

OAC-114-024(研究報告)

小型的海草移植水下無人載具模型  
(成果報告)

海洋委員會補助研究

中華民國 114 年 8 月

「本研究報告僅供海洋委員會施政參考，並不代表該會政策，該會保留採用與否之權利。」

OAC-114-024(研究報告)

## 小型的海草移植水下無人載具模型

(成果報告)

學校：國立高雄科技大學

指導教授：許介璋

學生：劉茵姍

研究期程：中華民國 114 年 4 月至 114 年 10 月

研究經費：新臺幣 96,500 元

海洋委員會補助研究

中華民國 114 年 8 月

「本研究報告僅供海洋委員會施政參考，並不代表該會政策，該會保留採用與否之權利。」

「本研究報告絕無侵害他人智慧財產權之情事，如有違背願自負民、刑事責任。」

# 目次

圖次.....	2
摘要.....	4
第一章 前言.....	5
第一節 研究起源.....	5
第二節 現況分析與問題背景 .....	5
第三節 研究目的與預期目標 .....	10
第四節 研究重點.....	10
第二章 研究方法及過程 .....	11
第一節 水下無人載具的設計.....	12
第二節 海草移植模組的設計.....	23
第三節 電路與程式設計.....	28
第四節 防水與整合測試.....	37
第三章 結果與討論 .....	41
第一節 結果.....	41
第二節 討論.....	42
第四章 結論.....	48
第一節 結論.....	48
第二節 未來展望.....	48
參考文獻.....	49

## 圖次

圖 1-1 潛水員將海草植入生物墊中 .....	5
圖 1-2 成功復育的海草 .....	6
圖 1-3 不同設置下釘固法的存活率百分比 .....	6
圖 1-4 不同設置下插塞的存活率百分比 .....	7
圖 1-5 移植作業平台 Optimal 880 型樹鏟式移植機 .....	7
圖 1-6 2024 年 7 月於拉沃托(Larvotto)進行的海草移植作業 .....	8
圖 1-7 海草移植的機器 ECOSUB1 的照片 .....	9
圖 1-8 一台 ECOSUB2(左)和一台草皮梭子(右) .....	9
圖 2-1 研究流程圖 .....	11
圖 2-2 ROV 左前方的模型圖 .....	12
圖 2-3 ROV 右後方的模型圖 .....	13
圖 2-4 ROV 的側板(右邊為 ROV 的前頭) .....	14
圖 2-5 ROV 的內部得上板(右邊為 ROV 的前頭) .....	14
圖 2-6 ROV 的內部得下板(右邊為 ROV 的前頭) .....	15
圖 2-7 3D 建模的水下推進器支架 .....	16
圖 2-8 實體的水下推進器支架 .....	17
圖 2-9 側面的 3D 建模訊號線固定支架 .....	18
圖 2-10 左後方的 3D 建模訊號線固定支架 .....	18
圖 2-11 實體的側面訊號線固定支架 .....	19
圖 2-12 實體的正上方訊號線固定支架 .....	19
圖 2-13 3D 建模的水下鏡頭固定支架 .....	20
圖 2-14 側面的實體水下鏡頭固定支架 .....	21
圖 2-15 正面的實體水下鏡頭固定支架 .....	21
圖 2-16 正面的實體水車輪 .....	22
圖 2-17 側面的實體水車輪 .....	22

圖 2-18 四連桿機構.....	23
圖 2-19 四連桿機構中挖取沉積物的位置圖 .....	24
圖 2-20 凸輪機構.....	25
圖 2-21 水草盒開口關閉的狀態 .....	26
圖 2-22 水草盒開口開啟的狀態 .....	26
圖 2-23 正面的實體擋板.....	27
圖 2-24 側面的實體擋板.....	27
圖 2-25 硬體設備連接圖.....	29
圖 2-26 防水盒水密測試.....	37
圖 2-27 將所有電路元件安裝於防水盒內進行測試.....	38
圖 2-28 防水盒與 ROV 本體組裝 .....	39
圖 2-29 ROV 準備進行測試.....	40
圖 3-1 最終成果.....	41
圖 3-2 四連桿機構中大齒輪帶動小齒輪 .....	42
圖 3-3 凸輪機構直接由凸輪驅動 .....	43
圖 3-4 嘗試於 ROV 前端上方加裝浮塊.....	44
圖 3-5 調整浮塊將其移至 ROV 第二層 .....	45
圖 3-6 進行浮塊微調.....	45
圖 3-7 使 ROV 能夠維持與水面平行的狀態.....	46
圖 3-8 推進器均能完全浸沒於水中 .....	46

## 摘要

### 關鍵字：水下無人載具、海草、藍碳固存

海草床復育對於恢復沿海生態系統、提升藍碳固存及生物多樣性具有重要意義。然而，現有的人力潛水移植方式存在高風險與作業範圍受限的問題，而大型機械化復育技術則因體積龐大與成本高昂而限制了應用。本研究旨在開發一款小型海草移植水下無人載具模型，藉此提供一種低風險且靈活的復育工具。

研究方法包含四大面向：一、水下無人載具的設計，採用壓克力板作為主要的結構，並搭配3D列印之零件使設備可以固定在壓克力板上；二、海草移植模組的設計，透過水車輪驅動四連桿與凸輪機構，完成挖掘、植入與覆蓋沉積物的動作；三、電路與程式設計，結合 Xbox的搖桿、Arduino UNO 板與 Python，並透過 RS485 模組驅動水下推進器；四、防水與整合測試，進行水密性、浮力與泳池操控之測試。

研究結果顯示，所開發之水下無人載具能穩定完成前進、後退、轉向及升沉等操作，並能驅動移植模組完成基本動作驗證。測試亦揭示限制，包括四連桿與凸輪機構之頻率不同步及缺乏前端視覺輔助。

綜合而言，本研究成功驗證小型水下無人載具搭載釘固法移植模組的可行性，提供海草復育一項新型低風險且具操作彈性的技術方案。未來若能進行實際海草移植測試並完成改良，將有助於推廣應用於受限環境下的藍碳復育工作。

# 第一章 前言

## 第一節 研究起源

海草床復育在恢復沿海生態系統和增強藍碳固存方面發揮了關鍵作用，對氣候變遷緩解和生物多樣性保育具有重要意義(Duarte et al., 2013; Serrano et al., 2021; Unsworth et al., 2019)。然而，近幾十年來，海草面積因人類活動如沿海開發、污染、水質惡化、氣候變遷等因素，持續出現大規模的減少與退化(Montefalcone, 2009)，為了復育水下生態，目前許多國家復育海草的方式是利用水肺潛水讓潛水員潛入海底進行海草的移植(Orth et al., 2006; Marre et al., 2019)。但是這樣的方法伴隨著風險。相關研究指出，潛水員死亡有 56%-86%是因為溺水(Lawrence and Cooke, 2006)，另一部分則是因為壓力性肺損傷及腦動脈氣栓。在高水動力環境的海域，仍是無法透過潛水人員復育的區域(Paling et al., 2001b)。為了降低降低潛水人員的風險以及增加可復育的區域，透過無人水下無人載具(underwater Remotely Operated Vehicle; ROV)進行海草移植與復育是重要的。

## 第二節 現況分析與問題背景

目前海草復育的方法主要可分為人力移植與機械化輔助兩大類。在人力移植中，一項研究採用椰纖墊結合海草切片，透過潛水人員將面積為 1 平方公尺的椰纖墊分別移植於兩個區域(圖1-1)，復育各 50 平方公尺的海床。經兩個月的監測(圖1-2)，復育成功率分別為約 66%(PAG 區)與 39%(CRE 區)(Chimienti et al.,2023)。



圖 1-1 潛水員將海草植入生物墊中

資料來源( Chimienti et al.,2023)



圖 2-2 成功復育的海草

資料來源(Chimienti et al.,2023)

另一項利用人力移植的研究，是使用了兩種不同的復育方式(圖1-3、圖1-4)，分別為釘固法(sprigs)和插塞法(plugs)，並在個別使用這些方法時，分別再以裸露種植跟塑膠網格固定種植來做比較，研究發現在使用釘固法(sprigs)裸露種植跟網格固定種植時，對於海草長期的存活率來說並沒有太大的差異(圖1-3)，而在使用插塞法(plugs)時有明顯的差異，使用網格固定的海草存活率較高(圖1-4)，但是這項差異只有在大於10公分以上的海草苗才能顯現出來(Van Keulen et al.,2003)。

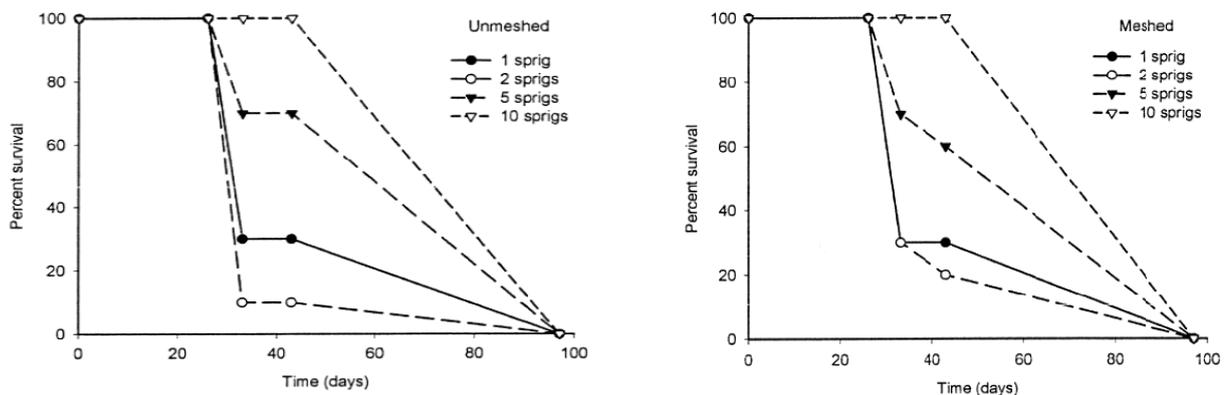


圖3-3 不同設置下釘固法的存活率百分比

說明：(左) 無網處理；(右) 有網處理

資料來源(Van Keulen et al.,2003)

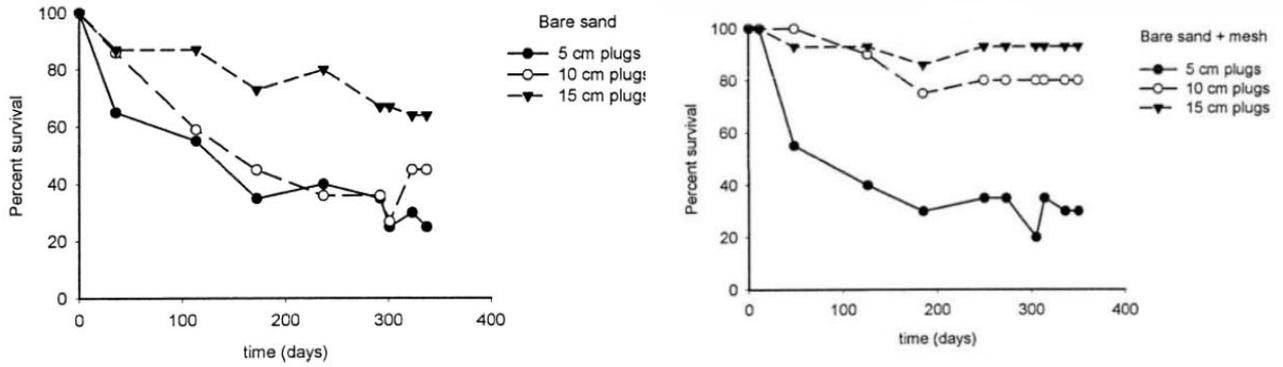


圖4-4 不同設置下插塞的存活率百分比

說明：(左) 移植至裸砂地；(右) 移植至帶有網覆的裸砂地。

資料來源(Van Keulen et al.,2003)

在機械化輔助移植中，一項研究應用改裝的挖掘機(圖1-5)，搭配直徑約80公分的海草皮移植單元，將完整的海草草皮(含根系與底泥)直接植入海床，成功復育超過1公頃(10,000 平方公尺)的海草棲地(圖1-6) (Descamp et al. ,2025)。

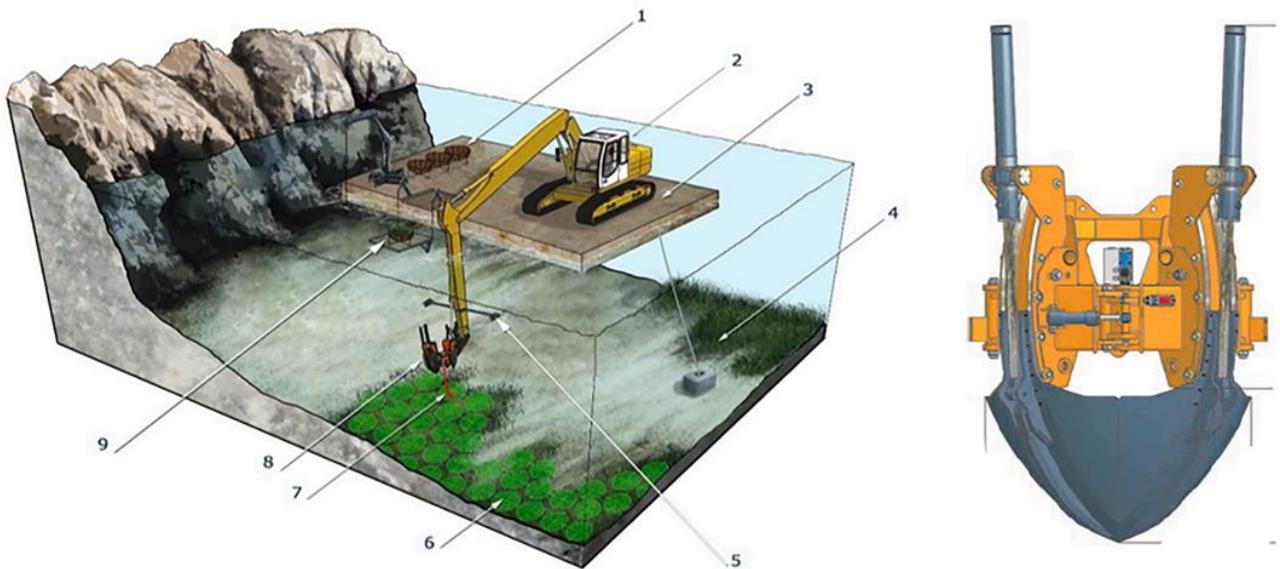


圖 5-5 移植作業平台 Optimal 880 型樹鏟式移植機

說明：(1)運輸籃；(2)長臂挖掘機；(3)頂升式作業平台；(4)平台固定錨腳；(5)監控攝影機；  
(6)有標記的預定移植草皮；(7)雷射指示器；(8)移植機；(9)籃子升降裝置。

資料來源(Descamp et al. ,2025)



圖 6- 6 2024 年 7 月於拉沃托(Larvotto)進行的海草移植作業

資料來源(Descamp et al. ,2025)

另一項研究澳洲西部的研究亦投入機械化海草復育技術。研究人員開發水下無人載具 ECOSUB1(圖1-7),可提取與移植面積 0.25 平方公尺、深度 0.5 公尺的海草草皮,並於 Success Bank 海域進行復育。平均存活率達 57%,且持續穩定超過兩年(Paling et al., 2001b)。雖然單次操作時間不到 1 小時,但整體部署與運輸耗時可達 3 小時,且 ECOSUB1 體積龐大(5m × 3m × 3m),在水流中易產生機械應力,影響作業效率(Paling et al., 2001a)。為提升效率,後續開發了 ECOSUB2,改以分工式流程進行復育:由一台 ECOSUB2(圖1-8)執行採集,並將草皮裝入運輸梭後運送至受體地點,再由另一台 ECOSUB2 進行移植。此方法每日理論可移植 40 平方公尺,並有更高存活率(Paling et al., 2001a)。然而,ECOSUB1 與 ECOSUB2 雖具成效,但仍因體積大與適用範圍侷限,主要適用於開闊且海流較強的區域,不利於小範圍復育需求。



