



嶺東科技大學
LING TUNG UNIVERSITY

資訊管理系

魚樂園

指導教授：陳健忠 教授

組員名單：陳昱廷 B08C017

陳捷恩 B08C002

林秉驊 B08C011

王雨新 B08C048

柳承佑 B08C049

中 華 民 國 1 1 4 年 5 月



嶺東科技大學
LING TUNG UNIVERSITY

資訊管理系專題口試委員審定書

魚樂園

指導教授：陳健忠 教授

組員名單：陳昱廷 B08C017

陳捷恩 B08C002

林秉驊 B08C011

王雨新 B08C048

柳承佑 B08C049

指導教授：_____

口試委員：_____

中 華 民 國 1 1 4 年 5 月

謝 誌

本專題報告得以順利完成，首先要感謝恩師陳健忠老師細心引導我們，耐心的協助我們，克服研究過程中所面臨的困難，給予我們最大的協助，使本專題得以順利完成。

研究報告口試期間，感謝林孟源老師、陳元瓊老師不辭辛勞細心審閱，不僅給予我們指導，並且提供寶貴的建議，使我們的專題內容以更臻完善，在此由衷的感謝。

最後，感謝系上諸位老師在各學科領域的熱心指導，增進商業管理知識範疇，在此一併致上最高謝意。

陳昱廷
陳捷恩
林秉驊 謹誌
王雨新
柳承佑

中華民國114年05月於嶺東

摘要

隨著數位時代快速發展，資訊與設備的能力得以大幅提升，這正是物聯網技術的核心。不論是生活、醫療、交通，甚至軍事領域，物聯網都展現了顯著的效益。尤其是遠端監控技術的進步，用戶可隨時掌握室內外環境狀況，並迅速應對突發情況。本計劃主要目的將遠端監控技術應用於魚缸管理，打造一套智慧魚缸系統。結合自動化與物聯網技術，提供用戶即時掌握魚缸狀態的能力。系統功能包括即時影像監控、控制日光燈，以及整合溫度感測器，確保水質適宜。這些功能共同實現了一個高效能的魚缸管理方案。本專題使用 Arduino IDE 撰寫程式到 ESP32 開發板連接到溫感器、LED 和監視器，將數據傳輸到 LINE Bot 上。以達到顯示溫度、即時監控魚缸和控制燈光變化。

關鍵詞：物聯網、遠端監控、智慧魚缸管理

目 錄

摘 要	I
目 錄	II
圖目錄	II
第壹章 緒論	1
1.1 研究動機	1
1.2 研究目的	1
第貳章 文獻回顧與探討	2
2.1 Arduino 介紹	2
2.2 Arduino IDE	2
2.3 開發版的選擇	3
2.3.1 ESP32 DevKit V1	3
2.3.2 Arduino UNO	3
2.4 ESP32-CAM	4
2.5 溫度感測器	4
2.5.1 DS18B20防水溫度感測器	4
2.5.2 LM35 溫度感測器	4
2.5.3 功能比較	5
2.6 LED WS2812	5
2.7 LINE Bot	5
2.8 ChatGPT	6
第參章 研究方法	7
3.1 研究步驟	7
第肆章 系統分析與設計	13
4.1 摘要	13
4.2 系統架構	13
4.3 實作流程	14
4.4 操作步驟	19
第伍章 結論	22
參考文獻	23
附錄	24
輔導紀錄	31

表目錄

表2.1 功能比較.....	3
----------------	---

圖目錄

圖 2.1 Arduino.....	2
圖 2.2 LINE Bot.....	4
圖 2.3 ChatGPT.....	4
圖 3.1 甘特圖.....	5
圖 3.2 架構圖.....	6
圖 3.3 整體流程圖.....	7
圖 4.1 日光燈流程圖.....	12
圖 4.2 溫感器流程圖.....	13
圖 4.3 USB-TTL 跟 ESP32-CAM 連接流程圖.....	14
圖 4.4 日光燈圖.....	16
圖 4.5 DS18B20防水溫度感測器圖.....	16
圖 4.6 ESP32-CAM 圖.....	16
圖 4.7 LINE Bot 介面.....	18
圖 4.8 ESP32-CAM 畫面.....	19
圖 4.9 ESP32-CAM 畫面-2.....	19

第壹章 緒論

1.1 研究動機

隨著現代生活型態的改變與寵物飼養觀念的普及，尤其是年輕世代成為主要飼主，魚類作為寵物的飼養比例逐年提升。相比於傳統的貓狗寵物，魚類飼養具有空間需求較小、觀賞價值高等優勢，吸引越來越多忙碌的上班族選擇飼養。然而，現代多數飼主多為朝九晚五的上班族，日常工作繁忙且時間有限，導致無法隨時照顧家中的魚類寵物，這使得魚缸的日常管理和維護變得更加困難。

在此背景下，如何提升魚類飼養的便利性與管理效率，成為一項重要的課題。傳統魚缸缺乏即時監控與遠端管理功能，飼主無法隨時掌握魚缸內的環境狀況，容易忽略水質、水溫等關鍵因素，進而影響魚類的健康與生長。因此，結合物聯網技術與即時通訊工具，開發一套智慧魚缸系統，成為解決此問題的有效途徑。

基於上述需求，本專題設計並實作了一套智慧魚缸系統，透過 LINE Bot 平台實現遠端監控與控制功能。飼主可利用手機隨時查看魚缸的即時影像，遠端調控日光燈照明，並即時監測水溫等環境參數，確保魚缸環境維持在適合魚類生存的狀態。此系統不僅提升了魚類飼養的便利性，也大幅提高了管理的效率與精準度，為現代忙碌的飼主提供了一個智慧化、便捷性的飼養解決方案。

1.2 研究目的

本專題旨在設計並開發一套智慧魚缸管理系統，透過手機應用介面，讓飼主能夠隨時隨地即時掌握魚缸的各項狀況，無論身處家中或外出，皆能輕鬆且有效地照料魚類。系統的設計核心在於提升魚類飼養的便利性與管理效率，減少飼主的操作負擔，並確保魚群能在穩定且適宜的環境中健康成長。

具體而言，本系統主要包含以下功能：

- **即時監測水溫：**透過感測器持續監控魚缸水溫，確保水溫維持在魚類適合的範圍內，避免因環境變化造成魚群壓力或疾病發生。
- **遠端控制日光燈照明：**模擬自然光照週期，透過手機遠端調控燈光開關與亮度，減少飼主手動操作的需求，並促進魚類的正常生理節律。
- **即時影像監控：**提供清晰的魚缸即時影像，讓飼主能夠隨時觀察魚群的活動狀態與健康情況，及早發現異常並採取適當措施。

透過這套智慧系統，本專題期望為魚類飼養者打造一個便捷、智慧且高效的管理平台，不僅提升飼主的養殖體驗，也促進魚類的健康與福祉，進一步推動現代化寵物飼養的發展。

第貳章 文獻探討

2.1 Arduino 介紹

Arduino（發音 /ɑ:r'dwi:nou/）是一個義大利開源硬體與軟體平台，廣泛應用於數位裝置與物聯網系統的開發。Arduino 公司設計並製造多款單板微控制器，方便使用者快速建構各種智慧控制系統。

其硬體設計採用 Creative Commons CC BY-SA 授權，允許自由複製與改良；軟體則依據 GNU LGPL 或 GPL 授權，鼓勵開源共享。Arduino 板具備多種輸入輸出介面，方便連接感測器與致動器，降低開發難度。

本專題利用 Arduino 作為智慧魚缸系統的核心控制器，負責水溫監測、燈光控制及即時影像傳輸，展現其靈活且可靠的特性。文獻證明 Arduino 平台適合用於遠端監控與智慧管理，為本系統提供穩固的技術基礎。[1]



圖2.1 Arduino

2.2 Arduino IDE

Arduino 程式的撰寫與上傳主要透過 Arduino IDE（整合開發環境）完成。使用者只需將 Arduino 板透過 USB 連接到 Windows、Linux 或 Mac 電腦，利用 IDE 將程式編譯並上傳至板子中。這個過程簡單直觀，降低了硬體開發的門檻。Arduino IDE 也提供序列監控等工具，方便即時監控與除錯，提升開發效率。在本專題中，透過 Arduino IDE 開發與上傳程式，確保智慧魚缸系統能穩定執行水溫監測、燈光控制及影像傳輸等功能，並具備良好的擴充性與維護性。[2]

2.3 開發版的選擇

在專題開發初期，我們選擇使用 Arduino UNO 板作為控制核心。UNO 板因其簡單易用、社群資源豐富，適合初學者快速上手。然而，我們發現 UNO 板有一些限制，最明顯的是無法直接支援 Wi-Fi 或藍牙功能，導致無法實現遠端操控，這對於我們專題中需要遠端監控與控制的需求來說是一大缺憾。

為了解決這個問題，我們改用 ESP32 DevKit V1 開發板。ESP32 內建 Wi-Fi 和藍牙模組，能輕鬆連接無線網路，實現遠端控制功能。使用 ESP32，我們能將其設置為一個 Web 伺服器，透過手機或電腦的瀏覽器遠端控制 GPIO 腳位上的 LED 燈或其他裝置。此外，ESP32 支援 MQTT 等物聯網通訊協定，讓遠端控制更為穩定且擴充性高。

在開發流程上，ESP32 雖然需要在 Arduino IDE 中安裝額外的開發板套件，但整體編程環境與 UNO 相似，學習曲線不會太陡峭。ESP32 的程式上傳過程稍有不同，需要按住 BOOT 鍵配合上傳，但成功後能透過序列監視器獲得 IP 位址，方便後續遠端存取。

2.3.1 ESP32 DevKit V1

本專題使用的 ESP32 DevKit V1 開發板搭載了穩定的 CP2102 USB 轉 TTL 晶片，確保與電腦之間的串口通訊順暢，方便程式上傳與除錯。ESP32 集成了天線開關、射頻 Balun、功率放大器、低噪聲放大器、過濾器及電源管理模組，整合度高且佔用較少 PCB 空間。

採用 TSMC 40 奈米低功耗製程，ESP32 提供 2.4 GHz Wi-Fi 與藍牙雙模功能，具備優異的功耗與射頻性能，安全可靠且易於擴展。這些特點使其成為本專題智慧魚缸系統穩定且高效的硬體基礎。[4]

2.3.2 Arduino UNO

Arduino UNO 的設計簡潔且穩定，擁有豐富的教學資源與社群支援，是入門實體運算與物聯網開發的理想選擇。使用者可透過 Arduino IDE 輕鬆編寫與上傳程式，快速實現感測器讀取與執行器控制等功能，以下表格是跟 ESP32 DevKit V1 的功能比較，如表 2.1。

