | New_Process | Klasa, w której zaimplementowane zostały algorytmy początkowej auto-lokalizacji lokomotyw oraz kontroli ruchu kolejowego |

Ponadto w programie znajduje się deklaracja funkcji *void On_Interrupt*, która jest wywoływana w momencie nadejścia przerwania. Jej jedynym zadaniem jest zmiana wartości logicznej zmiennej *bool STATE*, która niesie informację o zmianie stanu czujników zajętości toru. Na rysunku 6.2 przedstawiono szkielet programu centrali sterującej, widziany z poziomu kompilatora Arduino.

```
Control_program | Arduino 1.8.9 (Windows Store 1.8.21.0)
Plik Edytuj Szkic Narzędzia Pomoc

Control_program $
1 #include "RF24.h"
2 #include <SPI.h>
3 #include <PCF8574.h>
4 #include <Wire.h>
5
6 #define DIR1 1
7 #define DIR2 0
8 #define FAST 150
9 #define SLOW 50
10 #define LOC1 0
11 #define LOC2 1
12 #define DEBUG_PRINTER Serial
13
14 // #define PROGRAM_DEBUG
15 #ifdef PROGRAM_DEBUG
16 #define DEBUG_PRINT(...) { DEBUG_PRINTER.print(__VA_ARGS__); }
17 #define DEBUG_PRINTLN(...) { DEBUG_PRINTER.println(__VA_ARGS__); }
18 #define DEBUG_START(...) { DEBUG_PRINTER.begin(__VA_ARGS__); }
19 #define DEBUG_STATUS DEBUG_PRINTER
20 #else
21 #define DEBUG_PRINT(...) {}
22 #define DEBUG_PRINTLN(...) {}
23 #define DEBUG_START(...) {}
24 #define DEBUG_STATUS 1
25 #endif
26
27 byte address[][6] = {"1Node", "2Node", "3Node", "4Node"};
28 bool STATE;
29 void On_Interrupt();
30
31 //-----PARAMETRY PRZESYLANE DO LOKOMOTYWY-----
32 class Control_Data {
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76 //-----DANE ODEBRANE Z LOKOMOTYWY-----
77 class Transmitted_Data {
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102 //-----STAN LOKOMOTYWY-----
103 class Loc_State {
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159 //-----ODBIERANIE I WYSYLANIE DANYCH PRZEZ MODUL RADIOWY-----
160 class RF_Communication {
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229 //-----ODBIERANIE I PRZECHOWYWANIE DANYCH Z CZUJNIKÓW ODBICIOWYCH-----
230 class Sensor_Data {
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501 //-----FUNKCJA OBSŁUGI PRZERWANIA-----
502 void On_Interrupt() {
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115 //-----PRZETWARZANIE DANYCH I PRZELICZANIE STEROWANIA-----
1116 class Processing_Data {
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2
```

6.1.2. Komunikacja radiowa

Przebieg prac nad rozwojem algorytmu komunikacji:

Kluczowym elementem do zapewnienia pełnej funkcjonalności oraz niezawodnego działania układu było dopracowanie algorytmu komunikacji pomiędzy centralą sterująco-nadzorującą, a lokomotywami. Jednakże na przestrzeni czasu, kiedy program był rozwijany oraz testowany, element ten sprawiał największe trudności oraz najczęściej zawodził.

Początkowo autor pracy zamierzał ukształtować formę komunikacji w taki sposób, żeby lokomotywy mogły wysyłać informacje odnośnie swojego statusu w dowolnym, wybranym przez siebie momencie, bez konieczności przesyłania specjalnego zapytania przez centralę. Pomysł ten został jednakże szybko zrewidowany, ponieważ występowały liczne problemy:

- a) Pomimo stosowania różnych częstotliwości nadawania przez moduły radiowe znajdujące się w lokomotywach, przesyłane sygnały potrafiły zostać zniekształcone lub wzajemnie się zakłócać. Chociaż taka sytuacja powinna zdarzać się niezwykle rzadko to w rzeczywistości do bufora dosyć często trafiały niepełne dane. Przyczynami takiego zachowania są:
 - specyfika otoczenia (zamknięte pomieszczenie, występowanie innych urządzeń elektronicznych np. komputera, routera WiFi), która negatywnie wpływa na propagację fal radiowych,
 - krótki zasięg nadawania modułów radiowych, dodatkowo ograniczony poprzez ścianki wagonu,
 - przechodzenie obydwu modułów (w centrali i w lokomotywie) w tryb nadawania w tym samym czasie.
- b) W wielu wypadkach moduły nie przesyłały odpowiedzi zwrotnej, pomimo stosowania zaimplementowanego mechanizmu ponawiania prób wysyłania.
- c) Odbierana była nieprawidłowa wartość zmiennej, która określa rozmiar przesyłanej informacji lub następowało przepełnienie bufora odczytu, w efekcie moduł radiowy zawieszał się, a odbieranie danych trwało w nieskończoność.

W wyniku napotkanych problemów struktura komunikacji została zmieniona na model master-slave, aby możliwie jak najbardziej ograniczyć częstotliwość przesyłanych informacji. Lokomotywy wysyłały dane odnośnie statusu jedynie w sytuacji jeśli wcześniej otrzymały instrukcje jazdy od centrali. Ponadto wprowadzono dodatkowe ograniczenia czasowe stanu, w którym moduł pozostawał w trybie odbierania. Jeżeli występowały błędy w komunikacji to następowało automatyczne resetowanie modułu. Wprowadzone poprawki pozwoliły ustanowić

stabilną pracę układu i wymianę danych, chociaż nie udało się wyeliminować w pełni problemu gubienia pakietów odpowiedzi zwrotnych. Jednakże stanowi to jedynie niedogodność natury programowej i nie wpływa na funkcjonalność układu.

Architektura komunikacyjna modułów radiowych nRF24L01:

Moduły nRF24L01 działają w paśmie fal radiowych o częstotliwości 2,4 – 2,525 GHz i pozwalają stworzyć 125 kanałów o szerokości 1MHz. Każdy z kanałów fizycznych dodatkowo podzielony jest na sześć kanałów logicznych, które noszą nazwę „rury” (*ang. pipe*). Domyślnie jedna rura w danym kanale służy wyłącznie do nadawania, natomiast pozostałych pięć do odbierania danych. Kanały logiczne można ponadto adresować, przy czym pełen 5-bajtowy adres można przypisać jedynie rurze nadającej (o indeksie „0”) oraz pierwszej odbierającej (indeks „1”),. Pozostałe cztery kanały logiczne posiadają adresy, których pierwsze 4 bajty są identyczne jak bajty adresu rury odbierającej „1” i różnią się jedynie ostatnim bajtem (który można ustawić programowo).

Zasada działania algorytmu nadawania:

- 1) Wpisanie odpowiednich parametrów jazdy do zmiennych przechowywanych przez obiekt *New_Com*.
- 2) Przejście z trybu nasłuchiwania do trybu wysyłania.
- 3) Otworzenie rury nadającej o adresie odpowiadającym adresowi lokomotywy, do której mają zastać przesłane dane.
- 4) Wysłanie danych oraz informacji o ich rozmiarze.
- 5) Oczekiwanie na odpowiedź zwrotną. W przypadku błędu lub przekroczenia czasu odpowiedzi, wystawienie odpowiedniego kodu błędu.
- 6) Powrót do trybu nasłuchiwania.

Zasada działania algorytmu odbierania:

- 1) Sprawdzenie, czy w buforze znajdują się dane do odebrania,
- 2) Odebranie informacji o statusie lokomotywy poprzez nadpisanie obiektu *New_Data*
- 3) Wysłanie odpowiedzi zwrotnej.

Schemat przebiegu odbierania i wysyłania poprzez moduły radiowe znajdujące się w lokomotywach jest bardzo podobny jak w przypadku centrali, a różni się jedynie rodzajem przesyłanego obiektu oraz adresowaniem rur.

6.1.3. Metoda auto-lokalizacji lokomotyw

Kolejnym bardzo ważnym elementem w strukturze algorytmu automatycznego sterowania jest implementacja metody określającej warunki brzegowe (początkowe pozycje lokomotyw oraz pierwszeństwo jazdy). Założono, że w momencie uruchomienia programu, pociągi będą znajdować się w różnych sekcjach torowych tak, aby nie doszło do kolizji. Poniżej przedstawiono uogólniony schemat postępowania, który został zaimplementowany w metodzie auto-lokalizacji (*void auto_localization*) :

- 1) Przesłanie do lokomotywy pierwszej (od tego momentu umownie nazywanej lokomotywą „szarą”) rozkazu jazdy z ustaloną prędkością.
- 2) Oczekiwanie na nadejście przerwania, wskazującego na zmianę stanu czujników zajętości toru.
- 3) Odczyt sygnałów z czujników i określenie pozycji pociągu.
- 4) Przesłanie rozkazu natychmiastowego zatrzymania do lokomotywy szarej.
- 5) Wysłanie sygnału rozpoczęcia jazdy przez lokomotywę drugą („czarną”).
- 6) Powtórzenie czynności z punktów 2), 3) oraz 4) dla lokomotywy czarnej.
- 7) Określenie numeru sekcji, w której przebywa każdy z pociągów.
- 8) Nadanie statusu pierwszeństwa dla lokomotywy, która znajduje się w sekcji pierwszej. Jeżeli żaden pociąg nie spełnia tego warunku to pierwszeństwo otrzymuje lokomotywa szara.
- 9) Ustalenie dodatkowych warunków jazdy i pozwoleń dla drugiego pociągu.
- 10) Przesłanie odpowiednich instrukcji jazdy do obydwu lokomotyw.
- 11) Inicjalizacja właściwego algorytmu sterowania ruchem kolejowym.

