

# 機械實作期末報告

組員(學號+姓名):

110611108 蔡佑晟

110611070 沈宇韜

110611103 鄭庭安

110611054 黃品瑜

# 目錄

1. 設計構想
2. 機構設計方案
3. 機器人組合圖
4. 機構運動模擬
5. 電控設計
6. 組員分工表
7. 經費使用及報支情形
8. 機器人實作進度表
9. 遇到的問題
10. 實作、討論照片
11. 成員心得
12. 總結檢討

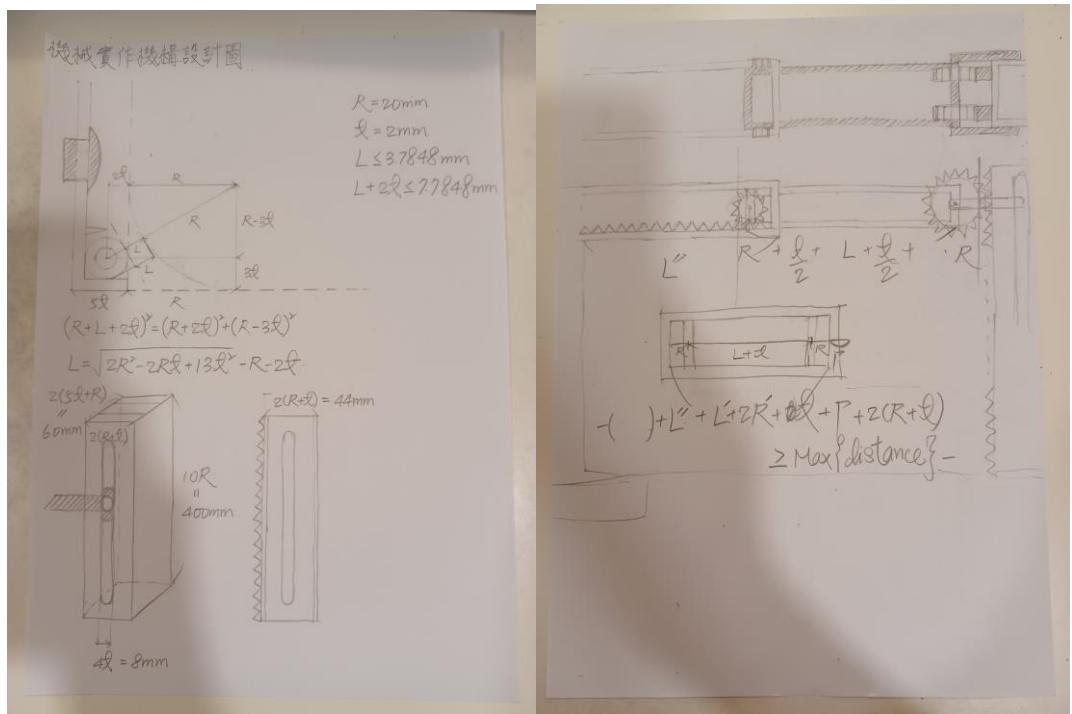
## 1. 設計構想

- 車體：密集板 or 壓克力板
- 車輪：玩具車輪(66mm)
- 取物：夾取後放入箱子(最終版為單向通行機構取物)
- 置物：箱子後方的門打開，使球滾落(最終版為桶子另一端設置閥門，使球滾落)
- 控制：芽控
- 驅動：四輪驅動

## 2. 機構設計方案

2.1 移動機構：減速馬達+四個玩具車輪+馬達驅動模組\*2，轉彎實現為左右輪相反轉向

2.2 取物機構：夾爪設計初稿(後來因加工困難部分並未採用)，之後使用桶子+單向通行機構實現



Case 2. 收納主最短

$$L_1 + d = L_2 + 2d + 2R' - l \quad (\text{不考慮牙齒公差})$$

外部 因部  
組件 組件

$$\rightarrow L_1 + d = L_2 + 2d + 2R' + l - T \quad (\text{不考慮公差})$$

T須露出來才不會讓與之接合之夾球箱之齒條  
卡在出口

$$\rightarrow L_1 - L_2 - 2R' - T - l = 0$$

結論

$$\begin{cases} L_1 + L_2 = \text{Max}\{ \dots \} - 2R' - 2T - \frac{3}{2}l + T' \\ L_1 - L_2 = -2R' + T + l \end{cases} \xrightarrow{-R'} \begin{cases} L_1 = \frac{1}{2}(\text{Max}\{ \dots \} - T - \frac{3}{2}l + T' - R') \\ L_2 = \frac{1}{2}(\text{Max}\{ \dots \} - 4R' - 3T - \frac{3}{2}l + T - R') \end{cases}$$

①我們之前令 $l = 2\text{mm}$ , 所以要再決定 $\text{Max}\{ \dots \}$ (決定擺放位置),  $T$  &  $T'$ (齒輪種類)

③公差項

② $T$  &  $T'$ 在初步設計可先不必考慮)

假設齒輪沒有長牙齒及公差

令齒輪半徑 $R'$ , 所有收納組件厚度 $\ell$ , 內部長度分別為 $L_1, L_2$

Case 1. 拉伸至最長 → 齒輪中心固定在厚度中間

$$[(L_1 + \ell) + (L_2 + 2\ell + 2R' - \ell)] - (R + \frac{\ell}{2}) = \text{前臂最大長度(不考慮牙齒及公差)}$$

→ 外部長度 因內部齒輪抵住零件

內牆損失的長度

$$\rightarrow [(L_1 + \ell) + (L_2 + \ell + 2R' + 2P)] - (R + \frac{\ell}{2} + P) = L_1 + L_2 + 2R' - R + \frac{3}{2}\ell + P \quad \text{公差}$$

= 前臂最大長度(考慮牙齒高度, 不考慮)

夾到最遠的球 → (前臂最大長度) + (基座半徑) + (夾球箱寬度的一半) +  $P - P'$   
整個前臂最大長度 整個夾球長度 ↑  
≥ (球至基座中心之最遠距離) 跟基座位置 1  
牙齒重疊長度

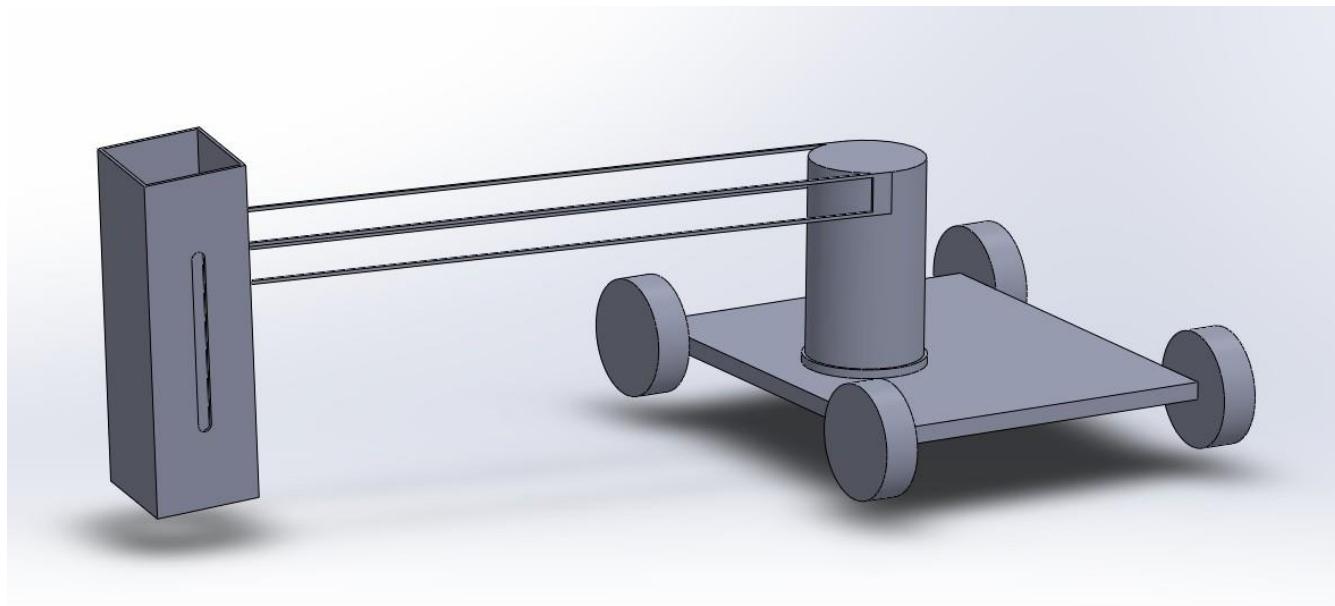
$$\rightarrow (L_1 + L_2 + 2R' - R + \frac{3}{2}\ell + P) + R'' + \frac{1}{2}(2(R + \ell)) + P - P' \geq \text{Max}\{distance\}$$

