

# Lego Spike Essential



Miroslav Dvořák

# Obsah

Příprava na výuku .....	2
Instalace programů do zařízení .....	2
Roztřídění kostek stavebnice a popsání stavebnice .....	2
První připojení Hubu k softwaru .....	2
Zábavní park .....	3
Návod na stavbu .....	4
Programování .....	5
Sněžný skútr .....	8
Stavba sněžného skútru .....	9
Programování .....	9
Rozšíření pro zvidavé .....	10
Nocování .....	11
Programování .....	12
Pro zvidavé .....	12
Výlet .....	13
Programování .....	14
Prohlídka Safari parku .....	15
Programování autobusu .....	16
Zastavení u značky .....	16
Změna rychlosti .....	16
Jeřáb – přeprava nákladu .....	17
Stavba jeřábu .....	0
Programování .....	27

## Příprava na výuku

Před výukou bude potřeba si vše nachystat. Tato příprava je pro vás důležitá. Vyhnete se následně problémům při samotné výuce.

### Instalace programů do zařízení

Pro výuku je potřeba mít pro každou stavebnici zařízení s programem Lego Spike. Jako zařízení lze využít počítač, notebook nebo tablet s podporovaným operačním systémem (Windows 10, MacOS, iOS – tablet, Android – tablet, Chromebook). Zařízení musí mít podporu Bluetooth min. verze 4.0. Zařízení by také mělo mít přístup k internetu pro aktualizace a stažení podpůrných materiálů. I když lze využít i počítač, doporučuji použít notebooky případně tablety s dotykovou obrazovkou pro snadnější ovládání programu.

Program si lze stáhnout z obchodů jednotlivých operačních systémů a pro Windows lze stáhnout i instalátor přímo ze stránek Lego Education. Doporučuji instalovat verzi ze Store, protože se vám aplikace bude nadále automaticky aktualizovat.

Po instalaci programu je potřeba ještě stáhnout *Unit Plans*. Pokud je nemáte stažené budete mít prázdnou nabídku *Build*. Je to nabídka, kterou budeme využívat v následujících příkladech.

### Roztřídění kostek stavebnice a popsání stavebnice

Dalším krokem je roztřídění stavebnice a její popsání. Lego Spike Essential doporučuje třídění podle barev. Toto je u menších dětí mnohem výhodnější, protože se ve stavebnici snáze orientují a rychleji najdou požadovanou součástku. Je také důležité si popsat jedinečným názvem celou krabici a i Hub. Tento jedinečný název následně uložíme do Hubu. Protože se Huby připojují k zařízení pomocí Bluetooth je potřeba použít jedinečný název, aby se stavebnice od sebe odlišily.

### První připojení Hubu k softwaru

První připojení doporučuji udělat pomocí USB kabelu. S velkou pravděpodobností bude probíhat aktualizace firmwaru v Hubu a tato aktualizace probíhá mnohem rychleji po USB kabelu, než přes Bluetooth.

## Zábavní park

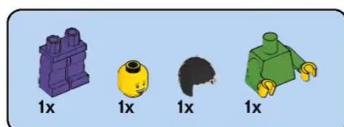
Byl jsi někdy v zábavním parku nebo na pouti?

Jaká atrakce tě nejvíce zaujala?

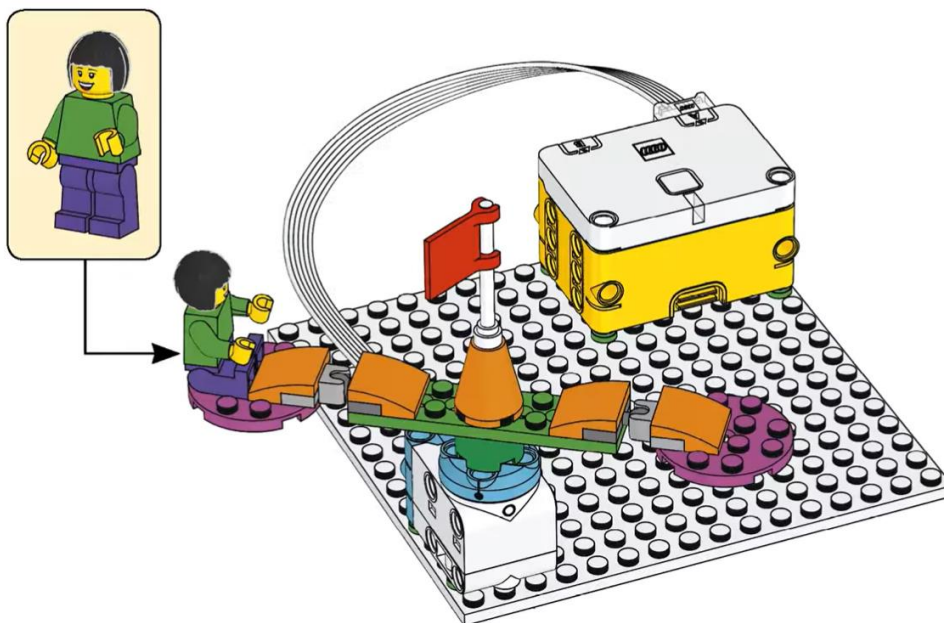
Proč děti rády navštěvují zábavní parky?

I Sofie navštívila zábavní park. Viděla tolik krásných věcí. Nejvíce ji zaujal kolotoč.

Pojď Sofii pomoci jej postavit.

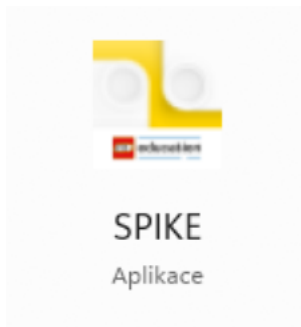


**12**

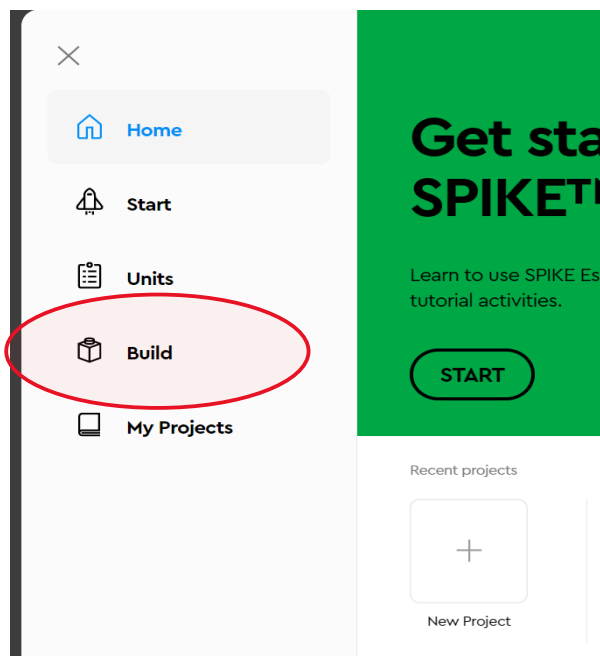


# Návod na stavbu

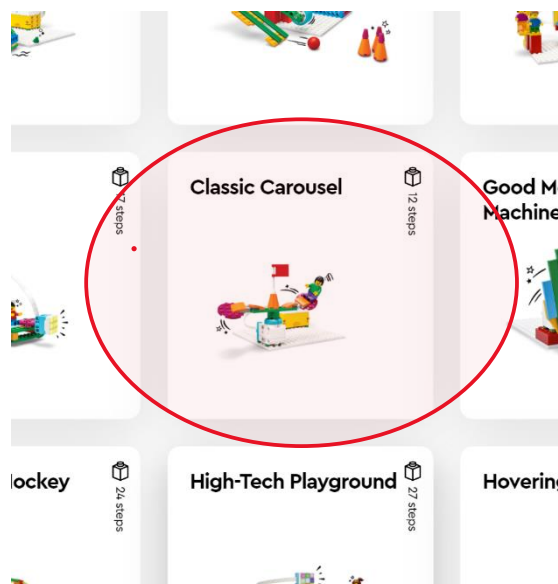
Všechny návody najdeš v aplikaci Lego Spike kterou najdeš v počítači.



Po spuštění najdeš návody na stavbu v nabídce *Build*.



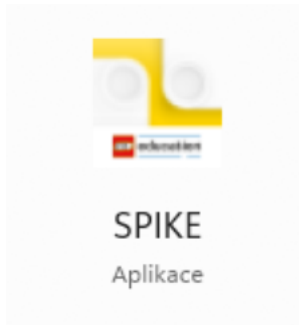
My si postavíme kolotoč.



Pokud nemáš v programu žádné návody na stavbu, popros o pomoc učitele.

## Programování

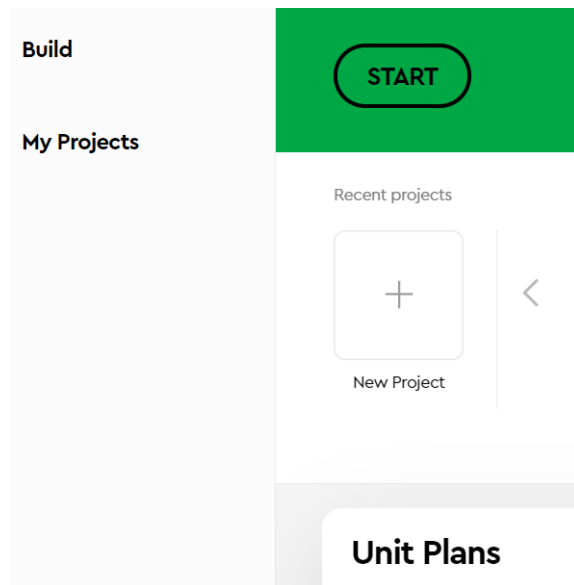
Jdeme na tvé první programování. Pokud máš již program spuštěný, můžeš přeskočit spuštění. Nejprve spusť program Spike.



Zvol verzi Spike Essential

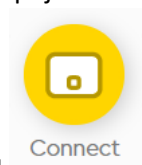


a nakonec New Project

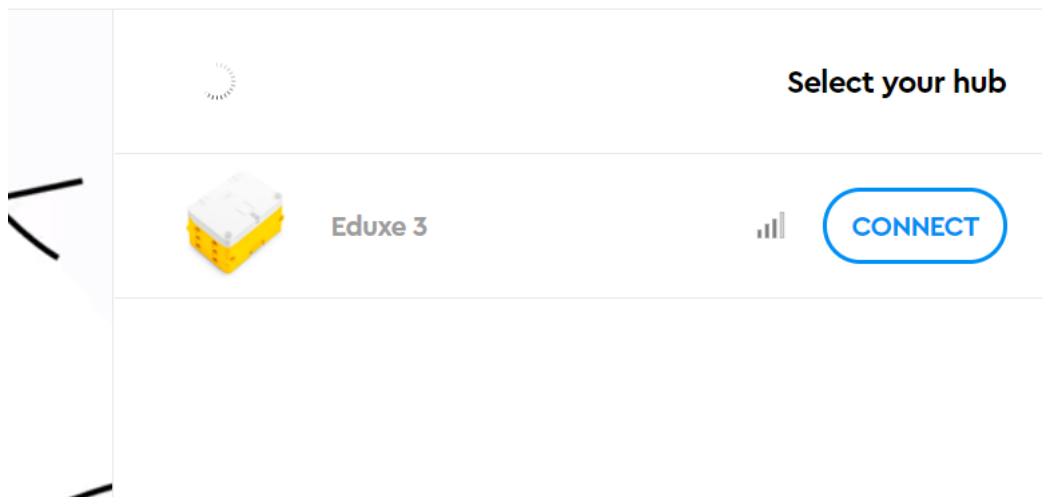


Z možností programování si vyber Icon Block, což je základní programování, ve kterém budeme tvořit. Volbu potvrď kliknutím na Create.

A jsem v prostředí pro programování. První úkolem bude připojit si náš Hub (kostku)



k programovacímu prostředí. Vlevo nahoře klikni na ikonku a po zobrazení názvu tvé kostky klikni na tlačítko Connect.



Tím se kostka připojí a barva její kontrolky se změní na modrou. Zpět do prostředí se vrátíš křížkem na připojovacím okně.



Na ploše vidíte první příkaz. je to spouštěcí událost. Každý program musí začínat nějakou událostí. Zde jde o kliknutí na trojúhelník. Později se naučíme i jiné události. Další příkazy,

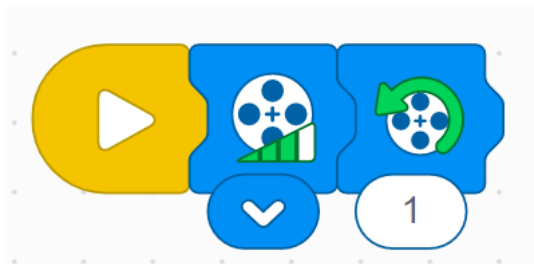


které budeme v tomto modelu využívat jsou:

Co myslíš, že znamenají? Víš, jak na to přijdeš? Vyzkoušej je.

Úkol číslo 1.

Je rozdíl v těchto programech?



Našel jsi rozdíly? Skvělé. Program se čte jako kniha. Takže je potřeba nejprve nadefinovat rychlost a potom směr a počet opakování.

Máš ještě chvíličku čas? Vyzkoušej různé jiné varianty kolotoče.





## Sněžný skútr

Naším úkolem bude postavit sněžný skútr. Víš, kde se používají sněžné skútry a kdo je používá? Co se pomocí sněžného skútru dá přepravovat?

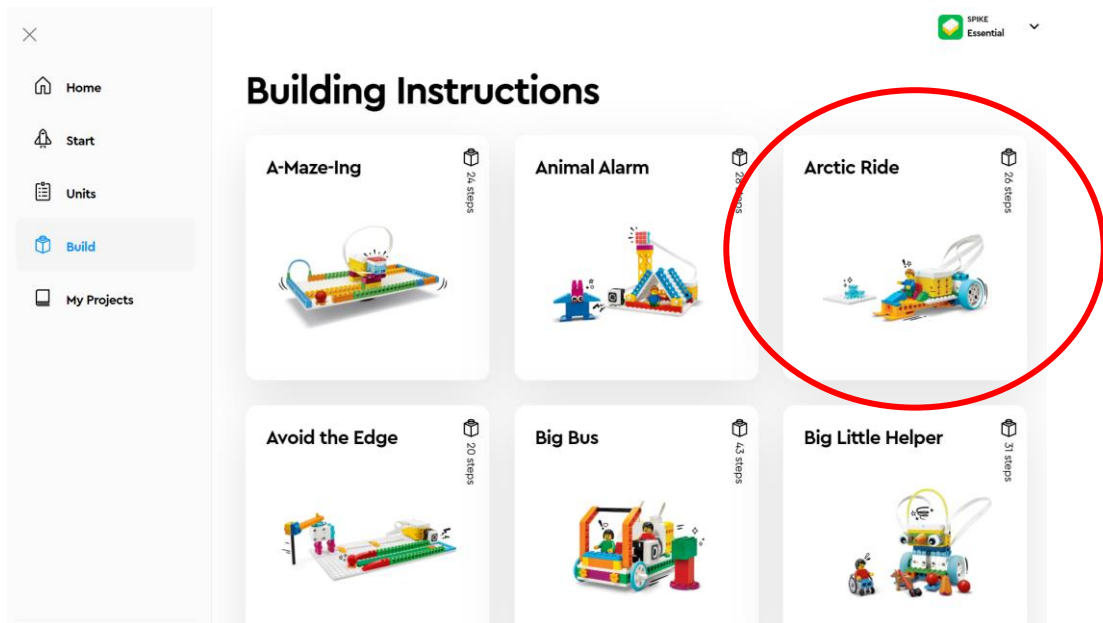
Sněžné skútry se používají za polárním kruhem. Chtěl bys tam bydlet a proč?



## Stavba sněžného skútru

Pojďme postavit svůj vlastní sněžný skútr, který nám pomůže prozkoumávat život za polárním kruhem.

Návod najdeš na programu Spike – Build – Artic Ride.



## Programování

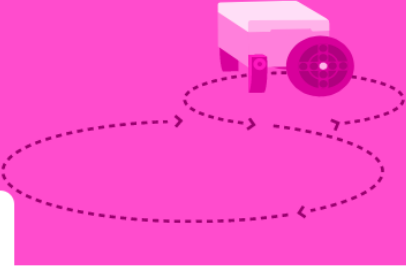
Nyní můžeme začít programovat. Spustíme si aplikaci Spike a zvolíme si Spike Essential a zvolíme nový projekt. Programovat budeme pomocí Icon Block.

Budeme si muset přidat rozšíření pro pohyb. Klikneme na tlačítko pro rozšíření



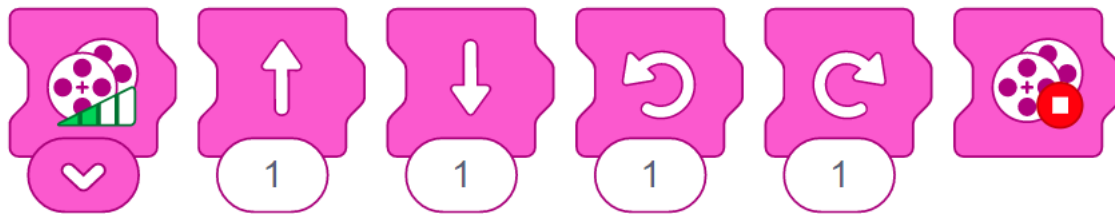
a z rozšíření si vybereme Movement. V prostředí se nám objeví nové možnosti programování.

