

BIBLIOTEKA KODÓW

100 Sekwencji kodowania

do Nauki i Zabawy



Spis treści

Sterowanie ruchem

1. Kierunki Modułu Spin
2. Kierunki Modułu Joint
3. Jazda po Kwadracie
4. Jazda po Trójkącie Pitagorejskim
5. Jazda po Kole
6. Jazda po Prostokącie
7. Odzworowanie Ruchów przez Serwomotory Modułu Joint
8. Kombinacja Sterowania Klawiszami Modułu Spin i Zdalnego Sterowania między Modułami Joint
9. Kontrola Prędkości Modułu Spin
10. Kontrola Prędkości Modułu Joint
11. Jazda Zygzakiem pod kątem 90°
12. Robot Czteronożny
13. Ruchomy Moduł Joint wyposażony w kółka
14. Jazda Modułem Spin według Ustalonych Kierunków
15. Ruch Modułu Joint według Ustalonych Kierunków
16. Ruch Trygonometryczny
17. Sterowanie Dotykaniem Modułu Spin

Pomiar

18. Pomiar Momentu Obrotowego Modułu Joint
19. Pomiar Prędkości Serwomotorów Modułu Joint
20. Pomiar Przyspieszenia Modułu Spin
21. Porównywanie Dwoch Wartości
22. Pomiar Obwodu Koła Fable
23. Pomiar Odległości na podstawie Obwodu Koła Fable

Wykrywanie

24. Wykrywanie Przeszkody
25. Unikanie Przeszkody - Metoda 1
26. Unikanie Przeszkody - Metoda 2
27. Wykrywanie Kolorów
28. Pomiar Światła w Otoczeniu
29. Pomiar Odległości w Poblżu
30. Oczekiwanie na Naciśnięcie Klawisza
31. Wykrywanie Kolorowej Linii
32. Wysyłanie/Odbieranie Komunikatów z Modułem Spin
33. Reakcja Kamery na Ruch
34. Radarowy Model Działania dla Modułu Spin
35. Maszyna Sortująca Kolory - Metoda 1
36. Maszyna Sortująca Kolory - Metoda 2
37. Wykrywanie Podczerwieni
38. Monitorowanie Poziomu Baterii
39. Moduł Joint Sterowany przez Wykrywanie Kolorów Kamery
40. Wykrywanie Twarzy za pomocą Kamery
41. Reakcja na Zmiany Natężenia Światła
42. Wykrywanie Kątów Pozycji Modułu Wspólnego
43. Zastosowanie Filtra na Zrzucie Ekranu z Kamery
44. Wyodrębnianie Ilości RGB
45. Sterowanie Modułem Spin za pomocą Klawiszy
46. Sterowanie Modułem Joint za pomocą Klawiszy
47. Fable Hello
48. Podążanie za Liderem
49. W Stronę Światła
50. Zdalne Sterowanie Modułem Spin

Kontrola i Przetwarzanie Danych

51. Moduł Joint jako Śrubokręt
52. Struktura Syjamska dla Modułów Spin
53. Zdalne Sterowanie Bariery
54. Moduł Joint jako Joystick dla Modułu Spin
55. Symulator Dźwigu
56. Całkowite Zatrzymanie Programu (STOP)
57. Użycie Pliku MP3
58. Zabawa z Nutami
59. Zakończenie Pętli Nieskończonej
60. Nagrywanie w Pliku .csv
61. Odczyt z Pliku .csv
62. Nagrywanie Wielu Danych w Pliku .csv
63. Tworzenie Zmiennej
64. Generowanie Losowych Wartości dla Serwomotora Modułu Joint
65. Obliczanie Średniej Arytmetycznej Ocen
66. Obliczanie Średniej Geometrycznej dla Dwoch Liczb
67. Konwersja Tekstu na Mowę
68. Odczyt Liczb z Ekranu Telefonu
69. Odczyt Liczb z Ekranu Telefonu / Formuła $n \cdot \text{choose } k$
70. Odczyt Liczb z Ekranu Telefonu / Formuła $n!$
71. Obliczanie Reszty z Dzielenia (Metoda 1)
72. Obliczanie Reszty z Dzielenia (Metoda 2)
73. Obliczanie Wystąpień
74. Sortowanie Liczb w Porządku Rosnącym
75. Wprowadzenie Danych w Określonej Kolejności (Lista)
76. Sortowanie Liczb Parzystych Rosnąco
77. Wyodrębnianie Liczb Parzystych/Nieparzystych z Listy i Funkcji
78. Użycie Polecenia „licz z”
79. Boolowski Operator Logiczny - I
80. Boolowski Operator Logiczny - LUB
81. Boolowski Operator Logiczny - NIE
82. Podwójna Negacja

Wyświetlanie

83. Światła Przerzywane z użyciem Modułu Spin
84. Światła Przerzywane z użyciem Huba
85. Wyświetlanie w Konsoli
86. Wyświetlanie Wykresu Szeregowego Czasu
87. Wyświetlanie Wykresu Punktowego
88. Wyświetlanie Wykresu Liniowego
89. Wyświetlanie Pozycji Dotknięcia Ekranu
90. Tworzenie Wykresu Obrotów Modułu Spin
91. Wyświetlanie Wykresu Kąta Modułu Joint
92. Porównywanie Dwoch Zaprogramowanych Wartości
93. Ustawianie Wyrazów Twarzy Fable
94. Mieszanie Światła R, G, B
95. Ustawianie Kolorów Tęczówek/Powiek
96. Ustawianie Kierunku Oczu
97. Wyświetlanie Scalonych Danych w Jednej Linii

Gry

98. Gra w Kości
99. Gra w Bramkarza
100. Zgadnij Liczbę

Pamiętaj o...

Przed programowaniem

1. Roboty są naładowane i sprawne, co można sprawdzić bezpośrednio w aplikacji.
2. Hub jest podłączony do urządzenia, z którego będziesz programować. Po podłączeniu zaświeci się jednym z sześciu kolorów, a kod będzie widoczny w aplikacji.
3. Aplikacja Fable Blockly jest zainstalowana i zaktualizowana na urządzeniu do programowania. Możesz pobrać najnowszą wersję [tutaj](#).
4. Roboty mają zaktualizowane oprogramowanie wbudowane. W razie potrzeby możesz zaktualizować oprogramowanie wbudowane z poziomu aplikacji. Przejdź do następujących linków dla: [Hub](#), [Moduł Spin](#), [Moduł Joint](#).
5. Aplikacja Fable Face jest zainstalowana i działa na Twoim telefonie. Możesz ją pobrać [tutaj](#).
6. Upewnij się, że Twoje połączenie internetowe działa.
7. Sprawdź, czy Bluetooth jest aktywny i działa na urządzeniu, z którego będziesz programować.

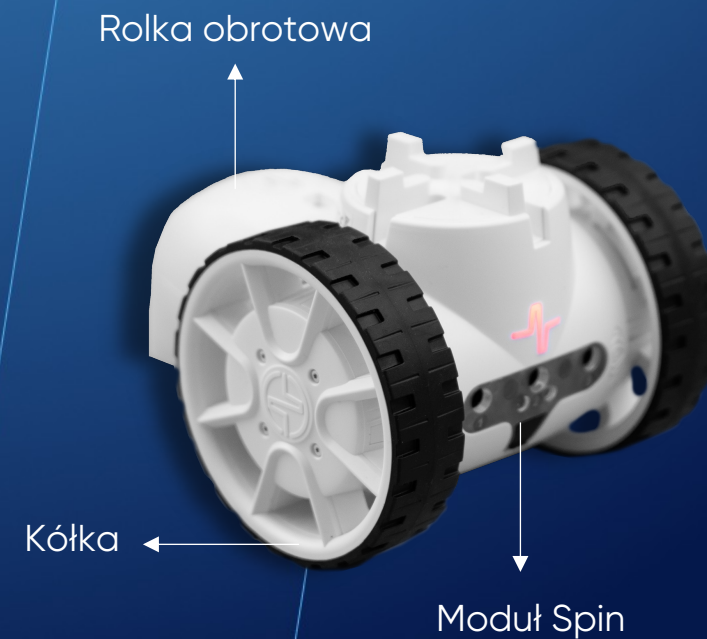
Podczas korzystania z programów

1. Roboty i Hub powinny wyświetlać ten sam kolor.
2. Użyj odpowiednich kodów dla robotów w aplikacji.
3. Zwracaj uwagę na komunikaty o przeciążeniu momentu obrotowego! Komunikaty te wskazują, kiedy silnik jest przeciążony i wymaga regulacji w programie, podzespole, w którym się znajduje lub środowisku w którym działa.
4. Kąt Modułu Wspólnego wynoszący zero stopni oznacza pionową pozycję.

Kod 1: Kierunki Modułu Spin


- Ten program umożliwi robotowi poruszanie się do przodu, do tyłu, oraz skręcanie w prawo lub w lewo.

OBEJRZYJ WIDEO



repeat forever

do

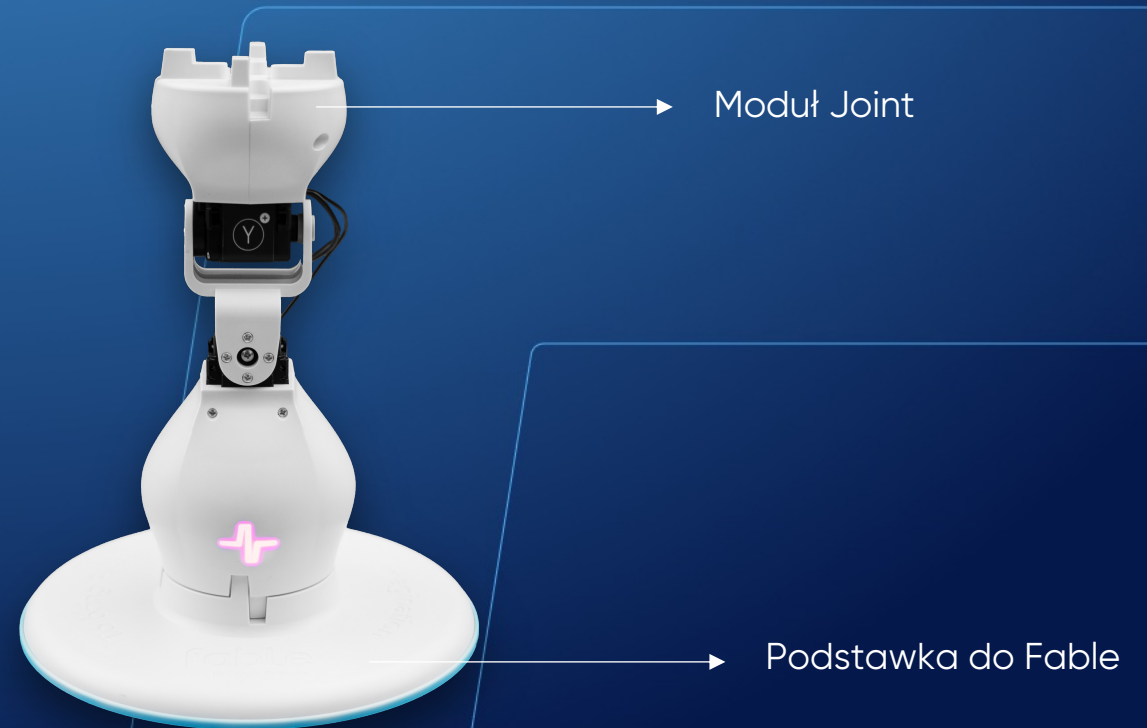
 move forward ▼ on 143 ▼

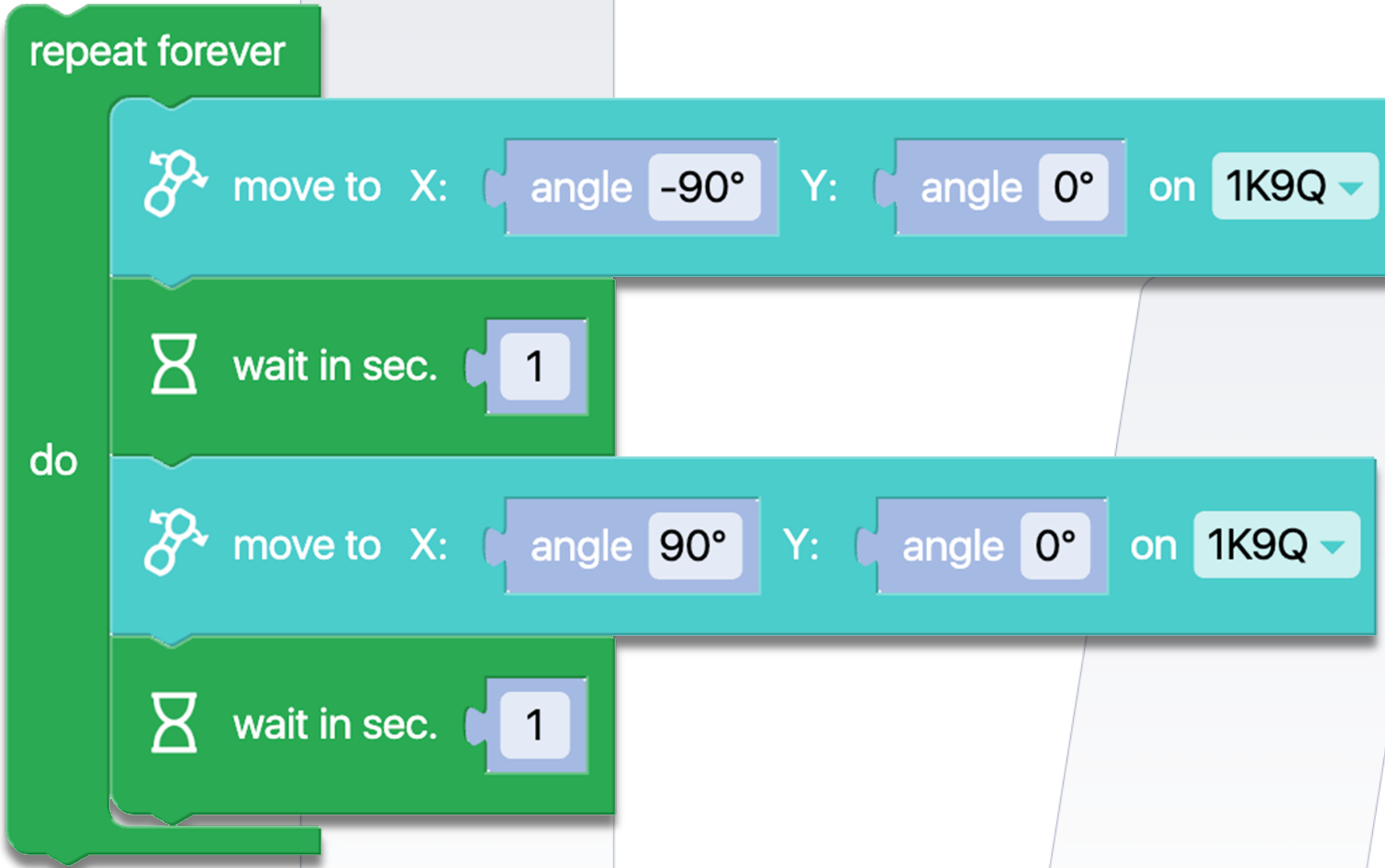
- ✓ move forward
- move backward
- stop moving
- left
- right

Kod 2: Kierunki Modułu Joint

- Moduł Joint wykonuje powtarzalny ruch w zakresie od -90 stopni do 90 stopni przy użyciu serwomotoru X, podczas gdy serwomotor Y pozostaje nieruchomy. Dzięki zamontowaniu wyciętej i spersonalizowanej kartonowej dłoni, robot może machać na pożegnanie.

OBEJRZYJ WIDEO

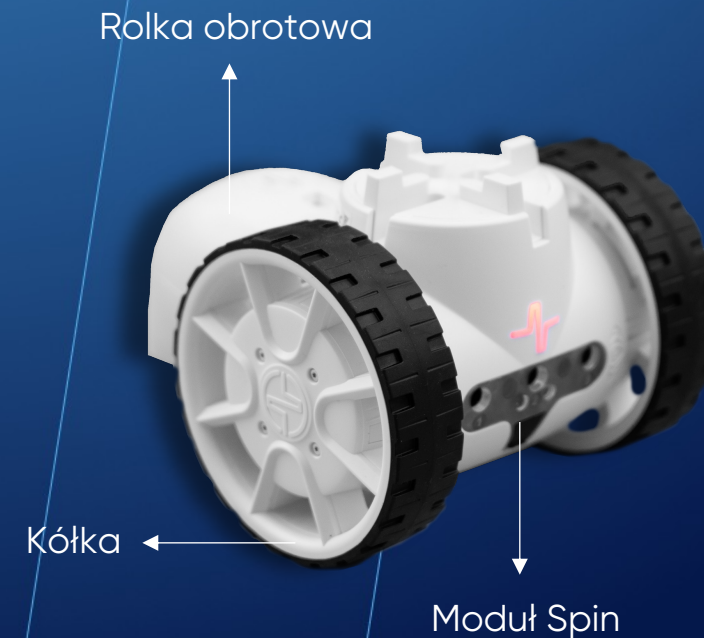


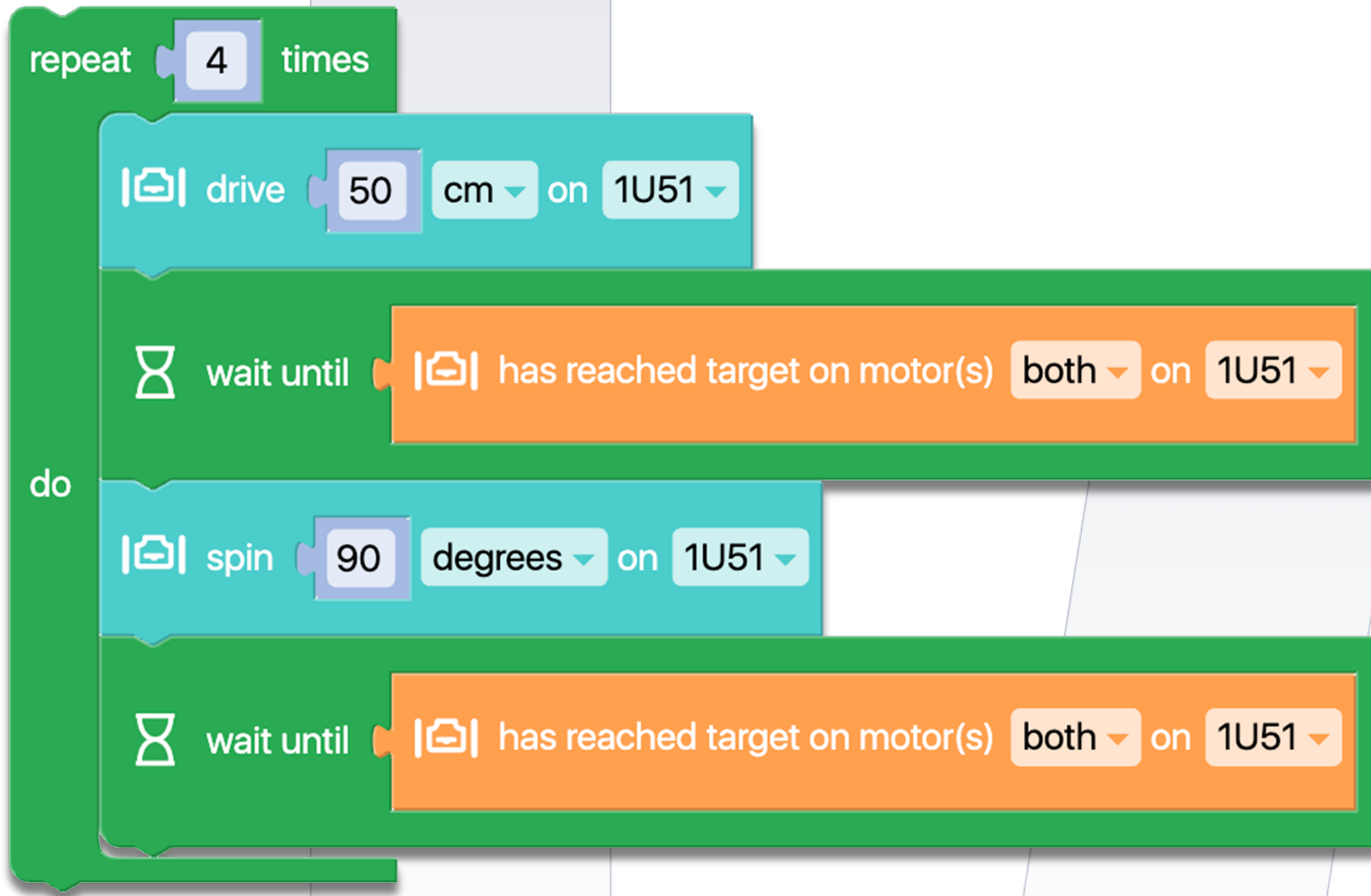


Kod 3: Jazda po kwadracie

- Moduł Spin porusza się po obwodzie kwadratu, gdzie każdy bok ma długość 50 centymetrów. Pomarańczowy blok zapobiega rozpoczęciu nowego polecenia przed ukończeniem poprzedniego.

OBEJRZYJ WIDEO



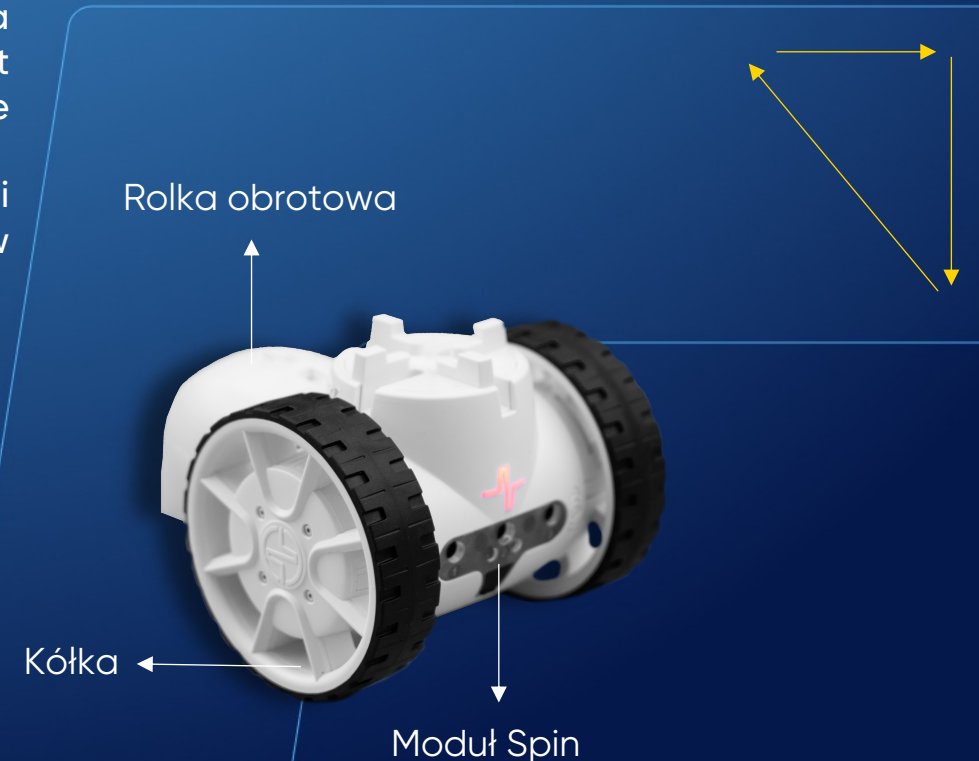


Kod 4:

Jazda po trójkącie pitagorejskim

- Trójkąt o bokach mierzących odpowiednio 30, 40 i 50 cm spełnia założenia trójkąta pitagorejskiego. Trójkąt ten jest trójkątem prostokątnym i jest zgodny ze wzorem: kwadrat przeciwprostokątnej równa się sumie kwadratów boków.
- W tym trójkącie kąty, które nie są kątami prostymi, mają wartości: 36.87° i 53.13° . Ponieważ obrót odbywa się po zewnętrznej stronie, wartości kątów obrotu wynoszą 180° minus wartości kątów.

OBEJRZYJ WIDEO

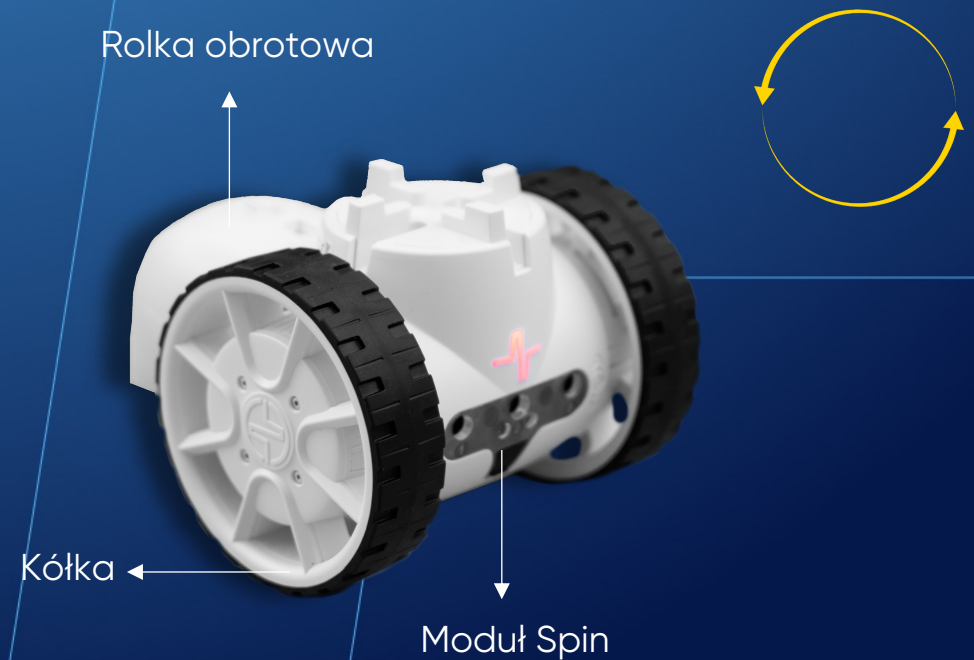




Kod 5: Jazda po kole

- Sekwencja kodu wydaje Modułowi Spin polecenie poruszania dwoma silnikami z różnymi prędkościami, co skutkuje ruchem okrężnym do przodu.

OBEJRZYJ WIDEO

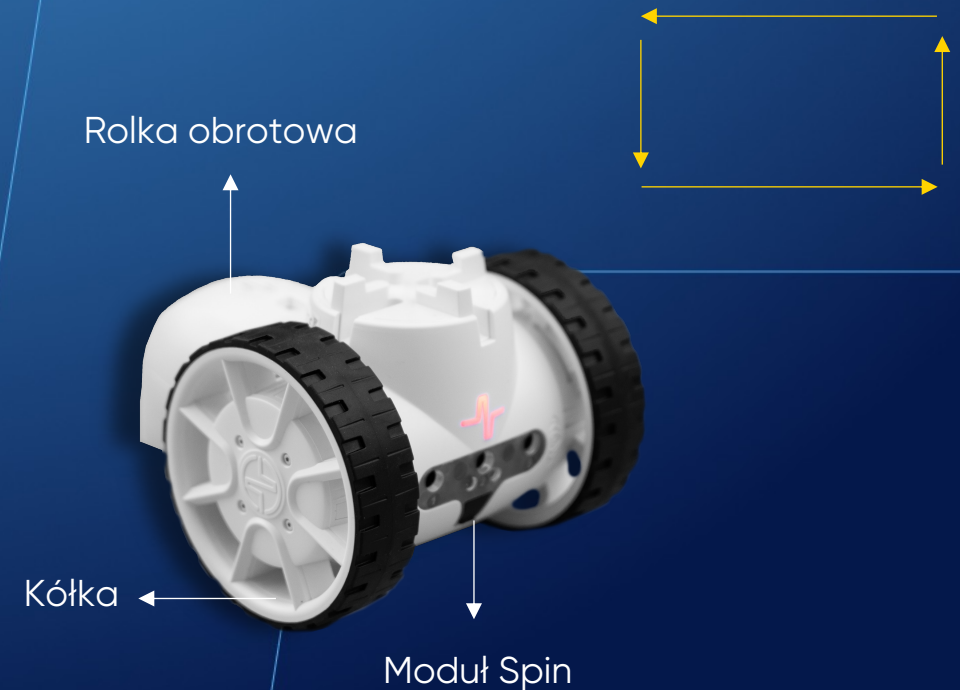


```
repeat forever
do |🗑️| set speed A: -40 B: 70 on 143 ▼
```

Kod 6: Jazda po prostokącie

- Moduł Spin porusza się wzdłuż obwodu prostokąta o wymiarach 50 cm na 20 cm. Pomarańczowy blok zapobiega rozpoczęciu nowego polecenia przed ukończeniem poprzedniego.

OBEJRZYJ WIDEO

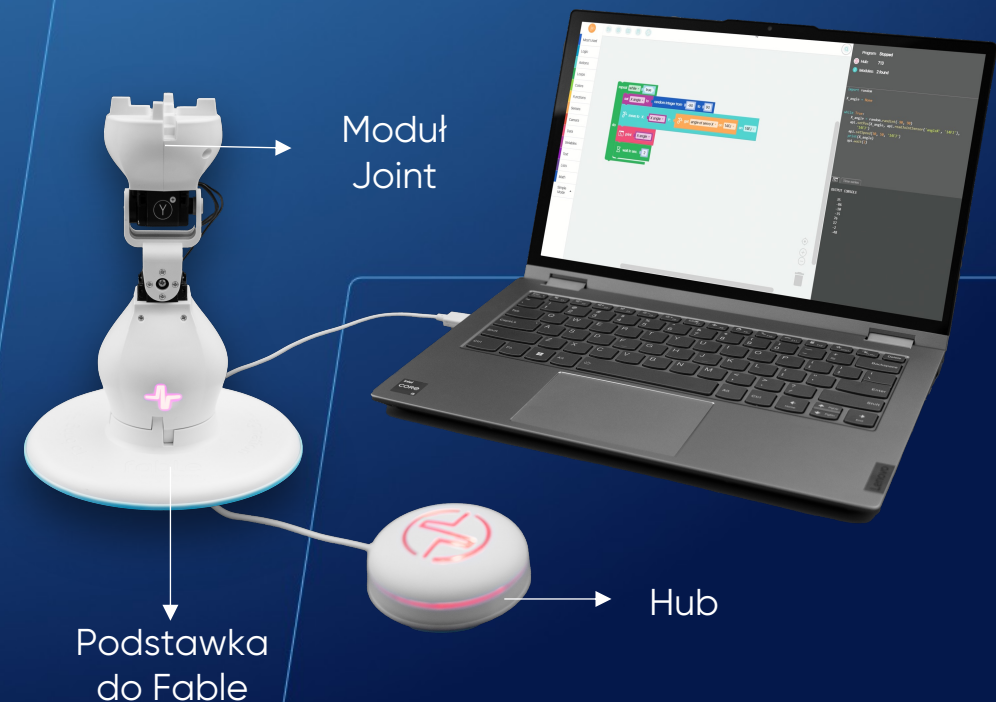


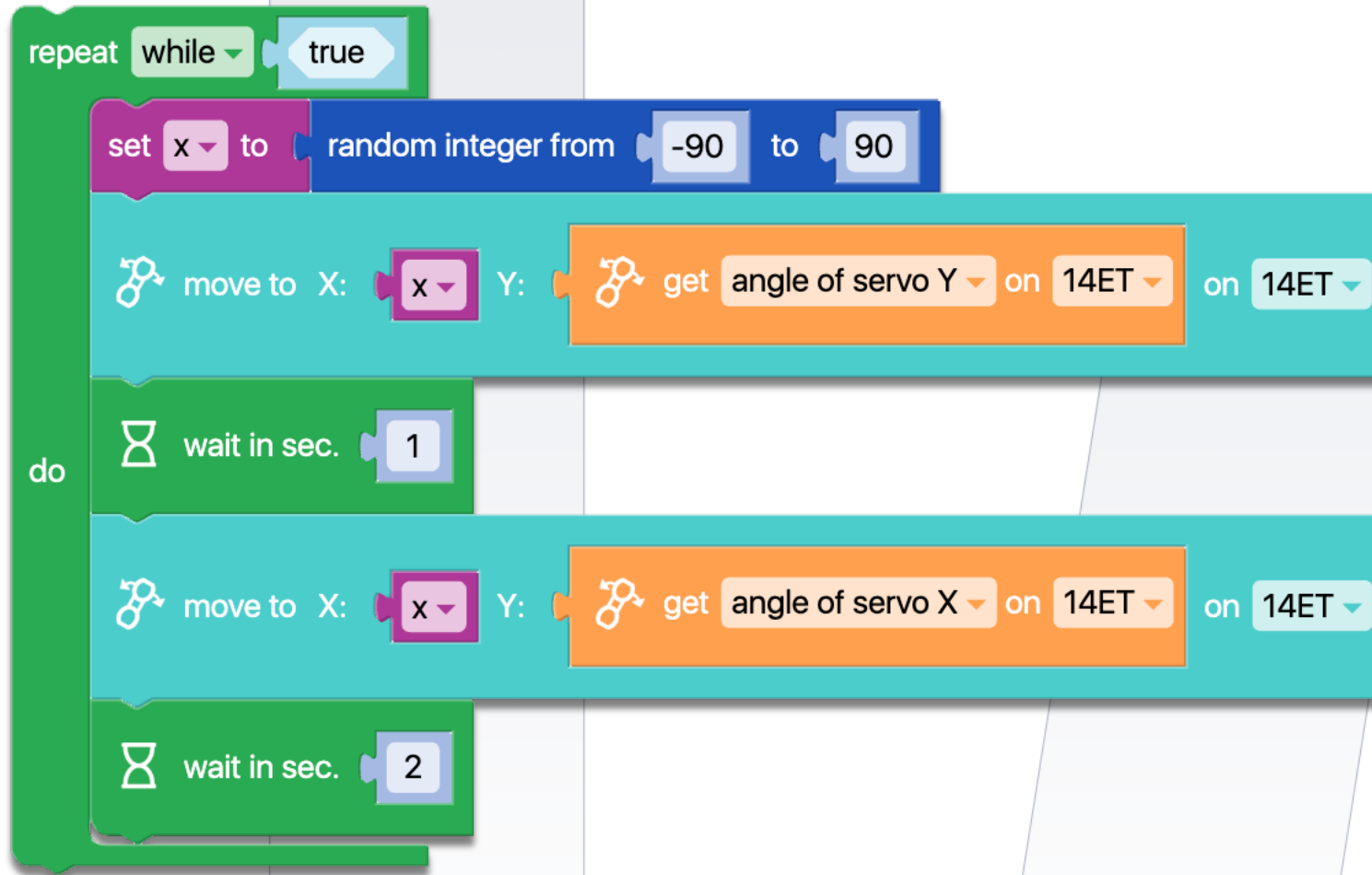
```
repeat 2 times
  drive 50 cm on A21B
  wait until | has reached target on motor(s) both on A21B
  spin 90 degrees on A21B
  wait until | has reached target on motor(s) both on A21B
do
  drive 20 cm on A21B
  wait until | has reached target on motor(s) both on A21B
  spin 90 degrees on A21B
  wait until | has reached target on motor(s) both on A21B
```

Kod 7: Odwzorowanie ruchów za pomocą serwomotorów Modułu Joint

- Program co dwie sekundy generuje zmienną i przypisuje jej losową wartość z zakresu od -90 do 90 stopni. Następnie zmienna ta jest używana do sterowania Modułem Joint. W rezultacie silnik X zostanie przesunięty do pozycji określonej przez zmienną, a silnik Y odzwierciedli ten sam ruch co silnik X.

OBEJRZYJ WIDEO

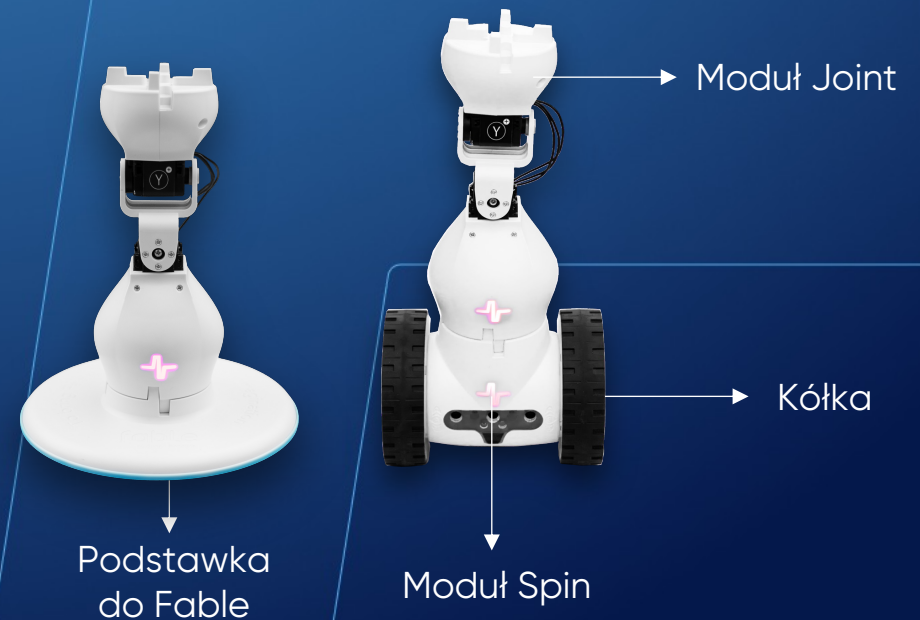




Kod 8: Kombinacja sterowania klawiszami Modułu Spin i zdalnego sterowania między Modułami Joint

- Program steruje ruchem Modułu Spin za pomocą klawiszy klawiatury oraz dołączonego Modułu Joint, który jest obsługiwany przez inny Moduł Joint. Ten podzespół może być wykorzystywany do celów zdalnego sterowania.

OBEJRZYJ WIDEO



```
set Leader to module 1D29
set Follower to module 1D2A

repeat while true
  move to X: get angle of servo X on # Leader Y: get angle of servo Y on # Leader on # Follower

  if key pressed? up
  do move forward on 1111
  else if key pressed? down
  do move backward on 1111
  else if key pressed? left
  do left on 1111
  else if key pressed? right
  do right on 1111
  else
  do stop moving on 1111
```

Kod 9: Kontrola prędkości Modułu Spin

- Program kontroluje prędkość Modułu Spin. Zmienna „Prędkość” zmienia się przy każdym naciśnięciu klawisza w górę lub w dół, a wykres wyświetla jej wartość.

OBEJRZYJ WIDEO

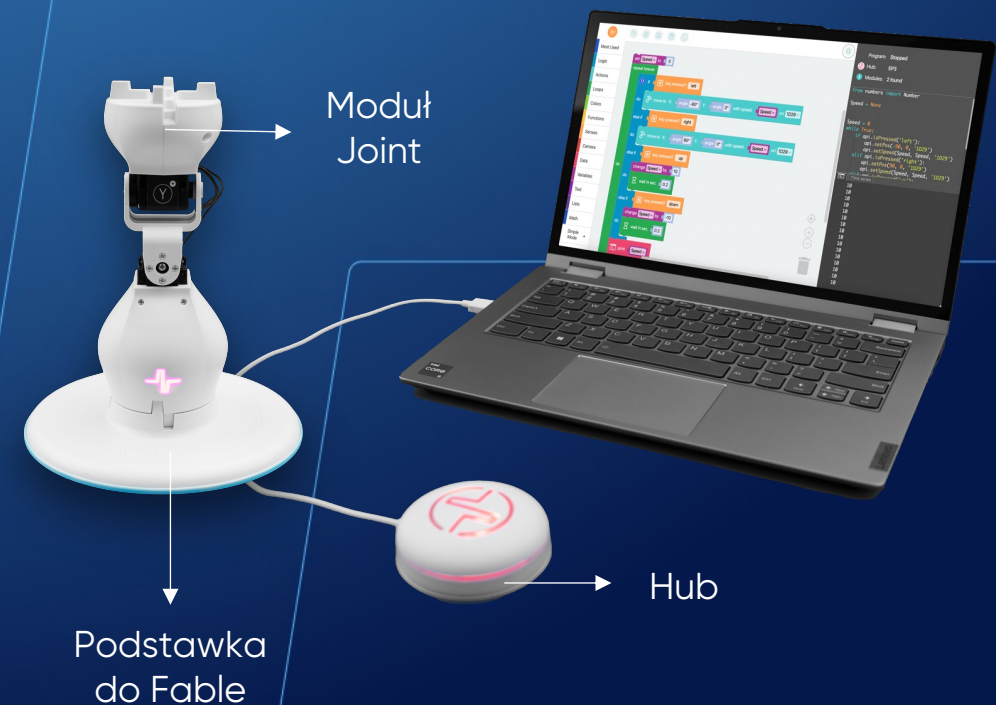


```
set Speed to 0
repeat while true
  if key pressed? up
    change Speed by 10
  do
    wait in sec. 0.2
  else if key pressed? down
  do
    change Speed by -10
  do
    wait in sec. 0.2
  set speed A: Speed B: Speed on 1111
  time series Speed with color red
```

Kod 10: Kontrola prędkości Modułu Joint

- Sekwencja programu zawiera polecenia, aby silnik X Modułu Joint przesunął się odpowiednio do -90 stopni i 90 stopni, gdy zostaną naciśnięte klawisze strzałek w lewo i w prawo. Prędkość jest określana przez wartość, która zwiększa się lub zmniejsza o 10 jednostek po naciśnięciu klawisza strzałki w górę lub w dół. Ta wartość jest przechowywana w zmiennej „Prędkość” i wyświetlana w Konsoli Wyjściowej.

OBEJRZYJ WIDEO



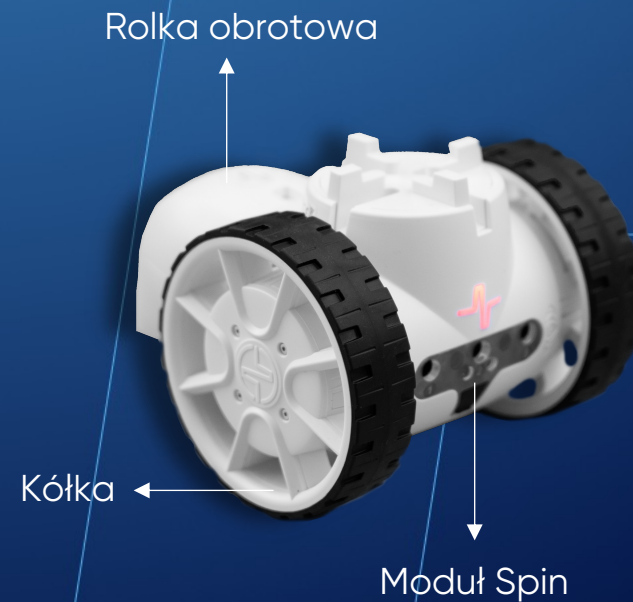
```
set Speed to 0
repeat forever
  if key pressed? left
    do move to X: angle -90° Y: angle 0° with speed: Speed on 1D29
  else if key pressed? right
    do move to X: angle 90° Y: angle 0° with speed: Speed on 1D29
  do
    else if key pressed? up
      change Speed by 10
      do wait in sec. 0.2
    else if key pressed? down
      change Speed by -10
      do wait in sec. 0.2
  print Speed
```

Kod 11:

Jazda zygzakiem pod kątem 90°

- Moduł Spin będzie poruszał się według wcześniej ustalonego wzoru, prostą linią z naprzemiennymi skrętami w prawo i w lewo o 90 stopni. Zmiana kąta obrotu tworzy wzory ruchu w stylu zygzak.

