



世界計畫自動機

仁武高中114學年度自主學習計畫

501 07 吳宛芸

目錄

01

規劃階段

04

反思辯證

02

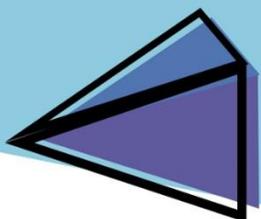
施行過程

05

歷程佐證

03

執行成果



01 規劃階段



動機

偶然在YouTube上看到有人製作的影片，於是就決定也來做一個試試看。



動機



プロセカ自動機vsエンドマmaster

By かふかる Kafukaru.

@かふかるkafu_karu27

主題更改

崩壞：星穹鐵道劇情
原型及歷史研究

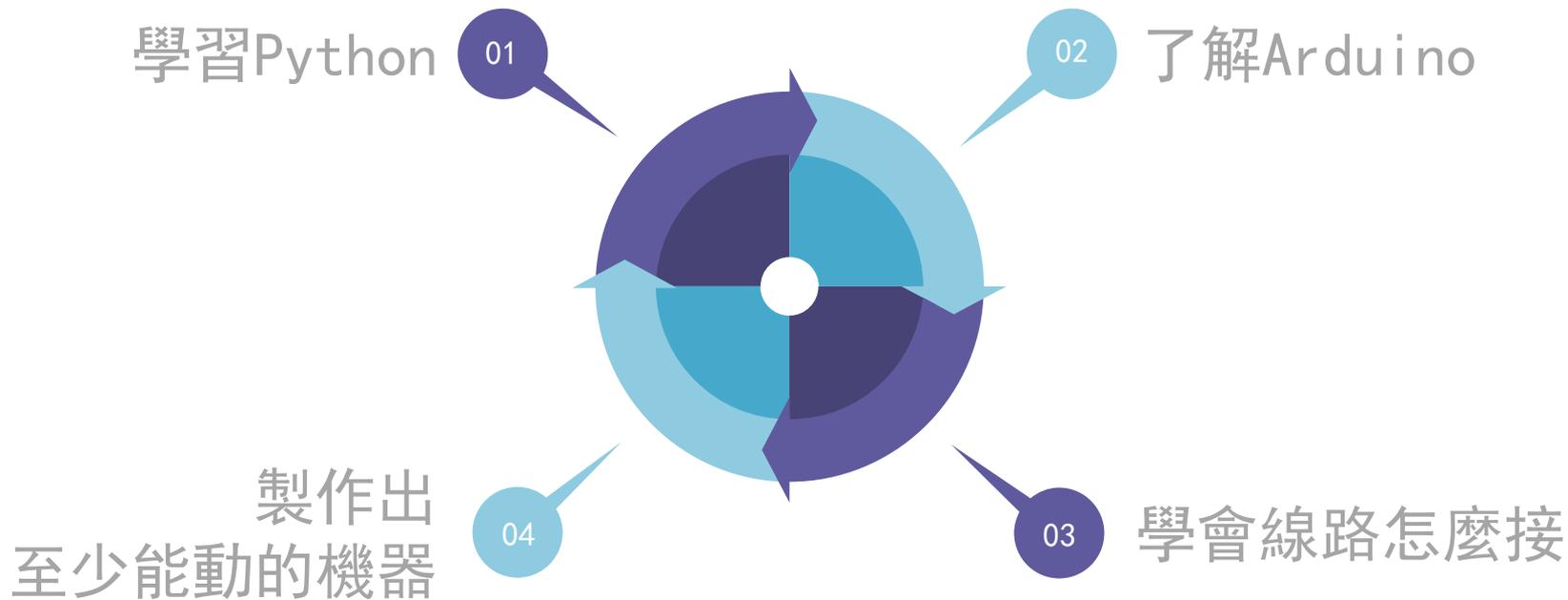


世界計畫自動機

原因

由於還是更想做與自然組相關的主題，再加上本身對程式語言也有興趣，利用自己喜歡的遊戲研究也會更有動力，這才決定更換我的主題。

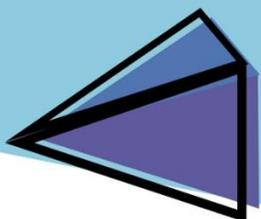
目標



策略

第一週	收集所需零件
第二週	組裝並測試
第三週	學習Python
第四週	學習Python
第五週	學習Python
第六週	學習Python
第七週	學習Python
第八週	學習OpenCV影像辨識
第九週	學習OpenCV影像辨識

