



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

MAŁY
INŻYNIER[®].PL

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



 **HIGH
TECHNIKA**

**Program zajęć
z techniki**

Robotyka

Podręcznik ucznia



**Autor programu zajęć:
mgr inż. Anna Fularz**



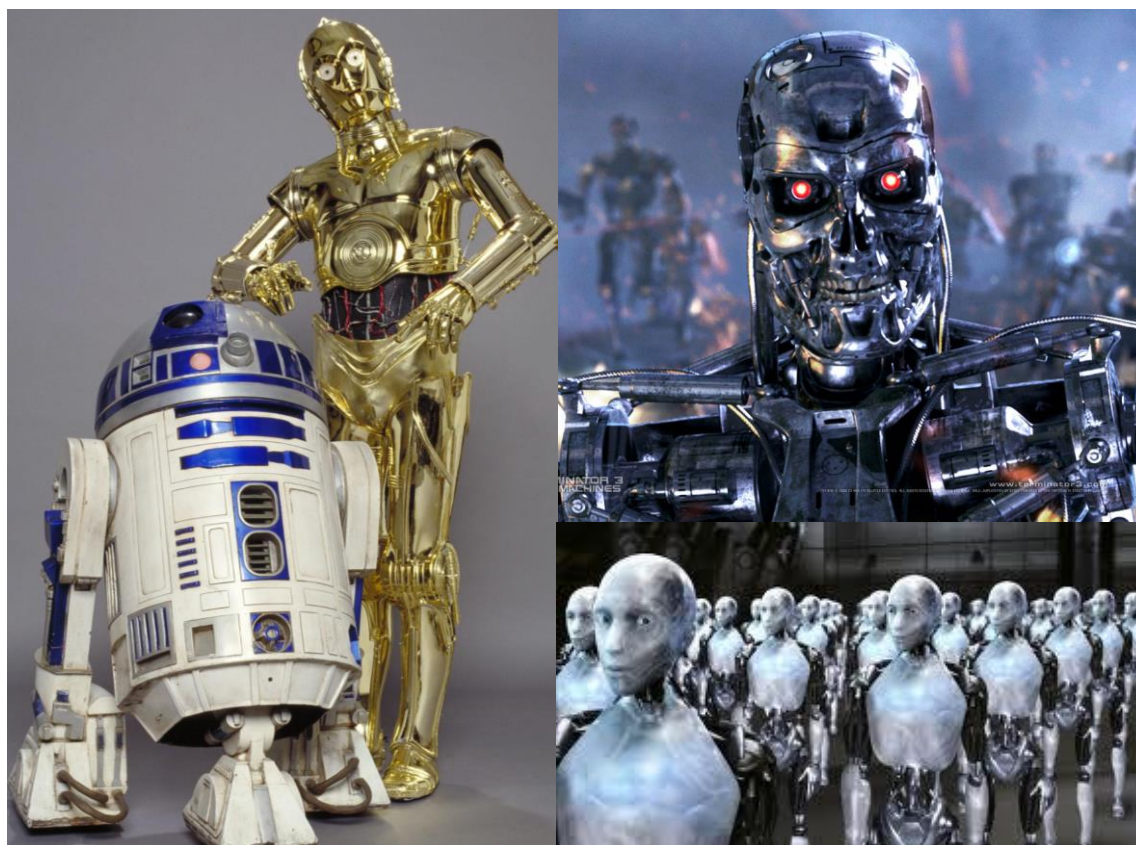
ROBOTYKA – CO TO WŁAŚCIWIE JEST?	2
PIERWSZA KONSTRUKCJA I PROGRAMOWANIE	15
ROBOT SPRZĄTAJĄCY	23
ROBOT WYŚCIGÓWKA	28
LINE FOLLOWER	33
ROBOT PRZEMYSŁOWY	36
LABIRYNT	43
POZYTYWKA	47
SUMO	52
SORTOWNICA	57
ROBOSIŁACZ	61
ROBOZWIERZAKI	64
JEŹDŹĄCY PO SZNURKU	69
ŻURAW	72
SKRĘTNE KOŁA	79



Robotyka – co to właściwie jest?

Czym jest robot?

Pierwszym skojarzeniem, jakie przychodzi na myśl na hasło Robotyka są roboty. Najczęściej przyjmują one postać maszyny przypominającej kształtem człowieka. Przykładem może być Terminator, androidy



z Gwiezdných Wojen, czy Transformersy. Współcześnie roboty to już nie tylko wytwór wyobraźni twórców fantastyki naukowej (science fiction). Codziennie stykamy się z robotami. Mają one różną postać, zależną od funkcji, do jakich są przeznaczone.

Najczęściej spotykane roboty mają wykonywać zadania bardzo niebezpieczne, monotonne (powtarzalne), czy uciążliwe. Można je spotkać w fabrykach samochodów, przy wykonywaniu operacji chirurgicznych, na liniach produkcyjnych czy w misjach kosmicznych. Niektóre roboty pomagają poznawać świat, docierając do miejsc niebezpiecznych dla człowieka takich, jak np. głębiny mórz i oceanów czy kosmos. Natomiast roboty-zabawki cieszą się dużą popularnością wśród dzieci i młodzieży.

Ale czym właściwie jest robot?

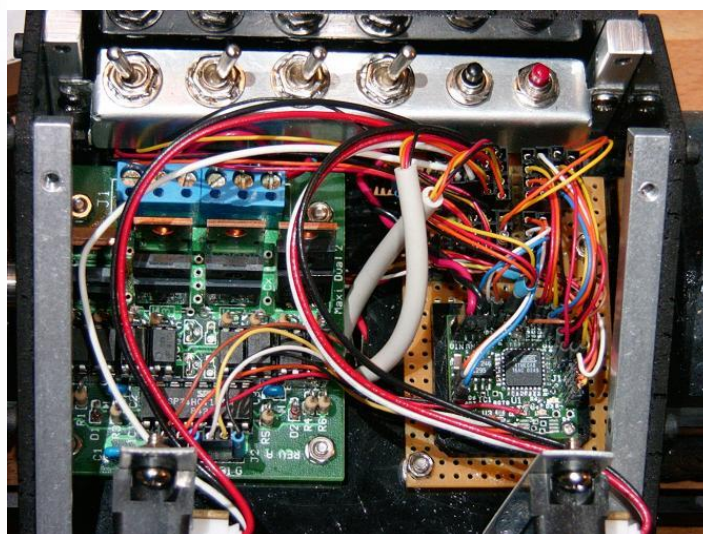
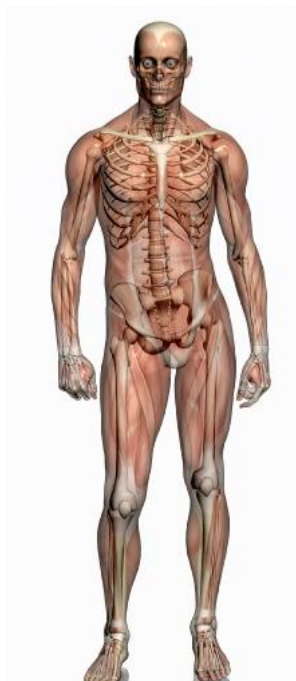
Słowo **Robot** jest stosunkowo nowym słowem. Na początku XX wieku zostało ono wykorzystane i spopularyzowane przez czeskiego pisarza Karelá Čapeka. Pochodzi od słowiańskiego słowa *robota*, oznaczającego ciężką pracę, wysiłek. W 1942 Isaac Asimov wprowadził termin Robotyka oraz stworzył trzy podstawowe jej prawa.

Prawa robotyki

- I. Robot nie może skrzywdzić człowieka, ani przez zaniechanie działania dopuścić, aby człowiek doznał krzywdy.
- II. Robot musi być posłuszny rozkazom człowieka, chyba że stoją one w sprzeczności z Pierwszym Prawem.
- III. Robot musi chronić sam siebie, jeśli tylko nie stoi to w sprzeczności z Pierwszym lub Drugim Prawem.

Robot - Mechaniczne urządzenie wykonujące automatycznie pewne zadania.

Budowa robota



Budowę robota można porównać do ciała człowieka. Podstawowym elementem budowy ciała człowieka jest **szkielet**. Stanowi on miejsce przyczepu mięśni, nadaje kształt ciału oraz służy do osłony narządów. W budowie robota tę samą funkcję stanowi **konstrukcja mechaniczna**. **Mięśniami** robota są **silniki**, zwane również aktuatorami. Jednym z ważniejszych elementów ciała człowieka jest **serce i układ krwionośny**. U robota funkcje odpowiedzialne za „żywołność” organizmu pełni **zasilanie**, które musi być doprowadzone do wszystkich elementów. Aby człowiek mógł się bezproblemowo poruszać potrzebuje **narządów zmysłów**. To one umożliwiają wykrycie określonych zmian zachodzących w otoczeniu i powodują odpowiednią reakcję organizmu. W przypadku robota funkcję tę pełnią **czujniki** zwane sensorami. Najważniejszym elementem budowy ciała człowieka, odpowiedzialnym za funkcje decyzyjne jest **mózg**. U robota za decyzje odpowiada **komputer** lub układ mikroprocesorowy.

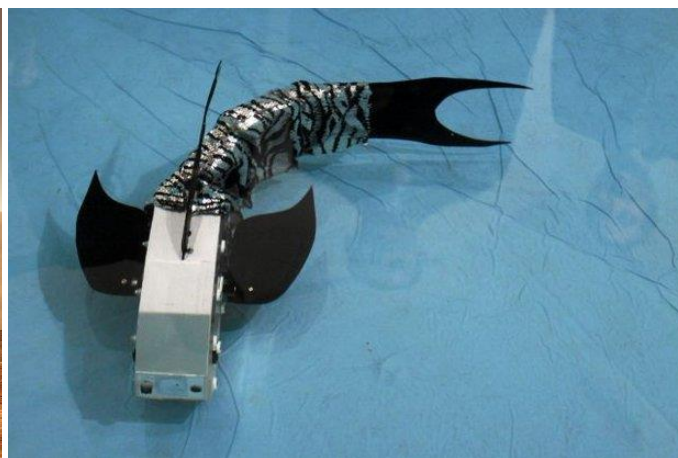
Rodzaje robotów

Większość ludzi zapytana o opis robota odpowie, że wygląda on jak człowiek. Co ciekawe, robot, który swym kształtem przypomina człowieka jest prawdopodobnie najtrudniejszy do skonstruowania. Robot musi być przede wszystkim funkcjonalny i zaprojektowany tak, aby realizował określone zadania. To właśnie od tego, zależy jego wielkość, ciężar który będzie w stanie przenieść lub sposób przemieszczania. Zadanie, do którego przeznaczony jest robot determinuje zatem jego budowę. W zależności od obranego kryterium można wyróżnić bardzo wiele typów robotów.

Ze względu na **sposób poruszania** wyróżnia się:

- Roboty mobilne
- Roboty stacjonarne

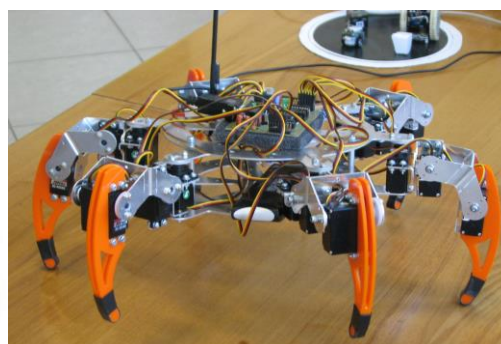
Roboty mobilne



Roboty mobilne cechuje zdolność do poruszania się – mogą pływać, jeździć lub latać. Wykorzystuje się je do przeszukiwania różnorodnych terenów. Roboty mobilne są nieopisaną pomocą dla ofiar zawałonych budynków. Wykorzystywane są do zadań, w których ludzie nie daliby sobie rady – gdy jest zbyt niebezpiecznie lub przestrzeń są zbyt małe, by człowiek mógł się do nich dostać. Przykładem takiego robota może być Mars Explorer – zaprojektowany do poruszania się po powierzchni Marsa.

Roboty mobilne można podzielić na podkategorie:

Roboty kołowe – do poruszania się wykorzystują koła. Ten typ robotów przemieszcza się najszybciej i najłatwiej mu przeszukiwać środowisko. Jednak są one przydatne tylko na płaskich terenach, nie radzą sobie na kamienistych podłożach.

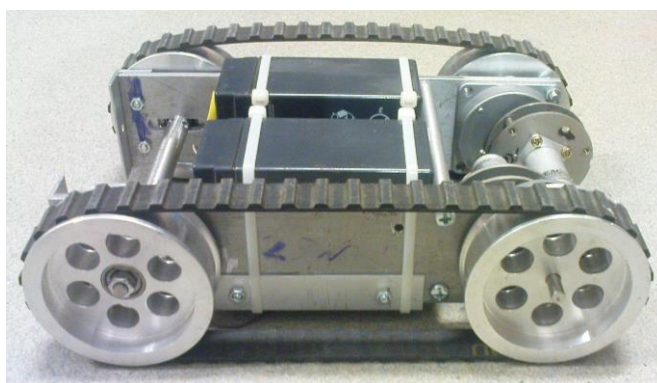




Roboty kroczące – do poruszania się wykorzystują „nogi”. Najlepiej sprawdzają się na terenach, w których roboty kołowe zawiodą, np. tereny skaliste. Roboty te muszą zachować równowagę w trakcie poruszania, aby nie ulec upadkowi. Dlatego większość robotów kroczących ma co najmniej 4 kończyny. Konstrukcje tego typu robotów wzorowane są na owadach lub rakach.

Roboty na gąsienicach – do przemieszczania wykorzystują gąsienice. Mogą poruszać się po terenach płaskich, wyboistych, a nawet schodach, dlatego znajdują one zastosowanie w robotach, np. przy strzeżeniu granic, poszukujących ładunków wybuchowych.

Od kilku lat w większych miastach Polski (Poznań, Warszawa, Wrocław, itd.) organizowane są zawody robotów. Gromadzą one zarówno zapalonych konstruktorów – hobbystów z ludźmi zainteresowanymi elektronicznymi nowinkami. Można na nich obejrzeć z bliska różne roboty – od tych wielkości 1cm x 1cm x 1cm, przez roboty kołowe, na gąsienicach aż po roboty kroczące.



Roboty stacjonarne

Roboty stacjonarne przeznaczone są do wykonywania zadań powtarzalnych, wymagających dokładności, które należy wykonywać cały dzień i całą noc, bez narzekania i skarg. Roboty z opisaną funkcjonalnością najczęściej można spotkać w przemyśle, są to tzw. roboty przemysłowe. Swoją budową przypominają rękę. Są tak skonstruowane, że mogą wykonywać różnorodne prace, a jedynym, co należy w nich zmienić, jest program.



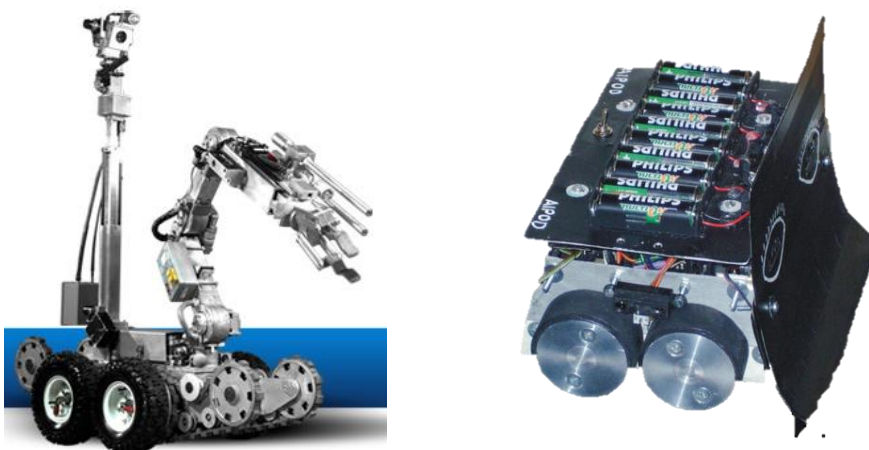
Ze względu na **rodzaj sterowania** wyróżnia się:

- Roboty autonomiczne
- Roboty zdalnie sterowane



Roboty autonomiczne

Roboty autonomiczne to roboty samowystarczalne, tzn. decyzje za wykonywane akcje podejmuje ich „mózg”, czyli komputer lub układ mikroprocesorowy. Roboty te działają zgodnie z wgranym programem, który opisuje zachowanie robota w konkretnych sytuacjach. Robot na podstawie odczytów z czujników sam podejmuje wszystkie decyzje. W niektórych przypadkach roboty autonomiczne mogą uczyć się nowych zachowań, np. omijać przeszkody, chodzić po trudnym terenie.



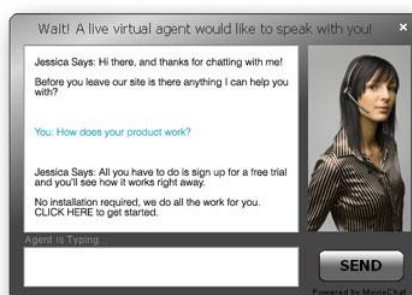
Roboty zdalnie sterowane

W przypadku, gdy zadanie jest skomplikowane i wcześniej niezdefiniowane, najlepszym rozwiązaniem jest przejęcie sterowania robotem przez człowieka. Rozwiązanie to jest bezpieczne zarówno dla człowieka jak i robota, który może wykonywać trudne i niebezpieczne prace na odległość. Przykładem sytuacji, w których wykorzystuje się roboty zdalnie sterowane jest poszukiwanie i detonacja bomb.

Oprócz wymienionych powyżej robotów można znaleźć takie, które trudno zakwalifikować do jakiejś kategorii. Należą do nich **roboty wirtualne** czy **roboty BEAM**.

Roboty wirtualne

Roboty wirtualne są to roboty, które nie istnieją w rzeczywistości. Są jedynie programem komputerowym. Umożliwiają one symulacje, czy rzeczywisty robot jest w stanie wykonywać określone zadania. Specjalnym rodzajem robota jest robot przeszukujący strony WWW. Nazywa się on WebCrawler i zbiera informacje o stronach, po czym przesyła je do silnika przeglądarki. Innym popularnym robotem wirtualnym jest Chatterbot. Jego zadaniem jest symulowanie rozmów z użytkownikami Internetu.





Roboty BEAM

BEAM jest skrótem od *Biology, Electronics, Aesthetic and Mechanics* (Biologia, Elektronika, Estetyka, Mechanika). Są to proste konstrukcje budowane głównie przez początkujących robotyków-hobbystów. Elementy wykorzystywane do ich budowy pochodzą ze starych, zepsutych urządzeń elektronicznych.

Biology (Biologia) – są to roboty przybierające kształty pochodzące z natury. Wiele robotów BEAM wygląda jak owady, których forma mechaniczna jest stosunkowo łatwa w budowie. Nie tylko mechaniczna część stanowi inspirację, ale również ograniczone zachowanie, które można łatwo zaprogramować w ograniczonej ilości pamięci i mocy obliczeniowej.

Electronics (Elektronika) – jak wszystkie roboty, te również zawierają elektronikę – bez przewodów elektronicznych nie można byłoby sterować napędami. Wiele robotów BEAM dodatkowo wykorzystuje energię słoneczną do zasilania układu.

Aesthetics (Estetyka) – roboty BEAM powinny wyglądać miło i atrakcyjnie.

Mechanics (Mechanika) – w przeciwieństwie do dużych robotów, roboty BEAM są tanie, proste, zbudowane, np. z materiałów pochodzących z recyklingu.



Poznanie zestawów i testowanie czujników

W trakcie zajęć wykorzystywany będzie zestaw klocków Lego Mindstorms EV3. Umożliwia on budowanie i programowanie dowolnych konstrukcji. W jego skład wchodzi:

- **Kłoczek EV3 (EV3 Brick)** – serce i mózg robota. Odbiera on sygnały z czujników i przekazuje impulsy działania do odpowiednich elementów.
- **Silnik duży (Large Motor)** – umożliwia ruch robota.
- **Silnik średni (Medium Motor)** – umożliwia ruch robota.
- **Czujnik dotyku (Touch Sensor)** – czujnik odbierający stany: wciśnięcie i zwolnienie.
- **Czujnik koloru (Color Sensor)** – rozpoznaje 7 kolorów oraz umożliwia pomiar natężenia światła.
- **Czujnik podczerwieni (Infrared Sensor)** – czujnik wykrywający obiekty, umożliwiający ich śledzenie oraz odnajdujący nadajnik (Remote Infrared Beacon).
- **Nadajnik podczerwieni (Remote Infrared Beacon)** – umożliwia zdalne sterowanie robotem.

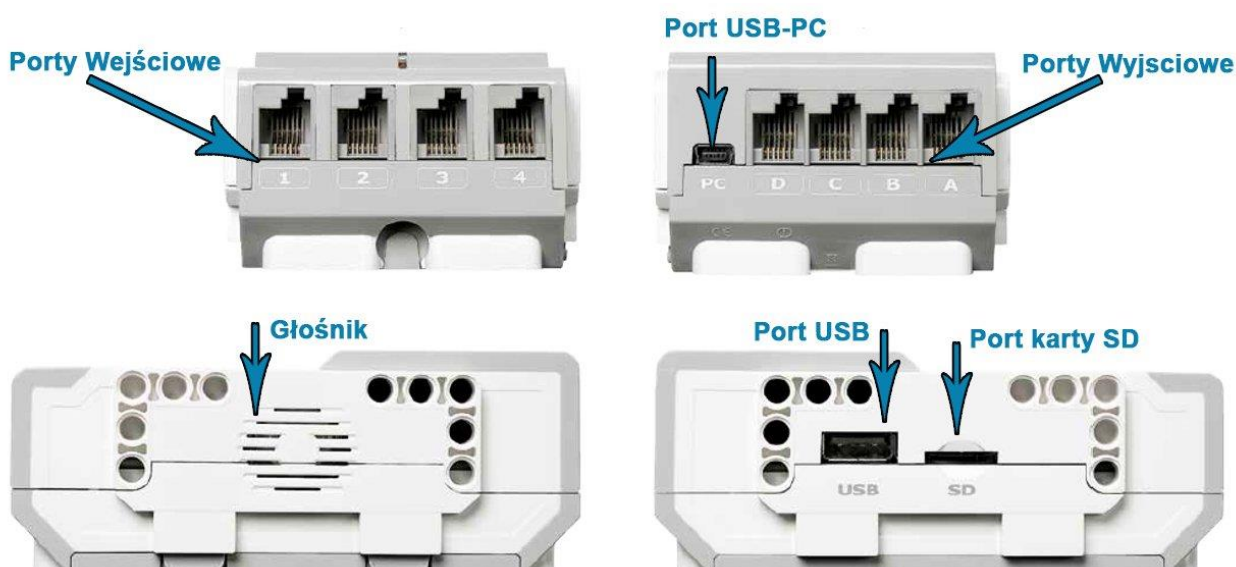


Oprócz wymienionych powyżej części w zestawie znajduje się również komplet klocków Lego.

Kostka EV3 (EV3 Brick)

Głównym elementem zestawu jest kostka EV3 – mózg naszego robota. Pobiera on informacje z czujników oraz przesyła sygnały na wyjście.

Wejścia i wyjścia umieszczone są na górnej i dolnej ścianie klocka. Porty wejściowe oznaczone są numerami 1 – 4 i przeznaczone są do podłączania czujników. Porty wyjściowe oznaczone są literami A – D i przeznaczone są do podłączania silników. Na ścianie z portami wyjścia znajduje się również port Mini – USB umożliwiający podłączenie kostki do komputera. Z lewej strony znajduje się głośnik, natomiast z prawej strony port USB oraz port kart SD. Port USB umożliwia połączenie z siecią bezprzewodową albo połączenie kilku klocków EV3 ze sobą. Port karty SD można wykorzystać w celu powiększenia pamięci kostki EV3.



Czujniki

Wspominano o nich wcześniej już kilkakrotnie. Ale czym właściwie są czujniki? Otóż czujniki to swego rodzaju zmysły każdego robota. Człowiek, ma oczy by widzieć, co go otacza, natomiast robot ma czujniki, by – analogicznie – odbierać sygnały ze świata zewnętrznego. W celu przybliżenia działania czujników zostanie przeprowadzonych kilka prostych ćwiczeń wprowadzających do tematyki czujników.

Czujnik dotyku

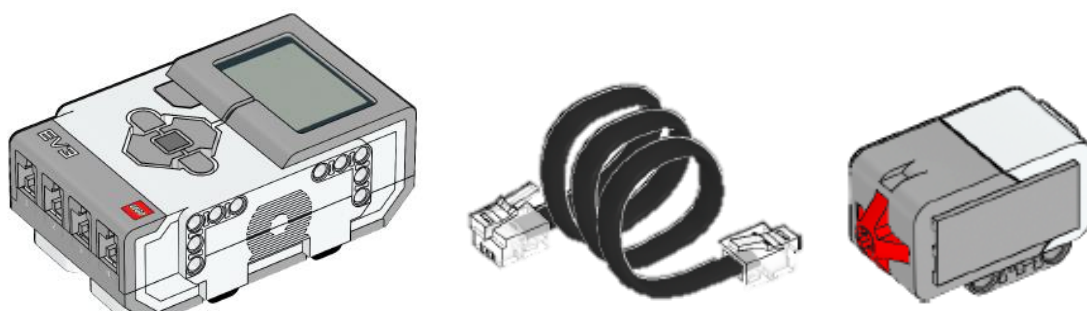


Czujnik dotyku jest to urządzenie, które odpowiada za zmysł dotyku robota. Pod obudową znajduje się układ elektroniczny składający się z przełącznika, rezystora oraz gniazda połączeniowego. Naciśnięcie przełącznika powoduje zamknięcie obwodu elektrycznego. Informacja o stanie czujnika przekazywana jest dalej do kostki EV3.

Testowanie czujnika dotyku

Krok 1. Przygotuj następujące elementy:

- Klocek EV3,
- Kabelek,
- Czujnik dotyku.



Krok 2. Podłącz jeden koniec kabelka do wyjścia nr 1 umieszczonego na kločku EV3, a drugi do czujnika.

Krok 3. Uruchoom klocek EV3.

Krok 4. Wybierz trzecią zakładkę w oknie, poruszając się klawiszami kostki EV3 (*Port View*).

Krok 5. Wybierz opcję *Port View*, zatwierdzając środkowym klawiszem. Na ekranie wyświetli się napis **1: Touch** oraz **0**.

Krok 6. Naciśnij czujnik dotyku i zobacz, co się stanie.

Wnioski

Za każdym razem, gdy czujnik dotyku jest aktywny (wykrywa nacisk) na ekranie kostki wyświetlana jest wartość **1**, w przeciwnym wypadku wyświetla **0**.

Zastosowanie:

- Sterowanie funkcjami robota,
- Robot omijający przeszkody.

Czujnik koloru



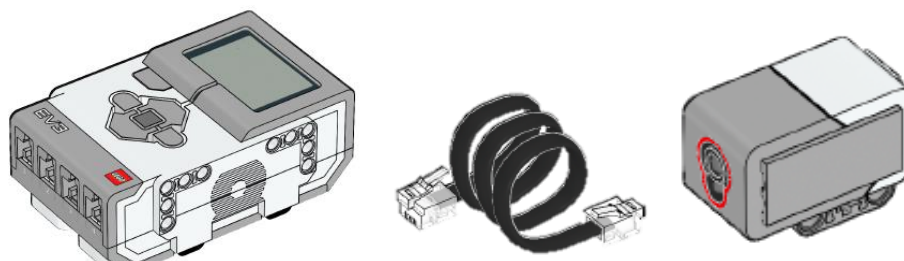
Czujnik koloru może wykryć kolor lub intensywność światła, które odbiera. Może być wykorzystany w trzech różnych trybach:

- Koloru,
- Pomiaru Intensywności Odbitego Światła,
- Pomiaru Intensywności Światła Otoczenia.

Testowanie czujnika koloru

Krok 1. Przygotuj następujące elementy:

- Kostka EV3,
- Kabelek,
- Czujnik koloru.



Krok 2. Podłącz jeden koniec kabla do wyjścia nr 1 na kostce EV3, a drugi do czujnika koloru.

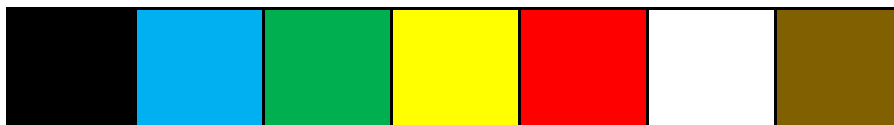
Krok 3. Uruchom kostkę EV3.

Krok 4. Lewym klawiszem kostki wybierz trzecią zakładkę (*Port View*), zatwierdzając wybór środkowym klawiszem.

TRYB KOLORU

Krok 5. Gdy znajdziesz się już w oknie testowania czujnika, ponownie naciśnij środkowy klawisz i wybierz opcję **COL - COLOR**.

Krok 6. Przyłóż do czujnika znacznik kolorów. Co zaobserwowałeś/łaś?



Wnioski

W zależności, jaki kolor przykładany jest do czujnika, na wyświetlaczu wyświetla się inny numer. Numer ten symbolizuje kolor. W tabelce poniżej numery zostały przypisane odpowiednim kolorom.

