

第五組 - 實驗一 尋跡投筆車

教師：黃漢邦、林沛群 教授

R09522849 陳品存

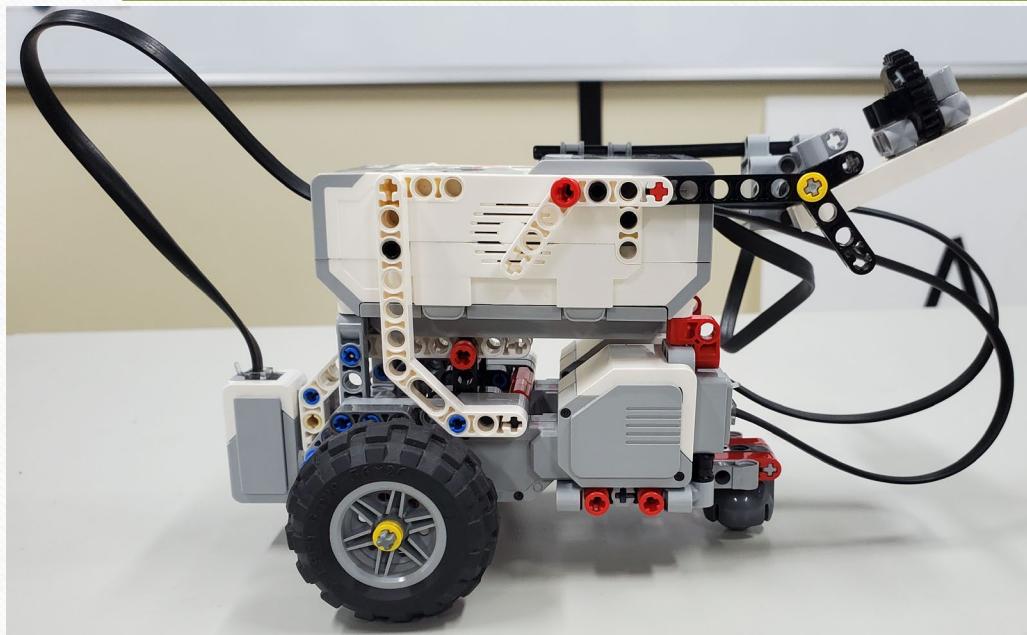
R10522801 吳政彥

R10522814 陳政豪

R10522834 戴承寧

車體機構

成品 (1/3)

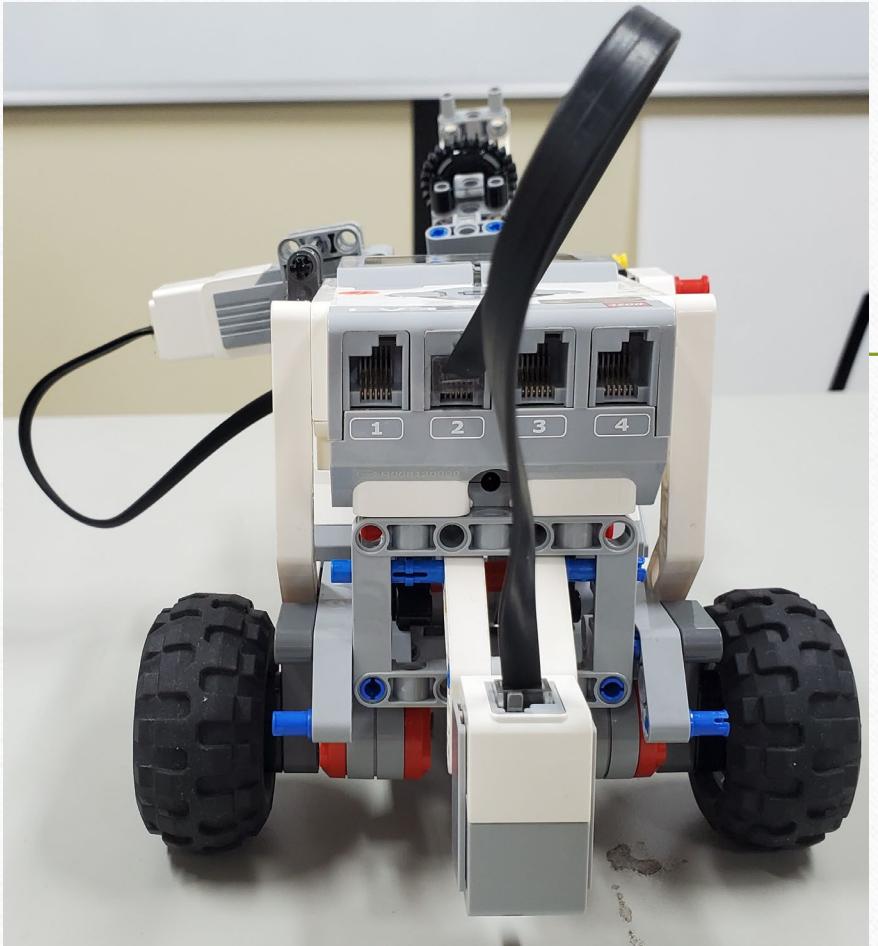


左視圖

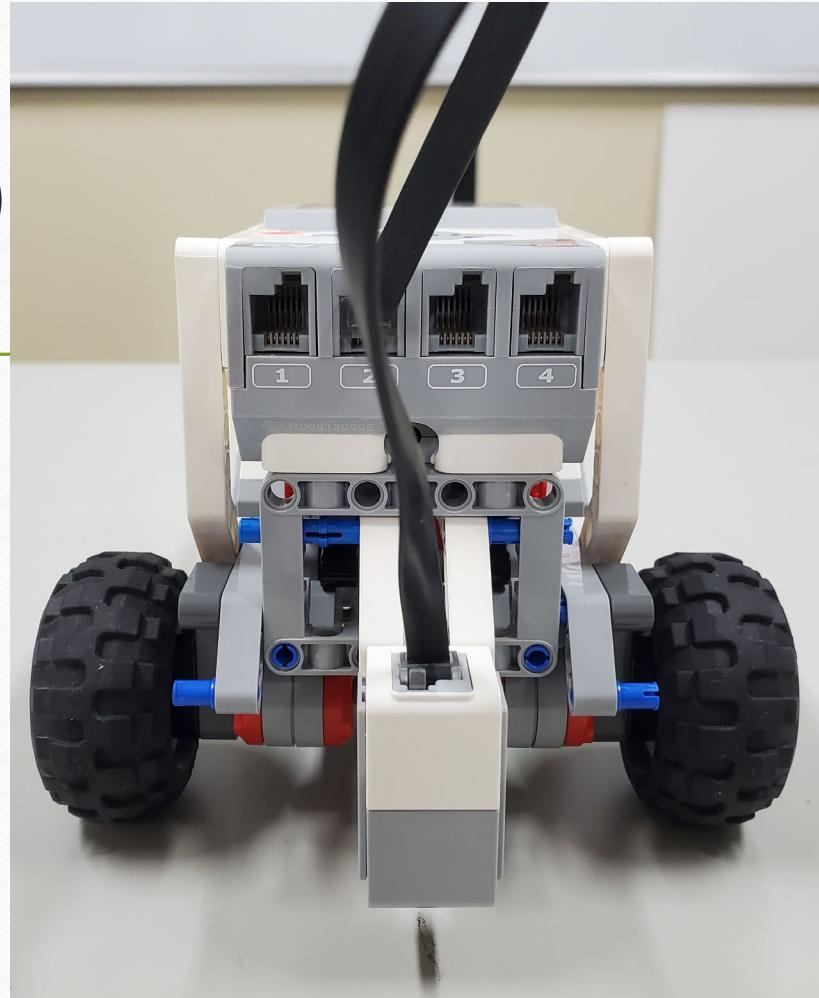


右視圖

成品(2/3)

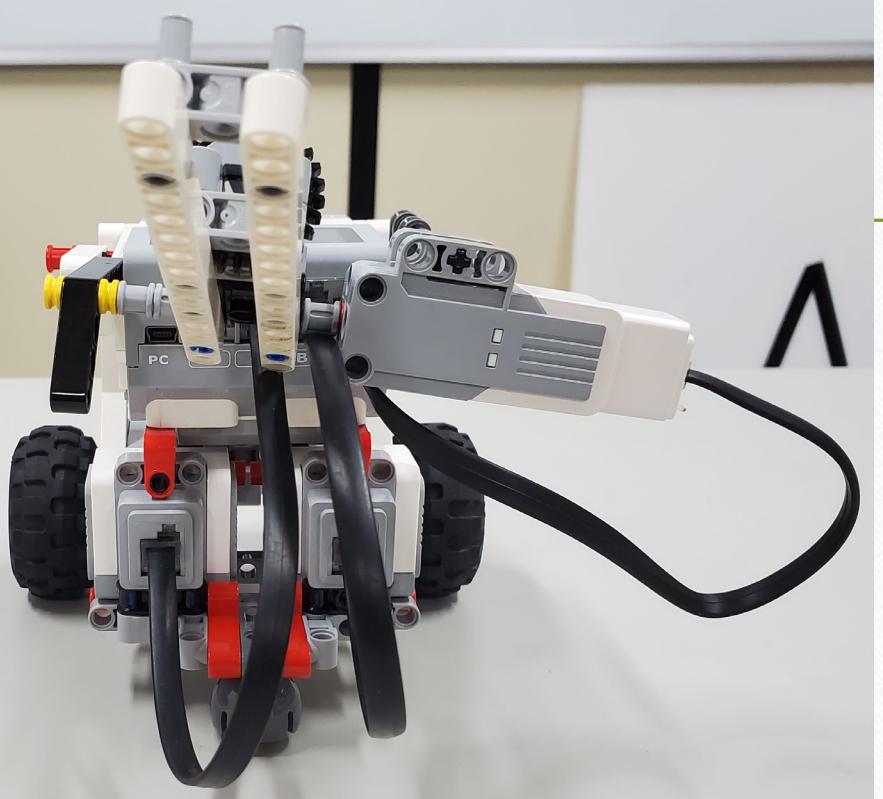


前視圖－含投筆機構

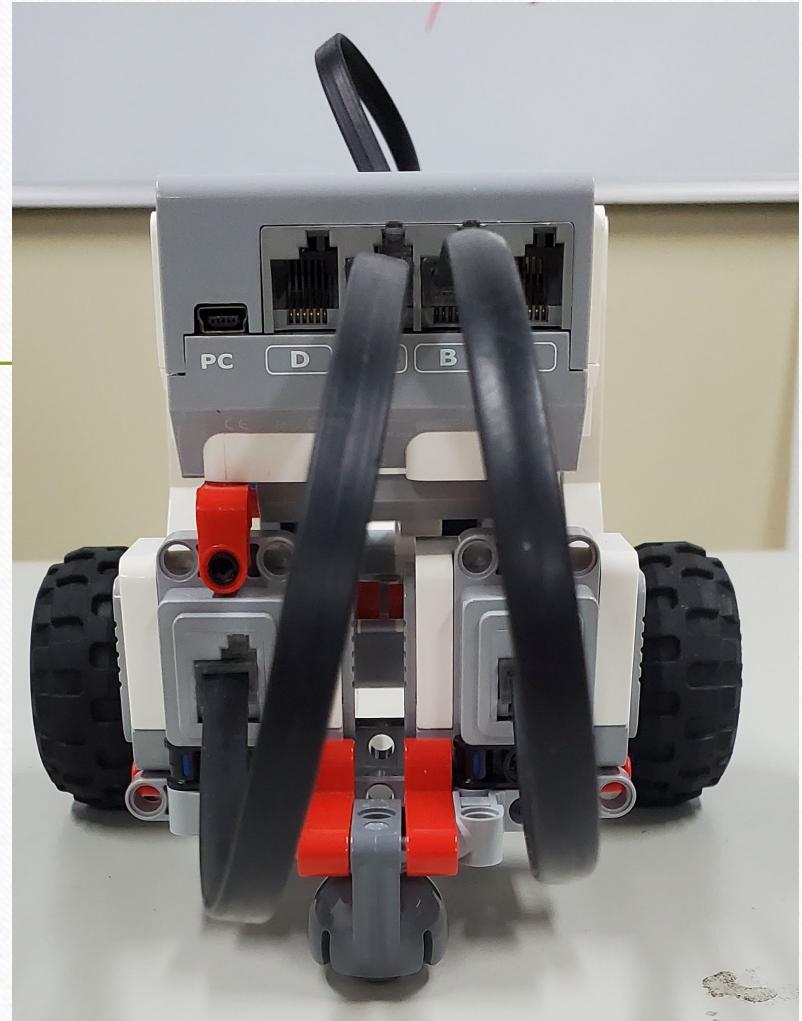


前視圖－不含投筆機構

成品(3/3)

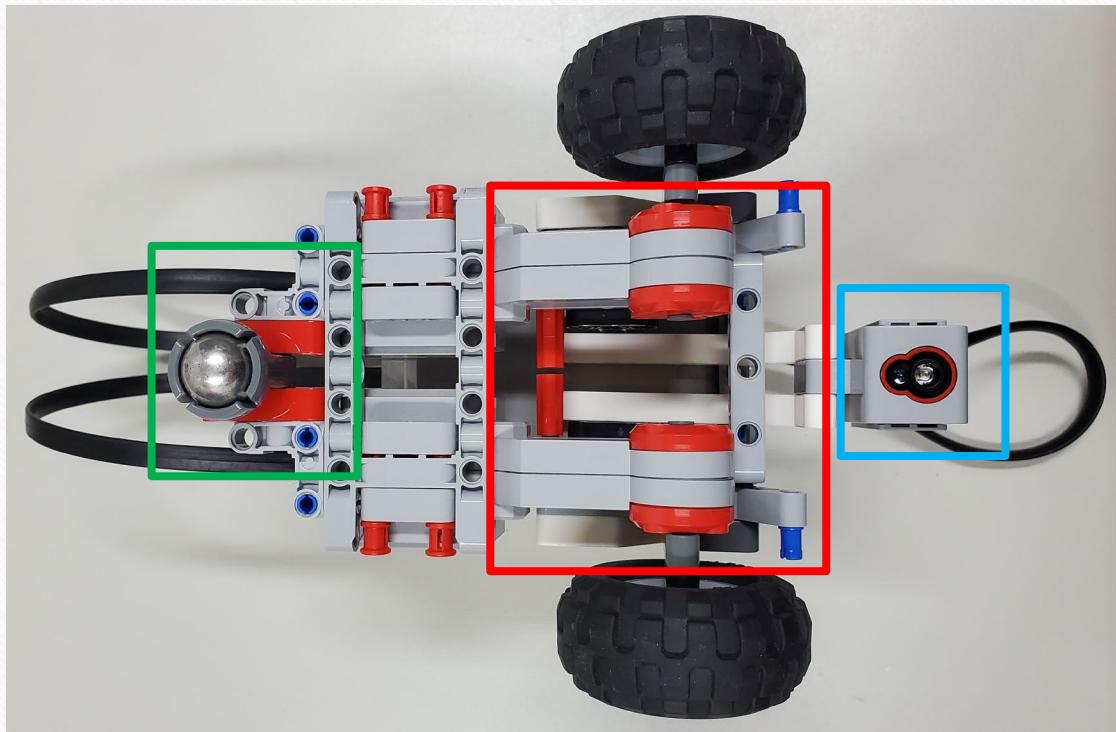


後視圖－含投筆機構



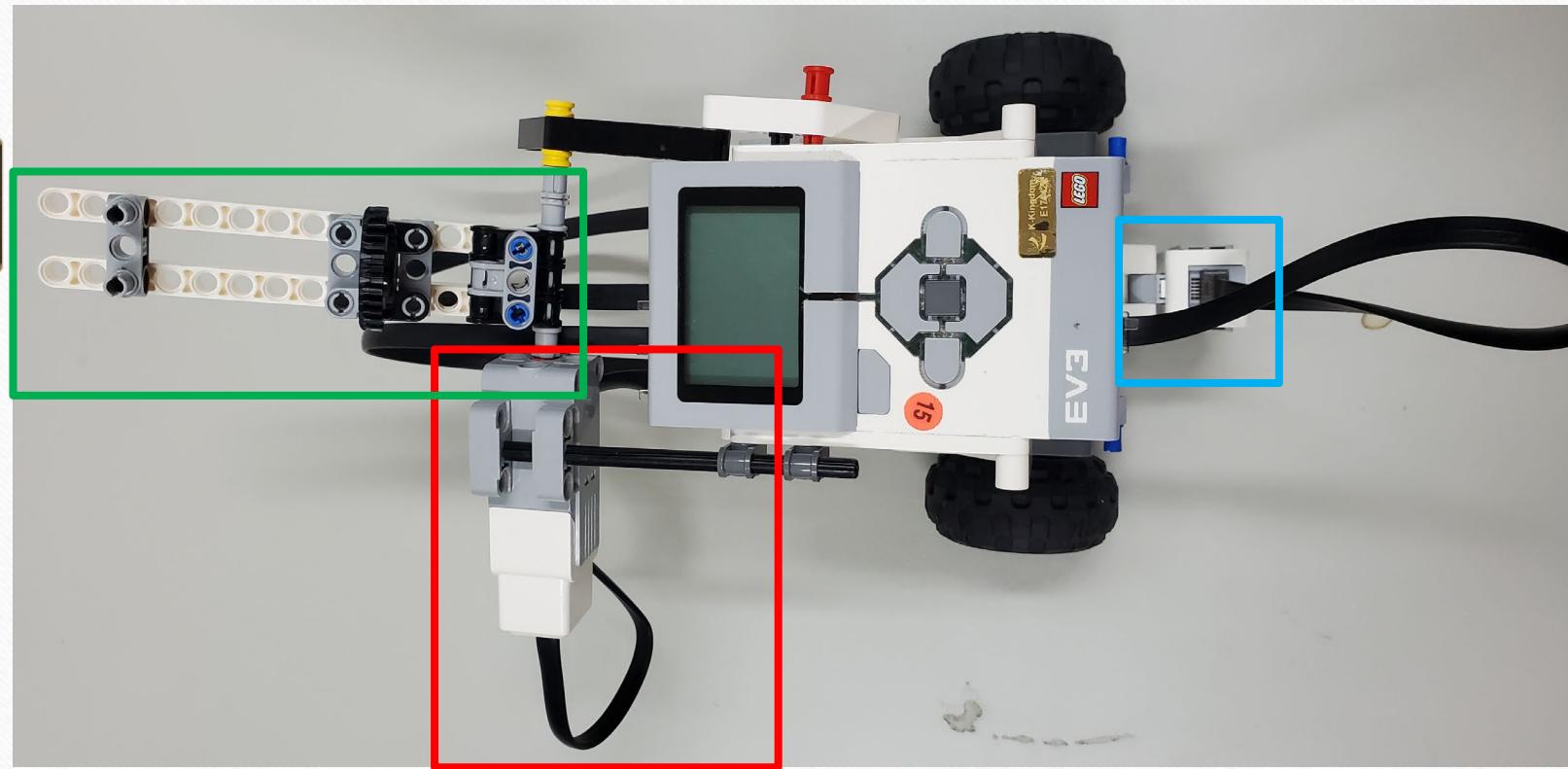
後視圖－不含投筆機構

下視圖



- 從下視圖可知，我們使用了兩顆EV3伺服馬達（紅色框框部份）及一顆滾輪（綠色框框部份），使兩個伺服馬達當驅動，而滾輪當作第三個支撐點以及從動機構。如此的好處是，三個支撐點的車子比四個支撐點的車子更容易在小旋轉半徑的情況下進行轉向。
- 最後再加上一顆顏色感測器（藍色框框部份），以利尋跡。

