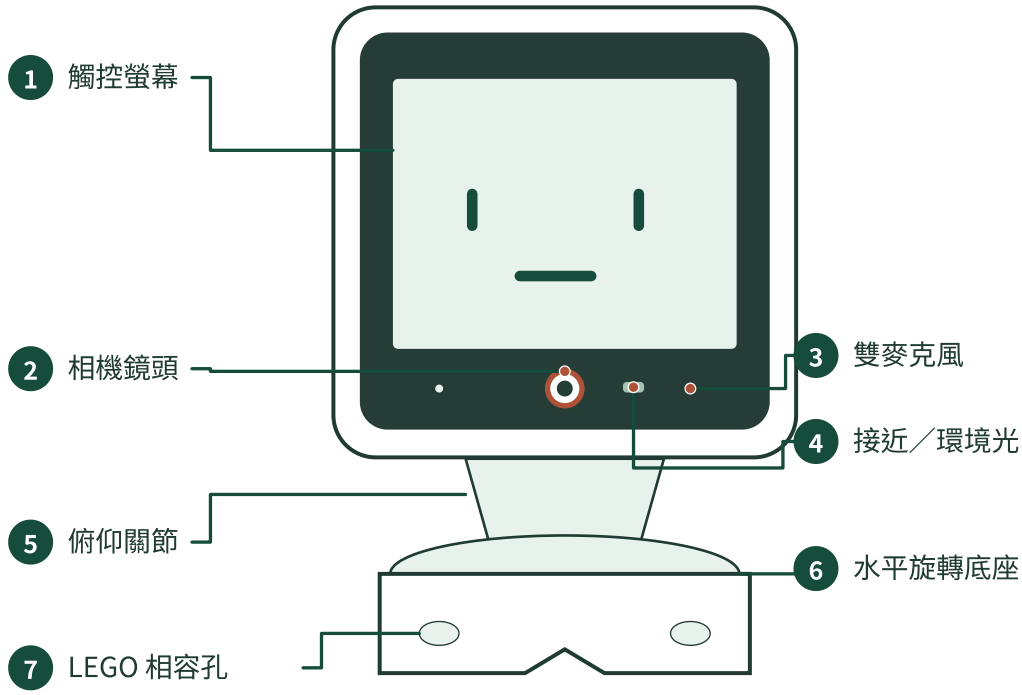
我的第一台 AI 提醒機器人

認識 StackChan，和 AI 一起改程式，
把一個生活需求做成可以測試的作品。

正面示意｜面向螢幕



位置辨識圖，比例與孔位間距經簡化；不是按鈕的圓孔不要按。

姓名／小組：_____　日期：_____

2026-10-27 編寫。本書圖解由課程團隊以 SVG 原創繪製，依官方資料簡化重畫；不是原廠說明書或工程尺寸圖。未完成的營隊裝置整合另有標示。

先知道要做什麼，再一步一步完成

兩天的作品方向是一台「健康習慣提醒機器人」。從書桌前的小需求出發，例如提醒自己換個姿勢或休息一下；你要學的是把需求說清楚、修改一小段程式、觀察結果，再依使用者回饋調整。

> **每次都做這四件事**
>
> 先說預期 → 只改一件事 → 實際操作 → 記下結果。程式沒有報錯，也要看看機器人是否真的做對。

內容	頁次	你會完成什麼
硬體圖解	3–6	找對按鈕與插槽，說出零件用途
首次開機與 App	7–10	完成到貨檢查、找到安裝連結、理解配對
AI 協作與韌體	11–15	閱讀專案、修改顏色、編譯與 USB 驗證
提醒與受控工具	16–17	在電腦練習計時與狀態回報
健康習慣專題與測試	18–22	互相試用、排錯、展示可驗證的成果
紀錄與課前驗收	23–25	留下實驗筆記，查閱講師準備與資料來源
進階教材	26–30	認識 GitHub 與 Stack-chan 開源家族，換韌體指向天空，讓機器人跟著人轉，用 PID 微調
參考資料	31	課程提到的開源專案簡述與授權

讀懂三種狀態

已實測：本次實機已開機、完成馬達測試，並進入等待 App 配對畫面。馬達動作正常由使用者現場回報，完成頁有照片佐證。10-28 另以 USB 讀到原廠韌體開機紀錄，確認筆電與機器人的資料連線可用。

已編譯：Camp Lab 補丁曾在 ESP-IDF 5.5.4 完整編譯成功。這還不能證明燒錄、螢幕按鈕、退出重開與 USB 紀錄正常。

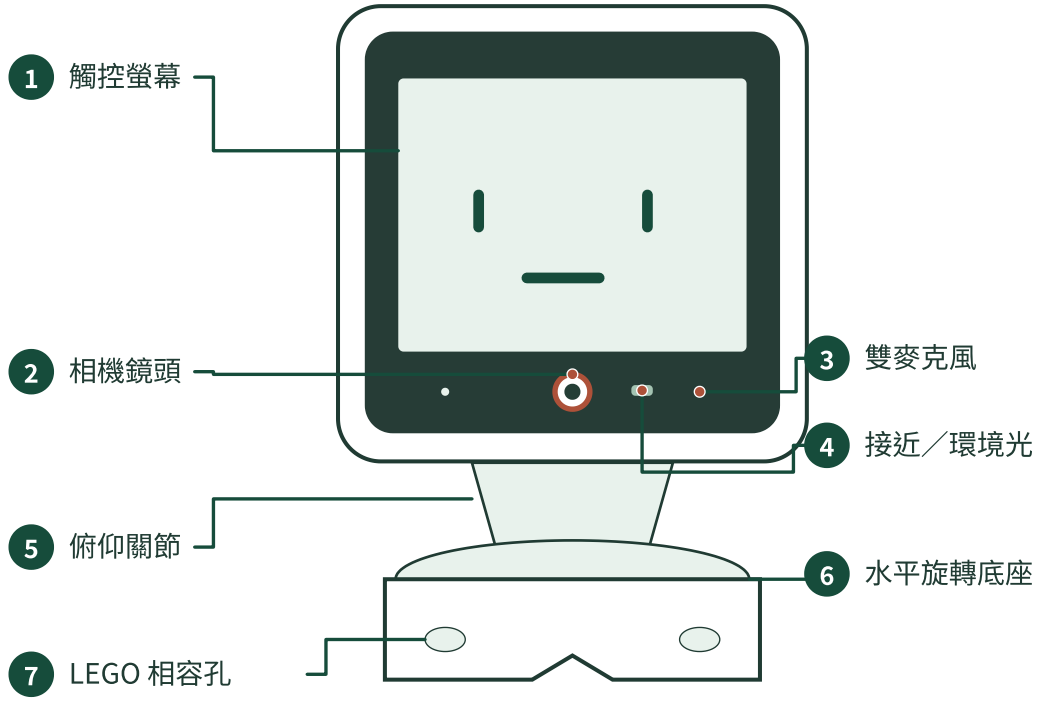
待整合：營隊提醒韌體、裝置身分驗證、受控工具與正式 AI 後端，仍需講師課前完成。看到這個標記時，先做本機練習或等待講師指示。