圖 7-7 海草移植的機器 ECOSUB1 的照片

資料來源((Paling et al., 2001a)



圖 8-8 一台 ECOSUB2(左)和一台草皮梭子(右)

資料來源(Paling et al., 2001a)

然而，這兩種方法各自面臨不同限制：人力移植雖具精細性與環境適應性，但需潛水員執行，存在作業風險與人力負荷問題；而機械化操作雖可進行大規模復育，但是成本較高且體積大，適用範圍受限於水深與地形條件。

### 第三節 研究目的與預期目標

為了解決人力潛水作業的風險，並降低機械設備在海草復育上的成本負擔，本研究將設計一台小型的海草移植水下無人載具模型，具備良好的水下機動性與操作靈活性，適用於復育有限制範圍的海域。載具將搭配釘固法海草移植結構，實現準確且穩定的移植作業，期望提供一種低風險、高靈活的復育方案。

本研究之預期目標如下：

1. 設計利用釘固法移植海草的機械結構。
2. 開發小型的移植海草水下無人載具模型。

### 第四節 研究重點

本計畫的研究重點在於將水草移植模組整合至水下無人載具(Remotely Operated Vehicle ; ROV，本計畫以下簡稱ROV)中，並透過 ROV 前進時所產生的動力驅動該模組運作。計畫中將設計一組四連桿機構與凸輪機構作為水草移植模組的核心，使其能夠挖開水底沈積物、植入水草，再利用擋板將沈積物覆蓋回孔中，以完成水草的移植作業。在 ROV 的控制設計方面，將採用 Xbox 搖桿搭配 Arduino Uno 控制板操作水下推進器，以確保 ROV 能穩定移動並順利帶動移植模組。同時，ROV 上也將搭載水下攝影鏡頭模組，使操控者能即時觀測前方環境與水草移植狀況，提升操作的精準度與成效。

## 第二章 研究方法及過程

此計畫的研究流程，將包含以下四項；(一)水下無人載具的設計、(二)水草移植模組的設計、(三)電路與程式設計、(四)整合測試與驗證，以下就上述的四個項目作研究方法與過程的詳細說明：

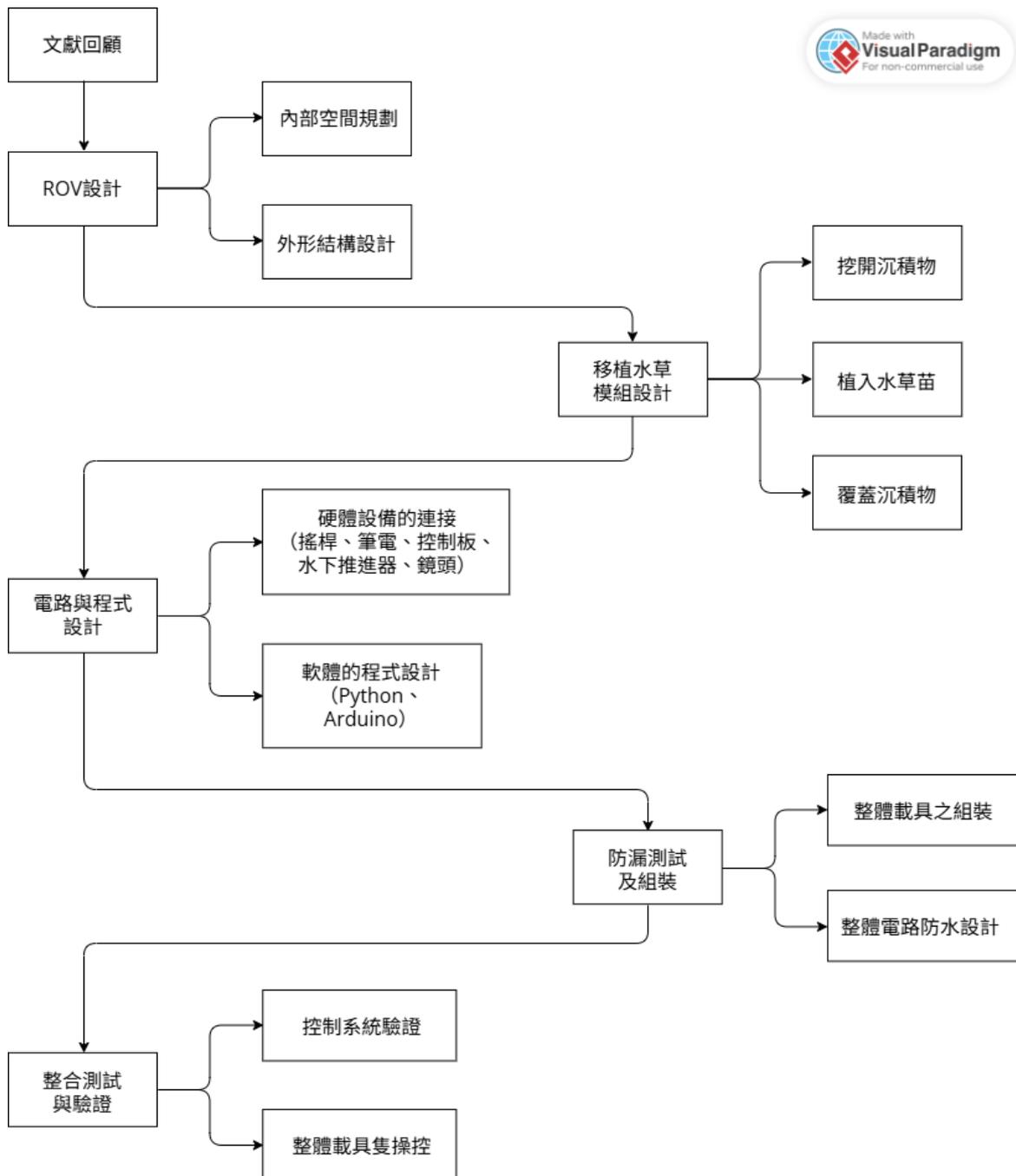


圖 2-1 研究流程圖

資料來源(本計畫繪製)

## 第一節 水下無人載具的設計

### 一、ROV的主體設計

本研究所設計之 ROV 外型結構分為三層，整體材料採用厚度 10 mm 的壓克力板製作，主要構件包括兩片側板以及內部的上板與下板。最上層配置泡棉浮塊如圖2-2中的藍色部分(圖2-2)，用以調整並控制 ROV 在水中之浮力。中間層則安排電力相關設備如圖2-2中的黃色部分(圖2-2)，包括水下推進器、水下攝影系統及主控板等，以支援 ROV 的動力及控制功能。最下層則專門預留水草移植模組之安裝空間如圖2-2中的綠色部分(圖2-2)，以完成水下作業任務。

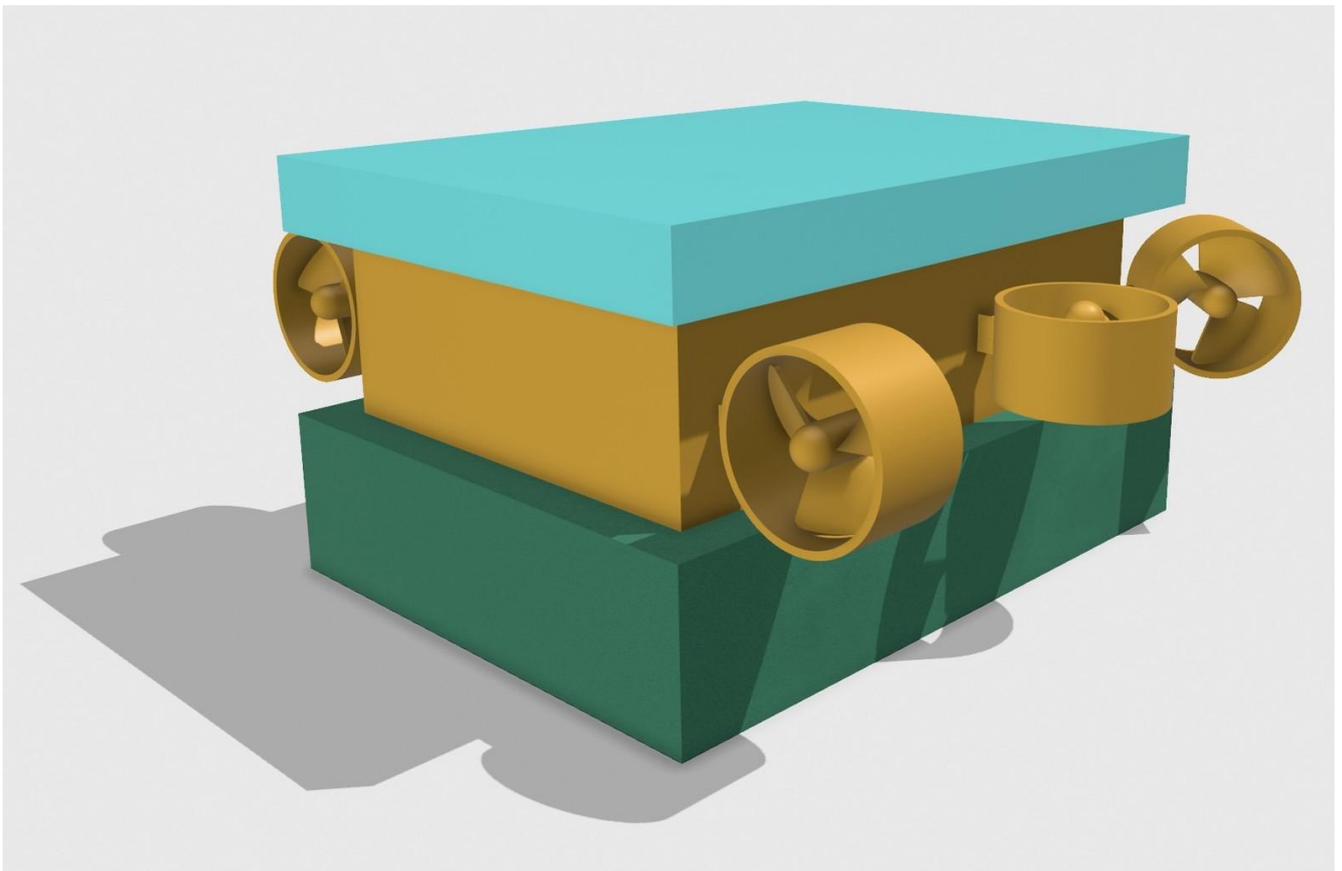


圖 2-2 ROV 左前方的模型圖

資料來源(本計畫繪製)

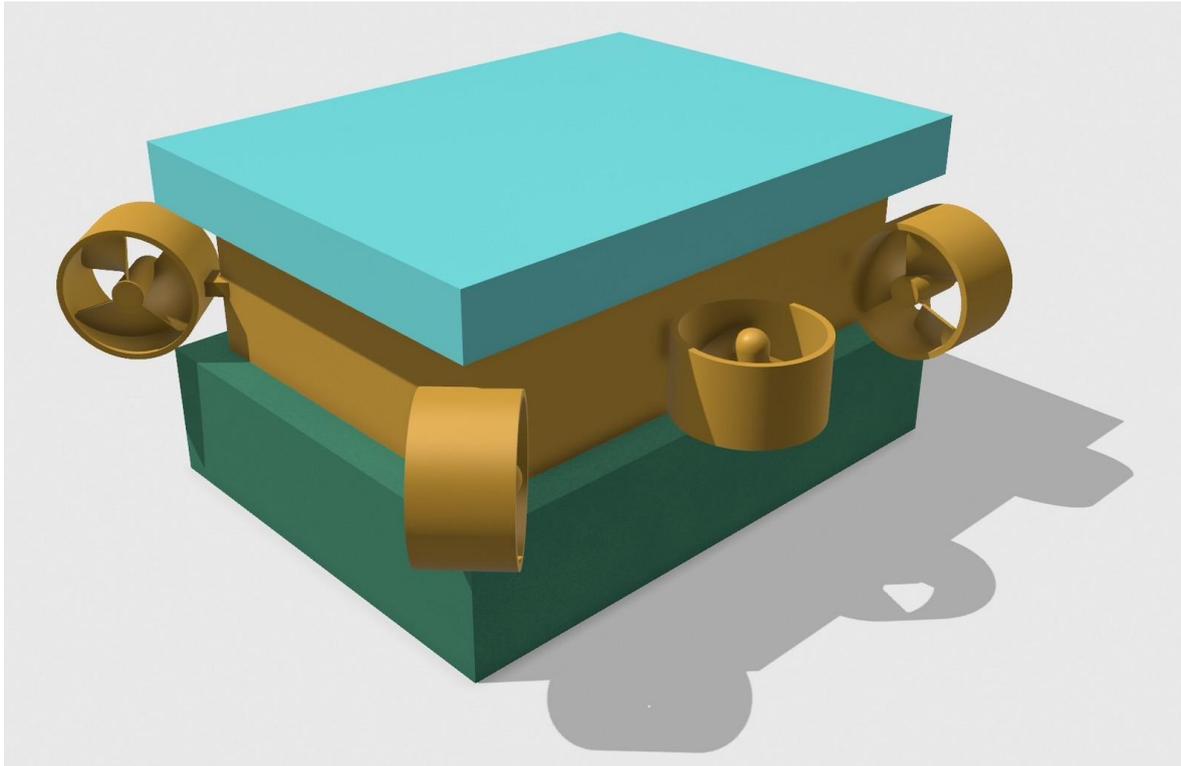


圖 2-3 ROV 右後方的模型圖

資料來源(本計畫繪製)

側板(圖 2-4)設計上，上層設有固定用開孔，中層開孔用於安裝水下推進器及其電路，下層則為水草移植模組的組裝孔位。上板(圖 2-5)兩側設有與側板固定之開孔，前方大部分區域配置泡棉浮塊之固定孔，後方則設有固定 30 米電纜之孔位。下板(圖 2-6)兩側亦設有與側板固定之開孔，前方用於水草移植模組的安裝，後方設置防水盒，以安置主控板、電子變速集成板等需供電之設備，而板尾端則固定水下攝影系統的位置，以完成整體功能配置。

