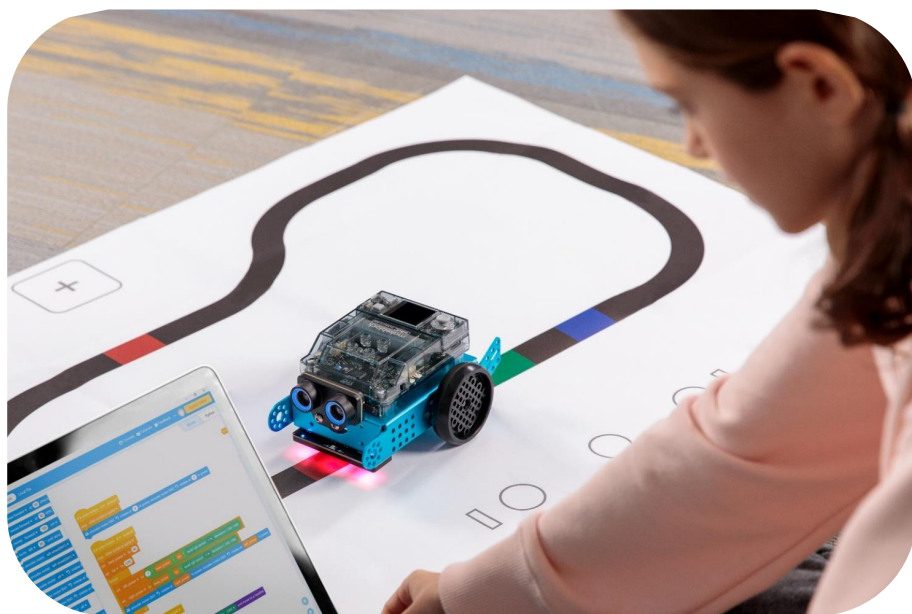


# mBot2

## Aktywności na dobry początek







# Wprowadzenie

## 1. Podejście pedagogiczne

W Makeblock Education rozwijanie wiedzy i umiejętności polega na łączeniu narzędzi cyfrowych i fizycznych, które współpracują ze sobą, zapewniając praktyczne doświadczenie edukacyjne. Uczniowie aktywnie angażują się w temat lekcji, aby rozwiązać problem lub stworzyć coś nowego.

Dzięki temu rozwijają pojęcia teoretyczne na podstawie doświadczeń praktycznych, a następnie przenoszą je na zadania praktyczne i stosują w nich, pogłębiając swoje rozumienie. Obejmuje to uczenie się na błędach poprzez szczegółową i systematyczną analizę zachodzących procesów.

Dzięki nauce przez działanie uczniowie aktywnie kształtują i wspierają własny proces edukacyjny podczas zajęć, zamiast jedynie słuchać wykładu nauczyciela.

To wprowadzenie przedstawia przegląd nowych funkcji mBot2 oraz korzyści płynących z jego wykorzystania w edukacji. Krótko pokazuje kroki potrzebne do rozpoczęcia programowania robota za pomocą mBlock5, opar-tego na blokach edytora Makeblock Education, a następnie przedstawia poszczególne Aktywności na dobry początek. Aktywności te tworzą uporządkowaną ścieżkę nauki i krok po kroku prezentują nowe funkcje. Punktem wyjścia dla zadań są odniesienia do rzeczywistego świata; omówiono w nich działanie czujników, elementów wykonawczych i myślenia komputacyjnego, a także przykładowe kody, które można łatwo zrozumieć i rozwijać w ramach każdego zadania. Aktywności zawsze sugerują wykorzystanie dodatkowych mediów cyfrowych, zastanowienie się nad zastosowaniem zdobytej wiedzy w nowych wyzwaniach oraz zakończenie aktywności odbiciem.

## 2. Wprowadzenie do mBot2

mBot2 to edukacyjny robot nowej generacji, przeznaczony do nauki informatyki i STEAM. Dzięki rozszerzonym możliwościom jest idealnym rozwiązaniem na poziomie szkoły ponadpodstawowej, ale można go również wykorzystywać w szkołach ponadpodstawowych aż do poziomu szkoły średniej II stopnia i dalej. mBot2 został zaprojektowany tak, aby uczniowie mogli realizować interaktywne i inteligentne lekcje, które angażują, bawią i odzwierciedlają rzeczywiste zastosowania najnowocześniejszych technologii.

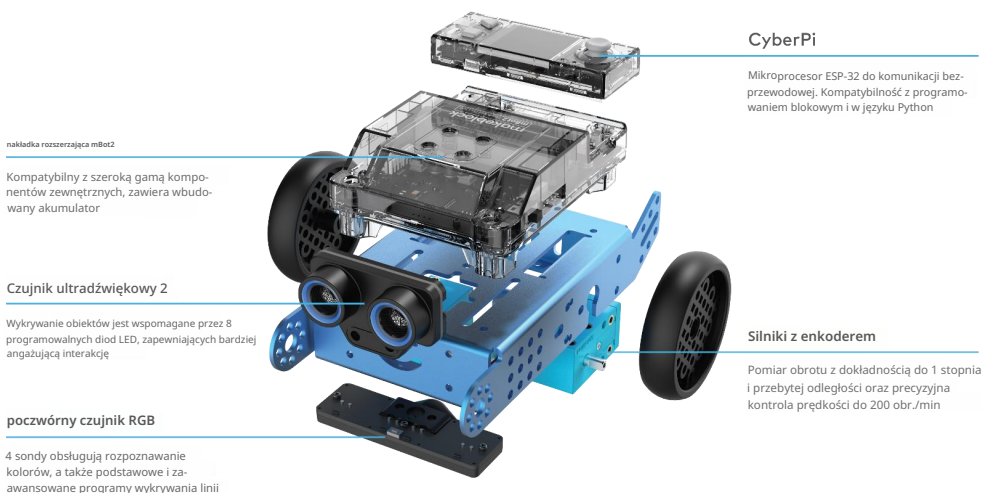
mBot2 działa dzięki CyberPi, wydajnemu i wszechstronnemu mikrokontrolerowi. Jego zintegrowane czujniki i elementy wykonawcze — takie jak mikrofon i głośnik, jednostka pomiaru inercyjnego z żyroskopem i czujnikiem przyspieszenia, czujnik światła, przyciski do obsługi menu oraz kolorowy wyświetlacz — są uzupełnione o ultradźwiękowy czujnik odległości i czujnik podążania za linią z czterema elementami RGB.

Oprócz tego zestawu czujników i elementów wykonawczych mBot2 może komunikować się przez Wi-Fi, co pozwala na szeroki zakres zastosowań w ramach tematów programowania, Robotyki, analizy danych i sztucznej inteligencji, w połączeniu z innymi przedmiotami, takimi jak matematyka, fizyka itp.

Nauczyciele mogą na przykład bezprzewodowo połączyć wiele robotów mBot w klasie, aby utworzyć lokalną sieć robotów, które komunikują się ze sobą, udostępniają informacje i wykonują zadania. Mogą również używać samodzielnego urządzenia CyberPi do komunikacji z mBot2, tworząc inteligentny ekosystem lub zabawny pilot zdalnego sterowania. Po połączeniu z Internetem mBot2 może korzystać z zaawansowanych funkcji, takich jak rozpoznawanie mowy, lub komunikować się z chmurą w celu uzyskiwania informacji.

Czujniki i elementy wykonawcze charakterystyczne dla mBot2 zostaną przedstawione w kolejnych podsekcjach, a pełny ich zakres będzie częścią Aktywności na dobry początek. Ich przegląd znajdziesz również na końcu tego dokumentu.

## Oto przegląd różnych elementów, z których składa się mBot2:

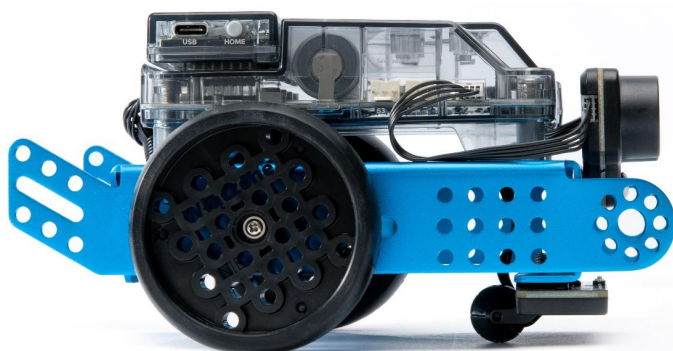


## 2.1 Silniki z enkoderem

Silniki mBot2 są wyposażone w enkodery optyczne, które umożliwiają bardzo precyzyjne sterowanie. Dzięki temu uczniowie mogą precyzyjnie sterować obrotem, prędkością i położeniem kół oraz robota. Ponadto silniki mogą być również używane jako serwomechanizmy, a nawet jako pokręta, przekazując dane do systemu, podobnie jak czujnik.

Dzięki tym funkcjom lekcje mogą być bardziej realistyczne i edukacyjne w porównaniu z robotami, które oferują mniejszą kontrolę nad silnikami.

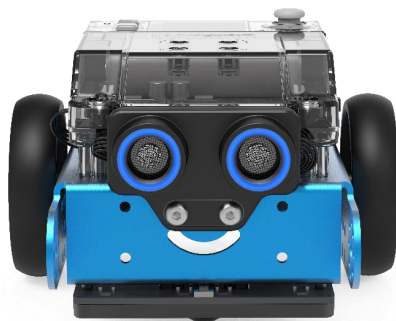
Aktywności mogą właściwie integrować pojęcia matematyczne, na przykład poprzez pokonywanie precyzyjnie określonych odległości, obliczanie dokładnych obrotów, a nawet mapowanie tras przez labirynt i przysyłanie wyników z powrotem do komputera.



Lekcja 1 z Aktywności na dobry początek wprowadza różne bloki programistyczne do silników z enkoderem w środowisku programistycznym mBlock, a także zawiera proste zadanie, które pozwala uczniom samodzielnie odkrywać ich możliwości.

## 2.2 Czujnik ultradźwiękowy i poczwórny czujnik RGB

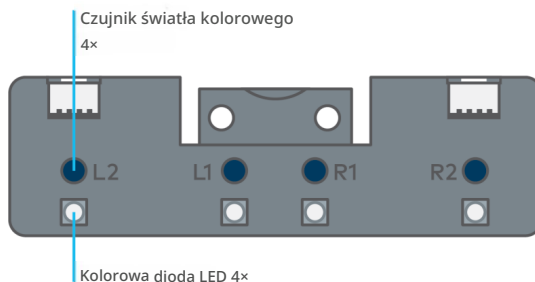
Czujnik ultradźwiękowy dołączony do mBot2 może wykonywać dokładniejsze i bardziej spójne pomiary niż jego poprzednicy. Ponadto wyposażono go w niebieskie diody LED, które umożliwiają nowe sposoby interakcji z robotem. Uczniowie mogą używać tych dodatkowych świateł do wyświetlania reakcji.



mBot2 integruje kolejny zaawansowany komponent: poczwórny czujnik RGB. Czujnik ten potrafi nie tylko rozpoznawać kolory, lecz także śledzić linie, aby pomagać robotowi mBot2 podążać wyznaczonymi trasami oraz wykrywać skrzyżowania lub zakręty o  $90^\circ$  — a wszystko to jednocześnie.

Podstawowe funkcje można realizować przy niewielkiej wiedzy wstępnej, jednak ze względu na zaawansowane możliwości czujnika ważne jest poznanie najlepszych praktyk, aby w pełni go wykorzystać.

Zamiast używać 2 czujników do śledzenia pojedynczej linii przed robotem, poczwórny czujnik RGB ma 4 czujniki, które pozwalają mu rozpoznawać szerszy zakres sytuacji: podczas podążania trasą czujnik może wykrywać skrzyżowania po lewej stronie, po prawej stronie lub nawet po obu stronach, dzięki czemu mBot2 może poruszać się po bardziej złożonych mapach niż inne roboty. A ponieważ każdy z czterech czujników jest osobnym czujnikiem koloru, kolorowe oznaczenia mogą wskazywać położenie na mapie — robot „wie”, gdzie się znajduje.



Ten szerszy zakres wykrywania można wykorzystać na mapach z siecią linii, po których wiele robotów mBot2 przemieszcza się między różnymi stacjami, jak w „Inteligentnym magazynie”, albo do symulowania dróg w mieście, z różnymi sygnalizatorami i zasadami ruchu drogowego.

Wcześniejsze czujniki podążania za linią wykorzystywały światło podczerwone, które jest niewidoczne dla ludzkiego oka. Czujniki te działały jednak w oparciu o jedną długość fali, podczas gdy nowy czujnik jednocześnie analizuje trzy różne długości fal, podobnie jak ludzkie oko: czerwoną, zieloną i niebieską. Połączenie natężenia tych długości fal jest interpretowane przez nasz mózg jako postrzeganie kolorów. Czujnik działa podobnie, a oprócz rozpoznawania kolorów próbuje także odróżnić trasę od tła podczas zadań związanych z podążaniem za linią.

Na dostarczonej mapie, wewnątrz trasy, znajdują się dodatkowe znaczniki kolorów uruchamiające dodatkowe działania, jeśli zostaną odpowiednio zaprogramowane. Aby czujnik rozpoznawał te kolory jako część linii, należy go skalibrować na najjaśniejszy kolor na trasie, czyli żółty.

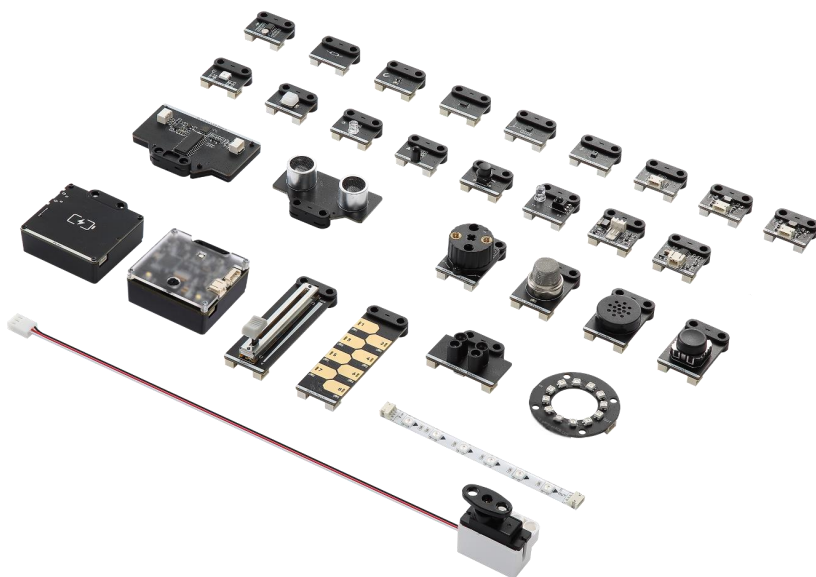
Umieść czujnik nad żółtym kodem koloru, gdy mBot2 jest włączony, i kliknij dwukrotnie mały przycisk na górnej części czujnika (zobacz ilustrację poniżej). Gdy diody LED zaczną migać, przesuwaj czujnik po kolorze i białym tle, aż miganie ustanie (2–3 sek.). Od tej chwili kolor będzie uznawany za część trasy (szczegółowe omówienie znajduje się w dodatkowych informacjach dotyczących poczwórnego czujnika RGB w różnych środowiskach oraz w lekcji 5).



Lekcja 5 z Aktywności na dobry początek wyjaśnia działanie różnych bloków dostępnych w mBlock dla poczwórnego czujnika RGB, a uczniowie mogą również przećwiczyć zdobytą wiedzę w ramach prostego zadania.



Niektóre przykłady czujników i elementów wykonawczych to: inteligentna kamera, czujnik wielodotykowy, suwak, czujnik temperatury itp. Wszystkie te czujniki zapewniają nauczycielom szeroki wybór przy tworzeniu scenariuszy z życia codziennego i dostosowywaniu ich do potrzeb nauczania i programu nauczania, a także zainteresowań uczniów.



1

### 3. Programowanie mBot2

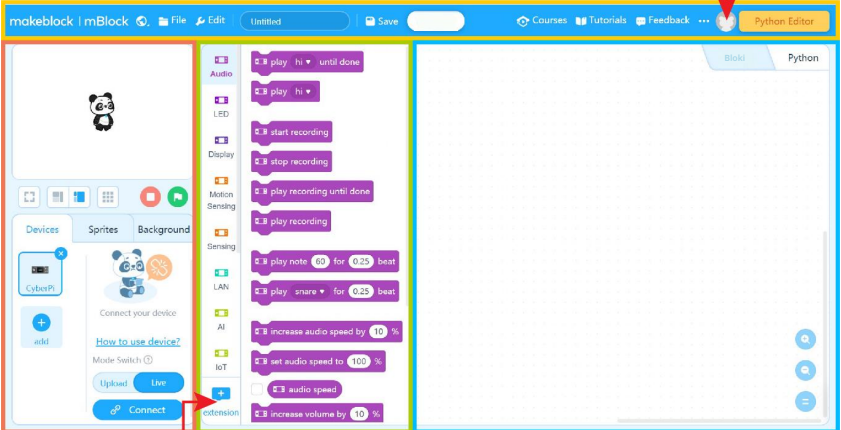
mBlock to platforma do programowania dla mBot2, zaprojektowana z myślą o zapewnieniu uczniom lepszych doświadczeń edukacyjnych i ciągłej ścieżki rozwoju. Dzięki rozszerzeniom w mBlock5 nauczyciele mogą z łatwością włączać do swoich lekcji niektóre z najnowszych i najbardziej znaczących technologii, takich jak Internet rzeczy czy sztuczna inteligencja. Ponadto, dzięki integracji programowania blokowego i języka Python, mBlock5 oferuje uczniom ścieżkę nauki umożliwiającą rozwijanie umiejętności obliczeniowych od poziomu podstawowego do profesjonalnego.

mBlock5 może być używany jako zainstalowane oprogramowanie na komputerach, laptopach i urządzeniach mobilnych lub w przeglądarce internetowej. Jest kompatybilny z różnymi systemami operacyjnymi, takimi jak Windows, Mac, Linux, Chromebook, iOS i Android. Jako oprogramowanie open source mBlock5 umożliwia tworzenie nowych rozszerzeń do oprogramowania i sprzętu, co oznacza, że nauczyciele mogą dostosowywać narzędzia do programowania do swoich potrzeb. Użytkownicy mogą również wyszukiwać projekty w społeczności Math Makeblock oraz udostępniać w niej własne projekty.

Rozpoczęcie programowania w mBlock jest tak proste jak przeciąganie i upuszczanie bloków. Interfejs mBlock zawiera następujące elementy:

Skonfiguruj konto, aby uzyskać dostęp do usług chmurowych

**Pasek menu**  
Wybierz język, otwórz lub zapisz plik, znajdź przykładowe projekty itd.



**Obszar sceny**  
Wyświetl scenę projektu, dostosuj duziki i tła, połącz urządzenia.

**Obszar bloków**  
Biblioteka bloków kodu, posortowane według kategorii oznaczonych kodem koloru.

**Obszar skryptu**  
Przeciągnij bloki programistyczne do tego obszaru i ułóż je, aby sterować sceną lub podłączonymi urządzeniami.

Centrum rozszerzeń

Szczegółowe informacje o charakterystyce i zastosowaniu mBlock można znaleźć na stronie <https://education.makeblock.com/help/category/mblock-block-based/>

#### 4. Zarys aktywności

Aktywności na dobry początek z mBot2 zostały opracowane z myślą o uczniach w wieku od 11 do 14 lat oraz ich nauczycielach. Ogólnie rzecz biorąc, lekcje zaprojektowano tak, aby były przystępne, mimo różnego i rosnącego poziomu złożoności.

Każda lekcja przedstawia inną, interesującą funkcję mBot2 oraz kilka podstawowych koncepcji programowania. Wiedza zdobyta podczas każdej lekcji pomaga przygotować się do bardziej zaawansowanych aktywności, a ich poziom złożoności wzrasta w miarę postępów uczącego się. Na przykład ruchy robota są wprowadzane w lekcji 1 i wykorzystywane w wielu kolejnych lekcjach. Możliwości poczwórnego czujnika RGB przedstawiono w lekcji 5, a następnie wykorzystano w lekcji 8.

Poniżej krótko opisano poszczególne lekcje:

| Nazwa aktywności              | Opis                                                                                                                                                                                                                                                                                  | Najważniejsze pojęcia                                                                                                                                                                                                   |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Wyruszymy w drogę          | Uczniowie poznają mBot2 i oprogramowanie mBlock oraz uczą się precyzyjnie sterować robotem. Ta wiedza zostanie wykorzystana w większości kolejnych lekcji. Uczniowie zaprojektują również prosty labirynt i zaprogramują mBot2 tak, aby ( ręcznie ) poruszał się po nim.              | Precyzyjne ruchy i odpowiadające im bloki programistyczne.                                                                                                                                                              |
| 2. Dane z czujników           | Uczniowie będą pracować z różnymi czujnikami zintegrowanymi z mBot2; dowiedzą się, jak używać ich wraz z odpowiadającymi im blokami kodu oraz jak wyświetlać dane z czujników na zintegrowanym wyświetlaczu pełnokolorowym.                                                           | Sposób działania czujników.<br>Różne sposoby wyświetlania i wizualizowania danych na wyświetlaczu.<br>Różnice między trybem Live a trybem przesyłania w mBlock 5.                                                       |
| 3. Posłuchaj mBot2            | Uczniowie nauczą się sterować głośnikiem i mikrofonem za pomocą bloków kodu w mBlock 5. Stworzą również program, w którym mBot2 odtworzy nagrany dźwięk, jeśli podczas jazdy spełniony zostanie określony warunek.                                                                    | Synteza mowy (TTS) i rozpoznawanie mowy (zamiana mowy na tekst, STT) za pomocą wbudowanego głośnika i mikrofonu. Uruchamianie wielu zadań obok siebie.                                                                  |
| 4. Widzenie za pomocą dźwięku | Uczniowie dowiedzą się, czym są ultradźwięki i jak wykorzystuje się je w czujniku, a także stworzą program, dzięki któremu mBot2 będzie poruszać się po pętli, skręcając po wykryciu przeszkód na drodze.                                                                             | Wykrywanie przeszkody lub określanie odległości za pomocą czujnika ultradźwiękowego.<br>Korzystanie z pętli i instrukcji warunkowych, aby mBot2 poruszał się, omijając przeszkody.                                      |
| 5. Zwiedzanie                 | Uczniowie dowiedzą się, jak działa czujnik koloru i jak jest wykorzystywany w życiu codziennym, a następnie zaprogramują mBot2 tak, aby stał się autobusem wycieczkowym odwiedzającym różne punkty orientacyjne w mieście. Ta wiedza zostanie również wykorzystana na lekcjach 7 i 8. | Sposób działania (fizyka światła) czujnika koloru / czujnika podążania za linią .<br>Rozpoznawanie kolorów i linii — programowanie mBot2 tak, aby podążał za linią i wykonywał działania na podstawie wykrytego koloru. |



|                    |                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                   |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6. Ostrożna jazda  | Uczniowie dowiedzą się, jak korzystać z akcelerometru żyroskopowego mBot2 i jego bloków kodu, i zaprogramują mBot2 tak, aby dostosowywał sposób jazdy, jeśli wykryje nachylenia na drodze.                                                                                                                                     | Zasada działania żyroskopów i akcelerometrów (jako jednostek pomiaru inercyjnego, IMU). Programowanie mBot2 tak, aby dostosowywał się do warunków na drodze na podstawie danych z IMU.                                                                            |
| 7. Gra sieciowa    | Uczniowie nauczą się, jak umożliwić wielu robotom mBot2 bezprzewodową komunikację między sobą bez potrzeby korzystania z punktu dostępu Wi-Fi. Zaprogramują prostą grę, w której wiele robotów mBot2 będzie szukać koloru, a ten, który znajdzie go jako pierwszy, wygra. Ta wiedza zostanie również wykorzystana na lekcji 8. | Bezprzewodowe przysyłanie danych w sieciach ad hoc. Wymiana danych w pętlach i zdarzeń.                                                                                                                                                                           |
| 8. mBot2 do usług  | Uczyć się, jak skonfigurować połączenie Wi-Fi z mBot2, uczniowie nauczą się również korzystać z wbudowanego rozpoznawania mowy i zastosują tę wiedzę w aktywności, w której mBot2 stanie się robotem kelnerem rozmawiającym z klientami.                                                                                       | Wykorzystanie trybu infrastruktury Wi-Fi z mBot2 do rozpoznawania mowy i syntezy mowy.<br>Przenoszenie intensywnych obliczeniowo zadań, takich jak rozpoznawanie mowy, do usług chmurowych.<br>Porządkowanie kodu poprzez stosowanie „właśnych bloków” (funkcji). |
| 9. mBot2 w terenie | Na tej wyjątkowej lekcji uczniowie poznają niektóre zasady sztucznej inteligencji, korzystając z rozszerzenia Teachable Machine w mBlock5. Wykorzystają swoją wiedzę, aby odtworzyć naturalny ekosystem, w którym mBot2 będzie zachowywać się jak zwierzę.                                                                     | Poznanie uczenia maszynowego i stosowanie go wyłącznie przy lokalnym przetwarzaniu w programowaniu blokowym.<br>Ustanowienie nowego protokołu komunikacyjnego między mBot2 a komputerem.                                                                          |

W trakcie zajęć zachęca się uczniów, za zgodą nauczycieli, do dokumentowania efektów swojej nauki za pomocą nagrań wideo i ich publikowania, aby mogli czerpać dumę ze swojej pracy oraz prowadzić rozmowę o informatyce i STEAM w klasie.

## Lekcja 1:

### Ruszajmy!

Temat: STEAM

Czas trwania: 45 minut

Klasy: 5. i starsze

Poziom trudności: początkujący

#### ★ Cele lekcji

Pod koniec tej lekcji uczniowie będą potrafili:

Precyzyjnie sterować robotem mBot2.

#### ★ Wprowadzenie

Roboty to autonomiczne maszyny, które zastępują pracę człowieka; potrafią postrzegać swoje otoczenie i wykonywać programy komputerowe, aby podejmować decyzje i działania. Program komputerowy to zestaw instrukcji i warunków wykonywanych przez komputer w celu realizacji zadania. Możesz uważać, że nigdy nie widziałeś(-aś) robota ani z niego nie korzystałeś(-aś), ale jest to mało prawdopodobne. Roboty mogą przyjmować wiele różnych kształtów i mieć różne możliwości, zależnie od ich zastosowania. Na przykład roboty są obecne w gospodarstwach domowych, takie jak robot sprząający czy robot koszący trawę; można je również znaleźć w fabrykach, gdzie montują produkty. Wysłaliśmy nawet roboty na inne planety.

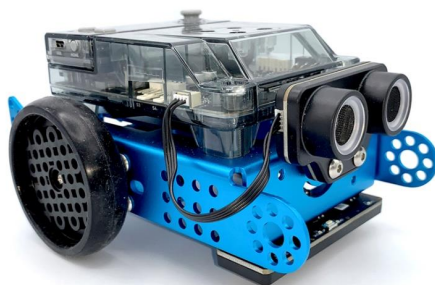
#### & Skupienie

Pod koniec lekcji uczniowie będą wiedzieć:

Jakie ruchy może wykonywać mBot2

Jakich różnych bloków programistycznych

można użyć, aby wprawić mBot2 w ruch



## □ Lista kontrolna przed lekcją

Czego potrzebujesz?

Komputer PC lub laptop (z wyjściem USB) z zainstalowanym oprogramowaniem mBlock, wersją internetową (również na Chromebooku) lub tablet z zainstalowaną aplikacją mBlock

mBot2 z urządzeniem CyberPi

Przewód USB-C lub klucz sprzętowy Bluetooth Makeblock

papier A3

Przybory do pisanie

### Plan lekcji

| Czas trwania | Treść                                                                                                                                                                                                                                                                                                                              |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 minut      | 1. Rozgrzewanie<br>Czujniki i dane w życiu codziennym.<br>Czym jest CyberPi?                                                                                                                                                                                                                                                       |
| 10 minut     | 2. Zajęcia praktyczne<br>Zapoznanie się z różnymi blokami programistycznymi robota mBot2.                                                                                                                                                                                                                                          |
| 25 minut     | 3. Wypróbowywanie<br>Programowanie własnego robota za pomocą samodzielnie stworzonego labiryntu.                                                                                                                                                                                                                                   |
| 5 minut      | 4. Podsumowanie<br>Czas na prezentację: pokaż, co zrobiłeś ze swoim robotem, w zabawnym, krótkim filmie do późniejszej dyskusji.<br>Jeśli nauczyciel na to pozwoli, udostępnij efekt końcowy w mediach społecznościowych z hashtagem #mBot2moves<br>Odbicie: Z czego jesteś najbardziej dumny? Co chcesz ulepszyć w swoim robocie? |

## 1. Rozgrzewanie (5 min)

### Krok 1: Rozgrzewka

Ten krok składa się z dwóch części:

1. Roboty w życiu codziennym
2. Poznajemy mBot2

#### 1. Roboty w życiu codziennym

Roboty spotykamy w wielu różnych miejscach . Niektórzy mają je w swoich domach, a inni pracują z nimi na co dzień . Roboty mogą przyjmować wiele różnych kształtów i mieć różne możliwości, zależnie od ich zastosowania. Czy potrafisz wymienić trzy roboty, z którymi masz do czynienia ( regularnie ) w codziennym życiu ?



Prawdopodobnie opracowano o wiele więcej robotów, których nigdy nie widziałeś .

Wyszukaj w Internecie trzy kolejne roboty, z których możesz korzystać na co dzień .

Jest duża szansa, że w późniejszym życiu zawodowym lub osobistym będziesz mieć do czynienia z robotami . W wielu różnych firmach pracują roboty . Zazwyczaj roboty te wykonują zadania, które są powtarzalne, wymagają dużej dokładności lub są niebezpieczne . W tej lekcji będziemy zajmować się głównie programowaniem robota tak, aby poruszał się z dużą dokładnością . Czy potrafisz wymienić trzy zawody lub rodzaje działalności, w których bardzo ważne jest, aby robot pracował z dużą dokładnością ?

## 2. Poznajemy mBot2

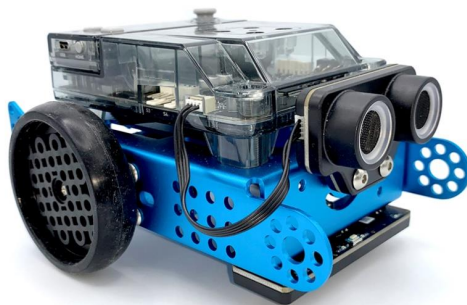
mBot2 to programowalny robot wyposażony w różne komponenty, które umożliwiają mu odbieranie informacji z otoczenia, działanie na nie i komunikowanie się z nim. mBot2 jest przeznaczony dla uczniów, którzy chcą poznać informatykę i technologie, w tym sposoby wykorzystywania robota w prawdziwym życiu.

mBot2 można programować za pomocą mBlock. W mBlock programowanie może być tak łatwe, jak przeciąganie i łączenie bloków. Jak zobaczysz w tej lekcji, możesz na przykład zaprogramować mBot2 tak, aby przejechał przez labirynt — z Twoją pomocą (nie autonomicznie). mBot2 jest wyposażony w parę specjalnych silników, które mogą rejestrować obrót osi, a tym samym prędkość oraz odległość, jaką pokonał robot.

