

MoboLab – roboty i tablety w Twojej szkole

Obszar I. „Zakoduj robota”

Scenariusze lekcji i zajęć pozalekcyjnych

SCENARIUSZ 8. UWAGA PRZESZKODA

scenariusz zajęć pozalekcyjnych

autor: Kamil Kociszewski

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

mBot, mBlock, Arduino, autonomiczny, trasa, ostrzeżenie, czujnik odległości, czujnik natężenia światła, zmienne

KRÓTKI OPIS ZAJĘĆ:

Podczas zajęć uczniowie i uczennice powtarzają wiedzę i rozwijają umiejętności nabyte na wcześniejszych lekcjach i zajęciach. Pracując w zespołach udoskonalają swój **pojazd autonomiczny**, czyli odpowiednio zaprogramowanego robota **mBot** z wykorzystaniem **trybu Arduino** dostępnego w środowisku **mBlock**. Dodają do programu sterującego robotem funkcje ostrzegające o przeszkodach oraz reagujące na otoczenie. Na zakończenie wykonują zadanie: uruchamiają robota, który porusza się i reaguje na przeszkody. Swoje programy i ich efekty (przejazd robota) prezentują pozostałym uczestnikom zajęć.

WIEDZA I UMIEJĘTNOŚCI ZDOBYTE PRZEZ UCZNIĄ / UCZENNICĘ:

- swobodnie porusza się po środowisku mBlock, wie, gdzie szukać bloków do tworzenia skryptów sterujących różnymi elementami robota,
- potrafi stworzyć prosty skrypt sterujący robotem,
- potrafi stworzyć programy z wieloma funkcjami.
- wie, czym jest zmienna i jak można ją wykorzystać,
- zna projekt Arduino, wie jak zaprogramować robota mBot w trybie specjalnym Arduino,
- potrafi zaprogramować robota tak, aby reagował na otoczenie.

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 minut (lub 2 x 45 minut)

STOPIEŃ TRUDNOŚCI/SKOMPLIKOWANIA

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

4, zadania dodatkowe na poziomie 5

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 brzęczyk oraz diody działają poprawnie,
- sprawdzić stan baterii zasilających robota,
- przygotować poprzednie programy i scenariusze dotyczące poruszanych zagadnień, tak aby ułatwić sobie prowadzenie zajęć.

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 czujnikami, brzęczykiem oraz do włączania i wyłączania diod,
- zna podstawowe pojęcia programistyczne (skrypt, program, pętla, instrukcja warunkowa),

- 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

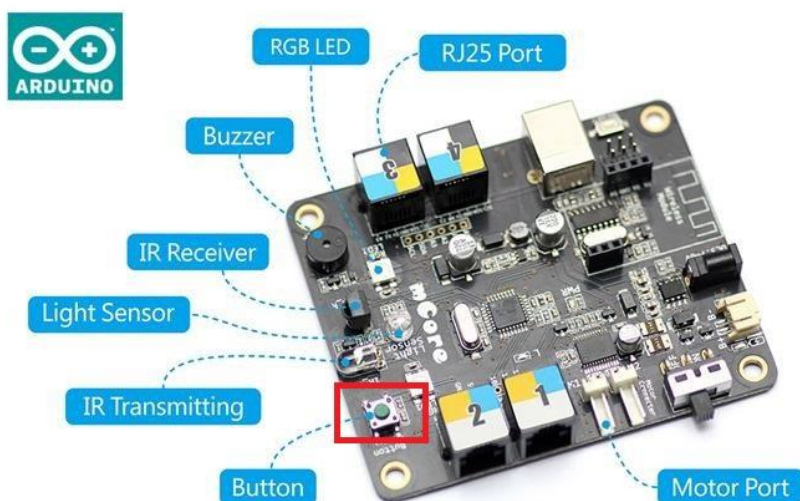
Wprowadzenie – 5 min

Cel: wprowadzenie uczniów i uczennic w tematykę zajęć i przedstawienie zadania do wykonania.