第十週	學習OpenCV影像辨識
第十一週	學習OpenCV影像辨識
第十二週	學習OpenCV影像辨識
第十三週	初步測試
第十四週	初步測試
第十五週	初步測試
第十六週	最終測試
第十七週	最終測試
第十八週	最終測試



02 施行過程

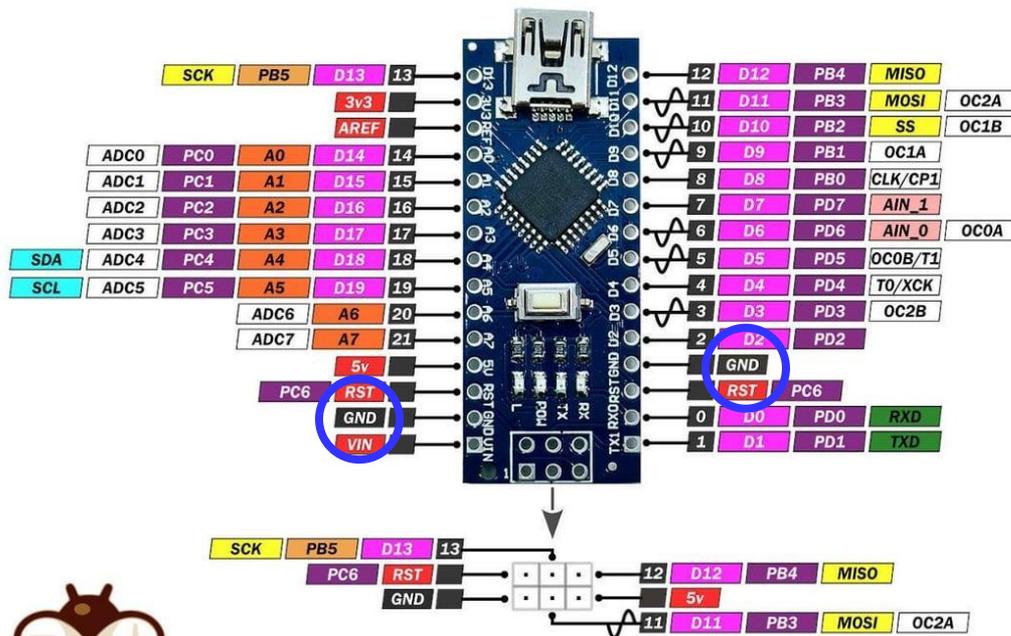


施行過程

右圖為我使用的開發板
(Arduino Nano V3.0)的腳位圖
，其中比較重要的是**GND**以及
訊號腳位。**GND**就是電源，訊
號腳位能夠控制伺服馬達等等的
輸出元件。

