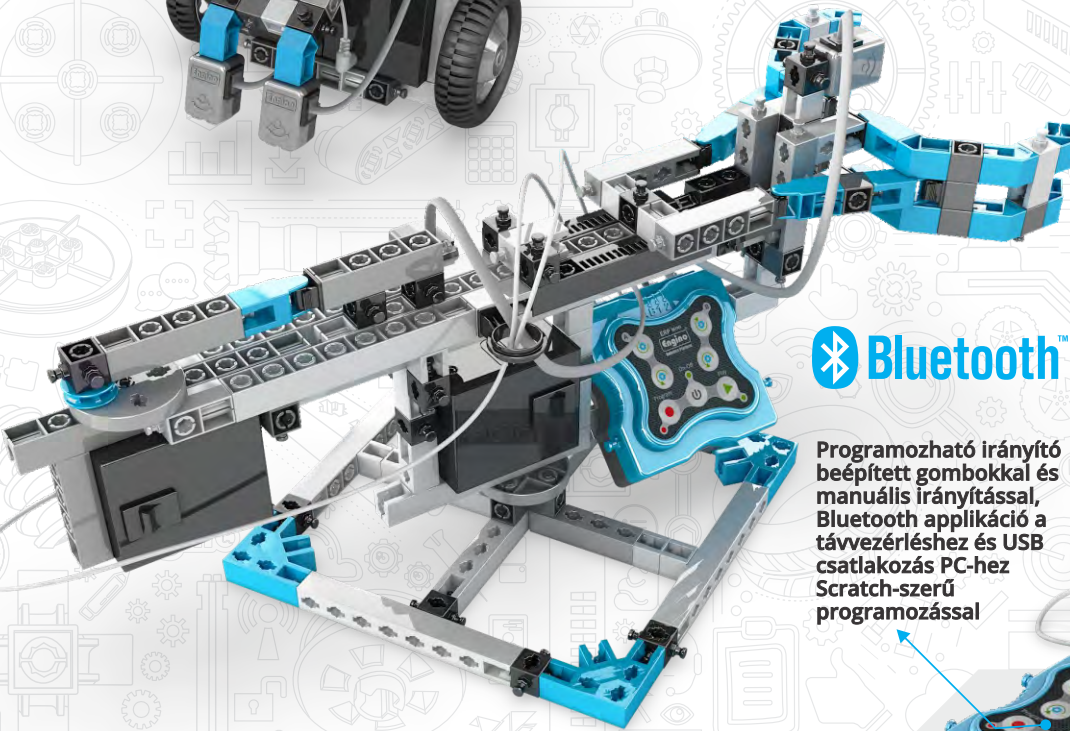
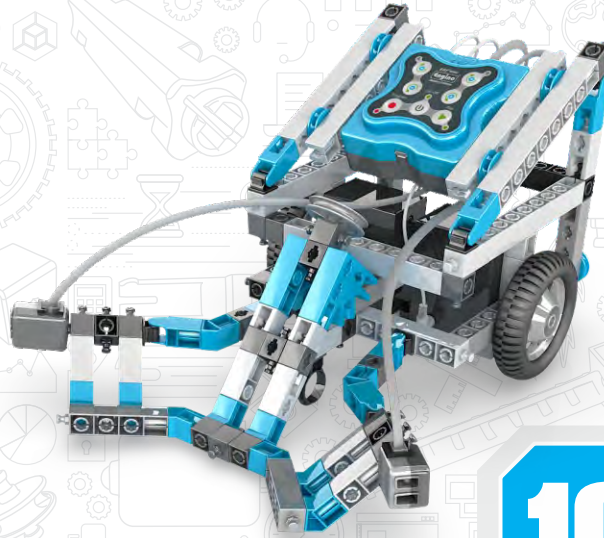


Engine

CREATIVE™ engineering maker Pro



100 IN 1 robotized MODELS

5
modell

nyomtatott
útmutató
(tartalmazza)



95
modell

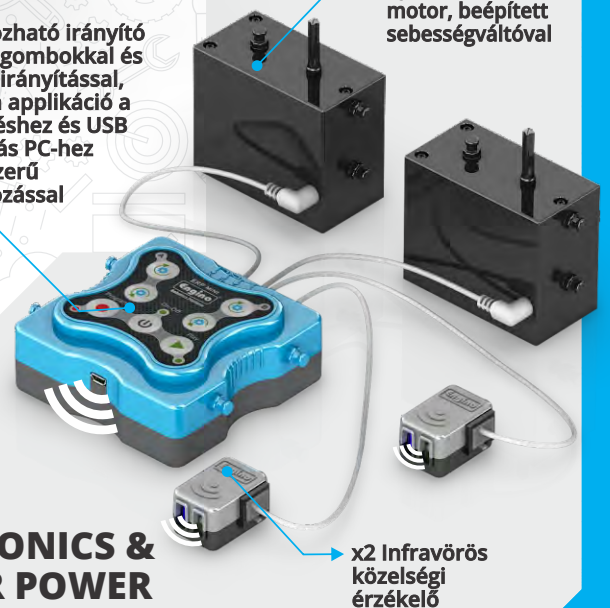
online
3D útmutató



 **Bluetooth™**

Programozható irányító
beépített gombokkal és
manuális irányítással,
Bluetooth applikáció a
távvezérléshez és USB
csatlakozás PC-hez
Scratch-szerű
programozással

x2 Nagy
nyomatékú
motor, beépített
sebességváltóval



+ **ELECTRONICS &
MOTOR POWER**

x2 Infravörös
közelségi
érezkélő

• INTERAKTÍV 3D ÚTMUTATÓ

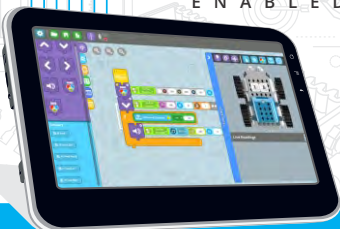
• ÉPÍTENI

• JÁTSZANI

• TANULNI

• FELTALÁLNI

STEM
ENABLED



10+

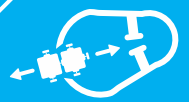
Ajánlott
Életkor



Építési
útmutató

3

Éves Garancia



Szétcsúszó eszköz
Az alkatrészek könnyű
szétválasztásához

CREATIVE™ engineering

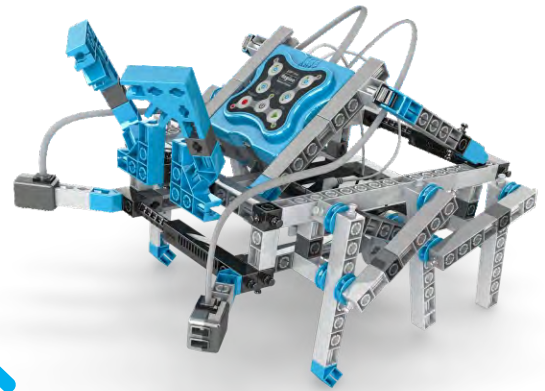
maker Pro



Építs ventilátort!

Ez a ventilátor modell bemutatja neked a programozás alapjait és az innovatív MINI 2.0 irányítót. Ezen kívül segít tesztelni a különféle változókat és programokat, amik a KEIRO szoftverrel készültek.

- Milyen bemenetek és milyen kimenetek vannak?
- Hogyan lehet manuálisan programozni egy robotot?



Építs hatlábút!

Kísérletezz a lábak mozgatásával kerekek helyett, ezt a hatlábú modellt használva! Állítsd az infravörös érzékelőt bármelyik oldalra, hogy jelezze, ha egy akadály közeledik, így a robot kikerülheti azt a lábait használva.

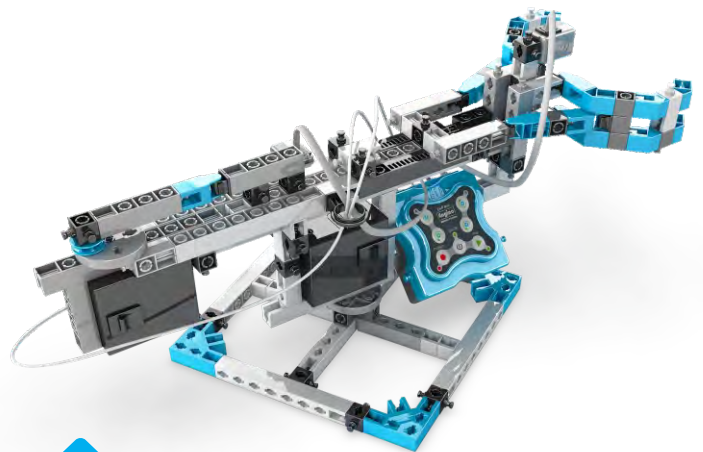
- Hogyan működik a lábak mozgatása
- Mi az az "IF" utasítás?



Építs egy robotot!

Építsd meg a robot modellt és használd ki a dupla képességét! A programozható érzékelőket használva a robot különbséget tud tenni a fekete és fehér színek között, hogy kövessen egy vonalat vagy észrevegye, ha az útjában van valami.

- Hogy működnek az infravörös érzékelők?
- Mik azok a logikai kapuk és az "WHILE" ciklusok?



Építs egy forgókart!

Ez a lenyűgöző forgókar robot egy valódi robotkart utánoz, amiket a gyakran használnak ipari környezetben. Programozd a modellt, hogy pontosan működjön és mozgasson tárgyakat maga körül az infravörös érzékelő technológiát használva.

- Hogyan kell dolgozni helyhez kötött robotokkal?
- Mi az az "REPEAT UNTIL" utasítás?

Tartalom



Teória

- 03** Miről fogunk tanulni
- 03** A Robotika történelme
- 05** A robot meghatározása
- 07** Alkalmazások a valós életben
- 09** Programozás



Használati utasítás

- 11** Engino Robotika filozófia
- 13** MINI 2.0 kontroller
- 14** Perifériák, kábelek és portok
- 15** Manuális programozási példák
- 16** KEIRO szoftver telepítése
- 16** MINI 2.0 kontroller firmware frissítése
- 16** A KEIRO felület
- 30** Folyamatábra programozás
- 31** Kód példák



Kísérletek

- 33** Építés és programozás
- 34** Robotikus mozgás és IF
- 35** WHILE és logikai kapuk
- 36** Valós példák: ipari robotkar



Építési útmutató

- 37** Építési tippek
- 38** Robot
- 43** Ventilátor
- 46** Hatlábú
- 52** Forgókar
- 58** Skorpió



Mit fogunk tanulni

"A robotika lényegében rólunk szól. A saját életünk elve alapján, mert tudni szeretnénk, hogyan működünk". Ez az idézet - ami Rod Grupentől, a Számítástechnika professzorától származik - írja le a legjobban, hogy a tudósok hogyan vélekednek a robotikáról: Az életfunkciók szimulációja mechanika és számítógépek segítségével. De képes lesz az emberiség valaha olyan robotokat készíteni, akiknek érzelmei vannak és képesek döntéseket meghozni olyan szituációkban, amit nem képes megjósolni a programjuk? Bármit is tartogat a jövő, a mai fiataloknak képesnek kell lenniük tisztán látni mik a robotok, a robotika és programozási tudomány alapjaitól kezdve egészen míg saját teljesen működőképes készülékeiket meg nem tudják építeni! A következő oldalak ennek szellemében lettek összeállítva!

Ez a füzet egy átfogó elméleti részleget tartalmaz építési kihívásokkal és érdekes információkkal, hogy megtanulhass mindent a robotokról és mindennapi használatukról. Ez a füzet továbbá tartalmaz egy felhasználói útmutatót, mely részletesen elmagyarázza az Engino innovatív robotrendszerét minden szempontból. Fedezz fel minden tudományos elvet kísérletezések során, a lépésről-lépésre útmutatóval és lenyűgöző gyakorlatokkal. Kövesd az építési útmutatókat és készíts izgalmas robotikus modelleket, mint egy skorpiót, egy amóbát, egy automatizált házat, egy ventilátort, egy hatlábút, egy mutatót, egy ásót, egy forgókart, egy kockát és egy robotot. Még több tananyag található online!



Automatikus eszközöket találtak a hieroglifákban

Az ókori feltalálók és kézművesek megpróbálták megépíteni a korai "robotokat". Ezek az egyszerű dizájntól a bonyolultabbakig terjedtek és más-más szinten voltak látszólagosan önrendelkezőek. Ezért is nevezték őket automatonoknak, a Görög "saját akarata alapján cselekszik" kifejezés után, pedig valójában nem voltak erre képesek. Időszámításunk előtti 4. évszázadban Arkhüasz állítólag létrehozott egy gőzmeghajtású repülő szerkezetet, melyet "galamb" névre keresztelt. Az akkori emberek szerint ez a gép képes volt 200 métert repülni! Az alexandriai Héron (Kr. u. 10-70) szintén feltaláló volt, számos kreatív ötlettel, mint az automatikusan nyíló templomajtók, boröntő szobrok és gőz- vagy szélmeghajtású gépek, mint a híres "eolipila", azaz Héron-labda.



A robotika nem létezne számítógép programozás nélkül

A robotika történelme

A robotika történelme már az ősidőkben elkezdődött. Az ókor óta gondolkodnak az emberek azon, hogyan helyettesíthetnék az embereket különböző feladatok elvégzésében mesterségesen épített eszközök, melyek képesek az önálló gondolkodásra. Egy görög mítosz meséli el Talos történetét, egy hatalmas bronz emberszerű gépét, akit Héphaisztosz (A görög mitológiában a kovácsok istene) épített, hogy megvédje Kréta szigetét. Robotszerű lényektől Norvégiától a Közel-Keletig, Indiában és Kínában is találtak feljegyzéseket különböző formákban: szövegesen, rajzolva, festményeken, sőt még az ősi egyiptomi hieroglifák között is.



A Héron-labda készülékének modern másolata



Al-Jazari rajza a vízmeghajtású lánc szivattyúról



A középkorban számos példát találhatunk automatonokra, amik általában emberek vagy állatok alakjában készültek. Az emberek mókás gépeket építettek és programozták, hogy egyszerű feladatokat végezzenek el, mint például mozgassák egy lábukat vagy fordítsák a fejüket, precíz óramű technológia segítségével. A muszlim mérnök, Al Jazari (1136-1206) 100 automatizált, mechanikus eszközt ír le "A Szellemes Mechanikus Eszközök Könyve" című könyvében. Később, a francia feltaláló, Jacques Vaucanson (1709-1782) játszott fontos szerepet a robotika fejlődésében, mikor megpróbálta bemutatni, hogyan működik a természet. Néhány találmánya a valódi hangszereken játszó figurák (fuvola és csörgődob) és a híres "emésztő kacs", ami bemutatta az emésztés folyamatát.

1989-ben a szerb-amerikai feltaláló, Nikola Tesla bemutatta az első rádió-vezérelt hajót. A valódi robotok, azaz azok, amik képesek fogadni és reagálni a környezetükből származó visszajelzésekre, csak a 20. században kezdtek megjelenni. 1948-ban Grey Walter készített apró robotteknősöket fény és érintés érzékelőkkel, akik képesek voltak "élelmet" keresni. Néhány évvel később, George Devol szabadalmaztatta az Unimate robotját, mely már képes volt ipari munkát végezni. 1961-ben helyezték üzembe egy gyárban, hogy forró fémdarabokat emeljen ki a fröccsöntő gépből és egymásra pakolja őket.



Az Unimate robotkar



Az 1960-as években a mérnökök próbálkoztak megalkotni a természet által ihletett robotkarok és lábak mozgását. Az 1970-es és 1980-as években a fejlett technológia és apró számítógépes alkatrészeknek köszönhetően lehetővé vált, hogy minden alkatrészt a robotra helyezzenek ahelyett, hogy kábelek segítségével kapcsolódtak volna külső számítógépekre, mint korábban. Az 1990-es évektől a robotok még fejlettebbé váltak és több, összetett feladat elvégzésére is képesek lettek. Manapság az emberek hétköznapijainak minden részén előfordulnak az otthoni szórakoztatástól kezdődően a kisegítő robotokig (mint AIBO a robotkutya 1999-ben vagy a Roomba robotporszívó 2008-ban) egészen az ipari felhasználásig (például autógyártásban) és felfedezésben (mint Epson, a repülő robot 2004-ből) akár az űrben is (mint a Marsjáró robotok, 2004-ből). A legizgalmasabb viszont, hogy ez még csak a robotika jövőjének kezdete és rendkívül ígéretesnek tűnik!



Tudtad?

A Sci-fi író, Isaac Asimov fogalmazta meg a "**Robotika három törvényét**" a "Körbe-körbe" című művében, 1942-ben. Ezek a következők: 1) A robot nem okozhat kárt egy emberi lényben vagy nem szabad engednie, hogy egy emberi lényt kár érjen; 2) A robotnak engedelmeskednie kell minden emberi utasításnak, kivéve ha az utasítás ütközik az első törvénnyel; 3) A robotnak meg kell védenie magát, de csak míg az önvédelem nem ütközik az első és második törvénnyel



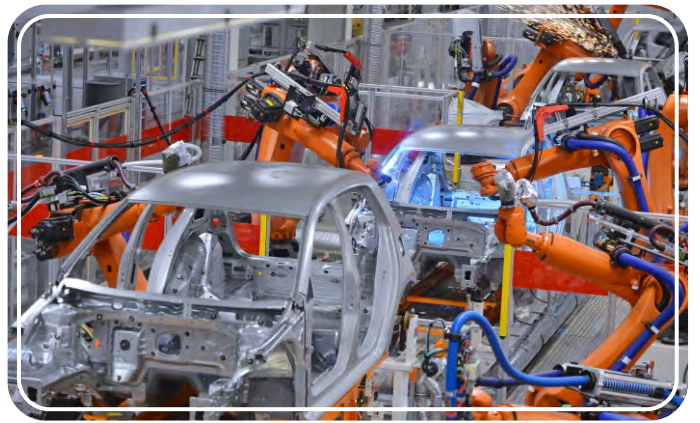
Isaac Asimov (1920 - 1992)



Két féle Roomba porszívó

A robot meghatározása

A "robot" kifejezés először 1920-ban került használatba a cseh Karel Capek által, aki az R. U. R. című regényében használta a kifejezést, mely eredetileg szláv nyelven "munkát" jelent. Ez jól mutatja, hogy az emberek miért így határozzák meg a robot szó jelentését: "az ember számára túl veszélyes, bonyolult vagy egyszerűen csak unalmas tevékenységet elvégző eszköz". Viszont egy még tudományosabb megfogalmazás szerint a robot egy mechanikus eszköz, ami visszajelzést kap környezetétől és képes annak megfelelően cselekedni. Tehát a robotok - valamilyen szinten - képesek az önrendelkezésre és az önálló gondolkodásra. Viszont ez nagyban függ a beléjük programozott és előre megírt utasításoktól.



Robotkarok az autógyárakban



Robotikus metróajtók érintős érzékelőkkel

Robotok típusai

A robotok különböző formákban és alakokban léteznek és egy vagy több feladatot is képesek elvégezni különféle beállításoktól és feltételektől függően. Ezáltal egy robot akár több kategóriába is tartozhat. Hogyan tudjuk tehát megkülönböztetni őket egymástól? Egy könnyű kategorizálás lehet a robotok funkciója szerinti. Ez információt ad nekünk a robot használhatóságáról és ötletet, hogyan nézhet ki vagy milyen alkatrészek kellene hozzá, hogy jól működjenek. A fő típusok funkciók szerint a következők:



Robotikus medencetisztító



Ipari robotkar

Az ipari robotok automatizált robotok, melyeket gyárakban használnak nehéz tárgyak emelésére vagy rutinmunkák elvégzésére, mint például hegesztés, festés vagy anyagok kezelése.

A szórakoztató robotok olyan robotok, amik többnyire csak szórakoztatásra készültek és kevés gyakorlati hasznuk van. Ilyen például a játékróbot.



Játékróbot

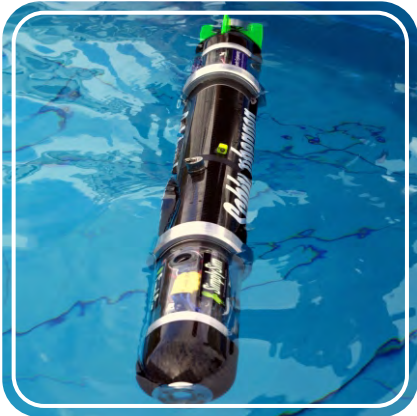


Bombasemlegesítő robot

Az orvosi robotokat az egészségügyi szektorban (kórházakban, gyógyszertárakban, klinikákon és gyógyintézetekben) az orvosok kiképzésére vagy precíz beavatkozásokhoz használják, például sebészeti robotok.

▶ **Katonai robotok**, amik a fejlettebb hadseregek vagy rendőrség részét képezik. Olyan szituációkban használják őket, amik életveszélyesek, mint például lőszerszállítás vagy bombák hatástalanítása, sőt akár harcászati célokra is. Ilyen például egy kis tankrobot.

A háztartásbeli és szolgáltató robotok, olyan robotok melyeket otthoni vagy munkahelyi mindennapos használatra terveztek, hogy megkönnyítsék az életünket. Ilyen például a robotporszívó.



Vízirobot

▶ **A felfedező robotok**, olyan robotok, melyekkel emberek számára elérhetetlen helyeket fedeznek fel, mint például a Föld barlangjai, óceánok (például: vízirobotok) vagy akár más bolygók.

▶ **A virtuális robotok**, olyan robotok, amiket virtuális körülmények között használunk. Ezek általában valamilyen nagy, szemet eltakaró, képernyővel ellátott szemüveggel és különleges kesztyűvel irányítható, hogy használója lássa és érezze mi van a kiterjesztett valóságban.



Virtuális valóság felszerelés



Tudtad?

Létezik egy különleges focibajnokság, melyet "RoboCup"-nak hívnak! Az első robotok által játszott meccs Japánban volt, 1997-ben. A játékosok képesek felismerni és megtalálni a labdát és elrúgni a lábukat használva. A robotok gyártói azt a célt tűzték ki, hogy a robotokból álló focicsapat versenyezhesen a 2050-es férfi Foci Világbajnokságon és nyerjen is meg legalább egy meccset! Szerinted a robotika technológiája el fog jutni ilyen szintre akkorra?



Robot focisták

A hobbi robotok, olyan robotok, amiket robot-rajongók építenek és nem nagy termelővállalatok. A funkciójuk lehet kísérletezés, különféle alkatrészek és technológiák tesztelésére vagy csak szabadidős célokra, mint például a repülő drónok.

A versenyrobotok, olyan robotokat, amik más robotok ellen versenyeznek különféle kihívásokban. Ez a kategória hasonlít az előzőhöz, mert általában az amatőr robotépítők szerveznek és versenyeznek a saját robotjaik képességének összemérése érdekében. A világ minden táján rendeznek versenyeket a robotfejlesztő csapatok számára, hogy segítsenek tesztelni és javítani a robot modelljeiket.



Repülő robot drón

Alkalmazások a valós életben

A 21. század a robotika tudomány és fejlett technológia kora! A robotok már mindenhol megtalálhatóak, nem csak az emberalakúak, de gyakran egész egyszerű formákban, mint a hőszabályzó-rendszerek otthonainkban vagy távolságérzékelők az autókban. A robotikus rendszerek használata a mindennapi életben igen széleskörű és az emberek egyre jobban támaszkodnak rájuk. Olvass tovább pár példáért a robotok felhasználásáról és próbáld meg magad is megépíteni és megprogramozni az ajánlott **Engino Robot** modellek közül egyet!

 **Építés** **Kihívás**

 Az útmutató megtalálható online

Ütközések elkerülése: a modern járművek már fel vannak szerelve mindenféle érzékelőkkel, így a sofőröknek kevesebb dolog miatt kell aggódniuk vezetés közben. Az eső-érzékelő bekapcsolja az ablaktörlőket, ha esik az eső vagy a fényérzékelő képes érzékelni nappal vagy éjszaka van-e és bekapcsolni a lámpákat ennek megfelelően. A biztonságot illetően számos autó rendelkezik távolságérzékelőkkel az elejükben és hátuljukban is, így jelezni tudják a vezetőnek, ha ütközése esélye áll fent, különösen parkolás vagy szűkös helyeken manőverezés közben. Ezek az érzékelők igazi életmentőknek bizonyultak, ha emberek vagy állatok kerültek túl közel az autóhoz anélkül, hogy a sofőr észrevette volna!

Építs meg egy Engino "mutató" modellt és kísérletezz vele, hogy kerüli el az ütközéseket. Tartsd az IR érzékelőt lefordítva, hogy kövessen egy specifikus utat vagy fordítsd őket oldalra, hogy kikerülje az akadályokat, amik a járműtől jobbra vagy balra vannak.



Engino "mutató" modell

 **Építés** **Kihívás**

 Az útmutató megtalálható online

Ütközések elkerülése: a modern járművek már fel vannak szerelve mindenféle érzékelőkkel, így a sofőröknek kevesebb dolog miatt kell aggódniuk vezetés közben. Az eső-érzékelő bekapcsolja az ablaktörlőket, ha esik az eső vagy a fényérzékelő képes érzékelni nappal vagy éjszaka van-e és bekapcsolni a lámpákat ennek megfelelően. A biztonságot illetően számos autó rendelkezik távolságérzékelőkkel az elejükben és hátuljukban is, így jelezni tudják a vezetőnek, ha ütközése esélye áll fent, különösen parkolás vagy szűkös helyeken manőverezés közben. Ezek az érzékelők igazi életmentőknek bizonyultak, ha emberek vagy állatok kerültek túl közel az autóhoz anélkül, hogy a sofőr észrevette volna!

Építs meg egy Engino "mutató" modellt és kísérletezz vele, hogy kerüli el az ütközéseket. Tartsd az IR érzékelőt lefordítva, hogy kövessen egy specifikus utat vagy fordítsd őket oldalra, hogy kikerülje az akadályokat, amik a járműtől jobbra vagy balra vannak.



Engino "automatizált ház" modell



Tudtad?

2014. november 12-én, a történelem során először egy robotikus modul, amit Philae-nek hívtak leszállt egy üstökös felszínére! Ez a lenyűgöző művelet a Rosetta Űrmisszió része volt, melyben egy robotikus űrszonda a 67P üstökös körül keringett 2014 januárjától, hogy tanulmányozza azt. A küldetése az volt, hogy derítse ki, hordoz-e az üstökös bármiféle kulcsinformációt a naprendszer és földi élet létrejöttével kapcsolatban.



A Rosetta űrmisszió képviselője

Építés Kihívás

Veszélyes küldetések: A robotok sikeresen helyettesíthetik az embereket olyan feladatok elvégzésében, amik veszélyesek lehetnek. Így az emberi élet kockáztatása nélkül teljesíthető egy küldetés gyorsabban és jobban a robotok fejlett képességei miatt. Például a robotok képesek érzékelni veszélyes anyagokat, amiket el tudnak távolítani az emberek által irányítva távolról. Az irányító a robot érzékelőinek használatával látja és hatástalaníthatja a bombát.

Építsd meg az Engino "Robot" modellt (38-42. oldal) és helyezd az IR érzékelőt arccal lefelé, hogy kövessen egy adott utat vagy arccal előre, ha szeretnéd, hogy érzékelje a "veszélyes" tárgyakat és mozgással jelezze.



Engino "Robot" modell

Építés Kihívás

Ipari felhasználás: egy másik egyre gyakoribb felhasználása a robotikának az ipari szektorban fordul elő, ahol a robotok átveszik az emberek munkáit. Ezek általában álló vagy adott vonalon mozgó robotok, rugalmas karokkal, hogy nehéz tárgyakat emeljenek vagy precíz munkavégzésre használják őket (pl. hegesztés). Ideálisak a gyárak összeszerelő soraihoz. Az emberek aggódnak, hogy az automatizált folyamatok miatt kevesebb munkahely lesz, de ez nem teljesen igaz, hisz új munkahelyek nyílnak meg a robotokat kezelő és szerelő emberek számára, ezen felül pedig jobb minőségű termékek előállítására is lehetséges a segítségükkel.

Építsd meg az Engino "forgókar" modellt (52-57. oldal) és vezérelj a mozgását az Infravörös érzékelő technológia segítségével.



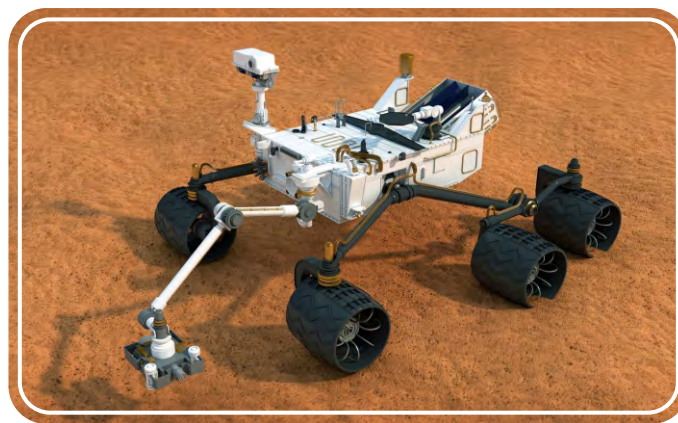
Engino "forgókar" modell

Úrkutatás

Az emberiséget mindig is lenyűgözték a világűr csodái és az új világok felfedezésének, idegen életformák találásának lehetősége. Viszont a Földön kívüli légköri körülmények halálosak lehetnek az emberek számára: napkitörések, oxigénhiány, gravitációs mezők és abszolút nulla hőmérséklet (-273 Celsius fok) csak pár ezek közül a veszélyek közül. Mindegyikhez rendkívül drága szkafanderre van szükség, több éves kiképzésre és tervezésre csak egy rövid Föld körüli vagy Hold felszíni sétához is. Ami pedig a távolságot illeti (pl. elutazni egy másik bolygóra) a felfedezőknél késznek kell lenniük rá, hogy csak évtizedek múlva tudnának visszatérni a Földre, ha egyáltalán vissza tudnak térni!



Marsjáró "Opportunity"

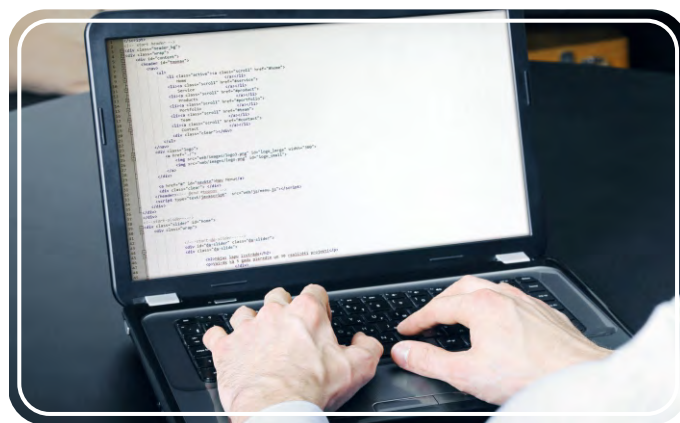


Marsjáró "Curiosity"

Azonban szerencsére léteznek robotok! Ők ideálisak az úrkutatáshoz és minden túlélési problémát megoldani tudnak, illetve nem bájnak mennyi időt kell eltölteniük egy küldetésen. A fejlett képességeiknek köszönhetően felfedezhetik a bolygó felszínét, megvizsgálhatják a talajt és légkört, vizet kereshetnek és más kémiai anyagokat, illetve rengeteg értékes, nagyfelbontású képet küldhetnek vissza a Földre. Jelenleg két aktív marsjáró van küldetésen: Opportunity (Lehetőség) és Curiosity (Kíváncsiság), de számos más robot is kering aszteroidák, üstökös és bolygók körül, mint az emberiség nagykövetei!

Programozás

A mechanikus alkatrészekon kívül a programozás is létfontosságú egy működőképes robothoz. Ahogy már korábban említettük, egy eszköz akkor számít robotikusnak, ha képes a környezetével interakcióba lépni. A robot feldolgozza az összes kapott információt és eldönteni annak megfelelően, hogy mi a megfelelő művelet a programja szerint, a mikrovezérlőn keresztül. A robotgyártók általában saját programozási nyelvet használnak, ezért rengeteg különböző robotikus szoftver létezik. Szerencsére van néhány alapfogalom, amit mindenkinek követnie kell, mikor programot fejleszt. Ezekről a következő bekezdésekben olvashatsz.



Egy programozó munka közben

Programozási nyelvek

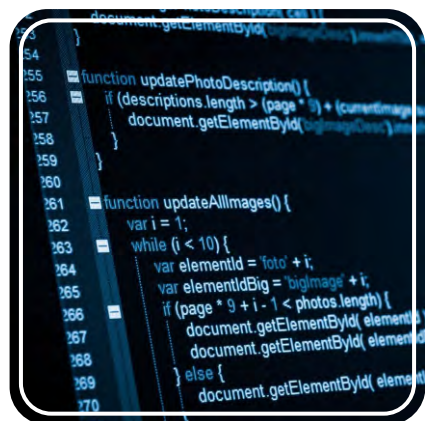
Bármely számítógép vagy robotfejlesztés legfontosabb eleme a programozási nyelv. A programozási nyelv általánosságban a robot különböző feladatok elvégzésére utasító szókincse és nyelvtana. A nyelv két fő összetevőből áll: a szöveg formájából (szintaxis) és jelentéséből (szemantika). Két fajta programozási nyelv létezik: az alacsony szintű, ami a gép értelmezési módját alkalmazza (gépi kóddal, számokkal és szimbólumokkal) és a magas szintű, ami emberi nyelvet tartalmaz (szöveges formában) és automatikus, ezért a programozási folyamat sokkal egyszerűbb és érthetőbb az emberek számára. A számítógépes szoftvereknél, a futtatható leírása egy programnak (programozási nyelven készítve) az úgy nevezett "forráskód". A forráskódon belül vannak olyan algoritmusok, amik meghatározott sorrendű műveleteket írnak le, melyeket egy egyedülálló programozási részek alkotnak, az utasítások (lásd lent).

Aki szeretne számítógépes programot fejleszteni, annak nagyon körültekintően kell kiválasztania a programozási nyelvet hozzá, mert mindegyik nyelvet más-más használatra és céllal terveztek. A leggyakoribb magas szintű programozási nyelvek:

- A C++ nyelv valószínűleg a legszélesebb körben elterjedt és a legtöbb alkalmazást ezzel fejlesztik.
- A Java használható minden eszközön, ahol a Java Virtuális Gép (JVM) telepítve van, a rendszertől függetlenül.
- A JavaScript egy webkliens alapú programnyelv, ami leginkább a hálózati szerverek helyett a felhasználók PC-jén hajt végre műveleteket.
- A Python az egyik legkönnyebben megtanulható programozási nyelvként ismert.



A HTML nyelvet specifikusan a weboldalak fejlesztésére használják



Az utasításokat szöveges sorokkal írják

Utasítások

A számítógép programozásban az utasítások lényegében útmutatások (parancsok), amik megmondják a számítógépnek, hogy mit csináljon. Amennyire csak lehet, ezeknek a parancsoknak pontosnak kell lenniük, hogy a robot jól működjön. Ezen kívül a parancsok sorrendje, azaz a "feladatok sorrendje" is ugyanennyire fontos. Képzeld el, hogy egy tortasütés útmutatóját követed. Ha a recept lépései rosszak vagy rossz sorrendben vannak írva, akkor nagy eséllyel nem sikerülni, csak rendetlenséget csinálunk a konyhában! A szöveges alapú programozási nyelvekben a parancsok általában soronként vannak írva és különleges karaktereket tartalmaznak a különböző feladatokhoz. Ezt a KEIRO szoftver is szimulálja, amiben egy ablakban (KEIRO kód) egy kezdetleges programozási nyelvet (nem valódi programozási nyelv), ami segít a felhasználóknak megtekinteni a különböző parancsokat végrehajtási sorrendben szövegesen.



Gép, ami manuális programozást használ

A hagyományos irányítóhoz hasonlító, oktató vezérlőegység használatával a felhasználó távolról megtaníthatja a robotnak, hogyan kell specifikus mozdulatokat végrehajtania pl.: felemelni egy autókabát részt és a gyárszalagra helyezni. A manuális programozás során a parancsok sorban felíródnak és automatikusan elmentődnek, így a robot képes lesz a munkát ugyanúgy végrehajtani később is. A programozók pedig tovább szerkesztheti a kódot, módosíthatják a sebességet, pontosságot vagy a mozgás folyamatosságát. Amikor minden készen van, a kész programot visszaküldik a robotnak.



Általában a gyári robotikus program létrehozásának folyamata a parancsok leírásával kezdődik programozási nyelven, majd ellenőrzik, hogy bizonyos feltételek mellett működnek-e megfelelően és oda-vissza állítgatják, módosítják, míg megfelelő eredményt nem érnek el vele. Ez a folyamat bonyolult kódolási képességeket igényel és gyakran csak meghatározott mozgásra korlátozódik, mint például a robot egyszerűen egyik helyről a másikra kell mozdítson tárgyakat. De mi van akkor, ha szeretnénk utánozni a kifinomult mozdulatait egy munkásnak, hogy a robot programja egy valódi ember munkáját tudja végezni anélkül, hogy bonyolult programokat írni hozzá? A modern robotikus technológia lehetővé teszi ezt számunkra a manuális programozás segítségével.



Egy dolgozó a oktató vezérlőegységet használ



A kézi programozás az Engino® KEIRO™ szoftver egyik fő funkciója. Ez a fajta programozás nagyon hasznos programok készítéséhez valós körülmények között anélkül, hogy ismernénk magát a szoftvert. A felhasználó egyszerűen csak megnyomja a gombokat az ERP MINI 2.0 kontrolleren és rögzíti a parancsok sorrendjét, amit a robot valós időben elvégez. Miután a program elmentésre kerül, később bármikor lefuttatható, akár sorozatosan is. Továbbá, ha a felhasználó szeretne változtatni az értékeken vagy extra feltételes utasításokat hozzáadni, ezt nagyon könnyen megteheti a készüléket a PC-hez kötve és a programot a KEIRO™ szoftverbe elküldve. Amikor a változtatások készen vannak, a programot vissza lehet küldeni a robotba tesztelésre.

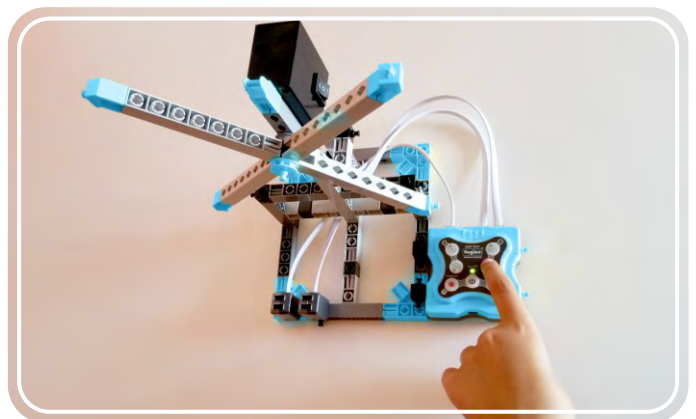


Tudtad?

Az első számítógépes programozó egy angol matematikus nő volt, Augusta Ada King (Lovelace grófnője). 1837-ben Ada írta a történelem első algoritmusát, amit egy gépnek kellett volna teljesítenie. Pontosabban, az algoritmust a Bernoulli számok analízáló gépéhez készítette, egy ötletként Charles Babbage mechanikus általános felhasználású számítógépéhez.



