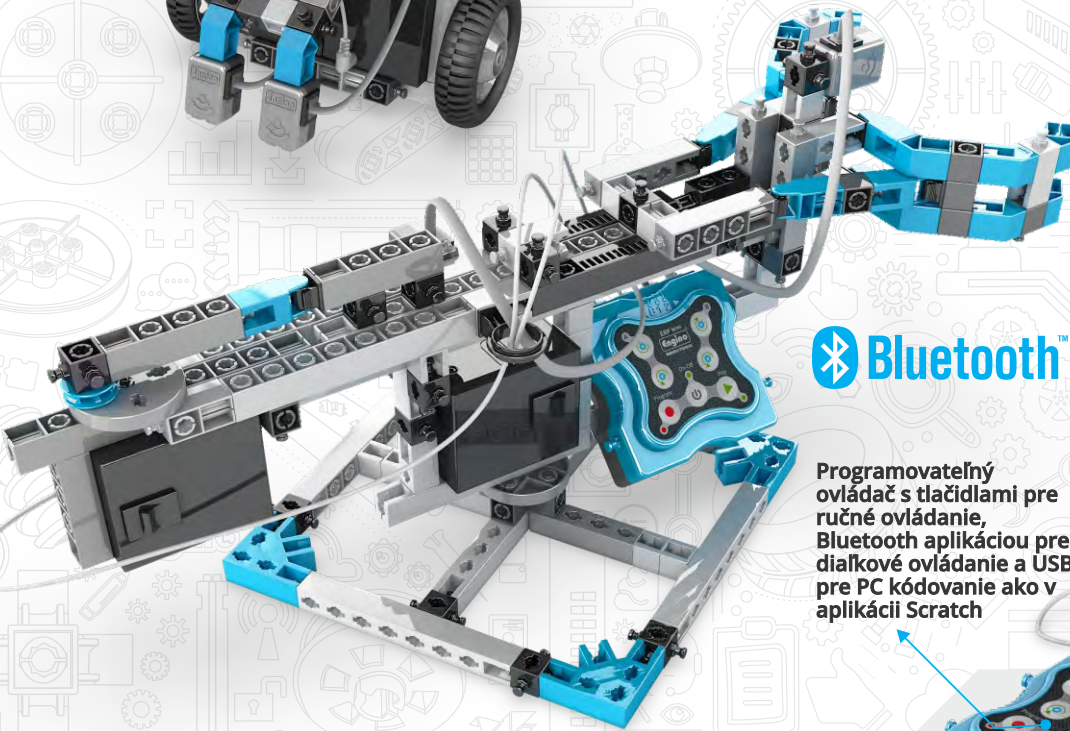


Engine

CREATIVE™ engineering

maker Pro



100 IN 1

robotized
MODELS

5
modelov

vytlačené
návodý
(v balení)



95
modelov

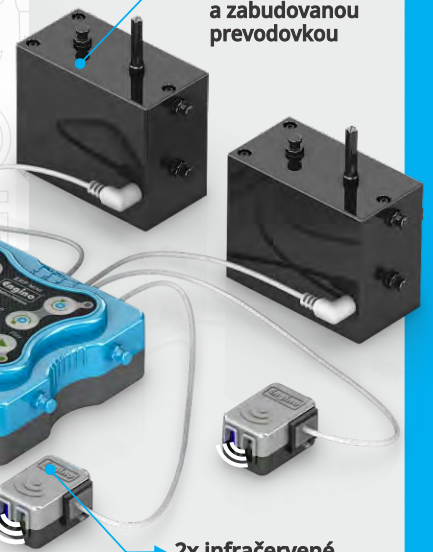
online
3D návody



Bluetooth™

Programovateľný ovládač s tlačidlami pre ručné ovládanie, Bluetooth aplikáciou pre diaľkové ovládanie a USB pre PC kódovanie ako v aplikácii Scratch

2x motory s vysokým krútiacim momentom a zabudovanou prevodovkou



**⊕ ELEKTRONIKA
A VÝKON MOTORA**

2x infračervené senzory pohybu

• INTERAKTÍVNE 3D NÁVODY

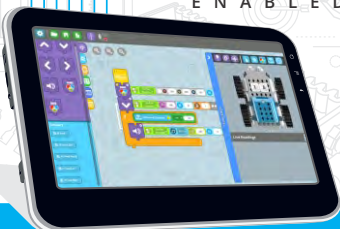
• STAVAJTE

• HRAJTE SA

• UČTE SA

• VYMÝŠĽAJTE

STEM
ENABLED



10+

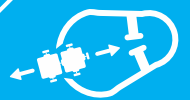
Vhodný vek



Návody
na stavanie

3

Roky záruky



Rozpojovací nástroj
Na jednoduché rozpojenie
malých súčiastok

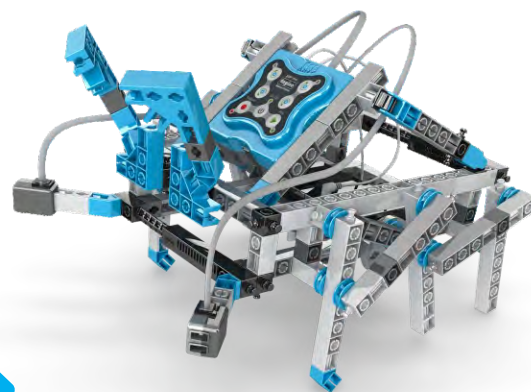
CREATIVE™ engineering **maker Pro**



postavte si ventilátor

Model ventilátoru vám predstaví hlavné programovacie pojmy a inovatívny ovládač MINI 2.0. Pomôže vám tiež otestovať rôzne premenné a programy vytvorené pomocou softvéru KEIRO.

- Čo sú vstupy a výstupy.
- Ako ručne naprogramovať robota.



postavte si hexapoda - šesťnohého robota

Experimentujte s pohybom nôh namiesto kolies s týmto zaujímavým modelom hexapoda! Nastavte na obe strany infračervené senzory, aby sa spustili pri blížiacej sa prekážke a robot sa jej tak pomocou svojich nôh vyhol.

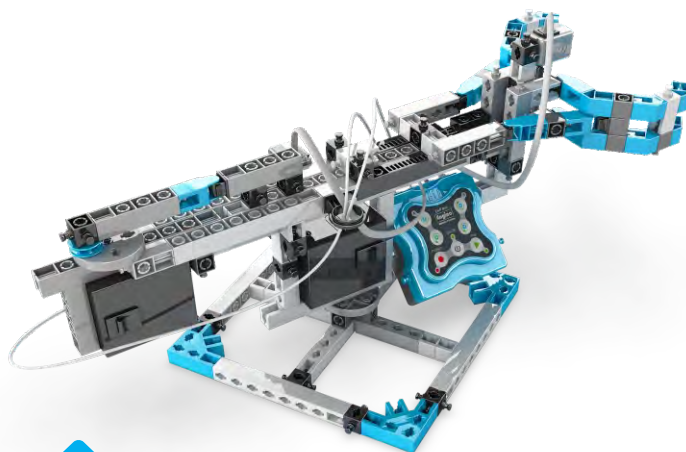
- Ako pracovať s pohybom nôh
- Čo je to príkaz IF.



postavte si robota

Postavte si model robota a využite jeho dvojité schopnosti! Pomocou programovateľných senzorov robot rozpozná čiernu a bielu farbu, čím môže ísť po čiare a rozpoznáť objekty na svojej ceste.

- Ako fungujú infračervené senzory.
- Čo sú to logické členy a príkazy WHILE



postavte si robota Twistera

Tento twister robot napodobňuje ozajstné robotické ruky, ktoré sa často používajú v priemyselných nastaveniach. Nastavte model tak, aby presne pracoval a presúval predmety pomocou technológie infračervených senzorov.

- Ako pracovať so stacionárnymi robotmi.
- Čo je príkaz REPEAT UNTIL.

Obsah



Teória

- 03** Čo sa naučíme
- 03** História robotiky
- 05** Definícia robota
- 07** Využitie v živote
- 09** Programovanie



Užívateľský manuál

- 11** Filozofia Engin Robotics
- 13** Ovládač MINI 2.0
- 14** Periférie, káble a porty
- 15** Príklad manuálneho programovania
- 16** Inštalácia softvéru KEIRO
- 16** Aktualizácia firmvéru ovládača a MINI 2.0
- 16** Rozhranie KEIRO
- 30** Programovanie vývojových diagramov
- 31** Ukážky kódov



Experimenty

- 33** Konštrukcia a programovanie
- 34** Robotický pohyb a IF
- 35** WHILE a logické členy
- 36** Príklad z praxe: priemyselné rameno



Návody k stavaniu

- 37** Tipy k stavaniu
- 38** Robot
- 43** Ventilátor
- 46** Hexapod
- 52** Robot Twister
- 58** Škorpión



Čo sa naučíme

"Hlboko vnútri je robotika o nás. Jedná sa o disciplínu napodobňovania našich životov, premýšľanie, ako to vlastne fungujeme." Tento citát od profesora matematickej informatiky Roda Grupena najlepšie popisuje, ako vnímajú robotiku vedci: ako simuláciu životných funkcií pomocou mechaniky a počítačov. Ale dokáže niekedy ľudstvo vytvoriť roboty, ktoré by mali city a rozhodovali by sa samostatne v situáciách, ktoré programovací jazyk nepredvídal? Ak už nás čaká v budúcnosti čokoľvek, dnešní mladí ľudia musia jasne vedieť, čo sú to roboty od základov robotiky a programovania po vyrobenie svojich vlastných funkčných zariadení! Nasledujúce stránky boli navrhnuté presne s touto predstavou!



Robotika by bez počítačového programovania nemohla existovať

Táto brožúra obsahuje rozsiahlu **teoretickú sekciu** so stavebnými výzvami a zaujímavými faktami, aby ste sa dozvedeli čo najviac o robotoch a ich využití v dennom živote. Brožúra obsahuje taktiež **užívateľský manuál**, ktorý do detailu vysvetľuje všetky aspekty inovatívneho robotického systému ENGINO. Objavte všetky vedecké zásady, a to experimentovaním, návody krok za krokom a fascinujúce úlohy. Riadte sa návodmi na postavenie zaujímavých robotických modelov, ako je napríklad **škorpión, améba, automatizovaný dom, ventilátor, hexapod, ukazovátka, lopata, twister, kocka a robot**. Veľa ďalších vzdelávacích materiálov nájdete online!



V hieroglyfoch boli objavené automatické zariadenia

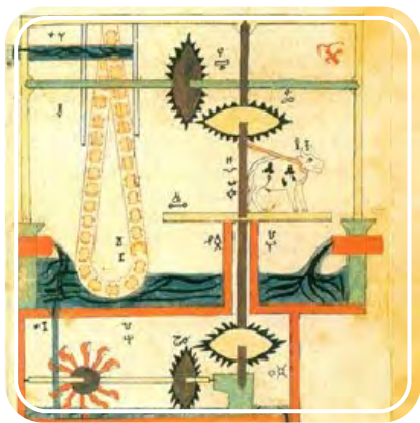
Antický vynálezcovia a remeselníci sa pokúšali o postavenie raných "robotov". Išlo o rôzne podoby od jednoduchých po komplexné zariadenie, ktoré plnili rôzne úlohy, zdanlivo autonómne. Preto sa im hovorilo automaty - z gréčtiny "konajúci svoju vlastnou vôľou", aj keď tomu tak nebolo. Už 4 storočie pr. N. L. Údajne Archytas vyrobil parný lietajúci stroj nazývaný "holub". Ľudia tej doby tvrdili, že jeho stroj preletí vzdialenosť 200 metrov! Ďalším vynálezcom bol Heron z Alexandrie (10-70 n. L.), ktorý mal mnoho kreatívnych nápadov ako napríklad automatické dvere od chrámu, sochy na vírenie vína alebo parou a vetrom poháňané stroje ako jeho slávny "aeolipile".

História robotiky

História robotiky začína už v antických dobách. Od antiky ľudia premýšľali nad umelými konštrukciami, ktoré by nahradili človeka v rôznych úlohách so svojim vlastným rozumom. Jeden grécky mýtus rozpráva príbeh Talosa, obrovského humanoida z bronzu, ktorého postavil Hefaistos (grécky boh kováčstva), aby chránil ostrov Kréta. Úvahy o tvoroch podobných robotom sa dochovali po celom svete - v Nórsku, na Blízkom východe, v Indii a Číne - v rôznych podobách: texty, kresby, maľby a dokonca v egyptských hieroglyfoch.



Moderná replika Heronovho stroja - aeolipile



Al-Džazářího kresba reťazového čerpadla poháňaného vodou



V stredoveku nájdeme mnoho príkladov automatov, väčšinou vo forme zvierat alebo ľudí. Ľudia stavali zábavné stroje, ktoré boli naprogramované na jednoduché úlohy, ako posúvanie nôh alebo otáčanie hlavy pomocou mechanizmov otáčavých koliesok. Moslimský inžinier Al-Džazáří (1136-1206) opísal vo svojej knihe "Kniha vedomostí dômyselných mechanických zariadení" 100 automatických mechanických strojov. Neskôr hral vo vývoji robotiky dôležitú úlohu francúzsky vynálezca a umelec Jacques Vaucanson (1709-1782), ktorý sa snažil ukázať, ako fungujú veci v prírode. Niektoré jeho vynálezy ukazovali postavy, ktoré hrali na ozajstné nástroje (flautu a tamburína), a slávnu "tráviacu kačicu", ktorá napodobňovala trávenie potravy.

V roku 1898 ukázal srbsko-americký vynálezca Nikola Tesla prvé rádiom riadené plavidlo. Skutočné roboty, čiže také, ktoré boli schopné prijímania spätnej väzby od svojho okolia a zároveň reakcia na ňu, sa objavili v 20. storočí. V roku 1948 vytvoril Grey Walters malé robotické korytnačky so svetelnými a dotykovými senzormi, ktoré tak mohli hľadať "jedlo". O niekoľko rokov neskôr si George Devolet nechal patentovať svojho robota Unimate, ktorý zvládol vykonávať priemyselnú prácu. Bol nainštalovaný v roku 1961 do továrne, aby zdvíhal horúce kusy kovu z tlakového lejacieho stroja a ukladal ich na seba.



Robotické rameno Unimate



Vedeli ste, že ...

Spisovateľ science fiction Isaac Asimov zostavil "**tri zákony robotiky**" vo svojej poviedke "Hra na naháňačku" vydané v roku 1942. Tieto pravidlá znejú: 1) Robot nesmie ublížiť človeku alebo svojou nečinnosťou dopustiť, aby bolo človeku ublížené.; 2) Robot musí poslúchnuť príkazy človeka, okrem prípadov, keď sú tieto príkazy v rozpore s prvým zákonom.; 3) Robot musí chrániť sám seba pred poškodením, okrem prípadov, keď je táto ochrana v rozpore s prvým, alebo druhým zákonom.



Isaac Asimov (1920 - 1992)

V 60. rokoch 20. storočia inžinieri riešili problém robotického pohybu vytváraním robotických ramien a nôh inšpirovaných prírodou. V 70. a 80. rokoch 20. storočia umožnili pokročilé technológie a malé počítačové súčiastky použitie všetkých častí priamo na robota namiesto ich prepájania káblami na externom počítači. Od 90. rokov 20. storočia roboty ešte viac pokročili a spracovávali ďalšie komplexné úlohy. V dnešnej dobe sa využívajú v každom aspekte ľudského života od domáceho vybavenia a asistencie (ako napríklad AIBO robotický pes, 1999 a Roomba, robotický vysávač, 2008) po priemyselné využitie (napríklad pri výrobe áut) a prieskumy (ako Epson, lietajúci robot, 2004) dokonca aj vo vesmíre (prieskumné roboty na Marse, 2004). Avšak toto všetko je zatiaľ len začiatok, pretože budúcnosť robotiky vyzerá veľmi sľubne!



Dva typy vysávačov Roomba

Definícia robota

Pojem "robot" predstavil prvýkrát český spisovateľ Karel Čapek v roku 1920 vo svojom dráme R.U.R. podľa slova významu "práca" v slovanských jazykoch. To nám ukazuje, ako ľudia spočiatku chápali predstavu robotov: "stroje, ktoré vykonávajú úlohy, ktoré sú príliš nebezpečné, komplikované alebo jednoducho nudné pre človeka". Avšak vedeckejšou definíciou je, že robot je mechanický stroj, ktorý vníma spätnú väzbu od svojho okolia a je schopný na ňu vhodne reagovať. Tým pádom roboty majú - do určitej miery - naozajstné autonómne správanie a teoreticky premýšľajú a správajú sa podľa seba. Toto však závisí od ich programu a prednastavených príkazov.



Robotické dvere s dotykovým senzorom v metre

Druhy robotov

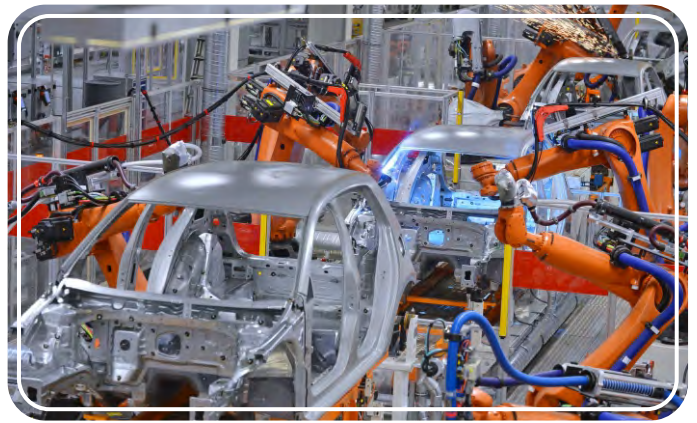
Roboty sa vyrábajú v rôznych podobách a tvaroch a zvládajú jeden alebo viacej úloh vďaka rôznym nastaveniam a podmienkam. Preto môže jeden robot patriť aj do viacerých kategórií. Takže ako rozpoznáme jedného robota od druhého? Môžeme využiť kategorizácie podľa funkcie robota. To nám ukáže užitočnosť robota a zároveň ako by mal asi vyzeráť, či aké súčiastky by mal mať, aby fungoval správne. Hlavnými typmi podľa funkcie sú:



Priemyselné robotické rameno

Priemyselné roboty sú automatizované roboty, ktoré sa využívajú v továrňach na nosenie ťažkých predmetov alebo plnenie rutinných úloh - napr. Zváranie, maľovanie a manipulácia s materiálmi.

Zábavné roboty sú väčšinou vyrobené pre zábavu a nemajú praktické využitie, napríklad robotické hračky.



Robotické ramená sa využívajú pri skladaní áut

Napríklad automaticky otvárajúce sa dvere (v obchodoch, hoteloch a podobne) sú robotické. Majú senzory pohybu, ktoré neustále vysielajú infračervený signál. Pokiaľ je tento signál narušený niekým, kto sa postaví pred neviditeľný lúč, centrálny procesor vydá príkaz dverám, aby sa otvorili. Po niekoľkých sekundách sa dvere zatvoria a postup sa opakuje. Ďalšími príkladmi robotov sú napríklad domáce systémy vykurovania a svetla, čo sa zapínajú pohybom. Napríklad autá a hračky na diaľkové ovládanie naopak nie sú roboty. Iba prijímajú priame rozkazy k pohybu cez ovládač. V tomto prípade sa nejedná o spätnú väzbu, nerozhodujú sa sami, keď ide o objekty okolo nich - napr. nezastavia pred múrom.



Robotický čistič bazénov



Robotická hračka

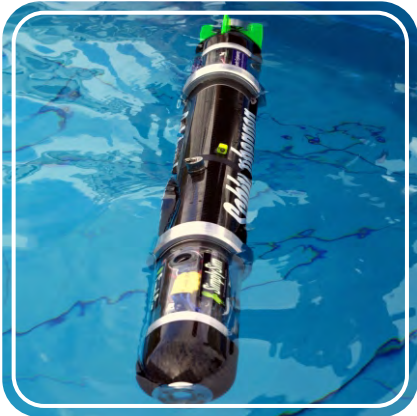


Robot na likvidáciu bômb

Lekárske roboty sa využívajú v zdravotníctve (v nemocniciach, lekárňach, klinikách a zdravotných centrách) pri tréningu doktorov alebo aj pri procedúrach, ktoré vyžadujú presnosť (operačný robot).

◀ **Vojenské roboty** sú súčasťou pokročilých armád a polície. Využívajú sa namiesto ľudí, väčšinou v nebezpečných situáciách, preprava munície, zneškodňovanie bômb alebo dokonca vojnové boje, napr. malé tankové roboty.

Domáce a servisné roboty sú navrhnuté pre komerčné využívanie doma alebo v práci, kde nám nejakým spôsobom uľahčujú život, napr. robotický vysávač.



Vodné roboty

◀ **Prieskumné roboty** sa využívajú pri preskúvaní drsných prostredí, ktoré sú pre ľudí príliš náročné. Ako podzemné jaskyne, oceány (napríklad vodné roboty) alebo iné planéty.

Virtuálne roboty sú roboty, ktoré sa používajú vo virtuálnych podmienkach. Spravidla ide o určitý typ veľkých okuliarov s obrazovkami, ktorými si zakryjeme oči, a špeciálnymi rukavicami, aby ste mohli kedykoľvek pozeráť a pracovať vo svete virtuálnej reality.



Vybavenie na virtuálnu realitu



Vedeli ste, že ...

Existuje špeciálny futbalový turnaj pre roboty - "RoboCup"! Prvé roboty, ktoré sa zúčastnili robotického futbalu boli vyrobené v Japonsku v roku 1997. Hráči zvládali lokalizovať loptu a kopnúť do nej nohami. Výrobcovia týchto robotov si dali za cieľ, že v roku 2050 sa robotický futbalový tím stretne s víťazným tímom Majstrovstiev sveta vo futbale a vyhrá! Myslíte si, že dotedy dosiahnu robotické technológie takú úroveň futbalových zručností?



Robotický futbalisti

Hobby roboty sú roboty, ktoré boli vyrobené nadšencami robotiky ako veľkými spoločnosťami. Ich účelom je testovanie rôznych súčiastok a technológií alebo iba jednoducho relax ako je napríklad lietajúci dron.

Súťažné roboty sú vyrobené na súťaženie s inými robotmi v rôznych výzvach. Táto kategória je podobná tej predošlej, pretože súťaže väčšinou organizujú fanúšikovia robotov a chcú na nich otestovať robotie zručnosti či získať cenné skúsenosti. Súťaže sa tiež konajú pre vývojárske tímy z celého sveta, aby sa mohli zdokonaľovať vo svojich návrhoch.



Robot lietajúci dron

Využitie v živote

21. storočie je zaiste dobou robotičkej vedy a pokročilých technológií! Roboty nájdeme všade, a to nielen tie humanoidného typu, ktoré si ľudia vždy predstavujú, ale často vo veľmi jednoduchých formách ako trebárs systémy vykurovania v domoch alebo senzory vzdialenosti v autách. Využitie robotických systémov v každodennom živote je rozsiahle a ľudia sú na nich viac a viac závislí. Prečítajte si nižšie o niektorých charakteristikách využitia robotov a predvedte sa postavením a naprogramovaním navrhovaných modelov od Engino Robotics.

Stavanie **Výzva**

Návody nájdete online

Vyhýbanie sa zrážkam: moderné vozidlá sú vybavené rôznymi senzormi, aby vodiči nemuseli premýšľať nad toľkými vecami pri riadení. Detektor dažďa môže spustiť stierače pri daždi alebo detektor svetla môže rozpoznať, či je deň alebo noc a zapnúť tak správne svetlá. Čo sa týka bezpečnosti, tak veľa áut má senzory vpredu a vzadu senzory vzdialenosti, ktoré varujú vodiča, keď sa môže schyľovať k zrážke, hlavne pri parkovaní alebo prechádzaní úzkymi priestormi. V situáciách, kedy je pri aute človek alebo zvierka a vodič si ich nevšimne, môžu tieto senzory aj zachrániť život!

Postavte si Engin model "Ukazovadlo" a experimentujte s vyhýbaním sa kolíziám. Nasmerujte infračervené senzory smerom dole, aby nasledovali určitú cestu, alebo ich otočte do strán, aby sa vyhýbali prekážkam z ľavej či pravej strany vozidla.



Engino® "pointer" model

Stavanie **Výzva**

návody nájdete online

Robotické domy: moderné domy začínajú byť čím ďalej "múdrejšie" a veľa predtým manuálnych funkcií sa deje automaticky! Napríklad automatické dvere sa otvárajú a zatvárajú automaticky pomocou detektorov pohybu a senzory tepla merajú výkyvy v teplote, aby mohli zapnúť systém chladenia alebo vyhrievania. Inžinieri experimentujú s inteligentnými chladničkami, ktoré by sa dokázali samé doplniť v prípade, že by im chýbali špecifické produkty (mlieko, vajčička, maslo), a objednali by si ich online!

Experimentujte s automatizáciou domu postavením Engin modelu "automatizovaný dom" (návod nájdete online). Zapnite dvere a ventilátor pomocou infračervených senzorov.



Engin model "automatizovaný dom"



Vedeli ste, že ...

12. novembra 2014 pristál prvýkrát v histórii robotický modul Philae na povrchu kométy! Tento obdivuhodný počin bol súčasťou vesmírnej misie Rosetta, pri ktorej od januára 2014 obiehala robotická vesmírna sonda okolo skúmanej kométy 67P. Cieľom misie bolo zistiť, či sa na kométe neskrývajú nápovedy k vzniku slnečnej sústavy a života na Zemi.



Reprezentácia vesmírnej misie Rosetta

Stavanie Výzva

Nebezpečné misie: roboty úspešne nahrádzajú ľudí v nebezpečných situáciách. Ľudia sa tak nemusia vystavovať riziku a misie bývajú splnené rýchlejšie a lepšie vďaka pokročilým robotickým schopnostiam. Robot napríklad rozpozná nebezpečné materiály, ktoré môže s ľudskou pomocou na diaľku zneškodniť. Operátor využíva na sledovanie a manipuláciu s bombou senzory robota.

Postavte si Engin model "Robot" (stránky 38-42) a umiestnite infračervené senzory dole, aby nasledovali určitú cestu, alebo dopredu, aby identifikovali "nebezpečné" objekty, pred ktorými budú varovať pohybom.



Engin model "Robot"

Prieskum vesmíru

Ľudstvo vždy fascinovali divy vesmíru a možnosť preskúmať nové svety alebo objaviť mimozemský život. Avšak podmienky mimo zemskú atmosféru môžu byť pre ľudí zničujúce: slnečné búrky, nedostatok kyslíka, gravitačné polia a teploty na absolútnej nule (-273 stupňov Celzia) sú len niekoľkými z možných nebezpečenstiev. Všetky vyžadujú veľmi drahé vesmírne obleky a roky tréningu a plánovania na malý výlet okolo Zeme alebo na Mesiaci. Čo sa vzdialenosti týka (napríklad pri ceste na inú planétu), musia byť prieskumníci ochotní vrátiť sa na Zem aj za niekoľko desaťročí alebo vôbec!

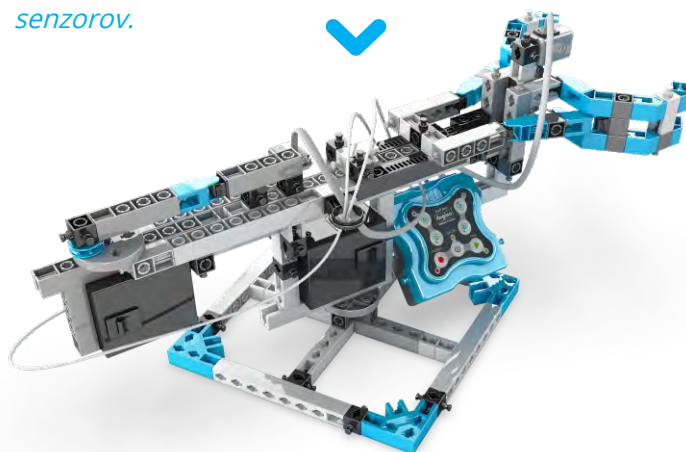


Mars Rover "Opportunity"

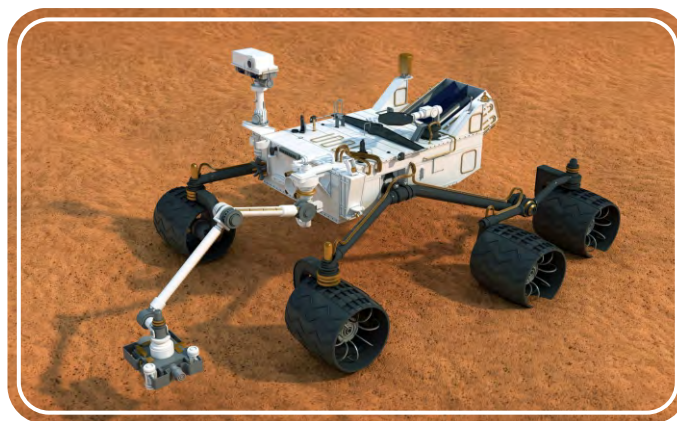
Stavanie Výzva

Priemyselné využitie: ďalším rozširujúcim sa využitím je priemyselný sektor, kde roboty nahrádzajú prácu ľudí. Ide väčšinou o stacionárne alebo vedené pohybujúce sa roboty s ohybným ramenom na zdvíhanie ťažkých predmetov alebo plnením precíznej práce (napríklad zváranie). Sú ideálne do tovární na montážne linky. Ľudia sa boja, že automatizované postupy povedú k zníženiu pracovných pozícií, tak to ale nie je, keďže vznikajú nové pozície na údržbu a opravu robotických strojov a zároveň sa tým zlepšuje kvalita produktov.

Postavte si Engine "Twister" (stránky 52-57) a riadte jeho pohyb pomocou technológie infračervených senzorov.



Engin model "Twister"

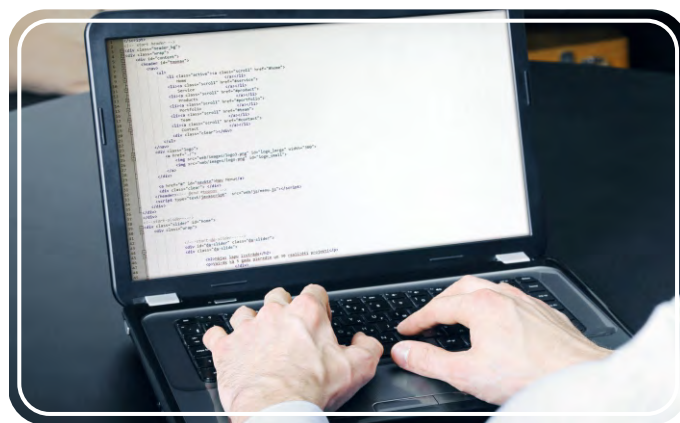


Mars Rover "Curiosity"

Našťastie existujú roboty! Tie sú ideálne na vesmírny prieskum, pretože nemajú problémy týkajúce sa prežitia a nevedia im stráviť na misii dlhšiu dobu. S ich pokročilými schopnosťami môžu preskúmať povrch planéty, analyzovať miestnu pôdu a atmosféru, hľadať vodu a ďalšie chemické zlúčeniny a tiež poslať späť na Zem veľa cenných fotografií vo vysokom rozlíšení. Momentálne sú aktívne dva prieskumné Rovy na Marse - Opportunity a Curiosity, a mnoho ďalších robotov, ktoré lietajú okolo asteroidov, komét a planét, a reprezentujú tak naše ľudstvo!

Programovanie

Okrem mechanických súčiastok je zásadným pre fungovanie robota tiež programovanie. Ako už sme spomenuli, zariadenie je považované za robotické, len ak je schopné reagovať, komunikovať so svojim okolím. Robot rozlúšti prijaté informácie a rozhodne sa pre príslušnú akciu podľa svojho programu pomocou mikro ovládača. Výrobcovia robotov často vymýšľajú svoje vlastné programovacie jazyky, a preto existuje mnoho typov robotických softvérov. Našťastie existuje pár základných pojmov, ktoré by sme mali pri vyvíjaní programu poznať, a tie sú nižšie popísané.



Pracujúci programátor

Programovacie jazyky

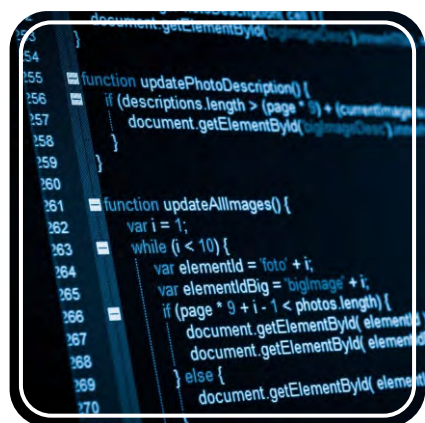
Najdôležitejším prvkom akéhokoľvek počítača alebo robotického vývoja je programovací jazyk. Všeobecne je programovací jazyk takýto súbor gramatických pravidiel a slovníček inštrukcií pre robota, aby plnil určité úlohy. Jazyk sa skladá z dvoch hlavných častí: z formy (syntaxu) a významu (sémantiky) textu. Existujú dva všeobecné druhy programovacích jazykov: nižšie, ktoré používajú stroje na chápanie (strojový kód s číslami a symbolmi) a vyššie, ktoré obsahujú ľudskú reč (v textovej podobe) a automatizáciu, čím uľahčujú proces programovania, ktorý je zároveň nám zrozumiteľný. V počítačovom softvéri sa spustiteľný súbor nazýva "zdrojový kód". V zdrojovom kóde sú algoritmy, ktoré sa skladajú z operácií popísaných krok za krokom, ktoré tvoria jednotlivé programovacie prvky, ktoré sa nazývajú príkazy (vysvetlené nižšie).

Každý, kto by si chcel vytvoriť počítačový program, by si mal vybrať programovací jazyk veľmi opatrne, pretože sú navrhnuté a používané pre rôzne účely. Najbežnejšími vyššími jazykmi sú:

- C ++ je pravdepodobne najpoužívanejší jazyk, v ktorom je vyvinutá väčšina aplikácií.
- Java sa využíva v zariadeniach, v ktorých je nainštalovaný Java Virtual Machine (JVM) nezávisle na systéme.
- Javascript je webový jazyk využívaný na vykonávanie operácií na počítačoch používateľov miesto sieťového servera.
- Python je považovaný za najjednoduchší programovací jazyk na naučenie.



Jazyk HTML je využívaný konkrétne pri vytváraní webových stránok



Príkazy sa zapisujú ako riadky textu

Príkazy

V počítačovom programovaní je príkaz prakticky inštrukcia, ktorá hovorí počítaču, čo má robiť. Asi si dokážete predstaviť, že tieto príkazy musia byť dostatočne presné, aby robot správne fungoval. Zároveň sekvencie príkazov čiže radenie úloh je rovnako dôležité. Predstavte si, že sa riadite návodom na pečenie torty. Ak sú kroky receptu nesprávne alebo sú napísané v nesprávnom poradí, pravdepodobne budete mať v kuchyni len bordel! V textových programovacích jazykoch sú príkazy väčšinou napísané riadok po riadku a obsahujú špeciálne znaky pre rôzne úlohy. Toto simuluje softvér KEIRO, v ktorom je okno (KEIRO Code), ktoré využíva pseudo-jazyk (nejde o naozajstný programovací jazyk), ktorý umožňuje užívateľom zobraziť rôzne objednávky podľa ich poradia v sekvencii ako riadky textu.

Manuálne programovanie



Stroj, ktorý využíva manuálne programovanie

Pomocou **programovacieho panela** (teach pendant), ktorý pripomína bežný ovládač, môže užívateľ ovládať robota diaľkovo, aby robil špecifické pohyby, napríklad zdvihol súčiastku auta a umiestnil ju na pás. Pri manuálnom programovaní sa tvoria sekvencie príkazov a automaticky sa ukladajú, takže robot môže pracovať rovnakým spôsobom aj naďalej. Programátori môžu kód neskôr upraviť a urobiť zmeny týkajúce sa rýchlosti, presnosti a plynulosti pohybu. Keď je všetko hotové, pošlú robotovi hotový program naspäť.

Väčšinou je postup na vytvorenie továrenského robotického programu taký, že sa spíšu všetky príkazy v programovacom jazyku, skontrolovanie či fungujú za určitých podmienok, a rôzne úpravy kým sa nepríde k uspokojivému výsledku. Tento proces si vyžaduje komplexné zručnosti v kódovaní a často je limitovaný špecifickým rozsahom pohybu, napríklad keď robot presúva objekty z jedného miesta na druhé. Ale čo keď chceme napodobniť presné pohyby pracovníka, naprogramovať robota, aby robil ľudskú prácu bez písania náročných programových inštrukcií? Moderné robotické technológie nám toto presne umožňujú prostredníctvom manuálneho programovania.



Pracovník používajúci programovací panel



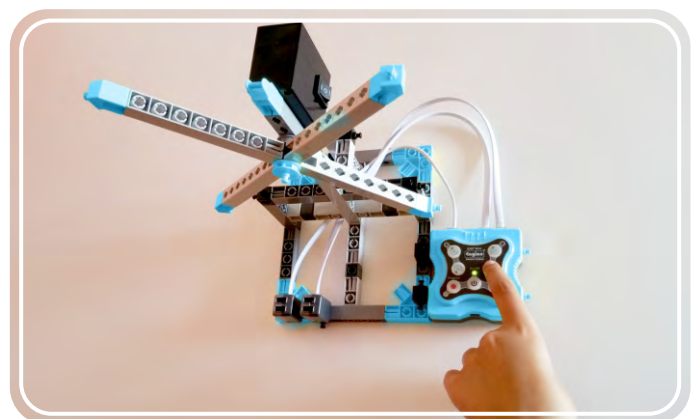
Vedeli ste, že ...

Prvým počítačovým programátorom bola anglická matematicka Augusta Ada King (grófká Lovelace). Ada napísala v roku 1847 prvý algoritmus v histórii, ktorý bol určený pre stroj. Jednalo sa konkrétne o algoritmus, ktorý by vypočítal Bernoulliho čísla pomocou analytického stroja - nápad ktorý Charles Babbage navrhol do mechanického počítača.



Obráz Augusta Ady King (1815 - 1852)

Manuálne programovanie je jednou z hlavných častí enginom KEIRO softvéru. Ide o užitočný spôsob ako vytvoriť program za bežných podmienok bez akejkoľvek znalosti samotného softvéru. Užívateľ jednoducho stláča tlačidlá na ovládači ERP MINI 2.0 a nahrá sekvenciu príkazov, ktoré bude robot v reálnom čase plniť. Potom čo je program uložený, môže byť kedykoľvek manuálne prehraný dokonca aj v slučke. Ak by užívateľ chcel navyše upraviť premenné príkazov a pridať podmienkové príkazy, môže to jednoducho urobiť po pripojení zariadenia k počítaču a zaslaním programu do KEIRO softvéru. Keď bude program hotový, stačí ho poslať robotovi na späť na testovanie.



Manuálne programovanie cez ovládač MINI 2.0

Užívateľský manuál

Filozofia Engino Robotics

Vývoj intelektu pomáha študentom rozvíjať ich nápady a zároveň získavať nové informácie. To je zásadné pri rozvoji a zlepšovaní kreativity a laboratórnych zručností. Kombinácia ovládača MINI 2.0 a softvéra KEIRO, ktorý čaká na patent a je podobný softvéru Scratch, je ideálnym prostriedkom k učeniu robotiky. Ovládač MINI 2.0 dáva **štyri prepojené možnosti programovania**, ktoré môžu byť skombinované tak, aby si každý užívateľ vybral metódu zodpovedajúcej jeho veku a skúsenostiam.



Manuálne programovanie

Prvou úrovňou programovania je fyzická interakcia s učebným materiálom. Stláčanie tlačidiel je metóda, ktorú poznajú všetky deti. Študenti môžu nahráť akúkoľvek sekvenciu príkazov cez tlačidlo "Program", uložiť ju na zariadenie a potom stlačiť tlačidlo "Play" pre zopakovanie akcie. Táto programovacia metóda je zásadná pri učení sekvencie udalostí a systému príkazov.