有關

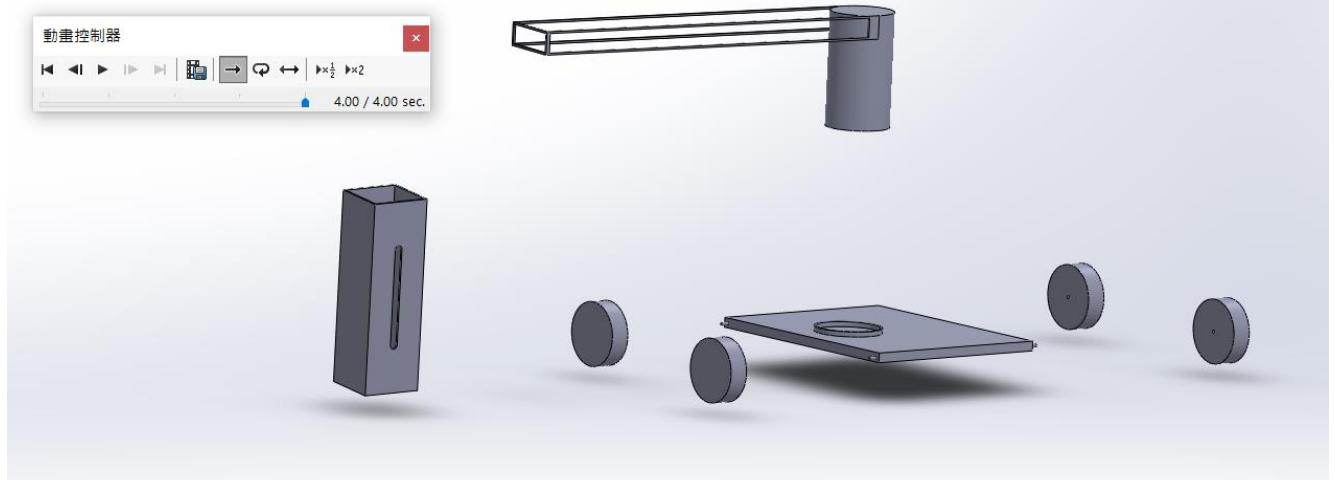
$$= L_1 + L_2 + 2(R + P) + \frac{5}{2}\ell - P' \geq \text{Max}\{distance\} - R''$$

### 3. 機器人組合圖

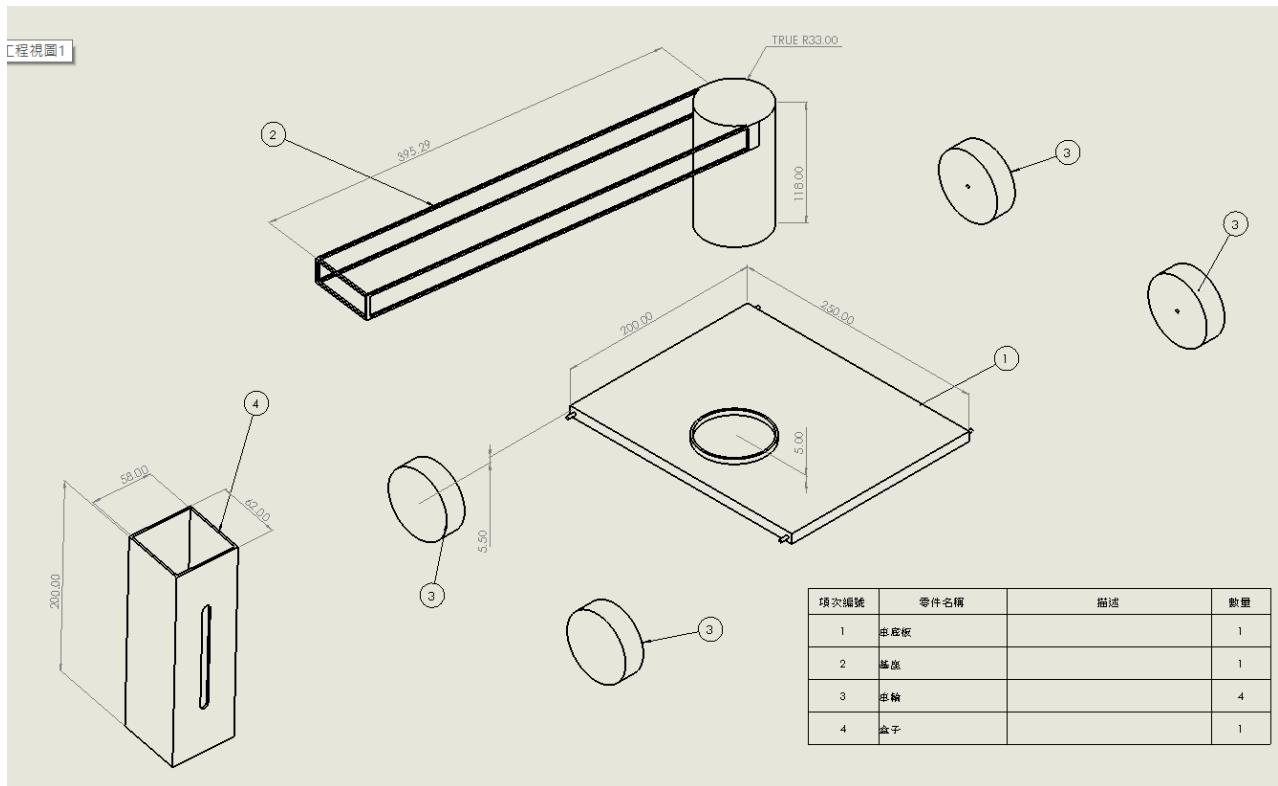
#### 3.1 組合圖



3.2 爆炸圖



3.3 組合件工程圖



### 3.4 材料

編號	品名	備註	數量
1	密集板	25cm*20cm*9mm	1
2	木板		1
3	海棉板		1
4	玩具車輪	66mm	4
5	減速直流馬達		4
6	L298N電機驅動板模組		2
7	HM-10藍芽模組		2
8	3號電池		3
9	電池盒		2
10	麵包板(小)		1
11	杜邦線(公)	20cm	10
12	杜邦線(母)	20cm	10
13	充電電池18650		3
14	油瓶	墊高用	2
15	SG伺服馬達		1
16	Arduino mega板		2
17	不鏽鋼吸管		2
18	伺服馬達MG996		6
19	鋰電池		2

## 5.電控設計

### 5.1 馬達規格

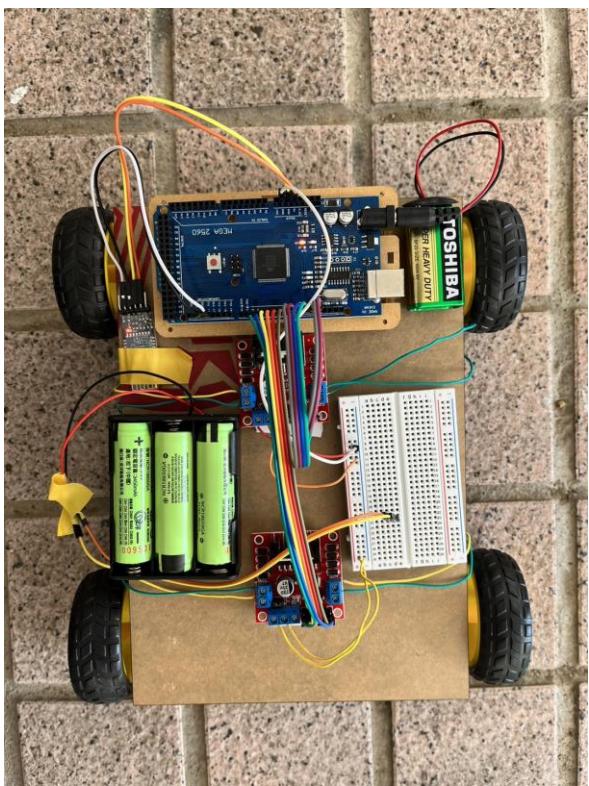
TT 減速馬達:直流電 3~6V、空載電流 160mA 齒輪比 1:48、200rpm

MG996R:直流電 4.8~7.2V、空載電流 120mA、扭矩  
9kg\*cm(4.8V)11kg\*cm(6.0V)