6.1.4. Metoda kontroli i sterowania ruchem kolejowym

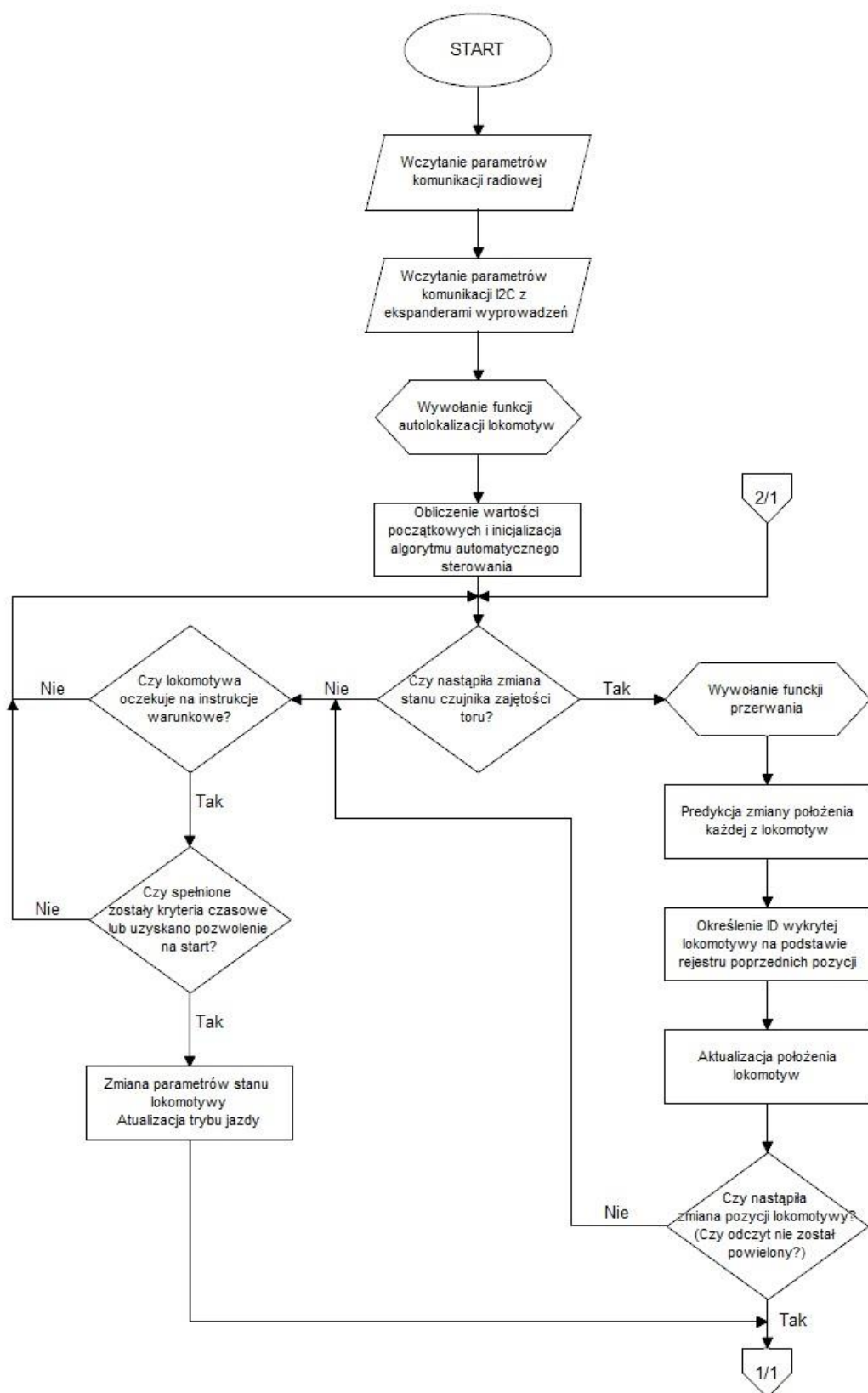
Najważniejszą częścią opracowanego w ramach projektu algorytmu jest mechanizm określający zachowanie poszczególnych lokomotyw w momencie zmiany pozycji lub sekcji. Odpowiedzialne są za to przede wszystkim dwie metody *void section_autopilot* oraz *void automatic_control_algorithm* należące do klasy *Processing_Data*. Pierwsza z nich definiuje tryby jazdy (zmienna *loc_mode*) w jakie przechodzić ma lokomotywa w zależności od jej zmiennych stanu oraz aktualnie osiągniętej pozycji. Ponadto metoda ta odpowiedzialna jest za dobór parametrów jazdy lokomotywy oraz wywołanie algorytmu wysyłania i odbierania danych.

Druga metoda ma natomiast za zadanie monitorować status zmiany stanu czujników zajętości toru (zmienna *STATE*), a w przypadku kiedy zarejestrowana zostanie zmiana pozycji przez jeden z pociągów, wywołać metodę *void section_autopilot*. Program ponadto rozpatruje kryteria czasowe i pozwolenia, które odnoszą się do warunkowych trybów jazdy.

W poniższej tabeli przedstawiono tryby jazdy zdefiniowane w algorytmie kontroli ruchu:

| Tryb | Prędkość jazdy | Opis |
|------|----------------|---|
| 1 | Szybka | Podstawowy tryb jazdy z dużą prędkością. |
| 2 | Wolna | Podstawowy tryb jazdy z małą prędkością. |
| 3 | Postój | Podstawowy tryb jazdy na postoju. |
| 4 | Postój | Warunkowy tryb jazdy. Postój przed przejazdem kolejowym lub na stacji aż do momentu spełnienia kryterium czasowego. |
| 5 | Szybka | Warunkowy tryb jazdy. Jazda z dużą prędkością do momentu spełnienia kryterium czasowego i zwolnienia przy dojeździe do stacji lub przejazdu. |
| 6 | Szybka | Warunkowy tryb jazdy. Jazda z dużą prędkością do momentu spełnienia kryterium czasowego i zwolnienia przy dojeździe do zwrotnicy (w sekcji drugiej). |
| 7 | Szybka | Warunkowy tryb jazdy. Jazda z dużą prędkością do momentu spełnienia kryterium czasowego i zwolnienia przy dojeździe do zwrotnicy (w sekcji trzeciej). |
| 8 | Postój | Warunkowy tryb jazdy. Postój przed zwrotnicą i oczekiwanie na otrzymanie pozwolenia dalszej jazdy. |
| 9 | Wolna | Warunkowy tryb jazdy. Jazda z małą prędkością przed zwrotnicą i oczekiwanie na sprawdzenie statusu zajętości kolejnej sekcji. |
| 10 | Wolna | Warunkowy tryb jazdy. Jazda z małą prędkością po otrzymaniu pozwolenia na przejazd przez zwrotnicę bez konieczności zatrzymywania się. |

Na rysunku 6.3 oraz 6.4 przedstawiono uogólniony schemat blokowy działania algorytmu kontroli i sterowania ruchem kolejowym.



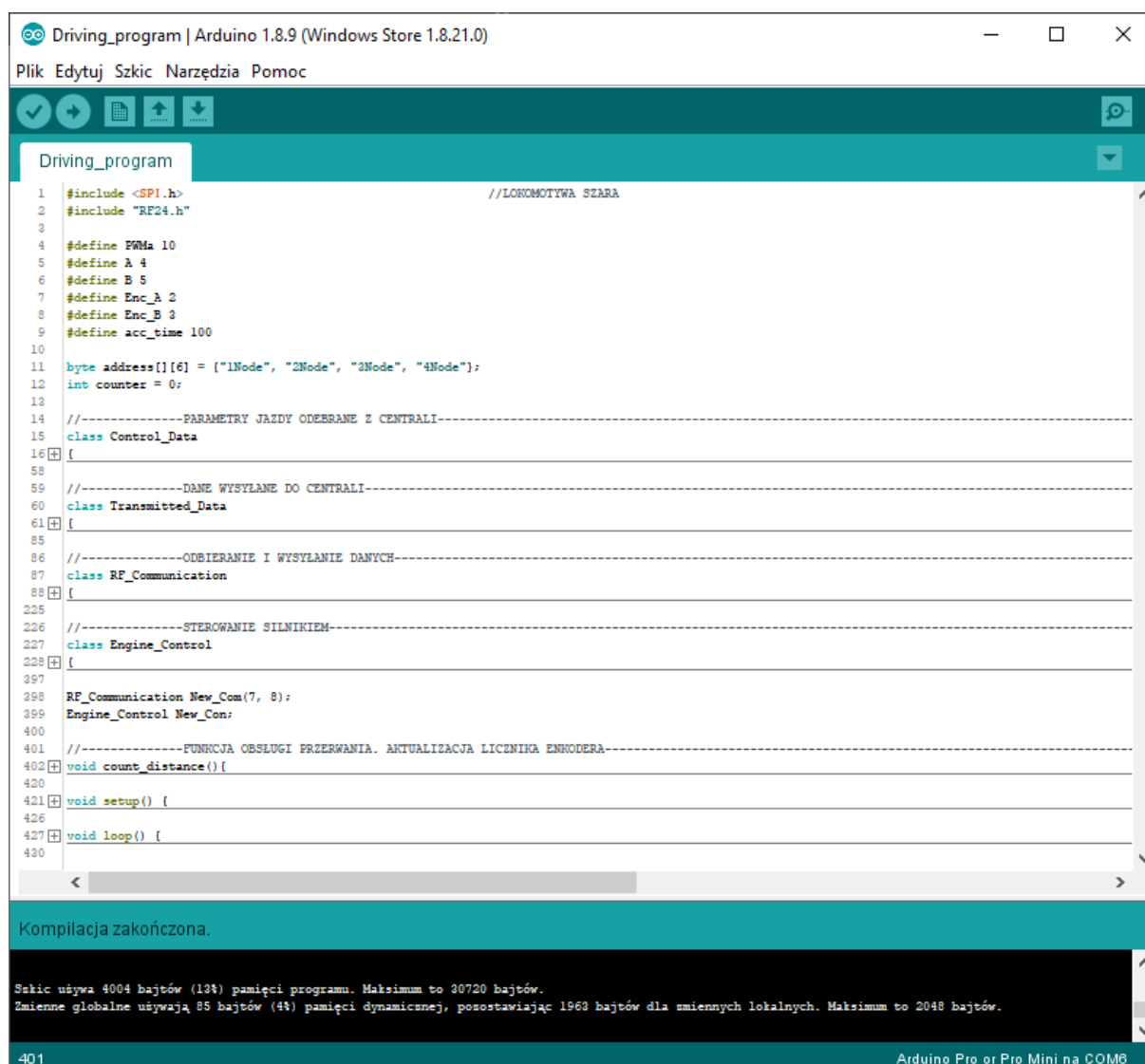
Rysunek 6.3 Schemat blokowy algorytmu kontroli ruchu kolejowego 1/2



Rysunek 6.4 Schemat blokowy algorytmu kontroli ruchu kolejowego 2/2

6.2. Algorytm sterowania lokomotywą

Dla modułów Arduino Pro Mini, które pełnią funkcje jednostek obliczeniowych w układach sterowania lokomotywami, przygotowane zostały dwa osobne algorytmy. Cechują się one jednak identyczną funkcjonalnością, a jedyna różnica pomiędzy nimi odnosi się do sposobu adresowania i indeksowania rur komunikacji radiowej. W skład struktury programu, podobnie jak w przypadku algorytmu centrali, wchodzi klasa przechowująca parametry odbierane i wysyłane przez lokomotywę, a także klasa odpowiedzialna za transmisję radiową. Ponadto sporządzona została nowa klasa *Engine_Control*, która zawiera metody odpowiadające za sterowanie silnikiem lokomotywy. Zaimplementowana została także funkcja obsługi przerwań (*void count_distance*) na sygnały przychodzące z enkodera. Na rysunku 6.5 przedstawiona została struktura algorytmu widziana z poziomu kompilatora, natomiast na rysunku 6.6 ukazany został uproszczony schemat blokowy działania programu.



