

Poradnik dla nauczycieli
ZESTAW ROBO MAKER PRO





Copyright © Clementoni Polska 2018

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione.



Spis treści

Strategia edukacyjna	5
Nowy sposób myślenia	7
Zajęcia	10
Metoda prób i błędów	12
Praktyczne porady	13
Słownik podstawowych pojęć	15
Scenariusz 1	18
Scenariusz 2	25
Scenariusz 3	31
Scenariusz 4	40
Scenariusz 5	45
Scenariusz 6	48



Nowe technologie, wykorzystujące technikę cyfrową, są już obecne w wielu obszarach życia codziennego. Otwierają się coraz nowsze dziedziny zastosowania szeroko rozumianych komputerów, robotów. Domowy sprzęt powszechnego użytku oparty jest w mniejszym lub większym stopniu o technologie cyfrowe. Dlatego współczesna szkoła powinna w sposób szczególny przygotować młodego człowieka do życia w zmieniającym się, technologicznym świecie. Dzisiejsi uczniowie szkoły podstawowej wejdą w dorosłe życie w momencie, gdy otaczająca ich rzeczywistość będzie permanentnie skomputeryzowana i zautomatyzowana. Nie chcemy, aby w dorosłym życiu stali się oni biernymi użytkownikami nowych technologii, podwładnymi świata robotów i automatów. Pragniemy, aby Nasze dzieci w dorosłym życiu potrafiły skutecznie i logicznie wykorzystywać nowe technologie, aby świat nasycony komputerami i robotami nie był dla nich obcy i niedostępny. Wymaga to podjęcia odpowiednich działań edukacyjnych. Drugi etap edukacyjny (klasy IV-VIII) powinien ugruntować wcześniej zdobyte umiejętności okołoprogramistyczne i rozszerzyć je na bardziej zaawansowane umiejętności algorytmicznego myślenia, znajomości podstaw programowania.

Robotyka edukacyjna daje niezwykle możliwości skutecznego wyrabiania umiejętności algorytmicznego myślenia, umiejętności podstaw programowania, a wszystko to przy zastosowaniu założeń myślenia komputacyjnego. Robotyka edukacyjna to również wprowadzanie młodego człowieka w pasjonujący świat techniki, rozwijający poza podstawowymi umiejętnościami programistycznymi również istotne umiejętności techniczne i konstrukcyjne, wyrabiając przy tym kreatywność i wytrwałość. Celem jest przyzwyczajanie młodego człowieka do obcowania z nowoczesną technologią w taki sposób, aby rozumiał ją krytycznie i miał na nią wpływ. Ideą jest wyrobienie adekwatnych do świata technologicznego sposobów myślenia i rozwiązywania problemów. Ma to również odzwierciedlenie w podstawie programowej dla szkoły podstawowej.

Wykorzystanie zestawu RoboMaker Pro, jak również przedstawione w niniejszym poradniku treści oraz przykładowe scenariusze zajęć są zgodne z wymogami zawartymi w podstawie programowej - Rozporządzenie Ministra Edukacji z dnia 24 lutego 2017 (Dz.U. 2017 poz. 356).

Zestaw RoboMaker Pro – Edukacyjne laboratorium robotyki nadaj się do zastosowań edukacyjnych uczniów klas od IV wzwyż (szkoła podstawowa).

Treść poradnika i scenariuszy uwzględnia najnowsze trendy edukacyjne.



Strategia edukacyjna

Dzisiejszy system edukacji jest tzw. systemem transmisyjnym, opartym o behawiorystyczne strategie edukacyjne. Z uwagi na nowe doświadczenia, szczególnie związane z rozwojem nowych technologii, wzrostem dostępu do informacji, koniecznością większej aktywności kreatywnej, niezbędne jest wprowadzenie zmodyfikowanego podejścia do ucznia jako osoby zdobywającej nową wiedzę i umiejętności, adekwatne do dynamicznie zmieniającej się rzeczywistości. Odpowiedzią na nowe potrzeby edukacyjne jest zmiana strategii edukacyjnej i wprowadzenie elementów wynikających z konstruktywizmu oraz z jego nurtu – konstrukcjonizmu.

Podstawowym założeniem konstruktywizmu jest potraktowanie ucznia jako osoby aktywnej będącej twórcą swojej własnej wiedzy. Wiedzy nie można przekazać, uczeń sam musi swoją wiedzę wybudować. Ważna jest aktywność ucznia w procesie budowania własnej wiedzy. Konstruktywizm wskazuje, że uczeń to nie puste naczynie, które wypełniane jest przez edukatora (rodzica, nauczyciela, wychowawcę), ale osoba aktywnie konstruująca swoją wiedzę.

Konstrukcjonizm zakłada, że w procesie uczenia się, uczniowie aktywnie angażują się w tworzenie własnych przedmiotów, zdarzeń, idei, pomysłów, którymi w ramach przyswajanych treści mogą podzielić się z innymi w celu wspólnej analizy i refleksji.

Saymour Papert (matematyk i informatyk, autor kultowego języka programowania dla dzieci – LOGO) przedstawił podejście konstrukcjonistyczne w postaci ośmiu wielkich idei, które doskonale wpisują się w nowoczesną edukację związaną z wykorzystaniem nowych technologii, a w tym **robotów edukacyjnych**.

1. **Uczenie się przez tworzenie.** Uczniowie uczą się lepiej, gdy uczenie się jest elementem uprawiania czegoś, co ich prawdziwie interesuje. Wykorzystanie **robotów edukacyjnych** generuje w uczących się większe zaangażowanie i zainteresowanie.
2. **Technologia jako tworzywo.** Dysponując technologią uczniowie mogą tworzyć znacznie więcej, szczególnie w obszarze tworzenia idei i generowania pomysłów.
3. **Ostra zabawa.** Uczymy się najlepiej, gdy to nas cieszy. Najwięcej satysfakcji daje tzw. ostra zabawa (*hard fun*), czyli stan, gdy uczeń zafascynowany jest przedmiotem zabawy, jest zafascynowany tą zabawą. Wtedy trud wysiłku umysłowego jest dla ucznia niezauważalny. Uczącego się nie trzeba dodatkowo motywować do zaangażowania. **Robotyka edukacyjna** sprzyja tworzeniu takich sytuacji.
4. **Uczenia się jak się uczyć.** Nikt nie jest w stanie nauczyć cię tego wszystkiego, co musisz umieć. Zabawa wykorzystująca nowe technologie w tym **roboty edukacyjne** wyrabia w uczących się chęć dochodzenia do rozwiązania, sprzyja zaangażowaniu i pragnieniu zgłębiania problemu.



5. **Daj sobie czas.** Należy tę ideę rozumieć następująco: każdy uczeń jest indywidualnością, każdy potrzebuje mieć „swoją” czas na rozwiązanie problemu. Powinniśmy uczyć współpracy w grupie, ale też powinniśmy dać uczniom indywidualny czas, indywidualne tempo rozwiązywania problemu. Czas realizacji zajęć wskazany w scenariuszach jest sugestią, nauczyciel sam, znając swoich uczniów powinien dopasować tempo realizacji zajęć do możliwości uczniów. Bardzo istotnym jest obserwowanie zaangażowania i skupienia się uczniów w celu zakończenia lub wydłużenia czasu zabawy i nauki z wykorzystaniem **RoboMaker Pro**.
6. **Nie ma sukcesu bez niepowodzeń.** Nic naprawdę ważnego nie działa od razu dobrze. Jedyną drogą do sukcesu jest staranne analizowanie, co i dlaczego nie funkcjonuje prawidłowo. By odnieść sukces, trzeba uwolnić się od strachu przed błędami. Szczególnie **robotyka edukacyjna** obfituje w porażki – za pierwszym razem zapewne żaden zestaw poleceń nie będzie prawidłowy, trzeba kilku prób. Uczeń oswaja się z poczuciem, że nie wszystko od razu się uda, wyrabia w sobie wytrwałość niezbędną do osiągania sukcesów w dorosłym życiu.
7. **Praktykuj sam, co zalecasz uczniom.** Najpierw Ty przećwicz funkcjonalności **RoboMaker Pro**, niech najpierw Tobie zdarzą się sytuacje nietypowe i niepowodzenia.
8. **Najważniejszym celem jest używanie technologii TERAZ do uczenia się innych rzeczy.** Zestaw **RoboMaker** jest tego doskonałym przykładem – TERAZ, na każdym etapie edukacyjnym. Rozwój technologii jest tak szybki, że z edukacją w tym zakresie nie można czekać, nie można odkładać na potem.

Robotyka edukacyjna sama w sobie jest nierozzerwalnie związana z technologią, jak również stymuluje twórcze myślenie i konieczność poszukiwania optymalnych rozwiązań. Robotyka w nauczaniu to również realizacja koncepcji nauczania myślenia algorytmicznego i myślenia komputacyjnego, a RoboMaker Pro doskonale się do tego nadaje szczególnie w nauczaniu na drugim etapie edukacyjnym.



Nowy sposób myślenia

Już ponad 20 lat temu Seymour Papert pisał o myśleniu komputacyjnym, znacznie wyprzedzając swoimi ideami ówczesne możliwości technologii. W tamtych czasach uczenie w szkole podstawowej podstaw programowania i algorytmiki było czymś wyjątkowym nawet w skali światowej. W roku 2006 Janette Wing (dyrektor Avaneessians w Data Sciences Institute na Columbia University, gdzie jest także profesorem informatyki. Była m. innymi wiceprezesem firmy Microsoft Research) określiła myślenie komputacyjne jako „**Użyteczne postawy i umiejętności, jakie każdy, nie tylko informatyk powinien starać się wykształcić i stosować**” i przyjęła, że:

Myślenie komputacyjne to procesy myślowe angażowane w formułowanie problemu i przedstawianie jego rozwiązania w taki sposób, aby komputer – człowiek lub maszyna – mógł skutecznie je wykonać.

Wing uznała, że myślenie komputacyjne, powinno towarzyszyć cały czas procesowi formułowania i rozwiązywania problemu, stanowi naturalne poszerzenie kompetencji o umiejętności stosowania metod pochodzących z informatyki i analitycznego myślenia przy rozwiązywaniu problemów pochodzących z różnych dziedzin.

W najprostszym ujęciu myślenie komputacyjne można przedstawić następująco:

1. Zdefiniowanie problemu.
2. Określenie danych wejściowych, rozpoznawanie wzorców.
3. Dekompozycja (podział na mniejsze, łatwiejsze do rozwiązania części) lub/i redukcja problemu, poprzedzone abstrakcją (wyłonienie, modelowanie najważniejszych cech sytuacji problemowej).
4. Tworzenie rozwiązania w postaci algorytmicznej.
5. Realizacja rozwiązania (np. program komputerowy – rozwiązanie problemu realizuje maszyna, ale również np. instrukcja, sposób, przepis postępowania – rozwiązanie realizuje człowiek).
6. Testowanie i modyfikacje rozwiązania.

Ucząc umiejętności myślenia komputacyjnego nauczyciel powinien zwrócić uwagę na wybrane aspekty pracy uczniów, kształcone kompetencje uczniów są często dla nich nieświadome, ale powinny być świadome dla nauczyciela. Rolą nauczyciela jest więc świadome wyłonienie w pracy ucznia elementów myślenia komputacyjnego, ukierunkowanie działań edukacyjnych tak, aby procesy rozwiązywania problemu (zadania) w miarę możliwości uczyły i ćwiczyły myślenie komputacyjne.



Dla przykładu - analizując załączony **scenariusz 3** proces myślenia komputacyjnego można przedstawić następująco:

1. **Sytuacja:** na dowolnej powierzchni znajduje się robot, który potrafi przemieszczać się w przód i w tył, wykonać zwrot, rozpoznać obecność przedmiotu znajdującego się przed jego przodem.
2. **Zadanie (problem):** robot porusza się zgodnie z opisem zawartym w scenariuszu.
3. **Dane (abstrakcja):** przedmiot przed robotem, ile razy przedmiot został wykryty przez czujnik, zwrot – jak rozpoznać przedmiot, jak policzyć ilość reakcji robota na przedmiot, jak wykonać zwrot (jaki musi upłynąć czas gdy silniki są włączone, w jaką stronę mają się obracać gąsienice robota).
4. **Zasada w danych (+abstrakcja):** przedmiot przed robotem – czujnik IR aktywny, policzenie wykonanych reakcji robota – wykonanie konkretnej ilości razy instrukcji wewnątrz pętli, wykonanie zwrotu.
5. **Dekompozycja:** 1. robot cofa się przed, jeśli czujnik wykryje przedmiot (obiekt); 2. obrót robota, 3. przemieszczanie się do przodu do momentu napotkania przeszkody.
6. **Algorytm:** Zestaw poleceń realizujących podproblemy 1, 2 i 3. Scalenie poprzez realizację pętli.
7. **Komputer, maszyna – program:** programowanie robota – automatyzacja sytuacji wyabstrahowanej.
8. **Testowanie i modyfikacja:** sprawdzenie poprawności rozwiązania, poszukiwanie lepszych rozwiązań.

Realizując zajęcia z wykorzystaniem nowych technologii, a w tym robotów edukacyjnych nauczyciel powinien zwracać szczególną uwagę na przebieg procesu myślowego swoich podopiecznych, ukierunkowując go i stymulując tak, aby wykształcić umiejętności organizowania procesu rozwiązywania problemu zgodnie z zasadami myślenia komputacyjnego.