Augusta Ada King festménye (1815 - 1852)



A manuális programozás használata a MINI 2.0 vezérlővel



Használati útmutató

Engino robotika filozófiája

A szellemi fejlődés segít a tanulóknak, hogy fejlesszék ötleteiket új információk megszerzésével. Ez nélkülözhetetlen a kreativitás és laboratóriumi képességek fejlődéséhez és javításához. A MINI 2.0 kontrollert és a Scratch-szerű KEIRO™ szoftver kombinációja ideális megoldás a robotika tanításához. A MINI 2.0 kontrollert segítségével négy egymással összekapcsolt programozási módot biztosít, hogy a felhasználók kiválaszthassák a számukra megfelelő módszert életkoruk és tapasztalataik szerint.



Manuális programozás

A programozás első szintje a közvetlen kapcsolat a tananyaggal. Gombokat nyomkodni minden gyerek tud. A diákok rögzíthetik a parancsok sorrendjét a "Program" gomb segítségével, elmenthetik a készüléken és a "Play" gomb segítségével futtathatják, hogy megismételje a folyamatot. Ez a programozási módszer nélkülözhetetlen az programozási parancsok sorrendjének megismeréséhez.

Szimulátor

Ez egy speciális ablak a KEIRO szoftverben, ami szimulálja a valódi kontrollert digitális gombokkal a fizikai gombok helyett. Amint a MINI 2.0 kontrollert hozzá van kötve egy készülékhez (akár PC-hez, akár Tablet-hez), a felhasználó rögzítheti a programot és látható visszajelzést kaphat a robottól. Míg a program rögzítésre kerül, egy folyamatábra generálódik és megjelenik látható blokkoként.



```
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
```

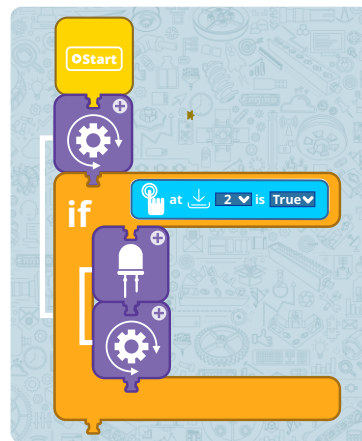
KEIRO kód

Ez kezdetleges programozási nyelv (Nem valódi programozási nyelv), amit specifikusan a KEIRO™ szoftverhez készítették. Számos ismerős programozási kifejezést tartalmaz, mint a BEGIN, IF, END, stb. Ez egy ideális eszköz a haladó programozás bemutatására és gyors betekintést ad a program szöveges formájához.

Folyamatábra

A Folyamatábra a fő programozási funkció a KEIRO™ szoftverben. Itt a felhasználó használhat minden kimenetet és bemenetet (output/input) egy robotnál, hogy kifinomult programot készíthessen. A platform úgy lett tervezve scratch-szerű programozási nyelvként blokkokkal mutatja be a manuálisból digitálisba változó programot.

A fogd és vidd (Drag-and-drop) programozási környezet hasznos oktatási eszköznek bizonyult, amivel könnyedén interakcióba lehet lépni a valós világgal és intuitív ember-gép közötti kezelőfelületet kialakítani.

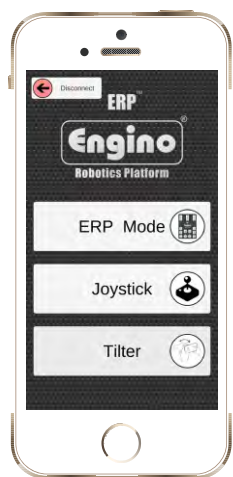


A folyamatábra úgy készült, hogy könnyedén lehet fogni és arrébb tenni a blokkokat.

Okoskészülékes csatlakozás

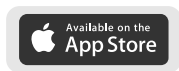


A modern technológia fejlődéseinek megfelelően, a MINI 2.0 kontroller irányítható egy okoskészülékkel (mobil vagy Tablet) Bluetooth kapcsolaton keresztül. A KEIRO™ szoftver egy Scratch-alapú grafikai programozási platform, ami használható PC-n vagy okoskészüléken. A robot programozása mókás módon történik, így minden lehetőséget adott, hogy a diákok örömmel álljanak neki dolgozni a projektjeiken.



KEIRO Szoftver

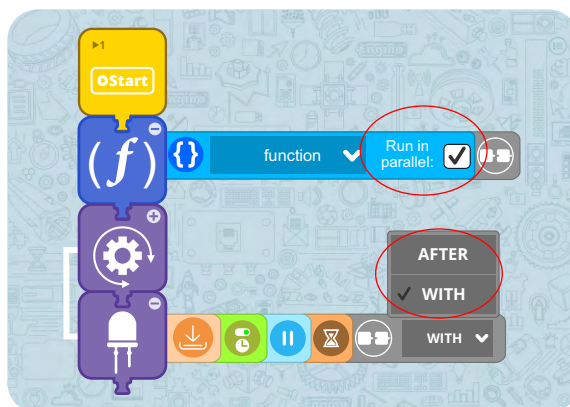
EnginoRobot BT



Ezen kívül az EnginoRobotBT letölthető Google Play-ről vagy az Apple Store-ból is, mellyel szimulálható egy tényleges kontroller kezelőfelülete. A felhasználók irányíthatják a modelleket, rögzíthetnek és futtathatnak programokat, pontosan úgy, mintha a MINI 2.0 kontroller fizikai gombjait nyomkodnák. Ezeket mind megtehetik távolról és digitálisan! Az applikáció két extra irányító módot is tartalmaz, a Joystick módot és a Tilter módot. Töltsd le az applikációt ingyen és bővítsd ki a manuális programozási lehetőségeidet!

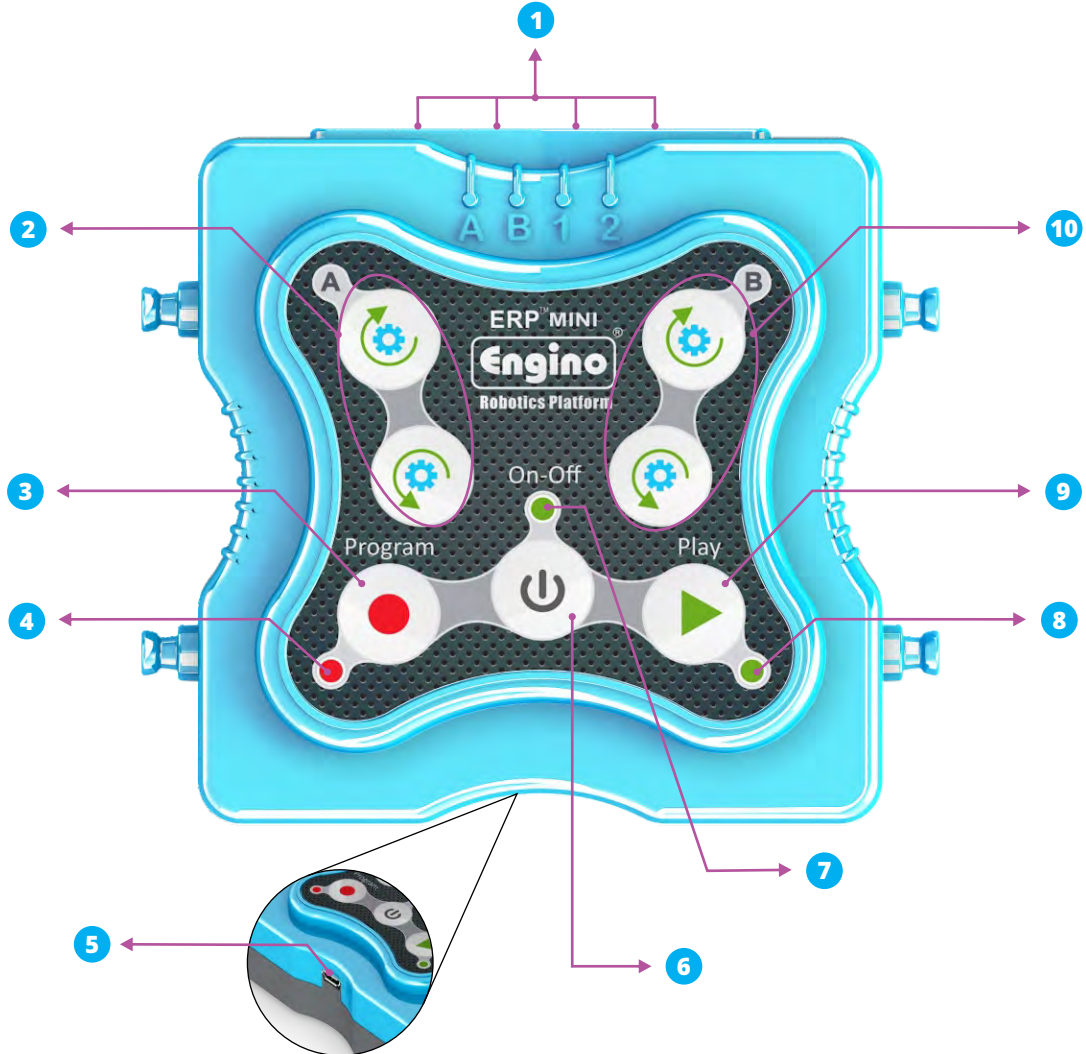
Párhuzamos programozás

Az egyik fő újítás az Engino által a robotika oktatásban a párhuzamos programozás funkciója. Akció blokkok a Keiro™ szoftverben ötletesen vannak kitalálva, segítségükkel a felhasználó választhat, hogy a parancs teljesüljön az előző után vagy vele együtt. Továbbá a szoftver meghívhat egy funkciót (parancs többszörözés) azzal a lehetőséggel, hogy a kód egy későbbi műveletblokkjával párhuzamosan fusson. Ezek a lehetőségek csökkentik a programozás összetettségét és segítenek, hogy a kifinomult feladatokat könnyebben lehessen megvalósítani.



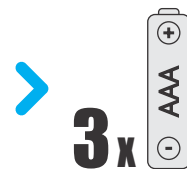
MINI 2.0 kontrolller

Az Engino MINI 2.0 kontrolller egy robotikus eszköz, ami tartalmazza a fő vezérlőt és számos más gombot és jelzőt. Az eszközön található 4 porton keresztül csatlakoztathatók különféle perifériák, mint motorok, led-ek, infravörös és érintésérzékelők. A MINI 2.0 kontrolller egy erős, flexibilis és könnyen használható eszköz. Ez az útmutató segít megismerni a fő funkcióit a kontrolllernek és telepíteni a KEIRO™ szoftvert és megtanít, hogyan programozd a robotodat különböző kód példák alapján.



- | | |
|---|---|
| 1 A,B,1 és 2 RJ portok
<i>perifériák csatlakoztatása a kontrolllerhez</i> | 6 POWER GOMB |
| 2 MOTOR A GOMBOK
<i>óráirány és óráiránnyal ellentétes</i> | 7 BE/KI LED
<i>bekapcsol, ha a kontrolller be van kapcsolva</i> |
| 3 PROGRAM GOMB
<i>rögzít egy parancssort</i> | 8 FUTTATÁS LED
<i>bekapcsol, ha épp lejátszik egy programot</i> |
| 4 PROGRAM LED
<i>világít rögzítés közben</i> | 9 FUTTATÁS GOMB
<i>lefuttatja a tárolt programot</i> |
| 5 Mini USB PORT
<i>csatlakoztatja a kontrollert PC-hez</i> | 10 MOTOR B GOMBOK
<i>óráirány és óráiránnyal ellentétes</i> |

Elemek: A MINI 2.0 kontrollert használatához szükséged lesz három (3) AAA elemre, melyeket a hátoldalán tudsz behelyezni a kontrollerbe. Szükséged lesz továbbá egy csillagcsavarhúzóra, hogy kinyisd az elemfedelelet. Helyezd be az elemeket a helyes polaritással, a + és - jelek mutatják az elemek helyes pozícióját. Miután behelyezted az elemeket, csavard vissza az elemfedelelet a helyére.



Biztonsági előírások: Ne töltsd túl az elemeket. Ne használj vegyesen újratölthető és hagyományos elemeket. Nem tölthető elemeket ne próbálj meg feltölteni. Ne keverd a használt elemeket az új elemekkel. Távolítsd el a lemerült vagy rossz elemeket a készülékből, ha nem tervezed használni a készüléket hosszabb ideig. Sose dobd a használt elemeket a tűzbe! Ne zárd rövidre a készüléket!

Perifériák, kábelek és portok



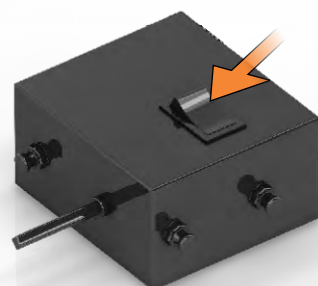
RJ kábel port



RJ kábel port



RJ kábel port



Motor kapcsolója

LED (piros)

Használj egy LED lámpát, hogy jól nézzen ki a robotod!

Érintés érzékelő

A kapcsoló gomb segítségével aktiválhatod a mozgást, ha hozzáérsz.

Infravörös érzékelő

Az infravörös érzékelő használható tárgyak érzékelésére vagy útvonal követésére.

Egyenáramú motor

Egy nagynyomatékú egyenáramú motor állítható sebességgel, ami képes meghajtani a kerekeket és mozgásra bírni a robotodat.



Érzékelő kábel

Egy kábel, ami összeköti az érzékelőket és LEDeket a MINI 2.0 kontrollerral, mindkét végén RJ csatlakozó van.



Motor kábel

Egy kábel, ami összeköti az egyenáramú motort a MINI 2.0 kontrollerral, egyik vége RJ csatlakozó a másik pedig jack csatlakozó.





Mini USB kábel

Egy kábel, ami összeköti a MINI 2.0 kontrollert és a PC-t.

Az I pozícióban a motor üzemel a kontrollert és a KEIRO™ szoftver parancs blokkjai által megadott irányba. Amikor II pozícióra váltasz, az irány megváltozik. A motor kikapcsol, ha a pozíciót O-ra állítod.



Perifériák és portok: Az Engino MINI 2.0 kontrolleren négy (4) RJ port található a motorok, LEDek és érzékelők irányításához. Viszont különféle perifériák csak a controller specifikus portjain működnek. A lenti táblázatban áttekintheted, hogy melyik port melyik perifériához* való.

PERIFÉRIÁK	PORTOK			
	A	B	1	2
	✓	✓		
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓

1. táblázat: Perifériák és hozzájuk tartozó portok

*A perifériák (LED és érintésérzékelő) nem részei a csomagnak, az Engino oktató csomag STEM & Robotics MINI kiadásban találhatóak (www.enginoeducation.com)

Manuális programozási példa

A MINI 2.0 controller manuális irányítást ad és lehetőséget a program rögzítésére a controller membrán gombjainak segítségével. A felhasználó fokozatosan válthat a fizikai manuális programozásról a szoftveres irányításra az innovatív visszafejtő módszerrel! A rögzített program a készüléken könnyedén feltölthető PC-re vagy okoskészülékre.

*Mikor a motor hozzá van kötve a port A-hoz
(Állítsd a motort az I pozícióba)*

1. Nyomd meg a "Program" gombot, hogy elkezd a rögzítést (A piros jelzés villogni kezd).
2. Tartsd nyomva az óramutató irány gombot a motor A-n 3 másodpercig
3. Tartsd nyomva az óramutató iránnyal ellentétes gombot a motor A-n 2 másodpercig
4. Nyomd meg újra a "Program" gombot, hogy elmentsd a programot a memóriába.
5. Nyomd meg a "Play" gombot, hogy futtasd a programot (a zöld jelölő villog).



Tippek és trükkök:

- Ha nyomva tartod a futtatás gombot 3 másodpercnél tovább, a program folyamatosan ismétlődni fog
- A készülék csak a legutolsó rögzített/elküldött programot tudja tárolni a memóriájában

KEIRO Szoftver telepítés

Egy gép, ami képes parancsok sorozatát teljesíteni nem egy igazi robot; az igazi robotot programozni lehet arra, hogy döntést hozzon az érzékelőtől kapott visszajelzések szerint. Viszont a manuális programozás nem használ érzékelőket. Ahhoz, hogy egy haladó programot készítsünk módosítani kell a változókat és hozzáadni érzékelőket, ehhez pedig telepíteni kell a **KEIRO™ szoftvert**. Letölthető itt: www.engineerrobotics.com

1. Látogass el a www.engineerrobotics.com oldalra. Nyomj a "Download software" fülre. Válaszd ki az ERP MINI 2.0 csomagot.


2. Válaszd ki a megfelelő operációs rendszert és nyomj a "Download" gombra.


3. Telepeítsd a programot a letöltött fájlból.

4. Amikor a telepítés befejeződött, indítsd el a programot

Letöltheted a KEIRO szoftvert a Google Play vagy Apple Store áruházakból is egy okoskészüléket használva.

ERP MINI 2.0 BLUETOOTH-AL

 [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

MINI 2.0 controller rendszerfrissítés

Ha egy új rendszerfrissítés érhető el a controller számára, a KEIRO™ szoftver értesíteni fog róla egy felugró ablakban, amint összekapcsoltad a controllerrel.

Nyomj a "YES" gomra és töltsd le a legfrissebb rendszerfrissítést.

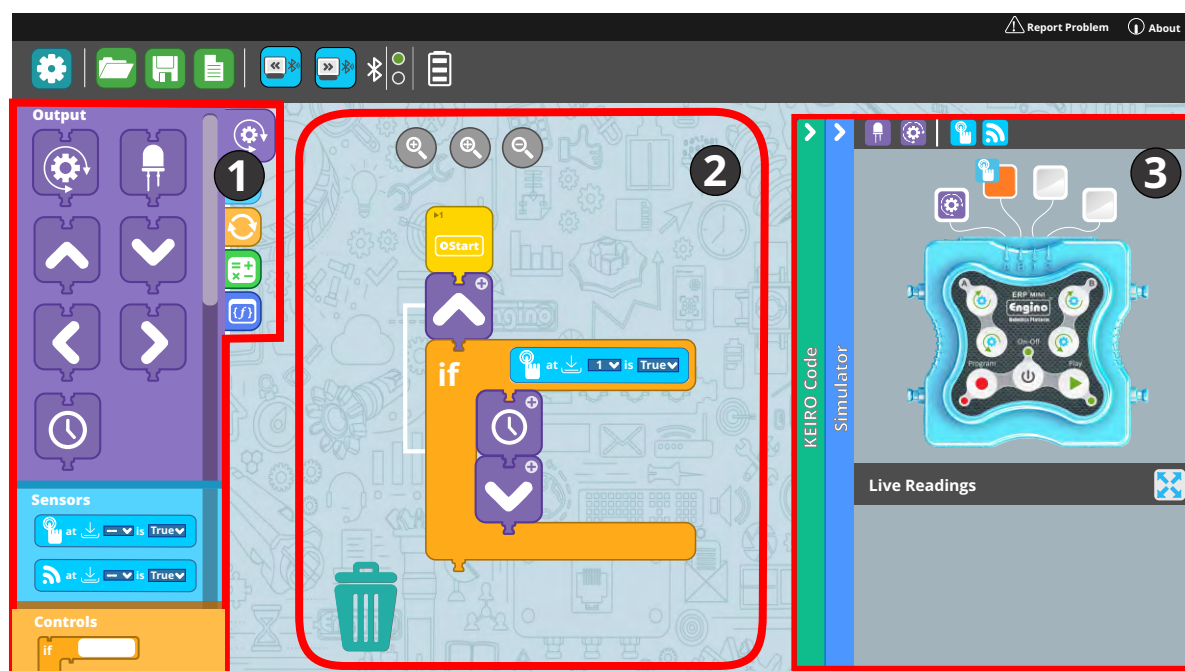


A KEIRO kezelőfelület

1. **Akció Blokk lista** - itt található az összes akció blokkot, amiket használhatsz a parancssorok készítéséhez. A könnyű használat érdekében a blokkok kategóriák szerint vannak rendezve.

2. **Folyamatábra** - A terület, ahová pakolászhatod a blokkokat és programoz hozhatsz létre a robot számára. A program szerkesztéséhez a különböző értékeket kell változtatni a különféle blokkokban.

3. **Szimulátor/KEIRO™ kód** - A szimulátor beállítása szükséges, hogy a szoftver tudja, melyik portokon melyik perifériák vannak csatlakoztatva. A kód rész mutatja a programot szöveges formában blokkok helyett. (Kattints rá a különböző sorokra, hogy elrejtésd vagy kinagyítsd az ablakot)



Menü

A menüsor a KEIRO szoftver tetején helyezkedik el. Itt található a különböző gombokat és feladatokat, mint például a csatlakozás a MINI 2.0 controllerhez, visszatérés a főképernyőre, program megnyitása/mentése stb.

Beállítások

Itt válthatsz nyelvet vagy egyéb beállításokat módosíthatsz

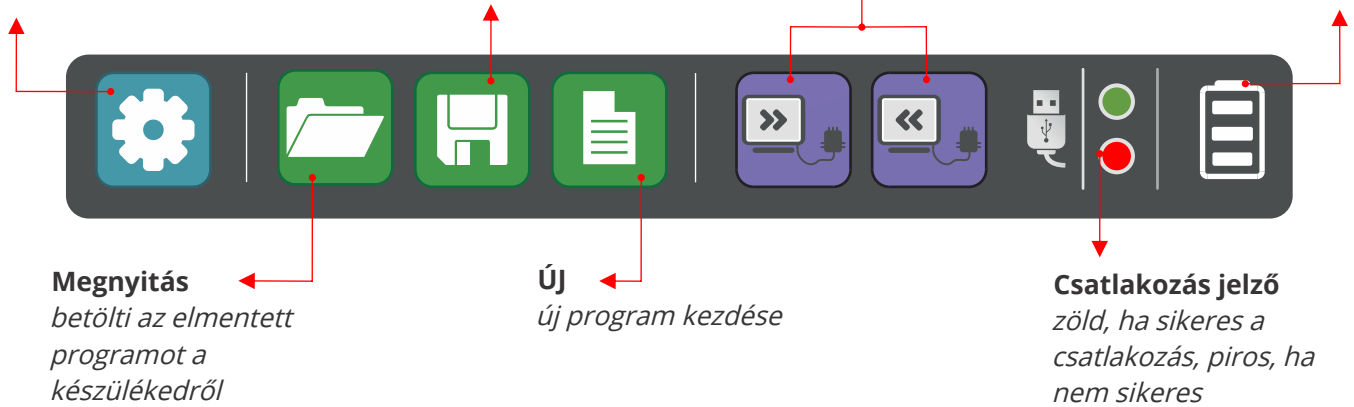
Mentés

mentsd el a programot a készülékedre

Küld/Fogad

A program küldése/fogadása a robotnak/tól

Elemek
A töltöttség mutatója



Megnyitás

betölti az elmentett programot a készülékedről

Új

új program kezdése

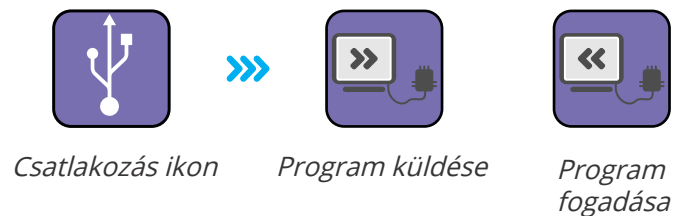
Csatlakozás jelző

zöld, ha sikeres a csatlakozás, piros, ha nem sikeres

Csatlakozás PC-vel

Csatlakoztathatod a MINI 2.0 kontrollert a PC-hez a hozzá tartozó mini USB kábellel és utána a menüben megfelelő ikonra nyomással. Amint sikerült csatlakozni a piros jelölő a menüsoron zöldre vált, mutatva, hogy sikeres a csatlakozás a PC és controller között.

Ha a csatlakozás sikeres volt, két új gomb jelenik meg a menüsoron. A "Send" gomb segítségével küldhetsz programot a PC-ről a MINI 2.0 controllerre vagy vissza, a megfelelő gombra nyomva.

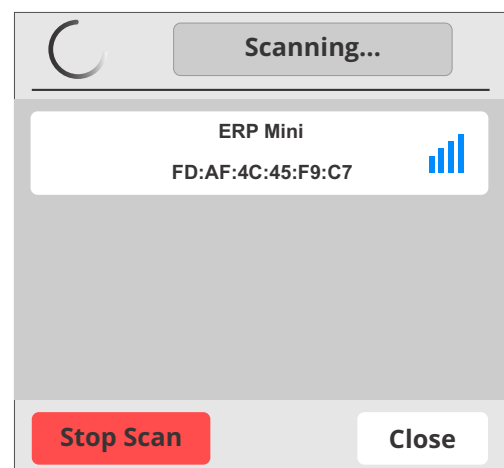


Csatlakozás Okoskészülékkel

Ha a KEIRO szoftvert okoskészülékeken (Androidos vagy IOS-es készülékkel) használod, a bluetooth ikon jelenik meg a menüsoron az USB helyett.



Hogy sikerüljön csatlakoztatni a MINI 2.0 kontrollert, engedélyezd a bluetooth használatát a készüléken és kapcsold be a MINI 2.0 kontrollert. A bluetooth ikonra kattintva a szoftver megkeresi a csatlakoztatható eszközöket. Az újonnan megjelenő ablakban válaszd ki a MINI 2.0 kontrollert, hogy csatlakoztasd a Tabletedhez.



Keress bluetoothos készülékeket és csatlakoztasd az ERP Mini-t

Szimulátor Panel

Szimulátor beállítása

A Szimulátor egy különleges ablak, amiben a KEIRO szoftver beállítja a robot különböző perifériáit. Mielőtt használod a folyamatábrát, fontos beállítanod a Szimulátort és hozzácsatlakoztatni a különböző perifériákat a robothoz az adott portokon. Ezt megteheted az ikonok megfogásával és megfelelő portra húzásával. A szimulátor tetején megtalálod a mindegyik perifériához az ikonokat



LED lámpa
ikon blokk



Egyenáramú Motor
ikon blokk



Érintés érzékelő
ikon blokk



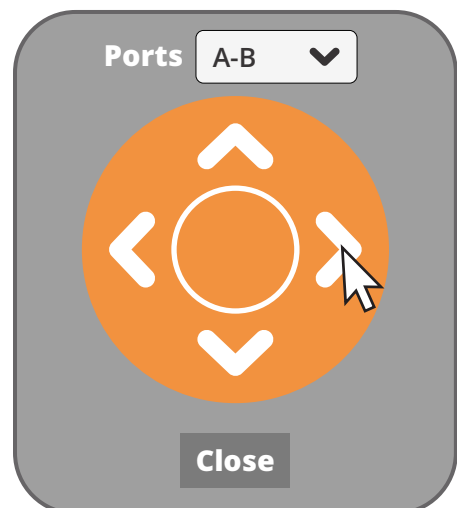
Infravörös (IR) érzékelő
ikon blokk



GamePad Panel



A GamePAD panel megjelenik, ha egy speciális ikonra nyomsz a szimulátoron belül. Használható a két motor port egyidejű irányítására. Emlékezz, hogy a motorok csak a port A és port B-hez csatlakoztathatóak, ezért ez a kombináció kell legyen beállítva. A GamePad hasznos opció a robot jármű irányításához egy könnyű és egyszerű módon. Csak kattints (vagy érintsd meg okoskészüléken) a nyilakra és a motorok reagálni fognak a parancsoknak megfelelően.



A GamePad panel

A szimulátor segítségével távolról is lehet irányítani a modellt. Az ikonok, amik a controller gombjait imitálják valójában aktív gombok, amikre rányomhatsz. Így képes vagy rögzíteni vagy futtatni a programot a szimulátorból is. Figyelem! A rögzítő program akció blokkjai automatikusan generálódnak a folyamatábrán.

Élő adás

Az **élő adás** részleg azonnali betekintést nyújt a perifériáktól kapott visszajelzésekre, amint a kontroller és a szoftver kapcsolódnak egymáshoz. A felhasználó figyelemmel követheti minden csatlakoztatott eszköz állapotát, mint például a sebességét és forgási irányát a motornak, be van-e kapcsolva a LED vagy meg van-e nyomva az érintésérzékelő. Ezen kívül az IR érzékelő is kalibrálható specifikus távolság érzékelésére (Az IR érzékelő kalibrálásáról a 25. oldalon olvashatsz bővebben).

The 'Live Readings' interface displays four sensor modules, each with a download icon, a port selection dropdown, and a status indicator:

- Motor:** Port: A, Forgás: ACL, Sebesség: 80
- LED:** Port: 2, Állapot: ON
- Érintésérzékelő:** Port: 1, Állapot: TRUE
- Infravörös érzékelő:** Port: 1, Állapot: [checked], Érzékenységi skála: 65, Kalibrálás: [Calibrate]

Motor: Mutatja a forgás irányát (Óramutatóval megegyező CL,

- Óramutatóval ellentétes ACL, Kikapcsolt OFF) és sebességét (0-100)

LED: Mutatja, hogy be van-e

- kapcsolva vagy sem. (ON - Be van kapcsolva, ha világít, OFF - ha nem világít, ki van kapcsolva)

Érintésérzékelő: Mutatja, hogy meg

- van-e nyomva vagy sem. (TRUE - ha nyomva van, FALSE - ha nincs)

Infravörös érzékelő: Mutatja hogy

- érzékel-e épp egy tárgyat vagy sem. (✓ - ha érzékel, X - ha nem.) (bővebben a 25. oldalon)

KEIRO kód

A KEIRO szoftver egy kezdetleges programnyelvet generál a folyamatábra diagramm blokkjai alapján. Ez egy alapvető oktatási eszköz, ami bevezeti a diákokat a programozás valódi világába, fokozatosan váltva a képalapú blokkokról az érthető szöveges formára.

A KEIRO kód tartalmazza a legismertebb kifejezéseket a programozásban, úgy mint a BEGIN, IF, IF/ELSE, WHILE stb. Így a diákok könnyedén viszonyíthatják az akció blokkokat a szöveghez és fordítva!

Kattints a menüsorra, hogy kinagyítsd vagy elrejtisd a panelt

```
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
```


Akcio Blokk sor



A program elkészítéséhez használható blokkok mind ebben a menüben találhatóak. A felhasználó csak kiválasztja milyen akció blokkot szeretne használni és odahúzza a folyamatábrára, hogy programoz készítsen.

Az Ikon Blokk 3 kategóriába vannak csoportosítva:

1. Kimenetek (Outputs):

Motor
Mozgás (Előre, Hátra, Balra, Jobbra)
LED
Várakozás

2. Érzékelők

Érintésérzékelő
Infravörös (IR) érzékelő

3. Irányítás:

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

4. Műveletek

5. Funkciók

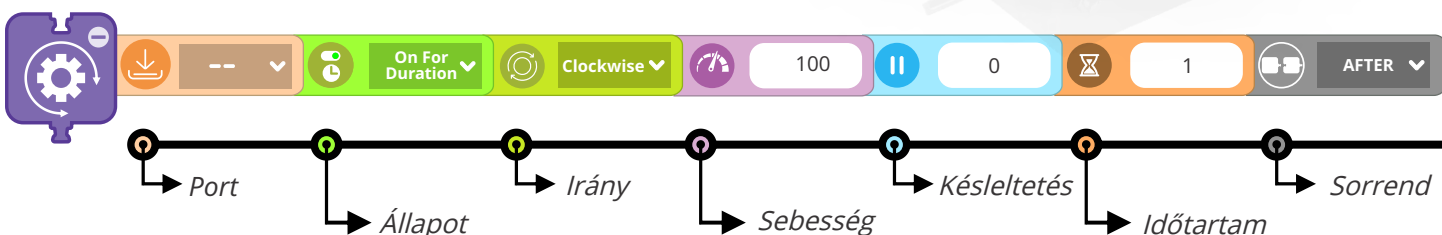
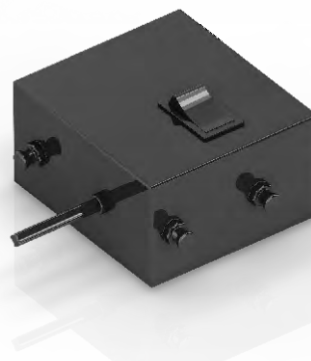
1. Kimenetek (Outputs):

Ebben a kategóriában pakolászhatod a kimeneteket a folyamatábrára, hogy elkészítsd a különböző akciókból álló sorozatot. Amikor egy blokkot ráhelyezel a folyamatábrára, megnyílik a tulajdonságát tartalmazó menüsor, hogy szerkeszthesd a blokkot. Például, ha be kell állítanod, hogy melyik porthoz van kötve a kontrolleren a kimeneti eszköz. Ezen kívül szerkesztheted az akció blokkot, hogy megváltoztasd mennyi ideig és milyen sebességgel üzemeljen a motor.



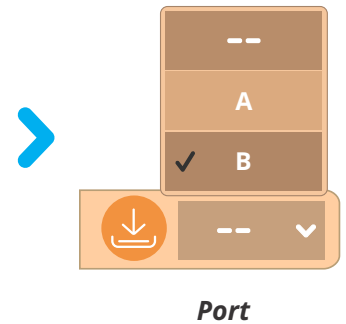
Motor Blokk

Lent láthatod a motor blokkot részletesen, benne az alapértelmezett értékeivel és tulajdonságaival.

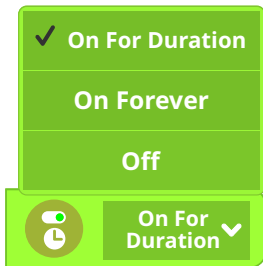


Add meg a Portot, amihez a motor csatlakoztatva van a kontrolleren. Az alapbeállítás szerint ez nincs beállítva, ezért nyomj a doboz részre és válaszd ki a megfelelő portot. Az elérhető portokat sötétebb színnel jelzi.

A szimulátorhoz legalább egy motor megadása szükséges, hogy engedélyezze a portot.



Port



Állapot

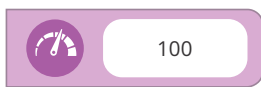
Állítsd be az Állapotát a motor mozgásának. Három lehetőség közül választhatsz: Adott időre bekapcsolás (alapbeállítás): Motor forgatása egy előre megadott ideig, az időtartam megadásával (lentebb olvasható). Folyamatosan bekapcsolva: A motor folyamatosan forog, amíg nem kap másik parancsot. Kikapcsolva: A motor forgása megáll.

Válassz ki egy irányt, hogy merre forogjon a motor. Választhatsz az óramutatóval megegyező (alapbeállítás) vagy óramutatóval ellentétes forgási irányt. Emlékezz, hogy a motor kapcsolója az I pozícióba kell állítva legyen, hogy egyezzen a választott iránnyal!

Ha az állapota OFF, akkor ez a beállítás nem elérhető.



Irány



Sebesség

Állítsd be a forgási sebességét a motornak. Írd be a szövegdobozba az értéket, ami lehet 0-100 között. Az érték valójában százalékos értéke a motor energiaellátásának, így a 100 (alapbeállítás) jelenti, hogy maximális teljesítménnyel megy a motor.

Ha az állapota OFF, akkor ez a beállítás nem elérhető.

A Késleltetés idejét másodpercekben lehet megadni, hogy a motor blokk műveletét későbbi időben kezdje. Ha az értéke 0 másodperc (alapbeállítás), akkor nincs késleltetés.



Késleltetés



Időtartam

Add meg az időtartam idejét másodpercekben, hogy a motor forgása egy adott ideig tartson. A motor megáll, ha eltelt a megadott idő.

Ez a beállítás csak akkor elérhető, ha az "On For Duration" állapotra van kapcsolva.

A blokkok sorrendje alapbeállítás szerint egymás után következik. Ahhoz, hogy egyidőben teljesüljenek bizonyos feladatok válaszd a "with" (előző) opciót, így az adott blokk egyszerre kerül végrehajtásra az előző blokkal. Ez a beállítás lehetővé teszi, hogy több blokk is végrehajtsódjon párhuzamosan.



Navigáció (Mozgás) blokk

Azért, hogy a jármű megmozduljon, mindkét motornak működni kell egyidőben. Ahelyett, hogy két blokkot húznál a folyamatábrára, használj mozgás segéd blokkokat a jármű irányításához, hogy előre, hátra, balra vagy jobbra haladjon. Így sokkal egyszerűbb navigálni, csökkenti a bonyolultságot és kevesebb blokk használata kell a folyamatábrához.



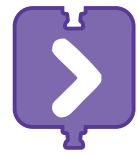
Előre



Hátra

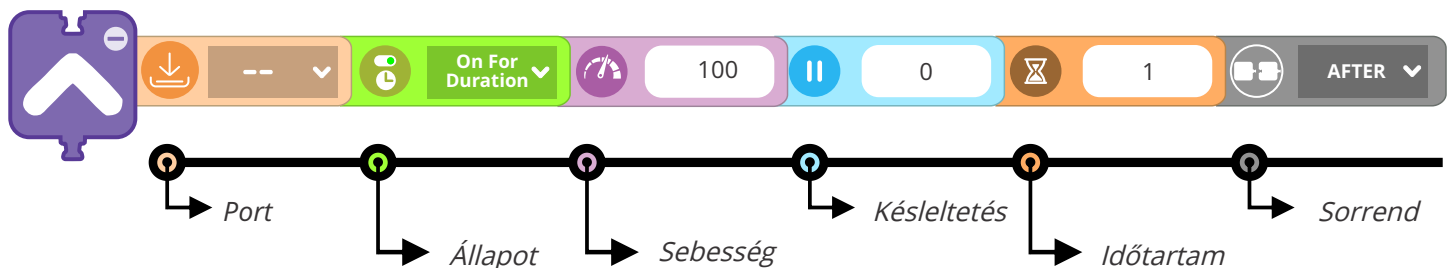


Balra



Jobbra

Lent láthatsz egy példát az előre blokk kiterjesztett beállításával és értékeivel. Ez ugyanígy működik a többi navigációs bloknál.



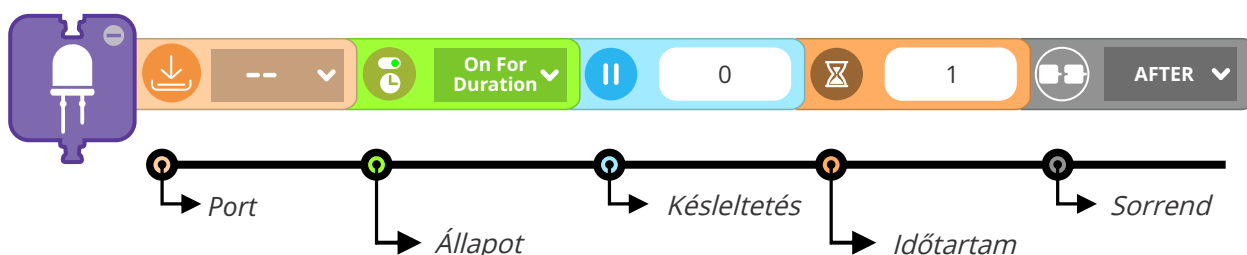
A navigációs blokkok ugyanolyan beállításokkal rendelkeznek, mint a motor blokk. Emlékezz, hogy mindkét motor kapcsolónak I pozícióra kell legyen állítva, hogy a navigációs blokkok helyesen mozgassák a robotot.