OBEJRZYJ WIDEO




```
repeat while true  
  drive 20 cm on A21B  
  wait until | has reached target on motor(s) both on A21B  
  spin -90 degrees with speed: 50 on A21B  
  wait until | has reached target on motor(s) both on A21B  
do  
  drive 20 cm on A21B  
  wait until | has reached target on motor(s) both on A21B  
  spin 90 degrees with speed: 50 on A21B  
  wait until | has reached target on motor(s) both on A21B
```

Kod 12: Robot czteronożny

- Program używa czterech Modułów Joint do wykonywania ruchów robota czteronożnego. Biorąc pod uwagę dużą liczbę zmiennych i ich użycie w wielu miejscach bloków poleceń, zaleca się staranne złożenie robota, ponieważ każde nieprawidłowe umieszczenie Modułu Wspólnego spowoduje nieudany ruch. Za pomocą przycisków można sterować robotem zarówno w zakresie ruchu do przodu/do tyłu, jak i obrotu.

OBEJRZYJ WIDEO



Moduł Joint



```

set frontRight to module LUB
set frontLeft to module EZE
set backRight to module HUB
set backLeft to module UWB
set freq to 300
set xAmp to 40
set yAmp to 60
repeat while true
  UpdateMotorVariables
  if key pressed? up
  do Forward
  else if key pressed? down
  do Back
  else if key pressed? left
  do CounterClockwise
  else if key pressed? right
  do Clockwise
  
```

```

to UpdateMotorVariables
set x0 to xAmp * sin * freq * time in sec.
set x1 to xAmp * cos * freq * time in sec.
set y0 to yAmp * cos * freq * time in sec.
set y1 to yAmp * cos * freq * time in sec.
  
```

```

to Forward
move to X: x0 - 75 Y: y1 + 30 with speed: 100 on # frontLeft
move to X: x0 - 75 Y: y1 - 30 with speed: 100 on # frontRight
move to X: x0 - 75 Y: y0 + 0 with speed: 100 on # backLeft
move to X: x0 - 75 Y: y1 - 0 with speed: 100 on # backRight

to Back
move to X: x0 - 75 Y: y1 + 30 with speed: 100 on # frontLeft
move to X: x0 - 75 Y: y1 - 30 with speed: 100 on # frontRight
move to X: x0 - 75 Y: y0 + 0 with speed: 100 on # backLeft
move to X: x0 - 75 Y: y1 - 0 with speed: 100 on # backRight

to CounterClockwise
move to X: x0 - 75 Y: y1 + 30 with speed: 100 on # frontLeft
move to X: x0 - 75 Y: y1 + 30 with speed: 100 on # backRight
move to X: x0 - 75 Y: y1 - 0 with speed: 100 on # frontRight
move to X: x0 - 75 Y: y1 - 0 with speed: 100 on # backLeft

to Clockwise
move to X: x0 - 75 Y: y1 + 30 with speed: 100 on # frontLeft
move to X: x0 - 75 Y: y1 + 30 with speed: 100 on # backRight
move to X: x0 - 75 Y: y1 - 0 with speed: 100 on # frontRight
move to X: x0 - 75 Y: y1 - 0 with speed: 100 on # backLeft
  
```

Kod 13: Ruchomy Moduł Joint wyposażony w kółka

- Pomimo braku obracających się kół, Moduł Joint nadal może być wprowadzony w ruch. Ta sekwencja programu wykorzystuje ruch silnika X do napędzania robota do przodu poprzez dostosowanie przekładni. Wartość kąta jest wyświetlana w sposób graficzny, co ułatwia zrozumienie i dostosowanie prędkości lub trybu ruchu.

OBEJRZYJ WIDEO



Moduł Joint



Kod 14: Jazda Modułem Spin według ustalonych kierunków

- Program pozwala nam na wcześniejsze określenie ruchów Modułu Spin. Naciskając klawisze strzałek, wypełniamy listę o nazwie „Kroki”. Po naciśnięciu Spacji, program przechodzi przez listę i zleca Modułowi Spin poruszanie się zgodnie z wcześniej ustalonymi instrukcjami.

OBEJRZYJ WIDEO



Rolka obrotowa ←

← Kółka

Moduł Spin

→ Hub



```
set Spin module to module A21B
set Steps to create empty list

repeat until key pressed? spacebar
  if key pressed? up
    in list Steps insert at last as "UP"
    do
      wait in sec. 0.5
      speak "10cm" English
  else if key pressed? left
    in list Steps insert at last as "LEFT"
    do
      wait in sec. 0.5
      speak "Left" English
  else if key pressed? right
    in list Steps insert at last as "RIGHT"
    do
      wait in sec. 0.5
      speak "Right" English
  else if key pressed? down
    in list Steps insert at last as "DOWN"
    do
      wait in sec. 0.5
      speak "-10cm" English
```

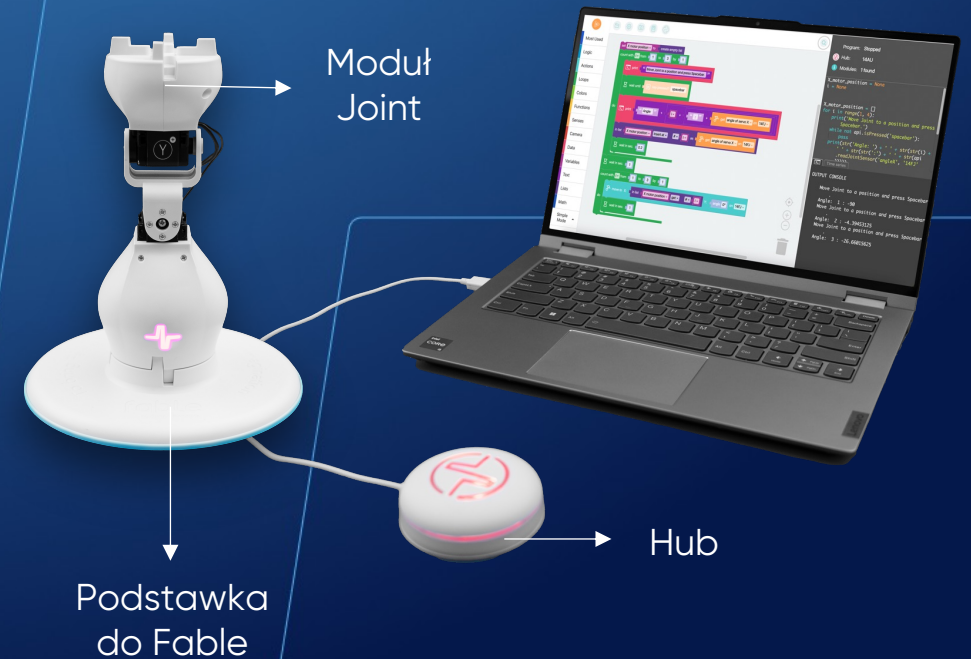


```
for each item Steps in list Steps
  if Steps = "UP"
    drive 10 cm with speed: 50 on # Spin module
    do
      wait until has reached target on motor(s) both on A21B
  else if Steps = "LEFT"
    spin -90 degrees with speed: 50 on # Spin module
    do
      wait until has reached target on motor(s) both on A21B
  else if Steps = "RIGHT"
    spin 90 degrees with speed: 50 on # Spin module
    do
      wait until has reached target on motor(s) both on A21B
  else if Steps = "DOWN"
    drive -10 cm with speed: 50 on # Spin module
    do
      wait until has reached target on motor(s) both on A21B
  wait in sec. 1
```

Kod 15: Ruch Modułu Joint według ustalonych kierunków

- Program przechowuje trzy wartości kąta silnika X w Module Joint. Ustaw silnik X pod kątem i naciśnij Spację, aby go zapisać. Przechowywanie odbywa się za pomocą listy, która jest następnie używana do pobierania zapisanych wartości. Dzięki tym wartościom Moduł Joint będzie mógł odtworzyć nauczone ruchy.

OBEJRZYJ WIDEO

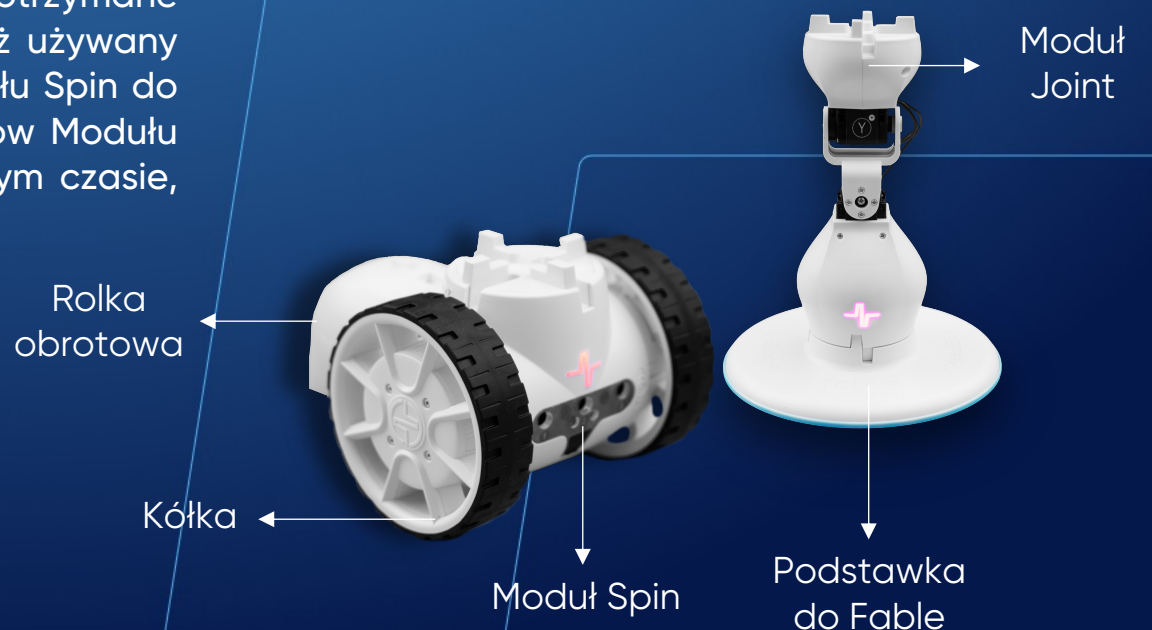


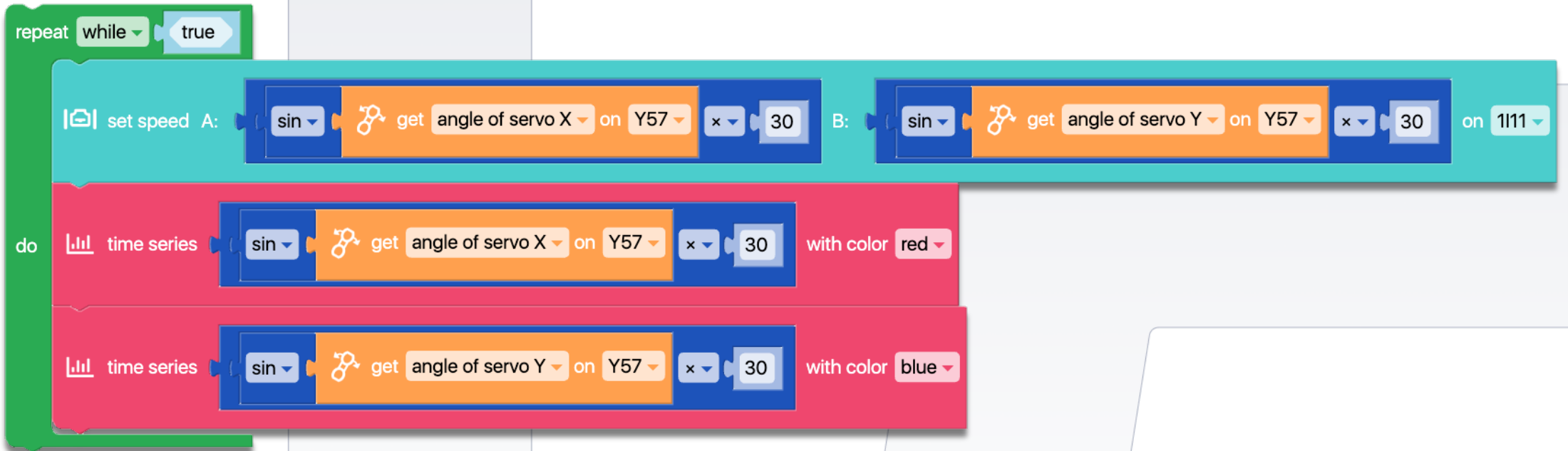

```
set X motor position to create empty list
count with i from 0 to 3 by 1
  print "Move Joint to a position and press Spacebar."
  wait until key pressed? spacebar
  do
    print "Angle: " + i + " : " + get angle of servo X on 14FJ
    in list X motor position insert at # i as get angle of servo X on 14FJ
    wait in sec. 0.2
  wait in sec. 1
count with i from 0 to 3 by 1
  do
    move to X: in list X motor position get # i Y: angle 0° on 14FJ
    wait in sec. 1
```

Kod 16: Ruch trygonometryczny

- Program wykorzystuje kąty silników w Module Joint do obliczania wartości ich sinusów. Te wartości są następnie mnożone i przesyłane do dwóch silników Modułów Spin, aby kontrolować ich ruchy. Mnożenie jest niezbędne, ponieważ sama wartość sinusa nie wystarcza do skutecznego poruszania silnikami. Wykres wyświetla dane otrzymane przez silniki Modułów Spin. Ten program może być również używany jako gra - celem jest jak najszybsze przemieszczenie Modułu Spin do określonej lokalizacji (poprzez kombinację kątów dla silników Modułu Wspólnego). Osoba, która wykona to zadanie w najkrótszym czasie, wygrywa grę!

OBEJRZYJ WIDEO





Kod 17: Sterowanie dotykiem Modułu Spin

- Sekwencja kodu steruje ruchem Modułu Spin na podstawie liczby palców wykrytych przez telefon Fable Face podłączony do komputera, na którym wykonywane jest programowanie. Wykres ilustruje liczbę wykrytych palców w każdym momencie.

OBEJRZYJ WIDEO

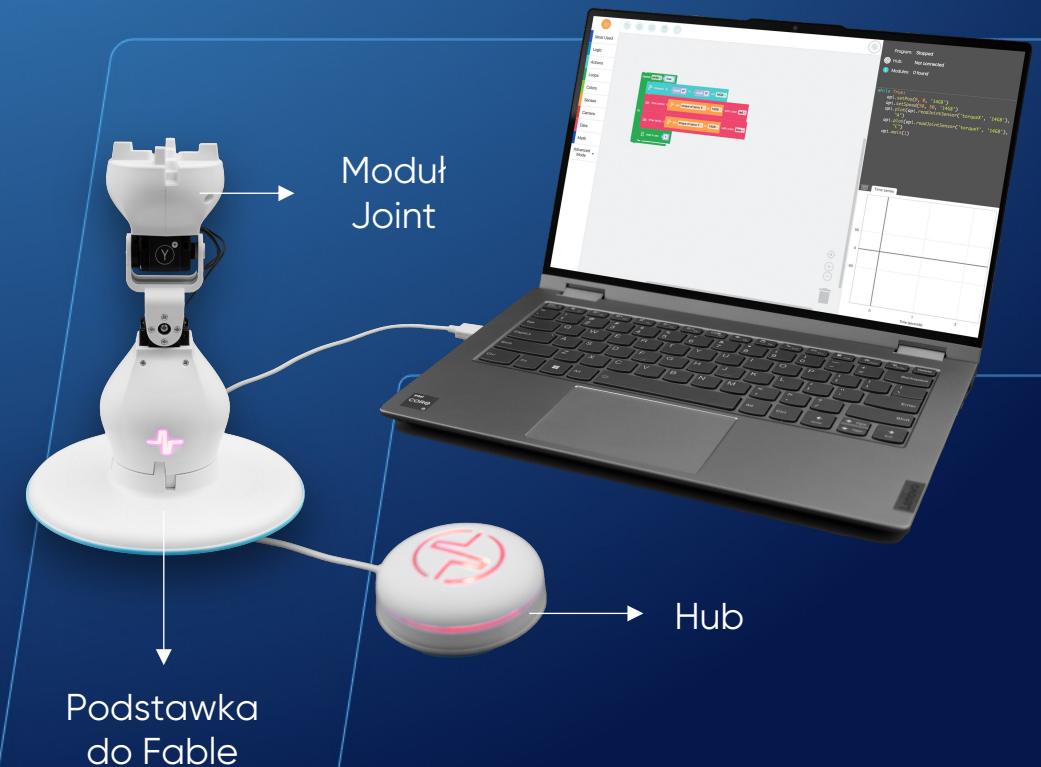


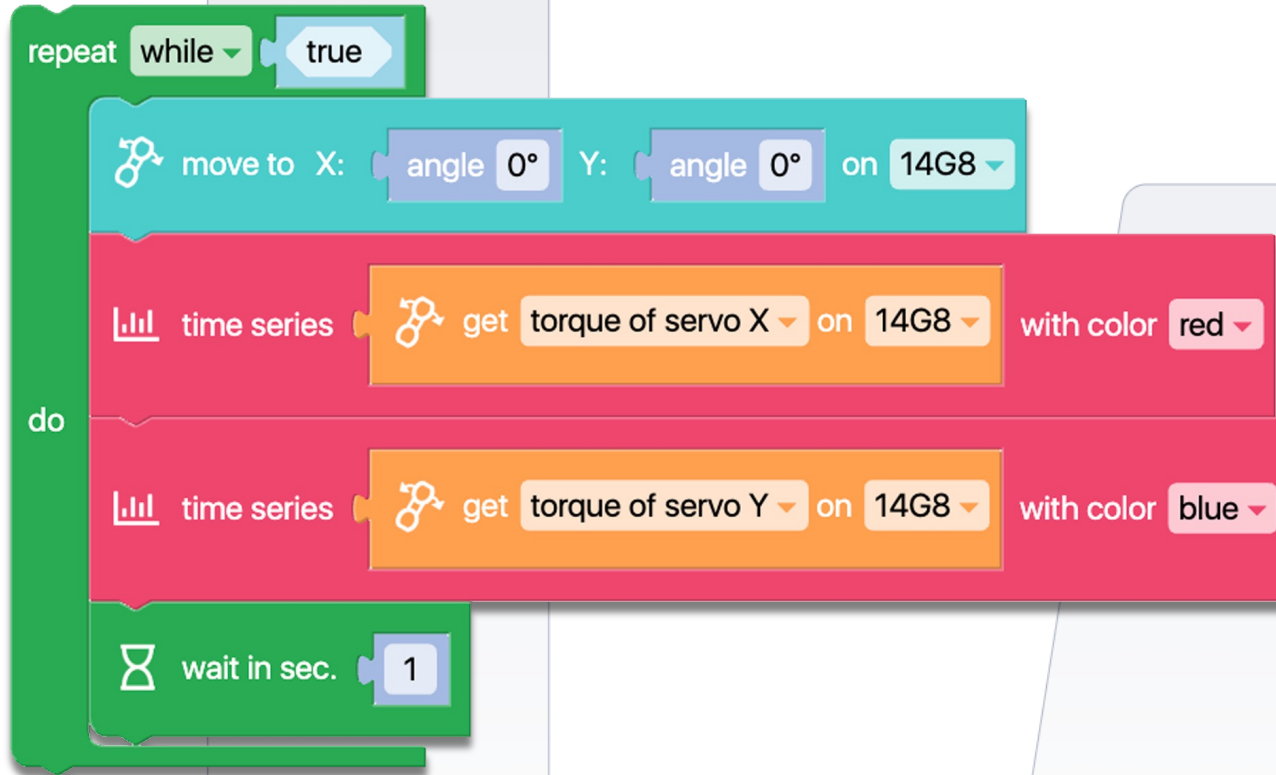
```
repeat while true  
  if get tap count = 1  
    do move forward on 1111  
  else if get tap count = 2  
    do move backward on 1111  
  else if get tap count = 3  
    do right on 1111  
  else if get tap count = 4  
    do left on 1111  
  else stop moving on 1111  
time series get tap count with color red
```

Kod 18: Pomiar momentu obrotowego Modułu Joint

- Program na bieżąco monitoruje moment obrotowy silników X i Y Modułu Joint. Każdy odczyt jest natychmiast wyświetlany w Konsoli Wyjściowej i może być również zapisany w pliku .csv. Aby zwiększyć czytelność, zastosowano blok tekstowy zawierający wartości odczytane przez sensory. Funkcja ta zapewnia bardziej przejrzyste i zrozumiałe wyświetlanie. Co więcej, program można ulepszyć poprzez zaimplementowanie polecenia operatora porównania w celu wykrycia przypadków, w których moment obrotowy przekracza wstępnie zdefiniowany próg. Ten dodatek umożliwiłby stworzenie jeszcze bardziej przyjaznego do użytkownika i łatwego w interpretacji wyświetlania.

OBEJRZYJ WIDEO

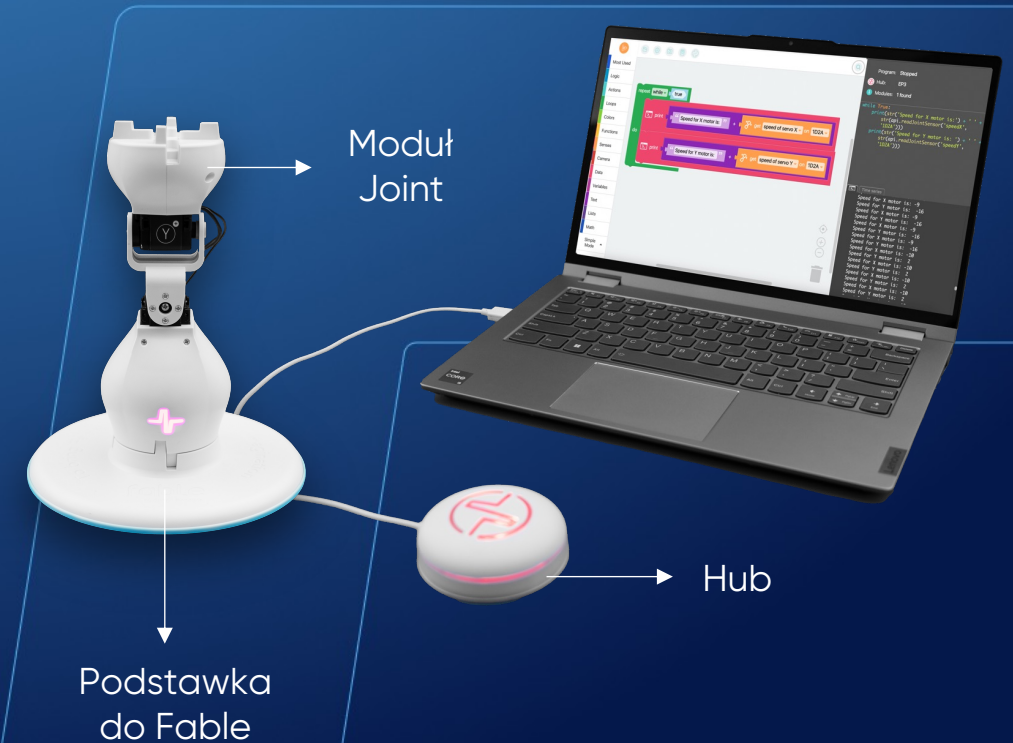




Kod 19: Pomiar prędkości serwomotorów Modułu Joint

- Program wykorzystuje bloki do "Pomiaru Momentu Obrotowego Modułu Joint" - w tym przypadku odczytywana jest prędkość ruchu silników X i Y Modułu Wspólnego. Wyświetlanie odbywa się w formacie numerycznym przy użyciu Konsoli Wyjściowej. Program można rozszerzyć dodając polecenie operatora porównania do wykrywania, kiedy prędkość przekracza określoną wartość.

OBEJRZYJ WIDEO




```
repeat while true  
do  
  print "Speed for X motor is:" + get speed of servo X on 1D2A  
  print "Speed for Y motor is:" + get speed of servo Y on 1D2A
```

Kod 20:

Pomiar przyspieszenia Modułu Spin

- Program umożliwia sterowanie Modułem Spin za pomocą klawiszy kierunkowych na klawiaturze. Jednocześnie wyświetla wartość przyspieszenia zarejestrowaną na jednej z osi ruchu na Twoim telefonie, który znajduje się na Module Wspólnym. Należy pamiętać, aby Fable Face był włączony i połączony z Hubem.

OBEJRZYJ WIDEO



Rolka obrotowa

2X Moduł

Kółka

Moduł Spin



```
repeat while true
  if key pressed? up
  do move forward on 1111
  else if key pressed? down
  do move backward on 1111
  else if key pressed? left
  do left on 1111
  else if key pressed? right
  do right on 1111
  else stop moving on 1111
  print "Module Acceleration is: " + get acceleration on Y-axis
```

Kod 21:

Porównywanie dwóch wartości

- Ten program może być również użyty w formie gry. Po uruchomieniu, silnik X Modułu Wspólnego zostanie ustawiony w losowym kącie między -90 a 90 stopni. Celem gry jest dostosowywanie przez użytkownika pozycji drugiego Modułu Joint aż do momentu określenia kąta pierwszego silnika. Po zidentyfikowaniu kąta, Konsola Wyjściowa wyświetli jego wartość, Hub zaświeci się na zielono, i odegrany zostanie dźwięk muzyczny 'Si'. Ważne jest, aby unikać zbyt dużego nacisku na silniki poprzez zbyt gwałtowne lub szybkie obracanie Modułem Joint.

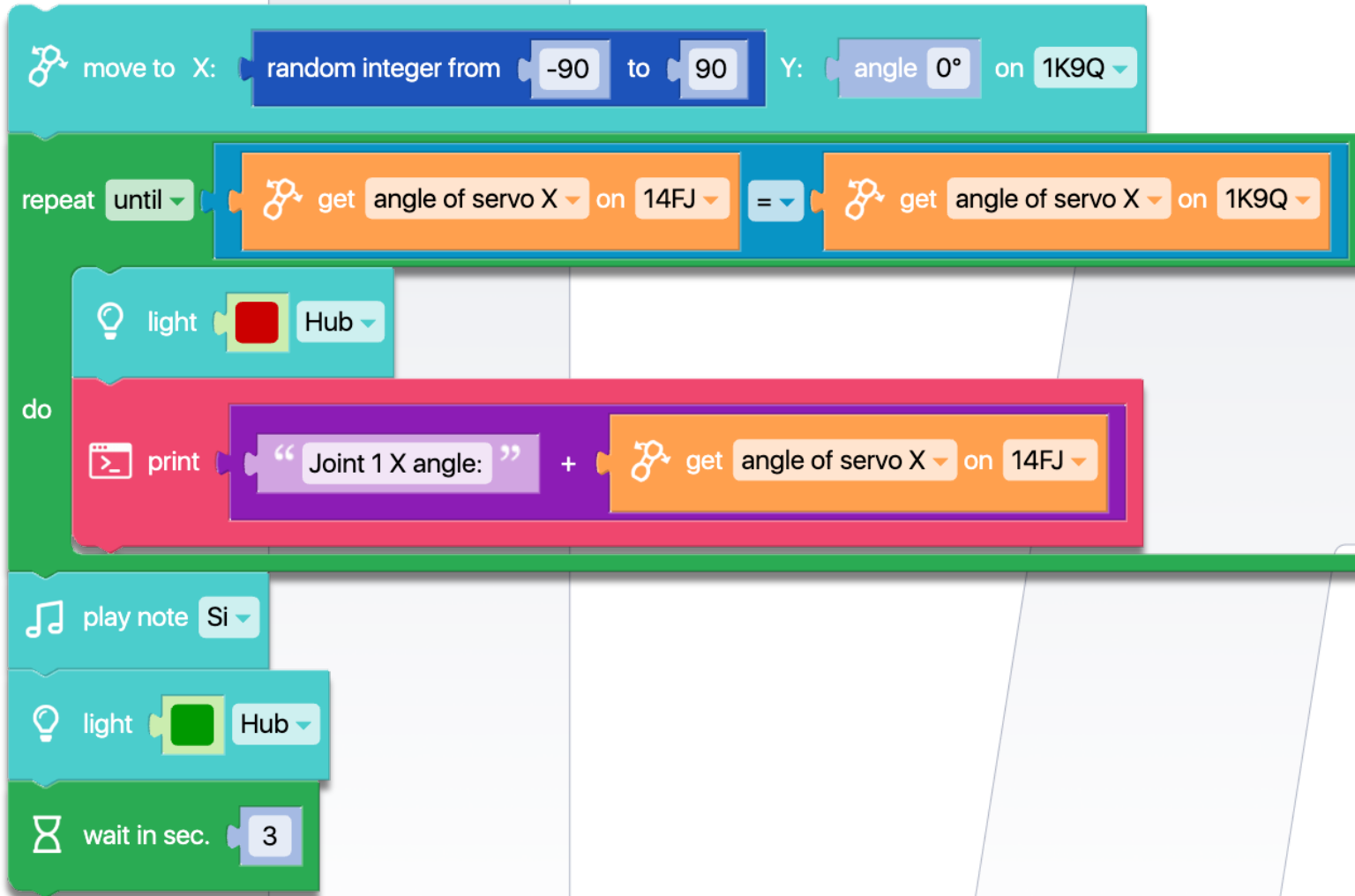
OBEJRZYJ WIDEO



Moduł Joint ←

Podstawka do Fable ←

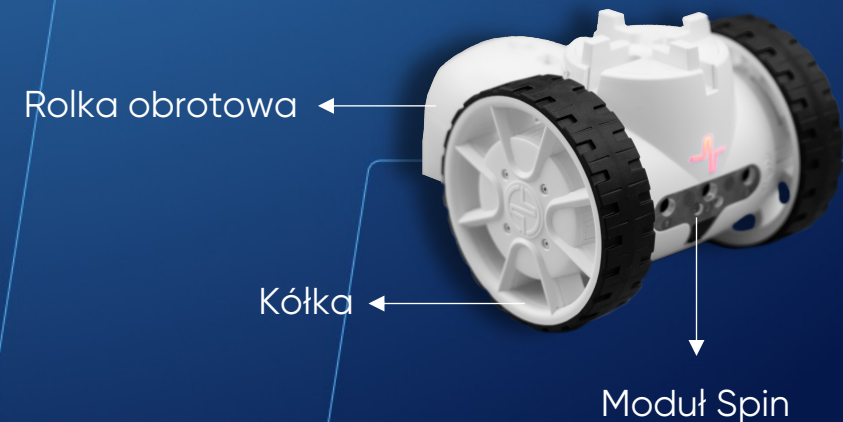


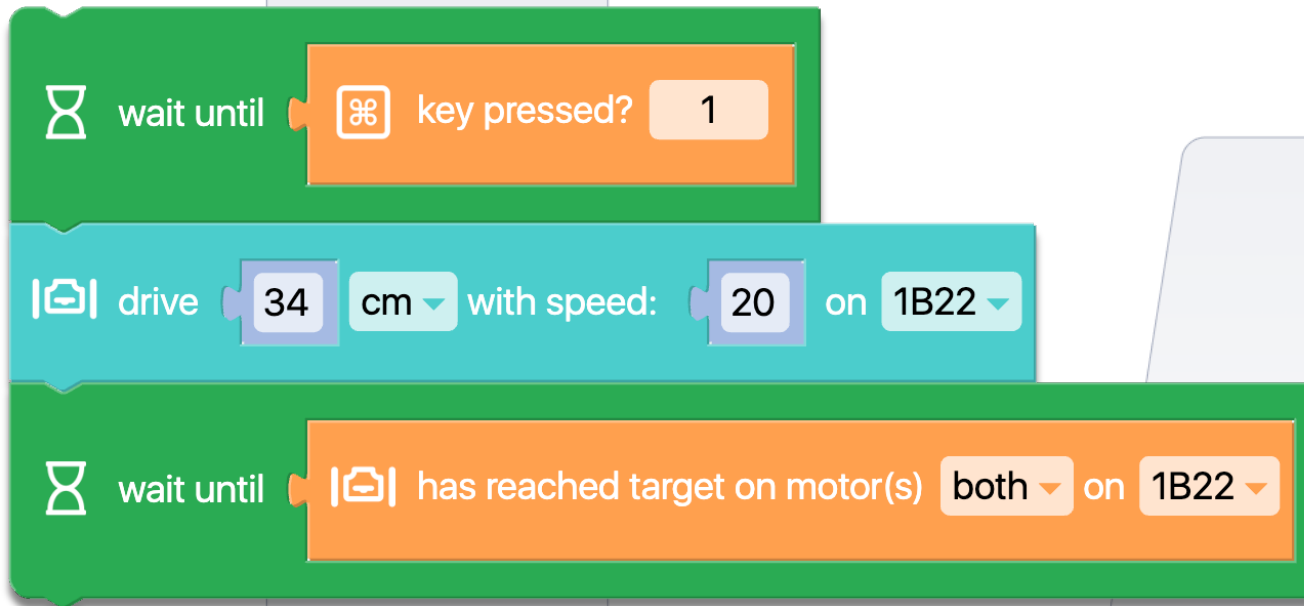


Kod 22: Pomiar obwodu Koła Fable

- Ten kod pokazuje, jak zmierzyć obwód koła Fable. Najpierw oznaczamy punkt na kole, następnie uruchamiamy kod, a Moduł Spin przesunie się do przodu na odległość 34 cm. Po zakończeniu obserwujemy, że znak powrócił do swojej początkowej pozycji, co wskazuje na ukończenie pełnego obrotu. Zatem rozwijając okrąg, ustalamy, że obwód koła wynosi 34 cm. Ta informacja jest przydatna podczas pomiaru przebytej odległości.

OBEJRZYJ WIDEO

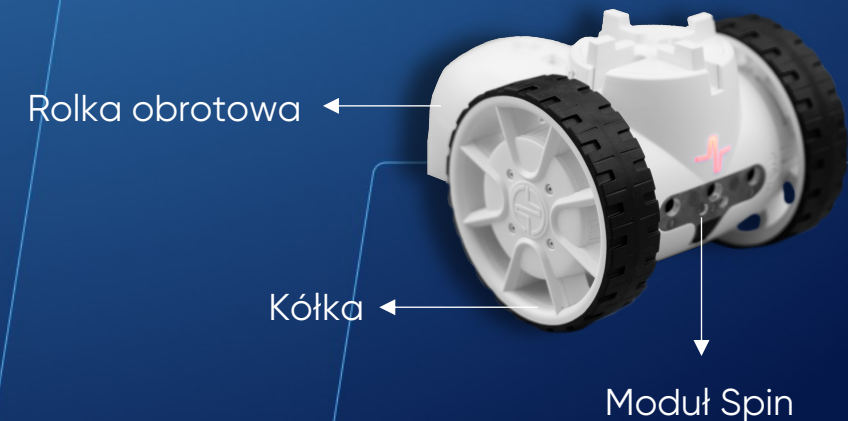


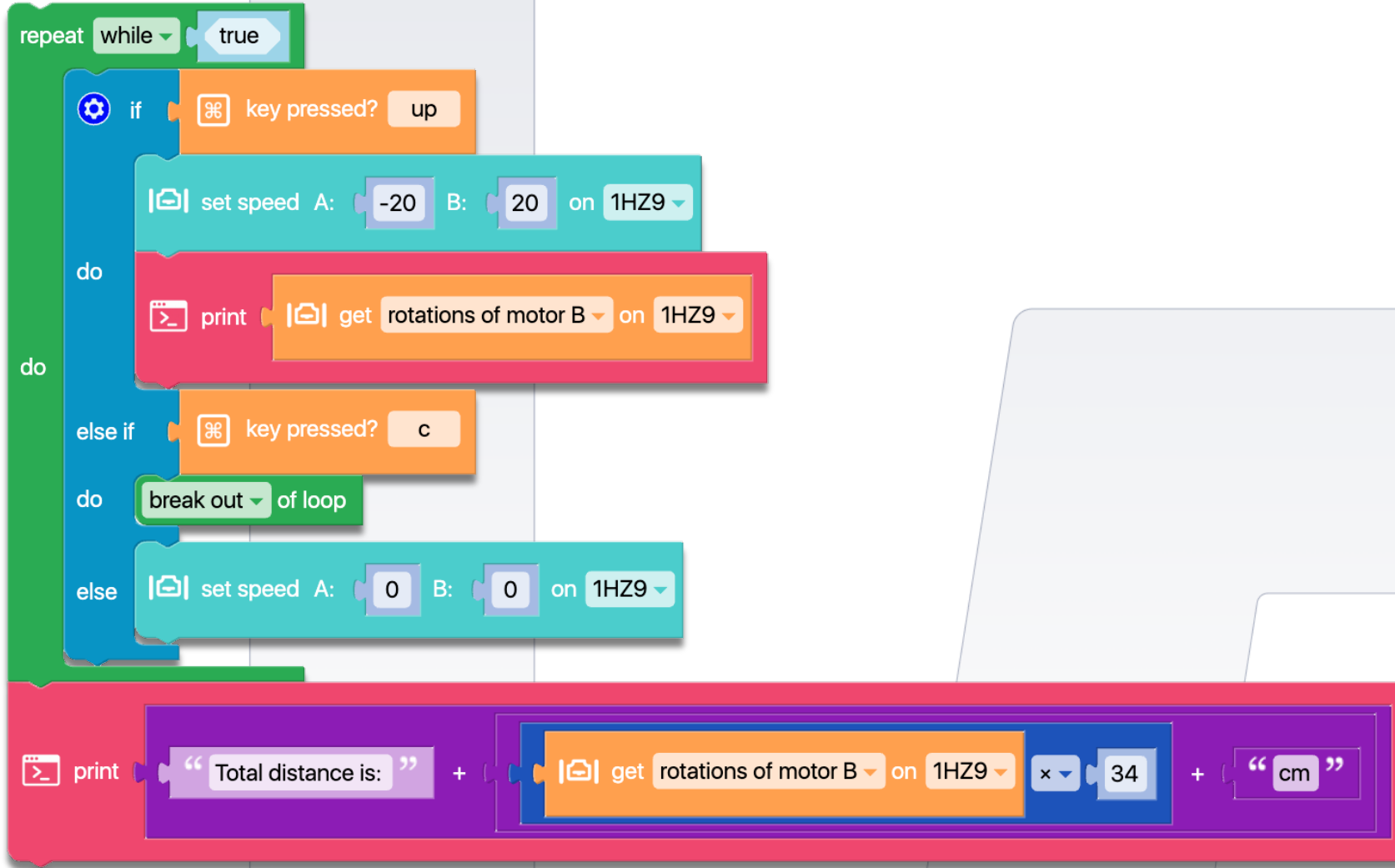


Kod 23: Pomiar odległości na podstawie obwodu Koła Fable

- Program mierzy odległość przebytą przez Moduł Spin, uwzględniając obwód koła. W naszym przypadku koło ma obwód 34 cm. Na bieżąco monitorujemy liczbę pełnych obrotów wykonanych przez silnik B (który obraca się w kierunku dodatnim). Wyjście z pętli i finalne obliczenia następują po naciśnięciu klawisza 'c'.

OBEJRZYJ WIDEO

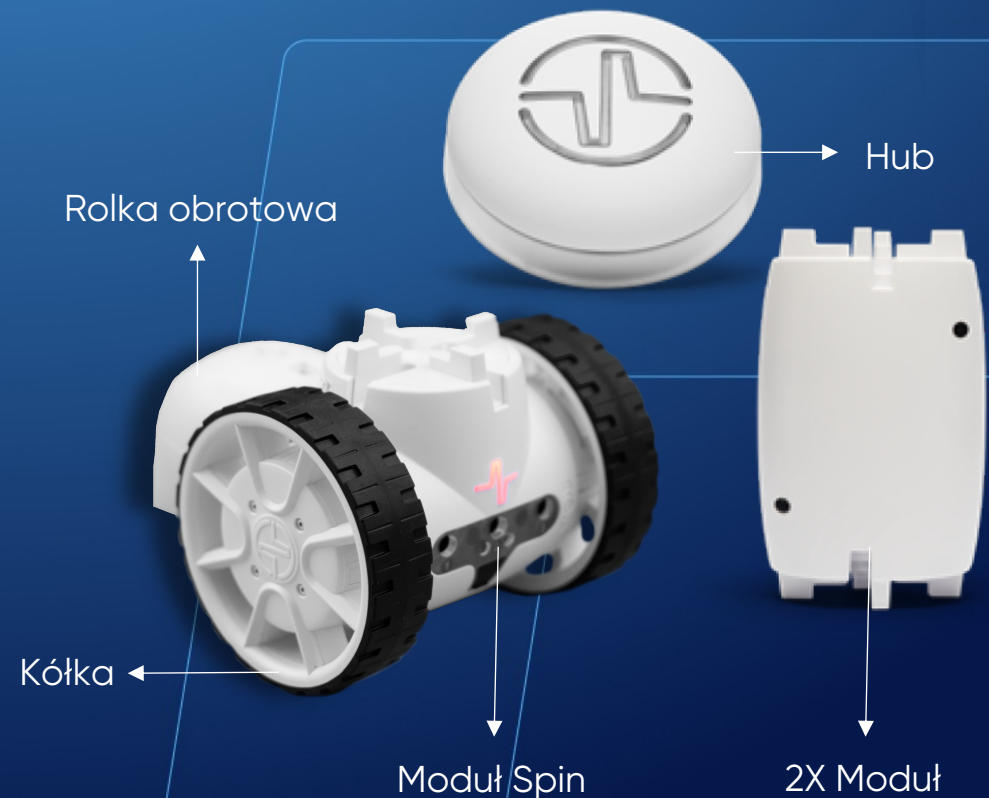


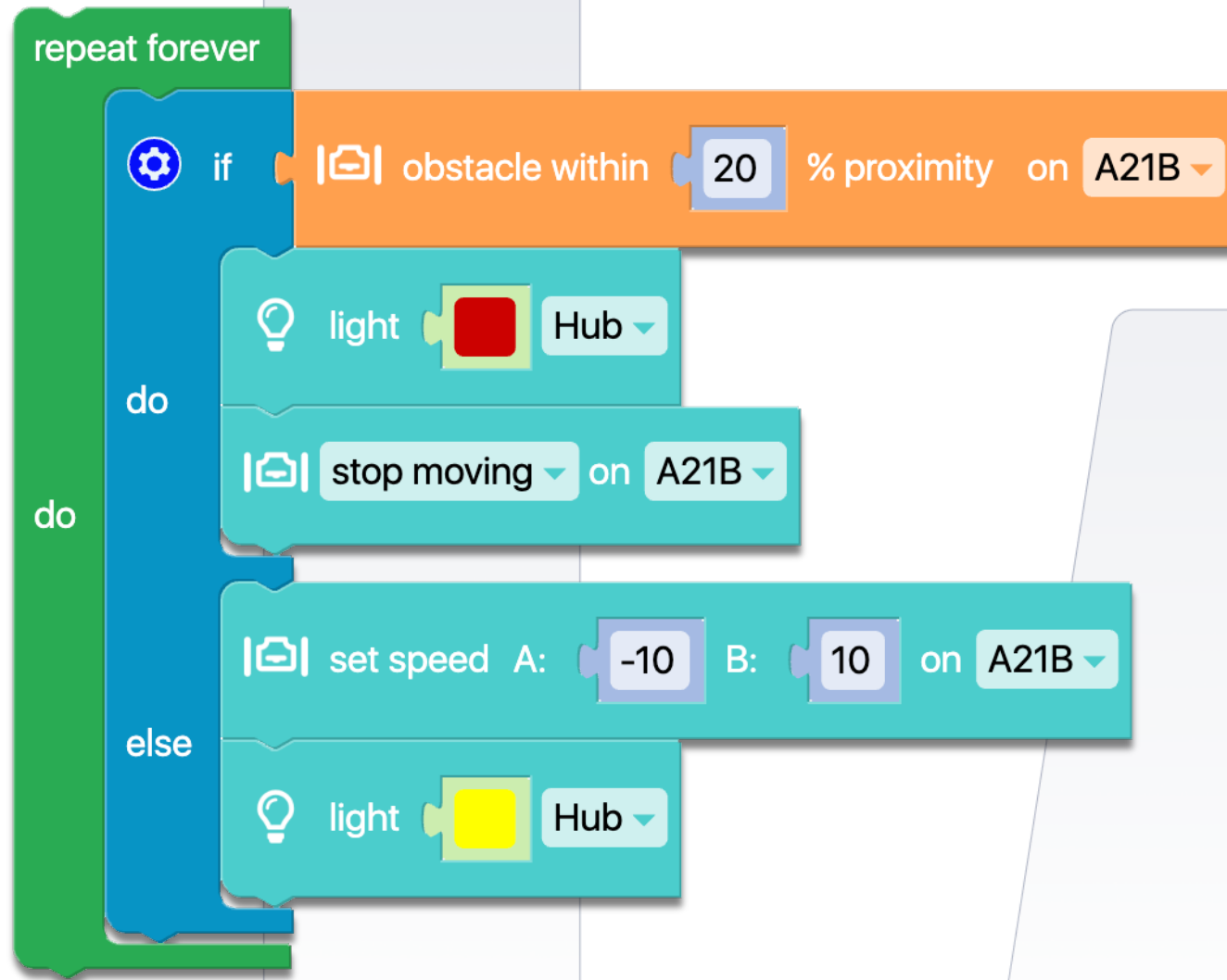


Kod 24: Wykrywanie przeszkody

- Moduł Spin porusza się do przodu tylko wtedy, gdy nie ma przed nim przeszkód. W tym czasie Hub jest podświetlony na żółto. Konsola Wyjściowa na bieżąco wyświetla dane wykrywane przez sensor zbliżeniowy. Gdy pojawi się przeszkoda, Moduł Spin zatrzymuje się i pozostaje w tym stanie, dopóki przeszkoda znajduje się przed nim. W tym czasie Hub jest podświetlony na czerwono.