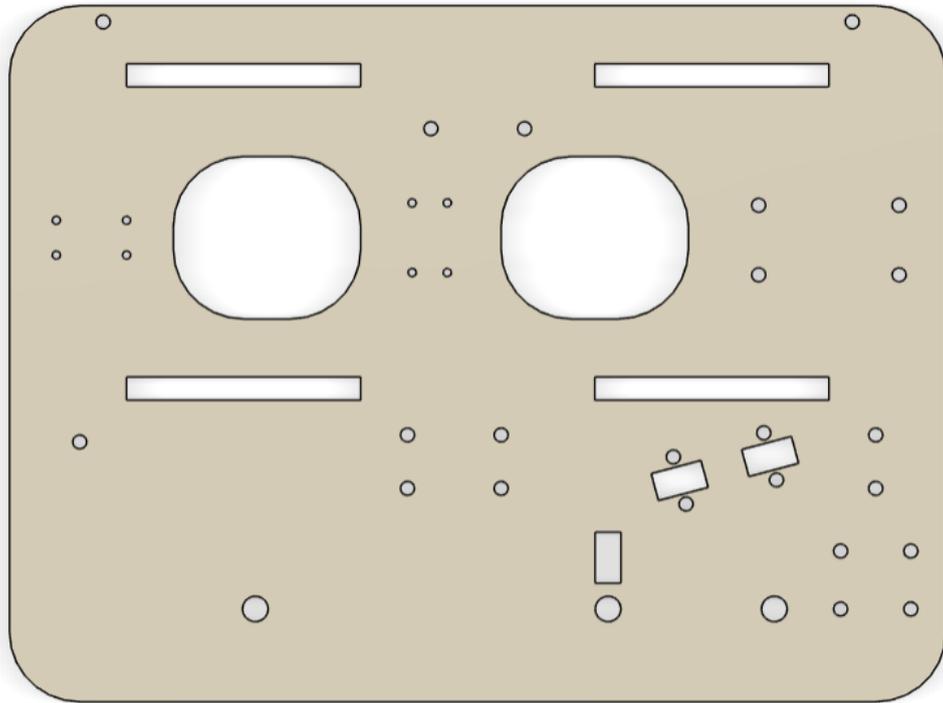


圖 2- 4 ROV 的側板(右邊為 ROV 的前頭)

資料來源(本計畫繪製)

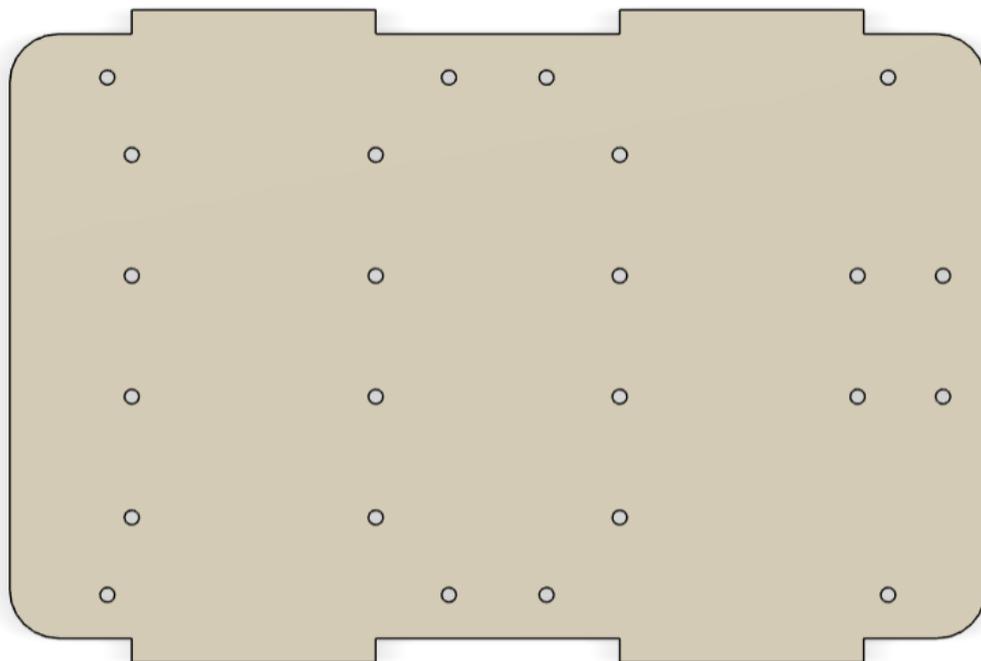


圖 2- 5 ROV 的內部得上板(右邊為 ROV 的前頭)

資料來源(本計畫繪製)

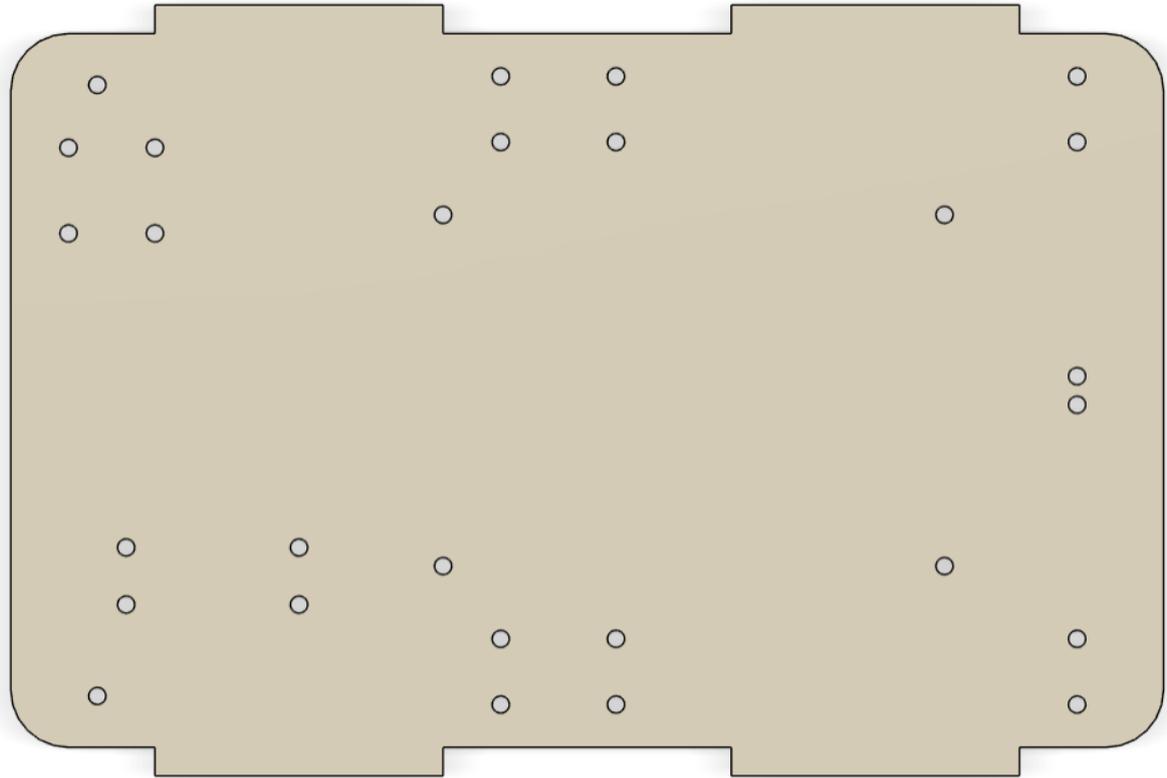


圖 2- 6 ROV 的內部得下板(右邊為 ROV 的前頭)

資料來源(本計畫繪製)

## 二、ROV 的零件設計

### (一)水下推進器支架

本研究所設計之 ROV 搭載六顆水下推進器，分別負責控制其前後、上下及左右方向的運動。然而，在執行左右轉向時，若推進器僅以水平方式安裝，則只能產生前進或後退的推力，無法有效產生旋轉力矩。為解決此限制，本設計特別在推進器安裝結構上進行改良，透過固定支架使推進器向外傾斜 25 度。經由此角度的配置，當推進器正轉時，ROV 即可向左轉動；反之，當推進器反轉時，則可使 ROV 向右轉動，藉此達成靈活的方向控制。

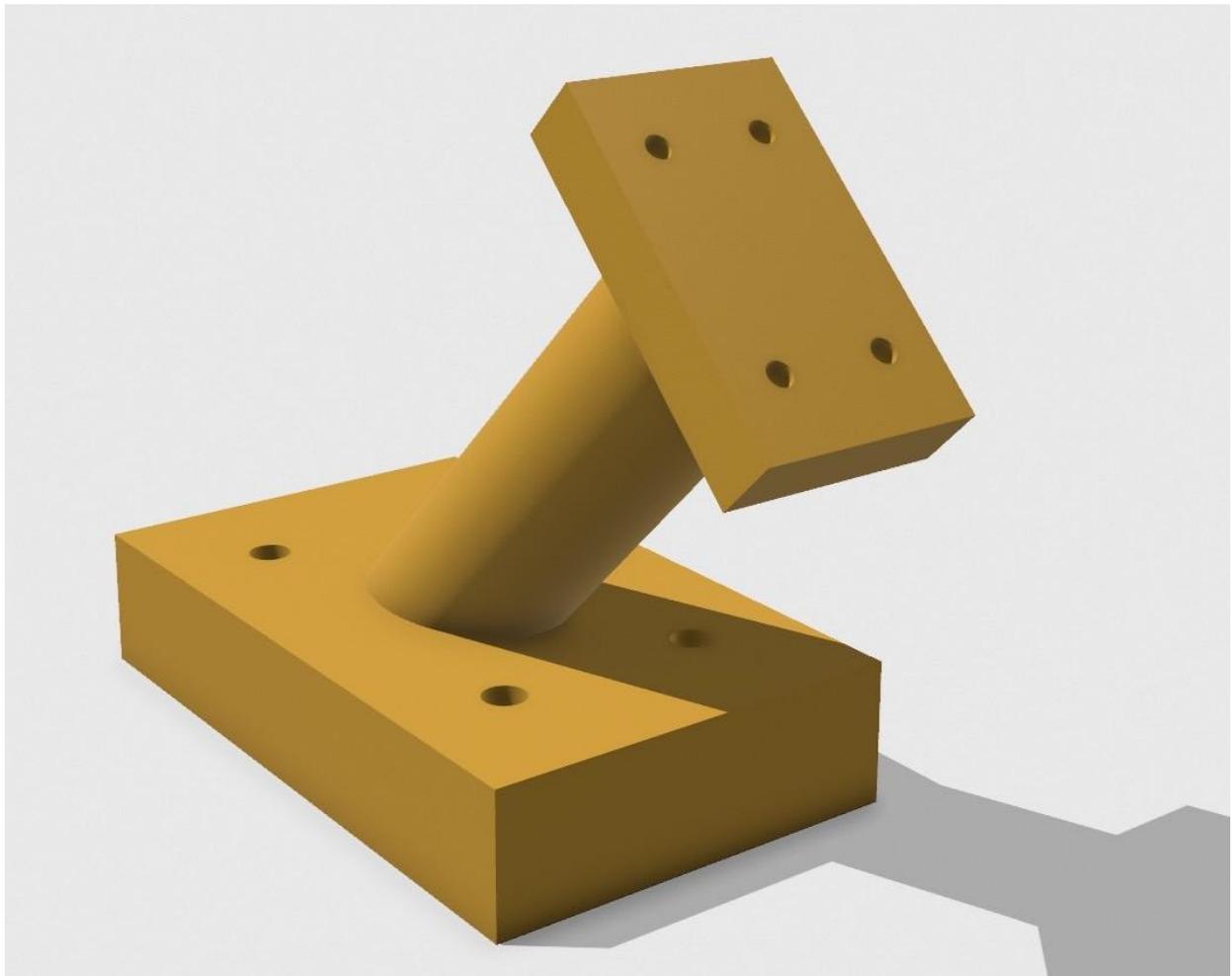


圖 2-7 3D 建模的水下推進器支架

資料來源本計畫繪製)



圖 2-8 實體的水下推進器支架

資料來源(本計畫拍攝)

## (二)訊號線固定支架

在操作 ROV 的過程中，若未設置訊號線固定支架，訊號線容易隨水流擺動，進而造成 ROV 與訊號線發生纏繞，或在操控時直接對 ROV 中訊號線接點的位置產生拉扯，增加接點斷裂的風險。為避免此問題，本設計於 ROV 上加裝訊號線固定支架，使訊號線在接點處有一段 U 型緩衝，藉此降低外力拉扯的影響，提升線材的耐用性與系統操作的可靠度。

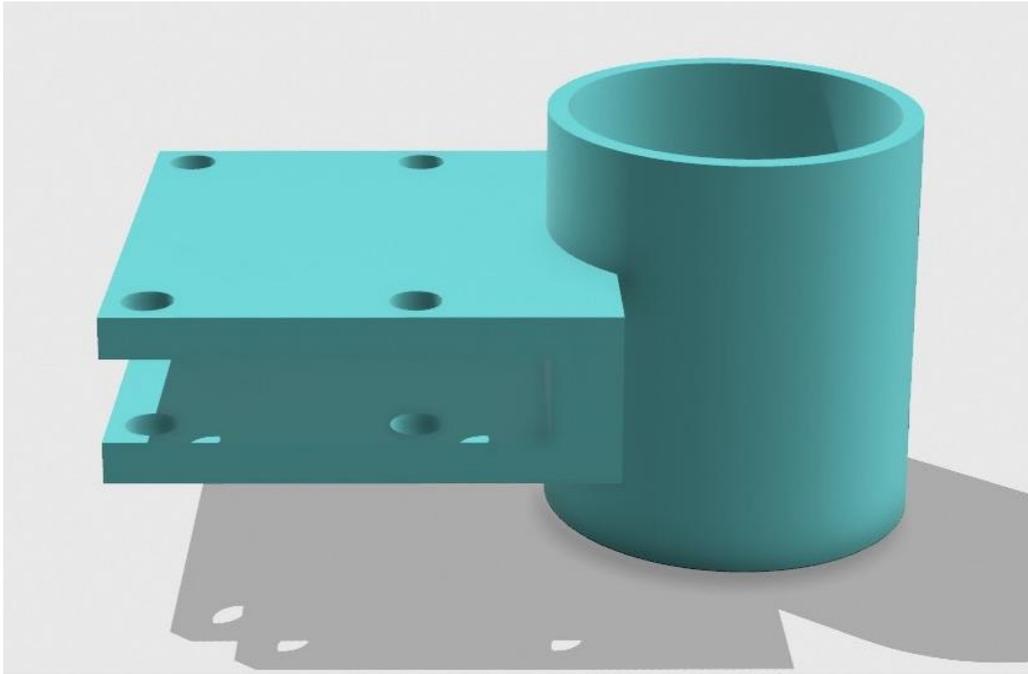


圖 2-9 側面的 3D 建模訊號線固定支架

資料來源(本計畫繪製)

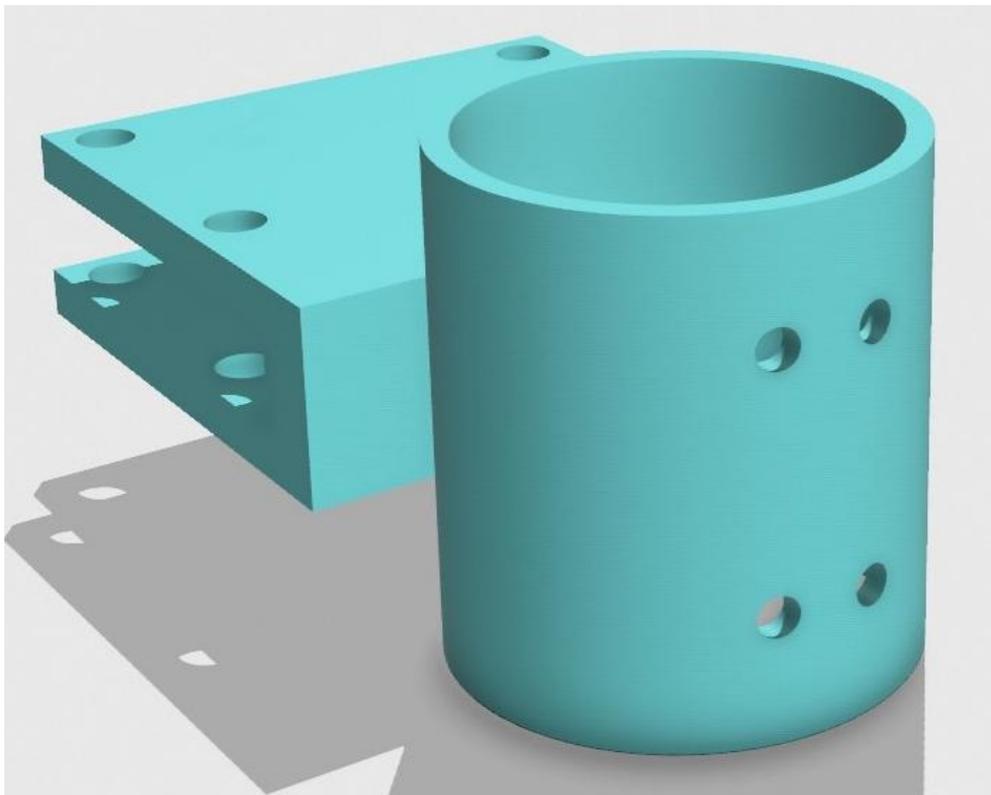


圖 2-10 左後方的 3D 建模訊號線固定支架

資料來源(本計畫繪製)