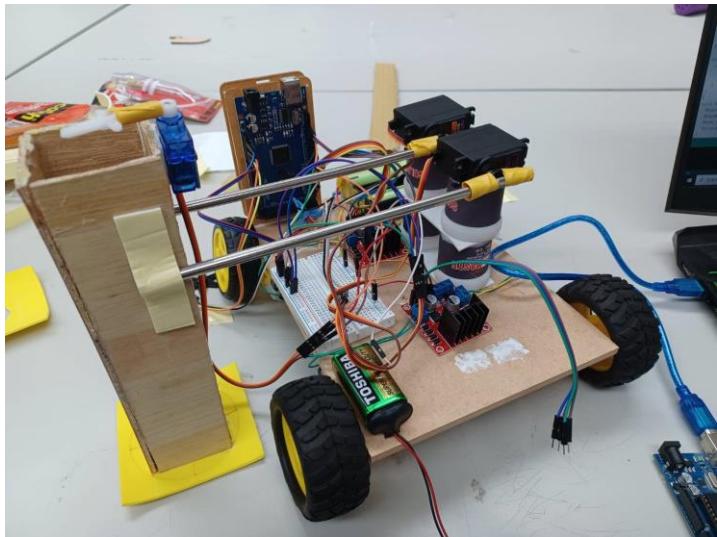
SG90 伺服馬達: 直流電 4.8V、扭矩 1.8kg\*cm(4.8V)

### 5.2 電控系統選用&控制元件

選用 arduino，驅動車輪需要馬達驅動模組，故使用了兩個 L298N(一個 L298N 驅動兩個馬達)，SG90 與 MG996R 能直接用 arduino 控制，不須再接去驅動模組，藍芽模組有很多選擇，由於我們的遙控介面使用 Dabble app，因為 android\ios 皆可用，且有一些 github 上的範例可以供我們參考，熟悉介面操作方式，Dabble 只支援某些藍芽模組，我們選擇 HM-10、BT05，兩個 arduino mega 板，分別燒錄控制車輪及爪子（因 pwd pin 數，一個板子不夠）



車輪部分



夾爪部分

在夾爪的部分，我們選用 2 種不同的伺服馬達，一種是藍色外觀的 SG90，另一種是體積較大的黑色 MG996R。這樣的選用邏輯是因為我們取物的機制是用木盒子往下蓋乒乓球，所以需要一定的力道升降盒子，而 MG996R 可以提供最大 11kg 的扭力，因此我們選用兩個 MG996R 來進行操作。

至於藍色的 SG90，則置於木盒子最上方，當要放球時，再旋轉馬達的白色塑膠片，讓乒乓球從盒子滾落到遞件區。

### 5.3 程式架構

車輪部分：

```

#define CUSTOM_SETTINGS
#define INCLUDE_GAMEPAD_MODULE
#include <Dabble.h>
const unsigned long BAUD_RATE = 9600;
int FRONT_ENA_value = 200;
bool circlePressedPrev = false;
bool trianglePressedPrev = false;
//TODO:add LED indicators on car
// SET circle to accelerate, triangle to decelerate

#define MOTOR_FRONT_LEFT_FORWARD 6
#define MOTOR_FRONT_LEFT_BACKWARD 7
#define MOTOR_FRONT_RIGHT_FORWARD 4
#define MOTOR_FRONT_RIGHT_BACKWARD 5
#define MOTOR_REAR_LEFT_FORWARD 10
#define MOTOR_REAR_LEFT_BACKWARD 11
#define MOTOR_REAR_RIGHT_FORWARD 8
#define MOTOR_REAR_RIGHT_BACKWARD 9

#define FRONT_ENA 2
#define FRONT_ENB 3
#define REAR_ENA 12
#define REAR_ENB 13

```

一開始先定義好 pin 腳位，四個輪子的向前向後分別定義，前後 ENA、ENB 用來控制轉速（若不加，則輪子會以最大轉速 255rpm 轉動）而太快，藍牙遙控時，按下介面按鈕的時間不好控制，因此決定訂為 200(因四輪轉速一樣，因此僅定義 FRONT\_ENA\_VALUE，原先有加輪子加減速的程式，但後來決定固定轉速就好，沒用到。

```

void setup() {
    pinMode(MOTOR_FRONT_LEFT_FORWARD, OUTPUT);
    pinMode(MOTOR_FRONT_LEFT_BACKWARD, OUTPUT);
    pinMode(MOTOR_FRONT_RIGHT_FORWARD, OUTPUT);
    pinMode(MOTOR_FRONT_RIGHT_BACKWARD, OUTPUT);
    pinMode(MOTOR_REAR_LEFT_FORWARD, OUTPUT);
    pinMode(MOTOR_REAR_LEFT_BACKWARD, OUTPUT);
    pinMode(MOTOR_REAR_RIGHT_FORWARD, OUTPUT);
    pinMode(MOTOR_REAR_RIGHT_BACKWARD, OUTPUT);
    pinMode(FRONT_ENA, OUTPUT);
    pinMode(FRONT_ENB, OUTPUT);
    pinMode(REAR_ENA, OUTPUT);
    pinMode(REAR_ENB, OUTPUT);
    Serial.begin(BAUD_RATE); //usb
    Serial3.begin(BAUD_RATE); //Serial3 for communication with HM-10,RX 14 TX 15
    Dabble.begin(BAUD_RATE, 14, 15);
    Serial.println("setup complete");
}
...

```

接著定義 pinMode 為輸出和初始化 serial 和 serial 3（用來硬體連接 HM-10 藍牙模組，一開始原本用 serial 1 的硬體來連接，但後一直連不上，後來到github 查 example，發現 Dabble app 的藍芽連線在 arduino mega board 上要用 serial 3，因此後來便順利連接上。

`Serial.println()` 這行，是一開始除錯時，用來配合 arduino serial monitor 看是否完成 setup 的。

```
void loop(){
    Dabble.processInput();

    analogWrite(FRONT_ENA, FRONT_ENA_value); //>70 to move and little buzzing
    analogWrite(FRONT_ENB, FRONT_ENA_value); //start speed: 100, ENA speed: 70--255
    analogWrite(REAR_ENA, FRONT_ENA_value);
    analogWrite(REAR_ENB, FRONT_ENA_value);

    if (GamePad.isUpPressed())
    {
        moveForward();
    }

    else if (GamePad.isDownPressed())
    {
        moveBackward();
    }

    else if (GamePad.isLeftPressed())
    {
        turnLeft();
    }

    else if (GamePad.isRightPressed())
    {
        turnRight();
    }
    else
    {
        analogWrite(FRONT_ENA, 0);
        analogWrite(FRONT_ENB, 0);
        analogWrite(REAR_ENA, 0);
        analogWrite(REAR_ENB, 0);
        stopCar();
    }
}
```

接著 `loop()` 內部，`Dabble.processInput()` 先初始化 `dabble` 介面，接著設定轉速，發現大概  $>=70\text{rpm}$  時，即能去除轉速太慢的 buzzing sound，但後來發現轉速若太慢，則輪子摩擦力不足，造成無法順利左右轉，因此最終轉速定為  $200\text{rpm}$ 。

在 [github](#) 上，查詢標頭檔，看 `dabble app` 定義好的函數如何應用（和 [Youtube](#) 範例）`isDownPressed()` 是其中定義好的函數，`GamePad` 是宣告好的全域物件，可以呼叫其 `method`，而 `moveForward()` 等，五個函數（前後左右停），為後方自行定義的。

```

void moveForward() {
    digitalWrite(MOTOR_FRONT_LEFT_FORWARD, HIGH);
    digitalWrite(MOTOR_FRONT_LEFT_BACKWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_FORWARD, HIGH);
    digitalWrite(MOTOR_FRONT_RIGHT_BACKWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_FORWARD, HIGH);
    digitalWrite(MOTOR_REAR_LEFT_BACKWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_FORWARD, HIGH);
    digitalWrite(MOTOR_REAR_RIGHT_BACKWARD, LOW);
}
void moveBackward() {
    digitalWrite(MOTOR_FRONT_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_LEFT_BACKWARD, HIGH);
    digitalWrite(MOTOR_FRONT_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_BACKWARD, HIGH);
    digitalWrite(MOTOR_REAR_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_BACKWARD, HIGH);
    digitalWrite(MOTOR_REAR_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_BACKWARD, HIGH);
}
void turnLeft() {
    digitalWrite(MOTOR_FRONT_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_LEFT_BACKWARD, HIGH);
    digitalWrite(MOTOR_FRONT_RIGHT_FORWARD, HIGH);
    digitalWrite(MOTOR_FRONT_RIGHT_BACKWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_BACKWARD, HIGH);
    digitalWrite(MOTOR_REAR_RIGHT_FORWARD, HIGH);
    digitalWrite(MOTOR_REAR_RIGHT_BACKWARD, LOW);
}
void turnRight() {
    digitalWrite(MOTOR_FRONT_LEFT_FORWARD, HIGH);
    digitalWrite(MOTOR_FRONT_LEFT_BACKWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_BACKWARD, HIGH);
    digitalWrite(MOTOR_REAR_LEFT_FORWARD, HIGH);
    digitalWrite(MOTOR_REAR_LEFT_BACKWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_BACKWARD, HIGH);
}

