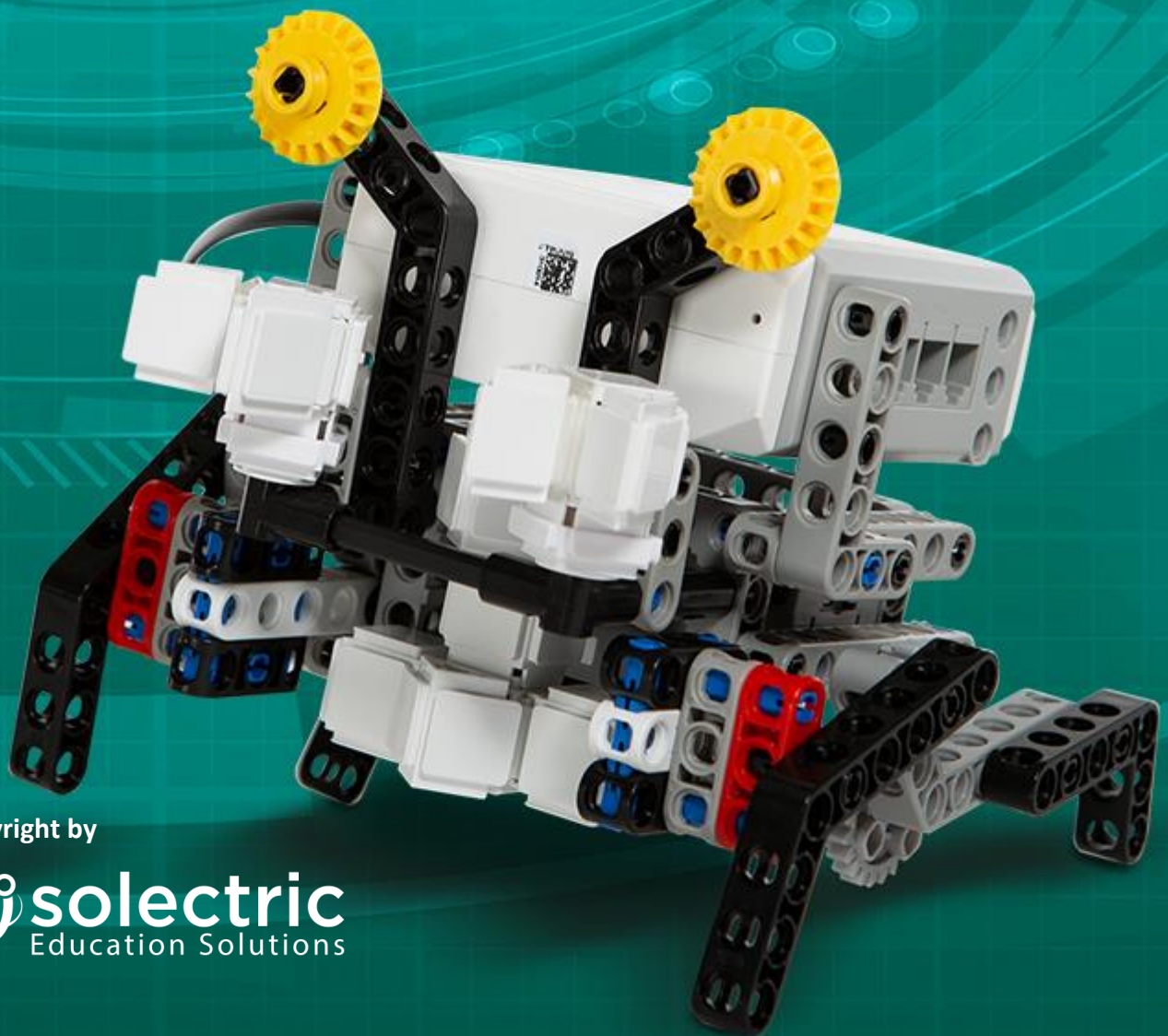


# Abilix

EDUCATIONAL ROBOT

## JĘZYK SCRATCH

Manual użytkownika



Copyright by

 **solectric**  
Education Solutions

## Spis treści:

1.	Funkcje i bloki programu Scratch .....	4
1.1.	Program główny.....	4
1.2.	Biblioteka bloków wykonywania programu.....	4
1.2.1.	Zakładka <i>czynności</i> .....	4
1.2.2.	Zakładka „ <i>Pokaż</i> ” .....	5
1.3.	Biblioteka bloków wskazania i odczytywania wartości czujników.....	7
1.4.	Biblioteka bloków kontroli, pętli, warunków w programie.....	11
1.5.	Biblioteka obsługi wyrażeń matematycznych i logicznych.....	14
2.	Programowanie .....	18
2.1.	Jak napisać program.....	18
2.2.	Proces programowania .....	18
2.3.	Używanie instrukcji oceny warunku (porównywanie wartości).....	21
3.	Jak usunąć blok.....	24
4.	Jak zapisać projekt/program w pamięci robota.....	25

## Język programowania Scratch – podręcznik użytkownika

Poniżej przedstawiamy główny interfejs Krypton Scratch.




Moduł Abilix Scratch jest podzielony na następujące biblioteki - zakładki:

1. Biblioteka bloków wykonywania programu (patrz 1.2):
  - biblioteka bloków sterowania motorem (niebieska zakładka)
  - biblioteka bloków sterowania LED, dźwiękiem i ekranem (różowa zakładka)
2. Biblioteka bloków wskazania i odczytywania wartości czujników (żółta zakładka) (patrz 1.3)
3. Biblioteka bloków kontroli, pętli, warunków w programie (czerwona) (patrz 1.4)
4. Biblioteka obsługi wyrażeń matematycznych i logicznych (zielona) (patrz 1.5)

## 1. Funkcje i bloki programu Scratch

### 1.1 Program główny


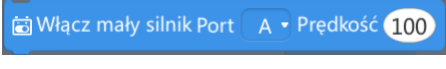

Program główny  jest obszarem służącym do edytowania i nie można go usunąć. Do tego bloku dołączane są kolejne bloki czujników, sterowania motorami etc. Program wykonuje jedynie działania bloków połączone z programem głównym bezpośrednio, bądź pośrednio przez inne bloki.

### 1.2 Biblioteka bloków wykonywania programu

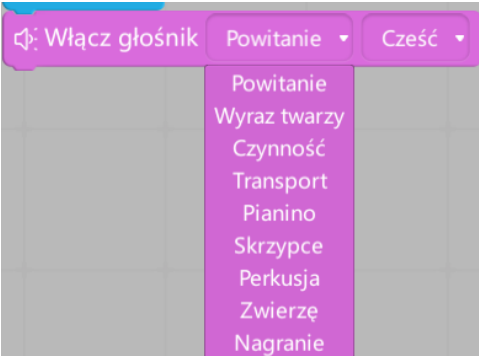

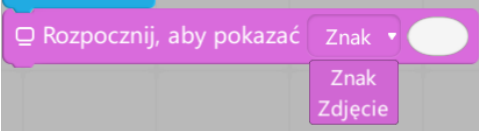

Bloki wykonywania programu znajdują się w dwóch zakładkach.

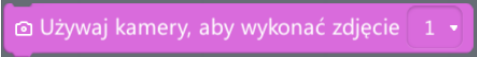
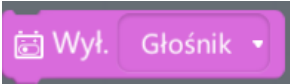
1.2.1. Niebieska zakładka *czynności* odpowiada za sterowanie prędkością i obrotami małego i dużego silnika.

Ta biblioteka zawiera bloki ruchu silnika. Ikony, nazwy i funkcje są następujące:

Nr	Ikona	Nazwa	Funkcja
1	<p>Włącz (duży) silnik, Port <u>A</u>, prędkość <u>100</u></p>  <p>Włącz (mały) silnik, Port <u>A</u>, prędkość <u>100</u></p> 	Ruch silnika	<p>Sterowanie kierunkiem obrotów prędkością dużych i małych silników.</p> <p>Dostępne porty: A, B, C lub D Zakres prędkości: -100 do 100.</p> <p>Jeśli w projekcie podłączone są np. 2 silniki – możesz wybrać 2 porty na raz, jeśli mają działać w tym samym kierunku i z tą samą prędkością. Jeśli każdy z silników ma działać niezależnie, przeciągnij pod blok „główny” 2 bloki np. „włącz mały/duży silnik” i nadaj im parametry niezależnie.</p>
2		Wyłączanie silnika	Wyłączanie silników

1.2.2. Różowa zakładka „Pokaż” odpowiada za pracę głośnika, diody LED i elementów, które można wyświetlać na ekranie sterownika Krypton.



