第五組 - 實驗三
辨識分類機器人

教師：黃漢邦、林沛群　教授

R09522849　陳品存

R10522801　吳政彥

R10522814　陳政豪

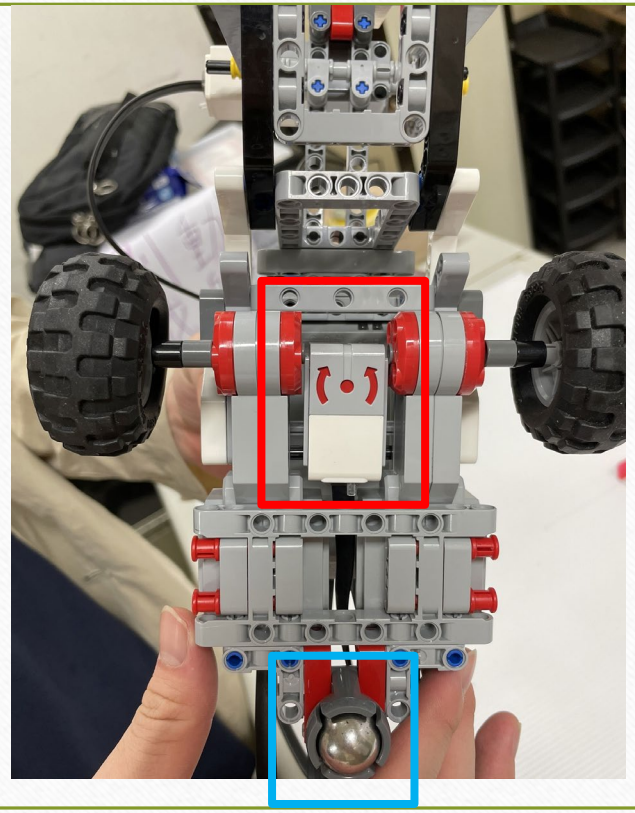
R10522834　戴承寧

車體機構

車體介紹

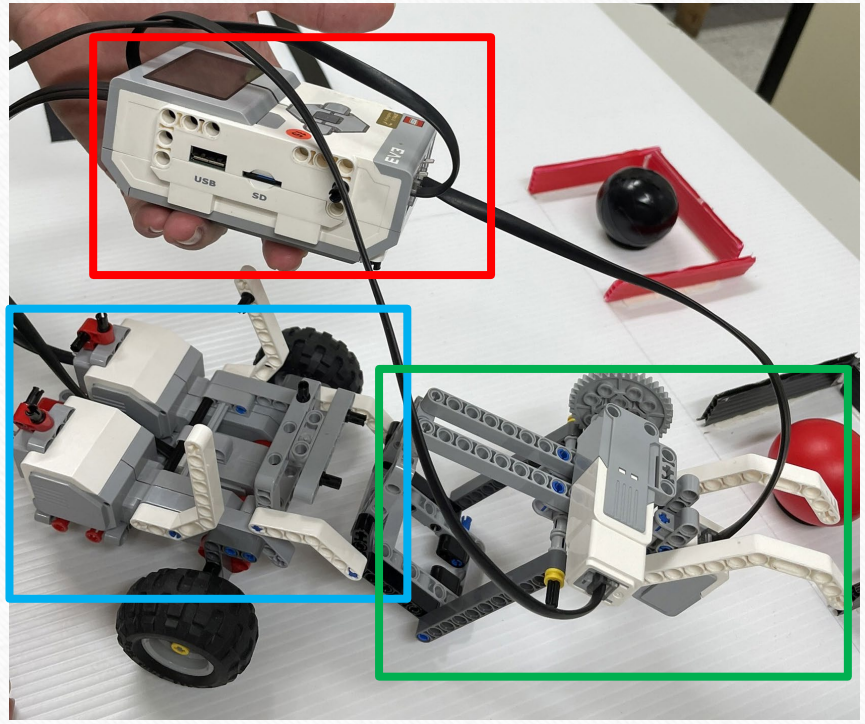
- 此次實驗，我們從「實驗一－尋跡投筆車」的車體進行改良，並加上一顆陀螺儀至車體中，因此詳細設計步驟請見「附錄一：車體設計圖」。

車體下視圖



- 從下視圖可知，我們使用了陀螺儀（紅色框框部份）用來感測與維持方向，使我們再行進間移動可以更加精確。
- 以及及一顆滾輪（藍色框框部份），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。