## Movement



Set the individual motor speed on a Driving Base or make it hold its position.

> LEARN MORE

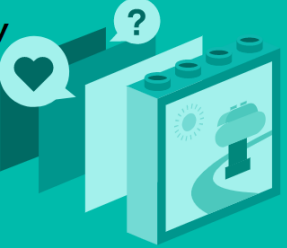


Vyzkoušej jednotlivé příkazy a hurá za objevováním.

### Rozšíření pro zvědavé

Když už ti tak hezky funguje sněžný skútr tak co opravdu vyrazit na sníž. Naučme se měnit pozadí na obrazovce. Tak jak jsme si přidali rozšíření pro pohyb, můžeme přidat rozšíření i pro displej.

## Display



This extension allows you to display images or text - Streaming Mode only.

> LEARN MORE



Toto rozšíření umožňuje měnit pozadí obrazovky tvého zařízení. Vyzkoušej jak na to a určitě zvládneš jezdit po sněžných pláních.

## Nocování

Tak to bylo tedy zajímavé. Blíží se večer, a tak si postavme přístřešek a hop na kutě. V noci je také hodně věcí k pozorování. Mám nápad. Postavíme si automatický hlásič, který nás upozorní, pokud se někdo bude pohybovat okolo našeho přístřešku. A teď mě napadlo, nemáte doma něco podobného?



Jdeme stavět. Náš model má název – **Animal Alarm**.

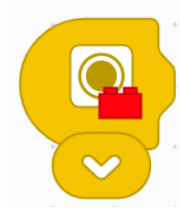
### Animal Alarm



Využijeme zde nový senzor. Jde o barevný senzor. Tento senzor umí poznávat barvy. A náš program se tedy spustí, pokud zaznamená modrou barvu. Už tušíš jak na to?

## Programování

Doposud jsme začínali program vždy pomocí bílé šipky, ale program může začínat i jinak. Náš program se spustí až uvidí konkrétní barvu. Vyměň tedy začátek programu za následující ikonu.



Pokud rozbališ nabídku, uvidíš, jaké barvy umí rozpoznat a jaké tedy mohou spouštět program. Super, ale co znamená ten přeškrtnutý kruh? Jaká je to barva? Určitě mi odpovíš.

Tak máme začátek a nyní již ten program. Úkol zní:

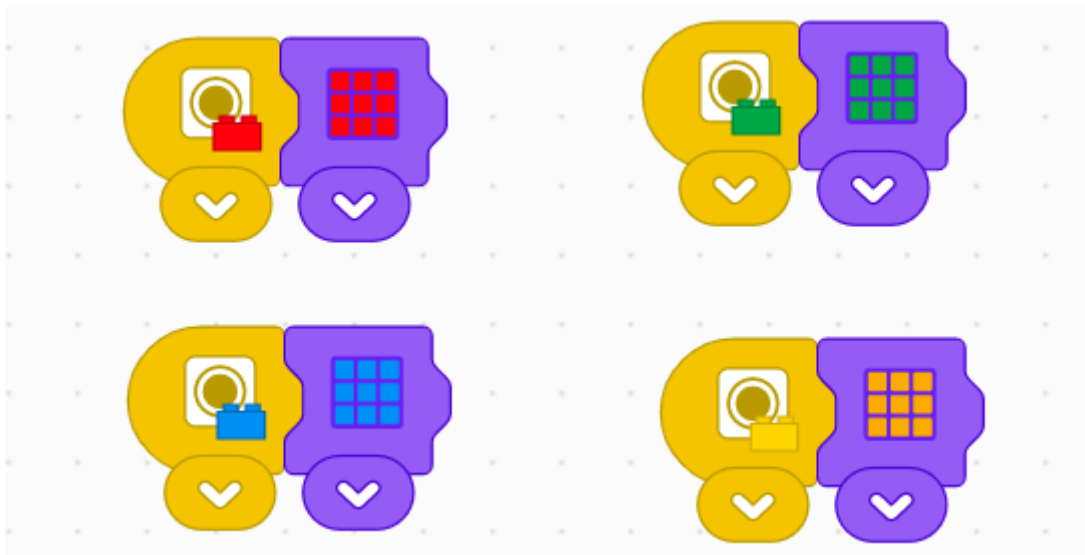
Pokud přijde zvíře modré barvy, přehraj fanfáru číslo 1 a na matrici zobraz modrou barvu.

Že to není těžké.

## Pro zvědavé

Už vše funguje a máš ještě čas. Tak pojďme náš program vylepšit.

Co kdybychom hned mohli poznat jakou barvu má zvíře, které je venku. Šlo by to? Zatím jsme tvořili vždy jen jeden program. Ale my můžeme udělat programů víc a spustit všechny naráz. Následně se bude spouštět konkrétní program.



# Výlet

Určitě jste někdy byli na výletě. Jak na výlet jezdíte? Který výlet tě nejvíce zaujal? Kam bys chtěl jet na výlet?

Pojedme tedy společně na jeden takový. Kam to bude?

- Hrad?



- Zoologická zahrada?



- Hory?



Taxi! Taxi!

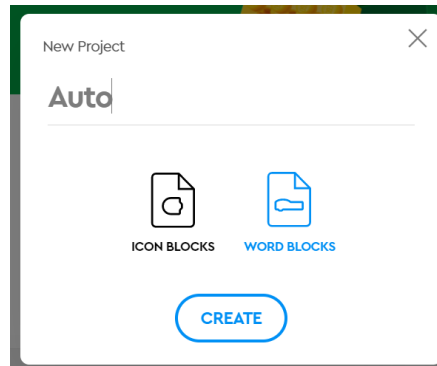
29 steps

My pojedeme autem, pojďme si jej postavit.

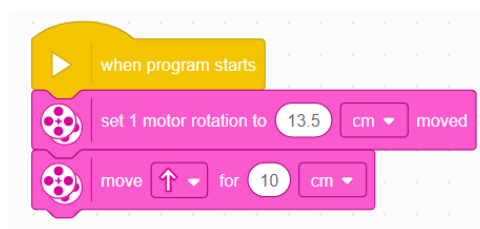


## Programování

Protože jste již zkušení programátoři, postoupíme o krok dále. Budeme programovat pomocí *Word Block*



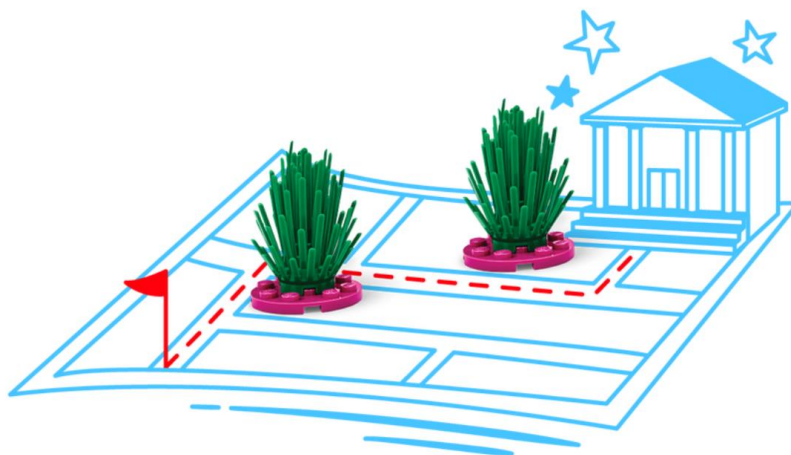
Toto programování má stejné principy jako pomocí *Icon Block*. Jen přináší více možností. Pojďme tedy rozhýbat naše auto. Pro pohyb využijeme nabídku *Movement*. První příkaz, který musíme přidat je: *Set 1 motor rotation to ...*. Pomocí příkazu nastavujeme obvod kol. A pokud jsme použili stejná kola jako v návodu můžeme nechat hodnotu 13,5 cm. Nastavení této hodnoty je důležité, pokud chceme jezdit na předem danou vzdálenost. Nyní můžeme přidat příkaz pro pohyb vpřed o 10 cm.



Vyzkoušej, co se stane, pokud změníš hodnotu pro obvod kola. Popojede autíčko pokaždé o 10 cm?

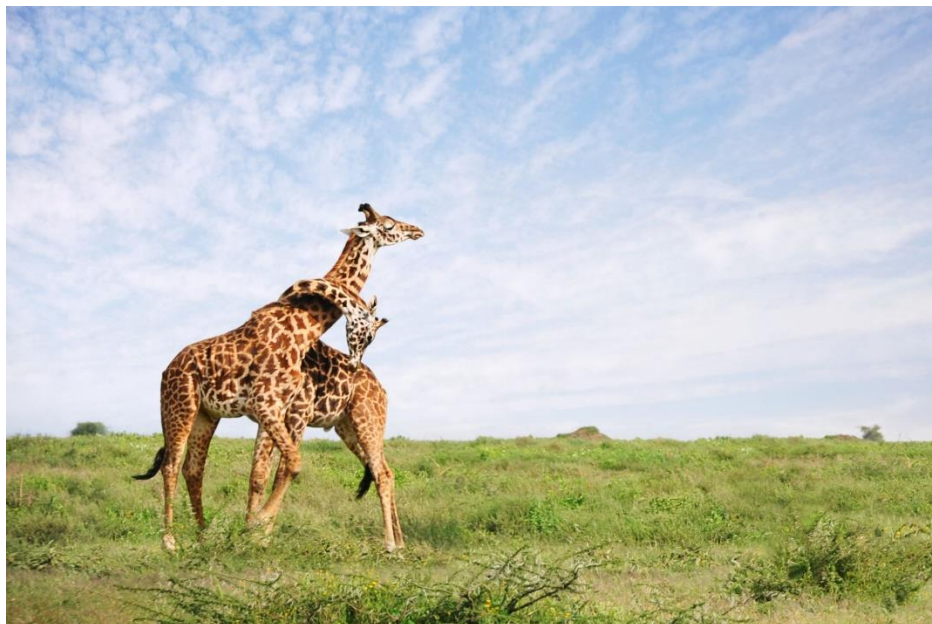
A jak zatáčet? Můžeš využít dvě možnosti. Pomocí příkazu *Move vlevo (vpravo)* se autíčko otáčí jako tank. Jedno kolo se točí dopředu a jedno dozadu. Nebo pomocí *Move right (left)*, kdy autíčko zatačí stejně, jako běžné auto, když otočíš volant.

Nyní již víš vše a můžeme vyrazit na cestu. Pojď si postavit cestu a zvládni ji projed celou.




## Prohlídka Safari parku

Jsme na výletě. Mám moc rád Safari park. Zvířata jsou těsně kolem mě. Jen se jich dotknout. Také bys to chtěl zažít? Pojď se mnou.



Pojedeme spolu autobusem. Už jen jej postavit. Honem.

**Big Bus**

 43 steps





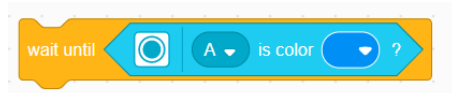
## Programování autobusu

Už asi nebude problém se s autobusem projet. Jen malé upozornění, používáme jeden motor, takže využij příkazy z modré části *Motors*. Ale co třeba využít světelný senzor pro zastavení i značky. To je už výzva.

### Zastavení u značky

Cílem bude, aby jedoucí autobus zastavil přesně u modrého sloupku. Jak na to?

Využij příkaz *Wait until*, který najdete v sekci Control, do něj vložíme hodnotu ze světelného senzoru. Mohlo by to vypadat následovně:

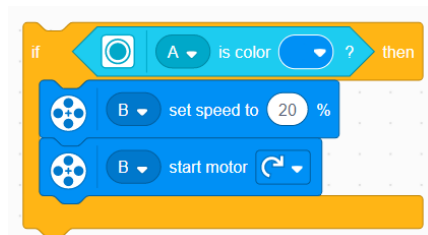


Potom už nám stačí pouze zastavit pohyb. (*Stop moving*)

Funguje to. Tak co třeba doplnit program o melodii, která by se přehrála po zastavení?

### Změna rychlosti

Tak už umíš zastavit? Skvělé. Ale co když chceme použít více barev? Mám takový úkol. Autobus pojedě k modrému sloupku rychlostí 75% a od modrého sloupku k červenému pomocí rychlosti 20%. Přeci si chceme vše prohlédnout. Ještě trochu ti poradím. Budeš potřebovat příkaz *If*. Ten funguje tak, že pokud je podmínka (barva na senzoru) splněna tak udělá příkaz co bude uvnitř. Pokud není splněna jde dál. Podmínka by mohla vypadat takto:



Zbytek již nechám na tobě.



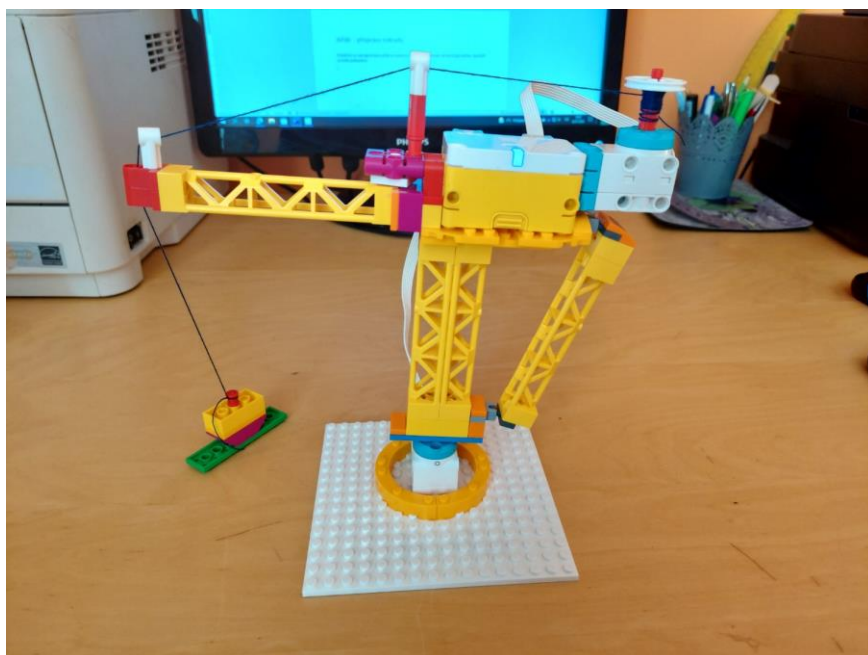
Že to nefunguje. Pozor. Program se čte jako kniha. Spustíš je a on se jednou přečte. Pokud se má číst víckrát musíš podmínky zabalit do příkazu *Forever*.

## Jeřáb – přeprava nákladu

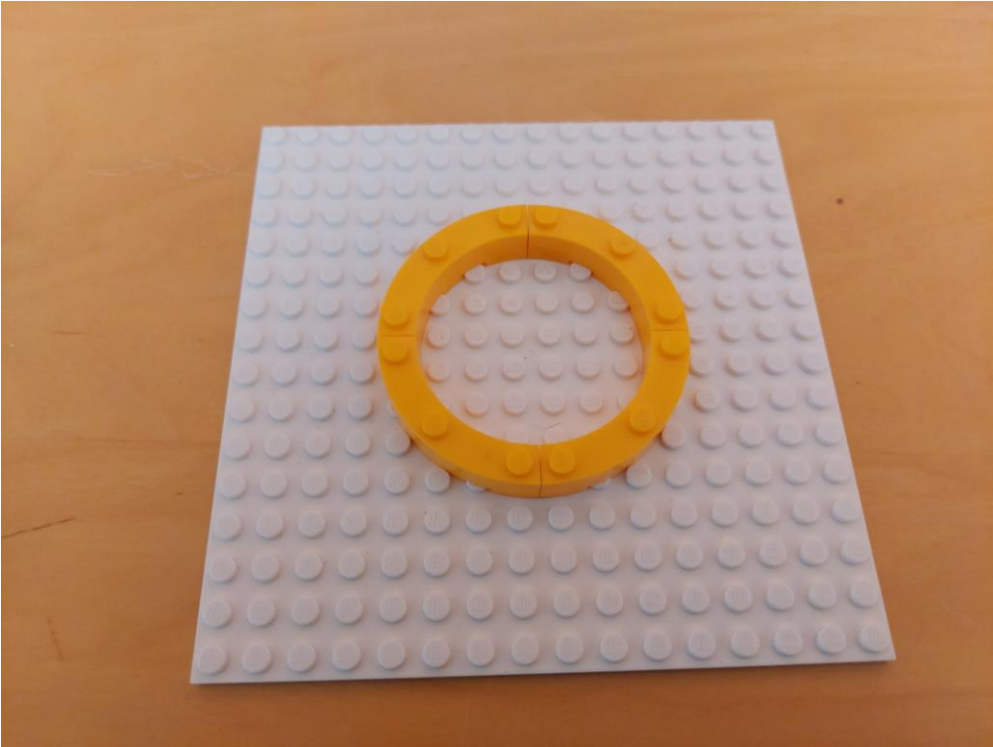
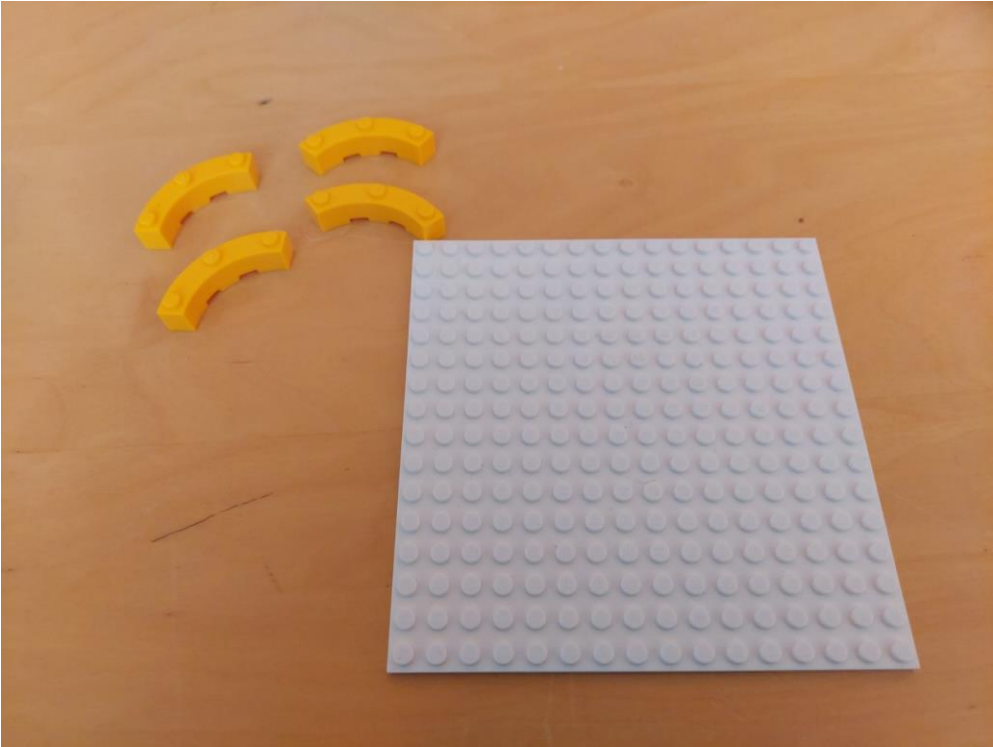
K čemu se používají jeřáby? Viděl jsi skutečný jeřáb? Co přepravoval? Zkus odhadnout jeho výšku. Postavme si a naprogramujme jeřáb pro přepravu nákladů. Jeřáb bude mít dvě krajní polohy. Spouštět se bude poklepáním.

