

MoboLab – roboty i tablety w Twojej szkole

Obszar I. „Zakoduj robota”

Scenariusze lekcji i zajęć pozalekcyjnych

SCENARIUSZ 13. POJAZD PODĄŻAJĄCY ZA ŚCIEŻKĄ (CZARNĄ LINIĄ)

scenariusz zajęć pozalekcyjnych

autor: Michał Kłosiński

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

mBlock, mBot, Scratch, robot, sterowanie, czujnik linii

KRÓTKI OPIS ZAJĘĆ:

Podczas zajęć uczniowie i uczennice rozwijają i utrwalają wiedzę o środowisku programistycznym **mBlock** opartym na **języku Scratch**. Za pomocą bloków służących do programowania czujnika linii tworzą skrypt sterujący robotem **mBot**, tak aby robot podążał za czarną linią. Wykonują zadanie: programują robota i modyfikują program w taki sposób, by robot powrócił na czarną linię, gdy z niej zjedzie.

WIEDZA I UMIEJĘTNOŚCI ZDOBYTE PRZEZ UCZNIĄ / UCZENNICĘ:

- zna budowę robota mBot,
- zna podstawowe elementy interfejsu programu mBlock,
- wie, co to jest czujnik linii/dotyku,
- swobodnie porusza się po środowisku mBlock, wie, gdzie szukać bloków do tworzenia skryptów sterujących robotem,
- potrafi stworzyć skrypt, który pozwoli na warunkowy ruch robotem z wykorzystaniem czujnika linii/dotyku,
- potrafi sterować programem za pomocą pilota.

GRUPA DOCELOWA:

Starsze klasy szkoły podstawowej (VII-) i klasy gimnazjalne (po dostosowaniu: możliwość realizacji w młodszych klasach: I-III i IV-VI szkoły podstawowej)

LICZBA UCZNIÓW / UCZENNIC W KLASIE:

Liczba optymalna: 12, liczba maksymalna: 16

CZAS TRWANIA ZAJĘĆ:

90 min (lub 2 x 45 minut)

STOPIEŃ TRUDNOŚCI/SKOMPLIKOWANIA

(w skali od 1 do 5 dla obszaru I. „Zakoduj robota”):

3 (5 wraz z zadaniami dodatkowymi)

POTRZEBNY SPRZĘT I OPROGRAMOWANIE:

- komputer (przenośny lub stacjonarny),
- program mBlock (do pobrania ze strony: <http://www.mblock.cc/download/>),
- roboty mBot (złożone) – 1 robot na 1 ucznia / uczennicę, a w przypadku mniejszej liczby robotów: 1 robot na 2 lub 3 uczniów / uczennic,
- kable USB (po 1 dla każdego robota),
- projektor i laptop (w części teoretycznej).

CO NALEŻY PRZYGOTOWAĆ PRZED ZAJĘCIAMI:

- zainstalować program mBlock,
- sprawdzić poprawne działanie robota mBot oraz połączenie z programem mBlock (jeśli wystąpią problemy, warto zainstalować ponownie sterownik Arduino),
- dopasować stopień trudności zadania do potrzeb i możliwości klasy, dla której organizowane są zajęcia według wskazówek zawartych w scenariuszu,
- sprawdzić, czy wszystkie elementy robota są prawidłowo podpięte i czy diody działają poprawnie,
- sprawdzić stan baterii zasilających robota.

KOMPETENCJE OSOBY PROWADZĄCEJ:

- zna i rozumie działanie wykorzystywanych bloków w programach Scratch i mBlock,
- potrafi podłączyć robota do komputera, używając kabla USB,
- wie, jakich bloków należy użyć do sterowania robotem,
- zna podstawowe pojęcia programistyczne (skrypt, program, pętla),
- wie, dlaczego warto uczyć się programowania i jakie korzyści daje posiadanie umiejętności programistycznych,
- potrafi zachęcić do nauki programowania zarówno chłopców, jak i dziewczynki.

