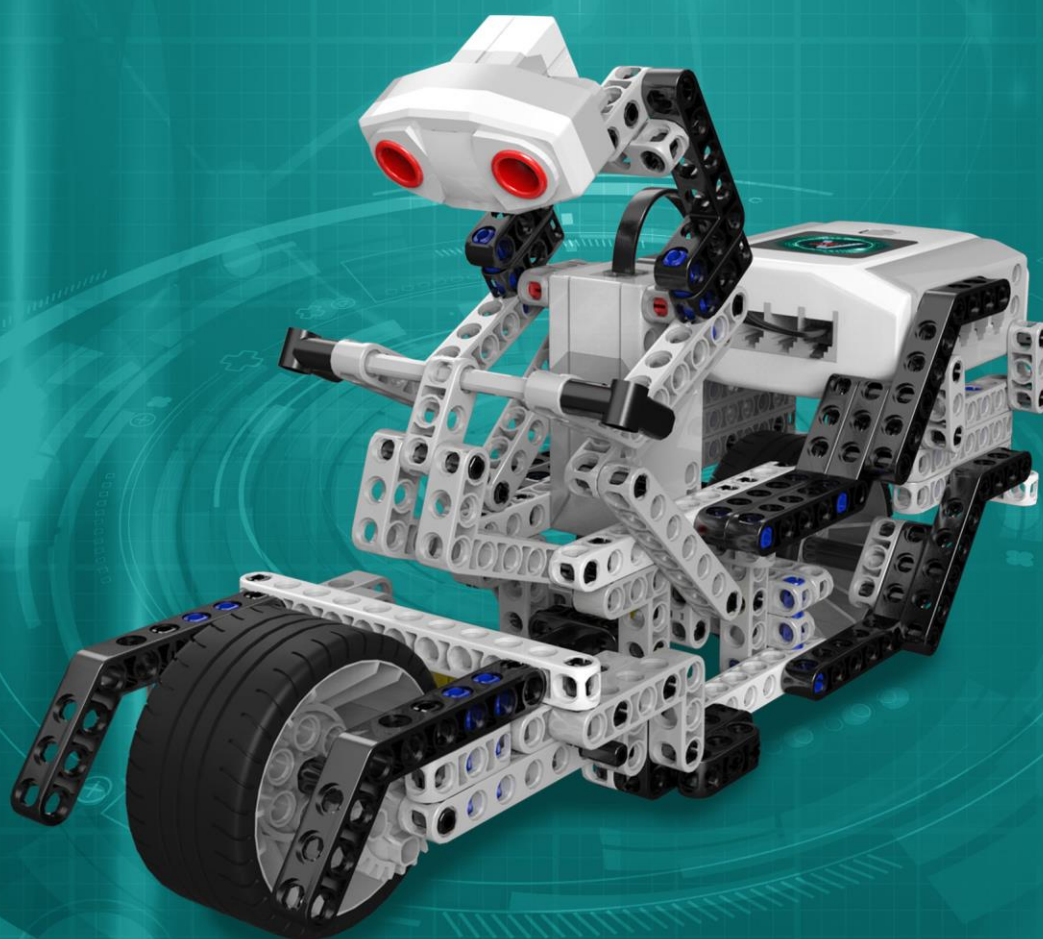




EDUCATIONAL ROBOT

JĘZYK SCRATCH

Manual użytkownika



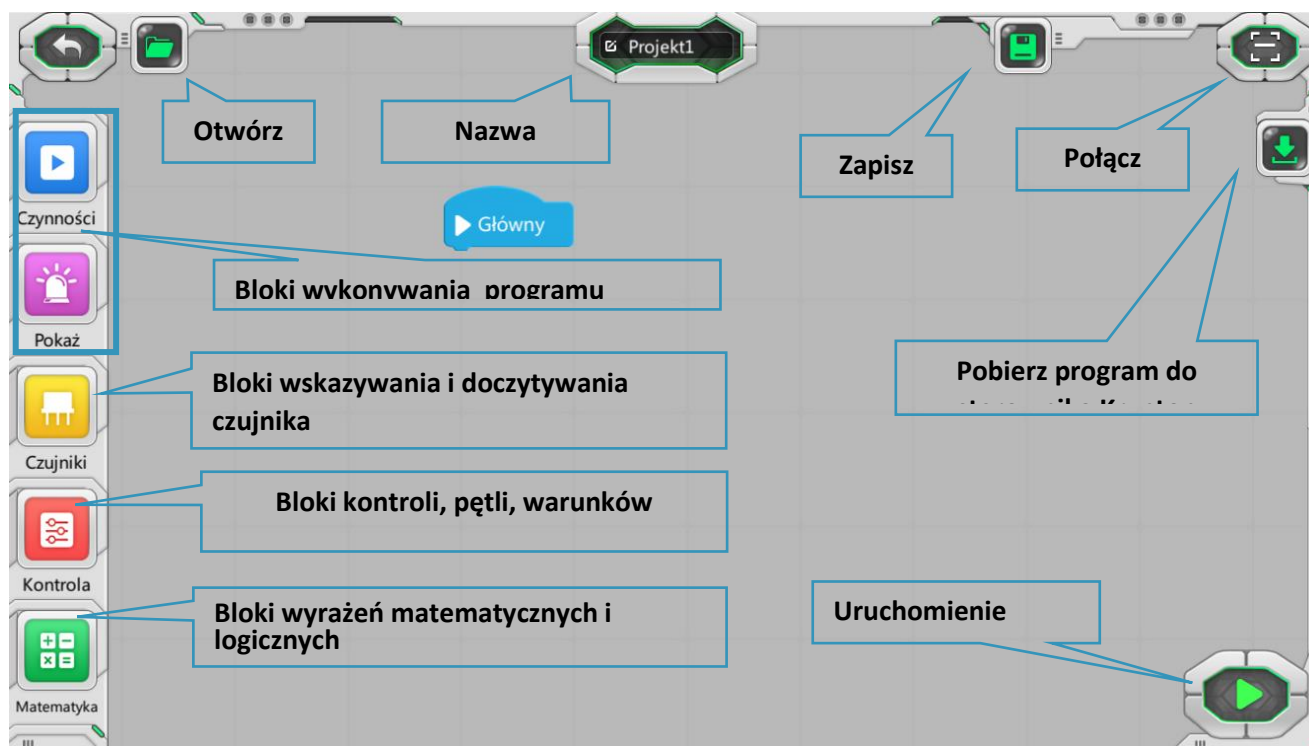
Copyright by  **soletric**
Education Solutions

