

探討大眾對於寵物餵食器之看法與自製寵物餵食器

新北市 112 學年度埔墘國民小學一般智能資優生獨立研究
投稿類別：設計與創作類

篇名：

不在其「餵」，不「食」其「器」

作者：

徐振維。新北市埔墘國民小學。六年 11 班。

指導老師：

林伯軒老師

中華民國 112 年 12 月

摘要

寵物可能會因為食用腐壞的食物而生病，因此本實驗採用真空的方式存放食物並自製具有解真空功能且能自動定時餵食的寵物餵食器。首先，整理並分析市面上販售的寵物餵食器之功能，接著製作功能重視程度的問卷，瞭解大眾對寵物餵食器的認知及需求，再根據個人需求，篩選較為重要的餵食器功能作為製作的目標。為了製作符合標準的餵食器，我最後使用樂高積木設計出類似機械手臂的機構，並搭配 EV3 主機、EV3 相關配件及市售的相關模組，製作出寵物餵食器的原型。

壹、前言

一、研究動機

養寵物對於現代人來說是一件常見的事--我的家中剛好也有飼養一隻貓咪，有時我們會到外國旅遊而長時間不在家無法照顧寵物，若使用一般的定時餵食器，存放在內的食物可能因潮濕的環境使食物滋生細菌與腐壞，而導致寵物生病！因此，我想設計一種具"真空"功能的寵物餵食器來解決這個問題發生。

二、研究目的

- (一) 探討及分析市面上販售的寵物餵食器。
- (二) 透過問卷瞭解市場及大眾對於市售寵物餵食器的認知及需求。
- (三) 製作符合大眾需求的寵物餵食器。

貳、文獻探討


1. 寵物餵食器的功能

參考以下功能及各大有關寵物餵食器的網站，我挑選四種市售常見的寵物餵食器，並結合自身使用寵物餵食器的經驗，再整理出幾種不同的產品功能(請見表1)。

表1 市售餵食器寵物餵食器比較表

產品名稱	優點	缺點	分析及討論
【PETWANT】 甜甜圈分格 餵物食器 	<ol style="list-style-type: none"> 六餐分格，乾溼食物都適用。 具備防卡腳設計。 有電池與插電式的雙重供電方式。 	缺少了飼料儲存槽，6格飼料吃完後就得手動補充，對於有時候得長時間外出的飼主來說便利性稍嫌不足。	可達到一般餵食的需求及標準，價格也較能為大眾所接受。
【Meoof】 寵物自動餵食器 	<ol style="list-style-type: none"> 可以雙碗餵食。 可搭配手機應用程式連接連線使用，定時定量出糧追蹤紀錄。 有電池與插電式的雙重供電方式。 	<ol style="list-style-type: none"> 兩邊的出糧量不均。 鐵碗可以拿起來，因此機器出糧時飼料的屑屑都會掉到碗下面，需要常常清理。 	價格不至於太貴，獨特的雙碗設計，對於有兩隻寵物的家庭是一個好選擇。
【Petkit 佩奇】 智能寵物 餵食器 SOLO 	<ol style="list-style-type: none"> 有防碰撞和防誤起鎖鍵，避免寵物自行打開。 可搭配手機應用程式連接連線使用，定時定量出糧追蹤紀錄。 電源線採用尼龍繩的編織材質。 	機體較高，寵物容易推倒，裡面的飼料可能溢出。	可適應各類型的飼料(如：凍乾、乾糧及風乾糧)並且擁有各種實用的功能，雖價格偏貴，但仍是一個好的選擇。
【小米】 米家小頑智能 寵物餵食器	<ol style="list-style-type: none"> 餵食盤是以磁吸的方式拆裝，方便拆洗。 	米家 App 內可以根據寵物的品種、年紀，以及食糧本身的成份，為寵	一般人會期待的智能功能，例如監察寵物的進食

探討大眾對於寵物餵食器之看法與自製寵物餵食器

	<ol style="list-style-type: none"> 2. 可搭配手機應用程式連接使用多項其他應用功能。 3. 擁有三道的保存程序，密封性相當的不錯。 	<p>物計算出每天建議餵食的份量，但計算出的結果總比平時多，寵物也吃不完。</p>	<p>情況、糧桶內的餘糧份量、聲音誘食等，大部份都未能提供。雖說在此價位未能要求太多，如果可以增加更多功能，將會更完美。</p>
---	---	---	--

我所參考的四款餵食器多體現在食物保存，也符合我的個人猜想，我也將會參考後進行問卷調查及作為設計餵食器時設計調整的依據。

2. 寵物餵食器之 SWOT 分析

我根據以上對餵食器功能的分析製作 SWOT 分析，瞭解寵物餵食器在市場中的競爭程度和評估大眾的需求及購買意願，以便更好的確立製作寵物餵食器的規劃方向，讓我製作出的餵食器能符合大眾的需要(請見表 2)。

表 2 寵物餵食器之 SWOT 分析

優勢	劣勢
<ol style="list-style-type: none"> 1. 自動化餵食：市售餵食器能夠自動餵食寵物，減少使用者的負擔，尤其是當使用者外出無法照顧寵物時。 2. 適應多種寵物：許多市售餵食器可以適應不同種類和大小的寵物，並可調整份量的選，以滿足不同寵物的需求。 	<ol style="list-style-type: none"> 1. 技術故障：一些餵食器可能存在技術故障或機械故障的風險，導致餵食計劃失敗，可能需要進行維修或更換。 2. 高端的市售餵食器通常價格較高，對於一些寵物主人來說可能是一筆額外的支出。
機會	威脅
<ol style="list-style-type: none"> 1. 市場增長：隨著人們對寵物生活品質的重視和忙碌的現代生活方式，市場對於方便的寵物餵食解決方案的需求正在增加。 2. 技術創新：隨著科技的發展，市售餵食器可以不斷進行技術創新，增加更多的功能，能吸引更多消費者。 	<ol style="list-style-type: none"> 1. 競爭激烈：市場上存在著眾多餵食器品牌，競爭激烈，需要不斷提升產品質量和服務水平以保持競爭優勢。

3. 寵物餵食器問卷製作

我參考「陳志柏（2015）應用 ISPM 與 AHP 分析法於寵物產品設計之概念發展。朝陽科技大學工業設計系碩士論文。」中所使用設計問卷的方法。他將已有的寵物餵食器功能列出後，分成四類分別為實際功能、審美功能、標誌功能、象徵功能(如下表 2)。

表 3 「應用 ISPM 與 AHP 分析法於寵物產品設計之概念發展」分類方式

項目	功能	類別
1	產品取得	實際功能
2	運輸安全	
3	使用操作	
4	物品存放	
5	廢棄放置	
6	尺寸造型	審美功能
7	使用材質	
8	表面質感	
9	色彩規劃	
10	整體排列	標誌功能
11	辨識範圍	
12	歸類分組	
13	表面紋路	
14	顏色對比	
15	操作方向	象徵功能
16	時代精神	
17	團體歸屬	
18	感情牽絆	

探討大眾對於寵物餵食器之看法與自製寵物餵食器

而我在參考市面上的寵物餵食器後，按照產品功能、研究目的及陳志柏的論文，將寵物餵食器分類成六項類別，分別為食物存放、安全性及保護、供電方式、便利性、普遍性及機身設計(如下表 3)。

表 4 個人對寵物餵食器的分類方式

項目	優缺點	類別
1	分開存放不同種類食物	食物存放
2	食物要能隔絕細小灰塵或蟲類	
3	要能避免寵物自行打開寵物餵食器	
4	能緊急停止動作	安全性及保護
5	具備防卡設計(轉盤運作時不會使寵物受傷)	
6	防止寵物破壞餵食器(機身、電線)	供電方式
7	多項供電方式	
8	方便拆洗	便利性
9	APP 遠端遙控	
10	遠程觀看寵物狀況、錄音及播放的功能	
11	價格	普遍性
12	設計複雜，使用者在操作上可能會有困難	機身設計
13	機器運作聲響過大	
14	機身不夠堅固，易被寵物破壞	
15	機體較高，寵物容易推倒，裡面的飼料可能溢出	

4. 寵物餵食器製作材料及工具

為了製作符合我特殊要求的餵食器，我參考有關製造和建構工具的網站，選擇幾種具「建築構成性」，且能達到「自定義製造」功能的製造和建構工具，分別有：樂高積木(樂高 EV3 機器人套裝)、3D 印表機、手工藝品材料(如紙張、布料、線條等)和機械結構套件(如 VEX 機器人套件、Meccano 結構套件等)，並整理其優缺點，如下表 4。

表 5 寵物餵食器製作材料比較表

工具名稱	優點	缺點	分析及討論	結論
樂高積木	1. 多樣性：提供各種不同主題和套裝，適合不同年齡層次的使用者。	1. 有限的建構複雜性：對尋求更高層次建構挑戰的使用者而言，樂高積木的建構複雜性相對有限。	其具備了高度的創造性和自定義性，能根據自己的需求進行設計，並且可以添加額外的功能。樂高的易修改性也可以經由不斷的測試而做調整，適合設計一個基本的寵物餵食器。	採用此項目。
3D 印表機	1. 自定製造：提供無限自定製造的可能性，從個性化物品到原型制作。 2. 多材料使用：可以使用各種不同的材料，包括塑膠、金屬、陶瓷等。 3. 快速原型：適用於製作快速原型，有助於快速產品開發。	1. 成本較高：3D 印表機的初始投資和材料成本較高。 2. 學習曲線：使用 3D 印表機需要學習相應的軟體和技術，可能有一定的學習曲線。	3D 印表機被排除的主要原因在於其高昂的成本和相對複雜的操作。首先，擁有一臺 3D 印表機需要投入額外的資金，對於學生的經濟能力較不理想。此外，3D 印表機的操作涉及到複雜的軟體和技術，需要相當程度的學習和熟練度，這對於初學者而言較困難。	不採用此項目。

探討大眾對於寵物餵食器之看法與自製寵物餵食器

<p>手工藝品材料</p>	<ol style="list-style-type: none"> 1. 廉價和容易獲得：手工藝品的材料通常相對便宜，容易在當地商店或線上獲得。 2. 多樣性：提供各種不同的材料，如紙張、布料、線條等，適用於各種手工藝品專案。 	<ol style="list-style-type: none"> 1. 依賴手工技巧：依賴使用者的人工技能，對於不擅長手工藝的人可能需要一些練習。 2. 製作速度較慢：與3D印表機相比，手工製作的速度相對較慢。 	<p>手工藝品材和工具雖然具有可取得性和相對低廉的優勢，但其製作速度較慢是主要不採用的原因之一。製作寵物餵食器可能需要一定的工時，而手工藝製作相對繁複，可能需要更長的時間完成。</p>	<p>不採用此項目。</p>
<p>機械結構套件</p>	<ol style="list-style-type: none"> 1. 教育價值：有助於教育機構進行STEM教育，提高學生對機械原理的理解。 2. 建構複雜性：提供複雜的機械結構挑戰，適合尋求更高層次建構的使用者。 3. 多樣性：有多種不同的套裝和模型，滿足不同需求。 	<ol style="list-style-type: none"> 1. 有限主題：相對於樂高積木，機械結構套件的主題較為專一，可能不適合所有使用者。 2. 組裝難度：對於初學者，機械結構套件的組裝難度可能較高。 	<p>機械結構套件雖然具有教育價值，但其有限的靈活性使其難以滿足寵物餵食器的定制需求。這些套件可能提供了特定的機械結構，但要根據不同的功能進行調整可能相對複雜。</p>	<p>不採用此項目。</p>

在挑選的過程中，我考慮到我所製作的餵食器將有真空功能，可能需要較複雜的機構來達成，並且因為製作後需要不斷的修正，因此我更傾向於能夠簡易組裝的材料，最後選擇樂高積木作為製作餵食器的材料。未來有更多技術及需求可能會考慮使用機械結構套件，因為他已廣泛應用，可輕易在網上找尋相關知識和購買零件。

5. 產品藍圖設計

為了將產品的功能做分類，對目標進行優先級排序。因此我參考「MoScow 優先級排序法。2023年9月27日。」（取自：MoSCoW 优先级排序法 -PraxisFramework）並使用

「MoScow 優先級排序法」，安排每個時間盒上所要完成的項目，在設計的過程裡，我將MoScow 優先級排序法分成四個階段。

MoScow 優先級排序法的四個階段，如下表 5。

表 6 MoScow 優先級排序法標準

標準	標準內容要求定義
必須具有	正在開發餵食器的基礎功能，沒有這些功能，開發是行不通的。
應該具有	很重要，但如果沒有完成，有可替代方案。
可以具有	在當前時間盒中並不重要，可以留到下一個時間盒。
想具有	如果它們可以在當下這個時間盒裡完成，就具有被採用的價值，但預期它們要在後續的時間盒裡才能完成。

我根據以上的優先級排序法標準，結合我個人的分類方式與 MoScow 優先級排序法中的時間盒概念，設計了產品藍圖，如下表 6。

表 7 自製寵物餵食器標準時間盒

標準	標準類別	標準內容要求
必須具有	基本功能	<ol style="list-style-type: none"> 1. 定時餵食：能設定餵食的時間。 2. 自動餵食：當指定時間到達，可由機器餵食寵物。 3. 具有真空功能：保持食物新鮮並防止變質。 4. 食物分類容器：分開存放不同種類食物。
應該具有	維護和清潔	<ol style="list-style-type: none"> 1. 易於清潔：設計易於取出不鏽鋼盒來清洗的部件。 2. 定期保養提醒：提醒主人進行維護和清潔。
	安全性和保護	緊急停用功能：在出現問題時能夠遠程停止餵食器運作。
可以具有及想具有	遠程控制與監視	<ol style="list-style-type: none"> 1. 手機應用程式連接：透過手機應用程式控制餵食器。 2. 網路攝像頭：觀察寵物，確保它們受到妥善照顧。 3. 通知功能：接收餵食進度、低食物水平等通知。 4. 語音指令：讓使用者通過語音指令與寵物互動。

6. DFC 開發方式

為了製作寵物餵食器，我參考《教育工作者的設計思考實踐手冊》這本書，書中將將研究流程分為五個步驟，分別是：探究問題、解讀問題、盡情發想、反覆測試和反覆改良，請見表 8。

步驟	項目	說明	對應的研究流程
1	探究問題	在這一階段，強調對問題的明確定義，需徹底瞭解問題的核心本質，並深入了解問題的背後原因及影響因素。	1. 寵物餵食器製作 2. 瞭解市場需求
2	解讀問題	在這一階段，我將問題分解並進行解釋。我將需要理解問題的各個方面。通過對問題的解讀，他們可以更清楚地理解問題的本質，並確定解決方案所需的目標和方向。	1. 整理並分析市面上的寵物餵食機、製作 SWOT 分析 2. 分析並製作寵物餵食機調查問卷、整理並分析
3	盡情發想	在這一階段，作者鼓勵盡情發揮創意，提出各種可能的解決方案，並可使用各種創意工具和技巧，如腦力激盪、角色扮演和故事板，來生成新的想法和概念，盡可能多地收集解決方案，無論這些解決方案聽起來多麼不切實際。	1. 依照調查結果製作寵物餵食機
4	反覆測試	在這一階段，我將會通過不斷的測試原型及討論，以獲取改進的建議及方向。	1. 對寵物餵食器進行測試
5	反覆改良	在這一階段，我將會將根據測試的結果，不斷地調整和改進解決方案。這可能涉及修改原型，重新設計某些功能，或者完全重新思考解決方案的方向。通過持續的改進過程，最終完成整個餵食器。	1. 依據測試結果做改良 2. 討論並得出結果、結論

總體而言，這些方法不僅僅是教育工作者的設計工具，更是一種思考和解決問題的好方法。我將會依照此方法製作我的寵物餵食器。

參、研究方法

一、研究流程

我透過問卷調查並整理和分析市面上寵物餵食器的功能後，製作了關於餵食器的問卷，瞭解大眾對於寵物餵食器的看法和需求，後製作標準表，確定製作寵物餵食器的方向及功能。待製作完成後，再經由測試得出問題並加以改良，並進一步討論、分析，最後得出結論。