Numer	0	1	2	3	4	5	6	7
Kolor	Brak	Czarny	Niebieski	Zielony	Żółty	Czerwony	Biały	Brązowy

TRYB INTENSYWNOŚCI ODBITEGO ŚWIATŁA

Krok 5. Ponownie naciśnij środkowy klawisz i wybierz opcję **COL - REFLECT**.

Krok 6. Przyłóż do czujnika znacznik kolorów. Co zaobserwowałeś/łaś?

Wnioski

W trybie tym czujnik podaje jasność światła, które do niego wraca po odbiciu się od powierzchni przykładanej do czujnika. Wynik podawany jest w procentach (pct - %) i przyjmuje wartości od 0 do 100, przy czym 0 to kolor bardzo ciemny, a 100 bardzo jasny. Jest to spowodowane tym, że ciemne odcienie słabiej odbijają czerwone światło czujnika. Czujnik musi znajdować się blisko (około 1 cm) powierzchni odbijającej oraz być ustawiony pod kątem 90° od powierzchni.



TRYB INTENSYWNOŚCI ŚWIATŁA OTOCZENIA

Krok 5. Ponownie naciśnij środkowy klawisz i wybierz opcję **COL – AMBIENT**.

Krok 6. Przyłóż do czujnika źródło światła (np. latarkę). Co zaobserwowałeś/łaś?

Wnioski

W trybie tym czujnik kolorów wykrywa światło, które do niego dochodzi. Na ekranie wyświetla się otrzymany wynik. Podawany jest on w procentach (pct - %) od 0 do 100, przy czym 0 to bardzo ciemno, a 100 bardzo jasno.

Zastosowanie:

- Sterowanie funkcjami robota,
- Robot sprzątający,
- Linefollower,
- Pozytywka,
- Sortowanie piłeczek.

Czujnik podczerwieni



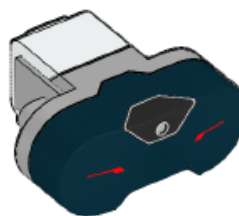
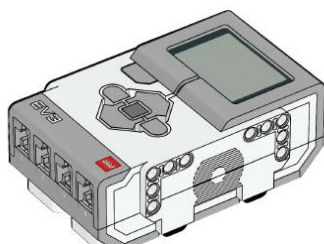
Czujnik podczerwieni potrafi wykrywać przeszkody oraz odbierać sygnały wysyłane przez nadajnik. Czujnik działa na zasadzie pomiaru czasu, jaki jest potrzebny od chwili nadania i odebrania fal podczerwieni. Ma możliwość pracy w trzech trybach:

- Przybliżenia,
- Nadajnika,
- Zdalnym.

Testowanie czujnika podczerwieni

Krok 1. Przygotuj następujące elementy:

- Kostka EV3,
- Kabelek,
- Czujnik podczerwieni,
- Nadajnik podczerwieni.



Krok 2. Podłącz jedną końcówkę kabelka do wyjścia nr 1 na kostce EV3, a drugą do czujnika podczerwieni.

Krok 3. Uruchoom kostkę EV3.

Krok 4. Lewym klawiszem kostki wybierz trzecią zakładkę (*Port View*), zatwierdzając wybór środkowym klawiszem.



TRYB PRZYBLIŻENIA

Krok 5. Gdy znajdziesz się już w oknie testowania czujnika, ponownie naciśnij środkowy klawisz i wybierz opcję **IR – PROX.**, zatwierdzając klawiszem środkowym.

Krok 6. Połóż przed czujnik jakiś obiekt (np. zeszyt), a następnie odsuń go od czujnika. Co zauważasz?

Wnioski

W trybie tym czujnik wyświetla rzeczywisty czas potrzebny na przebycie przez wiązkę światła odległości od obiektu, podając wynik w procentach (pct - %), w zakresie 0 – 100, przy czym 0 oznacza bardzo blisko przeszkody, a 100 bardzo daleko. Jest to możliwe dzięki temu, że czujnik mierzy czas pomiędzy wystąpieniem i odebraniem sygnału odbitego od przeszkody. Czas ten wykorzystany jest do oszacowania odległości od obiektu. Warto zwrócić uwagę, że wynik ten będzie inny w zależności od materiału, z jakiego wykonana jest powierzchnia, od której odbity będzie sygnał.

TRYB NADAJNIKA

Krok 5. Ponownie naciśnij środkowy klawisz i wybierz opcje **IR – SEEK**, zatwierdzając klawiszem środkowym.

Krok 6. Uruchom nadajnik poprzez naciśnięcie jego górnego klawisza. Powinna się zapalić zielona dioda na znak uruchomienia.

Krok 7. Przed czujnikiem porusz nadajnikiem (zbliżając/oddalając, zataczając kręgi wokół czujnika). Co zauważasz?

Wnioski

W tym trybie czujnik wykrywa przybliżoną pozycję nadajnika. Z czujnika można otrzymać trzy rodzaje danych:

1. **Wykrycie** – wskazuje wówczas wartość logiczną: 0 – gdy nadajnik nie został wykryty, 1 – gdy udało się go wykryć na konkretnym kanale.
2. **Przybliżenie** – przybliżoną odległość do nadajnika w procentach (pct - %) w zakresie od 0 – bardzo blisko, do 100 – bardzo daleko.
3. **Położenie** – wartości od -25 do 25. 0 oznacza, że nadajnik jest bezpośrednio przed czujnikiem, wartości ujemne oznaczają lewą stronę, a dodatnie prawą.

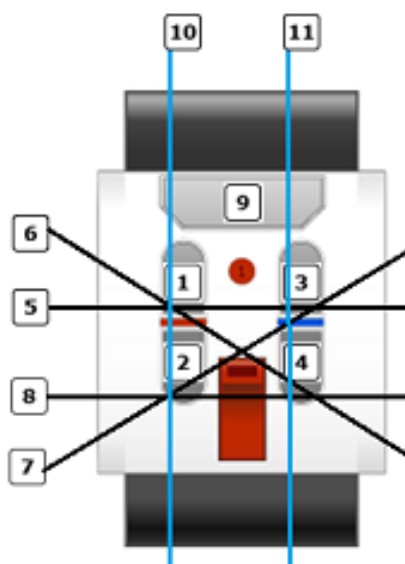
TRYB ZDALNY

Krok 5. Ponownie naciśnij środkowy klawisz i wybierz opcję **IR – REMOTE**, zatwierdzając klawiszem środkowym.

Krok 6. Uruchom nadajnik poprzez naciśnięcie jego górnego klawisza (9). Powinna się zapalić zielona dioda na znak uruchomienia.

Krok 7. Naciśnij klawisze na nadajniku. Co zauważasz?

Krok 8. Sprawdź, czy numery z rysunku poniżej pokrywają się z otrzymanymi na klocek EV3.



Wnioski

W tym trybie czujnik wykrywa, który przycisk lub ich kombinacja została naciśnięta na nadajniku.

Zastosowanie:

- Sterowanie funkcjami robota,
- Robot Sumo – poszukiwanie przeciwnika,
- Omijanie przeszkód,
- Zdalne sterowanie robotem.

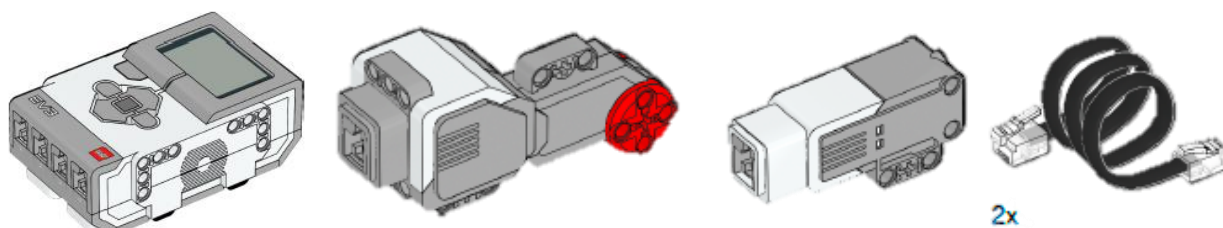
Napędy

Aby ruch robota był możliwy niezbędne są odpowiednie napędy, czyli silniki lub akтуatory. W zestawie Lego Mindstorms EV3 do dyspozycji są dwa rodzaje silników: duży i średni. W trakcie dalszej nauki zapoznacie się bliżej z każdym z nich.

Testowanie silnika średniego i dużego

Krok 1. Przygotuj następujące elementy:

- Kostka EV3,
- 2 Kabelki,
- Silnik duży,
- Silnik średni.



Krok 2. Podłącz kabelki do wejść – odpowiednio A i B – oraz do silników.

Krok 3. Uruchom kostkę EV3.



Krok 4. Wybierz trzecią zakładkę poruszając się klawiszami kostki (*Port View*) i zatwierdź wybór środkowym klawiszem.

SILNIK DUŻY

Krok 5. Naciskając prawy klawisz przesuwasz się pomiędzy wyjściami (1 – 4) i wejściami (A – D) portów. Naciśnij kilkakrotnie prawy klawisz, by ustawić podgląd podłączonego portu.

Krok 6. Spróbuj obrócić czerwony element silnika dużego raz w jedną, a raz w drugą stronę. Co zauważasz?

SILNIK ŚREDNI

Krok 5. Ustaw podgląd podłączonego portu, poruszając się po menu prawym klawiszem kostki EV3.

Krok 6. Spróbuj obrócić czerwony element silnika średniego raz w jedną, a raz w drugą stronę. Co zauważasz?

Wnioski

Na ekranie wyświetla się pomiar, o ile stopni obrócił się silnik. Możliwe jest zarówno sprawdzenie liczby obrotów silnika, jak i wartości w stopniach. Aby zmienić tryb należy nacisnąć środkowy klawisz na kostce i tym samym zatwierdzić wybór. Pełen (jeden) obrót silnika to 360 stopni. Należy jednak pamiętać, że napędy podłącza się tylko i wyłącznie do wyjść od A do D.

Zastosowanie:

- Roboty mobilne (np. robot Sumo, Linefollower, Wyścigówka),
- Roboty stacjonarne (np. dźwig, pozytywka),
- Wszędzie tam, gdzie potrzebny jest ruch.



Pierwsza konstrukcja i programowanie

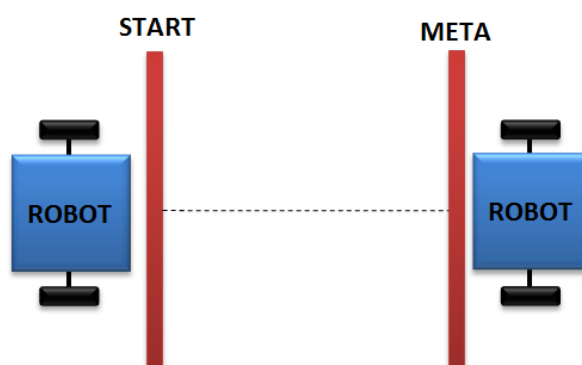
Wprowadzenie

Tematem dzisiejszych zajęć będzie praktyczne zapoznanie się z zestawem Lego Mindstorms EV3, zbudowanie pierwszego robota oraz zaprogramowanie go. Poznasz obsługę programu umożliwiającego programowanie robota oraz dowiesz się, na jakiej zasadzie przebiega proces skręcania/- zawracania w robotach mobilnych. Będziesz miał możliwość przetestowania czujnika podczerwieni i dotyku oraz sterowania napędami.

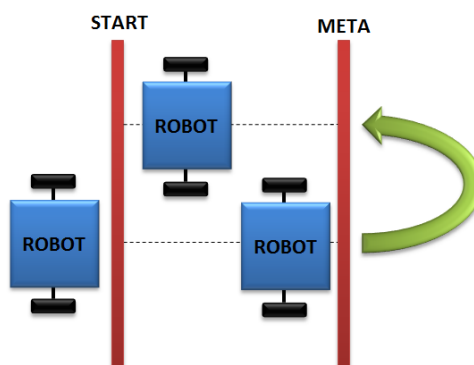
Zadania do wykonania

Robot będzie miał kilka zadań do wykonania:

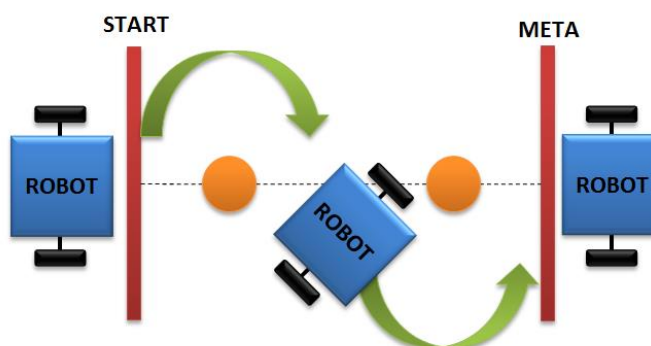
Zadanie 1. Jazda od linii do linii,



Zadanie 2. Jazda od linii do linii z zawracaniem,



Zadanie 3. Jazda slalomem.



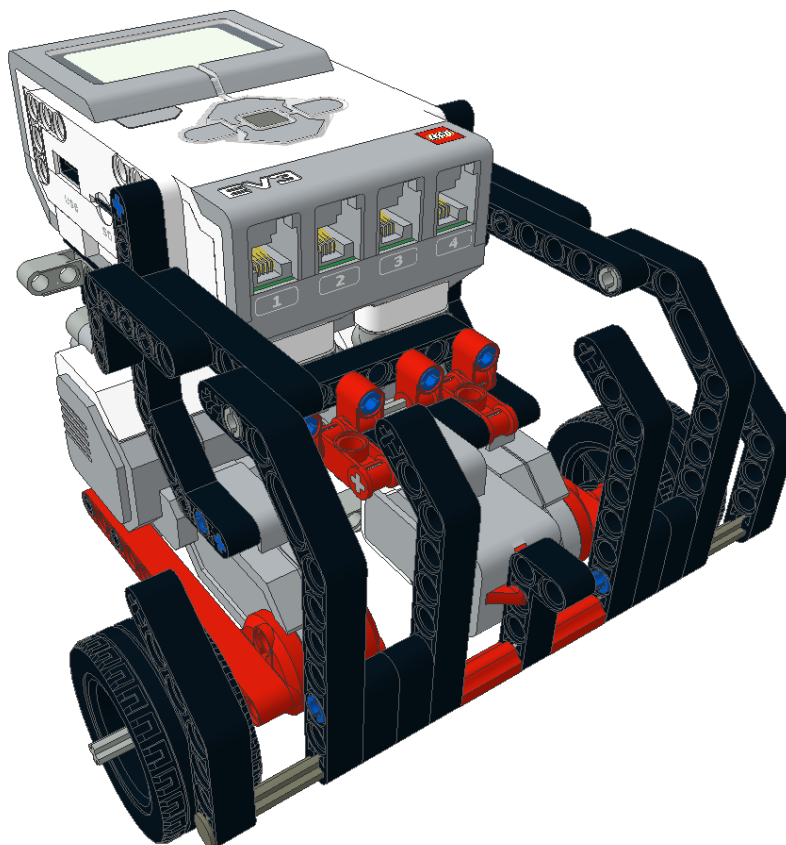


Budowa robota

Robot będzie się składał z:

- Czujnika dotyku,
- 2 silników dużych,
- Kostki EV3.

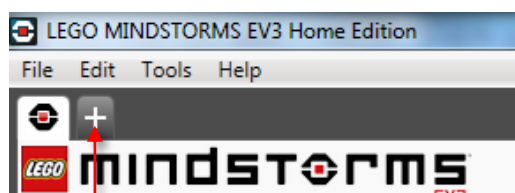
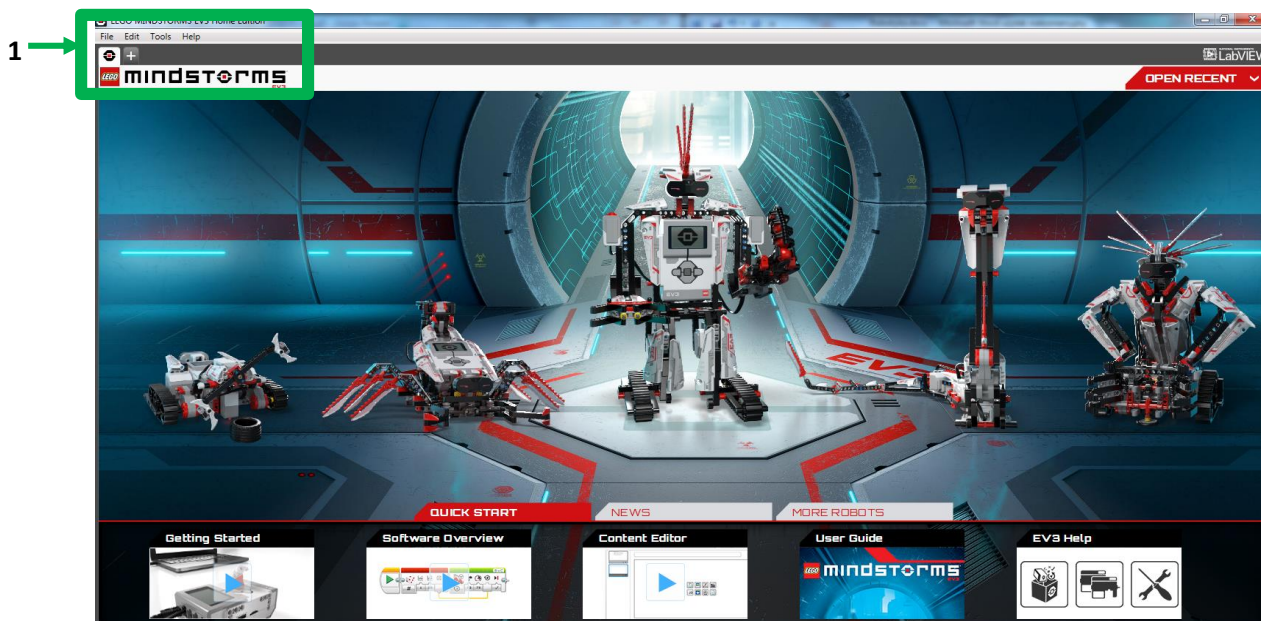
Na rysunku poniżej przedstawiono efekt końcowy. Proces budowy robota został opisany w instrukcji, która została dołączona do książki na płycie CD (2. *Easybot.pdf*).



Programowanie

Wprowadzenie

Zbudowany robot składa się z kostki EV3, dwóch silników dużych, czujnika dotyku oraz podczerwieni. Skonstruowany robot zacznie się poruszać i spełniać określone wcześniej zadania, tylko i wyłącznie wtedy, gdy programista najpierw napisze program, w którym szczegółowo opíše jak – w danej sytuacji, na dany sygnał bądź zdarzenie, ma zareagować zbudowany robot. Nie trzeba do tego wykorzystywać programów i języka specjalistycznego, którego nauka zajęłaby bardzo dużo czasu. Firma Lego, wraz z firmą National Instruments, myśląc o początkujących programistach stworzyła własne narzędzie, które w łatwy i przystępny dla użytkownika sposób pomoże wykonać nawet najbardziej skomplikowane zadania. Program ten nosi nazwę LEGO® MINDSTORMS® EV3 Software. Program, który napiszemy nie przypomina standardowego języka programowania, jakim jest np. C/C++. Programowanie będzie przypominać budowanie z klocków, w których każdy spełnia określone zadanie. W dalszej części poznamy ten program nieco lepiej.

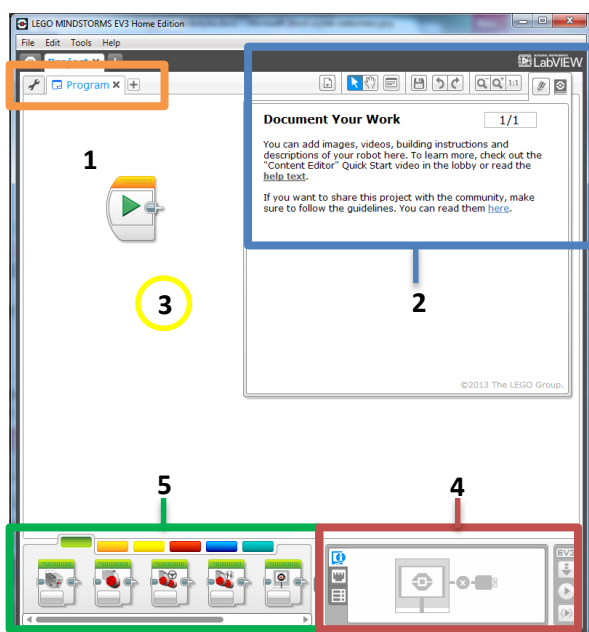


2

Po uruchomieniu oprogramowania otwiera się okno główne programu. W nagłówku umieszczone zostały podstawowe opcje: File (Plik) – umożliwia zapis/odczyt projektów, Edit (Edytuj) – umożliwia cofanie/ponawianie działań programu, Tools (Narzędzia) – zawiera kilka programów umożliwiających tworzenie plików dźwiękowych, obrazu itd., Help (Pomoc) – informacje o produkcie.

Zakładka przedstawiająca + (2) umożliwia rozpoczęcie nowego projektu

Okno główne projektu składa się z 5 części. Zaznaczone one zostały na rysunku poniżej.



1. **Zakładki** – umożliwiają operowanie pomiędzy zakładkami programu a jego właściwościami.

2. **Pasek narzędzi** – znajduje się po prawej stronie, w górnej części okna. Zawiera funkcje umieszczone w menu Edit oraz dodatkowe opcje, jak powiększanie/zmniejszanie. Oprócz tego zawiera możliwość wstawiania komentarzy oraz tworzenie dokumentacji swojej pracy.

3. **Główne okno** – to właśnie tu pisany będzie program.

4. **Karta sprzętu** – umożliwia sprawdzenie łączności z kostką EV3 i zarządzanie nią. Wskazuje, gdzie są podłączone poszczególne silniki i czujniki. Umożliwia również wczytywanie programów do kostki EV3.

5. Bloki funkcyjne – zbiór klocków wykorzystywanych do oprogramowania. Składa się on z 6 zakładek. Każda z nich zawiera inny typ bloków.

Bloki działań – zakładka zawiera bloki dedykowane dla konkretnych elementów zestawu (silniki, kostka EV3).

Bloki przepływu – bloki odpowiadające za przebieg programu. Blok startu rozpoczyna każdy program.

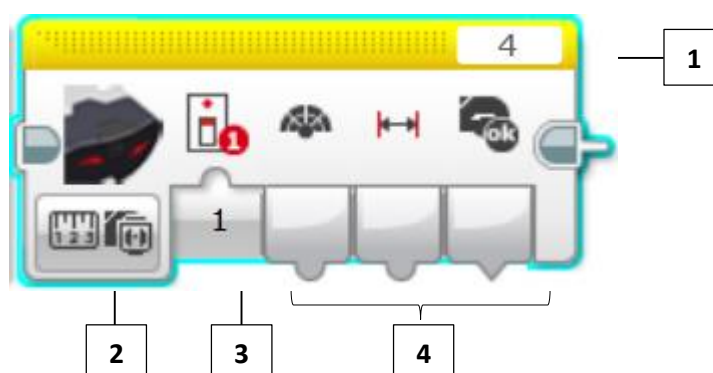
Bloki czujników – bloki zarezerwowane do obsługi czujników.

Bloki obsługi danych – bloki umożliwiające działania na danych (dodawanie, odejmowanie, itd.).

Bloki funkcji zaawansowanych – bloki umożliwiające bardziej zaawansowane działania, jak wysyłanie wiadomości, komunikację przez Bluetooth.

Bloki własne – bloki, utworzone przez użytkownika.

Każdy z bloków ma podobną budowę. Składa się z (1) wyboru portu, (2) wyboru trybu działania, (3) sygnałów wejściowych oraz (4) sygnałów wyjściowych.



Programowanie krok po kroku

Po zapoznaniu się z częścią teoretyczną, czas najwyższy na praktykę. Zanim zaczniemy programować, musimy dobrze przemyśleć, jak powinien działać program. Dobrze przemyślany program sprawi, że:

1. Łatwiej i szybciej napiszemy kod programu,
2. Robot będzie przygotowany na każdą sytuację i – co za tym idzie – będzie bardziej autonomiczny.

Podstawową zasadą jest przemyślenie funkcji robota. Gdy cel jest już określony, można zacząć przygotowywać algorytm działania robota.

Algorytm działania

Czym właściwie jest algorytm?

Algorytm jest to ciąg konkretnie określonych czynności, które należy wykonać by osiągnąć zamierzony cel.



Przykładem algorytmu, z którym każdy zapewne się zetknął, jest przepis kulinarny. Zawiera on na początku spis wszystkich elementów potrzebnych do przygotowania potrawy oraz szczegółowy opis poszczególnych etapów jej powstawania. Istnieje wiele przykładów algorytmów, m. in. algorytm Euklidesa – wyznaczanie największego wspólnego dzielnika dwóch liczb naturalnych. Algorytmy znajdują swoje zastosowanie najczęściej w informatyce, gdyż ułatwiają opisywanie i obliczanie ogromnych ilości danych w bardzo krótkim czasie.

JAZDA OD LINII DO LINII



Krok 1. Uruchomienie robota.

Krok 2. Natychmiastowe uruchomienie silnika lewego i prawego na określony czas/- liczbę obrotów.

Kod programu

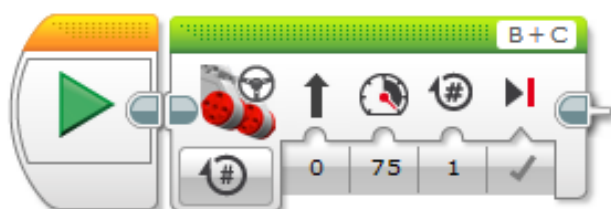
Krok 1. Kod programu rozpoczyna **blok Startu**.

Krok 2. Za jazdę wprzód dwóch silników odpowiada blok **Move Steering**. Należy wybrać jeden z trzech trybów działania bloku:

- W zależności od czasu (*On for Seconds*),
- W zależności od liczby stopni (*On for Degrees*),
- W zależności od liczby obrotów (*On for Rotations*),

kierunek jazdy robota, moc działania silników.

Krok 3. Wgraj program i testuj.



JAZDA OD LINII DO LINII Z ZAWRACANIEM

W tym miejscu warto się zastanowić, w jaki sposób robot skręca.

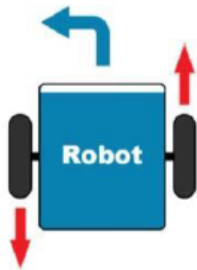
W JAKI SPOSÓB SKRĘCA ROBOT?

Istnieją dwa sposoby skręcania:

- W miejscu,
- Po łuku.



SKRĘT W MIEJSCU

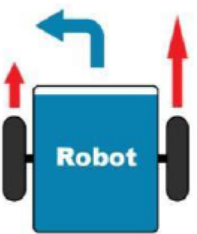


Aby robot skręcał w miejscu, należy:

- Ustawić kierunek działania silnika lewego do tyłu,
- Ustawić kierunek działania silnika prawego do przodu,
- Prędkość taka sama dla lewego i prawego silnika.

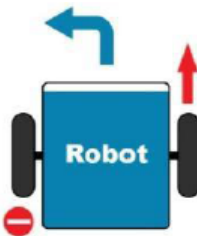
SKRĘT PO ŁUKU

Skręt po łuku można zrealizować na dwa sposoby:



SPOSÓB I

- Ustawić taki sam kierunek ruchu silnika lewego i prawego do przodu,
- Ustawić różną prędkość pracy silników: silnik prawy kręci się szybciej, niż lewy lub odwrotnie



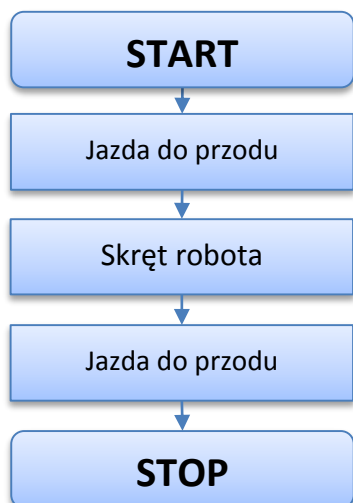
SPOSÓB II

- Zatrzymać silnik lewy,
- Ustawić kierunek ruchu silnika prawego do przodu.

Ciekawostka

Zastanawiało Cię kiedyś, jak skręca samochód? Wykorzystuje on do tego **mechanizm różnicowy**. Jest to mechanizm umożliwiający pojazdowi pokonywanie zakrętów. Kiedy auto jest na zakręcie, zewnętrzne koło ma do pokonania drogę dłuższą niż koło znajdujące się od strony wewnętrznej. Może to spowodować, że koło zewnętrzne obróci się szybciej niż wewnętrzne – wtedy, jedno z kół będzie się ślizgać po drodze. Mechanizm różnicowy zapobiega temu zjawisku.

Znając powyższe zasady, możemy powrócić do analizy pracy robota z opcją zawracania.



Krok 1. Uruchomienie robota.

Krok 2. Natychmiastowe uruchomienie silnika lewego i prawego na określony czas/liczbę obrotów.

Krok 3. Zawracanie – realizowane według uznania.

Krok 4. Natychmiastowe uruchomienie silnika lewego i prawego na określony czas/liczbę obrotów.

Kod programu

Krok 1. Kod programu rozpoczyna **blok Startu**.

Krok 2. Jazda od linii do linii z obrotem realizowana jest przy pomocy trzech bloków *Move Steering*. Pierwszy blok odpowiada za jazdę do linii. Wybierz odpowiedni tryb bloku, kierunek działania silników, liczbę obrotów/czas jazdy.