```

向前：四個輪子的 Forward pin 設為 high，Backward pin 設為 low

向後：四個輪子的 Forward pin 設為 low，Backward pin 設為 high

向左：右輪向前，左輪向後

向右：左輪向前，右輪向後

```

void stopCar() {
    digitalWrite(MOTOR_FRONT_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_LEFT_BACKWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_BACKWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_BACKWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_BACKWARD, LOW);
}

```

停車：all pin low

```

if (GamePad.isCirclePressed() && !circlePressedPrev) {
    motorAccel();
    circlePressedPrev = true;
} else if (!GamePad.isCirclePressed()) {
    circlePressedPrev = false;
}

if (GamePad.isTrianglePressed() && !trianglePressedPrev) {
    motorDecel();
    trianglePressedPrev = true;
} else if (!GamePad.isTrianglePressed()) {
    trianglePressedPrev = false;
}

}

void motorAccel() {
    if (FRONT_ENA_value < 256) {
        FRONT_ENA_value += 10;
        updateMotorSpeed();
    }
}

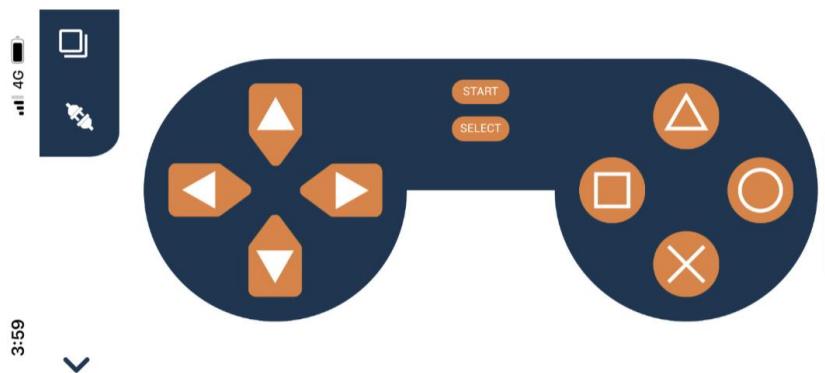
void motorDecel() {
    if (FRONT_ENA_value < 256) {
        FRONT_ENA_value -= 10;
        updateMotorSpeed();
    }
}

void updateMotorSpeed() {

    analogWrite(FRONT_ENA, FRONT_ENA_value);
    analogWrite(FRONT_ENB, FRONT_ENA_value);
    analogWrite(REAR_ENA, FRONT_ENA_value);
    analogWrite(REAR_ENB, FRONT_ENA_value);
}

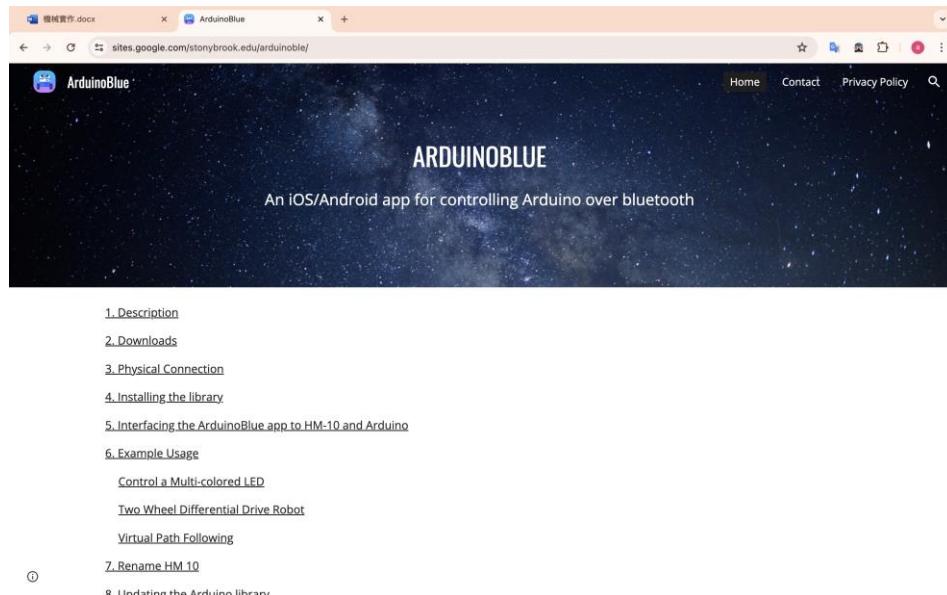
```

加減速的部分：原本是設定介面上的圓、三角型為加減速按鈕，後來沒用到。



而選擇操作介面時，一開始選擇 stony brook university 的 arduinoble app，但

後來因學校開發的軟體雖然免費，但因維護較少，仍有一些 Bug，且因較新，範例較不足，因此遇到問題時，沒有查找的資源，所以後來改成 dabble app(ios\android)手機皆可用，且範例多，可以參考。



## 夾爪部分：

```
#define CUSTOM_SETTINGS
#define INCLUDE_GAMEPAD_MODULE
#include <Dabble.h>
#include <Servo.h>

const unsigned long BAUD_RATE = 9600;

#define BLACK_MOTOR1_Pin 10
#define BLACK_MOTOR2_Pin 9
#define BLUE_MOTOR_Pin 8

Servo Black1_Servo;
Servo Black2_Servo;
Servo Blue_Servo;
```

一開始先宣告 3 根 pin 角(8,9,10)當作 3 個馬達的訊號傳輸角位，再分別宣告 3 個 Servo 類別的物件，代表 3 個伺服馬達。

```
void setup() {
    pinMode(BLACK_MOTOR1_Pin, OUTPUT);
    pinMode(BLACK_MOTOR2_Pin, OUTPUT);
    pinMode(BLUE_MOTOR_Pin, OUTPUT);

    Black1_Servo.attach(BLACK_MOTOR1_Pin);
    Black2_Servo.attach(BLACK_MOTOR2_Pin);
    Blue_Servo.attach(BLUE_MOTOR_Pin);

    Serial.begin(BAUD_RATE); // USB
    Serial3.begin(BAUD_RATE); // Serial3 用於與 HM-10 通訊，RX 14 TX 15
    Dabble.begin(BAUD_RATE, 14, 15);
}
```

接下來這部分是將 3 根 pin 角設為 output 端，再使用 Servo 類別裡的方法 :attach，將 3 根 pin 角與伺服馬達接。

最後設定藍芽模組的 baud rate。

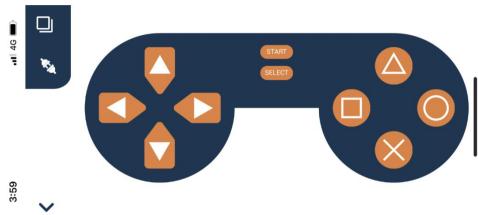
```
void loop() {
    Dabble.processInput();

    if (GamePad.isCirclePressed()) { // 黑色馬達順時針旋轉
        stop = 0;
        Black_CW(stop);
    } else if (GamePad.isTrianglePressed()) { // 黑色馬達逆時針旋轉
        stop = 0;
        Black_CCW(stop);
    } else if (GamePad.isSquarePressed()) { // 藍色馬達
        stop = 0;
        Blue(stop);
    } else {
        stop = 1;
        Stop();
    }
}
```

在 loop 內是程式的主要邏輯

如果按下介面上的圓形，兩顆 MG996R 將會順時針轉動，舉升木盒，反之按下三角形，則逆時針轉動，下降木盒。

按下方形按鍵則是轉動 MG90，控制白色塑膠片的打開或關上。



```
void Black_CW(int stop) {
    if (stop == 0) {
        Black1_Servo.write(105);
        Black2_Servo.write(105);
        delay(15);
    }
}

void Black_CCW(int stop) {
    if (stop == 0) {
        Black1_Servo.write(85);
        Black2_Servo.write(85);
        delay(15);
    }
}

void Blue(int stop) {
    if (stop == 0) {
        Blue_Servo.write(60);
        delay(15);
    }
}
```