Nazywa się je silnikami z enkoderem ze względu na wbudowany czujnik (enkoder).

Ten typ silników umożliwia kontrolowanie określonych parametrów, takich jak kąt obrotu i prędkość silników.

W kolejnych krokach nauczysz się, jak korzystać z tych silników.



## 2. Zajęcia praktyczne (10 min)

### Krok 2 : Praca praktyczna

Ten krok składa się z dwóch części:

1. Poznawanie różnych bloków programistycznych mBot2.
2. Odtwarzanie i testowanie kilku przykładów programistycznych do sterowania mBot2.

### 1. Zapoznanie się z różnymi blokami programistycznymi mBot2

Jak zauważyłeś wcześniej, roboty można wykorzystywać na wiele różnych sposobów. Jeśli zastanawiałeś się nad przeznaczeniem robota, najprawdopodobniej oznacza to, że wiesz, co robot musi zrobić, aby ci pomóc, a niektóre z tych zadań mogą wymagać wykonania z określoną precyzją.

Kiedy zaczniesz programować mBot2, zauważysz, że możesz użyć wielu różnych bloków programistycznych, aby wprawić robota w ruch. Bloki te znajdziesz w mBlock 5 w kategorii „podwozie”. Te bloki programistyczne są niebieskie.



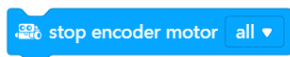
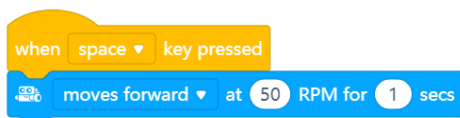
Poniżej możesz zobaczyć kilka przykładów bloków programistycznych potrzebnych do wprawienia robota w ruch. Niektóre z nich wyglądają podobnie, ale każdy z nich robi coś innego. W tej lekcji będziemy pracować w trybie na żywo. Upewnij się, że w oprogramowaniu wybrano właściwy tryb. Różnice między trybem na żywo a trybem przesyłania zostaną wyjaśnione w lekcji 2.

#### Blok kodu:



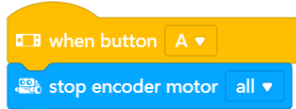
Ten blok kodu umożliwia poruszanie mBot2 do przodu, do tyłu, w lewo i w prawo z określoną prędkością obrotową kół i przez określoną liczbę sekund.

Poniższy przykład pokazuje, jak sprawić, aby mBot2 poruszał się przez dwie sekundy z prędkością 50 obr./min. Jest to przydatne na przykład wtedy, gdy mBot2 musi pchać ładunek do przodu.



Ten blok kodu umożliwia zatrzymanie robota. Ten blok może być przydatny podczas testowania. Jeśli program nie działa zgodnie z oczekiwaniami, możesz użyć tego bloku, aby natychmiast zatrzymać robota.

Na przykład możesz sprawić, aby robot zatrzymał się po naciśnięciu przycisku A na CyberPi. Aby to zrobić, użyj poniższego przykładu programistycznego.

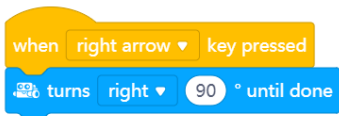
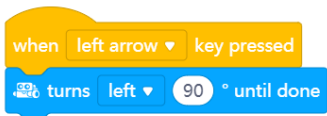


## Blok kodu:



Za pomocą tego bloku kodu możesz sprawić, że mBot2 obróci się o określoną liczbę stopni, a także możesz wybrać, czy obrót ma nastąpić w lewo, czy w prawo.

Po uruchomieniu poniższego przykładu programistycznego mBot2 można sterować za pomocą klawiszy strzałek. Po naciśnięciu prawej strzałki mBot2 skręci o 90 stopni w prawo, a po naciśnięciu lewej strzałki skręci o 90 stopni w lewo.

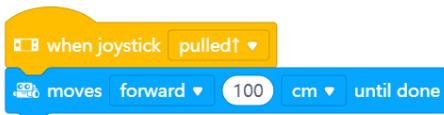


## Blok kodu:



Za pomocą tego bloku kodu możesz sprawić, że mBot2 będzie poruszać się do przodu lub do tyłu na określoną odległość.

Po użyciu poniższego przykładu programistycznego mBot2 przejedzie 100 cm do przodu. Aby rozpocząć, przechyl joystick urządzenia CyberPi do góry.



**Blok kodu:**

 encoder motor EM1  rotates 50 RPM, encoder motor EM2  rotates at 50 RPM

Możesz także niezależnie sterować silnikami mBot2 — to jeden z bloków

których możesz do tego użyć.

W poniższym przykładzie mBot2 będzie poruszać się po łuku przez 3 sekundy, zanim się zatrzyma. Zauważ, że wartość dla jednego z silników jest ujemna; dzieje się tak, ponieważ silniki są zamontowane w przeciwnych kierunkach, więc aby robot poruszał się w jednym kierunku, jeden z silników musi obracać się w przeciwną stronę. Co by się stało, gdyby oba koła obracały się z prędkością 40 obr./min?

**Blok kodu:**

 encoder motor (1) EM1  rotated angle (°)

Ponieważ silniki mBot2 mogą mierzyć swoją prędkość i obrót, możesz również odczytywać te wartości. Mierzona prędkość to prędkość obrotowa wyrażona w obrotach na minutę ( $360^\circ = 1$  obrót; w nauce częściej używa się określenia „stopień na sekundę” ( $1 \text{ obr./min} = 6^\circ \text{ na sekundę}$ )). W mBlock 5, jeśli zaznaczysz pole obok tego bloku, możesz odczytywać wartości na scenie nad pandą. Poniższy przykład programistyczny zmienia wartości kąta obrotu z powrotem na zero, a następnie nakazuje robotowi mBot2 poruszać się do przodu przez jedną sekundę. Jaką wartość odczytujesz na scenie po wykonaniu ruchu przez robota?



## 2. Odtwarzanie i testowanie kilku przykładów programistycznych do sterowania mBot2.

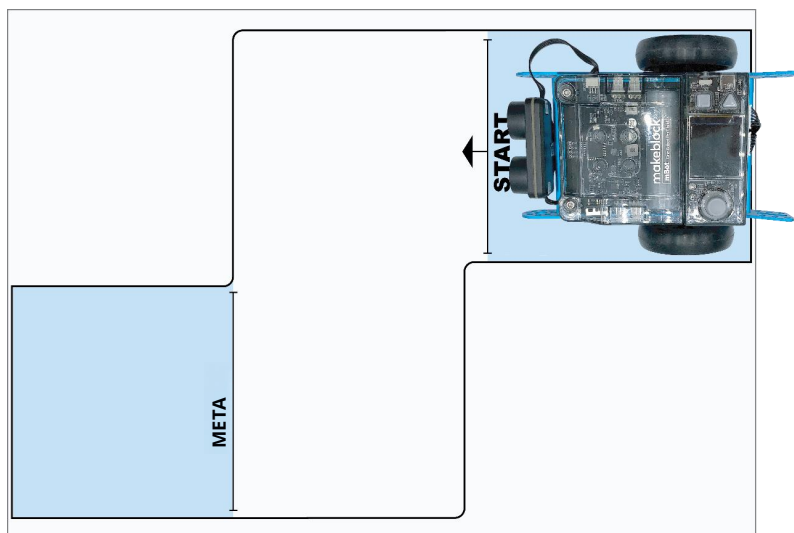
Poniżej możesz zobaczyć kilka par bloków. Wypróbuj każdą parę, umieszczając bloki obok siebie. Co się dzieje? Jakie są różnice między blokami kodu?

|   | Opcja 1 | Opcja 2 |
|---|---------|---------|
| 1 |         |         |
| 2 |         |         |
| 3 |         |         |

## 3. Wypróbowywanie (25 min)

### Krok 3: Wypróbuj

Nadszedł czas, aby wprawić mBot2 w ruch. Weź papier A3 i narysuj labirynt, przez który mBot2 ma przejechać. Nie komplikuj go zbyt i uwzględnij szerokość robota. mBot2 nie musi samodzielnie znaleźć właściwej trasy. mBot2 musi jedynie przejechać trasę, którą zaprogramujesz. Potrzebujesz inspiracji? Poniżej znajdziesz przykład.





Wykorzystaj wiedzę zdobytą w „Kroku 2” tej lekcji. Oczywiście możesz samodzielnie przeprowadzić wiele eksperymentów z różnymi przykładami programistycznymi w mBlock5.

Podczas wykonywania tego zadania pomocne będzie skorzystanie z poniższego planu krok po kroku.

|                                                            | Wyjaśnienie                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Krok 1: Co chcesz zrobić?                                  | Jaką trasą ma jechać mBot2?<br>Z jakich elementów składa się labirynt?<br>Jakie odległości musi pokonać robot?<br><br>Czy robot musi również skręcać? W którą stronę i o ile stopni?                                                                                                                                                                                                                   |
| Krok 2: Czego potrzebujesz?                                | Czego potrzebujesz oprócz mBot2?                                                                                                                                                                                                                                                                                                                                                                       |
| Krok 3: Jakich bloków kodu potrzebujesz, aby mBot2 jechał? | Jak zamierzasz sprawić, aby mBot2 jechał?<br>Jakich bloków kodu użyjesz?<br>Przygotuj krótki opis działania swojego programu ( pseudokod / język naturalny, schemat blokowy lub UML)<br><br>Jeśli potrzebujesz dodatkowych wyjaśnień, możesz omówić temat z innymi uczniami lub nauczycielem albo poszukać informacji na ten temat. W mBlock dostępna jest również pomoc dotycząca każdego bloku kodu. |
| Krok 4: Testowanie i wdrażanie                             | Czy pierwsza wersja jest gotowa? Przetestuj ją!<br>Podczas rundy testowej zapisz obszary wymagające poprawy.<br><br>Pracuj nad punktami do poprawy, aż Twój mBot2 przejedzie przez labirynt bez błędów.                                                                                                                                                                                                |

Czy uda się sprawić, aby mBot2 przejechał przez labirynt bez przeszkód? Następnie rzuć sobie wyzwanie i wzbierz trudniejszy labirynt. Możesz też poprosić jednego z kolegów z klasy o wyznaczenie dla Ciebie trasy.

#### 4. Podsumowanie (5 min)

##### Krok 4: Podsumowanie

Czy udało Ci się przeprowadzić robota przez cały labirynt bez żadnego błędu?

W tej lekcji poznaliście roboty obecne w życiu codziennym oraz mBot2. Wiecie, jak sterować ruchami mBot2 i których bloków programistycznych możecie do tego użyć.

Teraz czas na krótkie odbicie. Pomyśl samodzielnie i omów to w grupie:

Jak myślisz, co się udało?

Co można było zrobić lepiej?

Które części lekcji były dla Ciebie łatwe, a które sprawiły Ci większą trudność?

O czym chcielibyście uzyskać więcej wyjaśnień?

Kto mógłby Ci w tym pomóc?

## Lekcja 2:

# Czujniki = dane

Przedmiot: STEAM

Czas trwania: 45 minut

Klasy: 5. i starsze

Poziom trudności: początkujący

### ★ Cele lekcji

Pod koniec tej lekcji uczniowie będą potrafili:

Rozpoznawać i używać bloków kodu do obsługi czujników

Wyświetlać dane z czujników na wyświetlaczu mBot2

Tworzyć własny program komputerowy w mBlock dla mBot2 do sterowania czujnikami i wyświetlania danych na wyświetlaczu



### ★ Wprowadzenie

Robot może być zabawną zabawką, ale czy wiesz, że można go również wykorzystywać w badaniach? Robot ma kilka czujników (= zmysłów), dzięki którym może postrzegać swoje otoczenie, a także gromadzić dane. Na przykład możesz użyć mBot2 do pomiaru poziomu dźwięku i temperatury. Dane możesz odczytać na wyświetlaczu mBot2 i wykorzystać je na przykład do badania poziomu dźwięku lub temperatury w klasie.

### 🧠 Skupienie

Pod koniec tej lekcji uczniowie będą wiedzieć:

Jakie czujniki ma mBot2

Jak działają czujniki mBot2

Jaka jest różnica między trybem na żywo a trybem przesyłania

Jak wyświetlać dane z czujników na wyświetlaczu CyberPi

## □ Lista kontrolna przed lekcją

Czego potrzebujesz?

Komputer PC lub laptop (z wyjściem USB) z zainstalowanym oprogramowaniem mBlock, wersją internetową (również na Chromebooku) lub tablet z zainstalowaną aplikacją mBlock  
mBot2 z urządzeniem CyberPi  
Przewód USB-C lub klucz sprzętowy Bluetooth Makeblock

### Plan lekcji

Ta lekcja składa się z czterech kroków i trwa łącznie 45 minut.

| Czas trwania | Spis treści                                                                                                                                                                                                                                                                                                                        |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 minut      | 1. Rozgrzewanie<br>Czujniki i dane w życiu codziennym.<br>Czym jest CyberPi?                                                                                                                                                                                                                                                       |
| 10 minut     | 2. Zajęcia praktyczne<br>Zapoznanie się z czujnikami mBot2.<br>Rozszerzenie i przetestowanie kilku przykładów programistycznych dotyczących czujników.<br>Wyświetlanie danych z czujników na wyświetlaczu.<br>Różnica między trybem na żywo a trybem przesyłania.                                                                  |
| 25 minut     | 3. Wyróbowywanie<br>Pisanie własnego programu dla robota.                                                                                                                                                                                                                                                                          |
| 5 minut      | 4. Podsumowanie<br>Czas na prezentację: pokaż, co zrobiłeś ze swoim robotem, w zabawnym, krótkim filmie do późniejszej dyskusji.<br>Jeśli nauczyciel na to pozwoli, udostępnij efekt końcowy w mediach społecznościowych z hashtagem # mBot2data<br>Odbicie: Z czego jesteś najbardziej dumny? Co chcesz ulepszyć w swoim robocie? |

## 1. Rozgrzewanie (5 min)

### Krok 1: Rozgrzewka

Ten krok składa się z dwóch części:

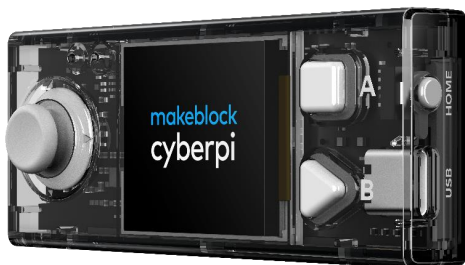
1. Czujniki i dane w życiu codziennym
2. Czym jest CyberPi?

#### 1. Czujniki i dane w życiu codziennym

Czujniki gromadzące dane można znaleźć w wielu różnych miejscach codziennego życia. Jest ich znacznie więcej, niż prawdopodobnie początkowo myślisz. Na przykład za pomocą czujników oświetlenie zewnętrzne twojego domu włącza się automatycznie, gdy na zewnątrz robi się ciemno. Albo ogrzewanie wyłącza się automatycznie, gdy w klasie robi się zbyt gorąco. Natężenie światła na zewnątrz i temperatura w klasie są rejestrowane za pomocą czujników. Czy ty i twoi koledzy z klasy potraficie podać inne przykłady?

#### 2. Czym jest CyberPi?

Aby mBot2 działał, musisz napisać program komputerowy. Robisz to w edytorze kodu mBlock. Napisany przez ciebie program komputerowy składa się z serii poleceń, które mBot2 musi wykonać.



Na mBot2 zamontowano CyberPi. CyberPi to mały, programowalny mikrokontroler. Wydajesz polecenia CyberPi za pomocą bloków kodu w mBlock.

CyberPi przekazuje te polecenia z powrotem do mBot2. Możesz wyjąć CyberPi z mBot2.

CyberPi ma wiele różnych funkcji, takich jak mikrofon, głośnik i joystick.

Na CyberPi znajduje się również wiele czujników. Przyjrzyj się poniższemu obrazowi.



## 2. Zajęcia praktyczne (10 min)

### Krok 2: Praktyczne zajęcia

Ten krok składa się z czterech części:

1. Zapoznanie się z czujnikami mBot2.
2. Rozszerzenie i przetestowanie kilku przykładów programistycznych dotyczących czujników.
3. Wyświetl dane z czujników.
4. Różnica między trybem na żywo a trybem pobierania.

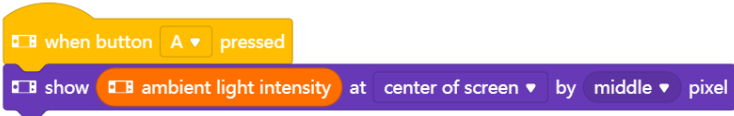


### 1. Poznajemy czujniki mBot2

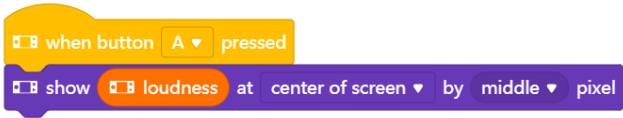
Każdy robot działa z wykorzystaniem czujników. Czujniki można porównać do zmysłów (= smaku, dotyku, węchu, słuchu i wzroku). Dzięki tym czujnikom mBot2 „widzi” swoje otoczenie. Istnieją różne rodzaje czujników, dzięki którym mBot2 może „widzieć”, takie jak:

- Czujnik światła
- Czujnik dźwięku
- Akcelerometr żyroskopowy
- poczwórny czujnik RGB
- Timer

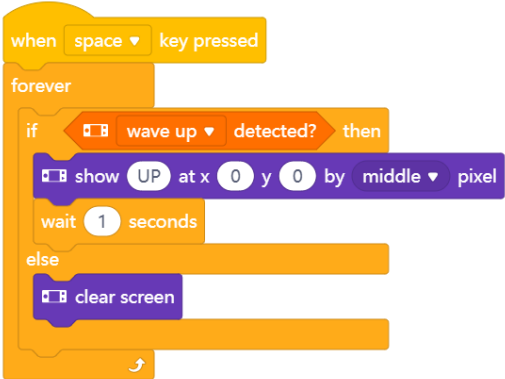
Poniższa tabela zawiera listę czujników mBot2. Każdemu czujnikowi towarzyszy krótkie wyjaśnienie i przykład programistyczny.

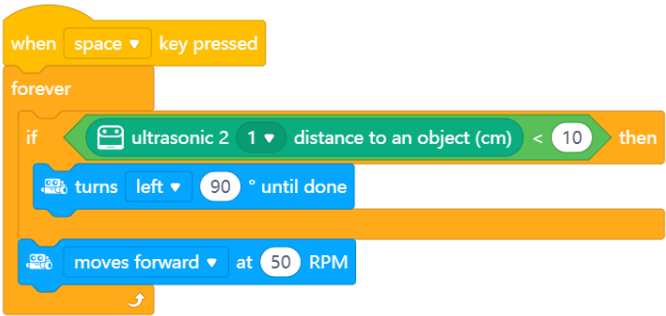
| Typ czujnika    | Co robi ten czujnik?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Czujnik światła | Światło ma określone natężenie (= energia świetlna). Czujnik światła to urządzenie, które przekształca energię świetlną w energię elektryczną.<br><br>Czujnik światła jest często używany na przykład do:<br><br>Automatycznego dostosowywania jasności wyświetlacza smartfonu do otoczenia;<br><br>Sterowania oświetleniem w domach lub automatycznego włączania reflektorów pojazdów<br><br>W poniższym przykładzie programistycznym energia świetlna otoczenia jest wyświetlana na wyświetlaczu CyberPi. Możesz rozszerzyć przykład programistyczny tak, aby mBot2 przestawał jechać, na przykład gdy energia świetlna otoczenia spadnie poniżej określonej wartości, czyli gdy zrobi się ciemno. |
|                 | Przykład programistyczny                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |



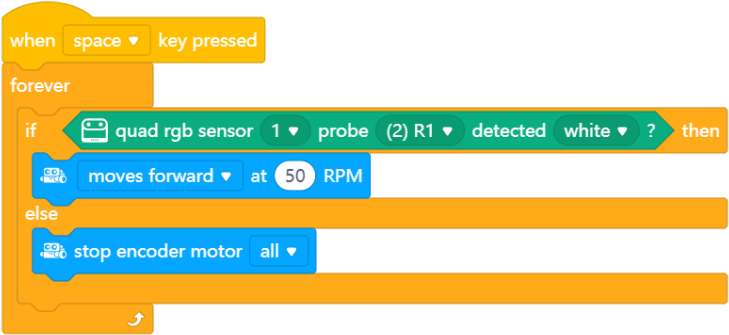
| Typ czujnika    | Co robi ten czujnik?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Czujnik dźwięku | <p>Dźwięk to drgania mechaniczne rozchodzące się w postaci fal. W powietrzu są to wahania ciśnienia i gęstości. Jeśli drgania mieszczą się w zakresie słyszalnym (od 16 do 20000 drgań na sekundę) i są wystarczająco intensywne, możemy słyszeć je jako tony lub dźwięki.</p> <p>Moc dźwięku nazywa się natężeniem dźwięku.</p> <p>Czujnik w mikrofonie przekształca dźwięk w sygnał elektryczny który można oceniać pod względem wysokości i mocy.</p> <p>Czujnik dźwięku jest często używany do następujących celów:</p> <p style="padding-left: 40px;">Połączenia telefoniczne, nagrywanie dźwięku</p> <p style="padding-left: 40px;">Systemy wspomagające sterowane głosem, na przykład w telefonie komórkowym lub inteligentnym domu</p> <p>W poniższym przykładzie programistycznym natężenie dźwięku w otoczeniu jest wyświetlane na wyświetlaczu CyberPi. Możesz rozszerzyć przykład programistyczny tak, aby mBot2 stale mierzył natężenie dźwięku podczas obchodu sali lekcyjnej i wyświetlał je na wyświetlaczu.</p> <p style="text-align: center;"><b>Przykład programistyczny</b></p>  <pre> when button A pressed   show loudness at center of screen by middle pixel   </pre> |

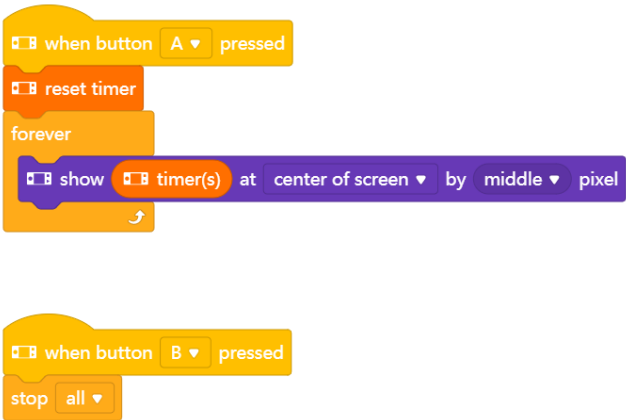


| Typ czujnika            | Co robi ten czujnik?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Żyroskop i akcelerometr | <p>Żyroskop mierzy ruchy przechylania, a dokładniej prędkość skrętu lub ruchów przechylania. Akcelerometr mierzy zmianę prędkości. Czujniki te mogą być zaimplementowane jako mikromechaniczne elementy elektroniczne. Oba czujniki dostarczają różnych informacji o położeniu w przestrzeni.</p> <p>Żyroskop jest często używany na przykład do:</p> <ul style="list-style-type: none"><li>utrzymywania stabilności statków na pełnym morzu lub utrzymywania Segwaya albo „hoverboardu” w równowadze, aby nie przewrócić się z nim zbyt szybko.</li></ul> <p>Czujniki przyspieszenia z kolei wskazują, czy smartfon jest skierowany ekranem do góry czy do dołu albo czy pojazd miał wypadek (gwałtowną, bardzo szybką zmianę prędkości) — aby następnie uruchomić poduszki powietrzne. W poniższym przykładzie programistycznym na wyświetlaczu pokazano ruch przechylenia mBot2. Możesz rozszerzyć przykład programistyczny, aby mBot2 stale mierzył i rejestrował ruch przechylenia na wyświetlaczu podczas okrążenia sali lekcyjnej.</p> <p>Więcej informacji o tym czujniku i jego możliwościach poznasz w Lekcji 6.</p> <p><b>Przykład programistyczny</b></p>  |

| Typ czujnika                  | Co robi ten czujnik?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Ultradźwiękowy czujnik</p> | <p>Dźwięk, będący drganiem w postaci zmian gęstości i ciśnienia, rozchodzi się w powietrzu ze stałą prędkością w postaci fal (prędkość dźwięku w powietrzu wynosi około 334 m / s). Im większa częstotliwość drgań, tym wyższy wydaje się dźwięk — aż do granicy, powyżej której ludzie nie mogą już go odbierać. Te bardzo wysokie częstotliwości drgań nazywają się ultradźwiękami. Ponieważ prędkość dźwięku w powietrzu jest znana, dźwięku można używać do określania odległości od obiektów; w tym celu emituje się dźwięk i mierzy czas do momentu, gdy dźwięk ten odbije się od obiektu. Ta odbita fala dźwiękowa jest również nazywana echem. Do tego celu zwykle wykorzystuje się ultradźwięki.</p> <p>Na przykład czujnik ultradźwiękowy jest często używany:</p> <ul style="list-style-type: none"> <li>do obrazowania, np. podczas ciąży (każdy piksel odpowiada pomiarowi odległości) lub</li> <li>● w sterowaniu robotami, aby zapobiegać kolizjom</li> </ul> <p>W poniższym przykładzie programistycznym czujnik ultradźwiękowy służy do zapobiegania wjechaniu mBot2 w przeszkodę. Gdy mBot2 znajduje się w odległości mniejszej niż 10 cm od przeszkody, robot skręca o 90° w lewo, a następnie po prostu jedzie dalej. Możesz rozbudować ten przykład programistyczny tak, aby mBot2 poruszał się losowo po sali lekcyjnej i nie uderzał w stoły ani krzesła.</p> <p>Więcej o tym czujniku i jego możliwościach dowiesz się w Lekcji 4.</p> |
|                               | <p style="text-align: center;"><b>Przykład programistyczny</b></p>  <pre> when space key pressed   forever loop     if ultrasonic 2 1 distance to an object (cm) &lt; 10 then       turns left 90° until done       moves forward at 50 RPM   </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |



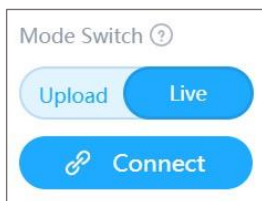
| Typ czujnika          | Co robi ten czujnik?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poczwórny czujnik RGB | <p>Poczwórny czujnik RGB składa się z czterech niezależnych czujników światła i koloru. Mierzą one natężenie światła docierającego do czujnika z czerwonego, zielonego i niebieskiego zakresu widma. Dzięki temu czujnik może wykrywać kolory obiektów znajdujących się bezpośrednio przed nim, takich jak oznaczenia na podłożu, a także umożliwia robotowi śledzenie czarnej linii w celu orientacji.</p> <p>Poczwórny czujnik RGB jest szeroko stosowany w zastosowaniach takich jak:</p> <ul style="list-style-type: none"><li>pokaż robotowi magazynowemu drogę przez magazyn (także różne kolory dla różnych tras) lub</li><li>zapewnij użycie dokładnie właściwego koloru podczas malowania</li></ul> <p>W poniższym przykładzie programistycznym mBot2 jedzie do przodu, gdy robot wykryje biały kolor. Możesz rozszerzyć przykład programistyczny tak, aby mBot2 jechał wyznaczoną trasą przez klasę, zatrzymywał się lub skręcał w zależności od koloru.</p> <p>Więcej o tym czujniku i jego możliwościach dowiesz się w Lekcji 5.</p> |
|                       | <p>Przykład programistyczny</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| Typ czujnika | Co robi ten czujnik?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Timer        | <p>Stoper mierzy czas w sekundach od momentu włączenia lub zresetowania CyberPi . Ten licznik można również wyzerować za pomocą polecenia, aby ułatwić pomiary czasu.</p> <p>Na przykład możesz użyć stopera, aby urządzenia mBot ściagały się ze sobą i dokładnie zmierzyć, jak szybko poruszają się w chwili przekroczenia linii mety.</p> <p>Poniższy przykład programistyczny pokazuje, jak ustawić i używać minutnika . Naciśnięcie klawisza „ A ” resetuje minutnik do zera, a następnie wyświetla go stale na wyświetlaczu . Naciśnięciem przycisku „ B ” można zakończyć wyświetlanie oraz cały program . Spróbuj rozszerzyć przykład programistyczny tak, aby minutnik rozpoczął odliczanie, gdy mBot2 zacznie się poruszać .</p> |
|              | <p>Przykład programistyczny</p>  <pre> when button A pressed   reset timer   forever     show timer(s) at center of screen by middle pixel  when button B pressed   stop all   </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

## 2. Testowanie i rozszerzanie przykładów programistycznych dotyczących czujników

W powyższej tabeli znajduje się przykład programistyczny dla każdego czujnika. Odtworzysz te przykłady programistyczne w mBlocku i je przetestujesz. Pomyśl o rozszerzeniu dwóch przykładów programistycznych. W powyższej tabeli znajduje się już sugestia rozszerzenia dla każdego przykładu programistycznego. Może znasz znacznie ciekawsze rozszerzenie!

Podczas pisania programu komputerowego możesz od razu testować to, co tworzysz. Robisz to, korzystając z trybu na żywo. Wybierasz tryb na żywo, przesuwając przełączanie trybu w prawo. Przyjrzyj się poniższemu obrazowi.



### 3. Wyświetlanie danych z czujników

Podczas testowania przykładów programistycznych zauważyłeś, że każdy czujnik pokazuje coś na wyświetlaczu CyberPi. Na przykład czujnik światła rejestruje natężenie światła w otoczeniu. Timer rejestruje czas, a czujnik żyroskopowy śledzi kierunek przechylenia mBot2. Wszystko, co rejestruje czujnik, nazywa się danymi.

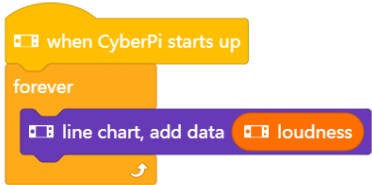
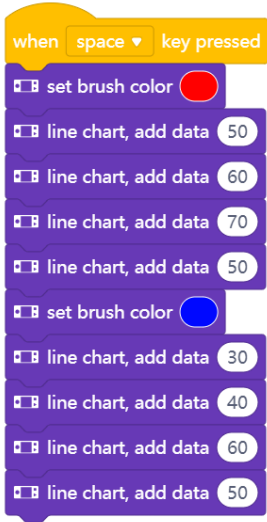
Dane z czujnika możesz wyświetlić na wyświetlaczu mBot2. W przykładach programistycznych widziałeś, że dane można wyświetlać na przykład jako liczbę lub tekst.

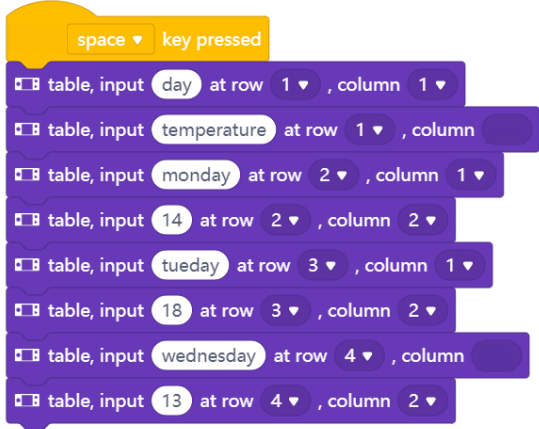


Dane z czujników możesz wyświetlać na wyświetlaczu mBot2 na różne sposoby. Możesz w tym celu używać różnych bloków kodu. Te bloki kodu znajdziesz w mBlocku, w kategorii „Wyświetlacz”.

