

國立清華大學 電機工程學系

實作專題研究成果摘要

Swimming Bio-robotics

仿生游泳機器人

專題領域：系統領域

組 別：B536

指導教授：楊雅棠 副教授

組員姓名：汪致鈞

研究期間：2025年2月10日至2025年12月19日止，共10個月

摘要

本專題旨在開發一套具備無線控制與水下影像拍攝功能的仿生機械魚平台。為提升客製化能力與模組化彈性，本專題以市售塑膠魚外殼為基礎，重新設計其內部電子系統，包括控制模組、驅動電路、電源轉換與獨立影像擷取系統。主控制單元採用 NodeMCU-32S，並透過 ULN2803APG 達靈頓陣列驅動馬達，使裝置能夠利用手機瀏覽器經由 Wi-Fi 進行尾部推進速度調整。電源部分以 3.7V 鋰電池搭配 MT3608 升壓模組穩定輸出 5V，確保各模組在有限空間內仍具備穩定電力供應。

影像擷取方面，本專題採用 ESP32-CAM 進行獨立運作的水下定時拍照，拍攝的 JPEG 影像會以時間戳記或儲存順序編號方式儲存於 microSD 卡中。考量影像處理所需的運算負載較高，攝影模組與主控制模組採用完全分離的架構，以避免對馬達控制造成延遲或不穩定。此設計亦提高系統擴充性，未來能新增其他功能而不影響核心控制運作。

系統測試結果顯示，NodeMCU-32S 提供的 Wi-Fi 控制具備低延遲與良好穩定性，馬達在不同 PWM 佔空比下能呈現明顯速度變化；ESP32-CAM 於 QVGA 解析度下可穩定拍攝清晰水下影像並成功寫入 SD 卡。經整合後，機械魚能在水中維持穩定前進，並同步完成影像紀錄，展現小型水下載具的可行性。

本專題實作出之系統具備低成本、架構簡潔、易於擴充等特性，可作為水下展示教具、基礎觀測平台與水下機電整合的入門研究基礎。此成果亦展現商用外殼結合開源硬體所能達成的高彈性與開發潛力，為後續加入多馬達、感測器或姿態控制等功能奠定良好基礎。

目錄

摘要	i
目錄	ii
1. 背景與研究動機	1
2. 研究目的	1
3. 實作方法	2
3.1. 結構拆解與內部空間配置	2
3.2. 電源系統設計與 MT3608 升壓調校	2
3.3. ULN2803APG 馬達驅動電路	3
3.4. NodeMCU-32S 韌體開發 (Wi-Fi 控制)	3
3.5. ESP32-CAM 安裝與影像擷取流程	3
4. 研究結果	4
4.1. 控制效能：PWM 與 Wi-Fi 遠端操作的整合性驗證	4
4.2. 電源穩定性：電流負載測試與電壓掉落分析	4
4.3. 水下運動表現：推進力、穩定性與限制	5
4.4. 影像品質與資料儲存：水下影像成功率提升	5
5. 結論	6
6. 心得	6
Appendix	7
Appendix A — Source Code Repository	7

1. 背景與研究動機

水下仿生機器人因其靈活、安靜且自然的運動行為，逐漸受到工程教育、環境監測與入門型水下平台研究的關注。其中以仿生機械魚最具代表性。然而市售機械魚往往採封閉式設計，使用者無法調整內部電路，也難以加入影像、感測或無線通訊功能，使其在教學與研究應用層面受到限制。

本研究基於此問題，選擇以市售魚殼為物理外型，重新建立其電子與控制架構，使其成為一個開放式、可擴充的水下教學平台。開發採用 Arduino 生態系，原因在於其開源、模組化、文件完整、社群龐大，並且 NodeMCU-32S 與 ESP32-CAM 等模組均能以 Arduino IDE 進行開發，能有效縮短開發時間，降低硬體整合難度。

研究動機包含三個面向：

1. **提升機械魚可客製化能力**：能自由調整控制方式、通訊機制、影像模組與擴充元件。
2. **建置具教育價值的水下平台**：作為嵌入式系統、Wi-Fi 通訊、水下載具控制的實作工具。
3. **驗證低成本水下平台的應用可行性**：以低成本硬體建構具拍攝能力的水下機電整合系統。