這部分是函式定義，我們使用的伺服馬達， $0\sim 90^\circ$  是逆時針旋轉，並且度數越小轉速越快。而  $90\sim 180^\circ$  是順時針旋轉，度數越大轉速越快。

```
void Stop() {  
    Black1_Servo.write(90);  
    Black2_Servo.write(90);  
    Blue_Servo.write(90);  
    delay(15);  
}
```

最後，如果介面上的按鍵都沒有被按下，STOP 函式就會被呼叫，將 3 個伺服馬達的角度設為 90 度，這是經過我們實驗發現的，當 90 度時，馬達會停止轉動。

## 6.組員分工表

	<p>110611070 沈宇韜</p>	<p>1.大部分材料選購 2.構想主要提案人 3.車輪電控 4.構想提案報告 5.車子本體組裝與電</p>
---	--------------------------	---

		路接線 6.車子操控員
	110611103 鄭庭安	<p>1. 部分材料選購</p> <p>2. 車輪程式（前後左右、控制速度）</p> <p>3. 選擇合適的程式操作介面 app(測試與手機、藍芽的相容性)</p> <p>4. 車輪、車輪藍芽的接、配線</p>
	110611054 黃品瑜	<p>1. 機構設計(原理與規格計算)</p> <p>2. 車體與夾爪繪圖</p> <p>3. 夾爪材料選購</p> <p>4. 車體組裝</p> <p>5. 夾爪程式(檢查與測試，非主要設計)</p>

	<b>110611108</b> <b>蔡佑晟</b>	1.夾爪程式  (主要設計)  2.木板裁切加工  3.車體與夾爪繪圖  4.夾爪組裝、配線
---	--------------------------------	--

## 7. 經費使用與報支情形

項次	發票號碼/收據號碼/收據日期	品名	數量	單價	小計(元)
1	ZB88635823	113/4/11 72mm 車輪 紅色 金屬光	4	75	300
		服務手續費	1	60	60
2	18879139	113/3/25 雷射加工壓克力	1	619	619
3	YA34293381	113/3/20 HM-10 藍芽4.0模組	1	220	220
4	ZB45921786	113/3/14 泡棉膠 3M 113 18mm*5M	1	32	32
5	13258875	113/3/19 3號電池	1	169	169
6	ZJ46400992	113/3/20 R3 附USB線 升級版CH3406 : A套餐(主板+傳輸)	1	660	660
7	ZK13642155	113/4/2 [創客市集] 微型130馬達	6	9	54
		[創客市集] 微型130馬達	6	9	54
		[創客市集]Arduino 塑膠直流1:4 8減速馬達	3	15	45
		物流費	1	60	60
8	ZK01083024	113/3/5 [創客市集] SG90伺服馬達	4	32	128
		[創客市集] 碳鋅電池 電池盒 帶蓋子 附開關	2	15	30
		[創客市集] L298N電機驅動板模組	3	40	120
		[創客市集]彩巴邦尼線 每對半兩	1	8	8
		[創客市集]0公分長1單位10條	4	15	60
		[創客市集]彩巴邦尼線 公對公2	1	12	12
		[創客市集]0公分長1單位10條	6	12	72
		配件 塑膠輪胎 輪子	1	60	60
9	ZH92004586	113/3/29 18650電池充電器	1	150	150
		3M電工膠帶 1500	1	35	35
		9V東芝環保電池	1	42	42
		EPROM 2Mb 100nS	1	140	140
		充電鋰電池(平頭) 無保護板 3.6V/3300-345	3	260	780
10	ZH92004441	113/3/28 9V東芝環保電池	1	42	42
		3號4只電池盒方形	1	55	55
		帶DC頭 T型9V電池扣	1	45	45
		40P彩色杜邦線 公對母	1	130	130
		40P彩色杜邦線 公對公	1	130	130
11	ZX13227943	113/5/14 MG996R伺服馬達	2	340	680
		合計			4992

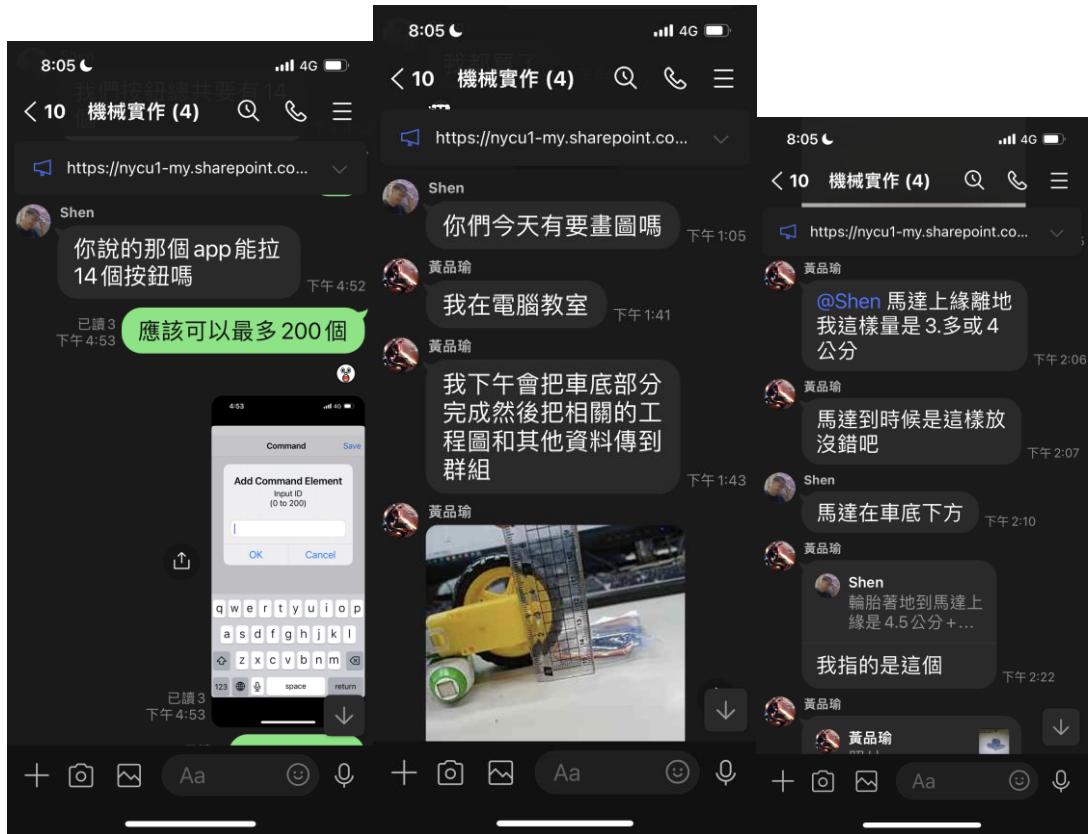
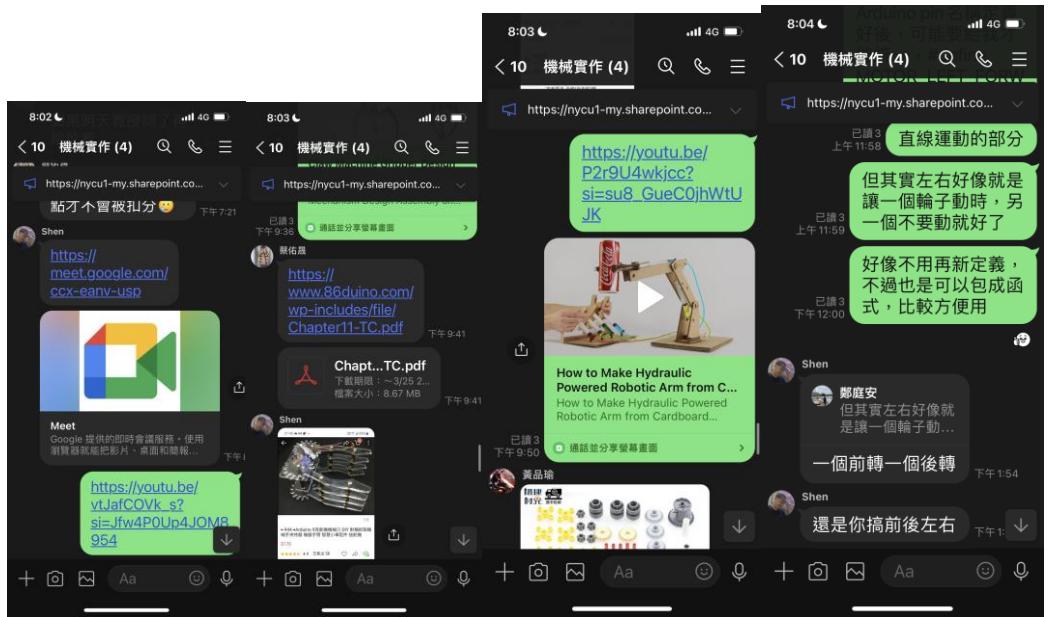
上面全材料均已報帳