開始前準備

- 一台 StackChan、可傳資料的 USB 線、筆電和平穩桌面。
- 依講師安排使用手機與帳號；密碼自己輸入，不寫入課本或投影畫面。
- 做韌體課之前，講師須準備工具、基準程式與復原方式。

先看清楚：哪些是輸入，哪些會動？

正面示意｜面向螢幕



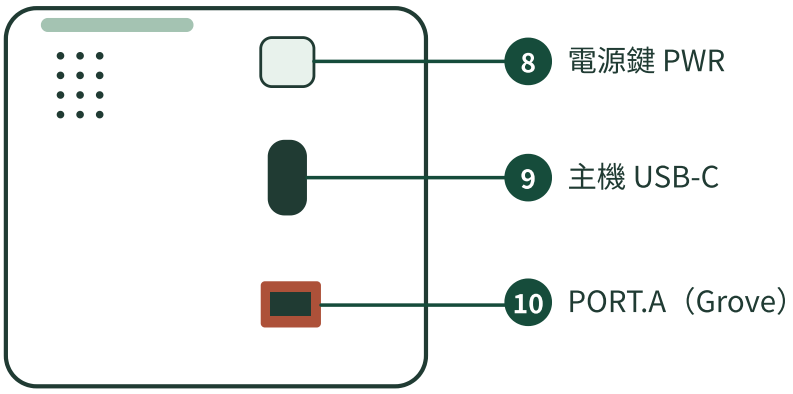
位置辨識圖，比例與孔位間距經簡化；不是按鈕的圓孔不要按。

圖號／部位	用途	第一次使用要知道
1 觸控螢幕	顯示表情、選單，接收點按與滑動	2 吋、320 × 240；不是機身頂部觸控板
2 相機／3 麥克風	取得影像／接收聲音	鏡頭和收音孔都不是按鈕，不用物品戳入
4 接近與環境光	量測附近物體與環境亮度的變化	讀值需要程式處理；不會自動理解你的行為
5 俯仰／6 水平旋轉	兩顆帶回授伺服馬達控制頭部	留出活動空間，通電受控時不要硬轉
7 LEGO 相容孔	固定裝飾或擴充結構	不是充電孔或訊號接口

依據：官方 StackChan 外觀與 CoreS3 感測器說明。示意圖簡化比例；光學窗與微小孔位請同時比對實物。鏡頭位於螢幕下方，並非螢幕上的表情。

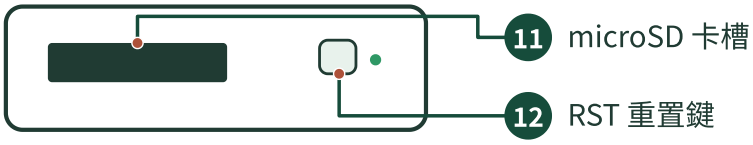
開關機、重置與插線，要找對位置

主機左側｜面向螢幕時



側面淺色長窗：RGB LED；小圓孔群：喇叭。

主機下緣｜microSD／RST 面展開示意



底座 USB-C｜連電腦優先選這裡



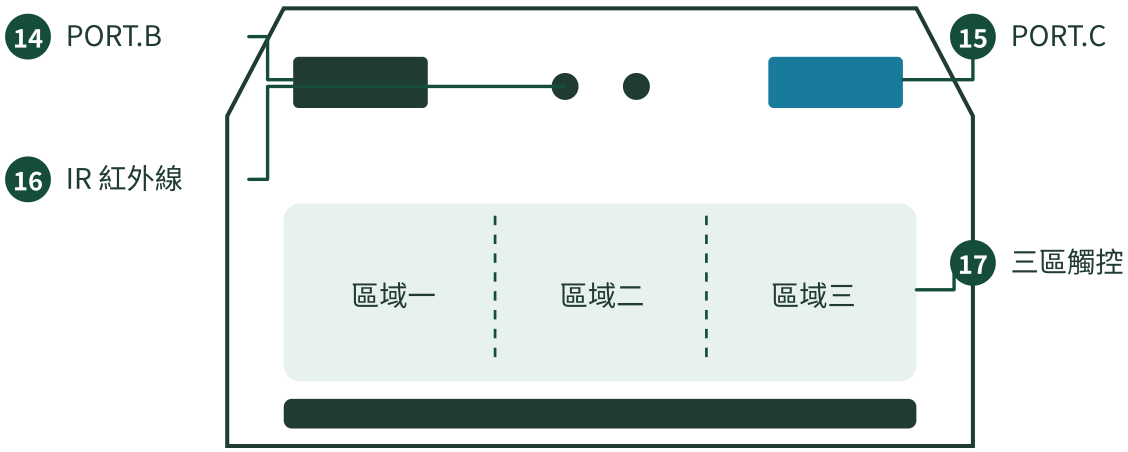
兩個 USB-C 均支援資料傳輸；底座接線可減少轉頭時拉扯。

圖號／部位	用途與操作
8 電源鍵 PWR	主機左側、USB-C 旁。短按開機；長按約 6 秒關機。
9 主機 USB-C／13 底座 USB-C	兩者都可供電、傳資料與燒錄；課堂優先接底座。
10 PORT.A	4-pin Grove 擴充接口，常用於 I^2C；不是 USB。
11 microSD／12 RST	卡槽用於檔案儲存；旁邊 RST 短按重啟，長按約 3 秒進下載模式。

官方操作依據：StackChan「開關機」「下載模式」。圖中底面以展開視角呈現，不需要拆主機。microSD 讀寫時不要拔卡；插卡與更換擴充由講師確認後操作。

頂部也會感應，插孔也有分工

頂部功能示意｜排列經展開，請比對實物標示



這是機身觸控板；螢幕觸控是另一組輸入。

接口與電極區排列為教學示意，不用圖面左右判斷接線。

圖號／部位	可以用來做什麼
14 PORT.B	Grove GPIO 擴充；接指定的數位／類比模組前，先確認相容性。
15 PORT.C	Grove UART 串列通訊；依標示與講師準備的模組連接。
16 紅外線 IR	發送與接收紅外訊號。不是一般相機，也不是 Wi-Fi。
17 機身三區觸控	偵測碰觸或滑動；課程可設計確認提醒，功能仍需程式實作。

還有兩種常見輸出

RGB LED：機身共有 12 顆，可顯示顏色和狀態。原廠 AI 模式的單顆狀態燈：綠色聆聽、藍色說話。

喇叭：主機側面有出音孔，1 W 喇叭播放語音與音效。

擴充前先問三件事

1. 這個模組應該接哪一個 PORT？
2. 供電、訊號腳位與程式庫是否相符？
3. 講師提供的範例是否已實機測過？

NFC 線圈與 RGB 元件位於機身內，沒有需要插入卡片的 NFC 卡槽。頂面圖只說明功能分區，接口左右與感應區邊界請以實物標示核對；不用來判定程式 API 的通道編號。