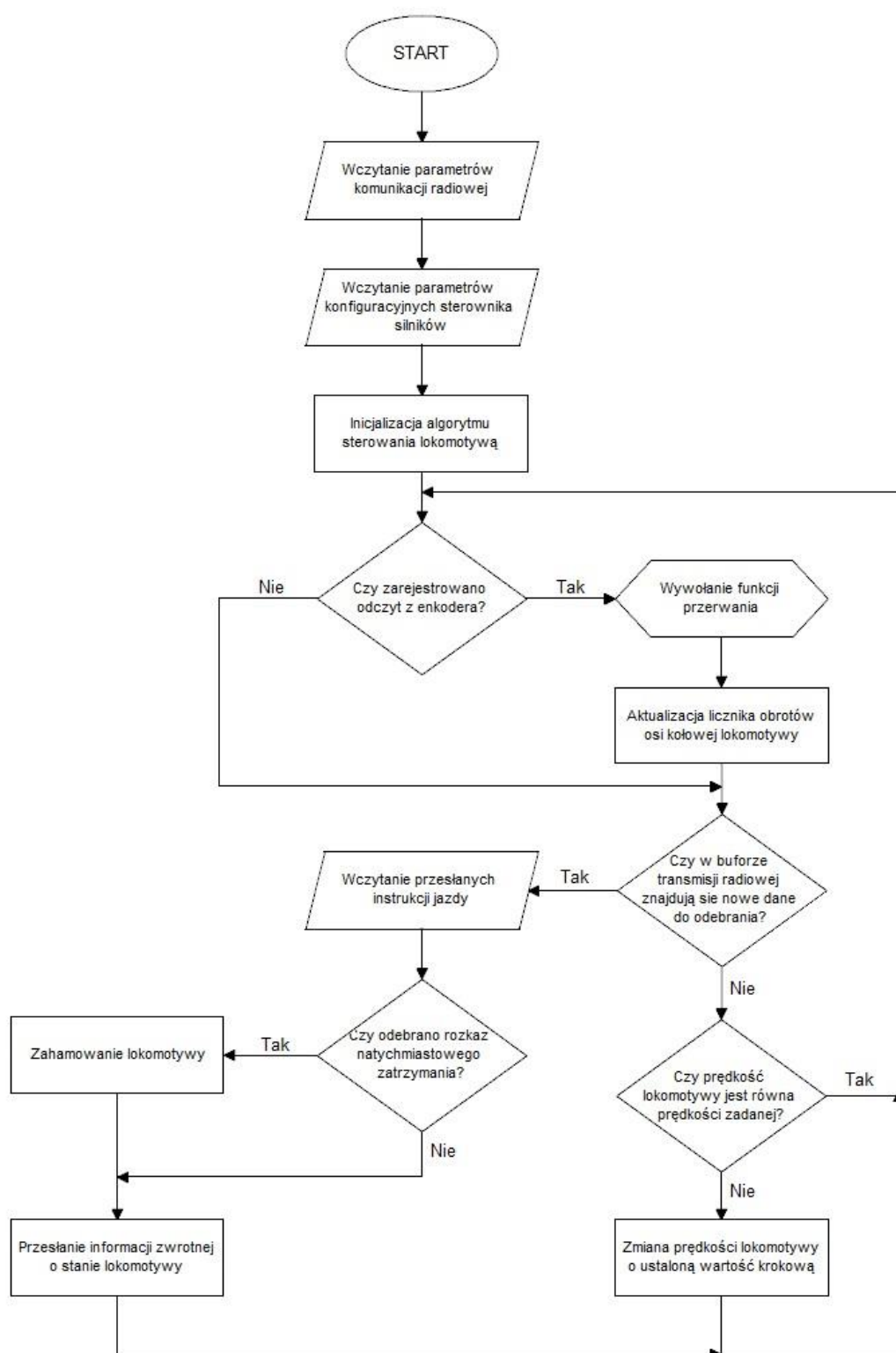
```
1  #include <SPI.h>                                //LOKOMOTYWA SZARA
2  #include "RF24.h"
3
4  #define PWMa 10
5  #define A 4
6  #define B 5
7  #define Enc_A 2
8  #define Enc_B 3
9  #define acc_time 100
10
11 byte address[4] = {"1Node", "2Node", "3Node", "4Node"};
12 int counter = 0;
13
14 //-----PARAMETRY JAZDY ODEBRANE Z CENTRALI-----
15 class Control_Data
16 {
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59 //-----DANE WYSYLANE DO CENTRALI-----
60 class Transmitted_Data
61 {
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86 //-----ODBIERANIE I WYSYLANIE DANYCH-----
87 class RF_Communication
88 {
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126 //-----STEROWANIE SILNIKIEM-----
127 class Engine_Control
128 {
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401 //-----FUNKCJA OBSŁUGI PRZERWANIA. AKTUALIZACJA LICZENIA ENKODERA-----
402 void count_distance() {
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421 void setup() {
422
423
424
425
426
427 void loop() {
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Kompilacja zakończona.

Szkic używa 4004 bajtów (13%) pamięci programu. Maksimum to 30720 bajtów.
Zmienne globalne używają 85 bajtów (4%) pamięci dynamicznej, pozostawiając 1963 bajtów dla zmiennych lokalnych. Maksimum to 2048 bajtów.

401 Arduino Pro or Pro Mini na COM6

Rysunek 6.5 Struktura algorytmu sterowania lokomotywą



Rysunek 6.6 Schemat blokowy algorytmu sterowania lokomotywą

7. Analiza przeprowadzonych badań i eksperymentów.

Po zakończeniu pracy nad algorytmem, autor przystąpił do przeprowadzania badań mających na celu sprawdzenie, czy układ jest w pełni funkcjonalny oraz czy spełnia narzucone założenia. W trakcie przeprowadzania wstępnych testów zauważono, że wagony, w których znajduje się układ sterowania silnikiem lokomotywy, często ulegają wykolejeniu. Rozwiązaniem problemu okazało się odpowiednie dociążenie wagonów za pomocą odważników przymocowanych do dachu.

Ponadto w ramach przeprowadzanych badań, algorytm kilkakrotnie zmodyfikowano w taki sposób, aby na ekran monitora portu szeregowego wypisywane były kluczowe w danym momencie parametry stanu lokomotyw.

7.1. Badanie funkcjonalności układu pod kątem spełnienia założonych kryteriów.

W rozdziale 3 autor przedstawił szereg założeń jakie powinien spełniać zaprojektowany układ. W niniejszym podrozdziale przeprowadzona zostanie analiza badań mających na celu sprawdzenie, czy:

- 1) lokomotywy będą poruszać się bezkolizyjnie i nie wystąpi sytuacja, w wyniku której obydwie lokomotywy znajdą się w tej samej sekcji,
- 2) pociągi będą respektować występowanie przejazdów kolejowych oraz stacji, zwalniając w bezpiecznej odległości przed nimi oraz zatrzymując się w wyznaczonych strefach.

Zgodnie z ustaleniami opisanymi w rozdziałach 4 i 5, na makiecie znajduje się dwanaście czujników zajętości toru (Rysunek 5.12), które zostały programowo przypisane do trzech sekcji torowych. Każda sekcja wyznaczona jest za pomocą zwrotnic, tzn. jej początek i koniec stanowi odpowiednia para rozjazdów. W poniższej tabeli przedstawiono programowe przyporządkowanie czujników do sekcji oraz opis spodziewanego zachowania lokomotywy.

| Sekcja | Czujniki (pozycje lokomotywy) | Opis zachowania pociągów |
|----------|-------------------------------|--|
| Pierwsza | 1, 8, 9, 10, 11, 12 | W tej sekcji znajdują się dwa przejazdy kolejowe, które zostały wyznaczone poprzez pary czujników 9,10 oraz 11,12. Lokomotywy powinny zwalniać w pewnej odległości przed każdym z przejazdów, a następnie zahamować po osiągnięciu zadanej pozycji. Po upływie określonego czasu, pociągi powinny wznowić jazdę. |
| Druga | 3, 4, 5, 6 | W sekcji drugiej znajduje się stacja wyznaczona poprzez parę czujników 4 oraz 5. Pociąg powinien zwolnić przed dojechaniem do stacji, a następnie zatrzymać się na niej na pewien określony czas. Ponadto w przypadku, kiedy lokomotywa dojeżdżać będzie do pozycji 3 lub 6 i nie otrzyma pozwolenia na dalszą jazdę, powinna zahamować i ustąpić pierwszeństwa przejazdu. |
| Trzecia | 2, 7 | Ta sekcja stanowi tor mijania, a lokomotywa która dojedzie do pozycji 2 lub 7 powinna ustąpić pierwszeństwa w przypadku braku pozwolenia na dalszą jazdę. |

Przed przystąpieniem do pierwszego badania, lokomotywę szarą ustawiono na torze pomiędzy czujnikami 11 i 10 w taki sposób, że jej czoło skierowane zostało w stronę czujnika 10. Lokomotywę czarną ulokowano natomiast w sekcji drugiej w kierunku czujnika 4. Następnie uruchomiono układ, obserwując działanie algorytmu samo-lokalizacji.

Lokomotywy nie napotkały żadnych problemów i po określeniu warunków brzegowych rozpoczęły regularną jazdę. Badanie trwało około 30 min i przez cały ten czas nie stwierdzono żadnych nieprawidłowości w zaimplementowanym algorytmie. Zdecydowano się natomiast na korektę wartości regulujących czas, po którym lokomotywy zwalniały przy dojeździe do miejsc postojowych.

Po wprowadzeniu poprawek lokomotywy ustawiono w identycznych miejscach, a układ uruchomiono ponownie przy włączonym monitoringu portu szeregowego. Parametry jazdy były aktualizowane i wyświetlane tylko w momencie, kiedy następowała zmiana stanu czujników zajętości toru. Zarejestrowane wartości przedstawione zostały na rysunku 7.1.

COM6

Wyslij

ROZPOCZYNAJ PROGRAM

| | | | | | | |
|-----------------------|---------------|-------------|------------------------|---------------|-------------|-----------|