二、研究架構

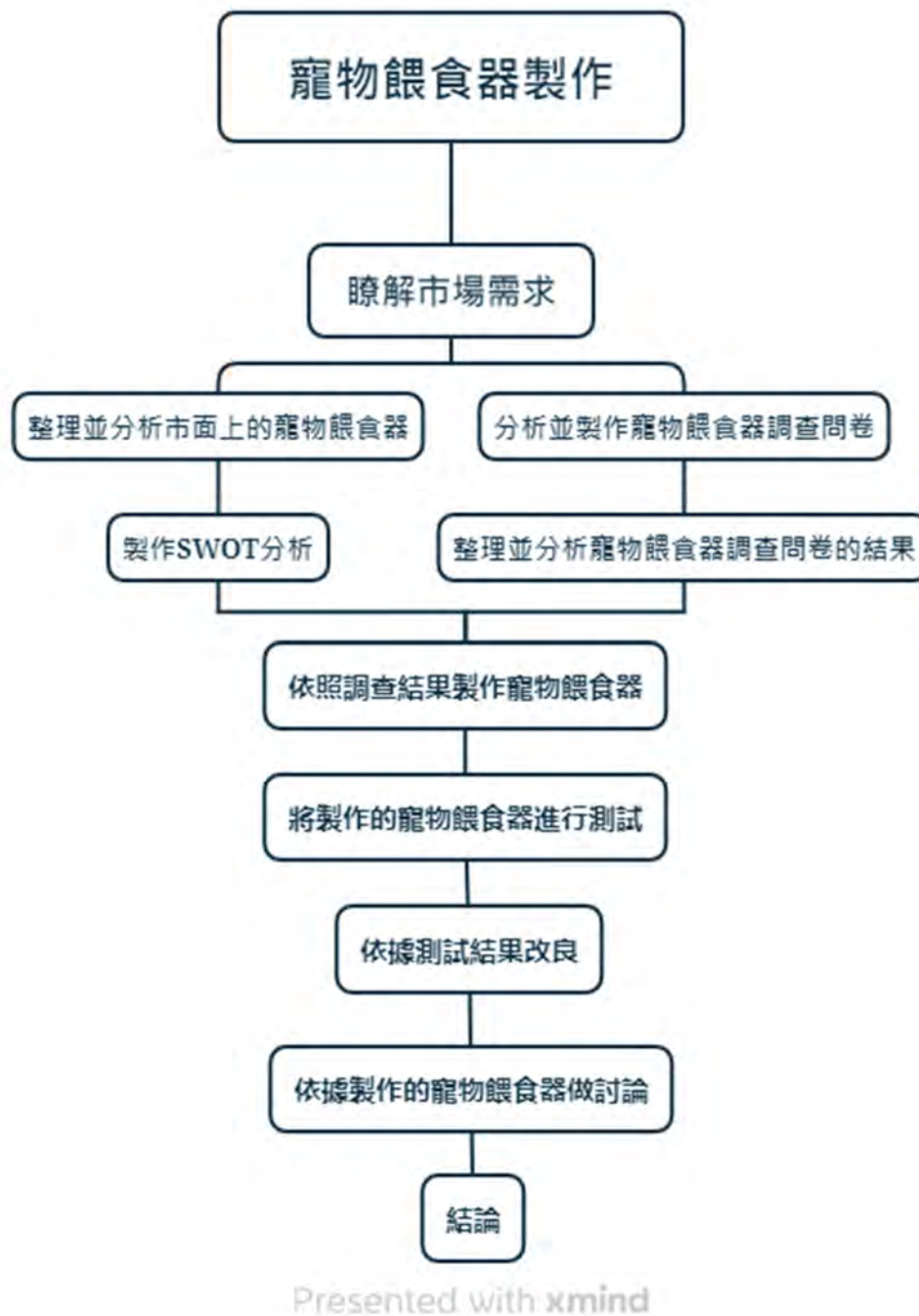


圖 1 研究架構圖

三、研究方法

我根據「設計思考實踐手冊」，採用此書中提到的開發方式—DFC(Design For Change)，將研究流程分為五個步驟，分別是：探究問題、解讀問題、盡情發想、反覆測試和反覆改良，我將根據以上步驟來製作我的寵物餵食器，詳細執行方法請見表 9。我也採用 MoScow 優先級排序法標準，將寵物餵食器的功能分類到不同的時間盒，使產品品質符合標準。我還參考了朝陽科技大學中的問卷設計、各大寵物餵食器的網站，再結合自己使用餵食器的經驗，製作出問卷，瞭解大眾對於寵物餵食器功能的重視程度，最後再使用 EV3 和 Lego 積木來製作寵物餵食器。

表 9 開發方式—DFC 的詳細執行方法

步驟	項目	具體的執行方法
1	探究問題	探究多層面的問題。在這一階段，我定義了此研究的最終目標「製作餵食器」，並參考市面上的餵食器，審視這個問題的各種條件和釐清可能遇到的問題。
2	解讀問題	制定餵食器標準表。在這一階段，我將透過各大有關寵物餵食器的網站，瞭解市場需求和統整各種功能，接著根據搜尋結果製作問卷，製作 SWOT 分析來得知大眾較重視的功能，確定解決方案所需的目標和方向
3	盡情發想	製作餵食器的過程。在這一階段，我將根據餵食器的標準表，並嘗試用各種材料來製作餵食器。
4	反覆測試	測試所製作的寵物餵食器並記錄需改良的問題。
5	反覆改良	根據反覆測試發現的問題做改良。在這一階段，我將會不斷的重複測試和改良這兩個步驟，直到符合我製作餵食器的標準。

四、研究對象

本次共回收 102 份問卷，其中有 88 份有效問卷，14 份無效問卷。因主要目的是為了瞭解市場及大眾對於市售寵物餵食器的認知及需求，本研究將有效問卷中的受訪者分為三類族群，分別為有養寵物且有使用寵物餵食器、有養寵物且沒有使用寵物餵食器和沒有養寵物且沒有使用寵物餵食器(請見表 8)。本研究將著重在有養寵物且有使用寵物餵食器和有養寵物且沒有使用寵物餵食器這兩類族群的意見上。

表 10 受訪對象之族群

族群	是否飼養寵物	是否使用寵物餵食器
A	是	是
B	是	否
C	否	否

五、研究器材及用途

圖示				
名稱	筆記型電腦	EV3 主機	EV3 相關配件	LEGO 積木
用途	筆記型電腦是一種便攜式計算機，通常具有摺疊式屏幕，適用於各種工作和娛樂用途，例如文字處理、上網、多媒體播放等。	EV3 主機是一個小型的單板計算機，用於各種嵌入式計算和 DIY 項目。它基於 Linux 操作系統，具有各種輸入/輸出接口，可用於程式開發和控制應用。	EV3 是一套由 LEGO 推出的可程式機器人積木套件，其中包含多種配件，包括：大(中)型 EV3 馬達、顏色(陀螺儀)感應器、觸控感應器、超音波(紅外線)感應器等。	LEGO 積木是一種經典的構建玩具，由彩色塑料積木組件組成(動力機械組有新增齒輪、有孔橫桿等)，可以互相嵌合，用於創建各種結構、模型和建築。

圖示		
名稱	ESP32	不鏽鋼真空碗
用途	ESP32 是一系列低成本，低功耗的單晶片微控制器，整合了 Wi-Fi 和雙模藍牙。ESP32 系列採用 Tensilica Xtensa LX6 雙核和單核微處理器，內建無線開關，RF 換能器，功率放大器，低噪聲接收放大器，濾波器和電源管理模組。	是一種製成不鏽鋼材料的容器，通常用於在真空環境中處理、儲存食品或其他物體。不鏽鋼碗通常具有耐腐蝕性和高溫耐受性。

六、寵物餵食器的實際製作方法

(一) 寵物餵食器功能及設計結構

根據我在「表 3 本篇參考市售寵物餵食器分類方式」中的寵物餵食器之功能，我上網找尋相關資料並思考其製作方法，讓我在完成問卷統計後，能直接製作寵物餵食器的原型，如下表 7。

表 11 寵物餵食器的實際製作方法

項目	問題	解決方法
1	分開存放不同種類食物	以市售寵物餵食器為例，若是要能分開存放不同種類食物，大多採用分隔的餵食碗。當定時時間一到，機器就會透過轉動等方式進行餵食餵食。
2	食物要能隔絕細小灰塵或蟲類	關於寵物餵食器內的食物要能隔絕細小灰塵或蟲類，我採用真空盒搭配密封條、防潮乾燥系統(如乾燥劑等)，並透過大氣壓力來阻隔外界的空氣。
3	能避免寵物自行打開寵物餵食器	寵物能打開餵食器的方式有很多種，例如：推翻餵食器使飼料傾洩而出，抑或是較聰明的貓狗能自行打開餵食器的蓋子，因此可以使用真空設計來密封蓋子，或避免設計較挑高的機體(詳見項目 6、14 和 15)。
4	能緊急停止動作	分為主動和被動兩種：被動是指由飼主可透過 APP 或是實體餵食器自行停止餵食計劃；主動則是寵物餵食器偵測到情況後自行判斷，並停止現在正在執行的動作(詳見項目 5)。
5	具備防卡設計	分為紅外線感測和壓力感測：紅外線感測是透過紅外線感測器，當偵測到有異物或寵物在機器運作的範圍之內(如：電動轉盤、旋轉出口)，就會停止現在正在執行的動作；壓力感測通常運用在多格旋轉餵食器，若是寵物踩進碗內，下方的壓力感測器就會偵測到而停止動作。
6	防止寵物破壞餵食器(包括機身、電線等)	使用較堅固的材質，如：木頭、鋁或高強度塑膠等，作為寵物餵食器的外殼材質。
7	2 種以上的供電方式	常見的供電方式有：自行裝入電池、充電(內建電池)和插座直接供電，可根據需求選擇供電方式。
8	方便拆洗	為了讓寵物餵食器方便清潔，大多市售的餵食器會將餵食碗(或儲糧倉)和餵食器本體做成可拆解式的，並且不具有太多細小的縫隙或卡榫(易藏汙納垢)。
9	能用 APP 遠端遙控	要能用 APP 遠端遙控，需要撰寫程式，使用遠端的電腦，透過網際網路，遠距控制家裡的餵食器，達成遠程觀看寵物狀況、錄音及播放等功能。

探討大眾對於寵物餵食器之看法與自製寵物餵食器

10	具視訊(能遠程觀看寵物狀況)、錄音及播放的功能	可在寵物餵食機上裝設攝像機、麥克風和喇叭，再透過 APP 遠端遙控。
11	價格過高	盡量減少開發時的成本和選擇價格較為低廉的材料製作餵食器。
12	設計過於複雜，使用者在操作上可能會有困難	減少選項和步驟，讓整個操作流程更直觀，降低使用門檻。同時，使用圖形化的指南或圖示替代文字說明，有助於提高使用者對每個功能和設定的理解。
13	機器造成的聲響過大	機器造成的聲響過大可能會驚嚇到家中的寵物，可使用機械潤滑油(齒輪油等)減少零件之間的摩擦還有設計合理的結構。
14	機身不夠堅固，易被寵物破壞	可設計無電線的充電餵食機，或者增加電線防咬套的產品，以避免貓咪咬電線而發生意外的可能。
15	寵物容易推倒，裡面的飼料可能溢出	要防止這個問題發生，可透過底部增加吸盤(若吸附表面不平滑可能無法使用)，或是避免設計較挑高的機體(多格旋轉餵食器較有可能達到此標準)等。

肆、研究方法與過程

一、寵物餵食器功能問卷統計

以下是針對 A 群體進行統計：

(一) 寵物的相關問題

a. 1-1 的題目為「是否曾經擁有寵物」，但此統計的分類為「有寵物/有使用餵食器」，故不將此題列入統計。

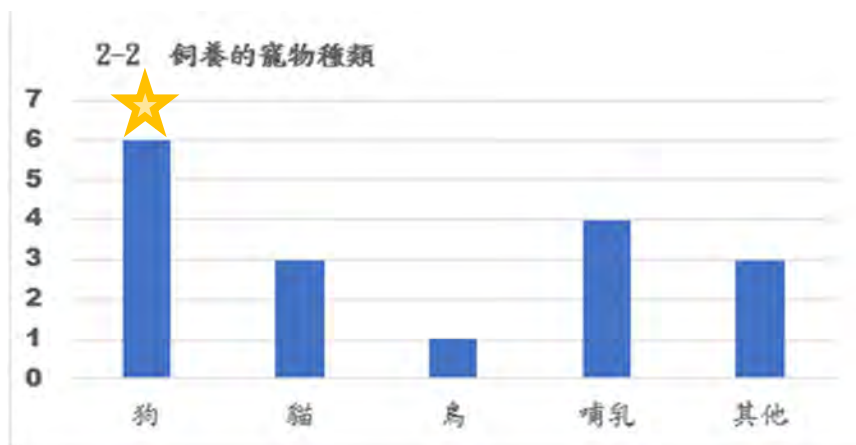
b. 1-2 寵物種類(複選)

在本次的受訪者中(12-45 歲、男+女)，飼養狗的有 6 位；飼養貓的有 3 位；飼養鳥的有 1 位；飼養小型哺乳類動物（例如兔子、倉鼠等）的有 4 位；其他的有 3 位(補充請見下圖)，相關比例請見下圖。

群體 A

其他補充：未滿12歲
男：烏龜2/昆蟲2/守宮1/ 甲蟲1/蝸牛1/魚2
女：魚5/甲蟲3/烏龜1/ 昆蟲1/守宮1/蜘蛛1 /蝸牛1/青蛙1/蝦1

年齡	性別	狗	貓	鳥	哺乳	其他	合計
12歲-45歲	男+女	6	3	1	4	3	17



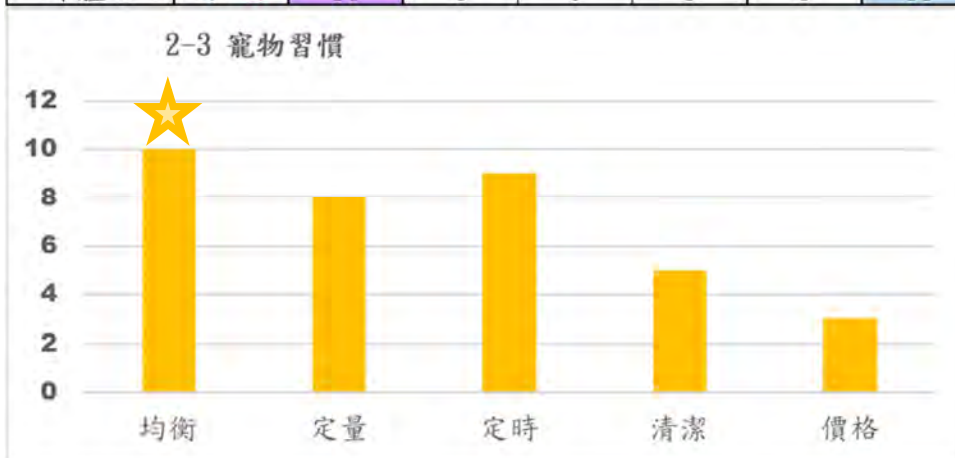
▲寵物種類(複選)

c. 1-3 寵物習慣(複選)

在分析及整理市面上的寵物餵食器時，我發現以下幾種條件會影響使用者對於購買餵食器的意願，故將此題列入問卷的問題中。

在本次的受訪者中(12-45 歲、男+女)，飼養的寵物之習慣，覺得寵物需要飲食 均衡(均衡)的有 10 位；需要定量餵食(定量)的有 8 位；需要固定時間餵食(定時)的有 9 位；寵物餵食器要能方便清潔(清潔)的有 5 位；寵物餵食器價格是考量的因素(價格)的有 3 位，相關比例請見下圖。

年齡	性別	均衡	定量	定時	清潔	價格	合計
群體 A	男+女	10	8	9	5	3	35



▲寵物習慣(複選)

d. 1-4 寵物進食餐數

在本次的受訪者中(12-45 歲、男+女)，飼養的寵物每天通常進食 1 餐的有 1 位；2 餐的有 4 位；3 餐的有 2 位；4 餐的有 1 位；5 餐的有 5 位；無固定時間餵食的有 3 位，其中有 2 位是在寵物表現出飢餓行為時餵食，1 位則是在其他原因時餵食，相關比例請見下圖。

年齡	性別	1餐	2餐	3餐	4餐	5餐	無固定	合計	無固定	
群體 A	男+女	1	4	2	1	0	3	11	飢餓	其他
									2	1



▲寵物進食餐數

(二) 對寵物餵食器的使用現況及認知

a. 2-1 的題目為「是否使用寵物餵食器」，但此統計的分類為「有寵物/有使用餵食器」，故不將此題列入統計。

b. 2-2 寵物餵食器使用主要原因(複選)

在本次的受訪者中(12-45 歲、男+女)，關於寵物餵食器使用主要原因，方便定時餵食的有 7 位；避免忘記餵食的有 5 位；寵物有特殊飲食需求的有 2 位；超過一天無人在家的有 8 位；其他原因的有 0 位，相關比例請見下圖。

年齡	方便	怕忘記	特需	無人	其他	合計
群體 A	7	5	2	8	0	22



▲ 寵物餵食器使用主要原因(複選)

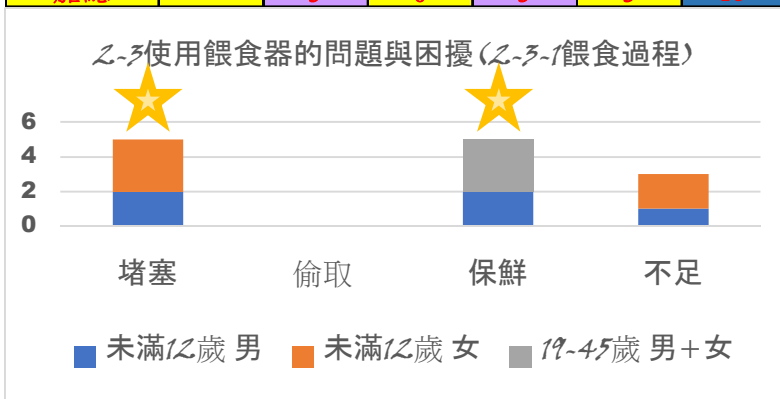
c. 2-3 「曾經遇到使用寵物餵食器的問題或困擾？」

因題目的排列順序有誤，導致部分沒有使用過寵物餵食器的受訪者可能不清楚題意，因而回答了這題需要有使用寵物餵食器經驗才能回答的問題。但在此統計中，受訪對象為「有寵物/有使用餵食器」的人，因此還是將此題列入統計。