Na poniższym diagramie widzisz trzy przykłady wyświetlania danych na wyświetlaczu CyberPi.

Odtwórz przykłady programistyczne i przetestuj je w trybie na żywo . Zobacz, co dzieje się na wyświetlaczu mBot2 .

|                          | Przykład programistyczny                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Wykres liniowy           |  <pre> when CyberPi starts up   forever loop     line chart, add data [loudness]           </pre>                                                                                                                                                                                                                                |
| Wykres liniowy ( kolor ) |  <pre> when space key pressed   set brush color [red]   line chart, add data [50]   line chart, add data [60]   line chart, add data [70]   line chart, add data [50]   set brush color [blue]   line chart, add data [30]   line chart, add data [40]   line chart, add data [60]   line chart, add data [50]           </pre> |

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tabela |  <pre> when green flag clicked   when space key pressed     table, input day at row 1, column 1     table, input temperature at row 1, column 2     table, input monday at row 2, column 1     table, input 14 at row 2, column 2     table, input tuesday at row 3, column 1     table, input 18 at row 3, column 2     table, input wednesday at row 4, column 1     table, input 13 at row 4, column 2   </pre> |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### 4. Tryb na żywo i tryb przesyłania

Musisz przesłać program komputerowy utworzony w mBlock do CyberPi.

CyberPi uruchamia mBot2 zgodnie z poleceniami zapisanymi w mBlock. Podczas pisania programu komputerowego możesz bezpośrednio testować to, co tworzysz. Robisz to, korzystając z trybu na żywo. Wybierasz tryb na żywo, przesuując przełączanie trybu w prawo. Nie wolno odłączać CyberPi od komputera podczas testowania w trybie na żywo!

CyberPi ma również tryb przesyłania. W trybie przesyłania program komputerowy jest przesyłany do CyberPi. Program jest przechowywany na CyberPi, dopóki nie zastąpisz go innym programem.

Możesz po prostu odłączyć CyberPi.

Na poniższym diagramie możesz zobaczyć różnice między trybem na żywo a trybem przesyłania.



| Tryb             | Wyjaśnienie różnic                                                                                                                                                                                                                                                                                                                      |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tryb na żywo     | <p>Programy uruchamiają się na komputerze ( = komputerze PC lub laptopie ) .</p> <p>Nie będą przechowywane na CyberPi .</p> <p>CyberPi musi pozostać połączone z komputerem .</p> <p>Edytor kodu mBlock musi być zawsze otwarty na komputerze .</p> <p>Ten tryb służy do programowania na scenie .</p>                                  |
| tryb przesyłania | <p>Programy uruchamiają się na CyberPi zamiast na komputerze i są również przechowywane na CyberPi .</p> <p>CyberPi nie musi być połączone z komputerem .</p> <p>Edytor kodu mBlock nie musi być otwarty na komputerze .</p> <p>Program przesłany do CyberPi jest zapisywany na CyberPi do momentu zastąpienia go innym programem .</p> |

### 3. Wypróbowywanie (25 min)

#### Krok 3: Wypróbuj

Wiesz już dużo o czujnikach mBot2 i o tym, jak wyświetlać dane z czujników na wyświetlaczu CyberPi . Teraz samodzielnie zbierzesz dane za pomocą czujników mBot2 i wyświetlisz je na wyświetlaczu CyberPi . Wykonasz to, jeżdżąc mBot2 po sali lekcyjnej lub szkole . To, jakie dane zbierzesz i jak pokażesz je na wyświetlaczu CyberPi, zależy od ciebie .

Wykorzystaj wiedzę zdobytą w „Kroku 2” tej lekcji. Oczywiście możesz samodzielnie przeprowadzić mnóstwo eksperymentów z różnymi możliwościami programowania w mBlock.

Rozważając to zadanie, warto skorzystać z następującego planu krok po kroku. Czy masz pomysł, co chcesz stworzyć? Jeśli tak, najpierw omów z nauczycielem, czy jest to wykonalne.



|                                                                           | Wyjaśnienie                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Krok 1: Co chcesz zrobić?                                                 | Jaką trasą ma jechać mBot2?<br>Jakie dane chcesz, aby mBot2 zbierał?                                                                                                                                                                                                                                                                                                                                                                 |
| Krok 2: Czego potrzebujesz?                                               | Czego potrzebujesz oprócz mBot2?                                                                                                                                                                                                                                                                                                                                                                                                     |
| Krok 3: Jakich bloków kodu potrzebujesz, aby mBot2 jechał?                | Jak sprawisz, że mBot2 będzie jeździł?<br>Jakie dane ma zbierać mBot2?<br>Jakich bloków kodu użyjesz?<br>Przygotuj krótki opis działania swojego programu (pseudokod / język naturalny, schemat blokowy lub UML)<br>Jeśli potrzebujesz dodatkowych wyjaśnień, możesz omówić temat z innymi uczniami lub nauczycielem albo poszukać informacji na ten temat. W mBlock dostępna jest również pomoc do każdego bloku programistycznego. |
| Krok 4: W jaki sposób chcesz wyświetlić dane z czujników na wyświetlaczu? | Jak chcesz wyświetlić dane na wyświetlaczu?<br><br>Jakich bloków kodu użyjesz?<br>Przygotuj krótki opis działania swojego programu (pseudokod / język naturalny, schemat blokowy lub UML)<br>Jeśli potrzebujesz dodatkowych wyjaśnień, możesz omówić temat z innymi uczniami lub nauczycielem albo poszukać informacji na ten temat. W mBlock dostępna jest również pomoc do każdego bloku programistycznego.                        |
| Krok 5: Testowanie i wdrażanie                                            | Czy pierwsza wersja jest gotowa? Przetestuj to! Podczas rundy testowej zapisz obszary wymagające poprawy. Pracuj nad punktami do poprawy, aż mBot2 będzie działał dokładnie tak, jak sobie wyobrażasz.<br>Udało się? Nagraj efekt końcowy i zapytaj nauczyciela, czy możesz opublikować go w mediach społecznościowych z hashtagem #mbot2.                                                                                           |

## 4. Podsumowanie (5 min)

### Krok 4: Podsumowanie

Jakie dane zebrał Twój mBot2? Czy wszystko poszło dobrze? Czy udało Ci się zobaczyć wszystkie dane na wyświetlaczu CyberPi?

W tej lekcji dowiedziałeś się o różnych czujnikach mBot2 i o tym, gdzie możesz się z nimi spotkać w życiu codziennym. Wiesz, jak programować te czujniki i jak wyświetlać dane na wyświetlaczu. Dowiedziałeś się również, jaka jest różnica między trybem wgrywania a trybem na żywo w mBlock.

Nadszedł teraz czas na krótką refleksję. Pomyśl samodzielnie i omów to w grupie:

• Jak myślisz, co się udało?

Co można było zrobić lepiej?

Które części lekcji były dla Ciebie łatwe, a które sprawiły Ci większą trudność?

• O czym chcielibyście uzyskać więcej wyjaśnień?

Kto mógłby Ci w tym pomóc?

## Lekcja 3:

### Posłuchaj mBot2

Przedmiot: STEAM

Czas trwania: 45 minut

Klasy: 5. i starsze

Poziom trudności: początkujący

#### ★ Cele lekcji

Pod koniec lekcji uczniowie będą potrafili:

Rozpoznawać i używać bloków kodu do sterowania głośnikiem i mikrofonem

Tworzyć własny program komputerowy w mBlock, aby mBot2 nagrywał dźwięk i go odtwarzał

Uruchamiać wiele zadań programistycznych obok siebie w programie komputerowym

#### ★ Wprowadzenie

Roboty często potrafią mówić. Weźmy na przykład robota odkurzającego, który za pomocą komunikatu głosowego informuje o niskim poziomie baterii. Albo robota-zabawkę, który śpiewa piosenkę. mBot2 również potrafi odtwarzać i nagrywać dźwięk. Ma do tego mikrofon i głośnik. Za pomocą mikrofonu mBot2 nagrywa dźwięk, a głośnik odtwarza nagrany dźwięk.

#### 🌀 Skupienie

Pod koniec tej lekcji uczniowie będą wiedzieć:

Jak korzystać z głośnika w programowaniu i robotyce

Jak nagrywać i odtwarzać dźwięk

Jak uruchamiać wiele zadań programistycznych obok siebie

## Lista kontrolna przed lekcją

Czego potrzebujesz?

Komputer stacjonarny lub laptop (z wyjściem USB) z zainstalowanym oprogramowaniem mBlock, wersją internetową (także na Chromebooku) lub tablet z zainstalowaną aplikacją mBlock  
mBot2 z urządzeniem CyberPi

- Kabel USB-C lub klucz sprzętowy Bluetooth Makeblock

## # Plan lekcji

Ta lekcja składa się z czterech kroków i trwa łącznie 45 minut.

| Czas trwania | Spis treści                                                                                                                                                                                                                                                                                                                           |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 minut      | 1. Rozgrzewanie<br>Głośniki w codziennym życiu<br>Jak działa głośnik?                                                                                                                                                                                                                                                                 |
| 10 minut     | 2. Zajęcia praktyczne<br>Zapoznanie się z różnymi blokami kodu głośnika i mikrofonu.<br><br>Odtwórz, przetestuj i rozbuduj kilka przykładów programistycznych dotyczących głośnika i mikrofonu.                                                                                                                                       |
| 25 minut     | 3. Wypróbowywanie<br>Pisanie własnego programu dla robota.                                                                                                                                                                                                                                                                            |
| 5 minut      | 4. Podsumowanie<br>Czas na prezentację: pokaż, co zrobiłeś ze swoim robotem, w zabawnym, krótkim filmie do późniejszej dyskusji.<br>Jeśli nauczyciel na to pozwoli, udostępnij efekt końcowy w mediach społecznościowych z hasztagiem #mBot2speaker<br>Odbicie: Z czego jesteś najbardziej dumny? Co chcesz ulepszyć w swoim robocie? |

## 1. Rozgrzewanie (5 min)

### Krok 1: Rozgrzewka

Ten krok składa się z dwóch części:

1. Głośniki w codziennym życiu
2. Jak działa głośnik?

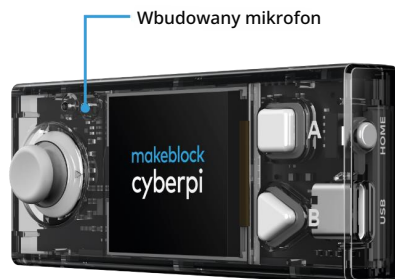
#### 1. Głośniki w codziennym życiu

Głośniki otaczają nas w codziennym życiu. Duże zestawy głośnikowe można znaleźć w teatrach, salach koncertowych i salach widowiskowych. Głośniki te służą do wzmacniania dźwięku dla dużej grupy osób. Mniejsze zestawy głośnikowe znajdują się w urządzeniach takich jak telewizory, instrumenty muzyczne i smartfony. Czy ty i koledzy z grupy potraficie wymyślić więcej zastosowań?

#### 2. Jak działa głośnik?

Zadaniem głośnika jest wytwarzanie dźwięku. W tym celu głośniki przekształcają fale elektromagnetyczne w fale dźwiękowe. Oprócz swojego głównego zadania głośnik może również przekształcać fale dźwiękowe z powrotem w fale elektromagnetyczne. Mikrofony to urządzenia działające na tej samej zasadzie, ale zoptymalizowane pod kątem przekształcania fal dźwiękowych w sygnały elektryczne.

W mBot2 znajdują się głośnik i mikrofon. Mikrofon znajduje się w lewym górnym rogu CyberPi. Głośnik znajduje się z przodu CyberPi. Za pomocą mikrofonu możesz nagrywać dźwięki. Dźwięki te można później odtwarzać za pomocą głośnika.



Korzystając z mikrofonu i głośnika w połączeniu z pozostałymi czujnikami mBot2, można na przykład odtworzyć własny nagrany tekst o określonej porze. Pomyśl na przykład o autobusie miejskim, który informuje, do którego przystanku prawie dojeżdżasz. Istnieje niezliczona liczba innych możliwości wykorzystania mikrofonu i głośnika. Wspólnie z kolegami i koleżankami z grupy zastanówcie się nad innymi przykładami, w których mBot2 może skutecznie wykorzystać głośnik i mikrofon.

## 2. Zajęcia praktyczne (10 min)

### Krok 2 : Praca praktyczna

Ten krok składa się z dwóch części:

1. Zapoznanie się z różnymi blokami kodu dla głośnika i mikrofonu.
2. Odtwórz, przetestuj i rozbuduj kilka przykładów programistycznych dotyczących głośnika.

1. Poznanie różnych bloków kodu do głośnika i mikrofonu W mBlock 5 dostępnych jest kilka bloków kodu, których można używać do programowania głośnika i mikrofonu. Te bloki kodu znajdziesz w kategorii „Audio” w obszarze bloków w mBlock.

Te bloki kodu są fioletowe.

W poniższej tabeli zobaczysz niektóre z tych bloków kodu służących do programowania głośnika i mikrofonu.



Audio

#### Blok kodu:

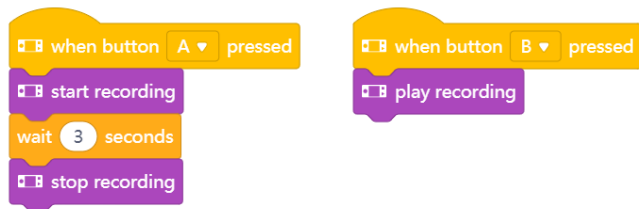
 start recording

 stop recording

 play recording

Podczas nagrywania i odtwarzania dźwięku zawsze w rzeczywistości używa się trzech bloków kodu. Za pomocą bloku kodu „Rozpocznij nagrywanie” polecasz CyberPi rozpoczęcie nagrywania dźwięku. Sygnał elektryczny z mikrofonu jest przechowywany w pamięci CyberPi, a nagrywanie automatycznie zatrzymuje się po 10 sekundach. Możesz także wskazać, kiedy nagrywanie powinno się zatrzymać. Służy do tego blok kodu „Zatrzymaj nagrywanie”. Nagrany dźwięk jest tymczasowo przechowywany w pamięci CyberPi. Wyłączasz mBot2? Wtedy nagrany dźwięk zostaje utracony. Odtwarzaniem nagranego dźwięku steruje blok kodu „odtwórz nagranie”.

W poniższym przykładzie programistycznym, gdy naciśniesz przycisk A, CyberPi rozpoczyna nagrywanie. Po 3 sekundach nagrywanie się zatrzymuje. Gdy naciśniesz przycisk B, usłyszysz to, co zostało nagrane. Dzięki zastosowaniu dwóch bloków zdarzeń dwa zadania programistyczne mogą działać niezależnie od siebie, bez potrzeby używania pętli głównej.

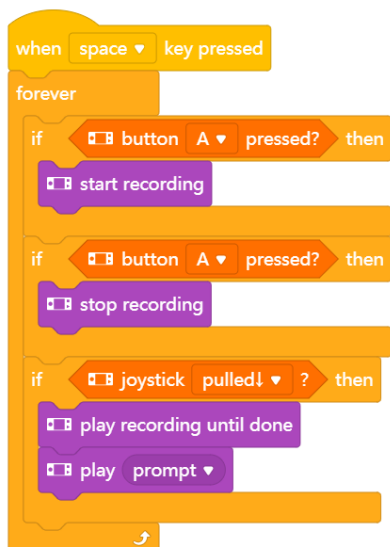


Blok kodu:



Właśnie nauczyłeś się, jak odtwarzać nagrany dźwięk . Za pomocą tego bloku kodu upewniasz się, że dźwięk nagrany przez mikrofon zostanie odtworzony do końca . Oznacza to, że wykonanie kolejnych bloków jest wstrzymane do zakończenia tego bloku . Po zakończeniu odtwarzania dźwięku możesz sprawić, aby mBot2 zrobił coś innego . Na przykład możesz odtworzyć określony dźwięk, aby wiedzieć, że nagrywanie zostało zakończone .

W przykładzie programistycznym po prawej stronie CyberPi rozpoczyna nagrywanie po naciśnięciu przycisku A. Jeśli naciśniesz przycisk B, nagrywanie zostanie zatrzymane . Jeśli poruszysz joystickiem w dół, usłyszysz to, co nagrał mBot2 . Po zakończeniu całego nagrania usłyszysz dźwięk „monitu” . Bloki kodu w tym przykładzie są umieszczone w tak zwanej strukturze pętli . Dzięki temu dane wejściowe z przycisków i joysticka są stale monitorowane .





2. Odtwarzanie, testowanie i rozwijanie niektórych przykładów programistycznych prezentowanych przez głośnik. W powyższej tabeli każdemu blokowi kodu głośnika towarzyszy przykład programistyczny.

Odtworzysz te przykłady programistyczne w mBlocku i je przetestujesz. Pomyśl o jednym rozszerzeniu przykładu programistycznego. Przetestuj przykłady programistyczne i własne rozszerzenie w trybie na żywo.

### 3. Wypróbowywanie (25 min)

#### Krok 3: Wypróbuj

Wiesz już dużo o głośniku i mikrofonie w mBot2. Teraz będziesz pracować nad nimi samodzielnie. Użyjesz także czujnika ultradźwiękowego. Ten czujnik potrafi wykryć obiekt z odległości. Więcej o czujniku ultradźwiękowym dowiesz się w lekcji 4.

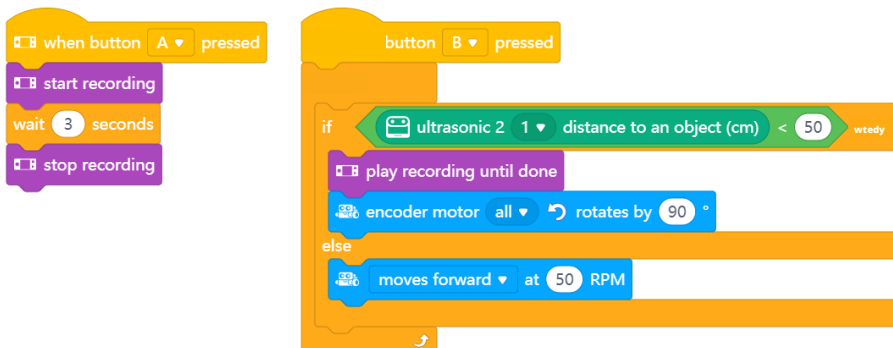
W tym zadaniu sprawisz, że mBot2 przejedzie losową trasę po klasie.

Losowa oznacza, że nie ustalasz wcześniej trasy. Jeśli mBot2 zbliży się do przeszkody, takiej jak krzesło, stół lub plecak, usłyszysz dźwięk, a mBot2 zawróci. Samodzielnie nagrasz ten dźwięk za pomocą mikrofonu. Może to być na przykład głośny sygnał dźwiękowy albo odgłos zwalniającego samochodu. Możesz także nagrać tekst albo sprawić, że mBot2 odtworzy piosenkę.

Samodzielne napisanie programu komputerowego do tego jest dość trudne! Na szczęście możesz uzyskać pomoc. Na następnej stronie zobaczysz dwa przykłady programu komputerowego. Możesz samodzielnie rozwijać i dostosowywać te programy komputerowe. Bądź kreatywny!

### Przykład 1

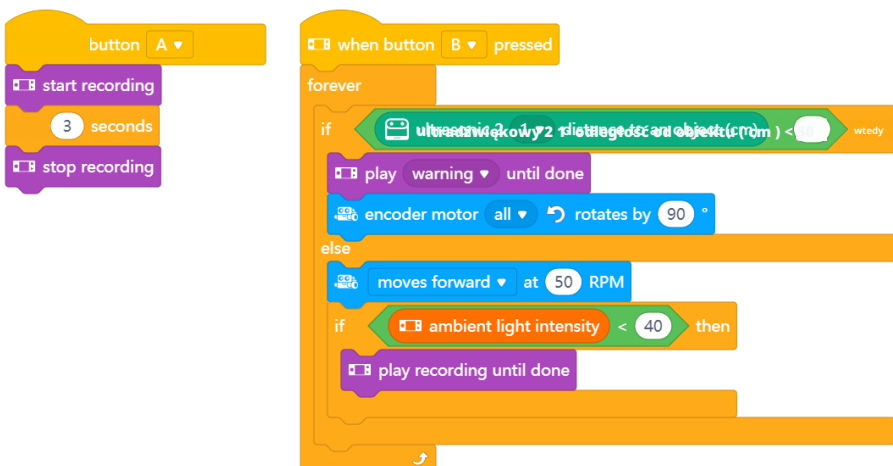
mBot2 wydaje dźwięk, gdy zbliża się do przeszkody, i zawraca. Wcześniej samodzielnie nagrałeś ten dźwięk.



### Przykład 2

mBot2 jeździ i odtwarza nagranie, gdy natężenie światła otoczenia jest niskie .

Gdy mBot2 zbliża się do przeszkody, wydaje ostrzegawczy dźwięk i zawraca .



Wykorzystaj wiedzę zdobytą w „Kroku 2” tej lekcji. Oczywiście możesz samodzielnie przeprowadzić mnóstwo eksperymentów z różnymi możliwościami programowania w mBlock.

Rozważając to zadanie, warto skorzystać z następującego planu krok po kroku. Czy masz pomysł, co chcesz stworzyć? Jeśli tak, najpierw omów z nauczycielem, czy jest to wykonalne.

|                                                                           | Wyjaśnienie                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Krok 1: Co chcesz zrobić?                                                 | Kiedy mBot2 powinien zawrócić ?<br><br>Jaki dźwięk powinien wydawać mBot2, gdy zbliża się do przeszkody ?                                                                                                                                                                                                                                                                                                    |
| Krok 2: Czego potrzebujesz?                                               | Czego potrzebujesz oprócz mBot2?                                                                                                                                                                                                                                                                                                                                                                             |
| Krok 3: Jakich bloków kodu potrzebujesz, aby mBot2 jechał?                | Jak zamierzasz sprawić, aby mBot2 jechał?<br>Jakich bloków kodu użyjesz?<br>Przygotuj krótki opis działania swojego programu ( pseudokod / język naturalny, schemat blokowy lub UML)<br><br>Jeśli potrzebujesz dodatkowych wyjaśnień, możesz omówić temat z innymi uczniami lub nauczycielem albo poszukać informacji na ten temat. W mBlock dostępna jest również pomoc do każdego bloku programistycznego. |
| Krok 4: W jaki sposób chcesz wyświetlić dane z czujników na wyświetlaczu? | Jaki dźwięk powinien wydawać mBot2 ?<br>Jakich bloków kodu użyjesz?<br>Przygotuj krótki opis działania swojego programu ( pseudokod / język naturalny, schemat blokowy lub UML)<br><br>Jeśli potrzebujesz dodatkowych wyjaśnień, możesz omówić temat z innymi uczniami lub nauczycielem albo poszukać informacji na ten temat. W mBlock dostępna jest również pomoc do każdego bloku programistycznego.      |



|                                |                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Krok 5: Testowanie i wdrażanie | <p>Czy pierwsza wersja jest gotowa? Przetestuj to! Podczas rundy testowej zapisz obszary wymagające poprawy. Pracuj nad punktami do poprawy, aż mBot2 będzie robić dokładnie to, co zamierzasz .</p> <ul style="list-style-type: none"><li>• Udało się ? Nagraj efekt końcowy i zapytaj nauczyciela, czy możesz opublikować go w mediach społecznościowych z hashtagem # mBot2</li></ul> |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### 4. Podsumowanie (5 min)

##### Krok 4: Podsumowanie

Jak poszło to zadanie? Czy mBot2 od razu zrobił to, co mieliście na myśli?

W tej lekcji nauczyliście się, jak działają głośniki i gdzie spotykacie je w życiu codziennym.

Wiedziecie, jak zaprogramować CyberPi i mBota, aby nagrywały za pomocą wbudowanego mikrofonu i odtwarzały nagrania przez głośniki. Wiedziecie również, jak uruchamiać wiele programów niezależnie od siebie.

Nadszedł teraz czas na krótką refleksję. Pomyśl samodzielnie i omów to w grupie:

Jak myślisz, co się udało?

Co można było zrobić lepiej?

Które części lekcji były dla Ciebie łatwe, a które sprawiły Ci większą trudność?

O czym chcielibyście uzyskać więcej wyjaśnień?

Kto mógłby Ci w tym pomóc?

## Lekcja 4:

### Widzenie za pomocą dźwięku

Przedmiot: STEAM

Klasy: 5. i starsze

Czas trwania: 45 minut

Poziom trudności: początkujący

#### ★ Cele lekcji

Pod koniec tej lekcji uczniowie będą wiedzieć:

Czym są ultradźwięki i jak działa czujnik wykorzystujący ultradźwięki

Zastosowania tych czujników w życiu codziennym i Robotyce

#### ★ Wprowadzenie

Autonomiczne roboty muszą być świadome swojego otoczenia podczas jazdy bez nadzoru, aby unikać kolizji z przeszkodami lub osobami. Dotyczy to również mBot2. mBot2 ma z przodu specjalny czujnik przeznaczony do tego celu. Nazywa się on czujnikiem ultradźwiękowym i umożliwia robotowi wykrywanie obiektów na jego drodze za pomocą dźwięków, których ludzie nie słyszą — stąd nazwa (ultradźwięki).

#### 🌀 Skupienie

Pod koniec tej lekcji uczniowie będą potrafili:

Używanie czujnika ultradźwiękowego do pomiaru odległości i wykrywania przeszkód

Sprawianie, aby mBot2 reagował na przeszkody i omijał je podczas jazdy

## Lista kontrolna przed lekcją

Czego potrzebujesz?

Komputer PC lub laptop (z wyjściem USB) z zainstalowanym oprogramowaniem mBlock, wersją internetową (również na Chromebooku) lub tablet z zainstalowaną aplikacją mBlock

mBot2 z urządzeniem CyberPi

Przewód USB-C lub klucz sprzętowy Bluetooth Makeblock

Małe przeszkody (minimalny rozmiar to 10x10x10 cm)

Obszar roboczy o powierzchni co najmniej 1m<sup>2</sup>, po którym mBot2 może jeździć

## # Plan lekcji

Ta lekcja składa się z czterech kroków i trwa łącznie 45 minut.

| Czas trwania | Spis treści                                                                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 minut      | 1. Rozgrzewanie<br>Czujniki ultradźwiękowe w życiu codziennym<br>Jak działa czujnik ultradźwiękowy?                                                                                                                                                                                                                                 |
| 10 minut     | 2. Zajęcia praktyczne<br>Poznanie różnych bloków kodu czujnika ultradźwiękowego.<br><br>Odtwarzanie i testowanie kilku przykładów programistycznych z użyciem czujnika ultradźwiękowego.                                                                                                                                            |
| 25 minut     | 3. Wypróbowywanie<br>Pisanie własnego programu dla robota.                                                                                                                                                                                                                                                                          |
| 5 minut      | 4. Podsumowanie<br>Czas na prezentację: pokaż, co zrobiłeś ze swoim robotem, w zabawnym, krótkim filmie do późniejszej dyskusji.<br>Jeśli nauczyciel wyrazi zgodę, udostępnij efekt końcowy w mediach społecznościowych, używając hashtagu #mBot2<br>Odbicie: Z czego jesteś najbardziej dumny? Co chcesz ulepszyć w swoim robocie? |

## 1. Rozgrzewanie (5 min)

### Krok 1: Rozgrzewka

Ten krok składa się z dwóch części:

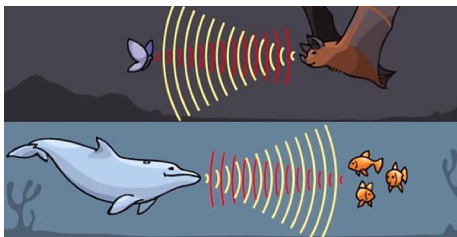
1. Czujniki ultradźwiękowe w życiu codziennym
2. Jak działa czujnik ultradźwiękowy?

#### 1. Czujniki ultradźwiękowe w życiu codziennym

Ultradźwięki to dźwięki o bardzo wysokiej częstotliwości. Tak wysokiej, że ludzie nie mogą ich usłyszeć. Za pomocą specjalnych mikrofonów dźwięk ten można nagrywać i wyświetlać na komputerze: wygląda wtedy jak drgania w postaci fali. Ponieważ ultradźwięki mają bardzo wysoką częstotliwość, przebieg fali przedstawia doliny i szczyty znajdujące się bardzo blisko siebie. Ultradźwięki mogą być przydatne. Można na przykład wykorzystać drgania ultradźwięków do czyszczenia przedmiotów albo użyć ich do wykrywania obiektów.

Delfiny i nietoperze wykorzystują ultradźwięki w ten sposób.

Emitują dźwięk ultradźwiękowy przypominający krótki krzyk. Kiedy ten dźwięk odbija się od czegoś, wychwytyują uszami powracające drgania (echo). Dzięki temu wiedzą na przykład, gdzie znajdują się inne zwierzęta albo czy na ich drodze znajduje się przeszkoda. Współczesny radar działa na tej samej zasadzie, ale zamiast dźwięku wykorzystuje fale radiowe.

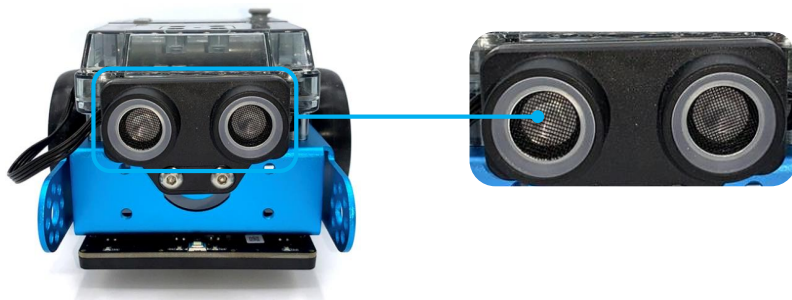


Rysunek 2. Przykład nietoperza i delfina

#### 2. Jak działa czujnik ultradźwiękowy?

mBot2 ma ultradźwiękowy nadajnik i odbiornik. Znajdują się one z przodu mBot2, w małych cylindrach, które można uznać za „oczy”. Ten czujnik emituje krótkie impulsy dźwięku ultradźwiękowego i nasłuchuje echa. Czy mBot2 zbliża się do przeszkody? Wówczas dźwięk ultradźwiękowy odbija się z powrotem do mBot2. Na podstawie czasu, po jakim dźwięk powraca do czujnika, mBot2 oblicza wewnętrznie odległość od przeszkody.

Dane te można wykorzystać do ustalenia, jak robot powinien zareagować : informujesz o tym robota, programując go . Takim działaniem może być na przykład zatrzymanie mBot2 lub wykonanie skrętu .



## 2. Zajęcia praktyczne (10 min)

### Krok 2 : Praca praktyczna

Ten krok składa się z dwóch części:

1. Zapoznanie się z różnymi blokami kodu czujnika ultradźwiękowego
2. Odtwarzanie i testowanie kilku przykładów programistycznych dotyczących czujnika ultradźwiękowego.

1. Zapoznanie się z różnymi blokami kodu czujnika ultradźwiękowego W mBlock dostępnych jest kilka bloków kodu czujnika ultradźwiękowego, których można używać w programach. W tym celu należy dodać rozszerzenie „Ultrasonic Sensor 2” z biblioteki rozszerzeń. Następnie bloki kodu znajdziesz w kategorii „Ultrasonic Sensor” w polu bloków w mBlock. Te bloki kodu są zielone.