Simulátor

Ide o špeciálne okno v softvéri Keir, ktoré simuluje funkcie naozajstného ovládača s digitálnymi tlačidlami namiesto fyzických. Po pripojení ovládača MINI 2.0 k zariadeniu (počítaču alebo tabletu) môže užívateľ nahráť program a vidieť spätnú väzbu od robota. Pri nahrávaní programu sa generuje vývojový diagram a ukazuje sa v podobe vizuálnych blokov.



```
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
```

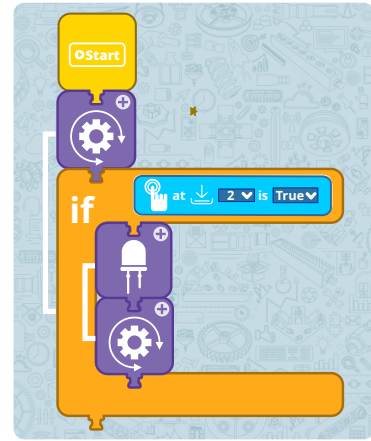
KEIRO Code

"Pseudojazyk" (nejde o naozajstný programovací jazyk) vytvorený priamo pre softvér Keir. Používa veľa známych pojmov z programovania ako napríklad BEGIN, IF, END pod. Je to ideálny nástroj na predstavenie pokročilejšieho programovania, pretože ukazuje rýchly náhľad programu v textovej podobe.

Vývojový diagram

"Vývojový diagram" je hlavným programovacím prvkom softvéru Keir. Tu môže užívateľ využiť všetky outputy a inputy robota na vytvorenie sofistikovanejších programov. Platforma je vytvorená v blokovom programovacom jazyku podobnom SCRATCH, ktorý umožňuje postupný prechod z manuálneho programovania k digitálnemu.

Programovacie prostredie v štýle drag-and-drop (pretiahni a pušť) boli preukázané ako cenné vzdelávacie nástroje, ktoré prinášajú jednoduchú cestu k interakcii s reálnym svetom a vývojom intuitívneho rozhrania ľudského stroja.

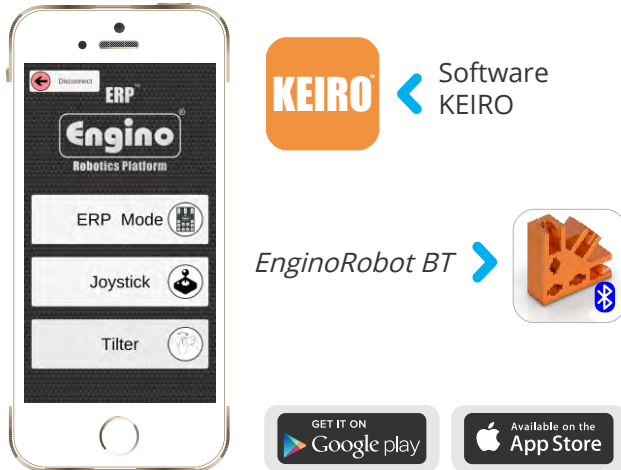


Vývojový diagram sa tvorí jednoducho pretáhaním a púšťaním akčných blokov

Pripojenie k inteligentným zariadeniam



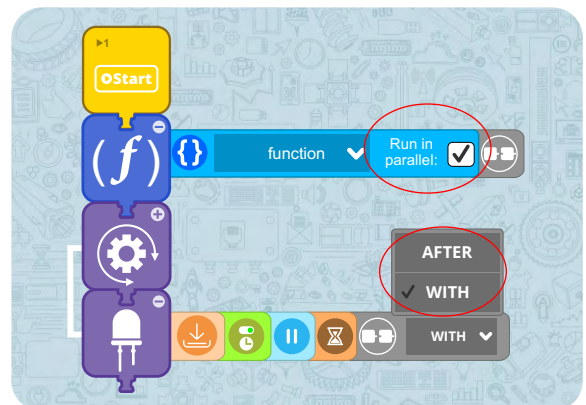
Ovládač MINI 2.0 ide s moderným technologickým pokrokom, a tak ho možno ovládať cez chytré zariadenie (mobil alebo tablet) pomocou pripojenia bluetooth. Keir softvér je grafická programovacia platforma podobná SCRATCH, ktorú je možné ovládať cez **počítač** alebo intel **inteligentných zariadení**. Programovanie robota tak môže byť zábava a využívajú sa všetky možnosti, aby boli študenti nadšení do práce na svojich projektoch.



EnginoRobot BT, ktorý je k dispozícii na Google Play alebo Apple Store, prakticky simuluje rozhranie ožajstného ovládača. Užívateľia môžu ovládať model, nahrávať a prehrávať programy rovnako ako stláčaním fyzických tlačidiel na ovládači MINI 2.0. A to všetko ide na diaľku a digitálne! Aplikácia má dve dodatočné metódy ovládania - **Joystick** a **Tilter**. Stiahnite si bezplatne aplikáciu a rozšírite svoj programovacie potenciál.

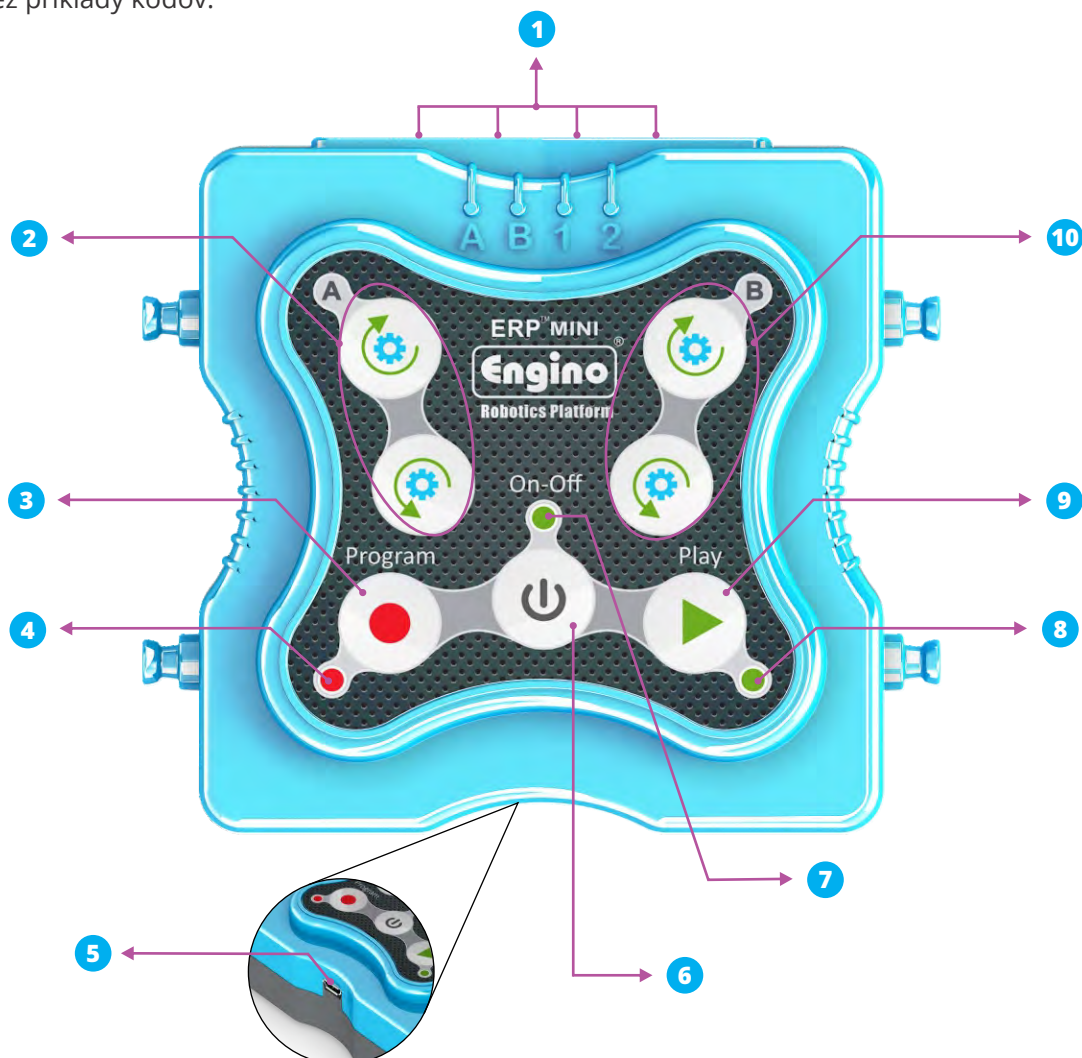
Paralelné programovanie

Veľkou novinkou vo vzdelávacej robotike od engine je funkcia paralelného programovania. Aktívne bloky v softvéri Keira majú šikovnú vlastnosť, vďaka ktorej si užívateľ môže vybrať, či bude príkaz splnený po alebo s predošlým príkazom. Softvér vie naviac nastaviť funkciu (x príkazov), aby fungovala **paralelne** (run in parallel) s nasledujúcimi akčnými blokmi kódu. Tieto možnosti znižujú zložitosť programovania a zjednodušujú sofistikované úlohy.



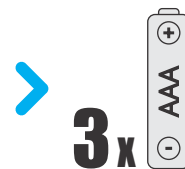
Ovládač MINI 2.0

Ovládač Engine MINI 2.0 je robotické zariadenie pozostávajúce z hlavného ovládača, niekoľkých tlačidiel a indikátorov. Cez 4 porty zariadenia možno ovládač pripojiť k perifériám ako sú motory, LED, infračervené a dotykové senzory. Ovládač MINI 2.0 je silný, flexibilný a jednoduchý nástroj na používanie. Tento manuál obsahuje informácie o jeho hlavných funkciách, návod ako nainštalovať softvér Keira a ako naprogramujete svoje roboty cez príklady kódov.



- | | | | |
|---|--|----|--|
| 1 | PORTY A, B, 1 a 2 RJ
<i>pripájajú k ovládaču periférie</i> | 6 | TLAČIDLO POWER |
| 2 | TLAČIDLÁ MOTOR A
<i>po alebo proti smeru hodinových ručičiek</i> | 7 | LED ON/OFF
<i>zapne sa, keď je zapnutý ovládač</i> |
| 3 | TLAČIDLO PROGRAM
<i>nahrá sekvenciu príkazov</i> | 8 | LED PLAY
<i>zapne sa pri prehrávaní programu</i> |
| 4 | LED PROGRAM
<i>bliká pri nahrávaní</i> | 9 | TLAČIDLO PLAY
<i>prehrá uložený program</i> |
| 5 | mini USB PORT
<i>pripája ovládač k počítaču</i> | 10 | TLAČIDLÁ MOTOR B
<i>po alebo proti smeru hodinových ručičiek</i> |

Batérie: Ovládač MINI 2.0 funguje s tromi (3) AAA batériami na zadnej strane. Na odstránenie krytu batérií budete potrebovať krížový skrutkovač. Umiestnite batérie so správnou polaritou, symboly + a - podľa ich určenia na každú pozíciu batérie. Po vložení batérií priskrutkujte naspäť kryt.



Bezpečnostné pokyny: Neprebíjajte batérie. Nemiešajte alkalické, štandardné a nabíjateľné batérie. Nenabíjateľné batérie nenabíjajte. Ak neplánujete vybité alebo staré batérie v blízkej dobe použiť, vyberte ich zo zariadenia. Nikdy batérie nespálujte. Napájacie terminály nesmú byť skratované.

Periférie, káble a porty



RJ káblový port

LED (červené)

K dodaniu vizuálnych efektov vášmu robotovi použijete **LED svetlo**.



RJ káblový port

Dotykový senzor

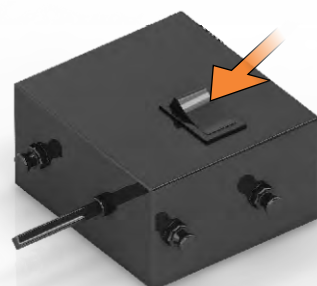
Stlačením tohto **tlačidla** možno spustiť **akciu** alebo pohyb.



RJ káblový port

Infračervený senzor

Infračervený senzor možno využiť na **rozpoznanie** predmetov alebo **nasledovanie čiary!**



vypínač motora

Jednosmerný motor

Jednosmerný motor s **vysokou rýchlosťou otáčania** umožňuje meniť rýchlosť otáčania **koliesok** alebo **rozpohybovať** vášho robota.



Senzorový kábel

Kábel, ktorý prepája senzory a LED svetlá s ovládačom MINI 2.0. Na oboch stranách je RJ konektor.



Motorový kábel

Kábel, ktorý spája jednosmerný motor a ovládač MINI 2.0., Kedy na jednom konci je RJ a na druhom je jack konektor.



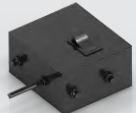



Kábel Mini USB

Kábel, ktorý spája ovládač MINI 2.0 a počítač.

V pozícii **Aj** funguje motor **v súlade** so smerom ovládača a príkazových blokov softvéru Keir. V pozícii **II** funguje v obrátenom smere. S vypínačom v pozícii **O** je motor vypnutý (OFF).



Periférie a porty: Ovládač Engin MINI 2.0 disponuje štyrmi (4) RJ portami na ovládanie motorov, LED a senzorov. Avšak rôzne typy periférií fungujú iba v niektorých portoch ovládača. Tabuľka nižšie ukazuje, ktoré porty sú použiteľné pre ktoré periférie *.

PERIFÉRIE	PORTY			
	A	B	1	2
	✓	✓		
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓

Tabuľka 1. Periférie a disponibilné porty

* Periférie (LED a dotykový senzor) nie sú v tomto balení zahrnuté, nájdete ich v balení engine Educational set STEM & Robotics MINI (www.engoeducation.com).

Príklad manuálneho programovania

Ovládač MINI 2.0 umožňuje manuálne ovládanie a nahrávanie pomocou tlačidiel na ovládači. Užívateľ môže postupne prejsť od fyzického manuálneho programovania na ovládanie cez software pomocou inovatívnej **metódy reverzného inžinierstva!** Nahraný program na zariadení možno jednoducho nahráť do počítača alebo inteligentného zariadenia.

*Akonáhle je motor pripojený k portu A
(Nastavte vypínač motora do pozície I):*

1. Stlačte tlačidlo "Program" k začatiu nahrávania (červený indikátor začne blikať).
2. Pridržte tlačidlo v smere hodinových ručičiek motora A na 3 sekundy.
3. Pridržte tlačidlo proti smeru hodinových ručičiek motora A na 2 sekundy.
4. Znovu stlačte tlačidlo "Program" na uloženie programu do pamäte.
5. Stlačte raz tlačidlo "Play" pre jedno prehranie programu (zabliká zelený indikátor)



Rady a tipy:

- Ak budete opakovane stláčať tlačidlo Play na 3 sekundy, bude sa program prehrávať v nekonečnej slučke.
- Zariadenie zvládne uložiť do svojej pamäti len posledný nahraný / zaslaný program.

Inštalácia softvéru KEIRO


Stroj, ktorý plní sériu príkazov nie je naozajstný robot - naozajstný robot je ten, ktorého možno naprogramovať k samostatnému rozhodovaniu na základe vnemov zo senzorov. Avšak manuálne programovanie na senzory nefunguje. K vytvoreniu pokročilého programu, v ktorom možno upravovať premenné a pridať technológiu senzorov, si potrebujete nainštalovať **softvér KEIRO**. Ten možno stiahnuť z webových stránok:


www.enginorobotics.com

1. Prejdite na stránky www.enginorobotics.com. Kliknite na "**download softvér**". Vyberte si balíček ERP MINI 2.0.
2. Vyberte vhodný operačný systém a kliknite na "**Download**".
3. Nainštalujte si program zo stiahnutého súboru.
4. Po dokončení inštalácie spustíte program.

Softvér KEIRO si môžete stiahnuť aj na svoje inteligentné zariadenie z Google Play alebo Apple Store.

ERP MINI 2.0 S BLUETOOTH

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

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

 [MacOS](#)

GET IT ON
 Google play

Available on the
 App Store

Aktualizácia firmvéru ovládača MINI 2.0

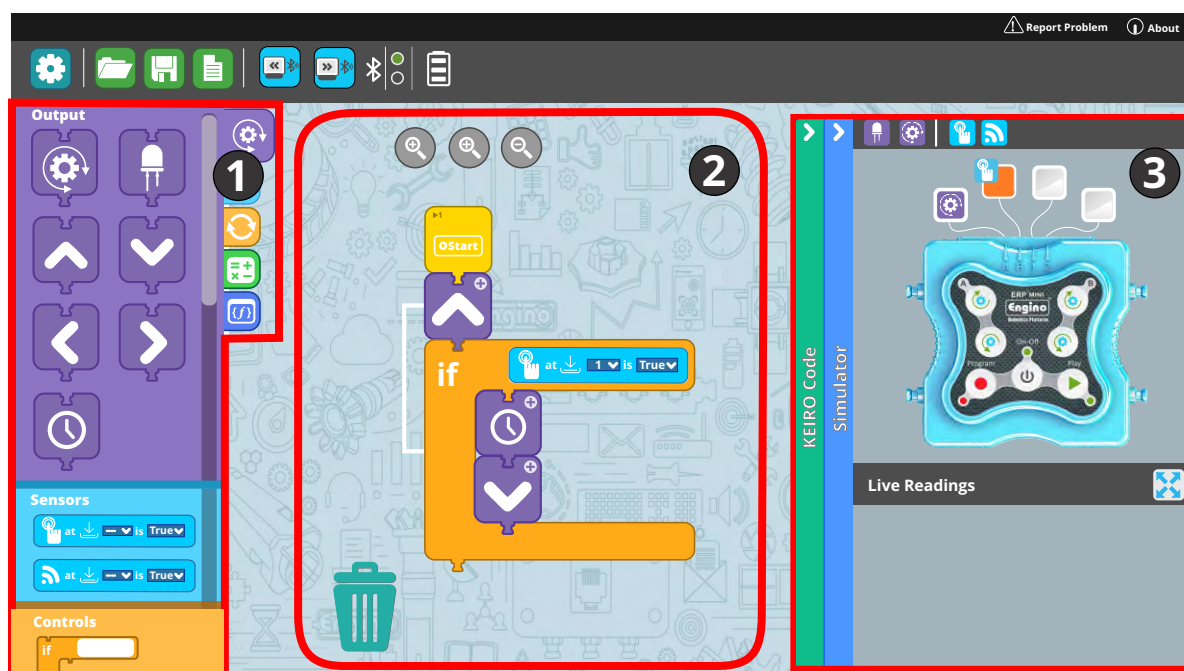
Softvér KEIRO vás bude informovať o nových aktualizáciách firmvéru ovládača MINI 2.0 vyskakovacím upozornením pri vašom pripojení.

Vyberte "YES" pre stiahnutie najnovších aktualizácií firmvéru.



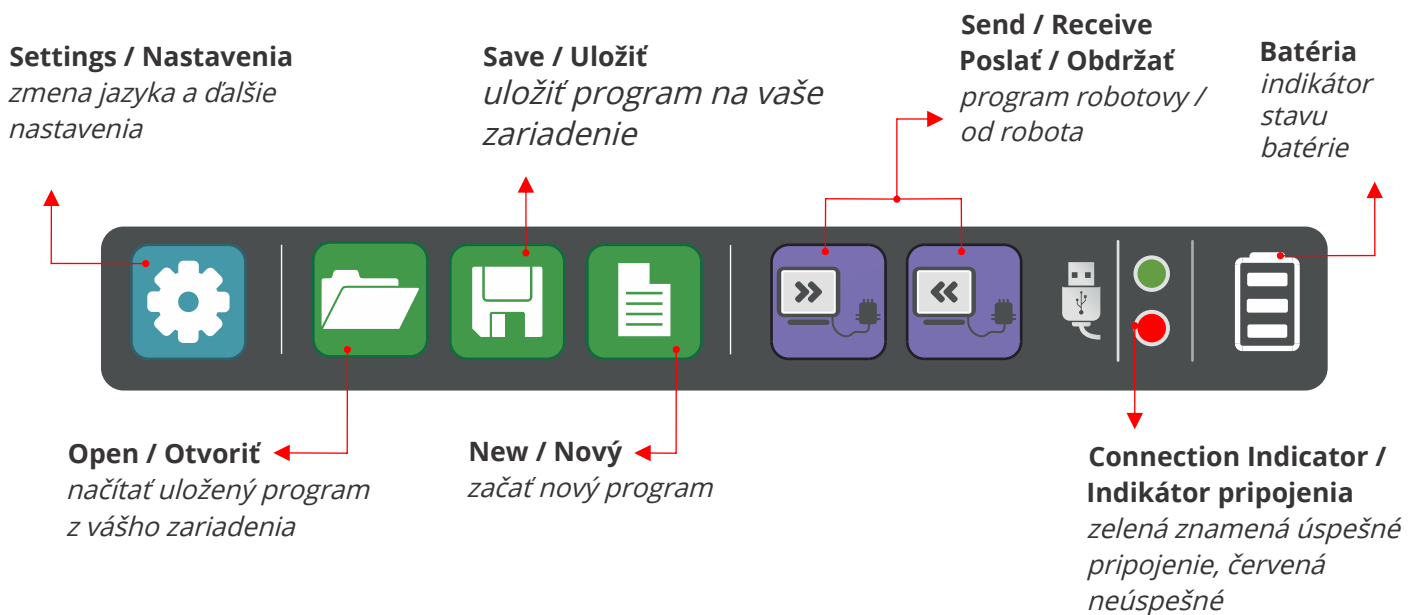
Rozhranie KEIRO

1. **Lišta s akčnými blokmi** - Tu nájdete všetky akčné bloky, ktoré možno využiť na vytvorenie sekvencie príkazov. Pre uľahčenie pri ich výbere sú bloky zoskupené do kategórií.
2. **Vývojový diagram** - Do tejto oblasti môžete preťahovať bloky a vytvárať tak program pre robota. Úpravy programu robíte cez úpravy vlastností jednotlivých blokov.
3. **Simulátor / KEIRO Code** - Aby softvér vedel, v ktorom porte sú pripojené ktoré periférie, je dôležitá konfigurácia simulátora. Code panel vám ukazuje váš program v textovej forme namiesto blokov. (Pre skrytie alebo zobrazenie okien kliknite na lištu).



Lišta menu

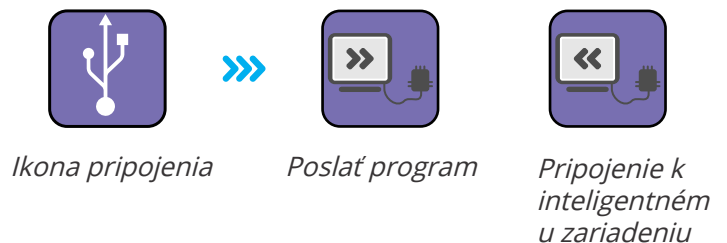
Lišta menu je umiestnená zvrchu softvéra Keir, v ktorej nájdete rôzne tlačidlá pokynov ako napríklad nadviazanie spojenia s ovládačom MINI 2.0, návrat na domovskú obrazovku, otvoriť / uložiť program pod.



Pripojenie k počítaču

Ovládač MINI 2.0 môžete **pripojiť k počítaču** mini USB káblom z balenia a potom kliknutím na správnu ikonu v menu. Po pripojení sa červený indikátor na lište menu prepne na zelený, čo znamená, že sa vytvorilo spojenie medzi počítačom a ovládačom.

Po úspešnom pripojení sa objavia dve nové tlačidlá na lište menu. Môžete **"Send / Poslať"** program z počítača do ovládača MINI 2.0 a naopak kliknutím na správne tlačidlo.

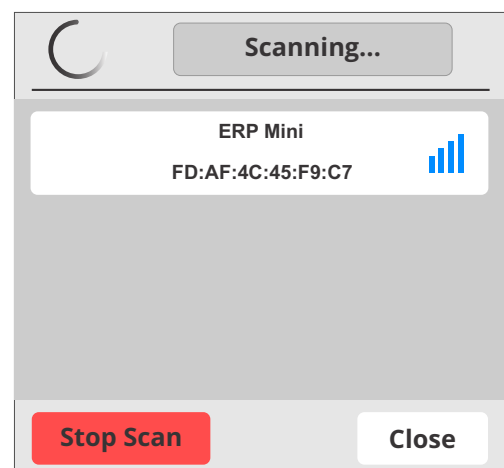


Pripojenie k inteligentnému zariadeniu

V prípade užívania softvéru **KEIRO** na **inteligentných zariadeniach** (platformy Android a iOS) sa objaví na lište menu ikona bluetooth namiesto USB.



Na pripojenie ovládača MINI 2.0 zapnite bluetooth na vašom zariadení a zapnite ovládač MINI 2.0. Kliknutím na ikonu bluetooth, naskenuje softvér dostupné zariadenia. Z nového okna, ktoré sa objaví, vyberte ovládač MINI 2.0 a pripojte ho k svojmu tabletu.



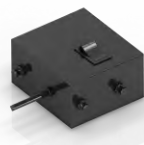
Naskenujte zariadenia bluetooth a pripojte ho k svojmu ERP Mini

Panel Simulátore

Simulátor je špeciálne okno v softvéri KEIRO, ktoré je určené na nastavovanie periférií robota. Pred nastavením vývojového diagramu je dôležité nastaviť Simulátor umiestnením periférií na rovnaké porty ako sú pripojené k robotovi. To dosiahnete pretiahnutím ikon periférií k správnym portom. Na hornej lište simulátora nájdete ikony všetkých periférií.



LED svetlo
Ikona bloku



Jednosmerný motor
Ikona bloku



Dotykový senzor
Ikona bloku



Infračervený senzor
Ikona bloku

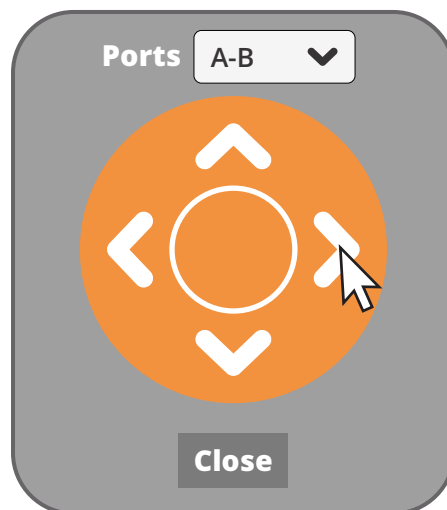
Panel GamePad



Panel GamePad sa objaví po kliknutí na špeciálnu ikonu v Simulátore.

Je vhodný na ovládanie dvoch motorových portov naraz. Nezabudnite, že motory môžu byť zapojené len do portov A-B, preto by toto mala byť zvolená kombinácia. GamePad predstavuje užitočnú možnosť, ako pohybovať robotickým vozidlom ľahko a jednoducho.

Jednoducho kliknite (alebo sa dotknite na inteligentnom zariadení) na šípky a motory budú reagovať na váš príkaz.



panel GamePad

Simulátor možno tiež využiť na diaľkové ovládanie modelu. Ikony, ktoré napodobňujú rozhranie ovládača sú aktívnymi tlačidlami, na ktoré môžete kliknúť. Tým pádom je možné nahráť či prehrať program priamo zo Simulátora. Všimnite si, že pri nahrávaní programu sa akčné bloky automaticky tvoria na vývojovom diagrame.

Live readings

V sekcii **Live Readings** dostanete po pripojení ovládača k softvéru okamžitú spätnú väzbu od periférií. Užívateľ môže sledovať stav každého pripojeného zariadenia napr. Rýchlosť a otáčanie motora, či je zapnuté LED alebo či je stlačený dotykový senzor. Infračervený senzor môžete navyše skalibrovať na rozpoznanie určitej vzdialenosti (viac o kalibrovanie IR senzorov, str. 25).

The screenshot shows the 'Live Readings' interface with four rows of sensor configurations. Each row has a download icon, a port selection dropdown, a status indicator, and a 'Calibrate' button. The Motor row shows Port 'A', direction 'ACL' (Anticlockwise), and speed '80'. The LED row shows Port '2' and status 'ON'. The Touch sensor row shows Port '1' and status 'TRUE'. The IR sensor row shows Port '1', status 'checked', sensitivity '65', and a 'Calibrate' button.

Motor: Ukazuje smer otáčania (V smere hodinových ručičiek /

- Clockwise- **CL**, Proti smeru hodinových ručičiek / Anticlockwise - **ACL**) a úroveň rýchlosti (**0-100**)

LED: Ukazuje, či je zapnuté alebo nie je. (**ON** ak svieti, inak **OFF**)

- **Dotykový senzor:** Ukazuje, či je stlačený alebo nie je. (**TRUE** ak je stlačený, inak **FALSE**)

Infračervený senzor: Ukazuje, či rozpoznáva nejaký objekt. (✓ ak áno, X ak nie). (Viac na str. 25)

KEIRO Code

Softvér KEIRO generuje na základe vývojového diagramu blokov pseudojazyk. Ide o zásadnú vzdelávaciu pomôcku, ktorá predstaví študentom ozajstné programovanie postupným prechodom od vizuálnych blokov k pochopiteľnému textovému formátu.

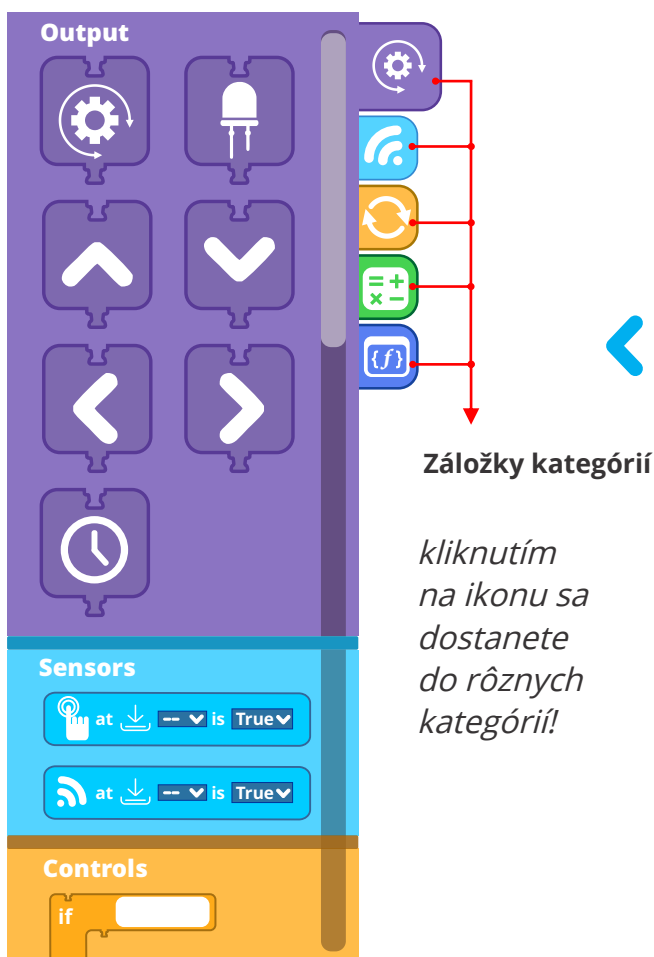
KEIRO Code obsahuje väčšinu pojmov bežne používaných v najrozšírenejších jazykoch, napr. BEGIN, IF, IF / ELSE, WHILE pod. Študenti si tak môžu priamo spojiť akčné bloky s textovými príkazmi a naopak!

Pre skrytie či zobrazenie panela kliknite na lištu

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

KEIRO Code

Lišta akčných blokov



Všetky bloky, ktoré môžete využívať na vytvorenie programu sú v tomto menu. Užívateľ si jednoducho vyberie akčný blok a ten pretiahne do vývojového diagramu - tak sa tvorí program.

Ikonové bloky sú roztriedené do 3 kategórií:

- 1. Output:**
 - Motor
 - Pohyb (vpred, vzad, vľavo, vpravo)
 - LED
 - Čakanie
- 2. Senzory**
 - Dotykový senzor
 - Infračervený (IR) senzor
- 3. Ovládanie:**
 - IF, IF/ELSE
 - WHILE
 - REPEAT TIMES, REPEAT SECONDS
 - REPEAT FOREVER,
 - REPEAT UNTIL, WAIT UNTIL
- 4. Operátory**
- 5. Funkcie**

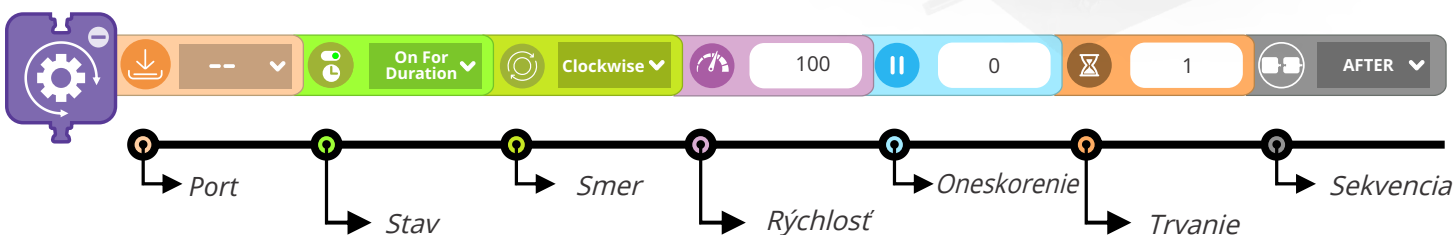
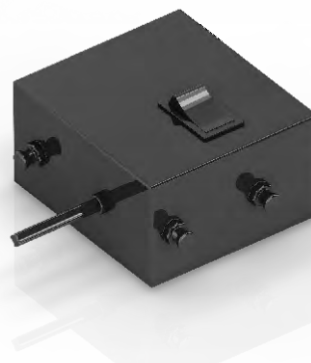
1. Outputy

Z tejto kategórie môžete do vývojového diagramu pretiahnuť ktorýkoľvek output a vytvoríť tak sekvenciu akcií. Pri umiestnení blokov do vývojového diagramu si všimnite, že sa otvorí **lišta vlastností**, v ktorej môžete blok upraviť. Napríklad keď potrebujete upraviť, do ktorého portu je output zapojený. Môžete si tiež upraviť akčný blok zmenou času trvania a rýchlosťou motora.



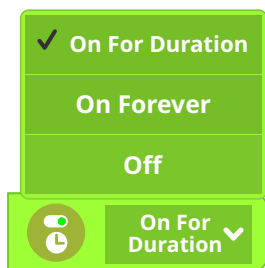
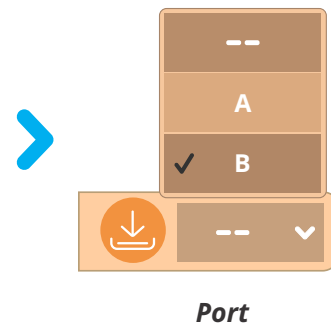
Blok motora

Nižšie vidíte rozpísaný blok motora s jeho východiskovými **vlastnosťami** a hodnotami



Vyberte **Port**, ktorým je motor k ovládaču pripojený. Táto vlastnosť je **najprv nevyplnená**, preto kliknite na pole a vyberte vhodný port. Dostupné porty sú zafarbené tmavšou farbou.

Poznámka: simulátor by mal byť nakonfigurovaný aspoň s jedným motorom, aby bolo možné vybrať port.



Stav

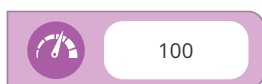
Nastavte Engin kidCAD (3D Viewer) aplikáciu motora. Máte na výber z troch možností: **On For Duration (predvolené)**: Motor sa bude otáčať určitú dobu, ktorá je nastavená vo vlastnosti Trvanie (popísaná nižšie). **On Forever**: Motor sa bude otáčať nekonečne, kým nedostane iný príkaz. **Off**: Zastaví rotáciu motora.

Vyberte Smer, ktorým by sa mal motor otáčať. Na výber máte medzi rotáciou **clockwise / v smere hodinových ručičiek (predvolené)** a **anticlockwise / proti smeru hodinových ručičiek**. Nezabudnite, že aby motor fungoval správne s vybraným smerom, mal by mať vypínač v **pozícii I**.

Poznámka: ak je stav nastavený na OFF, táto vlastnosť nie je k dispozícii.



Smer



Rýchlosť

Nastavte **Rýchlosť** rotácie motora. Napíšte do poľa hodnotu medzi **0** a **100**. Hodnota predstavuje percento prísunu energie do motora, preto **100 (predvolené)** predstavuje maximálny výkon.

Poznámka: ak je stav nastavený na OFF, táto vlastnosť nie je k dispozícii.

Nastavte čas **Oneskorenie** v sekundách k oneskoreniu spustenia bloku motora. Ak je vybratá hodnota 0 sekúnd (**predvolené**), nie je nastavené žiadne oneskorenie.



Oneskorenie



Trvanie

Nastavte čas **Trvania** v sekundách, aby sa motor otáčal určitú dobu. Po uplynutí času sa motor prestane otáčať.

Poznámka: Táto vlastnosť je spustená, len ak je stav nastavený na "On For Duration".

Sekvencie blokov sú v **predvolenom** nastavení vykonané **jedna za druhou** (AFTER). K súbežnému vykonávaniu úloh vyberte "**WITH**" (s predošlým), čím sa blok vykoná spoločne s predošlým. Táto vlastnosť umožňuje spustenie niekoľkých blokov naraz.



Navigačné (Pohybové) bloky

K rozhýbaniu vozidla by mali fungovať oba motory zároveň. Namiesto dvoch blokov vo vývojovom diagrame vám pomôžu **pohybové bloky**, ktoré ukazujú, či má vozidlo ísť **vpred**, **vzad**, **doľava alebo doprava**. Je to tak oveľa lepšie k navigácii, znižujú komplexitu a počet použitých blokov na vývojovom diagrame.



Vpred



Vzad

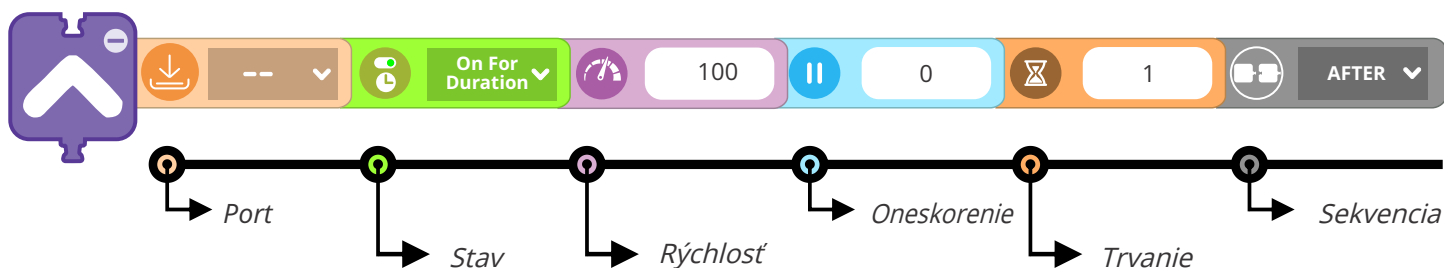


Vľavo



Vpravo

Nižšie vidíte príklad bloku pohybu vpred s jeho východiskovým nastavením **vlastností** a hodnôt. Tieto vlastnosti a funkcie sú rovnaké u všetkých ostatných navigačných blokov.



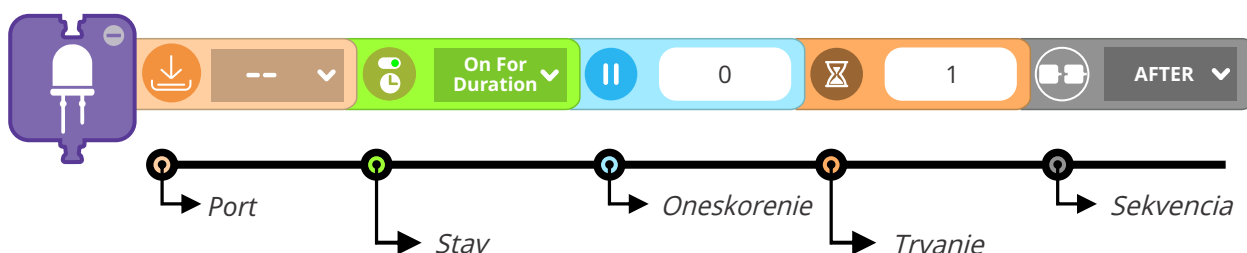
Navigačné bloky používajú rovnaké vlastnosti ako blok motora. Nezabudnite, že oba vypínače motorov musia byť nastavené do **pozície I**, aby navigačné bloky mohli viesť k pohybu robota.

