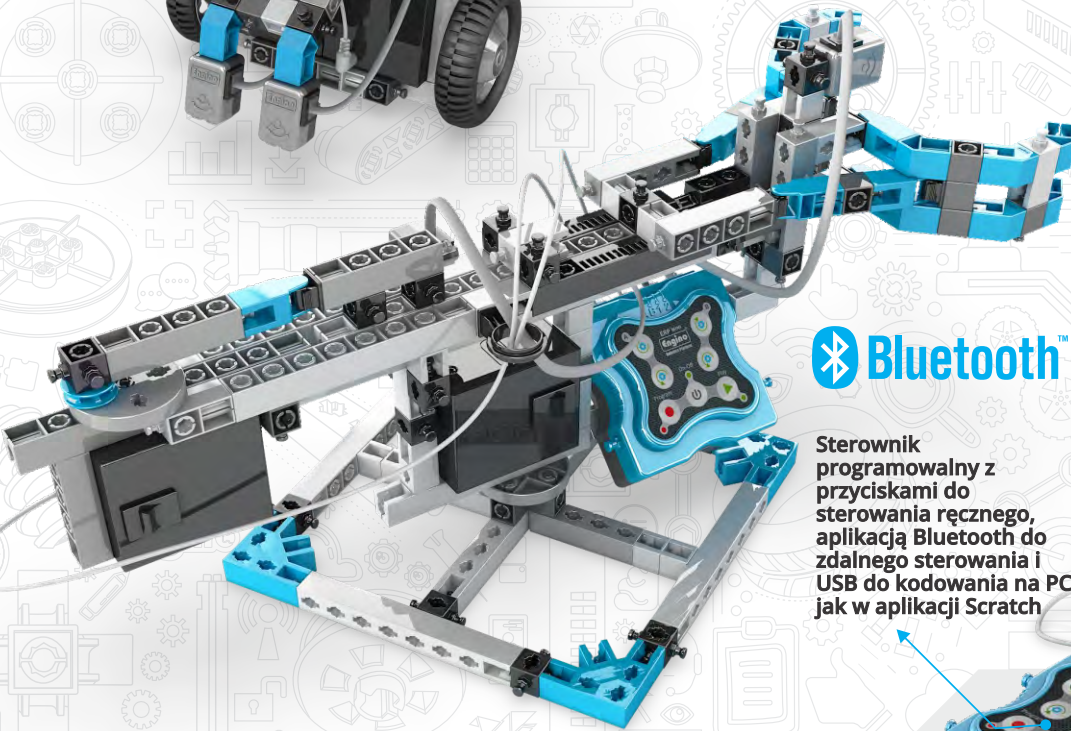
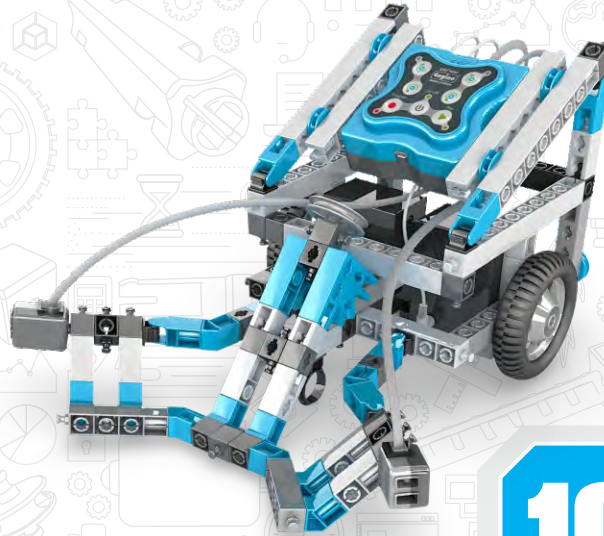


# Engine

# CREATIVE™ engineering maker Pro



## 100 IN 1 robotized MODELS

**5**  
modeli

drukowane  
instrukcje  
(w zestawie)



**95**  
modeli

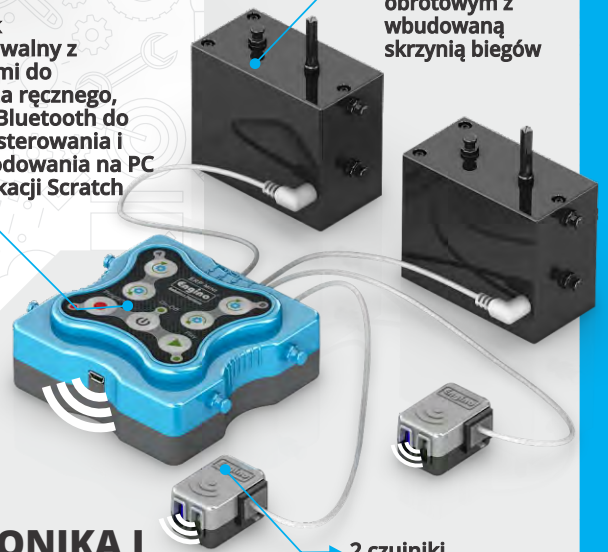
instrukcje  
3D online



### Bluetooth™

Sterownik programowalny z przyciskami do sterowania ręcznego, aplikacją Bluetooth do zdalnego sterowania i USB do kodowania na PC jak w aplikacji Scratch

2 silniki o wysokim momencie obrotowym z wbudowaną skrzynią biegów



### ⊕ ELEKTRONIKA I MOC SILNIKA

2 czujniki  
zbliżeniowe na  
podczerwień

• INTERAKTYWNE INSTRUKCJE 3D

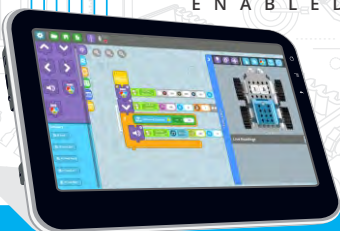
• BUDUJ

• BAW SIĘ

• UCZ SIĘ

• WYNAJDŹ

**STEM**  
ENABLED



## 10+

Odpowiedni  
wiek



Instrukcje  
budowania

## 3

Lata gwarancji



Narzędzie  
do ekstrakcji  
Do łatwego demontażu  
małych części

# CREATIVE™ engineering

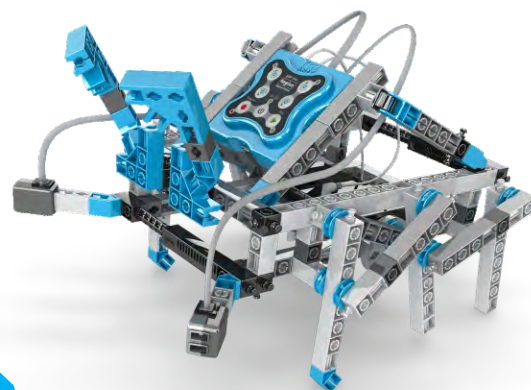
## maker Pro



### zbuduj wentylator

Model wentylatora wprowadzi Cię w główne pojęcia programowania i innowacyjny sterownik MINI 2.0. Może również pomóc w testowaniu różnych zmiennych i programów utworzonych za pomocą oprogramowania KEIRO.

- O funkcjach input i output.
- Jak ręcznie zaprogramować robota.



### zbuduj hexapoda - sześcionożnego robota

Eksperymentuj z ruchem za pomocą nóg zamiast kół dzięki temu ekscytującemu modelowi hexapoda! Ustaw czujniki podczerwieni po obu stronach tak, aby uruchamiały się, gdy zbliży się przeszkoda i robot mógł ją ominąć, używając swoich nóg.

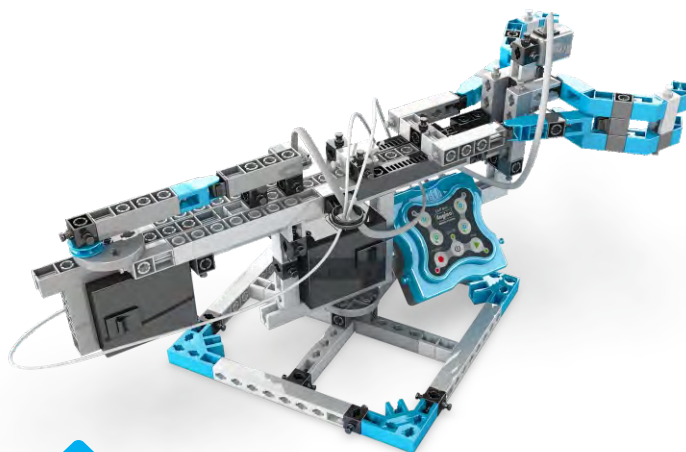
- Jak pracować z ruchem nóg
- Co to jest polecenie IF.



### zbuduj robota

Zbuduj model robota i wykorzystaj jego podwójne zdolności! Korzystając z programowalnych czujników robot może rozróżniać kolor czarny i biały, co umożliwi mu naśladować linię lub wykrywać objekty na drodze.

- Jak działają czujniki podczerwieni
- O bramkach logicznych i poleceniu WHILE.



### zbuduj robota TWISTER-a

Ten niesamowity robot TWISTER imituje prawdziwe ramiona robotów, które są często wykorzystywane w warunkach przemysłowych. Zaprogramuj model tak, by działał precyzyjnie i przesunął objekty przy użyciu technologii wykrywania podczerwieni.

- Jak pracować z robotami stacjonarnymi.
- O poleceniu REPEAT UNTIL.



# Zawartość



## Teoria

- 03** Czego się nauczymy
- 03** Historia robotyki
- 05** Definicja robota
- 07** Wykorzystanie w życiu codziennym
- 09** Programowanie



## Instrukcja obsługi

- 11** Filozofia Engino Robotics
- 13** Sterownik MINI 2.0
- 14** Urządzenia peryferyjne, kable i porty
- 15** Przykład programowania ręcznego
- 16** Instalacja oprogramowania KEIRO
- 16** Aktualizacja firmware sterownika MINI 2.0
- 16** Interfejs KEIRO
- 30** Programowanie diagramu przepływu danych
- 31** Przykłady kodu



## Eksperymenty

- 33** Budowa i programowanie
- 34** Ruch robotyczny i IF
- 35** WHILE i bramki logiczne
- 36** Przykład z życia: ramię przemysłowe



## Instrukcje budowania

- 37** Wskazówki dotyczące budowania
- 38** Robot
- 43** Wentylator
- 46** Hexapod
- 52** Robot Twister
- 58** Skorpion



## Czego się nauczymy

„W gruncie rzeczy robotyka jest o nas. To dyscyplina polegająca na naśladowaniu naszego życia, zastanawiania się, jak właściwie funkcjonujemy”. Ten cytat profesora informatyki Roda Grupena najlepiej opisuje sposób, w jaki naukowcy postrzegają robotykę: jako symulację funkcji życiowych z wykorzystaniem mechaniki i komputerów. Ale czy kiedykolwiek ludzkość będzie w stanie stworzyć roboty, które mają uczucia i mogą podejmować prawdziwie własne decyzje w sytuacjach, których nie przewiduje język programowania? Cokolwiek przyniesie przyszłość, dzisiejsi młodzi ludzie muszą mieć jasny obraz tego, czym są roboty, począwszy od podstaw robotyki i nauk programowania, a skończywszy na budowaniu własnych, w pełni funkcjonalnych urządzeń! Kolejne strony zostały zaprojektowane z myślą o tym!

Ta broszura zawiera obszerną **sekcję teoretyczną** z wyzwaniem konstrukcyjnym i interesującymi faktami, dzięki czemu możesz dowiedzieć się wszystkiego o robotach i ich zastosowaniu w życiu codziennym. Broszura zawiera również **Instrukcję obsługi**, która szczegółowo wyjaśnia wszystkie aspekty innowacyjnego systemu robotycznego Engino. Odkryj wszystkie reguły naukowe poprzez **eksperymenty**, dzięki wykonywaniu instrukcji krok po kroku i fascynującym ćwiczeniom. Postępuj zgodnie z **instrukcjami** budowania i zmontuj ekscytujące modele robotów, takich jak **skorpion**, **ameba**, zautomatyzowany dom, wentylator, hexapod, wskaźnik, łopata, twister, kostka i robot. Wiele innych materiałów edukacyjnych jest dostępnych online!



*Automatyczne urządzenia zostały znalezione w hieroglifach*

Starożytni wynalazcy i rzemieślnicy rozpoczęli budowę wczesnych „robotów”. Miały one różne formy, od prostych do skomplikowanych urządzeń, które pozornie samodzielnie wykonywały różne zadania. Stąd nazywane są automatonami, co w tłumaczeniu z języka greckiego oznacza „działanie z własnej woli”, mimo że technicznie tak się nie działo. Już w IV wieku p.n.e. Archytas rzekomo stworzył parowe urządzenie latające zwane „gołębiem”. W tamtych czasach ludzie twierdzili, że jego maszyna może latać na odległość 200 metrów! Heron z Aleksandrii (10 - 70 n.e.) był również wynalazcą, który miał wiele kreatywnych pomysłów, takich jak automatyczne otwieranie drzwi świątyń, automaty do wydawania napojów i maszyny napędzane parą lub wiatrem, takie jak jego słynny „aeolipile”.



*Robotyka nie mogłaby istnieć bez programowania komputerowego*

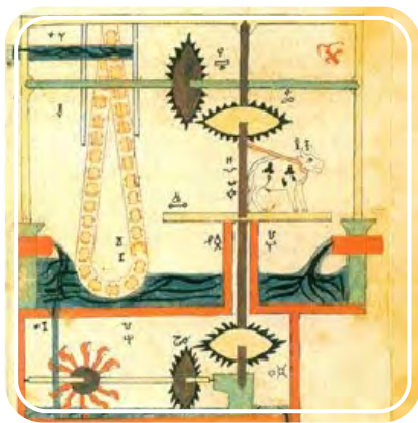
## Historia robotyki

Historia robotyki sięga czasów starożytnych. Już w antyku ludzie myśleli o sztucznych konstrukcjach, które mogłyby zastąpić ludzi wykonywaniem różnych zadań, jakby z własnym umysłem. Jeden z mitów greckich opowiada historię Talosa, gigantycznego humanoida z brązu, który został zbudowany przez Hefajstosa (greckiego boga kowali) w celu ochrony Krety. Relacje o stworzeniach podobnych do robotów można znaleźć na całym świecie, od Norwegii po Bliski Wschód, Indie i Chiny, w różnych formach: teksty, rysunki, obrazy, a nawet starożytne egipskie hieroglify.



*Nowoczesna replika turbiny (bani) Herona*





Rysunek Al-Dżazariego przedstawiający pompę łańcuchową z napędem wodnym

W 1898 roku serbsko-amerykański wynalazca Nikola Tesla zademonstrował pierwszy statek sterowany radiowo. Prawdziwe roboty, czyli takie, które potrafią odbierać informacje zwrotne z otoczenia i na nie reagować, pojawiły się dopiero w XX wieku. W 1948 roku William Grey Walter stworzył małe roboty-żółwie z czujnikami światła i dotyku, które mogły wyszukiwać „pożywienie”. Kilka lat później George Devol opatentował swojego robota Unimate, który był w stanie wykonywać prace przemysłowe. Został on zainstalowany w 1961 roku w fabryce, aby podnosić gorące kawałki metalu z maszyny odlewniczej i układać je w stos.

W średniowieczu możemy znaleźć wiele przykładów automatonów, zwykle w formie zwierzęcej lub humanoidalnej. Ludzie budowali zabawne maszyny zaprogramowane do wykonywania prostych zadań, takich jak poruszanie stopami lub obracanie głowami, przy użyciu precyzyjnej technologii zegarowej. Muzułmański inżynier Al-Dżazari (1136–1206) opisał 100 zautomatyzowanych urządzeń mechanicznych w swojej książce: „Al-Jami Bain Al-Ilm Wal-Amal Al-Nafi Fi Sinat'at Al-Hiyal” (Księga wiedzy o przemysłnych urządzeniach mechanicznych). Później francuski wynalazca i artysta Jacques Vaucanson (1709-1782) odegrał ważną rolę w rozwoju robotyki, próbując pokazać, jak rzeczy działają w przyrodzie. Wśród jego wynalazków znalazły się figury grające na prawdziwych instrumentach (flecie i tamburynie) oraz słynna „trawiąca kaczka”, która imitowała trawienie pokarmu.



Ramię robotyczne Unimate

W latach sześćdziesiątych inżynierowie zajmowali się problemem ruchu robotów, tworząc ramiona i nogi robotów inspirowane naturą. W latach 70-tych i 80-tych XX wieku zaawansowana technologia i niewielkie części komputerowe umożliwiły umieszczenie wszystkich komponentów bezpośrednio na robocie zamiast podłączania ich przewodami do zewnętrznego komputera. Od lat 90. roboty stały się jeszcze bardziej zaawansowane, wykonując wiele złożonych zadań. Obecnie są wykorzystywane w każdym aspekcie ludzkiego życia, od domowej rozrywki i pomocy (jak robot-pies AIBO, 1999 i robot odkurzacz Roomba, 2008), po zastosowania przemysłowe (np. budowanie samochodów) i eksploracje (jak latający robot Epson, 2004), nawet w kosmosie (jak roboty eksploracyjne na Marsie, 2004). Jednak to na razie dopiero początek, bo przyszłość robotyki zapowiada się bardzo obiecująco!



## Czy wiedziałeś, że...

Pisarz science fiction Isaac Asimov opracował „**Trzy prawa robotyki**” w swoim opowiadaniu „Runabout” („Zabawa w berka”) opublikowanym w 1942 r. Są to:

- 1) Robot nie może skrzywdzić człowieka, ani przez zaniechanie działania dopuścić, aby człowiek doznał krzywdy.
- 2) Robot musi być posłuszny rozkazom człowieka, chyba że stoją one w sprzeczności z Pierwszym Prawem.
- 3) Robot musi chronić samego siebie, o ile tylko nie stoi to w sprzeczności z Pierwszym lub Drugim Prawem.



Isaac Asimov (1920 - 1992)



Dwa rodzaje odkurzaczy Roomba

## Definicja robota

Termin „robot” po raz pierwszy wprowadził czeski pisarz Karel Čapek w 1920 roku w swoim dramacie R.U.R., pochodzi od słowa oznaczającego „pracę” w języku słowiańskim. To pokazuje nam, w jaki sposób ludzie pierwotnie rozumieli ideę robotów:

„urządzenia wykonujące zadania, które są zbyt niebezpieczne, skomplikowane lub po prostu nudne dla ludzi”. Jednak bardziej naukowa definicja mówi, że robot to urządzenie mechaniczne, które odbiera informacje zwrotne z otoczenia i jest w stanie odpowiednio na nie zareagować. Dlatego roboty mają, do pewnego stopnia, rzeczywiste autonomiczne zachowanie oraz sposób, w jaki same myślą i działają. Zależy to jednak od ich języka programu i z góry określonych poleceń.



*Zrobotyzowane drzwi metra z czujnikiem dotykowym*

## Rodzaje robotów

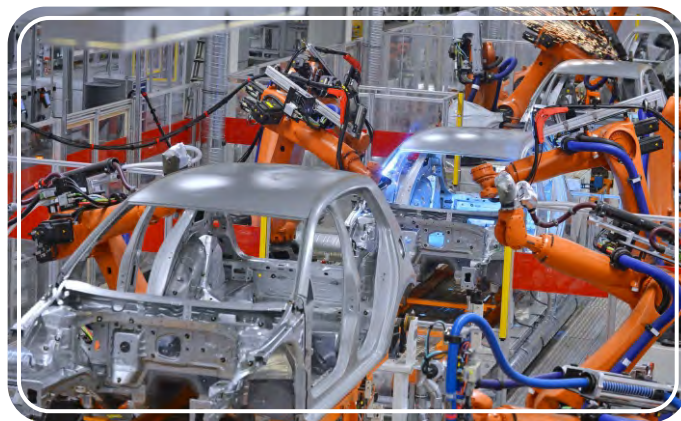
Roboty mają różne formy i kształty i mogą wykonywać jedno lub wiele zadań w różnych ustawieniach i warunkach. W ten sposób jeden robot może należeć do więcej niż jednej kategorii. Jak więc możemy odróżnić jeden od drugiego? Możemy zastosować kategoryzację według funkcji robota. Daje nam to kilka informacji o użyteczności robota i wyobrażenie o tym, jak powinien wyglądać lub jakie komponenty powinny mieć, aby działał poprawnie. Poniżej opisano główne typy według funkcji:



*Przemysłowe ramię robota*

**Roboty przemysłowe**, czyli zautomatyzowane roboty wykorzystywane w fabrykach do podnoszenia ciężkich przedmiotów lub wykonywania rutynowych prac, takich jak spawanie, malowanie i przenoszenie materiałów.

**Roboty rozrywkowe**, czyli roboty stworzone głównie do zabawy i mające mniej praktyczne zastosowanie, np. roboty-zabawki. A blue arrow points to the right from the text above.



*Ramiona robotyczne stosowane na linii montażowej samochodów*

Na przykład automatyczne otwieranie drzwi (w sklepach, hotelach itp.) jest zrobotyzowane – posiadają czujnik ruchu, który wysyła ciągły sygnał w podczerwieni. Jeśli ten sygnał zostanie zakłócony, ponieważ ktoś znajdzie się przed niewidzialną wiązką, centralny procesor wydaje polecenie otwarcia drzwi. Po kilku sekundach drzwi zamykają się i procedura jest powtarzana. Inne przykłady robotów obejmują domowe systemy grzewcze i światła uruchamiane ruchem. Jednak zdalnie sterowane samochody czy zabawki nie są robotami. Po prostu otrzymują bezpośrednie polecenia ruchu przez kontroler. Ponieważ nie jest to informacja zwrotna, nie są w stanie podejmować decyzji dotyczących otaczających ich obiektów, np. zatrzymaj się przed uderzeniem w ścianę.



*Robot do czyszczenia basenu*



*Robot-zabawka*





Robot do rozbrajania bomb

**Roboty medyczne**, wykorzystywane w służbie zdrowia (szpitale, apteki, przychodnie i centra medyczne) do szkolenia lekarzy czy nawet wykonywania dokładnych procedur (np. robot chirurgiczny).

◀ **Roboty wojskowe**, które są częścią zaawansowanych armii i sił policyjnych. Służą one zamiast ludzi, zwykle w sytuacjach zagrożenia życia, do transportu amunicji, składowania bomb, a nawet angażowania się w walki wojenne, np. małe roboty czołgowe.

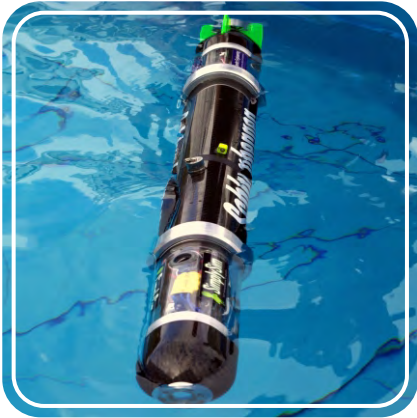
**Roboty domowe i usługowe**, czyli roboty przeznaczone do komercyjnego codziennego użytku w domu lub w pracy, ułatwiające nieco życie (np. robot odkurzający).

**Roboty eksploracyjne**, czyli roboty używane do eksploracji trudnych środowisk, do których dostęp jest zbyt trudny dla ludzi, takich jak jaskinie, oceany (np. roboty wodne) czy nawet inne planety.

**Roboty wirtualne**, czyli roboty używane w warunkach wirtualnych. Zwykle są to jakieś duże, zakrywające oczy okulary z ekranami, które są połączone ze specjalnymi rękawicami, aby zobaczyć i poczuć świat rozszerzonej rzeczywistości.



Sprzęt rzeczywistości wirtualnej



Robot wodny



## Czy wiedziałeś, że...

Odbyna się specjalny turniej piłkarski dla robotów o nazwie „RoboCup”! Pierwsze roboty, które wzięły udział w automatycznej grze w piłkę nożną, powstały w Japonii w 1997 roku. Zawodnicy byli w stanie zlokalizować piłkę i kopnąć ją. Producenci tych robotów postawili sobie za cel, aby w 2050 roku robotyczna drużyna piłkarska rywalizowała ze zwycięską drużyną mistrzostw świata i wygrała! Czy myślisz, że do tego czasu technologie robotyczne osiągną taki poziom umiejętności piłkarskich?



Roboty-piłkarze

**Roboty hobbystyczne** to roboty, które zostały stworzone głównie przez entuzjastów robotów, a nie przez duże firmy. Ich cel może być eksperymentalny, np. testowanie różnych komponentów i technologii, lub po prostu rekreacyjny, np. latający dron.

**Roboty współzawodniczące**, czyli roboty stworzone do konkurowania z innymi robotami w różnych wyzwaniach. Ta kategoria jest podobna do poprzedniej, ponieważ zawody organizowane są najczęściej przez fanów robotów i chcą oni sprawdzić umiejętności robotów lub zdobyć cenne doświadczenie. Konkursy odbywają się również wśród zespołów programistów robotów z całego świata, pomagając im ulepszać ich projekty.



Latający robot (dron)

## Zastosowanie w życiu codziennym

XXI wiek to z pewnością wiek robotyki i zaawansowanej technologii! Roboty można znaleźć wszędzie, nie tylko te o humanoidalnym wyglądzie, który ludzie znają najlepiej, ale często w bardzo prostych formach, takich jak systemy wykrywania ciepła w domach i czujniki odległości w samochodach. Zastosowania systemów robotycznych w życiu codziennym są ogromne, a ludzie stają się od nich coraz bardziej zależni. Przeczytaj poniżej o kilka charakterystycznych przykładach wykorzystania robotów i rzuć sobie wyzwanie, budując i programując proponowane modele firmy **Engino Robotics**.

 **Budowanie** **Wyzwanie**

 instrukcje można znaleźć online

**Unikanie kolizji:** nowoczesne pojazdy są wyposażone w różnego rodzaju czujniki, dzięki czemu kierowcy nie muszą myśleć o tylu rzeczach podczas jazdy. Detektor deszczu może uruchamiać wycieraczki, gdy pada deszcz, lub czujnik światła może określić, czy jest dzień, czy noc, aby odpowiednio włączyć światła.

Jeśli chodzi o bezpieczeństwo, wiele samochodów ma czujniki pomiaru odległości z przodu i z tyłu, które ostrzegają kierowcę o prawdopodobnej kolizji, zwłaszcza podczas parkowania lub manewrowania w ciasnych miejscach. Czujniki te mogą ocalić życie, jeśli w pobliżu samochodu znajdują się ludzie lub zwierzęta, a kierowca ich nie zauważa!

*Zbuduj model Engino „Pointer” i eksperymentuj, jak unikać kolizji. kieruj czujniki podczerwieni w dół, aby podążały określoną ścieżką, lub obróć je na boki, aby ominąć przeszkody po lewej lub prawej stronie pojazdu.*



Engino® „pointer” model

 **Budowanie** **Wyzwanie**

 instrukcje można znaleźć online

**Zrobotyzowane domy:** nowoczesne domy stają się coraz bardziej „inteligentne” a wiele wcześniej wykonywanych ręcznie funkcji jest wykonywanych automatycznie! Na przykład automatyczne drzwi otwierają się i zamykają za pomocą czujników ruchu, podczas gdy czujniki ciepła wykrywają wahania temperatury i aktywują system chłodzenia lub ogrzewania. Inżynierowie eksperymentują z inteligentnymi lodówkami, które będą w stanie samodzielnie uzupełniać zapasy, wykrywając brak określonych produktów (mleko, jajka, masło) i zamawiając je online!

*Eksperymentuj z automatyką domową, budując model „zautomatyzowanego domu” ® Engino (instrukcje można znaleźć online). Wpraw drzwi i wentylator w ruch za pomocą czujników podczerwieni.*



Model Engino „zautomatyzowany dom”



### Czy wiedziałeś, że...

12 listopada 2014 r. po raz pierwszy w historii robotyczny moduł (ładownik) o nazwie Philae wylądował na powierzchni komety! Ten imponujący wyczyn był częścią misji kosmicznej Rosetta, robotycznej sondy kosmicznej, która orbitowała i badała kometę 67P od stycznia 2014 roku. Misja ma na celu sprawdzenie, czy kometa może dostarczyć klucza do rozszyfrowania pochodzenia Układu Słonecznego i życia na Ziemi.



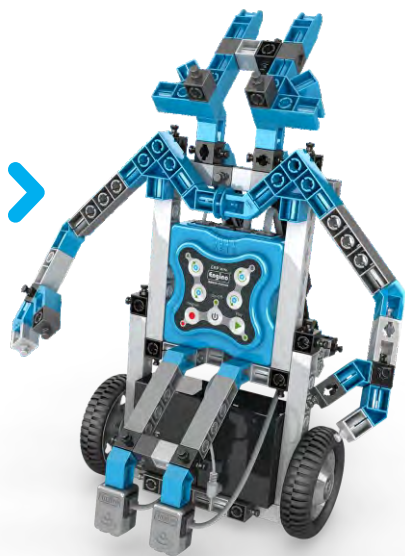
Reprezentacja misji kosmicznej Rosseta



## Budowanie Wyzwanie

**Niebezpieczne misje:** roboty skutecznie zastępują ludzi w niebezpiecznych sytuacjach. Ludzie nie muszą podejmować ryzyka, a dzięki zaawansowanym umiejętnościom robotów misje są wykonywane szybciej i lepiej. Na przykład robot rozpoznaje niebezpieczne materiały, które może usunąć zdalnie przy pomocy człowieka. Operator korzysta z czujników robota, aby widzieć i manipulować bombą.

Zbuduj model Engino „Robot” (strony 38-42) i umieść czujniki podczerwieni skierowane w dół, aby podążały określoną ścieżką lub skierowane do przodu, aby wykryć „niebezpieczne” obiekty, o których będą ostrzegać ruchem.



Model Engino „robot”

## Budowanie Wyzwanie

**Zastosowanie przemysłowe:** kolejne coraz powszechniejsze zastosowanie robotyki dotyczy sektora przemysłowego, w którym roboty zastępują pracę ludzi. Są to zwykle roboty stacjonarne lub poruszające się po przewodzie linowym, które mają elastyczne ramię do podnoszenia ciężkich przedmiotów lub wykonywania delikatnych prac (takich jak spawanie). Idealnie sprawdzają się na fabrycznych liniach montażowych. Ludzie boją się, że zautomatyzowane procedury doprowadzą do redukcji etatów, ale tak nie jest, ponieważ powstają nowe stanowiska do obsługi i naprawy zrobotyzowanych maszyn, a jednocześnie poprawia się jakość produktów.

Zbuduj Engino „Twister” (strony 52-57) i kieruj jego ruchem za pomocą technologii wykrywania podczerwieni.



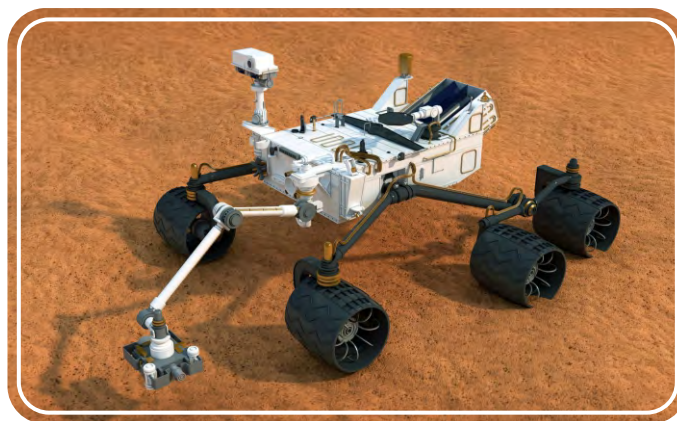
Model Engino „twister”

## Eksploracja kosmosu

Ludzkość zawsze fascynowały cuda wszechświata i możliwość odkrywania nowych światów, a nawet odkrywania obcych form życia. Jednak warunki poza atmosferą ziemską mogą być katastrofalne dla ludzi: burze słoneczne, brak tlenu, pola grawitacyjne i bezwzględne temperatury zerowe (-273 stopnie Celsjusza) to tylko kilka z możliwych zagrożeń. Wszystko to wymaga bardzo drogich skafandrów kosmicznych oraz lat treningu i planowania tylko na krótki spacer dookoła Ziemi lub na Księżycu. Jeśli chodzi o odległość (np. podczas podróży na inną planetę), odkrywcy muszą liczyć się z tym, że powrócą na Ziemię po dziesięcioleciach lub wcale!



Łazik marsjański „Opportunity”

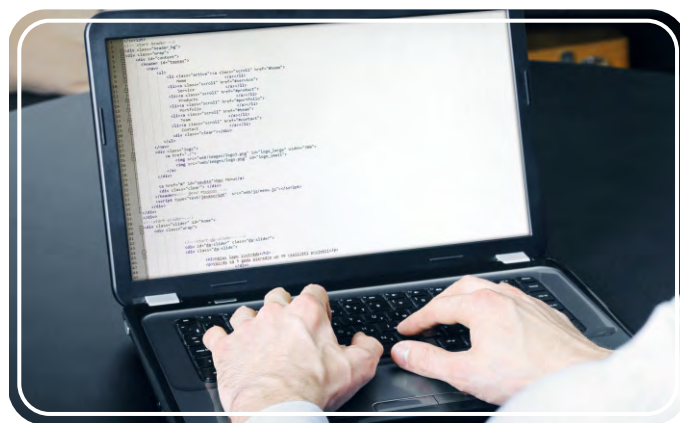


Łazik marsjański „Curiosity”

Na szczęście istnieją roboty! Są idealne do eksploracji kosmosu, ponieważ eliminują wszystkie problemy związane z przetrwaniem i nie mają nic przeciwko spędzaniu więcej czasu na misji. Dzięki swoim zaawansowanym możliwościom mogą badać powierzchnię planety, analizować jej glebę i atmosferę, szukać wody i innych substancji chemicznych oraz przesyłać na Ziemię tony cennych zdjęć o wysokiej rozdzielczości. Obecnie istnieją dwa aktywne łaziki do eksploracji Marsa, Opportunity i Curiosity, a także różne inne roboty, które krążą wokół asteroid, komet i planet, co czyni je godnymi reprezentantami ludzkości!

# Programowanie

Oprócz elementów mechanicznych, również programowanie jest niezbędne do działania robota. Jak wspomniano wcześniej, urządzenie jest uważane za zrobotyzowane tylko wtedy, gdy jest zdolne do interakcji ze swoim otoczeniem. Robot interpretuje wszystkie otrzymane informacje i decyduje za pośrednictwem mikrokontrolera o odpowiednich działaniach zgodnie ze swoim programem. Producenci robotów zwykle udostępniają własne języki programowania, co skutkuje wieloma rodzajami oprogramowania robotów. Na szczęście istnieje kilka podstawowych pojęć, którymi należy się kierować przy tworzeniu programu, które zostaną opisane poniżej.



Programista w pracy

## Języki programowania

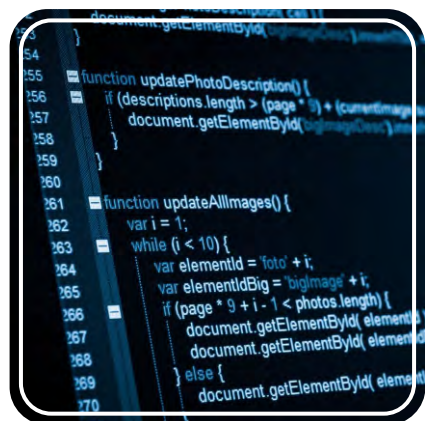
Najważniejszym elementem rozwoju każdego komputera lub robota jest jego język programowania. Ogólnie język programowania to zbiór reguł gramatycznych i słownictwa służących do instruowania robota, aby wykonywał określone zadania. Na język składają się dwa główne komponenty: forma (składnia) i znaczenie (semantyka) tekstu. Istnieją dwa ogólne typy języków programowania: niskopoziomowe, które wykorzystują maszynowy sposób rozumienia (kod maszynowy z liczbami i symbolami) oraz wysokopoziomowe, które zawierają język ludzki (w formie tekstowej) i automatyzację, dzięki czemu proces programowania jest prostszy i łatwiejszy do zrozumienia. W oprogramowaniu komputerowym wykonywalny opis programu (utworzony za pomocą języka programowania) nazywany jest „kodem źródłowym”. W kodzie źródłowym istnieją algorytmy, które są konkretnymi zestawami operacji, które należy wykonać krok po kroku, utworzonymi przez pojedyncze elementy programowania zwane instrukcjami (wyjaśnione poniżej).

Każdy, kto chciałby stworzyć program komputerowy, powinien bardzo uważnie wybrać język programowania, ponieważ są one zaprojektowane i używane do różnych celów. Najpopularniejsze języki wysokiego poziomu to:

- C++ jest prawdopodobnie najczęściej używanym językiem, w którym tworzona jest większość aplikacji.
- Języka Java można używać na dowolnym urządzeniu, na którym zainstalowana jest wirtualna maszyna języka Java (JVM), niezależnie od systemu.
- JavaScript to język sieciowy używany do wykonywania operacji na komputerach użytkowników zamiast na serwerze sieciowym.
- Python jest uważany za jeden z najłatwiejszych do opanowania języków programowania.



Język HTML jest używany specjalnie do tworzenia witryn internetowych.



Instrukcje są zapisywane jako wiersze tekstu

## Instrukcje

W programowaniu komputerowym instrukcja jest poleceniem, która ogólnie mówi komputerowi, co ma robić. Prawdopodobnie możesz sobie wyobrazić, że te polecenia muszą być wystarczająco dokładne, aby robot działał poprawnie. Ponadto równie ważna jest sekwencja instrukcji, czyli kolejność zadań. Wyobraź sobie, że postępujesz zgodnie z instrukcjami dotyczącymi pieczenia ciasta. Jeśli kroki w przepisie są nieprawidłowe lub są napisane w złej kolejności, istnieje duże prawdopodobieństwo, że skończysz z bałaganem w kuchni! W tekstowych językach programowania polecenia są zwykle pisane wiersz po wierszu i zawierają znaki specjalne dla różnych zadań. Jest to symulowane w oprogramowaniu KEIRO, które zawiera okno (KEIRO Code) używające pseudojęzyka (a nie faktycznego języka programowania), umożliwiając użytkownikom wyświetlanie różnych poleceń zgodnie z ich kolejnością sekwencji jako wiersze tekstu.





*Maszyna wykorzystująca programowanie ręczne*

Używając ręcznego programatora (teach pendant), który przypomina tradycyjny kontroler, użytkownik steruje robotem zdalnie tak, aby wykonywał określone ruchy np. podniósł część samochodową i umieścił ją na przenośniku taśmowym. Podczas programowania ręcznego sekwencja poleceń jest generowana i zapisywana automatycznie, dzięki czemu robot może kontynuować pracę w dokładnie taki sam sposób. Programiści mają możliwość dalszej edycji kodu, wprowadzając wszelkie zmiany związane z szybkością, dokładnością i płynnością ruchu. Kiedy wszystko jest ustawione, gotowy program jest wysyłany z powrotem do robota.



## Programowanie ręczne

Zwykle procedura tworzenia fabrycznego programu robotycznego polega na napisaniu wszystkich poleceń w języku programowania, sprawdzeniu, czy działa on w określonych warunkach i dokonaniu różnych korekt, aż do uzyskania zadowalającego wyniku. Ten proces wymaga skomplikowanych umiejętności kodowania i często jest ograniczony w określonym zakresie ruchu, np. robot po prostu przenosi obiekty z jednego miejsca do drugiego. Ale co, jeśli chcemy naśladować dokładne ruchy pracownika, aby zaprogramować robota do wykonywania rzeczywistej pracy człowieka bez konieczności pisania złożonych instrukcji programu? Nowoczesna technologia robotyczna umożliwia nam to właśnie dzięki programowaniu ręcznemu.



*Pracownik korzystający z ręcznego programatora (teach pendant)*



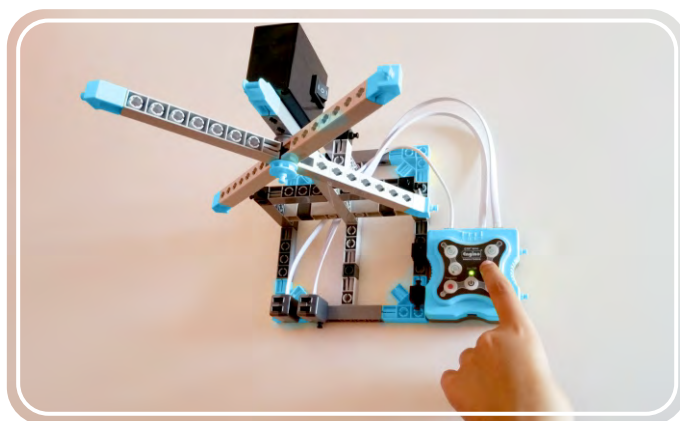
## Czy wiedziałeś, że...

Pierwszą programistką była angielska matematyk Augusta Ada King (hrabina Lovelace). Ada napisała pierwszy algorytm w historii, który miał być wykonywany przez maszynę w 1837 roku. Dokładnie był to algorytm obliczania liczb Bernoulliego przy użyciu silnika analitycznego, zaproponowany przez Charlesa Babbage'a pomysł na mechaniczny komputer ogólnego przeznaczenia.



*Obraz Augusty Ady King (1815 - 1852)*

Programowanie ręczne jest jedną z głównych funkcji oprogramowania Engino KEIRO. Jest to bardzo przydatny sposób tworzenia programu w rzeczywistych warunkach bez znajomości samego oprogramowania. Użytkownik może po prostu nacisnąć przyciski na kontrolerze ERP MINI 2.0 i wgrać sekwencję poleceń, które robot wykona w czasie rzeczywistym. Po zapisaniu programu można go w każdej chwili odtworzyć ręcznie, nawet w pętli. Ponadto, jeśli użytkownik chce doprecyzować zmienne poleceń i dodać instrukcje warunkowe, robi to w bardzo prosty sposób, podłączając urządzenie do komputera i wysyłając program do oprogramowania KEIRO. Po zakończeniu regulacji program można przesłać z powrotem do robota w celu jego przetestowania.