車體爆炸圖

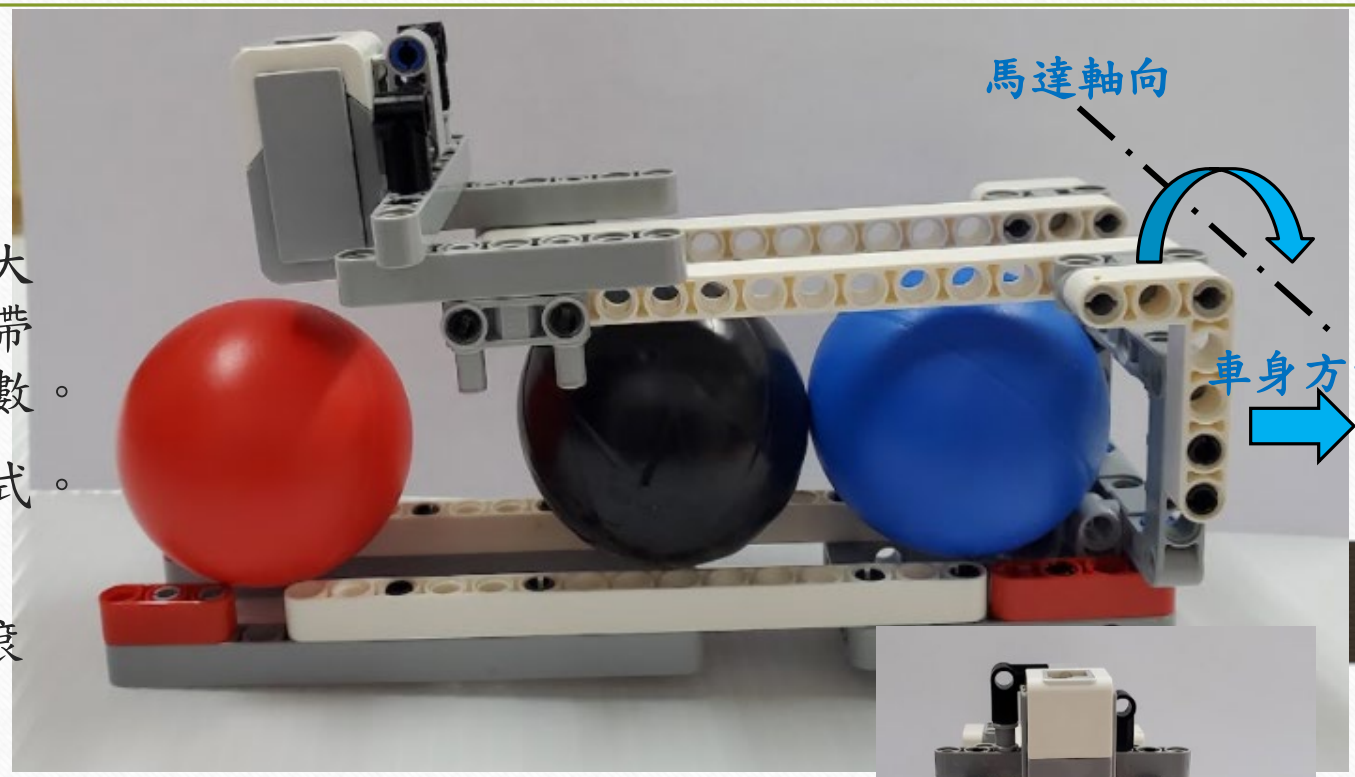


- 可以看到我們整體由三個部分所組成，分別是主機（紅色框框部份）、移動機構（藍色框框部份）以及夾爪機構（綠色框框部份）。

夾爪（球）機構

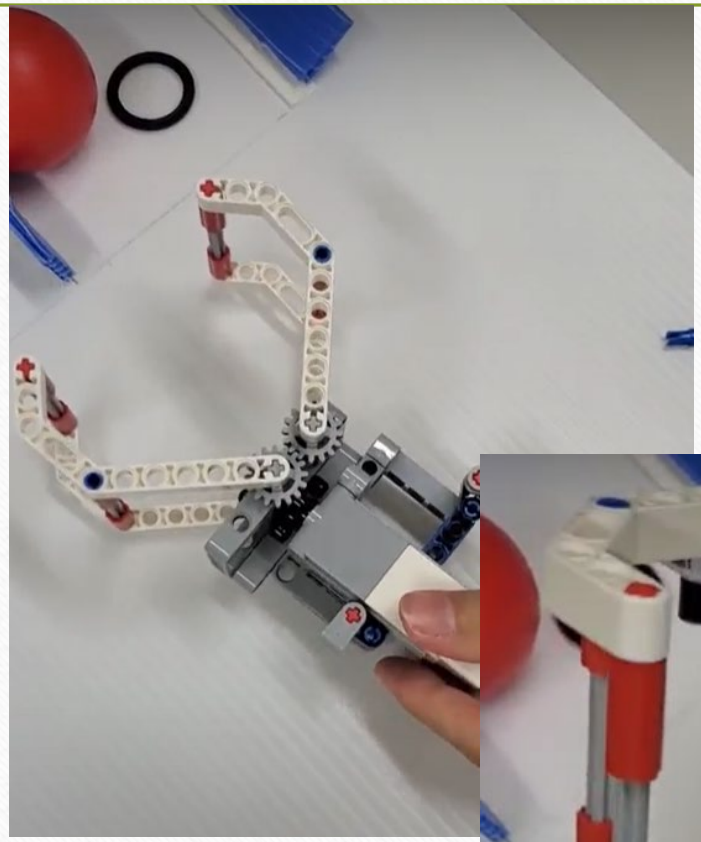
第一代夾爪

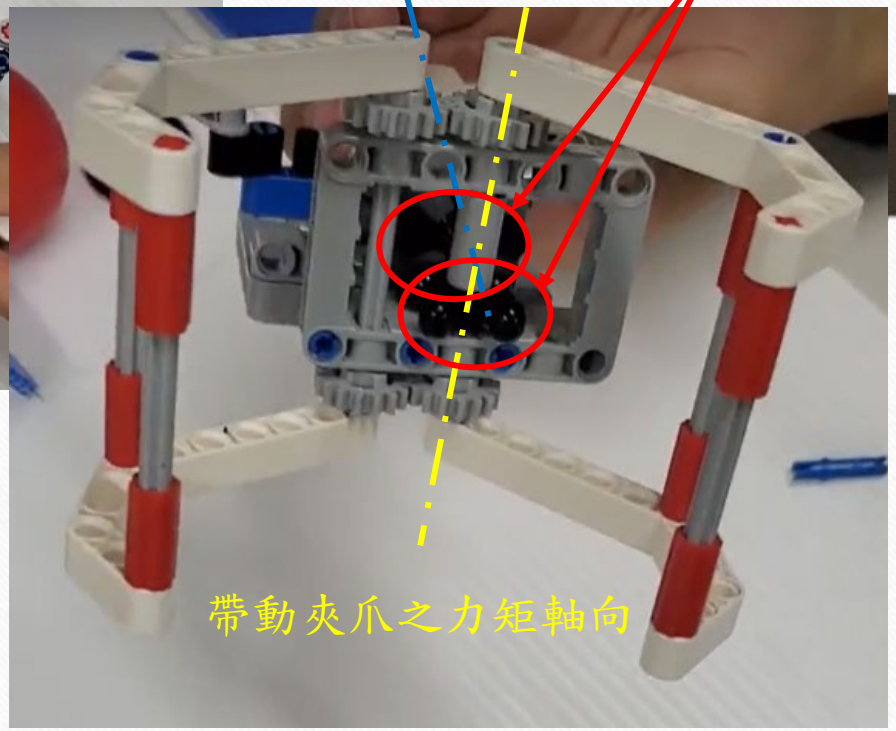
- **動機**：因車輛移動趟數愈多會造成愈大的位置偏差，希望能將三顆球同時攜帶在車輛上，以減少來回拿球的移動次數。
- **方法**：設計成能儲放三顆球的軌道形式。
  - 1. 拾取球時將下方軌道插入球兩側，鏟起球後軌道後傾使球向後滾入軌道內。
  - 2. 放球時將軌道前傾，使球向前滾出。
- **缺點**：軌道機構較大，對馬達負擔吃重且不穩定。之後改變策略以一次夾取單球來完成任務。



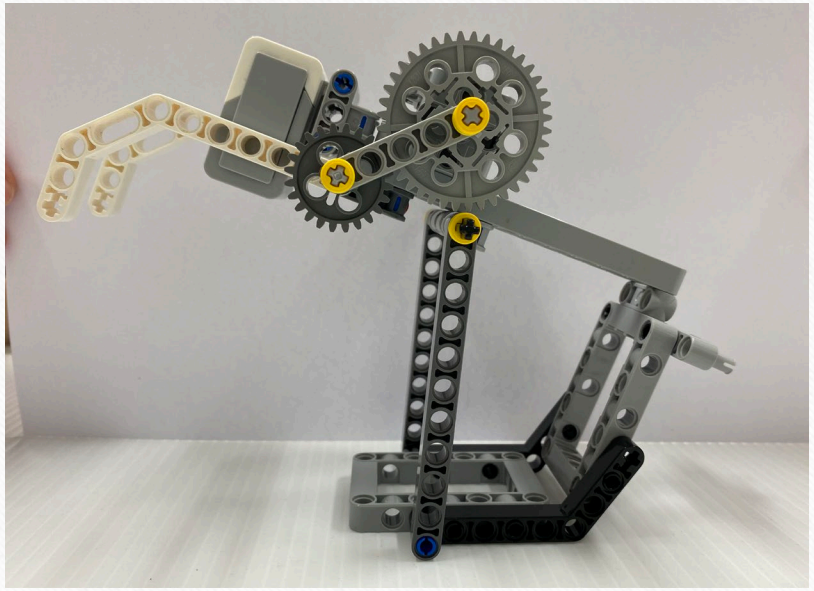
第二代夾爪

- **動機**：以側邊夾取的夾爪分次夾取球來完成任務
- **方法**：利用四齒轉向齒輪使馬達水平為轉軸的力矩轉向為垂直的力矩，再由四顆馬達同步帶動兩邊四支夾爪同步夾取。
- **缺點**：
  - 左右張開夾爪時很容易和邊界碰撞
  - 轉向齒輪只有四齒，齒輪背隙十分大，造成夾爪開合角度不好掌握



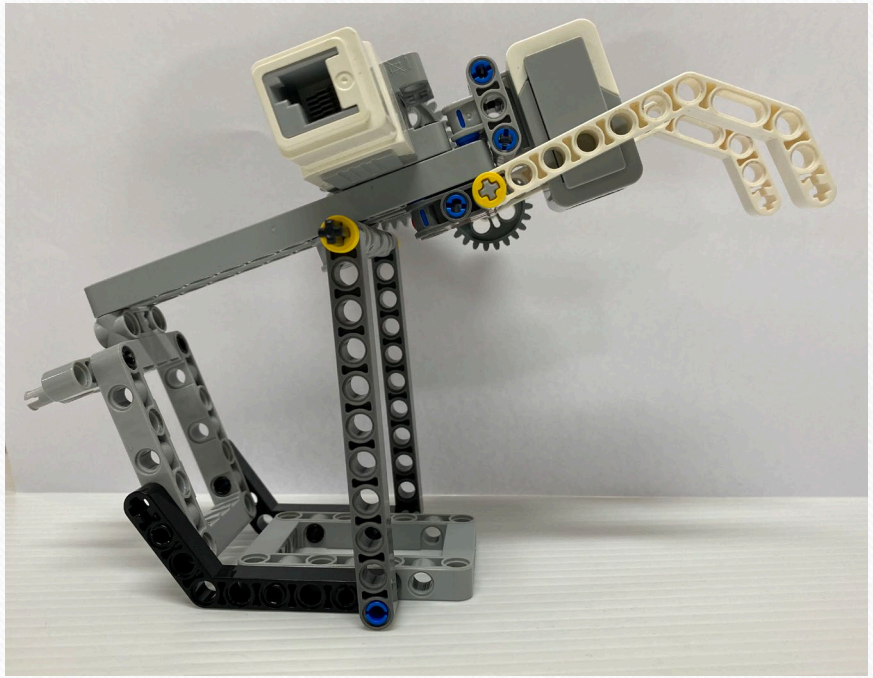


第三代夾爪 (1/4)

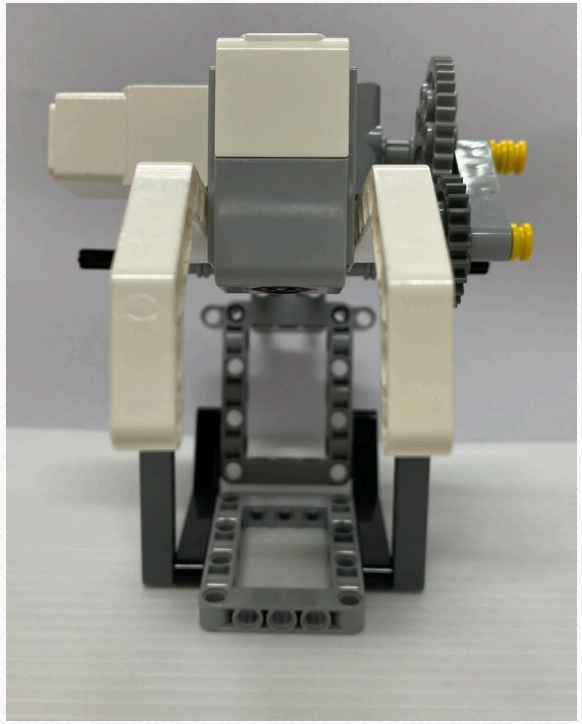


左視圖

右視圖

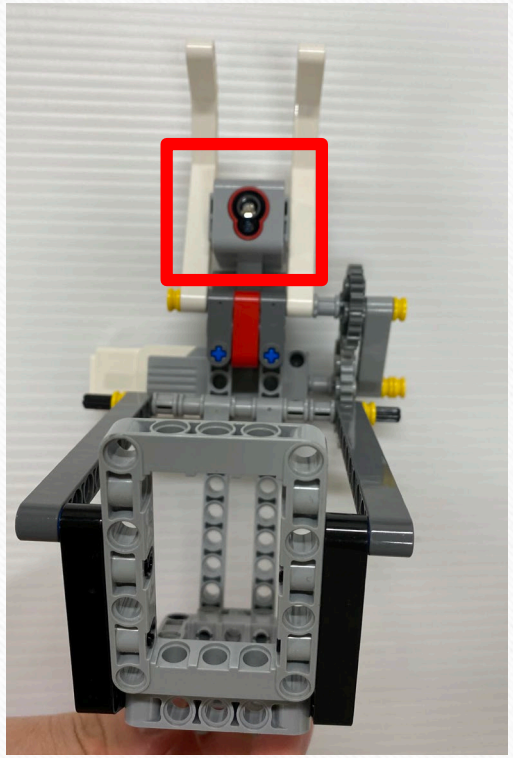


第三代夾爪 (2/4)