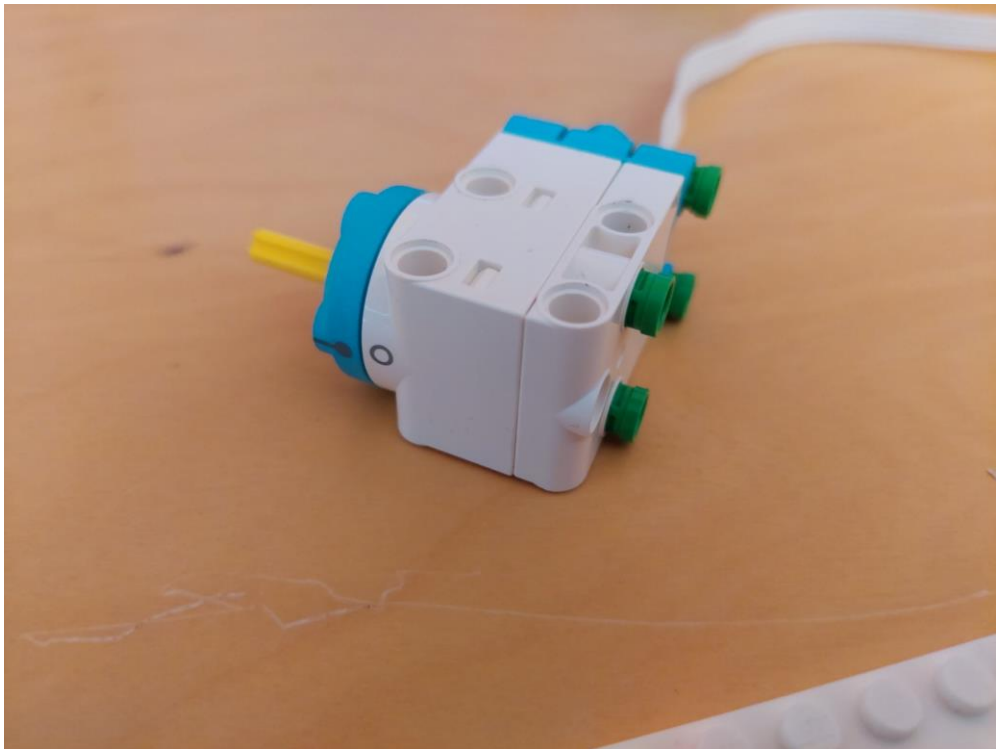
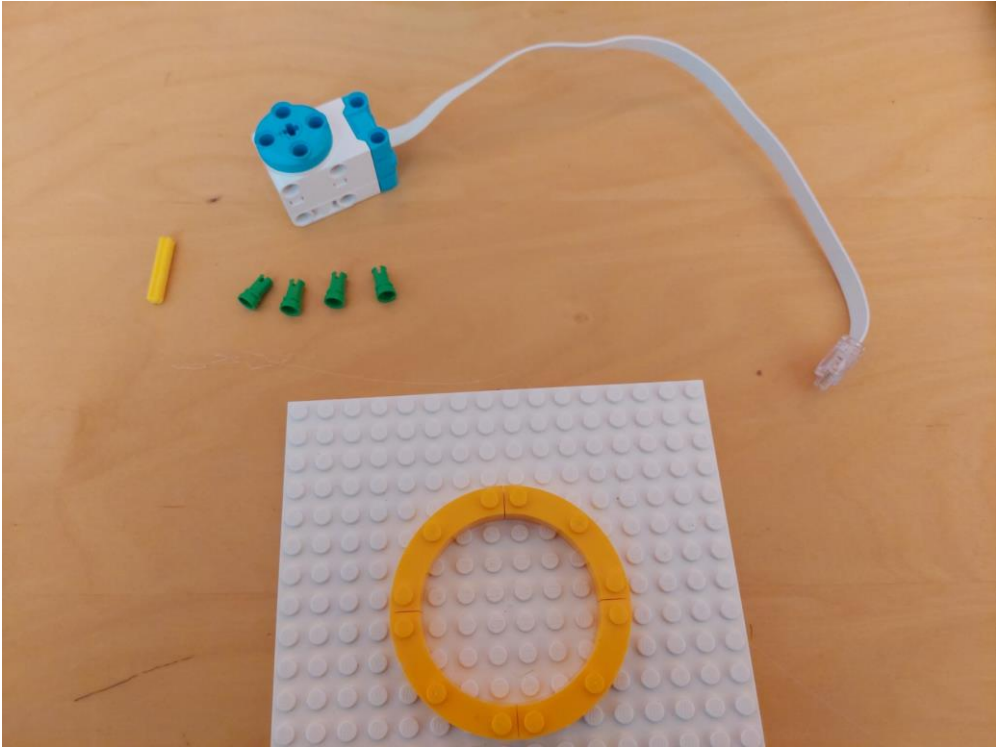


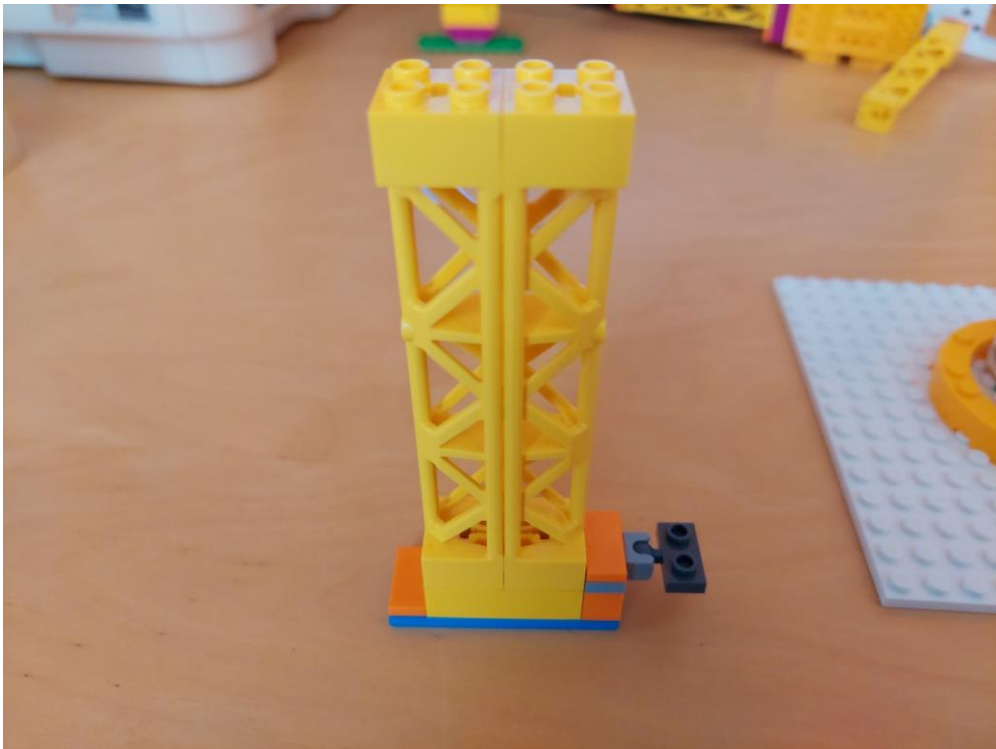
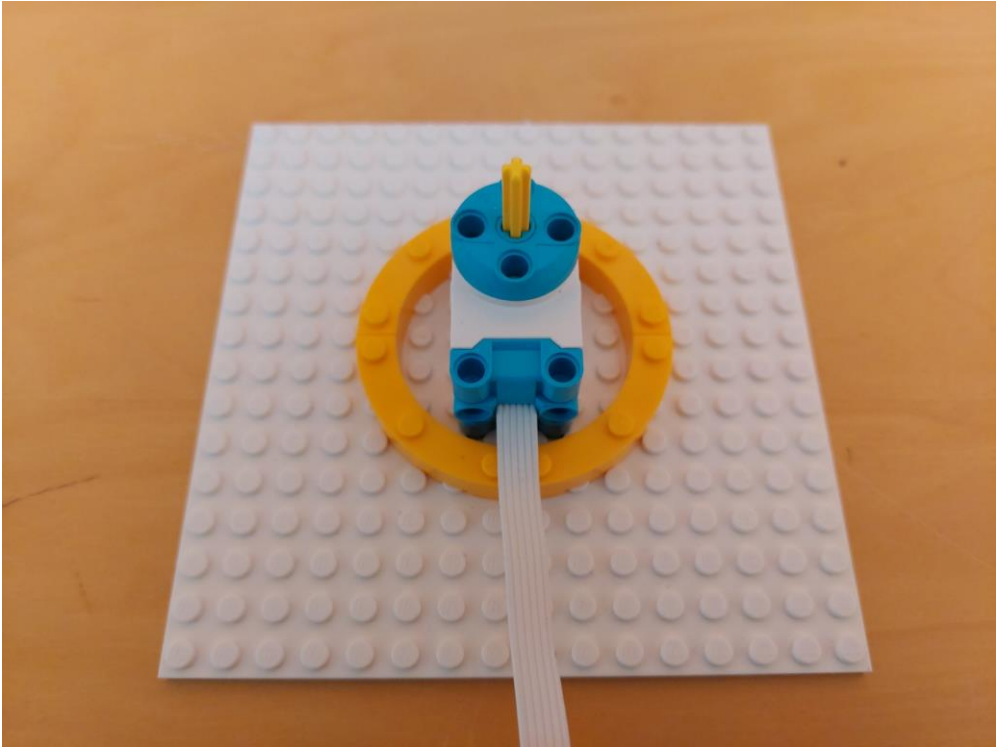
Tento příklad je už pro zkušené programátory a konstruktéry, ale to už přeci jsi. Jen musíš přidat provázek ten ve stavebnici nenajdeš.

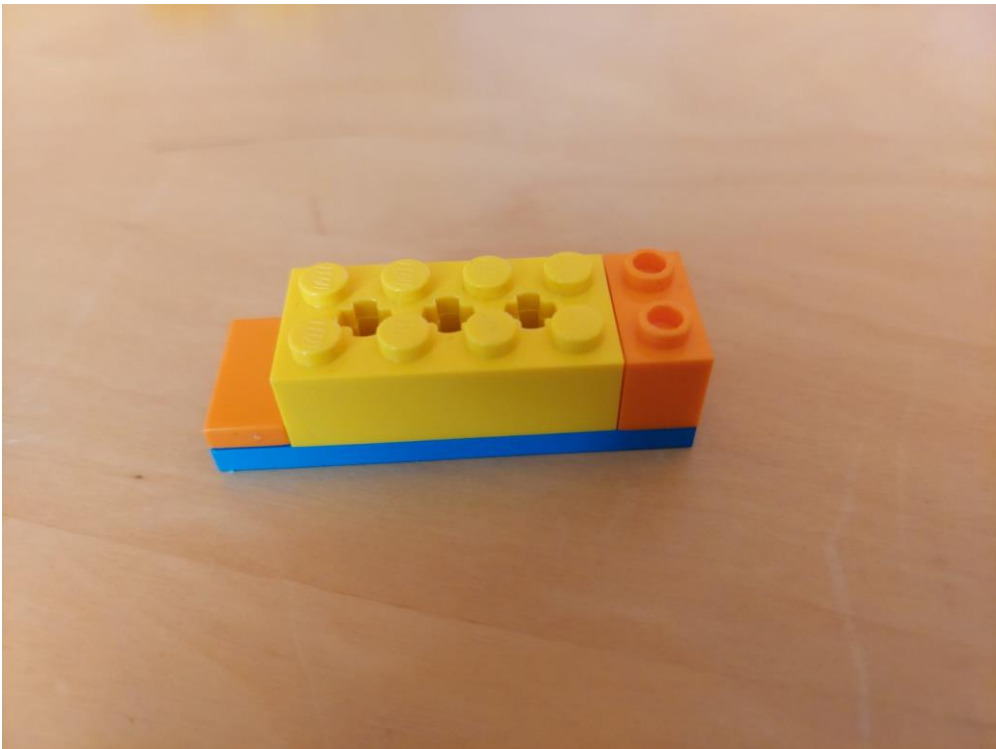
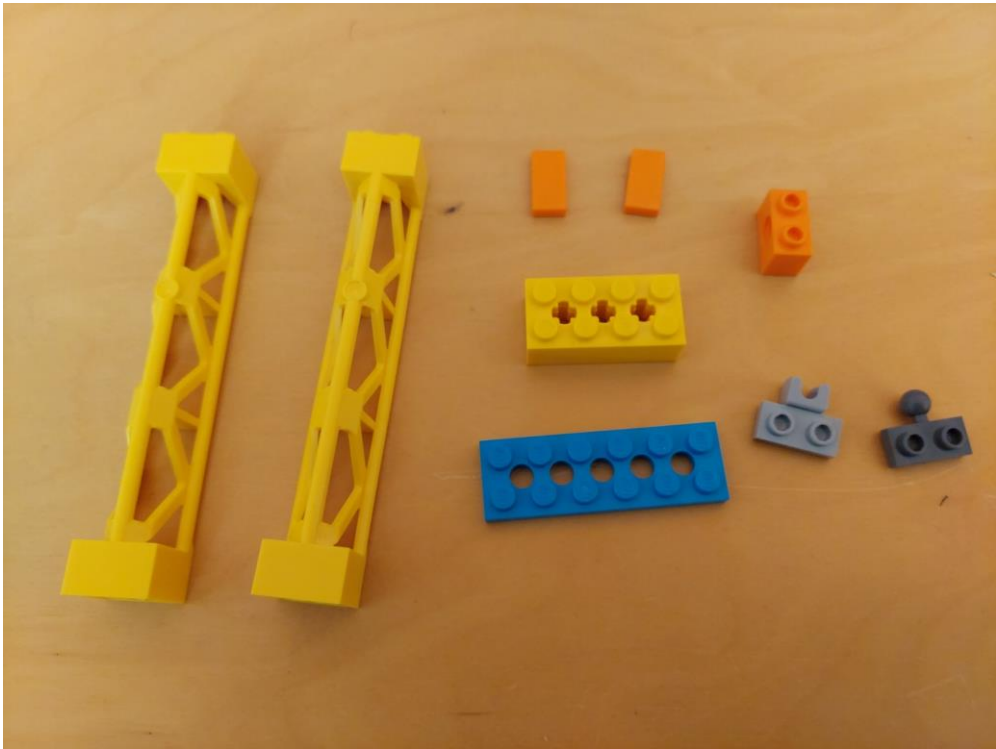