	<p>Głośnik</p>	<p>Sterowanie robotem pod kątem odtwarzania dźwięków.</p> <p>Na przykład, wypowiedanie słowa <i>Hi!</i> (Cześć), dźwięki zwierząt, gama muzyczna.</p> <p>Uwaga:</p> <p>Wybranie w pierwszym pustym polu opcji „nagranie” pozwala odtworzyć nagrane przez siebie dźwięki. Podczas odtwarzania nagrania: wartość od 1 do 10 w drugim pustym polu odpowiada nagranej nazwie pliku.</p>
	<p>LED</p>	<p>Sterowanie kolorem wskaźnika świetlnego robota (zielony, czerwony i niebieski), znajdującego się wokół przycisku STARTU.</p>
	<p>Wyświetlacz</p>	<p>Sterowanie wyświetlanymi znakami (wpisanymi z klawiatury), zdjęciami i wartościami zwracanymi z czujników.</p> <p>Uwaga: podczas wyświetlania zdjęć: wartość od 1 do 10 w drugim pustym polu odpowiada zapisanej nazwie zdjęć.</p>
	<p>Mikrofon</p>	<p>Uruchamianie mikrofonu w celu przeprowadzenia nagrania.</p> <p>Pierwsze puste pole: 1 do 10 odpowiada nazwie pod którą nagranie zostanie zapisane.</p>


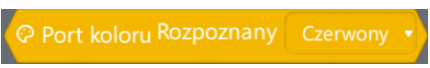
		<p>Drugie puste pole: określa czas nagrywania. Sterowanie dźwiękiem jest dostępne w interfejsie „Multimedia” (multimedia)--&gt; „Record” (Nagrywanie).</p>
	<p>Zdjęcie</p>	<p>Wykonywanie zdjęć za pomocą czujnika z kamerą.</p> <p>Czujnik z kamerą musi być podłączony do portu CAM.</p> <p>Wartość od 1 do 10 w pustym polu odpowiada nazwom zdjęć, pod którą zdjęcie zostanie zapisane. Zdjęcia są przechowywane w interfejsie „Multimedia” (Multimedia) --&gt; „Photo” (Zdjęcie).</p>
	<p>Wyłączanie</p>	<p>Wyłączanie pojedynczego bloku wykonywania programu (np. wyłączanie głośnika, diody LED, wyświetlacza.)</p>

### 1.3. Biblioteka bloków wskazania i odczytywania wartości czujników (żółta zakładka)


Biblioteka bloku czujnika dotyczy czujników, które są podłączane do portów sterownika przy użyciu przewodów połączeniowych.






Ikony, nazwy i funkcje tych bloków są następujące:

Nr	Ikona	Nazwa	Funkcja
1		Czujnik ultradźwiękowy Wykrywanie przeszkody	Wykrywanie przeszkody w odległości mniejszej niż 150 mm. Jeżeli przeszkoda zostanie wykryta w ustawionej odległości czujnik zwróci wartość 1 (logiczne: YES). Brak przeszkody w ustalonej odległości to wartość 0.  Wybór portu: „Automatyczny” lub 1~7.  Jeżeli wybrana zostanie opcja wyboru automatycznego, sterownik będzie pobierał odczyt z pierwszego czujnika ultradźwiękowego. Na przykład, jeżeli do portów 1 i 3 podłączone są czujniki ultradźwiękowe, sterownik będzie automatycznie pobierał wartość zwracaną z portu 1 i tylko z tego portu będzie pobierał informacje.
2		Czujnik ultradźwiękowy Zmierz odległość	Pobiera wartość zwracaną z czujnika ultradźwiękowego.  Im większa odległość, tym większa wartość zwracana przez czujnik. Zakres: 50 mm~1500 mm.

			<p>Wybór portu: „Automatyczny” lub 1~7.</p> <p>Jeżeli wybrana zostanie opcja wyboru automatycznego, sterownik będzie pobierał odczyt z pierwszego czujnika ultradźwiękowego. Na przykład, jeżeli do portów 1 i 3 podłączone są czujniki ultradźwiękowe, sterownik automatycznie będzie pobierał tylko wartość zwracaną z portu 1.</p>
3		Czujnik kolizji	<p>Pobiera wartość zwracaną przez czujnik kolizji.</p> <p>Po naciśnięciu czujnika kolizji, czyli zetknięciu go z innym przedmiotem, zwracana wartość z czujnika wynosi 1. Gdy czujnik nie jest dotknięty, wartość odczytana przez sterownik będzie wynosiła 0.</p> <p>Wybór portu: „Automatyczny” lub 1~7.</p> <p>Jeżeli wybrana zostanie opcja wyboru automatycznego, sterownik będzie pobierał odczyt z pierwszego czujnika kolizji. Na przykład, jeżeli do portów 1 i 3 podłączone są czujniki kolizji, sterownik będzie automatycznie pobierał wartość zwracaną tylko z portu 1.</p>
4		Czujnik kolorów	<p>Sprawdza kolor zgodnie z zaprogramowaną wartością progową.</p>



			<p>Rozpoznaje pięć kolorów: czerwony, żółty, zielony, niebieski i biały.</p> <p>0- Jeżeli czujnik wykryje kolor czerwony</p> <p>1- Jeżeli czujnik wykryje kolor żółty</p> <p>2- Jeżeli czujnik wykryje kolor zielony</p> <p>3- Jeżeli czujnik wykryje kolor niebieski</p> <p>4- Jeżeli czujnik wykryje kolor biały.</p>
5		<p>Czujnik skali szarości</p>	<p>Odczytuje wartość zwracaną przez czujnik skali szarości.</p> <p>Wartości dla koloru białego są małe (ok 200-300), natomiast dla pozostałych kolorów są większe. Maksymalna wartość zwrotna może być równa 4000, (czerni).</p> <p>Wybór portu: „Automatyczny” lub 1~7.</p> <p>Jeżeli wybrana zostanie opcja wyboru automatycznego, sterownik będzie pobierał odczyt z pierwszego czujnika szarości. Na przykład, jeżeli do portów 1 i 3 podłączone są czujniki skali szarości, sterownik będzie automatycznie pobierał wartość zwracaną tylko z portu 1.</p>

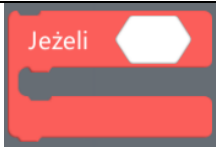
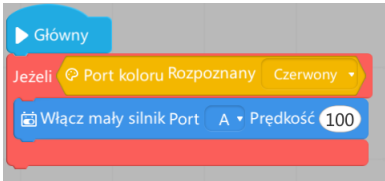
6		Kompas	<p>Umożliwia pobieranie wartości kąta z kompasu (w górnej części sterownika). Zakres zwracanych wartości wynosi 0-359 stopni. Wartości 0 i 360 stopni dotyczą północy geograficznej; 90 stopni wschodu geograficznego; 180 stopni południa geograficznego a 270 stopni wschodu geograficznego.</p> <p>Przed rozpoczęciem używania kompasu, należy go skalibrować.</p>
7		Kalibracja kompasu	<p>Kalibracja jest możliwa po pobraniu programu do sterownika. Po włączeniu programu kompas należy kalibrować zgodnie z instrukcjami wyświetlanymi na ekranie sterownika.</p>
8		Żyroskop	<p>Umożliwia sprawdzenie za pomocą wbudowanego żyroskopu stopień wychylenia sterownika względem zapisanego ustawienia.</p>
9		Resetowanie zegara	<p>Resetuje czas systemowy na sterowniku, dzięki czemu algorytmy programu mogą pracować przez określony czas (nadrzędny)</p>
10		Czas systemowy	<p>Pokazuje wartość czasu od chwili rozpoczęcia operacji (bądź resetowania czasu) do chwili obecnej. Jednostka: sekunda.</p>