感測器收集訊號，程式決定怎麼回應

一個機器人，四種工作



硬體	規格／用途	怎麼理解
ESP32-S3 主控	雙核心，最高 240 MHz	執行韌體、讀感測器、控制設備
Flash／PSRAM	16 MB／8 MB	程式儲存空間／執行時的工作空間
相機與音訊	GC0308，640 × 480；雙麥克風與喇叭	拍攝、收音和播放；AI 理解另需軟體與服務
IMU 與地磁	BMI270＋BMM150，合計九軸	感知加速度、旋轉和磁場；本課不靠它診斷健康
接近／環境光	LTR-553ALS-WA	取得附近物體與亮度的量測值
NFC	ST25R3916	近距離與相容標籤互動；可讀卡型與距離須驗證
無線連線	2.4 GHz Wi-Fi、Bluetooth LE	配對與連網；不支援 5 GHz Wi-Fi
電源與時鐘	550 mAh 電池、AXP2101、BM8563	電源管理與時間；續航依動作、音量及網路而變

有硬體，不代表功能已經做好。

有觸控板，仍要寫「碰一下如何確認提醒」；有 Wi-Fi，仍要設定網路、後端和裝置權限。原廠功能、自寫韌體和本機模擬要分開驗證。

StackChan 原裝核心以 CoreS3 為基礎。一般 CoreS3 不能直接當作原裝 StackChan 主控綁定 World App；不要把其他 CoreS3 套件的電池或供電規格套用到整台機器人。

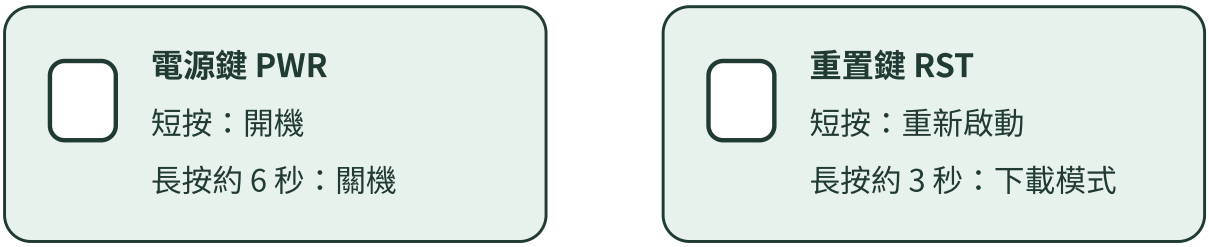
開機、馬達測試，與今天怎麼收工



已取得開機與馬達測試結果；配對、Wi-Fi、語音仍待實機驗證。

先做四個動作

1. **接線與開機：**USB 資料線接底座和電腦。需要時短按電源鍵，記下第一個畫面。
2. **開始測試：**歡迎頁點 Start；把機器放在堅硬平穩的桌面，手離開機身，留出轉頭空間，再點 Start。
3. **觀察：**看是否向左、向右、向上轉動，最後回到正面中央。完成頁應顯示 `Servo test completed.`。
4. **繼續：**點 Next 進 App 安裝頁，再依畫面前進到等待配對。已設定過的機器不一定出現相同精靈。



RST 下載模式：接資料線 → 長按 → 綠燈亮 → 放開。

今天的觀察

開機第一頁：_____　馬達測試結果：_____

我如何分辨電源鍵和 RST？_____

下載模式供講師需要時使用：資料線連電腦，長按 RST 約 3 秒，綠燈亮起後放開。它不是日常關機方法。垂直馬達的程式角度依官方建議控制在 5–85°；不要用手硬轉受控關節。

安裝 StackChan World

選擇你的手機系統，打開相機掃描下方 QR code。已在手機閱讀課本時，直接點下載連結即可。請確認 App 名稱為 **StackChan World**，開發者為 M5Stack。

iPhone／iPad

開啟 App Store 下載

Android

開啟 Google Play 下載

掃不到時，不用一直對準機器人的小螢幕。

講師開啟同資料夾的 projection.html 課堂投影頁，可切換單一平台放大。也可在商店搜尋「StackChan World」，核對開發者後安裝。

安裝後做什麼？

1. 依講師安排登入 M5Stack 帳號。密碼自己輸入。
2. 回到機器人，依畫面點 Next，看到裝置 ID。
3. 進入 App 的新增裝置流程，選擇與螢幕 ID 相符的機器人。

連結由 M5Stack 官方文件與兩個商店頁核對，查證日 2026-10-27。QR 本身可離線顯示；前往商店、下載 App 與登入仍需要網路。帳號安排由講師課前確認。

我的完成條件

☐ 找到正確 App　☐ 成功安裝　☐ 依講師安排登入　☐ 記下遇到的問題

選自己的那一台，設定它的連線

同一間教室可能有很多機器人。「距離最近」不代表一定是你的；要比對機器人螢幕顯示的 ID。課本不放任何一台真實機器的 ID，你要看自己面前的螢幕。

機器人上

ID：你的裝置識別碼

Look for me in the app
to start setup.

這個畫面表示等待手機開始設定，不能據此認定已連上 Wi-Fi 或 AI。

手機 App 上

1. 點 `Add a new StackChan`。
2. 允許 App 執行配對所需的藍牙等系統權限。
3. 比對螢幕 ID，選擇正確機器。
4. 依 App 提示設定名稱與 AI 角色；語言、模型與音色照講師指定（中文選「普通话」）。
5. 最後填 Wi-Fi；機器人自動重開後，綁定才算完成。

連線先核對這四件事

問題	課堂做法
是不是 2.4 GHz？	StackChan 的 Wi-Fi 不支援 5 GHz；使用講師測過的 2.4 GHz 網路。
是不是要網頁登入？	不要使用需要額外開網頁驗證的訪客網路。
設定是否完成？	App 首頁出現你的機器人，進 `AI.AGENT` 後能連網，再分項測試功能。
先更新韌體	綁定後先進 `AI.AGENT`，它會自動下載新版並重開。失敗時一樣會出現臉的畫面，就回主選單進 `SETUP` → `Firmware update` 再試；還是失敗就交給講師。更新中不要拔線、按鍵或開 monitor。