Krok 3. Drugi blok odpowiada za obrót robota. Ustaw tryb działania bloku, zakres obrotu, moc silników oraz liczbę obrotów/czas jazdy.

Krok 4. Ostatni blok odpowiada za jazdę do linii początkowej. Opcje ustawień, jak w bloku pierwszym.



JAZDA ŚLALOMEM

Krok 1. Uruchomienie robota.

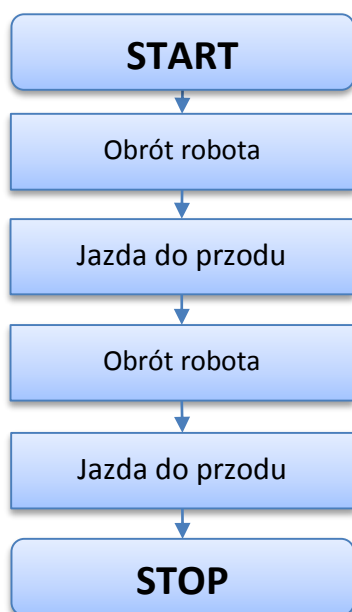
Krok 2. Obrót robota w miejscu o określony kąt.

Krok 3. Natychmiastowe uruchomienie silnika lewego i prawego do jazdy na wprost przez określony czas.

Krok 4. Obrót robota w miejscu o określony wcześniej kąt z przeciwnym kierunkiem.

Krok 5. Jazda na wprost przez określony czas.

Krok 6. Kroki powyżej opisują pokonanie jednej przeszkody. W zależności od liczby przeszkód wyznaczonych do pokonania, należy powtarzać powyższe kroki.



Kod programu

Znając podstawy działania programu oraz mając skonstruowanego robota, możemy rozpocząć pierwszy program.

Krok 1. Kod programu rozpoczyna **blok Startu**.

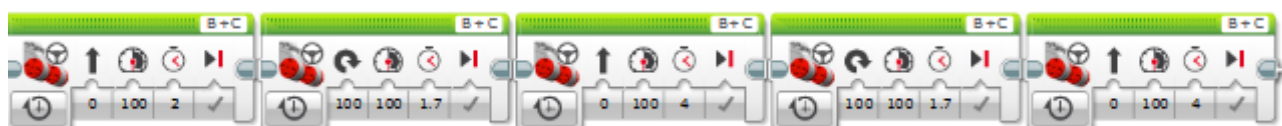
Krok 2. Jazda slalomem wykorzystuje bloki *Move Steering*. Rozpoczyna się od ustawienia robota pod pewnym kątem.

Krok 3. Następnie robot przejeżdża kilka obrotów silnika do przodu.

Krok 4. Gdy robot znajduje się obok przeszkody, blok *Move Steering* realizuje obrót.

Krok 5. Po obrocie o odpowiedni kąt, robot ponownie porusza się do przodu w linii prostej.

Krok 6. W zależności od liczby przeszkód, powtarzamy powyższe kroki.



Robot sprzątający

Robota sprzątającego można spotkać na co dzień w niektórych domach, służącego za odkurzacz. Jest to przykład praktycznego wykorzystania robota, wykonującego nieprzyjemne prace za człowieka – Omijanie przeszkód jest możliwe dzięki zastosowaniu czujników odległości lub sonarów.



Czujniki odległości

Czujniki odległości, zwane również dalmierzami, są urządzeniami służącymi do pomiaru odległości. Metoda najczęściej stosowana w tych czujnikach bazuje na pomiarze czasu przelotu impulsu energii od nadajnika do przeszkody i z powrotem.

Najczęściej stosowanymi nośnikami energii są:



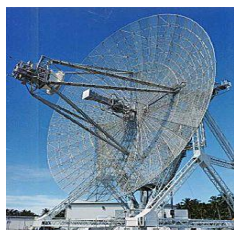
PODCZERWIEŃ – wykorzystywana w dalmierzach IR (*Infrared*). Czujniki są małych rozmiarów i pobierają stosunkowo niewiele energii. Niestety jakość pomiaru zależy od barwy przeszkody i oświetlenia w pomieszczeniu.



ULTRADŹWIĘKI – wykorzystywane w sonarach. Pomiar wykonywany za pośrednictwem ultradźwięków jest pomiarem pośrednim. Polega on na pomiarze czasu powrotu wysłanych z nadajnika impulsów o określonej częstotliwości od przeszkody.



WIĄZKA ŚWIATŁA – wykorzystywana w dalmierzach laserowych. Pomiar odległości polega na pomiarze czasu, potrzebnego na wysłanie i powrót wiązki światła. Na tej podstawie oblicza się odległość do obiektu. Czujniki tego typu znajdują zastosowanie w wojsku, policji, budownictwie, a nawet w sporcie.



FALE RADIOWE – wykorzystywane w radarach. Pomiar odbywa się poprzez pomiar czasu potrzebnego impulsowi radiowemu na przebycie drogi pomiędzy urządzeniem a celem (nadajnik – cel – odbiornik).

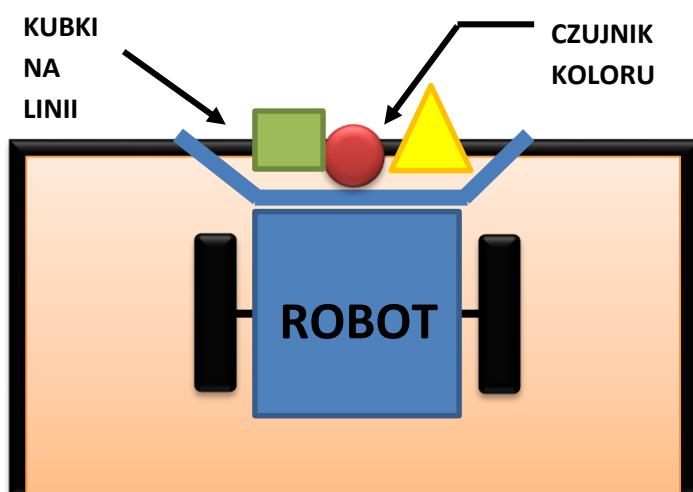
Powyższe czujniki pełnią funkcję oczu robota. Pomagają tworzyć mapę otoczenia, dzięki której robot wie w jakim miejscu aktualnie się znajduje. Technika wykorzystywana przez roboty i pojazdy autonomiczne do budowania map nieznanych środowisk i równoczesnego ich aktualizowania nazywane jest **SLAM** (*Simultaneous localization and mapping*).

Zadanie do wykonania

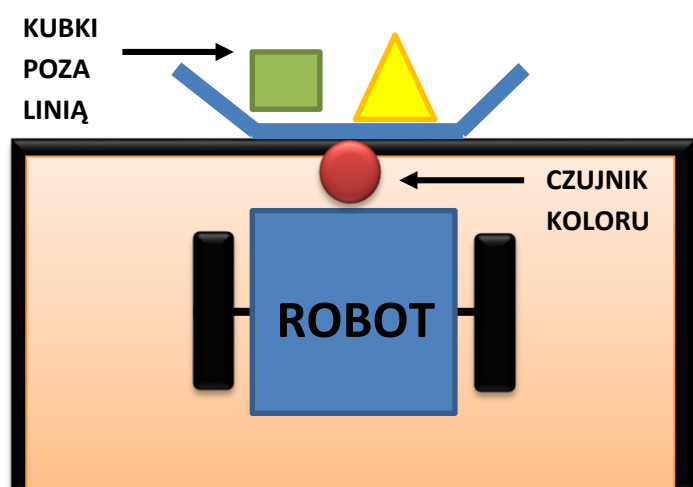
Zadaniem robota sprząającego będzie oczyszczenie planszy, ograniczonej czarną linią z przedmiotów znajdujących się na niej, w jak najkrótszym czasie. W tym celu robot wyposażony zostanie w:

- silniki umożliwiające poruszanie się,
- czujnik koloru, który będzie sprawdzał, czy robot nie wyjechał poza pole planszy.

Robot wyposażony jest w długi **zderzak** umożliwiający przesuwanie jak największej liczby przedmiotów. Istotnym zagadnieniem przy budowie tego robota, jest kwestia umiejscowienia czujnika koloru. Rozważmy dwie możliwości: przed oraz za zderzakiem.



Zamontowanie czujnika koloru **przed zderzakiem** powoduje wykrycie linii – wycofanie się, zanim robot wypchnie kubki.



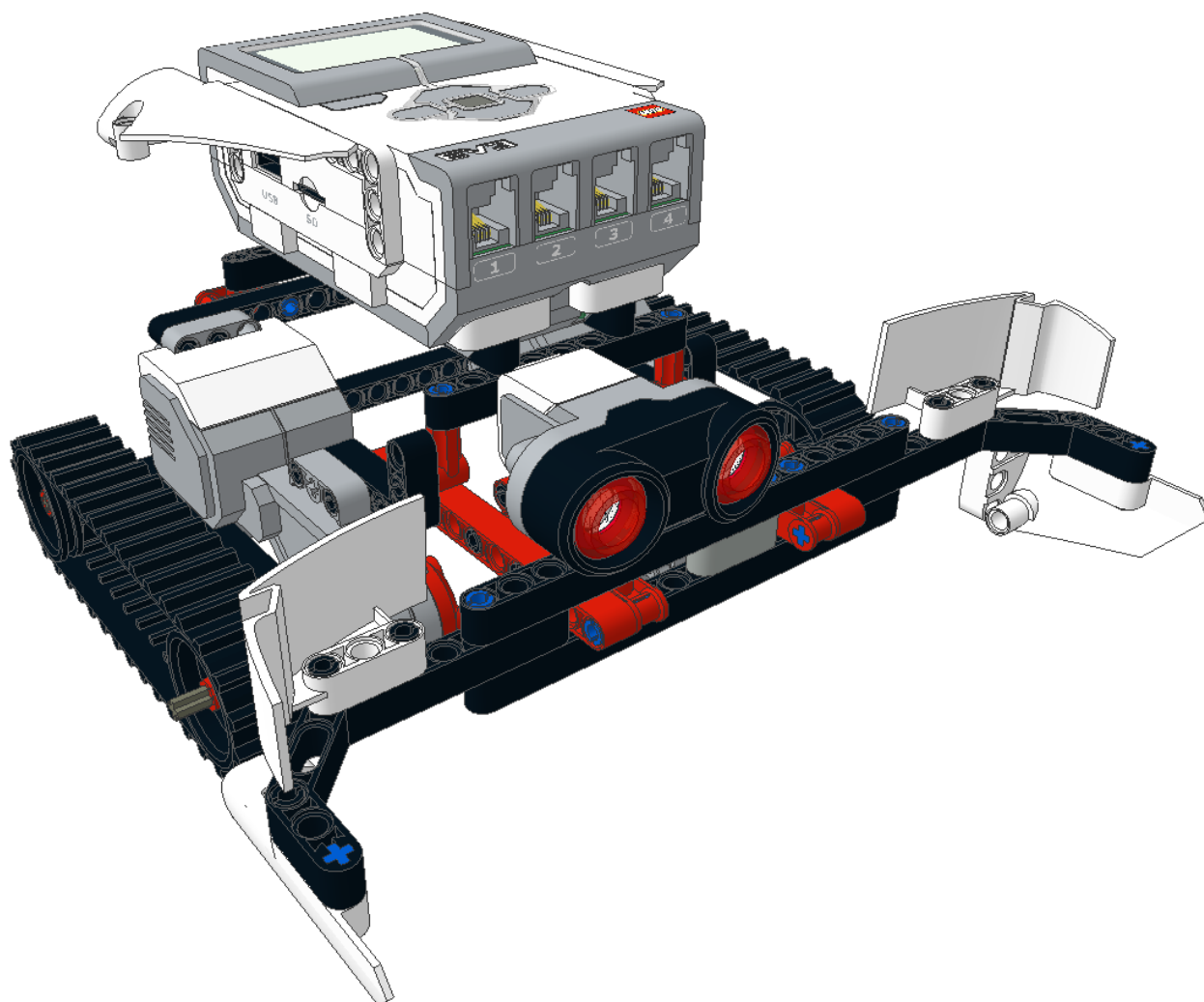
W przypadku, gdy czujnik zostanie zamontowany **za zderzakiem**, najpierw to kubki są wypychane poza linię planszy. Następnie czujnik koloru wykrywa linię i robot wycofuje się. Jest to prawidłowy sposób zamontowania czujnika.



Budowa robota

Konstrukcję robota sprzątającego przedstawia rysunek poniżej. Będzie on składać się z:

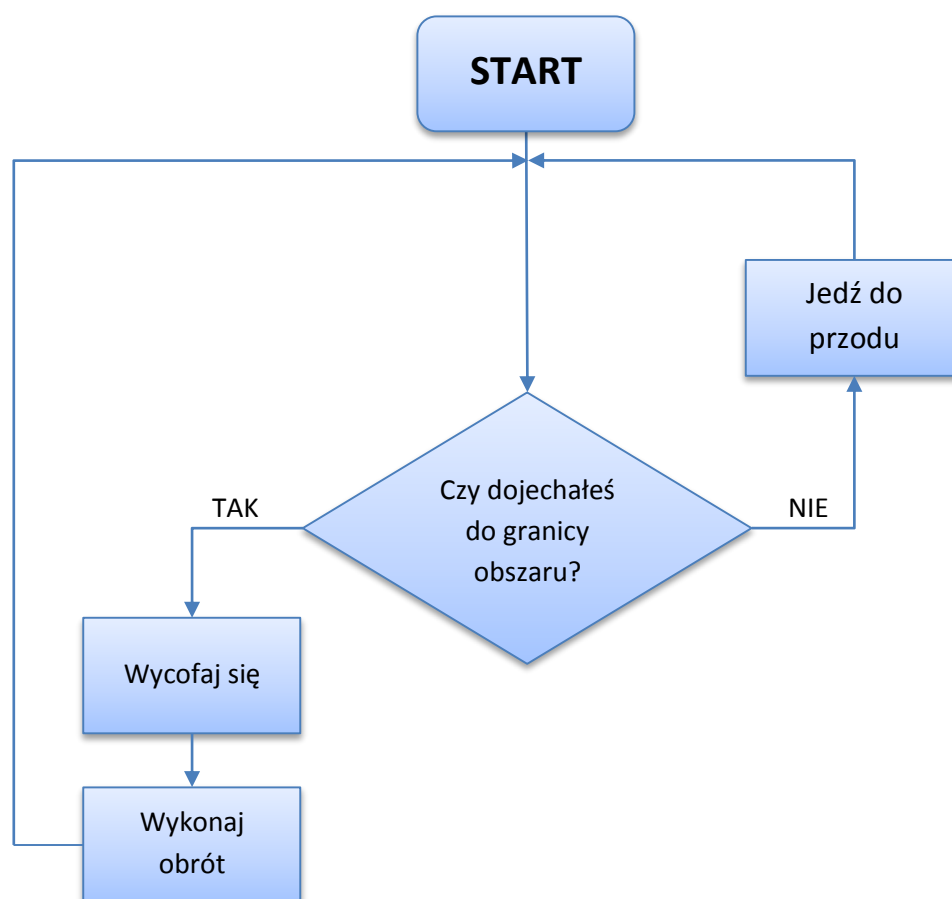
- Silników napędzających gąsienice robota,
- Zderzaka,
- Czujnika koloru,
- Czujnika odległości.



Konstrukcja robota sprzątającego musi zmieścić się na kartce formatu A4. Instrukcja realizacji robota została umieszczona na płycie CD dołączonej do książki (3. *Spychacz.pdf*).

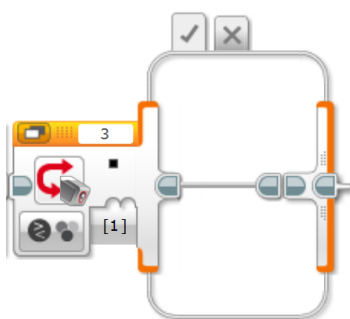
Programowanie krok po kroku

Robot będzie poruszał się po planszy ograniczonej czarnym pasem. Startuje z wyznaczonego przez prowadzącego miejsca na planszy, ustawiony pod niewielkim kątem. Zadaniem robota jest wypchnięcie jak największej ilości obiektów z planszy.



Algorytm działania

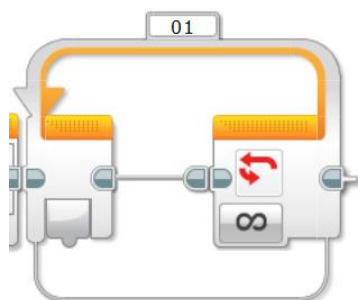
Aby robot działał sprawnie należy podane kroki powtarzać wielokrotnie. W tym celu stosuje się tzw. **Pętle**. Jest to blok w programie, który umożliwia powtarzanie konkretnych kroków dowolną liczbę razy.



Programiści często wykorzystują blok zwany **Instrukcją warunkową**. Pozwala ona zrealizować rozgałęzienie w programie (patrz schemat), czyli podjąć decyzję odnośnie zachowania robota w zależności od pewnych ustalonych warunków (np. stanu czujnika).

Kod programu dla robota sprzątającego

Krok 1. Program rozpoczyna blok Startu.



Krok 2. W związku z tym, że kod ma być wielokrotnie powtarzany wykorzystany zostanie **blok Pętli (Loop)**.

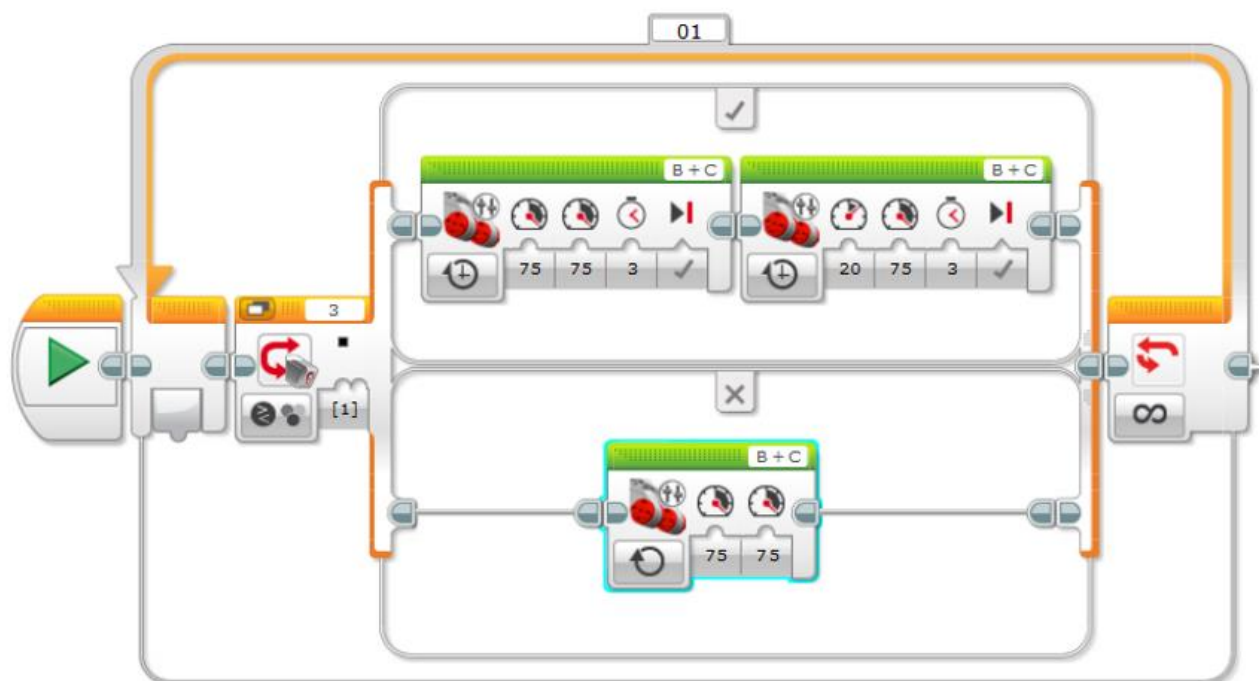
Krok 3. Znając już w teorii instrukcję warunkową, teraz możesz wykorzystać ją w praktyce. Zatem kolejnym blokiem, który zostanie wykorzystany, będzie **blok Instrukcji warunkowej (Switch)**. W ustawieniach wybiera się czujnik, który będzie wykorzystywany i z jaką opcją (*Color Sensor – Compare – Color*).



Krok 4. Teraz wystarczy wprowadzić poszczególne warunki:

Krok 4A. W przypadku, gdy kolor nie jest wykryty (znak X w instrukcji warunkowej) silniki lewy i prawy są uruchamiane z jazdą prosto.

Krok 4B. Gdy linia zostanie wykryta silniki powinny zacząć działać w przeciwnym kierunku (w tył) oraz obrócić robota o określony kąt.





Robot wyścigówka



Wyścigi samochodowe, od początków motoryzacji cieszą się ogromną popularnością. Jest to swego rodzaju wyścig zbrojeń, gdzie koncerny motoryzacyjne rywalizują, projektując coraz szybsze samochody. Jednakże zawody samochodowe to bardzo droga zabawa, na którą nie wszystkich stać oraz nie każdy ma do tego predyspozycje. Wiążą się one również z dużym niebezpieczeństwem. Można jednak ominąć te wszystkie przeszkody i dobrze się bawić, nie będąc zawodowym kierowcą. Każdy może samodzielnie zbudować swoją wyścigówkę, z części jakimi dysponuje, oraz z programem zgodnym z własnym pomysłem. Na dzisiejszych zajęciach Ty zostaniesz konstruktorem, który zostawi rywali daleko w tyle!

Na miejsca... Gotowi... START!

Co to robot wyścigówka?

Robot wyścigówka jest robotem, którego głównym zadaniem jest jak najszybsze przejechanie określonej trasy. Charakteryzuje się symetryczną budową, dzięki czemu zapobiega się nieprzewidzianym skrętom pojazdu. Każdy silnik załączony do zestawu Lego Mindstorms EV3 ma tę samą moc, która pozwala uzyskać określoną prędkość. Aby ją zwiększyć zastosujemy **przekładnię**.



Czym jest przekładnia?



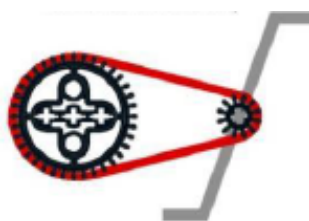
Każdy chociaż raz jeździł na rowerze. Stanowi on najprostszy przykład zastosowania przekładni. Spójrz na rysunek poniżej i przeanalizuj zasadę jej działania.

Współczesne rowery wyposażone są zazwyczaj w przerzutki. Jaka jest zasada działania przekładni wykorzystanej w budowie roweru? Przerzutki w rowerze składają się z dwóch zestawów kół zębatych połączonych przy pomocy łańcucha. Najpopularniejsze rowery posiadają z przodu 3 zębatki, natomiast z tyłu od 7 do 9. W zależności od konfiguracji zębatek otrzymuje się inny efekt końcowy.



JAZDA POD GÓRKĘ (przekładnia na siłę)

Aby wjechać pod górę przerzutki powinno się ustawić:



- Mała zębatka z przodu,
- Duża zębatka z tyłu.

Efekt:

Powyzsza konfiguracja zębatek powoduje, że pedałuje się bardzo lekko, jedzie się wolniej, ale bez trudu podjeżdża się pod górę. Pozwala ona uzyskać **dużą siłę**, przy równoczesnym **zmniejszeniu prędkości**.

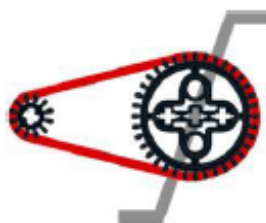
Przekładnia

jest to połączenie dwóch kół zębatych, pozwalające uzyskać wzrost prędkości lub mocy.



WYŚCIGI (przekładnia na prędkość)

Aby jechać bardzo szybko, przerzutki powinny się ustawić:



- Duża zębatka z przodu
- Mała zębatka z tyłu

Efekt:

Taka konfiguracja zębatek powoduje, że pedałuje się wolniej i ciężiej, ale jedzie się szybciej. Pozwala ona na uzyskanie **dużej prędkości**, przy równoczesnym **zmniejszeniu siły**.

W zależności od ustawień kół zębatych przekładni można uzyskać różne efekty końcowe. Gdy zależy nam na wzroście siły, wykorzystujemy przekładnię opisaną jako Jazda pod górkę, natomiast gdy zależy nam na wzroście prędkości wykorzystujemy przekładnię na prędkość (Wyścigi).



Zastosowanie przekładni:

- W rowerach,
- W samochodach,
- W robotach.

Zadanie do wykonania

Zbuduj robota, który przejdzie wyznaczoną trasę, w jak najkrótszym czasie.



Przebieg zawodów

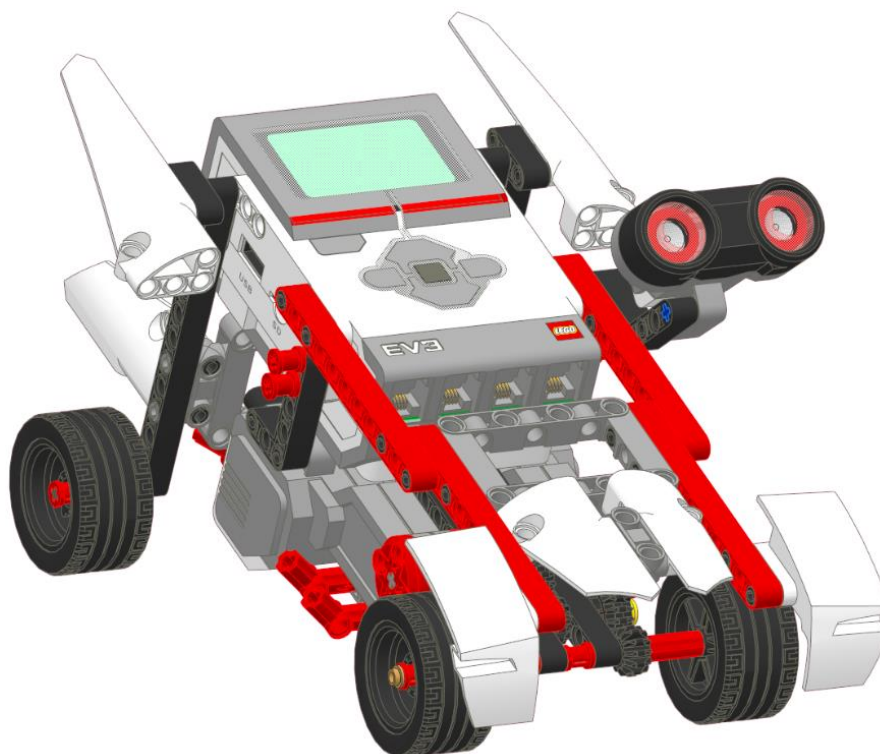
W trakcie zajęć przeprowadzone zostaną zawody, które wyłonią najszybszą konstrukcję. Każda drużyna będzie wystawiać swojego robota do walki przeciwko robotowi innej drużyny. Pojazdy umieszczane będą na linii startu i uruchamiane na polecenie prowadzącego. Wygrywa drużyna, której robot osiągnął najlepszy wynik. W przypadku, gdy żaden robot nie dojedzie do mety, wygrywa ten, który będzie się znajdował najbliżej niej.

Budowa robota

Do budowy robota wyścigowego wykorzystane zostaną:

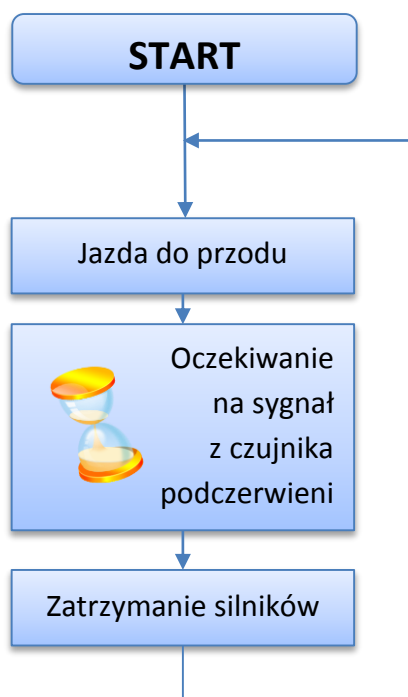
- Dwa silniki,
- Czujnik odległości,
- Kostka EV3.

Konstrukcja robota powinna być lekka, aby uzyskać dobre przyspieszenie i symetryczna, aby uniknąć nieprzewidzianych skrętów robota. W celu zwiększenia prędkości pojazdu wykorzystana zostanie omawiana wcześniej przekładnia mechaniczna. Wykonanie podwójnej przekładni z przełożeniem na prędkości, spowoduje zwiększenie szybkości robota. Czujnik odległości zastosowany w konstrukcji, będzie miał za zadanie zapobiegać ewentualnym zderzeniom. Instrukcja budowy robota znajduje się na płycie CD dołączonej do książki (*4. Wyścigówka.pdf*). Rysunek poniżej przedstawia finalną konstrukcję robota.



Programowanie krok po kroku

Zbudowana konstrukcja mechaniczna robota to dopiero połowa sukcesu. Równie ważny jest program, który będzie sterował robotem.



Krok 1. Po uruchomieniu robota ustawienie pełnej moc silników.

Krok 2. Oczekiwanie na sygnał z czujnika odległości (wykrycie przeszkody).

Krok 3. Zatrzymanie silników.

