

Krystian Machowczyk

Jak rozpocząć pracę z robotem **Lego Mindstorms EV3**



Wrocław 2023

Spis treści

1. Wstęp
2. O modelu STREAM
3. Od czego zacząć ?
 - o zestawie Lego Mindstorms EV3
 - budowa konstrukcji robota bazowego
4. Oprogramowanie sterujące
 - oprogramowanie EV3 Classroom
 - oprogramowanie EV3 Lab
5. Przykładowe Instrukcje programowania robota w EV3 Lab
 - zadanie 1. Uruchomienie robota. Jazda prosto
 - zadanie 2. Jazda robota po obwodzie kwadratu
 - zadanie 3. Reakcja czujnika koloru w czasie pracy (CK)
 - zadanie 4. Ruch robota wzdłuż wyznaczonej linii (CK)
 - zadanie 5. Reakcja robota na przeszkodę na torze (CO)
 - zadanie 6. Reakcja robota na zaporę na torze (CD)
 - zadanie 7. Operatory matematyczne podstawowe
 - zadanie 8. Operatory matematyczne zaawansowane
 - zadanie 9. Czytnik kolorów - barwy podstawowe (CK)
 - zadanie 10. Wykrywacz kolorów na obszarze pracy (CK)
 - zadanie 11. Kodowanie liczb za pomocą pilota (CP)
 - zadanie 12. Jazda i powrót robota sterowane pilotem (CP)
 - zadanie 13. Reakcja na przeszkodę. Zmiana toru jazdy (CP)
 - zadanie 14. Rozruch robota pilotem w trybie Beacon (CP)
 - zadanie 15. Jazda robota wzdłuż krawędzi ściany (CP)
 - zadanie 16. Jazda robota wzdłuż krawędzi - f. Equation (CP)

* (CK) - czujnik koloru, (CO) - czujnik odległości, (CD) - czujnik dotyku, (CP) - czujnik podczerwieni
6. Literatura proponowana do zajęć z robotyki

Wstęp

Wiele szkół i placówek oświatowych zostało wyposażonych w procesie wdrażania różnych programów edukacyjnych, w zestawy do nauki robotyki. Często były to zestawy proponowane przez różne firmy lub sprzedawców, zawarte w gotowych ofertach dla szkół. Jednak po realizacji zakupu zestawów przez placówkę zdarzało się, że nauczyciele uznawali to zadanie za zbyt skomplikowane do realizacji w szkole. Okazywało się bowiem, że zestawy do pracy przychodziły w drobnych częściach, z których najpierw należało zbudować robota bazowego, następnie rozpoznać funkcję pracy poszczególnych podzespołów, dobrać odpowiednie oprogramowanie sterujące na komputer i dopiero wtedy rozpocząć naukę programowania maszyny. Tu z kolei nauczyciele stawali przed problemem związanym z budową algorytmów oraz z obawą, że uczniowie w oczekiwaniu na pomoc od nauczyciela będą pytali o zagadnienia zbyt dla niego kłopotliwe. Wynika to z tradycyjnych metod nauczania, gdzie nauczyciel z założenia zna wszystkie treści dydaktyczne, których uczy a nieznanie pewnych zagadnień w obrębie nauczanego przedmiotu naraża go na utratę autorytetu. Sytuacje tego typu zdarzają się w szczególności w nauczaniu treści i umiejętności na przedmiotach technicznych, gdzie zwykle spora część uczniów wykazuje duże zainteresowanie tematyką i mają oni do tego predyspozycje, albo też w dobie cyfryzacji są po prostu bardziej biegli w obsłudze nowoczesnych urządzeń komputerowych. Okazuje się jednak, że istnieją metody pracy dydaktycznej, dzięki którym nauczyciel wraz z grupą uczniów z powodzeniem może rozpocząć zajęcia z robotyki a praca stopniowo będzie przynosiła efekty.

O modelu STREAM

Planując pracę dydaktyczną z wykorzystaniem mechatronicznych urządzeń technicznych i oprogramowania sterującego musimy zdawać sobie sprawę z tego, że tradycyjne metody nauczania mogą być dość trudne. Częściej sprawdzają się one w takich sytuacjach, kiedy nauczyciel posiada dużą wiedzę i umiejętności techniczne. To raczej zdarza się tylko w przypadku nauczycieli pracujących jednocześnie poza szkołą w branżach, np. programistycznej lub mechatronicznej. W przypadku nauczycieli, dla których robotyka i mechatronika są zagadnieniami równie nowymi, co dla uczniów, z pewnością na początku warto jest to zmienić. Metodyka pracy nauczyciela nie będzie oparta na nauczaniu podawczym, lecz na metodach opartych o naukę poprzez samodzielne poszukiwanie przez uczniów źródeł wiedzy i uczenie się poprzez doświadczenie. Dlatego planując pracę dydaktyczną związaną z nauczaniem robotyki, warto jest budować ją na założeniach modelu STREAM. Nazwa ta łączy w sobie skrót z ang. Science, Technologies, Robotics, Engineering, Arts, Mathematics. Edukacja STREAM opiera się na takich filarach jak: nauka, technologia, robotyka, sztuka oraz matematyka. To nowoczesna metodyka pracy, nastawiona na samodzielne poszukiwanie, odkrywanie i rozwiązywanie problemu a także na samodzielne stawianie wniosków przez ucznia lub zespół zadaniowy.

Więcej o edukacji STREAM (link):

<https://mow.walbrzych.dolnyslask.pl/wp-content/uploads/2022/11/STREAM.pdf>

Wiele szkół i placówek oświatowych wprowadziło już takie metody pracy dydaktycznej, ponieważ doskonale sprawdzają się one właśnie w sferze technicznej z wykorzystaniem nowoczesnych technologii, gdzie uczniowie samodzielnie lub z niewielką pomocą nauczyciela rozwiązują stawiane przed nimi zadania.

Przykłady i efekty pracy (linki):

<http://mow.walbrzych.dolnyslask.pl/metodyka-stream-na-zajeciach-kola-informatycznego/>

<http://mow.walbrzych.dolnyslask.pl/dzien-nowych-technologii-w-edukacji-2023/>

Stosując w pracy założenia metodyki STREAM nauczyciel wykorzystuje wiedzę techniczną tylko w zakresie swoich możliwości. Często są one na różnym poziomie, więc podczas pracy na zajęciach głównie nadzoruje pracę ucznia lub zespołu dbając o zapewnienie wszelkich środków i pomocy dydaktycznych do samodzielnej pracy, sam jedynie wskazuje tylko na bezpieczeństwo pracy prawidłową obsługę sprzętu. Oczywiście nauczyciel obserwując pracę uczniów sam również nabywa nowych umiejętności, które stopniowo rosną i będą wykorzystywane w pracy dydaktycznej w kolejnych latach.

Od czego zacząć?

Jesteśmy już spokojni o metody pracy. Uczniowie wiedzą od początku, że ich głównym zadaniem podczas zajęć z robotyki będzie samodzielna, twórcza praca poprzez doświadczenie. To oni będą wykazywać się inwencją twórczą i chwalić się sukcesem. Jednak do nauczyciela należy przygotowanie warsztatu pracy i wszelkich pomocy dydaktycznych do pracy zespołu. Jeśli to uczniom zapewnimy, praca powinna od początku stać się dla nas i dla uczniów nową, ciekawą przygodą. Co zatem zrobić?

Jednym z zestawów robotycznych, z którym pracę należy rozpocząć od podstaw, jest zestaw **Mindstorms EV3** firmy **LEGO**. Jest sporo szkół, które takie zestawy posiadają w swoim wyposażeniu. Z perspektywy czasu okazało się, że jest to jeden z lepszych modeli tej firmy, dający duże możliwości pracy zarówno z robotem, jak i z oprogramowaniem.

Pierwszym krokiem nauczyciela będzie zapoznanie uczniów z zawartością zestawu i podzielenie części do specjalnych organizatorów według zaleceń producenta. Instrukcja dołączona jest do zestawu. Można to zrobić na pierwszych zajęciach z robotyki, gdzie uczniowie sami dokonują klasyfikacji części na:

- elementy konstrukcyjne (listwy otworowe i krzyżowe),
- elementy łączeniowe (kołki, łączniki),
- elementy napędowe (silniki),
- elementy przekładni (koła zębate, cierne, zębatki ślimakowe),
- mikrokontroler,
- źródło zasilania (bateria litowo jonowa),
- czujniki odległości, dotyku i koloru,
- przewody zasilające.

Do rozpoznania przez uczniów poszczególnych części robota bardzo pomocna będzie wydana w języku polskim Instrukcja użytkownika, która w wersji PDF znajduje się pod linkiem:

<https://www.robotworld.pl/downloads/instrukcja-obslugi-lego-mindstorms-ev3.pdf>

Już podczas pierwszych zajęć uczniowie poznają funkcje poszczególnych części zestawu. Dowiadują się, że budowa robota będzie polegała na złożeniu poszczególnych części w całość. Części te są już poukładane i poklasyfikowane w naszym organizerze.

Należy tu podkreślić, że niektóre zestawy Mindstorms EV3 nie zostały wyposażone w ładowarki baterii mikrokontrolerów i trzeba ten element dokupić. Jednak w początkowej fazie zajęć z robotyki, na czas zakupu ładowarki możemy zamiennie wykorzystać 6 baterii AA 1,5V co z powodzeniem zapewnia pracę robota zasilając mikrokontroler, silniki i czujniki.

Podczas drugich zajęć uczniowie mogą już przystąpić do budowy konstrukcji **robota bazowego**. W tym celu nauczyciel przygotowuje wydruk instrukcji „**Education Vehicle**”, która krok po kroku wskazuje, jak uczniowie konstruują robota. Instrukcja znajduje się pod linkiem:

https://www.neorobots.pl/ftp/Lego/Instrukcje_Lego_EV3.html

Warto dodać, że wielu nauczycieli zajmujących się robotyką z wykorzystaniem robota **Lego Mindstorms EV3** posiada w swoim warsztacie pracy instrukcje do budowy konstrukcji na bazie tego zestawu, pochodzące właśnie z podanego wyżej źródła. Daje to uczniom możliwość pracy na zajęciach z robotyki na długi okres czasu.

Oprogramowanie sterujące

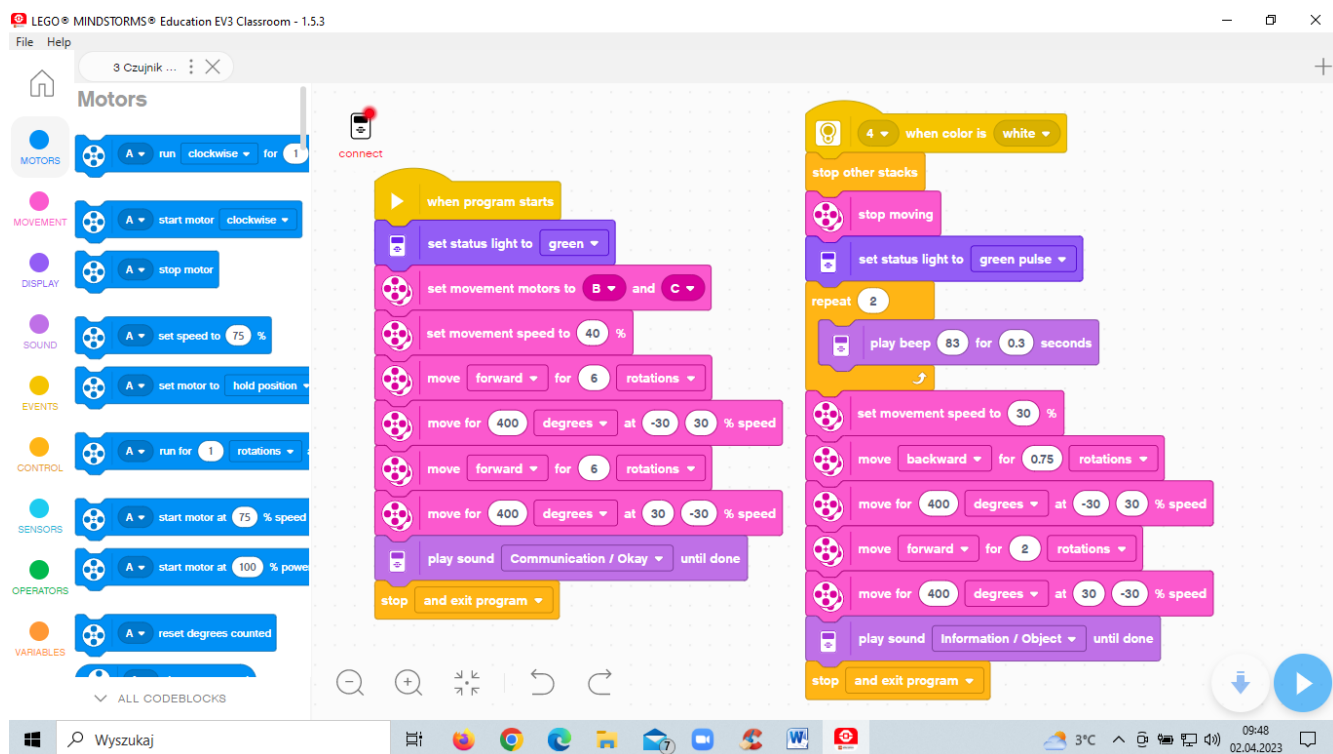
Gdy konstrukcja robota bazowego jest już gotowa, kolejne zajęcia z robotyki możemy prowadzić z wykorzystaniem oprogramowania komputerowego, współpracującego z tą maszyną. W tym celu możemy wykorzystać kilka aplikacji, które doskonale nadają się do pracy. Do rozpoczęcia zajęć, szczególnie z dziećmi z młodszych klas, doskonale nadaje się aplikacja **EV3 Classroom App**. Jest to ciekawe oprogramowanie, w którym skrypty programowe buduje się w oparciu o język programowania **Scratch**, więc każdy uczeń, który miał wcześniej zajęcia z informatyki z wykorzystaniem tego języka, doskonale da sobie radę z budową programów w tej aplikacji. Oprogramowanie działa na systemie operacyjnym Windows 8, 10 i 11 oraz na Mac i Android. Ponadto jest oprogramowaniem, którego wymogi techniczne spełnia większość szkolnych komputerów, co jest obecnie sporym udogodnieniem dla nauczycieli. Jednak warunkiem wygodnej pracy jest funkcja Bluetooth w komputerze, zaleca się więc do pracy laptop.