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 9 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 19 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 8 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 21 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 8 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 22 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 24 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 6 | Czas: 33 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 8 | Czas: 36 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 9 | Czas: 37 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 9 | Czas: 39 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 10 | Czas: 41 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 11 | Czas: 46 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 12 | Czas: 50 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 1 | Czas: 51 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 3 | Czas: 53 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 55 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 58 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 59 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 12 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 60 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 11 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 5 | Czas: 66 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 11 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 67 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 10 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 77 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 9 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 82 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 8 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 84 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 86 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 8 | Czas: 91 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 9 | Czas: 92 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 10 | Czas: 97 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 10 | Czas: 102 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 11 | Czas: 104 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 12 | Czas: 108 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 1 | Czas: 109 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 3 | Czas: 111 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 113 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 116 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 116 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 12 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 118 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 11 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 5 | Czas: 124 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 11 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 125 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 10 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 134 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 9 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 139 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 8 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 141 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 144 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 8 | Czas: 148 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 9 | Czas: 149 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 10 | Czas: 154 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 10 | Czas: 158 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 11 | Czas: 161 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 12 | Czas: 165 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 1 | Czas: 166 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 3 | Czas: 168 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 170 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 173 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 173 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 12 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 175 |

☒ Autoscroll ☐ pokaż znacznik czasu

Nowa linia

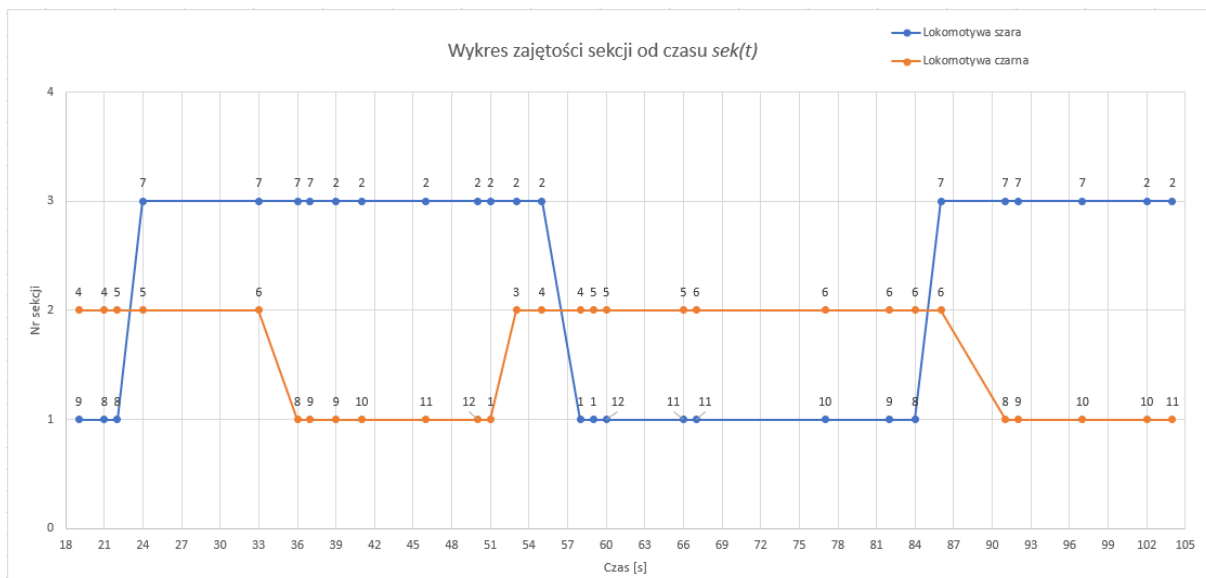
9600 baud

Wyczyść okno

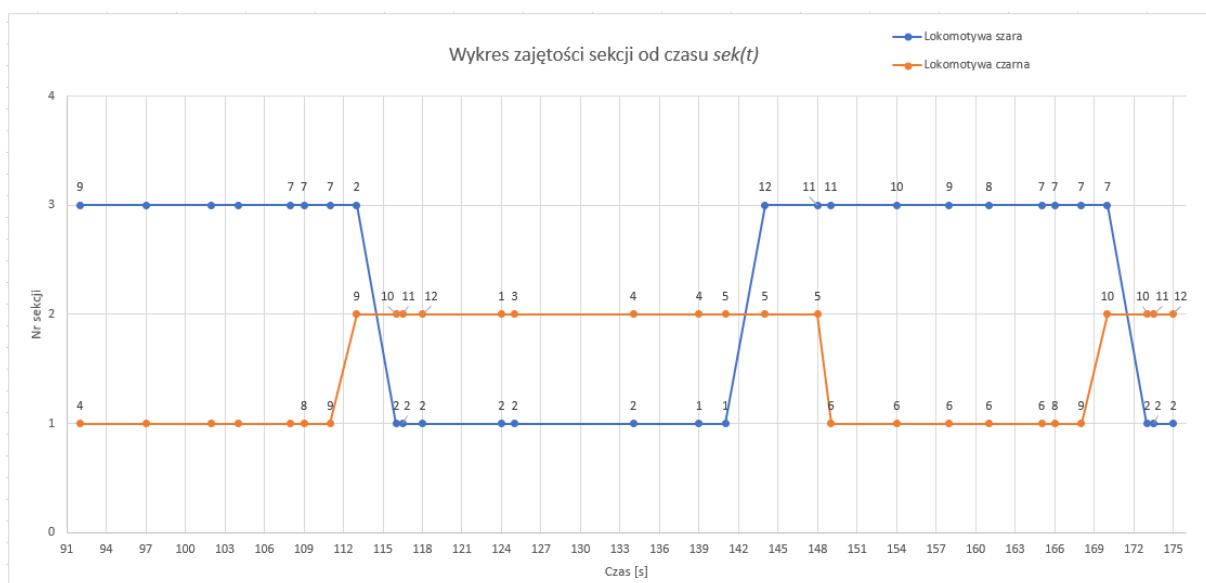
Rysunek 7.1 Monitoring stanu lokomotyw (badanie nr 1)

Przedstawiona na powyższym rysunku wartość prędkości lokomotywy ma jedynie charakter podglądowy, ponieważ nie reprezentuje ona żadnej fizycznej wielkości. W rzeczywistości jest to współczynnik wypełnienia impulsu PWM, który może przyjmować wartości od 0 do 255 (zmienna typu *byte*). Dokładne parametry fizyczne lokomotyw, takie jak maksymalna prędkość (wyrażona np. w cm/s) czy moc silnika [W], nie są znane. Jednakże ze względu na fakt, że prędkość obrotowa silnika zależy liniowo od napięcia zasilania, można założyć, że lokomotywa przy zadanej wartości PWM = 150 będzie poruszała się około trzy razy szybciej niż dla PWM = 50.

Na podstawie powyższego założenia oraz po przeanalizowaniu następujących po sobie stanów lokomotyw, można stwierdzić, że pociągi zwalniają, a następnie hamują w odpowiednich strefach postojowych zgodnie z założeniami. Dodatkowo sporządzono wykres (Rysunek 7.2 i 7.3), na którym przedstawiono status zajętości sekcji w czasie.



Rysunek 7.2 Wykres zajętości sekcji od czasu dla badania pierwszego 1/2



Rysunek 7.3 Wykres zajętości sekcji od czasu dla badania pierwszego 2/2

Po pomyślnym przeprowadzeniu pierwszego eksperymentu zdecydowano się na powtórzenie badania, ale dla innych warunków początkowych. Tym razem lokomotywę szarą umieszczono w sekcji trzeciej, czołem w stronę czujnika nr 2, natomiast lokomotywę czarną ulokowano w sekcji drugiej przed czujnikiem nr 6. Uzyskane wyniki przejazdu przedstawiono na rysunku 7.4.

COM6

Wyslij

ROZPOCZYNAJ PROGRAM

| | | | | | | |
|-----------------------|---------------|-------------|------------------------|---------------|-------------|-----------|
