1. 數位輸出控制
   1. LED閃爍控制

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：讓LED 每隔\_秒閃爍一次。

程式需求：

1. 匯入需要的模組

2. LED腳位接在GPIO\_\_ (ESP32內建LED在GPIO2)

3. 將GPIO腳位和閃爍時間的變數獨立出來，方便日後修改

4. 程式碼須可直接在 ESP32 上運行

* 1. 控制多個LED

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：讓\_顆 LED 依序點亮，每顆亮\_秒後熄滅，循環重複。

程式需求：

1. 匯入需要的模組

2. 各LED腳位分別為GPIO\_、GPIO\_、GPIO\_、...

3. LED初始為熄滅狀態

4. 程式碼須可直接在 ESP32 上運行

* 1. 定時繼電器控制(版本1)

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：利用esp32內建的RTC計時，可以透過互動式命令列環境(REPL)設定時間，到達指定時打開繼電器。

程式需求：

1. 匯入需要的模組

2. 繼電器腳位接在GPIO\_，初始為OFF

3. RTC時間和電腦對時

4. 程式碼須可直接在 ESP32 上運行

* 1. 定時繼電器控制(版本2)

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者， 請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼， 功能是：利用 ESP32 內建的 RTC 實作一個非阻塞式的定時繼電器控制器。使用者可以透過互動式命令列環境 (REPL) 設定 RTC 時間與繼電器的觸發時間。時間檢查的任務需在背景運行，不可鎖定或阻塞 REPL 的操作。

程式需求：

1. 匯入 machine 與 time 模組。

2. 使用硬體計時器 (machine.Timer) 在背景中定期檢查時間，以實現非阻塞式監控，確保 REPL 始終保持互動性。

3. 繼電器控制腳位為 GPIO\_，並在程式啟動時，將其初始狀態設為關閉 (OFF)。

4. 提供清晰、可在 REPL 中呼叫的函式來進行以下操作：

* 同步 RTC 的日期與時間
* 顯示目前時間
* 設定繼電器觸發的目標時間 (時、分、秒)
* 取消繼電器觸發時間設定
* 啟動背景時間檢查
* 停止背景時間檢查

1. 數位輸入控制
   1. 按鈕控制LED

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：使用按鈕即時控制LED，當按鈕被按下時 LED 亮起，按鈕放開時 LED 熄滅。 程式需求：

1. 匯入需要的模組

2. 按鈕接在GPIO\_，LED接在GPIO\_

3. 程式碼須可直接在 ESP32 上運行

* 1. 按鈕切換控制LED

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼， 功能是：按鈕按下時切換 LED 的開關狀態（亮 ↔ 滅），並且加入機械彈跳的的處理(debouncing)，確保按鍵觸發穩定。

程式需求：

1. 匯入需要的模組

2. 按鈕腳位為GPIO\_、LED腳位為GPIO\_

3. 每次按鈕觸發時，切換 LED 狀態

4. 程式碼須可直接在 ESP32 上運行

* 1. 控制閃爍LED燈

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼， 功能是：透過按鈕開啟或關閉閃爍的 LED(頻率：亮3秒，暗1秒)。當按鈕按下時，LED熄滅；按鈕放開後，LED恢復閃爍。

程式需求：

1. 匯入需要的模組

2. LED腳位為GPIO\_，初始為閃爍狀態；按鈕腳位為GPIO\_。

3. 程式碼須可直接在 ESP32 上運行

* 1. 控制閃爍LED燈 - ISR

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼， 功能是：透過按鈕開啟或關閉閃爍的 LED(頻率：亮3秒，暗1秒)。當按鈕按下時，LED熄滅；按鈕放開後，LED恢復閃爍。

程式需求：

1. 匯入需要的模組

2. LED腳位為GPIO\_，初始為閃爍狀態；按鈕腳位為GPIO\_

3. 不要使用輪詢的方式偵測，用中斷方式(ISR)的方式處理

4. 程式碼須可直接在 ESP32 上運行

1. 類比輸入讀值
   1. 讀取類比輸入數值

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：讀取[感測器名稱]的類比電壓，並將數值輸出到REPL。

程式需求：

1. 匯入需要的模組

2. [感測器名稱]接在GPIO\_

3. 設定ADC量測範圍為 0~[1、1.34、2、3.3]V、解析度為[9、10、11、12]位元

4. 每\_秒更新讀取的數據

5. 程式碼須可直接在 ESP32 上運行

1. PWM輸出控制
   1. LED
      1. PWM控制LED閃爍

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：使用 PWM 控制LED，亮\_秒，暗\_秒。(備註：亮+暗秒數有限制，不能超過1秒)