Zadaniem, które uczniowie będą wykonywać w czasie zajęć, jest zaprogramowanie robota, który będzie samodzielnie się poruszał. W przypadku napotkania przeszkody robot zmieni kierunek jazdy, a jeśli znajdzie się w ciemnym miejscu, zaświeci diody LED. W pierwszej części zajęć, będziemy tworzyć program, który umożliwia wyłączenie robota. Nauczymy się korzystać ze zmiennych, które pozwolą nam wyłączyć robota. W drugiej części zajęć dodamy unikanie przeszkód oraz reakcję robota na wjazd w ciemne miejsca.

Omówienie wykorzystania przycisku na płytce i zmiennych – 20 minut

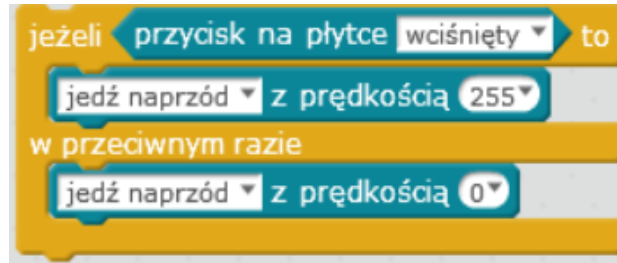
Cel: uczniowie poznają działanie przycisku na płytce oraz sposoby wykorzystania zmiennych w programie.



Na płytce (mikrokontrolerze) robota znajduje się przycisk, który na rysunku powyżej oznaczony jest jako „button” – pokazujemy uczniom umiejscowienie przycisku. Z kolei, blok odpowiadający za reagowanie na przycisk znajduje się w kategorii „Roboty”.

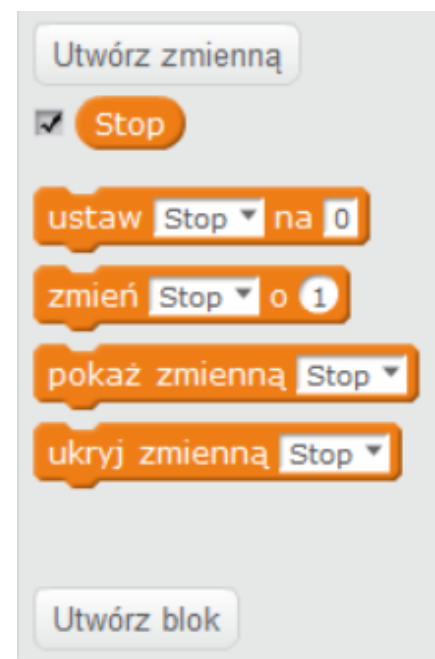
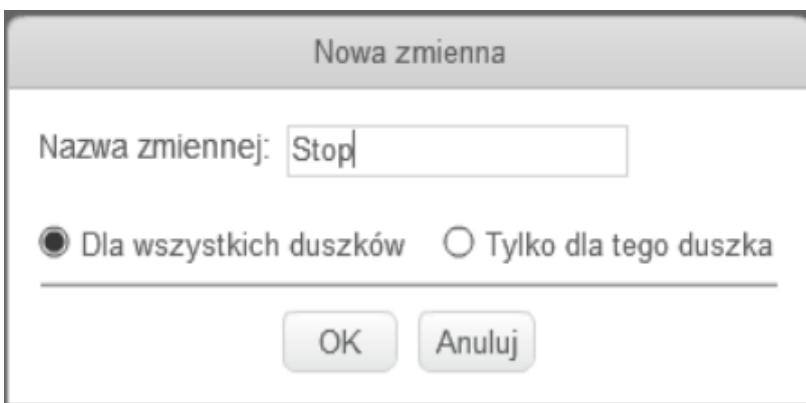
przycisk na płytce wciśnięty ▾

Przycisk na płytce można wykorzystać w instrukcjach warunkowych: np. jeżeli przycisk będzie wciśnięty, robot pojedzie do przodu, w przeciwnym razie robot będzie stał w miejscu.