圖 2- 11 實體的側面訊號線固定支架

資料來源(本計畫拍攝)

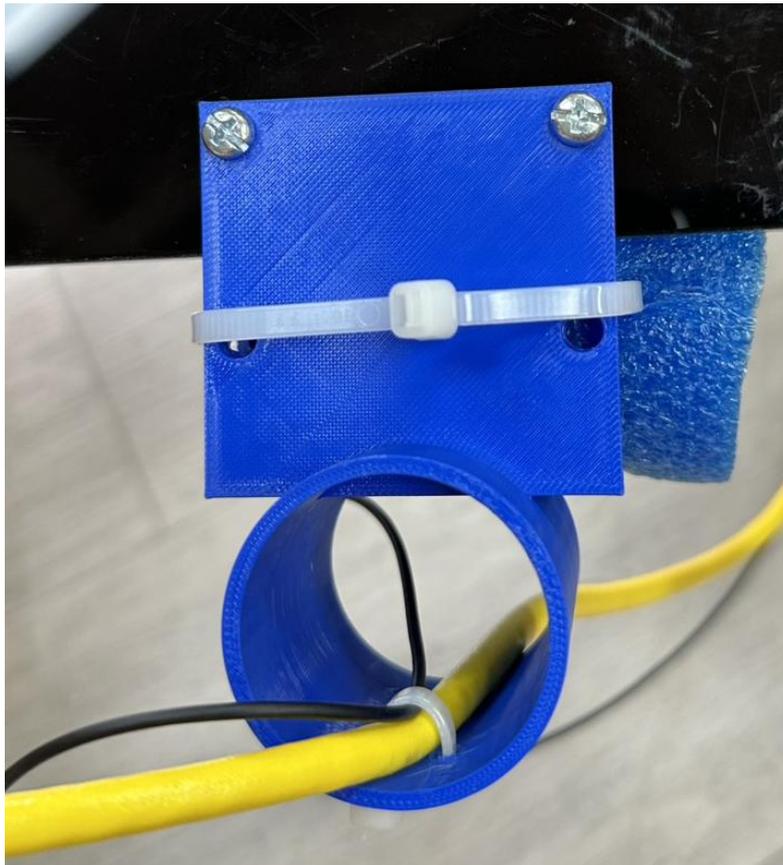


圖 2- 12 實體的正上方訊號線固定支架

資料來源(本計畫拍攝)

### (三)水下鏡頭固定支架

為了確保 ROV 在操控過程中的穩定性，並能即時監控海草移植模組的運作狀態，因此將水下鏡頭安裝於 ROV 第二層的後段位置，使其具備向後並朝下的視角，以便清楚掌握模組的工作情形。



圖 2- 13 3D 建模的水下鏡頭固定支架

資料來源(本計畫繪製)

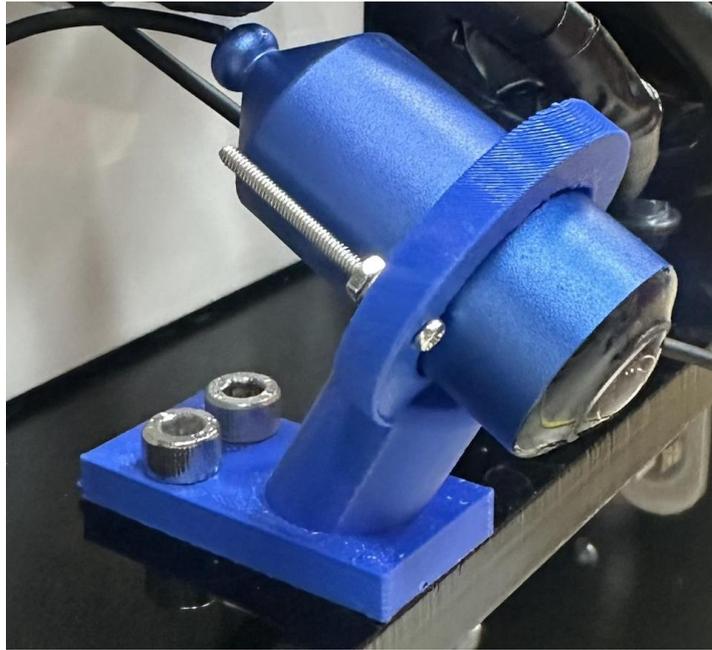


圖 2- 14 側面的實體水下鏡頭固定支架

資料來源(本計畫拍攝)

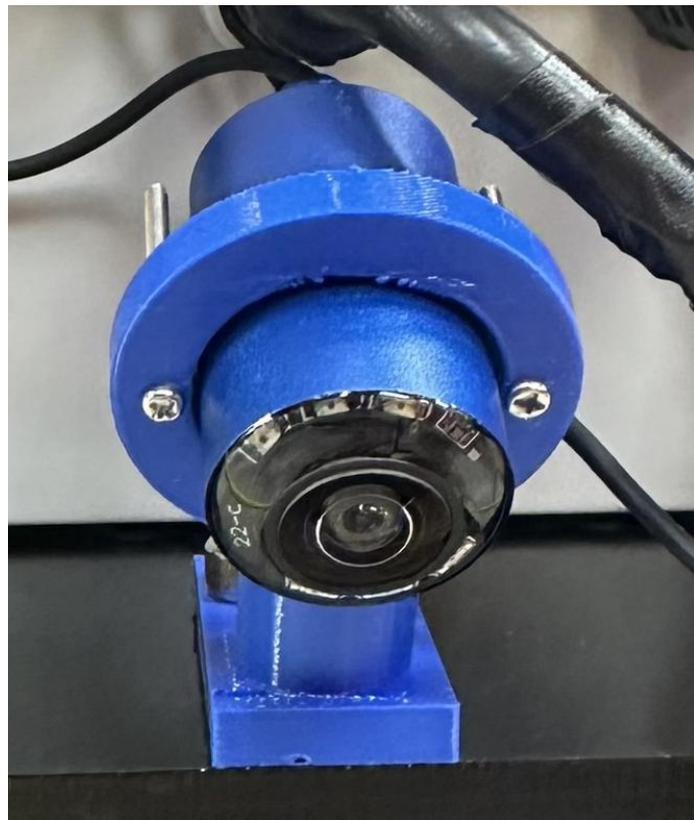


圖 2- 15 正面的實體水下鏡頭固定支架

資料來源(本計畫拍攝)

#### (四)水車輪設計

本研究之水車輪設計係參考現有市售水車輪，其主要功能在於當 ROV 接觸至沉積物時，水車輪能嵌入其中，並隨著 ROV 前進而帶動海草移植模組的機械結構運作。

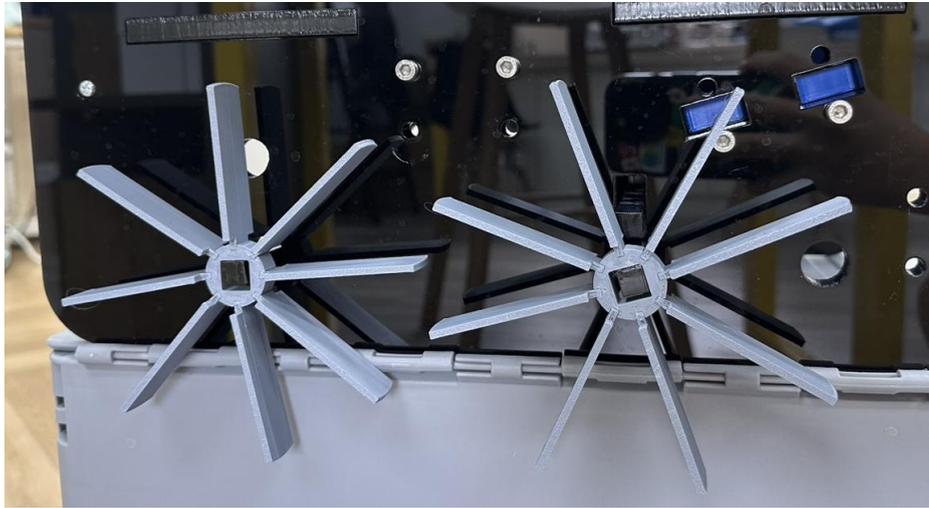


圖 2- 16 正面的實體水車輪

資料來源(本計畫拍攝)



圖 2- 17 側面的實體水車輪

資料來源(本計畫拍攝)

## 第二節 海草移植模組的設計

在水草移植模組中將利用插秧機的結構來設計，插秧機在農業中扮演著與本研究相似的角色，不僅能快速地進行大面積種植，還能精準控制株距，插秧機主要是透過輪胎向前推進時，帶動連桿系統來完成插秧的動（黃國益、林圣泉,1995）。在移植水草時，根據不同種類的水草需挖開沈積物至3-10公分的深度，再將水草苗植入，水草苗之間的株距需在在10-15公分，最後再將沈積物覆蓋回去，因此將模組分成了三個部分，分別為一、挖開沈積物、二、植入水草苗、三、覆蓋沈積物，以下就上述的三個項目做詳細的說明：

### 一、挖開沈積物

在ROV前方底部設計並安裝一組四連桿機構，由兩顆齒輪驅動。四連桿中包含驅動桿、固定桿、連桿及從動桿，其長度分別為5、10、9及11公分。當ROV向前行進時，前方的水車輪會驅動第一顆齒輪逆時針旋轉，進而帶動第二顆齒輪順時針旋轉。齒輪旋轉後推動連桿中的從動桿，使從動桿產生向下擺動的循環運動，最大可向下擺動至5公分的深度，從而在沈積物中完成開孔的動作，如影片A所示(本計畫拍攝)。

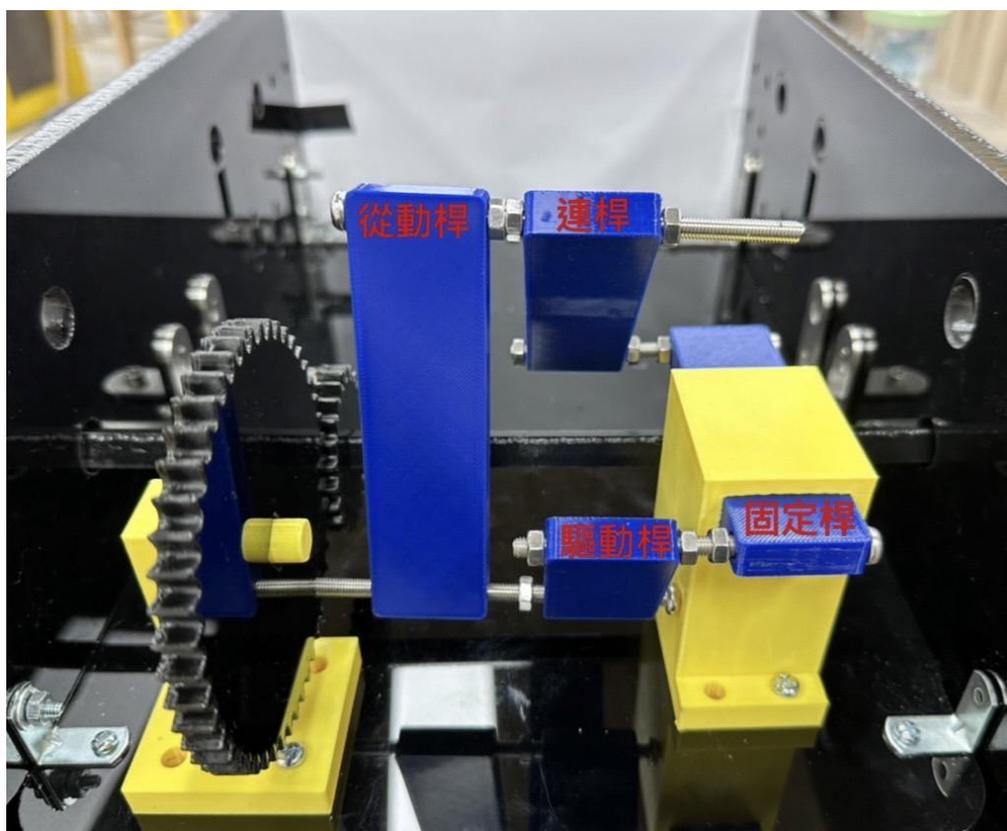


圖 2- 18 四連桿機構

資料來源(本計畫拍攝)

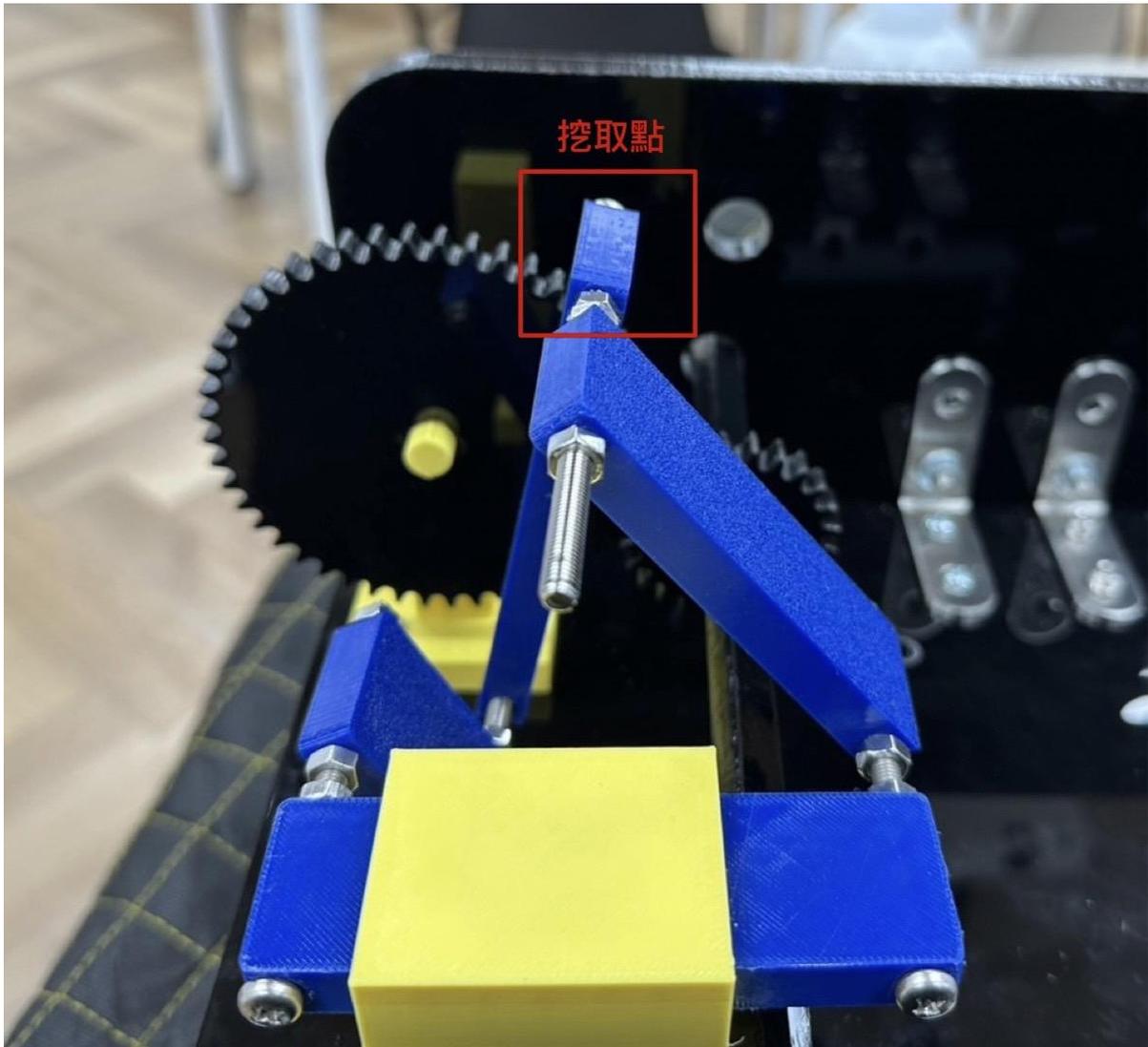


圖 2- 19 四連桿機構中挖取沉積物的位置圖

資料來源(本計畫拍攝)