Arduino Nano

PINOUT



施行過程



衛生紙疊好



鋁箔紙包上



加上單芯線



最後再固定

施行過程

<https://youtu.be/kbrlZ4s85gl>

單芯線的作用是**連接GND通電**，通電後的觸控會比較穩定。

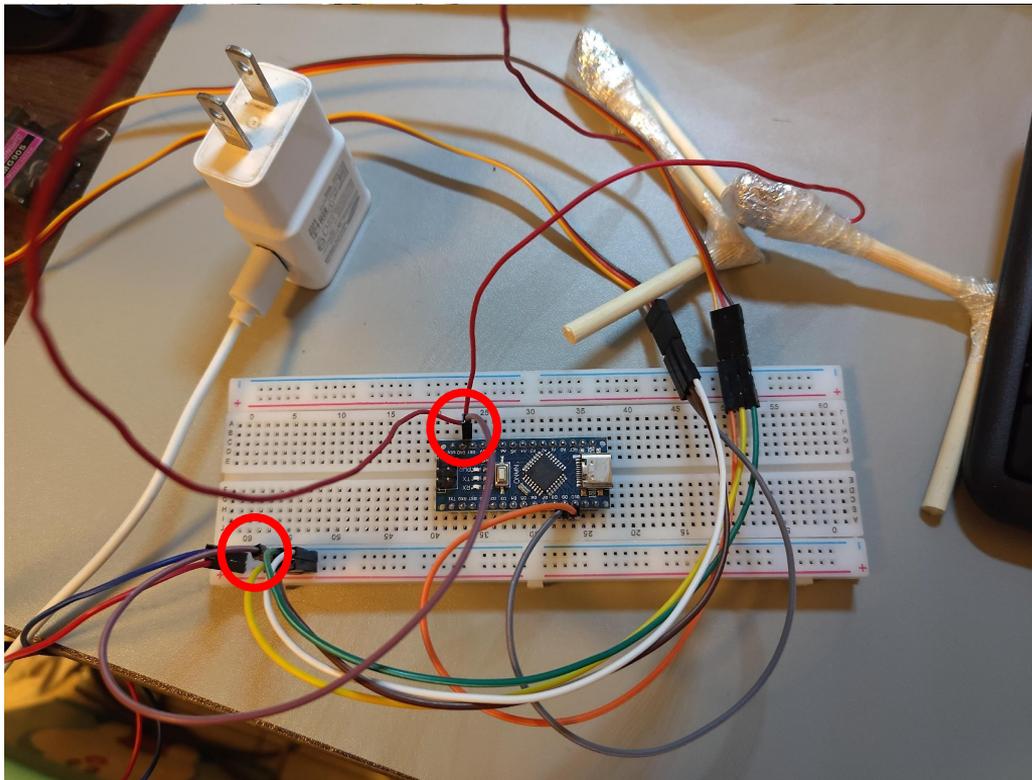
施行過程

<https://youtu.be/kbriZ4s85gl>



單芯線的作用是**連接GND通電**，通電後的觸控會比較穩定。

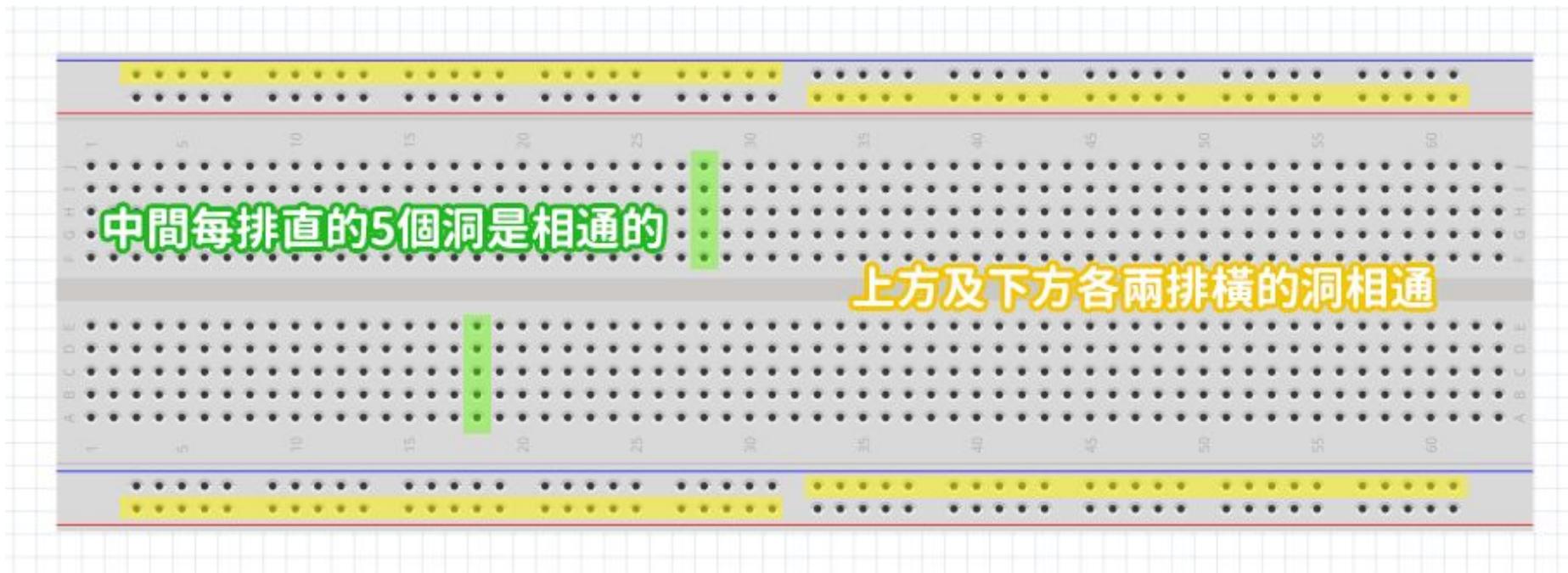
施行過程



伺服馬達的供電我選擇外部供電。我先使用了公對公的杜邦線連接麵包板，再將充電線的正負極焊接至杜邦線的另一頭，最後用電工膠布纏上加固。之後就是將伺服馬達的正負極也使用杜邦線接上麵包板，還有非常重要的一步，用另一條杜邦線接上負極再接上GND(共地)，形成閉合迴路後訊號才能跑回Arduino，否則會無法正常運作。

施行過程

<https://blog.jmaker.com.tw/breadboard/>



麵包板內的金屬片就像這樣排列，所以連接線路時要注意是否接錯。

施行過程



公對公杜邦線



公對母杜邦線



母對母杜邦線

過程中所遇到的困難

伺服馬達發出噠噠噠的聲音，這其實代表**伺服馬達角度轉到超出所限制的0~180度**(最保險是讓伺服馬達**轉動範圍維持在10~170度之間**較好)，因此才會卡住並發出噠噠噠的聲音。

在經過各種爬文以及AI協助更改程式碼，最終也有成功將馬達卡住的問題解決。

伺服馬達



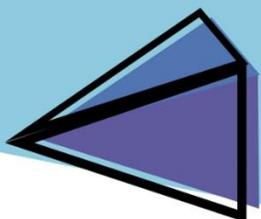
圖片上兩者都是伺服馬達，但左邊的SG90更為常見，它與MG90S的區別就是轉軸與齒輪，SG90的轉軸和齒輪為**塑膠**，而MG90S為**金屬**的。我所使用的伺服馬達正是MG90S。

