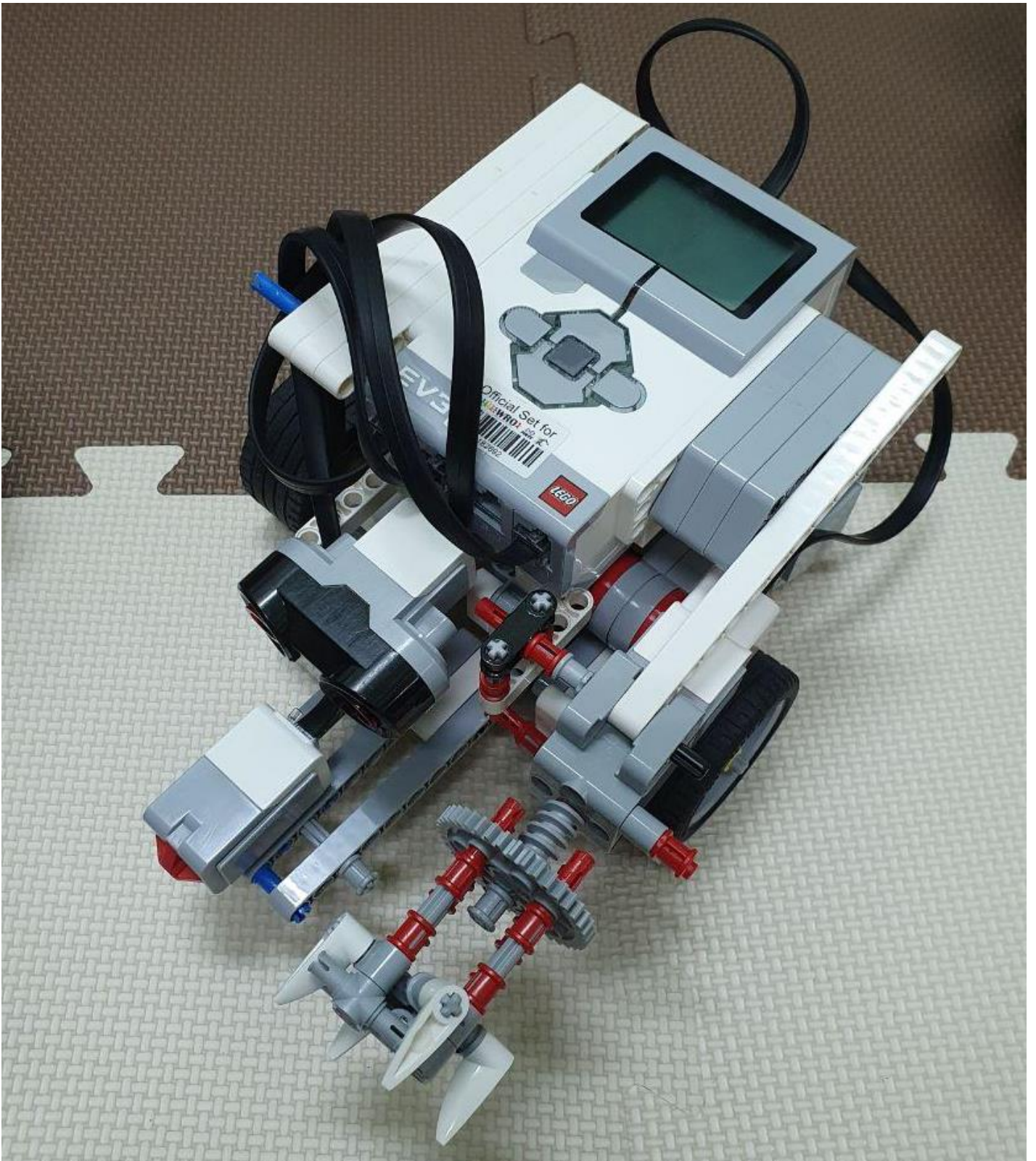


機器人成品：



## 設計理念

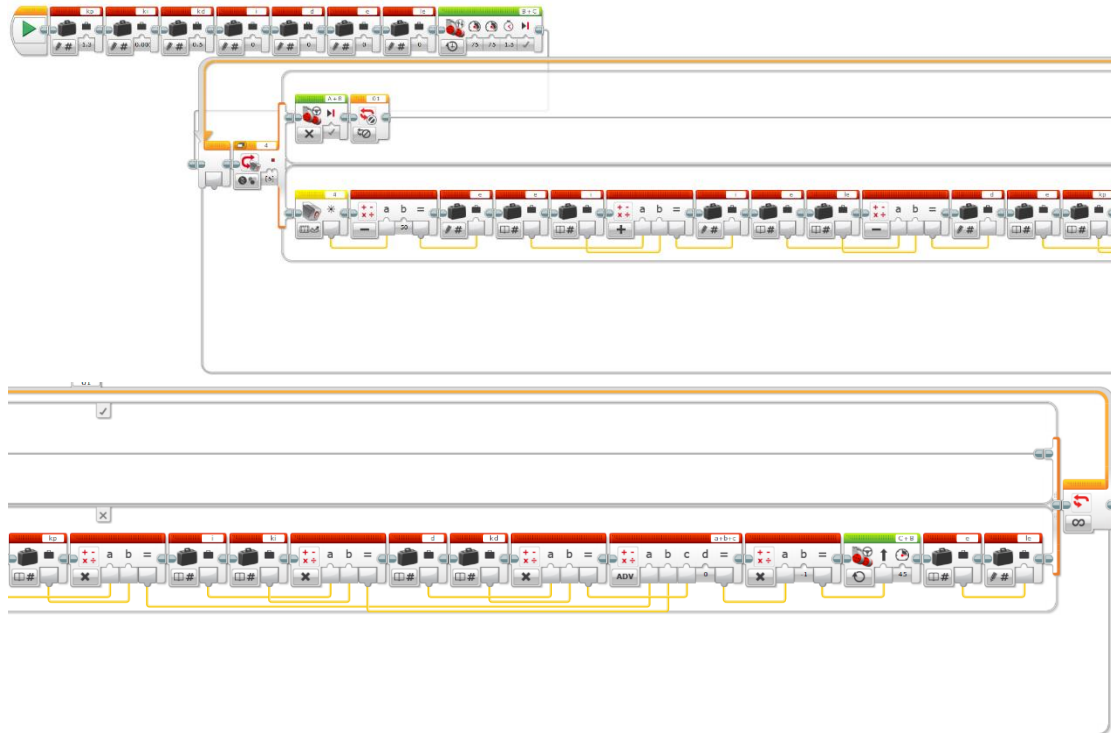
總體來說，我們的設計理念是先穩而後進。要達成此目的，先決條件是保持穩重，把重心放低，將基本需要的功能(超音波，按壓，光線感應器)做好固定，並且連線之間以沒有衝突為優先。保持好整體平衡和基本功能後，在做更多的設計，以利達成戰鬥和防禦的目的，最後採用重心偏後而機體前傾的設計

分開來看，先將機器人的底盤(輪子、馬達)組裝好，使用小齒輪扣住馬達，大齒輪扣住輪子增加力臂以加強推力，更有效率的施力；並且在底面後方加上滾球避免機器人往後倒。

我們在向前面延伸幾根積木並加上觸碰感應器，受到敵方的攻擊時，讓馬達加速反制對方，並且向前延伸的設計可以避免敵方直接攻擊輪子，更可用來攻擊。最後，在機器左方加上以中馬達製作持續旋轉的鑽頭，鑽頭前方設計成可以變形，可以卡住對手的結構再將對手翹起，或者是將對手的 sensor 弄歪，除了用以震懾對手外，更具有威脅。

## 圖形化程式說明

循跡：



易讀 Pseudocode:

```
kp = 1.3
ki = 0.001
kd = 0.5
e = i = d = le = 0
while True:
    if color is "red":
        break
    else:
        e = reflected_light_intensity - 50
        i += e
        d = e - le
        steering = kp * e + ki * i + kd * d
        le = e
# start combat ...
```

主要用到了 PID 控制系統。在 PID 系統中，P 代表比例，I 代表積分，以誤差累積表示，D 代表微分，以前後兩次誤差之差值表示。

先假設 color sensor 的光反射量之基準值為 50，代表這次循跡的底色與軌道顏色值的平均（一半黑一半白），再將 sensor 回傳的值減去 50 作為誤差  $e$ ，代表了 PID 中的 P，接著將每一次迴圈的  $e$  累加到誤差合 I 做累積。再將誤差  $e$  減去上一次迴圈的誤差  $le$ ，並儲存到誤差差值 D。