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 czynności	4
1.2.2.	Zakładka „Pokaż”	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

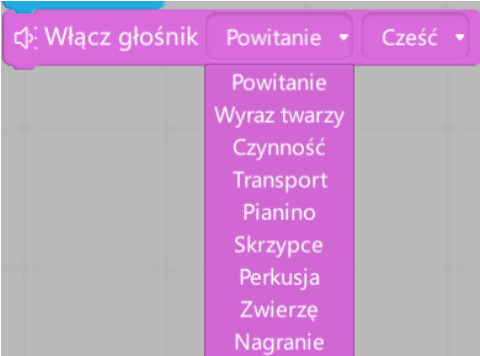
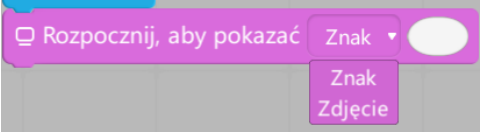
Bloki wykonywania programu znajdują się w dwóch zakładkach.

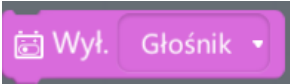
1.2.1. Niebieska zakładka *czynności* odpowiada za sterowanie prędkością i obrotami małego i dużego silnika.

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

Nr	Ikona	Nazwa	Funkcja
1	Włącz (duży) silnik, Port <u>A</u>, prędkość <u>100</u>  Włącz (mały) silnik, Port <u>A</u>, prędkość <u>100</u> 	Ruch silnika	Sterowanie kierunkiem obrotów prędkością dużych i małych silników. Dostępne porty: A, B, C lub D Zakres prędkości: -100 do 100. 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.
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.

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

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

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

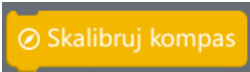
Biblioteka bloku czujnika dotyczy czujników, które są podłączane do portów sterownika przy użyciu przewodów połączeniowych.

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.

			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 automatycznie będzie pobierał tylko wartość zwracaną z portu 1.
3		Czujnik kolizji	Pobiera wartość zwracaną przez czujnik kolizji. 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. Wybór portu: „Automatyczny” lub 1~7. 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.
4		Czujnik kolorów	Sprawdza kolor zgodnie z zaprogramowaną wartością progową.

			Rozpoznaje pięć kolorów: czerwony, żółty, zielony, niebieski i biały. 0- Jeżeli czujnik wykryje kolor czerwony 1- Jeżeli czujnik wykryje kolor żółty 2- Jeżeli czujnik wykryje kolor zielony 3- Jeżeli czujnik wykryje kolor niebieski 4- Jeżeli czujnik wykryje kolor biały.
5		Czujnik skali szarości	Odczytuje wartość zwracaną przez czujnik skali szarości. Wartości dla koloru białego są małe (ok 200-300), natomiast dla pozostałych kolorów są większe. Maksymalna wartość zwrotna może być równa 4000, (czerni). Wybór portu: „Automatyczny” lub 1~7. 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.

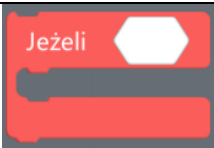
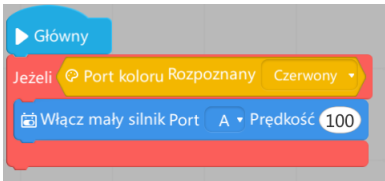
6		Kompas	Umożliwia pobieranie wartości kąta z kompasu (w górnej części sterownika). Zakres zwracanych wartości wynosi 0-359 stopni. Wartości 0 i 360 stopni dotyczą północy geograficznej; 90 stopni wschodu geograficznego; 180 stopni południa geograficznego a 270 stopni wschodu geograficznego. Przed rozpoczęciem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

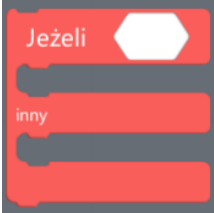
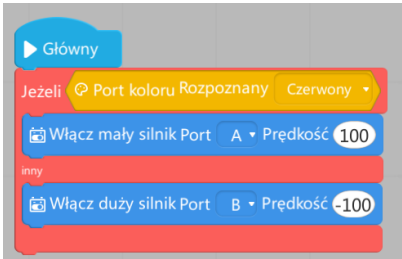
1.4. Biblioteka bloków kontroli, pętli, warunków w programie (czerwona)

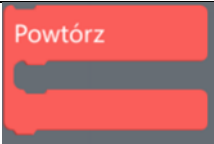
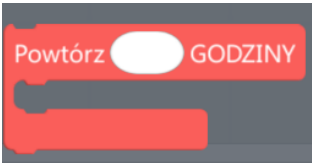
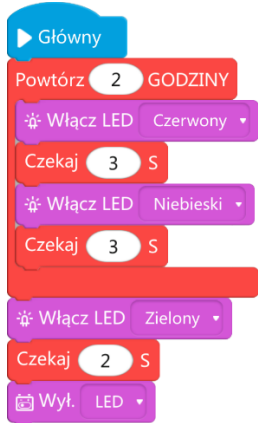
W programie dostępne są dwie metody pobierania wartości zwracanych z czujników: zapisywanie i porównywanie, gdzie porównywanie ma charakter nadrzędny. Dla oceny warunku moduł Abilix Scratch oferuje cztery schematy blokowe:

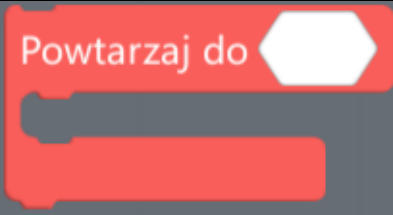
- wyrażenie POWTARZAJ...
- wyrażenie JEŻELI ...
- wyrażenie W PRZECIWNYM RAZIE,
- wyrażenie DLA
- i wyrażenie CZEKAJ. Wszystkie wyrażenia są dostępne w bibliotece.