Na tych zajęciach pojawią się też zmienne. Zmienna to taki element, w którym można „przechowywać” jakąś wartość. Pojęcie zmiennej mogliśmy poznać już na lekcjach matematyki. Można porównać zmienną do pudełka, w którym przechowujemy piłeczki. Możemy wrzucać piłeczki do pudełka, zabierać je, a następnie sprawdzać, ile piłeczek jest w pudełku.

Zmienne znajdziemy w kategorii **Dane i bloczki**. Po kliknięciu w tę kategorię, zobaczymy dwa polecenia: „Utwórz zmienną” oraz „Utwórz blok”. Na naszych zajęciach będziemy wykorzystywać tylko pierwsze z nich. Gdy klikniemy w „Utwórz zmienną”, wyświetli się okno, w którym możemy nadać nazwę zmiennej oraz zdecydować, dla jakich „duszków” będzie ona dostępna. W programach tworzonych na tych zajęciach skorzystamy z opcji „Dla wszystkich duszków”. Gdy utworzymy zmienną, w kategorii „Dane i bloczki” pojawiają się bloczki z poleceniami dla naszej zmiennej.



Pierwszy blok „ustaw... na...” pozwala ustawić wartość naszej zmiennej. Kolejny blok – „zmień... o...” pozwala zmienić wartość zmiennej o daną liczbę. Bloki „pokaż zmienną...” i „ukryj zmienną...” umożliwiają wyświetlenie wartości naszej zmiennej oraz ukrycie jej podczas pracy z duszkiem. W programach tworzonych na naszych zajęciach wykorzystamy przycisk na płytce oraz zmienną, którą nazwiemy „Stop” (jej wartość ustawimy na „0”). Naciśnięcie przycisku ma spowodować zmianę wartości zmiennej na „1”. Program będzie sprawdzał wartość zmiennej – jeśli wyniesie ona „0”, robot będzie się poruszał, jeśli „1”, program wyłączy silniki i robot stanie w miejscu.

Bardzo ważne jest, aby na samym początku programu ustawić wartość zmiennej na „0”, aby być pewnym, że właśnie taką będzie ona mieć wartość. Jeśli po wciśnięciu przycisku na płytce i zatrzymaniu robota chcemy, by znów zaczął jechać, wystarczy że uruchomimy program od początku przyciskiem na płytce. Spowoduje to ponowne ustawienie zmiennej na „0”.

Poruszanie się robota do przodu z wykorzystaniem zmiennej jako wyłącznika – 20 minut

Dla ułatwienia pracy uczniom pokazujemy im szkielet programu, który będzie wykorzystywany na zajęciach i krótko go omawiamy. Przypominamy o włączeniu **trybu Arduino** poznanego na poprzednich zajęciach.



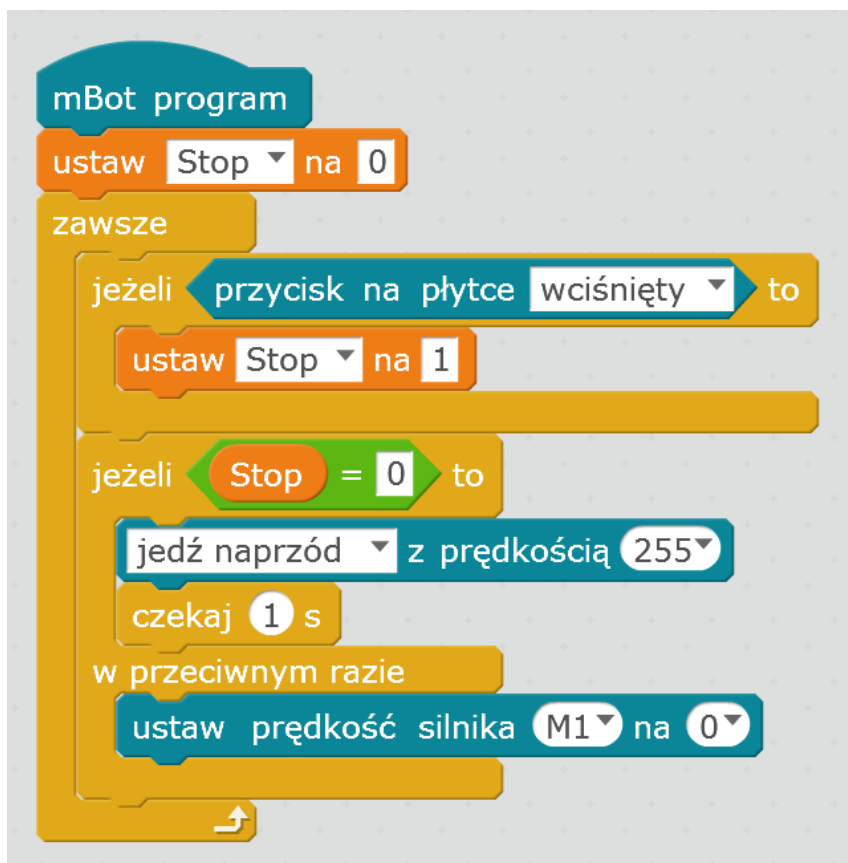
Program rozpoczynamy od bloku „mBot program”, do którego możemy dołączyć kolejne bloki. Ważne jest, aby na starcie ustawić wartość naszej zmiennej na „0”, aby mieć pewność co do jej wartości.