(a) 2-3-1 餵食過程

在本次的受訪者中(12-45 歲、男+女)，曾使用寵物餵食器遇到食物堵塞或卡住的有 5 位；寵物偷取機器內食物的有 0 位；食物存儲及保鮮問題的有 5 位；儲存食物的容器容量不足的有 3 位，相關比例請見下圖。

年齡	性別	堵塞	偷取	保鮮	不足	合計
未滿12歲	男	2	0	2	1	5
	女	3	0	0	2	5
19-45歲	男+女	0	0	3	0	3
加總		5	0	5	3	13

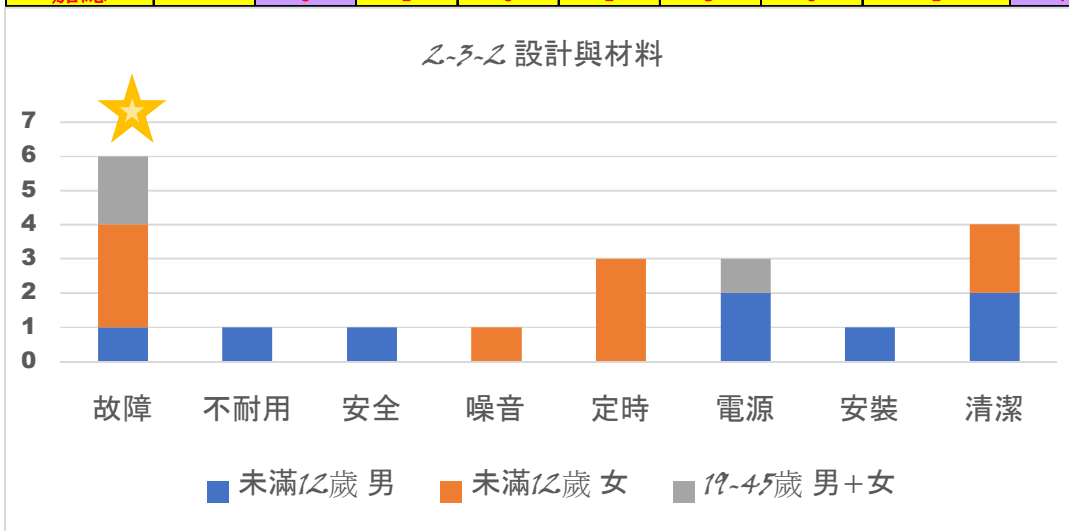


▲使用餵食器的問題與困擾

(b) 2-3-2 設計與材料

在本次的受訪者中(12-45 歲、男+女)，曾使用寵物餵食器遇到餵食器故障的有 6 位；有不耐用材料的有 1 位；有安全疑慮的有 0 位；有噪音問題的有 1 位；有定時錯誤的有 3 位；有電池或電源問題的有 0 位；有餵食器安裝問題的有 1 位；有清潔或維護餵食器困難的有 4 位，相關比例請見下圖。

年齡	性別	故障	不耐用	安全	噪音	定時	電源	安裝	清潔	合計
未滿12歲	男	1	1	1	0	0	2	1	2	8
	女	3	0	0	1	3	0	0	2	9
19-45歲	男+女	2	0	0	0	0	1	0	0	3
加總		6	1	0	1	3	0	1	4	20



▲設計與材料

(c) 2-3-3 遠程控制和技術

在本次的受訪者中(12-45 歲、男+女)，曾使用寵物餵食器遇到網絡連接問題的有 5 位；無法通知問題的有 5 位；維護困難的有 5 位，相關比例請見下圖。

年齡	性別	網路	通知	維護	合計
未滿12歲	男	3	1	1	5
	女	1	3	3	5
19-45歲	男+女	1	1	1	3
加總		5	5	5	15



▲遠程控制和技術

(d) 2-3-4 其他問題。

本次的受訪者中(12-45 歲、男+女)，關於使用寵物餵食器的問題與困擾，沒有其他問題的有 10 位；有其他問題的有 1 位，問題補充為狗會破壞(補充請見下圖)，相關比例請見下圖。

年齡	性別	否	是	合計
未滿12歲	男	2	1	3
	女	5	0	5
19-45歲	男+女	3	0	3
加總		10	1	11

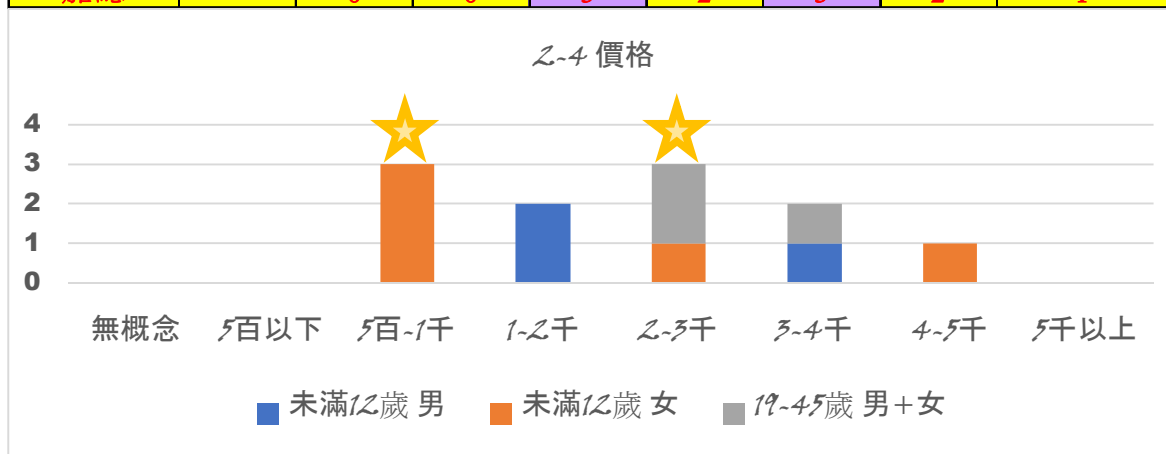
▲其他問題

d. 2-4 價格

在分析及整理市面上的寵物餵食器時，我發現由於餵食器的功能、材質、品牌等因素，皆會間接影響餵食器的價格，因此想瞭解大眾對於餵食器的理想價格。以市面上的寵物餵食器為例，基本款的寵物餵食器價格大約在 900 到 1500 元，而具有更多功能或特殊設計的高端餵食器則可能價格更高，大約在 1500 到 2000 元甚至更高。

在本次的受訪者中(12-45 歲、男+女)，關於一款理想的寵物餵食器的價格，無概念的有 0 位；5 百以下的有 0 位；5 百-1 千的有 3 位；1-2 千的有 2 位；2-3 千的有 3 位；3-4 千的有 2 位；4-5 千的有 1 位；5 千以上的有 0 位，相關比例請見下圖。

年齡	性別	無概念	5百以下	5百-1千	1-2千	2-3千	3-4千	4-5千	5千以上
未滿12歲	男				2		1		
	女			3		1		1	
19-45歲	男+女					2	1		
加總		0	0	3	2	3	2	1	0



▲ 價格

e. 2-5 只是為確保此問卷為有效問卷，無效問卷皆已排除，故不將此題列入統計。

(三) 重視程度調查

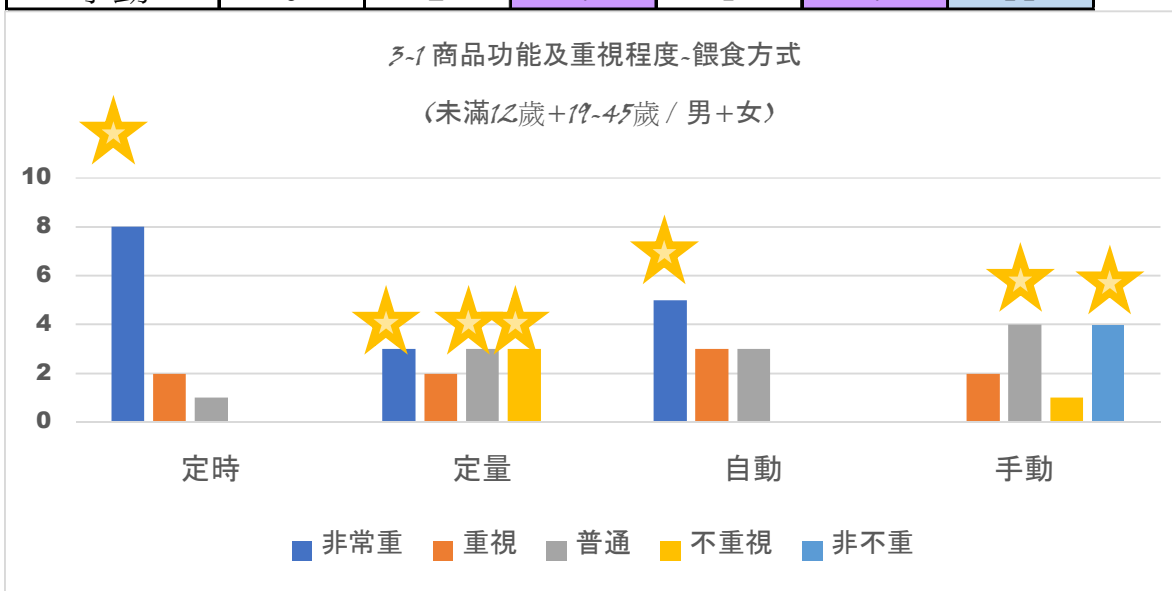
a. 3-1 商品功能及重視程度

(a) 3-1-1 餵食方式的重視程度

原本我想針對使用餵食器經驗的多寡來分類，但因問卷中沒有類似的題目可得知受訪者使用寵物餵食器的時間長短，故將數據合併後再進行分析。

在本次的受訪者中(12-45 歲、男+女)，關於商品功能及重視程度(餵食方式)，「可定時餵食」的項目覺得非常重視的有 8 位；重視的有 2 位；普通的有 1 位；不重視及非常不重視的皆為 0 位。「可定量餵食」的項目覺得非常重視的有 3 位；重視的有 2 位；普通的有 3 位；不重視的有 3 位；非常不重視的有 0 位。「可自動餵食」的項目覺得非常重視的有 5 位；重視的有 3 位；普通的有 3 位；不重視及非常不重視的皆為 0 位。「可臨時手動餵食」的項目覺得非常重視的有 0 位；重視的有 2 位；普通的有 4 位；不重視的有 1 位；非常不重視的有 4 位。相關比例請見下圖。

餵食方式	非常重	重視	普通	不重視	非不重	合計
定時	8	2	1	0	0	11
定量	3	2	3	3	0	11
自動	5	3	3	0	0	11
手動	0	2	4	1	4	11

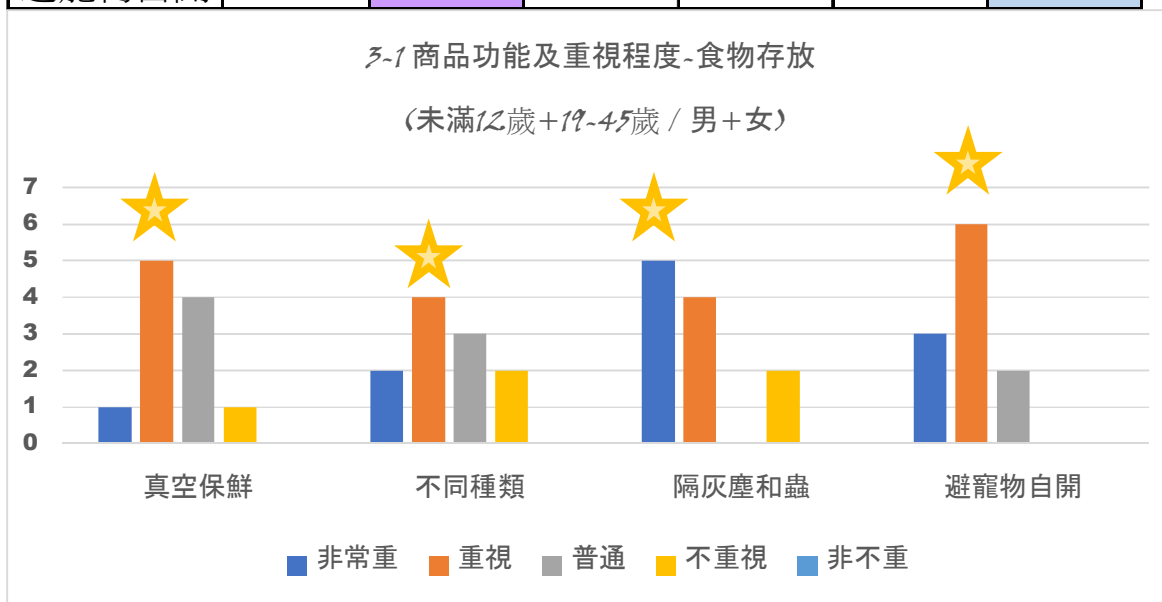


▲ 餵食方式的重視程度

(b) 3-1-2 食物存放的重視程度

在本次的受訪者中(12-45 歲、男+女)，關於商品功能及重視程度(食物存放)：「具真空保鮮的功能」的項目覺得非常重視的有 1 位；重視的有 5 位；普通的有 4 位；不重視的有 1 位；非常不重視的有 0 位。「分開存放不同種類的食物」的項目覺得非常重視的有 2 位；重視的有 4 位；普通的有 3 位；不重視的有 2 位；非常不重視的有 0 位。「食物要能隔絕細小灰塵或蟲類」的項目覺得非常重視的有 5 位；重視的有 4 位；普通的有 0 位；不重視的有 2 位；非常不重視的有 0 位。「能避免寵物自行打開餵食器」的項目覺得非常重視的有 3 位；重視的有 6 位；普通的有 2 位；不重視及非常不重視的皆為 0 位。相關比例請見下圖。

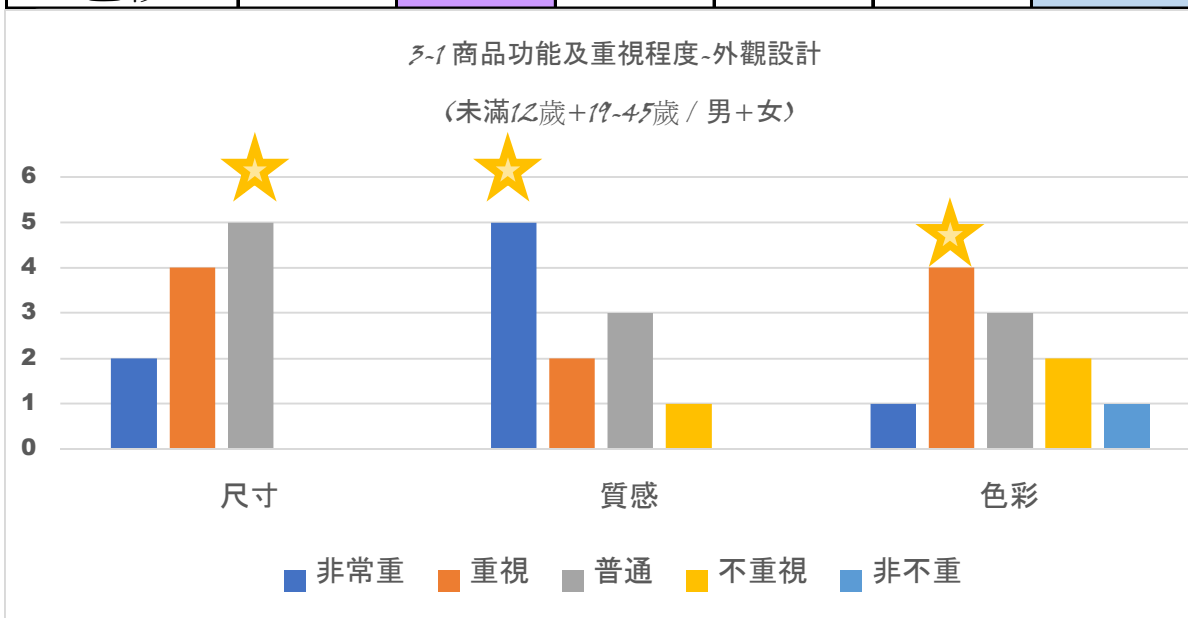
食物存放	非常重	重視	普通	不重視	非不重	合計
真空保鮮	1	5	4	1	0	11
不同種類	2	4	3	2	0	11
隔灰塵和蟲	5	4	0	2	0	11
避寵物自開	3	6	2	0	0	11



(c) 3-1-3 外觀設計的重視程度

在本次的受訪者中(12-45 歲、男+女)，關於商品功能及重視程度(外觀設計)：「尺寸造型」的項目覺得非常重視的有 2 位；重視的有 4 位；普通的有 5 位；不重視的有 0 位；非常不重視的有 0 位。「整體質感」的項目覺得非常重視的有 5 位；重視的有 2 位；普通的有 3 位；不重視的有 1 位；非常不重視的有 0 位。「色彩規劃」的項目覺得非常重視的有 1 位；重視的有 4 位；普通的有 3 位；不重視的有 2 位；非常不重視的有 1 位。相關比例請見下圖。