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 20 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 12 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 22 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 11 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 27 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 10 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 38 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 9 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 43 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 8 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 44 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 47 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 8 | Czas: 51 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 9 | Czas: 53 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 10 | Czas: 57 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 10 | Czas: 63 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 11 | Czas: 64 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 12 | Czas: 69 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 1 | Czas: 69 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 3 | Czas: 71 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 74 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 76 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 76 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 12 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 78 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 11 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 5 | Czas: 84 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 11 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 84 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 10 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 94 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 9 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 99 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 8 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 100 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 103 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 8 | Czas: 107 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 9 | Czas: 108 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 10 | Czas: 113 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 10 | Czas: 116 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 11 | Czas: 119 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 12 | Czas: 123 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 1 | Czas: 124 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 3 | Czas: 125 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 128 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 131 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 131 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 12 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 133 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 11 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 5 | Czas: 138 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 11 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 139 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 10 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 148 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 9 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 153 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 8 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 154 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 6 | Czas: 157 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 8 | Czas: 161 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 9 | Czas: 162 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 7 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 10 | Czas: 167 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 10 | Czas: 169 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 11 | Czas: 172 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 12 | Czas: 177 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 1 | Czas: 178 |
| Lokomotywa: 1 (szara) | Prędkość: 0 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 150 | Pozycja: 3 | Czas: 179 |
| Lokomotywa: 1 (szara) | Prędkość: 150 | Pozycja: 2 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 182 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 50 | Pozycja: 4 | Czas: 184 |
| Lokomotywa: 1 (szara) | Prędkość: 50 | Pozycja: 1 | Lokomotywa: 2 (czarna) | Prędkość: 0 | Pozycja: 5 | Czas: 184 |

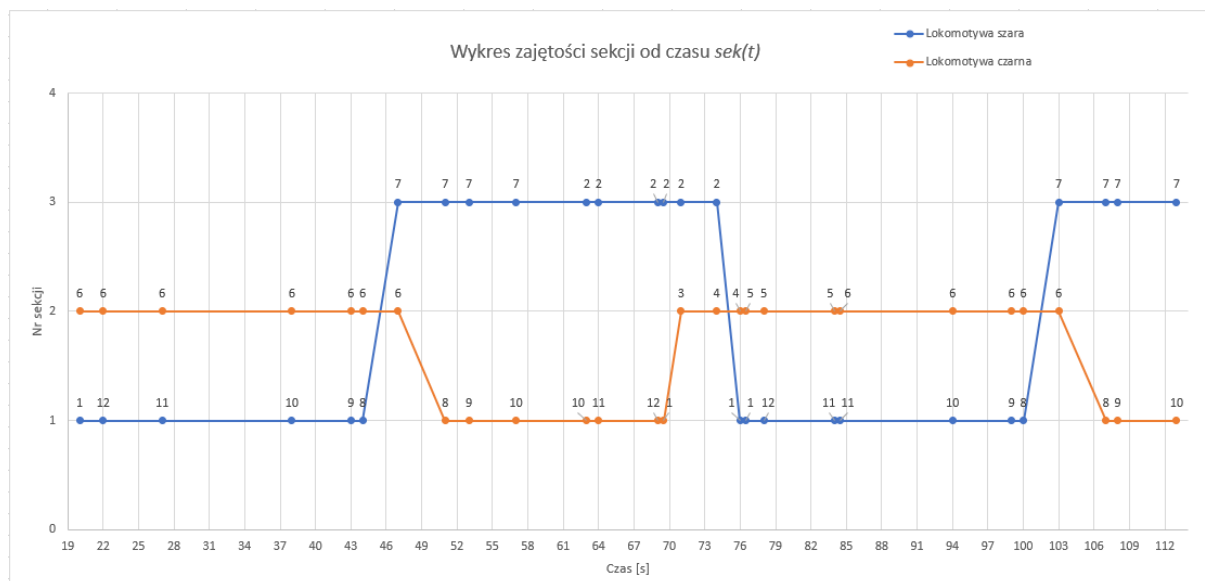
☒ Autoscroll☐ pokaż znacznik czasu

Nowa linia9600 baudWyczyść okno

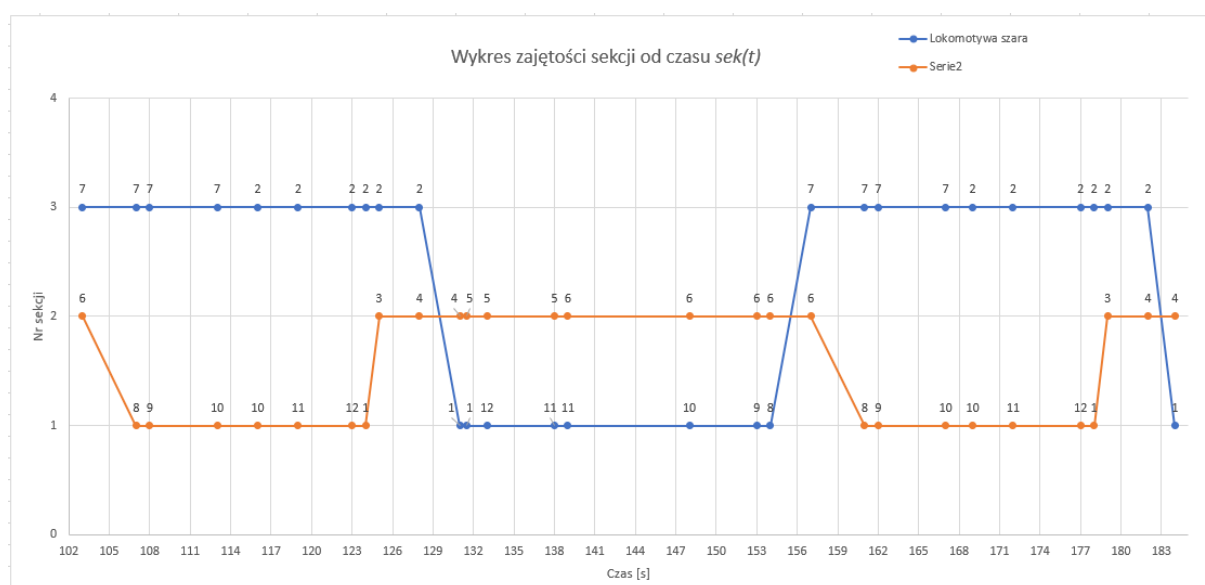
Rysunek 7.4 Monitoring stanu lokomotyw (badanie nr 2)

Również w tym wypadku po udanym określeniu pozycji początkowych, algorytm bezproblemowo przeszedł do metody sterowania i kontrolowania ruchu lokomotyw. Ze względu na specyfikę układu torowiska, lokomotywy zawsze dążą do ustalonego wzorca osiąganych kolejno pozycji. Jedyne różnice, jakie można zauważyć analizując wyniki obydwu badań, pojawiają się na samym początku działania algorytmu. Wystarczy natomiast jeden cykl przejazdowy, żeby pociągi powróciły do stanu zgodnego z poprzednimi eksperymentami.

Na rysunku 7.5 i 7.6 przedstawiono wykresy stanu zajętości sekcji od czasu, uzyskane w drugim badaniu.



Rysunek 7.5 Wykres zajętości sekcji od czasu dla badania drugiego 1/2



Rysunek 7.6 Wykres zajętości sekcji od czasu dla badania drugiego 2/2

Podsumowując, na podstawie wyników przeprowadzonych badań w zakresie funkcjonalności i zachowania układu, stwierdzono, że spełnione zostały założenia projektowe. Zgodnie z wykresami zajętości sekcji, ani razu nie doszło do sytuacji, w której obydwie lokomotywy znalazłyby się w tej samej sekcji torowej doprowadzając do kolizji. Ponadto pociągi zgodnie z oczekiwaniami zwalniają oraz zatrzymują się w wyznaczonych strefach. Algorytm auto-lokalizacji spełnia swoją funkcję bez względu na początkowe ustawienie lokomotyw, o ile spełnione jest założenie, że znajdują się one w różnych sekcjach.

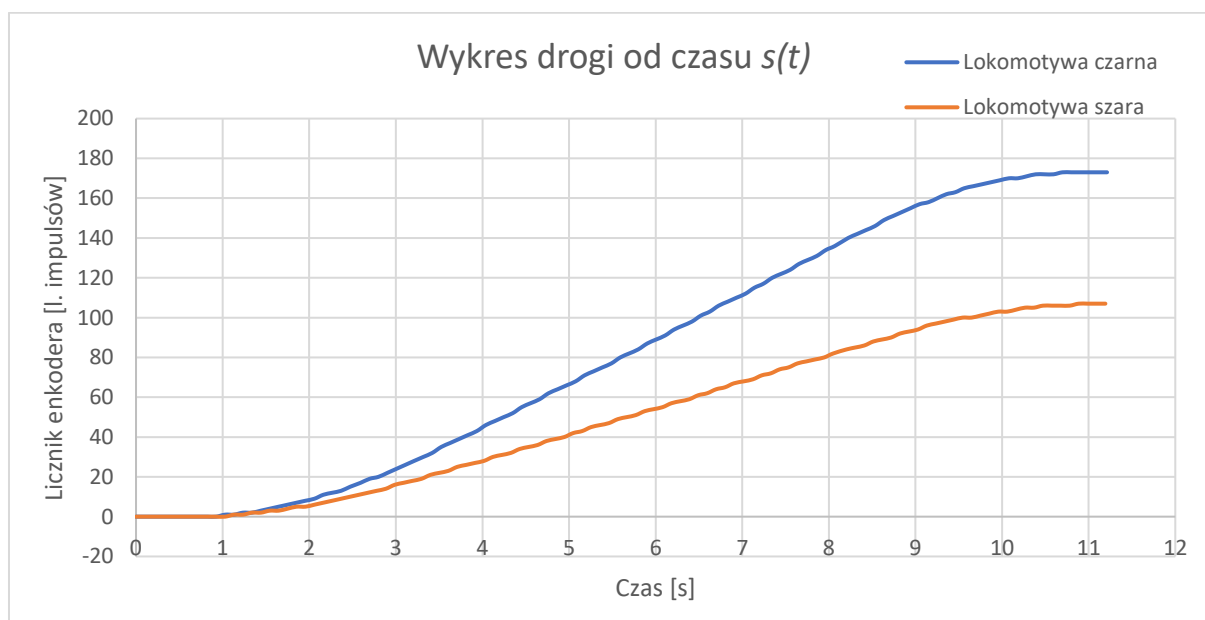
7.2. Badanie profilu prędkości lokomotyw