上視圖

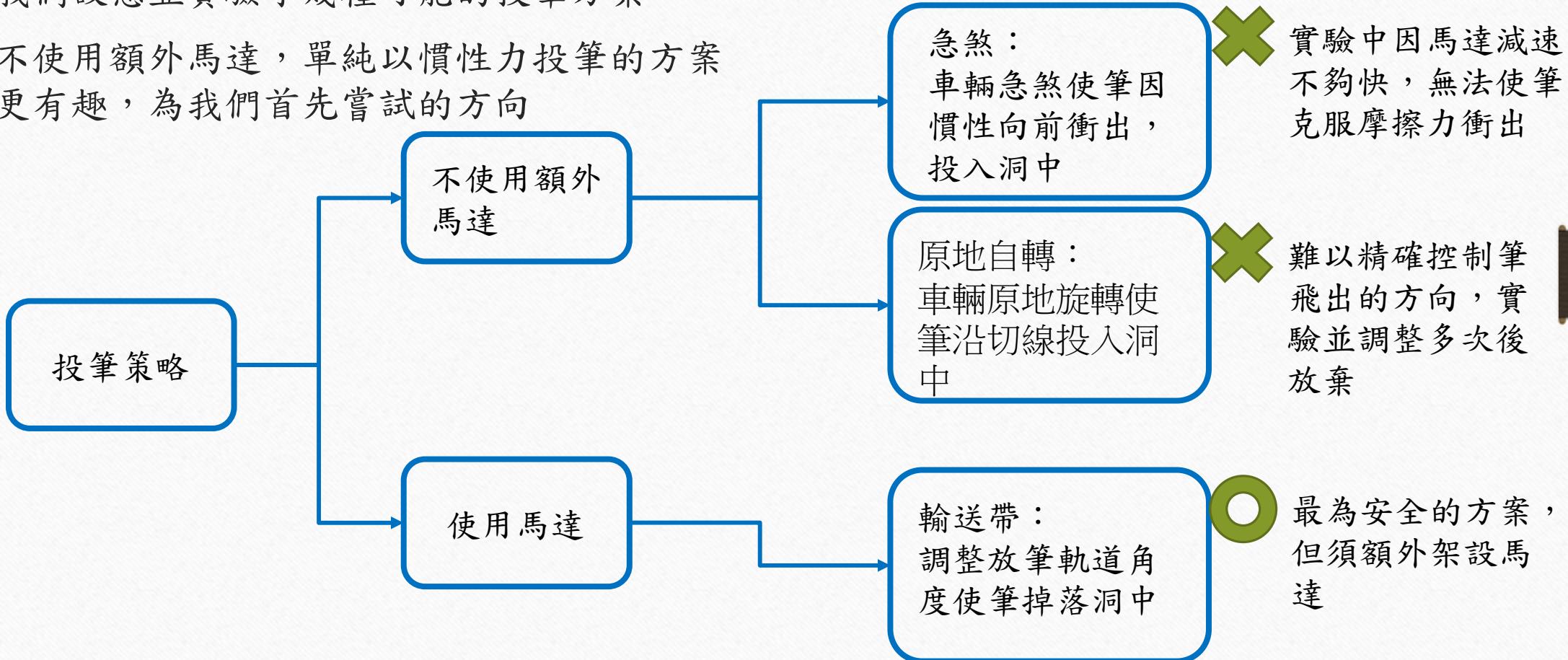


- 從上視圖可知，我們使用了一顆EV3伺服馬達（紅色框框部份）來控制投筆機構（綠色框框部份）。
- 而上一頁提到的顏色感測器則是藍色框框部份。

投筆機構

概念發想

- 我們設想了幾種可能的投筆方案：
- 不使用額外馬達，單純以慣性力投筆的方案更有趣，為我們首先嘗試的方向

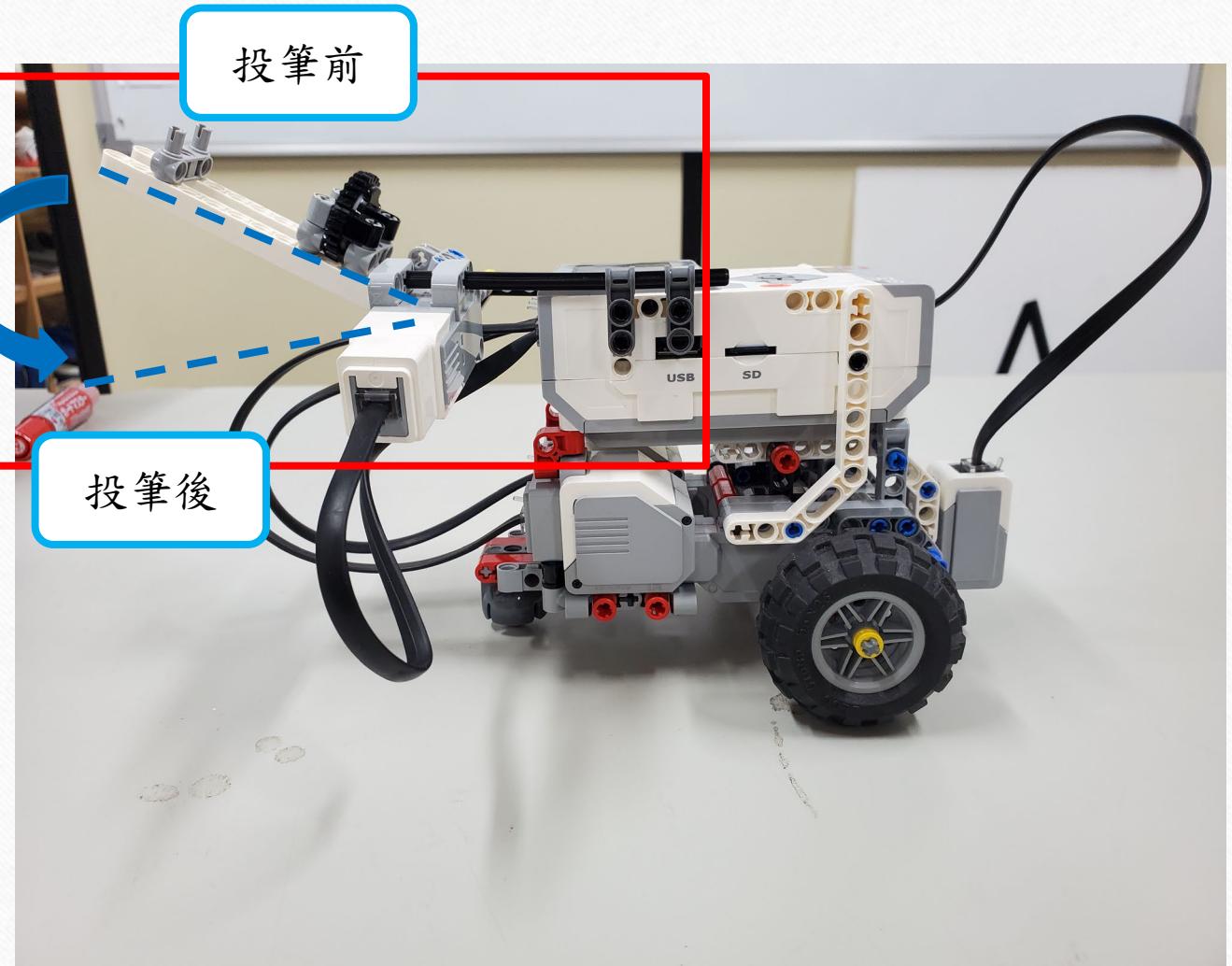


原理

藉由固定於側邊的馬達帶動軌道的仰角達到投放筆的功能。

車在循跡行走時軌道仰角 >0 使筆固定於軌道上。投筆時調整軌道仰角 <0 使筆落入洞中。

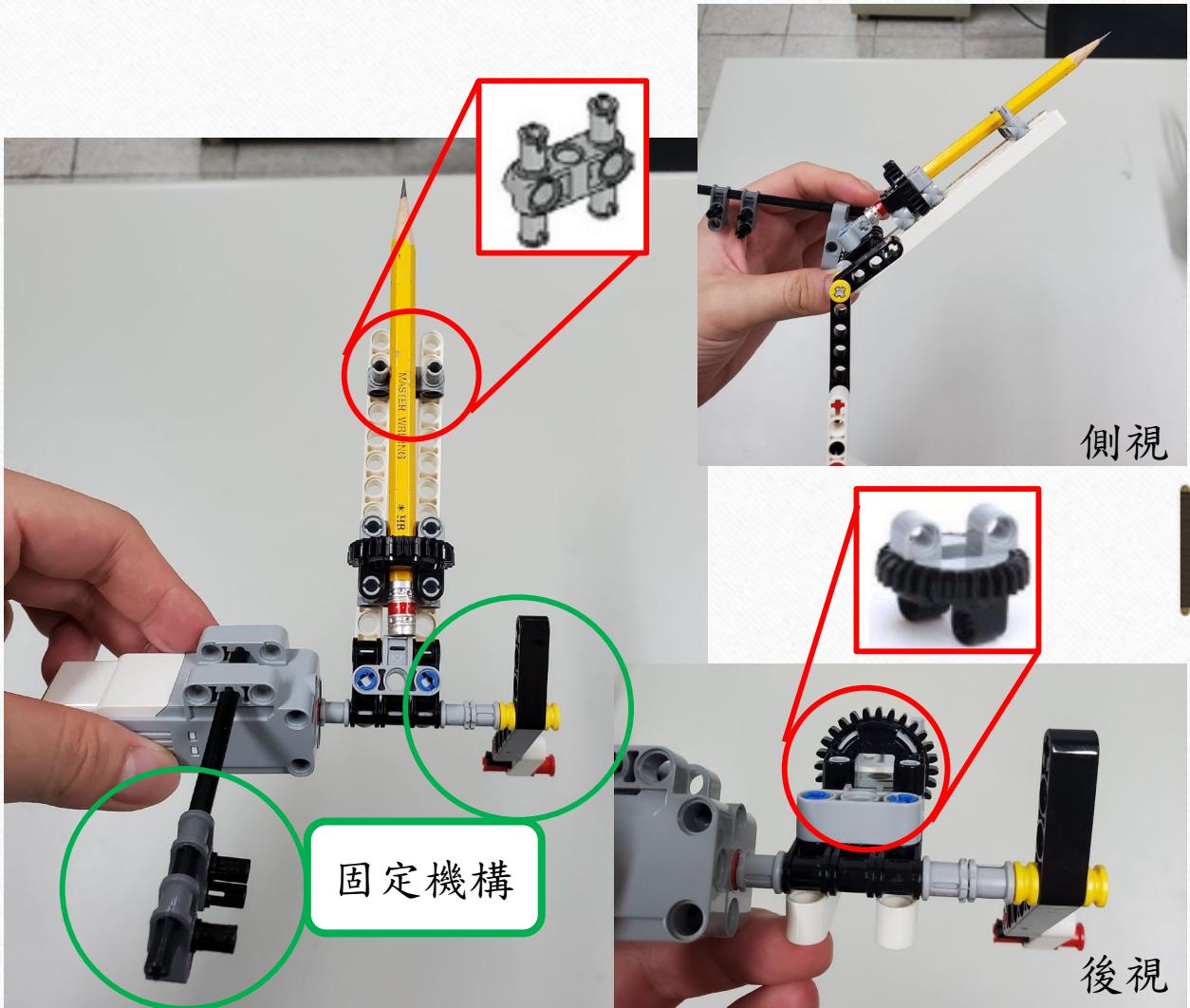
我們將此機構架設於車尾，因此在循跡時不會在第二個彎道和前方洞口的塑膠板碰撞，在紅線時車轉180度並啟動馬達投筆



機構

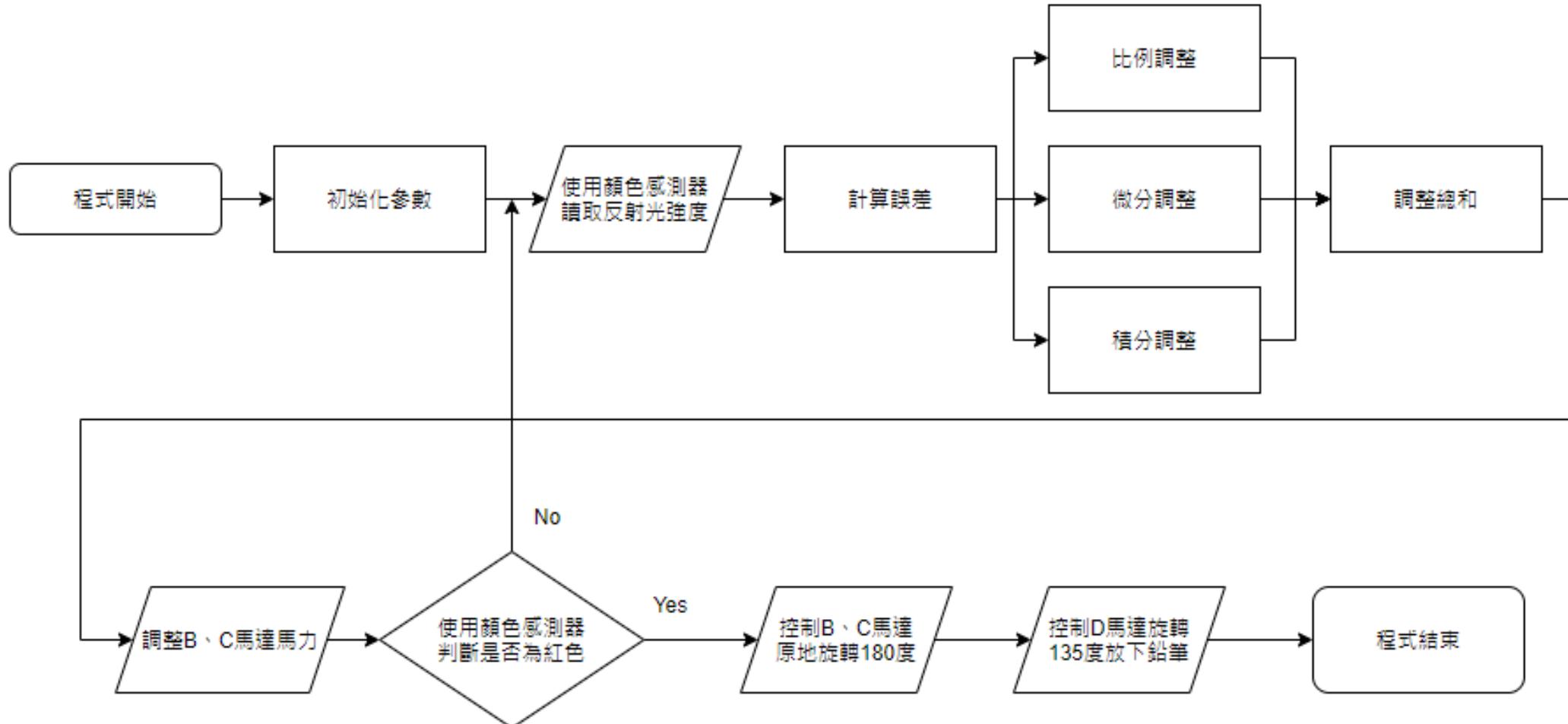
運用Lego Ev3零件中兩根9*1零件做為投筆軌道主幹，之間架設齒輪傳動件做為約束筆上下左右運動的主要部件，加上前方H型零件支撐筆身。筆後方用T型零件作為防止筆向後滑動的部件。

使用十字長桿連結馬達和投筆軌道，並作為馬達的傳動裝置。馬達本身以另一個十字長桿支撐，右方也以結構支撐，左右邊皆固定於EV3主機上。



程式解說

程式流程圖



程式參數設定

```
# Hyperparameter
black = 6                                # 黑色反射光強度
white = 30                                 # 白色反射光強度
threshold = (black + white) / 2             # 取兩者中間值當閾值
proportion = 8                             # 比例常數
integral = 0.001                            # 積分常數
derivative = 25                            # 微分常數
dt = 10                                   # 時間間隔
motor_speed = 50                           # 馬達馬力
err = 0                                    # 記錄當下誤差
previous_err = 0                          # 記錄前次誤差
sum_err = 0                                # 記錄誤差總和
```

P型控制器

```
class P_controller:  
    def __init__(self, P):  
        self.P = P  
  
    def control(self, error):  
        return self.P * error
```

