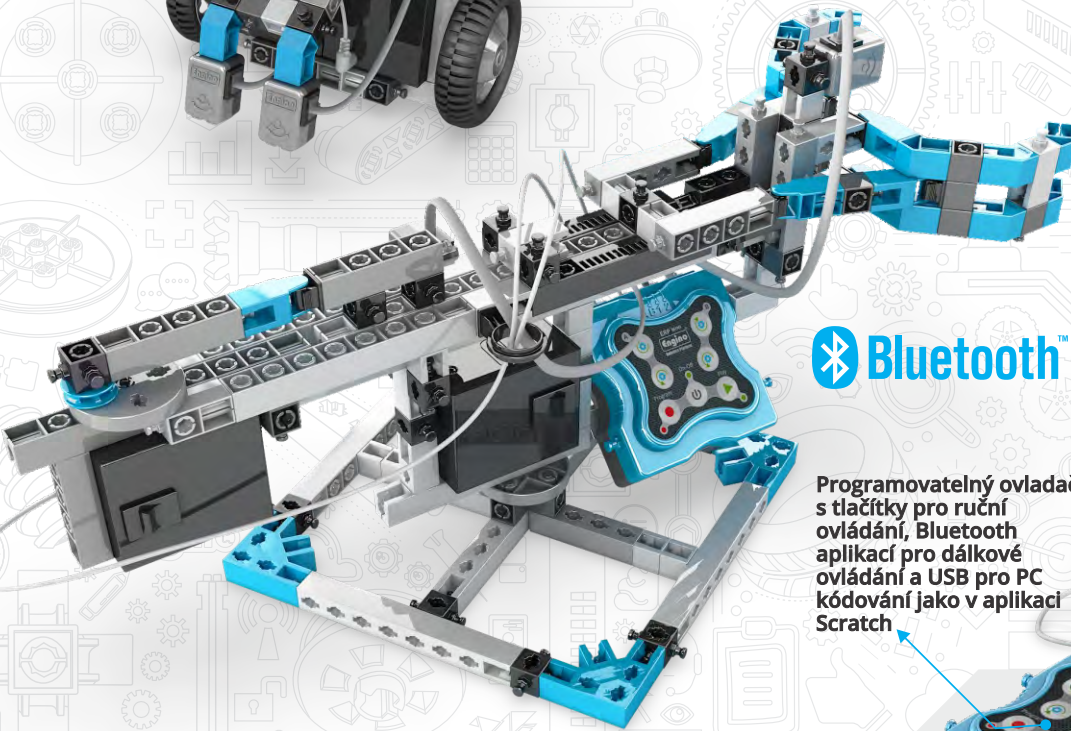
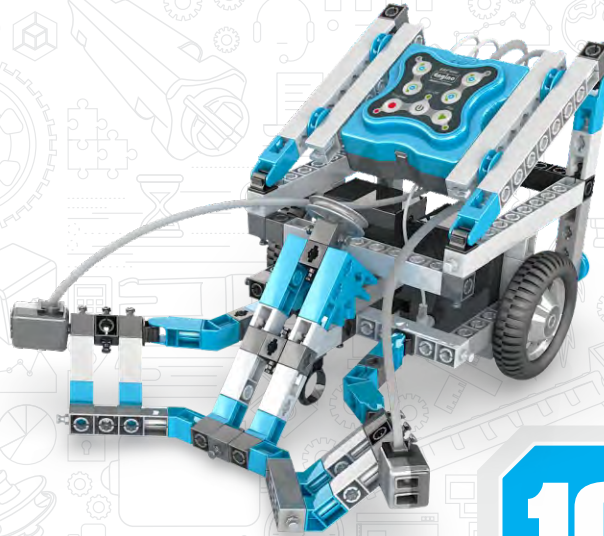


# Engine

# CREATIVE™ engineering maker Pro



**100** in 1  
robotized  
MODELS

**5**  
modelů

vytiskněné  
návodů  
(v balení)



**95**  
modelů

online  
3D návody



 **Bluetooth™**

Programovatelný ovladač  
s tlačítky pro ruční  
ovládání, Bluetooth  
aplikací pro dálkové  
ovládání a USB pro PC  
kódování jako v aplikaci  
Scratch

2x motory s  
vysokou otáčivostí  
a vestavěnými  
kolečky



**+** **ELEKTRONIKA  
A VÝKON MOTORU**

2x infračervené  
senzory pohybu

• INTERAKTIVNÍ 3D NÁVODY

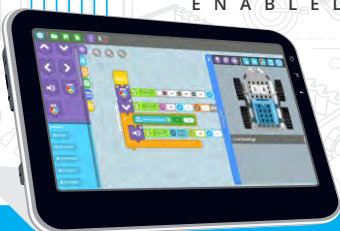
• STAVTE

• HRAJTE SI

• UČTE SE

• VYMÝŠLEJTE

**STEM**  
ENABLED



**10+**

Vhodný věk



Návody  
na stavění

**3**

Roky záruky



Rozpojovací nástroj  
Na jednoduché rozpojení  
malých součástek

# CREATIVE™ engineering

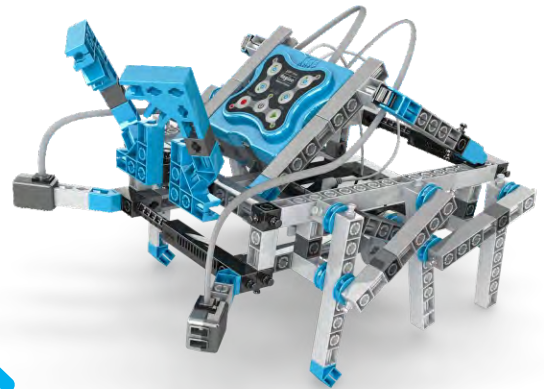
## maker Pro



### postavte si větrák

Model větráku vám představí hlavní programovací pojmy a inovativní ovladač MINI 2.0. Pomůže vám také otestovat různé proměnné a programy vytvořené pomocí softwaru KEIRO.

- Co to jsou inputy a outputy.
- Jak ručně naprogramovat robota.



### postavte si hexapoda - šestinožého robota

Experimentujte s pohybem nohou místo kol s tímto zajímavým modelem hexapoda! Nastavte na obě strany infračervené senzory, aby se spustily při blížící se překážce a robot se jí tak pomocí svých nohou vyhnul.

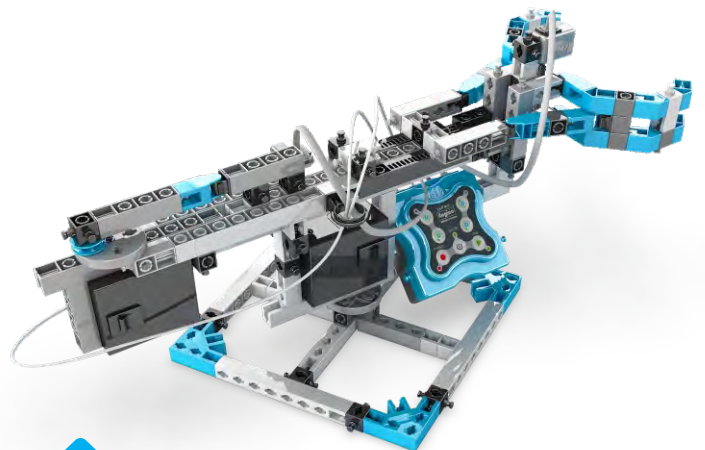
- Jak pracovat s pohybem nohou
- Co je to příkaz IF.



### postavte si robota

Postavte si model robota a využijte jeho dvojitě schopnosti! Pomocí programovatelných senzorů robot rozpozná černou a bílou barvu, čímž může jet po čáře a rozpoznat objekty ve své cestě.

- Jak fungují infračervené senzory.
- Co jsou to logické členy a příkazy WHILE



### postavte si robota twistera

Tento twister robot napodobuje opravdové robotické ruce, které se často používají v průmyslových nastaveních. Nastavte model tak, aby přesně pracoval a přesouval předměty pomocí technologie infračervených senzorů.

- Jak pracovat se stacionárními roboty.
- Co je to příkaz REPEAT UNTIL.



## Teorie

- 03** Co se naučíme
- 03** Historie robotiky
- 05** Definice robota
- 07** Využití v životě
- 09** Programování



## Uživatelský manuál

- 11** Filozofie Engino Robotics
- 13** Ovladač MINI 2.0
- 14** Periferie, kabely a porty
- 15** Příklad manuálního programování
- 16** Instalace softwaru KEIRO
- 16** Aktualizace firmwaru ovladače MINI 2.0
- 16** Rozhraní KEIRO
- 30** Programování vývojových diagramů
- 31** Ukázky kódů



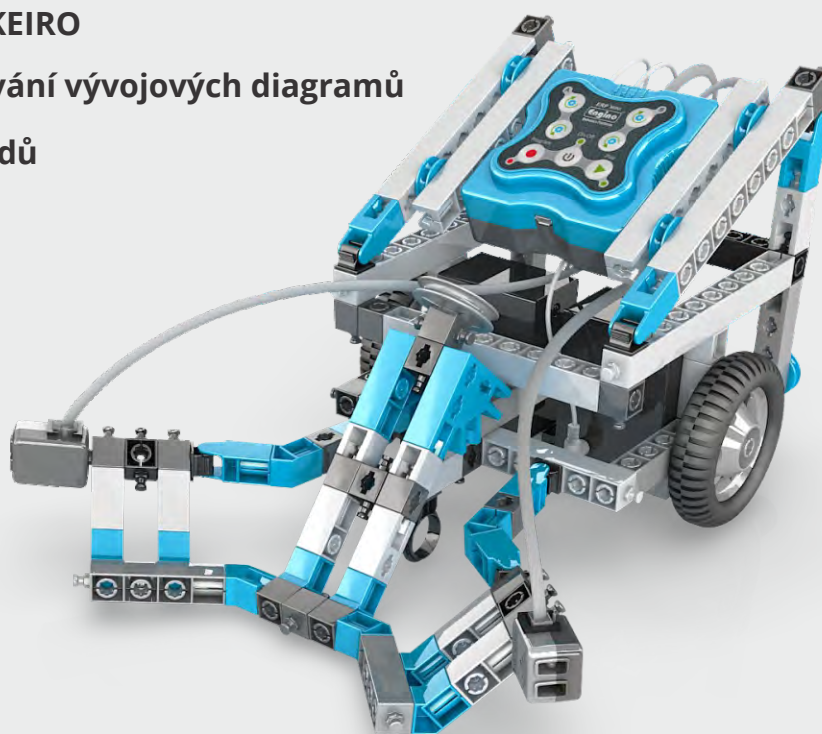
## Experimenty

- 33** Konstrukce a programování
- 34** Robotický pohyb a IF
- 35** WHILE a logické členy
- 36** Příklad z praxe: průmyslové rameno



## Návody ke stavění

- 37** Tipy ke stavění
- 38** Robot
- 43** Větrák
- 46** Hexapod
- 52** Robot Twister
- 58** Škorpión



## Co se naučíme

"Hluboko uvnitř je robotika o nás. Jedná se o disciplínu napodobování našich životů, přemýšlení, jakže to vlastně fungujeme." Tento citát od profesora matematické informatiky Roda Grupena nejlépe popisuje, jak vnímají robotiku vědci: jako simulaci životních funkcí pomocí mechaniky a počítačů. Ale dokáže kdy lidstvo vytvořit roboty, kteří by měli city a rozhodovali by se samostatně v situacích, které programovací jazyk nepředvídal? Ať už nás čeká v budoucnu cokoliv, dnešní mladí lidé musí jasně vědět, co jsou to roboti od základů robotiky a programování po vyrobení svých vlastních funkčních zařízení! Následující stránky byly navrženy přesně s touto představou!

Tato brožura obsahuje rozsáhlou **teoretickou sekci** se stavebními výzvami a zajímavými fakty, abyste se dozvěděli co nejvíce o robotech a jejich využití v denním životě. Brožura obsahuje také **Uživatelský manuál**, který do detailu vysvětluje všechny aspekty inovativního robotického systému Engino. Objevte všechny vědecké zásady, a to **experimentováním**, návody krok za krokem a fascinujícími úkoly. Řiďte se **návody** k postavení zajímavých robotických modelů, jako je například **škorpión, améba, automatizovaný dům, větrák, hexapod, ukazovátko, lopata, twister, kostka a robot**. Spoustu dalších vzdělávacích materiálů najdete online!



V hieroglyfech byla objeveny automatická zařízení

Antičtí vynálezci a řemeslníci se pokoušeli o postavení raných "robotů". Jednalo se o různé podoby od jednoduchých po komplexní zařízení, která plnila různé úkoly zdánlivě autonomně. Proto se jim říkalo automaty - z řečtiny "jednající svou vlastní vůlí", ikdyž tomu tak nebylo. Již 4 století př. n. l. údajně Archytas vyrobil parní létající stroj nazývaný "holub". Lidé té doby tvrdili, že jeho stroj uletí vzdálenost 200 metrů! Dalším vynálezcem byl Heron z Alexandrie (10-70 n. l.), který měl mnoho kreativních nápadů jako například automatické dveře od chrámu, sochy na víření vína nebo párou a větrem poháněné stroje jako jeho slavný "aeolipile".



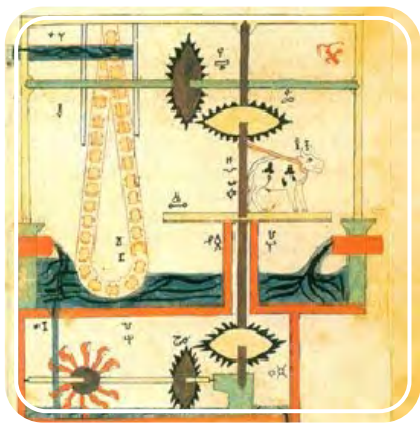
Robotika by bez počítačového programování nemohla existovat

## Historie robotiky

Historie robotiky začíná už v antických dobách. Od antiky lidé přemýšleli nad umělými konstrukcemi, které by nahradily člověka v různých úkolech se svým vlastním rozumem. Jeden řecký mýtus vypráví příběh Talose, obrovského humanoida z bronzu, kterého postavil Héfaistos (řecký bůh kovářství), aby chránil ostrov Kréta. Úvahy o tvorech podobných robotům se dochovaly po celém světě - v Norsku, na Blízkém východě, v Indii a Číně - v různých podobách: texty, kresby, malby a dokonce v egyptských hieroglyfech.



Moderní replika Heronova stroje - aeolipile



*Al-Džazářího kresba řetězového čerpadla poháněného vodou*

V roce 1898 ukázal srbsko-americký vynálezce Nikola Tesla první rádiem řízené plavidlo. Opravdoví roboti, neboli takoví, kteří byli schopni přijímání zpětné vazby od svého okolí a zároveň reakce na ni, se objevili ve 20. století. V roce 1948 vytvořil Grey Walters malé robotické želvy se světelnými a dotekovými senzory, které tak mohly hledat "jídlo". O několik let později si George Devol nechal patentovat svého robota Unimate, který zvládal vykonávat průmyslovou práci. Byl nainstalován v roce 1961 do továrny, aby zvedal horké kusy kovu z tlakového licího stroje a skládal je na sebe.



## Věděli jste, že...

Spisovatel science fiction Isaac Asimov sestavil **"tři zákony robotiky"** ve své povídce "Hra na honěnou" vydané v roce 1942. Tato pravidla zní: 1) Robot nesmí ublížit člověku nebo svou nečinností dopustit, aby bylo člověku ublíženo.; 2) Robot musí uposlechnout příkazů člověka, kromě případů, kdy jsou tyto příkazy v rozporu s prvním zákonem.; 3) Robot musí chránit sám sebe před poškozením, kromě případů, kdy je tato ochrana v rozporu s prvním, nebo druhým zákonem.



*Isaac Asimov (1920 - 1992)*

Ve středověku najdeme mnoho příkladů automatů většinou ve formě zvířat nebo lidí. Lidé stavěli zábavné stroje, které byly naprogramovány na jednoduché úkoly, jako posouvání nohou nebo otáčení hlavy pomocí mechanismů otáčivých koleček. Muslimský inženýr Al-Džazáří (1136-1206) popsal ve své knize "Kniha znalostí důmyslných mechanických zařízení" 100 automatických mechanických strojů. Později hrál ve vývoji robotiky důležitou roli francouzský vynálezce a umělec Jacques Vaucanson (1709-1782), který se snažil ukázat, jak fungují věci v přírodě. Některé jeho vynálezy ukazovaly postavy, které hrály na opravdové nástroje (flétnu a tamburínu), a slavnou "trávicí kachnu", která napodobovala trávení potravy.



*Robotické rameno Unimate*

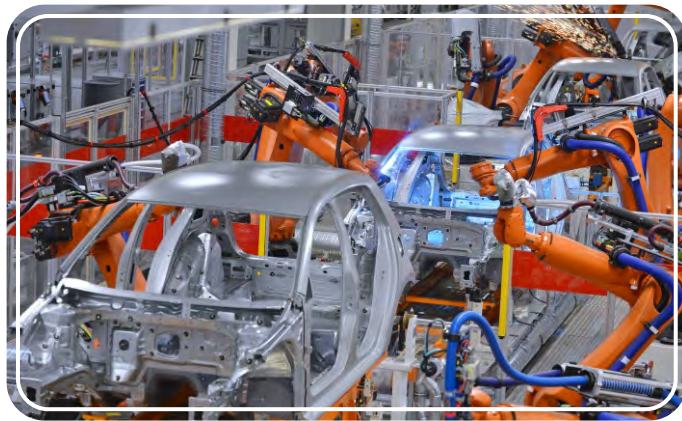
V 60. letech 20. století inženýři řešili problém robotického pohybu vytvářením robotických ramen a nohou inspirovaných přírodou. V 70. a 80. letech 20. století umožnily pokročilé technologie a malé počítačové součástky použití všech částí přímo na robota namísto jejich propojování kabely na externím počítači. Od 90. let 20. století roboti ještě více pokročili a zpracovávali další komplexní úkoly. V dnešní době se využívají v každém aspektu lidského života od domácího vybavení a asistence (jako například AIBO robotický pes, 1999 a Roomba, robotický vysavač, 2008) po průmyslové využití (například při výrobě aut) a průzkumy (jako Epson, létající robot, 2004) dokonce i ve vesmíru (průzkumní roboti na Marsu, 2004). Avšak tohle všechno je prozatím jen začátek, jelikož budoucnost robotiky vypadá velice slibně!



*Dva typy vysavačů Roomba*

## Definice robota

Pojem "robot" představil poprvé český spisovatel Karel Čapek v roce 1920 ve svém dramatu R.U.R. podle slova významu "práce" ve slovanských jazycích. To nám ukazuje, jak lidé zprvu chápali představu robotů: "stroje, které vykonávají úkoly, které jsou příliš nebezpečné, komplikované nebo jednoduše nudné pro člověka". Nicméně vědecktější definicí je, že robot je mechanický stroj, který vnímá zpětnou vazbu od svého okolí a je schopen na ni vhodně reagovat. Tím pádem roboti mají - do určité míry - opravdové autonomní chování a teoreticky přemýšlejí a chovají se podle sebe. Toto však závisí na jejich programu a přednastavených příkazech.



Robotická ramena využívaná při skládání aut



Robotické dveře s dotykovým senzorem v metru

Například automaticky otvírající se dveře (v obchodech, hotelech a podobně) jsou robotické. Mají senzory pohybu, který neustále vysílají infračervený signál. Pokud je tento signál narušen někým, kdo si stoupne před neviditelný paprsek, centrální procesor vydá příkaz dveřím, aby se otevřely. Po několika vteřinách se dveře zavřou a postup se opakuje. Dalšími příklady robotů jsou například domácí systémy vytápění a světla, co se zapínají pohybem. Například auta a hračky na dálkové ovládání naopak nejsou roboti. Pouze přijímají přímé rozkazy k pohybu přes ovladač. V tomhle případě se nejedná o zpětnou vazbu, nerozhodují se sami, když jde o objekty kolem nich - např. nezastaví před zdí.

## Druhy robotů

Roboti se vyrábějí v různých podobách a tvarech a zvládají jeden nebo mnoho úkolů díky různým nastavením a podmínkám. Proto může jeden robot patřit i do více kategorií. Takže jak rozpoznáme jednoho robota od druhého? Můžeme využít kategorizace podle funkce robota. To nám ukáže užitečnost robota a zároveň jak by měl asi vypadat či jaké součástky by měl mít, aby fungoval správně. Hlavními typy dle funkce jsou:



Robotický čistič bazénů



Průmyslové robotické rameno



**Průmysloví roboti** jsou automatizovaní roboti, kteří se využívají v továrnách na nošení těžkých předmětů nebo plnění rutinních úkolů - např. svařování, malování a manipulace s materiály.

**Zábavní roboti** jsou většinou vyrobeni pro zábavu a nemají praktické využití, například robotické hračky.



Robotická hračka

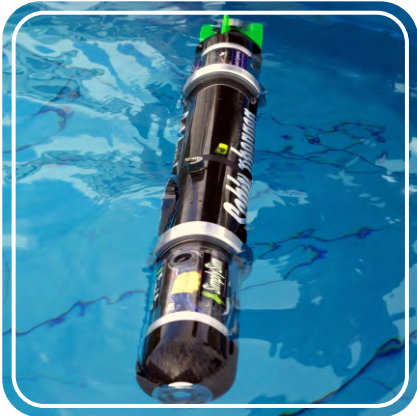


Robot na zneškodnění bomby

**Lékařští roboti** se využívají ve zdravotnictví (v nemocnicích, lékárnách, klinikách a zdravotních centrech) při tréninku doktorů nebo i při procedurách, které vyžadují přesnost (operační robot).

◀ **Vojenští roboti** jsou součástí pokročilých armád a policie. Využívají se místo lidí většinou v nebezpečných situacích pro přenos munice, zneškodnění bomb nebo zahájení válečných bitev, například malé robotí tanky.

**Domácí a servisní roboti** jsou navrženi pro komerční užívání doma nebo v práci, kde nám nějakým způsobem ulehčují život, např. robotický vysavač.



Vodní robot

◀ **Průzkumní roboti** se využívají při prozkoumávání drsných prostředí, která jsou pro lidi příliš náročná jako jeskyně, oceán (například vodní roboti) nebo jiné planety.

**Virtuální roboti** jsou roboti, kteří se využívají ve virtuálních podmínkách. ▶  
Většinou se jedná o druh s velkými brýlemi s obrazovkami, kterými si zakryjeme oči, a speciálními rukavicemi, abychom mohli koukat a fungovat ve světě virtuální reality.



Vybavení na virtuální realitu



## Věděli jste, že...

Existuje speciální fotbalový turnaj pro roboty - "RoboCup"! První roboti, kteří se zúčastnili robotického fotbalu byli vyrobeni v Japonsku v roce 1997. Hráči zvládali lokalizovat míč a kopnout do něj nohama. Výrobci těchto robotů si dali za cíl, že v roce 2050 se robotí fotbalový tým utká s vítězným týmem Mistrovství světa ve fotbale a vyhraje! Myslíte si, že do té doby dosáhnou robotické technologie takové úrovně fotbalových dovedností?



Robotí fotbalisti

**Koníčkoví roboti** jsou roboti, kteří byli vyrobeni většinou nadšenci do robotů spíše než velkými společnostmi. Jejich účelem je testování různých součástek a technologií nebo prostá rekreace jako je například létající dron.

**Soutěžní roboti** jsou vyrobeni na soutěžení s jinými roboty v různých výzvách. Tato kategorie je podobná té předešlé, protože soutěže většinou organizují fanoušci robotů a chtějí na nich otestovat robotí dovednosti či získat cenné zkušenosti. Soutěže se také konají pro vývojářské týmy z celého světa, aby se mohli zdokonalovat ve svých návrzích.



Robot létající dron

## Využití v životě

21. století je jistě dobou robotické vědy a pokročilých technologií! Roboty najdeme všude, a to nejen ty humanoidního typu, které si lidé vždy představují, ale často ve velice jednoduchých formách jako třeba systémy vytápění v domech nebo senzory vzdálenosti v autech. Využití robotických systému v každodenním životě je rozsáhlé a lidé jsou a nich více a více závislí. Přečtěte si níže o některých charakteristikách využití robotů a vytáhněte se postavením a naprogramováním navrhovaných modelů od **Engino Robotics**.



**Vyhýbání se srážkám:** moderní vozidla jsou vybavena různými senzory, aby řidiči nemuseli přemýšlet nad tolika věcmi při řízení. Detektor deště může spustit stěrače při dešti anebo detektor světla může rozpoznat, jestli je den nebo noc a zapnout tak správně světla. Co se týče bezpečnosti, tak spousta aut má senzory vepředu a vzadu senzory vzdálenosti, které varují řidiče, když se může schylovat ke srážce, hlavně při parkování nebo projíždění úzkými prostory. V situacích, kdy je u auta člověk nebo zvíře a řidič si jich nevšimne, mohou tyto senzory i zachránit život!

*Postavte si Engino model "Ukazovátko" a experimentujte s vyhýbáním se kolizím. Nasměrujte infračervené senzory směrem dolů, aby následovaly určitou cestu, nebo je otočte do stran, aby se vyhýbaly překážkám z levé či pravé strany vozidla.*

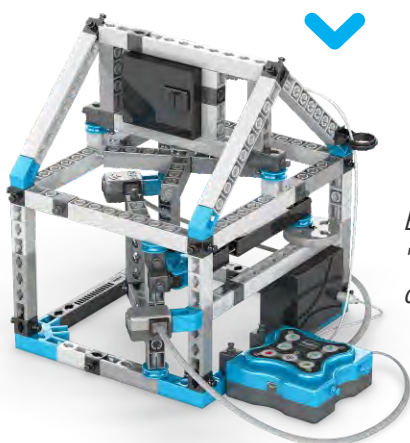


Engino® "pointer" model



**Robotické domy:** moderní domy začínají být čím dál "chytřejší" a spousta dříve manuálních funkcí se děje automaticky! Například automatické dveře se otvírají a zavírají automaticky pomocí detektorů pohybu a senzory tepla měří výkyvy v teplotě, aby mohly zapnout systém chlazení nebo vyhřívání. Inženýři experimentují s chytrými ledničkami, které by se dokázaly samy doplnit v případě, že by jim chyběly specifické produkty (mléko, vejčička, máslo), a objednaly by je online!

*Experimentujte s automatizací domu postavením Engino modelu "automatizovaný dům" (návod najdete online). Zapněte dveře a větrák pomocí infračervených senzorů.*



Engino model "automatizovaný dům"



### Věděli jste, že...

12. listopadu 2014 přistál poprvé v historii robotický modul Philae na povrchu komety! Tento obdivuhodný počín byl součástí vesmírné mise Rosetta, při které od ledna 2014 obíhala robotická vesmírná sonda kolem zkoumané komety 67P. Cílem mise bylo zjistit, zda se na kometě neskrývají nápovědy ke vzniku slunečné soustavy a života na Zemi.



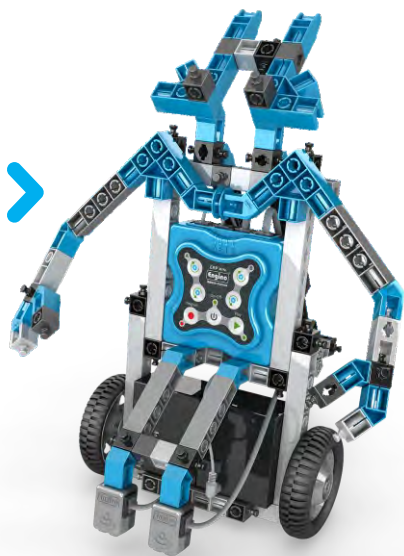
Reprezentace vesmírné mise Rosetta



## Stavění Výzva

**Nebezpečné mise:** roboti úspěšně nahrazují lidi v nebezpečných situacích. Lidé se tak nemusí vystavovat riziku a mise bývají splněny rychleji a lépe díky pokročilým robotickým schopnostem. Robot například rozpozná nebezpečné materiály, které může s lidskou pomocí na dálku zneškodnit. Operátor využívá ke sledování a manipulaci s bombou senzory robota.

*Postavte si Engino model "Robot" (stránky 38-42) a umístěte infračervené senzory dolů, aby následovaly určitou cestu, nebo dopředu, aby rozpoznávaly "nebezpečné" objekty, před kterými budou varovat pohybem.*



Engino model "Robot"

## Průzkum vesmíru

Lidstvo vždy fascinovalo divy vesmíru a možnost prozkoumávat nové světy nebo objevit mimozemský život. Nicméně podmínky mimo zemskou atmosféru mohou být pro lidi zničující: sluneční bouře, nedostatek kyslíku, gravitační pole a teploty na absolutní nule (-273 stupňů Celsia) jsou jen několika z možných nebezpečí. Všechna vyžadují velice drahé vesmírné obleky a roky tréninku a plánování na malý výlet kolem Země nebo na Měsíci. Co se vzdálenosti týče (například při cestě na jinou planetu) musí být průzkumníci ochotní vrátit se na Zemi i za několik desetiletí a nebo vůbec!

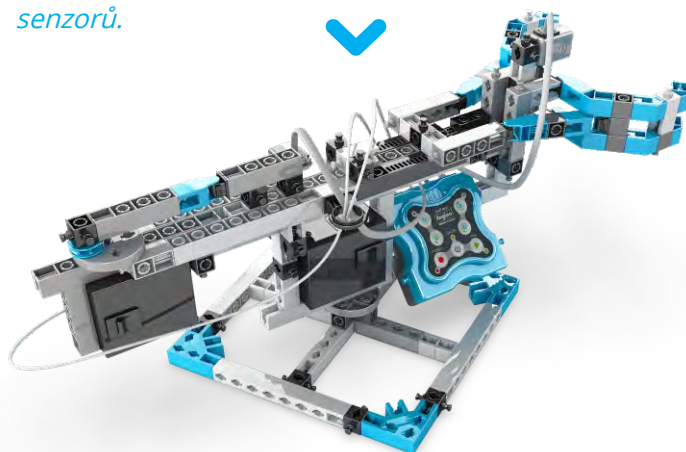


Mars Rover "Opportunity"

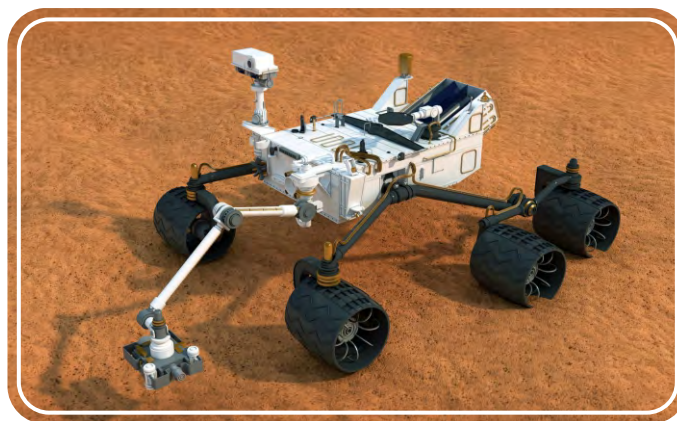
## Stavění Výzva

**Průmyslové využití:** dalším rozšiřujícím se využitím je průmyslový sektor, kde roboti nahrazují práci lidí. Jedná se většinou o stacionární nebo vedené pohyblivé roboty s ohebným ramene na zvedání těžkých předmětů nebo plněním precizní práce (například svařování). Jsou ideální do továren na montážní linky. Lidé se bojí, že automatizované postupy povedou ke snížení pracovních pozic, tomu tak ale není, jelikož vznikají nové pozice na údržbu a opravu robotických strojů a zároveň se tím zlepšuje kvalita produktů.

*Postavte si Engino "Twister" (stránky 52-57) a řiďte jeho pohyb pomocí technologie infračervených senzorů.*



Engino model "Twister"

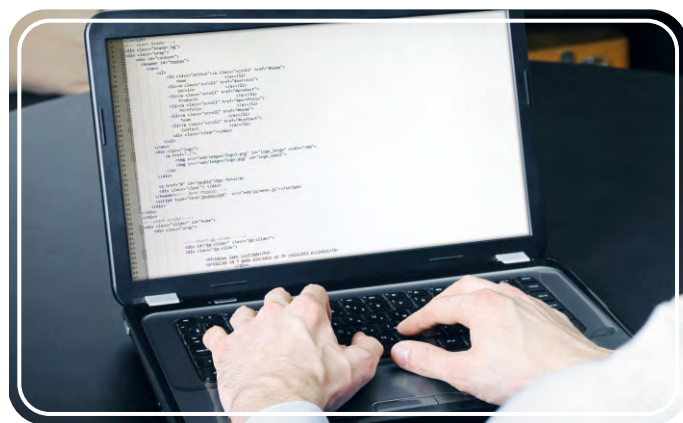


Mars Rover "Curiosity"

Naštěstí existují roboti! Ti jsou ideální na vesmírný průzkum, jelikož nemají problémy týkající se přežití a nevdají jim strávit na misi delší dobu. S jejich pokročilými schopnostmi mohou prozkoumávat povrch planety, analyzovat místní půdu a atmosféru, hledat vodu a další chemické sloučeniny a také poslat zpět na Zem spoustu cenných fotografií ve vysokém rozlišení. Momentálně jsou aktivní dva průzkumné rovery na Marsu - Opportunity a Curiosity, a mnoho dalších robotů, kteří létají kolem asteroidů, komet a planet, a reprezentují tak naše lidstvo!

# Programování

Kromě mechanických součástí je zásadní pro fungování robota také programování. Jak už jsme zmínili, zařízení je považováno za robotické, pouze pokud je schopné reagovat, komunikovat se svým okolím. Robot rozluští přijaté informace a rozhodne se pro vhodnou akci podle svého programu, pomocí mikro ovladače. Výrobci robotů často vymýšlejí své vlastní programovací jazyky, a proto existuje mnoho typů robotických softwarů. Naštěstí existuje pár základních pojmů, které bychom měli při vyvíjení programu znát, a ty jsou níže popsány.



Pracující programátor

## Programovací jazyky

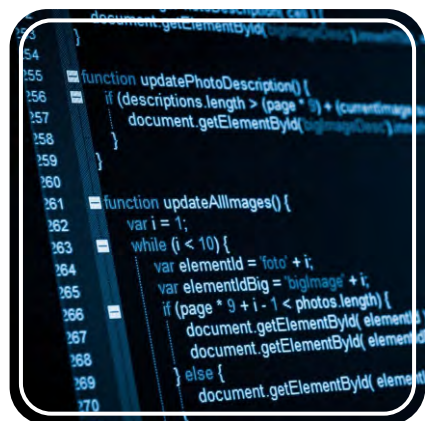
Nejdůležitějším prvkem jakéhokoliv počítače nebo robotického vývoje je programovací jazyk. Obecně je programovací jazyk takový soubor gramatických pravidel a slovníček instrukcí pro robota, aby plnil určité úkoly. Jazyk se skládá ze dvou hlavních složek: z formy (syntaxu) a významu (sémantiky) textu. Existují dva obecné druhy programovacích jazyků: nižší, které používají stroje k chápání (strojový kód s čísly a symboly) a vyšší, které obsahují lidskou řeč (v textové podobě) a automatizaci, čímž usnadňují proces programování, který je zároveň nám srozumitelný. V počítačovém softwaru, se spustitelný soubor nazývá "zdrojový kód". Ve zdrojovém kódu jsou algoritmy, které se skládají z operací popsaných krok za krokem, které tvoří jednotlivé programovací prvky, které se nazývají příkazy (vysvětleno níže).

Každý, kdo by si chtěl vytvořit počítačový program, by si měl vybrat programovací jazyk velice opatrně, jelikož jsou navrženy a používány pro různé účely. Nejběžnějšími vyššími jazyky jsou:

- C++ je nejspíše nejpoužívanější jazyk, ve kterém je vyvinuta většina aplikací.
- Java se využívá v zařízeních, ve kterých je nainstalován Java Virtual Machine (JVM) nezávisle na systému.
- Javascript je webový jazyk využívaný k provádění operací na počítačích uživatelů místo síťového serveru.
- Python je považován za nejjednodušší programovací jazyk na naučení.



Jazyk HTML je využíván konkrétně při vytváření webových stránek



Příkazy se zapisují jako řádky textu

## Příkazy

V počítačovém programování je příkaz prakticky instrukce, která říká počítači, co má dělat. Asi si dokážete představit, že tyto příkazy musí být dostatečně přesné, aby robot správně fungoval. Zároveň sekvence příkazů neboli řazení úkolů je stejně tak důležitý. Představte si, že se řídíte návodem na upečení dortu. Pokud jsou kroky receptu nesprávné nebo jsou napsány v nesprávném pořadí, pravděpodobně budete mít v kuchyni jenom bordel! V textových programovacích jazycích jsou příkazy většinou napsány řádek po řádku a obsahují speciální znaky pro různé úkoly. Toto simuluje software KEIRO, ve kterém je okno (KEIRO Code), které využívá pseudo-jazyka (nejedná se o opravdový programovací jazyk), který umožňuje uživatelům zobrazit různé příkazy podle jejich pořadí v sekvenci jako řádky textu.



*Stroj který využívá manuální programování*

Pomocí **programovacího panelu** (teach pendant), který připomíná běžný ovladač, může uživatel ovládat robota dálkově, aby dělal specifické pohyby, například zvednul součástku auta a umístil ji na pás. Při manuálním programování se tvoří sekvence příkazů a automaticky se ukládá, takže robot může pracovat stejným způsobem i nadále. Programátoři mohou kód později upravit a udělat změny týkající se rychlosti, přesnosti a plynulosti pohybu. Když je vše hotové, pošlou robotovi hotový program nazpět.



Většinou je postup k vytvoření továrního robotického programu takový, že se sepiší všechny příkazy v programovacím jazyce, kontrola zda fungují za určitých podmínek a různé úpravy dokud se nedojde k uspokojivého výsledku. Tento proces vyžaduje komplexní dovednosti v kódování a často je limitován specifickým rozsahem pohybu, například když robot přesouvá objekty z jednoho místa na druhé. Ale co když chceme napodobit přesné pohyby pracovníka, naprogramovat robota, aby dělal lidskou práci bez psaní náročných programových instrukcí? Moderní robotické technologie nám tohle přesně umožňují prostřednictvím manuálního programování.