*Programowanie ręczne za pomocą sterownika MINI 2.0*



# Instrukcja obsługi

## Filozofia Engino Robotics

Rozwój intelektualny pomaga uczniom rozwijać swoje pomysły, jednocześnie zdobywając nowe informacje. Jest to niezbędne do rozwijania i doskonalenia kreatywności i umiejętności laboratoryjnych. Połączenie kontrolera MINI 2.0 z podobnym do Scratch oprogramowaniem KEIRO (patent w trakcie) jest idealnym rozwiązaniem do nauki robotyki. Kontroler MINI 2.0 umożliwia **cztery połączone ze sobą sposoby programowania**, które można łączyć tak, aby każdy użytkownik mógł wybrać metodę odpowiednią dla swojego wieku i doświadczenia.



## Programowanie ręczne

Pierwszy poziom programowania dotyczy fizycznej interakcji z materiałem do nauki. Naciskanie przycisków to metoda, którą znają wszystkie dzieci. Uczniowie mogą nagrać dowolną sekwencję poleceń za pomocą przycisku „Program”, zapisać ją w urządzeniu, a następnie ponownie nacisnąć przycisk „Play”, aby powtórzyć czynność. Ta metoda programowania jest niezbędna do nauki procedury komend i sekwencji zdarzeń.



## Symulator

Jest to specjalne okno w oprogramowaniu KEIRO, które symuluje funkcje prawdziwego kontrolera, z przyciskami cyfrowymi zamiast fizycznych. Po podłączeniu kontrolera MINI 2.0 do urządzenia (komputera lub tabletu), użytkownik może nagrać program i może załadować program i zobaczyć informacje zwrotne od robota. Podczas nagrywania programu generowany jest schemat blokowy wyświetlany w postaci bloków wizualnych.



```

KEIRO Code
BEGIN:
  MOTOR: PORT: A
        STATE: ON FOREVER
        Direction: CLOCKWISE
        Speed: 100
        Delay (s): 0
        AFTER PREVIOUS
  IF: TOUCH: Port: 1 == TRUE
     LED:   Port: A
          State: ON FOR DURATION
          Delay (s): 0
          Duration (s): 1.5
          AFTER PREVIOUS
ENDIF
END

```



## Kod KEIRO

„Pseudojęzyk” (nie jest to prawdziwy język programowania) stworzony specjalnie dla oprogramowania KEIRO. Wykorzystuje wiele pojęć znanych z programowania, takich jak BEGIN, IF, END, itp. Jest idealnym narzędziem do wprowadzania zaawansowanego programowania, ponieważ oferuje szybki podgląd programu w formie tekstowej.

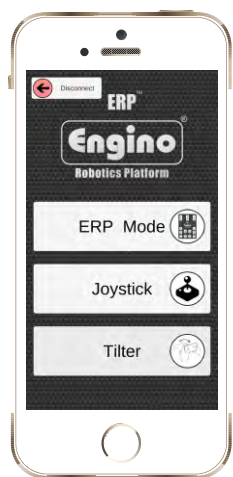


## Diagram przepływu danych

„Diagram przepływu” (Flow Diagram) jest głównym elementem programistycznym oprogramowania KEIRO. Tutaj użytkownik może wykorzystać wszystkie wyjścia i wejścia (output, input) robota do tworzenia bardziej wyrafinowanych programów. Platforma została zaprojektowana w języku programowania blokowego, który umożliwia stopniowe przejście od programowania ręcznego do programowania cyfrowego. Udowodniono, że środowiska programowania typu drag-and-drop (przeciągnij i upuść) są cennymi narzędziami edukacyjnymi, które zapewniają łatwy sposób interakcji z rzeczywistym światem i rozwijania intuicyjnego interfejsu człowiek-maszyna.

## Połączenie z urządzeniami typu smart

Zgodnie z nowoczesnymi osiągnięciami technologicznymi, kontroler MINI 2.0 może być również sterowany za pomocą urządzenia typu smart (telefonu komórkowego lub tabletu) poprzez połączenie Bluetooth. Oprogramowanie KEIRO to oparta na Scratch graficzna platforma programistyczna, której można używać na **komputerze PC** lub **urządzeniu inteligentnym**. Programowanie robota odbywa się w zabawny sposób, wykorzystując wszystkie możliwości w taki sposób, aby uczniowie zawsze chętnie pracowali nad swoimi projektami.



Oprogramowanie KEIRO

EnginoRobot BT

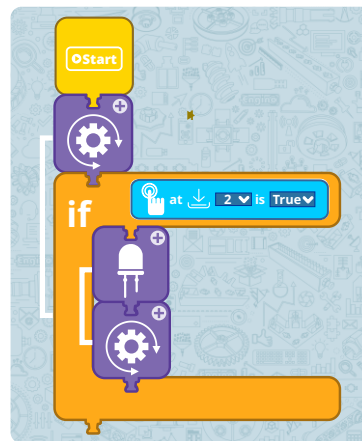
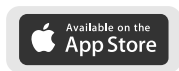


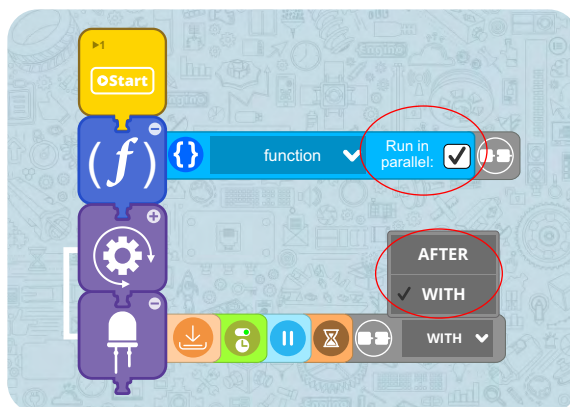
Diagram przepływu jest tworzony przez proste przeciąganie i upuszczanie bloków akcji



Ponadto **EnginoRobot BT**, dostępny w Google Play lub Apple Store, praktycznie symuluje interfejs rzeczywistego kontrolera. Użytkownicy mogą sterować modelem, nagrywać i odtwarzać program tak samo, jak mogliby to zrobić, naciskając fizyczne przyciski na kontrolerze MINI 2.0. Wszystko to można osiągnąć zdalnie i cyfrowo! Aplikacja oferuje dwie dodatkowe metody sterowania: **Joystick** oraz **Tilter**. Pobierz aplikację za darmo i rozszerz możliwości programowania ręcznego.

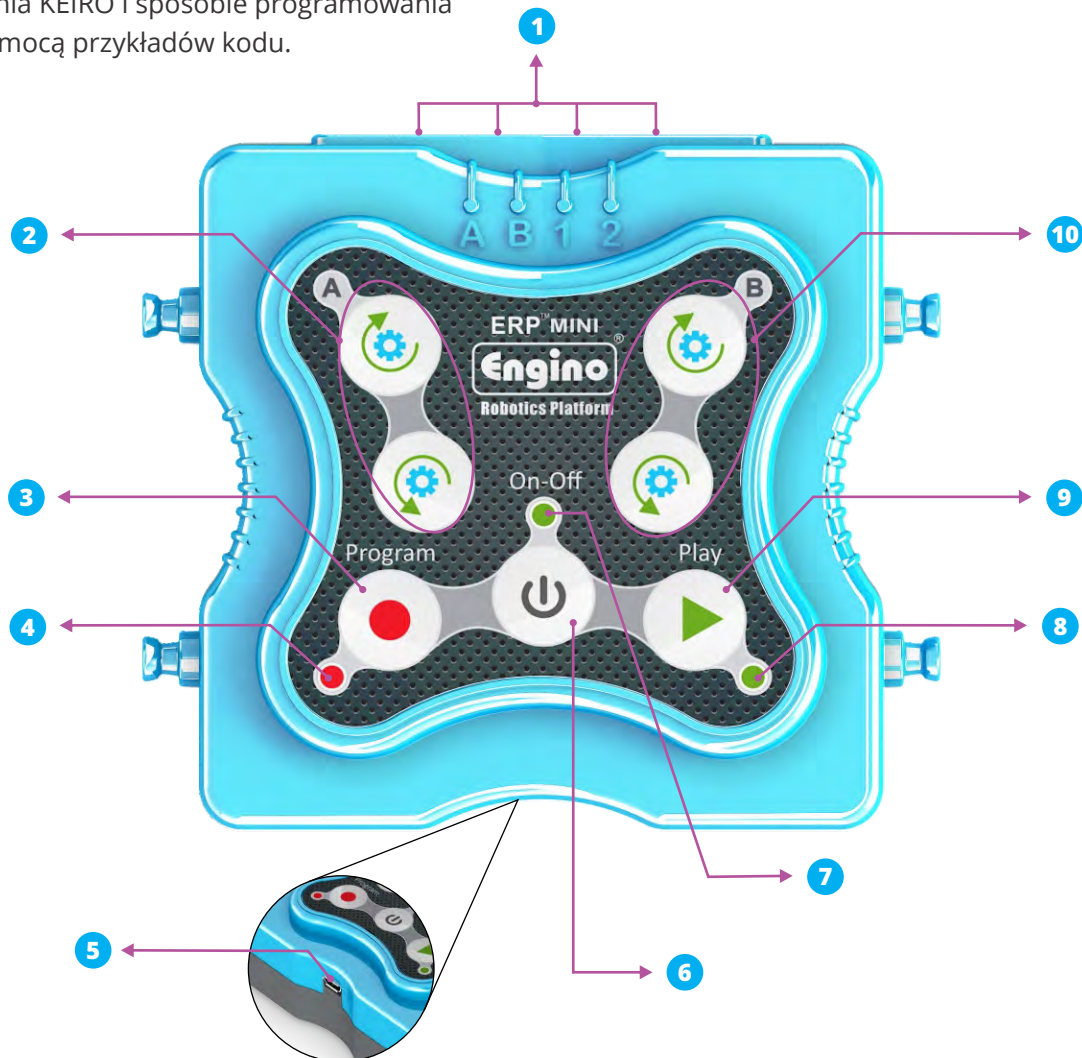
## Programowanie równoległe

Główną innowacją Engino w robotyce edukacyjnej jest funkcja programowania równoległego. Bloki akcji w oprogramowaniu KEIRO zawierają przydatną funkcję, która pozwala użytkownikowi wybrać, czy polecenie zostanie wykonane **po**, czy **razem** z poprzednim. Dodatkowo oprogramowanie może wywołać funkcję (krotka komand, tuple of commands) z opcją uruchomienia **równoległe** (run in parallel) z kolejnymi blokami akcji kodu. Te opcje zmniejszają złożoność programowania i pozwalają na łatwe wykonanie skomplikowanych zadań.



## Kontroler MINI 2.0

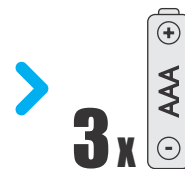
Kontroler Engino MINI 2.0 to zrobotyzowane urządzenie składające się z kontrolera głównego, kilku przycisków i wskaźników. Poprzez 4 porty urządzenia kontroler można podłączyć do urządzeń peryferyjnych, takich jak silniki, diody LED, czujniki podczerwieni i dotykowe. Kontroler MINI 2.0 to potężne, fleksybilne i łatwe w użyciu narzędzie. Niniejsza instrukcja zawiera informacje o jego głównych funkcjach, instrukcje dotyczące instalacji oprogramowania KEIRO i sposobie programowania robotów za pomocą przykładów kodu.



- |  |   |
|--|---|
| <b>1</b> <b>PORTY A, B, 1 i 2 RJ</b><br><i>łączy urządzenia peryferyjne z kontrolerem</i>                              | <b>6</b> <b>PRZYCISK POWER</b>  |
| <b>2</b> <b>PRZYCISKI MOTOR A</b><br><i>zgodnie z ruchem wskazówek zegara lub przeciwnie do ruchu wskazówek zegara</i> | <b>7</b> <b>LED ON/OFF</b><br><i>zapala się, gdy kontroler jest włączony</i>                                  |
| <b>3</b> <b>PRZYCISK PROGRAM</b><br><i>ładuje sekwencję komend</i>   | <b>8</b> <b>LED PLAY</b><br><i>włącza się podczas odtwarzania programu</i>                                    |
| <b>4</b> <b>LED PROGRAM</b><br><i>bmiga podczas nagrywania</i>   | <b>9</b> <b>PRZYCISK PLAY</b><br><i>odtwarza zapisany program</i>   |
| <b>5</b> <b>PORT mini USB</b><br><i>łączy kontroler z PC</i>   | <b>PRZYCISKI MOTOR B</b><br><i>zgodnie z ruchem wskazówek zegara lub przeciwnie do ruchu wskazówek zegara</i> |
|  | <b>10</b>   |



**Baterie:** Aby włączyć kontroler MINI 2.0, należy z tyłu zainstalować trzy (3) baterie AAA. Do zdjęcia pokrywy baterii potrzebny będzie śrubokręt krzyżakowy. Umieść baterie z zachowaniem prawidłowej biegunowości, zgodnie ze znakami + i - pokazanymi na każdej pozycji baterii. Po włożeniu baterii zakręć pokrywę z powrotem na swoje miejsce.



**Wskazówki dotyczące bezpieczeństwa:** Nie przeładuj baterii. Nie mieszaj baterii alkalicznych, standardowych i akumulatorów. Nie ładuj baterii jednorazowych. Nie mieszaj starych baterii z nowymi. Jeśli w najbliższej przyszłości nie planujesz używać zużytych lub starych baterii, wyjmij je z urządzenia. Nigdy nie wrzucaj baterii do ognia. Nie wolno zwierać zacisków zasilania.

## Urządzenia peryferyjne, kable i porty



Port kabla RJ

### LED (czerwona)

Użyj **lampki LED**, aby dodać ładne efekty wizualne swojemu robotowi.



Port kabla RJ

### Czujnik dotyku

Ten **przycisk** przełącznika może po naciśnięciu wywołać **akcję** i ruch.



Port kabla RJ

### Czujnik podczerwieni

Czujnik podczerwieni może być używany do **wykrywania** obiektów lub do **śledzenia linii**.



Przełącznik silnika

### Silnik prądu stałego

Silnik prądu stałego o **wysokim momencie obrotowym** i regulowanej prędkości, który może obracać kołami lub powodować **ruch** robota.



Kabel czujnika

Kabel łączący czujniki i diody LED z kontrolerem MINI 2.0. Po obu stronach znajduje się złącze RJ.



Kabel silnika

Kabel łączący silnik prądu stałego z kontrolerem MINI 2.0, z RJ na jednym końcu i złączem jack na drugim.



Kabel Mini USB

Kabel łączący kontroler MINI 2.0 z komputerem PC.

W pozycji **I** silnik pracuje **zgodnie** z kierunkiem pracy kontrolera i blokami poleceń oprogramowania KEIRO. Po przełączeniu do pozycji **II** kierunek jest odwrócony. Silnik jest wyłączony (OFF), gdy przełącznik znajduje się w położeniu **O**.



**Urządzenia peryferyjne i porty:** Kontroler Engino MINI 2.0 zawiera cztery (4) porty RJ do sterowania silnikami, diodami LED i czujnikami. Jednak różne typy urządzeń peryferyjnych działają tylko w określonych portach kontrolera. Poniższa tabela zawiera przegląd dostępnych portów dla każdego z urządzeń peryferyjnych\*.

URZĄDZENIA PERYFERYJNE	PORTY			
	A	B	1	2
	✓	✓		
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓

**Tabela 1.** Urządzenia peryferyjne i dostępne porty połączeniowe

\* Urządzenia peryferyjne (LED i czujnik dotykowy), które nie są zawarte w tym zestawie, znajdują się w zestawie edukacyjnym Engino Educational set STEM & Robotics MINI ([www.enginoeducation.com](http://www.enginoeducation.com)).

## Przykład programowania ręcznego

Kontroler MINI 2.0 umożliwia ręczne sterowanie i nagrywanie poprzez naciśnięcie membranowych przycisków kontrolera. Użytkownik może stopniowo przejść od fizycznego programowania ręcznego do sterowania za pomocą oprogramowania, stosując innowacyjną **metodę inżynierii odwrotnej!** Program nagrany na urządzeniu można łatwo wgrać do komputera lub urządzenia inteligentnego.

Po podłączeniu silnika do portu A (ustaw przełącznik silnika w położeniu I):

1. Naciśnij przycisk „Program”, aby rozpocząć nagrywanie (czerwony wskaźnik zacznie migać).
2. Przytrzymaj przycisk silnika A zgodnie z ruchem wskazówek zegara przez 3 sekundy.
3. Przytrzymaj przycisk silnika A przeciwny do ruchu wskazówek zegara przez 2 sekundy.
4. Ponownie naciśnij przycisk „Program”, aby zapisać program w pamięci.
5. Naciśnij raz przycisk „Play”, aby jeden raz uruchomić program (miga zielony wskaźnik).



### Porady i wskazówki:

- Jeśli przytrzymasz przycisk odtwarzania przez 3 sekundy, program będzie odtwarzany w nieskończonej pętli.
- Urządzenie może zapisać w swojej pamięci tylko ostatnio nagrany/wysłany program.



# Instalacja oprogramowania KEIRO

Maszyna, która wykonuje serię poleceń, nie jest prawdziwym robotem; prawdziwy robot to taki, który można zaprogramować tak, aby podejmował własne decyzje w oparciu o informacje zwrotne otrzymane z jego czujników. Jednak programowanie ręczne nie dotyczy czujników. Aby stworzyć zaawansowany program, w którym można edytować zmienne i dodawać technologię czujników, należy zainstalować **oprogramowanie KEIRO**. Można go pobrać ze strony internetowej: [www.engineerobotics.com](http://www.engineerobotics.com)

1. Przejdź do witryny [www.engineerobotics.com](http://www.engineerobotics.com). Kliknij na zakładkę „**download software**”. Wybierz pakiet **ERP MINI 2.0**.


2. Wybierz odpowiedni system operacyjny i kliknij na „**Download**”.


3. Zainstaluj program z pobranego pliku.

4. Po zakończeniu instalacji uruchom program.

Możesz także pobrać **oprogramowanie KEIRO** z Google Play lub Apple Store, aby korzystać z niego za pośrednictwem urządzenia typu smart.

## ERP MINI 2.0 Z BLUETOOTH

 [Windows \(7, 8.1, 10\) 32-Bit](#)

 [Windows \(7, 8.1, 10\) 64-Bit](#)

 [MacOS](#)

GET IT ON  
 Google play

Available on the  
 App Store

## Aktualizacja firmware kontrolera MINI 2.0

Oprogramowanie KEIRO powiadomi Cię o nowych aktualizacjach firmware kontrolera MINI 2.0 za pomocą wyskakującego powiadomienia zaraz po nawiązaniu połączenia. Naciśnij „YES”, aby pobrać najnowszą aktualizację firmware.

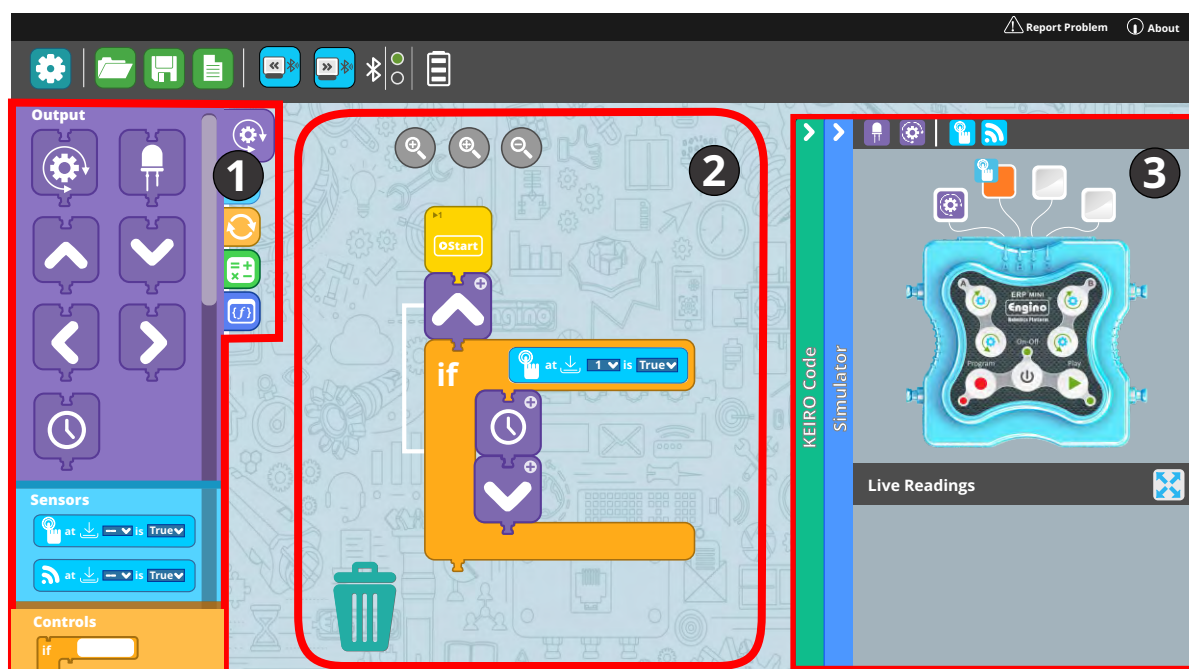


## Interfejs KEIRO

1. **Pasek bloków akcji** – tutaj znajdziesz wszystkie bloki akcji, których można użyć do utworzenia sekwencji poleceń. Aby ułatwić ich wybieranie, bloki są podzielone na kategorie.

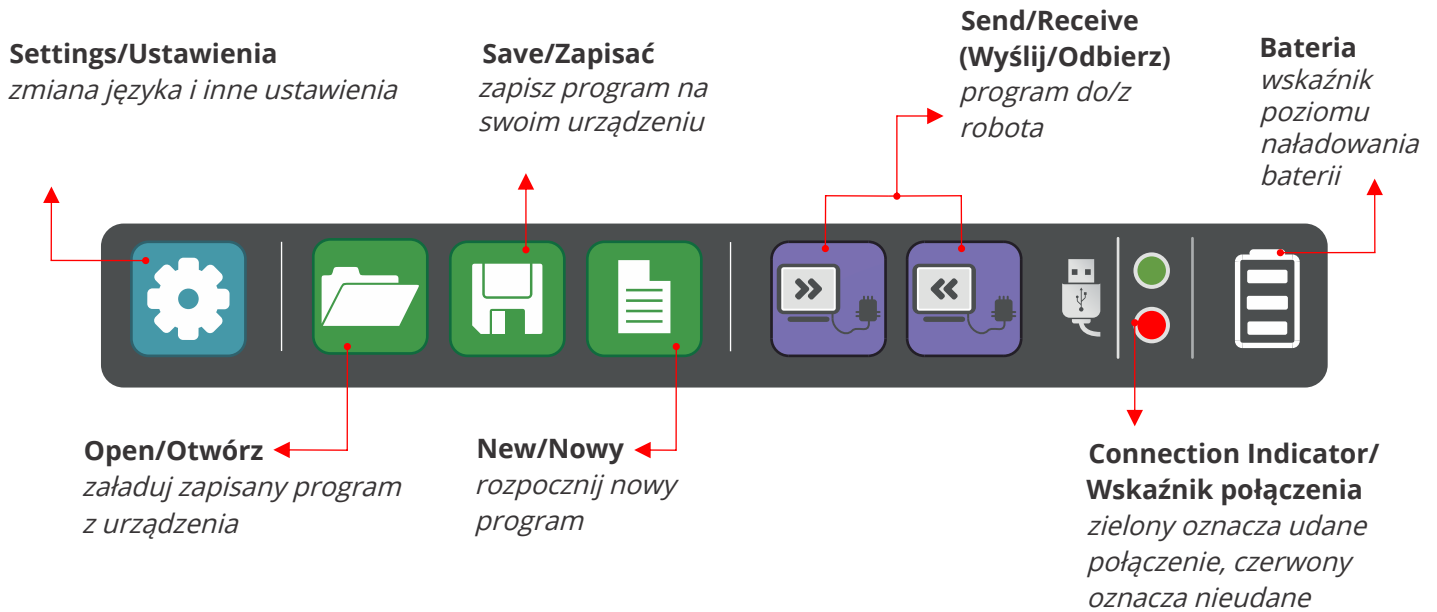
2. **Diagram przepływu** - obszar, w którym można przeciągać i upuszczać bloki oraz tworzyć program dla robota. Zmiany w programie dokonuje się poprzez modyfikacje właściwości poszczególnych bloków.

3. **Symulator/kod KEIRO** - aby oprogramowanie zostało poinformowane, do którego portu podłączone są urządzenia peryferyjne, ważne jest skonfigurowanie symulatora. Panel kodu pokazuje program w formie tekstowej zamiast bloków. (Kliknij na każdy pasek, aby ukryć lub pokazać okna).



## Pasek menu

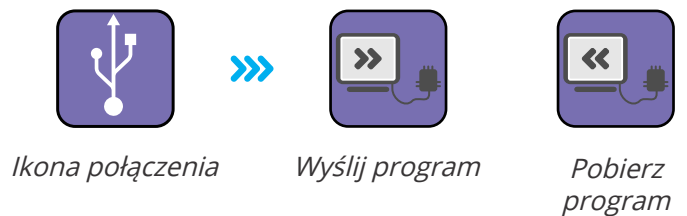
Pasek menu znajduje się w górnej części oprogramowania KEIRO, w którym można znaleźć różne przyciski do zadań takich jak inicjowanie połączenia z kontrolerem MINI 2.0, powrót do ekranu głównego, uruchamianie/zapisywanie programu itp.



## Podłączanie do komputera

Możesz podłączyć **kontroler MINI 2.0** do **komputera** za pomocą dołączonego kabla mini USB, a następnie klikając na odpowiednią ikonę w menu. Po nawiązaniu połączenia czerwony wskaźnik na pasku menu zmieni kolor na zielony, informując, że połączenie między komputerem a kontrolerem zostało nawiązane.

Po pomyślnym nawiązaniu połączenia na pasku menu pojawią się dwa nowe przyciski. Możesz „**Send/Wysłać**” program z komputera PC do kontrolera MINI 2.0 lub odwrotnie, klikając odpowiednie przyciski.

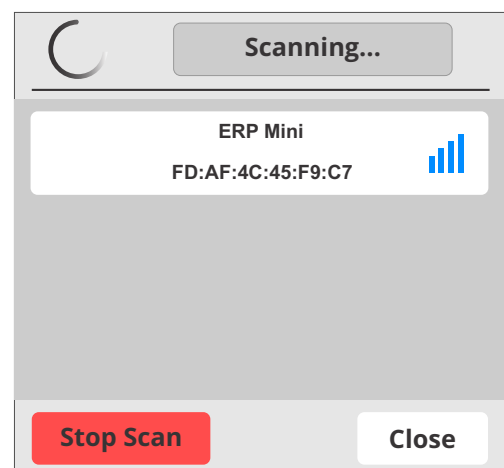


## Połącz się z urządzeniem inteligentnym

Korzystając z oprogramowania KEIRO za pośrednictwem urządzeń typu smart (platformy Android i iOS), na pasku menu pojawi się ikona bluetooth zamiast USB.



Aby podłączyć kontroler MINI 2.0, włącz bluetooth w urządzeniu i włącz kontroler MINI 2.0. Po kliknięciu ikony bluetooth oprogramowanie wyszukuje dostępne urządzenia. W nowym oknie, które się pojawi, wybierz kontroler MINI 2.0, aby połączyć go z tabletem.



Wyszukaj urządzenia bluetooth i podłącz je do ERP Mini



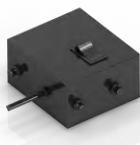
# Panel Symulatora

## Konfiguracja Symulatora

Symulator to specjalne okno w oprogramowaniu KEIRO, służące do konfiguracji urządzeń peryferyjnych robota. Ważne by przed użyciem diagramu przepływu najpierw **skonfigurować Symulator**, umieszczając urządzenia peryferyjne w tych samych portach, co podłączone do robota. Aby to zrobić, **przeciągnij** ikony urządzeń peryferyjnych do odpowiednich portów. Na górnym pasku symulatora znajdują się ikony odpowiadające każdemu z urządzeń peryferyjnych.



**Lampka LED**  
*Ikona bloku*



**Silnika prądu stałego**  
*Ikona bloku*



**Czujnik dotykowy**  
*Ikona bloku*

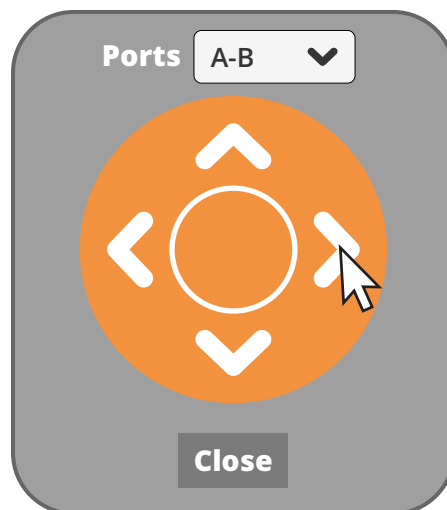


**Czujnik podczerwieni**  
*Ikona bloku*

## Panel GamePad



Panel GamePad pojawia się po kliknięciu specjalnej ikony znajdującej się w Symulatorze. Może być używany do jednoczesnego sterowania dwoma portami silnika. Pamiętaj, że silniki można podłączyć tylko do portów A-B, więc powinna to być wybrana kombinacja. GamePad to przydatny sposób na łatwe i proste poruszanie robotem. Wystarczy kliknąć (lub dotknąć w przypadku urządzenia inteligentnego) na strzałki, a silniki odpowiednio zareagują na polecenie.



*Panel GamePad*

Symulator może również służyć do zdalnego sterowania modelem. Ikony imitujące membranę kontrolera są w rzeczywistości aktywnymi przyciskami, na które można kliknąć. Dlatego możesz nagrywać lub odtwarzać program bezpośrednio z Symulatora. Zauważ, że podczas nagrywania programu bloki akcji są automatycznie generowane na schemacie blokowym.

## Live readings

Sekcja **Live readings** przekazuje po podłączeniu kontrolera do oprogramowania natychmiastową informację zwrotną z urządzeń peryferyjnych. Użytkownik może monitorować stan każdego podłączonego urządzenia, taki jak prędkość i kierunek obrotów silnika, czy dioda LED jest włączona, czy czujnik dotykowy jest wciśnięty, czy nie itd. Ponadto czujnik podczerwieni można skalibrować do określonej odległości wykrywania (więcej informacji na temat kalibracji czujnika podczerwieni można znaleźć na stronie 25).

The screenshot shows the 'Live Readings' interface with four panels:

- Motor:** Port: A, Obroty: ACL, Prędkość: 80
- LED:** Port: 2, Stan: ON
- Touch:** Port: 1, Stan: TRUE
- Infrared:** Port: 1, Stan: (checked), Zakres czułości: 65, Kalibrowanie: Calibrate

**Silnik:** pokazuje kierunek obrotów (zgodnie z ruchem wskazówek zegara/ przeciwnie do ruchu wskazówek zegara/ Anticlockwise - **ACL**) i poziom prędkości (**0-100**)

**LED:** wskazuje, czy jest włączony, czy nie. (**ON**, gdy świeci, w przeciwnym razie **OFF**)

**Czujnik dotyku:** wskazuje, czy jest wciśnięty, czy nie. (**TRUE** po naciśnięciu, w przeciwnym razie **FALSE**)

**Czujnik podczerwieni:** wskazuje, czy wykrywa obiekt, czy nie. (✓ w przypadku wykrycia, X gdy nie). (patrz strona 25)

## Kod KEIRO

Oprogramowanie KEIRO generuje pseudojęzyk zgodnie z diagramem przepływu danych. Jest to podstawowe narzędzie edukacyjne, które wprowadza uczniów w prawdziwy świat programowania, zapewniając stopniowe przejście od bloków wizualnych do zrozumiałego formatu tekstowego.

KEIRO Code (kod KEIRO) zawiera większość terminów powszechnie używanych w najpopularniejszych językach, takich jak BEGIN, IF, IF / ELSE, WHILE itp. W ten sposób uczniowie mogą bezpośrednio łączyć bloki akcji z poleceniami tekstowymi i odwrotnie!

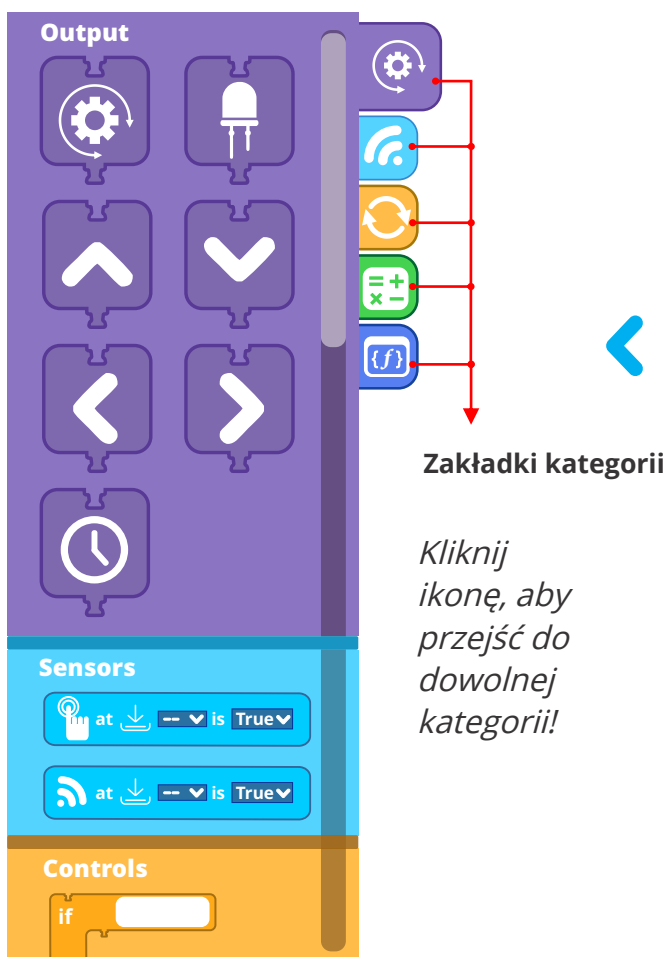
Kliknij na pasek, aby rozwinąć lub ukryć panel

```
BEGIN:
MOTOR:
  PORT: A
  STATE: ON FOREVER
  Direction: CLOCKWISE
  Speed: 100
  Delay (s): 0
  AFTER PREVIOUS
IF (TOUCH - Port: 1 == TRUE)
  LED:
    Port: B
    State: ON FOR DURATION
    Delay (s): 0
    Duration (s): 1.5
  AFTER PREVIOUS
END IF
BUZZER
  STATE: ON FOR DURATION
  Frequency (Hz): 200
  Delay (s): 0
  Duration (s): 2
  AFTER PREVIOUS
END
```

KEIRO Code



# Pasek bloków akcji



W tym menu znajdują się wszystkie bloki, których można użyć do utworzenia programu. Użytkownik po prostu wybiera żądany blok akcji, aby przeciągnąć go i upuścić na diagramie przepływu, aby utworzyć program.

Bloki ikon są podzielone na 3 kategorie:

## 1. Outputy (Wyjścia):

- Silnik
- Ruch (do przodu, do tyłu, w lewo, w prawo)
- LED
- Czekanie

## 2. Czujniki:

- Czujnik dotyku
- Czujnik podczerwieni (IR)

## 3. Sterowanie:

- IF, IF/ELSE
- WHILE
- REPEAT TIMES, REPEAT SECONDS
- REPEAT FOREVER,
- REPEAT UNTIL, WAIT UNTIL

## 4. Operatory

## 5. Funkcje

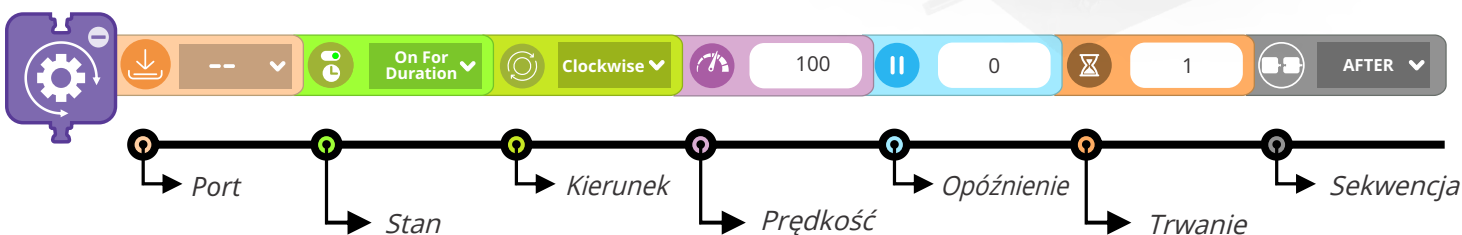
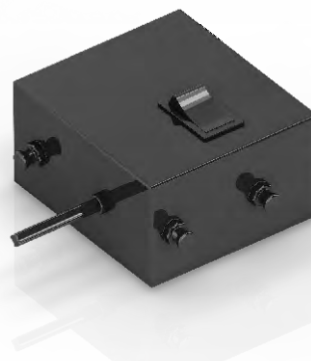
## 1. Outputy (Wyjścia)

Z tej kategorii możesz przeciągać i upuszczać dowolne dane wyjściowe (output) do diagramu przepływu i tworzyć sekwencję działań. Po umieszczeniu bloku w diagramie przepływu należy zauważyć, że otwiera się **pasek właściwości** w celu edycji bloku. Na przykład musisz przypisać port, w którym urządzenie wyjściowe jest podłączone do kontrolera. Możesz także dostosować blok akcji, zmieniając czas trwania i prędkość silnika.



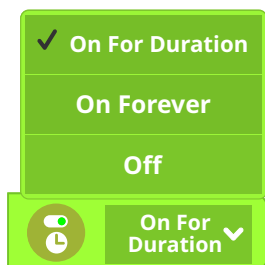
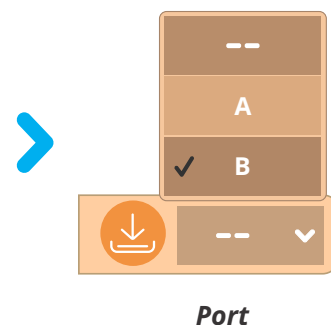
## Blok silnika

Poniżej możesz zobaczyć szczegółowy blok silnika, pokazany z jego domyślnymi **właścwościami** i wartościami.



Przypisz **Port**, w którym silnik jest podłączony do kontrolera. Domyślnie ta właściwość **jest nieprzypisana**, dlatego kliknij obszar pola, aby wybrać odpowiedni port. Dostępne porty zaznaczono ciemniejszym kolorem.

*Należy pamiętać, że Symulator powinien być skonfigurowany z co najmniej jednym silnikiem, aby było możliwe wybrać port.*



**Stan**

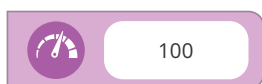
Skonfiguruj aplikację Engino kidCAD (3D Viewer): silnika. Do wyboru są trzy opcje: **On For Duration** (Włączony przez czas trwania – domyślnie): Silnik będzie działał przez okres czasu, który jest ustawiony we właściwości Czas trwania (patrz poniżej). **On Forever**: Silnik działa w nieskończoność lub do momentu otrzymania nowego polecenia. **Off**.: Zatrzymuje obroty silnika.

Wybierz **Kierunek**, w którym silnik ma się obracać. Wybierz pomiędzy obrotami **clockwise/zgodnymi z ruchem wskazówek zegara (domyślne) lub anticlockwise/przeciwnie do ruchu wskazówek zegara**. Pamiętaj, że przełącznik powinien być ustawiony w pozycji I, aby silnik działał poprawnie z wybranym kierunkiem.



**Kierunek**

*Uwaga: jeśli stan jest ustawiony na OFF, ta właściwość nie jest dostępna.*

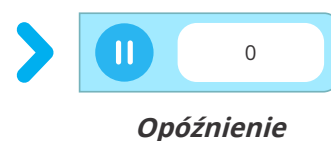


**Prędkość**

Ustaw **Prędkość** obrotową silnika. Wpisz w polu wartość **od 0 do 100**. Wartość reprezentuje procent energii dostarczonej do silnika, więc 100 (wartość domyślna) oznacza maksymalną moc.

*Uwaga: jeśli stan jest ustawiony na OFF, ta właściwość nie jest dostępna.*

Ustaw **Czas opóźnienia** (w sekundach), aby odroczyć uruchomienie bloku silnika. Jeśli wartość jest ustawiona na 0 sekund (**domyślnie**), nie jest ustawiony żaden czas opóźnienia.



**Opóźnienie**



**Trwanie**

Ustaw **Czas trwania** w sekundach, aby silnik obracał się przez określony czas. Silnik zatrzyma się po upływie tego czasu.

*Należy pamiętać, że ta właściwość jest włączona tylko wtedy, gdy stan jest ustawiony na „On For Duration”.*

**Domyślnie Sekwencje** bloków są wykonywane jedna po drugiej (AFTER). Aby wykonywać zadania jednocześnie, wybierz „WITH”, aby blok był wykonywany jednocześnie z poprzednim. Ta funkcja umożliwia uruchamianie kilku bloków jednocześnie.





## Bloki nawigacji (ruchu)

Aby wprowadzić pojazd w ruch, oba silniki powinny działać jednocześnie. Zamiast przeciągać dwa bloki na diagram przepływu, **bloki ruchu** pomogą wskazać, czy pojazd powinien poruszać się **do przodu, do tyłu, w lewo czy w prawo**. Jest to znacznie prostszy sposób nawigacji, który zmniejsza złożoność i liczbę używanych bloków na diagramie przepływu.



