EdChallenge!

Zbudujmy alarm antywłamaniowy

Powierzono ci cenny skarb, którego masz strzec. Podstępni złodzieje chcą się do niego dostać! Stwórz alarm antywłamaniowy, aby skarb pozostał bezpieczny i nietknięty.

Aby sprostać wyzwaniu i zbudować alarm antywłamaniowy, musisz skonstruować alarm przy użyciu robota Edison. Twój alarm musi:

- wykorzystywać co najmniej jeden z czujników Edisona, który ma spowodować zadziałanie alarmu, oraz
- zrobić coś, co odstraszy intruza.

www.meetedison.com

CC-BY-SA ABIX Edukacja | robotedison.pl | cyfrowaszkola.waw.pl

Edison
robotics for everyone

EdBuild

Budujemy EdCrane

Poziom trudności:

1 2 3 4 5

EdCrane EdBuild opracowany przez Microbric Education jest objęty licencją Creative Commons Attribution-ShareAlike 4.0 International License.

EdBuild – EdCrane

EdCrane jest zdalnie sterowanym dźwigiem, który można zaprogramować przy użyciu kodów kreskowych zamieszczonych w tej instrukcji, w połączeniu z dostępnym do pobrania programem, i którym można sterować przy pomocy zwykłego pilota do TV lub DVD. Gotowa konstrukcja umożliwia obracanie się dźwigu na podstawie, zgodnie z ruchem wskazówek zegara i w przeciwnym kierunku, a także podnoszenie i opuszczanie magnetycznego „haka”.

Poziom trudności konstrukcji

Poziom trudności konstrukcji EdCrane został oceniony na 4.

Poziom trudności	Zalecany wiek	Niezbędne umiejętności programowania
1	8+	- Kody kreskowe i zdalne sterowanie
2	8+	- Kody kreskowe i zdalne sterowanie
3	8+	- Kody kreskowe i zdalne sterowanie
4	10+	- Kody kreskowe i zdalne sterowanie ***ORAZ*** - EdWare ***LUB*** - EdPy
5	12+	- EdPy

Do zbudowania dźwigu i korzystania z jego funkcji potrzebne będą podane poniżej umiejętności programowania.

Dowiedz się więcej na temat programowania z użyciem Edisona, odwiedzając następujące strony:

- **Kody kreskowe:** Odpowiednie od 4 lat wzwyż. Nie zakłada się ani nie wymaga wcześniejszej styczności z programowaniem. www.meetedison.com/robot-activities/youre-a-controller/
- **EdWare:** Odpowiednie dla uczniów w wieku 11-13 lat. Rozumienie podstaw programowania może okazać się pomocne. www.meetedison.com/robot-programming-software/edware/
- **EdPy:** Odpowiednie dla uczniów w wieku 13-16 lat. Rozumienie podstaw programowania może okazać się pomocne. www.meetedison.com/robot-programming-software/edpy/

Przygotowanie

Zanim rozpoczniesz konstruowanie dźwigu, sprawdź czy masz wszystko, czego potrzebujesz i czy potrafisz go zaprogramować. Przyda się też zapoznanie się ze sposobem tworzenia konstrukcji przy użyciu robota Edison.

Czego potrzebujesz

Do zbudowania EdCrane potrzebne są:

- 1 robot Edison
- 1 EdCreate – zestaw konstruktora przeznaczony dla robota Edison
- 1 pilot do TV lub DVD
- 1 wydruk strony z kodami kreskowymi (z tej instrukcji)
- dostęp do komputera z Internetem

Zaleca się, aby przed rozpoczęciem budowy dźwigu hak magnetyczny używany w EdCrane został złożony przez osobę dorosłą lub pod jej nadzorem. Ponieważ hak wykorzystuje prawdziwy magnes i posiada niewielkie elementy składowe należy trzymać go w miejscu niedostępnym dla dzieci poniżej 3 lat. Szczegółowe informacje znajdują się w części dotyczącej składania haka magnetycznego.

Wymagane programowanie

Aby programować EdCrane, musisz zaprogramować w pilocie cztery różne polecenia, wykorzystując kody kreskowe umieszczone w tej instrukcji.

Musisz też zaprogramować robota Edison, za pomocą EdWare lub EdPy, języka programowania w robotyce. Linki do potrzebnych programów znajdują się w instrukcji.

Budowanie z Edisonem

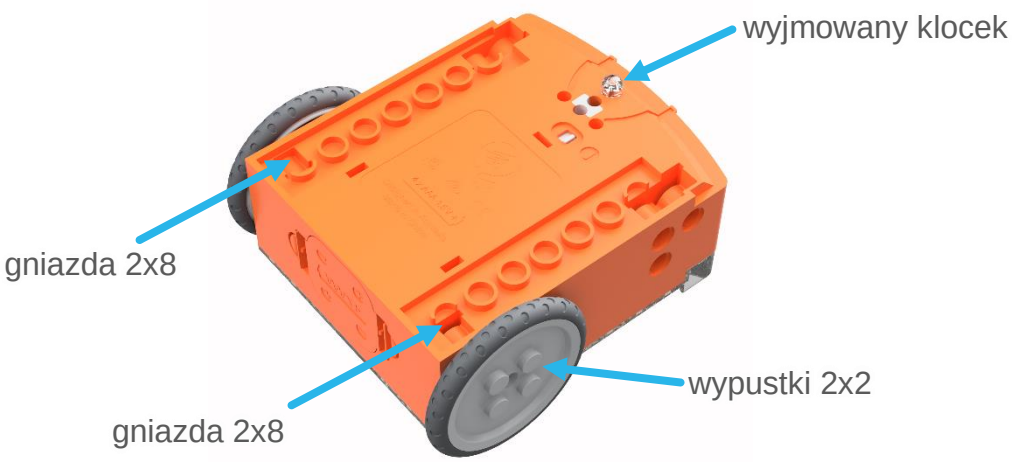
Używając Edisona w połączeniu z zestawem konstruktora EdCreate lub innym kompatybilnym systemem klocków LEGO można tworzyć różne konstrukcje.