Instalator oprogramowania pobieramy ze strony education.lego.com, pod linkiem:

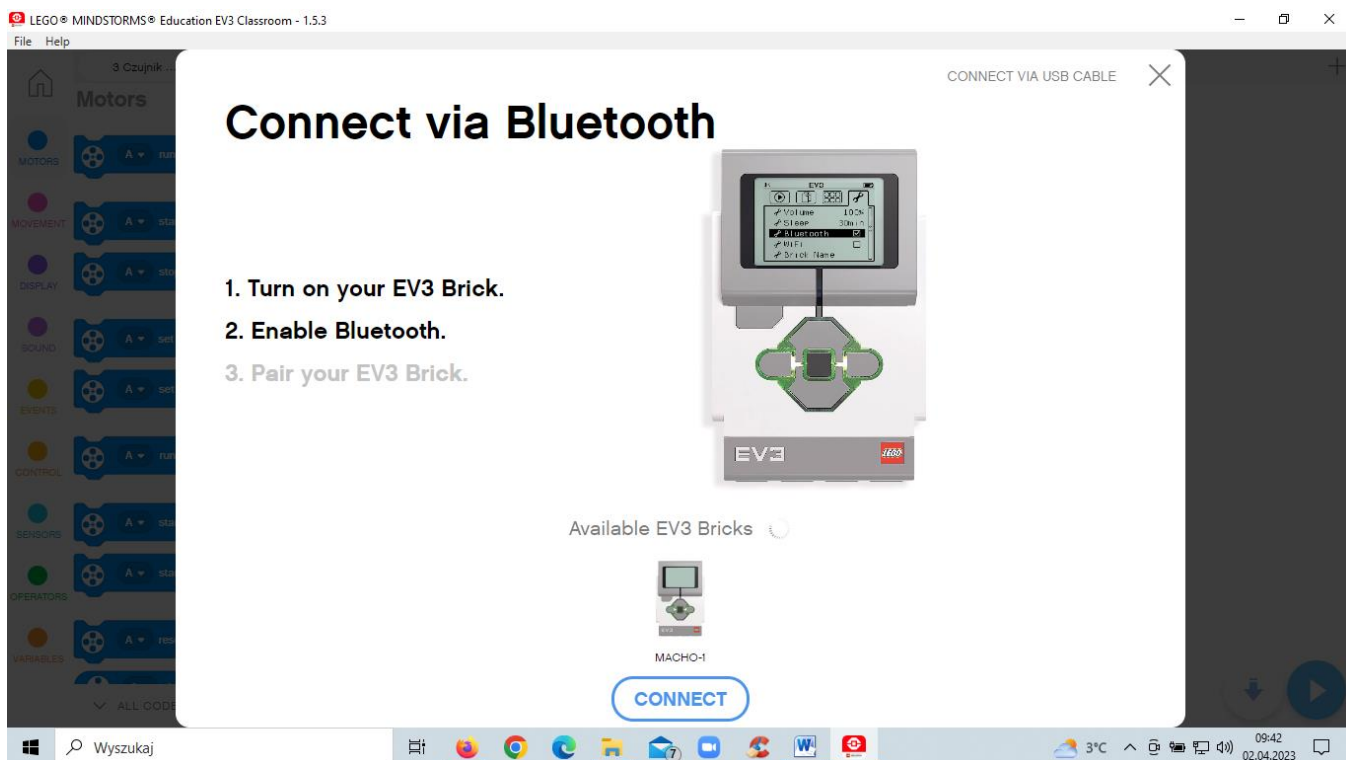
<https://education.lego.com/pl-pl/downloads/mindstorms-ev3/software>

Oprogramowanie pobiera się w pełni jako pakiet instalatora więc plik instalacyjny nadaje się do wielokrotnego użycia na komputerach i przechowywania na nośniku zewnętrznym do ponownego użycia w razie reinstalacji.

Wygląd panelu roboczego programu Classroom App v. 1.5.3



Wygląd panelu połączenia Bluetooth z mikrokontrolerem w Classroom App v. 1.5.3



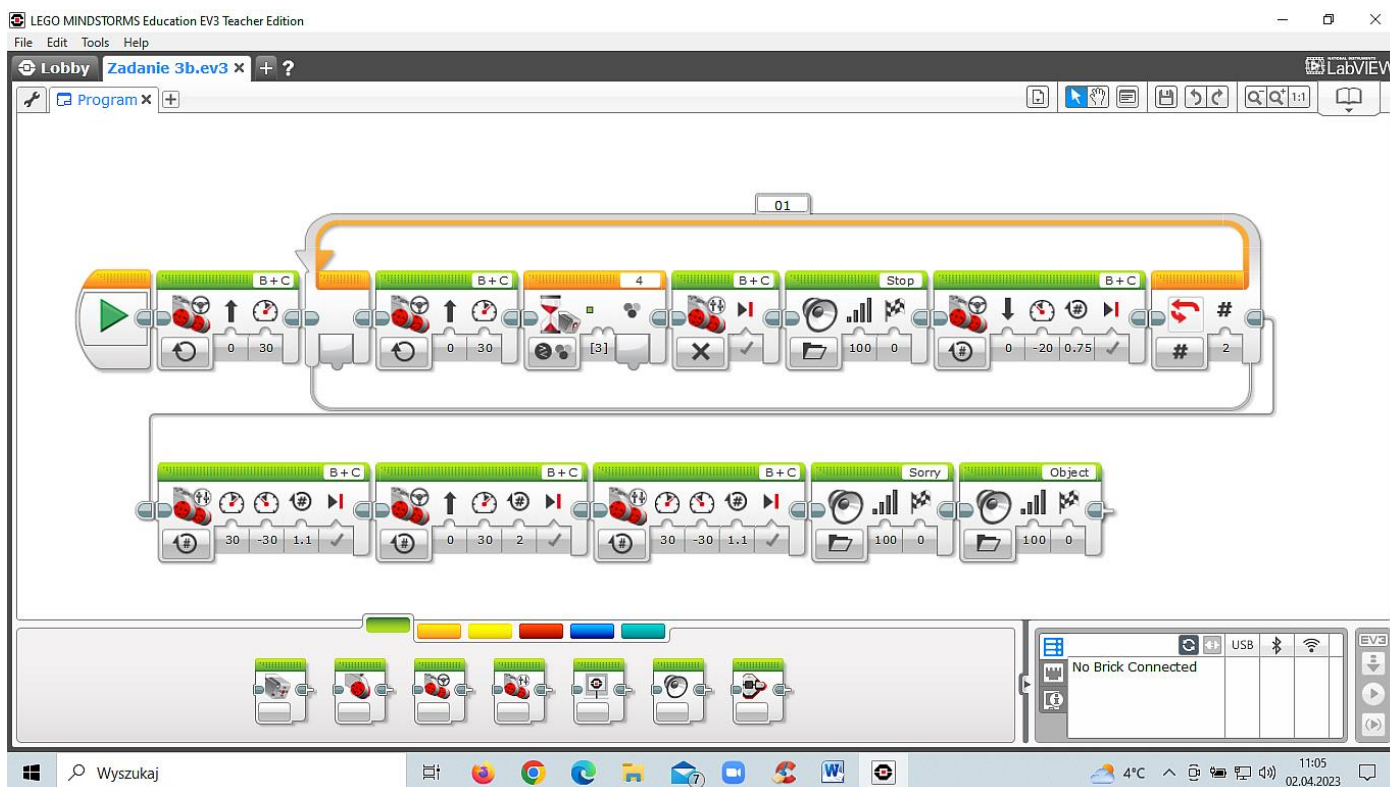
Przed połączeniem komputera z mikrokontrolerem warto jest nadać własną nazwę robotu w opcji mikrokontrolera **Settings** (ostatnia zakładka) – **Brick Name**. Podczas pierwszego połączenia należy w pole łączeniowe wpisać numer widniejący na panelu Mikrokontrolera. Zwykle jest to numer nadany fabrycznie: 1 2 3 4.

Oprogramowaniem, które sprawdza się w pracy z uczniami klas starszych, jest **EV3 Lab Software**. Jest to bardzo ciekawa aplikacja z dużymi możliwościami programowania. O ile oprogramowanie **Classroom** opiera się na powszechnie znanym uczniom **Scratch**, o tyle **EV3 Lab Software** oferuje całkiem inny, odmienny interfejs programowania blokowego. Jest on bardziej rozbudowany pod względem graficznym i z doświadczeń nauczycieli, bardziej atrakcyjny dla uczniów szkół ponadpodstawowych. W interfejsie programu można dostrzec większe możliwości rozwoju predyspozycji technicznych uczniów, przy jednoczesnym wprowadzeniu do nauczania czegoś całkiem nowego, raczej nie spotykanego na wcześniejszym etapie nauczania w szkole podstawowej. Oprogramowanie to nauczyciel może pobrać bezpośrednio ze strony producenta, pod linkiem:

<https://education.lego.com/pl-pl/downloads/retiredproducts/mindstorms-ev3-lab/software>

Oprogramowanie to również pobiera się w pełni jako pakiet instalatora więc plik instalacyjny nadaje się do wielokrotnego użycia na komputerach i przechowywania na nośniku zewnętrznym do ponownego użycia w razie potrzeby reinstalacji. Z doświadczenia powiem, że **aktualny, sprawdzony i rozpoznany** pakiet instalacyjny przechowywany na nośniku zewnętrznym, który nie wymaga każdorazowego pobierania z internetu, jest cennym elementem warsztatu pracy każdego nauczyciela.

Wygląd panelu roboczego programu **LAB EV3 ver. 1.4.5**

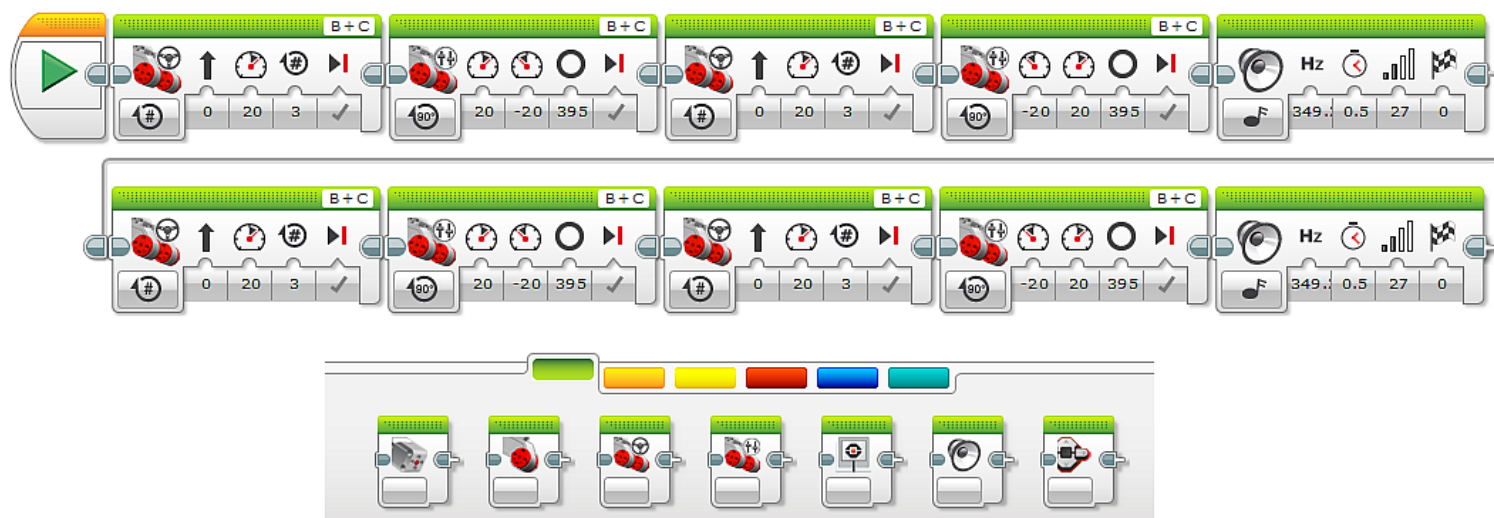


Jeśli uczniowie skonstruowali już robota bazowego i wiedzą, jak połączyć go z aplikacją programistyczną, kolejne zajęcia z robotyki będą już dla nich naprawdę atrakcyjne.

W dalszej części opracowania przygotowaliśmy dla Państwa szesnaście gotowych rozwiązań, które podczas początkowych zajęć pomogą uczniom w pracy z aplikacją **LAB EV3**.