LED blokk

A világító dióda (Light Emitting diode - LED) egy fényforrás, ami a kristályokon keresztül haladó elektromosságot használja a fény generálásához. Mivel a LEDek a leghatékonyabbak, biztonságosabbak és keveset fogyasztanak, számos helyen használják őket a robotikában.

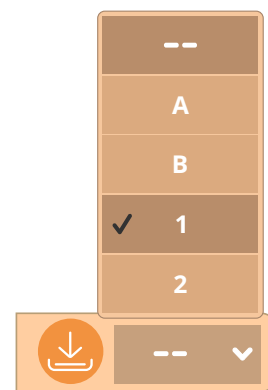


Lent láthatod a LED blokkot kiterjesztve, bemutatva ezzel az alapbeállításait és értékeit.

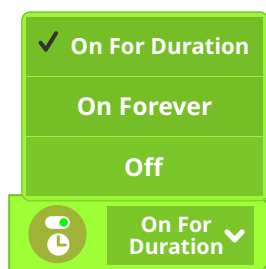


Add meg a Portot, amihez a motor csatlakoztatva van a MINI 2.0 kontrolleren. Az alapbeállítás szerint ez nincs beállítva, ezért nyomj a doboz részre és válaszd ki a megfelelő portot. Az elérhető portokat sötétebb színnel jelzi.

A szimulátornak meg kell adva legyen legalább egy LED, hogy használható legyen a port.



Port



Állapot

Állítsd be a LED állapotát. Három opció közül választhatsz: Adott időre bekapcsolás (alapbeállítás): A LED lámpa csak megadott ideig világít, a megadott időtartam szerint (lásd lejjebb). Folyamatos bekapcsolva: Bekapcsolja a LED-et, vagy bekapcsolva tartja, amíg nem kap másik parancsot. Kikapcsolva (OFF): Kikapcsolja a LED-et.



Add meg a Késleltetést másodpercekben, hogy később kapcsoljon be a LED blokk. Ha a beállított érték 0 másodperc (alapbeállítás), akkor nincs késleltetés.



Késleltetés

Add meg az Időtartamot másodpercekben, hogy a LED lámpa meddig legyen bekapcsolva. A LED kikapcsol, ha eltelt a megadott idő.

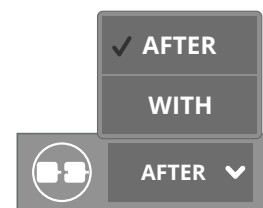


Időtartam

Ez a beállítás csak akkor elérhető, ha az "On For Duration" állapotra van kapcsolva.



A blokkok sorrendje alapbeállítás szerint egymás után következnek. Ahhoz, hogy egyidőben teljesüljenek bizonyos feladatok válaszd a "with" (előző) opciót, így az adott blokk egyszerre kerül végrehajtásra az előző blokkal. Ez a beállítás lehetővé teszi, hogy több blokk is végrehajtsd párhuzamosan.

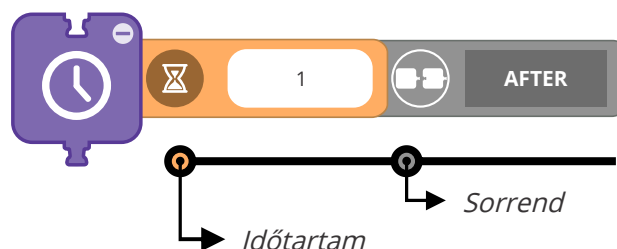


Sorrend

Szünet blokk

A Szünet blokk megállítja a programot egy adott időre. Ez a szünet egy megadható időtartamig tart, ami alapbeállítás szerint 1 másodperc.

A szünet értelem szerűen nem használható párhuzamosan az előző vagy utána következő feladattal.



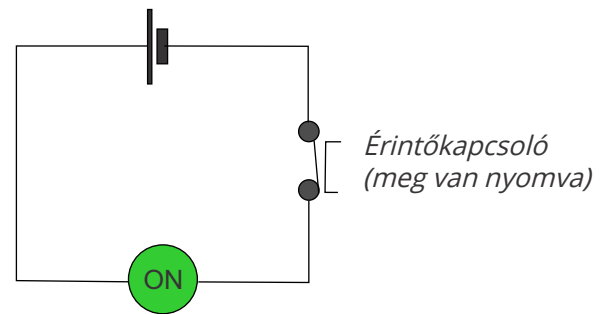
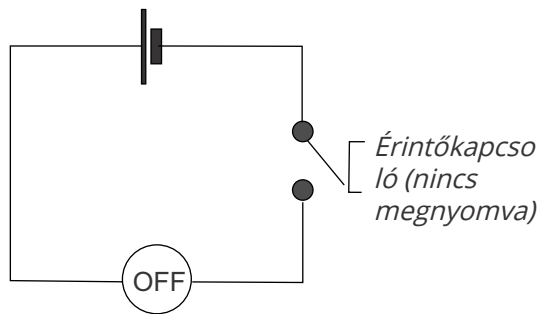
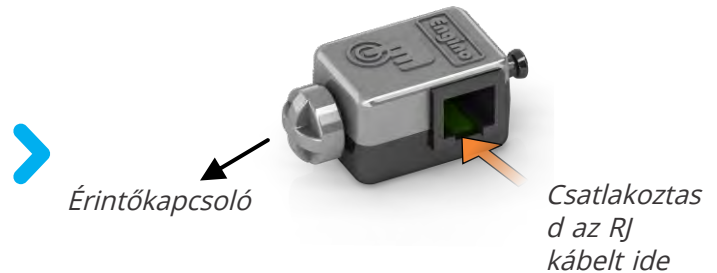
2. Érzékelő blokk

Az érzékelők olyan bemenetek, amik a robot viselkedését irányítják. Pontosabban, ezek szükséges elemei annak, hogy a gépet robotikusnak nevezhessük automatizált helyett. A robot valójában mérlegeli a környezetétől kapott információt, hogy annak megfelelően végezze el a specifikus feladatait a lehető legjobban.



Érintésérzékelő

Fizikailag rákötheted az Engino érintésérzékelőjét az RJ kábel segítségével a kontroller bármelyik portjára. Az Engino® érintésérzékelő valójában egy érintőkapcsoló. Az érzékelő két állapotról ad jelet (TRUE - IGAZ vagy FALSE - HAMIS), ami attól függ, hogy meg lett-e nyomva vagy sem. Ezt a két lenti áramkör szemlélteti.



Amikor a kapcsolót megnyomod az áramkör bezárul és az elektromosság keresztül megy rajta és az érzékelő TRUE értékre vált.
Amikor a kapcsoló nincs megnyomva az értéke FALSE, mert nem folyik keresztül az elektromosság az áramkörön.

Állítsd be, hogy az érintésérzékelő melyik porton csatlakozik a kontrollerhez.



Válaszd ki, hogy ellenőrizze az állapota TRUE vagy FALSE. Az alapállapota TRUE.

Az érzékelő nem használható a folyamatábrán, mint önálló akció block. Pontosabban az érzékelő blokk egy bemenet logikai értékkel. Ha az érzékelő állapota megegyezik a blokk állapotával, akkor a bemenet TRUE, ha nem egyezik, akkor FALSE.

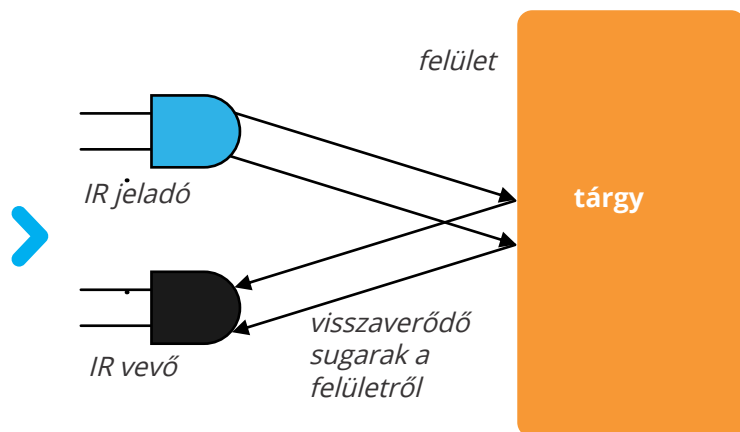
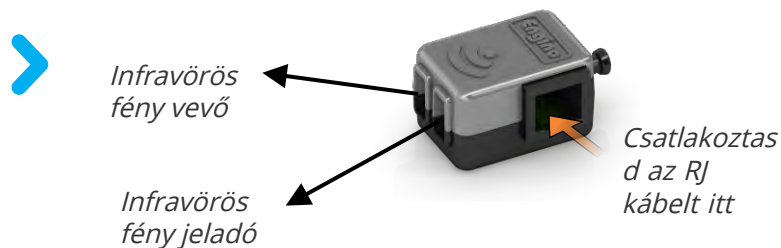
Így az érzékelőt együtt kell használni egy irányító utasítással, mint az IF, WHILE stb., hogy megmondja a robotnak mi történjen, ha a feltétel teljesül (igaz) vagy ha nem (hamis). Olvasd el a következő részeket az Irányító blokkokról és kezelőkről, hogy megtanuld, hogyan kell az érzékelő blokk logikai értékeit használni.

Infravörös (IR) érzékelő

Csatlakoztathatod az Engino® Infravörös érzékelőt az RJ kábelen keresztül a controllerhez. Az Engino infravörös érzékelő egy aktív IR érzékelő, ami két részből áll: az infravörös jeladóból (forrás) és az infravörös vevőből (érzékelő). Az Infravörös érzékelő használható tárgyak érzékelésére és a fekete és fehér felületek megkülönböztetésére.

Az IR adóvevő infravörös jelet ad magából közel az elektromágneses spektrum infravörös tartományában (700-1400nm). Ezek a hullámhossza az emberi szem számára láthatatlanok.

Amikor egy jel elér egy tárgyat, egy része visszatükröződik. Az IR vevő egy fotótranzisztor, ami képes érzékelni az IR sugárzást. Így az érzékelő állapota TRUE - Igaz lesz, ha visszaverődik rá a sugárzás egy tárgyról és FALSE - hamis, ha nem verődik vissza sugárzás rá.

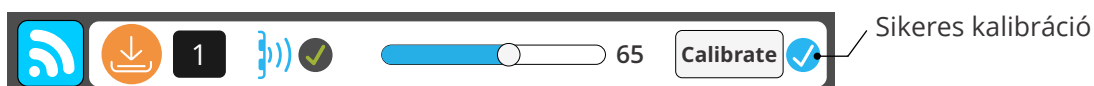


Infravörös érzékelő kalibrálása

Az IR érzékelőt be kell állítani, hogy mi az optimális érzékelési hatótávolsága, így a robot "tudni" fogja, hogy mikor kell cselekednie TRUE(igaz) vagy FALSE(hamis) szituáció esetén. A kalibrációt az élő adás panel alatt lehet teljesíteni.

A TRUE(igaz) állapot aktív az IR érzékelőben ha fehér színt érzékel és a távolság kisebb vagy egyenlő a megadott érzékelési hatótávnál.

A FALSE(hamis) állapot aktív az IR érzékelőben, ha fekete színt érzékel és a távolság nagyobb, mint a megadott érzékelési hatótáv.

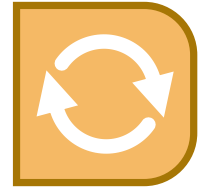


A tárgyak érzékeléséhez: Helyezd az IR érzékelőt pont olyan működési távolságra, ahol az érzékelő működni fog. Tartsd a modellt stabilan és kattints a kalibráló gombra. A kalibráció sikeresen teljesült, ha a "✓" jel megjelenik a kalibráló gomb mellett.

A vonal követéshez: Helyezd az IR érzékelőt fehér felületre, pontosan olyan működési távolságra, ahol az érzékelő működni fog. Végezd el a kalibrációt a fent leírt módon.

*A kalibrációt manuálisan is el lehet végezni, ha a csíkot a kívánt érzékelési szintre állítod.

3. Vezérlő Blokkok

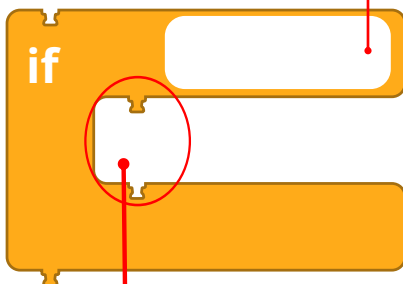


Ebben a kategóriában találsz azokat a blokkokat, amelyek vezérlő parancsokat készítenek. Az alap feltételes utasítások, mint az IF(HA), WHILE(AMÍG), REPEAT (ISMÉTLÉS) és UNTIL (EDDIG), segítenek megoldani kifinomultabb és valóságosabb feladatokat. Az érzékelőktől kapott bemeneti adatok segítségével futtasd a blokkokat, amiket a feltételes utasításba pakoltál.

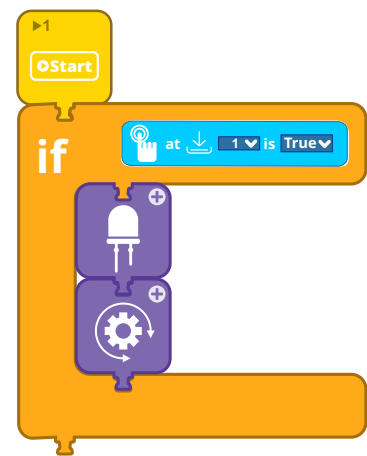
IF(HA):

Az IF blokk aktiválódik, ha az érzékelő megadott állapota teljesül (Igaz vagy Hamis). Például, ha az érintésérzékelőt megnyomják a LED bekapcsolódik és a motor forogni kezd. Ha az érzékelő aktiválva lett másodjára, mielőtt végigfutott a blokkok sorozatán, a sorozat nem indul újra.

Rakd az érzékelő ikon blokkját ide, hogy megadd melyik állapotot kell figyelnie

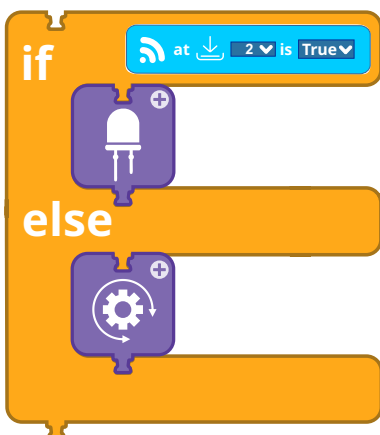


Rakd az akció blokkokat ide, hogy megalkossd a sorozatot, aminek le kell futnia, ha a feltétel teljesül.



IF/ELSE (HA/EGYÉBKÉNT)

Az IF/ELSE blokknak két része van. A sorozat az első részen aktiválódik, ha a megadott feltétel teljesül (például, ha Igaz lesz az állapot, mikor hozzáérnek az érzékelőhöz), ugyanúgy, mint az IF blokk. Ha a feltétlen ellentéte teljesül (például Hamis, tehát az érintésérzékelőt nem nyomták meg) a robot azt a blokk sorozatot fogja teljesíteni, ami a második felébe lett téve.

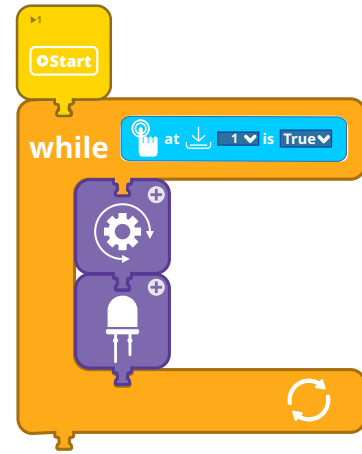


Ebben a példában a LED bekapcsol, ha az IR érzékelő állapota TRUE lesz (pl.: érzékel egy tárgyat). Ellenkezőleg, a motor forog.

A blokkok végrehajtása az ELSE részen megszakad, ha a feltétel teljesül, függetlenül attól, hogy milyen állapotban/folyamatban vannak és a program elkezd az IF részben lévő blokkok futtatását.

Amíg:

A WHILE (AMÍG) blokk aktív, amíg az érzékelőnek előre megadott feltétel teljesül (Igaz vagy Hamis), így futtat egy alprogramot folyamatosan ismételve. Amikor a feltétel nem teljesül tovább, a WHILE blokkban lévő folyamat azonnal megáll és a program halad tovább a következő akció blokkra. Például, amíg az érintésérzékelő nyomva van, a motor óramutatóval megegyező irányba forog és a LED világít (egészen addig, amíg az érzékelőt el nem engedik).

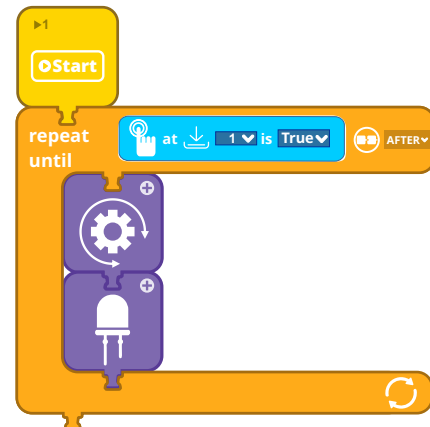


Várakozás eddig:

Ez a blokk várakozó módba állítja a programot, amíg a megadott feltétel nem teljesül. Például, a program nem halad tovább, hogy felkapcsolja a LED lámpát, amíg az érzékelőt meg nem nyomják.

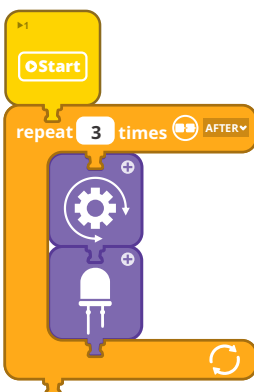
Ismétlés eddig:

Az ismétlő blokk folyamatosan ismétli magát és csak akkor áll meg, ha teljesül a feltétel. Az ismétlés megszakad, ha a feltétel teljesül és a program továbbhalad a következő akció blokkra.

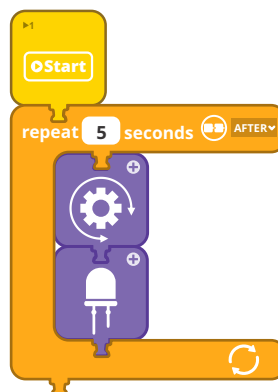


Ismétlés:

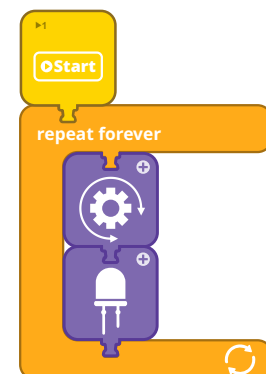
Három fajta ismétlő blokk van, amiket nem az érzékelők vagy különböző előre megadott forgatókönyv szerinti input irányít. A "repeat times" (ismétlés ennyiszor) blokk megadja a lehetőséget, hogy adott számú ismétlés teljesüljön, míg a "repeat seconds" (ismétlés másodpercig), ismételi a megadott időtartamig. Végül a "repeat forever" (ismétlés örökké) blokk hasznos lehet ha megállás nélkül szeretnénk ismételtetni egy folyamatot. Az utóbbinak nincs csatlakozó pont az alján, mert nem lehet alá blokkot hozzáadni.



Futtatás háromszor



Futtatás öt másodpercig



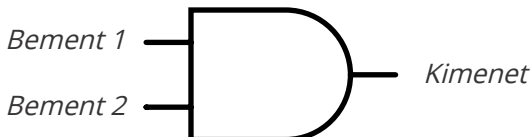
Végtelen lejátszás

4. Operátor Blokkok



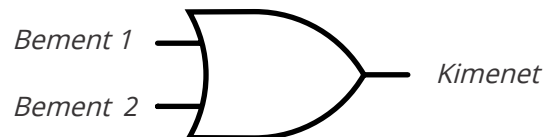
Ebben a kategóriában a logikai operátorokat találod, amik hasznosak mikor egy akcióra van szükség az esetek kombinálásához. Általánosságban az operátorok mondják meg a számítógépnek, hogyan változzon és haladjon tovább az információ.

A logikai operátorok valószínűleg a legfontosabb operátorok és széleskörűen használják őket a programozási nyelvekben. Kombinálni tudnak két bemeneti információt, hogy döntést hozzanak. Tegyük fel, hogy szeretnénk két érzékelő eredményeit kombinálni, hogy parancsot adjunk a robotnak. Aztán az AND (ÉS) és az OR (VAGY) logikai operátorok segítenek nekünk megkapni a végeredményt.



AND (ÉS) szimbólum

AND (ÉS) operátor: Amikor mindkét bemeneti érték "igaz", akkor a kimeneti érték is "igaz". Ellenkezőleg, a kimeneti érték "hamis".



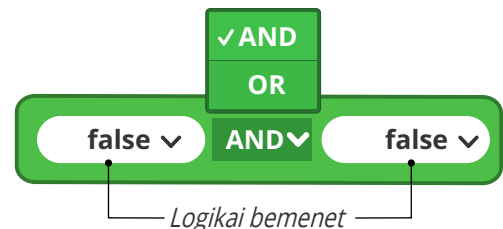
OR (VAGY) szimbólum

OR (VAGY) operátor: Amikor egy vagy mindkettő bemeneti érték "igaz", akkor a kimeneti érték is "igaz" lesz. Csak akkor lesz a kimeneti érték "hamis", ha mindkét bemeneti érték "hamis".

A Logikai operátor blokkok

A KEIRO szoftver Logikai operátorának két bemeneti része van, amik logikai értéket tartalmaznak, TRUE (IGAZ) vagy FALSE (HAMIS) (alapbeállítás). A középső dobozra klikkelve a felhasználó kiválaszthatja az AND/ OR (ÉS/VAGY) operátort..

Emlékezz vissza, hogy az érzékelő blokk is valójában egy logikai érték, ha betesszük ebbe részbe, megadja az értéket. Tehát, ha mindkét helyre érzékelő blokk van betéve, a logikai operátor kimenete ezen a két bemeneten alapul.



Logikai bemenet



Tegyük fel, hogy van egy robotikus járművünk, két infravörös érzékelővel, amikkel az akadályokat érzékeli. Amikor nem érzékel akadályt, szeretnénk utasítani, hogy haladjon előre. Tehát ellenőriznie kell, hogy mindkét érzékelő hamis állapotban van-e. Az előre blokk futtatva lesz, amíg ez a két feltétel teljesül.

5. Függvények

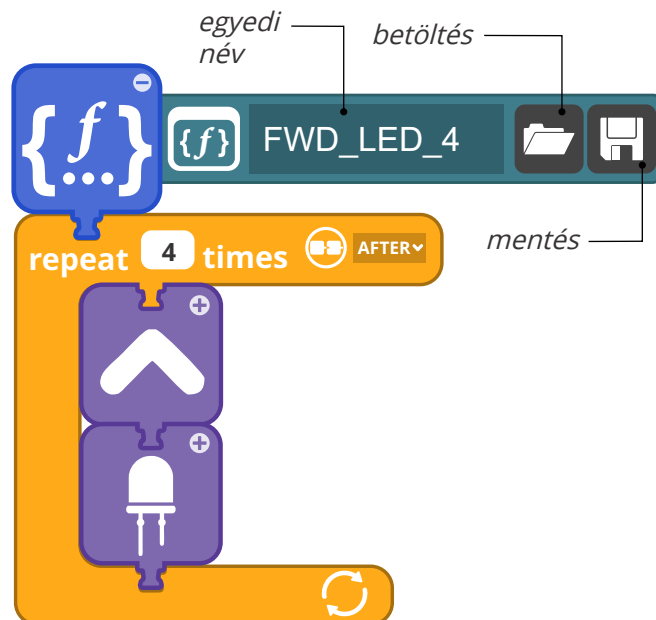
Egy függvény a program alprogramja, ami egy meghatározott parancssort tartalmaz. Alapvetően egy rövid program, amit meghívhat és futtathat a fő program, akár sorozatosan vagy párhuzamosan is. A függvények hasznosak a folyamatábrában használt blokkok számának csökkentésére, különösen azokban az esetekben, amikor hosszú parancs sorozat végrehajtása szükséges több alkalommal.



Függvény beállítása

Ahhoz, hogy egy új függvényt készítsünk fogd meg a "set function"(függvény beállítása) blokkot és helyezd a folyamatábrára. Ez a blokk nem tud csatlakozni a többi blokkhoz, mert ez a kezdő blokkja egy új parancssornak.

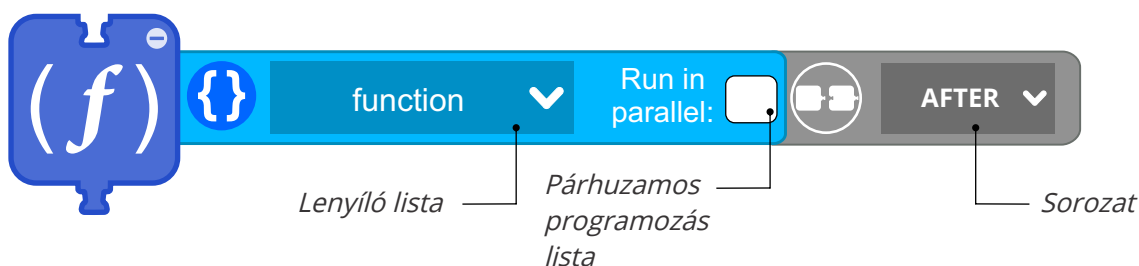
Készíthetsz egy alprogramot a "set function" blokk alá, bármilyen más akció blokkot használva (kimenetek, érzékelők, vezérlők stb.). Továbbá elnevezheted megfelelően, hogy könnyen azonosítható legyen, különösen akkor, ha több függvényt is beállítasz. Egy függvény tárolható a PC-n/Tableten és importálható bármikor a save és load ikonokat használva.



Egy függvény parancsok alprogramját tartalmazza

Függvény használata

Egy függvény meghívásához a felhasználónak a "Use function" (függvény használata) blokkot kell elhelyeznie a sorba egy folyamatábrában. Bizonyos esetekben, mikor több függvény van beállítva, a felhasználó kiválaszthatja a függvényt egy lenyíló listából.



Sorozatos mód:

Az alap mód a függvények futtatásához a sorozatos mód. Ez az, amikor a program odaér a "use function" blokkhoz és teljesíti a függvénybe helyezett feladatokat. Mikor minden feladat véget ért a függvényben a következő blokkra tér rá a folyamatábrában.

"Párhuzamosan futtatás" mód:

A párhuzamos mód engedélyezhető a kipipálható részre kattintva. Ebben a módban a program elkezd végrehajtani a feladatokat a függvény blokkban és azonnal tovább lép végrehajtani a folyamatábrában lévő következő blokkot. Ezáltal az alprogram feladatai a függvényen belül párhuzamosan futnak a folyamatábra többi részével.

Folyamatábrás programozás

Lent láthatod a példát a folyamatábrára leírásokkal a főbb funkcióiról. Egy program készítéséhez csak fogd és húzd a blokkokat a Blokk sorból (a képernyő bal oldalán) a "Start" ikon alá a fő ablakban. Miközben egy akció blokkot közel teszel egy lehetséges kapcsolódási ponthoz, a terület sötétebben jelenik meg. Kattints a "+" jelre a blokkon, hogy átállítsd a tulajdonságait minden akció blokknak, mint például a használt portot, időtartamot stb. Amikor a program elkészül, kattints a "Send program" (program küldése) gombra a menüben és a program továbbításra kerül a MINI controllerbe.

The image shows a programming interface with several callout boxes explaining features:

- Közelítés** (Zoom in), **Alap méret** (Reset), **Távolítás** (Zoom out) icons at the top.
- Callout 1: "Azok a blokkok, amik nem csatlakoznak a folyamatábrához fakóbb színűek és figyelmen kívül lesznek hagyva. Kivételek a 'Set Function' (függvény beállítása) blokk." (Blocks that do not connect to the flowchart are dimmer and will be ignored. Exceptions are the 'Set Function' block.)
- Callout 2: "Nézd meg/rejtsd el a tulajdonságait bármelyik blokknak a +/- jelekre kattintva. Készíthetsz módosításokat vagy egyedivé szabhatod a feladatokat." (View/hide the properties of any block by clicking the +/- signs. You can make modifications or make the tasks unique.)
- Callout 3: "Csatlakoztasd a blokkokat egymás alá, hogy egy parancs sorozatot készíts. Ha egy blokk nem része a sorozatnak, nem lesz része a programnak sem." (Connect the blocks one below the other to create a sequence of commands. If a block is not part of the sequence, it will not be part of the program either.)
- Callout 4: "Helyezd a blokkokat a kukába, hogy kitöröld őket." (Put the blocks in the trash to delete them.)

The interface includes a 'Start' button, a block palette, an 'if' block with a condition '1 is True', and 'Duplicate'/'Delete' buttons.



Használhatod az egeredet, hogy több információt tudj meg vagy változtass dolgokon a következő módokon:

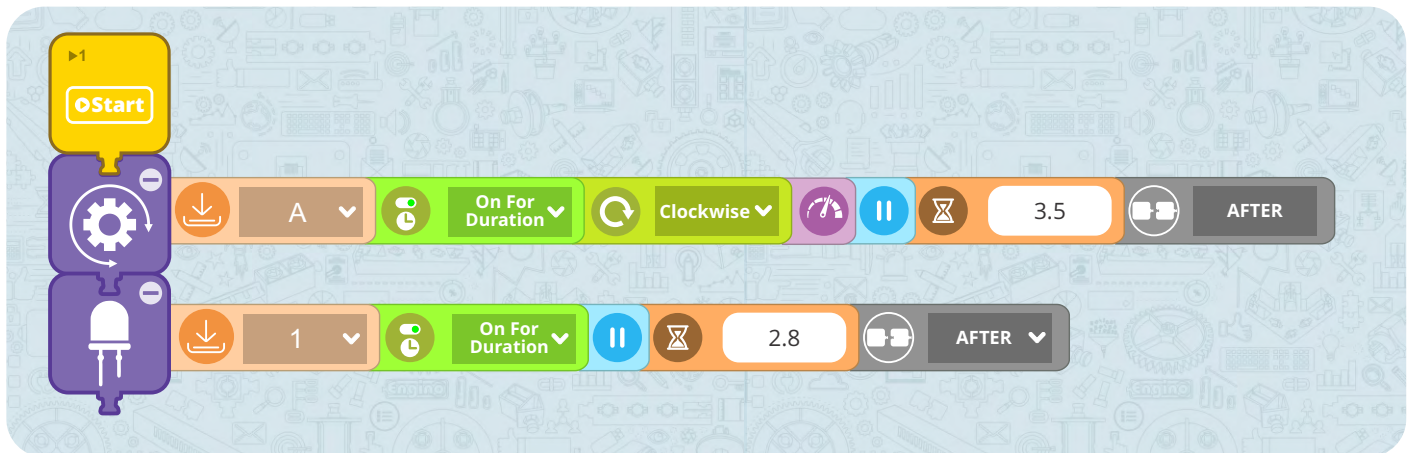
- Helyezd a kurzorod az ikonokra és blokkokra, hogy elolvashasd a leírásukat vagy velük kapcsolatos tippeket.
- Kattints a "+" jelre a folyamatábra blokkjain, hogy megnézd vagy módosítsd a tulajdonságaikat.
- Kattints a "-" jelre, hogy elrejtsd a tulajdonságokat.
- Kattints jobb egérgombbal a blokkra, hogy lemásold vagy töröld egyesével vagy több blokkot egyszerre.

Kód példák

1. Példa - Egyszerű programozás motorral és LED-del

Ez a példa egy motor és egy LED használatával történik az A és 1-es portra csatlakoztatva. A program először forgatja a motort 3,5 másodpercig óramutatóval megegyező irányba, majd a LED bekapcsol 2,8 másodpercre. A program csak egyszer kell végig fusson.

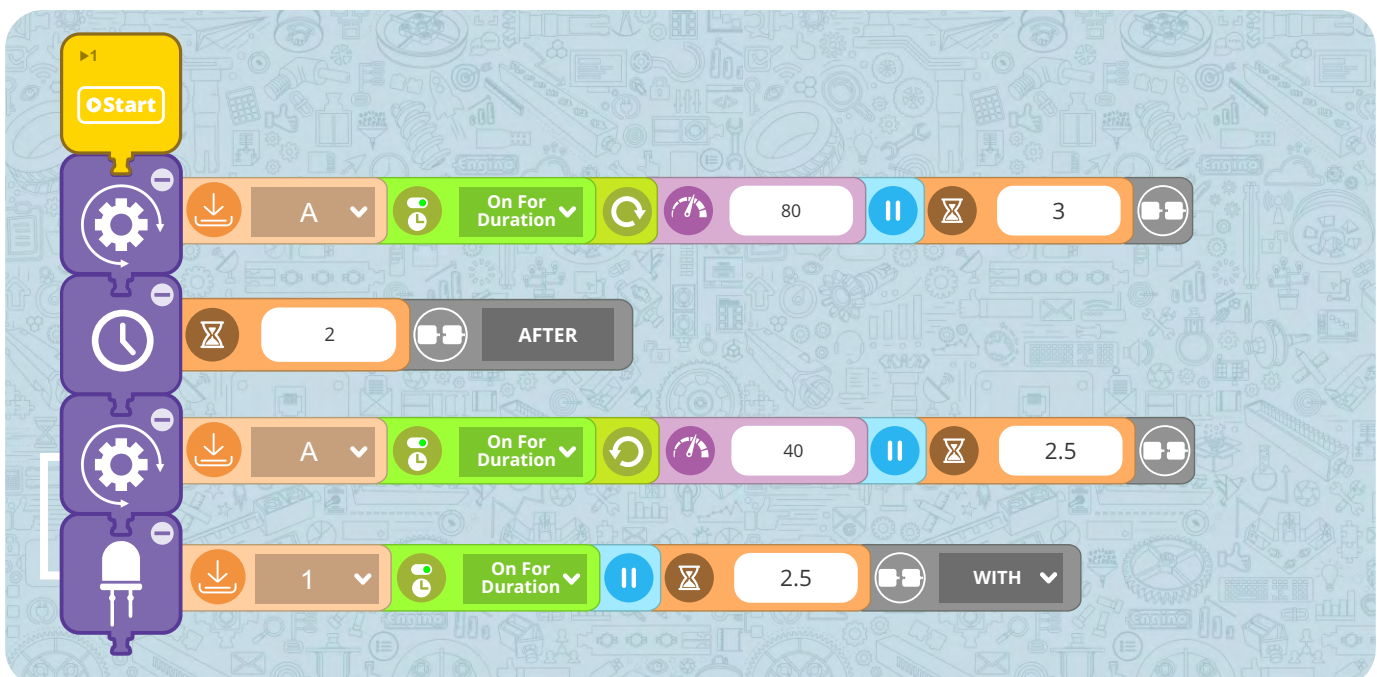
- Használd az adott ideig üzemelő állapotot
- Állíts be az időtartamot
- Adj meg időtartam tartományokat számokkal (lebegőpontos)



2. Példa - Állíts be egyidejű feladatokat és Szünet blokkot

Ez a példa egy motor és egy LED használatával történik az A és 1-es portra csatlakoztatva. A program először forgatja a motort 3 másodpercig 80%-os sebességgel. Majd a program 2 másodpercre szünetet tart. Utána pedig a motor 40%-os sebességgel forog tovább és a LED bekapcsol 2,5 másodpercre.

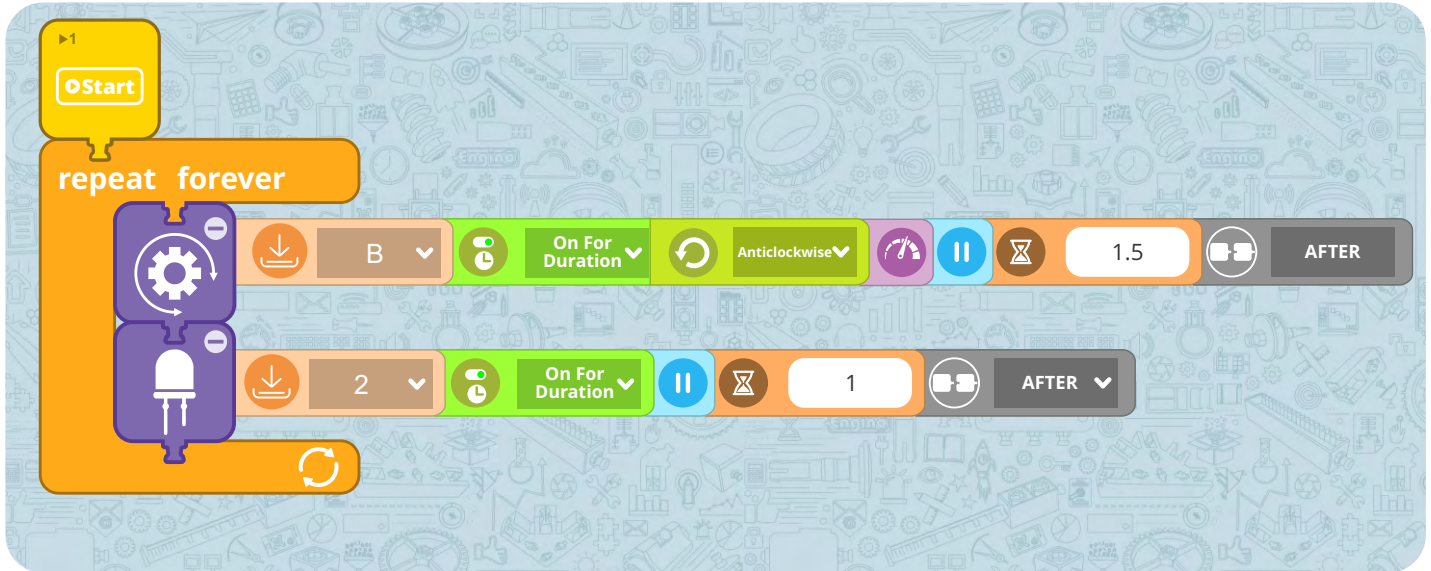
- Változtasd meg a motor sebességét
- Használd a Szünet blokkot
- Szinkronban futó feladatok programozása



3. Példa - A program futtatása ismétlődve

A következő példa egy motor és egy LED használatával történik a B és 2-es portokra csatlakoztatva. A feladat, hogy motor óramutatóval ellentétes irányba forogjon 1,5 másodpercig, majd kapcsoljon be a LED 1 másodpercre és ez a folyamat ismétlődjön magától.

- *Forgás irányának változtatása*
- *Futtasd a programot egy végtelen ismétléssel*



4. Példa - Használd az "IF" feltételes utasítást

Ez a példa egy motor, egy LED és egy érintésérzékelő használatával történik, amik a B, 1-es és 2-es portokra vannak csatlakoztatva. Amíg az érintésérzékelő nincs megnyomva (állapota FALSE), a motor az óramutatóval megegyező irányba forog. Ha az érzékelőt megnyomják (állapota TRUE lesz), a LED kapcsoljon be 1,5 másodpercre és a motor forduljon az óramutatóval ellentétes irányba 1 másodpercig, de 0,5 másodperces késleltetéssel.