Myślenie komputacyjne jest istotnym elementem umożliwiającym stosowanie narzędzi i metod wywodzących się z informatyki w celu zrozumienia i rozwiązywania problemów związanych zarówno z procesami naturalnymi, jak i sztucznymi. Ponadto ten rodzaj myślenia, strategia rozwiązania problemu staje się wręcz niezbędna w świecie wypełnionym nowymi technologiami.

W kreowaniu umiejętności komputacyjnego myślenia (jeśli uczeń do tej pory nie miał styczności z rozwiązywaniem problemów poprzez myślenie komputacyjne) lub w ich doskonaleniu i rozwijaniu, szczególne znaczenie ma zastosowanie robotów edukacyjnych.

Zastosowanie robotów edukacyjnych w znacznym stopniu przyczynia się do lepszego poznania algorytmiki i zasad programowania. Wykonywane przez robota zadania są



rzeczywistym odzwierciedleniem algorytmu – tworu czysto abstrakcyjnego. Zastosowanie robotów edukacyjnych ułatwia uczniom korelację abstrakcji, idei z ich fizyczną realizacją. Robot fizycznie odzwierciedlający realizowany zestaw poleceń jest wręcz idealnym rozwiązaniem w kształtowaniu zasad myślenia komputacyjnego i algorytmicznego.



Zajęcia

Przed rozpoczęciem cyklu zajęć należy zapoznać się dokładnie z zestawem RoboMaker Pro oraz z dedykowaną zestawowi aplikacją, należy dokładnie przeczytać załączoną do zestawu instrukcję. Robotyka edukacyjna wymaga od nauczyciela dokładnego przeanalizowania możliwości sprzętu i oprogramowania. Nie jest to zadanie łatwe szczególnie dla nauczycieli rozpoczynających swoją przygodę z tym tematem. Niezbędna jest choćby minimalna wiedza praktyczna, którą można zdobyć jedynie samemu próbując budować własne konstrukcje i programy. Traktując ten temat jako pewnego rodzaju rozrywkę i dobrą zabawę, można jednak w przyjemny sposób przebrnąć przez tajniki robotyki edukacyjnej. Zestaw RoboMaker Pro nie będzie sprawiał problemu nauczycielom, którzy mieli już styczność z robotami edukacyjnymi.

Zajęcia z zestawem RoboMaker Pro mogą stanowić uzupełnienie lekcji informatyki lub mogą być przeprowadzane jako zajęcia dodatkowe (kółko). Aplikacja RoboMaker zawiera instrukcje budowania kilku robotów, jak również, po wybraniu trybu **Ucz się**, podpowiada programującemu kolejne bloki poleceń, jakie należy użyć tworząc program. Przed przystąpieniem do korzystania z gotowych instrukcji i programów należy zapoznać uczniów z podstawowymi zasadami programowania robotów RoboMaker Pro. Zanim uczniowie zaczną tworzyć programy w oparciu o przykłady zamieszczone w aplikacji, powinni zapoznać się z poszczególnymi blokami poleceń używanych w programowaniu RoboMaker Pro.

Cykl zajęć związanych z programowaniem robotów powinien zacząć się od wprowadzenia w temat robotyki. Następnie należy przeprowadzić zajęcia zgodnie z pierwszymi trzema scenariuszami załączonymi do poradnika. Są to przykłady scenariuszy zajęć, których celem jest wprowadzenie ucznia w tematykę programowania robotów i w podstawy algorytmiki. Tym zagadnieniom nauczyciel powinien poświęcić szczególną uwagę, bowiem wiedza i umiejętności zdobyte podczas tych zajęć będą stanowiły fundament do dalszej pracy. Po zrealizowaniu zajęć, na których uczniowie zapoznają się z zasadami tworzenia algorytmów i programów, poznają podstawowe konstrukcje algorytmiczne, poznają sposób działania elementów zestawu RoboMaker Pro, można przejść do dalszego zgłębiania tematu.

W zależności od wieku uczniów, w zależności od ich predyspozycji i zdolności, niektóre, trudniejsze treści przykładowych scenariuszy można pominąć. Również zdolności uczniów, ich zainteresowanie robotyką, przesądzą o charakterze dalszych zajęć. Nauczyciel ma do wyboru dwie główne strategie dalszego edukowania.

Pierwsza strategia - to bazowanie na robotach budowanych zgodnie z instrukcjami zamieszczonymi w aplikacji RoboMaker oraz wykorzystanie programów tworzonych w trybie **Ucz się**. Podczas programowania zgodnie z zawartymi w aplikacji podpowiedziami należy



zwrócić uwagę, aby uczniowie nie tworzyli programów mechanicznie przeciągając bloki poleceń, tylko aby ze zrozumieniem czytali pojawiające się wskazówki. Po zaprogramowaniu robota należy przetestować działanie programu oraz omówić zasadność zastosowania poszczególnych bloków poleceń. Uczniowie mogą próbować modyfikować program sprawdzając efekty swoich modyfikacji. Nauczyciel powinien jednak zwrócić uwagę, aby modyfikacje nie były bezmyślne i pozbawione sensu. Ta strategia jest wskazana dla uczniów młodszych oraz dla uczniów nie posiadających odpowiednich predyspozycji i zdolności.

Druga strategia – to budowanie własnych kreatywnych konstrukcji oraz próba kreatywnego modyfikowania programów znajdujących się w aplikacji RoboMaker. W przypadku tej strategii należy zwrócić szczególną uwagę na dogłębne opanowanie przez uczniów materiału zawartego w trzech pierwszych scenariuszach. Czwarty, dołączony do poradnika scenariusz jest przykładem mini pracy badawczej, natomiast piąty i szósty powinny być inspiracją do wymyślania własnych scenariuszy opartych o typowe konstrukcje i programy dostępne w dedykowanej aplikacji.

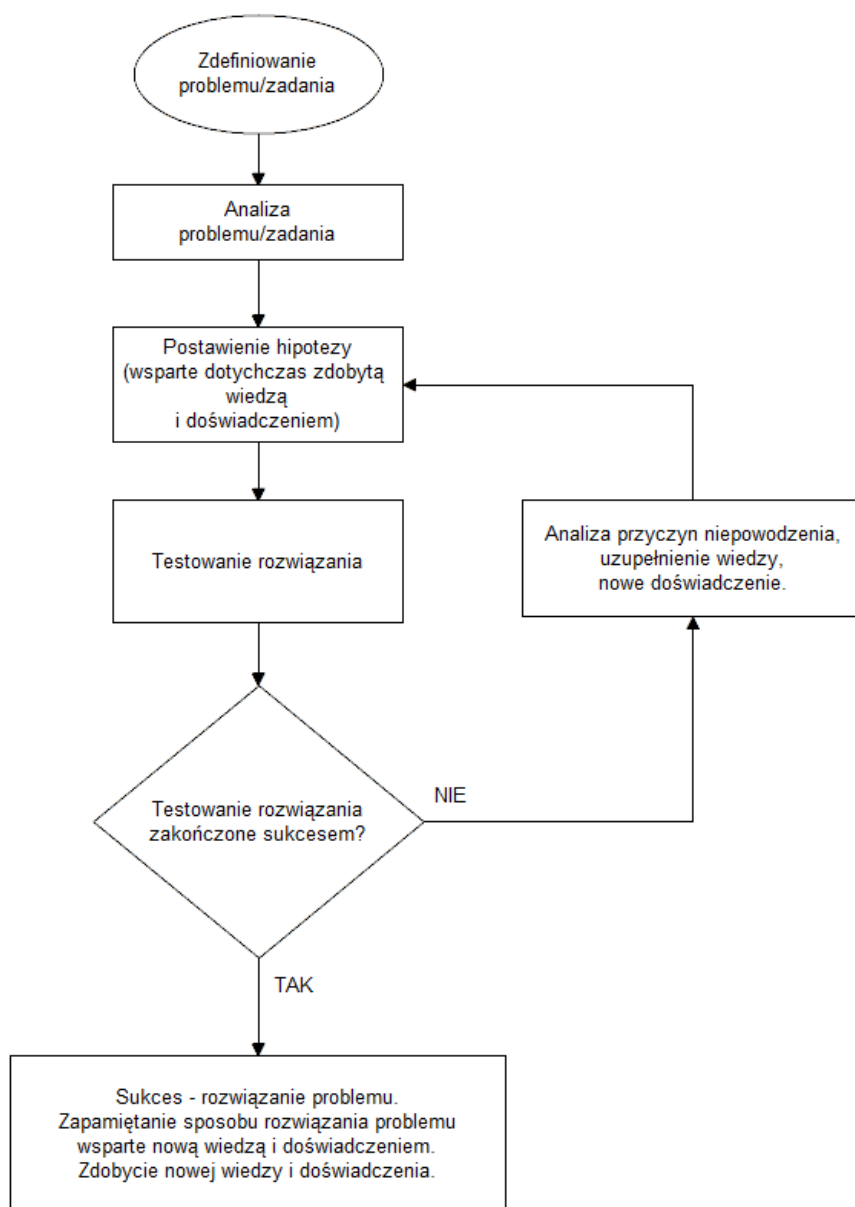
Uwagi do scenariuszy.

Zestaw RoboMaker Pro może być stosowany na zajęciach z uczniami klas od IV do VIII (szkoła podstawowa). Podczas pracy z uczniami klasy IV i V należy więcej czasu poświęcić na zajęcia wstępne. Dla uczniów tych klas rysowanie algorytmów będzie sprawiało trudność, wynikającą zarówno z braku zdolności manualnych, jak również z braku umiejętności abstrakcyjnego wyobrażania sobie sposobu rozwiązania zadania. Mimo tych problemów należy starać się, aby u uczniów wyrobić umiejętność algorytmicznego myślenia i umiejętność zapisu rozwiązań w sposób algorytmiczny. Bloki na schemacie algorytmu uczniowie mogą wykonać w formie dowolnych rysunków lub znaków graficznych, które kojarzą im się z wykonywaną czynnością. Szczególną uwagę należy poświęcić blokowi warunkowemu. Przed przystąpieniem do tworzenia programów z wykorzystaniem warunków wskazane jest przećwiczenie tego typu konstrukcji algorytmicznych na przykładach z życia codziennego – np. gotowanie jajka (... 1. woda z jajkiem gotuje się; 2. patrzę na zegarek; 3. zapamiętuję czas, 4. po chwili patrzę na zegarek; 5. Sprawdzam czy minęły 3 minuty - jeśli nie - wykonuję punkt 4, jeśli tak - wyjmuję jajko z wody). Czas realizacji zajęć, podany w scenariuszach, jest orientacyjny. W zależności od zdolności i zaangażowania uczniów może on ulec zmianie. Nauczyciel powinien dobierać czas na zrealizowanie poszczególnych treści w zależności od postępów uczniów. Załączone scenariusze nie obejmują całego obszaru możliwych zastosowań zestawu RoboMaker Pro. Scenariusze powinny być inspiracją dla nauczycieli do kreatywnego tworzenia własnych zajęć.



Metoda prób i błędów.

Metoda prób i błędów jest jedną z podstawowych metod rozwiązywania problemów. Metoda ta często wykorzystywana jest na zajęciach z robotyki. Należy jednak zwrócić uwagę na odpowiednie jej stosowanie. Dzisiejsi uczniowie, otoczeni technologią, często bezwiednie i intuicyjnie stosują tę metodę. Użytkowanie nowych technologii przyzwyczajają nas do szybkiej wręcz natychmiastowej odpowiedzi na nasze działania. Np. uczeń próbujący „rozgryźć” nową grę komputerową, bardzo rzadko sięgnie do opisu gry, natomiast zdecydowanie częściej zastosuje metodę prób i błędów. Natychmiastowa reakcja technologii (np. gry komputerowej) umożliwia zastosowanie metody prób i błędów wielokrotnie w stosunkowo krótkim czasie. To prowadzi do utrwalania niewłaściwego nawyku: *Testuj jak najwięcej sposobów, w końcu któryś okaże się dobry. Zapamiętaj dobry sposób*. Takie podejście nie buduje żadnej wiedzy związanej z poszukiwaniem rozwiązania, nie buduje żadnych refleksji, nie sprzyja analizie dlaczego rozwiązanie jest takie, a nie inne. Również, podczas zajęć z wykorzystaniem robotów edukacyjnych, uczniowie modyfikują programy w bardzo szybkim tempie, sprawdzając jedynie efekt działania programu po kolejnych modyfikacjach. Nie towarzyszą temu istotne elementy metody prób i błędów jako metody świadomego poszukiwania rozwiązania. Brak jest działań takich jak: postawienie hipotezy, analiza złej hipotezy, wyciągnięcie wniosków, skorygowanie hipotezy i postawienie nowej hipotezy. Metoda prób i błędów będzie wtedy skuteczna (przez skuteczność należy rozumieć nie tylko znalezienie rozwiązania, ale zdobywanie wiedzy i doświadczenia w trakcie testowania), jeżeli będzie przeprowadzona w sposób właściwy:



Jedynie stosowanie metody prób i błędów zgodnie z powyższym schematem ma sens dydaktyczny. Należy o tym pamiętać podczas prowadzenie zajęć, nauczyciel w tym przypadku powinien być przewodnikiem i doradcą ingerującym w działania uczniów tylko wtedy, gdy jest konieczne dokonanie korekty ich sposobu działania.