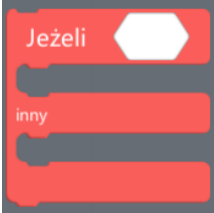
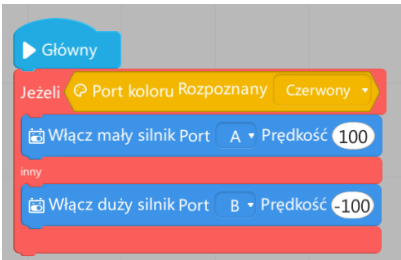

#### 1.4. Biblioteka bloków kontroli, pętli, warunków w programie (czerwona)

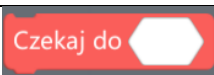
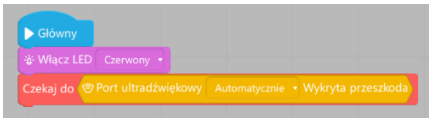
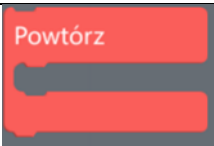
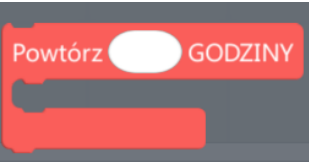
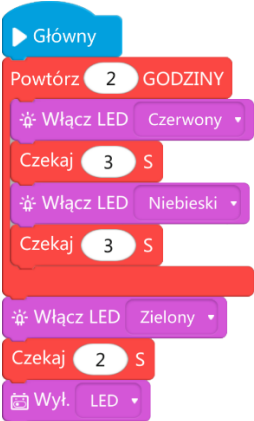
W programie dostępne są dwie metody pobierania wartości zwracanych z czujników: zapisywanie i porównywanie, gdzie porównywanie ma charakter nadrzędny. Dla oceny warunku moduł Abilix Scratch oferuje cztery schematy blokowe:

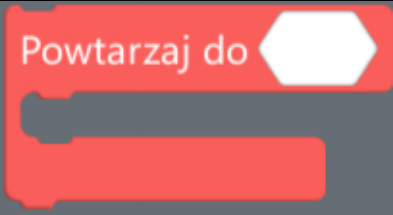
- wyrażenie POWTARZAJ...
- wyrażenie JEŻELI ...
- wyrażenie W PRZECIWNYM RAZIE,
- wyrażenie DLA
- i wyrażenie CZEKAJ. Wszystkie wyrażenia są dostępne w bibliotece.

Ikony, nazwy i funkcje przedstawiono w tabeli poniżej:

Nr	Ikona	Nazwa	Funkcja
1	 <p>„Jeśli warunek jest spełniony, wykonaj akcję. Jeśli nie to kontynuuj program”</p>  <p>PRZYKŁAD: jeżeli port koloru wykryje kolor czerwony → włącz mały silnik z portu A z prędkością 100</p>	Ocena warunku	<p>Wyrażenie JEŻELI (warunek) w języku C. Wartość jest warunkiem ustawianym przez użytkownika. Jeżeli warunek zostanie spełniony nastąpi uruchomienie algorytmu zapisanego wewnątrz warunku.</p> <p>W przeciwnym razie program przejdzie do wykonywania kolejnych operacji (odczyta kolejne bloki)</p>

<p>2</p>	 <p>„Jeśli warunek jest spełniony wykonaj zadanie A ..... jeżeli nie jest wykonaj akcję B”</p>  <p>Np. PRZYKŁAD: jeżeli port koloru wykryje kolor czerwony → włącz mały silnik z portu A z prędkością 100 → w przeciwnym razie włącz duży silnik z portu B z prędkością -100</p>	<p>Ocena warunku</p>	<p>Wyrażenie JEŻELI (warunek) .... W PRZECIWNYM RAZIE ... w języku C.</p> <p>Jeżeli warunek zostanie spełniony nastąpi uruchomienie algorytmu zapisanego wewnątrz warunku. W przeciwnym razie program wykona algorytm przypisany do „inny” czyli sytuacji kiedy warunek nie jest spełniony. Następnie sterownik przejdzie do wykonywania kolejnych operacji programu(odczyta kolejne bloki)</p>
<p>3</p>	 <p>„Czekaj X sekund”</p>	<p>Czekaj</p>	<p>Wyrażenie CZEKAJ (wartość) w języku C. Wartość jest czasem oczekiwania wyrażonym w sekundach (s). Służy do przerywania, bądź zatrzymania programu na czas X sekund, a następnie wykonywania następnego wyrażenia.</p>





<p>4</p>	 <p>„Czekaj do....”</p> <p>Np. włącz czerwony LED niech świeci aż port ultradźwiękowy wykryje przeszkodę</p> 	<p>Blok Pauzy warunkowej</p>	<p>Wyrażenie GDY (warunek) w języku C. Wartość jest warunkiem ustawianym przez użytkownika. Służy do zatrzymania programu do czasu aż wskazany warunek zostanie spełniony. Gdy warunek jest spełniony, sterownik przechodzi do wykonania kolejnych bloków programu.</p>
<p>5</p>	 <p>„Powtarzaj w nieskończoność do zakończenia programu”</p>	<p>Pętla</p>	<p>Wyrażenie POWTÓRZ (prawda) w języku C. Nie posiada wartości. Służy do powtarzania odgałęzionego wyrażenia do momentu zakończenia programu.</p>
<p>6</p>	 <p>„Powtarzaj X razy,,</p> 	<p>Pętla wielokrotna</p>	<p>Powtórz X razy. Dla wyrażen w języku C. Wartością jest liczba pętli. Służy do powtarzania instrukcji w pętli przez określoną ilość razy.</p> <p>Po uruchomieniu programu dioda LED zaświeci na czerwono, odczeka świecąc 3 sekundy i zmieni kolor na niebieski, znowu na 3 sekundy. Koniec pętli ale ma ona być powtórzony 2 razy (2 godziny-błąd tłumaczenia) – powtórzy zatem świecenie diod LED niebieskiej i czerwonej drugi raz a następnie wyjdzie z pętli i przejdzie dalej do działania programu – zaświeci LED na kolor zielony, odczeka świecąc 2 sekundy i je wyłączy</p>

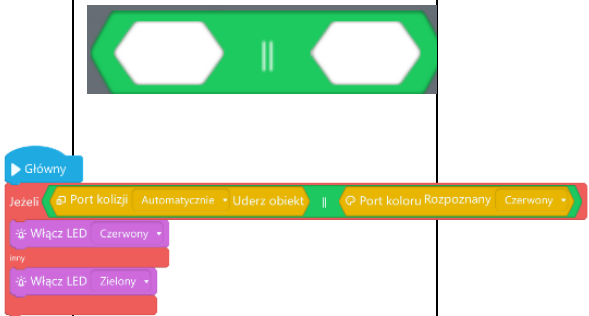
7	 <p>"Powtarzaj aż do momentu, gdy podany warunek będzie spełniony"</p>	Pętla warunkowa	Służy do powtarzania wyrażenia w pętli, do czasu spełnienia ustalonego warunku. Gdy warunek jest spełniony, sterownik przechodzi do wykonywania kolejnych algorytmów.
---	---	-----------------	---

### 1.5. Biblioteka obsługi wyrażeń matematycznych i logicznych (zielona)

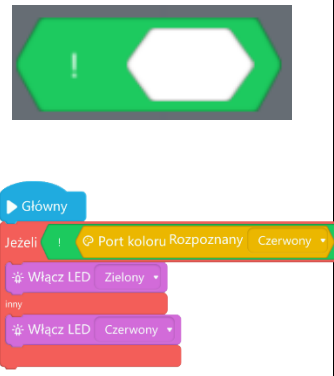
Ta biblioteka oferuje różne metody obliczania zmiennych i stałych. Ikony, nazwy i zasadę działania przedstawiono w tabeli poniżej:

Nr	Ikona	Nazwa	Funkcja
1		Blok operacji matematycznych	Działania $+$ , $-$ , $\times$ , $\div$ w języku C. Algorytm ten można zagnieżdżać (tzn. umieszczać wewnątrz niej inne bloki) w celu uzyskania obliczeń łącznych.
2		Blok operacji matematycznych	Warunek odpowiadający za sprawdzenie czy pierwsza wartość jest mniejsza od drugiej  Np. Jeżeli czujnik odległości wykryje przeszkodę w odległości mniejszej niż 40cm to LED świeci na czerwono. W innym wypadku na zielono.