表 2.1 功能比較

項目	Arduino UNO	ESP32 DevKit V1
無線連接	無內建 Wi-Fi/藍牙	內建 Wi-Fi 與藍牙
遠端控制	不支援	支援 Web 伺服器、MQTT 等遠端控制方式
編程環境	Arduino IDE，簡單易用	需安裝 ESP32 板套件，類似 Arduino IDE
硬體資源	8 位元微控制器，記憶體較少	32 位元雙核處理器，記憶體較大
適用場景	基本控制與感測	物聯網應用、遠端控制、複雜運算

2.4 ESP32-CAM

ESP32-CAM 是由安信可科技研發的 ESP32 開發板，配備了一個 200 萬像素（1600×1200 像素）的 OV2640 攝影模組（攝像頭）。該開發板的 PCB 上標示的 VCC 引腳預設輸出電壓為 3.3V。

ESP32-CAM 開發板本身並未內建 USB 轉 TTL 序列訊號的 IC，也沒有 USB 介面，因此在燒錄程式碼時需要外接 USB 轉 TTL 序列訊號模組。此外，為了讓 ESP32 進入燒錄模式，必須將 GPIO 0 腳接地。

由於 ESP32-CAM 不具備內建的 USB 轉 TTL 功能，且在燒錄過程中需要外接 USB 轉 TTL 模組，強烈建議選擇具備 3.3V 邏輯電位的序列訊號轉換模組，以確保與 ESP32-CAM 的兼容性。[5][11]

2.5 溫度感測器

我們在魚缸溫度感測專題中選用 DS18B20 作為溫度感測器，主要原因在於 DS18B20 是數位溫度感測器，經過工廠校準，能直接輸出準確的溫度數值，測量範圍廣（-55°C 到 +125°C），且精度高，誤差通常在 ±0.5°C 以內，數據穩定且易於使用，無需額外校正或複雜的訊號轉換。

相較之下，LM35 是類比溫度感測器，輸出與溫度成比例的電壓，Arduino 需透過 ADC 將電壓轉換為溫度值，受限於 ADC 解析度與電源穩定性，容易產生較大誤差，且在實驗中 LM35 測得的溫度與真實溫度差距可達 7 至 9 度，數據較不穩定。此外，LM35 的測量範圍較窄（約 2°C 至 120°C），且需要較多的外部元件與校正，增加系統複雜度。

因此，DS18B20 在準確度、穩定性與使用便利性上均優於 LM35，特別適合需要精確且穩定水溫監測的魚缸環境，這也是我們選擇 DS18B20 的主要原因。

2.5.1 DS18B20 防水溫度感測器

DS18B20 是美國達拉斯半導體在美國產生的第一個溫度傳感器，以支持“單一總線”界面。它具有低功耗，強大的抗干擾能力，易於匹配處理器的優勢，可以將溫度直接通過線路轉換為數字信號。DS18B20 使用 1 線路通信，僅是數據線（以及地面）和微控制器通信。傳感器的溫度檢測範圍為 -55°C 至 125°C，當溫度範圍超過 -10°C 至 85°C 時，精度也為 ±0.5°C。此外，DS18B20 可以直接從數據線供電，而無需外部電源。

與傳統的熱敏電阻不同，它利用單一總線技術有效地最大程度地減少了外部干擾並提高測量精度。同時，它可以直接將測得的溫度轉換為微型計算機處理的串行數字信號，從而使數據傳輸和處理通過簡單的接口。[6][10]

2.5.2 LM35 溫度感測器

LM35 溫度感測器的測量範圍依不同型號略有差異，但一般基本型號的測量範圍約為 2°C 至 120°C。部分資料與官方規格表則指出 LM35 可測量的溫度範圍更廣，從 -55°C 到 150°C，但實際應用中常以 2°C 至 120°C 作為較穩定的有效測量區間。因此，LM35 適合用於一般環境與工業溫度監測，能涵蓋大部分常見的溫度需求。

2.5.3 功能比較

LM35 為類比型溫度感測器，具有線性輸出、精度約 $\pm 0.5^{\circ}\text{C}$ ，適合一般環境的單點溫度量測，但本身不防水且抗干擾能力較弱。DS18B20 則為數位型感測器，採 1-Wire 通訊，精度與 LM35 相當，解析度更高且抗干擾能力佳，並有防水設計，適合在水中或潮濕環境及多點分布式監控。

整體而言，DS18B20 更適用於需防水與多點量測的場合，也符合我們專題中水溫監測的需求；LM35 則較適合一般環境下的單點溫度監測，適用於成本較為有限的應用場景。

2.6 LED WS2812

WS2812 是一款內建控制晶片的可編程 RGB LED，將紅、綠、藍三色 LED 與驅動電路整合於單一封裝中。它採用單線數位控制協定，可透過一條訊號線串接多顆 LED，並對每顆 LED 的顏色與亮度進行獨立控制，常用於燈光特效與動態顯示應用。

每顆 WS2812 具備 8-bit PWM 控制，可呈現 24-bit 全彩顏色（約 1670 萬種）。常見搭配 Arduino、ESP32 等開發板使用，支援 FastLED、NeoPixel 等函式庫，開發方便、彈性高。

WS2812 適用於需要多彩變化與視覺提示的系統設計，如智慧照明、互動裝置或裝飾燈光，是物聯網應用中常見的顯示元件之一。

2.7 LINE BOT

LINE Bot 是什麼？簡單來說，它就是能在 LINE 上自動回應訊息的小幫手。現在，LINE Bot 不僅侷限於 LINE 平台，還能與 Facebook、Instagram 等社群媒體整合，讓企業可以同時在多個平台上觸及客戶。這樣一來，不論你在哪裡，都能快速獲得所需服務。

隨著人工智慧技術的進步，LINE Bot 利用自然語言處理（NLP）技術，更精準地理解和回應你的問題，大大提升使用體驗。有沒有想過它怎麼知道你喜歡什麼？其實是因為它會分析你的互動資料，了解你的偏好，以便推薦更符合你需求的產品或服務。這樣的個人化體驗，不僅讓溝通更順暢，也提高了轉換率。[7]



圖 2.2 LINE Bot

2.8 ChatGPT

是由 OpenAI 開發的人工智慧聊天機器人程式，於 2022 年 12 月推出。它基於 GPT-3.5 和 GPT-4 架構的大型語言模型，並通過強化學習進行訓練。ChatGPT 主要通過文字與使用者互動，能夠執行包括自動生成文字、自動問答、自動摘要等複雜語言處理任務。此外，它還具備編寫和除錯電腦程式的能力，並在各種知識領域提供詳細和清晰的回答。[8]



圖 2.3 ChatGPT

第參章 研究方法

3.1 研究步驟

1. 規劃：

本專題計劃結合硬體與軟體技術，設計並實作一套智慧魚缸管理系統，整合水溫監測、日光燈光照控制及即時影像傳輸功能。系統透過 LINE Bot 平台，讓用戶能隨時隨地即時查看魚缸狀況並進行遠端操作，確保魚缸環境的穩定與魚類的健康。

硬體部分：

- 採用 ESP32 DevKit 作為主要控制核心，搭配 ESP32-CAM 模組實現即時影像監控。
- 整合 DS18B20 溫度感測器，持續監測魚缸水溫，確保環境適宜。
- 利用 LED WS2812 晶片與 ESP32 DevKit 結合，實現可調節的日光燈光照控制，模擬自然光照環境。

軟體部分：

- 程式開發採用 Arduino IDE，利用其豐富的函式庫與開發工具，提高開發效率與系統穩定性。
- 使用 LINE Messaging API 建立與用戶的互動介面，實現遠端即時監控與控制功能，提升使用者體驗。

本研究規劃透過硬軟體的緊密整合，打造一個智能化、便捷且高效的魚缸管理系統，滿足現代飼主對寵物照護的需求。

月份 項目	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月	一月	二月	三月	四月
擬定主題															
系統架構															
系統撰寫															
資料收集															
文件撰寫															
系統整合															
系統測試															
期中報告															
系統修正															
完成系統															

圖3.1 甘特圖

圖例：淺綠色代表完成 

2. 分析：

本專題將溫度感測器、日光燈控制與即時監視系統結合，旨在全面提升魚群的生活品質，為其創造一個穩定且健康的飼養環境。首先，溫度感測器能夠實時監測魚缸水溫，當偵測到水溫異常時，系統會即時提醒飼主，確保水溫維持在魚類適宜的範圍內，避免因環境波動導致魚群壓力或疾病。

其次，日光燈透過模擬自然光照的變化，幫助魚群維持穩定的生物鐘，促進其正常的生理活動。同時，適當的光照也有助於水草的生長，進一步改善魚缸的生態環境，提升整體水質與生態平衡。

最後，即時監視器讓飼主能透過遠端裝置隨時查看魚缸內的狀況，及早發現魚群的異常行為或環境問題，減少日常巡視的負擔。這不僅提升了飼養的便利性，也增強了管理的安全性與效率。