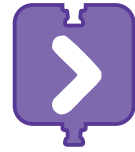
*Do przodu*



*Do tyłu*

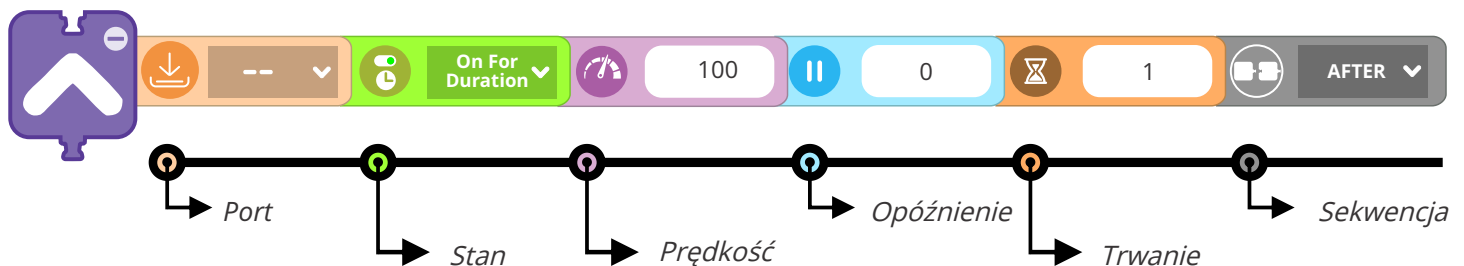


*W lewo*



*W prawo*

Poniżej znajduje się przykład rozwiniętego bloku ruchu do przodu z jego domyślnymi **właścwościami** i wartościami. Zauważ, że te cechy i funkcje są takie same dla wszystkich innych bloków nawigacji.



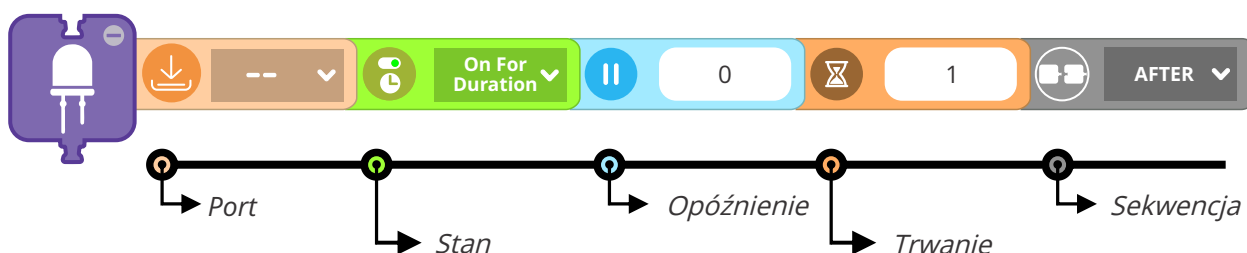
Bloki nawigacji mają te same właściwości co blok silnika. Pamiętaj, że oba przełączniki silnika muszą być ustawione w **pozycji I**, aby bloki nawigacyjne mogły poruszać robotem.

## Blok LED

Dioda elektroluminescencyjna (**L**ight **E**mitting **D**iode – **LED**) to źródło światła, które wykorzystuje energię elektryczną przechodzącą przez kryształ w celu wytworzenia światła. Ze względu na to, że diody LED są jednym z najbardziej wydajnych i najbezpieczniejszych źródeł światła o niskim zużyciu, znajdują szerokie zastosowanie w urządzeniach elektronicznych i robotyce.

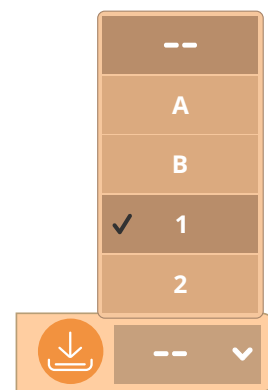


Poniżej możesz zobaczyć szczegółowy blok diod LED z jego domyślnymi **właścwościami** i wartościami.

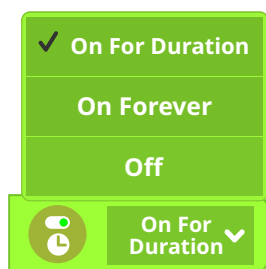


Przypisz **Port**, w którym silnik jest podłączony do kontrolera MINI 2.0. **Domyślnie** ta właściwość jest **nieprzypisana**, dlatego kliknij obszar pola, aby wybrać odpowiedni port. Dostępne porty zaznaczono ciemniejszym kolorem.

*Uwaga: Aby port działał, symulator powinien być skonfigurowany z co najmniej jedną diodą LED.*



**Port**



**Stan**

Ustaw **Stan** diody LED. Do wyboru są trzy opcje: **On For Duration (domyślna)**: lampka LED świeci przez określony czas, podany we właściwości Trwanie (patrz poniżej). **On Forever**: włącza diodę LED, dopóki nie otrzyma kolejnego polecenia. **Off**: wyłącza diodę LED.



Ustaw czas **Opóźnienia** w sekundach, aby opóźnić wykonanie bloku LED. Jeśli wartość jest ustawiona na 0 sekund (**domyślnie**), nie zostaje zastosowane żadne opóźnienie.



**Opóźnienie**



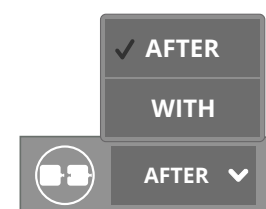
**Trwanie**

Określ czas **Trwania** w sekundach, przez jaki lampka LED powinna być włączona. Po upływie tego czasu dioda zgaśnie.



*Uwaga: ta właściwość jest włączona tylko wtedy, gdy stan jest ustawiony na „On For Duration”.*

Domyślnie **sekwencje** bloków są wykonywane jedna po drugiej (**AFTER**). Aby wykonywać zadania równocześnie, wybierz „**WITH**” (z poprzednim), aby wykonać blok razem z poprzednim. Ta funkcja umożliwia uruchamianie kilku bloków jednocześnie.

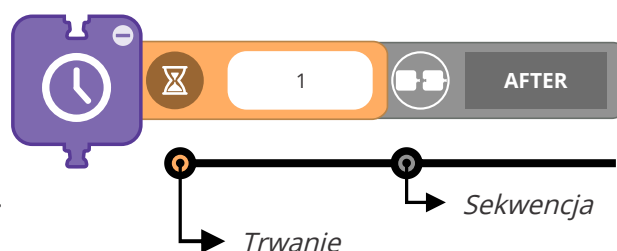


**Sekwencja**

## Blok czasu bezczynności

Blok czasu bezczynności wstrzymuje program na określony czas. Ten czas bezczynności jest ustawiany przez właściwość czasu **trwania**, która domyślnie jest ustawiona na 1 sekundę.

Ze względu na definicję czasu bezczynności, blok nie może być używany równolegle z poprzednim ani kolejnym zadaniem.





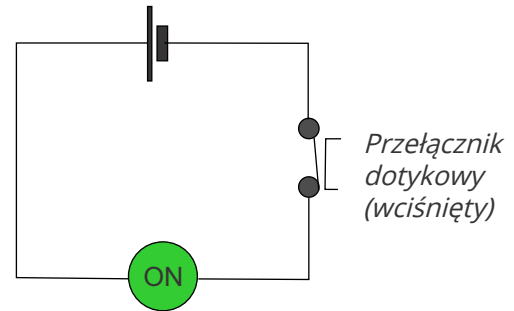
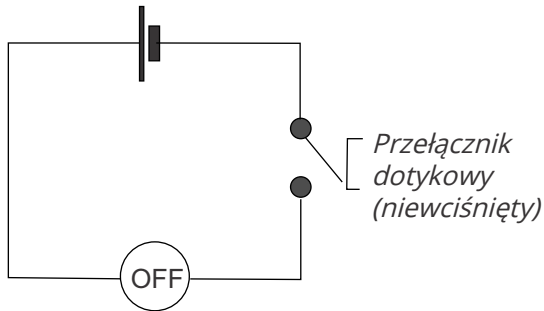
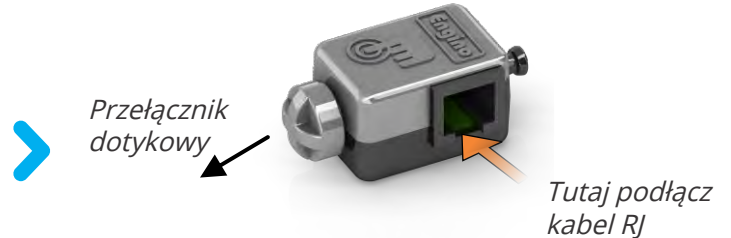
## 2. Blok czujników



Czujniki to wejścia (inputy) sterujące zachowaniem robota. W rzeczywistości są to elementy niezbędne do zaklasyfikowania maszyny jako zrobotyzowanej, a nie automatyzowanej. Robot bierze pod uwagę informacje uzyskane z otoczenia, aby móc wykonywać zadania zgodnie z zaistniałą sytuacją.

### Czujnik dotyku

Możesz fizycznie podłączyć czujnik dotykowy Engino za pomocą kabla RJ do dowolnego portu sterownika. Czujnik dotykowy Engino jest w zasadzie **przełącznikiem dotykowym**. Z czujnika wychodzą dwa rodzaje sygnałów (TRUE i FALSE), które zależą od tego, czy jest wciśnięty. Poniżej znajduje się schematyczny opis obwodów.



Po **naciśnięciu** przełącznika obwód zamyka się, powstaje przepływ prądu i czujnik jest w stanie **TRUE**:  
Gdy przełącznik nie jest wciśnięty, czujnik jest w stanie **FALSE**, ponieważ przez obwód nie płynie prąd.

Przypisz port, w którym do kontrolera podłączony jest czujnik dotyku.



Wybierz stan, który ma być sprawdzony, TRUE lub FALSE. Stan domyślny to TRUE.

Blok czujnika nie może być używany w diagramie przepływu jako niezależny blok akcji. W rzeczywistości blok czujnika jest wejściem o **wartości logicznej** (typ boolowski). Gdy stan czujnika zgadza się z blokiem, wejście (input) jest TRUE, a gdy nie, wtedy FALSE.

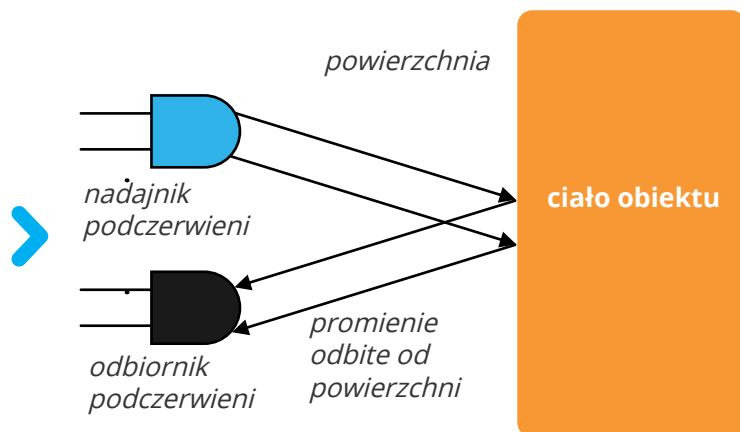
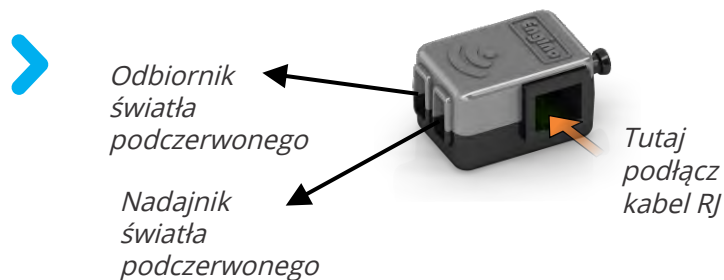
Dlatego blok czujnika musi być używany w połączeniu z instrukcją sterującą, taką jak IF, WHILE itp., aby poinformować robota, czy warunek jest spełniony (true) czy nie (false). Więcej informacji na temat wartości logicznej bloku czujnika można znaleźć w sekcjach **Bloki sterujące i Operatory**.

## Czujnik podczerwieni (IR)

Czujnik podczerwieni Engino można podłączyć do kontrolera za pomocą kabla RJ. Czujnik podczerwieni Engino to **aktywny** czujnik podczerwieni składający się z dwóch elementów: **nadajnika podczerwieni** (źródła) i **odbiornika podczerwieni** (detektora). Czujniki podczerwieni mogą być stosowane do wykrywania obiektów i odróżniania czarnych powierzchni od białych.

Nadajnik podczerwieni emituje sygnał w podczerwieni w obszarze bliskiej podczerwieni (700nm do 1400nm) widma elektromagnetycznego. Te długości fal są niewidoczne dla ludzkiego oka.

Gdy sygnał dociera do obiektu, jego część odbije się z powrotem. Odbiornik podczerwieni to fototranzystor zdolny do wykrywania promieniowania podczerwonego. W związku z tym czujnik jest w stanie **TRUE**, gdy odbiera dużo promieniowania z obiektu, oraz w stanie **FALSE**, gdy dociera do niego niewielka ilość promieniowania lub nie ma go wcale.

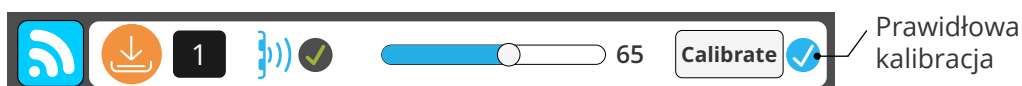


## Kalibracja czujnika podczerwieni

Czujnik podczerwieni należy skalibrować w optymalnym zakresie wyzwalania, aby robot „wiedział”, kiedy zachowywać się w zależności od sytuacji TRUE lub FALSE. Kalibrację można rozpocząć z poziomu panelu **Live readings**.

Stan **TRUE** aktywuje czujnik podczerwieni dla **koloru białego** i dla odległości **równych lub mniejszych niż odległość wyzwalania**.

Stan **FALSE** aktywuje czujnik podczerwieni dla koloru czarnego i dla odległości **większych niż odległość wyzwalania**.

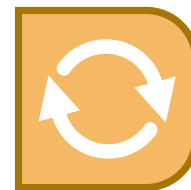


**Rozpoznawanie obiektów:** Umieść czujnik podczerwieni w tej samej odległości, z której będzie działał czujnik. Przytrzymaj model na miejscu i kliknij przycisk Kalibruj (Calibrate). Kalibracja jest zakończona, gdy obok przycisku Calibrat pojawi się znak „√”. **W celu śledzenia linii:** Umieść czujnik podczerwieni nad białą powierzchnią, w tej samej odległości operacyjnej, w której będzie działał czujnik. Przeprowadź kalibrację w taki sam sposób, jak opisano powyżej.

\* Kalibrację można również przeprowadzić ręcznie, przeciągając pasek dożądanego poziomu wyzwalania.



### 3. Bloki sterujące

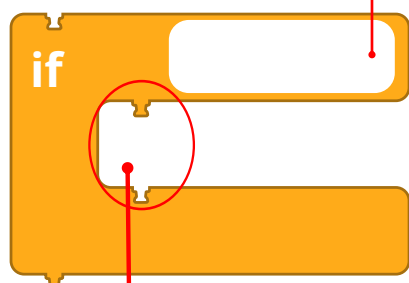


W tej kategorii można znaleźć bloki, które tworzą polecenia sterujące. Podstawowe instrukcje warunkowe, takie jak **IF**, **WHILE**, **REPEAT** i **UNTIL**, pomagają w bardziej wyrafinowanych i praktycznych zastosowaniach. Dane wejściowe (input) z czujników służą do wykonywania bloków, które są umieszczane w instrukcji warunkowej.

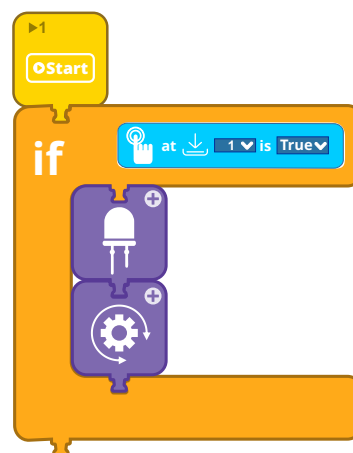
#### **IF:**

Blok IF jest aktywowany, gdy warunek czujnika jest spełniony (True lub False). Na przykład, jeśli zostanie naciśnięty czujnik dotykowy, zaświeci się dioda LED, a następnie silnik zacznie się obracać. Należy zwrócić uwagę, że jeśli czujnik zostanie aktywowany po raz drugi przed wykonaniem sekwencji bloków znajdujących się w środku, sekwencja nie rozpocznie się ponownie.

*Przeciągnij tutaj bloku ikon czujnika, którego stan ma zostać sprawdzony*

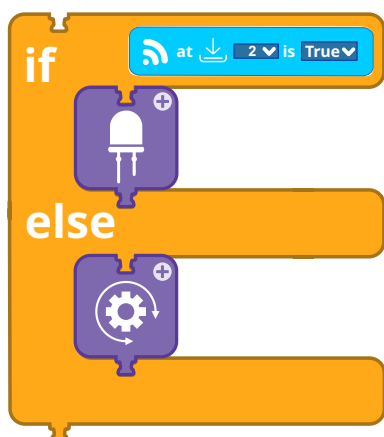


*Przeciągnij bloki akcji, aby utworzyć sekwencję, która powinna zostać wykonana, gdy warunek zostanie spełniony*



#### **IF/ELSE :**

Blok IF/ELSE składa się z dwóch części. Sekwencja w pierwszej części uruchomi się, gdy warunek czujnika zostanie spełniony (na przykład True, gdy zostanie naciśnięty czujnik dotyku), tak samo jak blok IF. Gdy warunek jest odwrotny (np. False, czujnik dotyku nie jest wciśnięty) robot wykona sekwencję bloków umieszczonych wewnątrz drugiej części bloku.

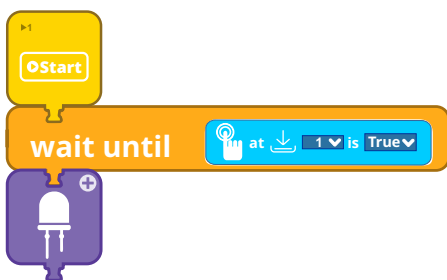
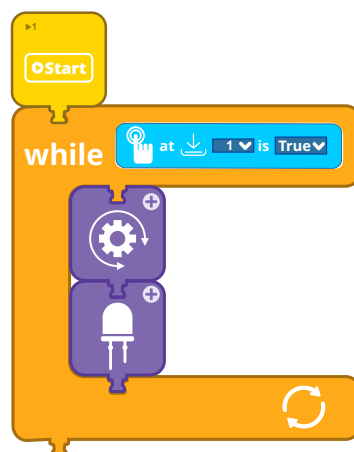


*W tym przykładzie dioda włączy się, jeśli czujnik podczerwieni stanie się TRUE (tj. wykryje obiekt). W przeciwnym razie silnik będzie się obracał.*

*Zauważ, że bloki wewnątrz ELSE zostaną przerwane, gdy warunek zostanie spełniony, niezależnie od ich stanu / czasu trwania, a program wykona sekwencję wewnątrz bloku IF.*

## While:

Blok WHILE będzie aktywny tak długo, jak długo warunek czujnika zostanie spełniony (True lub False), tym samym odtwarzając podprogram w pętli. Po przerwaniu warunku sekwencja wewnątrz bloku WHILE zostanie natychmiast zatrzymana i program przejdzie do następnego bloku akcji. Na przykład, *gdy czujnik dotykowy pozostaje wciśnięty, silnik obraca się zgodnie z ruchem wskazówek zegara i dioda LED będzie się świecić (do momentu zwolnienia czujnika).*

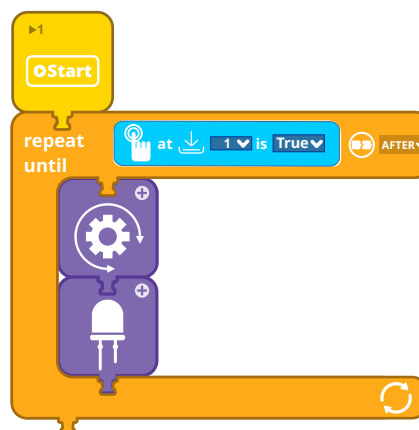


## Wait Until:

Ten blok ustawia program w tryb bezczynności, dopóki nie zostanie spełniony określony warunek. Na przykład program nie włączy diody LED, dopóki nie naciśniesz czujnika dotykowego.

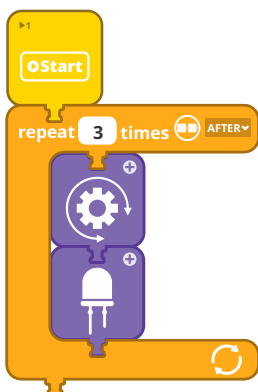
## Repeat Until:

Blok Repeat until (powtarzaj aż) odtwarza w pętli sekwencję wewnątrz bloku i zatrzymuje się tylko wtedy, gdy warunek zostanie spełniony. Pętla zostanie przerwana, gdy warunek zostanie spełniony, a program przechodzi do następnego bloku akcji.

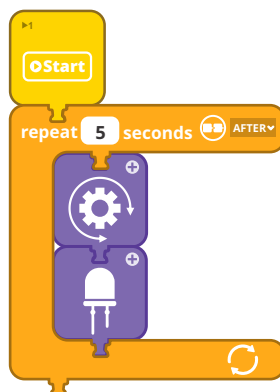


## Repeat:

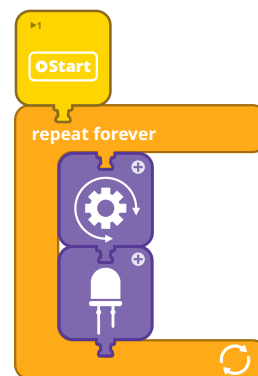
Istnieją trzy typy powtarzających się bloków, które nie zależą od czujników lub zmieniających się wejść, ale od stałych scenariuszy. Blok „**repeat times**” pozwala ustawić liczbę wymaganych powtórzeń, natomiast „**repeat seconds**” będzie powtarzać sekwencję przez pewien czas. Ostatnią opcją jest blok „**repeat forever**”, który jest przeznaczony do odtwarzania sekwencji w nieskończonej pętli. Zauważ, że ten blok nie ma opcji połączenia z innym blokiem, ponieważ nie można już żadnego dodać.



Pętle trzykrotnie



Pętla przez pięć sekund



Nieskończona pętla

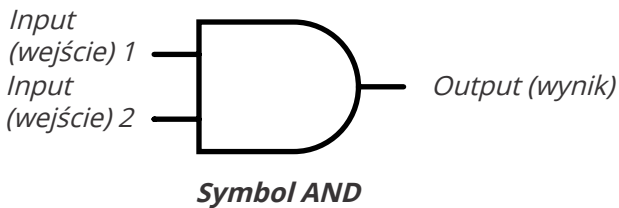


## 4. Bloki operatorów

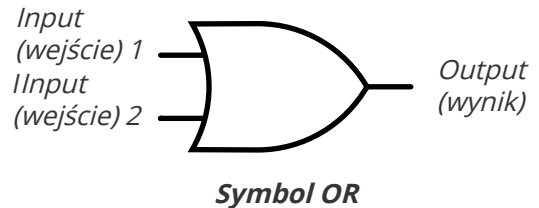


W tej kategorii znajdziesz operatory logiczne, które są przydatne w sytuacjach, w których powinna nastąpić akcja, gdy wystąpi kombinacja warunków. Ogólnie rzecz biorąc, operatory mówią komputerom, jak postępować z informacjami.

**Operatory logiczne** są prawdopodobnie najważniejszymi operatorami i są szeroko stosowane w każdym języku programowania. Mogą łączyć dwa sygnały wejściowe (inputy), aby podjąć decyzję. Załóżmy, że chcemy połączyć wyniki dwóch czujników, aby wydać polecenie robotowi. Następnie operatory logiczne **AND** i **OR** pomagają nam w uzyskaniu ostatecznego wyniku.



**Operator AND:** Gdy oba wejścia mają wartość „true”, to wynik ma również wartość „true”. W przeciwnym razie wynik ma wartość „false”.

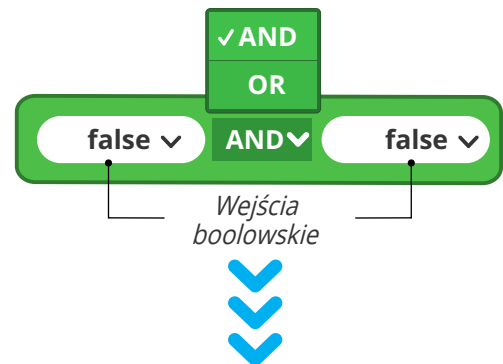


**Operator OR:** Gdy jedno lub oba wejścia są „true”, wówczas wynik ma wartość „true”. Wynik będzie miał wartość „false” tylko wtedy, gdy oba wejścia będą miały wartość „false”.

### Blok operatora logicznego

Operator logiczny oprogramowania KEIRO ma dwa gniazda wejściowe, które przyjmują wartości logiczne, TRUE lub FALSE (domyślnie). Kliknięcie w środkowe pole pozwala użytkownikowi wybrać operator AND/OR.

Pamiętaj, że blok czujnika jest praktycznie wartością logiczną, więc przeciągając go do tych slotów powstanie wartość. Stąd, gdy do obu gniazd jest przypisany blok czujnika, operator logiczny zwraca dane wyjściowe w oparciu o te dwa wejścia.



Założmy, że mamy zrobotyzowany pojazd z dwoma czujnikami podczerwieni służącymi do wykrywania przeszkód. Gdy nie zostaną wykryte żadne przeszkody, chcemy poinstruować robota, aby ruszył do przodu. Zatem oba czujniki muszą być w stanie False. Blok do przodu będzie powtarzany, dopóki oba warunki nie zostaną spełnione.

## 5. Funkcje

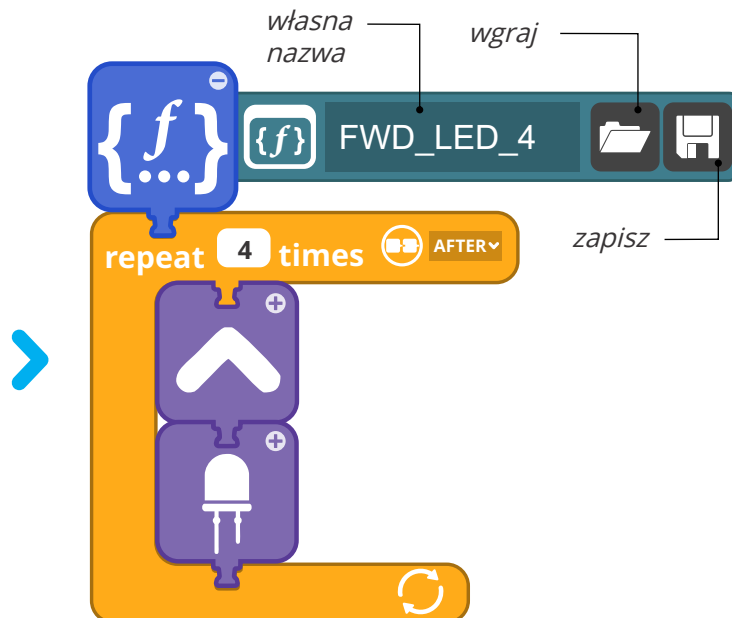
Funkcja to podprogram zawierający określoną sekwencję poleceń. Jest to po prostu krótki program, który może używać lub uruchamiać program główny w trybie sekwencyjnym lub równoległym. Funkcje są przydatne w zmniejszaniu liczby bloków pojawiających się na diagramie przepływu, szczególnie w przypadkach, gdy długa sekwencja poleceń musi być wykonana kilka razy.



### Ustawienia funkcji

W celu stworzenia nowej funkcji przeciągnij blok „**set function**” i upuść go na diagramie przepływu. Zauważ, że ten blok nie może być połączony z innymi blokami, ponieważ jest to początkowy blok nowej sekwencji poleceń.

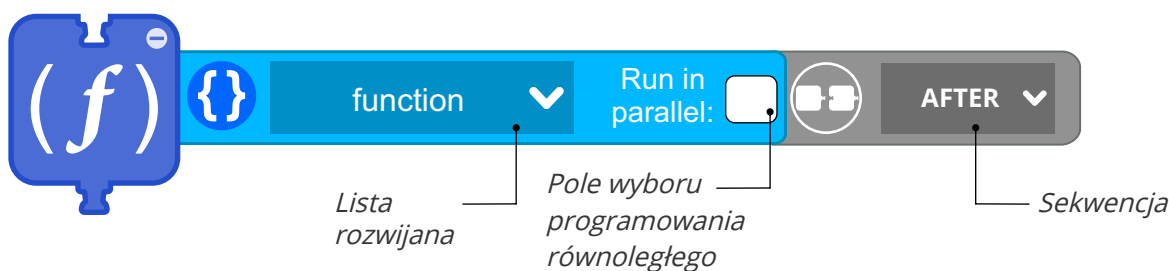
Można utworzyć podprogram pod blokiem „set function” (ustawianie funkcji), używając dowolnego bloku akcji (outputy, czujniki, elementy sterujące itp.). Możesz również nadać mu **odpowiednią nazwę**, aby ułatwić identyfikację, szczególnie podczas ustawiania wielu funkcji. Funkcję można zapisać na komputerze/ tablecie i zaimportować w dowolnym momencie za pomocą ikon **zapisz** i **wgraj**.



Funkcja zawiera podprogram z poleceniami

### Korzystanie z funkcji

Aby wywołać funkcję, użytkownik musi umieścić blok „**Use a function**” (użyj funkcji) w sekwencji diagramu przepływu. W przypadkach, gdy ustawionych jest wiele funkcji, użytkownik może wybrać funkcję za pomocą listy rozwijanej.



#### Tryb sekwencyjny:

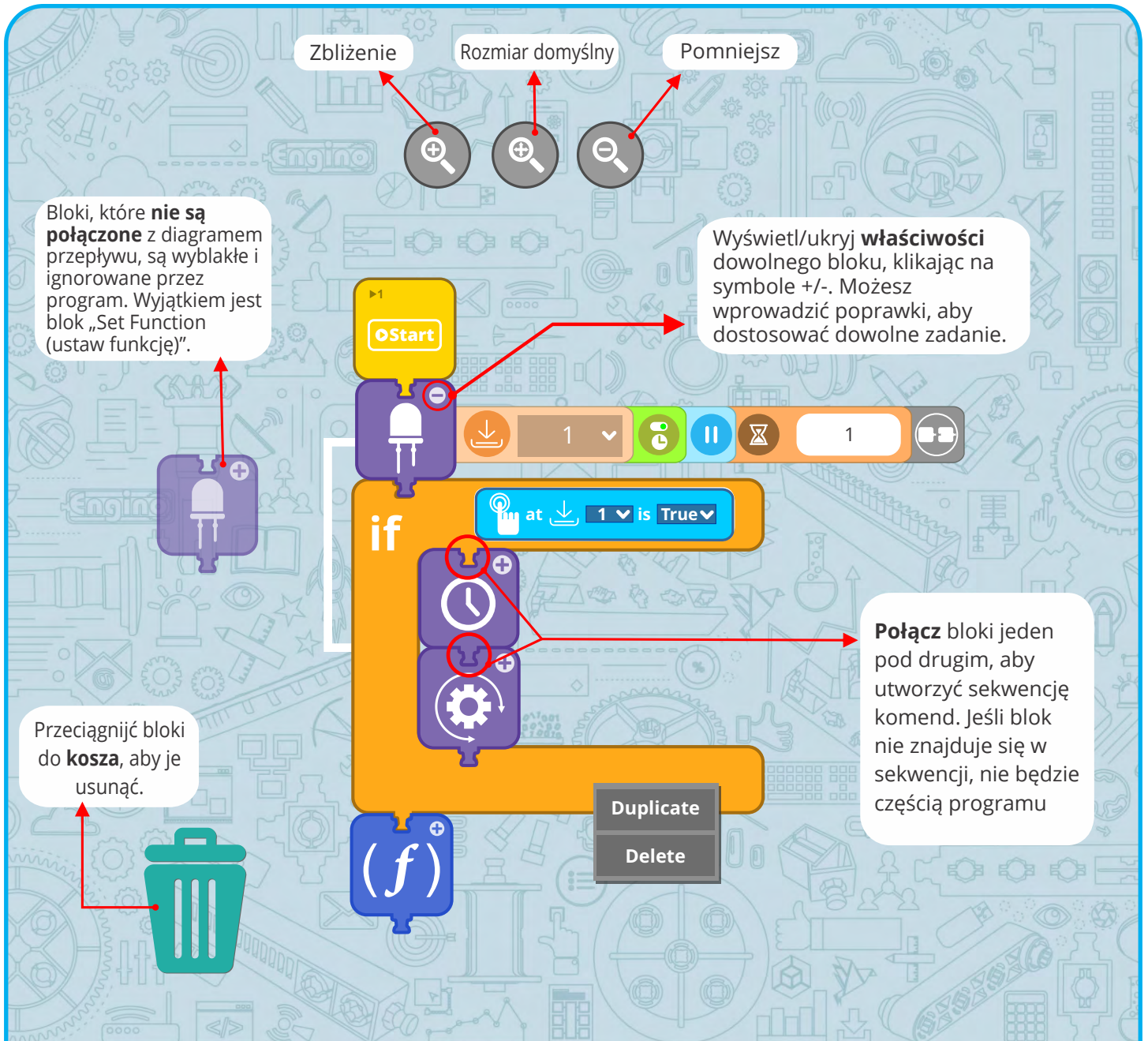
**Domyślnym** trybem uruchamiania funkcji jest **tryb sekwencyjny**. Oznacza to, że gdy program dotrze do bloku „use function” (użyj funkcji), wykona wszystkie zadania umieszczone wewnątrz funkcji. Po wykonaniu wszystkich zadań w funkcji przechodzi do następnego bloku diagramu przepływu.

#### Tryb „Run in parallel” (równoległe):

Aby uruchomić tryb równoległy, zaznacz pole wyboru. W tym trybie program uruchamia zadania w funkcji i natychmiast rozpoczyna następny blok akcji w diagramie przepływu. W ten sposób podprogramy wewnątrz funkcji będą działały równoległe z resztą diagramu przepływu.

# Programowanie diagramu przepływu

Poniżej możesz zobaczyć przykład diagramu przepływu wraz z opisami jego głównych funkcji. Aby utworzyć program, przeciągnij bloki z Paska bloków (lewa strona ekranu) pod ikonę „Start” w oknie głównym. Zauważ, że gdy przeciągasz blok akcji w pobliżu wolnego punktu połączenia, obszar ciemnieje. Kliknij symbol „+” na blokach, aby dostosować właściwości każdego bloku akcji, takie jak port, czas trwania itp. Po zakończeniu programu kliknij przycisk „Send program” (wyślij program) na pasku menu, a program zostanie przesłany do kontrolera MINI.



Możesz użyć myszy, aby uzyskać więcej informacji i wprowadzić zmiany, wykonując następujące czynności:

- **Umieść** kursor myszy nad ikonami i blokami, aby wyświetlić opisy i wskazówki.
- Kliknij znak „+” obok bloków diagramu przepływu, aby wyświetlić i zmodyfikować ich właściwości.
- Kliknij znak „-”, aby ukryć właściwości bloku.
- Możesz kliknąć na blok **prawym przyciskiem myszy**, aby go powielić lub szybko usunąć jeden lub więcej bloków.

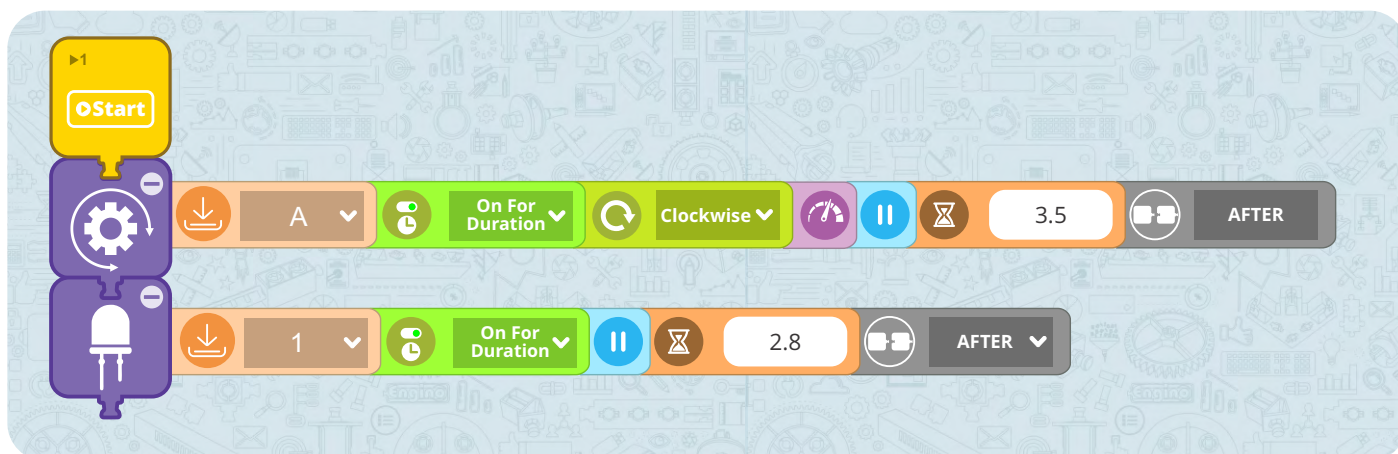


## Przykłady kodu

### Przykład 1 - Proste programowanie z silnikiem i diodą LED

W tym przykładzie użyto jeden **silnik** i jedną **diodę LED**, podłączone odpowiednio do portów A i 1. W programie silnik najpierw obraca się zgodnie z ruchem wskazówek zegara przez 3,5 sekundy, a następnie dioda LED włączy się na 2,8 sekundy. Program zostanie wykonany tylko raz.

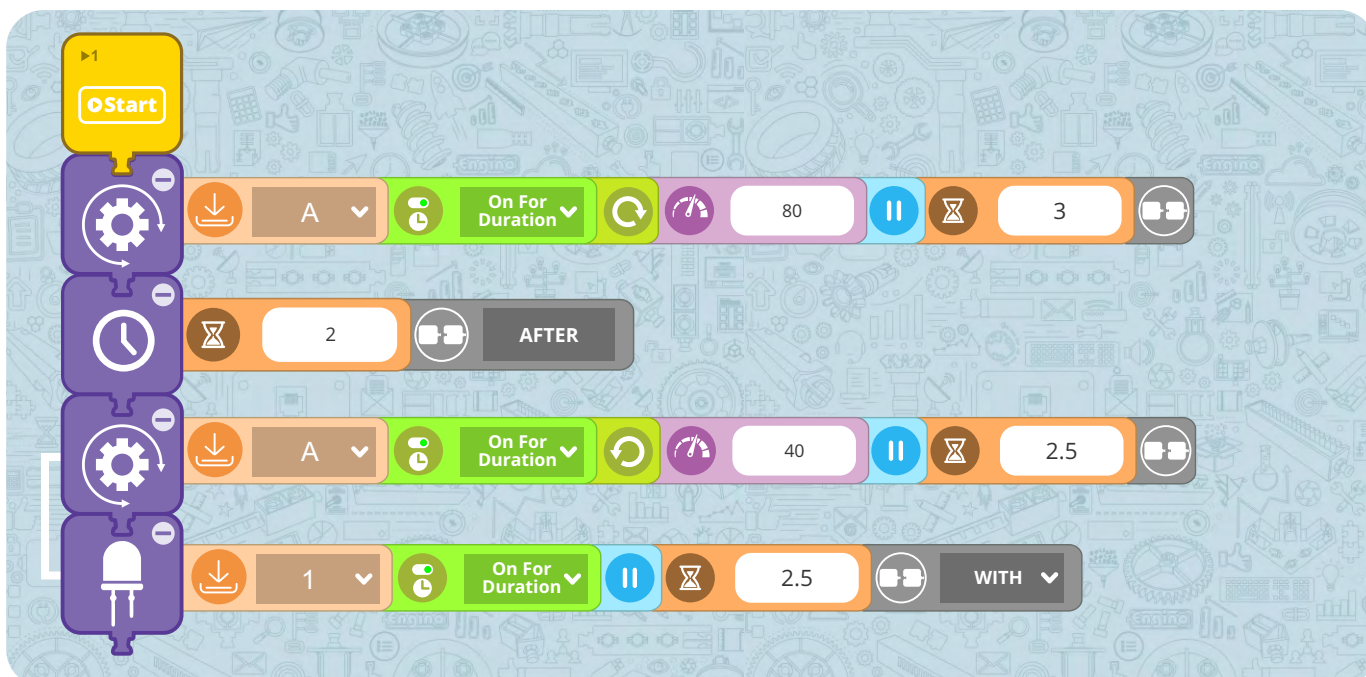
- Użyj stanu **On For Duration**
- Zmień czas trwania
- Wypełnij przedział czasu liczbą rzeczywistą (float)



### Przykład 2 - Stosowanie zadań synchronicznych i Bloku czasu bezczynności

W tym przykładzie zastosowano jeden **silnik** i jedną **diodę LED**, podłączone odpowiednio do portów A i 1. W programie silnik najpierw będzie się obracał przez 3 sekundy z prędkością ustawioną na 80%. Następnie program przejdzie w stan bezczynności przez 2 sekundy. Następnie silnik będzie się obracał z 40% prędkością z włączoną diodą przez 2,5 sekundy.

- Zmień prędkość silnika
- Korzystanie z bloku czasu bezczynności
- Programowanie zadań synchronicznych



### Przykład 3 - Odtwarzanie programu w pętli

Poniższy przykład używa jednego **silnika** i jednej **diody LED** podłączonych odpowiednio do portów B i 2. Zadanie polega na tym, aby silnik obracał się w kierunku przeciwnym do ruchu wskazówek zegara przez 1,5 sekundy, następnie zapalił diodę na 1 sekundę i tę sekwencję powinien powtarzać w nieskończoność.