*Pracovník používající programovací panel*



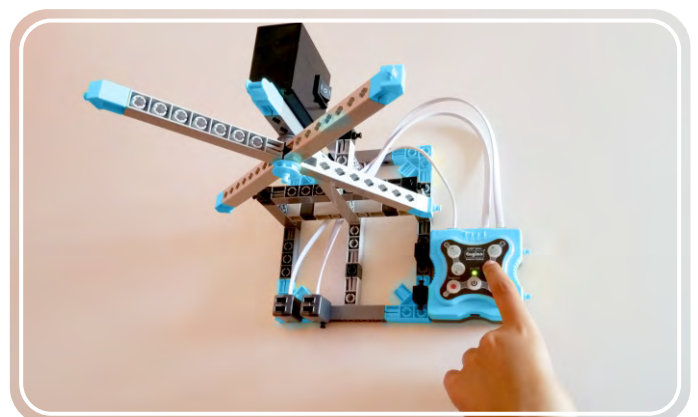
## Věděli jste, že...

Prvním počítačovým programátorem byla anglická matematicka Augusta Ada King (hraběnka Lovelace). Ada napsala v roce 1847 první algoritmus v historii, který byl určený pro stroj. Jednalo se konkrétně o algoritmus, který by vypočítal Bernoulliho čísla pomocí analytického stroje - nápad který Charles Babbage navrhl do mechanického počítače.



*Obráz Augusty Ady King (1815 - 1852)*

Manuální programování je jednou z hlavních částí Engino KEIRO softwaru. Jedná se o užitečný způsob jak vytvořit program za běžných podmínek bez jakékoliv znalosti samotného softwaru. Uživatel jednoduše mačká tlačítka na ovladači ERP MINI 2.0 a nahraje sekvenci příkazů, které bude robot v reálném čase plnit. Poté co je program uložen, může být kdykoliv manuálně přehrán dokonce i ve smyčce. Pokud by uživatel chtěl navíc upravit proměnné příkazů a přidat podmínkové příkazy, může to jednoduše udělat po připojení zařízení k počítači a zaslání programu do KEIRO softwaru. Až bude program hotový, stačí ho poslat robotovi na zpět na testování.



*Manuální programování přes ovladač MINI 2.0*



# Uživatelský manuál

## Filozofie Engino Robotics

Vývoj intelektu pomáhá studentům rozvíjet jejich nápady a zároveň získávat nové informace. To je zásadní při rozvoji a zlepšování kreativity a laboratorních dovedností. Kombinace ovladače MINI 2.0 a softwarem KEIRO, který čeká na patent a je podobný softwaru Scratch, je ideálním prostředkem k učení robotiky. Ovladač MINI 2.0 dává **čtyři propojené možnosti programování**, které mohou být zkombinovány tak, aby si každý uživatel vybral metodu odpovídající jeho věku a zkušenostem.



## Manuální programování

První úroveň programování je fyzická interakce s učebním materiálem. Mačkání tlačítek je metoda, kterou znají všechny děti. Studenti mohou nahrát jakoukoliv sekvenci příkazů přes tlačítko "Program", uložit ji na zařízení a poté zmáčknout tlačítko "Play" pro zopakování akce. Tato programovací metoda je zásadní při učení sekvence událostí a systému příkazů.

## Simulátor

Jedná se o speciální okno v softwaru KEIRO, které simuluje funkce opravdového ovladače s digitálními tlačítky místo fyzických. Po připojení ovladače MINI 2.0 k zařízení (počítači nebo tabletu) může uživatel nahrát program a vidět zpětnou vazbu od robota. Při nahrávání programu se generuje vývojový diagram a ukazuje se v podobě vizuálních bloků.