PI控制器

```
class PI_controller:  
    def __init__(self, P, I, timestep):  
        self.P = P  
        self.I = I  
        self.dt = timestep  
  
    def control(self, error, sum_error):  
        integral = (sum_error + error) * self.dt  
        return self.P * error + self.I * integral
```

PD控制器

```
class PD_controller:  
    def __init__(self, P, D, timestep):  
        self.P = P  
        self.D = D  
        self.dt = timestep  
  
    def control(self, error, previous_error):  
        derivative = (error - previous_error) / self.dt  
        return self.P * error + self.D * derivative
```

PID控制器 (主要)

```
class PID_controller:  
    def __init__(self, P, I, D, timestep):  
        self.P = P  
        self.I = I  
        self.D = D  
        self.dt = timestep  
  
    def control(self, error, sum_error, previous_error):  
        integral = (sum_error + error) * self.dt  
        derivative = (error - previous_error) / self.dt  
        return self.P * error + self.I * integral + self.D * derivative
```

程式實現

```
ev3.speaker.beep()                                # 程式開始提示音

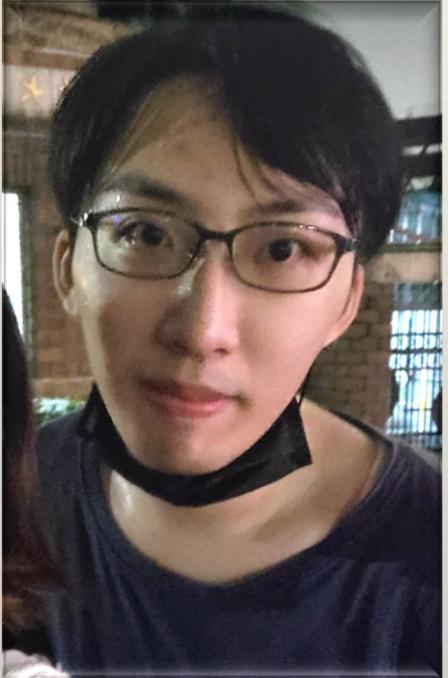
while True:
    if cSensor_reflect.color() == Color.RED:        # 顏色感測器讀到紅色則停止馬力調整
        break

    err = threshold - cSensor_reflect.reflection()  # 計算誤差
    # gain = ctrl.control(err)                      # 選擇P控制器
    # gain = ctrl.control(err, previous_err)         # 選擇PD控制器
    # gain = ctrl.control(err, sum_err)              # 選擇PI控制器
    gain = ctrl.control(err, sum_err, previous_err)   # 選擇PID控制器 (主要)
    previous_err = err                               # 記錄當下誤差
    sum_err += err                                  # 記錄累積誤差
    mobile_car.drive(motor_speed, gain)            # B、C馬達馬力調整
    wait(dt)                                       # 等待一小段時間

mobile_car.turn(180)                             # 原地旋轉180度
carrierMotor.run_target(100, 135)                 # D馬達旋轉135度放下鉛筆

ev3.speaker.beep(1000, 500)                      # 程式結束提示音
```

組員心得



陳品存 (R09522849) 心得：

雖然以前在大學時期便有修過同樣是組裝樂高的課程，但那堂課較為基本，因此對於硬體的使用沒有相對這堂課來說更加熟悉。

以前我一直以為機構設計是十分簡單的，原本我們打算僅使用兩顆馬達便達到將筆投入洞口的想法，但後來幾經失敗後決定再多使用一顆馬達，也就是增加一個自由度，不過我依舊認為，如果樂高是精度更高的馬達，或許真的能僅僅用兩顆馬達，再搭配轉向時的加速度將筆「甩」進洞口也說不定。

吳政彥 (R10522801) 心得：

藉由這次的實驗一，了解如何在EV3使用Micro Python的語法，並且實現多種控制器來完成循跡任務，雖然最後使用PID controller，不過多次實驗下來其實與比例控制表現差不多。實驗過程中有嘗試多種方法來完成將筆投入洞口中，但想省下一個馬達來完成非常困難，因此我們還是使用正常的方式來完成。





陳政豪 (R10522814) 心得：

之前大學時是雖然也是用零件拼湊成機器人進而搭配Arduino進行操作，但是那個都是固定好的零件，不像這門課是有很多零件可以使用，而且可以依照各自的創意創造出每組獨一無二的樂高車，在設計結構的同時也要同時思考軟體要怎麼寫，因為硬體以及軟體的設計是相輔相成的，有些時候是軟體能做到硬體不能，因此我認為這個樂高模型可以顯現出各組的創意發想，非常有趣！

戴承寧 (R10522834) 心得：

第一次實驗，最主要的是熟悉EV3的操作方法以及實際驗證包括馬達、程式控制零件所能做出的功能效果為何。過程中最有趣的一天晚上花不少時間嘗試各種神奇的方法讓筆不靠額外的主動動力而投進洞，當時不斷的修改機構與程式調整並再次實驗，雖最後仍回到較保險的方法，也是收穫很多的過程。



小組分工表

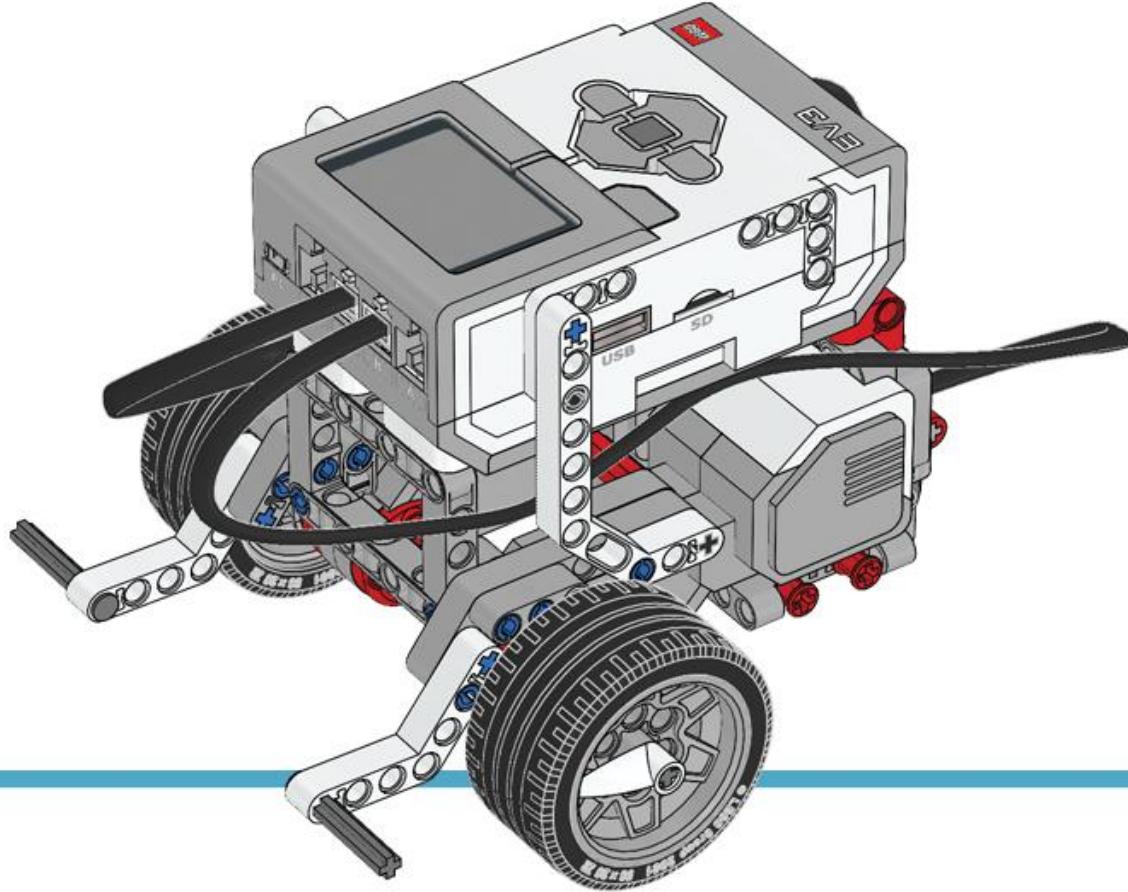
小組分工表

姓名	負責項目與內容
陳品存	機構設計與PPT統整
吳政彥	程式設計
陳政豪	程式設計
戴承寧	機構設計

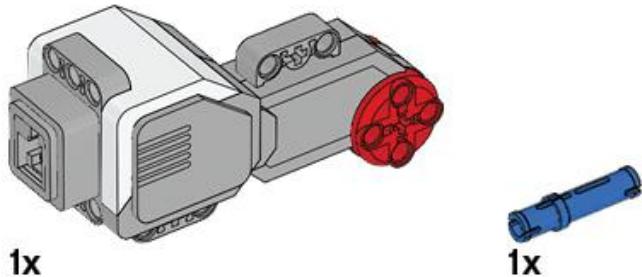
附錄一：車體設計圖

取自樂高LEGO官方網站：

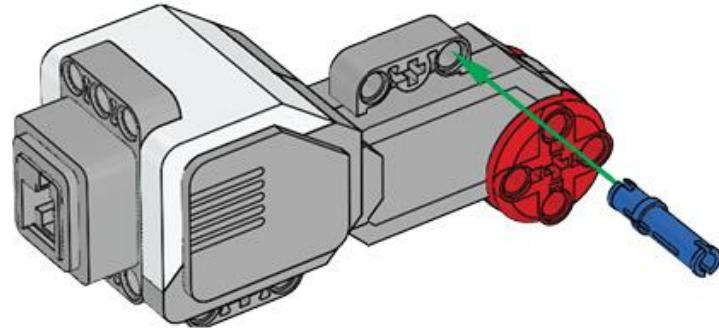
<http://robotsquare.com/wp-content/uploads/2013/10/45544_educator2.pdf?fbclid=IwAR3jZTeTXEsWJSgitKz14kpZ-RHtg56L_2806T0z7MU8S2lJC_3VepFiyI4>

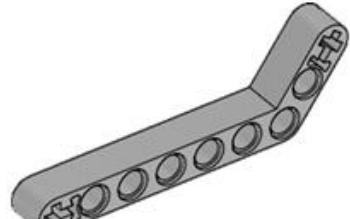


LEGO **MINDSTORMS**
education **EV3**



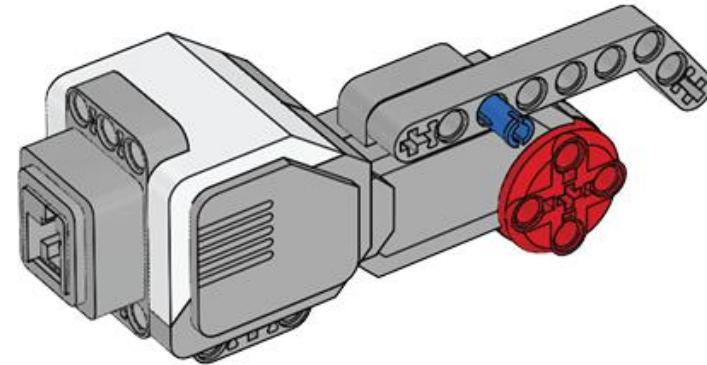
1

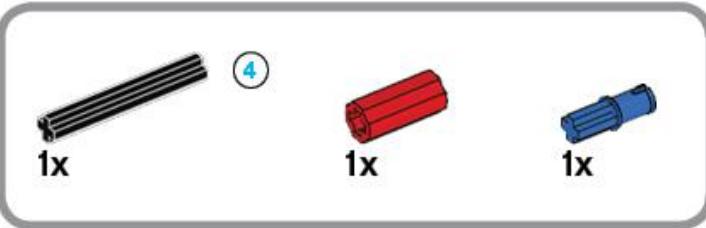




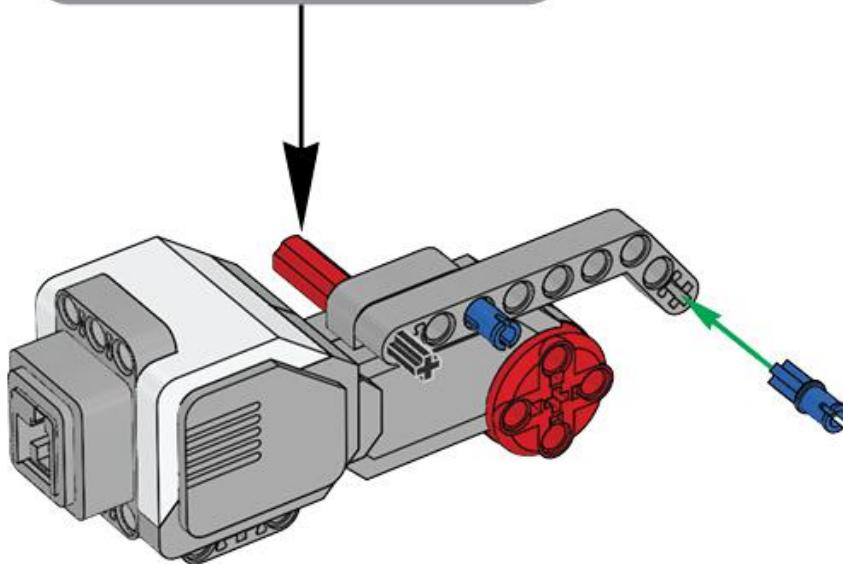
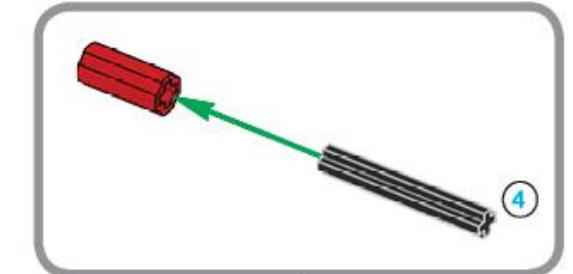
1x