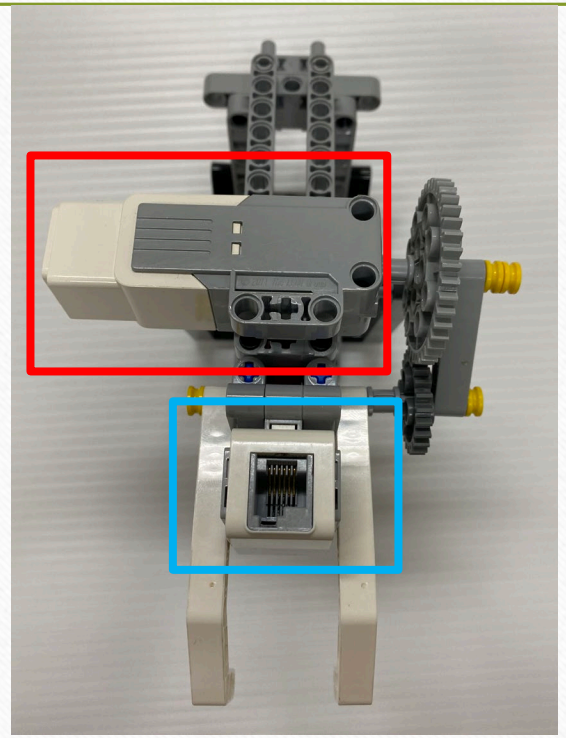
前視圖

第三代夾爪 (3/4)



根據這次實驗夾球的需求，我們將夾爪設計成夾娃娃機夾爪型態，方便我們夾取球，而在夾爪正後方，也就是圖中紅框位置，我們放置顏色感測器用來辨認不同顏色的球。

第三代夾爪 (4/4)

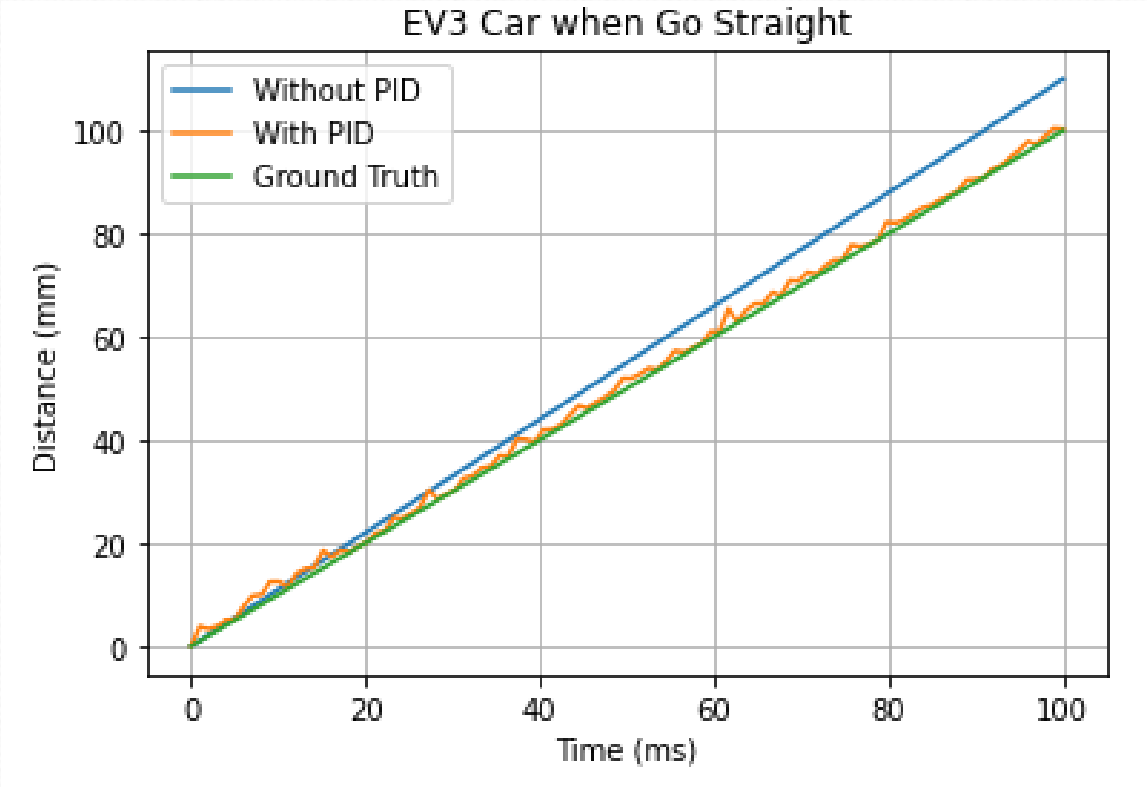


- 從上視圖可知，我們使用了一顆EV3伺服馬達（紅色框框部份）來控制夾爪的開闔。
- 而上一頁提到的顏色感測器則是藍色框框部份。

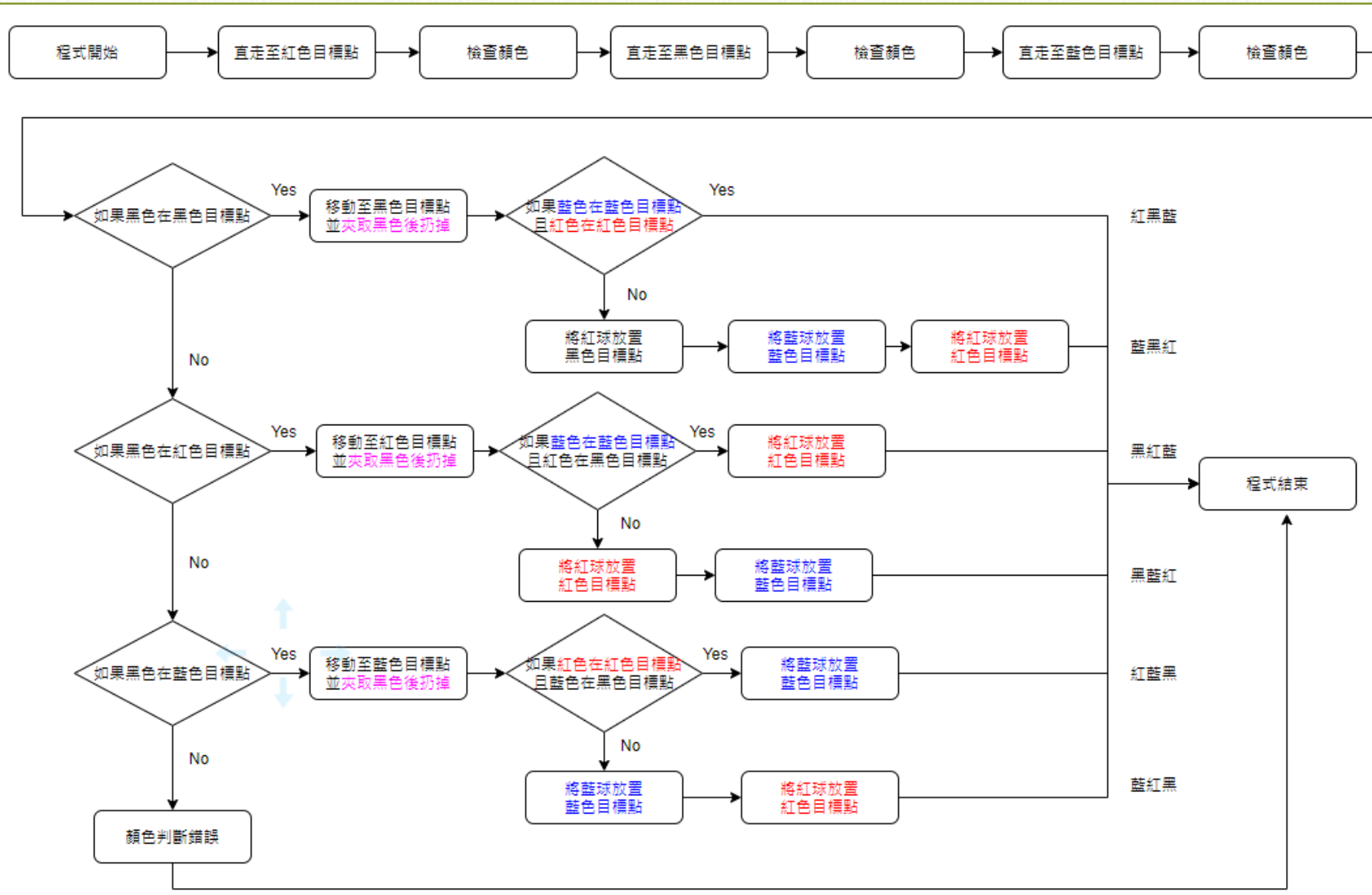
程式解說

程式邏輯發想

透過讓EV3車總共走4趟，第1趟先辨識各個位置分別放什麼顏色的球，第2趟將黑球取出並丟置一旁，第3趟將紅球分類正確，最後一趟將藍球分類正確。但礙於輪子在行走時會變形造成外八，且馬達也無法如預期轉至設定的角度，使得車子行走距離過長會出現誤差累積的現象，因此多加裝陀螺儀並結合PID控制器來處理。