找不到裝置時

先確認它仍在等待配對、手機藍牙已開啟、App 已取得必要權限。記下是「清單找不到」還是「選到後失敗」，再請講師協助。不要反覆解除綁定或重燒韌體。

驗收紀錄

☐ ID 比對正確　☐ App 設定完成　☐ Wi-Fi 成功　☐ 功能另行驗證

目前畫面／錯誤文字：_____

不用猜：指給同學看，再說明用途

兩人一組。一人指認、一人核對課本，完成後交換。需要操作按鍵或改接線時，先依講師指示；辨認插孔不需要拆機。

任務	寫下位置或用途	同學核對
找到電源鍵和 RST	它們分別在： 短按／長按差異：	☐ 正確
找到兩個 USB-C	課堂優先使用： 原因：	☐ 正確
找到 microSD 和 PORT.A	哪個放記憶卡？ 哪個接擴充模組？	☐ 正確
找到 PORT.B、PORT.C 與三區觸控	螢幕觸控與機身觸控的差別：	☐ 正確
找到相機、麥克風和喇叭	兩項輸入： 一項輸出：	☐ 正確

把三種「成功」分清楚

螢幕亮起、電腦出現 USB 裝置、AI 能回答問題，各自代表什麼？哪一項需要網路服務？

過關方式

能指對兩個按鍵、兩個 USB-C、microSD 與三個 Grove 接口；能說出至少兩項輸入、兩項輸出；能說明為什麼 USB 辨識不等於程式功能正常。

AI 與機器人如何分工？

目標：分辨寫程式時的 AI、作品執行時的 AI，以及設備各自負責什麼。StackChan 裡的 ESP32 執行韌體，處理螢幕與按鈕等設備工作；它不等於電腦上協助寫程式的 AI。

一個需求的旅程

想像你說：「請提醒我休息。」作品若將來接上 AI 和受控工具，流程可以是：理解需求 → 選擇允許的工具與參數 → 服務檢查 → 設備執行並回報 → 根據結果回答。檢查失敗或沒有回報時，應停下並說明，不能猜成功。

實機例子（10-28，原廠 1.6.0）：說「十分鐘後提醒我喝水」，AI 呼叫 `create_reminder`（600 秒、「该喝水啦！」）；問「有哪些提醒」呼叫 `get_reminders`；說「取消」呼叫 `stop_reminder`。一分鐘提醒到時，螢幕跳出 reminder 畫面。營隊自己的受控工具服務仍待整合。

跟機器人說話前

先點螢幕或說「Hi, StackChan」叫醒它，等頭頂綠燈亮了再說。說「你好」叫不醒它。大約一分鐘沒人說話，它會說再見、綠燈熄掉，要再點一次螢幕。

動手畫

在紙上畫出「使用者 → AI → 服務 → StackChan → 回報」，替每一段寫一個工作。再圈出可能失敗的一段，寫下使用者應看到什麼。

完成條件：能說出兩種 AI 的不同用途，並指出沒有設備回報時的狀態。
想一想：指令送出後，為什麼還需要回報？

兩天的任務地圖

目標：知道每個時段要做什麼，並準備可用的帳號、電腦和練習資料夾。依講師提供的最新版指引登入與開啟工具；畫面可能更新，不必死背按鈕位置。未成年使用服務前，先依服務規定及課程安排取得必要同意。

第一天

- 09:00–10:30　`d1-desktop-setup`：開啟本機專案、建立規則與角色網頁。
- 10:45–12:00　`d1-task2`：改一個畫面常數，編譯並依課前驗證狀況操作。
- 13:00–14:30　`d1-tool-map`：把需求拆成工具、參數和回報。
- 14:45–16:30　`d1-tool-service`：用固定範例觀察檢查與回報。

第二天

- 09:00–10:30　`d2-tool-connect`：固定指令或標示清楚的模擬。
- 10:45–12:00　`d2-tool-change`：改一項功能並比較。
- 13:00–14:30　`d2-tool-test`：測正常、離線與錯誤設定。
- 14:45–16:30　`d2-tool-show`：展示證據與限制。

每天 12:00–13:00 午休，16:30–17:00 回顧。講師課前需確認帳號、使用額度、網路、資料線、Python、ESP-IDF 和練習版本；不要把密碼或金鑰貼進 AI 對話。**完成條件：**能打開指定資料夾並指出兩天的產出。**提問：**哪些活動需要講師先完成實機驗證？

先讀對檔，再動手

目標：讓 AI 知道專案位置、規則與任務範圍。README.md 是專案說明；AGENTS.md 是給協作工具的工作規則。先確認開啟的是講師準備的 stackchan-camp，不要在陌生資料夾執行修改。

先看，再請它解釋

```
請先讀 AGENTS.md、firmware/README.md 和
firmware/main/apps/app_camp_lab/app_camp_lab.cpp。
用高中生能懂的話說明這個 App 如何開啟、更新與關閉。
列出你讀到的檔案和這次不能修改的範圍；先不要改檔。
```

檢查回答有沒有列出正確檔名、提到畫面常數，以及不修改 Wi-Fi、馬達和開機流程。若找不到檔案，先請講師確認版本與路徑；不要讓 AI 自行猜一個檔案代替。

寫一條自己的規則

例如：「每次只改一件事，先列出修改差異。」若專案已有 AGENTS.md，先閱讀並請講師協助合併，不要覆蓋。把你希望 AI 遵守的另一條規則寫下：

_____

完成條件：能指出 AI 實際讀過的檔案，並說出一項禁止修改的範圍。**提問：**AI 的解釋與程式不一致時，要相信哪個證據？

一次只改一個顏色

目標：在講師建立的 Camp Lab 基線上，請 AI 修改單一常數，再自己讀懂差異。Camp Lab 是教學 App 補丁：目前只有編譯通過，尚未完成燒錄與實機操作驗證。

把需求說清楚

```
只修改 firmware/main/apps/app_camp_lab/app_camp_lab.cpp。
把 kHappyColor 改成 0x365C8D，保留其他行為。
不要修改 Wi-Fi、後端位址、帳號、馬達、分割表或開機流程。
完成後列出差異，告訴我螢幕應該出現什麼變化。先不要燒錄。
```

在專案根目錄執行 `git diff`：減號是原本內容，加號是新內容。確認變動只涉及要求的顏色常數；有其他變動就先停下，和講師一起檢查。不要用清除或重設指令處理不明變更。

預測與觀察

預測按 `Change mood` 進入開心狀態時，背景會變為指定色。先把預測寫下；真正畫面仍需講師允許並完成燒錄後才能觀察。

預測：_____

完成條件：能指認修改的檔案、舊值和新值。**提問：**為什麼編譯成功還不能證明畫面真的變色？

從程式到實際畫面

目標：分別辨認編譯、燒錄與設備觀察。以下操作以講師事先驗證的 ESP-IDF 5.5.4 環境及準備好的專案為前提。若尚未驗證或未獲講師允許，停在編譯或閱讀步驟，不要自行燒錄。

在 ESP-IDF 終端機

```
cd firmware
idf.py build
python -m serial.tools.list_ports
```