- Zmień kierunek obrotów
- Program w nieskończonej pętli



### Przykład 4 - użycie instrukcji warunkowej IF

W tym przykładzie zastosowano jeden **silnik**, jedną **diodę LED** i **czujnik** dotyku, które są podłączone odpowiednio do portów B, 1 i 2. Do momentu **naciśnięcia** czujnika dotykowego (stan **FALSE**) silnik będzie się obracał w kierunku przeciwnym do ruchu wskazówek zegara. Po naciśnięciu czujnika dotykowego (stan to **TRUE**), dioda LED zaświeci się na 1,5 sekundy, a silnik będzie się obracał przeciwnie do ruchu wskazówek zegara przez 1 sekundę, ale z opóźnieniem 0,5 sekundy.

- Korzystanie ze stanu **On Forever**
- Korzystanie z czujnika dotykowego
- Instrukcja warunkowa **IF**
- Wykorzystanie z czasu opóźnienia





- Co to są dane wejściowe i wyjściowe.
- Jak ulepszyć konstrukcję robota.
- Jak ręcznie zaprogramować robota.
- Czym jest diagram przepływu danych.

## Budowa i programowanie

Robotyka obejmuje dwa główne aspekty: część konstrukcyjną i część programistyczną. Model wentylatora pozwoli ci eksperymentować z oboma. Dowiedz się, jak zmodyfikować konstrukcję, aby była bardziej wydajna. Następnie zaprogramuj model ręcznie przy pomocy kontrolera TM i użyj bloków wizualnych wewnątrz oprogramowania KEIRO.

Poziom trudności ★★★★★

### Potrzebne materiały:

- Creative Engineering 100 w 1 Robotized Maker Pro (CE101MP-A).
- PC lub urządzenie typu smart z pobranym i zainstalowanym oprogramowaniem KEIRO.

### Procedura:

**1.** Znajdź instrukcje na **stronach 43-45** i zbuduj model **wentylatora**. Upewnij się, że każdy kabel jest podłączony do właściwego portu. Owiń kable wokół innych części, aby nie przeszkadzały. Z tyłu kontrolera zainstaluj 3 baterie AAA i włącz urządzenie przyciskiem On-Off.

**2.** Przełącznik obu silników powinien być ustawiony w położeniu **I**. Należy pamiętać, że w położeniu **O** silnik jest wyłączony, a w położeniu **II** kierunek jest przeciwny. Naciśnij przyciski silnika (polecenie) na kontrolerze, jak pokazano w **ćwiczeniu 1**, i zapisz obserwowane wyniki (output).

**3.** Ulepsz swój model, postępując zgodnie z instrukcjami w **ćwiczeniu 2**.

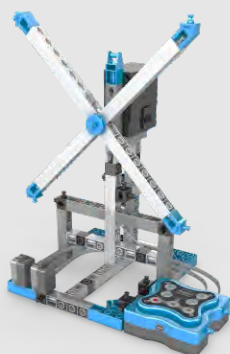
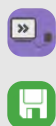
**4.** Wykonaj **ćwiczenie 3**, aby utworzyć program ręczny na kontrolerze. Możesz potrzebować kilku prób, zanim się uda.

**5.** Podłącz kontroler do komputera lub urządzenia typu smart. Uruchom oprogramowanie KEIRO i kliknij na przycisk połączenia.

**6.** Kliknij na przycisk „**Receive Program**” (pobierz program), aby załadować nagraną sekwencję do oprogramowania KEIRO, tworząc w ten sposób „**diagram przepływu**”.

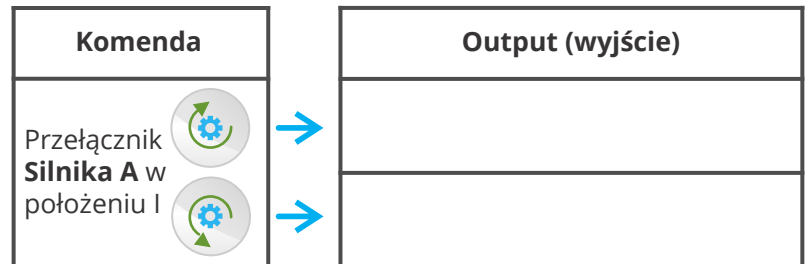
**7.** Zastosuj zmiany na diagramie przepływu na podstawie **ćwiczenia 4**.

Wyślij program do kontrolera klikając na przycisk „**Send Program**”. Zapisz program jako „**wentylator**”.

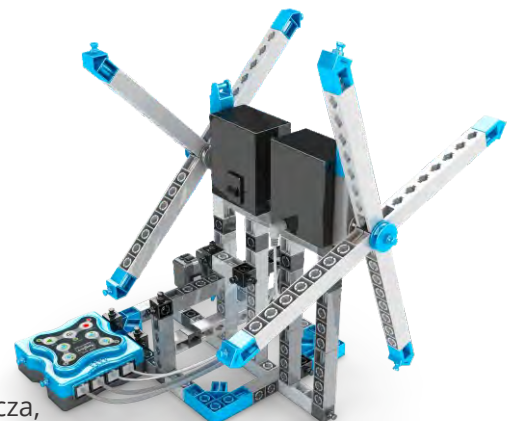


Model wentylatora Engino

**Ćwiczenie 1.** Naciśnij przyciski na kontrolerze (input) i z grubsza zapisz wynik, który obserwujesz na modelu (output).



**Ćwiczenie 2.** Stwórz drugi wentylator po przeciwnej stronie wentylatora i podłącz drugi silnik do portu B. Po prawej stronie możesz zobaczyć przykład konstrukcji.



**Ćwiczenie 3.** Wciśnij „Program” (czerwona migająca lampka oznacza, że program jest nagrywany), a następnie niezbędne przyciski, aby utworzyć następującą sekwencję: wentylator A obraca się zgodnie z ruchem wskazówek zegara; wentylator B obraca się zgodnie z ruchem wskazówek zegara; oba wentylatory obracają się jednocześnie w przeciwnych kierunkach;

Naciśnij ponownie przycisk „**Program**”, aby zapisać program, a następnie naciśnij przycisk „**Play**”, aby sprawdzić, czy sekwencja komend jest prawidłowa. Jeśli naciśniesz i przytrzymasz przycisk Play przez dłuższy czas, program się zapętli.

**Ćwiczenie 4.** Na diagramie przepływu danych zmodyfikuj właściwości każdego bloku akcji, aby utworzyć program opisany poniżej. Możesz dodać więcej akcji (silniki, czas postoju) z lewego menu: wentylator A obraca się zgodnie z ruchem wskazówek zegara przez 2 sekundy z prędkością 100; wentylator B obraca się zgodnie z ruchem wskazówek zegara przez 2 sekundy z prędkością 100; robot nic nie robi przez 1 sekundę (użyj bloku czasu bezczynności); oba wentylatory obracają się w tym samym czasie w tych samych kierunkach przez 3 sekundy, A z prędkością 100 i B z prędkością 40.



# Nauka o: ROBOTYCE

## Ruch robota i komenda (instrukcja) IF

Ruch robota jest bardzo ważny i czasami trudny do osiągnięcia. Istnieje wiele sposobów poruszania się robotów w zależności od ich przeznaczenia: niektóre mają koła lub gąsienice, inne używają nóg, a inne latają za pomocą śmigieł. Eksperymentuj z ruchem nóg za pomocą modelu hexapod!

## Dowiedz się:

- Jak robot porusza się przy pomocy nóg.
- Jak korzystać z Symulatora i Gamepada.
- Jak używać czujników podczerwieni z komendą warunkową IF.

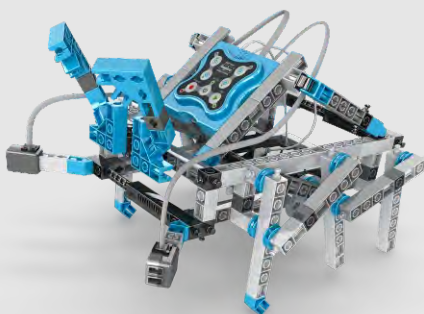
Poziom trudności ★★★★★

### Potrzebne materiały:

- Creative Engineering 100 w 1 Robotized Maker Pro (CE101MP-A).
- PC lub urządzenie typu smart z pobranym i zainstalowanym oprogramowaniem KEIRO.

### Procedura:

1. Wyszukaj instrukcje na **stronach 46-51** i zbuduj model **hexapod**.
2. Podłącz kontroler do komputera lub urządzenia typu smart. Uruchom oprogramowanie KEIRO i kliknij na przycisk połączenia. Wykonaj **ćwiczenia 1 i 2**, dzięki którym nauczysz się pracy z **Symulatorem** w programie KEIRO.
3. Otwórz okno **Gamepad**, klikając na odpowiedni przycisk i używaj go do sterowania robotem.
4. Zwróć uwagę, że aby uzyskać prosty ruch do przodu lub do tyłu, nogi muszą zaczynać z tej samej pozycji. W przypadku zastosowania kół nie ma takiego problemu, a ruch prosty zależy tylko od prędkości obrotowej silników. Czy przychodzą Ci do głowy jakieś inne problemy związane z używaniem nóg (powiązania)? Zapisz je w **ćwiczeniu 3**.
5. Wykonaj ćwiczenie 4, w którym zaprogramujesz hexapoda tak, aby omijał przeszkody po obu stronach. W tym celu musisz użyć instrukcji warunkowej IF.



Model Engino hexapod

**Ćwiczenie 1.** Przeciągnij urządzenia peryferyjne (silniki i czujniki) do portów Symulatora zgodnie z modelem hexapoda. Zaprojektuj również bloki tutaj na obrazku.



**Ćwiczenie 2.** Steruj heksapodem za pomocą Symulatora w następującej sekwencji: *hexapod skręca w prawo przy pomocy jednej nogi; hexapod obraca się w lewo przy pomocy jednej nogi; hexapod skręca w prawo używając obu nóg.*

**Ćwiczenie 3.** Jakie problemy mogą wystąpić podczas używania nóg zamiast kół?

.....

.....

.....

.....

**Ćwiczenie 4.** Stwórz następujący program, dzięki któremu hexapod będzie omijał przeszkody po obu stronach w niekończącej się pętli: hexapod porusza się do przodu w sposób ciągły; **jeśli** lewy czujnik podczerwieni wykryje przeszkodę, prawa noga powinna się cofnąć a lewa przesunąć do przodu; **jeśli** prawy czujnik podczerwieni wykryje przeszkodę, lewa noga powinna się cofnąć a prawa przesunąć do przodu.

Umieść odpowiednie bloki wewnątrz bloku IF. Sprawdź na **stronie 25**, jak skonfigurować czujniki podczerwieni do wykrywania obiektów.

- Jak używać czujników podczerwieni z instrukcjami warunkowymi WHILE i IF.
- Co to są bramki logiczne.

## Instrukcja WHILE i bramki logiczne

Czujniki podczerwieni są bardzo często używane w robotyce w dwóch bardzo ważnych zastosowaniach: do rozpoznawania czerni i bieli oraz do wykrywania obiektów. Wypróbuj oba wyzwania programistyczne i dowiedz się, jak skonfigurować czujniki podczerwieni, jak korzystać z instrukcji warunkowej WHILE i bramek logicznych.

Poziom trudności ★★★★★

### Potrzebne materiały:

- Creative Engineering 100 w 1 Robotized Maker Pro (CE101MP-A).
- PC lub urządzenie typu smart z pobranym i zainstalowanym oprogramowaniem KEIRO.
- 4 grubsze arkusze papieru w formacie A3, białe.
- Czarna taśma klejąca lub czarny marker.
- 4 białe objekty do wykrywania.

### Procedura:

**1. Znajdź instrukcje na stronach 38-42 i zbuduj model robota.** Upewnij się, że łączysz każde koło z portem na tej samej stronie.

**2.** Sklej papiery A3 razem tak, aby utworzyły duży biały prostokąt. Narysuj czarną elipsę (grubość linii 1,5 cm) za pomocą markera lub taśmy klejącej (rysunek po prawej). Możesz pobrać inny kształt pod adresem [www.enginorobotics.com/teaching\\_resources/examples/](http://www.enginorobotics.com/teaching_resources/examples/)

**3.** Uruchom program KEIRO i postępuj zgodnie z instrukcjami z **ćwiczenia 1**, aby stworzyć **program do śledzenia bieli**. W tym celu oba czujniki podczerwieni powinny być skierowane w stronę ziemi, więc ustaw je w dół. Robot będzie poruszał się po bieli i zatrzyma każde koło na czerni, podążając w ten sposób po linii.

**4.** Przeczytaj **ćwiczenie 2**, aby przekształcić robota w takiego, który **wykrywa objekty**. W tym celu ustaw dwa czujniki podczerwieni tak, aby były **skierowane do przodu**. Po utworzeniu programu przetestuj go, umieszczając cztery białe objekty (wysokie tak jak czujniki), aby utworzyć wymyślony kwadrat, po bokach którego robot będzie się poruszał (patrz rysunek po prawej).

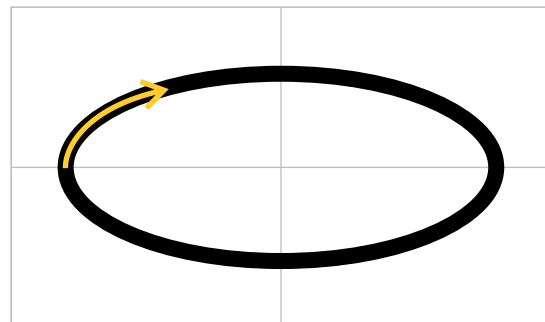


Engino model robota

**Ćwiczenie 1.** Na PC lub urządzeniu typu smart umieść w Symulatorze odpowiednie ikony zgodnie z modelem robota. Przeciągnij niezbędne bloki akcji wewnątrz diagramu przepływu tak, aby utworzyć następujący program, który będzie poruszał się po bieli i ustalał drogę po czerni: **podczas gdy (while)** czujnik podczerwieni w porcie 1 wykrywa biel, silnik w porcie A powinien obracać się do przodu; **podczas gdy (while)** czujnik podczerwieni w porcie 2 wykrywa biel, silnik w porcie B powinien obracać się do przodu.

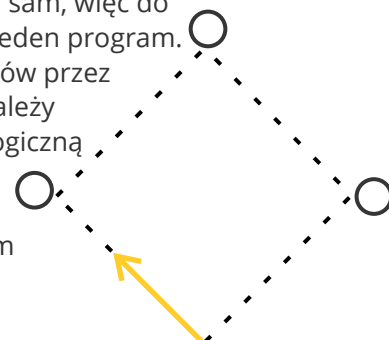
Po umieszczeniu właściwych bloków będziesz musiał **skalibrować czujniki podczerwieni**, aby podążyły za linią. Zapoznaj się z instrukcją obsługi w rozdziale „Kalibracja czujnika podczerwieni” na stronie 25.

Dzięki temu programowi robot będzie podążał za czarną linią. Aby przetestować program, umieść robota na utworzonej czarnej elipsie i zobacz, czy porusza się po czarnej linii.



**Ćwiczenie 2.** Przeciągnij i upuść niezbędne bloki wewnątrz diagramu przepływu danych, aby utworzyć następujący program: robot porusza się do przodu; **jeśli (if)** wykryje biały przedmiot za pomocą **jednego lub obu** czujników, robot powinien się zatrzymać; po 5 sekundach humanoid powinien obrócić się o 90 stopni w prawo, aby wykryć następny obiekt.

Wskazówki: kąt jest zawsze taki sam, więc do skręcania potrzebny jest tylko jeden program. Aby ustawić wykrywanie obiektów przez oba czujniki lub jeden z nich, należy wybrać odpowiednią bramkę logiczną (AND lub OR). Powinieneś także ponownie **skalibrować czujniki podczerwieni**, aby tym razem aktywować je do rozpoznawania obiektów.



- Jak działają roboty stacjonarne.
- Jak działa instrukcja warunkowa REPEAT TIMES.
- Jak działa instrukcja warunkowa REPEAT UNTIL.

### Przykład z życia: ramię przemysłowe

Roboty są często wykorzystywane w warunkach przemysłowych, odgrywając kluczową rolę na liniach produkcyjnych fabryk. Najczęstszą formą zastosowania stanowią ramiona robotyczne, które mogą przesuwać elementy z jednego punktu do drugiego za pomocą technologii czujnikowej. Dzięki modelowi twistera możesz wypróbować symulowaną wersję tych sytuacji.

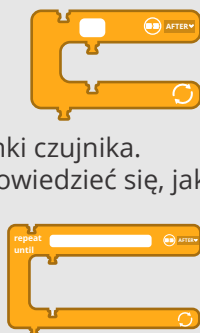
Poziom trudności ★★★★★

#### Potrzebne materiały:

- Creative Engineering 100 w 1 Robotized Maker Pro (CE101MP-A).
- PC lub urządzenie typu smart z pobranym i zainstalowanym oprogramowaniem KEIRO.
- Taśma klejąca i 1 pusta lekka puszka.

#### Procedura:

1. Znajdź instrukcje **na stronach 52-57** i zbuduj model **twistera**.
2. Robot stacjonarny, to znaczy, że jest ustawiony na stałej podstawie. Wykonaj **ćwiczenie 1**, aby model był bardziej stabilny podczas obracania się.
3. Precyzyjne wykonywanie powtarzalnych zadań jest bardzo ważną cechą programowania. Wykonaj **ćwiczenie 2**, aby utworzyć program pętli za pomocą bloku „repeat times”.
4. Czasami chcemy, aby pętla powtarzała się, dopóki nie zostaną spełnione warunki czujnika. Wykonaj **ćwiczenie 3**, aby dowiedzieć się, jak to zrobić, używając bloku „repeat until”.
5. Roboty stacjonarne często działają na podstawie precyzyjnych instrukcji, aby wykonać rutynową pracę. W tym celu wykorzystuje się technologię wykrywania podczerwieni, dzięki czemu robot może sam korygować, gdzie ma się poruszać. Jeśli ustawienie czasu jest jedyną stosowaną metodą, bez względu na to, jak jest dokładny, ruch robota będzie nieprawidłowy po wielu powtórzeniach. Wykonaj **ćwiczenie 4**, aby zasymulować rutynową pracę robotów.



Model Engino twister

**Ćwiczenie 1.** Utwórz ręcznie prosty program do obracania ramienia. Umieść robota na podłodze i zobacz, co się stanie, gdy uruchomisz program. Zauważysz, że podstawa również się obraca. Aby rozwiązać ten problem, podłącz żółte koło pasowe pod każdym z 4 rogów podstawy. Aby uzyskać większą stabilność, możesz umieścić koła pasowe na taśmie klejącej. Spróbuj ponownie.



**Ćwiczenie 2.** Na komputerze lub urządzeniu typu smart umieść odpowiednie ikony na Symulatorze zgodnie z modelem twistera. Przeciągnij i upuść niezbędne bloki wewnątrz diagramu przepływu, aby utworzyć następujący program: *twister powinien obracać się przez 1 sekundę z prędkością ustawioną na 50, a następnie zamknąć i otworzyć ramię; ta procedura powinna powtórzyć się 5 razy.*

Wskazówka: użyj bloku „repeat times”.

**Ćwiczenie 3.** Stwórz następujący program (na początku uchwyt ramienia powinien być otwarty): *twister powinien obracać się w sposób ciągły dopóki (until) czujnik podczerwieni nie wykryje jakiś przedmiot lub dłoń; jeśli w czasie, gdy model jest nieaktywny, postawimy puszkę przed czujnikiem podczerwieni ramienia, chwyt powinien się zamknąć by ją złapać.*

Wskazówka: użyj bloku „repeat until” dla pierwszej części programu i w razie potrzeby wyreguluj ramię, aby uzyskać lepszy chwyt.

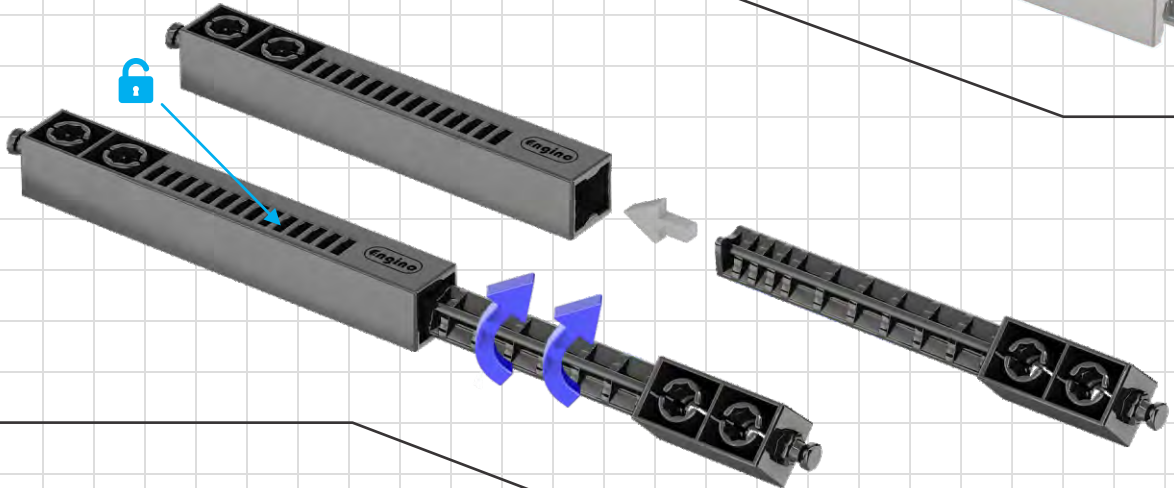
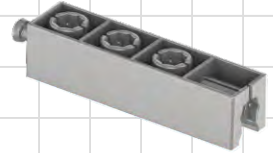
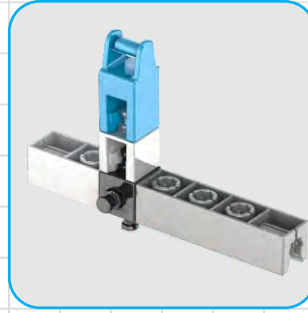
**Ćwiczenie 4.** Umieść puszkę w ramieniu modelu i utwórz następujący program: *twister powinien obrócić się o 180 stopni w lewo trzymając puszkę; gdy model się zatrzyma, ramię powinno się otworzyć i upuścić puszkę; następnie twister powinien obrócić się o 180 stopni w prawo (czyli z powrotem do pozycji wyjściowej); jak tylko umieścisz puszkę przed czujnikiem ramienia, model powinien chwycić ją i powtórzyć procedurę (obrót o 180 stopni, upuszczenie puszki i odwrócenie).*

Wskazówka: umieść dwa przedmioty (lub użyj dłoni) po jednej ze stron środkowego czujnika, aby go uruchomić.

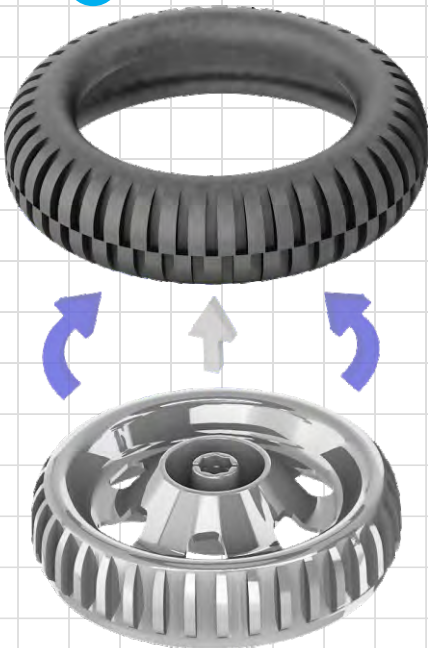




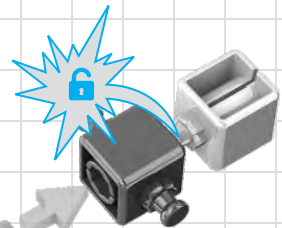
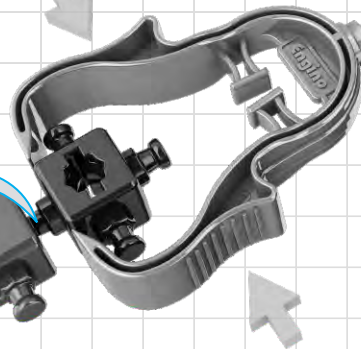
## Wskazówki dotyczące budowania



tire  
L



Podwójne narzędzie do rozłączania



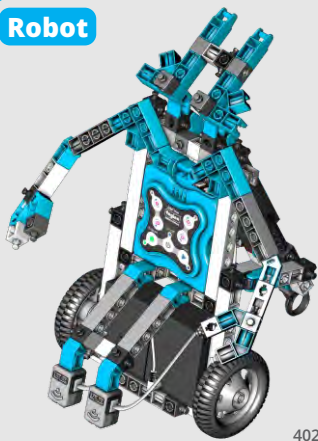
**OPCJA 2:**  
Wciśnij do otworu, aby wysunąć zablokowaną część!

**OPCJA 1:**  
Ściśnij narzędzie, aby zdemontować małe części!



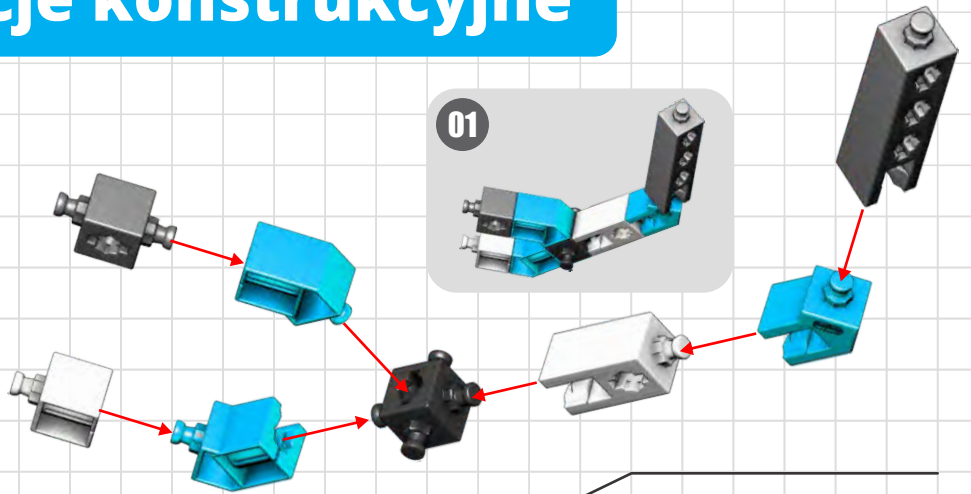
# Instrukcje konstrukcyjne

Robot

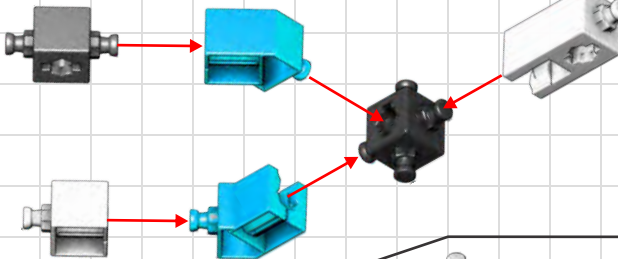
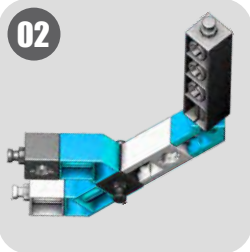


4023

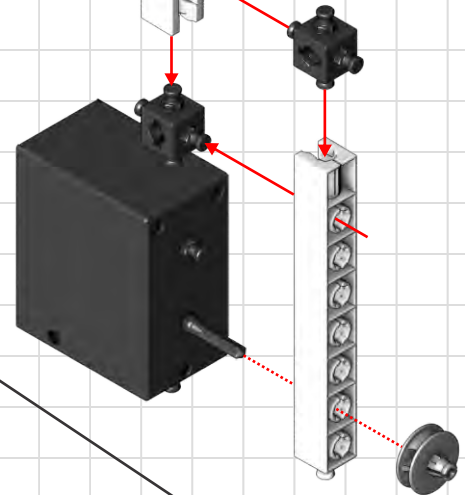
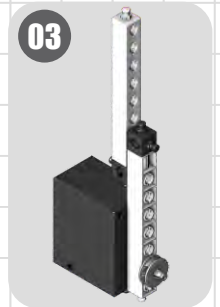
01



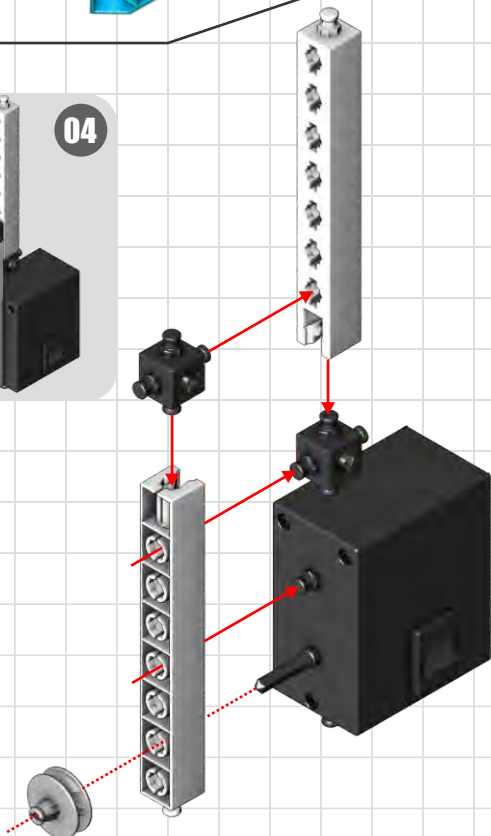
02



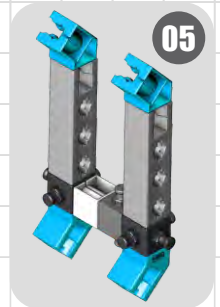
03

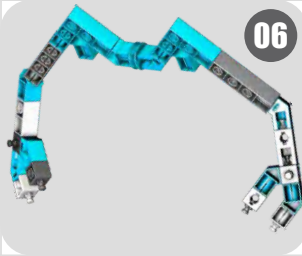


04

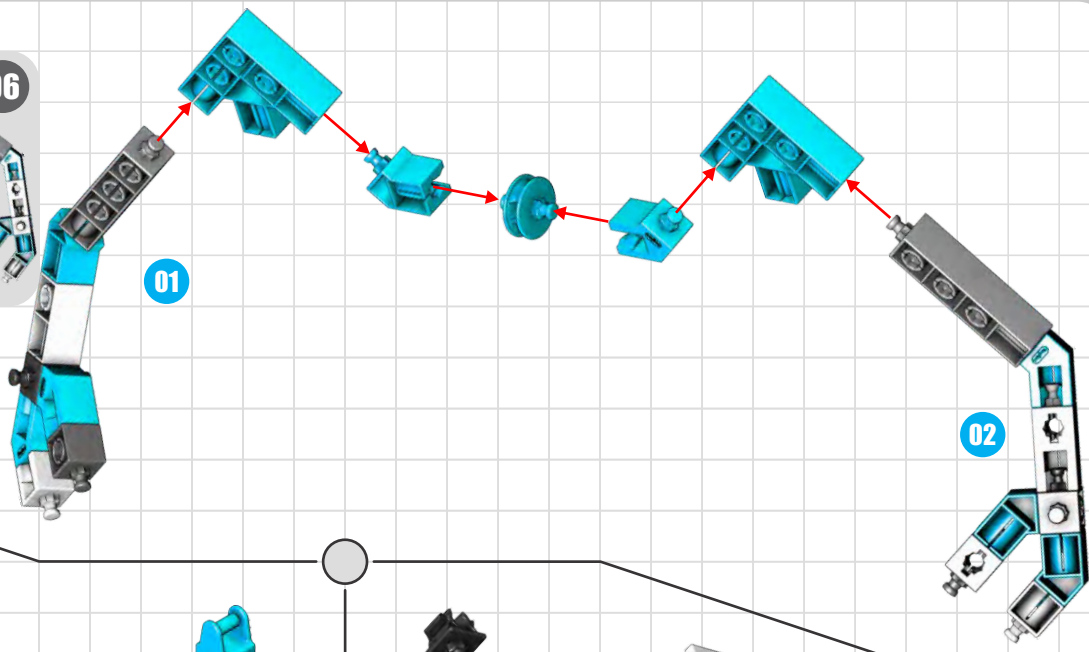


05

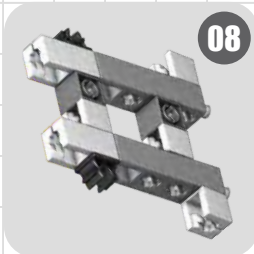




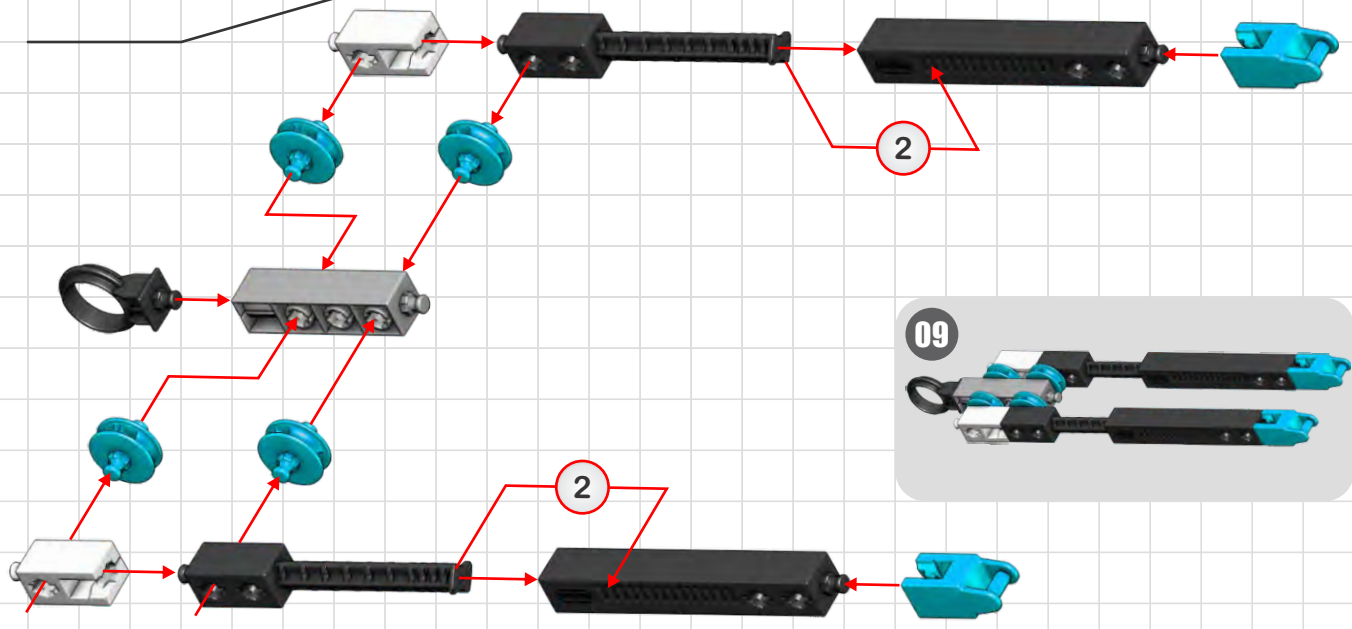
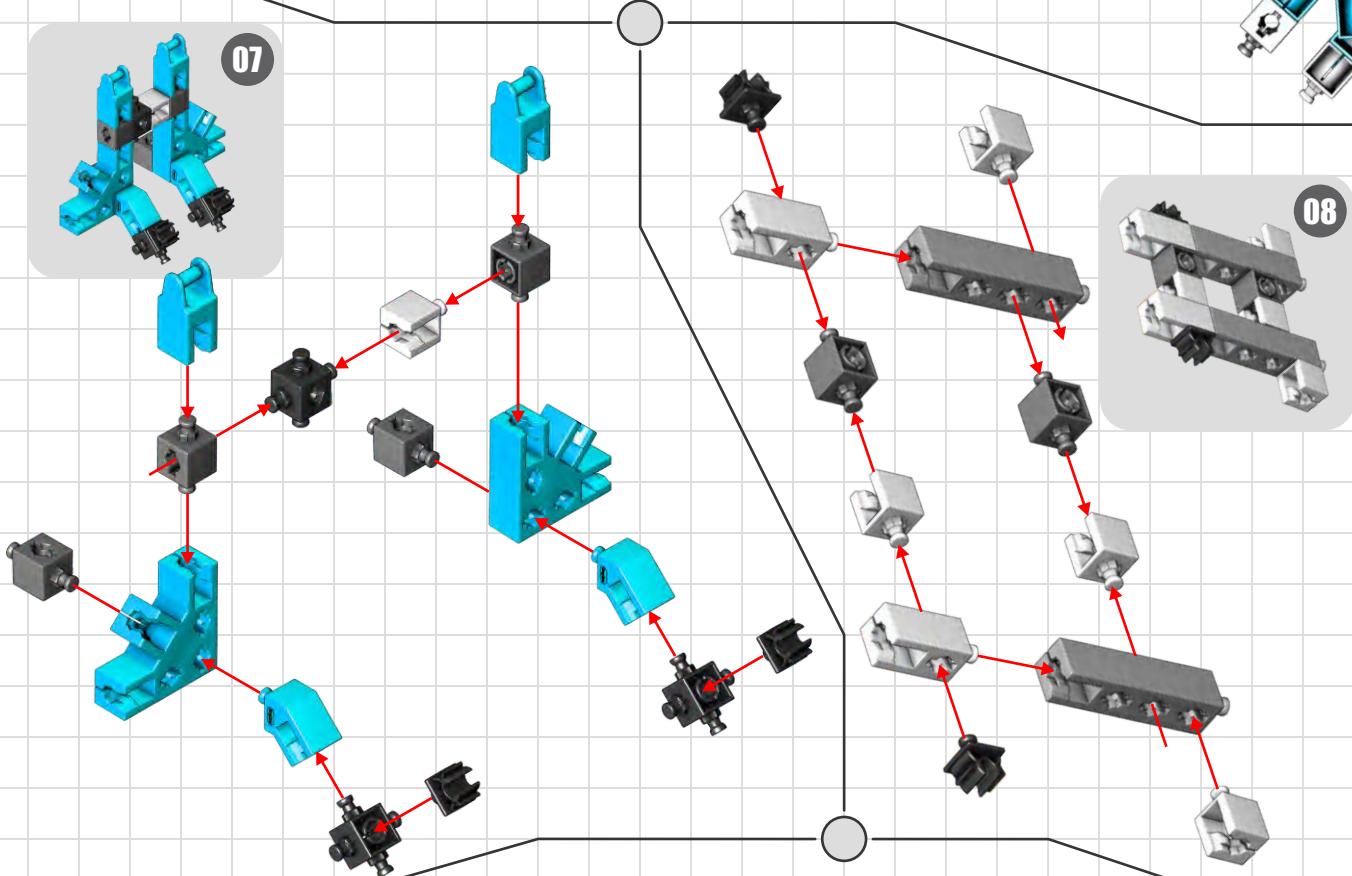
06