3		Blok operacji matematycznych	Warunek odpowiadający za sprawdzenie czy dwie wartości są równe															
4		Blok operacji matematycznych	Warunek odpowiadający za sprawdzenie czy pierwsza wartość jest większa od drugiej															
	 	„i”	<p>Blok wartości logicznej „i” – wartość zwrócona będzie równa 1 wtedy i tylko wtedy gdy obie wartości będą prawdziwe.</p> <p>Przykład:</p> <p>-Jeżeli czujnik dotyku zostanie wciśnięty(1) LUB port koloru rozpozna kolor czerwony(1) wtedy = (1) = dioda LED zaświeci na czerwono.</p> <p>Czyli:</p> <table border="1" data-bbox="893 1366 1372 1926"> <thead> <tr> <th>Czujnik dotyku wciśnięty</th> <th>Czujnik koloru wykrywa czerwony</th> <th>WYNIK</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Czujnik dotyku wciśnięty	Czujnik koloru wykrywa czerwony	WYNIK	1	1	1	0	1	1	1	0	1	0	0	0
Czujnik dotyku wciśnięty	Czujnik koloru wykrywa czerwony	WYNIK																
1	1	1																
0	1	1																
1	0	1																
0	0	0																

		<p>or</p>	<p>Blok wartości logicznej „i” – wartość zwrócona będzie równa 1 jeżeli jedna lub druga z wartości będą prawdziwe.</p> <p>Przykład: Jeżeli czujnik dotyku zostanie wciśnięty(1) i port koloru rozpozna kolor czerwony(1) wtedy = (1) = dioda LED zaświeci na czerwono. W innym wypadku:</p> <ul style="list-style-type: none"> <li>- czujnik nie wciśnięty(0) i kolor czerwony (1) = 0</li> <li>- czujnik wciśnięty(1) ale kolor inny (0) = (0)</li> <li>- czujnik nie wciśnięty(0) i kolor inny (0) = (0)</li> </ul> <table border="1" data-bbox="890 1227 1375 1805"> <thead> <tr> <th>Czujnik dotyku wciśnięty</th> <th>Czujnik koloru wykrywa czerwony</th> <th>WYNIK</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Czujnik dotyku wciśnięty	Czujnik koloru wykrywa czerwony	WYNIK	1	1	1	0	1	0	1	0	0	0	0	0
Czujnik dotyku wciśnięty	Czujnik koloru wykrywa czerwony	WYNIK																
1	1	1																
0	1	0																
1	0	0																
0	0	0																



		<p>not</p>	<p>Blok negacji warunku</p> <p>„jeżeli czujnik dotyku nie jest wciśnięty, świeć na czerwono. W przeciwnym razie na niebiesko”</p> <p>Przykład: Jeżeli czujnik koloru NIE rozpozna czerwonego koloru wtedy LED świeci na zielono. W przeciwnym razie (kiedy rozpozna czerwony) świeci na czerwono.</p> <p>Uwaga: „nie rozpozna” oznacza „każdy inny niż czerwony, oraz brak koloru”</p>
--	---	------------	--

## Rozdział 2

## Programowanie

### 2.1 Jak napisać program

Punktem rozpoczęcia programu jest „Program główny”



Następnie w określonej kolejności, zaplanowanej przez użytkownika, przeciągane są poszczególne bloki i umieszczane jeden pod drugim pod programem głównym aż do zakończenia programu.

Blok przeciągany do obszaru edytowania jest wpisywany w program tylko jeżeli jest przyłączony bezpośrednio lub pośrednio (przez inne bloki) do bloku „Główny”. Bloki, które znajdują się w obszarze edytowania ale nie są podłączone bezpośrednio lub pośrednio z blokiem głównym nie włączają się do programu i nie ma możliwości ich wgrania na sterownik.

### 2.2. Proces programowania jest następujący:

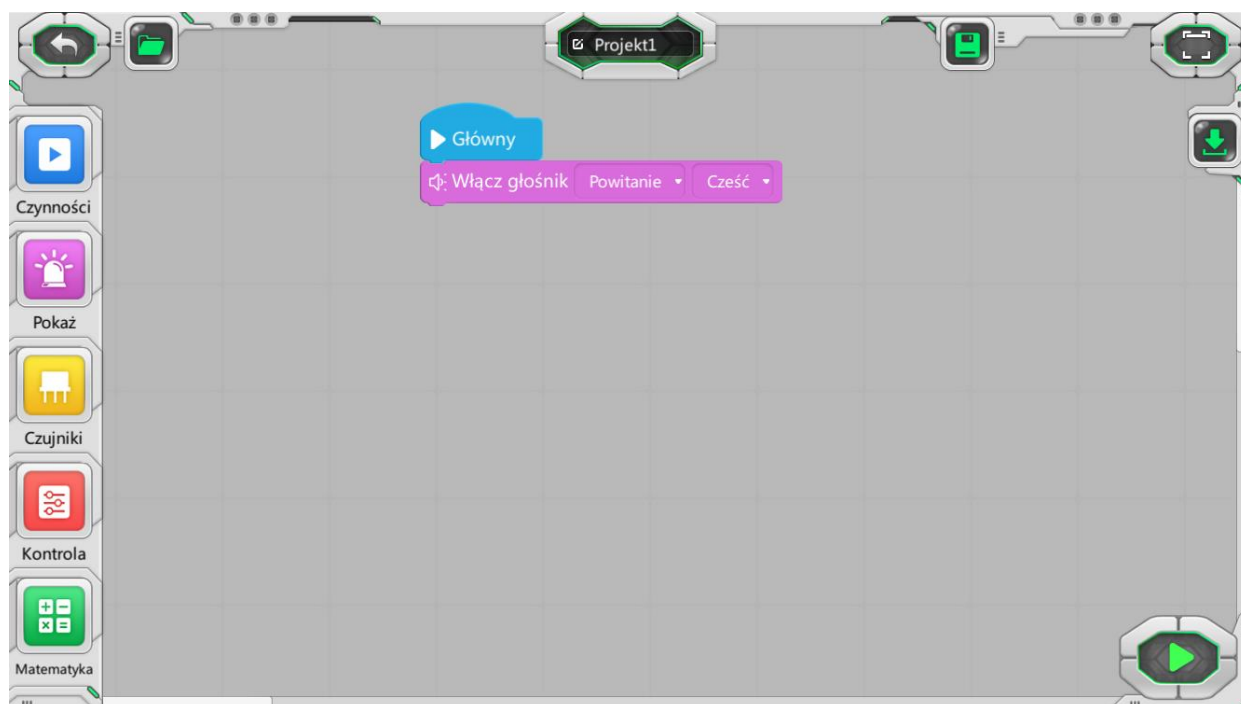
1. Wybierz wymagany blok z biblioteki bloków i przeciągnij go do obszaru edytowania po prawej stronie.
2. Połącz blok z łańcuchem programu głównego.
3. Dodaj lub zmień wartości dla bloku.