綜合以上三項技術的整合，智慧魚缸系統能有效提升魚類飼養的品質與管理水平，為現代忙碌的飼主提供一個智能化、便捷且可靠的飼養解決方案。

3. 設計：

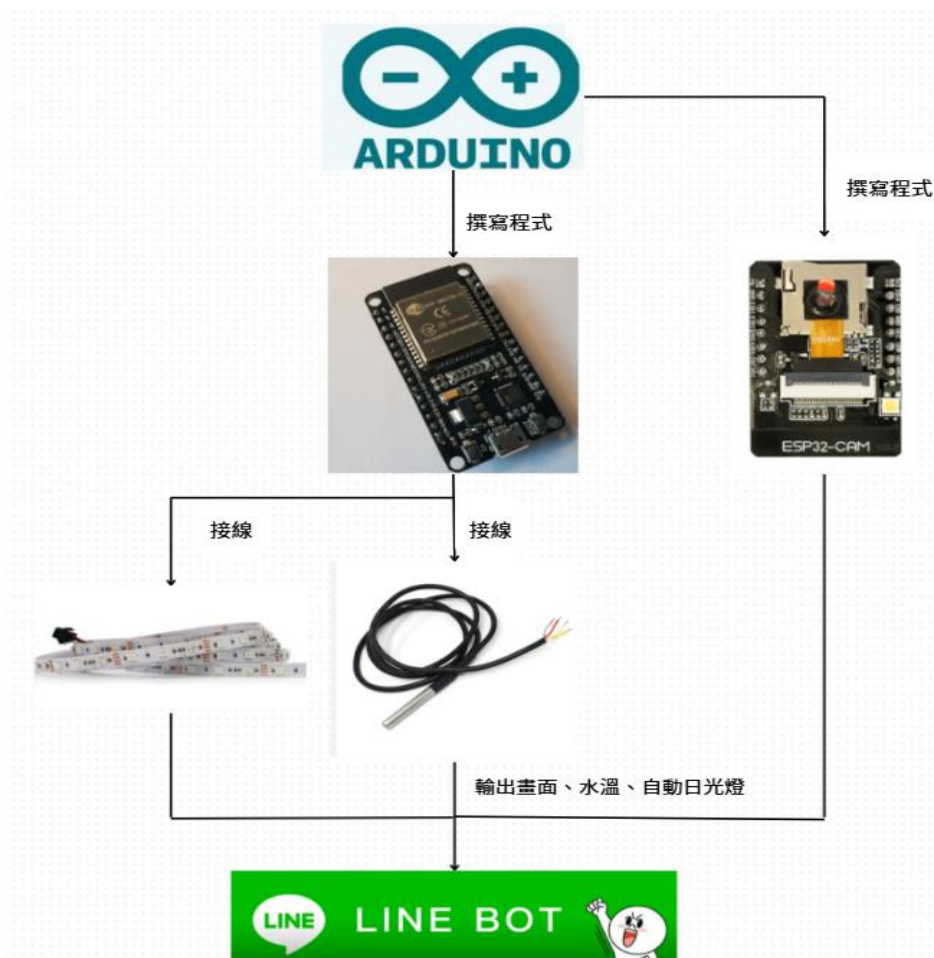


圖3.2 架構圖

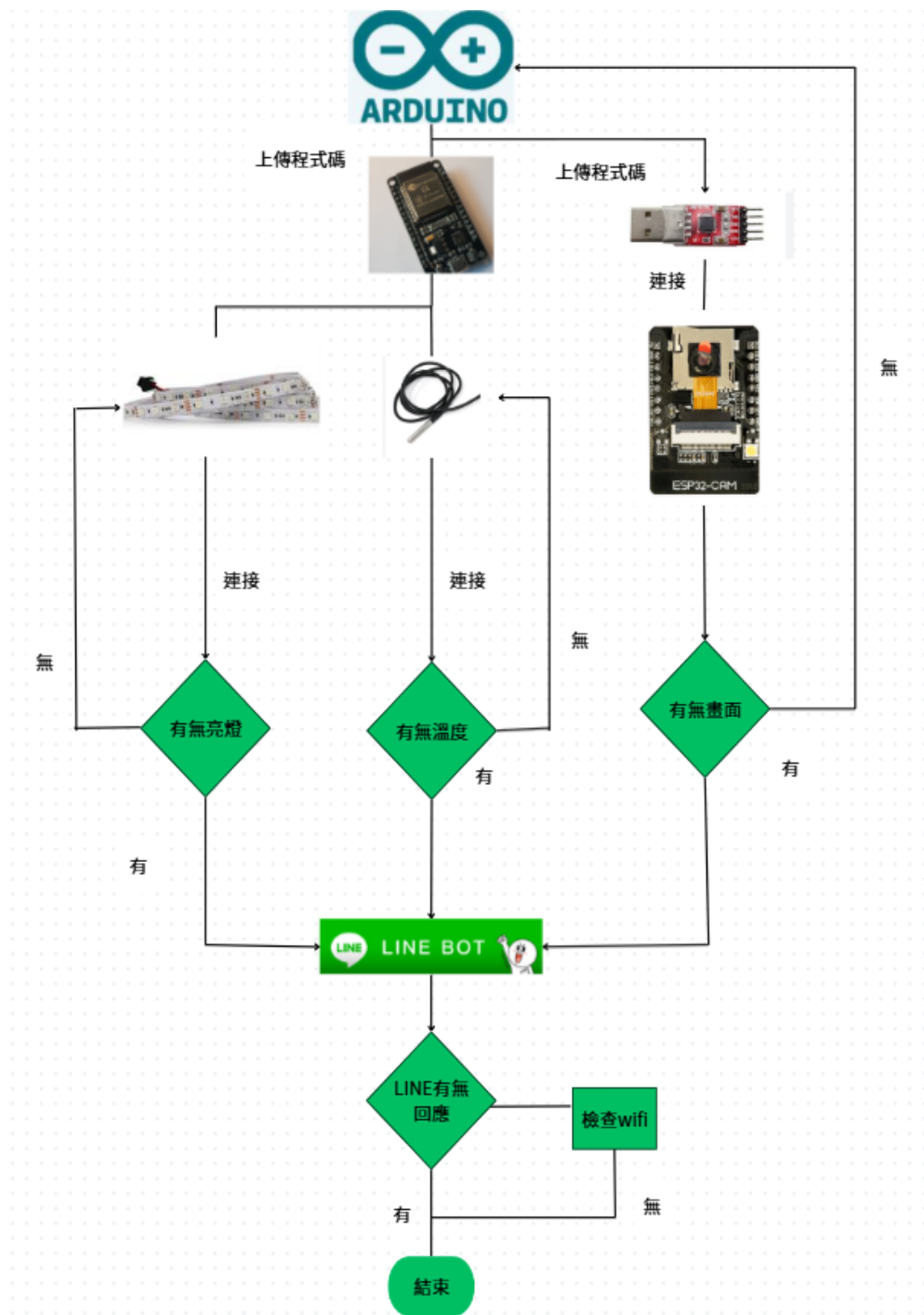


圖3.3 整體流程圖

3.1 功能設計:

本專題智慧魚缸系統的功能設計涵蓋三大核心模組，旨在提升魚缸環境的監控與管理之便利性，具體功能如下：

1. 水溫監測:

系統採用 DS18B20 溫度感測器，持續且即時地監控魚缸內的水溫變化。當感測到水溫異常或超出預設安全範圍時，系統會即時發出警示通知用戶，確保魚類生活環境的穩定與安全，避免因水溫波動而造成魚群壓力或疾病。

2. 日光燈開關:

透過程式設計，系統模擬自然光照的日夜變化，控制 LED 日光燈的開關與燈色變化。此功能不僅節省飼主照顧的時間與精力，也有助於維持魚群的生物鐘，促進魚類及水草的健康生長，提升整體生態環境品質。

3. 即時影像監控:

利用 ESP32-CAM 模組進行即時影像捕捉與傳輸，使用者可透過手機或其他遠端裝置隨時查看魚缸內部狀況。此功能讓飼主能夠及時掌握魚群活動與環境變化，迅速發現異常情況，提升管理的即時性與便利性。

綜合以上功能，本系統不僅實現了魚缸環境的智能化監控與自動化管理，更大幅提升了使用者的操作體驗與養殖效率，為現代化魚類飼養提供了創新且實用的解決方案。

3.2 系統流程圖:

本系統的主要流程涵蓋三大核心功能，透過硬體與軟體的整合，實現智慧魚缸的即時監控與遠端管理：

1. 感測器數據監測與顯示:

系統透過 DS18B20 溫度感測器持續監測魚缸水溫，將即時數據傳送至 ESP32 控制器。控制器接收並處理感測數據後，透過 LINE Bot 平台將水溫資訊即時回傳給使用者，確保飼主能隨時掌握魚缸環境狀況。

2. 日光燈自動控制:

根據預設的光照時間表或使用者指令，系統控制 LED WS2812 日光燈的開關，模擬自然光照環境。此功能不僅提升魚類與水草的生活品質，也減少飼主手動操作的負擔。

3. 即時影像:

利用 ESP32-CAM 模組捕捉魚缸內部即時影像，並將影像傳至 LINE Bot。使用者可透過手機隨時查看魚缸內的實況畫面，及時發現魚群異常或環境問題，提升管理的即時性與便利性。

整體系統流程以使用者體驗為核心，結合感測、控制與影像三大模組，實現智慧化、遠端化的魚缸管理，為現代飼主提供高效且便捷的養殖解決方案。

3.3-1 架構圖[圖 3.2]詳細說明:

本系統架構從程式開發與硬體連接兩大部分著手，最終整合至 LINE Bot 平台，實現智慧魚缸的遠端監控與管理。

1. 程式上傳階段:

首先，利用 Arduino IDE 將整合溫度感測器、即時影像監控與日光燈控制的程式碼編譯並上傳至 ESP32 開發板。上傳成功時，Arduino IDE 不會顯示錯誤訊息，確保程式能在 ESP32 上穩定執行。

2. 硬體連接階段:

- 將 DS18B20 溫度感測器與 LED 日光燈模組透過麵包板與 ESP32 開發板進行連接，確保訊號與電源供應穩定。
- ESP32-CAM 監視器模組則直接連接至電腦或外部電源，提供即時影像捕捉與傳輸功能。

3. 系統整合與遠端控制:

所有硬體設備與程式運行正常後，系統透過 LINE Bot 平台整合，使用者可以透過手機遠端查詢水溫數據、控制日光燈的開關與亮度，並隨時觀看魚缸的即時影像。此架構實現了魚缸環境的自動化管理與遠端監控，大幅提升使用者的便利性與管理效率。

整體架構設計強調模組化與易擴展性，為未來功能升級與系統擴充提供良好基礎。

3.3-2 整體流程圖[圖 3.3]詳細說明:

本專題系統架構涵蓋程式開發、硬體連接與遠端通訊整合三大階段，確保智慧魚缸系統的穩定運作與便捷管理。

1. 程式撰寫與開發環境設定:

首先，在 Arduino IDE 中撰寫包含溫度感測器、日光燈控制與監視器功能的程式碼。完成程式撰寫後，準備 ESP32 開發板及 ESP32-CAM 監視器模組，並將其與電腦進行連接，為後續程式上傳做準備。

2. 硬體連接與測試:

將 DS18B20 溫度感測器與 LED 日光燈模組透過麵包板與 ESP32 開發板連接，確保接線正確且穩定。完成硬體連接後，將程式碼上傳至 ESP32。若上傳失敗，需先檢查程式碼是否有錯誤，再確認硬體接線是否正確。上傳成功後，可透過 Arduino IDE 的序列監控視窗即時查看水溫數據，並觀察日光燈是否依程式指令正常運作。

3. LINE Bot 建立與整合:

接著，於 LINE Developer 平台建立 LINE Bot，並將該 Bot 新增至手機上的 LINE 群組。完成後，將 LINE Bot 的相關程式碼整合進 Arduino 程式中，再次上傳至 ESP32。若 LINE Bot 無法回應，需檢查 Wi-Fi 連線設定與網路狀態。成功連接後，LINE Bot 將能即時回傳水溫數據與監視器影像連結，並控制日光燈的運作。

整體架構設計強調軟硬體的協同運作與遠端互動功能，實現智慧魚缸的自動化監控與便捷管理，為使用者提供高效且可靠的養殖體驗。

4.建置：

本計劃採用 ESP32 DevKit 和 ESP32-CAM 作為核心開發板，配合 DS18B20 溫度感測器進行水溫監測。攝影機部分使用 ESP32-CAM 內建的 OV2640 鏡頭，提供即時影像功能；日光燈則用 LED WS2812 來實現。系統程式撰寫採用 Arduino IDE，並透過 LINE Messaging API 實現與用戶的互動。