### 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
```

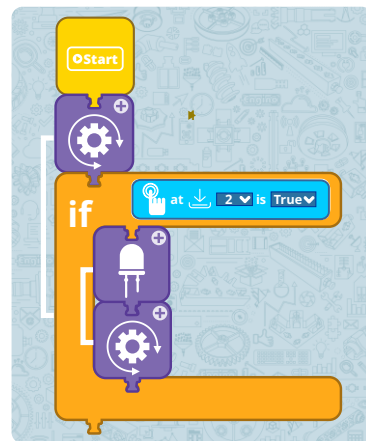
## KEIRO Code

"Pseudojazyk" (nejedná se o opravdový programovací jazyk) vytvořený přímo pro software KEIRO. Používá spoustu známých pojmů z programování jako například BEGIN, IF, END apod. Je to ideální nástroj k představení pokročilého programování, jelikož ukazuje rychlý náhled programu v textové podobě.

## Vývojový diagram

"Vývojový diagram" je hlavním programovacím prvkem softwaru KEIRO. Zde může uživatel využít všech outputů a inputů robota k vytvoření sofistikovanějších programů. Platforma je vytvořena v blokovém programovacím jazyce podobném scratchi, který umožňuje postupný přechod z manuálního programování k digitálnímu.

Programovací prostředí ve stylu drag-and-drop (přetáhni a pusť) byly prokázány jako cenné vzdělávací nástroje, které přinášejí jednoduchou cestu k interakci s reálným světem a vývoji intuitivního rozhraní lidského stroje.

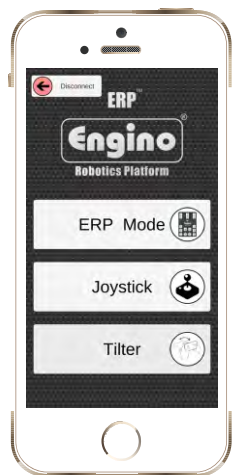


*Vývojový diagram se tvoří jednoduše přetahováním a použitím akčních bloků*

## Připojení k chytrým zařízením

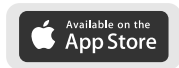


Ovladač MINI 2.0 jde s moderním technologickým pokrokem, a tak ho lze ovládat přes chytré zařízení (mobil nebo tablet) pomocí připojení bluetooth. KEIRO software je grafická programovací platforma podobná Scratchi, kterou lze ovládat přes **počítač** nebo **chytré zařízení**. Programování robota tak může být zábava a využívá se všech možností, aby byli studenti nadšení do práce na svých projektech.



Software KEIRO

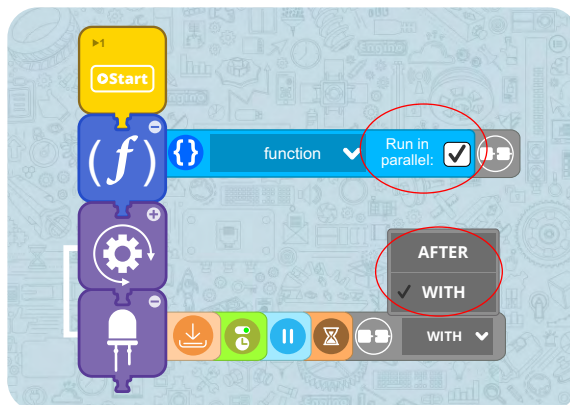
EnginoRobot BT



**EnginoRobot BT**, který je k dispozici na Google Play nebo Apple Store, prakticky simuluje rozhraní opravdového ovladače. Uživatelé mohou ovládat model, nahrávat a přehrávat programy stejně jako mačkáním fyzických tlačítek na ovladači MINI 2.0. A to vše jde na dálku a digitálně! Aplikace má dvě dodatečné metody ovládání - **Joystick** a **Tilter**. Stáhněte si bezplatně aplikaci a rozšiřte svůj programovací potenciál.

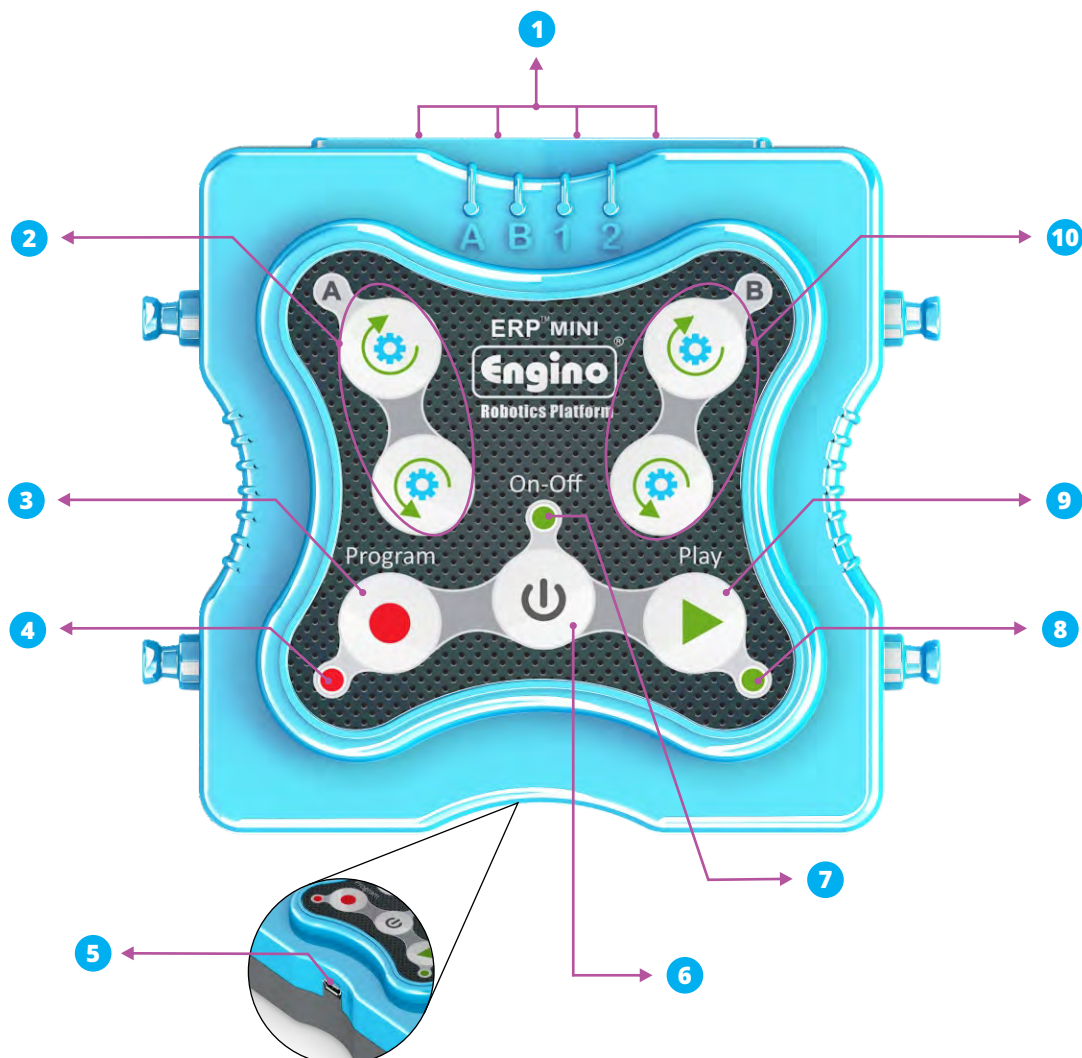
## Paralelní programování

Velkou novinkou ve vzdělávací robotice od Engino je funkce paralelního programování. Aktivní bloky v softwaru KEIRO mají šikvou vlastnost, díky které si uživatel může vybrat, zda bude příkaz splněn **po** nebo **s** předešlým příkazem. Software umí navíc nastavit funkci (x příkazů), aby fungovala **souběžně** (run in parallel) s následujícími akčními bloky kódu. Tyto možnosti snižují složitost programování a zjednodušují sofistikované úkoly.



## Ovladač MINI 2.0

Ovladač Engino MINI 2.0 je robotické zařízení skládající se z hlavního ovladače, několika tlačítek a indikátorů. Přes 4 porty zařízení lze ovladač připojit k perifériím jako jsou motory, LED, infračervené a dotykové senzory. Ovladač MINI 2.0 je silný, flexibilní a jednoduchý nástroj na používání. Tento manuál obsahuje informace o jeho hlavních funkcích, návod jak nainstalovat software KEIRO a jak naprogramujete své roboty přes příklady kódů.



**1** PORTY **A, B, 1** a **2 RJ**  
*připojují k ovladačí periférie*

**2** TLAČÍTKA **MOTOR A**  
*po nebo proti směru hodinových ručiček*

**3** TLAČÍTKO **PROGRAM**  
*nahrabe sekvenci příkazů*

**4** LED **PROGRAM**  
*bliká při nahrávání*

**5** **mini USB PORT**  
*připojuje ovladač k počítači*

**6** TLAČÍTKO **POWER**

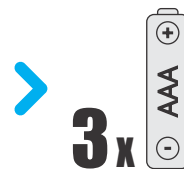
**7** LED **ON/OFF**  
*zapne se, když je zapnutý ovladač*

**8** LED **PLAY**  
*zapne se při přehrávání programu*

**9** TLAČÍTKO **PLAY**  
*přehraje uložený program*

**10** TLAČÍTKA **MOTOR B**  
*po nebo proti směru hodinových ručiček*

**Baterie:** Ovladač MINI 2.0 funguje se třemi (3) AAA bateriemi na zadní straně. K odstranění krytu baterií budete potřebovat křížový šroubovák. Umístěte baterie se správnou polaritou, symboly + a - podle jejich určení na každou pozici baterie. Po vložení baterií přišroubujte nazpět kryt.



**Bezpečnostní pokyny:** Nepřebíjejte baterie. Nemíchejte alkalické, standardní a nabíjecí baterie. Nenabíjecí baterie nenabíjejte. Pokud neplánujete vybité nebo staré baterie v blízké době použít, vyjměte je ze zařízení. Nikdy baterie nepalte. Napájecí terminály nesmí být zkratovány.

## Periferie, kabely a porty



RJ kabelový port

### LED (červené)

K dodání vizuálních efektů vašemu robotovi použijte **LED světlo**.



RJ kabelový port

### Dotykový senzor

Stisknutím tohoto **tlačítka** lze spustit **akci** nebo pohyb.



RJ kabelový port

### Infračervený senzor

Infračervený senzor lze využít k **rozpoznání** předmětů nebo **následování čáry!**



Vypínač motoru

### Stejnoseměrný motor

Stejnoseměrný motor s **vysokou rychlostí otáčení** umožňuje měnit rychlost otáčení **koleček** nebo **rozpohybovat** vašeho robota.



### Senzorový kabel

Kabel, který propojuje senzory a LED světla s ovladačem MINI 2.0. Na obou stranách je RJ konektor.



### Motorový kabel

Kabel, který spojuje stejnosměrný motor a ovladač MINI 2.0., kdy na jednom konci je RJ a na druhém je jack konektor.



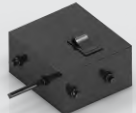



### Kabel Mini USB

Kabel, který spojuje ovladač MINI 2.0 a počítač.

V pozici I funguje motor **v souladu** se směrem ovladače a příkazových bloků softwaru KEIRO. V pozici II funguje v obráceném směru. S vypínačem v pozici **O** je motor vypnutý (OFF).



**Periferie a porty:** Ovladač Engino MINI 2.0 disponuje čtyřmi (4) RJ porty k ovládání motorů, LED a senzorů. Nicméně různé typy periférií fungují pouze v některých portech ovladače. Tabulka níže ukazuje, které porty jsou použitelné pro které periferie\*.

PERIFERIE	PORTY			
	A	B	1	2
	✓	✓		
	✓	✓	✓	✓
	✓	✓	✓	✓
	✓	✓	✓	✓

**Tabulka 1.** Periferie a disponibilní porty

\*Periferie (LED a dotekový senzor) nejsou v tomto balení zahrnuty, najdete je v balení Engino Educational set STEM & Robotics MINI ([www.enginoeducation.com](http://www.enginoeducation.com)).

## Příklad manuálního programování

Ovladač MINI 2.0 umožňuje manuální ovládání a nahrávání pomocí tlačítek na ovladači. Uživatel může postupně přejít od fyzického manuálního programování k ovládání přes software pomocí inovativní **metody reverzního inženýrství!** Nahraný program na zařízení lze jednoduše nahrát do počítače nebo chytrého zařízení.

*Jakmile je motor připojen k portu A (Nastavte vypínač motoru do pozice I):*

1. Stiskněte tlačítko "Program" k zahájení nahrávání (červený indikátor začne blikat).
2. Přidržte tlačítko po směru hodinových ručiček motoru A po 3 vteřiny.
3. Přidržte tlačítko proti směru hodinových ručiček motoru A po 2 vteřiny.
4. Stiskněte znovu tlačítko "Program" k uložení programu do paměti.
5. Stiskněte jednou tlačítko "Play" pro jedno přehrání programu (zabliká zelený indikátor)




### Rady a tipy:

- Pokud budete opakovaně mačkat tlačítko Play po 3 vteřiny, bude se program přehrávat v nekonečné smyčce.
- Zařízení zvládne uložit do své paměti pouze poslední nahraný/zaslaný program.

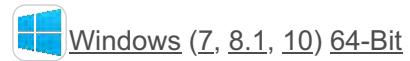
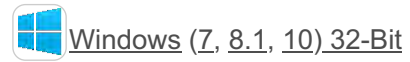


# Instalace softwaru KEIRO

Stroj, který plní sérii příkazů není opravdový robot - opravdový robot je ten, kterého lze naprogramovat k samostatnému rozhodování na základě vjemů ze senzorů. Nicméně manuální programování na senzory nefunguje. K vytvoření pokročilého programu, ve kterém lze upravovat proměnné a přidat technologii senzorů si potřebujete nainstalovat **software KEIRO**. Ten lze stáhnout z webových stránek: [www.engineerrobotics.com](http://www.engineerrobotics.com)

1. Přejděte na stránky [www.engineerrobotics.com](http://www.engineerrobotics.com). Klikněte na "**download software**". Vyberte si balíček **ERP MINI 2.0**.
2. Vyberte vhodný operační systém a klikněte na "**Download**".
3. Nainstalujte si program ze stáhnutého souboru. 
4. Po dokončení instalace spusťte program.

## ERP MINI 2.0 S BLUETOOTH



**Software KEIRO** si můžete stáhnout také na své chytré zařízení z Google Play nebo Apple Store.

## Aktualizace firmwaru ovladače MINI 2.0

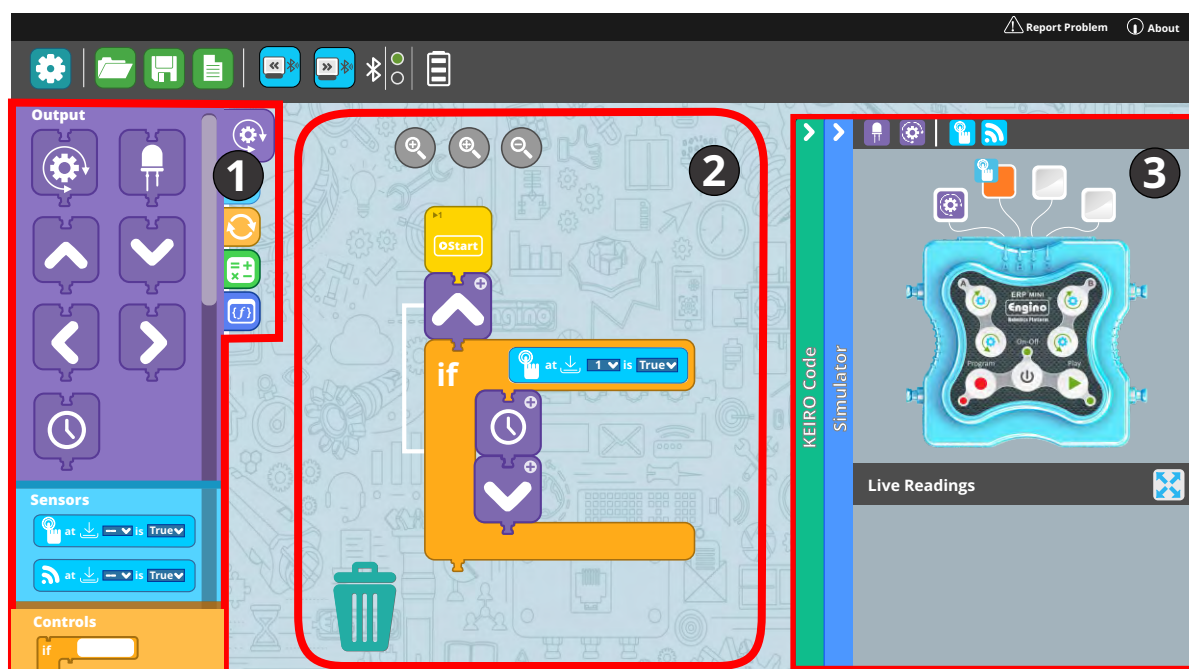
Software KEIRO vás bude informovat o nových aktualizacích firmwaru ovladače MINI 2.0 vyskakovacím upozorněním při vašem připojení.

Vyberte "YES" pro stažení nejnovějších aktualizací firmwaru.



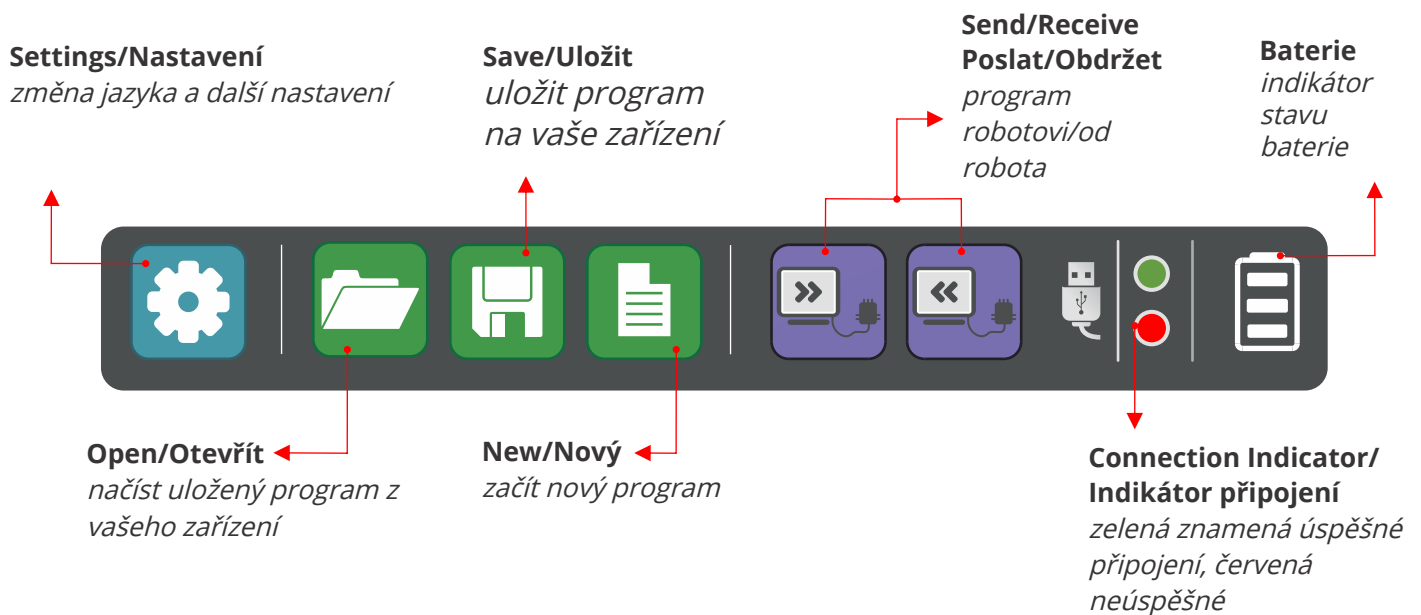
## Rozhraní KEIRO

1. **Lišta s akčními bloky** - Zde najdete všechny akční bloky, které lze využít k vytvoření sekvence příkazů. Pro ulehčení při jejich výběru jsou bloky seskupeny do kategorií.
2. **Vývojový diagram** - Do této oblasti můžete přetahovat bloky a vytvářet tak program pro robota. Úpravy programu děláte přes úpravy vlastností jednotlivých bloků.