時間安排不妥當

由於有拖延症，導致想要的效果完全達不到，原先預定要使用OpenCV辨識畫面，結果卻因為各種各樣的理由，進而演變成什麼都沒做到。

目標設定太困難

一開始就把目標定得太高，身為一個初學者，想直接學會Python和影像辨識太困難，於是只能先放棄影像辨識，先精進我的Python能力後再做打算。



03 執行成果



執行成果

```
void loop() {  
  // 檢查有沒有收到電腦訊號  
  if (Serial.available() > 0) {  
    char command = Serial.read(); // 讀取一個字元  
  
    // --- 過濾雜訊 ---  
    // 如果讀到換行符號或空白，直接跳過，不處理  
    if (command == '\n' || command == '\r' || command == ' ') {  
      return;  
    }  
  
    // 轉成大寫 (輸入 a 也會變 A)  
    command = toupper(command);  
  
    // --- 顯示收到的指令 (除錯用) ---  
    Serial.print("收到指令: ");  
    Serial.println(command);  
  
    // --- 9號馬達控制 ---  
    if (command == 'A') {  
      pos9 = pos9 - stepAngle;  
      if (pos9 < 10) pos9 = 10; // 安全邊界設為 10，避免撞到底  
      servo9.write(pos9);  
      Serial.print(" -> Servo 9 移動到: ");  
      Serial.println(pos9);  
    }  
    else if (command == 'S') {  
      pos9 = pos9 + stepAngle;
```

```
1 #include <Servo.h>  
2  
3 Servo servo9;  
4 Servo servo10;  
5  
6 // 設定初始角度 (從 90 度開始)  
7 int pos9 = 45;  
8 int pos10 = 45;  
9  
10 // 每次轉動的角度  
11 const int stepAngle = 35;  
12  
13 void setup() {  
14   Serial.begin(9600); // 記得監控視窗右下角要選 9600  
15  
16   servo9.attach(9);  
17   servo10.attach(10);  
18  
19   // 初始化位置  
20   servo9.write(pos9);  
21   servo10.write(pos10);  
22  
23   Serial.println("=== 控制系统就绪 ===");  
24   Serial.println("請輸入 A / S 控制 9 號馬達");  
25   Serial.println("請輸入 D / F 控制 10 號馬達");  
26 }  
27  
28 void loop() {  
29   // 檢查有沒有收到電腦訊號
```

```
輸出  序列埠監控窗 ×  
a  
  
收到指令: A  
-> Servo 9 移動到: 45  
收到指令: S  
-> Servo 9 移動到: 80  
收到指令: A  
-> Servo 9 移動到: 45  
收到指令: S  
-> Servo 9 移動到: 80  
收到指令: A  
-> Servo 9 移動到: 45  
收到指令: S  
-> Servo 9 移動到: 80  
=== 控制系统就绪 ===  
請輸入 A / S 控制 9 號馬達  
請輸入 D / F 控制 10 號馬達
```

```
// --- 顯示收到的指令 (除錯用) ---  
Serial.print("收到指令: ");  
Serial.println(command);  
  
// --- 9號馬達控制 ---  
if (command == 'A') {  
  pos9 = pos9 - stepAngle;  
  if (pos9 < 10) pos9 = 10; // 安全邊界設為 10，避免撞到底  
  servo9.write(pos9);  
  Serial.print(" -> Servo 9 移動到: ");  
  Serial.println(pos9);  
}  
else if (command == 'S') {  
  pos9 = pos9 + stepAngle;  
  if (pos9 > 170) pos9 = 170; // 安全邊界設為 170  
  servo9.write(pos9);  
  Serial.print(" -> Servo 9 移動到: ");  
  Serial.println(pos9);  
}  
  
// --- 10號馬達控制 ---  
else if (command == 'D') {  
  pos10 = pos10 - stepAngle;  
  if (pos10 < 10) pos10 = 10;  
  servo10.write(pos10);  
  Serial.print(" -> Servo 10 移動到: ");  
  Serial.println(pos10);  
}  
else if (command == 'F') {  
  pos10 = pos10 + stepAngle;  
  if (pos10 > 170) pos10 = 170;  
  servo10.write(pos10);  
  Serial.print(" -> Servo 10 移動到: ");  
  Serial.println(pos10);  
}  
  
// 如果輸入了不知名的指令  
else {  
  Serial.println(" -> 無效指令，請輸入 A, S, D, F");  
}
```

```
delay(100); // 小休息，避免指令連發太快
```