OBEJRZYJ WIDEO

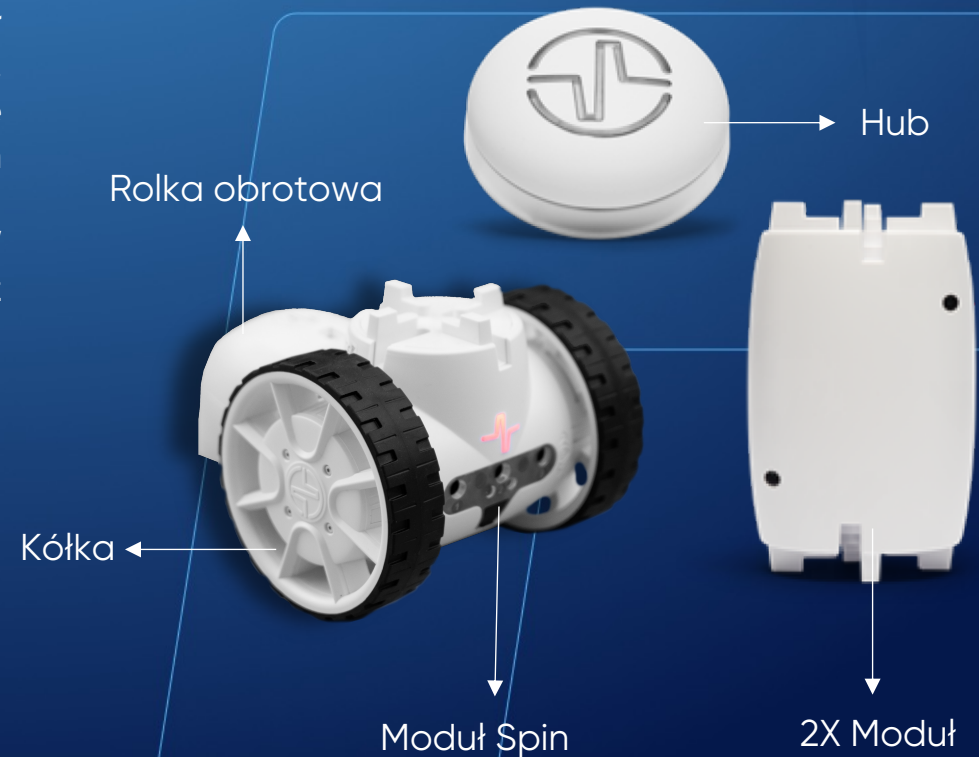




Kod 25: Unikanie przeszkody – Metoda 1

- Program opiera się na kodzie "Wykrywanie Przeszkody". Moduł Spin porusza się do przodu, chyba że napotka przeszkodę. Gdy zostanie wykryta przeszkoda, program uruchamia funkcję procedury unikania ("Avoid"), która używa podstawowych komend do wykonania akcji omijania przeszkody. Moduł 2X, który jest akcesorium Fable, został uznany za przeszkodę. W zależności od przeszkody, którą chcesz ominąć, będziesz musiał zmodyfikować wartości liczbowe w programie.

OBEJRZYJ WIDEO



```
repeat forever
  if obstacle within 10 % proximity on 1HZ9
  do Avoid procedure
  else
    move forward on 1HZ9
    light yellow Hub
  print get proximity from sensor 1 on 1HZ9
```

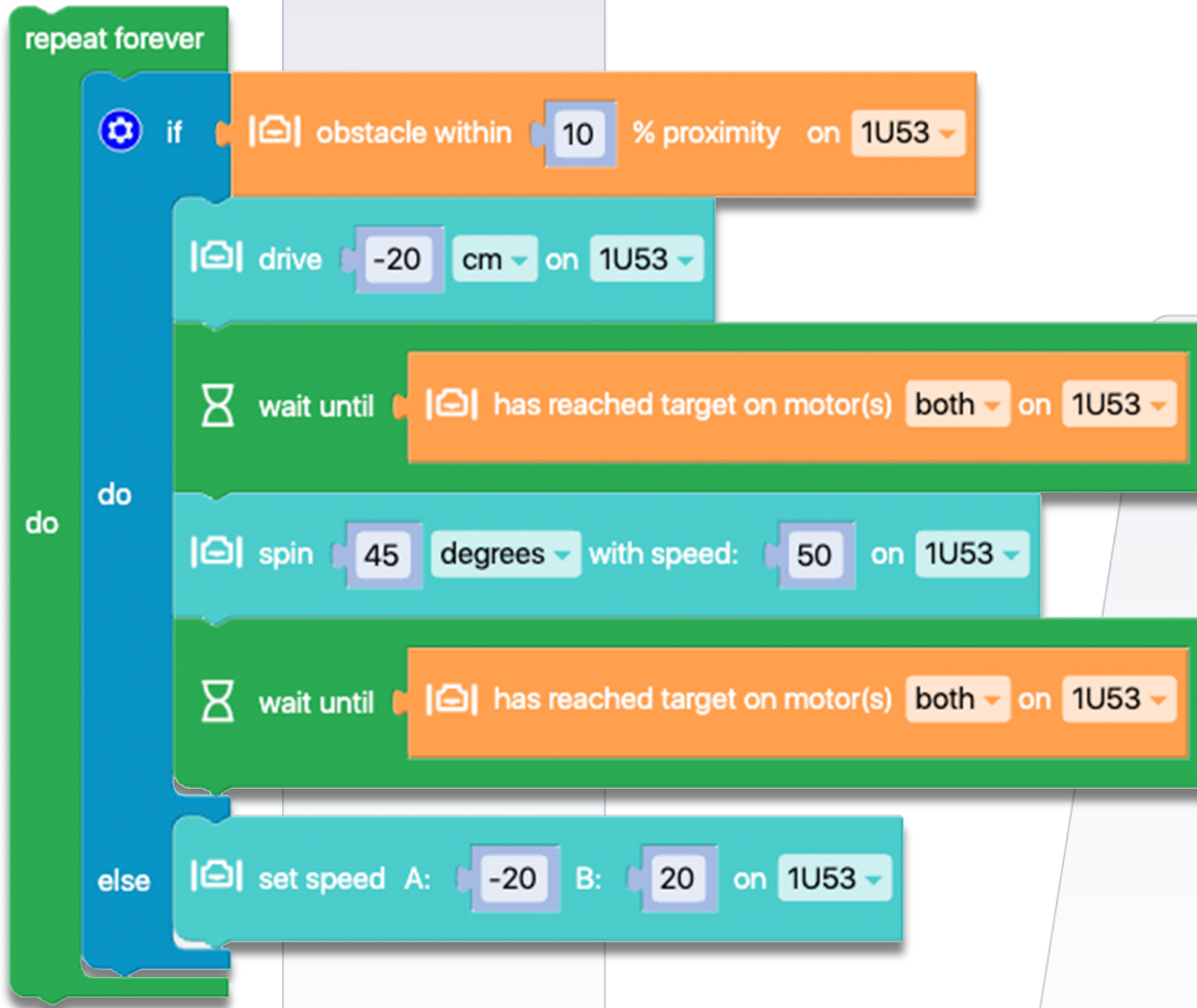
```
to Avoid procedure
  light red Hub
  stop moving on 1HZ9
  drive -30 cm with speed: 50 on 1HZ9
  wait until has reached target on motor(s) both on 1HZ9
  set speed A: -30 B: 60 on 1HZ9
  wait in sec. 1
  drive 10 cm with speed: 50 on 1HZ9
  wait until has reached target on motor(s) both on 1HZ9
  set speed A: -60 B: 30 on 1HZ9
  wait in sec. 2
  set speed A: -30 B: 60 on 1HZ9
  wait in sec. 1
```

Kod 26: Unikanie przeszkody – Metoda 2

- Moduł Spin jest zaprojektowany do poruszania się do przodu, ale tylko wtedy, gdy na jego drodze nie ma przeszkód. Za każdym razem, gdy wykryta zostanie przeszkoda, Moduł cofa się o 20 cm i obraca o 45 stopni, aby ominąć przeszkodę, a następnie kontynuuje ruch do przodu. Aby zapewnić precyzyjne wykonanie poleceń, użyto dwóch komend „Czy osiągnięto cel na motorze (motorach)” („Has reached target on motor(s)”), które zapobiegają wykonaniu następnego polecenia, dopóki poprzednie nie zostanie zakończone.

OBEJRZYJ WIDEO

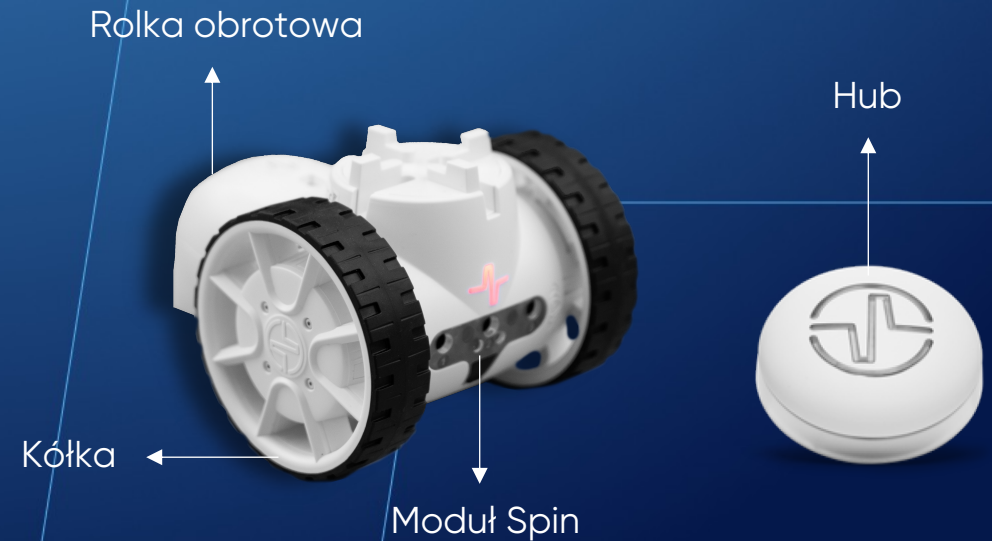




Kod 27: Wykrywanie kolorów

- Program aktywuje sensory koloru, wprowadzając Moduł Spin w tryb wykrywania. Jeśli robot wykryje przed sobą czerwony obiekt, program wyda Hubowi polecenie wyświetlenia czerwonego koloru.
- Taka sama procedura jest stosowana do wykrywania zielonych, niebieskich i żółtych obiektów. Jeśli żaden z tych kolorów nie zostanie wykryty przed robotem, Hub będzie podświetlał się na biało.

OBEJRZYJ WIDEO

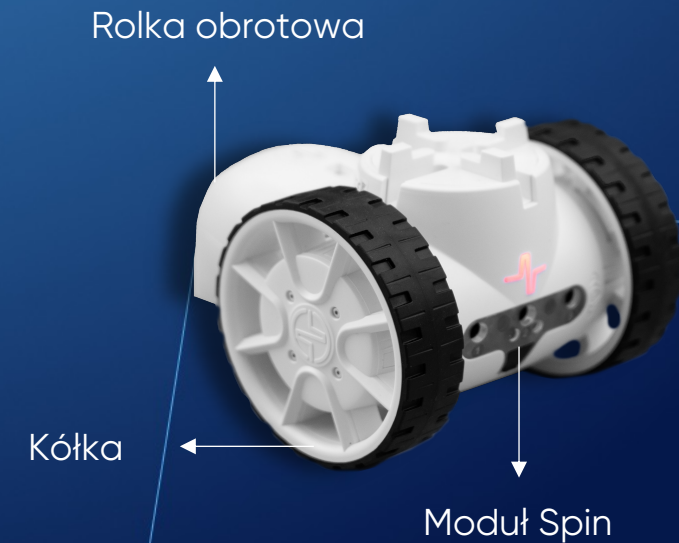




Kod 28: Pomiar światła otoczenia

- Program wykorzystuje sensor numer 2 do pomiaru poziomów światła otoczenia. Raz na sekundę Konsola Wyjściowa wyświetla wartość odczytaną przez sensor. Możliwe jest również użycie pozostałych dwóch sensorów.

OBEJRZYJ WIDEO

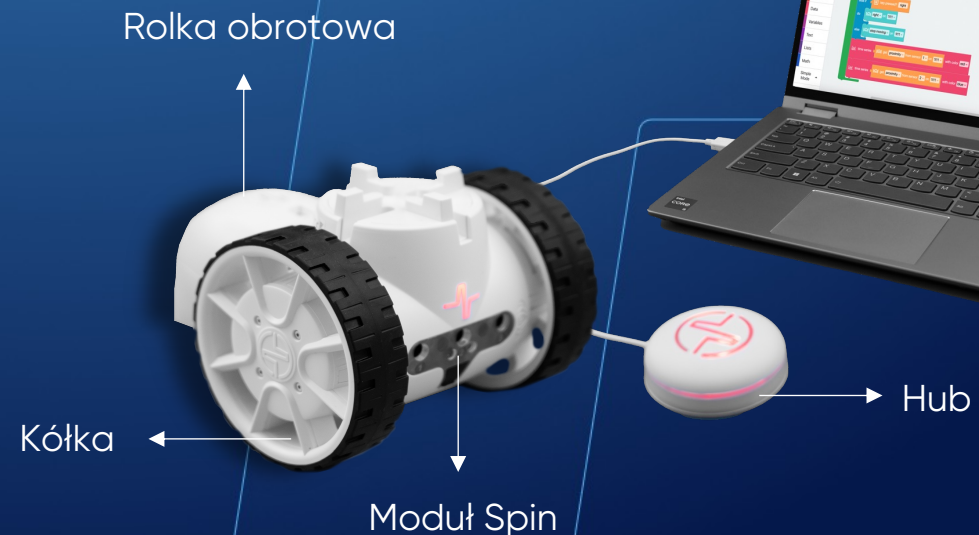




Kod 29: Pomiar odległości w pobliżu

- Program wykorzystuje "Sterowanie Klawiszami Modułu Spin". Dodano dwa nowe polecenia, które umożliwiają wyświetlanie wykresu zależnego od czasu, który przedstawia procent zbliżenia do przedmiotu wykrytego przez pierwszy i drugi sensor na robocie. Wykres jest wyświetlany w dwóch kolorach, aby umożliwić obserwację w czasie rzeczywistym oraz analizę wykrywania po zakończeniu programu.

OBEJRZYJ WIDEO

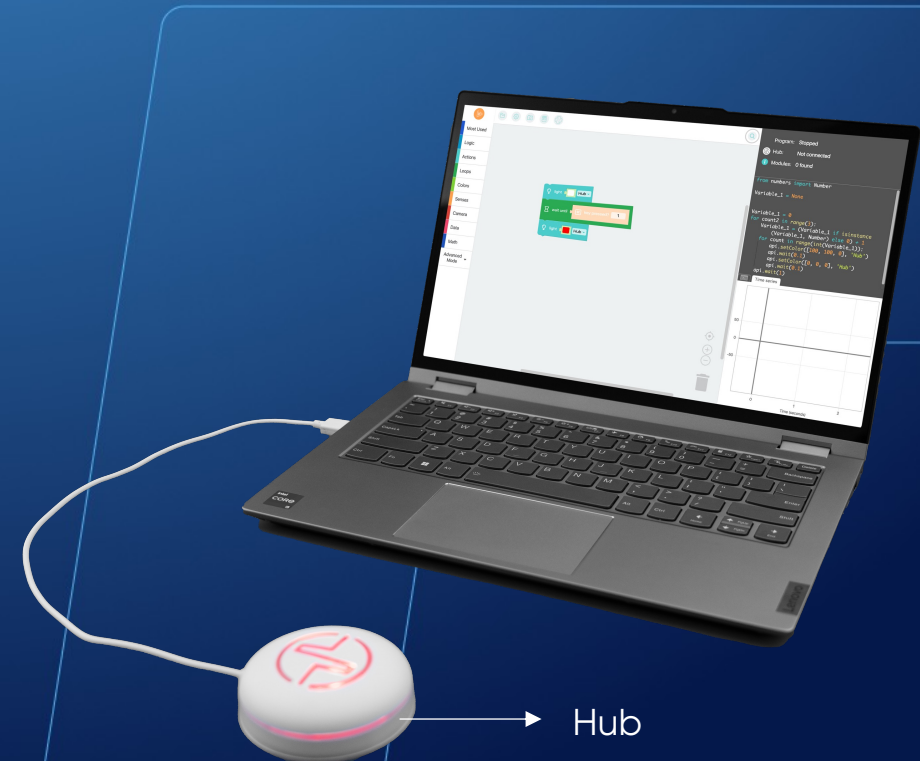


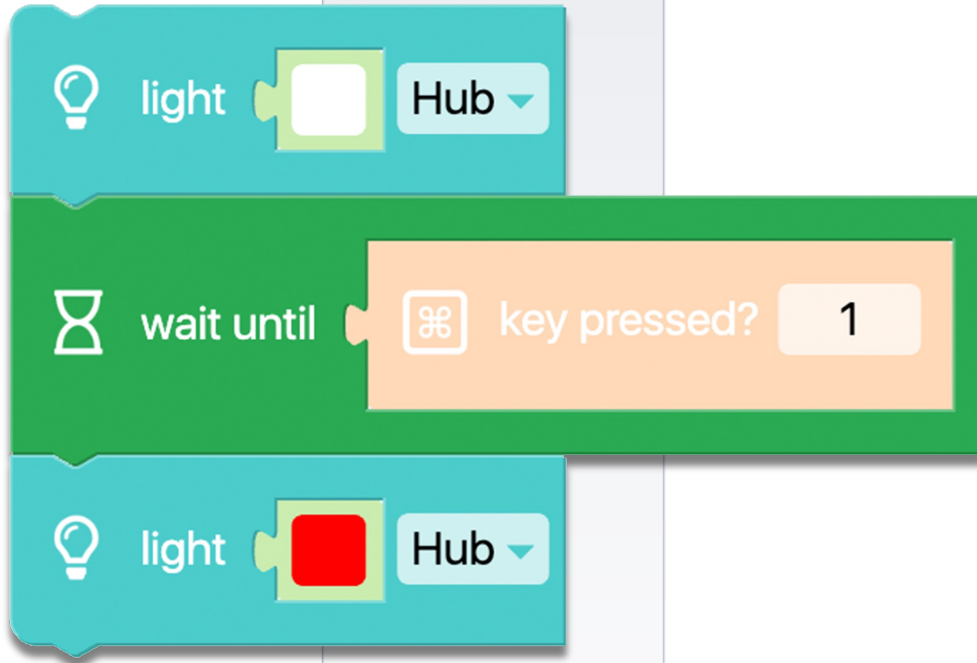
```
repeat forever
  if key pressed? up
  do move forward on 1111
  else if key pressed? down
  do move backward on 1111
  else if key pressed? left
  do left on 1111
  else if key pressed? right
  do right on 1111
  else stop moving on 1111
  time series get proximity from sensor 1 on 1111 with color red
  time series get proximity from sensor 2 on 1111 with color blue
```

Kod 30: Oczekiwanie na naciśnięcie klawisza

- Program uruchamia Hub z ustawionym białym światłem i pozostaje w tym stanie do momentu naciśnięcia klawisza 1. Żaden inny przycisk nie wpływa na kolor, lecz po naciśnięciu klawisza 1, Hub zmieni kolor na czerwony.

OBEJRZYJ WIDEO

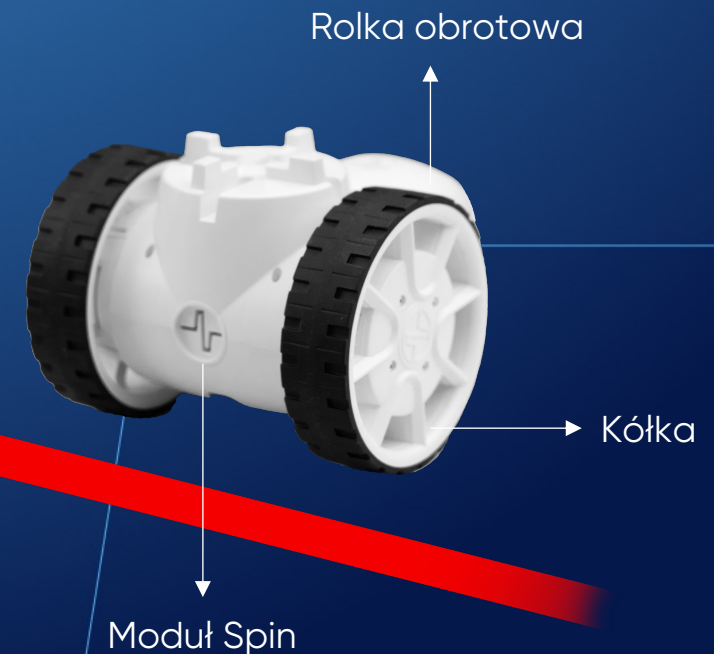




Kod 31: Wykrywanie kolorowej linii

- Moduł Spin porusza się do przodu w stałym tempie, aż napotka pod sobą czerwoną linię. Gdy wykryty zostanie czerwony kolor, program wydaje sygnał dźwiękowy, podświetla Hub na czerwono i rozpoczyna procedurę unikania przeszkody, cofając się i obracając pod określonym kątem, zanim wznowi ruch do przodu. Mamy możliwość dostosowania wartości, aby osiągnąć różne prędkości, wykrywać różne kolory i wykonywać inne manewry omijające.

OBEJRZYJ WIDEO

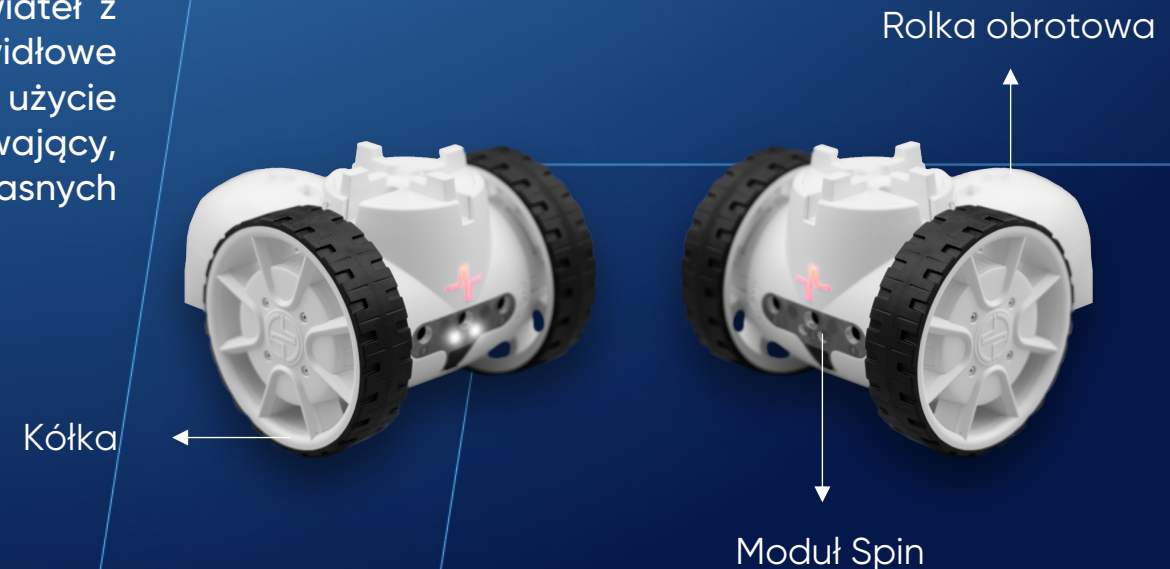


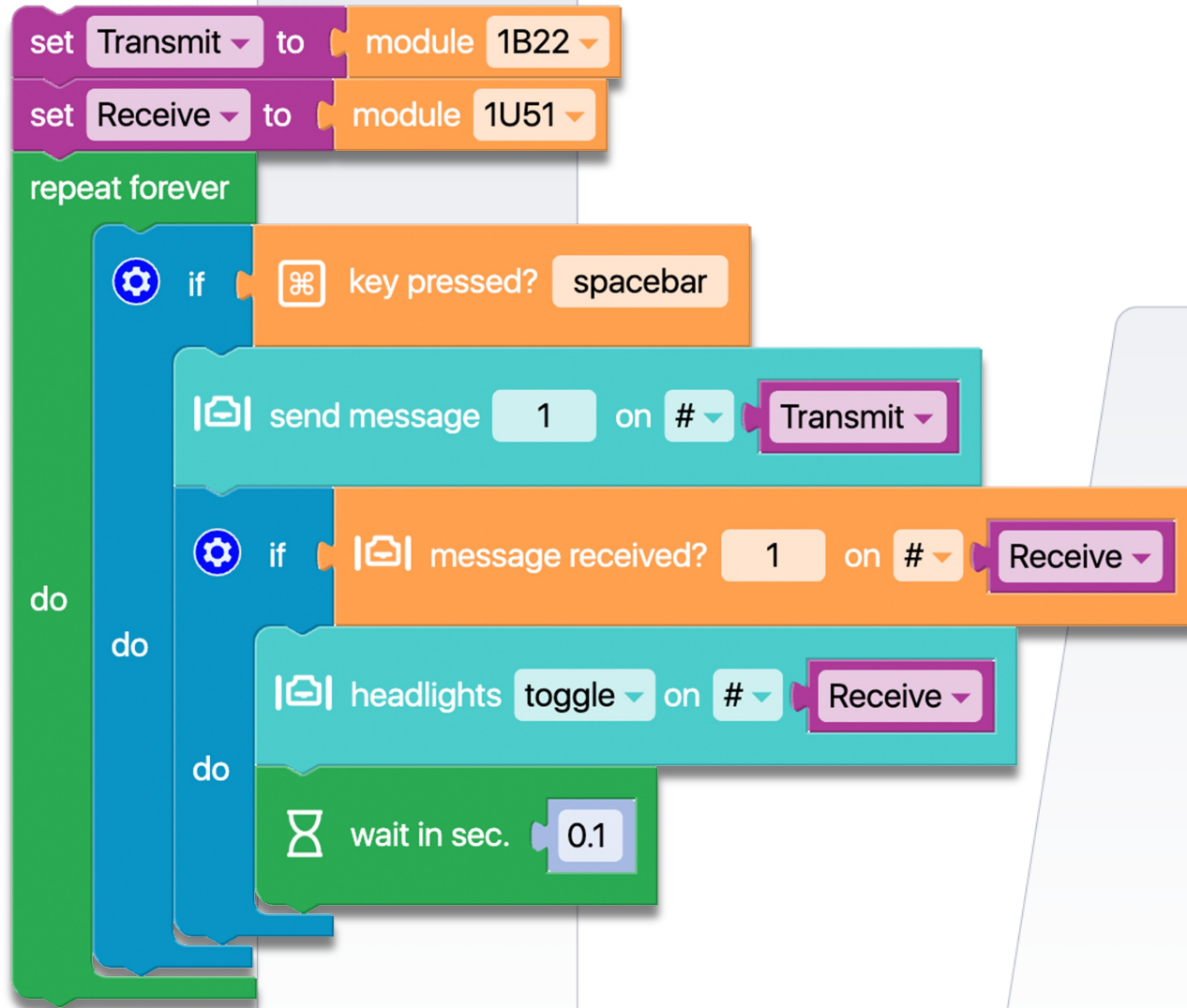

```
set Spin code to module BA11
repeat while true
  if check for color red on # Spin code
    play note Do (high)
    light red Hub
    drive 10 cm with speed: 50 on # Spin code
    do
      wait until has reached target on motor(s) both on # Spin code
    do
      spin 45 degrees with speed: 20 on # Spin code
      wait until has reached target on motor(s) both on # Spin code
  else
    light green Hub
    set speed A: 20 B: -20 on # Spin code
```

Kod 32: Wysyłanie/odbieranie komunikatów z Modułem Spin

- Moduły Spin są zdolne do przesyłania i odbierania komunikatów IR. W tym programie moduł nadający wykrywa naciśnięcia klawiszy (konkretnie spacji) i po aktywacji wysyła komunikat (klawisz 1) do odbierającego go modułu odbierającego.
- Moduł odbierający, po otrzymaniu komunikatu, zmieni stan świateł z wyłączonych na włączone lub odwrotnie. Ważne jest prawidłowe zidentyfikowanie, który moduł nadaje, a który odbiera, oraz użycie odpowiednich kodów identyfikujących. Zarówno klawisz wykrywający, jak i element przesyłany (klawisz 1) można dostosować do własnych potrzeb.
- Moduły Spin muszą być ustawione przodem do siebie.

OBEJRZYJ WIDEO



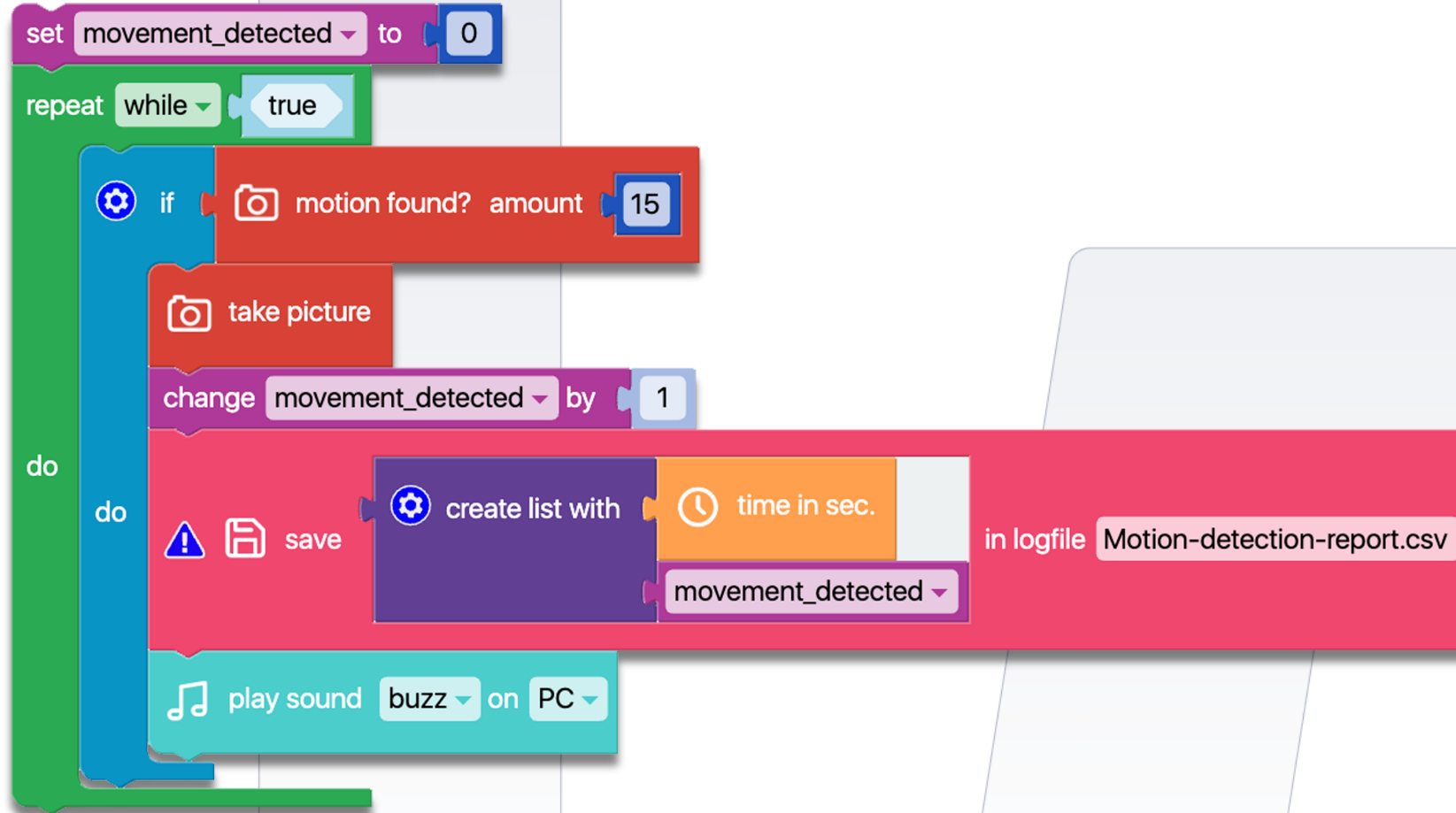


Kod 33: Reakcja kamery na ruch

- Program wykorzystuje wbudowaną kamerę komputera, na którym zainstalowano Fable Blockly. Na bieżąco analizuje obrazy przechwycone przez kamerę i uruchamia alarm dźwiękowy, gdy wykryje ruch powyżej określonego progu (ustawionego na 15).
- Dodatkowo program tworzy listę z dwoma wskaźnikami, które przechowują czas i liczbę wykryć dokonanych do tego momentu (która zwiększa się o 1 przy każdym wykryciu).

OBEJRZYJ WIDEO



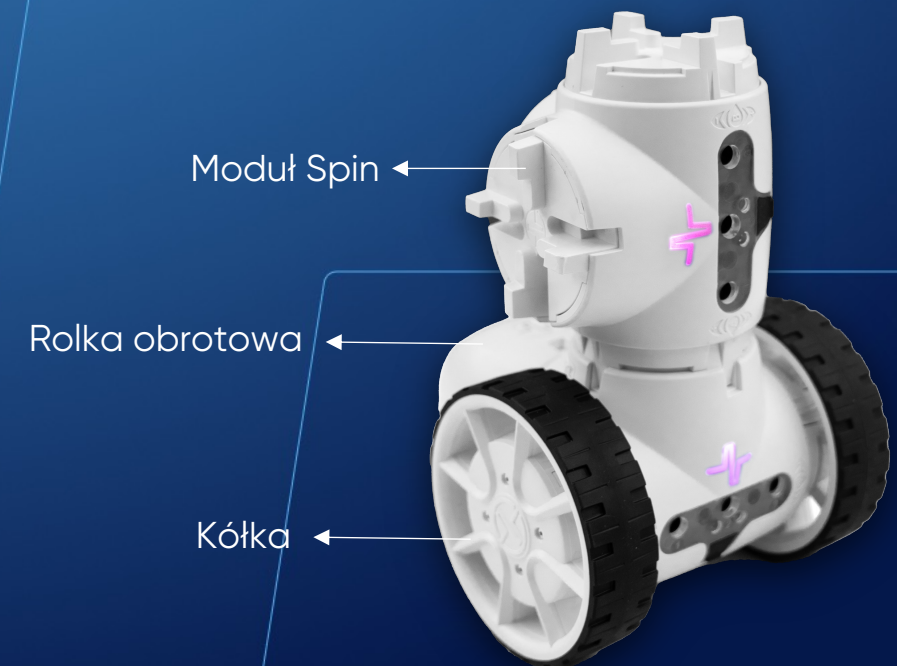


Kod 34:

Radarowy model działania dla Modułu Spin

- Program jest zaprojektowany do symulowania obrotowego ruchu systemu radarowego, oznaczonego zmienną „Radar”. Wirnik obraca się w sposób ciągły, podczas gdy sensor zbliżeniowy pozostaje aktywny. Po wykryciu obiektu w pobliżu, system radarowy uruchamia wizualny i dźwiękowy alarm, jednocześnie wyświetlając dokładny kąt wykrytego obiektu na Konsoli Wyjściowej.

OBEJRZYJ WIDEO



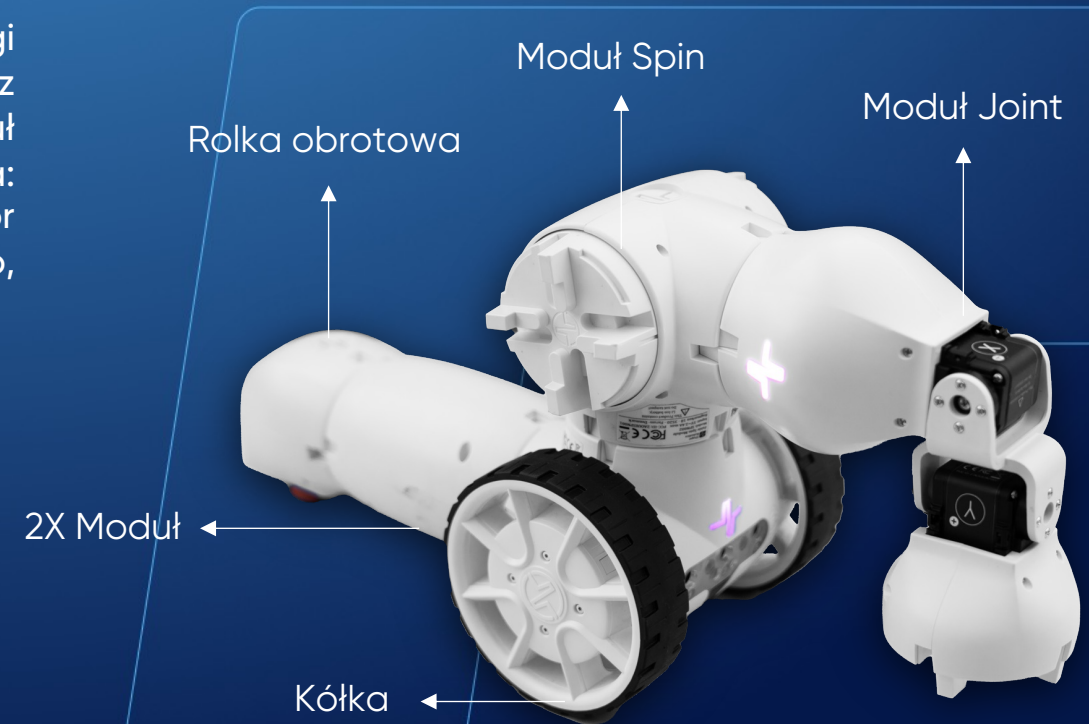
```
set Radar to module 1U4Z
repeat until obstacle within 10 % proximity on # Radar
do set speed A: 20 B: 0 on # Radar
set speed A: 0 B: 0 on # Radar
print "Detection angle: " + remainder of get angle of motor A on # Radar ÷ 360
```

Kod 35:

Maszyna sortująca kolory – Metoda 1

- Program wykorzystuje Moduł Spin do detekcji kolorów i drugi Moduł Spin do wspomagania orientacji w trybie Joint, oraz dodatkowe akcesoria do utrzymania równowagi. Pierwszy Moduł Spin identyfikuje kolory i rozpoczyna procedurę sortowania: objekty są kierowane w prawo, jeśli wykryty zostanie kolor czerwony, i w lewo, jeśli wykryty zostanie kolor żółty. Dodatkowo, Hub dostosowuje swoje światło na podstawie wykrytego koloru.

OBEJRZYJ WIDEO




```
move to X: angle 90° Y: angle 0° with speed: 50 on 1K9Q
repeat while true
  light Hub
  if check for color red on 1B22
  do Red detect
  else if check for color green on 1B22
  do Blue detect
```

```
to Red detect
  light Hub
  move to X: angle -90° Y: angle -90° with speed: 20 on 1K9Q
  wait in sec. 3
  move to X: angle -90° Y: angle 90° with speed: 20 on 1K9Q
  wait in sec. 2
  move to X: angle 90° Y: angle 90° with speed: 20 on 1K9Q
  wait in sec. 3
```

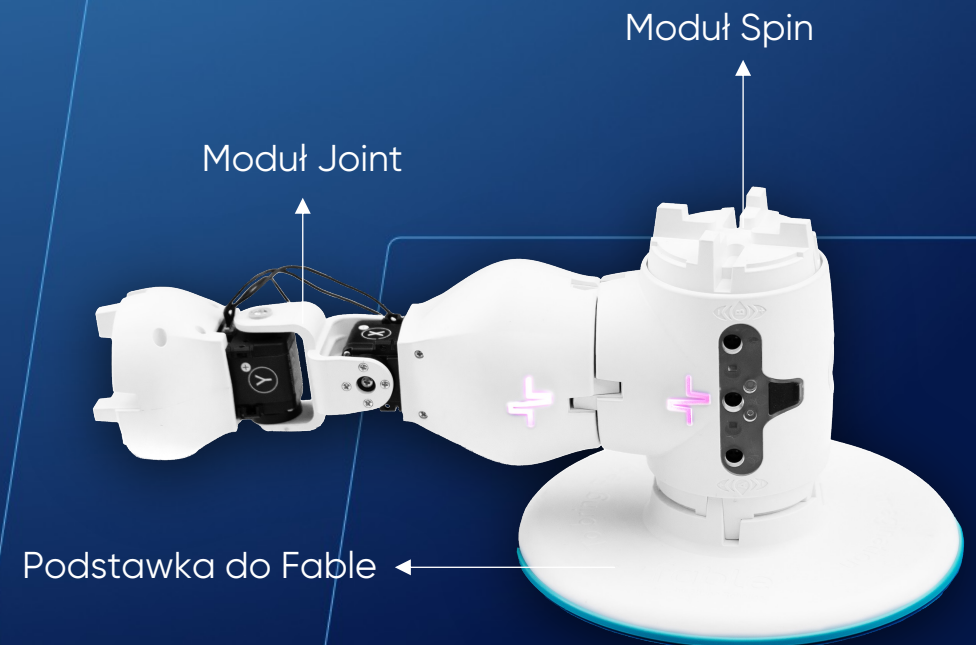
```
to Blue detect
  light Hub
  move to X: angle -90° Y: angle 90° with speed: 20 on 1K9Q
  wait in sec. 3
  move to X: angle -90° Y: angle -90° with speed: 20 on 1K9Q
  wait in sec. 2
  move to X: angle 90° Y: angle -90° with speed: 20 on 1K9Q
  wait in sec. 3
```

Kod 36:

Maszyna sortująca kolory – Metoda 2

- Program jest zaprojektowany do sortowania obiektów na podstawie dwóch kolorów: czerwonego i niebieskiego. Jest to realizowane przy użyciu dłuższego ramienia umieszczonego na wysokości około 4 cm. Wykrywanie obiektów oraz obrót są wykonywane za pomocą Modułu Spin.