外觀設計	非常重	重視	普通	不重視	非不重	合計
尺寸	2	4	5	0	0	11
質感	5	2	3	1	0	11
色彩	1	4	3	2	1	11

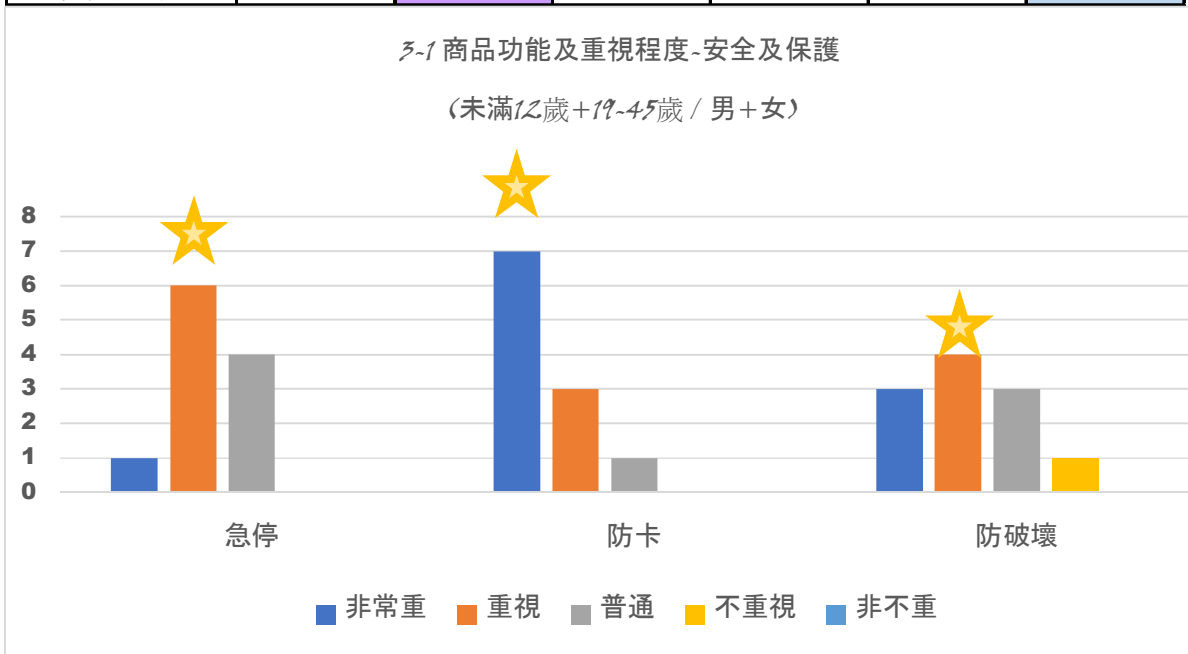


(d) 3-1-4 安全性及保護的重視程度

在本次的受訪者中(12-45 歲、男+女)，關於商品功能及重視程度(安全及保護)：

「能緊急停停止動作」的項目覺得非常重視的有 1 位；重視的有 6 位；普通的有 4 位；不重視及非常不重視的皆為 0 位。「具防卡設計(即使寵物踩進碗內，正巧碰上轉盤運作時也不會因此受傷)」的項目覺得非常重視的有 7 位；重視的有 3 位；普通的有 1 位；不重視及非常不重視的皆為 0 位。「防止寵物破壞餵食器」的項目覺得非常重視的有 3 位；重視的有 4 位；普通的有 3 位；不重視的有 1 位；非常不重視的有 0 位。相關比例請見下圖。

安全及保護	非常重	重視	普通	不重視	非不重	合計
急停	1	6	4	0	0	11
防卡	7	3	1	0	0	11
防破壞	3	4	3	1	0	11

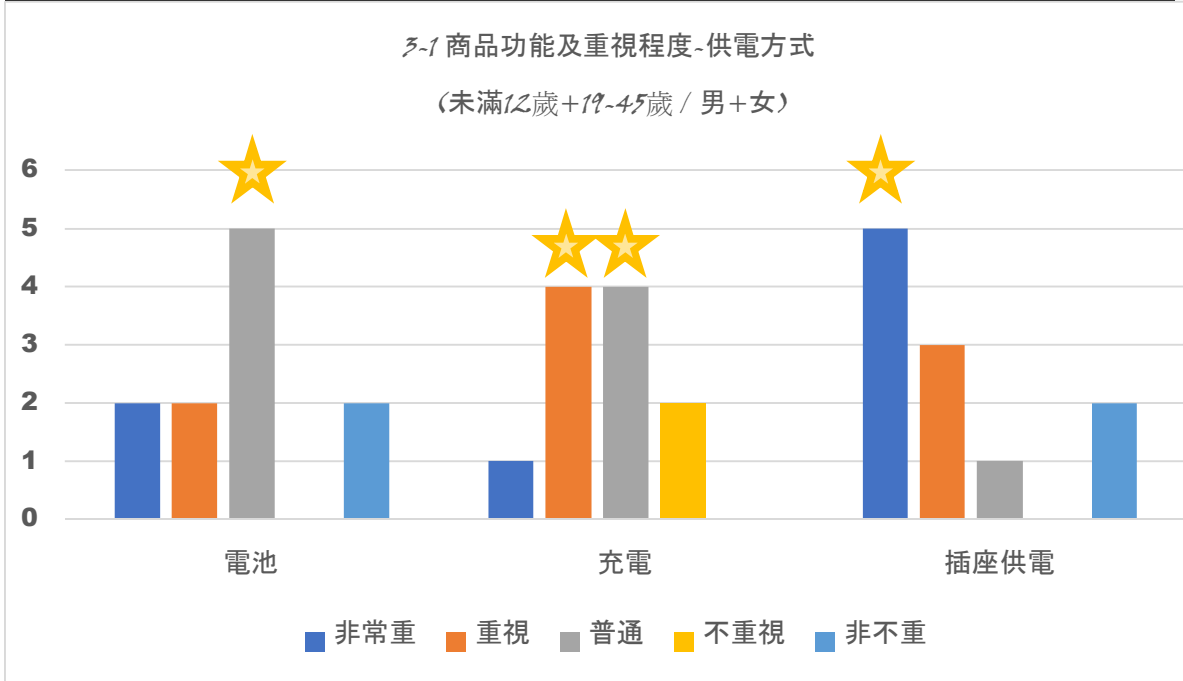


▲安全性及保護的重視程度

(e) 3-1-5 供電方式的重視程度

在本次的受訪者中(12-45 歲、男+女)，關於商品功能及重視程度(供電方式)：「自行裝入電池」的項目覺得非常重視的有 2 位；重視的有 2 位；普通的有 5 位；不重視的有 0 位；非常不重視的有 2 位。「充電(內建電池)」的項目覺得非常重視的有 1 位；重視的有 4 位；普通的有 4 位；不重視的有 2 位；非常不重視的有 0 位。「插座直接供電」的項目覺得非常重視的有 5 位；重視的有 3 位；普通的有 1 位；不重視的有 0 位；非常不重視的有 2 位。相關比例請見下圖。

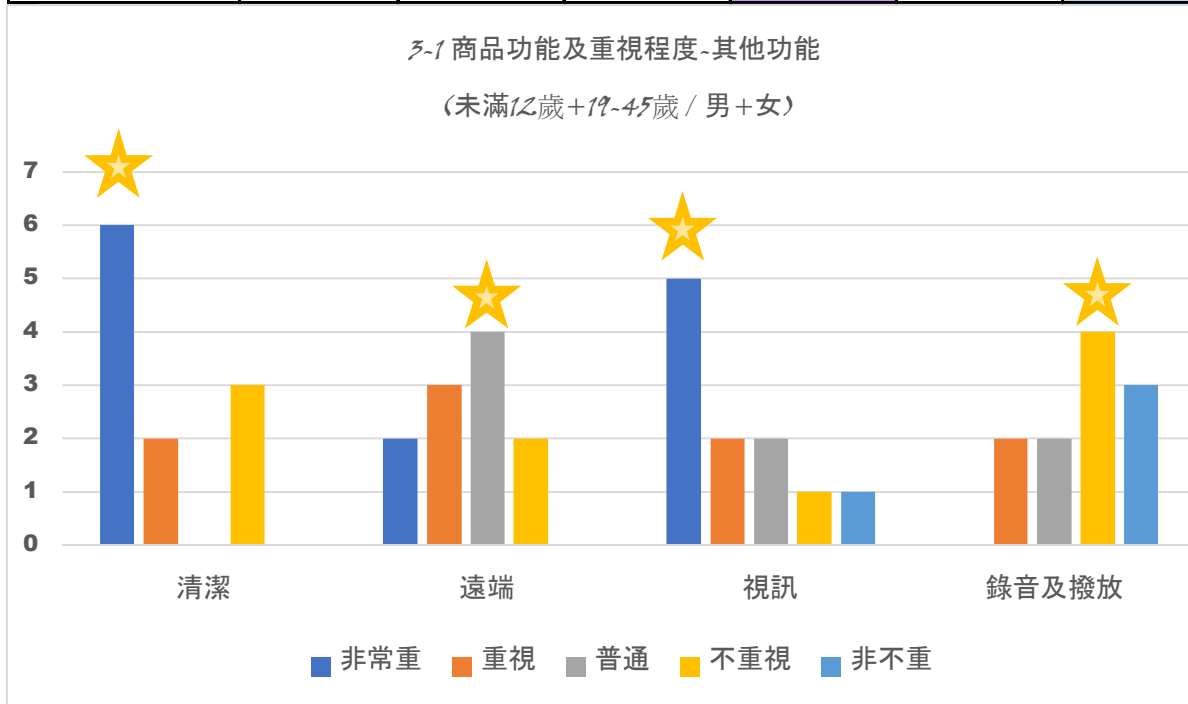
供電	非常重	重視	普通	不重視	非不重	合計
電池	2	2	5	0	2	11
充電	1	4	4	2	0	11
插座供電	5	3	1	0	2	11



(f) 3-1-6 其他功能的重視程度

在本次的受訪者中(12-45 歲、男+女)，關於商品功能及重視程度(其他功能)，「要能方便清潔」的項目覺得非常重視的有 6 位；重視的有 2 位；普通的有 0 位；不重視的有 3 位；非常不重視的有 0 位。「透過手機應用程式遠端遙控」的項目覺得非常重視的有 2 位；重視的有 3 位；普通的有 4 位；不重視的有 2 位；非常不重視的有 0 位。「具視訊功能(能遠程觀看寵物在家中的狀況)」的項目覺得非常重視的有 5 位；重視的有 2 位；普通的有 2 位；不重視的有 1 位；非常不重視的有 1 位。「具錄音及播放功能」的項目覺得非常重視的有 0 位；重視的有 2 位；普通的有 2 位；不重視有 4 位；非常不重視的有 3 位。相關比例請見下圖。

其他功能	非常重	重視	普通	不重視	非不重	合計
清潔	6	2	0	3	0	11
遠端	2	3	4	2	0	11
視訊	5	2	2	1	1	11
錄音及撥放	0	2	2	4	3	11



(四) 寵物餵食器功能問卷分析及結論

因為最終目的是製作寵物餵食器，因此我將著重在第四大題的重視功能調查。我將問卷調查中的李克特量表詳細列出，方便各個選項能夠有效的量化(非常重視為 2 分，重視為 1 分，普通為 0 分，不重視為-1 分，非常不重視為-2 分)並做討論，加總及依計分排列次第，以便得知各群體最重視的功能，並綜合各群體的有效問卷，探討其可能原因(請見表 11、12 和 13)。

1. 有養寵物且有使用寵物餵食器：11 份

表 12 有養寵物且有使用寵物餵食器量化計分

餵食方式的重視程度			
編號	項目	總計分	次第
1.	可定時餵食	18	1
2.	可定量餵食	5	3
3.	可自動餵食	13	2
4.	可臨時手動餵食	-7	4
食物存放的重視程度			
編號	項目	總計分	次第
1.	具真空保鮮的功能	6	2
2.	分開存放不同種類食物	6	2
3.	食物要能隔絕細小灰塵或蟲類	12	1
4.	能避免寵物自行打開餵食器	12	1
外觀設計的重視程度			
編號	項目	總計分	次第
1.	尺寸造型	8	2
2.	整體質感	11	1
3.	色彩規劃	4	3
安全性及保護的重視程度			
編號	項目	總計分	次第
1.	能緊急停止動作	8	3
2.	具防卡設計 (即使寵物踩進碗內，正巧碰上轉盤運作時也不會因此受傷)	17	1
3.	防止寵物破壞餵食器	9	2
供電方式的重視程度			
編號	項目	總計分	次第
1.	自行裝入電池	2	3
2.	充電 (內建電池)	4	2
3.	插座直接供電	9	1
其他功能的重視程度			
編號	項目	總計分	次第
1.	要能方便清潔	9	1
2.	透過手機應用程式遠端遙控	5	2
3.	具視訊功能 (能遠程觀看寵物在家中的狀況)	9	1
4.	具錄音及播放功能	-8	3

2. 有養寵物且沒有使用寵物餵食器：34 份

表 13 有養寵物且沒有使用寵物餵食器量化計分

餵食方式的重視程度			
編號	項目	總計分	次第
1.	可定時餵食	27	3
2.	可定量餵食	35	1
3.	可自動餵食	31	2
4.	可臨時手動餵食	1	4
食物存放的重視程度			
編號	項目	總計分	次第
1.	具真空保鮮的功能	25	2
2.	分開存放不同種類食物	21	3
3.	食物要能隔絕細小灰塵或蟲類	50	1
4.	能避免寵物自行打開餵食器	14	4
外觀設計的重視程度			
編號	項目	總計分	次第
1.	尺寸造型	19	2
2.	整體質感	41	1
3.	色彩規劃	5	3
安全性及保護的重視程度			
編號	項目	總計分	次第
1.	能緊急停止動作	41	2
2.	具防卡設計 (即使寵物踩進碗內，正巧碰上轉盤運作時也不會因此受傷)	54	1
3.	防止寵物破壞餵食器	19	3
供電方式的重視程度			
編號	項目	總計分	次第
1.	自行裝入電池	3	3
2.	充電 (內建電池)	30	2
3.	插座直接供電	33	1
其他功能的重視程度			
編號	項目	總計分	次第
1.	要能方便清潔	42	1
2.	透過手機應用程式遠端遙控	38	3
3.	具視訊功能 (能遠程觀看寵物在家中的狀況)	41	2
4.	具錄音及播放功能	-4	4

3. 沒有養寵物且沒有使用寵物餵食器：43 份

表 14 沒有養寵物且沒有使用寵物餵食器量化計分

餵食方式的重視程度			
編號	項目	總計分	次第
1.	可定時餵食	50	1
2.	可定量餵食	44	2
3.	可自動餵食	32	3
4.	可臨時手動餵食	-11	4
食物存放的重視程度			
編號	項目	總計分	次第
1.	具真空保鮮的功能	38	2
2.	分開存放不同種類食物	8	4
3.	食物要能隔絕細小灰塵或蟲類	51	1
4.	能避免寵物自行打開餵食器	30	3
外觀設計的重視程度			
編號	項目	總計分	次第
1.	尺寸造型	42	2
2.	整體質感	47	1
3.	色彩規劃	-1	3
安全性及保護的重視程度			
編號	項目	總計分	次第
1.	能緊急停止動作	37	2
2.	具防卡設計 (即使寵物踩進碗內，正巧碰上轉盤運作時也不會因此受傷)	71	1
3.	防止寵物破壞餵食器	26	3
供電方式的重視程度			
編號	項目	總計分	次第
1.	自行裝入電池	13	3
2.	充電 (內建電池)	34	1
3.	插座直接供電	32	2
其他功能的重視程度			
編號	項目	總計分	次第
1.	要能方便清潔	42	2
2.	透過手機應用程式遠端遙控	41	3
3.	具視訊功能 (能遠程觀看寵物在家中的狀況)	49	1
4.	具錄音及播放功能	-13	4

4. 各群體問卷之綜合分析

綜合問卷結果，我發現以下幾點原因及調查結果會成為製作餵食器功能的依據並整理如下表所示。(請見表 14)：

表 15 各群體問卷之綜合分析

項目	綜和分析	類別
1	所列出的各個選項：可定時、定量、自動餵食，儘管計分有高低之分，但皆是一般餵食器的 基礎功能和必要條件 。	餵食方式
2	「可臨時手動餵食」此功能普遍計分較低，因此較無建設此功能的必要性。	
3	我自製餵食器的最終功能是具有真空保鮮的功能(一定會達成)，我將採用真空盒來達成，因此一定可達到「分開存放不同種類食物」、「食物要能隔絕細小灰塵或蟲類」兩種功能。	食物存放
4	為了達成「能避免寵物自行打開餵食器」此功能，我只需將整個餵食器蓋上壓克力罩子即可解決，也可清楚的作演示和檢查錯誤。	
5	外觀設計的功能與餵食器功能較無關聯性，因此不列入考慮。	外觀設計
6	檢視問卷結果後，我發現「具防卡設計」的功能計分較高。可利用壓克力罩即可達成。	安全性及保護
7	有使用寵物餵食器經驗的，通常偏向於插座供電，我推測是由於插座除了天災或斷電等意外，可持續供電。因此我將使用此種供電方式。	供電方式
8	關於資訊類的功能，依我目前的能力可能還無法達成。而關於「要能方便清潔」的功能，會採個別餵食盒的方式餵食，來保證在整體餵食環境上能夠簡單控制。	其他功能