整體程式流程


程式開始
直走至紅色目標點
檢查顏色
直走至黑色目標點
檢查顏色
直走至藍色目標點
檢查顏色
如果黑色在黑色目標點
Yes
移動至黑色目標點
並夾取黑色後扔掉
如果藍色在藍色目標點
且紅色在紅色目標點
Yes
紅黑藍
No
將紅球放置
黑色目標點
將藍球放置
藍色目標點
將紅球放置
紅色目標點
藍黑紅
No
如果黑色在紅色目標點
Yes
移動至紅色目標點
並夾取黑色後扔掉
如果藍色在藍色目標點
且紅色在黑色目標點
Yes
將紅球放置
紅色目標點
黑紅藍
No
將紅球放置
紅色目標點
將藍球放置
藍色目標點
黑藍紅
No
如果黑色在藍色目標點
Yes
移動至藍色目標點
並夾取黑色後扔掉
如果紅色在紅色目標點
且藍色在黑色目標點
Yes
將藍球放置
藍色目標點
紅藍黑
No
將藍球放置
藍色目標點
將紅球放置
紅色目標點
藍紅黑
No
顏色判斷錯誤
程式結束

顏色資料存取

```
# [correct ball color, distance, current ball color, found order, is the color correct]
ballInfo = [[Color.RED, 225, None, 3, False], [Color.BLACK, 210, None, 3, False], [Color.BLUE, 210, None, 3, False]]
```

- 儲存目標點顏色與目前在目標點的球顏色是否相同
- 紀錄目標點顏色的球找到的順序
- 紀錄目前在目標點是什麼顏色的球
- 前往該目標點所需要移動的距離
- 目標點的顏色

直走程式實現

PID控制器

```python
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
```

EV3車直走

```python
if distance > 0: # 往前走
    while mobile_car.distance() < distance:
        err = -1 * degree - gSensor.angle()
        gain = ctrl.control(err, sum_err, previous_err) # PID 修正角度
        previous_err = err
        sum_err += err
        mobile_car.drive(100, -gain)
        wait(dt)
else:            # 向後走
    while mobile_car.distance() > distance:
        err = -1 * degree - gSensor.angle()
        gain = ctrl.control(err, sum_err, previous_err) # PID 修正角度
        previous_err = err
        sum_err += err
        mobile_car.drive(-100, -gain)
        wait(dt)
```

旋轉程式實現

PID控制器

```python
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
```

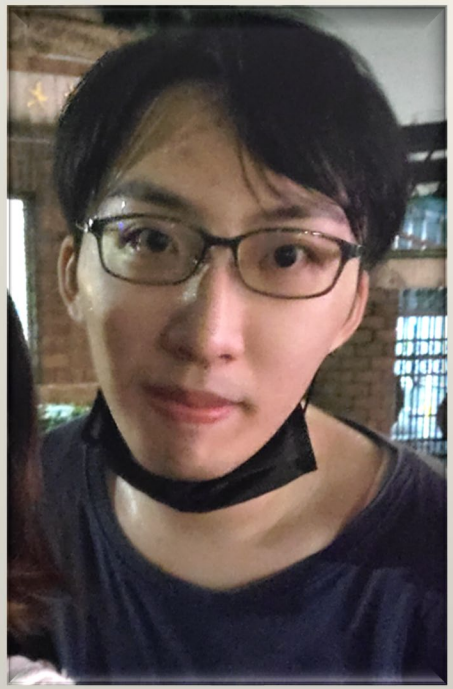
EV3車旋轉

```python
while True:
    if sign * gSensor.angle() <=  degree: # 先旋轉
        gain = 1.5
        err = 0
        sum_err = 0
        previous_err = 0
        dt = 5
        ctrl = PID_controller(2, 0.01, 5, dt)
        while True: # 再修正角度
            if sign * gSensor.angle() == degree:
                mobile_car.stop()
                break
            err = gSensor.angle() - sign * degree
            gain += ctrl.control(err, sum_err, previous_err)
            previous_err = err
            sum_err += err
            mobile_car.drive(0, gain)
            wait(dt)

        break

    mobile_car.drive(0, 30 * sign)
    wait(10)
```

組員心得



陳品存(R09522849)心得：

我認為這次實驗最麻煩的部份在於夾爪的設計，因為這會大大影響車體是否需要轉彎，而若要轉彎，則需要搭配陀螺儀以確保角度正確；如若不要轉彎，則需要將一個自由度放在夾爪從車體上收放，以及夾球放球共計兩個自由度。也因此，我們前期花了很大的時間在夾爪上的設計與改良，因此程式的部份也一再修改，真的辛苦大家了！

吳政秀(R10522801)心得：

在實驗三中花了許多時間在設計夾爪，雖然程式實現很快，但為了配合夾爪的長度與夾取角度須調整許多程式上的參數。原本程式想使用河內塔的邏輯來撰寫，但需要視黑球的位置來設定暫存點(Temp)、目標點(Target)與源點(Source)，且黑球位置不固定，所以實現上有些困難，幸好只有三種顏色球，因此可以直接透過窮舉6種可能情況來完成。實驗過程中發現隨著車子移動路徑增長，誤差累積越大，因此機構上的感測器使用陀螺儀，將讀取的角度與預設角度的誤差使用PID修正，使得車子能行走的更穩定與精確。不知道是不是陀螺儀老舊或接觸不良等問題，有時會讀不到值使誤差增大，經過PID修正後會瘋狂旋轉不過好險8次實驗中大概只會發生1次。這次實驗雖然花較久的時間來完成，但獲益良多。

陳政豪(R10522814)心得：

這次樂高實驗相比上次困難許多，也可以說這次的實驗相當富有變化性，在夾爪設計上可以看到每組的創意發想，上次實驗是單純循跡且到定點後再將筆放進洞即可，但這次需要將球跟車子做連結，不論是夾取抑或是拖拉，一開始我們也是在如何將球抓住想了許久，也因此做了三個版本後才確定最終夾爪形式，我覺得這次實驗十分有趣！

戴承寧(R10522834)心得：

為了達成實驗任務這次也嘗試了許多不同的方案。在過程中最遇到主要的問題是都是出現在實現想法時，必須學習如何克服硬體的限制，例如各種感測器的誤差、機構重量等，有些可以藉由改變機構解決，有些真的是需要不斷地調整程式參數，但這些需要機構與程式的相互配合。感謝組員提出很多新奇的方法也很願意做各種嘗試，讓我見識到集思廣益的威力強大。

小組分工表

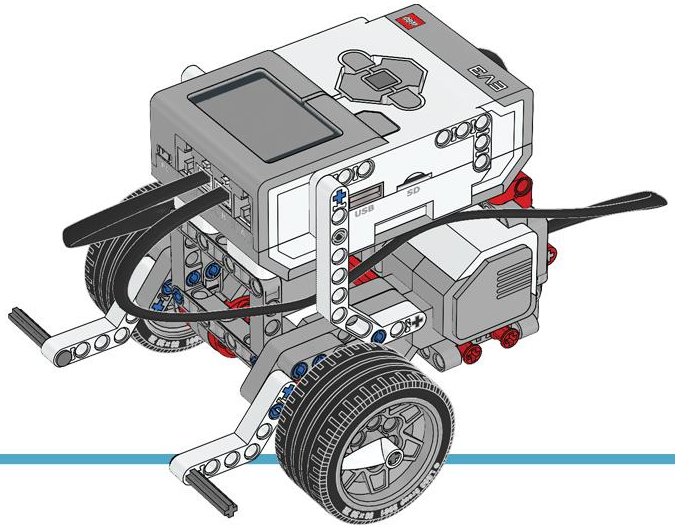
小組分工表

姓名	負責項目與內容
陳品存	機構設計（車體）與PPT統整
吳政彥	程式設計
陳政豪	機構設計（夾爪）
戴承寧	機構設計（夾爪）

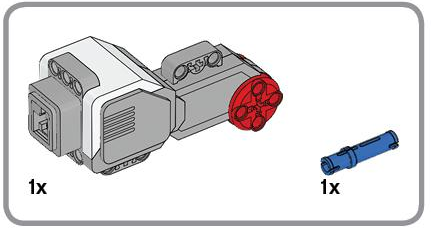
附錄一：車體設計圖

取自樂高LEGO官方網站：

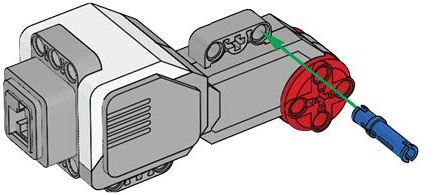
<http://robotsquare.com/wp-content/uploads/2013/10/45544_educator2.pdf?fbclid=IwAR3jZTeTXEsWJSgitKz14kpZ-RHtg56L_2806T0z7MU8S2lJC_3VepFiyl4>