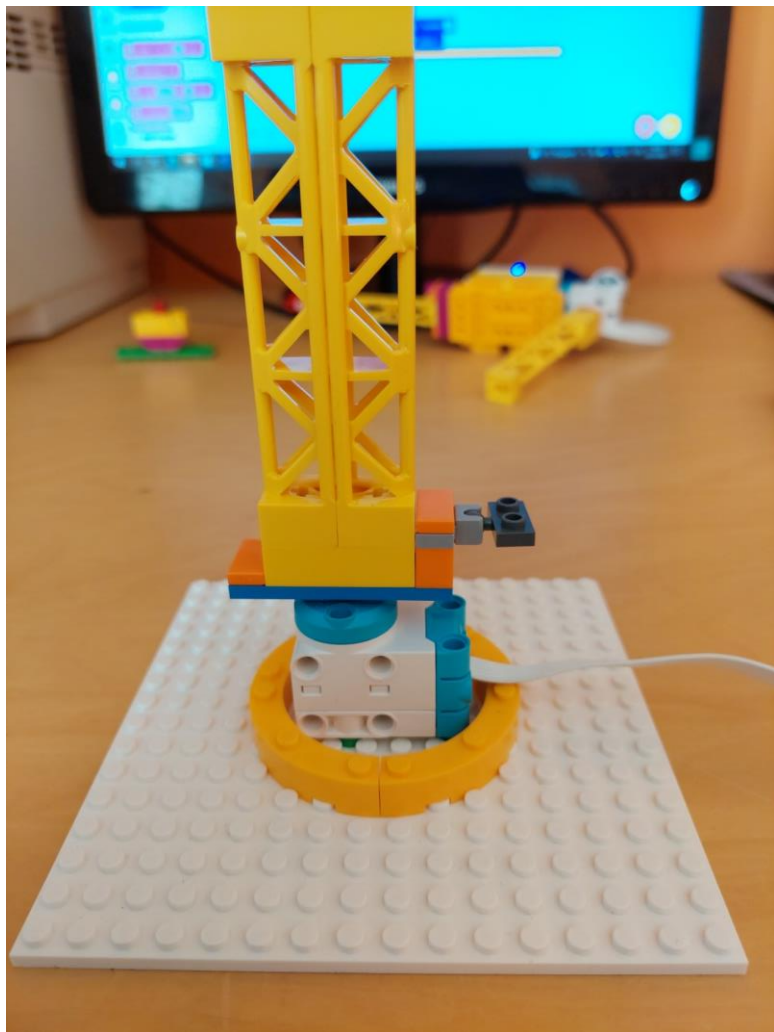
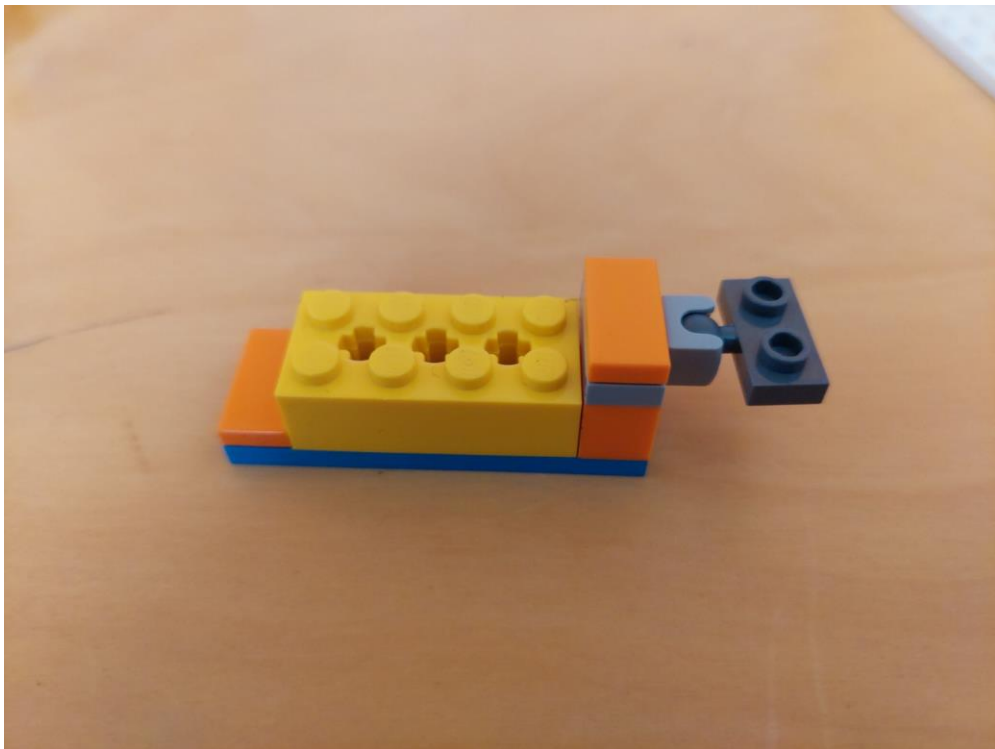
Stavba jeřábu

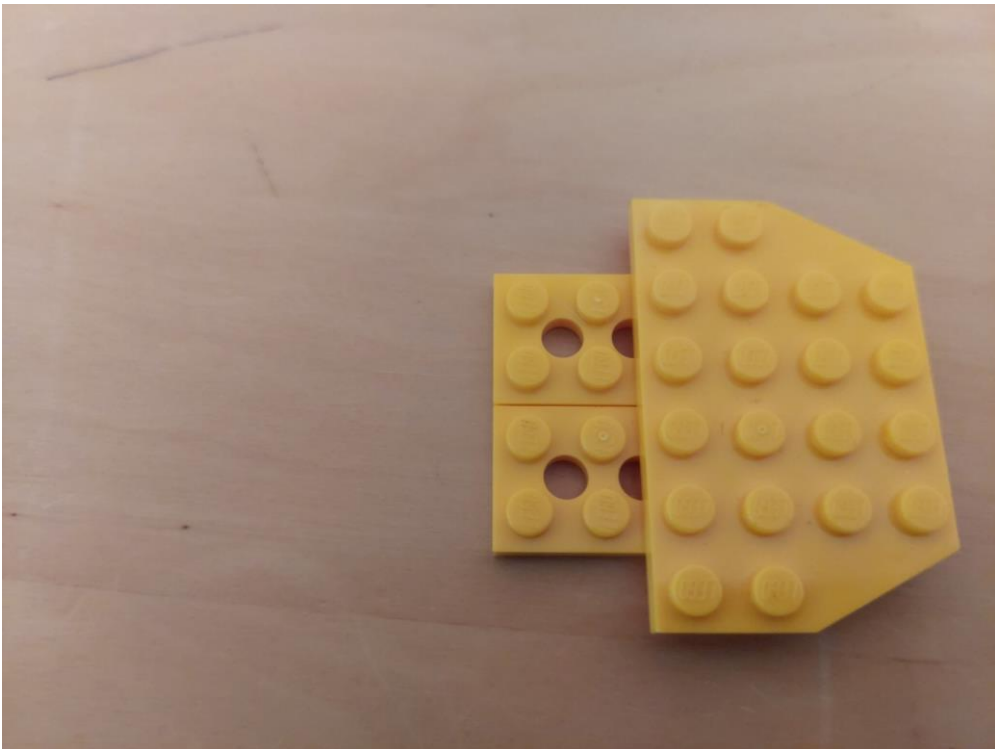
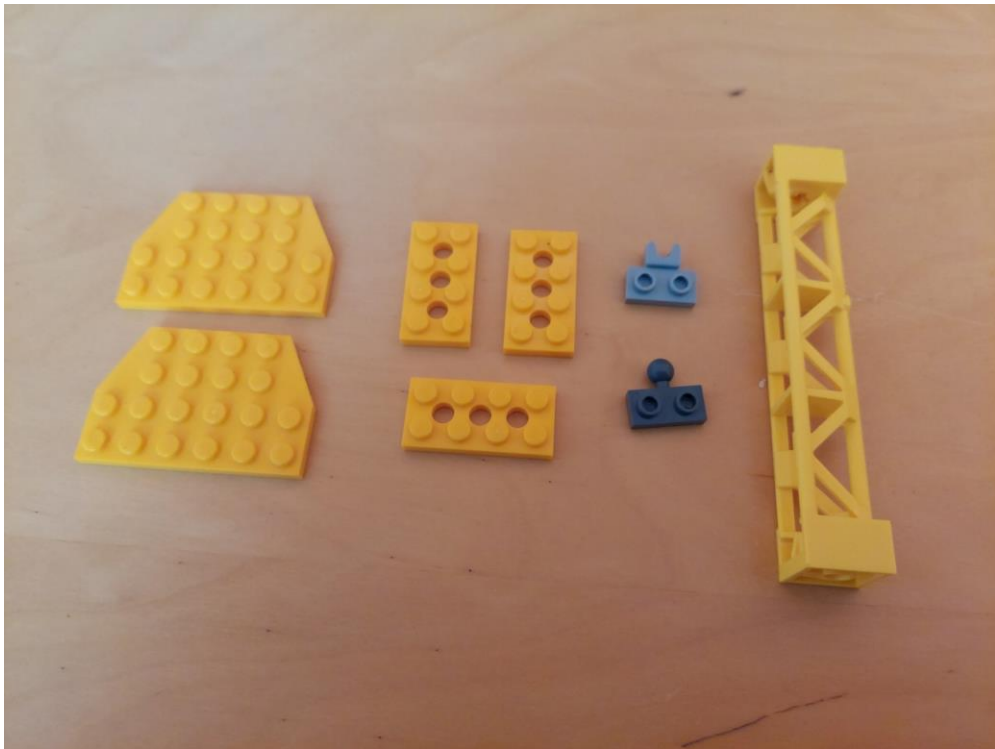




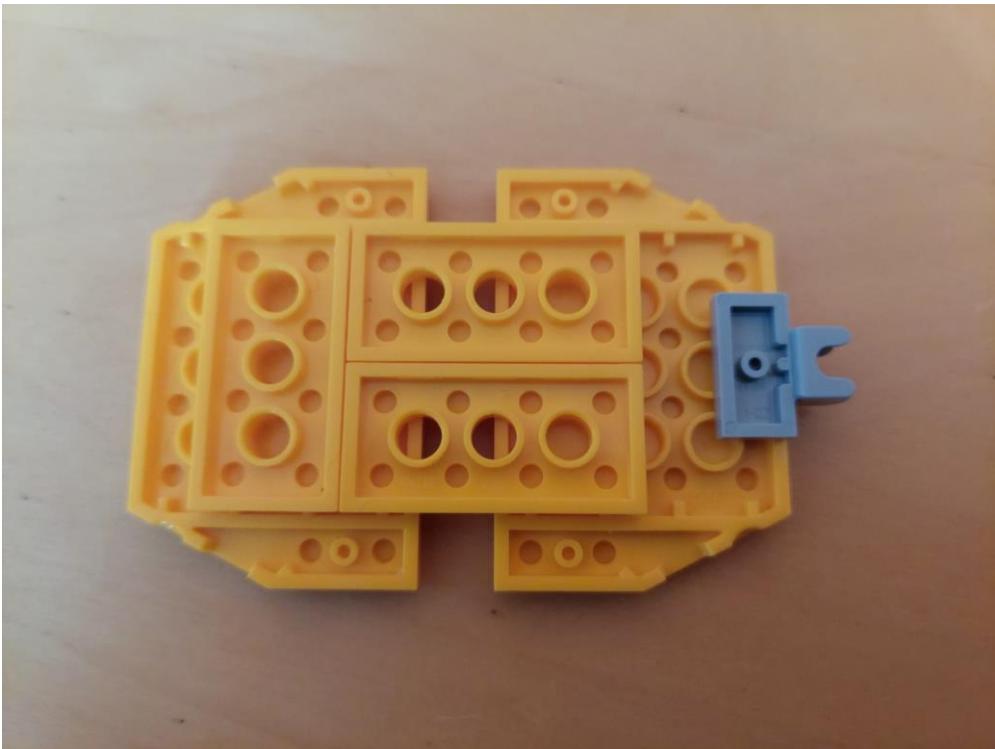
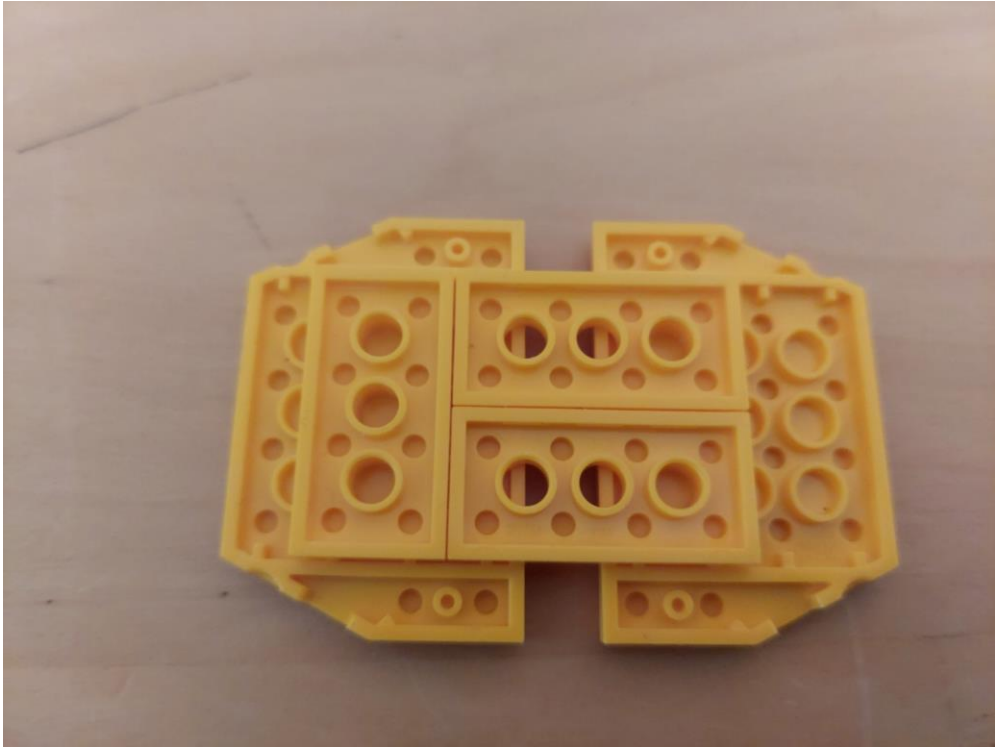


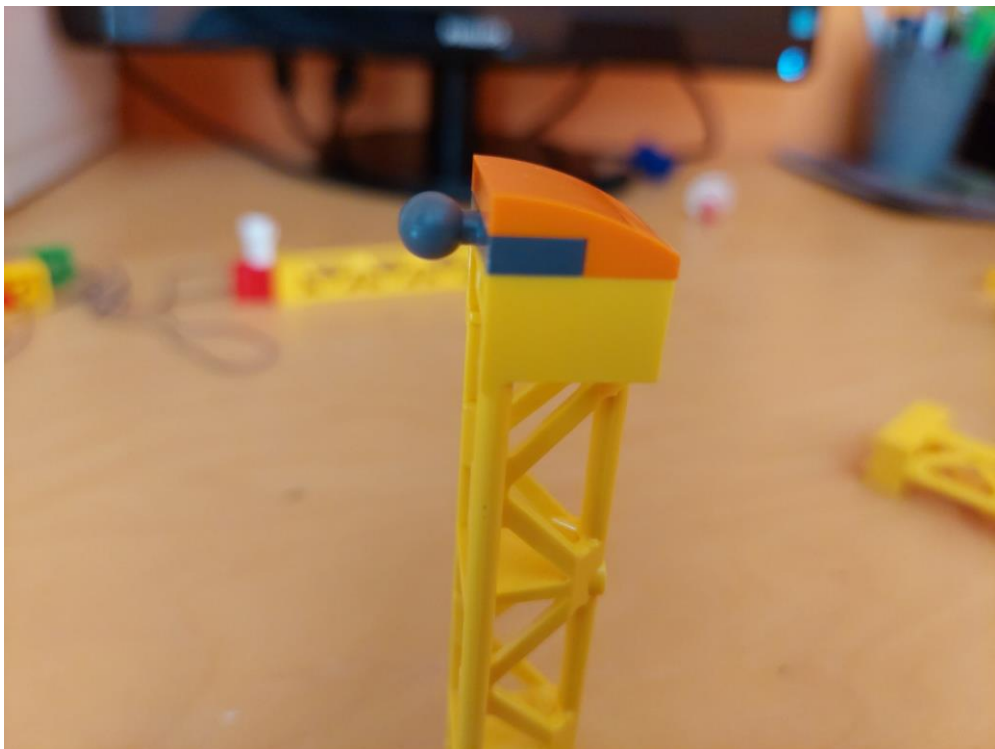
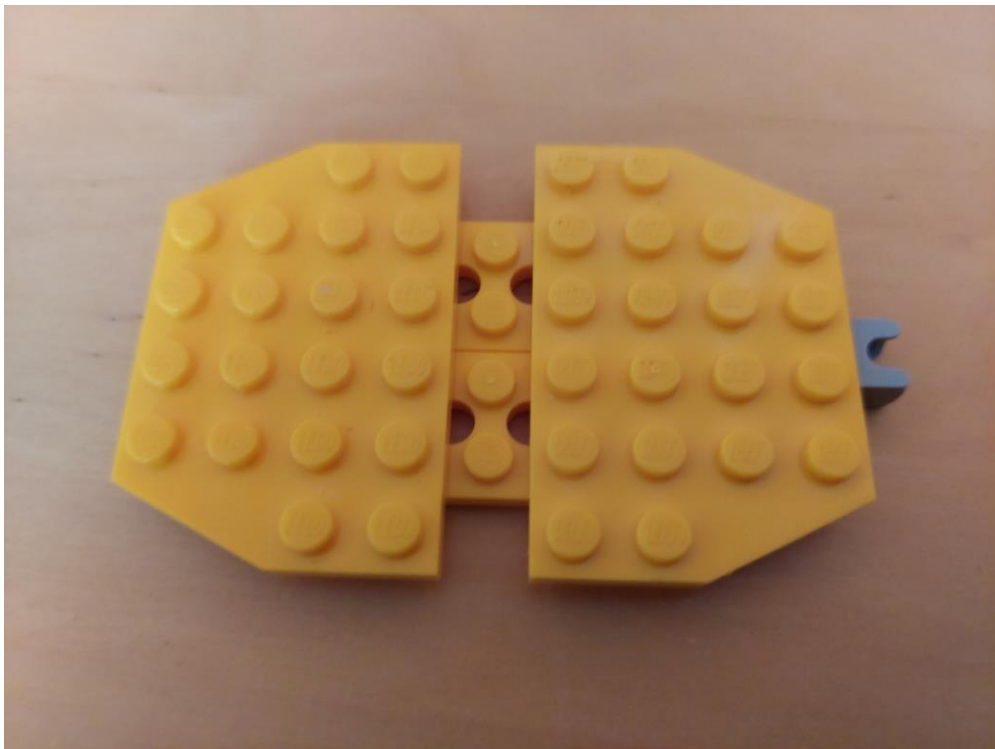