- *Használd a folyamatosan bekapcsolva állapotot*
- *Az érintésérzékelő használata*
- *Az "IF" (HA) feltételes utasítás*
- *A késleltetés használata*



Felépítés és programozás

A robotikának két fő szempontja van: az építési rész és a programozási rész. Mindkettővel kísérletezhetsz a ventilátor modellt használva. Tanuld meg módosítani az építményed, hogy hatékonyabb legyen! Majd programozd a modellt manuálisan a kontrolleren és használd a blokkokat a KEIRO szoftverben.

Felfedezés:

- Mik azok a bemenetek és kimenetek?
- Hogyan lehet konstruktívan javítani a roboton?
- Hogyan kell manuálisan programozni egy robotot?
- Mi az a folyamatábra?

A bonyolultság szintjei ★★★★★

Szükséges eszközök:

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A).
- PC vagy okostelefon letöltött és telepített KEIRO szoftverrel.

Folyamat:

1. Keresd ki az útmutatót a 43-45. oldalon és építsd meg a Ventilátor modellt. Győződj meg róla, hogy minden kábelt csatlakoztattál a megfelelő porthoz. Csavard köré a kábeleket a többi résznek, hogy ne legyenek útban. Helyezd be a 3db AAA elemet a controller hátuljába és indítsd el az eszközt az On-Off gombbal!



2. Kapcsold be mindkét motort és állítsd I. pozícióra őket. Az O pozícióban a motor ki van kapcsolva és a II. pozícióban fordított az irány. Nyomd meg a motor gombot (parancs) a kontrolleren, ahogy az 1. gyakorlatban mutatva van és írd le a tapasztalt eredményt (kimenet).

3. Javítsd a modelledet építéssel a 2. gyakorlat útmutatóját követve!

4. Kövesd a 3. gyakorlatot, hogy elkészíts egy manuális programot a kontrolleren. Lehet, hogy többször is kell próbálkoznod, mire sikerülni fog.

5. Csatlakoztasd a kontrollert PC-hez vagy okoskészülékhez. Nyisd meg a KEIRO szoftvert és kattints a kapcsolódás gombra.



6. Kattints a "Receive Program" (Program fogadása) gombra és töltsd fel a rögzített sorozatot a KEIRO szoftverbe, így készítve egy folyamatábrát.



7. Módosítsd a folyamatábrát a

4. gyakorlat szerint! Küldd el a programot a controllerre a "Send Program" (Program küldése) gombra kattintva. Mentsd el a programot "fan" néven.

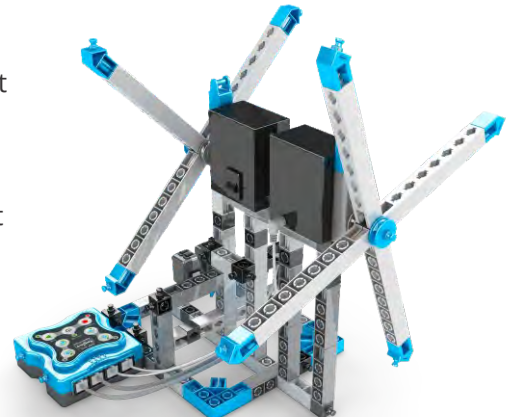


Engino ventilátor modell

1. Gyakorlat: Nyomd meg a gombokat a kontrolleren (bemenet) és írd le a megfigyeléseidet, hogy mi történt a modellel (kimenet)!

Parancs	Kimenet
Motor A pozíciójának váltása I	

2. Gyakorlat: Készíts egy második ventilátort a modelledhez az ellenkező oldalára az első ventilátornak és kösd a második motort a B porthoz. Az építési mintát jobb oldalt láthatod.



3. Gyakorlat: Nyomj a "Program" gombra (a piros fény jelzi, hogy a programot épp rögzíti) és utána nyomd meg a szükséges gombokat a következő sorrendhez: Ÿ Ventilátor A óramutató járásával megegyezően pörög; Ÿ Ventilátor B óramutató járásával megegyezően pörög; Ÿ mindkét ventilátor egyszerre pörög ellentétes irányba;



Nyomd meg a "Program" gombot újra, hogy elmentsd a programot és nyomd meg a "Play" gombot, hogy ellenőrizd helyes-e a parancsok sorrendje. Ha megnyomod és nyomva tartod a futtatás(play) gombot egy kis időre, akkor a program folyamatosan ismétli majd magát.



4. Gyakorlat: A folyamatábrán módosítsd minden akció blokk tulajdonságait, hogy a lent leírt programnak feleljen meg. Adhatsz hozzá több akciót (motor, szünet) a bal oldali menüből: Ÿ Ventilátor A az óramutató járásával megegyezően pörög 2 másodpercig 100%-os sebességen; Ÿ ventilátor B az óramutató járásával megegyezően pörög 2 másodpercig 100%-os sebességen; Ÿ a robot nem csinál semmit 1 másodpercig (használd a szünet blokkot); Ÿ mindkét ventilátor egyszerre pörög azonos irányba 3 másodpercig, az A 100%-os sebességgel, a B pedig 40%-os sebességgel.



Erről tanulunk: ROBOTIKA

Robotikus mozgás és HA utasítás

A robotikus mozgás fontos és bonyolult néha elérni. Számos módja van, hogyan mozoghat egy robot a céljától függően: néhánynak kerekei vannak, vagy sínen mozog, másoknak lábai vannak és megint mások repülnek propellerek segítségével. Kísérletezz a lábat használó mozgást a hatlábú modellel!

Felfedezni:

- Hogyan működik a robotikus lábmozgatás?
- Hogyan kell Szimulátort és a Gamepadot használni?
- Hogyan kell használni az Infravörös érzékelőket a HA feltételes paranccsal.

A bonyolultság szintjei ★★★★★

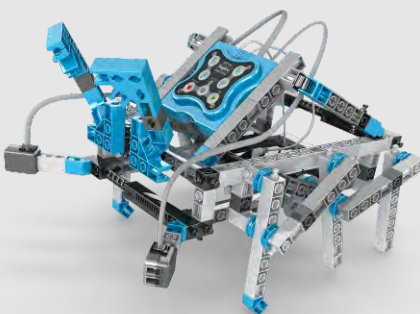
Szükséges eszközök:

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A).
- PC vagy okostelefon letöltött és telepített KEIRO szoftverrel.

Folyamat:

1. Keresd ki az útmutatót a 46-51. oldalon és építsd meg a hatlábú modellt!
2. Csatlakoztasd a kontrollert a PC-hez vagy az okoskészülékedhez. Nyisd meg a KEIRO szoftvert és kattints a csatlakozás gombra. Teljesítsd az 1-es és 2-es gyakorlatokat, hogy megtanuld a Szimulátort kezelni a KEIRO szoftverben.
3. Nyisd meg a Gamepad ablakot a megfelelő gombra kattintva és irányítsd vele a robotodat.
4. Ahhoz, hogy egyenesen előre vagy hátra mozogjon, a lábakkal ugyanabból a pozícióból kell indulniuk. Amikor kerekeket használunk, nincs ilyen probléma, mert az egyenes haladás csak a motorok forgási sebességétől függ. Tuddsz mondani másik problémát, ami a lábak használatával felmerülhet? Írd le őket a 3. gyakorlatban!

5. Kövesd a 4. gyakorlatot, hogy megprogramozd a hatlábút a kétoldali akadályok kikerülésére! Ehhez szükséges lesz az IF (HA) feltételes utasítás blokkra



Engino hatlábú modell

1. **Gyakorlat:** Húzd a perifériákat (motorok és érzékelők) a portokra a Szimulátorban, a hatlábú modell szerint. És rakd össze a blokkokat a kép szerint.



2. **Gyakorlat:** Irányítsd a hatlábút a szimulátort használva a következő módon: Ÿ a hatlábú jobbra fordul egy lábat használva; Ÿ a hatlábú balra fordul egy lábat használva; Ÿ a hatlábú jobbra fordul mindkét lábat használva.

3. **Gyakorlat:** Milyen problémák léphetnek fel, ha kerekek helyett lábakat használatálunk a robotikus mozgáshoz?

.....

.....

.....

.....

4. **Gyakorlat:** Készítsd el a következő programot úgy, hogy a hatlábú kikerüli mindkét oldalon az akadályokat folyamatosan ismételve: Ÿ a hatlábú előre halad folyamatosan; Ÿ ha a bal IR érzékelő akadályt érzékel, a jobb láb vissza kell mozduljon és a bal lábnak előre kell mozdulnia; Ÿ ha a jobb IR érzékelő akadályt érzékel, a bal lábnak vissza kell mozdulnia és a jobb lábnak előre kell mozdulnia.

Helyezd a megfelelő blokkokat az IF (HA) blokkon belülre. Nézd meg a 25. oldalon, hogyan kell beállítani az IR érzékelőt akadályok észleléshez.

AMÍG állítás és logikai kapuk

Az Infravörös érzékelőket gyakran használják a robotikában főleg kétféle módon: Különbséget tenni a fekete és fehér szín között és tárgyak érzékelésére. Próbáld meg mindkét programozási kihívást és tanuld meg beállítani az IR érzékelőket, hogy a WHILE (AMÍG) feltételes utasítást használja és a logikai kapukat.

- Hogyan kell használni az infravörös érzékelőket a WHILE (AMÍG) és IF (HA) feltételes utasításokat?
- Mik azok a logikai kapuk?

Bonyolultsági szint ★★★★★

Szükséges eszközök:

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A).
- PC vagy okostelefon letöltött és telepített KEIRO szoftverrel.
- 4 x fehér A3-s kartonpapír
- Fekete ragasztószalag vagy fekete filctoll
- 4 fehér tárgy könnyű észleléshez

Folyamat:

1. Keresd ki az útmutatót a 38-42. oldalon és építsd meg a Robot modellt! Győződj meg róla, hogy mindegyik kereket ugyanazon oldali portokra csatlakoztatód.

2. Ragaszd az A3-as papírokat össze, hogy egy nagy fehér téglalapot készíts. Rajzolj egy fekete ellipszist, 1,5cm vastag vonallal fekete filcet vagy ragasztószalagot használva (a jobb oldali kép szerint). Letölthetsz egy másik útvonalat a www.enginorobotics.com/teaching_resource/s/examples/ címen.

3. Nyisd meg a KEIRO szoftvert és kövesd az utasításokat a 1. gyakorlatban, hogy elkészítsd a fehér követő programot. Ehhez szükséges lesz két IR érzékelő,

amik a föld felé néznek, lefordítva. A robot a fehér részen fog mozogni és megáll minden kereke a feketén, így követi a vonalat.

4. Olvasd el a 2. gyakorlatot, hogy átalakítsd a Robotot egy tárgy észlelő robottá. Ehhez szükséged lesz két IR érzékelőre, amik előre néznek. Ha elkészítetted a programot, teszteld le úgy, hogy négy fehér tárgyat (olyan magasat, mint amilyen magasan az érzékelők vannak) elhelyezel, hogy egy képzeletbeli négyzetet hozz létre a robotnak, aminek az oldalait követve haladhat (jobb oldali képen láthatod).

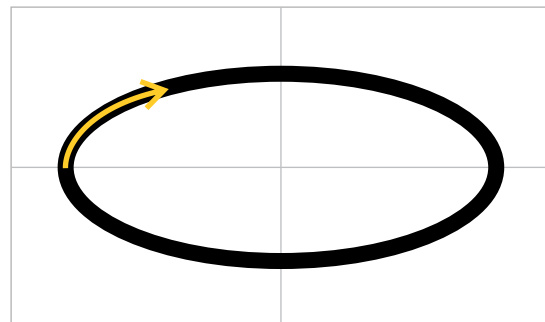


Engino Robot modell

1. Gyakorlat: A számítógépeden vagy okoskészlőre helyezd a megfelelő ikonokat a szimulátorra a Robot modell szerint. Húzd bele a megfelelő akció blokkokat a folyamatábrába, hogy a fehér hatására mozogjon és rögzítsd az útvonalát a feketén: \ddot{Y} amíg az IR érzékelő az 1. porton érzékel fehér színt, addig az A porton lévő motor előre kell forogjon; \ddot{Y} Amíg az IR érzékelő a 2. porton érzékel fehér színt, addig a B porton lévő motor előre kell forogjon.

Ha elhelyezted a megfelelő blokkokat, akkor kalibrálnod kell az infravörös érzékelőket vonalkövetéshez. Olvasd el, hogyan kell a felhasználói kézikönyvben az "Infravörös érzékelő kalibráció" rész alatt a 25. oldalon.

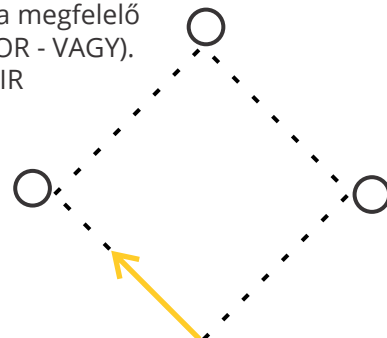
Ezzel a programmal a Robot követi a fekete csíkot. Ahhoz, hogy teszteld a programodat, helyezd a robotot a fekete ellipszisre, amit csináltál és nézd meg, hogy körbe megy-e.



2. Gyakorlat:

Húzd a szükséges blokkokat bele a folyamatábrába, hogy elkészítsd a következő programot: \ddot{Y} A robot előre halad; \ddot{Y} Ha érzékel egy fehér tárgyat az egyik vagy mindkét érzékelője által, a robotnak meg kell állnia; \ddot{Y} 5 másodperc után a robotnak 90 fokba el kell fordulnia jobbra, hogy érzékelje a következő tárgyat.

Tippek: A szög mindig ugyanakkora, ezért az egyetlen program, amire szükséged van a fordulás. Hogy beállítsd mindkét vagy az egyik érzékelőt a tárgyak észlelésére, ki kell választanod a megfelelő logikai kapukat (AND - ÉS vagy OR - VAGY). Ezen kívül kalibrálnod is kell az IR érzékelőket újra, hogy ezúttal a tárgyak érzékelésekor aktiválódjának.



- Hogyan működnek a helyhez kötött robotok
- Mi a REPEAT TIMES feltételes utasítás
- Mi a REPEAT UNTIL feltételes utasítás

Valódi világi példa: Ipari robotkar

A robotokat gyakran használják ipari környezetben, fontos szerepük van a gyártó sorok mellett. A leggyakoribb felhasználási mód a rögzített karok, amik képesek alkatrészeket pakolni egyik helyről a másokra érzékelő technológiákat használva. Kísérletezhetsz egy ilyen szimulált helyzetet a robotkar modellt használva.

Bonyolultsági szint ★★★★★

Szükséges eszközök:

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A).
- PC vagy okostelefon letöltött és telepített KEIRO szoftverrel.
- Ragasztószalag és 1 üres üdítő dobozra

Folyamat:

1. Keresd ki az útmutatót az 52-57. oldalon és építsd meg a Forgókar modellt!
2. A helyhez kötött robot azt jelenti, hogy az alapja fix helyen marad. Kövesd az 1. gyakorlatot, hogy a modellt még stabilabbá tedd forgás közben!
3. Ismétlődő feladatokat precízen elvégezni fontos része a programozásnak. Kövesd a 2. gyakorlatot, hogy készíts egy ismétlődő programot az "ismétlés többször" blokk segítségével!
4. Néha az ismétlésnek folyamatosan kell mennie, amíg az érzékelő feltételei teljesülnek. Kövesd a 3. gyakorlatot, hogy megtanuld, hogyan kell ezt elvégezni az "ismétlés amíg" blokk segítségével!
5. A helyhez kötött robotok gyakran specifikus úton mozognak, hogy teljesítsék a munkájukat. Ehhez IR érzékelő technológia szükséges, így a robot maga tudja pontosítani, hogy merre kell haladnia. Ha az időzítés az egyetlen használt mód, mindegy, hogy mennyire pontos, a robot mozgása hibás lehet sok ismétlés után. Kövesd a 4. gyakorlatot, hogy szimuláld a robot munkarutinját!



Engino Forgókar modell

1. Gyakorlat: Készíts egy egyszerű programot manuális, hogy forgasd a kart. Helyezd a robotot a földre és nézd meg mi történik, ha futtatod a programot! Észre fogod venni, hogy a robot alapja is forog a robottal együtt. Hogy ezt a problémát kiküszöböljük, csatlakoztass egy-egy sárga csigát az alap négy sarka alá. Még több stabilitásért pedig helyezd a csigákat ragasztószalagra. Majd próbáld futtatni a programot ismét!



2. Gyakorlat: A számítógép vagy okoskészülék használatával helyezd az ikonokat a szimulátorra a Forgókar modellnek megfelelően! Húzd a szükséges blokkokat bele a folyamatábrába, hogy elkészítsd a következő programot: Ÿ A forgókarnak fordulnia kell 1 másodpercig 50%-os sebességgel, majd össze kell zárnia a nyitott kezét; Ÿ Ezt a feladatot ismételd meg 5 alkalommal!

Tipp: Használd az "ismétlés többször" blokkot!

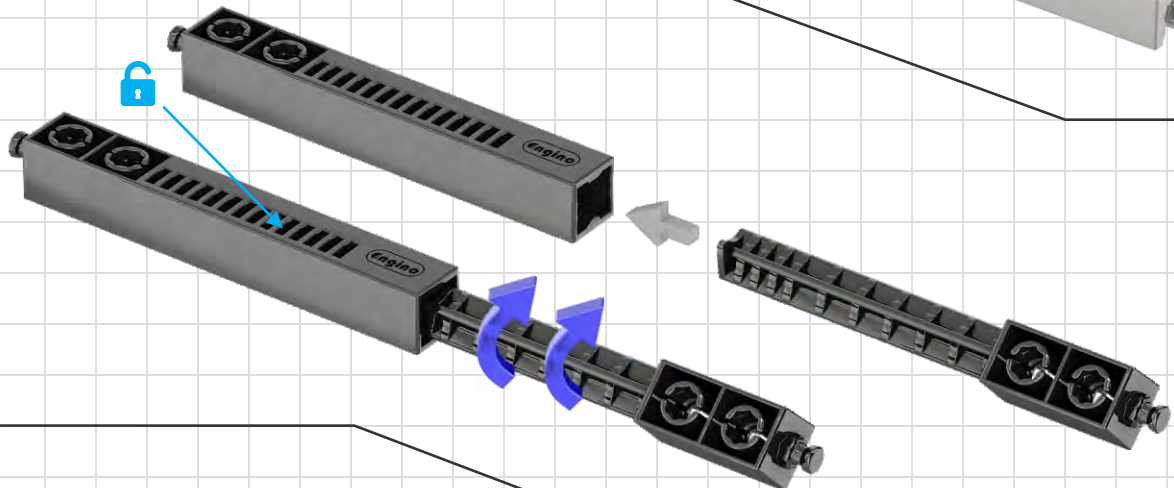
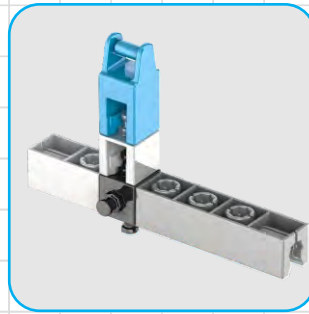
3. Gyakorlat: Készítsd el a következő programot (A forgókar keze nyitva kell, hogy legyen a kezdéskor): Ÿ a forgókarnak fordulnia kell folyamatosan, amíg az IR érzékelő aktiválódik egy tárgy vagy a kezed hatására; Ÿ amikor a modell megáll, ha egy üdítő doboz van a kar IR érzékelője elé helyezve, akkor meg kell, hogy fogja és tartsa.

Tipp: használd az "ismétlés amíg" blokkot a program első feléhez és módosítsd a kart, hogy jobban tudjon megfogni dolgokat, ha szükséges.

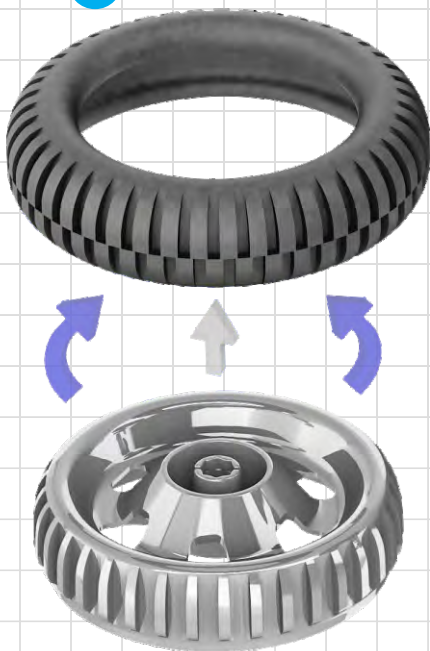
4. Gyakorlat: Helyezd az üdítő dobozt a forgókar kezébe és készítsd el a következő programot: Ÿ a forgókarnak fordulnia kell 180 fokba balra az üdítő dobozt fogva; Ÿ amikor a modell megáll, a kéznek ki kell nyílnia és elengednie az üdítő dobozt; Ÿ ezután a forgókarnak fordulnia kell 180 fokot jobbra (vissza az eredeti pozíciójába); Ÿ ha egy üdítő doboz lett a kar érzékelője elé téve, meg kell tudnia fogni és tartani, majd ismételnie a folyamatot (fordulni 180 fokkal, elengedni az üdítő dobozt és visszafordulni).

Tipp: helyezz két tárgyat (vagy használt a kezed) bármelyik oldalára a középső érzékelőnek, hogy aktiválja.

Építési tippek



tire
L



Kétoldalú
szétszedő eszköz



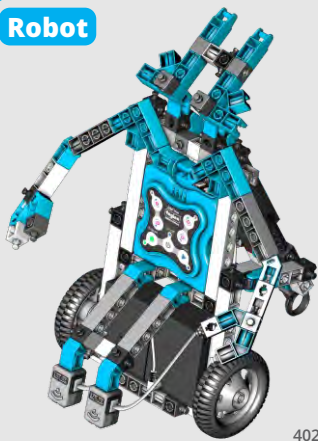
2. LEHETŐSÉG:
Nyomd a lyukba, hogy
kilökd a beszorult
alkatrészt!

1. LEHETŐSÉG:
Szorítsd az eszközt,
hogy szétszedd az apró
alkatrészeket!



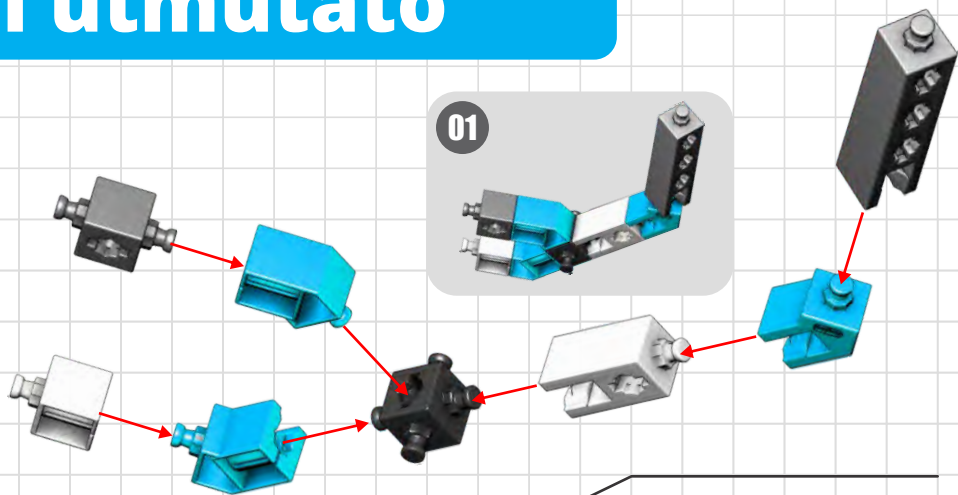
Építési útmutató

Robot

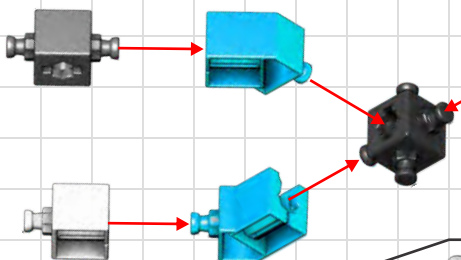
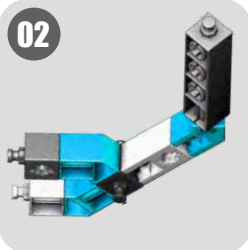


4023

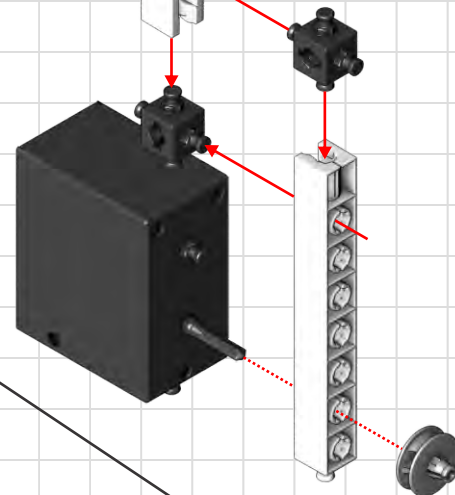
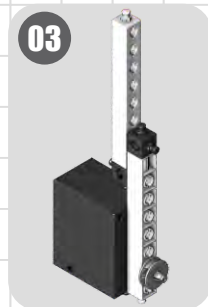
01



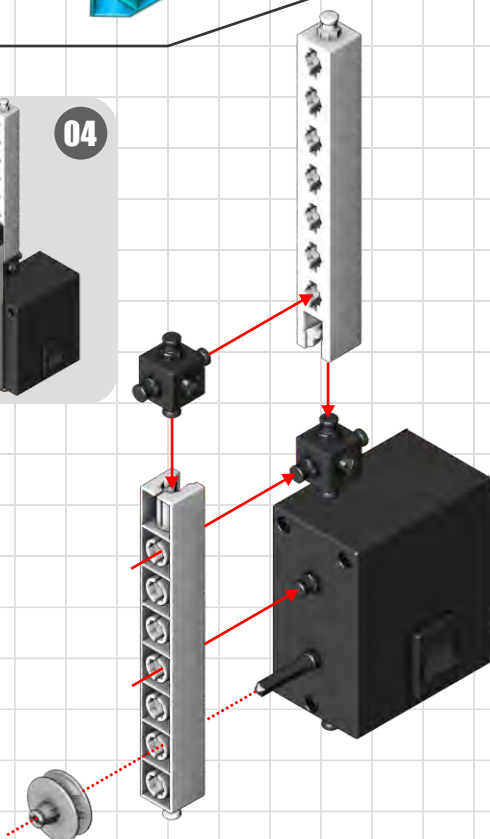
02



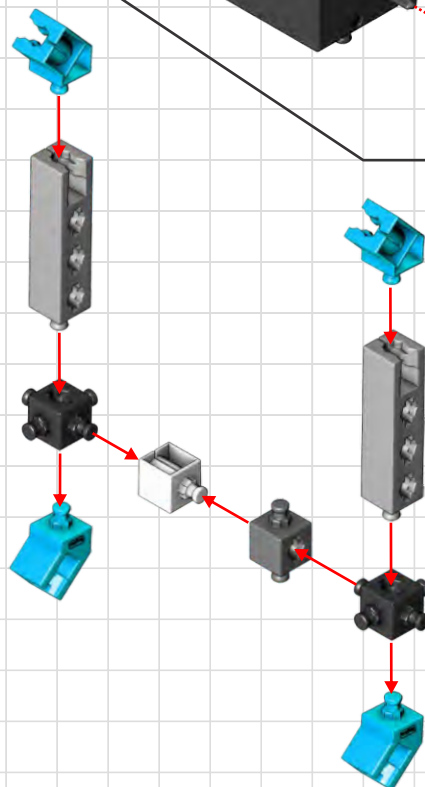
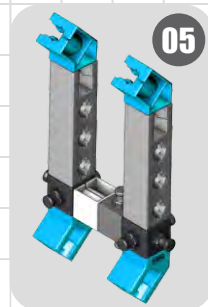
03

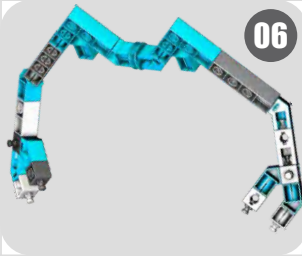


04

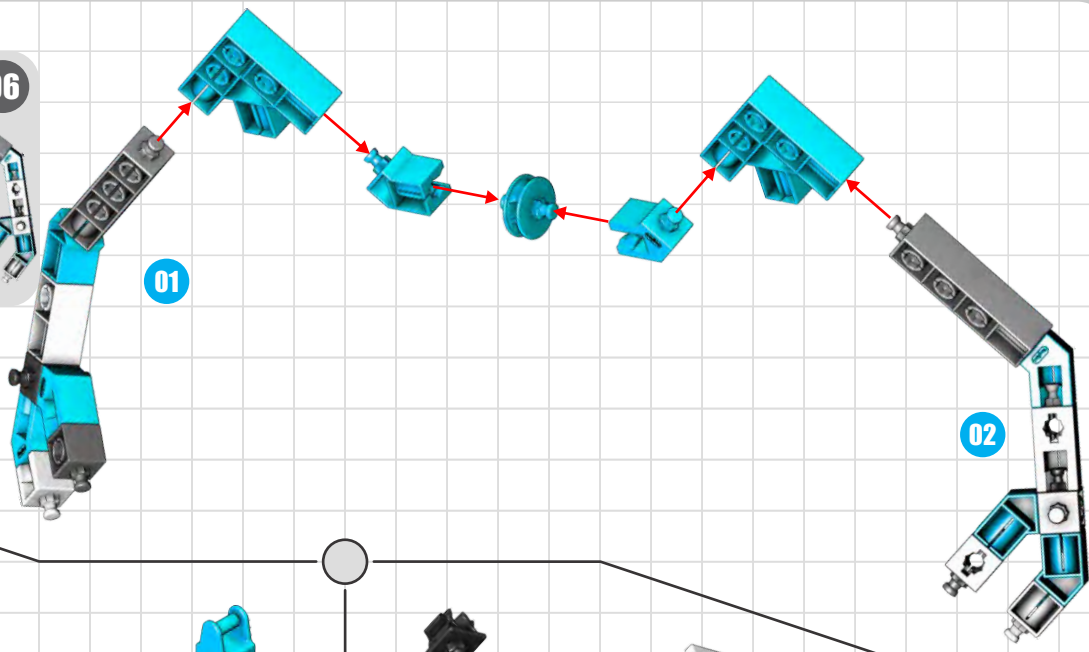


05

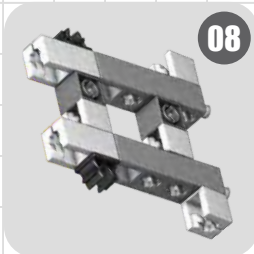




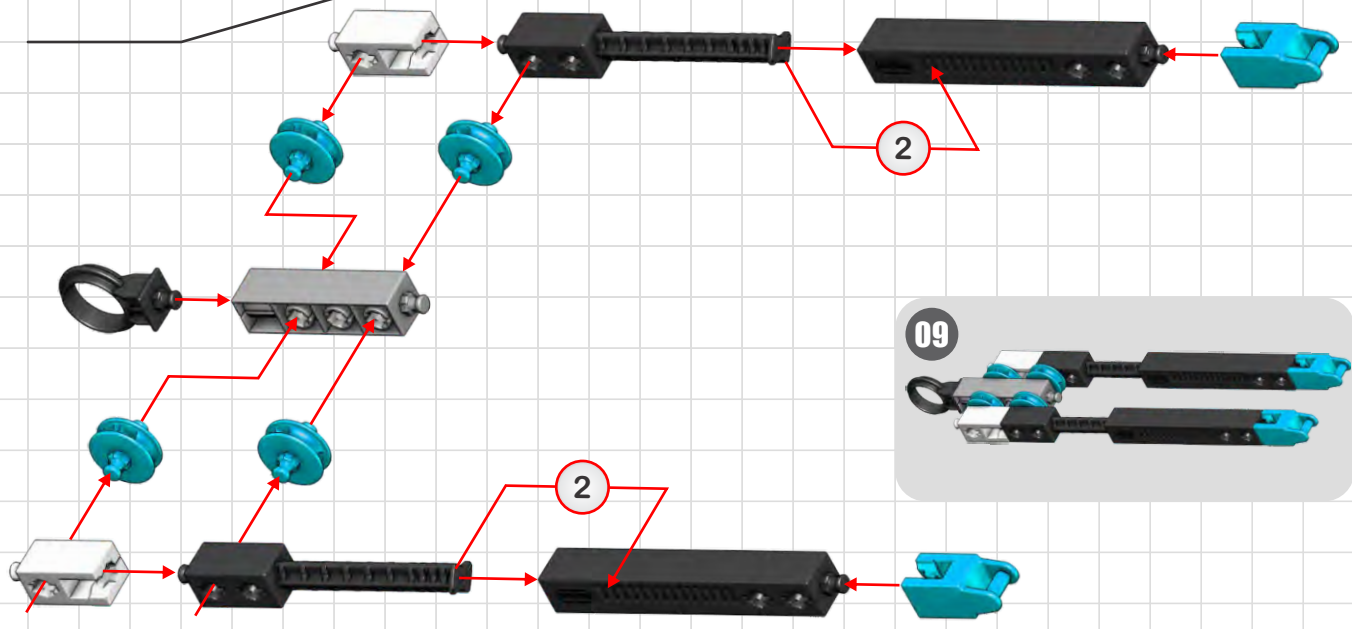
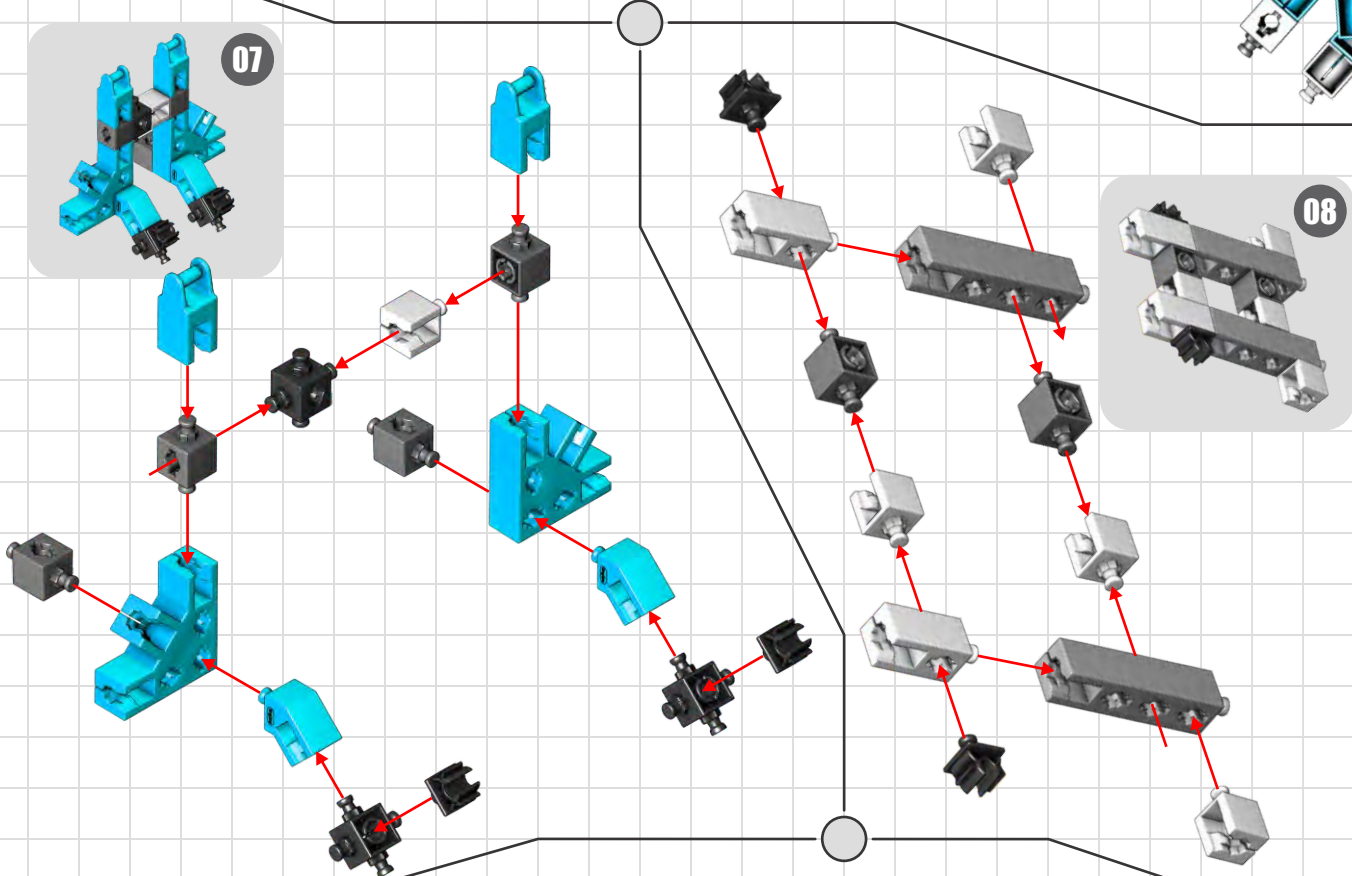
06