Następnie korzystamy z pętli „zawsze”, która pozwala wykonywać program w „kółko”

- oznacza to, że robot będzie się poruszać aż do jego wyłączenia. Następnie za pomocą instrukcji warunkowej „jeżeli” sprawdzamy, czy przycisk na płytce jest naciśnięty, i jeśli tak, ustawiamy nową wartość zmiennej, która zatrzyma działanie części odpowiedzialnej za ruch robota. Na koniec sprawdzamy wartość naszej zmiennej. Jeśli jest ona równa „0”, robot porusza się, jeśli nie, program zatrzymuje robota.

Prosimy uczniów, aby uzupełnili program o potrzebne elementy (związane ze zmiennymi oraz z ruchem robota). Następnie uczniowie wgrywają program na płytkę za pomocą polecenia „Załaduj na Arduino”, sprawdzają jego działanie i poprawiają ewentualne błędy. Jeśli zakończą zadanie wcześniej, mogą dodać funkcję jazdy do tyłu po ponownym naciśnięciu przycisku. W tym przypadku przycisk zamiast ustawiać wartość „0”, zmienia wartość o 1. Gdy wartość zmiennej jest równa „2”, pojazd porusza się do tyłu. Gotowy program może wyglądać następująco:

Wersja podstawowa programu:



Wersja rozbudowana programu:



W tym miejscu możliwa jest przerwa (kolejna część scenariusza będzie realizowana na następnej lekcji).