3. **Simulátor/KEIRO Code** - Aby software věděl, ve kterém portu jsou připojeny které periferie, je důležitá konfigurace simulátoru. Code panel vám ukazuje váš program v textové formě namísto bloků. (Ke skrytí nebo zobrazení oken klikněte na lištu).



## Lišta menu

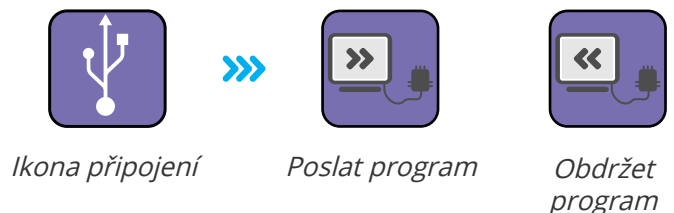
Lišta menu je umístěna zvrchu softwaru KEIRO, ve které najdete různá tlačítka pokynů jako například navázání spojení s ovladačem MINI 2.0, návrat na domovskou obrazovku, otevřít/uložit program apod.



## Připojení k počítači

Ovladač MINI 2.0 můžete **připojit k počítači** mini USB kabelem z balení a poté kliknutím na správnou ikonu v menu. Po připojení se červený indikátor na liště menu přepne na zelený, což znamená, že se vytvořilo spojení mezi počítačem a ovladačem.

Po úspěšném připojení se objeví dvě nová tlačítka na liště menu. Můžete "**Send/Poslat**" program z počítače do ovladače MINI 2.0 a naopak kliknutím na správné tlačítko.

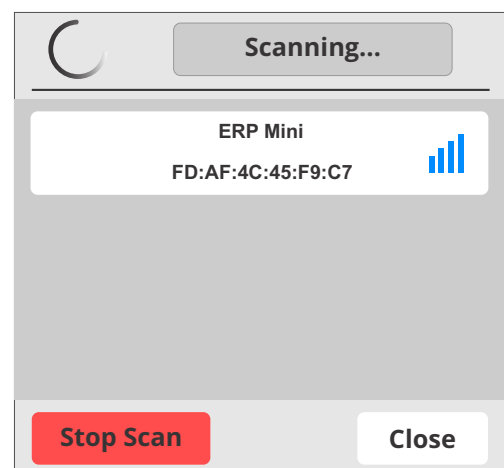


## Připojení k chytrému zařízení

V případě užívání softwaru **KEIRO** na **chytrých zařízeních** (platformy Android a iOS) se objeví na liště menu ikona bluetooth namísto USB.



K připojení ovladače MINI 2.0 zapněte bluetooth na vašem zařízení a zapněte ovladač MINI 2.0. Kliknutím na ikonu bluetooth naskenuje software dostupná zařízení. Z nového okna, které se objeví, vyberte ovladač MINI 2.0 a připojte ho ke svému tabletu.



Naskenujte zařízení bluetooth a připojte ho ke svému ERP Mini

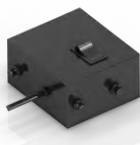
# Panel Simulátoru

## Konfigurace Simulátoru

Simulátor je speciální okno v softwaru KEIRO, které je určené k nastavování periférií robota. Před nastavením vývojového diagramu je důležité **nastavit Simulátor** umístěním periférií na stejné porty jako jsou připojeny k robotovi. Toho dosáhnete přetažením ikon periférií ke správným portům. Na horní liště simulátoru najdete ikony všech periférií.



**LED světlo**  
*Ikona bloku*



**Stejnoseměrný motor**  
*Ikona bloku*



**Dotykový senzor**  
*Ikona bloku*



**Infračervený senzor**  
*Ikona bloku*

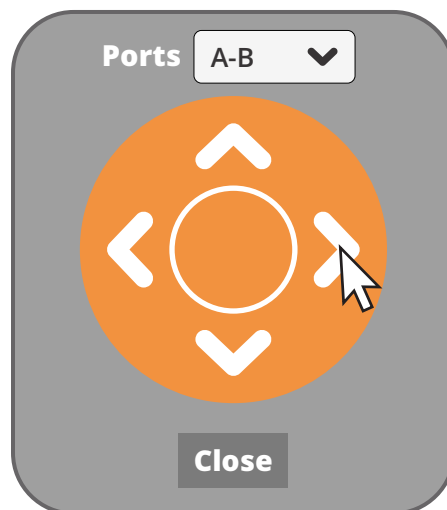
## Panel GamePad



Panel GamePad se objeví po kliknutí na speciální ikonu v Simulátoru.

Je vhodný k ovládání dvou motorových portů najednou. Nezapomeňte, že motory mohou být zapojeny pouze do portů A-B, proto by toto měla být zvolená kombinace. GamePad představuje užitečnou možnost, jak pohybovat robotickým vozidlem prostě a jednoduše.

Jednoduše klikněte (nebo se dotkněte na chytrém zařízení) na šipky a motory budou reagovat na váš příkaz.



*Panel GamePad*

Simulátor lze také využít k dálkovému ovládání modelu. Ikony, které napodobují rozhraní ovladače jsou aktivními tlačítky, na které můžete kliknout. Tím pádem lze nahrát či přehrát program přímo ze Simulátoru. Všimněte si, že při nahrávání programu se akční bloky automaticky tvoří na vývojovém diagramu.

## Live readings

V sekci **Live readings** dostanete po připojení ovladače k softwaru okamžitou zpětnou vazbu od periférií. Uživatel může sledovat stav každého připojeného zařízení např. rychlost a otáčení motoru, jestli je zapnuté LED nebo jestli je stisknutý dotekový senzor. Infračervený senzor můžete navíc zkalibrovat na rozpoznání určité vzdálenosti (více o kalibrování IR senzorů na str. 25).

The screenshot shows the 'Live Readings' interface with four sensor configurations:

- Motor:** Port A, Otáčení (Direction) ACL, Rychlost (Speed) 80.
- LED:** Port 2, Stav (State) ON.
- Dotykový senzor (Touch sensor):** Port 1, Stav (State) TRUE.
- Infračervený senzor (Infrared sensor):** Port 1, Stav (State) checked, Citlivost (Sensitivity) 65, Kalibrace (Calibrate) button.

**Motor:** Ukazuje směr otáčení (Po směru hodinových ručiček/

- Clockwise- **CL**, Proti směru hodinových ručiček/Anticlockwise - **ACL**) a úroveň rychlosti (**0-100**)

**LED:** Ukazuje, zda je zapnuté či není. (**ON** pokud svítí, jinak **OFF**)

- 

**Dotykový senzor:** Ukazuje, zda je stisknutý či není. (**TRUE** pokud je stisknutý, jinak **FALSE**)

- 

**Infračervený senzor:** Ukazuje, zda rozpoznává nějaký objekt. (✓ pokud ano, X pokud ne). (více na str. 25)

- 

## KEIRO Code

Software KEIRO generuje na základě vývojového diagramu bloků pseudojazyk. Jedná se o zásadní vzdělávací pomůcku, která představí studentům opravdové programování postupným přechodem od vizuálních bloků k pochopitelnému textovému formátu.

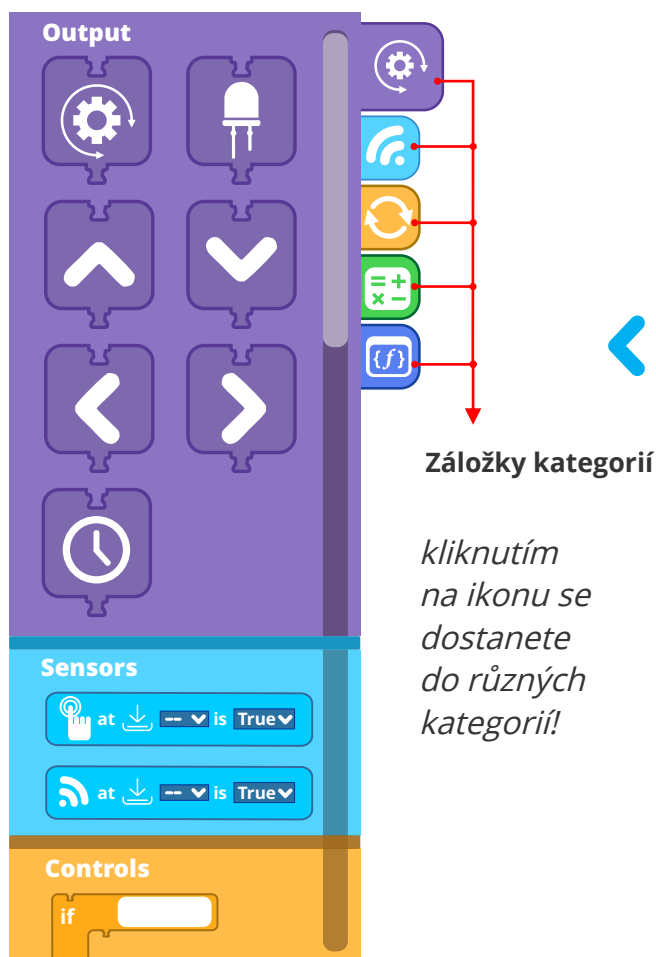
KEIRO Code obsahuje většinu pojmů běžně užívaných v nejrozšířenějších jazycích, např. BEGIN, IF, IF/ELSE, WHILE apod. Studenti si tak mohou přímo spojit akční bloky s textovými příkazy a naopak!

K skrytí či zobrazení panelu klikněte 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 akčních bloků



Veškeré bloky, které můžete využívat k vytvoření programu jsou na tomto menu. Uživatel si jednoduše vybere akční blok a ten přetáhne do vývojového diagramu - tak se tvoří program.

Ikonové bloky jsou rozříděny do 3 kategorií:

## 1. Outputy:

- Motor
- Pohyb (vpřed, vzad, vlevo, vpravo)
- LED
- Čekání

## 2. Senzory

- Dotykový senzor
- Infračervený (IR) senzor

## 3. Ovládání:

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

## 4. Operátory

## 5. Funkce

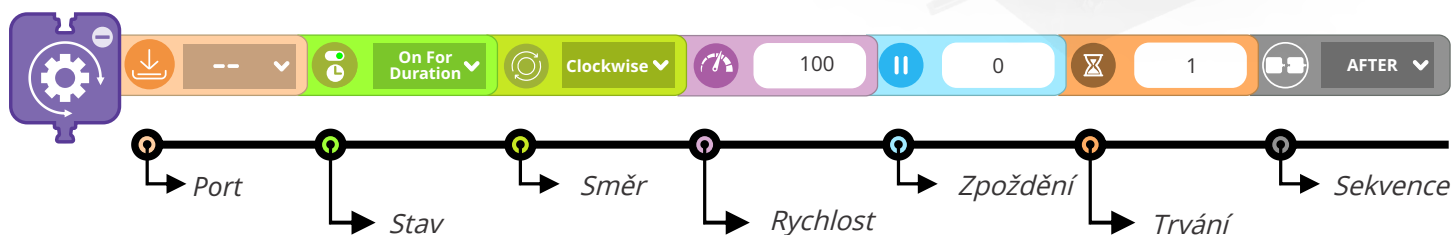
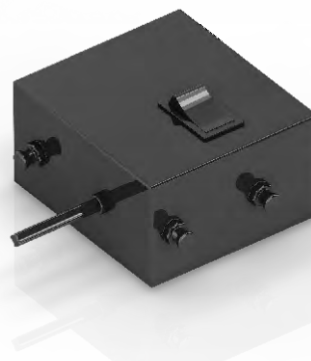
## 1. Outputy

Z této kategorie můžete do vývojového diagramu přetáhnout kterýkoliv output a vytvořit tak sekvenci akcí. Při umístění bloků do vývojového diagramu si všimněte, že se otevře **lišta vlastností**, ve které můžete blok upravit. Například když potřebujete upravit do kterého portu je output zapojen. Můžete si také upravit akční blok změnou času trvání a rychlostí motoru.



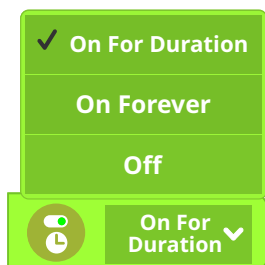
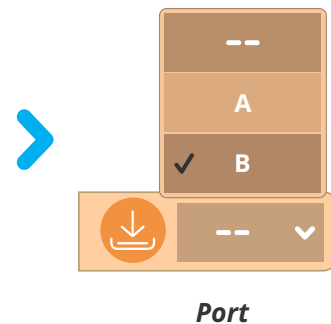
## Blok motoru

Níže vidíte rozepsaný blok motoru s jeho výchozími **vlastnostmi** a hodnotami.



Vyberte **Port**, kterým je motor k ovladači připojen. Tato vlastnost je **zprvu nevyplněna**, proto klikněte na pole a vyberte vhodný port. Dostupné porty jsou zabarveny tmavší barvou.

*Poznámka: simulátor by měl být nakonfigurován alespoň s jedním motorem, aby bylo možné vybrat port.*



**Stav**

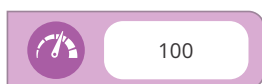
Nastavte Engino kidCAD (3D Viewer) aplikace: motoru. Máte na výběr ze tří možností: **On For Duration (výchozí)**: Motor se bude otáčet určitou dobu, která je nastavená ve vlastnosti Trvání (popsána níže). **On Forever**: Motor se bude otáčet nekonečně, dokud neobdrží jiný příkaz. **Off**: Zastaví rotaci motoru.

Vyberte Směr, kterým by se měl motor otáčet. Na výběr máte mezi rotací **clockwise/po směru hodinových ručiček (výchozí)** a **anticlockwise/proti směru hodinových ručiček**. Nezapomeňte, že aby motor fungoval správně s vybraným směrem, měl by mít vypínač v **pozici I**.

*Poznámka: pokud je stav nastavený na OFF, tato vlastnost není k dispozici.*



**Směr**



**Rychlost**

Nastavte **Rychlost** rotace motoru. Napište do pole hodnotu mezi **0 a 100**. Hodnota představuje procento přísunu energie do motoru, proto **100 (výchozí)** představuje maximální výkon.

*Poznámka: pokud je stav nastavený na OFF, tato vlastnost není k dispozici.*

Nastavte čas **Zpoždění** ve vteřinách k zpoždění spuštění bloku motoru. Pokud je vybrána hodnota 0 vteřin (**výchozí**), není nastavené žádné zpoždění.



**Zpoždění**

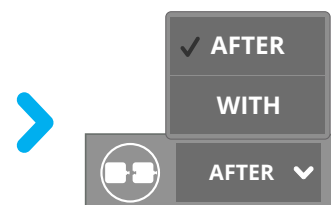


**Trvání**

Nastavte čas **Trvání** ve vteřinách, aby se motor otáčel určitou dobu. Po uplynutí času se motor přestane otáčet.

*Poznámka: Tato vlastnost je spuštěna pouze pokud je stav nastavený na "On For Duration".*

**Sekvence** bloků jsou ve **výchozím** nastavení provedeny **jedna za druhou (AFTER)**. K souběžnému provádění úkolů vyberte **"WITH"** (s předešlým), čímž se blok provede společně s předešlým. Tato vlastnost umožňuje spuštění několika bloků najednou.



## Navigační (Pohybové) bloky

K rozpoohybování vozidla by měly fungovat oba motory zároveň. Místo dvou bloků ve vývojovém diagramu vám pomůžou **pohybové bloky**, které ukazují, zda má vozidlo jet **vpřed, vzad, vlevo** či **vpravo**. Je to tak mnohem lepší k navigaci, snižují komplexitu a počet použitých bloků na vývojovém diagramu.



Vpřed



Vzad

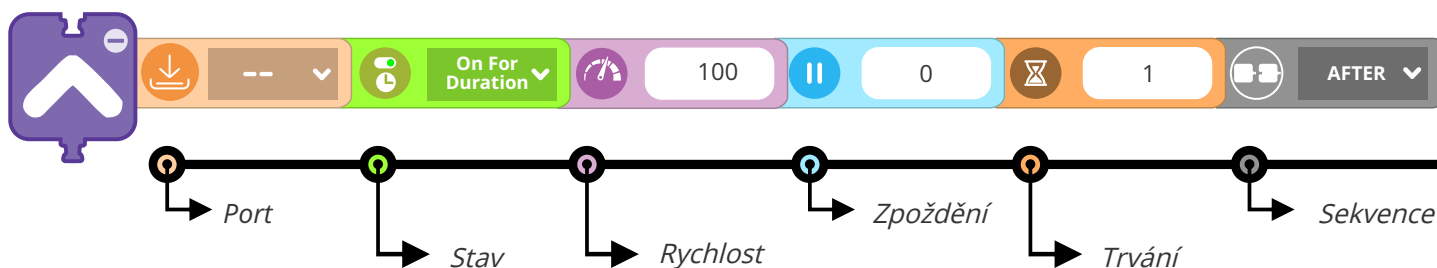


Vlevo



Vpravo

Níže vidíte příklad blok pohybu vpřed s jeho výchozím nastavením **vlastností** a hodnot. Tyto vlastnosti a funkce jsou stejné u všech ostatních navigačních bloků.



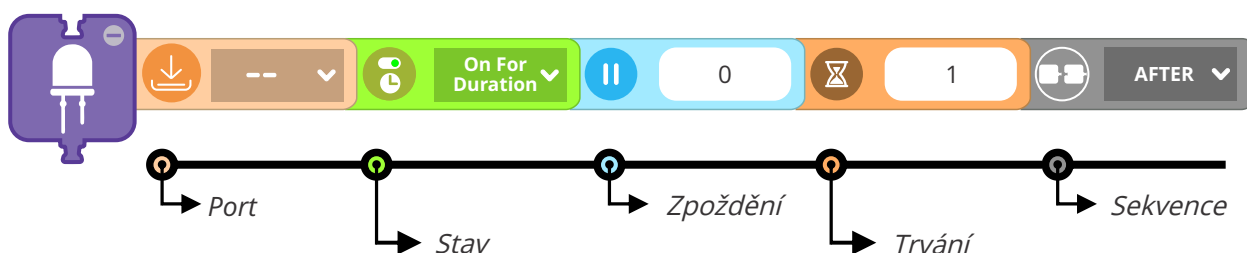
Navigační bloky používají stejné vlastnosti jako blok motoru. Nezapomeňte, že oba vypínače motorů musí být nastaveny do **pozice I**, aby navigační bloky mohly vést k pohybu robota.