OBEJRZYJ WIDEO



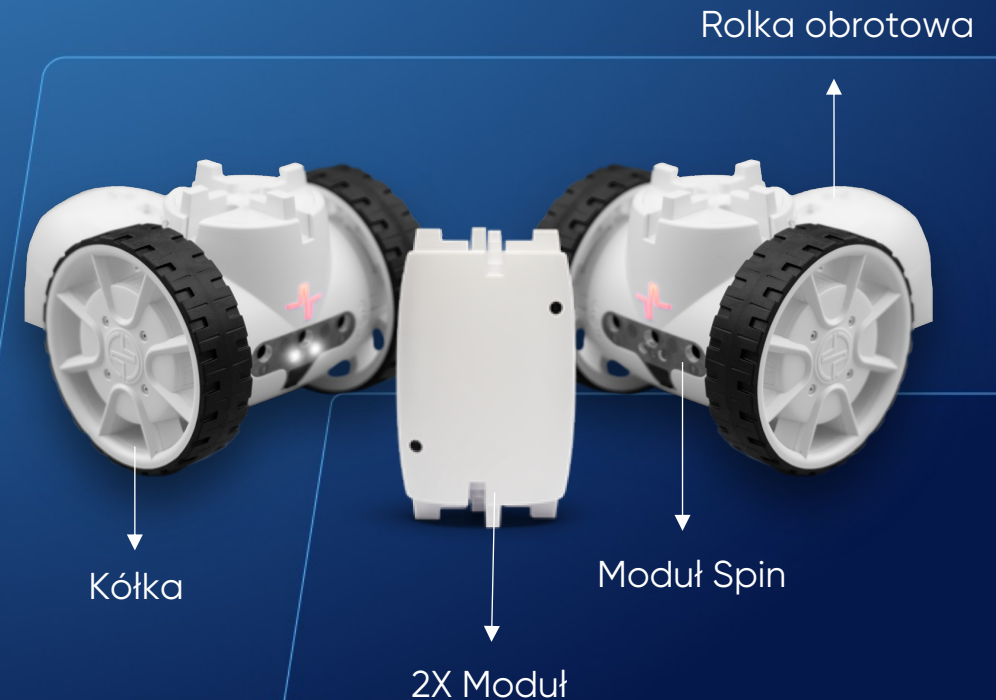
```
move to X: angle 0° Y: angle 0° on 1D2A
set Sort to module XEQ
set Color detect to module 1I11
repeat while true
  if check for color red on # Color detect
    do
      spin motor A by 360 degrees with speed: 20 on # Sort
      spin motor B by -360 degrees with speed: 20 on # Sort
    do
      else if check for color blue on # Color detect
        do
          spin motor A by -360 degrees with speed: 20 on # Sort
          spin motor B by 360 degrees with speed: 20 on # Sort
```

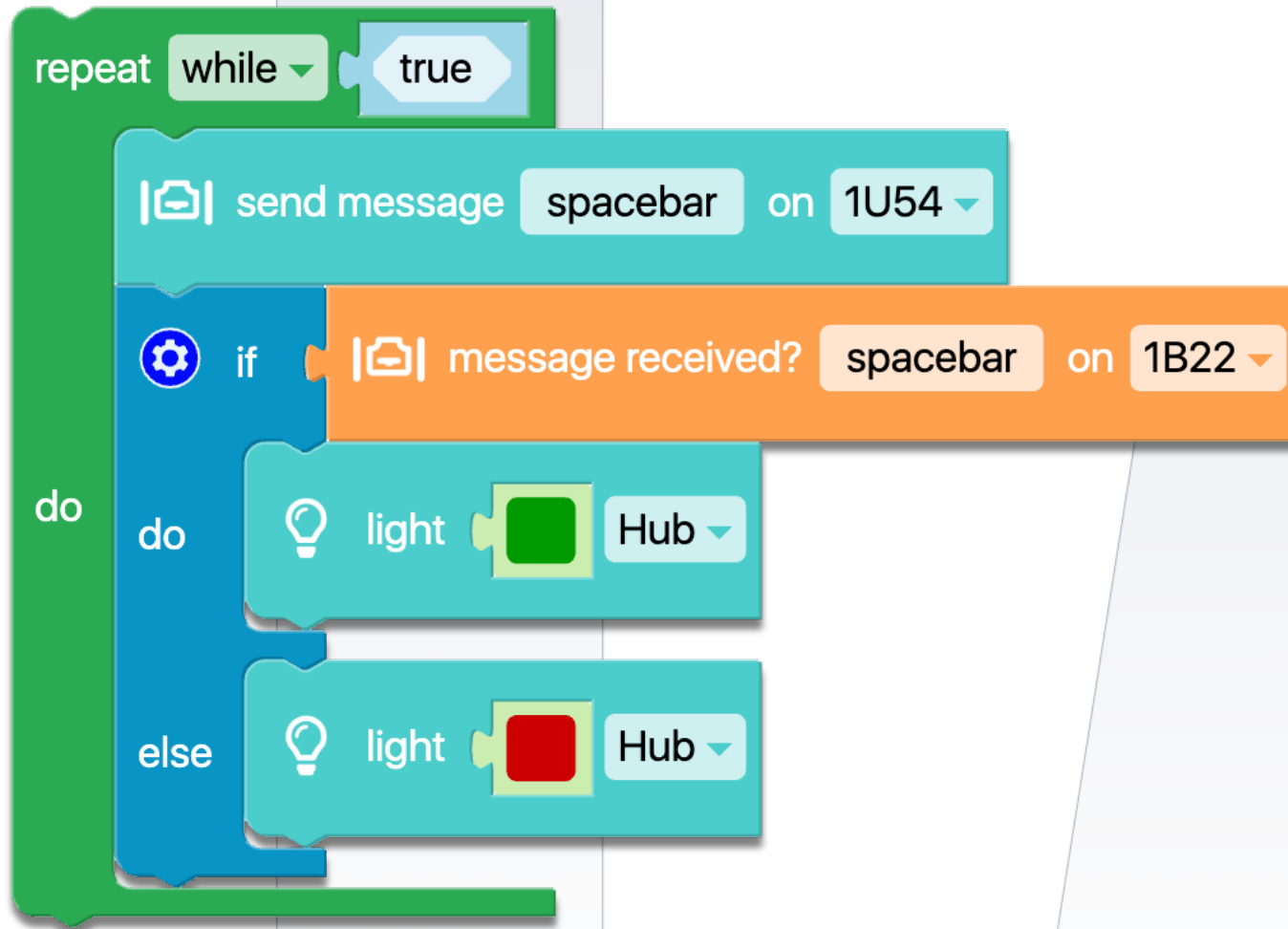
Kod 37:

Wykrywanie podczerwieni

- Dwa Moduły Spin są ustawione naprzeciwko siebie. Jeden z nich emituje ciągły sygnał przez kanał podczerwieni. Dopóki drugi Moduł Spin odbiera ten sygnał, Hub pozostaje podświetlony na zielono. Jednak jeśli przeszkoda (taka jak akcesorium Fable, inny robot, piłka lub ręka) przerwie przepływ podczerwieni, Hub zmieni kolor na czerwony. Program ten może być wykorzystywany do automatycznego wykrywania piłki podczas meczu piłki nożnej.

OBEJRZYJ WIDEO

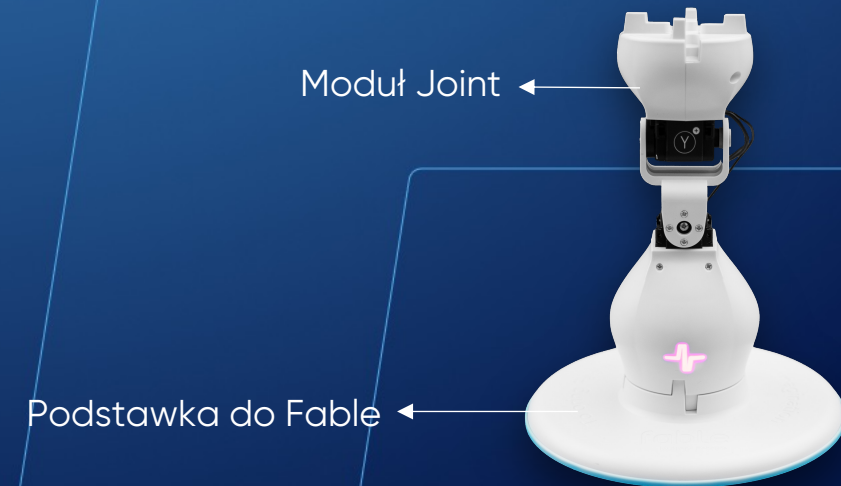


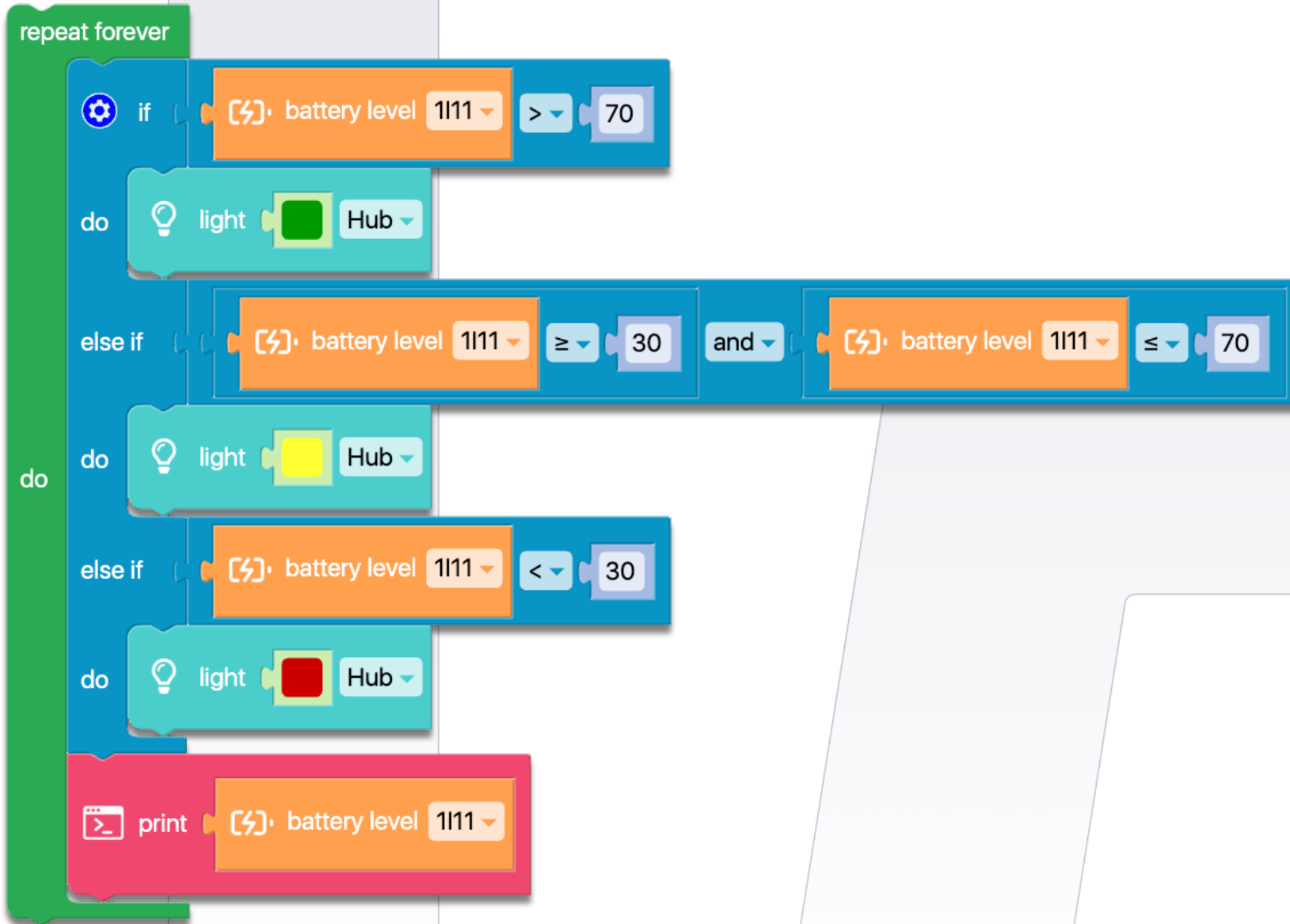


Kod 38: Monitorowanie poziomu baterii

- Program stale monitoruje poziom naładowania baterii Modułu Joint lub Modułu Spin i podświetla Hub kolorami odpowiadającymi poziomowi naładowania. Gdy poziom naładowania baterii wynosi od 70 do 100 procent, wyświetlany jest kolor zielony. Dla zakresu od 30 do 70 procent, wyświetlany jest kolor żółty, a dla zakresu od 0 do 30 procent, wyświetlany jest kolor czerwony.

OBEJRZYJ WIDEO



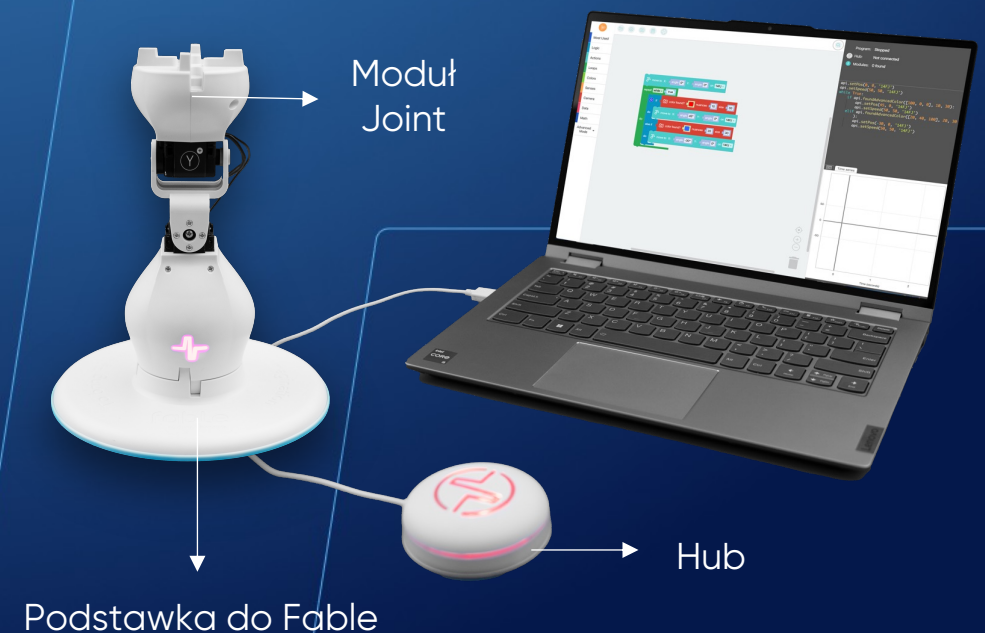


Kod 39:

Moduł Joint sterowany przez wykrywanie kolorów kamerą

- Poniższa sekwencja wykorzystuje kamerę urządzenia, na którym realizowane jest programowanie. Ustawia kolor, odcień i obszar (procentowy), który ma być wykryty. Za każdym razem, gdy wykryty zostanie określony kolor, sekwencja wydaje polecenie Modułowi Joint. Po wykryciu koloru niebieskiego, Moduł Joint przesuwa się do -30 stopni dla silnika X, a po wykryciu koloru czerwonego przesuwa się do 45 stopni dla silnika X.

OBEJRZYJ VIDEO



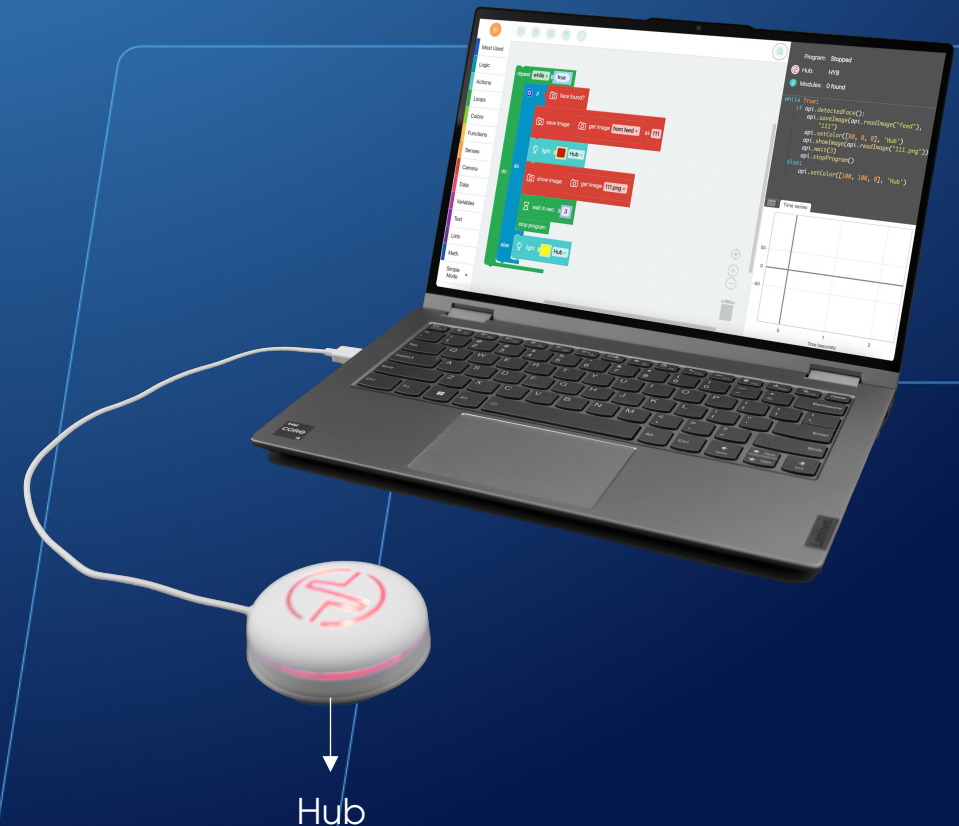

```
move to X: angle 0° Y: angle 0° on 14FJ
repeat while true
  if color found? red nuances 10 size 30
  do move to X: angle 45° Y: angle 0° on 14FJ
  do else if color found? blue nuances 20 size 30
  do move to X: angle -30° Y: angle 0° on 14FJ
```

Kod 40:

Wykrywanie twarzy za pomocą kamery

- Ta sekwencja aktywuje kamerkę urządzenia, na którym jest uruchomiona, i na bieżąco monitoruje obrazy przechwycone w poszukiwaniu obecności ludzkiej twarzy. Po wykryciu twarzy, Hub zmienia kolor z żółtego na czerwony. Jednocześnie program wykonuje zrzut ekranu z wykrycia i wyświetla go przez trzy sekundy, po czym całkowicie się wyłącza.

OBEJRZYJ WIDEO



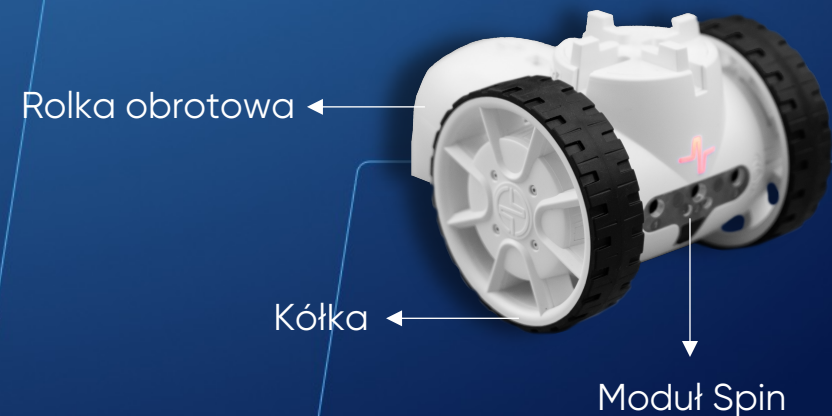
```
repeat while true  
  if face found?  
    save image get image from feed as 111  
    light Hub red  
  do  
    show image get image from feed  
    wait in sec. 3  
  stop program  
else  
  light Hub yellow
```

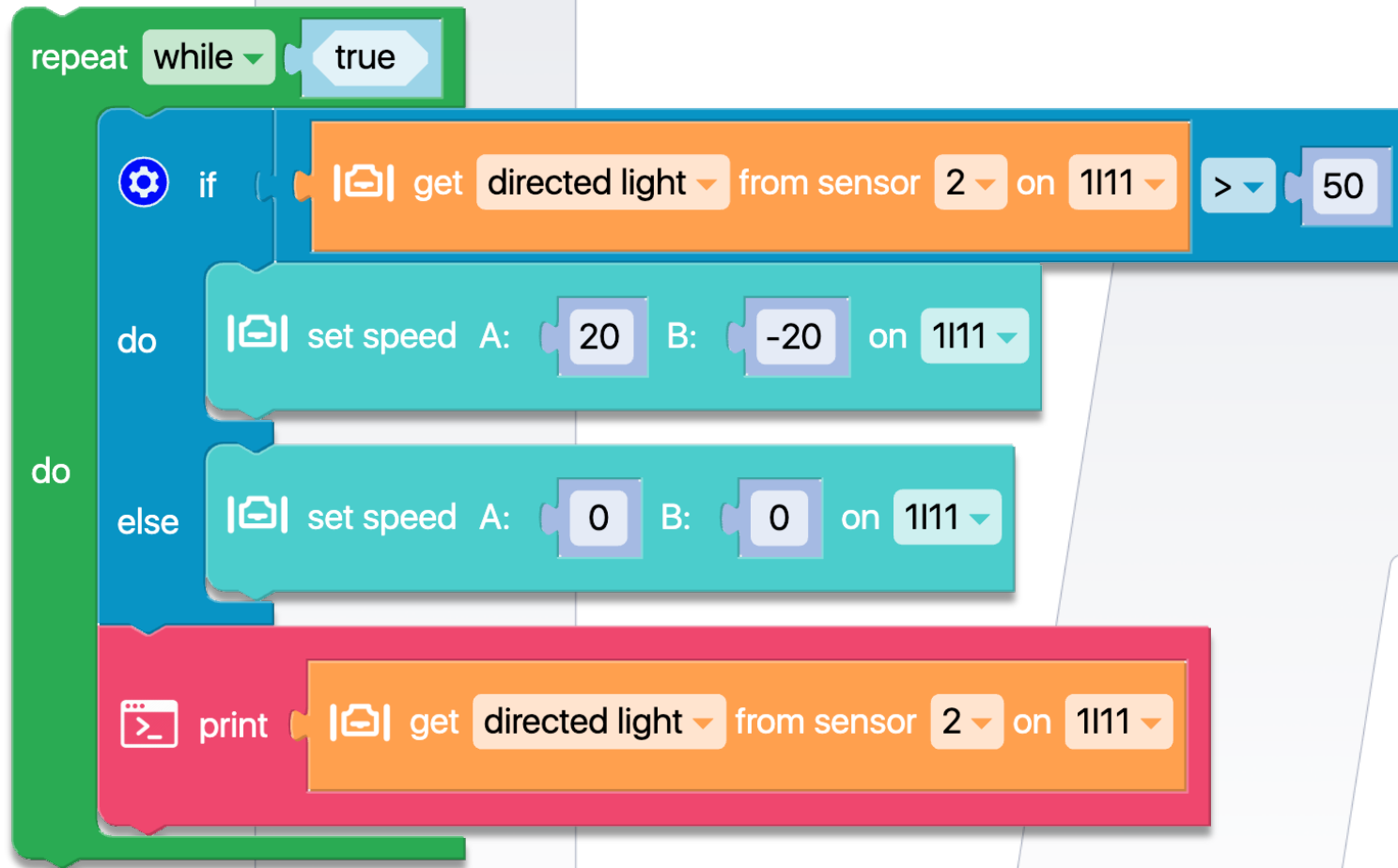
Kod 41:

Reakcja na zmiany natężenia światła

- Program jest przeznaczony do utrzymywania Modułu Spin w obszarze o zredukowanej ekspozycji bezpośredniego światła na sensory. Gdy ilość światła przekroczy ustaloną w kodzie wartość, moduł odwraca swój ruch, dopóki nie wróci do obszaru o oświetleniu poniżej wartości progowej. Ta sekwencja może być zastosowana w różnych projektach, na przykład w projekcie, gdzie Moduł Spin jest używany do obsługi zacienienia dla roślin, przemieszczając je w zależności od potrzeb.

OBEJRZYJ WIDEO



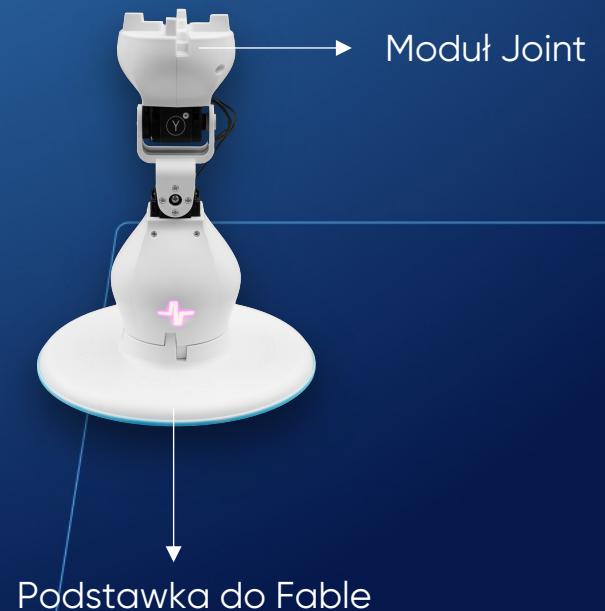


Kod 42:

Wykrywanie kątów pozycji Modułu Joint

- Sekwencja kodu jest przydatna w sytuacjach, gdy potrzebujemy ustawić serwomotory Modułu Joint pod kątem, który jest wykrywalny, ale nieznan z góry. Ta sekwencja na bieżąco wyświetla kąty serwowatorów.

OBEJRZYJ WIDEO



```
repeat while true  
do Test angle
```

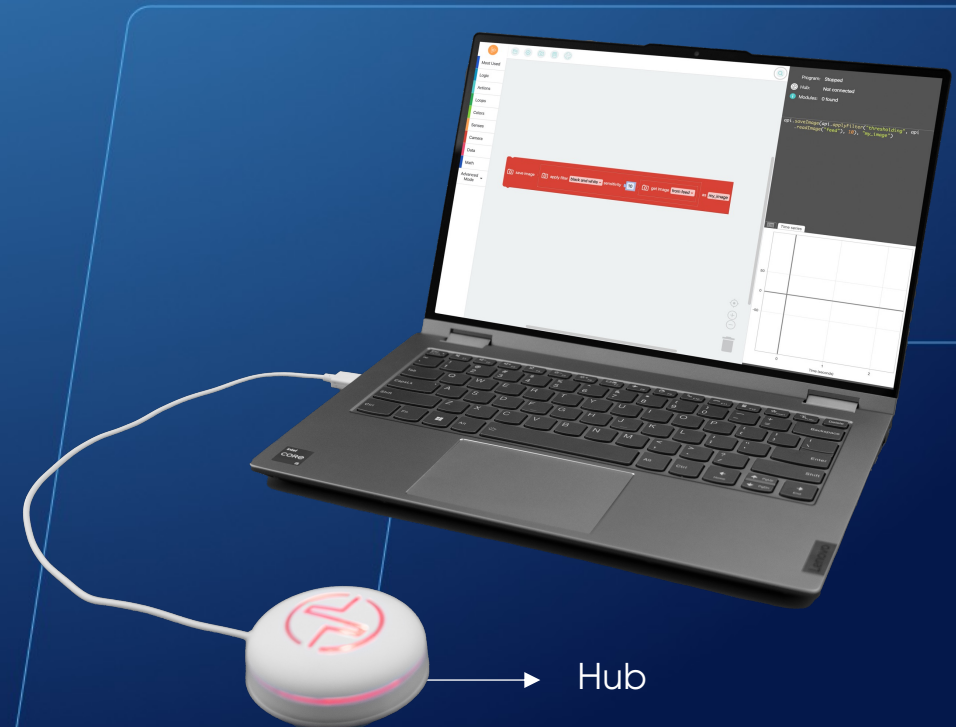
```
to Test angle  
  print "Angle for motor X is: " + round get angle of servo X on Y57  
  print "Angle for motor Y is: " + round get angle of servo Y on Y57  
  print "....."  
  wait in sec. 1
```

Kod 43:

Zastosowanie filtra na zrzucie ekranu z kamery

- Sekwencja łączy trzy polecenia: Zapisz obraz, Zastosuj filtr i Pobierz obraz z kamery. Razem przechwytyją one obraz z kamery internetowej urządzenia, na którym wykonywane jest programowanie, zapisując zdjęcie jako 'my_image'. Rodzaj zastosowanego filtra, czułość (dla niektórych filtrów) oraz nazwa przechwyconego obrazu mogą być dostosowane.

OBEJRZYJ WIDEO



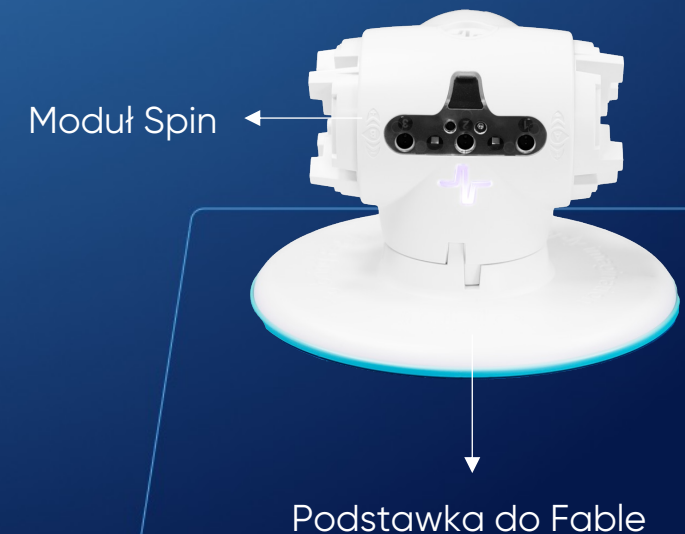
save image apply filter black and white sensitivity 10 get image from feed as my_image

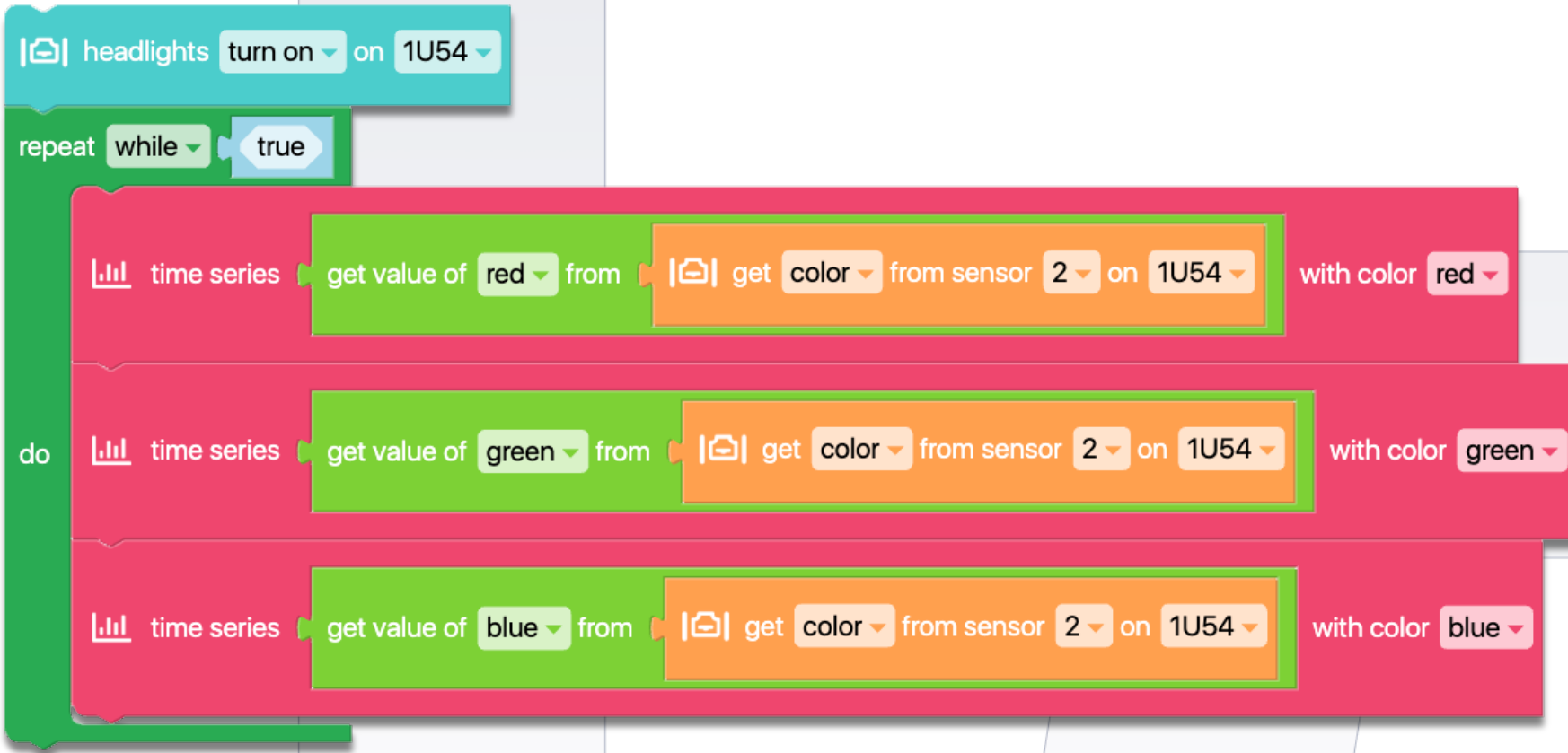
Kod 44:

Wyodrębnianie wartości RGB

- Sekwencja kodu "wyodrębnia" wartości R, G, B z koloru wykrytego przez Moduł Spin. Kolor jest analizowany, a następnie wartości R, G, B są wyświetlane na wykresach.

OBEJRZYJ WIDEO



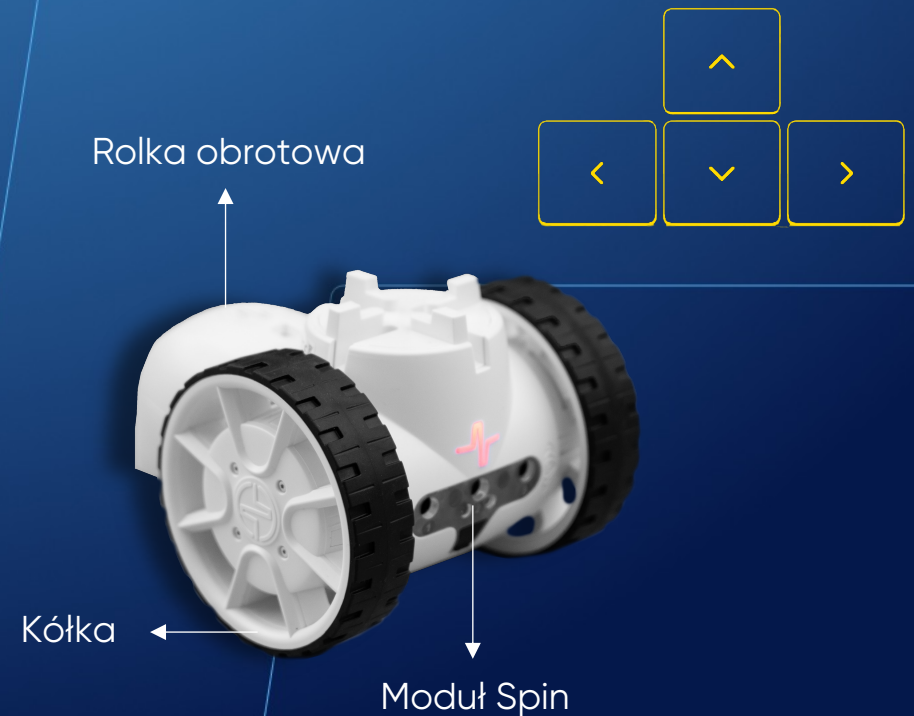


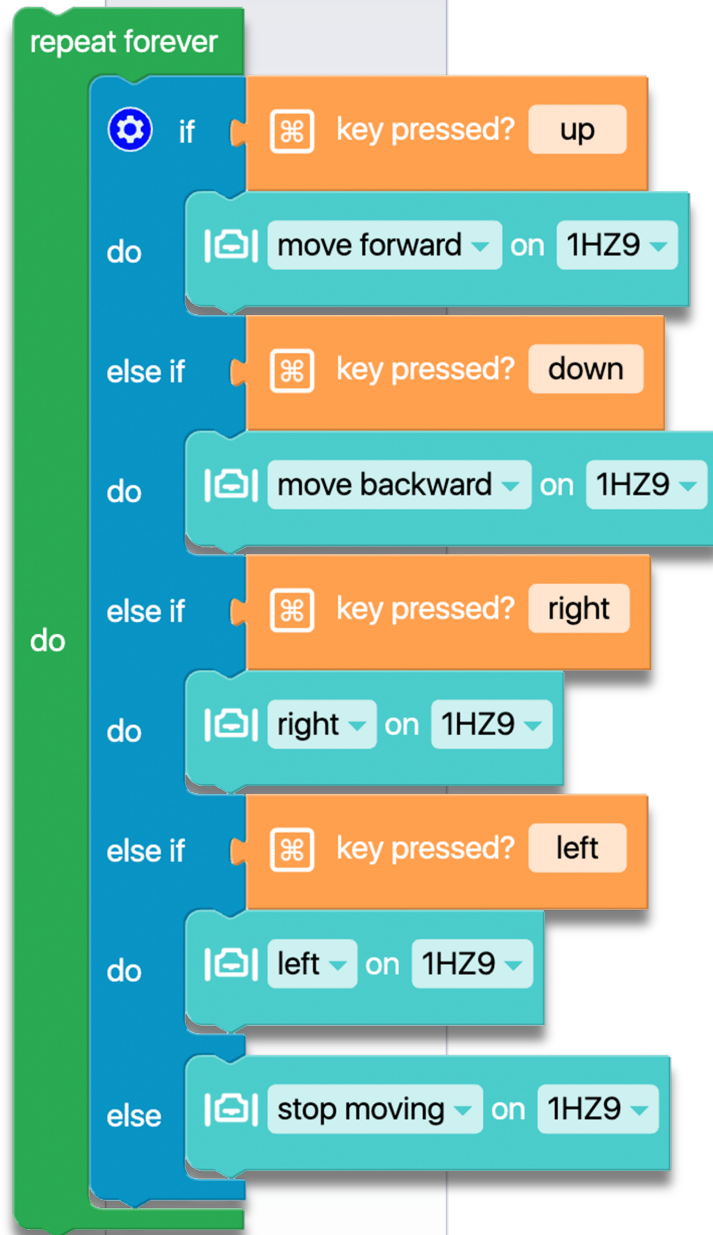
Kod 45:

Sterowanie Modułem Spin za pomocą klawiszy

- Program jest zaprojektowany do sterowania Modułem Spin przy użyciu klawiszy. Program działa w trybie ciągłej pętli, co pozwala na ciągłe monitorowanie wciśniętych klawiszy. Jeśli żaden klawisz nie jest wciśnięty, Moduł Spin zostanie zatrzymany, ponieważ jego poleceniem będzie "Stop".

OBEJRZYJ WIDEO



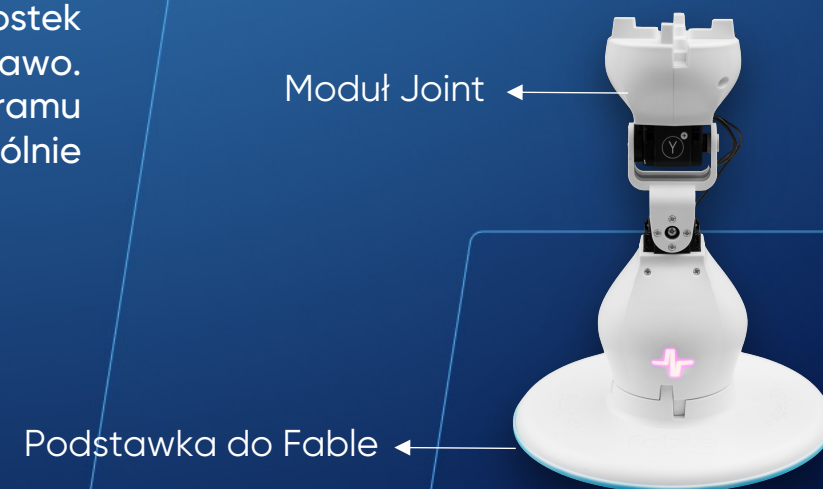


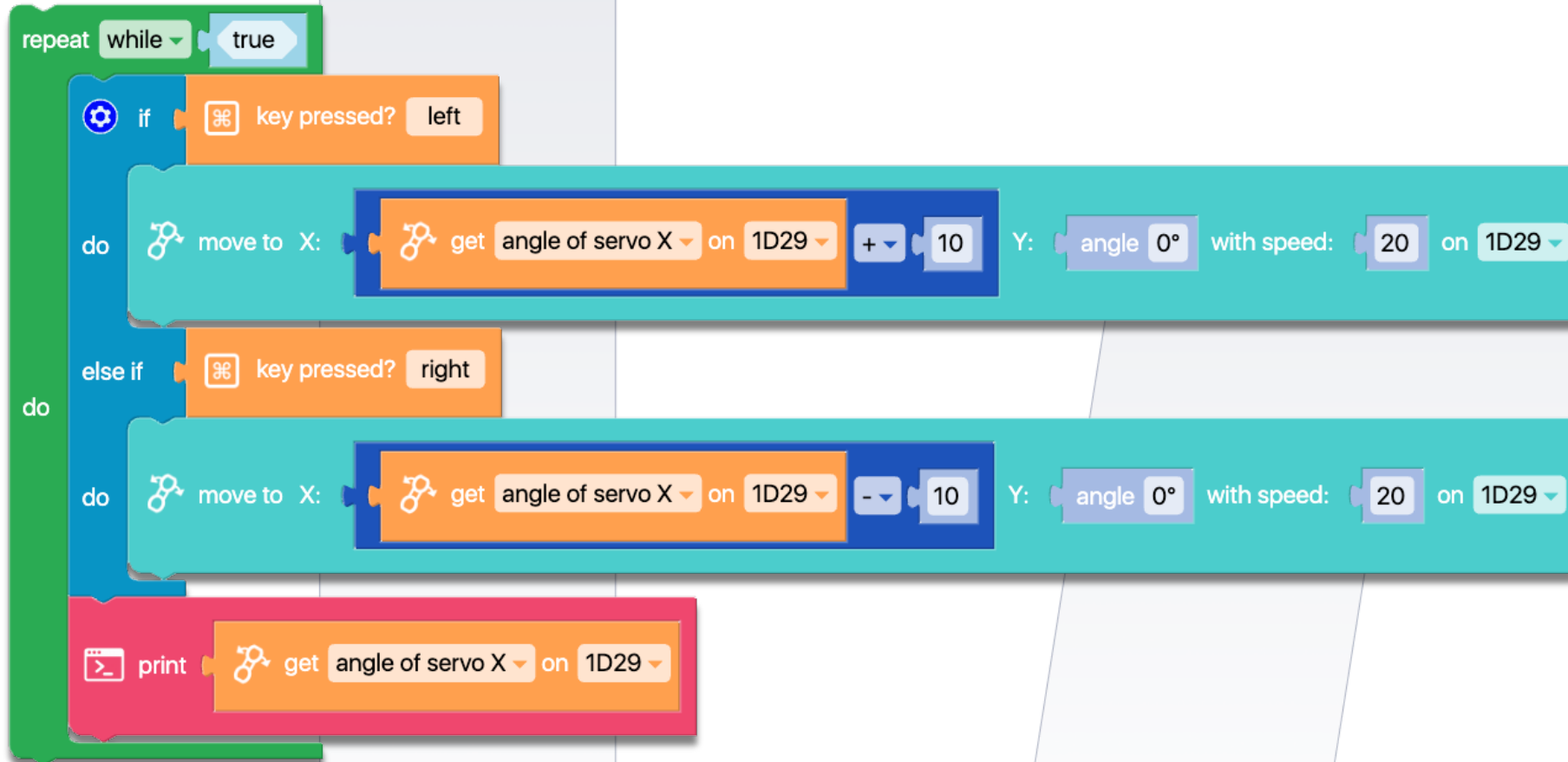
Kod 46:

Sterowanie Modułem Joint za pomocą klawiszy

- Moduł Joint jest umieszczony pionowo, przy czym oba silniki są ustawione pod kątem 0 stopni. Silnik Y będzie utrzymywał ten kąt, natomiast kąt silnika X może być dostosowany o +10 lub -10 jednostek poprzez naciśnięcie odpowiednio klawisza strzałki w lewo lub prawo. Wartość kąta jest wyświetlana w konsoli. Ta sekwencja programu umożliwia bardziej precyzyjną kontrolę Modułu Joint, co jest szczególnie przydatne przy obsłudze narzędzia podłączonego do niego.