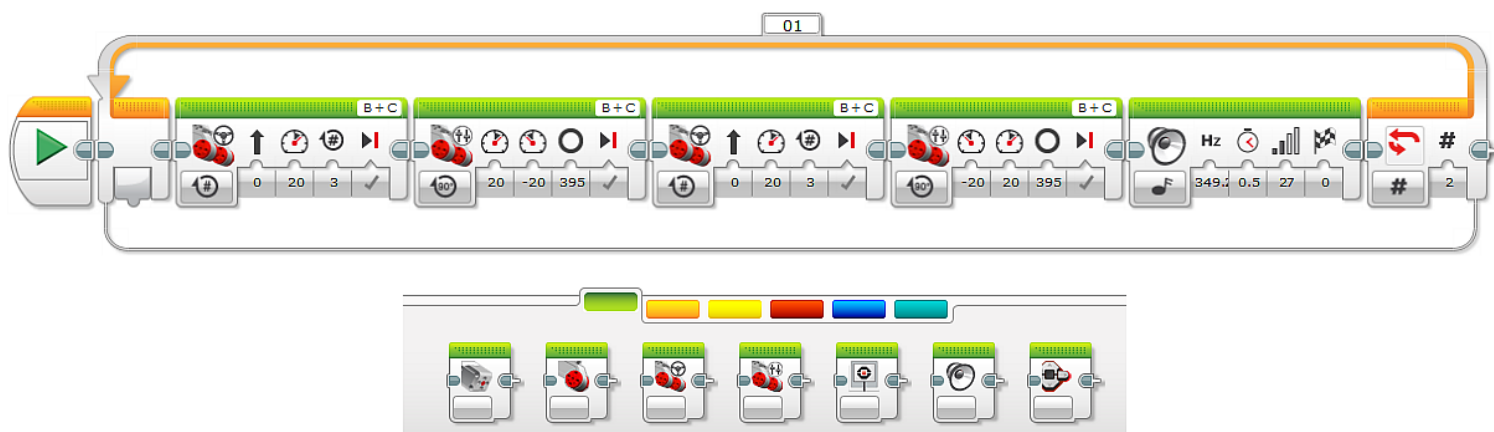
Zadanie 1.

Uruchomienie robota. Program: jazda i powrót w linii prostej



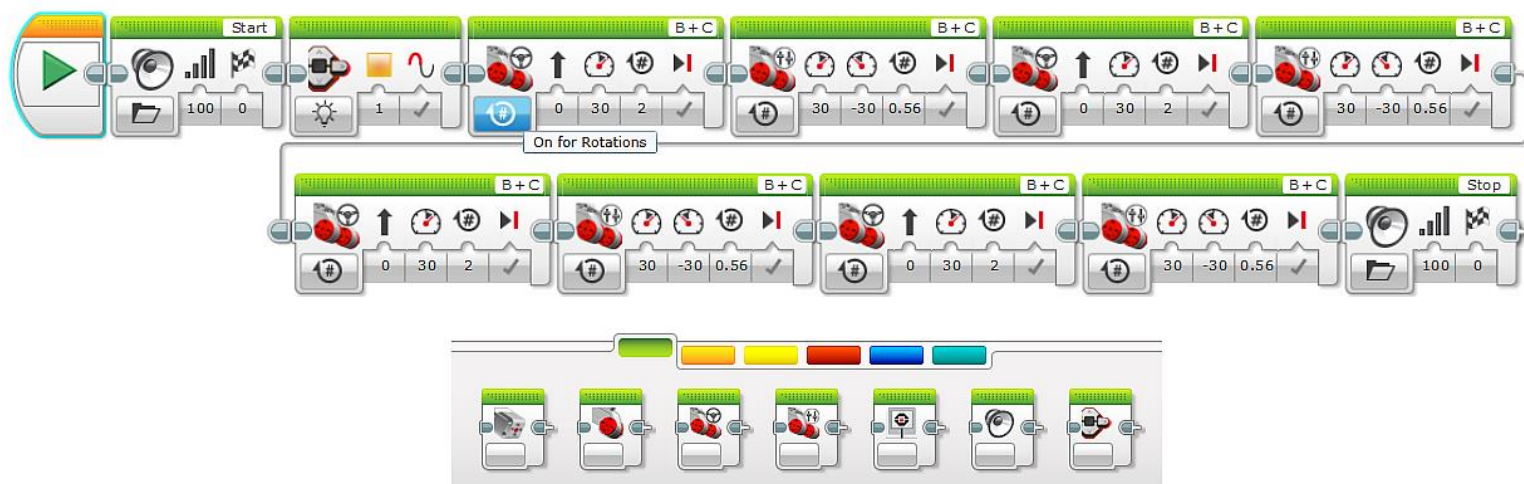
Opis działania:

1. Robot uruchamia program. Ustawia kolor lampy mikrokontrolera na zielony pulsacyjny (domyślny).
2. Prędkość obrotów silników kół robota ustawiona jest na 20 %.
3. Po rozpoczęciu pracy robot rusza w linii prostej. Porusza się z prędkością 20 % przez 3 obroty kół.
4. Po przejechaniu zaprogramowanej odległości robot wykonuje zwrot o 180 stopni.
5. Następnie robot wraca z prędkością 20 % przez 3 obroty kół.
6. Po przejechaniu tej samej odległości w odwrotnym kierunku robot ponownie wykonuje zwrot o 180 stopni.
7. Wykonanie ciągu zadań robot sygnalizuje dźwiękiem przez 0,5 sekundy.
8. W przypadku powyższym robot wykonuje zadaną pracę dwukrotnie bez użycia pętli.
9. W przypadku poniżej robot wykonuje analogiczne zadanie z użyciem pętli.



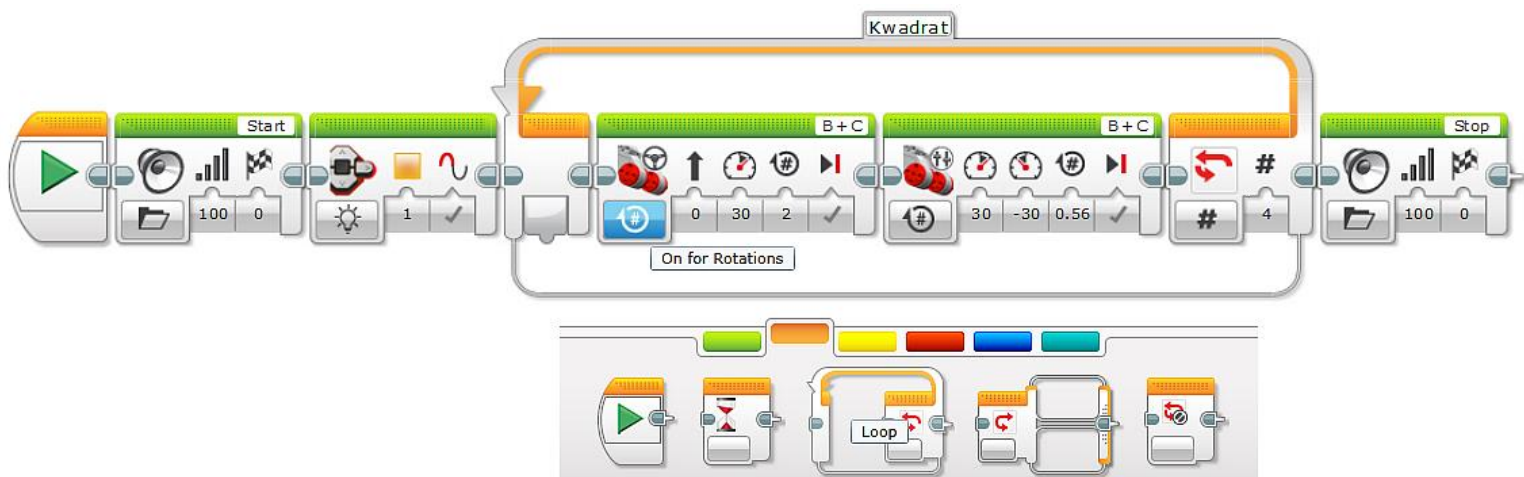
Zadanie 2.

Uruchomienie robota. Program: jazda po obwodzie kwadratu



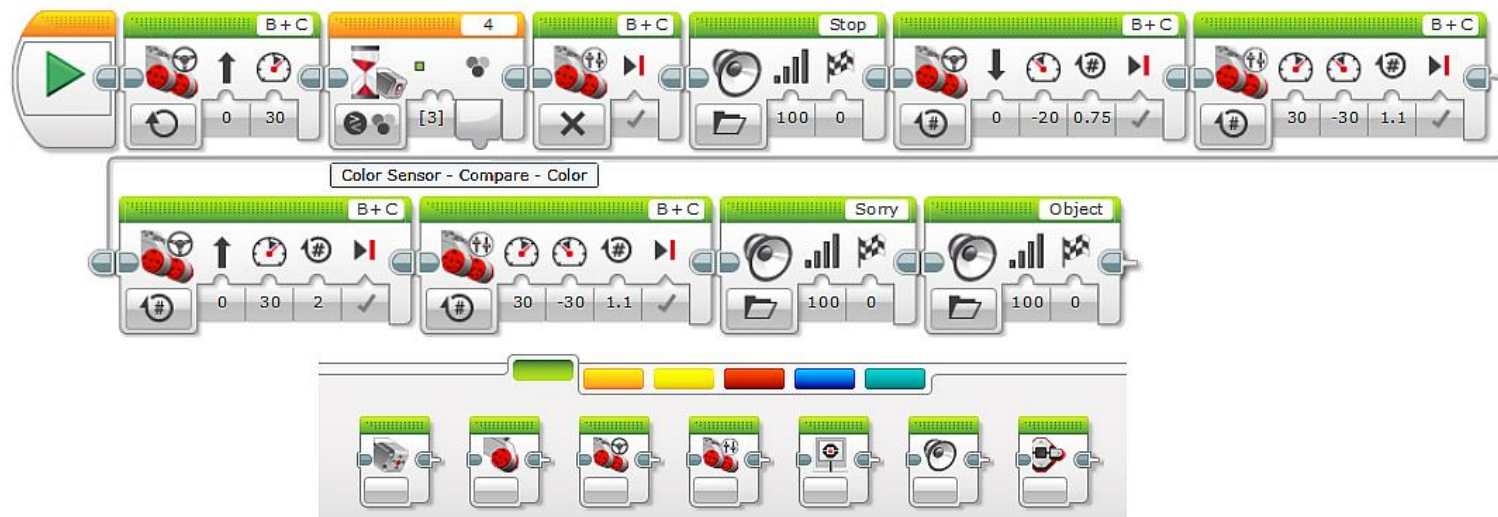
Opis działania:

1. Robot uruchamia program. Rozpoczęcie pracy sygnalizuje dźwiękiem „Start”. Ustawia kolor lampy mikrokontrolera na orange pulsacyjny.
2. Prędkość obrotów silników kół robota ustawiona jest na 30 %.
3. Po rozpoczęciu pracy robot rusza w linii prostej. Porusza się z prędkością 30 % przez 2 obroty kół.
4. Po przejechaniu zaprogramowanej odległości robot wykonuje zwrot o 90 stopni.
5. W trakcie pracy robot czterokrotnie powtarza zadania opisane w pkt. 3 i 4.
6. Wykonanie ciągu zadań robot sygnalizuje dźwiękiem „Stop”.
8. W przypadku powyższym robot wykonuje zadaną pracę bez użycia pętli.
9. W przypadku poniżej robot wykonuje analogiczne zadanie z użyciem pętli.



Zadanie 3.

Uruchomienie robota. Program: reakcja czujnika koloru



Opis działania:

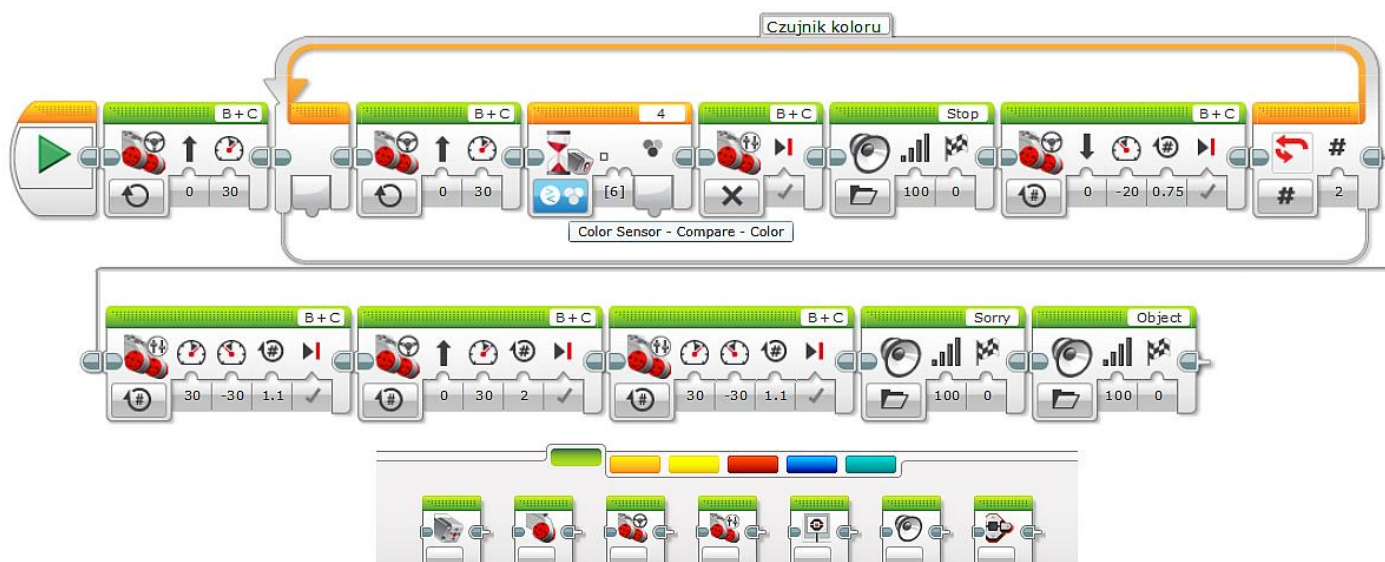
Moduł pierwszy – powierzchnia bez elementów koloru białego:

Podczas pracy robot wyposażony jest w czujnik koloru podłączony do złącza 4 mikrokontrolera.