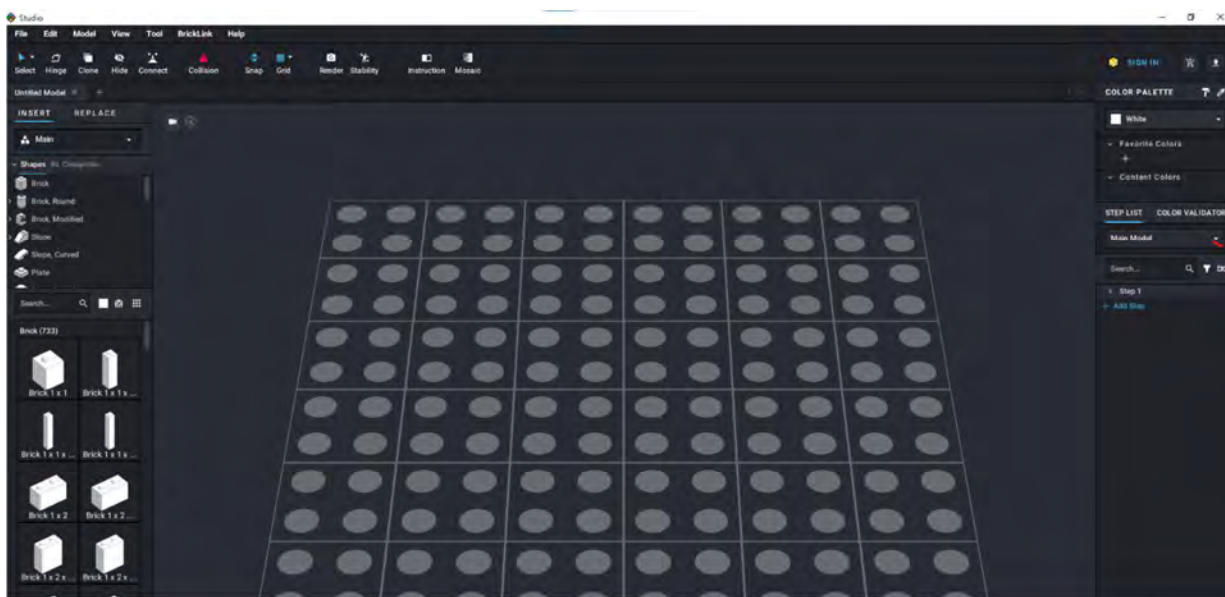
二、寵物餵食器原型製作

(一) 寵物餵食器的設計功能

根據本研究中的問卷及其分析結果，我選擇大眾較為重視的功能，並且依據我的個人需求，篩選較為重要的餵食器功能作為製作的目標，分別有：定時、定量、真空保鮮、防卡和方便清潔(如下表 11)，來做為擬定寵物餵食器大略的構造，如：機構、尺寸等，並透過樂高設計軟體 Bricklink studio 來呈現設計結果(請見圖 2)。

表 16 餵食器功能

項目	功能
1	可定時餵食。
2	可定量餵食。
3	可自動餵食。
4	具防卡設計(即使寵物踩進碗內，正巧碰上轉盤運作時也不會因此受傷)。
5	要能方便清潔。



▲ 圖 2 Bricklink studio(樂高設計軟體)的版面配置

(二) 寵物餵食器的機構設計及方法

1. 餵食器的餵食流程構想

從餵食器將餵食碗的食物給寵物吃，到等寵物吃完和回收餵食碗，都有一套完整的流程，並且針對這些流程和餵食真空碗的構造來設計相對應的機構。而我製作的寵物餵食器也有一套餵食的流程：我將設計一個能盛裝餵食真空碗的儲存匣，接著藉由輸送帶，將餵食真空碗運送到指定位子，再使用開盒機構(分成兩個步驟：解真空和打開蓋子)將餵食碗的蓋子打開並且給寵物吃(請見圖 3)。

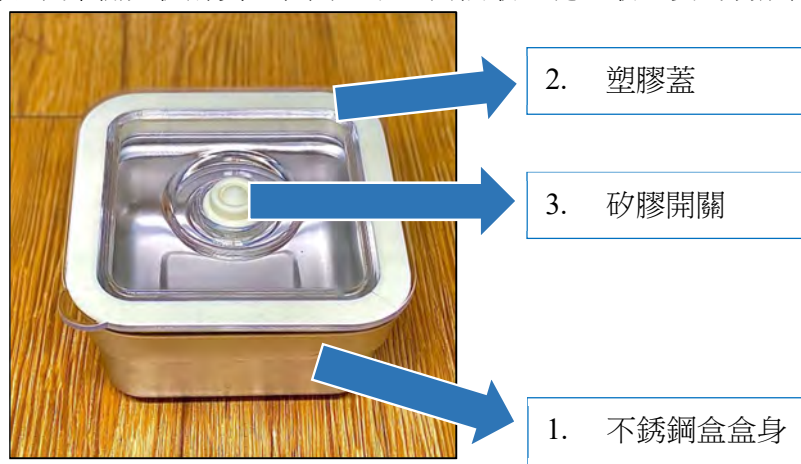


▲ 圖 3 餵食流程

2. 餵食器的初步發想

為了製造出真空的環境來盛裝食物，我採用不銹鋼的真空盒來達成。它主要的構造包括不銹鋼盒盒身(1)、塑膠蓋(2)和可控制真空環境的矽膠開關(3)。不銹鋼盒盒身用來盛裝要餵食的乾糧(若盛裝液狀食物，在進行抽真空時可能會造成真空機損壞)；塑膠蓋的周圍有一圈矽膠膠條，用來阻隔空氣的流通使得維持內部的真空狀態；位於塑膠蓋中的矽膠開關則可調整真空環境：當盒子內為真空狀態時，大氣壓力會將其壓住，必須將開關推開，讓盒子內的空氣和外界空氣流通，達到解真空的效果(請見圖 4)。

根據「1. 餵食器的餵食流程構想」的餵食流程，我將餵食器分成幾個部分，並設計針對餵食真空碗的構造來達成餵食需求。分別有餵食碗儲存匣(用來儲存餵食碗的小型箱子)和運輸台(用來運送餵食碗的檯子)，還有針對餵食碗所設計的開盒機構，而開盒機構，我又將其分成兩個步驟：解真空和開蓋子。以上功能中，因必須先打開蓋子再想辦法利用機構將食物給寵物吃，所以我準備先從解真空和開蓋子，兩個最基礎且最重要的功能開始設計。



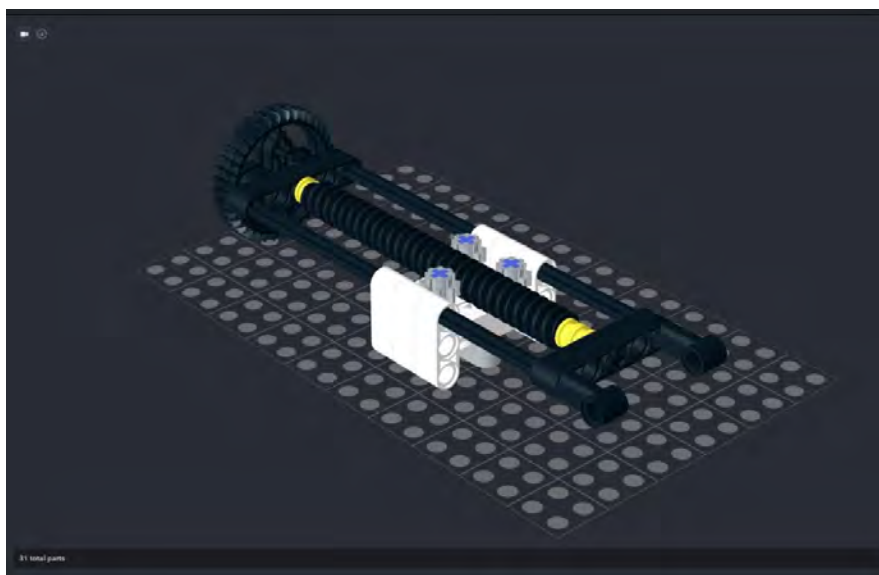
▲ 圖 4 真空盒構造

(三) 餵食器機構的實際製作

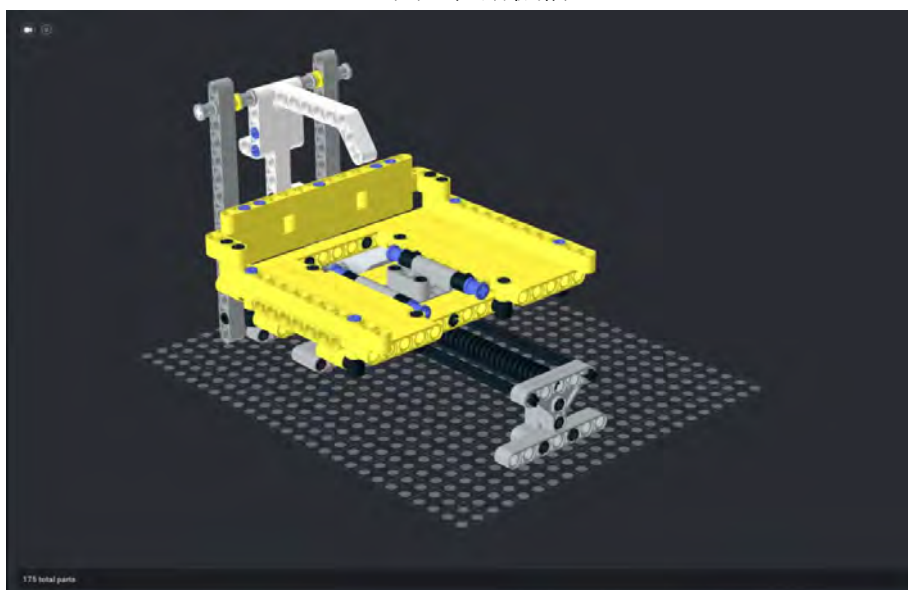
1. 開盒機構的運作模式

設計理念：

根據真空餵食盒的說明，我將設計一根桿子，搭配一個可自由前進和後退的底座，試圖讓餵食盒在移動中碰觸桿子而將矽膠開關頂開，達到解真空的效果；且為了使機構能承擔餵食盒內食物的重量，參考並使用了螺桿來帶動底座上固定不動的齒輪(請見圖 5 和圖 6)。



▲ 圖 5 移動機構



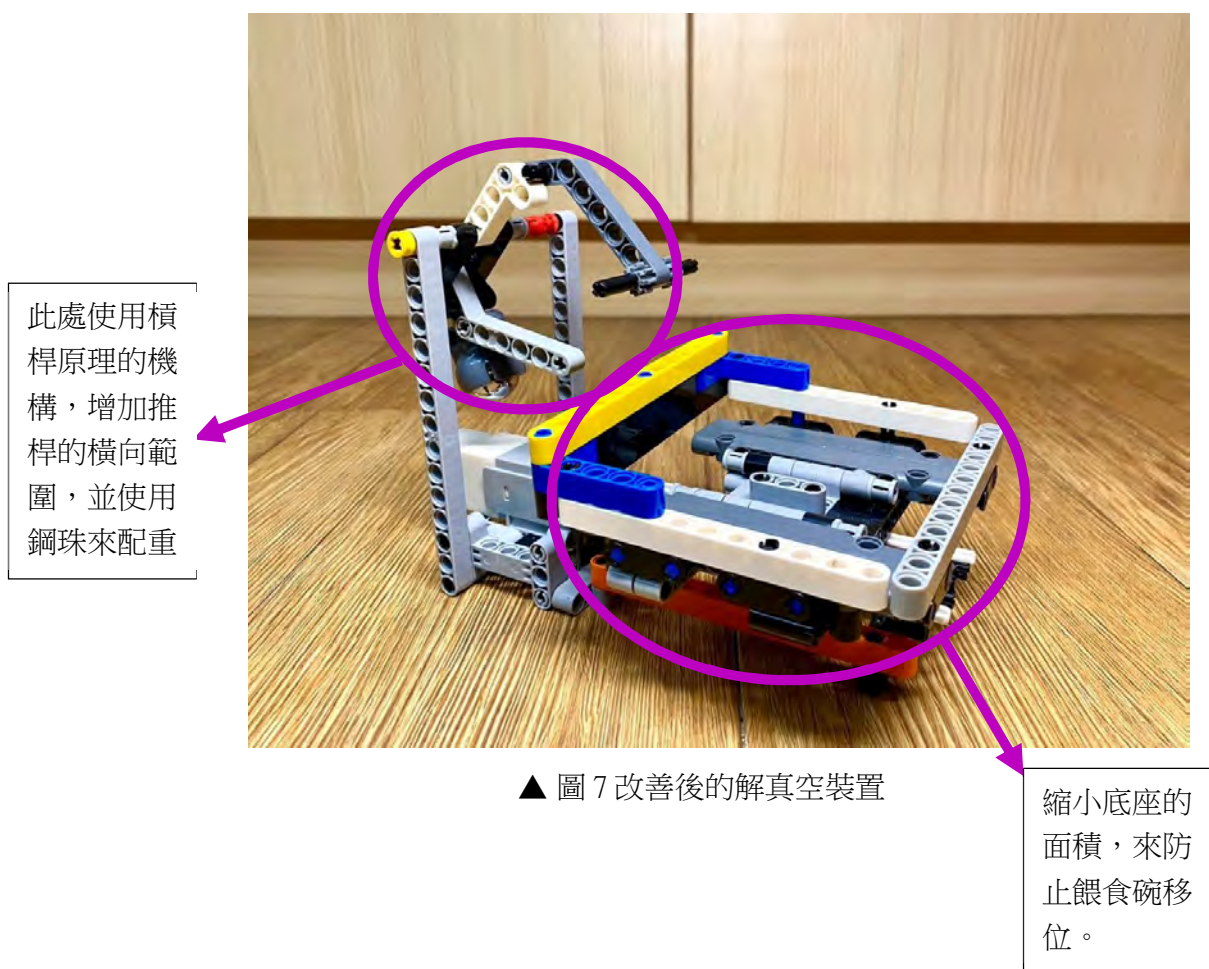
▲ 圖 6 加上底座和桿子後的移動機構

發現問題：

在反覆的測試中，我發現開合機構有三個問題：其一，當底座向後移動時，桿子頂到矽膠開關，其力道不足以將矽膠開關推開，因而導致餵食碗從側邊翻倒、其二，因底座過大導致餵食盒移位、其三，桿子頂不到矽膠開關的正確位子。

解決方法：

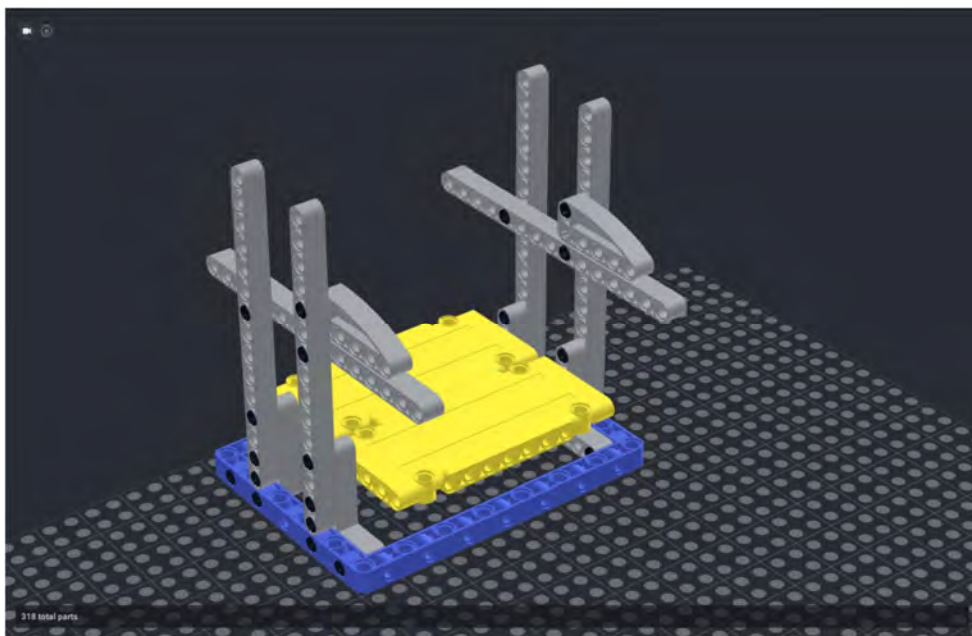
為了防止餵食盒移位，我縮小了底座的面積，並加高外圍的護欄；而為了讓桿子能順利將矽膠開關推開，我採用槓桿原理的方式來解決桿子推動力不足的問題，藉助底座向後移動的動，不僅增加了動力，也增加了伸展的範圍，並在下方放置重物來配重，無須馬達就能自動歸位；至於桿子頂不到矽膠開關的正確位子這個問題，我則透過增加推桿的橫向範圍(請見圖 7)。