Na górnej i dolnej powierzchni robota Edison można przymocować klocki. Elementy łączące (kołki, *pegs*) mogą być mocowane do boku Edisona. Po ściągnięciu kół Edisona osie krzyżowe można umieścić w napędzanych gniazdach.

Na boku Edisona znajdują się trzy rodzaje otworów: otwory sworzniowe (*pin holes*), otwory wypustkowe (*stud holes*) i gniazda dla osi krzyżowych. Patrząc na poniższe rysunki można dowiedzieć się więcej na temat sposobów łączenia i tworzenia konstrukcji z użyciem Edisona.

Połączenia z boku i na górze Edisona

Rodzaj otworu	Opis	Do łączenia
Otwór sworzniowy	Otwór o pełnej głębokości, przeznaczony dla kołków łączących.	
Otwór wypustkowy	Otwór o głębokości normalnej wypustki, w której można umieszczać „półkołki”.	
Gniazdo osi krzyżowej	Napędzane, obracające się gniazdo.	



Połączenia na spodzie i kołach Edisona

Zbudujmy EdCrane!

EdCrane jest zdalnie sterowanym dźwigiem z hakiem magnetycznym, który można podnosić i opuszczać. Magnetyczny hak EdCrane może podnosić lub opuszczać małe przedmioty z metali żelaznych, jak metalowe spinacze do papieru czy niewielkie śrubki. W zestawie EdCreate znajduje się metalowy kołek, który można przymocować do jednej z części zestawu, np. belki z trzema otworami, i używać razem z hakiem magnetycznym.

A oto jak wygląda gotowy EdCrane, używający haka magnetycznego:

Krok 1: Programowanie Edisona

Jak zaprogramować EdCrane

Zanim rozpoczniesz konstruowanie z użyciem części zestawu EdCreate, musisz zaprogramować kody do zdalnego sterowania i wgrać je do robota Edison, korzystając z kodów kreskowych podanych w instrukcji.

Podobnie jak inne konstrukcje EdBuild tworzone przy użyciu EdCreate, EdCrane reaguje na zdalne sterowanie pilotem do TV lub DVD. EdCrane nie wykorzystuje jednak kodów kreskowych w ten sam sposób. EdCrane stosuje polecenia zdalnego sterowania w połączeniu z wgranym programem, który musisz pobrać. Wgrany program instruuje EdCrane jakie działanie ma wykonać po wykryciu określonego kodu zdalnego sterowania.

Konieczne będzie wgranie programu do zdalnego sterowania do robota Edison, korzystając z EdWare lub EdPy. Program można wgrać przed lub po skonstruowaniu dźwigu.

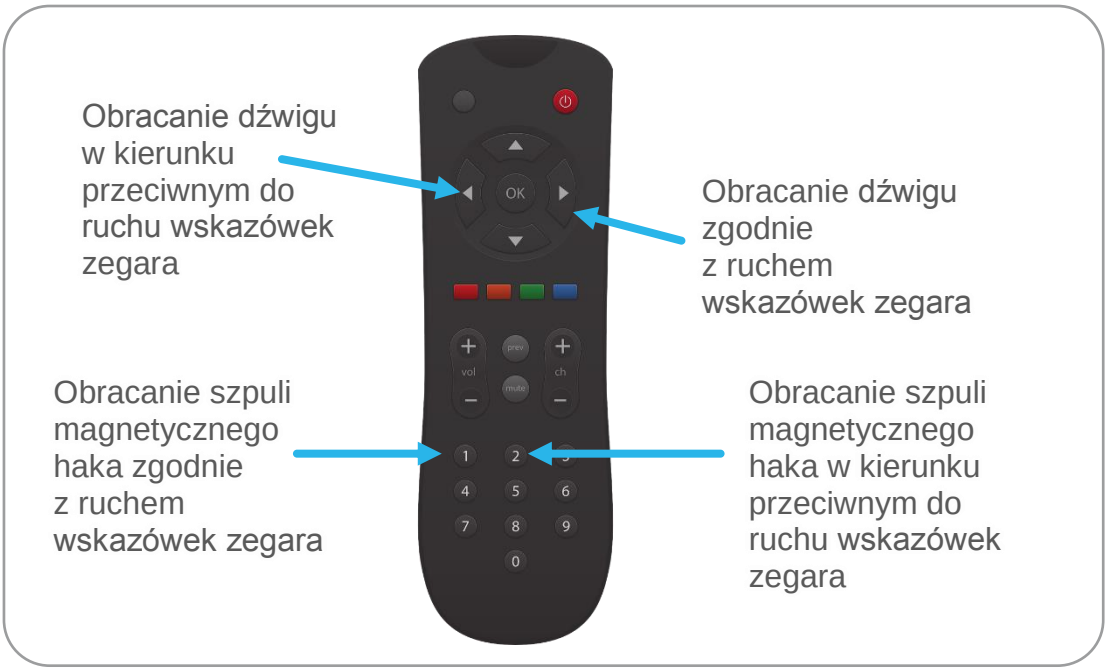
Programowanie kodów kreskowych

Na następnej stronie znajdują się kody kreskowe do zdalnego sterowania robotem Edison. Wydrukuj tę stronę, aby robot Edison mógł przejechać nad kodami kreskowymi.