2





3



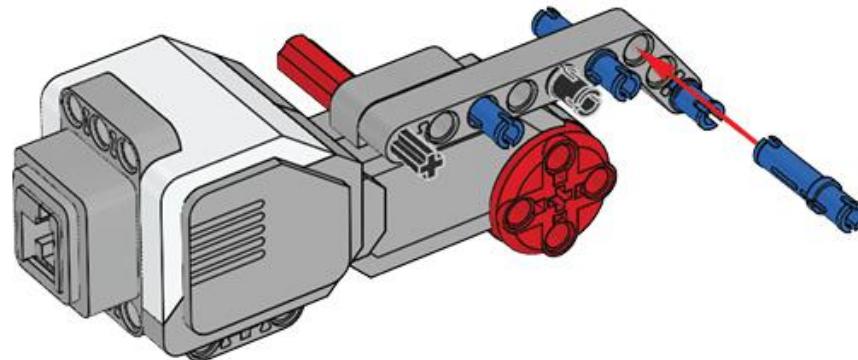


1x

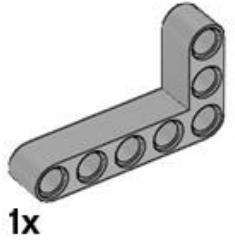


2x

4

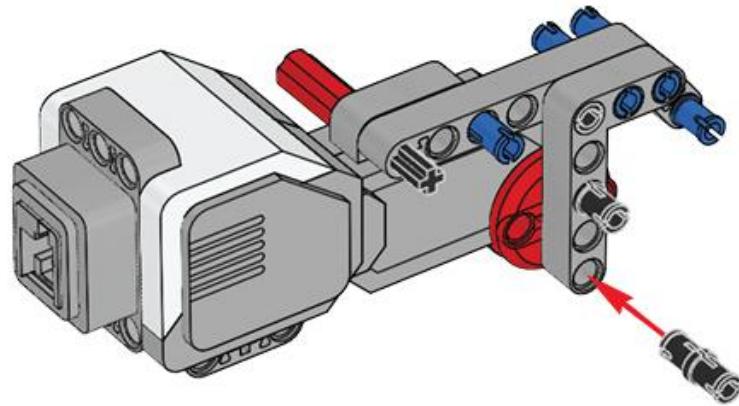


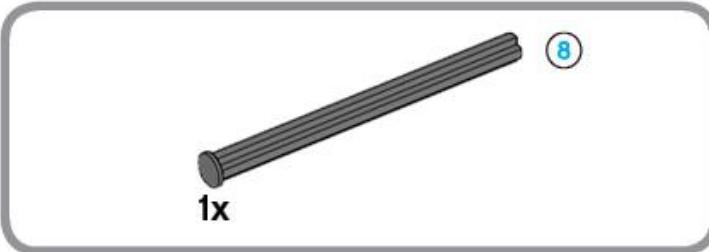
2x



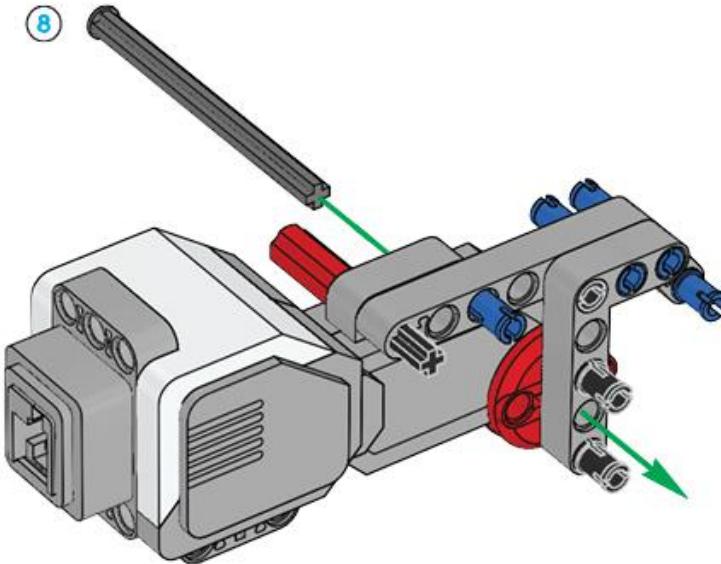
1x

5





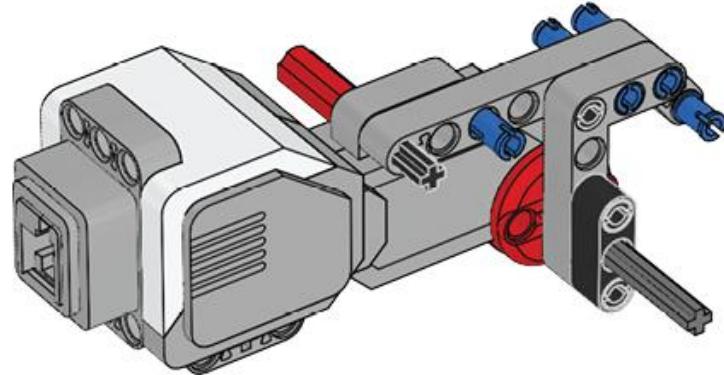
6

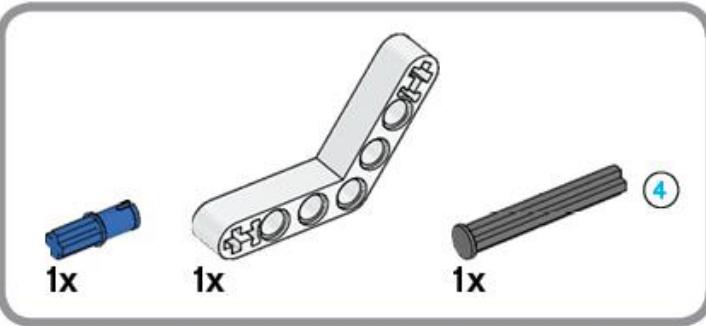




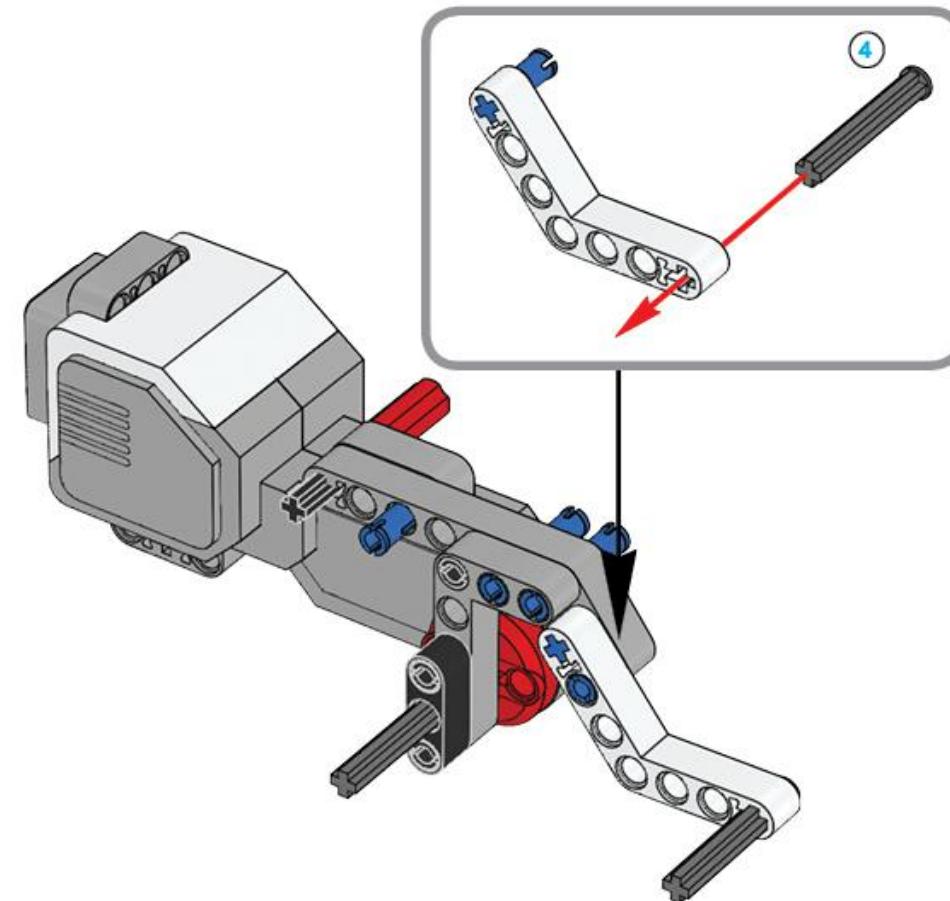
1x

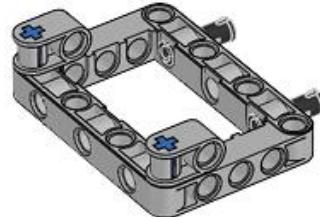
7





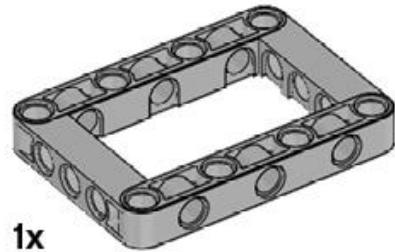
8



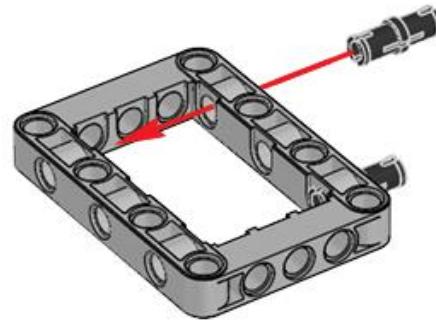


2x

1x



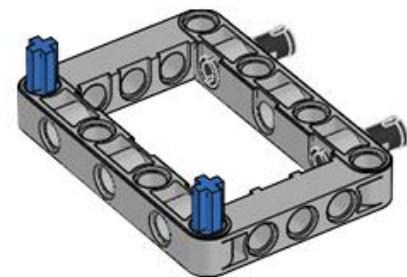
9





2x

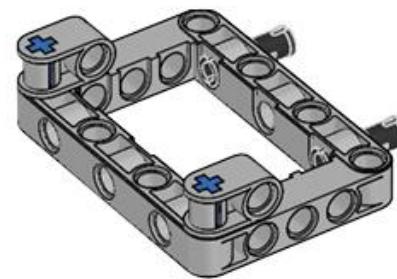
10



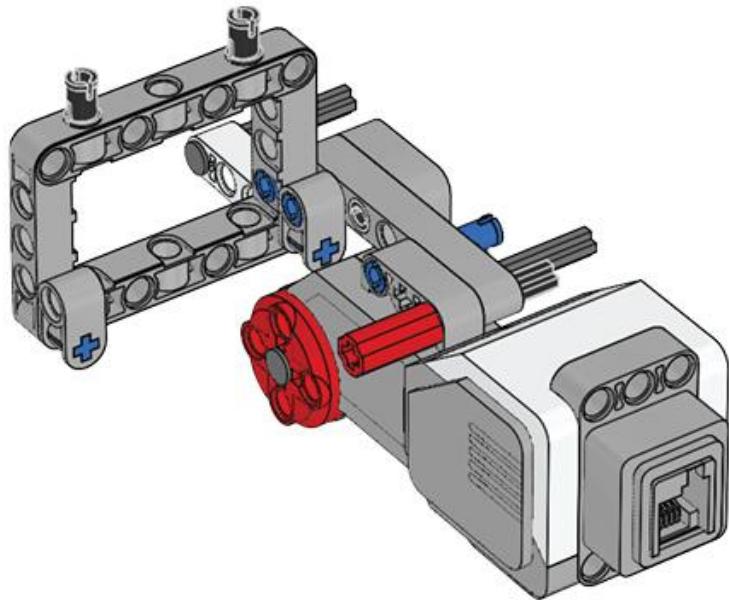


2x

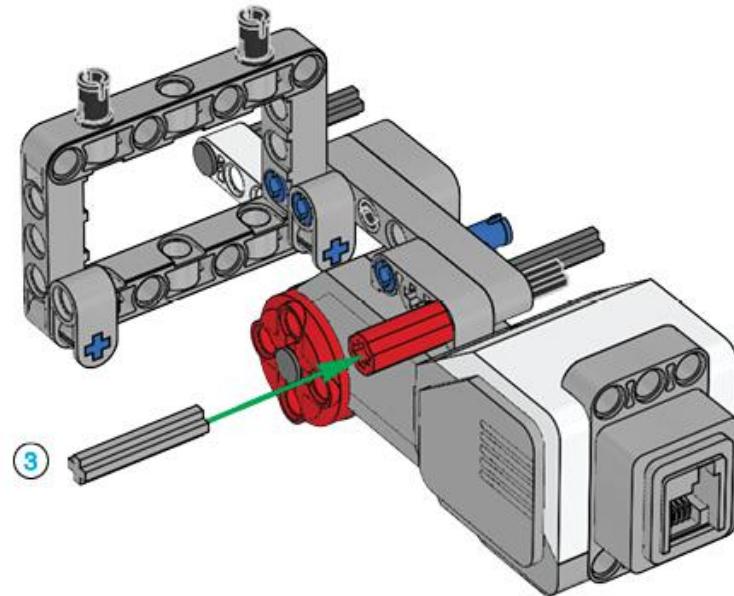
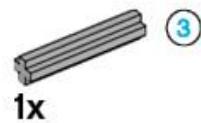
11

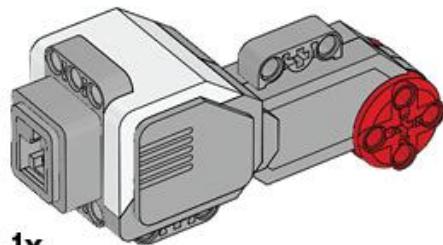
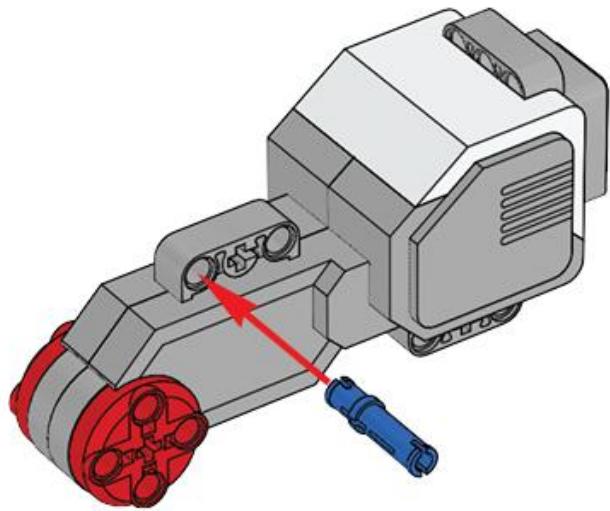
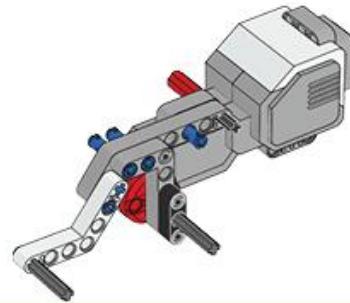


12



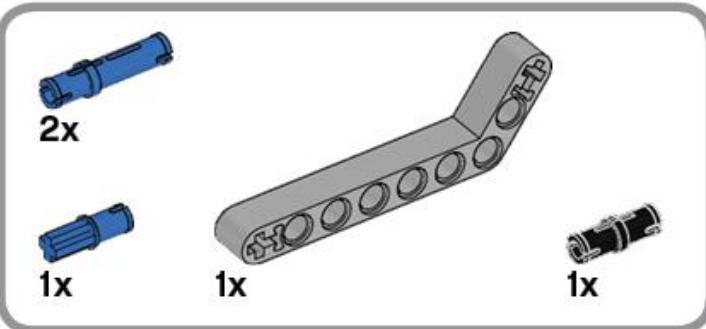
13



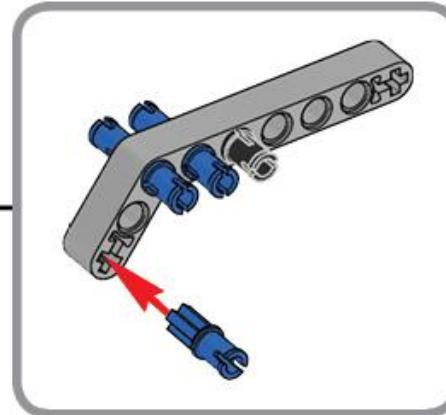
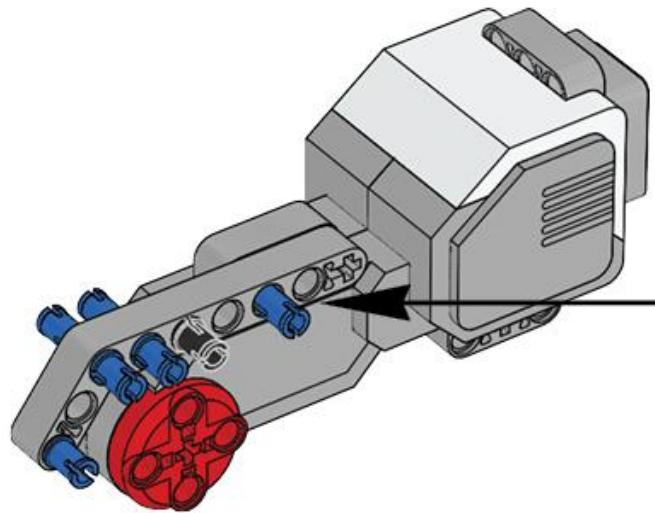


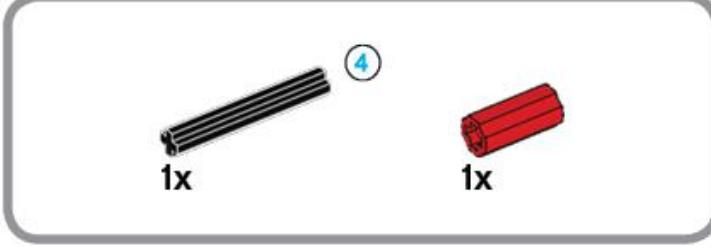
1x

14

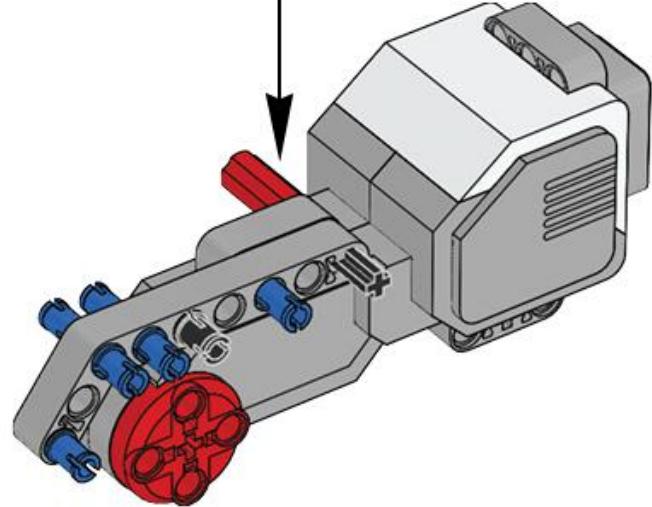
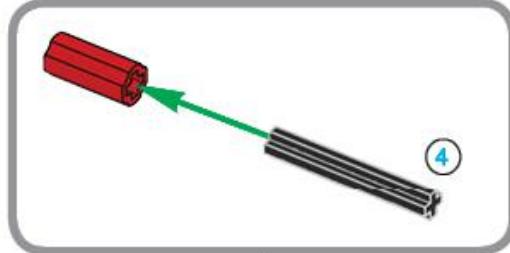