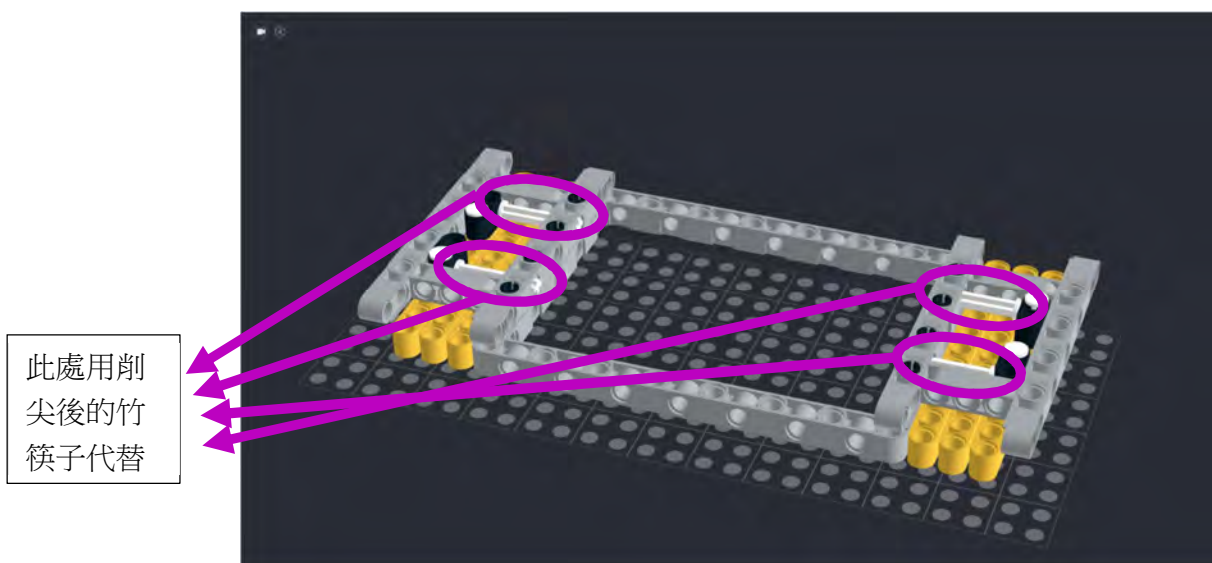
2. 開盒機構的運作模式

設計理念：

測試中發現，餵食盒的周圍有一圈矽膠膠條，與盒緣緊密，且不銹鋼的材質十分光滑，樂高積木不易抓住，因此要打開餵食碗的蓋子十分困難。我原本設計想用底座移動的力摩擦力，透過接觸側邊凸起物，順勢將蓋子「劃」開。但因實際操作後與原本預想得狀況相差甚遠(版本 1，請見圖 8)，所以改為以機械的夾爪為基礎的機構—從兩旁透過尖銳物體向餵食碗與蓋子的縫隙中，但因沒有合適的樂高零件，故採用竹筷子分段後自行磨成尖銳的形狀(版本 2，請見圖 9)。



▲ 圖 8 版本 1 的開盒機構



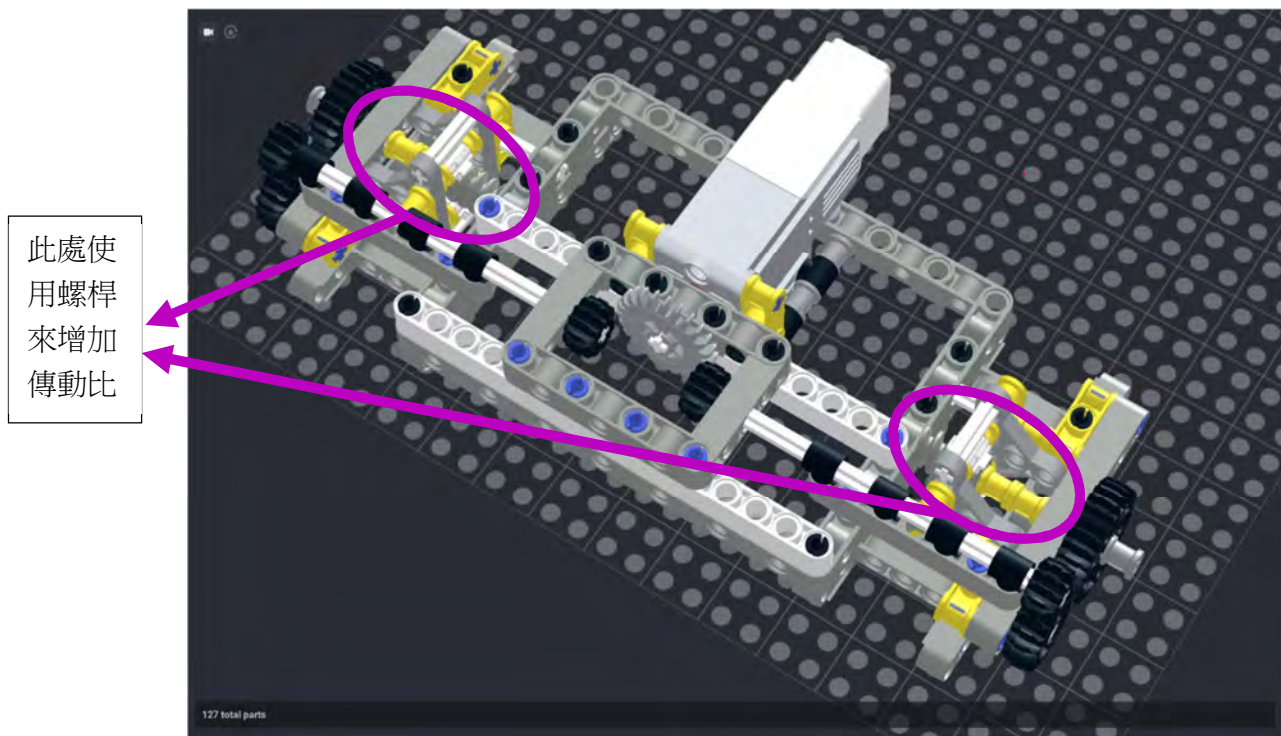
▲ 圖 9 版本 2 的開盒機構

發現問題：

在測試後我發現了以下問題：此裝置中將竹筷子推出的機構和竹筷子是分離的，為了讓機構歸位時竹筷子也能回到原本的位子，我在此裝置中安裝了具彈性性質的橡皮筋。橡皮筋在連接中發現到除了機構自動回彈外，並不具備可操控性，日後在控制時可能會有困難，除此之外因餵食碗與蓋子的縫隙十分緊密，在運行中需要出很大的力才能將其插進去。

解決方法：

我使用螺桿來增加傳動比，並且可以使用馬達精準控制竹筷子後退和前進，同時也可避免因力度不足或其他情況導致機構被回推(請見圖 10)。



▲ 圖 10 改善後的開盒機構

3. 運輸台

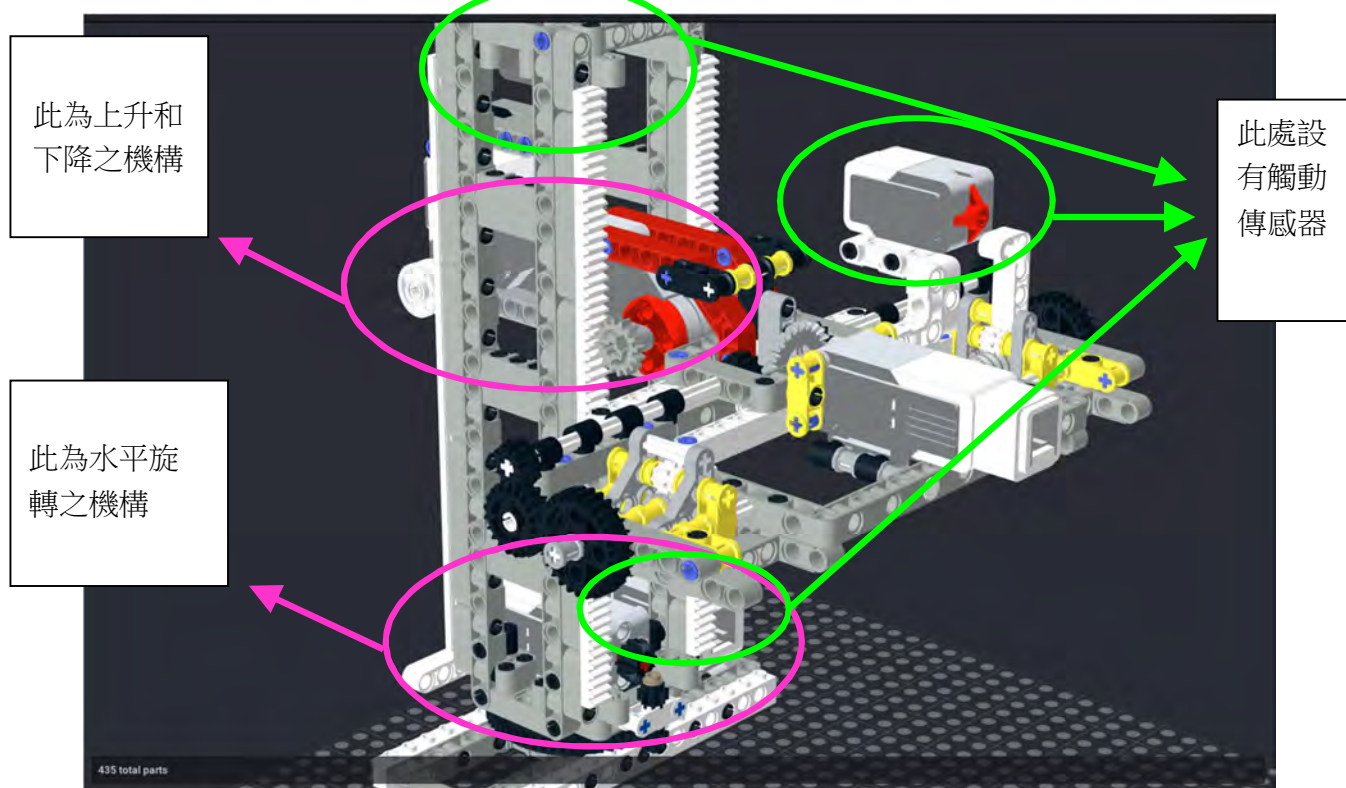
設計理念：

在測試開蓋子機構(改善後的版本)時，我發現此機構能在打開蓋子的同時，也能解除真空。因此我決定捨去原本的解真空裝置，直接使用此裝置同時完成解真空和打開蓋子兩個步驟。除此之外，此機構的外型及運作方式與機械手臂相似，在經過測試後，我發現此機構也可以夾住餵食盒或蓋子等物體，搭配上可上升和下降和水平旋轉的柱子，就可以達到與運輸台相同的功能，成為移動餵食盒的媒介。

我參考現實中的垂直齒輪升降機，它是一種用於定位貨物的設備，通常包括一根直立的柱子，柱子上有一根垂直的齒條，而底盤則通過與齒條相配合的齒輪來上升和下降。這種設計使得裝置能夠沿著柱子的垂直方向進行精確的升降，設計一種能上升和下降的機構，連結著開蓋子機構，就可以實現移動餵食盒的需求(請見圖 11)。

考量到機器在運作時，可能因為斷電或程式等因素，而導致旋轉軸產生誤差。為了確保旋轉軸能保持精準度，我在機器旁加上觸動傳感器，這樣當機器運作到一個角度時，就能觸動傳感器，作為定位的指標(請見圖 11)。

圖 11 機械夾爪

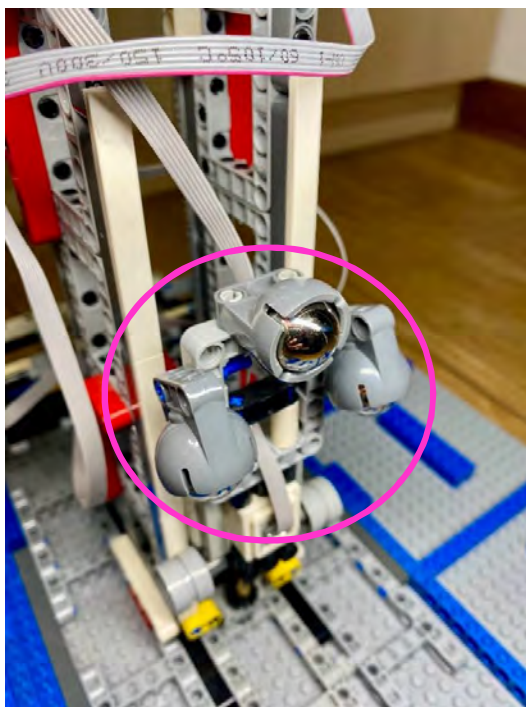


發現問題 1：

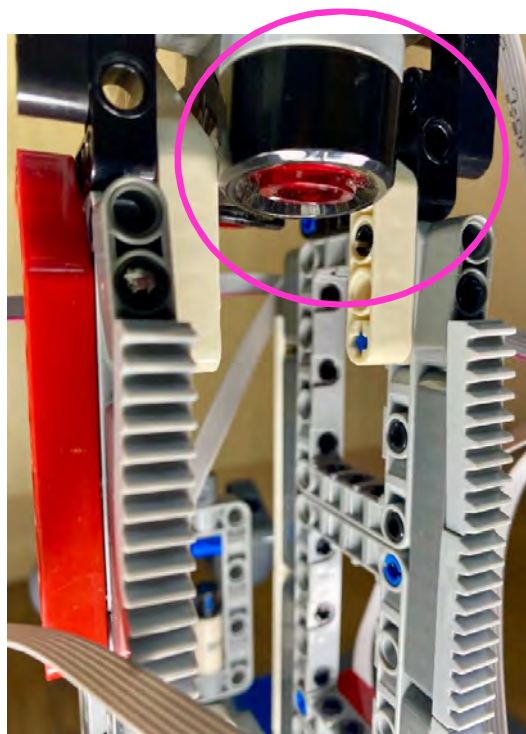
後來在測試時，我發現上升下降、水平轉動和夾爪機構移動的力量不足以推動觸動傳感器的開關，並且因爪子的重量，導致柱子會向一邊傾斜，間接影響柱子的水平轉動(因為會向其中一邊傾斜，所以在旋轉時，離心力會非常大，造成整根柱子搖晃)。

解決方法：

為了解決以上問題，我在控制上升和下降的馬達後面，加設了三顆鋼珠，用來平衡機械夾爪的重量(請見圖 11)。我也根據日後撰寫程式的需要，將上升下降機構的觸動傳感器改為超音波傳感器(超音波感測器是一種數位感測器，透過發出高頻聲波並測量聲音反射回感測器所需的時間，來測量與其前方物體的距離，請見圖 12)，如此一來，就可直接紀錄馬達的定位點及其他位置；至於水平轉動的機構，我則打算讓機器在操作完成後會自動回到原位(無須感測器)，抑或是利用程式的定位功能。



▲ 圖 11 配重裝置



▲ 圖 12 超音波傳感器

發現問題 2：

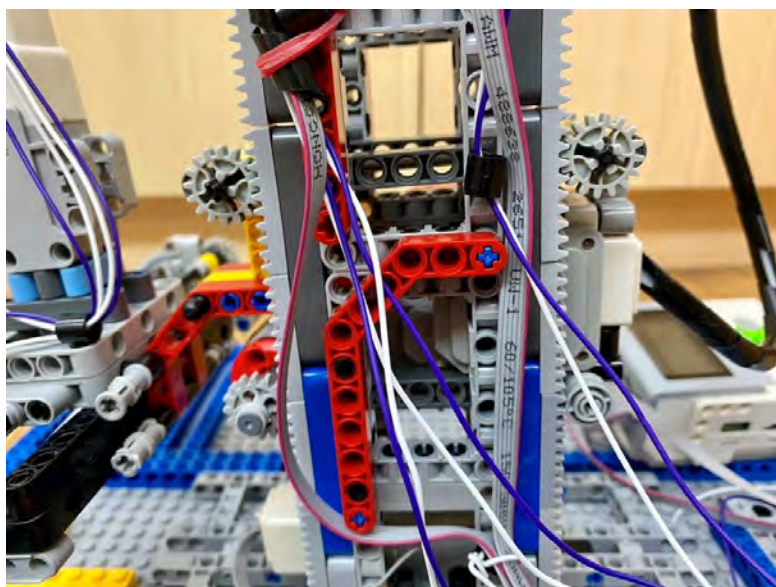
後來在測試時，我發現上升和下降機構因為只有兩個輔助輪，且因不夠緊密而造成在停駐時有些微的晃動，這可能會影響日後下降抓取餵食盒，而水平旋轉的機構在運作時，馬達及底盤的齒輪無法緊密咬合，使其無法旋轉或搖晃。

我也在實際將超音波傳感器連結 EV3 主機後發現，可能是因為周圍空間狹小，反射的超音波不準確，而造成有時雖然機器停在同樣位置，超音波傳感器所採集的數據卻不同的問題，如此一來我就無法依照距離定位。

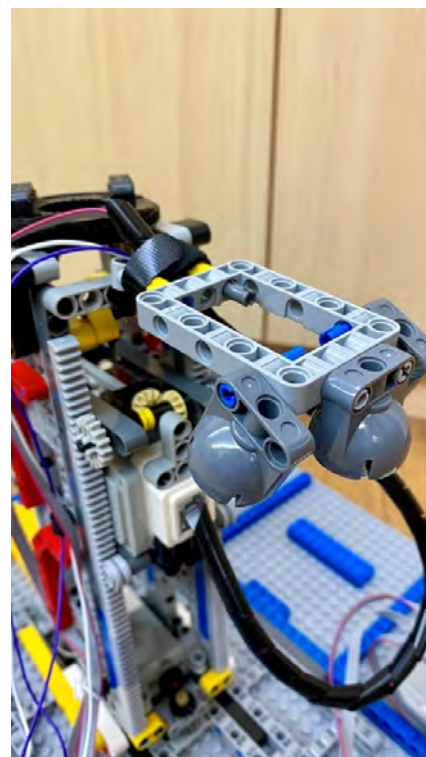
解決方法 2：

為了解決上升和下降機構晃動的問題，我將此改為四個輔助輪，其中三個為齒輪，一個為輪框，請見圖 13；而關於水平旋轉機構搖晃的問題，我加設了三顆鋼珠，並改變了固定和傳動方式，請見圖 14 和 15。