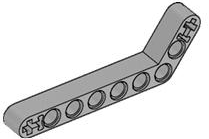


LEGO mindstorms education EV3



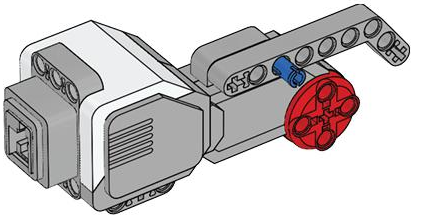
1

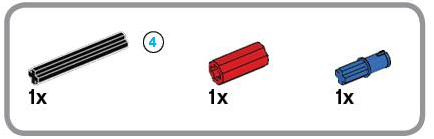




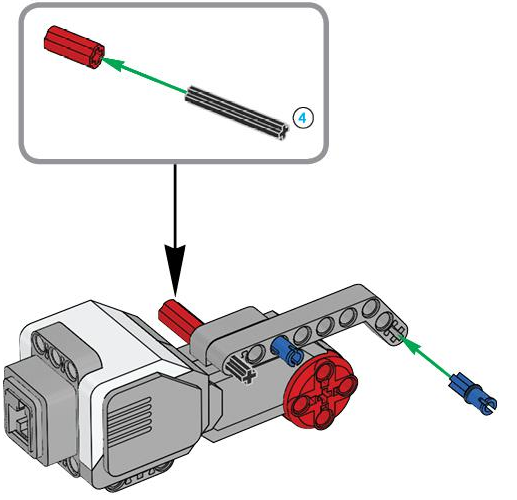
1x

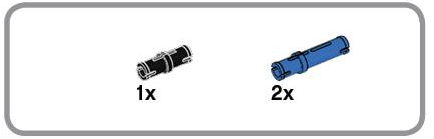
2



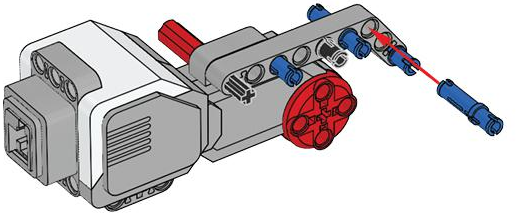


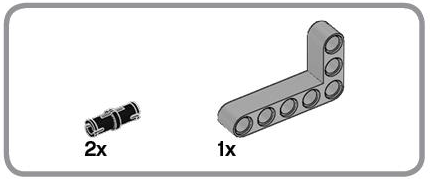
3



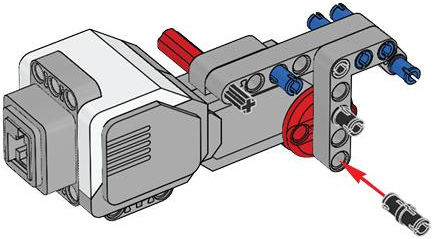


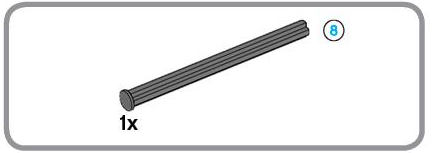
4



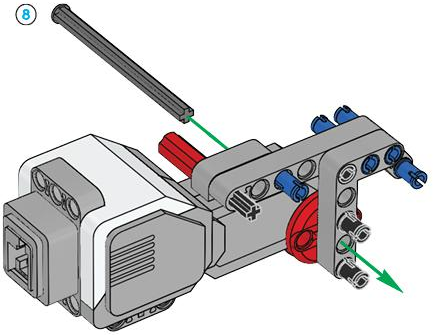


5



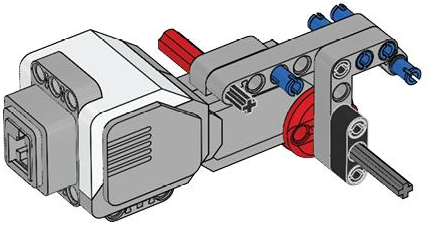


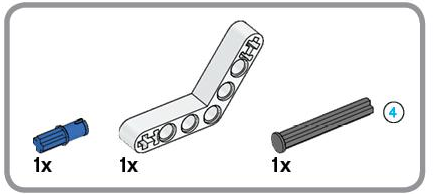
6



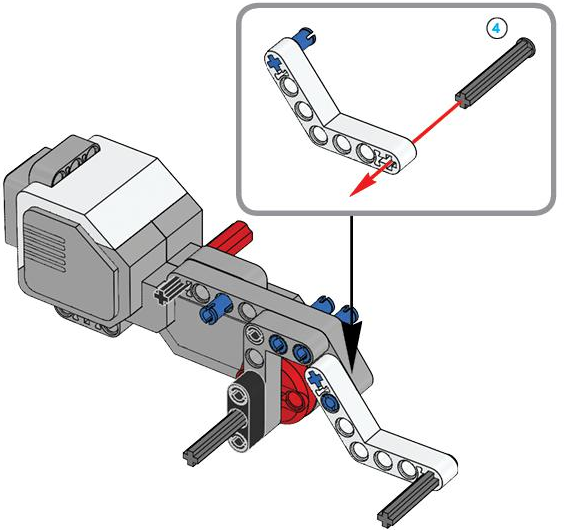
1x

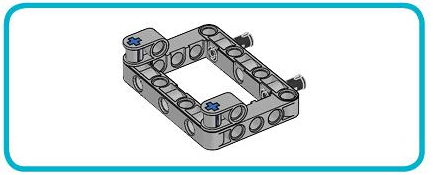
7

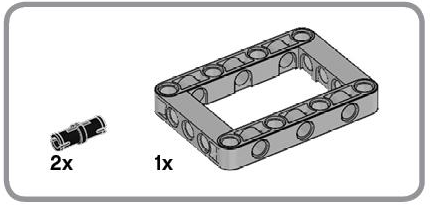




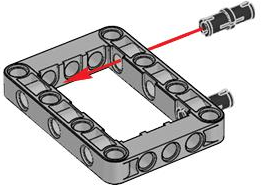
8





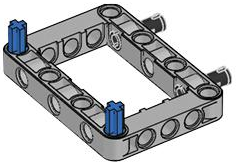


9



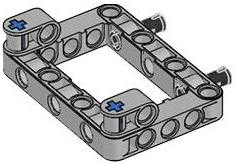
2x

10

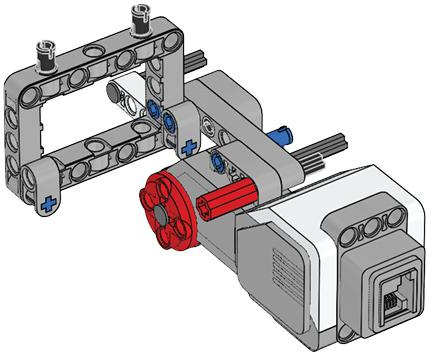