Ikony, nazwy i funkcje przedstawiono w tabeli poniżej:

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

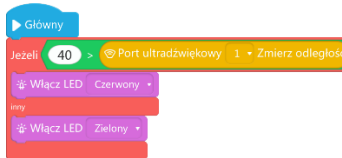
2	 „Jeśli warunek jest spełniony wykonaj zadanie A jeżeli nie jest wykonaj akcję B”  Np. PRZYKŁAD: jeżeli port koloru wykryje kolor czerwony → włącz mały silnik z portu A z prędkością 100 → w przeciwnym razie włącz duży silnik z portu B z prędkością -100	Ocena warunku	Wyrażenie JEŻELI (warunek) W PRZECIWNYM RAZIE ... w języku C. 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)
3	 „Czekaj X sekund”	Czekaj	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.

4	 „Czekaj do....” Np. włącz czerwony LED niech świeci aż port ultradźwiękowy wykryje przeszkodę 	Blok Pauzy warunkowej	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.
5	 „Powtarzaj w nieskończoność do zakończenia programu”	Pętla	Wyrażenie POWTÓRZ (prawda) w języku C. Nie posiada wartości. Służy do powtarzania odgałęzionego wyrażenia do momentu zakończenia programu.
6	 „Powtarzaj X razy,, 	Pętla wielokrotna	Powtórz X razy. Dla wyrażeń w języku C. Wartością jest liczba pętli. Służy do powtarzania instrukcji w pętli przez określoną ilość razy. 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

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

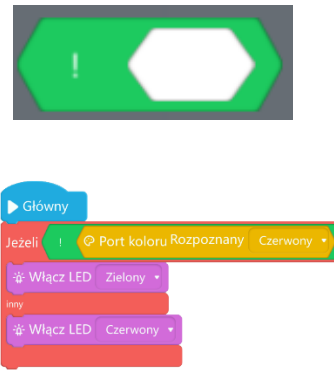
1.5. Biblioteka obsługi wyrażeń matematycznych i logicznych (zielona)

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”	Blok wartości logicznej „i” – wartość zwrócona będzie równa 1 wtedy i tylko wtedy gdy obie wartości będą prawdziwe. Przykład: -Jeżeli czujnik dotyku zostanie wciśnięty(1) LUB port koloru rozpozna kolor czerwony(1) wtedy = (1) = dioda LED zaświeci na czerwono. Czyli: <table><tr><th>Czujnik dotyku wciśnięty</th><th>Czujnik koloru wykrywa czerwony</th><th>WYNIK</th></tr><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></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																

<

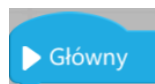
		not	Blok negacji warunku „jeżeli czujnik dotyku nie jest wciśnięty, świeć na czerwono. W przeciwnym razie na niebiesko” Przykład: Jeżeli czujnik koloru NIE rozpozna czerwonego koloru wtedy LED świeci na zielono. W przeciwnym razie (kiedy rozpozna czerwony) świeci na czerwono. Uwaga: „nie rozpozna” oznacza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

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:

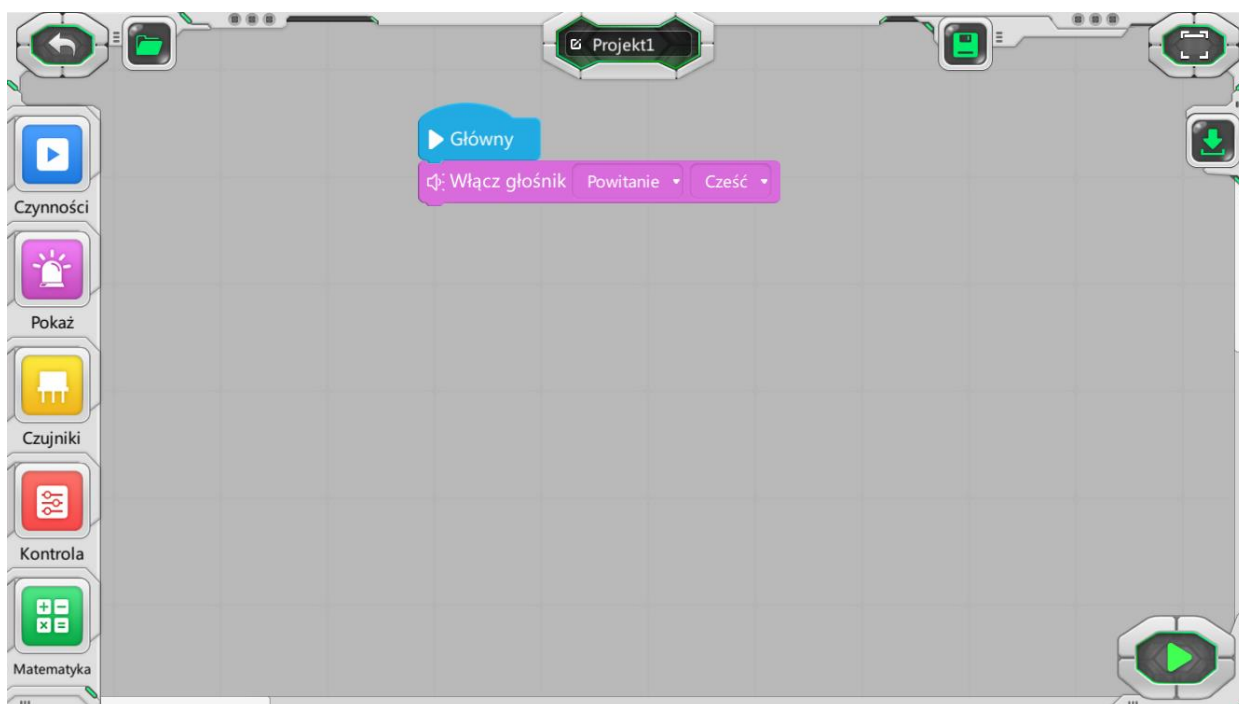
1. Wybierz wymagany blok z biblioteki bloków i przeciągnij go do obszaru edytowania po prawej stronie.
2. Połącz blok z łańcuchem programu głównego.
3. Dodaj lub zmień wartości dla bloku.