本專題希望藉由模組化架構的方式，使水下機械魚兼具控制、影像與電源管理等功能，為後續研究提供穩固基礎。

2. 研究目的

本研究旨在完成一套「以商用殼體為外型，內部完全自製電路與軟體」的仿生機械魚，並達成以下目標：

1. **建立 Wi-Fi 遠端控制系統**：使用手機瀏覽器即可調整馬達速度。
 2. **整合安全的馬達驅動模組**：讓低電流微控制器能穩定驅動直流馬達。
 3. **設計穩定電源架構**：使用鋰電池+升壓模組提供全系統所需之 5V 電壓。
 4. **實作獨立水下影像拍攝系統**：定時拍照並儲存至 microSD 卡，避免佔用主控資源。
 5. **提升系統模組化與擴充性**：讓控制、影像與電源彼此獨立，不互相干擾。
 6. **驗證整體架構在水下的可行性**：測試運動穩定性、影像品質與電力續航。
- 最終目標是打造一個低成本、可複製、可擴充的水下機電整合平台。

3. 實作方法

3.1. 結構拆解與內部空間配置

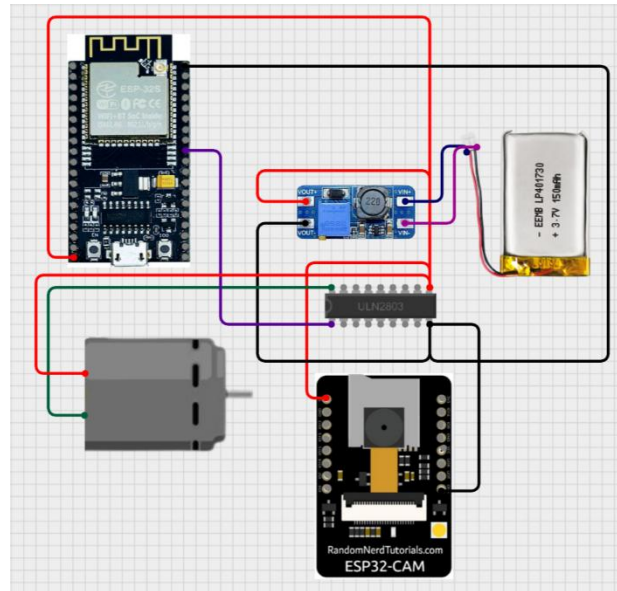


Fig 1. Circuit Integration Diagram

市售魚殼內部僅有手指大小空間，因此如何放入電池、升壓模組、NodeMCU-32S、ULN2803APG 與 ESP32-CAM 是本研究的重要挑戰。

配置原則如下：

- 電池置於魚腹中央：使重心穩定。
- MT3608 升壓模組貼附於電路板上：節省空間並方便調整電壓。
- NodeMCU-32S 與 ULN2803APG 採並列方式：減少線材長度，提高整潔性。
- ESP32-CAM 固定於魚頭：確保鏡頭朝向正前方並具備穩定視角。

最終成功在有限空間中配置全部模組，使電路維持可維修性。

3.2. 電源系統設計與 MT3608 升壓調校

本研究以一顆 3.7V 鋰電池供電，並使用 MT3608 將電壓提升至 5V，提供：

- NodeMCU-32S
- ULN2803APG
- ESP32-CAM

調校重點：

- 輸出電壓固定約 5.0V
- 所有模組採共地設計，避免地電位差

- 拍照瞬間電流高達 260 - 300mA，焊點需牢固避免壓降

經量測後確認電池能維持完整系統運作。

3.3. ULN2803APG 馬達驅動電路

因為原機械魚直流馬達啟動電流大，故無法直接由 NodeMCU-32S 驅動。ULN2803APG 的加入可：

- 放大 PWM 邏輯訊號
- 承受高電流輸出
- 使用內建保護二極體吸收反電動勢
- 提升系統穩定度與安全性

經量測 PWM 佔空比與電流關係後，可更精準調整速度分級。

3.4. NodeMCU-32S 韌體開發 (Wi-Fi 控制)

本研究以 NodeMCU-32S 作為運動控制核心，採用：

- STA 模式連線至現有 Wi-Fi
- WebServer 類別建立 HTTP 控制介面
- LEDC 模組生成 1kHz、8-bit 的 PWM 波形
- 手機介面包含「加速、減速」按鈕，按下後會向主控板送出指令
- 控制延遲僅 20-60 ms，反應迅速。