硬體接線部分：

1. DS18B20 接 ESP32 DevKit：

- 紅線 (VCC) 接 3.3V
- 黑線 (GND) 接 GND
- 黃線 (Data) 接 GPIO 數據腳位 (如 GPIO4)。

2. ESP32-CAM 燒錄方式：

- 使用 USB 轉 TTL 工具，將 ESP32-CAM 連接至電腦。
- GND 接 GND，VCC 接 3.3V，RX 接 TX，TX 接 RX。

3. LED WS2812 日光燈：

- 日光燈 VCC 接 ESP32 5V，GND 接 GND，信號腳 (IN) 接 GPIO 數據腳位。

程式部分，透過 Arduino IDE 編寫並上傳至 ESP32，LINE Bot 用於遠程操作與通知功能，包括查詢水溫、控制日光燈，以及提供即時影像連結等功能。

第肆章 系統分析與設計

4.1 摘要

本專題成功實作了一套結合多種硬體設備的智慧魚缸管理系統，主要硬體包括 ESP32 DevKit V1 開發板、ESP32-CAM 影像模組、DS18B20 溫度感測器以及 LED WS2812 日光燈條。系統透過這些元件的協同運作，實現了魚缸環境的遠端監控與控制功能。

在軟體部分，採用 Arduino IDE 作為開發環境，撰寫完整的控制與通訊程式。系統能即時讀取 DS18B20 感測器傳回的水溫數據，並將這些資訊透過 LINE Bot 平台傳送給使用者，方便遠端監控魚缸環境。根據水溫的變化，系統自動調整 LED WS2812 燈條的顏色，模擬不同光照效果，為魚群提供適宜的生活環境。

此外，ESP32-CAM 模組負責捕捉魚缸內部的即時影像，並將影像串流整合至 LINE Bot，使用者能隨時透過手機查看魚缸狀況，提升管理的即時性與便利性。整體系統設計不僅提升了魚缸的管理便利性，也展現了物聯網技術在智慧養殖領域的應用潛力。

4.2 系統架構

本專題系統架構由硬體與軟體兩大部分組成，透過緊密整合實現智慧魚缸的遠端監控與自動化管理。

1. 硬體組成

- **ESP32 DevKit V1**

作為系統的主要控制單元，ESP32 DevKit V1 負責接收感測器數據、處理控制邏輯，並透過內建的 Wi-Fi 功能實現無線資料傳輸，確保系統能與遠端平台即時通訊。

- **ESP32-CAM**

用於視覺監控，負責捕捉魚缸內的即時影像，並將影像資料透過無線網路傳送至使用者端，讓飼主能隨時觀察魚群的狀態，提升管理的即時性與便利性。

- **DS18B20 溫度感測器**

持續監測魚缸內的水溫，提供準確的環境溫度數據，作為系統自動調節光照及其他控制策略的重要依據。

- **LED WS2812 日光燈條**

根據水溫的變化自動調整燈光顏色，模擬自然光照環境，不僅提升魚群的生活品質，也增強魚缸的觀賞效果。

2. 軟體組成

- **Arduino IDE**

作為系統的開發環境，Arduino IDE 用於撰寫、編譯及上傳控制程式至 ESP32，並提供豐富的函式庫支援，簡化硬體控制與通訊開發流程。

- **LINE Bot API**

負責實現系統與使用者之間的遠端互動，透過 LINE Bot 平台即時傳送水溫數據、影像連結及控制指令，讓使用者能隨時掌握魚缸狀況並進行遠端操作。

4.3 實作流程

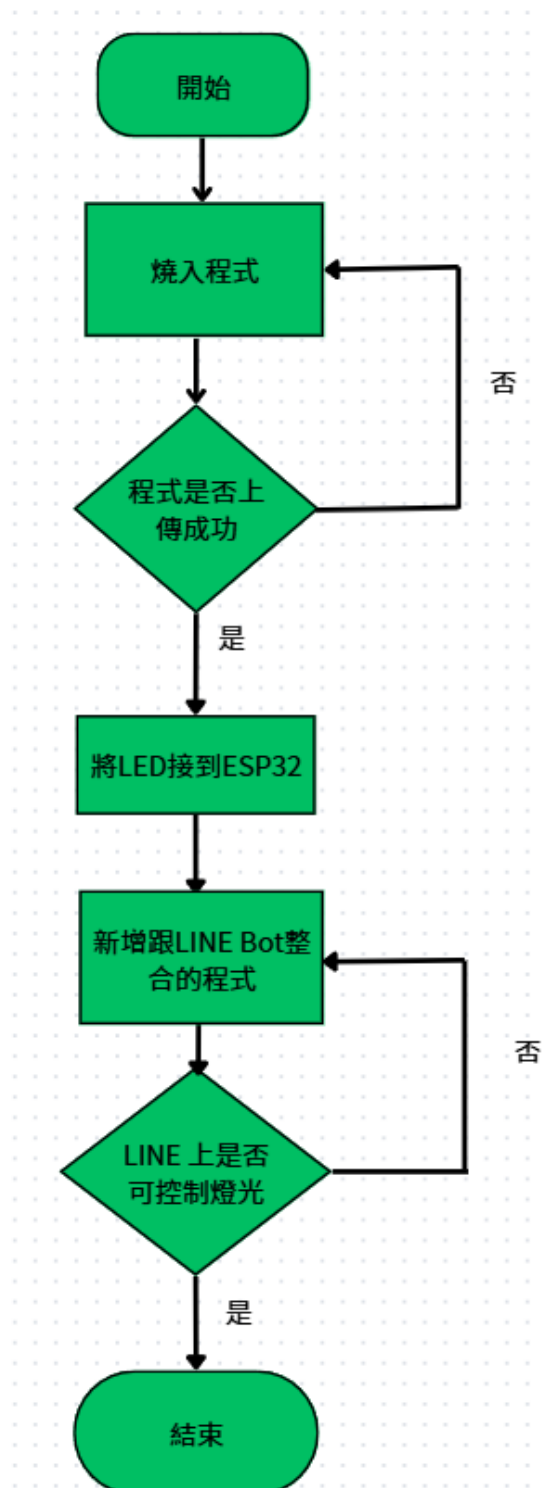


圖 4.1 日光燈流程圖

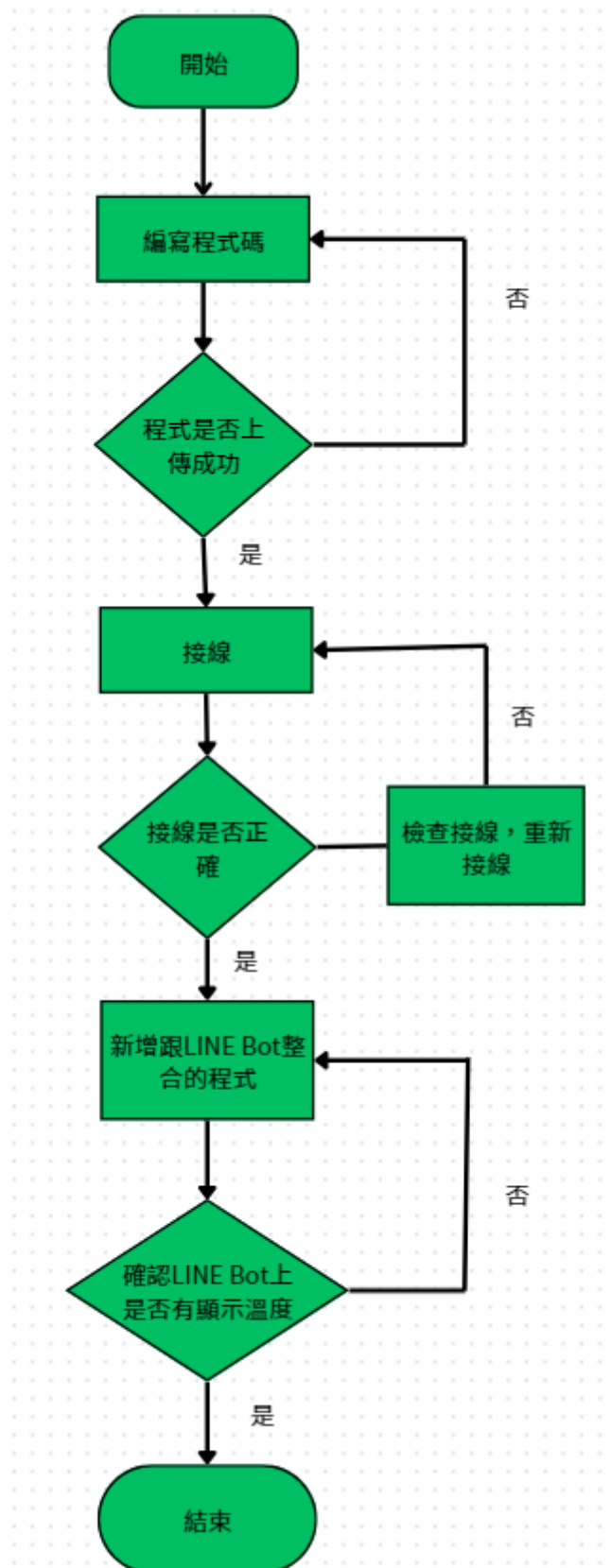


圖 4.2 溫感器流程圖

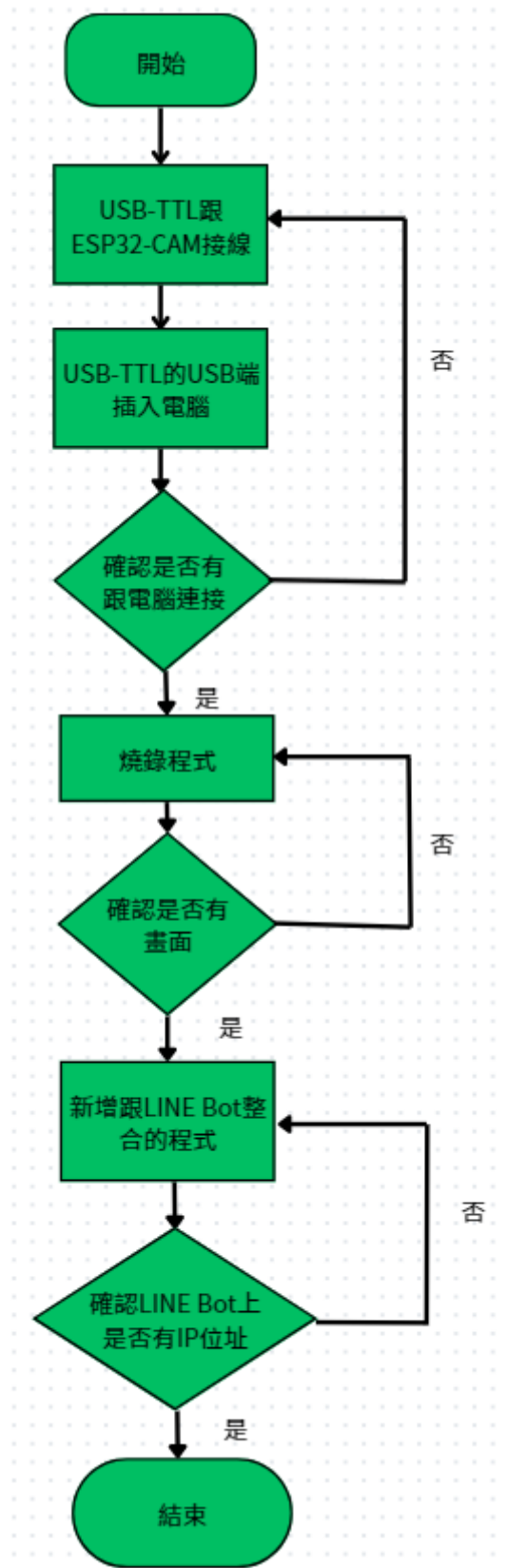


圖 4.3 USB-TTL 跟 ESP32-CAM 連接流程圖

4.3-1 日光燈控制流程[圖 4.1]詳細說明:

把日光燈程式用 Arduino IDE 寫好，將所有程式上傳到 ESP32 板之後，開始接線，接線方式如下:[圖 4.4]

- VCC: 連接到 ESP32 DevKit V1 的 5V 輸出端口。
- GND: 連接到 ESP32 DevKit V1 的 GND 端口。
- Do: 連接到 ESP32 DevKit V1 的數位腳，例如 G5。

接線完並檢查所有接線是否無誤，如果無誤就可以新增跟 LINE Bot 整合的程式，新增完就可以在 LINE Bot 上執行並確認是否可以控制日光燈開關，如果可以就結束了。

4.3-2 溫感器控制流程[圖 4.2] 詳細說明:

本計畫選用 DS18B20 溫度感測器。溫感器程式確認上傳到 ESP32 板後，就可以開始接線，本專題選用的溫感器有三條接線，分別是:[圖 4.5]

- 紅色線 (VCC): 接到 ESP32 DevKit V1 的 3.3V，提供電源給感測器。
- 黑色線 (GND): 接到 ESP32 DevKit V1 的 GND，作為負極。
- 黃色線 (訊號線): 連接到 ESP32 DevKit V1 的數位腳，例如 G15。

接下來，將這些接線用 杜邦線 連接到 麵包板 上。為了確保穩定的數據傳輸，在紅色(3.3V)與黃色 (訊號線)之間加入 10K 電阻，這是必要的拉高電阻，否則電流會過載。並檢查所有接線是否無誤，如果無誤溫感器就可以正常運作，就可以新增跟 LINE Bot 整合的程式，新增完就可以在 LINE Bot 上執行並確認是否有出現溫度，如果有就可以再 LINE Bot 上觀察溫度變化。

4.3-3 將 USB-TTL 轉接器與 ESP32-CAM 正確連接，接著，將 USB-TTL 的 USB 接頭插入電腦，確認設備是否成功連接至電腦。然後開啟 Arduino IDE，選擇 ESP32-CAM 和 序列埠，載入測試程式並進行燒錄。燒錄完成後，重啟 ESP32-CAM 並觀察 Arduino IDE 的序列監視器，檢查是否有正確的監控畫面輸出。若未出現監控畫面，就檢查程式碼和 WI-FI 是否有連接，並重新編譯、上傳，如有畫面且功能正常，就可以新增跟 LINE Bot 整合的程式，新增完就可以在 LINE Bot 上執行並確認是否有出現 IP 位址，如果有就結束。[圖 4.3] 接線方式如下:[圖 4.6]

- ESP32-CAM GND→USB-TTL GND，接地。
- ESP32-CAM U0R→USB-TTL TXD，ESP32-CAM 的 RX 接到 USB-TTL 的 TX。
- ESP32-CAM U0T→USB-TTL RXD，ESP32-CAM 的 TX 接到 USB-TTL 的 RX。
- ESP32-CAM 5V→USB-TTL 5V，提供 5V 電源。
- 燒錄前 ESP32-CAM GPIO0 短接到 GND，才會進入燒錄模式。



圖 4.4 日光燈

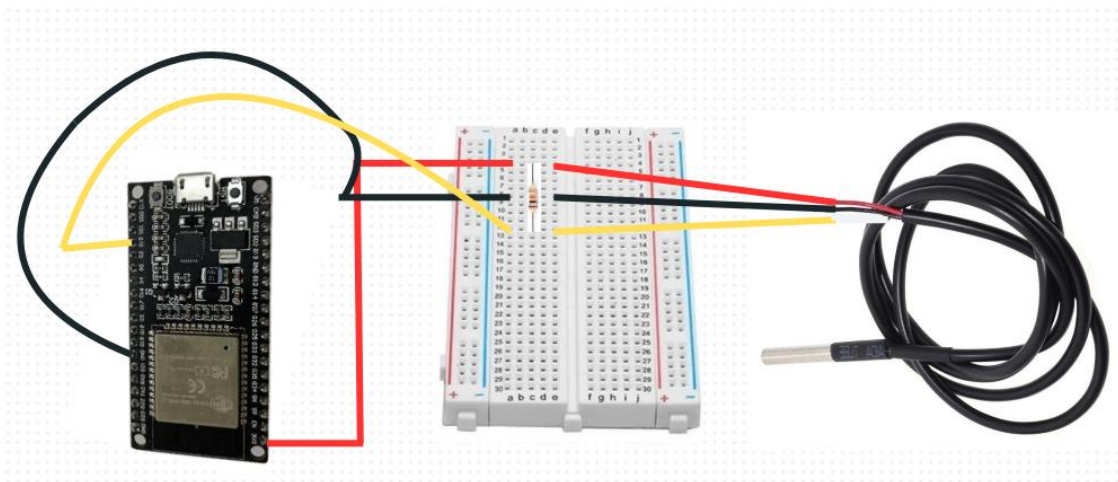


圖4.5 DS18B20防水溫度感測器

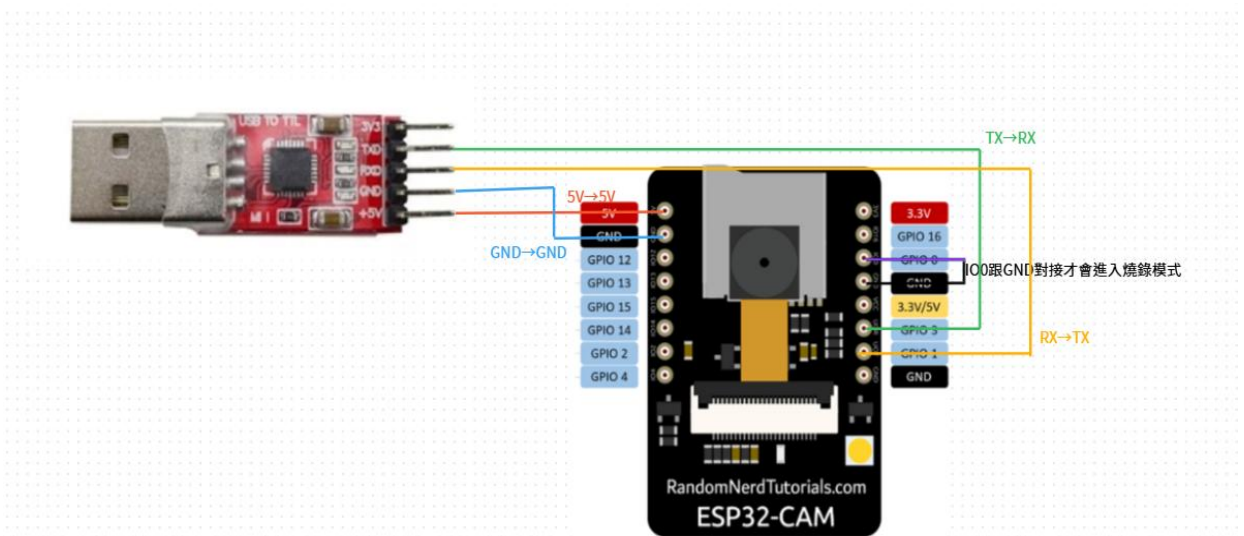


圖 4.6 ESP32-CAM

4.4 操作步驟

1. 環境設定:

- 下載並安裝 Arduino IDE。
- 安裝 ESP32 的板子套件，以便在 Arduino IDE 中使用 ESP32 系列板子。

2. 硬體連接:

- 將 DS18B20 溫度感測器連接到 ESP32 的數位引腳。
- 將日光燈條連接到 ESP32 的 PWM 引腳，以控制燈條的顏色。
- 將 ESP32-CAM 跟 USB-TTL 進行連接。

3. 程式設計:

- 使用 Arduino IDE 撰寫程式，讀取 DS18B20 的溫度數據。

根據溫度設定燈條顏色：

低於 25°C 顯示藍色；

25°C 至 30°C 顯示綠色；

30°C 以上顯示紅色。

- 使用 Wi-Fi 功能將溫度數據傳送到 LINE Bot。
- 使用 Arduino IDE 撰寫程式，使日光燈開啟，但在 LINE 上傳開燈是無法使用因為在 LINE 上沒辦法去控制 ESP32 所以要在別的地方建立程式讓他去控制。
- 使用 Arduino IDE 撰寫程式，使監視器有畫面。

4. 遠端顯示:

- 設定 LINE Bot 以接收溫度數據、控制日光燈開關，以及監視器的 IP 位址，並顯示給用戶。[圖 4.7]
- 輸入監視器的 IP 位址，就會顯示監視器的設定畫面，按下網頁最下面的 start stream，就會有畫面了。[圖 4.8]、[圖 4.9]