## 8.機器人實作進度表

<p>“0”為實際進度 藍底為預計進度 若有斷層為前為車子進度，後為取物機構進度</p>	第1周	第2周	第3周	第4周	第5周	第6周	第7周	第8周	第9周	第10周	第11周	第12周	第13周	第14周	第15周	第16周
	0	0	0													
			0	0	0											
				0	0											
					0	0										
						0										
							0									
								0								
									0							
										0						
											0					
												0				
													0			
														0		
															0	
																0

## 9.遇到的問題

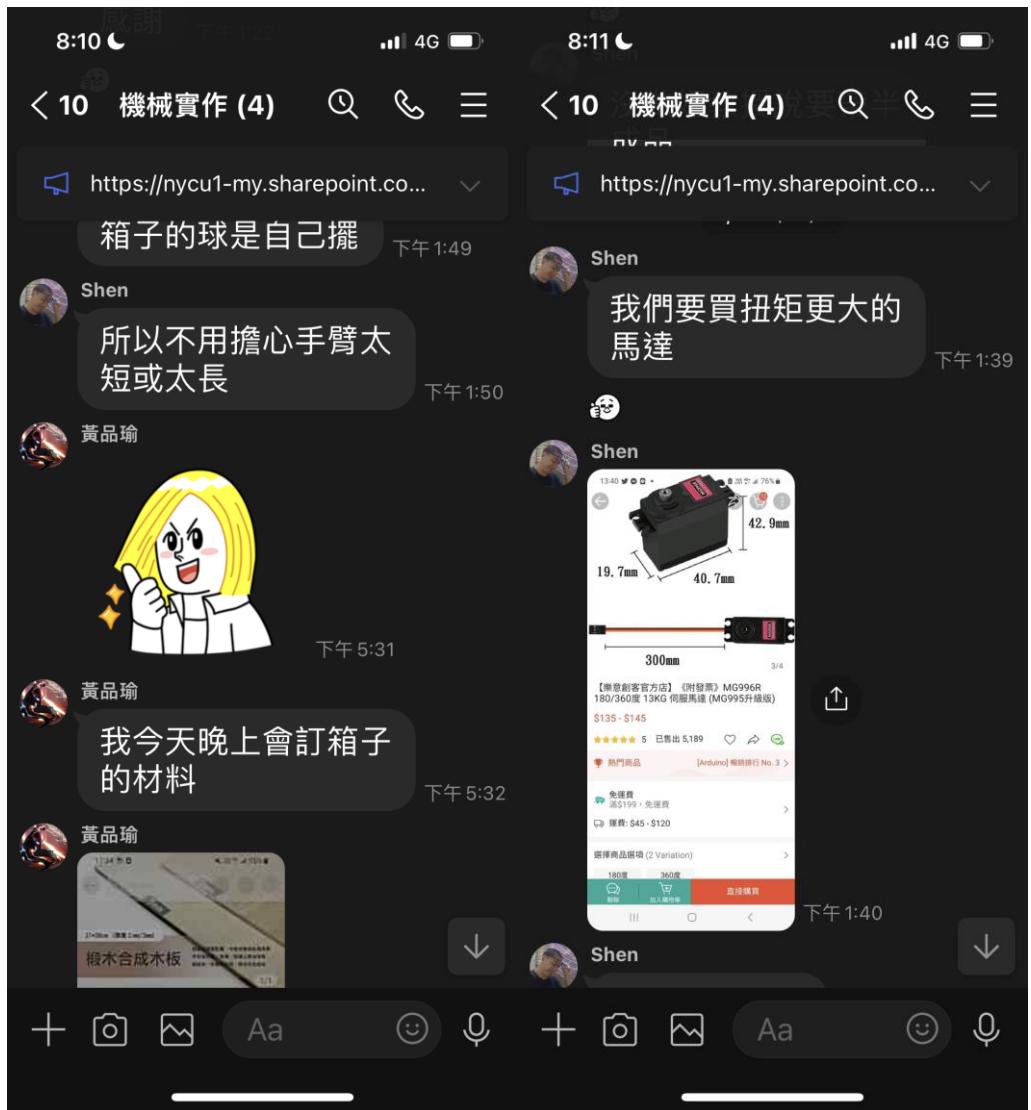
## 10.實作、討論照片(9.10一起)