為了解決超音波傳感器數據不準確的問題，我決定外接能以較小力量推動的限位開關作為解決方案(詳細外接方法請見(四)餵食器的程式設計)，並將所有的機構都配上一個限位開關，作為定位的指標，請見圖 16、17、18 和 19。而我則超音波傳感器改為架設於夾爪旁，可用於偵測餵食和是否以夾開並掉落，若沒有則繼續夾直到開啟，請見圖 20。



▲ 圖 13 上升和下降機構

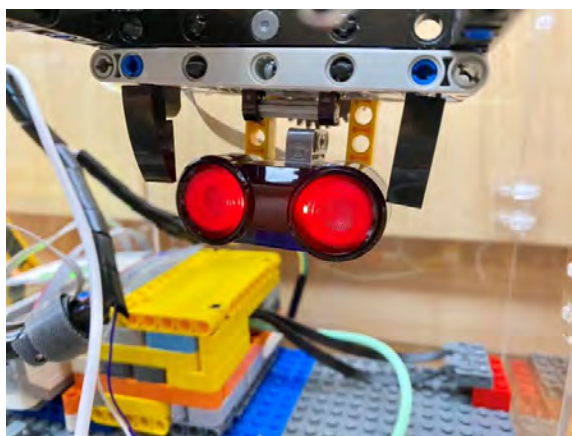


▲ 圖 14 水平旋轉機構

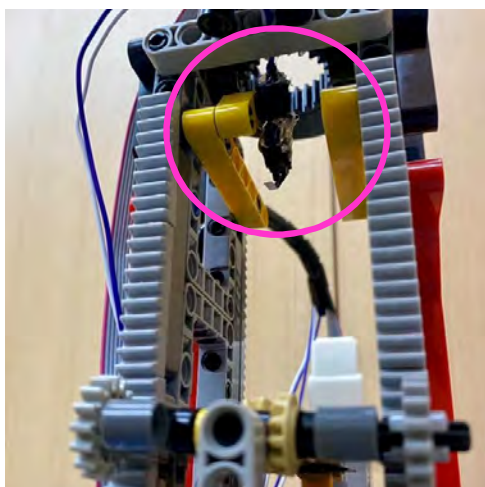
探討大眾對於寵物餵食器之看法與自製寵物餵食器



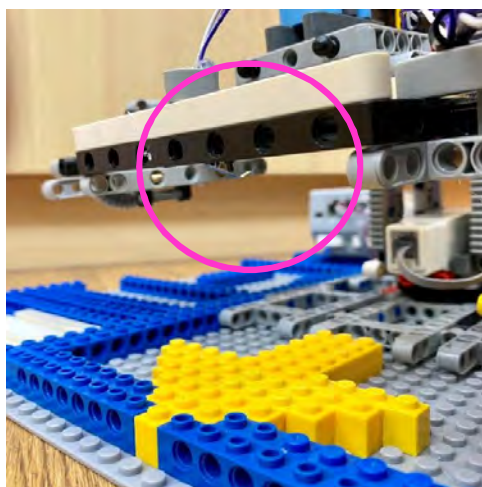
▲ 圖 15 水平旋轉機構的固定和傳動方式



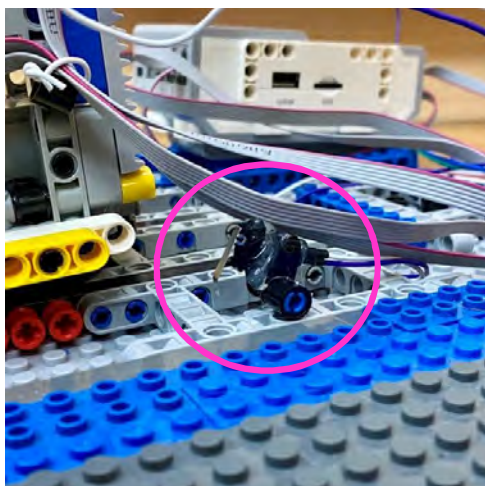
▲ 圖 20 超音波傳感器



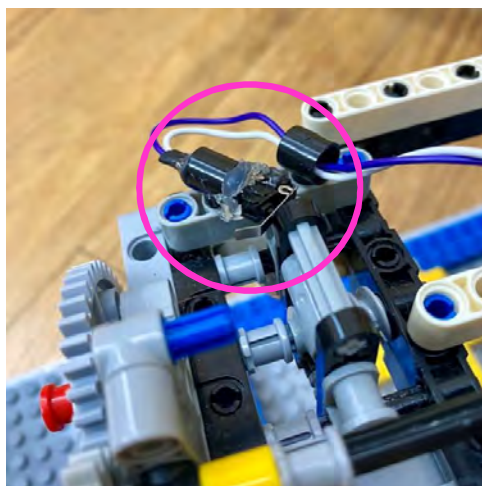
▲ 圖 16 上升和下降機構的限位開關



▲ 圖 17 夾爪機構(檢測餵食盒)的限位開關



▲ 圖 18 水平旋轉機構的限位開關



▲ 圖 19 夾爪機構(檢測夾爪位子)的限位開關

4. 餵食盒儲存匣

設計理念：

開盒夾爪上有限位開關，能自動監測是否觸碰到餵食盒，為了能讓開合夾爪在固定的範圍中夾取餵食盒，我在夾餵食盒處，搭建一個圍欄，並將數個餵食盒堆疊在一個固定的範圍中(請見圖 20)。



▲ 圖 20 餵食盒儲存匣

各項目總結表：

為了使各機構的製作過程(包括：設計理念、發現問題和解決方法)能一目了然，我擷取其中幾項要點，分成第一～二版，並將其彙整成三個表格(如下表 16 和 17)。

表 17 各版本一總結表

第一版			
機構類別項目	設計理念	發現問題	解決方法
解真空	<ol style="list-style-type: none"> 1. 透過螺桿將底座前進後退。 2. 在後方架設桿子，讓餵食盒在移動中打開矽膠開關。 	<ol style="list-style-type: none"> 1. 桿子無法打開矽膠開關。 2. 底座太大，導致餵食盒移動。 	<ol style="list-style-type: none"> 1. 利用槓桿原理設計出無須馬達的力矩機構，增加桿子升縮的範圍。 2. 縮小底座。 3. 加長橫向推桿的長度。
開蓋子	<ol style="list-style-type: none"> 1. 利用底座移動的力，接觸側邊凸起物，順勢將蓋子「劃」開。 	<ol style="list-style-type: none"> 1. 沒有適合的樂高零件可以剛好插進矽膠條中。 2. 底座移動的力不足，且在接觸時餵食盒會翻掉。 	<ol style="list-style-type: none"> 1. 設計其他類型的機構。
運輸台	<ol style="list-style-type: none"> 1. 在設計開蓋子機構時，我發現此機構類似於機械手臂，故採用此機構作為移動餵食盒的媒介。 2. 我參考垂直齒輪升降機的機構，設計一根直立的柱子，柱子上有一根垂直的齒條，而底盤則通過與齒條相配合的齒輪來上升和下降。設計一種能上升和下降的機構，並連結著開蓋子機構。 3. 增加觸動傳感器作為定位的指標。 	<ol style="list-style-type: none"> 1. 上升下降和水平轉動之機構移動的力量不足以推動觸動傳感器的開關。 2. 因爪子的重量，導致柱子會向一邊傾斜，所以在旋轉時，離心力會非常大，造成整根柱子搖晃。 3. 中型馬達有一部分凸出去，轉動範圍太大。 	<ol style="list-style-type: none"> 1. 將上升下降機構的觸動傳感器改為超音波傳感器，並直接紀錄馬達的定位點；而水平轉動的機構，打算讓機器在操作完成後會自動回到原位(無須感測器)，抑或是利用程式的定位功。 2. 在控制上升和下降的馬達後面，加設了三顆鋼珠，用來平衡機械夾爪的重量。 3. 將馬達改為直立式固定。
餵食盒儲存匣	<ol style="list-style-type: none"> 1. 在一定範圍中堆疊餵食盒。 	無	無

表 18 版本二總結表

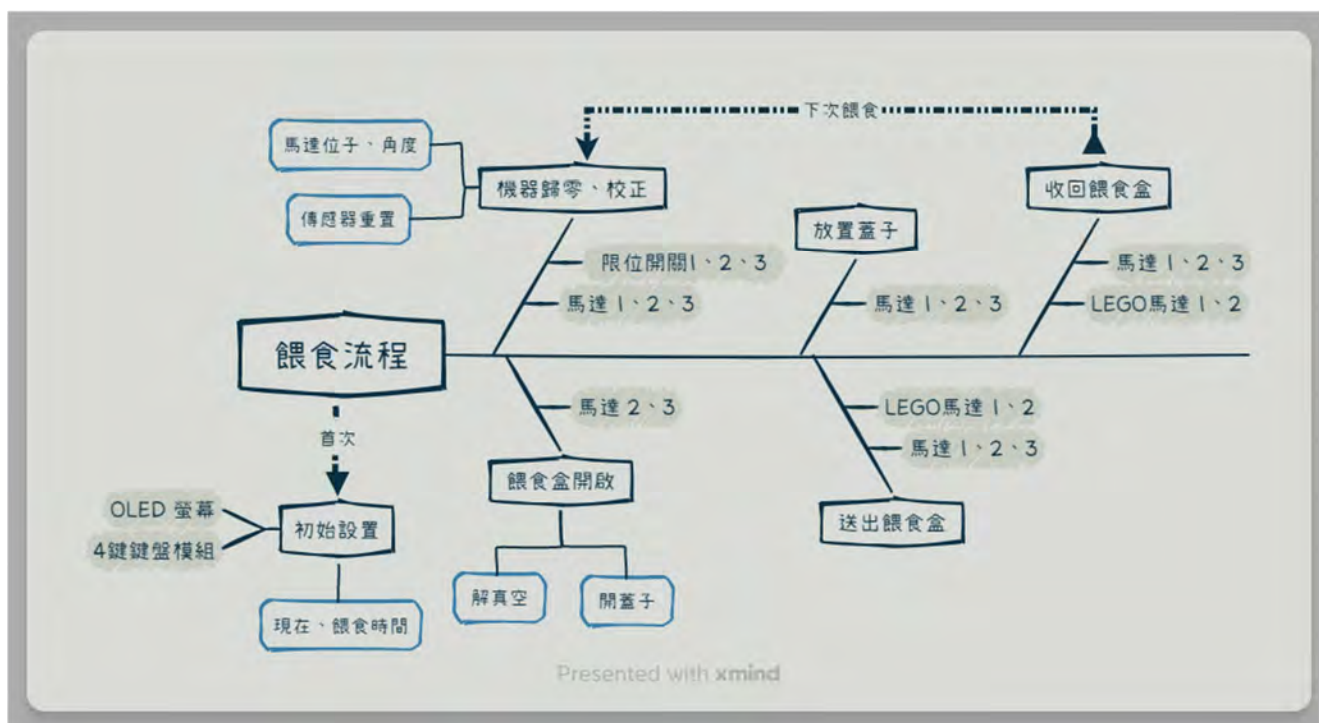
第二版			
機構類別項目	設計理念	發現問題	解決方法
解真空	1. 在設計開蓋子機構時，我發現此機構能解除真空，因此我決定捨去原本的解真空裝置。	無	無
開蓋子	1. 以類似機械夾爪的機構為基礎。 2. 從兩邊用削成尖銳形狀的竹筷子，插進餵食碗與蓋子的縫隙中。	1. 竹筷子無法自動歸位。 2. 力量不夠大。 3. 缺乏可操控性。	1. 加裝橡皮筋 2. 用螺桿來爭加傳動比，同時避免因力度不足等而回彈。
運輸台	無	1. 上升和下降機構有些微晃動。 2. 超音波傳感器反射的超音波不準確。 3. 水平旋轉機構無法轉動。	1. 將上升和下降機構改為四個輔助輪。 2. 外接能以較小力量推動的限位開關。 3. 改變了固定和傳動方式。
餵食盒儲存匣	無	無	無

(四) 餵食器的程式設計

1. 餵食及程式設計流程:

根據「餵食器的餵食流程構想」中提到的餵食流程，我將其分成幾個步驟，分別是從餵食碗儲存匣開始，接著再到輸送帶，經過解真空後，最後打開蓋子並給寵物吃。關於餵食器的程式設計，我將其拆解成更詳細的步驟，以便完成接下來的程式設計。

如下圖，在首次使用餵食器時，須先設定現在時間及各餐的餵食時間。開始餵食時，會先將所有馬達、傳感器歸零，接著解真空和開蓋子，放置蓋子並透過履帶裝置將餵食盒送出，並且待寵物吃完後收回，直到下次的餵食時間到，會重複循環以上步驟(請見圖 21)。



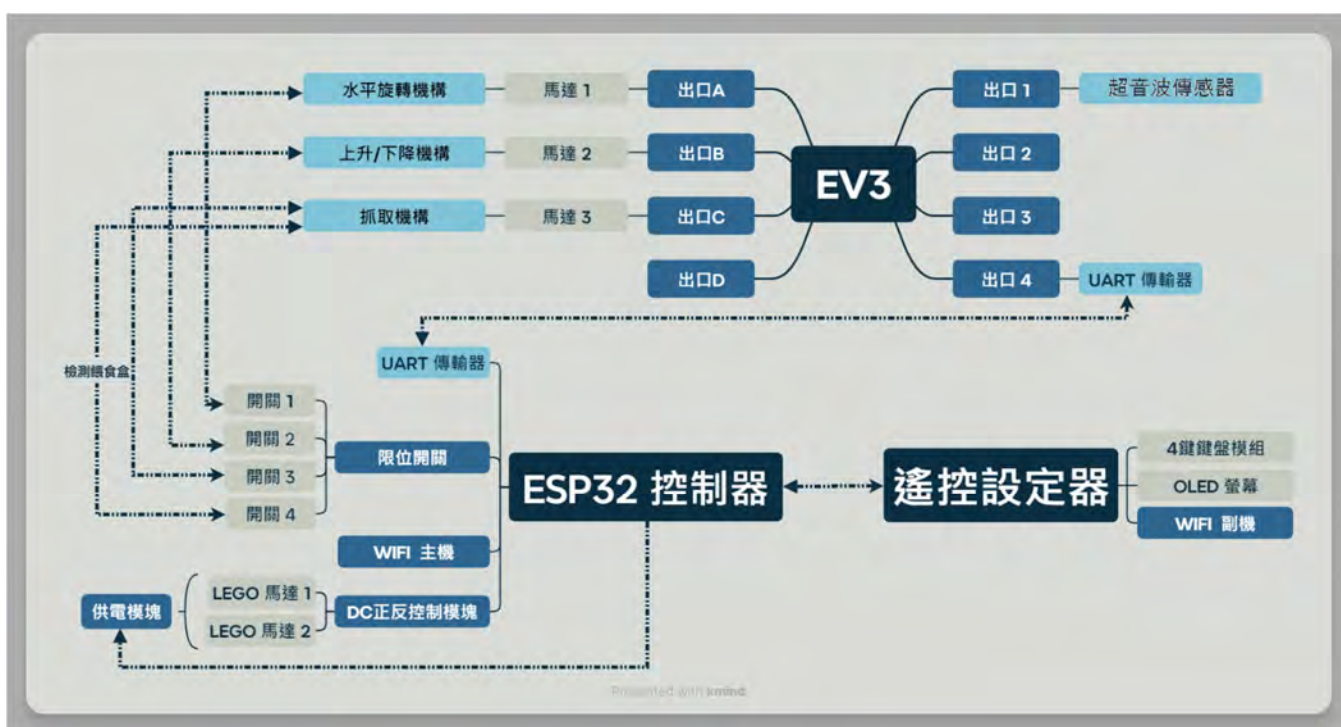
▲ 圖 21 程式設計流程

2.硬體設備配置

EV3 主機共有 8 個出口，最多可同時控制 4 個馬達和 4 個傳感器，其中我使用的 3 顆馬達分別控制水平旋轉機構、上升/下降機構、和抓取機構，而 4 個限位開關則分別用於提供馬達位置和校正馬達。

因為 EV3 主機出口數量的限制，所以我使用 ESP32 微控制器，利用 DC 正反控制模塊，並與供電模塊配合，即可控制一般的樂高馬達，我採用 UART 傳輸器與 EV3 主機對接，使其可互相傳輸和接收數據。

關於遙控設定，我使用了 OLED 螢幕和 4 鍵的鍵盤模塊，利用 WIFI 與 ESP32 微控制器連結，可顯示圖形或文字和進行餵食的相關基礎設置，相關配線請見下圖。



▲ 圖 22 接電圖

3. 程式範例

因篇幅限制，只有擷取較種較種重要的程式碼。

(1) 遙控設定器

載入相關應用的模組：

1. ESP32 內建功能
2. Micro Python 硬體控制模組
3. 計時模組
4. 正反轉模組
5. UART 連接模組
6. WIFI 連接相關模組
7. 正則表達式模組

```

1  #載入相關應用的模組
2  import esp
3  from machine import Pin, SoftI2C, UART
4  from utime import sleep
5  import oled
6  import keyboard
7  import rtc
8  import network, socket, uerrno
    
```