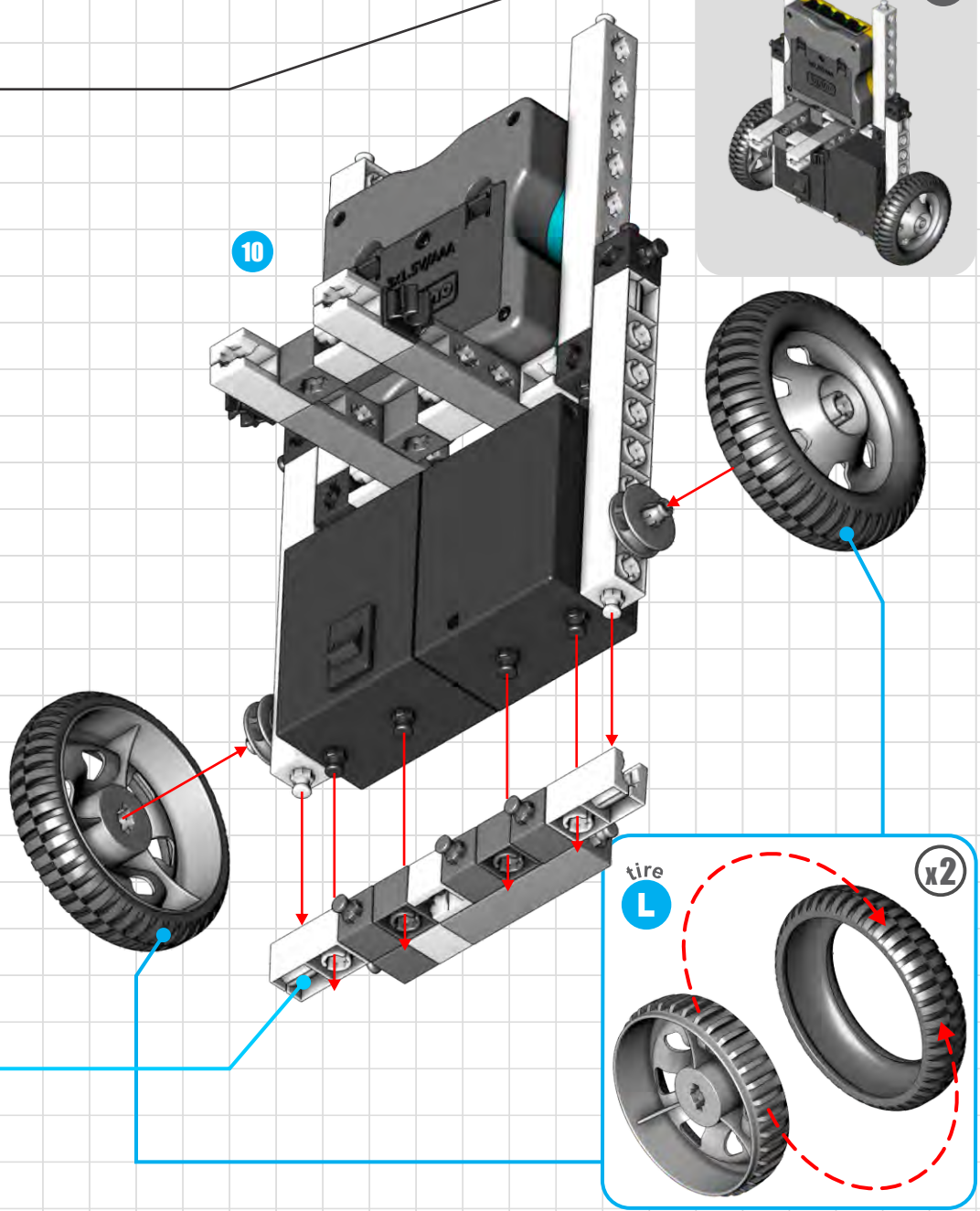
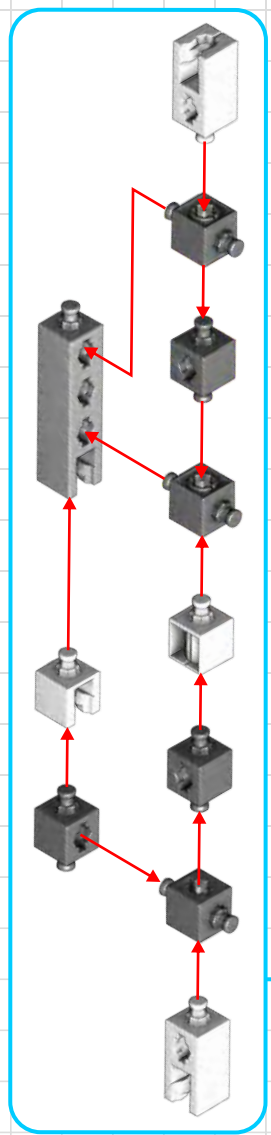
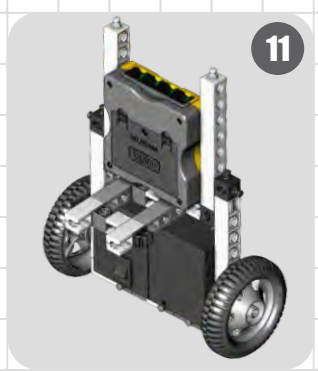
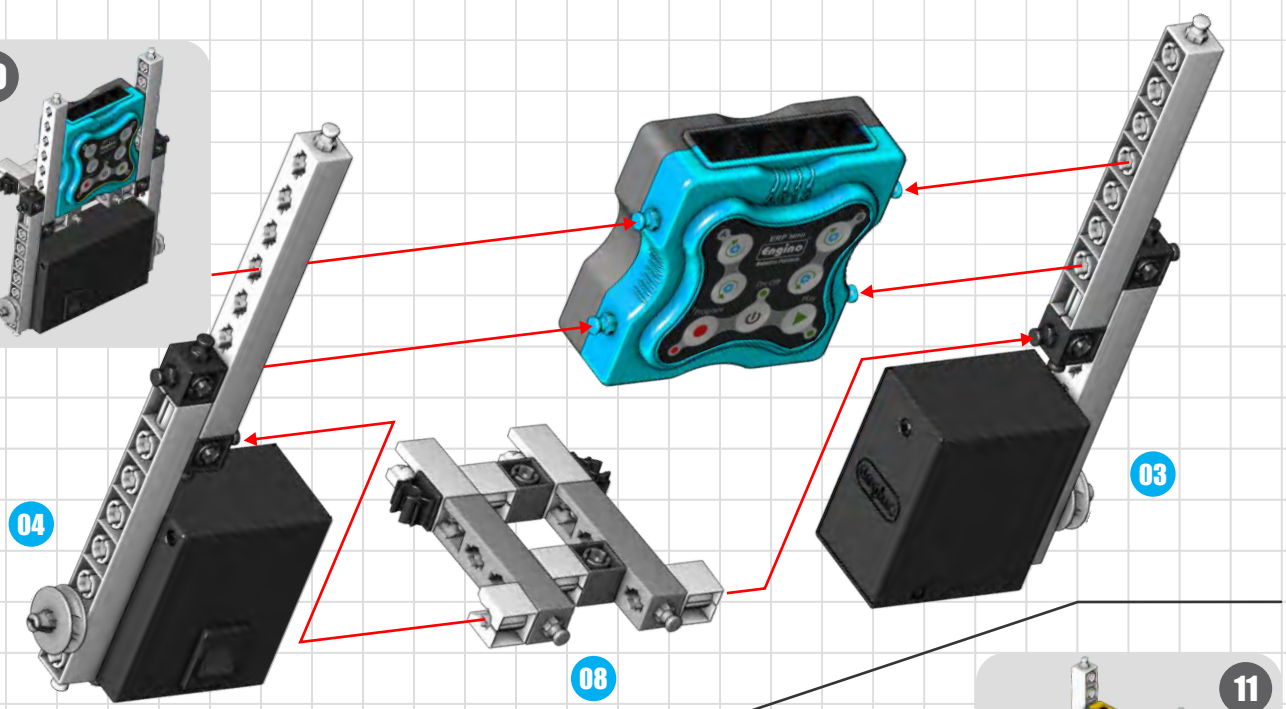
07

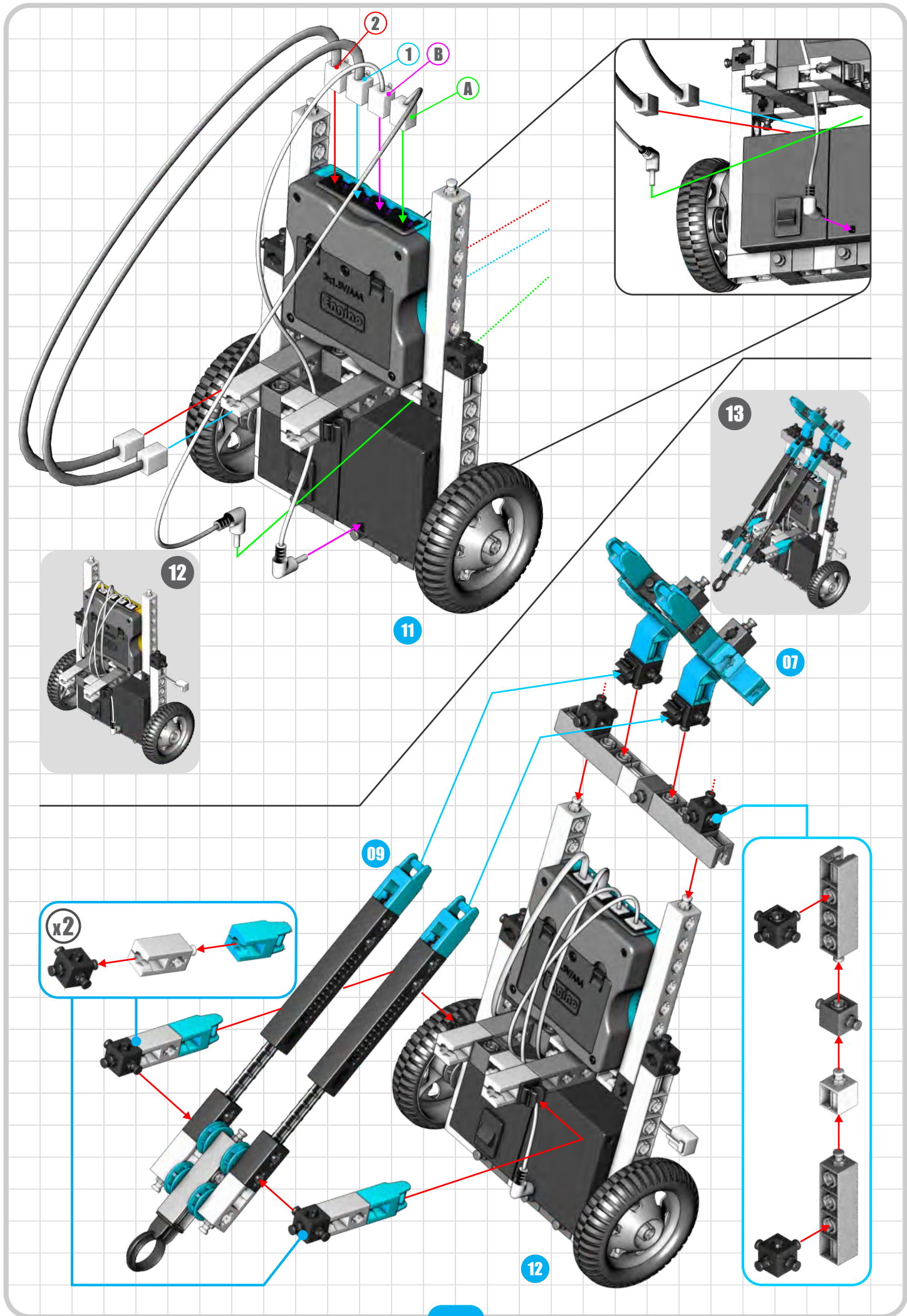


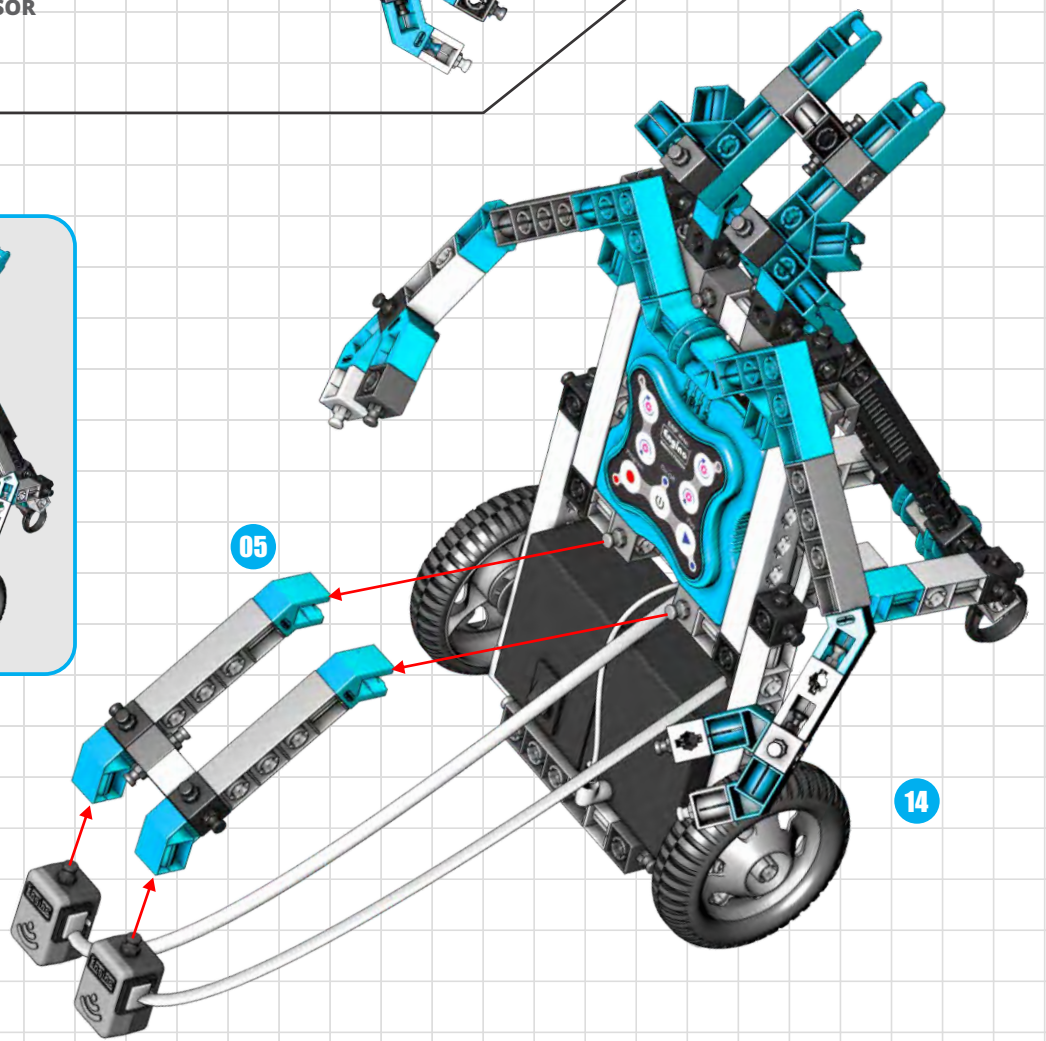
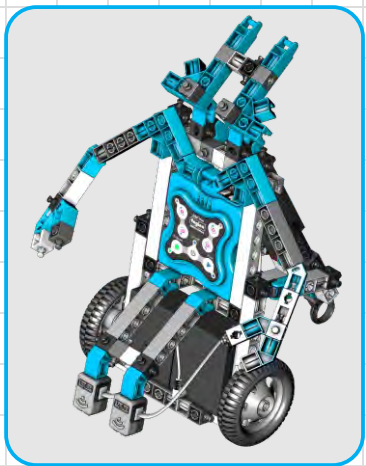
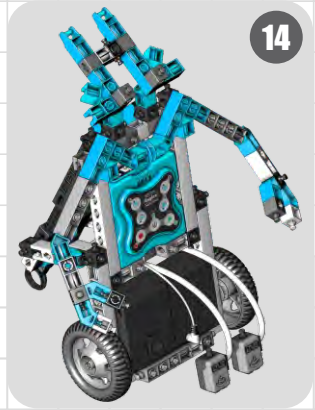
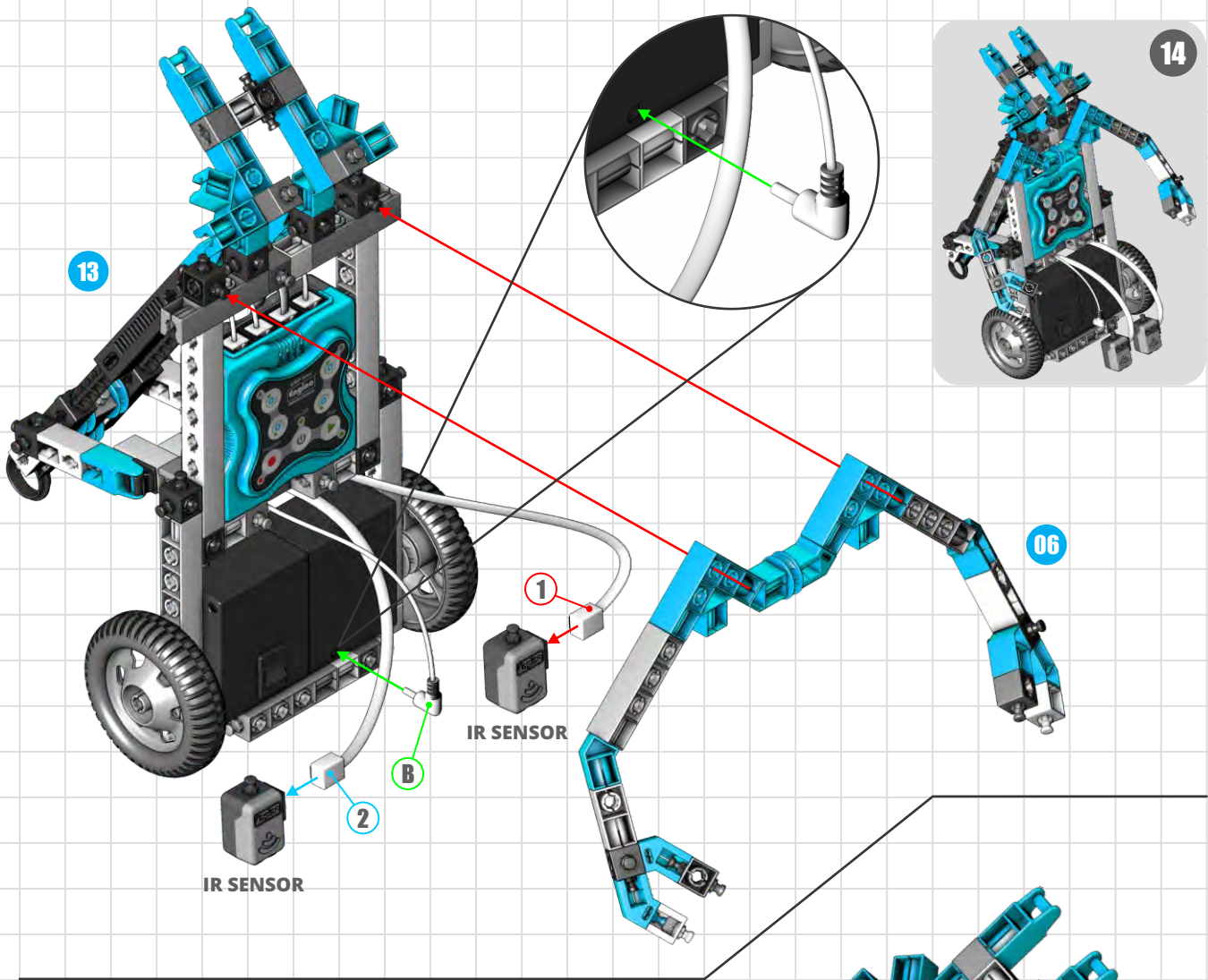
08



09



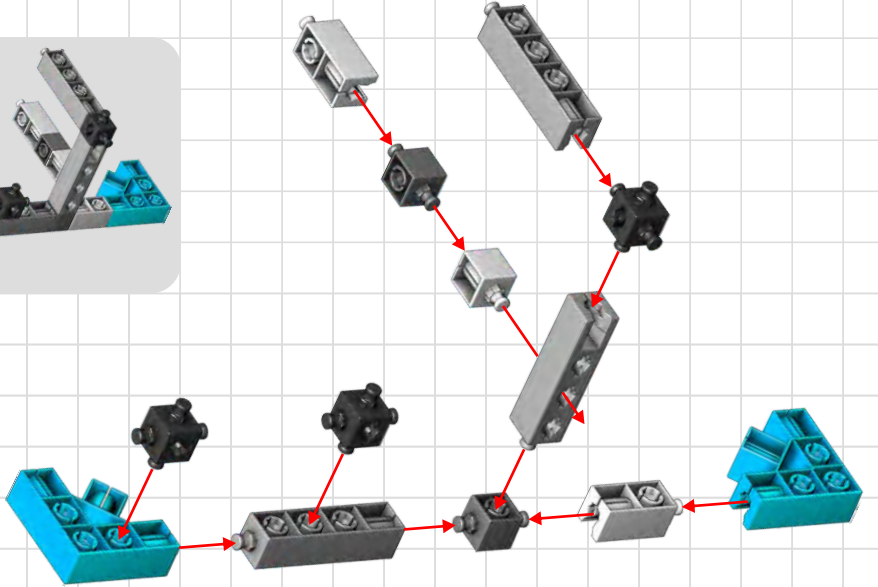
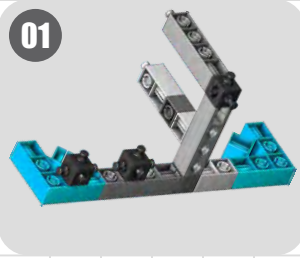




Ventilátor

4020

01



02

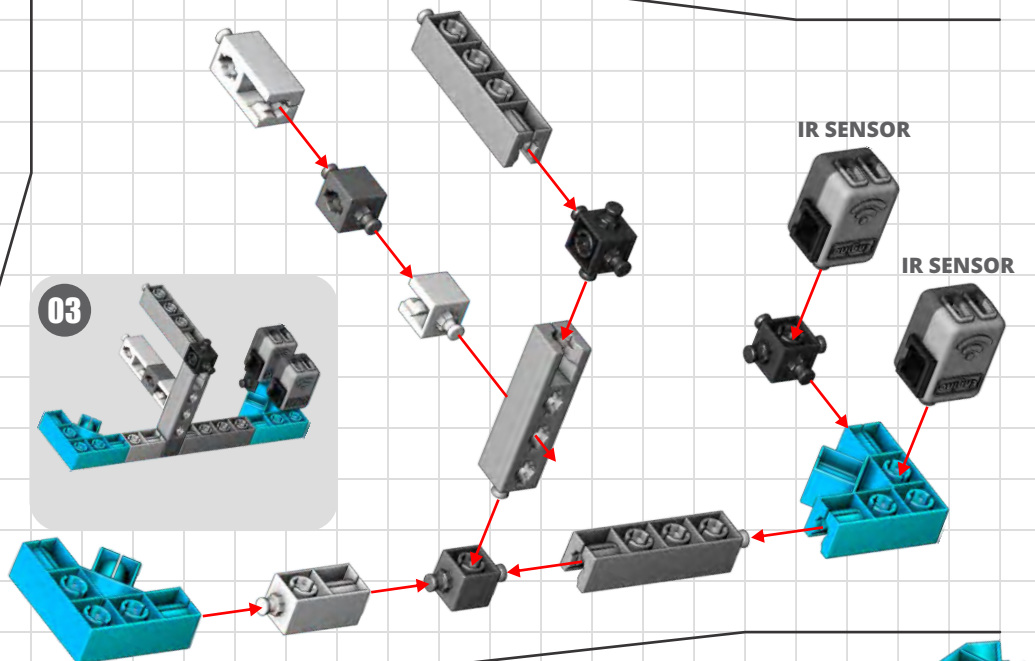


03

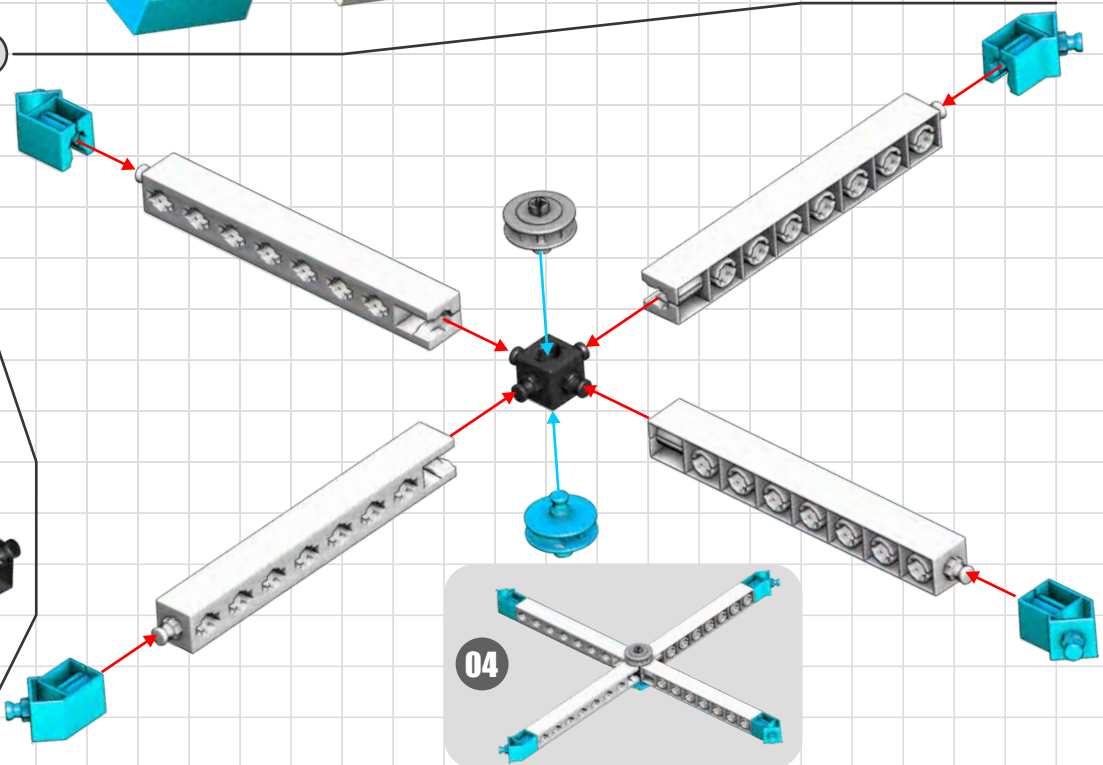
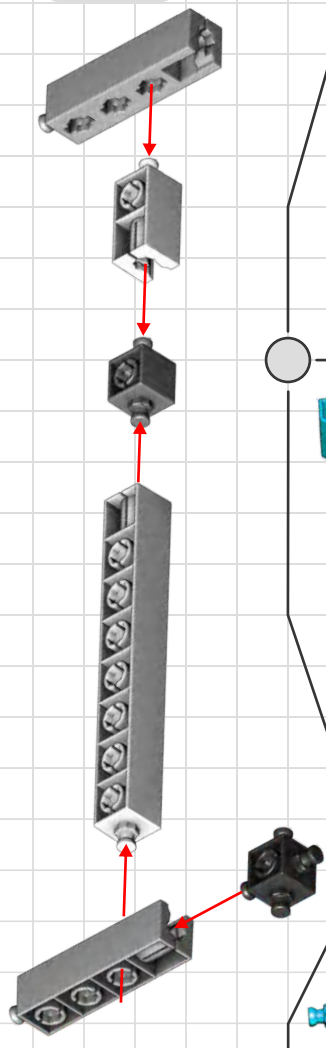
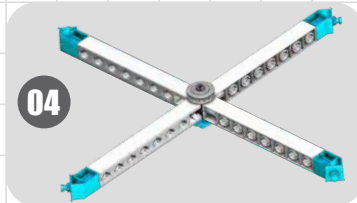


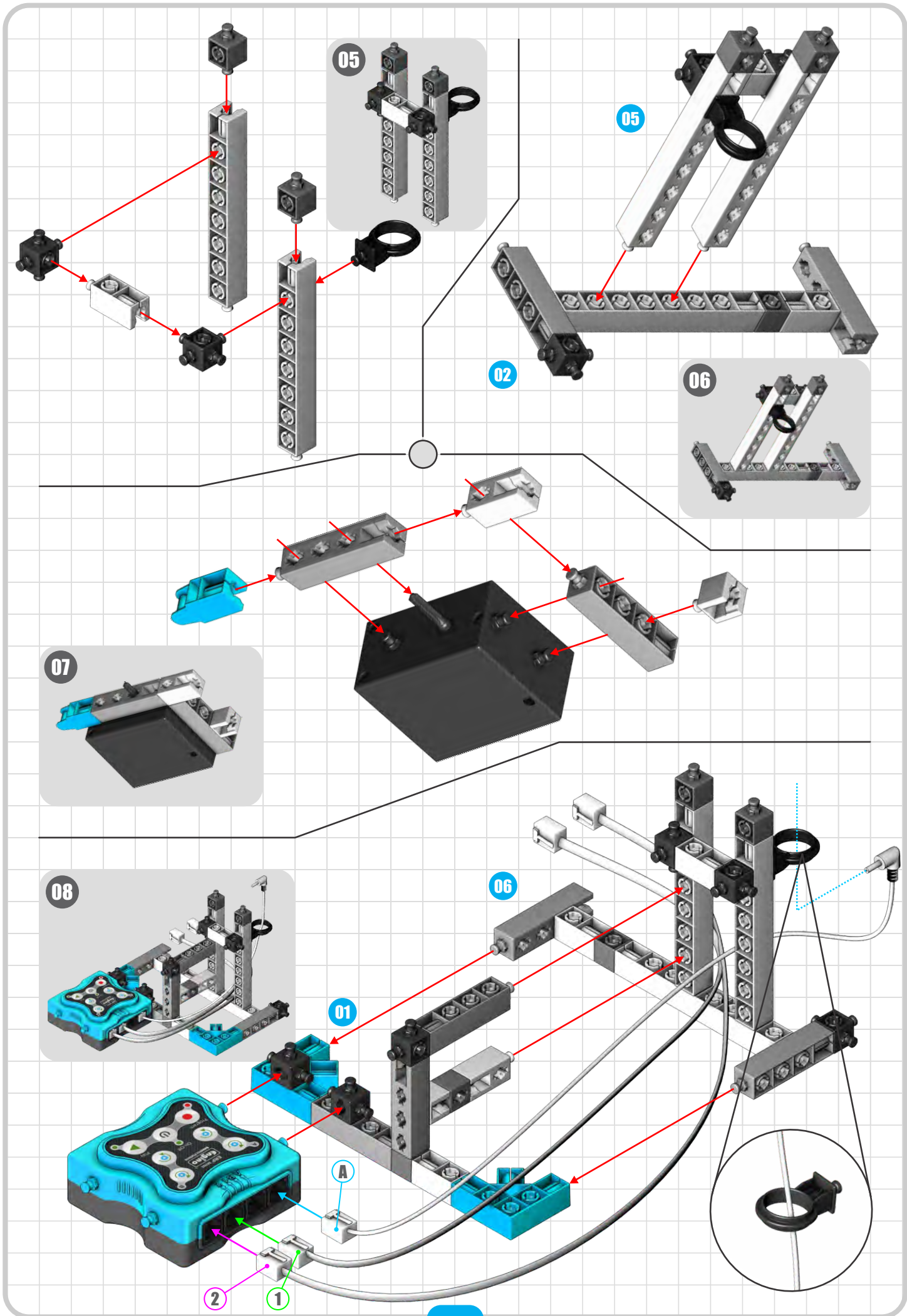
IR SENSOR

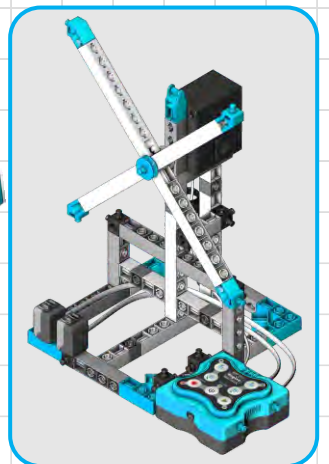
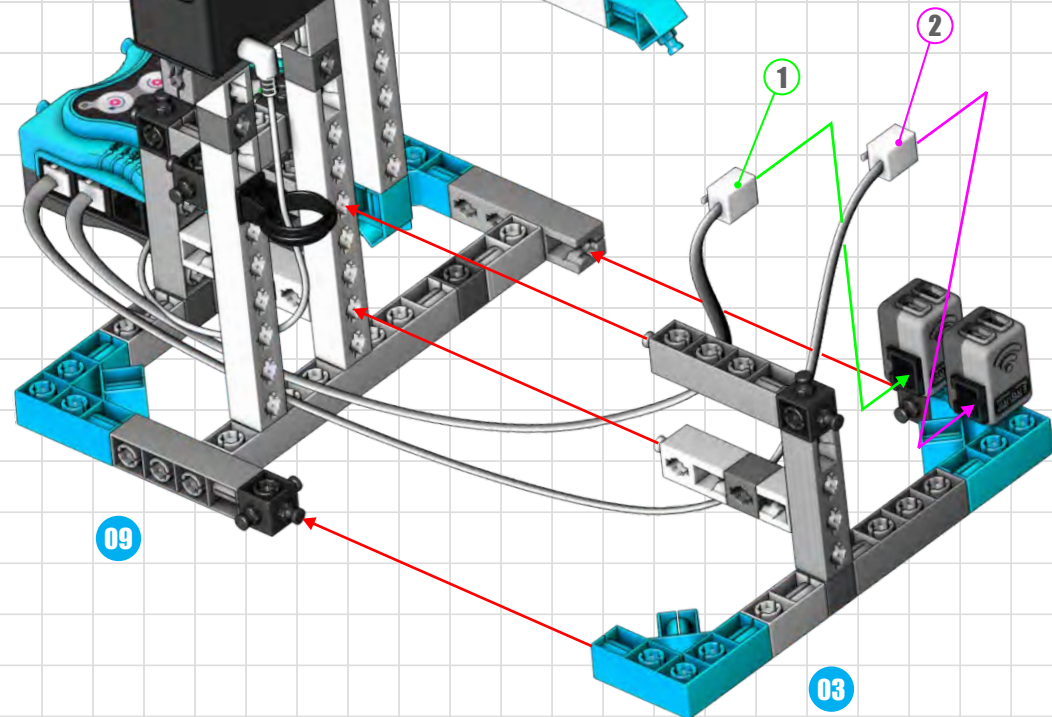
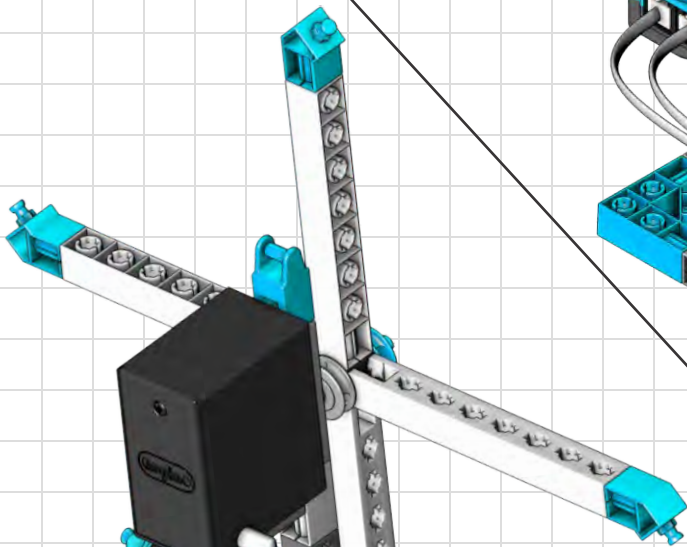
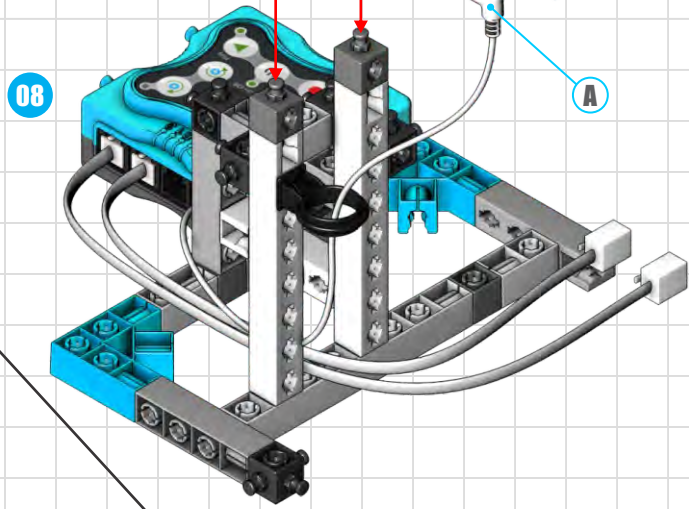
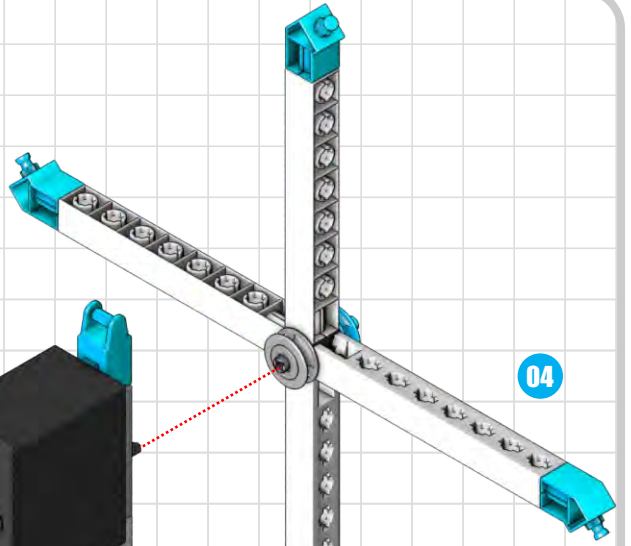
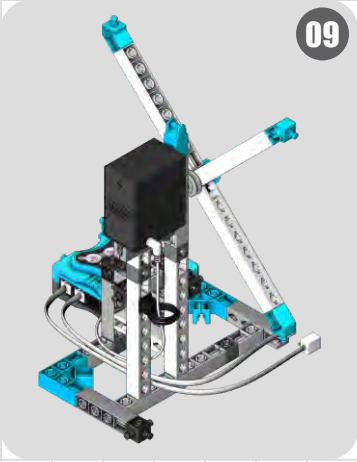
IR SENSOR