Kod programu dla robota wyścigowego

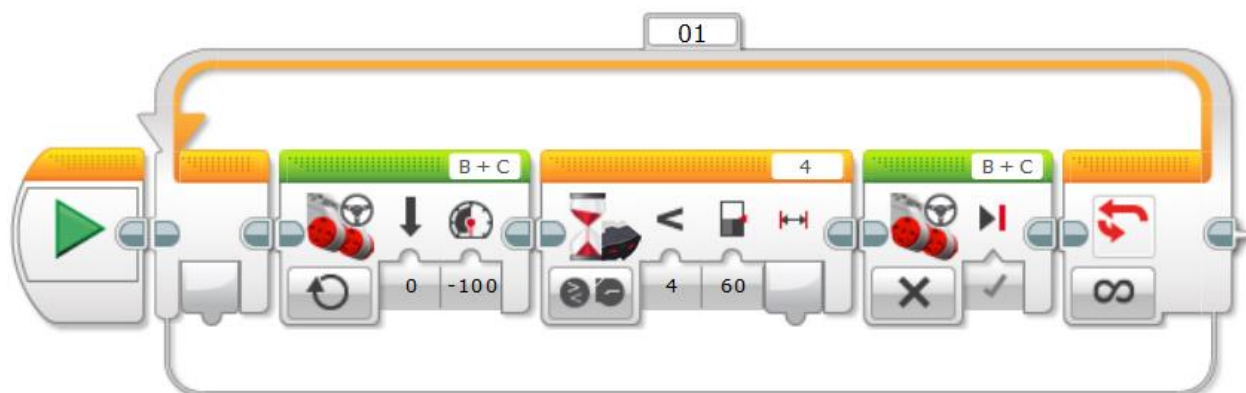
Krok 1. Sekwencję rozpoczyna **blok Startu**.

Krok 2. Aby program działał nieprzerwanie wykorzystana zostanie **Pętla** programowa.

Krok 3. Uruchomienie silników na pełną moc. Należy zwrócić uwagę, gdzie znajduje się przód robota.

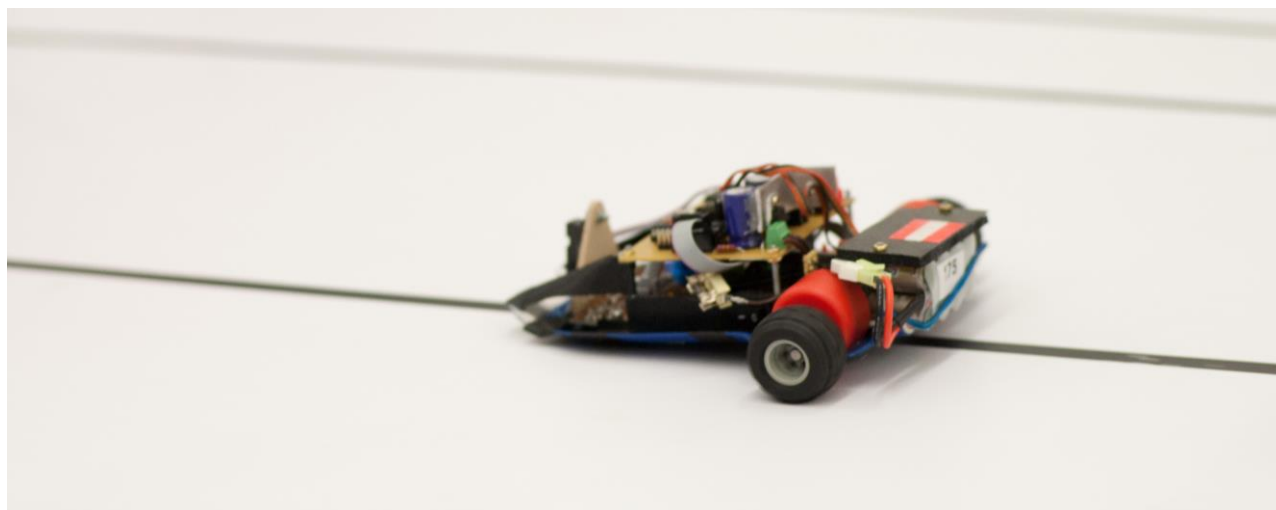
Krok 4. W celu sprawdzenia, czy robot dostrzega przeszkodę wykorzystany zostanie **blok Oczekiwania (Wait)** z odpowiednimi parametrami – tryb (*Infrared Sensor – Compare – Proximity*), typ porównania, odległość od przeszkody.

Krok 5. Jeśli czujnik podczerwieni napotka przeszkodę, wówczas należy zatrzymać silniki.





Line follower



Linefollower?

Nazwa Linefollower powstała w wyniku połączenia dwóch angielskich słów *Line* – linia oraz *Follow* – podążać, czyli podążać za linią. Robot ten jest przykładem autonomicznego robota mobilnego, którego zadaniem jest podążanie wzdłuż linii. Prawdopodobnie jest to najpopularniejszy rodzaj robota budowanego przez robotyków hobbystów. Spowodowane jest to względnie prostą konstrukcją mechaniczną robota. Coraz częściej wyścigi tych robotów można zobaczyć na różnego rodzaju konkursach. Zawody polegają na przejechaniu wyznaczonej trasy, w jak najkrótszym czasie. Czy takie roboty mogą mieć praktyczne zastosowanie? Jak najbardziej. Roboty te znajdują zastosowanie w fabrykach. Pokonują one ścieżkę na jednym piętze w celu dostarczania lub odbierania przedmiotów potrzebnych lub zbędnych w danym momencie produkcji.

Zadanie do wykonania



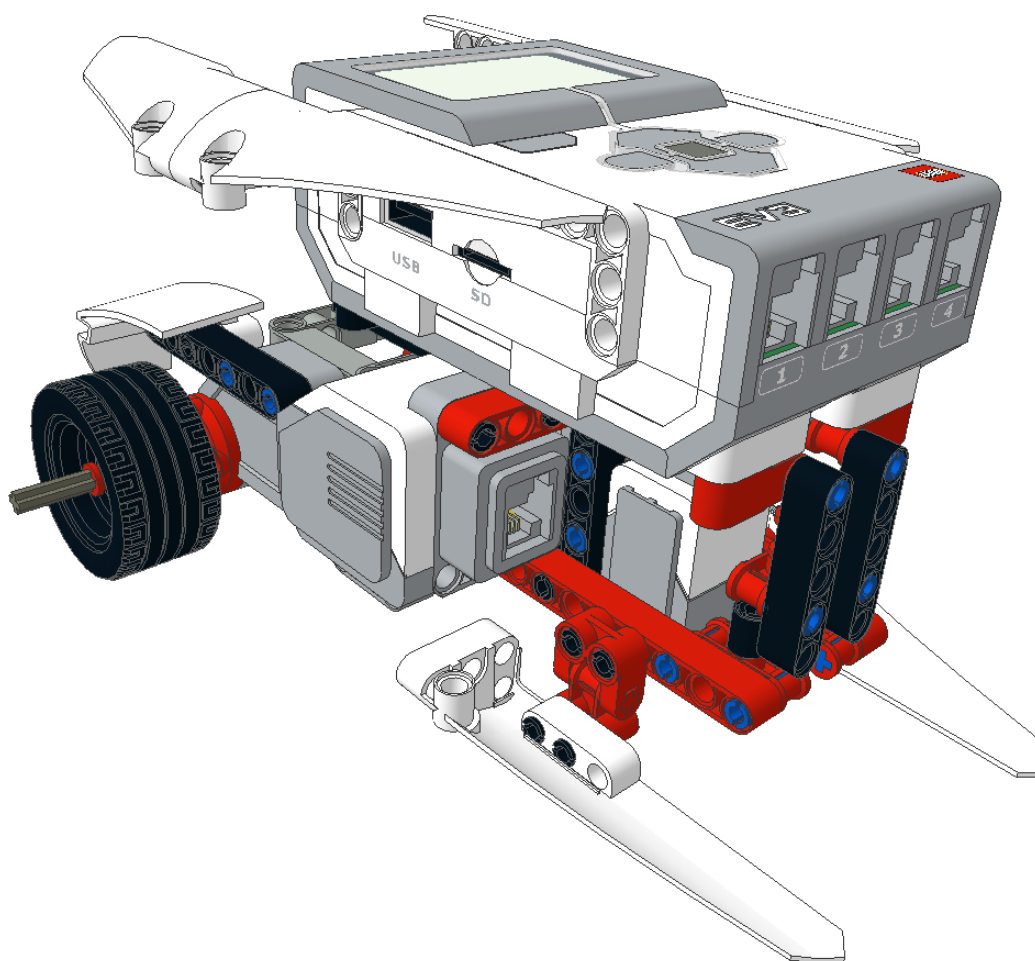
Zbudowanie i zaprogramowanie robota zdolnego do poruszania się po krętej trasie oznaczonej białą i czarną linią.

Budowa robota

Konstrukcja robota Linefollower może wyglądać różnie, w zależności od upodobań konstruktora. Elementy, które musi posiadać robot śledzący linię to:

- Minimum dwa silniki umożliwiające skręty kół,
- Czujnik koloru, umożliwiający śledzenie linii,
- Kostkę EV3.

Instrukcja budowy robota znajduje się na płycie CD dołączonej do książki (5. *Linefollower.pdf*). Rysunek poniżej przedstawia efekt końcowy.



Programowanie krok po kroku

Konstrukcja robota posiada jeden czujnik koloru oraz podwozie wyposażone w trzy koła. Informacja o liczbie czujników jest niezmiernie istotna, aby program, który napiszemy, wykonał prawidłowo swoje zadanie. Zastanów się nad algorytmem, według którego robot będzie się poruszał po wyznaczonej trasie. W naszym przypadku będzie to ruch w kształcie litery Z, tzw. zyg-zak.



Algorytm działania



Krok 1. Jeśli czujnik wykryje kolor **CZARNY**, wówczas odwraca się delikatnie od linii.

Krok 2. Jeśli czujnik wykryje kolor **BIAŁY** (nie czarny), wówczas robot skręca z powrotem w kierunku linii.

Podążając zgodnie z zaprezentowanym algorytmem, robot wykonuje tzw. kształt „Z”.

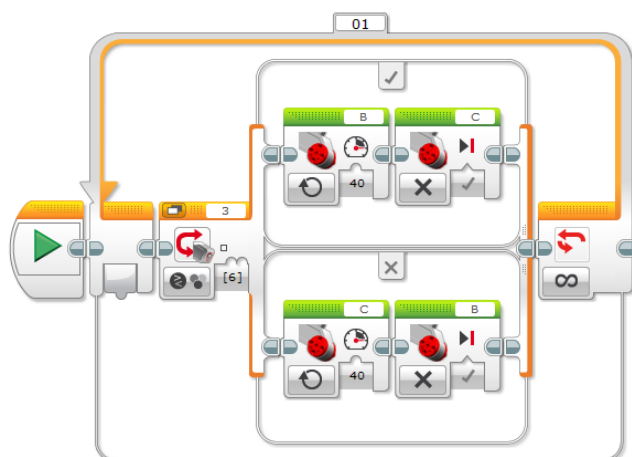
Kod programu dla robota klasy Linefollower z jednym czujnikiem

Krok 1. Sekwencję rozpoczyna **blok Startu**.

Krok 2. Aby dane z czujnika były nieustannie odczytywane, a sterowanie kół regularnie korygowane, wykorzystana zostanie **Pętla programowa**. Będzie ona powtarzana kilka razy na sekundę.

Krok 3. Wewnątrz pętli sprawdzany będzie stan czujnika i realizowana odpowiednia sekwencja kroków. W tym celu zostanie wykorzystany blok **Instrukcji warunkowej** (*Switch*). W ustawieniach należy wskazać rodzaj czujnika oraz tryb jego działania (*Color Sensor -> Compare -> Color*). Jako sygnał wejścia należy wybrać kolor, który będziemy sprawdzać (*[1] - Czarny*). Należy pamiętać również o sprawdzeniu portu wyjścia, do którego jest podłączony czujnik i wprowadzeniu jego poprawnego numeru.

Krok 4. Jeśli czujnik wykryje kolor czarny – skręca w lewo, a gdy nie wykryje koloru – skręca w prawo.





Robot przemysłowy

Co to jest robot przemysłowy?

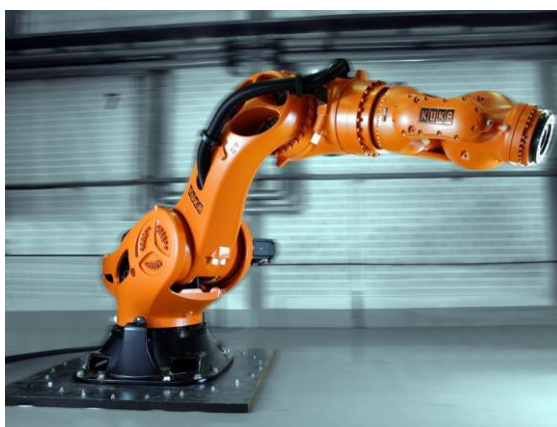
Robot przemysłowy, jak sama nazwa wskazuje wykorzystywany jest do pracy w przemyśle. Przeznaczony jest do wykonywania prac:

- Wymagających dużego wysiłku fizycznego,
- Powtarzalnych,
- Monotonnych,
- Wymagających dużej dokładności
- Odbywających się w warunkach szkodliwych dla życia i zdrowia człowieka.

Dział nauki zajmujący się tą tematyką nazywany jest **robotyką przemysłową**.

Najpopularniejszy robot wykorzystywany w przemyśle przedstawiony został na rysunku poniżej. Ten typ robota nazywamy manipulatorem. Służy on do wykonywania dokładnych ruchów (np. spawanie elementów karoserii samochodów, grawerowanie numerów seryjnych). Robot ten swoim kształtem przypomina ludzką rękę.

Robot przemysłowy jest przykładem robota stacjonarnego. Sterowanie realizowane jest autonomicznie. Może on działać bez ingerencji człowieka, po uprzednim zaprogramowaniu. Robot ten może wykonywać różne zadania, pomimo że jego konstrukcja nie ulega zmianie. Posiada minimum trzy stopnie swobody. Składa się z manipulatora (łącznie z siłownikami) oraz sterownika (panelu sterowania i dowolnego interfejsu).



Budowa robota przemysłowego

Ze względu na różnorodne zastosowanie robotów, nie istnieje jeden konkretny schemat konstrukcyjny manipulatora przemysłowego. Roboty przemysłowe najczęściej jednak mają postać ramienia mechanicznego, o z góry określonej liczbie **stopni swobody**.

W związku z tym można wyróżnić zespoły i układy stanowiące cały manipulator. W ich skład wchodzi:

1. **Podstawa** – pierwszy, nieruchomy człon manipulatora kinematycznego,
2. **Korpus** – element manipulatora, obudowuje elementy odpowiadające za ruch ramienia,
3. **Ramię** – część układu ruchu, odpowiedzialna za pozycjonowanie przyłącza (końcówki),
4. **Kiść** – część manipulatora, znajdująca się pomiędzy ramieniem a elementem roboczym, jego zadaniem jest orientacja elementu roboczego lub wprowadzanie zmian w jego położeniu,
5. **Chwytnik** – element manipulatora wykorzystywany do chwytania obiektu manipulacji, utrzymania ich i wypuszczenia po zakończeniu zadania.



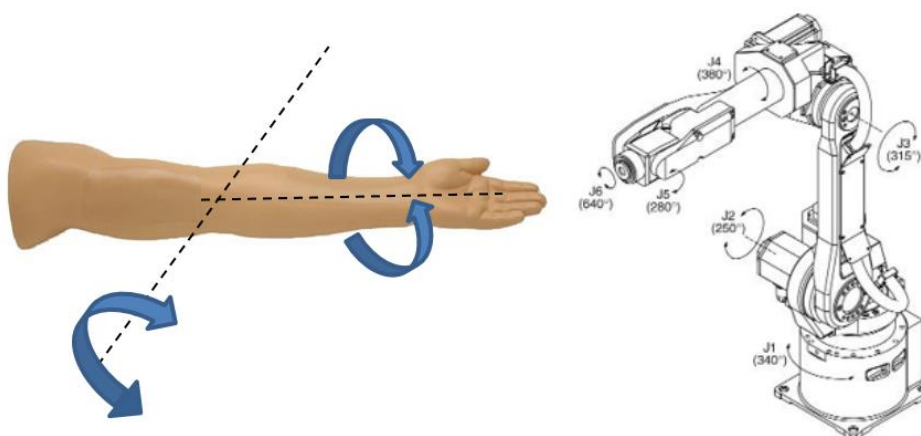
Oprócz wymienionych powyżej elementów w skład robota przemysłowego wchodzi również szafa z układem zasilania i sterowania, często z dodatkowym podręcznym panelem sterowania.



Stopnie swobody – co to właściwie jest?

Stopień swobody jest to zdolność wykonywania ruchu ciała niezależnie od innych ruchów.

Najłatwiej zobrazować je na przykładzie ludzkiego stawu łokciowego. Charakteryzuje się on dwoma stopniami swobody (2 *DOF* – *Degree of Freedom*), gdyż umożliwia poruszanie się w 1 płaszczyźnie oraz umożliwia rotacje wokół 1 osi. Na rysunku poniżej (*po lewej*) przedstawione są kierunki ruchu oraz obroty możliwe dla stawu łokciowego. Zaś na rysunku obok (*po prawej*) został przedstawiony manipulator przemysłowy z zaznaczonymi stopniami swobody. Elementy te, podobnie jak ludzki łokieć, mają możliwość poruszania się w określonych kierunkach oraz dokonywania obrotów. Można więc przyjąć, że robot również posiada swoje stawy ruchowe.





Zastosowanie

Ze względu na przeznaczenie manipulatory dzieli się na dwie grupy. Pierwszą stanowią manipulatory medyczne i rehabilitacyjne. Są to manipulatory mające za zadanie zastąpienie utraconych kończyn człowieka, ich części oraz wsparcie lub trening funkcji niesprawnych kończyn.

Drugą grupę stanowią manipulatory techniczne i to głównie one wykorzystywane są w przemyśle, np. do:

- Odlewnictwa,
- Obróbki cieplnej,
- Spawania,
- Malowania,
- Obsługi maszyn,
- Cięcia,
- Montażu,
- Pakowania,
- Pomiarów.

Ciekawostka



W fabryce Volkswagen Poznań co 109 sekund linię montażową opuszcza samochód.

Średni czas montażu auta od momentu wejścia na linię montażową do otrzymania zezwolenia na wyjazd z fabryki wynosi ok. 12 godzin.

W samej fabryce znajduje się ok. 350 robotów przemysłowych.

Zadanie do wykonania

Zbudowanie z klocków LEGO MINDSTORMS EV3 własnego manipulatora. Instrukcja budowy znajduje się na dołączonej do podręcznika płycie CD. (6. Przemysłowy.pdf)



Programowanie krok po kroku

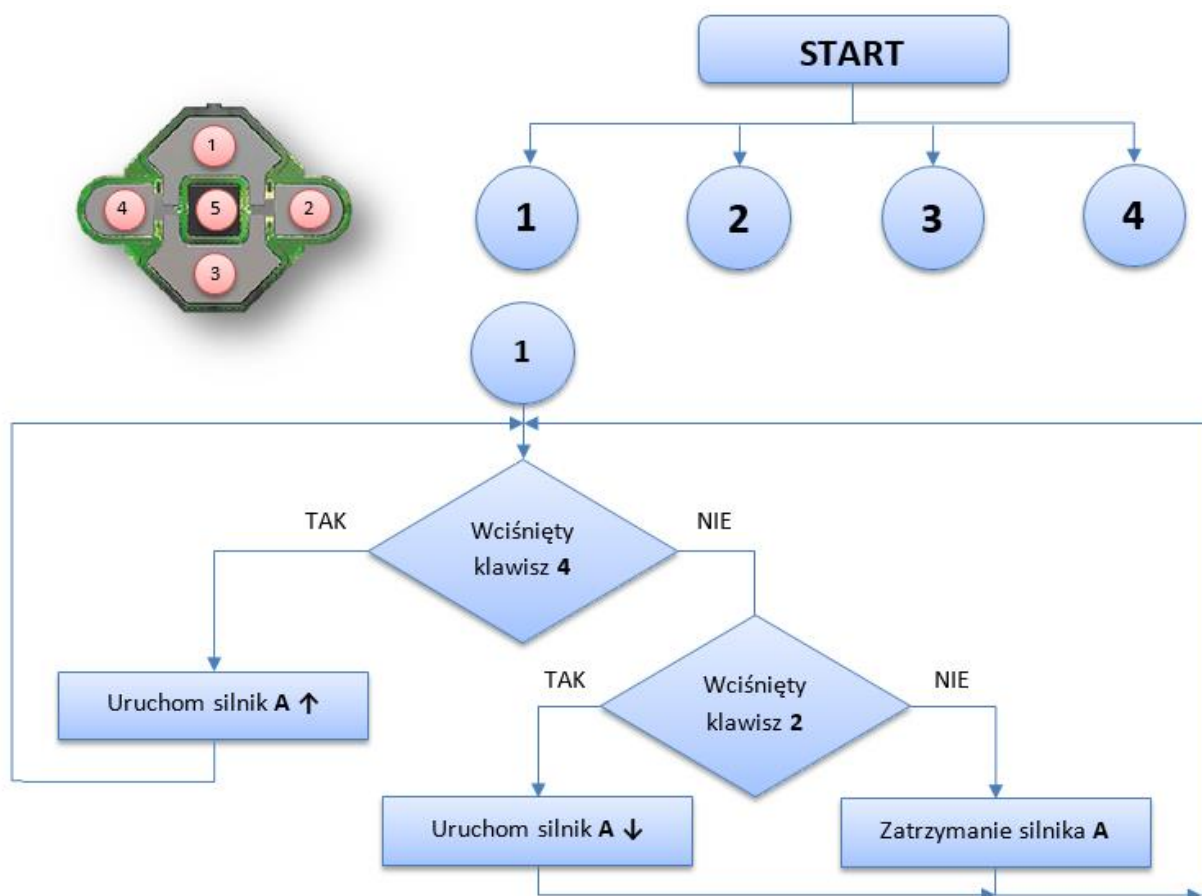
Zadaniem robota przemysłowego będzie przeniesienie lekkiego obiektu (np. klocka) z punktu A do punktu B, bez przemieszczania całej konstrukcji robota. Robot przemysłowy musi być w stanie obniżyć swe ramię, obrócić je o odpowiedni kąt oraz zamknąć chwytak, by pobrać ładunek i ponownie podnieść się do góry. Następnie należy obrócić konstrukcję ramienia i odpowiednio ustawić ją nad miejscem zrzutu, by ramię mogło się zniżyć i ułożyć obiekt. Sterowanie robotem będzie realizowane przy użyciu klawiszy na kostce EV.

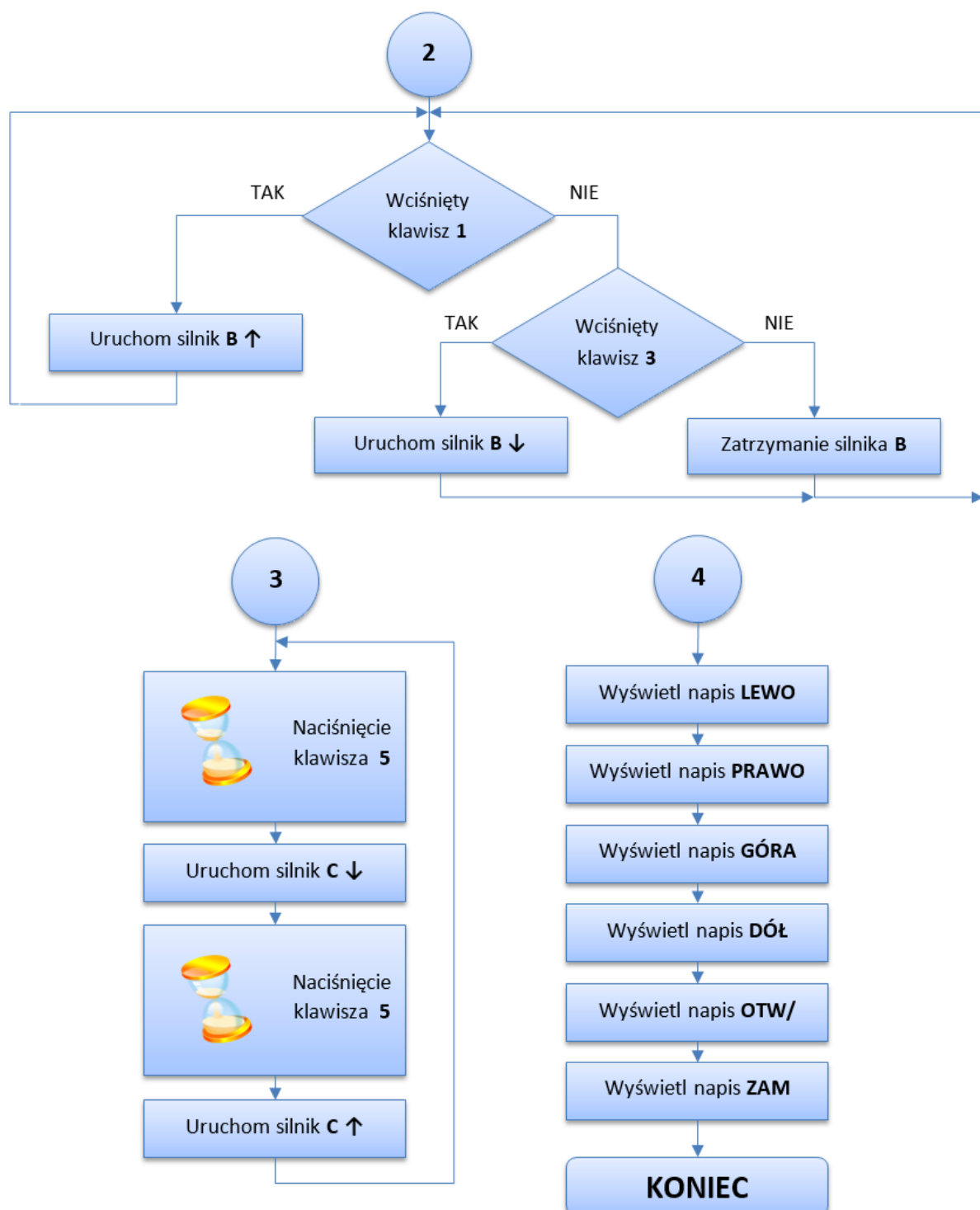
Zadanie to umożliwi zaprezentowanie jednej z zalet programowania zestawu Lego Mindstorms, mianowicie **wielowątkowości**. Jest to zdolność do równoczesnego zarządzania w tym samym czasie dwoma- lub większą liczbą zadań. Realizacja sterowania wymaga bowiem sprawdzania kilku klawiszy na raz i wykonywania wymaganych zadań niezależnie.

Wielowątkowość jest to zdolność programu do równoczesnego wykonywania dwóch lub większej liczby zadań.

Algorytm działania

Do sterowania robotem wykorzystane zostaną przyciski umieszczone na kostce. Na rysunku poniżej umieszczono przyjętą w algorytmie numerację klawiszy.



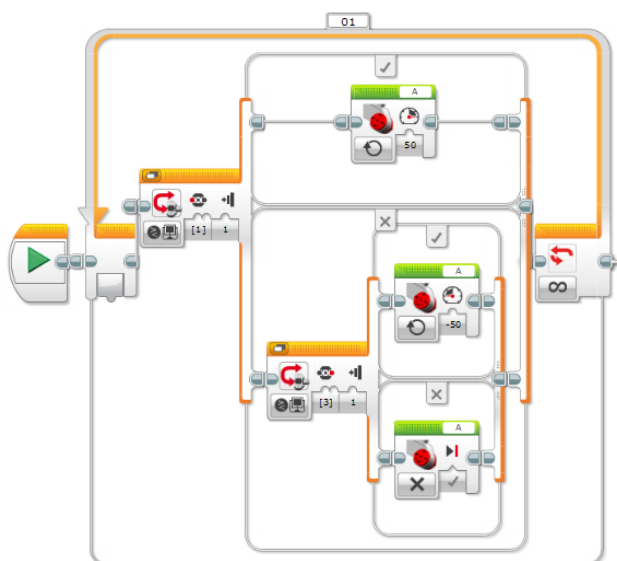


Kod programu

Program realizowany jest zgodnie z algorytmem. Program będzie się składał z niezależnych 4 części. Każda z nich została opisana poniżej.

Krok 1. Program rozpoczyna **blok Startu**.

Część I



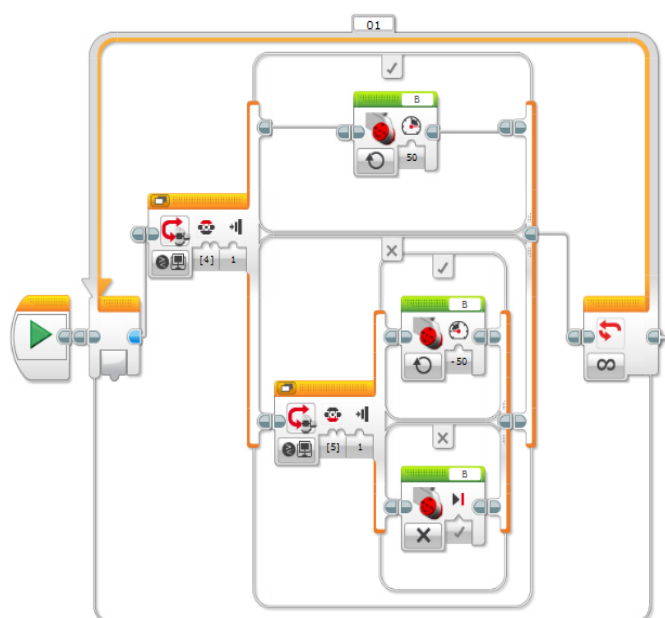
Krok 2. Powtarzalność programu zapewni **pętla programowa (Loop)**.