程式需求：

1. 匯入需要的模組

2. LED腳位為GPIO\_

3. 只能靠調整 PWM 的頻率和佔空比(Duty)來模擬閃爍，且不能使用時間暫停或中斷的程式

4. 程式碼須可直接在 ESP32 上運行

* + 1. 呼吸燈

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：使用 PWM 控制LED，讓 LED 亮度從 0%漸變增至100%，再漸變減至0%，形成「呼吸燈」效果，無限循環。

程式需求：

1. 匯入需要的模組

2. LED腳位為GPIO\_

3. 程式碼須可直接在 ESP32 上運行

* 1. N20馬達
     1. 按鈕控制N20馬達轉動

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：按鈕按下時切換N20馬達的轉動狀態(靜止↔ 旋轉），並且加入機械彈跳的的處理(debouncing)，確保按鍵觸發穩定。

程式需求：

1. 匯入需要的模組

2. 按鈕腳位為GPIO\_，L9110S馬達控制模組腳位

為GPIO\_和GPIO\_

3. 程式碼須可直接在 ESP32 上運行

* + 1. 可變電阻模組控制N20馬達轉速

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：透過可變電阻模組控制N20馬達轉速。

程式需求：

1. 匯入需要的模組

2. 可變電阻模組腳位為GPIO\_，L9110S馬達控制模組腳位

為GPIO\_和GPIO\_

3. 可變電阻數值需配合PWM的數值範圍

4. 程式碼須可直接在 ESP32 上運行

* 1. 伺服馬達
     1. 設定伺服馬達依照指定角度旋轉

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：SG90伺服馬達在0到180度之間來回轉動，每隔\_秒旋轉45度。

程式需求：

1. 匯入需要的模組

2. SG90伺服馬達腳位為GPIO\_，初始角度為0度

3. 程式碼須可直接在 ESP32 上運行

* + 1. 可變電阻模組控制伺服馬達旋轉角度

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：可變電阻模組控制SG90伺服馬達旋轉角度，旋轉範圍0-180度。

程式需求：

1. 匯入需要的模組

2. SG90伺服馬達腳位為GPIO\_，可變電阻模組腳位為GPIO\_

3. 將可變電阻輸入的最小值和最大值對應500微秒至2500微秒之間來控制伺服馬達0至180度的角度

4. 程式碼須可直接在 ESP32 上運行

* 1. 蜂鳴器
     1. 電話鈴聲

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：使用無源蜂鳴器模擬家用電話響鈴鈴聲。

程式需求：

1. 匯入需要的模組

2. 蜂鳴器的腳位為 GPIO\_

3. 程式碼須可直接在 ESP32 上運行

* + 1. 播放旋律

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：使用無源蜂鳴器播放生日快樂歌，並可重複播放。

程式需求：

1. 匯入需要的模組

2. 蜂鳴器腳位為GPIO 15

3. 建立一個字典 `notes` 存放音符名稱與對應頻率(Hz)，包含休止符

4. 建立歌曲旋律與節奏資料結構：使用列表儲存(音符名稱, 時間長度秒)

5. 程式碼須可直接在 ESP32 上運行

1. Micropython內建模組可控元件
   1. WS2812B
      1. WS2812B顏色切換

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：讓WS2812B的LED燈條，依序顯示[顏色]。

程式需求：

1. 匯入需要的模組

2. WS2812B的LED燈腳位為GPIO\_，LED總數為\_顆

3. 亮度設為\_%，每個顏色顯示\_秒後切換到下一個顏色

4. 程式碼須可直接在 ESP32 上運行

* + 1. WS2812B顏色漸變

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：讓WS2812B的LED燈條，流暢地顯示全光譜的顏色。

程式需求：

1. 匯入需要的模組

2. WS2812B的LED燈腳位為GPIO\_，LED總數為\_顆

3. 亮度設為\_%。

4. 程式碼須可直接在 ESP32 上運行

* 1. DHT11
     1. 讀取DHT11溫、溼度值

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：讀取DHT11的數值，並顯示溫度(攝氏)和溼度(%)。