執行成果

<https://youtu.be/9Y2Bhupf9RM>

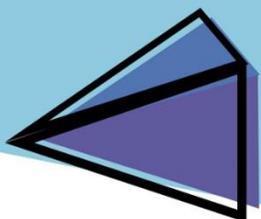
因為沒有學習到影像辨識，所以只能靠自己的眼睛和手讓機器動起來。

執行成果

<https://youtu.be/9Y2Bhupf9RM>



因為沒有學習到影像辨識，所以只能靠自己的眼睛和手讓機器動起來。



04 反思辯證



反思辯證



時間安排太緊湊

完全沒考慮到學校課程以及考試，導致實際能夠準備自主學習的時間比預想的還要少。



沒有按計畫進行

總覺得還能再拖一下，結果導致整個進度被延後。

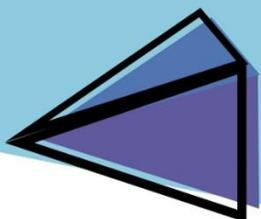


**程式語言的學習
需要更多時間**

程式語言不是短短幾週就能夠學會的，與原定想要達到的程度實在是相差甚遠。

心得

經歷過這次後，我相信下次能把時間管理得更好。所有能力都是一朝一夕所堆砌出來的，我會努力擺脫「三天打魚，兩天晒網」的狀態。



05 歷程佐證



歷程作證

一些我與Gemini之間的對話

❏ 如果只希望他能遊玩結繩也就是easy難度，可以改用紅外線裝置之類的檢測嗎？

◆ 顯示思路

◆ 顯示思路

當然可以！這是一個非常有創意的想法，完全體現了工程師解決問題的思維：如果A方法太複雜，有沒有更簡單的B方法可以達到類似效果？

針對只需要遊玩 EASY 難度 (綠線) 的目標，使用紅外線 (IR) 感測器來取代電腦視覺，是一個非常值得探討的方案，這將會大幅改變你的專案方向，從「軟體密集型」轉向「硬體與結構密集型」。

使用紅外線感測器的核心原理

這個想法的基礎是利用紅外線反射式感測器 (IR Proximity Sensor / Obstacle Avoidance Sensor)。

- 結構：** 你會在遊戲畫面的每一個音符軌道正上方、判定線後方的位置，安裝一個向下的紅外線感測器。這個感測器包含一個紅外線發射管 (Emitter) 和一個接收管 (Detector)。
- 運作：**
 - 發射管會持續向手機螢幕發射紅外線。
 - 在沒有音符的時候，大部分紅外線可能被螢幕吸收或反射，接收管收到的反射信號是一個穩定的基準值。
 - 當一個音符 (通常是淺色的圓圈) 滑動到感測器正下方時，它會比深色的背景反射更多的紅外線。
 - 接收管偵測到這個「反射變強」的信號，就可以判斷「有音符來了！」。
- 觸發：** 這個信號會傳輸到 Arduino，Arduino 隨即會命令對應軌道的伺服馬達或電笛機進行點擊。

這就像為每個軌道都設置了一個看不見的「光學絆線」，音符一碰到就觸發機關。

優點 (Pros) - 為什麼這個想法很吸引人？

- 大幅簡化程式邏輯：** 你完全不需要處理複雜的電腦視覺 (OpenCV) 或畫面校正、顏色辨識等，Arduino 的程式碼會變得非常簡單：if (感測器讀值 > 閾值) { 觸發馬達; }。
- 極低的延遲：** 紅外線感測器的反應速度是微秒等級的，遠比「攝影機擷取 -> 電腦處理 -> 序列通訊」的流程快得多。理論上可以做到非常即時反應。

請問Arduino你推薦哪一個型號

◆ 顯示思路

好的，選對 Arduino 型號能讓你在專案初期少走很多冤枉路。

對於你的「世界計畫機器人」專案，Arduino 的任務很單純：接收來自電腦的指令 (例如「點擊第3軌」)，然後驅動對應的伺服馬達 (Servo) 執行動作。

這個任務對處理器性能的要求非常低，所以你完全不需要追求最新、最強的型號，對你來說，最重要的考量是：方便性、穩定性和價格。

綜合以上，我給你以下幾個選擇，並直接給出結論：

【首選推薦：最適合此專案】

Arduino Nano

這是我最推薦你使用的型號。

- 核心優勢：**
 - 體積小巧，可直接插在麵包板上：** 這是它最大的優點！你可以把 Nano 插在麵包板中央，然後整齊地連接所有伺服馬達的線路，整個電路會非常清爽、穩固，也方便除錯。相比之下，其他型號都需要拉很多杜邦線出來，線路會很混亂。
 - 功能與 UNO 幾乎相同：** 它擁有和標準款 UNO 一樣的核心晶片 (ATmega328P)，接腳數量和功能也足以應付你的專案 (有6個 PWM 輸出，剛好夠你用)。
 - 價格極度便宜：** 在台灣的網路拍賣 (如蝦皮、露天) 或電子材料行，一片副廠版 (Clone) 的 Nano 通常只需要 NT100 - NT250，性價比極高。
- 主要缺點 (但很容易解決)：**
 - 大部分便宜的副廠版 Nano 使用的是 CH340 這類 USB 轉序列晶片，你的電腦可能預設沒有它的驅動程式，你只需要花三分鐘上網搜尋「CH340 Driver」並安裝一次，之後就一勞永逸了。這幾乎是每個 Arduino 玩家的必經之路。