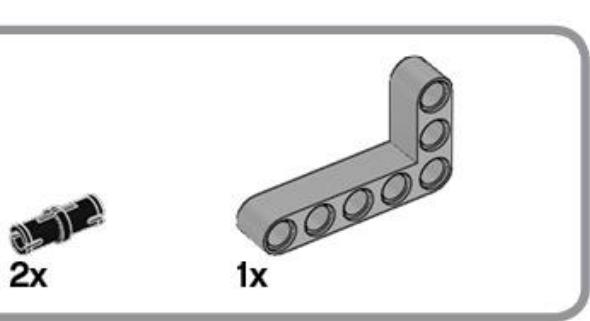
15



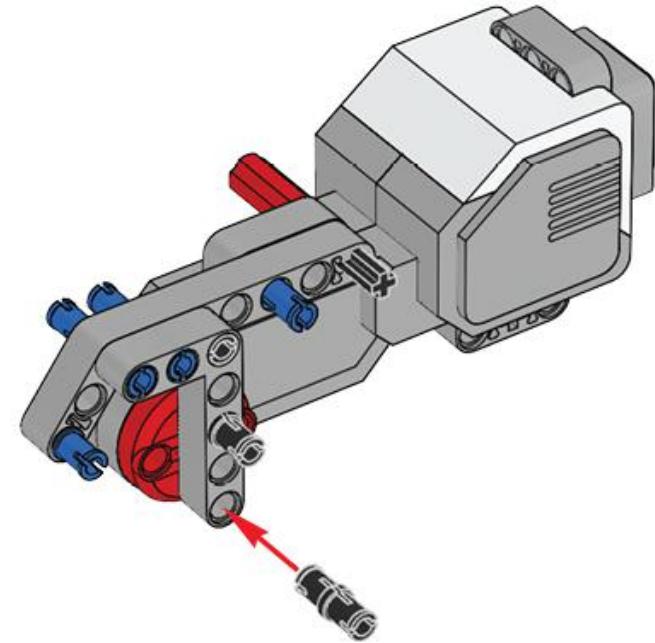


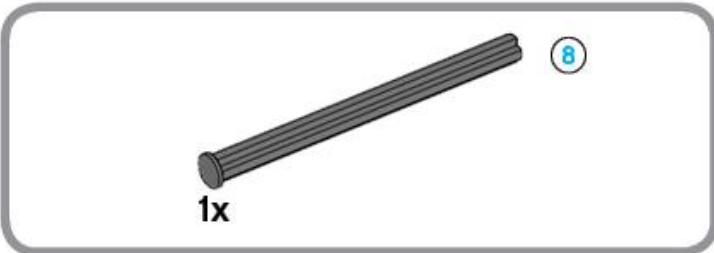
16



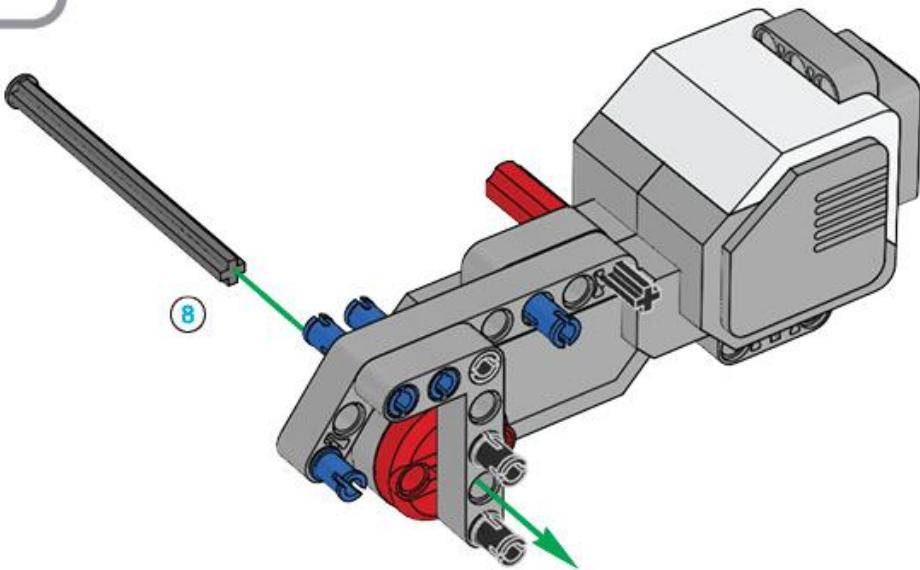


17





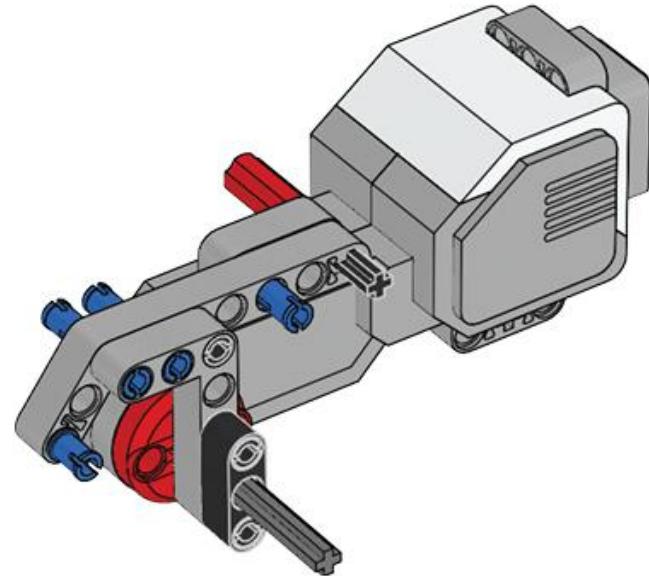
18

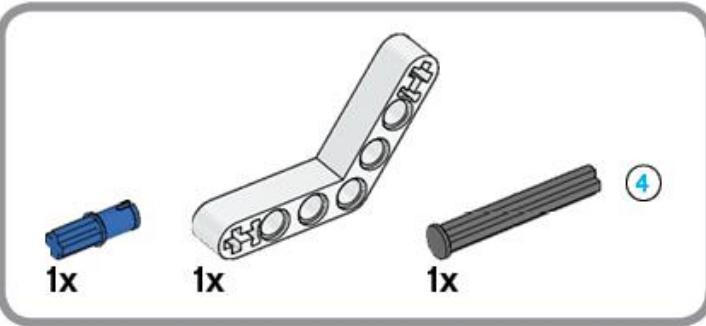




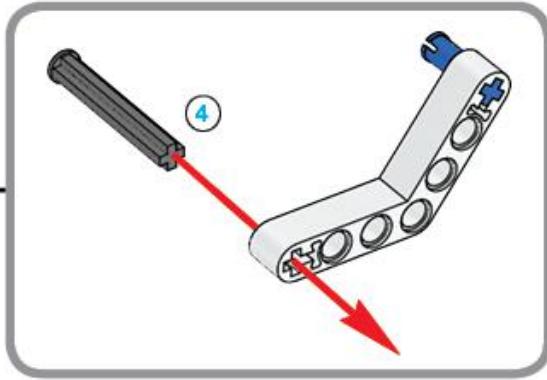
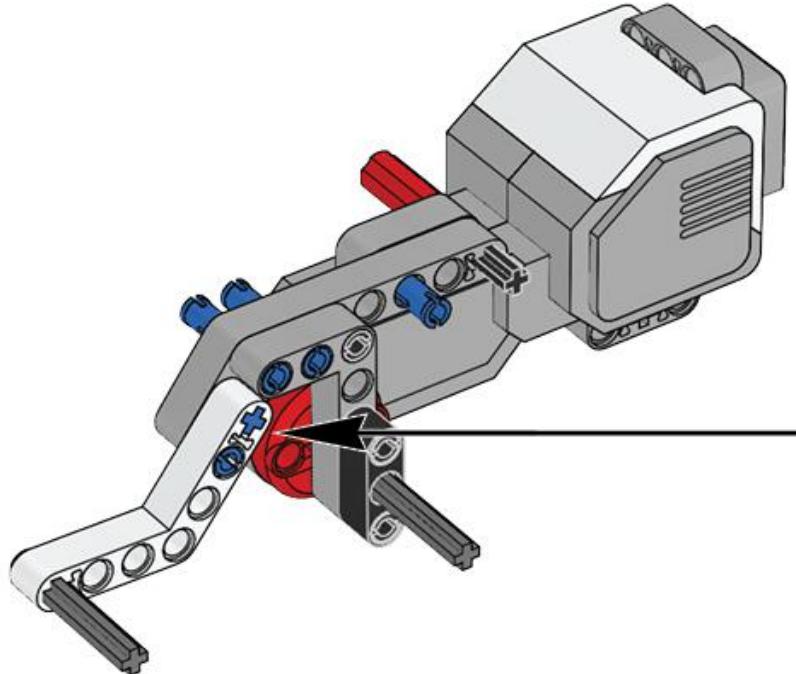
1x

19

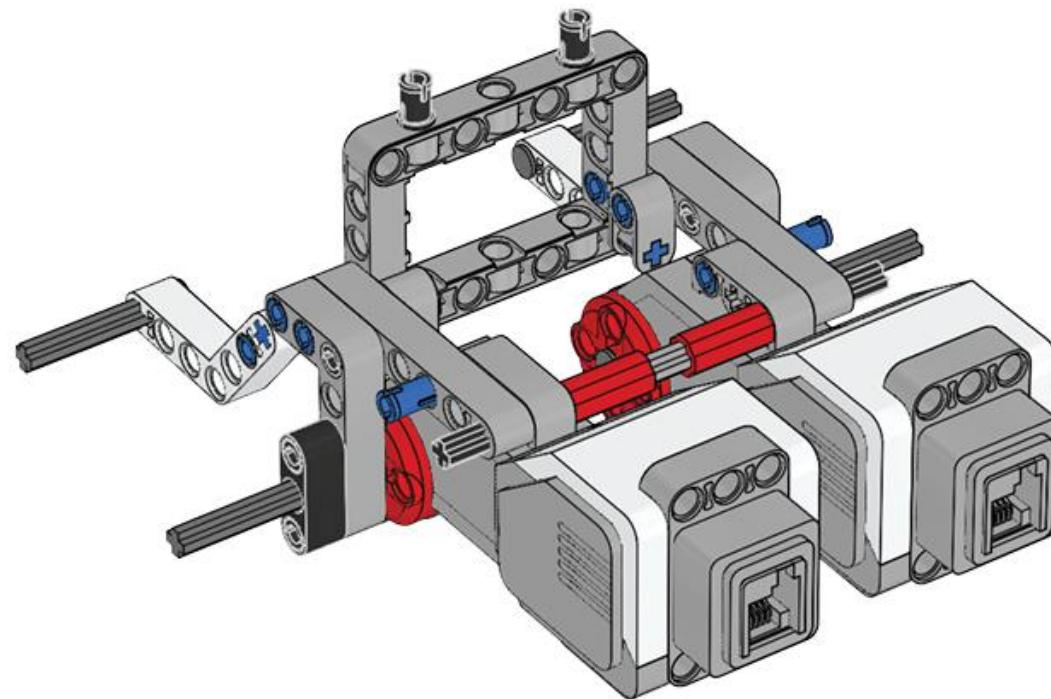


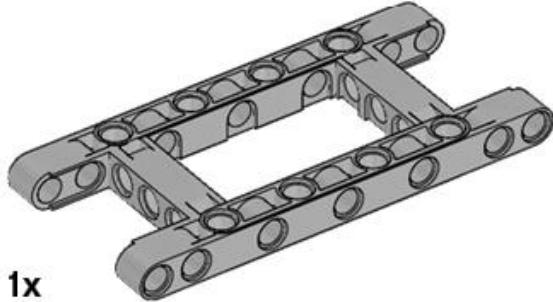


20



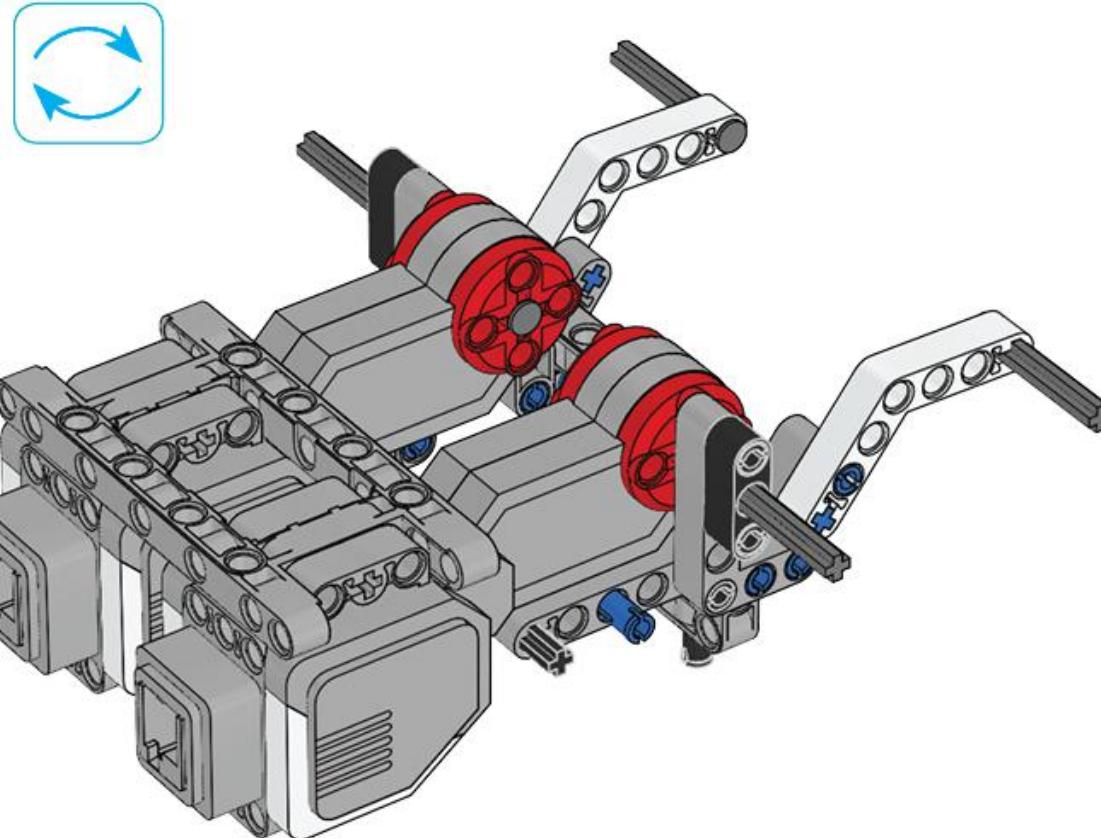
21



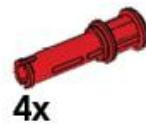


1x

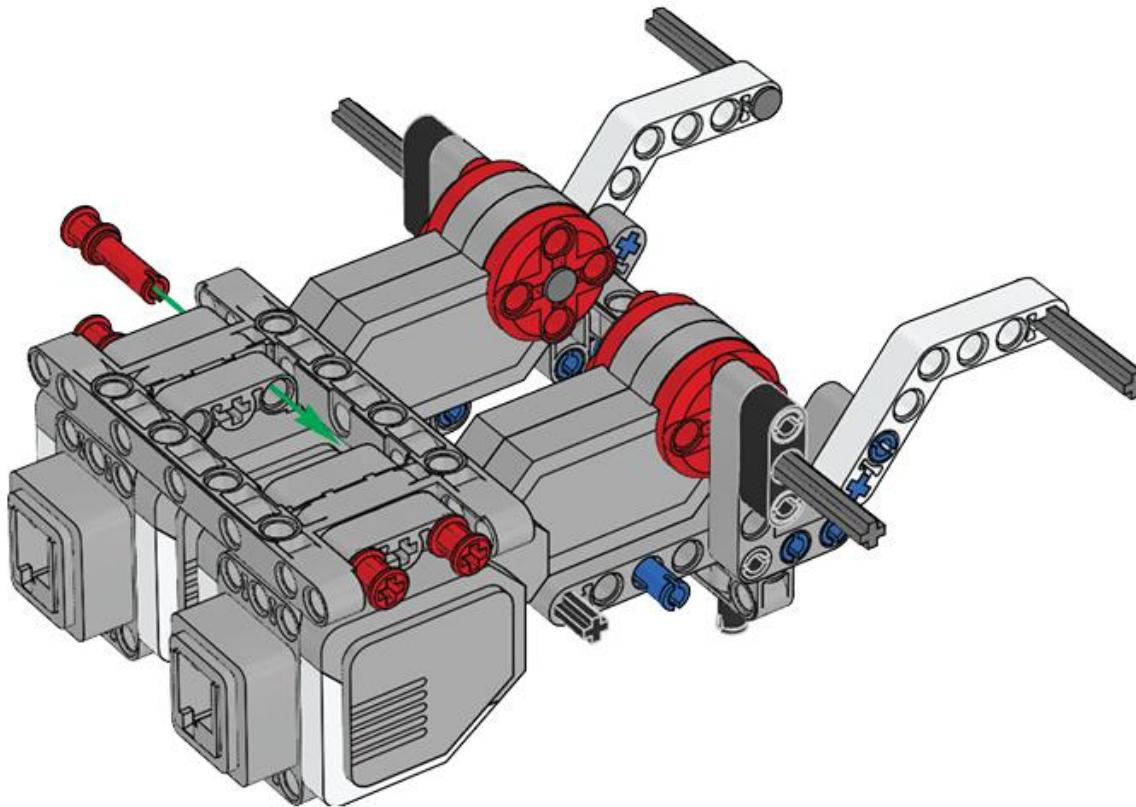
22

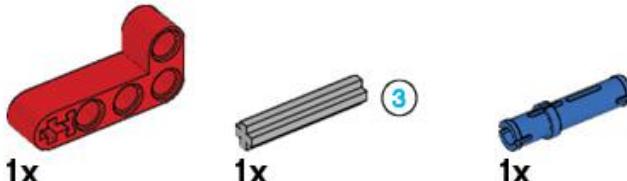
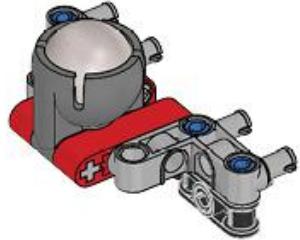