程式需求：

1. 匯入需要的模組

2. DHT11腳位為GPIO\_

3. 每\_秒更新數值

4. 程式碼須可直接在 ESP32 上運行

1. HC-SR04 超音波感測器
   1. 讀取超音波數值

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：利用HC-SR04超音波模組測量距離，單位為公分，顯示測量結果。

程式需求：

1. 匯入需要的模組

2. HC-SR04的Trig腳位為GPIO\_，Echo腳位為GPIO\_

3. 每\_秒更新數值

4. 程式碼須可直接在 ESP32 上運行

1. Max7219 8×8 LED
   1. 用8×8 LED顯示英文、數字

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：用Max7219模組驅動8\*8 LED矩陣顯示“D”、“o”、“g”，每秒依序切換每個字母，重複執行。

程式需求：

1. 匯入需要的模組

2. Max7219模組的CLK腳位為GPIO\_、CS腳位為GPIO\_、DIN腳位為GPIO\_

3. 程式碼須可直接在 ESP32 上運行

* 1. 用8×8 LED顯示英文、數字(參考外部模組)

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：用Max7219模組驅動8\*8 LED矩陣顯示“D”、“o”、“g”，每\_秒依序切換每個字母，重複執行。

程式需求：

1. 請分析並依照[\_模組](上傳給AI)或[上面的程式碼\_.py](貼上程式碼)提供的類別、函式、常數或資料來實現指定的功能。

2. Max7219模組的CLK腳位為GPIO\_、CS腳位為GPIO\_、DIN腳位為GPIO\_

3. 程式碼須可直接在 ESP32 上運行

* 1. 顯示自定義圖案、動畫

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：用Max7219模組驅動8\*8 LED矩陣顯示自定義的圖案，每\_秒依序切換每個圖案，重複執行。

程式需求：

1. 請分析並依照[\_模組](上傳給AI)或[上面的程式碼\_.py](貼上程式碼)提供的類別、函式、常數或資料來實現指定的功能

2. Max7219模組的CLK腳位為GPIO\_、CS腳位為GPIO\_、DIN腳位為GPIO\_

3. 有\_個自定義圖案，格式為[圖案1, 圖案2, 圖案3, ...]

4. 程式碼須可直接在 ESP32 上運行

* 1. 跑馬燈

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：用Max7219模組驅動8\*8 LED矩陣顯示“Dog”，且顯示時會由左向右移動(跑馬燈，Marquee)，重複執行。

程式需求：

1. 請分析並依照[\_模組](上傳給AI)或[上面的程式碼\_.py](貼上程式碼)提供的類別、函式、常數或資料來實現指定的功能

2. Max7219模組的CLK腳位為GPIO\_、CS腳位為GPIO\_、DIN腳位為GPIO\_

3. 程式碼須可直接在 ESP32 上運行

1. I2C - 1602 LCD
   1. 尋找I2C裝置位址

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：搜尋I2C的位址與裝置數。

程式需求：

1. 匯入需要的模組

2. I2C的SDA腳位為GPIO\_，SCL腳位為GPIO\_

3. 程式碼須可直接在 ESP32 上運行

* 1. 顯示文字

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：用I2C的1602 LCD模組顯示“Hello world!”。

程式需求：

1. 請分析並依照[\_模組](上傳給AI)或[上面的程式碼\_.py](貼上程式碼)提供的類別、函式、常數或資料來實現指定的功能。

2. I2C的SDA腳位為GPIO\_，SCL腳位為GPIO\_

3. “Hello”從第\_(0或1)列第\_(0-15)行開始顯示，“world!”從第\_(0或1)列第\_(0-15)行開始顯示

4. 程式碼須可直接在 ESP32 上運行

* 1. 文字跑馬燈

提示詞：

你是一位熟悉 ESP32 與 MicroPython 的專業開發者，請撰寫一段能在 ESP32 上執行的 MicroPython 程式碼，功能是：用I2C的1602 LCD模組顯示“Hello world!”，且文字會向左移動，具有跑馬燈(marquee)功能；左側消失的文字會重新從右側出現，重複執行：。

程式需求：

1. 請分析並依照[\_模組](上傳給AI)或[上面的程式碼\_.py](貼上程式碼)提供的類別、函式、常數或資料來實現指定的功能。

2. I2C的SDA腳位為GPIO\_，SCL腳位為GPIO\_

3. “Hello”從第\_(0或1)列第\_(0-15)行開始顯示，“world!”從第\_(0或1)列第\_(0-15)行開始顯示

4. 程式碼須可直接在 ESP32 上運行

* 1. 2