Poszczególne kroki zostały przedstawione na poniższych przykładach:

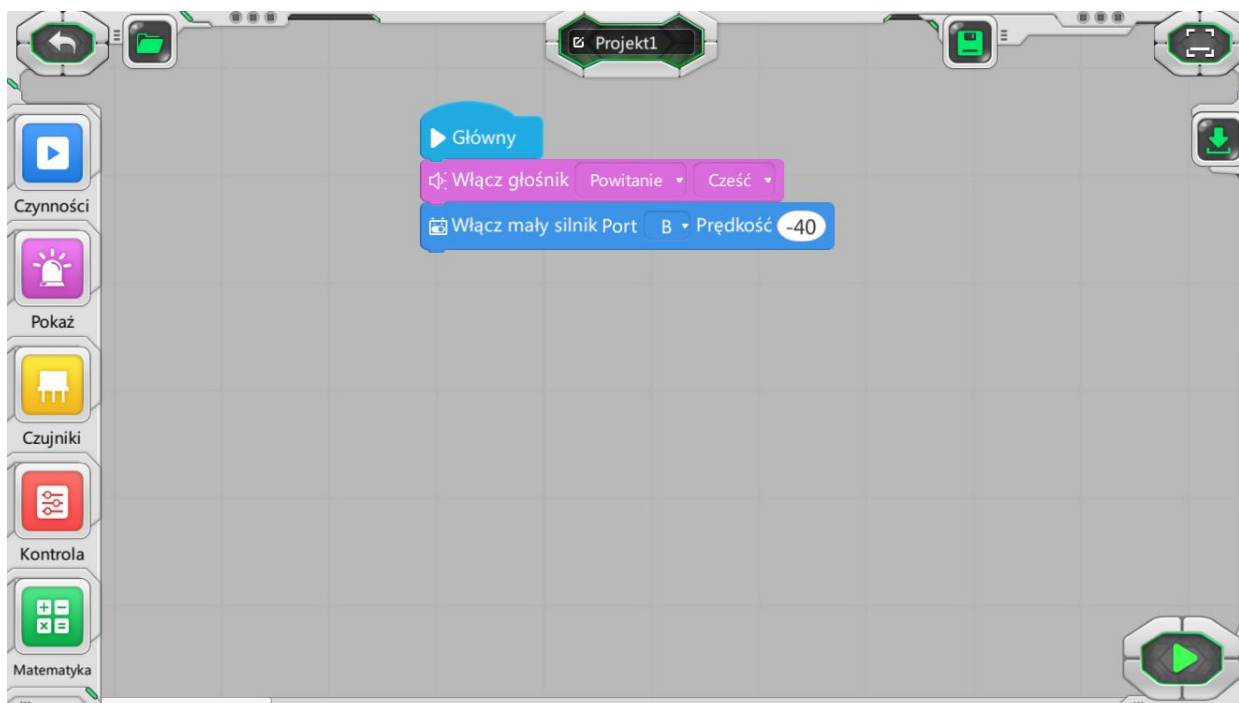
Przykład 1.

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.

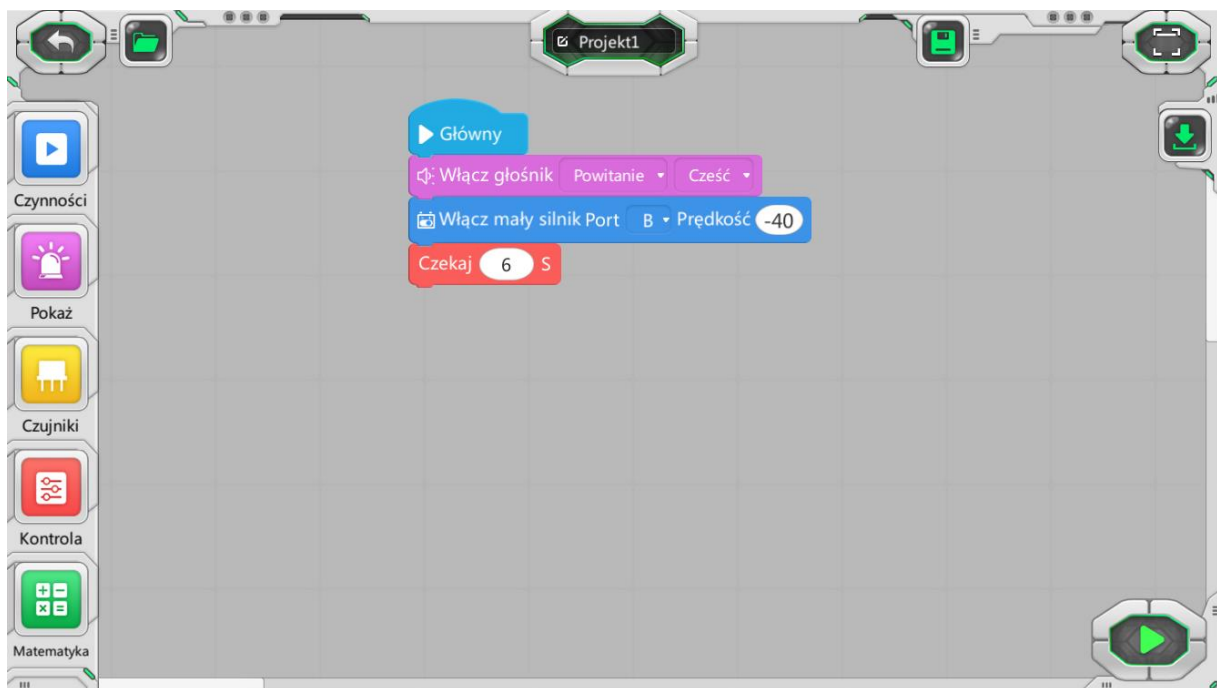
1. Przeciągnij blok głośnika z biblioteki bloku wykonawczego;