Praktyczne porady

Elementy konstrukcyjne zestawu RoboMaker Pro są spasowane ze sobą bardzo dokładnie. Może to na początku sprawiać problemy z montażem lub demontażem robotów



(konieczność użycia większej siły podczas łączenia lub rozłączania elementów). Zbyt luźne spasowanie części mogłoby doprowadzić do szybkiego i nadmiernego „wyrobienia” się elementów łączeniowych. Jednak kilkukrotne użycie elementów spowoduje, że montaż (demontaż) stanie się łatwiejszy.

Moduł centralny, czujniki i silniki wyposażone są w złącza zapewniające dobre przyleganie styków, a tym samym przepływ sygnałów niezakłócony np. drganiem spowodowanymi działaniem robota. Bezzakłócenia przepływ sygnałów wymaga ścisłego przylegania styków, dlatego złącza wyposażone są w mini zatrask, który może czasami utrudniać ich rozłączenie. Dobrze jest mieć pod ręką mały śrubokręt i w razie konieczności delikatnie podważyć boczną krawędź złącza, co spowoduje zwolnienie zatrasku. Nie należy wyciągać wtyczki z gniazda złącza poprzez ciągnięcie za przewody.

Podczas uruchamiania programu i wykonywania poleceń w nim zawartych, robot (jego moduł centralny) jest cały czas połączony bezprzewodowo z tabletem, z uwagi na zewnętrzne zakłócenia, może to czasami spowodować błędne zadziaływanie robota. Należy wtedy wstrzymać działanie programu i uruchomić go jeszcze raz. Należy również mieć na uwadze że sprawność działania aplikacji zależy od mocy obliczeniowej (rodzaju) procesora zastosowanego w tablecie. Nie bez znaczenia jest też ilość innych aplikacji pracujących w tle. Podczas korzystania z aplikacji RoboMaker należy starać się aby niepotrzebne aplikacje działające w tle były zamknięte.

Jeśli odnosi się wrażenie, że robot nie wykonuje, albo wadliwie wykonuje program, należy dokładnie sprawdzić poprawność konstrukcji robota. Przy stosowaniu czujników IR może zdarzyć się, że z uwagi na błędy popełnione podczas budowania robota czujnik jest zamontowany niestabilnie lub skierowany jest w złym kierunku. Czasami nawet kilkustopniowe odchylenie czujnika IR od założonego kierunku może spowodować, że czujnik „łapie” światło podczerwone odbite od przypadkowych elementów. Np. nawet lekkie pochylenie czujnika IR może spowodować, że zanim robot dotrze do przeszkody, czujnik aktywuje się poprzez odbicie światła podczerwonego od podłoża. Jeśli robot nie przemieszcza się wzdłuż prostej, a z konstrukcji programu wynika, że robot powinien przemieszczać się wzdłuż prostej – należy sprawdzić, czy mocowanie kół (gąsienic) jest prawidłowe. Często zbyt mocne zaciśnięcie elementów mocujących oś, doprowadza do nadmiernego tarcia, co powoduje niewłaściwą pracę robota. Podobny problem może wystąpić, gdy przewody łączące czujniki lub silniki z modułem centralnym, ocierają się o koła zębate lub o gąsienice.

Jeśli na zajęciach wykorzystywany jest tylko jeden zestaw RoboMaker Pro należy postarać się o takie skonfigurowanie posiadanego sprzętu (tablet, rzutnik multimedialny, komputer) aby możliwe było wyświetlanie ekranu tabletu na rzutniku multimedialnym. Jeśli posiadany sprzęt nie umożliwia bezpośredniego podłączenia tabletu do rzutnika można



zastosować rozwiązania pośrednie – w Internecie dostępne są informacje na ten temat. Wykorzystując jeden zestaw RoboMaker Pro nauczyciel powinien skupić się przede wszystkim na zagadnieniach związanych z rozwijaniem umiejętności myślenia algorytmicznego. W takim przypadku uczniowie indywidualnie lub w parach pracują nad rozwiązaniem algorytmicznym lub programem (jeśli posiadana ilość tabletów na to pozwala), a jedynie testują swoje rozwiązania kolejno łącząc swój tablet z wcześniej przygotowanym robotem. Rozwiązania mogą być dyskutowane w grupie – program sterujący robotem wyświetlany jest wtedy za pomocą rzutnika multimedialnego. W przypadku, gdy szkoła nie posiada wystarczającej ilości tabletów można wykorzystać jedną z nowych strategii edukacyjnych – BYOD (ang.: *bring your own device*). Uczniowie mogą uruchomić aplikację na własnych smartfonach i indywidualnie lub w parach tworzyć swoje rozwiązania, a następnie łącząc się kolejno z robotem, mogą testować swoje rozwiązania.

W przypadku wykorzystywania większej ilości zestawów RoboMaker Pro najkorzystniejszym przypadkiem jest posiadania takiej ich ilości, aby jeden zestaw przypadał na jednego lub dwóch uczniów. Należy być świadomym tego, że samodzielna praca z jednym zestawem czterech lub więcej uczniów radykalnie obniża efekty edukacyjne. Ewentualnie należy przeprowadzić zajęcia podobnie jak w przypadku posiadania tylko jednego zestawu (większa ilość zestawów RoboMaker Pro usprawni testowanie rozwiązań wypracowanych indywidualnie).

Słownik podstawowych pojęć

Robot – urządzenie mechaniczne wykonujące automatycznie określone czynności lub zadania. Robot może być sterowany poprzez program (zestaw poleceń).

Czujnik (sensor) – urządzenie, element będący częścią większego układu służący do wychwytywania sygnałów z otoczenia. Zestaw RoboMaker Pro posiada jeden czujnik dotyku (przycisk) oraz dwa czujniki IR. Czujniki IR wysyłają światło podczerwone i reagują na jego odbicie (patrz instrukcja do zestawu). W związku z powyższym mogą pełnić rolę czujników zbliżeniowych (zbliżający się przedmiot odbija światło podczerwone) lub czujników koloru (ciemny/jasny). Czujniki IR aktywują się jeśli dociera do ich światło odbite, najczęściej wiąże się to z rozróżnieniem koloru czarnego (słabo odbijający światło) i koloru białego (dobrze odbijający). Należy jednak mieć na uwadze to, że może się zdarzyć, że wiązka światła będzie lepiej odbijana przez ciemny kolor z połyskiem niż jasny matowy. Na czujnik może również mieć wpływ wiązka światła podczerwonego generowana przez np. pilota do sterowania rzutnikiem multimedialnym, właściwość tę można również wykorzystać do uruchamiania robota dowolnym pilotem do sprzętu RTV.

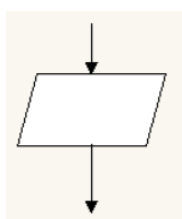


Element wykonawczy – w zestawie RoboMaker Pro elementami wykonawczymi są jedynie silniki.

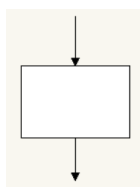
Algorytm – zestaw (ciąg) jasno zdefiniowanych poleceń, czynności, działań niezbędnych do wykonania jakiegoś działania. Algorytmy często przedstawia się za pomocą schematu blokowego. Najczęściej w schemacie blokowym algorytmu występują:



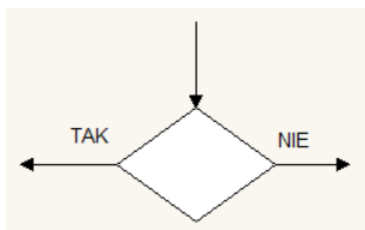
Blok graniczny – początek lub koniec.



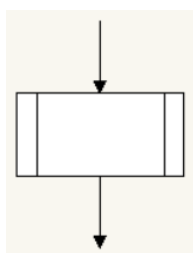
Blok wejścia/wyjścia – wprowadzenie/wyprowadzenie danych do/z programu.



Blok operacyjny – wykonanie operacji, czynności.



Blok warunkowy, decyzyjny – wybór jednego z dwóch wariantów wykonania programu w zależności od spełnienia warunku.



Blok fragmentu – część programu odrębnie zdefiniowana.

Program – sekwencja poleceń, sekwencja instrukcji koniecznych do wykonania, w celu realizacji zadania. Polecenia, instrukcje zapisane są za pomocą symboli, z zastosowaniem określonych reguł (języka programowania).



Kodowanie – w znaczeniu związanym z umiejętnościami programistycznymi – to proces zapisania algorytmu (kodowanie) w wybranym języku programowania.



Scenariusz 1.

Temat:

Wprowadzenie do programowania robotów.

Czas trwania zajęć: 2x45 minut

Cel ogólny:

Kształtowanie umiejętności tworzenia prostych programów w blokowym języku programowania.

Cele szczegółowe:

- kształtowanie umiejętności definiowania podstawowych konstrukcji algorytmicznych;
- kształtowanie umiejętności tworzenia prostego programu z wykorzystaniem czujników i silników.
- kształtowanie umiejętności poszukiwania i utrwalania wzorców.
- kształtowanie umiejętności podziału zadania na części i odniesienia poszczególnych elementów programu do zapisu algorytmu w postaci schematu blokowego;

Cele operacyjne:

- uczeń zna funkcjonalności aplikacji RoboMaker;
- uczeń wie co to jest sekwencja (kolejność) poleceń;
- uczeń potrafi ułożyć sekwencję poleceń sterującą elementami robota;
- uczeń potrafi ułożyć sekwencję poleceń wykorzystującą czujniki robota;
- uczeń zna podstawowe bloki języka programowania RoboMaker;
- uczeń zna zasady rysowania schematu blokowego algorytmu;
- uczeń potrafi narysować schematy blokowe prostych algorytmów.

Formy pracy:

- zespołowa;
- w grupach, w parach;
- indywidualna.

Metody pracy:

- asymilacja wiedzy;
- samodzielnego dochodzenia do wiedzy (problemowa);

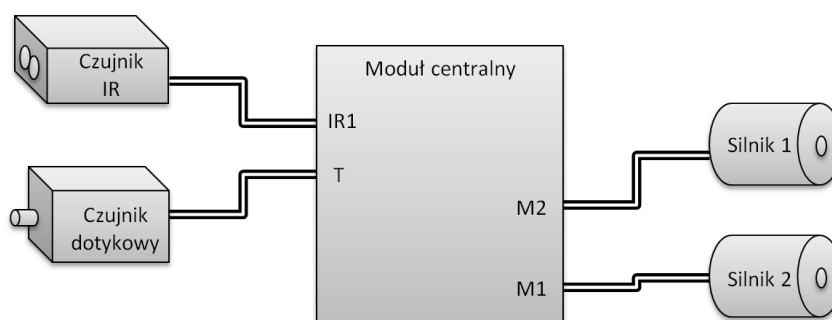


- praktyczna (ćwiczebna).

Środki dydaktyczne:

- zestaw RoboMaker Pro;
- tablet z zainstalowaną aplikacją RoboMaker, tryb: **Twórz – Konstrukcja własna**.

Poproś uczniów aby zbudowali urządzenie składającą się z dwóch silników, dwóch czujników IR i czujnika dotykowego. Urządzenie ma służyć tylko do zapoznania się z podstawami programowania robotów zbudowanych z klocków RoboMaker Pro. Wystarczy, że na podłużnych klockach zostaną stabilnie zamocowane silniki i czujniki tak aby elementy te mogły realizować swoje funkcje. Silniki i czujniki należy podłączyć do modułu centralnego:



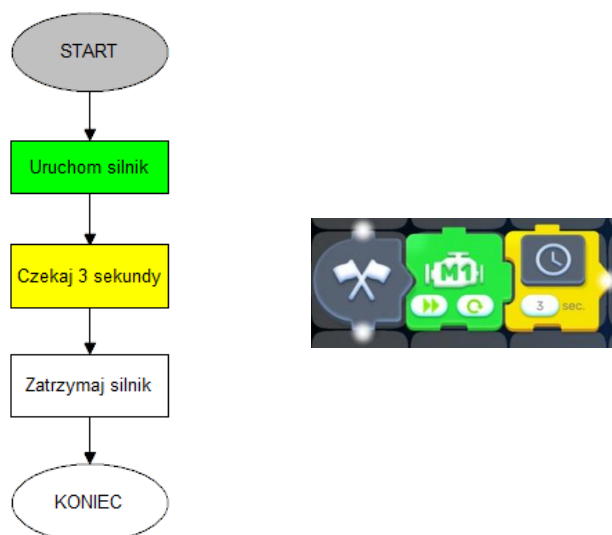
Pierwsze ćwiczenia będą wykonywane z wykorzystaniem trybu *Twórz – Konstrukcja własna* (aplikacja RoboMaker).

Po uruchomieniu aplikacji i wybraniu właściwego trybu pracy omów funkcjonalności aplikacji (sposób ustawiania bloków, usuwanie bloków). Zwróć szczególną uwagę na to, że nie wszystkie bloki pasują do siebie, bloki muszą być łączone wg konkretnych zasad.

Przy omawianiu nawet najprostszych zestawów poleceń (programów), należy je również przedstawiać w formie algorytmu (schematu blokowego).