圖 4.7 LINE Bot 介面

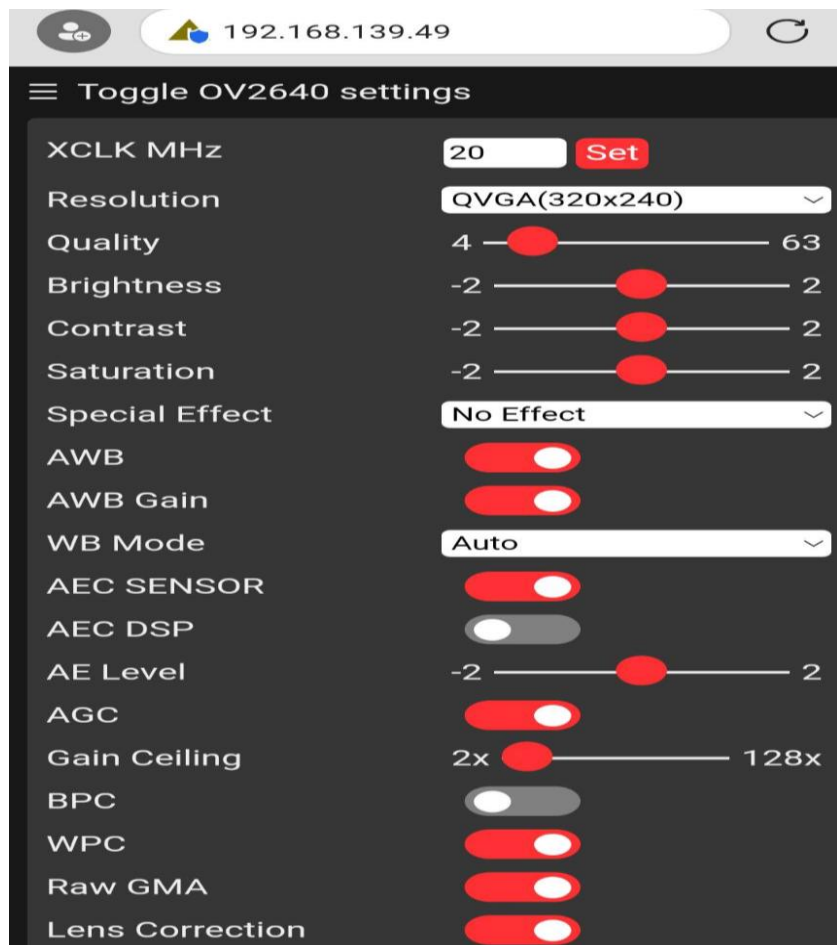


圖 4.8 ESP32-CAM 畫面

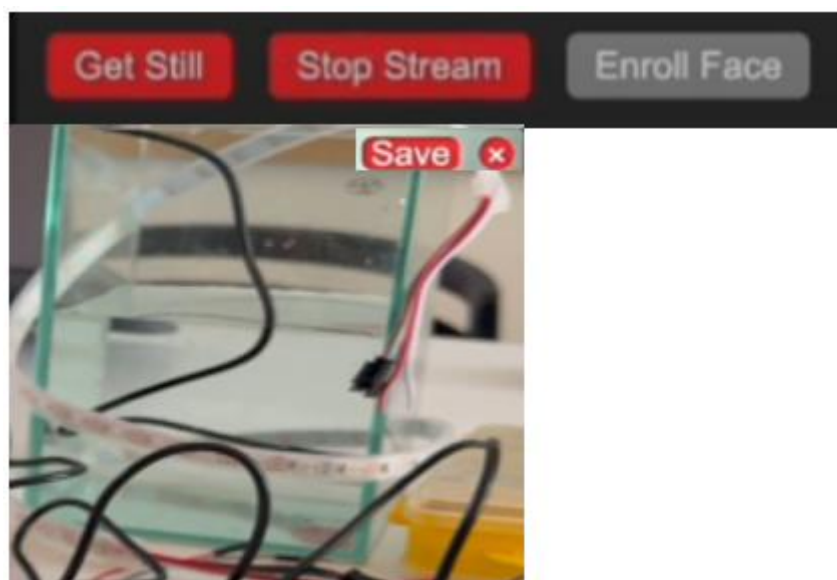


圖 4.9 ESP32-CAM 畫面-2

第伍章 結論

本專題成功建構了一套基於 ESP32 平台的智慧魚缸管理系統，整合了溫度感測、日光燈控制、影像監控與遠端通訊等多項功能，充分展現了物聯網技術在水族環境管理上的實用性與可行性。系統採用 DS18B20 感測器進行即時水溫監控，能準確掌握魚缸內環境變化，確保水溫維持在適合魚類生存的穩定範圍內，降低環境波動對魚群健康的影響。

此外，透過對日光燈的自動控制，使用者無需手動操作，即可維持魚缸內適當且穩定的光照條件，模擬自然光照節律，促進魚類與水草的健康成長。ESP32-CAM 模組則提供即時影像傳輸功能，使使用者能隨時透過遠端裝置查看魚缸狀況，及早發現異常，提升管理的即時性與便利性。

整體系統結合 LINE Bot 平台，實現遠端互動功能，使用者可透過手機查詢水溫數據及監控畫面，進一步提升系統的智慧化與易用性。該系統不僅實現了魚缸環境的自動化管理，也展現了物聯網系統整合的強大能力，為未來智慧養殖、智慧家庭等相關應用場域奠定了良好的技術基礎與實踐經驗。

參考文獻

- [1] Arduino 維基百科
<https://zh.wikipedia.org/zh-tw/Arduino>
- [2] 認識 Arduino 與 IDE 整合環境說明
<https://dic.vbird.tw/arduino/unit01.php>
- [3] ESP32 IOT 物聯網晶片介紹
<https://www.icsshop.com.tw/blog/posts/esp32-iot%E7%89%A9%E8%81%AF%E7%B6%B2%E6%99%B6%E7%89%87%E4%BB%8B%E7%B4%B9>
- [4] ESP32 DevKit V1
<https://www.sdmodel.com.tw/content.php?cn=item&tid=1263&start=0>
- [5] ESP32-CAM ESP32-CAM 開發版(一): 簡介與燒錄程式
<https://swf.com.tw/?p=1723>
- [6] DS18B20 介紹
<https://www.ariat-tech.tw/blog/the-ultimate-guide-to-the-ds18b20-digital-temperature-sensor.html#t1>
- [7] LINE Bot 介紹
<https://www.pintech.com.tw/tw/article/224/line-bot-explained>
- [8] ChatGPT 維基百科
<https://zh.wikipedia.org/zh-tw/ChatGPT>
- [9] Arduino 的應用
[A108011.pdf](#)
- [10] 浸泡式溫度感測器 DS18B20 Arduino 開發
[浸泡式溫度感測器 DS18B20 Arduino 開發 | 量測苦活就交給它](#)
- [11] EP32CAM 基礎範例教學
[ESP32CAM 基礎範例教學 | arduino wifi - YouTube](#)

附 錄

➤ 溫感器和日光燈程式碼

```
1 #include <WiFi.h>
2 #include <WiFiClientSecure.h>
3 #include <HTTPClient.h>
4 #include <ArduinoJson.h>
5 #include <DallasTemperature.h>
6 #include <OneWire.h>
7 #include <FastLED.h>
8
9 // Wi-Fi 設定
10 const char* WIFI_SSID = "realme 11 5G";
11 const char* WIFI_PASSWORD = "17zupdi5";
12
13 // Line Bot 設定
14 const String LINE_API_URL = "https://api.line.me/v2/bot/message/push";
15 const String CHANNEL_ACCESS_TOKEN = "Q168ufjRAHQ2Z1ZBTx01Y2hK1eWd3Q@vsmmc/0P+b15VZFTXPKRnCORfUee3mZklq8Sk2UJmP21Tm7NtXkX9YfHJDLUad3STrHmb07Ga8sA42LxsKvShjBoc9iVRa75/pokoVkyQxfgiPxn";
16 const String USER_ID = "U7222ec355bb97b2151bc4de717f12e60";
17
18 // Firebase 設定
19 const String FIREBASE_URL = "https://esp32-c2dc6-default-rtdb.asia-southeast1.firebaseio.com/app/ledCommand.json";
20
21 // Webhook 設定
22 const String WEBHOOK_URL = "https://script.google.com/macros/s/AKfycbyZqwc8lsFgnLm8o-Cihx5GICT36F9uXEBnxCH5kyvq115hR6UsOUgTuAN66qP-5Te/exec";
23
24 // 溫度感測器設定
25 #define DQ_PIN 15
26 OneWire oneWire(DQ_PIN);
27 DallasTemperature sensors(&oneWire);
28 unsigned long lastTemperatureTime = 0;
29 const unsigned long temperatureInterval = 5000;
30
31 // WS2812 LED 燈條設定
32 #define LED_PIN 5
33 #define NUM_LEDS 60
34 #define BRIGHTNESS 255
35 CRGB leds[NUM_LEDS];
36
37 bool lightState = false; // 預設燈光狀態為關閉
38
39 void setup() {
40   Serial.begin(115200);
41
42   FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS);
43   FastLED.setBrightness(BRIGHTNESS);
44   FastLED.clear();
45   FastLED.show();
46   Serial.println("WS2812 LED initialized");
47
48   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
49   Serial.print("Connecting to WiFi");
50   while (WiFi.status() != WL_CONNECTED) {
51     delay(500);
52     Serial.print(".");
53   }
54   Serial.println("\nWiFi connected");
55
56   sensors.begin();
57   Serial.println("Temperature sensor initialized");
58 }
59
60 void loop() {
61   unsigned long currentMillis = millis();
62
63   // 溫度感測
64   if (currentMillis - lastTemperatureTime >= temperatureInterval) {
65     lastTemperatureTime = currentMillis;
66     sensors.requestTemperatures();
67     float temperature = sensors.getTempCByIndex(0);
68     Serial.println("Temperature: " + String(temperature) + "°C");
69
70     // 溫度感測後根據燈光狀態來顯示顏色
71     if (lightState) {
72       if (temperature < 25) {
73         setLEDColor(0, 0, 255); // 藍色
74       } else if (temperature >= 25 && temperature < 30) {
75         setLEDColor(0, 255, 0); // 綠色
76       } else {
77         setLEDColor(255, 0, 0); // 紅色
78       }
79     }
80
81     sendLineMessage("當前溫度: " + String(temperature) + "°C");
82     sendToWebhook("當前溫度: " + String(temperature) + "°C");
83   }
84
85   // 檢查 Firebase 控制指令
86   checkFirebaseCommand();
87 }
88
89 // 設定 LED 顏色
90 void setLEDColor(int r, int g, int b) {
91   for (int i = 0; i < NUM_LEDS; i++) {
92     leds[i] = CRGB(r, g, b);
93   }
94   FastLED.show();
95   Serial.println("LED color set to: " + String(r) + "," + String(g) + "," + String(b));
96 }
97
98 // 發送 Line 訊息
99 void sendLineMessage(String message) {
100   if (WiFi.status() == WL_CONNECTED) {
101     WiFiClientSecure client;
102     client.setInsecure(); // 測試用，略過憑證驗證
103
104     HTTPClient https;
105     https.begin(client, LINE_API_URL);
106     https.addHeader("Content-Type", "application/json");
107     https.addHeader("Authorization", "Bearer " + CHANNEL_ACCESS_TOKEN);
```

```

108
109 DynamicJsonDocument doc(1024);
110 doc["to"] = USER_ID;
111 JsonObject messageObj = doc.createNestedArray("messages").createNestedObject();
112 messageObj["type"] = "text";
113 messageObj["text"] = message;
114
115 String jsonData;
116 serializeJson(doc, jsonData);
117
118 int httpResponseCode = https.POST(jsonData);
119 if (httpResponseCode > 0) {
120   Serial.printf("訊息發送成功，回應碼: %d\n", httpResponseCode);
121 } else {
122   Serial.printf("發送失敗，錯誤訊息: %s\n", https.errorToString(httpResponseCode).c_str());
123 }
124 https.end();
125 } else {
126   Serial.println("WiFi 未連接，無法發送訊息");
127 }
128 }
129
130 // 發送訊息到 Webhook
131 void sendToWebhook(String message) {
132   if (WiFi.status() == WL_CONNECTED) {
133     HTTPClient http;
134     http.begin(WEBHOOK_URL);
135     http.addHeader("Content-Type", "application/x-www-form-urlencoded");
136
137     String payload = "message=" + message;
138
139     int httpResponseCode = http.POST(payload);
140     if (httpResponseCode > 0) {
141       Serial.printf("訊息發送到 Webhook 成功，回應碼: %d\n", httpResponseCode);
142     } else {
143       Serial.printf("發送失敗，錯誤訊息: %s\n", http.errorToString(httpResponseCode).c_str());