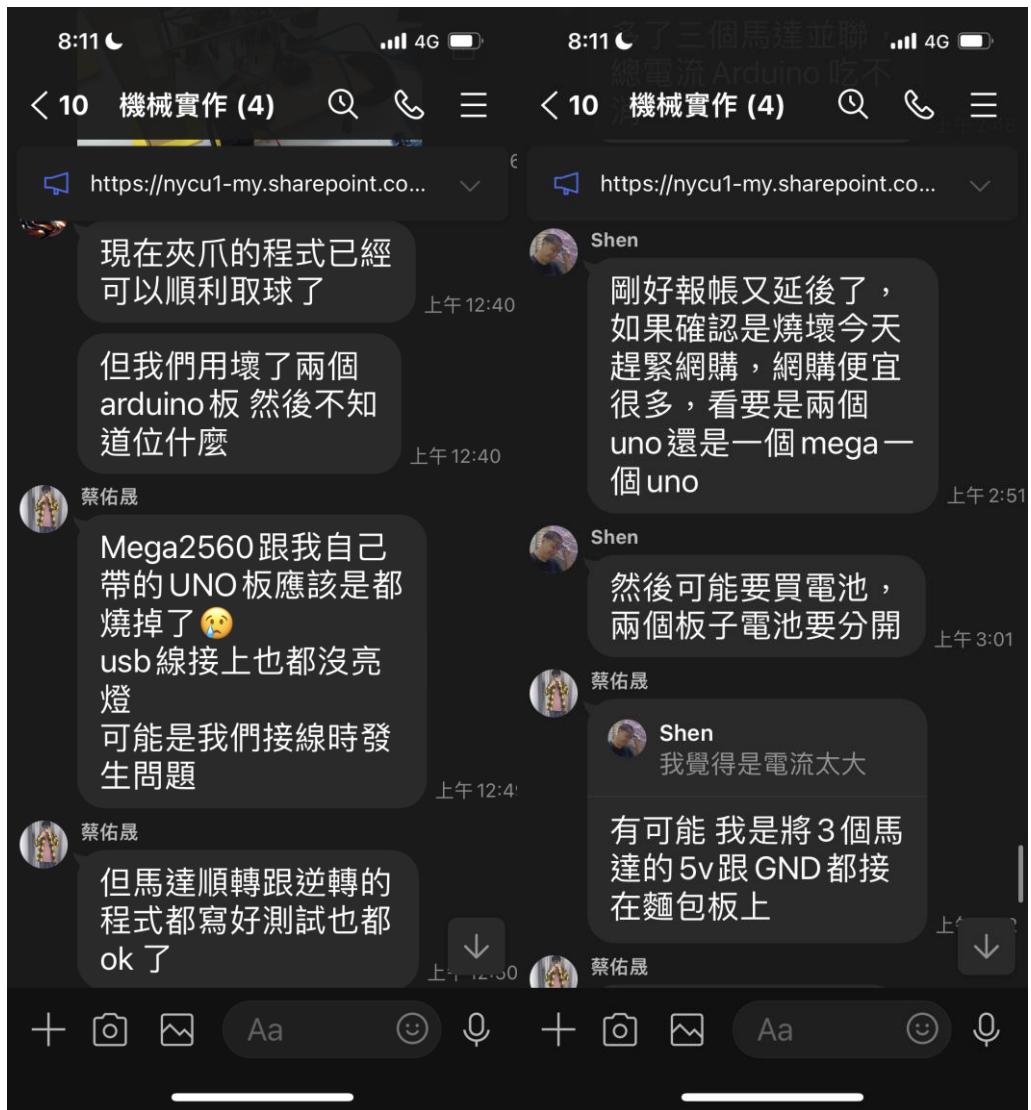


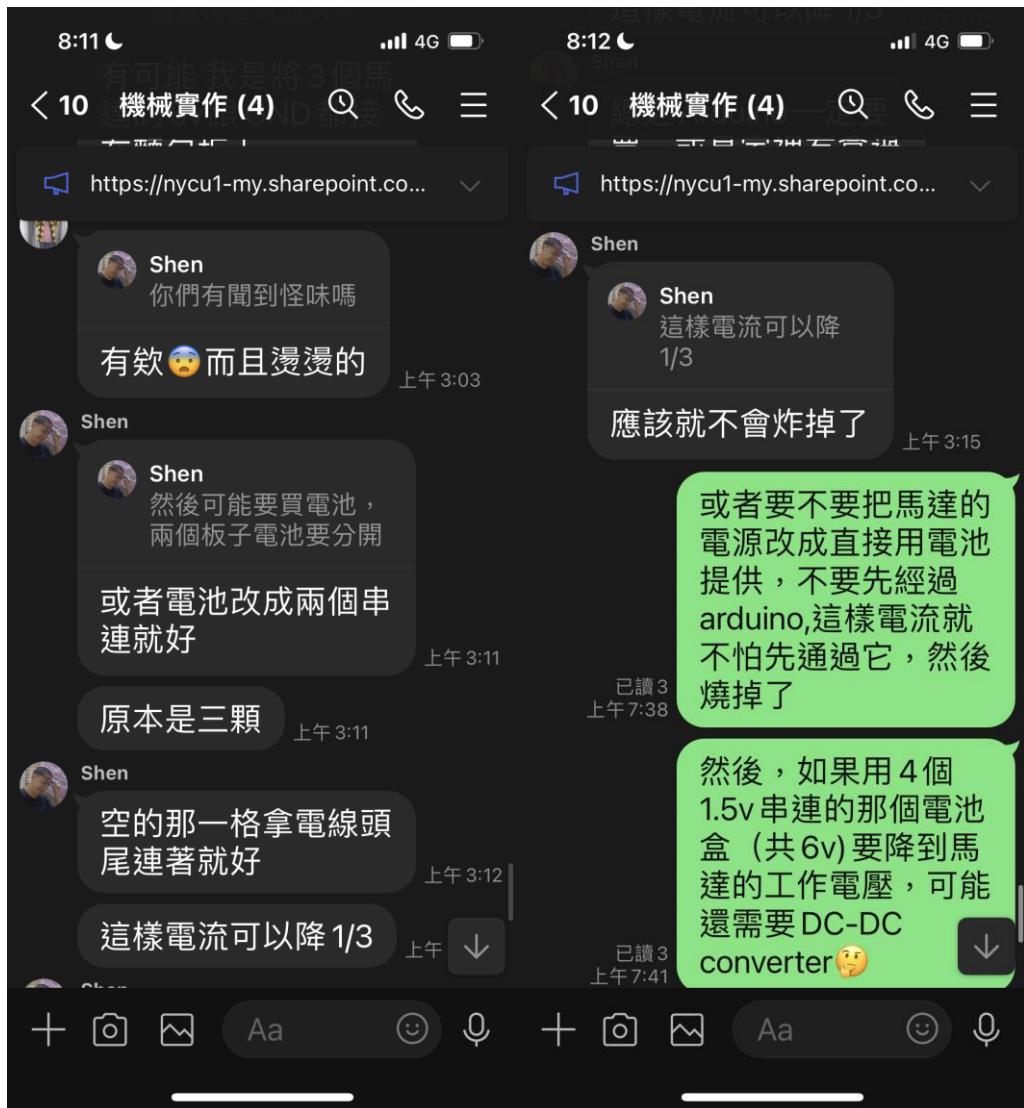


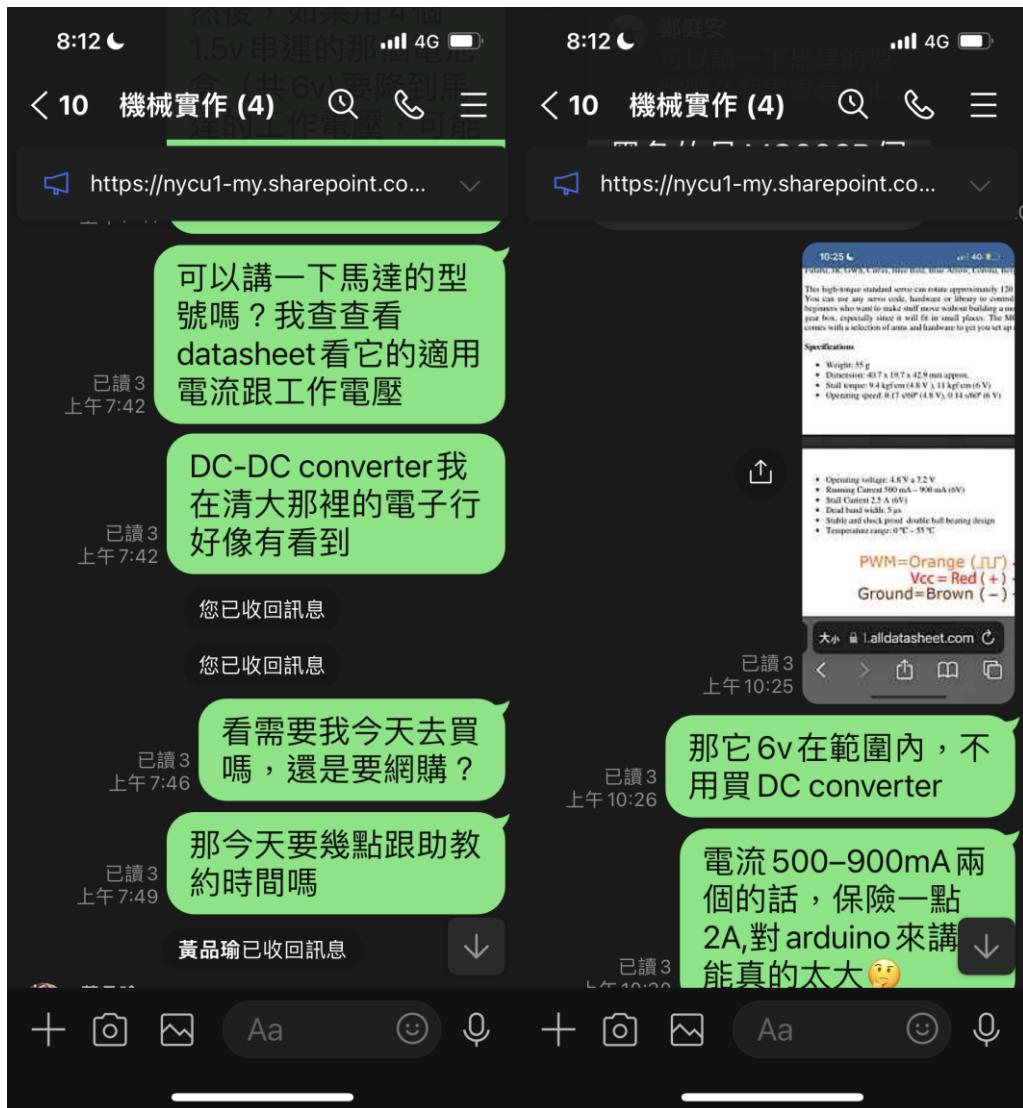
















## 附錄 B: 車輪部分：

```
#define CUSTOM_SETTINGS  
  
#define INCLUDE_GAMEPAD_MODULE  
  
#include <Dabble.h>  
  
const unsigned long BAUD_RATE = 9600;
```

```
int FRONT_ENA_value = 200;

bool circlePressedPrev = false;

bool trianglePressedPrev = false;

//TODO:add LED indicators on car

// SET circle to accelerate, triangle to decelerate

#define MOTOR_FRONT_LEFT_FORWARD 6

#define MOTOR_FRONT_LEFT_BACKWARD 7

#define MOTOR_FRONT_RIGHT_FORWARD 4

#define MOTOR_FRONT_RIGHT_BACKWARD 5

#define MOTOR_REAR_LEFT_FORWARD 10

#define MOTOR_REAR_LEFT_BACKWARD 11

#define MOTOR_REAR_RIGHT_FORWARD 8

#define MOTOR_REAR_RIGHT_BACKWARD 9

#define FRONT_ENA 2

#define FRONT_ENB 3

#define REAR_ENA 12

#define REAR_ENB 13

void setup() {

    pinMode(MOTOR_FRONT_LEFT_FORWARD, OUTPUT);

    pinMode(MOTOR_FRONT_LEFT_BACKWARD, OUTPUT);
```

```
pinMode(MOTOR_FRONT_RIGHT_FORWARD, OUTPUT);
pinMode(MOTOR_FRONT_RIGHT_BACKWARD, OUTPUT);
pinMode(MOTOR_REAR_LEFT_FORWARD, OUTPUT);
pinMode(MOTOR_REAR_LEFT_BACKWARD, OUTPUT);
pinMode(MOTOR_REAR_RIGHT_FORWARD, OUTPUT);
pinMode(MOTOR_REAR_RIGHT_BACKWARD, OUTPUT);
pinMode(FRONT_ENA, OUTPUT);
pinMode(FRONT_ENB, OUTPUT);
pinMode(REAR_ENA, OUTPUT);
pinMode(REAR_ENB, OUTPUT);
Serial.begin(BAUD_RATE);//usb
Serial3.begin(BAUD_RATE); //Serial3 for communication with HM-10,RX 14 TX
15
Dabble.begin(BAUD_RATE,14,15);
Serial.println("setup complete");
}

void loop(){
    Dabble.processInput();
    analogWrite(FRONT_ENA, FRONT_ENA_value);//>70 to move and little
    buzzing
    analogWrite(FRONT_ENB, FRONT_ENA_value);//start speed: 100,ENA
    speed:70--255
```

```
analogWrite(REAR_ENA, FRONT_ENA_value);
```

```
analogWrite(REAR_ENB, FRONT_ENA_value);
```

```
if (GamePad.isUpPressed())
```

```
{
```

```
    moveForward();
```

```
}
```

```
else if (GamePad.isDownPressed())
```

```
{
```

```
    moveBackward();
```

```
}
```

```
else if (GamePad.isLeftPressed())
```

```
{
```

```
    turnLeft();
```

```
}
```

```
else if (GamePad.isRightPressed())
```

```
{
```

```
    turnRight();
```

```
}
```

```
else
```

```
{  
    analogWrite(FRONT_ENA, 0);  
  
    analogWrite(FRONT_ENB, 0);  
  
    analogWrite(REAR_ENA, 0);  
  
    analogWrite(REAR_ENB, 0);  
  
    stopCar();  
  
}  
  
if (GamePad.isCirclePressed() && !circlePressedPrev) {  
  
    motorAccel();  
  
    circlePressedPrev = true;  
  
} else if (!GamePad.isCirclePressed()) {  
  
    circlePressedPrev = false;  
  
}  
  
  
if (GamePad.isTrianglePressed() && !trianglePressedPrev) {  
  
    motorDecel();  
  
    trianglePressedPrev = true;  
  
} else if (!GamePad.isTrianglePressed()) {  
  
    trianglePressedPrev = false;  
  
}  
  
  
}  
  
void motorAccel() {  
  
    if (FRONT_ENA_value < 256) {
```

```
    FRONT_ENA_value += 10;

    updateMotorSpeed();

}

}

void motorDecel() {

    if (FRONT_ENA_value < 256) {

        FRONT_ENA_value -= 10;

        updateMotorSpeed();

    }

}

void updateMotorSpeed() {

    analogWrite(FRONT_ENA, FRONT_ENA_value);

    analogWrite(FRONT_ENB, FRONT_ENA_value);