23

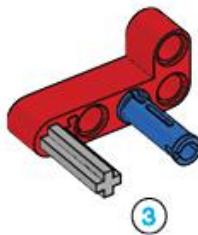


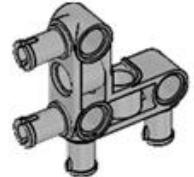
4x





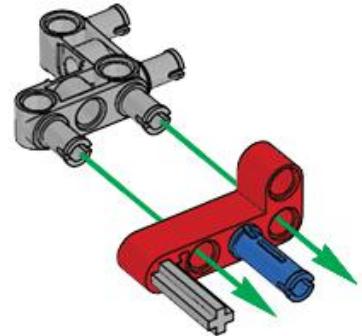
24





1x

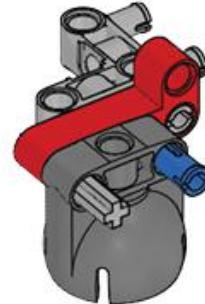
25





1x

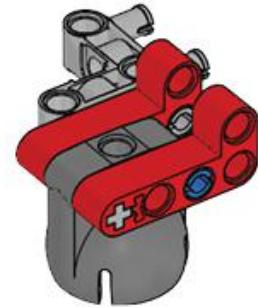
26

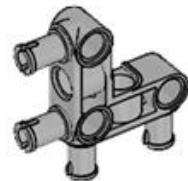




1x

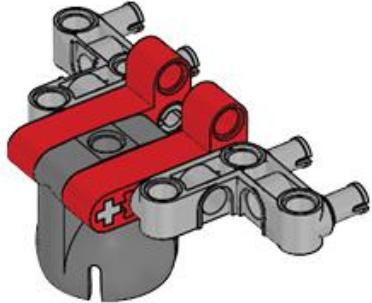
27





1x

28





2x

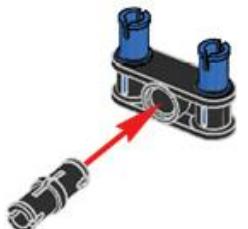


2x

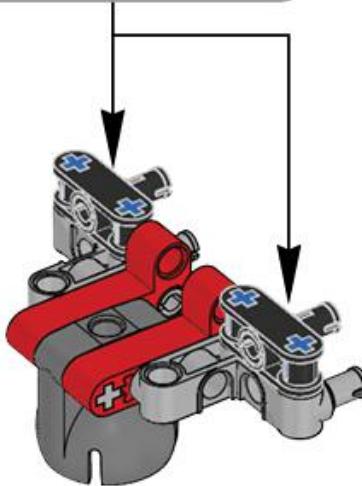


4x

29



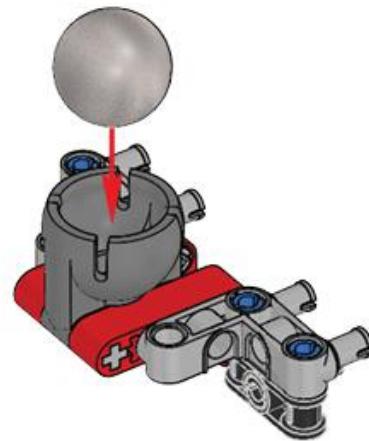
2x



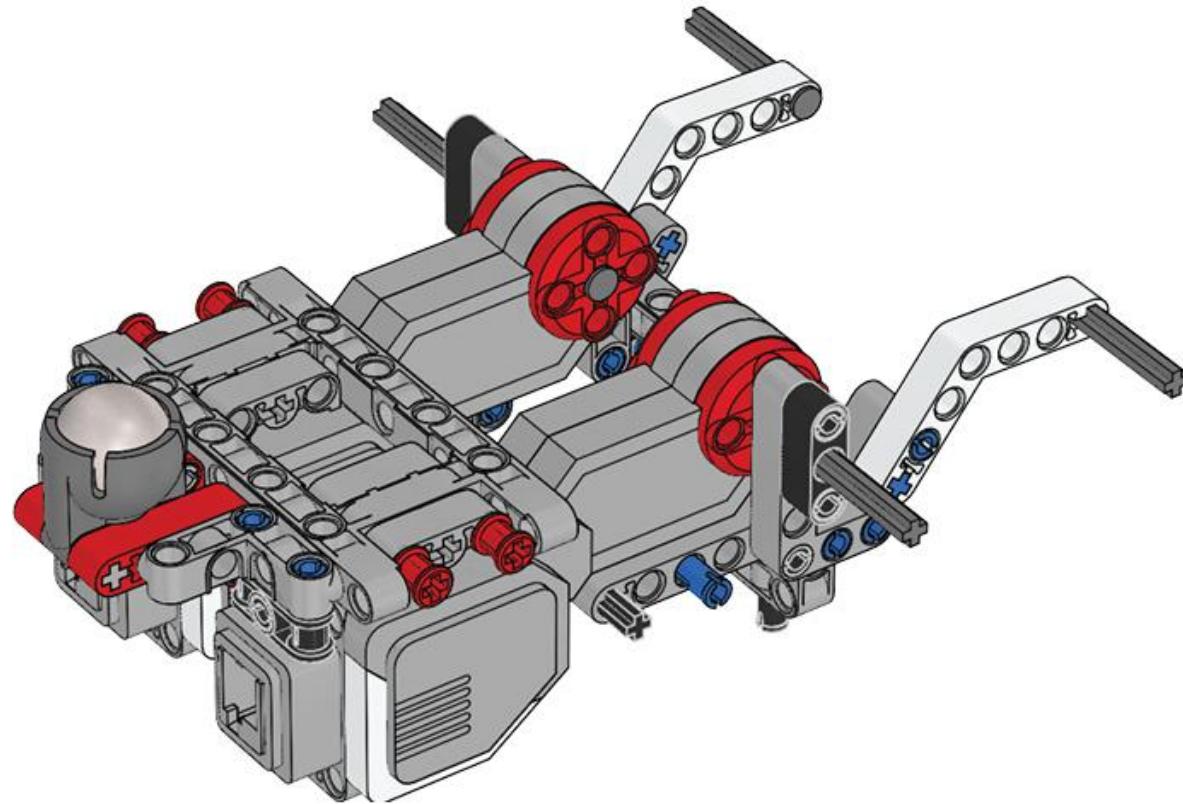


1x

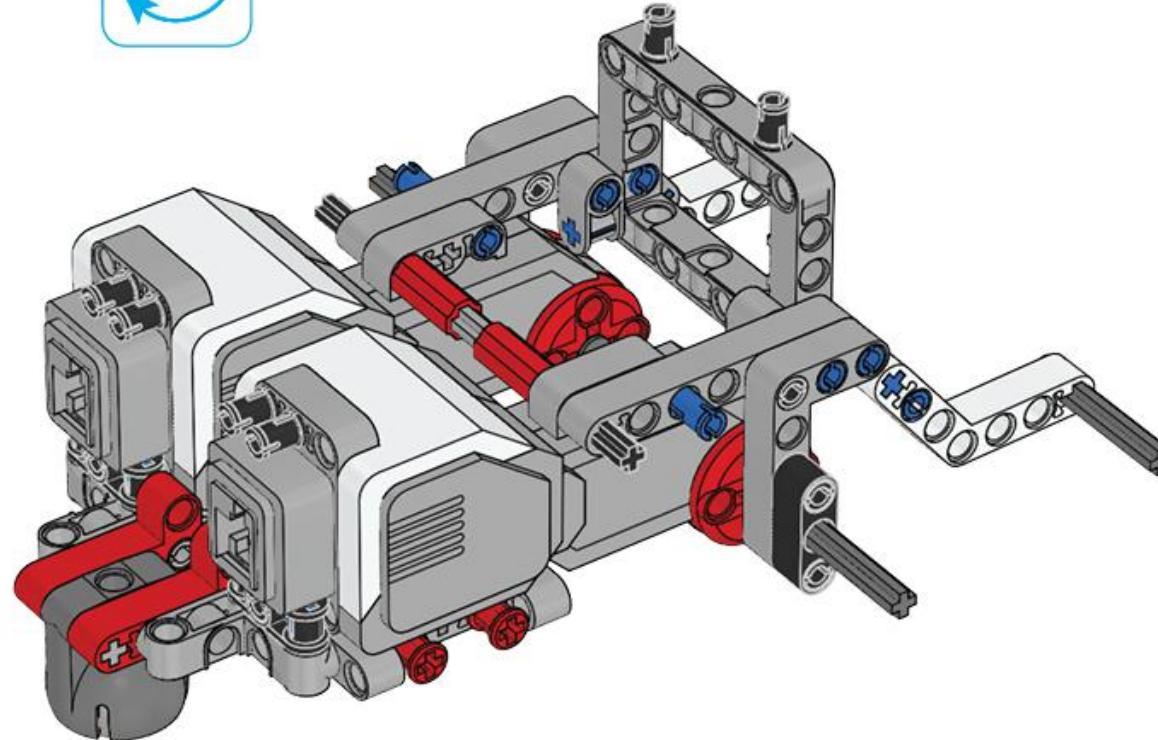
30

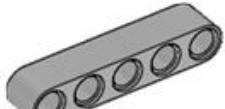


31



32





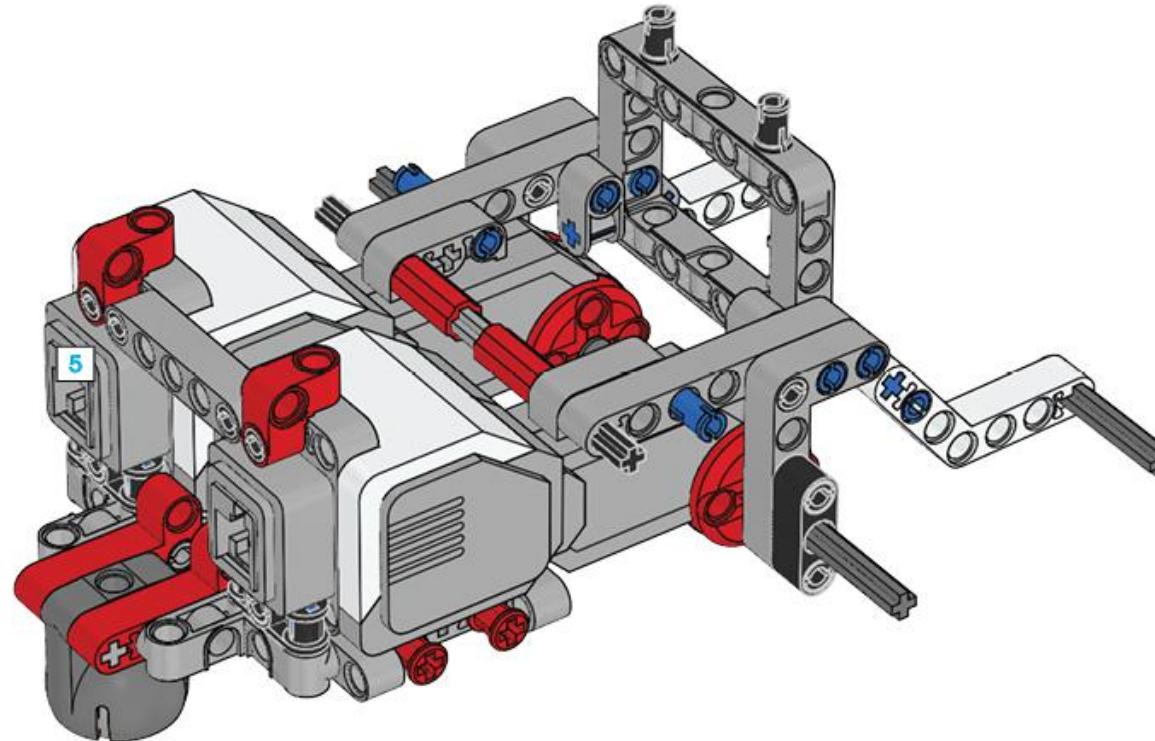
1x

5

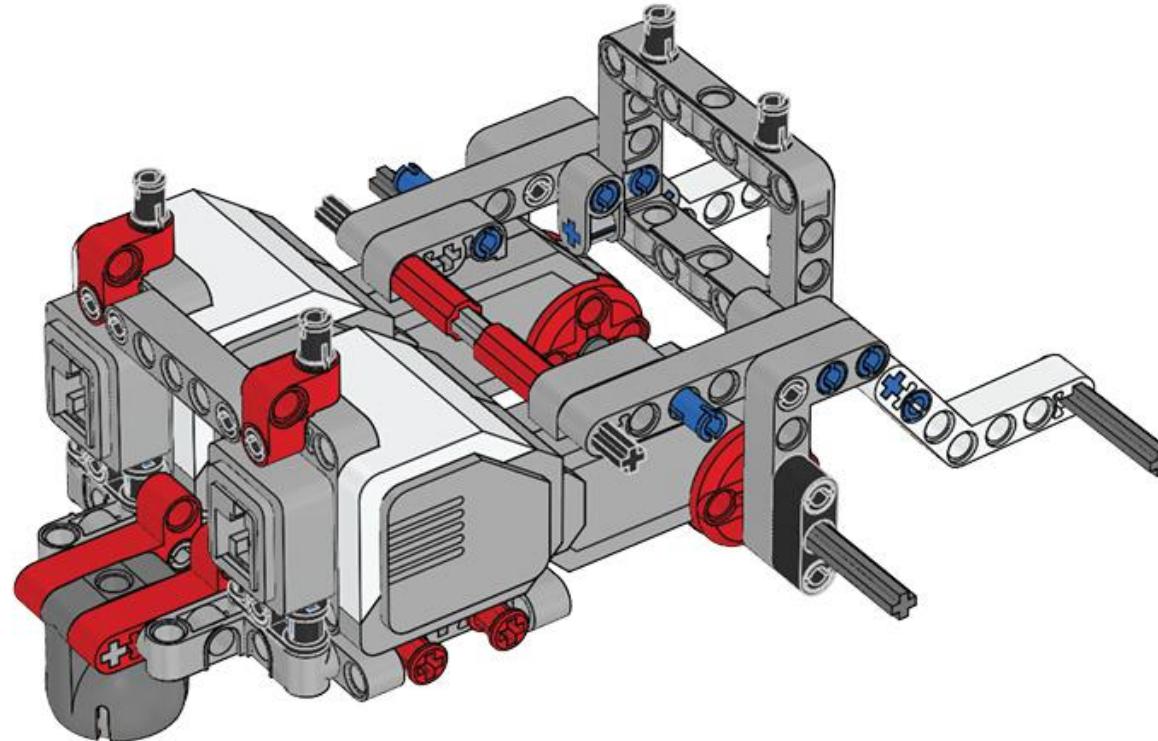


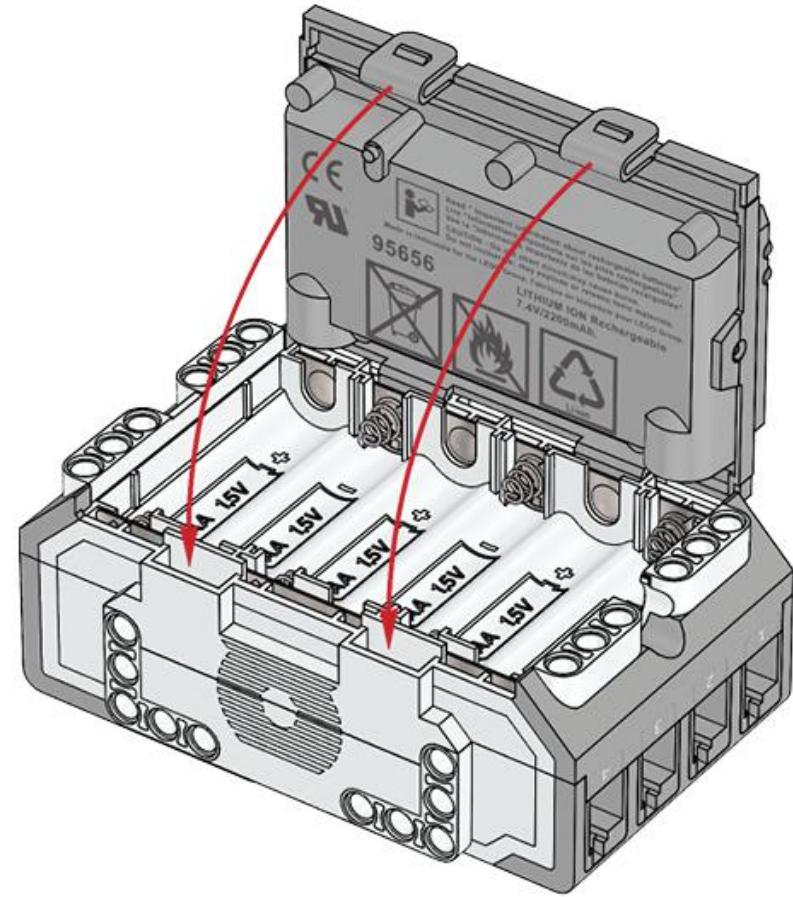
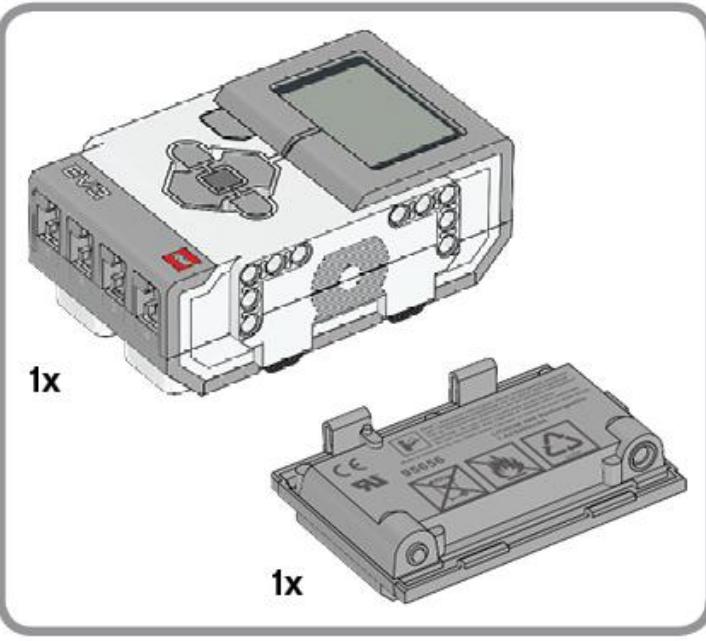
2x