UWAGA – Podane kody kreskowe do zdalnego sterowania są tymi samymi kodami, których używa się, aby zdalnie sterować ruchami robotów Edison. W związku z tym, po początkowym powiązaniu działania robota z przyciskiem, reakcja robota będzie polegać na zademonstrowaniu domyślnego działania przypisanego do danego kodu kreskowego. Na przykład, „TV/DVD remote code #1” domyślnie powoduje jazdę robota do tyłu.

Gdy wgrasz program i naciśniesz na robocie przycisk „play” (trójkątny), robot wykona zaprogramowane działanie, zamiast domyślnego ruchu przypisanego do kodu kreskowego.

Sugerowane ustawienie przycisków na pilocie dla EdCrane



Wydrukuj tę stronę!

Programowanie Edisona – kody kreskowe

Dźwig wykorzystuje robota Edison, który należy zaprogramować przy użyciu czterech kodów kreskowych.

Robot Edison steruje ruchami EdCrane na podstawie (obracanie zgodnie z ruchem wskazówek zegara lub w przeciwnym kierunku) oraz ruchem szpuli, która zwija lub odwija linkę haka, obracając się zgodnie z ruchem wskazówek zegara lub w przeciwnym kierunku, aby podnieść lub opuścić hak. Przejedź robotem Edison nad następującymi kodami kreskowymi, programując każdy z nich w powiązaniu z przyciskiem na pilocie.

Odczytywanie kodu kreskowego

1. Postaw Edisona po prawej stronie, przodem do kodu kreskowego
2. Naciśnij trzykrotnie przycisk „record” (okrągły)
3. Edison pojedzie do przodu i zeskanuje kod kreskowy
4. Naciśnij na pilocie TV/DVD przycisk, którym chcesz uruchamiać daną funkcję

Kod kreskowy – TV/DVD remote code #1 – obracaj szpulą linki haka magnetycznego zgodnie z ruchem wskazówek zegara

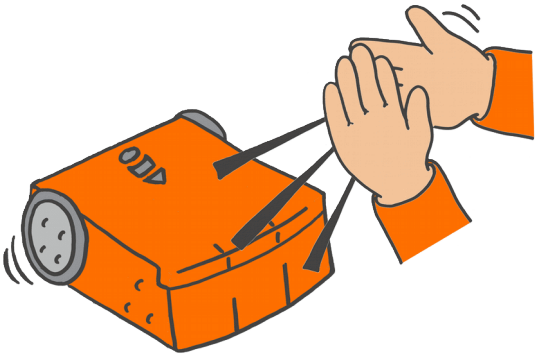
Kod kreskowy – TV/DVD remote code #2 – obracaj szpulą linki haka magnetycznego w kierunku przeciwnym do ruchu wskazówek zegara

Kod kreskowy – TV/DVD remote code #3 – obracaj EdCrane w kierunku przeciwnym do ruchu wskazówek zegara

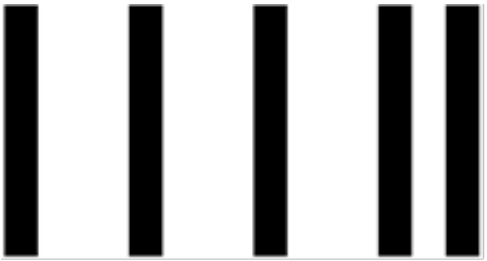
Kod kreskowy – TV/DVD remote code #4 – obracaj EdCrane zgodnie z ruchem wskazówek zegara

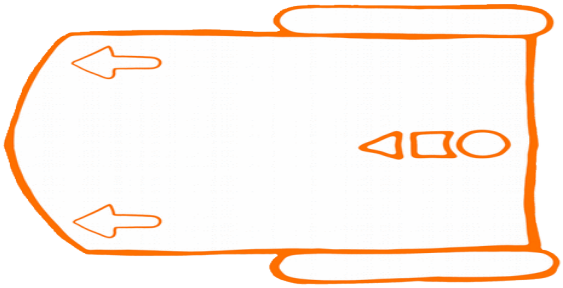
CC-BY-SA ABIX Edukacja | robotedison.pl | cyfrowaszkola.waw.pl

Jazda sterowana klaskaniem



Edison posiada czujnik dźwiękowy, który może wychwycić głośne dźwięki, takie jak klaskanie dłońmi. Poniższy kod kreskowy ma wbudowany program. Program ten sprawia, że Edison wykonuje polecenie, gdy wychwyci głośne dźwięki.



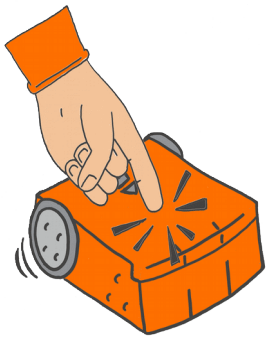


Odczytaj kod kreskowy

Umieść Edisona na powyższym zarysie.
Naciśnij trzy razy przycisk nagrywania (okrągły przycisk).
Edison szybko przejedzie do przodu i zeskanuje kod kreskowy.

Co dalej zrobić z Edisonem?

Umieść Edisona na podłodze lub stole. Naciśnij przycisk odtwarzania (trójkątny przycisk). Teraz klaśnij raz w dłonie. Edison skręci w prawo.
Klaśnij w ręce dwa razy. Edison pojedzie naprzód.