2x

11

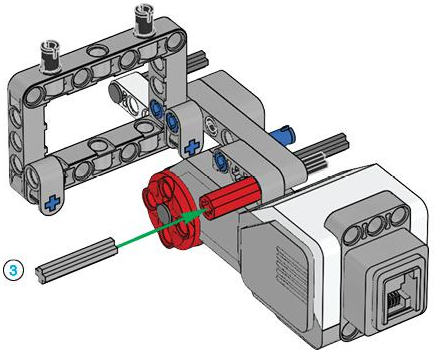


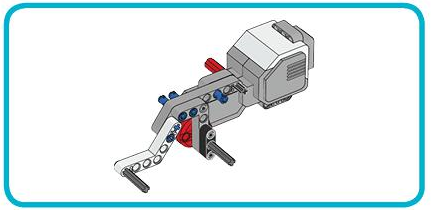
12

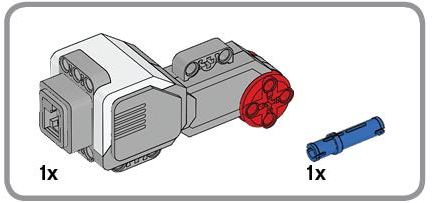


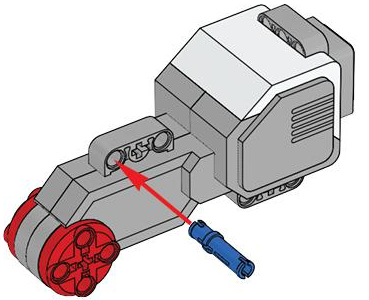
1x

13

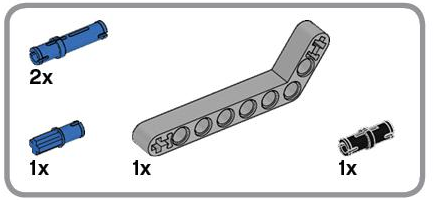




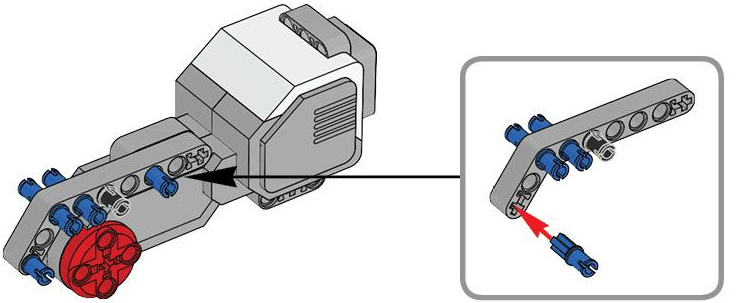


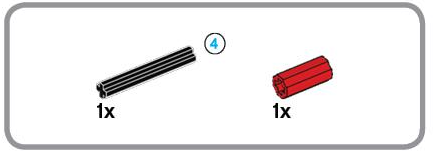


14

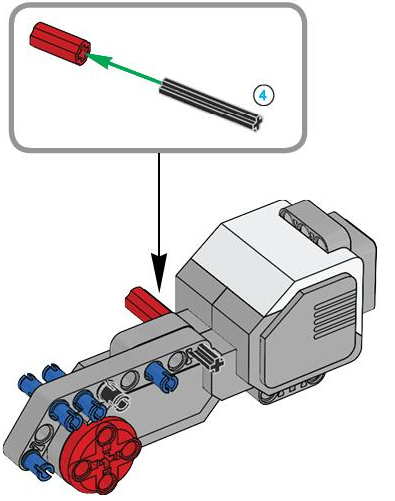


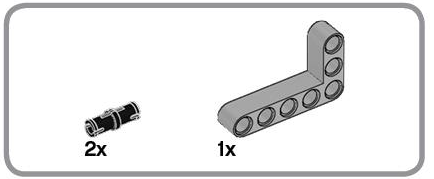
15



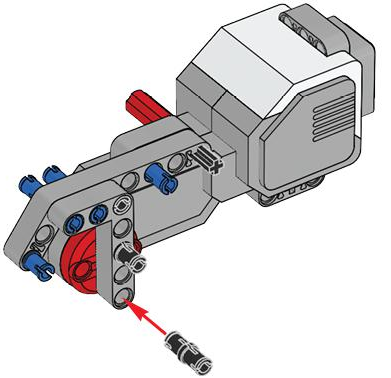


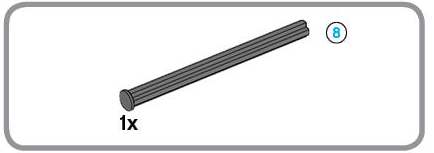
16



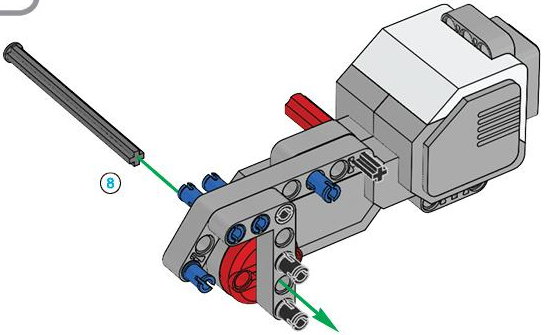


17

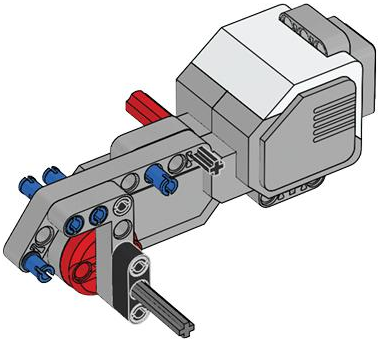


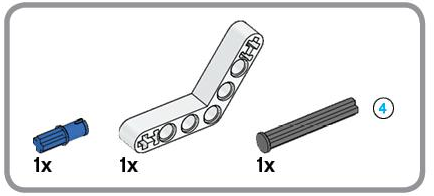


18

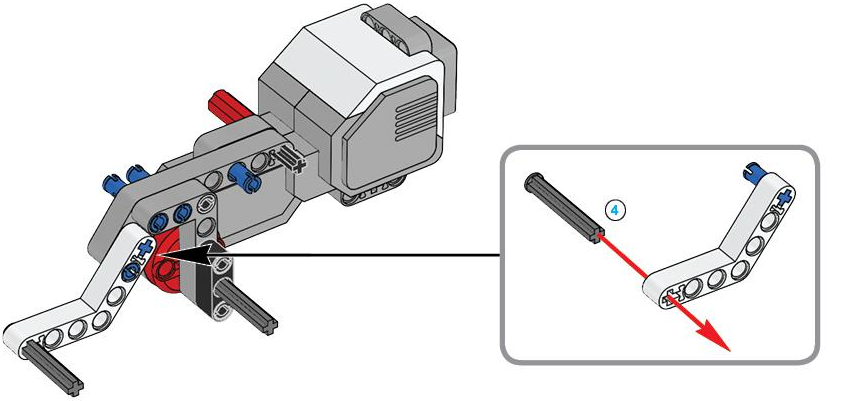


19

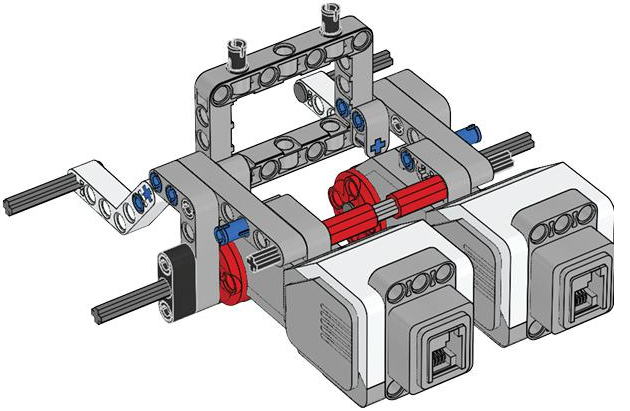


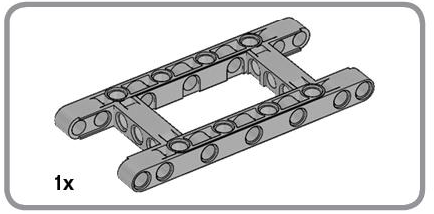


20

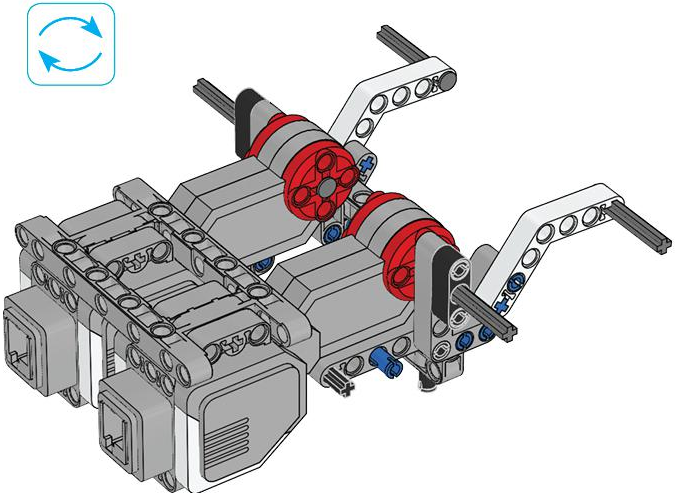


21

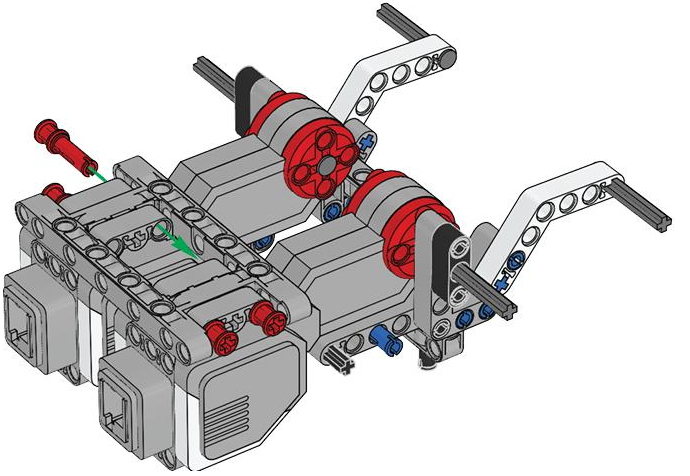