確認 build 無錯誤且成功結束；它只證明程式可編譯。使用支援資料傳輸的線接到底座 USB-C，比對連接前後新增的埠並請講師確認。埠號因電腦不同而異，不要照抄別人的。

```
idf.py -p SERIAL_PORT flash
idf.py -p SERIAL_PORT monitor
```

把 `SERIAL_PORT` 替換為確認過的實際埠號。燒錄成功後開啟 `CAMP LAB`，按 `Change mood`、`Exit`，重開再試。monitor 需講師已設定 USB console，按 `Ctrl + ]` 離開；沒有 log 不等於燒錄失敗。一次只讓一個程式占用連接埠。**打開 monitor 時機器人會重新開機**，畫面回到開機流程，這是正常現象，不是當機。

> 10-28 實機已用 USB 讀到原廠韌體的開機紀錄（App 1.4.4、ESP-IDF v5.5.4），並確認開啟序列埠會觸發重開。Camp Lab 仍只有 build 通過，尚未燒錄或驗證畫面；配對仍待完成。

完成條件：分別記下 build、flash 與畫面結果；沒做的寫「未測」。**提問：**哪一筆證據來自真實設備？

做一個可停止的提醒

目標：用筆電 Python 模擬計時；這不是 StackChan 韌體，也不會控制設備。建立 `reminder.py`，貼上完整程式後執行 `python reminder.py`。若電腦使用 `python3`，依講師環境改用該指令。

```
import queue
import threading
import time
commands = queue.Queue()
def read_input():
    while True:
        try:
            commands.put(input())
        except EOFError:
            commands.put("stop")
            return
threading.Thread(target=read_input, daemon=True).start()
deadline = None
print("輸入秒數設定；status 查詢；stop 結束")
while True:
    try:
        command = commands.get(timeout=0.1).strip()
        if command == "stop":
            print("已停止")
            break
        if command == "status":
            print("未設定" if deadline is None else "倒數中")
            continue
        seconds = int(command)
        if not 1 <= seconds <= 3600:
            raise ValueError
        deadline = time.monotonic() + seconds
        print("已設定", seconds, "秒")
    except queue.Empty:
        pass
    except ValueError:
        print("請輸入 1 到 3600 的整數")
    if deadline is not None and time.monotonic() >= deadline:
        print("提醒：休息一下！")
        deadline = None
```

輸入 3、status，觀察提醒後再次查詢；再輸入錯誤值與 stop。**完成條件：**看到設定、提醒及停止訊息。**提問：**這個結果證明了機器人會提醒嗎？

設定值通過，不代表設備完成

目標：辨認工具名稱、參數與回報。工具是系統允許使用的一項功能；參數是交給工具的設定。以下名稱只是紙上設計，並非已實作的 API。

把需求翻成一張卡

需求：「請顯示開心表情。」範例工具 `set_expression`，參數 `expression: happy`。服務應先確認操作對象屬於使用者，再檢查表情是否在允許清單中。這些是未來整合時需要建立的限制，目前不能假設已具備。

服務結果

設定有效、拒絕無效值，或送出後等待回覆；每一種結果都應清楚標示。

設備結果

有可信的設備回報與觀察，才可說設備完成；逾時或沒有回報寫「未確認」。

練習判斷

服務顯示「指令已送出」，螢幕仍看不到，設備也沒有回報。請寫一段給使用者的回答：

_____

固定範例能練習資料流，不能證明真 AI 已選擇工具或實機已執行。**完成條件：**分得出參數檢查與設備回報。**提問：**如果設備離線，應該如何回答？

設計一個健康習慣提醒

目標：把生活中的需要寫成可測試的提醒。從久坐、喝水、用眼中選一項；提醒由使用者自行決定，本活動不填寫真實健康紀錄，也不提供醫療診斷、劑量或療效宣稱。

需求卡

我選的習慣：☐ 久坐　☐ 喝水　☐ 用眼

使用者在什麼情況可能需要提醒？_____

提醒的時間或條件：_____

提醒文字：_____

使用者如何確認、延後或停止？_____

從第 6 課本機程式開始，可調整提醒文字與測試秒數；它目前只有設定、查詢與停止，尚未有延後或裝置觸控。原廠 AI 已能用語音在機器人上設提醒，可先拿它比較兩者差異。把未完成的功能畫在卡上，標為「待實作」。

> 遇到「我有症狀，該怎麼治療？」這類越界問題，示範回答應停止診斷或給藥建議，請對方找可信任的醫療專業人員；有緊急狀況時尋求當地緊急協助。

完成條件：同學讀完卡片，能說出何時提醒以及如何停止。**提問：**如果提醒打斷使用者，該怎麼修改設計？

互換角色，試用提醒

目標：觀察提醒是否符合使用情境，而不是替同學判斷健康狀況。兩人一組：一人設計提醒，另一人扮演使用者；交換角色後再試一次。課堂可以使用本機模擬或紙卡演練。

虛構角色卡（草稿）

小安正在準備報告，常常專心到忘記站起來；上課中不方便聽到大聲提示。小安想自己選提醒間隔，也希望忙的時候可以暫停。這是虛構情境，不代表任何真實病人的反應。

三步試用

1. 設計者說明提醒何時出現，請使用者設定一個測試用短間隔。
2. 使用者口述當下感受，嘗試確認或停止；未實作的按鈕用紙卡模擬並標記。
3. 交換角色，記下具體困難，提出一個改動並再試。

使用者說了什麼？_____

我觀察到什麼？_____

下一版只改一件事：_____

不要把一次課堂試用推論成病人的臨床依從性；若未來使用醫療情境，須另請專業人員審閱。**完成條件：**留下觀察與一項可測的改善。**提問：**使用者不想被提醒，可能有哪些不同原因？

四種情況，四筆證據

目標：先寫預期，再執行測試並區分資料來源。以下是待整合系統的測試設計；沒有相應功能時以紙上或講師提供的固定模擬演練，不能當作實機通過。

情況	輸入／條件	預期	實際與證據
成功	允許的參數、設備可用	取得設備完成回報，並核對畫面	_____
離線	設備不可達	失敗或未確認，不說成功	_____
無效參數	不在允許清單的值	拒絕，說明原因	_____
越界	請求醫療診斷或劑量	不診斷、不給劑量，建議找專業人員	_____

填寫方式

每一列標示「實機／本機模擬／紙上／未測」，再記下實際顯示的文字或你觀察到的畫面。預期與實際不同也要照實寫，不要為了通過而改結果。

第 6 課程式可測整數範圍：1、3600 為可設定邊界；0、3601 和文字應被拒絕。這只驗證本機程式的輸入檢查，不等於設備工具測試。

完成條件：四列都有預期與狀態，並至少完成一項可執行測試。**提問：**哪一列需要設備真正回報才能判定成功？

卡住時，先縮小問題

目標：用可重現的觀察請求協助。先記下你在哪個資料夾、做了什麼、預期與實際結果，再從最靠近問題的一步檢查。不要把密碼、金鑰或私人裝置 ID 貼到對話或紀錄。

依症狀檢查

- **找不到檔案：**確認講師提供的專案和路徑，請 AI 列出它讀到的檔案；不要自行猜路徑。
- **build 失敗：**確認 ESP-IDF 環境、版本與目前目錄；保留第一個相關錯誤，找講師核對依賴。
- **找不到埠：**檢查資料線、底座連接、權限；比對連接前後清單，請講師協助。
- **燒錄或 monitor 失敗：**確認埠與是否被其他程式占用；USB console 需事先設定，沒有 log 不能直接推論設備壞了。
- **畫面與預測不同：**確認燒錄的是哪個版本、按了什麼、設備實際顯示什麼。