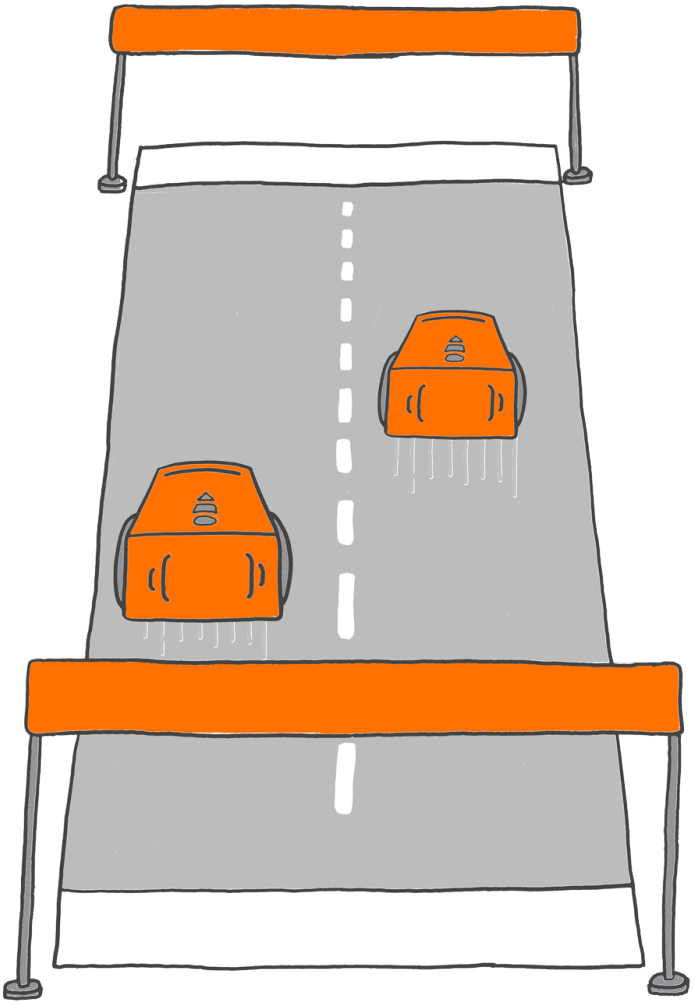
Jeśli Edison nie słyszy Twojego klaskania, możesz oprzeć palec na jego szczycie.

CC-BY-SA – ABIX Edukacja | robotedison.pl | cyfrowaszkola.waw.pl

Ćwiczenie

Zróbmy sobie wyścig dwóch robotów Edison.
Możesz stworzyć swój własny tor wyścigowy. Nie zapomnij o zeskanowaniu kodu kreskowego przez Edisona.
Przygotuj się do wyścigu!
Klaśnij w dłonie lub dotknij Edisona dwa razy, aby pojechał naprzód. Kiedy Edison się zatrzyma, ponownie klaśnij dwa razy.

Edison, który pierwszy znajdzie się na mecie wygrywa!



CC-BY-SA – ABIX Edukacja | robotedison.pl | cyfrowaszkola.waw.pl

Ćwiczenie 2 – Jazda sterowana klaskaniem

W tym ćwiczeniu uczniowie programu Edison używają kodu kreskowego, aby uruchomić program, który umożliwi im prowadzenie Edisona za pomocą klaskania w dłonie.

Jak to działa

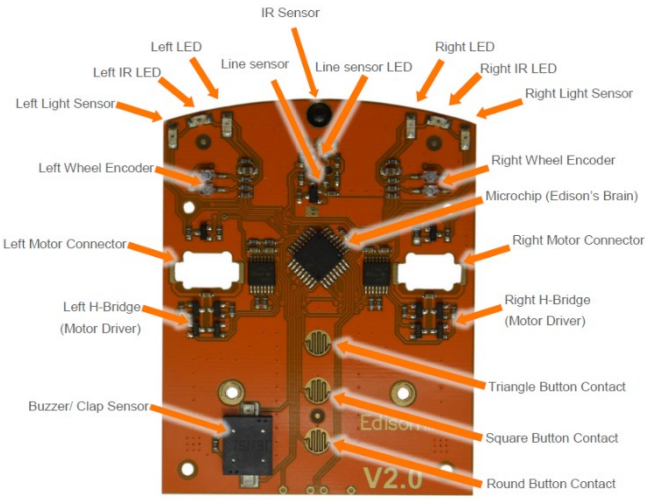
Program sterujący uruchamiany klaśnięciem wykorzystuje czujnik dźwiękowy robota Edison w celu wykrycia głośnych dźwięków, takich jak klaskanie. Wstępnie ustawiony program odwróci Edisona w prawo po wykryciu jednego klaśnięcia i będzie go prowadził naprzód przez kilka sekund, gdy wykryte zostaną dwa klaśnięcia.

Wskazówki i podpowiedzi

- W przeglądzie części Edisona pomocne może być podkreślenie położenia czujnika dźwięku Edisona, zanim uczniowie rozpoczną ćwiczenie. Możesz odnieść się do ilustracji użytych w ćwiczeniu. Aby uzyskać szczegółową instrukcję techniczną pobierz Schemat płyty głównej Edisona.
- Roboty mogą mieć problem z wykrywaniem dźwięków, gdy występuje wysoki poziom szumu w tle. Gdy uczniowie umieszczą palec w pobliżu czujnika dźwięku na swoim Edisonie da to taki sam efekt jak klaskanie.

Rozszerzenie ćwiczenia

1. Zbuduj prawdziwy tor wyścigowy z przeszkodami i zakrętami, aby móc się po nim poruszać. Może to stanowić dodatkowe wyzwanie dla uczniów, wymagające od nich używania zmiennie pojedynczych i podwójnych klaśnięć, aby kontrolować Edisona.
2. Wykonaj tory wyścigowe z lewoskrętnymi zakrętami, zachęcając uczniów do korzystania z kreatywnych umiejętności rozwiązywania problemów.
3. Przeprowadźcie eksperyment, aby dowiedzieć się jakie dźwięki inne niż klaskanie i stukanie może wykryć Edison. Zastanów się czy któryś z tych dźwięków może być używany, aby kontrolować Edisona.