Krok 3. Blok Instrukcji warunkowej sprawdzi, czy przycisk **4** został wciśnięty (*Switch – Brick Buttons – Compare; Id klawisza: 1, Stan: 1 - Wciśnięty*).

Krok 4. Jeśli wciśnięto klawisz **4**, uruchamiany jest **blok Large Motor** (tryb *On*, Moc: *50 %*).

Krok 5. Nienaciśnięcie klawisza **4** powoduje sprawdzanie stanu klawisza **2** - jeśli zostanie wciśnięty **blok Large Motor** (tryb *On*, Moc: *-50 %*), w przeciwnym wypadku silnik jest zatrzymywany (tryb: *Off*).

Część II



Krok 6. Powtarzalność programu zapewni **pętla programowa (Loop)**.

Krok 7. Blok Instrukcji warunkowej sprawdzi, czy przycisk **1** został wciśnięty (*Switch – Brick Buttons – Compare; Id klawisza: 1, Stan: 1 - Wciśnięty*).

Krok 8. Jeśli wciśnięto klawisz **1**, **blok Large Motor** uruchamia silnik (tryb: *On*, Moc: *50%*).

Krok 9. Nienaciśnięcie klawisza **1** powoduje sprawdzanie stanu klawisza **3** - jeśli zostanie on wciśnięty, **blok Large Motor** uruchamia silnik (tryb *On*, Moc: *-50 %*). W przeciwnym wypadku silnik jest zatrzymywany (tryb: *Off*).

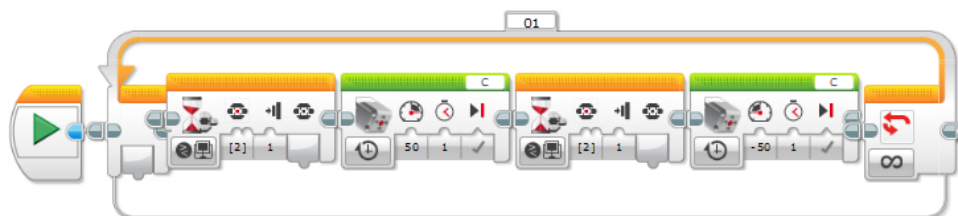
Część III

Krok 10. Powtarzalność programu zapewni **pętla programowa (Loop)**.

Krok 11. Część ta oczekuje na naciśnięcie klawisza **5**. W tym celu wykorzystuje się **Blok Oczekiwania (Wait – Brick Buttons - Compare)**.

Krok 12. Jeśli wciśnięto klawisz **5**, **blok Medium Motor** uruchamia silnik średni na 1 sekundę.

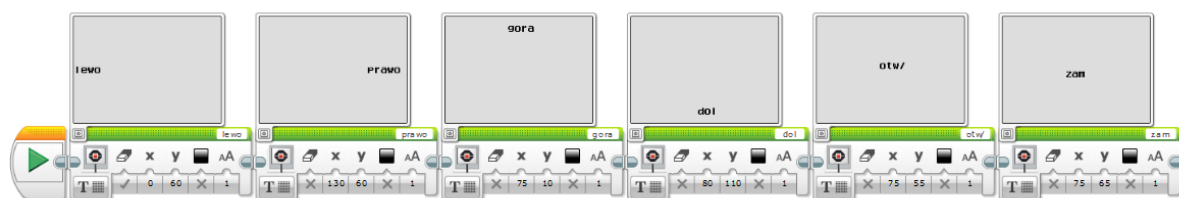
Krok 13. Następnie oczekuje się na ponowne wciśnięcie klawisza **5**. Powoduje on ruch silnika średniego w przeciwną stronę o ten sam czas, z tą samą mocą.



Część IV

Zadaniem tej części programu jest wyświetlenie opisów do klawiszy, którymi robot będzie sterowany.

Krok 14. Blok Display (tryb: Text – Pixel) będzie powtarzany by wyświetlić opisy klawiszy (*Lewo*, *Prawo*, *Góra*, *Dół*, *Otw/Zam*). Różnicę się będą jedynie współrzędnymi **X** i **Y**.





Labirynt



Labirynt jest budowlą odznaczającą się zawiłym układem, z dużą liczbą pomieszczeń i krętych korytarzy. Budowanie go miało na celu utrudnienie dostępu do pomieszczenia znajdującego się w jego wnętrzu lub zaraz za nim. Historia labiryntów sięga do czasów Mezopotamii oraz starożytnego Egiptu, gdzie stosowano jego strukturę przy budowie królewskich grobowców. Labirynty stosuje się również w architekturze zieleni (w parkach, ogrodach). Współcześnie, buduje się labirynty dla robotów – są jedną z konkurencji na zawodach robotów.

Zadanie do wykonania

Zbudowanie robota i zaprogramowanie go, tak by był w stanie przejechać przez labirynt.

Budowa robota

W celu zbudowania robota pokonującego labirynty trzeba wziąć pod uwagę kilka rzeczy:

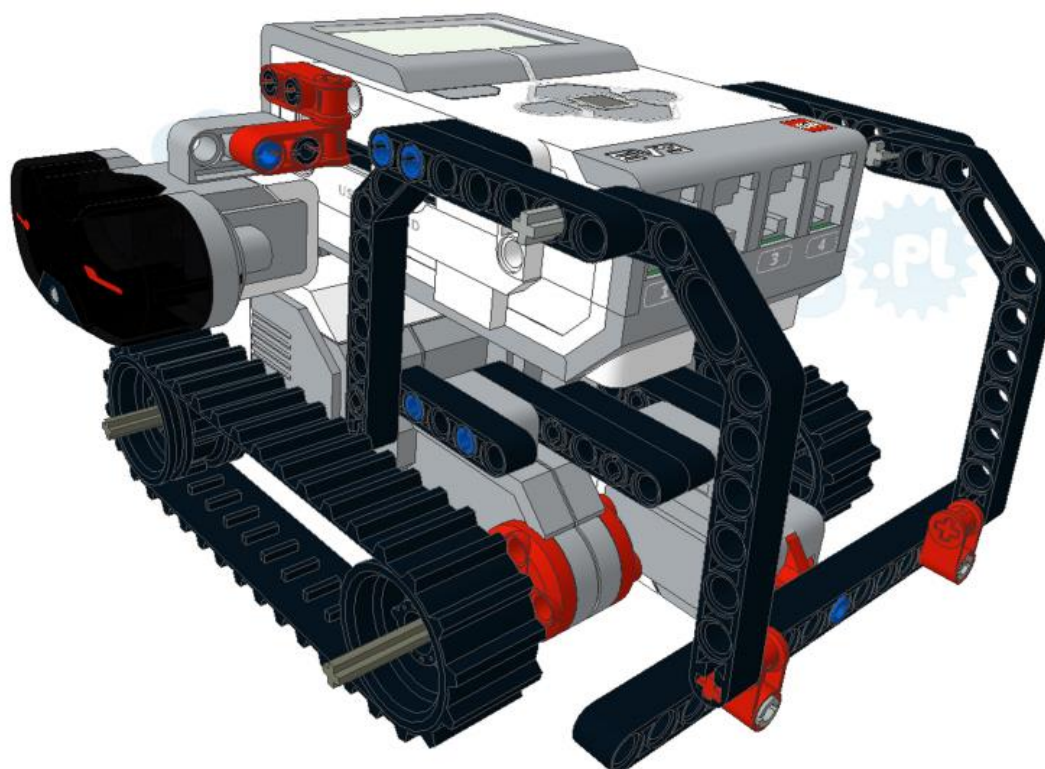
- Robot musi być mobilny,
- Musi mieć odpowiednie czujniki umożliwiające podążanie w odpowiednim kierunku,
- Powinien mieć wgrany program zawierający odpowiedni algorytm działania.



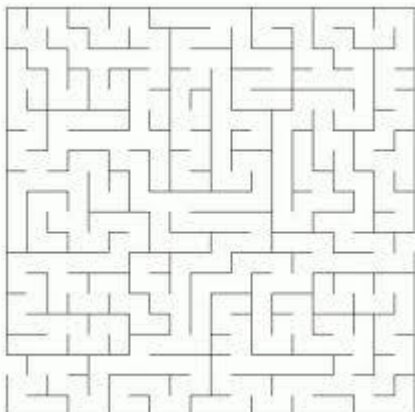
Mobilność robota zapewnią dwa silniki, które będą napędzać koła lub gąsienice. Robot pokonujący labirynt zostanie wyposażony w dwa czujniki:

- Czujnik dotyku w postaci zderzaka wykryje dojazd do ściany
- Czujnik odległości – wykrywający ścianę lub jej brak z prawej strony robota.

Na rysunku poniżej przedstawiony został robot, który zostanie zbudowany w trakcie zajęć. Instrukcja budowy została umieszczona na płycie CD dołączonej do książki (7. *Labirynt.pdf*).



Programowanie krok po kroku

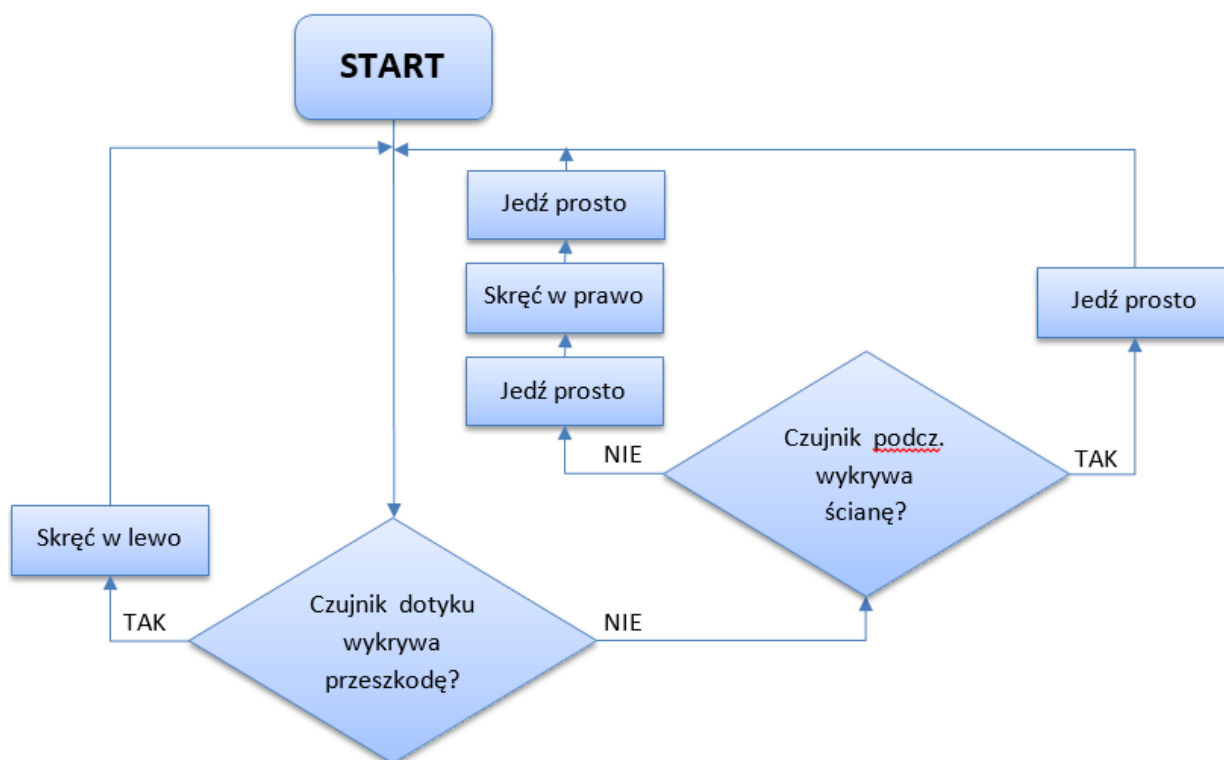


W jaki sposób można pokonać labirynt? Każdy napotkany może być inny. Czy istnieje jakiś ogólny algorytm, którego zastosowanie pozwoli robotowi poradzić sobie z każdą przeszkodą?

Otóż istnieje algorytm, który mówi, że należy się poruszać cały czas wzdłuż lewej lub prawej ściany.

Zastanówmy się, czy to faktycznie pomoże rozwiązać każdy labirynt. Jeśli tak, zastanów się, na jakiej zasadzie powinien działać robot, zbudowany w trakcie zajęć.

Algorytm działania



Krok 1. Robot sprawdza, wykorzystując czujnik dotyku, czy na jego drodze znajduje się przeszkoda. Jeśli **TAK** – robot skręca w lewo. Jeśli czujnik **NIE** wykrywa przeszkody, sprawdzany jest czujnik podczerwieni. Gdy czujnik wykryje przeszkodę (ścianę), robot jedzie do przodu, w przeciwnym wypadku (**NIE** - czujnik nie wykrywa ściany) robot jedzie do przodu, obraca się w prawo i ponownie jedzie do przodu.

Krok 2. Działając w pętli, ponownie sprawdzane są powyższe warunki.



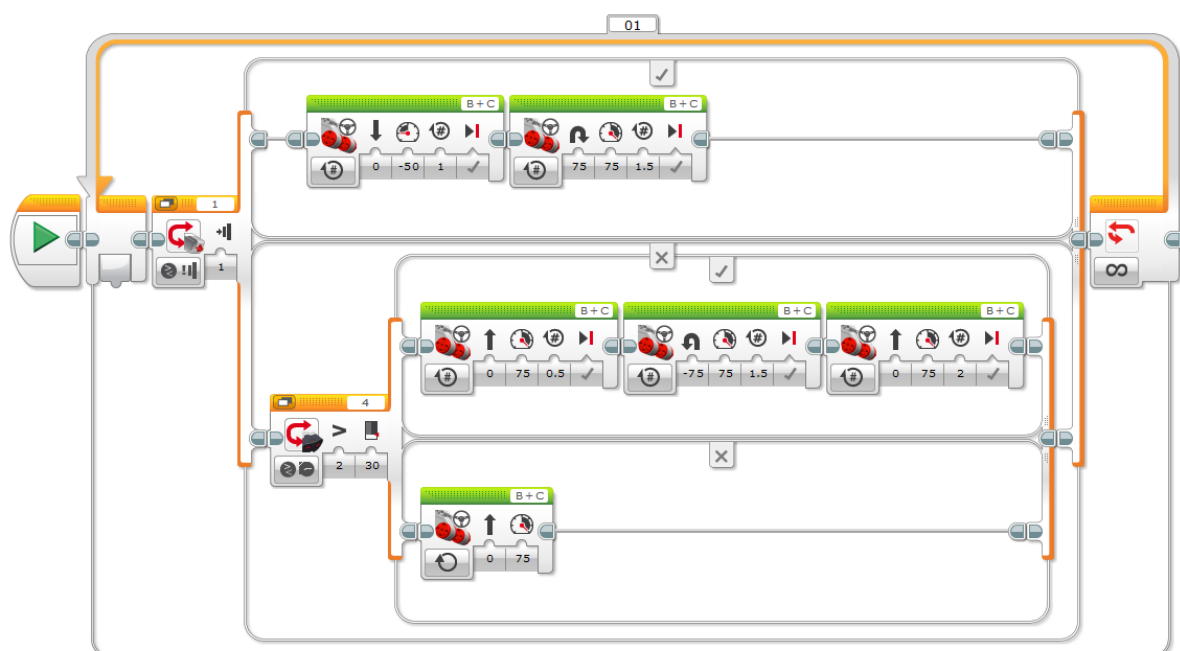
Kod programu dla robota pokonującego labirynt

Krok 1. Kod programu rozpoczyna **blok Startu**.

Krok 2. Algorytm realizowany jest nieprzerwanie – wykorzystana jest **Pętla**.

Krok 3. Blok Instrukcji warunkowej (Switch) umożliwia realizację dwóch niezależnych zadań w zależności od wyniku danych zebranych z czujnika dotyku.

Krok 4. W przypadku niespełnienia pierwszego warunku (wykrycia przeszkody przez czujnik dotyku), za pomocą kolejnej **Instrukcji warunkowej** sprawdzany jest czujnik podczerwieni.





Pozytywka



Pozytywka jest to automatyczny instrument muzyczny, który generuje dźwięk. Jest to możliwe dzięki specjalnej konstrukcji pozytywki. Znajdują się w niej metalowe wypustki, umieszczone na wałku obrotowym, które zaczepiają o zęby metalowego grzebienia. Powstanie pozytywki datuje się na przełom XIX i XX wieku. Jest uroczą zabawką dla dzieci, które zachwycają się mechanizmem muzyki uruchamiane po otwarciu pozytywki. Jednak, aby wydobyć z niej dźwięk, trzeba ją wcześniej nakręcić.





Zadanie do wykonania

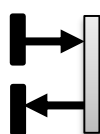
Zadaniem do wykonania będzie zbudowanie robota, który będzie odgrywać dźwięk, w zależności od wykrytego przez czujnik koloru. Działanie pozytywki sprawdzimy przy pomocy kolorowego paska dołączonego do książki.



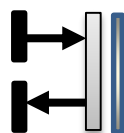
Zasada działania robota pozytywki ma swoje odzwierciedlenie w czujnikach używanych przez automatyków. Czujniki optoelektroniczne są elementami, które działają za zasadzie wysyłania przez nadajnik i odbierania przez odbiornik wiązki promieni świetlnych. Reagują one na przeszkody, które przecinają tę wiązkę światła lub wiązkę odbitą od obiektu. Znajdują one zastosowanie m. in. w:

- Kontroli położenia części
- Identyfikacji obiektów
- Określaniu poziomu cieczy lub innych materiałów.

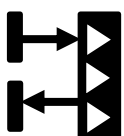
Rozróżniamy kilka rodzajów czujników optycznych:



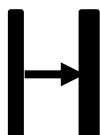
Odbiciowe – czujniki, które w jednej obudowie zawierają zarówno odbiornik, jak i nadajnik, bez potrzeby użycia reflektora.



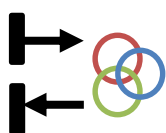
Odbiciowe z eliminacją wpływu tła – umożliwiają wykrywanie zarówno jasnych, jak i ciemnych obiektów, przy czym strefa ich działania jest regulowana. Wykorzystuje się je do wykrywania asortymentu przemieszczającego się w pobliżu podłoża, którego czujnik nie powinien wykrywać oraz elementów ze szkła czy materiałów transparentnych.



Refleksyjne – czujniki te wymagają dodatkowego reflektora. Umieszcza się go na osi wiązki światła wysyłanej przez czujnik. Czujniki te wykrywają obiekty znajdujące się pomiędzy czujnikiem a reflektorem. Niewątpliwym plusem tych czujników jest większy zasięg działania.



Bariera – czujnik ten składa się z dwóch osobnych elementów: nadajnika i odbiornika, które znajdują się wzdłuż osi wyznaczonej przez wiązkę nadajnika. Wykrywają wszystkie obiekty znajdujące się pomiędzy nadajnikiem i odbiornikiem. Obiekt zostaje wykryty w momencie przecięcia wiązki podczerwieni wysyłanej przez nadajnik. Ich zaletą jest najdłuższa strefa działania.



Kontrastu – czujniki umożliwiające wykrywanie koloru badanego obiektu w stosunku do tła.
Koloru – wykrywają barwę badanego obiektu.

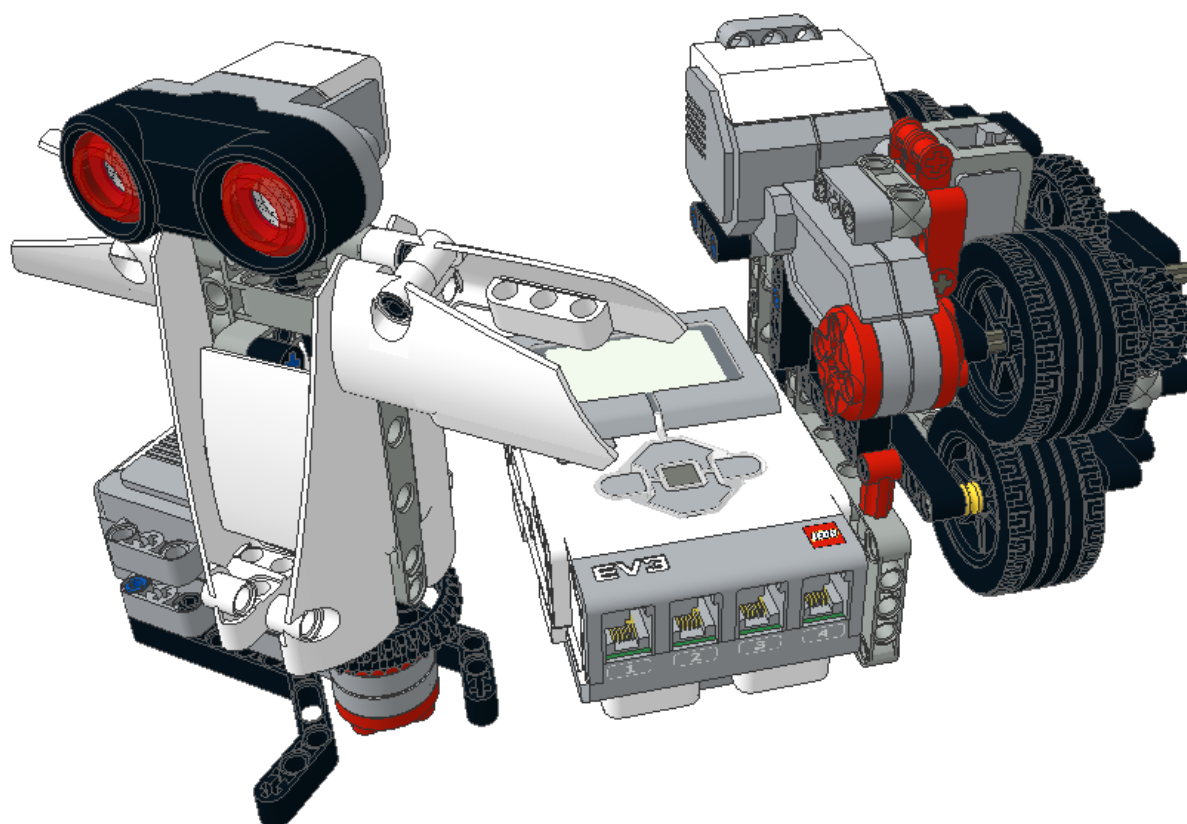


Budowa robota

Do budowy robota wykorzystamy m.in.:

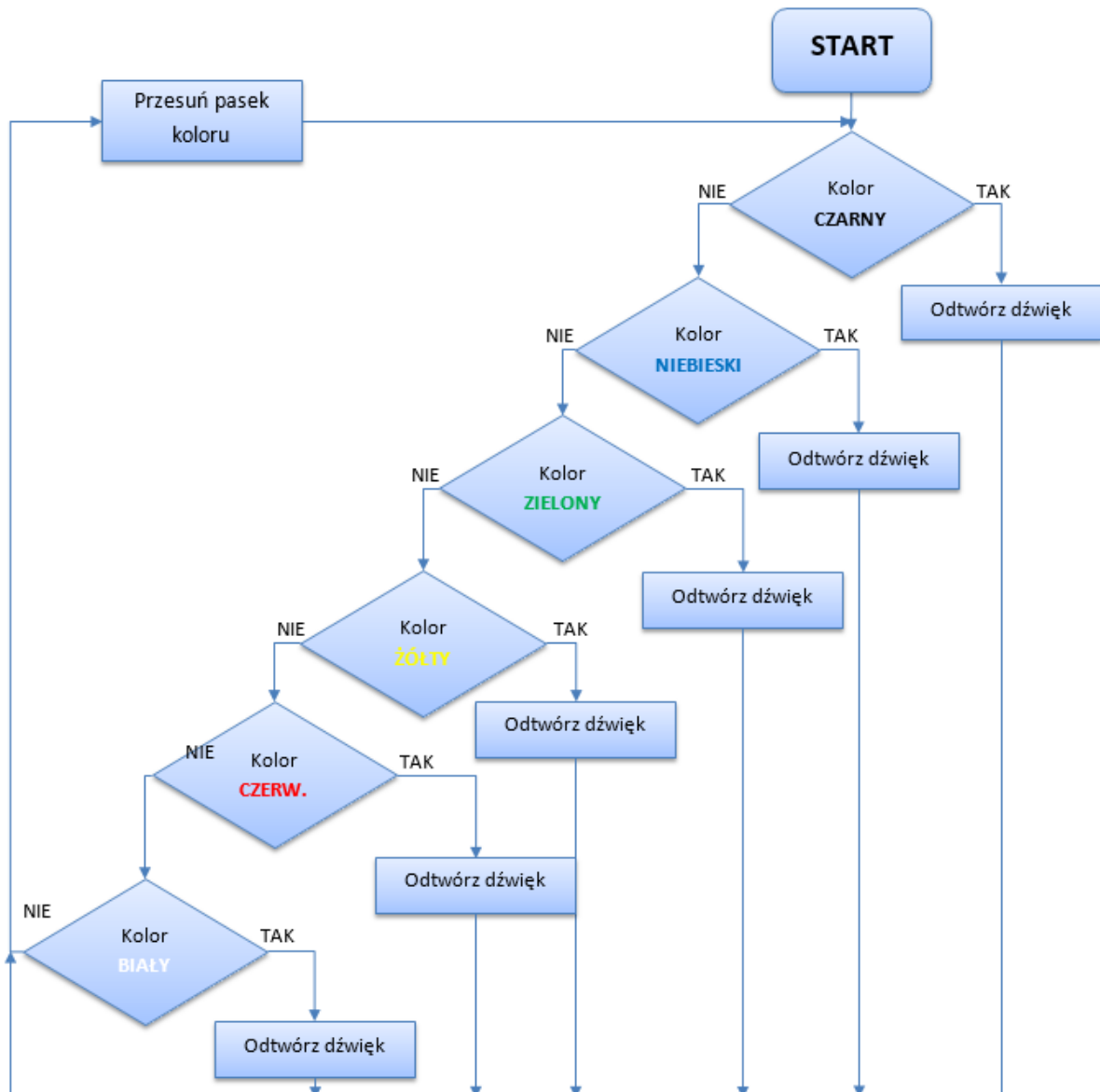
- Kostkę EV3,
- Duży silnik,
- Czujnik koloru
- Przekładnię mechaniczną.

Kłoczek EV3 będzie odpowiedzialny za odtwarzanie muzyki, silnik za ruch przekładni mechanicznej, która będzie wprawiała w ruch oponę. Zadaniem opony będzie podtrzymanie i przesunięcie paska kolorów. W zależności od wykrytego koloru pozytywka wygra inny dźwięk. Instrukcja budowy robota znajduje się na płycie CD dołączonej do książki (*8. Pozytywka.pdf*). Rysunek poniżej przedstawia gotową konstrukcję.



Programowanie krok po kroku

W programie wykorzystamy **Instrukcję Warunkową** uzależnioną od czujnika koloru. Całość umieszczona zostanie w **pętli**.





Algorytm działania

Krok 1. Sprawdzenie koloru paska przez czujnik.

Krok 2. Odtworzenie dźwięku na podstawie wykrytego koloru.

Krok 3. Przesunięcie paska.

Krok 4. Powtórzenie powyższych kroków.

Kod programu dla robota Pozytywki

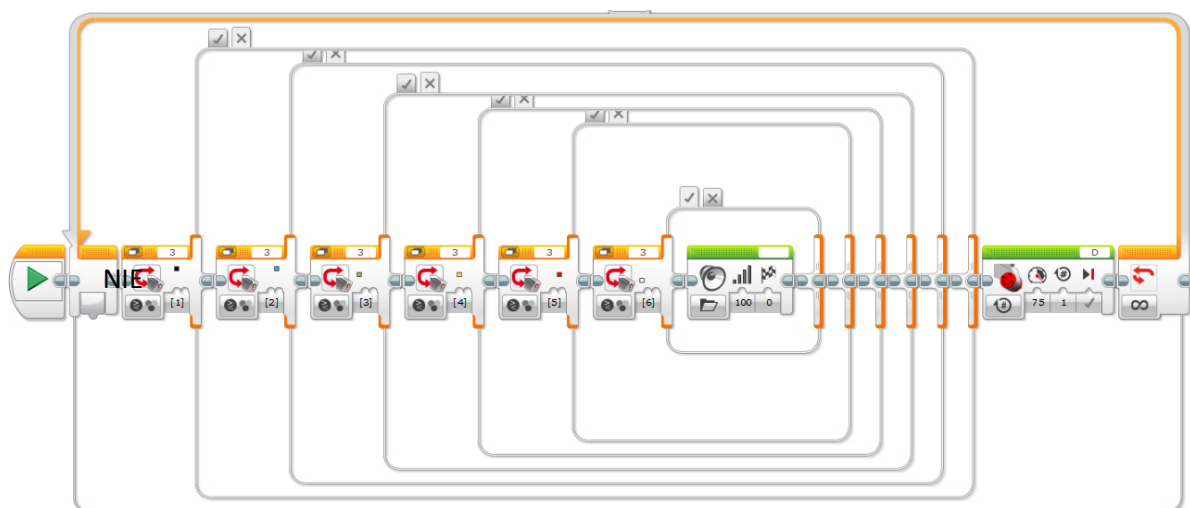
Krok 1. Kod programu rozpoczyna **blok Startu**.