33



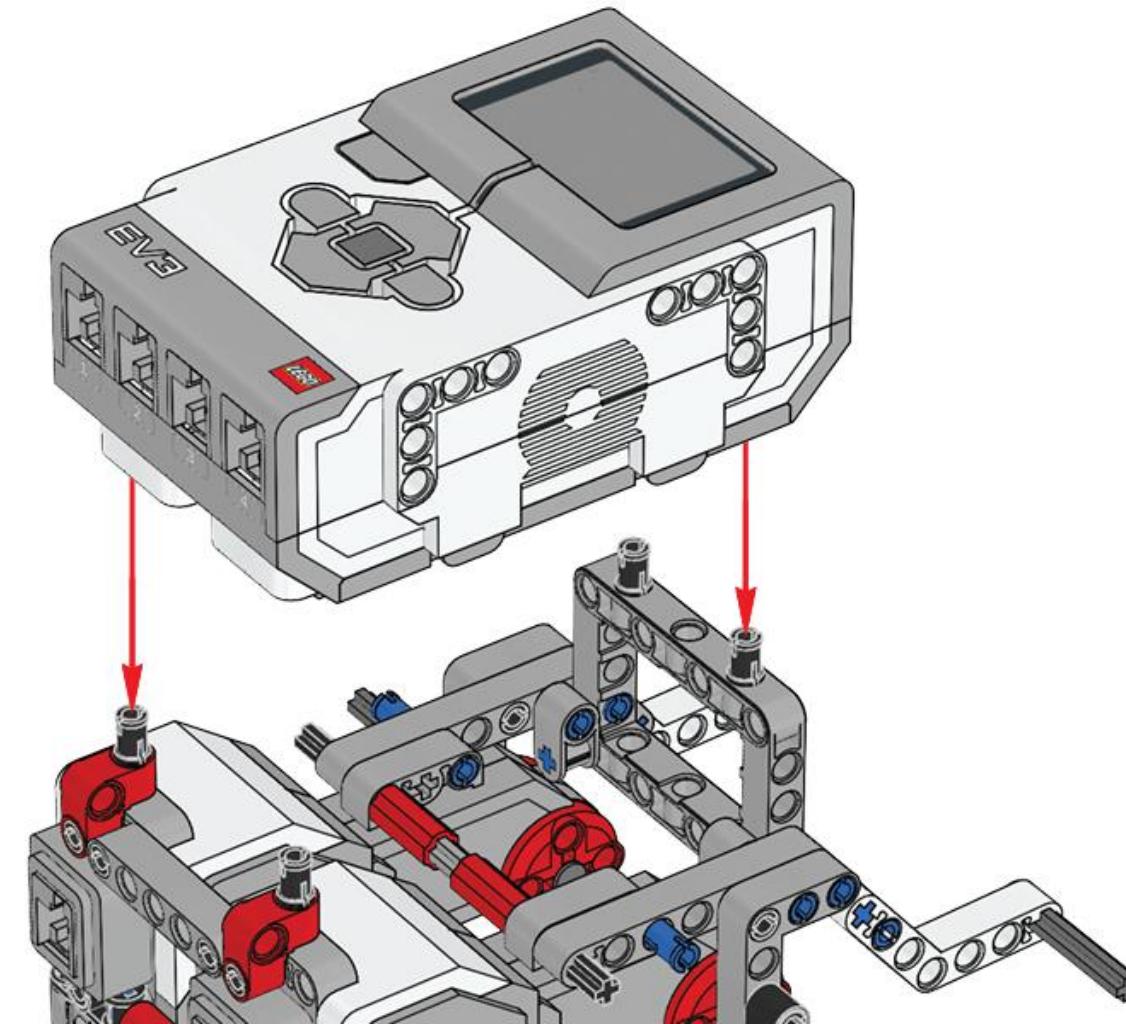
34

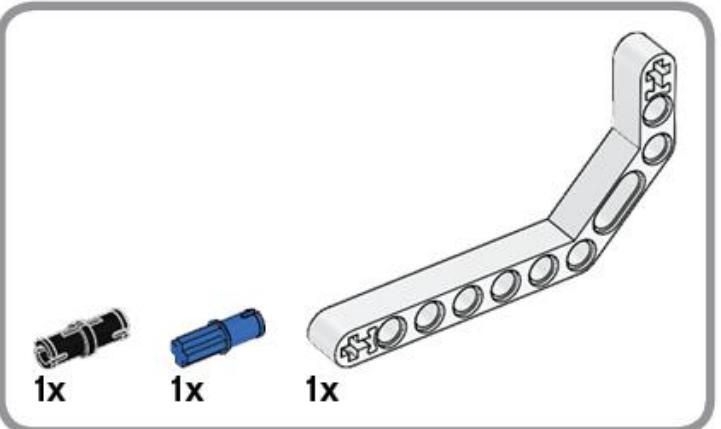




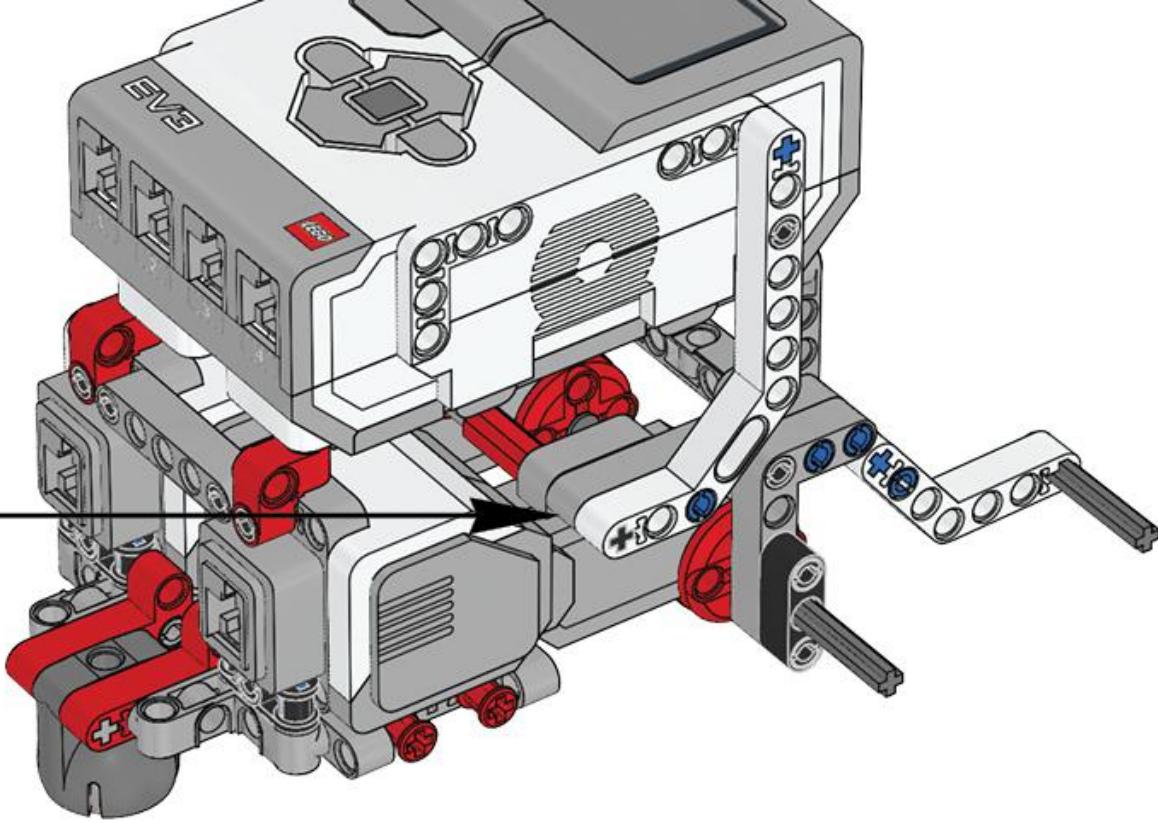
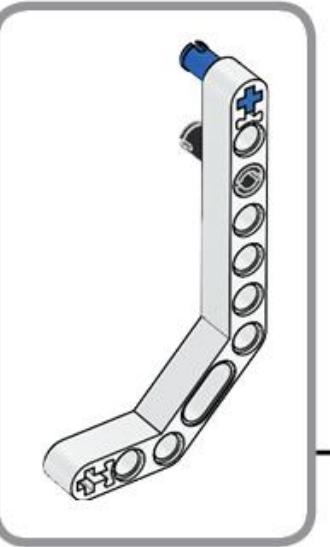
35

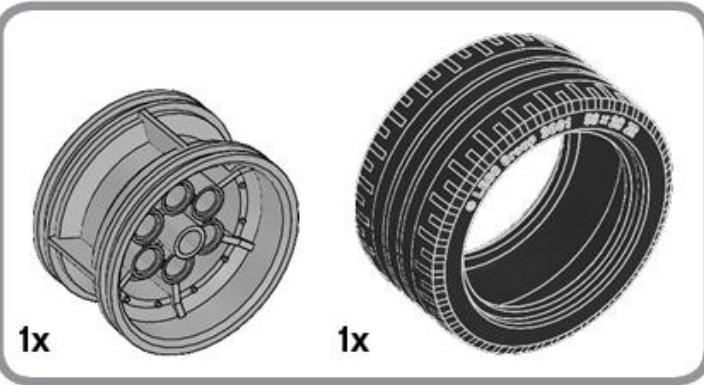
36



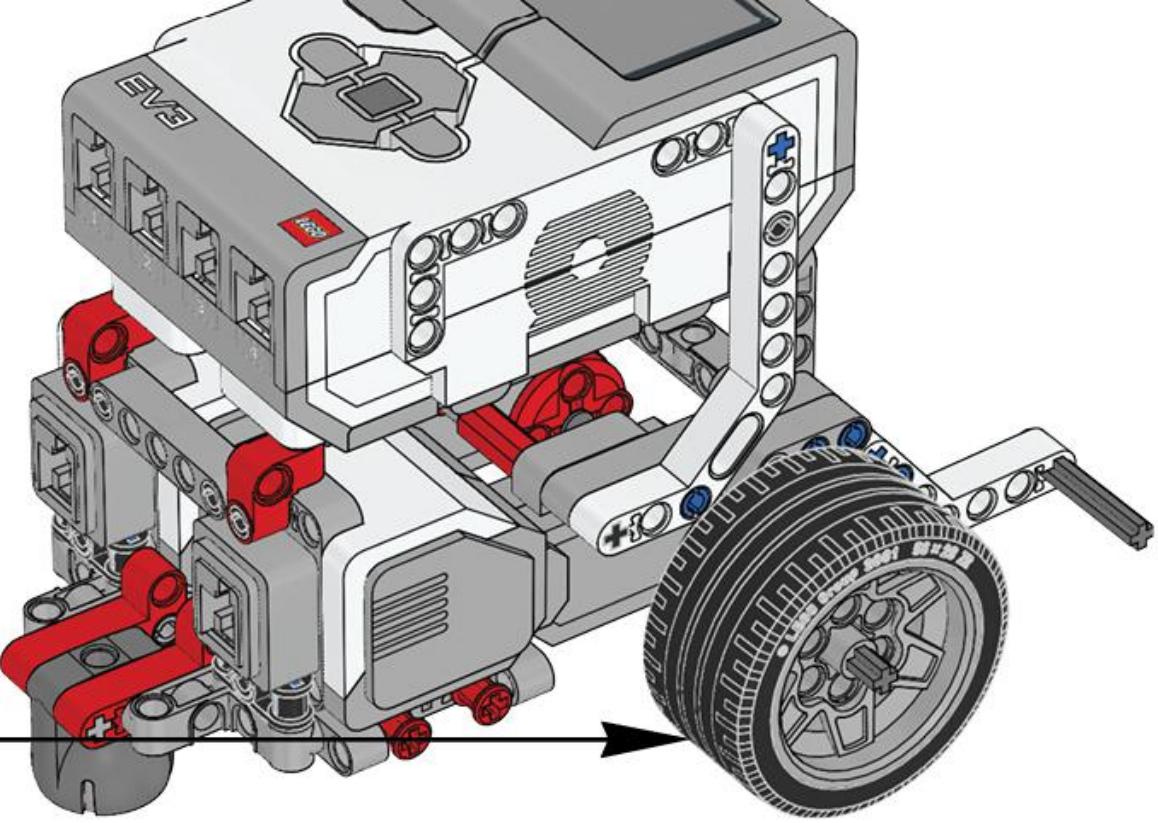
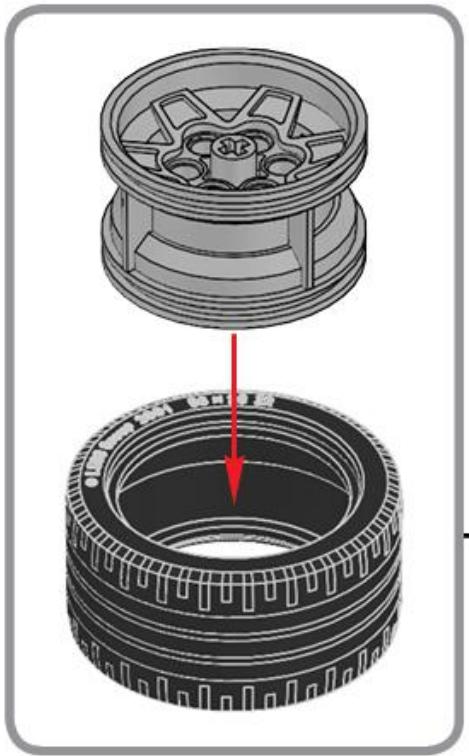


37

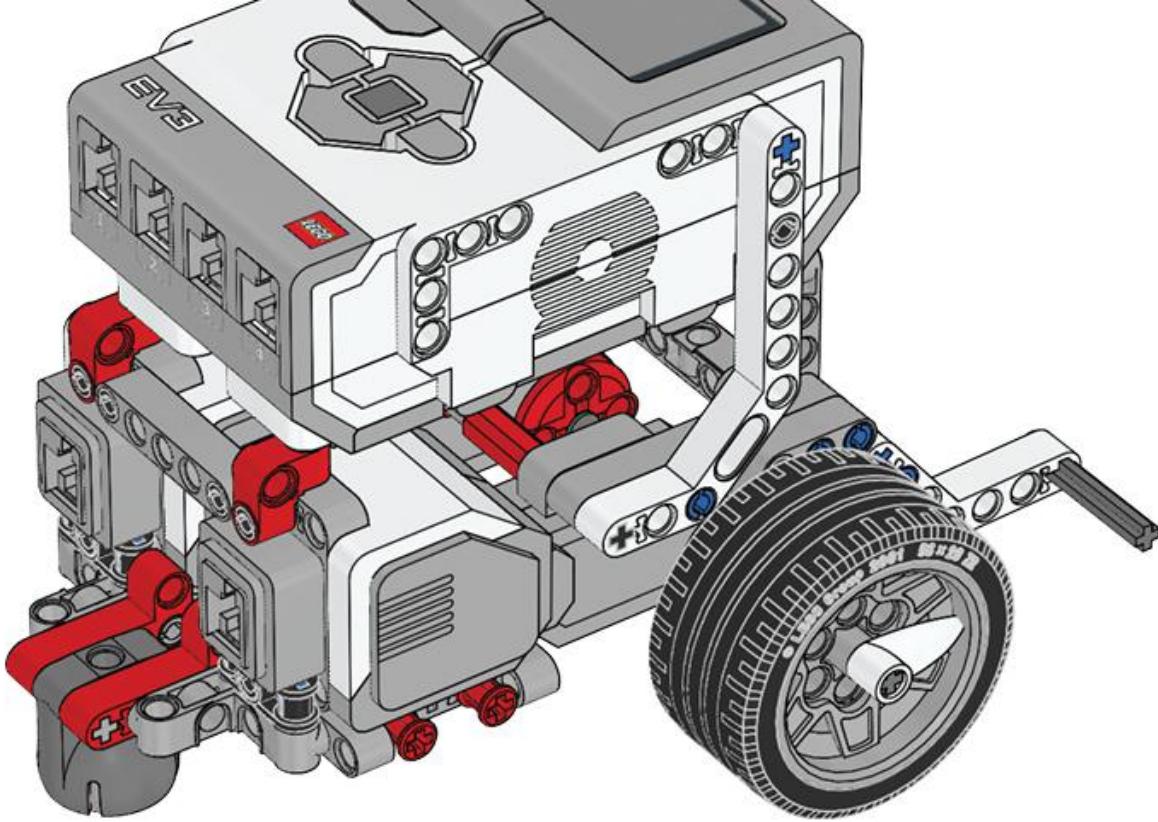
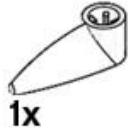


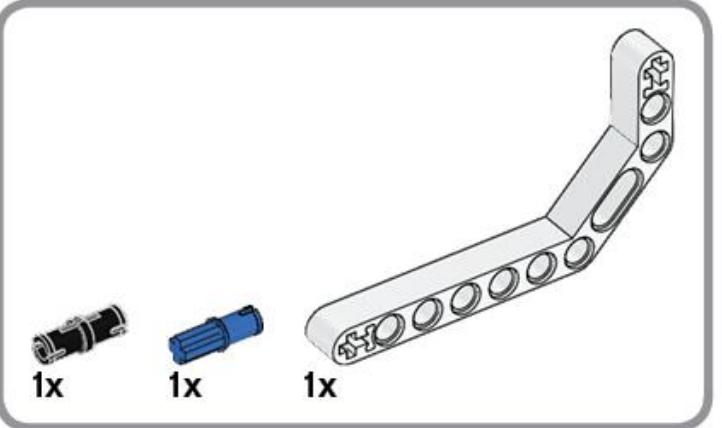


38

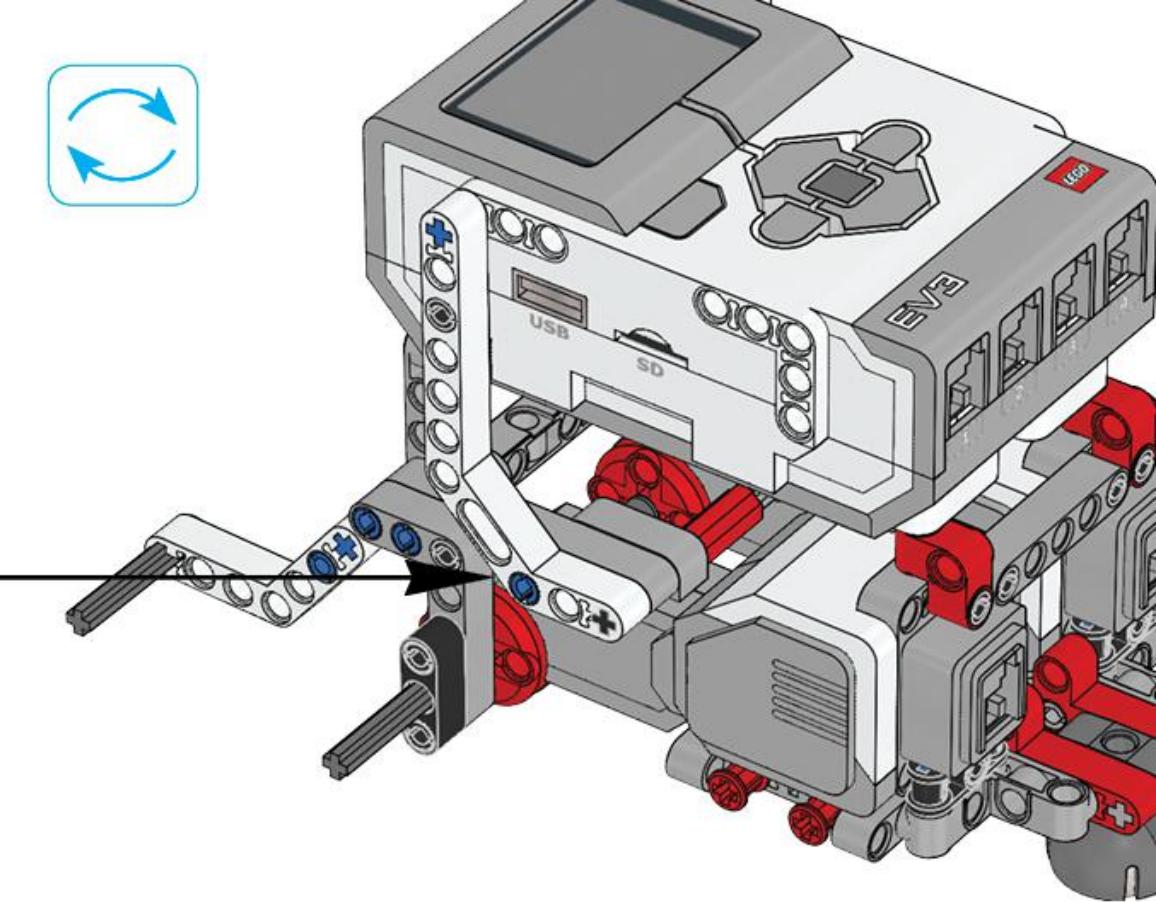
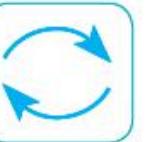
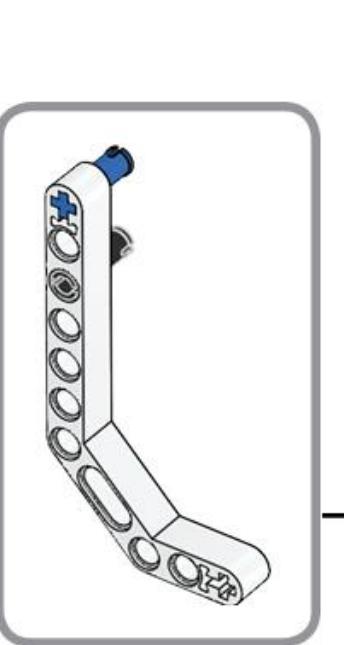