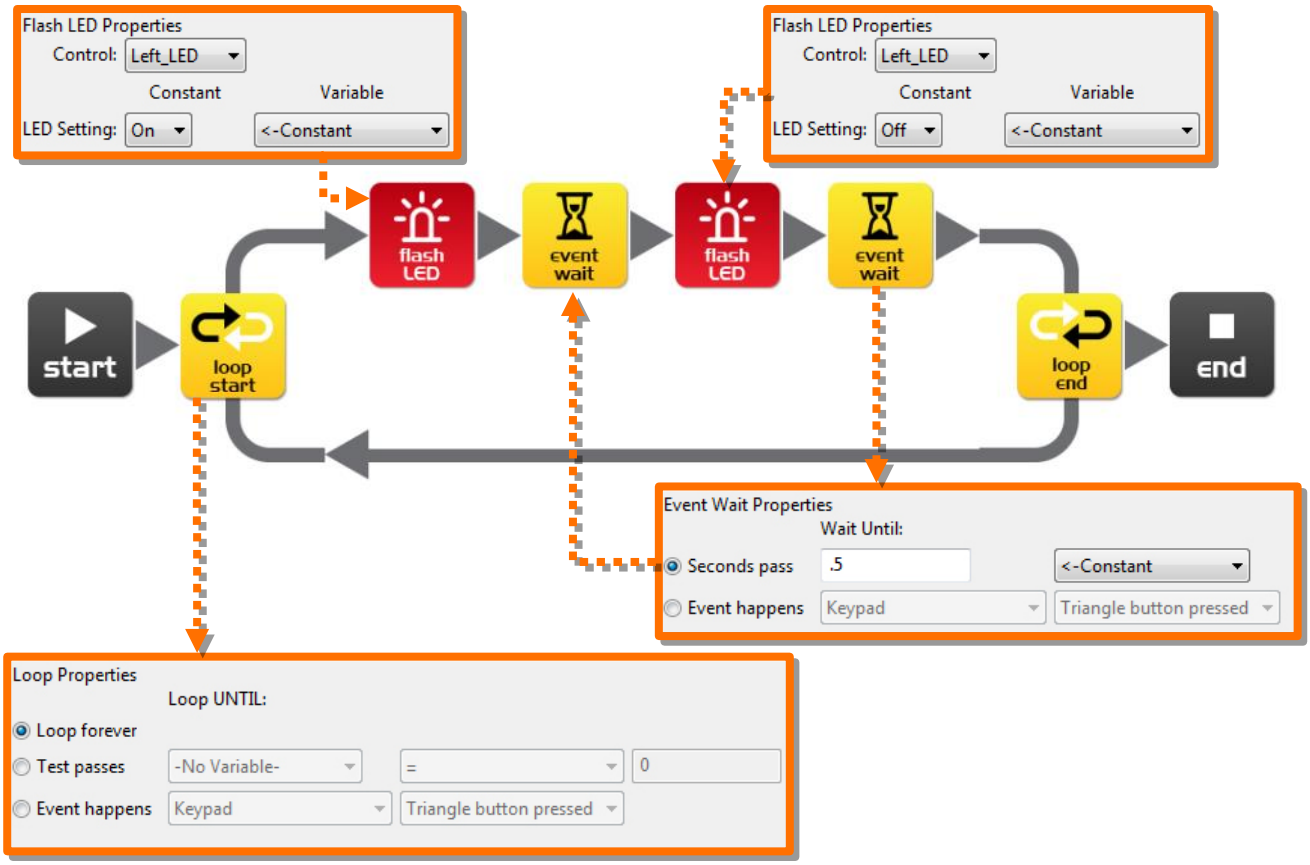
Pobierz pełną instrukcję płyty głównej z https://meetedison.com/content/Edison-Motherboard-layout-V2.pdf

EdPrzygoda 1- Zapal diodę LED

Prosty program do zapalenia lewej diody LED

Pierwszy program do napisania będzie zapalał lewą diode LED. Jest on bardzo prosty I jak sama nazwa sugeruje, włącza i wyłącza światło lewej diody LED.

Przeciągnij ikony w sposób, jaki pokazano poniżej, następnie kliknij w każdą ikonę i ustaw właściwości w taki sam sposób jak na obrazku.



Jak działa program?

Minikontroler (mózg) Edisona rozumie każdą ikonę jako krok, a strzałki pokazują mu ich kierunek. Załóżmy, że wykona czynności pokazane w programie powyżej.

Krok 1: Ikona startu rozpoczyna program.

Krok 2: Program zaczyna się od górnej strzałki ikony zapętlającej.

Krok 3: Dioda LED jest włączana przez ikonę diody LED.

Krok 4: Program czeka 0.5 sekundy (ikona klepsydry Event wait) - dioda LED jest włączona

Krok 5: Dioda LED jest wyłączana.

Krok 6: program czeka 0.5 sekundy (ikona klepsydry) - dioda LED jest wyłączona

Krok 7: To, co tutaj się dzieje jest bardzo ważne! Zamiast na strzałkę po prawej stronie ikony końcowej pętli, program przenosi się na strzałkę poniżej ikony i wraca do ikony startu pętli. Dzieje się tak dlatego, że właściowść ikony pętli jest ustawiona na “Powtarzaj zawsze”. Program zatem idzie do pierwszej ikony diody LED, a potem następuje ta sama sekwencja, którą opisano powyżej. Będzie to trwalo wiecznie, a przynajmniej dopóki baterie się nie wyczerpią.