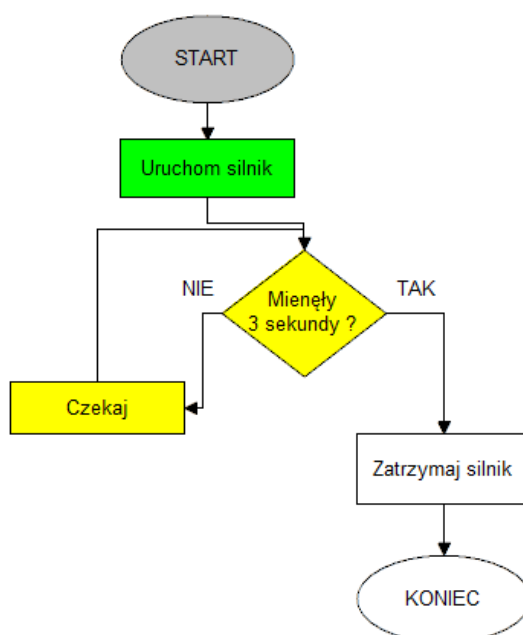
Pokaż uczniom sposób i zasady rysowania schematu blokowego algorytmu. Omów, czym jest algorytm.

Zaproponuj uczniom ułożenie banalnego programu – powodującego włączenie silnika oraz jego działanie przez 3 sekundy. Pokaż jak wygląda schemat blokowy algorytmu i jak wygląda odpowiadający mu program.



Zwróć uwagę, na to, że w programie nie ma bloku *Zatrzymaj silnik*. Blok *Zatrzymaj silnik* występuje natomiast w schemacie blokowym algorytmu. Często tak się zdarza, że polecenia (czynności) występujące w schemacie blokowym algorytmu nie będą miały bezpośredniego odpowiednika w programie, jednak zostaną wykonane – są jakby „ukryte” w innych blokach programu.

Pokaż uczniom inną postać algorytmu odpowiadającą wcześniej przedstawionemu programowi.



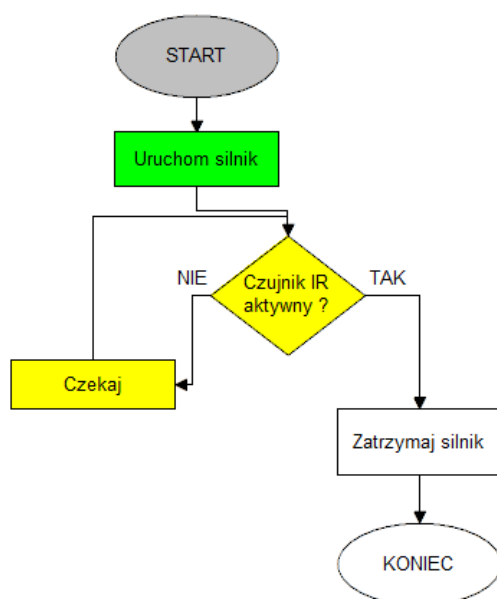


Sens zastosowania powyższego algorytmu stanie się zrozumiały po przeprowadzeniu kolejnych prób z programowaniem zbudowanego urządzenia.

Zadanie dla uczniów:

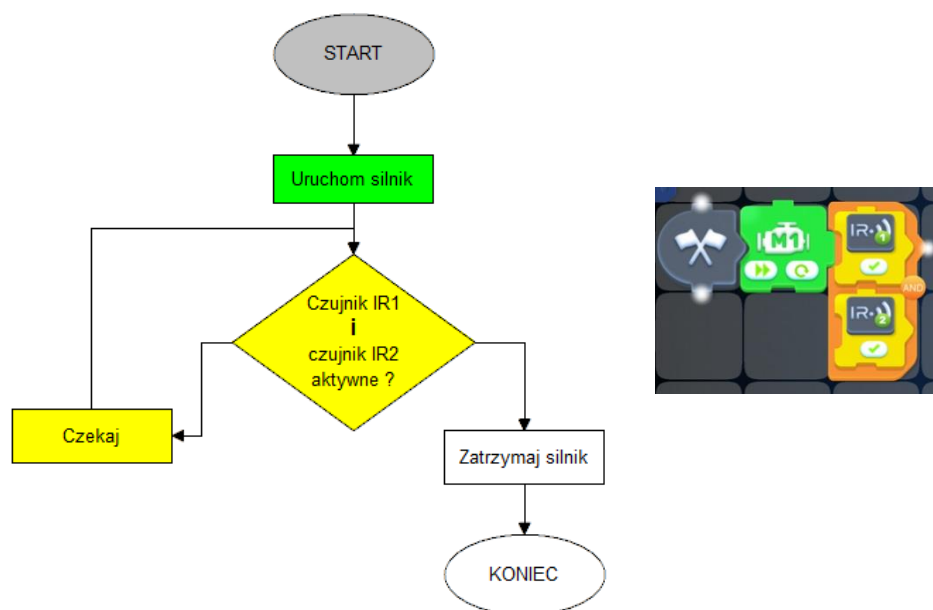
Należy samodzielnie zapoznać się z pozostałymi opcjami żółtego bloku (zatrzymanie silnika za pomocą przycisku lub czujnika IR).

Po kilku eksperymentach poproś, aby uczniowie samodzielnie narysowali algorytmy odpowiadające innym opcjom żółtego bloku. W przypadku zastosowania czujnika IR algorytm powinien wyglądać następująco:



Ten przykład pokazuje sens zastosowania na schemacie algorytmu bloku warunkowego, prezentującego sposób działania żółtego bloku w programie. Dlatego żółty blok programu nazywa się: *warunek prosty*.

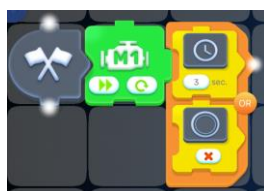
Pokaż uczniom, że można ułożyć tak program, aby silnik zatrzymał się w wyniku odpowiedniego stanu dwóch czujników. W tym celu należy użyć *warunku wielokrotnego*. Jeśli silnik miałby się obracać do momentu gdy obydwa czujniki IR będą aktywowane, algorytm i sekwencja poleceń (program) powinny wyglądać następująco:



Zadanie dla uczniów:

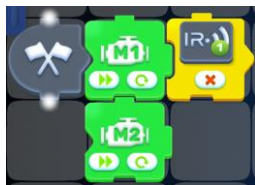
Ułóż taki program, aby jego wykonanie spowodowało uruchomienie silnika. Silnik powinien obracać się, jeśli czujnik dotyku (przycisk) będzie aktywny (naciśnięty). Silnik zatrzyma się po 3 sekundach lub wcześniej, gdy zwolnimy czujnik dotyku.

Prawidłowe rozwiązanie powinno wyglądać jak na rysunku:



Zaproponuj aby uczniowie zamienili *warunek wielokrotny OR* na *warunek wielokrotny AND*. Poproś aby spróbowali samodzielnie określić zasady działania warunków z użyciem OR i AND.

Pewne sekwencje mogą być uruchamiane w jednym momencie. Tę właściwość pokaż uczniom na przykładzie jednoczesnego uruchamiania dwóch silników:



Jeszcze raz przypomnij uczniom, że polecenia uruchamiające silnik działają w taki sposób, że wykonanie takiego polecenie spowoduje uruchomienie silnika i jego działanie dopóki nie nastąpi spełnienie warunku wskazanego w następnym poleceniu. Analogicznie działa blok oczekiwania – wykonanie tego polecenie spowoduje przejście programu w stan oczekiwania na spełnienie warunku wskazanego w następnym poleceniu.

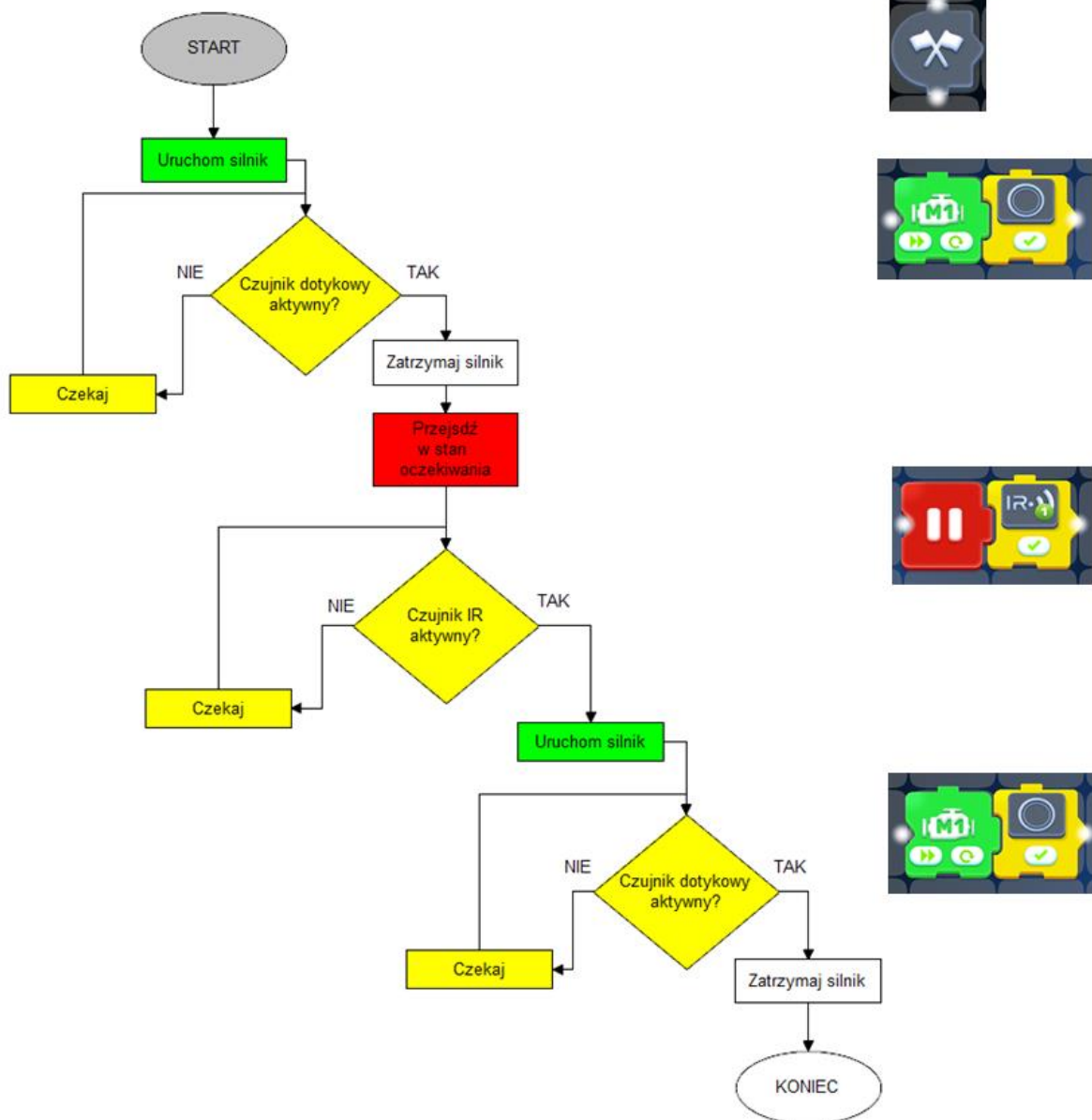
Zadanie dla uczniów:

Ułóż program realizujące następujące funkcje (przed przystąpieniem do tworzenia programu narysuj odpowiedni algorytm):

1. Po uruchomieniu programu silnik zaczyna się kręcić.
2. Silnik zatrzyma się po aktywowaniu czujnika dotykowego (przycisk).
3. Silnik ponownie uruchomi się po aktywowaniu czujnika IR.
4. Silnik będzie działał przez trzy sekundy.

Wskazówka: Po ułożeniu algorytmu, do jego poszczególnych fragmentów, dopasuj znane Ci sekwencje poleceń.

Skoryguj prace uczniów tak, aby w efekcie powstał schemat blokowy jak poniżej. Poproś aby uczniowie spróbowali wyłonić z algorytmu mniejsze części (podproblemy) – łatwiejsze do rozwiązania, takie do których można dopasować poznane wcześniej, proste zestawy poleceń (poszukiwanie wzorców, analogie).



Rozwiązanie problemu będzie polegało na złożeniu w całość, rozwiązań dla wyodrębnionych podproblemów:





Scenariusz 2

Temat:

Wprowadzenie do programowania robotów – blok *Rozdzielacz* i *Multiplekser*.

Czas trwania zajęć: 2x45 minut

Cel ogólny:

Kształtowanie umiejętności tworzenia prostych programów w blokowym języku programowania.

Cele szczegółowe:

- kształtowanie umiejętności definiowania podstawowych konstrukcji algorytmicznych z zastosowaniem klasycznej instrukcji warunkowej – blok *Rozdzielacz*;
- kształtowanie umiejętności tworzenia prostego programu z wykorzystaniem czujników i silników;
- kształtowanie umiejętności poszukiwania i utrwalania wzorców;
- kształtowanie umiejętności podziału zadania na części i odniesienia poszczególnych elementów programu do zapisu algorytmu w postaci schematu blokowego;

Cele operacyjne:

- uczeń wie co to jest sekwencja (kolejność) poleceń;
- uczeń potrafi ułożyć sekwencję poleceń sterującą elementami robota z wykorzystaniem bloku *Rozdzielacz* i *Multiplekser*;
- uczeń potrafi ułożyć sekwencję poleceń wykorzystując czujniki robota,
- uczeń zna podstawowe bloki języka programowania RoboMaker;
- uczeń zna zasady rysowania schematu blokowego algorytmu;
- uczeń potrafi narysować schematy blokowe algorytmów zawierającego warunek;

Formy pracy:

- zespołowa;
- w grupach, w parach;
- indywidualna.