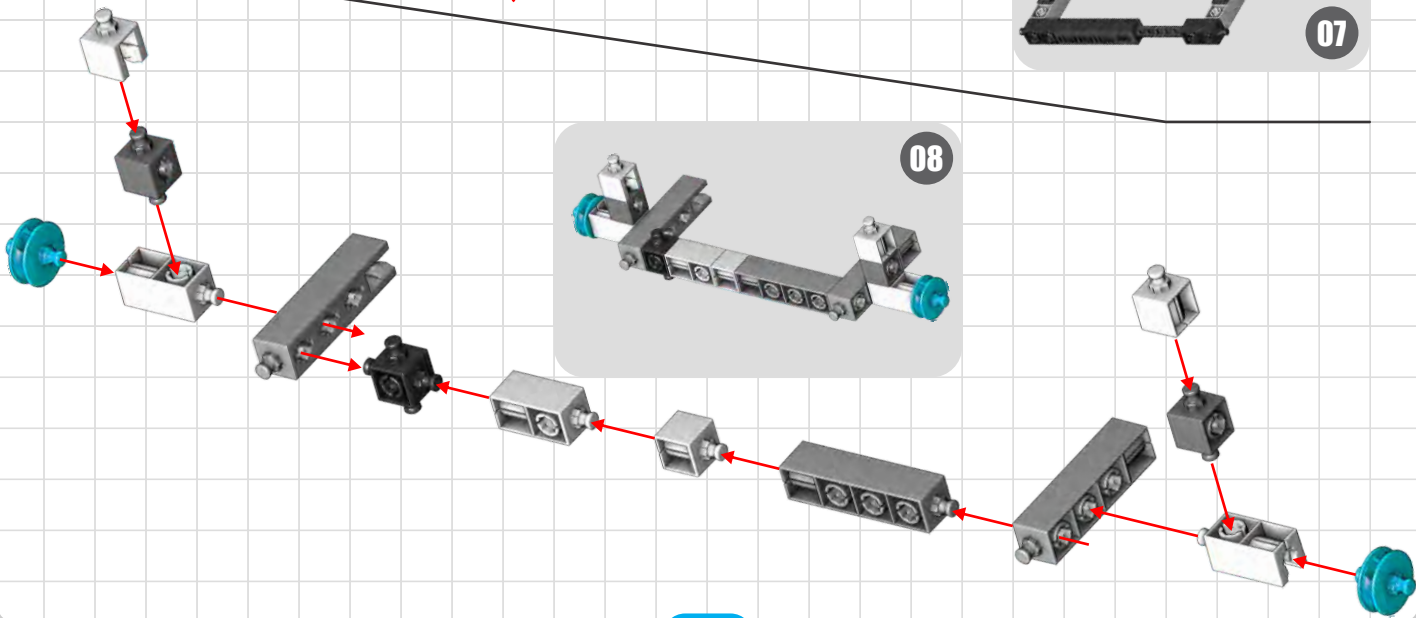
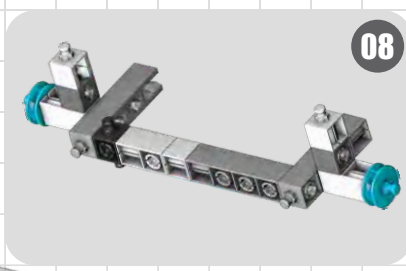
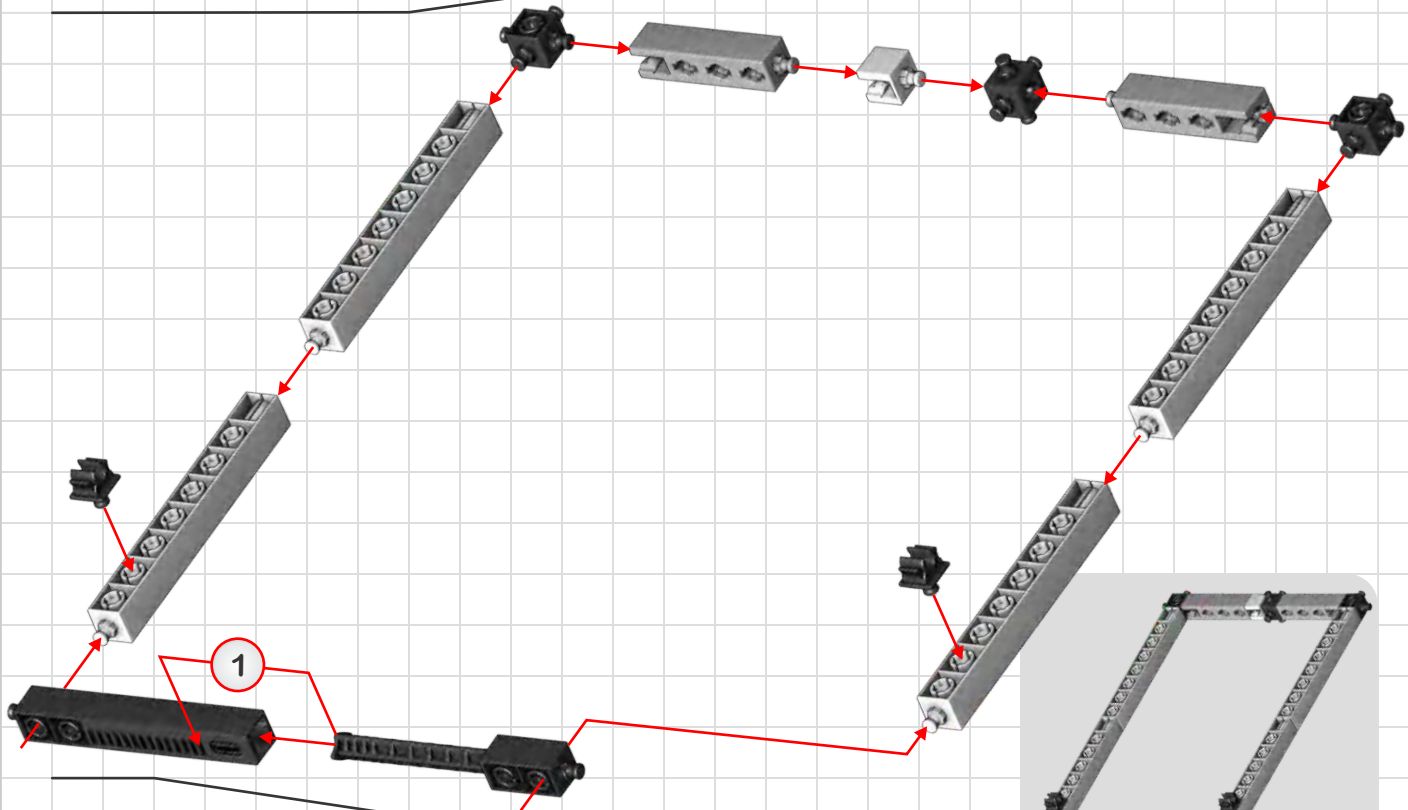
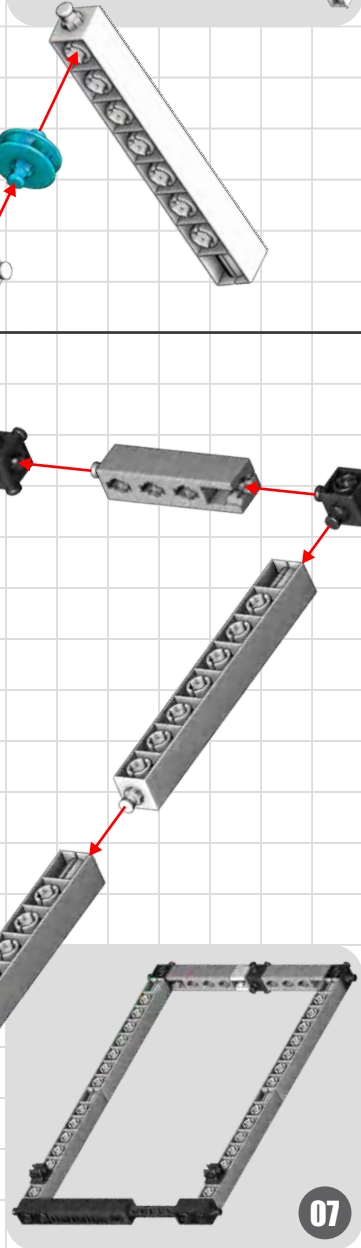
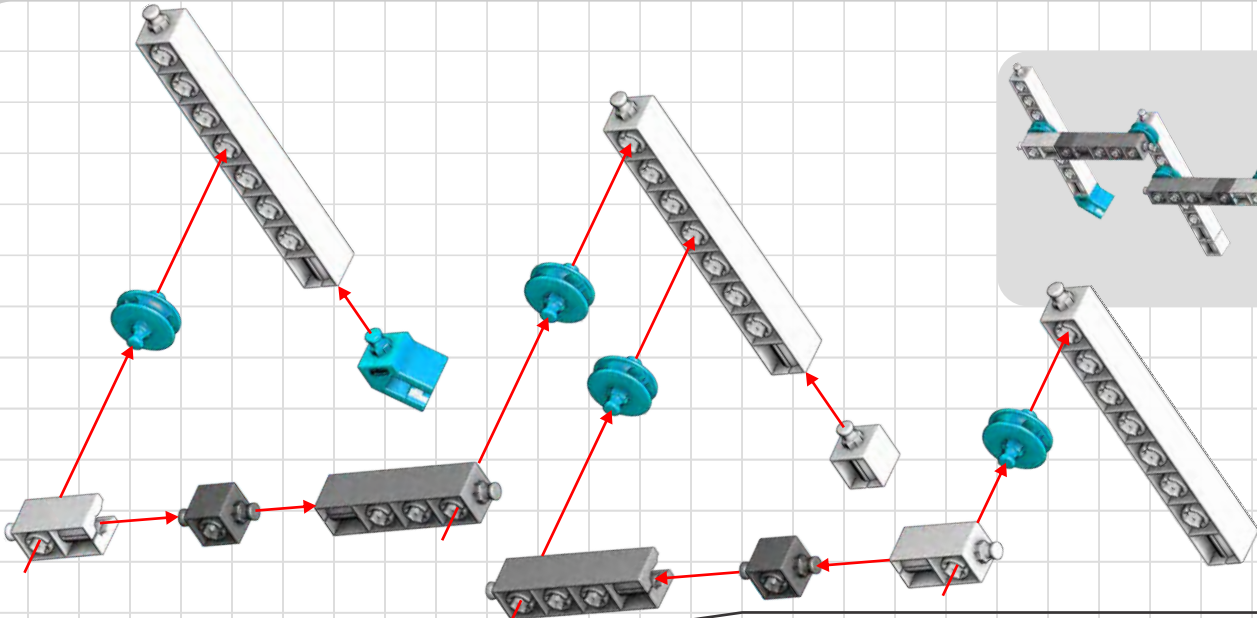
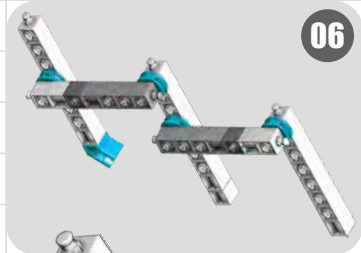


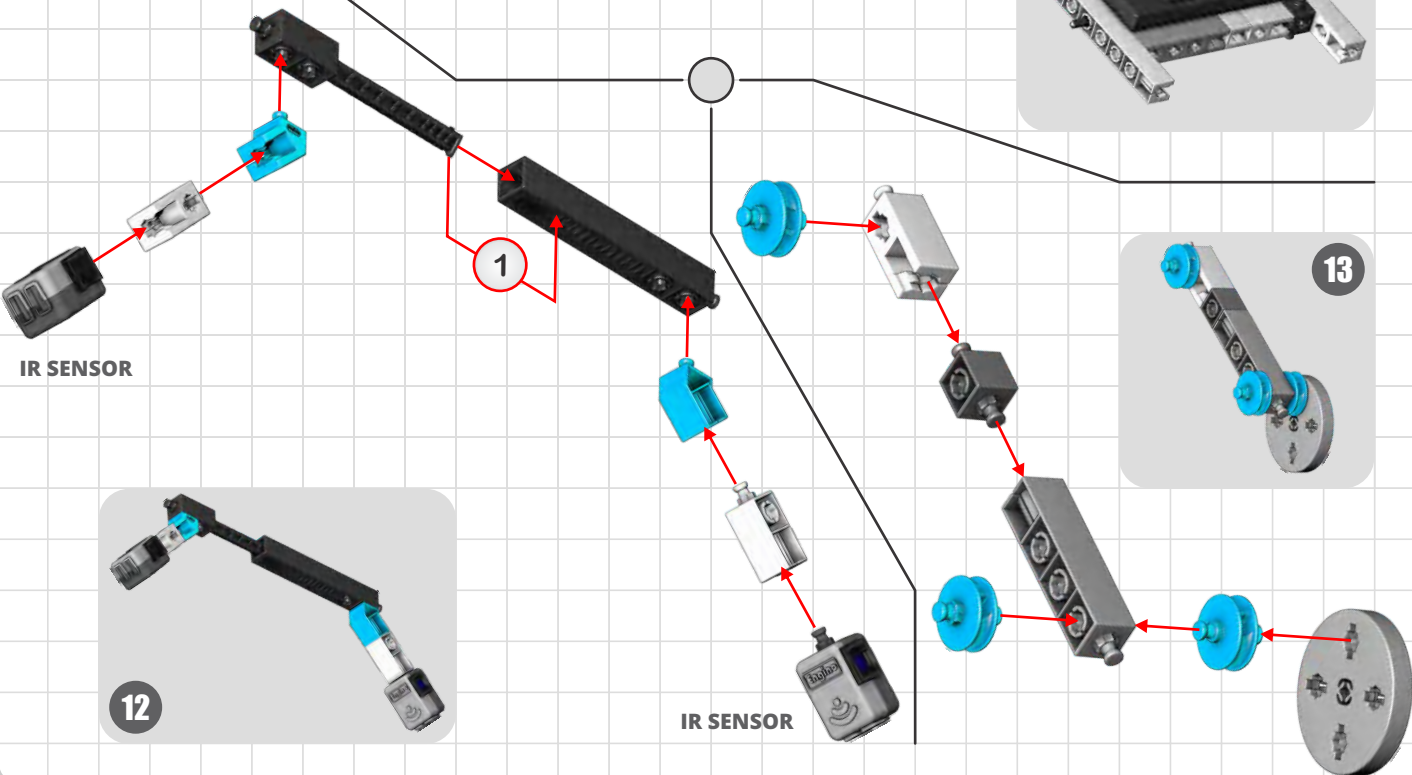
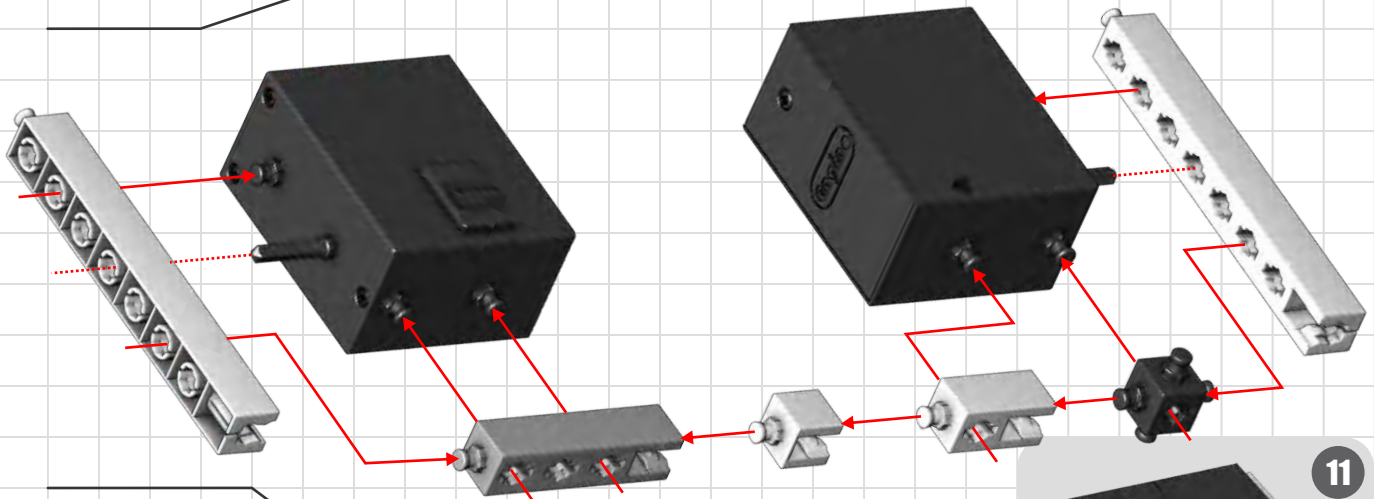
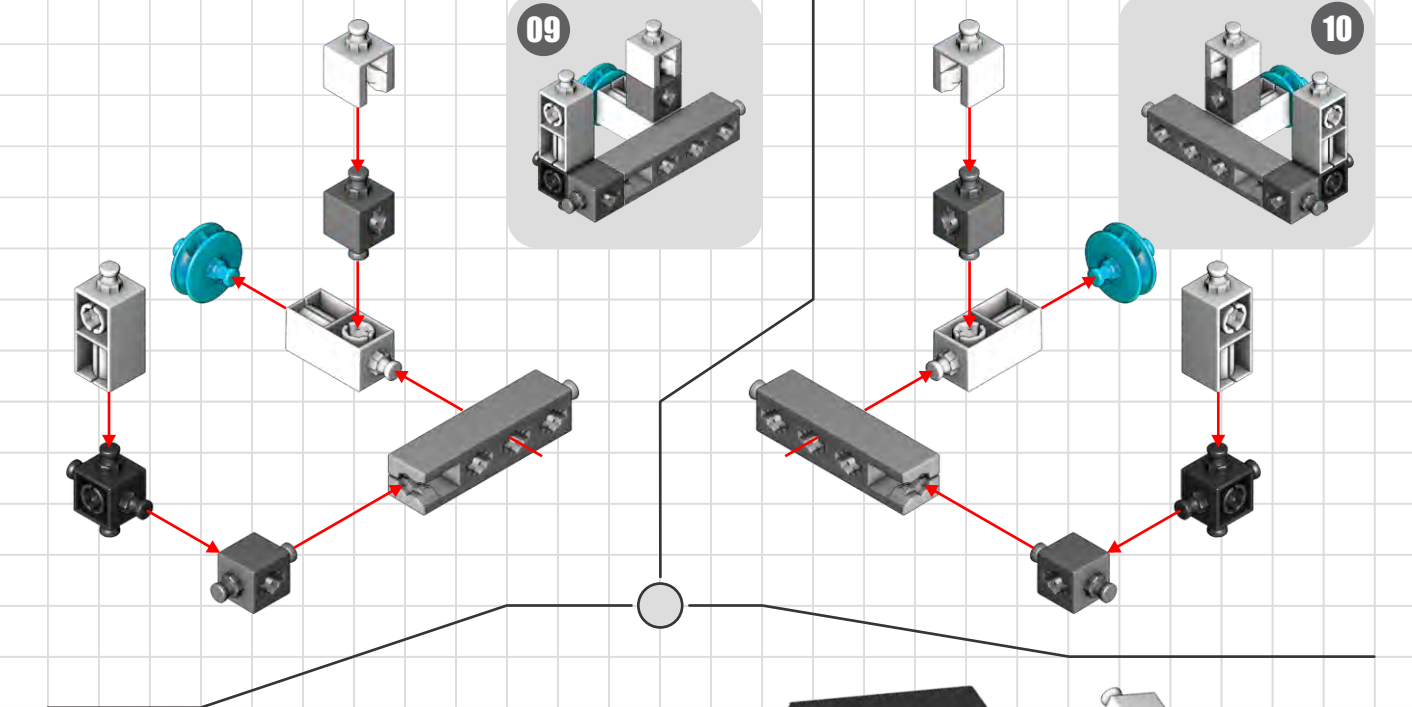
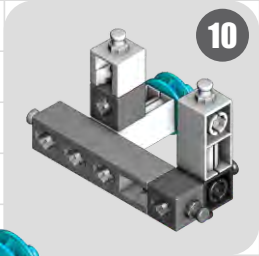
04

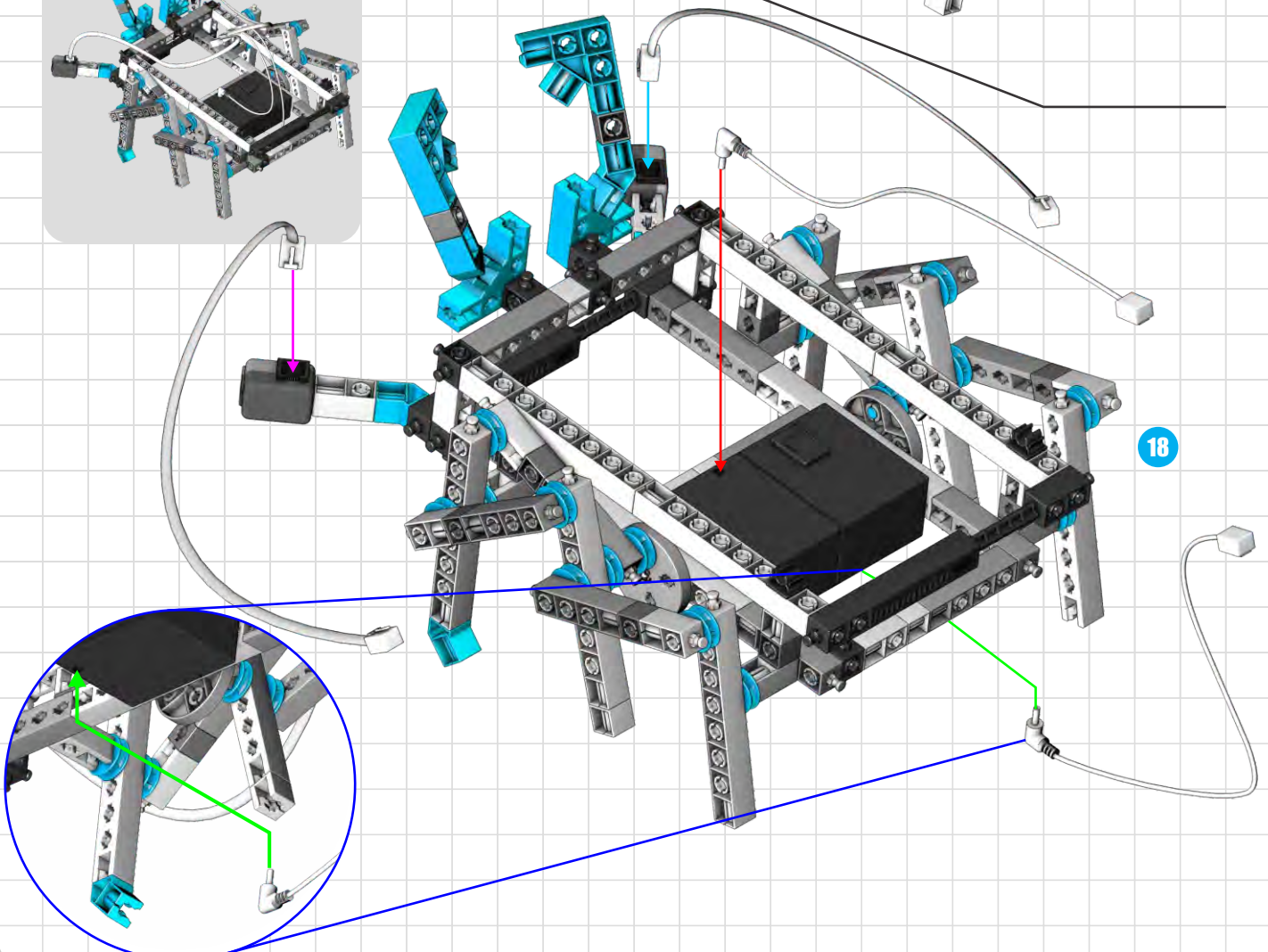
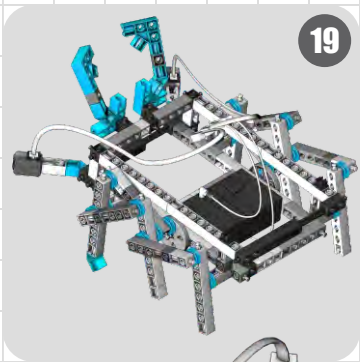
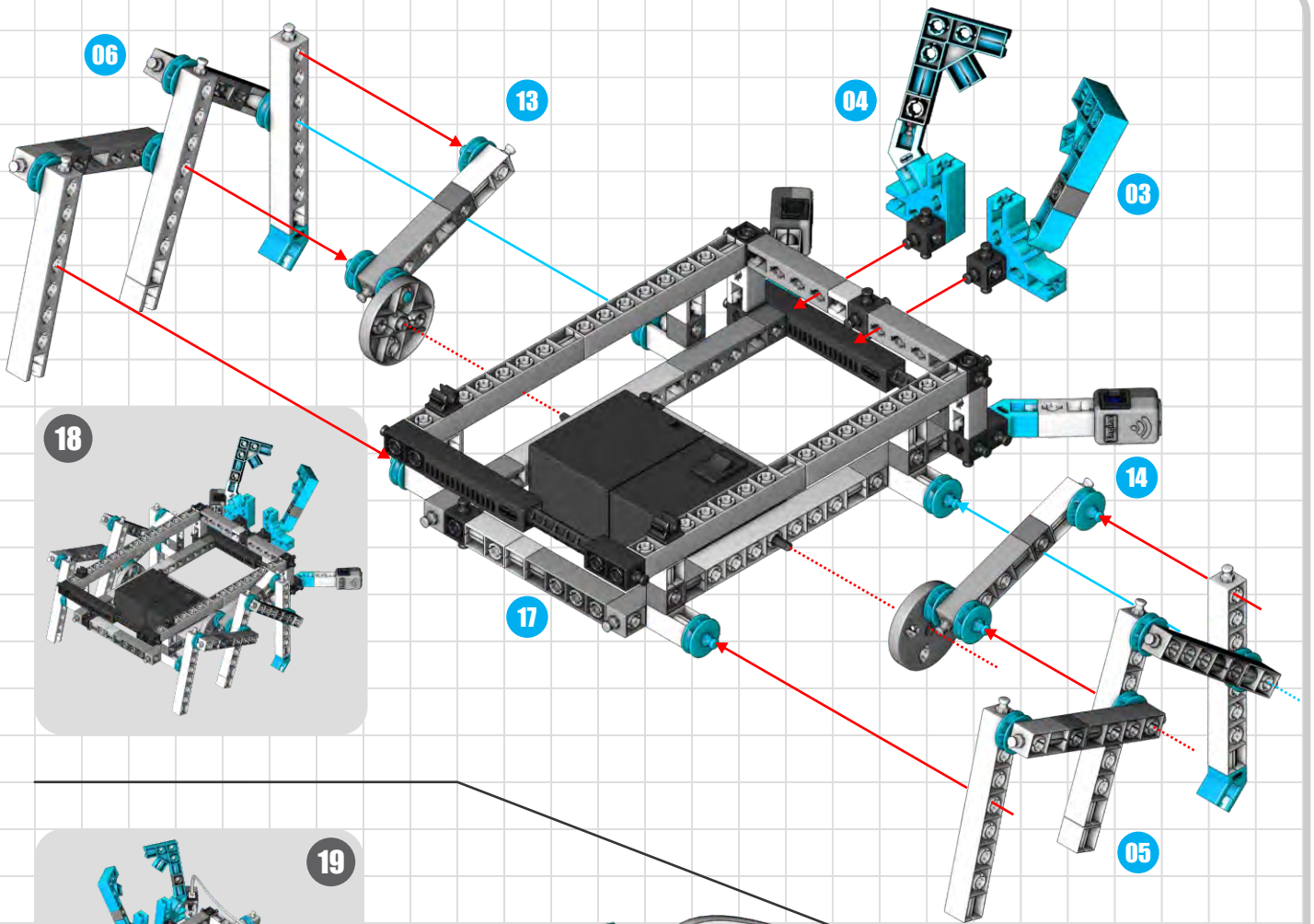


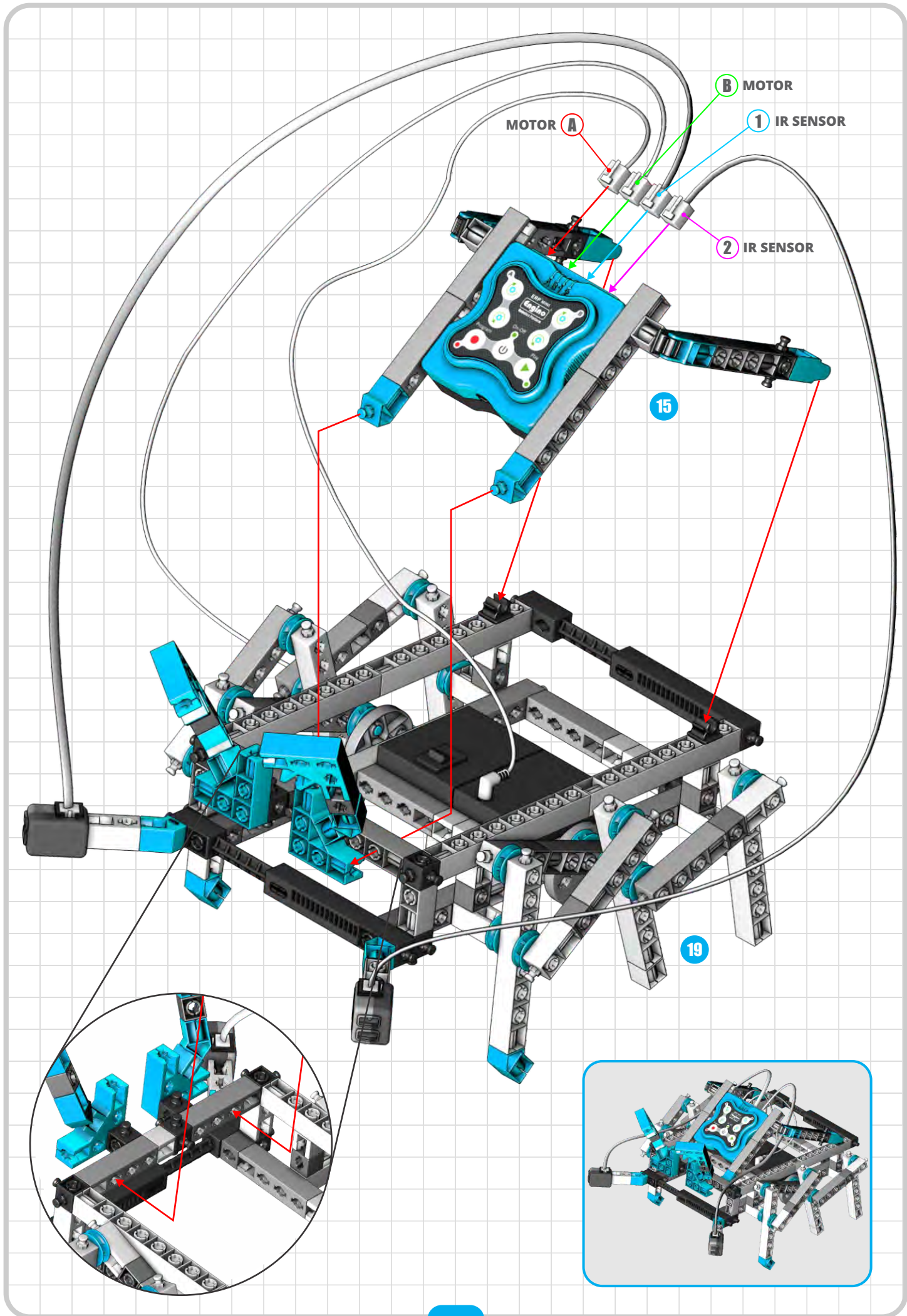












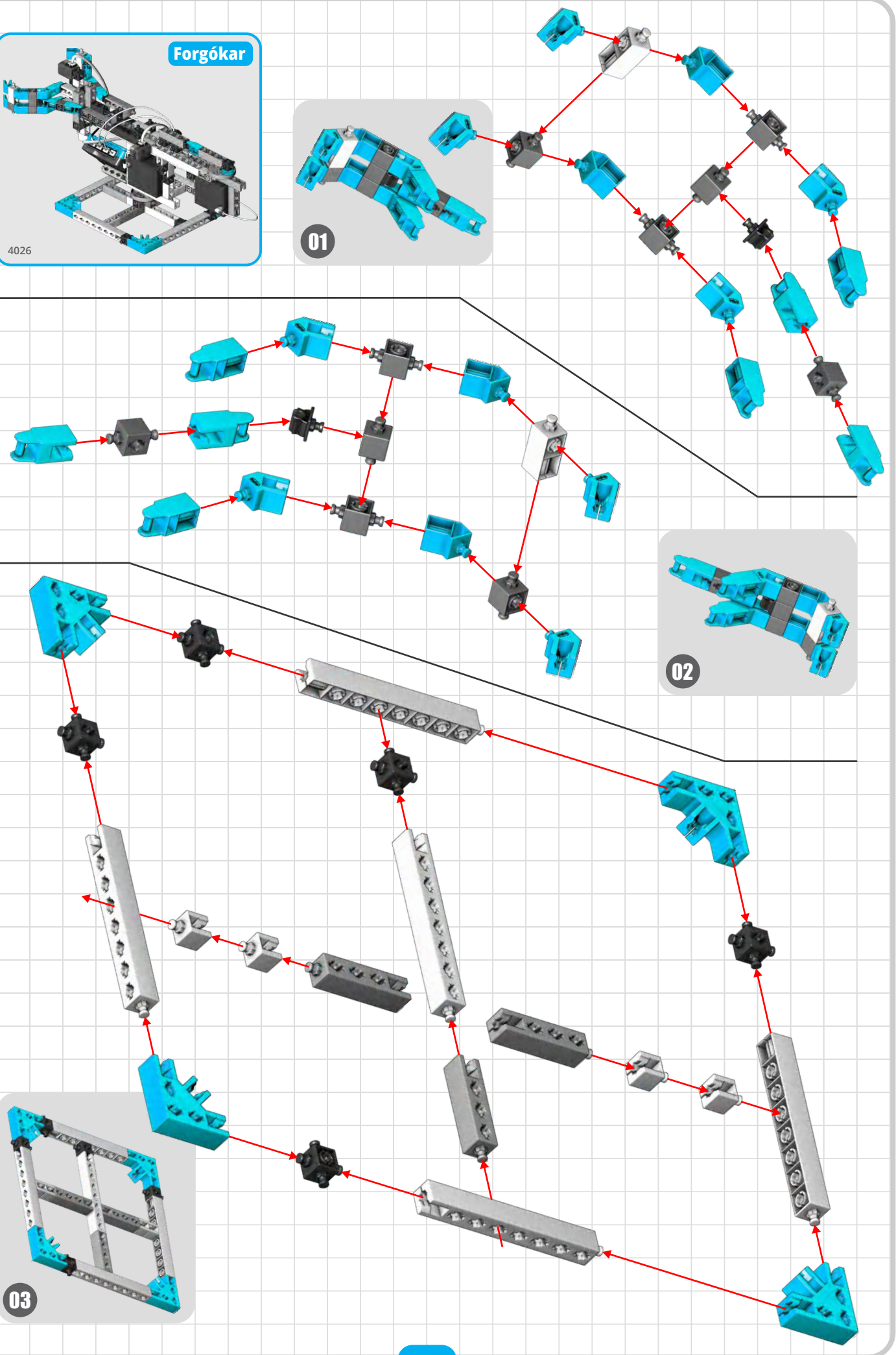
Forgókar

4026

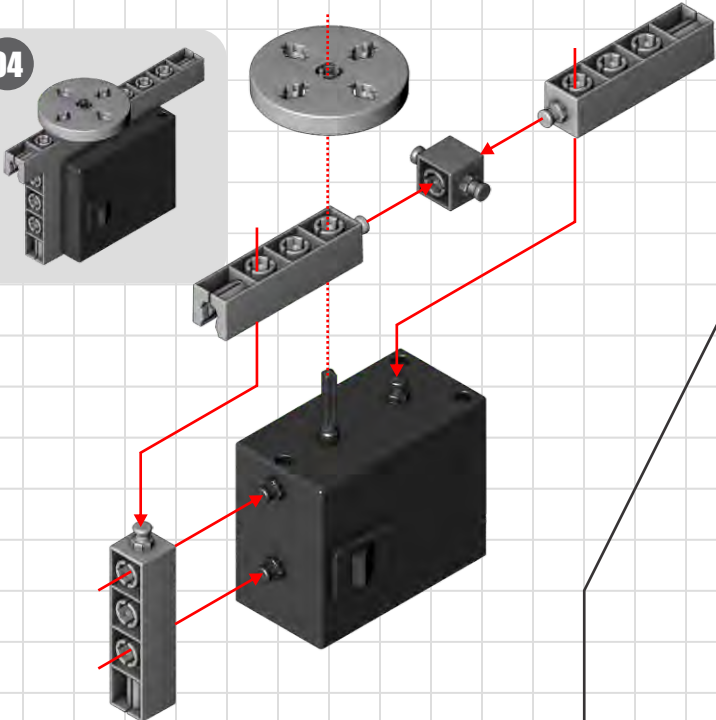
01

02

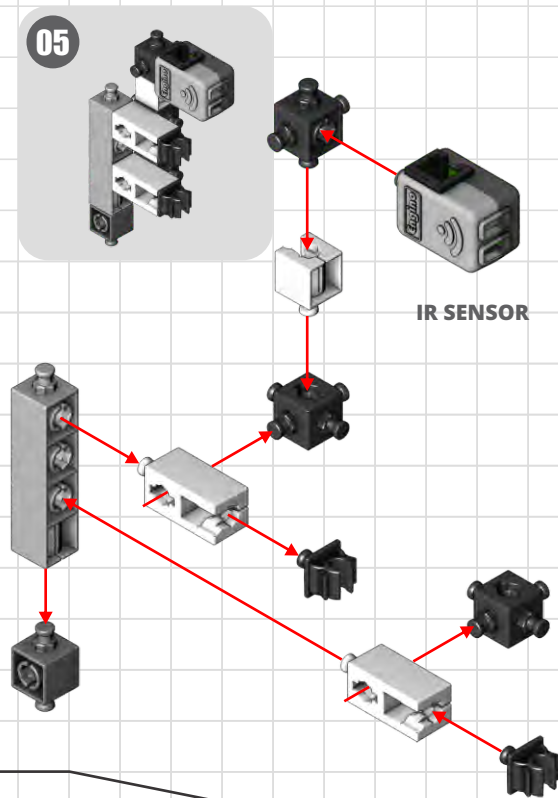
03



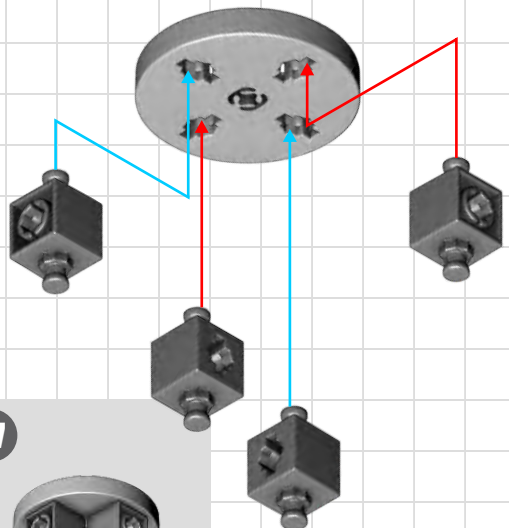
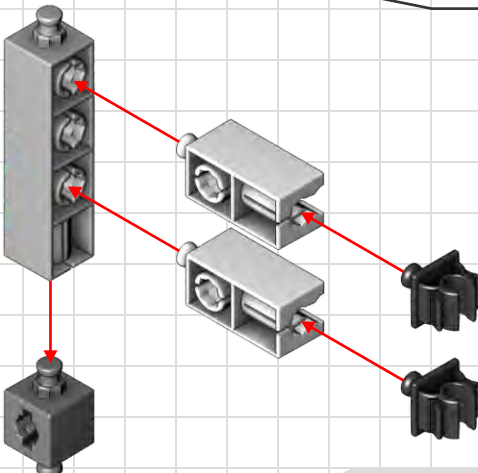
04