Zaprojektowany algorytm sterowania lokomotywami został opracowany przy założeniu, że lokomotywy mogą zmniejszać lub zwiększać swoją prędkość ze stałym przyśpieszeniem. W konsekwencji zastosowany został profil prędkości o charakterze trapezowym, który ze względu na brak obecności regulatora PID, w rzeczywistości może znacząco odbiegać od przebiegu prędkości zadanej.

Autor pracy zdecydował się zbadać rzeczywisty profil prędkości na podstawie zmian częstotliwości odczytu sygnałów z enkoderów. W tym celu zmodyfikował algorytm tak, żeby uzyskać następujące parametry jazdy:

- Przyśpieszenie lokomotywy: $\frac{5}{255}\%$ wypełnienia PWM co każde 100ms.
- Prędkość początkowa i końcowa: 0% wypełnienia PWM.
- Prędkość maksymalna: $\frac{150}{255}\%$ wypełnienia PWM.
- Czas jazdy z ustaloną prędkością: 5s.

Następnie uruchomił układ oraz zarejestrował kolejne odczyty z enkoderów w trakcie przejazdu obydwu lokomotyw. Uzyskane wyniki zostały poddane interpolacji wielomianowej, a następnie przedstawione na rysunku 7.7.



Rysunek 7.7 Wykres przebytej drogi przez lokomotywy dla zadanego profilu prędkości

Wyniki eksperymentu potwierdziły wcześniejsze przypuszczenia, że lokomotywa szara pomimo identycznych parametrów jazdy porusza się wolniej niż lokomotywa czarna.

Przyczyną takiej sytuacji są różnice w mocy silników poszczególnych pociągów, które prawdopodobnie należą do różnych serii produkcyjnych.

W celu sporządzenia wykresu prędkości, zastosowane zostały dwa sposoby prowadzenia obliczeń. Pierwszy z nich oparty został na metodzie szacowania przy pomocy różnic centralnych, opisanej poniższym wzorem [13]:

$$v_0 = \frac{\Delta s}{\Delta t} = \frac{s_1 - s_{-1}}{t_1 - t_{-1}}$$

gdzie s_1 – pomiar licznika enkodera z chwili następnej t_1

s_{-1} – pomiar licznika enkodera z chwili poprzedniej t_{-1}

Ze względu na fakt, że pomiar licznika enkodera oraz stopień wypełnienia impulsu PWM, nie są w żaden sposób powiązane fizycznie, otrzymane wyniki zostały odpowiednio przeskalowane. Wartość współczynnika przeskalowania dla każdej z lokomotyw została obliczona ze wzoru:

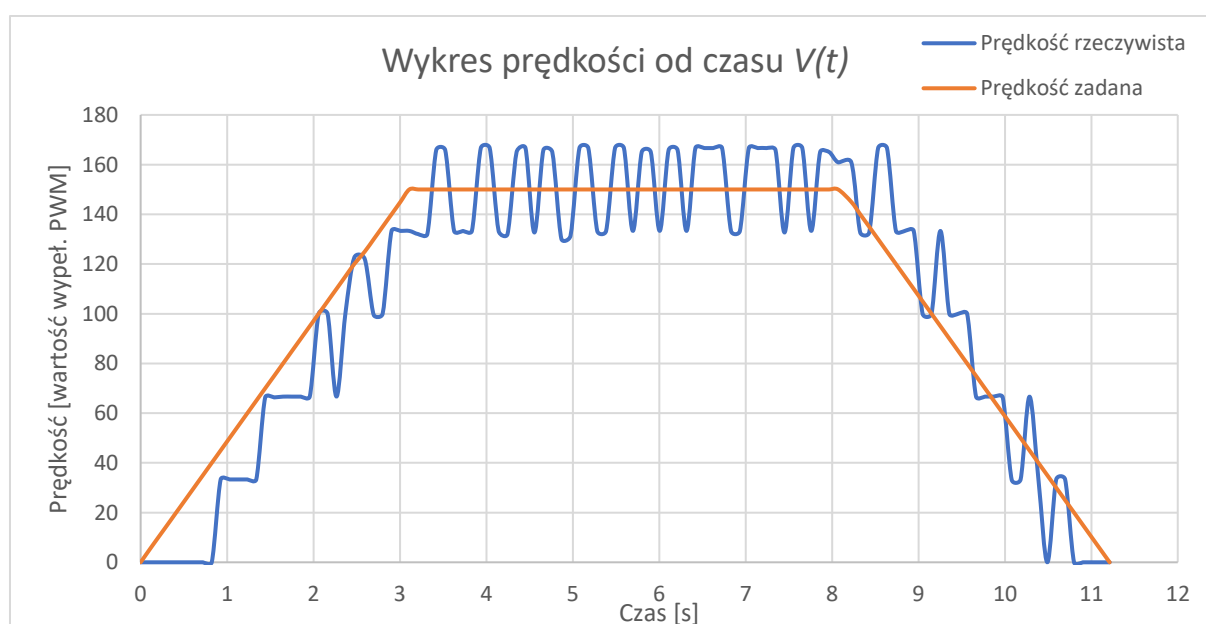
$$k = \frac{2 * v_{pwm}}{(v_{e_max} + v_{e_min})}$$

gdzie v_{pwm} – stopień wypełnienia PWM

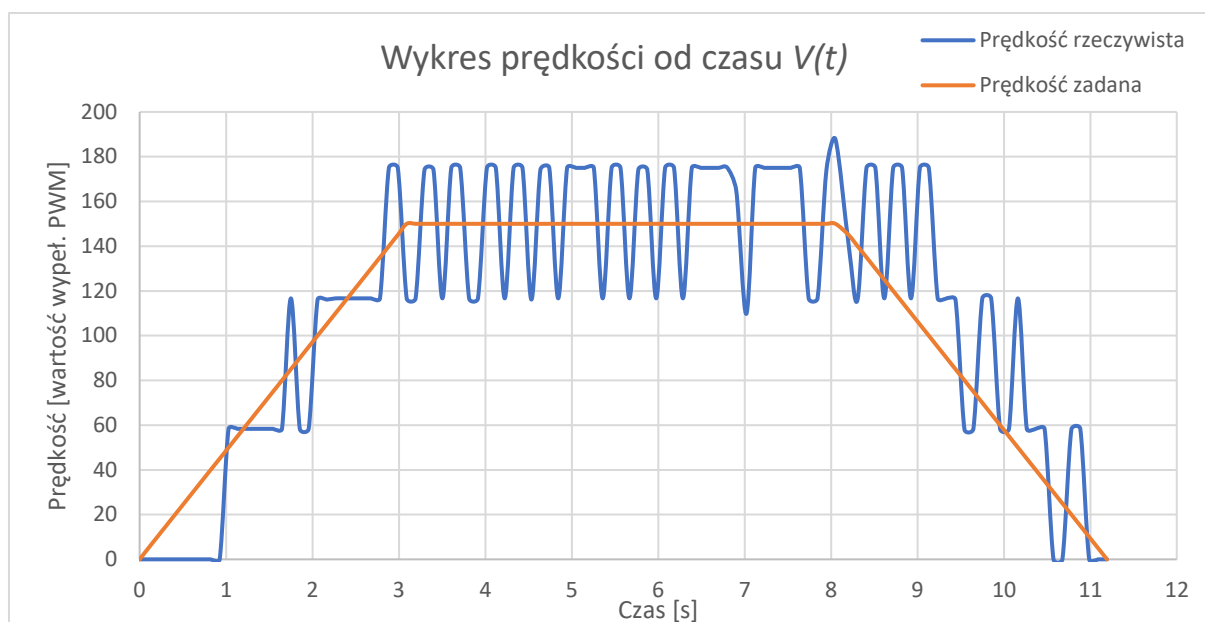
v_{e_min} – najmniejsza zarejestrowana wartość prędkości ustalonej

v_{e_max} – największa zarejestrowana wartość prędkości ustalonej

Uzyskane wyniki za pomocą metody różnic centralnych zostały przedstawione na poniższych wykresach.



Rysunek 7.8 Wykres prędkości uzyskany metodą różnic centralnych dla lokomotywy czarnej



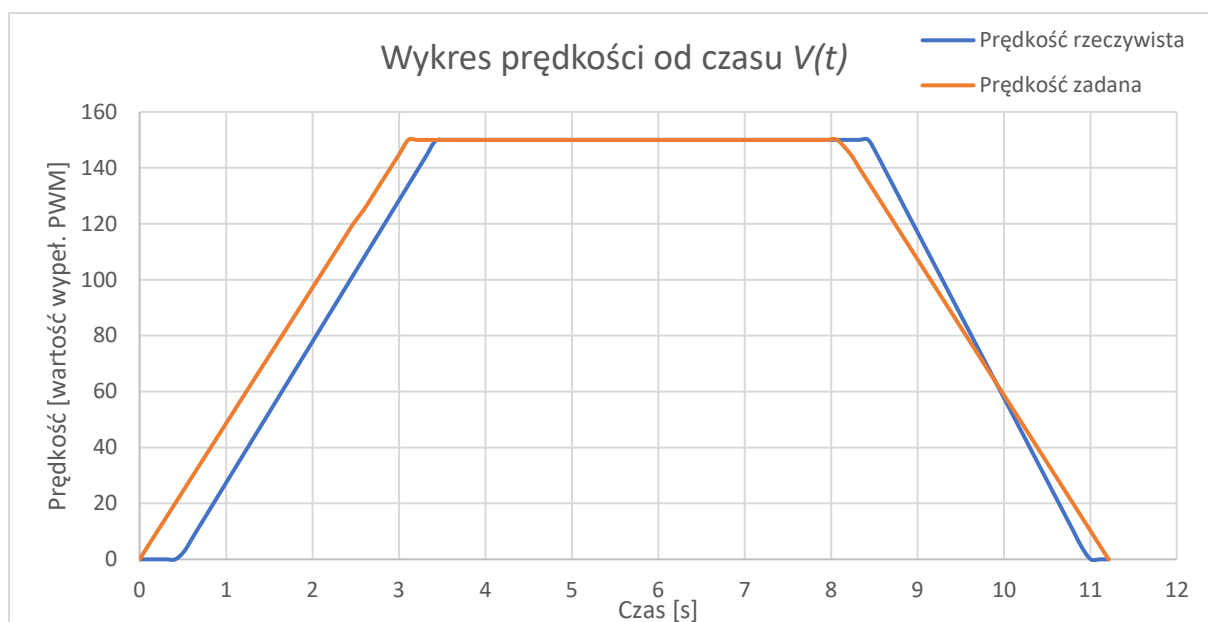
Rysunek 7.9 Wykres prędkości uzyskany metodą różnic centralnych dla lokomotywy szarej

Uzyskane profile prędkości cechują się dużą amplitudą niepewności pomiarowej. Amplituda ta jest tym większa, im większa jest wartość współczynnika przeskalowania, dlatego w przypadku lokomotywy szarej (która porusza się wolniej od lokomotywy czarnej) sięga ona prawie 40% maksymalnej wartości prędkości zadanej. Sama niepewność pomiarowa wynika z niskiej rozdzielczości enkodera, który przystosowany jest do współpracy z silnikami o dużej prędkości obrotowej. Widoczne jest to szczególnie w początkowej fazie wykresu, gdzie pierwsza zmiana licznika enkodera zarejestrowana została dopiero po upływie 1s od startu lokomotywy.

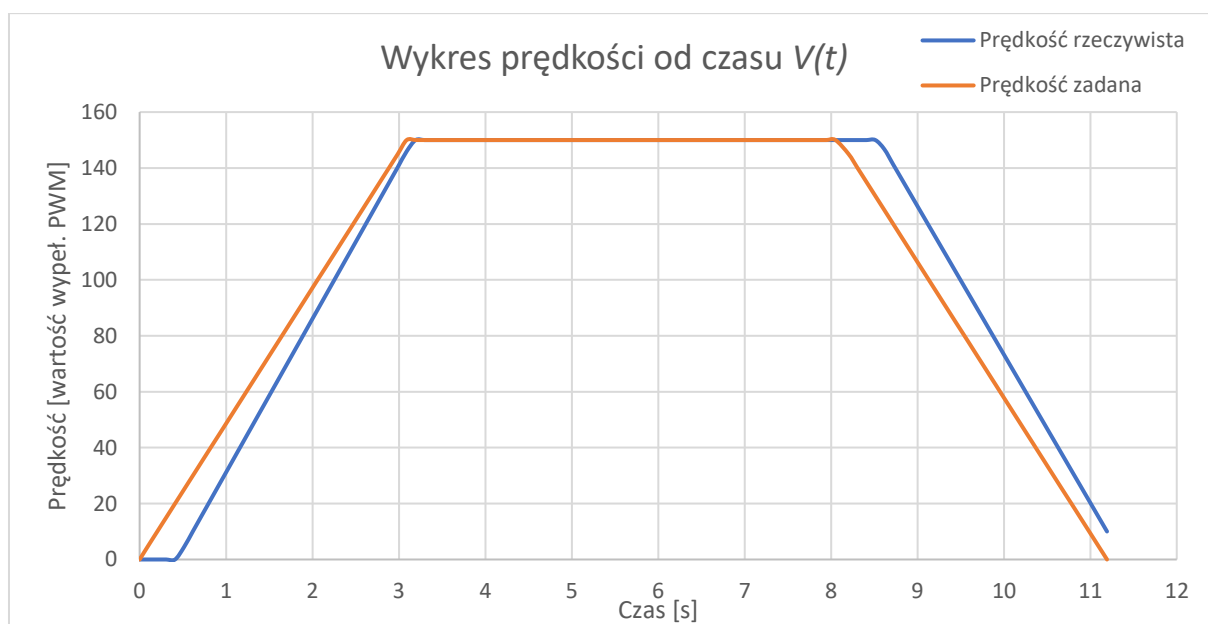
Ze względu na niską miarodajność wyników uzyskanych za pomocą różnic centralnych, zdecydowano się na użycie metody opartej na obliczaniu pochodnej za pośrednictwem funkcji aproksymującej. Podzielono zatem wykres Rysunek 7. na trzy niezależne części:

- 1) od 0s do 3,25s, kiedy lokomotywa przyśpiesza
- 2) od 3,25s do 8s, kiedy lokomotywa jedzie ze stałą prędkością
- 3) od 8s do 11s, kiedy lokomotywa zwalnia

Następnie 1) oraz 3) fragment wykresu aproksymowano niezależnymi funkcjami kwadratowymi, a fragment 2) funkcją liniową. Otrzymane funkcje zróżniczkowano i połączono ze sobą uwzględniając odpowiednie dziedziny czasowe (Rysunek 7.10 oraz 7.11). Wartości ujemne, występujące na początku oraz końcu przebiegu prędkości i wynikające z niedokładności funkcji aproksymujących, zastąpiono wartościami zerowymi.



Rysunek 7.10 Wykres prędkości uzyskany metodą pochodnej funkcji aproksymującej dla lokomotywy czarnej



Rysunek 7.11 Wykres prędkości uzyskany metodą pochodnej funkcji aproksymującej dla lokomotywy szarej

Na obydwu wykresach dostrzegalne jest przesunięcie fazowe profilu rzeczywistego względem zadanego. Na zaistniałą sytuację wpływa przede wszystkim występowanie dużego oporu toczenia lokomotywy, a także niska rozdzielczość enkodera, który zarejestrował moment startu z pewnym opóźnieniem. Ponadto zauważalne są różnice w kątach nachylenia fazy przyspieszania i hamowania. W rzeczywistości na przebieg rozruchu silnika w dużym stopniu wpływają straty mocy spowodowane tarciami oraz bezwładność obciążenia silnika. Otrzymane wykresy są obciążone wieloma błędami, na które wpływa konieczność stosowania interpolacji punktów pomiarowych oraz późniejsza aproksymacja funkcjami wielomianowymi

str. 62

w określonych przedziałach czasowych. W celu dokładnego zbadania profilu prędkości należałoby doposażyć układ w akcelerometr lub enkoder o wysokiej rozdzielczości, a także określić podstawowe parametry silników takie, jak moc i sprawność.

7.3. Badanie pomiarów przebytej drogi

Wykorzystane w projekcie lokomotywy wyposażone są w skrajne osie kołowe, które ułatwiają pokonywanie zakrętów, zmniejszając ryzyko wykolejenia. Jednakże dołączone do nich wagony, w których znajdują się układy sterowania oraz enkodery magnetyczne, posiadają sztywne osie kołowe, pozbawione możliwości obrotu względem osi normalnej do torowiska. W związku z tym, w momencie pokonywania przez wagon odcinka łukowego lub zakrętu, dochodzi do zjawiska poślizgu.

W niniejszym podrozdziale, autor pracy zbadał dokładność pomiaru przebytej przez pociąg drogi, zarejestrowanej za pomocą enkoderów magnetycznych, względem rzeczywistej drogi obliczonej na podstawie geometrii torowiska. Badania przeprowadzone zostały dla różnych odcinków torowych, uwzględniających łuki i zakręty, dla których mógł wystąpić efekt poślizgu.