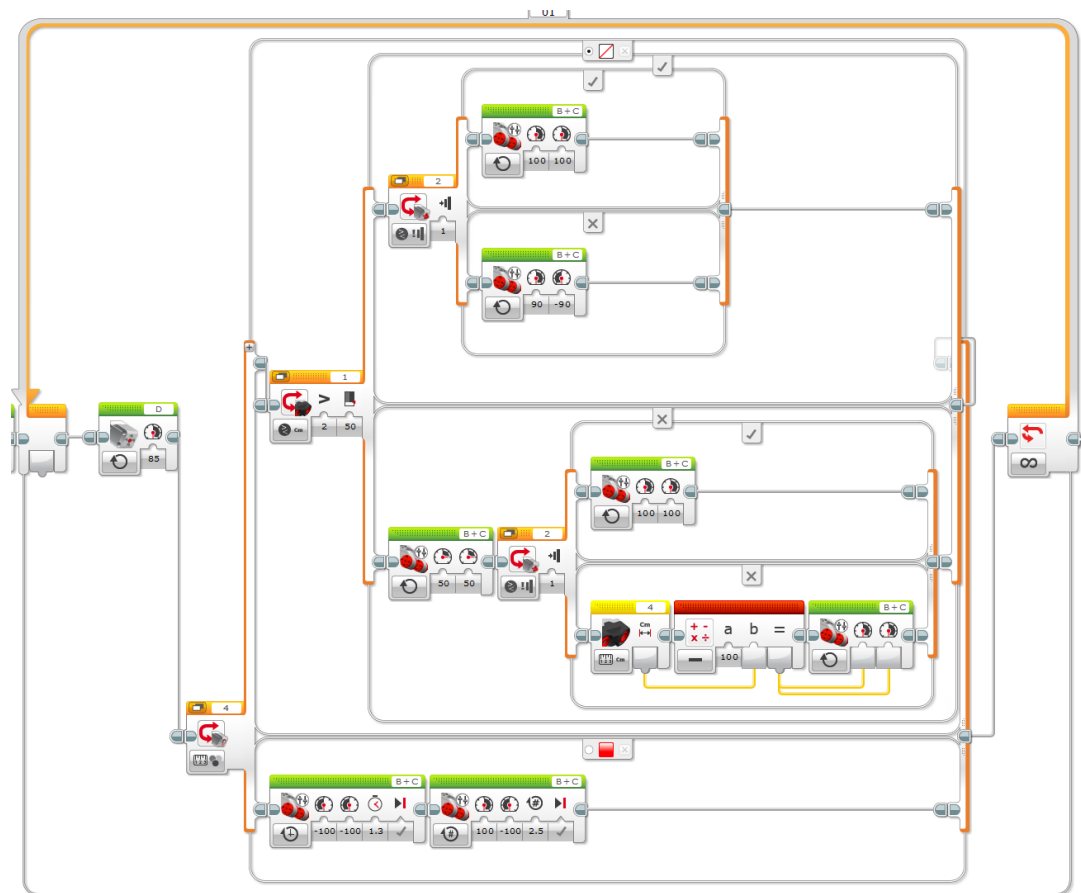
最後再將誤差  $e$  乘上  $K_p$ ，誤差合 I 乘上  $K_i$ ，誤差差值 D 乘上  $K_d$  等三個調整佔比的參數，再將三個值加總，就會得到總誤差（總調整量），接著將其值傳到馬達的轉彎方向值，就完成了一個簡單的 PID 控制系統，而三個值的代表意義分別如下：

$K_p$  值越大，每一次的修正幅度越大，其值過大時會使車子擺頭幅度很大，影響前進速度以及觀感。

$K_i$  最大的作用是在大幅度的轉彎處，因為誤差值累積越大，馬達的修正幅度也會變大，但若  $K_i$  值太大，反而會使平路的一點點誤差放大太多倍，便會開始原地旋轉。

$K_d$  代表的誤差差值可以在誤差有大量變化時就開始修正，讓轉彎更順利。

戰鬥：



鑽頭(port D)

進入紅圈後保持在 85 的轉速，不全速旋轉的原因是在實際測試時發現轉太快

反而不好卡住對手，因此將轉速調降一點

光線感應器(port 4)