Robot po powierzchni bez elementów koloru zielonego (lub białego) wykonuje pracę ciągłą – jazdę prosto.

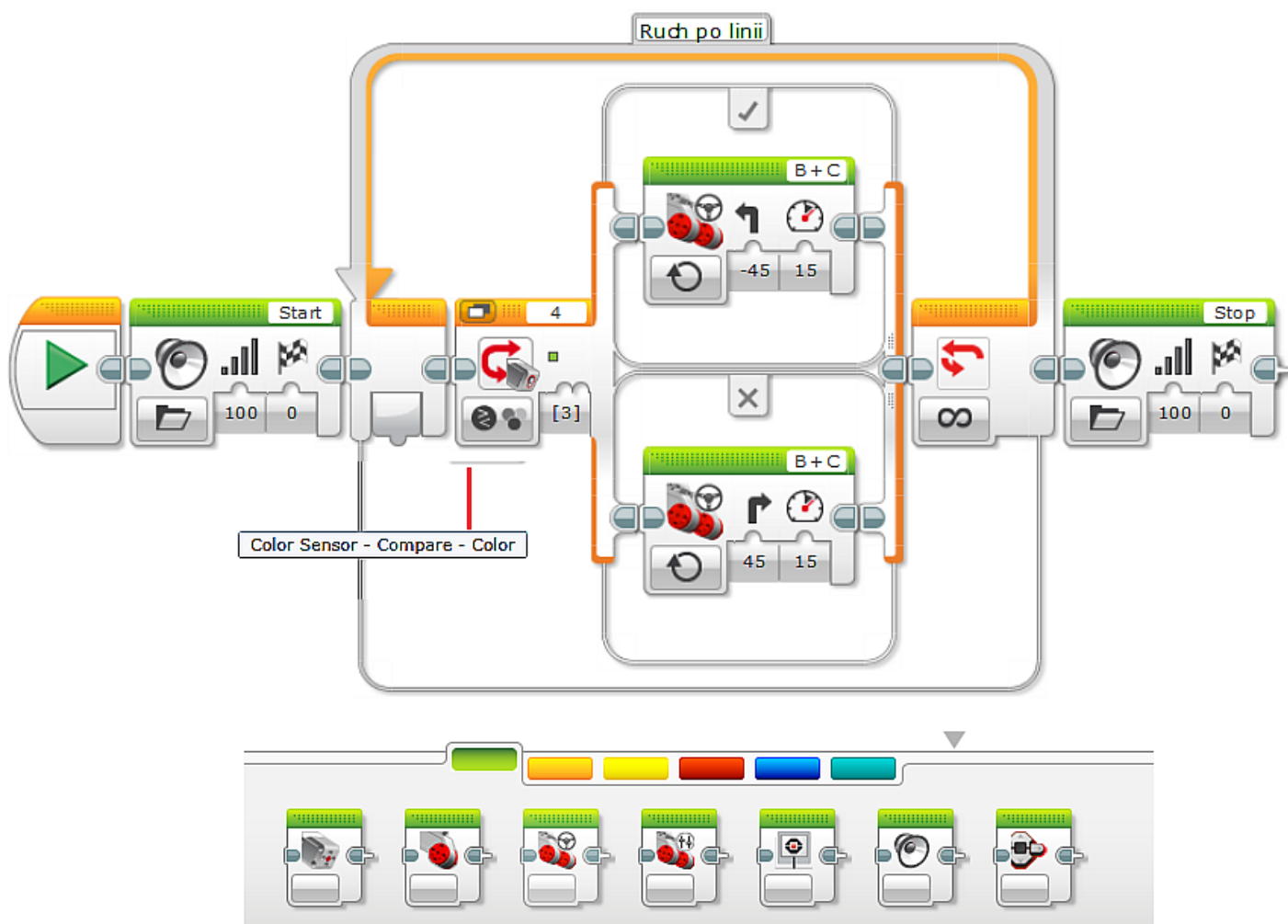
Moduł drugi – powierzchnia z elementem koloru zielonego:

1. Robot po wykryciu przez czujnik na torze elementu koloru zielonego, przerywa pracę skryptu pierwszego.
2. Po zatrzymaniu się generuje sygnał dźwiękowy „Stop”.
3. Zmienia prędkość obrotów silników kół robota na 20 %.
4. Po ustaniu sygnału robot wycofuje się o 0,75 obrotu kół w tył.
5. Po przejechaniu w tył zaprogramowanej odległości robot wykonuje zwrot o 180 stopni.
6. Następnie robot wraca z prędkością 30 % przez 2 obroty kół.
7. Po przejechaniu z powrotem zaprogramowanej odległości robot ponownie wykonuje zwrot o 180 stopni.
8. Wykonanie ciągu zadań robot sygnalizuje informacją dźwiękową „Sorry. Obiekt”.
9. W przypadku powyższym robot wykonuje zadaną pracę bez użycia pętli. Wykrywa kolor zielony.
10. W przypadku poniżej robot wykonuje zadanie z użyciem pętli, podejmując dwukrotnie próbę przejazdu przeszkody. Tym razem wykrywa kolor biały.



Zadanie 4.

Czujnik kolorów. Program: poruszanie się robota wzdłuż linii

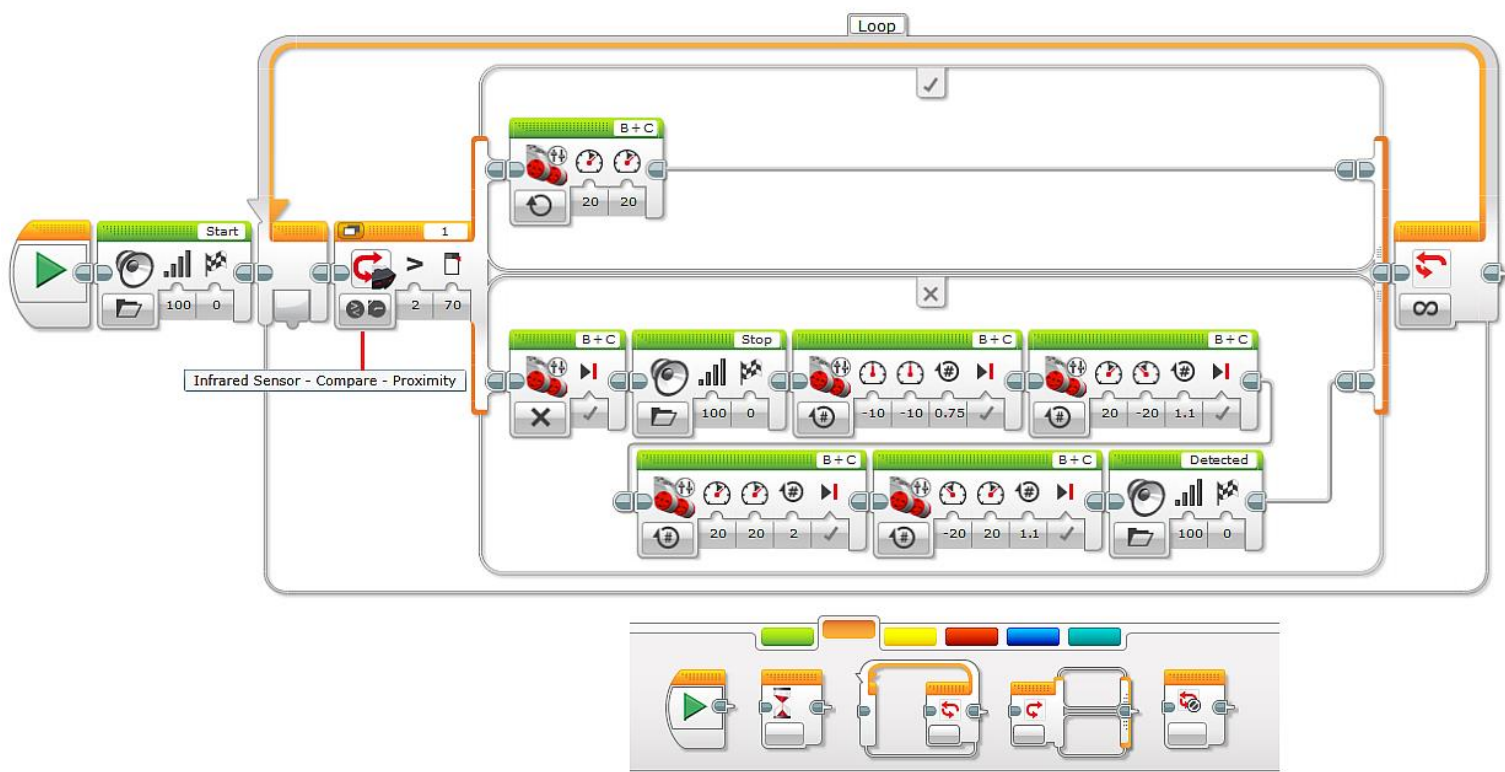


Opis działania:

1. Robot uruchamia program. Generuje komunikat „Start”.
2. Rozpoczynając pracę robot porusza się w prawo aż do momentu wykrycia linii o kolorze zielonym.
3. Po wykryciu przez sensor linii koloru zielonego, robot zaczyna poruszać się zygzakiem wzdłuż jej pasa.
2. Prędkość obrotów silników kół robota ustawiona jest na 15 %.
3. W pętli zadeklarowane są dwa warunki:
 - jeśli sensor wykrywa zieloną linię, robot skręca po łuku w lewo oddalając się od niej (blok górny),
 - jeśli sensor nie wykrywa zielonej linii, pojazd skręca po łuku w prawo w kierunku linii (blok dolny).
5. Robot porusza się w pętli bez ograniczenia czasowego lub obrotowego kół. W tym przypadku zatrzymanie pracy robota odbywa się z poziomu programu sterującego na komputerze.
6. Po zatrzymaniu robota generuje on komunikat „Stop”.

Zmniejszając wartość parametru **Steering** (-45 i 45) robot wykonuje łagodniejsze skręty, co jest zaletą podczas jazdy po linii prostej, jednak traci na precyzji w pokonywaniu skrętów.

Czujnik podczerwieni. Program: reakcja robota na przeszkodę na torze



Moduł pierwszy – powierzchnia bez przeszkody na torze:

Robot po powierzchni bez przeszkody wykonuje pracę ciągłą – jazdę prosto.

1. Robot po wykryciu przez czujnik podczerwieni przeszkody na torze, przerywa pracę skryptu pierwszego.

3. Zmienia prędkość obrotów silników kół robota na 10 % z jazdą wsteczną (znak minus).

5. Po przejechaniu w tył zaprogramowanej odległości robot wykonuje zwrot o 180 stopni.

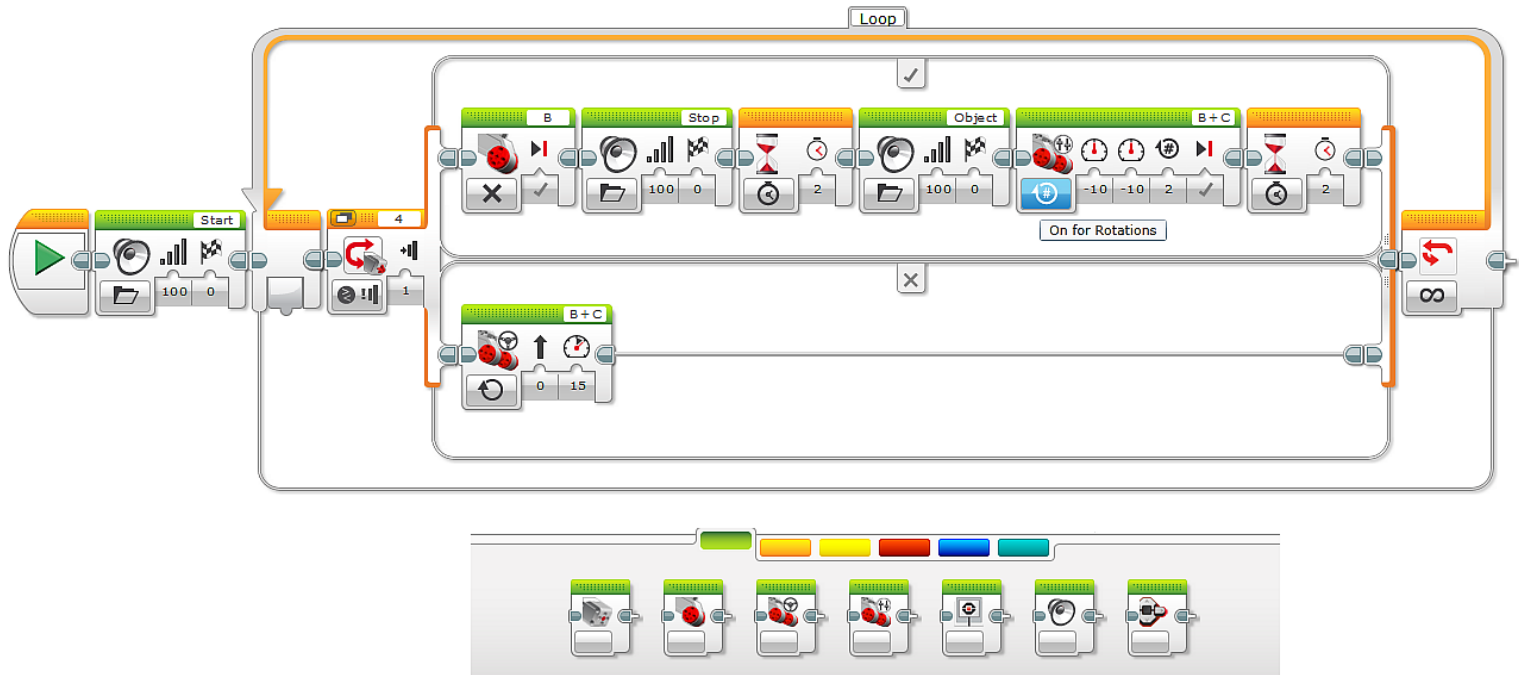
7. Po przejechaniu z powrotem zaprogramowanej odległości robot ponownie wykonuje zwrot o 180 stopni.