05



IR SENSOR



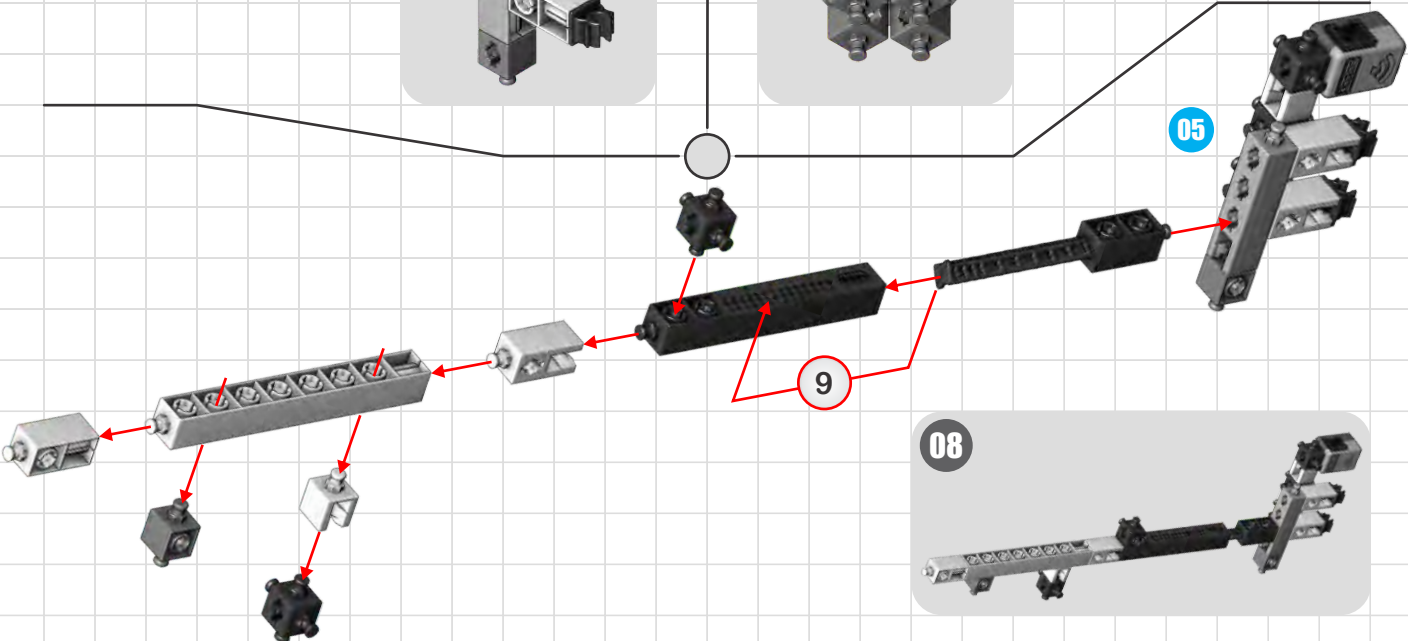
06



07

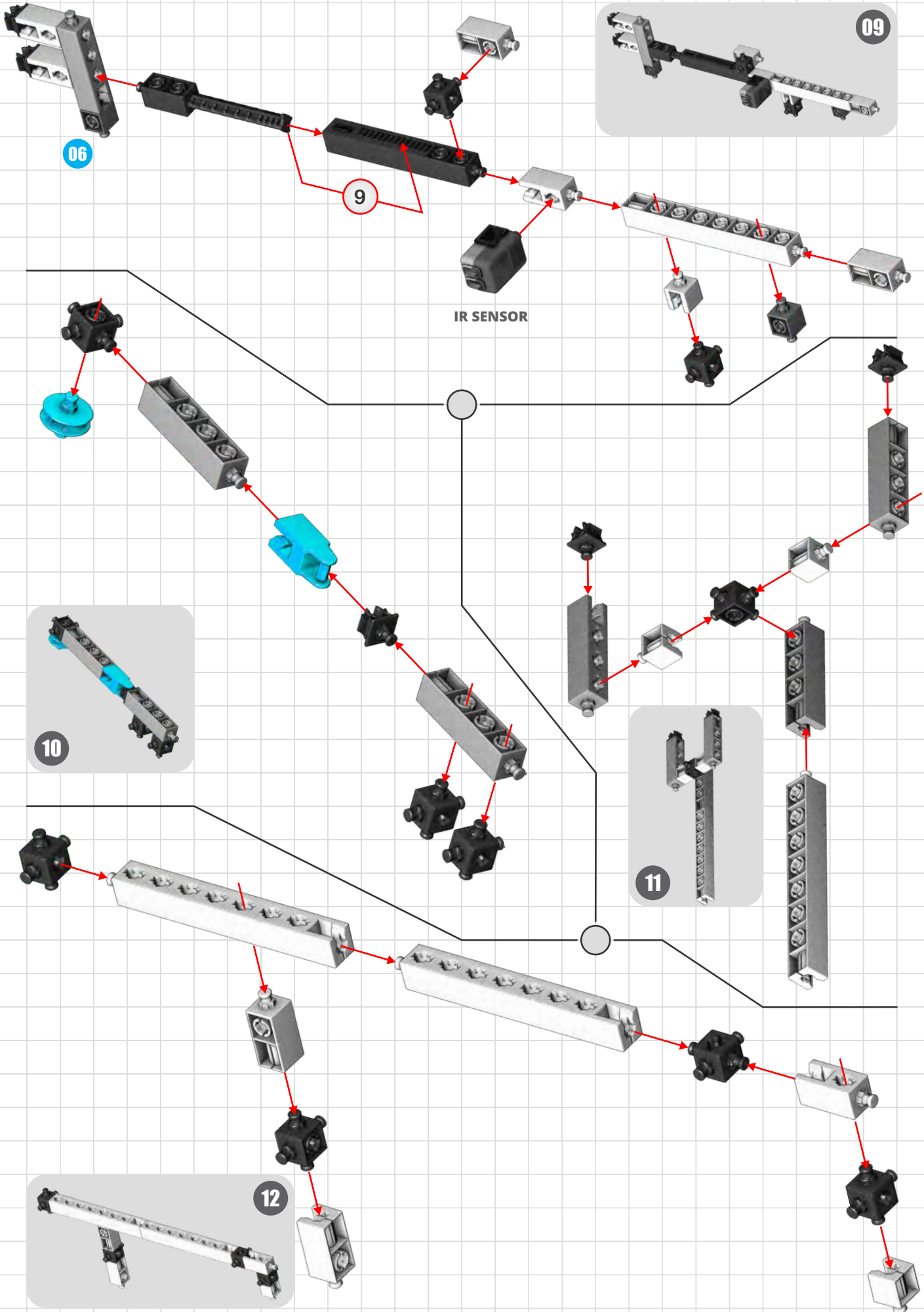


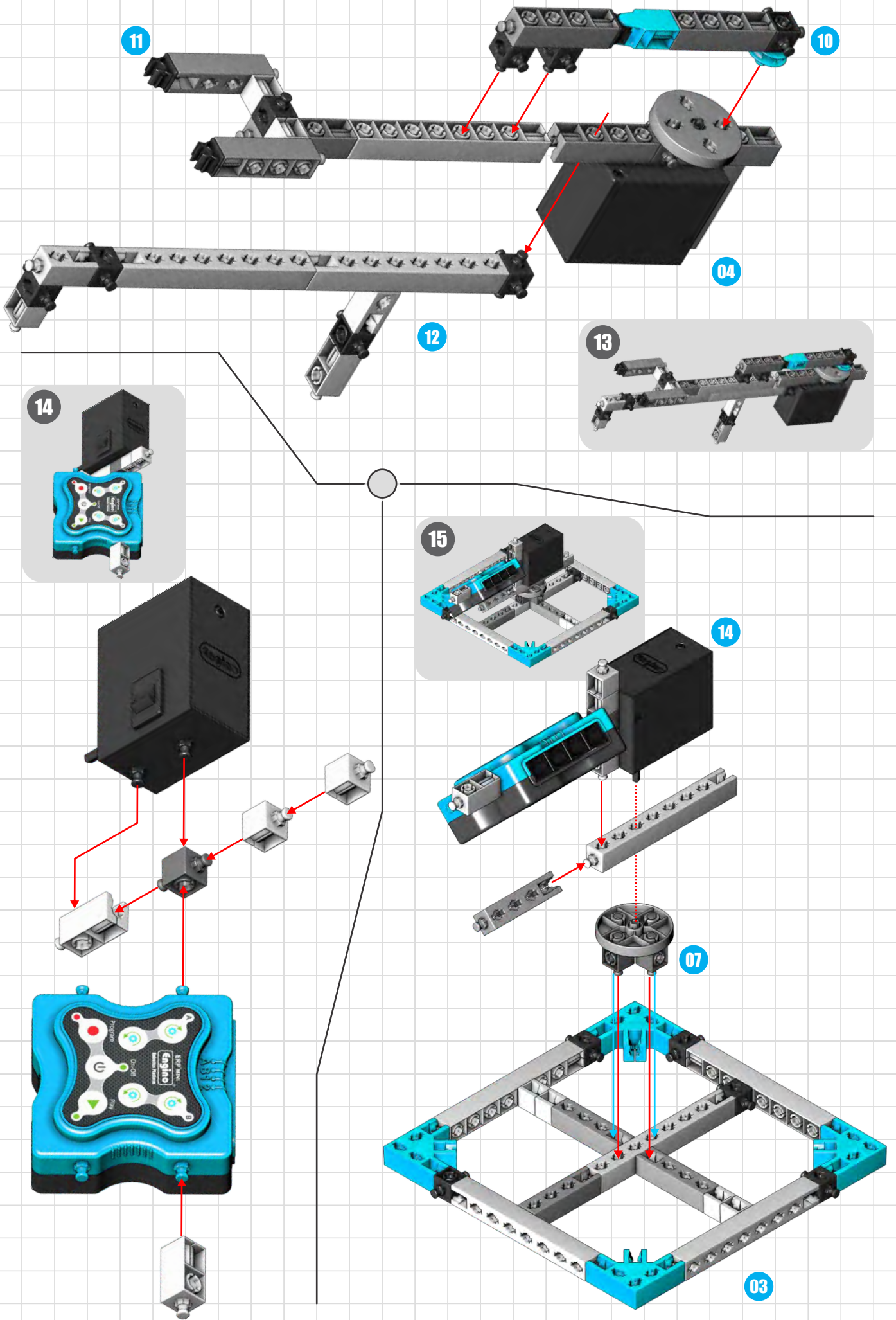
05



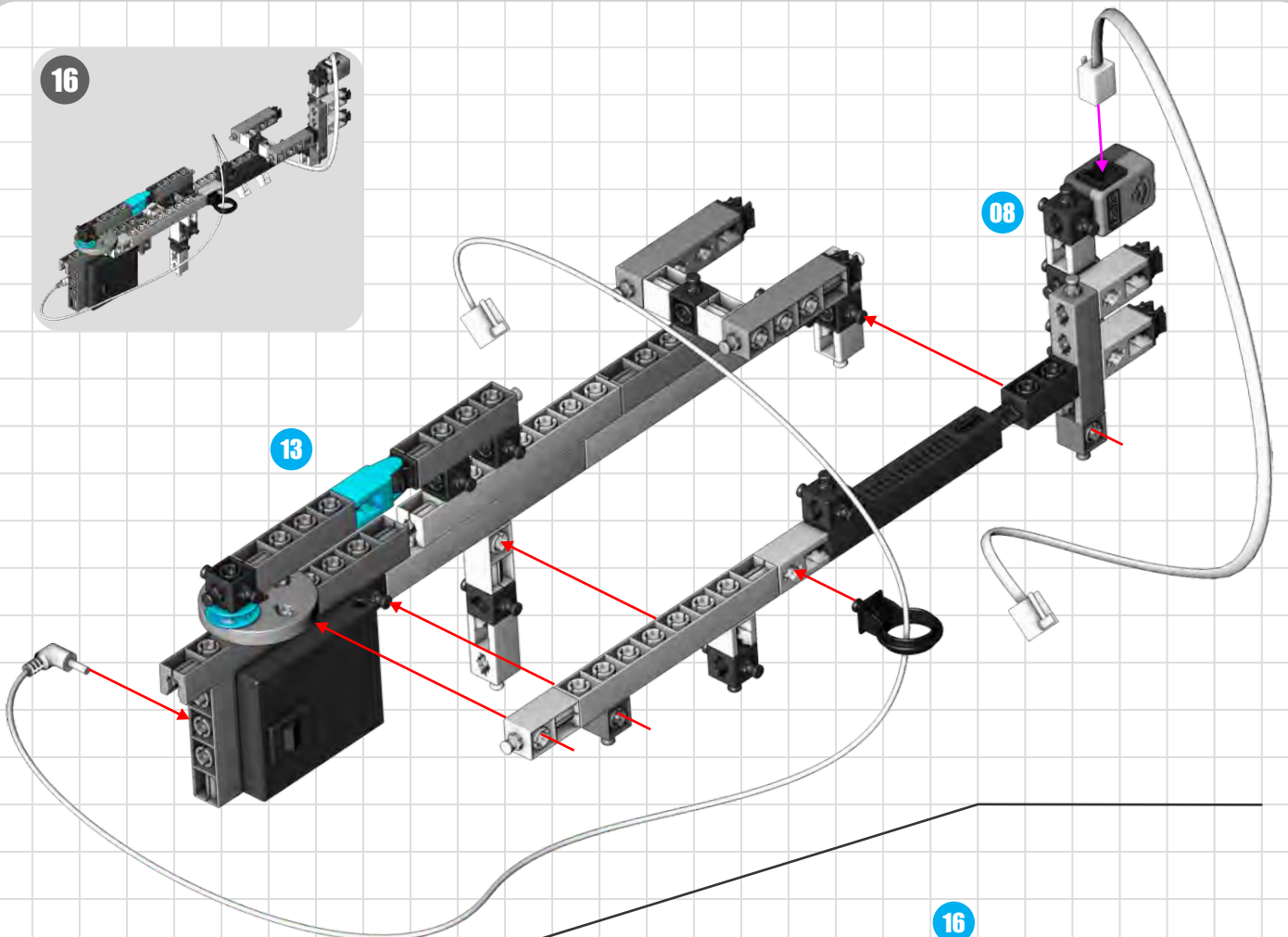
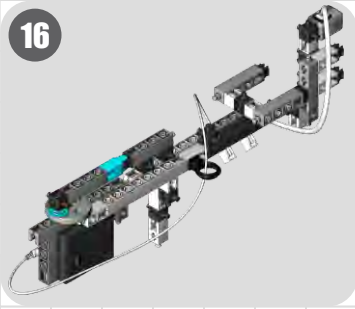
08

9

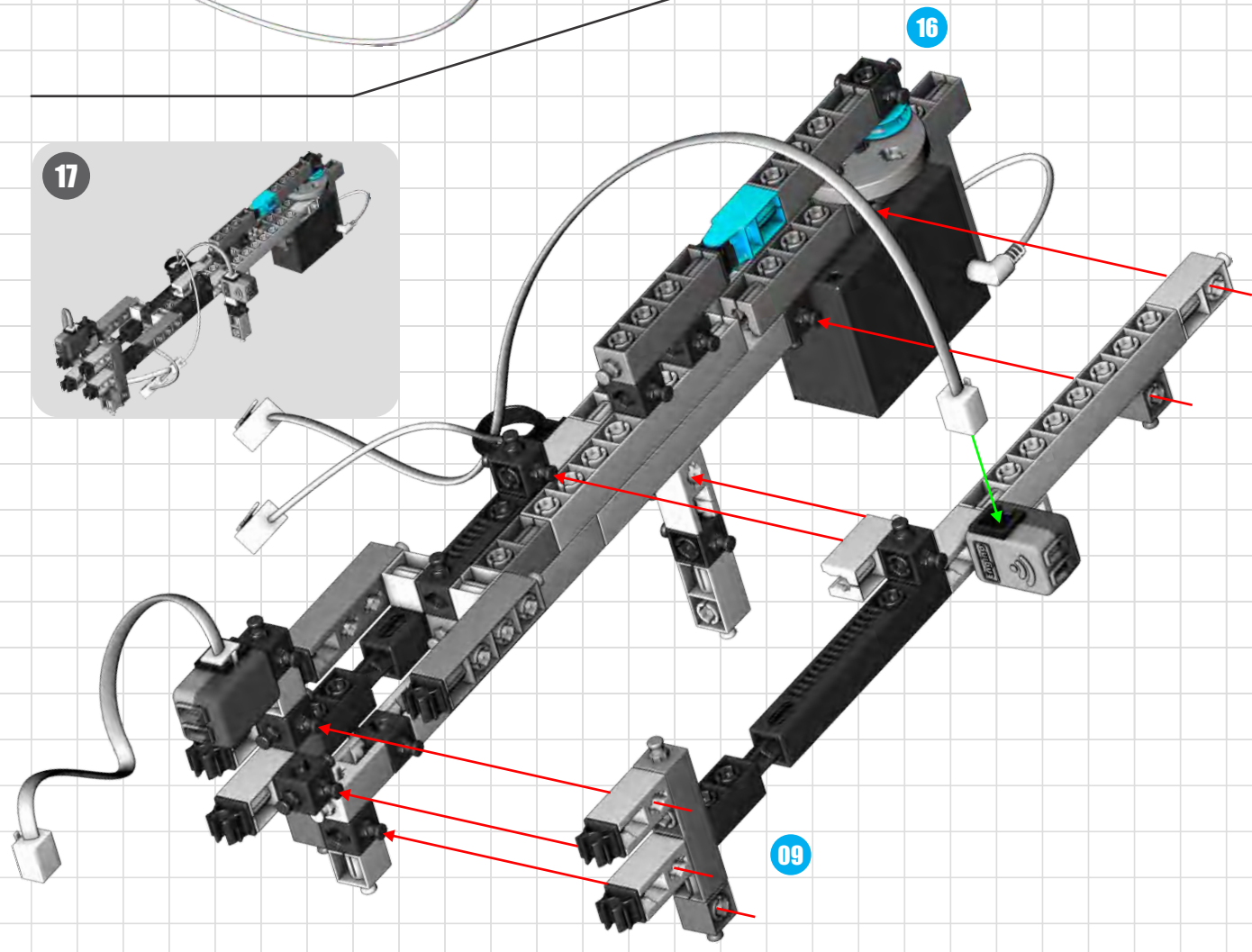
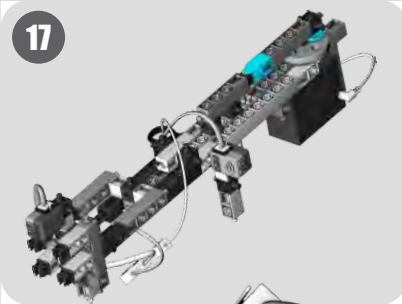


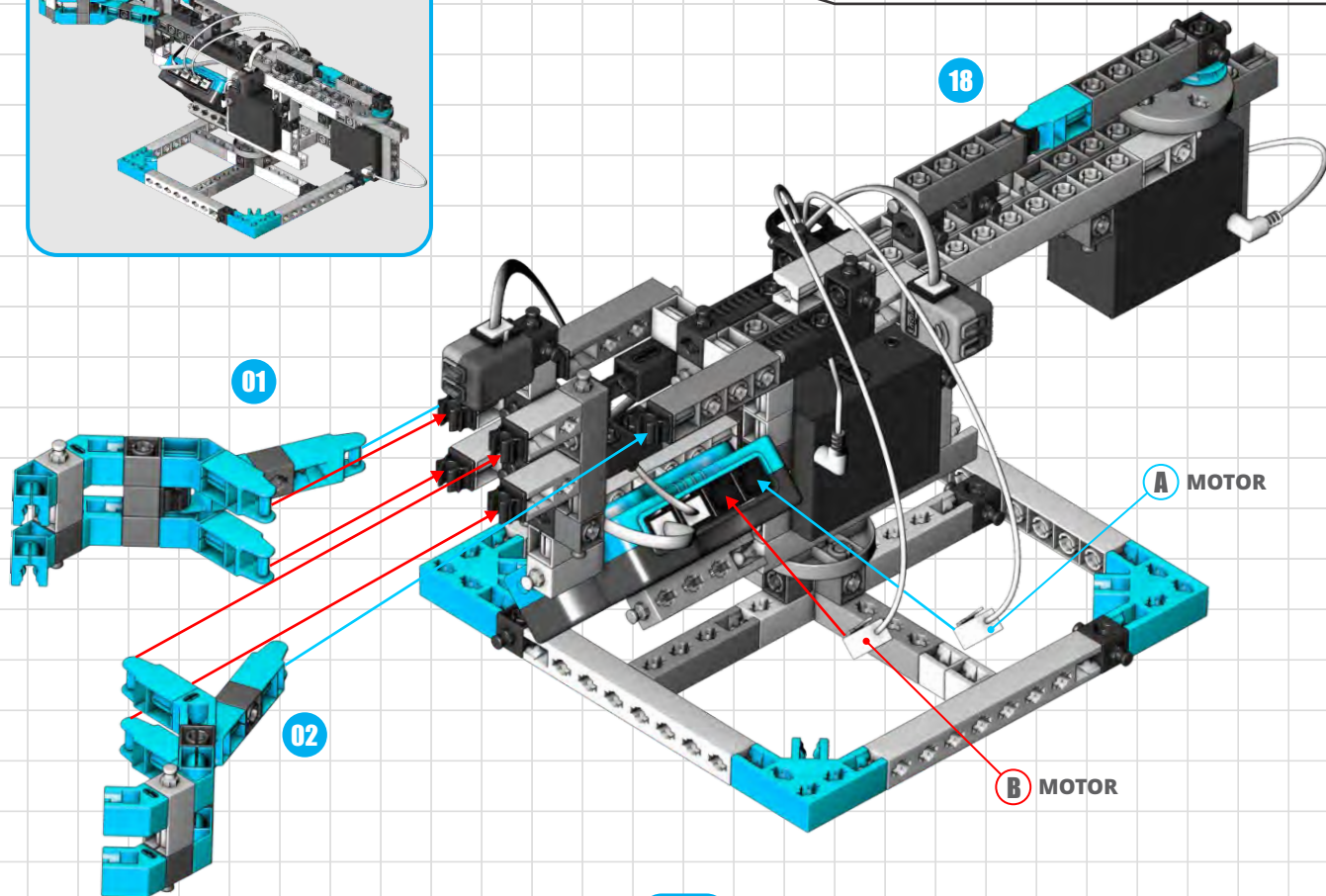
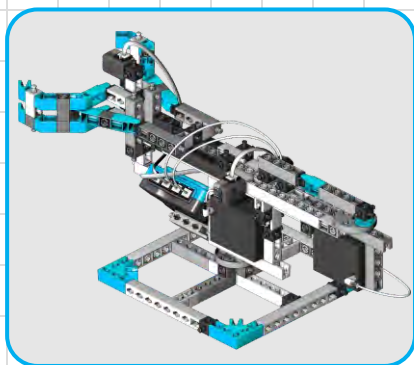
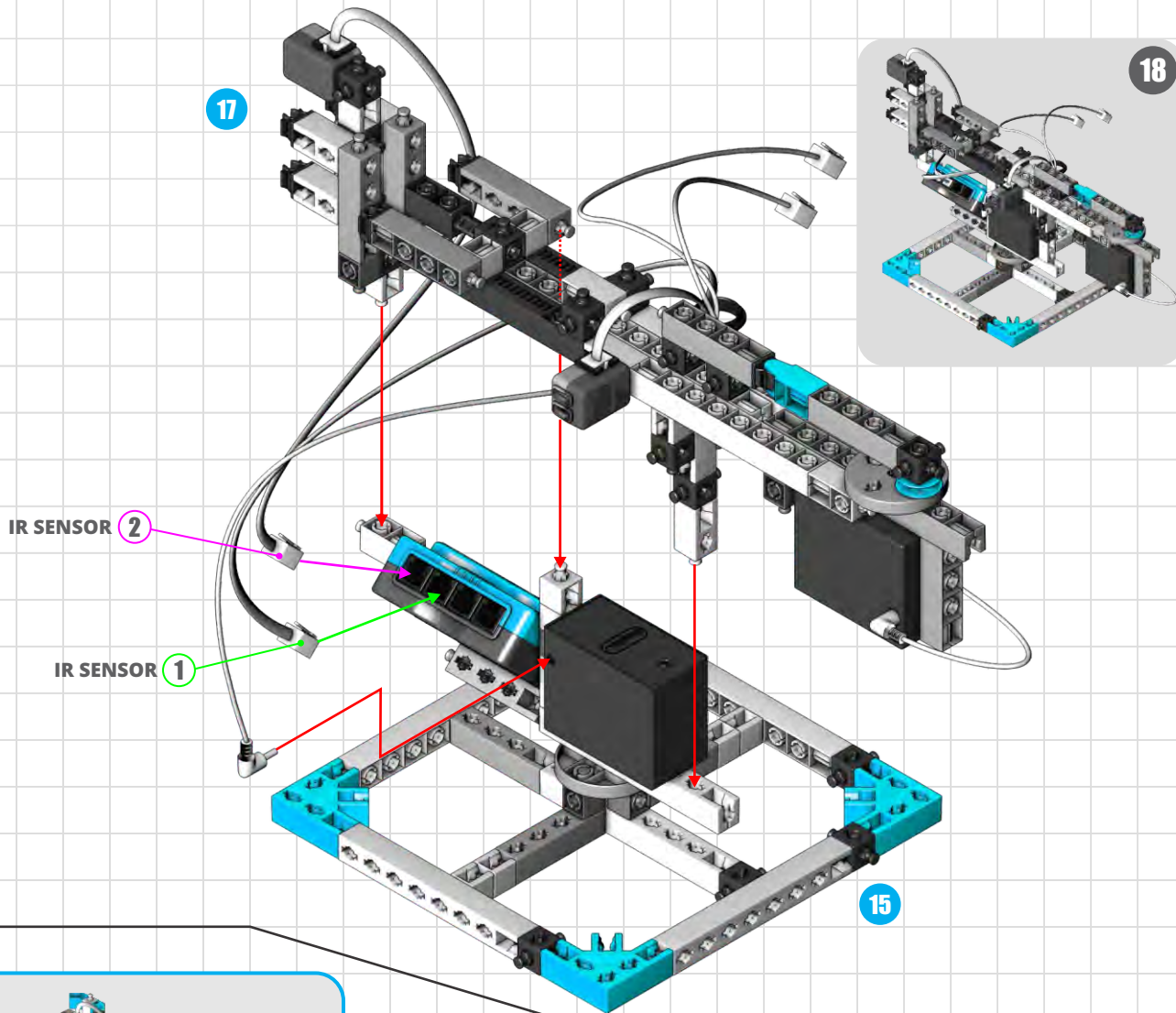


16



17

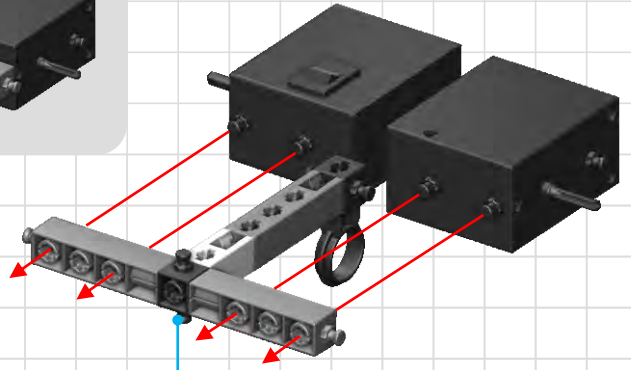




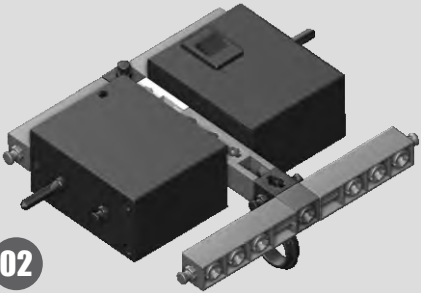
Skorpió

4025

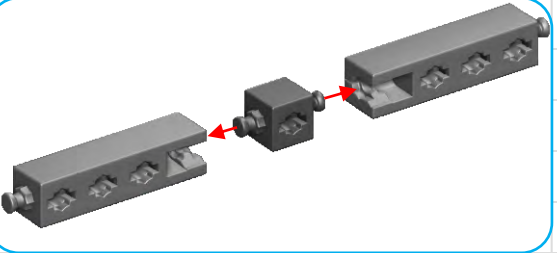
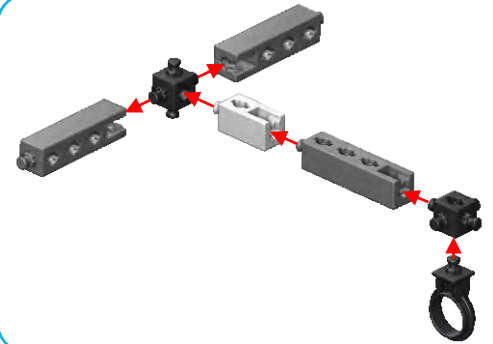
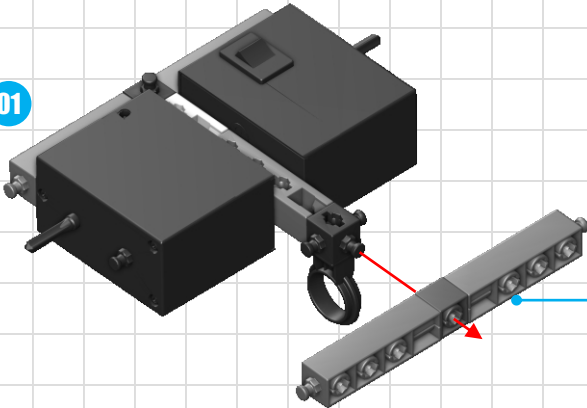
01



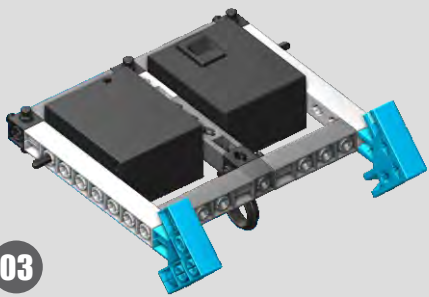
02



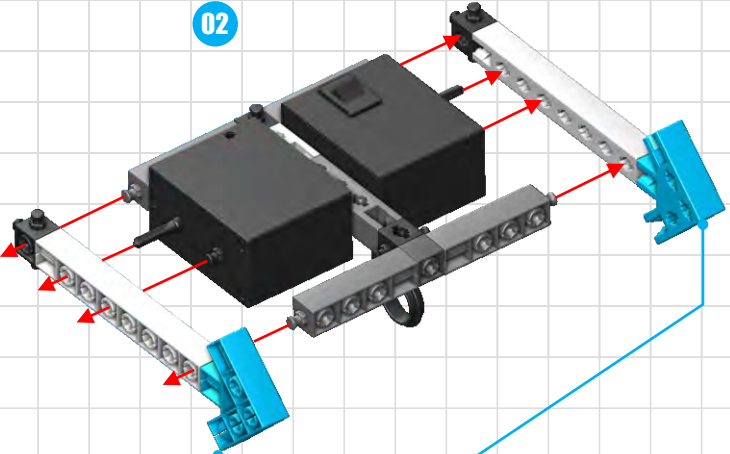
01



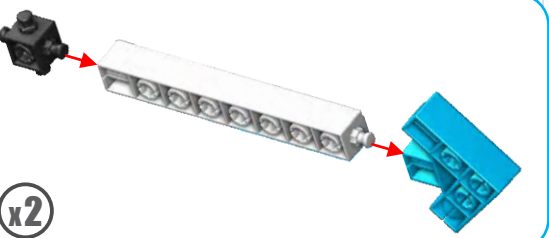
03



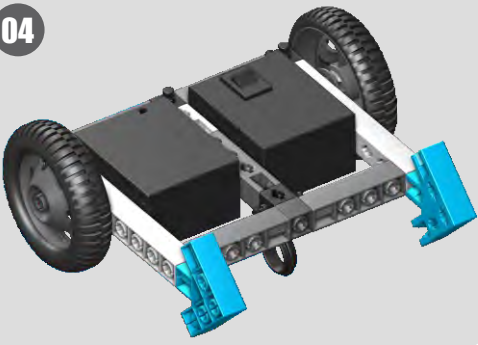
02



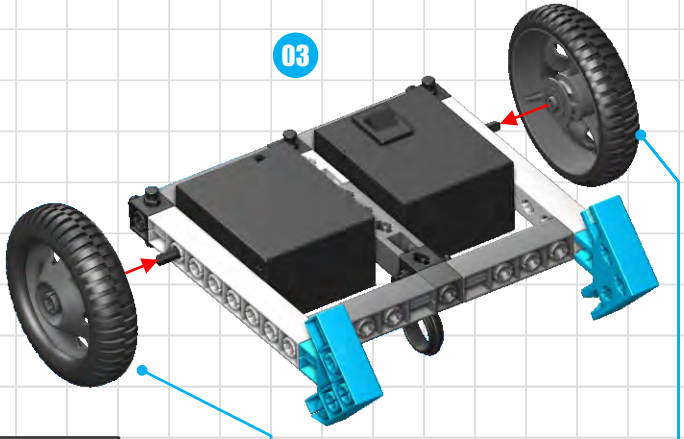
x2



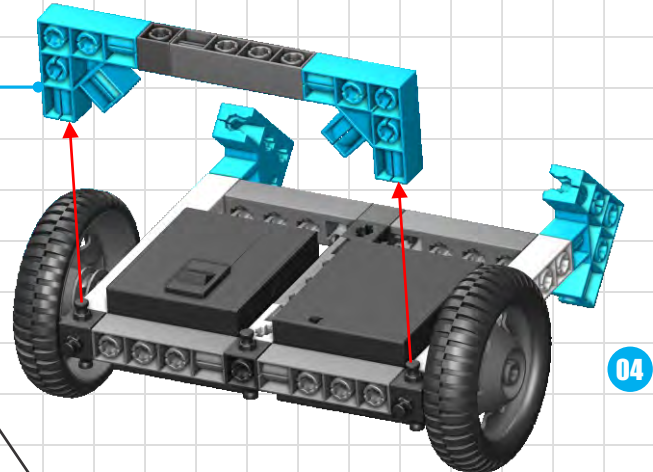
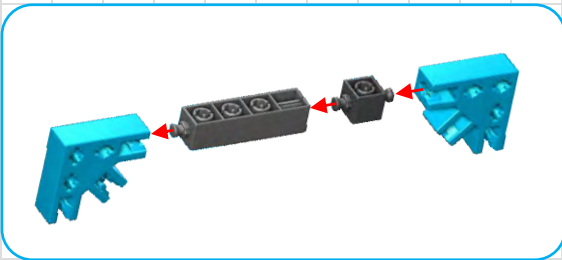
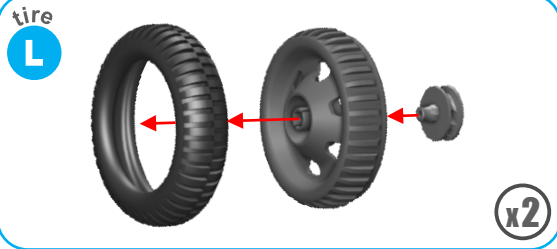
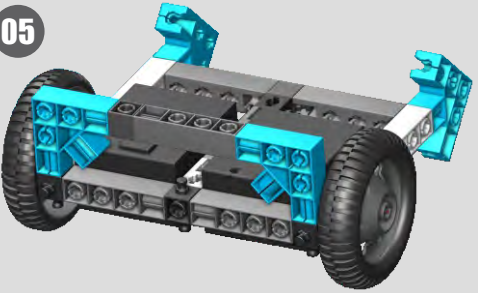
04



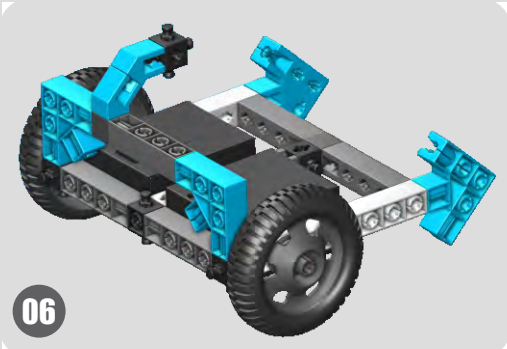
03



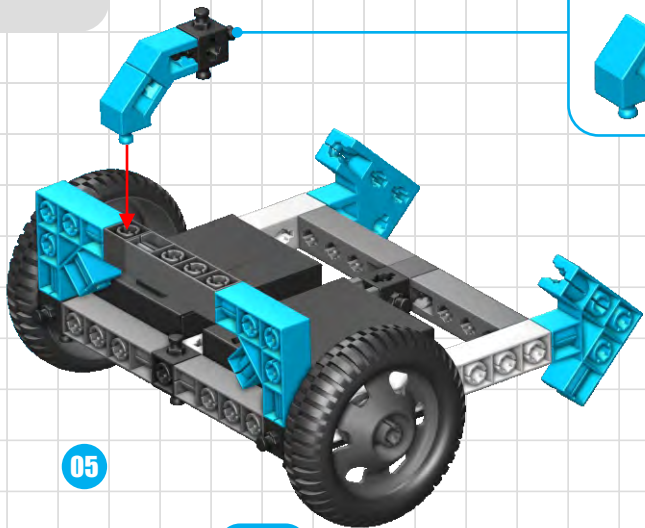
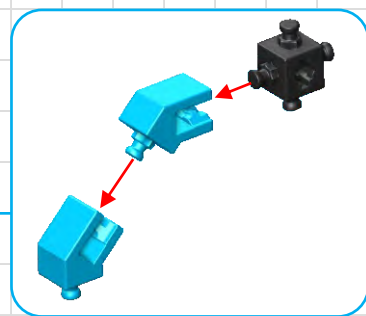
05



04

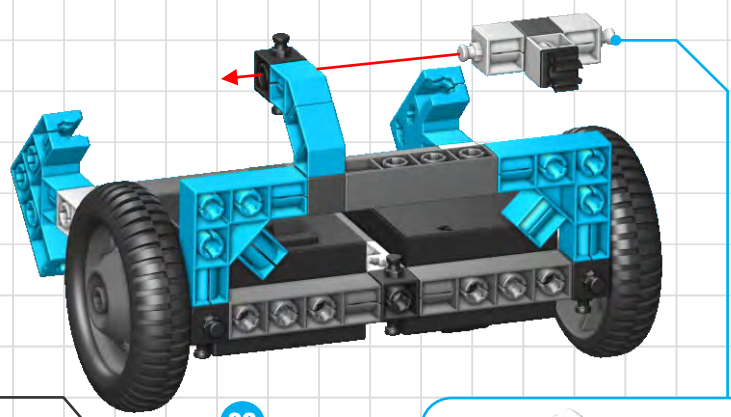
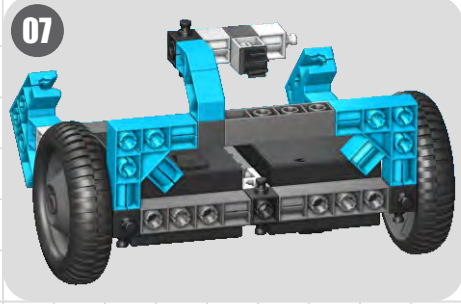


06

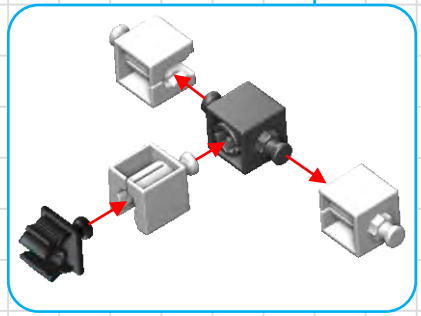


05

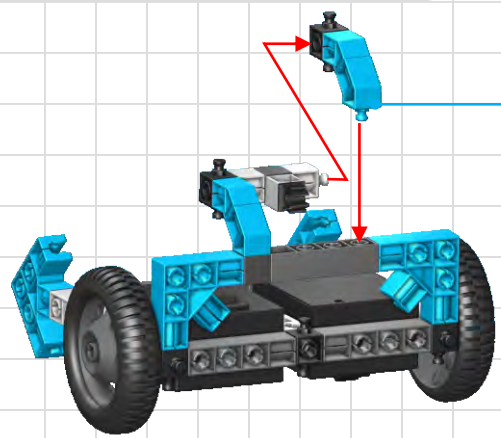
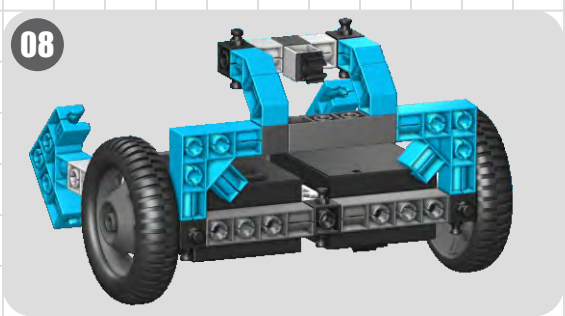
07