PRZEBIEG ZAJĘĆ:

Część 1. – 45 minut

Wprowadzenie, rozmowa o czujniku linii - 10 minut

Wprowadzamy uczniów w tematykę lekcji, przypominamy podstawowe wiadomości na temat budowy robota mBot i czujników, w jakie jest wyposażony robot. Krótko omawiamy zastosowania czujników. Czujniki stosowane są wtedy, gdy na działanie jakiegoś urządzenia mają wpływ czynniki zewnętrzne, np. występujące na drodze przeszkody czy zapadający zmrok. Przykłady działania czujników to między innymi:

- ⇒ robot odkurzający, który omija przeszkody, nie spada ze schodów, może włączyć się samoczynnie o określonej godzinie,
- ⇒ oświetlenie ulicy załączające się przy zapadającym zmroku,
- ⇒ oświetlenie klatki schodowej, które włącza się, gdy ktoś wchodzi,
- ⇒ wycieraczki w samochodzie uruchamiające się automatycznie podczas deszczu.

Podczas dzisiejszych zajęć zaprogramujemy robota, który będzie podążał za czarną linią, a jeśli z niej zboczy, automatycznie dokona korekty: wróci na tor linii. Wykorzystamy znany już wam czujnik linii – na wcześniejszych zajęciach budowaliśmy program robota, który nie spada ze stołu lub biurka wyczuwając brak podłoża. Ten sam czujnik może służyć do wskazania drogi naszemu robotowi.

Czujnik linii umiejscowiony jest na przodzie robota i niebieskimi diodami sygnalizuje swoje działanie. Na początku sprawdzamy, do którego portu jest podłączony czujnik linii. Czujnik linii składa się z dwóch czujników, które sprawdzają kolor podłoża w odległości ok. 1-2 cm. Po sprawdzeniu koloru, wartość czujnika przybiera postać 1 lub 0. Dzięki temu robot może wykrywać ciemne linie na jasnym podłożu lub krawędź stołu czy schodów.



źródło: <http://www.trobot.pl>

Określenie warunków programu spełniających założone zadanie – 20 minut

Cel: określenie warunków, które musi spełniać program, aby spełniał kryteria sukcesu.

Założenie jest takie, że robot startuje na czarnej linii – obydwa czujniki ją wykrywają. Należy zastanowić się:

- ⇒ *Jak będzie działał nasz program?*
- ⇒ *Jak powinien zachowywać się robot, żeby nie zjechał z kursu?*
- ⇒ *Czy zawsze obydwa diody na czujniku są zapalone? Co to oznacza? (Zapalona dioda oznacza wykrycie przez czujnik czarnej linii.)*

Należy wziąć pod uwagę różne przypadki:

1. obydwa czujniki wykrywają czarną linię,
2. tylko lewy czujnik wykrywa czarną linię,
3. tylko prawy czujnik wykrywa czarną linię,
4. żaden z czujników nie wykrywa czarnej linii.

Co powinien wykonać robot w każdym z powyższych przypadków?

1. obydwa czujniki wykrywają czarną linię – robot jedzie do przodu,
2. tylko lewy czujnik wykrywa czarną linię – robot zbacza z kursu, czyli musi lekko skręcić w lewo, żeby wrócić na kurs,
3. tylko prawy czujnik wykrywa czarną linię – robot zbacza z kursu, czyli musi lekko skręcić w prawo, żeby wrócić na kurs,
4. żaden z dwóch czujników nie wykrywa czarnej linii – robot przejechał za daleko

za czarną linię, musi więc trochę się cofnąć, żeby ją odnaleźć.

Należy pamiętać, że robot powinien jechać wolno, aby czujniki miały szansę zadziałać poprawnie – wtedy korekta ruchu robota będzie skuteczniejsza. Przy dużych prędkościach może to być utrudnione.

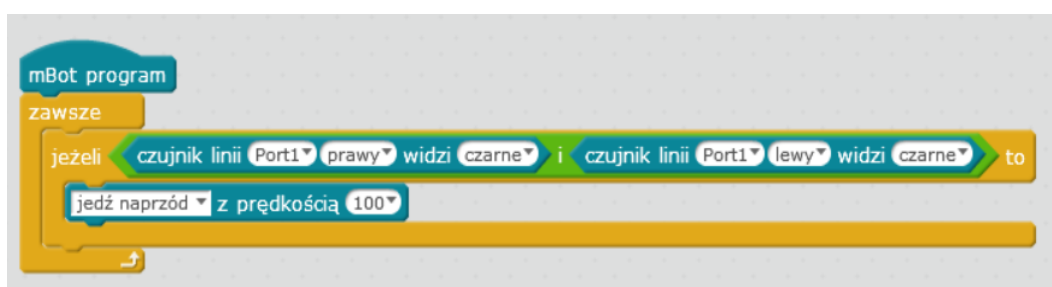
Dzielimy uczniów na zespoły. Liczba osób w zespole zależy od liczby robotów, które mamy do dyspozycji (optymalnie: 1 robot na 2 osoby + 1 robot dla osoby prowadzącej lekcję). Każdy zespół otrzymuje jednego robota. Przypomina uczniom o sprawdzeniu numeru portu, do którego podłączony jest czujnik linii w robocie.