9. W tym programie robot wykonuje zadanie z użyciem pętli, podejmując próbę przejazdu przeszkody.

zatrzymanie pracy robota odbywa się z poziomu programu sterującego na komputerze.

Zadanie 6.

Czujnik dotyku. Program: reakcja robota na zaporę na torze



Opis działania:

Moduł pierwszy – powierzchnia bez zaporę na torze:

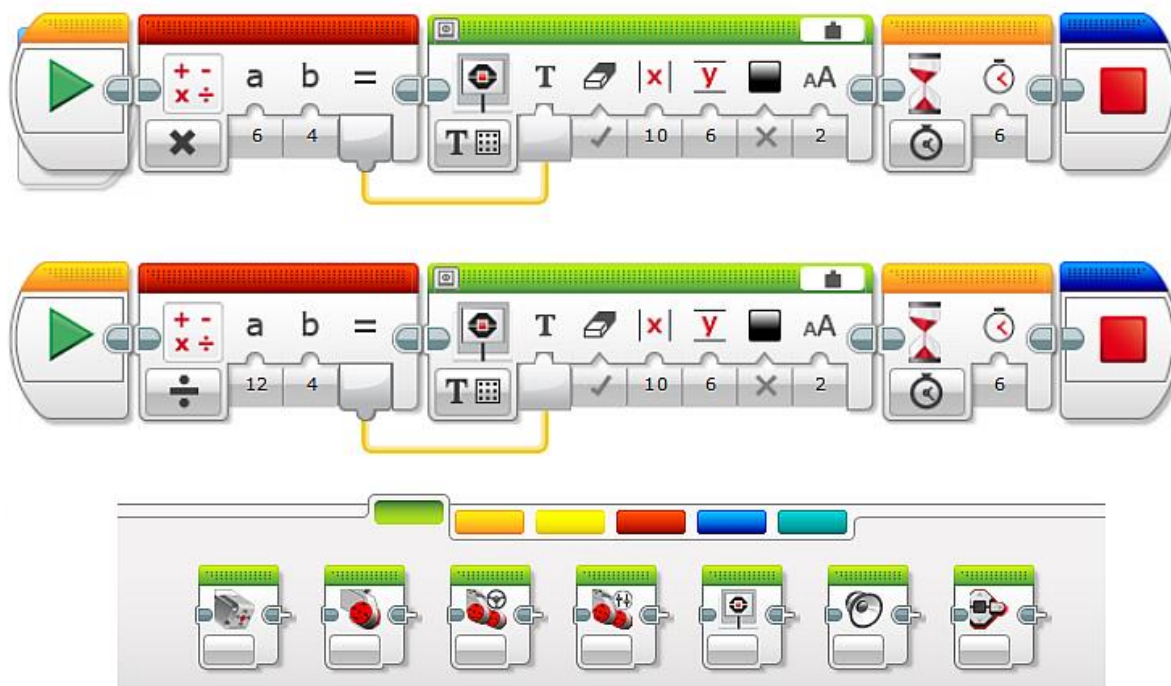
Podczas pracy robot wyposażony jest w czujnik dotyku podłączony do złącza 2 mikrokontrolera. Robot po powierzchni bez zaporę wykonuje pracę ciągłą – jazdę prosto.

Moduł drugi – powierzchnia z ustawioną zaporą na torze:

1. Robot po wykryciu przez czujnik dotyku zaporę na torze, przerywa pracę skryptu pierwszego.
2. Po zatrzymaniu się generuje sygnał dźwiękowy „Stop”.
3. Zmienia prędkość obrotów silników kół robota na 10 % z jazdą wsteczną (znak minus).
4. Po sygnale robot wycofuje się o 2 obroty kół w tył.
5. Wykonanie ciągu zadań robot sygnalizuje informacją dźwiękową „Obiekt”.
6. Następnie robot zaczyna sekwencję programu od początku.
7. W tym programie robot wykonuje zadanie z użyciem pętli, podejmując próbę wielokrotnie. Robot porusza się w pętli bez ograniczenia czasowego lub obrotowego kół. W tym przypadku zatrzymanie pracy robota odbywa się z poziomu programu sterującego na komputerze.

Zadanie 7.

Program: operatory matematyczne podstawowe



Opis działania:

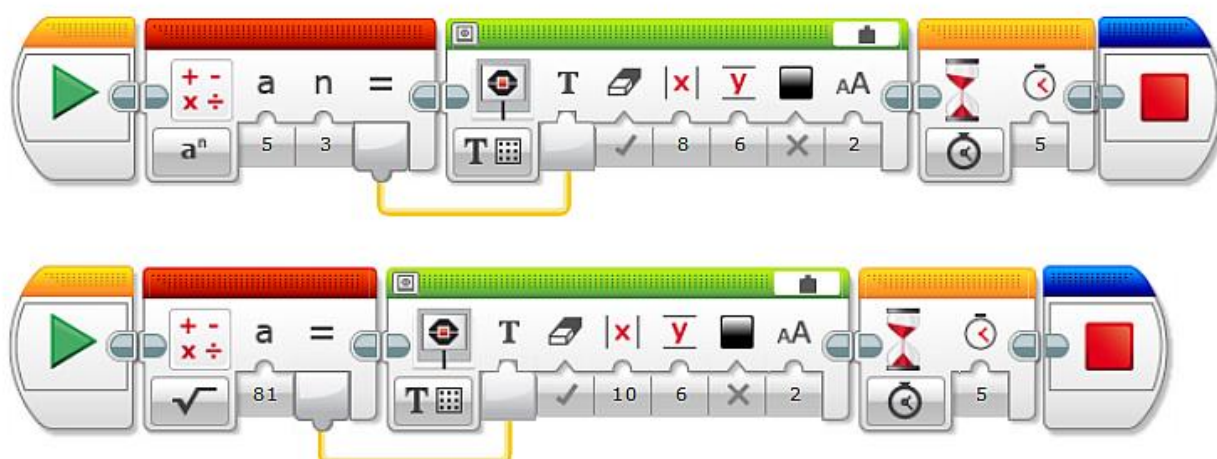
Program dokonuje obliczeń wyników działań na liczbach zadeklarowanych w blokach modułowych oznaczonych kolorem czerwonym. Wybrany jest rodzaj działania i wpisane są liczby. W blokach oznaczonych kolorem zielonym ustawione są współrzędne wyniku i wielkość czcionki, natomiast w bloku oznaczonym kolorem pomarańczowym ustawiony jest czas generowania wyniku. Blok niebieski kończy pracę operatora.

Po rozpoczęciu pracy program generuje wynik na ekranie mikrokontrolera przez 6 sekund.

Każdy blok modułowy uruchamiany jest odrębnie ponieważ każdy z nich pracuje niezależnie od innych.

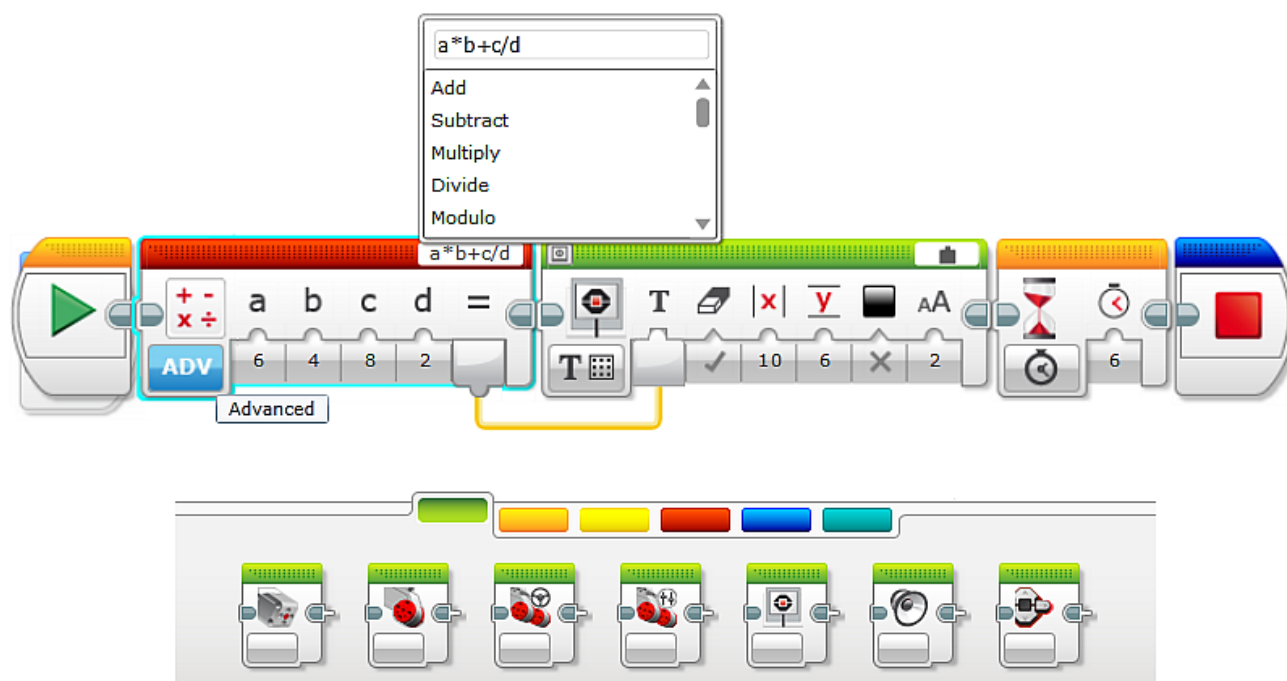
W przypadku powyżej program dokonuje obliczeń mnożenia i dzielenia liczb (6×4 lub $12 : 4$).

W przypadku poniżej program oblicza i generuje wynik potęgowania lub pierwiastka liczby (5^3 lub $\sqrt{81}$).



Zadanie 8.

Program: operatory matematyczne zaawansowane



Opis działania:

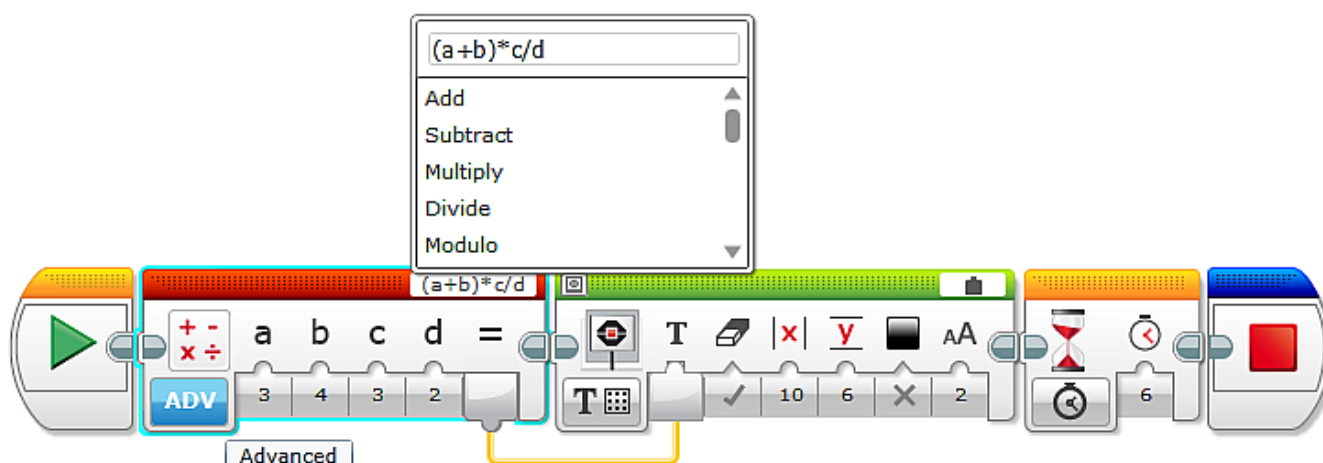
Program dokonuje obliczeń wyników działań na liczbach zadeklarowanych w blokach modułowych oznaczonych kolorem czerwonym. Wybrany jest rodzaj działania „Zaawansowane” i wpisane są liczby. W blokach oznaczonych kolorem zielonym ustawione są współrzędne wyniku i wielkość czcionki, natomiast w bloku oznaczonym kolorem pomarańczowym ustawiony jest czas generowania wyniku. Blok niebieski kończy prace operatora.