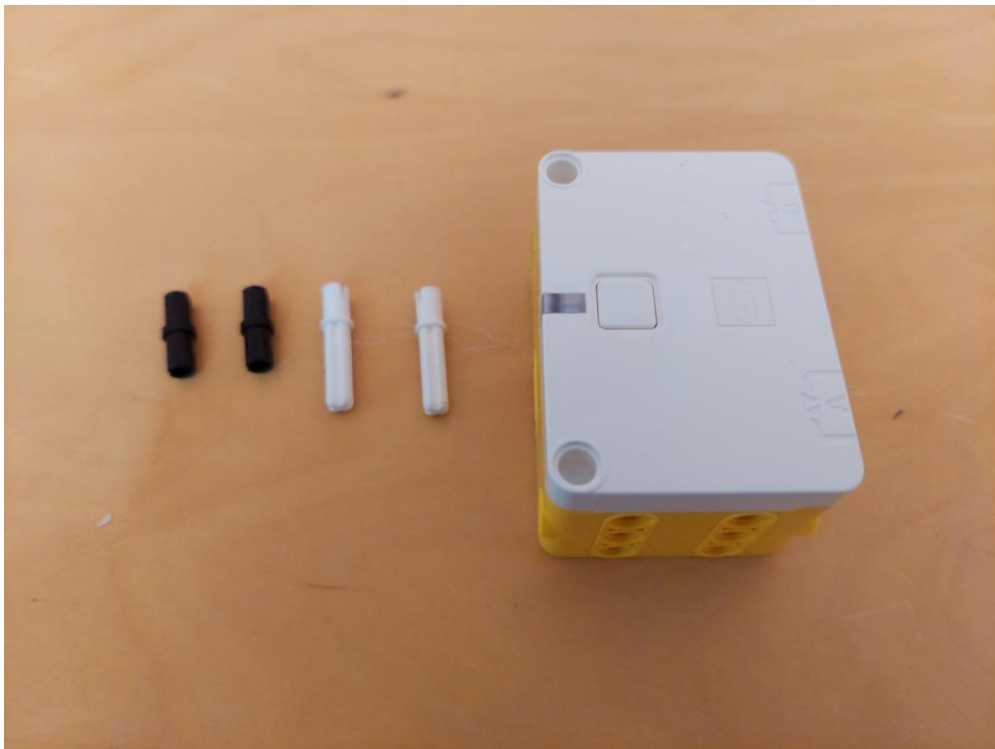




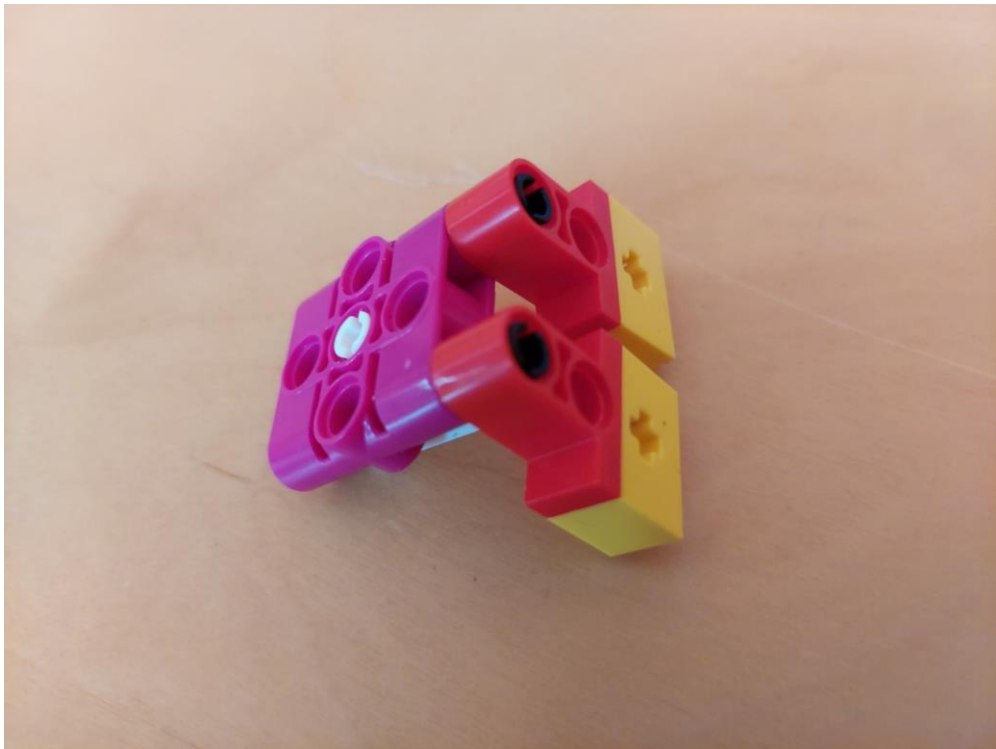
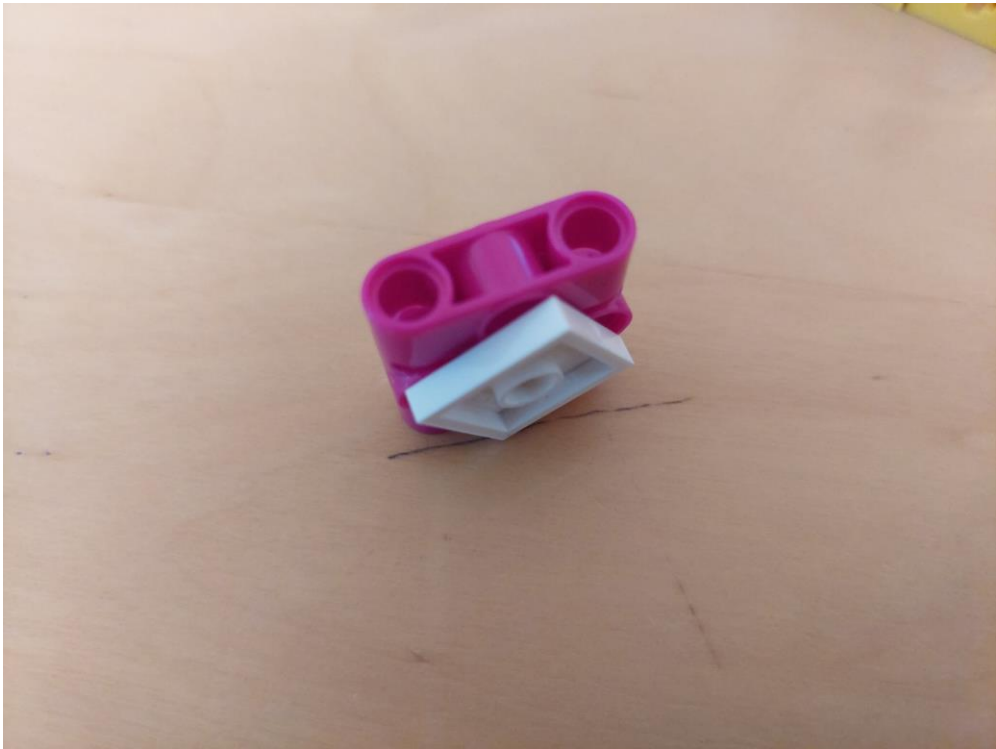


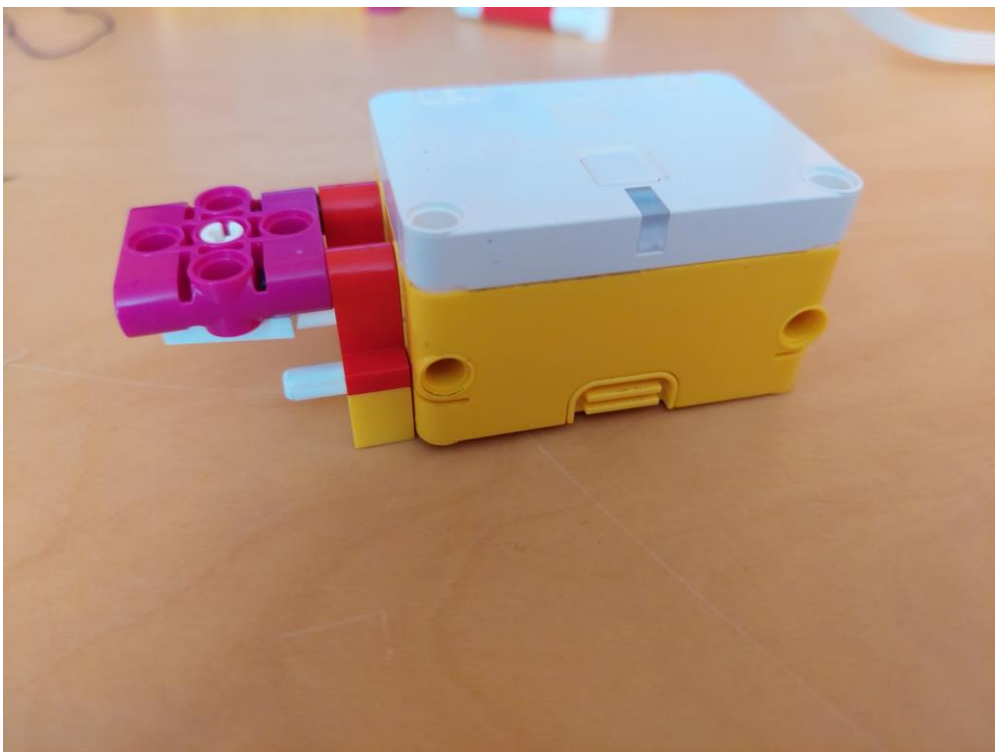
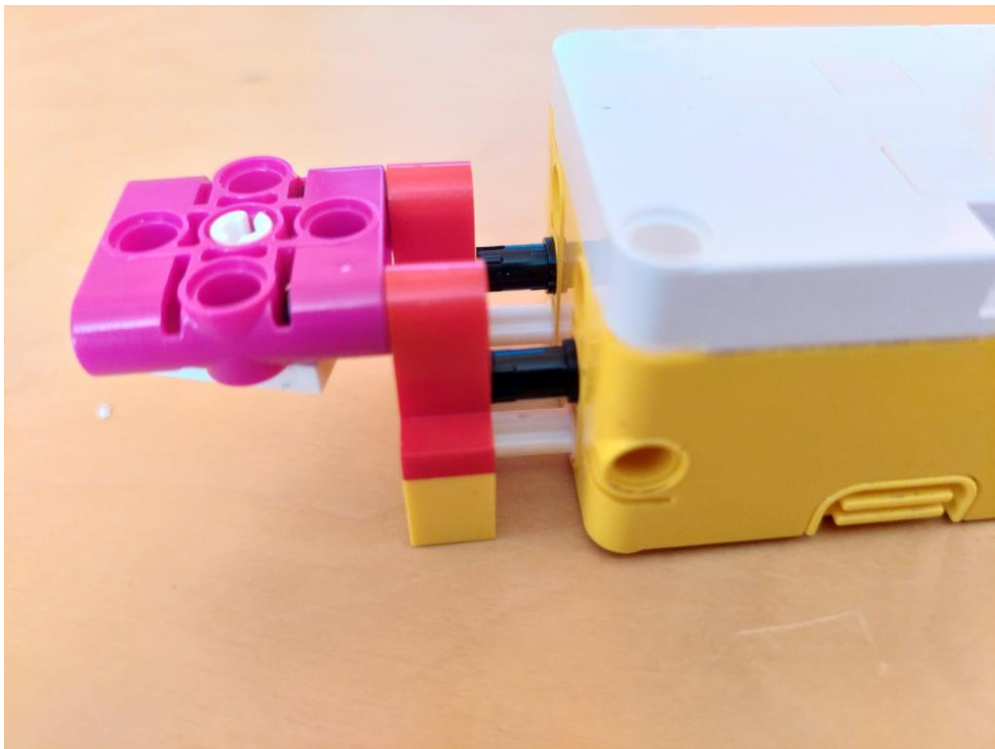


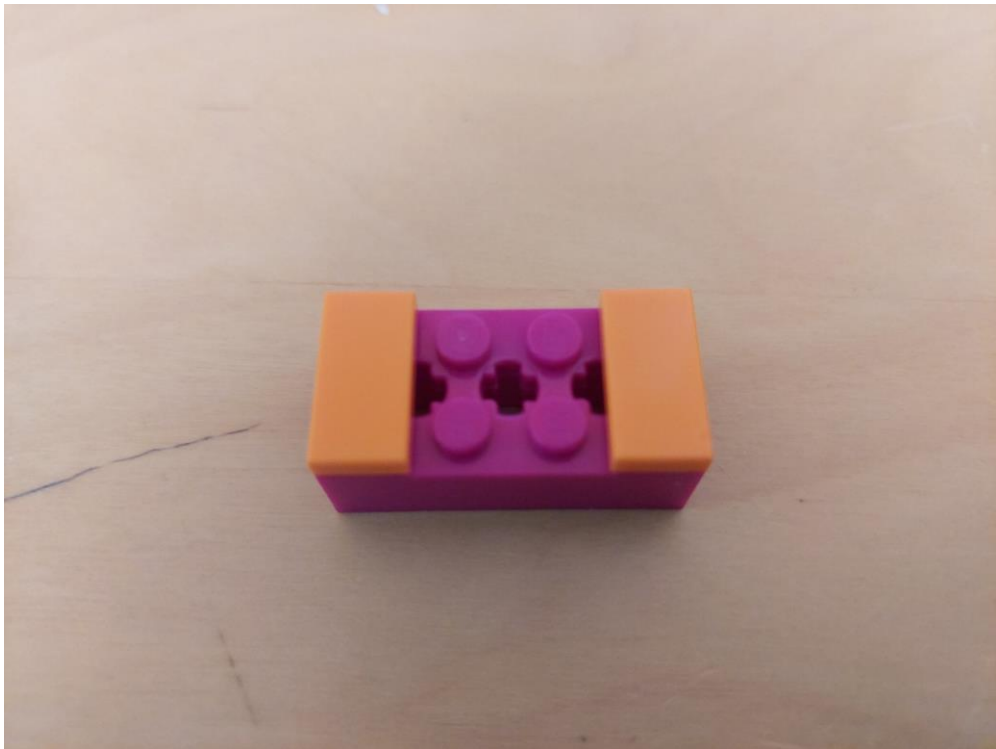
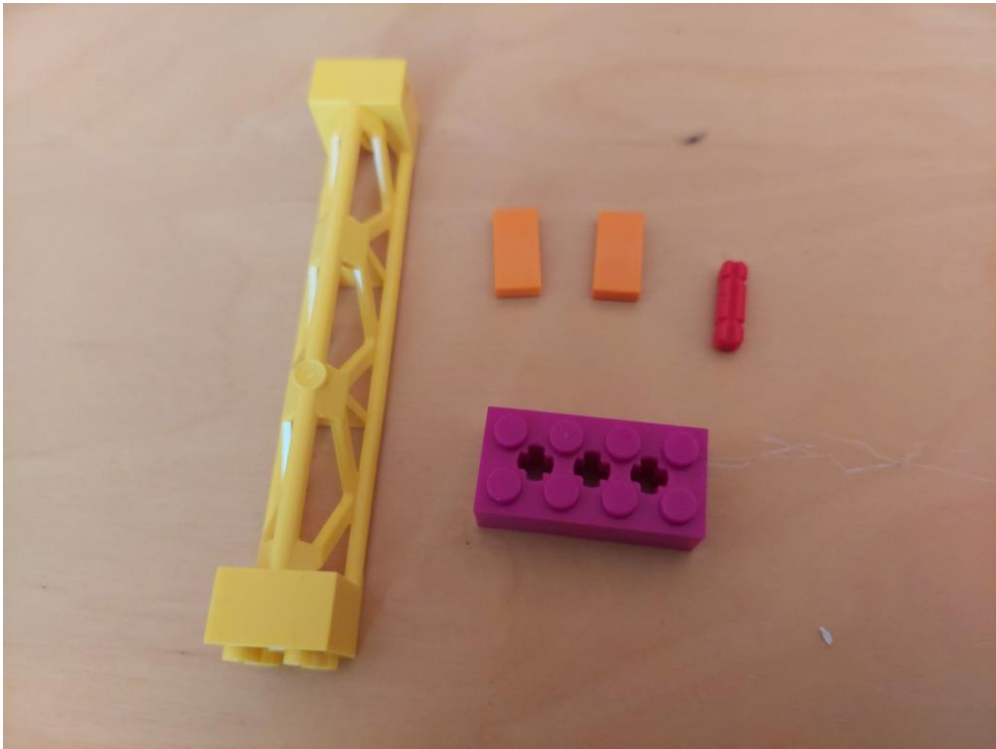