Poszczególne kroki zostały przedstawione na poniższych przykładach:

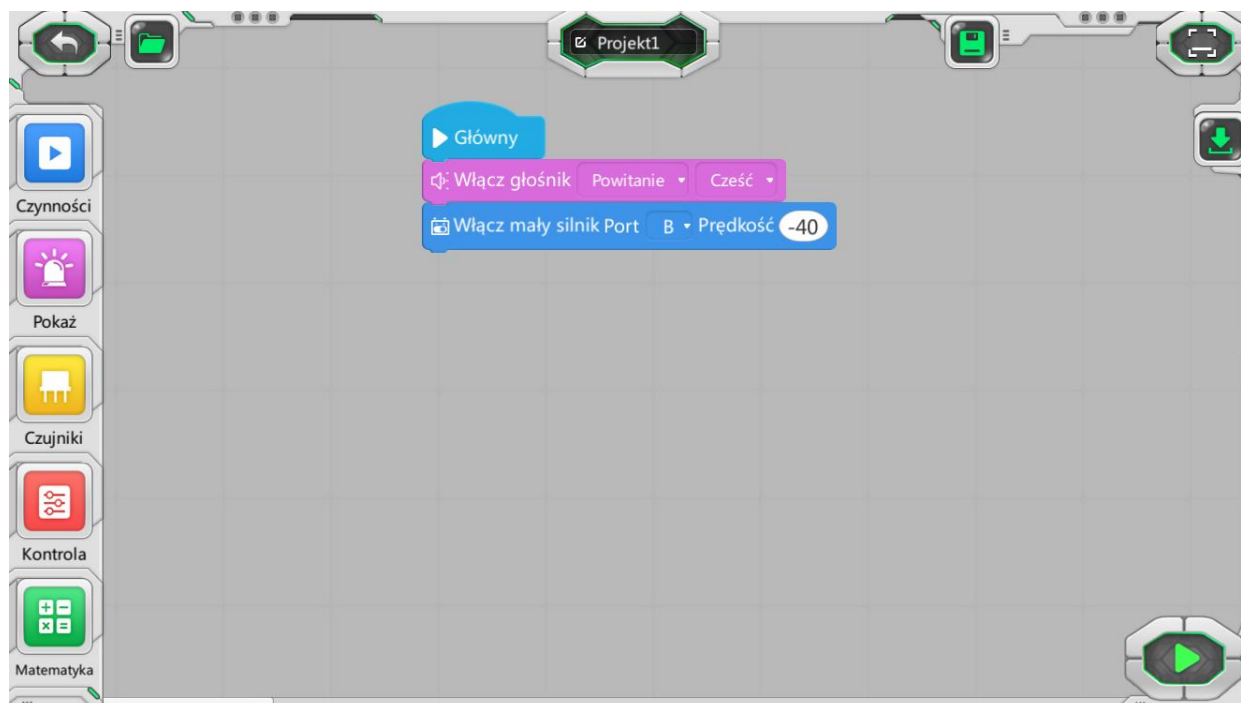
#### *Przykład 1.*

*Napiszmy prosty program, dla którego robot będzie mówił Hello (Cześć) i równocześnie silnik w porcie B będzie obracał się do tyłu z prędkością 40 przez czas 6 sekund, a następnie zatrzyma się.*

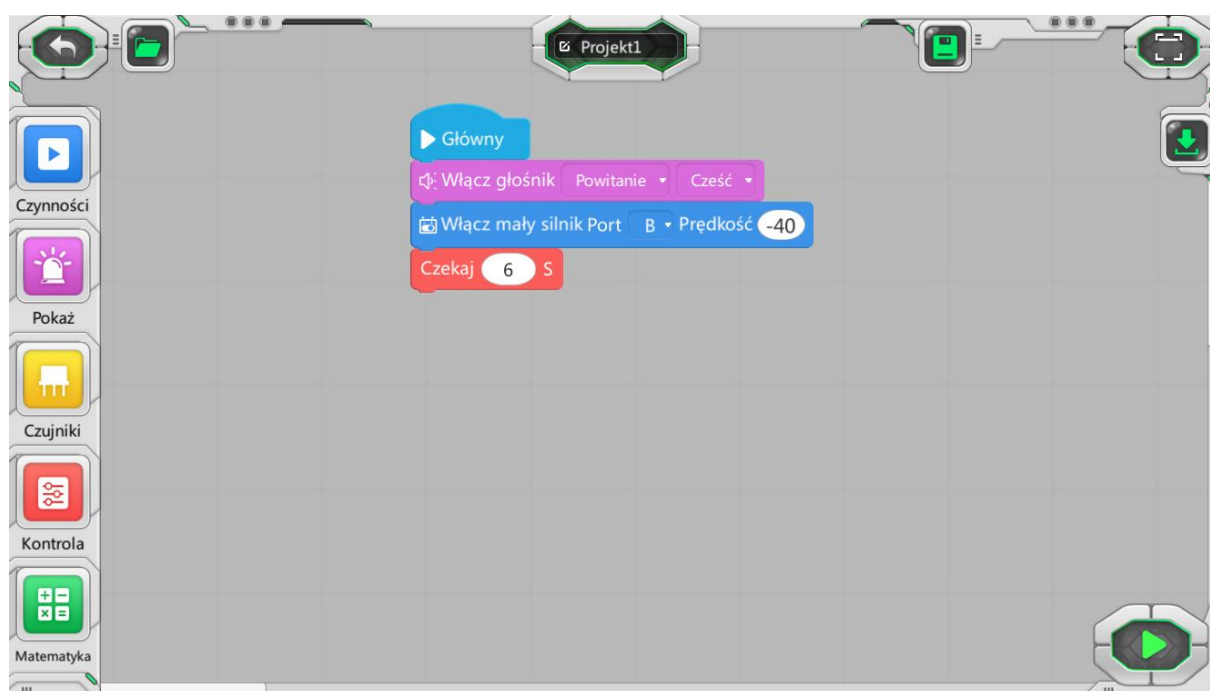
1. Przeciągnij blok głośnika z biblioteki bloku wykonawczego;



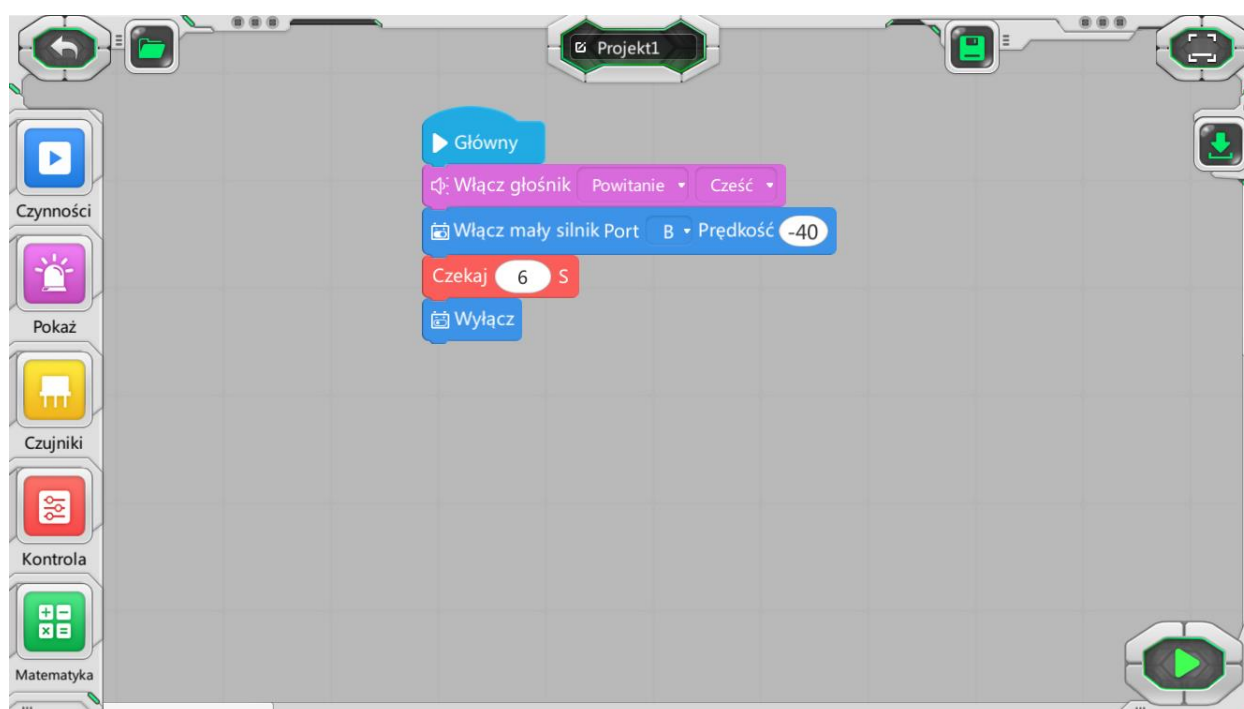
2. Przeciągnij blok silnika do łańcucha i ustaw wartość: port „B” i prędkość „-40”



3. Przeciągnij do łańcucha blok oczekiwania z biblioteki bloków poleceń sterowania, ustaw czas oczekiwania na 6 sekund, aby silnik działał przez 6 sec.