W poniższej tabeli możesz zobaczyć kilka przykładów bloków kodu do programowania czujnika ultradźwiękowego czujnik.

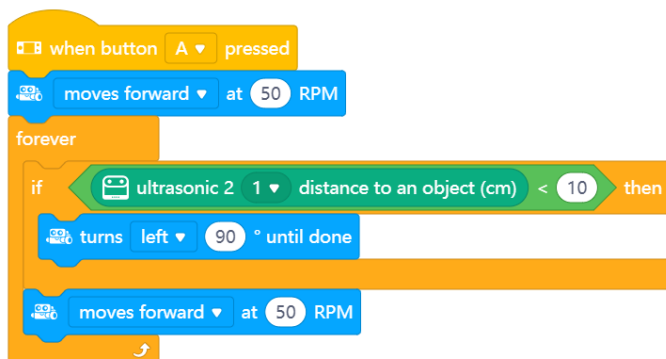


## Blok kodu:

 ultrasonic 2 1 ▼ distance to an object (cm)

Za pomocą tego bloku kodu mierzysz odległość między czujnikiem a przeszkodą. Zakres pomiaru wynosi od 3 do 300 cm. Możesz użyć wartości odległości, aby robot wykonał określone działanie. Na przykład możesz zapobiec zderzeniu mBot2 z przeszkodami.

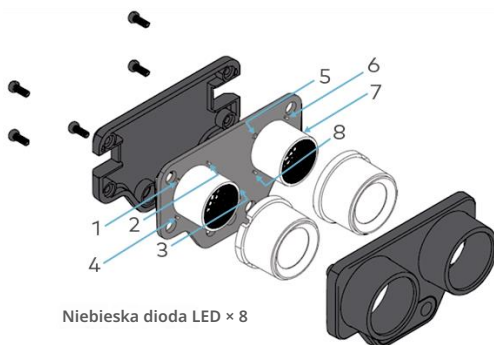
W poniższym przykładzie programistycznym mBot2 jedzie do przodu. Gdy mBot2 znajduje się w odległości mniejszej niż 10 cm od przeszkody, robot skręca o 90 stopni w lewo. Następnie mBot2 ponownie jedzie do przodu.



Blok kodu:



Czujnik ultradźwiękowy w mBot2 ma osiem niebieskich diod LED. Możesz użyć tych świateł, aby mBot2 pokazywał określoną „emocję”, na przykład, albo jako ogólnego wizualnego środka komunikacji niewerbalnej. Spraw, aby diody LED świeciły bardzo jasno, gdy mBot2 jest szczęśliwy, i mniej jasno, gdy jest ponury. Istnieje również specjalny blok kodu służący do wyświetlania wstępnie zdefiniowanych „emocji”. Na poniższym obrazie możesz zobaczyć dokładne położenie diod LED.



W mBlock znajduje się kilka bloków kodu, które umożliwiają zaprogramowanie jasności diod LED: możesz ustawić jasność diod LED, zmniejszając ją lub zwiększając o określony procent. Możesz też ustawić jasność bezpośrednio na określony procent. Zakres wynosi od 0 % do 100 %. Obie czynności można wykonać dla pojedynczej diody LED albo jednocześnie dla wszystkich. W poniższym przykładzie programistycznym jasność lampy LED 1 zwiększa się o 50 %.



2. Odtwarzanie i testowanie kilku przykładów programistycznych dotyczących czujnika ultradźwiękowego

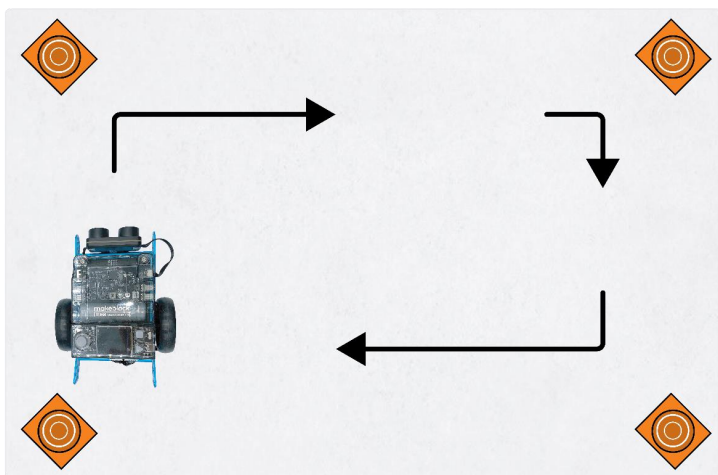
W powyższej tabeli każdy blok kodu czujnika ultradźwiękowego zawiera przykład programistyczny. Odtworzysz te przykłady programistyczne w mBlocku i je przetestujesz. W jednym z przykładów programistycznych wymyśl własne pomysły na rozszerzenie aplikacji.

### 3. Wypróbowywanie (25 min)

#### Krok 3: Wypróbuj

Wiesz już sporo o czujniku ultradźwiękowym w mBot2. Teraz utworzysz własny program komputerowy w mBlock, korzystając z czujnika ultradźwiękowego.

W tym zadaniu sprawisz, że mBot2 będzie wykonywał nieskończoną liczbę „okrążeń”. Aby to osiągnąć, umieścisz obiekt w każdym wierzchołku. Zaprogramujesz mBot2 tak, aby przy każdym obiekcie wykonywał skręt, a następnie wrócił do punktu początkowego. Przyjrzyj się poniższemu obrazowi.



Wykorzystaj wiedzę zdobytą w „Kroku 2” tej lekcji. Oczywiście możesz także eksperymentować z różnymi możliwościami programowania w mBlock.

Podczas wykonywania tego zadania pomocne jest skorzystanie z następującego planu krok po kroku. Czy masz pomysł, co chcesz stworzyć? Jeśli tak, najpierw omów z nauczycielem, czy jest to wykonalne.

|                             | Wyjaśnienie                                |
|-----------------------------|--------------------------------------------|
| Krok 1: Co chcesz zrobić?   | W jaki sposób mBot2 powinien się poruszać? |
| Krok 2: Czego potrzebujesz? | Czego potrzebujesz oprócz mBot2?           |

|                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Krok 3: Jakich bloków kodu potrzebujesz, aby wprowadzić mBot2 w ruch?</p> | <p>Jak sprawić, aby mBot2 zmieniał kierunek na każdym zakręcie?</p> <p>Jakich bloków kodu potrzebujesz, aby to zrobić?</p> <p>Przygotuj krótki opis działania swojego programu ( pseudokod / język naturalny, schemat blokowy lub UML)</p> <p>Jeśli potrzebujesz dodatkowych wyjaśnień, możesz omówić temat z innymi uczniami lub nauczycielem albo poszukać informacji na ten temat. W mBlock dostępna jest również pomoc do każdego bloku programistycznego</p> |
| <p>Krok 4: Testowanie i wdrażanie</p>                                        | <p>Czy pierwsza wersja jest gotowa? Przetestuj to!</p> <p>Podczas rundy testowej zapisz punkty do poprawy</p> <p>Pracuj nad punktami do poprawy, aż Twój mBot2 będzie działał bez błędów przez cały kurs</p> <p>Udało się? Nagraj efekt końcowy i zapytaj nauczyciela, czy możesz opublikować go w mediach społecznościowych z hashtagem # mBot2</p>                                                                                                              |

## 4. Podsumowanie (5 min)

### Krok 4: Podsumowanie

Czy udało Ci się sprawić, żeby mBot2 przejechał jedno okrążenie?

W tej lekcji poznasz czujniki ultradźwiękowe i dowiesz się, gdzie możesz spotkać je w życiu codziennym. Wiesz, jak zaprogramować czujnik ultradźwiękowy mBot2, aby na przykład zapobiec wjechaniu mBot2 w przeszkody. Wiesz również, jak używać diod LED w czujniku ultradźwiękowym, aby mBot2 wyrażał określone emocje.

Nadszedł teraz czas na krótką refleksję. Pomyśl samodzielnie i omów to w grupie:

Jak myślisz, co się udało?

Co można było zrobić lepiej?

Które części lekcji były dla Ciebie łatwe, a które sprawiły Ci większą trudność?

O czym chcielibyście uzyskać więcej wyjaśnień?

Kto mógłby Ci w tym pomóc?

## Lekcja 5: Zwiedzanie

Temat: STEAM

Klasy: 5. i starsze

Czas trwania: 45 minut

Poziom trudności: początkujący

### ★ Cele lekcji

Pod koniec tej lekcji uczniowie będą wiedzieć:

- Jak działają czujniki koloru

- Ich zastosowania w życiu codziennym i w Robotyce

### ★ Wprowadzenie

Wyobraź sobie, że jesteś w obcym mieście i jedziesz trenerem, zwiedzając najpiękniejsze atrakcje.

Być może samodzielnie odbyłeś taką podróż, a jeśli nie, prawdopodobnie znasz takie obrazy z telewizji i Internetu

. W tej lekcji mBot2 zostanie przekształcony w taki samochód turystyczny.

Ponieważ dla kierowcy bardzo nudne byłoby codzienne pokonywanie tej samej trasy (i wielokrotne opowiadanie o każdej atrakcji), zaprogramujemy tego trenera. Dzięki temu można przyjemnie podróżować, a kierowca nie nudzi się podczas wykonywania powtarzalnych czynności.

### ❁ Cel

Pod koniec tej lekcji uczniowie będą potrafili:

- Niech mBot2 podąża za linią

- Niech mBot2 wykonuje działania na podstawie kolorów

## Lista kontrolna przed lekcją

### Czego potrzebujesz?

Komputer stacjonarny lub laptop (z wyjściem USB) z zainstalowanym oprogramowaniem mBlock, wersją internetową (również na Chromebooku) albo tablet z zainstalowaną aplikacją mBlock

mBot2 z urządzeniem CyberPi

Kabel USB-C lub klucz sprzętowy Bluetooth Makeblock.

## Plan lekcji

Ta lekcja składa się z czterech kroków i jej ukończenie zajmuje łącznie 45 minut.

| Czas trwania | Spis treści                                                                                                                                                                                                                                                                                                                                  |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 minut      | <b>1. Rozgrzewanie</b><br>Czujniki koloru w życiu codziennym<br>Jak działają czujniki koloru? Co wyróżnia czujnik koloru w mBot2?                                                                                                                                                                                                            |
| 10 minut     | <b>2. Zajęcia praktyczne</b><br>Zapoznanie się z różnymi blokami kodu Quad RGB<br>czujnik<br>Odtwarzanie i testowanie przykładów programistycznych Quad RGB<br>czujnik<br>Jak mBot2 może podążać za linią?                                                                                                                                   |
| 25 minut     | <b>3. Wypróbowywanie</b><br>Napisanie własnego programu dla robota                                                                                                                                                                                                                                                                           |
| 5 minut      | <b>4. Podsumowanie</b><br>Czas na prezentację: pokaż, co udało Ci się zrobić z robotem, w krótkim, ciekawym filmie do późniejszej dyskusji<br>Jeśli nauczyciel wyrazi zgodę, udostępni efekt końcowy w mediach społecznościowych, używając hashtagu #mBot2<br>Odbicie: Z czego jesteś najbardziej dumny? Co chcesz ulepszyć w swoim robocie? |

## 1. Rozgrzewanie (5 min)

### Krok 1: Rozgrzewka

Ten krok składa się z dwóch części:

1. Czujniki koloru w życiu codziennym
2. Jak działają czujniki koloru? Co wyróżnia czujnik koloru w mBot2?

#### 1. Czujniki koloru w życiu codziennym

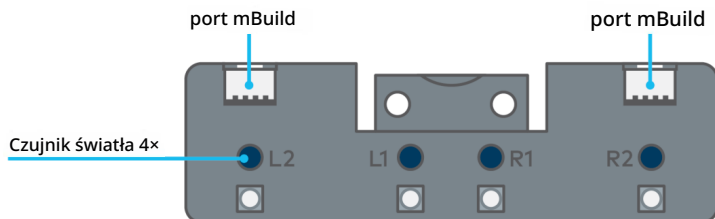
Czujniki koloru mogą wykrywać natężenie światła i kolory. Z tego rodzaju czujnikiem spotkasz się w wielu różnych miejscach w codziennym życiu.

Na przykład robot magazynowy może znaleźć właściwą trasę przez magazyn na podstawie różnych kolorów na podłodze. Ten rodzaj czujnika jest również używany, gdy wymagane jest precyzyjne dopasowanie kolorów, np. podczas lądowania pojazdów. Urządzenie pomiarowe z czujnikiem koloru, czyli miernik kolorów, może sprawdzić, czy do pomalowania pojazdu natryskowo użyto właściwego koloru. Czy wraz z kolegami i koleżankami z grupy potraficie wymyślić inne przykłady?

#### 2. Jak działają czujniki koloru? Co wyróżnia czujnik koloru w mBot2?

Czujnik koloru składa się z trzech różnych (wewnętrznych) czujników, które mierzą światło odbite od obiektu na podstawie natężenia składowych czerwonej, zielonej i niebieskiej. Dlatego czujniki koloru nazywa się również „czujnikami RGB”. Czujnik dekoduje każdy kolor na trzy wartości, a określona „mieszanka” tych wartości reprezentuje dany kolor.

mBot2 ma cztery takie czujniki RGB zintegrowane w jednym czujniku. Przyjrzyj się poniższemu obrazowi



To poczwórny czujnik RGB znajdujący się na mBot2. Czujniki RGB noszą nazwy L1, L2, R1 i R2 (L oznacza lewą stronę, a R prawą). Automatycznie wykrywają wartości RGB odbitego koloru i wewnętrznie porównują ich kombinację z różnymi zaprogramowanymi wzorcami kolorów. Znacznie ułatwia to programowanie, ponieważ czujnik może wtedy podawać kolor i nie trzeba samodzielnie sprawdzać kodów RGB. Czujnik może wykrywać 6 różnych kolorów oraz czerni i biel. Zamiast koloru może podawać odcienie szarości, a także wykrywać linie i rozgałęzienia.

Poczwórny czujnik RGB umożliwia mBot2 podążanie za linią na podłodze i reagowanie na różne znaczniki kolorów. Poczwórny czujnik RGB znajduje się z przodu mBot2. Na dolnej stronie zobaczysz cztery czujniki. U góry znajdziesz nazwy każdego z nich, dzięki czemu możesz rozpoznawać czujniki i programować je w określony sposób. Przyjrzyj się poniższym ilustracjom.



Poczwórny czujnik RGB — widok z przodu Widok z góry

Widok z dołu

Mały przycisk na górnej stronie czujnika służy do kalibracji. Jeśli klikniesz go dwukrotnie, diody LED zaczną migać i możesz „przesunąć” mBot2 wraz z czujnikiem w poprzek linii, za którą robot ma podążać. Spowoduje to skalibrowanie czujnika tak, aby odróżniał linię od tła. Jeśli używasz czarnej linii na białej powierzchni, zazwyczaj nie jest to konieczne.

Mapa dołączona do mBot2 zawiera kolory zaznaczone wewnątrz samej trasy, aby robot reagował na kody kolorów. Może to spowodować, że czujnik pomyli jasny kolor, taki jak żółty, z tłem zamiast z trasą.

Dlatego jako pierwsze ćwiczenie skalibruj czujnik do kodu koloru żółtego; będzie on wtedy automatycznie niezawodnie rozpoznawać ten kolor oraz wszystkie „ciemniejsze” kolory jako linię.

Podczas programowania mBot2 użyjesz jednego lub większej liczby tych czterech czujników, aby po drodze sprawdzać określone kolory na podłodze. Możesz sprawić, aby mBot2 wykonał polecenie, gdy wykryje kolor. Na przykład „zatrzymaj się”, gdy wykryje kolor czerwony, oraz „skręć w lewo”, gdy wykryje kolor zielony.



## 2. Zajęcia praktyczne (10 min)

### Krok 2 : Praca praktyczna

Ten krok składa się z trzech części:

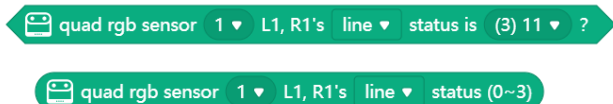
1. Zapoznanie się z różnymi blokami kodu poczwórnego czujnika RGB
2. Odtwarzanie i testowanie kilku przykładów programistycznych z użyciem poczwórnego czujnika RGB
3. Jak mBot2 może podążać za linią?

1. Poznanie różnych bloków kodu poczwórnego czujnika RGB W mBlock dostępnych jest kilka bloków kodu, których można użyć do programowania poczwórnego czujnika RGB. Bloki kodu te można znaleźć w kategorii „Quad RGB” w polu bloków w mBlock. Te bloki kodu są zielone.



W poniższej tabeli możesz zobaczyć niektóre bloki kodu, których możesz użyć w swoim programie z poczwórnym czujnikiem RGB. Pierwsze cztery bloki kodu służą do ( prostego ) wykrywania linii. Można ich używać do prostego i zaawansowanego podążania za linią, a także do wykrywania skrzyżowań i ostrych ( 90 ° ) zakrętów. Kalibracja czujnika do torów w różnych kolorach została krótko omówiona w powyższej sekcji. Niebieskie diody LED na górnej stronie czujnika wskazują wykrywanie linii / tła: gdy są wyłączone, wykrywają ciemny kolor ( linię ), a gdy świecą na niebiesko, wskazują jasny kolor ( tło ). Stan diody LED wskazuje zatem współczynnik odbicia podłoża lub mapy.

#### Bloki kodu:



Te bloki kodu uwzględniają podczas wykrywania linii tylko dwa czujniki wewnętrzne: L1 i R1. Te bloki są przeznaczone do prostego podążania za linią.

Użyj pierwszego bloku w instrukcjach warunkowych. Użyj drugiego bloku, jeśli chcesz wyświetlić wartość na ekranie lub zapisać ją w zmiennej.

Jeśli czujnik znajduje się nad czarną linią, obie wewnętrzne diody LED powinny być wyłączone. Ponieważ każdy czujnik (L1, R1) może za pomocą tego bloku wykryć linię albo tło, istnieją 4 możliwe kombinacje: będą one reprezentowane przez liczby od 0 do 3 (dla ułatwienia blok pokazuje również wzorzec binarny):

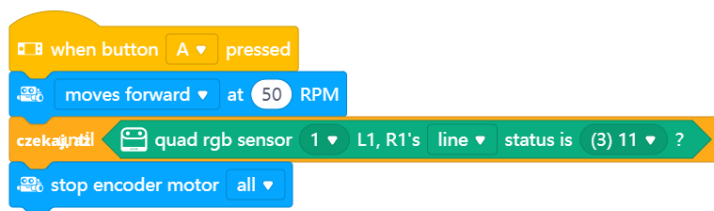
| L1                                                                                | R1                                                                                | Znaczenie                                                      | Stan<br>(dziesiętny) |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------|----------------------|
|  |  | Czujnik znajduje się na białym tle, obie diody LED są włączone | 0                    |
|  |  | R1 wykrywa czarną linię                                        | 1                    |
|  |  | L1 wykrywa czarną linię                                        | 2                    |
|  |  | Oba czujniki wewnętrzne wykrywają czarną linię                 | 3                    |

Pamiętaj, że diody LED pokazują stan tła (włączone lub logiczna wartość „true” = wykryto tło), podczas gdy ten blok kodu wykrywa kolor linii (logiczna wartość „true” = wykryto linię). Jeśli chcesz używać więcej niż jednego poczwórnego czujnika RGB, możesz wskazać programowi, którego czujnika ma użyć, podając pierwszą liczbę we wszystkich poniższych blokach kodu. Przyjrzyj się poniższemu obrazowi.

Numer 1 oznacza pierwszy podłączony czujnik, numer 2 — drugi podłączony czujnik i tak

dalej.

W poniższym przykładzie zostanie użyta funkcja wykrywania linii tego bloku, a robot zatrzyma się, gdy tylko dotrze do czarnej linii. Później ten przykład zostanie zmieniony tak, aby robot podążał za linią.



Ponieważ podczas jazdy czas reakcji może mieć kluczowe znaczenie, ważne jest użycie trybu przesyłania. Możesz spróbować samodzielnie i zobaczyć różnicę po przełączeniu trybów.

**Bloki kodu:**

quad rgb sensor 1 ▼

line ▼

status (0~1

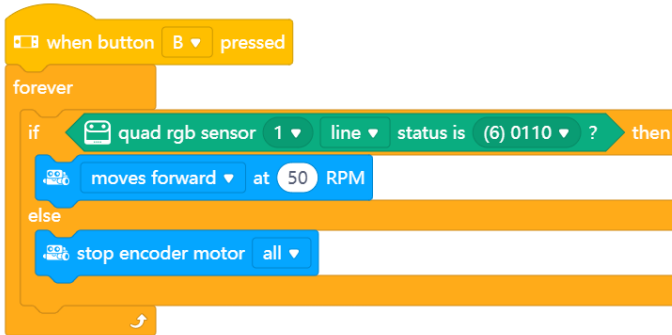
Użyj pierwszego bloku w instrukcjach warunkowych. Użyj drugiego bloku, jeśli chcesz wyświetlić wartość na ekranie lub zapisać ją w zmiennej.

Te bloki działają według dokładnie tej samej logiki co poprzednie, z wyjątkiem tego, że mają większy obszar wykrywania i mogą rozpoznawać skrzyżowania podczas podążania za linią. Pamiętaj, bloki sprawdzają stan linii, więc jeśli jeden z czterech czujników wykryje (ciemną) linię, odpowiadająca jej cyfra binarna zostanie ustawiona na "1" (wartość logiczna true), ale diody LED zgasną (brak odbicia od tła na tym czujniku). Teraz, dzięki 4 indywidualnym czujnikom, liczba możliwych stanów wzrasta do 16. Poniższa tabela przedstawia najważniejsze stany:

| Wzorec binarny | Znaczenie                                                                                                         | Stan (dziesiętny) |
|----------------|-------------------------------------------------------------------------------------------------------------------|-------------------|
| 0000           | wszystkie 4 czujniki na białym tle ( brak toru )                                                                  | 0                 |
| 0010           | tylko R1 na czarnej linii                                                                                         | 2                 |
| 0100           | tylko L1 na czarnej linii                                                                                         | 4                 |
| 0110           | czujniki wewnętrzne R1 i L1 na czarnej linii                                                                      | 6                 |
| 0111           | ostry skręt w prawo ( L1 i R1 na czarnej linii, skrzyżowanie w prawo; skrzyżowanie w kształcie litery L w prawo ) | 7                 |
| 1110           | ostry skręt w lewo ( L1 i R1 na czarnej linii, skrzyżowanie w lewo; skrzyżowanie w kształcie litery L w lewo )    | 14                |
| 1111           | skrzyżowanie w kształcie litery T ( L1 i R1 na czarnej linii, skrzyżowanie zarówno w prawo, jak i w lewo )        | 15                |

Skrzyżowanie w kształcie litery X można wykryć tylko, przejeżdżając przez skrzyżowanie w kształcie litery T i sprawdzając, czy środkowa linia jest kontynuowana.

Poniższy przykładowy program sprawi, że robot będzie jechał tak długo, jak długo za pomocą dwóch wewnętrznych czujników znajduje się na czarnej linii. Zatrzyma się, jeśli zjedzie z trasy, wykryje skrzyżowanie lub linia nagle się skończy:

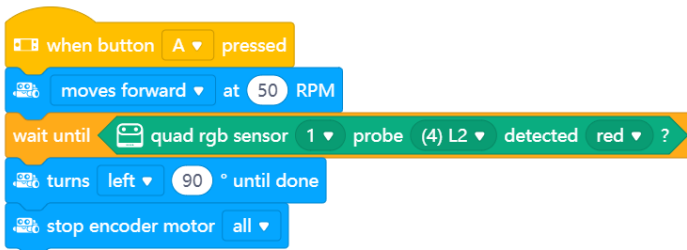


#### Blok kodu:



Za pomocą tego bloku kodu wybierasz jeden z poczwórnych czujników RGB ( L2 , L1 , R1 , R2 ) i określasz, czy ma rozpoznawać linię, tło, kolor biały, czarny lub dowolny z 5 pozostałych kolorów (czerwony, zielony, niebieski, żółty, cyjan, fioletowy). Do wykrywania i śledzenia linii zalecane są poprzednie bloki , ponieważ sprawdzają jednocześnie więcej niż jeden czujnik (są szybsze i dokładniejsze, ponieważ robot nie przemieszcza się między odczytami). Możesz jednak użyć tego bloku na przykład do wykrywania czerwonego znacznika koloru po lewej stronie robota — gdy robot podąża za linią.

W poniższym przykładzie programistycznym, jeśli naciśniesz przycisk B, mBot2 jedzie prosto (zakładając, że podąża za linią) i wykonuje obrót o 90°, gdy po lewej stronie (wymagowanej) trasy zostanie wykryty czerwony znacznik.

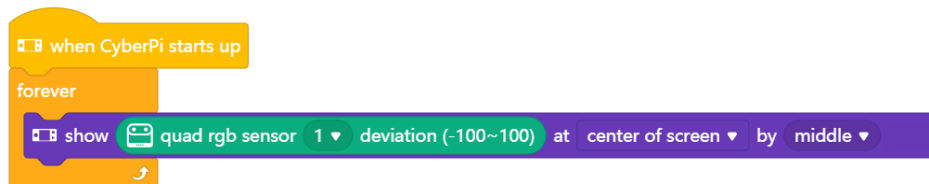


Blok kodu:

 quad rgb sensor 1 ▼ deviation (-100~100)

Ten blok kodu mierzy, jak bardzo robot odchyła się od czarnej trasy. Możesz go użyć do zaawansowanego i płynnego podążania za linią. Poprzedni blok kodu do podążania za linią wykorzystuje potrójne rozróżnienie: na trasie, z odchyleniem w lewo lub z odchyleniem w prawo (a w niektórych przypadkach także wykrywanie skrzyżowań). Jednak podczas codziennej jazdy na rowerze dostosowujesz sterowanie do zakrętu bardziej precyzyjnie. Ten blok umożliwia proporcjonalne podążanie za linią: im bardziej robot zbacza z trasy, tym bardziej skręca w przeciwnym kierunku. Jeśli odchylenie od trasy wynosi 0, sterowanie również wynosi 0 (jazda prosto).

W kroku 3 omówimy, jak możemy wykorzystać ten blok do „płynnego” podążania za linią, programując proporcjonalny moduł podążania za linią. Poniższy przykład programistyczny wyświetli odchylenie na ekranie, dzięki czemu możesz najpierw spróbować ręcznie przesuwając robota po trasie:



## 2. Odtwarzanie i testowanie kilku przykładów programistycznych z użyciem poczwórnego czujnika RGB

Na powyższym diagramie każdy blok kodu poczwórnego czujnika RGB zawiera przykład programistyczny.

Odtworzysz te przykłady programistyczne w mBlocku i je przetestujesz. W przypadku jednego przykładu programistycznego wymyślisz własną modyfikację kodu.

### 3. Jak mBot2 może podążać za linią?

Podstawą zadania w „kroku 3” tej lekcji jest to, że mBot2 będzie podążać trasą na mapie jako autokar wycieczkowy. W tym celu należy wykorzystać dotychczas wyjaśnione i przetestowane przykłady kodu oraz odpowiednio je zmodyfikować.

Najpierw użyj tego bloku programistycznego:

 quad rgb sensor 1 ▼ L1, R1's line ▼ status is (3) 11 ▼

Jeśli robot znajduje się na trasie (zarówno L1, jak i R1 wykrywają linię, wartość kodu 3), robot powinien jechać prosto.

Jeśli tylko jeden z dwóch czujników wewnętrznych wykrywa linię, robot powinien odpowiednio skrócić w lewo lub w prawo.

Użyj tych bloków sterowania jazdą (pokazano tylko skręt w prawo):



Uruchamiają one silniki w razie potrzeby, a następny blok kodu zostanie wykonany natychmiast.

Ponieważ nie wiesz, jak daleko robot znajduje się poza trasą ani kiedy pojawi się następny zakręt, polecenia jazdy nie mogą zawierać czasu ani odległości przejazdu (tak samo w przypadku skręcania). W następnej iteracji pętli programu zostaną sprawdzone nowe dane z czujnika i na ich podstawie zostanie wybrany sposób jazdy.

Jeśli korzystasz z dostarczonej mapy, pamiętaj o skalibrowaniu czujnika zgodnie z opisem, ustawiając go na najjaśniejszy kolor (żółty fragment trasy).

### 3. Wypróbowywanie (25 min)

#### Krok 3: Wypróbuj

Wiesz już dużo o poczwórnym czujniku RGB w mBot2. Teraz utworzysz własny program komputerowy w mBlock, korzystając z poczwórnego czujnika RGB.

W tym zadaniu zmienisz mBot2 w samochód turystyczny. Trener wycieczki zabiera uczestników na wycieczkę po punktach orientacyjnych miasta. W pudełku mBot2 znajduje się mapa z trasą. Trener podąża za czarną linią na mapie. W poprzedniej części powinieneś już zaprogramować prostego robota podążającego za linią, a teraz nadszedł czas, aby dodać kolejne usprawnienia.

Aby ułatwić zrozumienie programu, możesz „zbudować” własne bloki programistyczne.

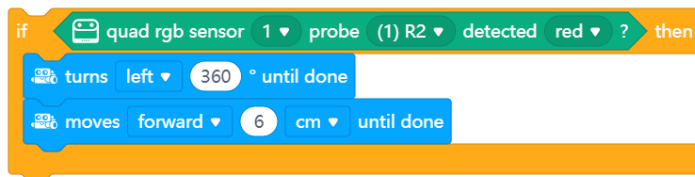
Całą część programu dotyczącą podążania za linią można połączyć w jedno proste polecenie, np. „simple\_line”. Czerwone/różowe „Moje bloki” w menu kodu pozwalają zdefiniować blok niestandardowy i uporządkować kod w przejrzysty sposób. Poniższy przykładowy program korzysta z bloków niestandardowych.



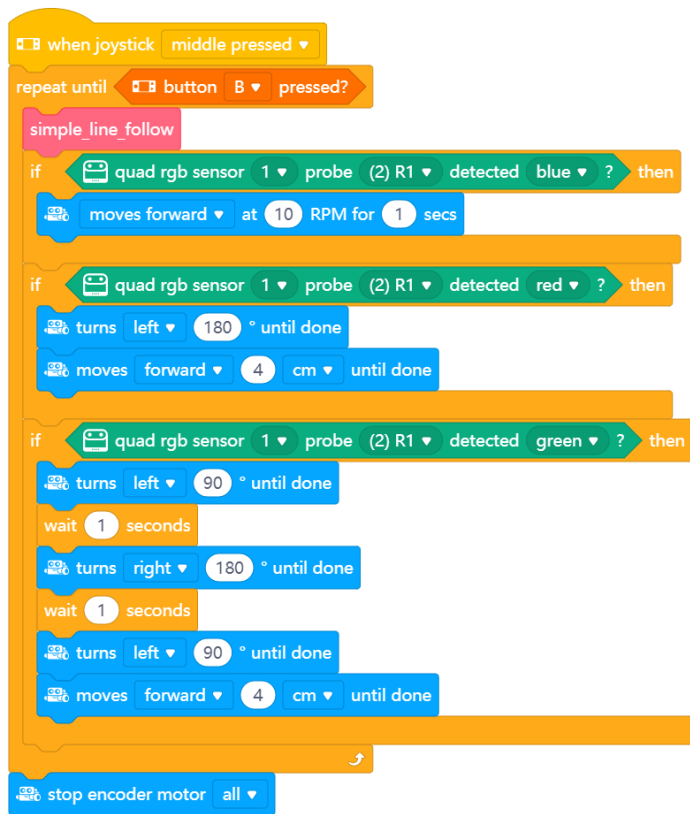
Oprócz podążania trasą trener wykonuje działanie przy niektórych atrakcjach. Atrakcje są oznaczone kolorowym obszarem (czerwonym, żółtym, zielonym lub niebieskim).