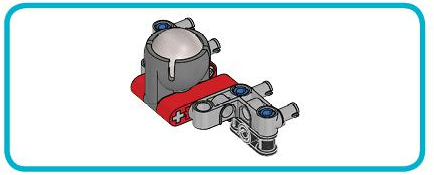


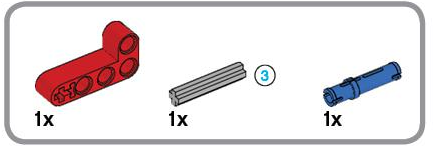
22



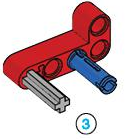
23

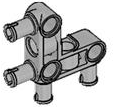






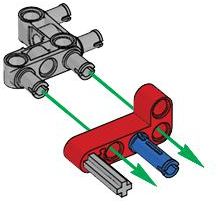
24





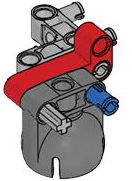
1x

25

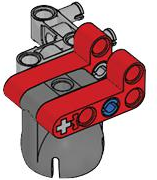


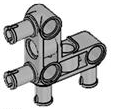
1x

26



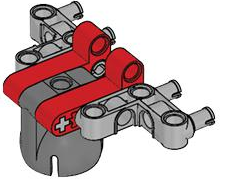
27

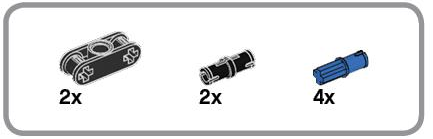




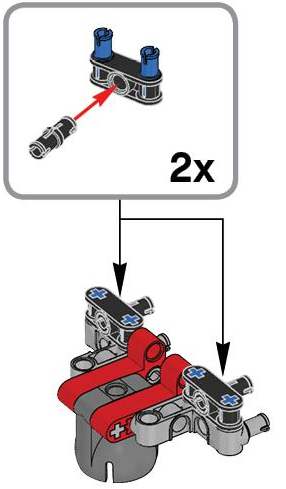
1x

28



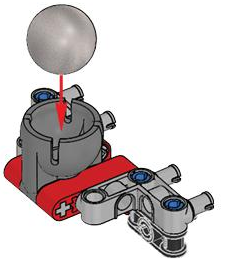


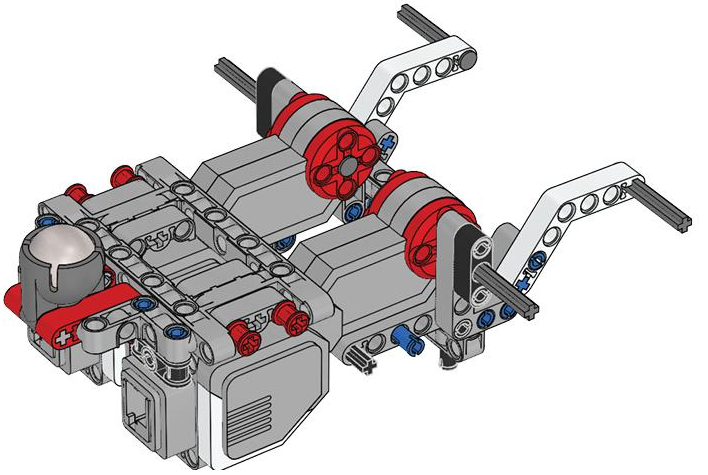
29



1x

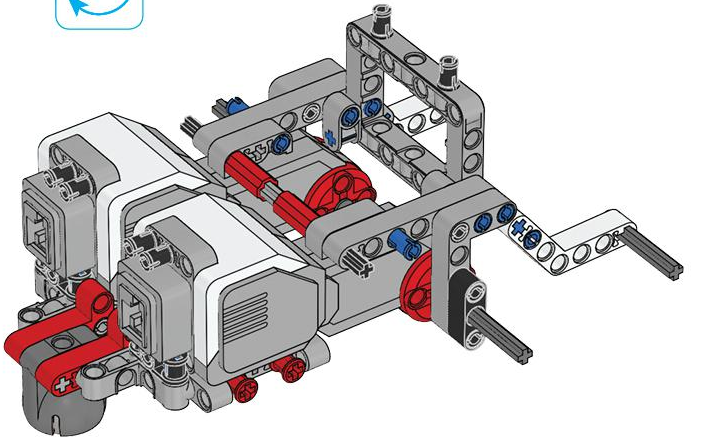
30

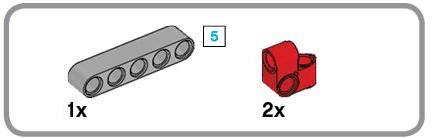




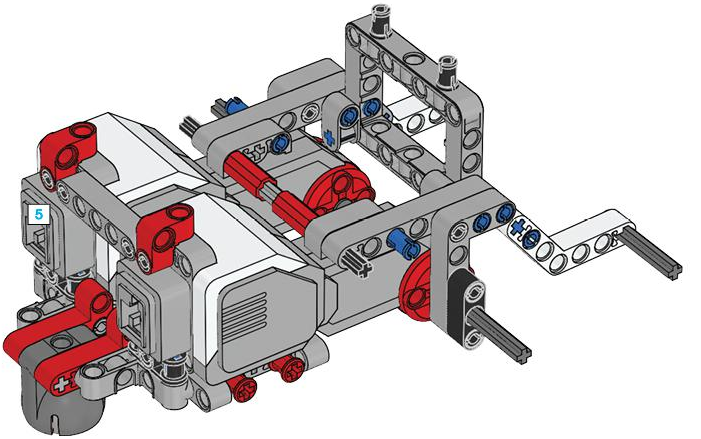
4x

32



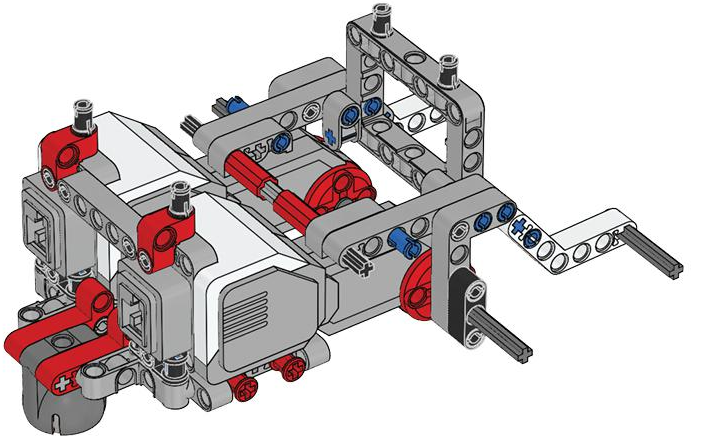


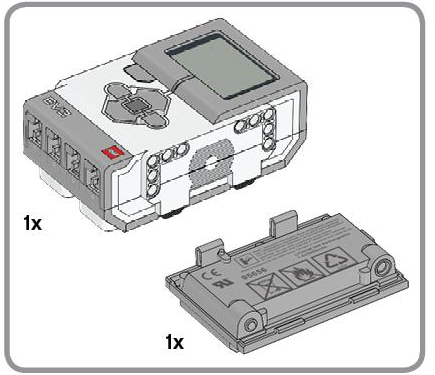
33



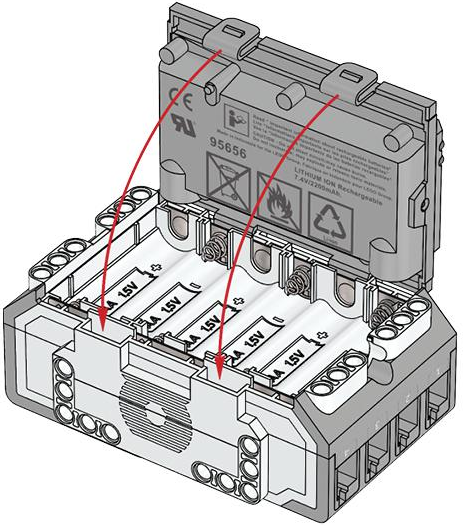
2x