4. Przeciągnij do łańcucha blok zamykania i ustaw wartość na „turn off Motor” (Wyłącz silnik).



Przy użyciu powyższych operacji napisaliśmy prosty program dla silnika i głośnika. Podłącz silnik do portu B.

### 2.3. Używanie instrukcji oceny warunku (porównywanie wartości)

Aby dokonać oceny potrzeba nam dwóch wartości, które należy porównać. Dzięki porównaniu sprawdzimy czy spełniają dany warunek bądź nie. Wartość, którą otrzymujemy z czujnika nazywa się wartością obiektywną. Wartość, do której się odnosimy (np. wpisana przez nas) nazywamy wartością referencyjną.

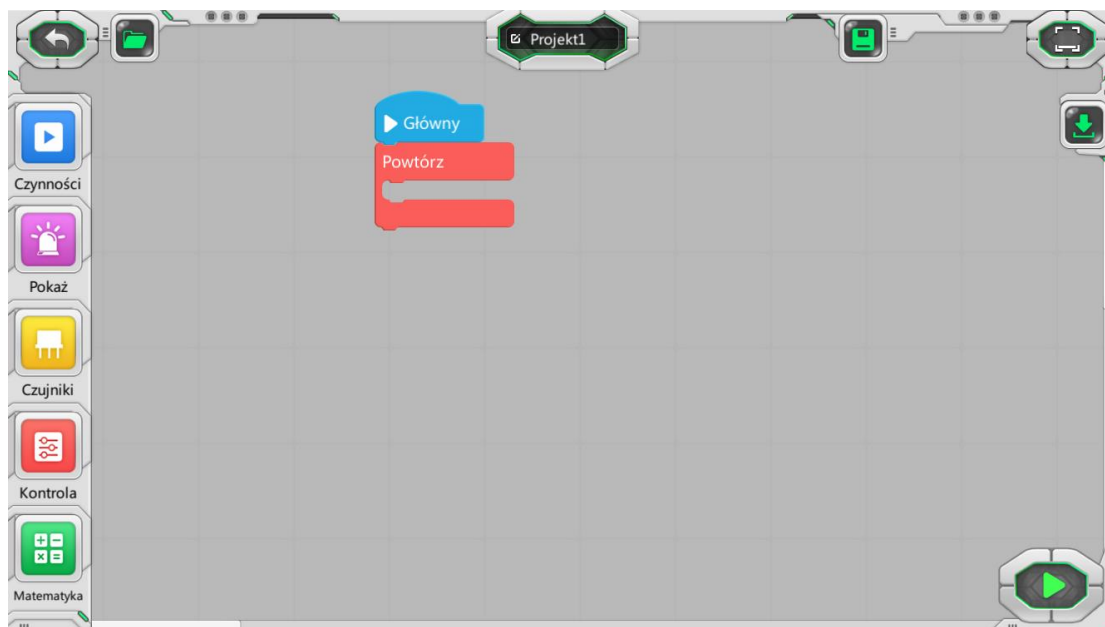
*Przykład 2:*

*Napisz program dzięki któremu robot będzie wykonywał następującą pracę:*

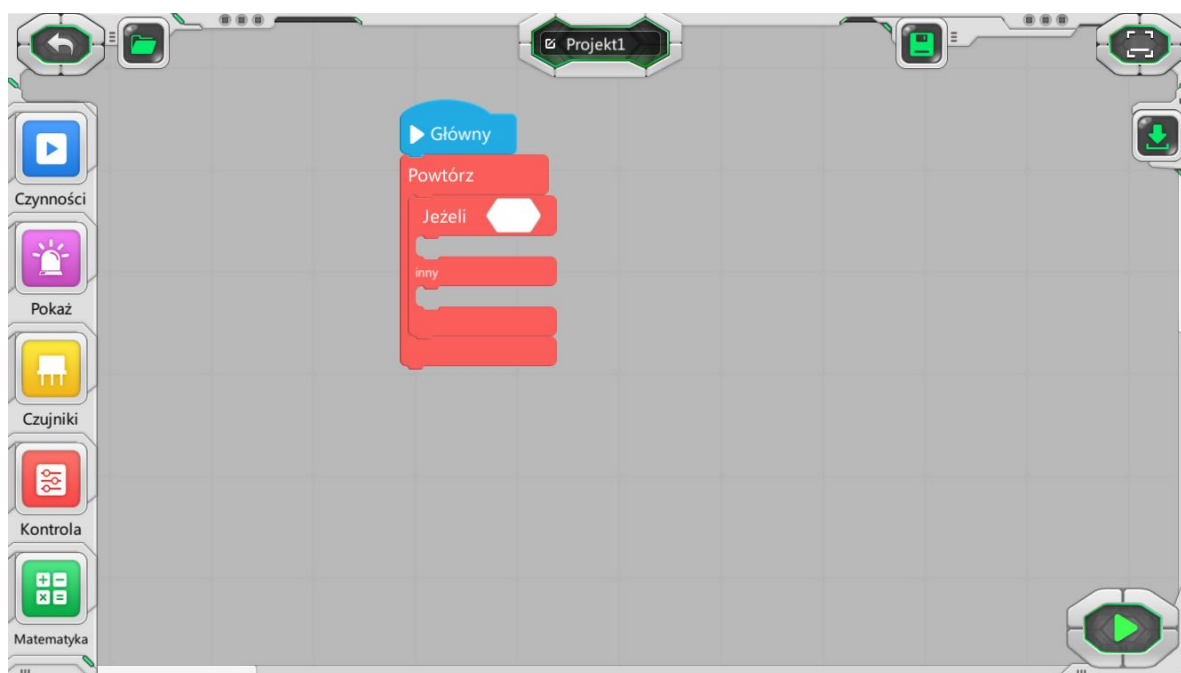
*- Jeżeli robot wykryje przeszkodę w odległości większej niż 30cm to świeci się zielona dioda LED i pracuje silnik podłączony do portu A*

*- Jeżeli robot wykryje przeszkodę w odległości mniejszej niż 30cm to świeci się czerwona dioda LED*