向 AI 或講師描述

「我在_____執行_____；預期_____，實際_____。請先解釋可能原因與安全的檢查步驟，不要自行燒錄或刪檔。」

完成條件：能描述重現步驟與一個下一步檢查。**提問：**如果錯誤訊息含敏感資訊，分享前應怎麼處理？

展示成果，也展示限制

目標：讓觀眾看得懂需求、操作和證據。準備約兩分鐘，用一句話介紹對象與習慣，再示範你的本機模擬或經講師驗證的實機流程。演示前先標示來源，不用把尚未接好的功能說成完成。

1. **需求：**誰在什麼情況需要提醒？使用者如何停止？
2. **修改：**展示一個程式或設計變更，以及修改前後差異。
3. **正常：**演示設定和結果；指出畫面、紀錄或設備回報。
4. **失敗：**展示無效輸入、離線或越界情境，說明實際回覆。
5. **下一步：**說清楚哪些是模擬、哪些未驗證，以及接下來要測什麼。

我的展示稿

我想幫助_____，在_____時收到_____提醒。

我修改了_____；證據來源是_____。

失敗時系統顯示_____；目前限制是_____。

如果設備沒有回報，就展示「未確認」並解釋原因。原廠 AI 的語音提醒已實機通過；營隊自訂工具與提醒韌體仍待整合，真實可用範圍以講師課前驗收為準。

完成條件：觀眾能指出一項修改、一項測試與一項限制。**提問：**哪個證據最能支持你的主張？

把發現留下來

目標：留下別人能理解並重做的紀錄。記版本和測試條件即可，不記私人裝置 ID、帳號、密碼、金鑰或同學的健康資料。

日期：_____　組別／作品代號：_____

我想解決的問題：_____

本次只修改：_____

專案／範本版本：_____

操作步驟：_____

_____

預期結果：_____

實際結果：_____

證據來源：☐ 實機　☐ 本機模擬　☐ 紙上　☐ 未測

測試狀態：☐ 通過　☐ 失敗　☐ 未確認　☐ 未測

遇到的問題與下一步：_____

_____

「未確認」也是有價值的結果：它告訴下一位測試者哪裡還需要證據。每次改一件事，就新增一筆紀錄，方便比較。**完成條件：**交換紀錄後，同學能說出你做了什麼及結果來源。**提問：**下一次測試要保持哪些條件相同？

課前驗收與參考資料

目標：讓課堂依實際可用的環境進行。每班開始前記錄版本、時間、操作者與結果；失敗或未測的項目留空間補證據，不用歷史 build 代替實機驗收。

☐ 帳號可用、未成年同意與使用額度已確認；登入資料不寫入教材。

☐ 電腦、網路、Python、ESP-IDF 5.5.4、資料線與學員基線已逐台試過。

☐ 每台先更新原廠韌體：綁定後進 AI.AGENT 自動更新；失敗改走 SETUP → Firmware update；再失敗就把機器靠近 AP 或改用課堂 2.4 GHz 熱點，最後交給 AI 以 USB 燒錄（見下方）。完成後記下版本。

☐ 保留已驗證、符合課堂設定的復原韌體；確認燒錄權限與實際埠。

☐ 實機驗證 Camp Lab 開啟、變色、連按、退出、重開及 USB monitor；未通過就改用清楚標示的模擬。

☐ 核對真 AI、受控工具、提醒韌體、配對後端是否整合並逐項測試；未接通不得宣稱可用。

☐ 準備角色卡、測試表、停止方式及越界回答；醫療情境草稿另請專業人員審閱。

> 目前已知（10-28 實機）：開機、馬達測試、App 綁定、Wi-Fi、校時、語音對話與 AI 轉頭／拍照工具正常；自動更新與 SETUP 更新因 Wi-Fi 訊號弱逾時，改請 AI 以 USB 燒錄成功更新到 1.6.0，綁定保留。開 monitor 會讓機器重開。Camp Lab 於 10-14 完整 build 通過，尚未燒錄實測。以上不是本班驗收結果。

熱點與換 AP 都失敗：請 AI 以 USB 燒錄

講師核准後，把下面這段交給 Codex 或 Claude，並在旁邊看它每一步的輸出。燒錄期間不要拔線或按鍵。

```
StackChan 原廠韌體 OTA 下載失敗，請用 USB 更新到官方版本：
1. 只用 VID 303a 的序列埠；先列出埠號給我確認。
2. 下載官方 OTA 檔，用 esptool image-info 確認 ESP32-S3、
   stack-chan、版本號與 checksum／hash 都 valid。
3. 先 read-flash 備份整顆 16MB，再只寫 ota_0（0x20000），
   清除 otadata（0xd000，長度 0x2000）；不要動 nvs 和 assets。
4. verify-flash 後重開，從開機 log 回報 App version。
任何一步失敗就停下來回報，不要換別的做法。
```

10-28 實機通過：和 OTA 一樣只換程式，綁定與 Wi-Fi 保留。分割位址取自本機，換版本前先讓 AI 重新讀分割表核對；備份含 Wi-Fi 密碼，只留講師電腦。

參考

營隊材料：`labs/stackchan-camp-lab/README.md`、`VALIDATION.md`、`course-site/content.js`。原始韌體：StackChan 基線；ESP-IDF 5.5.4 官方文件。

完成條件：每項已記錄通過、失敗或未測並安排替代方案。**提問：**哪個環節還缺真實設備證據？

這本課本從哪裡來？

硬體事實以 M5Stack 官方資料為依據，先用官方知識查詢工具核對，再查看產品各視角圖片。原創線圖用於教學辨識，簡化了外殼曲面、孔位間距與元件外形；接線仍須依實物標示與官方腳位資料。