07

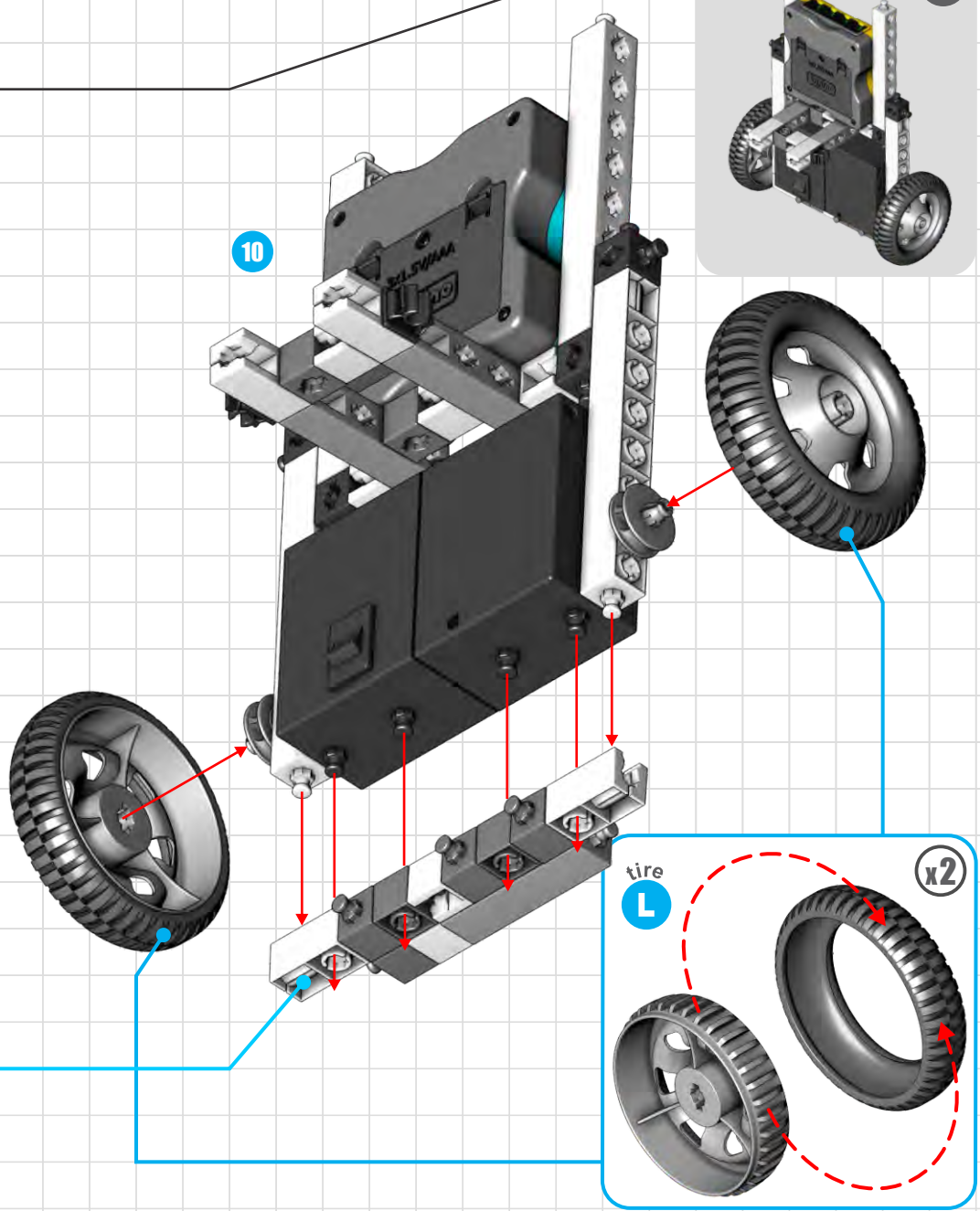
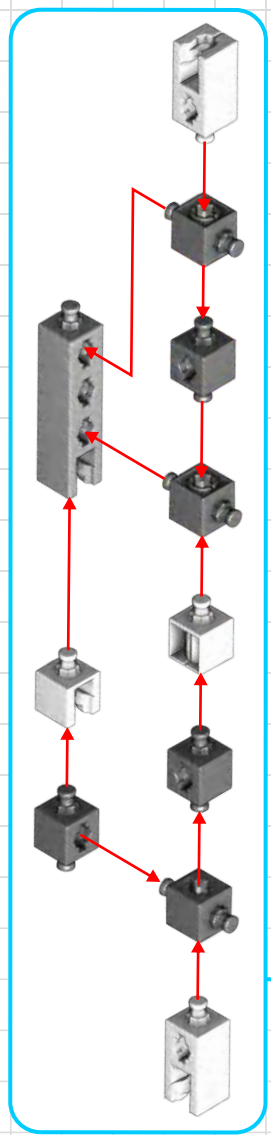
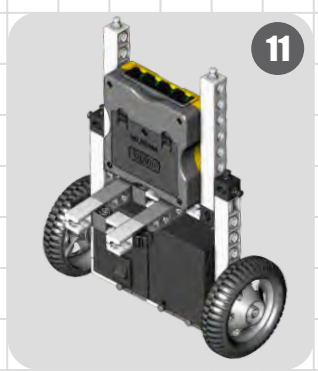
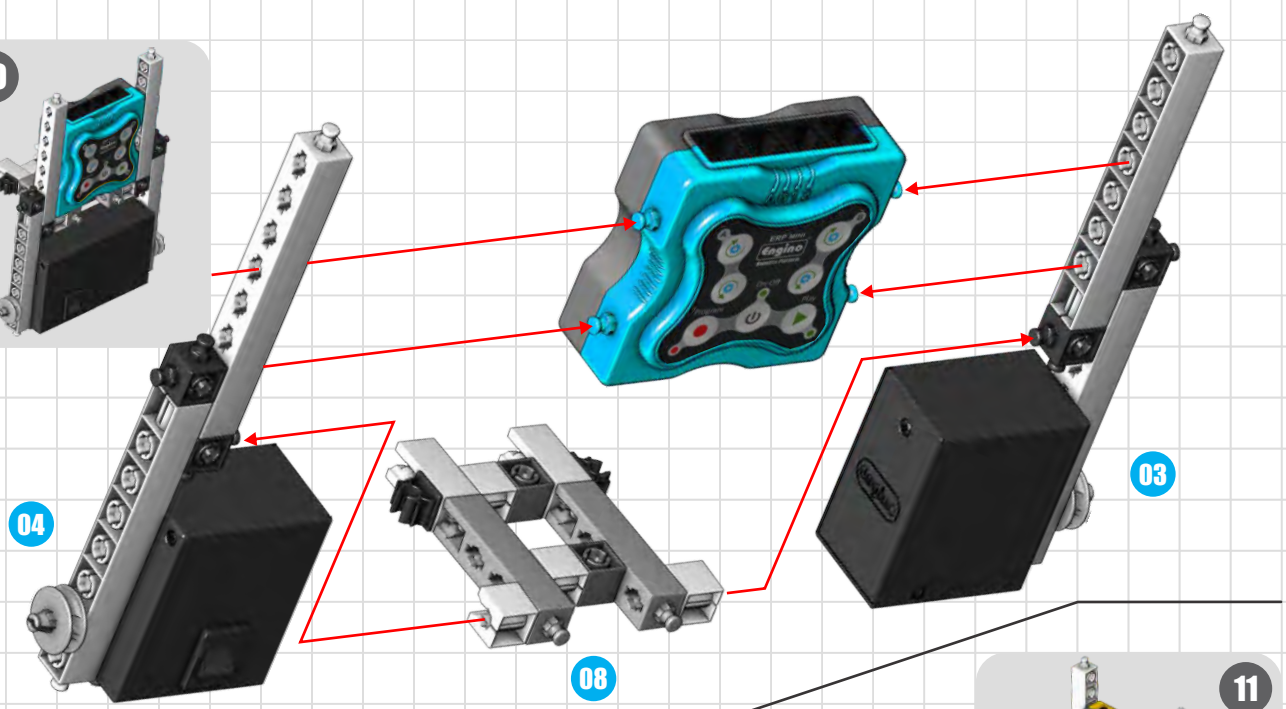


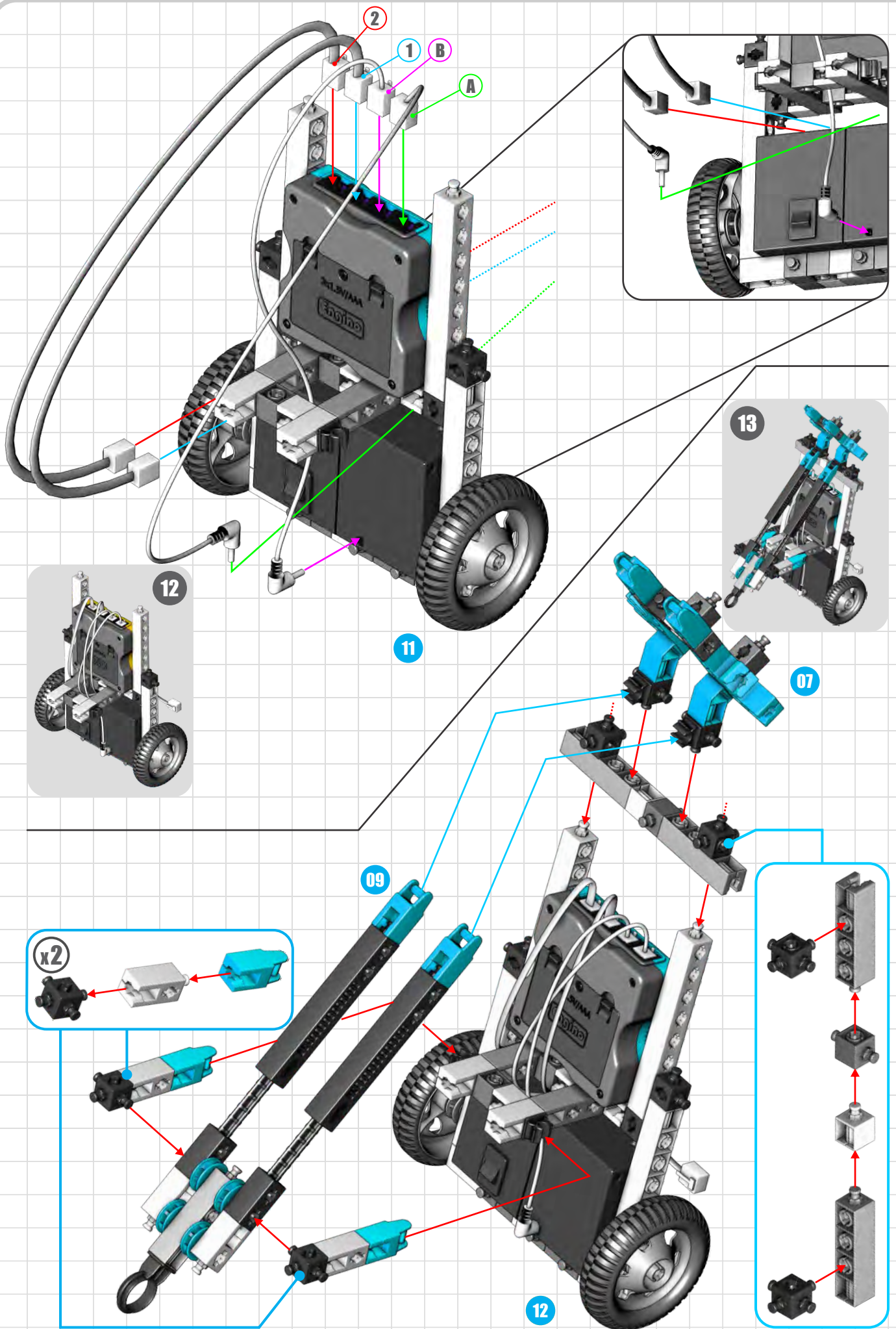
08



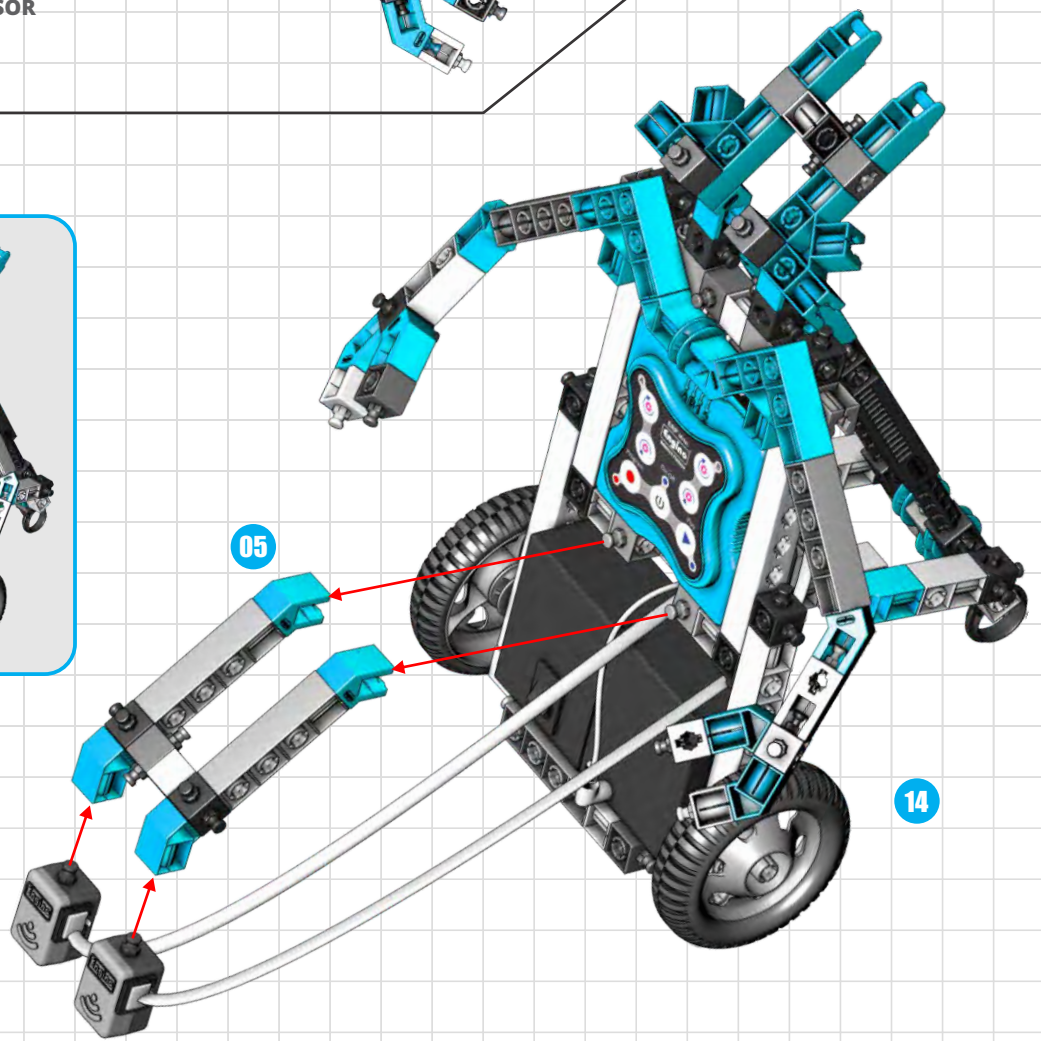
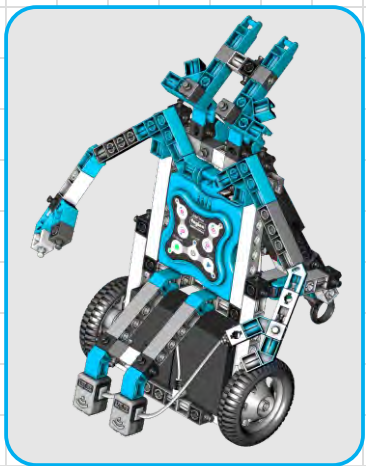
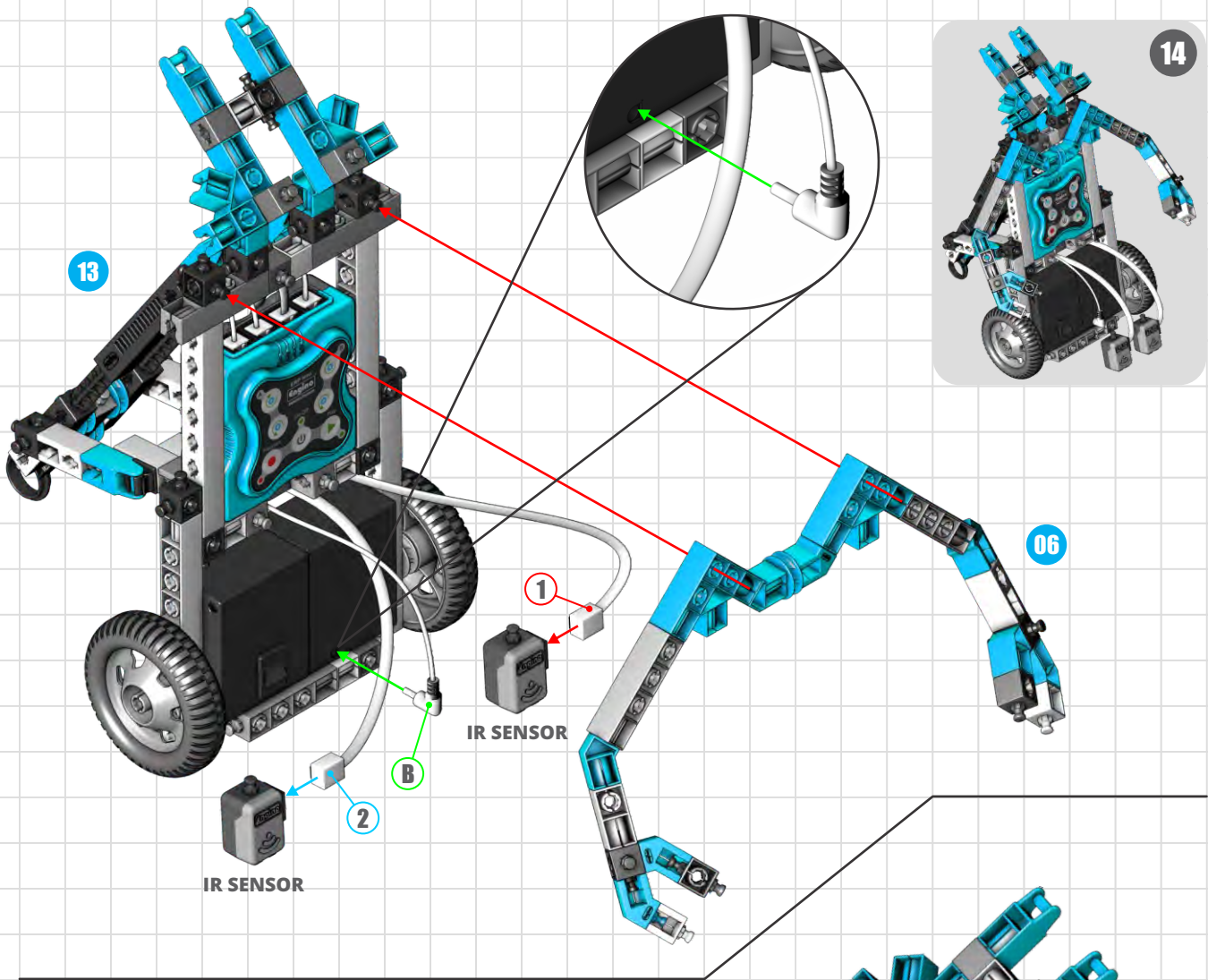
09









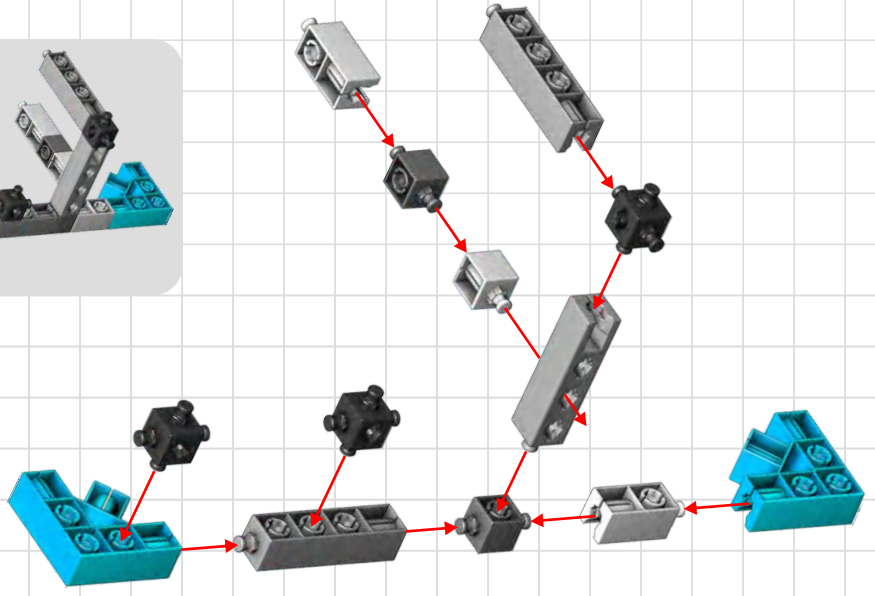
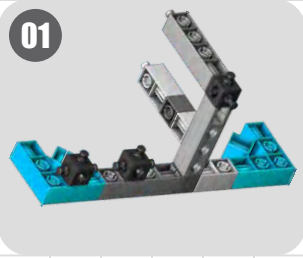




Wentylator

4020

01



02

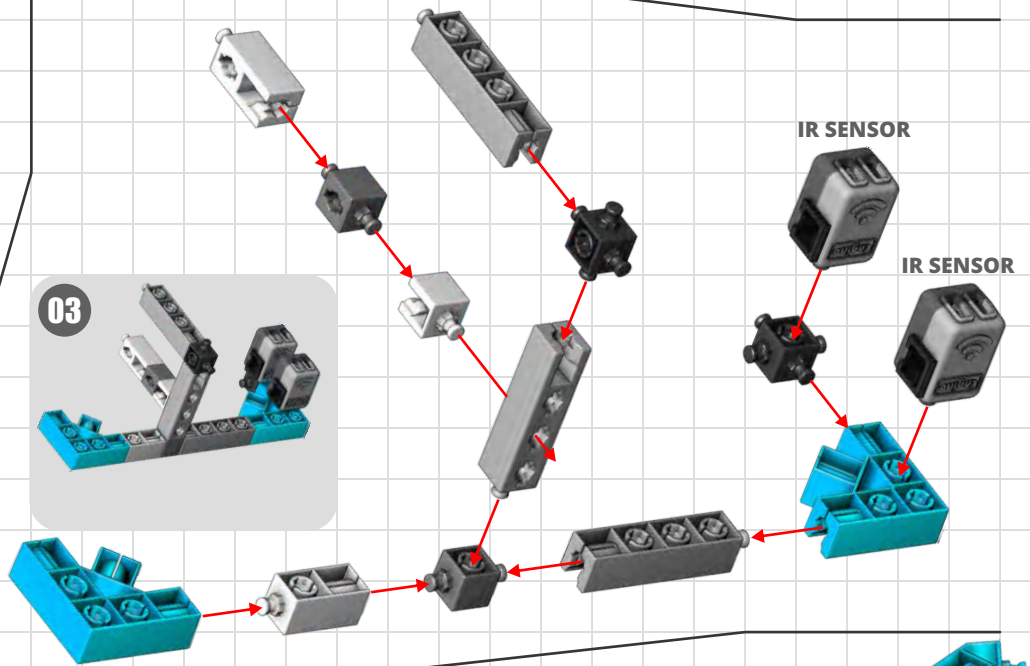


03

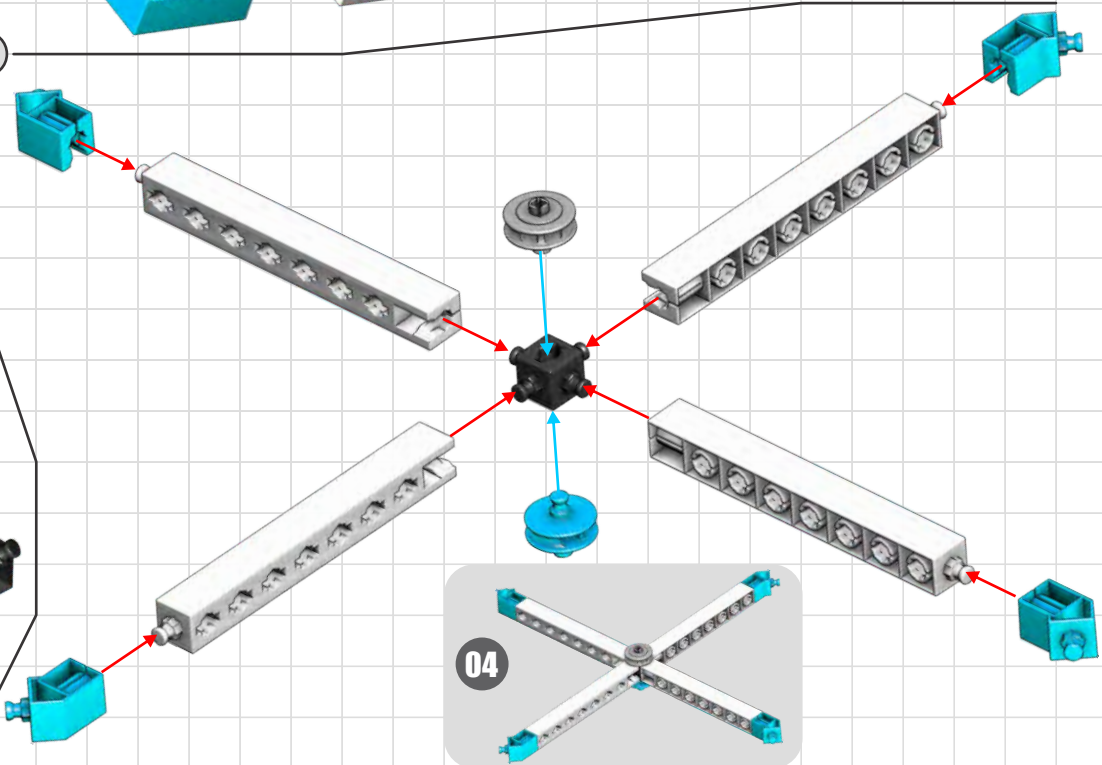
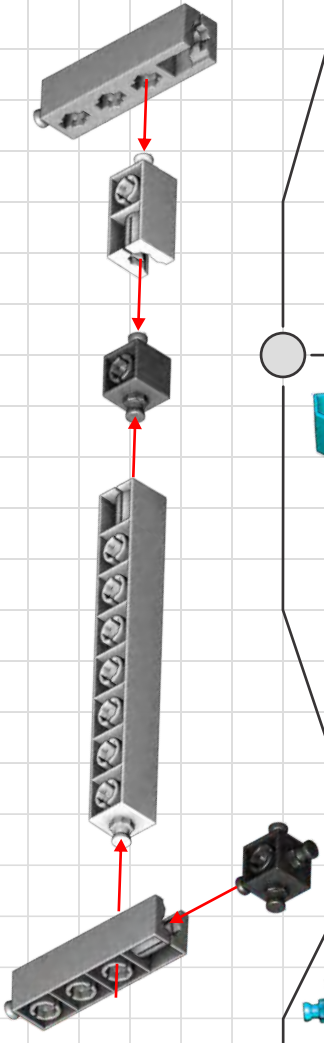
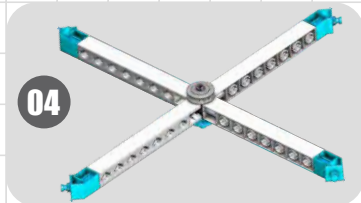


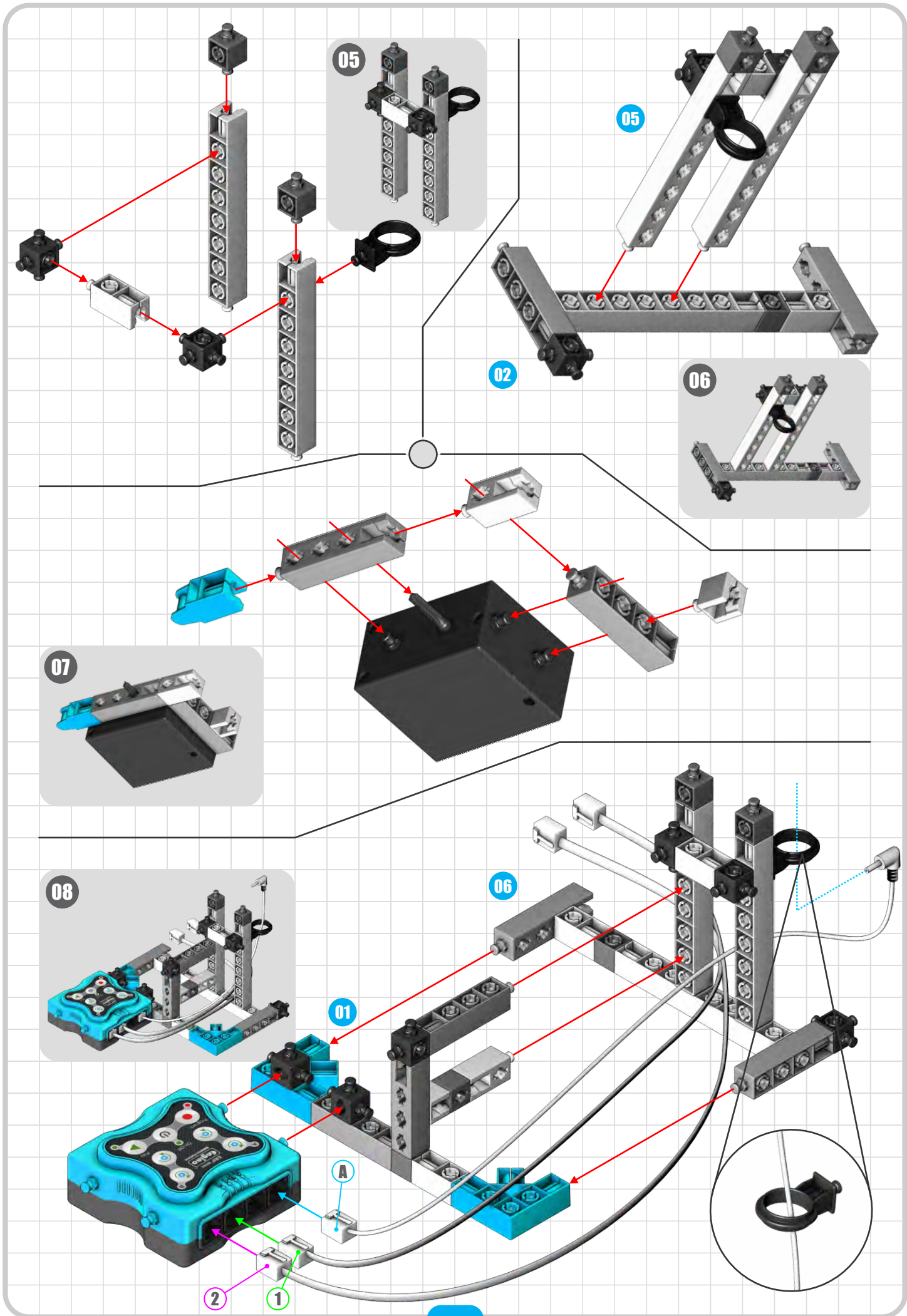
IR SENSOR

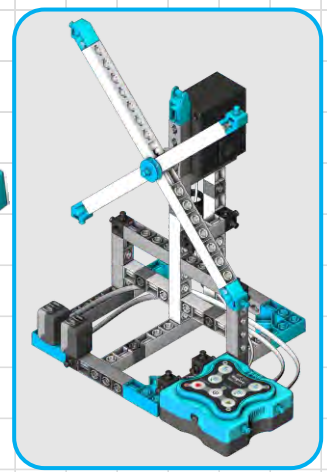
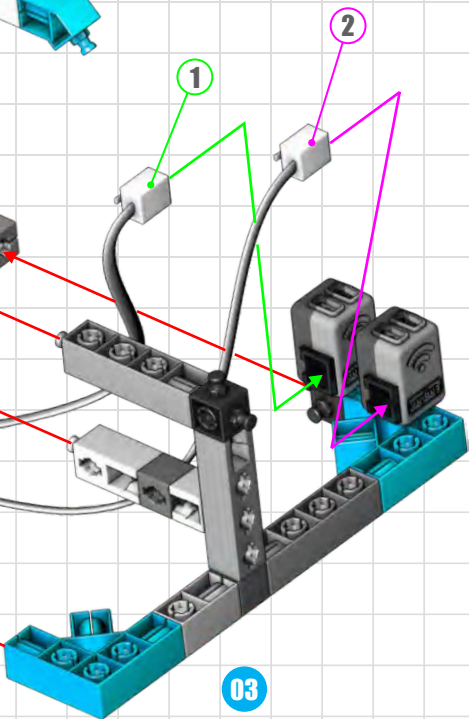
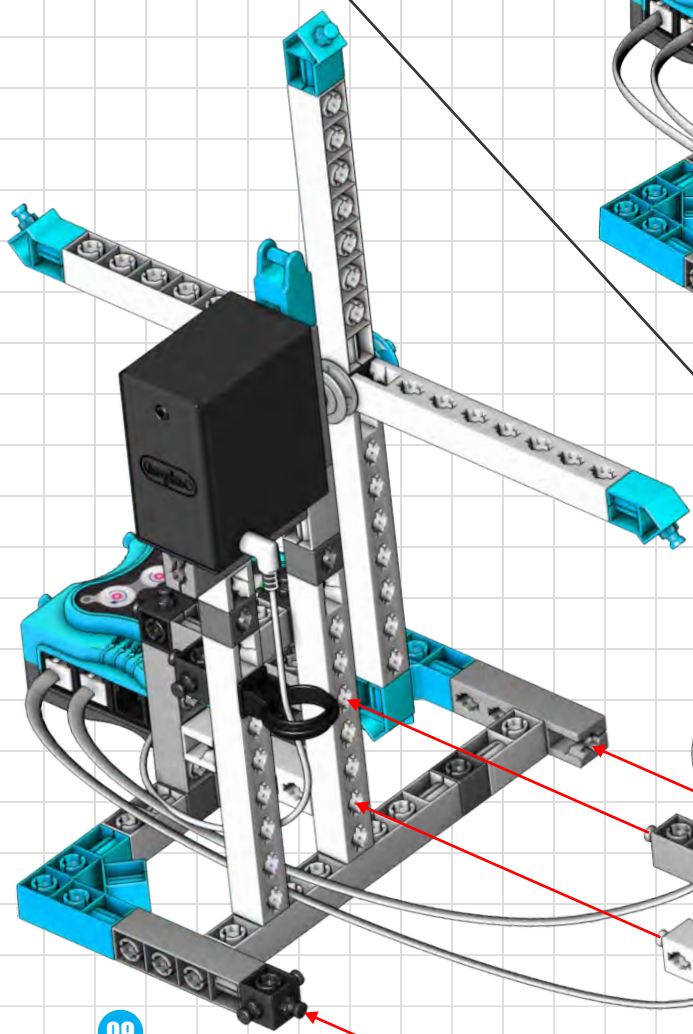
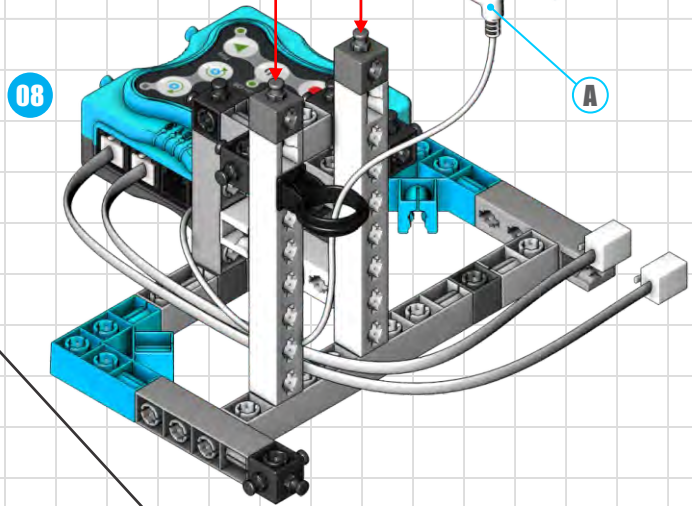
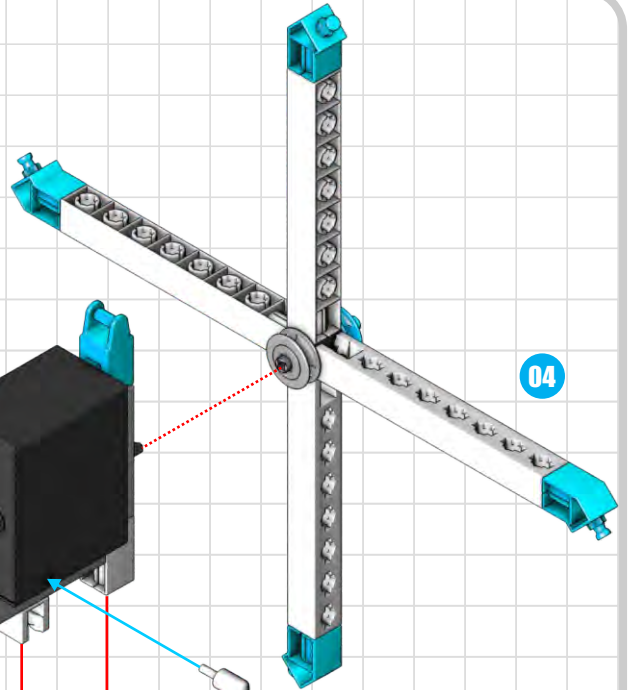
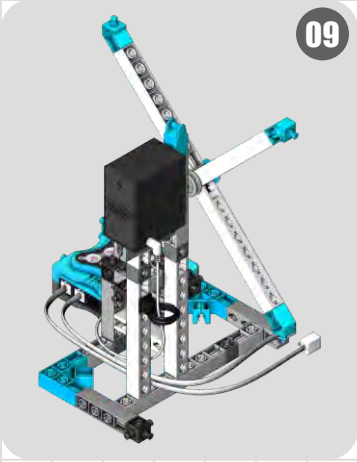
IR SENSOR



04





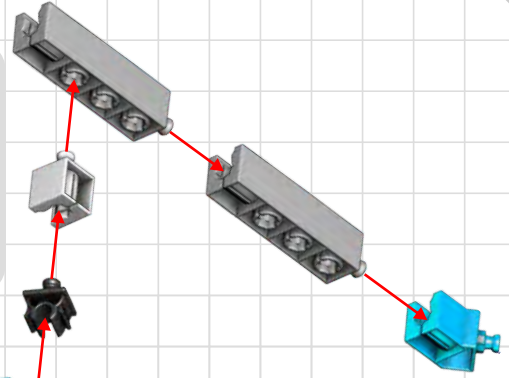
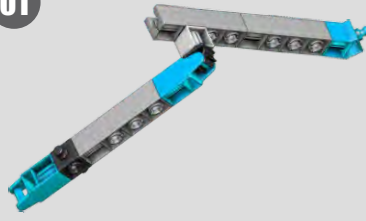




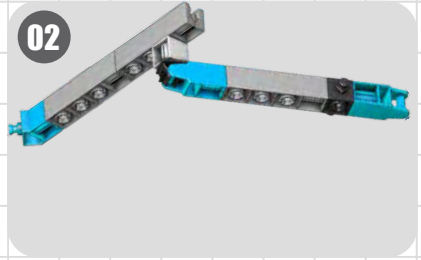
# Hexapod

4021

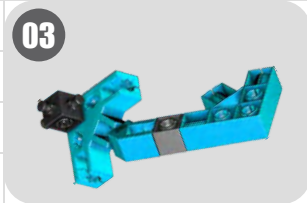
01



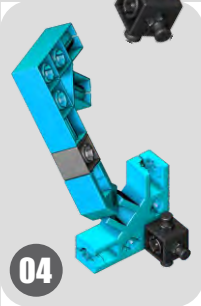
02



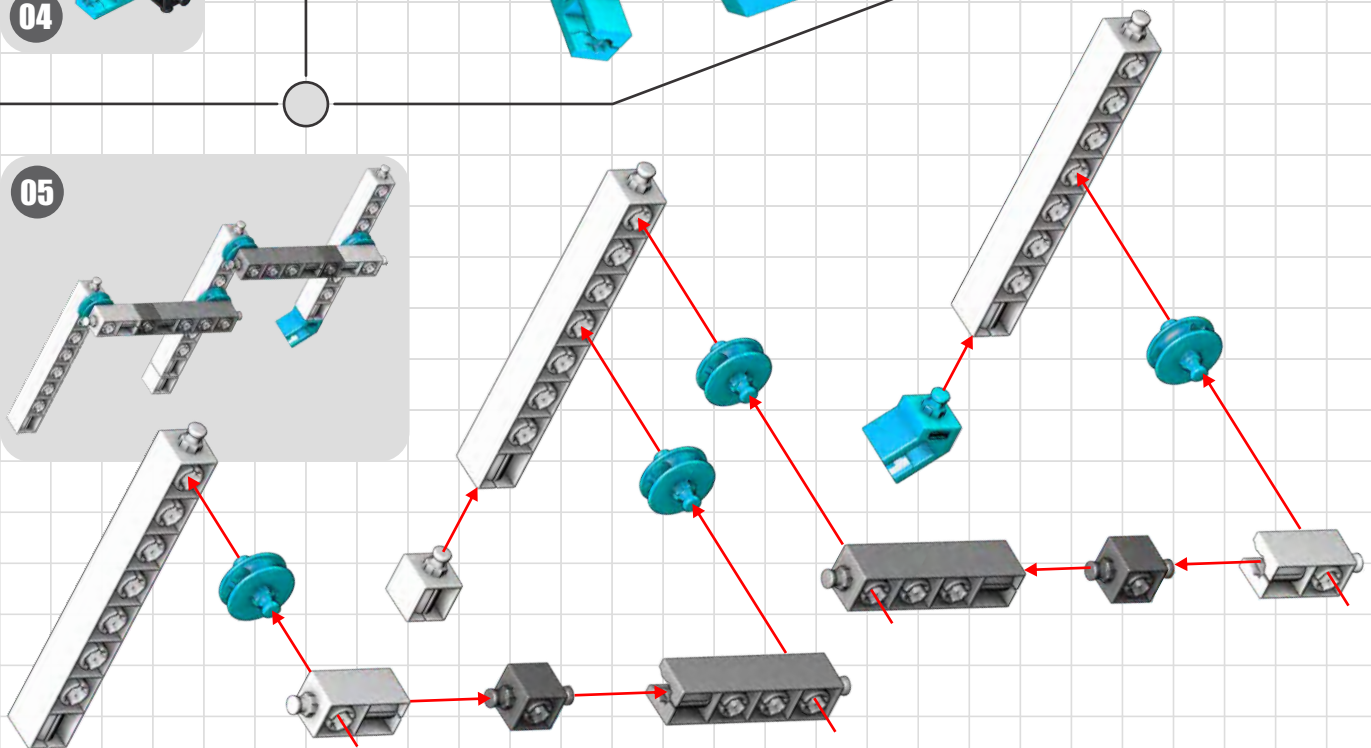
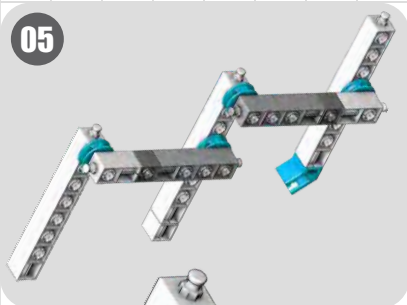
03