Część 2 – 45 minut

Przypomnienie materiału z poprzedniej części zajęć – 10 minut

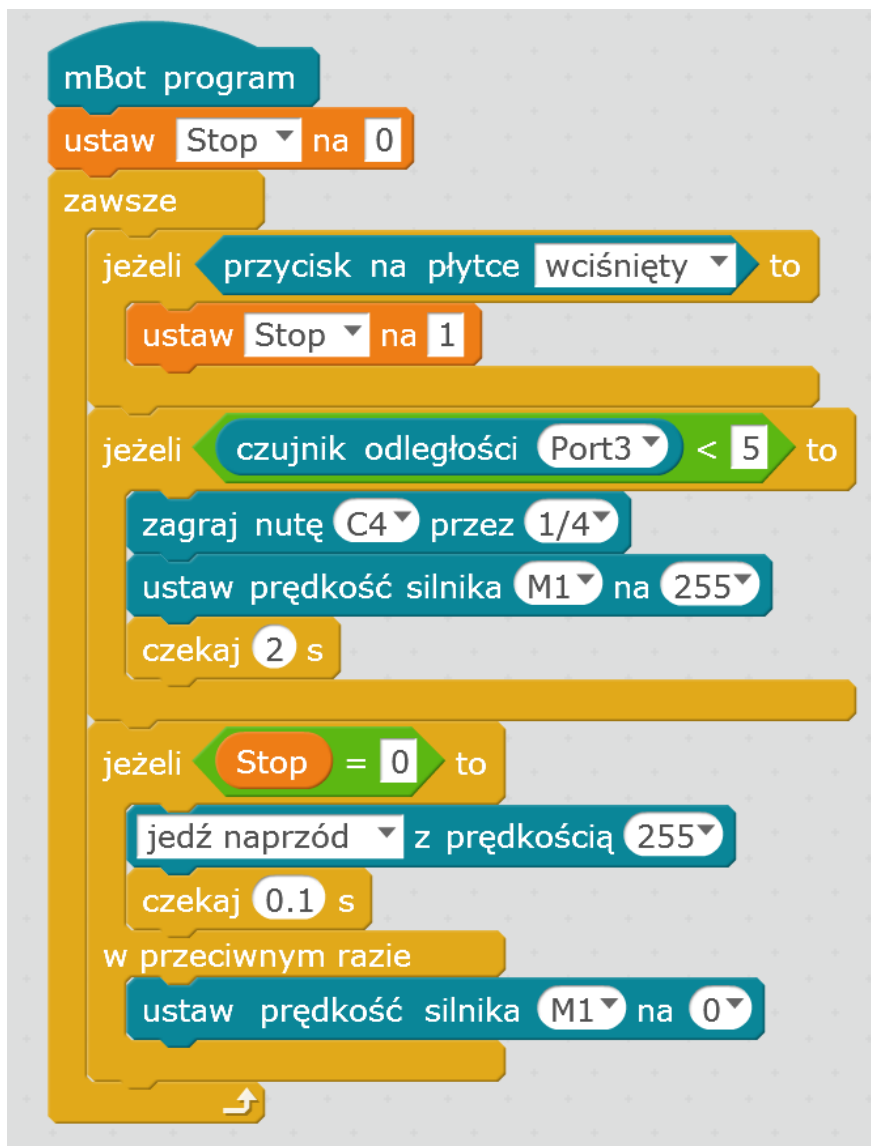
Rozpoczynamy od krótkiego przypomnienia materiału z poprzedniej części zajęć i odtworzenia powstałych skryptów.

Robot rozpoznaje przeszkodę – 15 minut

Kolejnym zadaniem do wykonania jest dodanie do stworzonych programów funkcji

omijania przeszkód. Można wykorzystać szkielet programu z poprzedniej części zajęć. Program powinien działać tak, że gdy robot zbliży się na określoną przez nas odległość, obróci się i pojedzie dalej. Warunek odległości możemy sprawdzać przed blokiem sterującym silnikami. Możemy – dla przypomnienia – pokazać program dla czujnika odległości, stworzony na wcześniejszych zajęciach. Uwaga: Przy ustawianiu parametrów czujnika odległości, należy sprawdzić, pod który port jest podłączony czujnik odległości (domyślnie jest to port 3).

Uczniowie tworzą program, sprawdzają jego działanie i poprawiają parametry (np. czas „czekania”). Zespołom, które zakończą zadanie wcześniej, można zaproponować rozszerzenie programu o sygnalizację dźwiękową przy zmianie kierunku poruszania się robota, z wykorzystaniem bloków „zagraj nutę...”. Można to także zrobić dwuetapowo: przy większej odległości od przeszkody robot wydaje ostrzeżenie dźwiękiem, a przy mniejszej zmienia kierunek. Gotowy program może wyglądać następująco (uwaga: podane wartości są przykładowe i należy je dobrać samemu):



Dodanie funkcji reagowania na wjazd w zacienione miejsce – 15 minut

Cel: uczniowie rozszerzają swoje programy wykorzystując czujnik odległości oraz diody LED.

Jeśli jakieś zespoły nadal nie dokończyły omówionych wcześniej zadań, pracują nad nimi dalej. Pozostałe grupy otrzymują nowe zadanie do wykonania. Tym razem robota należy zaprogramować w taki sposób, by zapalał diody LED, gdy wjedzie pod biurko lub znajdzie się w jakimś innym ciemnym miejscu. Kolor diody można wybrać dowolnie.