OBEJRZYJ WIDEO

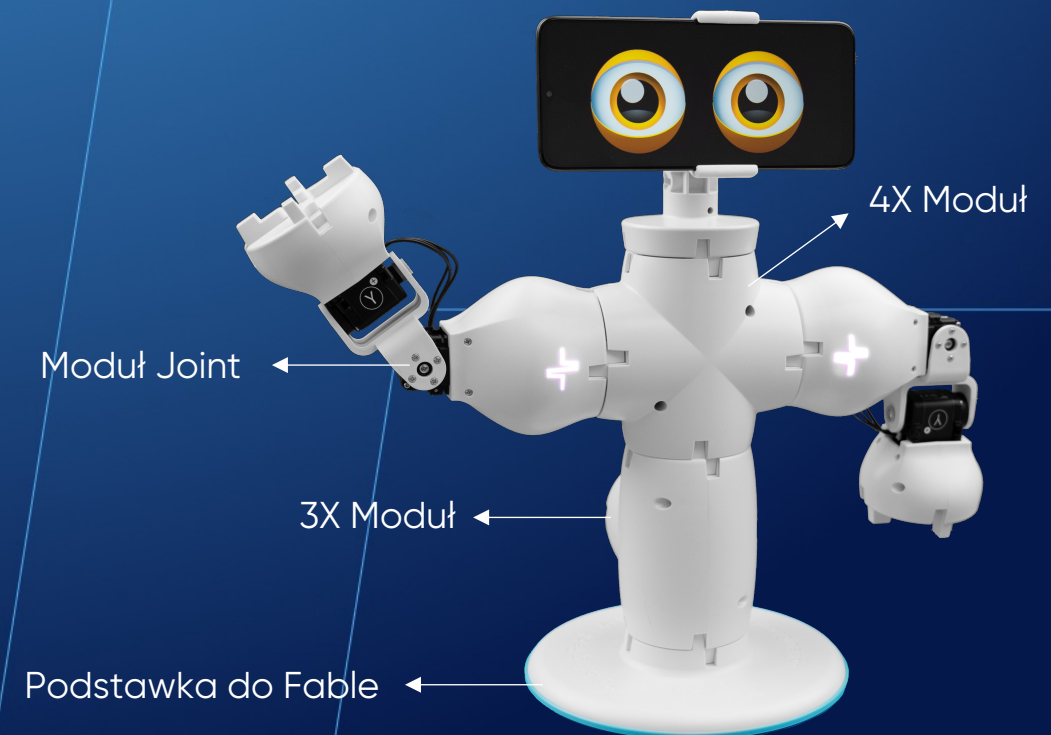




Kod 47: Fable Hello

- Program składa się z dwóch funkcji: „do_gymnastic” („gimnastykuj się”) i „wave_hello” („pomachaj na przywitanie”). Te funkcje są aktywowane przez naciśnięcie klawisza strzałki w górę („do_gymnastic”) i klawisza strzałki w dół („wave_hello”). Ruchy w tych funkcjach są już zdefiniowane. Dodatkowo zdefiniowane są zmienne, dzięki czemu roboty mogą być sprawnie zamieniane. Oznacza to, że jeśli masz nowego robota, wszystko, co musisz zrobić, to wpisać prawidłowy kod w zmiennej na początku programu.

OBEJRZYJ WIDEO




```
set LeftArm to module EZE
set RightArm to module LUB
repeat while true
  if key pressed? up
  do do_gymnastics
  else if key pressed? down
  do wave_hello
```

```
to wave_hello
  move to X: angle 0° Y: angle 90° on # RightArm
  wait in sec. 1
  move to X: angle 0° Y: angle 0° on # RightArm
  wait in sec. 1
```

```
to do_gymnastics
  wait in sec. 1
  move to X: angle 0° Y: angle 0° on # RightArm
  move to X: angle 0° Y: angle 0° on # LeftArm
  wait in sec. 1
  move to X: angle 90° Y: angle 90° on # RightArm
  move to X: angle -90° Y: angle 90° on # LeftArm
  wait in sec. 1
  move to X: angle 0° Y: angle 90° on # RightArm
  move to X: angle 0° Y: angle 90° on # LeftArm
  wait in sec. 1
```

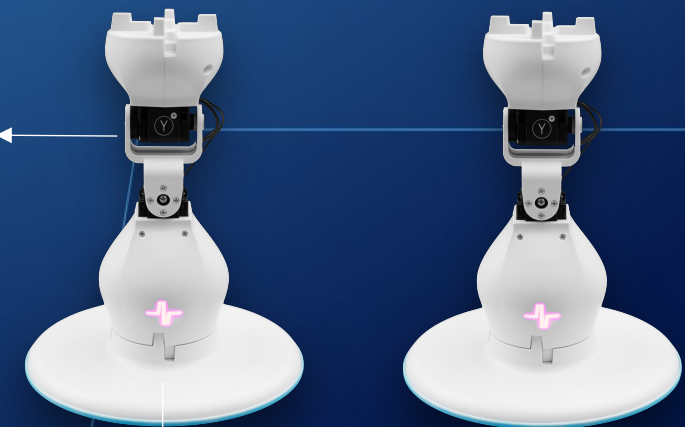
Kod 48: Podążanie za Liderem

- Program przypisuje role dwóm robotom - Leader (Przywódca) i Follower (Podążający) - za pomocą dwóch zmiennych. W trakcie działania program odczytuje kąty silników X i Y Roboty-Przywódcy i wykorzystuje je do sterowania ruchem Roboty-Podążającego. Innymi słowy, drugi robot odzwierciedla ruchy pierwszego.

OBEJRZYJ WIDEO



Moduł Joint ←



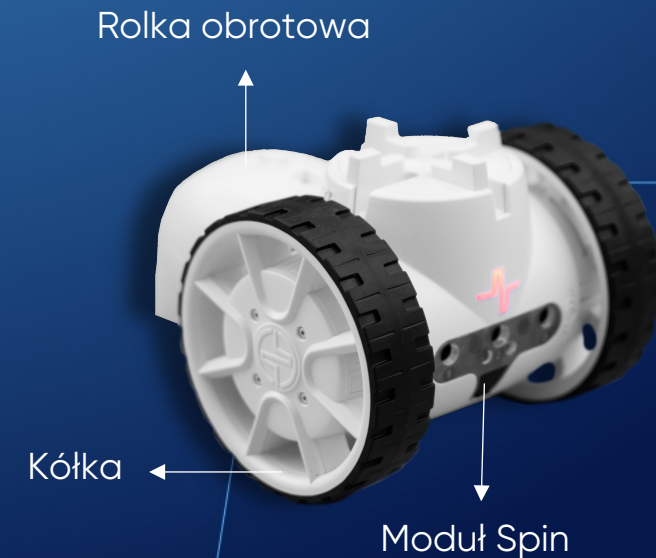
↓
Podstawka do Fable

```
set Leader to module EZE
set Follower to module LUB
repeat while true
do
  move to X:
    get angle of servo X on # Leader
  Y:
    get angle of servo Y on # Leader
  on # Follower
```

Kod 49: W Stronę Światła

- Program wykorzystuje dwa sensory światła do pomiaru wartości rejestrowanych przez sensor 1 i sensor 3. Następnie te wartości są używane do kontrolowania prędkości silników w Module Spin. Należy zauważyć, że silnik A wykorzystuje wartość ujemną, aby umożliwić Modułowi Spin ruch do przodu.

OBEJRZYJ WIDEO

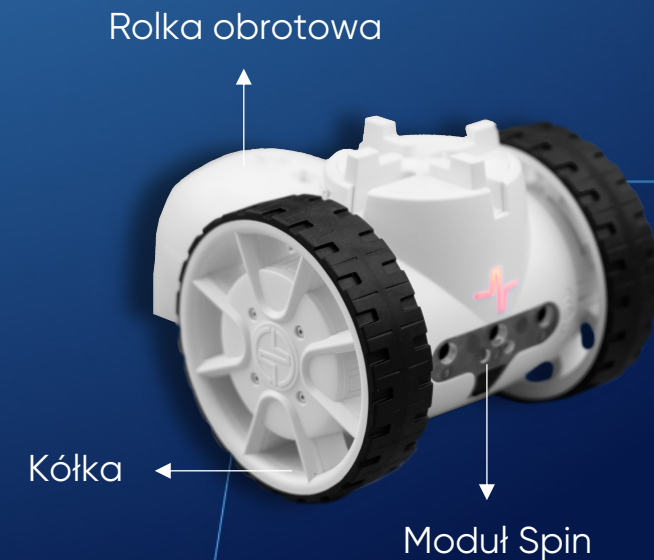


```
repeat while true  
do  
  set right to |📷| get directed light from sensor 1 on THF  
  set left to |📷| get directed light from sensor 3 on THF  
  |📷| set speed A: - B: left right on THF
```

Kod 50: Zdalne sterowanie Modułem Spin

- Program wykorzystuje Hub oraz Moduł Spin wyposażony w koła i rolkę obrotową i połączony z telefonem za pośrednictwem Fable Face. Moduł Spin otrzymuje dane o przyspieszeniu z osi telefonu i przekształca je na polecenia dla swoich silników.
- Poprzez połączenie dodatkowego akcesorium, można grać w piłkę nożną za pomocą piłki dostarczonej w zestawie, sterując robotem zdalnie za pomocą telefonu.

OBEJRZYJ WIDEO

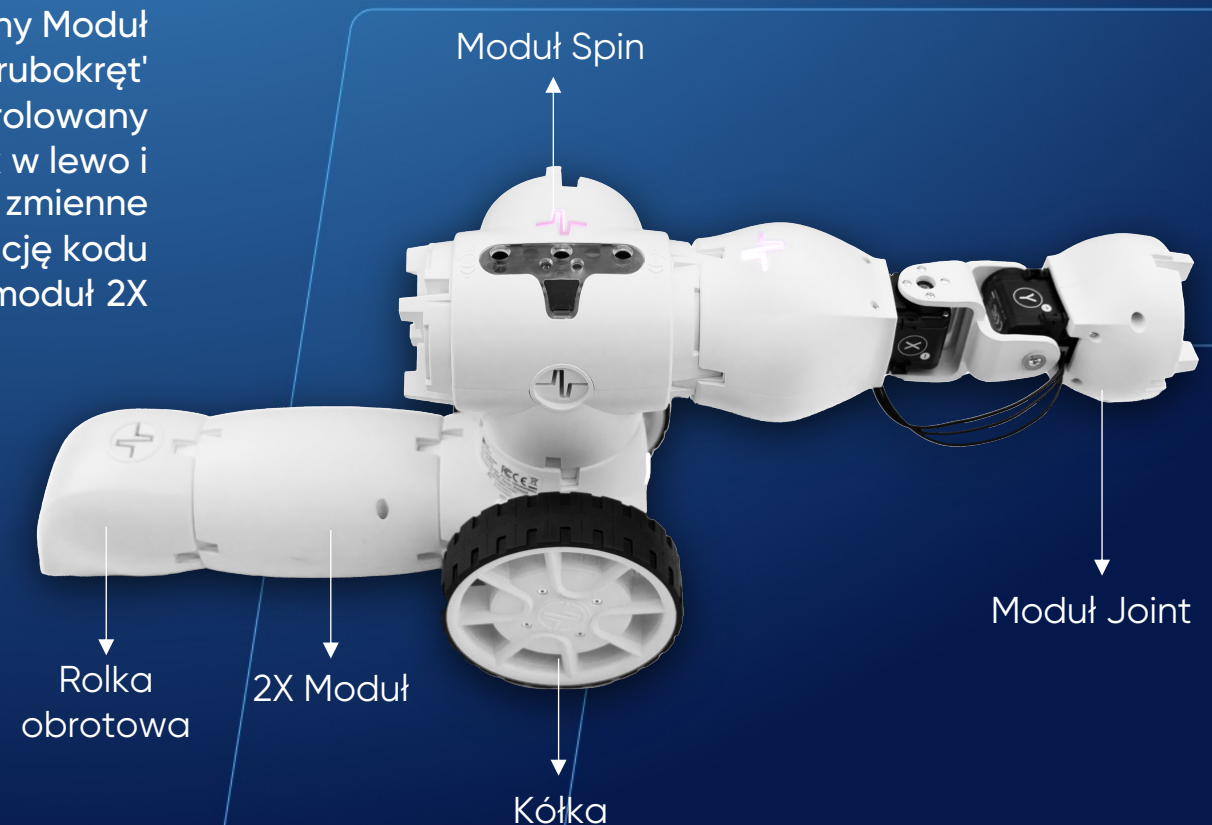


```
repeat forever
  set x_acc to get acceleration on X-axis
  set y_acc to get acceleration on Y-axis
  do
    set A to 10 x y_acc - 5 x x_acc
    set B to -10 x y_acc - 5 x x_acc
  set speed A: A B: B on IGF
```

Kod 51: Moduł Joint jako śrubokręt

- Program wykorzystuje Moduł Spin do kontroli ruchu, i kolejny Moduł Spin do regulacji obrotu 'śrubokręta', przy czym 'śrubokręt' reprezentowany jest przez Moduł Joint. Ruch jest kontrolowany przy użyciu klawiszy w, s, a i d, natomiast klawisze strzałek w lewo i w prawo sterują obrotem śrubokręta. Zadeklarowane zmienne umożliwiają łatwą wymianę robota przez prostą modyfikację kodu na początku programu. Dodatkowo, wbudowany został moduł 2X w celu zapewnienia równowagi.

OBEJRZYJ WIDEO




```
set Screwdriver to module 1D2A
set Rotate screwdriver to module 1HZ9
set Move to module 1I11
move to X: angle -90° Y: angle -90° with speed: 50 on # Screwdriver
repeat while true
  if key pressed? w
  do move forward on # Move
  else if key pressed? s
  do move backward on # Move
  else if key pressed? a
  do left on # Move
  else if key pressed? d
  do right on # Move
  else if key pressed? right
```



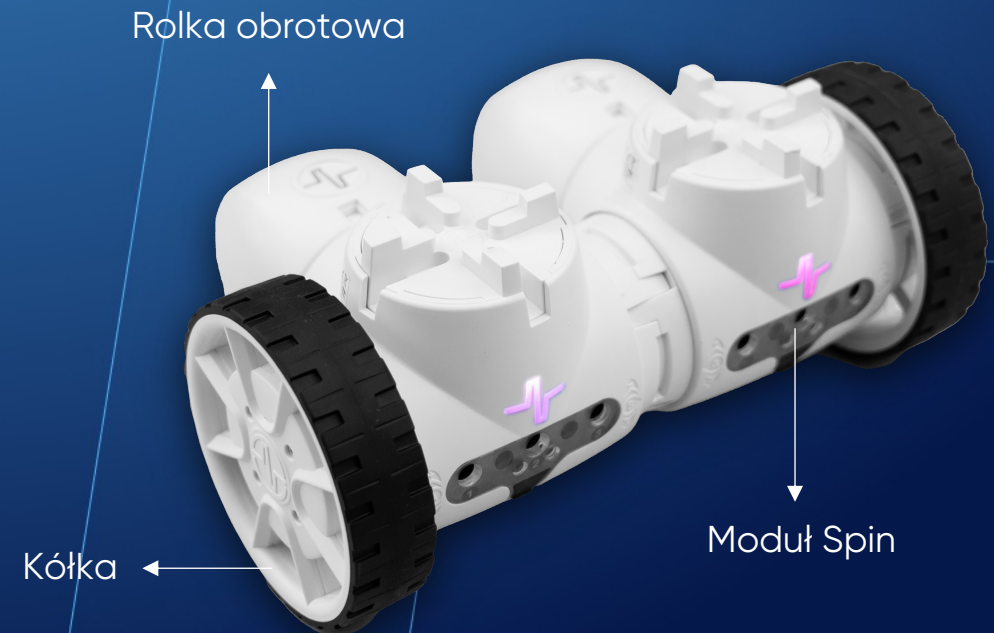
```
do
  move to X: angle 0° Y: angle 0° with speed: 20 on # Screwdriver
do
  wait in sec. 1.3
  set speed A: 0 B: 20 on # Rotate screwdriver
else if key pressed? left
  move to X: angle 0° Y: angle 0° with speed: 20 on # Screwdriver
do
  wait in sec. 1.3
  set speed A: 0 B: -20 on # Rotate screwdriver
  move to X: angle -90° Y: angle -90° with speed: 10 on # Screwdriver
else
  stop moving on # Move
  stop moving on # Rotate screwdriver
```

Kod 52:

Struktura syjamska dla Modułu Spin

- Projekt wykorzystuje dwa Moduły Spin oraz tylko dwa koła. Aby ułatwić ruch kół, dwa silniki są ze sobą połączone i utrzymywane przy zerowej prędkości. Podczas skręcania używane są jedno koło z jednego Modułu Spin i jedno koło z drugiego Modułu Spin.

OBEJRZYJ WIDEO

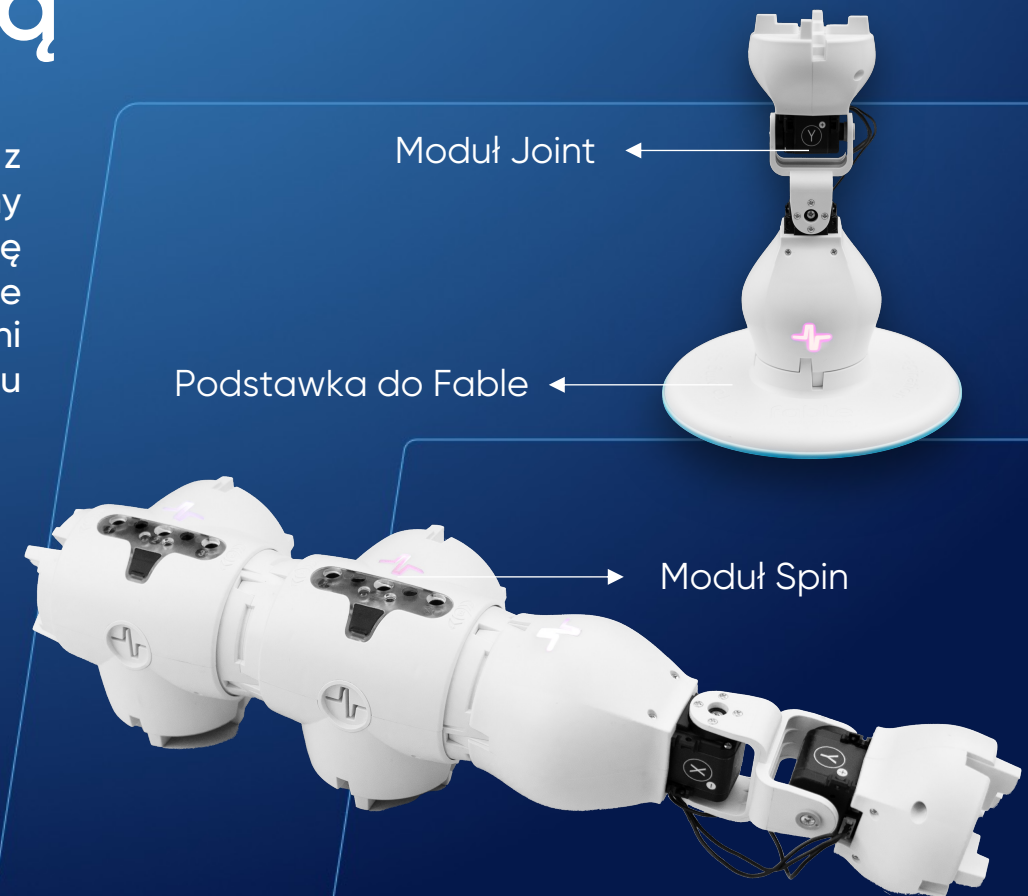


```
set Left Wheel to module 1111
set Right wheel to module 1HZ9
repeat while true
  if key pressed? up
    do
      set speed A: -20 B: 0 on # Right wheel
      set speed A: 0 B: 20 on # Left Wheel
  else if key pressed? down
    do
      set speed A: 20 B: 0 on # Right wheel
      set speed A: 0 B: -20 on # Left Wheel
  else if key pressed? left
    do
      set speed A: -20 B: 0 on # Right wheel
      set speed A: 0 B: -20 on # Left Wheel
  else if key pressed? right
    do
      set speed A: 20 B: 0 on # Right wheel
      set speed A: 0 B: 20 on # Left Wheel
  else
    do
      set speed A: 0 B: 0 on # Right wheel
      set speed A: 0 B: 0 on # Left Wheel
```

Kod 53: Zdalne sterowanie barierą

- Bariera jest kontrolowana przez Moduł Joint połączony z jednym z Modułów Spin. Kolejny Moduł Joint komunikuje kąt, pod jakim obrócony jest silnik X. Gdy kąt przekroczy 20 stopni, silnik kontrolujący barierę obraca się o 90 stopni. W przeciwnym przypadku, jeśli kąt spadnie poniżej -20 stopni, obrót jest odwrócony. Wartości 20 i -20 stopni zostały wybrane w celu zapewnienia zakresu ruchu dla zdalnego Modułu Joint. Kąt silnika X jest wyświetlany w Konsoli Wyjściowej.

OBEJRZYJ WIDEO



```
set Barrier motor to module XEQ
set Remote control to module Y57

move to X: angle 90° Y: angle 0° on 1D2A

repeat while true
  if get angle of servo X on # Remote control > 20
    spin motor B by 90 degrees with speed: 20 on # Barrier motor
    do
      wait in sec. 3
  else if get angle of servo X on # Remote control < -20
    spin motor B by -90 degrees with speed: 20 on # Barrier motor
    do
      wait in sec. 3
  print get angle of servo X on # Remote control
```

Kod 54:

Moduł Joint jako joystick dla Modułu Spin

- Program umożliwia zdalne sterowanie ruchem Modułu Spin za pomocą Modułu Joint, działając sekwencyjnie, bez mieszania ruchów. Silnik X Modułu Joint kieruje Modułem Spin do przodu lub do tyłu, podczas gdy silnik Y nakazuje mu obrócić się w lewo lub w prawo. Aby udoskonalić kontrolę nad Modułem Spin, dostępny jest zakres kątów, przy których nie jest brany pod uwagę żaden ruch. Te kąty będą wyświetlane w Konsoli Wyjściowej.

OBEJRZYJ WIDEO



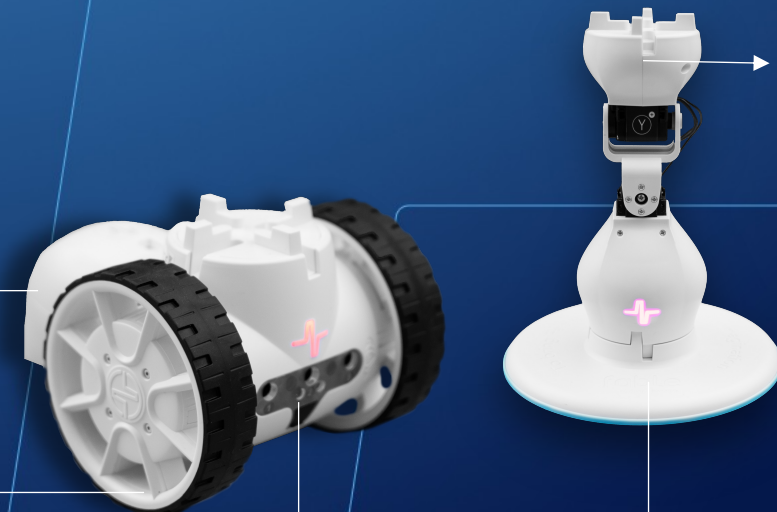
Rolka obrotowa

Kółka

Moduł Spin

Podstawka do Fable

Moduł Joint

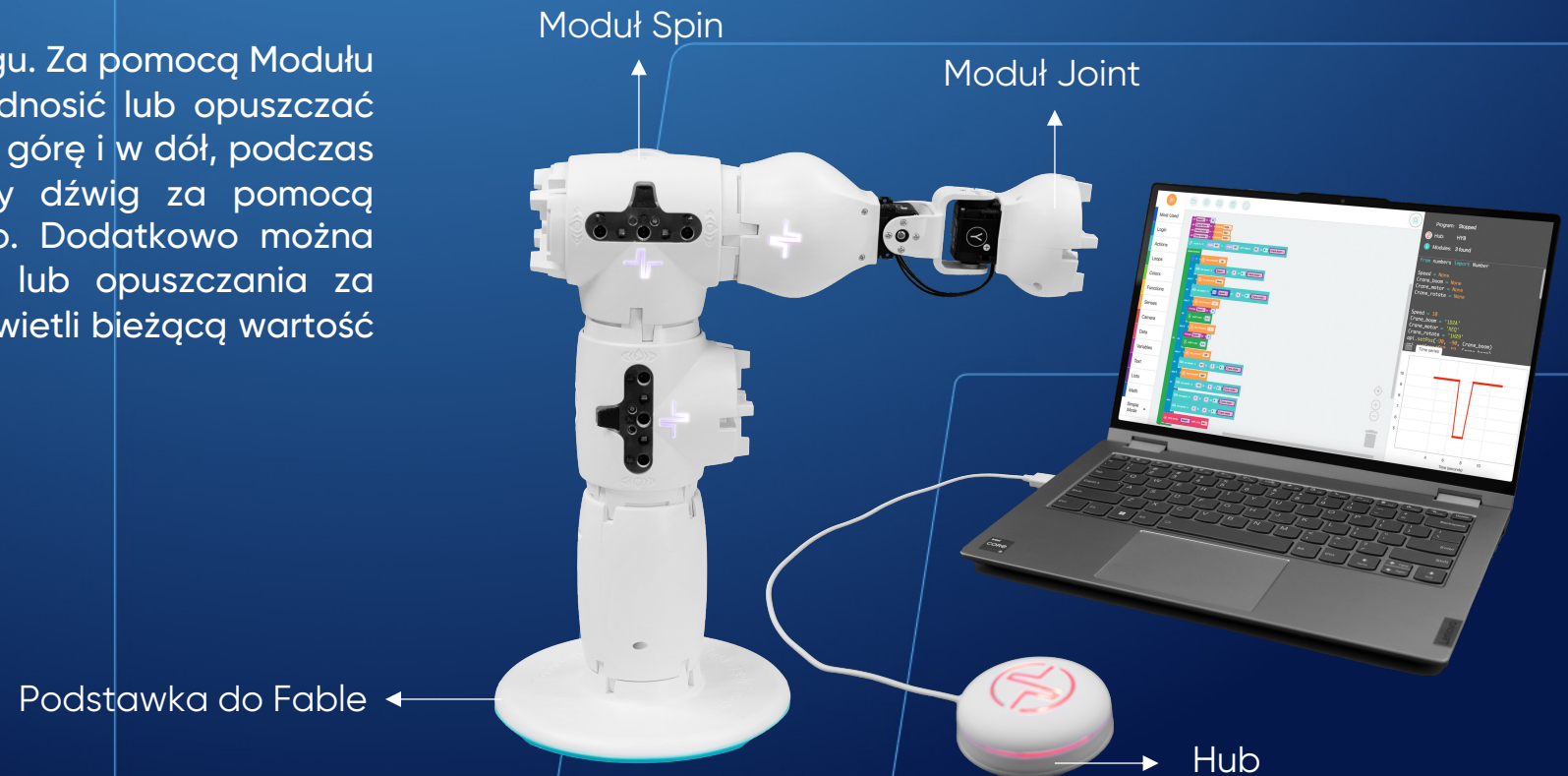


```
set Line to module 1D2A
set Rotate to module Y57
repeat forever
  if (get angle of servo X on Y57 > -90 and get angle of servo X on Y57 < -30 and get angle of servo X on Y57 < 0)
    do (set speed A: -20 B: 20 on XEQ)
  else if (get angle of servo X on Y57 > 30 and get angle of servo X on Y57 < 90 and get angle of servo X on Y57 > 0)
    do (set speed A: 20 B: -20 on XEQ)
  else if (get angle of servo Y on Y57 > 30 and get angle of servo Y on Y57 < 90 and get angle of servo Y on Y57 > 0)
    do (set speed A: 20 B: 20 on XEQ)
  else if (get angle of servo Y on Y57 > -90 and get angle of servo Y on Y57 < -30 and get angle of servo Y on Y57 < 0)
    do (set speed A: -20 B: -20 on XEQ)
  else (set speed A: 0 B: 0 on XEQ)
  print ("Angle servo X is: " + get angle of servo X on Y57)
  print ("Angle servo Y is: " + get angle of servo Y on Y57)
```

Kod 55: Symulator dźwigu

- Ten program symuluje działanie dźwigu. Za pomocą Modułu Spin można obracać wysięgnik i podnosić lub opuszczać ciężar za pomocą klawiszy strzałek w górę i w dół, podczas gdy drugi Moduł Spin obraca cały dźwig za pomocą klawiszy strzałek w lewo i w prawo. Dodatkowo można dostosować prędkość podnoszenia lub opuszczania za pomocą przycisków + i -. Wykres wyświetli bieżącą wartość prędkości w czasie rzeczywistym.

OBEJRZYJ WIDEO




```
set Speed to 10
set Crane boom to module 14FJ
set Crane motor to module 1B22
set Crane rotate to module B221

move to X: angle 0° Y: angle 0° with speed: 50 on # Crane boom

repeat forever
  if key pressed? up
  do set speed A: Speed B: 0 on # Crane motor
  else if key pressed? down
  do set speed A: - Speed B: 0 on # Crane motor
  else if key pressed? +
  change Speed by 5
  do wait in sec. 0.2
  else if key pressed? -
  change Speed by -5
```



```
do do wait in sec. 0.2
else if key pressed? left
do set speed A: 40 B: 0 on # Crane rotate
else if key pressed? right
do set speed A: -40 B: 0 on # Crane rotate
do set speed A: 0 B: 0 on # Crane motor
else
do set speed A: 0 B: 0 on # Crane rotate

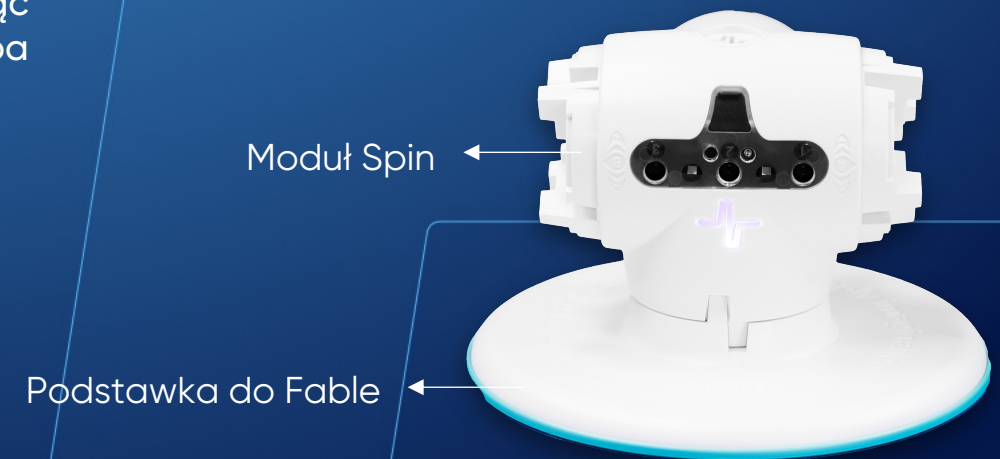
time series Speed with color red
```

Kod 56:

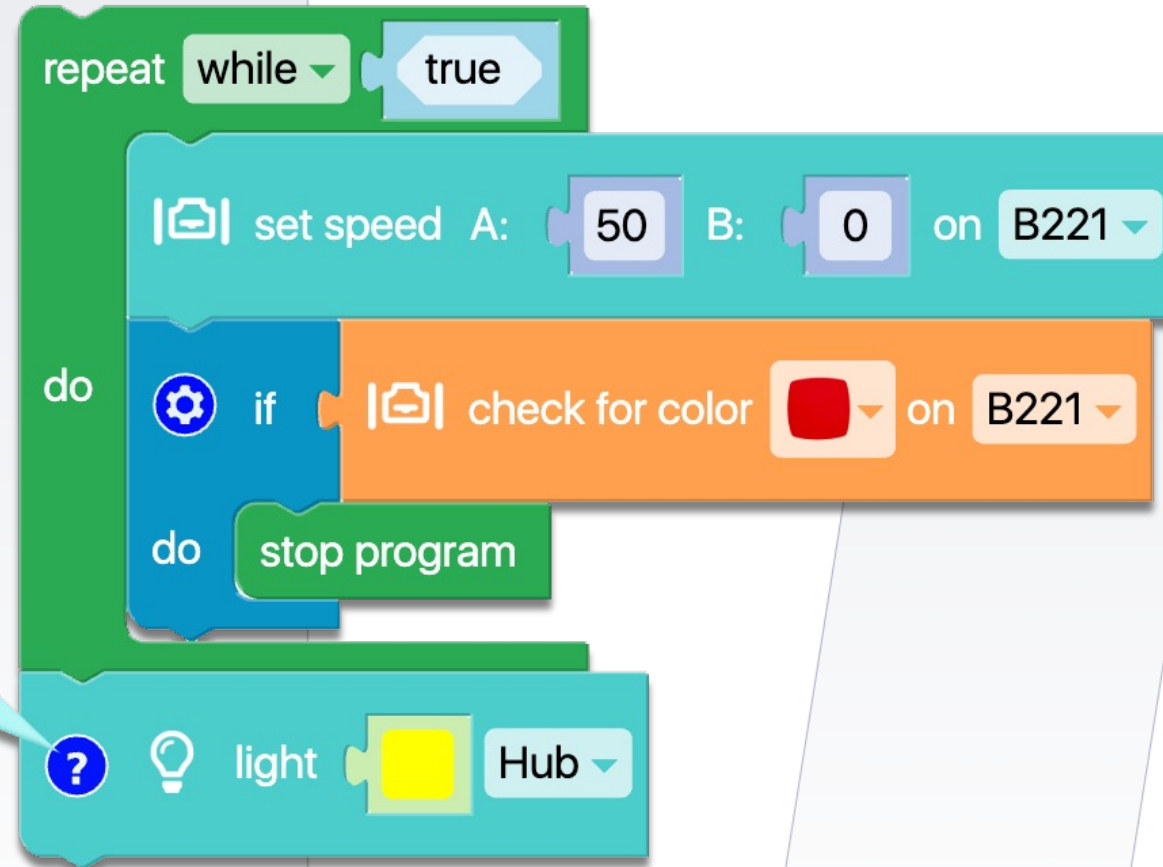
Całkowite zatrzymanie programu (STOP)

- Sekwencja programu uruchamia silnik A Modułu Spin, powodując jego ciągłe obracanie. Gdy Moduł Spin wykryje kolor czerwony przed swoimi sensorami, kod uruchamia polecenie „Stop” programu, zatrzymując cały program. W tym przypadku wszelkie kolejne polecenia Huba zostaną zignorowane i nie będą mogły zostać wykonane.

OBEJRZYJ WIDEO



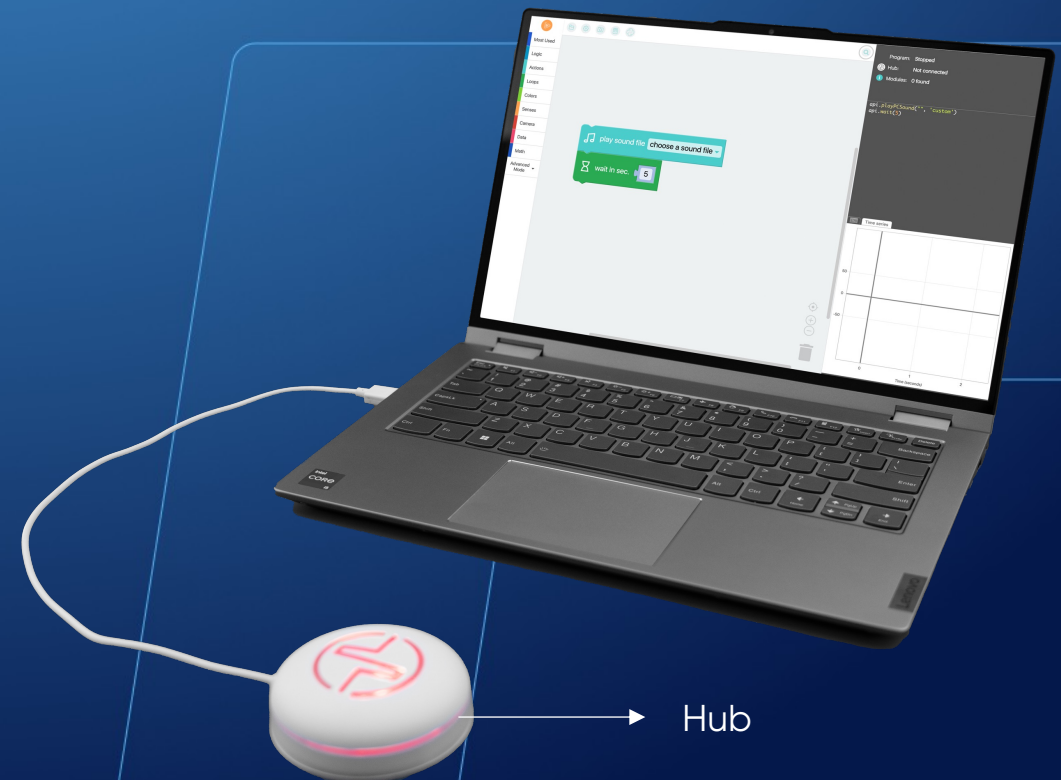
This command has no effect if red is detected!



Kod 57: Użycie pliku MP3

- Sekwencja kodu uruchamia plik MP3 z komputera. Należy zauważyć, że sekwencja odtwarza plik audio, podczas gdy program kontynuuje wykonywanie innych poleceń. Jeśli pojawi się inny dźwięk, konieczne jest użycie polecenia oczekiwania, aby upewnić się, że odtwarzanie audio zostanie zakończone przed kontynuacją. Plik musi znajdować się w Documents/Fable/My Fable Sounds.

OBEJRZYJ WIDEO



 play sound file

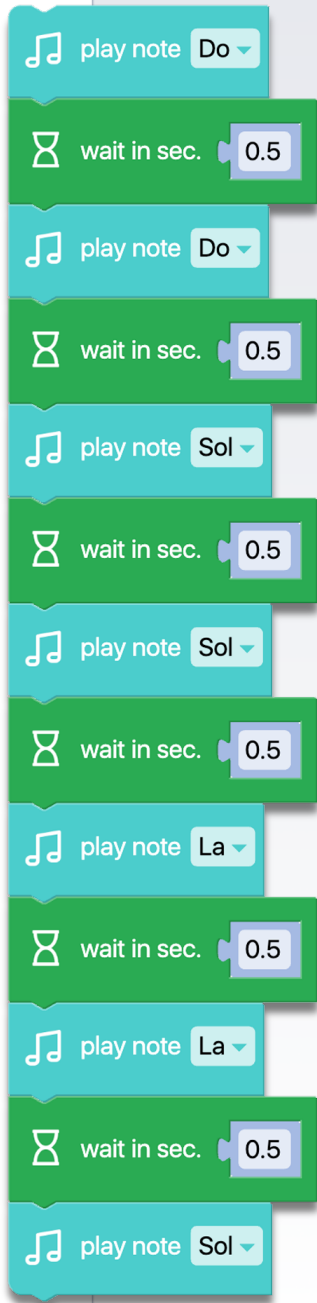
 wait in sec.

Kod 58: Zabawa z nutami

- Sekwencja kodu odtwarza pierwszą część piosenki "Twinkle, Twinkle Little Star". Rytm można zmienić, modyfikując tempo za pomocą przycisków pauzy po odtworzeniu każdej nuty.

OBEJRZYJ WIDEO





A Scratch script consisting of 13 blocks stacked vertically. The blocks alternate between playing a note and waiting for 0.5 seconds. The notes are Do, Sol, La, and Sol in that order, with two instances of each note.

- play note Do
- wait in sec. 0.5
- play note Do
- wait in sec. 0.5
- play note Sol
- wait in sec. 0.5
- play note Sol
- wait in sec. 0.5
- play note La
- wait in sec. 0.5
- play note La
- wait in sec. 0.5
- play note Sol

Kod 59:

Zakończenie pętli nieskończonej

- Zazwyczaj nie zaleca się używania nieskończonych pętli (blok „Powtarzaj w nieskończoność”) w programowaniu. Jednakże, w pewnych sytuacjach może zaistnieć potrzeba użycia takiej pętli i wyjścia z niej w odpowiednim momencie. Można to osiągnąć za pomocą komendy „Przerwij pętlę”. Po wykonaniu tego polecenia pętla zostanie zakończona, a kolejne polecenia będą nadal wykonywane w kolejności po pętli.
- Na przykład rozważmy scenariusz, gdzie silnik A Modułu Spin jest skonfigurowany tak, aby obracał się w sposób ciągły do momentu naciśnięcia klawisza spacji. Po naciśnięciu Spacji zostaje wykonane polecenie „Drukuj/wyświetl” w Konsoli Wyjściowej, co wskazuje, że pętla została przerwana.

OBEJRZYJ WIDEO

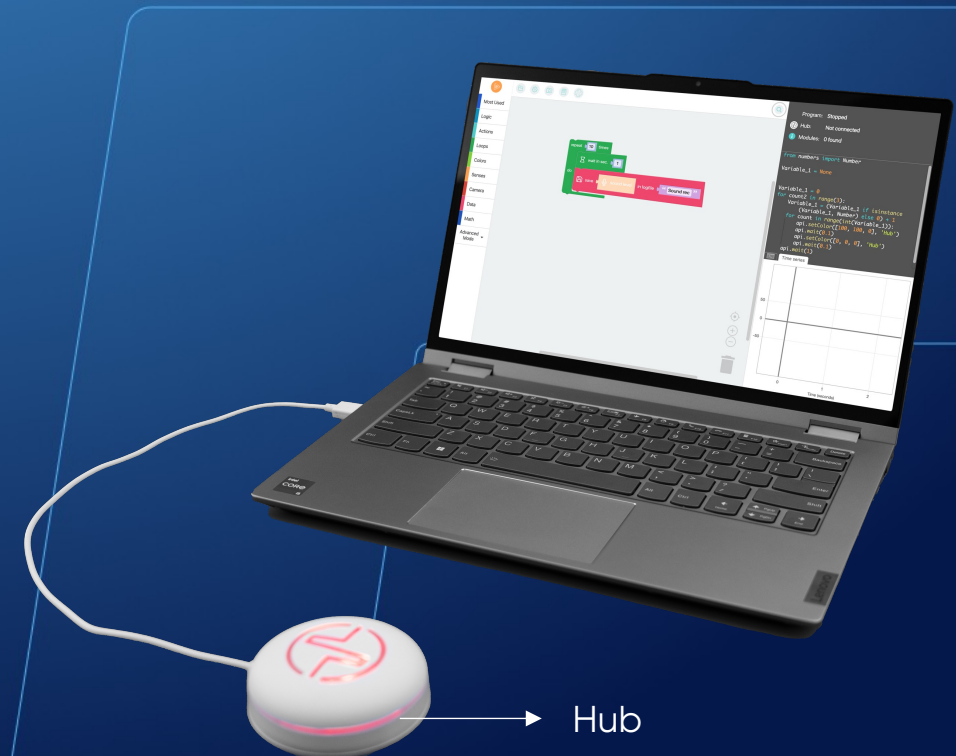



```
repeat forever
  set speed A: 50 B: 0 on 1|11
  do
    if key pressed? spacebar
      do
        break out of loop
  print "The continuous loop has been interrupted."
```

Kod 60: Nagrywanie w pliku .csv

- Program nagrywa poziom hałasu z mikrofonu komputera w pętli 10 powtórzeń na sekundę. Dane są zapisywane numerycznie w pliku CSV o nazwie określonej przez osobę programującą. Plik jest zapisywany w folderze 'Documents' pod folderem 'Fable'. Dane zapisane w pliku CSV mogą być wykorzystane w innych programach wedle indywidualnych potrzeb.