Na podstawie informacji zawartych w tabelce w Torowisko PIKO A-Gleis, obliczono długość drogi przebytej przez pociąg w ramach jednego okrążenia:

$$L_{okr} = 2 * (2 * L_{G231} + L_{239} + L_{WL}) + 2 * \pi * R_{R2} \approx 4530 \text{ cm}$$

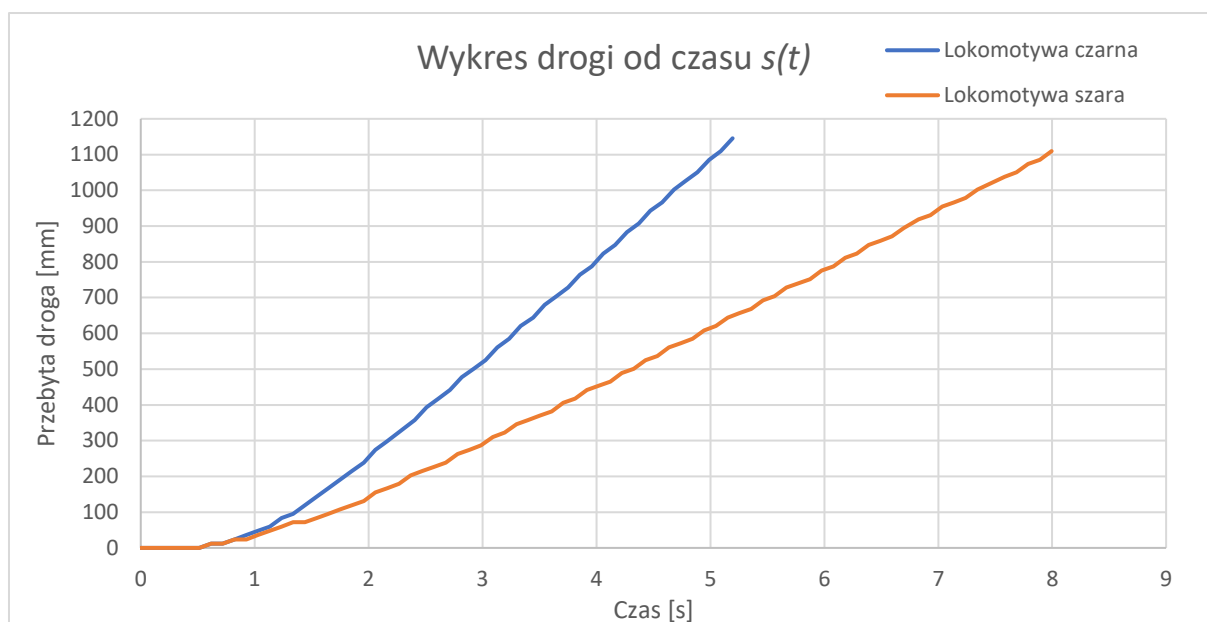
Średnica koła wagonu jest równa $R = 5,7 \text{ mm}$, zatem co każdy jeden pełen obrót koła, pociąg przemieszcza się o odległość:

$$L_{jedn} = 2 * \pi * R = 35,8 \text{ mm}$$

Rozdzielczość enkodera wynosi $x = 12 \text{ imp/obr}$, jednakże programowo jego licznik jest aktualizowany tylko wtedy, kiedy na kanale A wykryte zostanie zbocze opadające, czyli co każdy czwarty impuls. W związku z opisaną sytuacją można przyjąć, że w czasie pomiędzy kolejnymi pomiarami enkodera, pociąg pokonuje drogę równą:

$$L_{enk} = \frac{L_{jedn} * 4}{x} = 11,933 \text{ mm}$$

W pierwszym badaniu autor przeprowadził pomiary przejazdu lokomotywy czarnej i szarej na odcinku pomiędzy czujnikami nr 10 i 11 dla prędkości maksymalnej PWM = 150 (Rysunek 7.12). Następnie przy tej samej prędkości maksymalnej zbadał zarejestrowaną przez pociąg drogę dla całego okrążenia (Rysunek 7.13).



Rysunek 7.12 Wykres zarejestrowanej przez enkodery drogi na odcinku między czujnikami 10 i 11

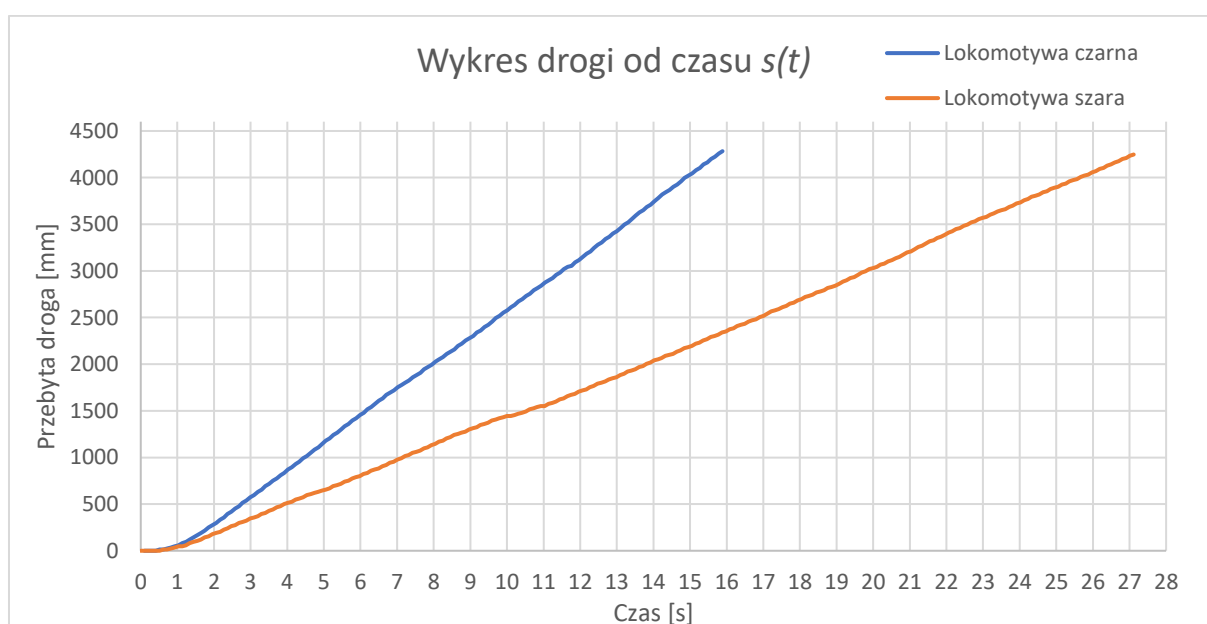
Rzeczywista odległość na łuku pomiędzy czujnikami 10 i 11 wynosi:

$$L_{10-11} = \frac{162}{360} * 2 * \pi * R = 1192,57 \text{ cm}$$

Odległości zarejestrowane przez lokomotywy wynoszą: $L_c = 1133,5 \text{ cm}$ i $L_s = 1110 \text{ cm}$

Otrzymany błąd względny przyjmuje wartość:

$$\delta_c = \frac{1192,57 - 1133,5}{1192,57} * 100\% = 4,95\%, \quad \delta_s = \frac{1192,57 - 1110}{1192,57} * 100\% = 6,92\%$$



Rysunek 7.13 Wykres zarejestrowanej przez enkodery drogi dla pełnego okrążenia

Pomiary z enkoderów dla całego okrążenia wynoszą:

$$L_c = 4283,6 \text{ cm} \quad i \quad L_s = 4247,8 \text{ cm}$$

Natomiast obliczony błąd względny wynosi:

$$\delta_c = \frac{4530 - 4283,6}{4530} * 100\% = 5,44\%, \quad \delta_s = \frac{4530 - 4247,8}{4530} * 100\% = 6,23\%$$

Obserwując wyniki przedstawione na obydwu wykresach można zauważyć, że wartość błędu względnego dla lokomotywy szarej oraz czarnej utrzymuje się na podobnym poziomie, niezależnie od przejechanej odległości. Oznacza to, że zjawisko poślizgu powoduje liniowe narastanie błędu, co pozwala w prosty sposób zamodelować funkcję korygującą pomiar licznika enkodera.

Zauważalne są jednak rozbieżności w wartościach błędu względnego dla obydwu lokomotyw na krótkim odcinku pomiarowym. Czynniki, które mogą mieć wpływ na uzyskany wynik są różne położenia enkoderów w momencie startu oraz odmienna charakterystyka przyspieszania lokomotyw.

Podsumowując, pomiar przebytej drogi za pomocą licznika enkoderów nie jest wystarczająco dokładny, aby stanowić jedyną metodę lokalizacji lokomotyw na makiecie. Błąd bezwzględny narasta liniowo, co w konsekwencji powoduje coraz większe rozbieżności względem rzeczywistych pozycji wraz z pokonaną drogą. W celu zwiększenia skuteczności wyznaczania przebytej drogi przy pomocy enkoderów, należy zamodelować i uwzględnić funkcję korekcyjną, a także co pewien czas resetować wartość błędu np. poprzez odczyt rzeczywistej pozycji lokomotywy za pomocą czujników zajętości toru.

8. Wnioski oraz propozycje dalszego rozwoju pracy

W niniejszej pracy autor zaprezentował istniejące systemy sterowania w rzeczywistym oraz modelarskim ruchu kolejowym. Następnie samodzielnie zaprojektował i zaprogramował układ sterowania dwoma lokomotywami wykorzystujący komunikację radiową. Na koniec przeprowadził badania pod względem funkcjonalności oraz zachowania układu.

Na etapie planowania układu sterowania autor zdecydował, że komunikacja pomiędzy centralą nadzorująco-sterującą, a lokomotywami, będzie prowadzona drogą radiową. Jednakże w trakcie prac nad algorytmem napotkano wiele trudności oraz przeszkód, które wymusiły zmiany w strukturze samej komunikacji, ograniczając w dużym stopniu częstotliwość wymiany informacji. Uzyskany efekt pozwolił ustanowić stabilną pracę modułów radiowych, jednakże nie udało się całkowicie wyeliminować problemów z otrzymywaniem odpowiedzi zwrotnych. Znacznie bardziej niezawodną formą wymiany danych byłoby skorzystanie z popularnych i dopracowanych standardów komunikacji bezprzewodowej, takich jak WiFi lub Bluetooth.

Po opracowaniu algorytmu przystąpiono do badań mających na celu sprawdzenie spełnienia przyjętych założeń. Przeprowadzono dwa eksperymenty, podczas których lokomotywy ruszały z różnych pozycji startowych, najpierw określając swoją lokalizację początkową, a następnie przechodząc do regularnej jazdy. Specyfika torowiska powodowała, że po wykonaniu początkowego cyklu ustalającego, lokomotywy zaczynały jeździć według pewnego niezmiennego schematu. Jednakże pociągi zgodnie założeniami ani razu nie znalazły się w tym samych sekcjach oraz zwalniały i zatrzymywały się według ustalonego schematu zachowań.

Algorytm został zatem zaprojektowany prawidłowo, jednakże w obecnej formie stanowi zaledwie podstawę systemu automatycznego sterowania i nie nadaje się do wprowadzenia do rzeczywistych układów SRK. W projekcie wykorzystane zostały jedynie podstawowe elementy torowiska, natomiast prawdziwe układy stacyjne są znacznie bardziej skomplikowane i wymagające. W celu przystosowania algorytmu do współpracy z układami SRK należy przebudować zaprojektowane torowisko, wzorując się na dowolnej, istniejącej stacji lub posterunku kolejowym. Ponadto należy zadbać o kwestie bezpieczeństwa wprowadzając urządzenia ochrony bocznej oraz zapewniając stabilną i bezawaryjną komunikację bezprzewodową.