Budowa programu, część 1. – 15 minut

Korzystając z projektora i programu mBlock zaczynamy od omówienia podstawowych bloków potrzebnych do wykonania podstawowego zadania. Bloki można znaleźć w kategoriach: **Roboty, Kontrola, Wyrażenia**.

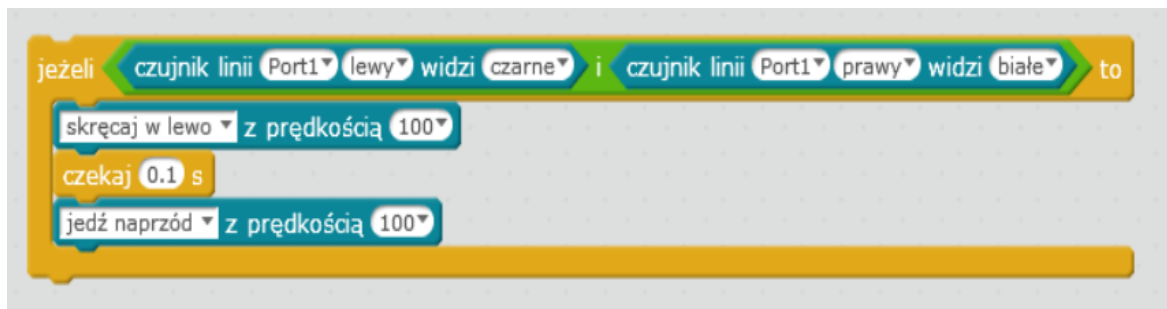


- ⇒ 1 warunek: robot ma jechać do przodu, jeśli obydwa czujniki „widzą” czarną linię. Program może wyglądać następująco:



Po ułożeniu bloków uczniowie wgrzywają program do robota i sprawdzają, czy działa – robot jedzie do przodu.

- ⇒ 2 warunek: tylko lewy czujnik wykrywa czarną linię – robot zbacza z kursu, musi lekko skręcić w lewo, żeby wrócić na kurs. Program może wyglądać następująco:



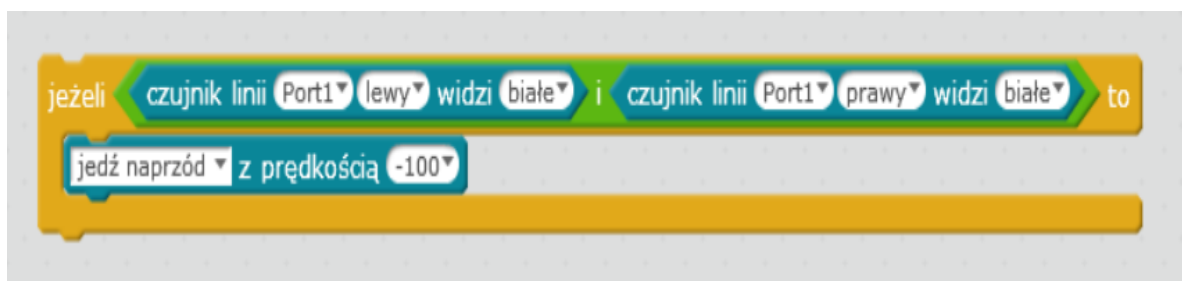
Po ułożeniu bloków uczniowie wgrzywają program do robota i sprawdzają, czy działa – po wykryciu na lewym czujniku linii, robot skręci w lewo, czyli dokona korekty swojego kursu.

- ⇒ 3 warunek: tylko prawy czujnik wykrywa czarną linię – robot zbacza z kursu, musi lekko skręcić w prawo, żeby wrócić na kurs. Program może wyglądać następująco:



Po ułożeniu bloków uczniowie wgrzywają program do robota i sprawdzają, czy działa – po wykryciu na prawym czujniku linii, robot skręci w prawo, czyli dokona korekty swojego kursu.