## Blok LED

Light **E**mitting **D**iode (**LED**) je zdroj světla, který využívá elektřiny procházející krystalem, čímž generuje světlo. LED světla patří mezi nejefektivnější a nejbezpečnější zdroje světla s nízkou spotřebou, a proto se často využívají do elektronických zařízení a v robotice.

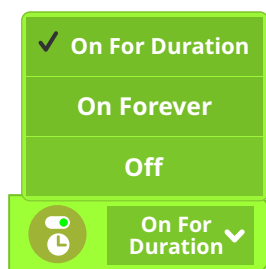
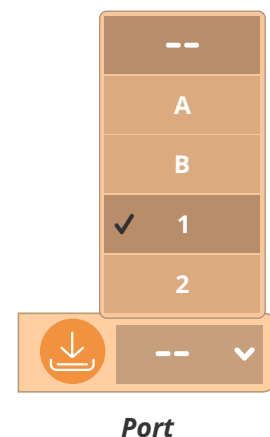


Níže vidíte rozepsaný blok LED s jeho výchozími **vlastnostmi** a hodnotami.



Vyberte **Port**, ve kterém je na ovladači MINI 2.0 motor připojený. **Výchozím** nastavením je pole **nevyplněné**, proto vyberte z pole vhodný port. Dostupné porty jsou zabarveny tmavější barvou.

*Poznámka: aby port fungoval, měl by být simulátor nakonfigurován s alespoň jedním LED.*



**Stav**

Nastavte **Stav** LED. Máte na výběr ze tří možností: **On For Duration (výchozí)**: LED bude svítit určitou dobu, která je nastavena ve vlastnosti Trvání (popsána níže). **On Forever**: Zapne LED, dokud neobdrží jiný příkaz. **Off**: LED se vypne.



Nastavte čas **Zpoždění** ve vteřinách k zpoždění spuštění LED. Pokud je vybrána hodnota 0 vteřin (**výchozí**), není nastavené žádné zpoždění.



**Zpoždění**



**Trvání**

Nastavte čas **Trvání** ve vteřinách, po jakou dobu bude LED světlo zapnuté. Po uplynutí času se LED vypne.



*Poznámka: Tato vlastnost je spuštěna pouze pokud je stav nastavený na "On For Duration".*

**Sekvence** bloků jsou ve **výchozím** nastavení provedeny jedna za druhou (**AFTER**). K souběžnému provádění úkolů vyberte **"WITH"** (s předešlým), čímž se blok provede společně s předešlým. Tato vlastnost umožňuje spuštění několika bloků najednou.

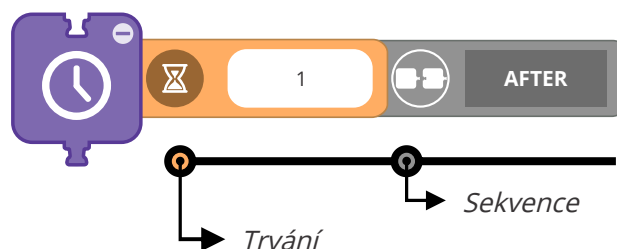


**Sekvence**

## Blok doby nečinnosti

Blok doby nečinnosti zastaví program po určitou dobu. Tato doba nečinnosti je nastavena ve vlastnosti **trvání** a ve výchozím nastavení je nastavena na 1 vteřinu.

Vzhledem k funkci doby nečinnosti není možné blok spustit paralelně s předešlým či následujícím úkolem.





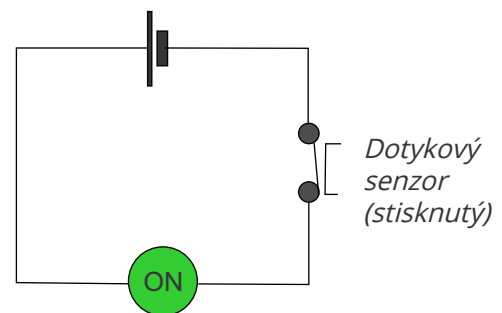
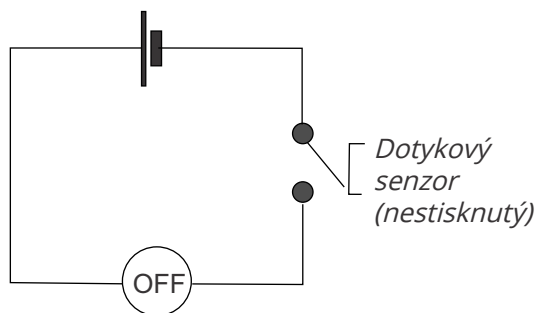
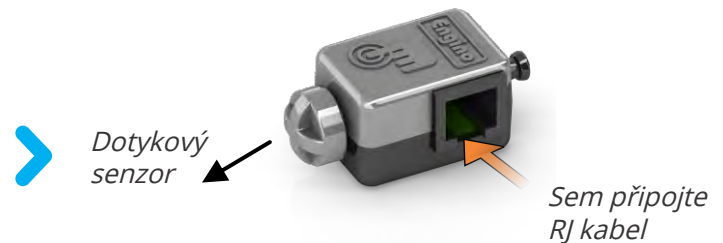
## 2. Bloky senzorů

Senzory jsou inputy, které ovládají chování robota. Vlastně se jedná o zásadní prvky, díky kterým můžeme nazývat stroj robotickým místo automatickým. Robot zvažuje získané informace ze svého okolí, aby mohl splnit úkoly dle situace, která nastala.



### Dotykový senzor

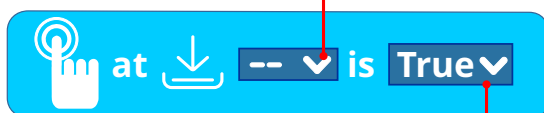
Dotykový senzor připojíte fyzicky k dotykovému senzoru Engino pomocí RJ kabelu přes kterýkoliv port ovladače. Engino dotykový senzor je v podstatě **dotykový vypínač**. Ze senzoru přicházejí dva typy signálů (TRUE a FALSE), které závisí na tom, jestli je stisknutý. Níže najdete schématický popis obvodů.



Při **stisknutí** vypínače se obvod uzavře, vytvoří se tok proudu a senzor je ve stavu **TRUE**:

Když vypínač **není stisknutý**, senzor je ve stavu **FALSE**, protože obvodem neprochází proud.

Vyberte port, ve kterém je k ovladači připojený dotykový senzor.



Vyberte stav, který by měl být zkontrolován, - TRUE nebo FALSE. Výchozím stavem je TRUE.

Blok senzoru nemůže být ve vývojovém diagramu jako nezávislý akční blok. Blok senzoru je vstup s **hodnotou boolean**. Pokud stav senzoru souhlasí s blokem, tak se jedná o input TRUE, když není, jedná se o FALSE.

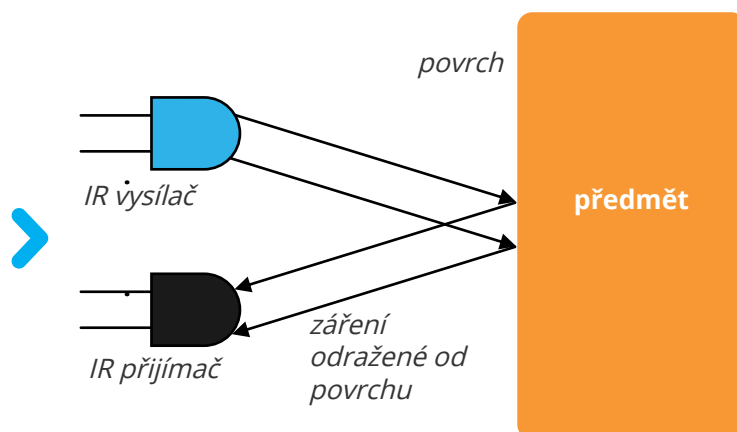
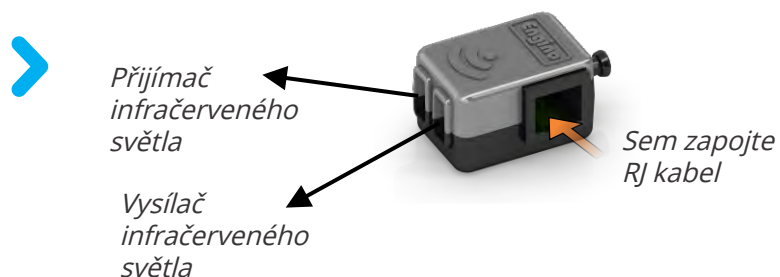
Blok senzoru tím pádem musíte použít ve spojení s příkazem ovládání, jako jsou IF, WHILE apod., aby robotovi řekli, zda je podmínka splněna (true) či není (false). O hodnotě boolean sensorového bloku se dozvíte více v sekcích **Ovládací bloky a Operátory**.

## Infračervený (IR) senzor

Engino Infračervený senzor lze připojit k ovladači RJ kabelem. Engino Infračervený senzor je **aktivní** IR senzor skládající se ze dvou prvků: **infračerveného vysílače** (zdroj) a **infračerveného přijímače** (detektor). Infračervené senzory lze využít k rozpoznání předmětů a rozpozná černé od bílých povrchů.

IR vysílač vysílá infračervené signály do dosahu infračervené oblasti (700nm až 1400nm) elektromagnetického pole. Tyto vzdálenosti lidské oko nemůže zachytit.

Pokud signál dojde k předmětu, jeho část se odrazí nazpět. IR přijímač je fototranzistor schopný rozpoznat IR záření. Senzor je tím pádem ve stavu **TRUE**, pokud se k němu odrazí hodně záření od předmětu, a ve stavu **FALSE**, pokud se odrazí záření málo a nebo žádné.

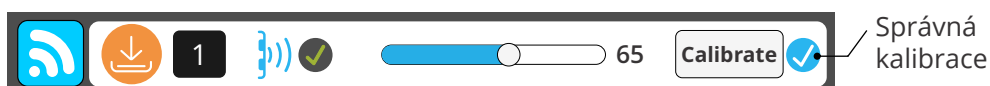


## Kalibrace infračerveného senzoru

IR senzor je nutné zkalibrovat na správný rozsah, který spustí robotovu reakci, aby robot "věděl", kdy se zachovat podle situace TRUE či FALSE. Kalibraci lze spustit z panelu **Live readings**.

Stav **TRUE** se zaktivuje při **bílé barvě** a vzdálenosti **stejně nebo menší než spouštěcí vzdálenost**.

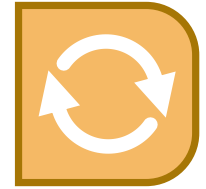
Stav **FALSE** se zaktivuje při **černé barvě** a vzdálenosti **větší než spouštěcí vzdálenost**.



**Rozpoznávání předmětů:** Umístěte IR senzor na stejnou vzdálenost, při které bude senzor fungovat. Držte model na místě a klikněte na **tlačítko Calibrate**. Kalibrace je hotova, jakmile se objeví vedle tlačítka Calibrate znak "✓". **Sledování čáry:** Umístěte IR senzor nad **bílý** povrch na stejnou vzdálenost, při které bude senzor fungovat. Proveďte kalibraci stejným způsobem, jak je popsáno výše.

\*Kalibraci lze provést také manuálně úpravou lišty na požadovanou spouštěcí úroveň.

### 3. Bloky řízení

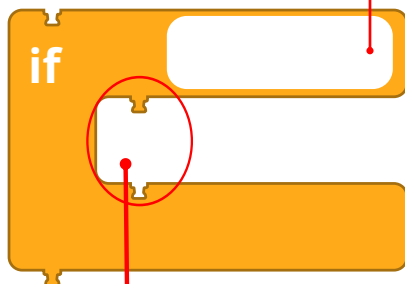


V této kategorii najdete bloky, které vytvářejí příkazy ovládání. Běžné podmínkové příkazy jako **IF**, **WHILE**, **REPEAT a UNTIL** pomáhají v sofistikovanějších a praktických využitích. Inputy ze senzorů se využívají k provedení bloků v podmínkových příkazech.

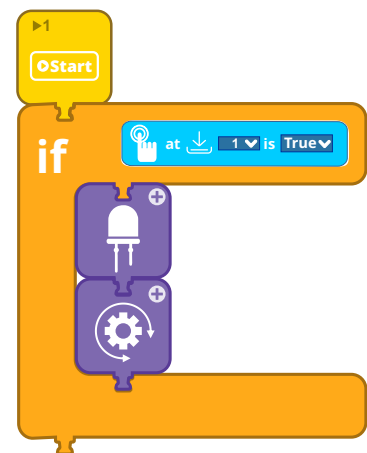
#### **IF:**

IF blok se spustí, pokud je splněna podmínka senzoru (True či False). Například, pokud je stisknutý dotykový senzor, zapne se LED a motor se začne otáčet. Pokud je senzor aktivován podruhé, ještě před provedením sekvence bloků, co jsou uvnitř, sekvence nezačne od znovu.

*Sem přetáhněte ikonu bloku, jehož podmínka by měla být kontrolována*



*Přetáhněte akční bloky k vytvoření sekvence, která by měla být provedena při splnění podmínky*



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

IF/ELSE má dvě části. Sekvence v první části se spustí v případě, že je splněna podmínka senzoru (například True při stisknutí dotykového senzoru), stejně jako v bloku IF. Pokud je podmínka opačná (například False, dotykový senzor není stisknut), robot rovede sekvenci bloků v druhé části bloku.

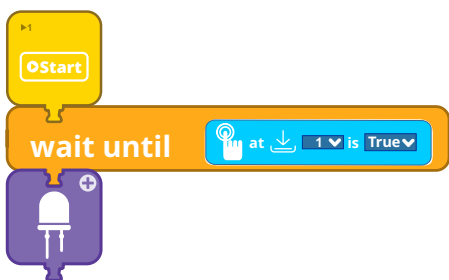
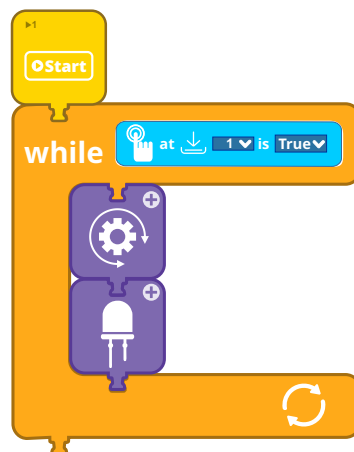


*V tomto případě se zapne LED, v případě, že IR senzor je TRUE (neboli rozpoznává předmět). V opačném případě se bude otáčet motor.*

*Bloky v ELSE budou přerušeny, jakmile se podmínka splní, ať už je jejich stav/trvání jakýkoliv a program spustí sekvenci v bloku IF.*

## While:

WHILE blok bude aktivní tak dlouho, dokud bude splněna podmínka senzoru (True nebo False, neboli bude plnit subprogram ve smyčce. Jakmile je podmínka přerušena, sekvence v bloku WHILE se okamžitě zastaví a program bude pokračovat k dalšímu akčnímu bloku. Například, *dokud bude stisknutý dotykový senzor, motor se bude otáčet po směru hodinových ručiček a LED bude svítit (dokud senzor nepustíte).*

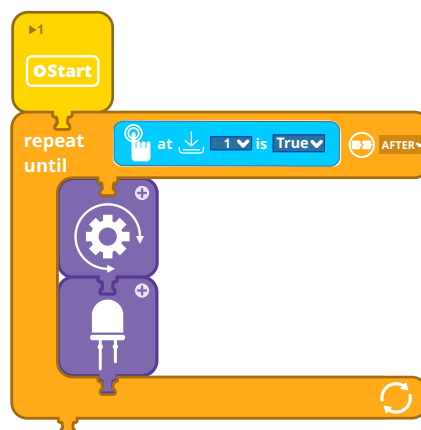


## Wait Until:

Tento blok nastaví program do módu nečinnosti, dokud není splněna určitá podmínka. Například, program nepřejde k rozsvícení LED dokud nestisknete dotykový senzor.

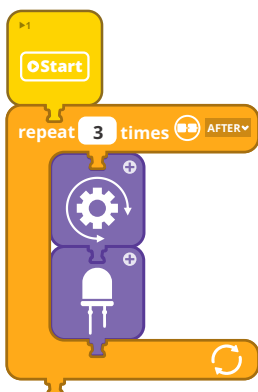
## Repeat Until:

Blok Repeat until bude přehrávat ve smyčce sekvenci bloků v něm a zastaví se jen v případě, že je splněna podmínka. Smyčka se přeruší, jakmile bude podmínka splněna a program bude pokračovat na další akční blok.

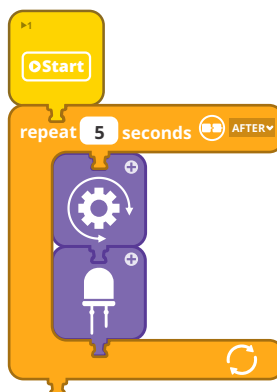


## Repeat:

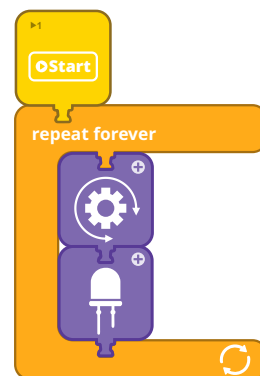
Opakovacích bloků, které nezávisí na senzorech nebo měnících se inputech, ale na pevných scénářích, najdete v softwaru tři typy. Blok "**repeat times**" vám dává možnost nastavit počet požadovaných opakování, zatímco "**repeat seconds**" povede k opakování sekvence po určitou dobu. Poslední možností je blok "**repeat forever**", který je určen k přehrávání sekvence v nekonečné smyčce. Všimněte si, že tento blok nemá žádný bod k připojení dalšího bloku, jelikož už není možné žádný přidat.



Smyčka třikrát



Smyčka 5 vteřin



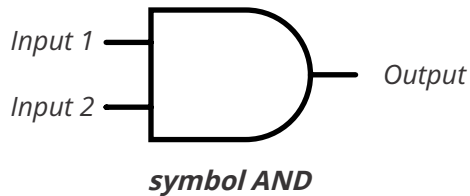
Nekonečná smyčka

## 4. Bloky operátorů

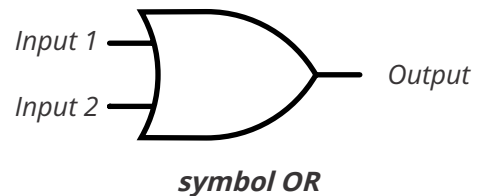