OBEJRZYJ WIDEO

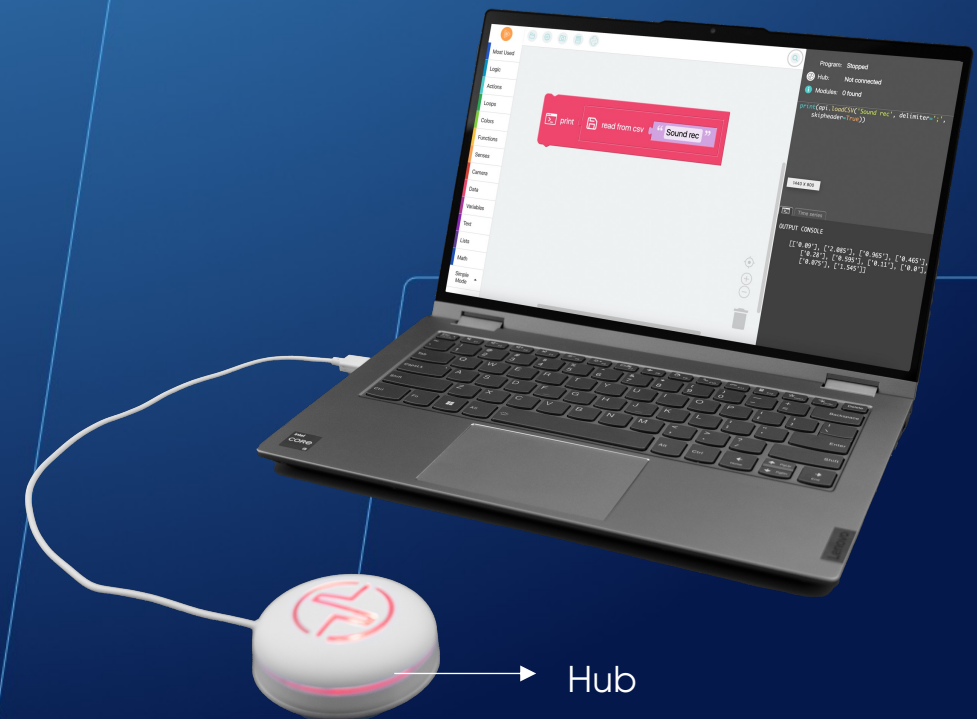


```
repeat 10 times  
  wait in sec. 1  
  do  
    save sound level in logfile "Sound rec"
```

Kod 61: Odczyt z pliku .csv

- Program odczytuje dane z pliku .csv o nazwie „Sound rec”. Następnie wyświetla te dane w Konsoli Wyjściowej. Dane mogą być przetwarzane i wykorzystywane do innych celów. Jako przykład użyliśmy pliku zapisanego w programie „Record in a .csv File”.

OBEJRZYJ WIDEO





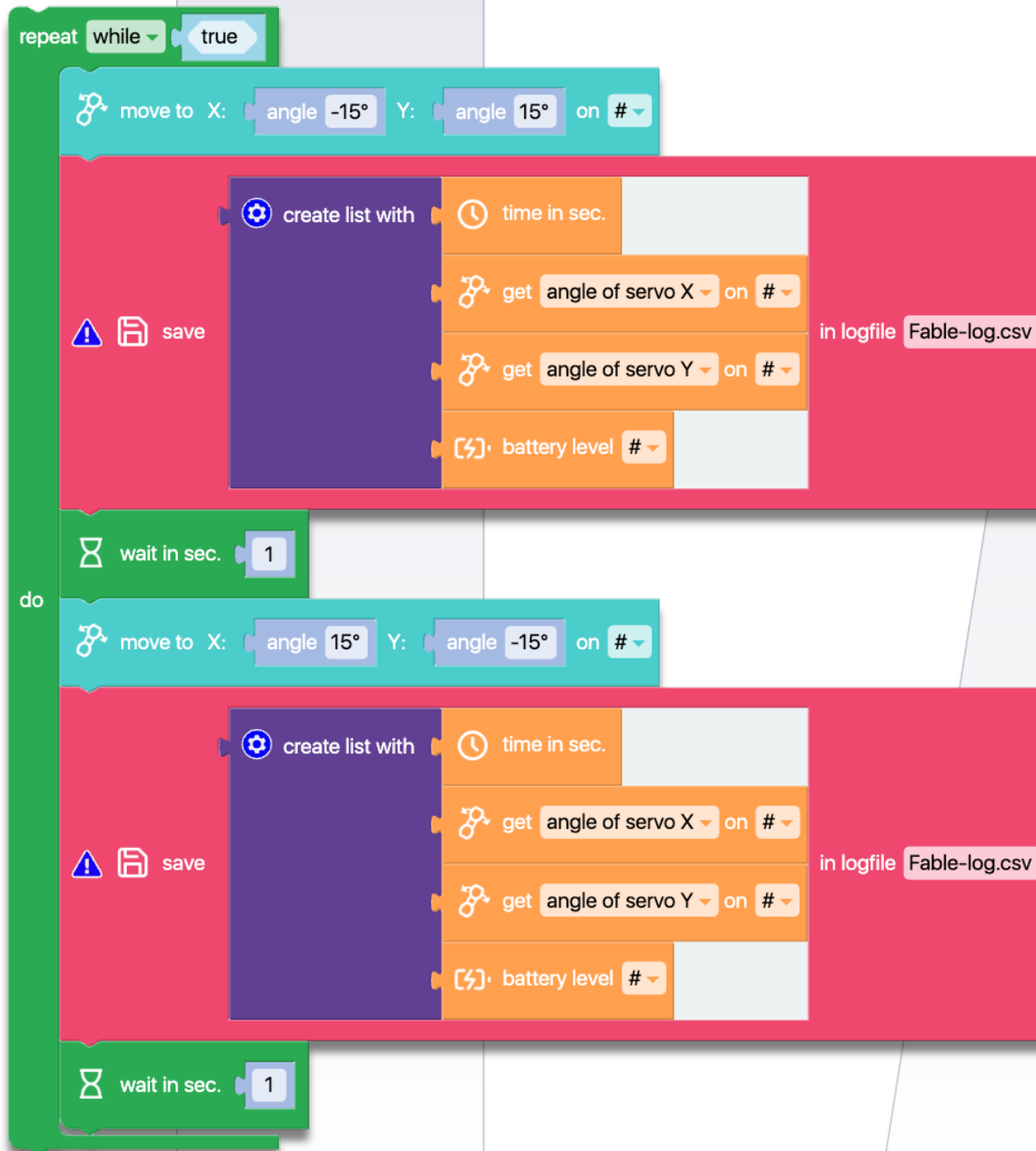
Kod 62:

Nagrywanie wielu danych w pliku .csv

- Po uruchomieniu programu, automatycznie zostanie wygenerowany plik z rozszerzeniem .csv (można ustawić nazwę pliku). Ten plik przechowuje różne wartości, takie jak: kąty silników X i Y Modułu Joint, który jest połączony z Hubem; poziom baterii Modułu Joint; czas, który upłynął od rozpoczęcia programu.

OBEJRZYJ WIDEO

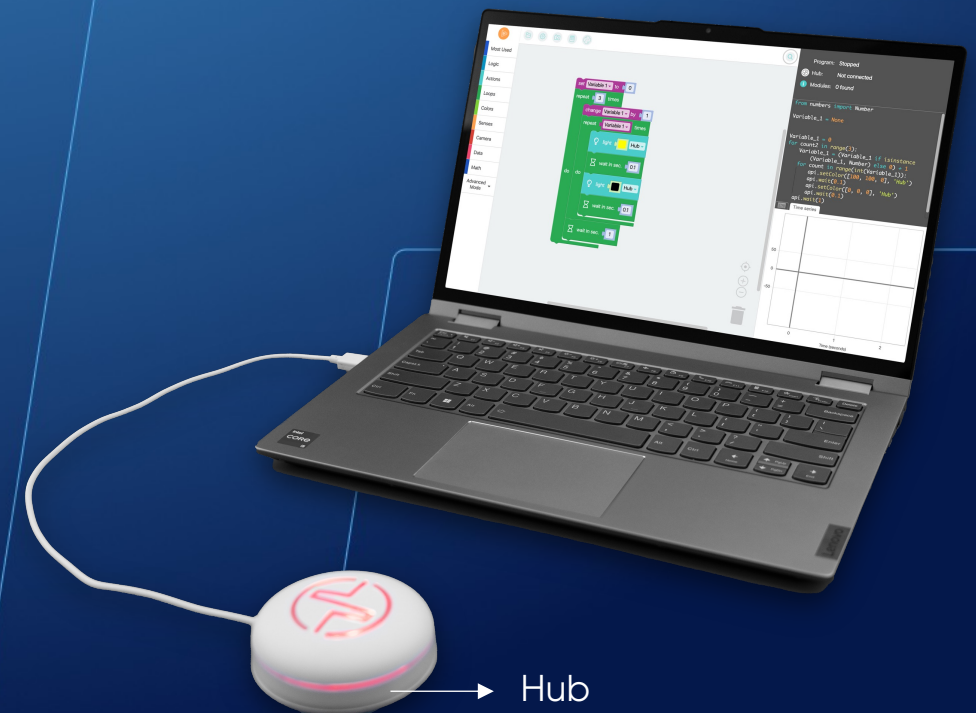


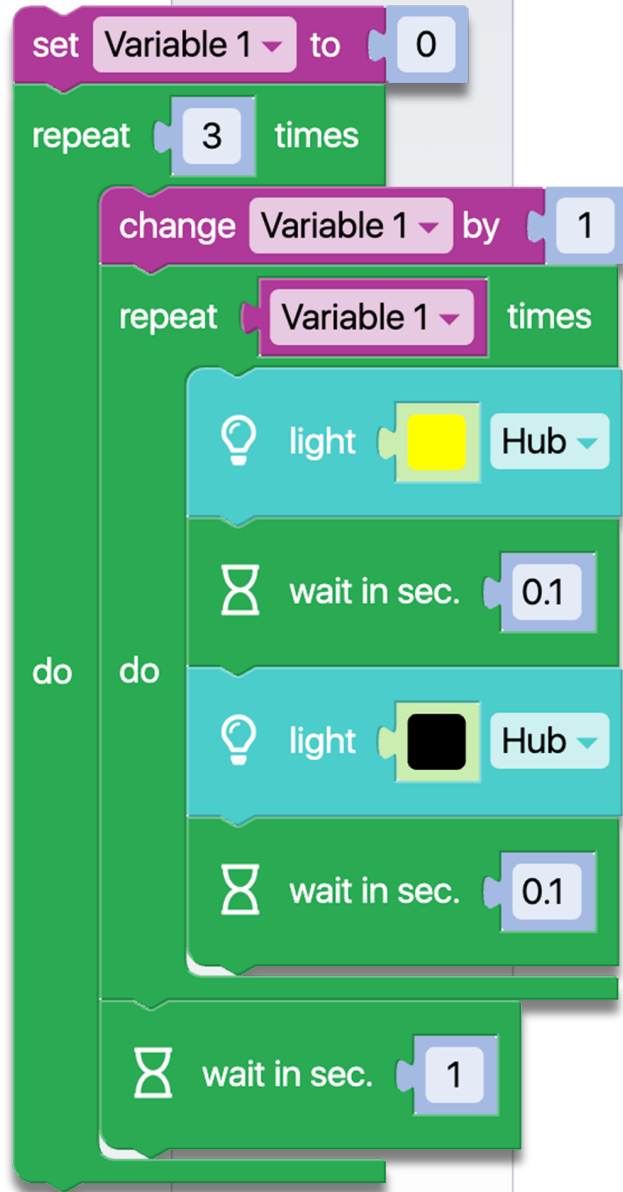


Kod 63: Tworzenie zmiennej

- Aby utworzyć zmienną w trybie zaawansowanym, możemy postępować według podanych kroków: Kliknij na „Zmienne”, a następnie na „Utwórz Zmienną”. Po wykonaniu tego możemy nadać naszej zmiennej nazwę. Na przykład możemy nazwać ją „Zmienna 1”.
- Na początku programu, Zmienna 1 ma wartość liczbową zero. Jednak podczas pętli trzykrotnych powtórzeń, wartość zmiennej będzie zmieniać się przez dodanie jednej jednostki. Ponadto, Hub zaświeci się na żółto z zaktualizowaną wartością.
- Po trzykrotnym zaktualizowaniu wartości zmiennej pętla zostanie zakończona ze względu na wykonanie trzech powtórzeń.

OBEJRZYJ WIDEO

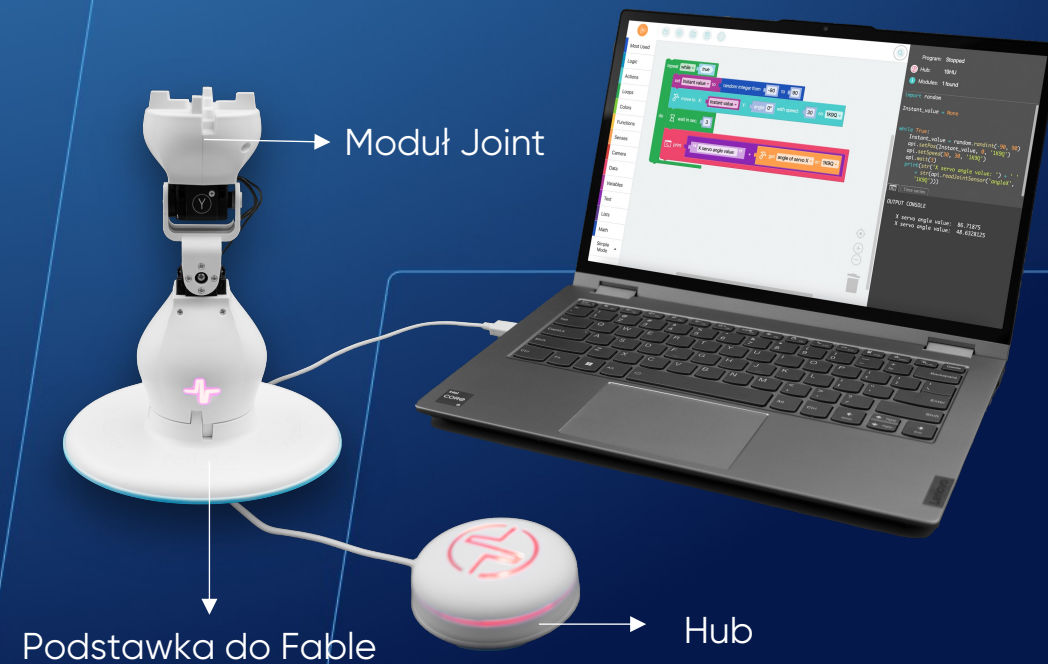


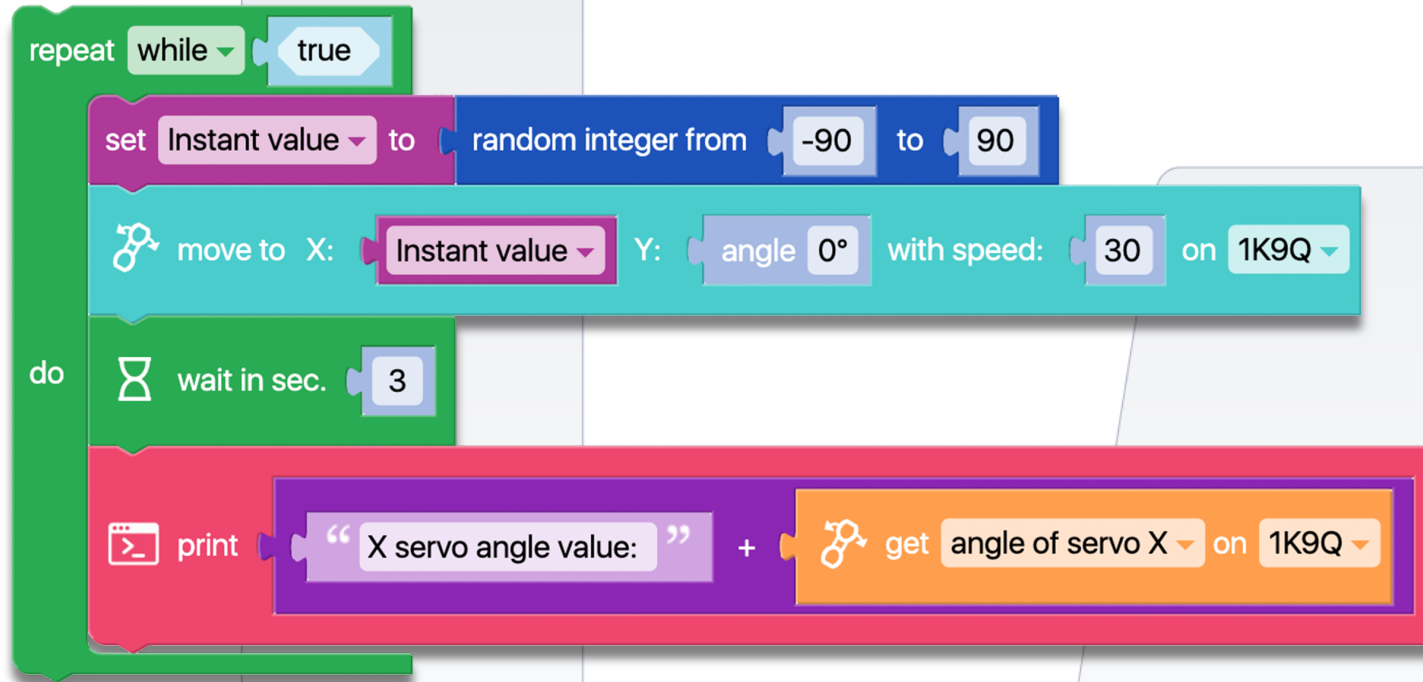


Kod 64: Generowanie losowych wartości dla serwomotora Modułu Joint

- Program generuje losową liczbę w zakresie od -90 do 90, reprezentującą zakres ruchu serwomotora X w Module Joint. Ta losowa liczba jest zapisywana w zmiennej „Wartość chwilowa” i wyświetlana w konsoli wyjściowej. Nowy ruch wykonywany jest co 3 sekundy.

OBEJRZYJ WIDEO





Kod 65: Obliczanie średniej arytmetycznej ocen

- Program jest zaprojektowany do obliczania średniej arytmetycznej ocen. Oceny, w zakresie od 1 do 5, są wprowadzane do programu poprzez dotknięcie ekranu aplikacji Fable Face na telefonie. Na przykład, jeśli na ekranie zostaną umieszczone trzy palce, program rejestruje wartość 3. Ta metoda jest używana zarówno do wprowadzania całkowitej liczby ocen, jak i każdej pojedynczej oceny.

OBEJRZYJ WIDEO



```
set Sum to 0
set Total to 0
set Note to 0

print "Put as many fingers on the screen as you have no..."

wait until key pressed? spacebar

set Total to get tap count

print "Total notes: " + "=" + Total

wait in sec. 3

repeat Total times
  print "your note please(use fingers):"
  wait until key pressed? spacebar
  do
    set Note to get tap count
    print "Note is: " + "=" + Note
    wait in sec. 1
    change Sum by Note

print "Your average grade is: " + Sum + ":" + Total + "=" + Sum ÷ Total
```

Kod 66: Obliczanie średniej geometrycznej dla dwóch liczb

- Program odczytuje dwie liczby z ekranu telefonu, oblicza i wyświetla ich średnią geometryczną.

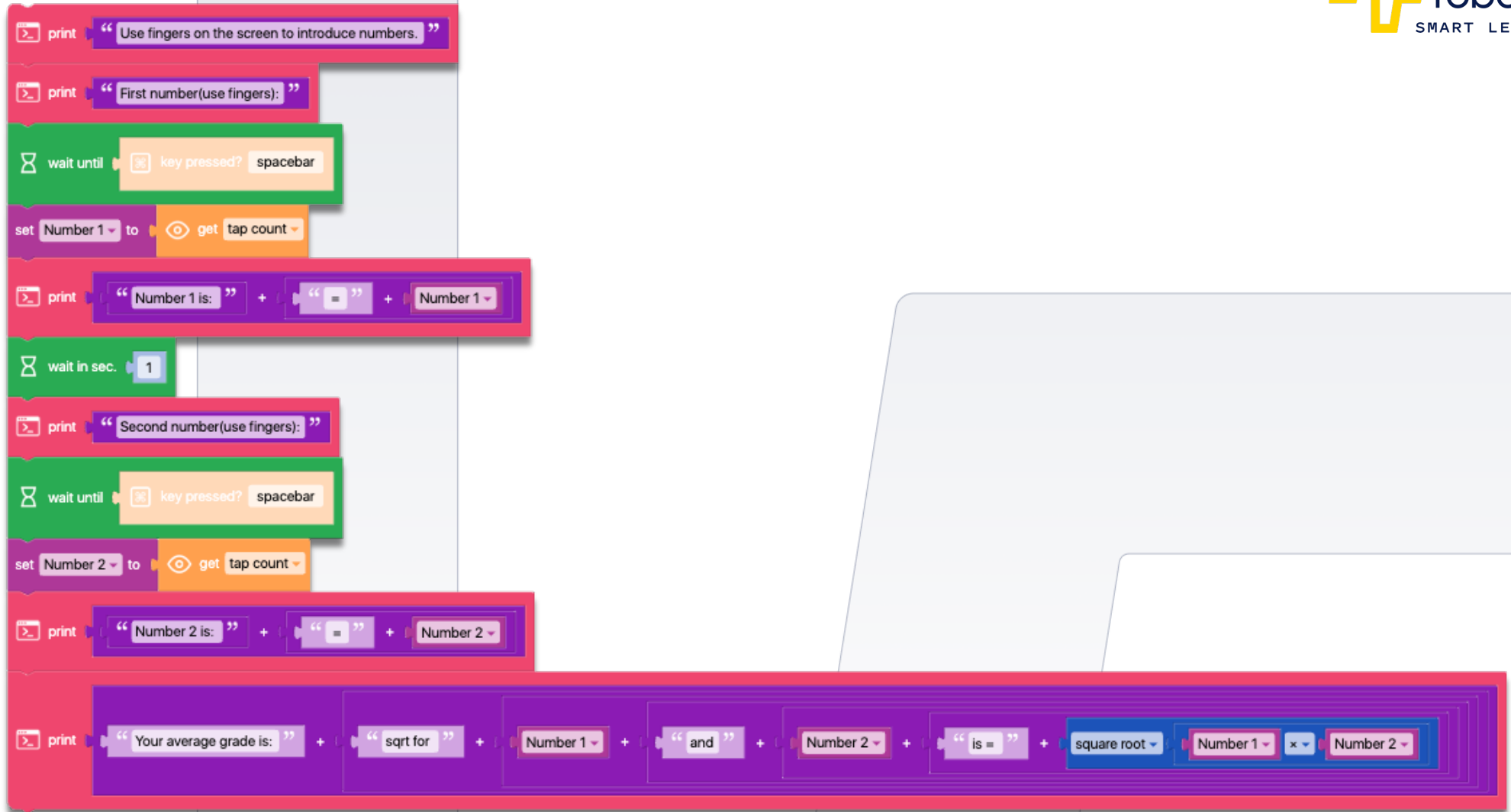
OBEJRZYJ WIDEO



Podstawa do Fable



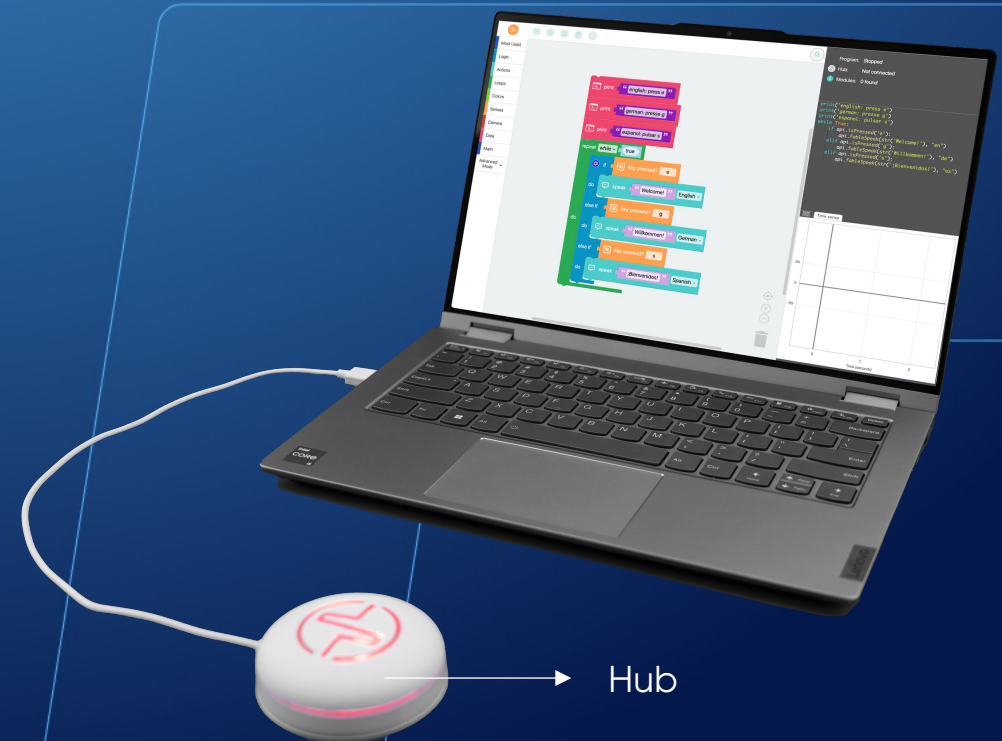
Hub



Kod 67: Konwersja tekstu na mowę

- Sekwencja kodu przetwarza tekst i konwertuje go na mowę. Użytkownicy mają możliwość wyboru języka, a wygenerowana mowa będzie odzwierciedlać akcent odpowiadający temu językowi. Ten kod może być przydatny do wysyłania wiadomości audio lub do różnych innych zastosowań wymagających funkcji zamiany tekstu na mowę.

OBEJRZYJ WIDEO



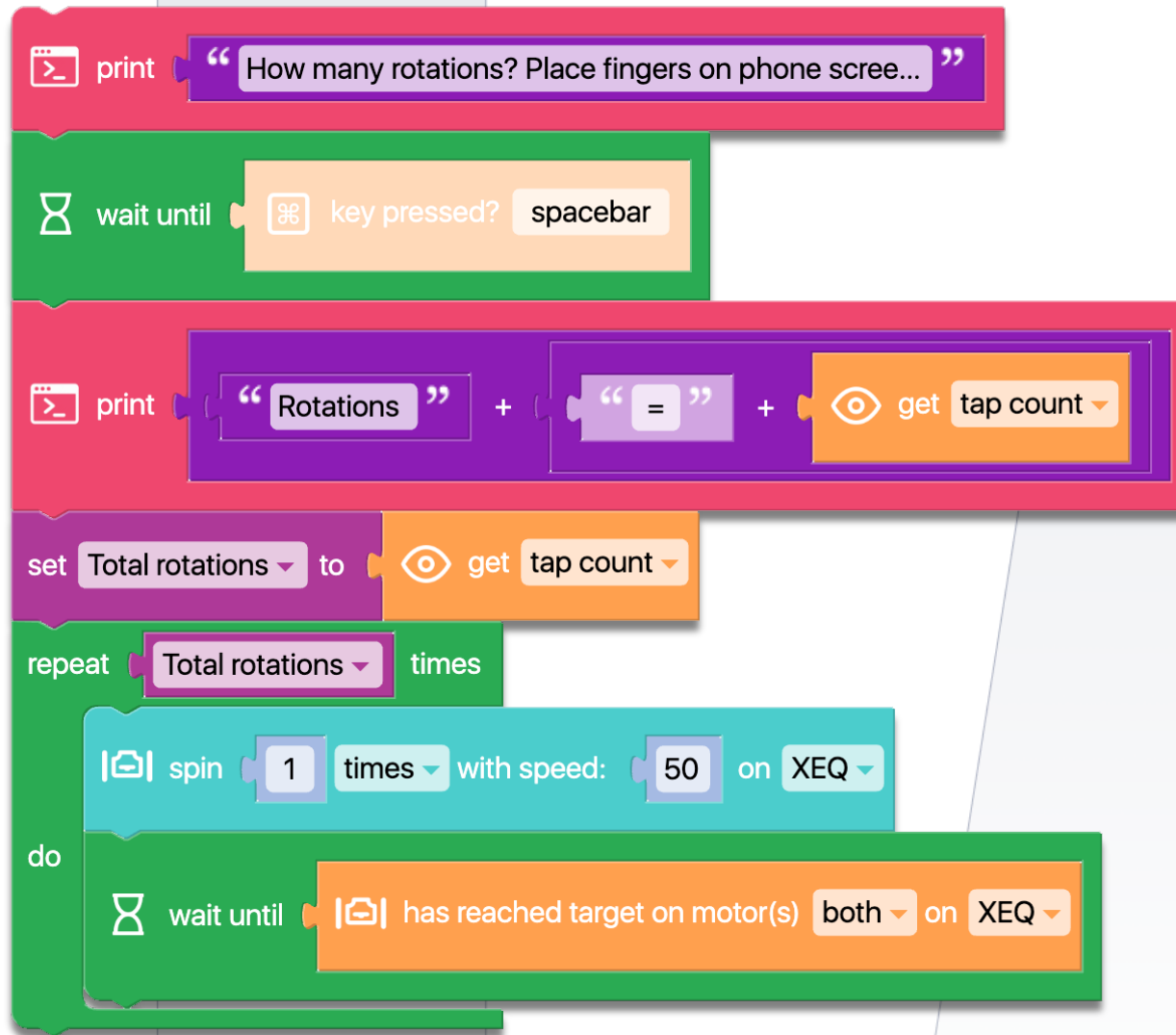

```
print "english: press e"  
print "german: presse g"  
print "espanol: pulsar s"  
repeat while true  
  if key pressed? e  
    do speak "Welcome!" English  
  else if key pressed? g  
    do speak "Willkommen!" German  
  else if key pressed? s  
    do speak "¡Bienvenidos!" Spanish
```

Kod 68: Odczyt liczb z ekranu telefonu

- Program jest zaprojektowany do odczytywania liczb z ekranu telefonu (z otwartą aplikacją Fable Face). Gdy dwa palce dotkną ekranu i zostanie naciśnięty klawisz Spacja, program wykorzystuje tę liczbę do obrócenia Modułu Spin tyle razy, ile wynosi wprowadzona liczba - czyli w tym przypadku dwa razy. Ta metoda wprowadzania liczb za pomocą telefonu może służyć jako klawiatura numeryczna. Możesz wypróbować tę funkcję z innymi liczbami.

OBEJRZYJ WIDEO





Kod 69: Odczyt liczb z ekranu telefonu Formuła n choose k

- Program wykorzystuje Fable Face do odczytu dwóch wartości, które są wprowadzane poprzez wykrycie liczby palców dotykających ekranu. Wartości te są następnie wykorzystywane do obliczania kombinacji n przyjmowanych jako k , gdzie n i k są liczbami wprowadzonymi za pomocą telefonu. Wynik jest wyświetlany w Konsoli Wyjściowej.

OBEJRZYJ WIDEO



```

set Numerator to 1
Input numbers
if n < k
do
  print "Because n<k, the result is zero"
  stop program
set Numerator to n
set Denominator to k
count with i from 1 to k-1 by 1
do set Numerator to Numerator * n-i
count with j from 1 to k-1 by 1
do set Denominator to Denominator * k-j
print "Combinations of " + n + " choose " + k + " equal " + Numerator / Denominator
  
```

```

to Input numbers
  print "input n(use fingers on phone screen)"
  wait until key pressed? spacebar
  set n to get tap count
  print "n" + "=" + n
  print "input k(use fingers on phone screen)"
  wait until key pressed? spacebar
  set k to get tap count
  print "k" + "=" + k
  
```

Print something in the box to the right

Kod 70: Odczyt liczb z ekranu telefonu Formuła n!

- Program oblicza n! poprzez wykrywanie liczby palców na ekranie Fable Face. Zarówno dane wejściowe, jak i ostateczny wynik są wyświetlane w Konsoli Wyjściowej.

OBEJRZYJ WIDEO



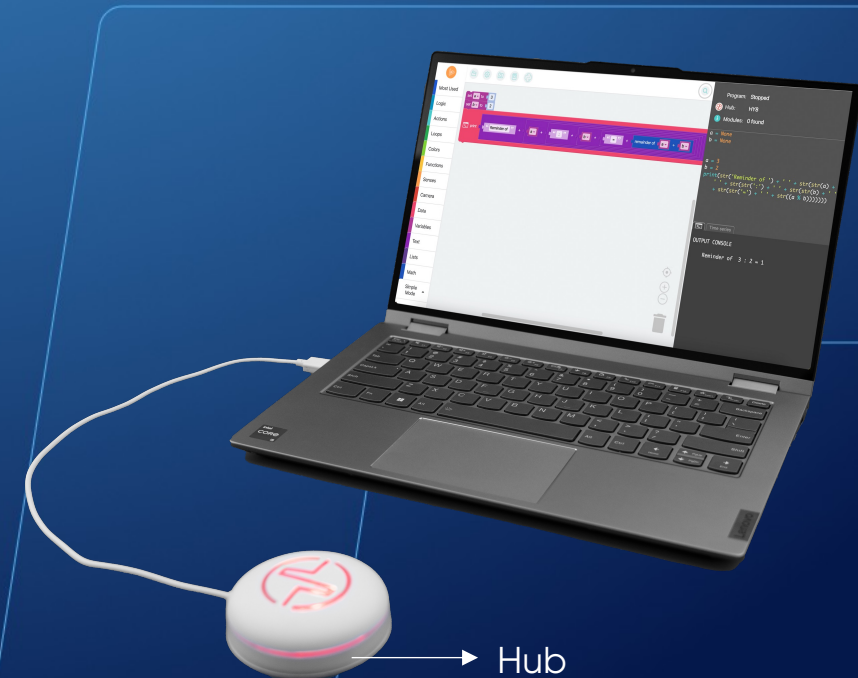
```
print "Input number by touching phone screen with your ..."  
wait until key pressed? spacebar  
print "Your number is: " + get tap count  
set n to get tap count  
if n = 0  
do  
  print n + " != " + 0  
  stop program  
set Result to 1  
count with i from 1 to n by 1  
do set Result to Result x i  
print n + " != " + Result
```

Kod 71:

Obliczanie reszty z dzielenia – Metoda 1

- Sekwencja programu wymaga wprowadzenia dwóch danych liczbowych i oblicza resztę z dzielenia pierwszej liczby przez drugą. Początkowe dane numeryczne muszą zostać wprowadzone ręcznie na początku sekwencji. Jednak w bardziej rozbudowanym programie wspomniane dane wejściowe mogą pochodzić z sensorów lub innych obliczeń wykonywanych w poprzednich liniach kodu. Wynik jest wyświetlany w Konsoli Wyjściowej.

OBEJRZYJ WIDEO



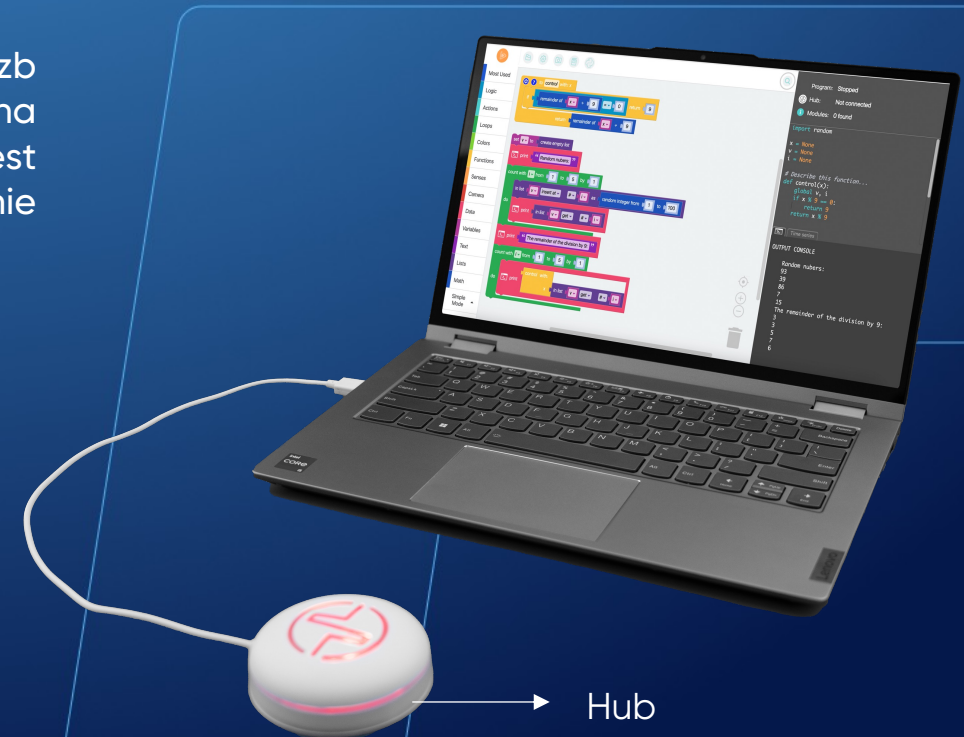

```
set a to 3
set b to 2
print "Reminder of " + a + " : " + b + " = remainder of " + a + " ÷ " + b
```

Kod 72:

Obliczanie reszty z dzielenia – Metoda 2

- Program wykorzystuje funkcję „Zwróć”. Generowanych jest pięć liczb losowych z zakresu [1,100]. Każda liczba jest następnie sprawdzana w celu określenia, czy jest podzielna przez 9, a reszta z dzielenia jest zapisywana. Wartości reszt są zapisywane na liście, a następnie wyświetlane w Konsoli Wyjściowej.

OBEJRZYJ WIDEO



```
to control with: x
  if remainder of x ÷ 9 = 0 return 9
  return remainder of x ÷ 9
```

```
set v to create empty list
print "Random nubers:"
count with i from 1 to 5 by 1
  in list v insert at # i as random integer from 1 to 100
do
  print in list v get # i
print "The remainder of the division by 9:"
count with i from 1 to 5 by 1
do
  print control with:
    x in list v get # i
```

Kod 73: Obliczanie wystąpień

- Ta sekwencja kodu jest zaprojektowana do zliczania wystąpień koloru czerwonego przed Modułem Spin. Wyświetli licznik i zatrzyma działanie, gdy liczba wystąpień osiągnie dziesięć. Ta sekwencja jest przydatna w programach do zarządzania sklepem i może być używana do monitorowania poziomów zapasów. Jest zdolna do dodawania i odejmowania z zapasów oraz ustawiania alertów, gdy poziomy zapasów spadają poniżej określonego progu.

OBEJRZYJ WIDEO

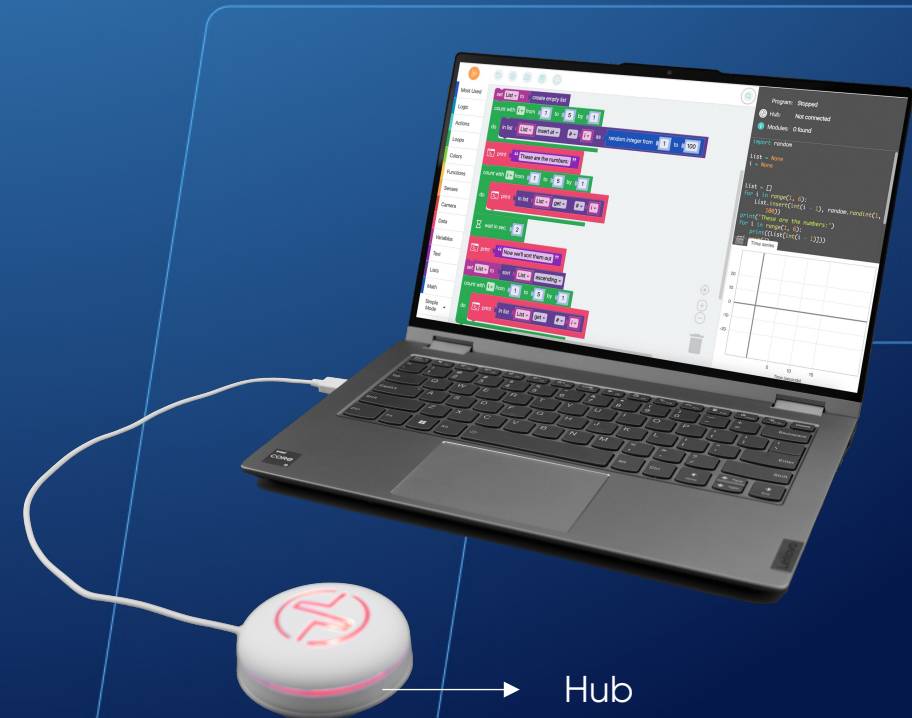


```
set Count to 0
repeat until Count = 10
  if check for color red on 1111
    do
      change Count by 1
      wait in sec. 1
  print Count
```

Kod 74: Sortowanie liczb w porządku rosnącym

- Program generuje 5 losowych liczb, dodaje je do listy, a następnie sortuje je rosnąco.

OBEJRZYJ WIDEO



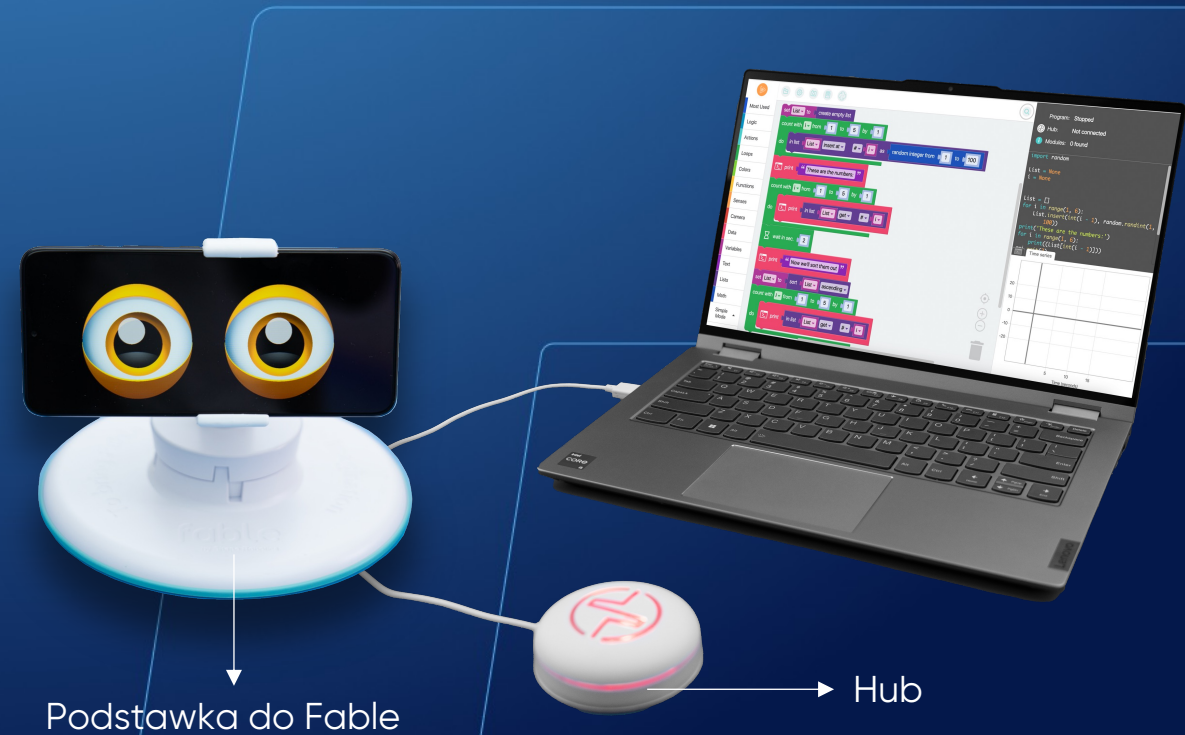
```
set List to create empty list
count with i from 1 to 5 by 1
do in list List insert at # i as random integer from 1 to 100
print "These are the numbers:"
count with i from 1 to 5 by 1
do print in list List get # i
wait in sec. 2
print "Now we'll sort them out"
set List to sort List ascending
count with i from 1 to 5 by 1
do print in list List get # i
```

Kod 75:

Wprowadzanie danych w określonej kolejności (Lista)

- Sekwencja kodu tworzy listę czterech elementów: indeks 0, indeks 1, indeks 2, indeks 3. Wartości każdego elementu (indeksu) są wprowadzane za pomocą Fable Face poprzez zliczanie liczby palców na ekranie. Na pytanie "Który element chcesz zobaczyć?", również korzystamy z Fable Face, aby wybrać indeks, który chcemy wyświetlić, i który następnie jest wyświetlany w Konsoli Wyjściowej.