34

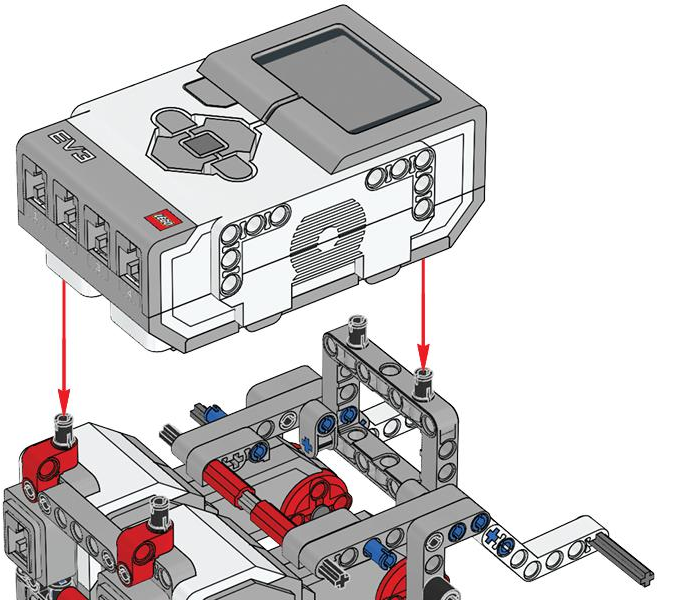


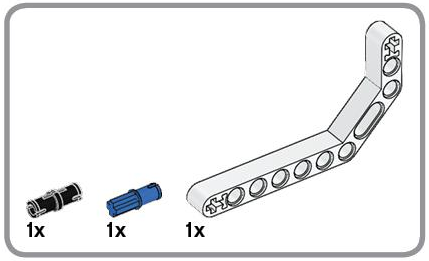


35

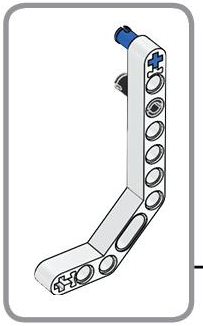


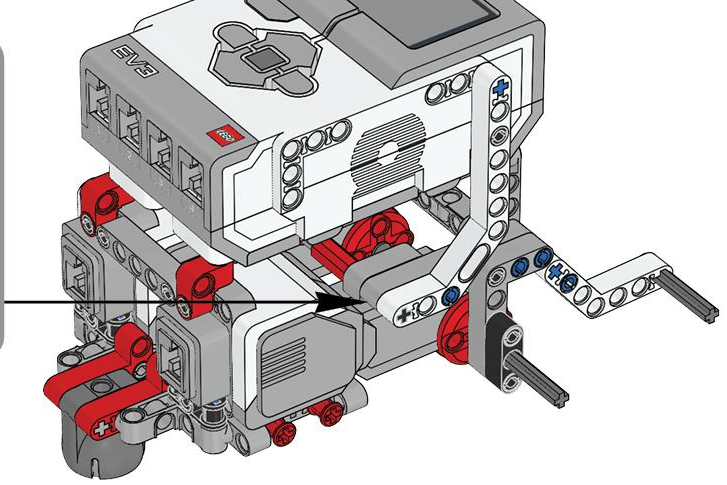
36

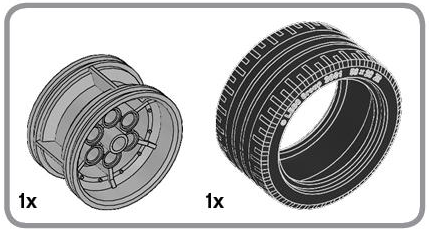




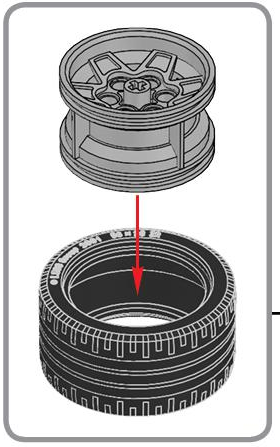
37

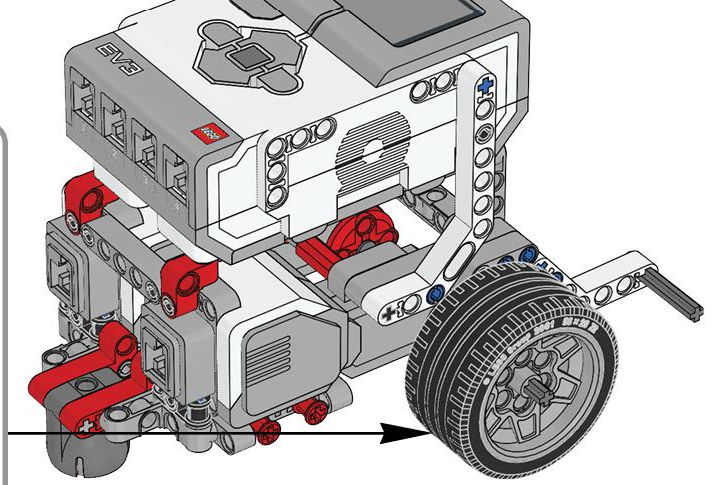






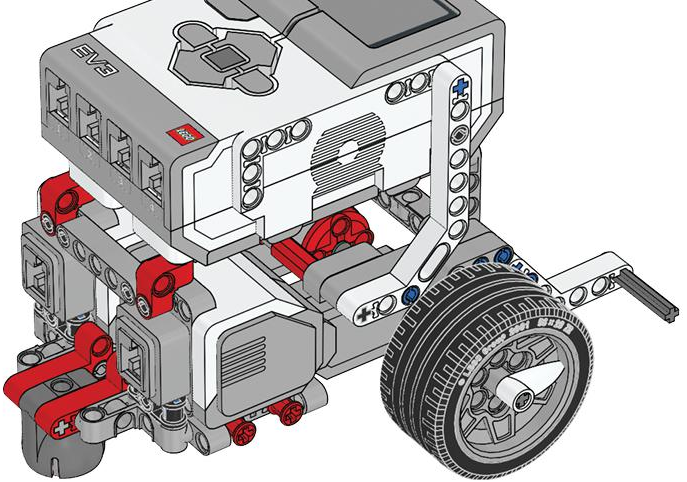
38

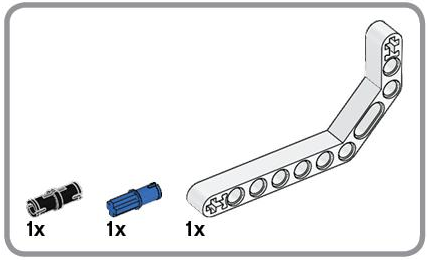




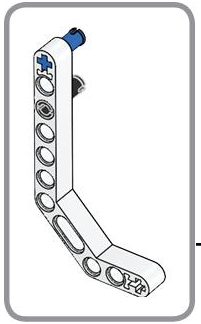
1x

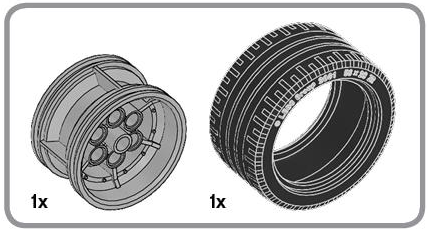
39



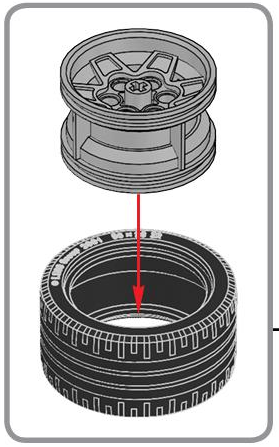


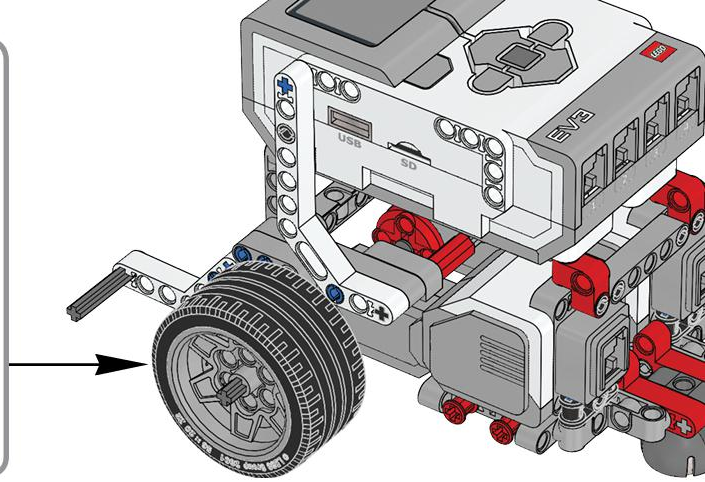
40





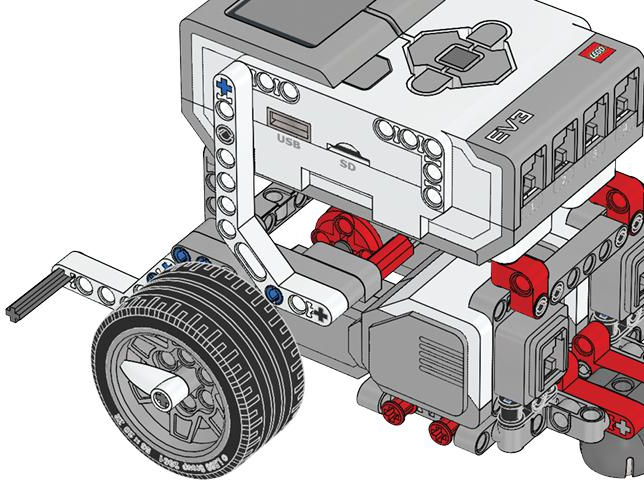
41

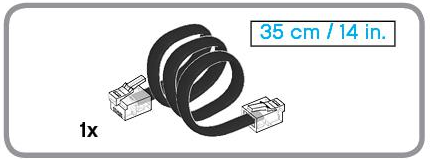




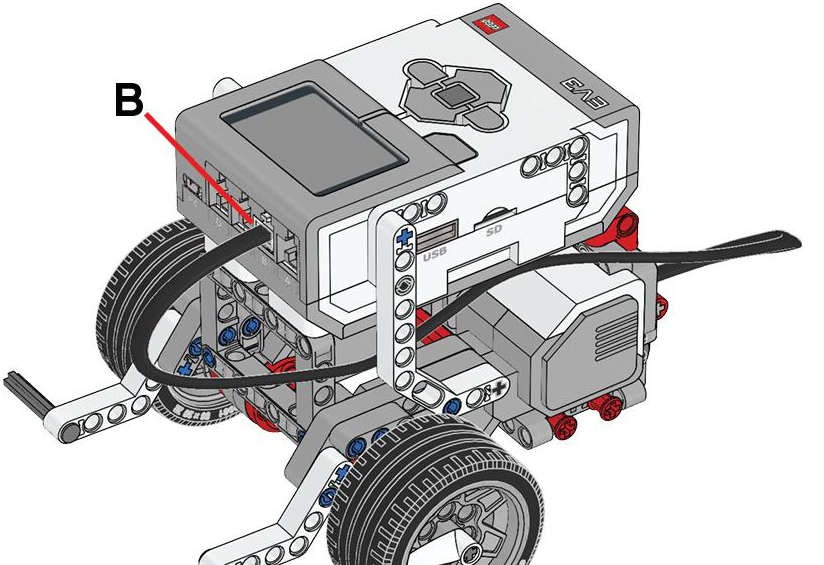
1x

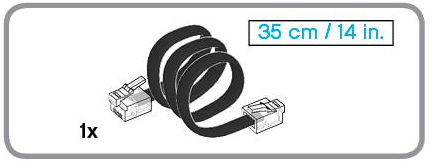
42



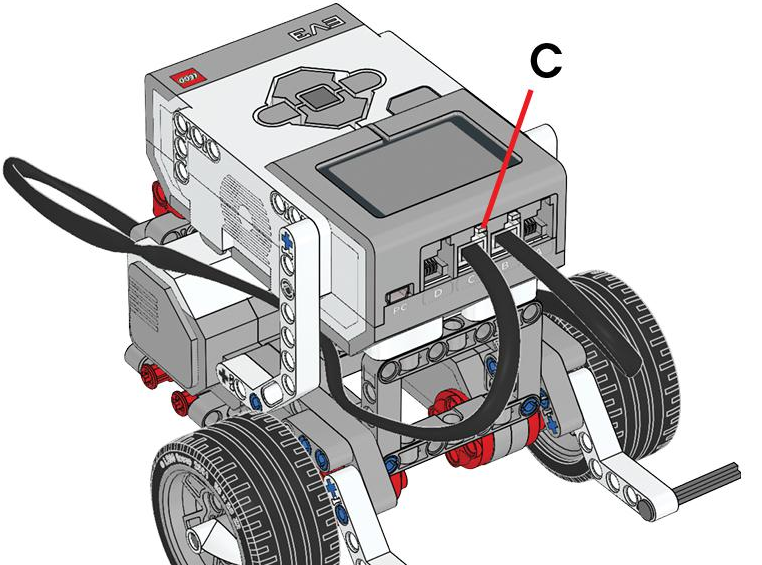


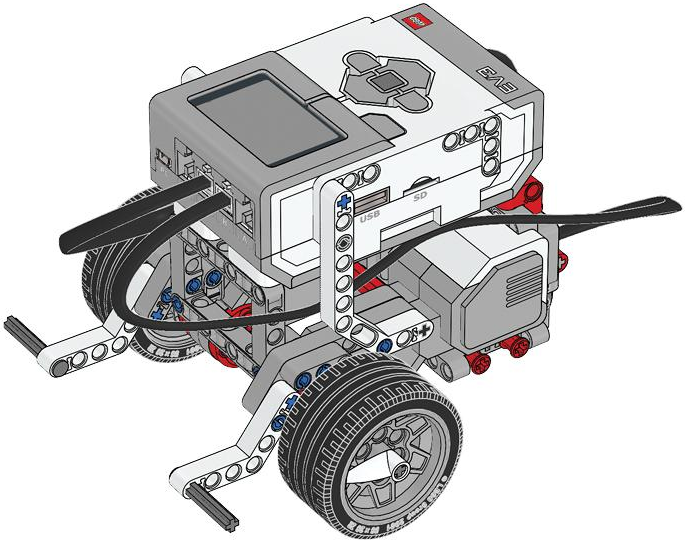
43





44




EV3
USB
SD