Blok LED

Light **E**mitting **D**iode (LED) je zdroj svetla, ktorý využíva elektrinu prechádzajúcu kryštálom, čím generuje svetlo. LED svetlá patria medzi najefektívnejšie a najbezpečnejšie zdroje svetla s nízkou spotrebou, a preto sa často využívajú do elektronických zariadení a v robotike.

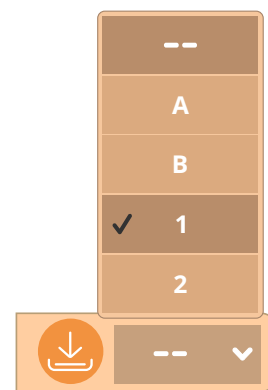


Nižšie vidíte rozpisovaný blok LED s jeho východiskovými **vlastnosťami** a hodnotami.

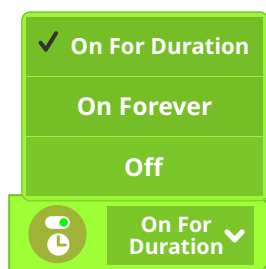


Vyberte **Port**, v ktorom je na ovládači MINI 2.0 motor pripojený. **Východiskovým** nastavením je miesto **prázdne**, preto vyberte z poľa vhodný port. Dostupné porty sú zafarbené tmavšie farbou.

Poznámka: aby port fungoval, mal by byť simulátor nakonfigurovaný s aspoň jedným LED.



Port



Stav

Nastavte **Stav** LED. Máte na výber z troch možností: **On For Duration (predvolené)**: LED bude svietiť určitú dobu, ktorá je nastavená vo vlastnosti Trvanie (popísaná nižšie). **On Forever**: Zapne LED, kým nedostane iný príkaz. **Off**: LED sa vypne.



Nastavte čas **Oneskorenie** v sekundách k oneskoreniu spustenia LED. Ak je vybratá hodnota 0 sekúnd (**predvolené**), nie je nastavené žiadne oneskorenie.



Oneskorenie

Nastavte čas **Trvanie** v sekundách, po akú dobu bude LED svetlo zapnuté. Po uplynutí času sa LED vypne.

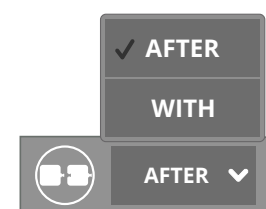


Trvanie

Poznámka: Táto vlastnosť je spustená len ak je stav nastavený na "On For Duration".



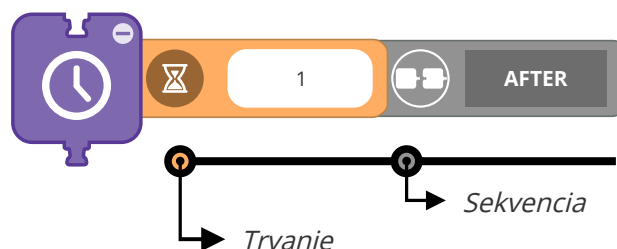
Sekvencie blokov sú v **predvolenom** nastavení vykonané jedna za druhou (**AFTER**). K súbežnému vykonávanie úloh vyberte "**WITH**" (s predošlým), čím sa blok vykoná spoločne s predošlým. Táto vlastnosť umožňuje spustenie niekoľkých blokov naraz.



Sekvencia

Blok doby nečinnosti

Blok doby nečinnosti zastaví program po určitú dobu. Táto doba nečinnosti je nastavená vo vlastnosti **trvania** a v predvolenom nastavení je nastavená na 1 sekundu.



Vzhľadom k funkcii doby nečinnosti nie je možné blok spustiť paralelne s predošlou či nasledujúcou úlohou.

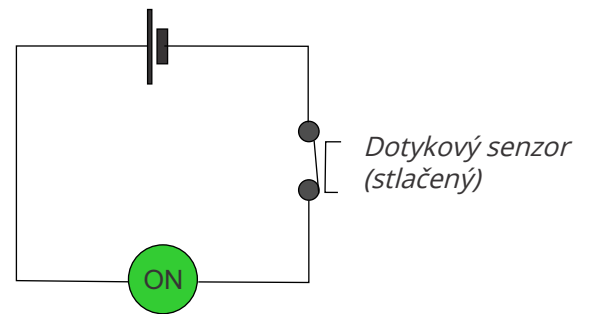
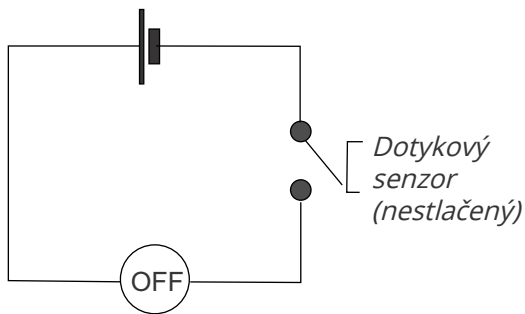
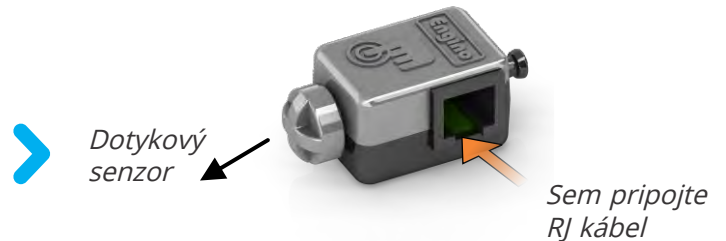
2. Bloky senzorov



Senzory sú input, ktoré ovládajú správanie robota. Vlastne sa jedná o zásadné prvky, vďaka ktorým môžeme nazývať stroj robotickým miesto automatickým. Robot zvažuje získané informácie zo svojho okolia, aby mohol splniť úlohy podľa situácie, ktorá nastala.

Dotykový senzor

Dotykový senzor pripojíte fyzicky k dotykovému senzoru Engino pomocou RJ kábla cez ktorýkoľvek port ovládača. Engino dotykový senzor je v podstate **dotykový vypínač**. Zo senzoru prichádzajú dva typy signálov (TRUE a FALSE), ktoré závisia od toho, či je stlačený. Nižšie nájdete schematický opis obvodov.



Pri **stlačení** vypínača sa obvod uzavrie, vytvorí sa tok prúdu a senzor je v stave **TRUE**:

Keď vypínač nie je stlačený, senzor je v stave **FALSE**, pretože obvodom neprechádza prúd.

Vyberte port, v ktorom je k ovládaču pripojený dotykový senzor.



Vyberte stav, ktorý by mal byť skontrolovaný, - TRUE alebo FALSE. Východiskovým stavom je TRUE.

Blok senzoru nemôže byť vo vývojovom diagrame ako nezávislý akčný blok. Blok senzora je vstup s **hodnotou boolean**. Ak stav senzora súhlasí s blokom, tak ide o input TRUE, keď nie je, ide o FALSE.

Blok senzoru tým pádom musíte použiť v spojení s príkazom ovládania, ako sú IF, WHILE pod., Aby robotovi povedali, či je podmienka splnená (true) alebo nie je (false). O hodnote boolean sensorového bloku sa dozviete viac v sekcii **Ovládacie bloky a Operátory**.

Infračervený (IR) senzor

Engin Infračervený senzor možno pripojiť k ovládaču RJ káblom. Engine Infračervený senzor je **aktívny IR senzor** skladajúci sa z dvoch prvkov: **infračerveného vysielateľa** (zdroj) a **infračerveného prijímača** (detektor). Infračervené senzory možno využiť na rozpoznanie predmetov a rozpoznať čierne od bielych povrchov.

IR vysielateľ vysielá infračervené signály do dosahu infračervenej oblasti (700nm až 1400nm) elektromagnetického poľa. Tieto vzdialenosti ľudské oko nemôže zachytiť.

Ak signál dôjde k predmetu, jeho časť sa odrazí naspäť. IR prijímač je fototranzistor schopný rozpoznať IR žiarenia. Senzor je tým pádom v stave **TRUE**, ak sa k nemu odráža veľa žiarenia od predmetu, a v stave **FALSE**, ak sa odráža žiarenie málo a alebo žiadne.

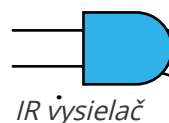


Prijímač infračerveného svetla

Vysielač infračerveného svetla

Sem zapojte RJ kábel

povrch



IR vysielateľ



IR prijímač

žiarenie odrazené od povrchu

predmet

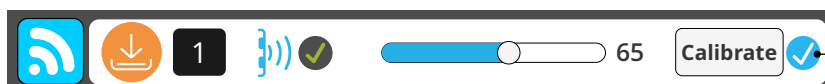


Kalibrácia infračerveného senzora

IR senzor je nutné skalibrovať na správny rozsah, ktorý spustí robotovu reakciu, aby robot "vedel", kedy sa zachovať podľa situácie TRUE či FALSE. Kalibráciu možno spustiť z panela **Live Readings**.

Stav **TRUE** sa zaktivuje pri bielej farbe a vzdialenosti rovnakej alebo menšej ako spúšťacia vzdialenosť.

Stav **FALSE** sa zaktivuje pri čiernej farbe a vzdialenosti **väčšej ako spúšťací vzdialenosť**.

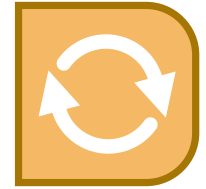


Správna kalibrácia

Rozpoznávanie predmetov: Umiestnite IR senzor na rovnakú vzdialenosť, pri ktorej bude senzor fungovať. Držte model na mieste a kliknite na **tlačidlo Calibrate**. Kalibrácia je hotová, akonáhle sa objaví vedľa tlačidla Calibre znak "✓". **Sledovanie čiary:** Umiestnite IR senzor nad **biely** povrch na rovnakú vzdialenosť, pri ktorej bude senzor fungovať. Nakalibrujte rovnakým spôsobom, ako je popísané vyššie.

* Kalibráciu možno vykonať tiež manuálne úpravou lišty na požadovanú spúšťacie úroveň.

3. Bloky riadenie

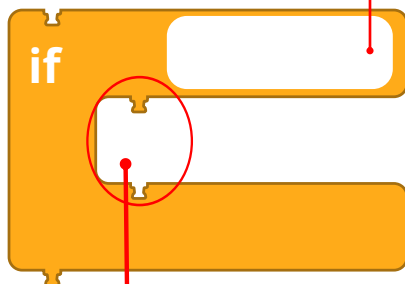


V tejto kategórii nájdete bloky, ktoré vytvárajú príkazy ovládanie. Bežné podmienkové príkazy ako **IF**, **WHILE**, **REPEAT** a **UNTIL** pomáhajú v sofistikovanejších a praktických využitíach. Input zo senzorov sa využívajú na implementáciu blokov v podmienkových príkazoch.

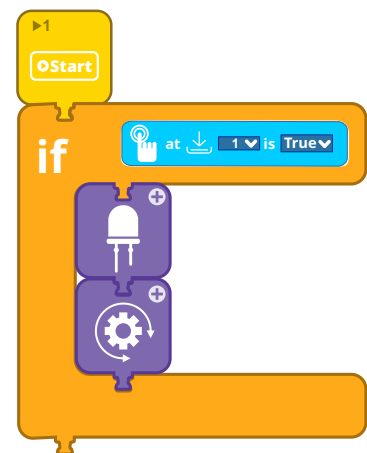
IF:

IF blok sa spustí, ak je splnená podmienka senzora (True alebo False). Napríklad, ak je stlačený dotykový senzor, zapne sa LED a motor sa začne otáčať. Ak je senzor aktivovaný druhýkrát, ešte pred vykonaním sekvencie blokov, čo sú vnútri, sekvencia nezačne od znova.

Sem presuňte ikonu bloku, ktorého podmienka by mala byť kontrolovaná

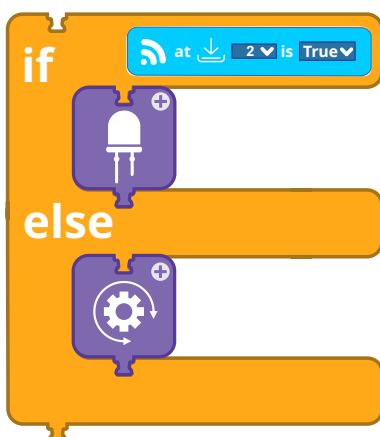


Pretiahnite akčné bloky na vytvorenie sekvencie, ktorá by mala byť vykonaná pri splnení podmienky



IF/ELSE :

IF / ELSE má dve časti. Sekvencia v prvej časti sa spustí v prípade, že je splnená podmienka senzora (napríklad True pri stlačení dotykového senzoru), rovnako ako v bloku IF. Ak je podmienka opačná (napríklad False, dotykový senzor nie je stlačený), robot vykoná sekvenciu blokov v druhej časti bloku.

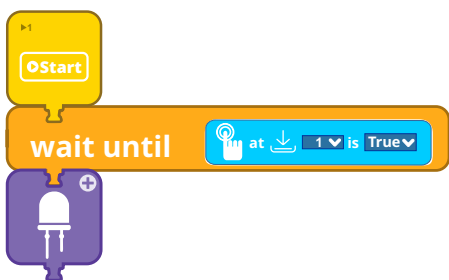
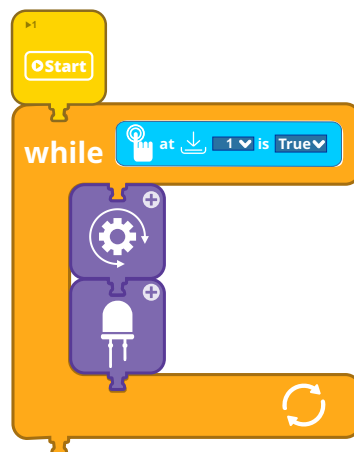


V tomto prípade sa zapne LED, v prípade, že IR senzor je TRUE (čiže rozpoznáva predmet). V opačnom prípade sa bude otáčať motor.

Bloky v ELSE budú prerušené, akonáhle sa podmienka splní, nech už je ich stav / trvanie akýkoľvek a program spustí sekvenciu v bloku IF.

While:

WHILE blok bude aktívny tak dlho, kým bude splnená podmienka senzora (True alebo False, čiže bude plniť čiastkový program v slučke. Akonáhle je podmienka prerušená, sekvencia v bloku WHILE sa okamžite zastaví a program bude pokračovať k ďalšiemu akčnému bloku. *Napríklad, kým bude stlačený dotykový senzor, motor sa bude otáčať v smere hodinových ručičiek a LED bude svietiť (kým senzor nepustíte).*

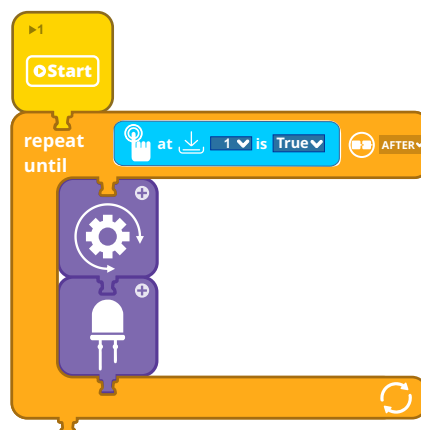


Wait Until:

Tento blok nastaví program do módu nečinnosti, kým nie je splnená určitá podmienka. Napríklad, program neprejde k rozsvieteniu LED kým nestlačíte dotykový senzor.

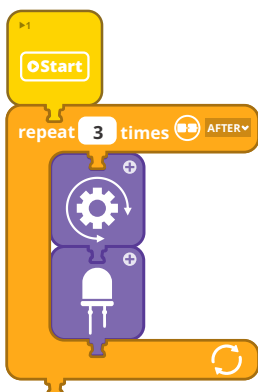
Repeat Until:

Blok Repeat until bude prehrávať v slučke sekvenciu blokov v ňom a zastaví sa len v prípade, že je splnená podmienka. Slučka sa zastaví, keď bude podmienka splnená a program bude pokračovať na ďalší akčný blok.

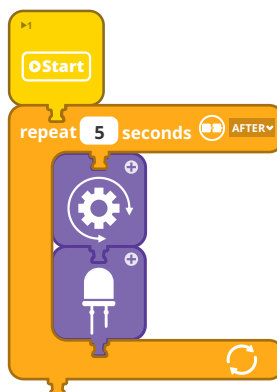


Repeat:

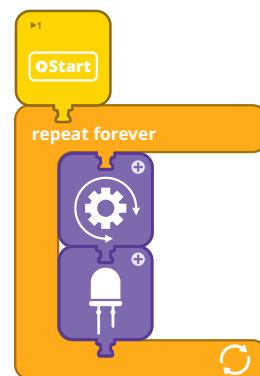
Opakovacích blokov, ktoré nezávisia na senzoroch alebo meniacich sa Input, ale na pevných scenároch, nájdete v softvéri tri typy. Blok "**repeat times**" vám dáva možnosť nastaviť počet požadovaných opakovaní, zatiaľ čo "**repeat seconds**" povedie k opakovaniu sekvencie po určitú dobu. Poslednou možnosťou je blok "**repeat forever**", ktorý je určený na prehrávanie sekvencie v nekonečnej slučke. Všimnite si, že tento blok nemá žiadny bod na pripojenie ďalšieho bloku, pretože už nie je možné žiadny pridať.



Slučka trikrát



Slučka 5 sekúnd



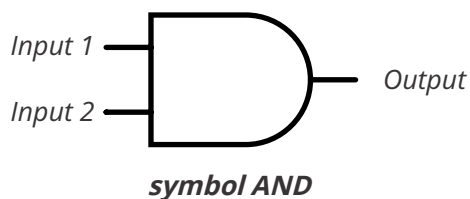
Nekonečná slučka

4. Bloky operátorov

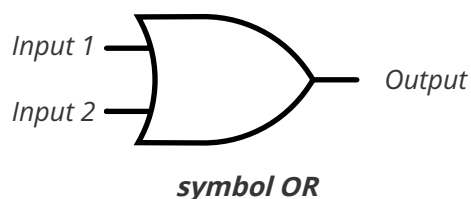


V tejto kategórii nájdete logické operátory, ktoré sú užitočné v situáciách, kedy by mala akcia prebehnúť pri kombinácii podmienok. Všeobecne, operátori hovoria počítaču ako zaobchádzať s informáciou.

Logické operátory sú pravdepodobne najdôležitejšími operátormi a využívajú sa v každom programovacom jazyku. Kombinujú dva inputy k vytvoreniu rozhodnutia. Predpokladajme, že chcete skombinovať dva senzory, ktoré povedú k príkazu robotovi. Logické operátory **AND** a **OR** by nám mali pomôcť.



Operátor AND: Keď sú **oba** inputy "true", tak je aj output "true". Inak je output "false".

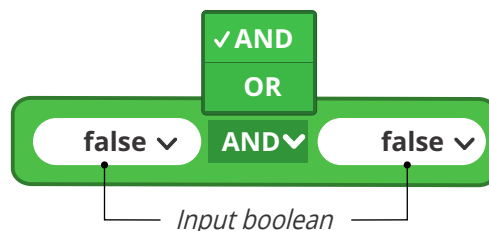


Operátor OR: Keď je jeden alebo oba inputy "true", tak je output "true". Output vyjde ako "false", iba ak sú oba inputy "false".

Blok logického operátora

Logický operátor softvéru KEIRO má dve okná inputov, do ktorých sa vkladajú hodnoty boolean, TRUE alebo FALSE (predvolené.). Kliknutím na pole uprostred môžete vybrať operátor AND / OR.

Spomeňte si, že blok senzoru je prakticky hodnotou boolean, takže pretiahnutím týchto slotov vznikne hodnota. Čiže, ak budú oba sloty spojené so senzorymi bloky, logický operátor vyhodnotí output na základe týchto dvoch inputov.



Predpokladajme, že máme robotické vozidlo s dvoma infračervenými senzormi na rozpoznávanie prekážok. Ak nerozpoznajú žiadne prekážky, chceme, aby sa robot pohyboval vpred. Takže **oba** senzory musia byť v stave False. Blok dopredu tak bude opakovať, pokiaľ budú obe dve podmienky splnené.

5. Funkcia

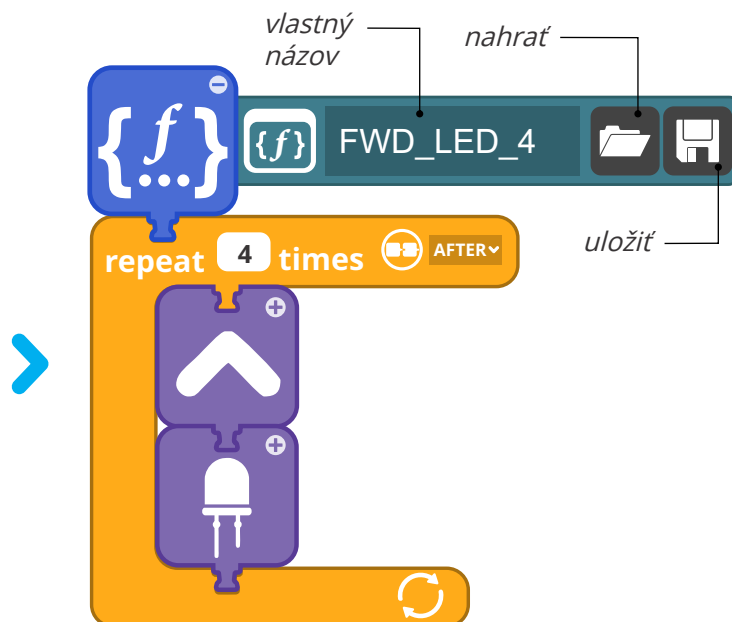
Funkcia je podprogramom, ktorý zahŕňa určitú sekvenciu príkazov. Ide jednoducho o krátky program, ktorý môže využiť či spustiť hlavný program buď v sekvenčnom alebo paralelnom móde. Funkcie sú užitočné pri menšom počte blokov vo vývojovom diagrame, hlavne v prípadoch, keď potrebujete vykonať opakovanú dlhú sekvenciu príkazov.



Nastavenie funkcie

K vytvoreniu novej funkcie pretiahnite blok "**set function**" (nastaviť funkciu) a pretiahnite ho do vývojového diagramu. Všimnite si, že blok nie je možné pripojiť k ostatným blokom, pretože je prvým blokom novej sekvencie príkazov.

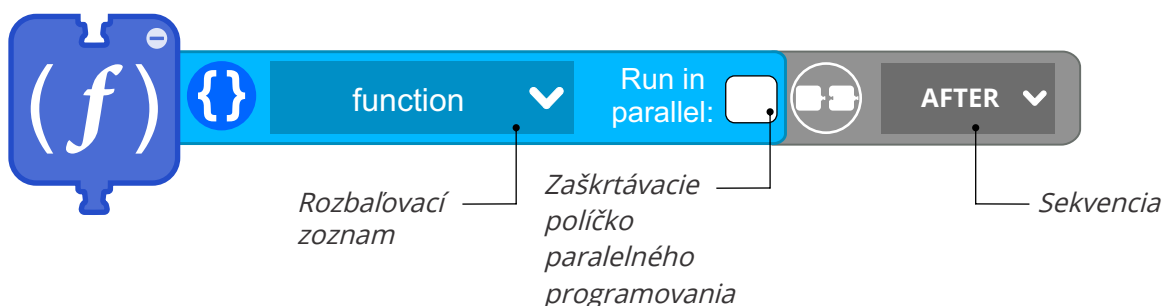
Môžete vytvoriť podprogram pod blokom "set function" (nastaviť funkciu) s akýmikoľvek akčnými blokmi (Output, senzory, ovládacie bloky, a pod.). Môžete si ich tiež **pomenovať pre** jednoduchšiu identifikáciu, hlavne v prípade, že budete nastavovať niekoľko funkcií. Funkciu si môžete uložiť do počítača / tabletu a kedykoľvek ju importovať pomocou ikon **uložiť** a **nahrať**.



Funkcia obsahuje podprogram s príkazmi

Použitie funkcie

K vyvolaniu funkcie musí užívateľ umiestniť blok "**Use a function**" (použiť funkciu) do sekvencie vývojového diagramu. V prípade, že máte nastavených niekoľko funkcií, vyberiete si funkciu z rozbaľovacieho zoznamu.



Sekvenčný režim

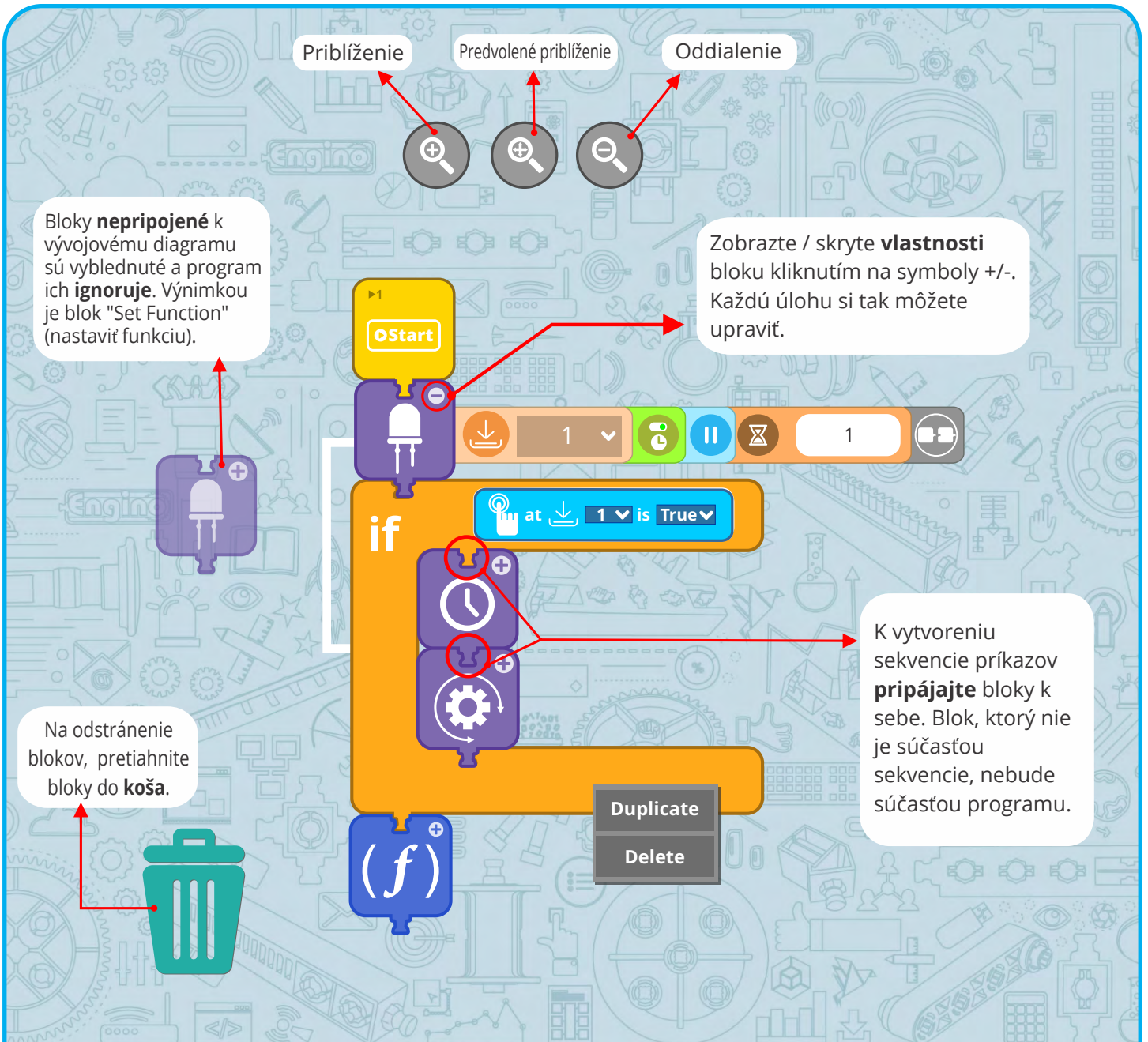
Východiskovým nastavením funkcie je **sekvenčný režim**. To znamená, že akonáhle program príde k bloku "use function" (použiť funkciu), spustí všetky úlohy vo funkcii. Akonáhle dokončí všetky úlohy vo funkcii, presunie sa k ďalšiemu bloku vývojového diagramu.

Režim "Run in parallel" (súbežne):

Paralelný režim spustíte zaškrtnutím zaškrtnutého políčka. V tomto režime program spustí úlohy vo funkcii a okamžite spustí ďalší akčný blok vo vývojovom diagrame. Týmto spôsobom sa podprogramové úlohy vo funkcii spustia paralelne so zvyškom vývojového diagramu.

Programovanie vývojového diagramu

Nižšie vidíte vývojový diagram s menovkami jeho hlavných prvkov. K vytvoreniu programu **pretahujete** bloky z Lišty blokov (ľavá strana obrazovky) pod ikonu "**Štart**" v hlavnom okne. Všimnite si, že akonáhle pretiahnete akčný blok blízko k voľnému bodu pripojenia, oblasť stmavne. Kliknite na symbol "+" na blokoch k upraveniu vlastností každého akčného bloku ako je napríklad port, trvania a pod. Po dokončení programu kliknite na lištu menu na tlačidlo "**Send program**" a program bude poslaný ovládačmi MINI.



Myšou môžete získať ďalšie informácie a robiť zmeny nasledujúcimi krokmi:

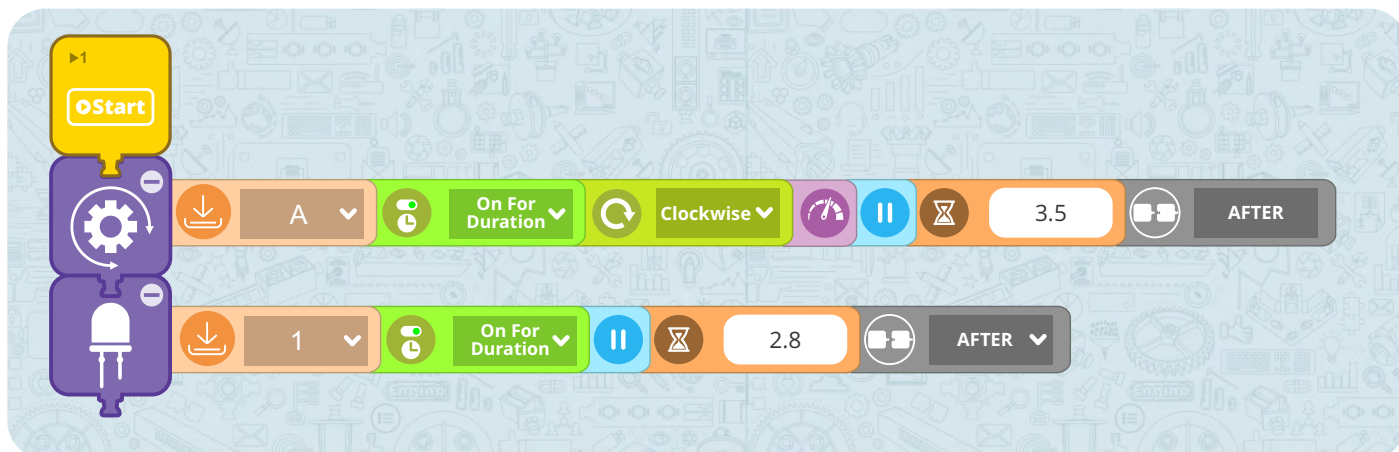
- **Presuňte** svoj kurzor myši nad ikony a bloky, čím zobrazíte popisy a typy.
- **Kliknutím** pri blokoch vývojového diagramu na symbol "+" zobrazíte vlastnosti blokov, ktoré môžete upraviť.
- **Kliknutím na symbol** "-" skryjete vlastnosti bloku.
- **Pravým kliknutím** na blok ho môžete duplikovať, alebo rýchlo zmazať jeden alebo viac blokov.

Ukážky kódov

Ukážka 1 - Jednoduché programovanie s motorom a LED

Táto ukážka využíva jeden **motor** a **LED**, ktoré sú v danom poradí pripojené k portom A a 1. V programe sa bude najprv otáčať motor v smere hodinových ručičiek, a to na 3,5 sekundy a potom sa zapne svetlo LED na 2,8 sekundy. Program bude vykonaný iba raz.

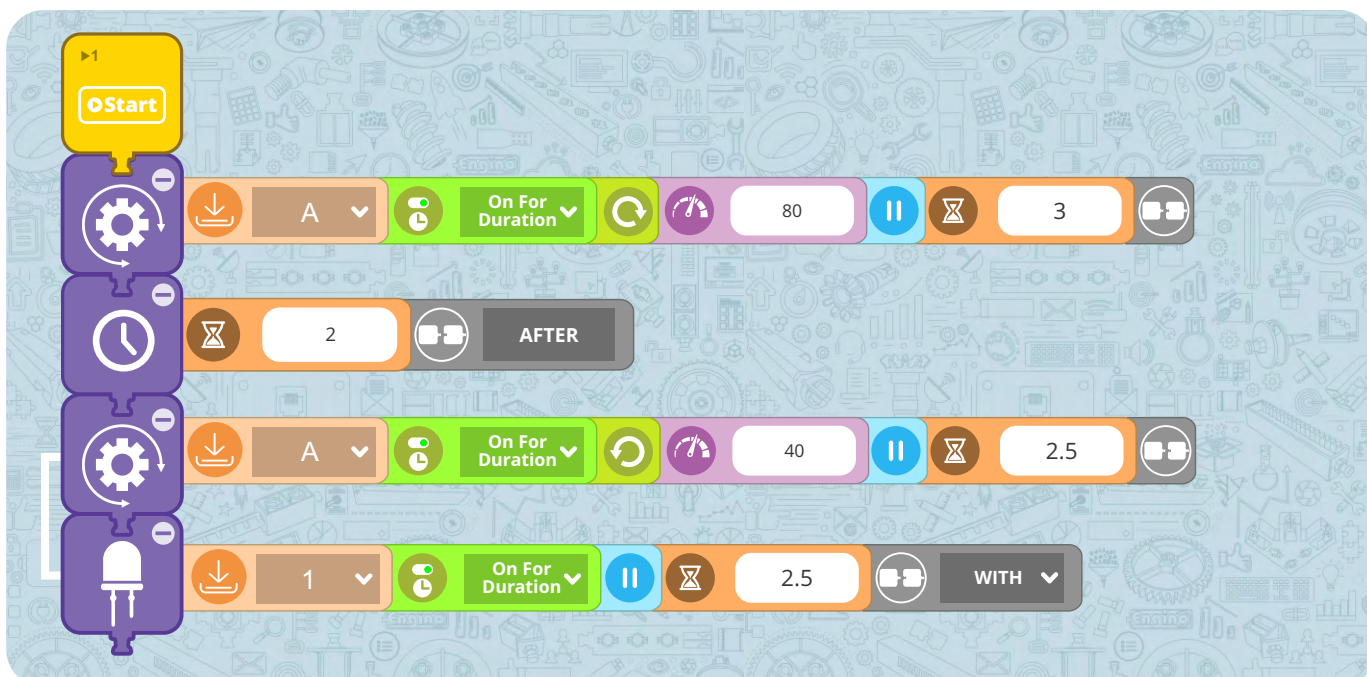
- Vyberte stav **On For Duration**
- Zmeňte čas trvania
- Interval trvania vyplňte reálnym číslom (float)



Ukážka 2 - Použitie súbežných úloh a Bloku doby nečinnosti

Táto ukážka využíva jeden **motor** a jedno **LED**, ktoré sú v danom poradí pripojené k portom A a 1. V programe sa bude najprv otáčať motor, a to na 3 sekundy na úrovni rýchlosti 80%. Program bude potom 2 sekundy nečinný. Potom sa motor začne otáčať rýchlosťou 40% a **zároveň** sa na 2,5 sekundy svetlo LED zapne.

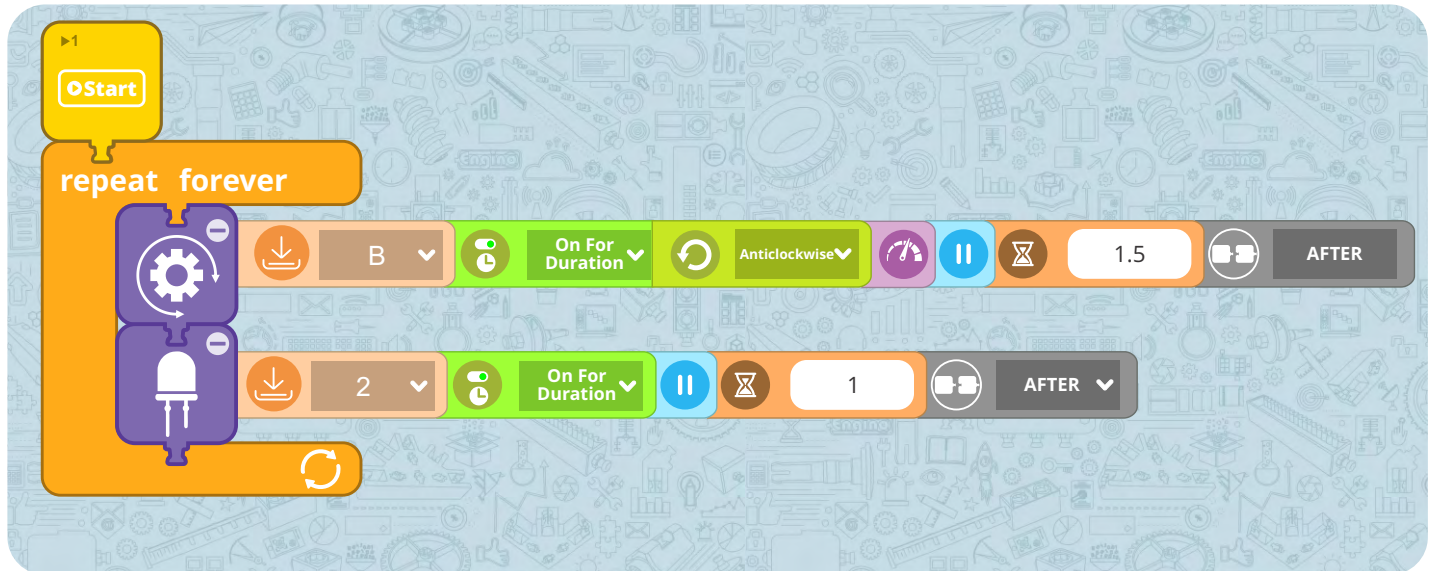
- Zmeňte rýchlosť motora
- Použitie bloku doby nečinnnosti
- Programovanie súbežných úloh



Ukážka 3 - Program v slučke

Tieto vzorky využíva jeden **motor** a **LED**, ktoré sú v danom poradí pripojené k portom B a 2. Úlohou je, aby sa motor otáčal proti smeru hodinových ručičiek 1,5 sekundy, potom by sa malo zapnúť svetlo LED na 1 sekundu a táto sekvencia by sa mala opakovať do nekonečna.

- Zmeňte smer otáčania
- Program v nekonečnej slučke



Ukážka 4 - podmienkový príkaz IF

Táto ukážka využíva jeden **motor**, jedno **LED** a **dotykový senzor**, ktoré sú v danom poradí pripojené k portom B, 1 a 2 .. Pokiaľ **nestlačíte** dotykový senzor (stav je **FALSE**) bude sa motor otáčať proti smeru hodinových ručičiek. Ak dotykový senzor stlačíte (stav je **TRUE**), zapne sa LED na 1,5 sekundy a motor sa bude otáčať proti smeru hodinových ručičiek 1 sekundu, ale s oneskorením 0,5 sekundy.