- ⇒ 4 warunek: żaden z czujników nie wykrywa czarnej linii – robot przejechał za daleko i opuścił czarną linię, musi więc trochę się cofnąć, żeby ją odnaleźć. Program może wyglądać następująco:



Po ułożeniu bloków uczniowie wgrzywają program do robota i sprawdzają, czy działa – po wykryciu na obydwu czujnikach braku linii, robot cofnie się do tyłu, czyli dokona korekty swojego kursu.

W tym miejscu możliwa jest przerwa (kolejna część scenariusza będzie realizowana na następnych zajęciach).

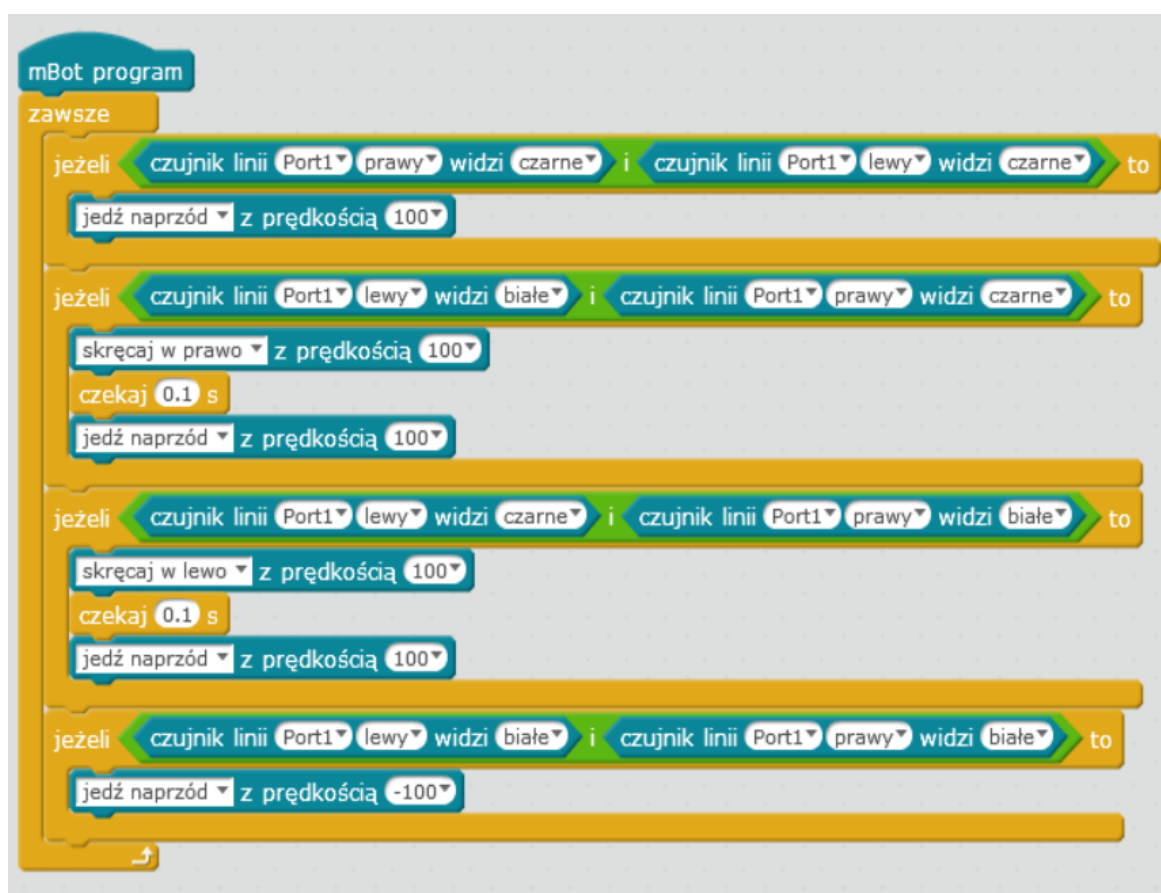
Część 2 – 45 minut

Przypomnienie materiału z pierwszej części zajęć – 15 minut

Rozpoczynamy od krótkiego przypomnienia materiału z poprzedniej części zajęć i odtworzenia skryptów, które na niej powstały. Jeśli jakiś program nie powstał, prosimy o stworzenie go. W przypadku problemów pokazujemy przykładowy program dla jednego z czterech warunków, a następnie prosimy o stworzenie programów dla pozostałych warunków.