Na podstawie pomiaru częstotliwości odczytów z enkoderów sporządzono wykresy profili prędkości lokomotyw. Wykorzystano dwie metody obliczeniowe, jednak ze względu na błędy pomiarowe i przybliżenia, nie udało się uzyskać w pełni miarodajnych wyników. Zastosowane enkodery posiadają zbyt niską rozdzielczość, która okazała się niewystarczająca do przeprowadzenia kompletnych obliczeń. Stwierdzono, że w celu poprawy jakości sterowania można zaimplementować w algorytmie regulator PID. Wiąże się to jednak z koniecznością montażu akcelerometru lub enkoderów o wyższej rozdzielczości.

W ramach badań na enkoderach przeprowadzono również analizę odczytów licznika przebytej drogi. Stwierdzono, że zjawisko poślizgu, występujące pomiędzy kołami wagonów, a szynami, powoduje powstawanie błędów pomiarowych, pomiędzy drogą zmierzoną, a rzeczywistą. Błąd ten wzrastał liniowo wraz z pokonywaną przez lokomotywę drogą, co powodowało coraz większe rozbieżności pomiędzy pozycją teoretyczną, a rzeczywistą. Pomiar licznika enkodera nie może zatem stanowić samodzielnej metody wyznaczania położenia lokomotyw. Należy stosować dodatkowo inne metody pomiarowe, które pozwolą co pewien czas skorygować lub całkowicie wyzerować błąd położenia.

Spis rysunków

| | |
|---|----|
| Rysunek 1.1 Model pierwszej lokomotywy George’a Stephensona - „Rakiety” | 8 |
| Rysunek 2.1 Klasyfikacja urządzeń SRK pod względem funkcjonalności [5] | 11 |
| Rysunek 2.2 Obwód odcinkowy | 12 |
| Rysunek 2.3 Obwód grupowy..... | 13 |
| Rysunek 2.4 Obwody pociągowe | 14 |
| Rysunek 2.5 Obwody zwrotnicowe | 15 |
| Rysunek 2.6 Przykładowy schemat systemu DCC [8] | 17 |
| Rysunek 2.7 Kodowanie bitów z układach DCC [9] | 18 |
| Rysunek 2.8 Przykładowy pakiet danych w systemach DCC [10] | 20 |
| Rysunek 2.9 Tabela kodowania prędkości lokomotywy [10] | 21 |
| Rysunek 4.1 Model torowiska PIKO | 25 |
| Rysunek 4.2 Schemat elektryczny lokomotywy PIKO BR 218 | 27 |
| Rysunek 4.3 Zdjęcie makiety wykorzystanej w projekcie..... | 28 |
| Rysunek 5.1 Arduino Leonardo | 30 |
| Rysunek 5.2 Arduino Pro Mini..... | 31 |
| Rysunek 5.3 Moduł radiowy | 31 |
| Rysunek 5.4 Stabilizator liniowy..... | 32 |
| Rysunek 5.5 Sterownik silnikowy | 32 |
| Rysunek 5.6 Czujnik odbiciowy | 33 |
| Rysunek 5.7 Ekspander wyprowadzeń | 33 |
| Rysunek 5.8 Enkodery magnetyczne Pololu..... | 34 |
| Rysunek 5.9 Akumulator Li-Pol..... | 34 |
| Rysunek 5.10 Moduł przekaźnikowy | 34 |
| Rysunek 5.11 Schemat elektryczny układu sterowania lokomotywą | 37 |
| Rysunek 5.12 Rozmieszczenie czujników odbiciowych na makiecie..... | 38 |
| Rysunek 5.13 Schemat elektryczny układu centrali sterującej | 39 |
| Rysunek 6.1 Środowisko programistyczne Arduino | 40 |
| Rysunek 6.2 Struktura algorytmu centrali nadzorująco-sterującej..... | 44 |
| Rysunek 6.3 Schemat blokowy algorytmu kontroli ruchu kolejowego 1/2 | 49 |

| | |
|---|----|
| Rysunek 6.4 Schemat blokowy algorytmu kontroli ruchu kolejowego 2/2 | 50 |
| Rysunek 6.5 Struktura algorytmu sterowania lokomotywą..... | 51 |
| Rysunek 6.6 Schemat blokowy algorytmu sterowania lokomotywą | 52 |
| Rysunek 7.1 Monitoring stanu lokomotyw (badanie nr 1)..... | 55 |
| Rysunek 7.2 Wykres zajętości sekcji od czasu dla badania pierwszego 1/2..... | 56 |
| Rysunek 7.3 Wykres zajętości sekcji od czasu dla badania pierwszego 2/2..... | 56 |
| Rysunek 7.4 Monitoring stanu lokomotyw (badanie nr 2)..... | 57 |
| Rysunek 7.5 Wykres zajętości sekcji od czasu dla badania drugiego 1/2..... | 58 |
| Rysunek 7.6 Wykres zajętości sekcji od czasu dla badania drugiego 2/2..... | 58 |
| Rysunek 7.7 Wykres przebytej drogi przez lokomotywy dla zadanego profilu prędkości | 59 |
| Rysunek 7.8 Wykres prędkości uzyskany metodą różnic centralnych dla lokomotywy czarnej | 60 |
| Rysunek 7.9 Wykres prędkości uzyskany metodą różnic centralnych dla lokomotywy szarej | 61 |
| Rysunek 7.10 Wykres prędkości uzyskany metodą pochodnej funkcji aproksymującej dla lokomotywy czarnej | 62 |
| Rysunek 7.11 Wykres prędkości uzyskany metodą pochodnej funkcji aproksymującej dla lokomotywy szarej..... | 62 |
| Rysunek 7.12 Wykres zarejestrowanej przez enkodery drogi na odcinku między czujnikami 10 i 11.. | 64 |
| Rysunek 7.13 Wykres zarejestrowanej przez enkodery drogi dla pełnego okrążenia | 64 |

Bibliografia

- [1] Antoni Bochen (2016), Narodziny parowozu, [online], dostęp 07.02.2019], Dostępny w Internecie: http://www.kolejnictwo-polskie.pl/default_295.html
- [2] Antoni Bochen (2016), Geneza szybkich pociągów, [online], [dostęp 07.02.2019], Dostępny w Internecie: http://www.kolejnictwo-polskie.pl/default_299.html
- [3] Dąbrowa-Bajon M., Podstawy sterowania ruchem kolejowym. Funkcje, wymagania, zarys techniki, wyd. 3 popr., Warszawa, Oficyna Wydawcza Politechniki Warszawskiej, 2014, ISBN: 978-83-7814-320-8
- [4] Dyduch J., Kornaszewski M., Systemy sterowania ruchem kolejowym, wyd. 1, Radom, Wydawnictwo Politechniki Radomskiej, 2003, ISBN 83-7351-131-8
- [5] Paweł Okrzesik (2016), Urządzenia sterowania ruchem kolejowym, [online], [dostęp 05.02.2019], Dostępny w Internecie: <https://www.bsk.isdr.pl/usrk.php>
- [6] Obwody torowe, [online], Portal miłośników DCC i małych kolei [portal internetowy], 2007, [dostęp 18.02.2019], Dostępny w Internecie: <http://wiki.gbbkolejka.pl/tiki-index.php?page=Obwody+torowe>
- [7] Karol Marcińczak (2012), Teoria DCC, [online], [dostęp 20.02.2019], Dostępny w Internecie: <https://www.dcc24.eu/node/3>
- [8] Artur Schwartz (2019), DCC, [online], [dostęp 21.02.2019], Dostępny w Internecie: <http://as.rumia.edu.pl/tt/DCC/>
- [9] S-9.1, 2006, Electrical Standards for Digital Command Control, National Model Railroad Association, Inc.
- [10] S-9.2, 2004, Communications Standards For Digital Command Control, All Scales, National Model Railroad Association, Inc.
- [11] Dokumentacja techniczna regulatora PIKO #55003, [online], [dostęp 11.03.2019], Dostępny w Internecie: <https://www.piko-shop.de/is.php?id=16720>
- [12] Dokumentacja techniczna lokomotywy PIKO BR 218, [online], [dostęp 10.03.2019], Dostępny w Internecie: <https://www.piko-shop.de/is.php?id=3279>
- [13] Gonet M., Excel w obliczeniach naukowych i inżynierskich, wyd. II, Helion, 2011 ISBN: 978-83-246-3066-0