1. M5Stack StackChan 官方產品與操作說明

   K151／K151-R，按鍵、USB、首次啟動、配對、規格、Grove 與原廠功能。

2. M5Stack CoreS3 官方資料

   主控、顯示、相機、音訊與感測器。整機電池以 StackChan 的 550 mAh 為準。

3. TCUnion StackChan 韌體固定基線

   課程 Camp Lab 以此基線新增，完整操作見隨附的 camp-lab 教材與其 VALIDATION.md。

4. Espressif ESP-IDF 5.5.4 官方文件

   建置、USB 燒錄與 console 設定。工具和相依套件須由講師事先驗證。

5. StackChan World：台灣 App Store ／ Google Play

   課本 QR 與投影頁直接連到這兩個商店，不經第三方 QR 服務。

本書的圖片

本書 8 張教學圖由課程團隊以程式自行繪製為 SVG：正面辨識、按鈕插槽、頂部功能、按鍵比較、系統分工、USB 接線、首次開機及 Stack-chan 開源家族。不是擷取原廠照片加字。QR 由標準編碼工具產生，不以繪圖方式猜畫。

進階教材提到的開源專案

專案名稱、簡述與授權整理在全書最後一頁「參考資料」。

目前版本的邊界

這是第一版課本，完成圖文、練習與講師驗收流程。開機、馬達測試與 USB 開機紀錄已有實機紀錄；配對、語音、營隊提醒韌體與裝置工具仍待課前驗證。程式可以在筆電練習，尚未接通的設備功能不可標為實機完成。

查證與編寫日期：2026-10-27；實機紀錄更新 2026-10-28。App、韌體與服務可能改版，講師每梯次須重新核對畫面及功能。後續修訂先更新實驗紀錄與驗收狀態，再更新本書版本。

認識 GitHub：別人的程式放在哪裡

目標：看懂一個 GitHub 專案頁，判斷它能不能拿來用。GitHub 是放程式碼的網站，大家把專案公開在上面，任何人都能讀、下載，照授權規定修改。StackChan 的很多玩法都是社群在這裡分享的。

專案頁上的名詞

名詞	意思	你可以怎麼用
repository（repo）	一個專案的資料夾，網址是「作者／專案名」	例如 `stack-chan/stack-chan`
README	專案首頁的說明書	先讀它：做什麼、怎麼裝、有什麼限制
commit	一次存檔紀錄，有編號與說明	指定編號，大家拿到的程式才會一樣
fork	複製一份到自己帳號再改	改壞了也不影響原作者
issue	問題回報區	看別人卡在哪、怎麼解決
LICENSE	授權條款	決定你能不能拿來改、要不要標出處

三種常見情況

- **MIT、Apache-2.0：**可以使用和修改，要保留原作者的授權聲明。
- **AGPL-3.0：**拿它的程式碼改了再提供給別人用，就要把你的版本也公開。課程裡只參考它的做法，不複製程式碼。
- **沒寫授權：**預設不能拿來改，只能閱讀。

安全規則

API 金鑰、Wi-Fi 密碼、帳號不能放進 repo，公開以後就收不回來。下載別人的程式，先讀 README 和授權，燒進機器人之前請講師確認。

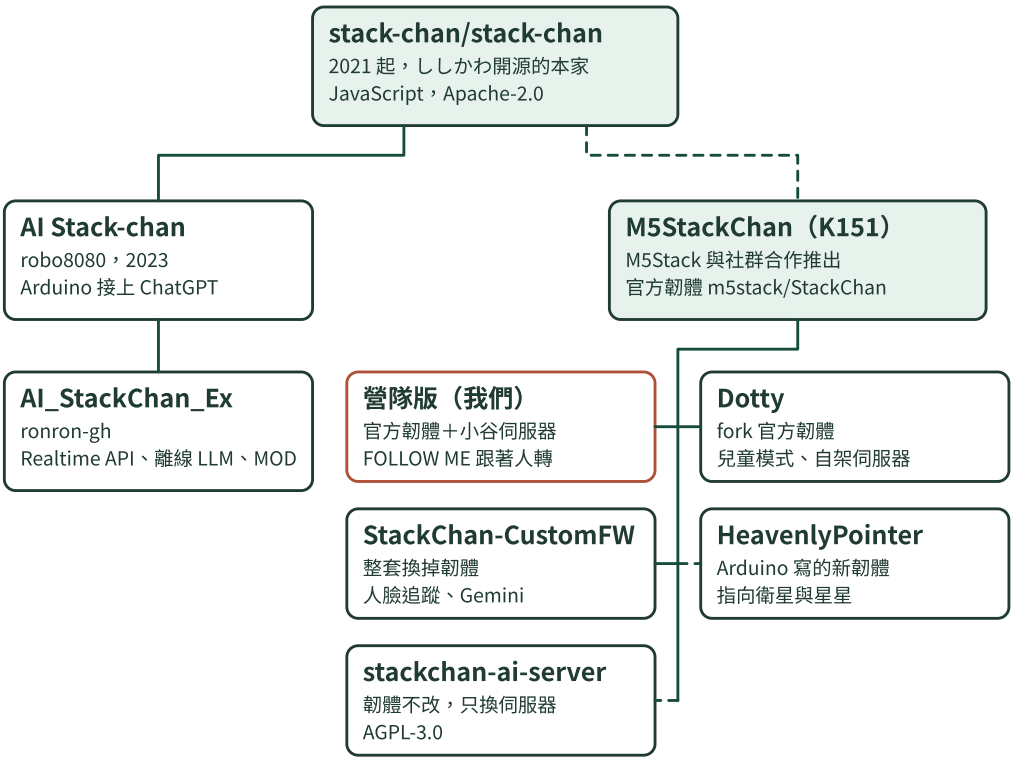
讀一個專案頁

打開 `github.com/stack-chan/stack-chan`，找出：它用什麼程式語言？授權是什麼？README 第一步要你做什麼？

語言_____　授權_____　第一步_____

Stack-chan 家族地圖

目標：看懂一個開源專案怎麼長出各種版本。Stack-chan 最早是ししかわ在 2021 年開源的手掌型機器人。後來有人接上 ChatGPT，M5Stack 也和社群合作推出現成的 M5StackChan，就是你手上這台。



實線：拿前一代的程式碼改出來。虛線：合作推出、沿用同一台硬體或同一套通訊方式。

專案	換掉了什麼	授權
營隊版	官方韌體加自製 App；伺服器改成營隊自架的小谷	官方原始碼標示 MIT
Dotty	fork 官方韌體，伺服器全部自架，預設兒童模式	MIT
StackChan-CustomFW	韌體與伺服器整套換掉，加人臉追蹤	MIT
HeavenlyPointer	用 Arduino 重寫韌體，改去指向天空	MIT
stackchan-ai-server	韌體不動，只把伺服器換成自己的	AGPL-3.0

找出共同點

挑兩個專案比較：哪一個改韌體，哪一個改伺服器？你要做自己的版本，會改哪一部分？_____

_____

同一台機器人，改指向天空

目標：看懂「韌體決定機器人做什麼」。HeavenlyPointer 是社群開發者 r3dfish 寫的 StackChan 韌體，MIT 授權。它用時間、所在位置和公開的軌道資料，算出衛星、飛機、行星、彗星在天空的方位，再把頭轉過去。螢幕上會顯示目標名稱、方位角和仰角。