Budowa programu, część 2. – 15 minut

Po połączeniu skryptów dla wszystkich warunków w jeden skrypt z wykorzystaniem pętli „zawsze”, uzyskujemy taki program, jak na poniższym rysunku:



Po ułożeniu bloków uczniowie wgrzywają program do robota i sprawdzają, czy działa. Robot podąża za czarną linią, jednocześnie dokonuje korekty własnego kursu, dzięki temu może pokonywać ósemki na załączonej planszy.

Zadania dodatkowe – 10 minut

Metodą eksperymentalną można zweryfikować działanie robota w poniższych przypadkach:

- ⇒ *Jaka szerokość linii jest optymalna dla utrzymania ciągłego ruchu robota?*
- ⇒ *Co spowoduje ustawienie robota na czarnym podłożu?*
- ⇒ *Jak wykorzystać powyższy program do zaprogramowania robota typu sumo?*

Robot typu sumo ma na celu zlokalizowanie i zepchnięcie przeciwnika poza okrąg wyznaczony czarną linią, jednocześnie sam nie może poza nią wyjść.

Podsumowanie – 5 minut

Krótko podsumowujemy materiał z zajęć. Uczniowie zapisują program i wyłączają robota.

MOŻLIWE MODYFIKACJE DLA KLAS I-III I IV-VI:

W klasach I-III możliwe jest przeprowadzenie zajęć w formie zabawy. Uczniowie bawią się robotami, następnie pokazujemy im skrypty i wyjaśniamy, co oznaczają poszczególne bloki. Zachowujemy odpowiednio uproszczoną część teoretyczną, w części zadaniowej rozmawiamy z uczniami na temat sposobu wykonania zadań, realizujemy ich pomysły i tworząc program pokazujemy działanie robota.

W klasach IV-VI ułatwieniem może być wykorzystanie wcześniej przygotowanych szkieletów programów. Można wyjaśnić uczniom teorię, a następnie zadać proste zadanie (np. pracę z gotowym programem, do którego trzeba dobrać właściwe parametry, aby robot wykonał określone działanie).

Najprostsza wersja do zaprogramowania to podążanie robota za linią prostą – wykorzystanie jednego warunku. Warto poświęcić więcej czasu na wyjaśnienie zasady działania programu, można eksperymentalnie określić maksymalną prędkość robota, przy jednoczesnym utrzymaniu jego ruchu zgodnie z wytycznymi projektu. Można urządzić wyścigi robotów.

ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS LEKCJI:

Uczeń / uczennica, pracując samodzielnie albo w dwu- lub trzyosobowym zespole tworzy skrypt sterujący robotem mBot i jego czujnikiem linii. Za pomocą stworzonego skryptu steruje robotem tak, aby robot podążał za czarną linią w kształcie ósemki i powracał na linię, gdy z niej zbczy.

FIGUŁKA WIEDZY I INSPIRACJI DLA OSÓB PROWADZĄCYCH:

Robot typu „sumo” ma na celu zlokalizowanie i zepchnięcie przeciwnika poza okrąg wyznaczony czarną linią, jednocześnie sam nie może poza nią wyjść. Organizacja

zawodów szkolnych robotów sumo może być ciekawym elementem szerszej imprezy szkolnej, nie tylko dla uczestników zadania.

Inne roboty typu „line follower” można zobaczyć na stronie:

http://www.forbot.pl/forum/robots_list.php

Czujnik linii wykorzystywany jest również w scenariuszu nr 5. *Uwaga: przepaść! Poznajemy i programujemy czujnik linii.*

Scenariusz został opracowany na potrzeby projektu „MoboLab – roboty i tablety w Twojej szkole”. Celem projektu jest zwiększenie kompetencji informatycznych z zakresu programowania i wykorzystywania technologii mobilnych w uczeniu się, a także kreatywności, innowacyjności i umiejętności współpracy w zespole z wykorzystaniem TIK, uczniów / uczennic z (UCZ) z 6 szkół podnadgimnazjalnych i 4 gimnazjów Wołomina i Zielonki. Projekt dofinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego (Regionalny Program Operacyjny Województwa Mazowieckiego na lata 2014-2020, Oś Priorytetowa X. Edukacja dla rozwoju regionu, Działanie 10.1. Edukacja ogólna i przedszkolna, Poddziałanie 10.1.2. Edukacja ogólna w ramach ZIT).



Ten utwór jest dostępny na licencji [Creative Commons Uznanie autorstwa 4.0 Międzynarodowe](https://creativecommons.org/licenses/by/4.0/).