## 二、植入水草苗

在 ROV 後方底部安裝一組凸輪機構及一個水草盒，水草盒的長、寬、高分別為120mm、100mm與30mm，此凸輪機構中的凸輪是直接與水車輪相接，當ROV向前行進時，後方的水車輪會驅動凸輪，當凸輪的凸邊轉到最頂端時水草盒的擋板會被上推20mm，讓水草苗可以從水草盒的開口中植入沉積物，如影片B所示(本計畫拍攝)，進而達成移植的目的。



圖 2- 20 凸輪機構

資料來源(本計畫拍攝)

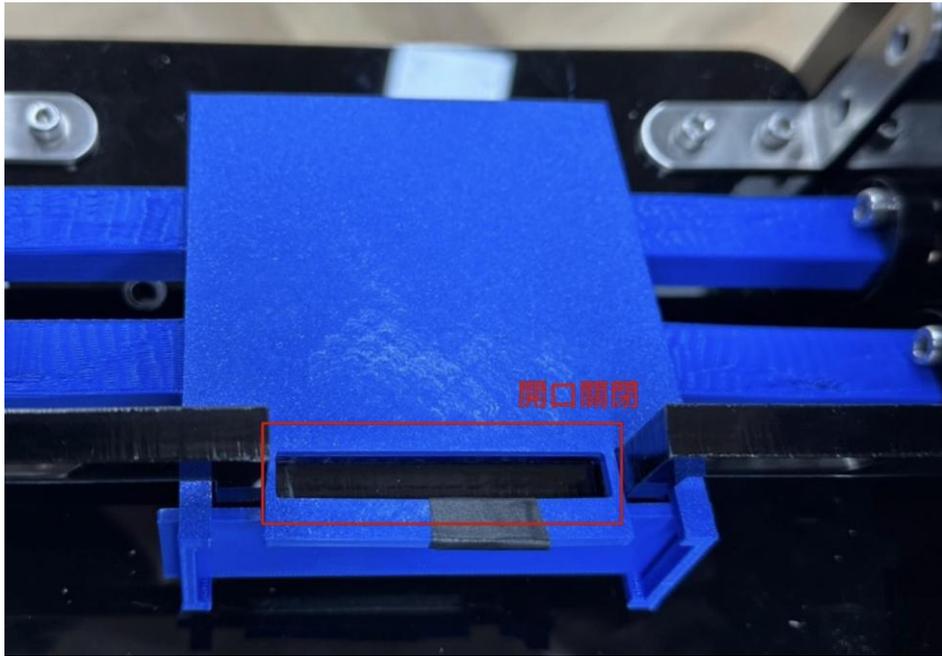


圖 2-21 水草盒開口關閉的狀態

資料來源(本計畫拍攝)

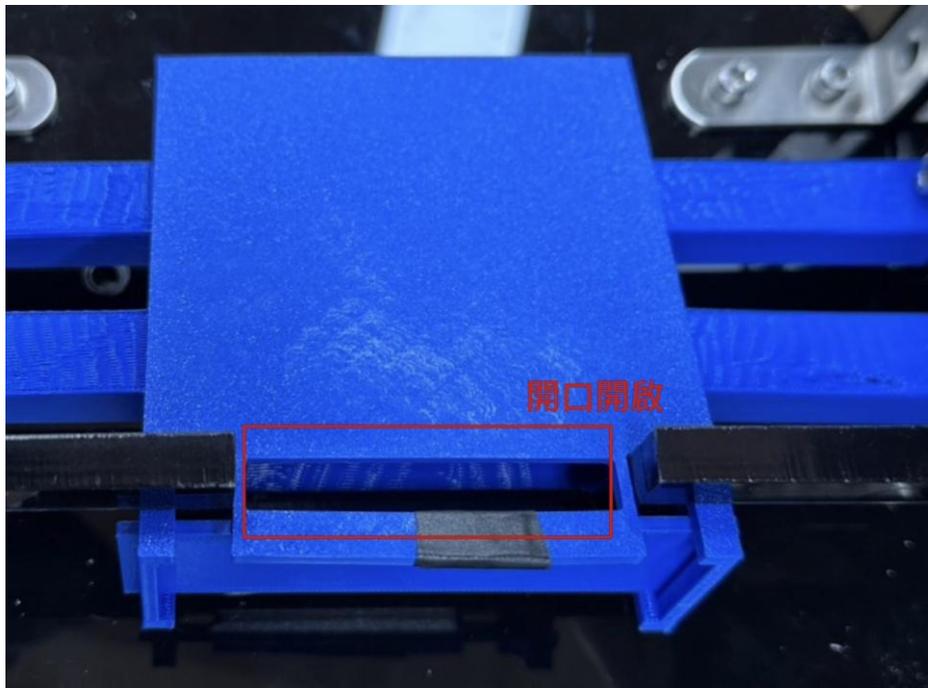


圖 2-22 水草盒開口開啟的狀態

資料來源(本計畫拍攝)

### 三、覆蓋沈積物

在ROV後方加裝兩片3D列印的擋板，將兩片擋板以45度的夾角安裝在ROV的後方，當ROV前進時兩塊擋板可以將周圍的沈積物聚集中到中央完成覆蓋的動作。

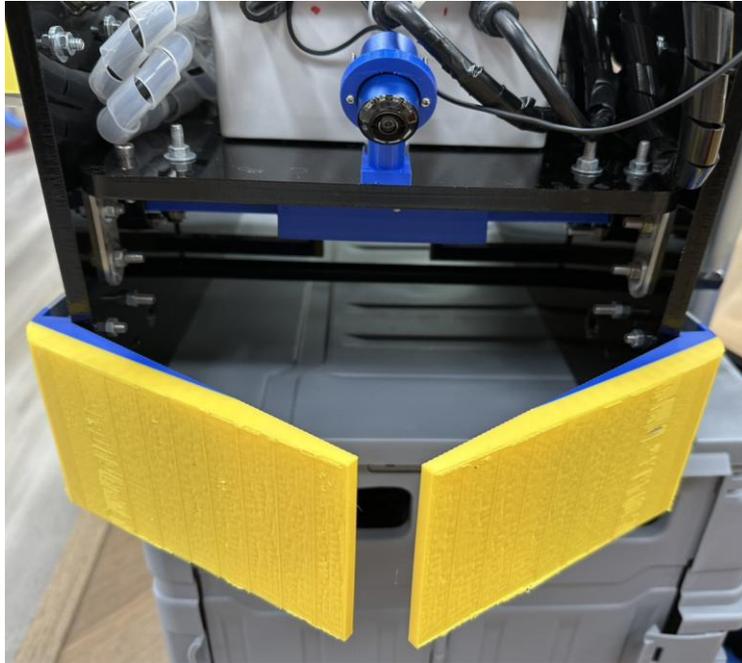


圖 2- 23 正面的實體擋板

資料來源(本計畫拍攝)

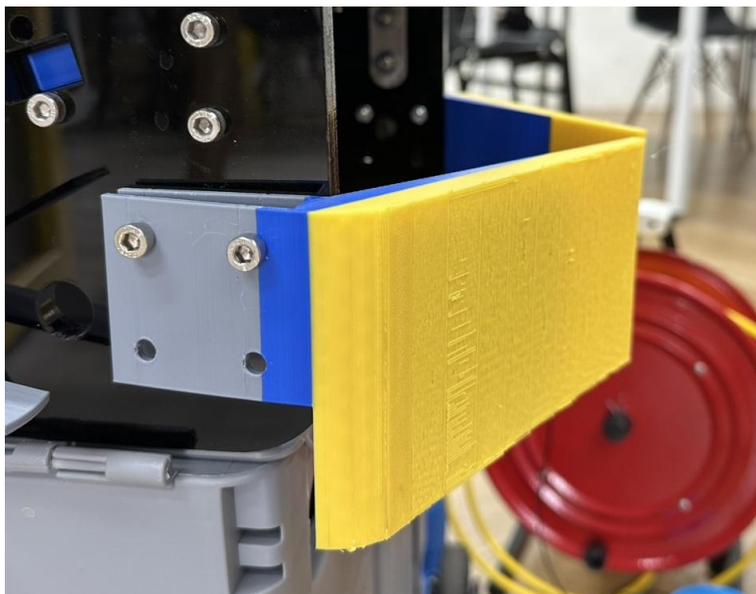


圖 2- 24 側面的實體擋板

資料來源(本計畫拍攝)

### 第三節 電路與程式設計

#### 一、硬體設備的连接

在陸上控制部分，首先將搖桿連接至筆記型電腦，再透過長30米的零浮力電纜，結合RS485通訊模組，將控制訊號穩定地傳輸至水下的Arduino Uno主控板。主控板接收到來自搖桿的訊號後，會將其轉換為 PWM控制訊號，再進一步輸出至電子變速集成板，以控制水下推進器的運作。

每組電子變速集成板可接上三個電子變速器(Electronic Speed Controller ; ESC)，每個 ESC對應一顆水下推進器。本計畫總共使用兩組電子變速集成板共六顆水下推進器，水下推進器分別的配置為：兩顆負責提供向下(垂直)推進的動力、兩顆負責向前推進、另兩顆則用於左右轉向，構成完整的三維移動控制系統。

電力供應方面，每組電子變速集成板皆接上一組提供14.8V的鋰電池，作為推進器的主要動力來源；同時，主控板則由一組提供7.4V的鋰電池供電，以確保其穩定運作。至於水下攝影機，則採用獨立設計，直接與筆記型電腦連接，以實現即時影像的回傳與監控功能。

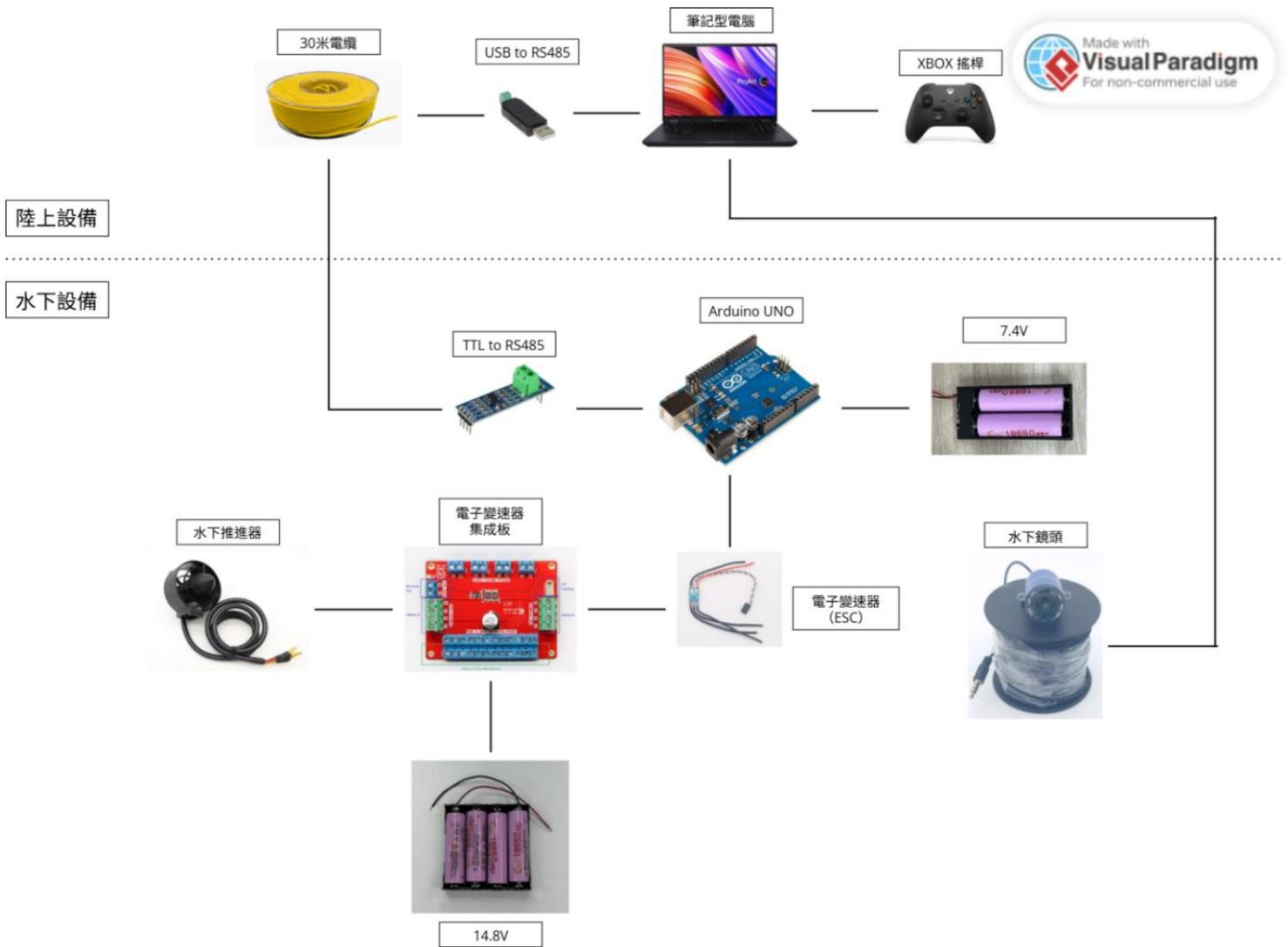


圖 2-25 硬體設備連接圖

資料來源(本計畫繪製)

## 二、軟體的程式設計

為了將搖桿的訊號傳輸至水下的推進器，系統需結合 Python 與 Arduino 軟體的應用。首先，由 Python 負責接收搖桿的控制訊號，並透過 RS485 通訊模組將這些訊號傳送至水下的 Arduino 主控板。接著，Arduino 將收到的訊號轉換為 PWM 控制訊號，並輸出至電子變速集成板，以驅動對應的水下推進器。以下將分為兩個部分：(一) Python 程式設計

、(二) Arduino 程式設計，來詳細說明程式設計的原理。

## (一) Python 程式設計

- 先引入需要使用的模組

```
import pygame # 用來接收搖桿控制訊號
import time   # 控制時間延遲
import serial  # 處理串列通訊 (RS485)
import crc8   # 計算 CRC8 校驗碼
```