OBEJRZYJ WIDEO



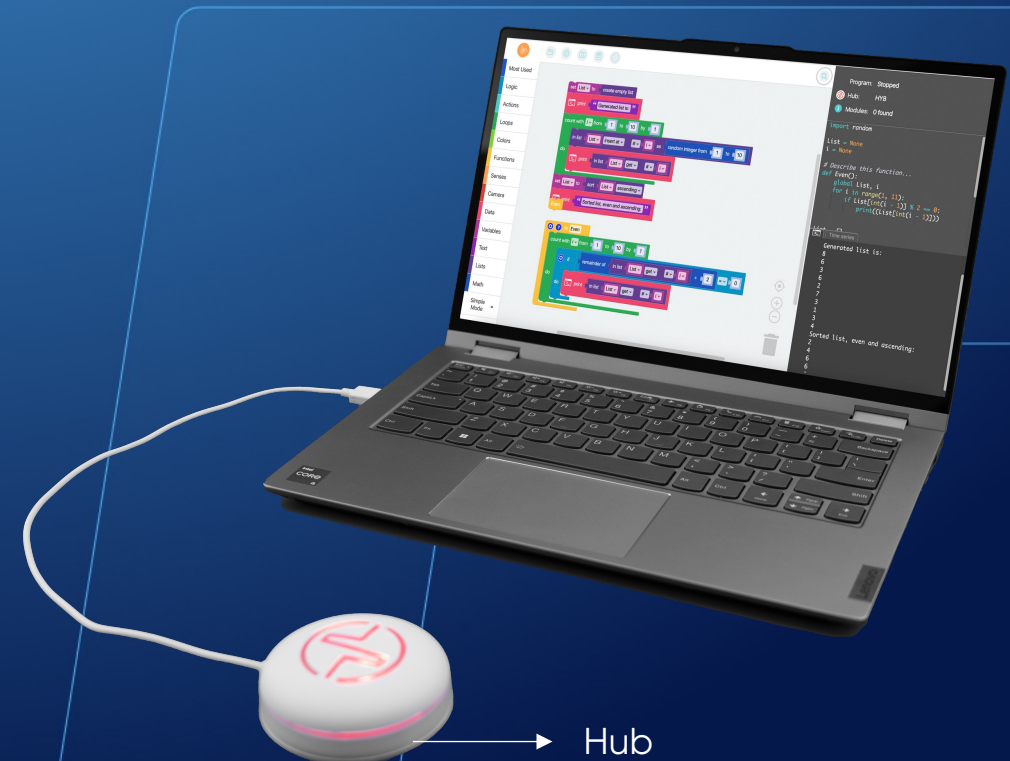

```
set List to create empty list
count with i from 0 to 3 by 1
  print "Tap the screen and press SPACE"
  wait until key pressed? spacebar
  do
    print "Index " + i + " : " + get tap count
    in list List insert at # i as get tap count
    wait in sec. 0.2
  print "Which item do you want to see? Tap the screen an..."
  wait until key pressed? spacebar
  set Index to get tap count
  print "Index " + Index + " is: " + in list List get # Index
```

Kod 76:

Sortowanie liczb parzystych rosnąco

- Program generuje listę 10 elementów, rejestrując losowo wygenerowane liczby z zakresu od 1 do 10. Elementy listy są sortowane rosnąco, a następnie uruchamiana jest funkcja, która wyodrębnia tylko liczby parzyste, sprawdzając resztę z dzielenia przez 2. Na końcu liczby są wyświetlane w Konsoli Wyjściowej.

OBEJRZYJ WIDEO



```
set List to create empty list
print "Generated list is:"

count with i from 1 to 10 by 1
do
  in list List insert at # i as random integer from 1 to 10
do
  print in list List get # i

set List to sort List ascending
print "Sorted list, even and ascending:"

Even
```

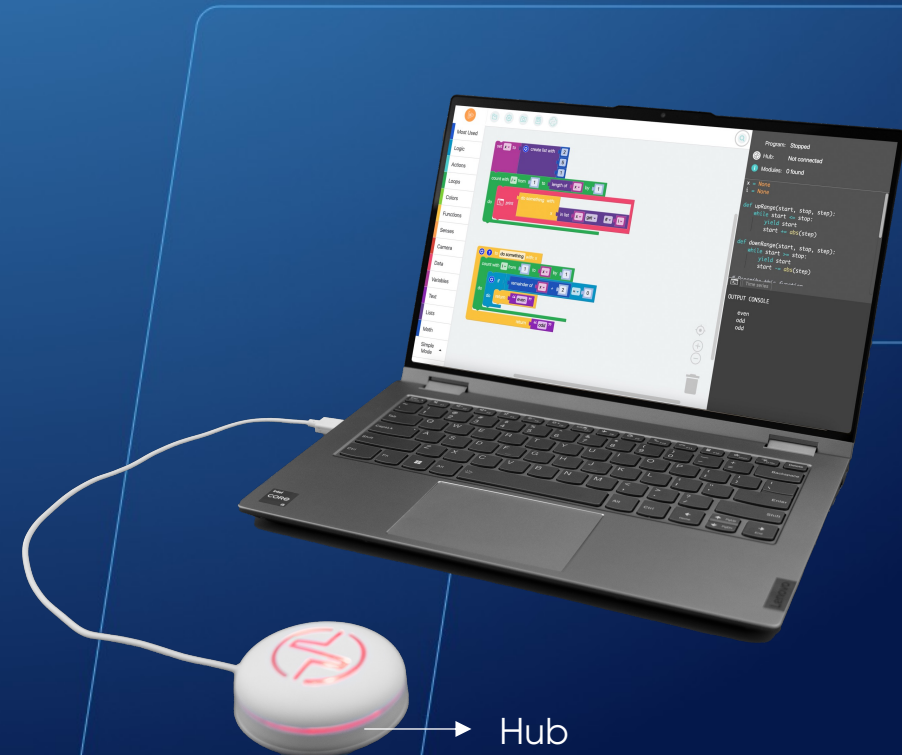
```
to Even
  count with i from 1 to 10 by 1
  if remainder of in list List get # i ÷ 2 = 0
  do
    print in list List get # i
```

Kod 77:

Wyodrębnianie liczb parzystych/nieparzystych z Listy i Funkcji

- Przed uruchomieniem programu użytkownik jest proszony o wprowadzenie trzech liczb za pomocą klawiatury. Liczby te mogą być pobrane z sensorów lub obliczeń pośrednich i są przechowywane w liście. Następnie używana jest funkcja do określenia, czy każda liczba w liście jest parzysta czy nieparzysta. Wynik jest wyświetlany w konsoli.

OBEJRZYJ WIDEO



```
set x to create list with 1 2 8
count with i from 1 to length of x by 1
do
  print do something with:
    x in list x get # i
```

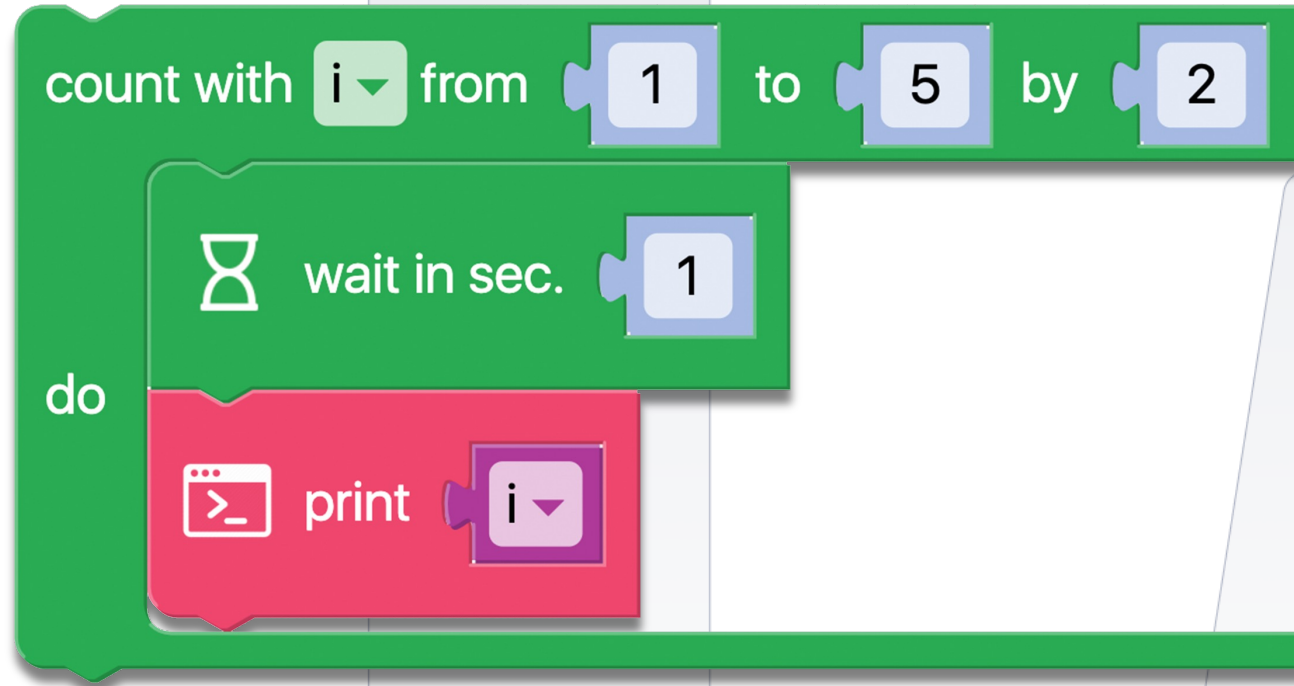
```
to do something with: x
  count with i from 1 to x by 1
  if remainder of x ÷ 2 = 0
  do
    return "even"
  return "odd"
```

Kod 78: Użycie Polecenia „licz z”

- Sekwencja programu to powtarzalna struktura, przydatna, gdy chcemy wykonać instrukcje określoną liczbę razy, zamiast wykonywać kod na podstawie warunku (while). W tej sekwencji wyświetlane są liczby nieparzyste w zamkniętym przedziale [1, 5].

OBEJRZYJ WIDEO



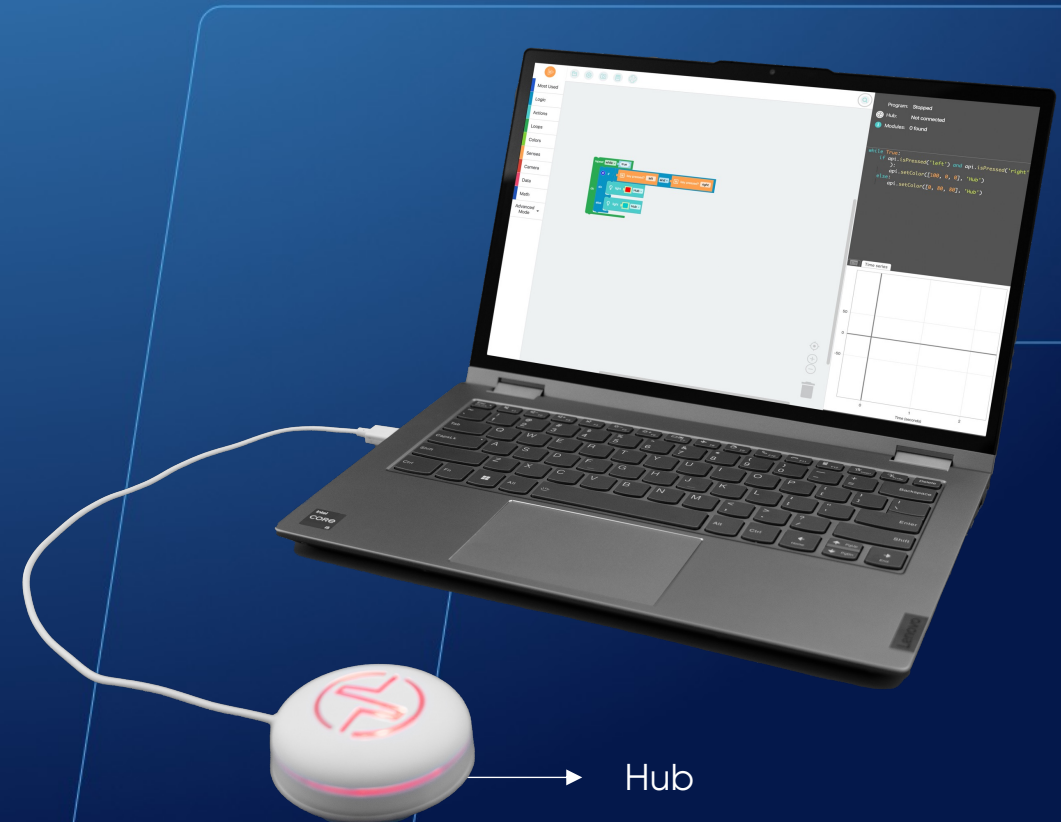


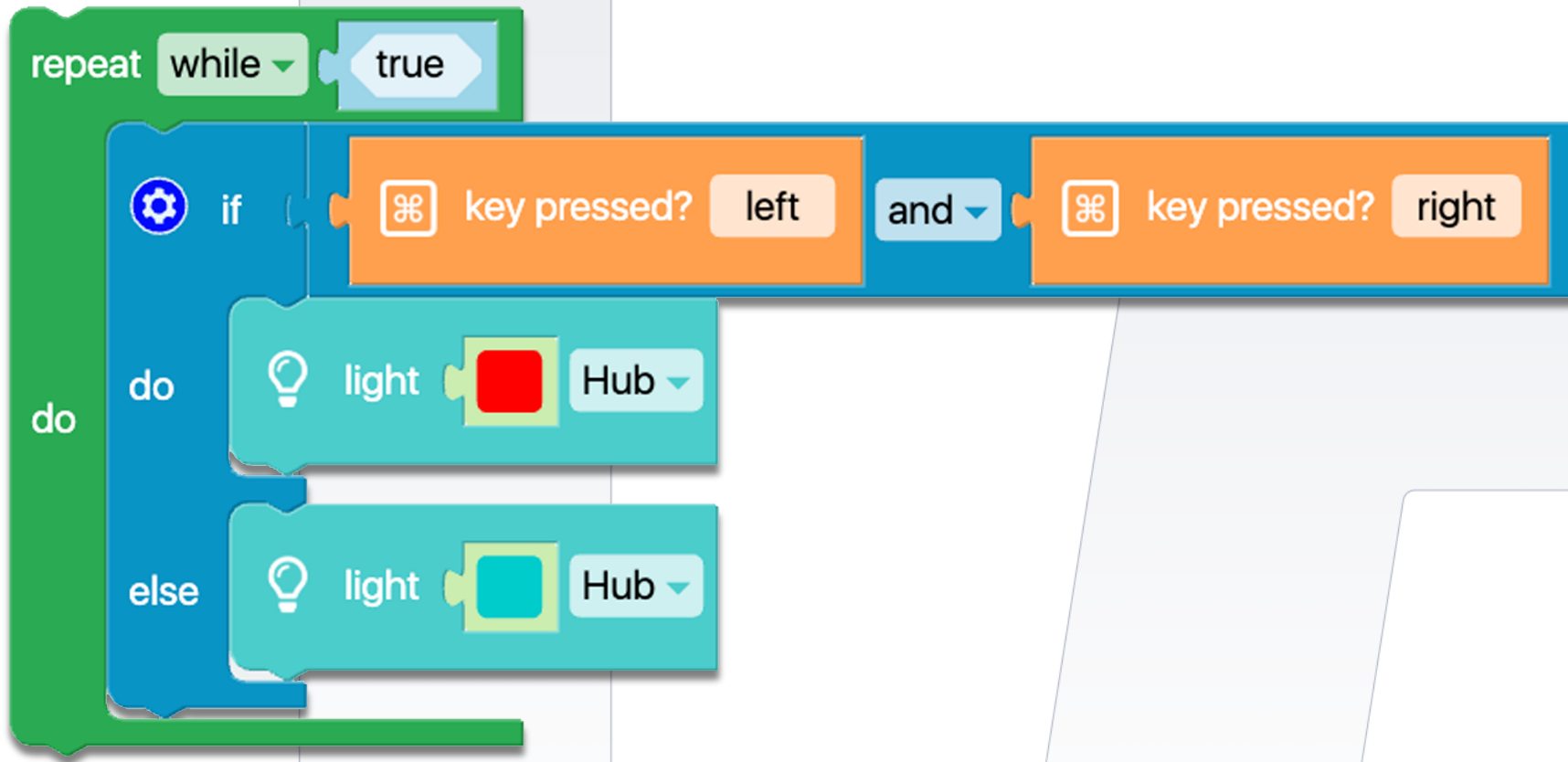
Kod 79:

Boolowski Operator Logiczny - I

- Po spełnieniu warunku, zostaje wykonana sekwencja kodu. Światło Huba będzie pozostawać zielone, dopóki nie zostaną naciśnięte jednocześnie klawisze strzałki w lewo i prawo. Jeśli oba klawisze zostaną naciśnięte jednocześnie, światło zmieni się na czerwone. Ten operator jest szczególnie użyteczny przy odczycie danych z wielu sensorów i podejmowaniu decyzji na podstawie kombinacji ich odczytów.

OBEJRZYJ WIDEO

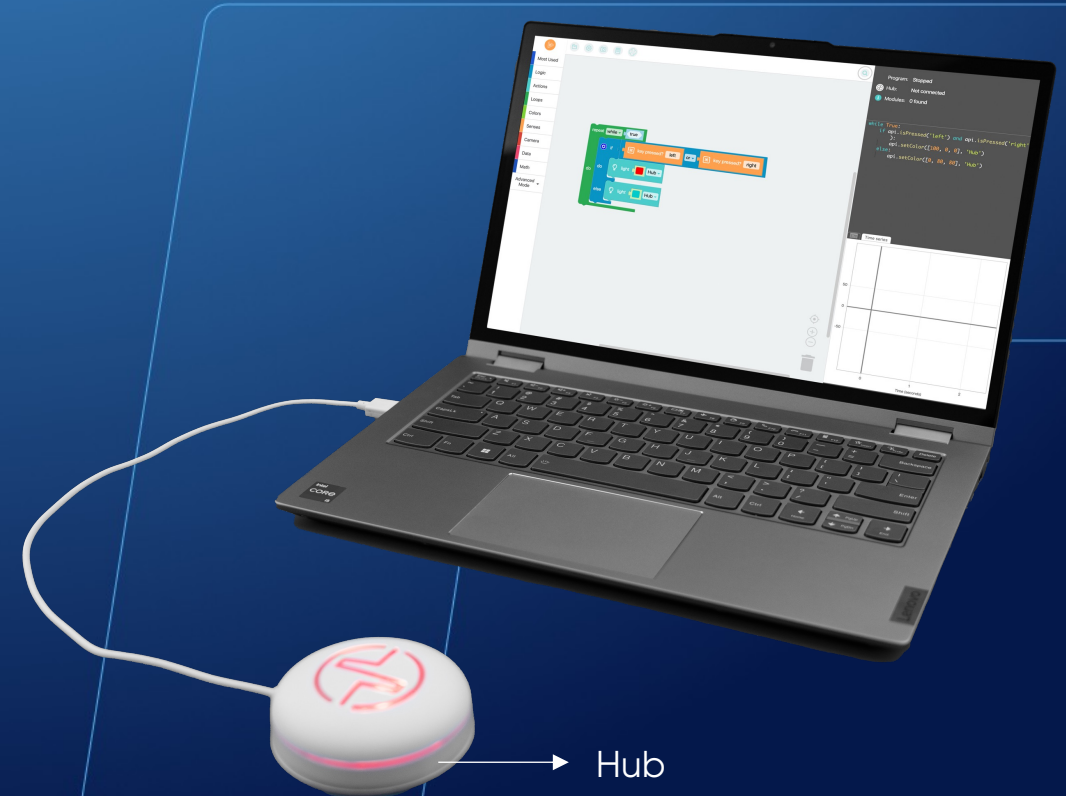


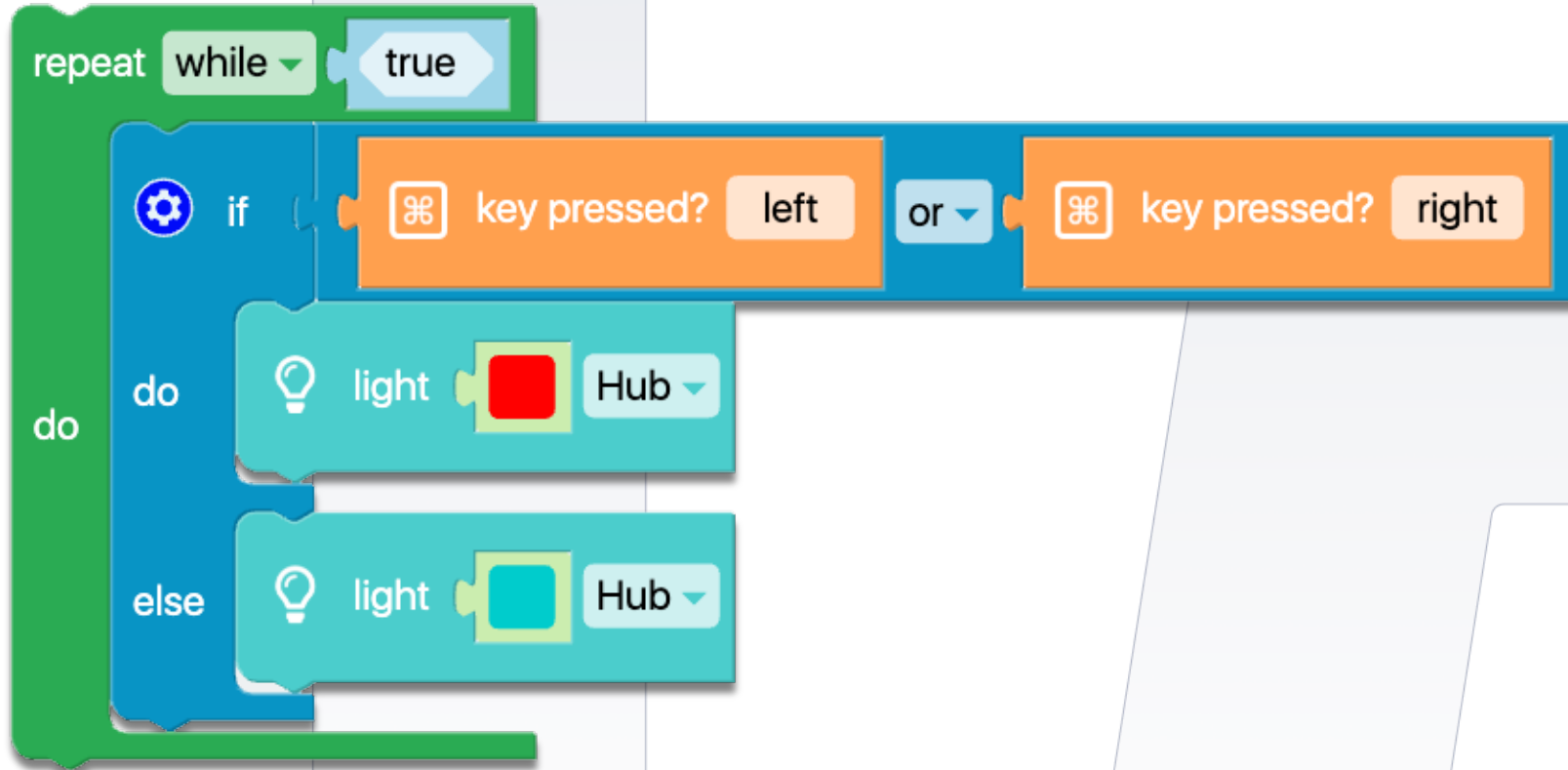


Kod 80: Boolowski Operator Logiczny – LUB

- Sekwencja kodu sprawia, że Hub pozostaje niebieski tak długo, jak na klawiaturze wciśnięty jest klawisz strzałki w lewo lub klawisz strzałki w prawo. Jeśli jednak naciśnięty zostanie jeden z tych klawiszy, Hub zmieni kolor na czerwony. Sekwencja ta jest przydatna w programach, w których warunek może zostać spełniony na dwa różne sposoby, np. poprzez naciśnięcie klawisza lub aktywację sensora.

OBEJRZYJ WIDEO





Kod 81: Boolowski Operator Logiczny – NIE

- Ta sekwencja kodu jest zaprojektowana tak, aby Moduł Spin poruszał się do przodu wyłącznie po naciśnięciu strzałki w górę. Wykorzystywany jest NON-operator, który uruchamia się, gdy żaden klawisz nie jest wciśnięty, aktywując blok „Jeśli” z poleceniem „Zatrzymaj ruch”, jeśli światła są wyłączone. Gdy żaden klawisz nie jest wciśnięty, aktywuje się blok „W przeciwnym razie”, powodując uruchomienie silnika i włączenie światła. Warto mieć na uwadze, że takie ustawienie warunków jest odmienne od tego, do którego jesteśmy przyzwyczajeni.

OBEJRZYJ WIDEO

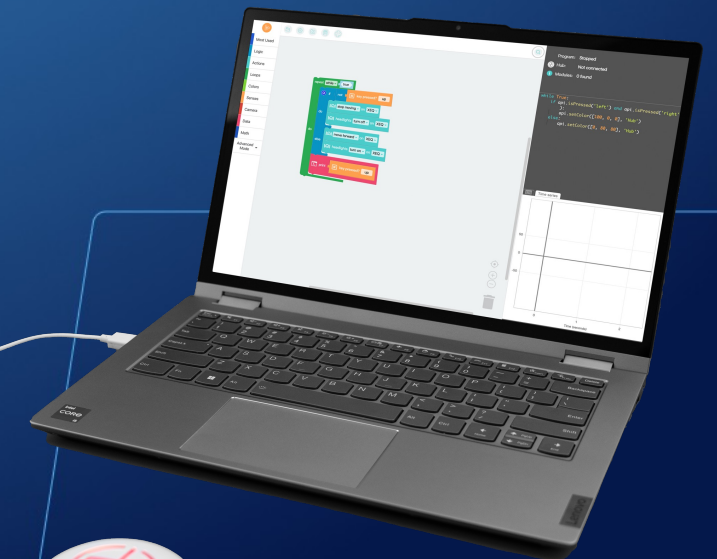


Rolka obrotowa ←

← Kółka

Moduł Spin

→ Hub





Kod 82: Podwójna negacja

- Ten kod wykorzystuje Boolowski operator logiczny NIE (Negacja logiczna), oznaczony jako "Boolowski Operator Logiczny – NIE". W związku z tym mamy ten sam program co wcześniej, ale z dodaniem dodatkowego polecenia NIE. Gdy występują dwie negacje, jest to równoważne braku negacji w ogóle. W rezultacie ten program nakazuje Modułowi Spin poruszać się do przodu, gdy żaden klawisz nie jest wciśnięty, i zatrzymać się, gdy jest wciśnięty klawisz strzałki w górę.

OBEJRZYJ WIDEO

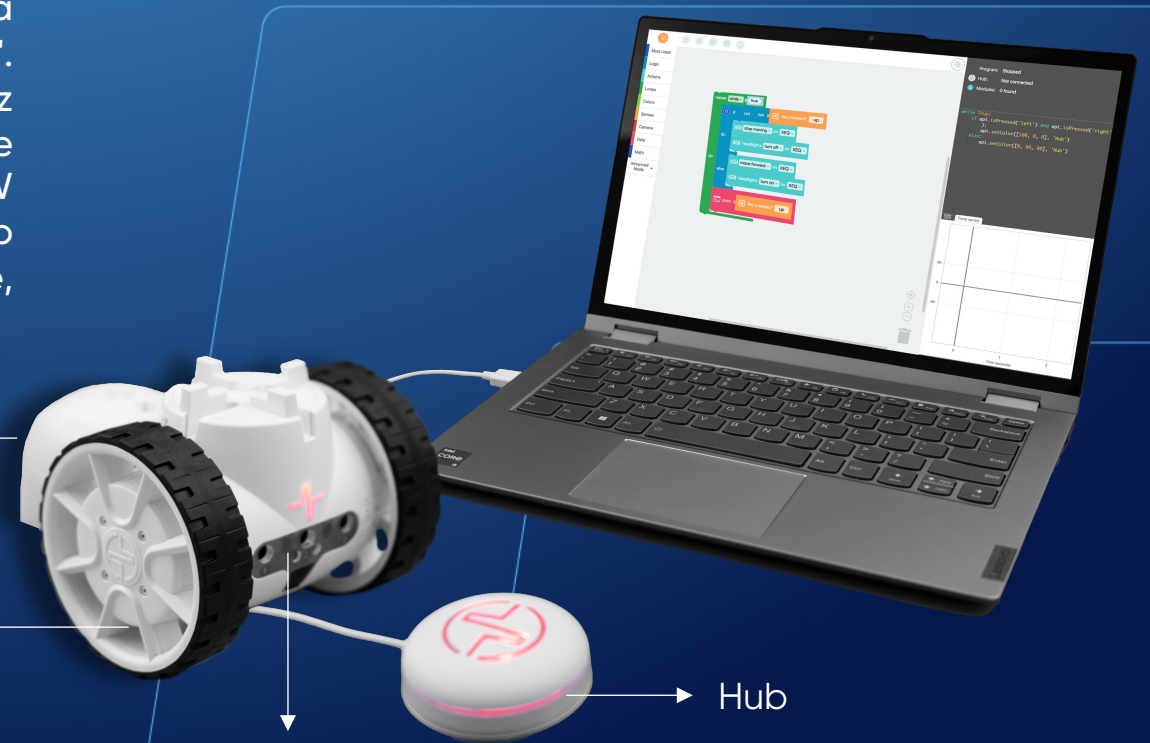


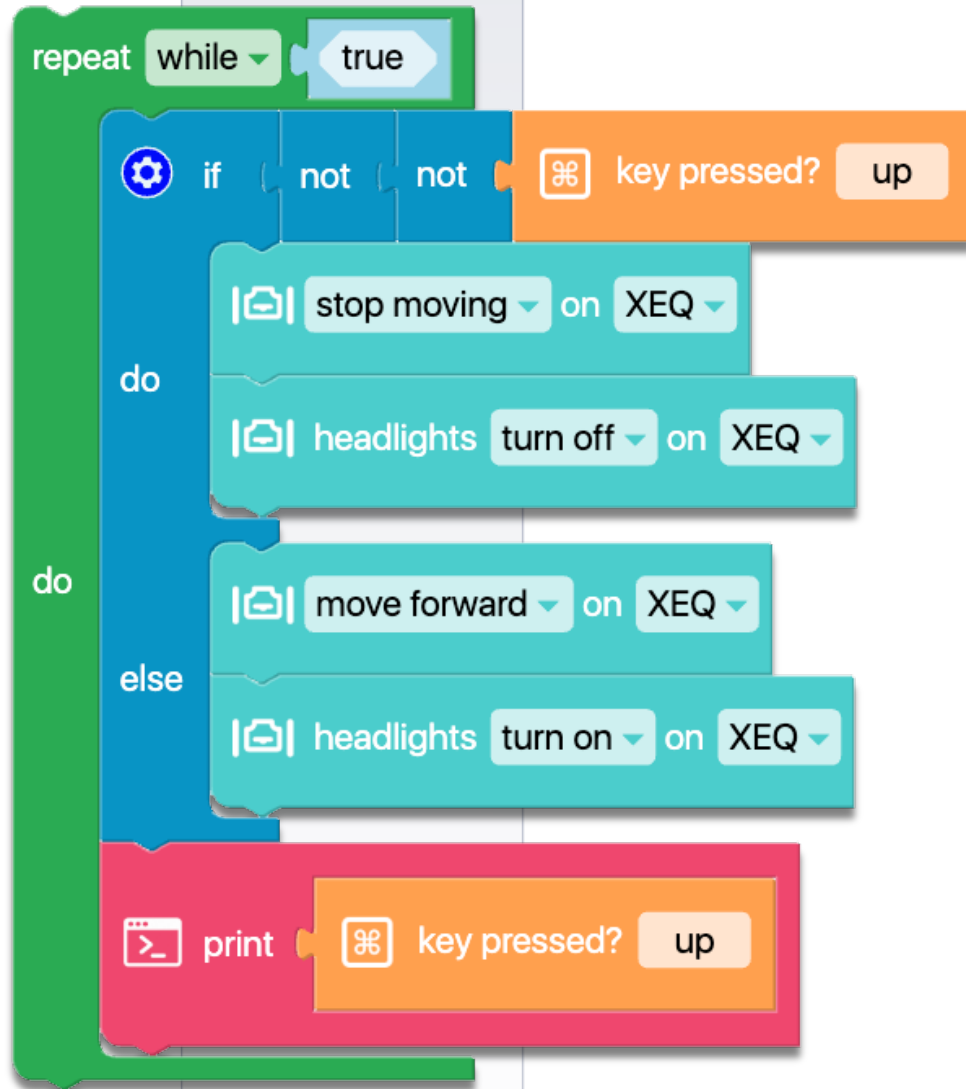
Rolka obrotowa

Kółka

Moduł Spin

Hub





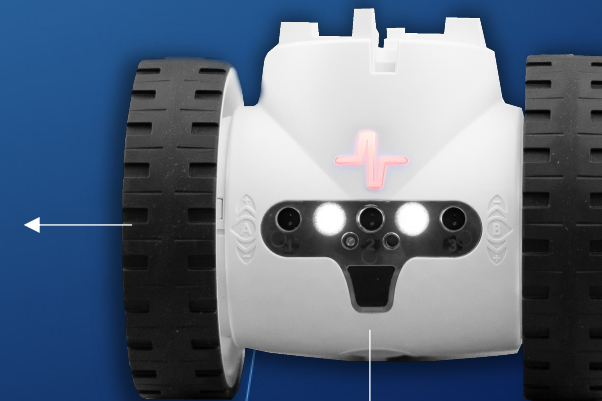
Kod 83: Światła przerywane z użyciem Modułu Spin

- Wykorzystując Moduł Spin oraz wbudowane diody LED, robot będzie wysyłał kod SOS za pomocą diod LED. Możesz dostosować czasy włączenia i wyłączenia, aby sprawdzić czy wiadomość jest nadal zrozumiała. Pauza pomiędzy literami jest reprezentowana przez komendę jednosekundową. Wykorzystano dwie oddzielne funkcje - jedną dla litery S i drugą dla litery O.

OBEJRZYJ WIDEO



Kółka



Moduł Obrotowy

S letter
O letter
S letter

```
to S letter
  repeat 3 times
    headlights turn on on 1U51
    wait in sec. 0.2
  do
    headlights turn off on 1U51
    wait in sec. 0.1
  wait in sec. 1
```

```
to O letter
  repeat 3 times
    headlights turn on on 1U51
    wait in sec. 0.8
  do
    headlights turn off on 1U51
    wait in sec. 0.3
  wait in sec. 1
```

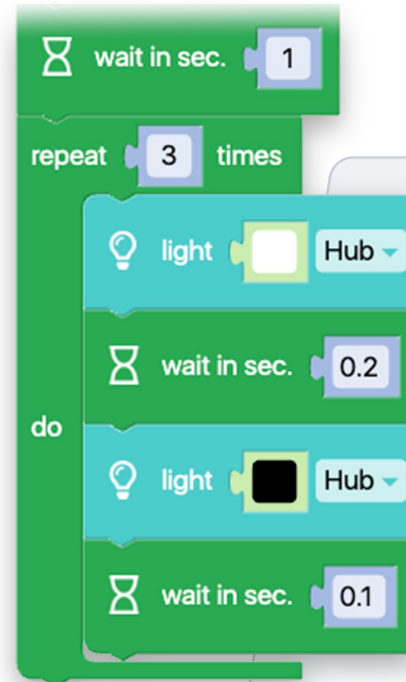
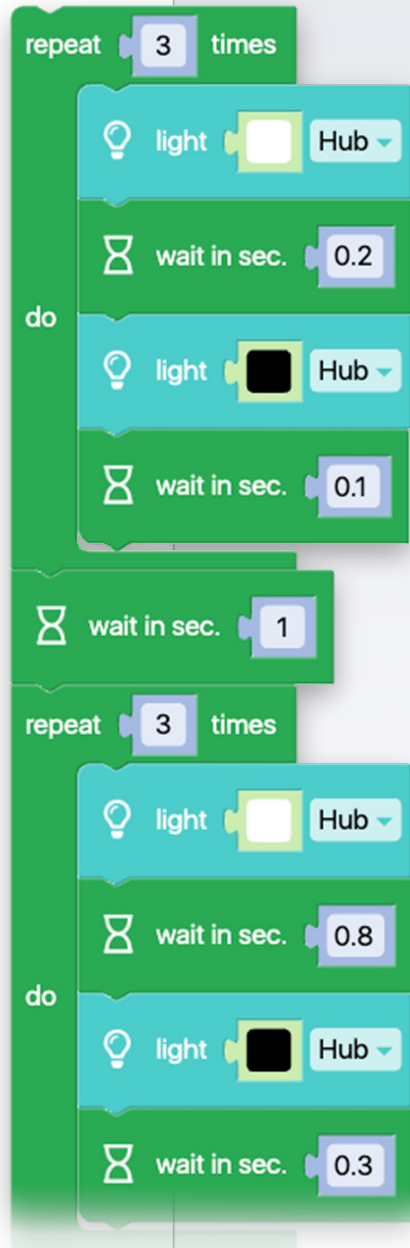
Kod 84: Światła przerywane z użyciem Huba

- Program wykorzystuje Hub do nadawania wyraźnego sygnału SOS, podobnego do alfabetu Morse'a. Zmiana czasu wpłynie na to, jak długo światło Hub będzie włączone lub wyłączone.

OBEJRZYJ WIDEO



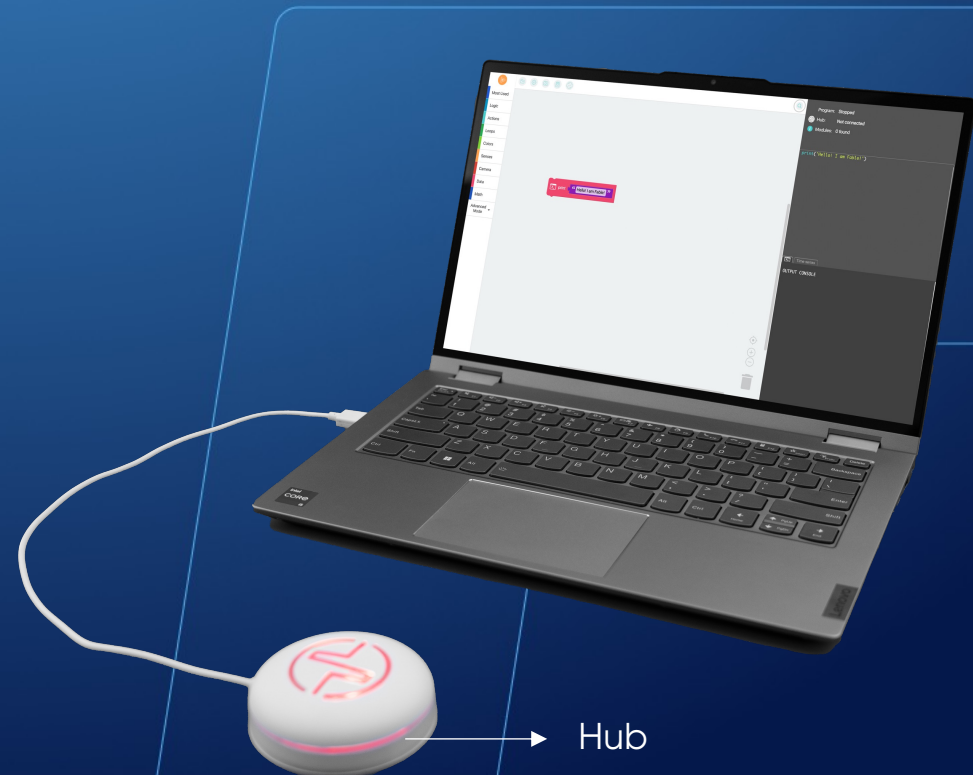
→ Hub



Kod 85: Wyświetlanie w Konsoli

- Dostęp do konsoli wyjściowej można uzyskać, klikając przycisk >_
- Po naciśnięciu przycisku uruchomienia programu, konsola wyświetla tekst "Hello! I am Fable!", który może być zmieniony przez programistę.

OBEJRZYJ WIDEO



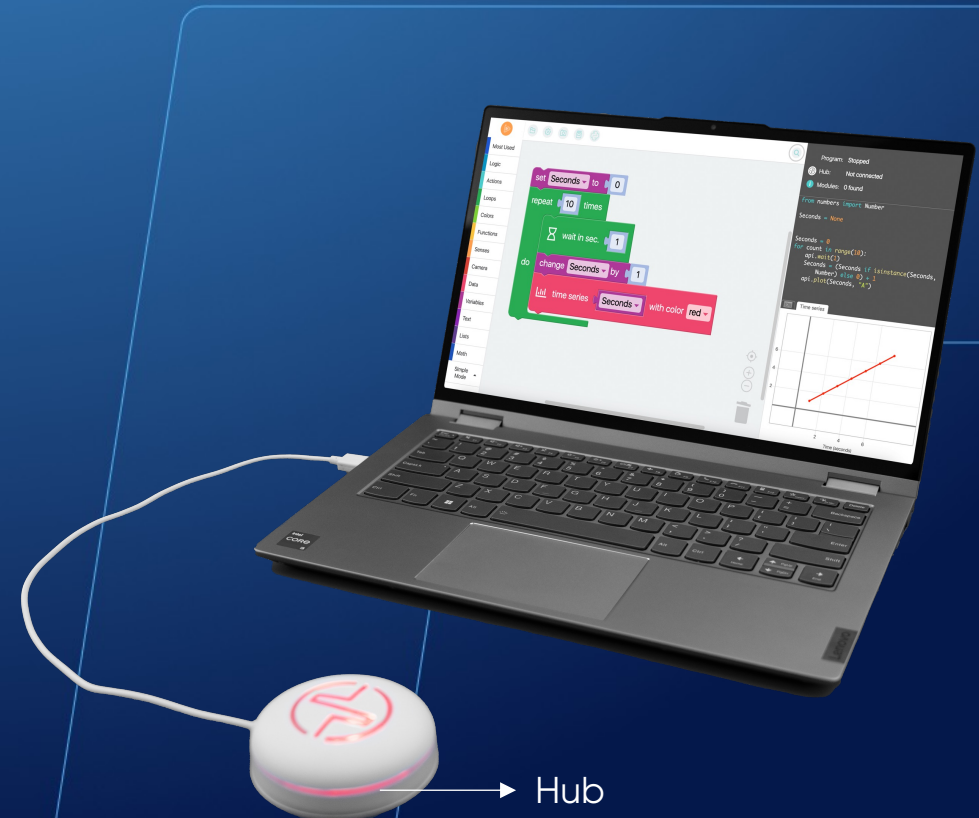
```
>_ print "Hello! I am Fable!"
```

Kod 86:

Wyświetlanie wykresu szeregowego czasu

- Program ustawia wartość zmiennej na 0 i zwiększa ją o 1 co sekundę. Wyświetla czerwoną grafikę z wartością zmiennej i pozwala użytkownikowi na zmianę koloru.