Metody pracy:

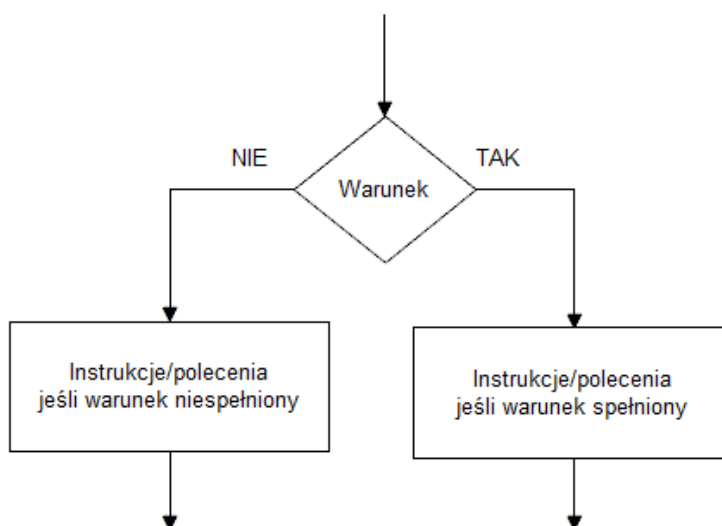


- asymilacja wiedzy;
- samodzielnego dochodzenia do wiedzy (problemowa);
- praktyczna (ćwiczebna).

Środki dydaktyczne:

- zestaw RoboMaker Pro – urządzenie zbudowane jak w scenariuszu 1;
- tablet z zainstalowaną aplikacją RoboMaker tryb: **Twórz – Konstrukcja własna**.

Zagadnienia zamieszczone w scenariuszu 1 dotyczyły podstaw programowania zestawu RoboMaker Pro. Układane programy nie zawierały klasycznej instrukcji (polecenia) warunkowego. Co prawda w algorytmie występował blok warunkowy, lecz równie dobrze mógłby być on zastąpiony działaniem prostym (porównaj pierwszy i drugi schemat blokowy algorytmu w scenariuszu 1). Klasyczną konstrukcję instrukcji warunkowej (polecenia warunkowego) prezentuje algorytm:





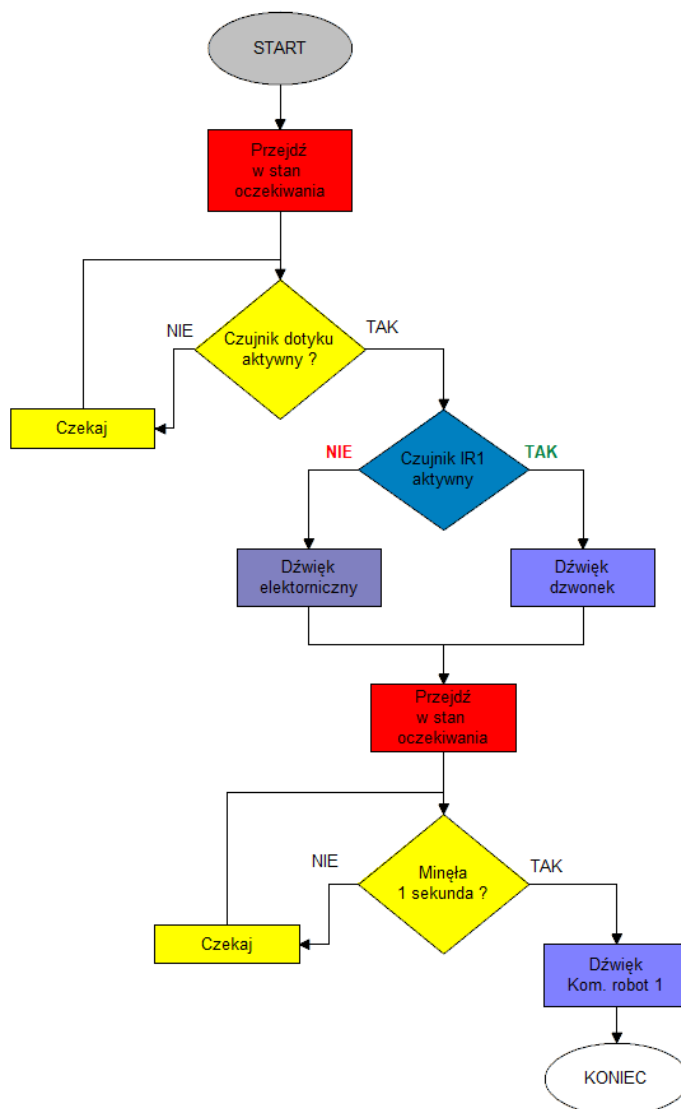
Zadanie dla uczniów:

Ułóż algorytm tak, aby opracowany na jego podstawie program realizował następujące funkcje:

1. Po uruchomieniu, program oczekuje na aktywację czujnika dotykowego. Aktywacja czujnika dotykowego powoduje przejście do wykonania kolejnych instrukcji programu.
2. Jeżeli czujnik IR jest aktywny program generuje *Dźwięk dzwonek*.
3. Następne polecenia programu są wykonywane jeśli czujnik IR przestanie być aktywny.
4. Jeżeli czujnik IR jest nieaktywny program generuje *Dźwięk elektroniczny*.
5. Następne polecenia programu są wykonywane jeśli czujnik IR będzie aktywny.
6. Po wykonaniu powyższych instrukcji program czeka 1 sekundę i generuje *Dźwięk kom. Robot 1*.

Wskazówka: Podziel zadanie na mniejsze, łatwiejsze do wykonania części.

Rozwiązanie zadania (algorytm) może wyglądać następująco:



Poproś aby uczniowie sami spróbowali ułożyć program odpowiadający pierwszej i ostatniej części algorytmu. Następnie wyjaśnij sposób stosowania bloku *Rozdzielacz* oraz *Multiplexer*. Ewentualnie naprowadź uczniów na poprawne rozwiązanie środkowego fragmentu algorytmu. Następnie poproś aby wszystkie elementy programu złożyli w całość:





Teraz, gdy już znany jest sposób wykonania polecenia warunkowego (blok *Rozdzielacz*), uczniowie mogą spróbować rozwiązać trudniejsze zadanie.

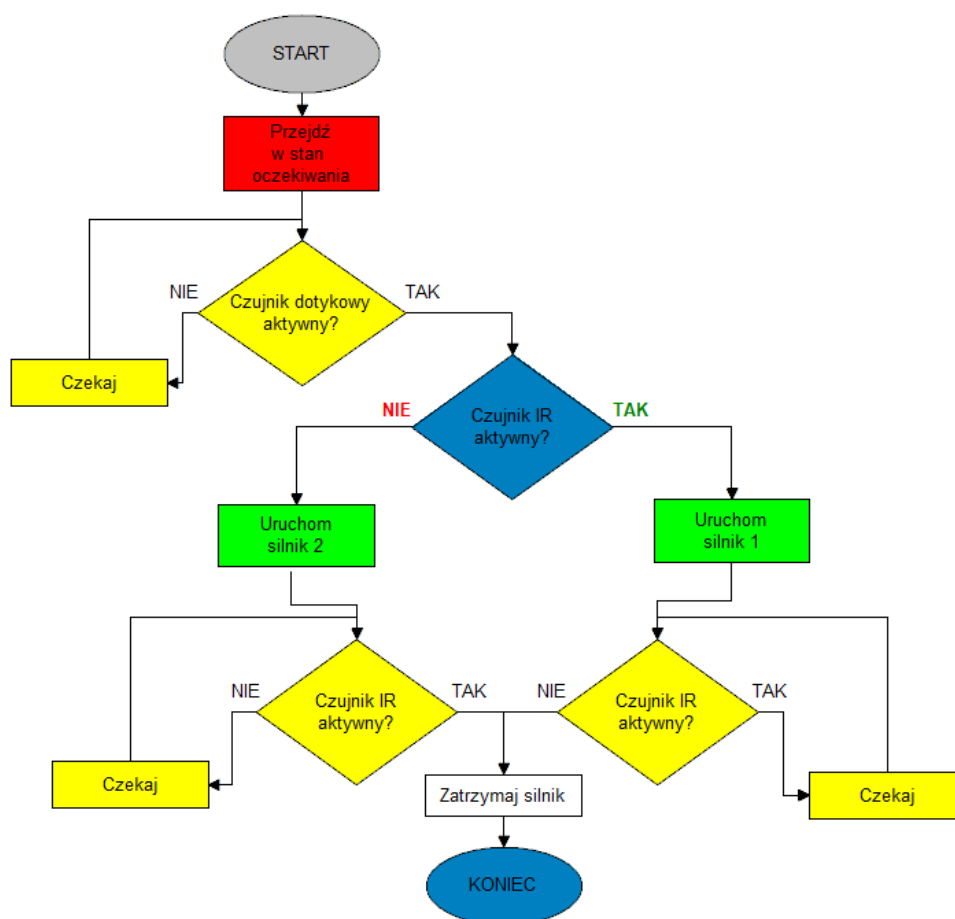
Zadanie dla uczniów:

Ułóż program realizujący następujące funkcje:

1. Po uruchomieniu, program oczekuje na aktywację czujnika dotykowego. Aktywacja czujnika dotykowego powoduje przejście do wykonania kolejnych instrukcji programu.
2. Jeśli czujnik IR jest aktywny uruchamia się silnik 1, a jeśli czujnik jest nie aktywny – uruchamia się silnik 2.
3. Silnik 1 działa tak długo dopóki czujnik IR będzie aktywny.
4. Silnik 2 działa tak długo dopóki czujnik IR będzie nieaktywny.

Wskazówka: podziel zadanie na mniejsze części, rozwiązania fragmentów zadania złącz w całość.

Poprawnie rozwiązanie zadanie może prezentować się następująco:





Scenariusz 3

Temat:

Pierwszy prosty robot - pętla.

Czas trwania zajęć: 2x45 minut

Cel ogólny:

Kształtowanie umiejętności budowania robotów z wykorzystaniem zestawu RoboMaker Pro, kształtowanie umiejętności wykorzystania pętli.

Cele szczegółowe:

- kształtowanie umiejętności budowania robotów z zastosowaniem zestawu RoboMaker Pro.
- kształtowanie umiejętności definiowania podstawowych konstrukcji algorytmicznych z wykorzystaniem pętli;
- kształtowanie umiejętności tworzenia prostego programu z wykorzystaniem pętli;
- kształtowanie umiejętności poszukiwania i utrwalania wzorców;
- kształtowanie umiejętności podziału zadania na części i odniesienia poszczególnych elementów programu do zapisu algorytmu w postaci schematu blokowego.

Cele operacyjne:

- uczeń zna zasadę działania pętli;
- uczeń potrafi ułożyć sekwencję poleceń sterującą elementami robota;
- uczeń potrafi ułożyć sekwencję poleceń wykorzystującą czujniki robota;
- uczeń potrafi ułożyć program wykorzystujący pętlę;
- uczeń rozróżnia rodzaje pętli;
- uczeń zna zasady rysowania schematu blokowego algorytmu;
- uczeń potrafi narysować schematy blokowe algorytmów wykorzystujących pętlę, w tym pętlę nieskończoną.

Formy pracy:

- zespołowa;
- w grupach, w parach;
- indywidualna.



Metody pracy:

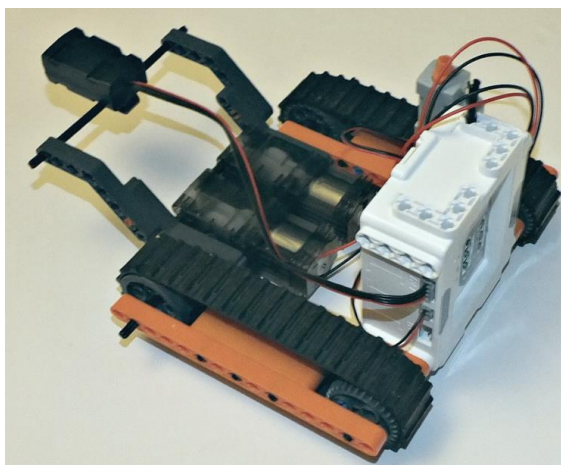
- asymilacja wiedzy;
- samodzielnego dochodzenia do wiedzy (problemowa);
- praktyczna (ćwiczebna).

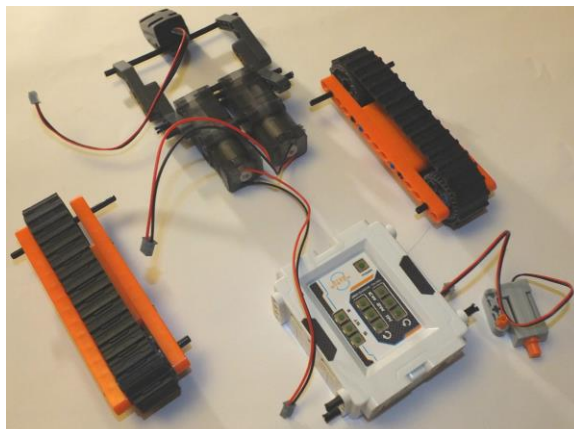
Środki dydaktyczne:

- zestaw RoboMaker Pro;
- tablet z zainstalowaną aplikacją RoboMaker, tryb: **Twórz – konstrukcja własna**.

Ostatnią konstrukcją algorytmiczną jaką powinni poznać uczniowie jest pętla. Pętla umożliwia cykliczne wykonywanie poleceń określoną liczbę razy lub do momentu spełnienia jakiegoś warunku.

