

第 8 章 4：MCU 整合專案：智慧家居節點

整合 ESP32、感測器、控制、雲端的完整 IoT 專案。

一、學習目標

- 系統架構規劃
- 多周邊整合
- Wi-Fi / MQTT 通訊
- 低功耗管理
- OTA 韌體更新
- PCB 設計與量產

二、專案規格

智慧家居節點（單一裝置）：

功能	說明
環境監測	T、RH、CO ₂ 、PM2.5
燈光控制	12V LED 條 PWM
開關	4 路 Relay 控制
PIR	人體感測
顯示	1.3" OLED
連網	Wi-Fi + BLE
雲端	MQTT + Home Assistant
電源	USB-C 5V 或 12V
韌體	ESP-IDF + FreeRTOS
更新	OTA

三、硬體設計

主控

ESP32-S3 模組（內建 PSRAM、Wi-Fi、BLE）。

電源

```

USB-C 5V — ESD — 5V Bus — LD0 3.3V — ESP32
                        |
                        └ Buck 12V→5V[?][?][?] 12V [?][?][?]
  
```

Relay 驅動

```

ESP32 GPIO → ULN2003 / MOSFET → Relay [?][?][?][?][?][?][?][?][?]
  
```

LED Driver

```

ESP32 PWM → Gate Driver → N-MOSFET → 12V LED Strip
  
```

感測器

- BME680：T、RH、IAQ (I²C)
- SCD41：CO₂ 真實量測 (I²C)
- PMS5003：PM2.5 (UART)
- PIR：HC-SR501 (GPIO)

OLED

- SSD1306 1.3" (I²C)

USB-C

- TypeC 24-pin
- CC 電阻 5.1k 兩顆
- ESD 保護
- USB-Serial：透過 ESP32 內建 USB

四、PCB 設計重點

1. 天線淨空：模組天線下方無銅皮，保留淨空區
2. 電源接地分區：類比 / 數位 / 高電流分開
3. 散熱：Buck IC 焊盤打 thermal via 到底層銅皮
4. 保護：USB ESD、12V TVS、Reverse polarity
5. Debug：JTAG / SWD pad、UART debug pad、Reset / Boot 鈕
6. EMC：Y 電容、共模扼流圈（在 USB / 12V 入口）

五、韌體架構 (FreeRTOS Tasks)

```
xTaskCreate(sensor_task, "sensor", 4096, NULL, 5, NULL);
xTaskCreate(display_task, "disp", 4096, NULL, 4, NULL);
xTaskCreate(network_task, "net", 8192, NULL, 6, NULL);
xTaskCreate(control_task, "ctrl", 4096, NULL, 7, NULL);
```

Task	工作
sensor	每 5 秒讀感測器
display	每 1 秒更新 OLED
network	Wi-Fi 維護、MQTT 收發
control	處理命令、控制 Relay/LED

共享資料

用 FreeRTOS Queue 與 Semaphore 同步。

六、MQTT 主題設計

```
home/livingroom/sensor/temperature
home/livingroom/sensor/humidity
home/livingroom/sensor/co2
home/livingroom/sensor/pm25
home/livingroom/light/state
home/livingroom/light/set
home/livingroom/relay1/set
home/livingroom/online      ← LWT
```

MQTT 範例 (ESP-IDF)

```
esp_mqtt_client_config_t mqtt_cfg = {
    .broker.address.uri = "mqtt://192.168.1.10:1883",
    .credentials = {
        .username = "user",
        .authentication.password = "pass",
    },
};

esp_mqtt_client_handle_t client = esp_mqtt_client_init(&mqtt_cfg);
esp_mqtt_client_register_event(client, ESP_EVENT_ANY_ID, mqtt_event_handler, NULL);
esp_mqtt_client_start(client);
```

七、OTA 更新

ESP-IDF 提供 `esp_https_ota` 函式：

```
esp_https_ota_config_t ota_config = {
    .http_config = &config,
};

esp_err_t ret = esp_https_ota(&ota_config);
if (ret == ESP_OK) {
    esp_restart();
}
```

實作要點：- 雙 Bank (OTA0 / OTA1) - 簽章驗證 - 回滾機制 (boot 失敗自動還原)

八、低功耗模式

Deep Sleep

ESP32 < 10 μ A。透過 RTC 計時或 EXT0/EXT1 喚醒：

```
esp_sleep_enable_timer_wakeup(60 * 1000000ULL); // 60s
esp_sleep_enable_ext0_wakeup(GPIO_NUM_25, 1);   // PIR ??
esp_deep_sleep_start();
```

Light Sleep

維持 RAM、需要快速喚醒場景。

九、安全考量

- Wi-Fi WPA2/WPA3
 - TLS 加密 MQTT (Port 8883)
 - Secure Boot (防偽韌體)
 - Flash 加密 (保護 IP)
 - OTA 簽章
 - 不在程式碼中硬編密碼，用 Provisioning
-

十、與 Home Assistant 整合

MQTT 自動發現

ESP32 推送 config 訊息：

```
{
  "name": "Living Room Temperature",
  "state_topic": "home/livingroom/sensor/temperature",
  "unit_of_measurement": "°C",
  "device_class": "temperature"
}
```

到 `homeassistant/sensor/livingroom_temp/config`。

HA 會自動加入此實體。

十一、量產 / 開源

1. PCB 量產：JLCPCB / PCBA 服務
 2. 外殼：Fusion 360 + 3D 列印
 3. 認證：BSMI、CE、FCC (自願 / 強制依市場)
 4. GitHub 開源：吸引社群改進
 5. 安全更新長期維護
-

十二、進階變化

- Matter / Thread 取代 MQTT

- 更小尺寸：用 ESP32-C3 或 ESP32-H2
 - 太陽能 + 鋰電池
 - 加 LCD 觸控
 - AI 邊緣推論：環境異常偵測
-

十三、本專案重點

✔ 您學到：

- [x] 完整 IoT 系統設計
 - [x] FreeRTOS 多任務
 - [x] MQTT 通訊
 - [x] OTA 與安全
 - [x] 低功耗
 - [x] PCB 量產考量
-

下一專案

05-?????.md