Możesz wymyślić działania, które powinien wykonywać trener. Czy zamierzasz kazać trenerowi jeździć po Paryżu? Wtedy może być ciekawie odtworzyć hymn Francji. Albo dźwięk dzwonów Big Bena, jeśli trener przejeżdża przez Londyn.

Oczywiście możesz też zachować to w nieco prostszej formie. W poniższym przykładzie trener zawraca przy czerwonym znaczniku i wykonuje różne działania przy innych kolorowych znacznikach.

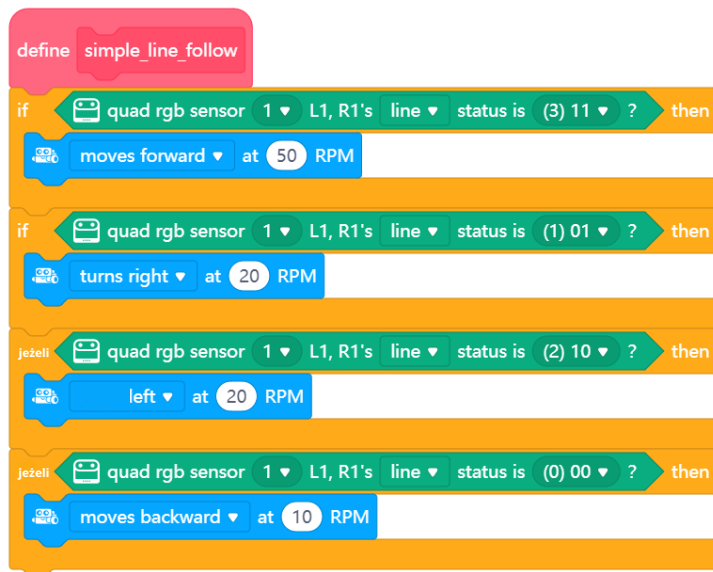


Wykorzystaj wiedzę zdobytą w „Kroku 2” tej lekcji. Oczywiście możesz samodzielnie przeprowadzać wiele eksperymentów, korzystając z różnych możliwości programowania w mBlock. Utknąłeś? W takim razie możesz skorzystać z poniższych przykładów programistycznych. Możesz je skopiować i dostosować do własnych życzeń i preferencji.



Ten program uruchamia się po naciśnięciu przycisku B i zatrzymuje się po naciśnięciu przycisku A. Wykonuje proste podążanie za linią przy użyciu niestandardowego kodu „ Moje bloki ”, a następnie sprawdza, czy czujnik R1 wykrywa kolor niebieski ( jedź bardzo powoli ), czerwony ( zawróć ) lub zielony ( spójrz w obie strony ). Aby mieć pewność, że kolor nie zostanie rozpoznany ponownie po wykonaniu aktywności, robot za każdym razem jedzie 4 cm do przodu

Blok „ simple\_line ” jest zdefiniowany w następujący sposób:



#### Wersja zaawansowana :

W poprzednim kroku wprowadzono odchylenie od linii . Ten blok pokazuje, jak bardzo robot znajduje się poza trasą ( od -100 do +100 ; 0 oznacza idealne położenie na środku ) i może służyć do obliczania sterowania robotem . Jest to proporcjonalne podążanie za linią — im bardziej robot znajduje się poza trasą, tym bardziej musi korygować kierunek jazdy . Jeśli robot znajduje się poza trasą po prawej stronie, odchylenie jest dodatnie ( robot musi jechać w lewo ) ; przy wartościach ujemnych robot znajduje się po lewej stronie trasy ( musi jechać w prawo )

Aby to zrealizować, potrzebujesz podstawowej prędkości — czyli prędkości poruszania się do przodu po prostej trasie. Aby sterować w lewo lub w prawo, jedno z kół musi obracać się szybciej niż drugie — im większa różnica, tym większe sterowanie.



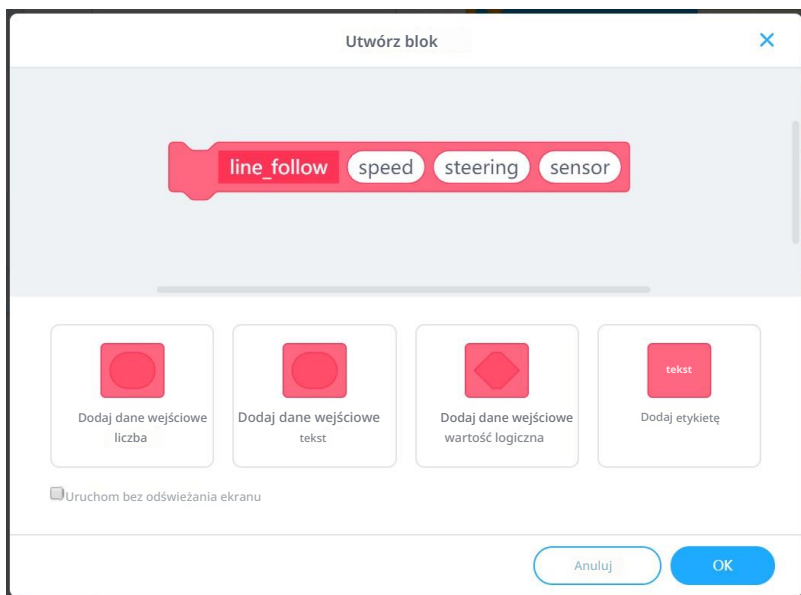
Aby obliczyć sterowanie, musimy dodać odchylenie do podstawowej prędkości jednego silnika i odjąć je od podstawowej prędkości drugiego. Aby umożliwić dokładniejszą regulację, zostanie użyty współczynnik dostosowujący siłę, z jaką odchylenie wpływa na sterowanie, ponieważ zakres odchylenia jest zbyt duży, aby można go było bezpośrednio wykorzystywać do sterowania silnikami.

Ale do którego silnika należy dodać lub od którego odjąć obliczoną wartość? Dodatnie odchylenie oznacza, że robot znajduje się poza trasą po prawej stronie i musi skrócić w lewo. Oznacza to, że prawe koło musi obracać się szybciej — dodajemy zatem przeskalowane odchylenie do prawego koła i odejmujemy je od lewego.

speed + steering \* sensor

speed - steering \* sensor

Podczas konfigurowania „Moich bloków” możesz dodać dane wejściowe i zmienić ich nazwy na znaczące:



Jedynie, o co trzeba jeszcze zadbać, to to, że blok ruchu (pokazany poniżej) wymaga ujemnych wartości dla prawego silnika i dodatnich dla lewego, aby jechać do przodu (ponieważ są zamontowane naprzeciwko siebie).

encoder motor EM1 rotates at 50 RPM, encoder motor EM2 rotates at 50

Aby zakończyć obliczenia, wartości dla prawego silnika należy pomnożyć przez -1. Oto definicja bloku proporcjonalnego podążania za linią:

```

define line_follow speed steering sensor
set left to -1 * speed + steering * sensor
set left to 1 * speed - steering * sensor
encoder motor EM1 rotates at left RPM, encoder motor EM2 rotates at right RPM
  
```

W pętli głównej programu wystarczy użyć tego bloku wraz z jednym poleceniem:

line\_follow 50 0.6 quad rgb sensor 1 deviation (-100~100)

Wywołuje to podprogram z podstawową prędkością 50 i współczynnikiem sterowania 0.6, a także przekazuje bieżące odchylenie odczytane przez czujnik. Możesz nieco poeksperymentować z wartościami — jeśli robot „oscyluje” (szybko zmienia kierunek skrętu z lewego na prawy itd.), współczynnik sterowania jest zbyt wysoki; jeśli zjeżdża z trasy nawet na łagodnych zakrętach, jest zbyt niski.

Podczas wykonywania tego zadania pomocne jest zastosowanie następującego planu krok po kroku. Czy masz pomysł, co chcesz stworzyć? Jeśli tak, najpierw omów z nauczycielem, czy jest to wykonalne.

|                                                                       | Wyjaśnienie                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Krok 1: Co chcesz zrobić?                                             | W jaki sposób mBot2 powinien się poruszać?                                                                                                                                                                                                |
| Krok 2: Czego potrzebujesz?                                           | Czego potrzebujesz oprócz mBot2?                                                                                                                                                                                                          |
| Krok 3: Jakich bloków kodu potrzebujesz, aby wprowadzić mBot2 w ruch? | <p>Jak sprawić, aby mBot2 zmieniał kierunek na każdym zakręcie?</p> <p>Jakich bloków kodu potrzebujesz, aby to zrobić?</p> <p>Przygotuj krótki opis działania swojego programu (pseudokod / język naturalny, schemat blokowy lub UML)</p> |

|                                |                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                | Jeśli potrzebujesz dalszych wyjaśnień, możesz omówić to z kolegami z klasy, nauczycielem lub poszukać informacji na ten temat. W mBlock dostępna jest również pomoc do każdego bloku programistycznego                                                                                                                           |
| Krok 4: Testowanie i wdrażanie | Czy pierwsza wersja jest gotowa? Przetestuj to!<br>Podczas rundy testowej zapisz punkty do poprawy<br>Pracuj nad punktami do poprawy, aż Twój mBot2 będzie bezbłędnie poruszał się po trasie<br>Udało się? Nagraj efekt końcowy i zapytaj nauczyciela, czy możesz opublikować go w mediach społecznościowych z hashtagem # mBot2 |

#### 4. Podsumowanie (5 min)

##### Krok 4: Podsumowanie

Czy udało Ci się zaprogramować mBot2 tak, aby jeździł po trasie jak samochód turystyczny?

W tej lekcji poznano poczwórny czujnik RGB i dowiedziano się, gdzie można się z nim spotkać w codziennym życiu. Wiesz, jak zaprogramować poczwórny czujnik RGB mBot2, aby robot podążał za linią.

Dowiedziałeś się także, jak zaprogramować mBot2, aby pokonywał trasę na podstawie różnych kolorów.

Nadszedł teraz czas na krótką refleksję. Pomyśl samodzielnie i omów to w grupie:

- Jak myślisz, co się udało?

Co można było zrobić lepiej?

Które części lekcji były dla Ciebie łatwe, a które sprawiły Ci większą trudność?

O czym chcielibyście uzyskać więcej wyjaśnień?

Kto mógłby Ci w tym pomóc?

## Lekcja 6:

### Ostrożna jazda

Temat: STEAM

Czas trwania: 45 minut

Klasy: 5. i starsze

Poziom trudności: początkujący

#### ★ Cele lekcji

Pod koniec tej lekcji uczniowie będą potrafili:

Rozpoznawać i używać bloków kodu akcelometru żyroskopowego w CyberPi

Tworzyć własny program komputerowy w mBlocku, aby mierzyć zmiany nachylenia mBot2 i odpowiednio dostosowywać jego ruchy

#### ★ Wprowadzenie

Żyroskop mierzy ruchy przechylenia, a dokładniej prędkość skrętu lub ruchów przechylenia. Akcelometr mierzy zmianę prędkości. Oba czujniki dostarczają różnych informacji o położeniu w przestrzeni.

Łącząc oba rodzaje danych — położenie i ruch pojazdów, robotów, a nawet sond kosmicznych — można obliczyć. Na przykład na wyboistych drogach roboty mogą jechać wolniej, aby zapobiec przewróceniu się.



#### 🌀 Skupienie

Pod koniec tej lekcji uczniowie będą wiedzieć:

Jak działają żyroskopy i akcelerometry oraz jak można łączyć wyniki ich pomiarów

Jak sprawić, aby robot reagował na nagłe zmiany i wykrywał zderzenia lub wyboiste drogi

## □ Lista kontrolna przed lekcją

Czego potrzebujesz?

Komputer stacjonarny lub laptop (z wyjściem USB) z zainstalowanym oprogramowaniem mBlock, wersją internetową (również na Chromebooku) albo tablet z zainstalowaną aplikacją mBlock  
mBot2 z urządzeniem CyberPi  
Przewód USB-C lub klucz sprzętowy Bluetooth Makeblock

### Plan lekcji

Ta lekcja składa się z czterech kroków i trwa łącznie 45 minut.

| Czas trwania | Spis treści                                                                                                                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 minut      | <b>1. Rozgrzewanie</b><br>Żyroskopy i akcelerometry w życiu codziennym<br>Jakie ruchy przechylania może wykrywać mBot2?                                                                                                                                                                                                                                                             |
| 10 minut     | <b>2. Zajęcia praktyczne</b><br>Zapoznanie się z różnymi blokami programistycznymi żyroskopu i akcelometru<br>Odtwarzanie i testowanie niektórych przykładów programistycznych dotyczących tych czujników.                                                                                                                                                                          |
| 25 minut     | <b>3. Wypróbowywanie</b><br>Napisanie własnego programu dla robota                                                                                                                                                                                                                                                                                                                  |
| 5 minut      | <b>4. Podsumowanie</b><br>Czego nauczyłeś się na tej lekcji?<br>Czas na prezentację: pokaż, co udało Ci się zrobić z robotem, w krótkim, ciekawym filmie do późniejszej dyskusji<br>Jeśli nauczyciel wyrazi zgodę, udostępnij efekt końcowy w mediach społecznościowych, używając hashtagu #mBot2<br>Odbicie: Z czego jesteś najbardziej dumny? Co chcesz ulepszyć w swoim robocie? |

## 1. Rozgrzewanie (5 min)

### Krok 1: Rozgrzewka

Ten krok składa się z dwóch części:

1. Żyroskopy i akcelerometry w życiu codziennym
2. Jakie ruchy przechylania może wykrywać mBot2?

#### 1. Żyroskopy i akcelerometry w życiu codziennym

W jaki sposób urządzenia, takie jak smartfony, rozpoznają swoje położenie (trzymane pionowo, z wyświetlaczem skierowanym do góry / do dołu)? Jak określić, czy zderzenie jest wystarczająco silne, aby uruchomić poduszki powietrzne? Szacowanie bieżącego położenia bez zewnętrznego, precyzyjnego punktu odniesienia od stuleci stanowiło wyzwanie na morzu przy zachmurzonym niebie — i nadal jest bardzo ważnym zadaniem w przypadku przedmiotów codziennego użytku, robotów, a nawet sond kosmicznych. Wszystkie wykorzystują co najmniej dwa różne czujniki: żyroskopy i akcelerometry. Żyroskop działa dzięki kryształowi znajdującemu się pod napięciem elektrycznym. Przekształca on kierunek siły powstającej podczas przechylania w sygnał elektryczny. Żyroskopy mierzą prędkość obrotową w stopniach na sekundę i za pomocą matematyki mogą również oszacować kąt obrotu — jednak ten drugi pomiar nie jest zbyt dokładny: żyroskopy z czasem „dryfują”. Akcelerometry informują o zmianie położenia (na przykład o zwiększeniu lub zmniejszeniu prędkości) — a w odniesieniu do ziemskiej grawitacji informują o przyspieszeniu grawitacyjnym (w porównaniu ze spadkiem swobodnym). Oba typy czujników mogą dokonywać pomiarów wzdłuż każdej osi przestrzennej, raportując prędkość obrotową lub przyspieszenie na osi X, Y i Z.

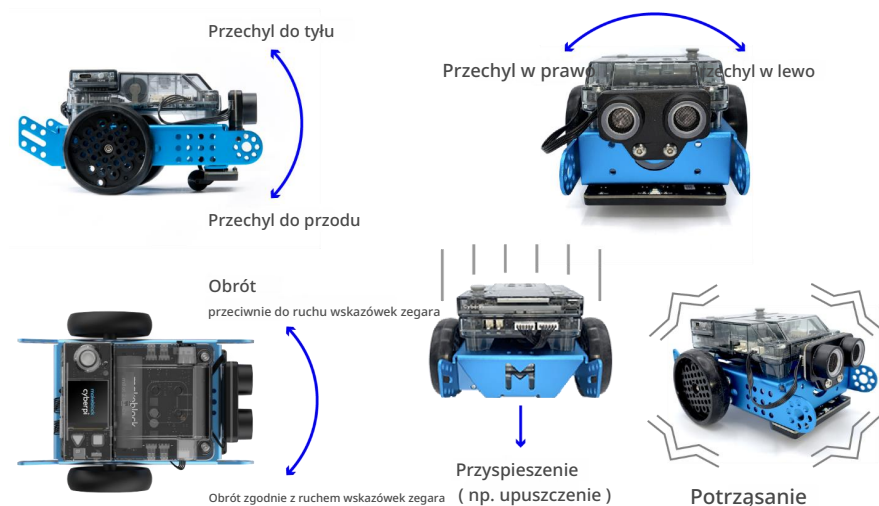
Łącząc dane z żyroskopu i akcelerometru, można obliczyć kierunek i siłę ruchów (w rzeczywistości nie jest to wyjątkowo precyzyjne, więc jest to raczej bardzo dobre przybliżenie). W Robotyce czujniki te służą do stabilizowania pojazdów na nierównym podłożu (lub w locie) i pomagają im utrzymywać zaprogramowaną ścieżkę. W przypadku mBot2 czujniki te są zintegrowane w małym układzie na płycie CyberPi.

Dlatego będziemy określać połączony czujnik mianem „czujnika ruchu”.

## 2. Jakie ruchy przechylania może wykrywać mBot2?

Zanim nauczysz się programować w mBlock5 z użyciem czujnika ruchu, warto wiedzieć, jakie ruchy przechylania może mierzyć mBot2. Pomiary ruchów przechylania możesz wyświetlać na wyświetlaczu CyberPi. W lekcji 2 dowiedziałeś się już, jak to zrobić.

mBot2 może wykrywać następujące ruchy przechylania:



## 2. Zajęcia praktyczne (10 min)

### Krok 2 : Praca praktyczna

Ten krok składa się z dwóch części:

1. Poznajemy różne bloki programistyczne czujnika ruchu
2. Odtwarzanie i testowanie kilku przykładów programistycznych z użyciem czujnika ruchu

### 1. Poznajemy różne bloki programistyczne czujnika ruchu

W mBlock dostępnych jest kilka bloków kodu dotyczących czujnika ruchu, których możesz użyć we własnych programach. Te bloki kodu można znaleźć w kategorii „Wykrywanie ruchu” w obszarze bloków w mBlock. Te bloki kodu są pomocnicze.



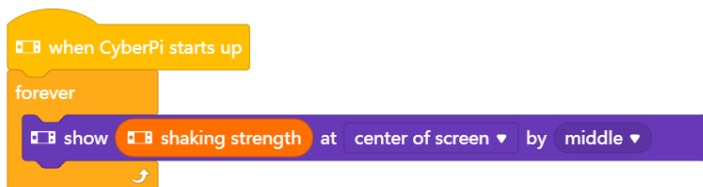
**Motion  
Sensing**

Poniższa tabela zawiera szczegółowe objaśnienia bloków kodu oraz przykładowy kod programu do ich testowania.

**Blok kodu:**

 shaking strength

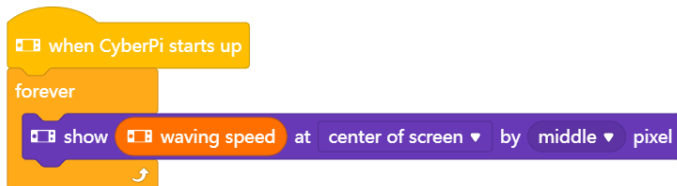
Ten blok kodu informuje, jak silnie potrząsany jest mBot2 . Przy wartości „ 0 ” mBot2 w ogóle nie jest potrząsany . Przy wartości „ 100 ” wykrywany jest bardzo silny ruch potrząsania . W poniższym przykładzie programistycznym siła wstrząsów mBot2 jest wyświetlana na wyświetlaczu .  
Jeśli mBot2 przestanie się trząść , wyświetlacz pozostaje pusty .



**Blok kodu:**

 waving speed

Ten blok kodu służy do rejestrowania prędkości machania . Prędkość machania mieści się w zakresie od 0 do 100 . Jednostki te nie oznaczają prędkości w m / s ani w stopniach / s , lecz stanowią jedynie prosty punkt odniesienia . W poniższym przykładzie programistycznym prędkość machania mBot2 jest wyświetlana na wyświetlaczu .  
Jeśli mBot2 przestanie machać , wyświetlacz pozostaje pusty .



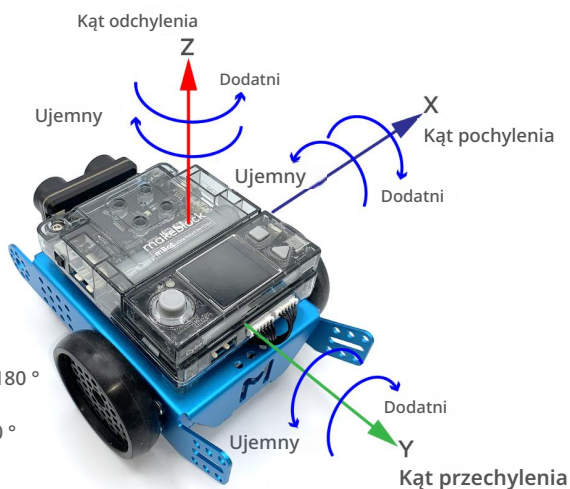


Blok kodu:

tilted forward ▼ angle (°)

Każde pochylenie powoduje ruch pod pewnym kątem w określonym kierunku. Przy określaniu kąta do wskazania kierunku używa się osi X, Y i Z. Zamiast tych terminów matematycznych stosuje się również określenia: przechylenie, pochylenie i odchylenie. Terminy te wywodzą się z nawigacji lotniczej. Przechylenie odnosi się do osi przód-tył, pochylenie oznacza obrót w górę lub w dół (oś biegnąca od lewej do prawej), a odchylenie to oś góra-dół odpowiadająca za obrót w lewo lub w prawo.

Przyjrzyj się poniższemu obrazowi.



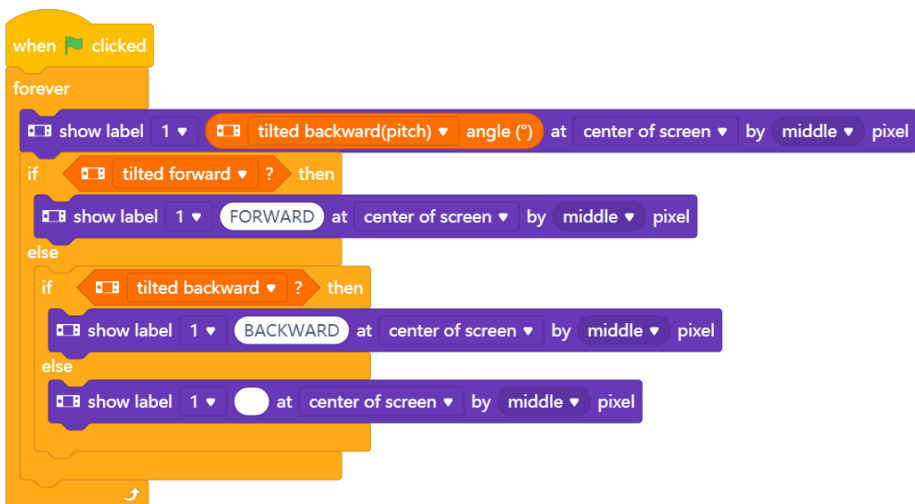
Kąty podawane są w stopniach :

Kąt przechylenia : od  $-180^{\circ}$  do  $+180^{\circ}$

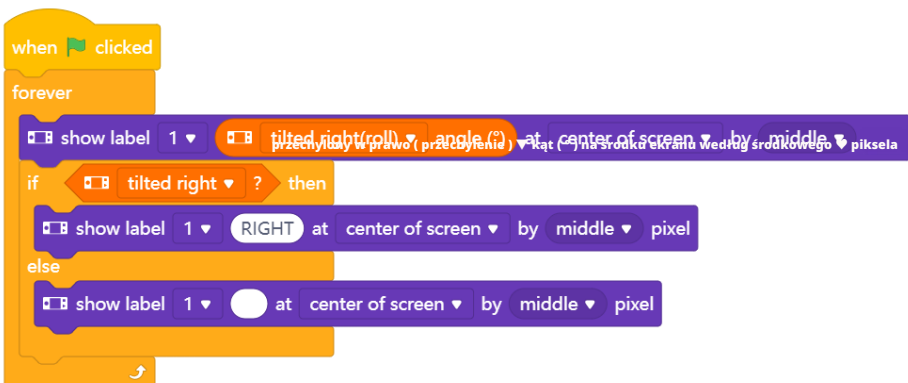
Kąt odchylenia : od  $-180^{\circ}$  do  $+180^{\circ}$

Kąt pochylenia : od  $-90^{\circ}$  do  $+90^{\circ}$

W poniższym programie kąt pochylenia będzie wyświetlany w górnym wierszu, a jeśli zostanie wykryty ruch przechylenia do przodu lub do tyłu, w ostatnim wierszu pojawią się słowa "do przodu" lub "do tyłu". Czy potrafisz wskazać próg wykrywania ?



Spróbuj poeksperymentować z różnymi ustawieniami z menu rozwijanego. Teraz przeanalizowano ruch przechylenia na osi pochylenia / x. Jeśli przełączysz na tilt\_left / right, pamiętaj, aby przełączyć również zgłaszane bloki Boolean:

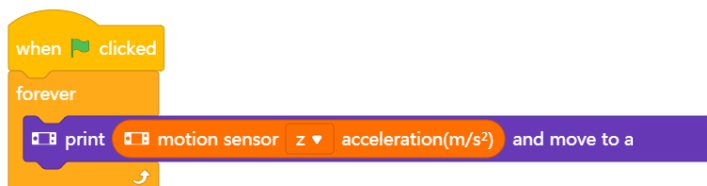


## Blok kodu:

motion sensor x acceleration(m/s<sup>2</sup>)

Do tej pory przyglądaliśmy się ruchom polegającym na przechylaniu, ale czujniki ruchu obejmują jeszcze jeden czujnik — akcelerometr. Mierzy on zmiany prędkości mBot2 w czasie. Może to być zarówno powolne zwiększanie prędkości jazdy, jak i nagłe zatrzymanie, a nawet uderzenie boczne przez inny pojazd. Ponieważ pomiary te opierają się na osiach x, y i z robota, możesz wykryć zderzenie, wyszukując gwałtowną zmianę przyspieszenia. Akcelerometr wskazuje przyspieszenie w m / s<sup>2</sup>.

W poniższym przykładzie programistycznym przyspieszenie mBot2 wzdłuż osi z jest wyświetlane na wyświetlaczu. Poruszaj ręcznie mBot2 i sprawdź wyświetlacz. Jakie wartości pokazuje, gdy robot „zderzy się” z twoją ręką lub książką? Czy możesz zmienić program i zasymulować uderzenie boczne?

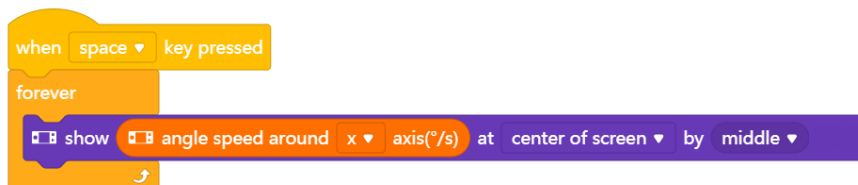


## Blok kodu:

angle speed around x axis(°)

Ten blok kodu pokazuje prędkość, z jaką mBot2 obraca się wokół określonej osi. Nazywa się to również prędkością kątową. Prędkość kątowa jest podawana w stopniach na sekundę (° / s).

W poniższym przykładzie programistycznym prędkość kątowa mBot2 wokół osi x jest wyświetlana na wyświetlaczu.



#### Blok kodu:

rotated angle around x ▼ axis

Ten blok kodu wskazuje kąt, o jaki mBot2 obraca się wokół danej osi. Kąt jest wyświetlany w stopniach (°) i mierzony względnie. Pozycja początkowa jest ustalana po włączeniu mBot2. Dane te są obliczane na podstawie prędkości obrotowej zgłaszanej przez żyroskop. Z czasem wartość może ulegać dryfowi i nie jest zbyt stabilna — spróbuj przechylić mBot2 na bok, korzystając z poniższego programu, i sprawdź wartości. Czy wracają do stanu początkowego?

W poniższym przykładzie programistycznym na wyświetlaczu pokazano kąt, o jaki mBot2 obraca się wokół osi y.



Należy zwrócić uwagę na dwa ważne aspekty: ten blok będzie śledzić ruchy obrotowe większe niż jeden pełny obrót. Użyj tego bloku, aby policzyć, ile razy robot się obrócił. Patrząc na oś z, obrót zgodnie z ruchem wskazówek zegara zwiększy wartości zgłaszane przez ten blok:

rotate clockwise ▼ angle (°)

Natomiast kąt obrotu wokół osi z zmniejsza się:

rotated angle around z ▼

Po 3 obrotach o 360° ostatni blok zgłosi „-1080°”, ale poprzedni blok pokaże tylko „0” (powrót do pozycji początkowej z pominięciem wielokrotnych obrotów). Jeśli następnie obrócimy robota 4 razy w kierunku przeciwnym do ruchu wskazówek zegara (jeszcze jeden raz wcześniej), kąt „rotate\_clockwise” ponownie wynosi 0°, podczas gdy „rotated\_angle\_around\_z axis” pokazuje teraz 360° (wszystkie wartości bez żyroskopu dryf).

Jeśli trzeba uwzględnić jedynie niewielkie ruchy obrotowe wokół osi x i y, omówione wcześniej bloki „kąt przechylenia (\_\_\_)” zapewniają bardziej niezawodny pomiar.

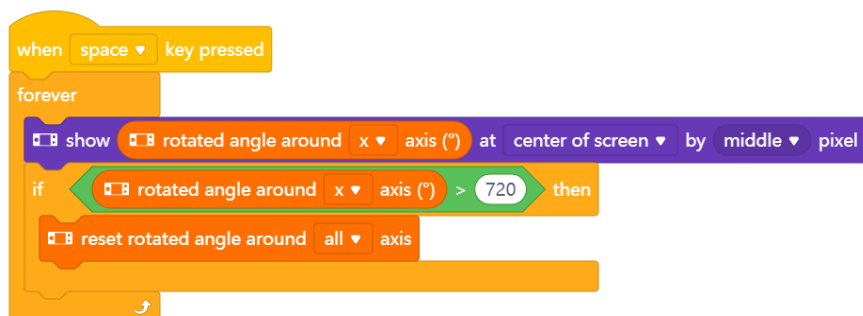
Jest jeden wyjątek: obroty wokół osi odchylenia / Z można mierzyć wyłącznie za pomocą żyroskopu. Jeśli chcesz zmierzyć, o ile stopni mBot2 obraca się w lewo / w prawo, musisz pracować z tymi danymi.

**Blok kodu:**



Jeśli chcesz określić kąt, o jaki obraca się mBot2, łatwiej jest zmierzyć zmianę względem wartości O. Ten blok kodu resetuje zmierzoną wartość kąta do początkowej wartości O, wyznaczając nową pozycję początkową dla przyszłych obrotów.