- Stav **On Forever**
- Dotykový senzor
- Podmienkový príkaz **IF**
- Čas oneskorenia



Konštrukcia a programovanie

Do robotiky zahŕňame dva hlavné aspekty: časť konštrukcie a časť programovanie. Na modeli ventilátora môžete experimentovať s obomi. Naučte sa, ako upraviť konštrukciu tak, aby bola efektívnejšia. Potom si manuálne naprogramujte ovládač a využite vizuálne bloky v softvéri KEIRO.

Zistite:

- Čo sú to vstupy a výstupy.
- Ako vylepšiť konštrukciu robota.
- Ako manuálne naprogramovať robota.
- Čo je to vývojový diagram.

Úroveň obtiažnosti ★★★★★

Potrebné materiály

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A).
- Počítač alebo inteligentné zariadenie so stiahnutým a nainštalovaným softvérom KEIRO.

Postup:

1. Na stránkach 43-45 si nájdite inštrukcie a postavte model ventilátora. Uistite sa, že ste pripojili všetky káble k správnym portom. Omotajte káble okolo ostatných súčiastok, aby sa vám neplietli okolo. Do zadnej časti ovládača vložte 3 AAA batérie a zapnite zariadenie pomocou tlačidla On-Off.



2. Vypínač oboch motorov by mal byť nastavený v pozícii I. V pozícii O je motor vypnutý a pozície II spúšťa opačný smer. Stlačte tlačidlá motora (príkaz) na ovládači, ako je ukázané v **cvičení 1** a zapíšte si výsledky, ktoré ste odpozorovali.

3. Vylepšite svoj model pomocou inštrukcií v **cvičení 2**.

4. V cvičení 3 si vytvoríte na ovládači manuálny program. Možno budete potrebovať pár pokusov, než sa vám to podarí.

5. Pripojte ovládač k počítaču alebo šikovnému zariadeniu. Spustíte softvér KEIRO a kliknite na tlačidlo **pripojiť**.



6. Kliknite na tlačidlo **"Receive Program"** (Obdržať program) k nahraniu nahrenej sekvencie do softvéru KEIRO, čím sa vám vytvorí "vývojový diagram".

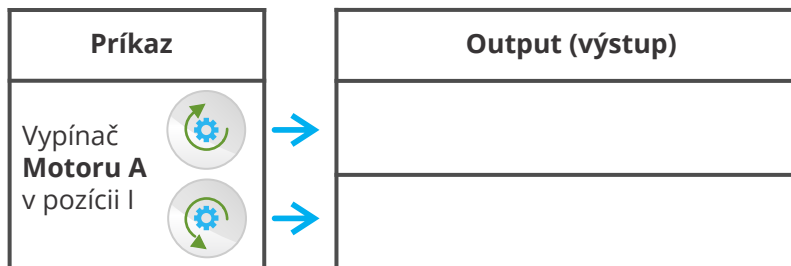


7. Urobte vo vývojovom diagrame zmeny opísané v **cvičení 4**. Pošlite potom program späť do ovládača kliknutím na tlačidlo "Send Program". Uložte si program ako **"ventilátor"**.



Engin model ventilátora

Cvičenie 1. Stlačte tlačidlá na ovládači (input) a spíšte stručne výstup, ktorý odsledujete na modeli (output).



Cvičenie 2.

Vytvorte na opačnej strane vášho ventilátora druhý ventilátor a pripojte ho do portu B. Napravo vidíte príklad konštrukcie.



Cvičenie 3.

Stlačte **"Program"** (červeno blikajúce svetlo ukazuje, že sa program nahráva) a potom potrebné tlačidlá k vytvoreniu nasledujúcej sekvencie: ventilátor A sa otáča v smere hodinových ručičiek; ventilátor B sa otáča v smere hodinových ručičiek; oba ventilátory sa točia zároveň v opačnom smere.



Znovu stlačte tlačidlo **"Program"** na uloženie svojho programu a potom stlačte tlačidlo **"Play"** pre skontrolovanie správnosti sekvencie. Ak tlačidlo Play stlačíte a podržíte, program sa spustí v slučke.



Cvičenie 4. Na vývojovom diagrame upravte vlastnosti všetkých akčných blokov na vytvorenie programu opísaného nižšie. Môžete pridávať ďalšie akcie (motory, doby nečinnosti) z ľavého menu: ventilátor A sa otáča v smere hodinových ručičiek 2 sekundy rýchlosťou 100; ventilátor B sa otáča v smere hodinových ručičiek 2 sekundy rýchlosťou 100; robot nerobí 1 sekundu nič (blok doby nečinnosti); oba ventilátory sa otáčajú zároveň rovnakým smerom po 3 sekundy, A rýchlosťou 100 a B rýchlosťou 40.



Naučte sa:

Robotiku

Pohyb robota a príkaz IF

Pohyb robota je veľmi dôležitý a občas je ťažké ho dosiahnuť. K rozhýbaniu robota existuje mnoho spôsobov, ktoré väčšinou záležia na ich využití: niektorí majú kolesá alebo dráhy, niektorí majú nohy, kým iní lietajú pomocou vrtule. Pokojne si zaexperimentujte s pohybom nôh pri modeli hexapodu!

Zistite:

- Ako funguje u robotov pohyb nôh.
- Ako sa používa Simulátor a Gamepad.
- Ako sa používajú infračervené senzory s podmienkovým príkazom IF.

Úroveň obtiažnosti ★★★★★

Potrebné materiály

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A).
- Počítač alebo inteligentné zariadenie so stiahnutým a nainštalovaným softvérom KEIRO.

Postup:

1. Na stránkach 46-51 si nájdite inštrukcie a postavte model **hexapodu**.

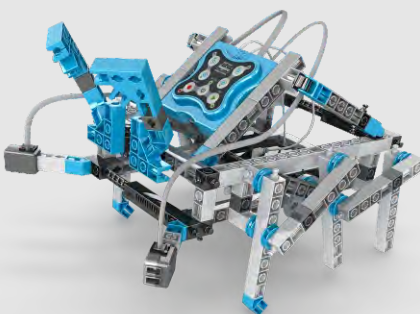
2. Pripojte ovládač k počítaču alebo inteligentnému zariadeniu. Spustíte softvér Keir a kliknite na tlačidlo **pripojiť**. Splňte **cvičenia 1 a 2**, v ktorých sa naučíte, ako pracovať so **Simulátorom** v softvéri KEIRO.

3. Otvorte si okno **Gamepad** kliknutím na správne tlačidlo a ovládajte ním svojho robota.



4. Všimnite si, že k pohybu dopredu alebo dozadu musia nohy začínať rovnakou pozíciou. Pri kolieskach by tento problém vôbec nenastal a pohyb dopredu je závislý len na rýchlosti otáčania motorov. Napadnú vás nejaké ďalšie problémy pri pohybe pomocou nôh (súvislosti)? Napíšte ich do **cvičenia 3**.

5. Splňte **cvičenie 4**, kde si naprogramujete hexapoda tak, aby sa vyhýbal prekážkam z oboch strán. K tomu budete potrebovať podmienkový príkaz IF.



Engin model hexapod

Cvičenie 1. Presuňte periférie (motory a senzory) na porty v Simulátore podľa modelu hexapodu. Bloky si zapíšte tiež sem do obrázka.



Cvičenie 2. Ovládajte hexapoda pomocou Simulátora podľa nasledujúcej sekvencie: hexapod zatočí doprava pomocou jednej nohy; hexapod zatočí doľava pomocou jednej nohy; hexapod zatočí doprava pomocou oboch nôh.

Cvičenie 3. Aké problémy môžu nastať, keď používate namiesto kolies nohy?

.....

.....

.....

.....

Cvičenie 4. Vytvorte uvedený program, vďaka ktorému sa hexapod bude vyhýbať prekážkam z oboch strán v nekonečnej slučke: hexapod sa opakovane pohybuje dopredu; **ak ľavý** IR senzor rozpozná prekážku, pravá noha by sa mala pohnúť dozadu a ľavá noha dopredu; **ak pravý** IR senzor rozpozná prekážku, ľavá noha by sa mala pohnúť dozadu a pravá noha dopredu.

Umiestnite do IF bloku správne bloky. Pre konfiguráciu IR senzorov na rozpoznávanie predmetov sa pozrite na **stránku 25**.

Príkaz WHILE a logické členy

Infračervené senzory sa v robotike využívajú veľmi často, a to pre dve využitia: rozpoznávanie medzi čiernou a bielou farbou a rozpoznávanie predmetov. Vyskúšajte si obe programovacie výzvy a naučte sa, ako nakonfigurovať IR senzory, ako sa používa podmienkový príkaz WHILE a logické členy.

Zistite:

- Ako fungujú infračervené senzory s podmienkovými príkazmi WHILE a IF.
- Čo sú to logické členy.

Úroveň obtiažnosti ★★★★★

Potrebné materiály

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A)
- Počítač alebo inteligentné zariadenie so stiahnutým a nainštalovaným softvérom Keir.
- 4x biele kartóny formátu A3
- Čiernu izolepu alebo čiernu fixu
- 4 biele predmety pre detekciu

Postup:

1. Na stránkach 38-42 si nájdite inštrukcie a postavte model robota. Uistite sa, že ste pripojili každé kolesko k portu na rovnakej strane.

2. Zlepte papiere A3 k sebe, aby ste vytvorili jeden veľký obdĺžnik. Nakreslite čiernu veľkú elipsu fixkou alebo izolepou (obrázok vpravo). Iný tvar si môžete stiahnuť na stránkach www.enginorobotics.com/teaching_resource/s/examples/

3. Otvorte si softvér KEIRO a nasledujte inštrukcie v **cvičení 1** k vytvoreniu programu **sledovania bielej. IR senzory by mali pozeráť do zeme**, takže ich nastavte smerom nadol. Robot sa bude pohybovať na bielej a zastaví kolesá na čiernej, čiže bude nasledovať čiaru.



4. Prečítajte si **cvičenie 2** k premene na robota, ktorý **rozpoznáva predmety**. Stačí upraviť **IR senzory** tak, aby **pozerali dopredu**. Po vytvorení programu ho môžete otestovať umiestnením štyroch bielych objektov (vysokých rovnako ako senzory) na vytvorenie imaginárneho štvorca, po ktorého stranách sa bude robot pohybovať (obrázok vpravo.)

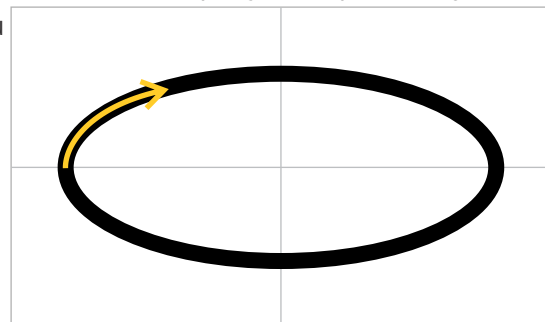


Engin model robota

Cvičenie 1. Na počítači alebo inteligentnom zariadení umiestnite v Simulátore správne ikony podľa modelu robota. Pretiahnite potrebné akčné bloky do vývojového diagramu na vytvorenie nasledujúceho programu pohybu po bielej a zafixovania cesty po čiernej: **zatiaľ čo** (while) IR senzor v porte 1 rozpoznáva bielu farbu, motor v porte A by sa mal otáčať dopredu; **zatiaľ čo** (while) IR senzor v porte 2 rozpoznáva bielu farbu, motor v porte B by sa mal otáčať dopredu.

Po umiestnení správnych blokov bude potrebné **skalibrovať infračervené senzory** pre nasledovanie čiaru. Viac nájdete na strane 25 v sekcii "Kalibrácia infračerveného senzora" užívateľského manuálu.

S týmto programom sa robot bude pohybovať po čiernej čiare. K otestovaniu vášho programu, robota umiestnite na čiernu elipsu, ktorú ste si vytvorili, a sledujte, či bude čiaru nasledovať.

**Cvičenie 2.**

Presuňte do vývojového diagramu potrebné bloky na vytvorenie nasledujúceho programu: robot sa pohybuje vpred; **ak** (if) rozpozná biely predmet **jedným alebo oboma senzormi**, robot sa zastaví; po 5 sekundách by sa humanoid mal otočiť o 90 stupňov doprava, aby mohol rozpoznať ďalší predmet.

Tipy: uhol je vždy rovnaký, takže k otáčaniu je nutný iba jeden program. Na nastavenie obidvoch alebo aspoň jedného senzora k rozpoznávaniu predmetov musíte použiť správny logický člen (AND a OR). Mali by ste tiež znovu **kalibrovať IR senzory**, aby boli aktivované k **rozpoznávaniu predmetov**.



Príklad z praxe: priemyselné rameno

Roboti sa často využívajú v priemysle, kde hrajú hlavnú úlohu výrobných liniek. Ich najčastejším využitím je podoba stacionárnych ramien, ktoré presúvajú časti z jedného miesta na druhé pomocou technológie senzorov. S twister modelom si môžete vyskúšať simulovanú verziu týchto situácií.

Zistite:

- Ako fungujú stacionárne roboty.
- Ako funguje podmienkový príkaz REPEAT TIMES.
- Ako funguje podmienkový príkaz REPEAT UNTIL.

Úroveň obtiažnosti ★★★★★

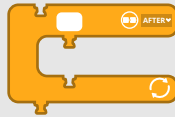
Potrebné materiály

- Kreativne inžinierstvo 100 v 1 Robotized Maker Pro (CE101MP-A).
- Počítač alebo inteligentné zariadenie so stiahnutým a nainštalovaným softvérom KEIRO.
- Izolepu a 1 ľahkú plechovku

Postup:

- 1. Na stránkach 52-57** si nájdite inštrukcie a postavte model **Twistera**.
- Stacionárne znamená, že stojí na pevnom mieste. V **cvičení 1** upevníte robota, aby sa nehýbal pri otáčaní.
- V programovaní sú veľmi dôležité programy s opakujúcimi sa presnými úlohami. V **cvičení 2** vytvoríte program v slučke, a to pomocou bloku "**repeat times**".

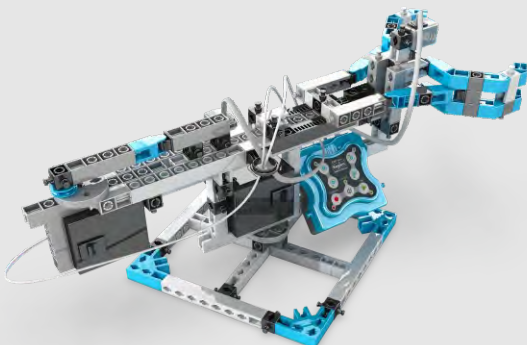
- Niekedy chceme, aby sa slučka opakovala, kým nie je splnená podmienka senzora. V **cvičení 3** sa naučíte, ako to doceliť pomocou bloku "**repeat until**".



- Stacionárne roboty často fungujú na základe presného návodu na dokončenie rutinnej práce.



K tomu sa využíva technológia IR senzorov, aby robot vedel, ako sa pohybovať. Pokiaľ využijete len časovanie, po mnohých opakovaní začne byť robotov pohyb chybný. V **cvičení 4** si vytvorte simuláciu rutinné robotické práce.



Engin model twister

Cvičenie 1. Vytvorte si manuálne jednoduchý program otáčania ramenom. Umiestnite robota na zem a sledujte, čo sa bude diať pri spustení programu. Uvidíte, že sa otáča aj základňa. Tento problém upravíte umiestnením žltej kladky pod všetky 4 rohy základne. Pre väčšiu stabilitu môžete umiestniť kladky na izolepu. Vyskúšajte program znova.



Cvičenie 2. V počítači alebo inteligentnom zariadení umiestnite správne ikony na Simulátor podľa modelu twister. Pretiahnite potrebné bloky do vývojového diagramu na vytvorenie nasledujúceho programu: twister sa bude otáčať 1 sekundu rýchlosťou 50, potom zavrie a otvorí svoje rameno; táto zostava by sa mala zopakovať 5 krát.

Tip: Použite blok "repeat times".

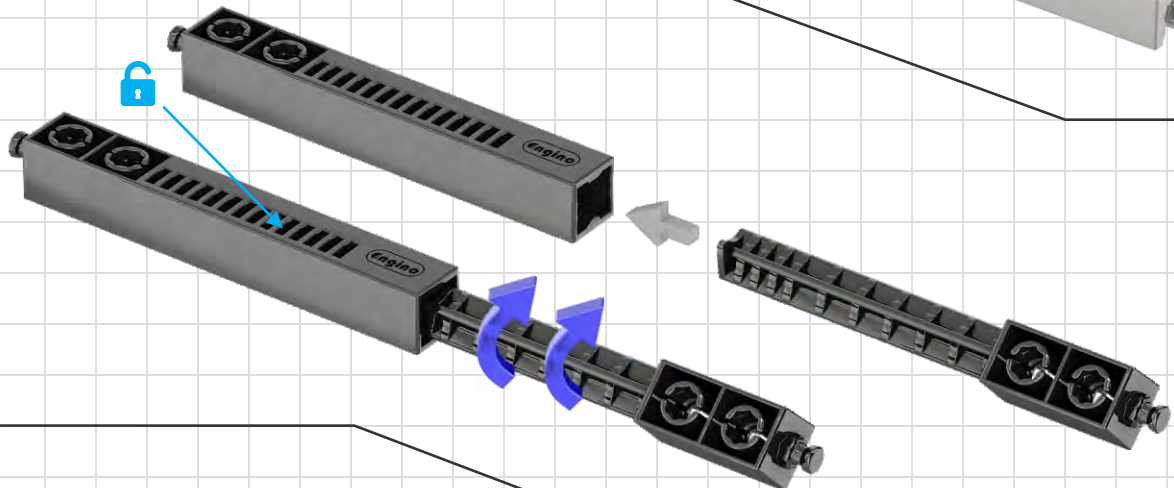
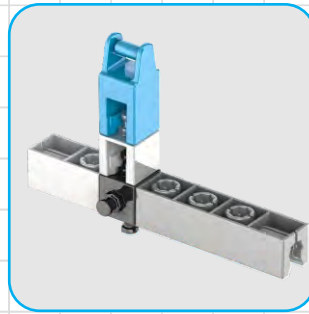
Cvičenie 3. Vytvorte nasledujúci program (zovretie ramena by malo byť na začiatku otvorené): twister sa bude otáčať, **kým** (until) bočný IR senzor nerozpozna predmet alebo ruku; ak vo chvíli nečinnosti modelu umiestnime pred IR senzor ramena plechovku, zovretie by sa malo uzavrieť a chytiť ju.

TIP: Použite blok "repeat until" pre prvú časť programu a upravte rameno pre lepšie zovretie, ak to bude potrebné.

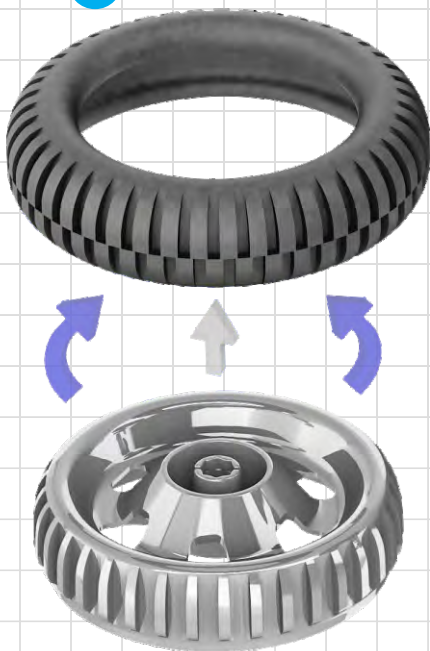
Cvičenie 4. Umiestnite do ramena modelu plechovku a vytvorte nasledujúci program: *twister sa otočí o 180 stupňov doľava s plechovkou v zovretí; akonáhle model zastaví, rameno by sa malo povoliť a upustiť plechovku; twister by sa potom mal otočiť o 180 doprava (čiže do svojej pôvodnej pozície); akonáhle umiestnite plechovku pred senzor ramena, model by ju mal zovrieť a postup opakovať (otočenie o 180 stupňov, upustenie plechovky a otočenie naspäť).*

Tip: umiestnite dva predmety (alebo svoju ruku) na jednu zo strán prostredného senzora k jeho spusteniu.

Tipy k stavaniu



tire
L



Dvojitý
rozpojovací nástroj



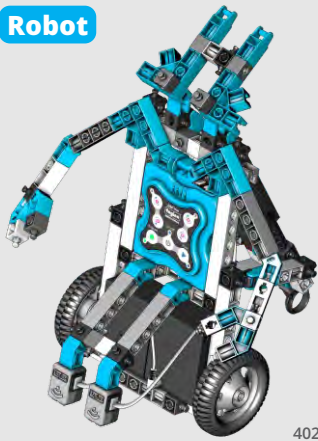
2. MOŽNOSŤ:
Vložte nástroj do otvoru
a vytlačte tak druhú
súčiastku!

1. MOŽNOSŤ:
Stlačte nástroj, čím
rozpojíte malé
súčiastky!



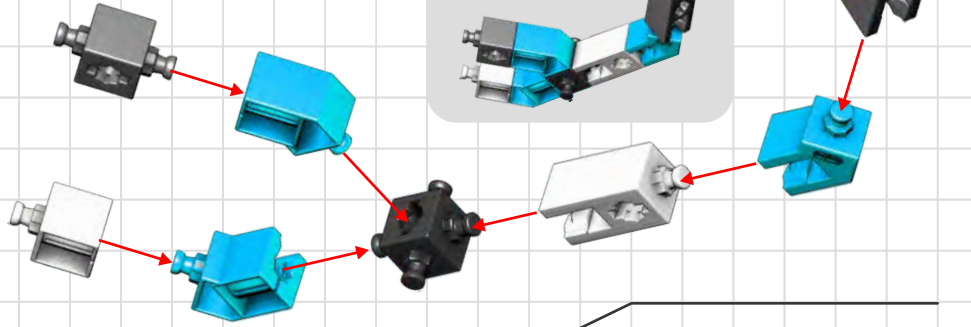
Návod na stavanie

Robot

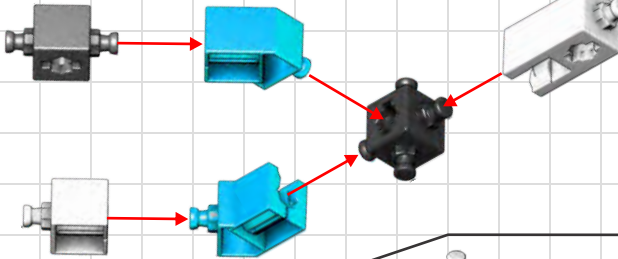


4023

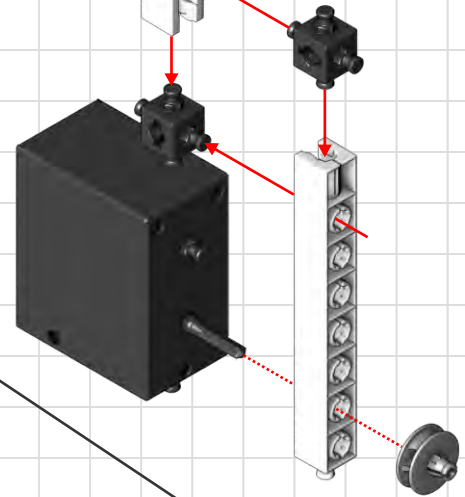
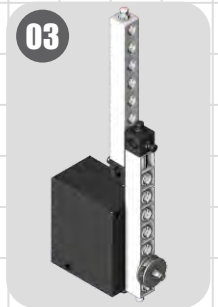
01



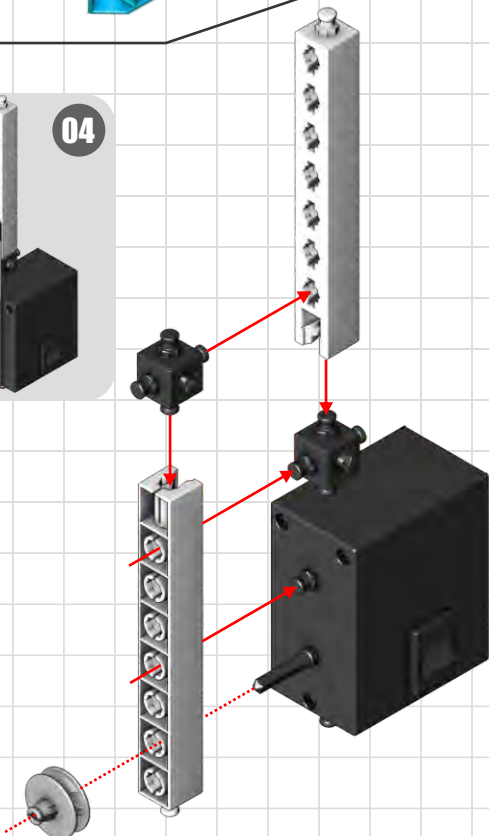
02



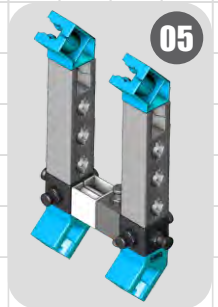
03

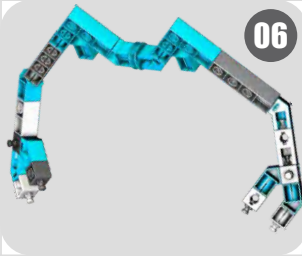


04

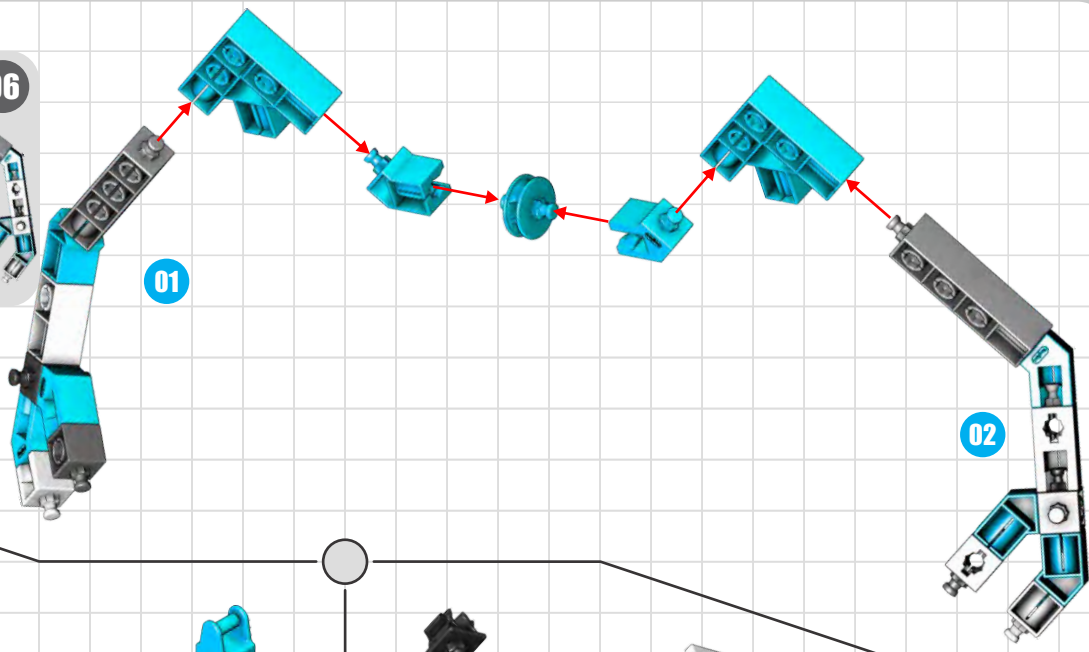


05

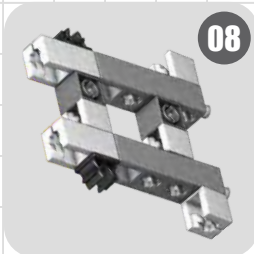
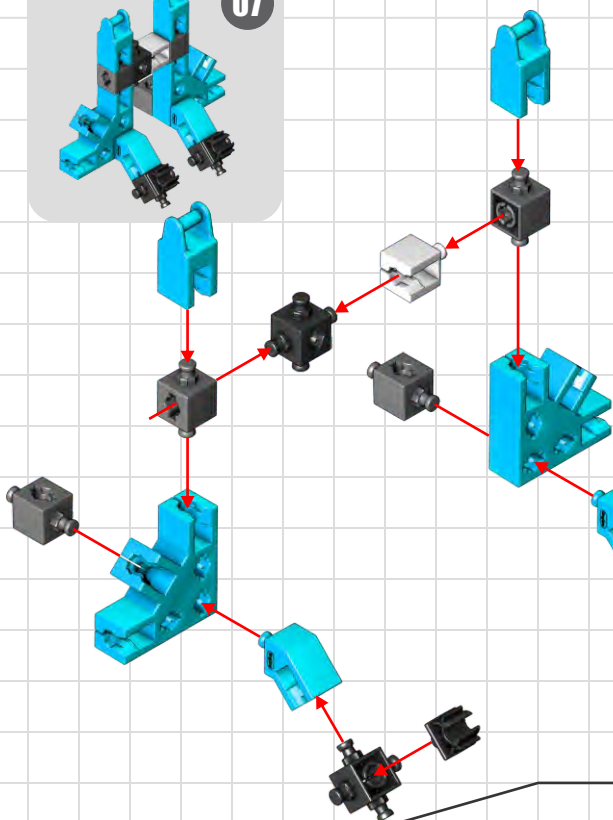




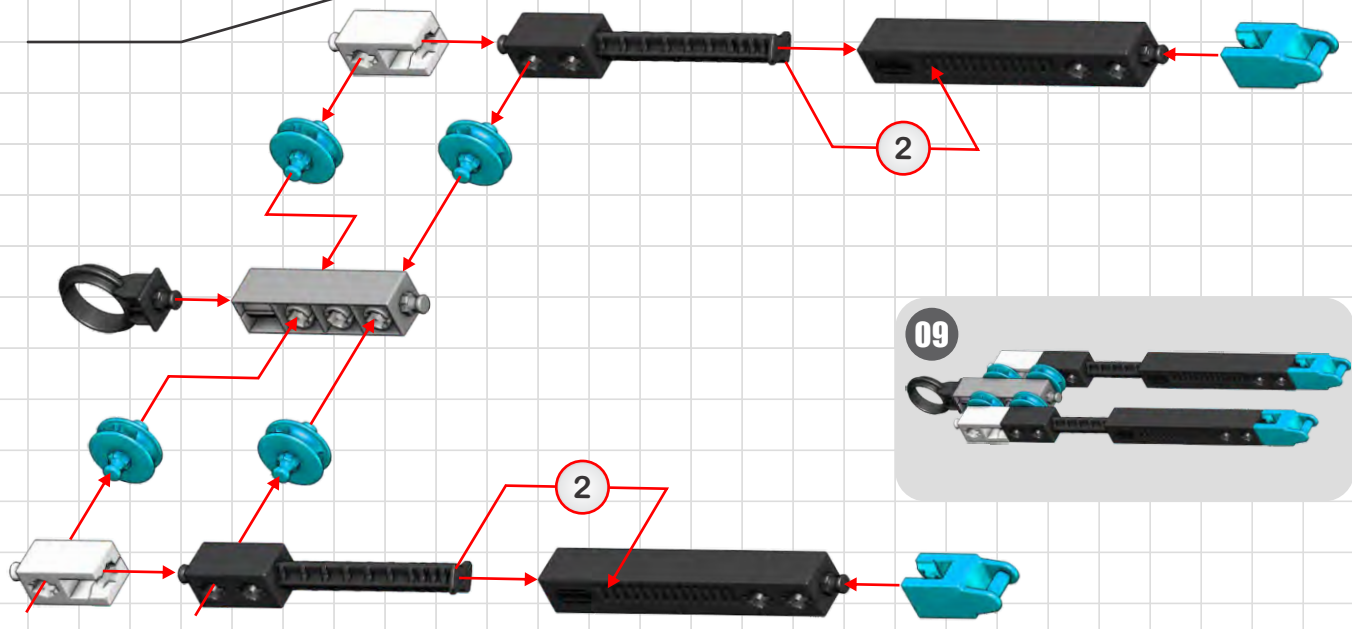
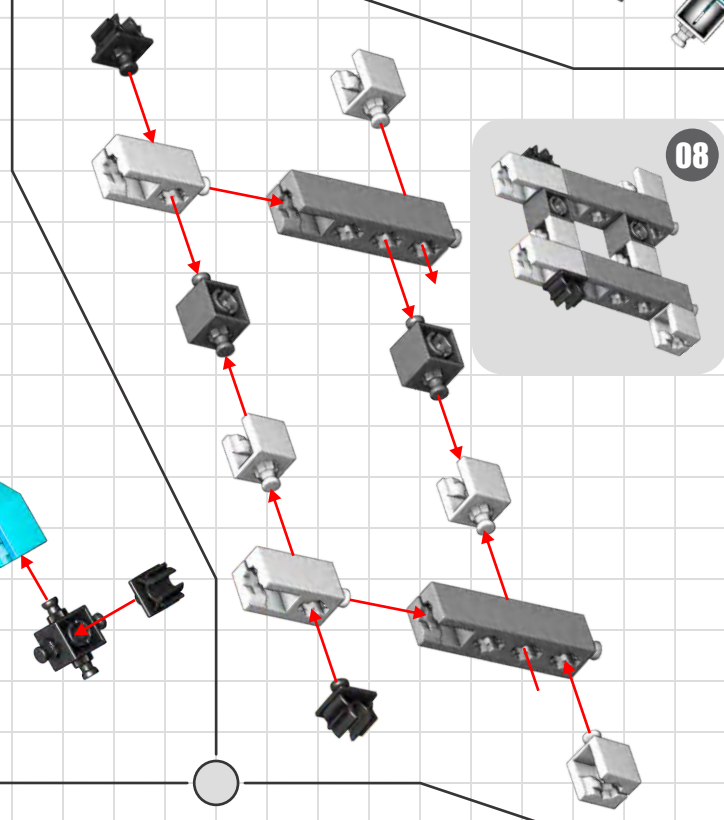
06



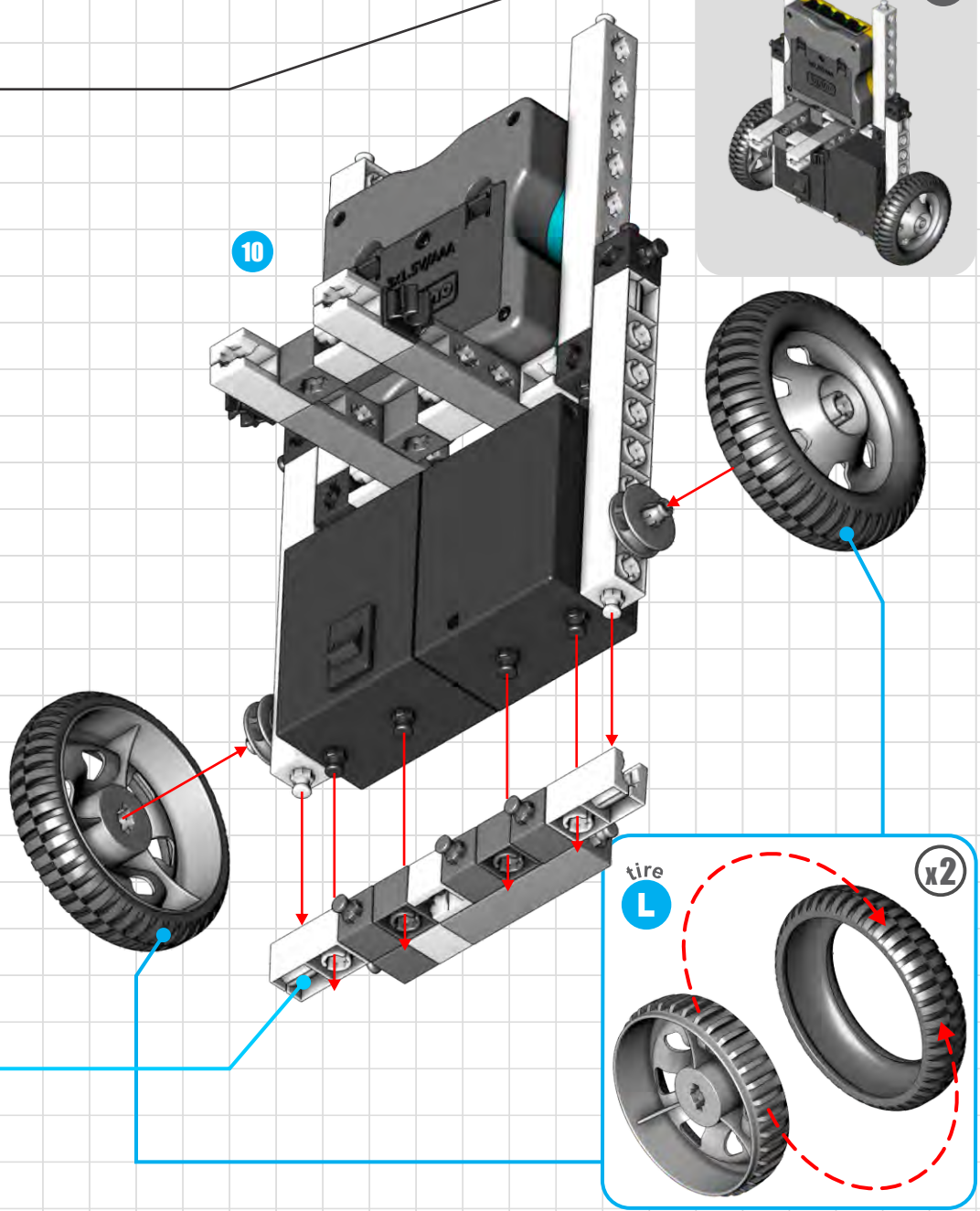
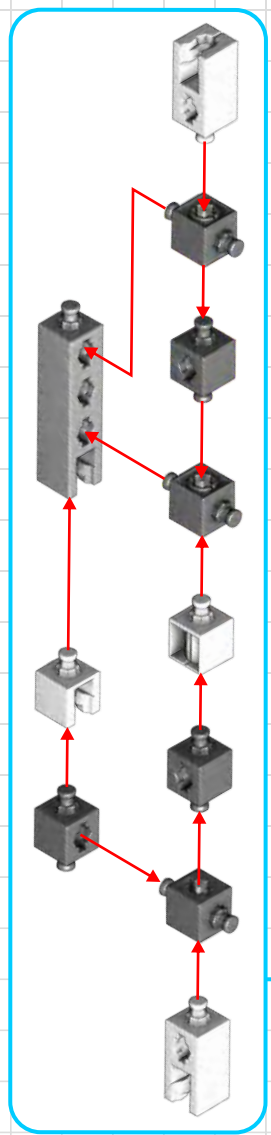
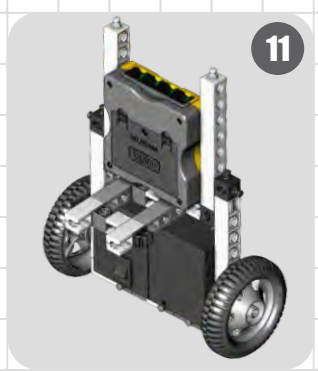
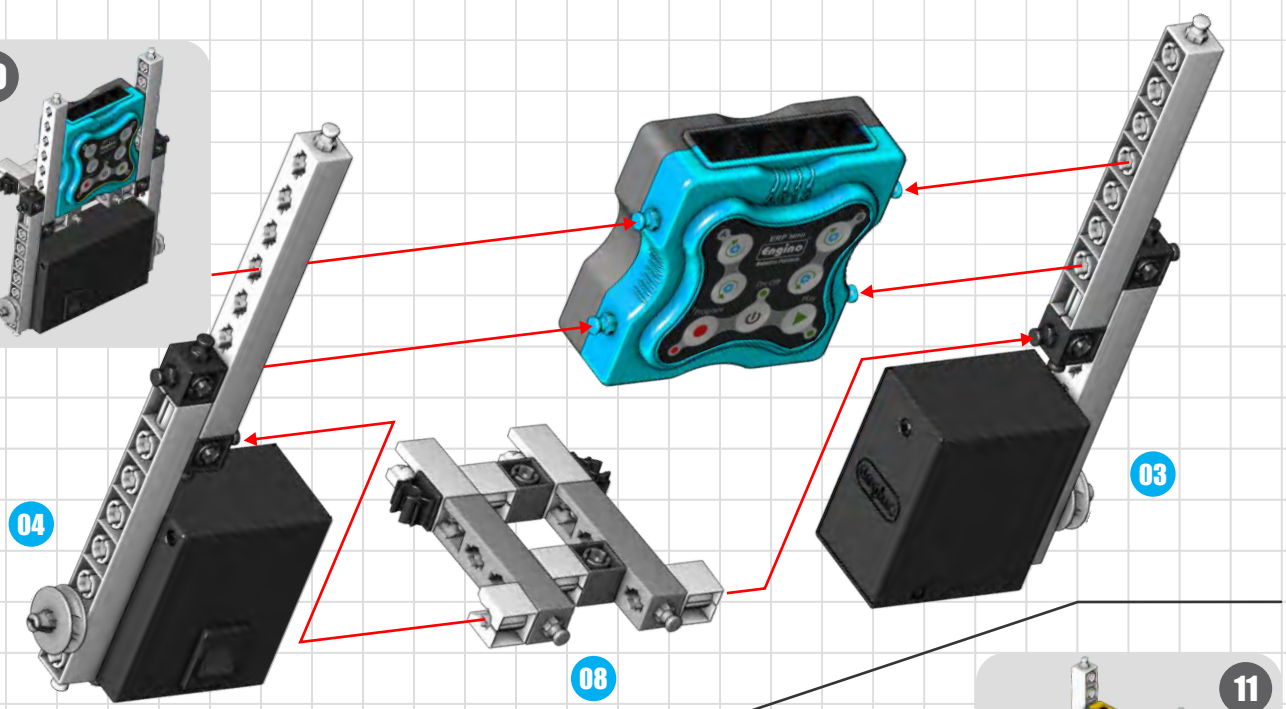
07