- 設定 RS485 在筆電中的連接埠為 COM5，鮑率為 9600。
- 等待 2 秒，讓 Arduino 完成初始化。

```
# RS485 連線到 Arduino
ser = serial.Serial('COM5', 9600)
time.sleep(2)
```

- 初始化 Pygame 模組和搖桿模組。

```
# 初始化 XBOX 搖桿
pygame.init()
pygame.joystick.init()
joystick = pygame.joystick.Joystick(0)
joystick.init()
```

- 設定 deadzone 處理搖桿在中立區時的微小飄移，當值的絕對值小於 0.05 時，視為 0，避免馬達抖動。
- 因為 Arduino 端的 PWM 通常是以整數表示，因此需要將 -1.0 ~ +1.0 的 joystick 範圍轉換成 0 ~ 255。

```
# 設定 deadzone 防止微小的訊號造成馬達抖動
def apply_deadzone(val, threshold=0.05):
    return 0 if abs(val) < threshold else val

def map_axis(val):
    return int((val + 1) * 127.5)
```

- 讀取搖桿的數值，需依照自己的搖桿輸入編號。
- 0: 左搖桿水平 (LX)
- 1: 左搖桿垂直 (LY)
- 2: 右搖桿垂直 (RY)
- 3: 右搖桿水平 (RX)

```
# 呼叫 pygame.event.pump() 來更新 joystick 的狀態
while True:
    pygame.event.pump()
```

```
#讀取搖桿的值並將數值轉換為 0-255
lx = map_axis(apply_deadzone(joystick.get_axis(0)))
ly = map_axis(apply_deadzone(joystick.get_axis(1)))
rx = map_axis(apply_deadzone(joystick.get_axis(3)))
ry = map_axis(apply_deadzone(joystick.get_axis(2)))
```

- 封包形式為："127,127,127,127"，總共四個 0~255 整數，分別對應 lx、ly、rx、ry。
- 使用 CRC8 驗證傳送的資料是否有誤。

```
# 封裝資料
payload = f"{lx},{ly},{rx},{ry}"

# 計算 CRC
hash = crc8.crc8()
hash.update(payload.encode())
crc_value = hash.hexdigest()
```

- 封裝成完整封包格式，E 表示ESC是起始字元；\* 用來分隔資料與 CRC；\n 是終止符號。
- ser.write() 寫入 RS485 傳輸。

```
# 加入起始符號與終止符號，封裝為完整資料封包
data = f"E{payload}*{crc_value}\n"
print("送出:", data.strip())
ser.write(data.encode())
```

- 讀取 Arduino 的回應，若串列埠有收到資料(Arduino 回傳的資料)，則讀出並印出。
- 目的是用來確認資料是否有正確到達。

```
# 讀取 Arduino 回傳的資料
if ser.in_waiting:
    response = ser.readline().decode().strip()
    print("回應:", response)
```

- 設定每次資料發送後等待 0.3 秒，降低傳輸頻率、避免 Arduino 資料爆炸或處理不及。

```
#傳送資料的頻率
time.sleep(0.3)
```

## (二) Arduino 程式設計

- 引入需要的函式庫，並定義 RS485 的接腳。

```
//引入函式庫與定義
#include <Servo.h>
#include <SoftwareSerial.h>

#define RS485_RX 2
#define RS485_TX 3
#define RS485_DIR 4
```

- 存放 RS485 接收資料的緩衝區，定義資料的大小為 64 bit。
- 創建一個模擬串口(SoftwareSerial)物件，名稱為 rs485Serial，用於和 RS485 模組通訊。

```
//存放 RS485 接收資料的緩衝區
const size_t rcdDataFromRS485_size = 64;
static char rcdDataFromRS485[rcdDataFromRS485_size];

//接收 RS485 的資料
SoftwareSerial rs485Serial(RS485_RX, RS485_TX);
```

- 宣告四個 ESC 對應的 Servo 物件，分別對應 LX、LY、RX、RY。
- 依照水下推進器的 PWM 工作區間，將所有水下推進器的中立設為 1480。

```
//初始化 Servo
Servo escLY, escLX, escRY, escRX;
int lastPWM_LY = -1, lastPWM_LX = -1, lastPWM_RY = -1, lastPWM_RX = -1;

void stopAllMotors() {
    escLY.writeMicroseconds(1480);
    escLX.writeMicroseconds(1480);
    escRY.writeMicroseconds(1480);
    escRX.writeMicroseconds(1480);

    lastPWM_LY = 1480;
    lastPWM_LX = 1480;
    lastPWM_RY = 1480;
    lastPWM_RX = 1480;
}
```

- 將收到的資料轉為四個整數，分別送給對應的水下推進器。
- 若解析失敗(格式錯誤)，則會停止所有水下推進器的運作。

```
//指令解析與執行
void dispatch_cmd(String input) {
    int lx, ly, rx, ry;
    if (sscanf(input.c_str(), "%d,%d,%d,%d", &lx, &ly, &rx, &ry) == 4) {
```

```

    updateESC(escLY, ly, lastPWM_LY);
    updateESC(escLX, lx, lastPWM_LX);
    updateESC(escRY, ry, lastPWM_RY);
    updateESC(escRX, rx, lastPWM_RX);
  } else {
    Serial.println("解析指令失敗");
    stopAllMotors(); // 安全處理
  }
}

```

- 使用 CRC-8 計算函式。

```

//CRC-8 檢查
byte crc8(const char* data, int len) {
  byte crc = 0x00;
  while (len--) {
    crc ^= *data++;
    for (byte i = 0; i < 8; ++i)
      crc = (crc & 0x80) ? (crc << 1) ^ 0x07 : (crc << 1);
  }
  return crc;
}

```

- 將搖桿值 0-255 映射成 ESC 的 PWM 值 1000-2000，因此當區間在 1000-1470 時為反轉，1490-2000 時為正轉。
- 死區(Deadzone)是推進器停止的區間，為搖桿中心值 127 正負20 的區間，也就是 107-147 之間，將這個區間的值都設為PWM值的 1480。
- 若 PWM 值有實際變化時才執行寫入 ESC，避免無意義的訊號改變造成抖動。

```

//ESC 控制核心
int mapEscSmooth(int val) {
  const int center = 127, deadZone = 20;
  if (val < center - deadZone) return map(val, 0, center - deadZone, 1000, 1470);
  else if (val > center + deadZone) return map(val, center + deadZone, 255, 1490, 2000);
  else return map(val, center - deadZone, center + deadZone, 1470, 1490);
}

void updateESC(Servo& esc, int inputVal, int& lastPWM) {
  int pwm = mapEscSmooth(inputVal);
  if (abs(pwm - lastPWM) > 3) {
    esc.writeMicroseconds(pwm);
    lastPWM = pwm;
  }
}

```

- 先給定常數值 `CRC_HEX_LEN = 2` 是 CRC 兩個十六進位字元，例如 `d0`，`CRC_STR_BUF_SIZE = 3` 是包含 `\0` 結尾符號，因此字串陣列大小為 3。
- `Message` 為完整接收的字串(格式：`E127,127,127,127*d0`)。
- `OutputPayload` 的作用為如果驗證成功，會將原本接收到的字串設定成 `127、127、127、127`。
- 若驗證失敗都會停止推進器的運作。

```
//驗證接收到的資料是否符合 CRC-8 校驗
constexpr int CRC_HEX_LEN = 2;
constexpr int CRC_STR_BUF_SIZE = CRC_HEX_LEN + 1;

bool verifyCRC(const char* message, String& outputPayload) {
    if (message[0] != 'E') return false; // 檢查字串開頭是否為 E

    char* starPos = strchr(message, '*'); //找到 * 的位置
    if (!starPos) return false;

    char receivedCRC[CRC_STR_BUF_SIZE]; //取出 CRC 字串
    strncpy(receivedCRC, starPos + 1, CRC_HEX_LEN);
    receivedCRC[CRC_HEX_LEN] = '\0';

    *starPos = '\0'; //截斷原始的字串，把 * 當成結束點

    String payload = String(message + 1); //把資料部分轉成 String
    byte calc = crc8(payload.c_str(), payload.length()); //計算 CRC 並轉成十六進位字串

    char calcHex[CRC_STR_BUF_SIZE];
    sprintf(calcHex, "%02x", calc);

    //比對原始 CRC 與計算結果
    if (strcmp(receivedCRC, calcHex) == 0) {
        outputPayload = payload;
        return true;
    }
    return false;
}
```

- 初始化 USB 序列與 RS485 模擬序列，並設定 RS485 為接收模式。
- 定義 ESC 在 ARDUINO 板上的接腳。

```
//初始化 USB 序列與 RS485 模擬序列，ESC 腳位的設定
void setup() {
    Serial.begin(9600);
```

```

rs485Serial.begin(9600);
pinMode(RS485_DIR, OUTPUT);
digitalWrite(RS485_DIR, LOW);

escLY.attach(5);
escLX.attach(6);
escRY.attach(9);
escRX.attach(10);

stopAllMotors();
delay(2000);
Serial.println("準備接收 RS485 ESC 控制資料");
}

```

- 設定主邏輯迴圈。
- 若 RS485 沒收到資料就跳過。
- 讀取收到的一整行封包直到讀到 \n(以 \n 結尾)，並補上 \0 結尾符。
- 驗證封包內的資料是否正確。
- 若錯誤，將停止所有推進器的運作，並回傳 CRC 錯誤。

```

//主邏輯迴圈
void loop() {
  digitalWrite(RS485_DIR, LOW);

  if (!rs485Serial.available()) return;

  size_t len = rs485Serial.readBytesUntil('\n', rcdDataFromRS485,
rcdDataFromRS485_size - 1);
  rcdDataFromRS485[len] = '\0';

  Serial.print("收到: ");
  Serial.println(rcdDataFromRS485);

  String payload;
  if (!verifyCRC(rcdDataFromRS485, payload)) {
    Serial.println("CRC 錯誤或格式錯誤!");
    stopAllMotors();
    return;
  }
}

```

- 若正確，則將原始資料回傳給 PYTHON，並傳送 CRC OK。

```

//資料通過 CRC 後才回傳給 PYTHON
digitalWrite(RS485_DIR, HIGH);
rs485Serial.println(rcdDataFromRS485); // 原樣回傳(回給 PC)
delay(1);

```

```
digitalWrite(RS485_DIR, LOW);  
  
Serial.print("回傳: ");  
Serial.println(rcdataFromRS485);  
  
dispatch_cmd(payload);  
Serial.println("CRC OK.");  
}
```

#### 第四節 防水與整合測試

##### 一、防水測試

在水下遙控載具 (Remotely Operated Vehicle, ROV) 的製作過程中，水密性設計是影響系統可靠度的關鍵環節。為確保驅動板、電池等需供電之核心元件的正常運作，所有相關設備均必須安置於防水盒內。若防水措施不完全，將可能導致整體系統失效。因此，本研究首先於防水盒上設置七個孔洞，其中六個孔用以連接推進器，另有一孔則用於穿出長度三十公尺之零浮力纜線以與電腦進行通訊。孔洞完成後安裝專用防水接頭，隨後將整個防水盒置於水中，並進行約40至60分鐘的浸水測試(圖2-14)，以檢驗其密封性能並確認內部是否存在滲水現象，如果防水盒沒有滲水情況，表示成功通過第一階段的水密性測試。



圖 2- 26 防水盒水密測試

資料來源(本計畫拍攝)

在完成第一階段的水密性測試並確認防水盒未發生滲水後，即進入第二階段的驗證如影片C所示(本計畫拍攝)。此階段將所有電路元件安裝於防水盒內，並連接單一推進器作為功能測試樣本。為進行持續運作試驗，本研究以Arduino撰寫簡易程式，使推進器於迴圈中持續接收最大PWM訊號輸出。隨後，將完整安裝之防水盒置入水中，觀察推進器是否能穩定運作。測試持續約30分鐘，結束後將防水盒取出並開啟檢查內部狀況。若確認推進器能正常運轉且盒內未出現滲水，即代表第二階段測試順利通過。在完成前述水密性測試後，本研究將防水盒與ROV本體進行組裝(圖2-15)，並開始浮力測試，以評估整體系統於水中之穩定性與適浮性。

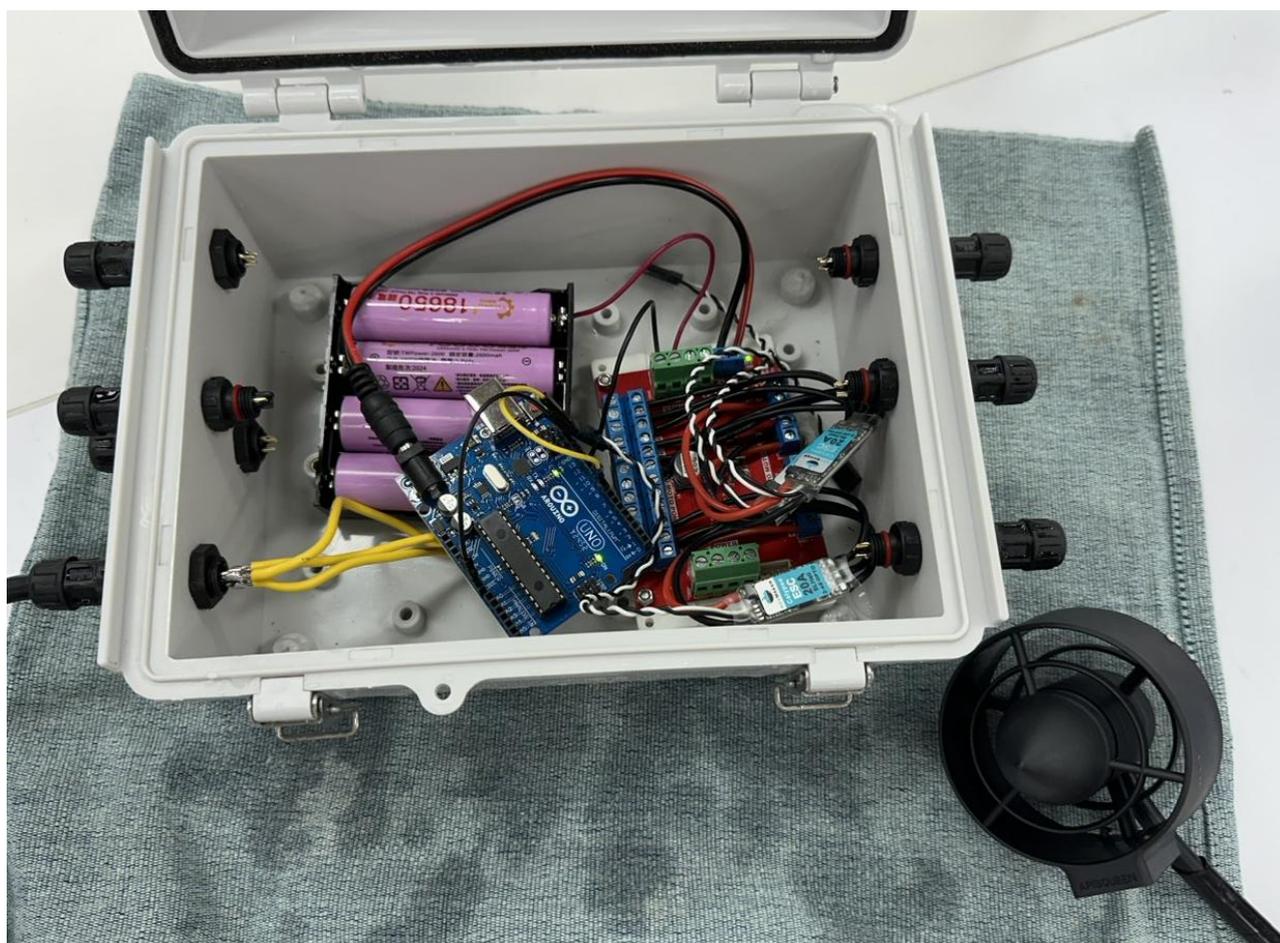


圖 2-27 將所有電路元件安裝於防水盒內進行測試

資料來源(本計畫拍攝)