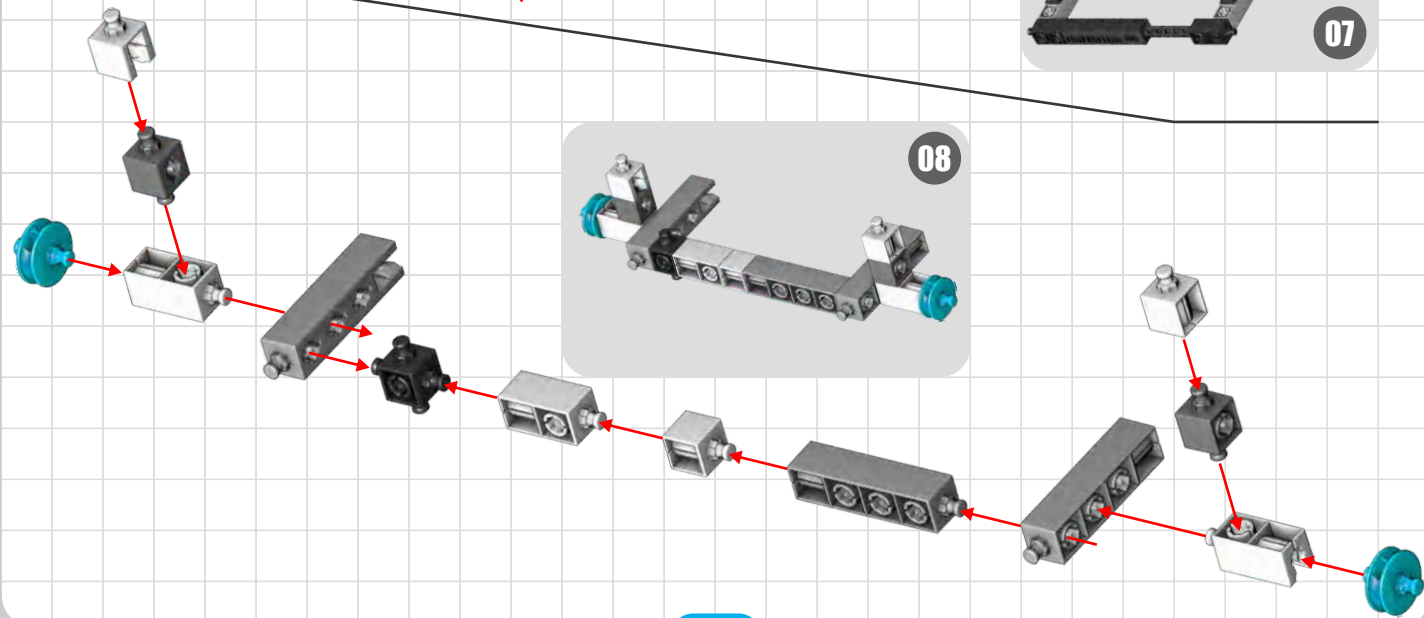
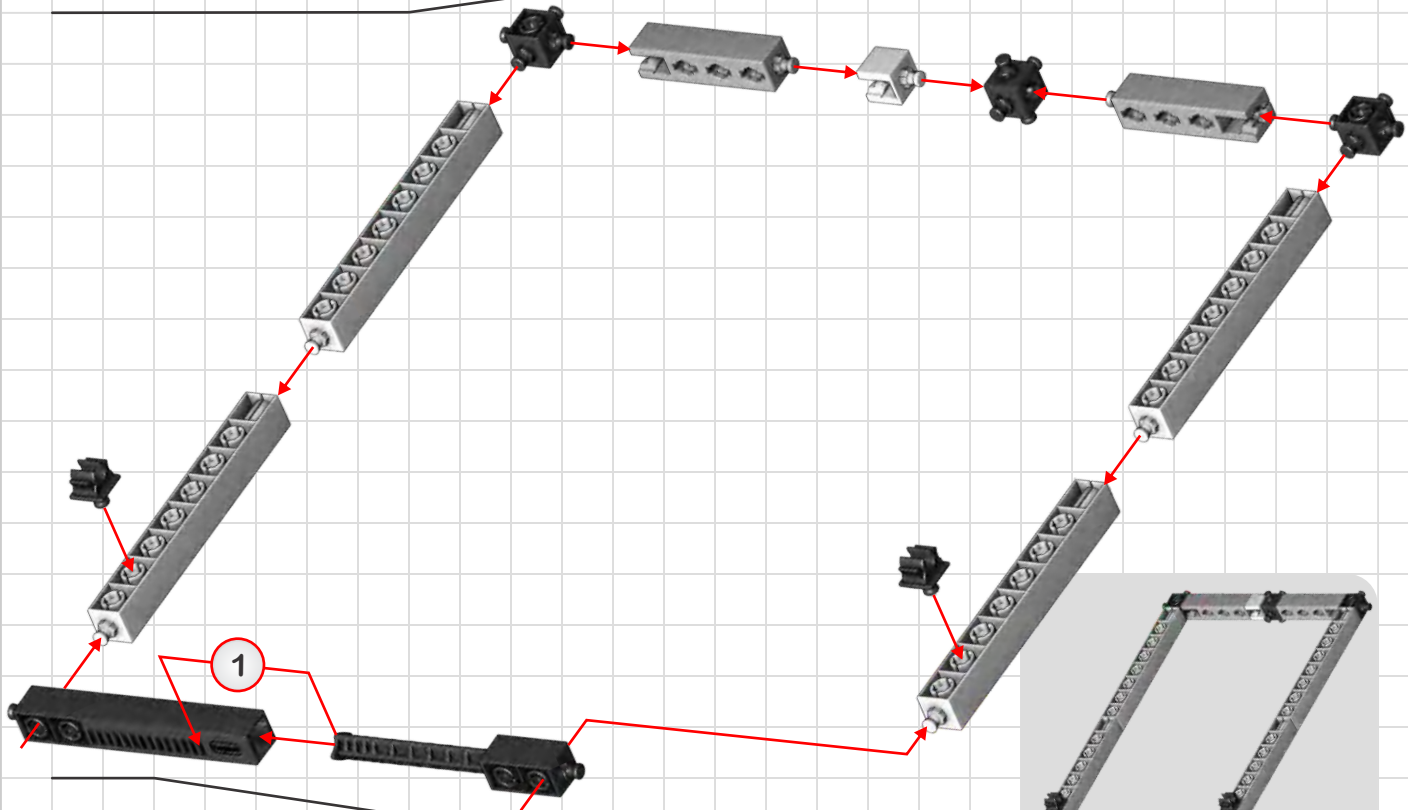
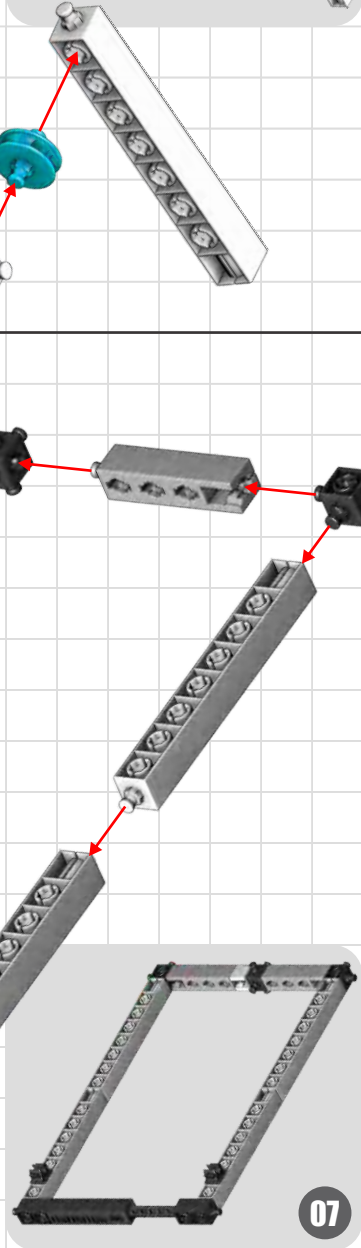
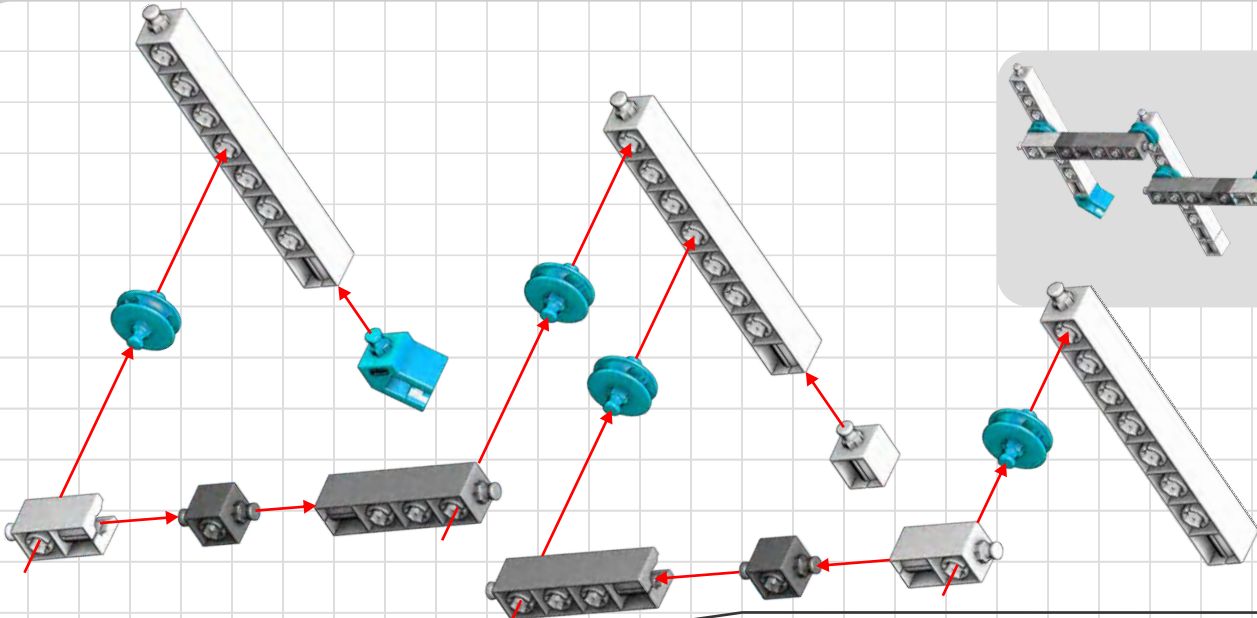
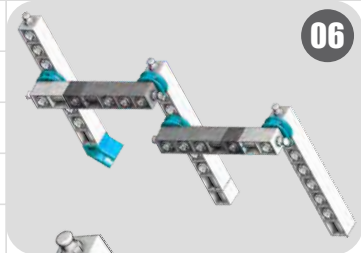


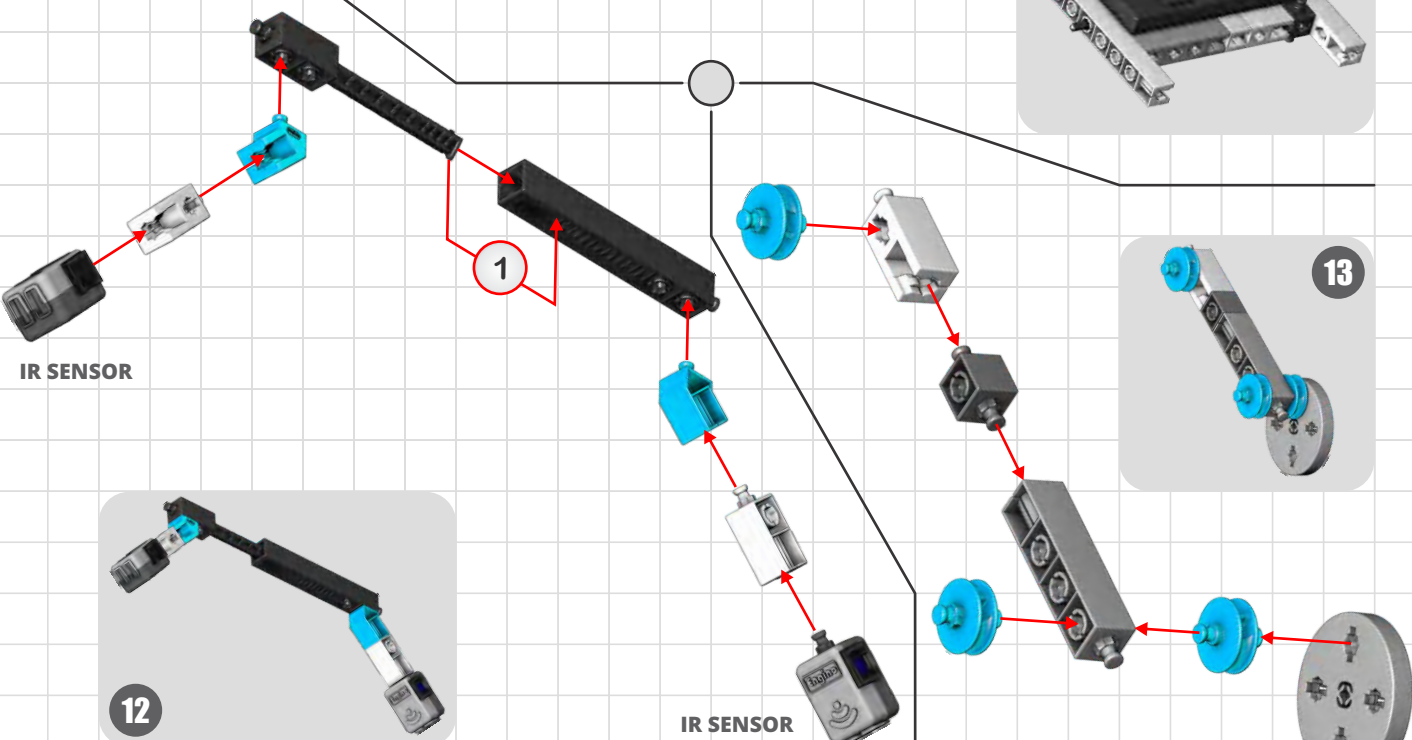
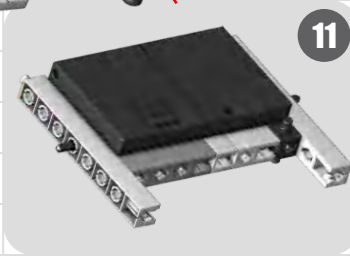
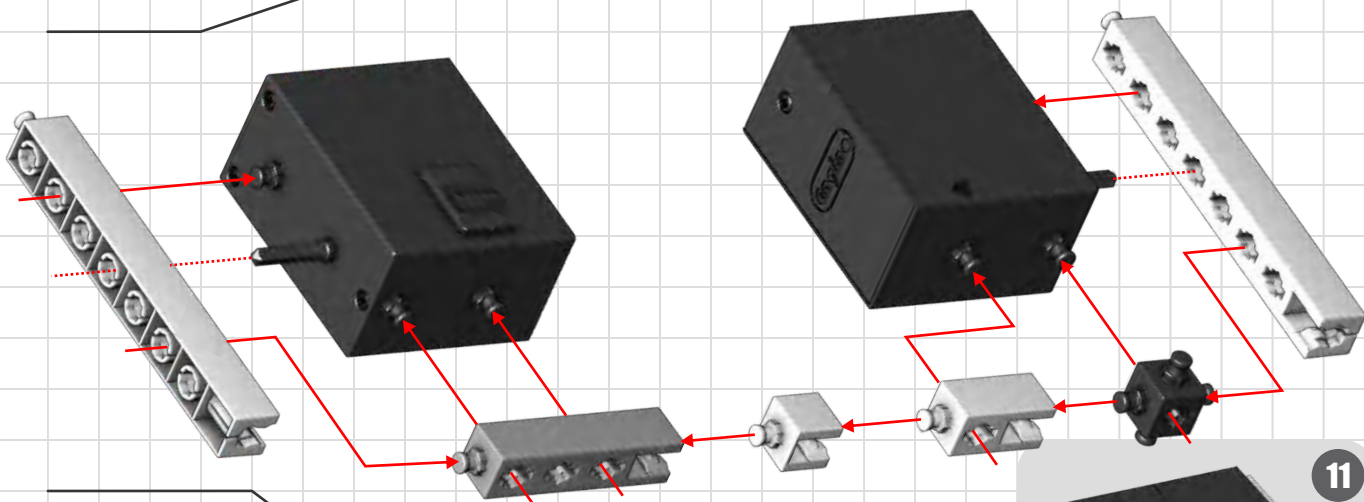
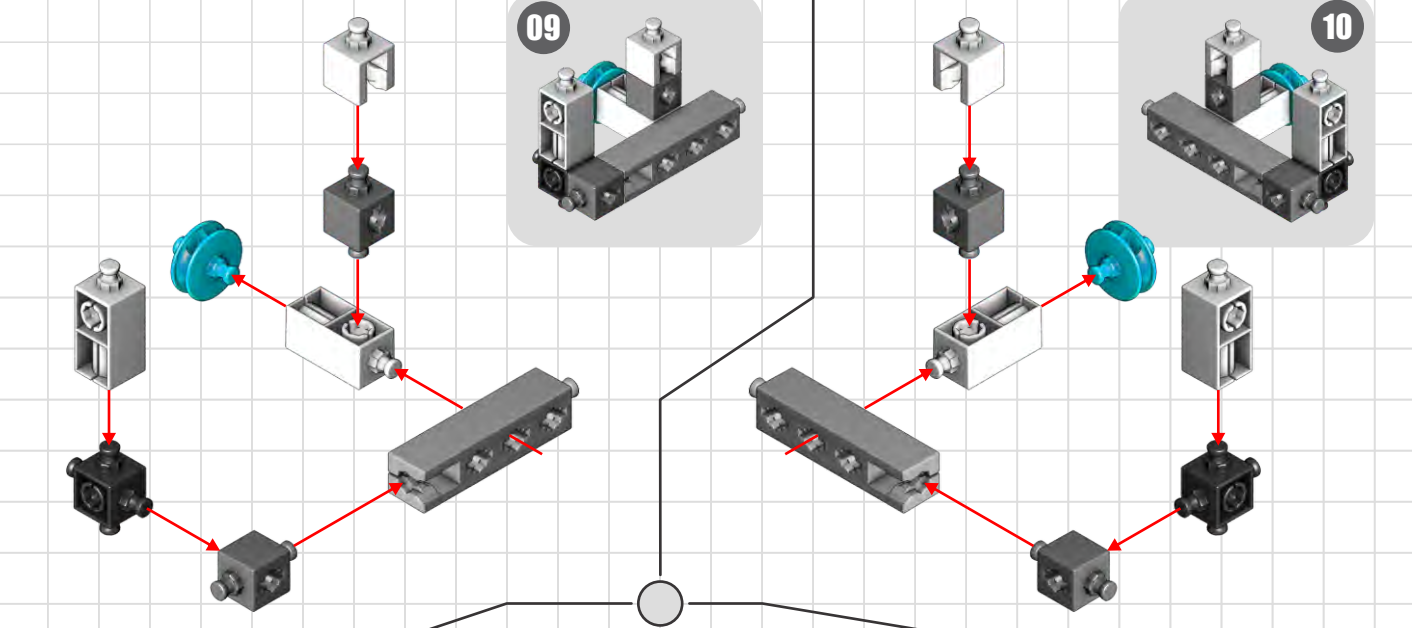
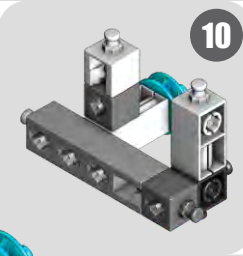
04



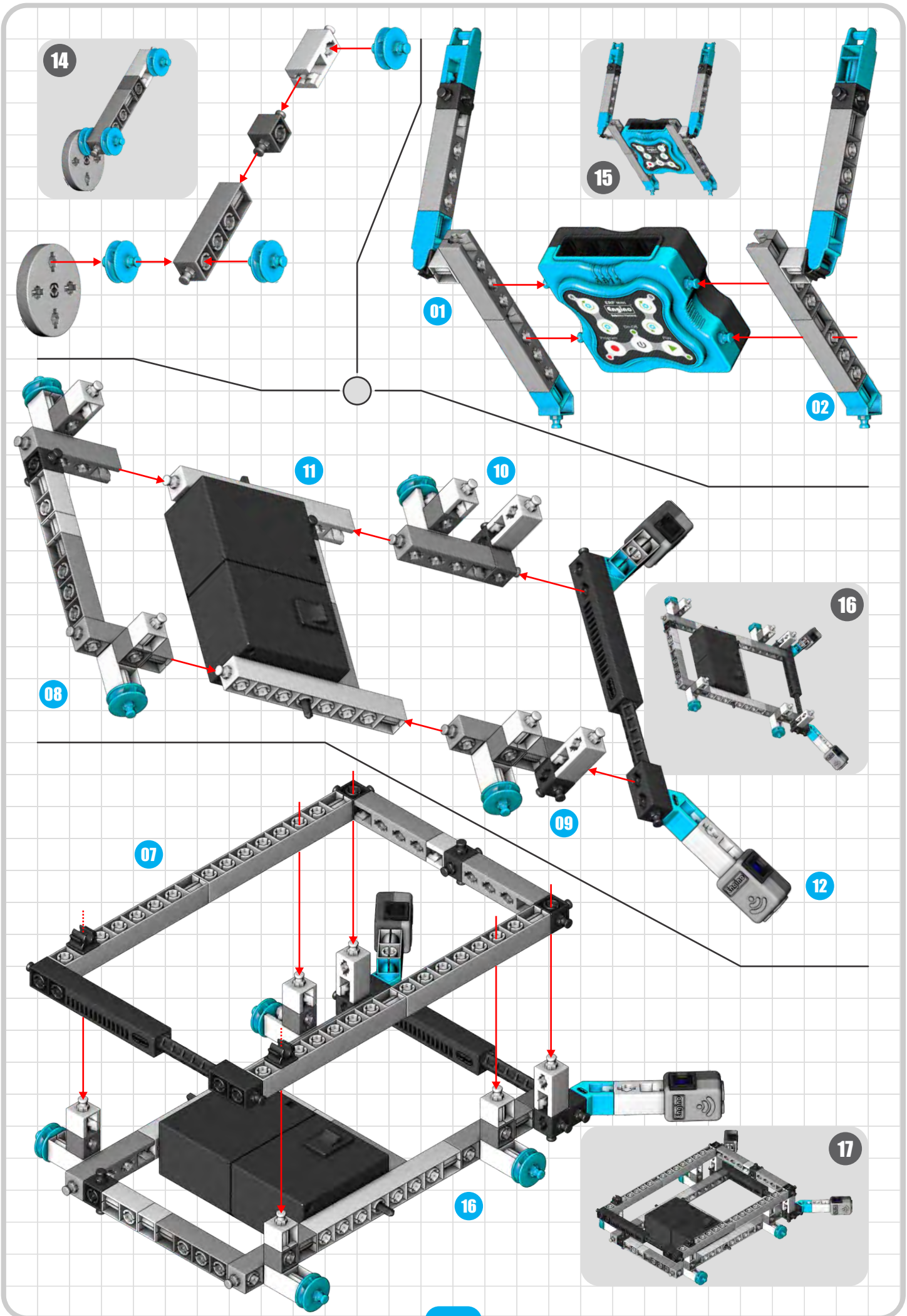
05

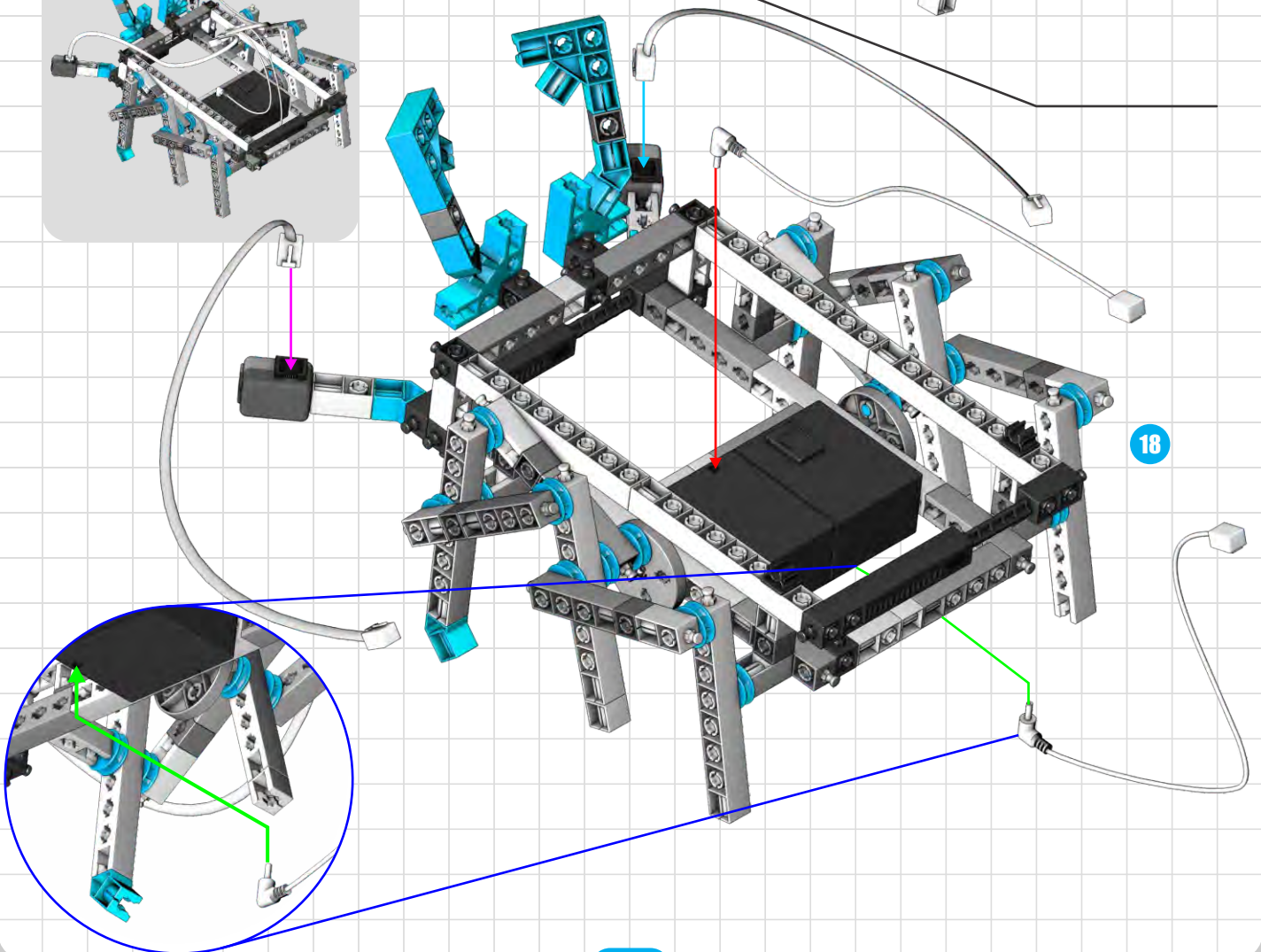
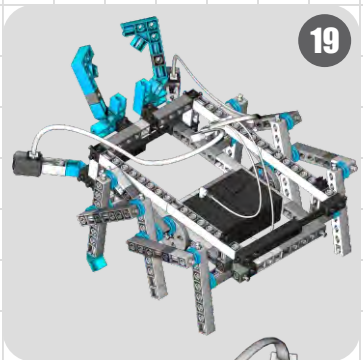
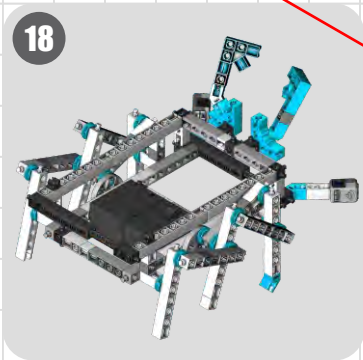
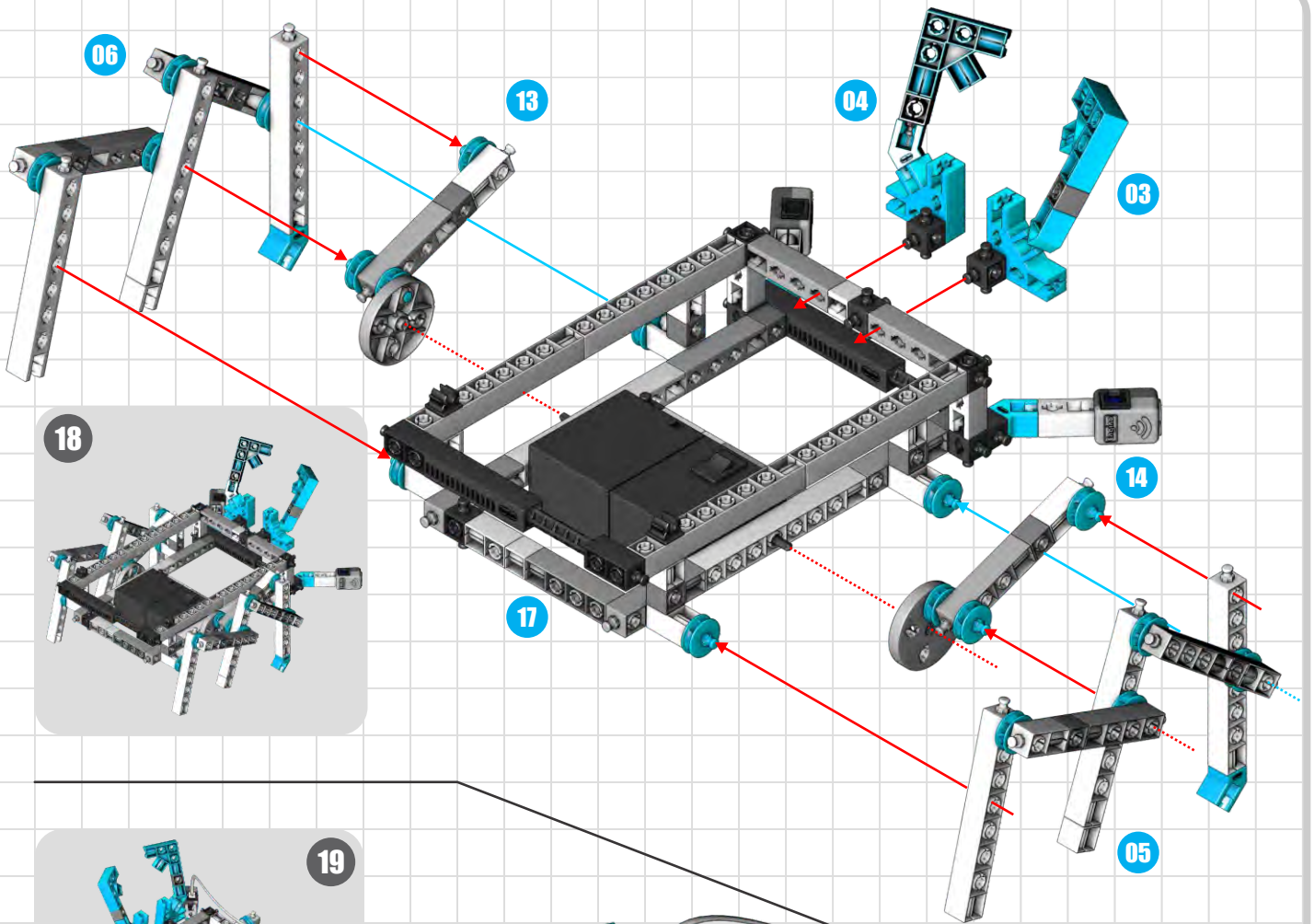




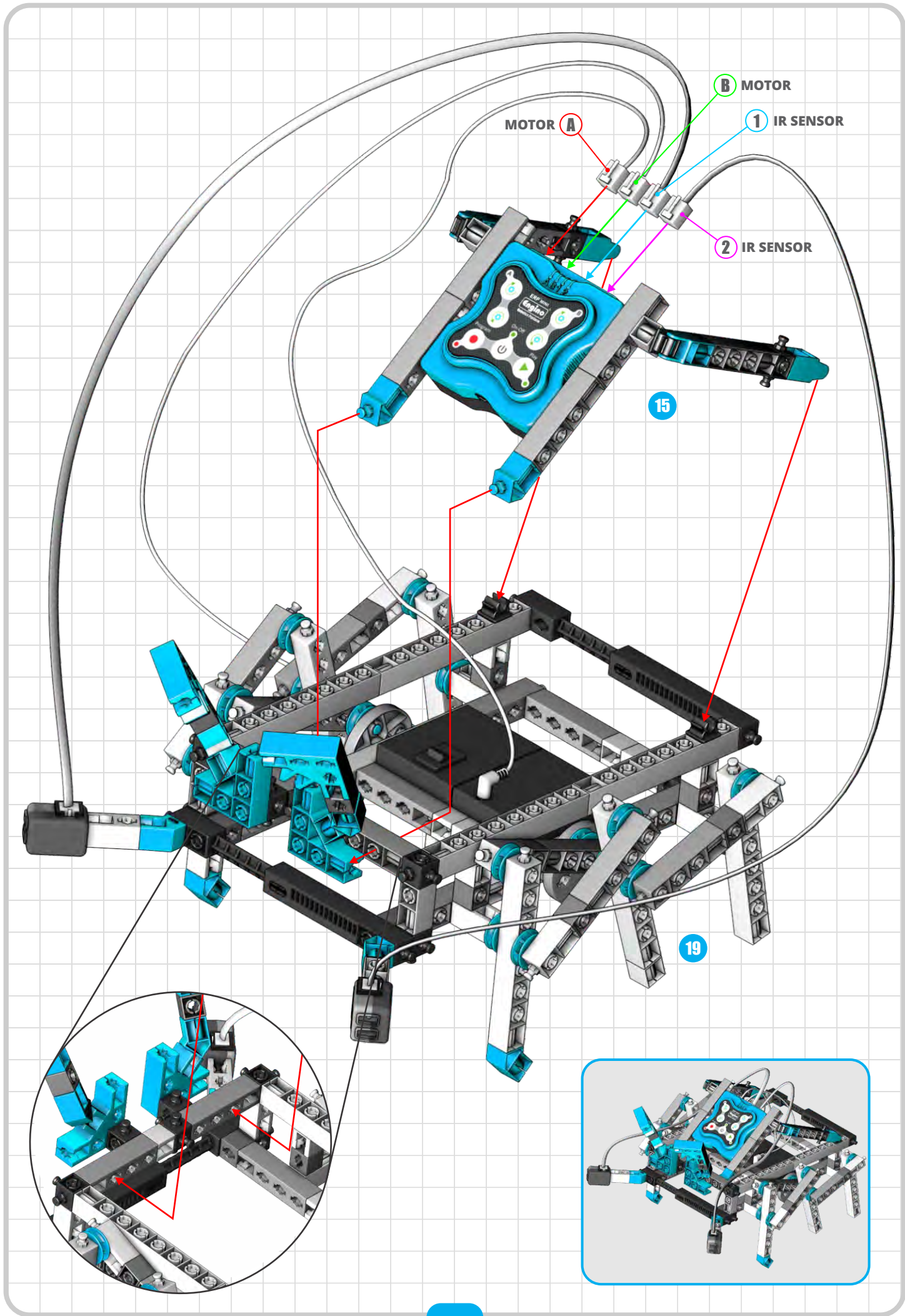














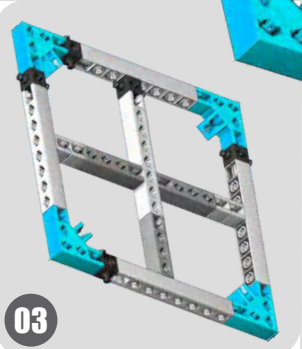
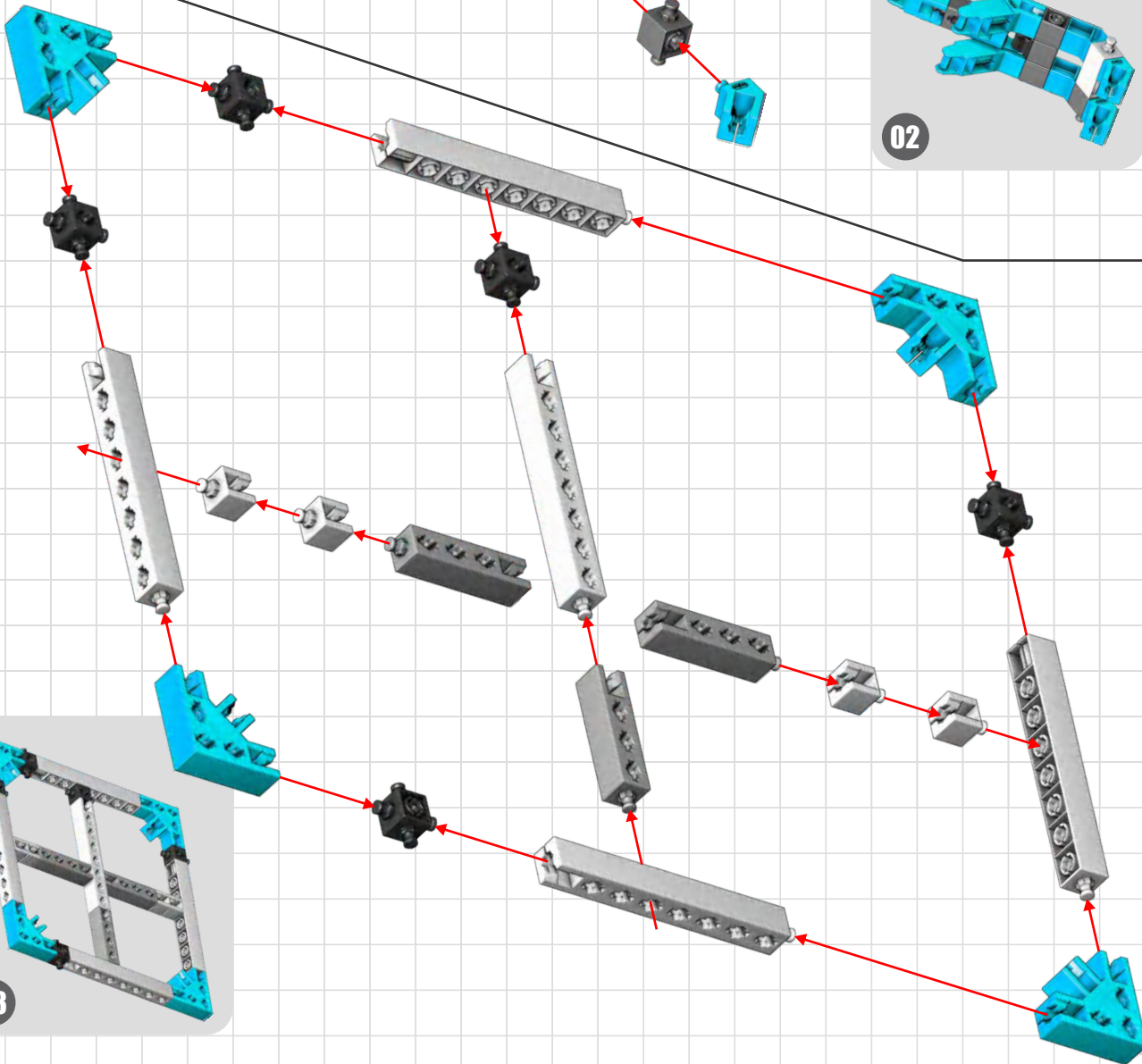
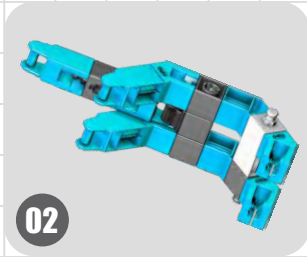
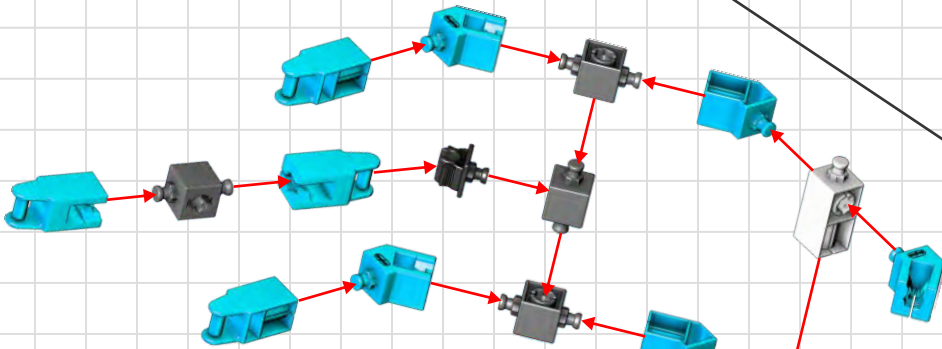
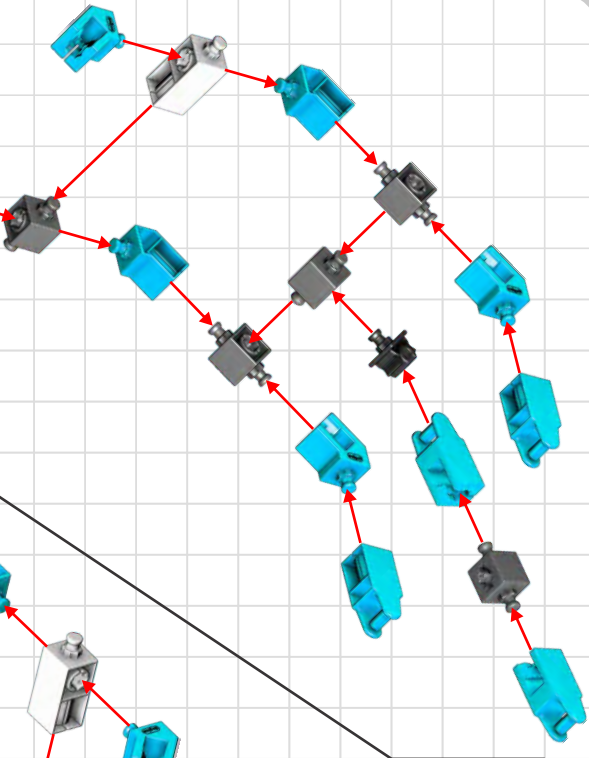
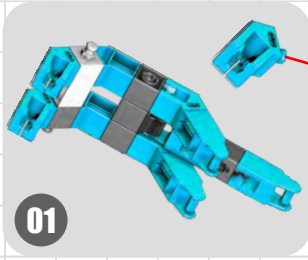
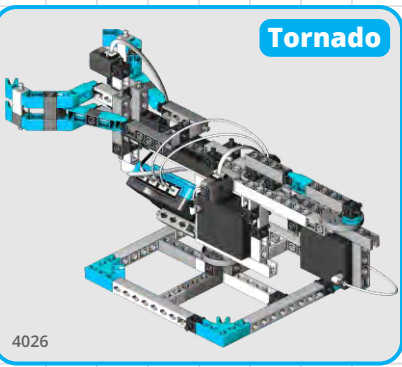
# Tornado