Po rozpoczęciu pracy program generuje wynik na ekranie mikrokontrolera przez 6 sekund.

Każdy blok modułowy uruchamiany jest odrębnie ponieważ każdy z nich pracuje niezależnie od innych.

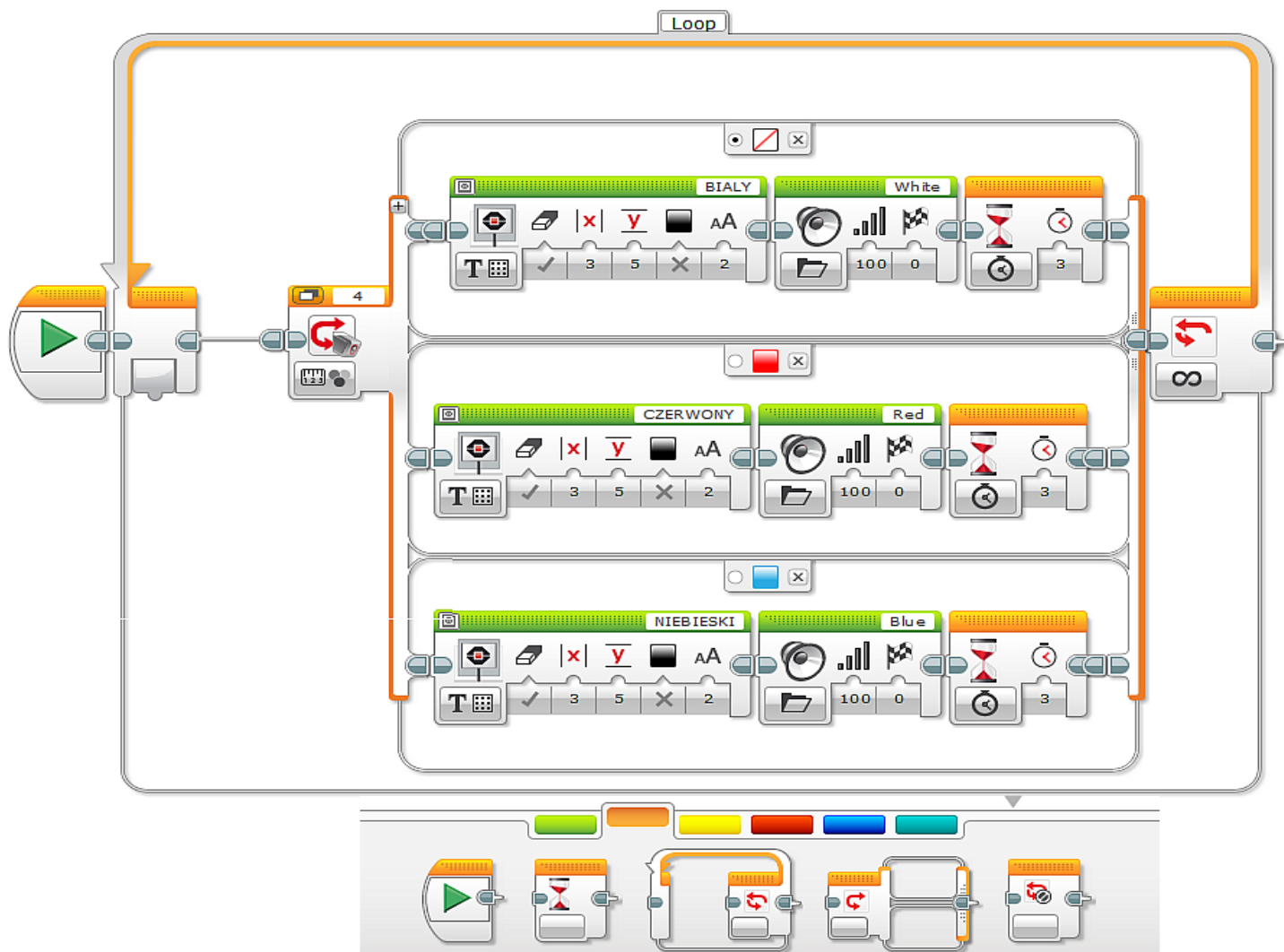
W przypadku powyżej program dokonuje obliczeń według dowolnego wzoru ułożonego przez użytkownika.

W przypadku poniżej program oblicza i generuje wartość pola trapezu.



Zadanie 9.

Czujnik koloru. Program: czytnik kolorów – barwy podstawowe

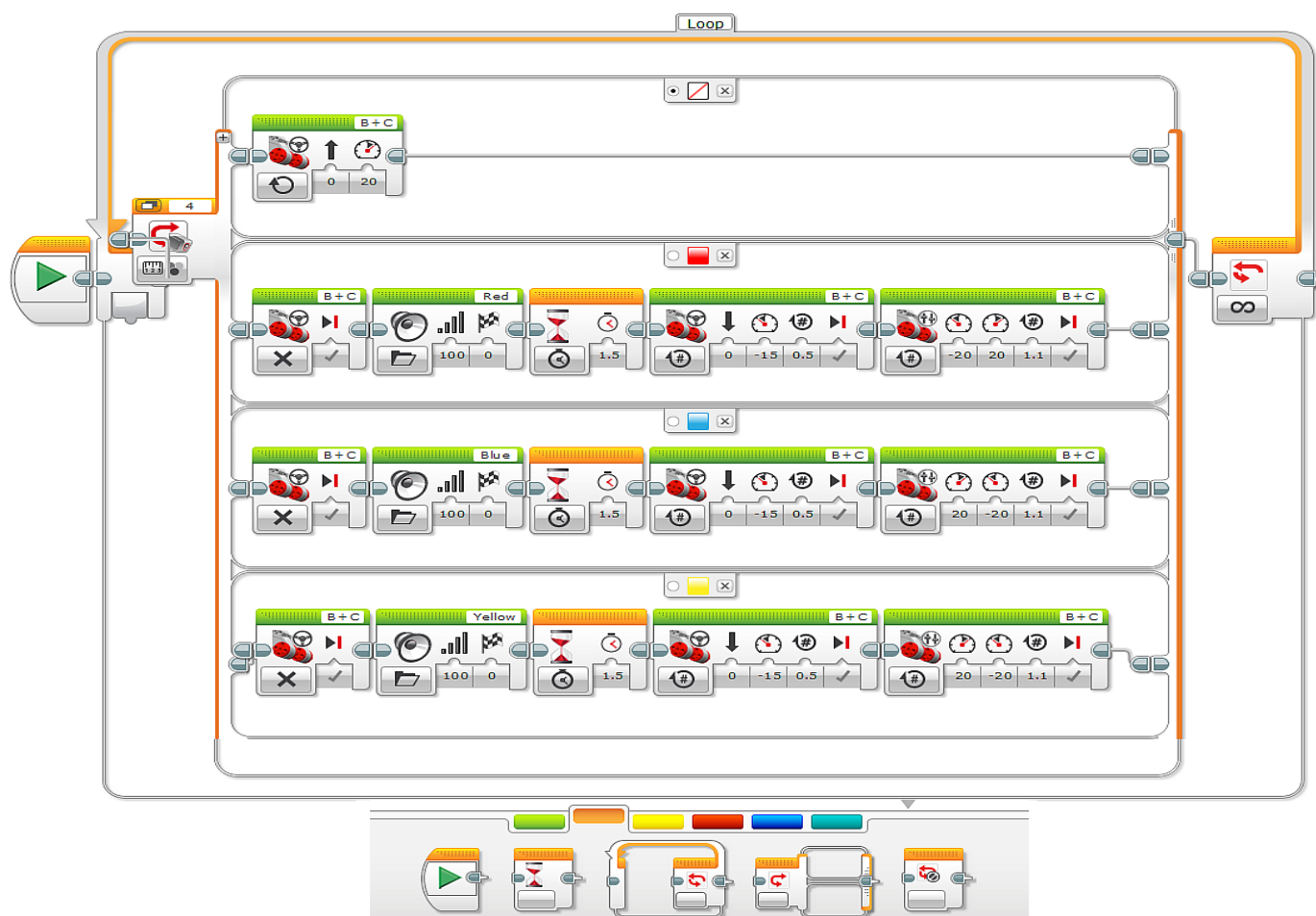


Opis działania:

Program rozpoznaje podstawowe kolory. Czujnik kolorów podłączony jest do gniazda 4 mikrokontrolera. Jeśli czujnik nie wykrywa żadnego koloru, wówczas światło dzienne rozpoznaje jako kolor biały. Na ekranie mikrokontrolera widnieje napis „BIAŁY”, co 3 sekundy robot generuje komunikat dźwiękowy „White”. Jeśli do czujnika przyłożymy element w kolorze czerwonym, na ekranie mikrokontrolera zmieni się napis na „CZERWONY”. Wówczas robot co 3 sekundy generuje komunikat dźwiękowy „Red”. Jeśli do czujnika przyłożymy element w kolorze niebieskim, na ekranie mikrokontrolera pojawi się napis „NIEBIESKI”. Robot co 3 sekundy generuje komunikat dźwiękowy „Blue”. Wszystkie moduły programu działają w pętli, dlatego generują poszczególne komunikaty do momentu zmiany barwy lub do zatrzymania programu z poziomu komputera. Deklarowane w blokach barwy mogą być zmieniane, można także rozbudować program dodając kolejne moduły z nowymi kolorami. Współrzędne X i Y są współrzędnymi pozycjonowania napisu na ekranie mikrokontrolera.

Zadanie 10.

Czujnik koloru. Program: wykrywacz kolorów na obszarze pracy



Opis działania:

Robot wykrywa i rozpoznaje podstawowe kolory na obszarze swojej pracy. Czujnik kolorów podłączony jest do gniazda 4 mikrokontrolera. Gdy czujnik nie wykrywa kolorów deklarowanych w programie, robot wykonuje pracę ciągłą - jazdę w przód. Podczas pracy czujnik robota reaguje na trzy kolory: czerwony, niebieski lub żółty. Po wykryciu na torze jazdy jednego z powyższych kolorów robot:

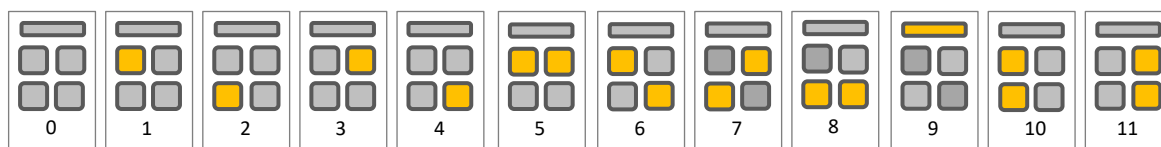
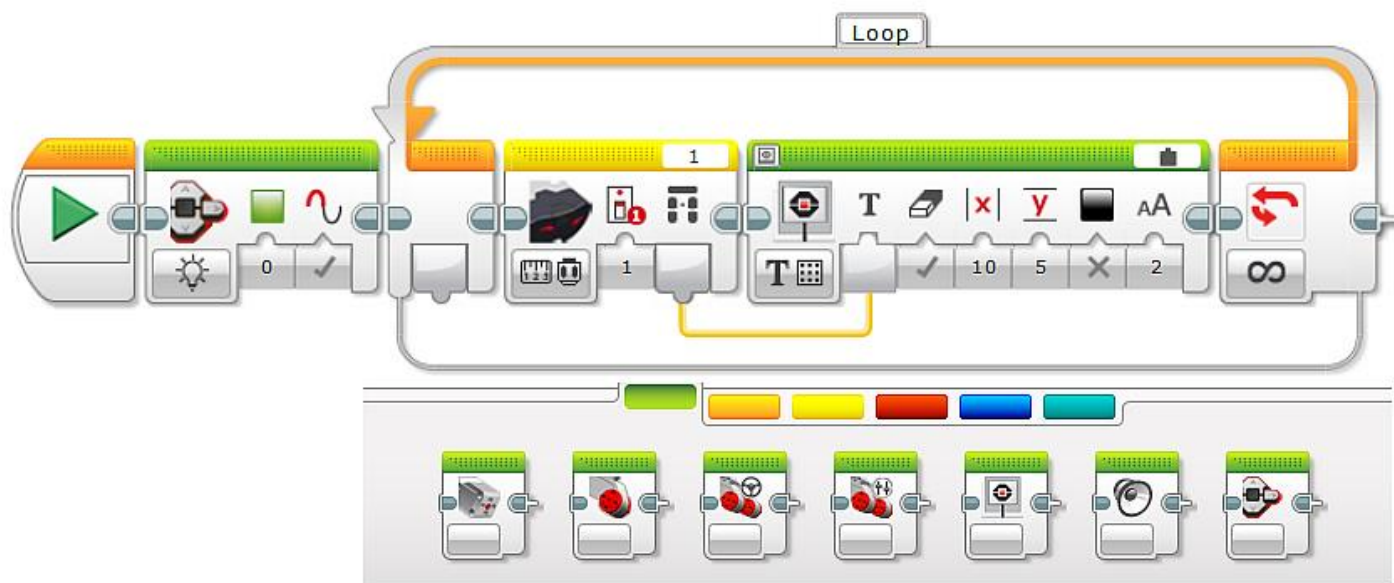
- zatrzymuje pracę silników kół,
- generuje sygnał dźwiękowy odpowiedni dla wykrytego koloru (Red, Blue lub Yellow),
- zatrzymuje pracę robota na 1,5 sekundy,
- wykonuje zwrot o 180° i rozpoczyna jazdę w przeciwnym kierunku.