```

```

144   }
145   http.end();
146 } else {
147   Serial.println("WiFi 未連接，無法發送訊息到 Webhook");
148 }
149 }
150
151 // 檢查 Firebase 控制指令
152 void checkFirebaseCommand() {
153   if (WiFi.status() == WL_CONNECTED) {
154     HTTPClient http;
155     http.begin(FIREBASE_URL);
156     int httpCode = http.GET();
157     if (httpCode == 200) {
158       String payload = http.getString();
159       Serial.println("Firebase payload: " + payload);
160
161       // 檢查 Firebase 是否有開燈指令
162       if (payload.indexOf("on") >= 0) {
163         if (!lightState) { // 如果目前燈光為關閉狀態，才會開燈
164           lightState = true;
165           setLEDColour(255, 255, 255); // 預設開燈顏色為白色
166           Serial.println("Firebase 命令：開燈");
167         }
168       }
169       // 檢查 Firebase 是否有關燈指令
170       else if (payload.indexOf("off") >= 0) {
171         if (lightState) { // 如果目前燈光為開啟狀態，才會關燈
172           lightState = false;
173           setLEDColour(0, 0, 0); // 關燈
174           Serial.println("Firebase 命令：關燈");
175         }
176       }
177     } else {
178       Serial.printf("Firebase 讀取失敗: %d\n", httpCode);
179     }

```

```

148   }
149 }
150
151 // 檢查 Firebase 控制指令
152 void checkFirebaseCommand() {
153   if (WiFi.status() == WL_CONNECTED) {
154     HTTPClient http;
155     http.begin(FIREBASE_URL);
156     int httpCode = http.GET();
157     if (httpCode == 200) {
158       String payload = http.getString();
159       Serial.println("Firebase payload: " + payload);
160
161       // 檢查 Firebase 是否有開燈指令
162       if (payload.indexOf("on") >= 0) {
163         if (!lightState) { // 如果目前燈光為關閉狀態，才會開燈
164           lightState = true;
165           setLEDColour(255, 255, 255); // 預設開燈顏色為白色
166           Serial.println("Firebase 命令：開燈");
167         }
168       }
169       // 檢查 Firebase 是否有關燈指令
170       else if (payload.indexOf("off") >= 0) {
171         if (lightState) { // 如果目前燈光為開啟狀態，才會關燈
172           lightState = false;
173           setLEDColour(0, 0, 0); // 關燈
174           Serial.println("Firebase 命令：關燈");
175         }
176       }
177     } else {
178       Serial.printf("Firebase 讀取失敗: %d\n", httpCode);
179     }
180   }
181   http.end();
182 }

```

➤ Apps Script 程式

```
1 // 替换成你自己的 LINE Bot Token (在 Line Developers 后台的 Channel access token (long-lived))
2 const LINE_CHANNEL_ACCESS_TOKEN = "Q168ufj8AHQ2I2BT81V2hK1eWdQ8vsmnc/0P+bISVZftXP1KrnCORfuee3m2Kglq8Sk2UJMW2iTm3NWntxkX9VFH0ULUd35TrNmB07GwBSA42LxsKhGBjBoc91VRa7S/poToVKyQxfgIPxhno1wDB04t89/10/v1cDny11FU=";
3
4 // 替换成你的 Firebase Realtime Database URL (结尾记得是 /ledCommand.json)
5 const FIREBASE_URL = "https://esp32-c2dc6-default-rtdb.asia-southeast1.firebaseiodatabase.app/ledCommand.json";
6
7 // 主程式 - 处理从 Line 传递来的讯息
8 function doPost(e) {
9   const contents = JSON.parse(e.postData.contents);
10   const userMessage = contents.events[0].message.text;
11   const replyToken = contents.events[0].replyToken;
12
13   // 判断讯息内容
14   let command = "";
15   if (userMessage === "开灯") {
16     command = "on";
17   } else if (userMessage === "关灯") {
18     command = "off";
19   } else {
20     replyToUser(replyToken, "请输入「开灯」或「关灯」来控制设备！");
21     return;
22   }
23
24   // 写入 Firebase Realtime Database
25   const response = UrIFetchApp.fetch(FIREBASE_URL, {
26     method: "put",
27     contentType: "application/json",
28     payload: '{}' + command + '{}',
29   });
30
31   // 回复用户讯息
32   replyToUser(replyToken, `已${userMessage}`);
33 }
34
35 // 回覆讯息给用户
36 function replyToUser(replyToken, msg) {
37   const url = "https://api.line.me/v2/bot/message/reply";
38   const payload = JSON.stringify({
39     replyToken: replyToken,
40     messages: [{ type: "text", text: msg }]
41   });
42
43   UrIFetchApp.fetch(url, {
44     method: "post",
45     contentType: "application/json",
46     headers: {
47       "Authorization": "Bearer " + LINE_CHANNEL_ACCESS_TOKEN
48     },
49     payload: payload
50   });
51 }
```

➤ firebase 程式

```
1 ledCommand
2 :
3 "off"
```

➤ 監視器程式碼

```
1 #include <WiFi.h>
2 #include <WiFiClientSecure.h>
3 #define CAMERA_MODEL_AI_THINKER // Has PSRAM
4 #include "esp_camera.h" // 相機晶片
5 #include "camera_pins.h"
6 #include <HTTPClient.h>
7
8 // =====
9 // WiFi 設定
10 // =====
11 const char *ssid = "realme 11 5G";
12 const char *password = "17zupdi5";
13 const String LINE_API_URL = "https://api.line.me/v2/bot/message/push";
14 const String CHANNEL_ACCESS_TOKEN = "q168ufjRAHQ321ZBTX01VZK1eWQ8vsmc/0P+b15Y2FTXP1KncORFuee3n2K1q8Sk2U7M42Itm7MttXkX9YFH0ULmd35TrHmb07Gw8sA42LxsX05h8jbc91VRa75/pokvKyQxg1Pxn01wdB04t89/10/wcDmy1lFu+";
15 const String USER_ID = "U7222ec355bb97b2151bc4de717f12e60";
16
17 void startCameraServer();
18 void setupLedFlash(int pin);
19
20 const unsigned long interval = 30000;
21 unsigned long previousMillis = 0;
22
23 void sendLineMessage(String message) {
24     HTTPClient http;
25     http.begin(LINE_API_URL);
26     http.addHeader("Content-Type", "application/json");
27     http.addHeader("Authorization", "Bearer " + CHANNEL_ACCESS_TOKEN);
28
29     // 正確的 JSON 格式
30     String postData = "{\"to\":\"" + USER_ID + "\",\"messages\":[{\"type\":\"text\",\"text\":\"" + message + "\"}]}";
31
32     int httpResponseCode = http.POST(postData);
33
34     if (httpResponseCode > 0) {
35         Serial.println("Message sent to Line successfully.");
36         Serial.println(http.getString()); // 顯示伺服器回應
37     } else {
38         Serial.printf("Error sending message to Line: %d\n", httpResponseCode);
39     }
40     http.end();
41 }
42
43 void setup() {
44     Serial.begin(115200);
45     Serial.setDebugOutput(true);
46     Serial.println();
47 }
```

```
48 camera_config_t config;
49 config.ledc_channel = LEDC_CHANNEL_0;
50 config.ledc_timer = LEDC_TIMER_0;
51 config.pin_d0 = Y2_GPIO_NUM;
52 config.pin_d1 = Y3_GPIO_NUM;
53 config.pin_d2 = Y4_GPIO_NUM;
54 config.pin_d3 = Y5_GPIO_NUM;
55 config.pin_d4 = Y6_GPIO_NUM;
56 config.pin_d5 = Y7_GPIO_NUM;
57 config.pin_d6 = Y8_GPIO_NUM;
58 config.pin_d7 = Y9_GPIO_NUM;
59 config.pin_xclk = XCLK_GPIO_NUM;
60 config.pin_pclk = PCLK_GPIO_NUM;
61 config.pin_vsync = VSYNC_GPIO_NUM;
62 config.pin_href = HREF_GPIO_NUM;
63 config.pin_sscb_sda = SIOD_GPIO_NUM;
64 config.pin_sscb_scl = SIOC_GPIO_NUM;
65 config.pin_pwdn = PWDN_GPIO_NUM;
66 config.pin_reset = RESET_GPIO_NUM;
67 config.xclk_freq_hz = 20000000;
68 config.frame_size = FRAMESIZE_UXGA;
69 config.pixel_format = PIXFORMAT_JPEG;
70 config.grab_mode = CAMERA_GRAB_WHEN_EMPTY;
71 config.fb_location = CAMERA_FB_IN_PSRAM;
72 config.jpeg_quality = 12;
73 config.fb_count = 1;
74
75 if (config.pixel_format == PIXFORMAT_JPEG) {
76     if (psramFound()) {
77         config.jpeg_quality = 10;
78         config.fb_count = 2;
79         config.grab_mode = CAMERA_GRAB_LATEST;
80     } else {
81         config.frame_size = FRAMESIZE_SVGA;
82         config.fb_location = CAMERA_FB_IN_DRAM;
83     }
84 } else {
85     config.frame_size = FRAMESIZE_240X240;
86 }
87
88 esp_err_t err = esp_camera_init(&config);
89 if (err != ESP_OK) {
90     Serial.printf("Camera init failed with error 0x%x", err);
91     return;
92 }
93 }
```

```

94     WiFi.begin(ssid, password);
95     WiFi.setSleep(false);
96     Serial.print("Connecting to WiFi...");
97     while (WiFi.status() != WL_CONNECTED) {
98         delay(500);
99         Serial.print(".");
100    }
101    Serial.println("\nWiFi connected!");
102    startCameraServer();
103
104    Serial.print("IP Address: ");
105    Serial.println(WiFi.localIP());
106
107    // 發送 IP 地址到 Line
108    String ipMessage = "ESP32-CAM IP 地址: " + WiFi.localIP().toString();
109    sendLineMessage(ipMessage);
110 }
111
112 void loop() {
113     unsigned long currentMillis = millis();
114
115     if (currentMillis - previousMillis >= interval) {
116         previousMillis = currentMillis;
117         String ipMessage = "ESP32-CAM IP 地址: " + WiFi.localIP().toString();
118         sendLineMessage(ipMessage);
119         Serial.println("IP 地址已發送到 Line bot");
120     }
121 }