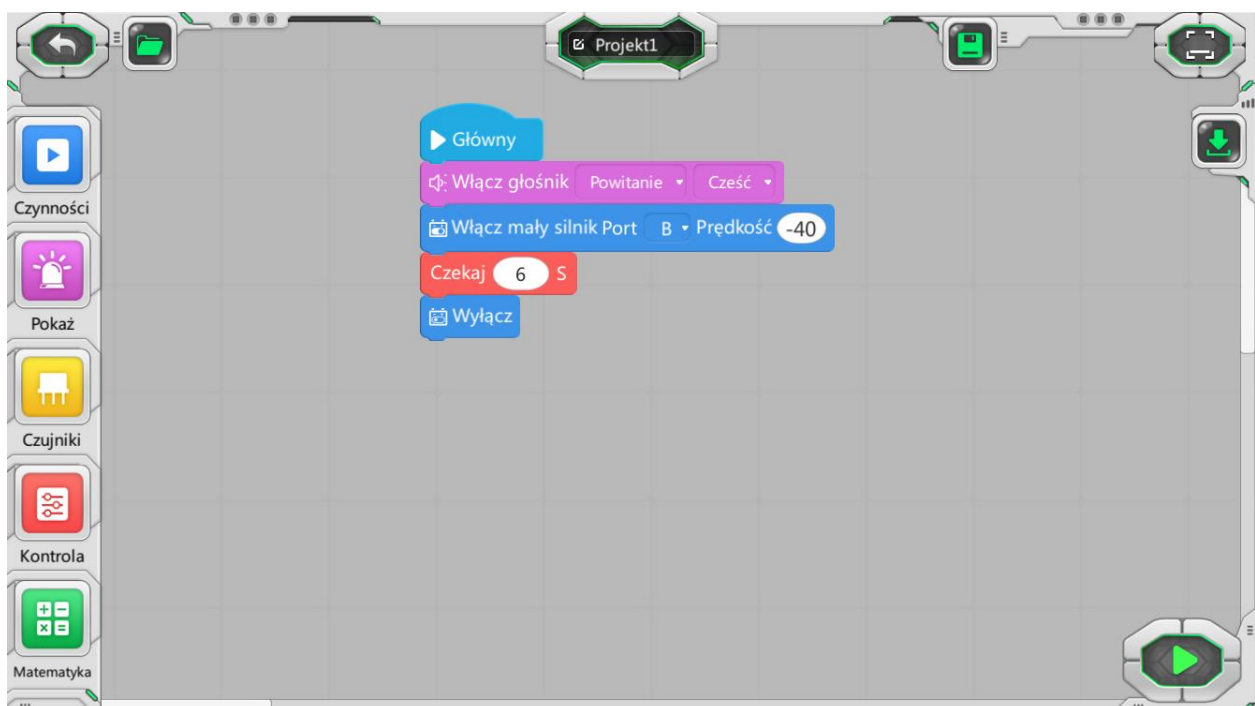
2. Przeciągnij blok silnika do łańcucha i ustaw wartość: port „B” i prędkość „-40”



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)

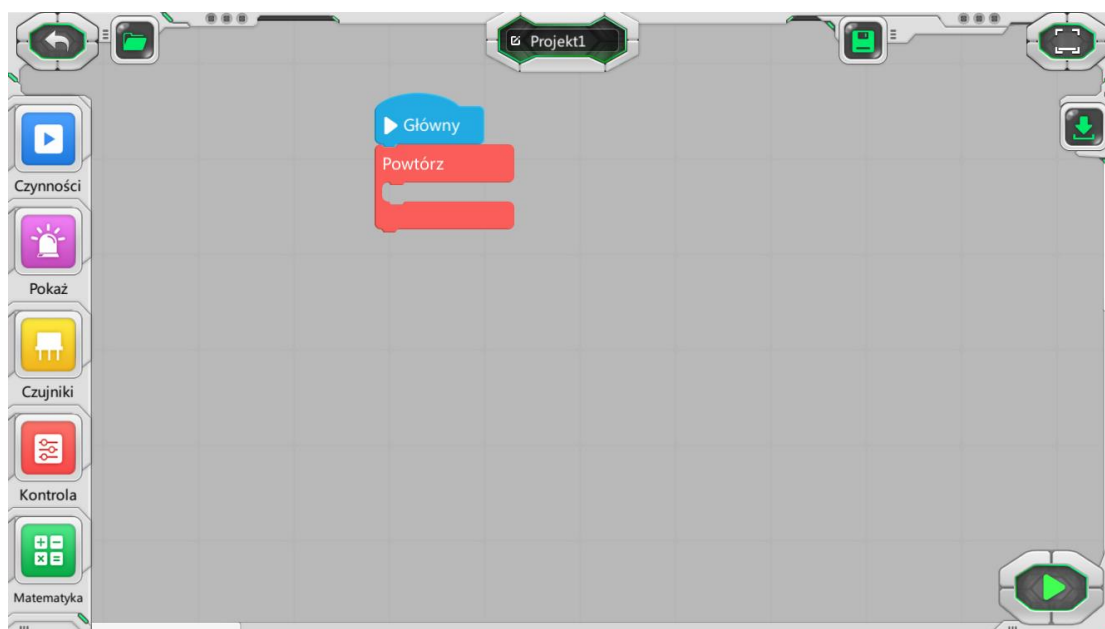
Aby dokonać oceny potrzeba nam dwóch wartości, które należy porównać. Dzięki porównaniu sprawdzimy czy spełniają dany warunek bądź nie. Wartość, którą otrzymujemy z czujnika nazywa się wartością obiektywną. Wartość, do której się odnosimy (np. wpisana przez nas) nazywamy wartością referencyjną.