## 二、浮力測試

在浮力測試階段，本研究將已完成組裝之ROV置入小型游泳池中進行實驗。在泳池測試中可觀察到，ROV較重的部位傾向下沉，而較輕的部分則因浮力較大而上浮(圖2-16)。為使ROV能在水中維持水平姿態，必須適當調整浮塊配置以達到浮力平衡(圖)。此外，所有推進器亦需完全浸沒於水中，以確保推進效率並避免因推進器未能完全「吃水」而導致推進力不足從而提升ROV操控時的穩定性與可操作性。



圖 2-28 防水盒與 ROV 本體組裝

資料來源(本計畫拍攝)

### 三、整體測試

在完成浮力配置調整後，進一步進行整體ROV之控制測試如影片D所示(本計畫拍攝)。測試過程中，將訊號線連接至筆記型電腦，並透過搖桿作為操控介面，以執行前進、後退、左轉、右轉、下潛及上升等動作。藉由觀察ROV於水中的反應與姿態，評估其操控靈敏度與運動性能，以確認系統能否達成靈活且穩定的操作需求。



圖 2- 29 ROV 準備進行測試

資料來源(本計畫拍攝)

## 第三章 結果與討論

### 第一節 結果

本計畫之研究目的在於設計可利用釘固法進行海草移植的機械結構，並開發一款小型水下無人載具（ROV）模型。經多次測試後，結果顯示該ROV能透過搖桿操作完成前進、後退、左轉、右轉、下潛與上升等基本動作如影片D所示(本計畫拍攝)，顯示其具備良好的操控靈活性。

在海草移植模組方面，雖未能於實際環境中進行海草移植測試，但泳池實驗結果證實四連桿機構與凸輪機構均可正常運作。然而，測試亦發現該模組於驅動頻率上存在差異：四連桿機構於每一循環僅能挖掘一次沉積物，而凸輪機構卻於同一循環內驅動水草盒開啟兩次，導致挖掘與釋放動作無法完全同步。此問題主要源自齒輪比設計不一致，造成模組運作不同步，進而可能影響實際移植作業之穩定性。

與既有的機械化移植研究相比，本研究所提出之ROV具有體積小巧與操控性佳的優勢，能提升在受限環境中進行復育作業的可行性。然而，目前的海草移植模組在操作同步性與移植穩定性方面仍有待改進，未來可透過齒輪比優化或同步化設計加以強化。

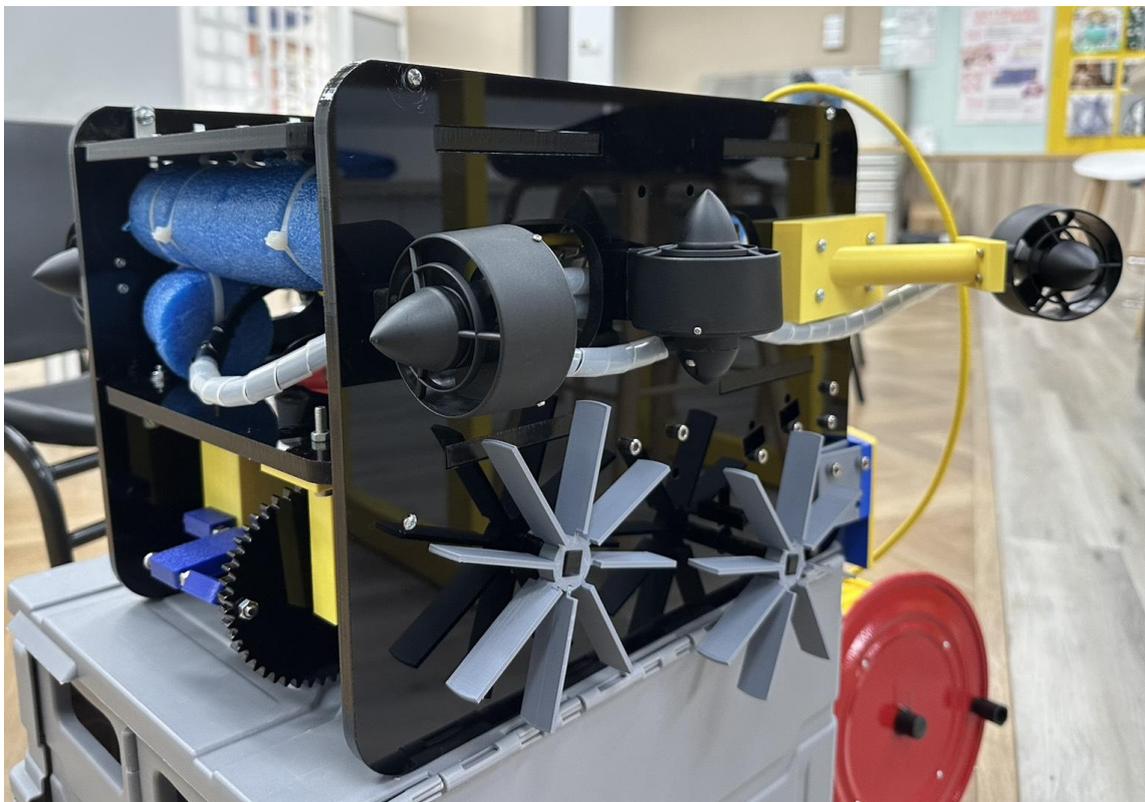


圖 3-1 最終成果

資料來源(本計畫拍攝)

## 第二節 討論

在小型的海草移植水下無人載具模型的開發過程中，所遭遇之問題討論如下：

### 一、海草移植模組頻率差異

在設計四連桿機構與凸輪機構時，原規劃均採用水車輪帶動24齒小齒輪再帶動48齒大齒輪進行驅動(圖3-1)。然而，在實際組裝過程中發現，若於凸輪機構中安裝大齒輪，將產生結構干涉問題。由於側板已經完成雷射切割，無法再進行修正，因此最終僅能將齒輪皆拆除，改以水車輪直接帶動凸輪(圖3-2)。此設計更動導致四連桿與凸輪機構之間出現驅動頻率差異，進而造成移植動作步驟未能完全同步。未來可透過調整四連桿機構的驅動桿設計，使其亦能以兩顆24齒的小齒輪作為驅動來源，以避免不同步的問題並提升模組運作的穩定性。

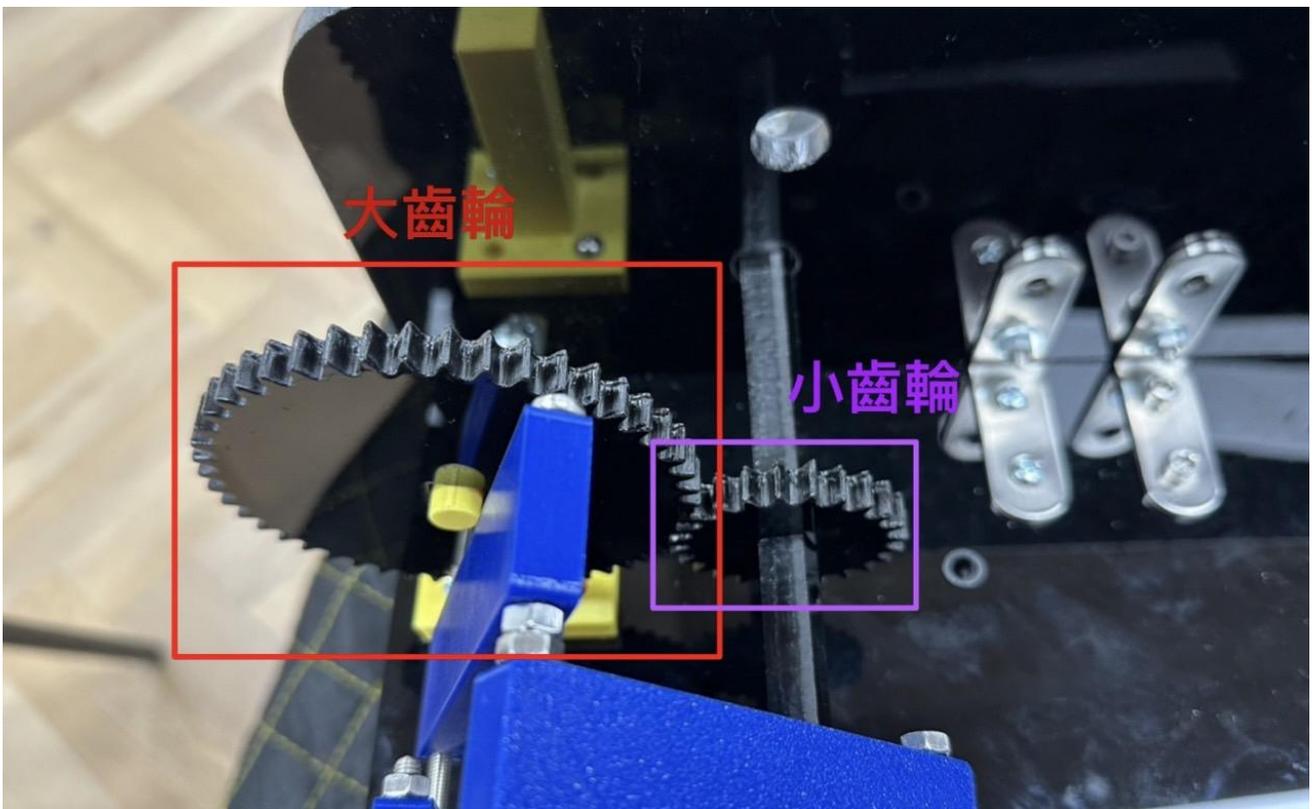


圖 3-2 四連桿機構中大齒輪帶動小齒輪

資料來源(本計畫拍攝)

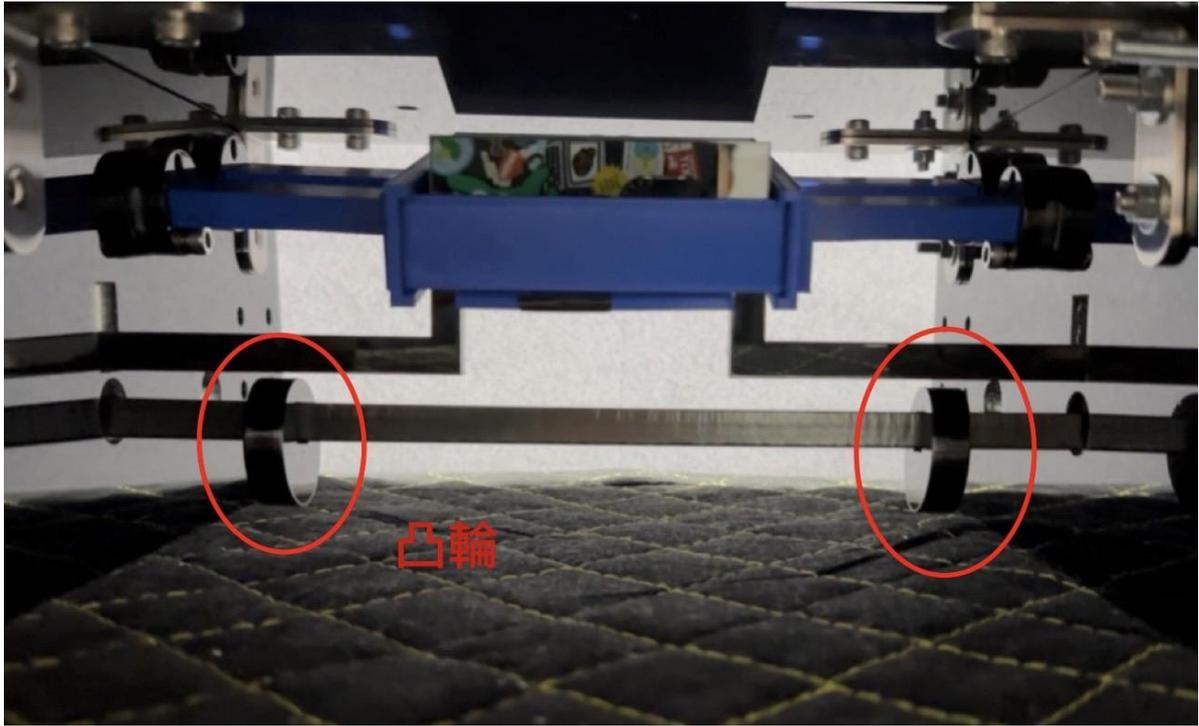


圖 3-3 凸輪機構直接由凸輪驅動

資料來源(本計畫拍攝)

## 二、水下推進器抖動問題

在水下推進器測試過程中發現，當搖桿緩慢推至PWM最大值或最小值時，推進器會產生抖動現象如影片F所示(本計畫拍攝)。然而，若直接將搖桿推至PWM的極值，抖動現象則較不明顯。經資料查驗後分析，推進器抖動的原因主要在於PWM訊號未落在ESC的有效工作區間。換言之，Arduino所設定的PWM範圍不符合ESC的工作需求，導致推進器無法穩定運作。

透過實驗測試，本計劃所使用的ESC工作區間為PWM值800–1470及1490–2000，而PWM值1470–1490為死區，推進器在此範圍內無法產生動力。根據此結果，重新調整Arduino內的PWM設定，使其符合ESC的工作區間，即可有效消除抖動現象如影片G所示(本計畫拍攝)，提升搖桿操控推進器時的穩定性。

### 三、防水盒滲水的現象

初步測試結果如影片E顯示(本計畫拍攝)，部分防水接頭在浸水過程中持續產生氣泡，顯示其水密性不足，推測防水接頭與防水盒之間存在縫隙。經多次重複試驗與觀察後發現，問題源自於初期防水盒孔徑設計不當。由於孔徑過小，防水接頭在安裝過程中被強行嵌入，導致接頭螺旋鎖緊部位受壓損壞，無法完全密合，進而造成滲水現象。為解決此問題，研究團隊將孔徑適度放大，並更換全新防水接頭，隨後進行再次測試。結果顯示，防水盒已無滲水情況，成功通過水密測試。

### 四、浮塊位置的設置

在未加裝任何浮塊的情況下，ROV因前端重量較大而呈現傾斜姿態(圖3-3)。為改善此問題，首先嘗試於ROV前端上方加裝浮塊(圖3-4)。然而，經多次測試後發現，由於浮塊位置過高，對於增加前端浮力的效果有限，無法有效改善傾斜現象，因此本計畫調整浮塊安置位置，將其移至ROV第二層(圖3-5)，結果顯示前端浮力確有明顯提升，但仍存在一定程度的傾斜。為進一步修正，於ROV前後及左右方向進行浮塊微調(圖3-6)，最終使整體裝置能夠維持與水面平行的狀態，並且使水下推進器都可以保持在水面之下(圖3-7)。