W poniższym przykładzie programistycznym na wyświetlaczu pokazano kąt, o jaki mBot2 obraca się wokół osi z. Jeśli kąt jest większy niż 720°, zmierzona wartość kąta zostaje ustawiona na O i jest mierzona ponownie.



Pamiętaj, że istnieją dwa bloki do pomiaru obrotu: jeden uwzględnia tylko pełne obroty, a drugi nadal zlicza.

Ten blok resetowania ustawia wszystkie bloki „rotated\_around\_axis” lub wybrane z nich na zero. Nie ma on wpływu na blok „obrót zgodnie z ruchem wskazówek zegara” / „obrót przeciwnie do ruchu wskazówek zegara”.

Ten blok ma własny blok resetowania:



Każde ustawienie na zero dotyczy wyłącznie obrotów zgłaszanych przez sam czujnik żyroskopowy. Kąt przechylenia jest zawsze \_\_\_ mierzony względem idealnie wypoziomowanej płaszczyzny prostopadłej do kierunku wyznaczonego przez środek Ziemi i nie można go dowolnie wyzerować.

## 2. Odtwarzanie i testowanie kilku przykładów programistycznych z użyciem czujnika ruchu

W powyższej tabeli każdy blok kodu czujnika ruchu ma przypisany przykład programistyczny. Odtworzysz te przykłady programistyczne w mBlock i je przetestujesz.

Czujnik ruchu w CyberPi rejestruje ruchy przechylające mBot2. Podczas testowania powyższych przykładów programistycznych; w rzeczywistości potrzebujesz więc tylko CyberPi. Jeśli tak będzie wygodniej, odłącz CyberPi od mBot2 i podczas testowania samodzielnie wykonuj ruchy przechylające CyberPi. Przykłady programistyczne możesz testować w trybie na żywo. Ponieważ bateria znajduje się w nakładce rozszerzającej mBot2, aby uruchomić programy, musisz połączyć CyberPi z komputerem za pomocą kabla USB-C.

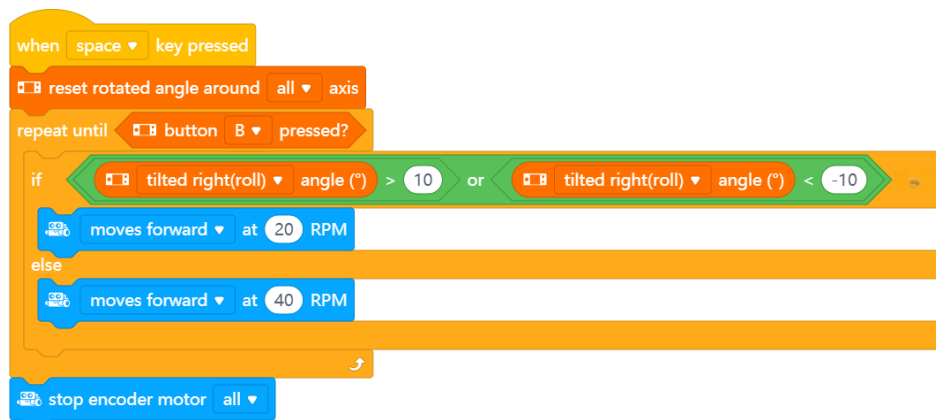
## 3. Wypóbowywanie (25 min)

### Krok 3: Wypóbowuj

Wiesz już dużo o czujniku ruchu w mBot2. Teraz utworzysz własny program komputerowy w mBlock, korzystając z tego czujnika.

W tym zadaniu utworzysz wyboisty tor dla mBot2. Możesz zbudować ten tor na przykład z (małych) kamieni lub plasteliny na kartonie. Chodzi o to, aby mBot2 przejechał po torze. Jeśli mBot2 zostanie (mocno) potrząśnięty, musi jechać wolniej. Na mBot2 możesz również umieścić przedmiot, który nie może spaść podczas jazdy.

Trudne zadanie, prawda? Na szczęście nie musisz wszystkiego wymyślać samodzielnie! Poniżej znajdziesz przykład programistyczny, który możesz rozbudować. W tym przykładzie programistycznym mBot2 zwolni, jeśli będzie wykonywać zbyt duże ruchy przechylania.



Wykorzystaj wiedzę zdobytą w „Kroku 2” tej lekcji. Oczywiście możesz samodzielnie przeprowadzić mnóstwo eksperymentów z różnymi możliwościami programowania w mBlock.

Podczas pracy nad tym zadaniem pomocne będzie zastosowanie następującego planu krok po kroku.

|                                                                 | Wyjaśnienie                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Krok 1: Co chcesz zrobić?                                       | <p>Jaką trasą ma jechać mBot2?</p> <p>Kiedy mBot2 powinien zwolnić?</p> <p>Czy chcesz, aby czujniki mBot2 również wyświetlały dane na wyświetlaczu?</p>                                                                                                                                                                                                                                                             |
| Krok 2: Czego potrzebujesz?                                     | Czego potrzebujesz oprócz mBot2?                                                                                                                                                                                                                                                                                                                                                                                    |
| Krok 3: Jakich bloków kodu potrzebujesz, aby mBot2 jechał?      | <p>Jak zamierzasz sprawić, aby mBot2 jechał?</p> <p>Jakich bloków kodu użyjesz?</p> <p>Przygotuj krótki opis działania swojego programu (pseudokod / język naturalny, schemat blokowy lub UML)</p> <p>Jeśli potrzebujesz dodatkowych wyjaśnień, możesz omówić temat z innymi uczniami lub nauczycielem albo poszukać informacji na ten temat. W mBlock dostępna jest również pomoc dotycząca każdego bloku kodu</p> |
| Krok 4: Jak chcesz wyświetlić dane z czujników na wyświetlaczu? | <p>Jak chcesz wyświetlić dane na wyświetlaczu?</p> <p>Jakich bloków kodu użyjesz?</p> <p>Krótko opisz, jak działa Twój program (pseudokod / język naturalny, schemat blokowy lub UML)</p> <p>Jeśli potrzebujesz dodatkowych wyjaśnień, możesz omówić temat z innymi uczniami lub nauczycielem albo poszukać informacji na ten temat. W mBlock dostępna jest również pomoc do każdego bloku programistycznego</p>    |
| Krok 5: Testowanie i wdrażanie                                  | <p>Czy pierwsza wersja jest gotowa? Przetestuj to! Podczas rundy testowej zapisz obszary do poprawy. Pracuj nad punktami do poprawy, aż Twój mBot2 będzie działał dokładnie tak, jak zamierzałeś.</p> <p>Udało się? Nagraj efekt końcowy i skonsultuj z nauczycielem, czy możesz opublikować go w mediach społecznościowych z hashtagem # mBot2</p>                                                                 |

## 4. Podsumowanie (5 min)

### Krok 4: Podsumowanie

Jak ci poszło z tym zadaniem? Czy podczas jazdy obiekt pozostał stabilnie na mBot2?

W tej lekcji poznaliście czujniki służące do pomiaru ruchu wokół osi i przyspieszenia: żyroskop oraz akcelerometr. Są one zintegrowane w jednym niewielkim komponencie umieszczonym na CyberPi. Te połączone czujniki służą do określania pozycji robota w przestrzeni 3D. Nazywa się je jednostkami pomiaru inercyjnego, IMU. Dowiedzieliście się, gdzie można się z nimi spotkać w życiu codziennym. Wiecie, jakie ruchy przechylania i zmiany ruchu może mierzyć mBot2 oraz jak używać różnych bloków kodu, aby wykorzystać te pomiary w programach z mBot2.

Nadszedł teraz czas na krótką refleksję. Pomyśl samodzielnie i omów to w grupie:

. Jak myślisz, co się udało?

Co mogłoby być jeszcze lepsze?

Które części lekcji były dla Ciebie łatwe, a które sprawiły Ci większą trudność?

O czym chcielibyście uzyskać więcej wyjaśnień?

Kto mógłby Ci w tym pomóc?



## Lekcja 7: Gra sieciowa

Temat: STEAM

Czas trwania: 45 minut

Klasy: 5. i starsze

Poziom trudności: początkujący

### ★ Cele lekcji

Pod koniec tej lekcji uczniowie będą potrafili:

Sprawić, aby wiele robotów mBot2 komunikowało się ze sobą bezprzewodowo, bez potrzeby korzystania z punktu dostępu Wi-Fi

Napisać własny program komputerowy w mBlock do sterowania wieloma robotami mBot2

Zastosować wybór losowy w programie komputerowym

### ★ Wprowadzenie

LAN to skrót od sieci lokalnej. mBot2, wyposażony we wbudowany moduł Wi-Fi, automatycznie ustanawia połączenie sieciowe, gdy w tym samym pomieszczeniu, w promieniu 30 metrów, znajduje się wiele robotów mBot2 lub urządzeń CyberPi. Za pośrednictwem tego połączenia sieciowego roboty mBot2 i urządzenia CyberPi mogą komunikować się ze sobą oraz wykonywać polecenia.

### ✿ Cel

Pod koniec tej lekcji uczniowie będą wiedzieć:

Czym jest połączenie LAN

Jak skonfigurować to połączenie i gdzie je stosować

Jak działa wybór losowy podczas programowania

## Lista kontrolna przed lekcją

### Czego potrzebujesz?

Komputer stacjonarny lub laptop (z wyjściem USB) z zainstalowanym oprogramowaniem mBlock, wersją internetową (również na Chromebooku) albo tablet z zainstalowaną aplikacją mBlock

mBot2 z urządzeniem CyberPi (jeśli to możliwe, na potrzeby kroku „Wypróbuj” — wiele robotów mBot2 lub urządzeń CyberPi)

- Kabel USB-C lub klucz sprzątowy Bluetooth Makeblock

Karty kolorów (zalecany jest kolor niebieski)

## Plan lekcji

Ta lekcja składa się z czterech kroków i trwa łącznie 45 minut.

| Czas trwania | Spis treści                                                                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 minut      | 1. Rozgrzewanie<br>Połączenie LAN w życiu codziennym<br>Jak skonfigurować połączenie LAN z użyciem mBot2 i CyberPi                                                                                                                                                                                                                  |
| 10 minut     | 2. Zajęcia praktyczne<br>Zapoznanie się z różnymi blokami kodu służącymi do konfigurowania połączenia LAN<br>Odtwarzanie i testowanie przykładów programistycznych dotyczących połączenia LAN                                                                                                                                       |
| 25 minut     | Sekcja 3 – Wypróbowywanie<br>Tworzenie własnej gry                                                                                                                                                                                                                                                                                  |
| 5 minut      | 4. Podsumowanie<br>Czas na prezentację: pokaż, co udało Ci się zrobić z robotem, w krótkim, ciekawym filmie do późniejszej dyskusji<br>Jeśli nauczyciel na to pozwoli, udostępnij efekt końcowy w mediach społecznościowych z hashtagem #mBot2LAN<br>Odbicie: Z czego jesteś najbardziej dumny? Co chcesz ulepszyć w swoim robocie? |

## 1. Rozgrzewanie (5 min)

### Krok 1: Rozgrzewka

Ten krok składa się z dwóch części:

1. Połączenie LAN w życiu codziennym
2. Jak skonfigurować połączenie LAN z użyciem mBot2 i CyberPi

#### 1. Połączenie LAN w życiu codziennym

LAN oznacza sieć lokalną i jest określeniem sieci komputerów oraz urządzeń peryferyjnych o określonym zasięgu przestrzennym, zwykle w obrębie budynku. Sieć umożliwia komputerom wymianę danych między sobą w ramach tej sieci oraz, opcjonalnie, z innymi sieciami, takimi jak internet. W związku z tym potrzebują one pewnego rodzaju połączenia (przewodowego, za pomocą kabla lub przewodu, albo bezprzewodowego / Wi-Fi, wykorzystującego fale radiowe) oraz wspólnego „języka” (zwanego protokołem). Zwykle poszczególne komputery łączą się z urządzeniem zarządzającym, które przełącza i kieruje ruchem danych („routerem”). W sieci Wi-Fi jest to dedykowany punkt dostępu, choć router internetowy może również oferować wbudowany punkt dostępu.

Czy potrafisz podać przykłady zastosowania sieci LAN? W tej lekcji mBot2 i CyberPi są różnymi urządzeniami, które komunikują się ze sobą za pośrednictwem połączenia LAN.

#### 2. Jak skonfigurować połączenie LAN między mBot2 a CyberPi

Aby skonfigurować połączenie LAN między mBot2 a CyberPi, nie potrzeba wiele. mBot2 i CyberPi (lub urządzenia mBot2 między sobą) będą automatycznie mogły komunikować się bezprzewodowo, więc nie trzeba w tym celu łączyć się z punktem dostępu Wi-Fi ani routerem.

Czy używany jest router? W takim przypadku wszystkie urządzenia mBot2 i CyberPi muszą zostać ustawione na tym samym kanale routera. Należy to ustawić w ustawieniach routera.

## 2. Zajęcia praktyczne (10 min)

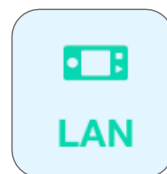
### Krok 2 : Praca praktyczna

Ten krok składa się z dwóch części:

1. Zapoznanie się z różnymi blokami kodu służącymi do komunikacji w sieci LAN
2. Odtwórz i przetestuj kilka przykładów programowania z użyciem komunikacji w sieci LAN

#### 1. Zapoznanie się z różnymi blokami kodu służącymi do komunikacji w sieci LAN

W mBlock dostępnych jest kilka bloków kodu, których można użyć do skonfigurowania połączenia LAN. Te bloki kodu znajdziesz w kategorii „LAN” w obszarze bloków w mBlock. Te bloki kodu są zielone.



Poniższa tabela przedstawia przykład tych bloków kodu służących do konfigurowania połączenia LAN.

#### Blok kodu:



Ten blok kodu umożliwia wysłanie komunikatu do wszystkich pozostałych urządzeń mBot2 i CyberPi w tej samej sieci.

Możesz nadać komunikatowi nazwę, podobnie jak tematowi.

#### Blok kodu:



Zamiast stale sprawdzać, czy odebrano komunikat rozgłoszeniowy, można użyć bloku zdarzeń. Kod dołączony do tego bloku jest wykonywany tylko raz, jeśli zostanie odebrany komunikat z określonym tematem.

W poniższym przykładzie programistycznym dwa urządzenia mBot2 i/lub CyberPi komunikują się ze sobą przez połączenie LAN, korzystając z tych dwóch bloków kodu. Pierwszy program to „jednostka rozgłaszająca”, przeznaczona dla pierwszego urządzenia mBot2 lub CyberPi. Drugi program to kod odbiornika dla drugiego urządzenia mBot2 lub CyberPi. Możesz przesłać obie części kodu do obu urządzeń CyberPi/mBot2, ponieważ kod bazowy zapewnia, że jednostka nadawcza nie odbierze własnego komunikatu.

```
when button A pressed
  broadcast message welcome on LAN
```

```
when receiving welcome broadcast on LAN
  turns left 360 ° until done
  play hi until done
```

Po przesłaniu kodów programu musisz ponownie uruchomić mBot2 i/lub CyberPi.

Teraz, gdy naciśniesz przycisk A na pierwszym mBot2 lub CyberPi, drugi mBot2 lub CyberPi odbierze wysłany komunikat (np. powitanie). Co się stanie, jeśli naciśniesz przycisk B na drugim CyberPi lub mBot2 ? Co trzeba by zmienić, aby wysłać z powrotem inne komunikaty?

#### Blok kodu:

```
broadcast message message with value 1 on LAN
```

Ten blok kodu umożliwia wysłanie komunikatu (z określonym wcześniej tematem) oraz konkretnej wartości (liczby lub tekstu) do wszystkich pozostałych urządzeń mBot2 i CyberPi w sieci.

#### Blok kodu:

W przypadku kodu odbiorczego możesz użyć tego samego bloku zdarzeń i określić temat, na który ma reagować, dokładnie tak jak w poprzednim przykładzie. Aby użyć wartości ( tekstowej lub liczbowej ), służy do tego specjalny blok raportowania :

```
LAN broadcast message value
```

W poniższym przykładzie programistycznym dwa urządzenia mBot2 i/lub CyberPi komunikują się ze sobą przez połączenie LAN, korzystając z tych dwóch bloków kodu. Pierwszy program to „jednostka rozgłaszająca”, przeznaczona dla pierwszego urządzenia mBot2 lub CyberPi. Drugi program to kod odbiornika dla drugiego urządzenia mBot2 lub CyberPi. Możesz przesłać obie części kodu do obu urządzeń CyberPi/mBot2, ponieważ kod bazowy zapewnia, że jednostka nadawcza nie odbierze własnego komunikatu.

when joystick middle pressed ▾

broadcast message robot with value 1 on

when receiving robot broadcast on LAN

show label 1 ▾ LAN broadcast robot value received at center of screen ▾ by middle ▾

Po przesłaniu kodów programu należy ponownie uruchomić mBot2 i / lub CyberPi . Jeśli teraz naciśniesz joystick na pierwszym mBot2 lub CyberPi , drugi mBot2 lub CyberPi odbierze wysłany komunikat ( wartość 1 ) .

## 2. Odtwarzanie i testowanie kilku przykładów programistycznych komunikacji w sieci LAN

W powyższej tabeli każdemu blokowi kodu połączenia LAN towarzyszy przykład programistyczny. Odtworzysz te przykłady programistyczne w mBlocku i je przetestujesz. W przypadku jednego przykładu programistycznego wymyśl własne rozszerzenie.

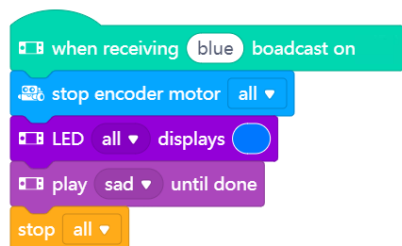
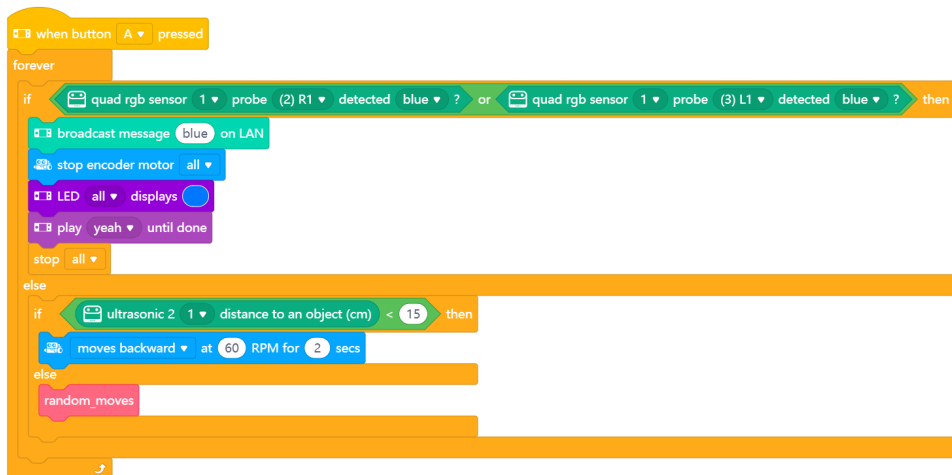
## 3. Wypróbowywanie (25 min)

### Krok 3: Wypróbuj

Wiesz już sporo o połączeniu LAN mBot2. Teraz utworzysz własny program komputerowy w mBlock, korzystając z komunikacji w sieci LAN.

W tym zadaniu samodzielnie zaprogramujesz grę. Nazywa się ona „Poszukiwanie... koloru niebieskiego”. W tej grze grupa robotów mBot2 szuka koloru niebieskiego na podłodze. Jeśli jeden z robotów mBot2 znajdzie ten kolor, poinformuje pozostałe za pośrednictwem komunikatu w sieci LAN i wygra grę.

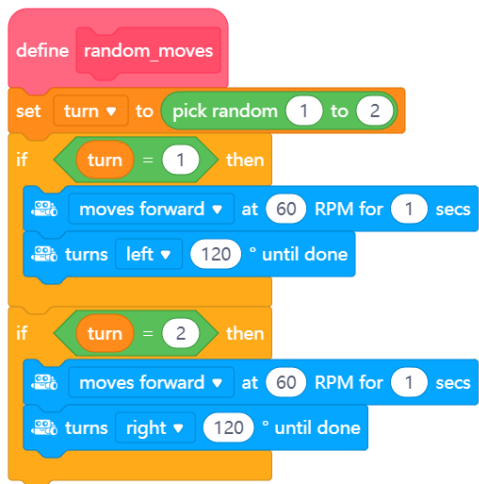
Brzmi dość skomplikowanie! Na szczęście nie musisz samodzielnie programować wszystkiego. Poniżej możesz zobaczyć przykład programistyczny. Ten sam program należy wgrać do wszystkich robotów mBot2, aby wszystkie wykonywały te same instrukcje.



Aby mBot2 poruszał się losowo, możesz użyć następującego bloku kodu:

pick random 1 to 10

Ten blok kodu służy do losowego wybierania wartości z określonego zakresu. Do każdej wartości z tego zakresu można przypisać określoną instrukcję. W ten sposób mBot2 może nieliniowo wybierać spośród wielu instrukcji. Aby ułatwić zrozumienie programu, możesz oddzielić bloki tej instrukcji i umieścić je w bloku niestandardowym, którego następnie możesz użyć w programie głównym:



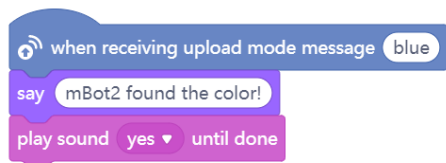
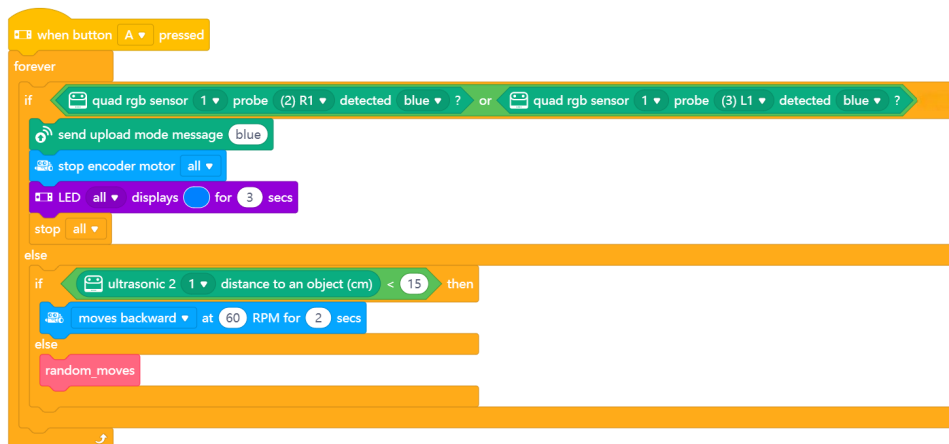
Masz tylko jednego mBot2? W takim razie możesz zasymulować komunikację między dwoma urządzeniami, dodając duszka do sceny mBlocka. Scena to obszar w lewym górnym rogu ekranu mBlocka.

Musisz dodać nowe rozszerzenie do bloków kodu duszka i mBot2 o nazwie „komunikat nadawany w trybie przesyłania”. Umożliwia ono wysyłanie i odbieranie wiadomości między duszkiem a robotem, gdy robot działa w trybie przesyłania, podobnie jak wiadomości wysyłane za pośrednictwem komunikacji w sieci LAN.

|            | Rozszerzenie                                                                        | Dodaj                                                                                                                                                                                                                                        |
|------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dla duszka |  | Kliknij „duszki”<br>Kliknij „Rozszerzenia”<br>Kliknij „Rozszerzenia Sprite”<br>Znajdź rozszerzenie o nazwie „komunikat nadawany w trybie przesyłania” i dodaj je                                                                             |
| Dla mBot2  |                                                                                     | Kliknij „Urządzenia”<br>Kliknij „Rozszerzenia”<br>Kliknij „Rozszerzenia urządzenia”<br>Znajdź rozszerzenie o nazwie „komunikat nadawany w trybie przesyłania” i dodaj je (może nie znajdować się na pierwszej stronie biblioteki rozszerzeń) |



W tym przykładzie programistycznym, gdy mBot2 wykryje kolor, wyśle komunikat do duszka pandy na scenie w mBlock, a panda powie, że mBot2 wykrył kolor.



Możesz skopiować powyższe przykłady programistyczne, ale możesz również wymyślić własne rozszerzenie. Następnie warto skorzystać z poniższego planu krok po kroku. Czy masz już jakiś pomysł, co chcesz stworzyć? Jeśli tak, najpierw omów z nauczycielem, czy jest to wykonalne.

|                                          | Wyjaśnienie                                                                                                                                                                                                           |
|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Krok 1: Co chcesz zrobić?                | Co chcesz zaprogramować?                                                                                                                                                                                              |
| Krok 2: Czego potrzebujesz?              | Jakich materiałów potrzebujesz, aby to zrobić?                                                                                                                                                                        |
| Krok 3: Jakich bloków kodu potrzebujesz? | <p>Jak zamierzasz sprawić, aby mBot2 jeździł i komunikował się?</p> <p>Jakich bloków kodu użyjesz?</p> <p>Przygotuj krótki opis działania swojego programu (pseudokod / język naturalny, schemat blokowy lub UML)</p> |



|                                |                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                | Jeśli potrzebujesz dodatkowych wyjaśnień, możesz omówić temat z innymi uczniami lub nauczycielem albo poszukać informacji na ten temat. W mBlock dostępna jest również pomoc do każdego bloku programistycznego.                                                                                                                   |
| Krok 4: Testowanie i wdrażanie | <p>Pierwsza wersja gotowa? Przetestuj to! Podczas rundy testowej zapisz punkty do poprawy</p> <p>Pracuj nad punktami do poprawy, aż mBot2 będzie robić dokładnie to, co zamierzałeś</p> <p>Udało się? Nagraj efekt końcowy i zapytaj nauczyciela, czy możesz opublikować go w mediach społecznościowych z hashtagem # mbot2LAN</p> |

#### 4. Podsumowanie (5 min)

##### Krok 4: Podsumowanie

Czy udało Ci się zaprogramować grę?

W tej lekcji dowiesz się, czym jest połączenie LAN i gdzie możesz się z nim spotkać w życiu codziennym. Dowiesz się, jak skonfigurować połączenie LAN między różnymi urządzeniami. Dowiesz się również, jak utworzyć własną sieć LAN za pomocą mBot2 i CyberPi. Ponadto dowiesz się, jak dodać wybór losowy do programu komputerowego.

Nadszedł teraz czas na krótką refleksję. Pomyśl samodzielnie i omów to w grupie:

● Jak myślisz, co się udało?

Co można było zrobić lepiej?

Które części lekcji były dla Ciebie łatwe, a które sprawiły Ci większą trudność?

O czym chcielibyście uzyskać więcej wyjaśnień?

Kto mógłby Ci w tym pomóc?

## Lekcja 8: mBot2 do usług

Temat: STEAM

Klasy: 5. i starsze

Czas trwania: 45 minut

Poziom trudności: średni

### ★ Cele lekcji

Pod koniec tej lekcji uczniowie będą potrafili:

. Spraw, aby mBot2 wykonywał zadania za pomocą rozpoznawania mowy i połączenia Wi-Fi

Łącz rozpoznawanie mowy ze zmiennymi, aby tworzyć dialogi

Wysyłaj wiadomości za pośrednictwem komunikacji w chmurze

Używaj wyrażeń logicznych z danymi z czujnika

### ★ Wprowadzenie

Rozpoznawanie mowy to technologia, która przekształca wypowiadane słowa (ale także całe zdania) z powrotem w tekst. Rozpoznanie nawet pojedynczego słowa wyłącznie na podstawie zmian częstotliwości jest dość wymagającym zadaniem obliczeniowym. Aby rozpoznawanie mowy było praktycznie użyteczne, powinno działać niezależnie od głosu danej osoby: wysoki lub niski głos oraz wolne lub szybkie mówienie nie powinny mieć znaczenia. Rozpoznawanie mowy jest częścią interakcji z użytkownikiem w elektronice użytkowej (takiej jak smartfony, telewizory i inteligentny dom), ale także w systemach wspomagających. W tym ćwiczeniu zaprogramujesz krótki dialog, w którym mBot2 będzie działać jako robot usługowy. Później możesz rozszerzyć to zadanie, aby symulować dialog z użytkownikiem również dla innych zadań.

### 🔗 Skupienie

Pod koniec tej lekcji uczniowie będą wiedzieć:

Czym jest rozpoznawanie mowy i jak je stosować

Jak nawiązać połączenie z internetem za pomocą Wi-Fi

Jak używać rozpoznawania mowy i tworzyć dialog z użytkownikiem

Wysyłaj wiadomości do zdalnych komputerów za pośrednictwem komunikacji w chmurze

Używaj zmiennych do opisywania stanu procesu (np. składania zamówienia, interakcji z użytkownikiem)

## Lista kontrolna przed lekcją

### Czego potrzebujesz?

Komputer stacjonarny lub laptop (z wyjściem USB) z zainstalowanym oprogramowaniem mBlock, wersją internetową (również na Chromebooku) albo tablet z zainstalowaną aplikacją mBlock  
mBot2 z urządzeniem CyberPi  
Kabel USB-C lub klucz sprzętowy Bluetooth Makeblock.

## Plan lekcji

Ta lekcja składa się z czterech kroków i trwa łącznie 45 minut.

| Czas trwania | Spis treści                                                                                                                                                                                                                                                                                                                            |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 minut      | 1. Rozgrzewanie<br>Rozpoznawanie mowy w życiu codziennym<br>Poznanie rozpoznawania mowy za pomocą mBot2                                                                                                                                                                                                                                |
| 10 minut     | 2. Zajęcia praktyczne<br>Konfigurowanie połączenia Wi-Fi<br>Programowanie i stosowanie rozpoznawania mowy<br>Komunikacja w chmurze                                                                                                                                                                                                     |
| 25 minut     | 3. Wypróbowywanie<br>Programowanie mBot2, aby stał się robotem kelnerem                                                                                                                                                                                                                                                                |
| 5 minut      | 4. Podsumowanie<br>Czas na prezentację: pokaż, co udało Ci się zrobić z robotem, w krótkim, ciekawym filmie do późniejszej dyskusji<br>Jeśli nauczyciel na to pozwoli, udostępnij efekt końcowy w mediach społecznościowych z hashtagem #mBot2WIFI<br>Refleksja: z czego jesteś najbardziej dumny? Co chcesz ulepszyć w swoim robocie? |

## 1. Rozgrzewanie (5 min)

### Krok 1: Rozgrzewka

Ten krok składa się z dwóch części:

1. Rozpoznawanie mowy w życiu codziennym
2. Poznajemy rozpoznawanie mowy za pomocą mBot2

#### 1. Rozpoznawanie mowy w życiu codziennym

Rozpoznawanie mowy to technologia, która potrafi rozpoznawać słowa i zdania w ludzkim głosie, a następnie przekształca je w tekst. Jest to bardzo wymagające zadanie obliczeniowe, ponieważ algorytmy mogą analizować jedynie zmiany częstotliwości w czasie. Jednak nawet ten parametr różni się w zależności od osoby — wystarczy pomyśleć o niskim lub wysokim głosie, szybkim mówieniu albo dialekcie. Nawet nastrój może zmienić „ton”, w jakim brzmi to samo zdanie. Dziesięciolecia badań w dziedzinie informatyki i lingwistyki pozwalają coraz częściej wykorzystywać te złożone zadania w codziennym życiu.