Przykład 2:

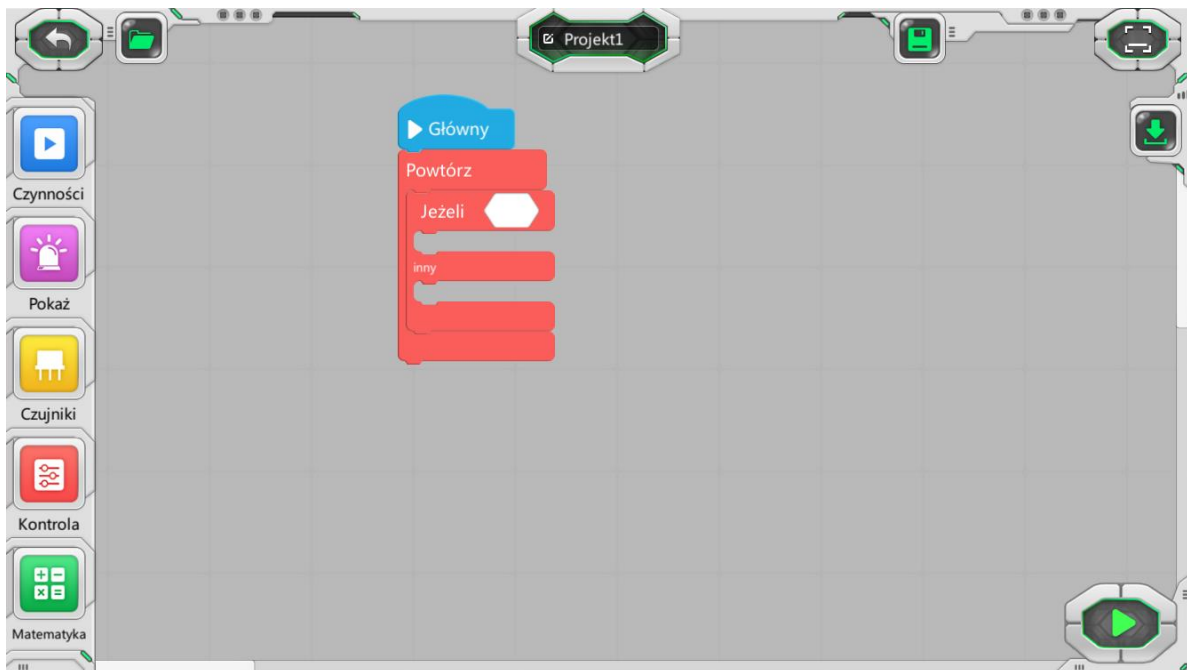
Napisz program dzięki któremu robot będzie wykonywał następującą pracę:

- *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*

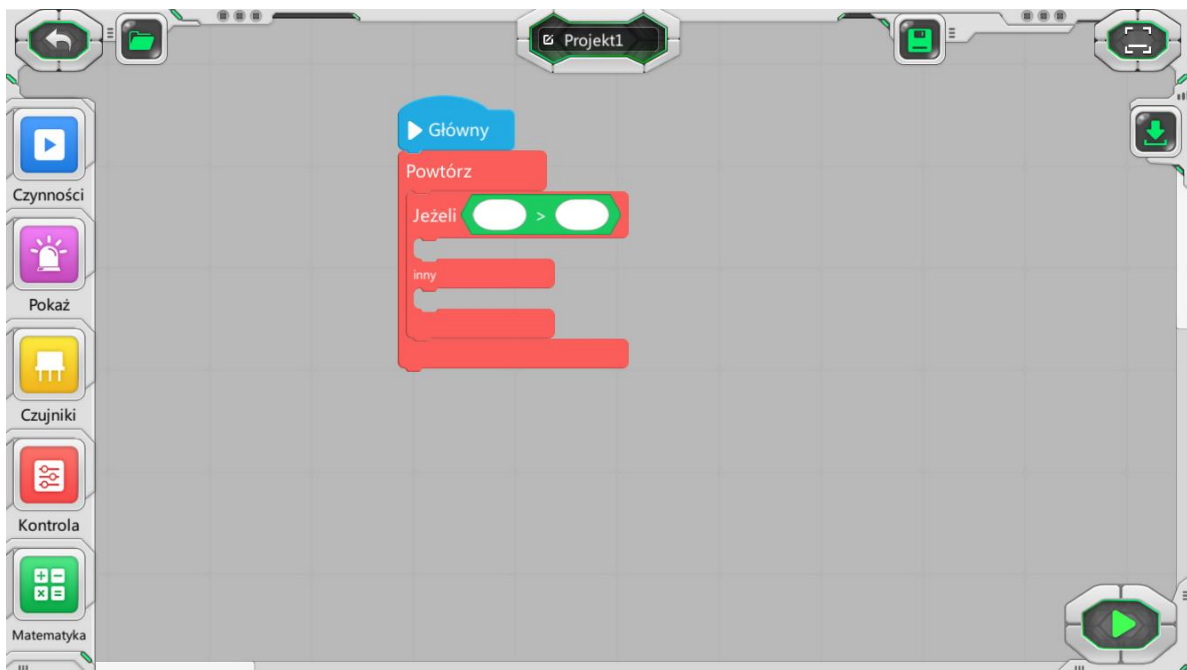
1. Przeciągnij do obszaru edytowania blok programu głównego i blok nieskończonej pętli. Połącz blok nieskończonej pętli pod programem głównym (ten blok jest dodawany w celu powtarzania programu i przeprowadzania sprawdzania w czasie rzeczywistym), jak przedstawiono na ilustracji poniżej:



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



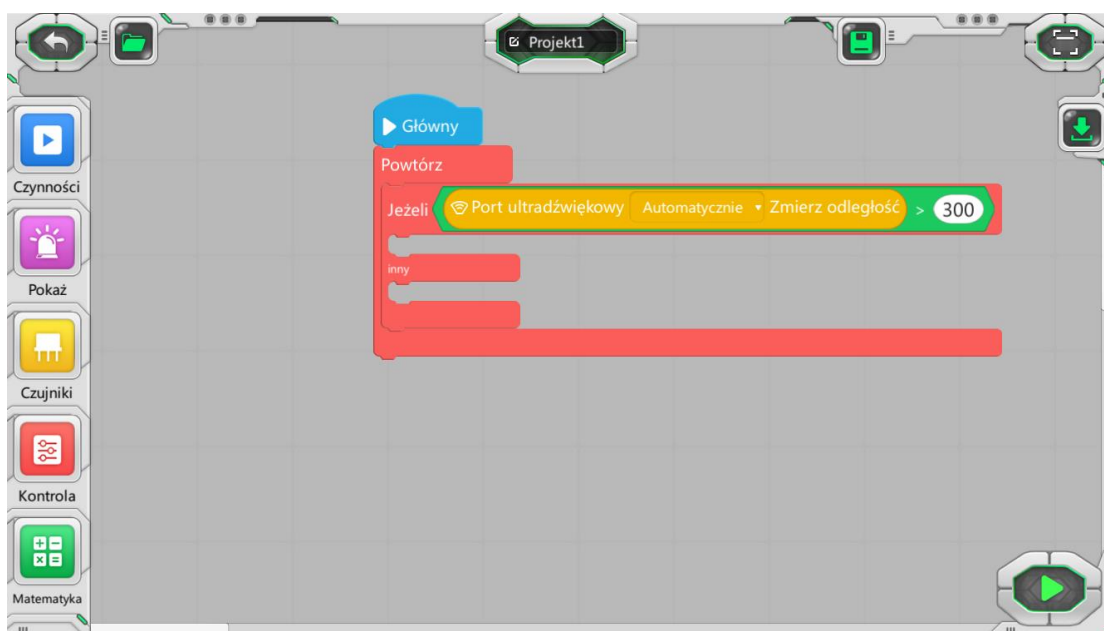
3. Przeciągnij blok operacji  i podłącz go do pustego pola bloku oceny warunkowej;



4. Przeciągnij blok czujnika ultradźwiękowego

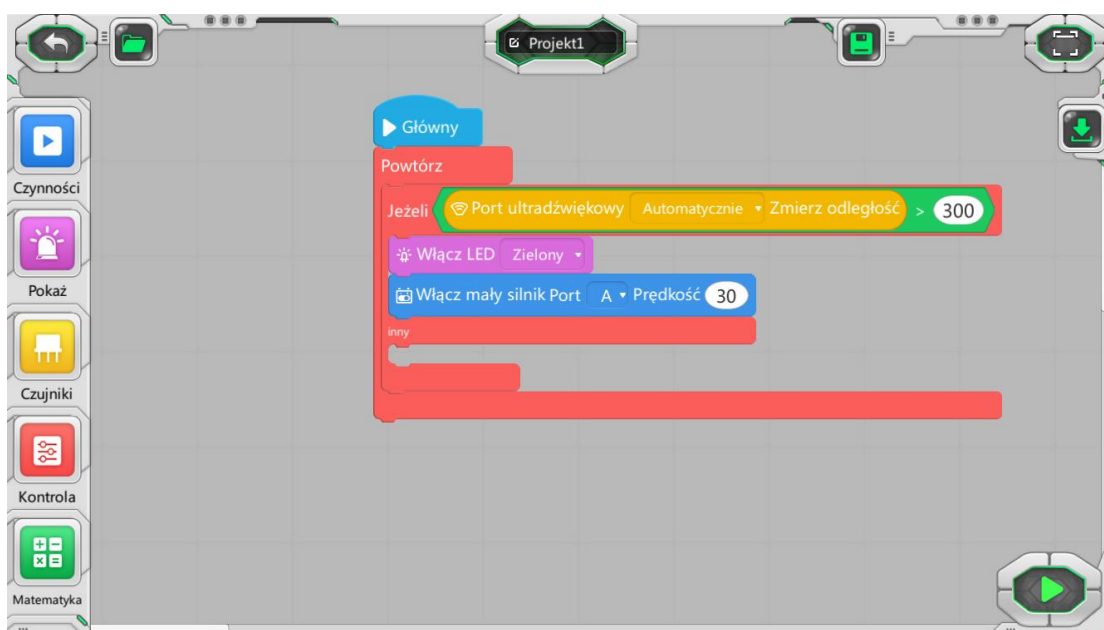


do pierwszego pustego pola bloku operacji i wpisz 300 w drugim polu;



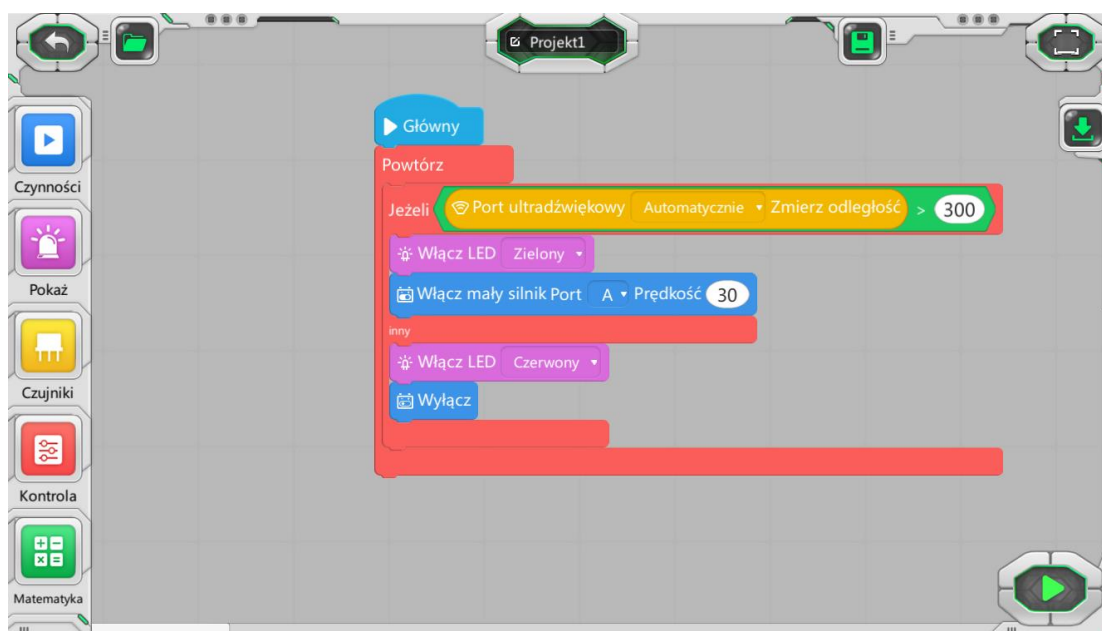
5. Dodaj blok LED i ustaw kolor podświetlenia na „Green” (zielony)

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



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

; Dodaj blok zamykania. Wstaw te bloki w kolejności do drugiego odgałęzienia, 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.