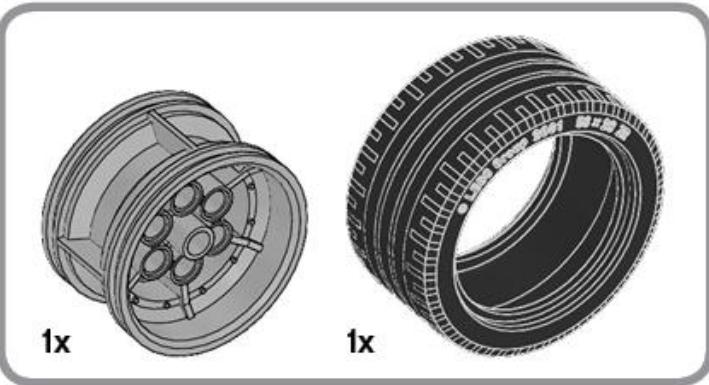
39



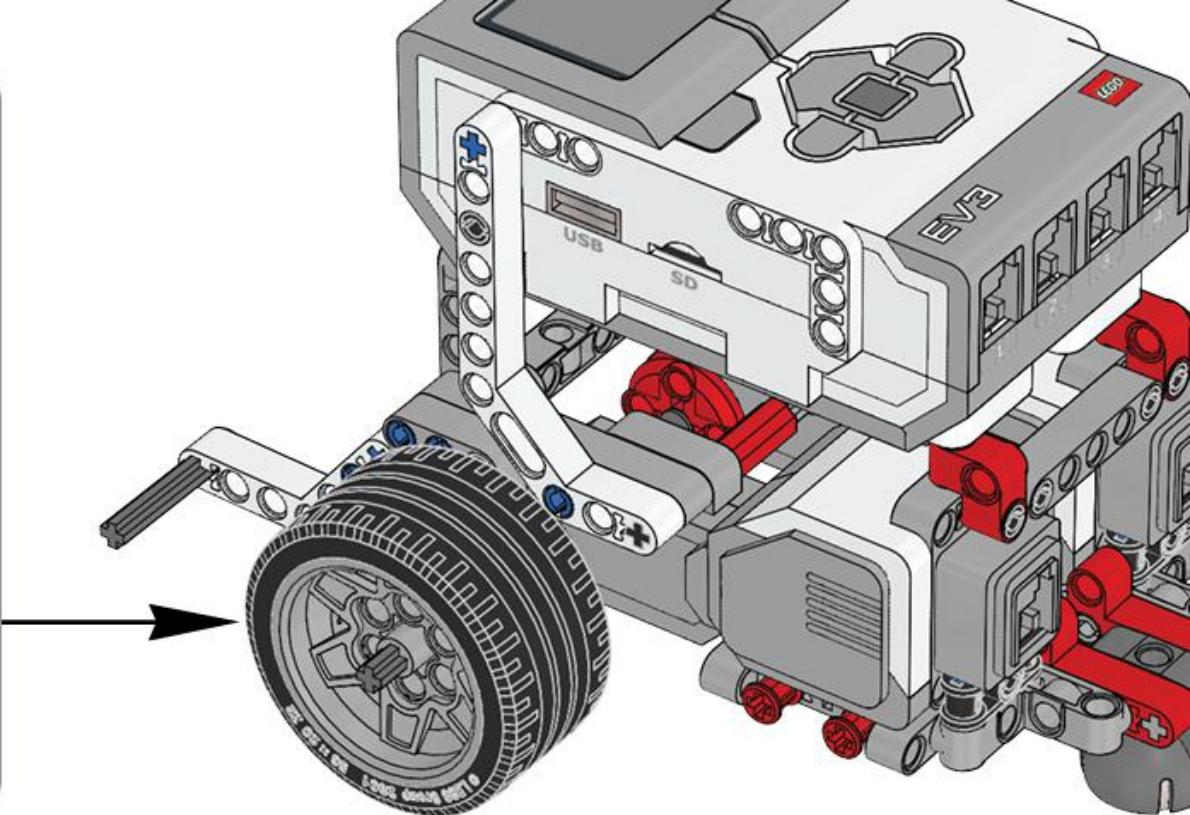
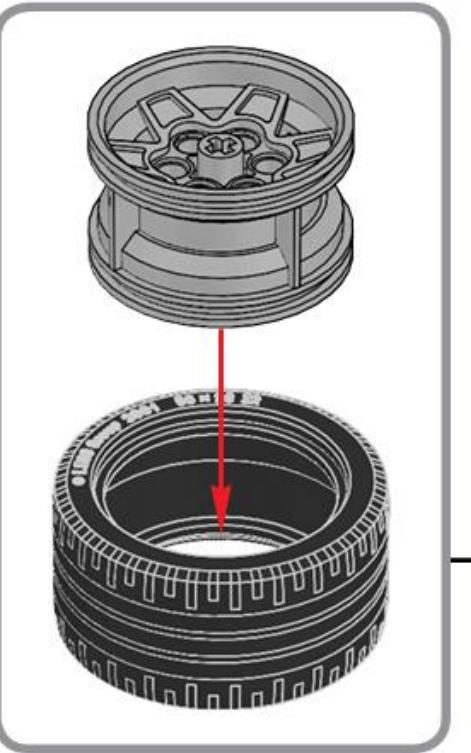


40

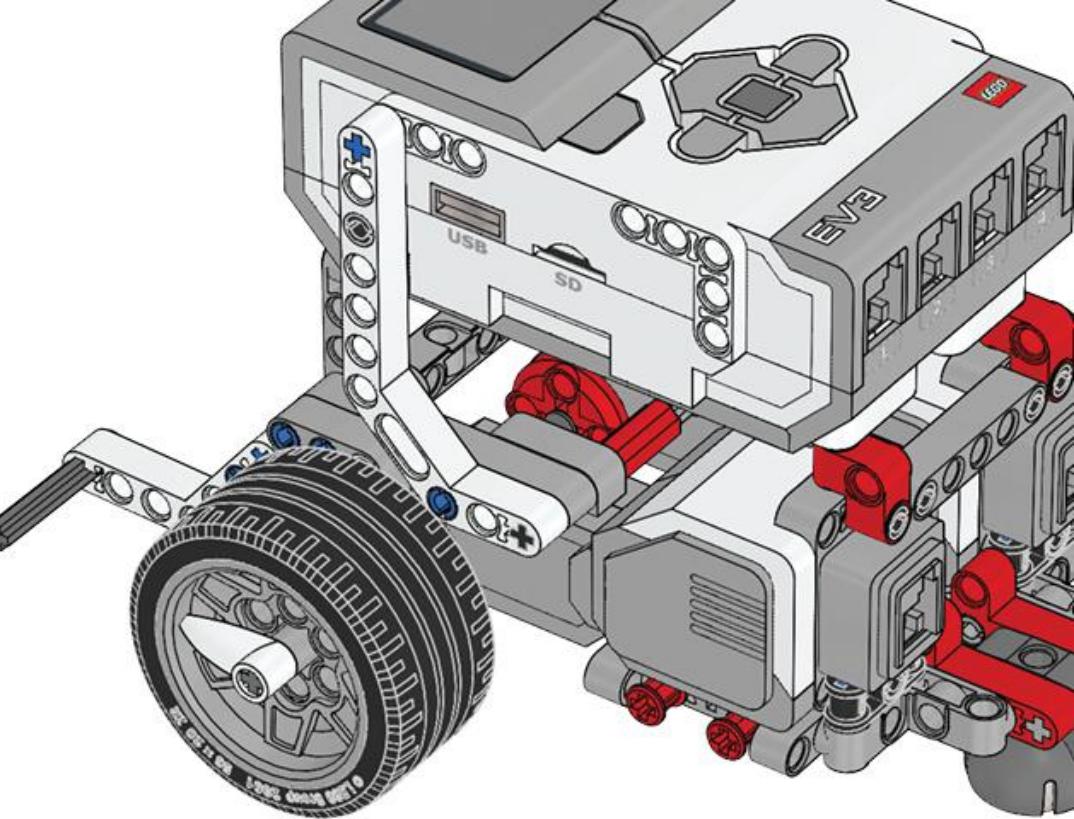
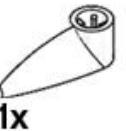




41

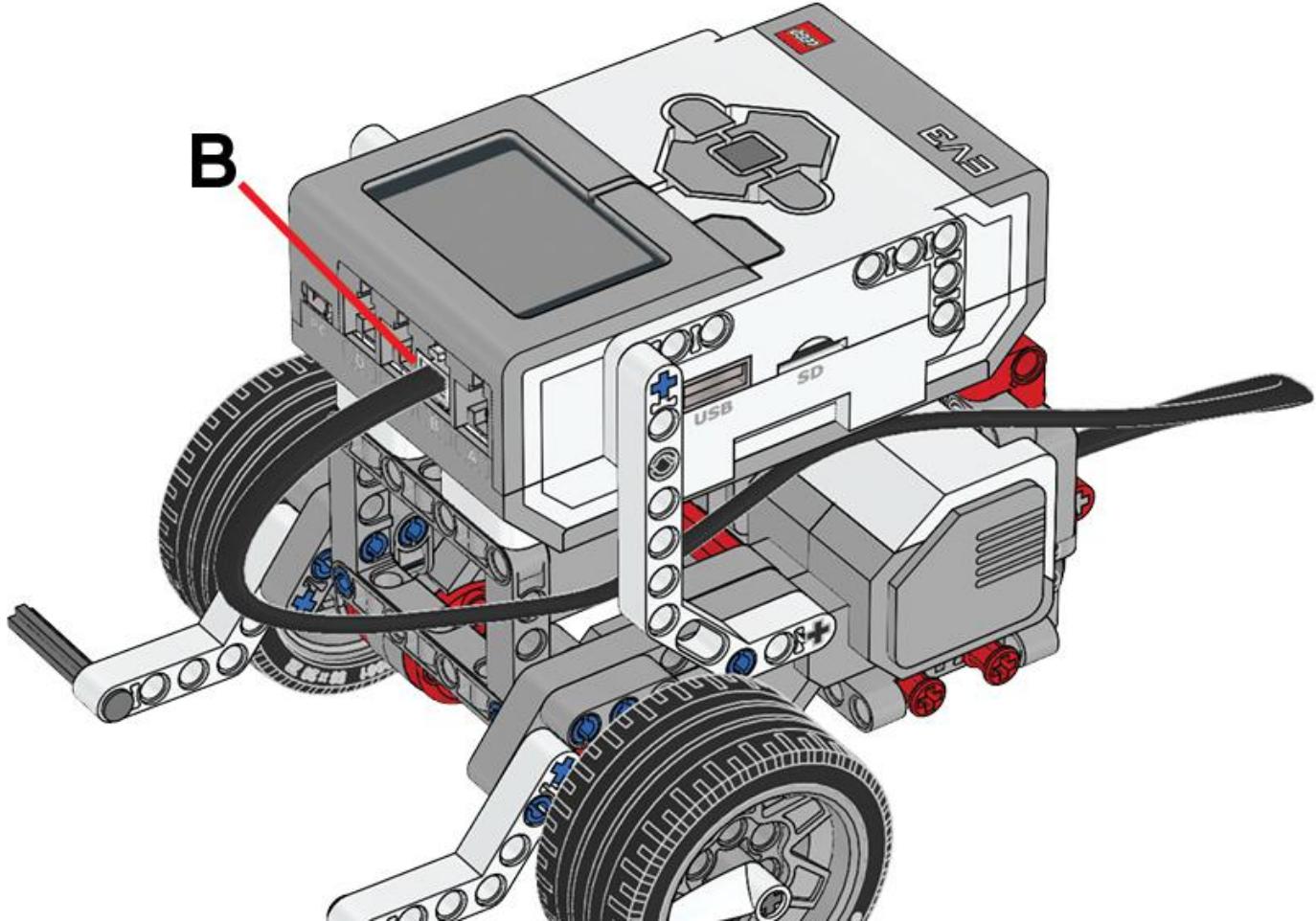


42





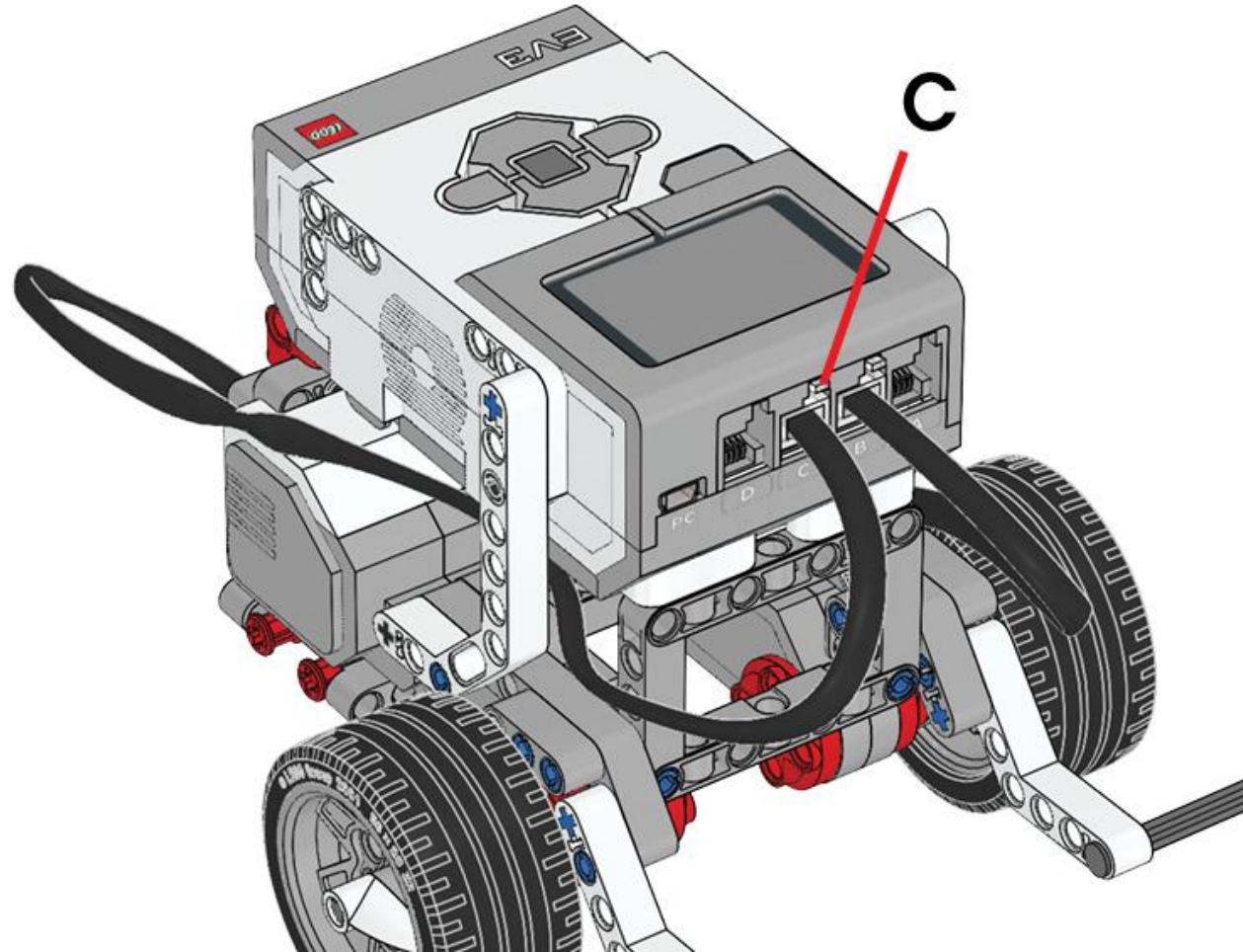
43





35 cm / 14 in.

44



45