Przypominamy o konieczności skalibrowania czujnika natężenia światła. Możemy – dla przypomnienia – pokazać fragment scenariusza z wcześniejszych zajęć dotyczących czujnika natężenia światła. Tę część można dodać w dowolnym miejscu skryptu, wewnątrz pętli „zawsze”, podobnie jak w poprzednim zadaniu

Zespoły, które zakończą zadanie wcześniej, dodają funkcję zapalania diod w przypadku napotkania przeszkody i funkcję gaszenia ich w przypadku wyłączenia robota przyciskiem na płytce.

Zadanie można wykonywać metodą prób i naprawiania błędów, tak jak poprzednie zadania.

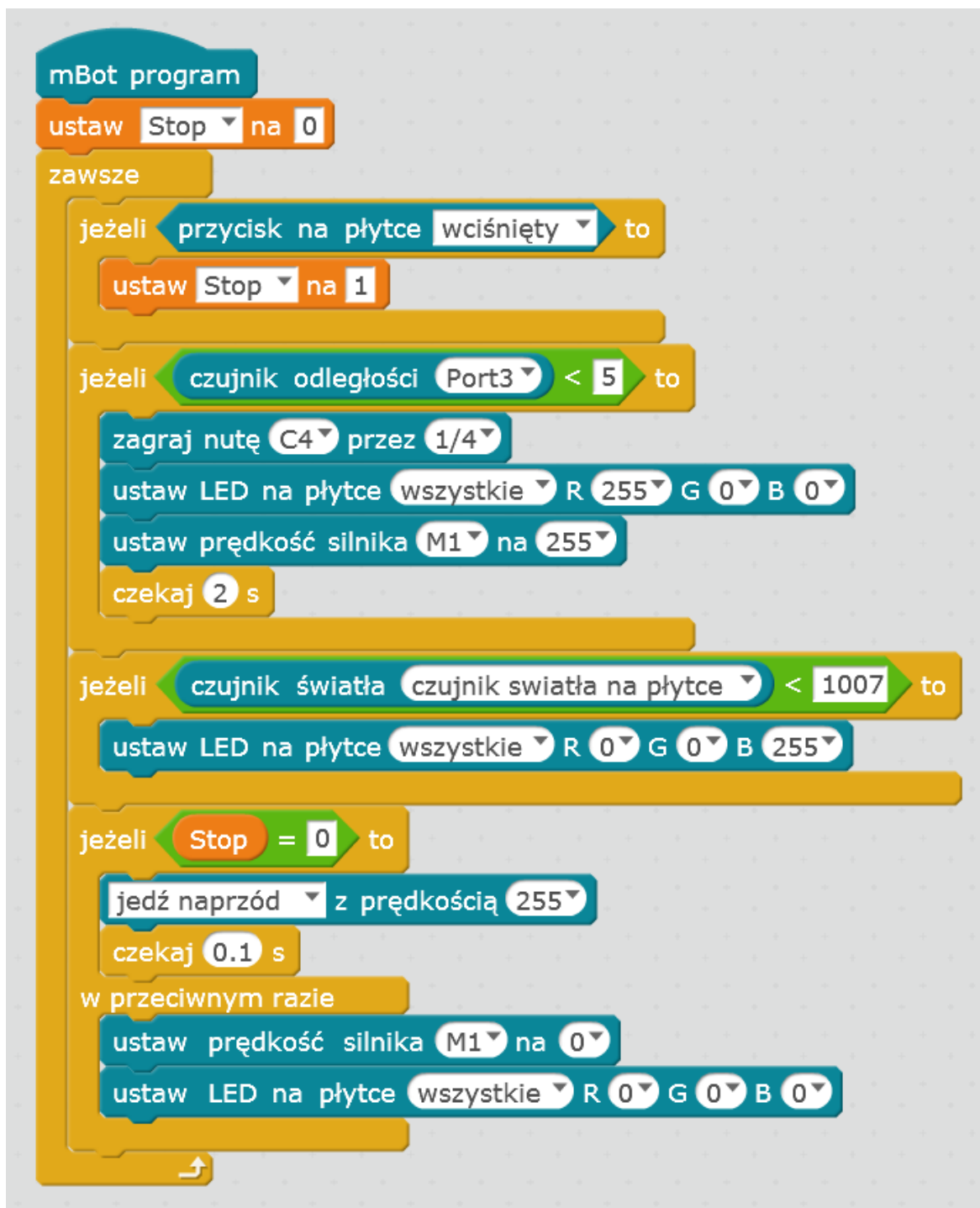
Na rysunkach niżej znajdują się przykładowe skrypty pokazujące sposoby wykonania zadania.