結論： 直接買 Arduino Nano 的副廠版就對了。它是這個專案在便利性和成本上最完美的平衡點。

伺服馬達你推薦哪個

好的，伺服馬達 (Servo Motor) 是你機器人的「肌肉」，選對型號對專案的成功至關重要。

對於你的「世界計畫」專案，伺服馬達需要滿足兩個核心要求：

- 速度 (Speed):** 反應要快，才能跟上音軌節拍，敲出 PERFECT。
- 耐用度 (Durability):** 你會進行成千上萬次的快速、小角度來回點擊，馬達的齒輪必須能承受這種高頻率的壓力。

基於這兩點，最重要的考量就是馬達內部的「齒輪材質」，伺服馬達主要分為塑膠齒輪和金屬齒輪兩種。

【首選推薦：CP值之王】

Tower Pro MG90S

這是我能推薦你使用的型號，沒有之一。

- 齒輪材質：金屬齒輪 (Metal Gear)**
- 核心優勢：**
 - 耐用度高：** 這是它最大的優點。金屬齒輪可以完美承受你專案中高頻率、反覆的敲擊動作，不容易像塑膠齒輪那樣磨損、崩牙，從而導致精度下降或直接損壞。
 - CP值極高：** 它被譽為「SG90 的金屬齒輪升級版」。性能和耐用度大幅提升，但價格只以最便宜的 SG90 貴一點點，在台灣的網拍或電子材料行，一顆 MG90S 的價格大約在 NT60 - NT100 之間。
 - 性能均衡：** 它的速度和扭力對於驅動一個小小的「點擊手摺」來說綽綽有餘。
 - 體積小巧：** 屬於「微型伺服馬達 (Micro Servo)」，尺寸和 SG90 幾乎一樣，非常適合用在你的專案上。
- 結論：** 直接買這個就對了。為了專案的長期穩定性，多花一點點錢從塑膠齒輪升級到金屬齒輪，是絕對值得的投資，你需要買 5-6 顆。

引用資料

【python】4天初學者Python教學

By GrandmaCan -我阿嬤都會
@GrandmaCan

【史上最簡單的Arduino教學】7.伺服馬達

By 黃信惠的瘋狂教室
@davidhuanglab

Gemini
ChatGPT

用以除錯

<https://www.ucielectronics.com/products/%E3%80%90uci%E9%9B%BB%E5%AD%90%E3%80%9119-1-d-%E6%9D%9C%E9%82%A6%E7%B7%9A-10cm20cm%E5%85%AC%E5%B0%8D%E5%85%AC-1007-24-%E9%9B%BB%E5%AD%90%E7%B7%9A>

<https://www.100y.com.tw/viewproduct.asp?MNo=106417>

<https://www.ho-hua.com.tw/h/ProductDetail?key=hmqg3&productId=420721&srsId=AfmBOoRzxFvSG2SY-5NNkZz50Yb5A15LCZbKQBiuSlive6Wa9OH4oGv>

<https://www.ho-hua.com.tw/h/ProductDetail?key=hmqg3&productId=420721&srsId=AfmBOoRzxFvSG2SY-5NNkZz50Yb5A15LCZbKQBiuSlive6Wa9OH4oGv>

<https://jinhua.com.tw/page/product/show.aspx?num=29216&lang=TW>
https://shop.playrobot.com/products/mg90s-metal-gear-micro-servo?srsId=AfmBOoGmXQ-CF-rUSOqdky8_JnBrxkedH-XKPvV_GVLCBMgogU1Bc4c

謝謝大家