Niektóre systemy rozpoznawania wymagają pewnej formy „trenowania”, podczas którego użytkownik odczytuje określone teksty znane algorytmowi rozpoznającemu, dostosowując model komputerowy do indywidualnego tonu głosu. Inne rozpoznają tylko bardzo konkretne słowa — są wykorzystywane głównie w systemach dialogowych dla użytkowników. Niektóre systemy potrafią rozpoznawać i tłumaczyć teksty niezależnie od kontekstu.

CyberPi i mBot2 mogą korzystać z rozpoznawania mowy bez potrzeby skomplikowanego programowania. Jednak ponieważ wynikiem jest „tylko” tekst, program musi go przeanalizować i poddać dalszemu przetwarzaniu. Rozpoznawanie mowy jest stosowane w elektronice użytkowej, takiej jak telewizory, smartfony czy inteligentne systemy domowe, aby zapewnić użytkownikowi dodatkowy komfort — umożliwiając nawiązanie połączenia, dodanie spotkania do kalendarza lub zmianę temperatury i oświetlenia w pomieszczeniu za pomocą samego głosu.

Oprócz zapewniania komfortu rozpoznawanie mowy może być bardzo pomocne dla osób z niepełnosprawnościami lub zwiększać bezpieczeństwo, np. kierowca samochodu może zatwierdzić zmianę trasy z powodu nadchodzących korków za pomocą samego głosu, zamiast korzystać z pokręteł lub interfejsu opartego na ekranie.

W ramach tego ćwiczenia wykorzystasz niektóre funkcje rozpoznawania mowy.

## 2. Poznaj rozpoznawanie mowy z mBot2

mBot2 oferuje rozpoznawanie mowy oparte na programowaniu blokowym — jest ono tak proste, jak uruchomienie rozpoznawania na określony czas (np. 2, 3 lub 5 sekund) za pomocą jednego bloku i pobranie rozpoznanego tekstu jako ciągu znaków za pomocą innego bloku. Bloki programistyczne zostaną wyjaśnione poniżej.

To rozpoznawanie mowy obsługuje wiele różnych języków i nie ogranicza się do niewielkiego zestawu wcześniej zdefiniowanych słów. Dlatego jest to bardzo wymagające zadanie obliczeniowe — tak wymagające, że mBot2 ani CyberPi nie mogą obsłużyć go samodzielnie... muszą poprosić o pomoc przez internet. Dlatego przede wszystkim należy ustanowić połączenie z internetem, a także wymagane jest konto użytkownika. Utworzenie konta użytkownika generuje „klucz” umożliwiający zalogowanie się do używanych usług (informacje o wykorzystaniu danych i prywatności znajdują się poniżej). Rozpoznawanie mowy

rejestruje dźwięk, tak jak miało to miejsce w trzeciej aktywności, a następnie wysyła to nagranie do bardzo wydajnej sieci komputerowej w celu przetworzenia. Zajmują się tłumaczeniem z powrotem na tekst i odsyłają tekst do CyberPi / mBot2. Sieć komputerowa jest zwykle określana jako „chmura”. Nie jest to już konkretny komputer, lecz raczej ogromny klaster komputerów, między którymi rozdzielane są zadania obliczeniowe.

Internetu można również używać do przesyłania danych między wieloma urządzeniami CyberPi znajdującymi się w różnych sieciach i lokalizacjach, na przykład w szkole i w domu. Ta usługa cyfrowa również jest przetwarzana w chmurze.

### Informacja dotycząca wykorzystania danych i prywatności:

Rozpoznawanie mowy wymaga połączenia z internetem w celu przesłania nagranego dźwięku i odesłania z powrotem tekstu transkrypcji. Nagrany dźwięk jest przetwarzany na serwerach w Europie, w chmurze Microsoft Azure, ale nie będzie przechowywany na stałe. Aby korzystać z usług, wymagany jest „klucz”, który zostanie automatycznie wygenerowany podczas tworzenia konta użytkownika na Makeblock.com i będzie używany wewnętrznie przez bloki programistyczne. Nie ma potrzeby ręcznego wprowadzania tego klucza w żadnym miejscu.

Oprócz adresu e-mail i samodzielnie wybranego hasła nie są wymagane żadne dodatkowe dane osobowe, takie jak imię i nazwisko, zdjęcia czy płeć...

Połączenie z internetem wymaga dostępu przez Wi-Fi za pomocą identyfikatora SSID (czyli nazwy sieci bezprzewodowej) i hasła – logowanie na podstawie certyfikatu nie jest obsługiwane. Identyfikator SSID i hasło są przechowywane wraz z programem w postaci zwykłego tekstu, dlatego dobrym pomysłem może być użycie tymczasowego hotspotu. Najlepiej użyć smartfona z taryfą z nielimitowanym transferem danych.

## 2. Zajęcia praktyczne (10 min)

### Krok 2: Zajęcia praktyczne:

W tym kroku zapoznasz się z różnymi blokami kodu funkcji AI i IoT.

Ten krok składa się z trzech części:

4. Konfigurowanie połączenia Wi-Fi
5. Programowanie i stosowanie rozpoznawania mowy
6. Przesyłanie wiadomości i komunikacja w sieci LAN i chmurze

### 1. Konfigurowanie połączenia Wi-Fi

Otwórz mBlock i połącz CyberPi w zakładce „Urządzenia”. Przełącz na tryb przesyłania. Po nawiązaniu połączenia znajdź kategorię bloków AI (sztuczna inteligencja) wśród bloków kodu i kliknij ją. Zobaczysz teraz dostępne funkcje, w tym rozpoznawanie mowy (jeśli bloki są wyszarzone, oznacza to, że nie przełączono na tryb przesyłania...).



#### Blok kodu:



Ten blok rozpoczyna próby łączenia z podaną siecią bezprzewodową. Potrzebujesz routera lub punktu dostępu, który umożliwi urządzeniom łączenie się z siecią i dostęp do Internetu.

W niektórych szkołach mogą być dozwolone tylko znane urządzenia — każdy adapter sieciowy (przewodowy lub bezprzewodowy) ma unikatowy numer, czyli adres MAC. Jeśli ten numer jest nieznany zabezpieczonym sieciom, odrzucą one połączenie, nawet jeśli hasło jest prawidłowe.

Najłatwiejszym rozwiązaniem jest wtedy tymczasowy hotspot, na przykład utworzony za pomocą smartfona z taryfą obejmującą stały pakiet danych, używanym wyłącznie przez CyberPi / mBot2 podczas zajęć (sprawdź przepisy obowiązujące w Twojej szkole).

Wszystkie bloki następujące po tym bloku kodu zostaną wykonane natychmiast, więc program nie będzie czekać na nawiązanie połączenia („nieblokujące”). Musisz samodzielnie zaprogramować procedurę oczekiwania za pomocą następnego bloku:

#### Blok kodu:

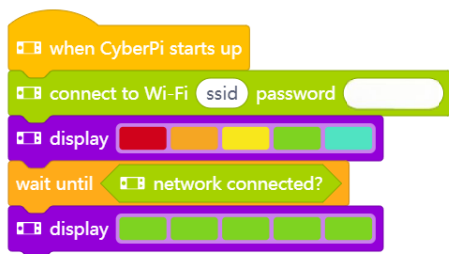


Ten blok logiczny sprawdza, czy udało się nawiązać połączenie z siecią — jeśli tak, zwraca wartość True. Możesz użyć go razem z blokiem oczekiwania:



Ta kombinacja zatrzymuje wykonywanie całego dalszego kodu w tym samym wątku do momentu pomyślnego nawiązania połączenia. Później możesz chcieć uczynić to rozwiązanie bardziej przyjaznym dla użytkownika, informując o próbach połączenia, sygnalizując pomyślne połączenie lub nawet dodając limit czasu, na przykład ponowne uruchomienie, jeśli nie uda się nawiązać połączenia przez 30 s.

Aby korzystać z rozpoznawania mowy, musisz połączyć CyberPi z Internetem. Poniżej zobaczysz przykład programistyczny, którego użyjesz, aby połączyć CyberPi z Internetem. Gdy tylko CyberPi połączy się z internetem, zobaczysz to na wyświetlaczu, a diody LED zaświecą się na zielono.



Te przykłady wykorzystują podejście oparte na zdarzeniach, dlatego podczas uruchamiania jeden wątek programu próbuje włączyć połączenie z internetem. Pozostałe wątki rozpoczynają rozpoznawanie. Jednak na razie nie sprawdzają, czy można ustanowić połączenie z internetem przed kontynuowaniem działania — nie przekazują też użytkownikowi żadnej informacji o tym, co się dzieje, chyba że połączenie się powiedzie. Później możesz zdecydować to zmienić, gdy będziesz wypróbowywać własne pomysły...



Upewnij się, że prawidłowo wpisujesz nazwę (SSID) i hasło sieci Wi-Fi (ilustracja przedstawia przykład!). Oba te elementy uwzględniają wielkość liter. Podłącz mBot2 do komputera za pomocą kabla USB i włącz tryb przesyłania. Odtwórz kod programu przedstawiony powyżej i sprawdź, czy działa. Wskazówka: najlepiej zrobić to za pomocą hotspotu w telefonie komórkowym, ponieważ szkolna sieć często zezwala na łączenie się tylko znanymi urządzeniami (urządzeniom z adresami MAC na białej liście). Upewnij się, że pakiet danych obejmuje korzystanie z Internetu! Teraz podłącz CyberPi do komputera

Teraz mBot2 jest już połączony z Internetem. Możesz teraz komunikować się z mBot2 za pośrednictwem chmury. Nazywa się to Internetem rzeczy (IoT).

Jeśli używasz osobnego CyberPi do wyświetlania statusu (na przykład wysłanego zamówienia do kuchni lub procesu dostawy), musi on być połączony z tą samą siecią Wi-Fi. Dzięki temu oba urządzenia (mBot 2 z CyberPi oraz samodzielny CyberPi) korzystają z tego samego kanału w sieci Wi-Fi. Router / punkt dostępu decyduje o kanałach. Dzięki temu wszystkie urządzenia korzystają z tego samego kanału wymiany danych. Możesz także odwołać się do komunikacji w sieci LAN w lekcji 7.

## 2. Programowanie i stosowanie rozpoznawania mowy

Teraz zaprogramujesz i przetestujesz rozpoznawanie mowy. Uzupełnisz kod programu, który umożliwia mBot2 połączenie z Wi-Fi, o kod rozpoznawania mowy.

Blok kodu:



Ten blok kodu zostanie użyty w poniższym przykładzie — działa odwrotnie niż rozpoznawanie mowy: Za pomocą tego bloku możesz sprawić, że CyberPi / mBot2 odczyta na głos podany tekst. Tekst jest tutaj przesyłany do usługi chmurowej, a zszyntetyzowany plik audio jest odsyłany. Język tekstu jest wykrywany automatycznie, dlatego język mówiony powinien być taki sam jak język tekstu.

Blok kodu:



Rozpoznawanie mowy jest dostępne w 12 różnych językach. Jeśli Twój język ojczysty nie jest obsługiwany, być może możesz użyć języka angielskiego. Ten blok kodu rozpoczyna nagranie audio na określony czas i wysyła je do usługi chmurowej w celu rozpoznania. Aby uzyskać wyniki, potrzebujesz następującego bloku kodu:

#### Blok kodu:

 speech recognition result

Dostęp do wyników rozpoznawania mowy można uzyskać za pomocą tego bloku raportowania . Możesz zapisać wynik w zmiennej , jeśli chcesz użyć go ponownie , wyświetlić go na ekranie lub poddać dalszej analizie ( na przykład sprawdzić , „ czy tekst zawiera słowo „ tak ” ? ” ) .

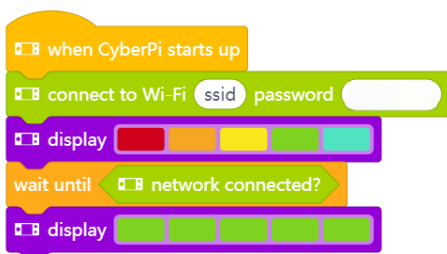
#### Blok kodu:

 translate hello into Chinese ▼

Funkcje AI umożliwiają nawet tłumaczenie tekstów na inne języki . Ten blok kodu jest blokiem raportowania , który umożliwia przetłumaczenie podanego tekstu na wybrany ( obsługiwany ) język . Możesz użyć tego bloku razem z poprzednimi blokami i wiedzą zdobytą podczas wcześniejszych aktywności , aby zbudować translator sterowany głosem : naciśnij przycisk i przetłumacz 5 sekund mowy na język obcy ... Możesz nawet wypróbować różne języki i sprawdzić swoje umiejętności językowe ② .

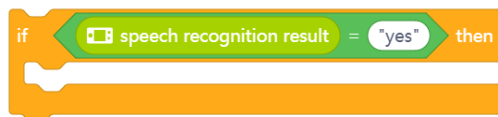
W poniższym przykładzie programistycznym mBot2 został zaprogramowany tak , aby po naciśnięciu przycisku B rozpoznawał mowę przez 3 sekundy . Wynik jest wyświetlany na wyświetlaczu CyberPi, a rozpoznana wypowiedź jest odczytywana na głos. Odtwórz poniższy kod programistyczny i przetestuj go. Sprawdź, co zostało powiedziane i co zostało rozpoznane oraz przetłumaczone z powrotem.

Czy to działa? Następnie sprawdź, co jeszcze możesz zrobić za pomocą rozpoznawania mowy!





Możesz również używać rozpoznawania mowy do sterowania głosowego robotem mBot2. Możesz przypisać określone instrukcje do określonych wyników rozpoznawania mowy, ale zwróć uwagę na to, jak precyzyjnie określasz te wyniki. Na przykład możesz zbudować instrukcję, aby rozpoznawać wyłącznie „tak”, gdy naciśnięty jest przycisk B. Ale co się stanie, jeśli powiesz „zdecydowanie tak” albo „tak, oczywiście”? Wtedy bezpośrednie dopasowanie do...



... zadziałałoby... wynik rozpoznawania byłby inny (ciąg „tak” nie jest równy ciągowi „zdecydowanie tak”). Aby umożliwić szerszy zestaw odpowiedzi oznaczających „tak”, możesz po prostu sprawdzić, czy ciąg odpowiedzi zawiera słowo „tak”:



### 3. Wysyłanie wiadomości i komunikacja w sieci LAN oraz w chmurze

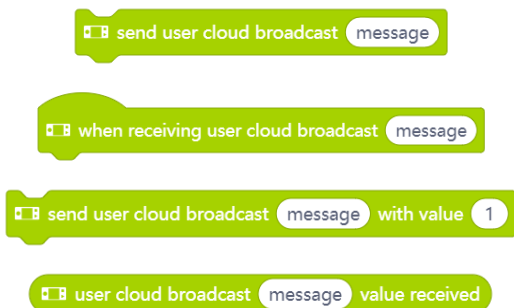
Możesz również użyć wyniku rozpoznawania mowy do interakcji z wieloma robotami mBot2 lub urządzeniami CyberPi. W lekcji 7 poznano już połączenie LAN oraz sposób jego konfigurowania.

Oprócz wysyłania wiadomości w sieci LAN, które wymaga, aby urządzenia znajdowały się „lokalnie” względem siebie, możesz skorzystać z innej usługi chmurowej, aby wysłać wiadomości do urządzeń znajdujących się w zdalnych lokalizacjach — pod warunkiem że mają połączenie z internetem.

Bloki te znajdują się w kategorii bloków IoT (Internet rzeczy).



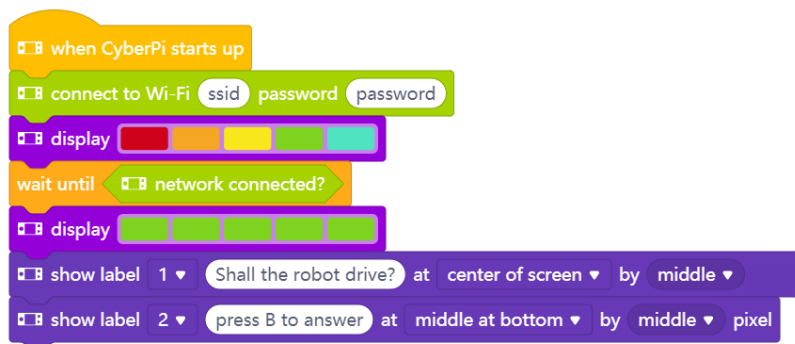
Bloki te działają tak samo jak bloki obsługujące komunikację w sieci LAN:

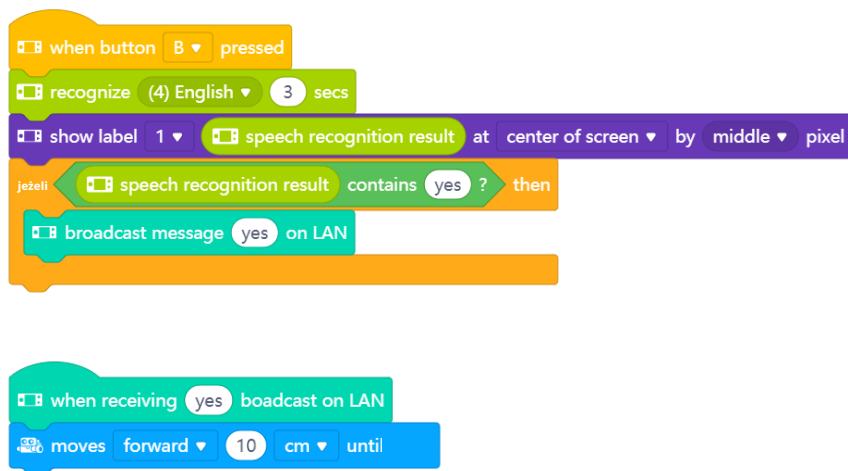


Aby mieć pewność, że zdalna strona może komunikować się wyłącznie z urządzeniami, z którymi ma się komunikować, musisz mieć konto użytkownika Makeblock. Spowoduje to utworzenie „klucza” do komunikacji w chmurze — klucz ten jest ukryty w blokach programistycznych i unikatowy dla użytkownika. Aby 2 zdalne urządzenia mogły komunikować się przez Internet, oba muszą być programowane z tego samego konta użytkownika.

Bloki rozgłaszania w chmurze mają tę zaletę, że umożliwiają komunikację na całym świecie, ale nie nadają się do szybkiej komunikacji – dotarcie wiadomości może zająć kilka sekund. Z kolei rozgłaszanie w sieci LAN jest szybkie, ale działa tylko lokalnie.

Poniżej znajduje się przykład programistyczny podsumowujący powyższe informacje. Najpierw odtwórz program i sprawdź, czy rozpoznawanie oraz komunikacja między CyberPi a innym mBot2 działają prawidłowo. W tym ćwiczeniu musisz przesłać ten sam kod programu w trybie przesyłania do obu urządzeń.





Gdy upewnisz się, że wszystko działa poprawnie, możesz zmodyfikować i rozszerzyć przykład — może o polecenia kierunkowe? Pomyśl o lekcji 1 i ćwiczeniu polegającym na ręcznym przeprowadzeniu mBot2 przez labirynt... Czy możesz zrobić to samo za pomocą poleceń głosowych?

### 3. Wypróbowywanie [ 25 minut ]

#### Krok 3: Wypróbuj

Teraz wiesz już, jak nawiązać połączenie Wi-Fi, zaprogramować rozpoznawanie mowy oraz wdrożyć komunikaty w chmurze lub w sieci LAN. Nadszedł czas, aby rozpocząć samodzielną pracę. Zaprogramujesz uprzejmego robota usługowego — robota kelnera. W niektórych krajach działają już restauracje, w których robot podjeżdża, aby przyjąć zamówienie, i przynosi jedzenie z kuchni. Obejrzyj następujący film: <https://www.youtube.com/watch?v=FFCPKmLAZb4>

Robot kelner, którego zamierzasz zbudować, musi potrafić wykonywać kilka czynności:

- Poruszać się po linii prostej

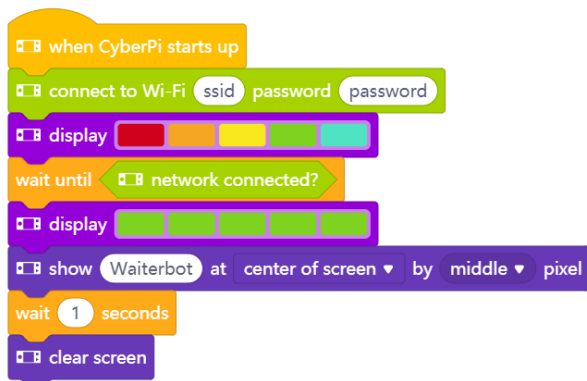
- Zatrzymać się przy czerwonym obszarze (twoim stoliku), przyjąć zamówienie i wysłać je do kuchni

- Opcja dodatkowa: jeśli dostępny jest drugi mBot2 lub CyberPi, wyświetlać zamówienia w kuchni (aby można było już rozpocząć przygotowywanie jedzenia)

- Kontynuować jazdę, aż mBot2 napotka zieloną powierzchnię (kuchnię), a następnie zatrzymać się, aby odebrać jedzenie z kuchni

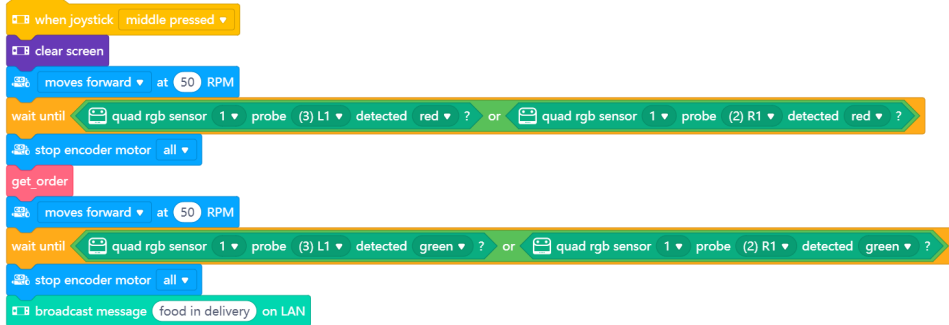
- Opcjonalnie: jeśli dostępny jest drugi mBot2 lub CyberPi, wysłać powiadomienie, że jedzenie jest dostarczane

Sekwencja uruchamiania jest niemal identyczna jak w poprzednich przykładach – pamiętaj, aby wprowadzić właściwy identyfikator SSID i hasło:



Kolejny blok kodu jest zaprogramowany w tym samym polu skryptu, pod poprzednim przykładem. W tym bloku kodu instruujesz mBot2, aby rozpoczął jazdę, a następnie zatrzymał się przy czerwonym znaczniku i przyjął zamówienie. Następnie mBot2 jedzie do zielonego oznaczenia. Kolorowe oznaczenia wskazują stół i kuchnię. Aby ułatwić rozpoczęcie, zakładamy, że znajdują się one w linii prostej. Jeśli wszystko działa, możesz rozbudować to zadanie, wykorzystać wiedzę zdobytą podczas wcześniejszych lekcji i sprawić, aby robot podążał za linią do stołu (na dostarczonej mapie na trasie znajdują się czerwone i zielone oznaczenia, które można do tego wykorzystać). Lekcja 5 wyjaśnia wszystko na temat podążania za linią...

Jako dodatkowe rozszerzenie rozpoznawania mowy, pozostając przy linii prostej między stołem a kuchnią, możesz użyć chmury lub komunikatów w sieci LAN, aby wysłać zamówienie do innego mBot2 lub urządzenia CyberPi i wyświetlić je.

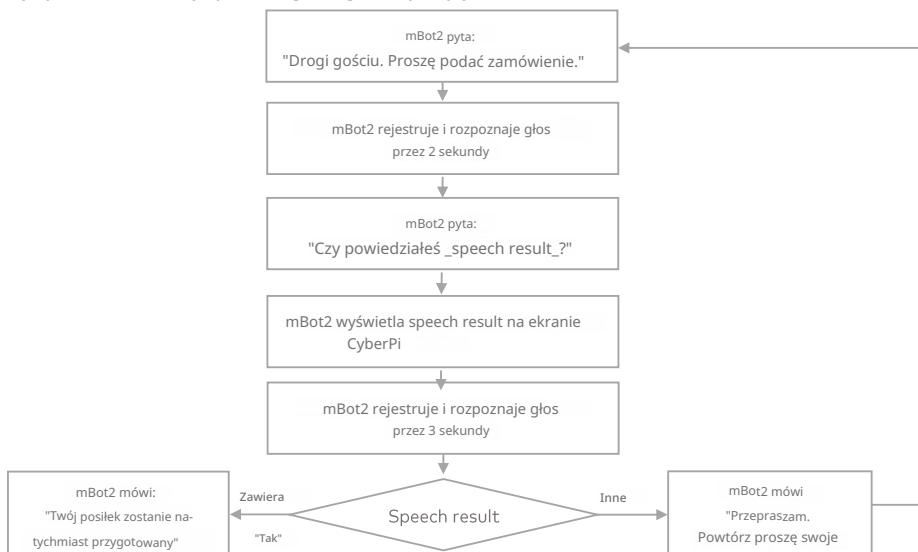


Musisz samodzielnie utworzyć blok „get\_order” w kategorii „moje bloki”. Ułatwia to zachowanie przejrzystości programu głównego – dzięki czemu wystarczy jechać do kolorowych znaczników, odbierając zamówienie pomiędzy.

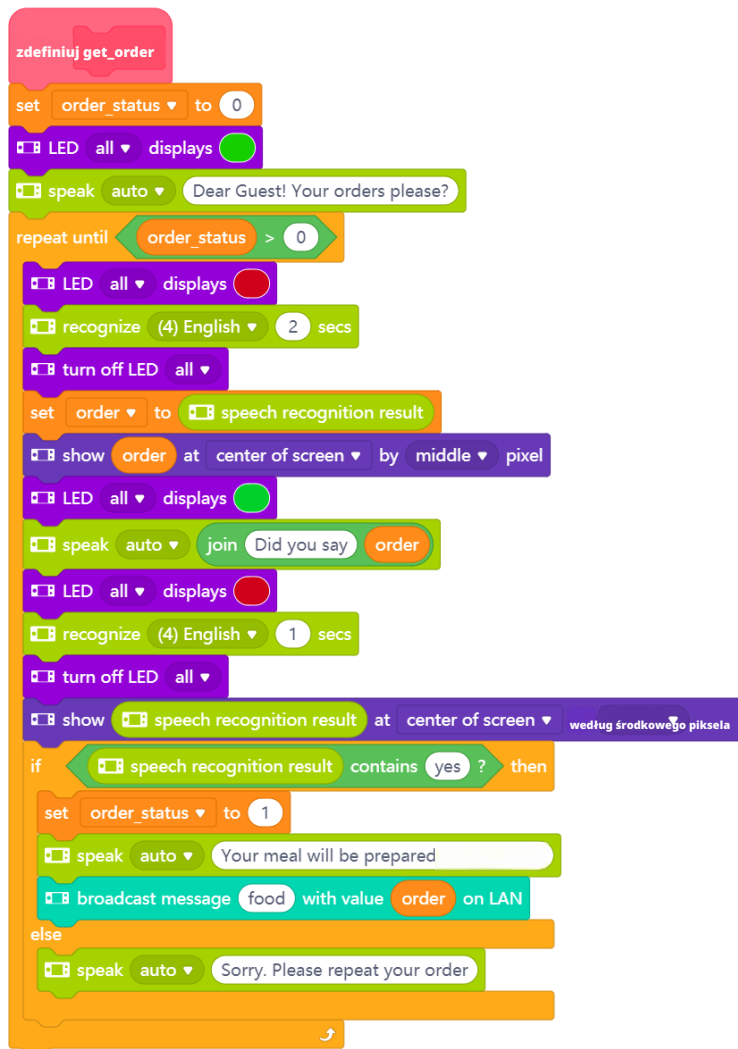
Aby właściwie zdefiniować znaczenie „get\_order”, najpierw zastanówmy się nad procesem zamawiania:

1. Poproś o zamówienie
  2. Rozpoznaj odpowiedź
  3. Sprawdź ponownie, czy kolejność jest prawidłowa
  4. Jeśli tak, przejdź dalej (wyślij zamówienie do kuchni i zakończ sekwencję składania zamówienia)
- Jeśli nie, zapytaj ponownie.

Sprawdzanie to rozsądny sposób na uniknięcie błędnych zamówień. Aby przedstawić te kroki w uporządkowany sposób, można użyć poniższego diagramu przepływu:



Od tego miejsca znacznie łatwiej jest utworzyć niezbędne bloki programistyczne: zaczynasz od bloku kodu „get order”  
. Tworzysz ten blok kodu ponownie w sekcji „Moje bloki”.



Ponadto musisz utworzyć dwie zmienne. W jednej zmiennej będziemy przechowywać wynik rozpoznawania zamówienia, aby w razie poprawnego zrozumienia wszystkiego można było wysłać go do kuchni (uwaga: ponowne sprawdzenie za pomocą odpowiedzi tak/nie nadpisałoby poprzedni wynik rozpoznawania w bloku raportu, dlatego musimy przechowywać rozpoznane zamówienie w zmiennej).



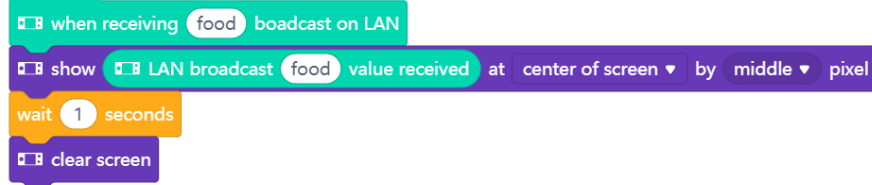
Druga zmienna śledzi cały proces: czy kolejność została ukończona (czyli poprawnie zrozumiana)?

Jeśli nie, proces porządkowania rozpocznie się od początku. To drugie zastosowanie zmiennej informuje o etapie procesu (scenie).

Możesz wykorzystać wiedzę z poprzednich lekcji w programie, np. włączyć diody LED, aby użytkownik zrozumiał, że robot zaraz zacznie mówić (zielony) lub słuchać (czerwony), albo użyć wyświetlacza kelnera mBot2 do wyświetlania danych. Używaj tego w punktach kontrolnych programu, aby sprawdzać, czy kod działa, i przekazywać użytkownikom informacje zwrotne.

Dodatkowo możesz użyć chmury lub komunikatów w sieci LAN, aby wysyłać dane do innego mBot2. Aby ułatwić naukę, w tej lekcji zamieszczono przykłady rozwiązywania konkretnych zadań. Możesz się do nich odwołać, jeśli masz problemy z własnym kodem, lub znaleźć w nich pomysły na rozwiązanie konkretnej sytuacji. Ogólnie rzecz biorąc, lepiej jest najpierw przygotować działający szkic i udoskonalać kod iteracyjnie, niż próbować od razu wszystko ukończyć i zaprogramować perfekcyjnie. Zacznij od sekwencji jazdy i zamiast przygotowywać blok `get_order`, po prostu wywołaj w tym miejscu piknięcie, aby pokazać, że do tej pory wszystko działa. Następnie zajmij się procesem zamawiania, ale pomiń jazdę. W ten sposób możesz szybciej wykonywać kolejne iteracje, ponieważ robot nie musi za każdym razem przejeżdżać między znacznikami, gdy chcesz przetestować kod. Na końcowym etapie połącz je, aby rozwiązać powierzone zadanie.