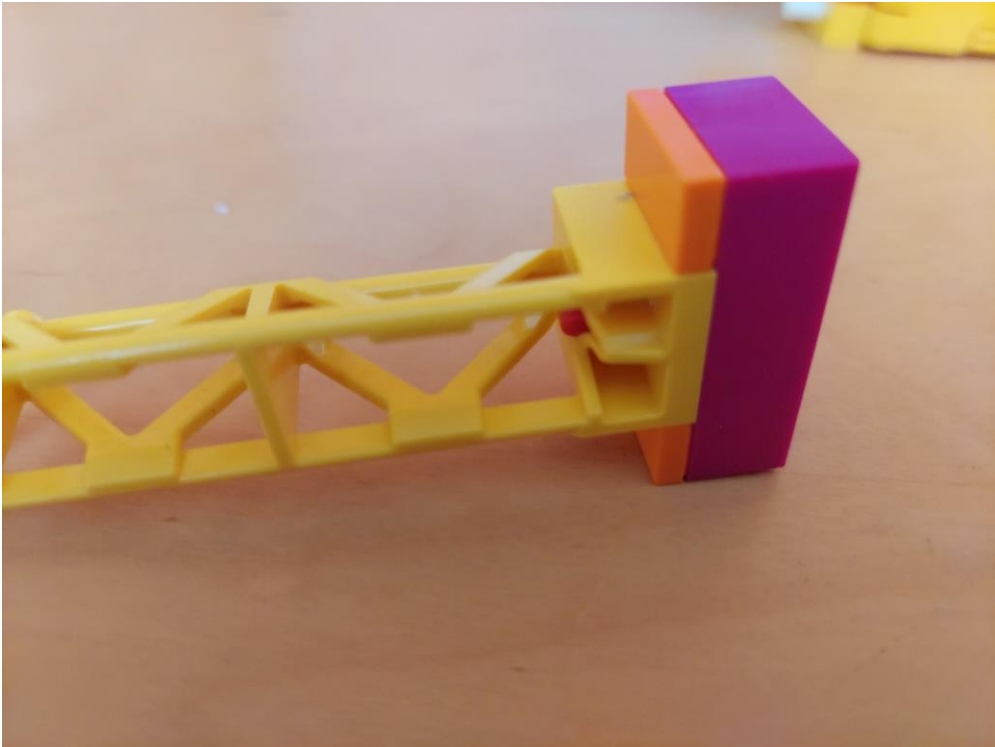
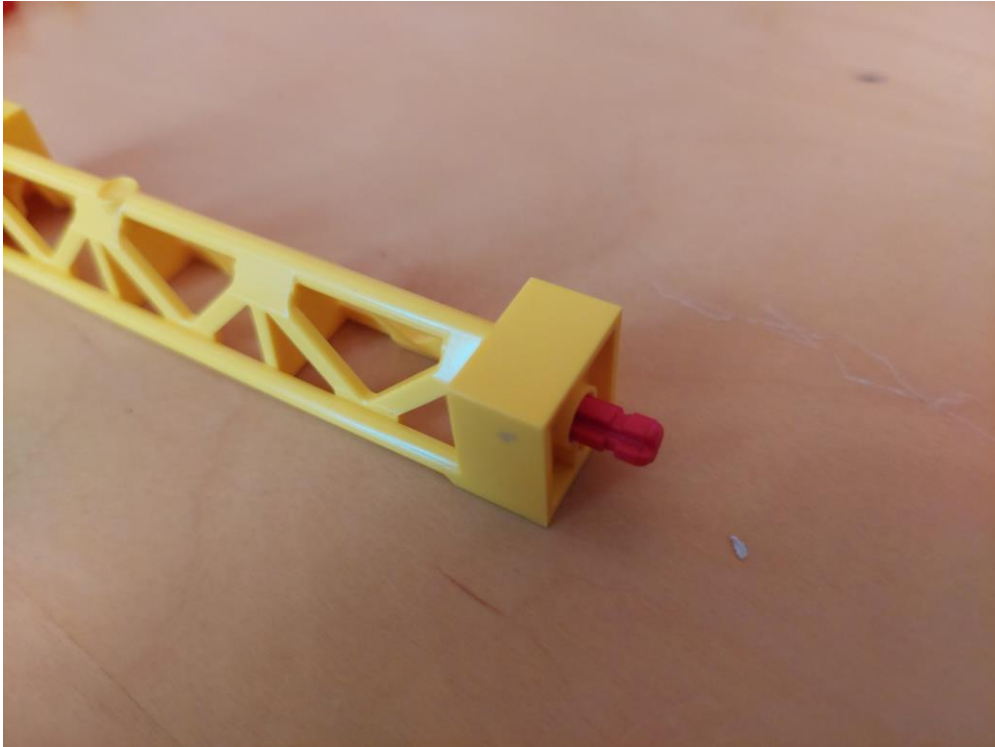


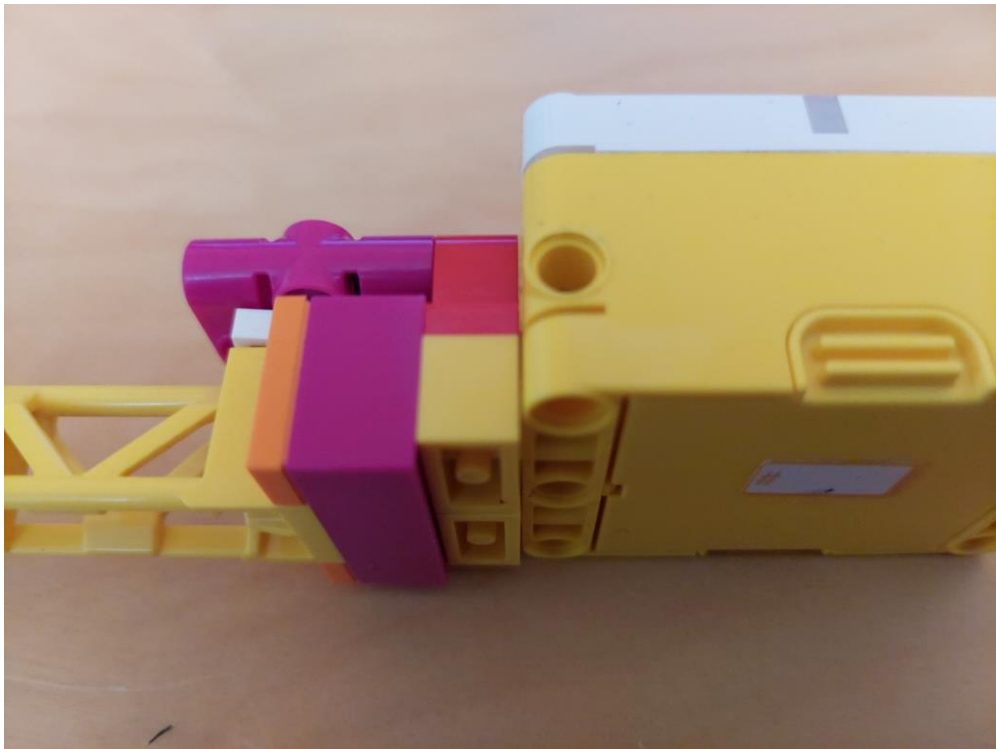
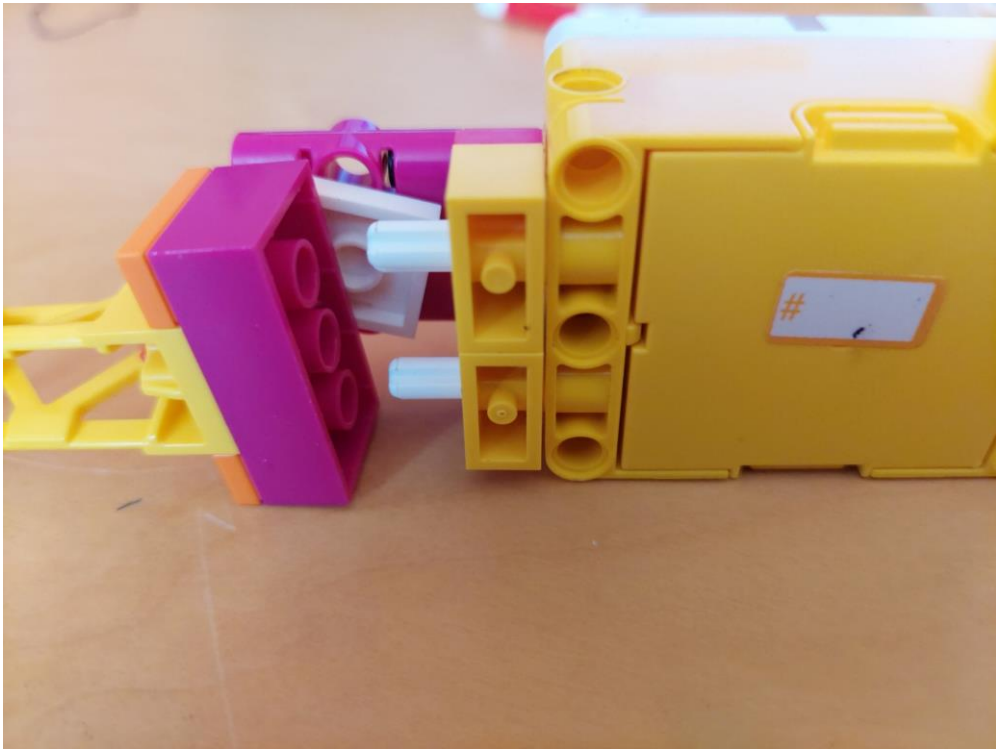


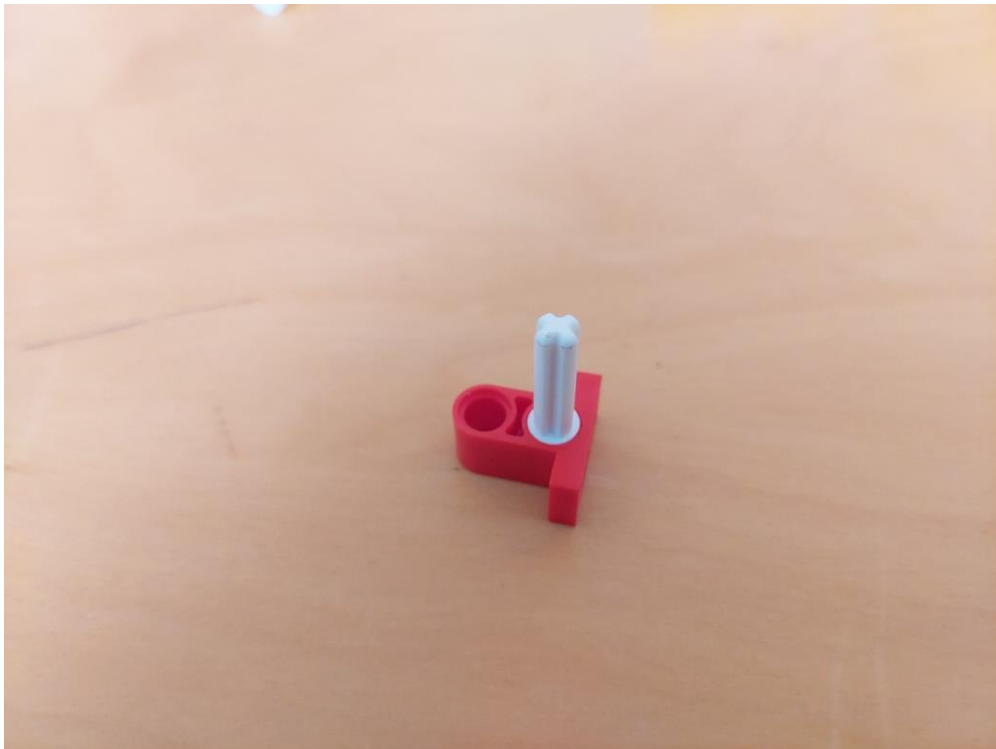
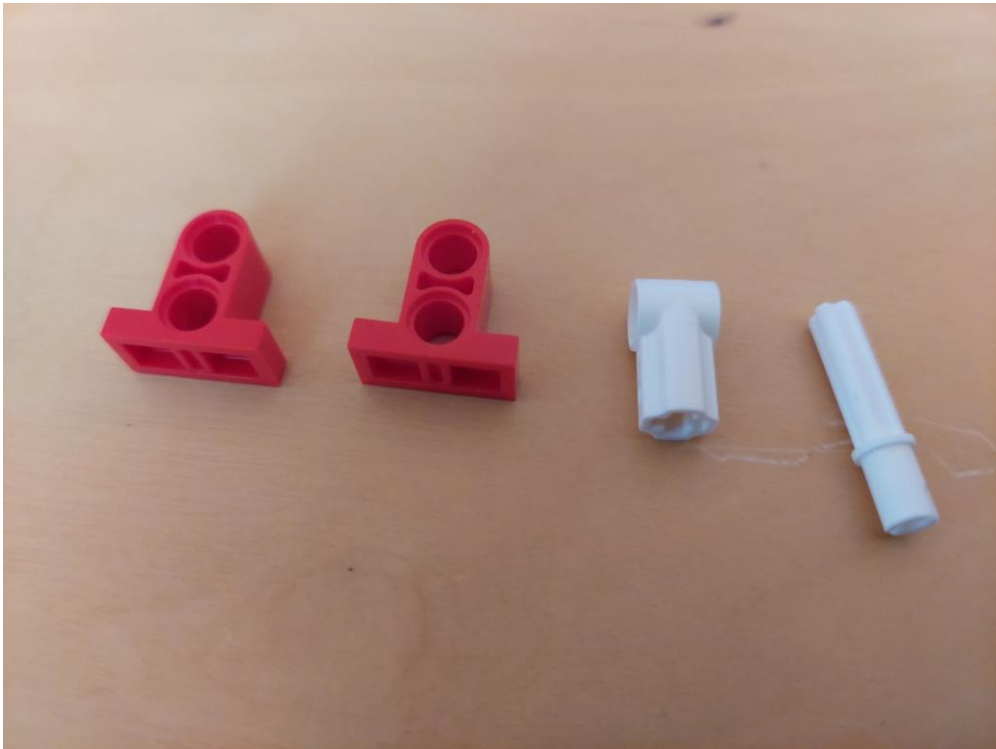