V této kategorii najdete logické operátory, které jsou užitečné v situacích, kdy by měla akce proběhnout při kombinaci podmínek. Obecně, operátoři říkají počítači, jak zacházet s informací.

**Logické operátory** jsou pravděpodobně nejdůležitějšími operátory a využívají se v každém programovacím jazyce. Kombinují dva inputy k vytvoření rozhodnutí. Předpokládejme, že chcete zkombinovat dva senzory, které povedou k příkazu robotovi. Logické operátory **AND** a **OR** by nám měli pomoci.



**Operátor AND:** Když jsou **oba** inputy "true", tak je i output "true". Jinak je output "false".

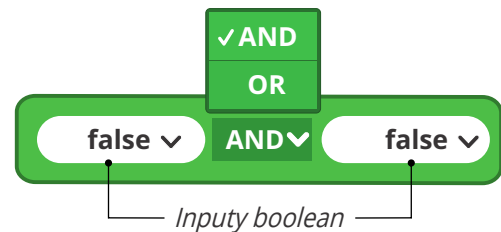


**Operátor OR:** Když je jeden nebo oba inputy "true", tak je output "true". Output vyjde jako "false", pouze pokud jsou oba inputy "false".

### Blok logického operátoru

Logický operátor softwaru KEIRO má dvě okna inputů, do kterých se vkládají hodnoty boolean, TRUE nebo FALSE (výchozí.). Kliknutím na pole uprostřed můžete vybrat operátor AND/OR.

Vzpomeňte si, že blok senzoru je prakticky hodnotou boolean, takže přetažením těchto slotů vznikne hodnota. Neboli, pokud budou oba sloty spojeny se senzorymi bloky, logický operátor vyhodnotí output na základě těchto dvou inputů.



Předpokládejme, že máme robotické vozidlo se dvěma infračervenými senzory na rozpoznávání překážek. Pokud nerozpoznají žádné překážky, chceme, aby se robot pohyboval vpřed. Takže **oba** senzory musí být ve stavu False. Blok dopředu tak bude opakován dokud budou obě dvě podmínky splněny.

## 5. Funkce

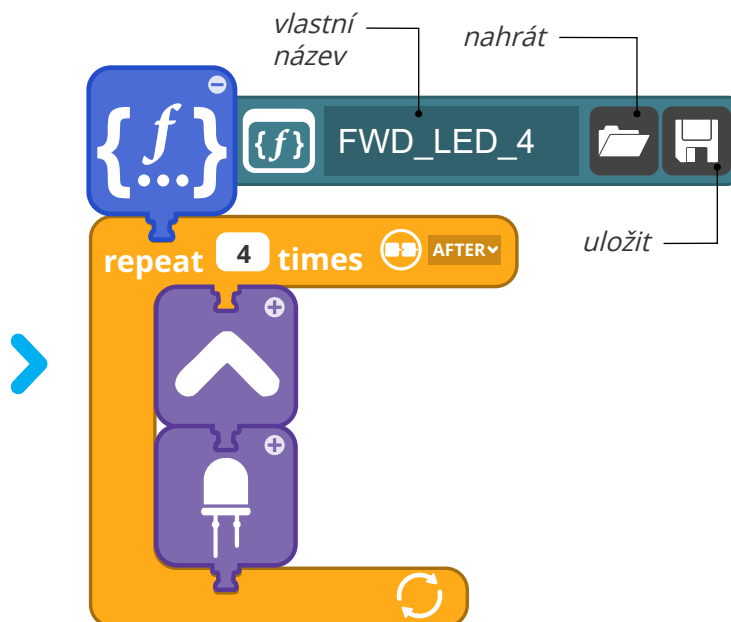
Funkce je podprogramem, který zahrnuje určitou sekvenci příkazů. Jedná se jednoduše o krátký program, který může využít či spustit hlavní program buď v sekvenčním či paralelním módu. Funkce jsou užitečné k menšímu počtu bloků ve vývojovém diagramu, hlavně v případech, kdy potřebujete provést opakovaně dlouhou sekvenci příkazů.



### Nastavení funkce

K vytvoření nové funkce přetáhněte blok "**set function**" (nastavit funkci) a přetáhněte ho do vývojového diagramu. Všimněte si, že blok není možné připojit k ostatním blokům, jelikož je prvním blokem nové sekvence příkazů.

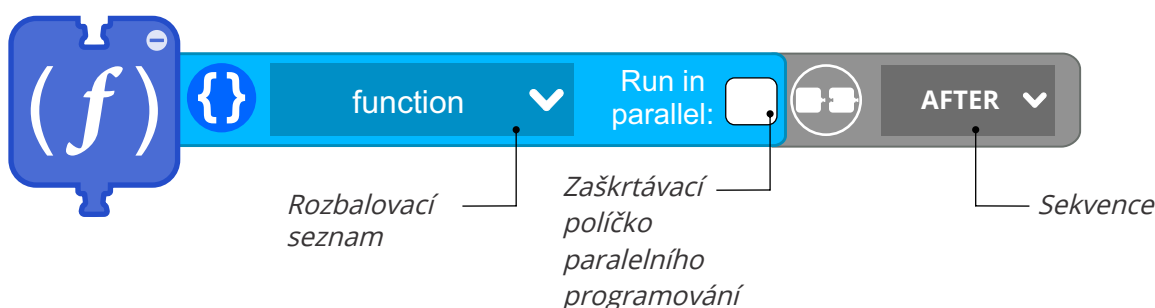
Můžete vytvořit podprogram pod blokem "set function" (nastavit funkci) s jakýmkoliv akčními bloky (outputy, senzory, ovládací bloky, apod.). Můžete si je také **pojmenovat** k jednodušší identifikaci, hlavně v případě, že budete nastavovat několik funkcí. Funkci si můžete uložit do počítače/tabletu a kdykoliv ji importovat pomocí ikon **uložit** a **nahrát**.



Funkce obsahuje podprogram s příkazy

### Použití funkce

K vyvolání funkce musí uživatel umístit blok "**Use a function**" (použít funkci) do sekvence vývojového diagramu. V případě, že máte nastavených několik funkcí, vyberete si funkci z rozbalovacího seznamu.



### Sekvenční režim

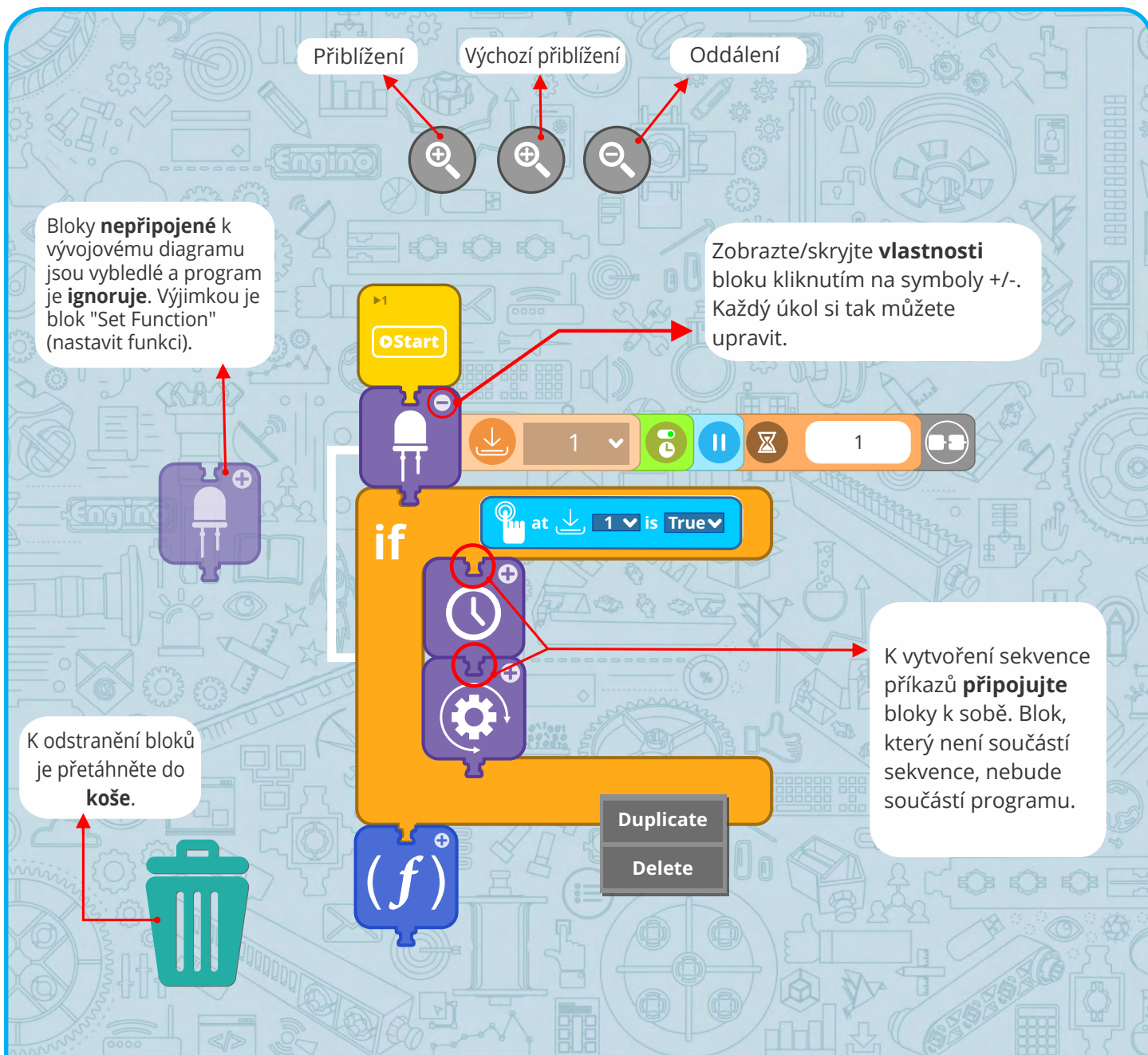
**Výchozím** nastavením funkce je **sekvenční režim**. To znamená, že jakmile program dojde k bloku "use function" (použít funkci) spustí všechny úkoly ve funkci. Jakmile dokončí všechny úkoly ve funkci, přesune se k dalšímu bloku vývojového diagramu.

### Režim "Run in parallel" (soubežně):

**Paralelní režim** spustíte zaškrtnutím zaškrťovacího pole. V tomto režimu program spustí úkoly ve funkci a okamžitě spustí další akční blok ve vývojovém diagramu. Tímto způsobem se podprogramové úkoly ve funkci spustí paralelně se zbytkem vývojového diagramu.

# Programování vývojového diagramu

Níže vidíte vývojový diagram s popisky jeho hlavních prvků. K vytvoření programu **přetahujete** bloky z Lišty bloků (levá strana obrazovky) pod ikonu "**Start**" v hlavním okně. Všimněte si, že jakmile přetáhnete akční blok blízko volnému bodu připojení, oblast ztmavne. Klikněte na symbol "+" na blocích k upravení vlastností každého akčního bloku jako je například port, trvání apod. Po dokončení programu klikněte na Liště menu na tlačítko "**Send program**" a program bude poslán ovladači MINI.



Myší můžete získat další informace a dělat změny následujícími kroky:

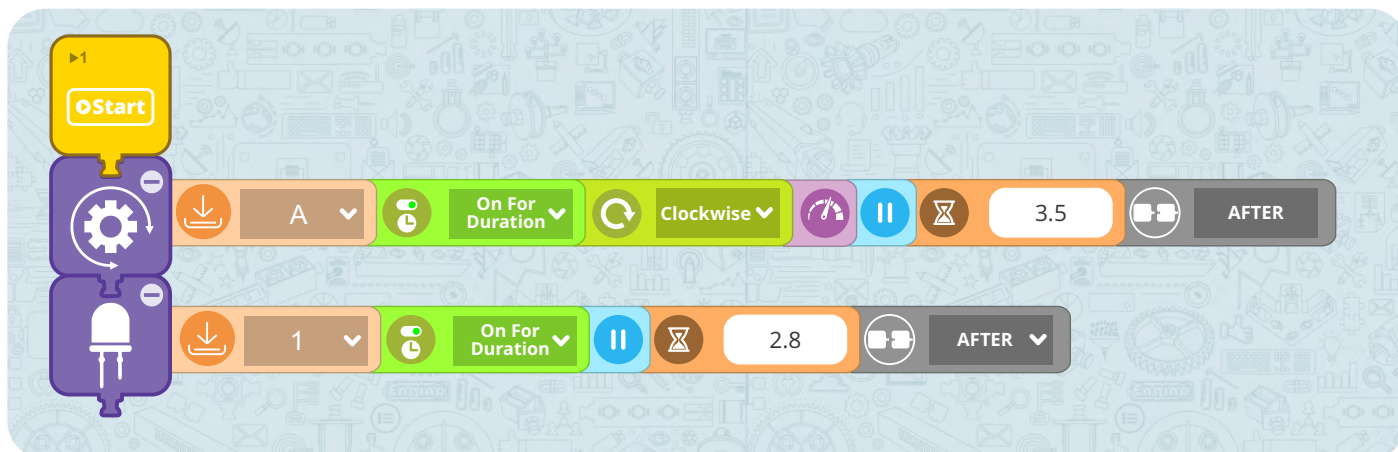
- **Přesuňte** svůj kurzor myši nad ikony a bloky, čímž zobrazíte popisy a tipy.
- **Kliknutím** u bloků vývojového diagramu na **symbol** "+" zobrazíte vlastnosti bloků, které můžete upravit.
- **Kliknutím** na **symbol** "-" skryjete vlastnosti bloku.
- **Pravým kliknutím** na blok ho můžete duplikovat nebo rychle smazat jeden či více bloků.

## Ukázky kódů

### Ukázka 1 - Jednoduché programování s motorem a LED

Tato ukázka využívá jeden motor a LED, které jsou v daném pořadí připojeny k portům A a 1. V programu se bude nejprve otáčet motor po směru hodinových ručiček, a to po 3,5 vteřiny a poté se zapne světlo LED na 2,8 vteřiny. Program bude proveden pouze jednou.

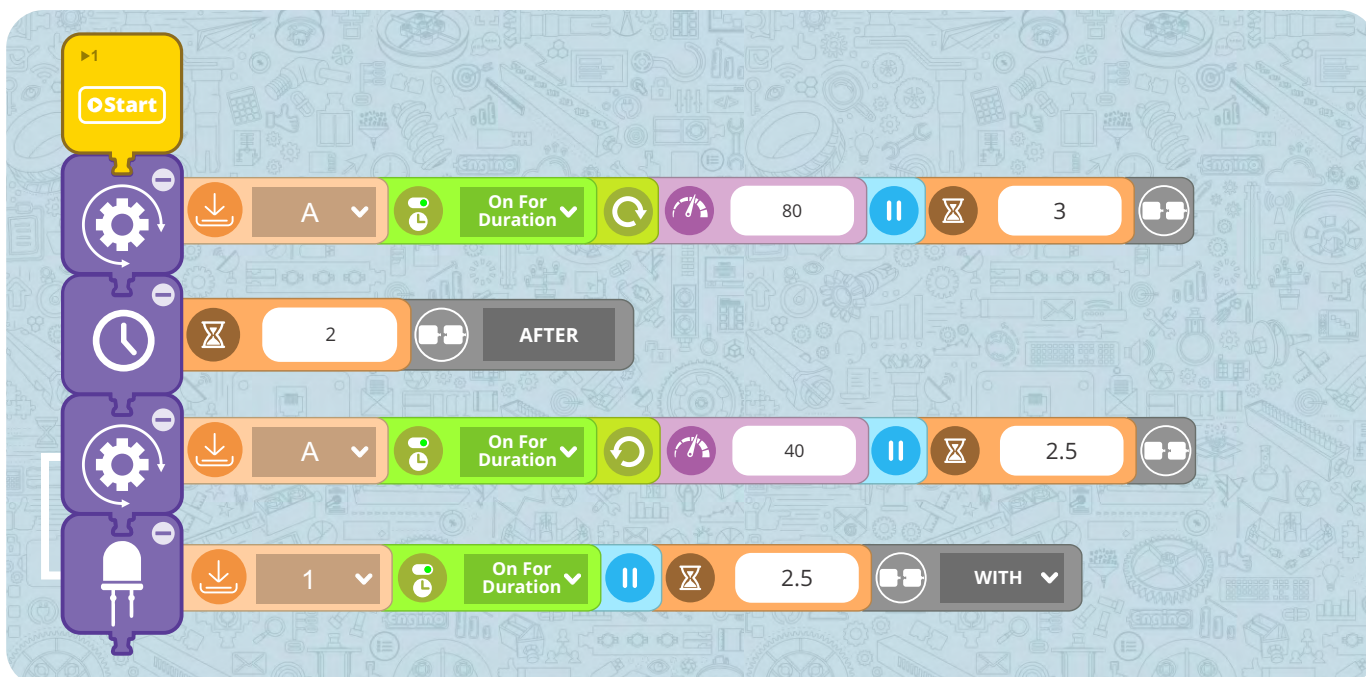
- Vyberte stav **On For Duration**
- Změňte čas trvání
- Interval trvání vyplňte reálným číslem (float)



### Ukázka 2 - Použití souběžných úkolů a Bloku doby nečinnosti

Tato ukázka využívá jeden motor a jedno LED, které jsou v daném pořadí připojeny k portům A a 1. V programu se bude nejprve otáčet motor, a to po 3 vteřiny na úrovni rychlosti 80%. Program se bude poté 2 vteřiny nečinný. Pak se motor začne otáčet rychlostí 40% a zároveň se na 2,5 vteřiny světlo LED.

- Změňte rychlost motoru
- Použití bloku doby nečinnosti
- Programování souběžných úkolů





### Ukázka 3 - Program ve smyčce

Následující ukázka využívá jeden **motor** a **LED**, které jsou v daném pořadí připojeny k portům B a 2. Úkolem je, aby se motor otáčel proti směru hodinových ručiček 1,5 vteřin, poté by se mělo zapnout světlo LED na 1 vteřinu a tato sekvence by se měla opakovat do nekonečna.

- Změňte směr otáčení
- Program v nekonečné smyčce



### Ukázka 4 - Podmínkový příkaz IF

Tato ukázka využívá jeden **motor**, jedno **LED** a **dotykový senzor**, které jsou v daném pořadí připojeny k portům B, 1 a 2.. Dokud **nestisknete** dotykový senzor (stav je **FALSE**) bude se motor otáčet proti směru hodinových ručiček. Pokud dotykový senzor stisknete (stav je **TRUE**), zapne se LED na 1,5 vteřiny a motor se bude otáčet proti směru hodinových ručiček 1 vteřinu, ale se zpožděním 0,5 vteřiny.

- Stav **On Forever**
- Dotykový senzor
- Podmínkový příkaz **IF**
- Čas zpoždění



## Konstrukce a programování

Do robotiky zahrnujeme dva hlavní aspekty: část konstrukce a část programování. Na modelu větráku můžete experimentovat s obojím. Naučte se, jak upravit konstrukci tak, aby byla efektivnější. Poté si manuálně naprogramujte ovladač a využijte vizuálních bloků v softwaru KEIRO.

## Zjistěte:

- Co jsou to inputy a outputy.
- Jak vylepšit konstrukci robota.
- Jak manuálně naprogramovat robota.
- Co je to vývojový diagram.

Úroveň obtížnosti ★★★★★

### Potřebné materiály

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A)
- Počítač nebo chytré zařízení se staženým a nainstalovaným softwarem KEIRO.

### Postup:

**1. Na stránkách 43-45** si najdete instrukce a postavte model **větráku**. Ujistěte se, že jste připojili všechny kabely ke správným portům. Obmotejte kabely kolem ostatních součástek, aby se vám nepletly kolem. Do zadní části ovladače vložte 3 AAA baterie a zapněte zařízení pomocí tlačítka On-Off.



**2.** Vypínač obou motorů by měl být nastaven v pozici **I**. V pozici **O** je motor vypnutý a v pozici **II** spouští opačný směr. Stiskněte tlačítka motoru (příkaz) na ovladači, jak je ukázáno ve **cvičení 1** a запиšte si výsledky, které jste zachytili (output).

**3.** Vylepšete svůj model pomocí instrukcí ve **cvičení 2**.

**4.** Ve **cvičení 3** si vytvoříte na ovladači manuální program. Možná budete potřebovat pár pokusů, než se vám to povede.

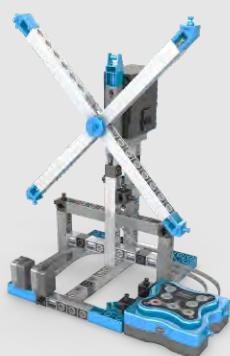
**5.** Připojte ovladač k počítači nebo chytrému zařízení. Spusťte software KEIRO a klikněte na tlačítko **připojit**.



**6.** Klikněte na tlačítko "**Receive Program**" (Obdržet program) k nahrání nahrané sekvence do softwaru KEIRO, čímž se vám vytvoří "**vývojový diagram**".





**7.** Udělejte ve vývojovém diagramu změny popsané ve **cvičení 4**. Pošlete pak program zpět do ovladače kliknutím na tlačítko "**Send Program**". Uložte si program jako "**větrák**".

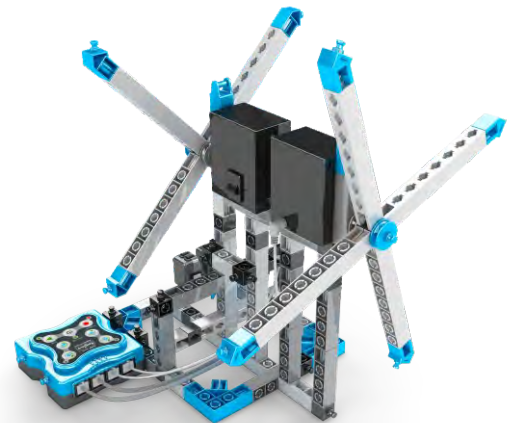


Engino model větrák

**Cvičení 1.** Stiskněte tlačítka na ovladači (input) a sepište stručně výstup, který vysledujete na modelu (output).

Příkaz	Output (výstup)
Vypínač Motoru A v pozici I 	
	