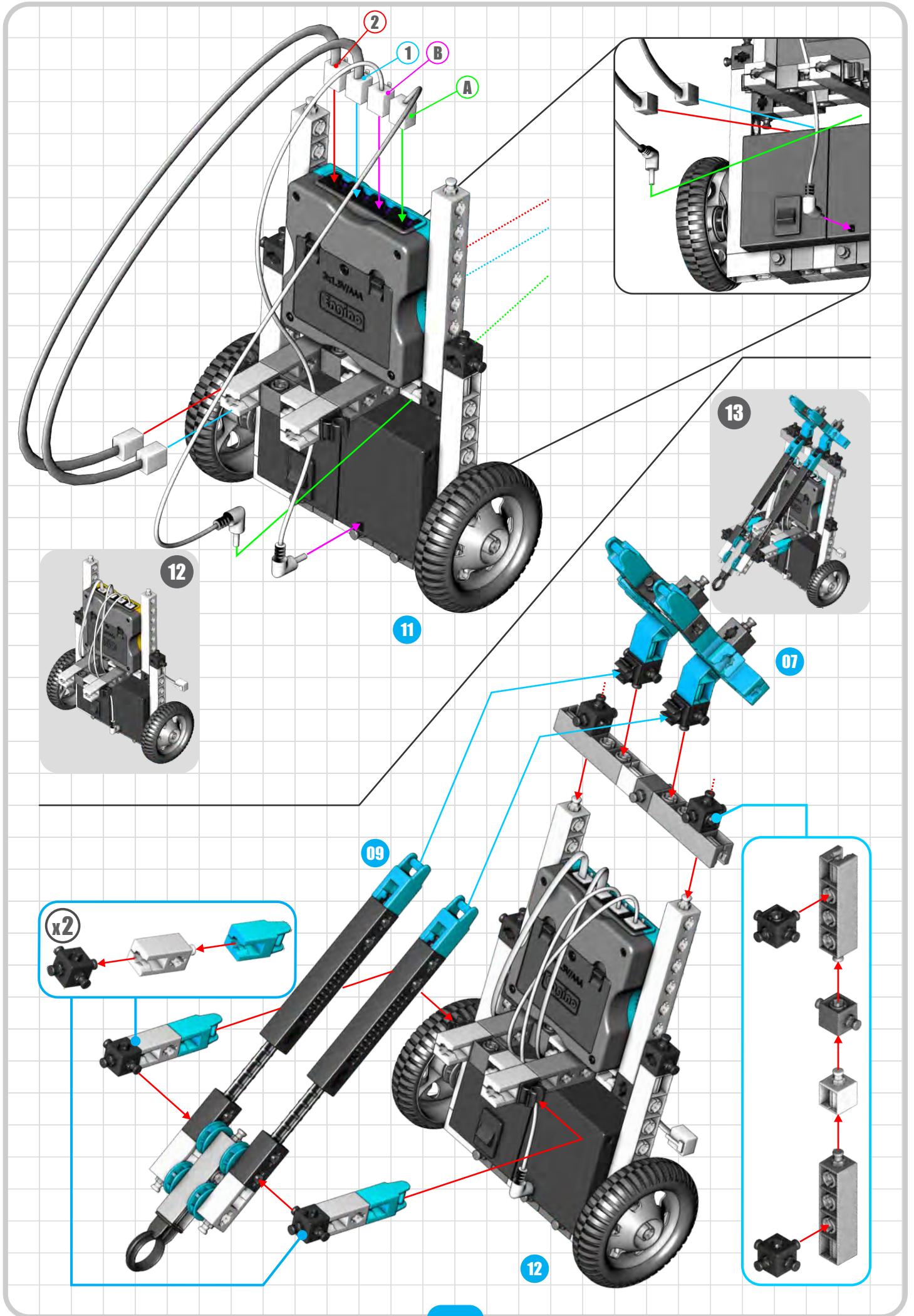


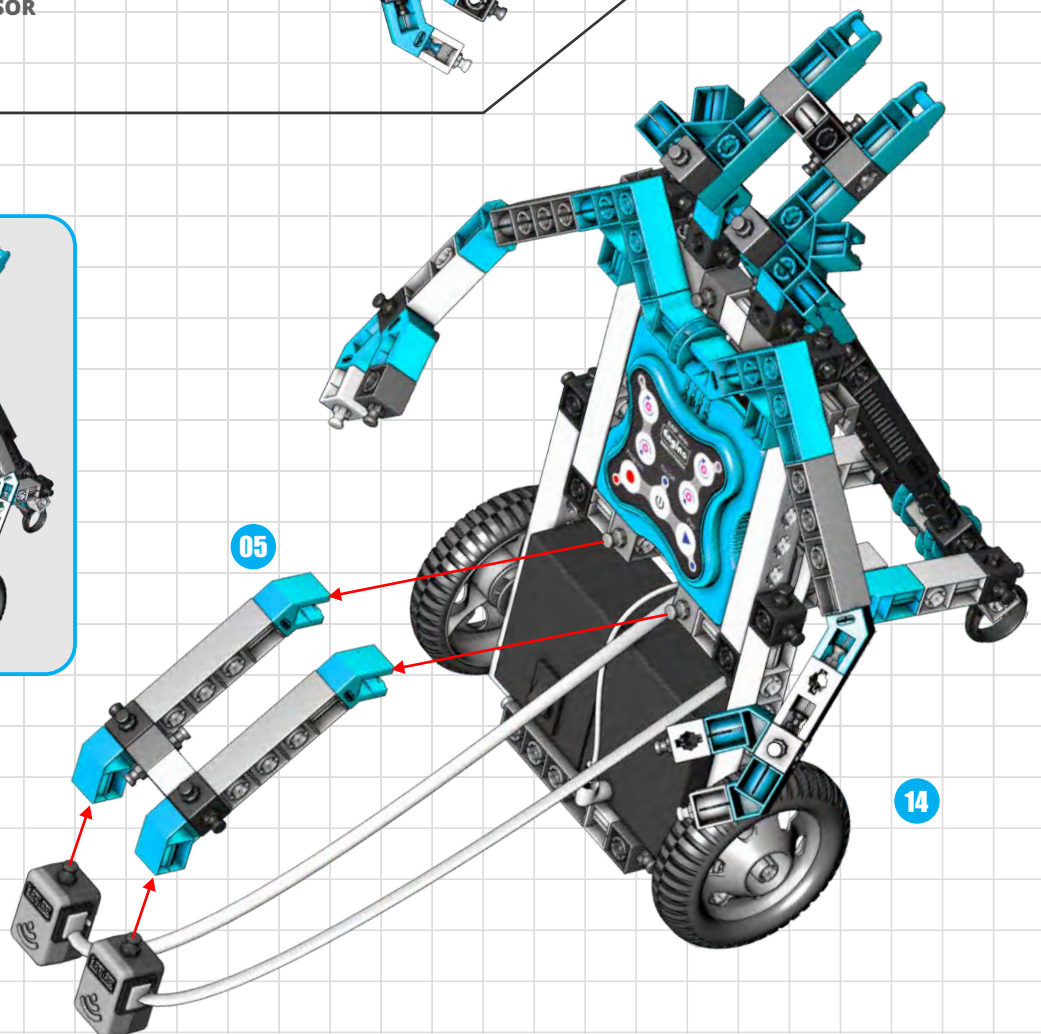
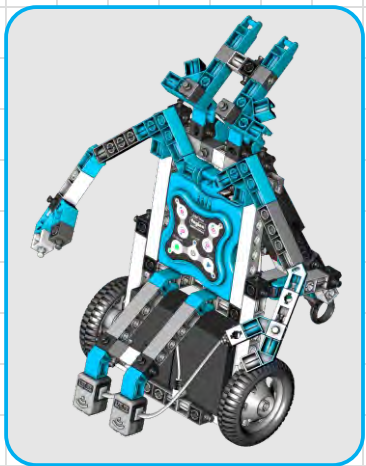
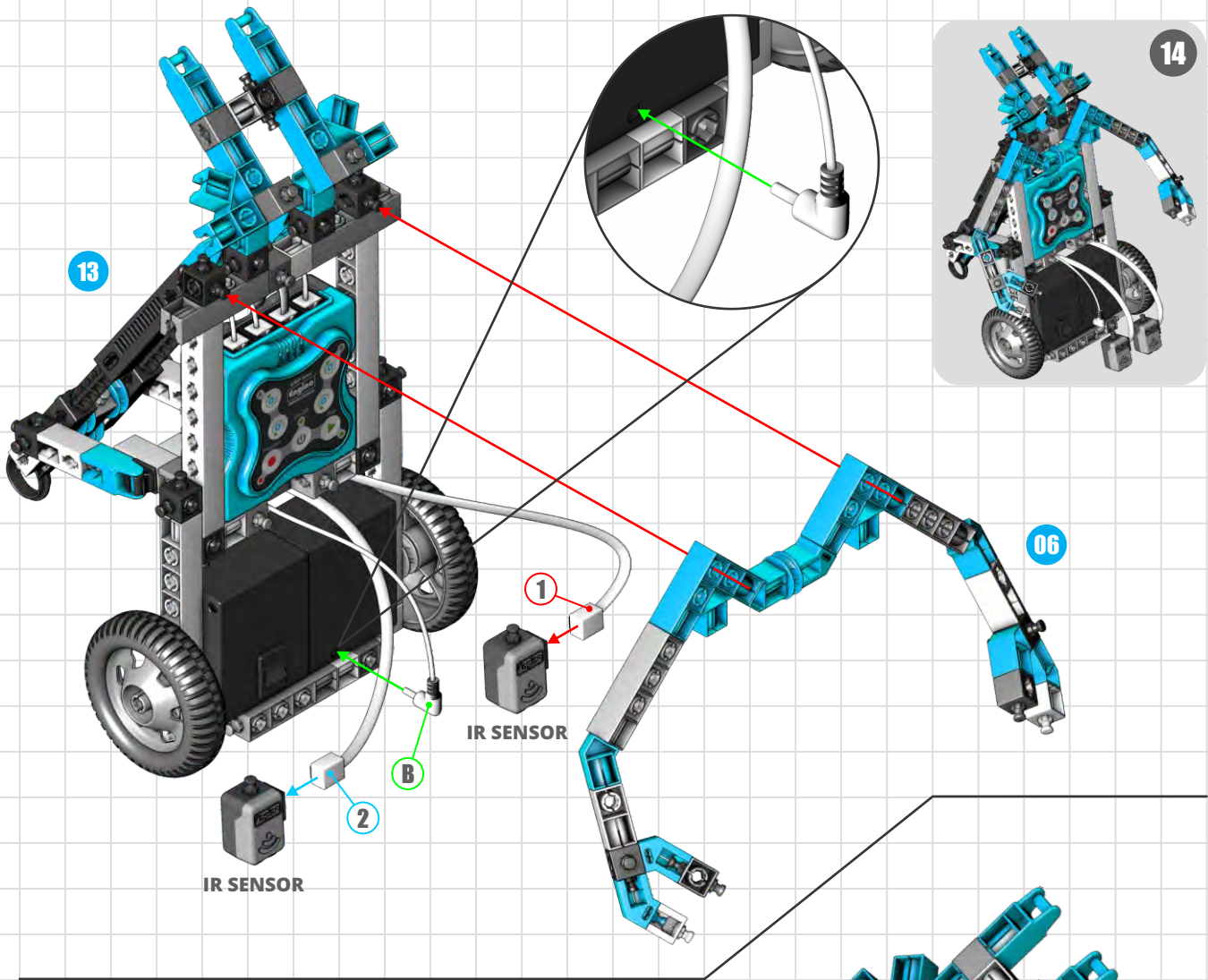
08



09



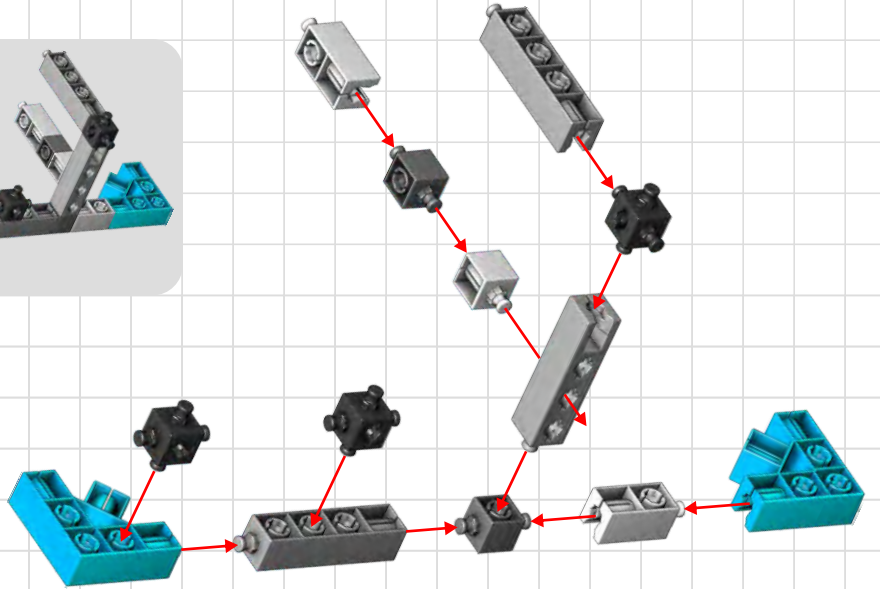
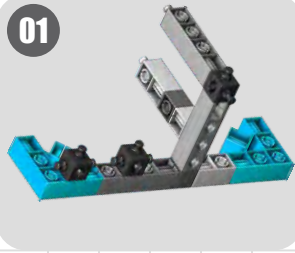




Ventilátor

4020

01



02

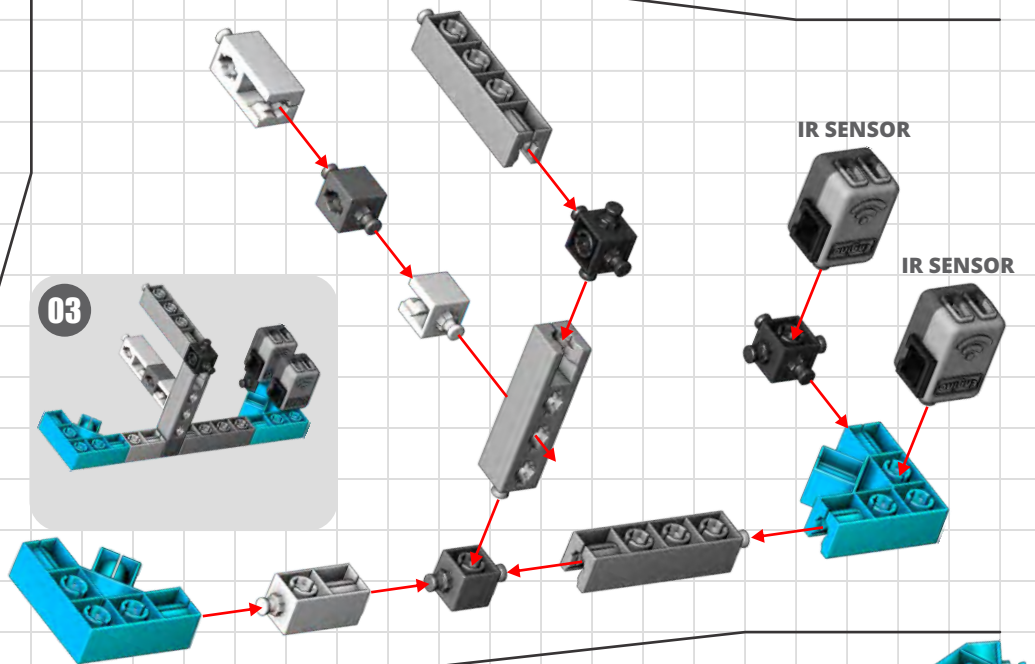


03

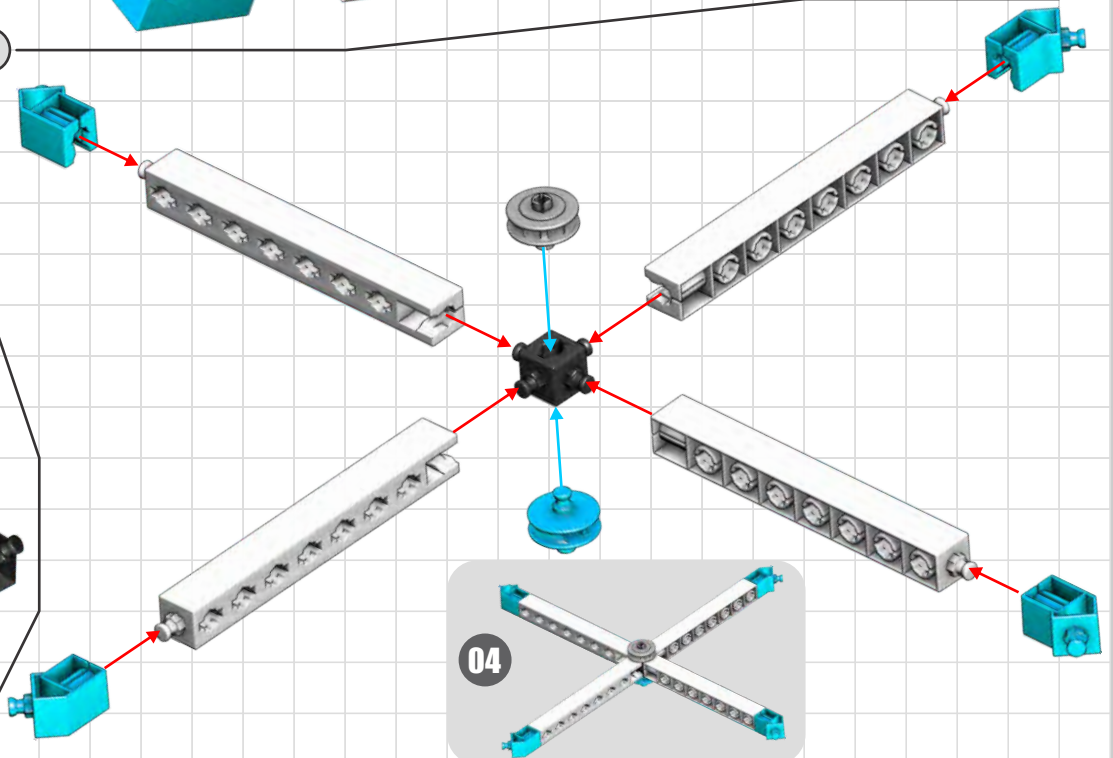
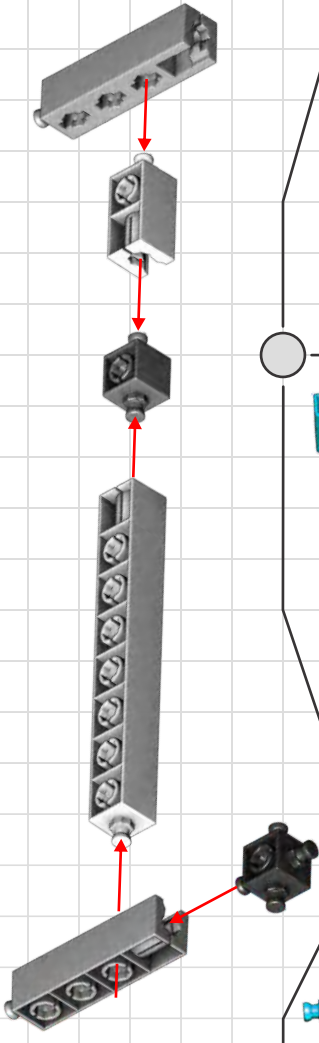
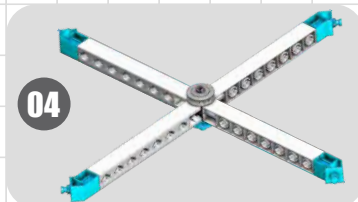


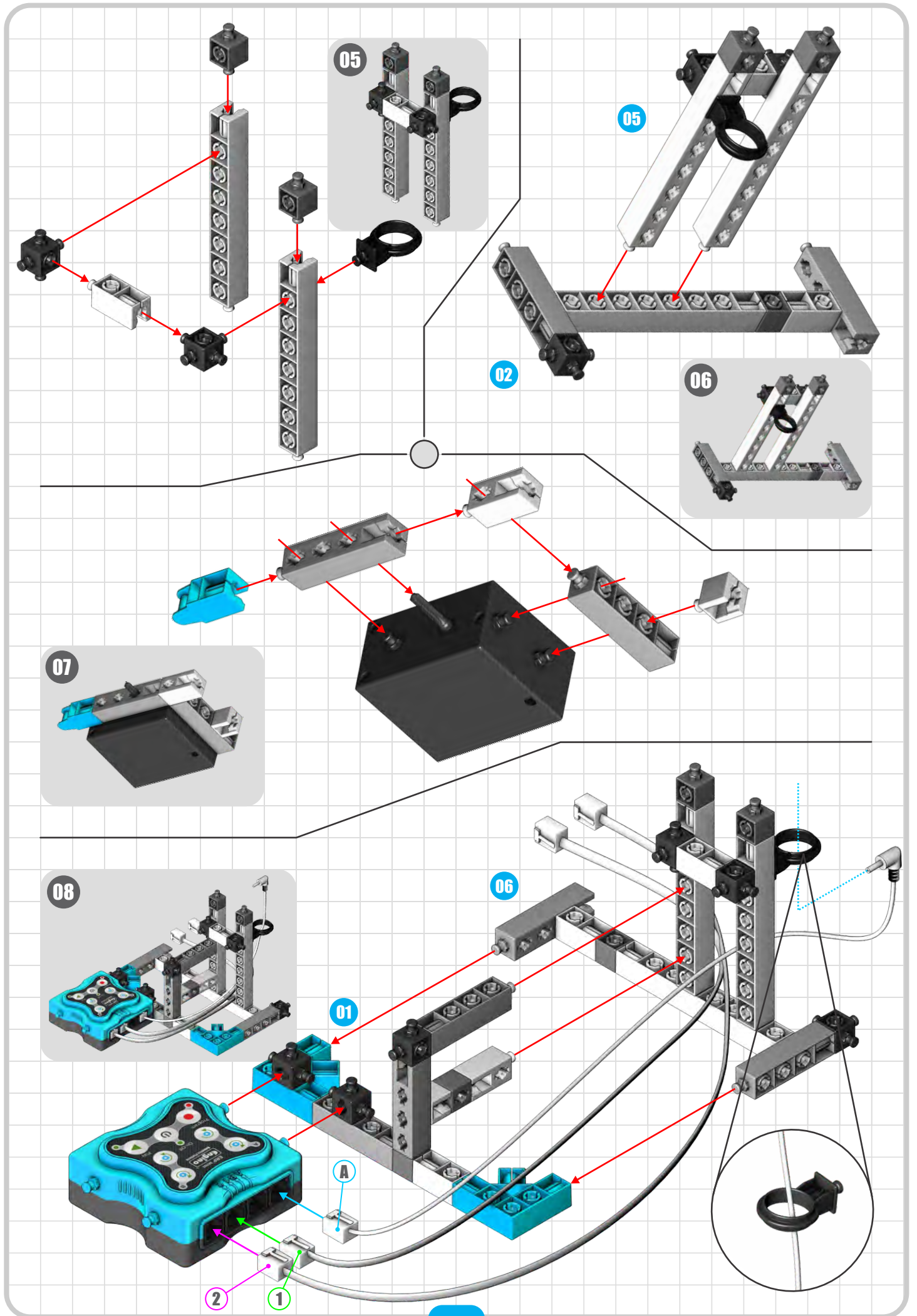
IR SENSOR

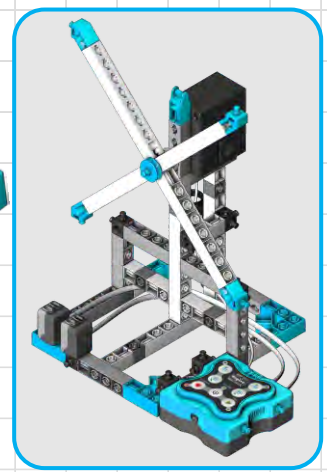
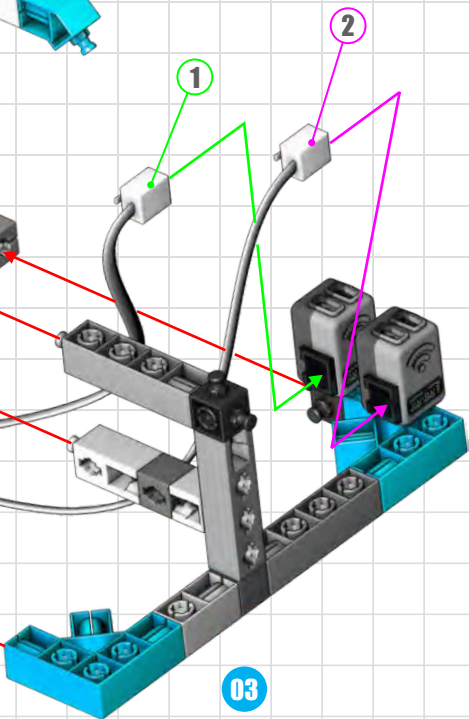
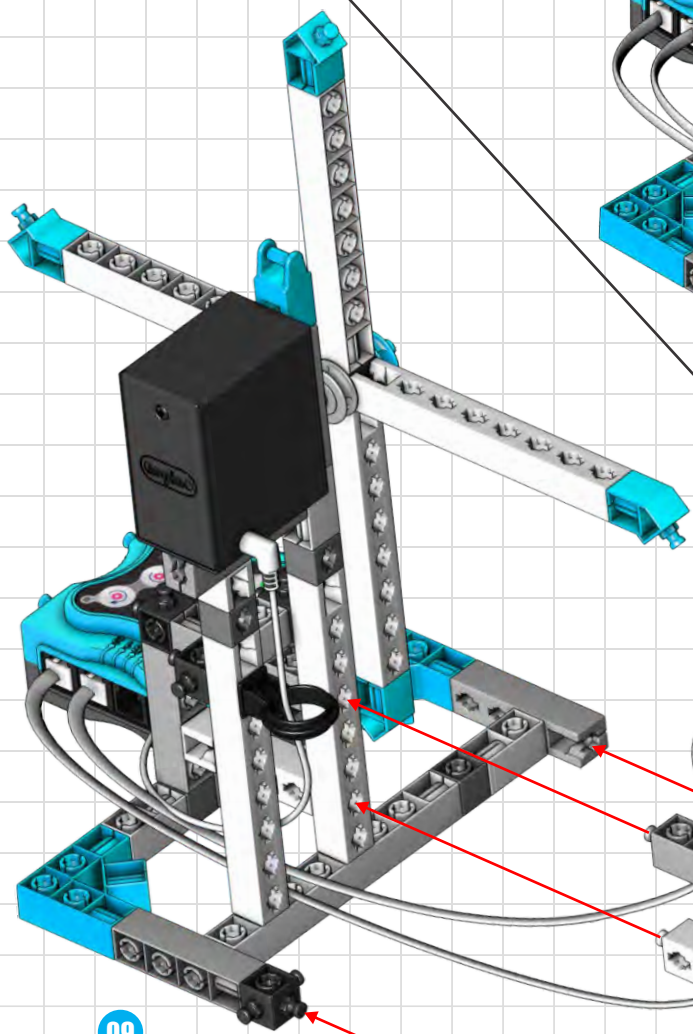
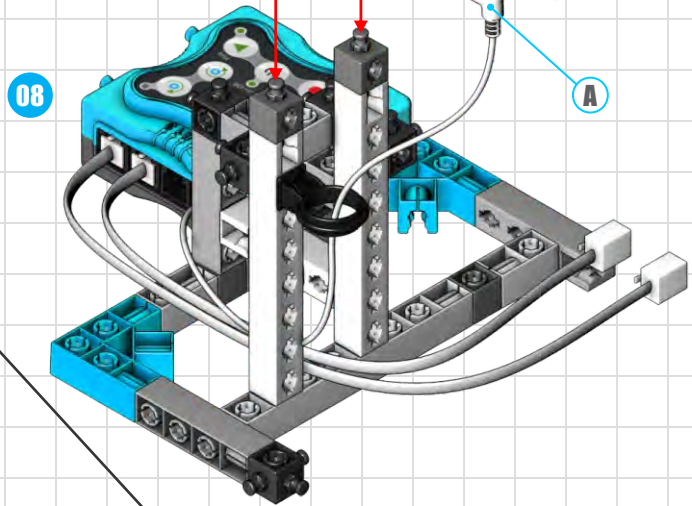
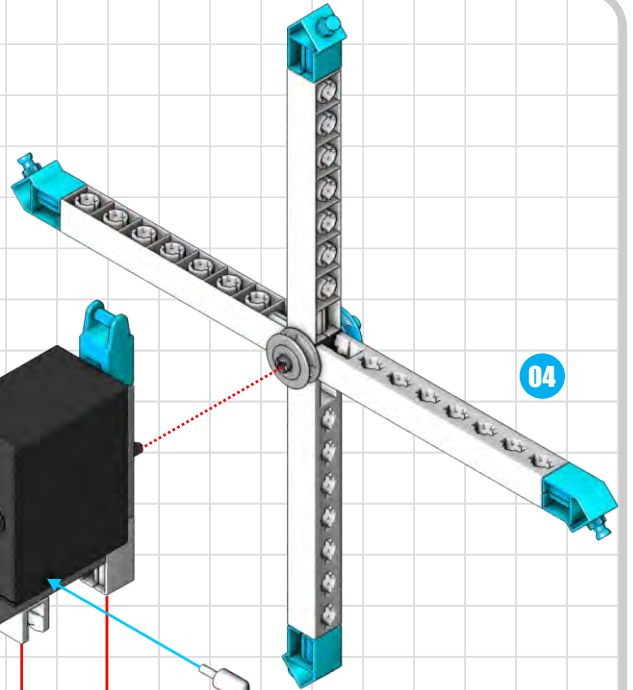
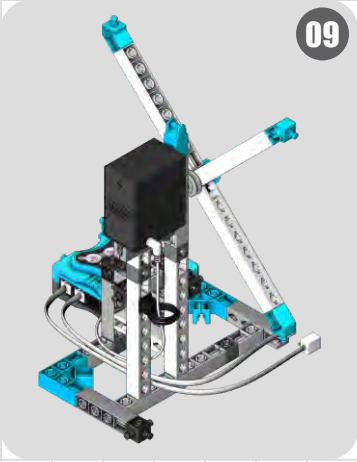
IR SENSOR



04



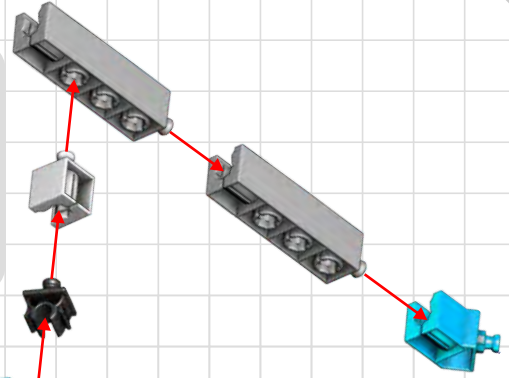
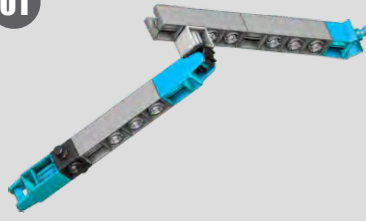




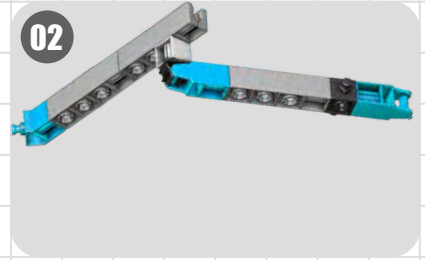
Hexapod

4021

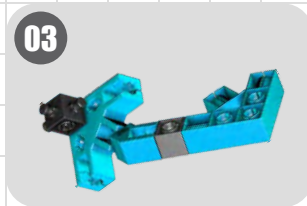
01



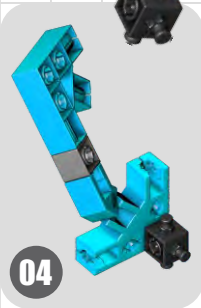
02



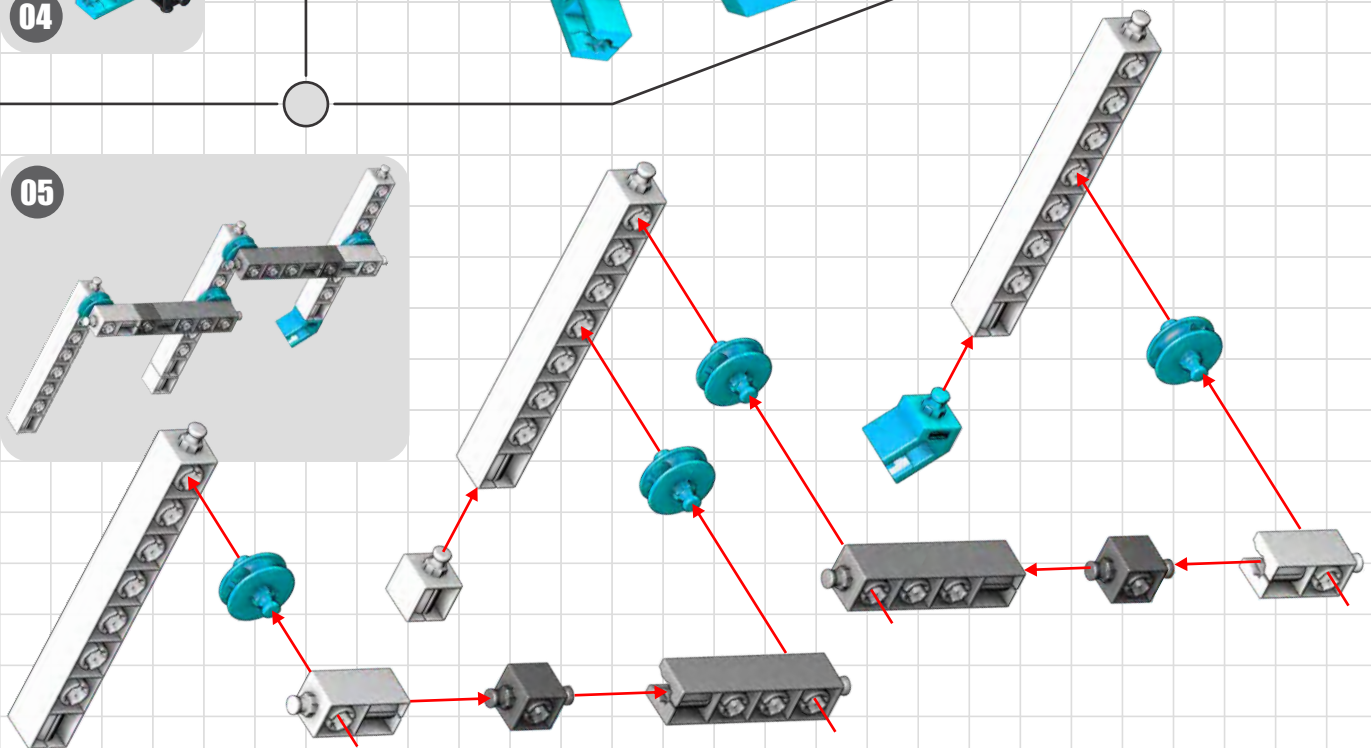
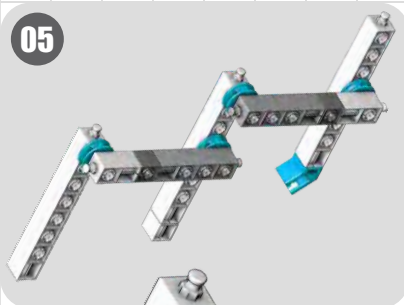
03