Pobieranie programu I jego uruchomienie

Podłącz kabel EdComm do Edisona I gniazda słuchawkowego w twoim komputerze lub tablecie. Kliknij przycisk Program Edison, a następnie przycisk nagrywania (okrągły). Teraz klikij przycisk Start Download.

Naciśnij przycisk Play (trójkątny). Lewa dioda LED będzie migać.

Gratulacje! Napisałeś I pobrałeś swój pierwszy program do Edisona.

Eksperyment

Spróbuj zmienić czas oczekiwania i dodaj więcej ikon lampy LED do kontroli światłem prawym. Czy potrafisz zrobić fajne pulsujące światła?

EdFakty

L.E.D. (Light Emitting Diode) oznacza diodę emitującą światło.

W przeciwieństwie do normalnej żarówki wynalezionej przez Thomasa Edisona (bez związku bezpośredniego z robotem Edison) dioda nie posiada żarnika lub specjalnego drutu, który wytwarza światło, gdy prąd w nim płynie. Zamiast korzystać z zaawansowanych diod LED z materiałem półprzewodnikowym, takich jakie można znaleźć wewnątrz chipów komputerowych.

Diody LED są lepsze od tradycyjnych żarówek, gdyż świecą dłużej, są bardziej wytrzymałe I zużywają znacznie mniej energii.

Od dzisiaj bądź świadom, że żarówki to nie jedyne źródła światła. Teraz trwa era diod LED!

Lekcja 2: Ruchy robota – jazda

Wprowadzenie do programowania sekwencyjnego – Uczniowie dowiadują się w jaki sposób robot Edison reaguje na wydawane w określonej kolejności polecenia Pythona oraz łączą ze sobą pojęcia czasu, prędkości i odległości.

Podczas tej lekcji uczniowie wykorzystują funkcje jazdy, wbudowane w aplikacji EdPy, aby otrzymać ruch robota w postaci jazdy do przodu i do tyłu na określone odległości.

W lekcji 2 uczniowie przerabiają cztery główne zadania programistyczne, korzystając z pięciu kart pracy i jednej karty zadań:

- Karta pracy 2.1 – Jazda robota do przodu
- Karta pracy 2.2 – Jazda robota do tyłu
- Karta pracy 2.3 – Do przodu, a następnie do tyłu
- Karta pracy 2.4 – Wyrażenia w Pythonie
- Karta pracy 2.5 – Jazda uruchamiana z klawiatury pomocniczej

- Karta zadań 2.1 (Linia startu/mety)

Lekcja 2, program 1 – Jazda robota do przodu (Karta pracy 2.1)

Uczniowie piszą program, który kieruje jazdą robota do przodu na określoną odległość, od punktu „start" do punktu „stop".

Karta pracy 2.1 prowadzi uczniów przez poszczególne etapy pisania programu, w których mają korzystać z funkcji, parametrów wejściowych oraz niektórych elementów pomocy w EdPy.

Wskazówki i porady:

- Poinstruuj uczniów, aby skorzystali z linii startu i mety znajdujących się w karcie zadań 2.1 lub przygotowali takie linie, przy pomocy kolorowej taśmy, na ławce lub na podłodze.
- Programy napisane przez uczniów muszą powodować zakończenie jazdy Edisona przed przekroczeniem linii mety.

Lekcja 2, program 2 – Jazda robota do tyłu (Karta pracy 2.2)

Uczniowie dowiadują się o możliwości napisania programu na wiele różnych sposobów, które dadzą ten sam efekt. Uczniowie analizują dwa różne sposoby napisania programu, który spowoduje jazdę robota do tyłu na określoną odległość, od punktu „start" do punktu „stop".

Karta pracy 2.2 przeprowadza uczniów przez kolejne etapy pisania programu, utrwalając korzystanie z funkcji, parametrów wejściowych oraz niektórych funkcji pomocy EdPy. Następnie uczniowie eksperymentują, używając stałej jako parametru wejściowego.

Wskazówki i porady:

- Poinstruuj uczniów, aby skorzystali z linii startu i mety znajdujących się w karcie zadań 2.1 lub przygotowali takie linie, przy pomocy kolorowej taśmy, na ławce lub na podłodze.
- Programy napisane przez uczniów muszą powodować zakończenie jazdy Edisona przed przekroczeniem linii mety.

Lekcja 2, program 3 – Do przodu, a następnie do tyłu (Karta pracy 2.3)

Uczniowie koncentrują się na programowaniu sekwencyjnym, tworząc kilkuetapowy program wykorzystujący kilka wierszy kodu. Uczniowie mają napisać program, który spowoduje jazdę robota do przodu, a następnie do tyłu na określoną odległość, od oznaczenia startu do oznaczenia mety i z powrotem do startu.

Karta pracy 2.3 przeprowadza uczniów przez pisanie programu, utrwalając podstawowe kroki programowania pozwalające używać Edisona i EdPy. Uczniowie ćwiczą korzystanie za stałej jako parametru wejściowego.

Wskazówki i porady:

- Poinstruuj uczniów, aby skorzystali z linii startu i mety znajdujących się w karcie zadań 2.1 lub przygotowali takie linie, przy pomocy kolorowej taśmy, na ławce lub na podłodze.
- Programy napisane przez uczniów muszą spowodować zakończenie jazdy Edisona przed przekroczeniem linii mety.