Krok 2. Ponieważ program ma być powtarzany na okrągło dodajemy blok Pętli (**Loop**).

Krok 3. W celu sprawdzania koloru wykrytego przez czujnik stosuje się instrukcję warunkową. Jeśli wykryje kolor np. CZARNY – odtworzy melodię X, w przeciwnym wypadku poszukuje innego koloru.

Krok 4. Proces opisany w Kroku 3. będzie powtarzany dla wszystkich kolorów.

Krok 5. Po odnalezieniu koloru i wygraniu dźwięku, uruchamiany jest silnik odpowiedzialny za przesunięcie paska, a algorytm zostaje powtórzony.



Sumo

Co to jest robot sumo?



Sumo to sport, który wywodzi się z Japonii. Jego początki datowane są na VIII wiek p.n.e.. Dosłowne tłumaczenie słowa sumo to „szybkie przeciwdziałanie uderzeniom przeciwnika”. Walka zawodników odbywa się na podwyższonym ringu, w kształcie koła, zwanym *dohyō*. Jaki związek ma ten sport z robotyką? Robotycy zapożyczyli ten sport, przystosowując go do własnych warunków. Na ring wystawia się dwa roboty, których zadaniem jest wypchnięcie się poza *dohyō*. Roboty wykorzystywane w tym sporcie nazywane są **robotami sumo**. Są one mobilne oraz autonomiczne.

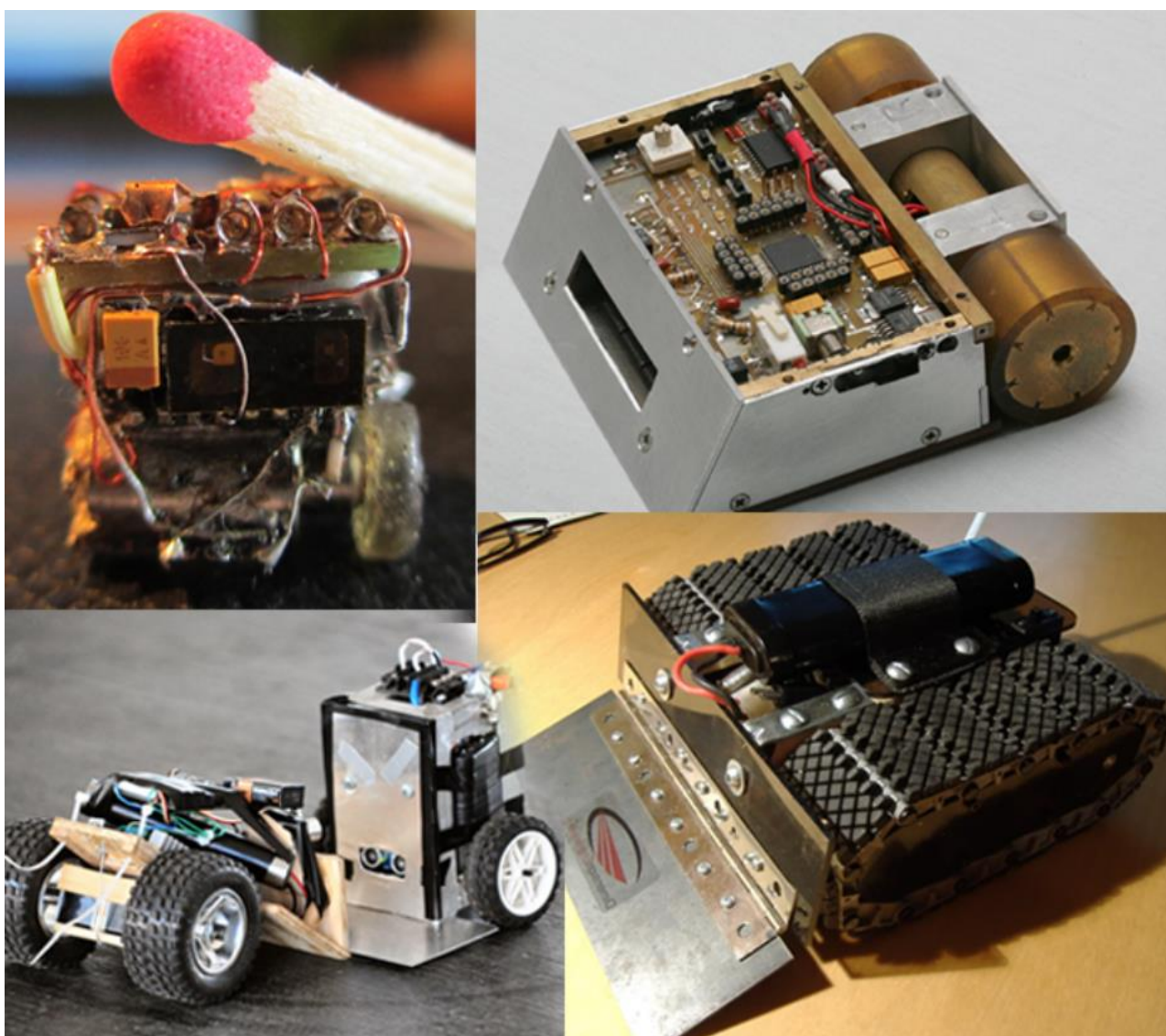


Głównym zadaniem robota, a przy okazji wyzwaniem dla konstruktora, jest odnalezienie przeciwnika i wypchnięcie go poza ring. Należy jednak uważać, aby robot spychający sam poza niego nie wypadł.

W tym celu wykorzystuje się różne czujniki: podczerwieni (koloru) czy ultradźwiękowe. Każdy robot powinien spełniać warunki ściśle określone w regulaminie.

Zawody Sumo robotów podzielone zostały na kilka kategorii określających rozmiar i wagę:

1. **Sumo** – waga do 3 kg, rozmiar 20 cm x 20 cm,
2. **Mini – sumo** – waga do 500 g, rozmiar 10 cm x 10 cm,
3. **Micro – sumo** – waga do 100 g, rozmiar 5 cm x 5 cm x 5 cm,
4. **Nano – sumo** – rozmiar 2,5 cm x 2,5 cm x 2,5 cm,
5. **Femto – sumo** – rozmiar 1 cm x 1cm x 1cm.



Zadanie do wykonania

Zbudowanie robota biorącego udział w zawodach Sumo.

Przebieg zawodów

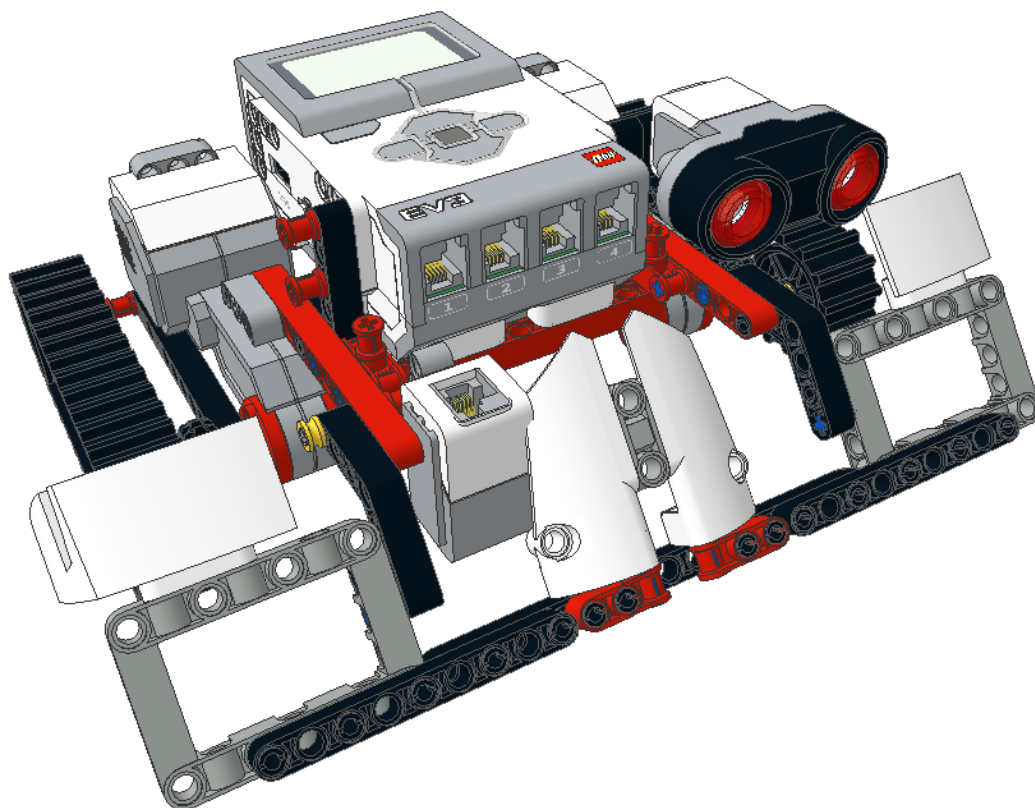
Zbudowane w trakcie zajęć roboty sumo są wystawiane na ringu i ustawiane naprzeciw siebie. Na znak sędziego roboty zostają uruchomione, a zawodnicy odsuwają się od ringu. Od tego momentu roboty zdane są tylko na własne czujniki oraz program napisany przez programistę.

Budowa robota

Konstrukcja robota sumo zależy tylko i wyłącznie od inwencji twórczej konstruktora. Ważne jest jednak, aby posiadał:

- Czujnik koloru, sprawdzający, czy robot nie wypadł z ringu,
- Czujnik podczerwieni do wykrycia przeciwnika,
- Dwa silniki duże napędzające koła i umożliwiające obrót robota.

Na rysunku poniżej przedstawiono robota sumo, który zostanie zbudowany w trakcie zajęć. Instrukcja budowy znajduje się na płycie CD dołączonej do książki (9. *Sumo.pdf*).



Programowanie krok po kroku

Analizując działanie robota sumo należy zadać pytanie o to, co jest ważniejsze – wykrycie przeciwnika, czy utrzymanie się w ringu. Oczywiście najważniejsze jest utrzymanie się na ringu, a dopiero później wykrycie przeciwnika. Jeśli czujniki przekażą informację o znalezieniu przeciwnika (**TAK**), robot zaczyna poruszać się wprost na przeciwnika, jeśli nie – robot wykonuje obrót o pewien kąt, w dalszym ciągu przeszukując teren. W momencie wykrycia koloru BIAŁEGO, robot wycofuje się i wykonuje obrót o pewien kąt α .

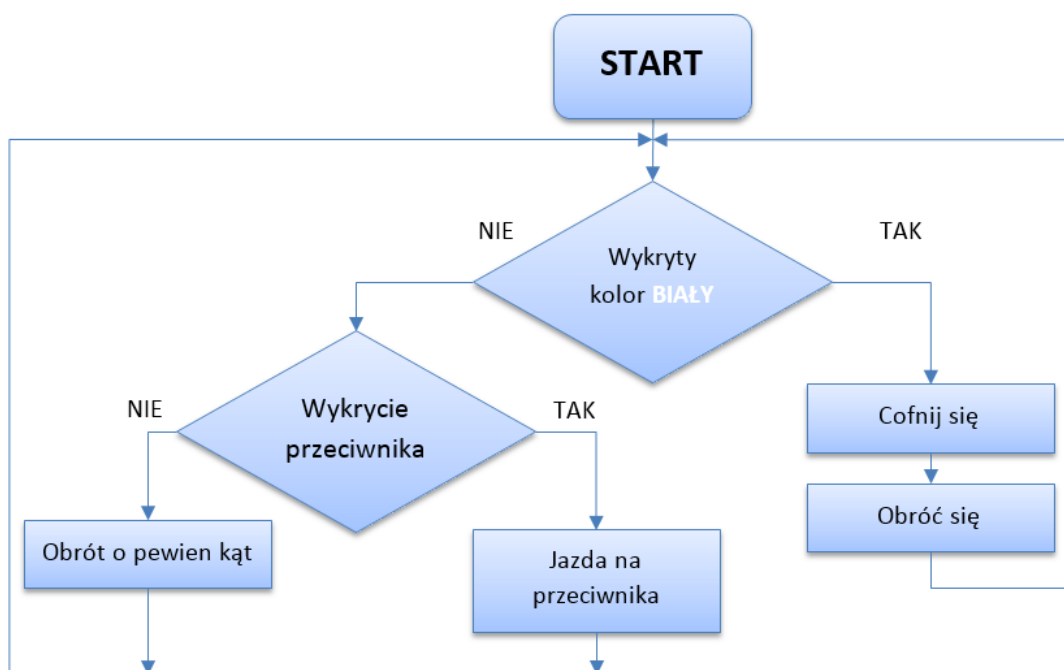
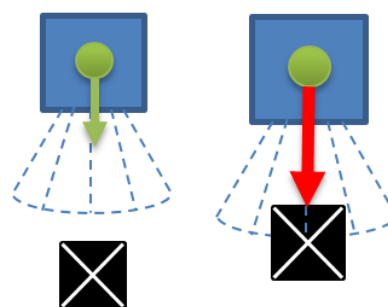
Algorytm działania

Krok 1. Sprawdzenie czujnikiem koloru, czy robot nie najechał na białą linię.

Krok 2. Sprawdzenie, czy czujnik podczerwieni wykrywa przeciwnika. Jeśli **NIE**, robot wykonuje obrót o pewien kąt. Jeśli **TAK**, z całą mocą silników rusza na przeciwnika.

Krok 3. Należy również rozpatryć przypadek, gdy robot najeżdża na białą linię. Wówczas powinien zostać przełączony kierunek działania silników, umożliwiając cofnięcie się robota, a następnie należy obrócić robota.

Warto pamiętać, że głównym celem tego sportu jest utrzymanie się na *dohyō*, więc w przypadku, gdy czujnik wykryje zarówno intruza, jak i białą linię, priorytetem powinno być jak najszybsze odsunięcie się od granicy ringu, a następnie natarcie na przeciwnika.



Kod programu dla robota klasy SUMO

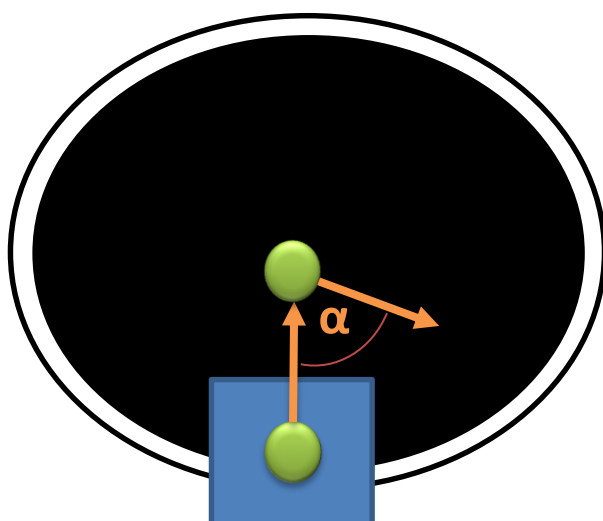
Krok 1. Program rozpoczynamy od bloku Startu.

Krok 2. Program realizowany będzie w pętli.

Krok 3. Pierwszym elementem, który będzie sprawdzany, jest stan czujnika koloru. W zależności czy kolor BIAŁY zostanie wykryty lub nie, realizowane będą inne działania. W związku z tym stosuje się instrukcję warunkową umożliwiającą rozdzielenie tych kroków. Oczywiście musi zostać odpowiednio ustawiony blok instrukcji – wybór trybu działania (*Color Sensor – Compare – Color*) – oraz kolor, jaki ma być wykrywany.

Krok 4. Jeśli kolor BIAŁY został wykryty robot powinien się wycofać i obrócić.

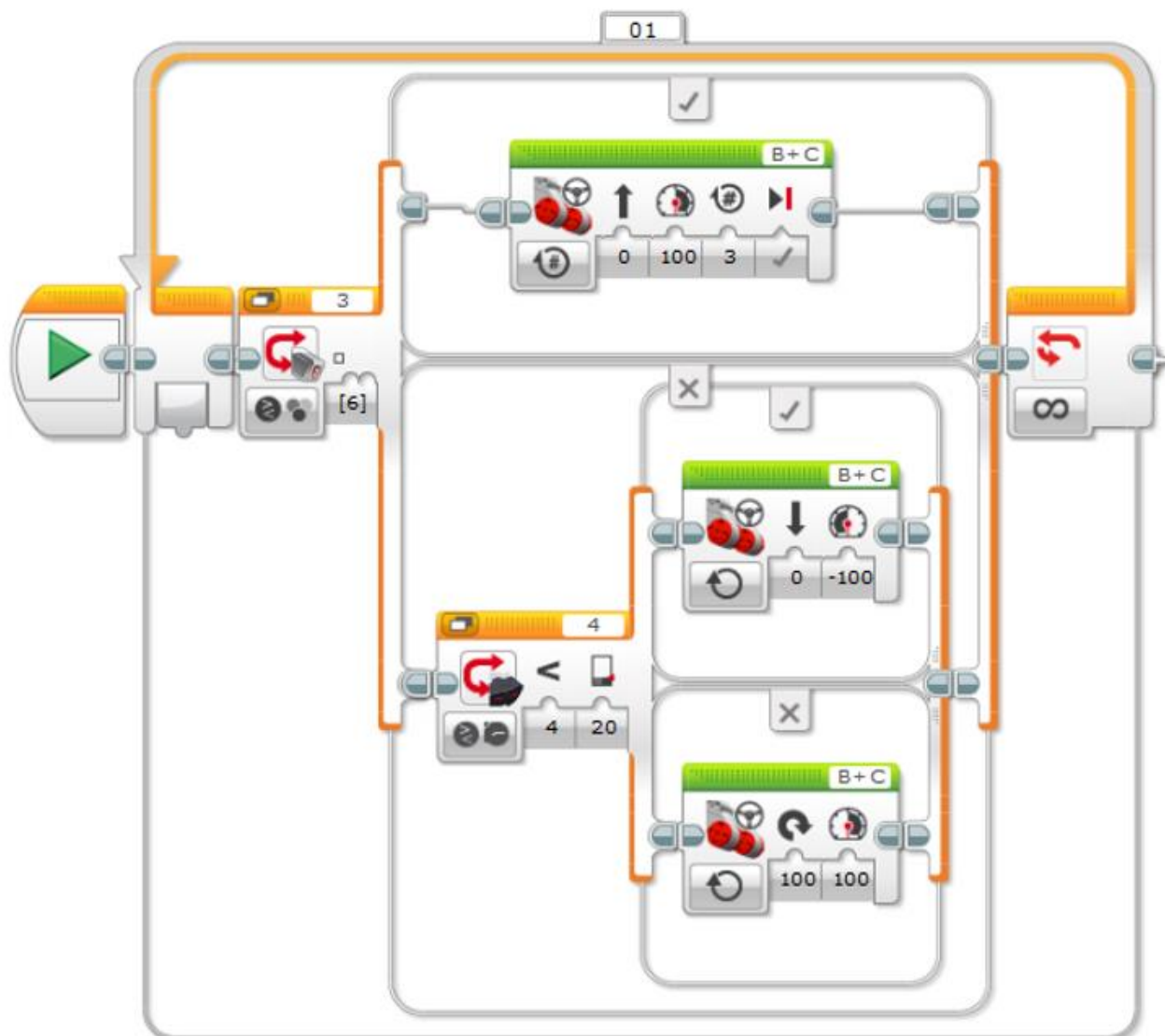
Krok 5. W Przeciwnym wypadku należy sprawdzić stan czujnika podczerwieni wykrywający przeciwnika. Również w tym przypadku należy



skorzystać z instrukcji warunkowej. Ustawia się tryb działania (*Infrared Sensor – Compare - Proximity*) oraz tryb porównania i odległość od przeciwnika.

Krok 6. Jeśli warunek jest spełniony (przeciwnik wykryty) należy natychmiast uruchomić silniki – jazda na przeciwnika.

Krok 7. W przeciwnym wypadku, silniki powinny obrócić robota, by z nowej pozycji sprawdzał powyższe kroki od początku.





Sortownica



Zakłady przemysłowe (produkcja słodyczy, samochodów, hurtownie owoców i warzyw) często wykorzystują proces sortowania w swoich fabrykach. Polega on na uporządkowaniu elementów względem pewnych charakterystycznych cech. Zadanie to wymaga dużej dokładności, a sam proces powinien przebiegać jak najszybciej. Kiedyś zadanie to było wykonywane przez człowieka, ale wraz z postępem nauki można było ułatwić i przyspieszyć proces produkcyjny, wykorzystując roboty. Dzięki temu, człowiek nie musi wykonywać zajęć monottonnych, męczących i czasochłonnych.





Jednymi z najczęściej stosowanych czujników do sortowania są **czujniki wizyjne**. Składają się one z:

- Kamery,
- Procesora,
- Oświetlacza.

Wykorzystywane są do odczytywania kodów kreskowych, wykonywania nieskomplikowanych pomiarów (koloru, gabarytu) czy do rozpoznawania obecności obiektów w zbiorze.

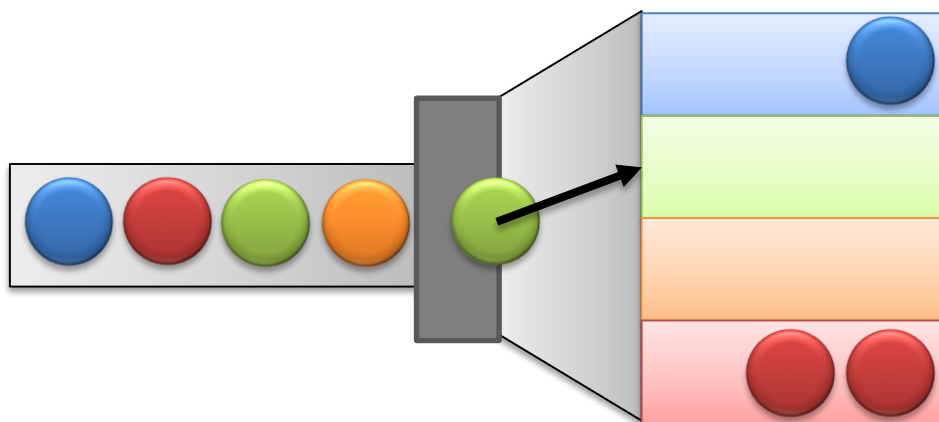
Ciekawostka

Proces sortowania wykorzystywany jest również w bankowości. Sortownie gotówki, były kiedyś w każdej pomniejszej placówce. Miały za zadanie zliczać banknoty i weryfikować je. Wiązało się to niestety z małą sprawnością, dużą pracochłonnością oraz wysokimi kosztami eksploatacji. Z tych względów większość banków zdecydowała się zlikwidować własne sortownie i zlecić tę pracę firmom przetwarzającym gotówkę, typu Cash In Transit (CIT). Obecnie banknoty weryfikowane są pod kątem:

- Rodzaju,
- Autentyczności
- Jakości (czujniki wykrywają plamy, brud, napisy, taśmy klejące czy ubytki w banknotach).

Zadanie do wykonania

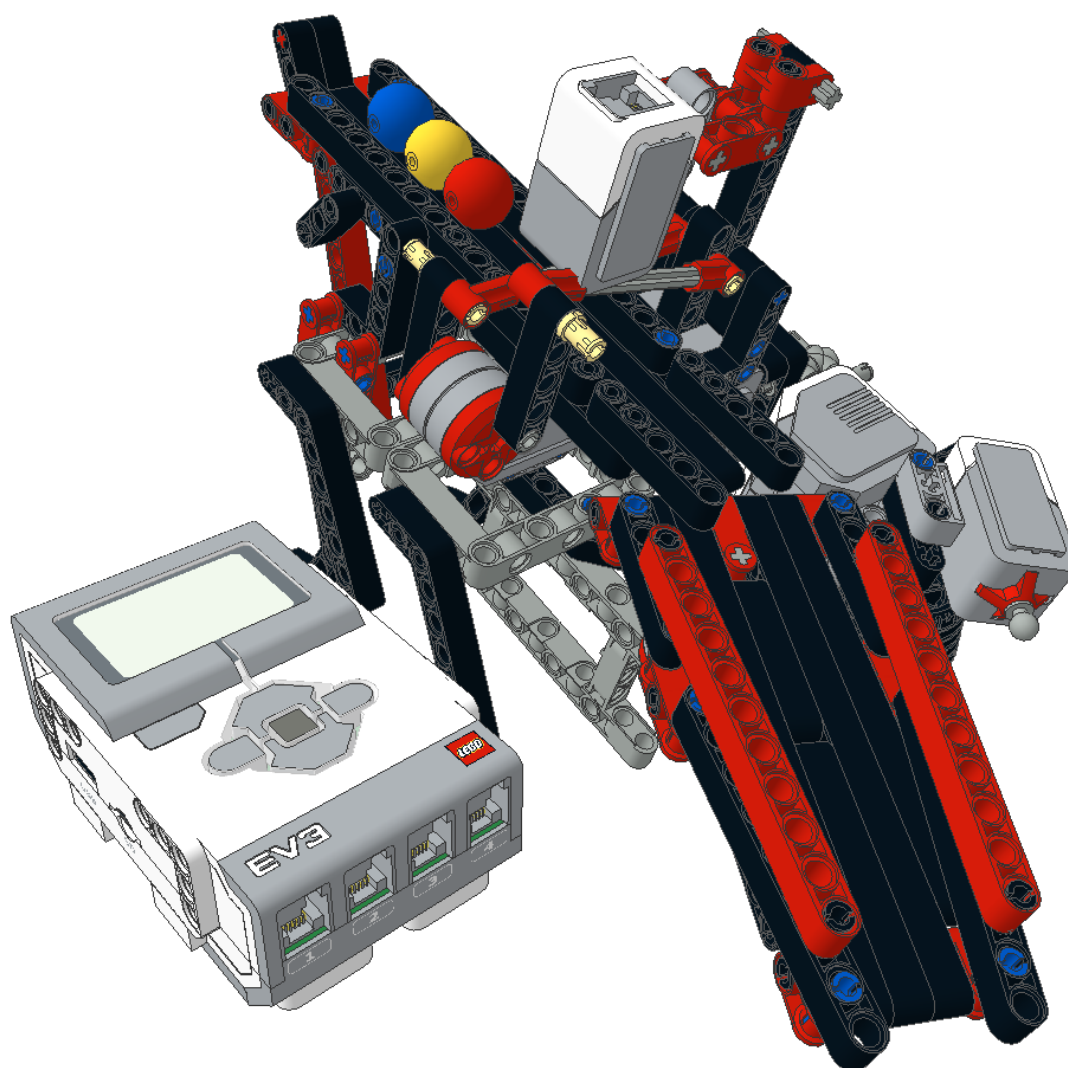
Robot, który zostanie zbudowany w trakcie zajęć będzie miał za zadanie posegregować kolorowe piłeczki ułożone w podajniku. Czujnik koloru rozpoznawszy barwę kulki przesyła otrzymaną informację do kostki EV3, która ustawia odpowiednią drogę zrzutu dla piłeczki.



Budowa robota

Robota zbudujemy według instrukcji dołączonej do płyty CD. (10. Sortownica.pdf) Na zdjęciu poniżej przedstawiono gotową konstrukcję robota. Robot zostanie wyposażony w:

- Kostkę EV3,
- Czujnik koloru,
- 2 duże silniki.