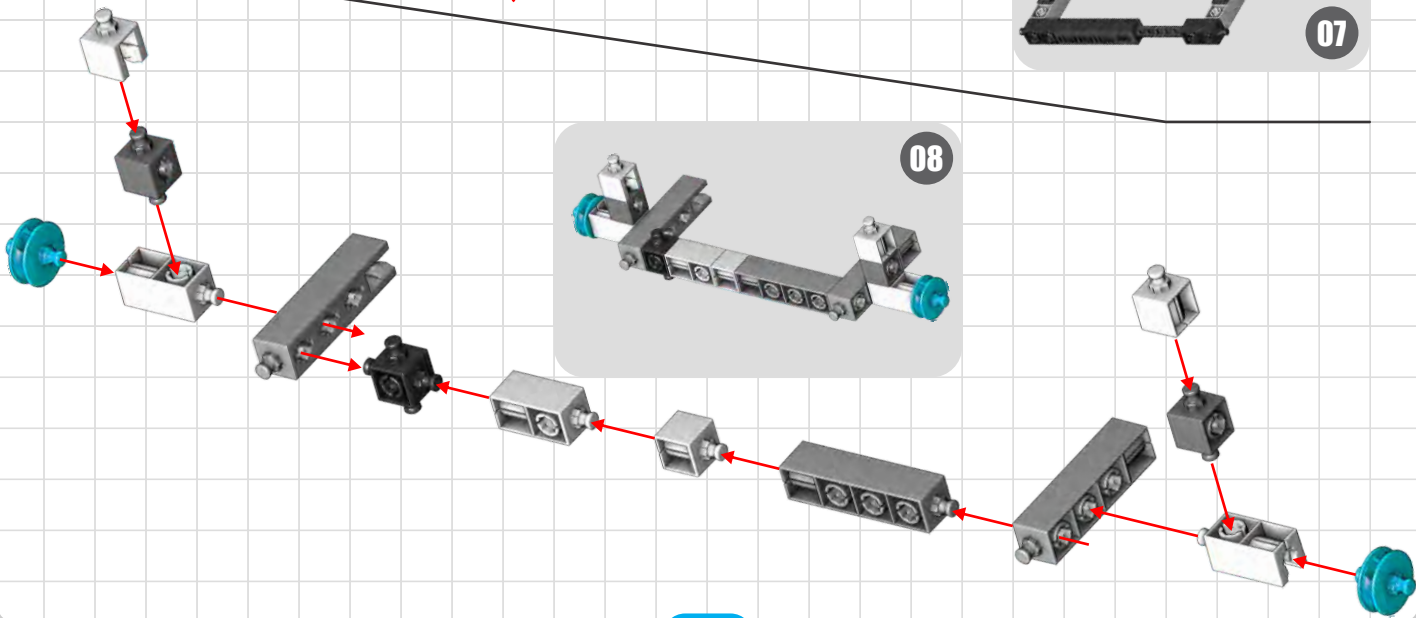
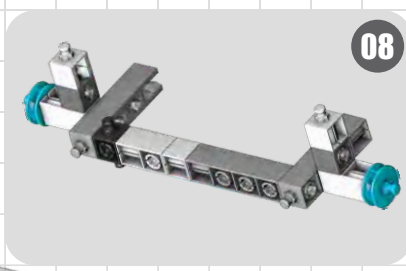
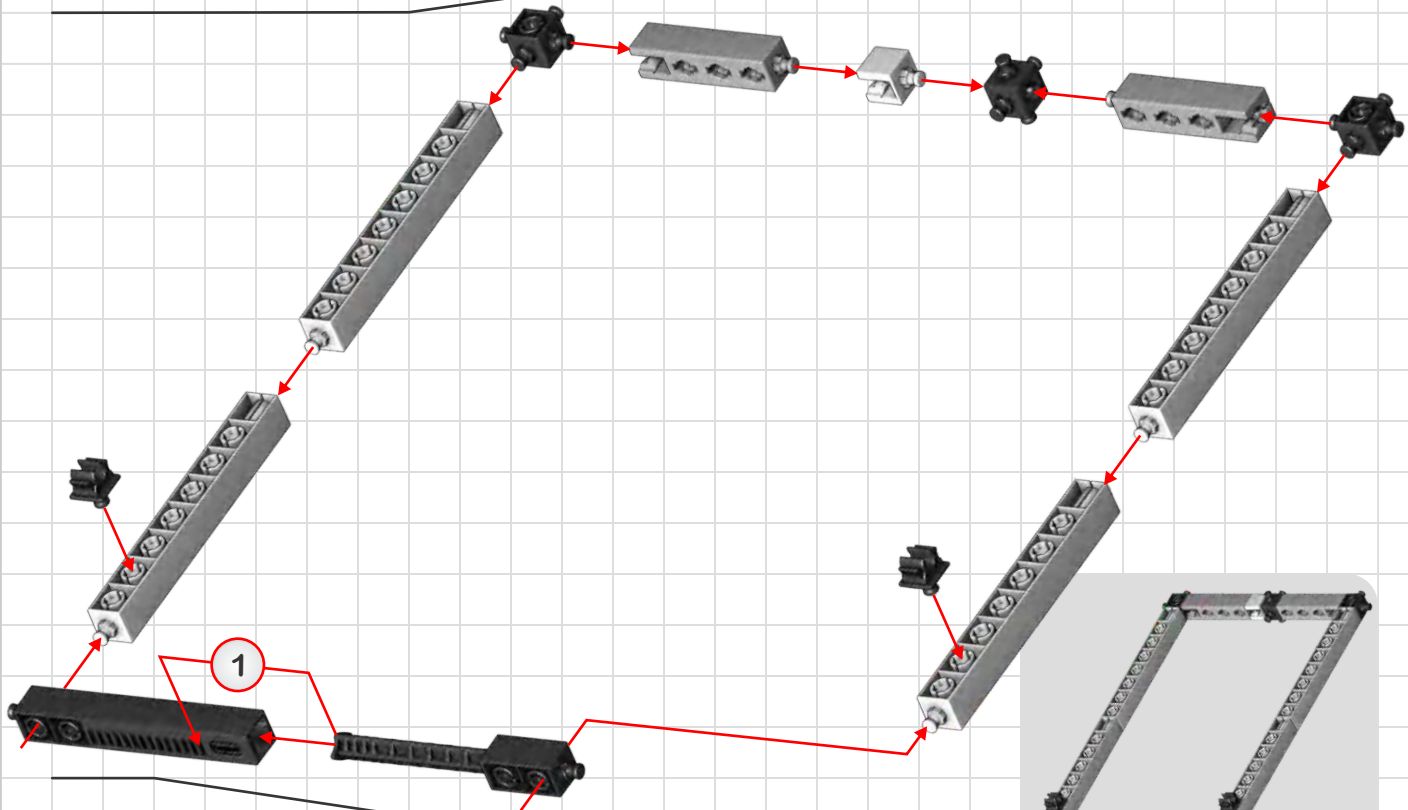
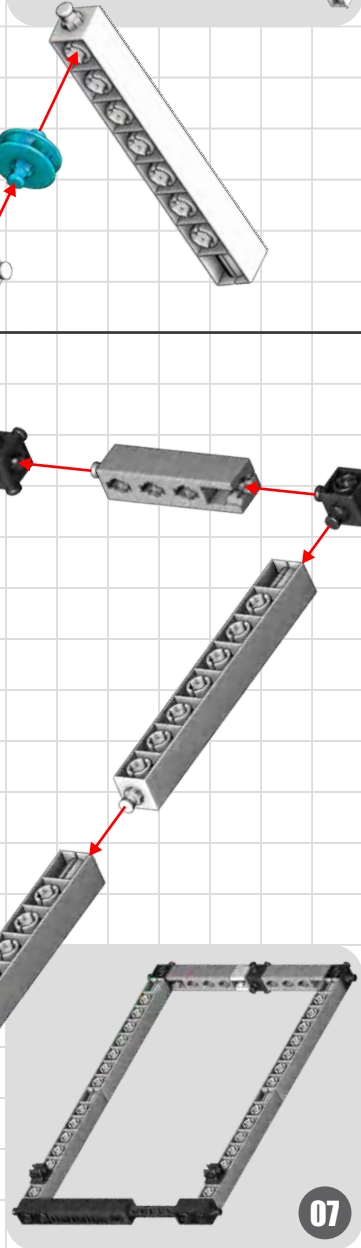
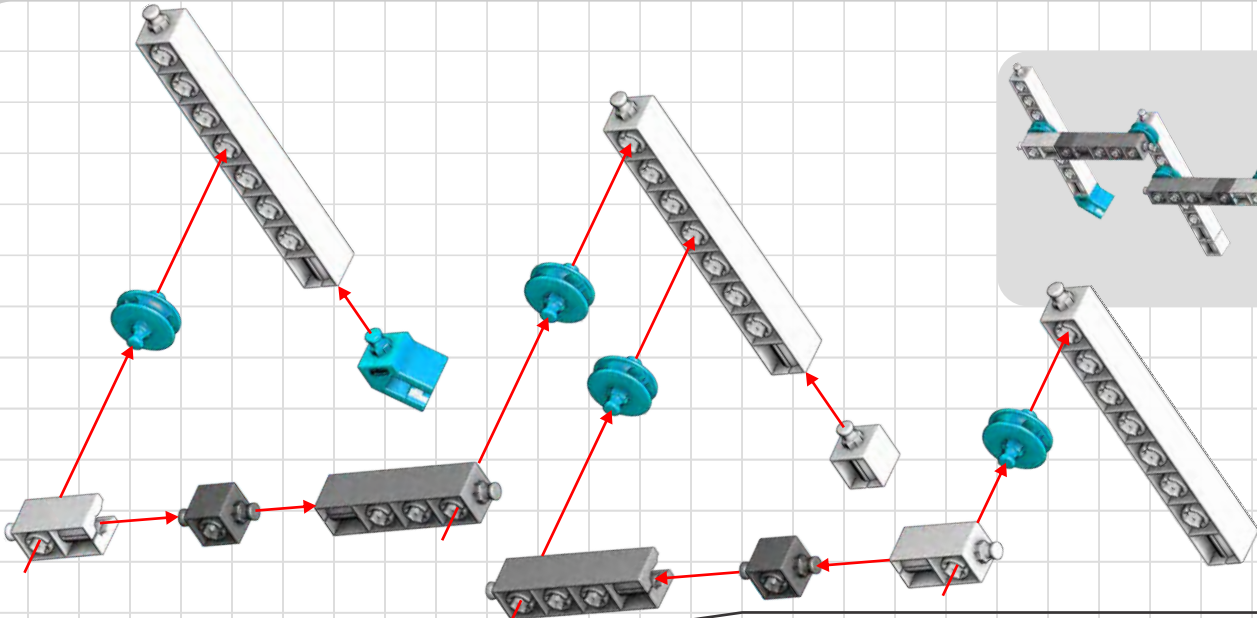
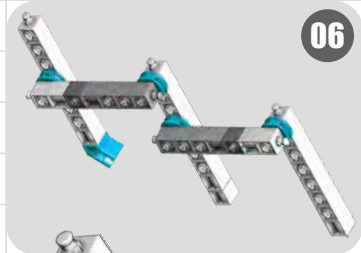


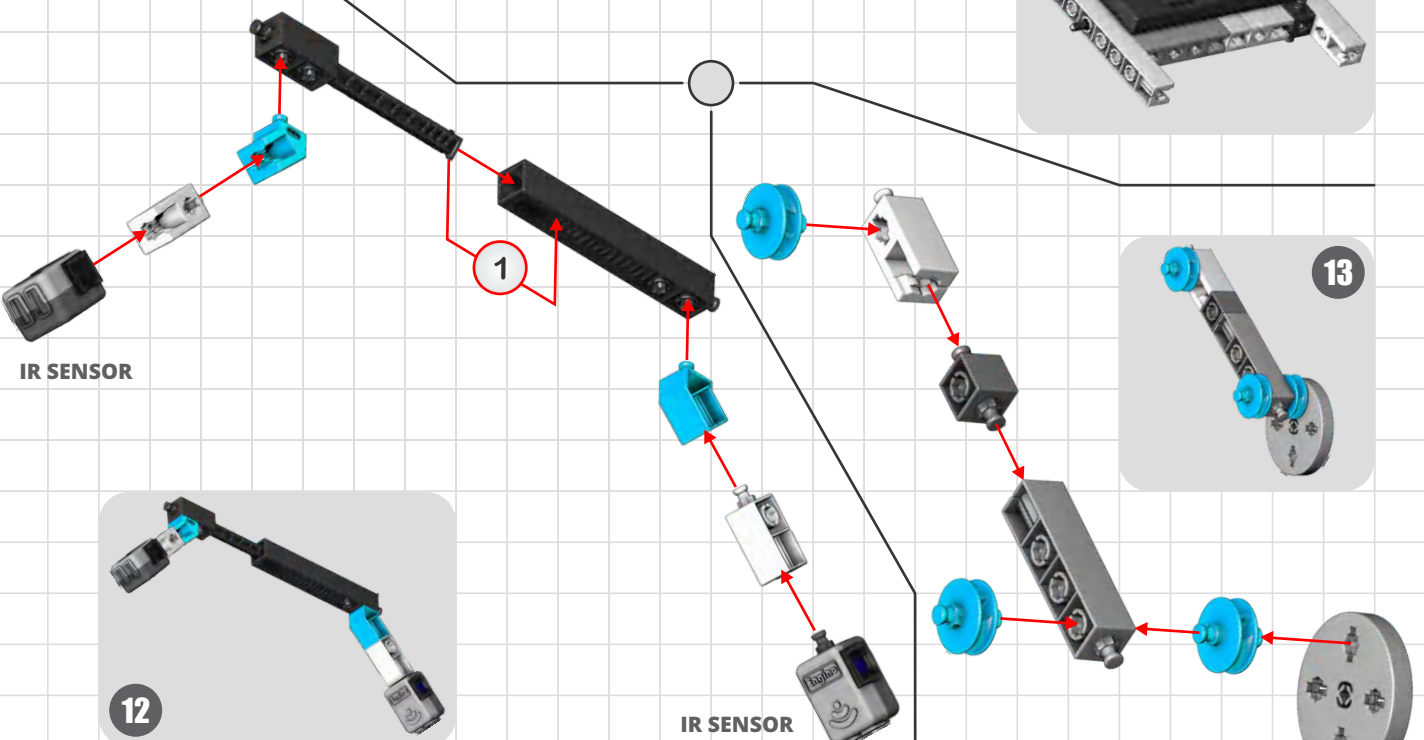
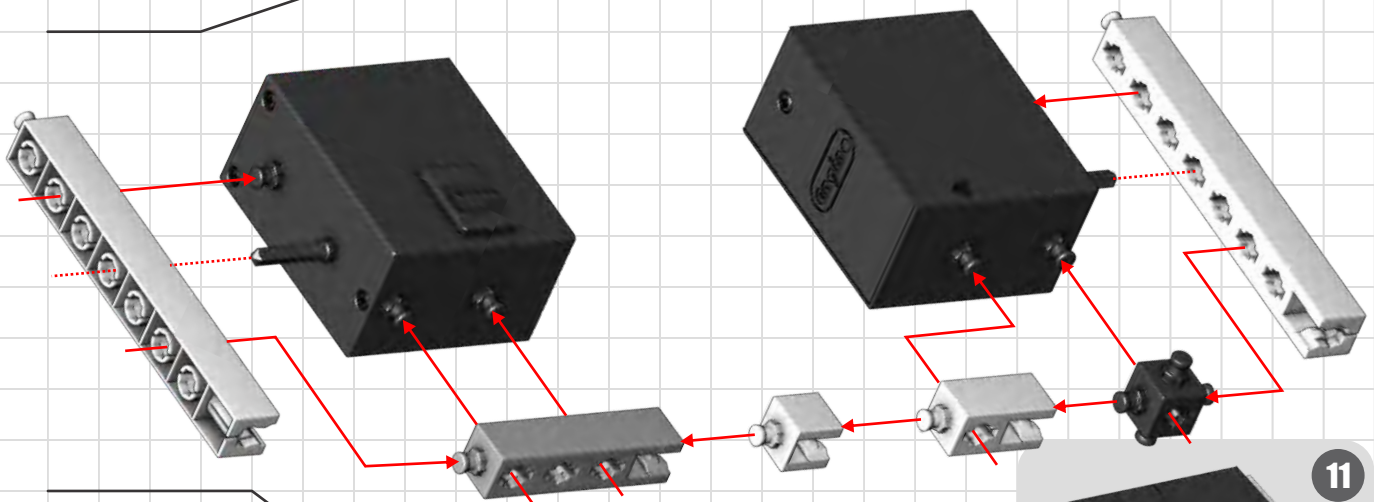
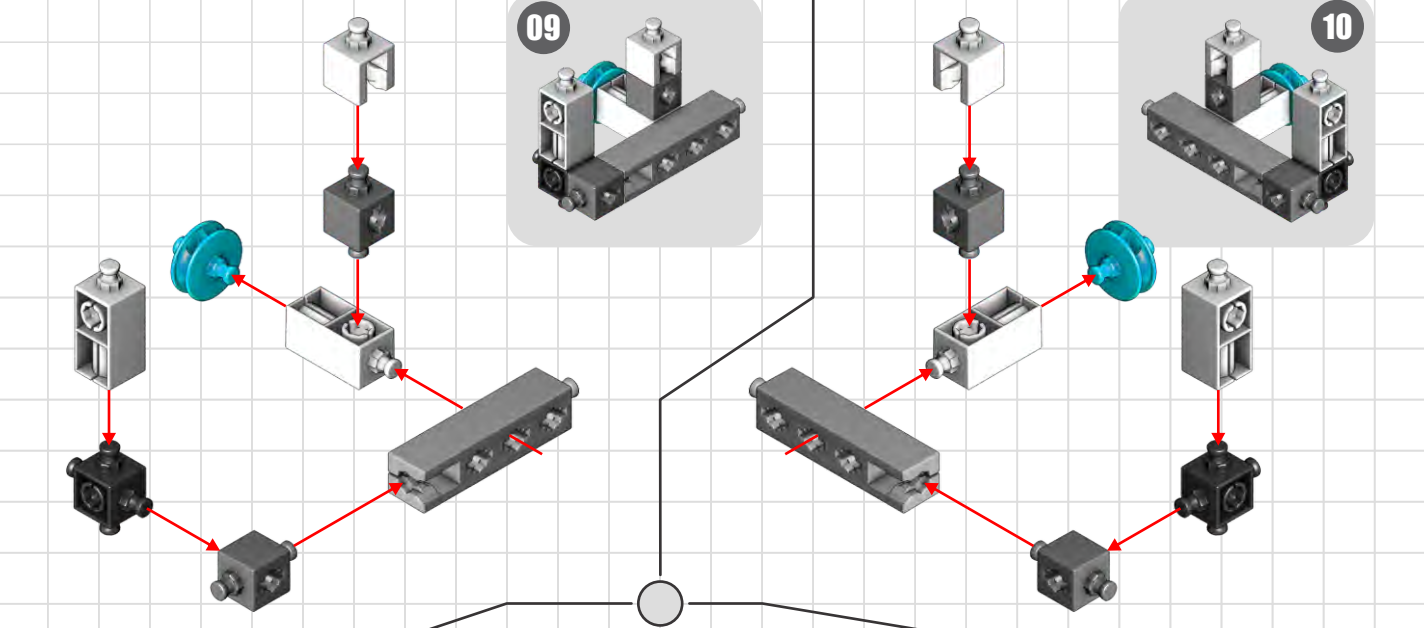
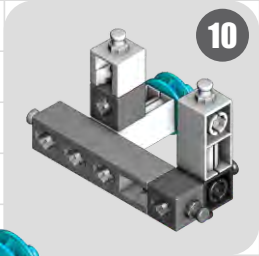
04



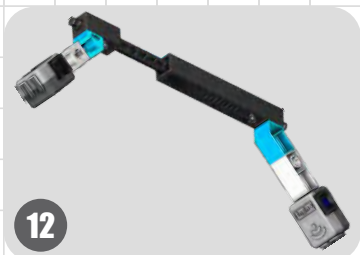
05



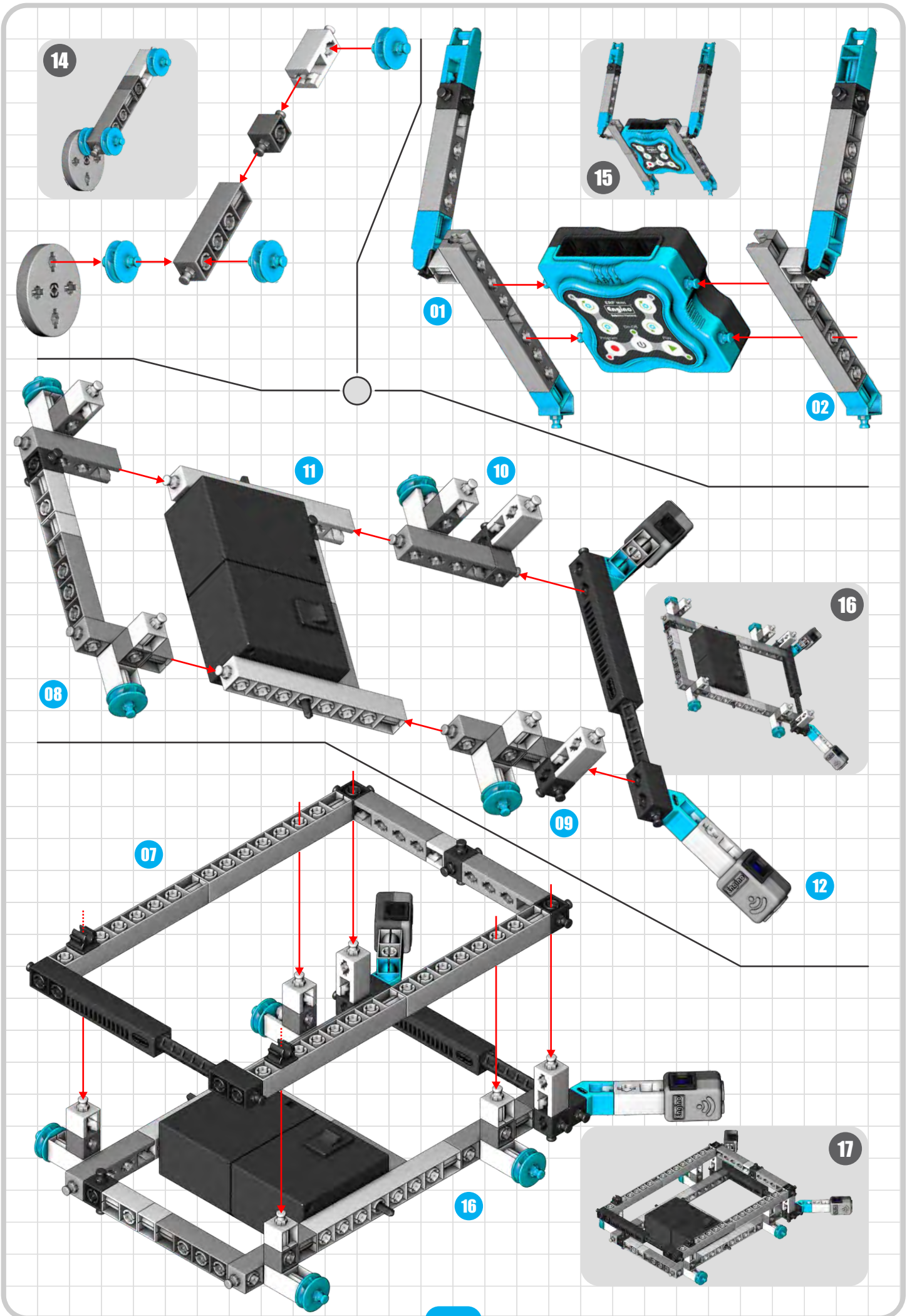


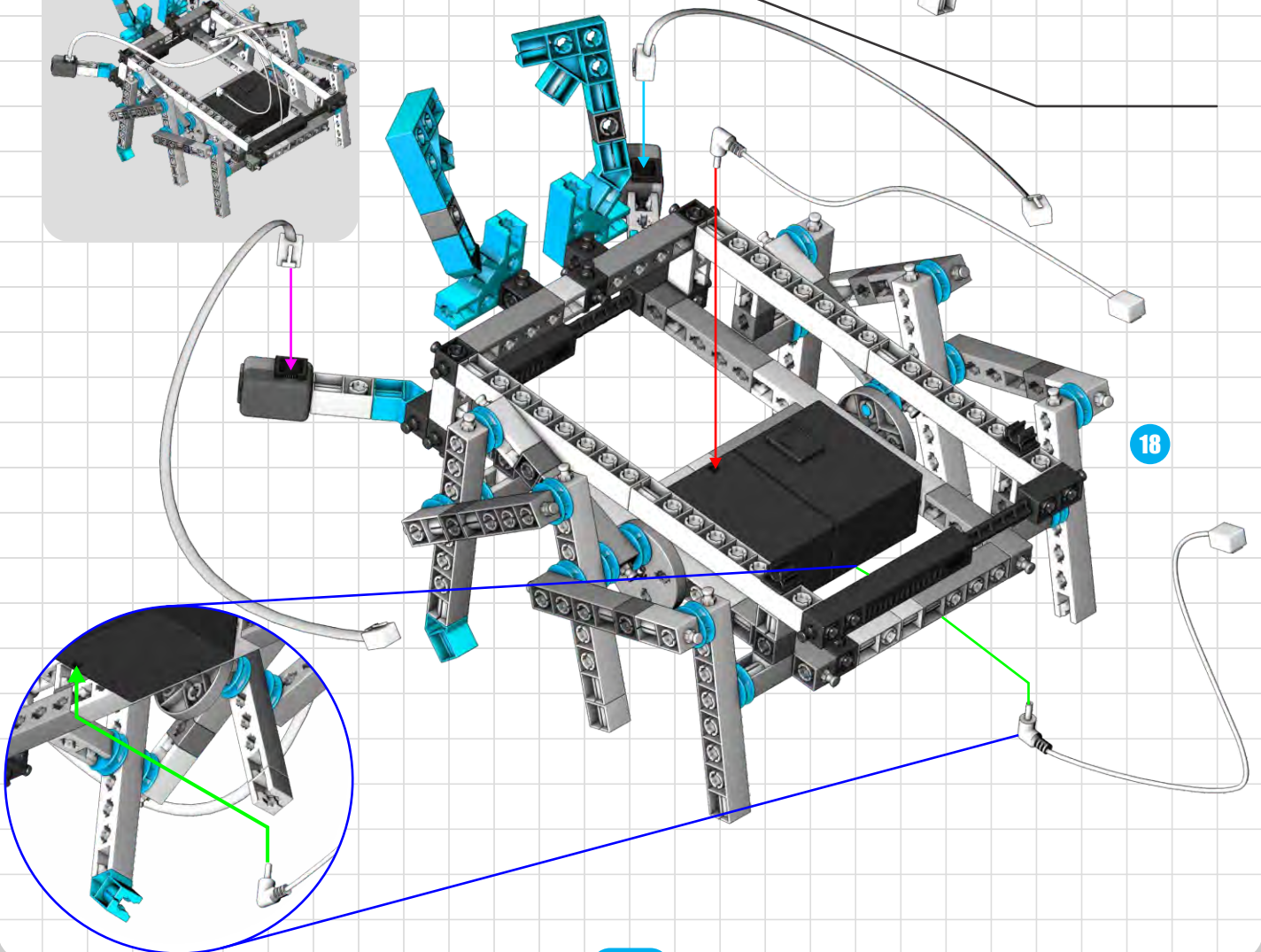
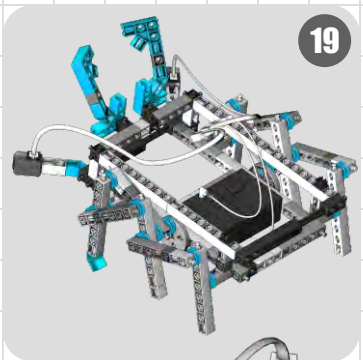
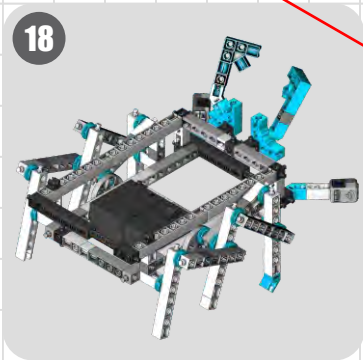
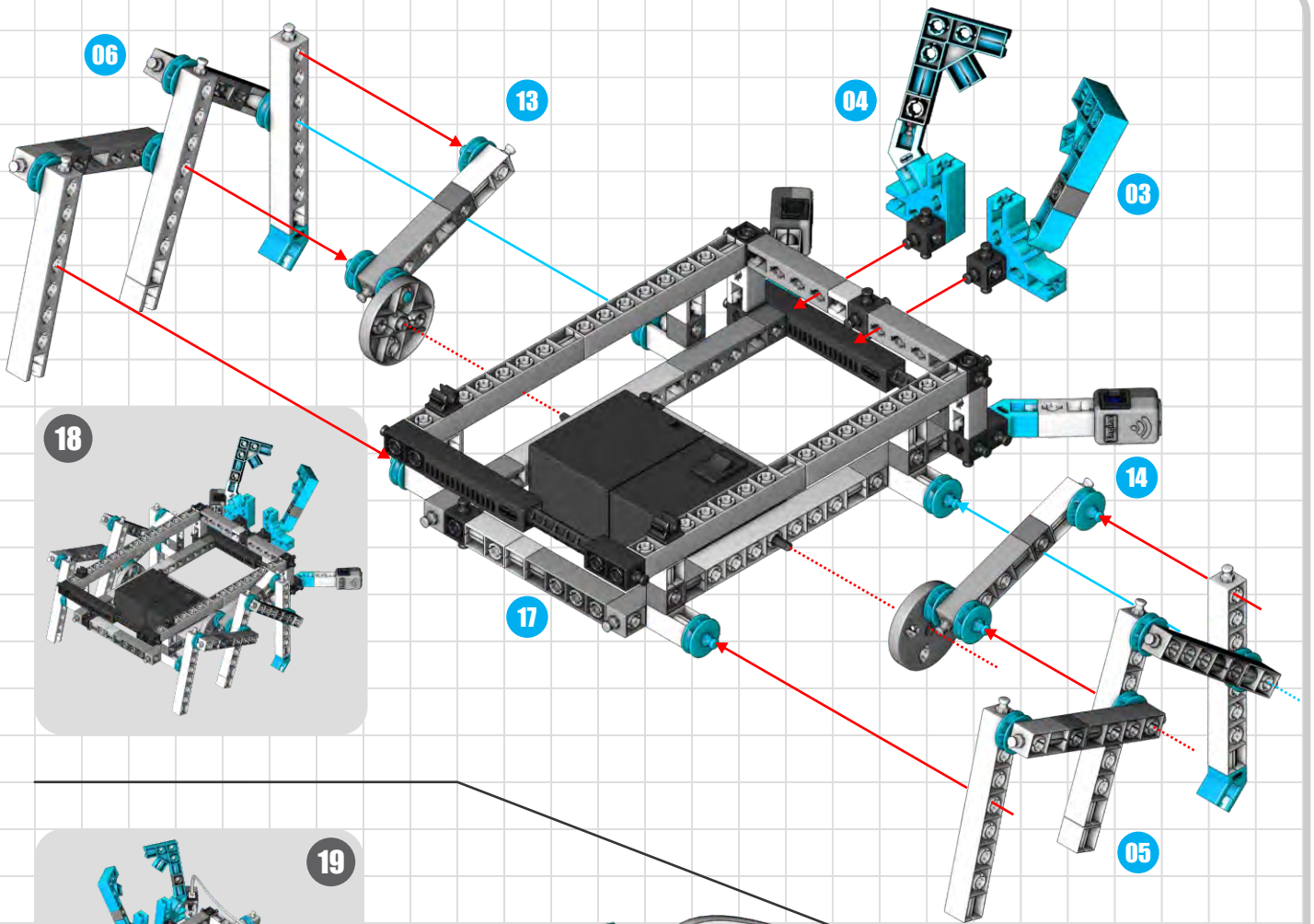