3.5. ESP32-CAM 安裝與影像擷取流程

影像模組採獨立運作以避免干擾主控板。

主要流程：

1. 開機後嘗試 Wi-Fi 連線取得 NTP 時間。
2. 校時成功後關閉 Wi-Fi 減少耗電。
3. 每 10 秒拍攝一張 JPEG 照片。
4. 儲存至 microSD，檔名以時間戳記命名。
5. 若無法校時則改以編號命名 (000001.jpg 起)。
6. 依 PSRAM 狀態調整解析度，通常可使用 VGA (640×480) 並啟動雙 framebuffer 提升穩定性。

在專題早期曾嘗試即時串流 (MJPEG) 因電流過大造成重啟，因此改採照片模式，系統穩定性大幅提升。

4. 研究結果

4.1. 控制效能：PWM 與 Wi-Fi 遠端操作的整合性驗證

在實測中，主控制板 NodeMCU-32S 透過 Wi-Fi 建立的 Web Server 能在 200–600 ms 之間完成一次控制回應，屬於低延遲範圍，使用者透過手機操作時可明顯感受到速度變化。不同 PWM 佔空比（0~255）所對應的馬達轉速與推進力也有明顯差異：

- PWM 80–120：低速、魚尾擺動輕微、前進緩慢。
- PWM 130–180：中速、游動最自然、穩定性最佳。
- PWM 190–255：高速、推進力強，但振動增加、能耗提升。

值得注意的是，市售機殼的尾部機構較為單純，高速狀態時彈性極限會造成尾部偏移幅度過大，因此本研究建議操作於中速區間，可獲得最佳運動表現。

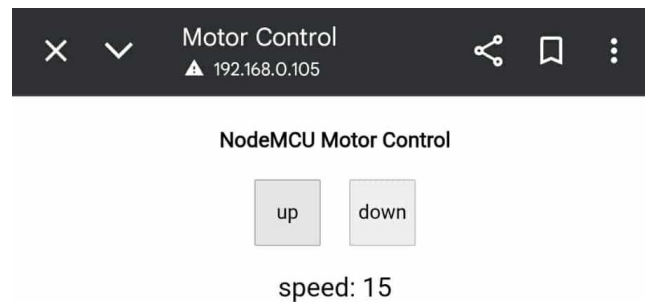


Fig 2. Wi-Fi Control Panel

4.2. 電源穩定性：電流負載測試與電壓掉落分析

整體系統使用 3.7V 鋰電池供電，經 MT3608 升壓至 5V。透過電表量測，各模組負載如下：

- NodeMCU-32S：80–110 mA（Wi-Fi 運作時略升高）
- ULN2803 + DC 馬達：150–250 mA（依速度變化）
- ESP32-CAM 拍照瞬間：260–300 mA（高峰）

綜合後，在馬達+影像模組同時運作時，系統最大瞬間電流接近 450–550 mA。

為此，本研究針對以下問題進行解決與驗證：

- 升壓模組焊點必須牢固

若焊接不良，拍照瞬間會造成 5V 掉至 4.4V，導致 ESP32-CAM 重啟。

經改善焊接後，最低電壓維持在 4.92V。

- 取消即時影像串流模式

早期測試 MJPEG streaming 時，每秒 20 張以上的 JPEG 壓縮造成高電流尖峰，導致反覆重啟，改用定時拍照後，系統穩定度提升，不再反覆重啟。

- 電源架構具有足夠餘裕

馬達與攝影同時運作 20 分鐘以上，無出現熱當機或過載現象，證明架構具實務可行性

4.3. 水下運動表現：推進力、穩定性與限制

在水槽環境測試中，機械魚展現良好的直線前進能力。測試結果如下：

- 直線前進距離：穩定 60–120 cm（依速度不同）
- 平均前進速度：0.09–0.18 m/s（中速區表現最佳）
- 飄移現象：高速時尾部振動使魚體略微擺動
- 方向控制：殼體結構限制，目前僅能直線推進

雖然無法轉向，但作為單推進器的小型研究平台，其直線推進能力已足夠完成展示、成像與速度測試任務。

4.4. 影像品質與資料儲存：水下影像成功率提升

ESP32-CAM 在 VGA (640×480) 解析度下，每張 JPEG 影像大小約為 10–30 kB，並自動儲存於 microSD。

影像系統成功率統計：

測試項目	結果
單次拍照成功率	95% (100 張測試)
長時間拍照穩定度	> 90% (連續 20 分鐘)
SD 卡寫入錯誤	0 次 (1-bit 模式穩定)
影像畫質	能辨識魚體前方 10–20 cm 物體