Przed przystąpieniem do właściwego zadania algorytmicznego, zbuduj z uczniami prostego robota. Np. jak poniżej (w przykładzie silniki podłączono do złącz M2 i M3, uczniowie powinni rozumieć, że silniki mogą być podłączone do dowolnego złącza – M1, M2 i M3, ważne jest aby tworząc program zastosować odpowiednie oznaczenia silników).





Możesz również wykorzystać konstrukcję zbudowaną wg instrukcji znajdującej się w aplikacji RoboMaker, wykorzystującą dwa silniki, zmień jedynie usytuowanie i ilość czujników.

Zadanie dla uczniów:

Stwórz program działający w następujący sposób:

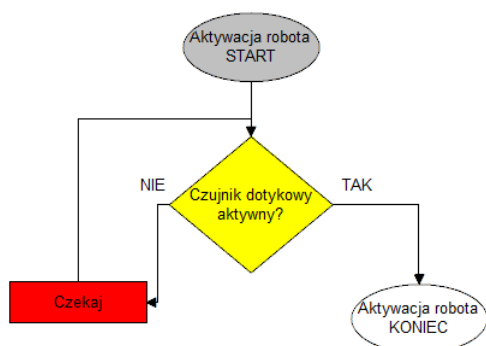
1. Po uruchomieniu programu robot czeka na aktywację za pomocą czujnika dotyku (przyciśnięcie). Po aktywacji, robot oczekuje na zbliżający się przedmiot lub dłoń ucznia.
2. Po zbliżeniu przedmiotu (lub dłoni) do czujnika IR, robot cofa się (przemieszcza się do tyłu- należy przyjąć, że przód robota jest po stronie czujnika IR).
3. Robot cofa się tak długo, dopóki czujnik IR przestanie być aktywny.
4. Po pięciokrotnym wykonaniu pkt 2 i 3 robot obraca się (np. o 180 stopni) i zaczyna przemieszczać się do przodu tak długo, dopóki nie napotka na przeszkodę.
5. Po napotkaniu przeszkody robot zatrzymuje się (program kończy swoje działanie). Działanie robota można wzbogacić dźwiękami.

Wskazówka:

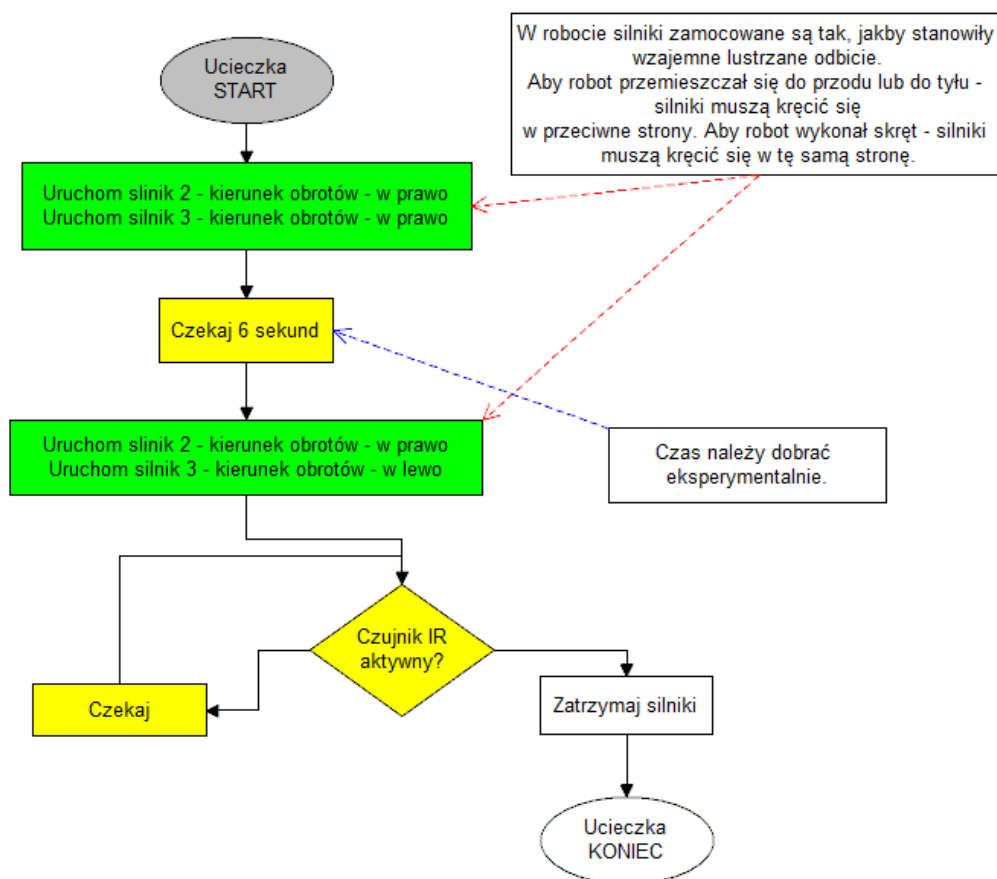
Podziel zadanie na mniejsze, łatwiejsze do wykonania części (podproblemy). Nazwij w dowolny sposób poszczególne części rozwiązania.



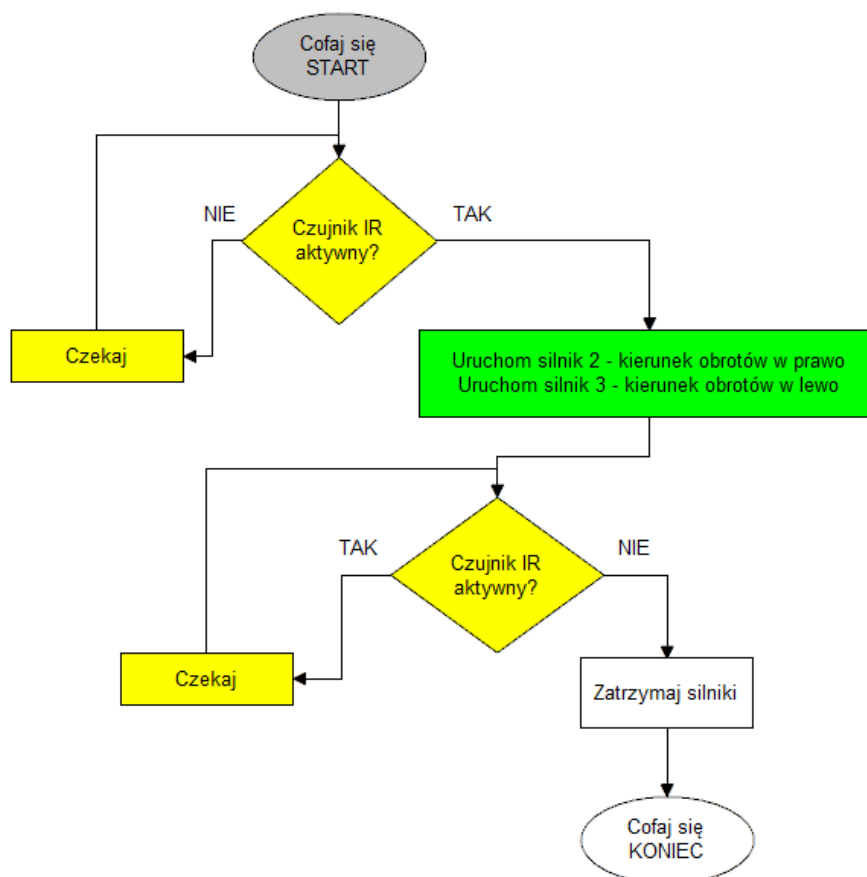
Zasugeruj uczniom, aby najpierw ułożyli fragment programu, który zrealizuje pierwszą część zadania – aktywację robota (pkt 1). Poproś uczniów, aby rozwiązanie tej części zadania przedstawili również w postaci algorytmu - schematu blokowego.



Następnie zasugeruj, aby uczniowie utworzyli algorytm i fragment programu realizujący ostatnią część zadania (robot wykonuje zwrot i porusza się do przodu do napotkania przeszkody).



Teraz kolej na najtrudniejszą część zadania – stworzenie algorytmu i fragmentu programu, który będzie powodował cofanie się robota, jeśli do jego czujnika IR zbliży się przedmiot (dłoń itp.). Poproś aby uczniowie nie rozważali pięciokrotnego powtarzania się reakcji robota, niech najpierw ułożą algorytm i program dla uproszczonego problemu: jeśli czujnik IR aktywny robot cofa się do momentu gdy czujnik IR przestanie być aktywny.

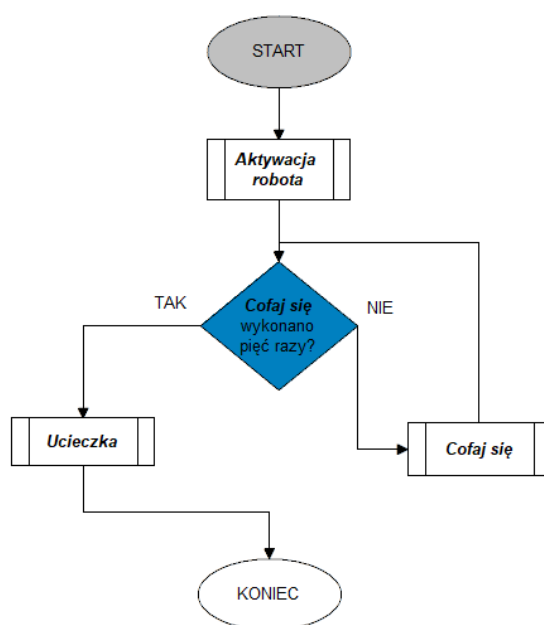


W przykładzie programu dodany został dźwięk, który nie został uwzględniony w algorytmie.

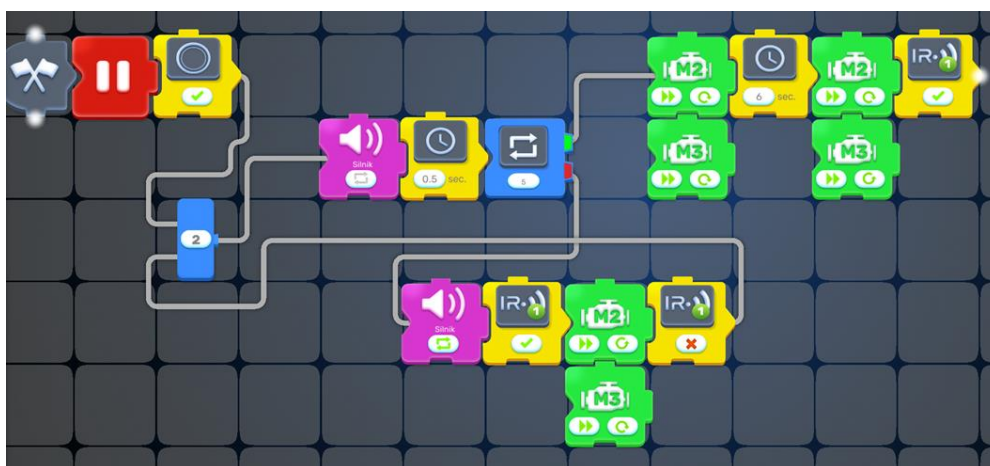
Powyższe algorytmy nazwano: „Aktywacja robota”, „Ucieczka”, „Cofaj się”. Uczniowie mogą nadać swoje nazwy.