**Cvičení 2.** Vytvořte na opačné straně vašeho větráku druhý větrák a připojte ho do portu B. Napravo vidíte příklad konstrukce.



**Cvičení 3.** Stiskněte "**Program**" (červeně blikající světlo ukazuje, že se program nahrává) a poté potřebná tlačítka k vytvoření následující sekvence: větrák A se otáčí po směru hodinových ručiček; větrák B se otáčí po směru hodinových ručiček; oba větráky se točí zároveň v opačném směru



Stiskněte znovu tlačítko "**Program**" k uložení svého programu a pak stiskněte tlačítko "**Play**", ke zkontrolování správnosti sekvence. Pokud tlačítko Play stisknete a podržíte, program se spustí ve smyčce.



**Cvičení 4.** Na vývojovém diagramu upravte vlastnosti všech akčních bloků k vytvoření programu popsaného níže. Můžete přidávat další akce (motory, doby nečinnosti) z levého menu: větrák A se otáčí po směru hodinových ručiček 2 vteřiny rychlostí 100; větrák B se otáčí po směru hodinových ručiček 2 vteřiny rychlostí 100; robot nedělá 1 vteřinu nic (blok doby nečinnosti); oba větráky se otáčejí zároveň stejným směrem po 3 vteřiny, A rychlostí 100 a B rychlostí 40.



## Naučte se:

## Robotiku

### Pohyb robota a příkaz IF

Pohyb robota je velice důležitý a občas je těžké ho dosáhnout. K rozpořádání robota existuje mnoho způsobů, které většinou záleží na jejich využití: někteří mají kola nebo drýhy, někteří mají nohy a jiní třeba létají pomocí vrtule. Zaexperimentujte si s pohybem nohou u modelu hexapodu!

## Zjistěte:

- Jak funguje u robotů pohyb nohou.
- Jak se používá Simulátor a Gamepad.
- Jak se používají infračervené senzory s podmínkovým příkazem IF.

Úroveň obtížnosti ★★★★★

#### Potřebné materiály

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A)
- Počítač nebo chytré zařízení se staženým a nainstalovaným softwarem KEIRO.

#### Postup:

**1. Na stránkách 46-51** si najdete instrukce a postavte model **hexapodu**.

**2.** Připojte ovladač k počítači nebo chytrému zařízení. Spusťte software KEIRO a klikněte na tlačítko **připojit**. Splňte **cvičení 1 a 2**, ve kterých se naučíte, jak pracovat se **Simulátorem** v softwaru KEIRO.

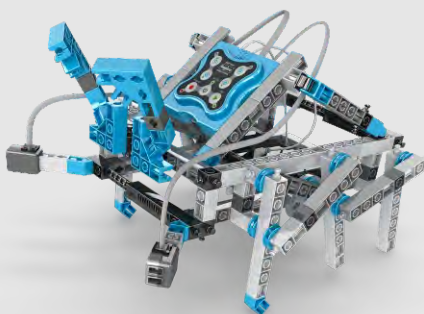
**3.** Otevřete si okno **Gamepad**

kliknutím na správné tlačítko a ovládejte jím svého robota.



**4.** Všimněte si, že k pohybu dopředu nebo dozadu musí nohy začínat stejnou pozicí. U koleček by tento problém vůbec nenastal a pohyb dopředu je závislý jen na rychlosti otáčení motorů. Napadnou vás nějaké další problémy při pohybu pomocí nohou (souvlosti)? Napište je do **cvičení 3**.

**5.** Splňte **cvičení 4**, kde si naprogramujete hexapoda tak, aby se vyhýbal překážkám z obou stran. K tomu budete potřebovat podmínkový příkaz IF.



Engino model hexapod

**Cvičení 1. Přetáhněte** periferie (motory a senzory) na porty v Simulátoru podle modelu hexapodu. Bloky si zapište také sem do obrázku.



**Cvičení 2.** Ovládejte hexapoda pomocí Simulátoru dle následující sekvence: hexapod zatočí doprava pomocí jedné nohy; hexapod zatočí doleva pomocí jedné nohy; hexapod zatočí doprava pomocí obou nohou.

**Cvičení 3.** Jaké problémy mohou nastat, když používáte místo kol nohy?

.....

.....

.....

.....

**Cvičení 4.** Vytvořte uvedený program, díky kterému se hexapod bude vyhýbat překážkám z obou stran v nekonečné smyčce: hexapod se opakovaně pohybuje dopředu; **pokud** levý IR senzor rozpozná překážku, pravá noha by se měla pohnout dozadu a levá noha dopředu; **pokud** pravý IR senzor rozpozná překážku, levá noha by se měla pohnout dozadu a pravá noha dopředu.

Umístěte do IF bloku správné bloky. Ke konfiguraci IR senzorů na rozpoznávání předmětů se podívejte na **stránku 25**.

**Příkaz WHILE a logické členy**

Infračervené senzory se v robotice využívají velice často, a to pro dvě využití: rozpoznávání mezi černou a bílou barvou a rozpoznávání předmětů. Vyzkoušejte si obě programovací výzvy a naučte se, jak nakonfigurovat IR senzory, jak se používá podmínkový příkaz WHILE a logické členy.

**Zjistěte:**

- Jak fungují infračervené senzory s podmínkovými příkazy WHILE a IF.
- Co jsou to logické členy.

Úroveň obtížnosti ★★★★★

**Potřebné materiály**

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A)
- Počítač nebo chytré zařízení se staženým a nainstalovaným softwarem KEIRO.
- 4x bílé čtvrtky formátu A3
- Černou izolepu nebo černou fixu
- 4 bílé předměty pro detekci

**Postup:**

**1. Na stránkách 38-42** si najdete instrukce a postavte model **robotu**. Ujistěte se, že jste připojili každé kolečko k portu na stejné straně.

**2.** Splete papíry A3 k sobě, abyste vytvořili jeden velký obdélník. Nakreslete černou velkou elipsu fixou nebo izolepou (obrázek napravo). Jiný tvar si můžete stáhnout na stránkách [www.enginorobotics.com/teaching\\_resource\\_s/examples/](http://www.enginorobotics.com/teaching_resource_s/examples/)

**3.** Otevřete si software KEIRO a následujte instrukce ve **cvičení 1** k vytvoření **programu sledování bílé. IR senzory by měly koukat do země**, takže je nastavte směrem dolů. Robot se bude pohybovat na bílé a zastaví kola na černé, neboli bude následovat čáru.

**4.** Přečtěte si **cvičení 2** k přeměně na robota, který **rozpoznává předměty**. Stačí upravit **IR senzory** tak, aby **koukaly dopředu**. Po vytvoření programu ho můžete otestovat umístěním čtyř bílých objektů (vysokých stejně jako senzory) k vytvoření imaginárního čtverce, po jehož stranách se bude robot pohybovat (obrázek napravo.)

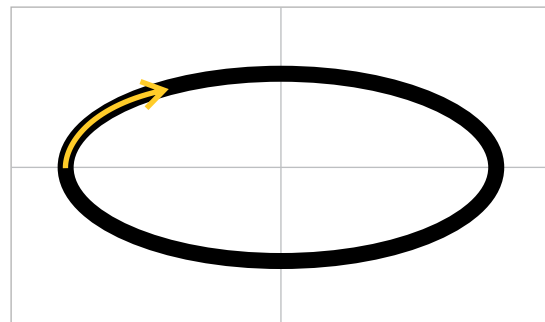


Engino model robota

**Cvičení 1.** Na počítači nebo chytrém zařízení umístěte v Simulátoru správné ikony podle modelu robota. Přetáhněte potřebné akční bloky do vývojového diagramu k vytvoření následujícího programu pohybu po bílé a zafixování cesty po černé: **zatímco** (while) IR senzor v portu 1 rozpoznává bílou barvu, motor v portu A by se měl otáčet dopředu; **zatímco** (while) IR senzor v portu 2 rozpoznává bílou barvu, motor v portu B by se měl otáčet dopředu.

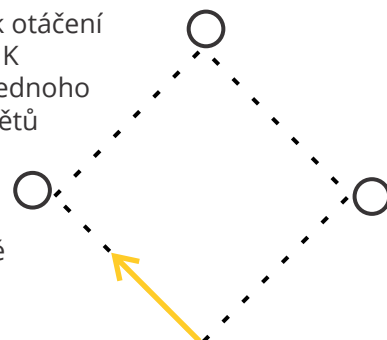
Po umístění správných bloků bude potřeba **zkalibrovat infračervené senzory** na následování čáry. Více najdete na straně 25 v sekci "Kalibrace infračerveného senzoru" uživatelského manuálu.

S tímto programem se robot bude pohybovat po černé lince. K otestování vašeho programu robota umístěte na černou elipsu, kterou jste si vytvořili, a sledujte, jestli bude linku následovat.

**Cvičení 2.**

Přetáhněte do vývojového diagramu potřebné bloky k vytvoření následujícího programu: robot se pohybuje vpřed; **pokud** (if) rozpozná bílý předmět **jedním nebo oběma senzory**, robot se zastaví; po 5 vteřinách by se humanoid měl otočit o 90 stupňů doprava, aby mohl rozpoznat další předmět.

Tipy: úhel je vždy stejný, takže k otáčení je nutný pouze jeden program. K nastavení obou nebo alespoň jednoho senzoru k rozpoznávání předmětů musíte použít správný logický člen (AND a OR). Měli byste také znovu **zkalibrovat IR senzory**, aby byly aktivované k **rozpoznávání předmětů**.



**Příklad z praxe: průmyslové rameno**

Roboti se často využívají v průmyslu, kde hrají hlavní roli výrobních linek. Jejich nejčastějším využitím je podoba stacionárních ramen, která přesouvají části z jednoho místa na druhé pomocí technologie senzorů. S twister modelem si můžete vyzkoušet simulovanou verzi těchto situací.

**Zjistěte:**

- Jak fungují stacionární roboti.
- Jak funguje podmínkový příkaz REPEAT TIMES.
- Jak funguje podmínkový příkaz REPEAT UNTIL.

Úroveň obtížnosti ★★★★★

**Potřebné materiály**

- Creative Engineering 100 in 1 Robotized Maker Pro (CE101MP-A)
- Počítač nebo chytré zařízení se staženým a nainstalovaným softwarem KEIRO.
- Izolepa a 1 lehkou plechovku

**Postup:**

**1. Na stránkách 52-57** si najdete instrukce a postavte model **twistera**.

**2.** Stacionární znamená, že stojí na pevném místě. Ve **cvičení 1** upevníte robota, aby se nehýbal při otáčení.

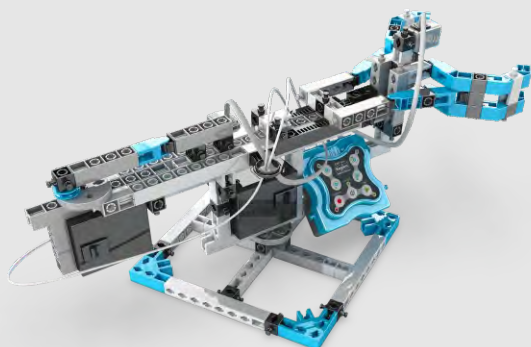
**3.** V programování jsou velice důležité programy s opakujícími se přesnými úkoly. Ve **cvičení 2** vytvoříte program ve smyčce, a to pomocí bloku "repeat times".



**4.** Někdy chceme, aby se smyčka opakovala, dokud není splněna podmínka senzoru. Ve **cvičení 3** se naučíte, jak toho docílit, pomocí bloku "repeat until".



**5.** Stacionární roboti často fungují na základě přesného návodu k dokončení rutinní práce. K tomu se využívá technologie IR senzorů, aby robot věděl, jak se pohybovat. Pokud využijete pouze časování, po mnoha opakováních začne být robotův pohyb chybný. Ve **cvičení 4** si vytvořte simulaci rutinní robotické práce.



Engino model twister

**Cvičení 1.** Vytvořte si manuálně jednoduchý program otáčení ramenem. Umístěte robota na zem a sledujte, co se bude dít při přehrání programu. Uvidíte, že se otáčí i základna. Tento problém spravíte umístěním žluté kladky pod všechny 4 rohy základny. Pro větší stabilitu můžete umístit kladky na izolepu. Vyzkoušejte program znovu.



**Cvičení 2.** Na počítači nebo chytrém zařízení umístěte správné ikony do na Simulátor podle modelu twister. Přetáhněte potřebné bloky do vývojového diagramu k vytvoření následujícího programu: *twister se bude otáčet 1 vteřinu rychlostí 50, pak zavře a otevře své rameno; tato sestava by se měla zopakovat 5 krát.*

Tip: Použijte blok "repeat times".