Lekcja 2, ćwiczenie w trybie offline – Wyrażenia w Pythonie (Karta pracy 2.4)

Uczniowie poznają:

- podstawową koncepcję wyrażeń,
- sposób zapisywania wyrażeń w Pythonie,
- jak działają wyrażenia oraz
- cel stosowania wyrażeń w programowaniu.

Karta pracy 2.4 wyjaśnia jak działają wyrażenia, w jaki sposób są zapisywane i jakie wartości przyjmują w Pythonie. Uczniowie ćwiczą odczytywanie znaczenia wyrażeń i określają czy przyjmują one wartość „prawda” czy „fałsz”.

Lekcja 2, program 4 – Jazda uruchamiana z klawiatury pomocniczej (Karta pracy 2.5)

Uczniowie zapoznają się:

- z pętlą „while” w Pythonie,
- ze znaczeniem wcięć w Pythonie oraz
- z funkcją Ed.ReadKeyPad().

Karta pracy 2.5 pokazuje w jaki sposób wyrażenia stosowane są w obrębie pętli „while” w EdPy. Uczniowie programują Edisona, aby jechał do przodu, gdy naciśnięty jest trójkątny lub okrągły przycisk, stosując pętlę „while” i wyrażenia. Następnie uczniowie piszą własną wersję programu, dostosowując ją w taki sposób, aby Edison jechał do przodu, gdy zostanie naciśnięty trójkątny lub okrągły przycisk, a następnie jechał do tyłu, gdy zostanie ponownie naciśnięty trójkątny lub okrągły przycisk.

Wskazówki i porady:

- Python jest bardzo drobiazgowy, jeśli chodzi o białe miejsca (np. puste znaki) w obrębie wierszy kodu. W składni Pythona użycie klawisza tab do zaznaczenia wcięcia wiersza kodu oraz użycie klawisza spacji, aby uzyskać taką samą wielkość wcięcia nie są równoważne. Uczniowie powinni używać klawisza tab.

Imię i nazwisko_____

Lekcja 3: Karta pracy 3.1 – Skręt w prawo

Podczas tego ćwiczenia masz za zadanie napisać program, który sprawi, że Edison skręci w prawo.

Popatrz na poniższy program:

```
1
2  #-------------Setup----------------
3
4  import Ed
5
6  Ed.EdisonVersion = Ed.V2
7
8  Ed.DistanceUnits = Ed.CM
9  Ed.Tempo = Ed.TEMPO_MEDIUM
10
11 #--------Your code below-----------
12 degreesToTurn = 90
13 Ed.Drive(Ed.SPIN_RIGHT,Ed.SPEED_6,degreesToTurn)
14
```

Zwróć uwagę na wiersz 13 programu. Pamiętaj, że funkcja Ed.Drive ma trzy parametry wejściowe: kierunek, prędkość i odległość. W tym programie parametr odległości nie jest liczbą, ale „degreesToTurn” (o ile stopni skręcić).

Jest to zmienna.

W Pythonie zmienne są miejscami w pamięci, zarezerwowanymi do przechowywania wartości. Oznacza to, że tworząc zmienną, rezerwujesz miejsce w pamięci programu.

Zmienna jest wartością ustawioną w pewnym miejscu programu.

Popatrz na wiersz 12 programu. W tym miejscu ustawiana jest wartość „degreesToTurn”. Nazywa się to przypisaniem wartości do zmiennej. W Pythonie znak równości (=) używany jest do przypisywania wartości zmiennym.

Zmienne możesz stosować do zachowywania wartości, które są używane w kilku miejscach w całym programie. Może to być bardzo przydatne, zwłaszcza jeśli wartość zmiennej ulega zmianie. Stosując zmienne, musisz wprowadzić zmianę tylko w jednym wierszu kodu, a dana wartość zostanie skorygowana we wszystkich miejscach, w których jest używana przez twój program.

Twoja kolej:

Napisz program, a następnie wgraj go do Edisona i uruchom. Skorzystaj z karty zadań 3.1 lub użyj kolorowej taśmy do oznaczenia „początku” i „końca” kąta na ławce lub na podłodze, aby przygotować obszar testowy dla swojego programu.

Imię i nazwisko_____

1. Opisz co robi robot i wyjaśnij dlaczego tak się zachowuje po uruchomieniu programu.

_____

_____

_____

Dodaj nowy wiersz na końcu swojego programu (po wierszu 13) i dołącz do swojego programu następujący kod:

```
Ed.Drive(Ed.SPIN_LEFT,Ed.SPEED_6,degreesToTurn)
```

Wgraj program do Edisona i uruchom w obszarze testowym.

2. Opisz co robi robot i dlaczego tak się zachowuje po uruchomieniu zaktualizowanego programu.

_____

_____

_____

Zmodyfikuj swój program w taki sposób, aby Edison najpierw skręcił o 180 stopni w prawo, a następnie o 180 stopni w lewo.

Podpowiedź: Pamiętaj o możliwości zmiany wartości zmiennej degreesToTurn.

Wgraj swój zaktualizowany program do Edisona i uruchom w obszarze testowym, aby sprawdzić czy zmiana została pomyślnie przeprowadzona.

3. Który wiersz lub które wiersze w programie zmieniłeś? Zapisz zaktualizowany wiersz lub wiersze.

_____

_____

_____

Nazwy zmiennych w Pythonie muszą stosować się do pewnych reguł. Na przykład, nie mogą zawierać żadnych znaków specjalnych, takich jak # czy $. Spróbuj zmienić nazwę zmiennej „degreesToTurn”. Poeksperymentuj z różnymi nazwami i użyj przycisku „Check Code”, aby zobaczyć, które z nazw są dozwolone, a które nie.