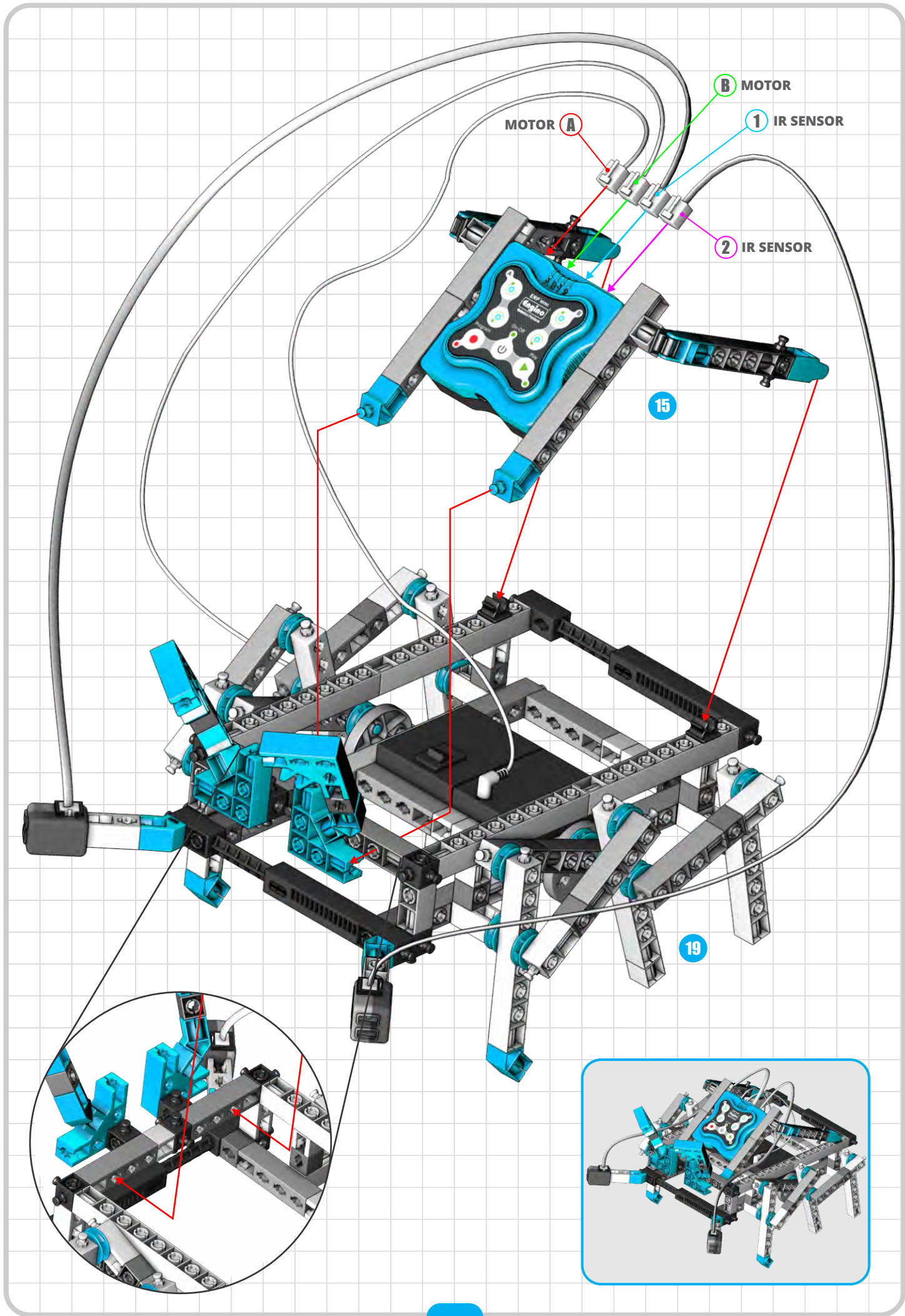
IR SENSOR



IR SENSOR







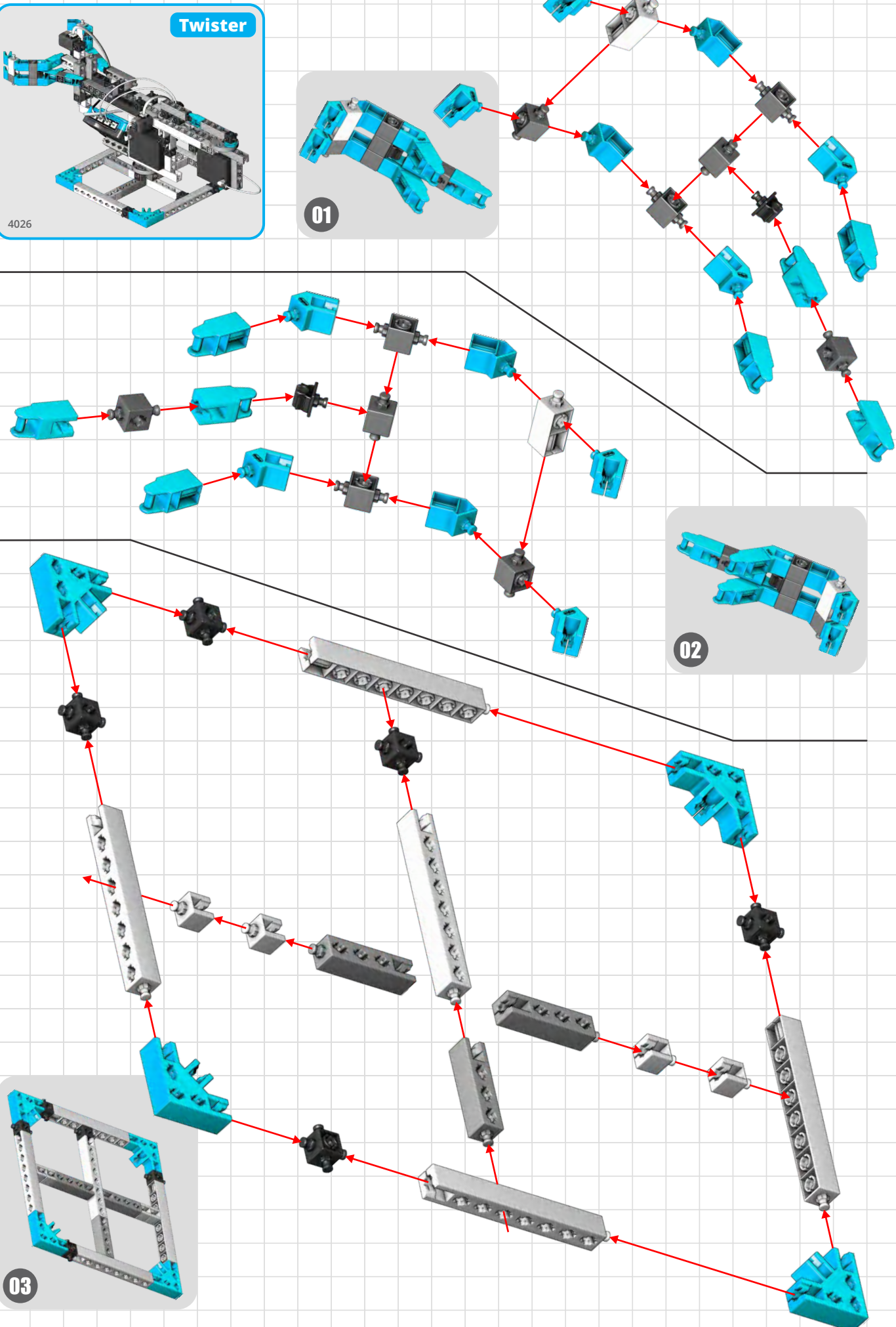
Twister

4026

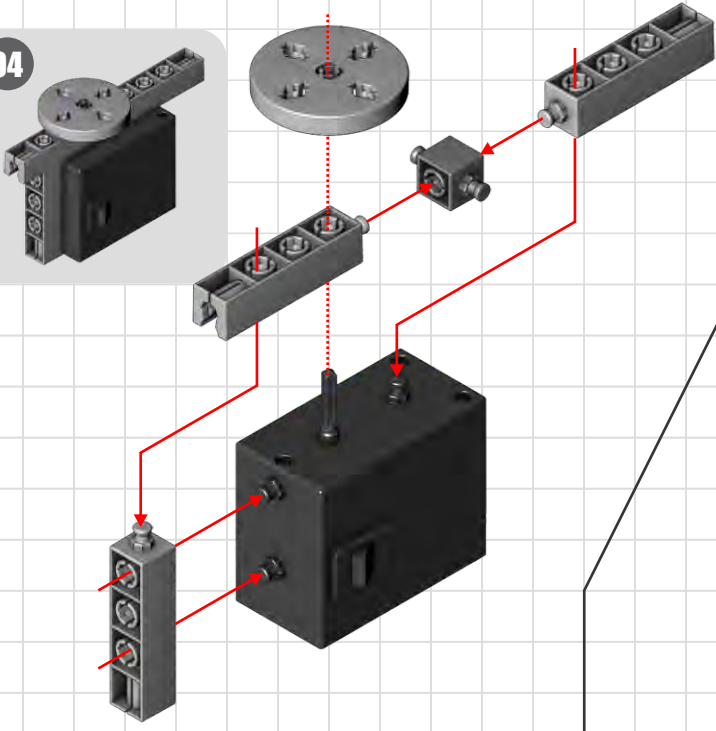
01

02

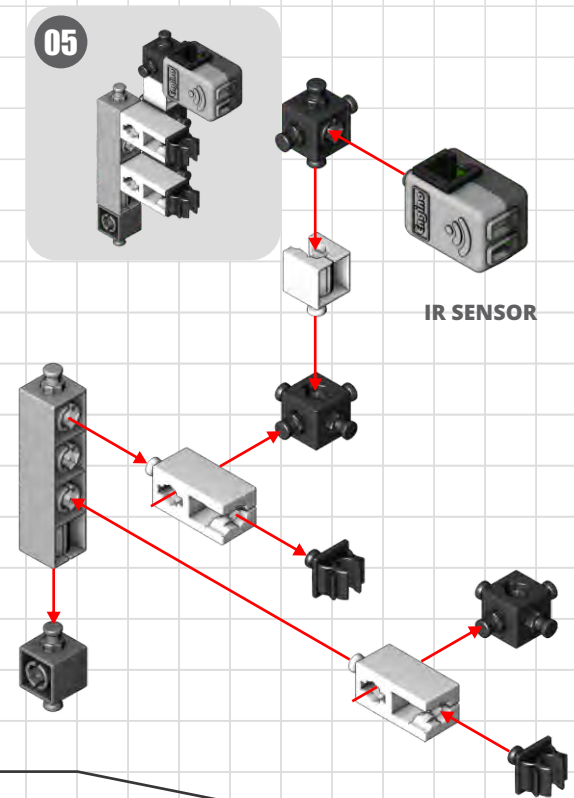
03



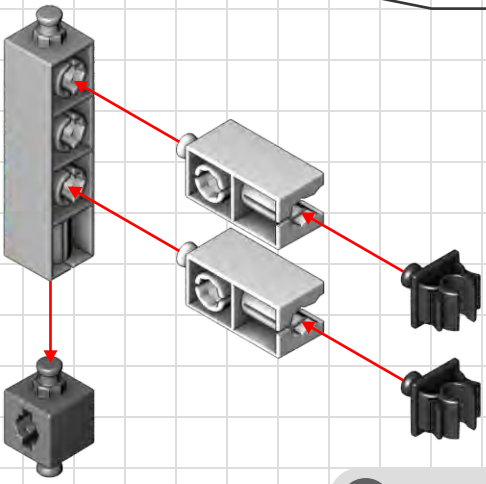
04



05



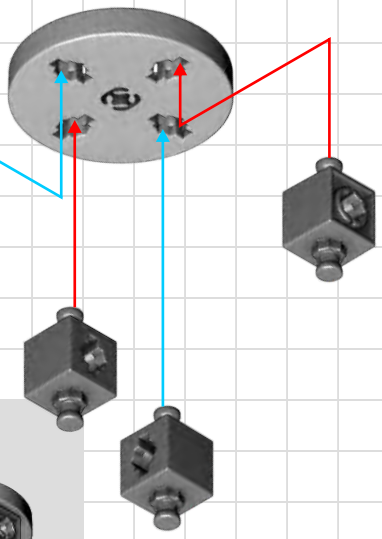
IR SENSOR



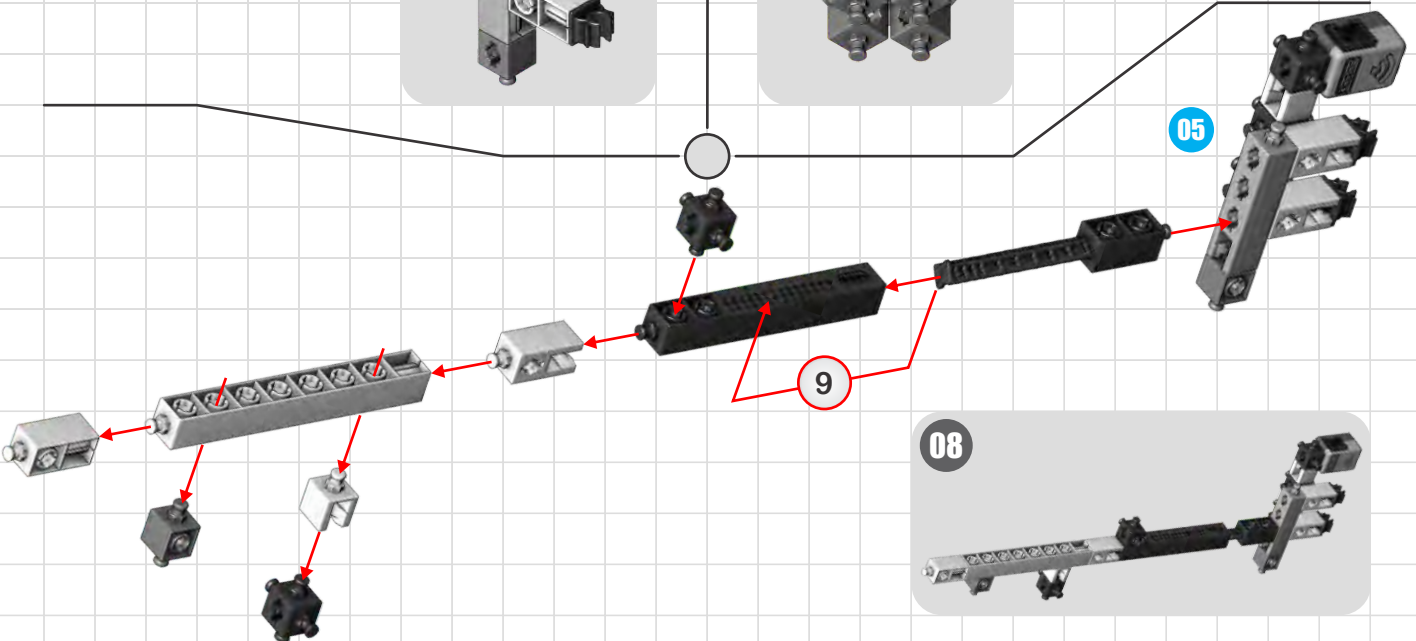
06



07

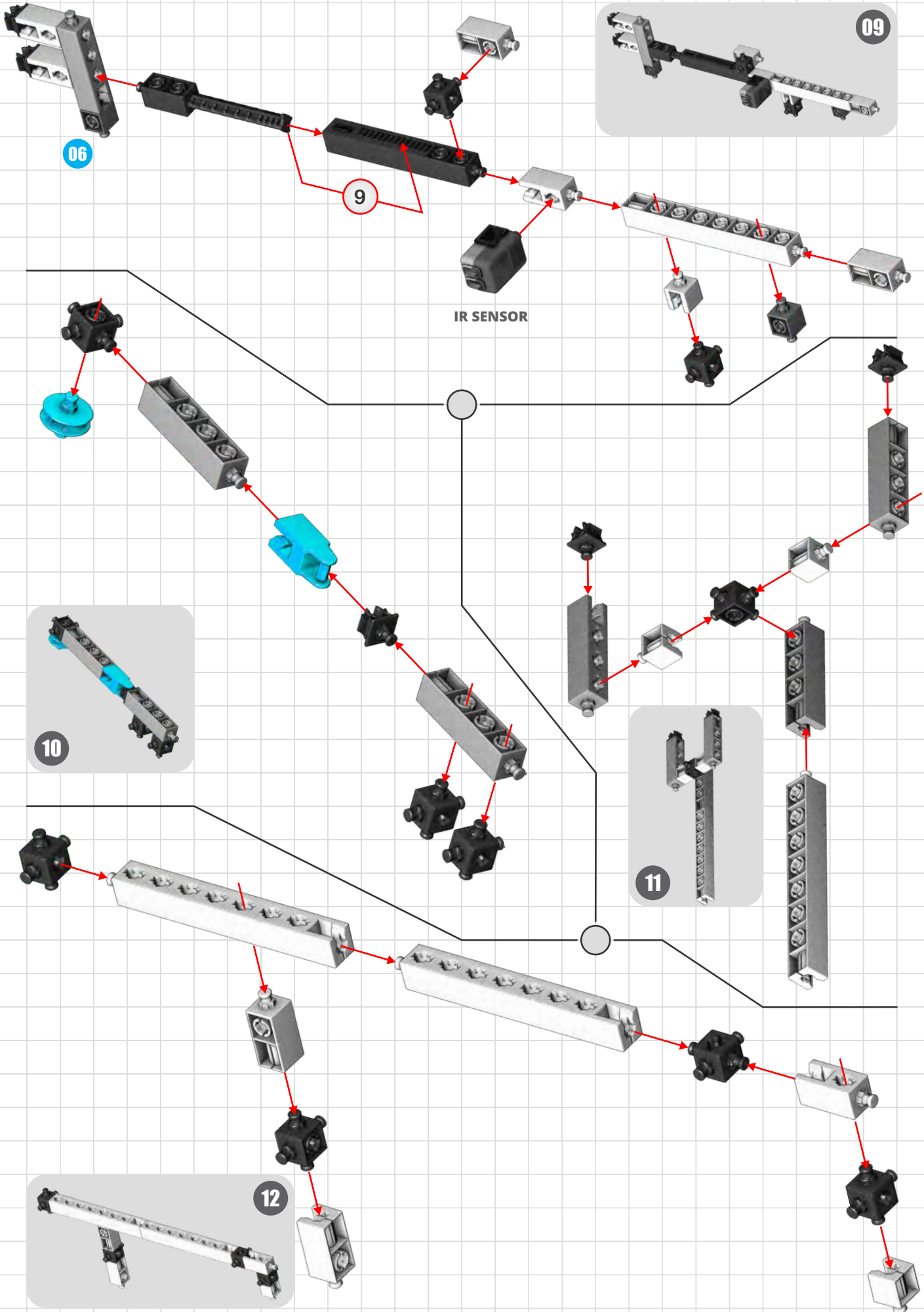


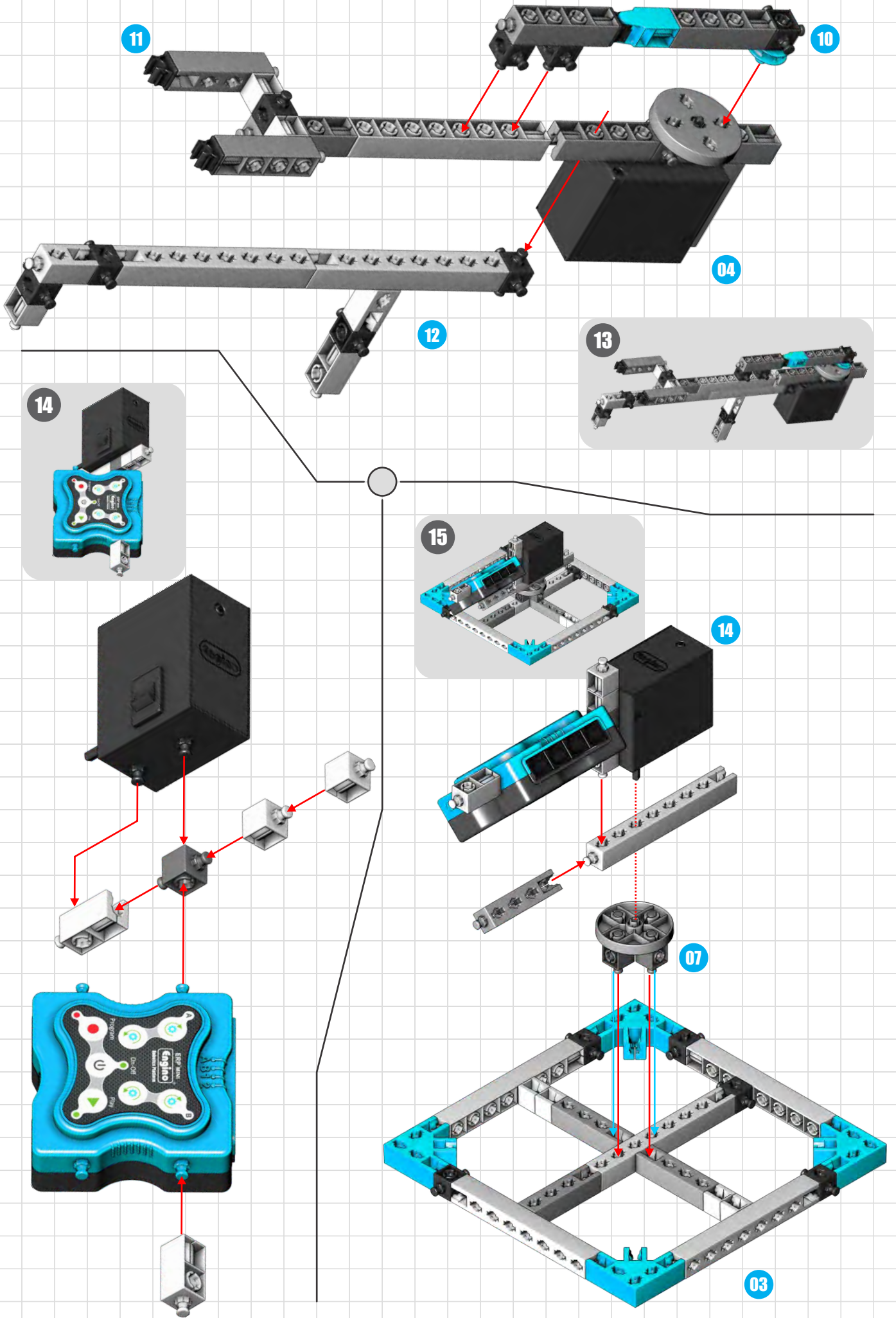
05



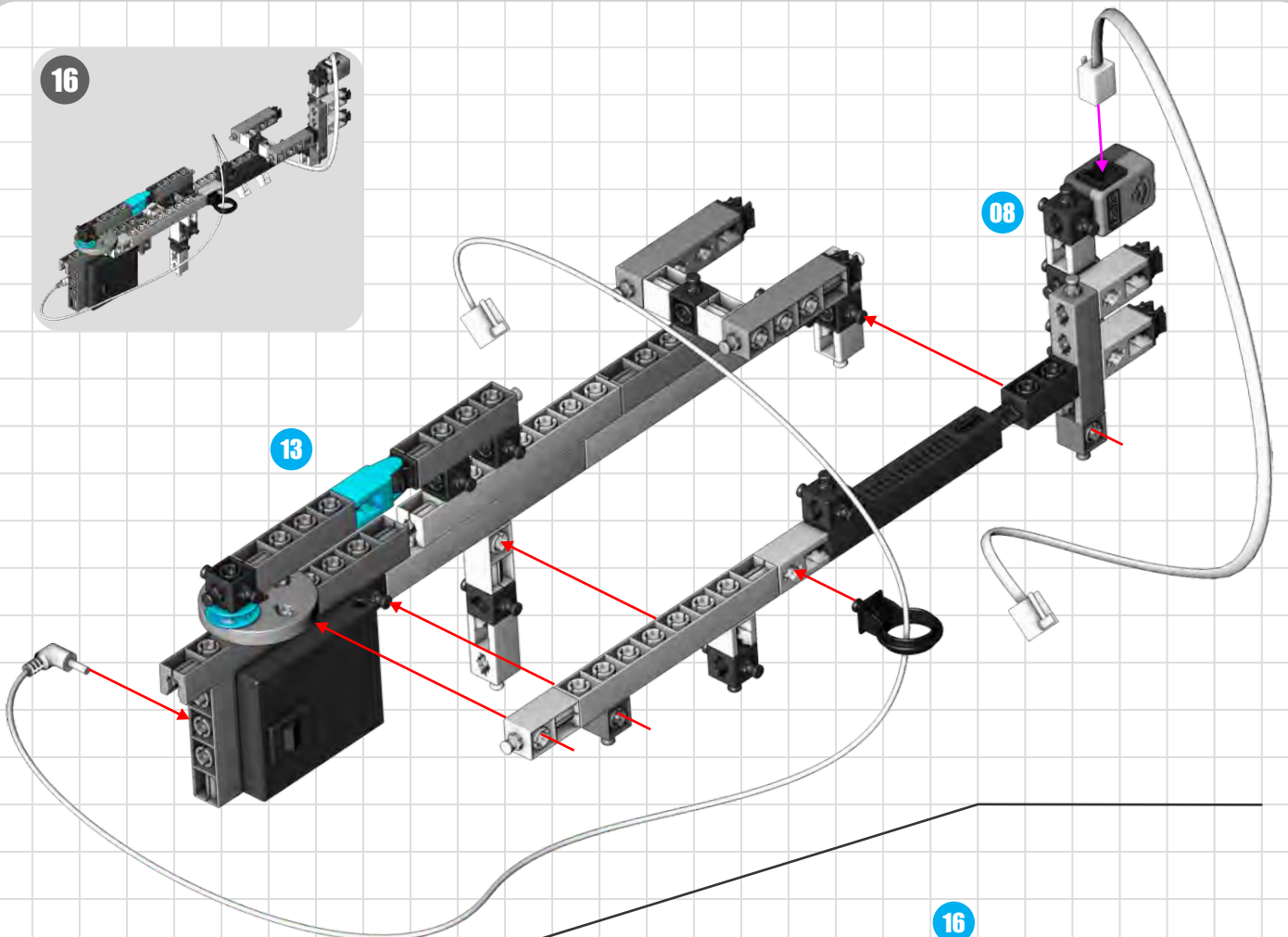
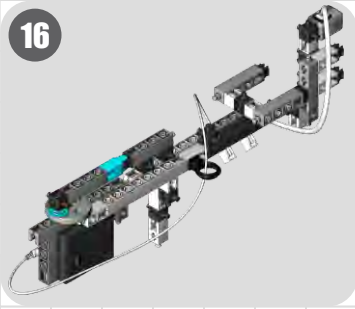
08



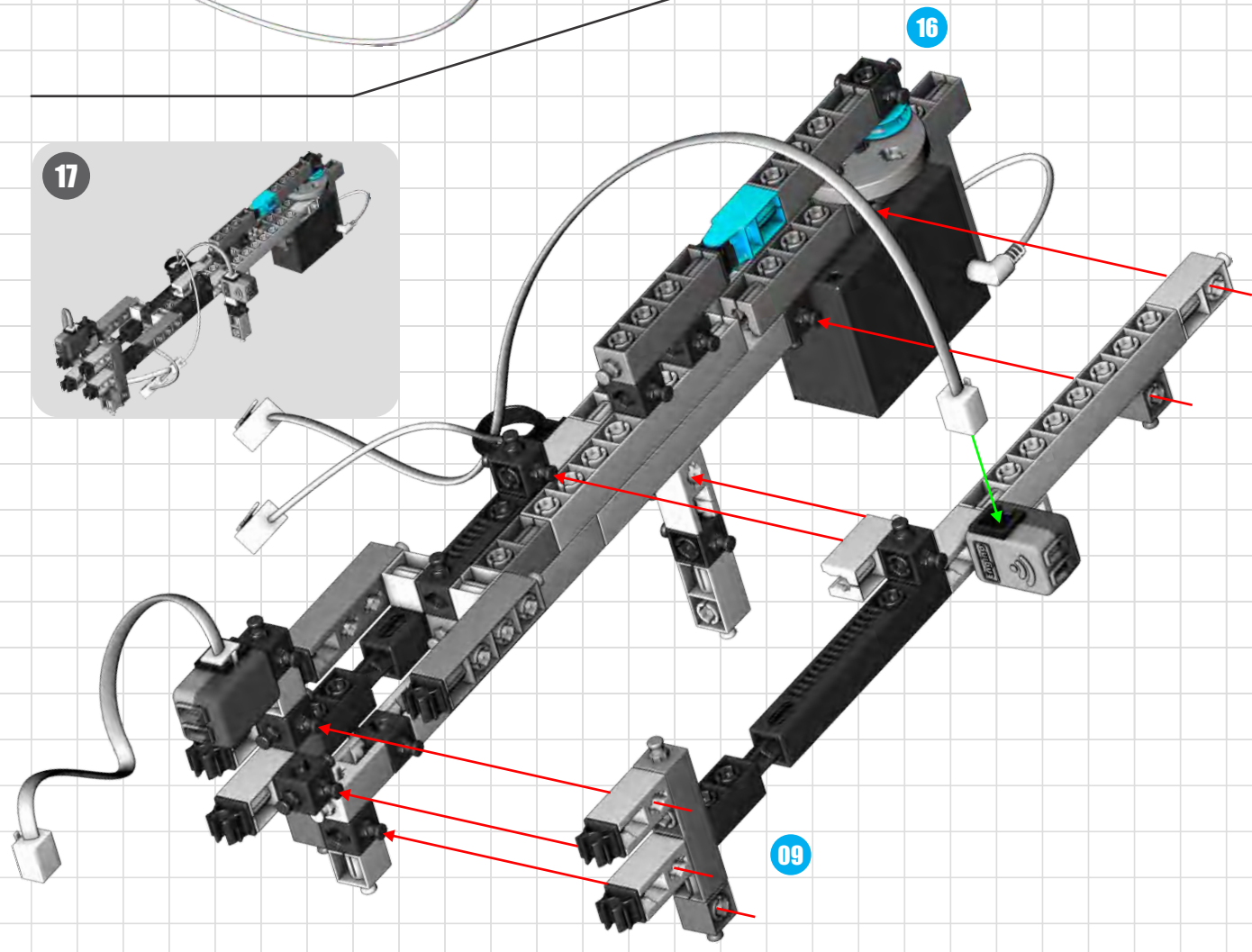
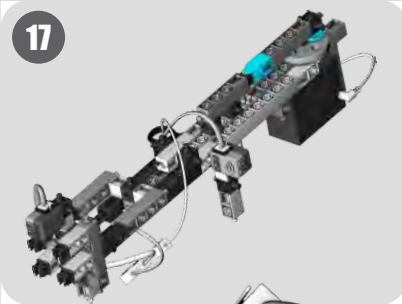


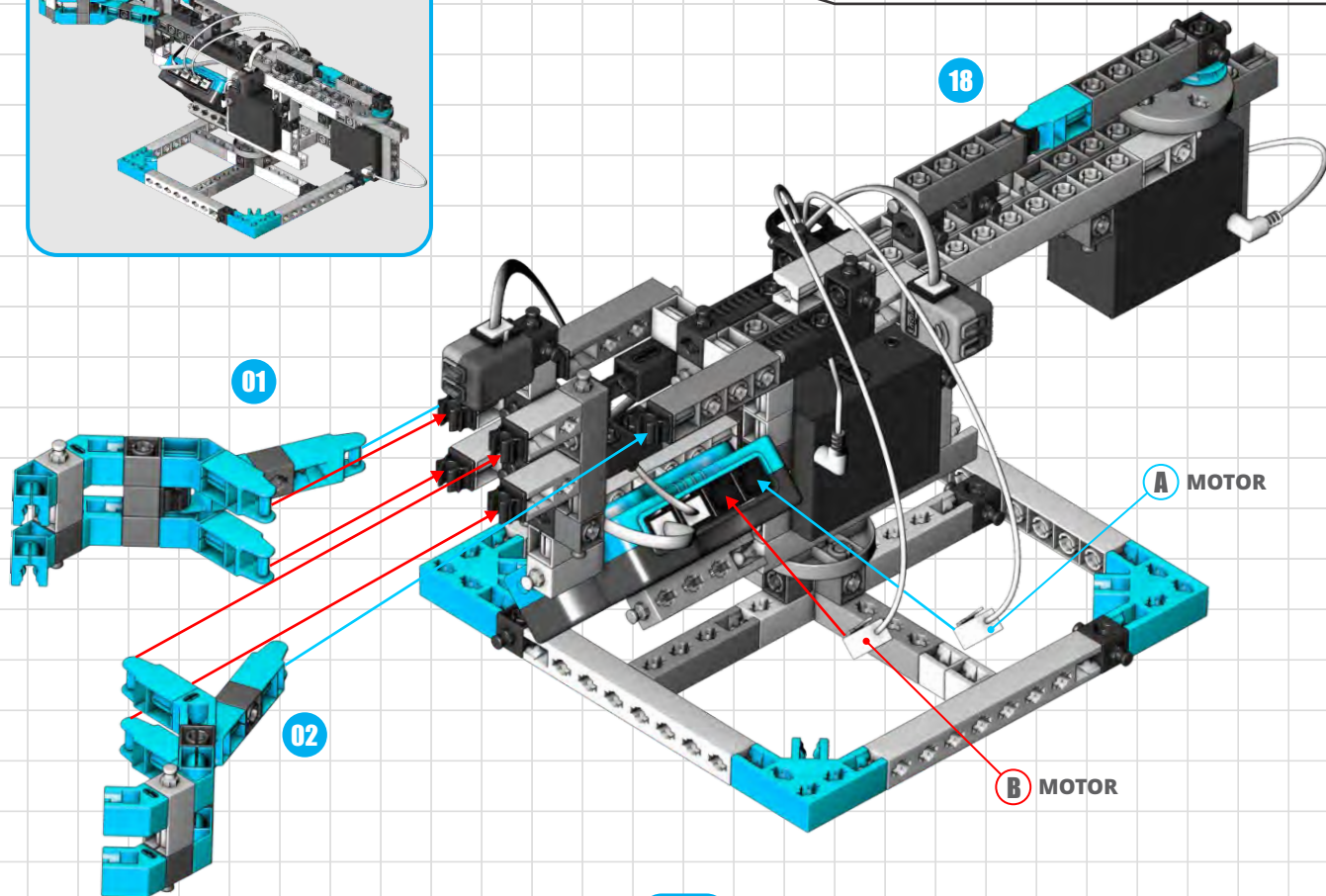
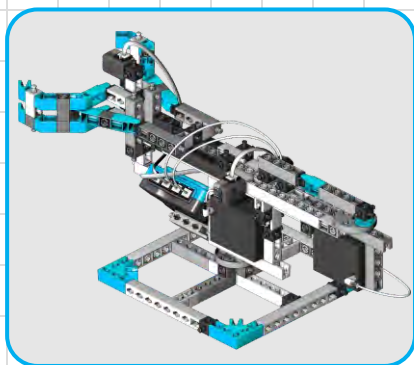
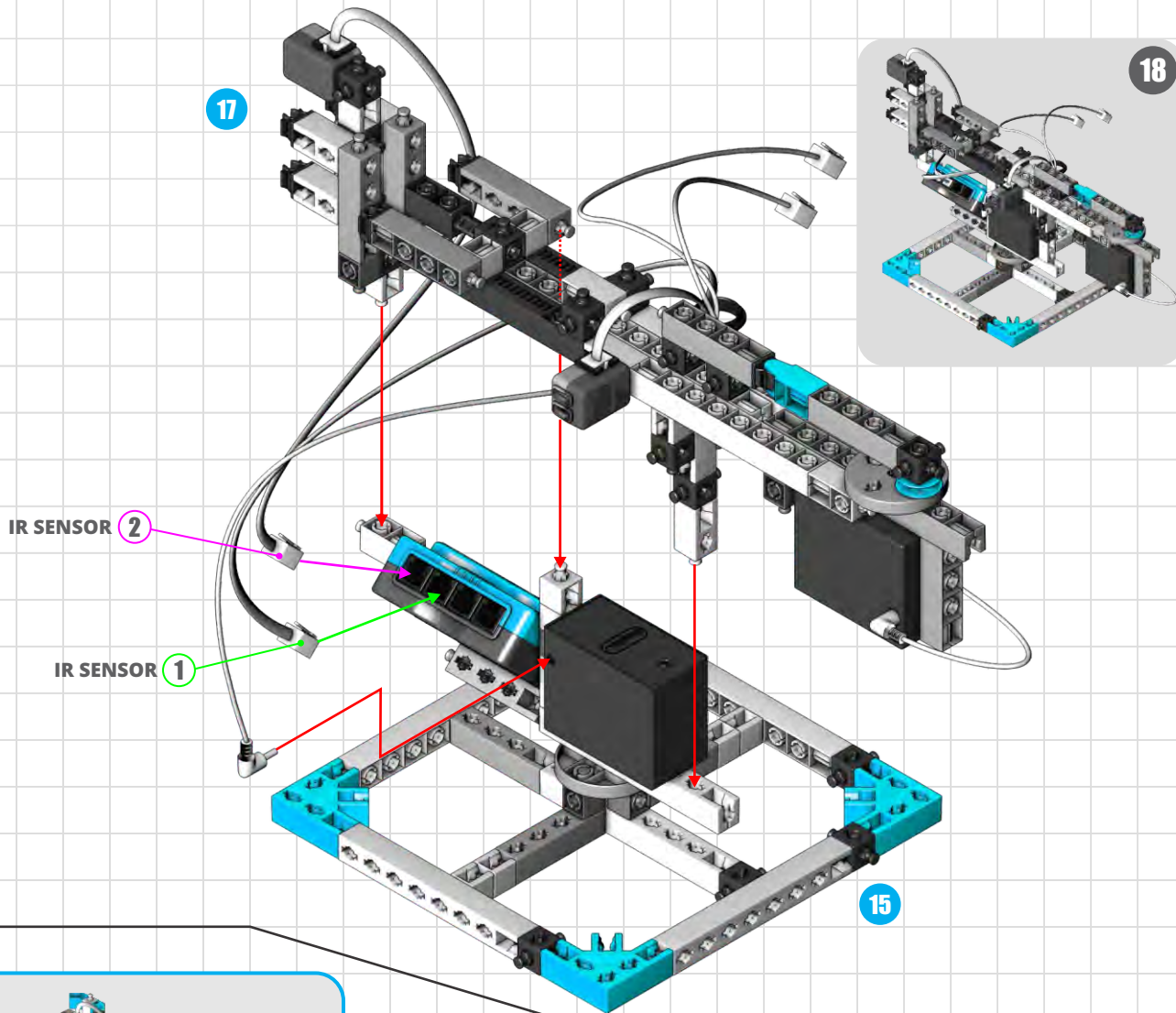


16



17

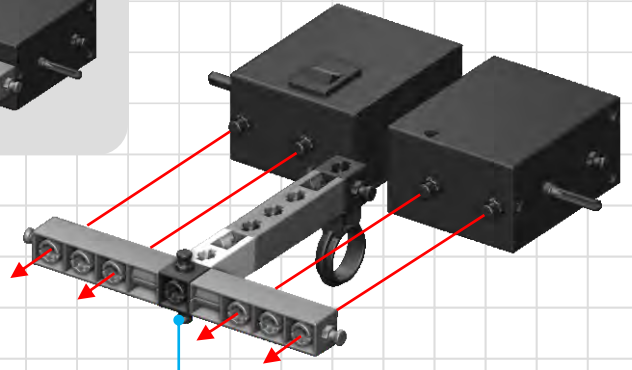




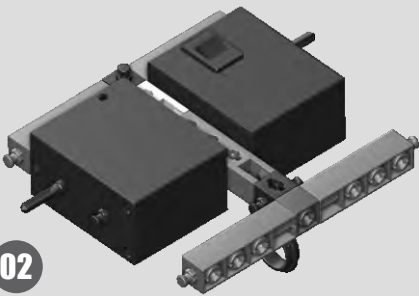
Škorpión

4025

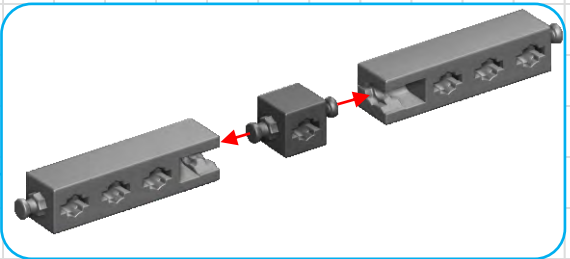
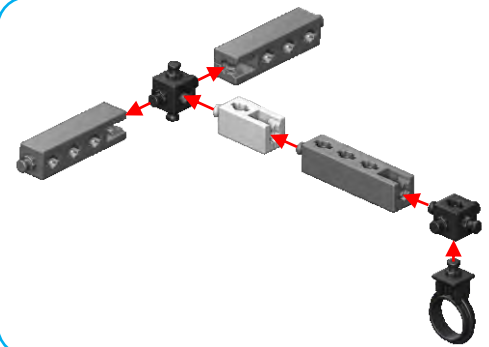
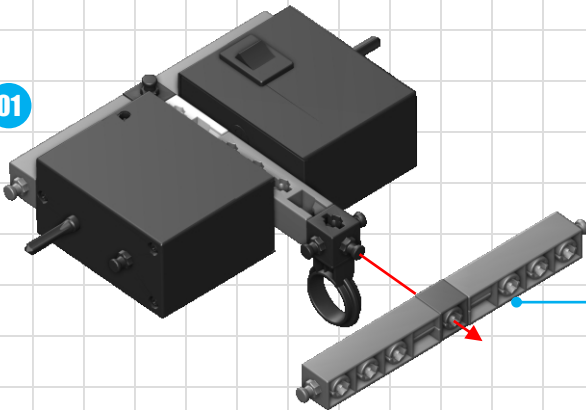
01



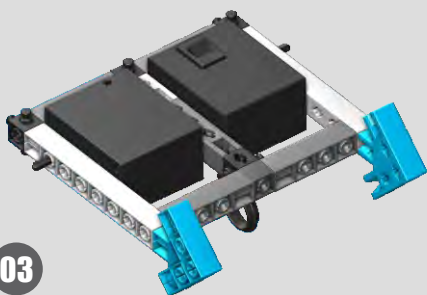
02



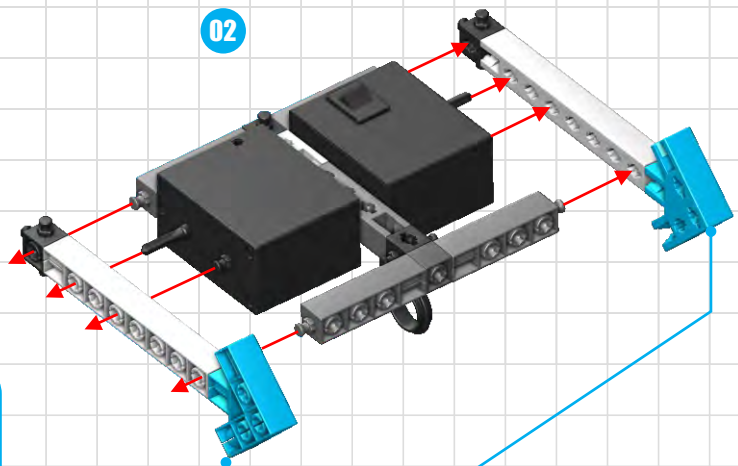
01



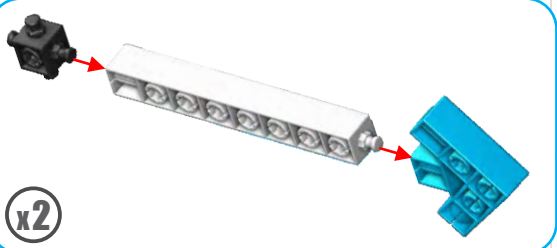
03



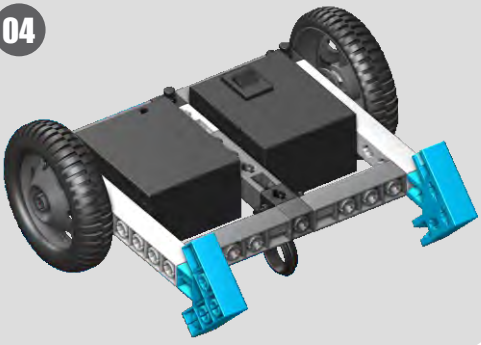
02



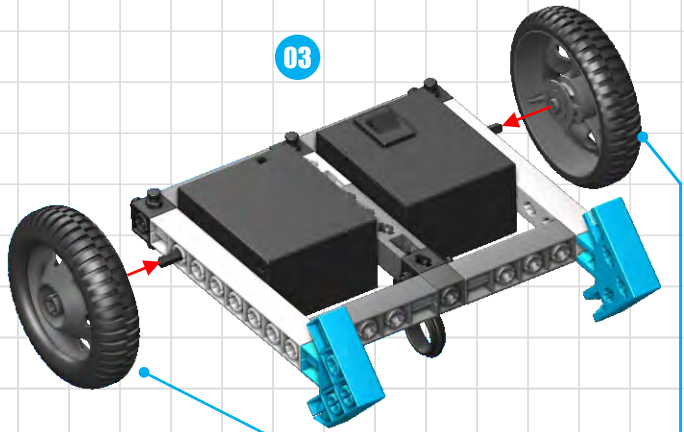
x2



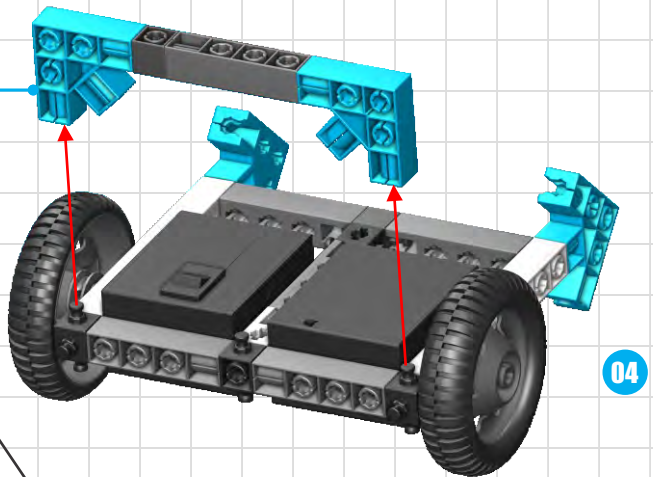
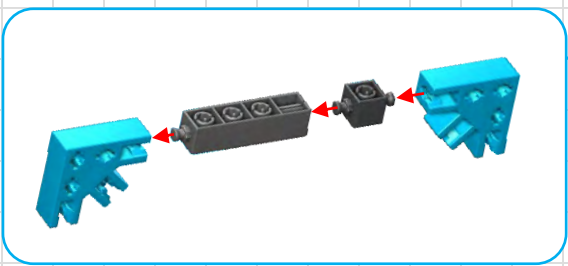
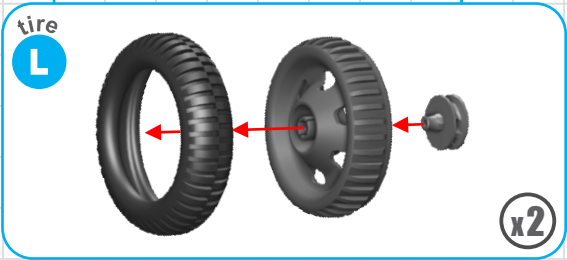
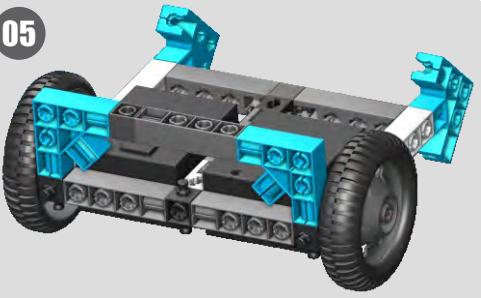
04



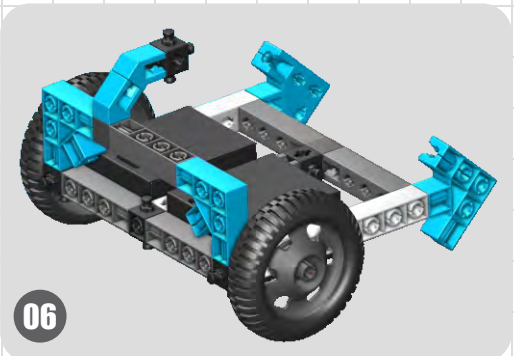
03



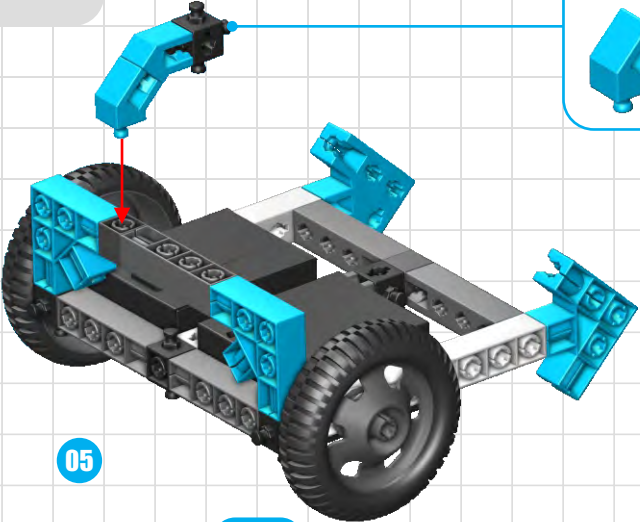
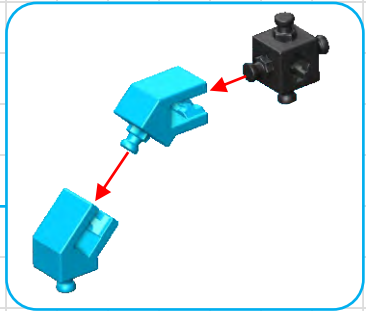
05



