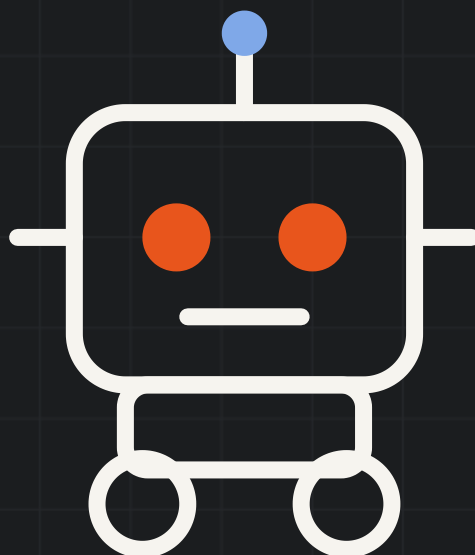


// TINYBOT FORGE · ESP32 ROBOTICS

ESP32 機器人 10 個專案

從一顆會呼吸的 LED，到手機遙控的機械手臂。
每一行程式都經過編譯驗證，每一步都附接線表。



這本書怎麼用

這本書有 10 個專案，難度由淺到深。專案 1、2 只需要 ESP32 和幾個小零件；專案 3 開始組一台小車，之後的專案都在這台車上加東西，所以**建議照順序做**。

每一章都有同樣的結構：**目標 → 零件 → 原理 → 接線 → 程式 → 程式重點 → 調整參數 → 挑戰題 → 常見問題**。程式可以直接從隨附資料夾開啟，不必從 PDF 複製。

安全提醒：ESP32 所有腳位都是 3.3V，輸入超過 3.3V 會損壞腳位。本書所有需要分壓的地方都有標示，請不要省略。18650 鋰電池請使用有保護板的電池盒，不要短路。

目錄

01	會呼吸的眼睛	LED、PWM 與按鈕	★☆☆☆☆
02	會揮手的桌上機器人	伺服馬達原理	★☆☆☆☆
03	避障小車	超音波測距與自主決策	★★☆☆☆
04	手機遙控車	Wi-Fi 熱點與網頁伺服器	★★☆☆☆
05	跟著手走的小車	比例控制（P 控制）	★★☆☆☆
06	循跡車	兩路紅外線與狀態判斷	★★★☆☆
07	三路循跡+PID	平滑又快的控制	★★★★☆
08	會追光的機器人	類比感測器與 Braitenberg 車	★★★★☆
09	超音波雷達	感測器+致動器+即時視覺化	★★★★☆
10	手機控制的機械手臂	多軸控制與動作錄製	★★★★★

開發環境（只需要設定一次）

1. 安裝 Arduino IDE 2（arduino.cc）。
2. 檔案 → 偏好設定 → 額外的開發板管理員網址，貼上：
`https://espressif.github.io/arduino-esp32/package\_esp32\_index.json`
3. 開發板管理員搜尋 **esp32**，安裝 Espressif Systems 版本 3.0 以上。
4. 工具 → 開發板選 **ESP32 Dev Module**，選對序列埠。抓不到序列埠：CP2102 晶片裝 Silicon Labs 驅動，CH340 晶片裝 WCH 驅動。

5. 上傳時出現 Failed to connect：按住板子上的 BOOT 鍵，直到出現 Connecting… 再放開。

注意版本：ESP32 套件 3.x 的 PWM 寫法是 `ledcAttach(pin, freq, bits)` 與 `ledcWrite(pin, duty)`。網路上很多舊教學用 `ledcSetup`／`ledcAttachPin`，在 3.x 會編譯失敗。

全書零件清單

零件	建議數量	用在哪些專案
ESP32 DevKit V1 (30 或 38 腳)	1	全部
Micro USB 或 USB-C 傳輸線 (要能傳資料)	1	全部
麵包板、杜邦線 (公對公、公對母、母對母)	各一包	全部
LED、220Ω 電阻、按鈕	各 2-4	1
SG90 伺服馬達 (可升級 MG90S)	2	2、9、10
HC-SR04 超音波感測器	1	2、3、5、9
1kΩ、2kΩ、10kΩ 電阻	各 2	2、3、5、8、9
L298N 馬達驅動板	1	3-8
兩輪車架套件 (含 2 顆 TT 馬達、萬向輪)	1	3-8
18650 鋰電池 ×2 與雙槽電池盒	1 組	3-8
TCRT5000 紅外線循跡模組	3	6、7
光敏電阻	2	8
電解電容 100-470μF	2	2、3、10
5V 2A 外接電源	1	10
黑色電工膠帶、白色海報紙	1	6、7