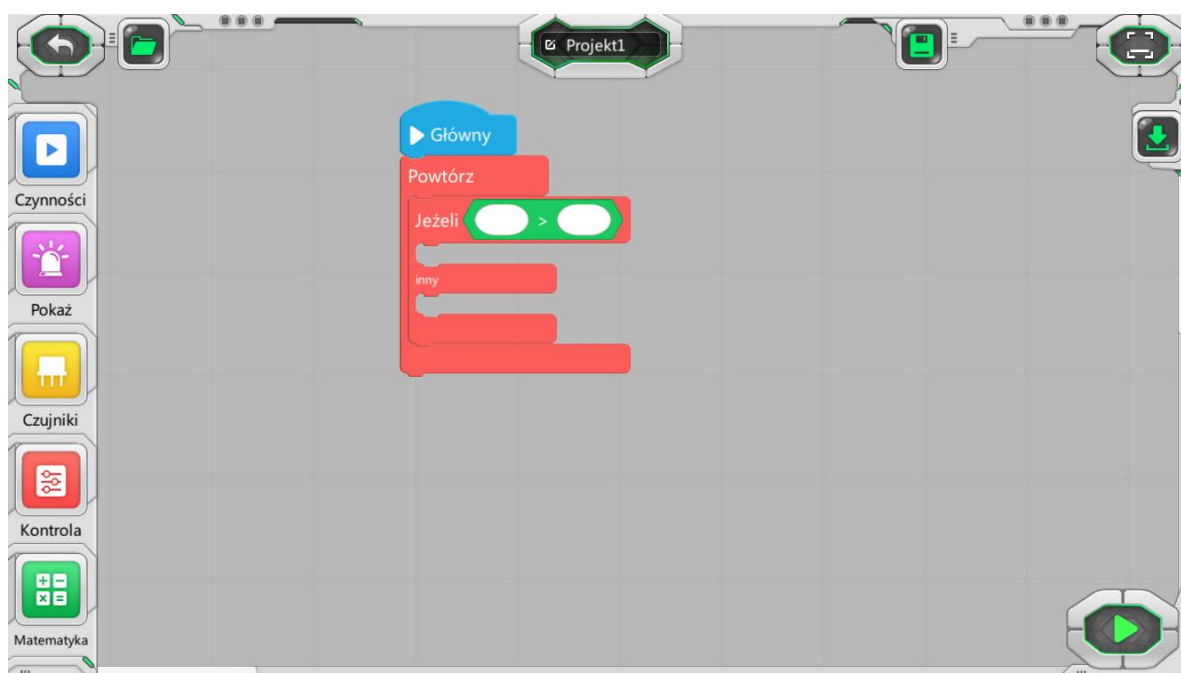
1. Przeciągnij do obszaru edytowania blok programu głównego i blok nieskończonej pętli. Połącz blok nieskończonej pętli pod programem głównym (ten blok jest dodawany w celu powtarzania programu i przeprowadzania sprawdzania w czasie rzeczywistym), jak przedstawiono na ilustracji poniżej:




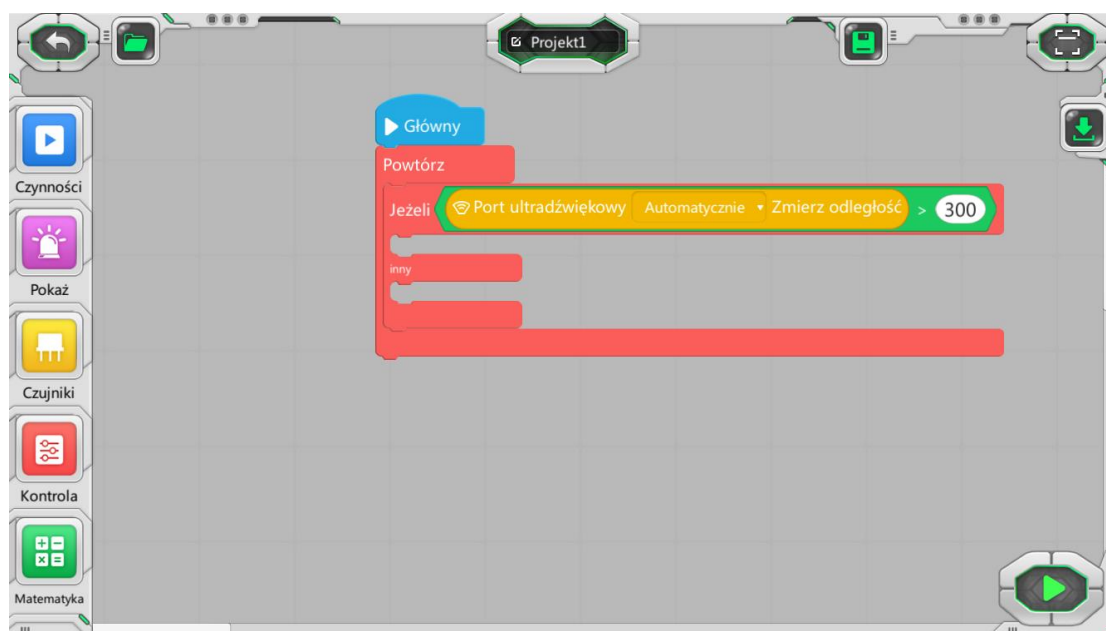
2. Przeciągnij blok oceny warunkowej (jeżeli/inny) i podłącz go do bloku nieskończonej pętli:




3. Przeciągnij blok operacji  i podłącz go do pustego pola bloku oceny warunkowej;

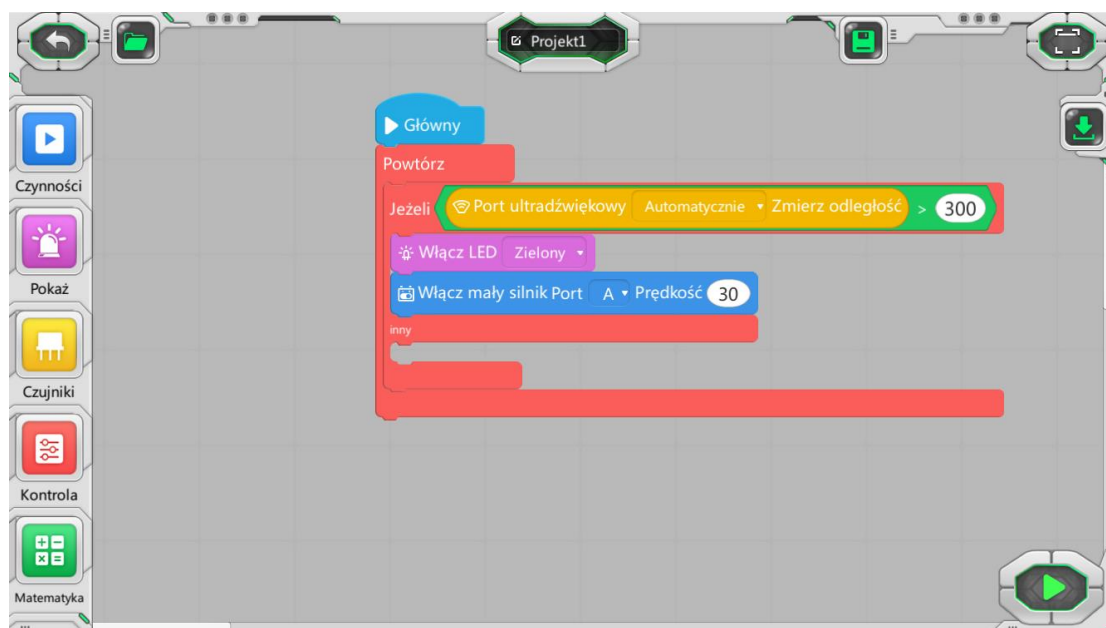


4. Przeciągnij blok czujnika ultradźwiękowego  do pierwszego pustego pola bloku operacji i wpisz 300 w drugim polu;




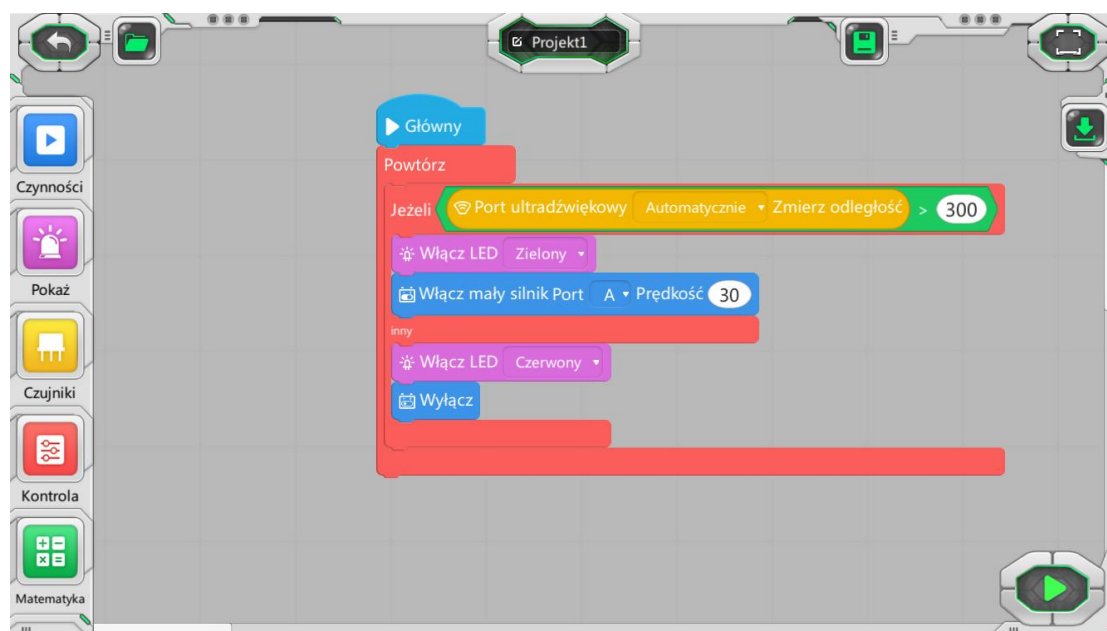
5. Dodaj blok LED i ustaw kolor podświetlenia na „Green” (zielony)