```

➤ 報名證明

2025/4/8 晚上8:15

Gmail - 2025第8屆「致青春·創未來全國選拔大賽」【科技應用組】初審報名收件表單

 陳昱廷

2025第8屆「致青春·創未來全國選拔大賽」【科技應用組】初審報名收件表單

1 封郵件

Google 寄件 2025年4月8日 晚上8:05

收件者: 魚樂園

Google Forms

感謝您填寫「2025第8屆「致青春·創未來全國選拔大賽」【科技應用組】初審報名收件表單」

以下是我們收到的回覆。

2025第8屆「致青春·創未來全國選拔大賽」【科技應用組】初審報名收件表單

此表單僅限【科技應用組】報名使用

2025/4/28 (一) 中午12:00截止

!! 請注意!!
1.依競賽辦法規定，每組以3-6人為限。
2.參賽作品曾獲得全國性或跨校性創競賽前三名，或已成立公司者，不得再參加本競賽，若僅參加團隊成員所屬學校之校內競賽，不在此限。
3.參賽者不可同時投遞一件以上作品，也不可投遞曾參加其他競賽之作品，同一題目不得跨組重複報名參加。
若有違反以上規定，將直接取消參賽資格，請多加注意，其餘相關規定請詳閱競賽辦法。
系統已在你提交這份表單時記錄你的電子郵件地址

團隊名稱 *

魚樂園

成員02 - 姓名 *

林秉輝

成員02 - 手機號碼 *

格式：09XX-XXX-XXX

https://mail.google.com/mail/u/0/?ik=79330ecd35&view=pt&search=all&permthid=thread-f:1828836030716192863&simpl=msg-f:1828836030716...

2025/4/8 晚上8:15

Gmail - 2025第8屆「致青春·創未來全國選拔大賽」【科技應用組】初審報名收件表單

範例：0912-123-456

成員02 - E-mail *

成員02 - 學校系級班級 *

格式：XX大學-XX系-年級-班
例：致理科技大學-企管系-三-A
嶺東科技大學-資訊管理系-四-A

成員03 - 姓名 *

陳捷憲

成員03 - 手機號碼 *

格式：09XX-XXX-XXX
範例：0912-123-456

作品名稱

*

https://mail.google.com/mail/u/0/?ik=79330ecd35&view=pt&search=all&permthid=thread-f:1828836030716192863&simpl=msg-f:1828836030716...

2025/4/8 晚上8:15

Gmail - 2025第8屆「致青春·創未來全國選拔大賽」【科技應用組】初審報名收件表單

魚樂園

隊長01 - 姓名

*

陳昱廷

隊長01 - 手機號碼

*

格式：09XX-XXX-XXX

範例：0912-123-456

*

隊長01 - E-mail

*

隊長01 - 學校系級班級

*

格式：XX大學-XX系-年級-班

例：致理科技大學-企管系-三-A

嶺東科技大學-資訊管理系-四-A

成員03 - E-mail

*

成員03 - 學校系級班級

*

<https://mail.google.com/mail/u/0/?ik=79330ecd35&view=pt&search=all&permthid=thread-f:1828836030716192863&simpl=msg-f:1828836030716192863>

2025/4/8 晚上8:15

Gmail - 2025第8屆「致青春-創未來全國競技大賽」【科技應用組】初審報名收件表單

格式：XX大學-XX系-年級-班
例：致理科技大學-企管系-三-A
嶺東科技大學-資訊管理系-四-A

成員04 - 姓名

柳承佑

成員04 - 手機號碼

格式：09XX-XXX-XXX
範例：0912-123-456

成員04 - E-mail

成員04 - 學校系級班級

格式：XX大學-XX系-年級-班
例：致理科技大學-企管系-三-A
嶺東科技大學-資訊管理系-四-A

成員05 - 姓名

王奕新

指導老師01-姓名

<https://mail.google.com/mail/u/0/?ik=79330ecd35&view=pt&search=all&permthid=thread-f:1828836030716192863&simpl=msg-f:1828836030716192863>

2025/4/8 晚上8:15 Gmail - 2025第8屆「致青春-創未來全國競技大賽」【科技應用組】初審報名收件表單

陳健忠

指導老師01-手機號碼

格式：09XX-XXX-XXX 例:0900-111-222

指導老師01-服務學校/單位

格式：XX大學-XX系 例：致理科技大學-企管系

嶺東科技大學-資訊管理系

指導老師01-職稱

例：副教授

助理教授

成員05 - 手機號碼

格式：09XX-XXX-XXX

範例：0912-123-456

<https://mail.google.com/mail/u/0/?ik=79330ecd35&view=pt&search=all&permthid=thread-f:1828836030716192863&simpl=msg-f:1828836030716192863>

2025/4/8 晚上8:15

Gmail - 2025第8屆「致青春-創未來全國競技大賽」【科技應用組】初審報名收件表單

成員05 - E-mail

成員05 - 學校系級班級

格式：XX大學-XX系-年級-班

例：致理科技大學-企管系-三-A

嶺東科技大學-資訊管理系-四-A

提案企劃書(你的PPT)

*

資料審查相關文件網址：<https://pse.js/5nbmzb>

提案企劃書：團隊名稱-隊長姓名-提案企劃書

已提交的檔案

 魚樂園-陳昱廷-提案企劃書-陳昱廷.pdf

資料審查(你的學生證、個資同意書、計畫提案同意書)

*

資料審查相關文件網址：<https://pse.js/5nbmzb>

文件名稱：團隊名稱-隊長姓名-資料審查

已提交的檔案

 魚樂園-陳昱廷-資料審查-陳昱廷.pdf

上傳前再次確認資訊是否有誤，團隊資料作用參賽名單和獎狀，務必再三檢查

表單填寫完會將副本寄至信箱，若有修改需求請直接重新填寫，將以最新的提交資料為主，謝謝。
若有問題請致電於(02)2257-6167#1292 致理科大育成中心詢問。

教師姓名		陳健忠		所屬系所		資訊管理系	
1	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)		
	實務專題(三)	陳健忠	資管4A	114年3月11日	共 1 時 0 分		
輔導項目	■ 專題 魚樂園	個人工作進度回報與任務指派： 監視器整合 LINE BOT			討論結果： ☐ 1. 缺席 ☐ 2. 分數：___		
2	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)		
	實務專題(三)	陳健忠	資管4A	114年3月11日	共 1 時 0 分		
輔導項目	■ 專題 魚樂園	個人工作進度回報與任務指派： 找專題競賽			討論結果： ☐ 1. 缺席 ☐ 2. 分數：___		
3	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)		
	實務專題(三)	林秉華	資管4A	114年3月11日	共 1 時 0 分		
輔導項目	■ 專題 魚樂園	個人工作進度回報與任務指派： 製作專題競賽的PTT			討論結果： ☐ 1. 缺席 ☐ 2. 分數：___		
4	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)		
	實務專題(三)	王雨新	資管4A	114年3月11日	共 1 時 0 分		
輔導項目	■ 專題 魚樂園	個人工作進度回報與任務指派： 溫度感測器和日光燈整合 LINE BOT			討論結果： ☐ 1. 缺席 ☐ 2. 分數：___		
5	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)		
	實務專題(三)	柳承佑	資管4A	114年3月11日	共 1 時 0 分		
輔導項目	■ 專題 魚樂園	個人工作進度回報與任務指派： 製作專題競賽的PTT			討論結果： ☐ 1. 缺席 ☐ 2. 分數：___		
輔導學生人次合計		5 人	輔導時間合計		5 時 0 分	教師簽名 陳健忠	

備註：

一、本表以月為單位。任課教師請於次月 5 日前將本表繳交至系(所)辦彙整，由系(所)辦彙整統計表後擲交教學發展中心。

二、本表留存各系評鑑備查，並請受輔導學生於專題複審前至系辦領回影本，放至專題報告書附錄之中。

三、學生簽名欄，由受輔導學生簽名。

教師姓名		陳建忠		所屬系所	資訊管理系	
1	輔導課程 實務專題(三)	學生簽名 陳建忠	班級 資管4A	輔導日期 114年4月1日	輔導時間(分) 共 時 0 分	
輔導項目	■ 專題 魚樂園	個人工作進度回報與任務指派： PPT製作			討論結果： ☐ 1. 缺席 ☐ 2. 分數：___	
2	輔導課程 實務專題(三)	學生簽名 陳建忠	班級 資管4A	輔導日期 114年4月1日	輔導時間(分) 共 時 0 分	
輔導項目	■ 專題 魚樂園	個人工作進度回報與任務指派： Word修改			討論結果： ☐ 1. 缺席 ☐ 2. 分數：___	
3	輔導課程 實務專題(三)	學生簽名 林秉華	班級 資管4A	輔導日期 114年4月1日	輔導時間(分) 共 時 0 分	
輔導項目	■ 專題 魚樂園	個人工作進度回報與任務指派： PPT製作			討論結果： ☐ 1. 缺席 ☐ 2. 分數：___	
4	輔導課程 實務專題(三)	學生簽名 王雨新	班級 資管4A	輔導日期 114年4月1日	輔導時間(分) 共 時 0 分	
輔導項目	■ 專題 魚樂園	個人工作進度回報與任務指派： 系統整合			討論結果： ☐ 1. 缺席 ☐ 2. 分數：___	
5	輔導課程 實務專題(三)	學生簽名 柳承佑	班級 資管4A	輔導日期 114年4月1日	輔導時間(分) 共 時 0 分	
輔導項目	■ 專題 魚樂園	個人工作進度回報與任務指派： Word修改			討論結果： ☐ 1. 缺席 ☐ 2. 分數：___	
輔導學生人次合計		5 人	輔導時間合計		5 時 0 分	教師簽名 陳建忠

備註：

- 一、本表以月為單位。任課教師請於次月 5 日前將本表繳交至系(所)辦彙整，由系(所)辦彙整統計表後擲交教學發展中心。
- 二、本表留存各系評鑑備查，並請受輔導學生於專題複審前至系辦領回影本，放至專題報告書附錄之中。
- 三、學生簽名欄，由受輔導學生簽名。

113

學年度

嶺東科技大學

資訊管理系

魚樂園