4026

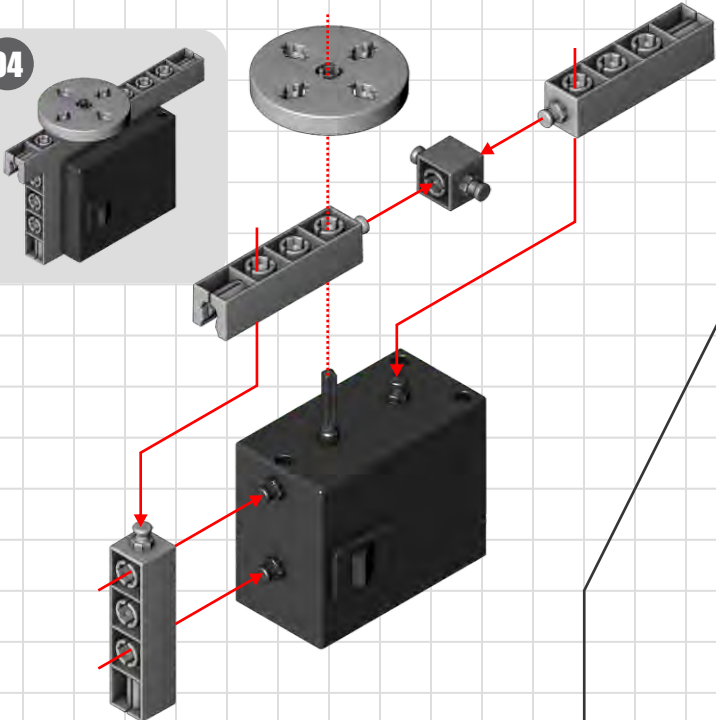
01

02

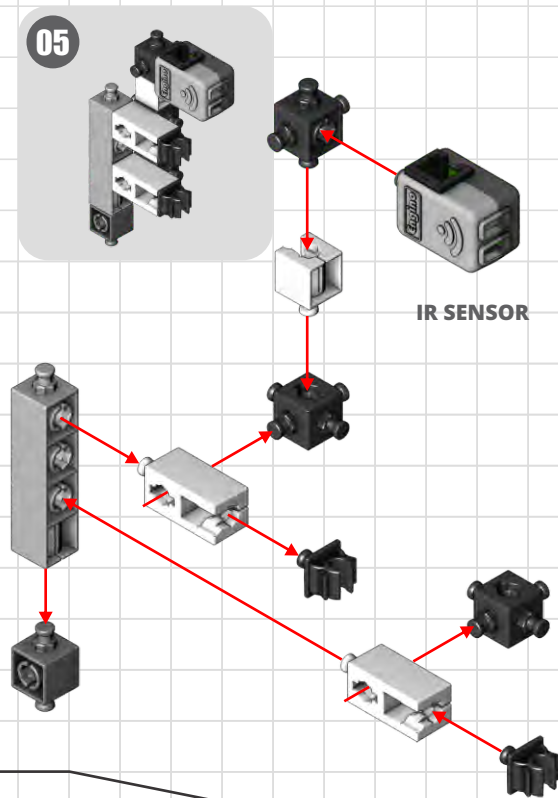
03



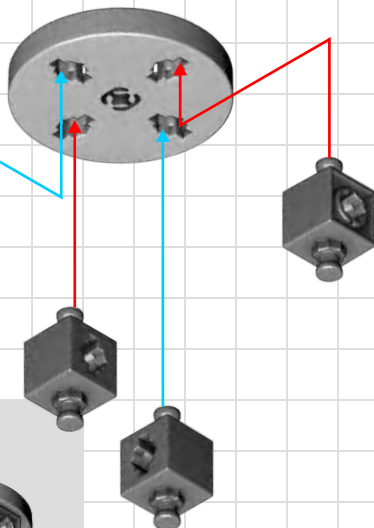
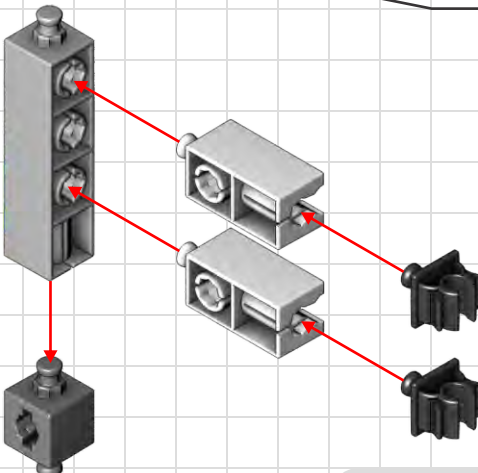
04



05



IR SENSOR



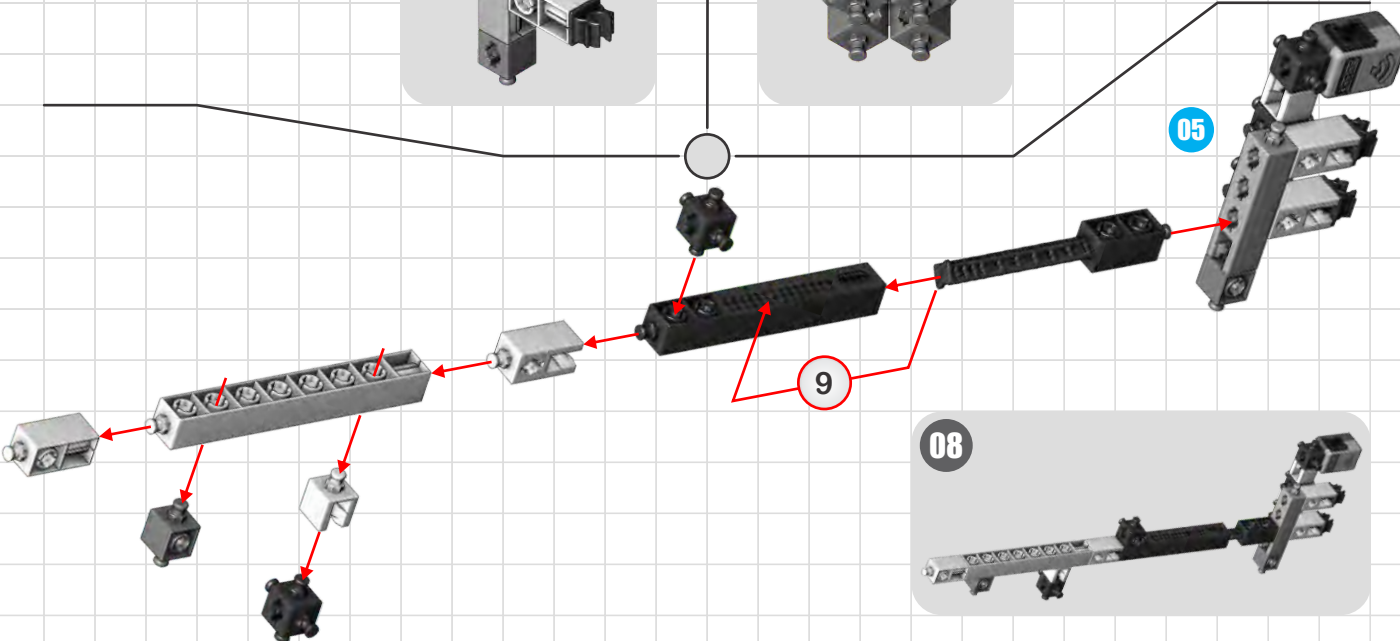
06



07



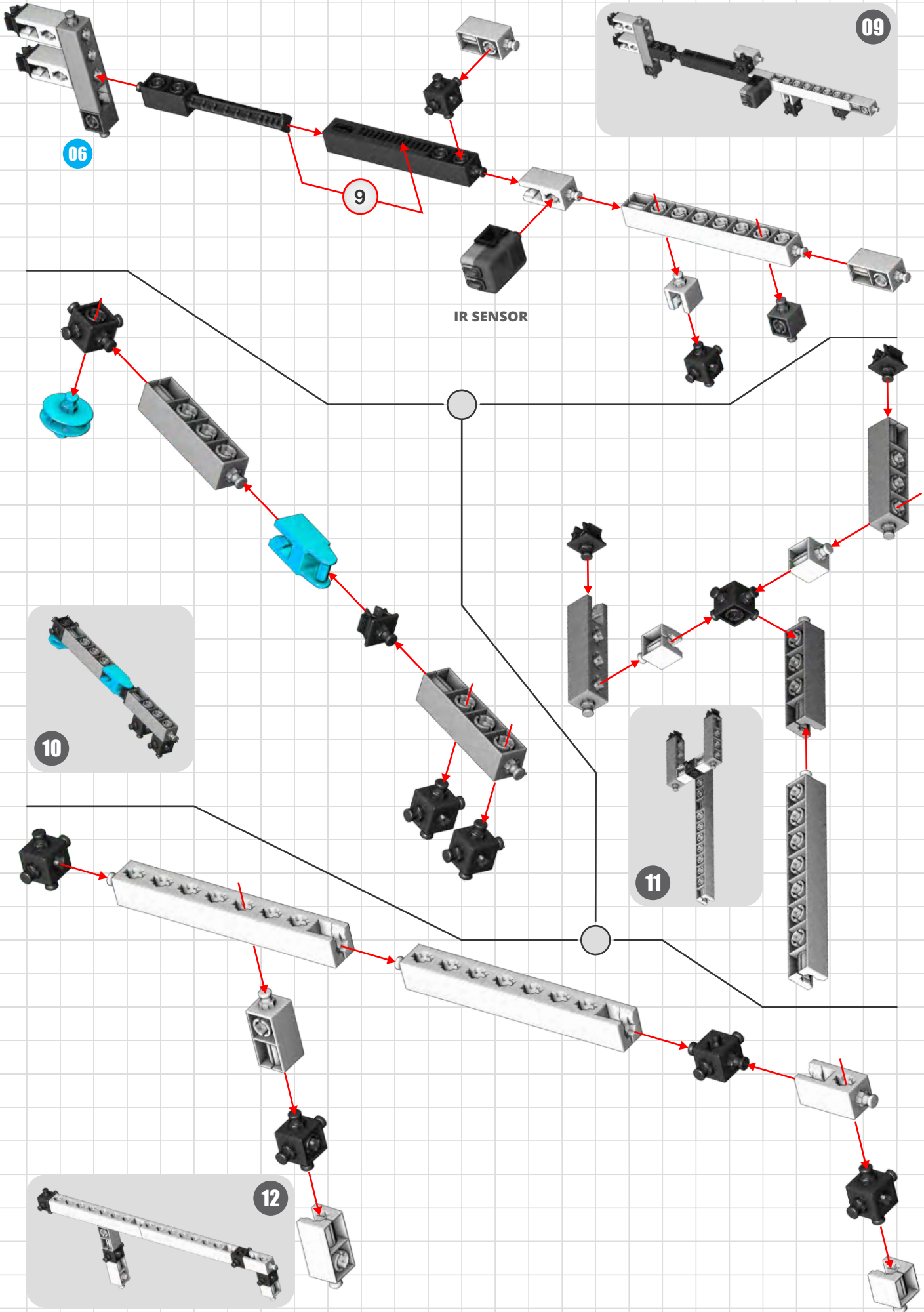
05



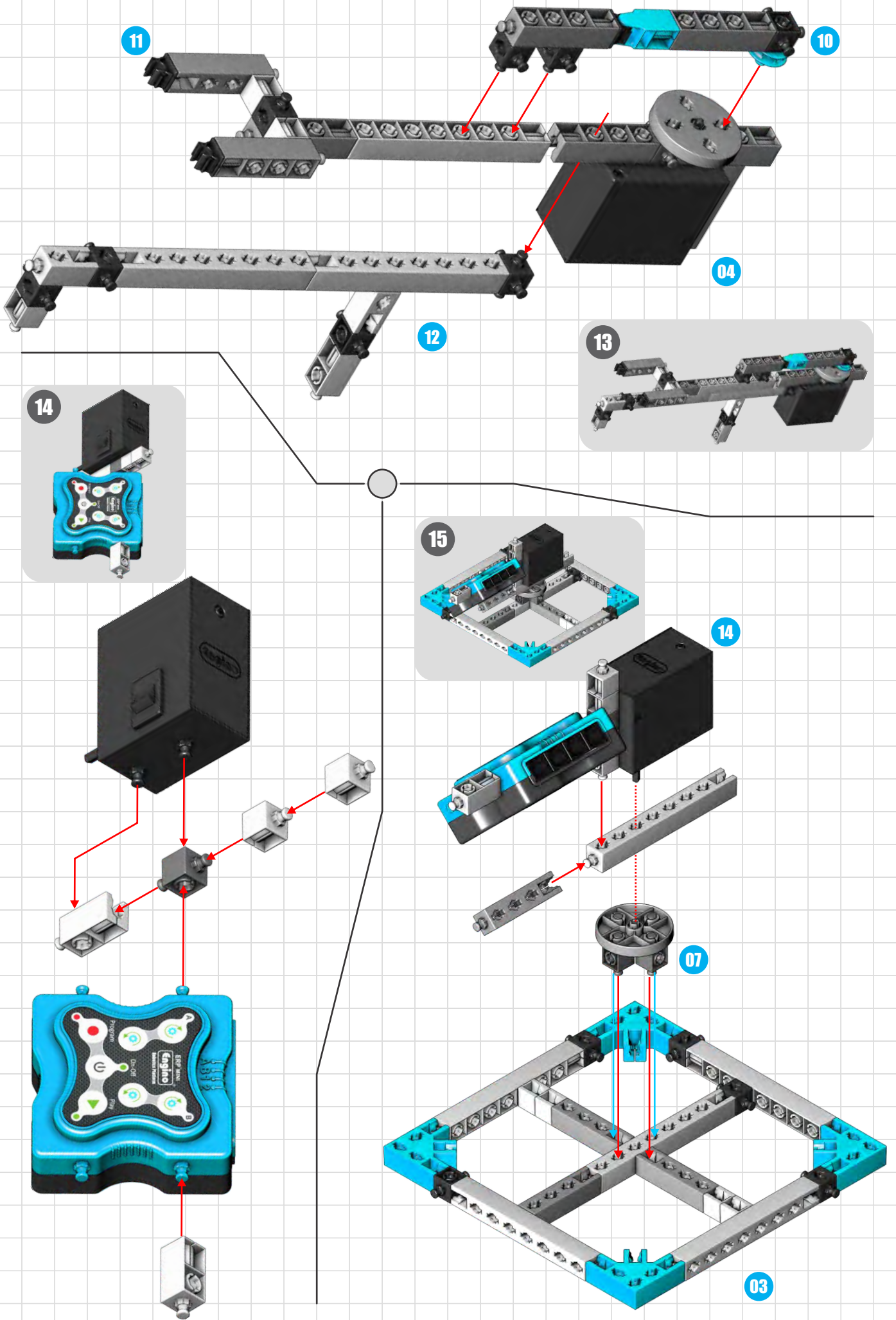
9

08

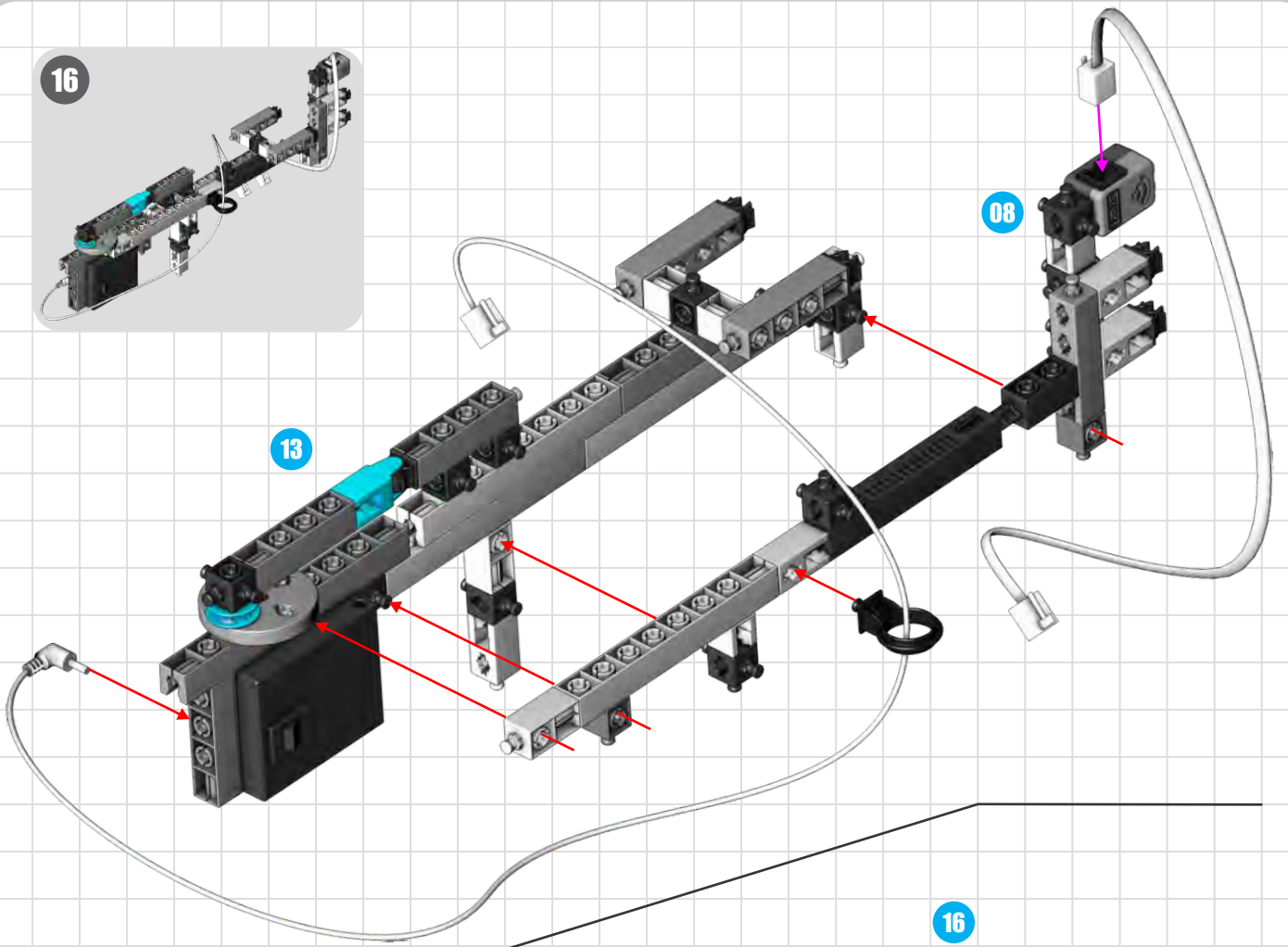
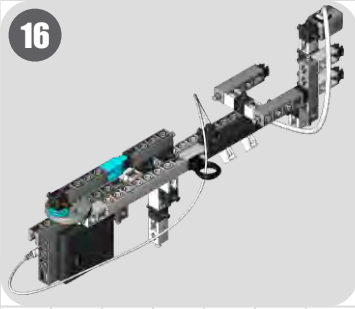








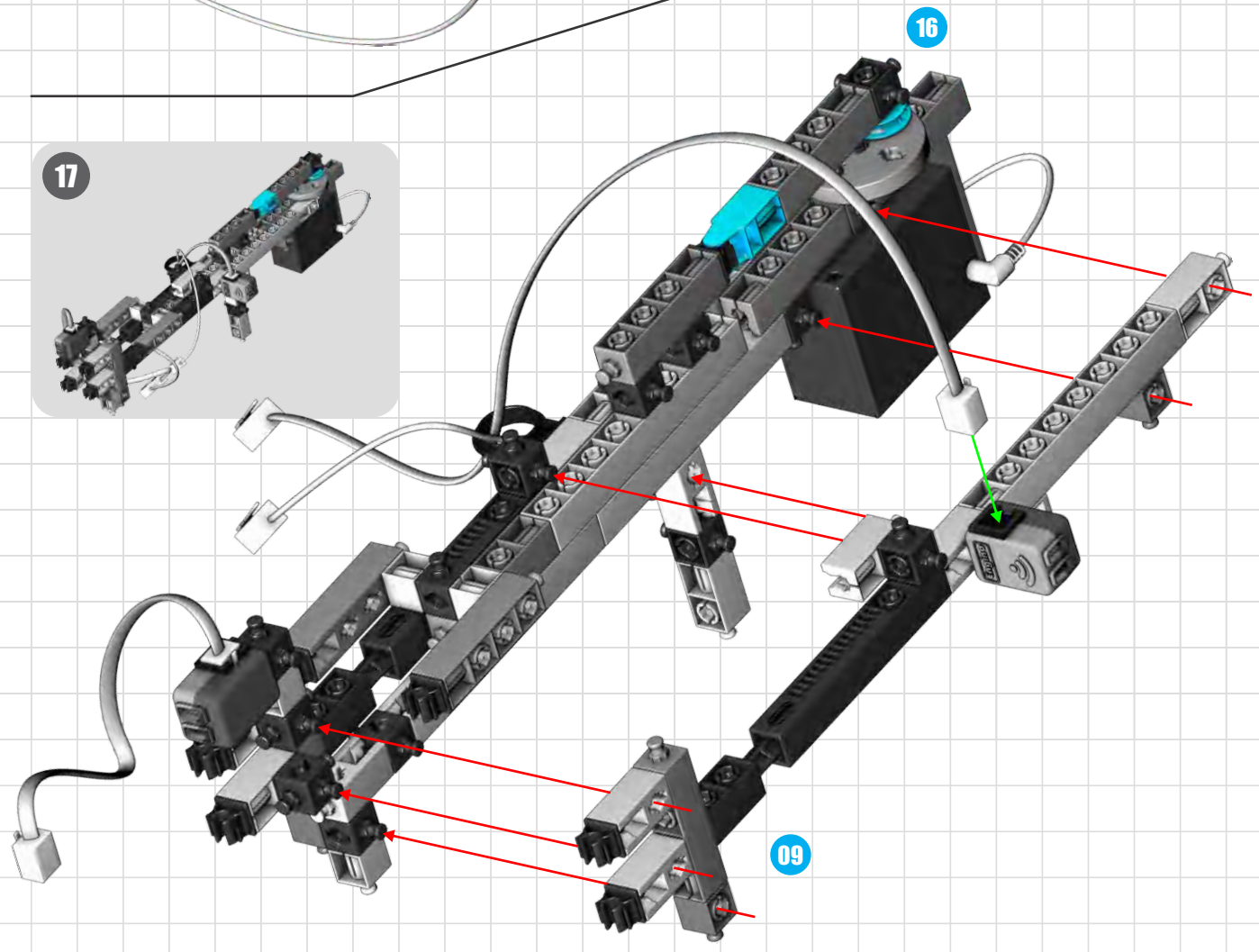
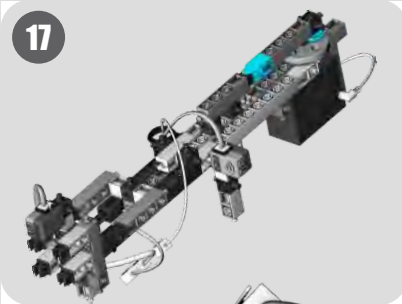
16



13

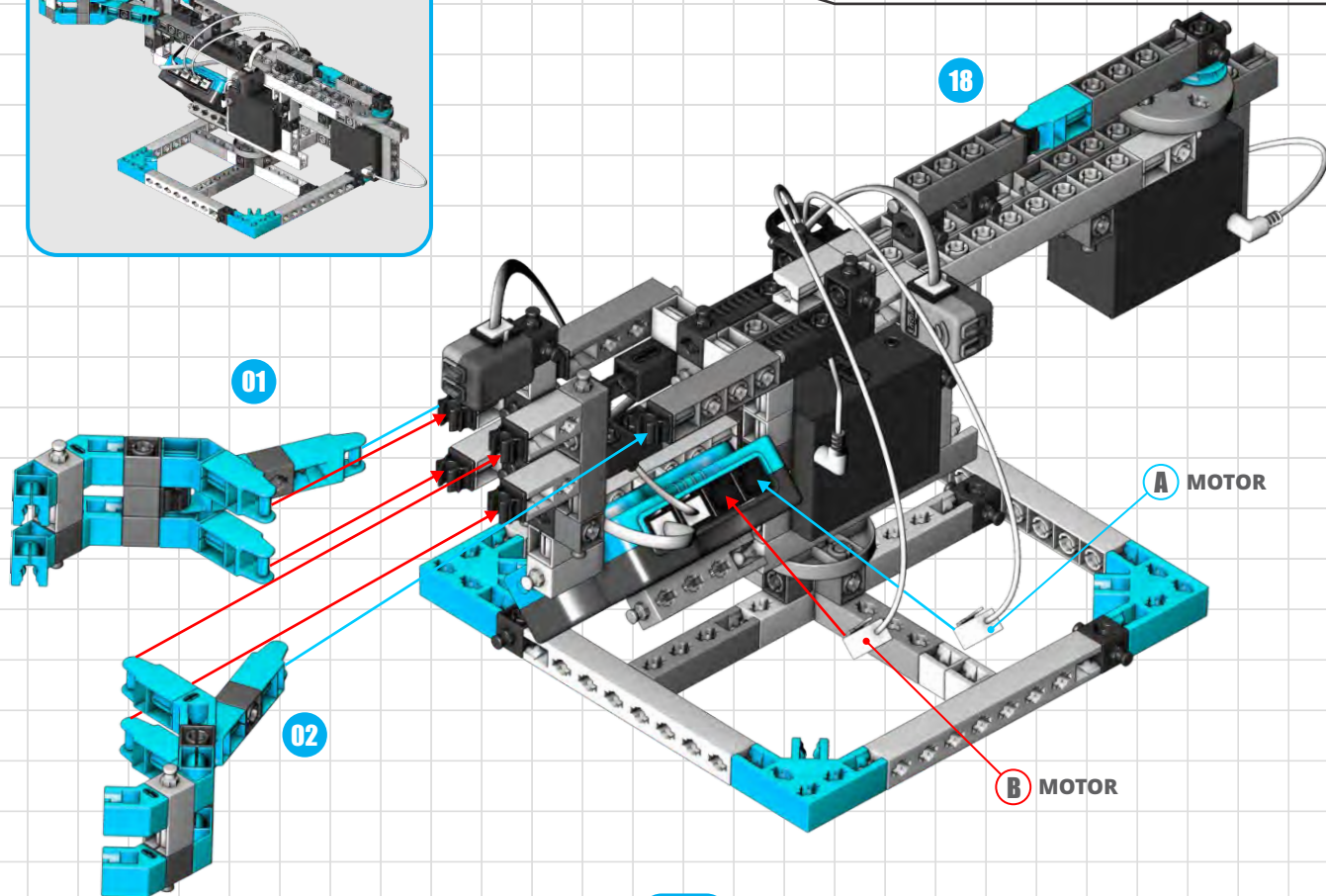
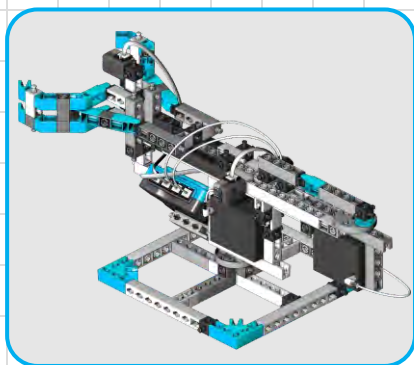
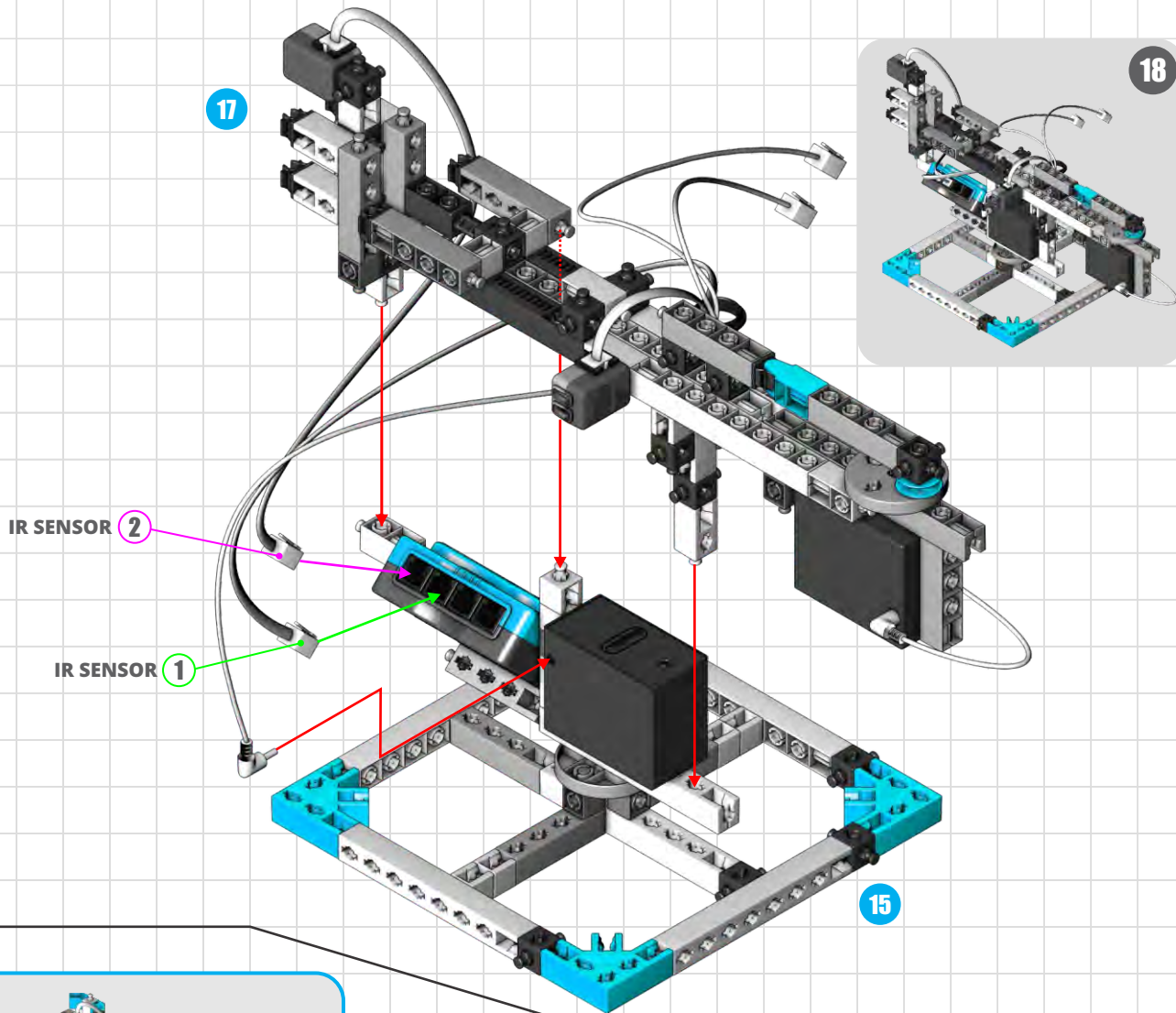
08

17



16

09

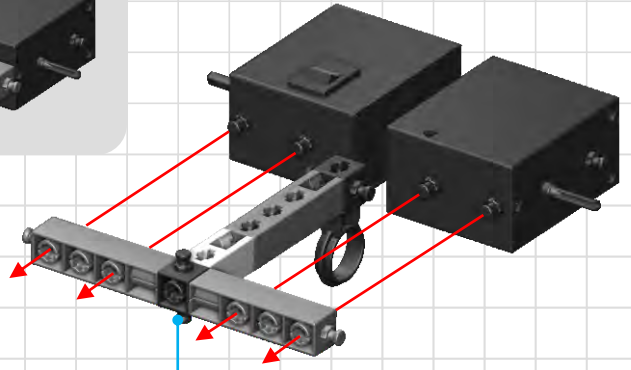




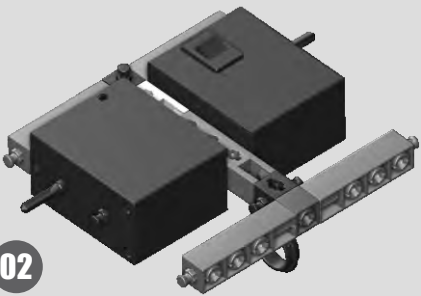
Skorpion

4025

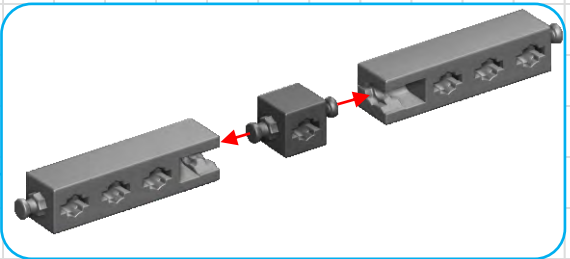
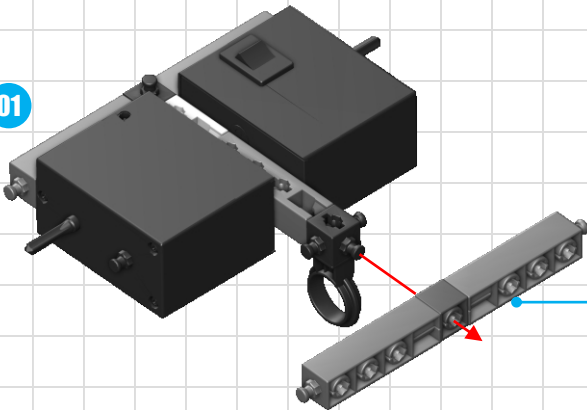
01



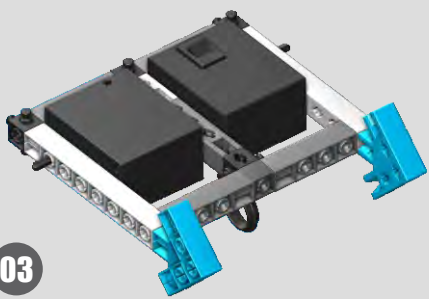
02



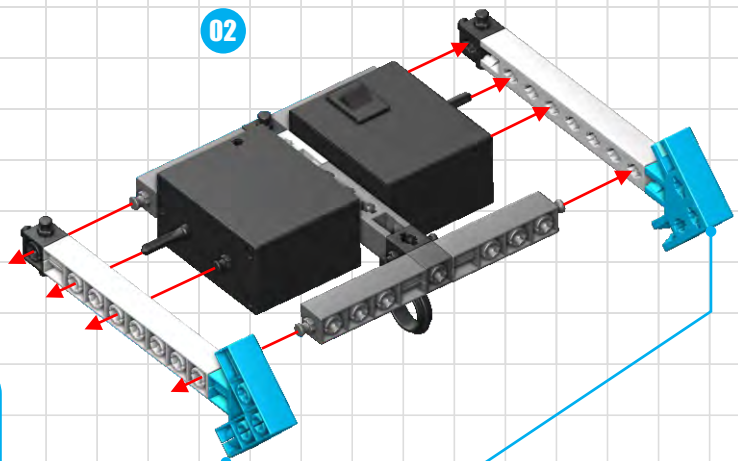
01



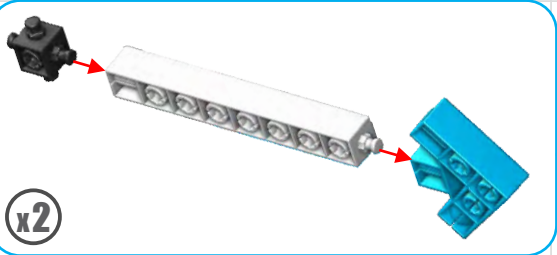
03



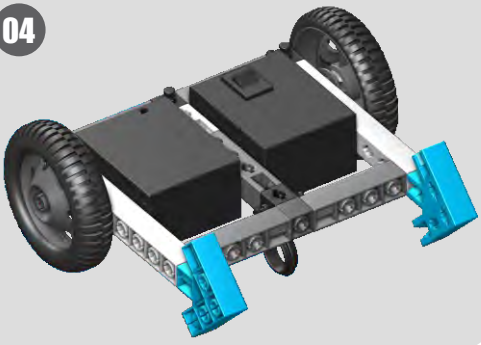
02



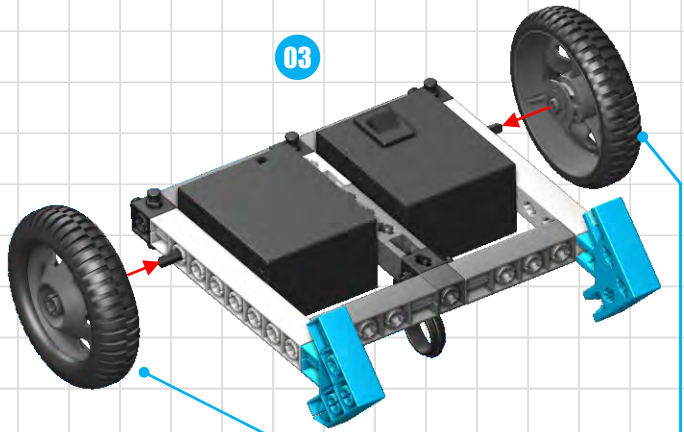
x2



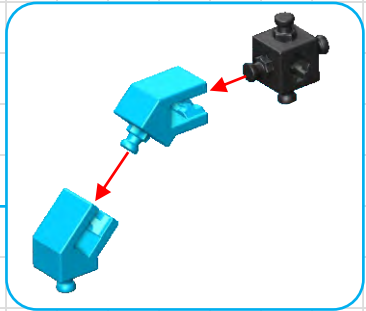
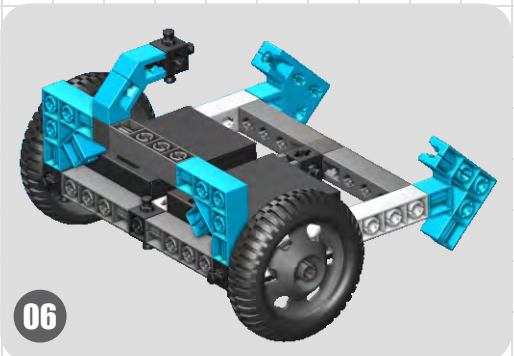
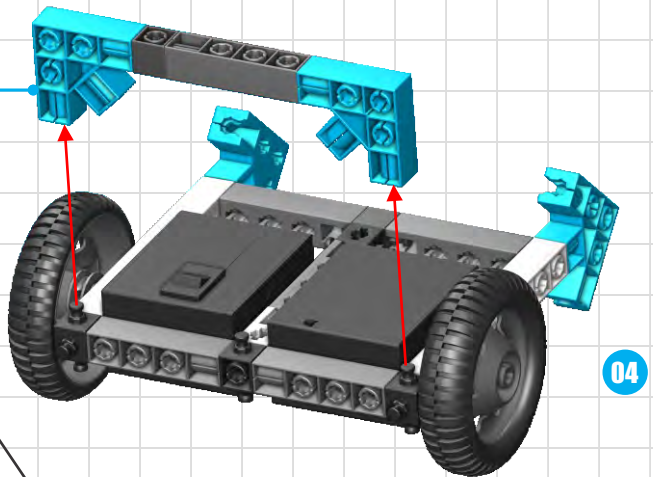
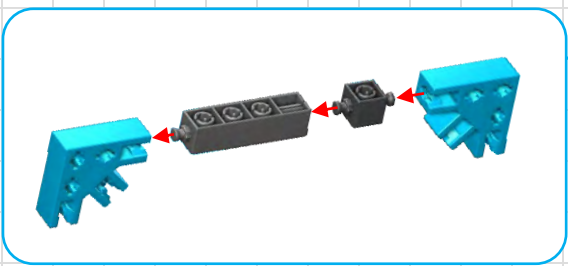
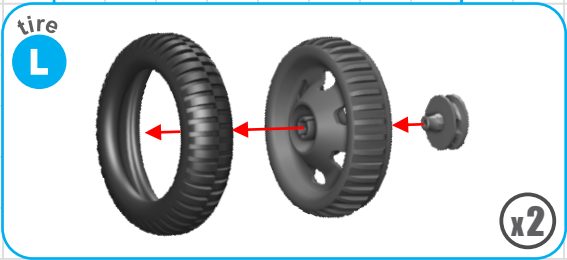
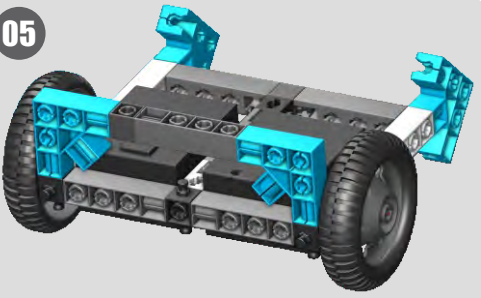
04



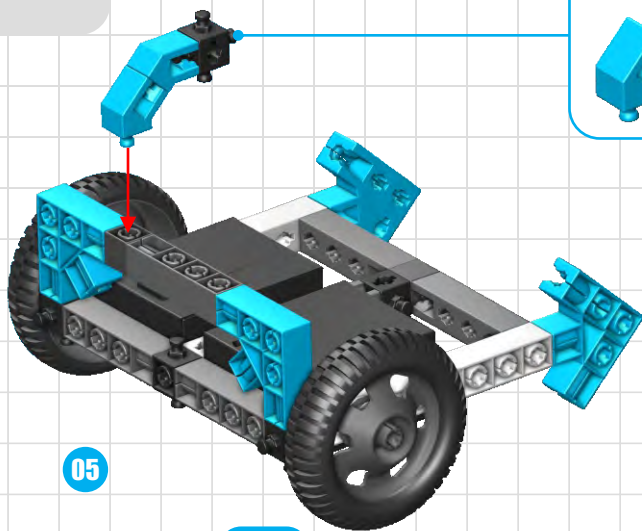
03



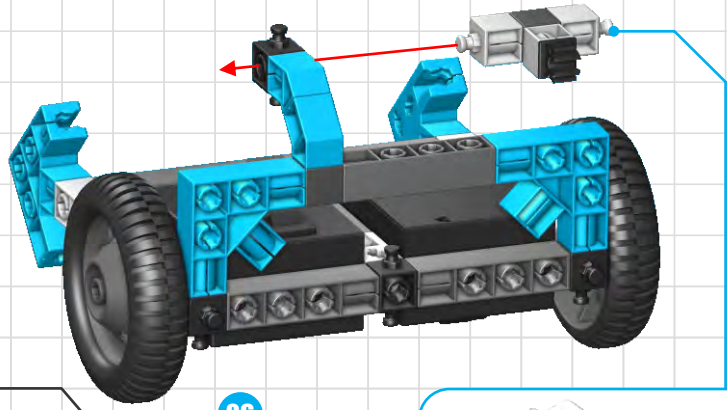
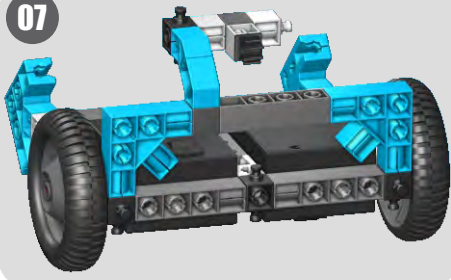
05