Końcowym etapem tworzenia programu jest złożenie w całość rozwiązań poszczególnych części całego problemu, pamiętając że zgodnie z treścią zadania fragment nazwany „Cofaj się” ma powtórzyć się pięć razy i dopiero potem wykonane zostaną polecenia końcowy (algorytm „Ucieczka”). Wyjaśnij uczniom, że cykliczne wykonywanie czynności (poleceń) określoną ilość razy nazywa się pętlą. Daj uczniom czas aby sami spróbowali dojść do tego jak przedstawić pętlę na schemacie blokowym. Skoryguj ich pomysły lub w razie trudności naprowadź na właściwe rozwiązanie. Po wprowadzeniu pętli do algorytmu poprawne rozwiązanie powinno wyglądać jak poniżej:



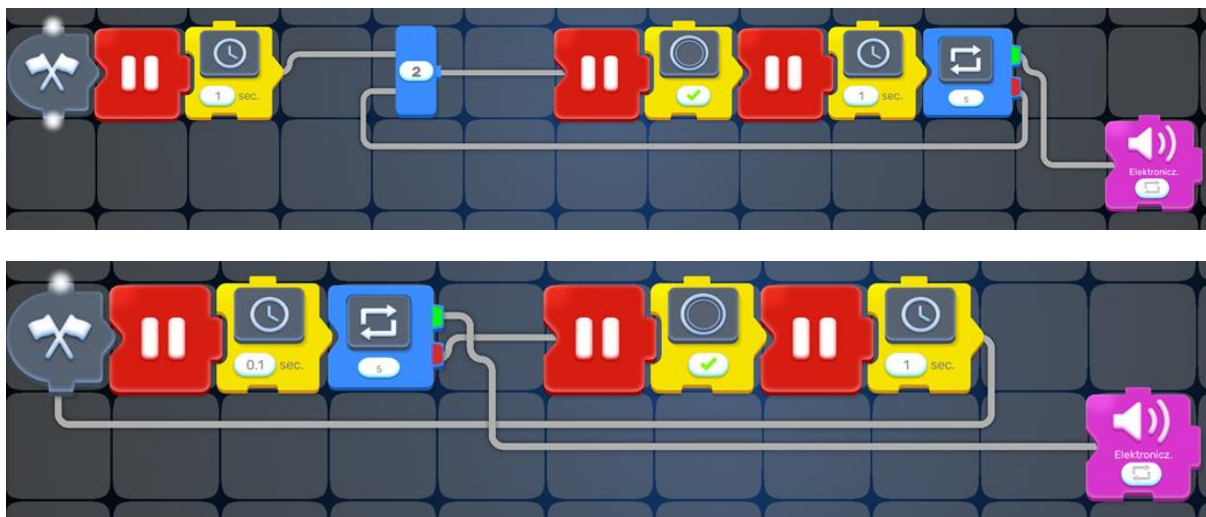
Poniżej przykład programu realizującego algorytm z dodatkowymi poleceniami generującymi dźwięk:





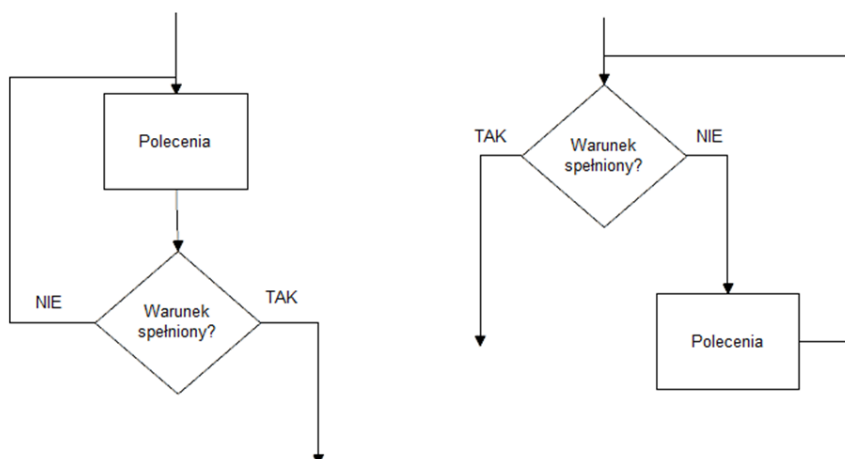
Po tym ćwiczeniu omów jeszcze raz działanie pętli, poproś aby uczniowie dokładnie policzyli ile razy wykonane zostaną polecenia zawarte wewnątrz pętli. Uczniowie powinni zauważyć, że część poleceń zostanie wykonana pięć razy, a część sześć razy.

Poproś aby uczniowie ułożyli programy zgodnie z poniższymi przykładami (np. połowa zespołu układa pierwszy program, druga połowa drugi):



Aby pojawił się *Dźwięk elektroniczny* obydwa programy wymagają kilkukrotnej aktywacji czujnika dotykowego (przyciśnięcie). Pomimo, że w bloku rozdzielacza podana jest liczba 5 – pierwszy program będzie wymagał sześciokrotnej aktywacji czujnika dotykowego, a drugi program – pięciokrotnej. Zwróć uwagę uczniów na położenie bloku rozdzielacza. W pierwszym programie blok rozdzielacza zamyka, a w drugim rozpoczyna sekwencję poleceń wykonywanych w pętli.

Uogólniając, pętle można przedstawić na dwa sposoby:

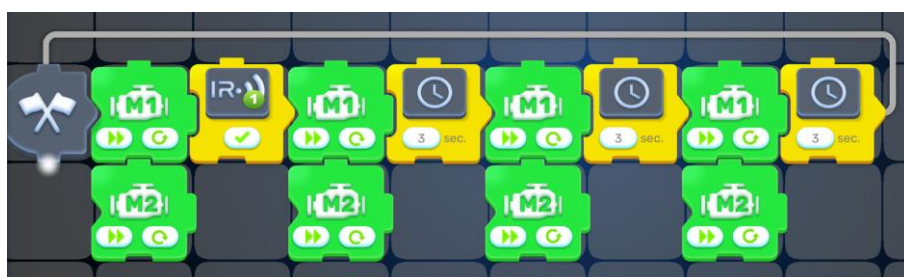




Różnica działania powyższych pętli jest taka, że w pierwszym przypadku najpierw zostaną wykonane polecenia wewnątrz pętli, a potem zostanie sprawdzony warunek (powodujący ewentualne opuszczenie pętli), w drugim przypadku - najpierw zostanie sprawdzony warunek, a potem zostaną wykonane instrukcje wewnątrz pętli.

Szczególnym przypadkiem pętli, która często stosowana jest w programowaniu robotów, jest pętla nieskończona. Taka pętla wykonuje się do momentu zatrzymania działania programu.

Przykład pętli nieskończonej:



Poproś aby uczniowie zaprogramowali robota wg powyższego przykładu oraz przetestowali jego działanie. Przeprowadź dyskusję nt. stosowania pętli nieskończonej.



Scenariusz 4

Temat:

Obroty silnika – mini praca badawcza.

Czas trwania zajęć: 2x45 minut

Cel ogólny:

Kształtowanie umiejętności wykorzystania elementów zestawu RoboMaker Pro do pracy badawczej.

Cele szczegółowe:

- kształtowanie umiejętności definiowania problemu;
- kształtowanie umiejętności doboru metod badawczych na bazie zestawu RoboMaker Pro;
- kształtowanie umiejętności tworzenia własnych konstrukcji i programów z wykorzystaniem czujników i silników;
- kształtowanie umiejętności matematycznych i fizycznych;
- utrwalanie wzorców, nabywanie doświadczenia praktycznego.

Cele operacyjne:

- uczeń zna pojęcie *prędkość obrotowa*, potrafi podać przykłady, wie jak się oblicza prędkość obrotową;
- uczeń potrafi wykorzystać zestaw RoboMaker Pro do rozwiązywania nietypowych problemów;
- uczeń potrafi określić niedoskonałości przyjętej metody badawczej;
- uczeń zna podstawowe bloki języka programowania RoboMaker i potrafi je kreatywnie wykorzystywać;
- uczeń potrafi konstruować algorytmy i tworzyć programy w sytuacji nietypowej;

Formy pracy:

- zespołowa;
- w grupach, w parach;
- indywidualna.

Metody pracy:

- asymilacja wiedzy;



- samodzielnego dochodzenia do wiedzy (problemowa);
- praktyczna (ćwiczebna).

Środki dydaktyczne:

- zestaw RoboMaker Pro;
- tablet z zainstalowaną aplikacją RoboMaker, tryb: **Twórz – Konstrukcja własna**;
- stoper (smartfon z uruchomioną funkcją stopera).

Zapoznaj uczniów z pojęciem „prędkość obrotowa” (ilość obrotów w określonej jednostce czasu – np. obroty/minutę lub obroty/sekundę). Aby określić prędkość obrotową silnika, należałoby zmierzyć (np. za pomocą stopera) czas jednego pełnego obrotu. Jest to jednak kłopotliwe, ponieważ trudno jest zmierzyć czas jednego obrotu, z uwagi na np. niewystarczająco dobry refleks osoby uruchamiającej stoper. Łatwiej byłoby zmierzyć czas np. pięciu obrotów – czas ten po podzieleniu przez 5 da nam czas jednego obrotu.

Zadanie dla uczniów:

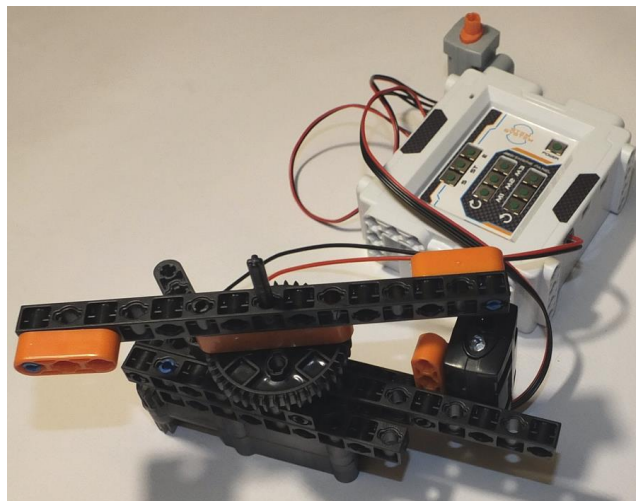
Należy określić prędkość obrotową silnika. Dla określenia prędkości obrotowej silnika należy zbudować urządzenie i ułożyć odpowiedni program. Urządzenie i program powinny działać następująco:

1. Oczekiwanie na aktywację czujnika dotyku.
2. Uruchomienie silnika.
3. Zatrzymanie silnika po wykonaniu pięciu obrotów.
4. Wygenerowanie dźwięku.

Taki sposób działania powinien umożliwić wykonanie odpowiedniego pomiaru:

1. W momencie aktywacji czujnika dotykowego należy włączyć stoper.
2. W momencie pojawienia się sygnału dźwiękowego należy wyłączyć stoper.
3. Czas wskazany na stoperze należy podzielić przez 5.

Przedyskutuj z uczniami problem sterowania silnikiem w celu wykonania konkretnej ilości obrotów. Zaproponuj uczniom, aby zbudować urządzenie składające się z czujnika dotykowego, czujnika IR i silnika. Czujnik dotyku będzie uruchamiał silnik, a czujnik IR będzie „nadzorował” ilość wykonanych obrotów. Przykład urządzenia poniżej:

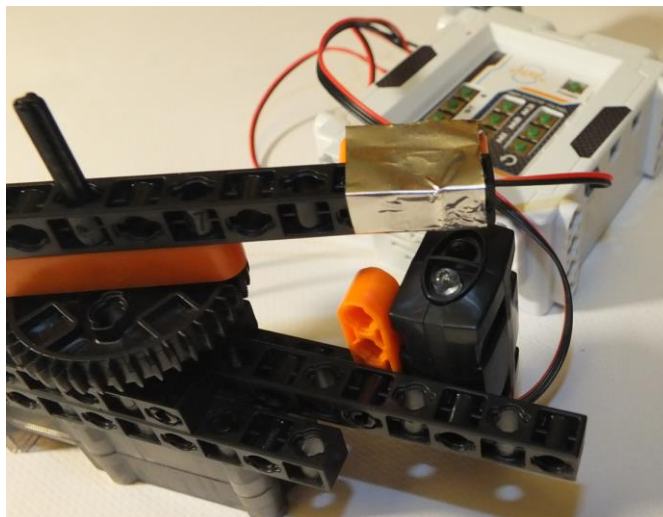


Poproś uczniów o próbę utworzenia algorytmu działania takiego urządzenia. Po przedyskutowaniu propozycji uczniów przedstaw program:



Zwróć uczniom uwagę, że tak skonstruowane urządzenie sterowane przedstawionym programem będzie uaktywniało czujnik IR dwa razy w ciągu pełnego obrotu silnika, dlatego ilość przebiegów pętli należy podwoić. Ponieważ blok rozdzielacza jest za instrukcją warunku prostego związanego z czujnikiem IR (patrz poprzedni scenariusz), aby uzyskać 10 przebiegów, w bloku rozdzielacza podana jest liczba 9.

Poproś aby uczniowie przetestowali działanie urządzenia i programu. W celu polepszenia działania czujnika IR owiń folią aluminiową elementy odbijające promienie podczerwone.

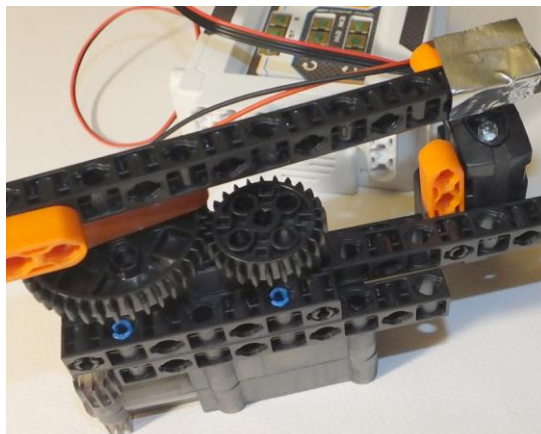


Zwróć uczniom uwagę na to, że specyfika blokowego języka programowania jest taka, że po aktywacji czujnika silnik jest zatrzymywany – powoduje to konieczność ponownego uruchomienia silnika. Przedstawiony program minimalizuje ten efekt. Należy również mieć na uwadze fakt, że podczas wykonywania programu, cały czas występuje transmisja danych pomiędzy kostką sterującą a tabletem, co może powodować przypadki, że czujnik IR „nie zdąży” się aktywować.

Poproś aby uczniowie wykonali kilka (np. 10) pomiarów czasu w jakim silnik wykona 5 obrotów. Ponieważ, jak już wspomniano, czasami może nastąpić błąd reakcji czujnika IR lub stoper mierzący czas zostanie włączony/wyłączony w nieodpowiednim momencie – należy odrzucić te pomiary, których wartości radykalnie odbiegają od pozostałych. Następnie z pozostawionych wartości należy wyliczyć średnią i podzielić ją przez 5, w ten sposób otrzymamy czas jednego obrotu – a tym samym prędkość obrotową silnika.

Przedyskutuj z uczniami na ile dokładne jest wyznaczenie prędkości obrotowej silnika taką metodą. Poproś, aby uczniowie zastanowili się czy można zmodyfikować urządzenie i program tak aby zminimalizować błędy pomiaru.