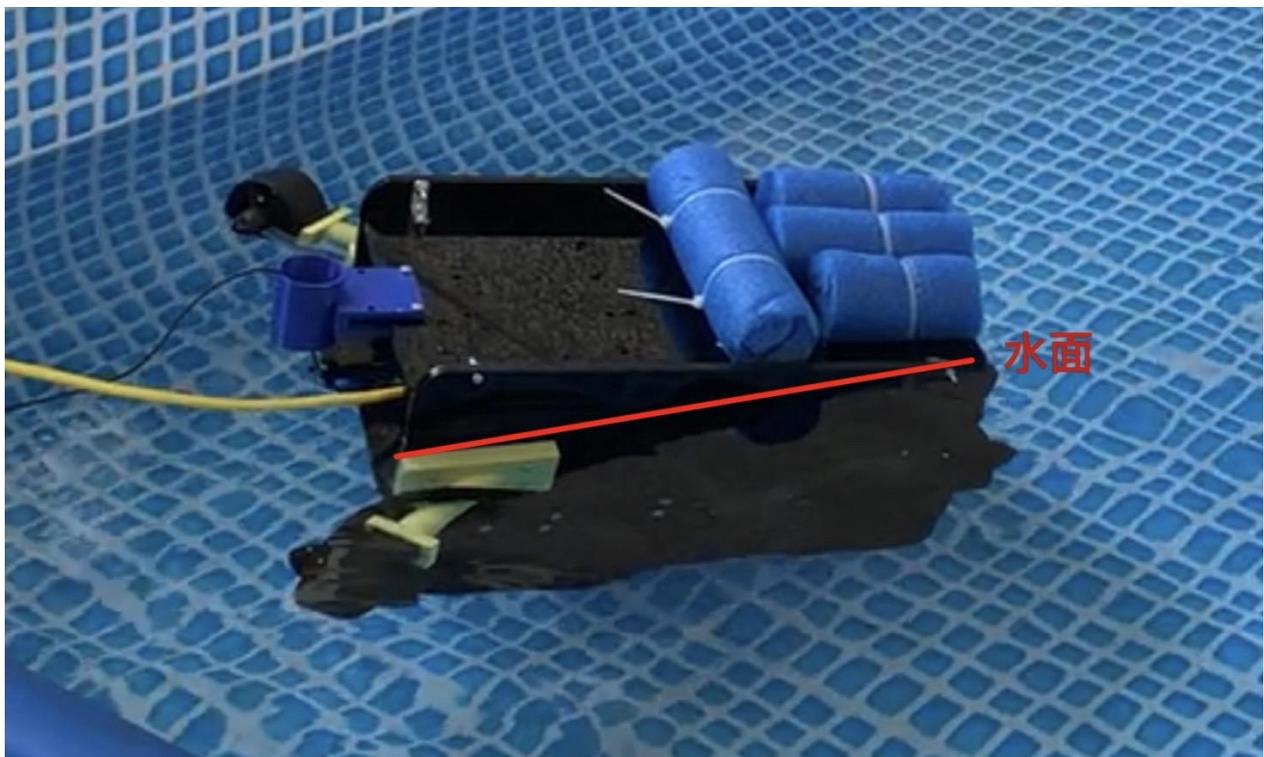


圖 3-4 嘗試於 ROV 前端上方加裝浮塊

資料來源(本計畫拍攝)

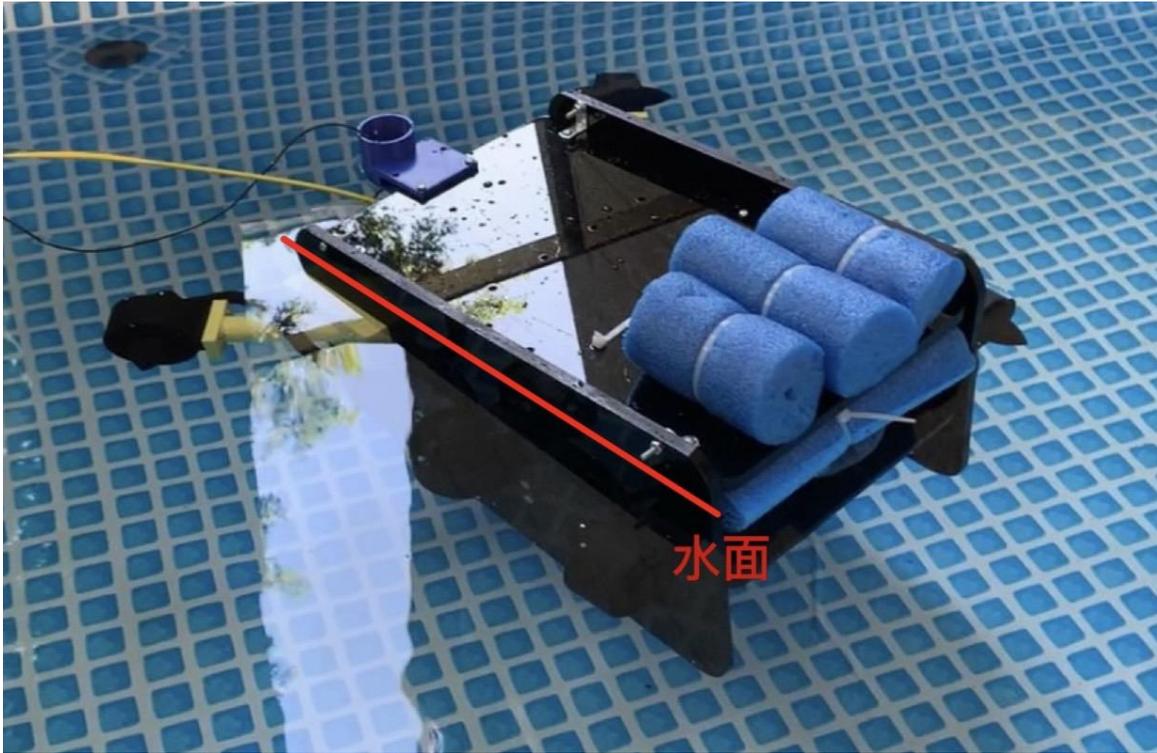


圖 3-5 調整浮塊將其移至 ROV 第二層

資料來源(本計畫拍攝)



圖 3-6 進行浮塊微調

資料來源(本計畫拍攝)

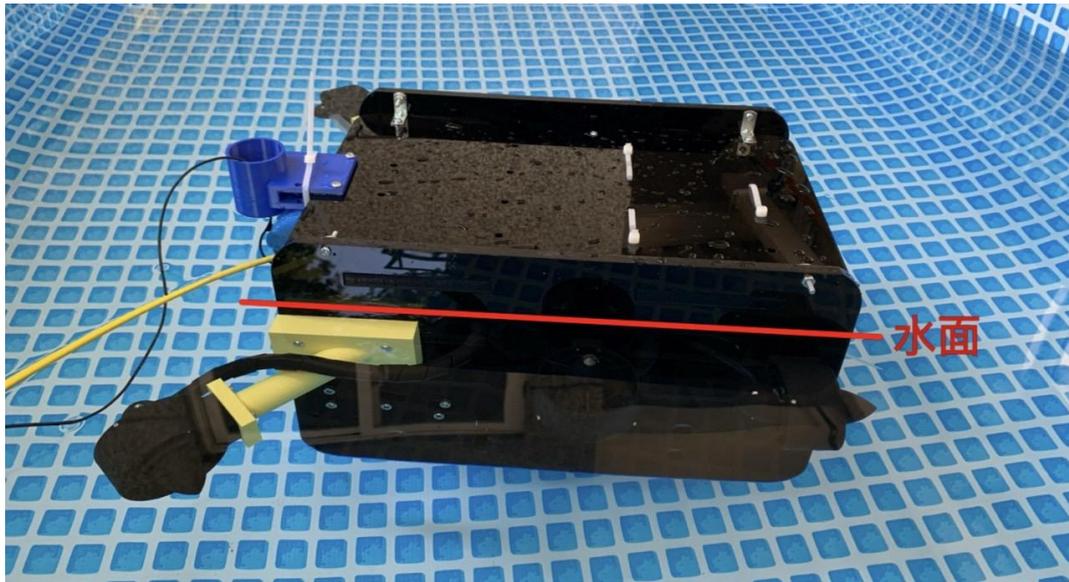


圖 3-7 使 ROV 能夠維持與水面平行的狀態

資料來源(本計畫拍攝)

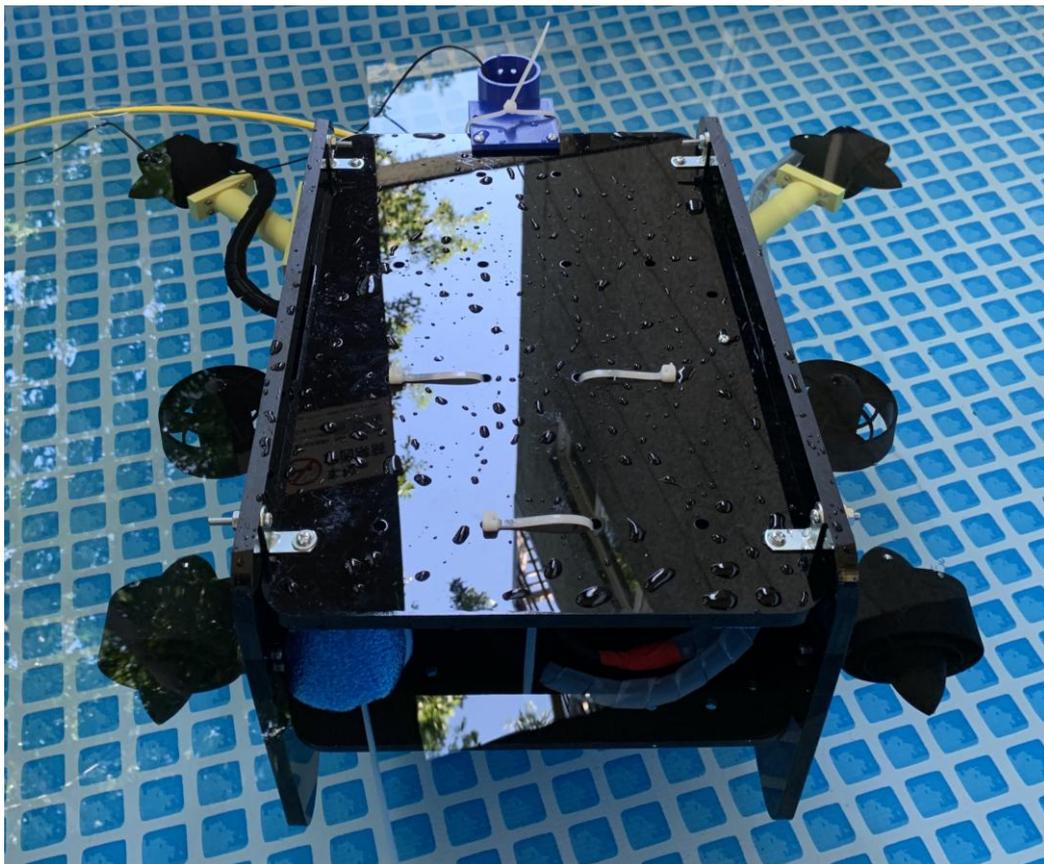


圖 3-8 推進器均能完全浸沒於水中

資料來源(本計畫拍攝)

## 五、水下鏡頭的設置

目前 ROV 僅於後方配置一組水下鏡頭。然而在實際操作過程中發現，若前方未設置鏡頭，操控者將難以掌握 ROV 的運動狀態，必須在能直接觀測 ROV 的範圍內才能進行操控。為提升操控靈活度與安全性，未來若需更精確地控制 ROV，或在下潛至更深水域時，建議於前端增設一組水下鏡頭，以提供更完整的視覺輔助。

## 第四章 結論

### 第一節 結論

本研究的目的是在於設計一套能以釘固法進行海草移植的模組，並開發小型水下無人載具模型以搭載該模組。研究方法涵蓋水下無人載具的結構設計、海草移植模組的設計、電路與程式的開發，以及防水與系統整合測試。

研究成果結論如下：

- 所開發之載具可透過搖桿進行操控，具備前進、後退、上升、下潛、左轉與右轉等運動控制功能，並能即時回傳後方影像。
- 移植模組採用水車輪驅動的四連桿與凸輪機構，在水中能順利完成機械結構的運作。
- 載具本體以壓克力板製作，並可與 3D 列印零件結合使用，以提升其模組化與擴充性。

### 第二節 未來展望

本載具已能實現基本的操控功能，但由於缺乏前方視覺輔助，在操作上仍存在一定不便。目前推進器係以 18650 鋰電池作為電力來源，若水密措施未能確實執行，則可能帶來潛在風險。未來可考慮於前方增設水下鏡頭，並更換更為安全的供電方式，以提升 ROV 的操作穩定性與安全性。在海草移植模組方面，則可進一步調整四連桿機構與凸輪機構，使其運作頻率一致，並進行實際海草移植測試，以驗證其效能。透過上述改良，期望能使整體載具操作更趨完善，並為藍碳固存的目標貢獻一份力量。

## 參考文獻

- Chimienti, G., A. Tursi, L. Saponari, V. Lovat, F. Santoro, and F. Mastrototaro. 2023. “A New Method for the Transplanting of the Seagrass *Posidonia oceanica*.” In *2023 IEEE International Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea)*, 83–87. La Valletta, Malta. <https://doi.org/10.1109/MetroSea58055.2023.10317405>.
- Descamp, P., S. Personnic, S. Gobert, A. Boulenger, M. Leduc, G. Delaruelle, A. Barroil, G. Marre, F. Holon, and J. Deter. 2025. “Seagrass Sod Transplantation: A Relevant Tool for Preventing the Destruction of Meadows in Coastal Construction Projects.” *Environmental Challenges* 18 (April): 101087. <https://doi.org/10.1016/j.envc.2025.101087>.
- Lawrence C. H., & Cooke C. T. (2006). Autopsy and the investigation of scuba diving fatalities. *Diving and Hyperbaric Medicine*, 36(1), 2–8.
- Marre, G., F. Holon, S. Luque, et al. 2019. “Fine-Scale Automatic Mapping of Living *Posidonia oceanica* Seagrass Beds with Underwater Photogrammetry.” *Marine Ecology Progress Series* 643 (June): 63–77. <https://doi.org/10.3354/meps13338>.
- Montefalcone, M. 2009. “Ecosystem Health Assessment Using the Mediterranean Seagrass *Posidonia oceanica*: A Review.” *Ecological Indicators* 9 (4): 595–604. <https://doi.org/10.1016/j.ecolind.2008.09.013>.
- Orth, Robert J., Tim J. B. Carruthers, William C. Dennison, Carlos M. Duarte, James W. Fourqurean, Kenneth L. Heck, A. Randall Hughes, Gary A. Kendrick, W. Judson Kenworthy, Suzanne Olyarnik, Frederick T. Short, Michelle Waycott, and Susan L. Williams. 2006. “A Global Crisis for Seagrass Ecosystems.” *BioScience* 56 (12): 987–996. [https://doi.org/10.1641/0006-3568\(2006\)56\[987:AGCFSE\]2.0.CO;2](https://doi.org/10.1641/0006-3568(2006)56[987:AGCFSE]2.0.CO;2).
- Paling, E. I., M. van Keulen, K. Wheeler, J. Phillips, and R. Dyhrberg. 2001a. “Mechanical Seagrass Transplantation in Western Australia.” *Ecological Engineering* 16: 331–339. [https://doi.org/10.1016/S0925-8574\(00\)00119-1](https://doi.org/10.1016/S0925-8574(00)00119-1).
- Paling, E. I., M. van Keulen, K. D. Wheeler, J. Phillips, R. Dyhrberg, and D. A. Lord. 2001b. “Improving Mechanical Seagrass Transplantation.” *Ecological Engineering* 18: 107–113. [https://doi.org/10.1016/S0925-8574\(01\)00065-9](https://doi.org/10.1016/S0925-8574(01)00065-9).

黃國益、林聖泉（1995）。〈插秧機橢圓齒輪式插植機構之運動分析〉。《農業機械學刊》，4(4)，35 - 45。 <https://doi.org/10.30062/JAM.199512.0004>