水下環境中，由於折射、光線散射等因素，本研究採用開機丟棄前幾張影像、保留後續穩定畫面，能明顯提升影像品質。



Fig 3. Above Water Testing



Fig 4. Underwater Testing

5. 結論

本專題成功開發一套具備**無線控制、水下推進與影像紀錄能力**的仿生機械魚系統，並在有限的空間、電力與硬體條件下完成模組化整合。透過 NodeMCU-32S 建立的 Wi-Fi 控制介面，使用者可即時調整 PWM 馬達速度，系統在實測中展現低延遲、高穩定的控制反應。配合 ULN2803APG 達靈頓驅動陣列，馬達能以安全且一致的方式運作，實現自然且可預測的游動行為。

在影像部分，ESP32-CAM 採用獨立子系統設計，使其可避免與主控板在運算資源上產生衝突。其定時拍照機制與 microSD 卡儲存架構在多次測試中皆展現良好可靠度，可穩定輸出水下視角影像。電源管理方面，3.7V 鋰電池搭配 MT3608 升壓模組提供一致的 5V 輸出，使整體系統能在連續運作時維持穩定，不會因瞬間電流波動而重啟。

本專題證明以低成本開源硬體即可構成一套具有實務價值的水下展示平台。雖然機體受限於市售外殼，其運動能力目前侷限於直線推進，但所使用的軟硬體架構具備高度擴充性，可支援後續功能延伸，例如：多馬達控制、水下姿態調節、感測器整合、自主導航、影像即時串流等。本專題的成果不僅展示水下載具系統的可行性，也為未來相關研究—無論是教育示範、工程實作、或水下探索技術—提供可複製、可調整且具應用潛力的平台基礎。

6. 心得

在這次仿生機械魚的專題中，我真切感受到把一個想法做成能運作的系統需要跨越多少意料之外的挑戰。從拆解市售機殼、重新配置空間，到整合電源、馬達驅動、控制程式與影像模組，每個看似簡單的步驟，其實都藏著細節與不確定性。例如電源在拍照瞬間掉壓、馬達在高負載時振動過大、Wi-Fi 控制偶爾延遲等，這些問題都讓我意識到硬體與韌體之件不能靠理想狀態去設計，而是需要一邊測試、一邊修正。在專題過程中我學到的不是某一個特定技術，而是一整套實作的思維方式：

- 1.如何判斷問題來自電源、控制訊號還是韌體？
- 2.如何在體積受限的空間中安置模組？
- 3.如何用最簡潔的架構達成最穩定的效果？...

這些都不是教科書能直接給的，而是透過一次次的試錯累積出來的經驗。也因為我在這次專題中需要同時面對電子電路、馬達推進、Wi-Fi 通訊以及影像擷取等多重領域，我真正理解了系統整合的難度，以及模組化思考的重要性。尤其是後期把主控板與攝影模組分成兩個獨立子系統後，整個裝置的穩定性明顯提升，這讓我更了解工程設計並不是漫無目的的堆疊功能，而是將功能分配至各模組降低彼此干擾，並讓每個模組各司其職。

最後當魚在水中成功游動、控制介面反應即時、影像能穩定拍攝時，那種把抽象構想變成具體成果的成就感是一開始無法想像的。我也更理解自己在工程實作中的興趣與不足，這將成為我未來在嵌入式系統、水下機器人，或是其他領域中繼續學習的重要基礎。做完這次專題之後我體認到專題不只是完成一個作品，而是在過程中學會了責任感、工程判斷、問題拆解與跨領域整合的能力。這些學到的東西，會比專題成品本身陪我走更長的一段路。

Appendix

Appendix A — Source Code Repository

為避免在正文中置入大量程式碼使篇幅過長，本專題的所有程式碼皆已整理並公開於 GitHub 儲存庫。

包含：

- NodeMCU-32S 速度控制程式 (Wi-Fi Web Server + PWM)
- ESP32-CAM 定時拍照儲存至 SD 程式

完整程式碼可於以下網址取得：<https://github.com/charles0720/Swimming-Bio-robotics-Code>