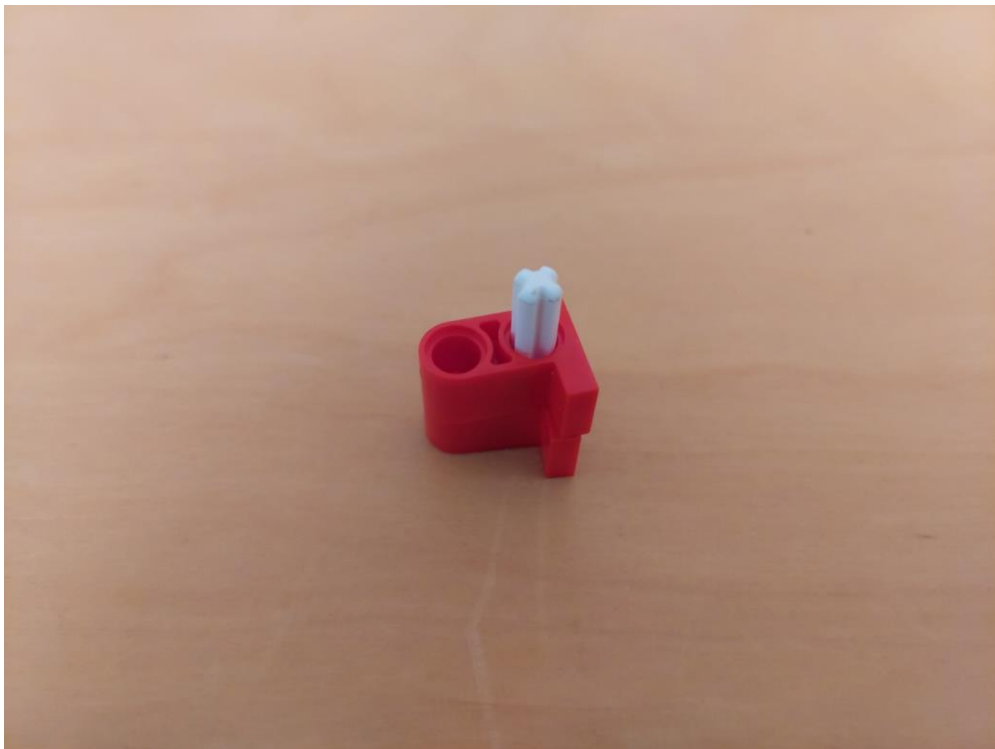


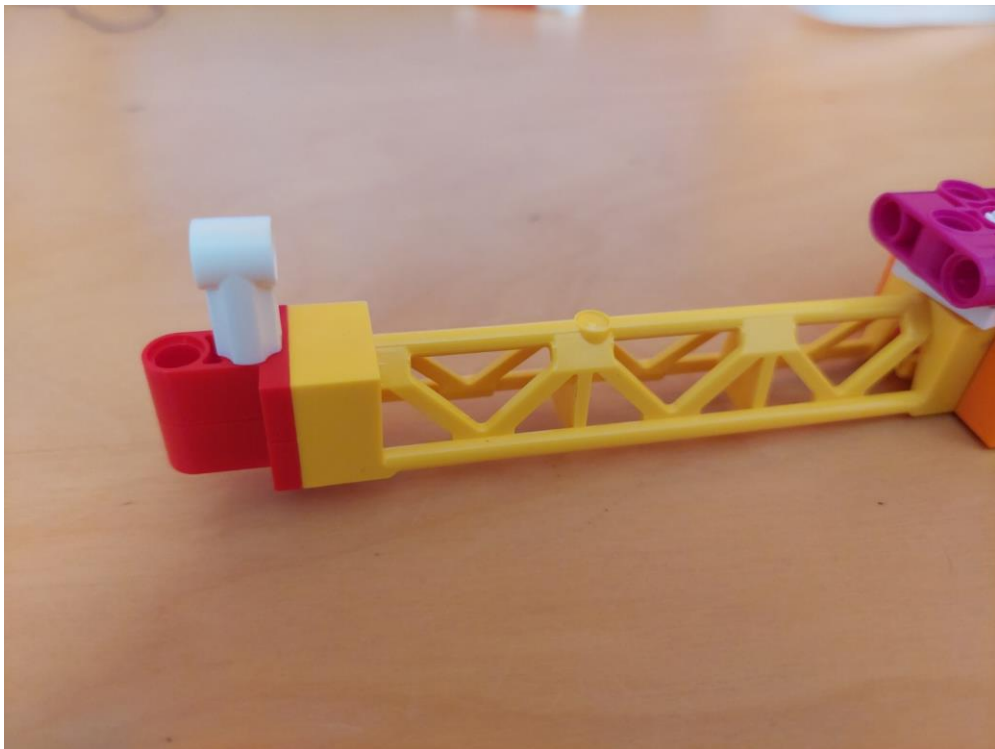






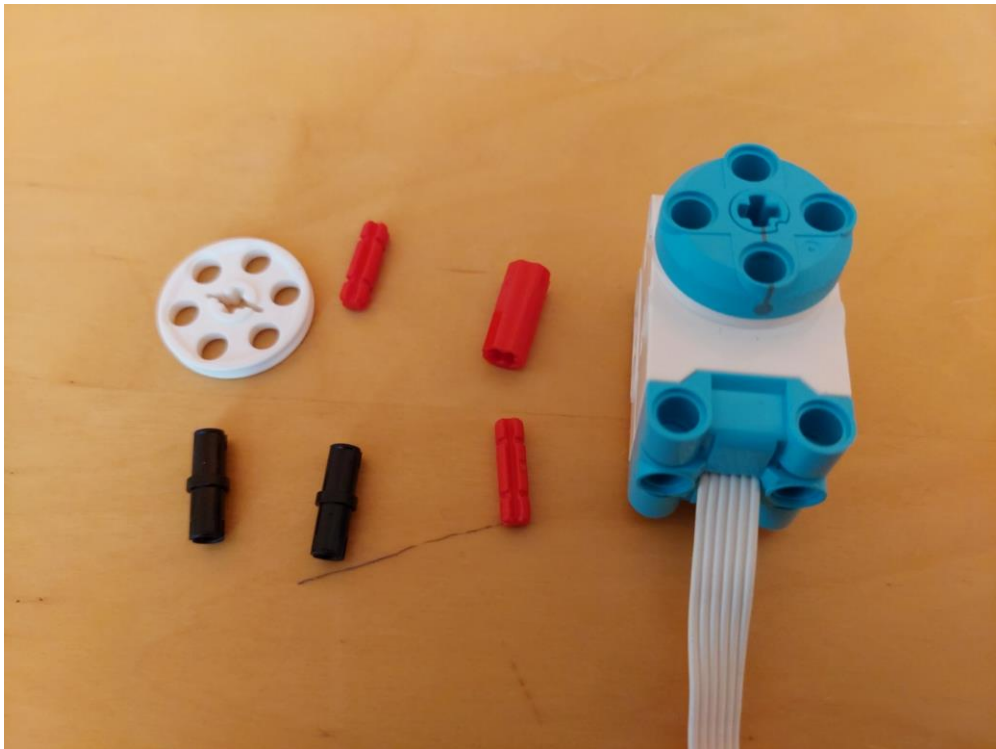
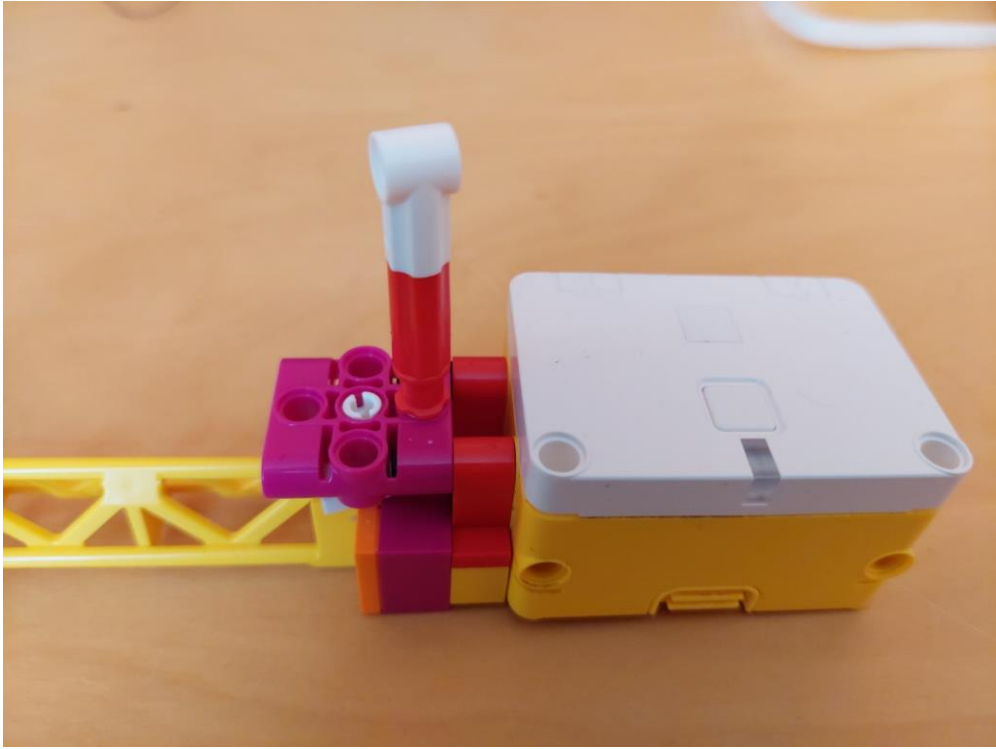




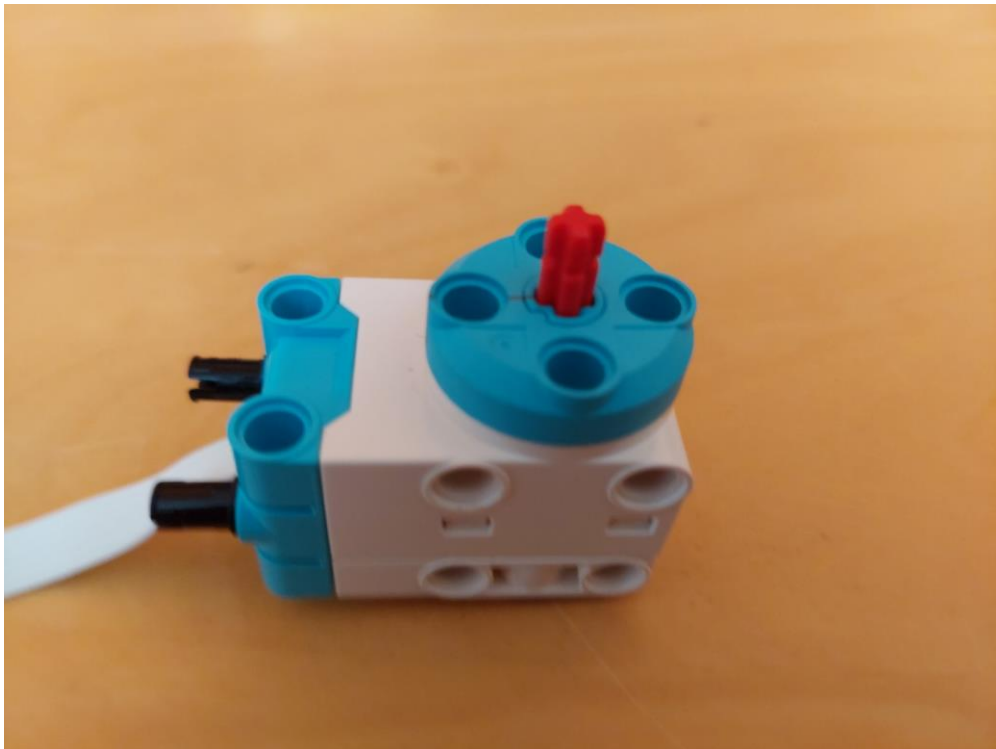
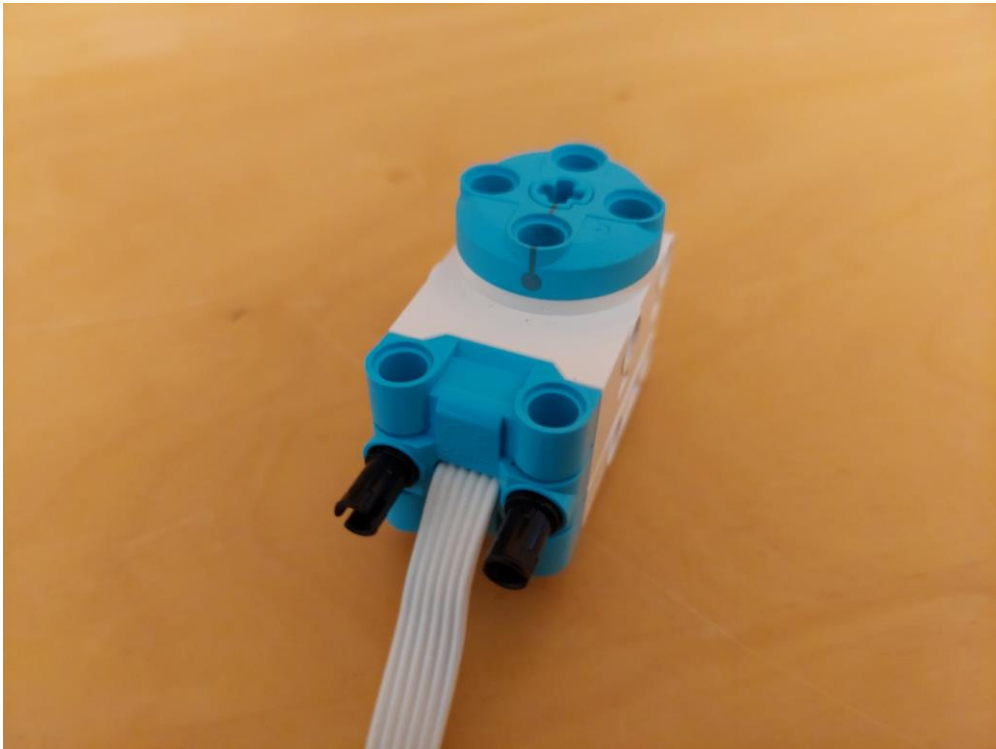


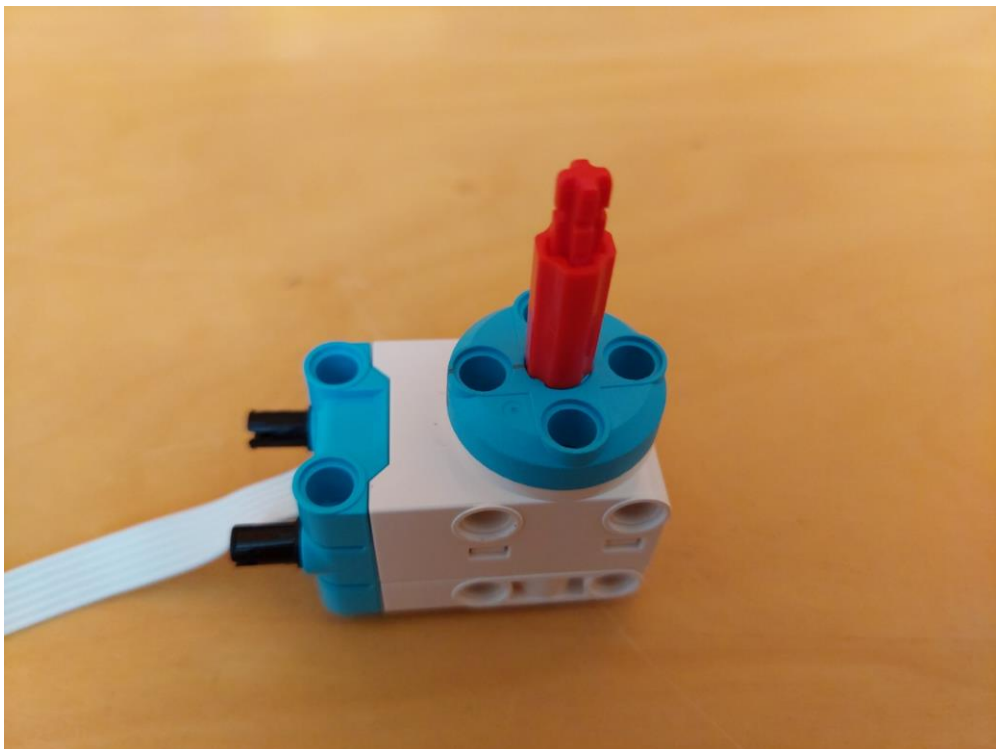
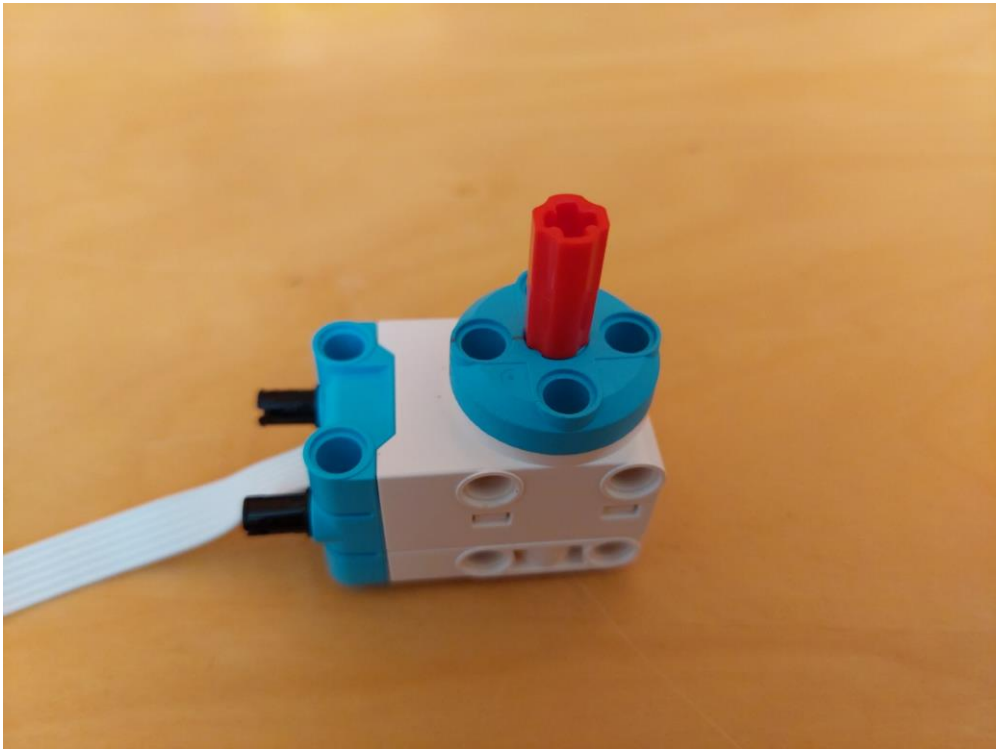