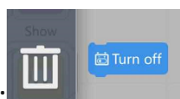
Rozdział 3. Jak usunąć blok

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



Po lewej Przeciągnij zostanie

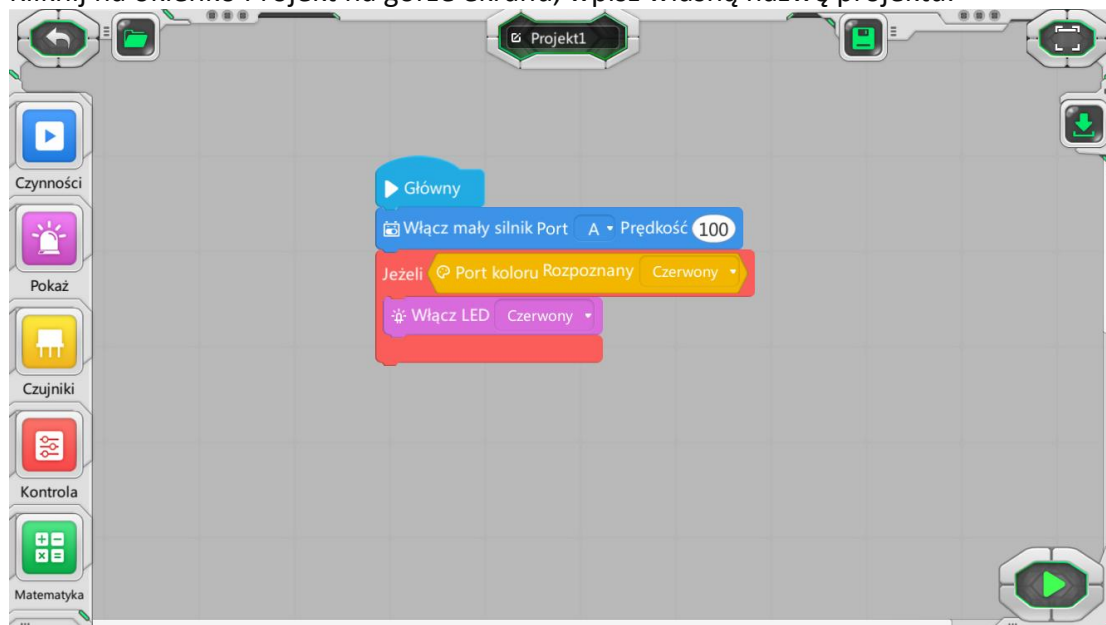
zakończona. Zostanie wyświetlona ikona:



Rozdział 4. Jak zapisać projekt/program w pamięci robota

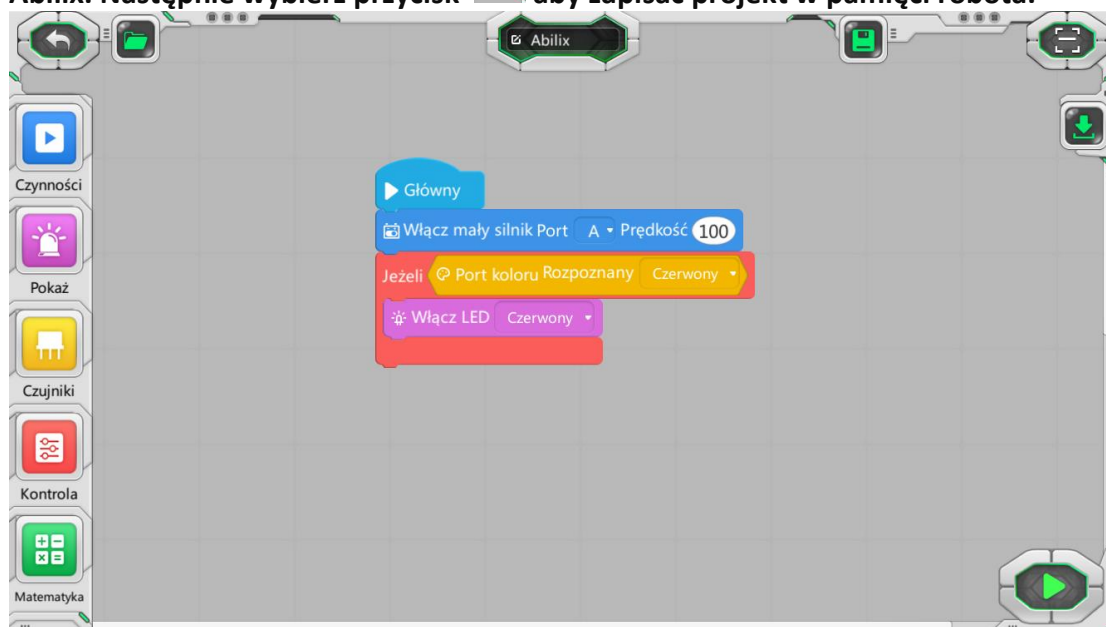
Po napisaniu programu możesz zapisać go w pamięci robota, aby móc odtworzyć go bezpośrednio ze sterownika, a nie z aplikacji na tablecie/komputerze.

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.

Dystrybutor nowoczesnych technologii dla edukacji

Solectric GMBH Polska

ul. Górczewska 216

01-460 Warszawa

www.solectric.pl

info@solectric.pl