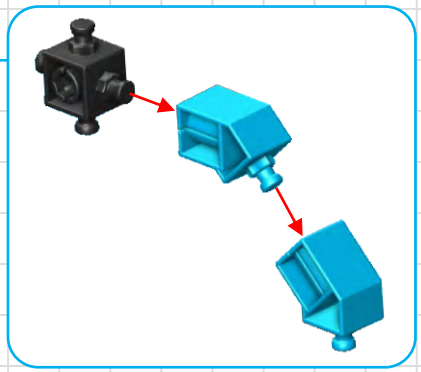
06



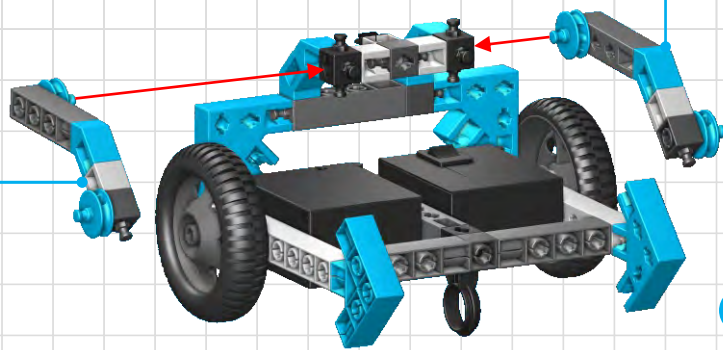
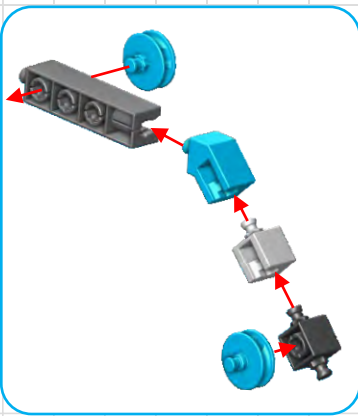
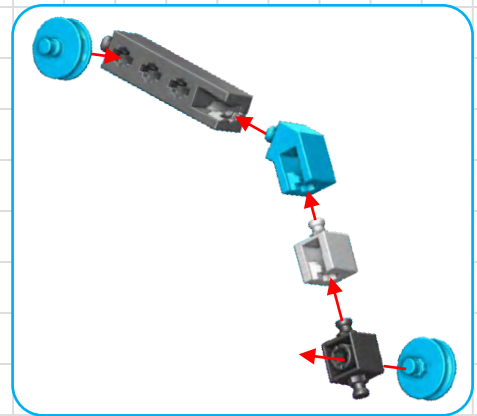
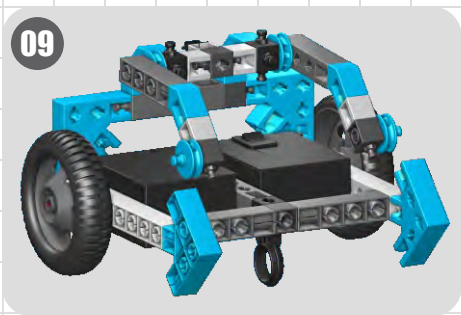
08



07

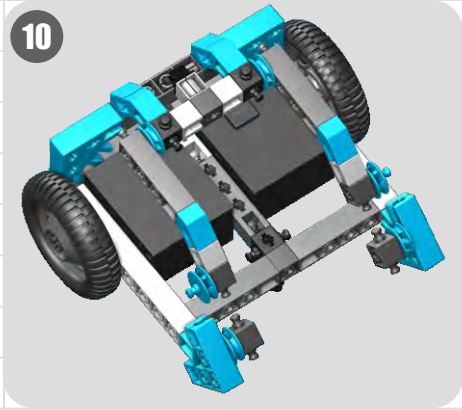


09

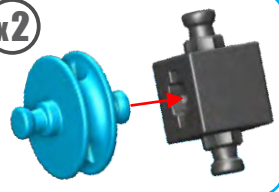


08

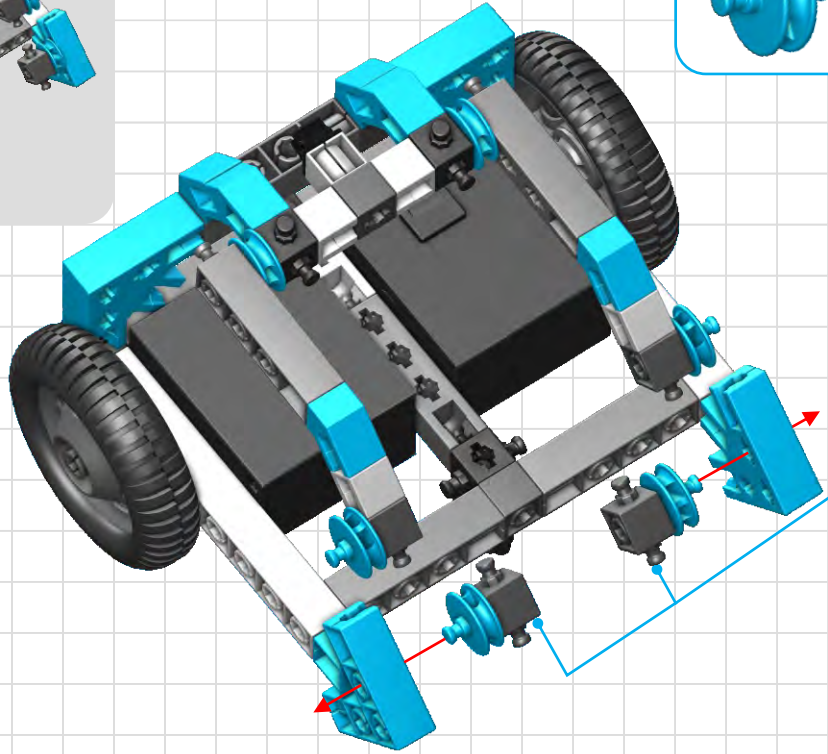
10



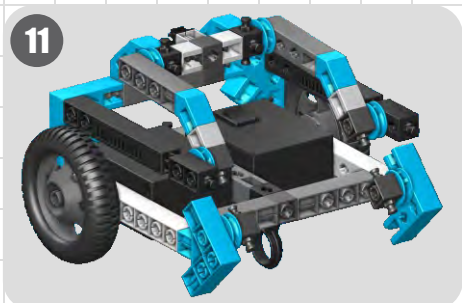
x2



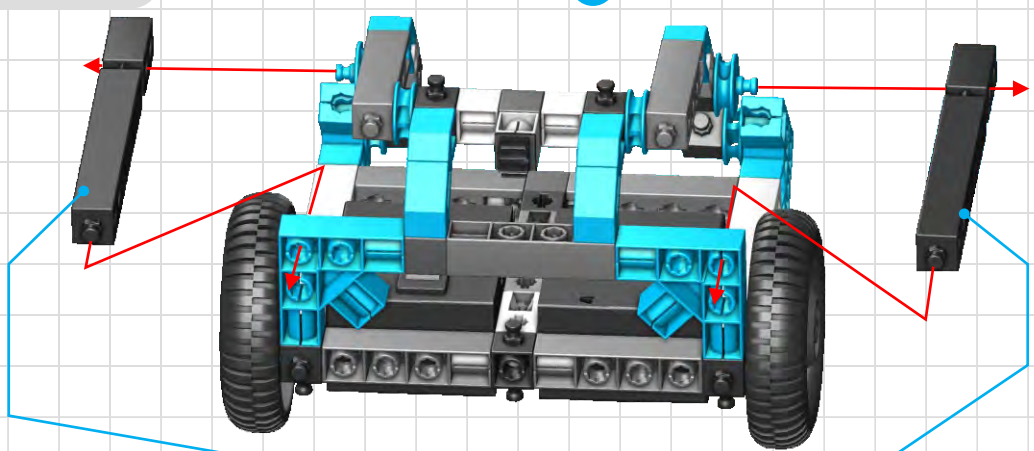
09



11

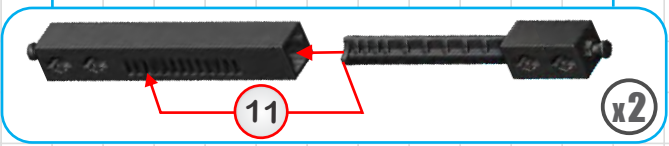


10

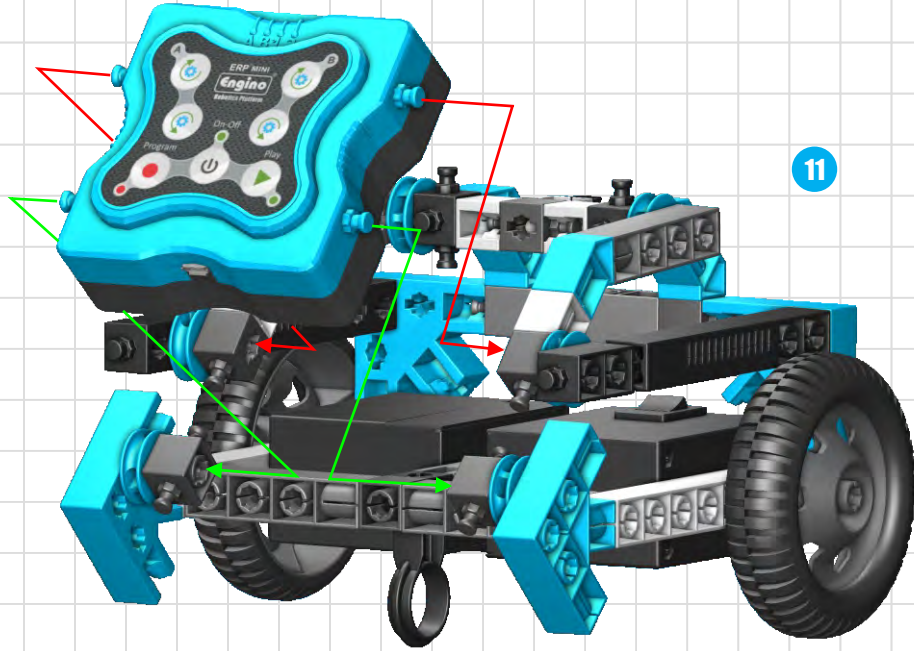


11

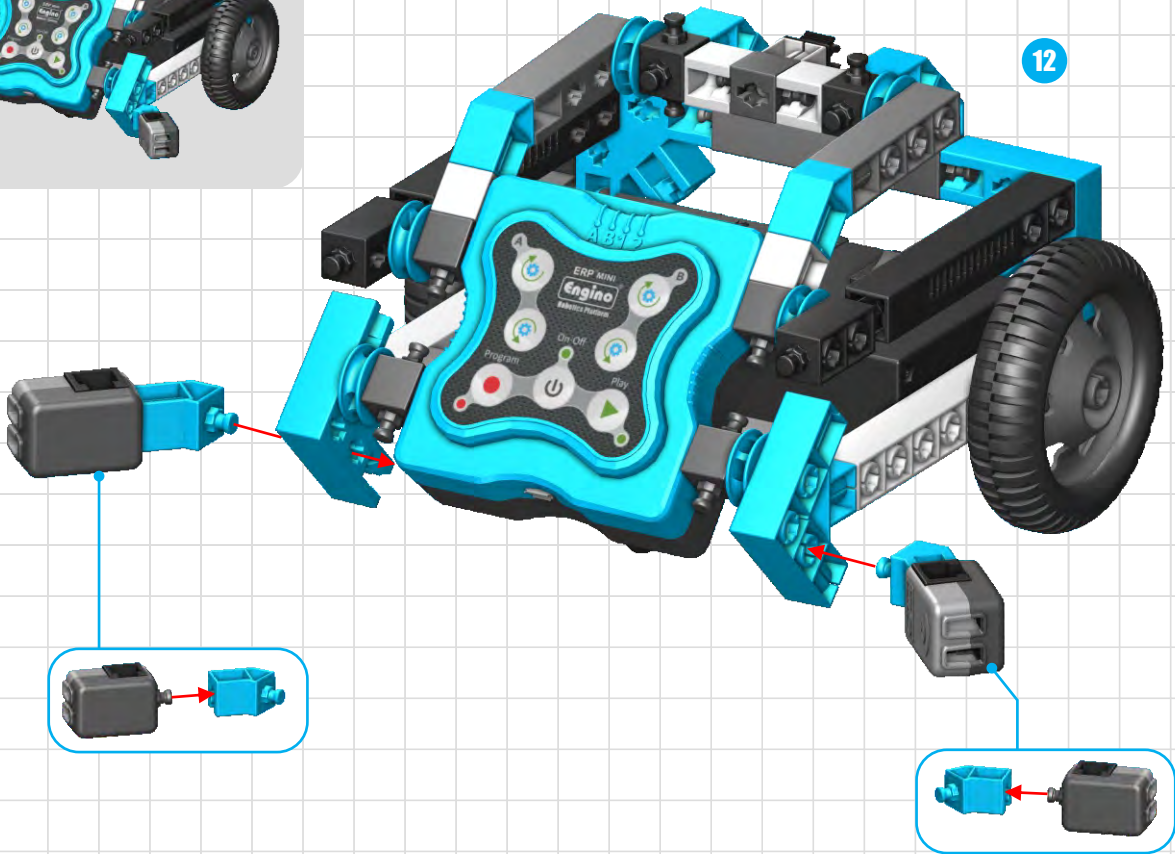
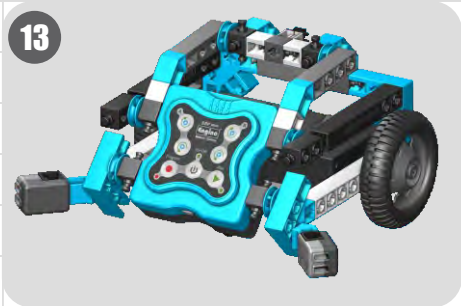
x2



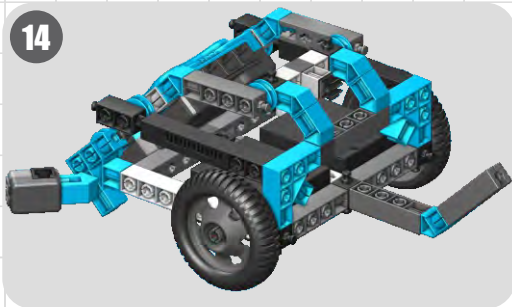
12



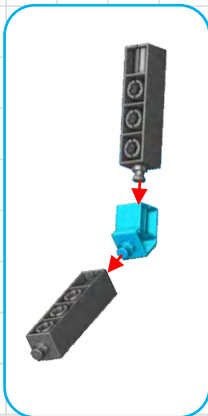
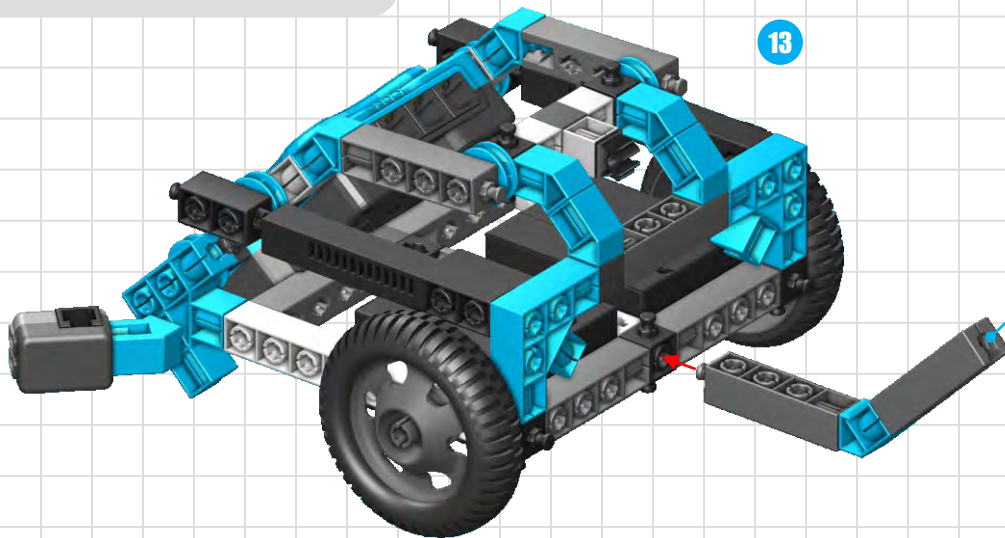
13



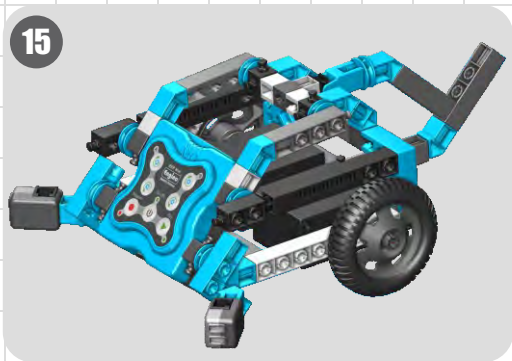
14



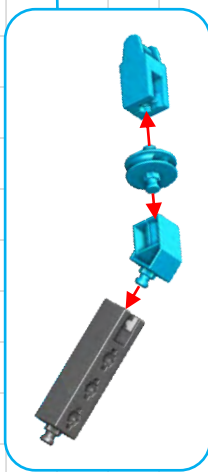
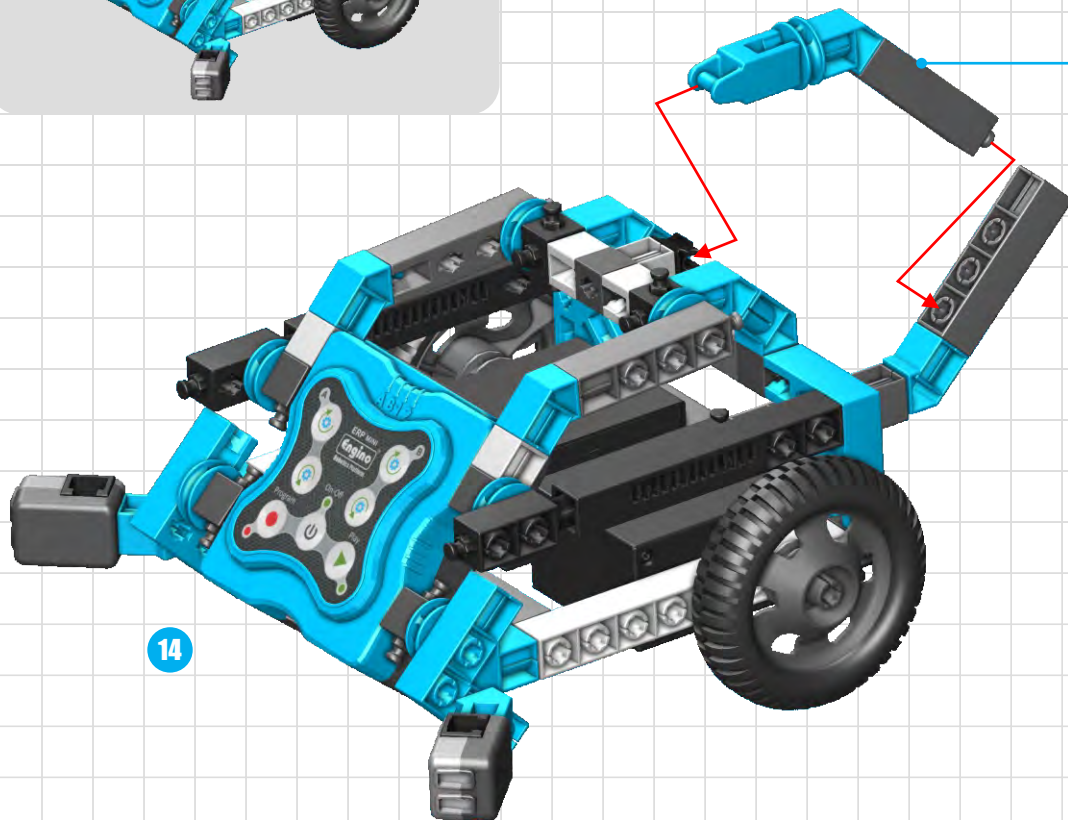
13



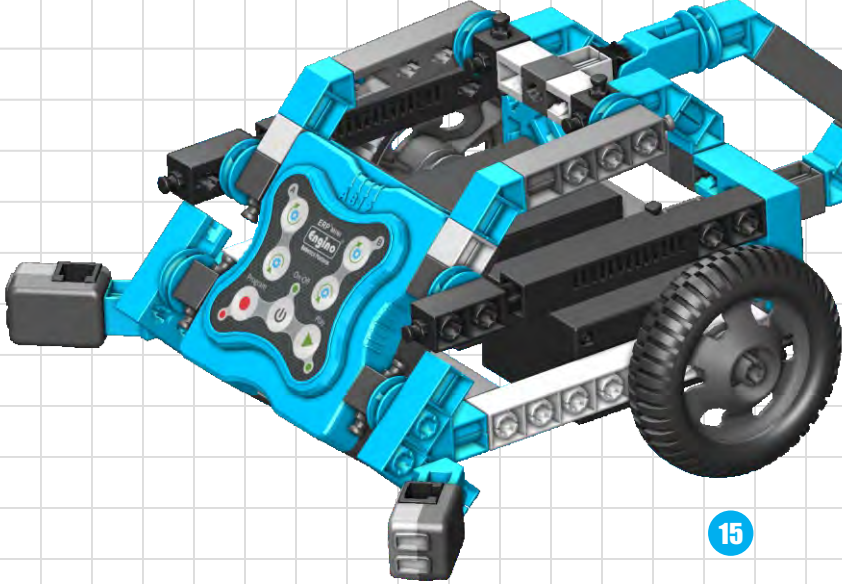
15



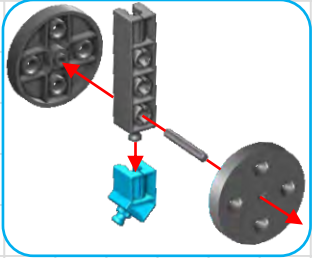
14



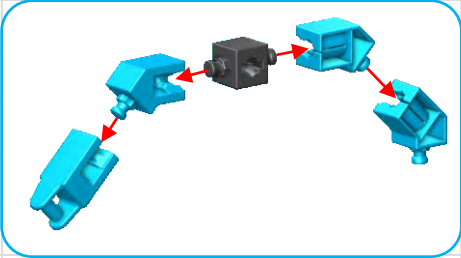
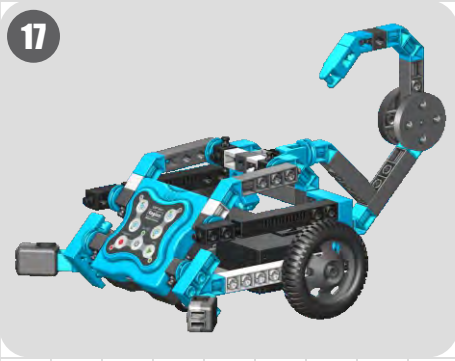
16



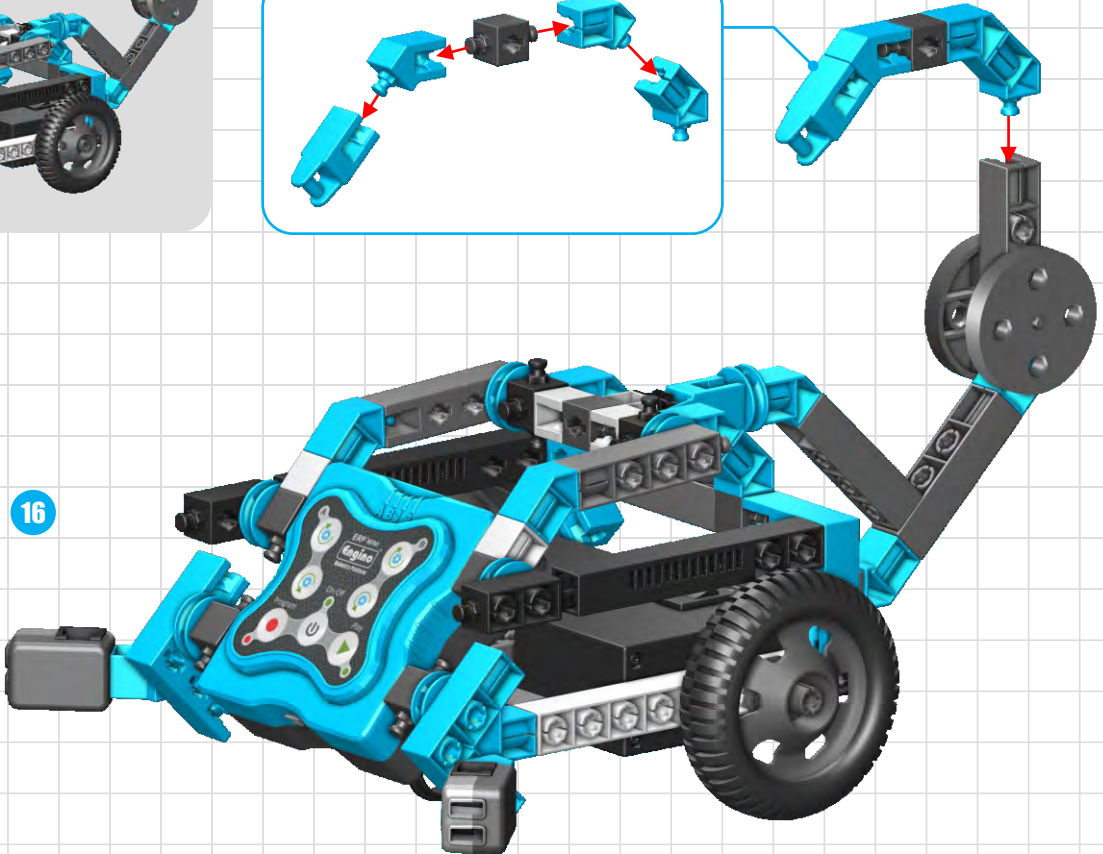
15



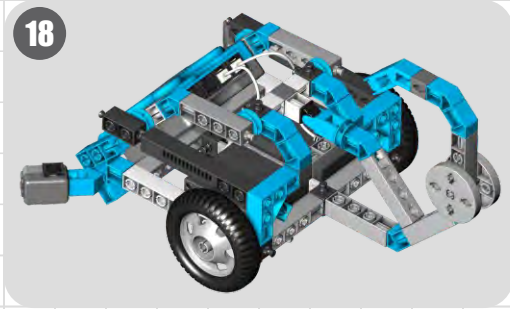
17



16

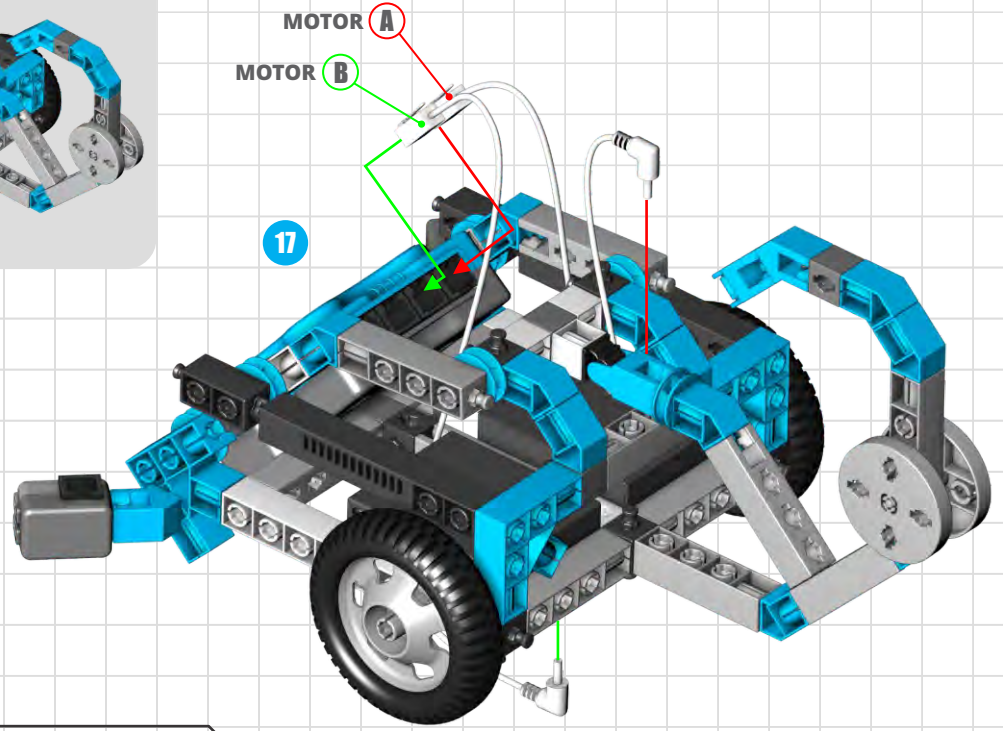


18



MOTOR A
MOTOR B

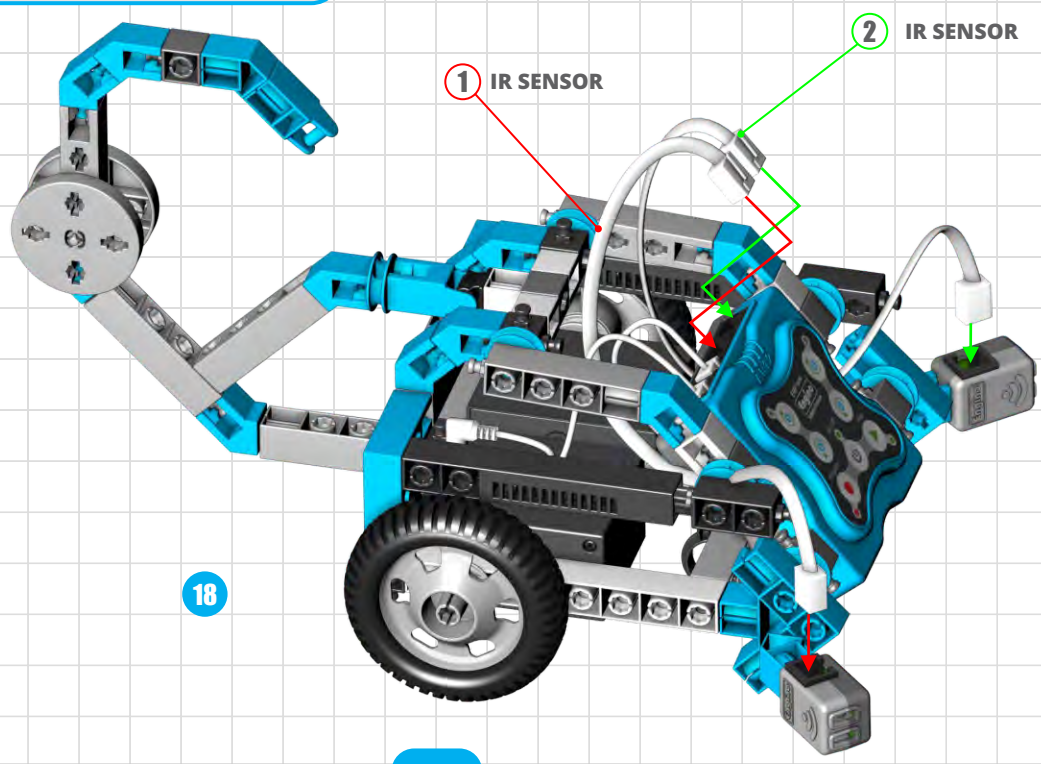
17



2 IR SENSOR

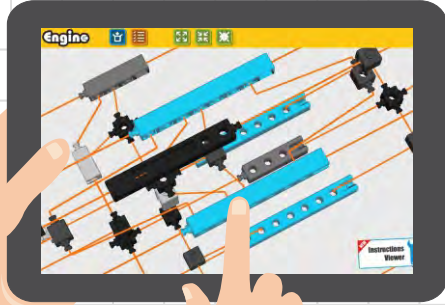
1 IR SENSOR

18



Ingyenes applikáció

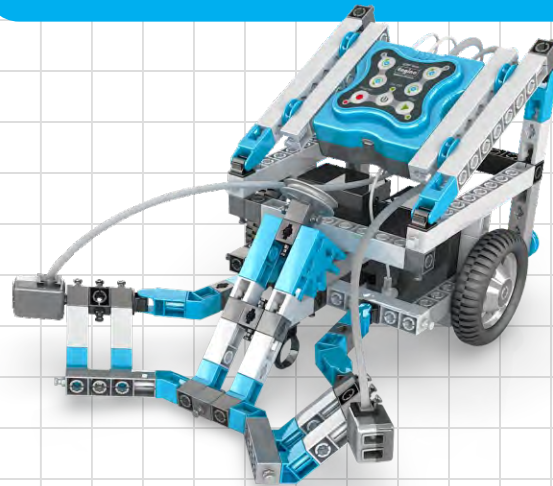
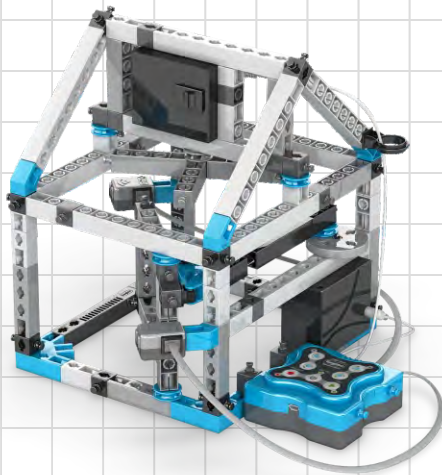
- A** Használd a számítógéped vagy tableted és kövesd a linket további modellekért:



www.engino.com/instructions/ce101mp-a

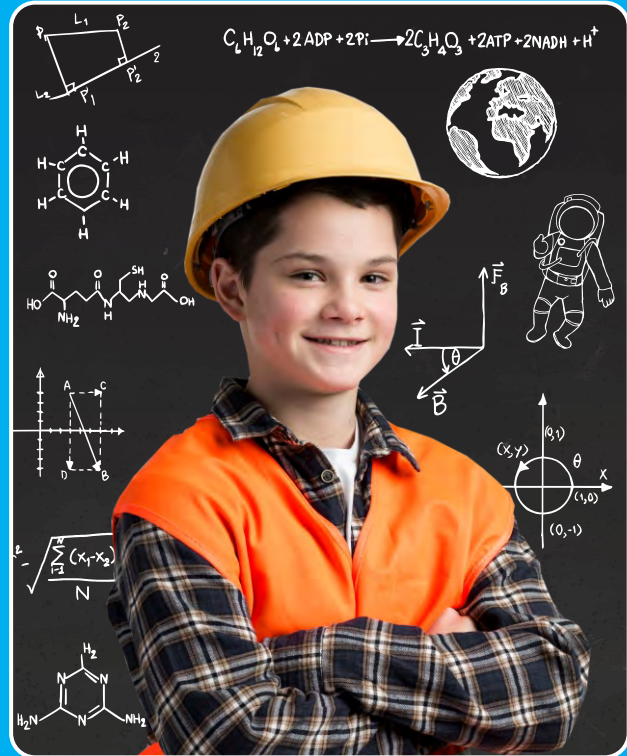
- B** Töltsd le az applikációt, hogy egy 3D-s nézetű, lépésről lépésre útmutatót kapj!

Engino kidCAD (3D Viewer) app:



Még több teória és kísérlet online!

www.engino.com/theory/ce101mp-a

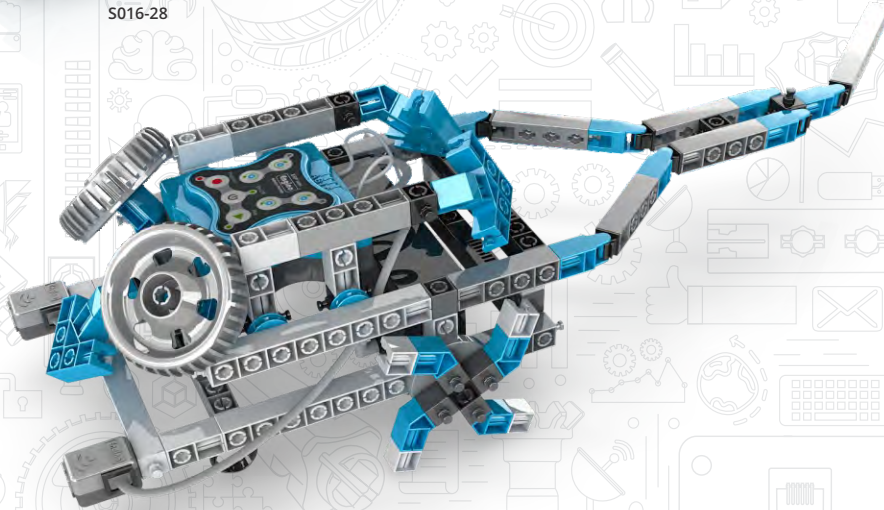
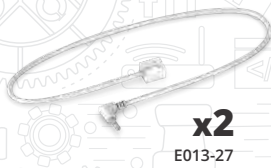
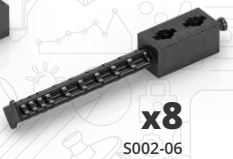
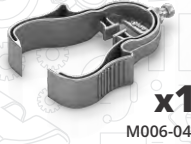
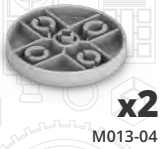


©123RF.com / baloncici, ella, James Steidl, Ales Zvolanek, Olga Serdyuk, spaxia, abidal, dipressionist, Inna Jacquemin, Miroslaw Dobrzanski, scyther5, hxdbxzy, gemenacom, pixinoo, Frederick, Joerg Hackemann, fastfun23, Ronalds Stikans.

Copyright © Engino-Net Limited

Minden jog fenntartva. Ezen oldalak egyetlen része sem használható fel más célra, csak személyes használatra. Ezért a sokszorosítás, módosítás, visszakeresési rendszerben történő tárolás vagy továbbítás bármilyen formában vagy bármilyen eszközzel, elektronikus, mechanikus vagy más módon, nem személyes használatból, a szerkesztő előzetes írásbeli engedélye nélkül szigorúan tilos.

Anyagjegyzék:



Forgalmazó:
ConQuest entertainment a.s.
Kolbenova 962/27d,
Praha 9
info@cqe.hu
www.toy.cz
tel: 284 000 120

FIGYELEM:

FULLADÁS VESZÉLY - Apró alkatrészek. 3 éven aluli gyermekek számára nem ajánlott.



Termékkód:
CE101MP-A