Wersja podstawowa programu:

The image shows a block-based programming script for an mBot. The script starts with a blue 'mBot program' block. It then sets a variable 'Stop' to 0. A yellow 'zawsze' (always) loop block contains several conditional blocks. The first is 'jeżeli przycisk na płytce wciśnięty to' (if button on board pressed then), which sets 'Stop' to 1. The second is 'jeżeli czujnik odległości Port3 < 5 to' (if distance sensor Port3 < 5 then), which plays a C4 note for 1/4 of a second, sets motor M1 speed to 255, and waits 2 seconds. The third is 'jeżeli czujnik światła czujnik światła na płytce < 1007 to' (if light sensor on board < 1007 then), which sets all LEDs to R=0, G=0, B=255. The fourth is 'jeżeli Stop = 0 to' (if Stop = 0 then), which moves forward with speed 255, waits 0.1 seconds, and then, in the 'w przeciwnym razie' (otherwise) block, sets motor M1 speed to 0. The loop ends with a yellow arrow block.

```
mBot program
ustaw Stop na 0
zawsze
  jeżeli przycisk na płytce wciśnięty to
    ustaw Stop na 1
  jeżeli czujnik odległości Port3 < 5 to
    zagraj nutę C4 przez 1/4
    ustaw prędkość silnika M1 na 255
    czekaj 2 s
  jeżeli czujnik światła czujnik światła na płytce < 1007 to
    ustaw LED na płytce wszystkie R 0 G 0 B 255
  jeżeli Stop = 0 to
    jedź naprzód z prędkością 255
    czekaj 0.1 s
  w przeciwnym razie
    ustaw prędkość silnika M1 na 0
```

Wersja ulepszona programu:



Podsumowanie – 5 minut

W tej części zajęć można kontynuować zabawę robotami, wprowadzać modyfikacje do programów, wymieniać uwagi czy omawiać trudności napotkane podczas zajęć.

Uczniowie zapisują program, wyłączają robota i porządkują stanowiska pracy.

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).

ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS LEKCJI:

Uczeń / uczennica, pracując samodzielnie albo w dwu- lub trzyosobowym zespole tworzy skrypt sterujący robotem mBot. Za pomocą stworzonego skryptu steruje robotem tak, aby robot poruszał się samodzielnie, wykrywał przeszkody i reagował na nie zapalając światła diod LED i emitując dźwięk. Zespoły prezentują przejazdy swoich robotów na forum grupy, omawiając wykorzystane funkcje w programie.

PIGUŁKA WIEDZY I INSPIRACJI DLA OSÓB PROWADZĄCYCH:

Wykorzystywane kategorie bloków:

Zdarzenia: bloki z tej kategorii służą do programowania interakcji z użytkownikami – tworzenia skryptów, które reagują na określone działania użytkownika.

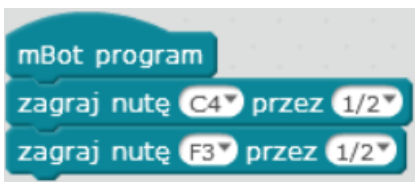
Roboty: bloki z tej kategorii służą do programowania interakcji z robotem – tworzenia skryptów, które umożliwiają sterowanie robotem i reakcję na zdarzenia oraz inicjowanie i kontrolę zdarzeń z udziałem poszczególnych elementów robota (np. czujników).

Kontrola: bloki z tej kategorii pozwalają sterować programem, na przykład dodawać do skryptu warunek, pętlę albo opóźnić wykonanie skryptu.