04

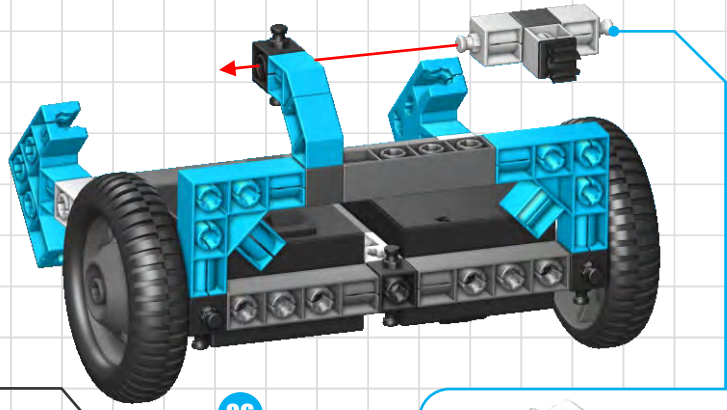
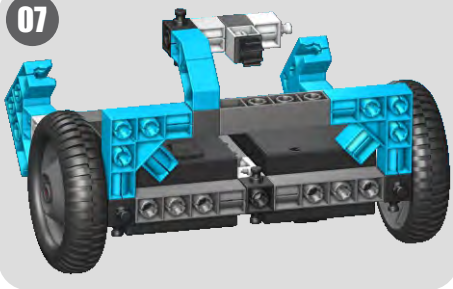


06

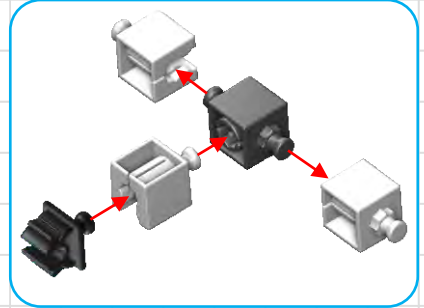


05

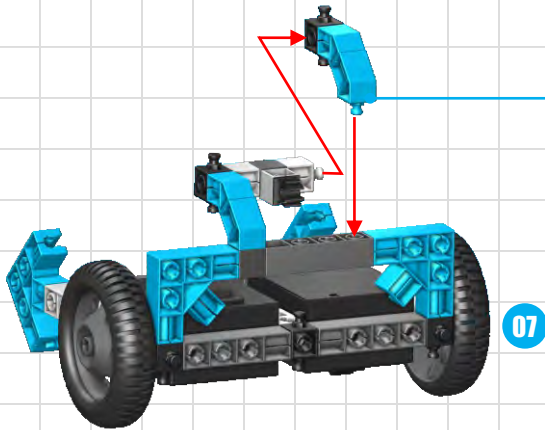
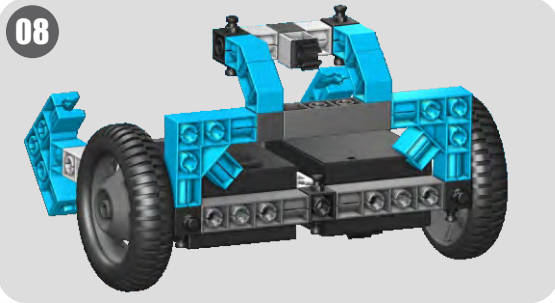
07



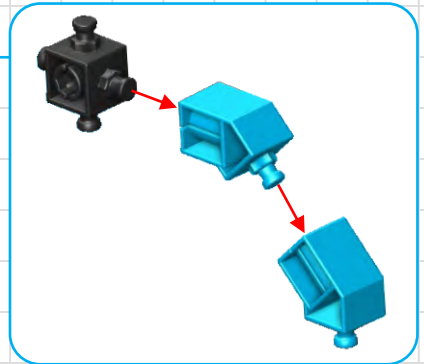
06



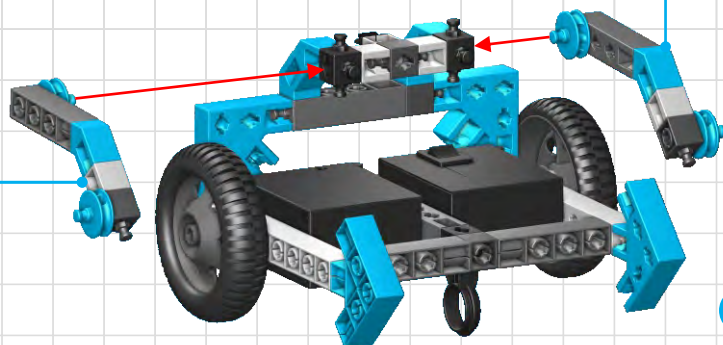
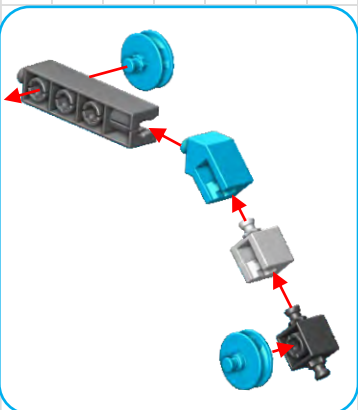
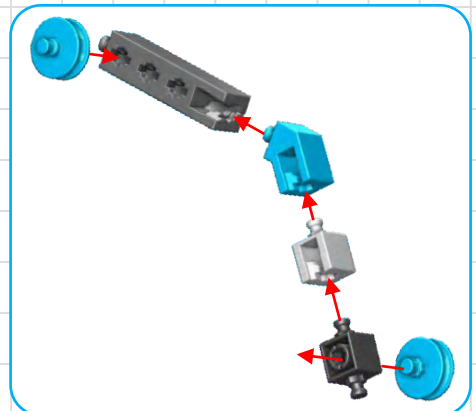
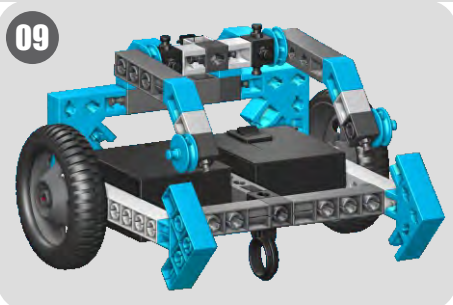
08



07

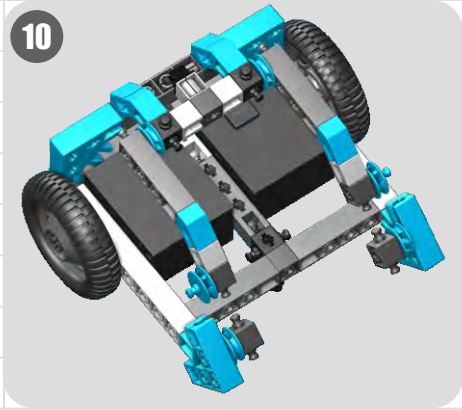


09

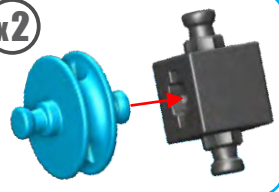


08

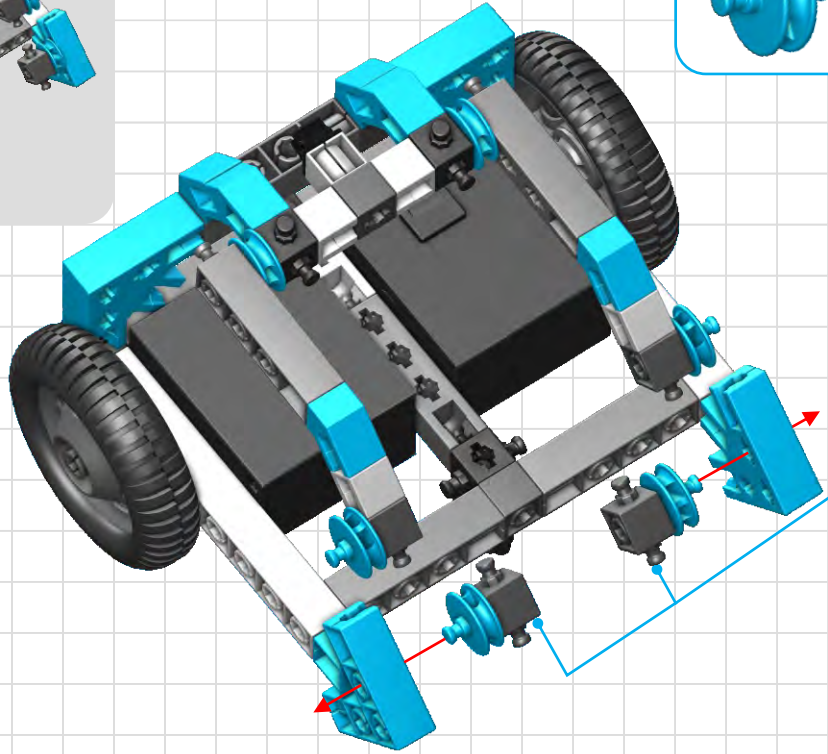
10



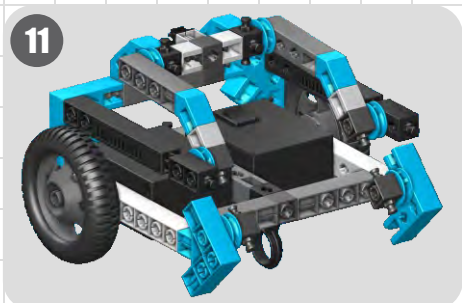
x2



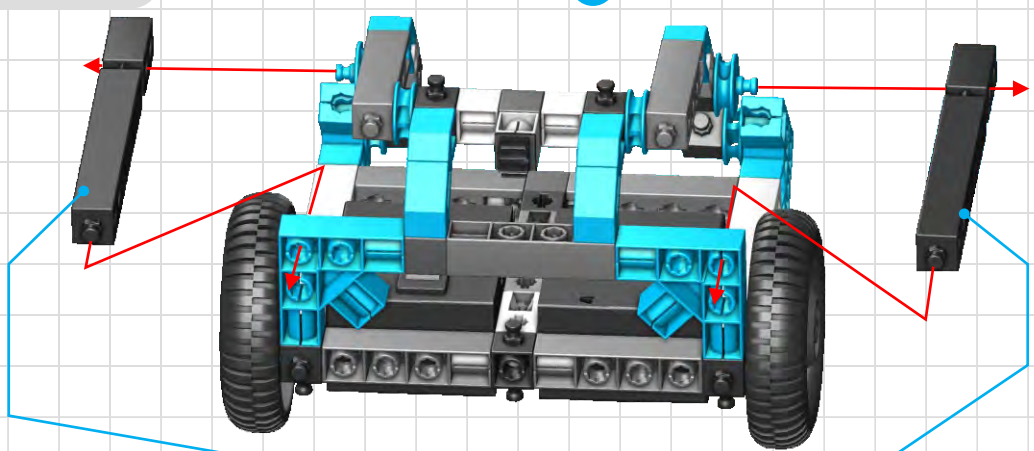
09



11

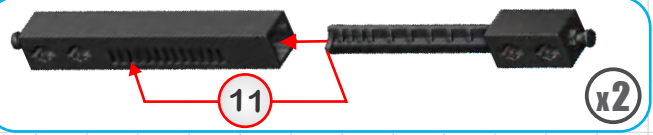


10

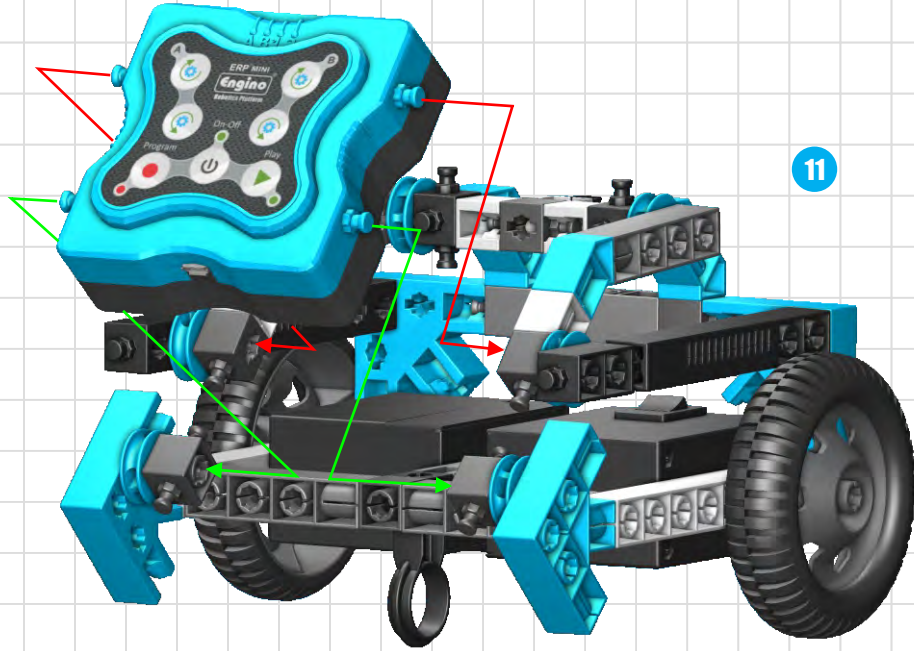


11

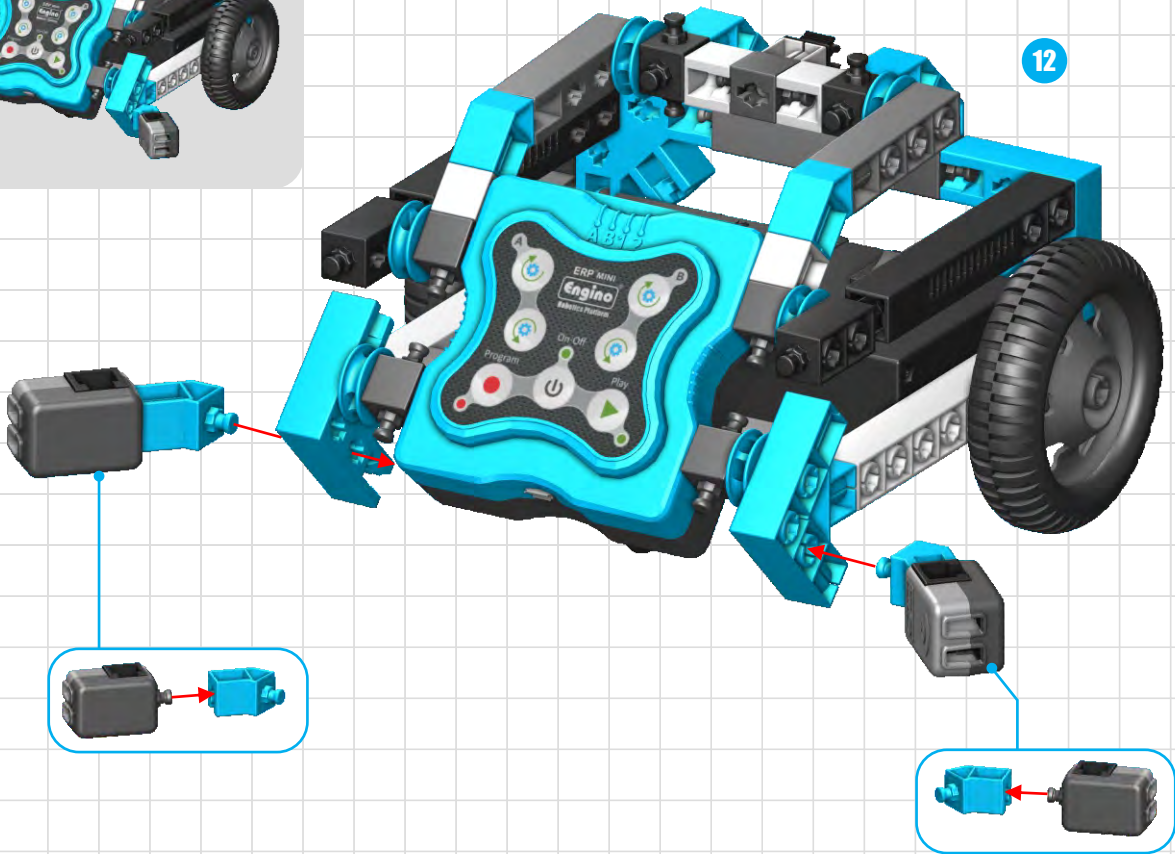
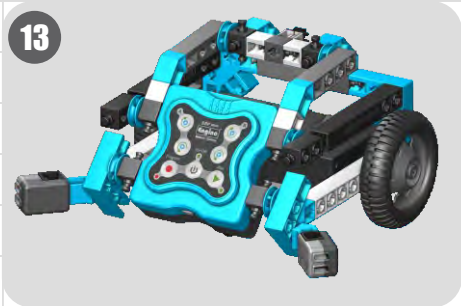
x2



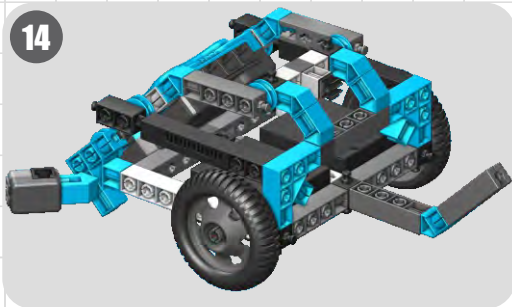
12



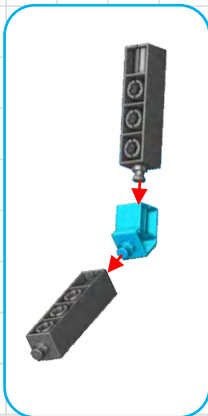
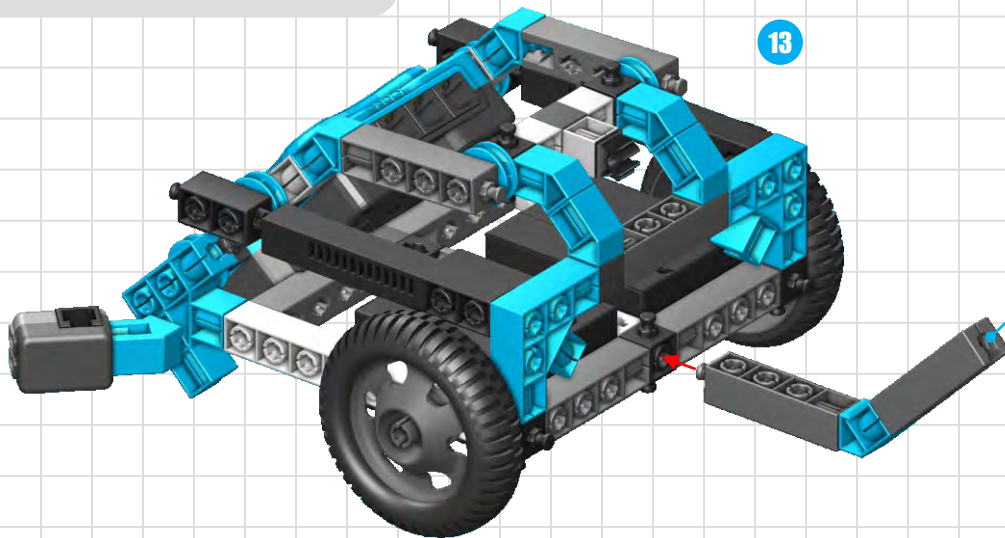
13



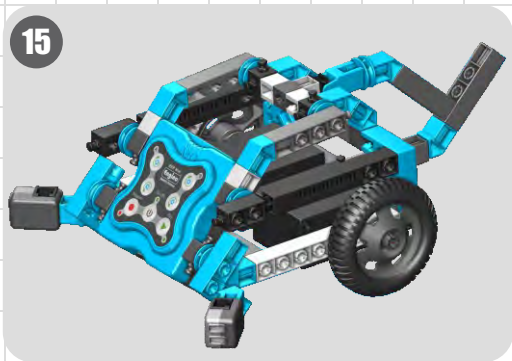
14



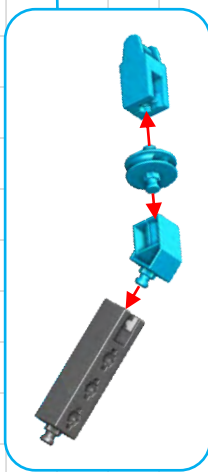
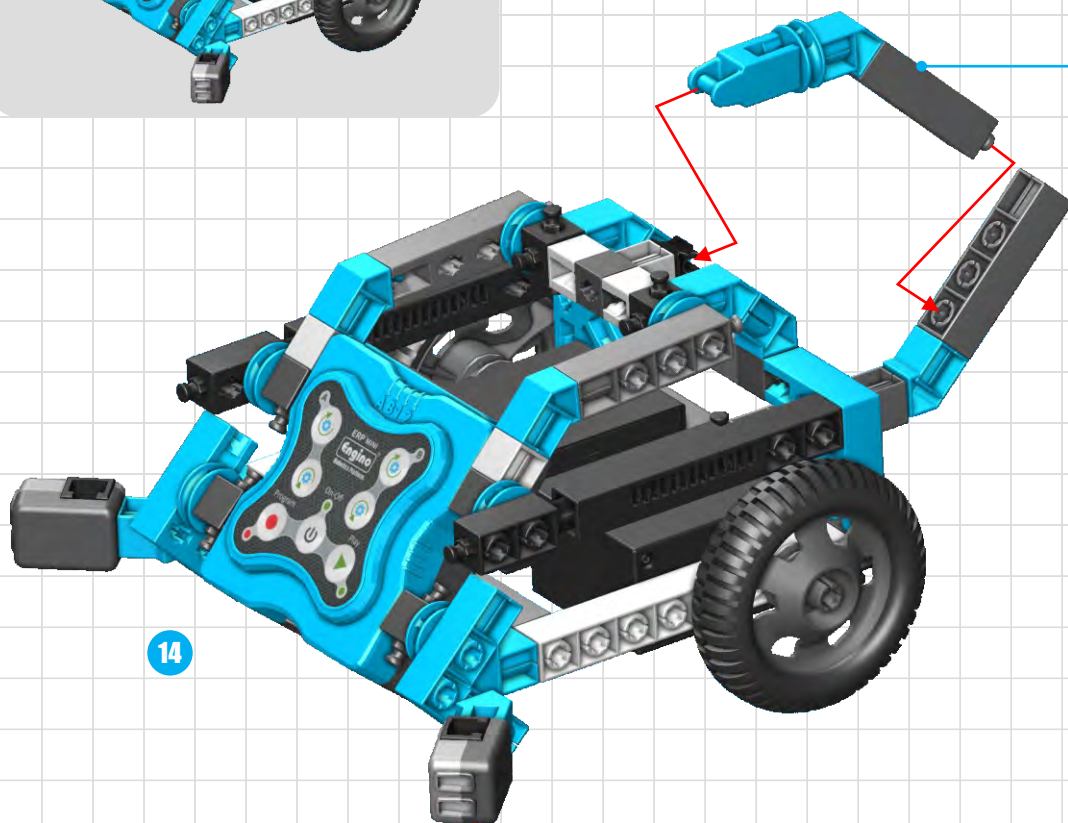
13



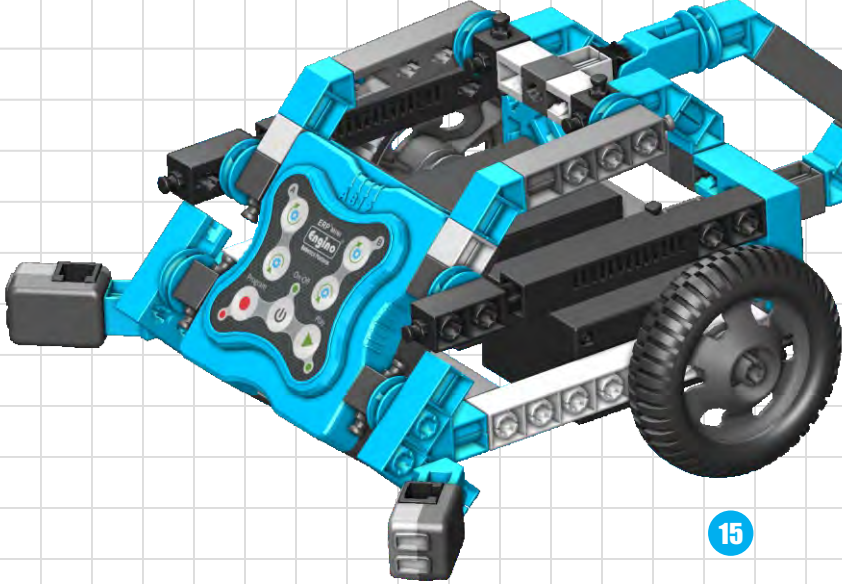
15



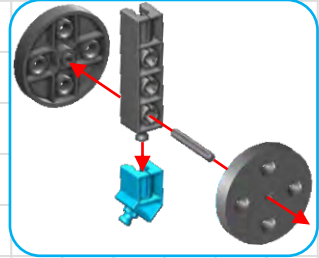
14



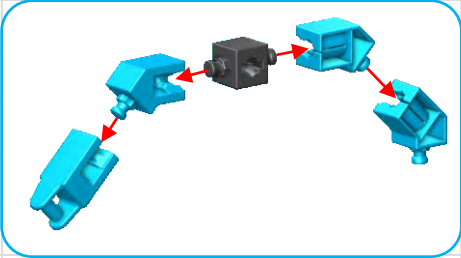
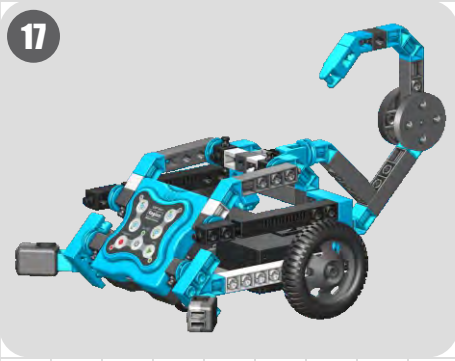
16



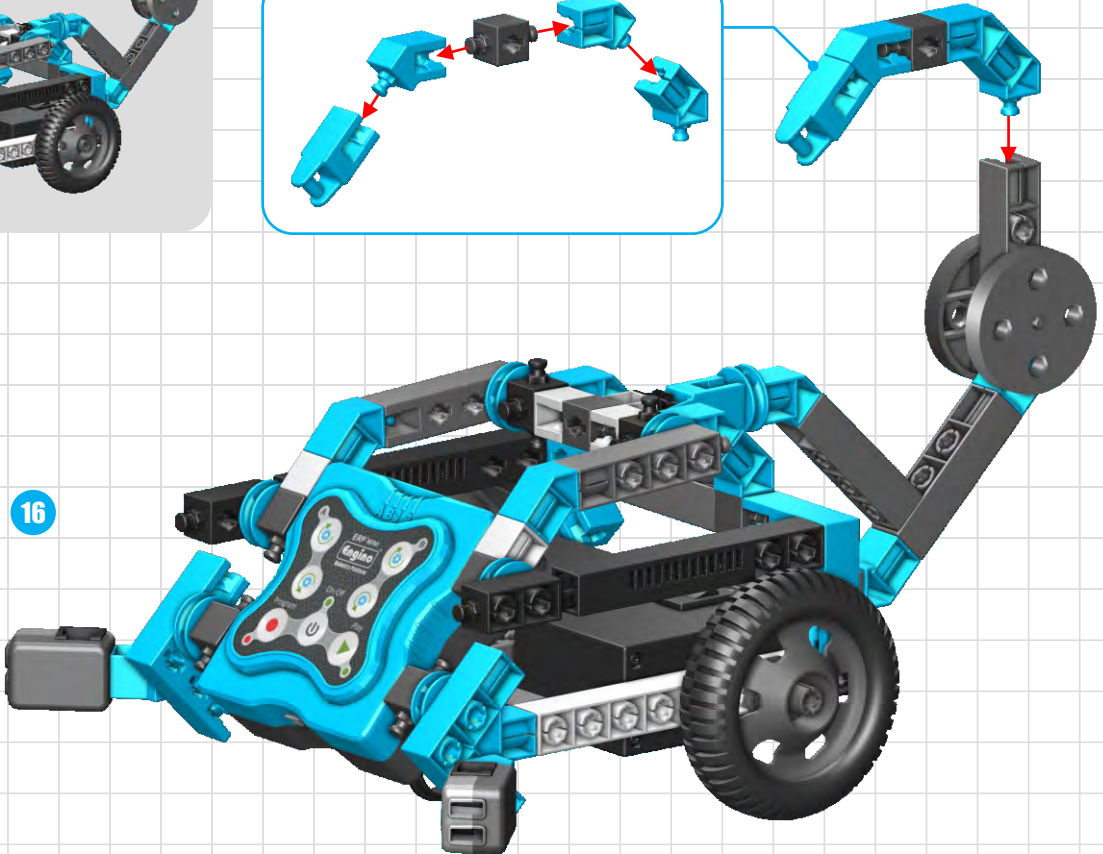
15



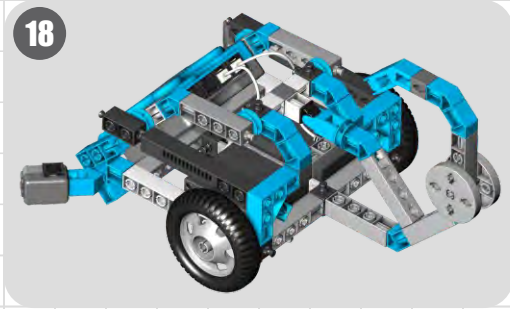
17



16

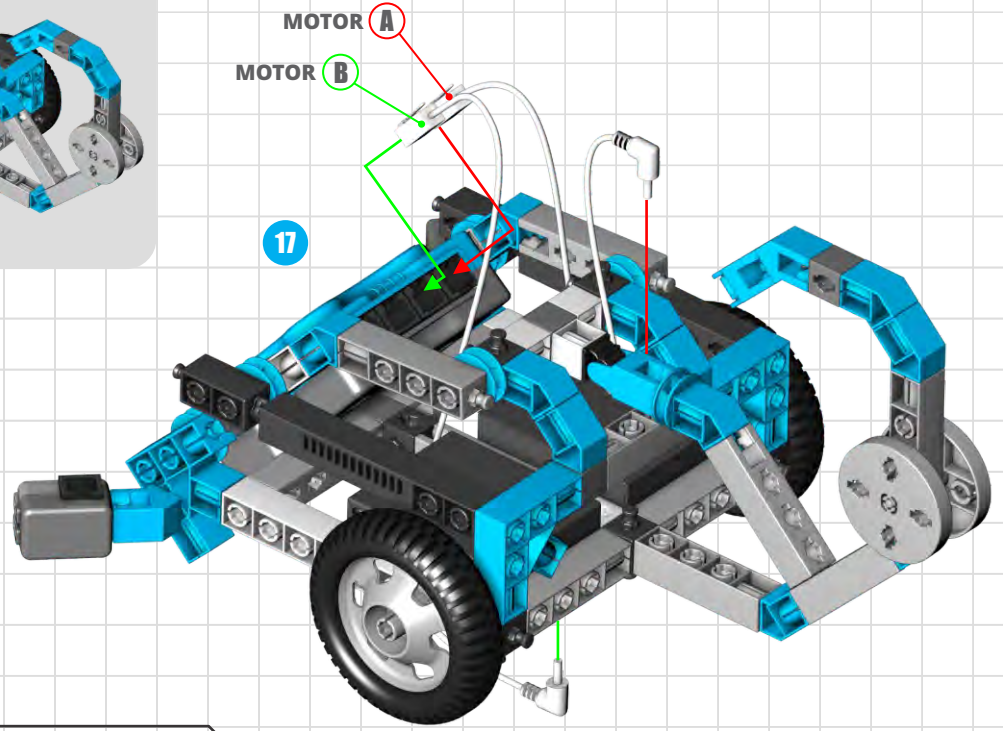


18



MOTOR A
MOTOR B

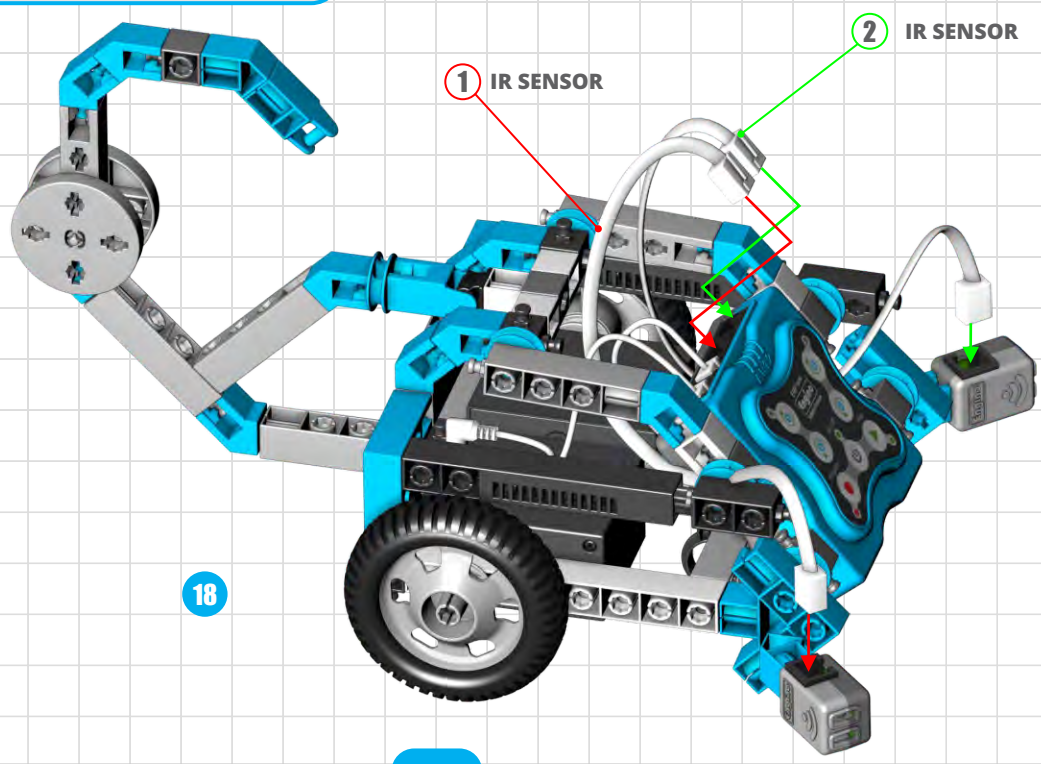
17



2 IR SENSOR

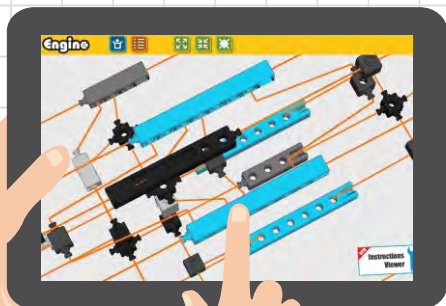
1 IR SENSOR

18



Bezplatná aplikácia

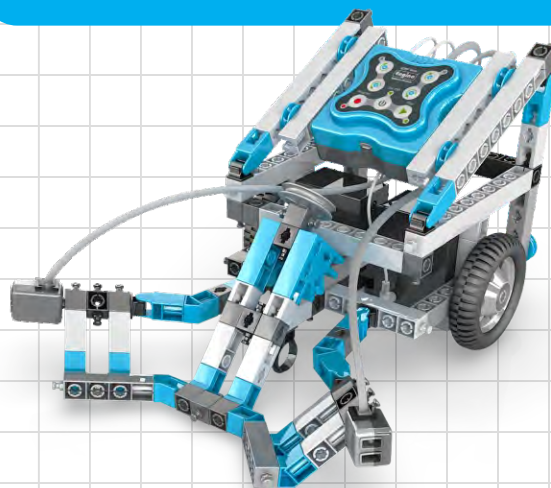
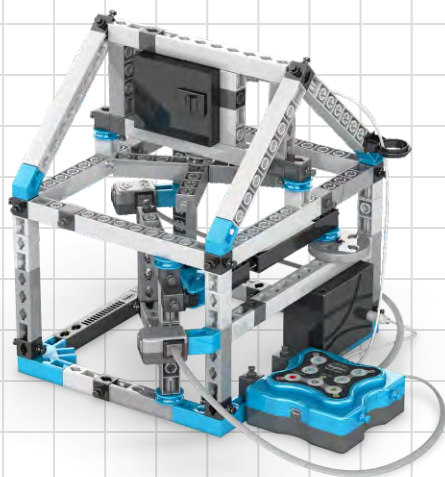
- A** Viac modelov nájdete so svojim počítačom alebo tabletom na tomto odkaze:



www.Engino.com/instructions/ce101mp-a

- B** Stiahnite si aplikáciu, kde nájdete 3D návody s popisom krok za krokom!

Engin kidCAD (3D Viewer) aplikácie:



Viac teórie a experimentov nájdete online!

www.Engino.com/theory/ce101mp-a

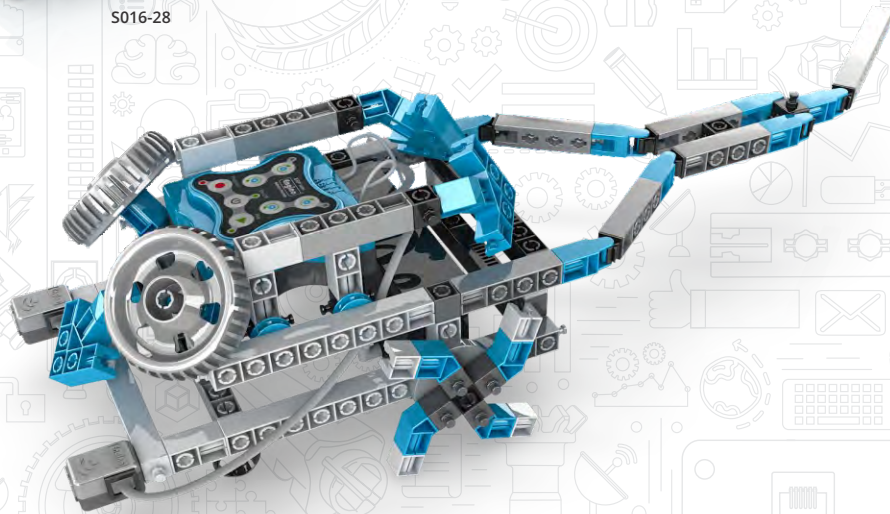
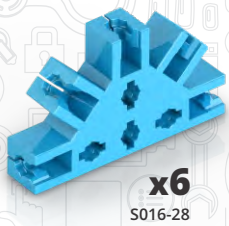
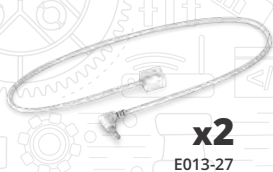
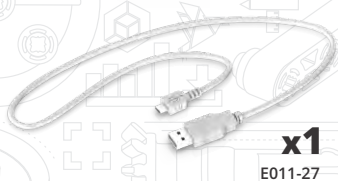
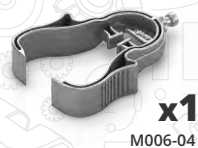


©123RF.com / baloncici, ella, James Steidl, Ales Zvolanek, Olga Serdyuk, spaxia, abidal, dipressionist, Inna Jacquemin, Miroslaw Dobrzanski, scyther5, hxdzbxy, gemenacom, pixinoo, Frederick, Joerg Hackemann, fastfun23, Ronalds Stikans.

Copyright © Engino-Net Limited

Všetky práva vyhradené. Tieto stránky sú povolené len na osobné použitie. Preto je striktno zakázaná ich reprodukcia, modifikácia, ukladanie vo vyhľadávacích systémoch, retransmisia akýmkoľvek spôsobom, elektronicky, mechanicky či inak na iný účel než osobný. Vynímkou môže byť činnosť schválená editorom.

Zoznam súčiastok



ConQuest entertainment a. s.
Kolbenova 967/27d,
198 00 Praha 9,
www.conquest.sk
info@conquest.sk

VAROVANIE:

NEBEZPEČENSTVO UDUSENIA - malé časti. Nie je určené pre deti mladšie ako 3 roky.



Kód produktu:
CE101MP-A