Program działa w pętli, dlatego po każdym wykryciu jednego z kolorów robot rozpoczyna pracę od początku. Możliwa jest zmiana bieżących kolorów do wykrycia przez robota lub rozbudowa programu o kolejne moduły z nowymi kolorami.

W tym przypadku zatrzymanie pracy robota odbywa się z poziomu programu sterującego na komputerze.

Zadanie 11.

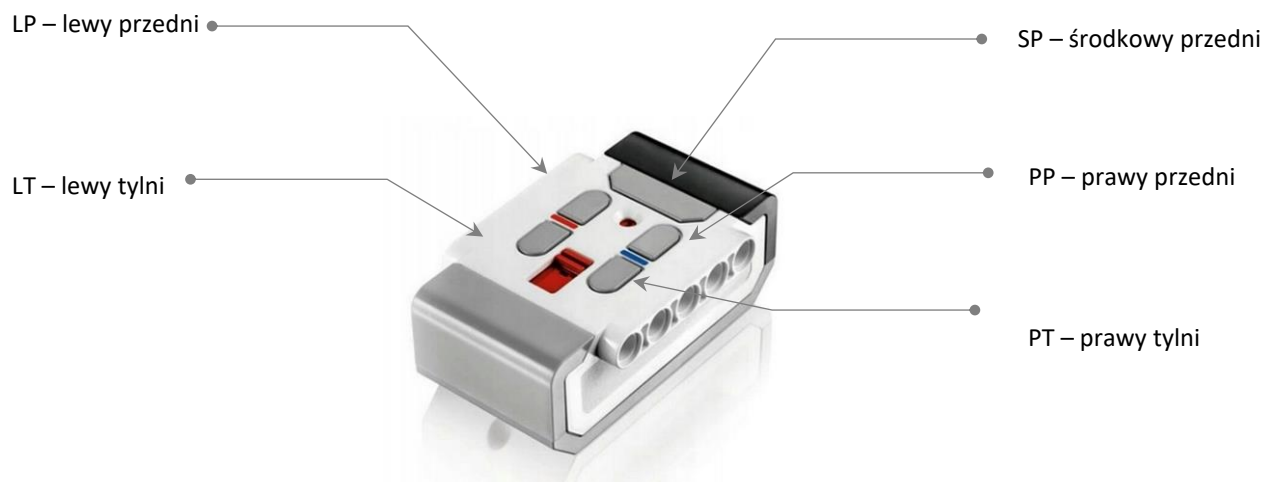
Czujnik podczerwieni. Program: generowanie liczb za pomocą przycisków pilota



Opis działania:

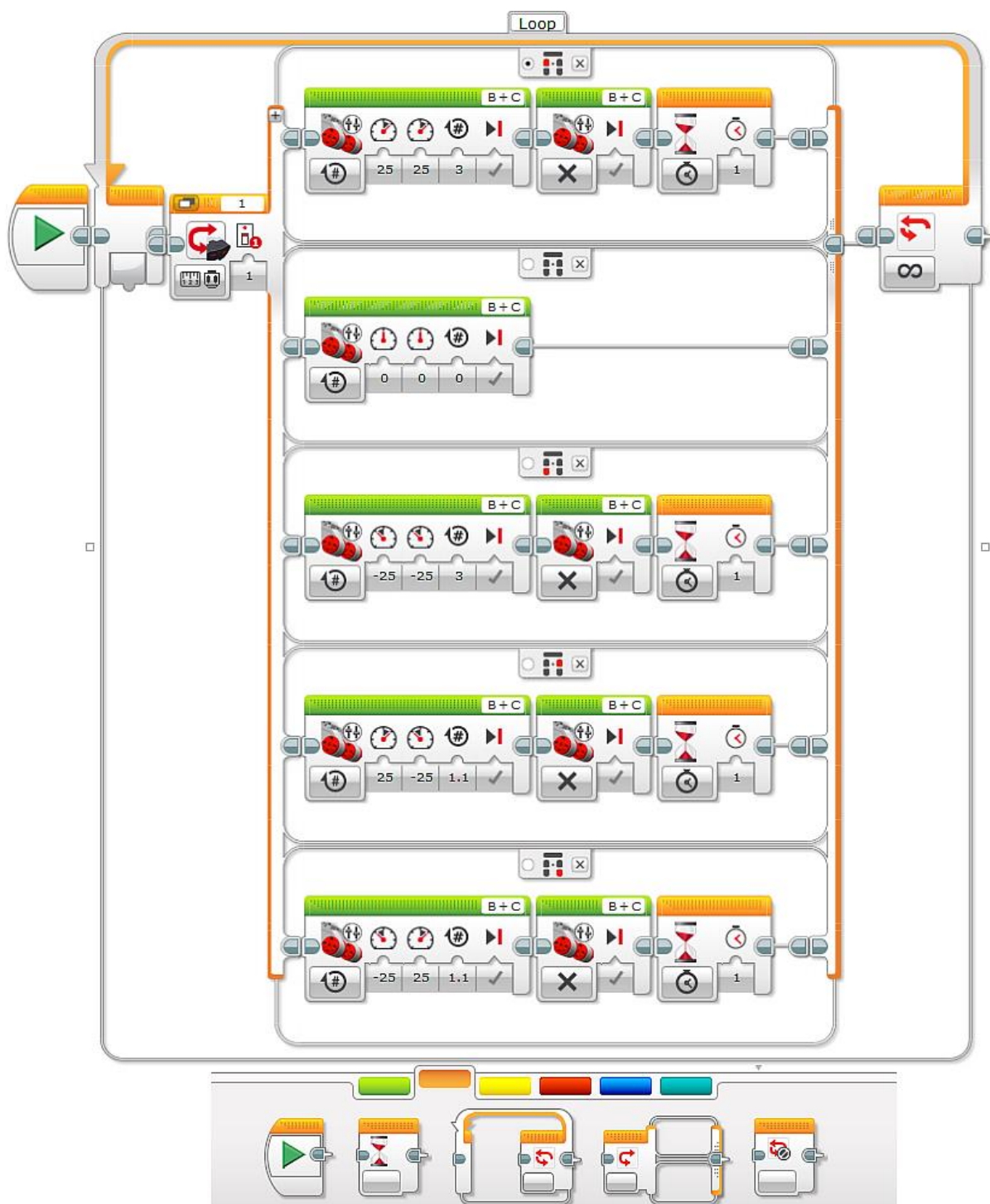
Robot wyposażony jest w czujnik podczerwieni oraz pilot sterujący (). Po uruchomieniu programu i wciśnięciu odpowiedniego przycisku lub kombinacji przycisków pilota, na mikrokontrolerze robota pojawiają się liczby od 0 do 11, w zależności od sekwencji podanej na ilustracji powyżej.

Opis poniżej przedstawia objaśnienia skrótów nazw przycisków używanych w dalszych modułach programowania robota, z użyciem pilota i czujnika podczerwieni.



Zadanie 12.

Czujnik podczerwieni. Program: jazda i powrót sterowane pilotem



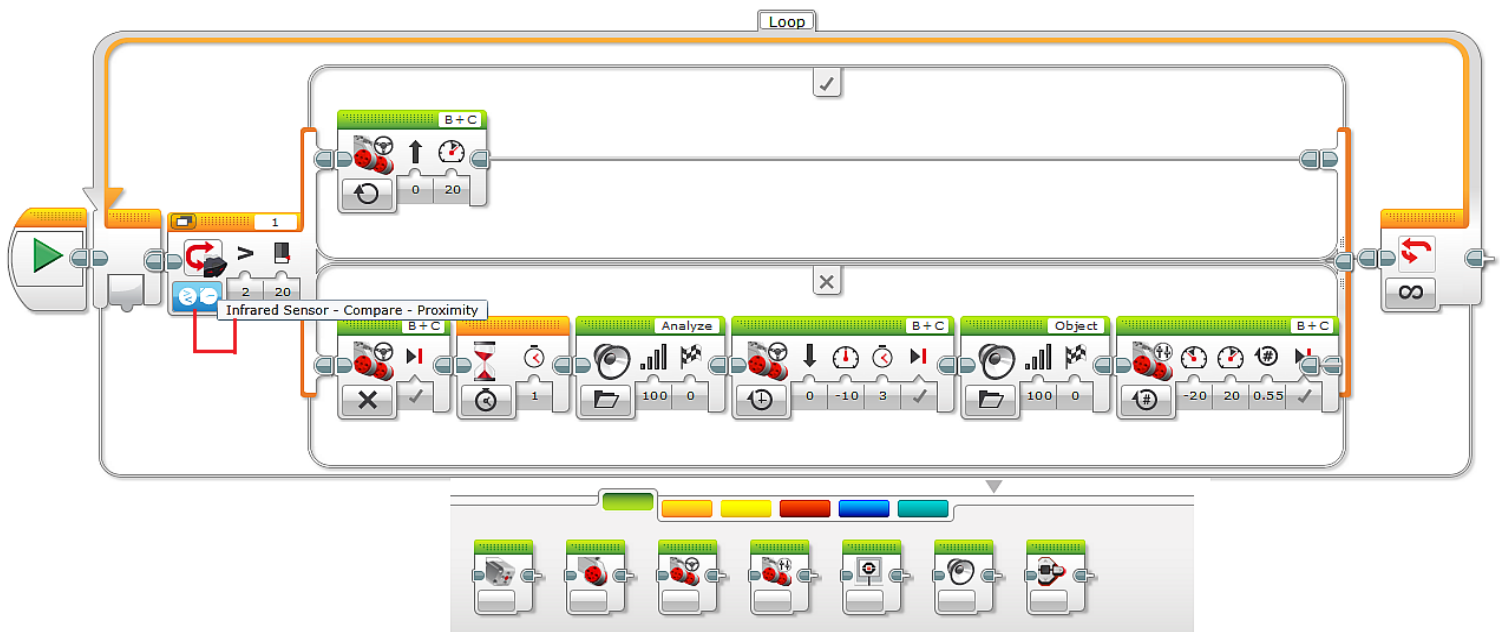
Opis działania:

Robot wyposażony jest w czujnik podczerwieni oraz pilot sterujący (•) Po uruchomieniu programu i wciśnięciu odpowiedniego przycisku na pilocie następuje rozpoczęcie pracy:

- przycisk LP – robot jedzie w przód na odległość 3 obrotów kół i zatrzymuje się.
- przycisk LT – robot jedzie w tył na odległość 3 obrotów kół i zatrzymuje się.
- przycisk PP lub PT – robot wykonuje zwrot w prawo lub w lewo o 18 stopni.



Czujnik podczerwieni. Program: reakcja robota na przeszkodę. Zmiana kierunku jazdy

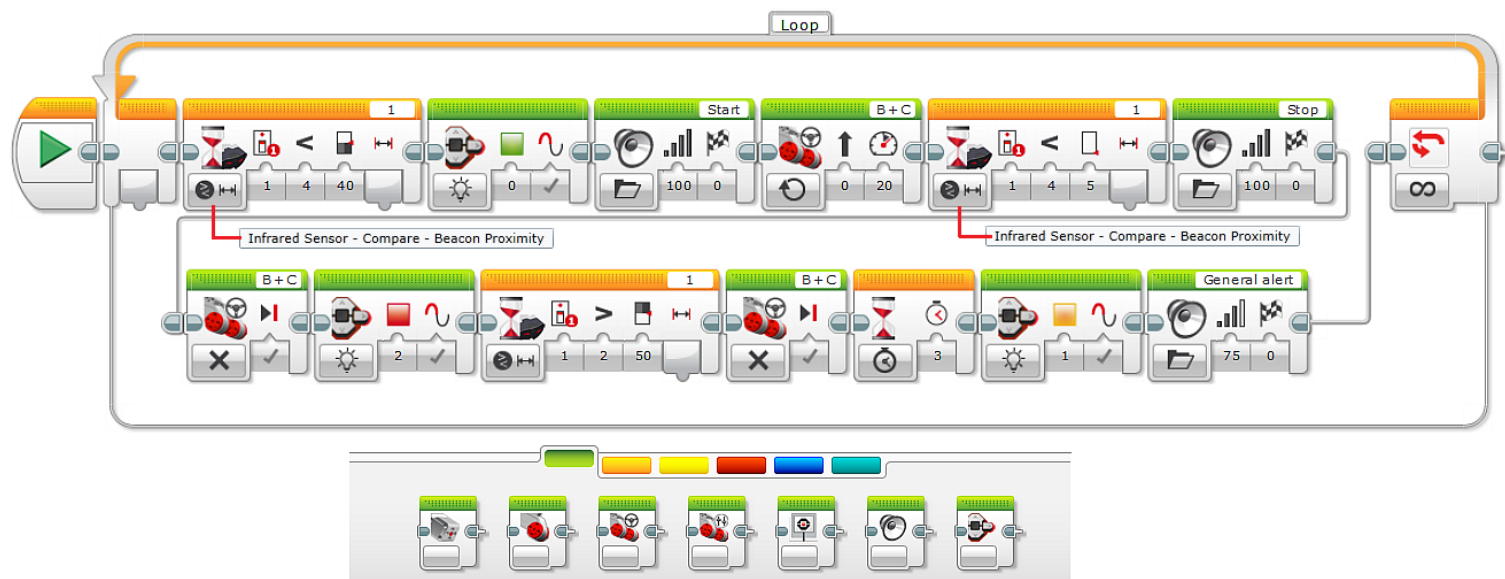


Robot wyposażony jest w czujnik podczerwieni zamontowany w przedniej, środkowej części robota. Podłączony jest przewodem do gniazda nr 4 mikrokontrolera.