1

會呼吸的眼睛

LED、PWM 與按鈕

難度 ★☆☆☆☆

時間 約 30 分鐘

程式檔 p01-breathing-eyes.ino

做一對機器人眼睛：平常像在呼吸一樣慢慢變亮變暗，按下按鈕就眨兩下眼。這是之後所有專案的地基——學會「輸出」和「輸入」。

零件

零件	數量
ESP32 DevKit V1	1
LED（任意顏色）	2
220Ω 電阻	2
按鈕開關	1
麵包板、杜邦線	若干

原理

PWM（脈寬調變）：ESP32 的腳位只能輸出 0V 或 3.3V，沒辦法輸出「一半的電壓」。但如果每秒開關幾千次，開的時間佔 50%，LED 看起來就是一半亮。開的時間比例叫做**工作週期（duty）**。

ESP32 套件 3.x 用兩行設定 PWM：

```
ledcAttach(腳位, 頻率, 解析度);    // 解析度 8 bit → 亮度 0-255
ledcWrite(腳位, 亮度);
```

INPUT_PULLUP：按鈕一端接 GPIO、一端接 GND。開啟內建上拉電阻後，沒按時讀到 HIGH，按下時讀到 LOW。不用額外接電阻。

為什麼亮度要平方？人眼對亮度的感覺不是線性的：亮度從 0 到 50 看起來變化很大，從 200 到 250 幾乎看不出差別。把 0-1 的數值平方再乘上 255，呼吸效果會自然很多。

接線

從	接到	備註
GPIO 16	220Ω → 左 LED 長腳	LED 短腳接 GND
GPIO 17	220Ω → 右 LED 長腳	LED 短腳接 GND
GPIO 21	按鈕一端	按鈕另一端接 GND

程式

完整檔案在隨附資料夾：`p01-breathing-eyes.ino`。以下程式已用 Arduino ESP32 開發板套件 3.3.12 編譯通過（全部警告開啟，0 個警告）。

```
/*
 * TinyBot Forge 《ESP32 機器人 10 個專案》專案 1：會呼吸的眼睛
 * 兩顆 LED 當機器人的眼睛：平常慢慢「呼吸」，按下按鈕就眨眼。
 * 學到：數位輸出、PWM 亮度、INPUT_PULLUP 按鈕、millis() 非阻塞計時
 * 環境：Arduino IDE 2.x + ESP32 開發板套件 3.x
 */
const int PIN_EYE_L = 16;
const int PIN_EYE_R = 17;
const int PIN_BUTTON = 21;          // 按鈕一端接 GPIO21，另一端接 GND
const int PWM_FREQ = 5000, PWM_BITS = 8;

const unsigned long BREATH_MS = 3000; // 一次呼吸週期
const unsigned long BLINK_MS = 120;   // 閉眼時間

void setEyes(int brightness) {
  ledcWrite(PIN_EYE_L, brightness);
  ledcWrite(PIN_EYE_R, brightness);
}

void setup() {
  pinMode(PIN_BUTTON, INPUT_PULLUP);
  ledcAttach(PIN_EYE_L, PWM_FREQ, PWM_BITS);
  ledcAttach(PIN_EYE_R, PWM_FREQ, PWM_BITS);
}

void loop() {
  unsigned long now = millis();

  if (digitalRead(PIN_BUTTON) == LOW) { // 按下：眨兩下
    for (int i = 0; i < 2; i++) {
      setEyes(0);   delay(BLINK_MS);
      setEyes(255); delay(BLINK_MS);
    }
    while (digitalRead(PIN_BUTTON) == LOW) delay(10); // 等放開
    return;
  }

  // 呼吸：用正弦波讓亮度平滑變化；人眼對亮度是非線性的，平方後看起來更自然
  float phase = (now % BREATH_MS) / (float)BREATH_MS; // 0-1
  float s = (1.0 - cos(phase * 2.0 * PI)) / 2.0;      // 0-1-0
  int brightness = (int)(8 + 247 * s * s);
  setEyes(brightness);
}
```

```
delay(10);  
}
```

程式重點

- `millis() % BREATH_MS`：取得「現在在呼吸週期的哪個位置」，不需要 `delay` 卡住整個程式。
- `(1 - cos(x)) / 2`：產生 $0 \rightarrow 1 \rightarrow 0$ 平滑變化的曲線。
- 亮度最低設 8 而不是 0，眼睛不會完全熄滅，看起來更像「活著」。
- 按下按鈕後用 `while` 等使用者放開，避免按一次眨好幾次。

調整參數

想調整	改這裡
呼吸太快或太慢	<code>BREATH_MS</code> ，3000 = 3 秒一次
眨眼太快	<code>BLINK_MS</code>

挑戰題

1. 讓左右眼錯開半個週期呼吸，像在「眨單眼」。
2. 長按按鈕 2 秒切換成「生氣模式」：眼睛快速閃爍。
3. 改用 RGB LED，呼吸時顏色也跟著變化。

常見問題

LED 完全不亮

LED 有方向性，長腳（正極）要接電阻那一側。

按鈕按了沒反應

確認按鈕接的是對角的兩隻腳。四腳按鈕同一邊的兩腳內部是相通的。