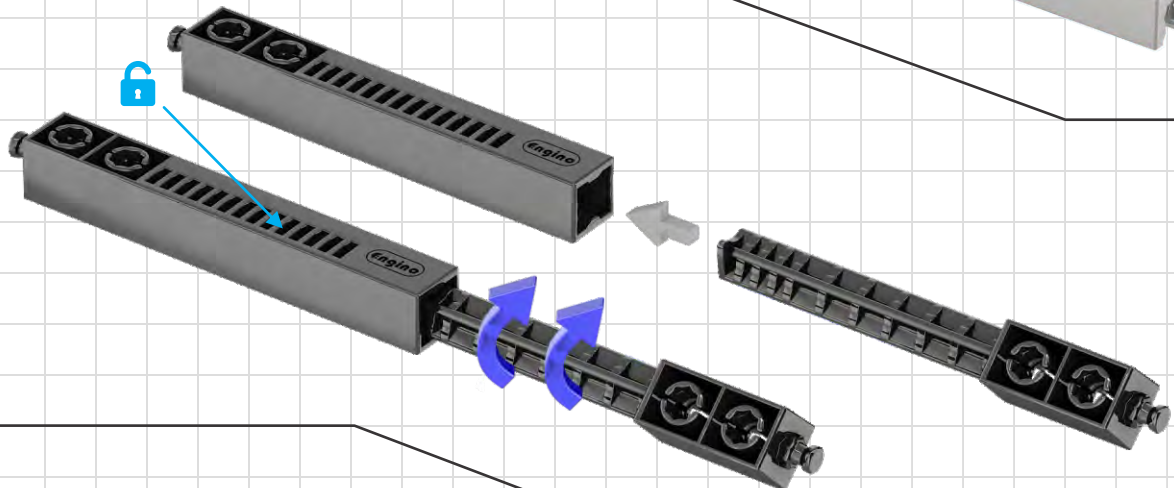
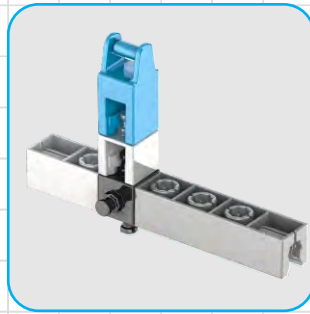
**Cvičení 3.** Vytvořte následující program (sevření ramene by mělo být na začátku otevřené): twister se bude otáčet, **dokud** (until) boční IR senzor nerozpozná předmět nebo ruku; pokud ve chvíli nečinnosti modelu umístíme před IR senzor ramene plechovku, sevření by se mělo uzavřít a chytit ji.

TIP: Použijte blok "repeat until" pro první část programu a upravte rameno pro lepší sevření, pokud to bude potřeba.

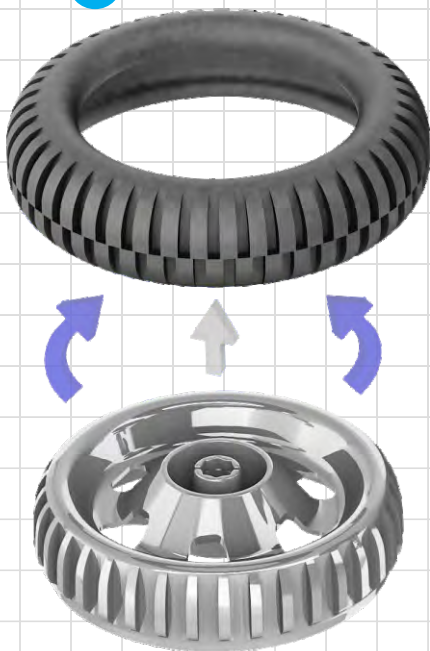
**Cvičení 4.** Umístěte do ramene modelu plechovku a vytvořte následující program: *twister se otočí o 180 stupňů doleva s plechovkou v sevření; jakmile model zastaví, rameno by se mělo povolit a upustit plechovku; twister by se poté měl otočit o 180 doprava (neboli do své původní pozice); jakmile umístíte plechovku před senzor ramene, model by ji měl sevřit a postup opakovat (otočení o 180 stupňů, upuštění plechovky a otočení nazpět).*

Tip: umístěte dva předměty (nebo svou ruku) na jednu ze stran prostředního senzoru k jeho spuštění.

# Tipy ke stavění



tire  
L



Dvojitý rozpojovací nástroj



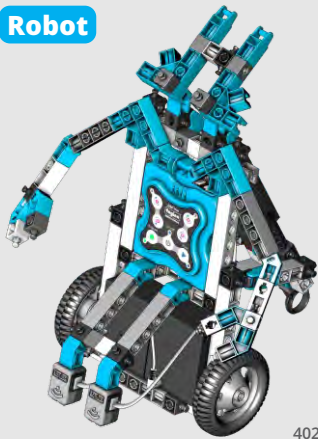
**2. MOŽNOST:**  
Vložte nástroj do otvoru a vytlačte tak druhou součástku!

**1. MOŽNOST:**  
Zmáčkněte nástroj, čímž rozpojíte malé součástky!



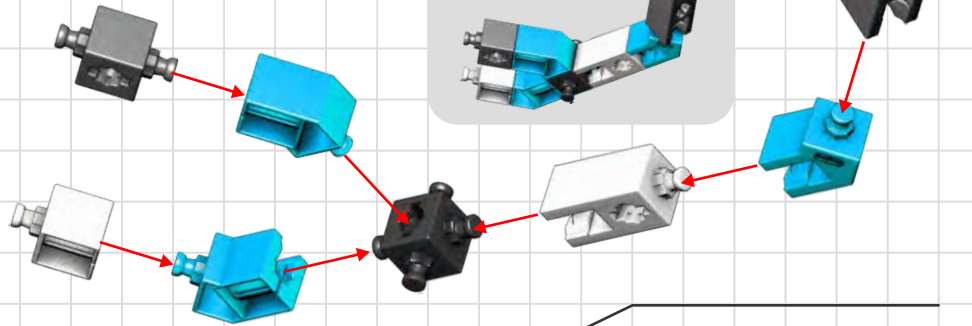
# Návod na stavění

Robot

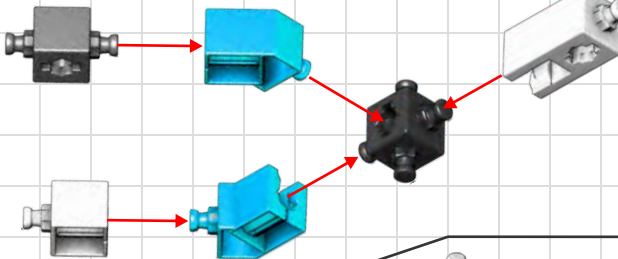
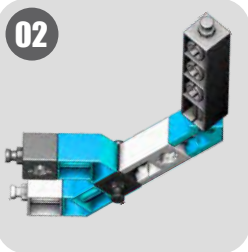


4023

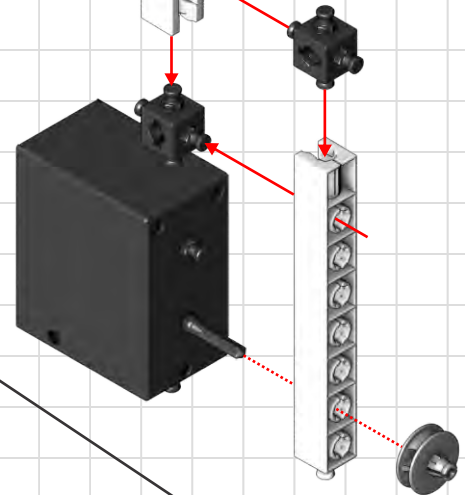
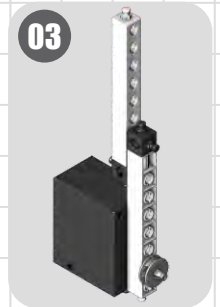
01



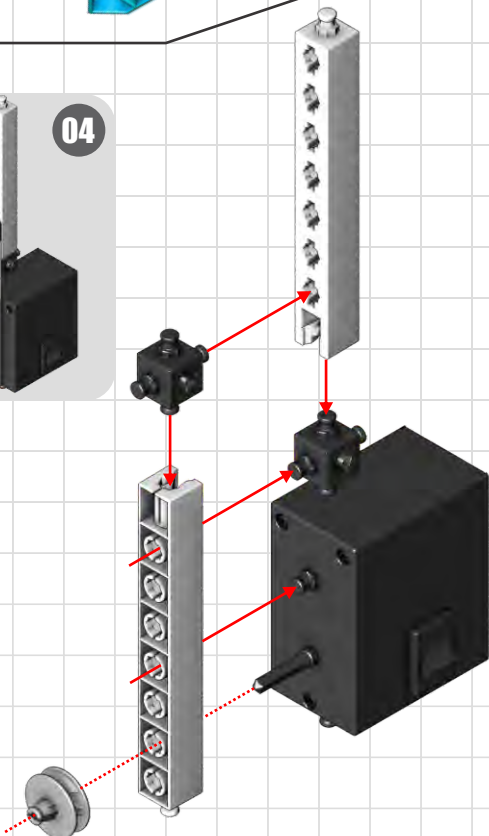
02



03

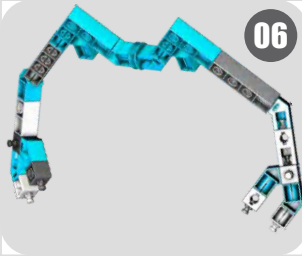


04

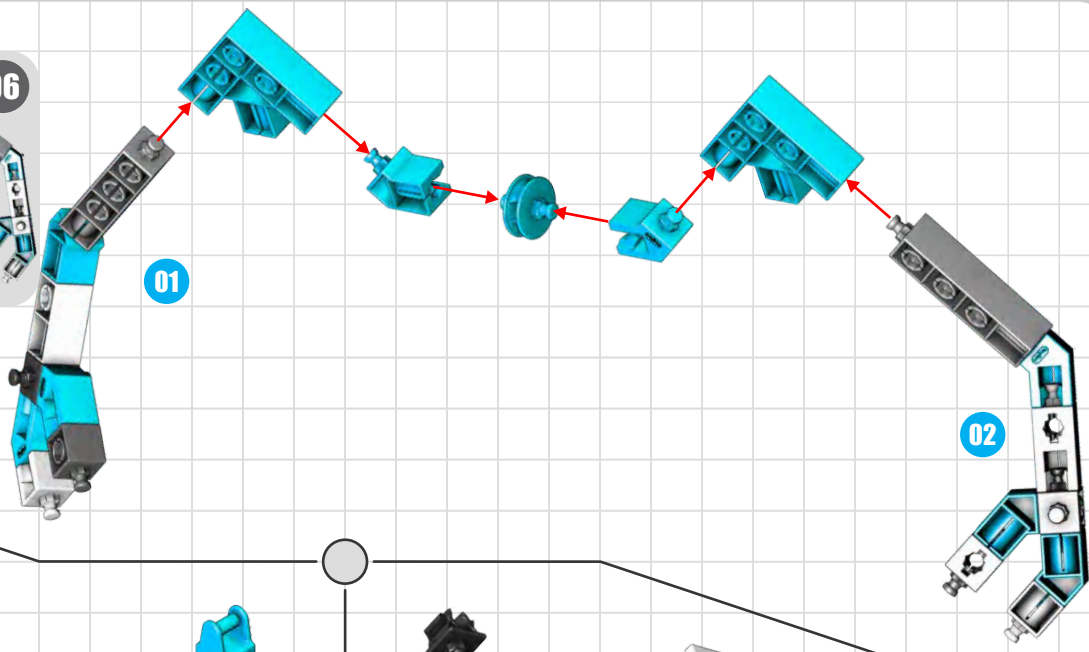


05

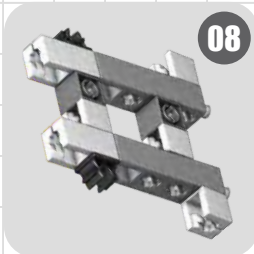




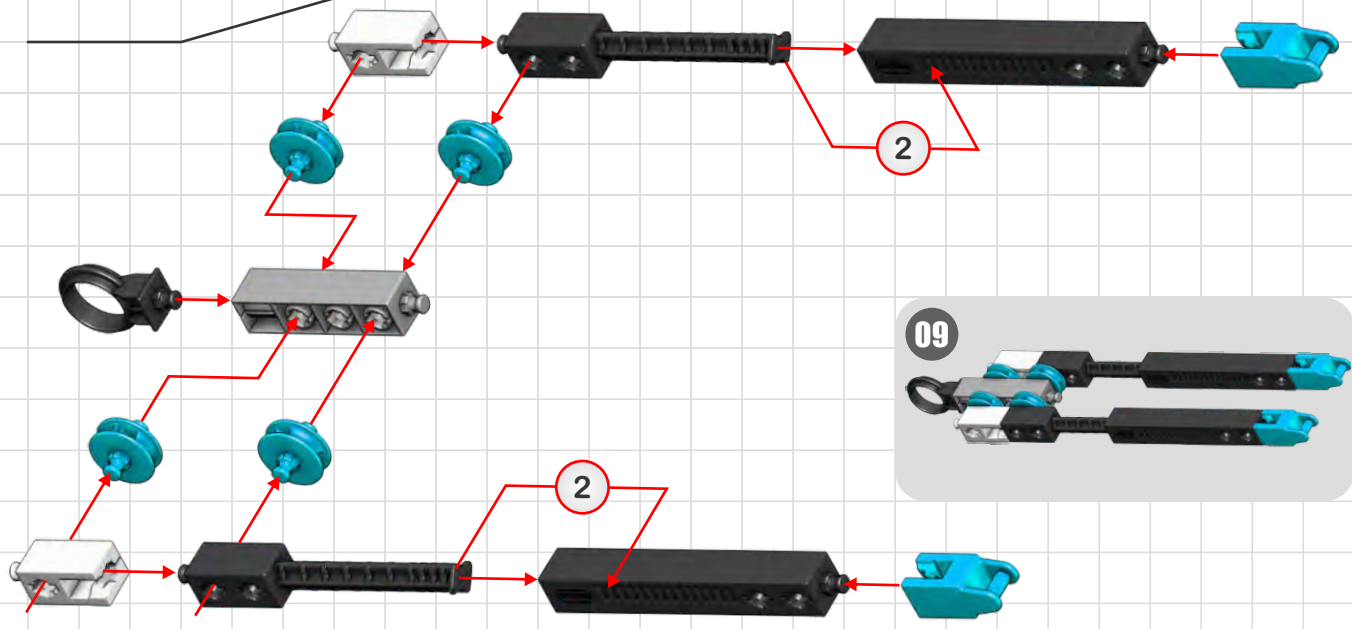
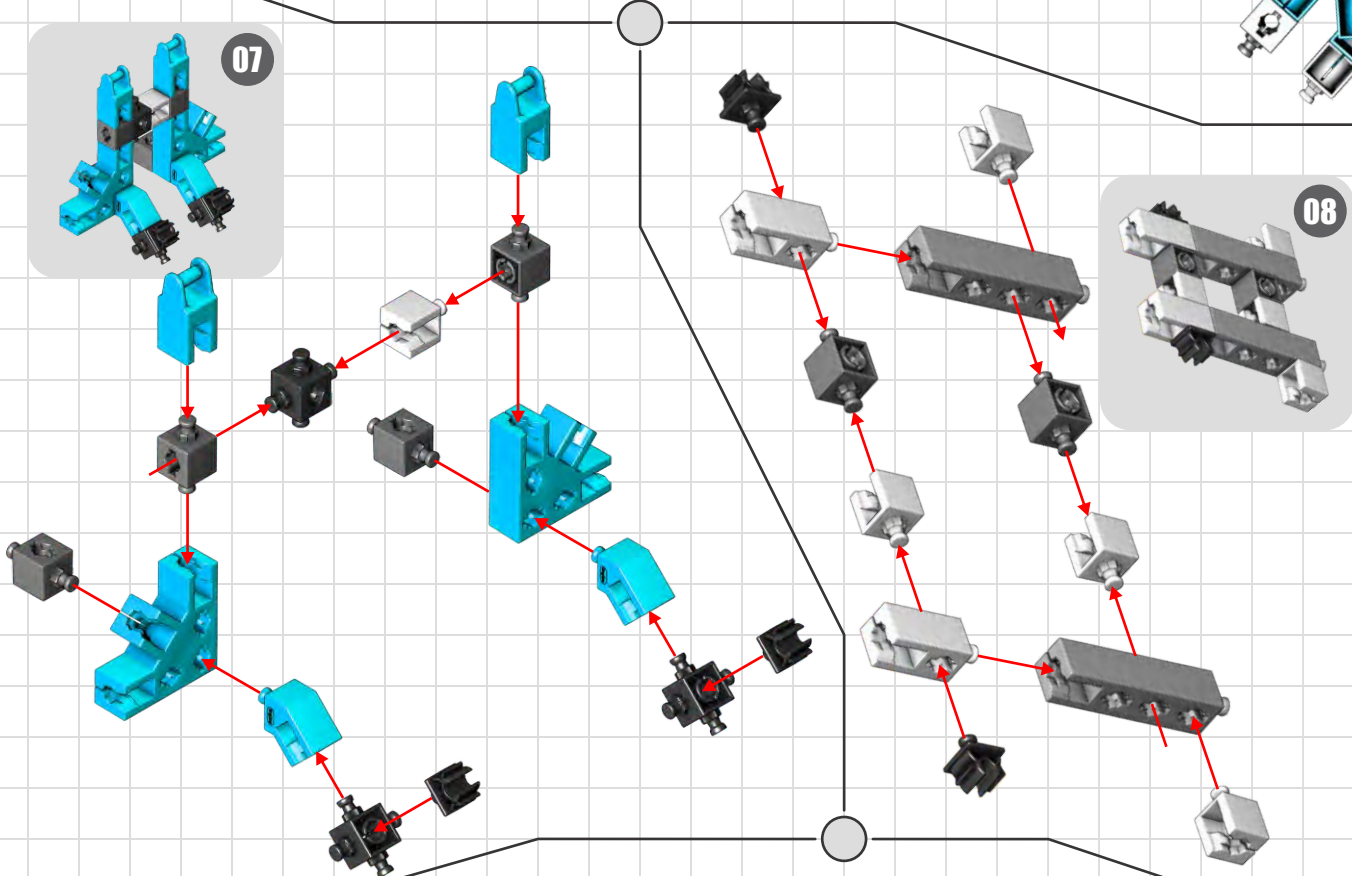
06



07

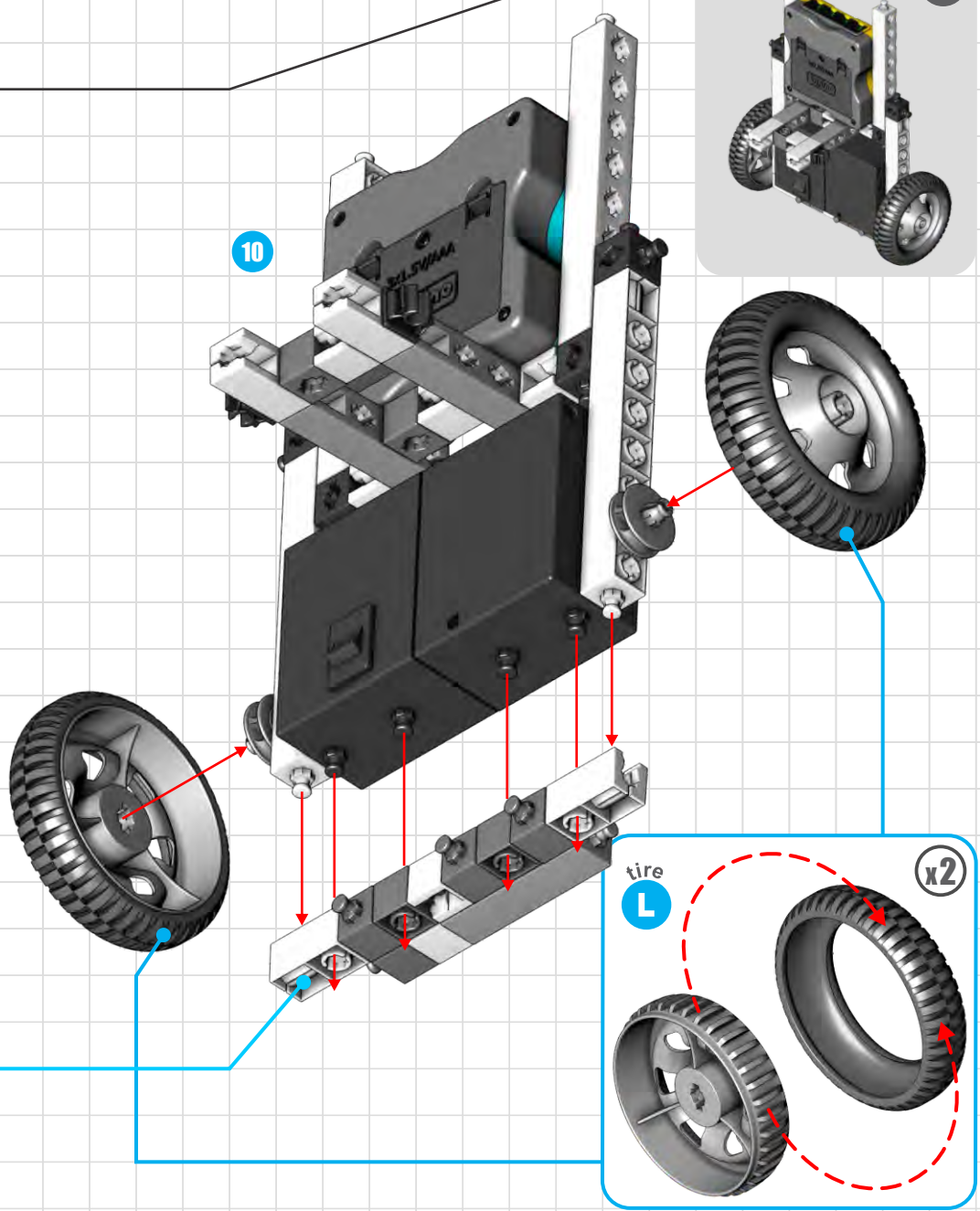
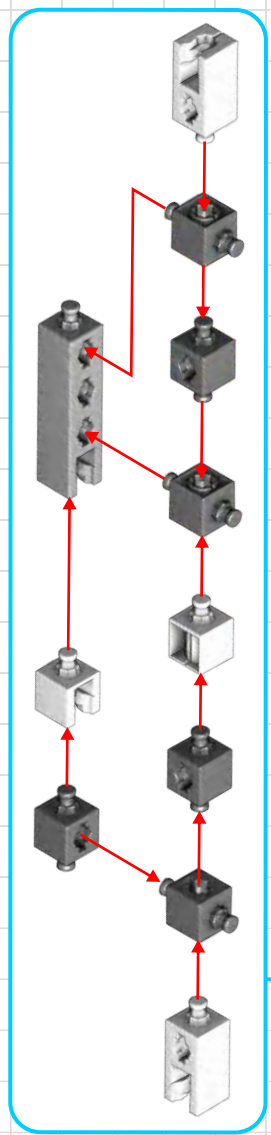
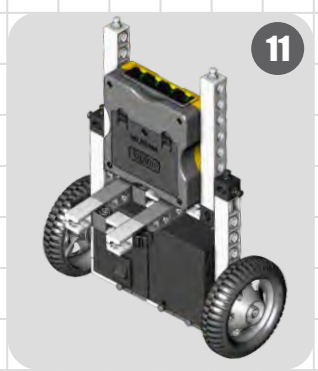
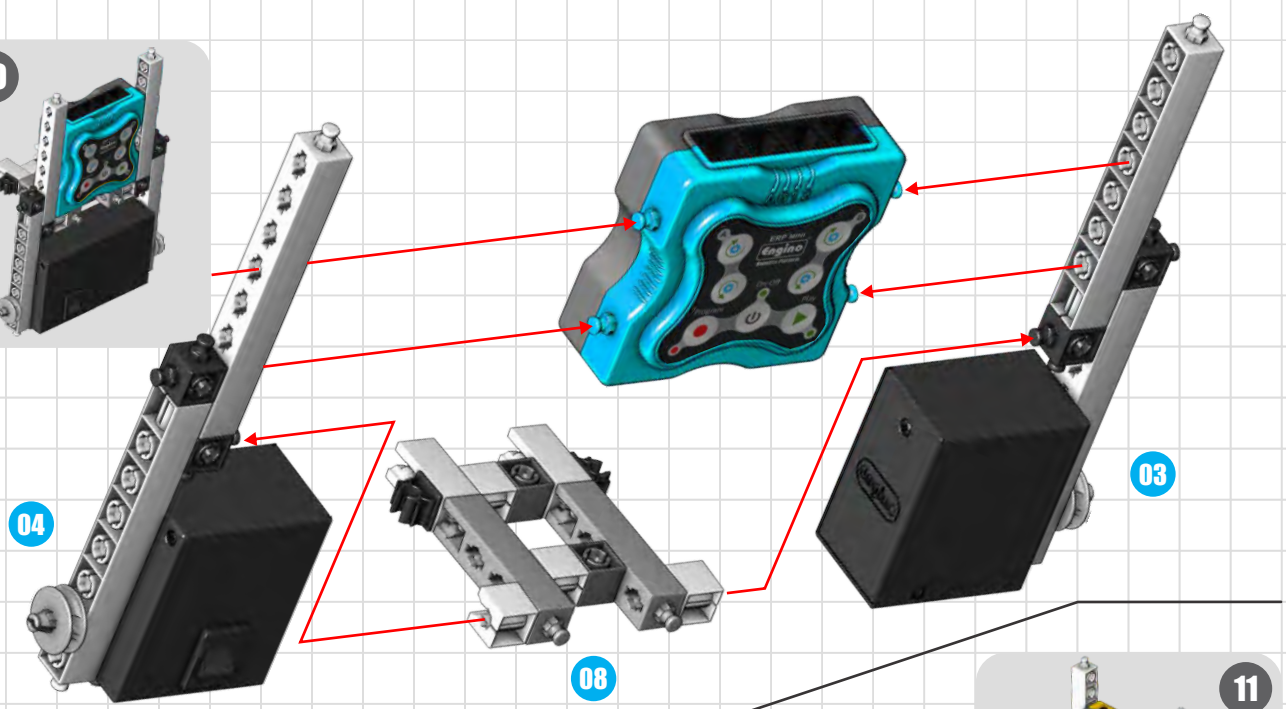


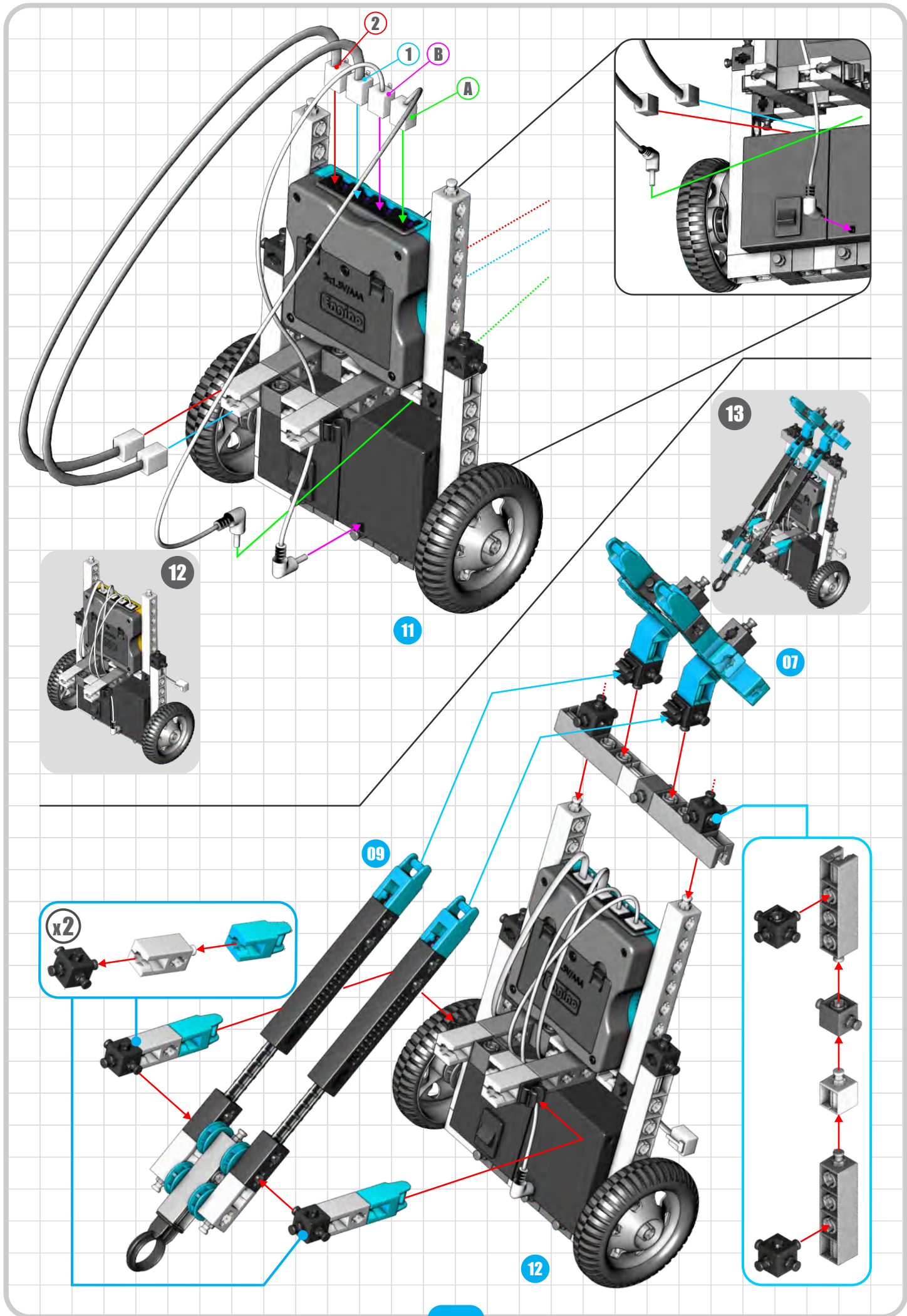
08

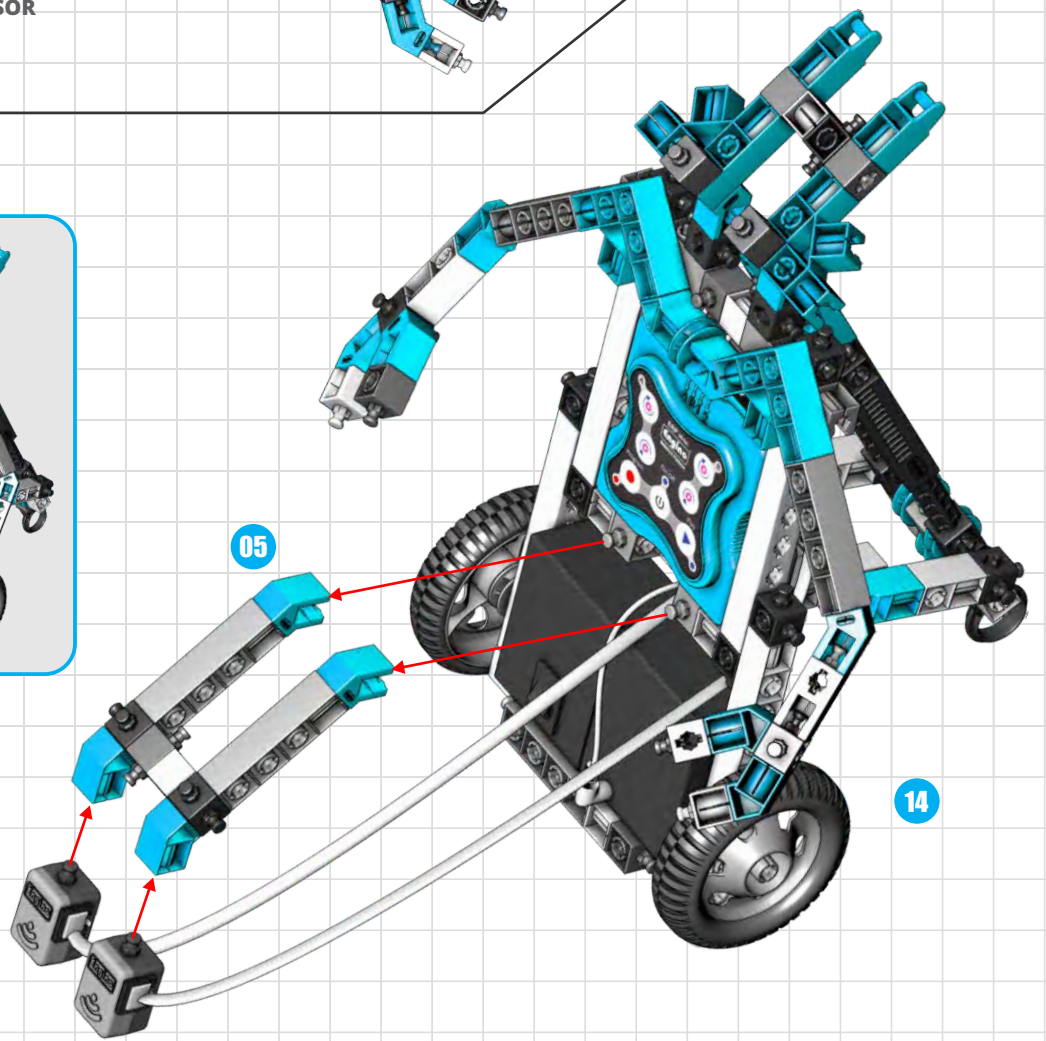
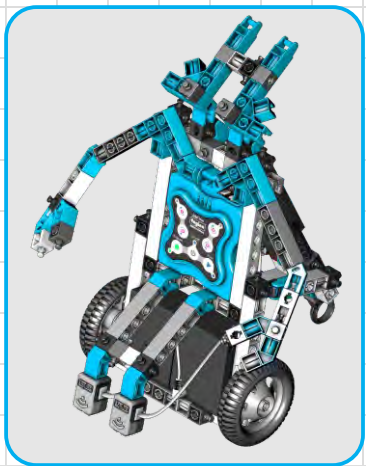
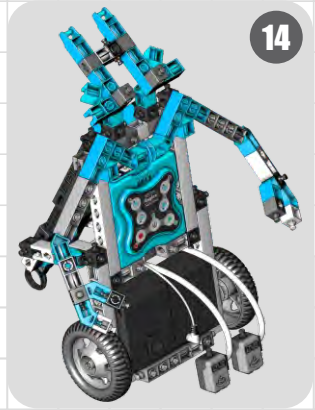
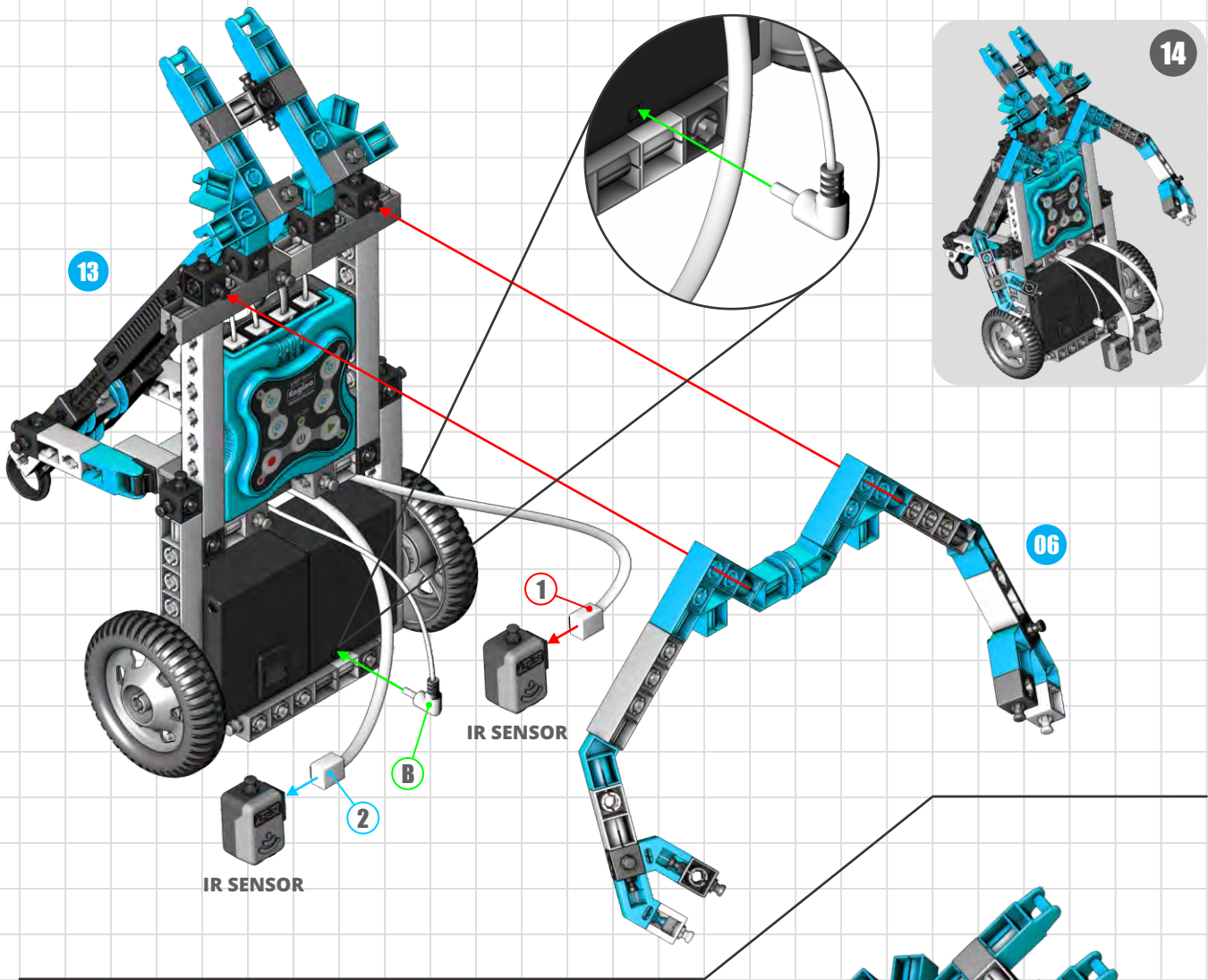


09





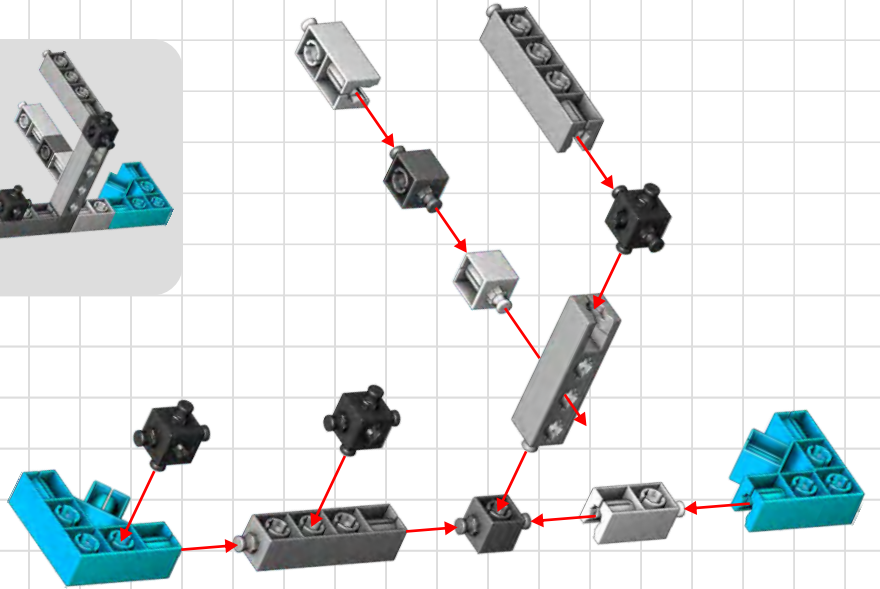
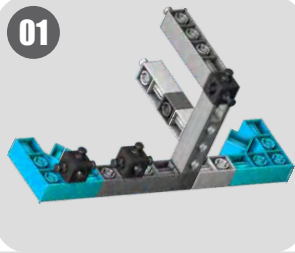




Větrák

4020

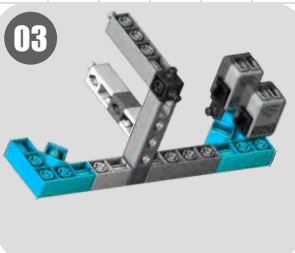
01



02

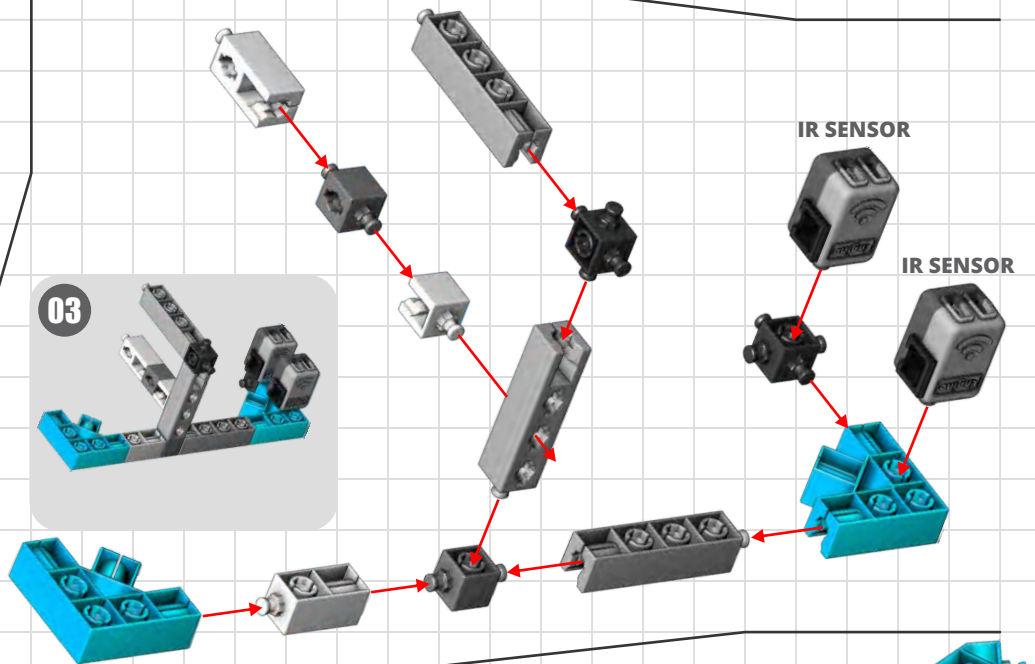


03

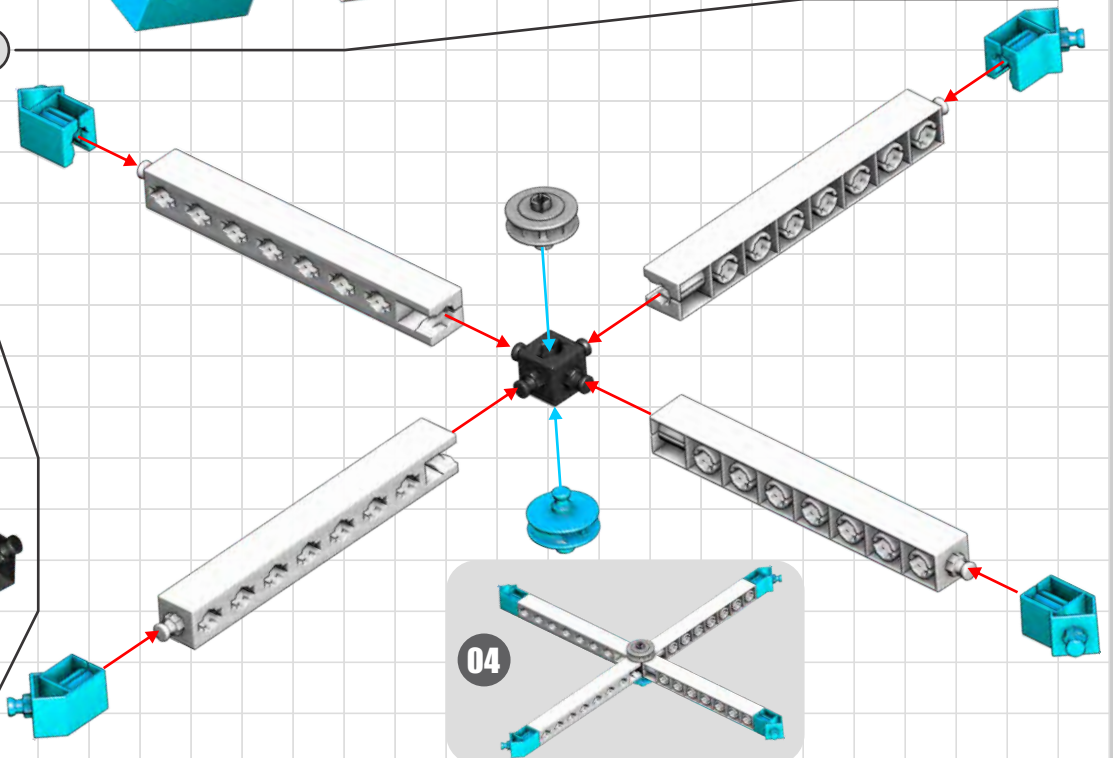
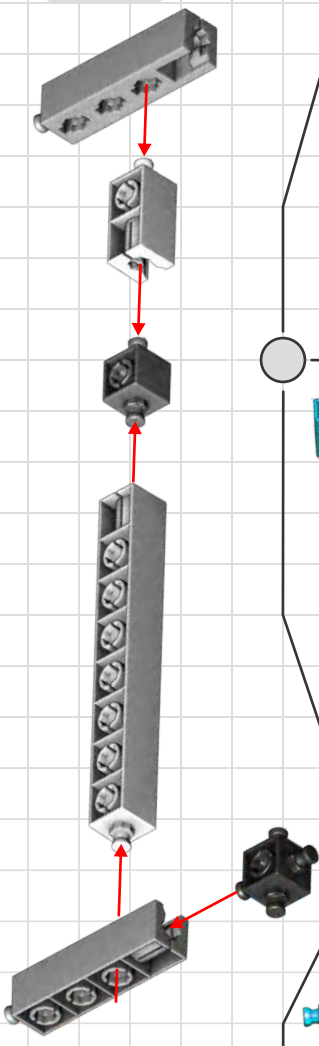
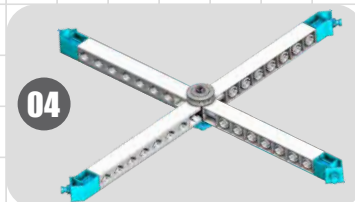


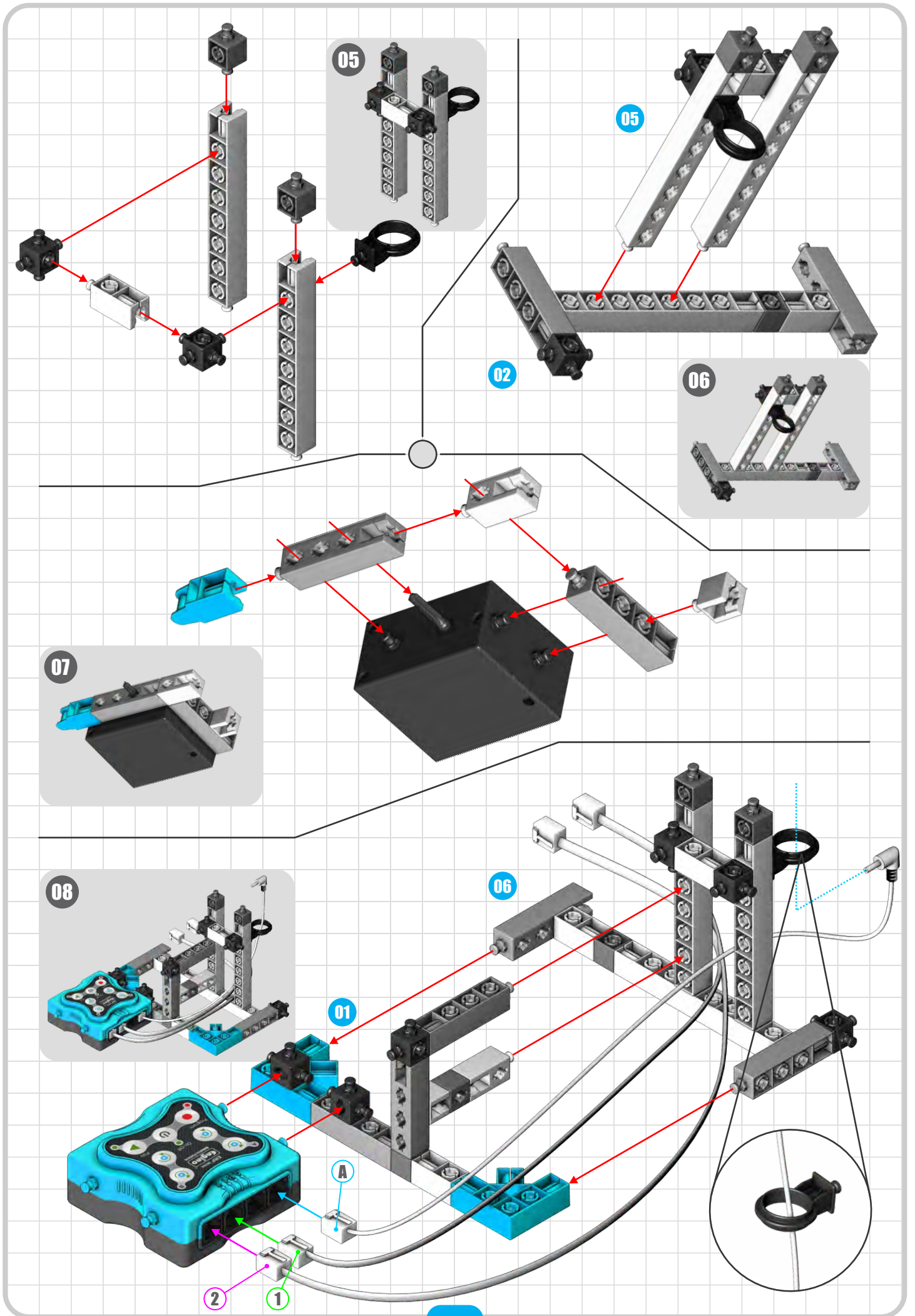
IR SENSOR

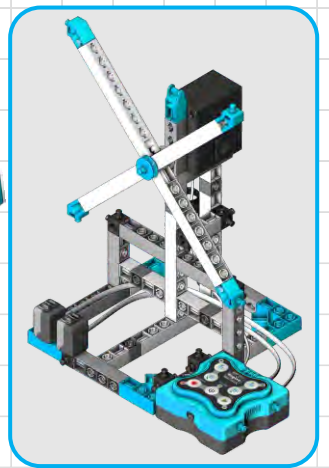
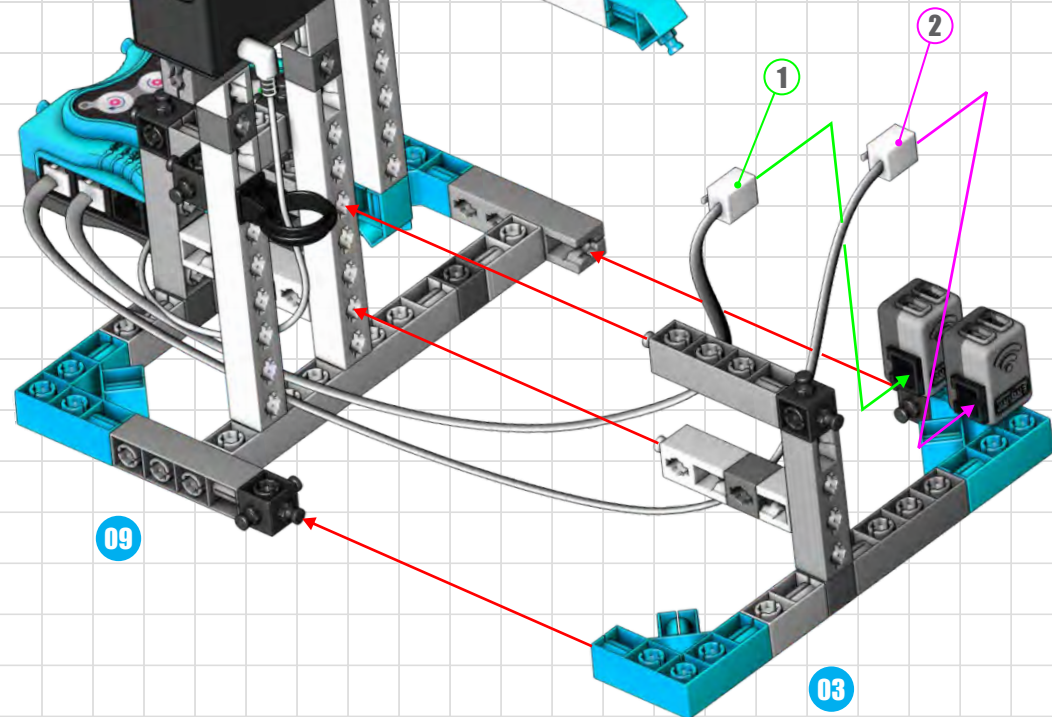
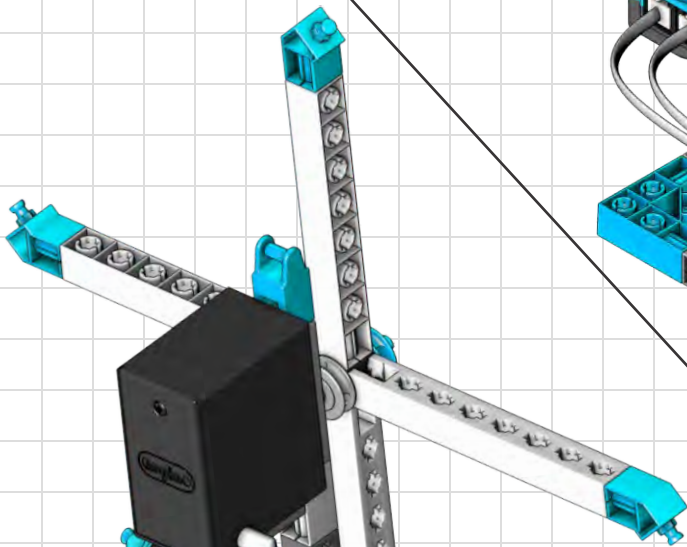
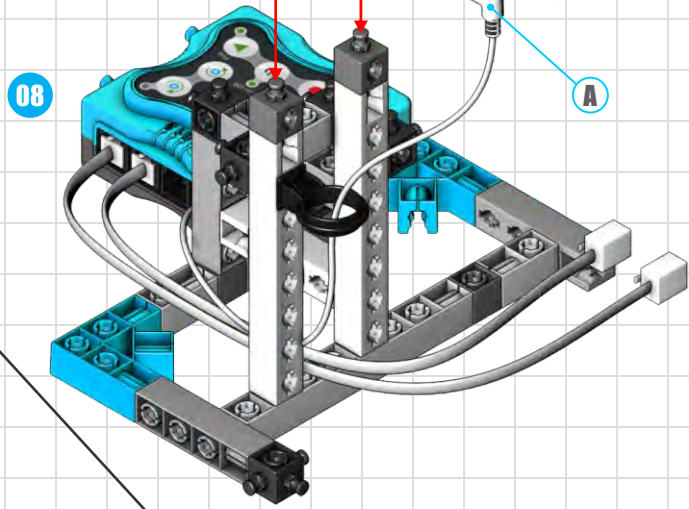
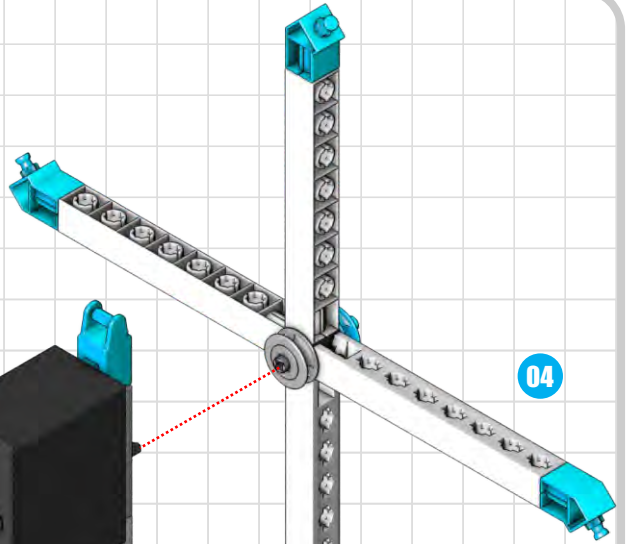
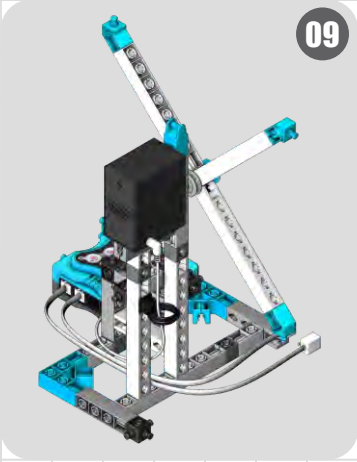
IR SENSOR



04



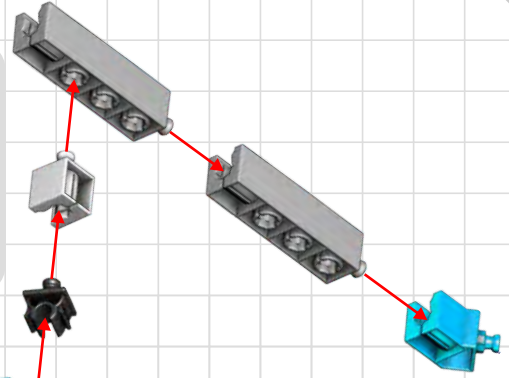
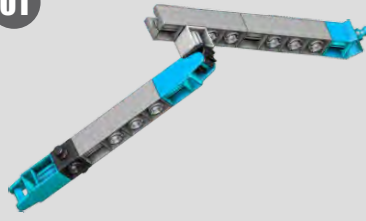




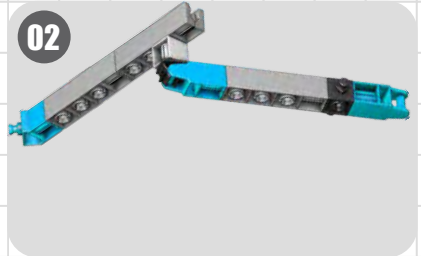
# Hexapod

4021

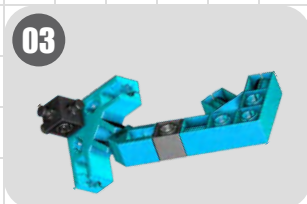
01



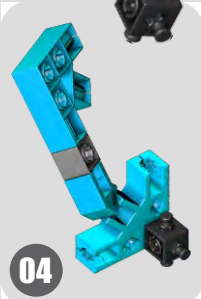
02



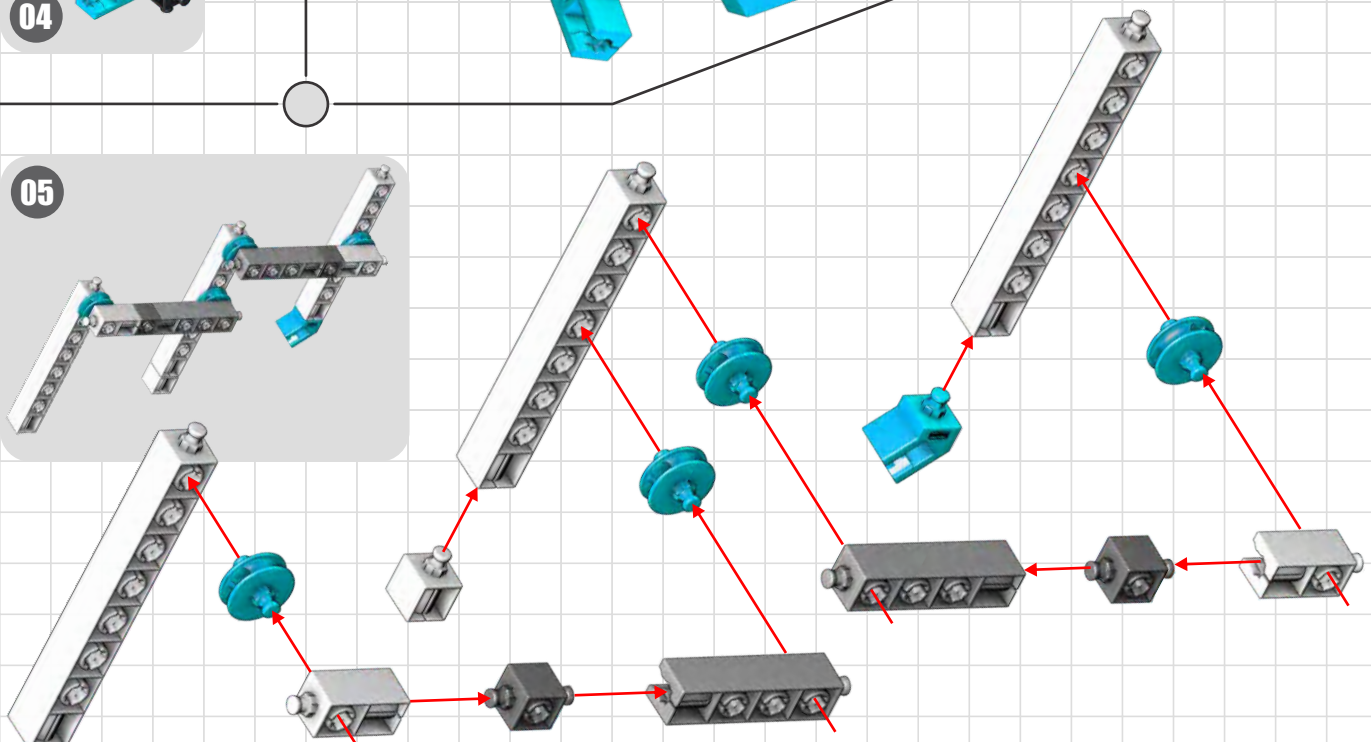
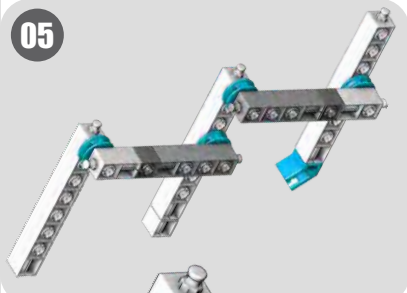
03

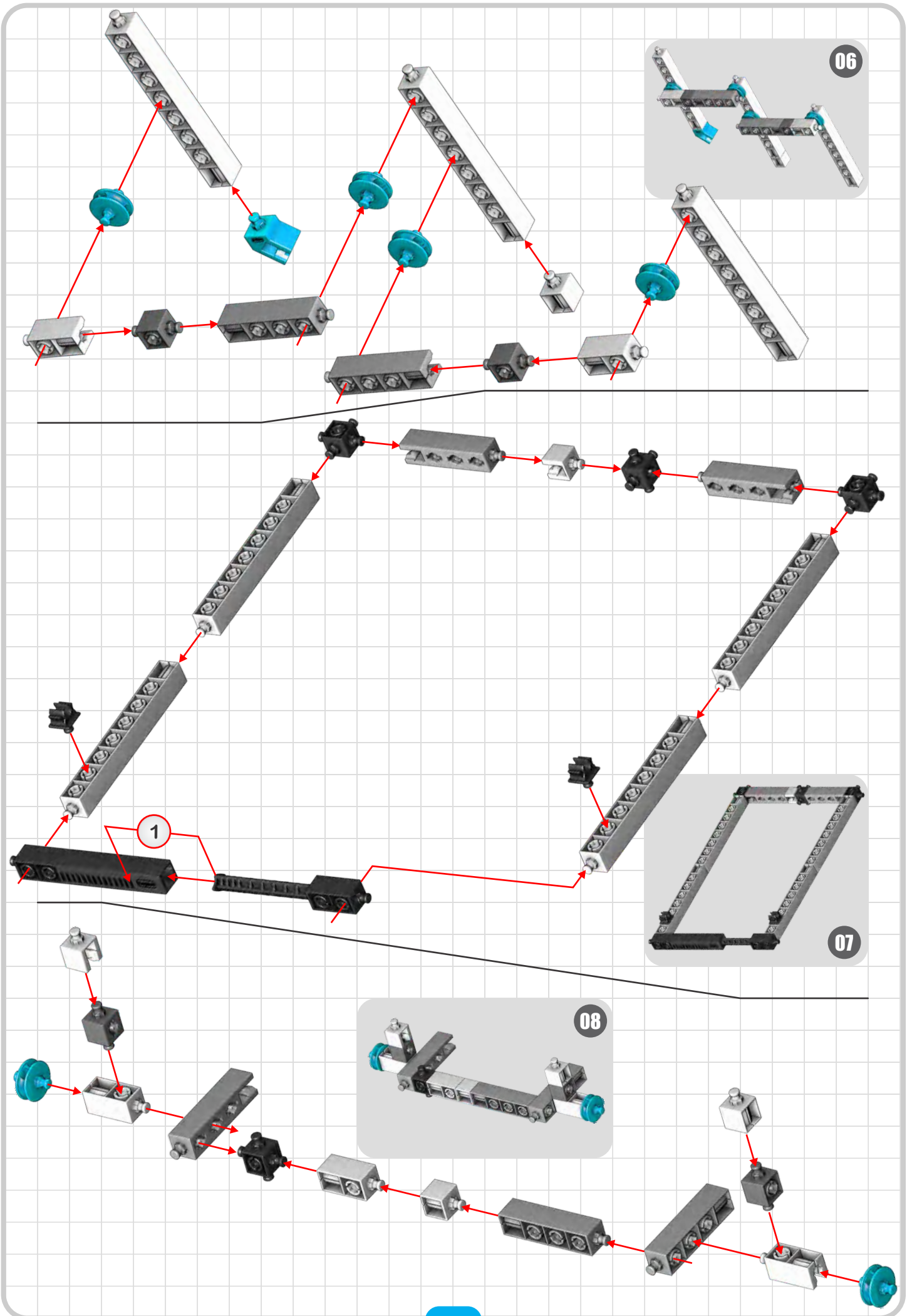


04



05





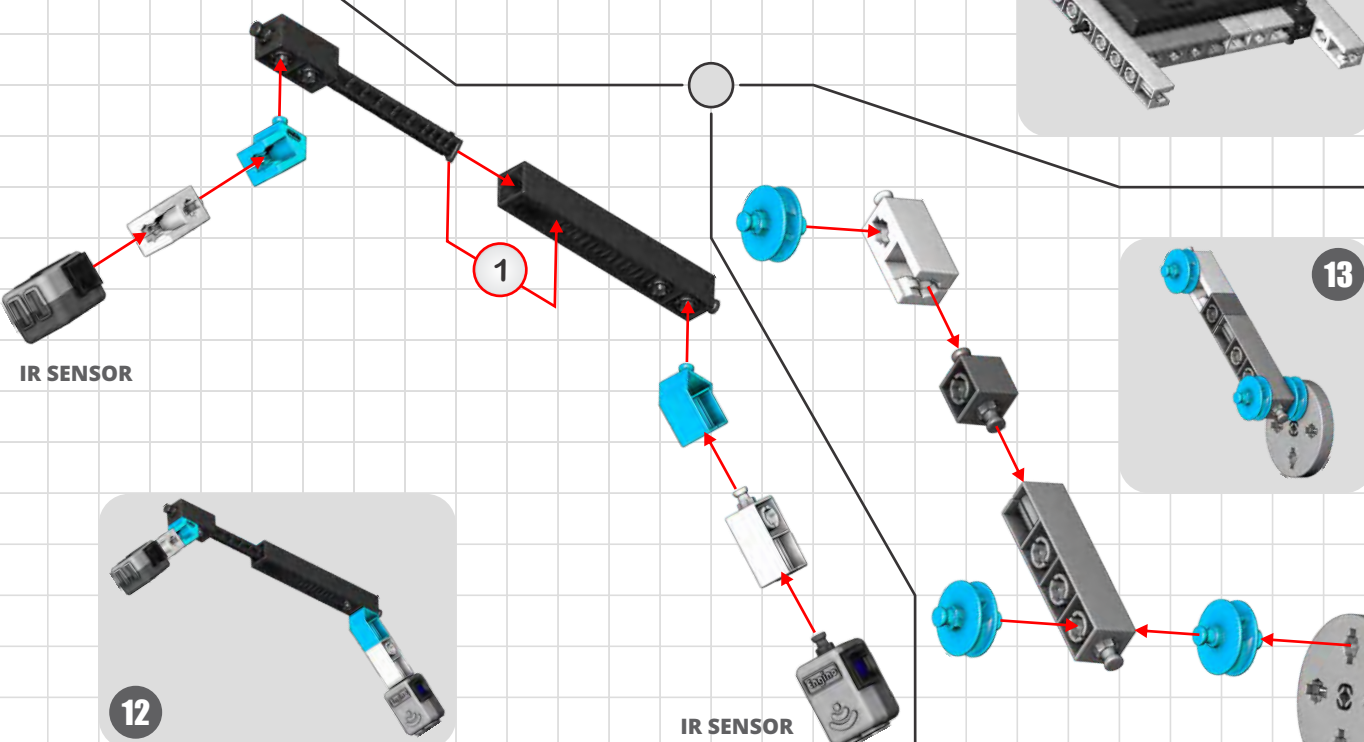
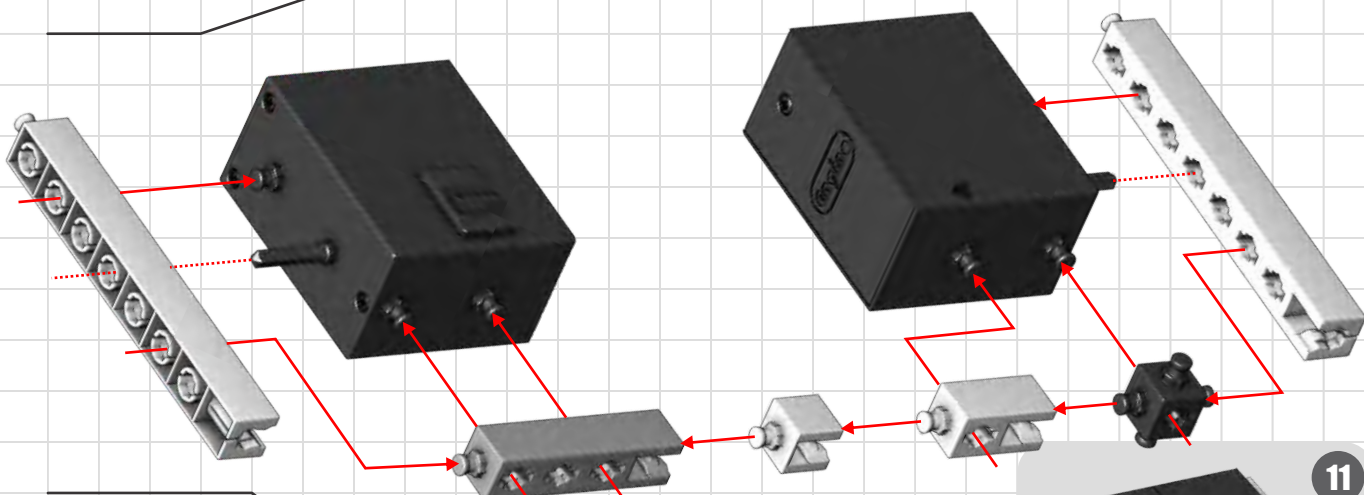
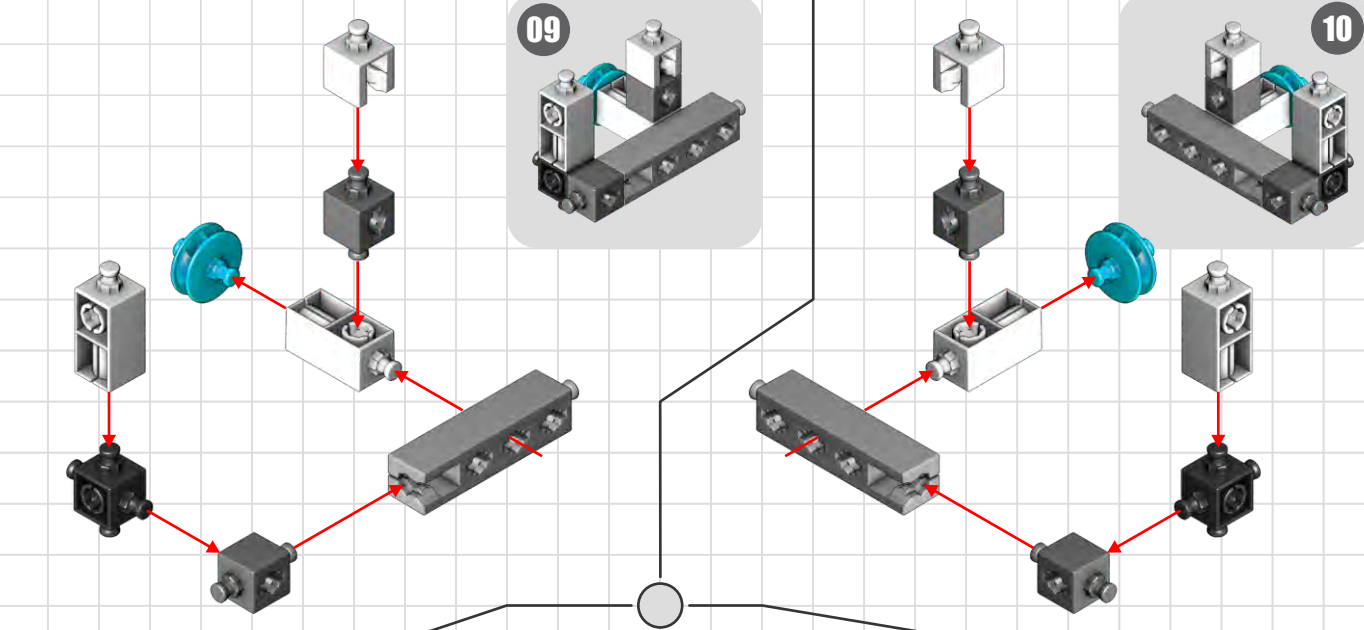
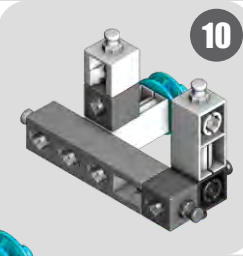
06

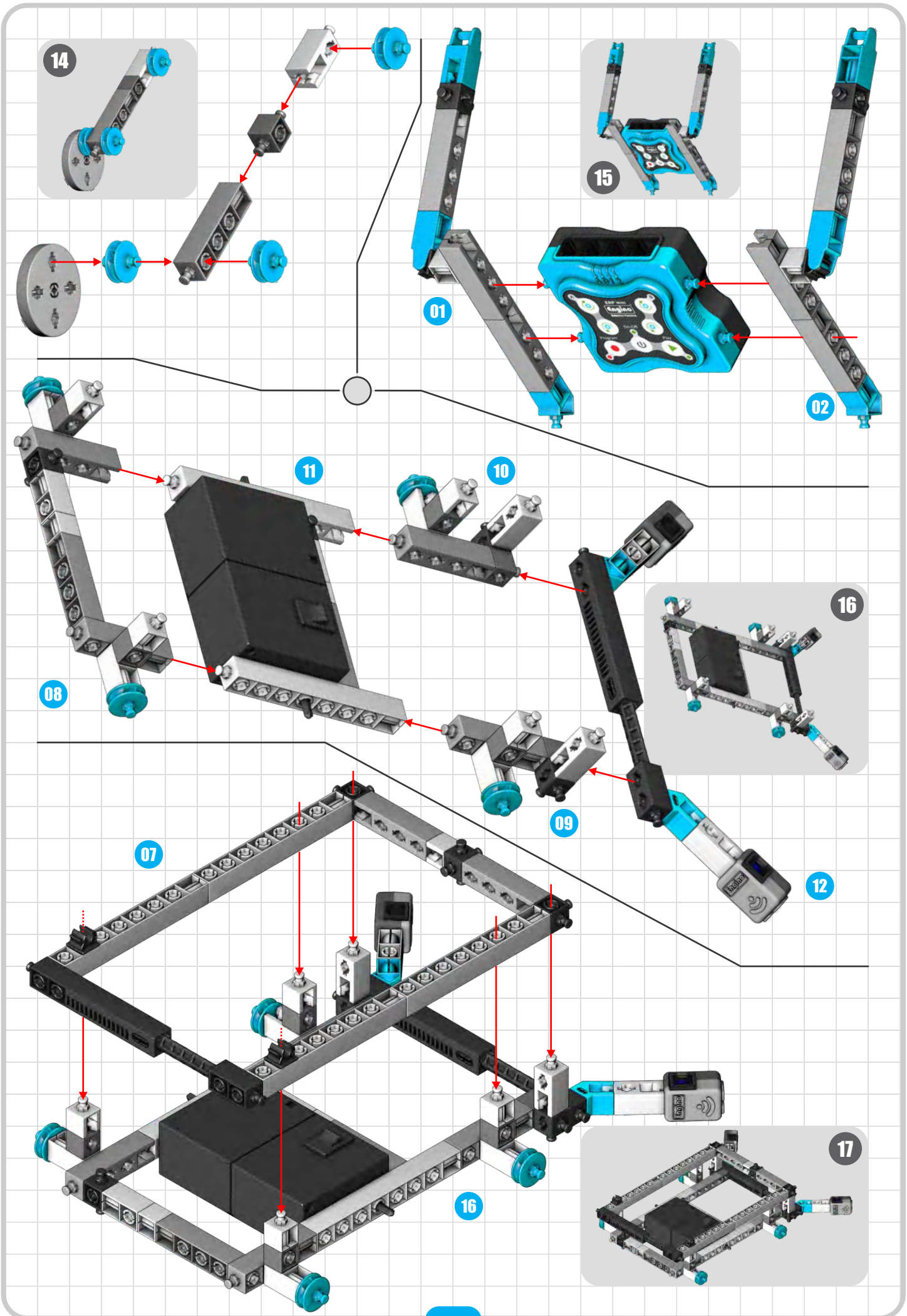
07

08

1







14

15

16

17

01

02

11

10

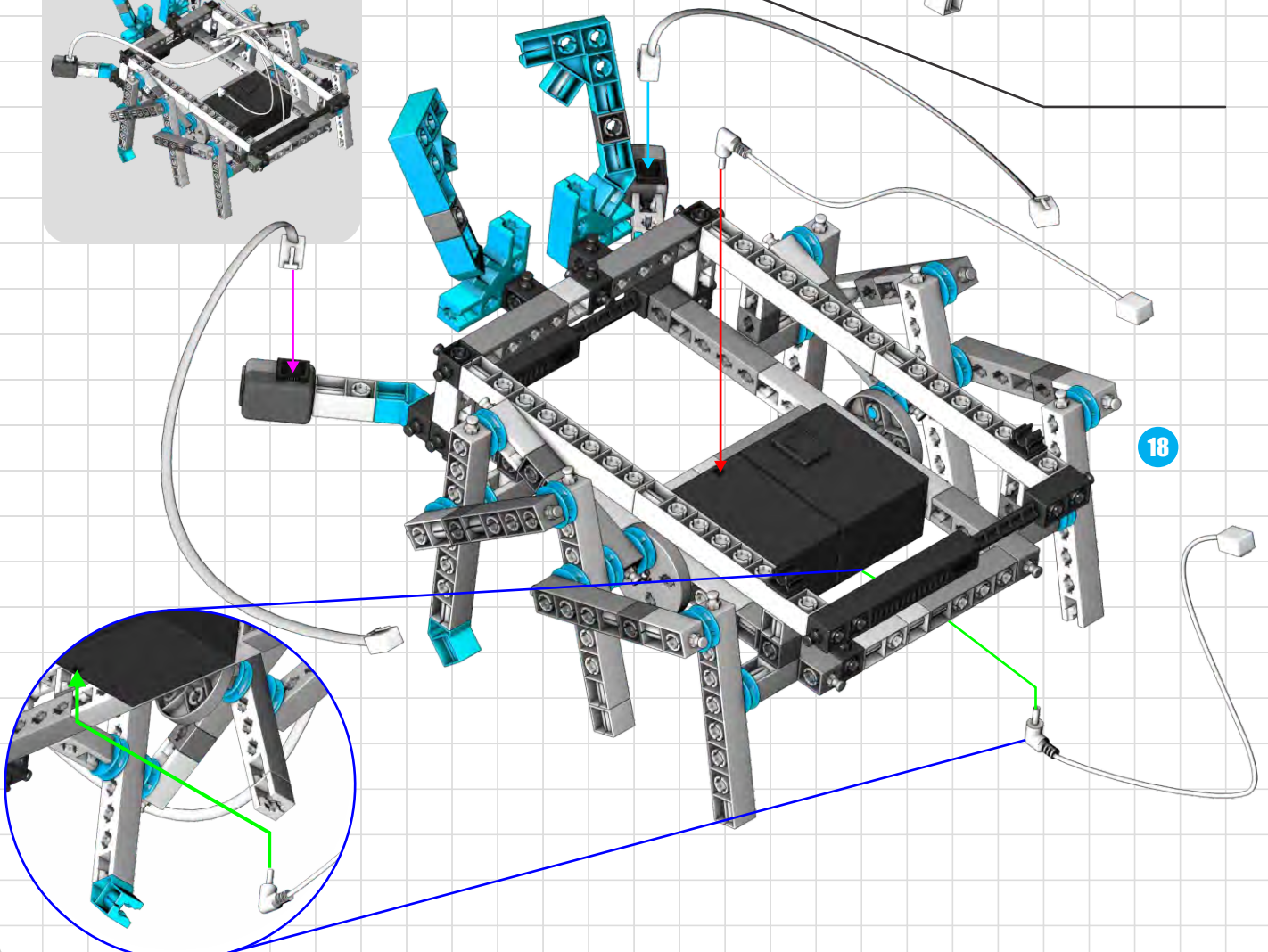
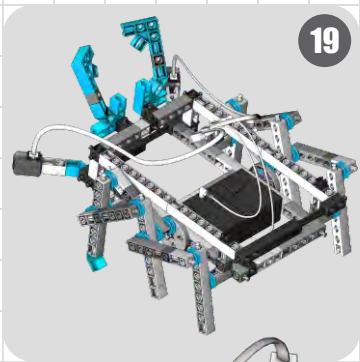
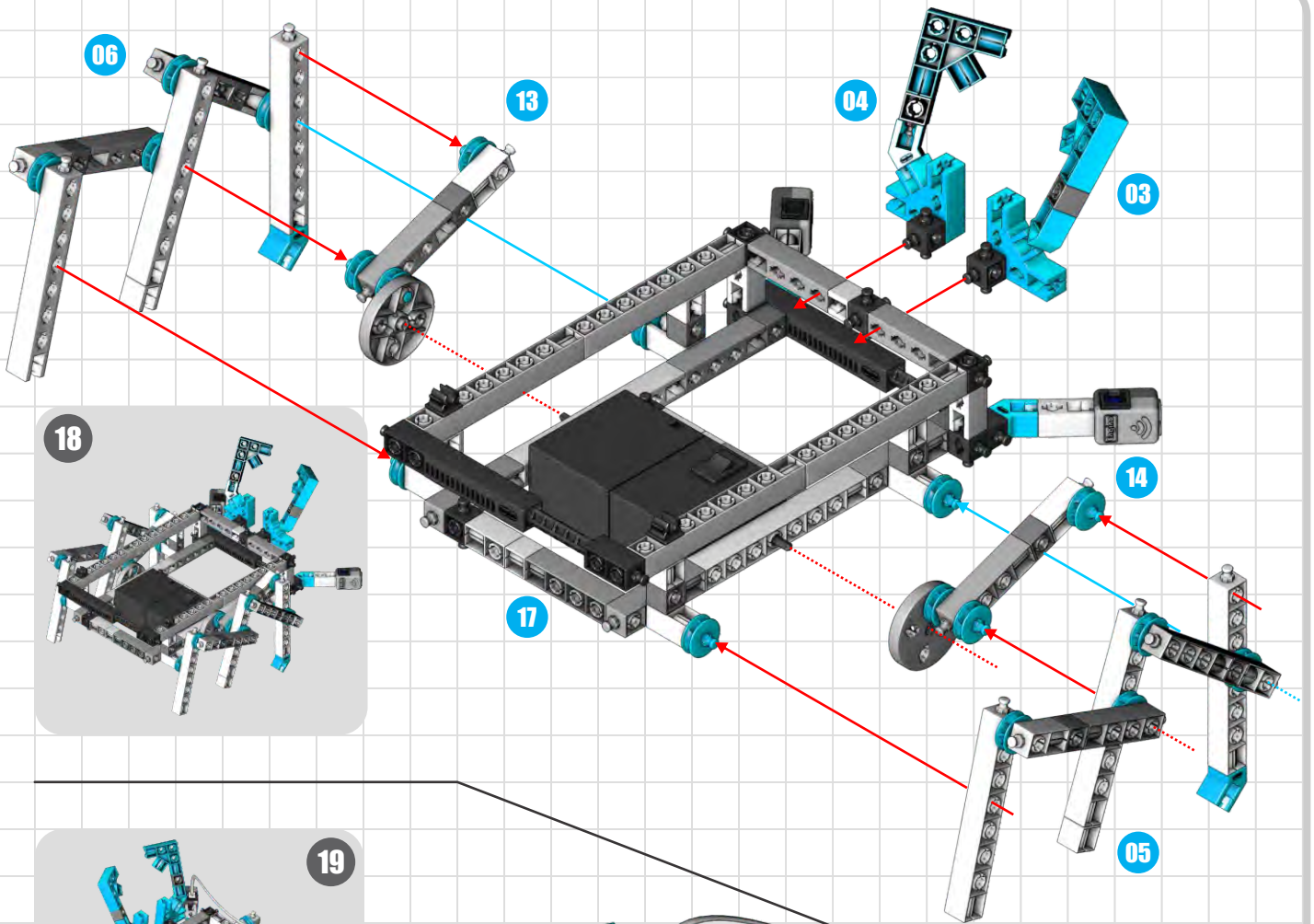
08

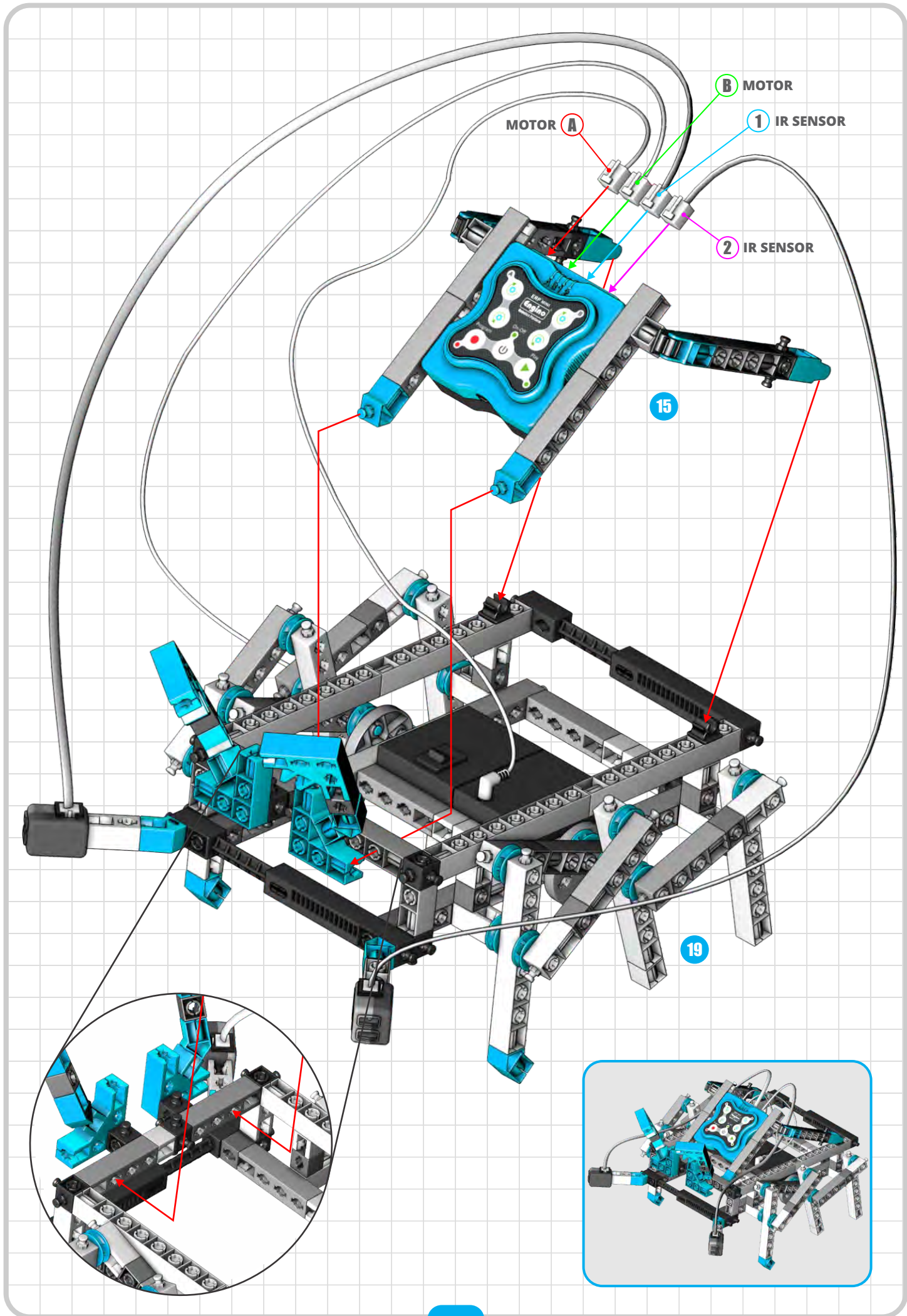
09

12

07

16





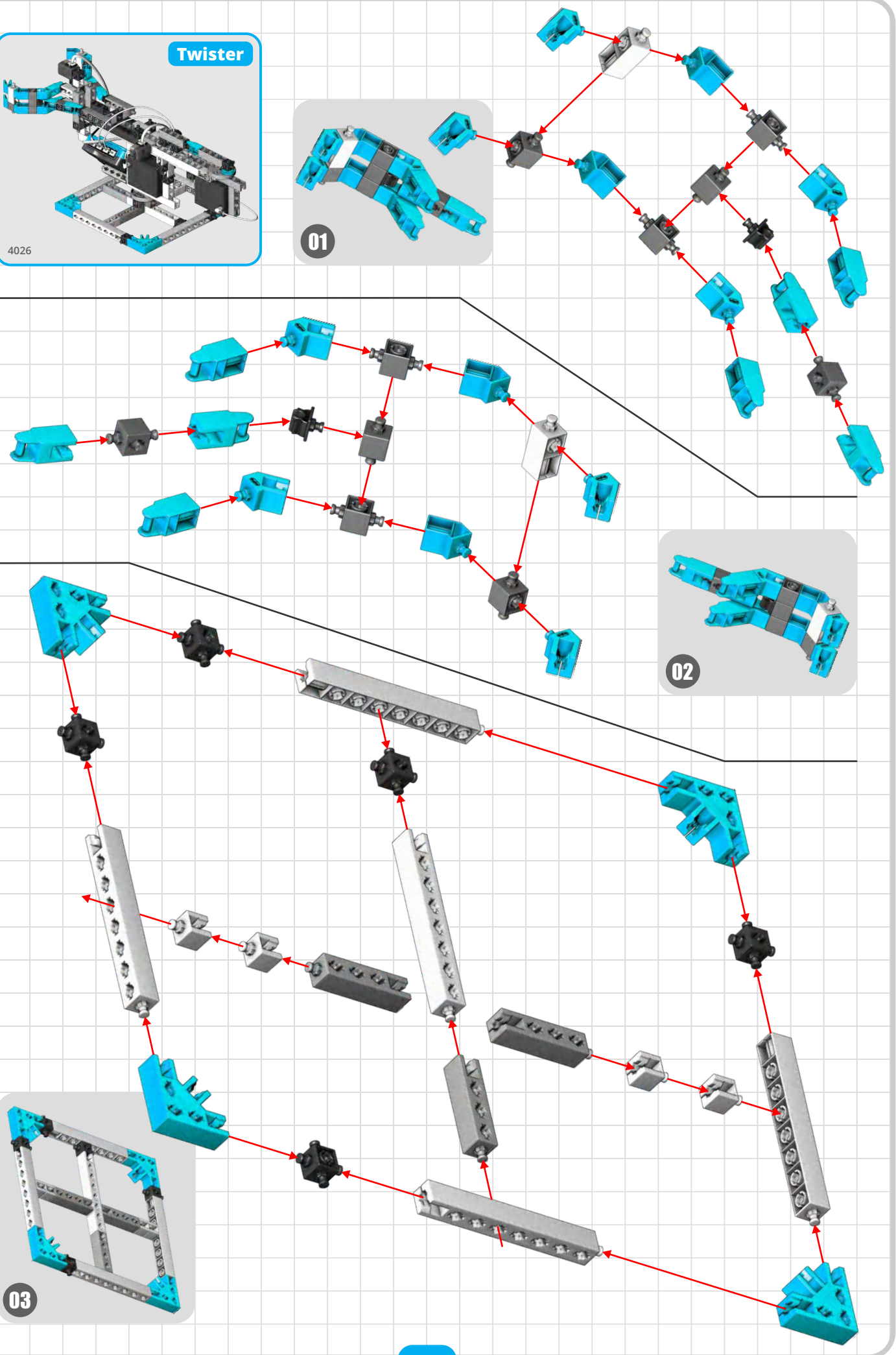
Twister

4026

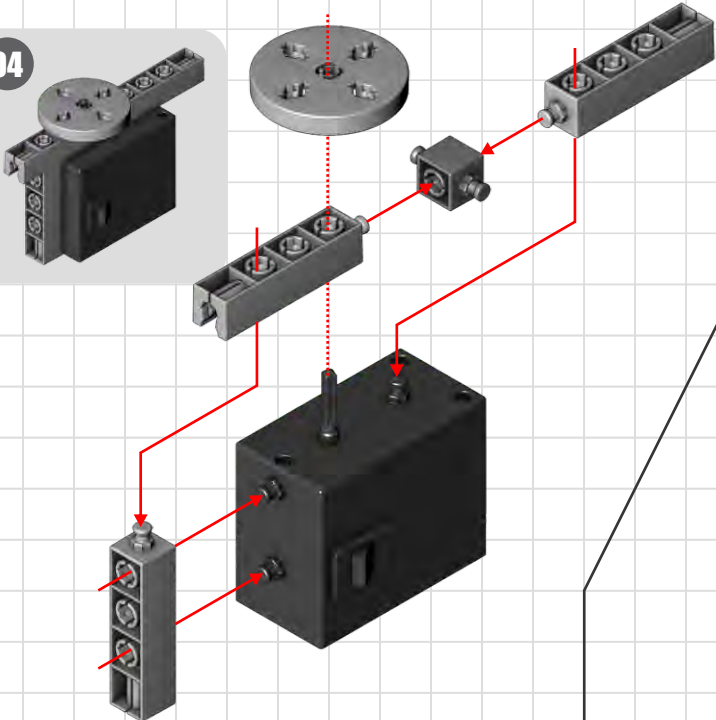
01

02

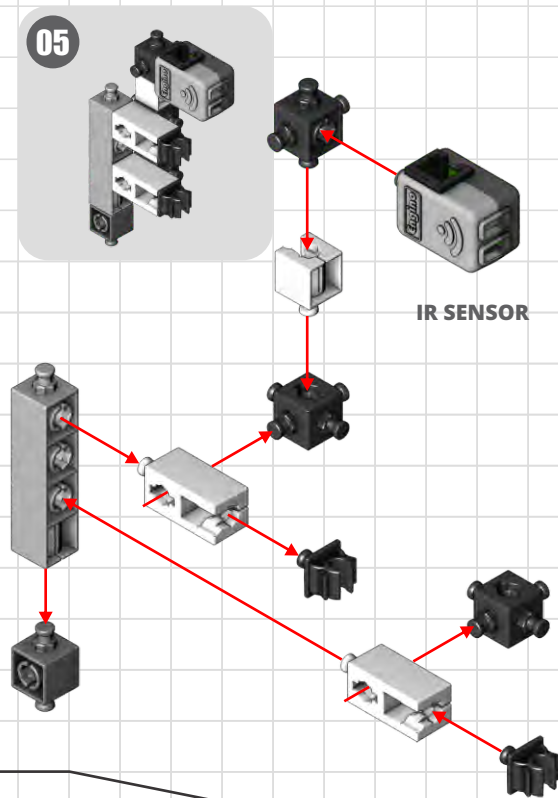
03



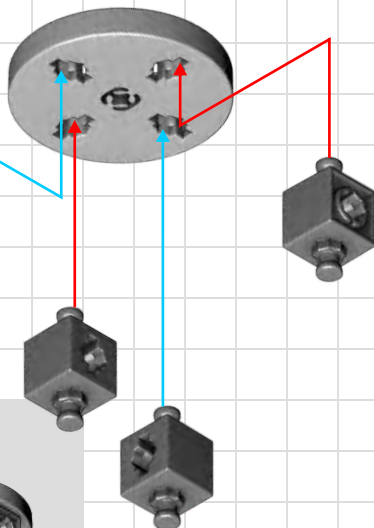
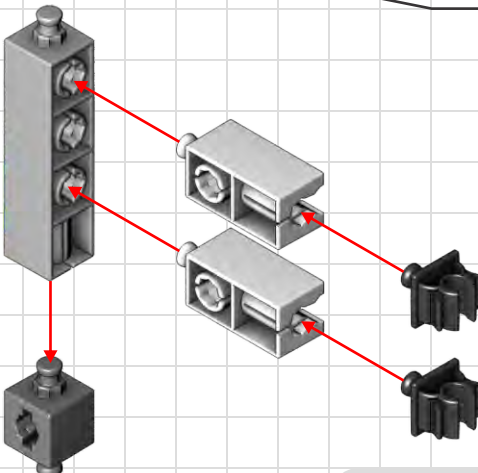
04



05



IR SENSOR



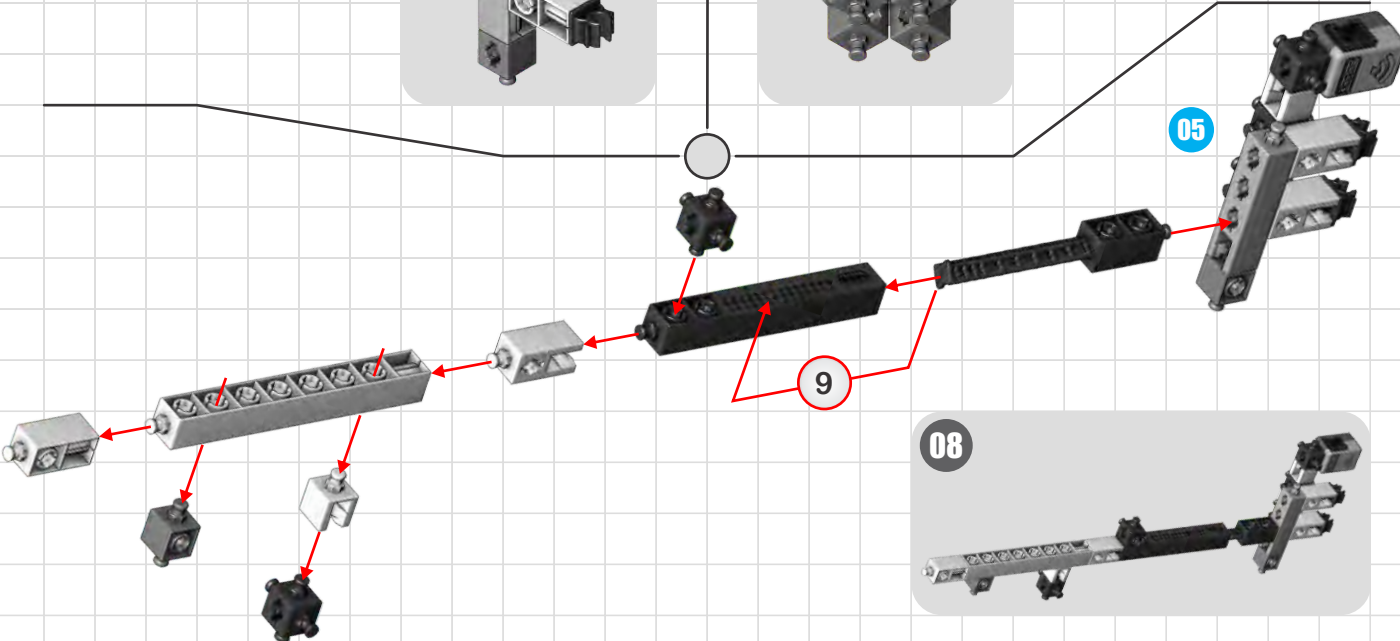
06



07

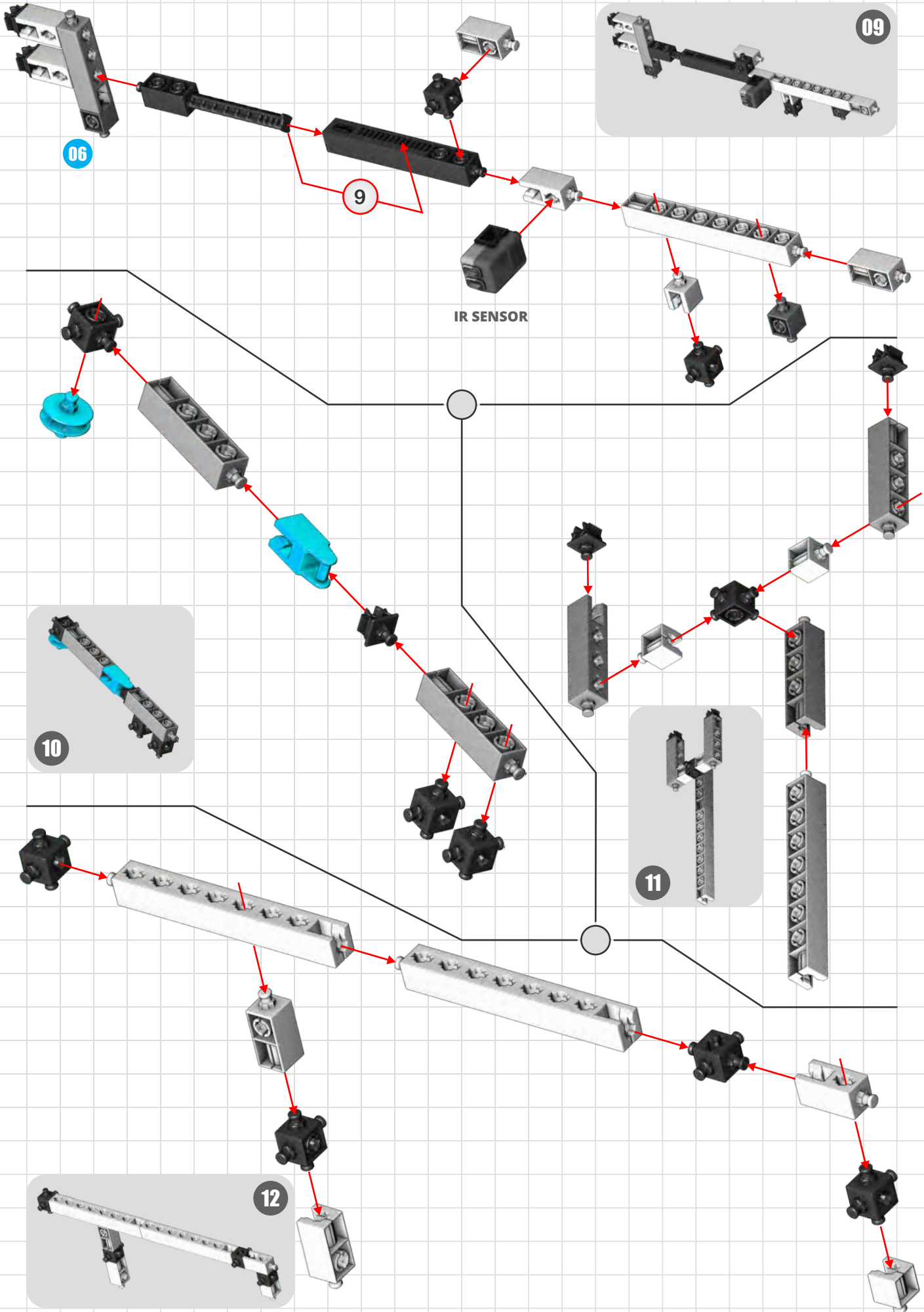


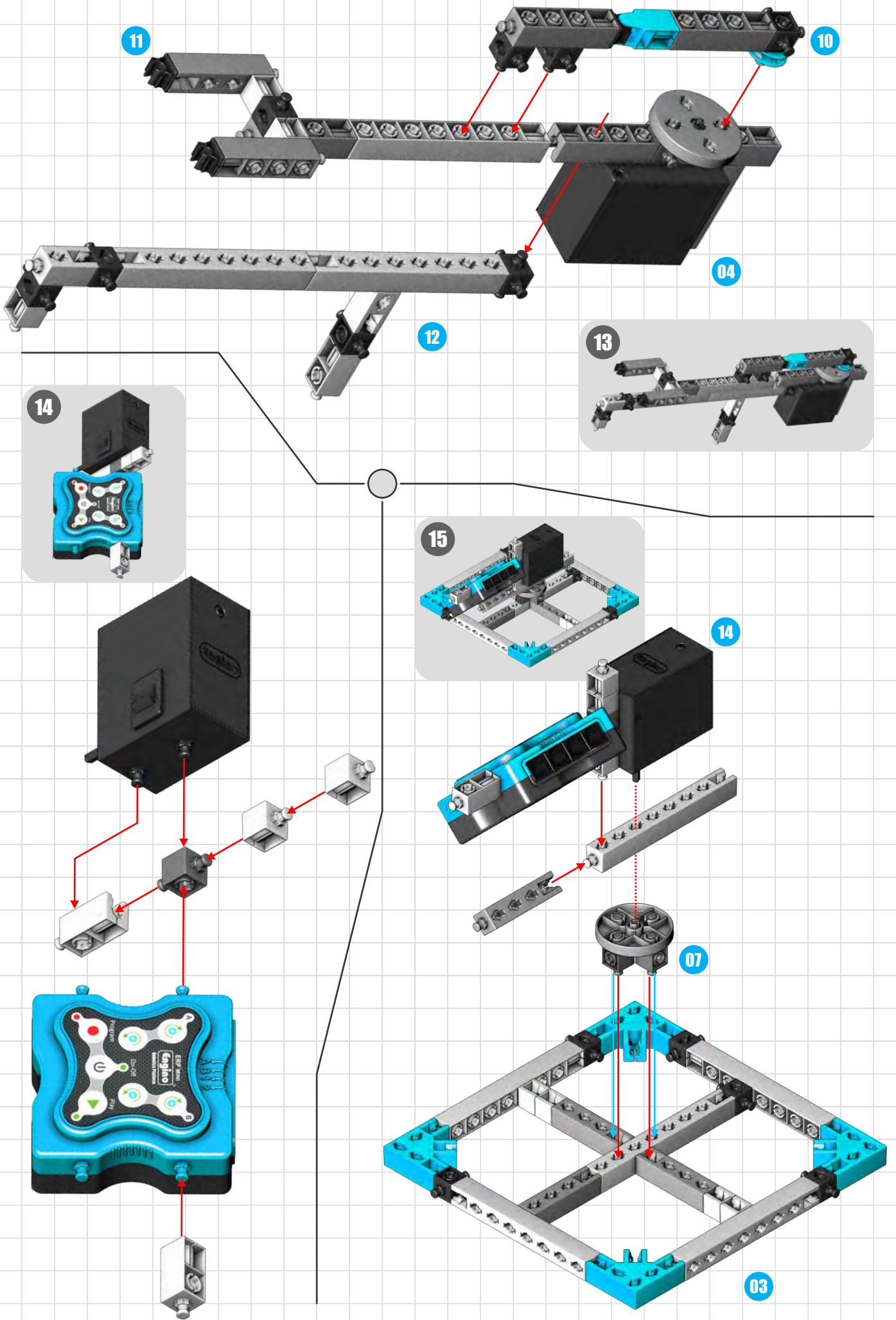
05



9

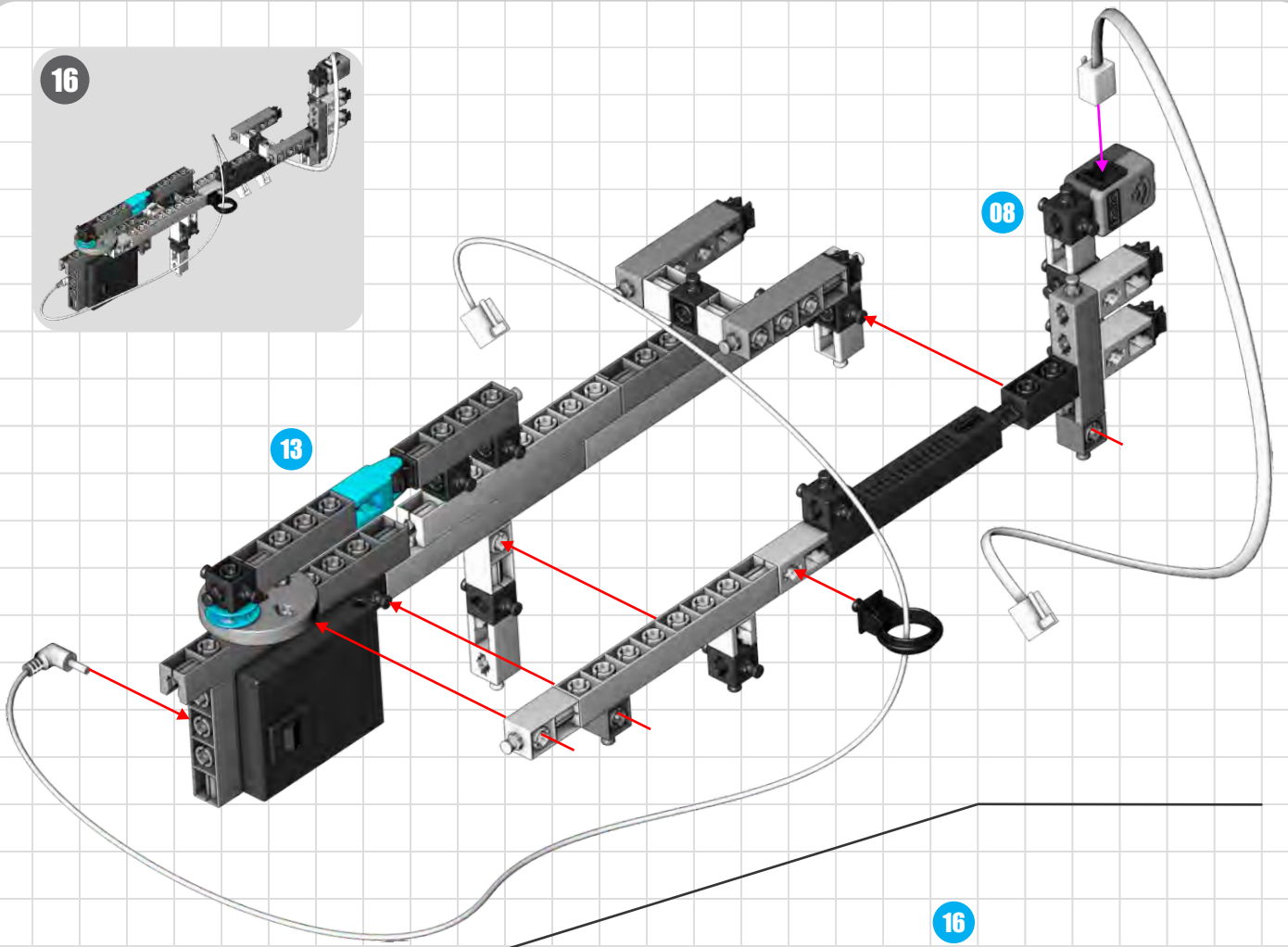
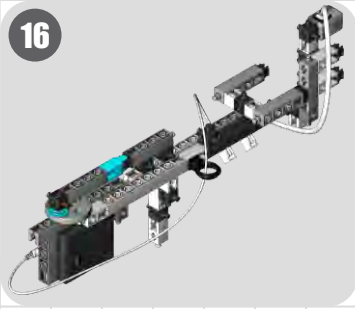
08



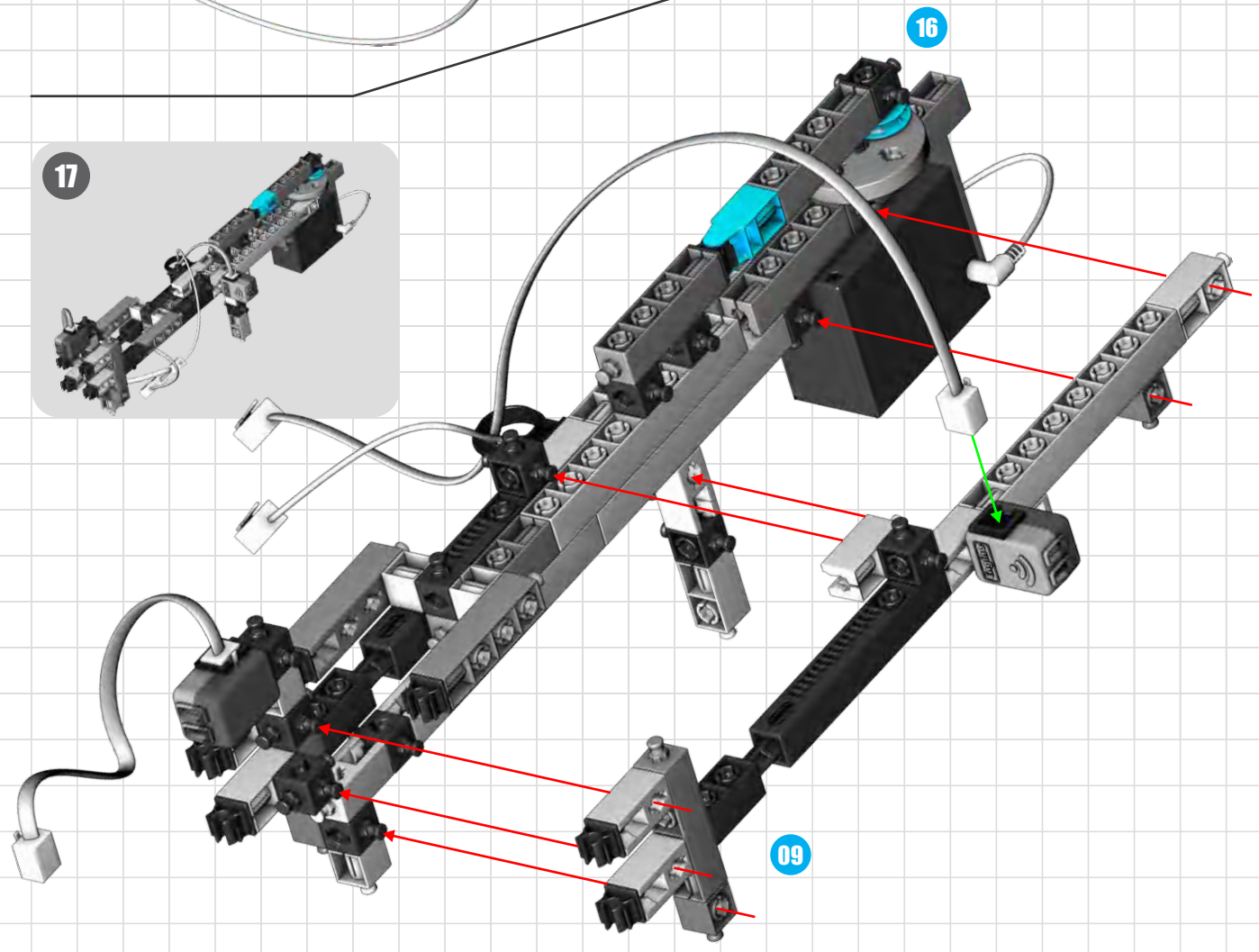
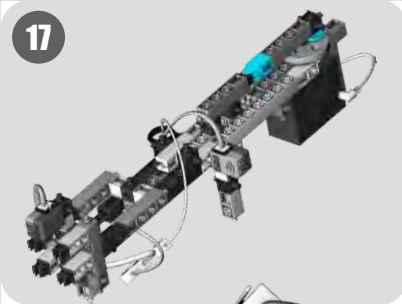


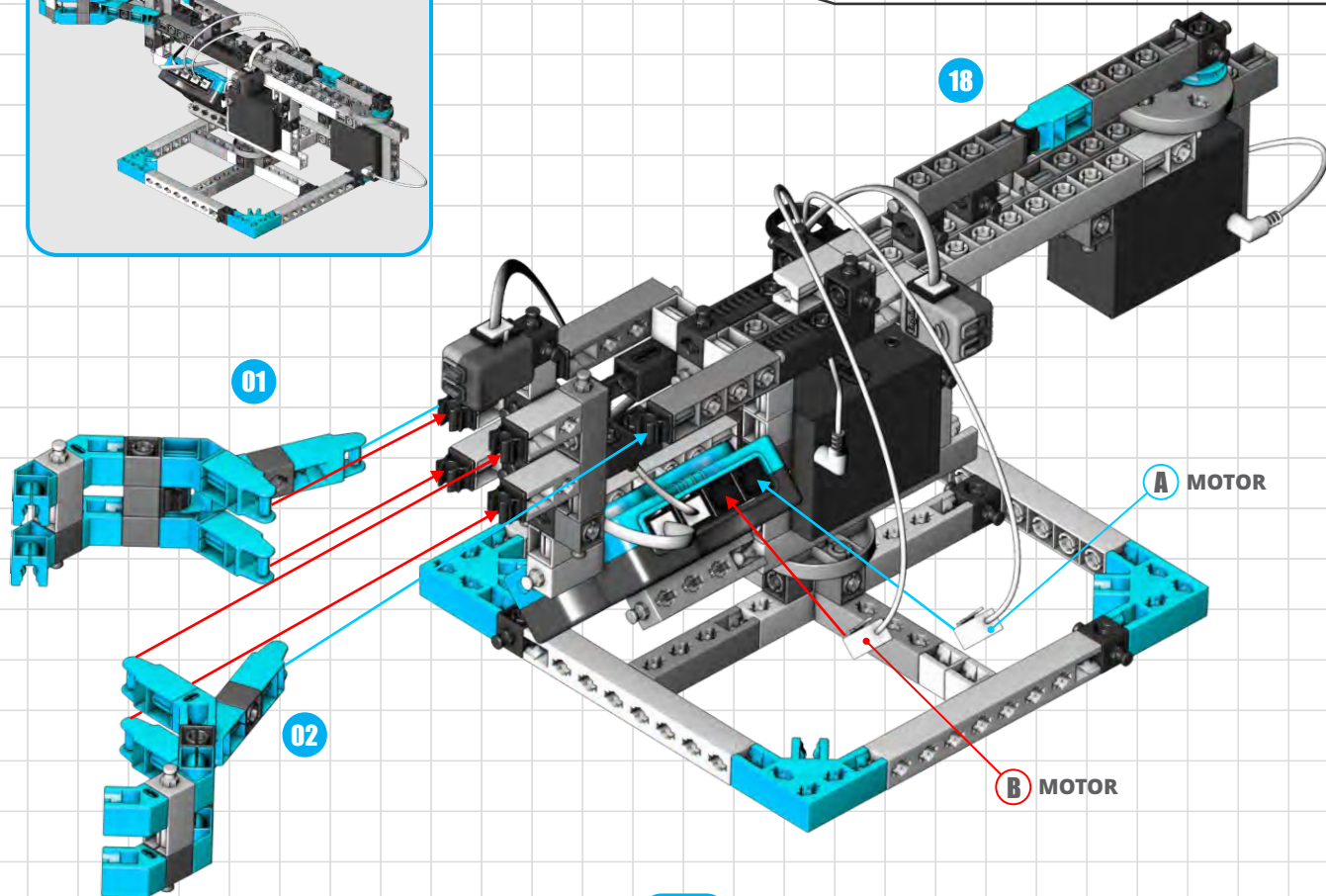
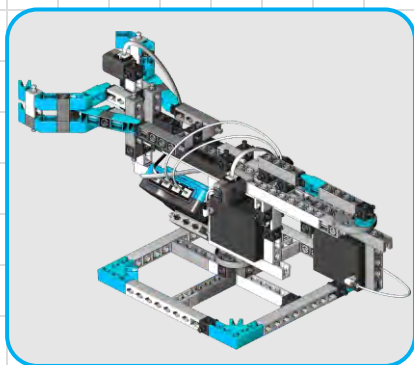
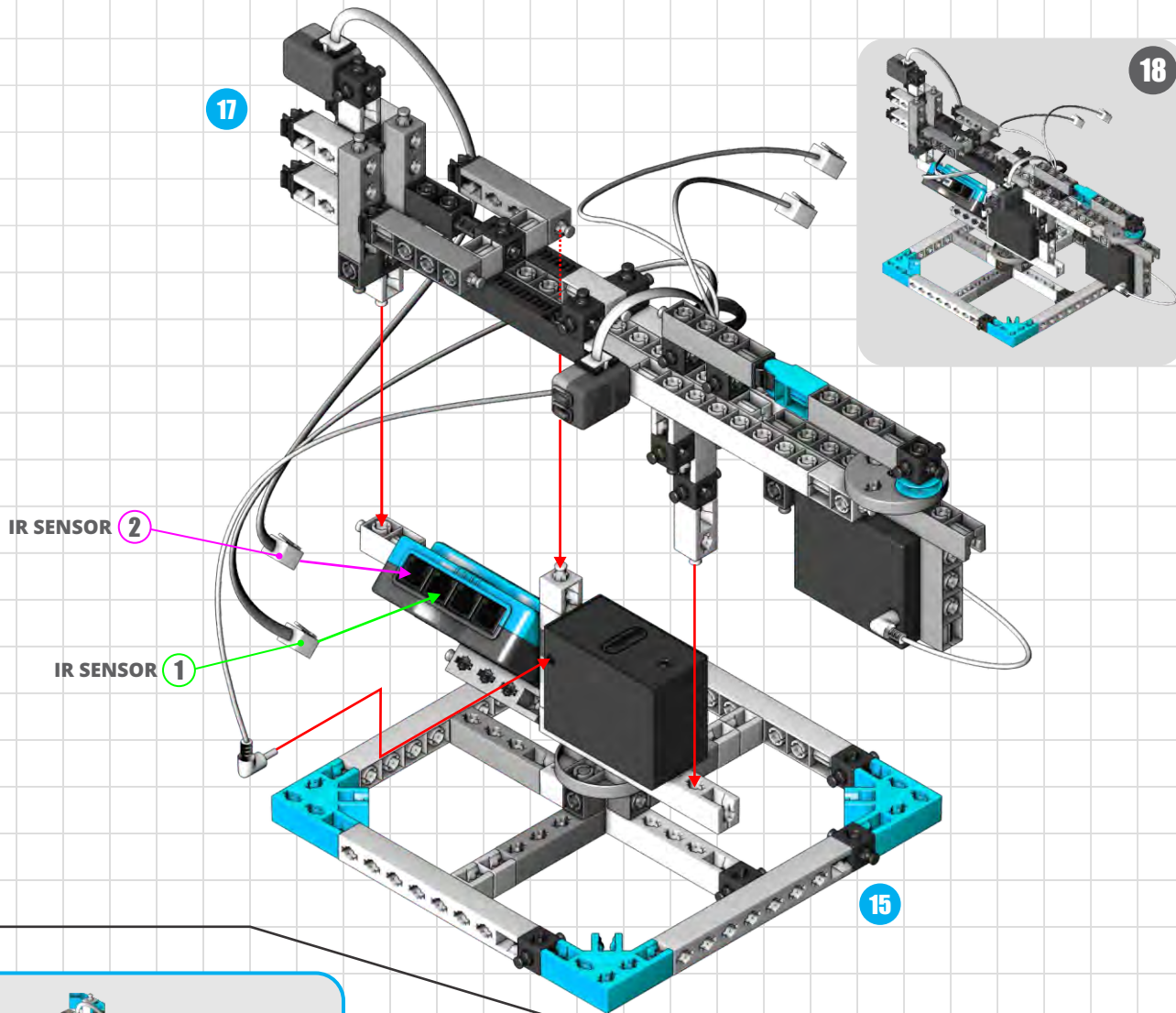


16



17

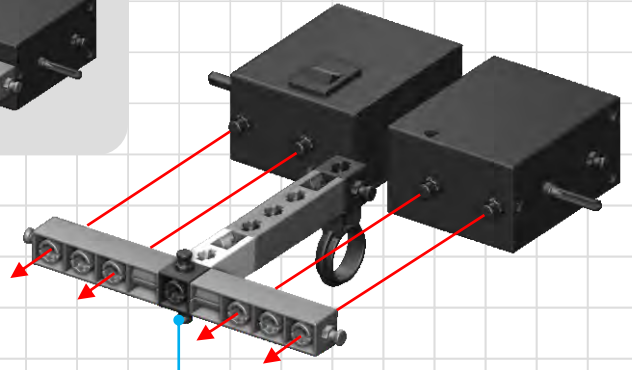




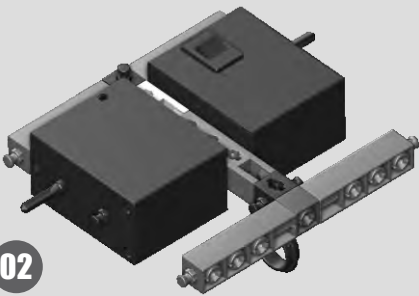
# Škorpión

4025

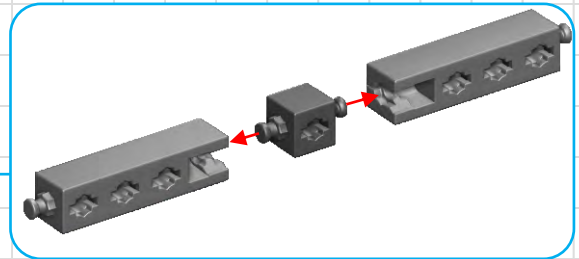
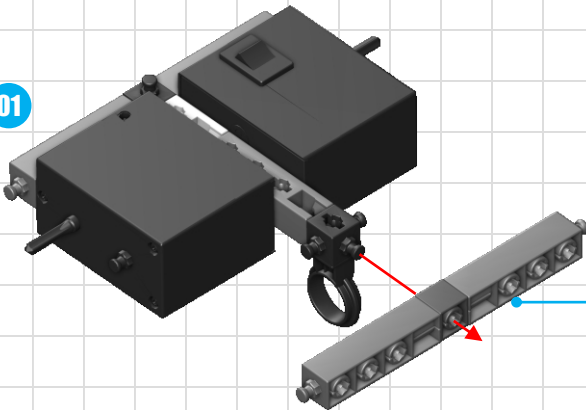
01



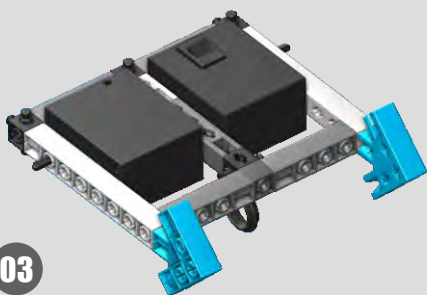
02



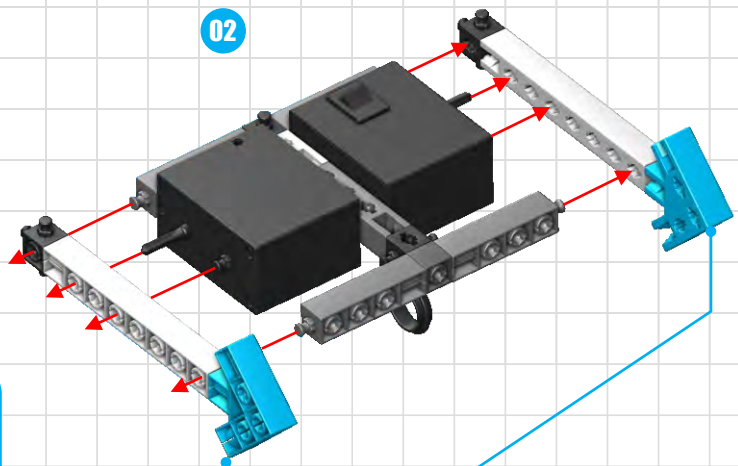
01



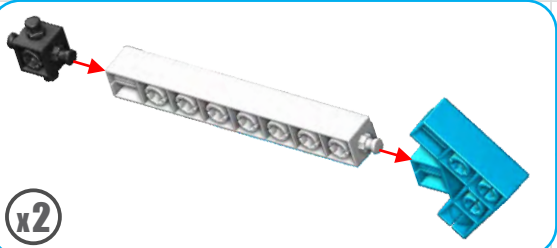
03



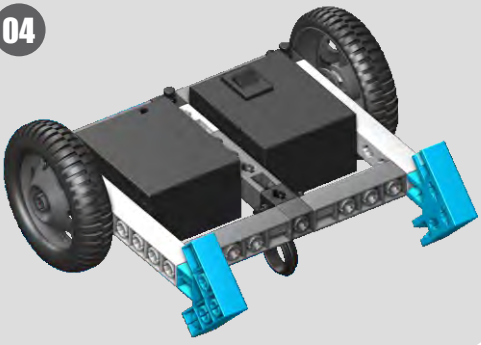
02



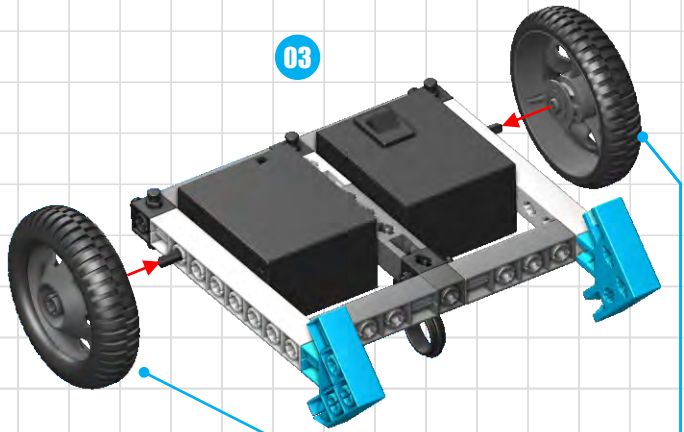
x2



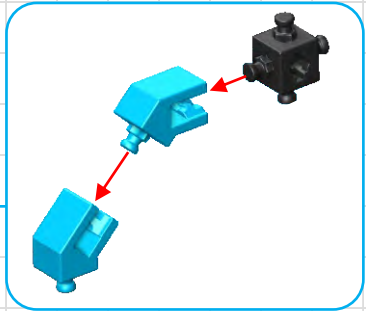
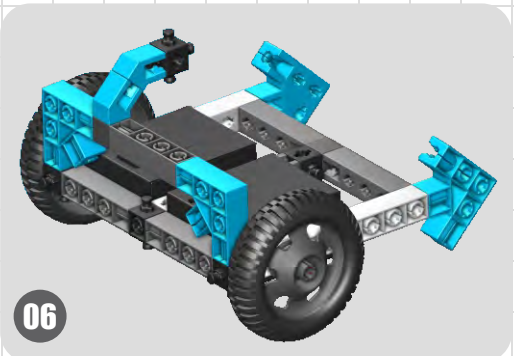
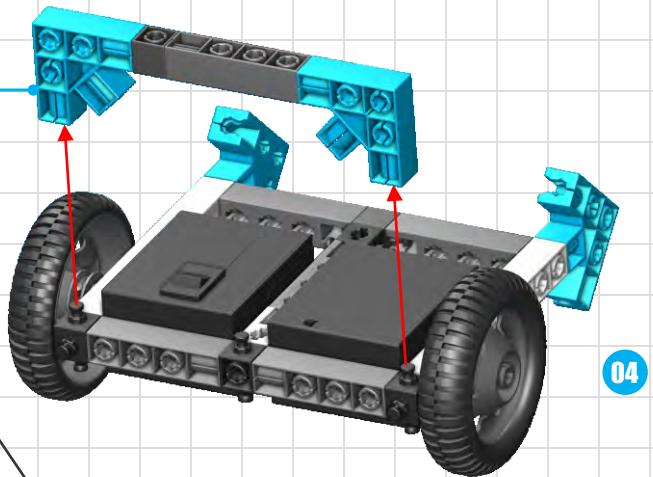
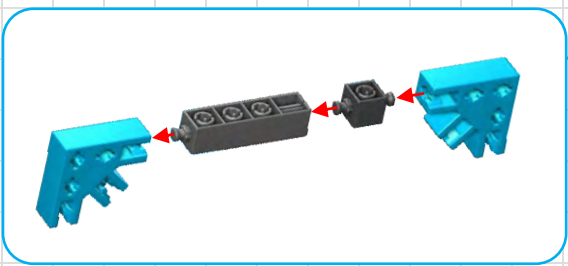
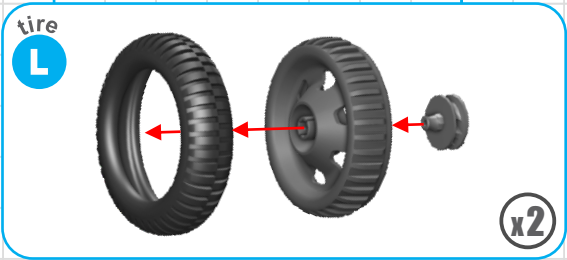
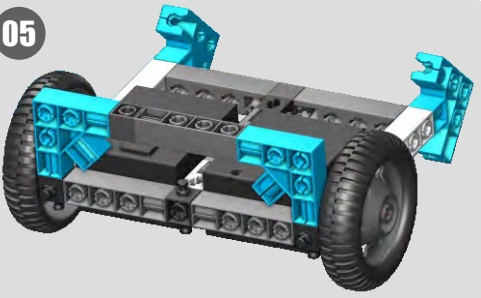
04



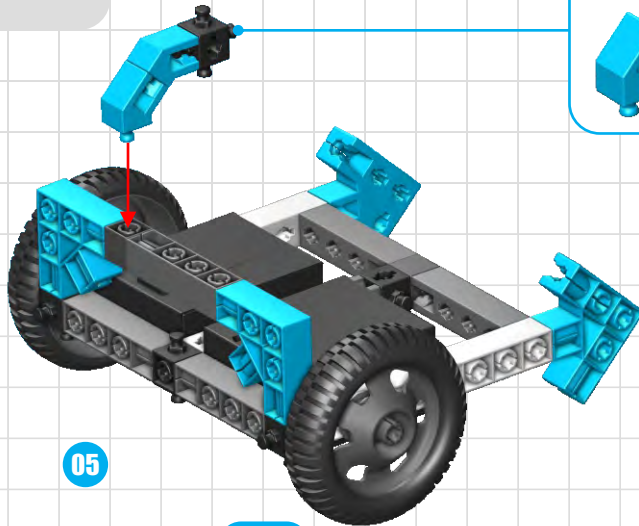
03



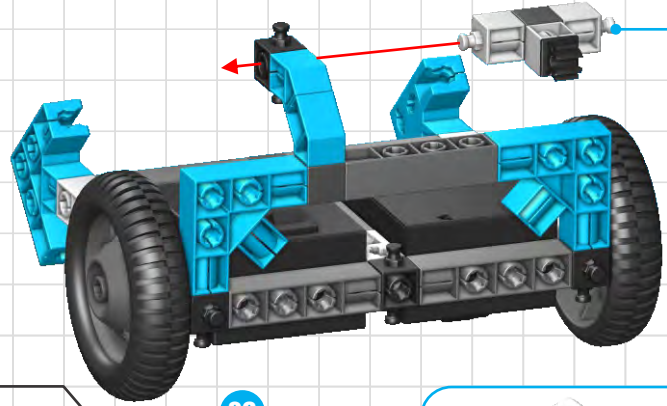
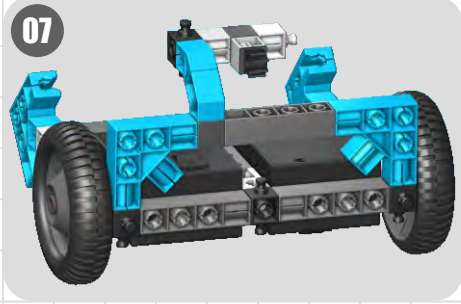
05



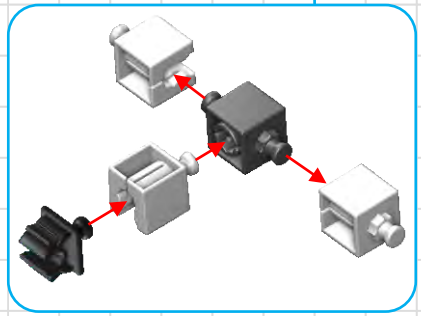
05



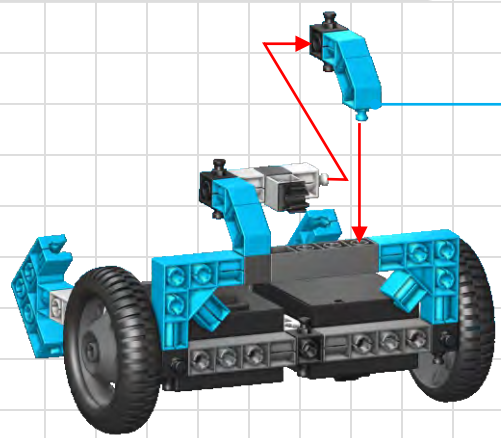
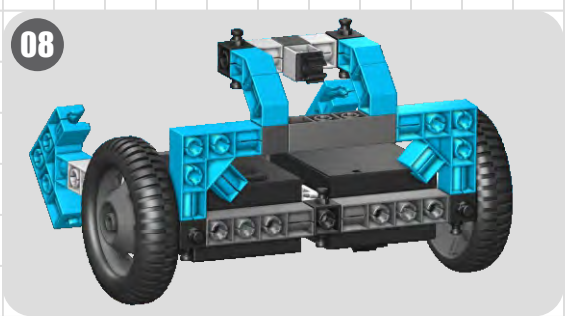
07



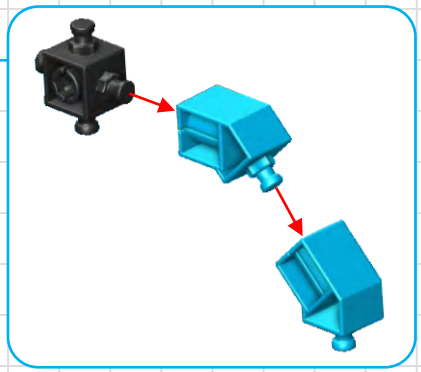
06



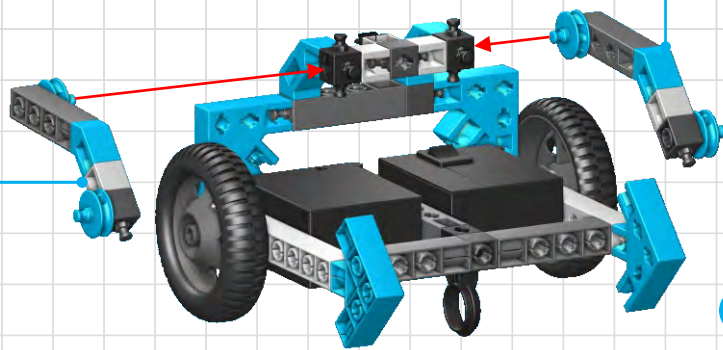
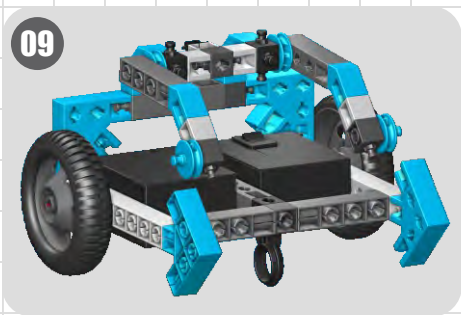
08



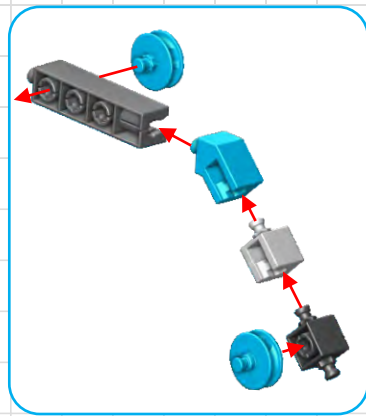
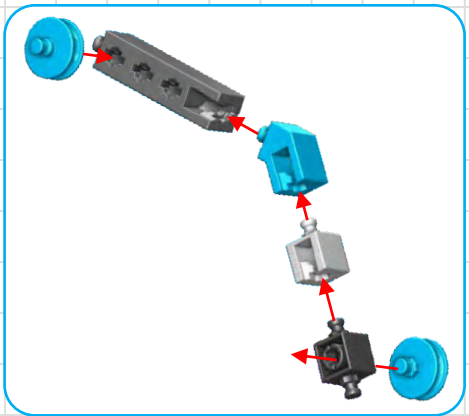
07



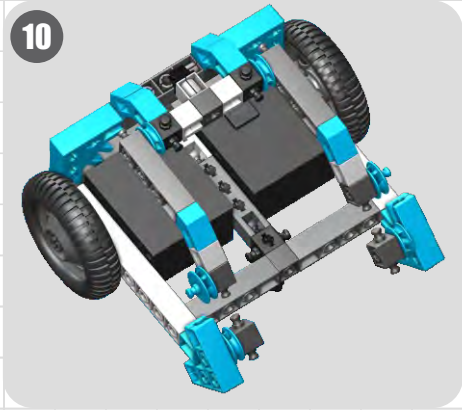
09



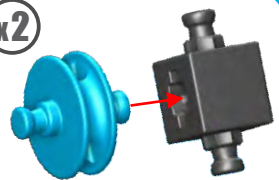
08



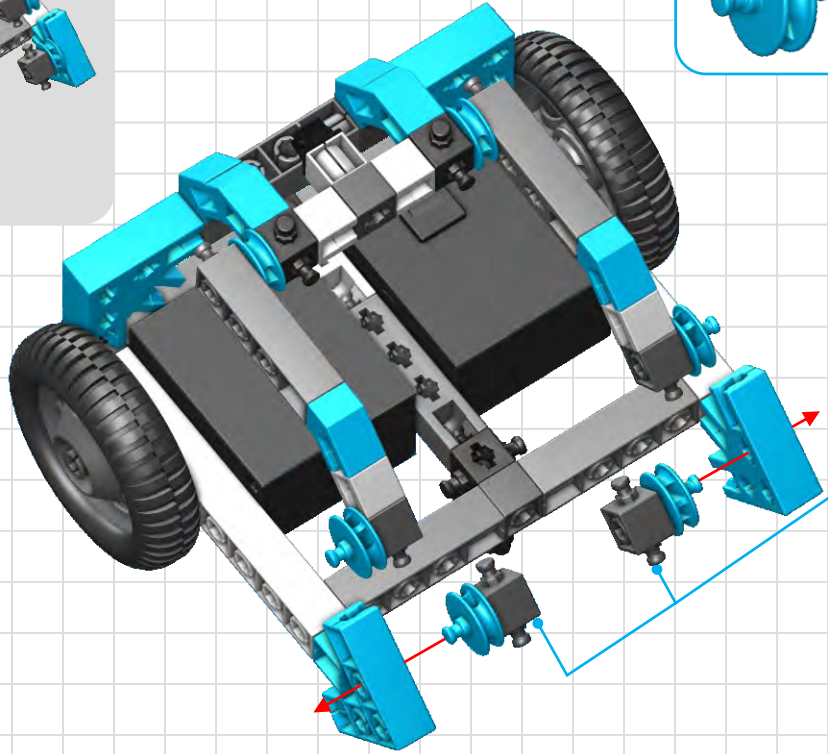
10



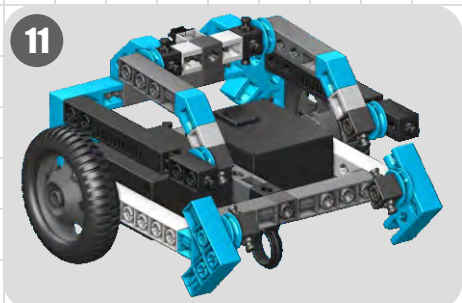
x2



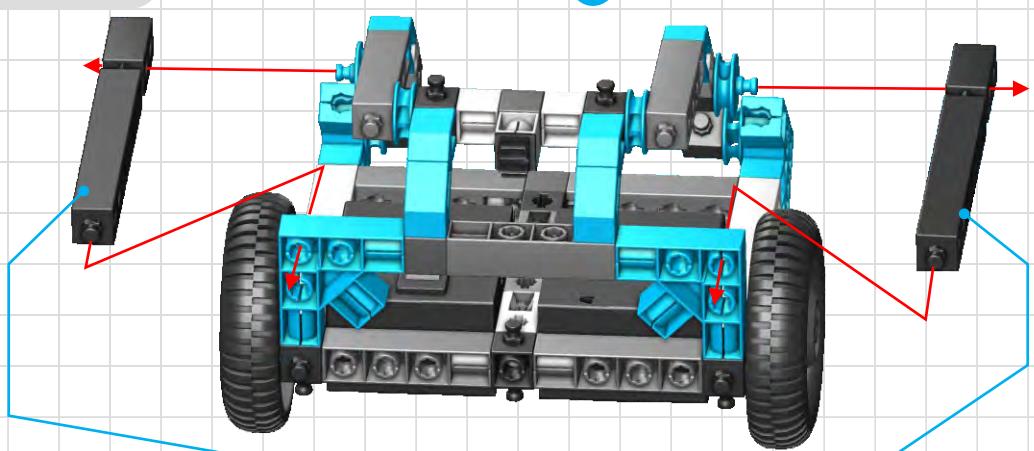
09



11

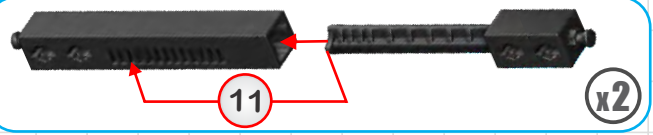


10

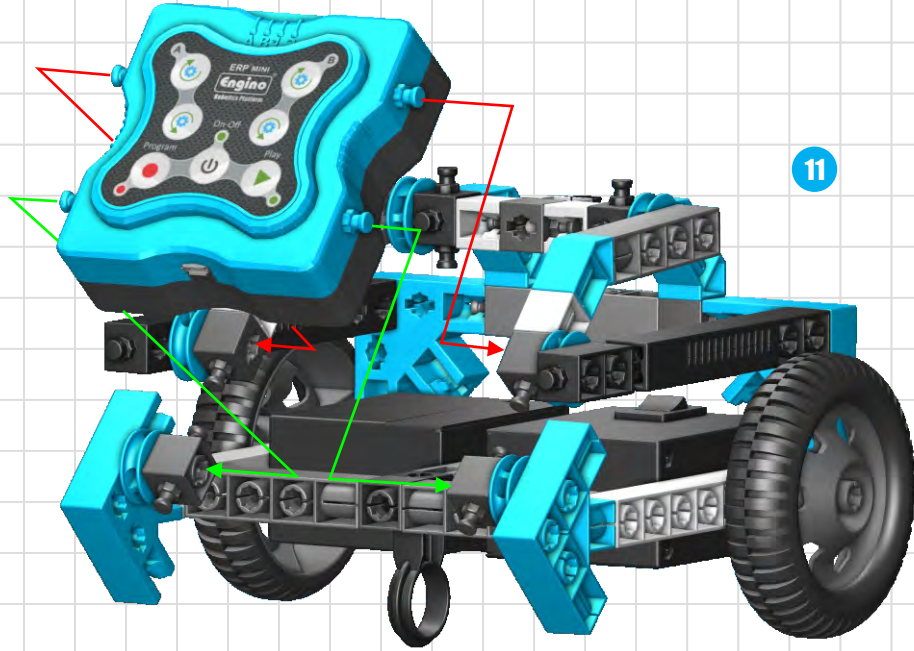


11

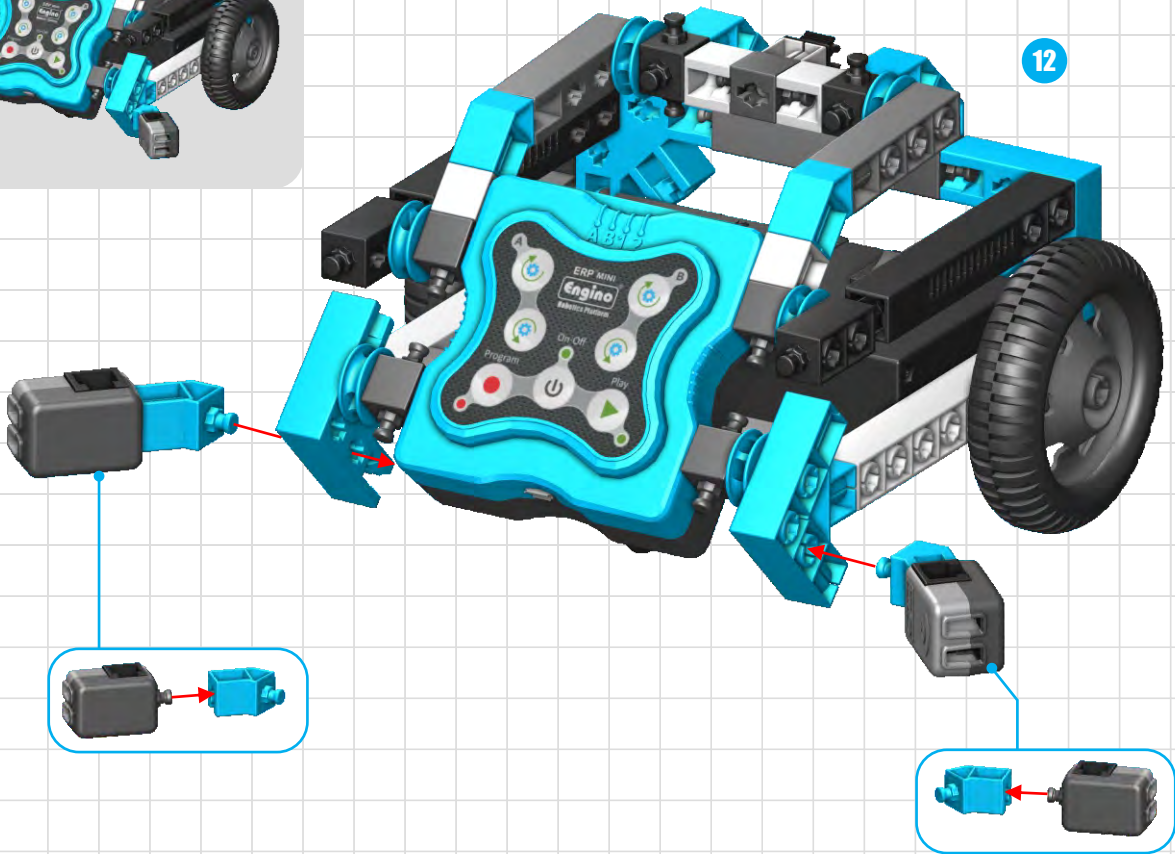
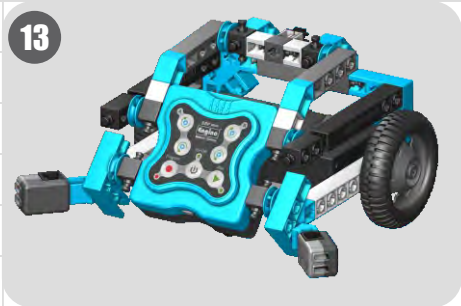
x2



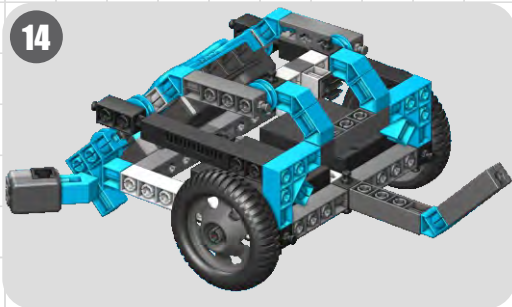
12



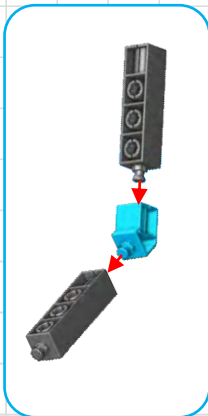
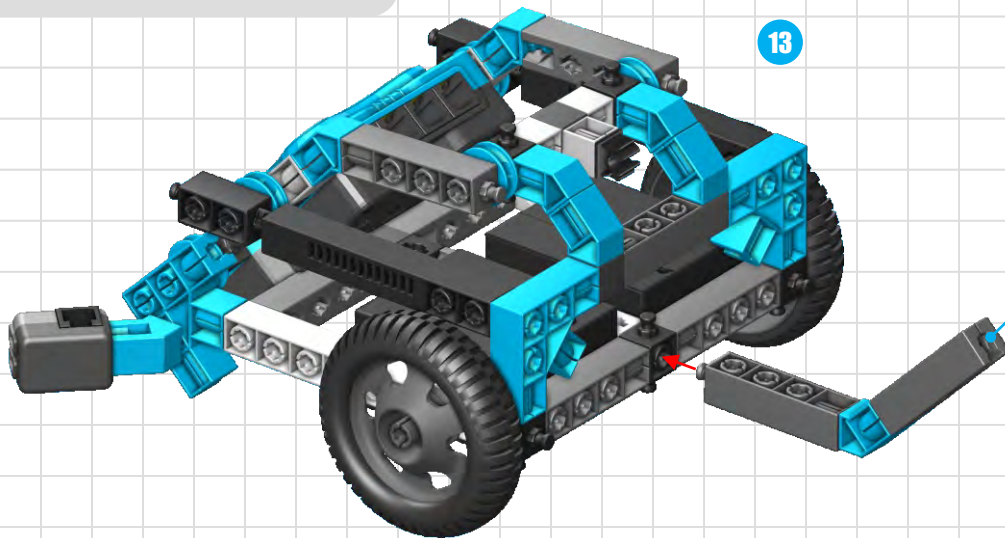
13



14



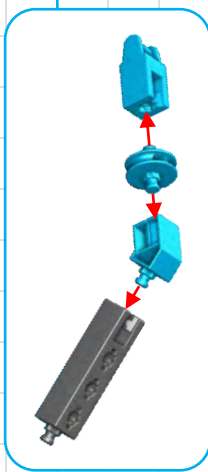
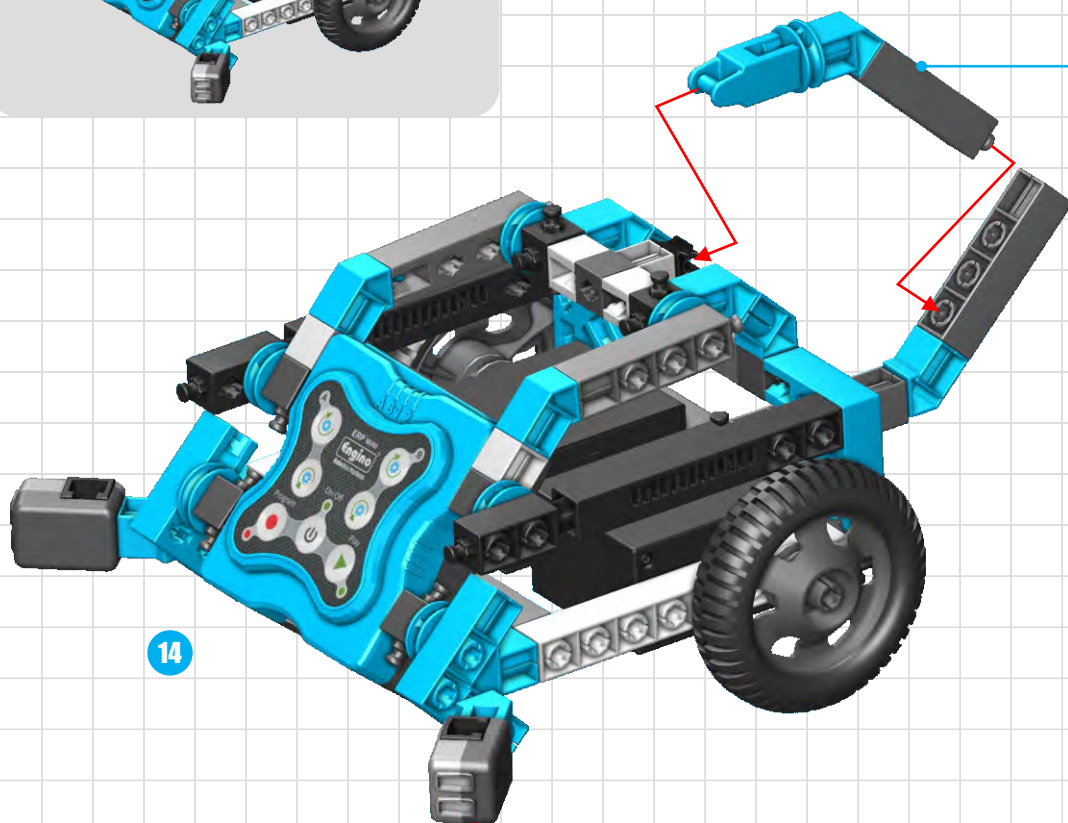
13



15

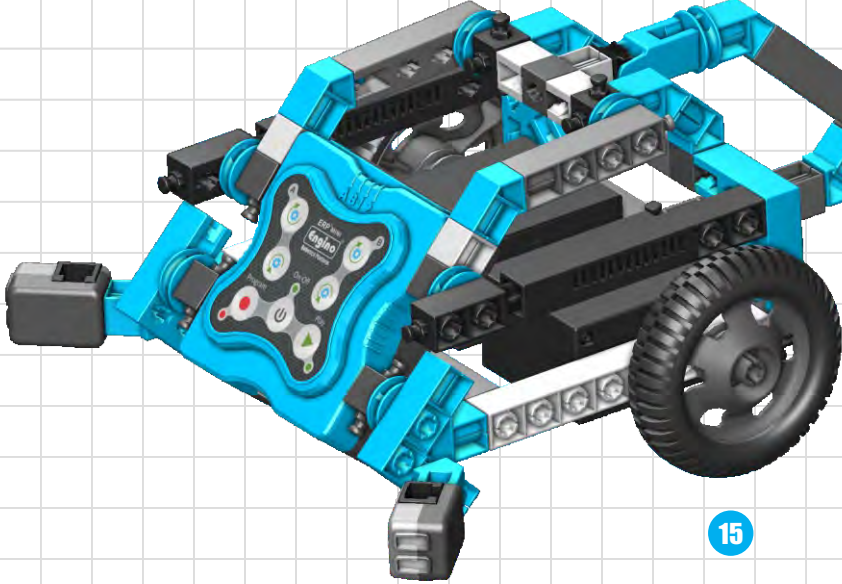


14

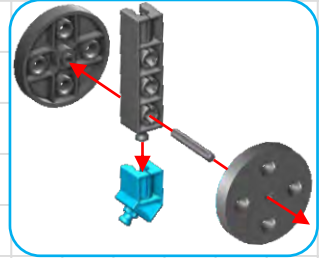




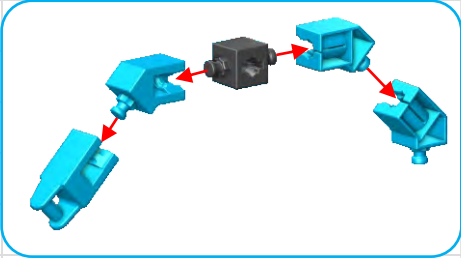
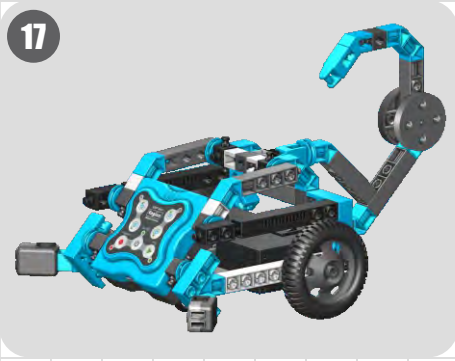
16



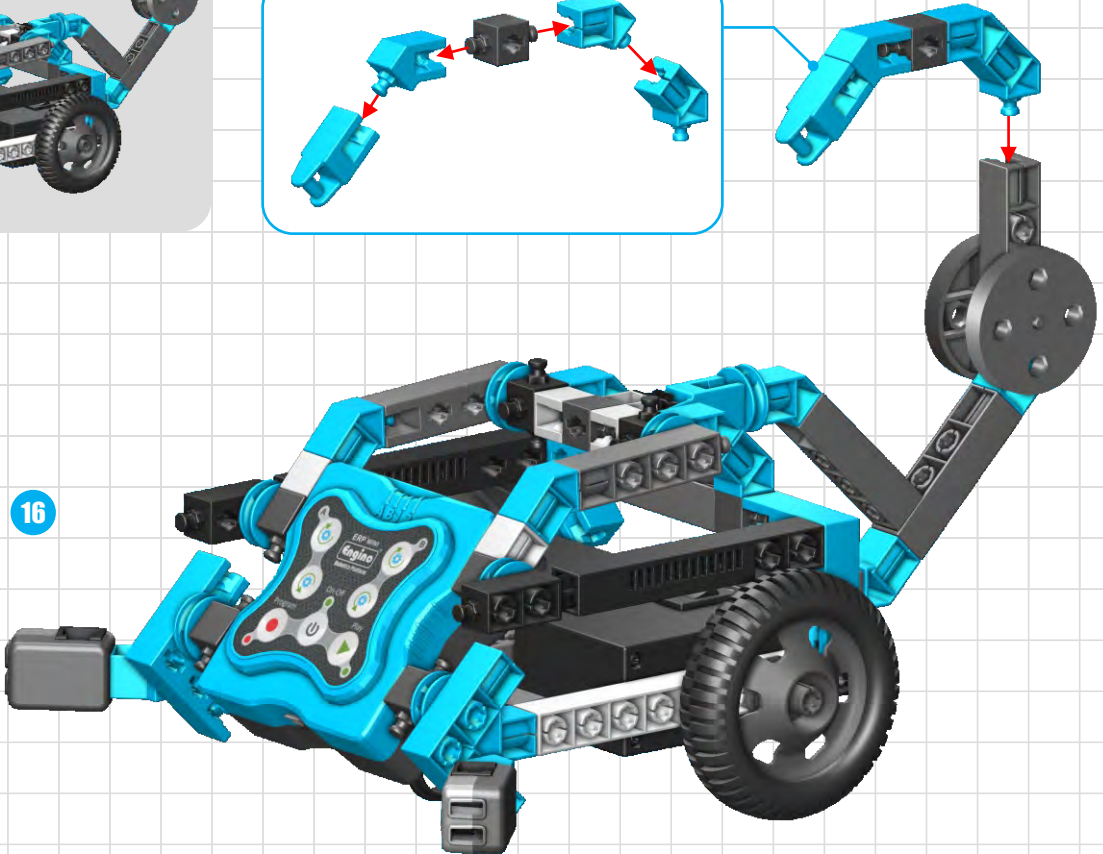
15



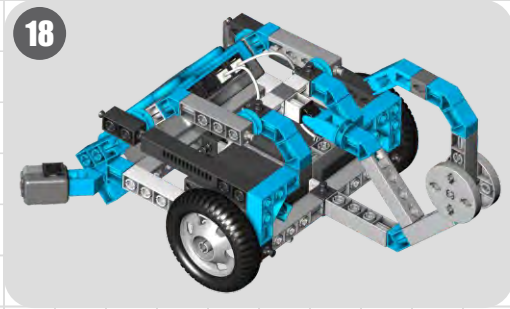
17



16

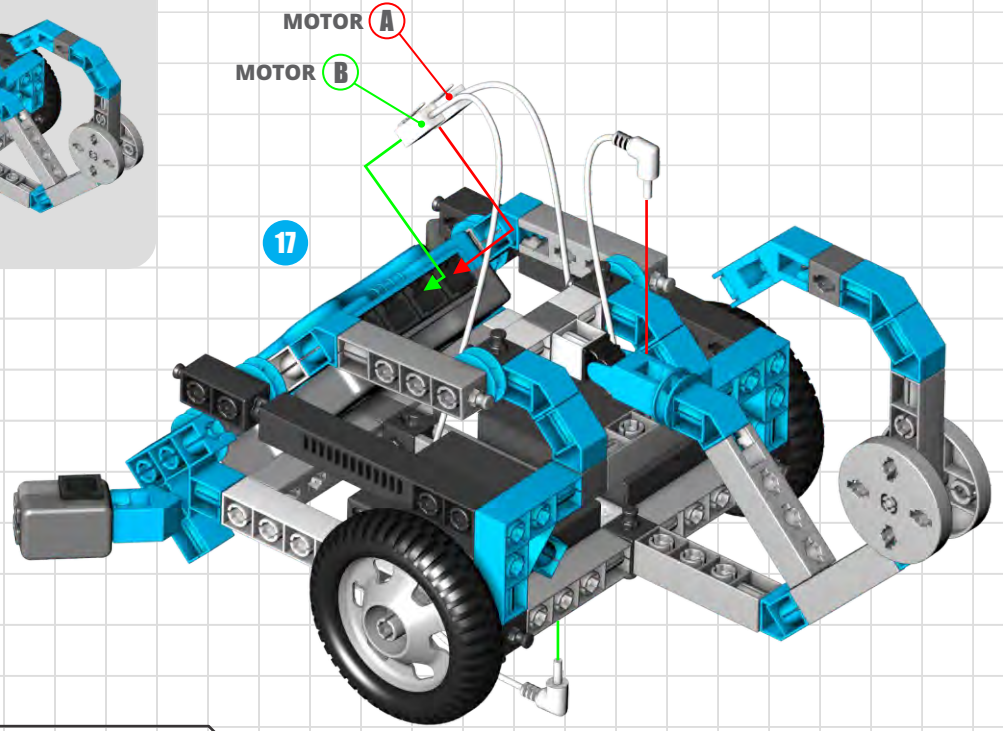


18



MOTOR A  
MOTOR B

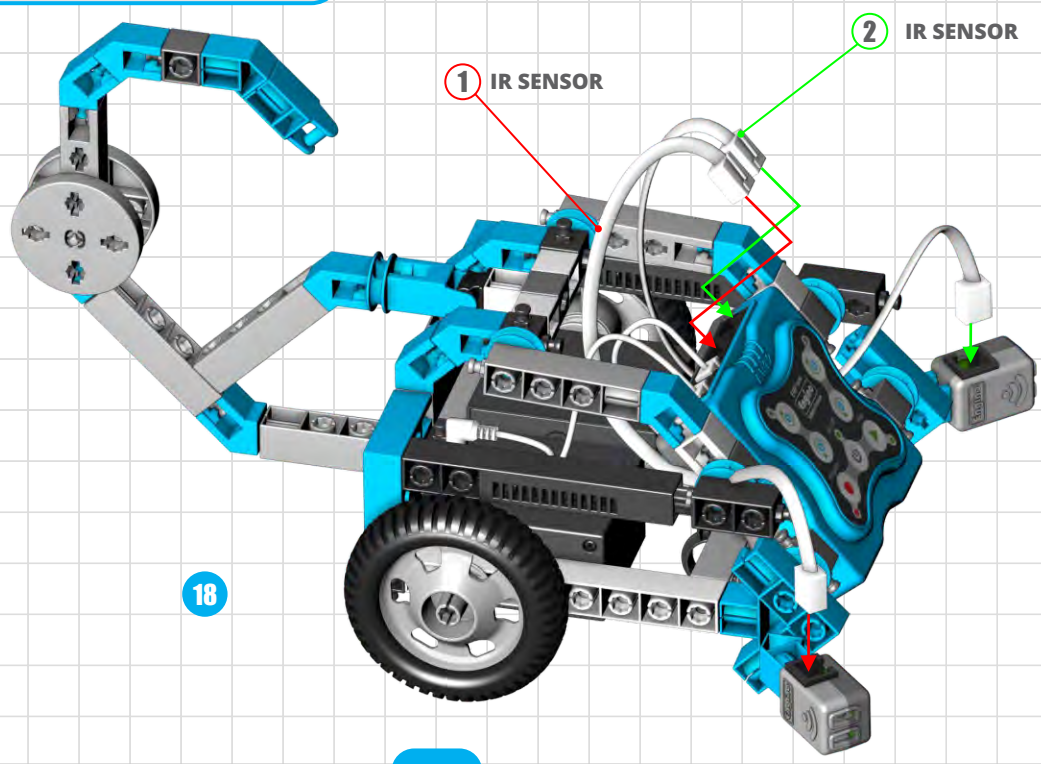
17



2 IR SENSOR

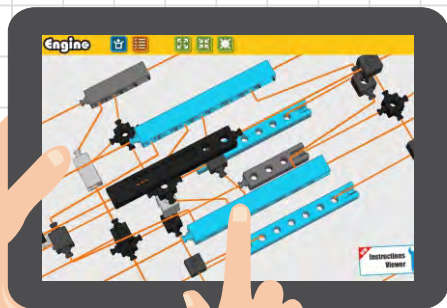
1 IR SENSOR

18



## Bezplatná aplikace

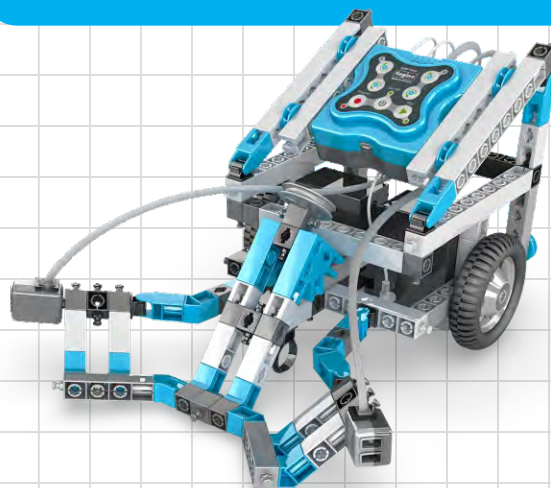
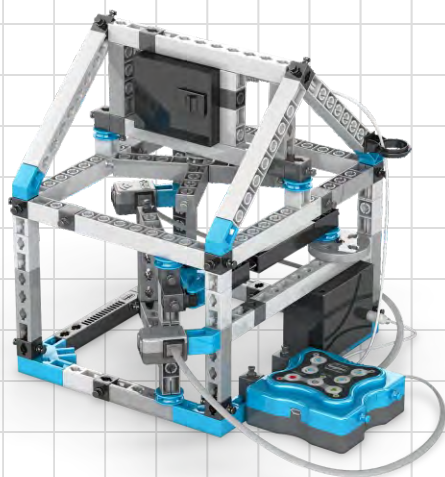
- A** Více modelů najdete se svým počítačem nebo tabletem na tomto odkaze:



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

- B** Stáhněte si aplikaci, kde najdete 3D návody s popisem krok za krokem!

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



## Více teorie a experimentů najdete online!

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

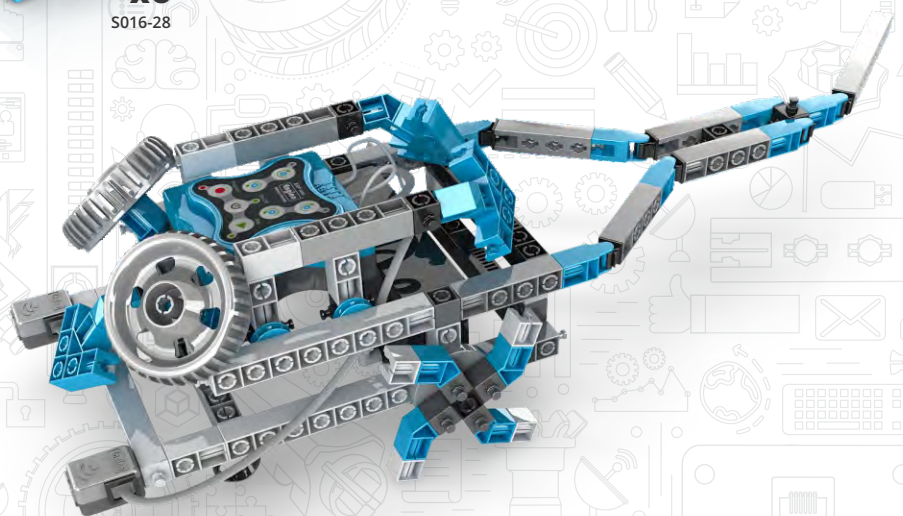
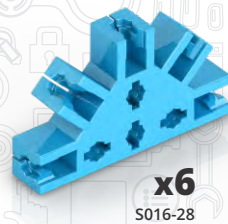
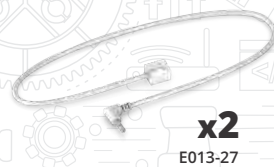
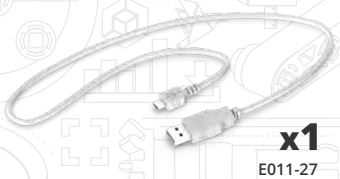
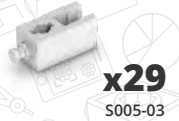
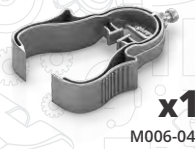


©123RF.com / baloncici, ella, James Steidl, Ales Zvolanek, Olga Serdyuk, spaxia, abidal, dipressionist, Inna Jacquemin, Miroslaw Dobrzanski, scyther5, hxdbxxy, gemenacom, pixinoo, Frederick, Joerg Hackemann, fastfun23, Ronalds Stikans.

Copyright © Engino-Net Limited

Všechna práva vyhrazena. Tyto stránky jsou povoleny pouze k osobnímu užití. Proto je striktně zakázána jejich reprodukce, modifikace, ukládání ve vyhledávacích systémech, retransmise jakýmkoliv způsobem, elektronicky, mechanicky či jinak pro jiný účel než osobní. Vyjímkou může být činnost schválena editorem.

# Seznam součástí



© Copyright Engino-Net Limited. All Rights Reserved



**Distributor:**  
ConQuest entertainment a.s.  
Kolbenova 962/27d  
198 00, Praha 9  
tel: 284 000 111  
info@toy.cz  
www.toy.cz

**VAROVÁNÍ:**

NEBEZPEČÍ UDUŠENÍ - malé části.  
Není určeno pro děti mladší 3 let.



Kód produktu:  
**CE101MP-A**