OBEJRZYJ WIDEO



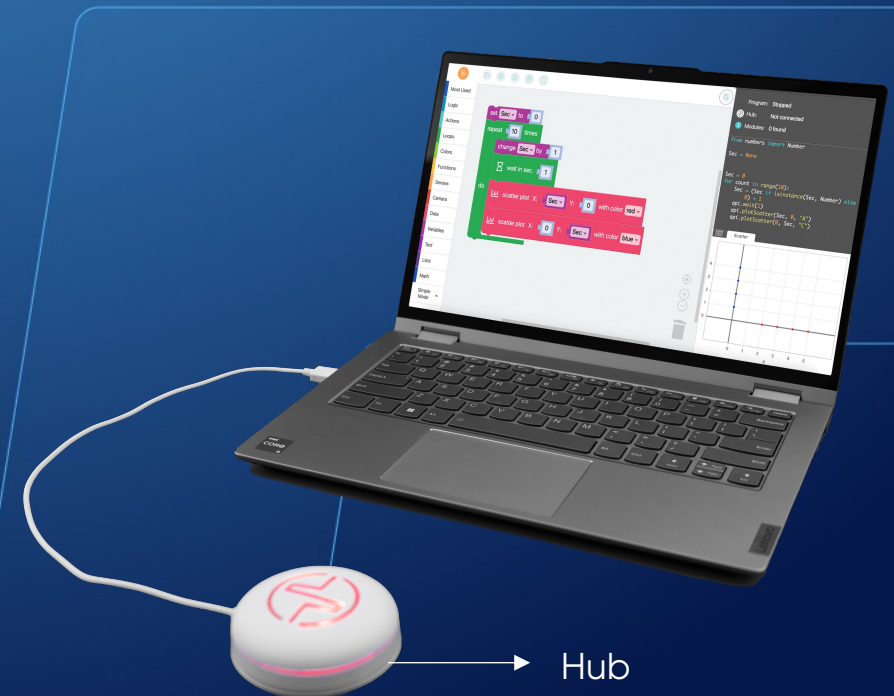
```
set Seconds to 0
repeat 10 times
  wait in sec. 1
  do
    change Seconds by 1
    time series Seconds with color red
```

Kod 87:

Wyświetlanie wykresu punktowego

- Program ustawia zmienną o nazwie "sec" (sekundy) i aktualizuje ją co sekundę. Następnie używa różnych kolorów do wyświetlenia wartości "sec" na osiach X i Y. Kolor niebieski jest używany do pokazania sekund na osi Y, natomiast czerwony pokazuje sekundy na osi X. Tego rodzaju wykres punktowy może być wykorzystywany do pobierania danych z Modułu Joint, Modułu Spin, lub telefonu (z aplikacją Fable Face).

OBEJRZYJ WIDEO

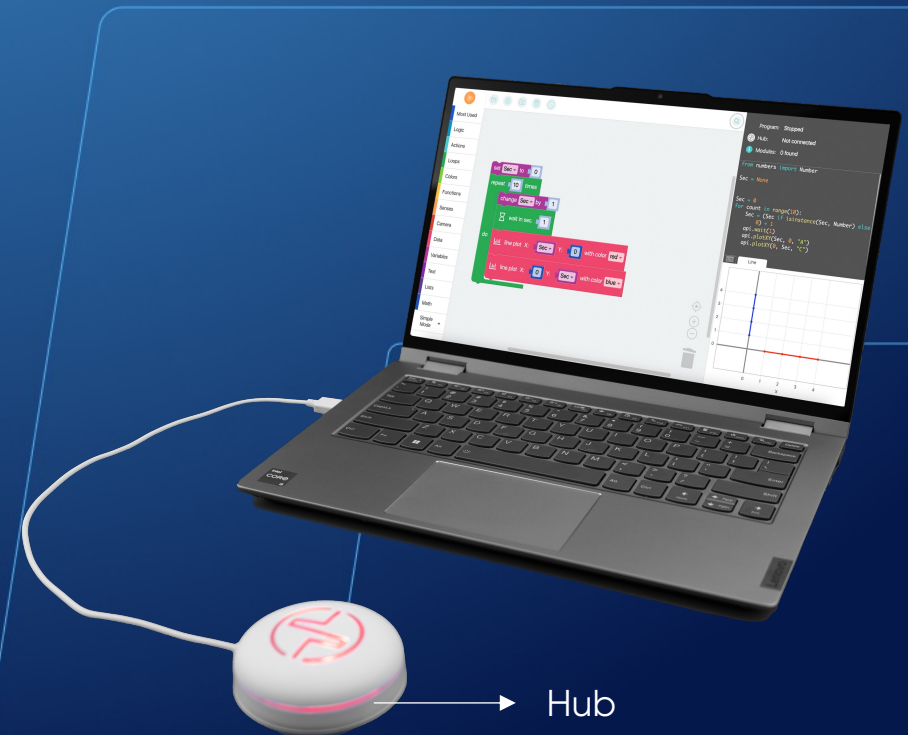



```
set Sec to 0
repeat 10 times
  change Sec by 1
  wait in sec. 1
  do
    scatter plot X: Sec Y: 0 with color red
    scatter plot X: 0 Y: Sec with color blue
```

Kod 88: Wyświetlanie wykresu liniowego

- Program ustawia zmienną "sec" (sekundy) i aktualizuje ją co sekundę. Następnie używa różnych kolorów do wyświetlania wartości zmiennej "sec" na osiach X i Y. Kolor niebieski jest używany do pokazywania sekund na osi X, a czerwony na osi Y. Tego rodzaju wykres liniowy może być wykorzystywany do pobierania danych z Modułu Joint, Modułu Spin, lub telefonu (z aplikacją Fable Face).

OBEJRZYJ WIDEO



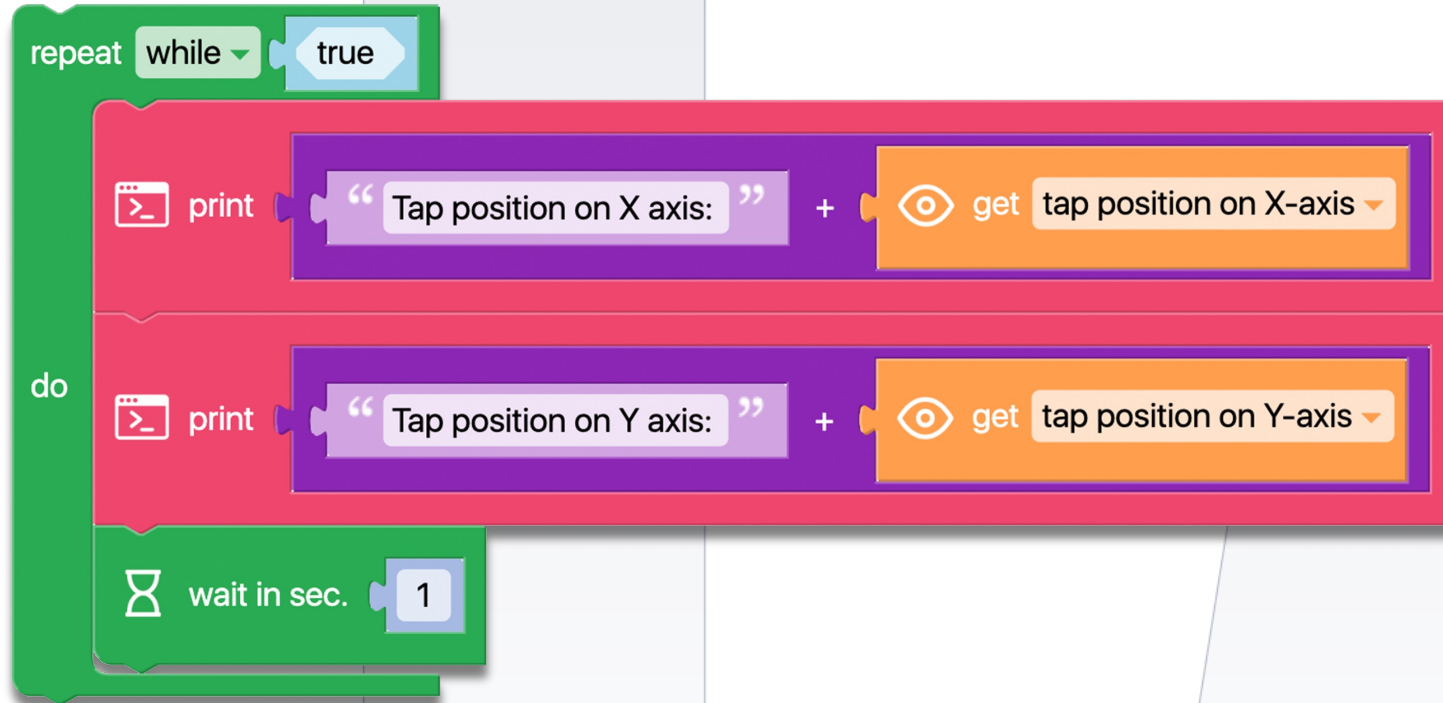
```
set Sec to 0
repeat 10 times
  change Sec by 1
  wait in sec. 1
  do
    line plot X: Sec Y: 0 with color red
    line plot X: 0 Y: Sec with color blue
```

Kod 89: Wyświetlanie pozycji dotknięcia ekranu

- Program co sekundę wyświetla w konsoli współrzędne położenia palca na ekranie telefonu podłączonego do Huba, przedstawiając je w formie numerycznej względem osi ortogonalnych.

OBEJRZYJ WIDEO



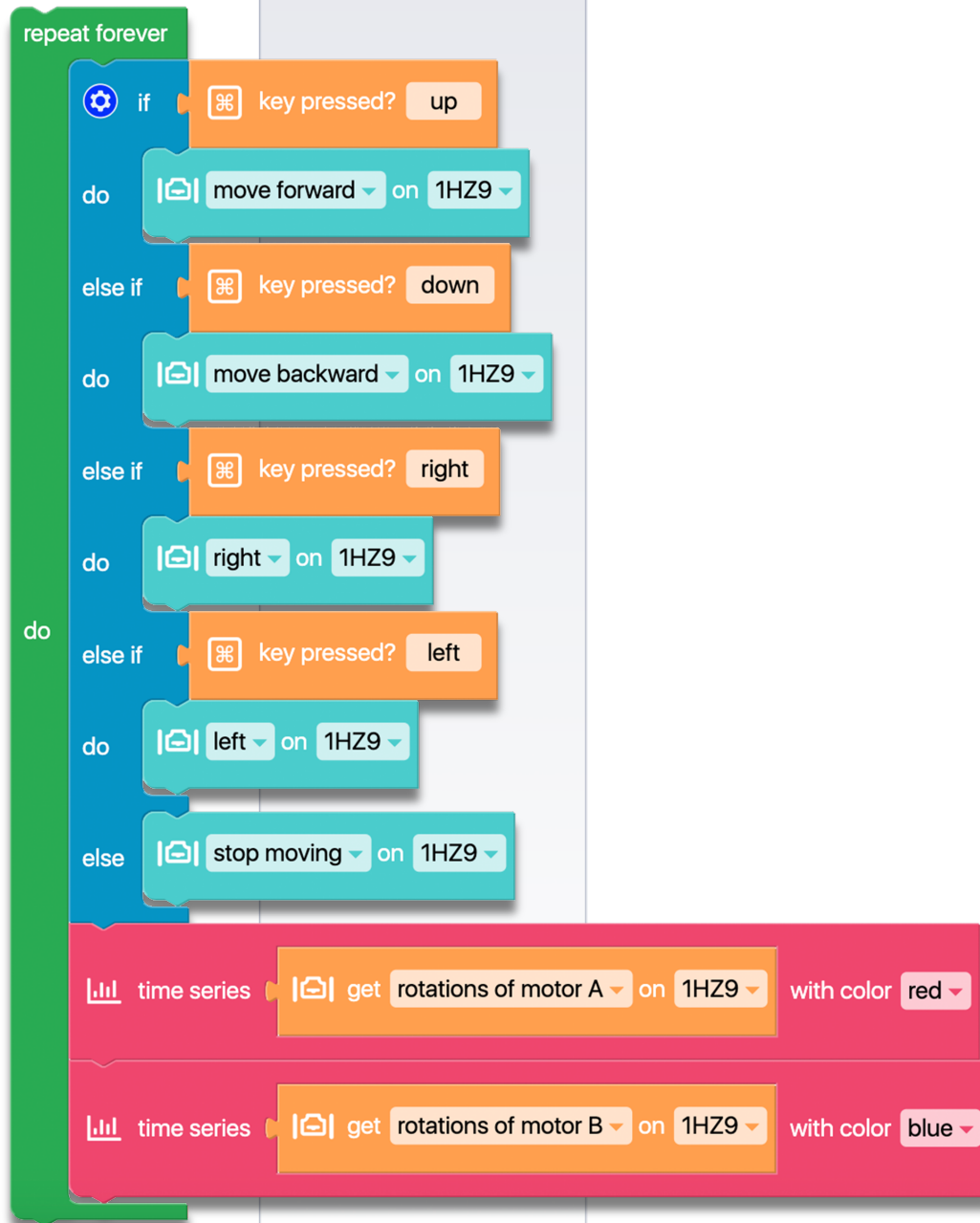


Kod 90: Tworzenie wykresu obrotów Modułu Spin

- Program "Sterowanie Modułem Spin za pomocą klawiszy" kontroluje Moduł Spin za pomocą klawiszy. Dodano dwa nowe polecenia ("Wykres Szeregowy Czasu") które wyświetlają grafikę w Konsoli Wyjściowej. Ten wykres pokazuje obroty dwóch silników, co pozwala analizować z uczniami kierunek ruchu Modułu Spin po zarejestrowaniu danych.

OBEJRZYJ WIDEO



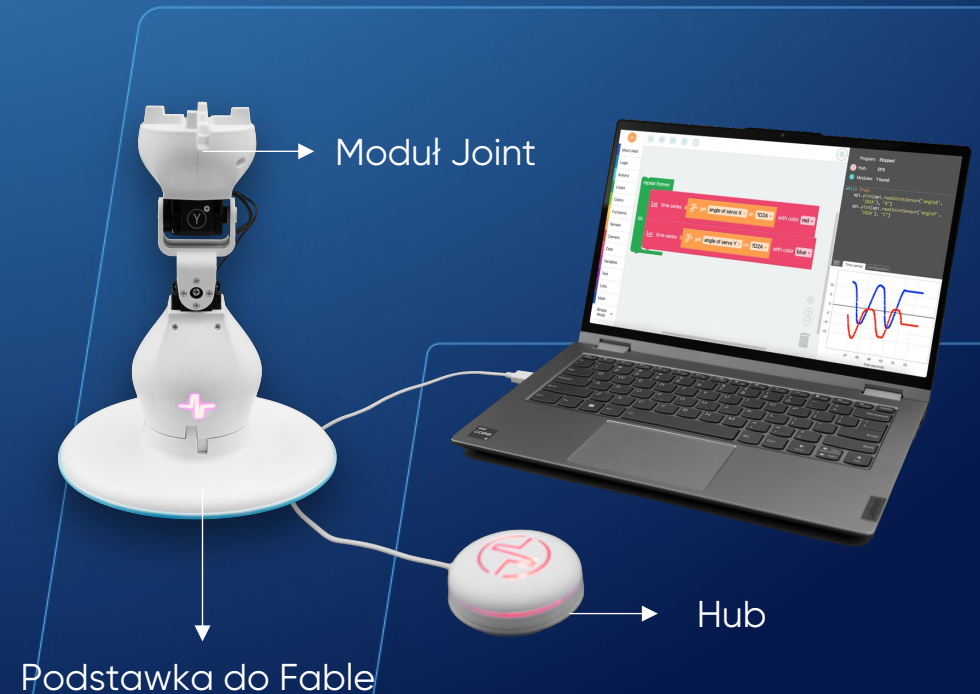


Kod 91:

Wyświetlanie wykresu kąta Modułu Joint

- Program generuje grafikę, która wyświetla kąty Modułu Joint. Silnik X jest reprezentowany kolorem czerwonym, a silnik Y kolorem niebieskim. Po uruchomieniu programu przesuń lekko robota Modułu Joint, aby zaobserwować zmiany odzwierciedlone na grafice.

OBEJRZYJ WIDEO



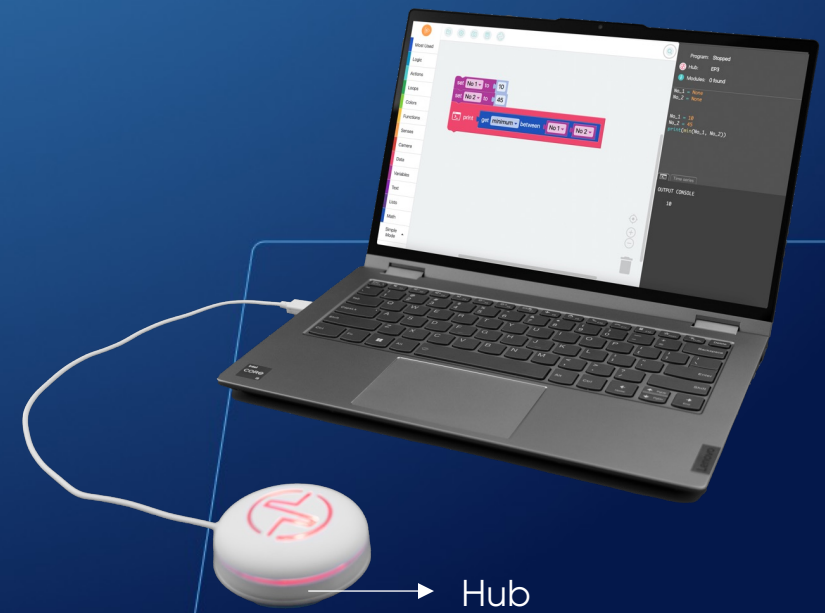

```
repeat forever
do
  time series get angle of servo X on 1D2A with color red
  time series get angle of servo Y on 1D2A with color blue
```

Kod 92:

Porównywanie dwóch zaprogramowanych wartości

- Program przypisuje wartości do dwóch zmiennych, a następnie je porównuje. Te wartości mogą być wprowadzone z klawiatury na początku programu lub pobrane z danych zebranych w programie za pomocą sensorów lub obliczeń. Przy użyciu polecenia „uzyskaj minimum pomiędzy” wraz z poleceniem „drukuj/wyświetl”, program może wyświetlić wartość niższej liczby w konsoli. W przypadku, gdy obie zmienne zawierają tę samą wartość, program wyświetli tę wartość. Aby wyświetlić wartość maksymalną, można użyć tego samego programu, zmieniając jedynie "minimum" na "maximum".

OBEJRZYJ WIDEO



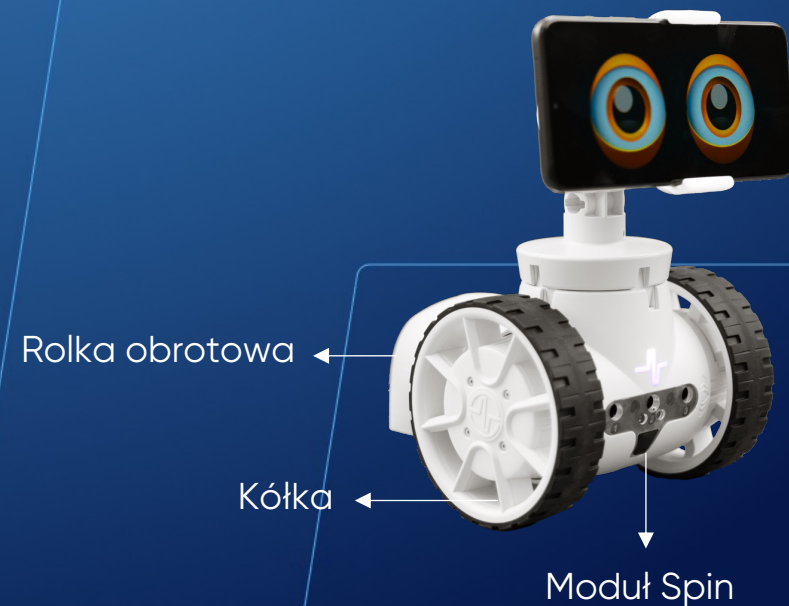
```
set No 1 to 10  
set No 2 to 45  
print get minimum between No 1 No 2
```

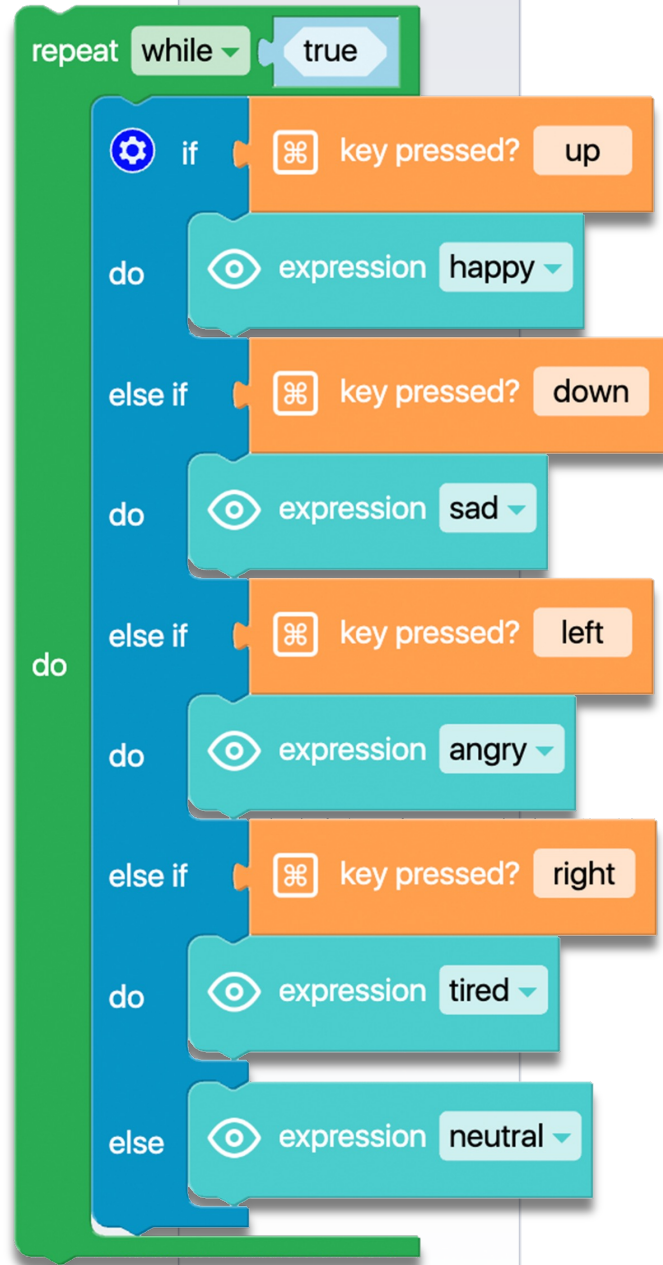
Kod 93:

Ustawianie wyrazów twarzy Fable

- Użytkownik może kontrolować cztery różne wyrazy twarzy – szczęśliwy, smutny, zły i neutralny – poprzez naciskanie klawiszy kierunkowych. Te wyrazy twarzy będą wyświetlane na ekranie telefonu tak długo, jak długo jest wciśnięty odpowiedni klawisz.

OBEJRZYJ WIDEO





Kod 94: Mieszanie światła R, G, B

- Co trzy sekundy zmiennym Czerwony, Zielony i Niebieski przypisywane są losowe wartości od 1 do 100. Wartości te są używane do tworzenia nowego koloru, który jest wyświetlany przez Hub.

OBEJRZYJ WIDEO



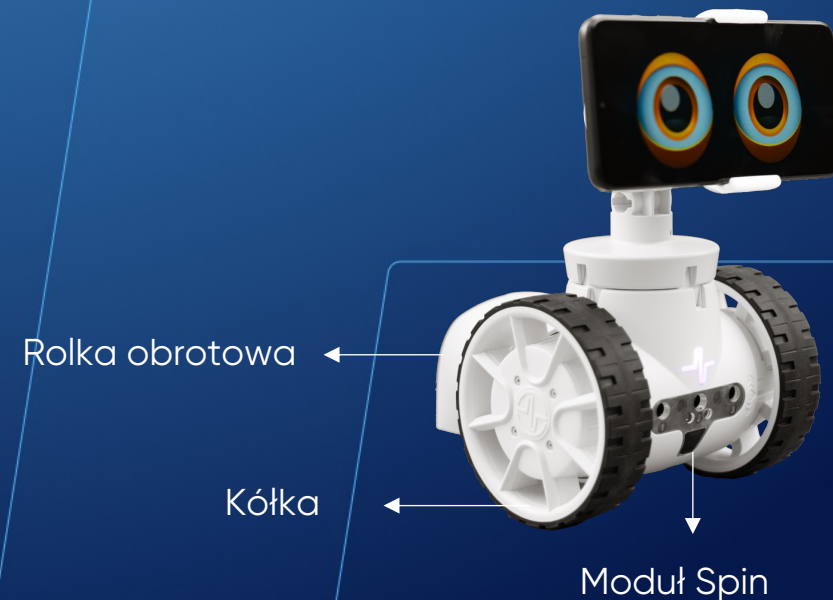
```
repeat while true  
  set Red to random integer from 1 to 100  
  set Green to random integer from 1 to 100  
  set Blue to random integer from 1 to 100  
  do  
    print "Red value is: " + Red  
    print "Green value is: " + Green  
    print "Blue value is: " + Blue  
  light color with red: Red green: Green blue: Blue Hub  
  wait in sec. 3
```

Kod 95:

Ustawianie kolorów tęczy/powiek

- Program generuje różne kolory tęczy i powiek oczu w aplikacji Fable Face, używając losowych liczb między 1 a 100 do tworzenia kombinacji Czerwieni, Zieleni i Niebieskiego. Zmiany kolorów następują co sekundę, a uzyskane wartości RGB są wyświetlane w Konsoli Wyjściowej. Dane w konsoli są przedstawione w czytelny sposób za pomocą polecenia „Wyświetl bez tekstu”, poprzez wstawienie spacji między liniami.

OBEJRZYJ WIDEO




```
repeat while true
  set Iris to color with red: random integer from 1 to 100 green: random integer from 1 to 100 blue: random integer from 1 to 100
  set Eyelids to color with red: random integer from 1 to 100 green: random integer from 1 to 100 blue: random integer from 1 to 100
  print " "
  print "Iris value: " + Iris
  do print "Eyelids value: " + Eyelids
  set color to iris Iris
  set color to eyelids Eyelids
  vibrate
  wait in sec. 1
```

Kod 96: Ustawianie kierunku oczu

- Sekwencja programu wydaje polecenie oczom Fable w aplikacji Fable Face aby przesunęły się do kwadrantu 1 określonego przez ortogonalne osie na ekranie. Wartość odciętej maleje co 0.2 sekundy, podczas gdy wartość rzędnej wzrasta co 0.2 sekundy.

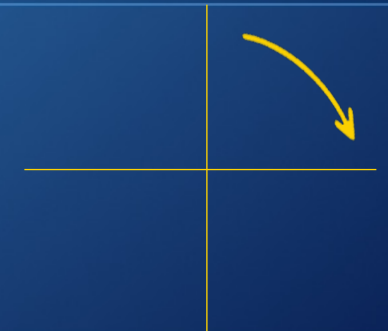
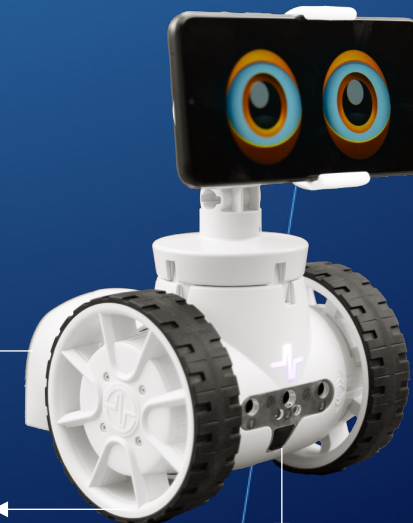
OBEJRZYJ WIDEO



Rolka obrotowa

Kółka

Moduł Spin



```
set X value to 0
set Y value to 90
repeat 9 times
  change X value by 10
  change Y value by -10
  wait in sec. 0.2
  do
    set eyes direction X: X value Y: Y value
    print "X value is: " + X value
    print "Y value is: " + Y value
```

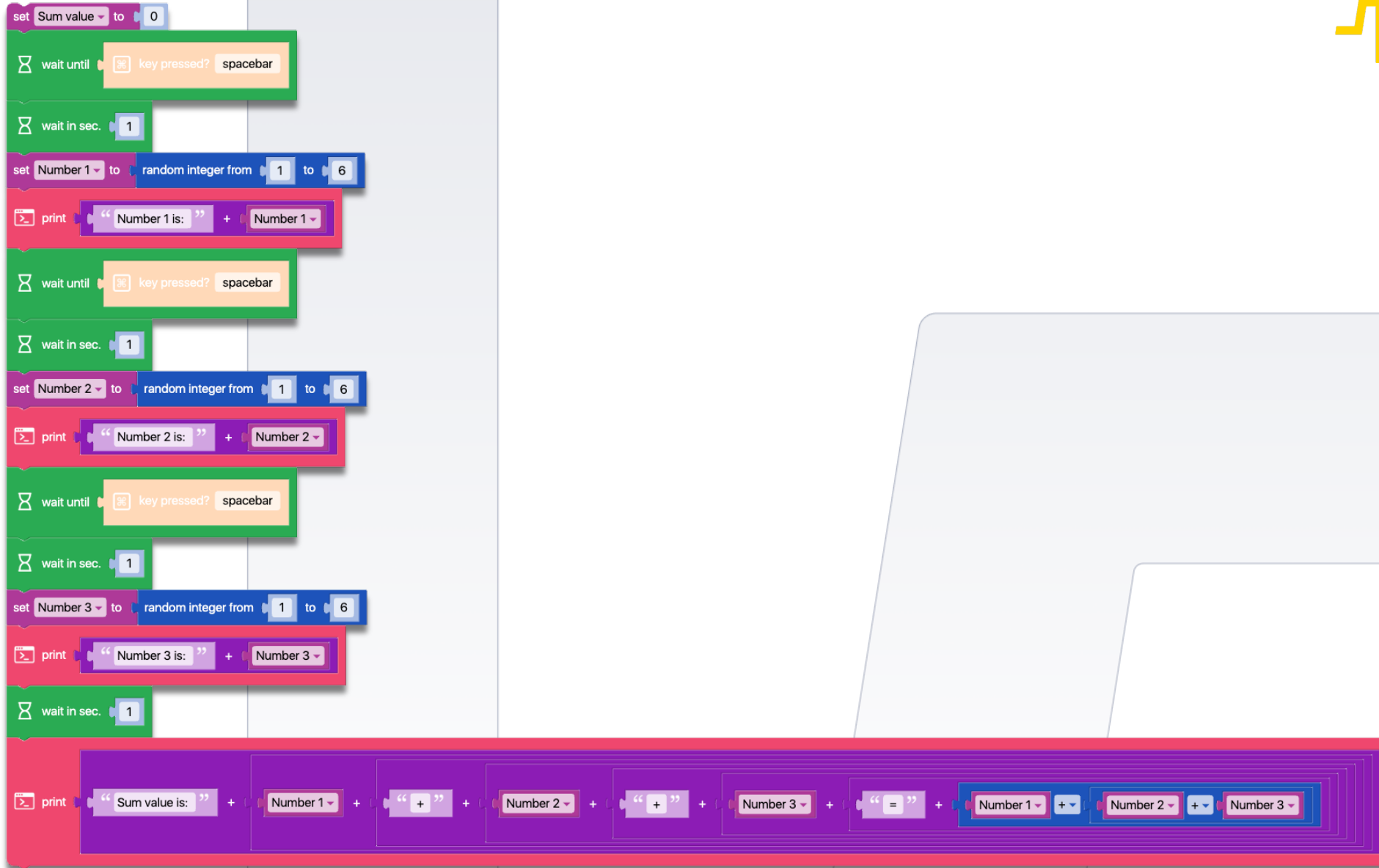
Kod 97:

Wyświetlanie scalonych danych w jednej linii

- Program generuje trzy losowe liczby za każdym razem, gdy naciśnięty zostanie klawisz spacji, i wyświetla je w jednej linii w Konsoli Wyjściowej. Następnie program wyświetla te liczby pojedynczo, oddzielone znakiem plus, po czym wyświetla ich sumę w jednej linii. Ten program służy jako przykład sytuacji, w której wiele danych musi być wyświetlanych w ramach jednej linii tekstu.

OBEJRZYJ WIDEO





Kod 98: Gra w Kości

- Program może funkcjonować jako gra w kości, gdzie gracz otrzymuje trzy rzuty dwiema kośćmi. Jeśli liczby na kościach są takie same, Moduł Spin przesuwają się o czterokrotność wartości pokazanej na kościach. Natomiast jeśli liczby na kościach nie są różne, Moduł Spin przesuwają się o odległość równą sumie dwóch liczb. Po trzech rzutach program ogłasza całkowity wynik, określając gracza, który osiągnął najdalszą odległość z Modułem Spin jako zwycięzcę. Wartości kości, średnie przemieszczenia oraz całkowite wyniki są wyświetlane na konsoli.

OBEJRZYJ WIDEO



Rolka obrotowa

Kółka

Moduł Spin

Hub



```
set Total score to 0
repeat 3 times
  wait until key pressed? spacebar
  wait in sec. 1
  set Dice 1 to random integer from 1 to 6
  set Dice 2 to random integer from 1 to 6
  print "Dice 1 value: " + Dice 1
  print "Dice 2 value: " + Dice 2
  do
    if Dice 1 = Dice 2
      do
        drive Dice 1 x 4 cm with speed: 50 on XEQ
      do
        change Total score by Dice 1 x 4
        print "Intermediate score: " + Total score
    drive Dice 1 + Dice 2 cm with speed: 50 on XEQ
```

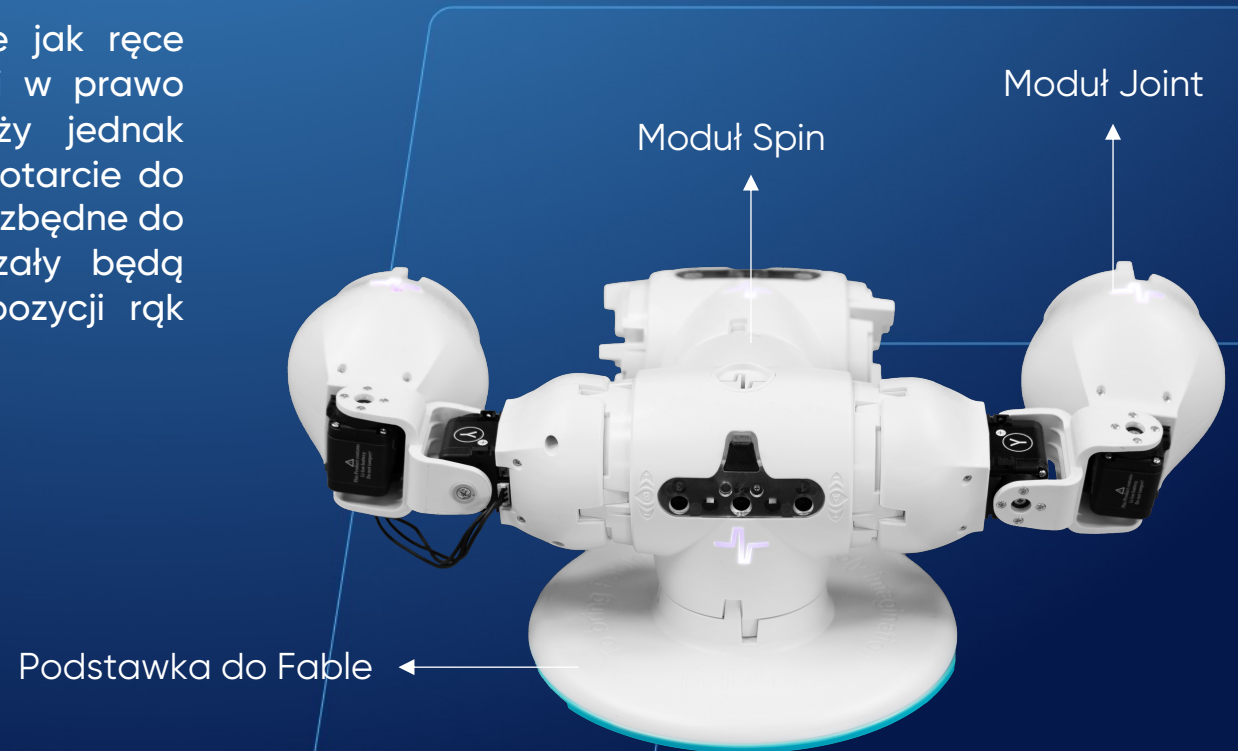


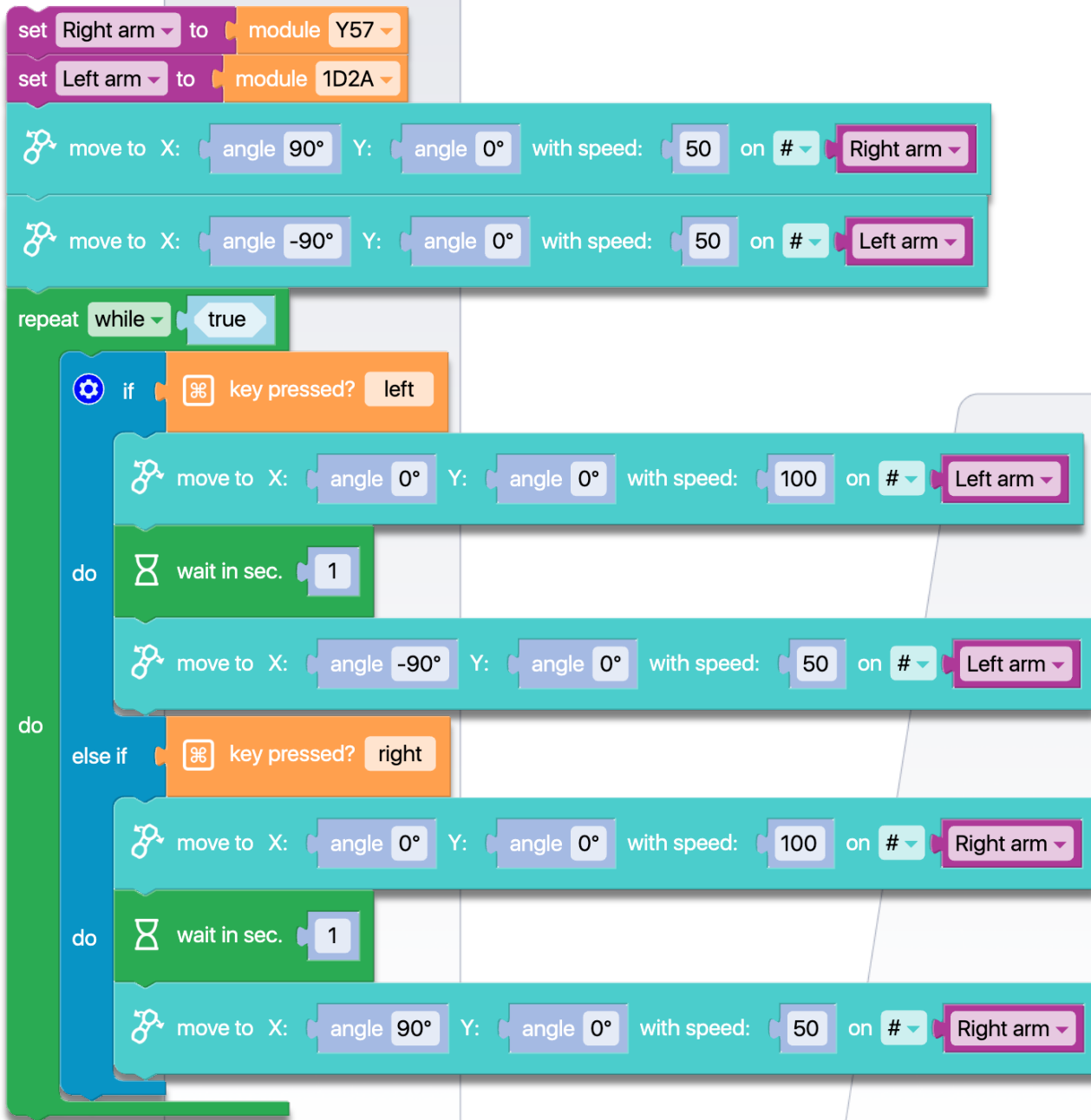
```
else change Total score by Dice 1 + Dice 2
print "Intermediate score: " + Total score
print "Total score: " + Total score
wait in sec. 2
repeat 3 times
  headlights turn on on XEQ
  wait in sec. 0.2
do
  headlights turn off on XEQ
  wait in sec. 0.1
```

Kod 99: Gra w Bramkarza

- Program obsługuje dwa Moduły Joint działające jak ręce bramkarza. Naciśnięcie klawiszy strzałki w lewo i w prawo pozwala na kontrolowanie tych ramion. Należy jednak pamiętać, że ze względu na czas potrzebny na dotarcie do pozycji obronnej, precyzyjne wyczucie czasu jest niezbędne do dokładnego zainicjowania polecenia ruchu. Strzały będą kierowane w lewo i w prawo, co odpowiada pozycji rąk bramkarza.

OBEJRZYJ WIDEO

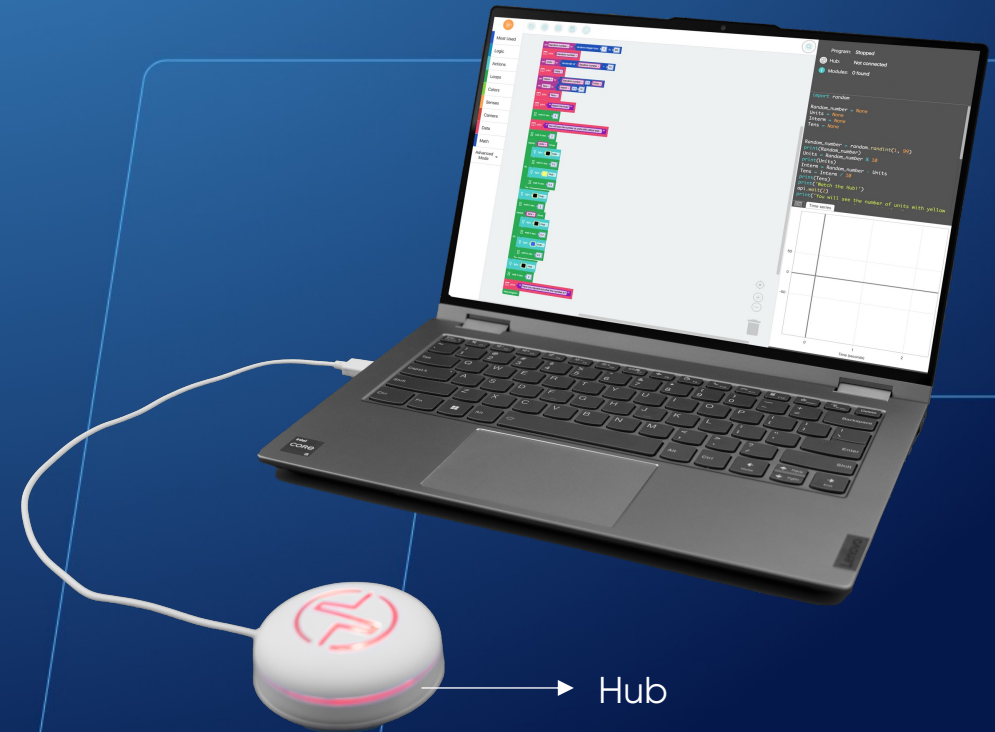




Kod 100: Zgadnij Liczbę

- Program może również działać jako gra. Program losowo wybiera liczbę między 1 a 99 i wyświetla ją na Hubie, podświetlając odpowiednie dziesiątki i jednostki. Na przykład, jeśli liczba to 14, Hub podświetli się cztery razy na żółto i raz na niebiesko. Celem gry jest odgadnięcie liczby wyłącznie poprzez obserwację Huba. Aby zwiększyć trudność, interwały włączania i wyłączania świateł na Hubie mogą być skrócone.

OBEJRZYJ WIDEO



```
set Random number to random integer from 1 to 99
print Random number
set Units to remainder of Random number ÷ 10
print Units
set Intern to Random number - Units
set Tens to Intern ÷ 10
print Tens
print "Watch the Hub!"
wait in sec. 2
print "You will see the number of units with yellow and..."
wait in sec. 2
repeat Units times
  light Hub [black]
  wait in sec. 0.2
do
  light Hub [yellow]
  wait in sec. 0.2
```



```
light Hub [black]
wait in sec. 2
repeat Tens times
  light Hub [black]
  wait in sec. 0.2
do
  light Hub [blue]
  wait in sec. 0.2
light Hub [black]
wait in sec. 2
print "Have you figured out what the number is?"
stop program
```

Podziel się!

Napędzaj pasję naszej społeczności!

Wyślij kody swoich robotów na adres hello@shaperobotics.com
i pozwól radości się szerzyć!