Imię i nazwisko_____

4. Podaj dwa przykłady niedozwolonych nazw zmiennych, które utworzyłeś.

_____

_____

CC BY-SA ABIX | robotedison.pl | cyfrowaszkola.waw.pl

Dokumentacja aplikacji EdPy

Niniejszy dokument to pełna kopia tekstu pomocy dostępnego w sekcji "Dokumentacja" aplikacji EdPy.

Spis treści

Ed.List() 4

Ed.LeftLed() 5

Ed.RightLed() 6

Ed.ObstacleDetectionBeam() 7

Ed.LineTrackerLed() 8

Ed.SendIRData() 9

Ed.StartCountDown() 10

Ed.TimeWait() 11

Ed.RegisterEventHandler() 12

Ed.PlayBeep() 14

Ed.PlayMyBeep() 15

Ed.PlayTone() 16

Ed.PlayTune() 18

Ed.TuneString() 20

Ed.Drive() 23

Ed.DriveLeftMotor() 26

Ed.DriveRightMotor() 29

Ed.ReadObstacleDetection() 32

Ed.ReadKeypad() 34

Ed.ReadClapSensor() 35

Ed.ReadLineState() 37

Ed.ReadRemote() 38

Ed.ReadIRData() 40

Ed.ReadLeftLightLevel() 41

Ed.ReadRightLightLevel() 42

Ed.ReadLineTracker() 43

Ed.ReadCountDown() 44

CC-BY-SA – ABIX Edukacja | robotedison.pl | cyfrowaszkola.waw.pl

Ed.ReadMusicEnd() 45
Ed.ReadDriveLoad() 46
Ed.ReadDistance() 47
Ed.SetDistance() 48
Ed.ResetDistance() 49
Ed.ON 50
Ed.OFF 50
Ed.NOTE_# 50
Ed.TEMPO_# 52
Ed.FORWARD 53
Ed.BACKWARD 53
Ed.FORWARD_RIGHT 54
Ed.BACKWARD_RIGHT 54
Ed.FORWARD_LEFT 54
Ed.BACKWARD_LEFT 55
Ed.SPIN_# 55
Ed.STOP 55
Ed.SPEED_# 56
Ed.DISTANCE_UNLIMITED 57
Ed.MOTOR_# 57
Ed.TIME_# 58
Ed.OBSTACLE_# 58
Ed.LINE_ON_# 59
Ed.KEYPAD_# 59
Ed.CLAP_# 60
Ed.DRIVE_# 60
Ed.MUSIC_# 61
Ed.REMOTE_CODE_# 61
Ed.EVENT_# 62
Ed.CM 64
Ed.INCH 64
Ed.TIME 64

CC-BY-SA – ABIX Edukacja | robotedison.pl | cyfrowaszkola.waw.pl

Ed.V1 65
Ed.V2 65
Ed.Tempo 65
Ed.DistanceUnits 66
Ed.EdisonVersion 68
abs() 69
len() 70
range() 71
True 72
False 72
import 73
def 73
pass 75
for 76
while 77
if/elif/else 78
class 79
return 82

Ed.List()

Ed.List(size)

Parametry:

Size:

Liczba całkowita dodatnia – określa ilość liczb całkowitych w nowej liście.

Wartość maksymalna to 250.

Zwraca:

nie dotyczy

Ed.List(size, initialList)

Parametry:

Size:

Liczba całkowita dodatnia – określa ilość liczb całkowitych w nowej liście.

Wartość maksymalna to 250.

Initial List:

Pythonowa lista np. [1,2,3] – ustawia początkowe wartości liczb całkowitych w nowym obiekcie Ed.List.

Zwraca:

nie dotyczy

Wyjaśnienie:

Tworzy listę zmiennych Edisona.

Przykłady:

Utworzenie pustej listy i wypełnienie jej zerami.

```
#--------Poniżej Twój kod-----------
zeros=Ed.List(5)
for x in range(5):
    zeros[x]=0
```

Utworzenie nowej listy z przykładowymi wartościami.

```
#--------Poniżej Twój kod-----------
example=Ed.List(5,[1,2,3,4,5])
```

Uważaj na:

Maksymalny rozmiar listy to 250.

Dodatkowe nowe elementy nie mogą być dodawane na końcu listy. Rozmiar listy jest stały. Pythonowe listy indeksowane są od 0, czyli pierwszy element listy ma indeks 0. Na przykład przy użyciu listy zdefiniowanej w powyższym przykładzie poniższy kod błyśnie lewą diodą Edisona 5 razy.

```
while example[0]!=0:
```

```
    Ed.LeftLed(Ed.ON)
    Ed.TimeWait(500, Ed.TIME_MILLISECONDS)
    Ed.LeftLed(Ed.OFF)
    Ed.TimeWait(500, Ed.TIME_MILLISECONDS)
    example[0]=example[0]-1
```

Ed.LeftLed()

Ed.LeftLed(state)

Parametry:

State:

- Ed.ON – dioda LED włączona
- Ed.OFF – dioda LED wyłączona

Zwraca:

nie dotyczy

Wyjaśnienie:

Włącza lub wyłącza lewą diodę LED Edisona.

Przykłady:

Szybkie błyśnięcie lewą diodą LED.

```
#--------Poniżej Twój kod-----------
Ed.LeftLed(Ed.ON)
Ed.TimeWait(500, Ed.TIME_MILLISECONDS)
Ed.LeftLed(Ed.OFF)
Ed.TimeWait(500, Ed.TIME_MILLISECONDS)
```

Lewa dioda LED włączona podczas jazdy.

CC-BY-SA – ABIX Edukacja | robotedison.pl | cyfrowaszkola.waw.pl