每一次的迴圈都會偵測是否有感應到紅色（界線），若偵測到則後退，希望能保持不

出界，並在後退後轉向，正面迎擊對手

## 超音波感測器(port 1)

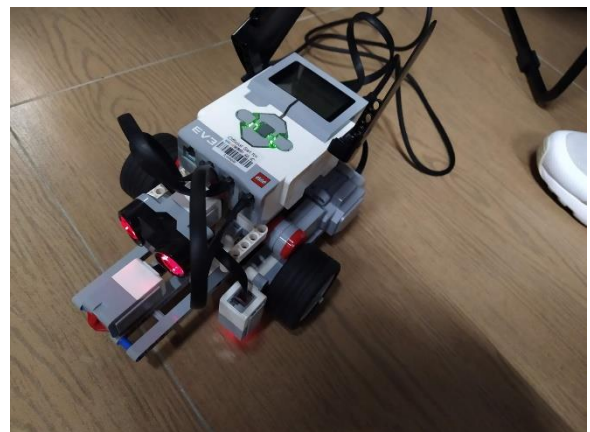
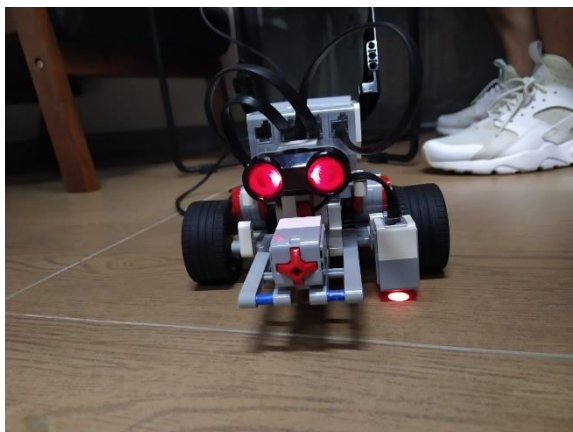
先偵測前方一定以內有沒有物體（測試時的重點），如果沒有的話，就原地右轉再次尋找，如果有物體則根據物體和機器的距離前進（具體為轉速 =  $100 - \text{距離 cm}$ ），途中如果發現跟丟對手則再次原地旋轉，調整方向，而遠距離時的低轉速可以避免因為誤差而不小心出界

## 觸碰感應器(port 2)

若前方的觸碰感應器感應到壓下，則代表接觸到對手本體了，因此將馬達轉速調整到最大，比拼馬力，可以彌補雖然撞到對手，但是歪掉導致超音波沒有偵測到的狀況

## 過程與成果

我們先將機器人組裝完成，並開始依據機器人的設計撰寫簡單程式測試行進的穩定性，此時希望能設計出一個穩固的機體，並未有設計其他功能





在程式設計的過程中，設計者逐漸發現可以加強和補全的功能與細節，完成了 PID 尋跡和基本的對手偵測等功能

最初版本的循跡：[影片](#)（有尖銳摩擦聲，建議靜音）

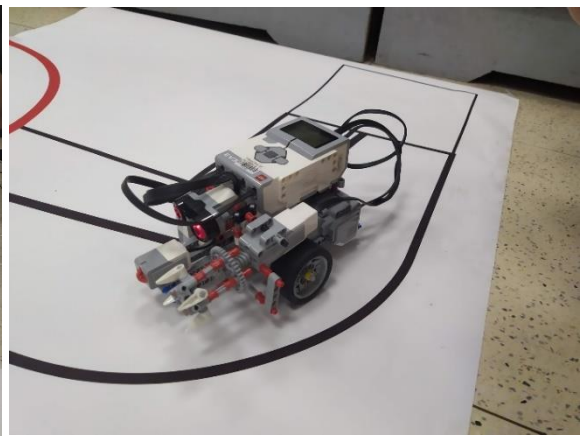
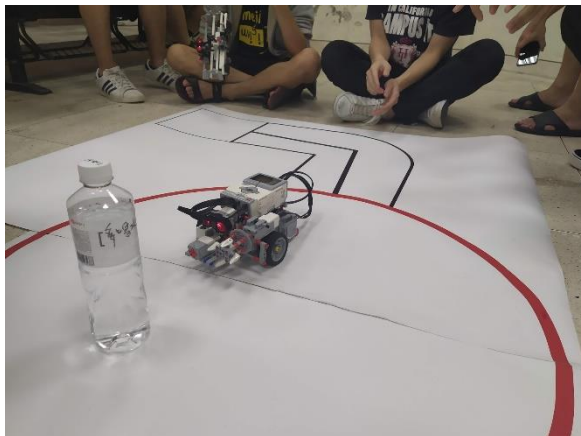