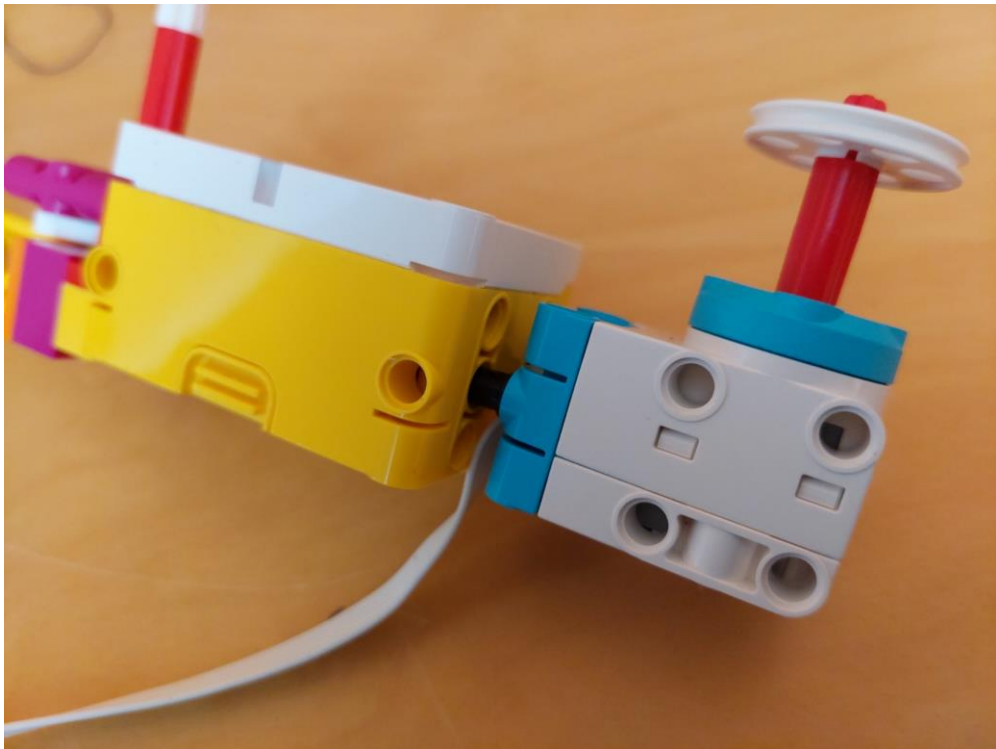
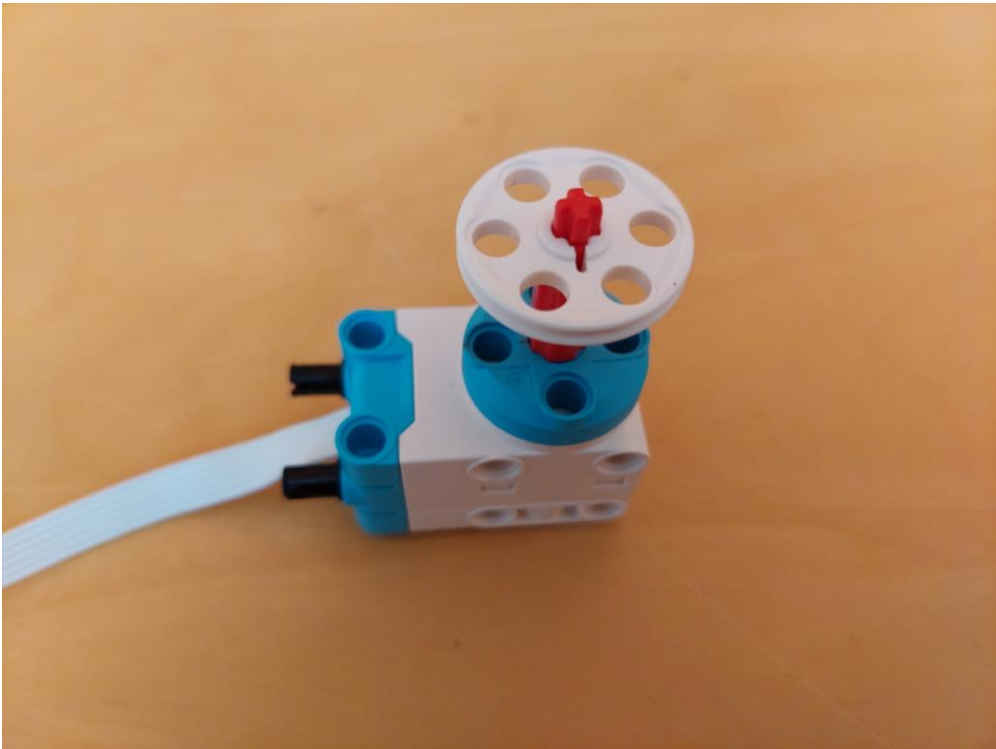


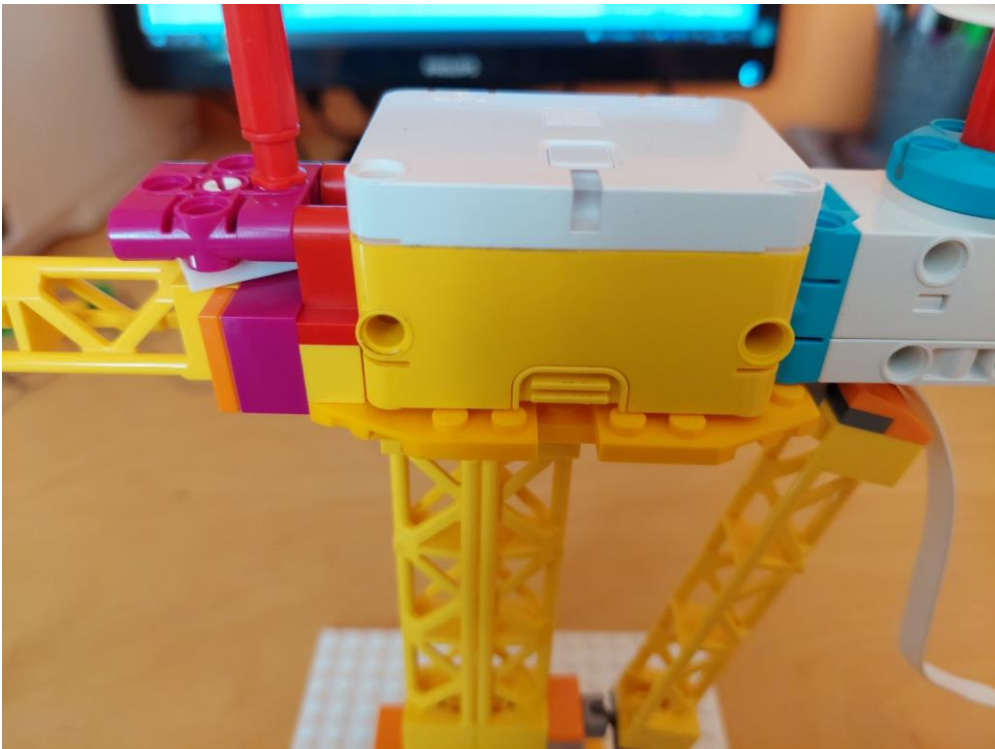
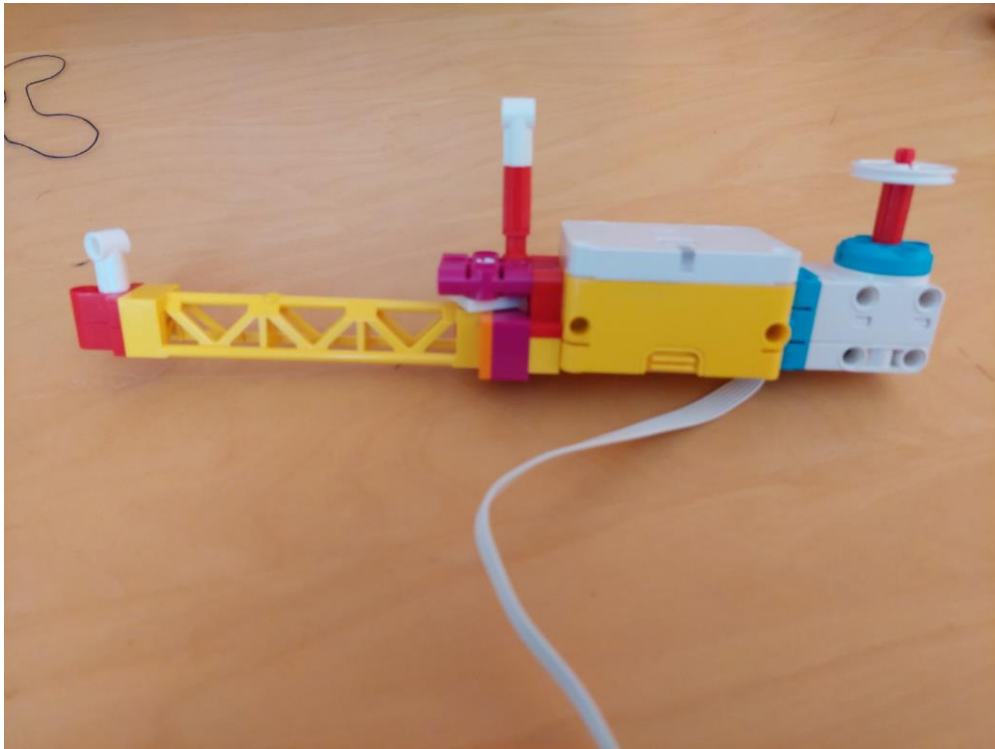


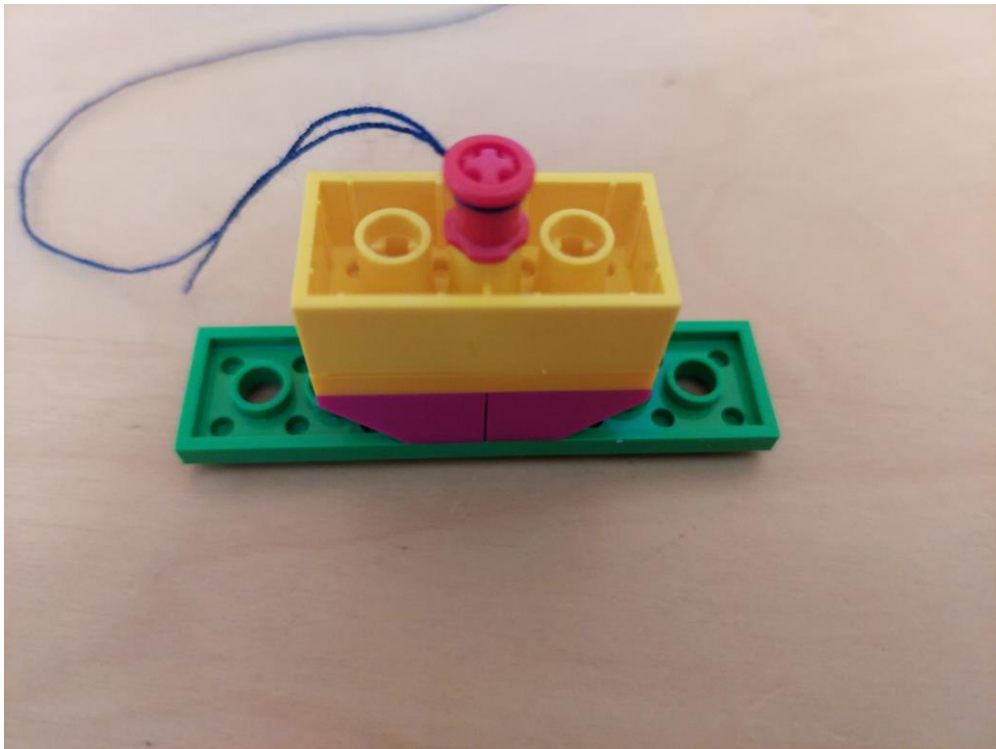
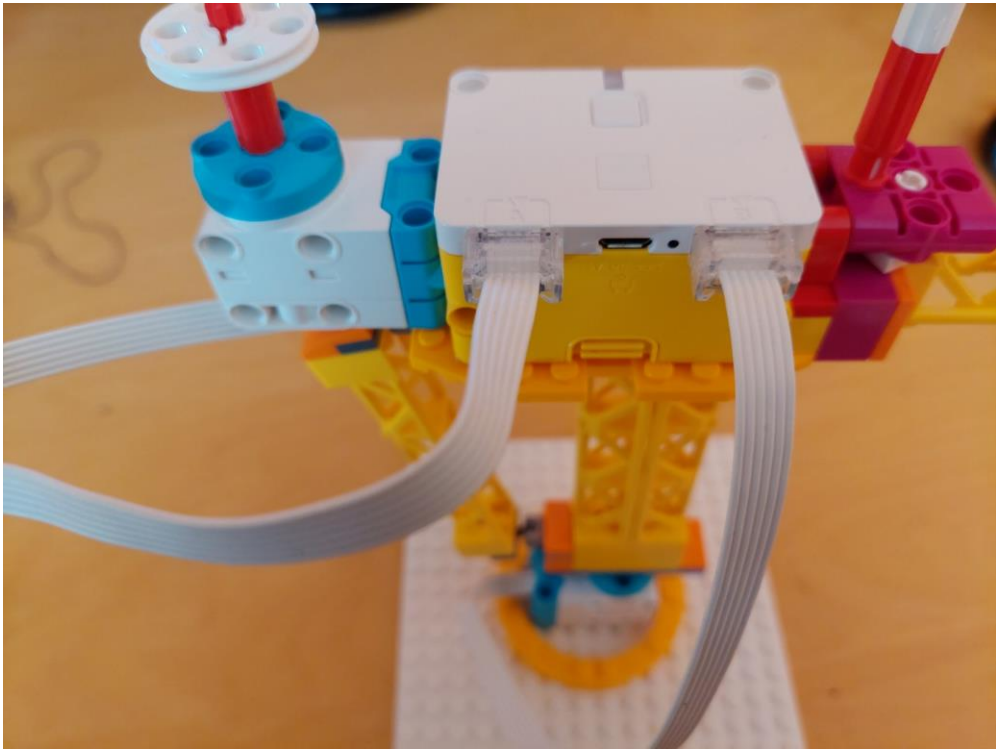










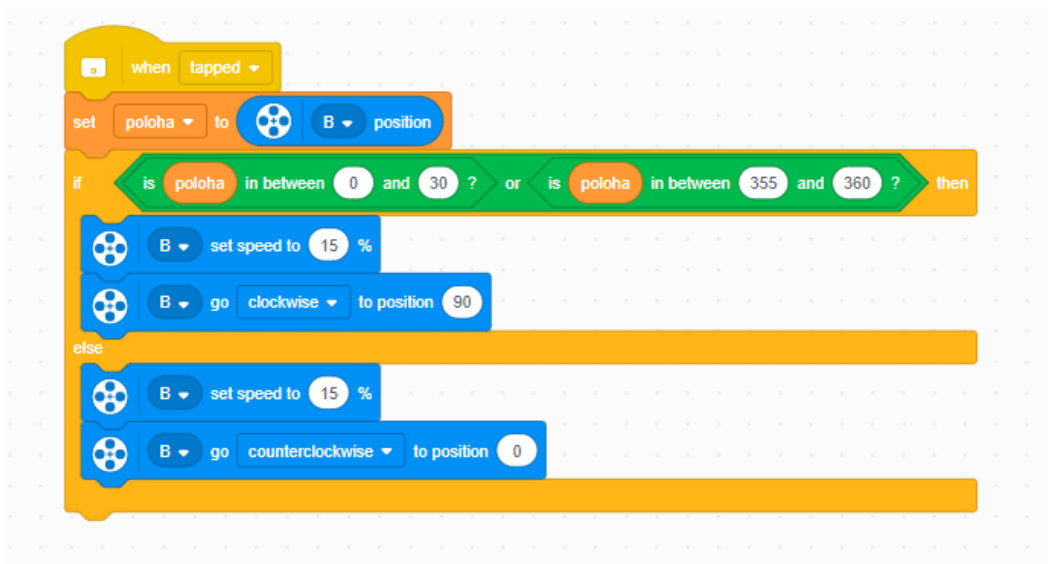


## Programování

Pokud máš jeřáb postavený můžeme začít programovat. Úkol je jednoduchý. Na jednom místě náklad vyzvednout a přemístit jej na druhé místo. A startovat budeme pomocí poklepání na HUB. Jde o složitější úkol, ale pokud jsi došel až sem, nebude to pro tebe těžké.

Pokud náhodou nevíš, ukážu ti moji cestu. Není to jediná správná cesta. Klidně můžeš najít lepší. Programování si rozdělíme do menších kroků.

1. Prvním problémem je zjistit, kde se zrovna náš jeřáb nachází. Vytvoř si novou proměnnou (Variables) s názvem např. Poloha.
2. Do této proměnné uložíme aktuální polohu z motoru, který otáčí jeřábem. K tomu slouží příkaz *Set* a blok motoru *Position*.
3. Jeřáb se nemůže točit neustále dokola, protože by přišel o kabel. Takže jej musíme otáčet mezi pozicí 0 a 90 a to buď po směru, nebo proti směru hodinových ručiček. (clockwise, counterclockwise)
4. Pro nastavení směru využijeme funkci *If*, kde porovnáme výchozí polohu a rozhodneme jakým směrem se má otáčet. V mém případě, pokud je jeřáb blízko polohy 0 tak jej přesunu do této polohy, v opačném případě do polohy 90. Zde je ukázka programu:



5. Pokud vše funguje stačí již jen spočítat kolik otočení je potřeba na zvednutí nákladu a zase na jeho spuštění. Ale to už umíš.
6. A co by ještě mohl. Např. před zvednutím může zatroubit na upozornění. Lze i počítat kolik břemen jsme převezli, ale to již nechám na tvé fantazii.

```
when tapped
  set poloha to B position
  if is poloha in between 0 and 30 ? or is poloha in between 355 and 360 ? then
    A run for 5 rotations
    B set speed to 15 %
    B go clockwise to position 90
    A run for 5 rotations
  else
    A run for 5 rotations
    B set speed to 15 %
    B go counterclockwise to position 0
    A run for 5 rotations
```