Możesz zaproponować uczniom zmodyfikowanie urządzenia montując przekładnię zębatą i pomiar prędkości obrotowej drugiej zębatki.



Porównując stosunek prędkości obrotowej silnika (pierwsza zębatka) i prędkości obrotową drugiej zębatki ze stosunkiem ilości zębów w drugiej i pierwszej zębatce, z łatwością wytłumaczysz uczniom zasadę działania przekładni zębatej. Pamiętaj jednak o tym, że jeśli za pomocą przekładni zmniejszysz prędkość obrotową to równocześnie zwiększysz „moc” układu, zwiększając prędkość obrotową – zmniejszysz „moc” (przekładania zwiększa/zmniejsza moment obrotowy, lecz ponieważ dla uczniów szkoły podstawowej to pojęcie nie jest znane, nie zawiera się w podstawie programowej fizyki, można zastępczo użyć pojęcia „moc”, pamiętając że jest to w ujęciu fizycznym błędne określenie).

Znając prędkość obrotową silnika można obliczyć np. czas działania silnika w celu przebycia przez robota określonej drogi (potrzebne będzie do tego zmierzenie średnicy koła napędzającego gąsienicę wraz z jej grubością). To zagadnienie możesz spróbować omówić i przećwiczyć jedynie z uzdolnionymi matematycznie uczniami starszych klas szkoły podstawowej.



Scenariusz 5

Temat:

Powtarzanie – droga do sukcesu.

Czas trwania zajęć: 45 minut (lub 2x45 jeśli na zajęciach będzie budowany robot)

Cel ogólny:

Kształtowanie umiejętności odnajdywania powtarzalnych elementów problemu.

Cele szczegółowe:

- kształtowanie umiejętności definiowania konstrukcji algorytmicznych;
- kształtowanie umiejętności tworzenia programu z wykorzystaniem czujników i silników;
- kształtowanie umiejętności poszukiwania i utrwalania wzorców;
- kształtowanie umiejętności podziału zadania na części i wykorzystania prostych rozwiązań do rozwiązania bardziej skomplikowanych problemów.

Cele operacyjne:

- uczeń potrafi ułożyć sekwencję poleceń sterującą robotem;
- uczeń potrafi wykorzystywać proste fragmenty programów do realizacji innych złożonych zadań;
- uczeń zna podstawowe bloki języka programowania RoboMaker;
- uczeń kreatywnie rozwiązuje proste zagadnienia (części całego zadania);
- uczeń kreatywnie rozwiązuje zadanie (problem) poprzez łączenie rozwiązań prostszych problemów;

Formy pracy:

- zespołowa;
- w grupach, w parach;
- indywidualna.

Metody pracy:

- asymilacja wiedzy;
- samodzielnego dochodzenia do wiedzy (problemowa);
- praktyczna (ćwiczebna).



Środki dydaktyczne:

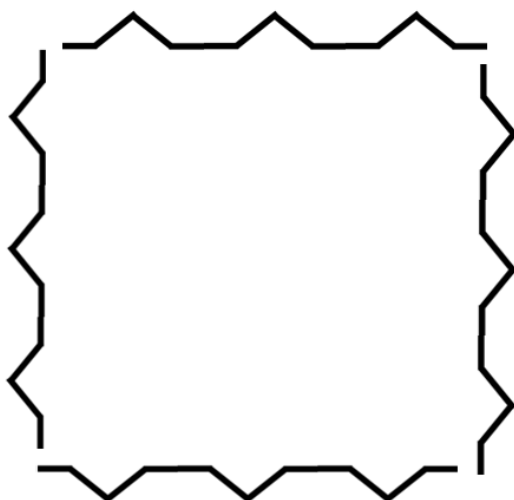
- zestaw RoboMaker Pro;
- tablet z zainstalowaną aplikacją RoboMaker, tryb: **Twórz – Konstrukcja własna** lub **Twórz – Sumobot – Line follower**;
- plansze wg opisu.

Przygotuj dwie plansze (namaluj na brystolu, zachowaj szerokość czarnej linii taką, jak na planszy dostarczonej z zestawem):

Plansza 1



Plansza 2



Zbuduj z uczniami robota Sumobot, wersja Line Follower (w oparciu o instrukcję zawartą w aplikacji RoboMaker).

Pokaż uczniom *Planszę 1* i poproś, aby ułożyli program, który spowoduje, że robot przejedzie trasę od początku do końca. Początek trasy – robot ustawiony tak, że początek linii znajduje się pomiędzy czujnikami IR robota, koniec trasy – koniec linii znajduje się między czujnikami IR. Program oczywiście będzie banalny. Jedynie metodą prób i błędów należy dobrać czas działania robota, tak aby przebył dokładnie wskazaną drogę.



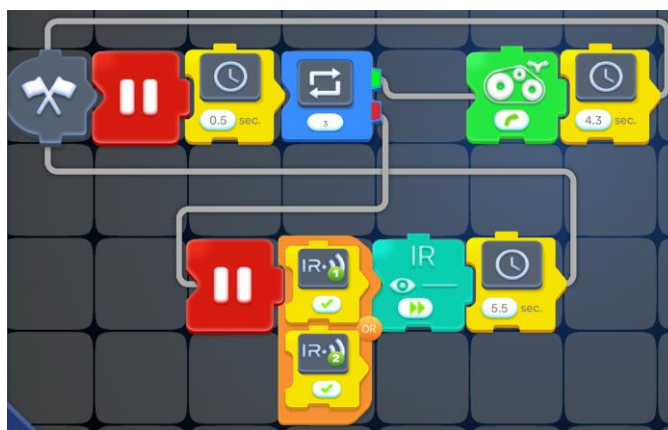
Pokaż uczniom Planszę 2 i przedstaw treść zadania.

Zadanie dla uczniów:

Ułóż program, aby robot ustawiony na początku dowolnej czarnej linii przebył drogę złożoną z wszystkich czarnych linii.

Zasugeruj uczniom aby wykorzystali sekwencję poleceń z wcześniej ułożonego programu.

Finalny efekt mógłby prezentować się następująco:



Zwróć uwagę uczniów na to, że gdyby od razu otrzymali zadanie do wykonania, to ich postępowanie mogłoby rozpocząć się od poszukiwania elementów powtarzalnych. Następnie należałoby ułożyć algorytm/program dla jednego powtarzalnego elementu i powtórzyć go odpowiednią ilość razy, wzbogacając o polecenia powodujące obrót robota o 90°. Jest to typowa strategia myślenia komputacyjnego, w którym występuje podział zadania (problemu) na mniejsze łatwiejsze do wykonania części (podproblemy).



Scenariusz 6

Temat:

Analiza programu robota.

Czas trwania zajęć: 45 minut (budowanie robota) + 45 minut

Cel ogólny:

kształtowanie umiejętności tworzenia algorytmów i analizy gotowego programu.

Cele szczegółowe:

- kształtowanie umiejętności odnajdywania w złożonym problemie podstawowych konstrukcji algorytmicznych;
- kształtowanie umiejętności analizy działania robota i programu;
- kształtowanie umiejętności poszukiwania i utrwalania wzorców.

Cele operacyjne:

- uczeń potrafi narysować algorytm do gotowego programu;
- uczeń analizuje i krytycznie ocenia rozwiązanie;
- uczeń świadomie eksperymentuje poszukując nowych możliwości;
- uczeń zna bloki języka programowania RoboMaker i rozumie ideę ich zastosowania;
- uczeń zna zasady rysowania schematu blokowego algorytmu;
- uczeń potrafi narysować schematy blokowe algorytmów;

Formy pracy:

- zespołowa;
- w grupach, w parach;
- indywidualna.

Metody pracy:

- asymilacja wiedzy;
- samodzielnego dochodzenia do wiedzy (problemowa);
- praktyczna (ćwiczebna).



Środki dydaktyczne:

- zestaw RoboMaker Pro;
- tablet z zainstalowaną aplikacją RoboMaker tryb, dla przykładu w tym scenariuszu: **Ucz się – Guardian - Strażnicy i złodzieje** oraz **Twórz – Guardian - Strażnicy i złodzieje**.

Zbuduj z uczniami dowolnego robota wg instrukcji zamieszczonej w aplikacji. Zaprogramuj robota wg wskazówek zawartych w aplikacji (tryb Ucz się). Poproś aby uczniowie przetestowali działanie programu.

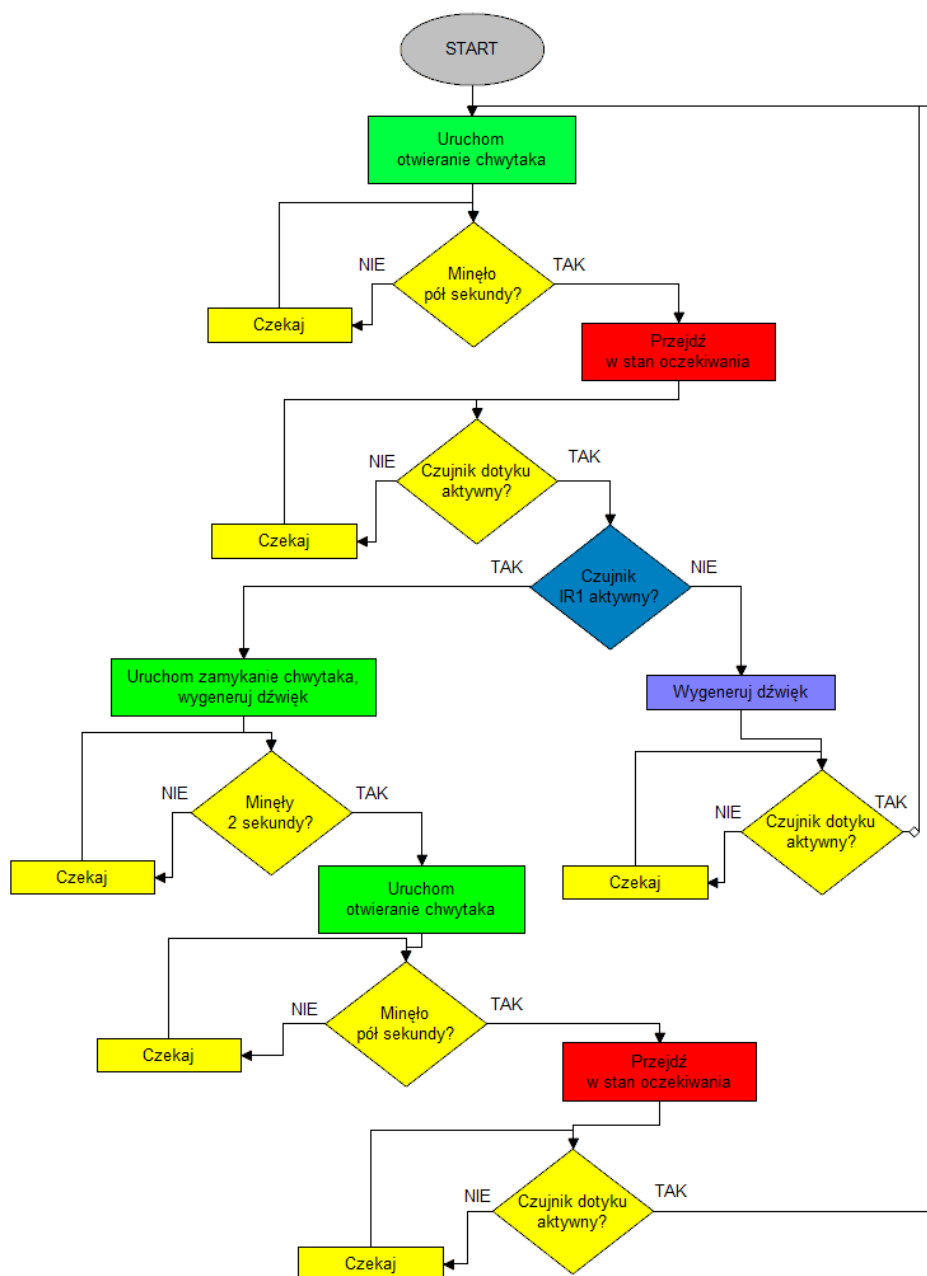
Zadanie dla uczniów:

Należy narysować algorytm programu.

Przykładowo, dla programu jak poniżej:



Schemat blokowy algorytmu powinien przedstawiać się następująco:



Przedyskutuj z uczniami program i algorytm. Dla przedstawionego przykładu zwróć uwagę, że algorytm (i program) zawiera pętlę nieskończoną. Poproś aby uczniowie zastanowili się dlaczego w programie użyto pętli nieskończonej, jaki jest sens zastosowania takiej pętli, co by było gdyby program był pozbawiony tej pętli. Poproś aby uczniowie spróbowali ułożyć analogiczny program z zastosowaniem pętli z konkretną ilością powtórzeń.



Poproś uczniów, aby przetestowali działanie robota po zmianie czasów działania chwytaka (dla kolejnych prób zmiany czasu powinny być nie większe niż 0,3 – 0,5 sekundy).

Na zakończenie zajęć podsumuj możliwości modyfikacji programu i ich wpływ na działanie robota.



Mam nadzieję, że niniejszy poradnik przybliży nauczycielom problematykę związaną z zastosowaniem robotyki w edukacji. Jestem przekonany, że zawarte tu informacje, przykładowe scenariusze będą dla Państwa inspiracją do twórczego wykorzystania robotów edukacyjnych RoboMaker Pro.

Wojciech Kolarz