在測試時，到實際的場地進行測試，與其他組的機器人進行模擬戰鬥，也發生了一些沒有預想到的情況：

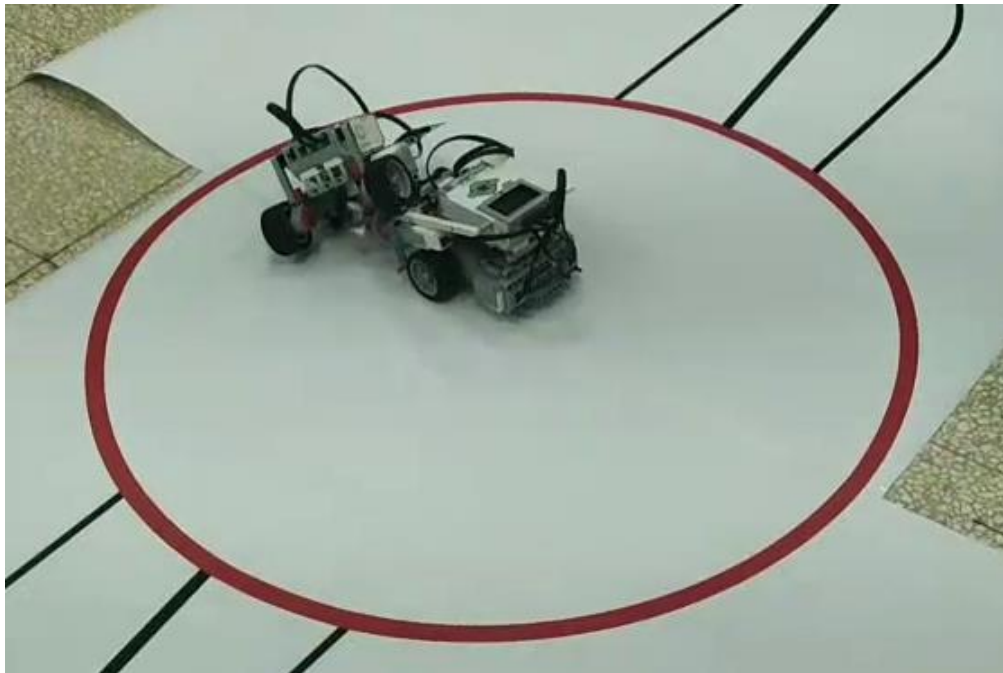
1.循跡時擺頭過於嚴重，調整  $k_p$  和  $k_i$  值使過程更順暢

以下為最終的參數值： $k_p = 1.3$ ， $k_i = 0.001$ ， $k_d = 0.5$

2.轉動尋找對手時轉速不足，容易被攻擊側面，調高速度差



最後為我們一致認為最精彩的一場比賽錄影：[影片](#)



## 心得

李勝維：

這次的作業讓我學習了很多，特別是在撰寫圖形化程式時，若沒有把優先順序整理好，很容易會出現預期以外的結果，而 sensor 的各種不準也讓我知道了實際測試的重要性，而這也是我第一次撰寫以 data flow 為本的程式語言，很多小地方都沒有接好，希望 HW3 時可以寫的更好。

林威宇：

很久沒有做這麼有趣的作業了。在設計相撲機器人的戰鬥程式時，我了解到就算寫出來的程式在螢幕上看起來沒什麼問題，也還是要設想到將其套用在硬體上的情況，並把現實中的各種條件列入考慮並進行調整，才算是好的程式。



符凱喻：

雖然我們這組沒有贏，但是我們的機器人有一個超屌的鑽頭在我心中就是第一了，而且差點把別人鑽到翻掉，可見不是只有蝦趴而已。最後感謝程式設計的組員在比賽前幾天一直調參數真是辛苦了。

李肇宸：

這份作業除了程式設計以外，還有組裝機器的部分，在大學的課程裡算是比較少見的，讓人有不同的體驗，也使作業不會乏味，最後的對抗賽也讓大家在驗收自己的成果時，也能與他人互動，非常有趣。

涂志瑋：

不論是組機器人，寫圖像化程式，都有種讓人回到童年時的感覺。在這之前，沒有想過多媒體能夠與此有所連結。在大學，學習許多課程卻不一定知道實用於何處。在這次和組員們互相討論、合作的經驗後，更能去思考自己學習的目的，確立未來的方向。

周孟謙：

這次的作業由於全部都要從零開始，一開始就要開始構想機器人的樣子然後慢慢將其組出，除了要組機器人外，我們還需要在規則內寫出成是讓機器人能進場並且搜敵對戰，並且加裝新的配件或是想出新戰術的過程都讓我覺得非常有趣。而最後的對戰也讓我們除了能與他人互動，也能驗收努力的成果，並且讓我對組裝機器人有了不一樣的體驗。