1. Robot uruchamia program. Ustawia kolor lampy mikrokontrolera na zielony pulsacyjny (domyślny).
2. Prędkość obrotów silników kół robota ustawiona jest na 20 %.
3. Po rozpoczęciu pracy robot rusza w linii prostej. Porusza się z prędkością 20 %. Wykonuje jazdę ciągłą.
4. Po wykryciu przez czujnik podczerwieni przeszkody na torze, robot zatrzymuje pracę na czas 1 sekundy.
5. Następnie robot generuje komunikat „Analize” i rusza w tył z prędkością 10% przez 3 sekundy.
6. Po wycofaniu się i zatrzymaniu, robot generuje komunikat „Object”.
7. Następnie robot wykonuje zwrot w lewo o 90 stopni i kontynuuje jazdę prosto.
8. Po napotkaniu kolejnej przeszkody, robot ponownie wykonuje pracę opisaną w punktach od 4 do 7.
9. Praca robota wykonywana jest w trybie ciągłym, ponieważ program działa w pętli. Zatrzymanie pracy robota odbywa się z poziomu programu na komputerze, lub po wciśnięciu przycisku „Back” (Wstecz) na kostce mikrokontrolera.

Zadanie 14.

Czujnik podczerwieni. Program: rozruch robota pilotem w trybie Beacon



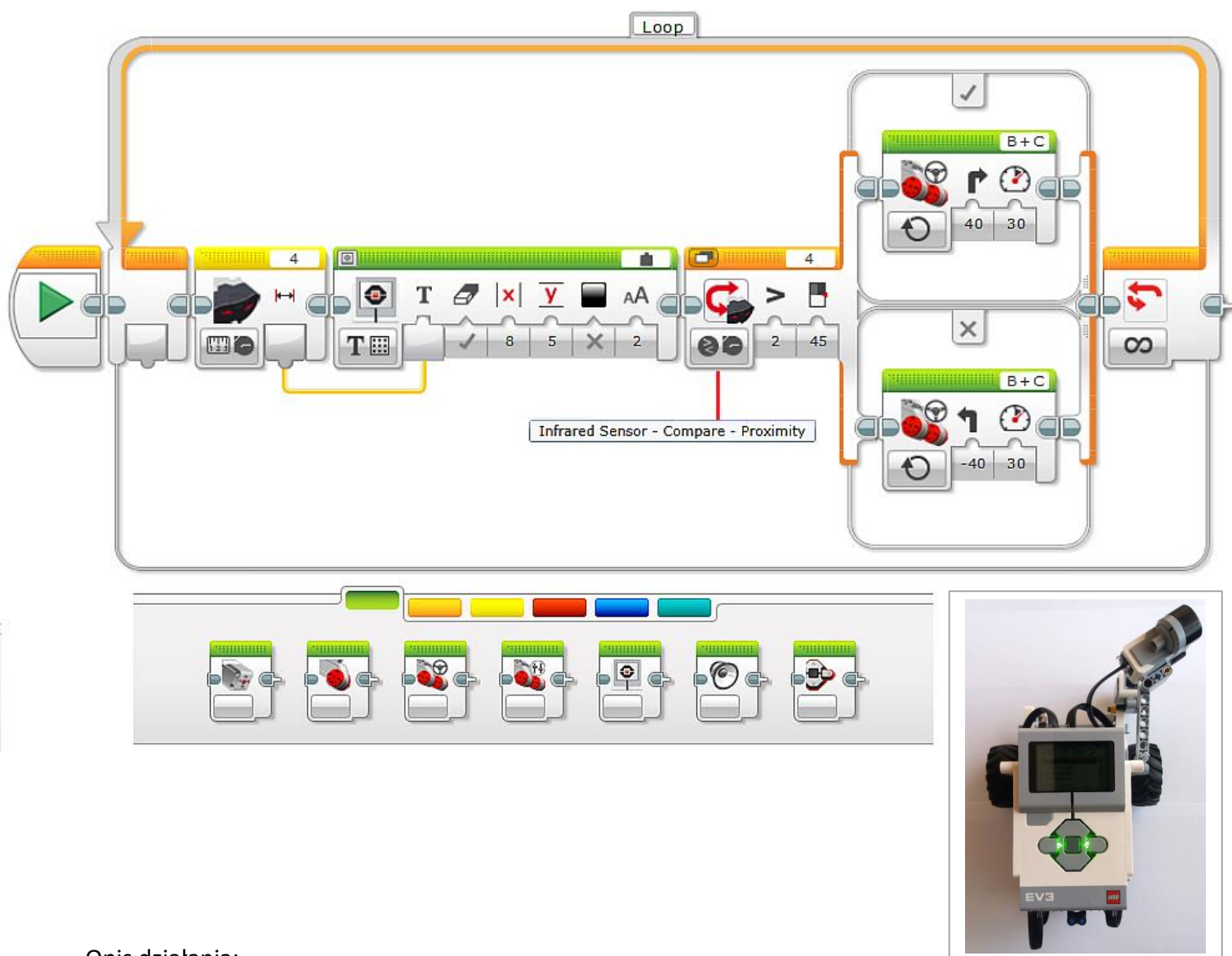
Opis działania:

Robot wyposażony jest w pilot oraz w czujnik podczerwieni zamontowany w przedniej, środkowej części robota. Czujnik podłączony jest przewodem do gniazda nr 4 mikrokontrolera. Pilot należy przełączyć w tryb pracy **Beacon**. Podczas pracy w tym trybie świeci zielona dioda pilota.

1. Robot uruchamia program. Ustawia kolor lampy mikrokontrolera na zielony pulsacyjny (domyślny).
2. Prędkość obrotów silników kół robota ustawiona jest na 20 %.
3. Po rozpoczęciu pracy programu robot nie wykonuje pracy mechanicznej, sygnalizuje gotowość zielonym kolorem pulsacyjnym lampy na mikrokontrolerze. Pilot znajduje się poza zasięgiem czujnika podczerwieni.
4. Po zbliżeniu pilota do czujnika podczerwieni na odległość mniejszą niż wartość 40, robot rusza w przód z prędkością 20 %. Rozpoczęcie jazdy poprzedzone jest komunikatem „Start”. Podczas jazdy robota w przód lampa mikrokontrolera pulsuje w kolorze zielonym.
5. Gdy pilot znajdzie się w odległości mniejszej niż wartość 5, wówczas robot zatrzymuje się, generując sygnał dźwiękowy „Stop”. Kolor pulsacyjny lampy zmienia się na czerwony.
6. W czasie, gdy pilot znajduje się w odległości mniejszej niż wartość 5, robot stoi w miejscu przez cały czas, lampa mikrokontrolera nadal miga w kolorze czerwonym.
7. Po oddaleniu pilota od czujnika podczerwieni na większą odległość, po 3 sekundach robot generuje komunikat „General alert”, kolor pulsacyjny lampy zmienia się na pomarańczowy.
8. Po ponownym zbliżeniu pilota do czujnika podczerwieni na odległość mniejszą niż wartość 40, robot ponownie rusza w przód, lampa mikrokontrolera pulsuje w kolorze zielonym.
9. Praca robota wykonywana jest w trybie ciągłym, ponieważ program działa w pętli. Zatrzymanie pracy robota odbywa się z poziomu programu na komputerze, lub po wciśnięciu przycisku „Back” (Wstecz) na kostce mikrokontrolera.

Zadanie 15.

Czujnik podczerwieni. Program: jazda robota wzdłuż krawędzi ściany



Opis działania:

Robot wyposażony jest w czujnik podczerwieni zamontowany z prawej strony, pionowo, pod kątem 45 stopni (jak na zdjęciu powyżej). Czujnik podłączony jest przewodem do gniazda nr 4 mikrokontrolera.

1. Robot uruchamia program. Ustawia kolor lampy mikrokontrolera na zielony pulsacyjny (domyślny).
2. Prędkość obrotów silników kół robota ustawiona jest na 30 %. Wartość skrętu podczas jazdy: 40 lub -40. Wartość progowa (Threshold Value) zbliżenia czujnika podczerwieni do ściany: 45.
3. Po rozpoczęciu pracy programu robot zaczyna pracę skręcając w lewo lub w prawo w zależności od wartości progowej (odległość czujnika od ściany $<$ lub $>$ 45).
4. Jeśli wartość progowa odległości czujnika od ściany $>$ 45, wówczas robot wykonuje skręt po łuku w prawo.
5. Jeśli wartość progowa odległości czujnika od ściany $\leq$ 45, wówczas robot wykonuje skręt po łuku w lewo. W ten sposób robot porusza się zygzakiem łukowym wzdłuż krawędzi ściany lub obiektu.
6. W czasie pracy na ekranie mikrokontrolera wyświetlana jest zmienna wartość odległości czujnika od ściany, w zależności od kierunku skrętu robota.
7. Praca robota wykonywana jest w trybie ciągłym, ponieważ program działa w pętli. Zatrzymanie pracy robota odbywa się z poziomu programu na komputerze, lub po wciśnięciu przycisku „Back” (Wstecz) na kostce mikrokontrolera.

The screenshot shows the LEGO Mindstorms software interface. The main workspace contains a sequence of blocks: a 'Start' block, followed by 'Measure - Proximity', 'Measure - Ultrasonic', 'Measure - Gyro', 'Measure - Acceleration', 'Measure - Rotation', and 'Measure - Color'. A 'Loop' block is connected to the 'Measure - Proximity' block, containing an equation block with the formula $(a-b)*2$ and a 'B+C' block. The 'Measure - Proximity' block is highlighted with a red box. Below the main workspace, there is a row of color-coded tabs (green, yellow, red, blue, cyan) and a row of icons representing different sensors and actuators.

Robot wyposażony jest w czujnik podczerwieni zamontowany z prawej strony, pionowo, pod kątem 45 stopni (jak na zdjęciu powyżej). Czujnik podłączony jest przewodem do gniazda nr 4 mikrokontrolera.

1. Robot uruchamia program. Ustawia kolor lampy mikrokontrolera na pomarańczowy pulsacyjny.
2. Prędkość obrotów silników kół robota ustawiona jest na 30 %.
3. W programie zastosowana jest funkcja matematyczna Equation: $(a-b)*2$ która oblicza wartość „Steering” dla zbliżenia się czujnika do ściany obiektu. Zmienna „a” ustalona jest automatycznie przez program, natomiast zmienna „b” ustalana jest przez użytkownika i w tym przypadku jej wartość wynosi 50. Jest to liczba dla najbardziej optymalnej odległości „Steering” robota w tym układzie konstrukcyjnym. Przy zmniejszaniu wartości „b” pojazd skraca odległość czujnika od ściany. Podczas okrążania obiektu skręty są bardziej ostre i robot może kołami zawadzać o krawędzie. W miarę zwiększania wartości „b” powyżej 50, robot wykonuje bardziej oddalone i bardziej łagodne skręty wokół ścian obiektu.
4. W tym przypadku, przy wartości zmiennej „b” ≈ 50 robot porusza się w linii prostej lub łagodnym łukiem wzdłuż krawędzi ściany lub obiektu. Wokół obiektu okrągłego porusza się łagodnym torem okrężnym.
5. Praca robota wykonywana jest w trybie ciągłym, ponieważ program działa w pętli. Zatrzymanie pracy robota odbywa się z poziomu programu na komputerze, lub po wciśnięciu przycisku „Back” (Wstecz) na kostce mikrokontrolera.

Zamieszczone powyżej przykłady programowania robota Mindstorms EV3 dotyczą prostych instrukcji jezdnych dla robota, niektóre z wykorzystaniem czujników koloru, dotyku, odległości i podczerwieni. Instrukcje 8 i 9 dotyczą budowy prostych operatorów matematycznych, z wynikami generowanymi na ekranie mikrokontrolera robota. Z doświadczenia można powiedzieć, że instrukcje te stanowią doskonały punkt wyjścia do rozbudowy przez uczniów, np. dodawania modułów działania w pętli lub generowania sygnałów dźwiękowych i świetlnych podczas pracy.

Więcej materiałów do zajęć z robotyki znajdziecie Państwo pod adresem internetowym:

<https://machowczyk.blogspot.com/p/robotyka-w-szkole.html>

Prowadząc zajęcia z robotyki, warto wyposażyć swój warsztat pracy o **fachową literaturę**. Niewątpliwie do zajęć proponowanych w niniejszym opracowaniu, jednym z najciekawszych i najbardziej pomocnych jest podręcznik autorstwa Henryka Brandysa, pod tytułem LEGO Mindstorms EV3 – podstawy programowania, wyd. Helion. Autor w bardzo przystępny sposób przedstawia w swojej książce podstawowe rozwiązania programistyczne, charakterystyczne dla środowiska LAB EV3. Polecam zaopatrzenie swojego warsztatu pracy w ten podręcznik do wykorzystania przez uczniów w szkole, chociaż dodam, że jest to także doskonałe źródło wiedzy także dla nauczyciela prowadzącego zajęcia z robotyki.