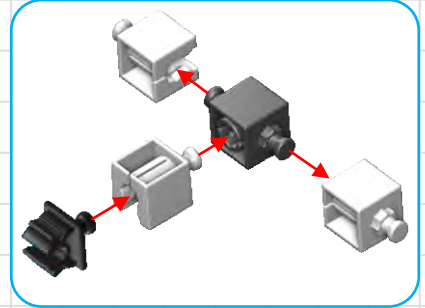
05



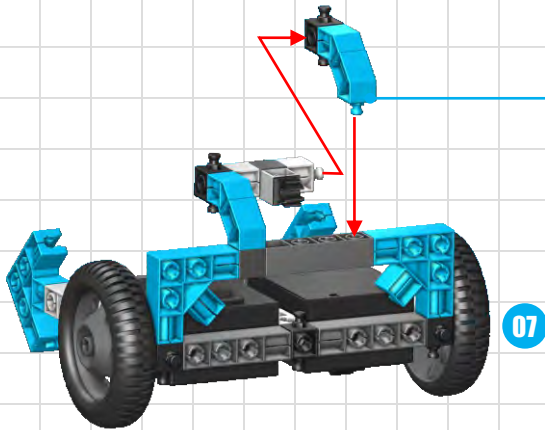
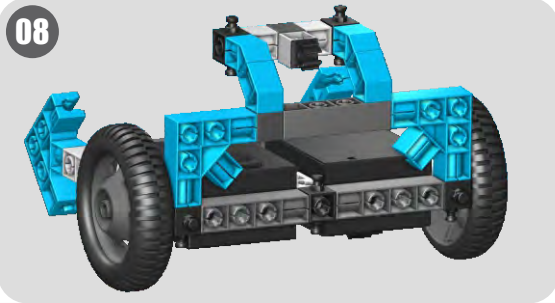
07



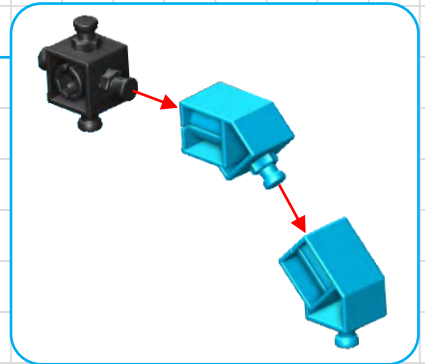
06



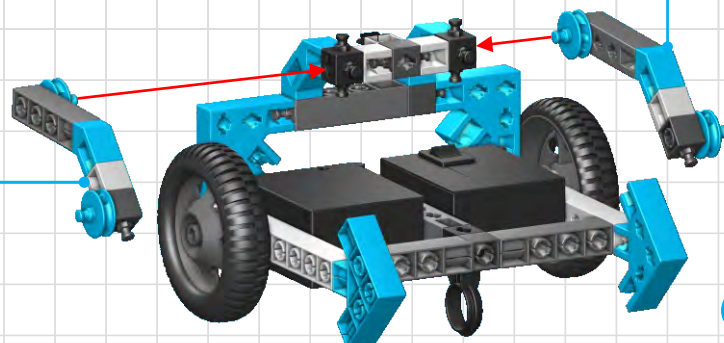
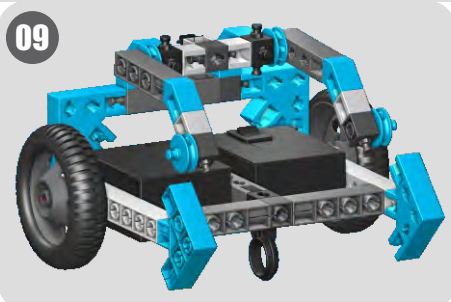
08



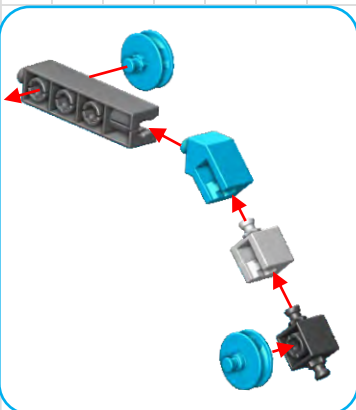
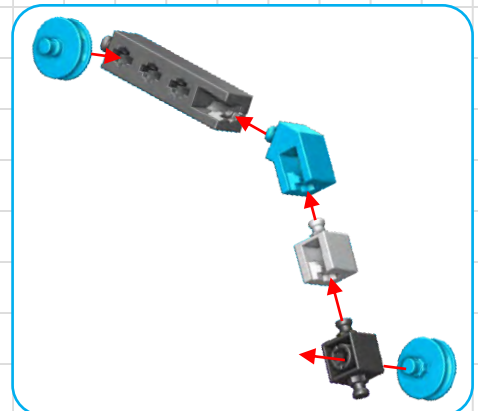
07



09

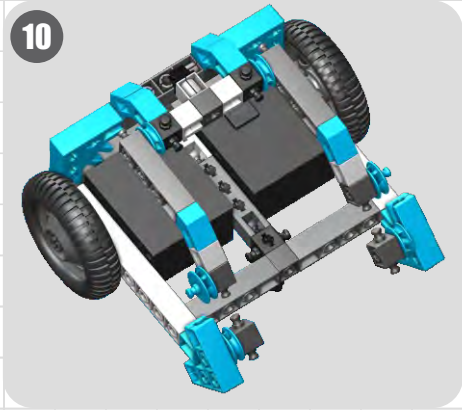


08

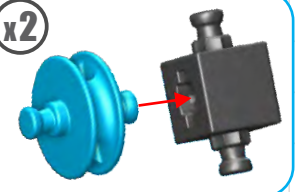




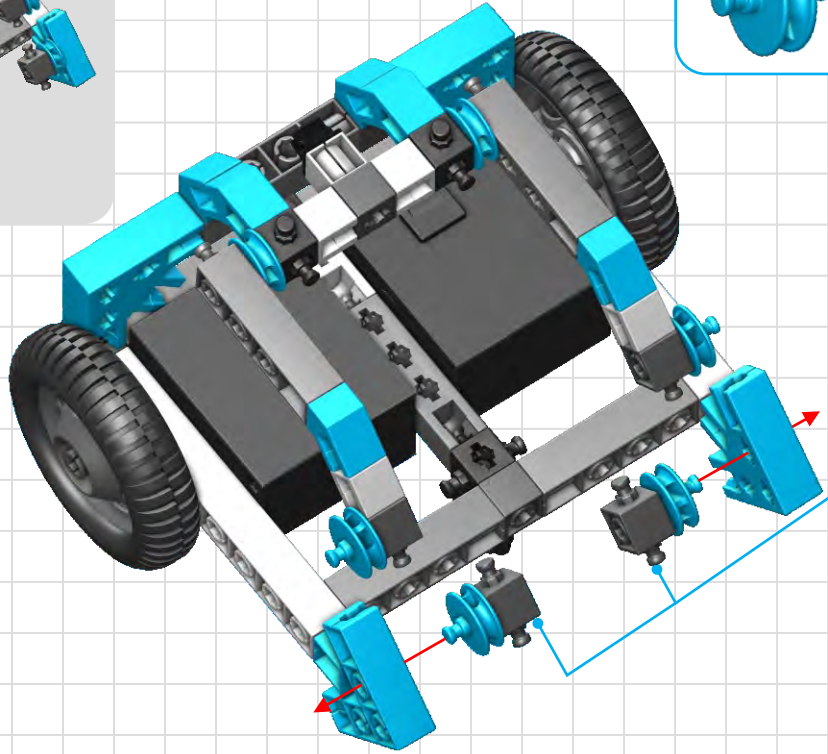
10



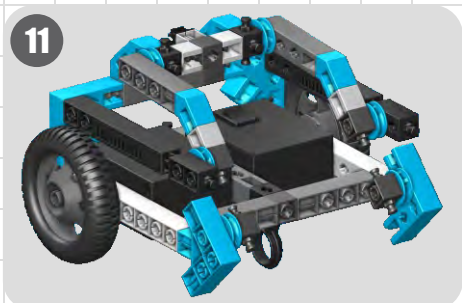
x2



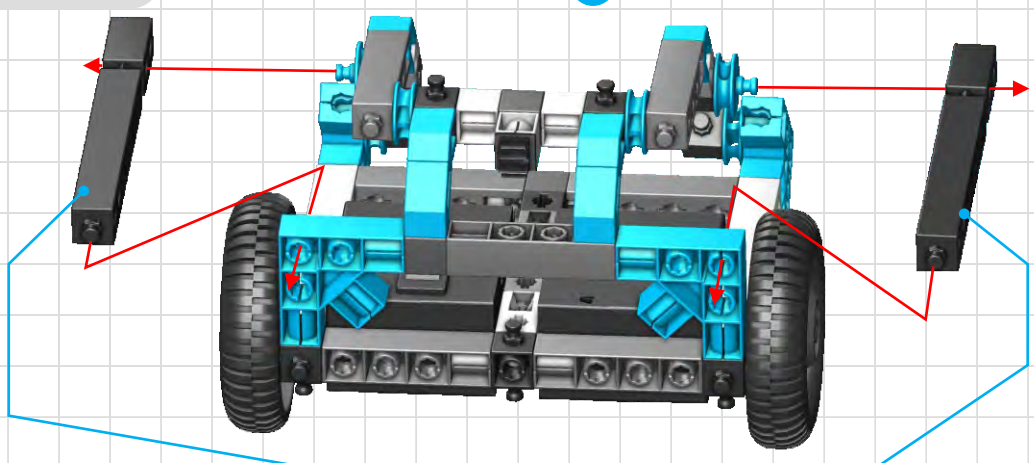
09



11

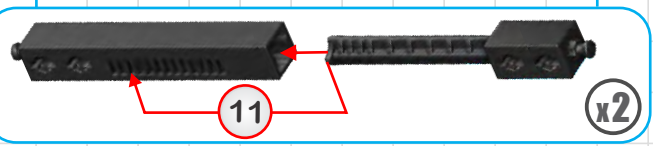


10

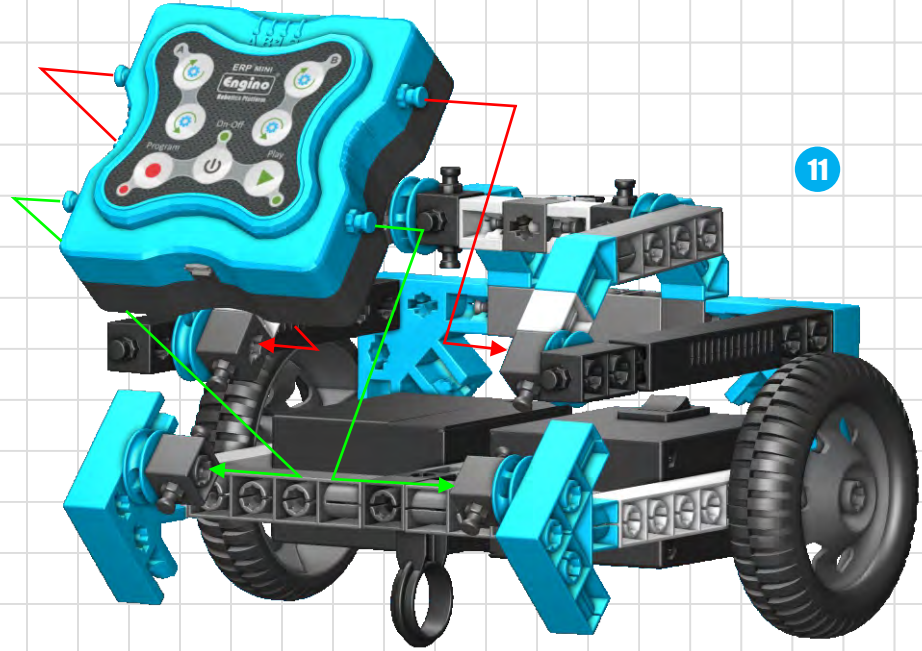


11

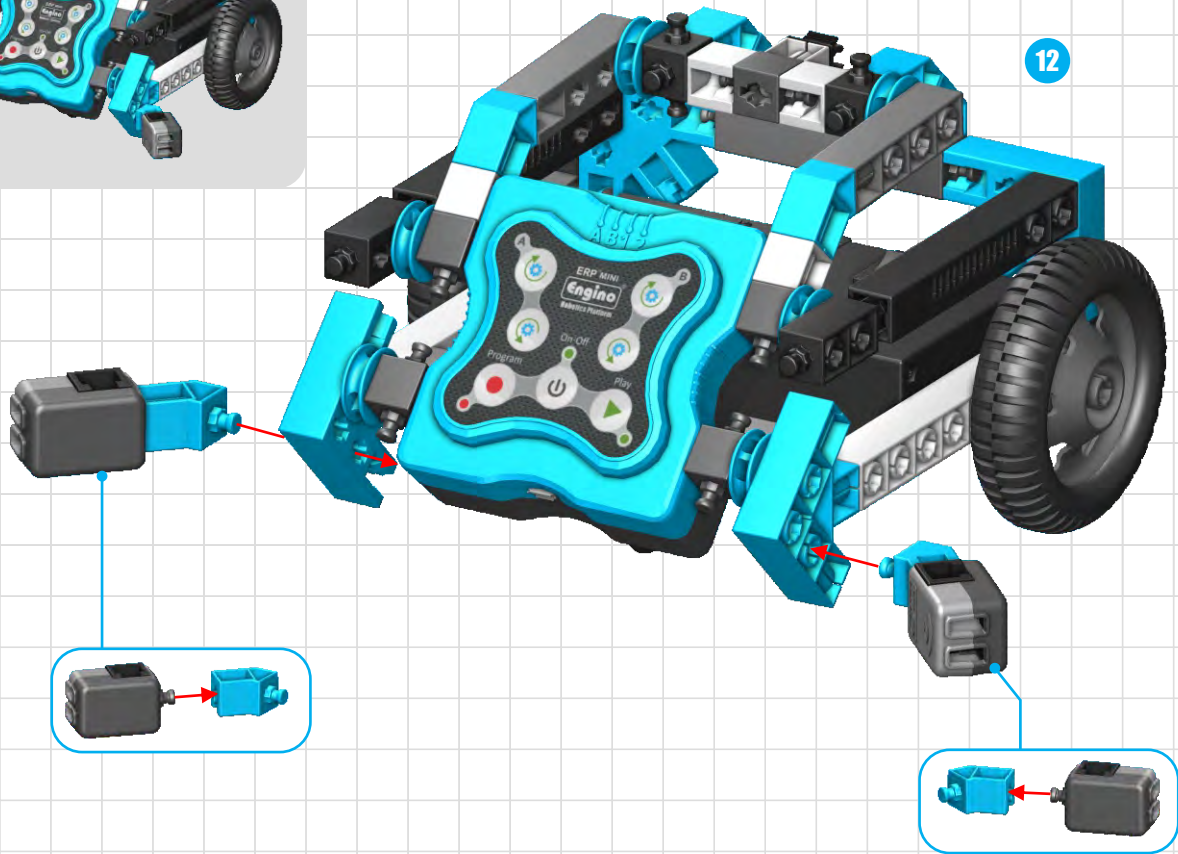
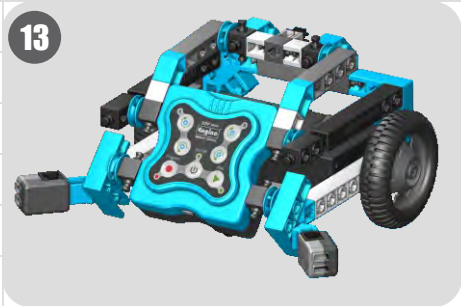
x2



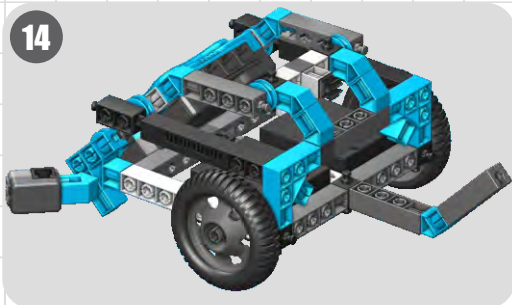
12



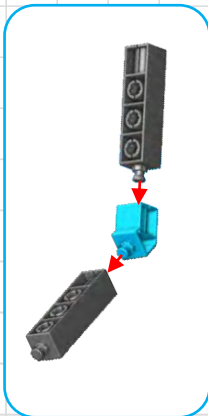
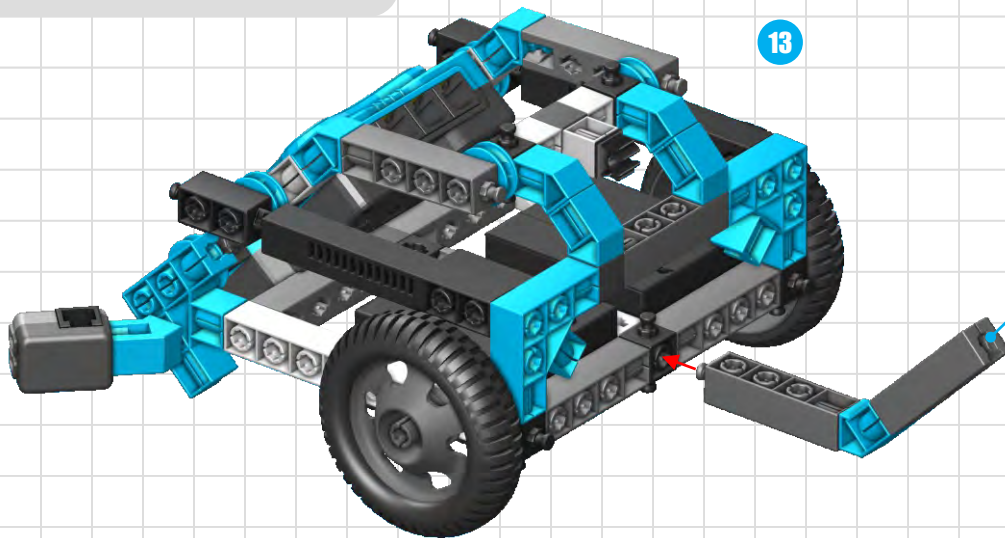
13



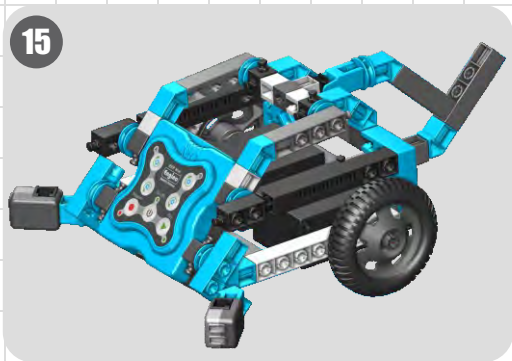
14



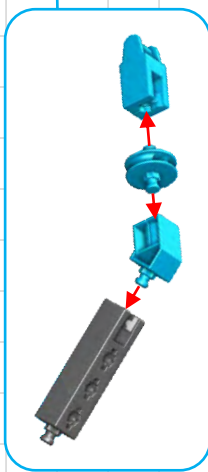
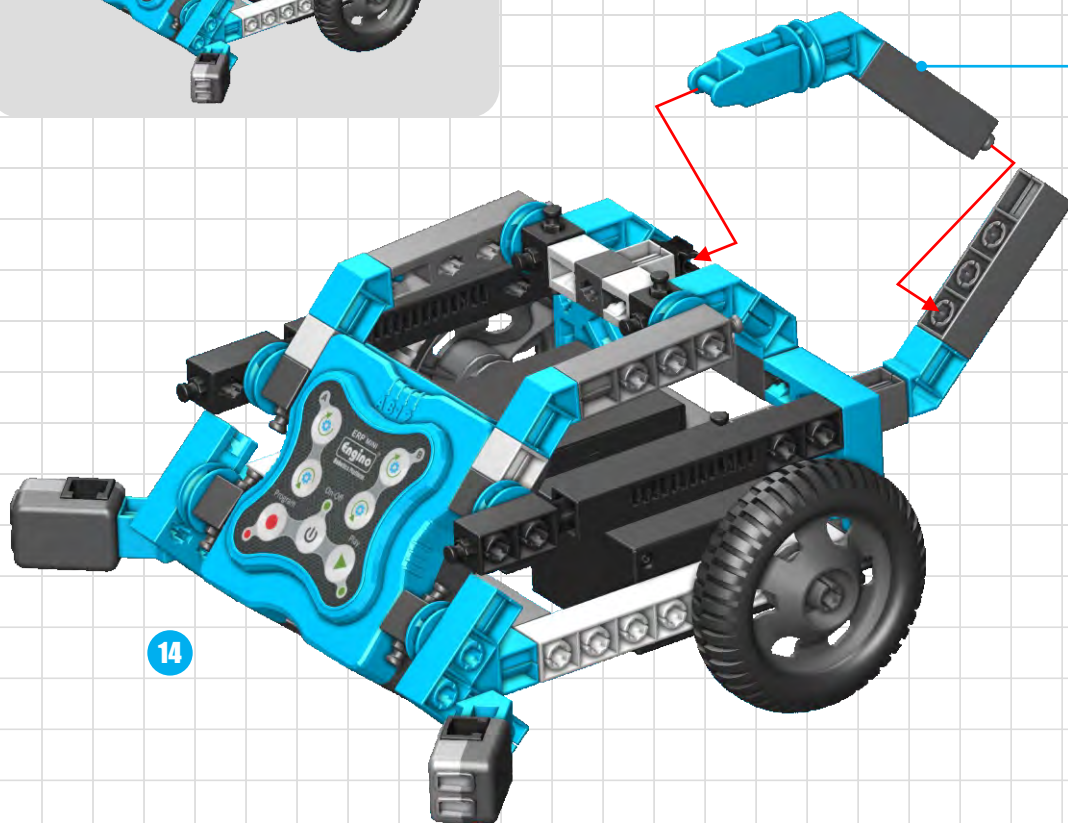
13



15

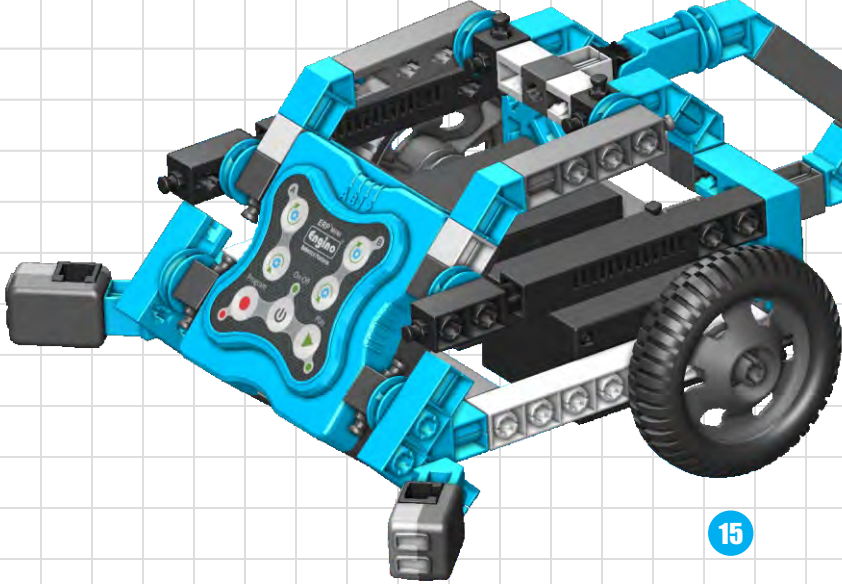


14

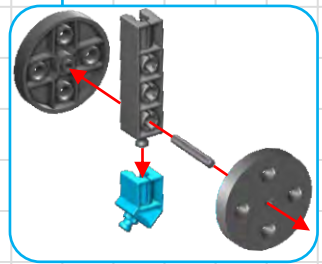




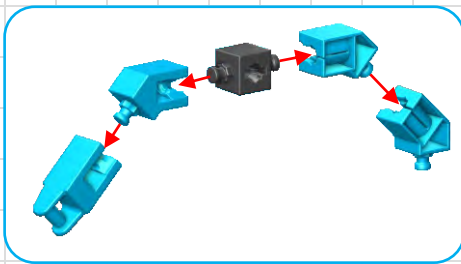
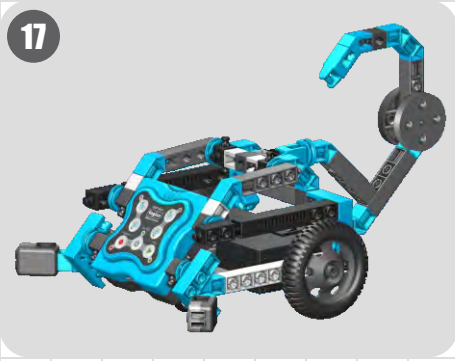
16



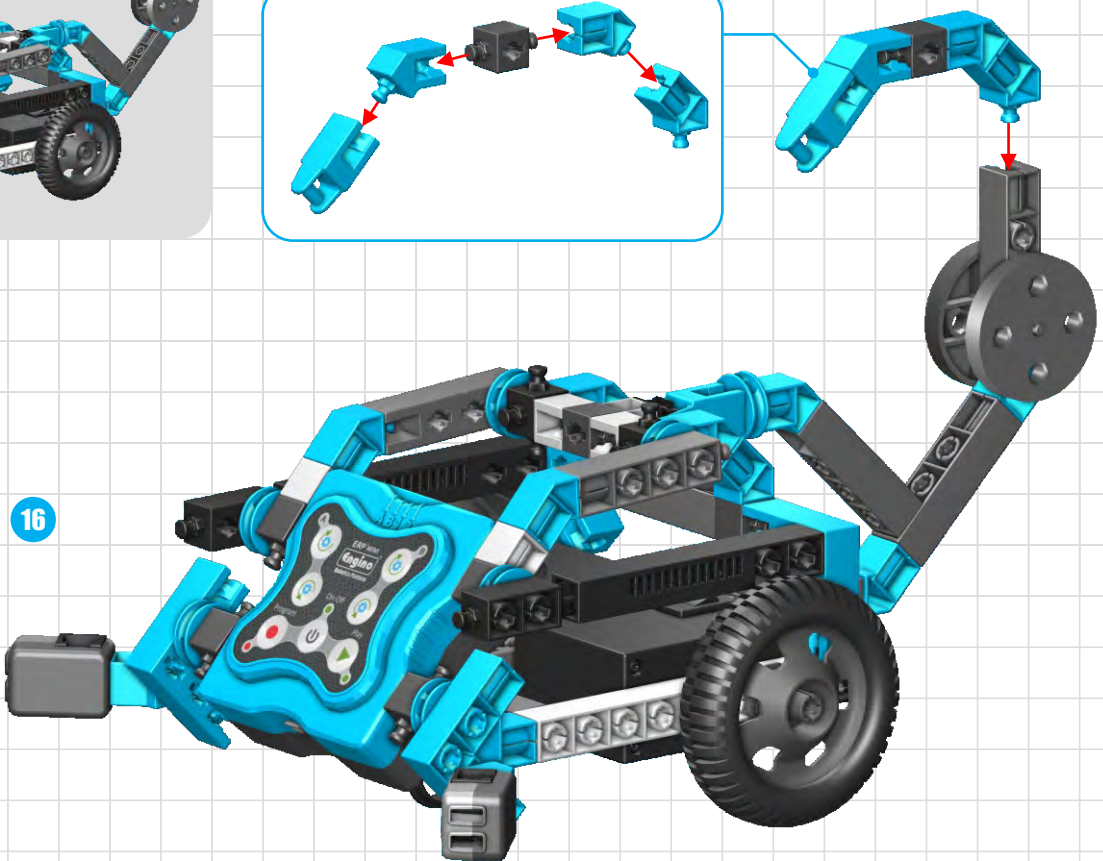
15



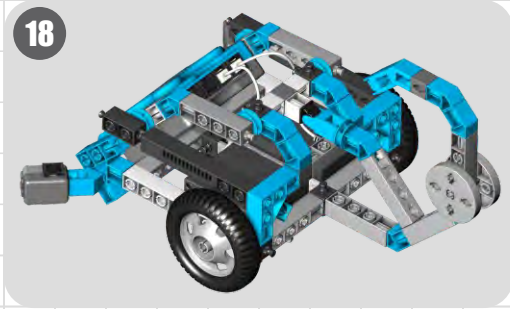
17



16

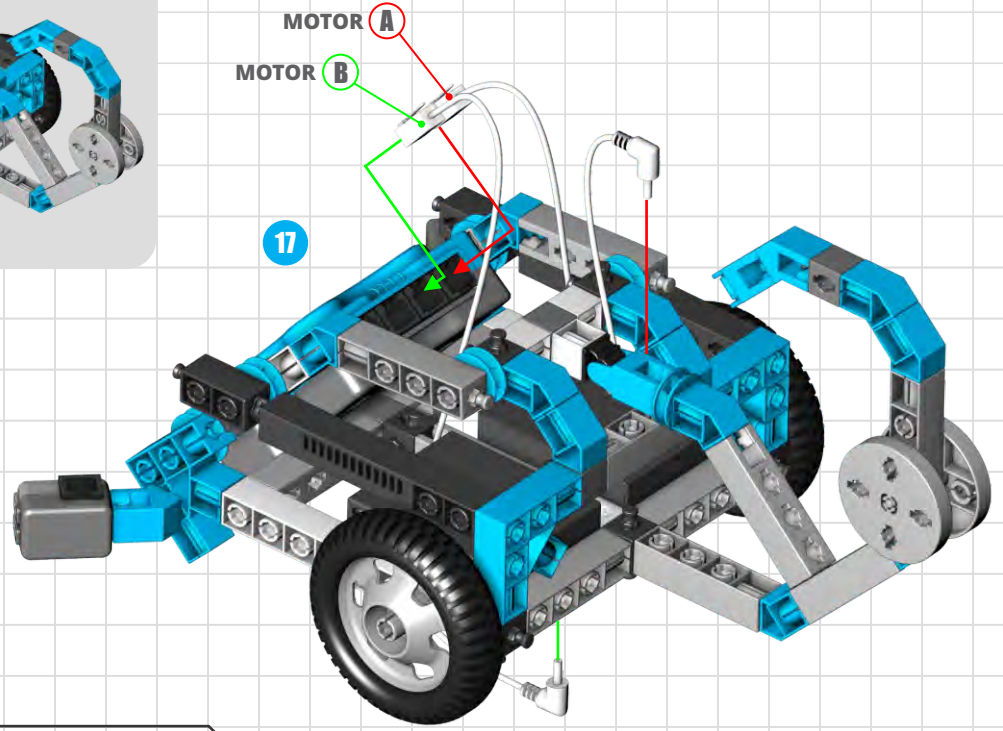


18



MOTOR A  
MOTOR B

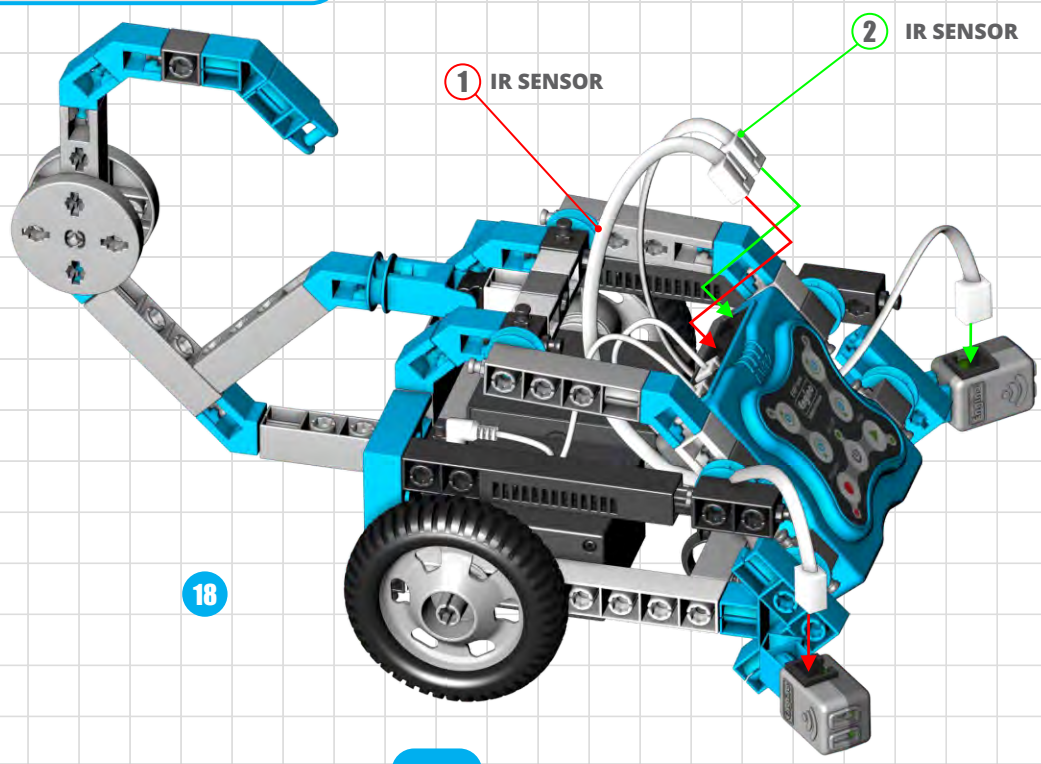
17



2 IR SENSOR

1 IR SENSOR

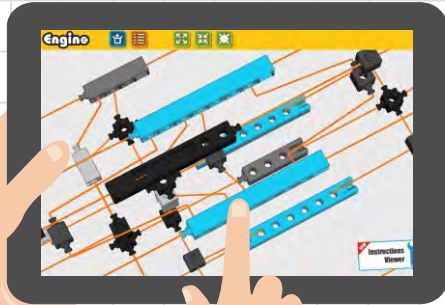
18





## Darmowa aplikacja

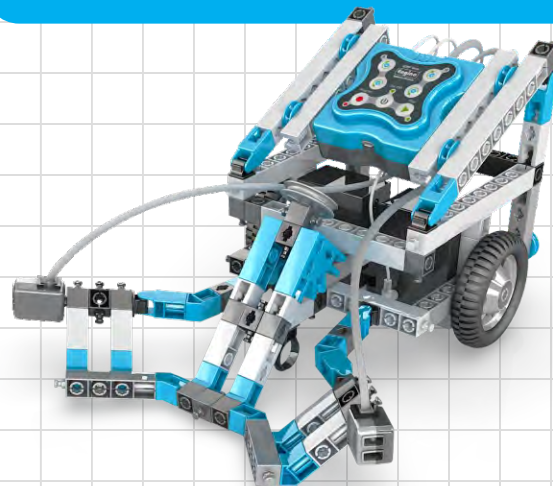
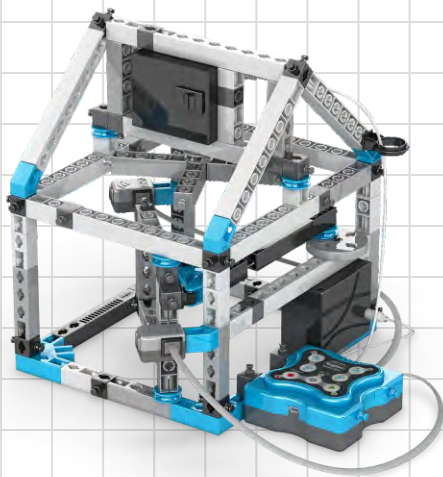
- A** Korzystając z komputera lub tabletu przejdź pod poniższy adres, aby uzyskać więcej modeli:



[www.engino.com/instructions/ce101mp-a](http://www.engino.com/instructions/ce101mp-a)

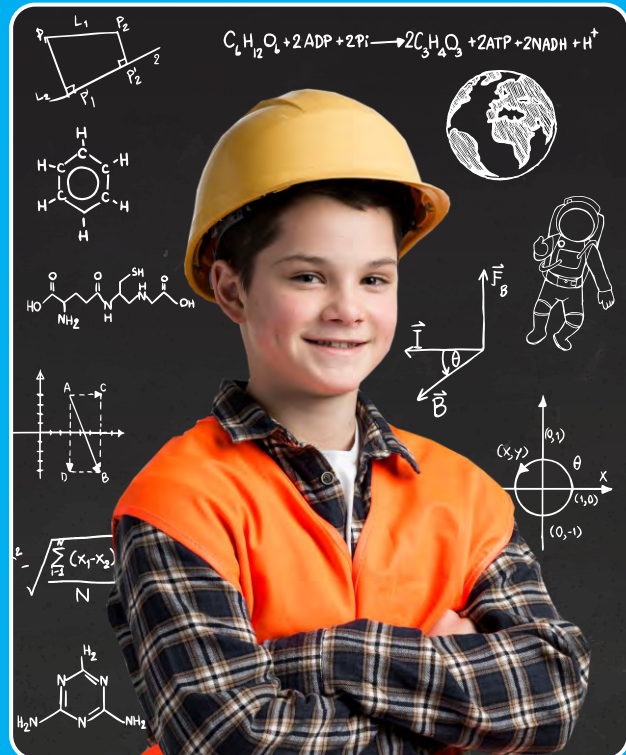
- B** Pobierz aplikację, aby zapoznać się ze szczegółowymi instrukcjami w 3D!

**Aplikacja Engino kidCAD (3D Viewer):**



## Więcej teorii i eksperymentów znajdziesz online!

[www.engino.com/theory/ce101mp-a](http://www.engino.com/theory/ce101mp-a)



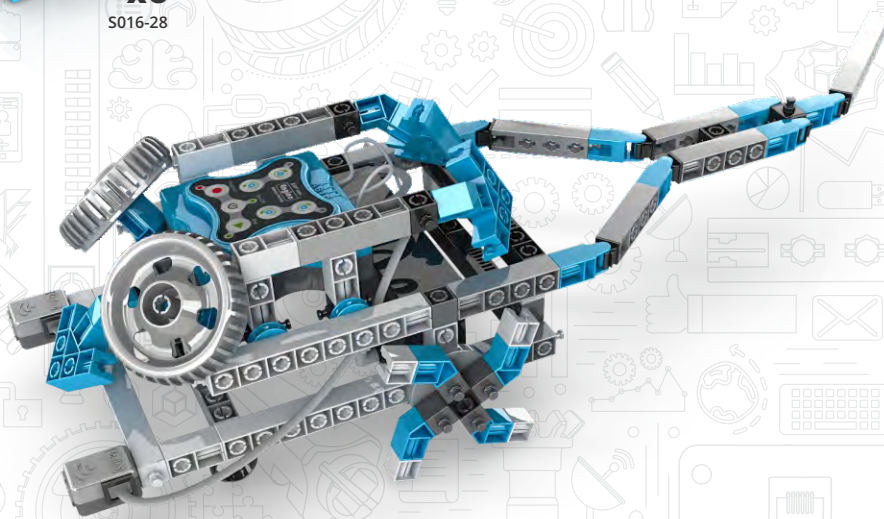
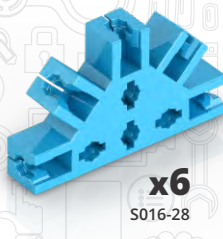
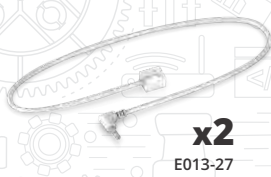
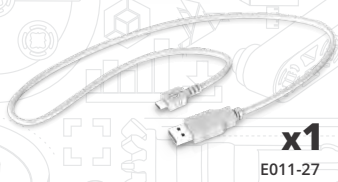
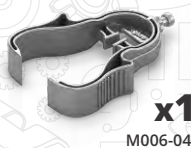
©123RF.com / baloncici, ella, James Steidl, Ales Zvolanek, Olga Serdyuk, spaxia, abidal, dipressionist, Inna Jacquemin, Miroslaw Dobrzanski, scyther5, hxdbxzy, gemenacom, pixinoo, Frederick, Joerg Hackemann, fastfun23, Ronalds Stikans.

Copyright © Engino-Net Limited

Wszelkie prawa zastrzeżone. Żadna część tych stron nie może być wykorzystywana do innych celów niż do użytku osobistego. Dlatego ich powielanie, modyfikowanie, przechowywanie w systemie wyszukiwania lub retransmisja, w jakiegokolwiek formie lub za pomocą jakichkolwiek środków elektronicznych, mechanicznych lub innych, w celach innych niż do użytku osobistego, jest bez uprzedniej pisemnej zgody wydawcy surowo zabronione.



# Lista elementów



Wyprodukowano w Europie



**Dystrybutor:**  
ConQuest entertainment a.s.  
Kolbenova 962/27d,  
Praha 9  
info@cqe.pl  
www.toy.cz  
tel: 284 000 160

**OSTRZEŻENIE:**

**NIEBEZPIECZYSTWO ZADŁAWIENIA**  
- małe części. Produkt nie jest przeznaczony dla dzieci poniżej 3 lat.



Kod produktu:  
**CE101MP-A**