; następnie przeciągnij blok silnika i ustaw wartości: Port „A”, „Rotate Forward” (Obroty do przodu) i „30”. Wstaw te wartości do pierwszego odgaślenia oceny.



6. Przeciągnij blok LED, ustaw kolor diody LED na „Red” (Czerwony)

; Dodaj blok zamykania. Wstaw te bloki w kolejności do drugiego odgaślenia, jak przedstawiono na ilustracji.



W ten sposób program sterujący diodą LED i silnikiem został ukończony. Podłącz czujnik ultradźwiękowy i silnik do odpowiednich portów. Sprawdź działanie programu: gdy przed czujnikiem ultradźwiękowym nie są wykrywane przeszkody, silnik obraca się do przodu; w przypadku wykrycia przeszkód przed czujnikiem, silnik zatrzymuje się i włącza się czerwona dioda LED.

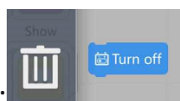
### Rozdział 3. Jak usunąć blok

Aby usunąć blok, należy go dotknąć i przytrzymać przez dłuższy czas. Na stronie obszaru menu wyświetlona zostanie ikona. Przeciągnij blok do tej ikony i zwolnij palec. Operacja usuwania



Po lewej stronie obszaru menu wyświetlona zostanie ikona. Przeciągnij blok do tej ikony i zwolnij palec. Operacja usuwania

zakończona. Zostanie wyświetlona ikona:

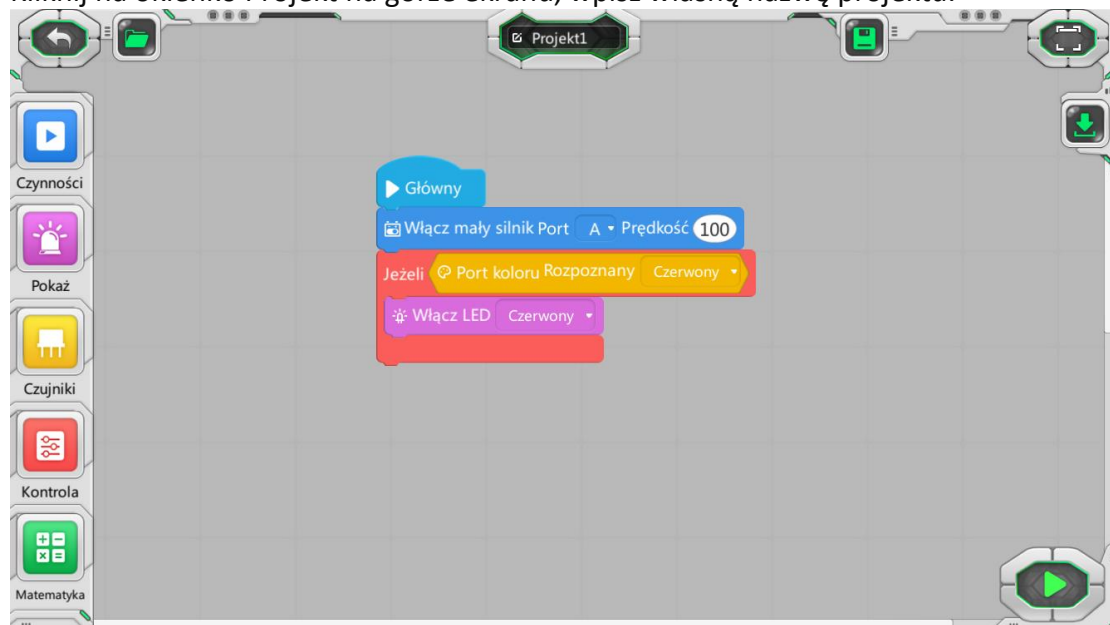




## Rozdział 4. Jak zapisać projekt/program w pamięci robota

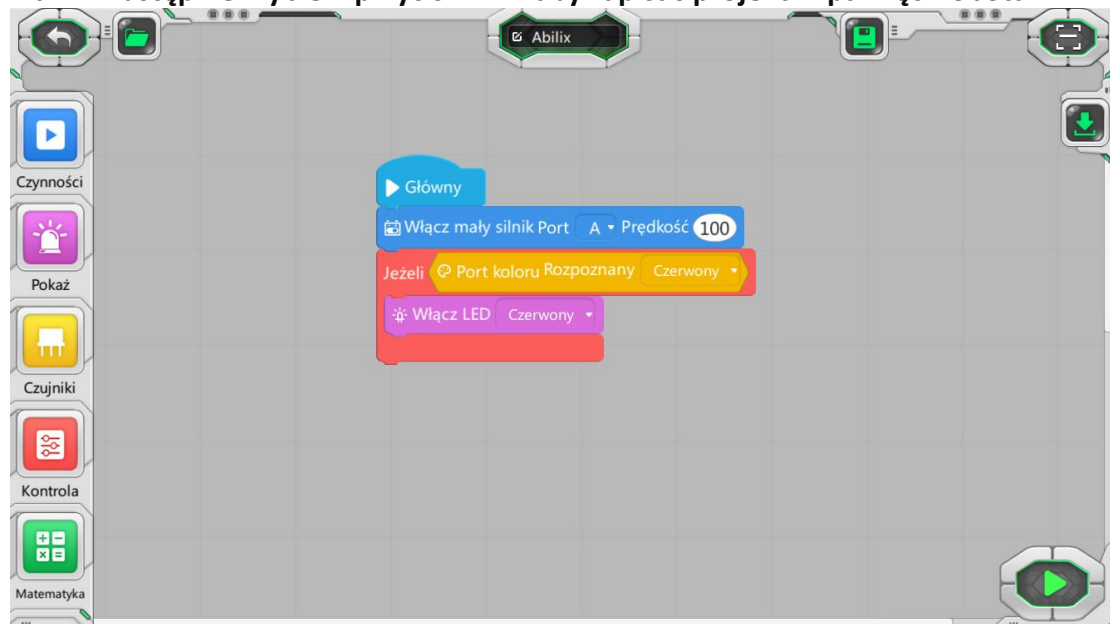
Po napisaniu programu możesz zapisać go w pamięci robota, aby móc odtworzyć go bezpośrednio ze sterownika, a nie z aplikacji na tablecie/komputerze.

Kliknij na okienko Projekt na górze ekranu, wpisz własną nazwę projektu.



Na górze ekranu pokazana zostanie zapisana przez siebie nazwa, w naszym przykładzie jest to

Abilix. Następnie wybierz przycisk  aby zapisać projekt w pamięci robota.



Projekt będziesz mógł odtworzyć z poziomu sterownika Krypton, poprzez wybranie na ekranie dotykowym odpowiedniej nazwy projektu, w tym przykładzie „Abilix”

Materiał został stworzony przez firmę SOLECTRIC GMBH Polska sp. Z o.o. spółka komandytowa i jest jej własnością.

Copyright by:  **solectric**  
Education Solutions

**Dystrybutor nowoczesnych technologii dla edukacji**

[www.solelectric.pl](http://www.solelectric.pl)