W powyższych przykładach programistycznych wysyłano również komunikat do drugiego mBot2 jako dodatkową opcję. Na przykład w procesie zamawiania mówisz, że po potwierdzeniu zamówienia przez klienta mBot2 wysyła komunikat za pośrednictwem chmury lub komunikatów w sieci LAN do drugiego mBot2 albo CyberPi, aby wyświetlać przychodzące zamówienia w kuchni. Teraz musisz zaprogramować drugiego mBot2 tak, aby odbierał komunikat.

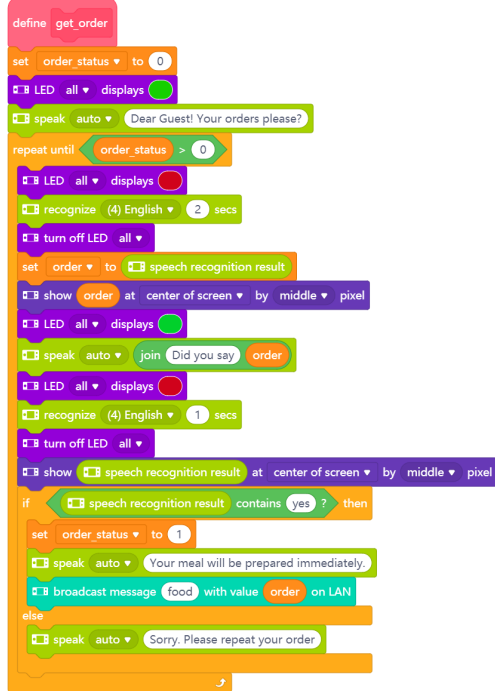


W poniższym przykładzie programistycznym kuchnia odbiera zamówienie i wyświetla je. Utwórz poniższy przykład programistyczny w tym samym skrypcie. Umieść go obok pierwszego przykładu programistycznego. Dzięki temu skrypt pozostanie przejrzysty.



W przykładzie programistycznym „sterowanie robotem” zaprogramowano, że po wykryciu zielonego obszaru do klienta wysyłana jest wiadomość w chmurze, aby poinformować go, że jego zamówienie jest w drodze. Jest to również zaprogramowane w tym samym polu skryptu, obok pozostałych bloków — możesz użyć innego CyberPi (jako inteligentnego urządzenia na stole klienta), aby poinformować o statusie zamówienia:

Po zaprogramowaniu wszystkiego obszar skryptu będzie wyglądał tak:



Musisz przesłać ten kod do wszystkich urządzeń mBot2 i / lub CyberPi, których chcesz używać . Może to być ten sam kod, ponieważ urządzenia łączą się automatycznie przez sieć LAN lub chmurę – a urządzenie wysyłające wiadomość nie zareaguje na nią .

Utwórz linię prostą, wzdłuż której będzie poruszać się mBot2, i upewnij się, że znajduje się na niej szerokie czerwone i zielone oznaczenie . Prześlij kod i sprawdź, czy robot kelner obsługuje zamówienia klientów .

Oczywiście, zawsze jest miejsce na ulepszenia. Dostosuj kod tam, gdzie Twoim zdaniem można go ulepszyć lub uczynić bardziej eleganckim. Może zaczniesz od robota podążającego za linią, który może przyjmować zamówienia? Albo rozwiń to ćwiczenie i stworzysz osobnego kelnera mBot2 przyjmującego zamówienia oraz robota dostawczego mBot2, który będzie dostarczał jedzenie? Możesz współpracować z innymi zespołami w klasie i zintegrować ich roboty...

Omówcie podczas podsumowywania dotychczasowych postępów możliwe rozszerzenia, takie jak te, i opracujcie przypadek użycia z wieloma robotami — nie musi to być restauracja...

## 4. Podsumowanie (5 min)

### Krok 4: Podsumowanie

Czy udało Ci się zaprogramować robota-kelnera?

Na tej lekcji nauczyłeś się rozpoznawania mowy i syntezy mowy oraz tego, jak stosować je za pośrednictwem usługi chmurowej. Udało Ci się połączyć mBot2 z Internetem, a tym samym wysyłać dane lokalnie, a nawet zdalnie.

Tworzenie dialogu, na przykład podczas zamawiania posiłku, było częścią ćwiczenia, a Ty nauczyłeś się filtrować odpowiedź tekstową pod kątem określonych słów wyzwajających, aby prowadzić bardziej „naturalną” rozmowę, bez oczekiwania, że użytkownik będzie odpowiadać wyłącznie określonymi odpowiedziami.

Nadszedł teraz czas na krótką refleksję. Pomyśl samodzielnie i omów to w grupie:

- Jak myślisz, co się udało?

Co można było zrobić lepiej?

Które części lekcji były dla Ciebie łatwe, a które sprawiły Ci większą trudność?

O czym chcielibyście uzyskać więcej wyjaśnień?

Kto mógłby Ci w tym pomóc?

## Lekcja 9: mBot2 w terenie

Temat: STEAM

Czas trwania: 45 minut

Klasy: 5. i starsze

Poziom trudności: początkujący

### ★ Cele lekcji

Pod koniec lekcji uczniowie będą potrafili:

Włączyć uczenie maszynowe do programowania mBot2

Spraw, aby wiele funkcji technicznych mBot2 współdziałało ze sobą

Nawiąż komunikację między robotem a komputerem, mimo że robot działa niezależnie w trybie przesyłania, umożliwiając interakcję między sprzętem fizycznym (robotem) a duszkiem na scenie oprogramowania mBlock

### ★ Wprowadzenie

Uczenie maszynowe jest podstawową formą sztucznej inteligencji (AI). Sztuczna inteligencja odnosi się do urządzeń lub maszyn naśladowujących ludzką inteligencję. Na podstawie zebranych danych uczenie maszynowe umożliwia systemom komputerowym lub robotycznym wybieranie działań na podstawie historii danych i prawdopodobieństwa, bez potrzeby jawnego formułowania algorytmu.

### 🔗 Skupienie

Pod koniec lekcji uczniowie będą wiedzieć:

Czym jest uczenie maszynowe i jak je stosować,

Jak sprawić, aby mBot2 działał na podstawie warunków,

Jak różne funkcje techniczne mBot2 współdziałają w ramach ekosystemu,

Nawiąż protokół komunikacyjny między robotem uruchamiającym programy niezależnie a komputerem

## Lista kontrolna przed lekcją

Czego potrzebujesz?

Komputer stacjonarny lub laptop (z wyjściem USB) z zainstalowanym oprogramowaniem mBlock, wersja internetowa ( także dla Chromebooka )

mBot2 z urządzeniem CyberPi

Przewód USB-C lub klucz sprzętowy Bluetooth Makeblock

Kamera internetowa w laptopie lub zewnętrzna

Drukarka ( opcjonalnie ) albo papier, ołówki i markery do rysowania lub smartfon do wyświetlania obrazów

## # Plan lekcji

Ta lekcja składa się z czterech kroków i trwa łącznie 45 minut.

| Czas trwania | Spis treści                                                                                                                                                                                                                                                                                                                |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 minut      | <b>1. Rozgrzewanie</b><br>Uczenie maszynowe w życiu codziennym<br>Poznanie uczenia maszynowego                                                                                                                                                                                                                             |
| 25 minut     | <b>2. Zajęcia praktyczne</b><br>Zastosowanie uczenia maszynowego<br>Uczenie maszynowe w połączeniu z mBot2 w mBlock<br>Ustanawianie protokołu komunikacyjnego wewnątrz uruchomionych programów                                                                                                                             |
| 10 minut     | <b>3. Wypróbowywanie</b><br>Zadanie końcowe: ekosystem myszy                                                                                                                                                                                                                                                               |
| 5 minut      | <b>4. Podsumowanie</b><br>Czas na prezentację: pokaż swój ekosystem w zabawnym, krótkim filmie do późniejszej dyskusji.<br>Jeśli nauczyciel wyrazi zgodę, udostępnij efekt końcowy w mediach społecznościowych, używając hashtagu #mBot2<br>Odbicie: Z czego jesteś najbardziej dumny? Co chcesz ulepszyć w swoim robocie? |

## 1. Rozgrzewanie (5 min)

### Krok 1: Rozgrzewka

Ten krok składa się z trzech części:

1. Uczenie maszynowe w życiu codziennym
2. Poznawanie rozszerzenia uczenia maszynowego w mBlock
3. Zastosowanie uczenia maszynowego

#### 1. Uczenie maszynowe w życiu codziennym

Uczenie maszynowe opiera się na podstawach statystycznych: algorytm otrzymuje dane treningowe, takie jak zdjęcia różnych obiektów wraz z ich opisami. Następnie podczas fazy uczenia algorytm stara się jak najlepiej rozróżnić te zdjęcia i przyporządkować każde z nich do właściwej kategorii. Po tej fazie wyuczone wzorce są testowane na nowych zbiorach danych, a proces uczenia może zostać wzmocniony. Różnica w porównaniu z „klasycznymi” programami sortującymi polega na tym, że wcześniej nie określa się jasno, jak rozstrzygać między różnymi kategoriami, tak jak można rozróżniać kolory.

Dzięki „uczeniu się na podstawie danych z przeszłości” program komputerowy może lepiej dostosowywać się do zmieniających się sytuacji.

Pomyśl o rozpoznawaniu twarzy w telefonie: za pierwszym razem po skonfigurowaniu tej funkcji telefonowi trudniej jest prawidłowo Cię rozpoznać, ale z czasem algorytm „uczy się” rozpoznawać Twoją twarz także w różnych warunkach. Uczenie maszynowe jest szeroko stosowane w klasyfikacji obrazów (np. w mediach społecznościowych) do automatycznego analizowania tego, co przedstawiają przesłane zdjęcia: kota, psa, meble, ludzi itd.

Uczenie maszynowe uważa się za „słabą sztuczną inteligencję”, ponieważ program komputerowy nie jest świadomy siebie ani nie potrafi „myśleć” tak jak my – jego obecne zachowanie zależy jednak od zebranych i przeanalizowanych danych.

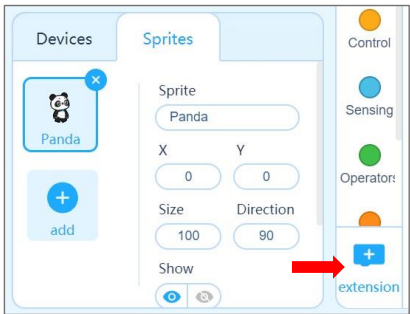


## 2. Poznanawanie rozszerzenia uczenia maszynowego w mBlock

W mBlock znajduje się narzędzie Uczenie maszynowe. Należy dodać do narzędzie z biblioteki rozszerzeń. Postępuj zgodnie z poniższymi instrukcjami:

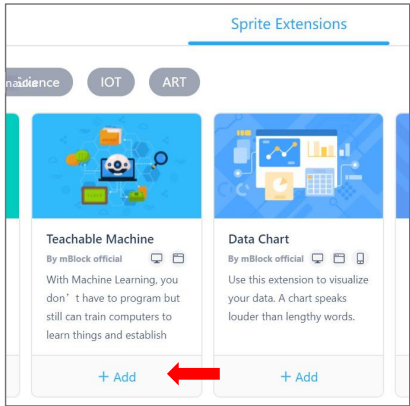
Kliknij „Duszki”.

. Kliknij „rozszerzenia” (na dole, po-  
środku ekranu).



Teraz przejdziesz do biblioteki roz-  
szerzeń. U góry ekranu kliknij „Rozsze-  
rzenia Sprite”.

Znajdź rozszerzenie „Teachable Ma-  
chine” i dodaj je.

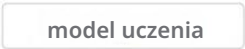


Rozszerzenie zostanie teraz dodane do  
bloków kodu . Rozszerzenie możesz roz-  
poznać po skrótce TM w kolorze zielonym .



Upewnij się, że masz działającą  
kamerę internetową .

Kliknij przycisk „Model uczenia” .





Rozszerzenie zawiera model uczenia, który możesz „uczyć” rozróżniania zestawów zdjęć wykonanych za pomocą kamery internetowej. Podstawowy model, który widzisz, składa się z 3 kategorii. Jeśli chcesz utworzyć ich więcej, kliknij „zbuduj nowy model” i wybierz liczbę kategorii.

Kilka wskazówek:

Zastanów się wcześniej nad liczbą kategorii i sposobem rozpoznawania. Nie możesz później dodać kategorii — musisz zacząć wszystko od początku.

Wybierz 1 dodatkową kategorię, która będzie używana jako „tło”. Jeśli więc chcesz rozpoznawać 2 różne zwierzęta na podstawie zdjęć trzymanyh przed kamerą internetową, wybierz 3 kategorie: trzecia będzie przeznaczona na sytuację, w której nie jest wyświetlane żadne zdjęcie zwierzęcia. W przeciwnym razie „zmusisz” model do podjęcia decyzji, np. między kotem a psem, nawet jeśli kamera internetowa pokazuje puste tło.

Tło powinno być na ogół proste i nieruchome (nie: klasa z uczniami, kamera internetowa skierowana na uczniów itp.) oraz wyraźnie kontrastować z rozpoznawanymi obiektami.

Obrazy będą analizowane wyłącznie na komputerze, a wynik będzie można zapisać. Same obrazy nie będą przechowywane! Jeśli więc zapiszesz plik programu w mBlock i otworzysz go ponownie później, nie zobaczysz oryginalnych obrazów, na których trenowano model. Zawsze możesz dodać nowe obrazy do istniejących kategorii, ale nie możesz usuwać pojedynczych obrazów (tylko wszystkie obrazy i wyniki uczenia z danej kategorii).

Możesz bezpośrednio „uczyć” kategorie za pomocą obrazów zrobionych kamerą. W tym ćwiczeniu użyjesz standardowych 3 kategorii:

1. Możesz zacząć od mBot2. Umieść go na biurku i ustaw kamerę tak, aby uzyskać dobry widok — lub trzymaj go stabilnie w dłoni, jeśli korzystasz z laptopa z wbudowaną kamerą. Następnie kliknij przycisk „Learn”. Przytrzymaj przycisk, aż kamera zarejestruje co najmniej 20 próbek. Nad mozaiką próbek zobaczysz liczbę próbek zapisanych dla tej kategorii. Nadaj tej kategorii nazwę, na przykład „mBot2”.
2. Teraz wybierz coś innego, np. zeszyt, przytrzymaj go przed kamerą i kliknij przycisk „Learn”. Przytrzymaj przycisk, aż kamera zarejestruje co najmniej 20 próbek. Nadaj tej kategorii inną nazwę, na przykład „zeszyt”.
3. Teraz nagraj „tło”. Należy tego użyć, gdy przed kamerą nie znajduje się żaden z pozostałych obiektów. Kliknij przycisk „Learn” i przytrzymaj go, aż kamera zarejestruje co najmniej 20 próbek. Nadaj tej kategorii inną nazwę, na przykład „tło”.

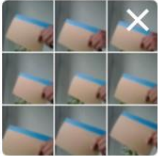
36 Samples



mBot2

Learn

41



40,0%

notebook

Learn

20 Samples



nothing

Learn

Teraz „wytrenowano” model uczenia maszynowego do rozpoznawania różnych obiektów (mBot 2, notatnika i tła), które algorytm będzie odtąd rozpoznawać na obrazie z kamery. Dla każdej kategorii widoczny jest pasek przedstawiający wartość procentową. Wartość ta oznacza poziom pewności algorytmu co do wyniku rozpoznawania. Możesz zwiększyć poziom pewności, zwiększając liczbę próbek i wprowadzając niewielkie zmiany, takie jak zmiana kąta ustawienia obiektu. Pomoże to kamerze jeszcze lepiej i szybciej rozpoznawać obiekty.

Możesz przetestować rozpoznawanie na żywo bez konieczności programowania na tym etapie. Dodawaj obrazy, aż rozpoznawanie będzie szybkie i dokładne (bez przełączania się między wynikami i przy wyświetlaniu wysokiej wartości > 90%). Czy jesteś zadowolony? Następnie kliknij „Użyj modelu”. Teraz wrócisz do ekranu programowania na poziomie duszka.

Pojawiły się teraz trzy nowe bloki kodu. Jeden blok będzie wyświetlał wynik rozpoznawania (wpisany przez Ciebie), drugi poziom pewności dla danej kategorii, a trzeci będzie blokiem logicznym używanym w instrukcjach warunkowych. Pozwala on wybrać kategorię z menu rozwijanego. Po kliknięciu jednego z bloków kodu pojawi się nowe okno. To okno rozpoznawania, w którym wyświetlany jest obraz z kamery oraz wynik rozpoznawania. Teraz kamera internetowa działa jako inteligentny czujnik rozpoznający różne obiekty lub obrazy.



## 2. Zajęcia praktyczne (25 min)

### Krok 2 : Praca praktyczna

Ten krok składa się z trzech części:

1. Zastosowanie uczenia maszynowego
2. Uczenie maszynowe w połączeniu z mBot2
3. Nawiązywanie połączenia LAN

#### 1. Zastosowanie uczenia maszynowego

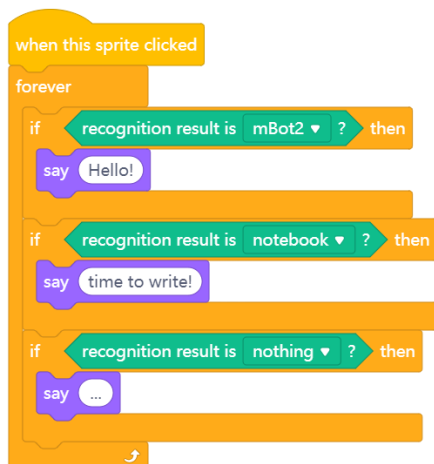
Nauczyłeś(-aś) model uczenia maszynowego dwóch różnych obiektów: mBot2 i notesu, a także tła. Teraz możesz używać wyniku rozpoznawania w kodzie programu tak jak dowolnych innych bloków czujników. Możesz używać instrukcji warunkowych, aby rozróżniać obiekty i przypisywać określone działanie osobno do każdego wyniku.

Teraz spróbujesz sprawić, aby duszek (w tym przypadku panda) mówił, gdy kamera internetowa rozpozna nauczony obiekt.

Po prawej stronie widzisz przykład programistyczny.

Odtwórz przykład programistyczny i zobacz, co robi panda, gdy kamera internetowa rozpozna nauczzone przez Ciebie obiekty.

Czy się udało? Następnie sprawdź, jakie są inne możliwości. Czy potrafisz sprawić, aby panda skakała lub chodziła?

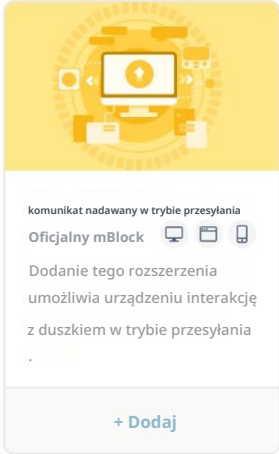


## 2. Uczenie maszynowe w połączeniu z mBot2

Teraz wiesz już, jak używać rozszerzenia uczenia maszynowego do rozpoznawania obrazów i sterować za jego pomocą duszkiem, na przykład pandą. Teraz nauczysz się, jak komunikować się z komputerem z mBot2, na którym działa jego własny program. W tym celu mBot2 musi pozostać „połączony” z komputerem — odbywa się to za pomocą kabla USB lub klucza sprzętowego Bluetooth Makeblock. Klucz sprzętowy pozwala robotowi poruszać się niezależnie od komputera; jeśli używasz kabla USB, pamiętaj, aby wykonywać za pomocą mBot2 tylko niewielkie ruchy, takie jak małe obroty i pokonywanie krótkich odległości.

Po nawiązaniu tego połączenia duszek (w tym przypadku panda) może wysyłać dane do mBot2. mBot2 odbiera te dane, a program działający na robocie może na nie reagować.

Musisz dodać do bloków kodu duszka i mBot2 nowe rozszerzenie o nazwie „komunikat nadawany w trybie przesyłania”.

|            | Rozszerzenie                                                                      | Dodaj                                                                                                                                                                                                                                                           |
|------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dla duszka |  | <p>Kliknij „duszki”</p> <p>Kliknij „Rozszerzenia”</p> <p>Kliknij „Rozszerzenia Sprite”</p> <p>Znajdź rozszerzenie o nazwie „komunikat nadawany w trybie przesyłania” i dodaj je</p>                                                                             |
| Dla mBot2  |                                                                                   | <p>Kliknij „Urządzenia”</p> <p>Kliknij „Rozszerzenia”</p> <p>Kliknij „Rozszerzenia urządzenia”</p> <p>Znajdź rozszerzenie o nazwie „komunikat nadawany w trybie przesyłania” i dodaj je (może nie znajdować się na pierwszej stronie biblioteki rozszerzeń)</p> |

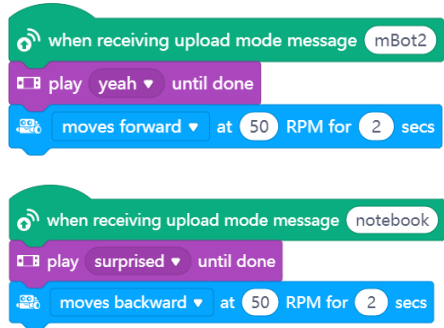
W ten sposób rozszerzenie zostanie dodane zarówno dla mBot2, jak i dla duszka. W mBlocku pojawia się w kolorze ciemnoniebieskim. Kliknij nowe rozszerzenie i zobacz różne bloki kodu z nim związane. Zasadniczo rozszerzenie dodaje protokół komunikacyjny. Umożliwia wysyłanie i odbieranie wiadomości między duszkiem a robotem, gdy robot działa w trybie przesyłania.

### 3. Nawiązywanie połączenia z mBot2

Zainstalowano już wszystkie niezbędne rozszerzenia sprzętowe i programowe, aby mogły się one ze sobą komunikować. Teraz zaprogramujesz mBot2 tak, aby był sterowany za pomocą uczenia maszynowego. Chodzi o to, aby mBot2 wykonywał działania na podstawie wyników rozpoznawania. Wcześniej zaprogramowałeś podobne zadanie tylko dla duszka (mBot2, notebooka i tła). Teraz wyniki rozpoznawania muszą

być wysyłane i odbierane przez połączenie między Bot2 a komputerem (przewodowo lub bezprzewodowo za pośrednictwem klucza sprzętowego Bluetooth). Kod odbierający wykorzystuje zdarzenia, dlatego odpowiedni kod jest wykonywany po odebraniu wiadomości. W przypadku zdarzeń nie trzeba sprawdzać w pętli, czy dostępny jest czujnik lub wiadomość, dzięki czemu kod jest wydajniejszy i łatwiejszy do zrozumienia.

Musisz utworzyć dwa programy komputerowe – jeden dla urządzenia (mBot2) i jeden dla duszka korzystającego z dodatku Machine Learning:



Po przeprogramowaniu powyższych bloków kodu upewnij się, że komputer pozostaje połączony z mBot2, jeśli używasz kabla (nie odłączaj kabla). Najlepiej byłoby połączyć je przez Bluetooth za pomocą klucza sprzętowego Bluetooth Makeblock, ponieważ zapewnia to mBot2 większą swobodę poruszania się.

|                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Jeśli używasz klucza sprzętowego:</p>  | <p>Podłącz klucz sprzętowy do komputera i włącz mBot2. Należy sparować mBot2 z kluczem sprzętowym Bluetooth. W tym celu kliknij przycisk na kluczu sprzętowym. Dioda LED zacznie migać, a klucz sprzętowy spróbuje połączyć się z najbliższym kontrolerem Makeblock (takim jak mBot2 lub inne urządzenie). W sytuacji klasowej upewnij się, że każdy zespół wykonuje tę czynność osobno, aby zapewnić prawidłowe sparowanie klucza sprzętowego z mBot2. Po zakończeniu parowania usłyszysz sygnał dźwiękowy z mBot2. Oznacza to, że komputer i mBot2 nawiązały połączenie bezprzewodowe.</p> |
|----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Klucz sprzętowy Bluetooth Makeblock zapamięta ostatnio sparowane urządzenie i natychmiast połączy się z nim ponownie.

Jeśli nie masz klucza sprzętowego Bluetooth Makeblock, możesz sprawdzić zgodność Bluetooth w komputerze, korzystając z następującego łącza: [https://www.yuque.com/makeblock-help-center-en / cyberpi / bluetooth - compatibility](https://www.yuque.com/makeblock-help-center-en/cyberpi/bluetooth-compatibility)

Teraz przełącz mBlock w tryb przesyłania i połącz mBot2 z mBlock. Kliknij przycisk „Połącz” i wgraj program komputerowy. Możesz teraz rozpocząć testowanie!

### 3. Wypróbowywanie (25 min)

#### Krok 3: Wypróbuje

Masz już całą wiedzę i wszystkie umiejętności potrzebne do korzystania z mBot2 w każdej sytuacji. Teraz będziesz pracować z ekosystemem małej myszy. Mysz żyje w naturze, gdzie musi szukać pożywienia i wody.

Czyha tam również wiele niebezpieczeństw, takich jak drapieżniki żywiące się myszami.

W tym zadaniu zaprogramujesz mBot2 tak, aby zachowywał się jak mała mysz szukająca pożywienia i uciekająca przed drapieżnikami.

Znajdź zdjęcie lecącego orła i trochę kukurydzy. Kukurydza jest pożywieniem małej myszy. Orzeł jest łowcą. Możesz wydrukować te obrazy lub użyć smartfona, aby wyświetlić je przed kamerą internetową. Jeśli chcesz, możesz również odręcznie narysować obrazki przedstawiające pożywienie i drapieżnika. Po prostu zastosuj tę samą metodę, którą wybierzesz do „uczenia” ich swojego algorytmu za pomocą rozszerzenia uczenia maszynowego i do testowania programu.

Zaprogramuj mBot2 tak, aby gdy pokażesz kamerze kukurydzę, mBot2 poruszał się do przodu (jeśli używasz kabla, wykonaj niewielki ruch), a gdy pokażesz obraz orła, poruszał się do tyłu.

Możesz również dodać więcej kodu: jeśli mała mysz zobaczy jedzenie, mówi „mniem”, a gdy zobaczy orła, możemy usłyszeć bicie jej serca.

Podczas tworzenia programu komputerowego do tego zadania pomocne jest skorzystanie z następującego planu krok po kroku.

|                                          | Wyjaśnienie                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Krok 1: Co chcesz zrobić?                | Co chcesz zaprogramować?                                                                                                                                                                                                                                                                                                                                                                                                          |
| Krok 2: Czego potrzebujesz?              | Czego potrzebujesz oprócz mBot2?                                                                                                                                                                                                                                                                                                                                                                                                  |
| Krok 3: Jakich bloków kodu potrzebujesz? | <p>Jak zamierzasz sprawić, żeby mBot2 się poruszał?</p> <p>Jakich bloków kodu użyjesz?</p> <p>Przygotuj krótki opis działania swojego programu (pseudokod / język naturalny, schemat blokowy lub UML)</p> <p>Jeśli potrzebujesz dodatkowych wyjaśnień, możesz omówić temat z innymi uczniami lub nauczycielem albo poszukać informacji na ten temat. W mBlock dostępna jest również pomoc do każdego bloku programistycznego.</p> |



|                                |                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Krok 4: Testowanie i wdrażanie | <p>Pierwsza wersja gotowa? Przetestuj to! Podczas rundy testowej zapisz punkty do poprawy</p> <p>Pracuj nad punktami do poprawy, aż mBot2 będzie robić dokładnie to, co zamierzałeś</p> <p>Udało się? Nagraj efekt końcowy i zapytaj nauczyciela, czy możesz opublikować go w mediach społecznościowych z hashtagem # mbot2</p> |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## 4. Podsumowanie (5 min)

### Krok 4: Podsumowanie

Czy udało Ci się odtworzyć ekosystem myszy?

W tej lekcji dowiedziałeś się o uczeniu maszynowym i o tym, jak je stosować. Wiesz także, jak korzystać z komunikacji między duszkiem a mBot2 (i dlaczego najlepiej, aby była bezprzewodowa) oraz jak wykorzystywać ją do wymiany danych między nimi. Ponadto udało Ci się sprawić, że różne funkcje techniczne mBot2 współdziałają ze sobą.

Nadszedł teraz czas na krótką refleksję. Pomyśl samodzielnie i omów to w grupie:

Jak myślisz, co się udało?

Co mogłoby być jeszcze lepsze?

Które części lekcji były dla Ciebie łatwe, a które sprawiły Ci większą trudność?

O czym chcielibyście uzyskać więcej wyjaśnień?

Kto mógłby Ci w tym pomóc?



## Ochrona środowiska



Zużyty sprzęt elektroniczny oznakowany zgodnie z dyrektywą Unii Europejskiej, nie może być umieszczany łącznie z innymi odpadami komunalnymi. Podlega on selektywnej zbiórce i recyklingowi w wyznaczonych punktach. Zapewniając jego prawidłowe usuwanie, zapobiegasz potencjalnym, negatywnym konsekwencjom dla środowiska naturalnego i zdrowia ludzkiego. System zbierania zużytego sprzętu zgodny jest z lokalnie obowiązującymi przepisami ochrony środowiska dotyczącymi usuwania odpadów. Szczegółowe informacje na ten temat można uzyskać w urzędzie miejskim, zakładzie oczyszczania lub sklepie, w którym produkt został zakupiony.



Produkt spełnia wymagania dyrektyw tzw. Nowego Podejścia Unii Europejskiej (UE), dotyczących zagadnień związanych z bezpieczeństwem użytkowania, ochroną zdrowia i ochroną środowiska, określających zagrożenia, które powinny zostać wykryte i wyeliminowane.

Niniejszy dokument jest tłumaczeniem oryginalnej instrukcji obsługi, stworzonej przez producenta.

Szczegółowe informacje o warunkach gwarancji dystrybutora / producenta dostępne na stronie internetowej <https://serwis.innpro.pl/gwarancja>

Produkt należy regularnie konserwować (czyścić) we własnym zakresie lub przez wyspecjalizowane punkty serwisowe na koszt i w zakresie użytkownika. W przypadku braku informacji o koniecznych akcjach konserwacyjnych cyklicznych lub serwisowych w instrukcji obsługi, należy regularnie, minimum raz na tydzień oceniać odmienność stanu fizycznego produktu od fizycznie nowego produktu. W przypadku wykrycia lub stwierdzenia jakiegokolwiek odmienności należy pilnie podjąć kroki konserwacyjne (czyszczenie) lub serwisowe. Brak poprawnej konserwacji (czyszczenia) i reakcji w chwili wykrycia stanu odmienności może doprowadzić do trwałego uszkodzenia produktu. Gwarant nie ponosi odpowiedzialności za uszkodzenia wynikające z zaniedbania.

Importer: **INNPRO**

INNPRO Robert Błędowski sp. z o.o.  
ul. Rudzka 65c  
44-200 Rybnik, Polska  
tel. +48 533 234 303  
[hurt@innpro.pl](mailto:hurt@innpro.pl)  
[www.innpro.pl](http://www.innpro.pl)

Producent Makeblock Co., Ltd. Address: Floor 4, Building C3, Nanshan iPark, No.1001 Xueyuan Avenue, Nanshan District, Shenzhen, Guangdong Province, 518000,

Podmiot odpowiedzialny

Makeblock Europe BV

Oostergracht 48

3763 LZ Soest

Holandia

support@xtool.com