    analogWrite(REAR_ENA, FRONT_ENA_value);

    analogWrite(REAR_ENB, FRONT_ENA_value);

}

void moveForward() {

    digitalWrite(MOTOR_FRONT_LEFT_FORWARD, HIGH);

    digitalWrite(MOTOR_FRONT_LEFT_BACKWARD, LOW);

    digitalWrite(MOTOR_FRONT_RIGHT_FORWARD, HIGH);

    digitalWrite(MOTOR_FRONT_RIGHT_BACKWARD, LOW);
```

```
digitalWrite(MOTOR_REAR_LEFT_FORWARD, HIGH);
digitalWrite(MOTOR_REAR_LEFT_BACKWARD, LOW);
digitalWrite(MOTOR_REAR_RIGHT_FORWARD, HIGH);
digitalWrite(MOTOR_REAR_RIGHT_BACKWARD, LOW);

}

void moveBackward() {
    digitalWrite(MOTOR_FRONT_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_LEFT_BACKWARD, HIGH);
    digitalWrite(MOTOR_FRONT_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_BACKWARD, HIGH);
    digitalWrite(MOTOR_REAR_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_BACKWARD, HIGH);
    digitalWrite(MOTOR_REAR_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_BACKWARD, HIGH);

}

void turnLeft() {
    digitalWrite(MOTOR_FRONT_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_LEFT_BACKWARD, HIGH);
    digitalWrite(MOTOR_FRONT_RIGHT_FORWARD, HIGH);
    digitalWrite(MOTOR_FRONT_RIGHT_BACKWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_BACKWARD, HIGH);
    digitalWrite(MOTOR_REAR_RIGHT_FORWARD, HIGH);
```

```
    digitalWrite(MOTOR_REAR_RIGHT_BACKWARD, LOW);

}
```

```
void turnRight() {

    digitalWrite(MOTOR_FRONT_LEFT_FORWARD, HIGH);
    digitalWrite(MOTOR_FRONT_LEFT_BACKWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_BACKWARD, HIGH);
    digitalWrite(MOTOR_REAR_LEFT_FORWARD, HIGH);
    digitalWrite(MOTOR_REAR_LEFT_BACKWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_BACKWARD, HIGH);

}
```

```
void stopCar() {

    digitalWrite(MOTOR_FRONT_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_LEFT_BACKWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_FRONT_RIGHT_BACKWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_LEFT_BACKWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_FORWARD, LOW);
    digitalWrite(MOTOR_REAR_RIGHT_BACKWARD, LOW);
```

}

## 11.成員心得：

鄭庭安：這學期的機械實作，增加了真實操作的經驗和 arduino 的程式及硬體連接經驗，之前都僅是學一些理論基礎，並沒有太多相關經驗，在討論的過程中，也能了解到個成員的思維，學到許多不同的想法，是一次很不錯的經驗。

沈宇韜：起初上這堂課可有點怕怕的，之前都是學理論，完全沒有實作經驗，好在有隊友們支持，一起解決問題，途中遇到許多難關，最後看到車子能動有感動到，看到別組的作品更是嚇到掉下巴，沒想到怎麼做出這麼細緻的

黃品瑜：這學期的實作課是個漫長的經歷，能夠親自體驗一個產品從構想到設計到加工到組裝到測試的整個過程是我在這門課得到最大的收穫，過程中組員間的工作分配和意見交流也讓我看到一個團隊作為整體是如何運作的，最後能夠順利做出來每個人的功勞都不能少。

蔡佑晟：最大的遺憾是沒有將夾爪設計得很完美，主要是在硬體的部分，因為組裝的不夠牢靠，讓夾爪在取球的過程晃動得很嚴重，導致在比賽時鬆脫，後來補測時才順利完成，但也只取到一顆球。其實這些隱患一直都有注意到，但我當時只專注在程式的部分，認為夾爪可以順利完成取球、放球的動作就好，沒有意識到機構的**穩定性**才是成敗的關鍵，亦即操作第一次、第十次、第一百次都要可以順利取球放球，而不是一下成功一下失敗。這帶給我很大的反思，一個優良機構須具備高穩定性，讓操作者在使用時可以更安心更順利的完成任務。

## 12.總結檢討

機構方面：

1. 在通過木條路面的時候車身上下晃動太過劇烈，可以增加車底板的長度尺寸，與使用半徑更大的輪胎來改善。
2. 在通過顛簸路面的時候，車子太容易因路面起伏而轉向，可以增加車底

板的寬度，與使用輪胎寬度更大的前輪來改善。

3. 取球過程花了太多時間，而且取球入口不容易對齊球的位置(因為球會滾動)，可以改用一整片格子 (像鬆餅一樣) 的方式來取球。
4. 用螺絲固定機構，有些組件用強力膠固定但太容易鬆脫。

電控方面：

- 1.一開始在接電路時，因為經驗不足，我們 3 個伺服馬達都由 Arduino 板供電，這導致電流過大，燒毀了兩個板子，損失慘重。後來經過小組討論，上網查資料後，我們將 3 顆伺服馬達的電源用 4 顆 1.5V 串聯電池取代，順利 解決問題。
- 2.在夾爪的部分，我們用 2 顆 MG996R 來升降木盒，但在過程中發現，當操作按鍵按太快時，夾爪會直接暴衝，不聽使喚，失去控制。後來經過不斷實驗調整，我們降低伺服馬達的轉速，讓夾爪升降的速度慢一點，就解決問題了。

小組默契方面：

討論事情、提供想法時，多利用開會時間，少用群組。這學期有很多事情在群組上說，結果常常討論到一半沒有結果，或是被其他新訊息蓋掉，最後就是開會時大家都不夠專注，或是意見反覆。