Programowanie krok po kroku

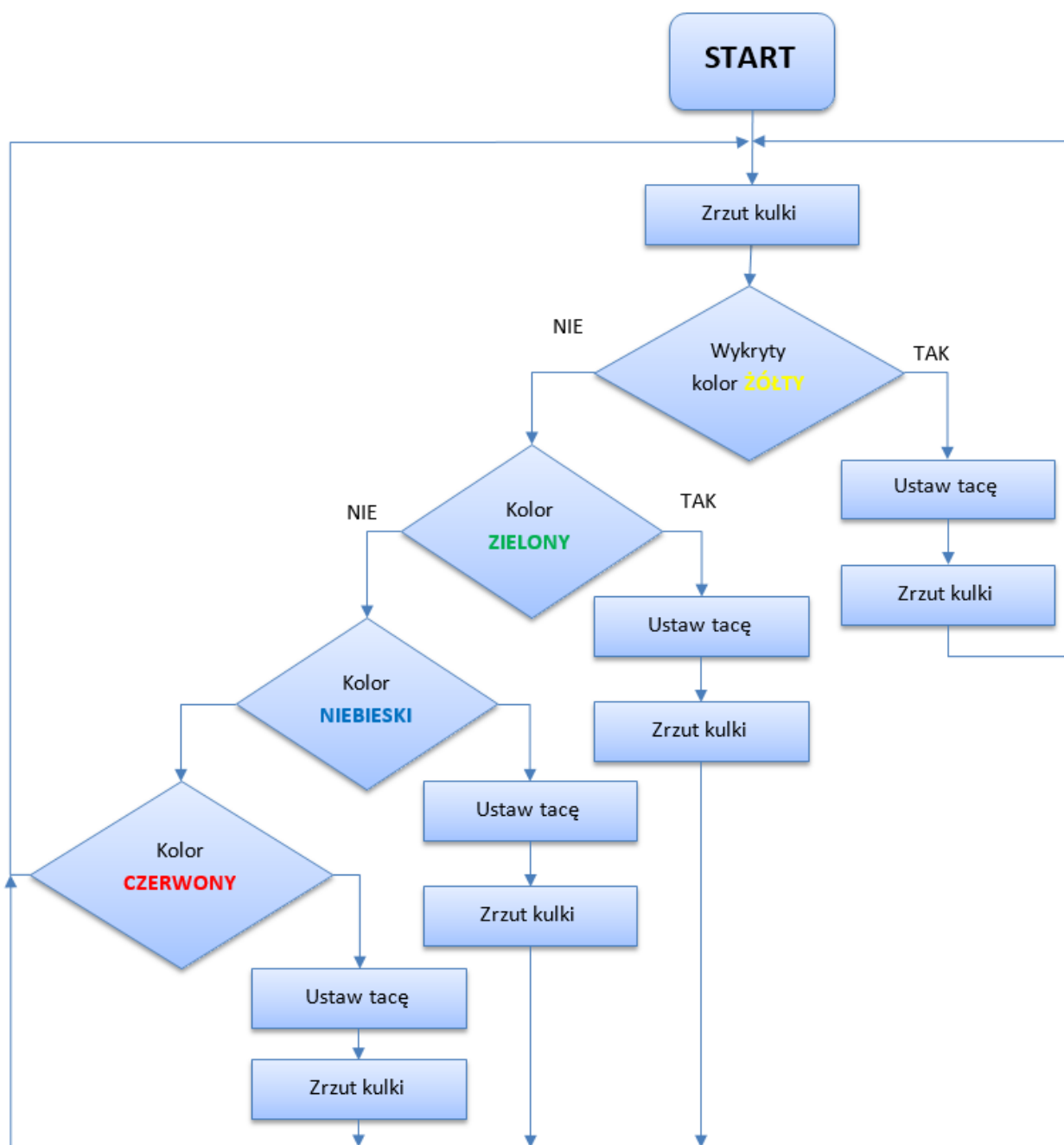
Algorytm działania

Krok 1. Aby kulka wpadła pod czujnik koloru, wykorzystany zostanie silnik nr 1.

Krok 2. Czujnik koloru rozpoznaje kolor piłeczki.

Krok 3. Silnik nr 2 ustawia tacę w zależności od wykrytego wcześniej koloru.

Krok 4. Silnik nr 1 przesuwa piłeczkę, aby wpadła na tacę.





Roboślacz

Jednym z atutów robotów jest fakt, że są w stanie udźwignąć zdecydowanie większy ciężar, niż człowiek. Odciąża się w ten sposób człowieka, przyspiesza i ułatwia pracę. Roboty te często stosuje się w dużych fabrykach do przewożenia towarów, palet, a w nowoczesnych szpitalach do przewożenia szkodliwych odpadów medycznych lub sprzętu medycznego.



Zadanie do wykonania

Zbudowanie robota, który przewiezie jak największy ciężar.

Ciekawostka

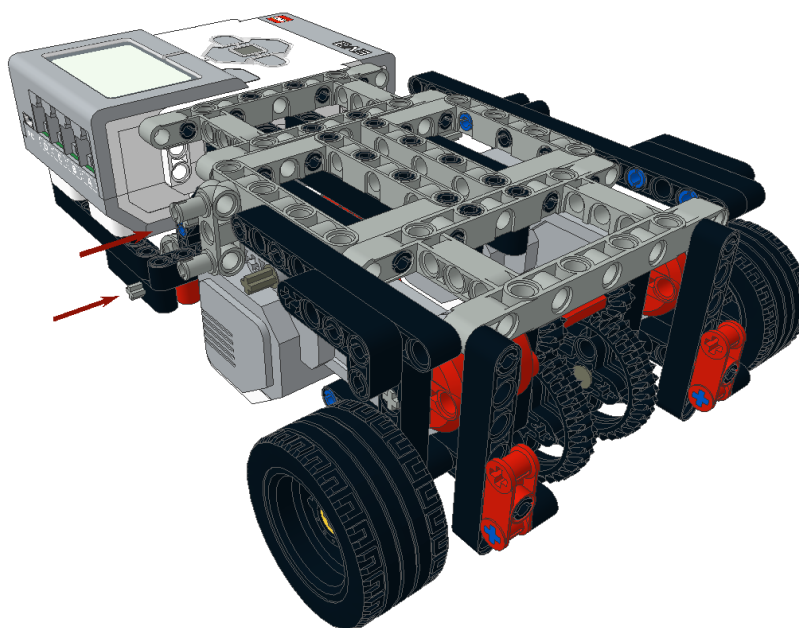


Największy udźwig wśród robotów przemysłowych ma robot FANUC M2000iA, a wynosi aż **1200 kg!**

Budowa robota



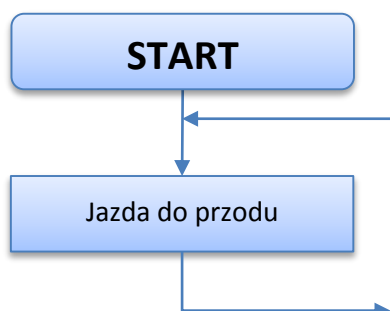
Konstrukcja Robosiłacza zostanie wyposażona w przekładnię, pozwalającą uzyskać większą moc silników. Ponadto robot musi być bardzo stabilny i mocny aby wytrzymać przewożone obciążenie. Rysunek poniżej przedstawia efekt końcowy. Instrukcja budowy znajduje się na płycie CD dołączonej do książki (*11. Siłacz.pdf*).



Programowanie krok po kroku

Przedstawiony poniżej algorytm opisuje przejazd z punktu A do punktu B po linii prostej.

Algorytm działania



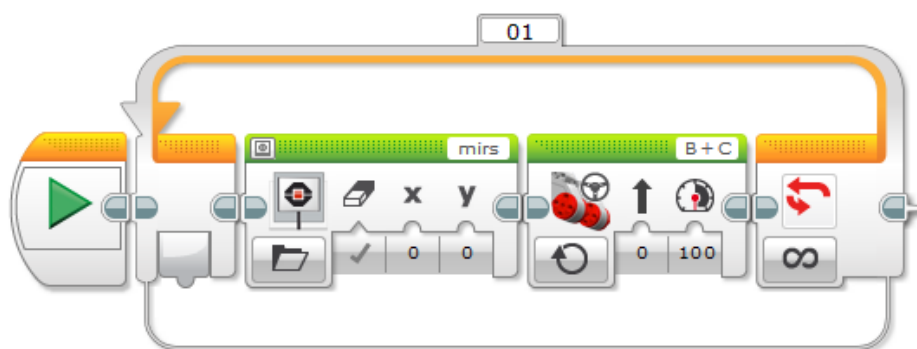
Kod programu dla robota Siłacza

Krok 1. Kod programu rozpoczyna **blok Startu**.

Krok 2. Program będzie wykonywany cyklicznie, więc wykorzystany zostanie blok pętli (**Loop**).

Krok 3. Uruchomienie silników na 100% mocy.

Krok 4. Włączenie wyświetlacza z wybranym obrazkiem.





Robozwierzaki

Dużym zainteresowaniem wśród dzieci cieszą się roboty, które swoim kształtem i zachowaniem przypominają prawdziwe zwierzęta. Jest to wynalazek dobry zarówno dla dzieci, jak i dla rodziców. Po robotach nie trzeba sprzątać, karmić ich, ani wychodzić z nimi regularnie na spacer. Robo zwierzaki znalazły swoich odbiorców również wśród starszego pokolenia. Robot o nazwie PARO wyposażony jest w sztuczną inteligencję oraz czujniki dotyku, które umożliwiają im odpowiedź na dźwięk i dotyk. Ukazuje dzięki

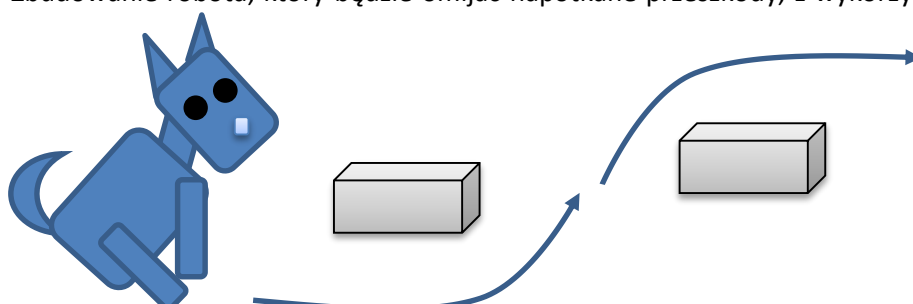


temu emocje, takie jak zaskoczenie, szczęście czy złość. Ma możliwość nauczenia się własnego imienia i reakcji na słowa, często używane przez właściciela. Produkt ten został przygotowany z myślą o ludziach starszych, z domów opieki społecznej, którzy potrzebują bliskości drugiej osoby.

Zwierzęta stanowią również popularny temat wśród robotyków. Analizują oni sposoby ich poruszania, próbując odwzorować zaobserwowane metody u robotów.

Zadanie do wykonania

Zbudowanie robota, który będzie omijać napotkane przeszkody, z wykorzystaniem czujnika podczerwieni.



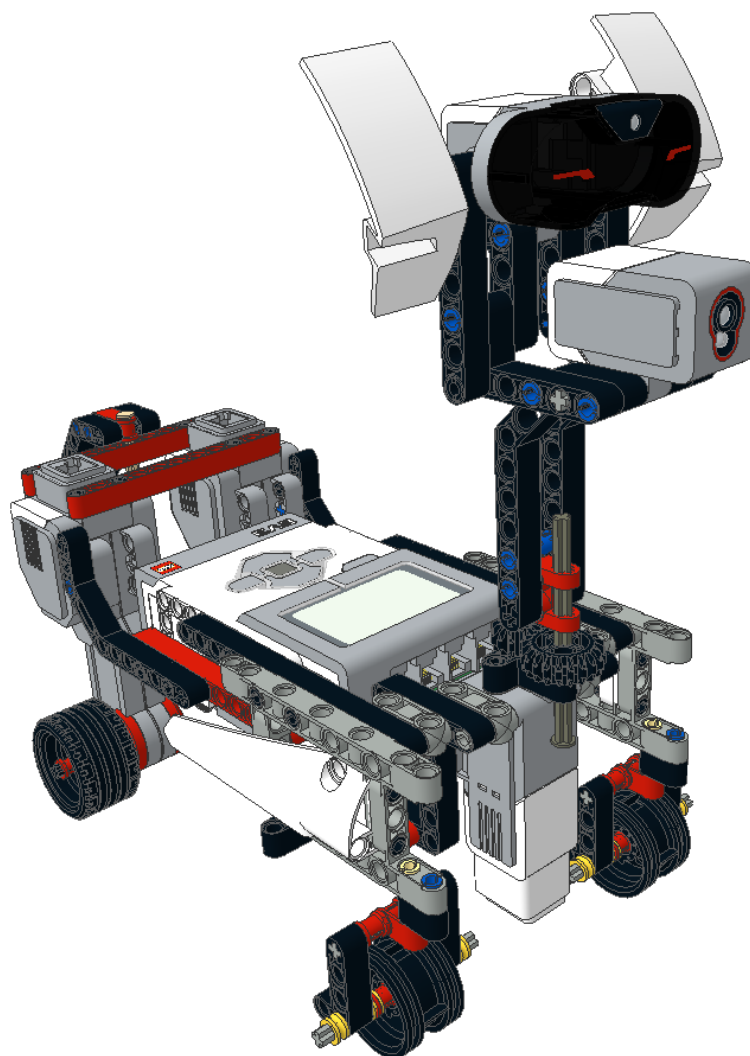


Budowa robota

Konstrukcja robota składa się z:

- Czujnika podczerwieni,
- 2 silników dużych – odpowiedzialnych za poruszanie się robota,
- Silnika średniego – umożliwiającego obrót głowy robota,
- Kostki EV3.

Rysunek poniżej przedstawia wygląd końcowy robota. Instrukcja dostępna jest na płycie CD dołączonej do książki (12. *Piesek.pdf*).

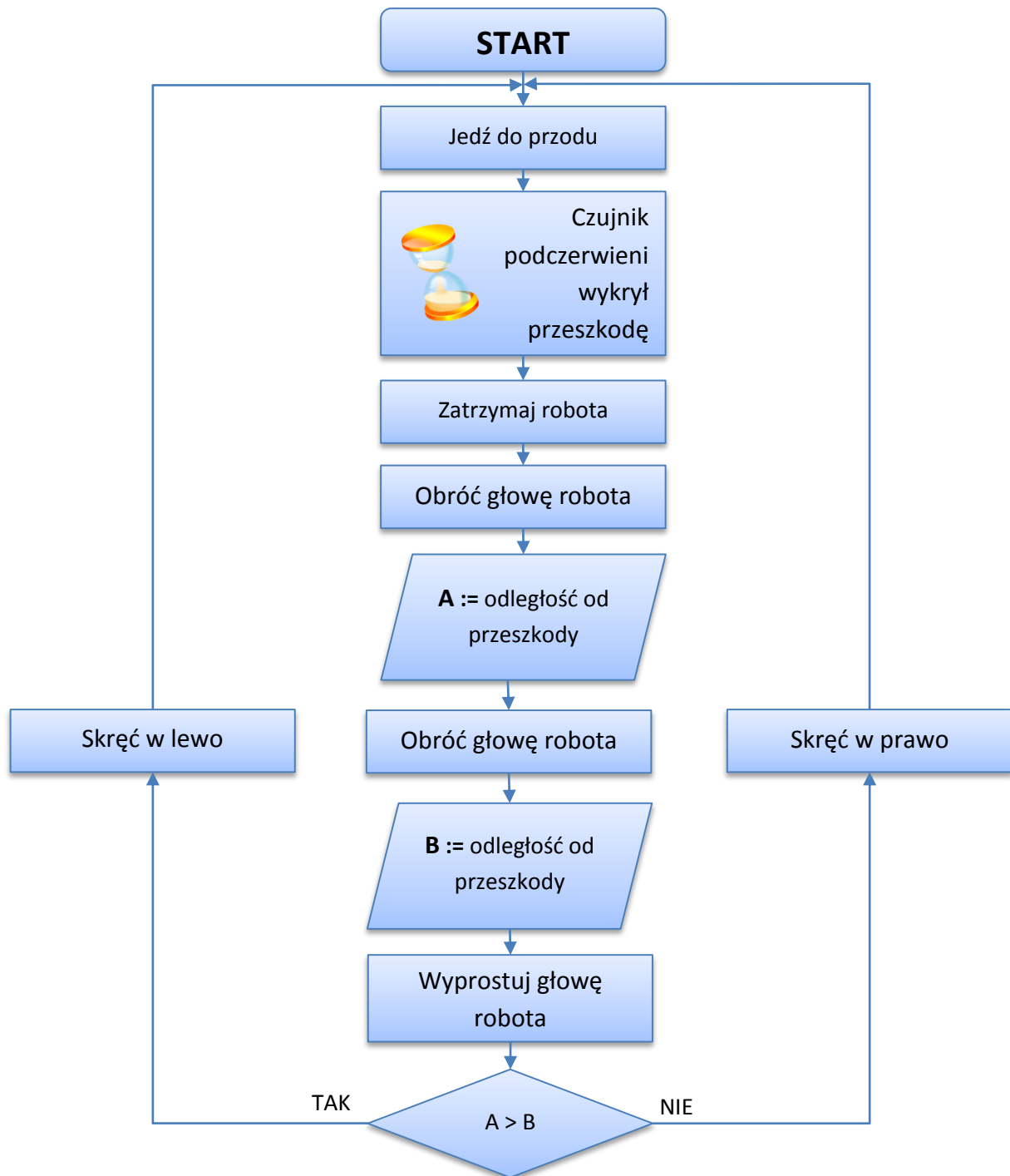


Programowanie krok po kroku

Realizacja zadania wymaga przeanalizowania zasady działania robota. Dwa duże silniki odpowiedzialne będą za poruszanie się robota. Czujnik podczerwieni został zamontowany na ruchomej głowie, za której ruch odpowiedzialny będzie silnik średni.

Algorytm działania

Zapis $A := \text{odległość}$ to przypisanie do zmiennej A wartości, zmierzonej odległości.



Kod programu dla robota Zwierzaka

Krok 1. Program rozpoczyna blok Startu.

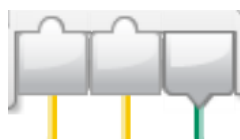
Krok 2. Instrukcje będą powtarzane wielokrotnie. Do tego celu zostanie zastosowana pętla programowa.

Krok 3. Zgodnie z algorytmem działania robot będzie jechał do przodu, dopóki czujnik podczerwieni nie odkryje przeszkody. W związku z tym zastosować należy **blok Move Steering(tryb ON)** oraz **Wait (Infrared Sensor – Compare – Proximity)**.

Krok 4. Gdy czujnik wykryje przeszkodę, silniki zatrzymują się (*Move Steering* – tryb *OFF*).

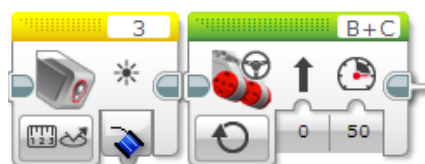
Krok 5. Silnik średni obraca głową z czujnikiem podczerwieni – *Medium Motor*, tryb *On for Degree*.

Krok 6. Czujnik mierzy odległość od przeszkody wykorzystując blok *Infrared Sensor*, tryb *Measure – Proximity*. W tym momencie warto wspomnieć o sygnałach wejścia i wyjścia.

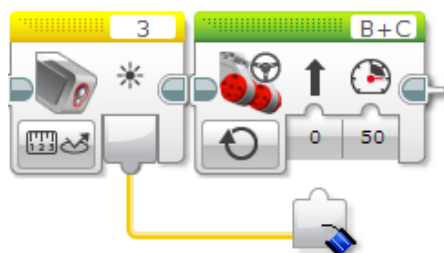


Każdy blok zawiera sygnały wejścia i wyjścia, wspomniano o tym w rozdziale „*Pierwsza konstrukcja i programowanie*”. Warto jednak dodać, że sygnały z jednego bloku mogą być przekazywane do kolejnego za pomocą połączeń. Typy danych, które można przekazywać dalej określone są za pomocą wypustek: zaokrąglonych (liczbowe), trójkątnych (logiczne) i kwadratowych (tekstowe). Na wejście/ wyjście bloku można przekazywać tylko i wyłącznie, taki typ danych, do którego został przystosowany (tj. do wejść numerycznych można przekazać tylko i wyłącznie sygnały numeryczne).

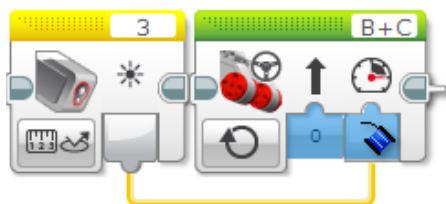
Aby przeprowadzić połączenie, należy:



- Zacząć od bloku zawierającego sygnał wyjściowy.



- Przeciągnąć sygnał wyjścia, trzymając blok lewym klawiszem myszy.



- Upuścić element sygnału wyjściowego na odpowiednim wejściu bloku.

TYP	LOGICZNY	NUMERYCZNY	TEKSTOWY	TABLICA NUM.	TABLICA LOG.
BLOK WEJŚCIA					
BLOK WYJŚCIA					

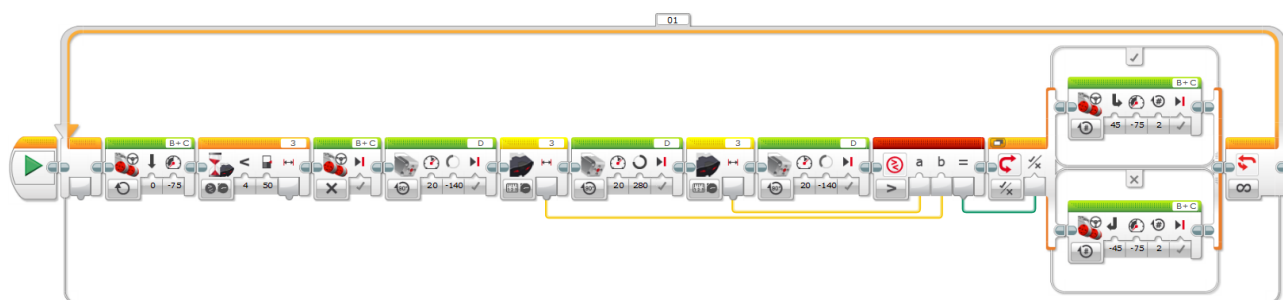
Krok 7. Przy pomocy bloku *Medium Motor* obrócić silnik średni w przeciwną stronę.

Krok 8. Ponownie dokonać pomiaru odległości od przeszkody (blok *Infrared Sensor*, tryb *Measure – Proximity*).

Krok 9. Platforma z czujnikiem powinna wrócić do pozycji początkowej (blok *Medium Motor*).

Krok 10. Następnie należy porównać otrzymane wyniki pomiaru odległości robota od przeszkody. Wykonuje się to za pomocą **bloku Porównania** (*Compare, Czy A > B ?*) znajdującego się w czerwonej zakładce (*Data Operations*). W opisany wcześniej sposób przeciągnąć odpowiednio sygnały wyjściowe otrzymane w wyniku pomiarów odległości z bloków czujnika podczerwieni do sygnałów wejściowych bloku Porównania.

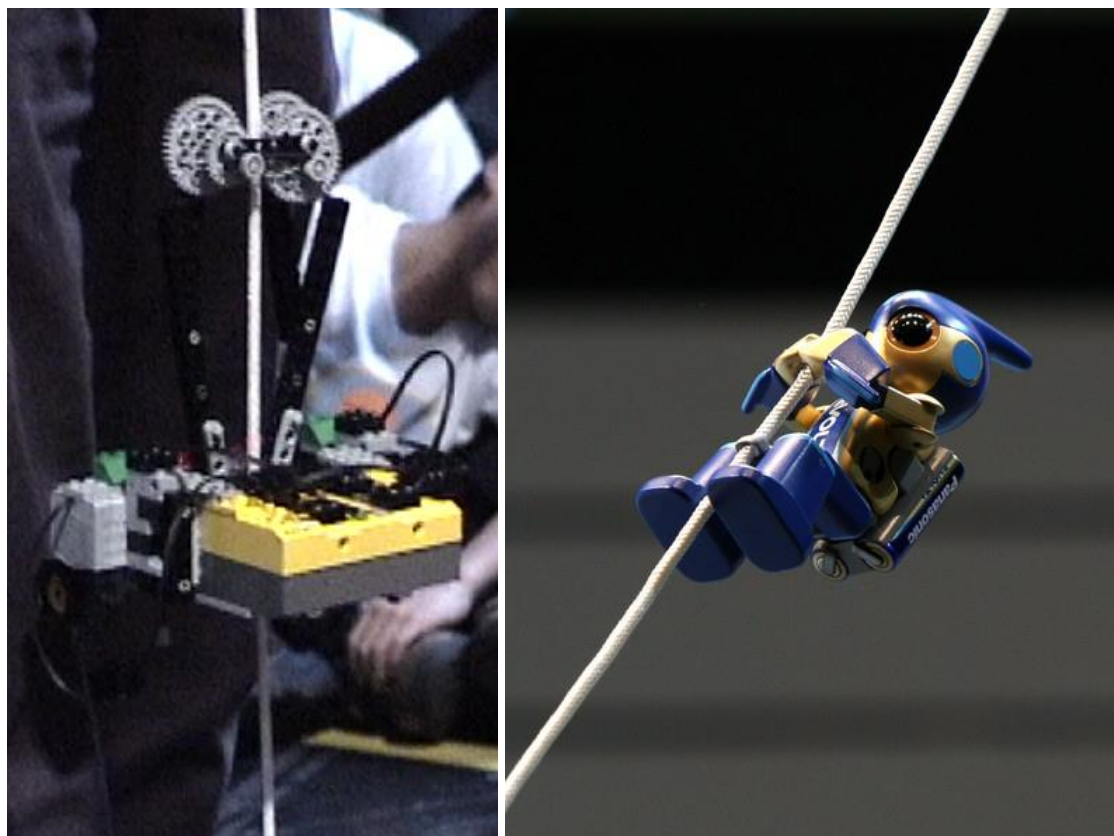
Krok 11. Rezultat porównania powinien zostać przesłany na **blok Instrukcji warunkowej** (*Switch, tryb Logiczny*). Jeśli wynik porównania jest poprawny (**True**) – wówczas robot powinien się obrócić w prawo. Jeśli wynik nie jest poprawny (**False**) – wówczas robot obraca się w lewo.



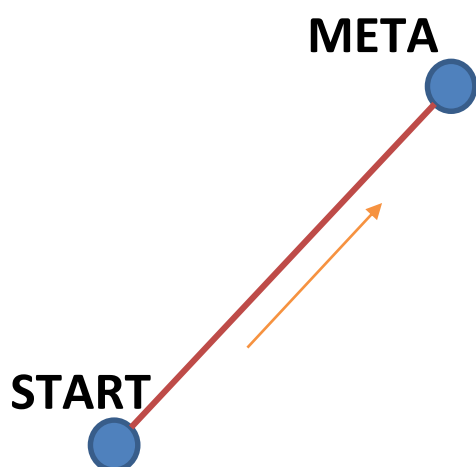


Jeżdżący po sznurku

Kolejnym robotem wspomagającym działania człowieka jest robot poruszający się na sznurku. Umożliwia on sprawne przenoszenie obiektów na wysokości pomiędzy dwoma budynkami.

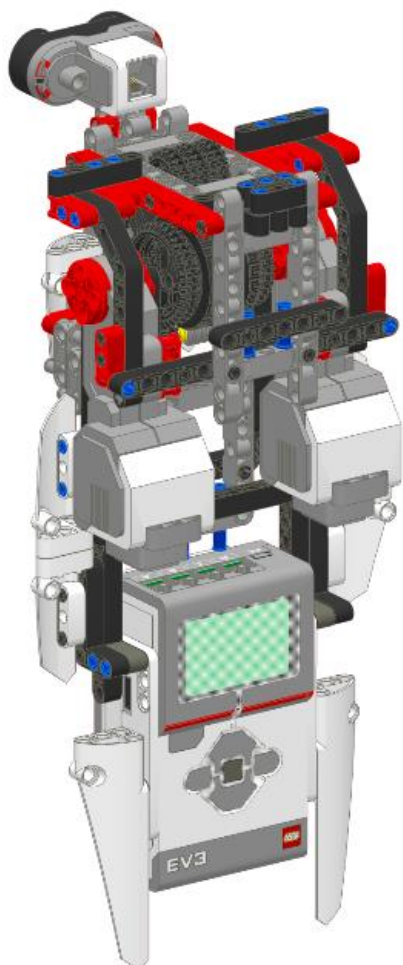


Zadanie do wykonania



Robot poruszający się po linii powinien być w stanie przejechać po sznurku zawieszonym na dwóch różnych poziomach. Kąt zawieszenia nie powinien mieć wpływu na jazdę robota.

Budowa robota



Ważnym aspektem realizacji robota jest dokładne rozpatrzenie przejazdu na linie, tj. w jaki sposób poruszać się po sznurku, by poślizg był jak najmniejszy. Robot z instrukcji wyposażony został w system kół zębatach, przez który zostaje ciasno przeciągnięty sznurek. Zwiększają one siłę tarcia i przeciwdziałają sile grawitacji. Przełożenie sznurka między kołem, a przekładnią powoduje, że ruch po linie jest realizowany bez problemów. Dodatkowo koło zaopatrzone jest w gumową oponę zmniejszającą poślizg robota i zapewniającą stabilny przejazd. Należy również pamiętać o prowadnicach, które zapewniają stabilne prowadzenie robota po sznurku.

Na rysunku obok przedstawiony został robot, który zostanie zbudowany w trakcie zajęć. Składa się on z :

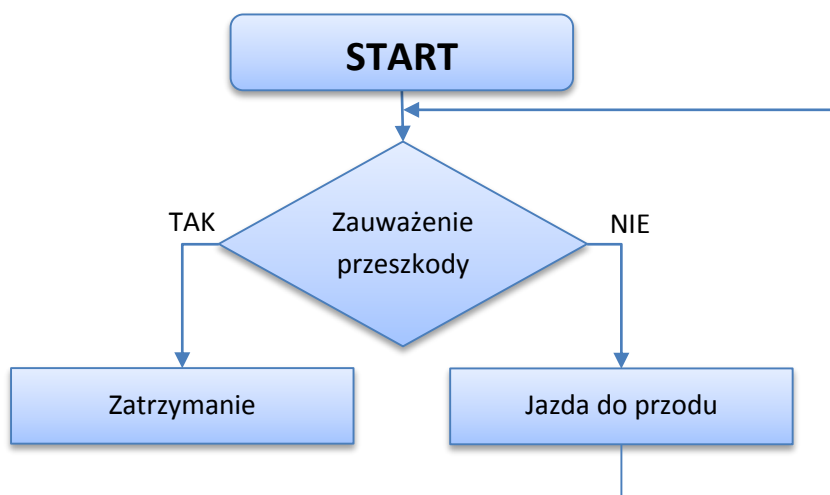
- Przekładni na moc,
- Dwóch silników,
- Czujnika odległości,
- Kostki EV3.