它怎麼知道要轉到哪裡？

1. 連上 Wi-Fi，同步時間，用網路位置估算自己在哪裡。
2. 下載衛星軌道資料存進機器人裡；行星和彗星直接用公式算，不必上網。
3. 告訴它臉朝哪個方位（北、東、南、西），或用內建磁力計自動判斷。
4. 每秒算一次目標的方位角和仰角，換算成兩顆馬達的角度。

燒錄前先知道

燒這套韌體會整個蓋掉營隊韌體，只能由講師示範。示範完要燒回營隊版；萬一出問題，用 M5Stack 官方文件 StackChan 頁「Factory Firmware」一節的 M5Burner 步驟燒回原廠韌體（docs.m5stack.com/en/StackChan）。頭部俯仰要留在 5°–85°，水平不要撞到底，長時間追蹤要插 USB 供電。

觀察與比較

講師切到「Planets」或「Moon」時，用手機的星空 App 對照，看機器人指的方向對不對。寫下一個你覺得會影響準確度的原因：

_____

原始專案：github.com/r3dfish/HeavenlyPointer（v1.2.0，提交 `55d2970`）。示範影片：dsrc12〈StackChan Part3: Heavenly Pointer〉youtube.com/watch?v=265R1aMZRtU。

完成條件：能用自己的話說出四個步驟中哪些需要網路。**提問：**如果它不知道自己在哪裡，會指錯多少？

FOLLOW ME：跟著人轉的機器人

目標：讀懂一段真實的程式，再改一個數字觀察差別。營隊韌體有一個 FOLLOW ME App：鏡頭找到前方的人臉，頭就跟著轉。它和語音 AI 在同一套韌體裡，不必換韌體。已在實機確認可以跟著人轉（營隊版 1.0.10），1.0.12 轉頭更平順（2026-10-29）。

它怎麼運作

1. 鏡頭拍一張 320 × 240 的小圖，交給 Espressif 的人臉偵測模型（esp-dl），模型回報臉框和分數。
2. 取最大的臉，算它離畫面中央多遠；離越遠，頭轉越多。
3. 4 秒沒看到人，頭回正，放開馬達休息。

畫面只在機器人裡計算，不存照片、不上傳，也不認是誰。鏡頭對著同學之前，先問對方同不同意。

工程師日誌：八個版本才轉對方向

第一版用「膚色」找人，結果機器人去看燈，因為暖色燈光跟膚色很像。換成人臉偵測後，又發現鏡頭畫面左右是鏡像的，像手機自拍。每一版都燒進機器人、看 log、只改一個地方。從錯誤裡找證據，就是除錯。

找出不順，只改一個數字

先慢慢左右移動：頭是平順地跟，還是一頓一頓？鏡頭每秒只找臉約 8 次，想想每找到一次馬達會怎麼動。

從 follow_me_config.h 選一個：FOLLOW_SMOOTH（多平順，0～1）、FOLLOW_KP_YAW（轉的速度）、FOLLOW_LOST_MS（多久沒看到人就回正）、FOLLOW_FACE_SCORE_MNP（多有把握才算臉）。先寫預測，講師核准後燒錄觀察。

我看到_____；我改了_____，從____改成____；預測_____；實際_____

完成條件：記下改前改後各一筆觀察，標明「實機」或「未測」。**提問：**如果把 FOLLOW_X_DIR 改成相反的數字，會發生什麼事？

從一頓一頓到順順跟著：PID 微調

目標：照工程師真正的做法，一次改一個地方，用觀察決定下一步。下面是營隊版 FOLLOW ME 的微調紀錄。

版本	看到什麼	改了什麼
1.0.11	會跟，但頭一頓一頓	（找原因）
1.0.12	比較順	每次找到臉，馬達原本都從靜止重新起步，改成保留原本的速度；臉的位置取平均，不被一次抓偏帶走
1.0.13	又比 1.0.12 更順	改用 PID 控制

PID 是什麼

想像騎腳踏車追前面的同學。每次看一眼，決定這次踩多用力：

- **P（比例）：**離得越遠，踩越用力。只用 P，常常衝過頭。
- **I（累積）：**一直差一點追不上，就慢慢加力。加太多會晃來晃去。
- **D（變化）：**距離正在變小就先放慢，正在變大就加速。太大會抖。

機器人每看一次臉（約每 0.13 秒），就用 `KP × 偏移 + KI × 累積 + KD × 變化` 算這次頭要轉幾度。數字在 `follow_me_config.h`，預設左右 KP 60、KI 0、KD 40。

自己微調一次

每輪只改一個數字，請講師核准後燒錄，同一個動作測三次：快速走開、回到正中間停住。

1. KD 改成 0，只留 P：回中間會不會轉過頭？
2. KP 加大到開始左右晃，再退回一半。
3. KD 從 0 慢慢加，直到回中間不會轉過頭。

KP____ KI____ KD____：走開_____；回中間_____

完成條件：至少記兩輪，每輪都寫實際看到的結果。**提問：**為什麼一次只改一個數字？

Stack-chan 開源專案一覽

課程中提到的 GitHub 專案。想繼續玩，可以從這裡找下一步。要燒錄任何專案的韌體之前，先讀它的 README 和授權，並請講師確認還原方式。2026-10-28 查核。

專案	簡述	可以參考的地方	授權
stack-chan/stack-chan	本家，ししかわ 2021 年開源	瀏覽器模擬器、積木與臉部編輯器（實機安裝要換韌體）	Apache-2.0
m5stack/StackChan	M5StackChan 官方韌體，營隊版的基礎	官方文件「Factory Firmware」可燒回原廠韌體	原始碼標示 MIT
robo8080／M5Unified_StackChan_ChatGPT_Global	早期接上 ChatGPT 的版本（2023）	回答開頭加 (Happy) 標籤控制表情	MIT
ronron-gh／AI_StackChan_Ex	延伸上一個，加入即時語音、離線 AI	MOD 小程式架構；番茄鐘可當提醒參考	MIT
BrettKinny／dotty-stackchan	自架語音助理，預設兒童模式	兒童模式、相機隱私規則、AI 透明聲明	MIT
7H3-CH053N／StackChan-CustomFW	自製韌體與伺服器，有人臉追蹤	人臉追蹤做法；`make_eyes.py` 產生 21 種眼睛動畫	MIT
r3dfish／HeavenlyPointer	頭指向衛星、飛機、行星	進階 C 示範；軌道與方位計算	MIT
rudyll／stackchan-ai-server	韌體不改，只換伺服器	指令不清楚先問一句的對話設計（只參考做法）	AGPL-3.0
espressif／esp-dl（v2.0）	Espressif 的 AI 推論函式庫	FOLLOW ME 的人臉偵測模型	MIT

使用提醒

星星數多不代表適合你的機器人。先看支援哪種硬體、最後更新時間、需不需要金鑰或付費服務。