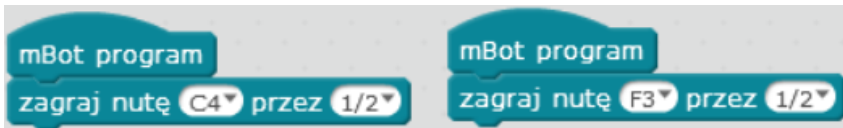
Wyrażenia: bloki z tej kategorii pozwalają wprowadzać do skryptu działania matematyczne lub wyrażenia logiczne.

Tryb Arduino pozwala stworzyć program z wykorzystaniem specjalnych bloków, a następnie wgrać go na Arduino. Aby uruchomić tryb Arduino, należy z górnego paska opcji wybrać „Edytuj” -> „Tryb Arduino”.

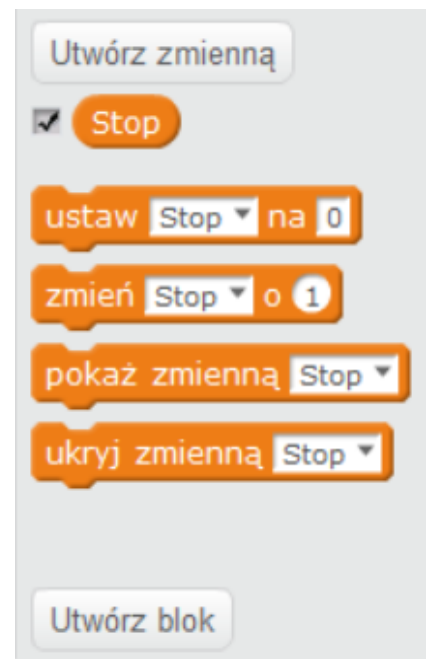
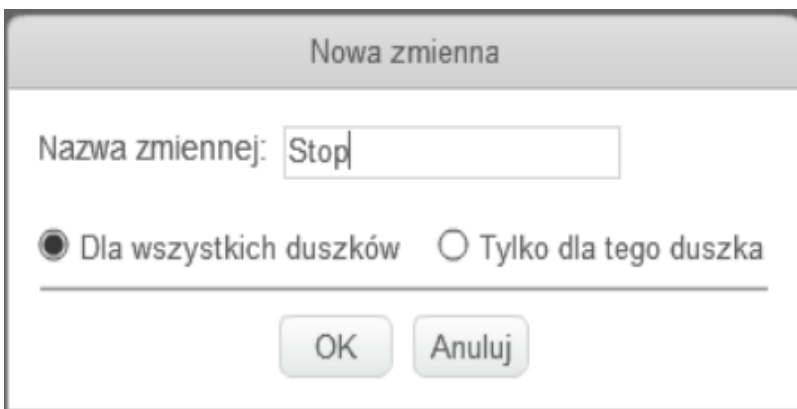
Prawidłowo stworzony program:



Nieprawidłowo stworzony program:



Zmienne znajdziemy w kategorii „Dane i bloczki”. Po kliknięciu w tę kategorię, zobaczymy dwa polecenia: „Utwórz zmienną” oraz „Utwórz blok”. Na naszych zajęciach będziemy wykorzystywać tylko pierwsze z nich. Gdy klikniemy w „Utwórz zmienną”, wyświetli się okno, w którym możemy nadać nazwę zmiennej oraz zdecydować, dla jakich „duszków” ta zmienna będzie dostępna. W programach tworzonych na tych zajęciach skorzystamy z opcji „Dla wszystkich duszków”.



Po utworzeniu zmiennej, pojawiają się kolejne polecenia w kategorii „Dane i bloczki”. Pierwszy blok „ustaw... na...” pozwala ustawić wartość naszej zmiennej. Kolejny blok – „zmień... o...” pozwala zmienić wartość zmiennej o daną liczbę. Bloki „pokaż zmienną...” i „ukryj zmienną...” umożliwiają wyświetlenie wartości naszej zmiennej oraz ukrycie jej podczas pracy z duszkiem.

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/).