Instrukcja budowy znajduje się na płycie CD załączonej do książki (13. *Sznurek.pdf*).

Programowanie krok po kroku

Robot będzie zawieszony na linie pomiędzy dwoma punktami A i B. Jego zadanie będzie polegało na przejechaniu trasy bez względu na kąt zawarty pomiędzy odcinkiem AB a poziomem.

Algorytm działania



Krok 1. Robot wyrusza z punktu Startu.

Krok 2. Jedzie, dopóki czujnik odległości nie zauważy przeszkody.



Kod programu dla robota poruszającego się po linii

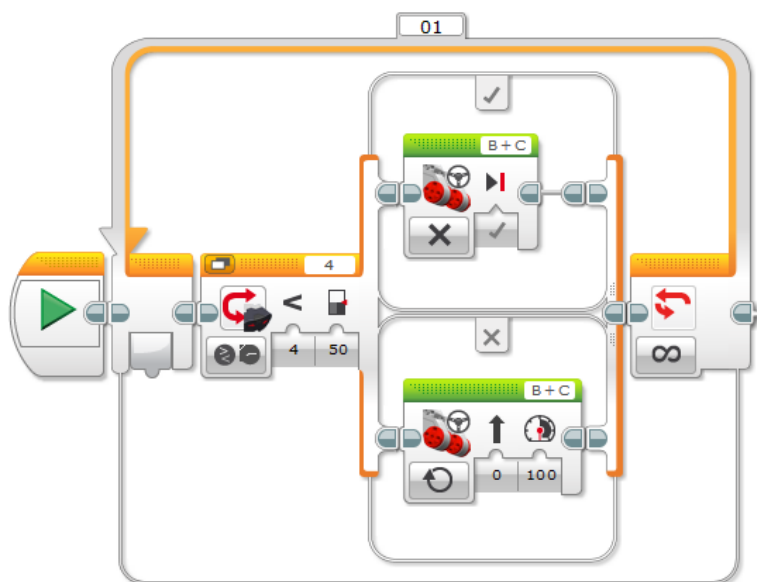
Krok 1. Program rozpoczyna **blok Startu**.

Krok 2. Kod programu będzie powtarzany wielokrotnie, w tym celu użyta zostanie **Pętla programowa**.

Krok 3. Czujnik odległości sprawdza, czy na jego drodze nie znajduje się żadna przeszkoda.

Krok 4. Jeśli czujnik dostrzeże przeszkodę, silniki są zatrzymywane.

Krok 5. W przeciwnym wypadku silniki są uruchamiane





Żuraw



Żuraw, potocznie zwany dźwigiem, służy do przemieszczania (podnoszenia) ładunków pomiędzy dwoma obiektami na różnych poziomach. Żurawie mogą być stacjonarne lub mobilne (na gąsienicach lub kołach).

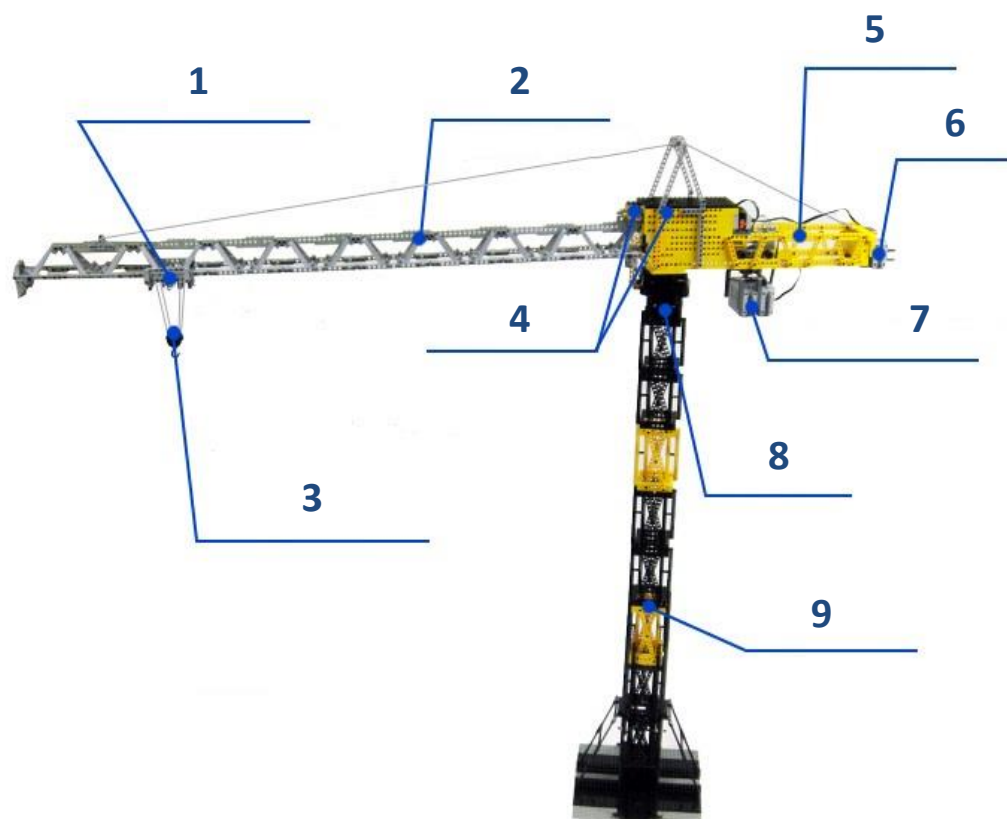
Znajdują zastosowanie głównie:

- Na budowach, gdzie muszą być przenoszone bardzo ciężkie płyty na bardzo duże wysokości,
- W portach, do przeładowywania towarów ze statków na ląd i odwrotnie.

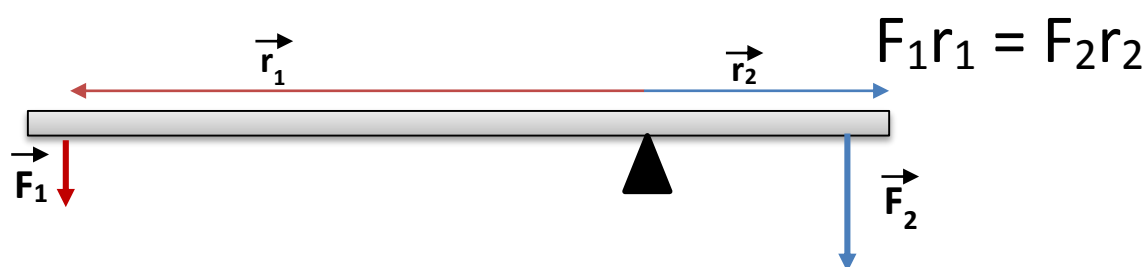
Zasada działania

Dźwig wieżowy składa się z:

1. Wózka,
2. Ramienia,
3. Bloku wciągającego,
4. Podnośnika,
5. Kontrramienia,
6. Przeciwwagi,
7. Ruchomej przeciwwagi,
8. Obrotnicy,
9. Wieży, masztu.



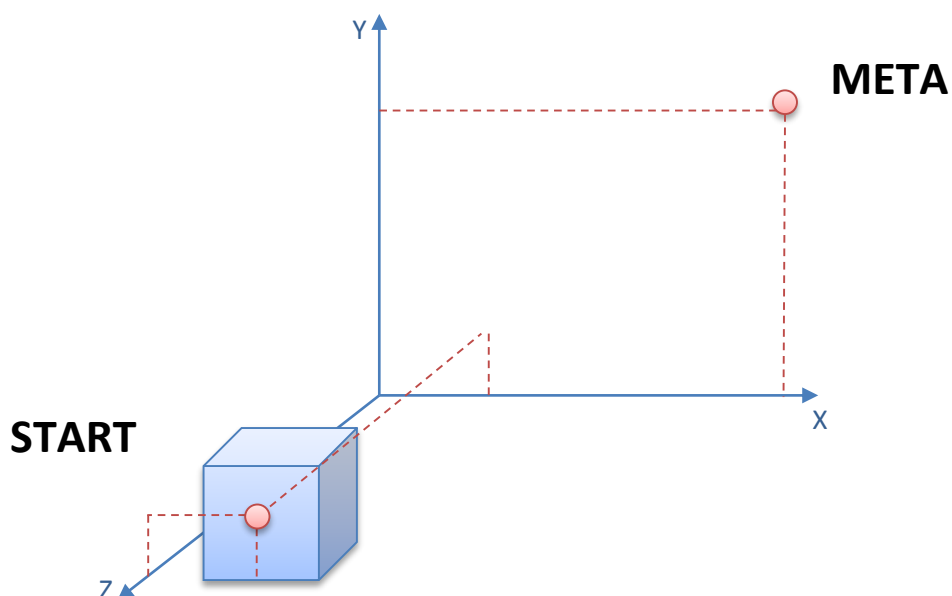
Najłatwiej zasadę działania dźwigu może opisać jedna z maszyn prostych, jaką jest dźwignia dwuramienna. Czym właściwie są maszyny proste? Są to maszyny umożliwiające, przy użyciu niewielkiej siły podnoszenie, przesuwanie ciężarów.



Zasada równowagi dźwigni dwustronnej mówi, że pozostaje ona w równowadze, gdy wypadkowy moment sił przyłożonej do niej wynosi zero. Ramiona dźwigni stanowią ramiona odpowiednich sił. Wynika z tego, że stosunek długości ramion dźwigni informuje o zysku siły (ile razy większą siłą F_2 można uzyskać działając siłą F_1). Dźwig budowlany działa na tej samej zasadzie.

Zadanie do wykonania

Zbudowanie robota, który będzie musiał przenieść pewien element z pozycji startowej do końcowej (*patrz rysunek poniżej*).



Budowa robota

Konstrukcja robota musi być łatwa w obsłudze dla operatora. Powinna zatem zawierać przyciski umożliwiające poruszanie się w przestrzeni XYZ oraz wykonywanie obrotów w tych przestrzeniach poszczególnymi elementami.

Robot musi posiadać:

- Przekładnie,
- Czujniki,
- Silniki,
- Chwytnak.

Na rysunku poniżej przedstawiono efekt końcowy budowy wg instrukcji umieszczonej na płycie CD podłączonej do książki (*14. Dźwig.pdf*).

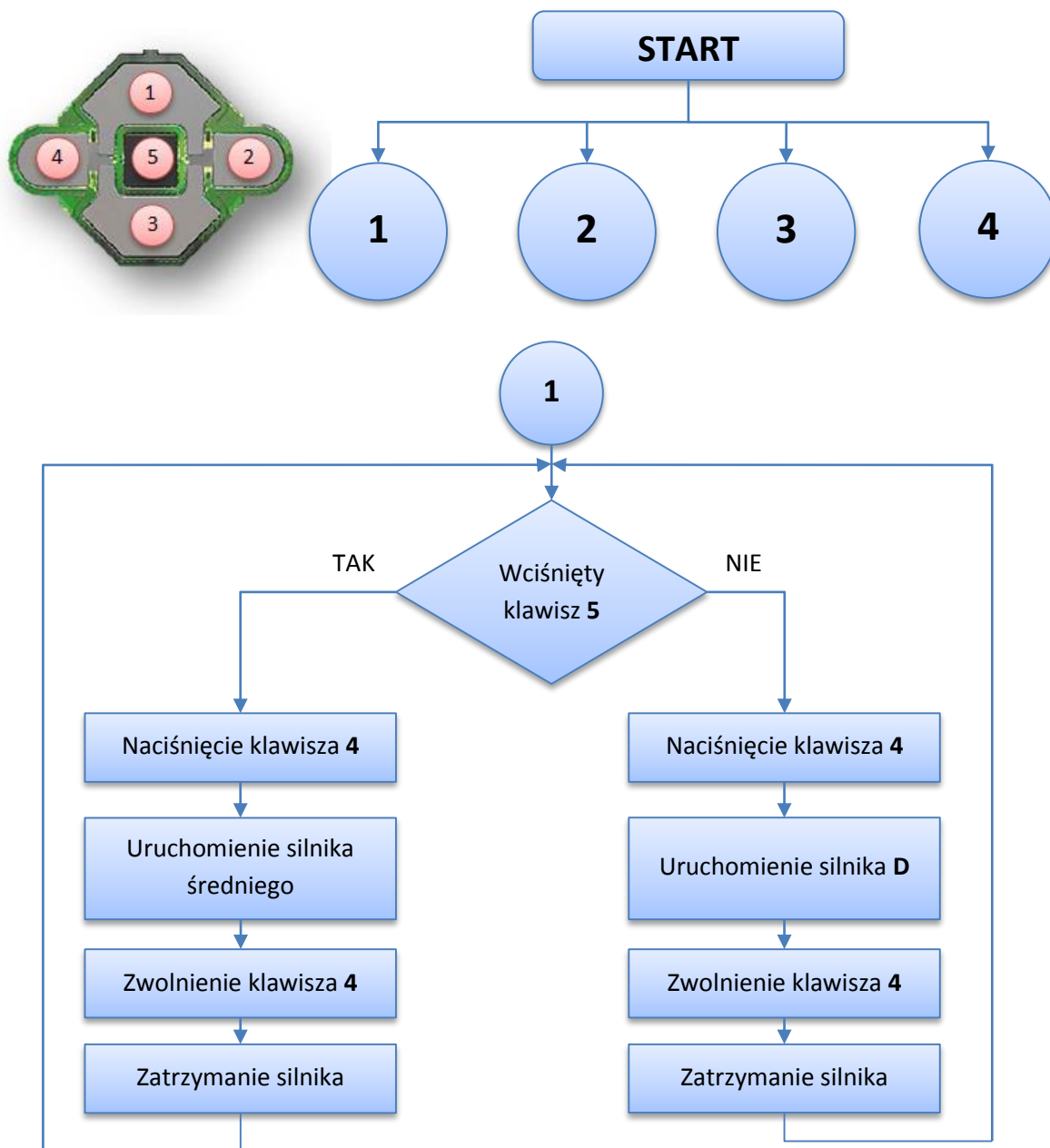


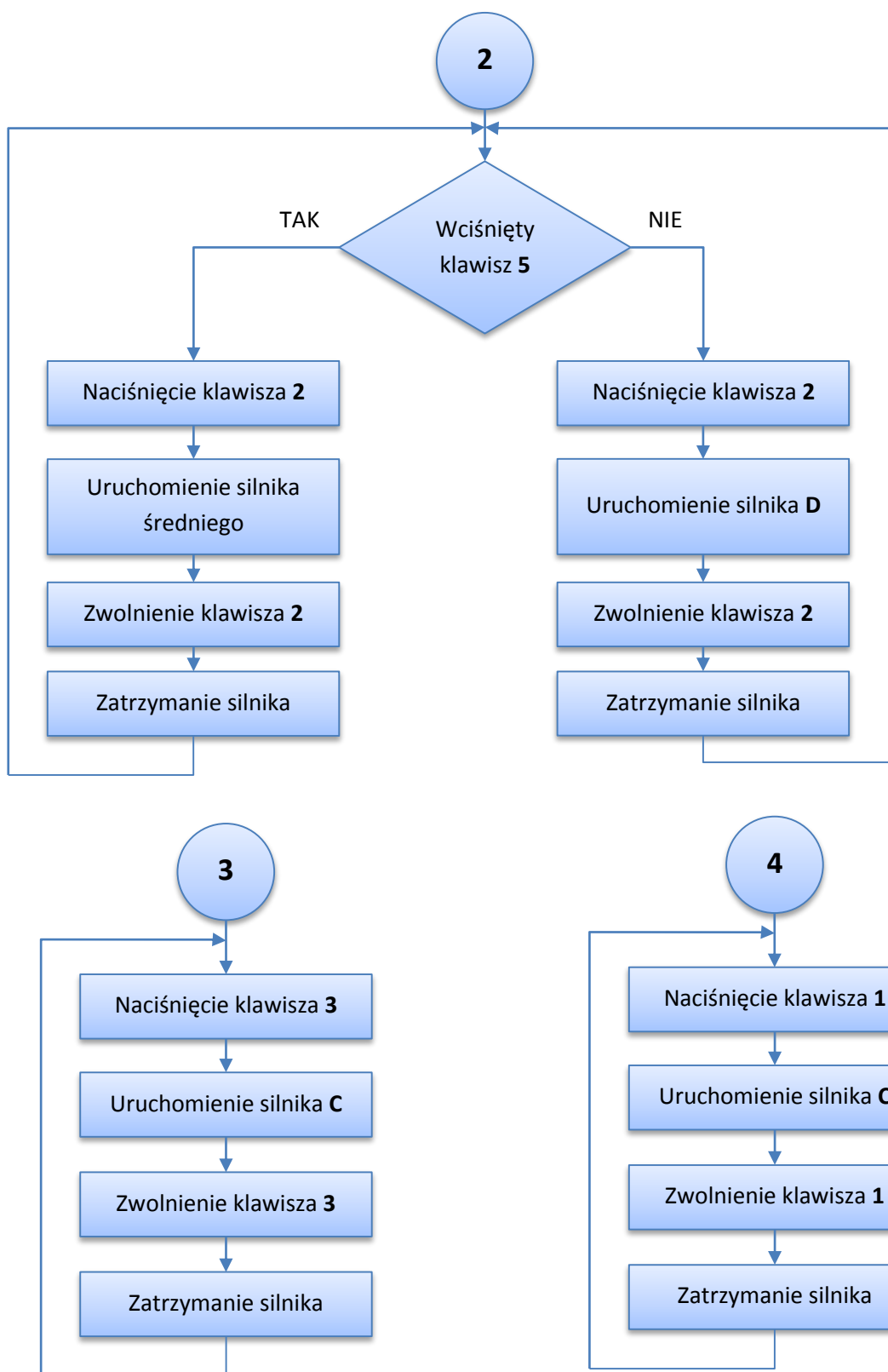
Programowanie krok po kroku

Znając podstawową zasadę działania dźwigu oraz jego schemat konstrukcyjny, należy zastanowić się nad algorytmem działania dźwigu.

Algorytm działania

Sterowanie będzie realizowane poprzez naciskanie klawiszy umieszczonych na kostce EV3.







Kod programu dla robota

Zgodnie ze schematem algorytmu, składa się on z czterech osobnych części. Każda z nich będzie realizowana niezależnie. W związku z tym kod programu zostanie zaprezentowany podobnie. Każdy fragment będzie osobną odnogą całego programu.

Krok 1. Program rozpoczyna **blok Startu**.

Część I.

Krok 2. Powtarzalność programu zapewni **pętla programowa (Loop)**.

Krok 3. Blok Instrukcji warunkowej sprawdzi, czy przycisk **5** został wciśnięty (*Switch – Brick Buttons – Compare; Id klawisza: 1, Stan: 1 - Wciśnięty*).

Krok 4. Jeśli wciśnięto klawisz **5**, sprawdzany jest stan klawisza **4** – jeśli zostanie on wciśnięty, **blok Medium Motor** uruchamia silnik średni. Po zwolnieniu klawisza, silnik zostaje natychmiast zatrzymany.

Krok 5. Nienaciśnięcie klawisza **5** powoduje sprawdzanie stanu klawisza **4** - jeśli zostanie on wciśnięty, **blok Large Motor** uruchamia silnik duży. Po zwolnieniu klawisza, silnik zostaje natychmiast zatrzymany.

Część II. – podobna do Części I, ale obracająca silniki w przeciwną stronę.

Krok 6. Powtarzalność programu zapewni **pętla programowa (Loop)**.

Krok 7. Blok Instrukcji warunkowej sprawdzi, czy przycisk **5** został wciśnięty (*Switch – Brick Buttons – Compare; Id klawisza: 1, Stan: 1 - Wciśnięty*).

Krok 8. Jeśli wciśnięto klawisz **5**, sprawdzany jest stan klawisza **2** – jeśli zostanie on wciśnięty, **blok Medium Motor** uruchamia silnik średni. Po zwolnieniu klawisza, silnik zostaje natychmiast zatrzymany.

Krok 9. Nienaciśnięcie klawisza **5** powoduje sprawdzanie stanu klawisza **2** - jeśli zostanie on wciśnięty, **blok Large Motor** uruchamia silnik duży. Po zwolnieniu klawisza, silnik zostaje natychmiast zatrzymany.

Część III.

Krok 10. Powtarzalność programu zapewni **pętla programowa (Loop)**.

Krok 11. Część ta oczekuje na naciśnięcie klawisza **3**. W tym celu wykorzystuje się **Blok Oczekiwania (Wait – Brick Buttons - Compare)**.

Krok 12. Jeśli wciśnięto klawisz **3**, **blok Large Motor** uruchamia silnik duży.

Krok 13. Po zwolnieniu klawisza, silnik zostaje natychmiast zatrzymany.

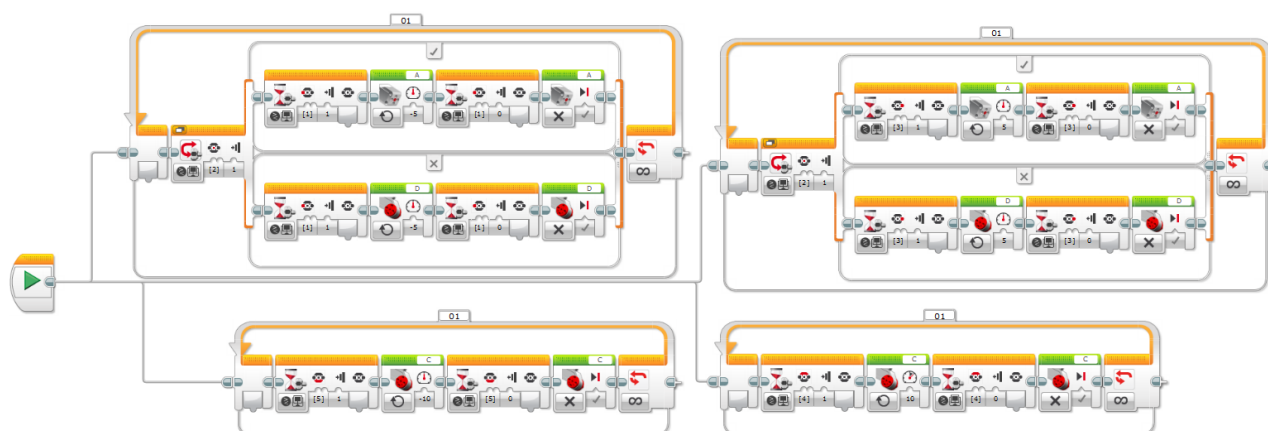
Część IV. – podobnie jak w Części III, tylko obraca silnik w przeciwną stronę.

Krok 14. Powtarzalność programu zapewni **pętla programowa (Loop)**.

Krok 15. Część ta oczekuje na naciśnięcie klawisza **1**. W tym celu wykorzystuje się **Blok Oczekiwania (Wait – Brick Buttons - Compare)**.

Krok 16. Jeśli wciśnięto klawisz **1**, **blok Large Motor** uruchamia silnik duży.

Krok 17. Po zwolnieniu klawisza, silnik zostaje natychmiast zatrzymany.





Skrećne koła

W rozdziale zatytułowanym „Linefollower” został przedstawiony robot przystosowany do poruszania się z możliwością skręcania. Był on wyposażony w dwa silniki odpowiedzialne za sterowanie każdym kołem z osobna. W tym rozdziale przedstawiony zostanie inny sposób poruszania się robota mobilnego. Jego realizacja będzie wymagać również dwóch silników, przy czym każdy z nich będzie odpowiedzialny za inne zadania. Pierwszy silnik będzie umożliwiał jazdę w przód i tył, a drugi wykorzystany zostanie do mechanizmu skręcającego.



Ciekawostka



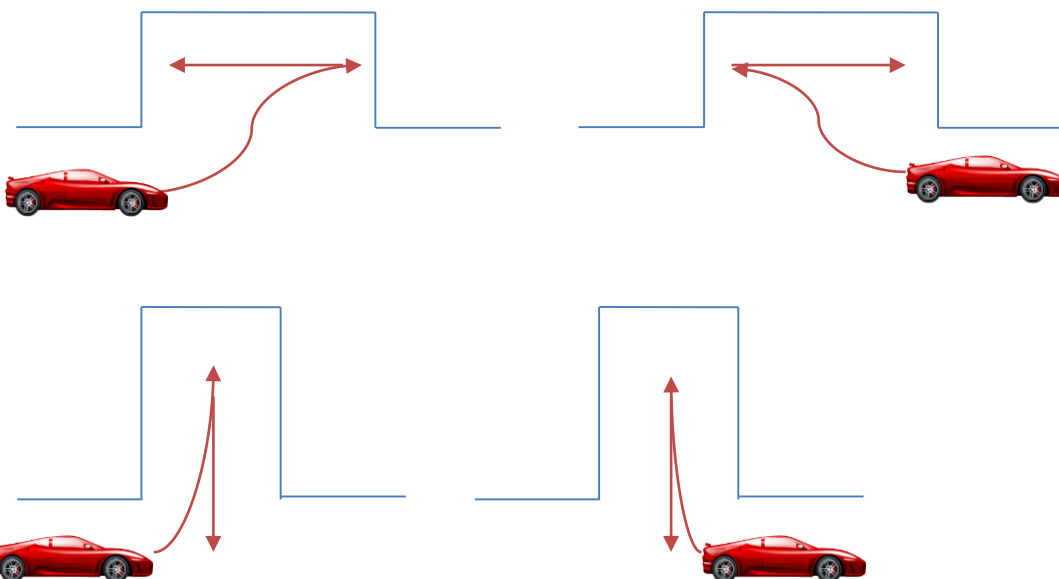
W samochodach mechanizm skręcania kół zarezerwowany był dla osi przedniej. Coraz popularniejszy jest jednak mechanizm czterech skrętnych kół (4WS). Koncepcja ta powstała na początku lat 20. XX wieku i była stosowana w ciężarówkach. Skręt kół tylnych w przeciwnym kierunku do skrętu kół przednich znacznie zmniejsza promień zawracania, jednak powoduje niestabilność przy dużych prędkościach. Dodatkowo zauważono, że skręt kół obu osi odbywał się w tym samym kierunku – zwiększa to

bezpieczeństwo, ale cechuje się małą zwrotnością. W celu uzyskania jak największej liczby korzyści połączono oba pomysły – przy niskich prędkościach tylne koła skręcały się przeciwnie do ruch kół przednich, a przy większych prędkościach skręty odbywały się w tę samą stronę.

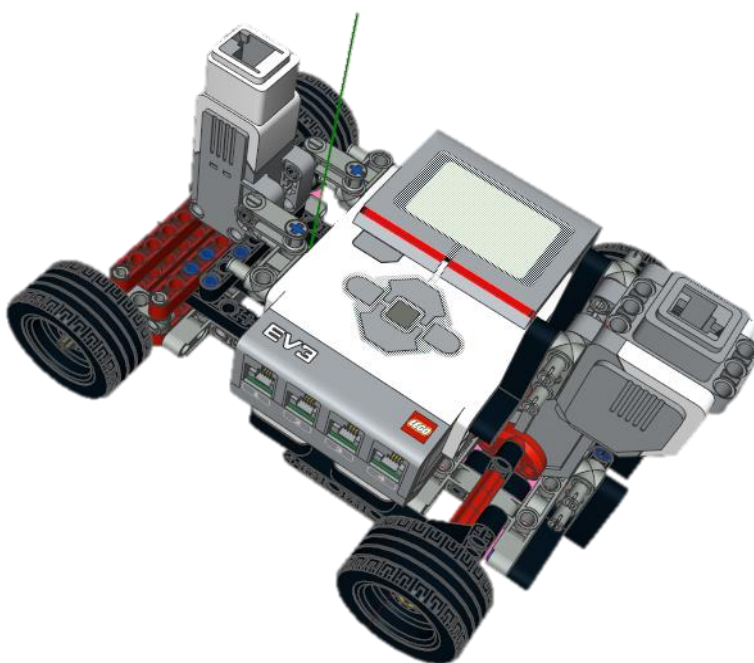
Zadanie do wykonania

Zbudowanie robota, który będzie mógł wykonać jeden z procesów parkowania:

- Parkowanie równoległe przodem,
- Parkowanie równoległe tyłem,
- Parkowanie prostopadłe przodem,
- Parkowanie prostopadłe tyłem.



Budowa robota



Konstrukcja robota o skrętnych kołach wymaga zastosowania dwóch silników. Przedni silnik odpowiada za skręt przednich kół, a tylni silnik napędza robota do jazdy. Rysunek obok przedstawia efekt końcowy budowy robota. Składa się on z:

- Silnika dużego,
- Silnika średniego,
- 4 kół,
- Kostki EV3.

Instrukcja budowy znajduje się na płycie CD dołączonej do książki (*15. Skrętne koła.pdf*).

Programowanie krok po kroku

Robot stawiany jest przy linii startu. Ma za zadanie wykonać jedno z wyżej wspomnianych zadań parkowania.

Algorytm działania



Realizacja zadań parkowania jest podobna. Różni się jedynie kątami skrętów kół robota.

Kod programu

Krok 1. Program rozpoczyna **blok Startu**.

Krok 2. Skręt kół realizowany za pomocą **bloku Medium Motor** (*On for Degrees*), realizujący obrót silnika o zadany kąt.

Krok 3. Jazdę do przodu realizuje **blok Large Motor**, który wykonuje określoną liczbę obrotów silnika (*On for Rotation*).

W zależności od realizowanego zadania należy odpowiednio obracać silnikami. Powyższe kroki należy odpowiednio powtarzać, aż do uzyskania określonego celu.