WIFI 連線程式：

16~18. 啟用 WIFI 客戶端接口，將接口設置為啟用狀態

20~31. 不斷檢查 WIFI 客戶端接口是否已連接到 WIFI

，若未連結，則繼續嘗試連接

```

15  #wifi連線程式
16  sta_if = network.WLAN(network.STA_IF)
17  sta_if.active(False)
18  sta_if.active(True)
19  #print(sta_if.scan())
20  while not sta_if.isconnected():
21      print('connecting to network...')
22      try:
23          sta_if.connect(SSID, PASSWORD)
24      except OSError as e:
25          sta_if.disconnect()
26
27      sleep(1)
28      if sta_if.isconnected():
29          break
30      else:
31          sleep(2)
    
```

利用 WIFI 進行通訊：

34~39. 在 WIFI 連接成功後，建立一個 TCP 套接字並連接到指定的遠端伺服器。

```

34  #print('network config:', sta_if.ifconfig())
35  info = sta_if.ifconfig()
36  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
37  #sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
38  sock.connect((str(info[2]), 3000))
39  sock.setblocking(False)
    
```

探討大眾對於寵物餵食器之看法與自製寵物餵食器

處理連線後
傳輸的資料:
45~60. 發送
一個包含日
期時間、頁
面 ID 和
old_key
的字符串

```
45 #處理連線後傳輸的資料
46 def socket_handle():
47     global old_key
48     #sock.send("this is mydata")
49     try:
50         send_str = str(oled.get_datetime()) + ";" + str(oled.get_pageid()) + ";" + str(str(old_key))
51         #print(send_str)
52         sock.settimeout(0)
53         sock.send(send_str.encode('utf-8'))
54         rec = sock.recv(128)
55         result = rec.decode('utf-8').split(';')
56     except Exception as e:
57         if e.args[0] != uerrno.EAGAIN:
58             print(e)
59         pass
60     #print(result)
```

偵測按鍵觸碰情形：
63~77. 若遠端設備
的藍鍵是否被按下
(0:無按鍵 1:紅 2:黃
3:綠 4:藍)，顯示現
在時間(每秒刷新)

```
63 while True:
64     key = keyboard.scan()
65     # 0:無按鍵 4:藍 3:綠 2:黃 1:紅
66     if key == 4 and old_key == 0:
67         oled.switch_page()
68         old_key = key
69
70     timer_counter = timer_counter + 1
71     if timer_counter >= 10:
72         oled.show_datetime()
73         timer_counter = 0
74
75     socket_handle()
76
77     sleep(0.1)
```

關於遙控設定器其他頁面的顯示程式，我寫在副程式 OLED 中。因此綜合以上，我共編寫了 5 個頁面，也針對 4 個按鍵分別設定不同的功能，請見以下範例和螢幕顯示結果(擷取部分程式碼，請見表 19 和 20)：

表 19 四個按鍵功能

按鍵顏色	功能	範例程式碼
紅	執行/停止 (在時間設定頁面可用於選擇時間)	<pre> 14 if abs(val - 3.15) <= 0.03 : 15 return 1 </pre>
黃	切換選項 (時間設定頁面在設定時間時可用於增加時間)	<pre> 16 elif abs(val - 2.59) <= 0.03: 17 return 2 </pre>
綠	切換選項 (時間設定頁面在設定時間時可用於減少時間)	<pre> 18 elif abs(val - 2.29) <= 0.03 : 19 return 3 </pre>
藍	切換頁面	<pre> 20 elif abs(val - 2.03) <= 0.03: 21 return 4 </pre>

表 20 五個螢幕頁面

項目	螢幕頁面	範例程式碼
1		<pre> 344 if id == 0: 345 write_font(display, '寵物真空餵食機', 0, 16) 346 #write_font(display, '歸零 測試 運轉', 20, 32) 347 write_font(display, '黃/綠:選項 紅:開/停', 0, 48) </pre>
2		<pre> 348 if id == 1: 349 write_font(display, '時間設定', 0, 16) 350 write_font(display, '設定一', 0, 32) 351 write_font(display, '設定二', 0, 48) 352 write_font(display, ':', 68, 32) 353 write_font(display, ':', 68, 48) 354 write_font(display, "{:02d}".format(set1_hour), 48, 32, False) 355 write_font(display, "{:02d}".format(set1_min), 77, 32, False) 356 write_font(display, "{:02d}".format(set2_hour), 48, 48, False) 357 write_font(display, "{:02d}".format(set2_min), 77, 48, False) </pre>
3		<pre> 358 if id == 2: 359 write_font(display, '動作測試', 0, 16) 360 write_font(display, '旋轉馬達', 0, 32) 361 write_font(display, '綠:右轉 黃:左轉', 0, 48) </pre>

4	<p>2024/03/31 17:40:42 動作測試 二 升降馬達 0000 0 綠上升 黃下降</p>	<pre> 362 if id == 3: 363 write_font(display, '動作測試 □', 0, 16) 364 write_font(display, '升降馬達', 0, 32) 365 write_font(display, '綠:上升 黃:下降', 0, 48) </pre>
5	<p>2024/03/31 17:41:28 動作測試 三 抓取馬達 0000 0 1 綠:抓取 黃:放鬆</p>	<pre> 366 if id == 4: 367 write_font(display, '動作測試 ▢', 0, 16) 368 write_font(display, '抓取馬達', 0, 32) 369 write_font(display, '綠:抓取 黃:放鬆', 0, 48) </pre>

(2) ESP32 微控制器

- 載入相關應用的模組：
1. ESP32 內建功能
 2. Micro Python 硬體控制模組
 3. 計時模組
 4. 正反轉模組
 5. UART 連接模組
 6. WIFI 連接相關模組
 7. 正則表達式模組

```

1  import esp
2  from machine import Pin, UART
3  from utime import sleep
4  import motor
5  import port
6  import network, socket, select
7  import re
                
```

- 硬體初始化與變數宣告：
- 9~14. 初始化硬體引腳
 - 15~16. 關閉直流電源和 LED
 - 17~21. 宣告全局變數

```

9  Limit1=Pin(10, Pin.IN, Pin.PULL_UP)
10 Limit2=Pin(9, Pin.IN, Pin.PULL_UP)
11 Limit3=Pin(1, Pin.IN, Pin.PULL_UP)
12 Limit4=Pin(2, Pin.IN, Pin.PULL_UP)
13 DcPowerOn = Pin(7, Pin.OUT, Pin.PULL_DOWN)
14 LED = Pin(8, Pin.OUT, Pin.PULL_DOWN)
15 DcPowerOn.off()
16 LED.off()
17 counter = 0
18 pageid = 0
19 keyin = 0
20 timeup = 0
21 keyinTimeout = 0
                
```

- WIFI 設定與 TCP 伺服器初始化：
- 22~25. 初始化 WIFI 存取點 (AP)
 27. 獲取 AP 的 IP 地址
 - 28~32. 初始化 TCP 伺服器套接字

```

22 ap = network.WLAN(network.AP_IF)
23 ap.config(essid=b'MyESP32AP')
24 ap.active(False)
25 ap.active(True)
26
27 hostip = ap.ifconfig()[0]
28 tcp_server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
29 tcp_server.bind((hostip, 3000)) #server ip and port
30 tcp_server.setblocking(False) # non - blocking
31 tcp_server.listen(1) #listen only 1 connection
32 inputs = [tcp_server]
                
```

TCP 伺服器操作與
客戶端連線處理：
39~40. 使用 *select*
監聽客戶端連線
41~43. 可讀事件處理
44~48. 接受客戶端連線
49~53. 處理客戶端數據

```
34 def tcp_server_do():
35     global pageid
36     global keyin
37     global timeup
38     global keyinTimeout
39     readable, _, err = select.select(inputs, [], inputs, 0.1)
40     keyinTimeout = keyinTimeout + 1
41     if readable:
42         for sck in readable:
43             if sck is tcp_server:
44                 client, addr = sck.accept()
45                 client.setblocking(False)
46                 client.settimeout(0)
47                 inputs.append(client)
48                 print(client)
49             else:
50                 req = sck.recv(128)
51                 if len(req) == 0:
52                     inputs.remove(sck)
53                 sck.close()
```

TCP 伺服器操作與
客戶端連線處理：
54~73. 解析收到的數據

```
54     else:
55         pattern = r'HH(.*)TT'
56         data_str = req.decode('utf-8')
57         match = re.search(pattern, data_str)
58         if not match:
59             return
60
61         matched_string = match.group(1)
62         result = matched_string.split(';')
63
64         if len(result) >= 3:
65             try:
66                 pageid = int(result[1])
67                 timeup = int(result[2])
68                 keyin = int(result[3])
69                 keyinTimeout = 0
70             except ValueError:
71                 pageid = 0
72                 timeup = 0
73                 keyin = 0
```

TCP 伺服器操作與
客戶端連線處理：
74~75. 發送響應數據
77~80. 處理錯誤事件
82~84. 超時處理

```
74         send_str = "HH" + str(port.motor1Pos) + "  
75         sck.send(send_str.encode('utf-8'))  
76  
77         for sck in err:  
78             print('socket error')  
79             inputs.remove(sck)  
80             sck.close()  
81  
82         if keyinTimeout >= 5:  
83             keyin = 0  
84             pageid = 0
```

迴圈與硬體控制：
86. DC 正反控制模
塊打開
88~102. 與EV3 通
訊並接受回授訊號
(限位開關)

```
86     DcPowerOn.on()  
87  
88     while True:  
89         tcp_server_do()  
90         data = port.read_command()  
91         counter = counter + 1  
92         if counter >= 5:  
93             counter = 0  
94             print("motor1 : ", port.motor1Pos, ", L2 : ", Limit2.value())  
95             print("motor2 : ", port.motor2Pos, ", L3 : ", Limit3.value())  
96             print("motor3 : ", port.motor3Pos, ", L1 : ", Limit1.value())  
97             print("L4 : ", Limit4.value(), "PageID : ", pageid, "TimeUp", timeup, "KeyIN : ", keyin)  
98             if LED.value() > 0:  
99                 LED.off()  
100            else:  
101                LED.on()  
102            port.write_command(Limit1.value(), Limit2.value(), Limit3.value(), Limit4.value(), keyin, pageid)  
103
```

(3) EV3 與馬達相關控制

載入相關應用的模組：

- 2. EV3 主機模組
- 3. 馬達模組
- 4. 出口模組
- 5. 等待模組
- 6. 按鍵模組
- 7. UART 連接模組
- 8. 超音波感測器模組
- 9. 正則表達式模組

```
1  #!/usr/bin/env pybricks-micropython
2  from pybricks.hubs import EV3Brick
3  from pybricks.ev3devices import Motor
4  from pybricks.parameters import Port, Stop
5  from pybricks.tools import wait
6  from pybricks.parameters import Button
7  from pybricks.iodevices import UARTDevice
8  from pybricks.ev3devices import UltrasonicSensor
9  import re
```

硬體初始化與變數宣告：

- 12. 初始化EV3 主機
- 14~29. 宣告全局變數

```
11  # Initialize the EV3 Brick.
12  ev3 = EV3Brick()
13
14  Limit1 = 0
15  Limit2 = 0
16  Limit3 = 0
17  Limit4 = 0
18  KB = 0
19  oldKB = KB
20  PID = 0
21  RunMode = 0
22  homeset_step = 4
23  motor1 = Motor(Port.B)
24  motor2 = Motor(Port.C)
25  motor3 = Motor(Port.A)
26  initMotors = False
27  connected = False
28  link_timer = 0
29  run_tep = 20
```

通訊初始化：

- 34. 初始化 UART 通信設備，並指定Port.S1

```
33  # 開啟EV3的Port 1與外設控制器通訊
34  ser = UARTDevice(Port.S1, baudrate=115200)
```

連接時間及情況：
37~56. 檢查連線是否超時。如果連線超時，則將重設變數，並將連線狀態設置為未連線。

```
37 def timeout_check():
38     global link_timer
39     global connected
40     global Limit1
41     global Limit2
42     global Limit3
43     global Limit4
44     global KB
45     global PID
46     global RunMode
47
48     if link_timer >= 5:
49         Limit1 = 0
50         Limit2 = 0
51         Limit3 = 0
52         Limit4 = 0
53         KB = 0
54         PID = 0
55         RunMode = 0
56         connected = False
```

連接時間及情況：
59~82. 從 UART 通信設備中讀取命令，檢查是否有數據可供讀取，並讀取所有可用的數據，將計時器加一，以便後續的超時檢查。

```
59 def read_command():
60     global Limit1
61     global Limit2
62     global Limit3
63     global Limit4
64     global KB
65     global PID
66     global RunMode
67     global Motor1_FUN
68     global Motor2_FUN
69     global Motor3_FUN
70     global link_timer
71     global connected
72     link_timer = link_timer + 1
73     timeout_check()
74
75     if not ser.waiting():
76         return
77     data = b''
78     while True:
79         data = data + ser.read_all()
80         wait(10)
81         if not ser.waiting():
82             break
```

解碼接收數據：
 84~86. 嘗試解碼接收到的數據並進行正則匹配，以擷取所需的若信息。出現 *UnicodeError*，則將重設變數。

```

84     try:
85         pattern = r'HH(.*)TT'
86         data_str = data.decode('utf-8')
87         match = re.search(pattern, data_str)
88     except UnicodeError:
89         Limit1 = 0
90         Limit2 = 0
91         Limit3 = 0
92         Limit4 = 0
93         KB = 0
94         PID = 0
95         return
    
```

解碼接收數據：
 97~124. 若成功匹配，將連線計時器重設，並將連線狀態設置為已連線。解析接收到的數據，將其轉換類型並分配給對應的變數。如果轉換失敗，則將一些變數重設為 0。

```

97     if not match:
98         return
99
100    matched_string = match.group(1)
101    result = matched_string.split(';')
102
103    if len(result) < 6:
104        return
105    # print(result)
106    link_timer = 0
107    connected = True
108
109    try:
110        Limit1 = int(result[1])
111        Limit2 = int(result[0])
112        Limit3 = int(result[2])
113        Limit4 = int(result[3])
114        KB = int(result[4])
115        PID = int(result[5])
116        RunMode = int(result[6])
117    except ValueError:
118        Limit1 = 0
119        Limit2 = 0
120        Limit3 = 0
121        Limit4 = 0
122        KB = 0
123        PID = 0
124        RunMode = 0
    
```

馬達角度校正：
127~134. 將馬達角度信息發送給控制器。获取每個馬達的角度，格式化為字符串後，發送到控制器。

```
127 def write_command():
128     ma1 = int(motor1.angle())
129     ma1_str = "{:+06d}".format(ma1)
130     ma2 = int(motor2.angle())
131     ma2_str = "{:+06d}".format(ma2)
132     ma3 = int(motor3.angle())
133     ma3_str = "{:+06d}".format(ma3)
134     ser.write("HH" + ma1_str + ";" + ma2_str + ";" + ma3_str + "TT")
```

機器運行步驟設定：
137~148. 設定一個變數 --run_tep，此變數的各數字對應一個動作，每做完一步就增加 1。

```
137 def machine_run(auto):
138     global run_tep
139     global Limit1
140     global Limit2
141     global Limit3
142     global Limit4
143
144     if run_tep > 20:
145         motor1.stop()
146         motor2.stop()
147         motor3.stop()
148         return
```

餵食動作：(詳細動作請見黃框中的綠字)

```
150     if run_tep == 0: # 鬆爪
151         if Limit3 == 0:
152             motor3.stop()
153             wait(1000)
154             motor3.reset_angle(0)
155             run_tep = run_tep + 1
156         else:
157             motor3.run(-400)
158     elif run_tep == 1: # 上升到頂
159         if Limit2 == 0:
160             motor2.stop()
161             wait(1000)
162             motor2.reset_angle(0)
163             run_tep = run_tep + 1
164         else:
165             motor2.run(-200)
166     elif run_tep == 2: # 到進貓裡區
167         if Limit1 == 0:
168             motor1.stop()
169             wait(1000)
170             motor1.reset_angle(0)
171             run_tep = run_tep + 1
172         else:
173             motor1.run(500)
```

餵食動作：
(詳細動作
請見黃框
中的綠字)

```

174 elif run_tep == 3: # 下降碰到貓糧盒
175     if Limit4 == 0 or motor2.angle() > 517:
176         motor2.stop()
177         wait(1000)
178         run_tep = run_tep + 1
179     else:
180         motor2.run(100)
181 elif run_tep == 4: # 抓緊貓糧盒
182     if Limit4 == 1 or motor3.angle() > 1100:
183         motor3.stop()
184         run_tep = run_tep + 1
185     else:
186         motor3.run(400)
187 elif run_tep == 5: # 上升到頂
188     if Limit2 == 0:
189         motor2.stop()
190         run_tep = run_tep + 1
191     else:
192         motor2.run_target(200, 0, Stop.COAST, False)
193 elif run_tep == 6: # 剎出貓糧區
194     if abs(motor1.angle() + 800) <= 20:
195         motor1.stop()
196         run_tep = run_tep + 1
197     else:
198         motor1.run_target(500, -800, Stop.COAST, False)
199 elif run_tep == 7: # 下降放下貓糧盒
200     if abs(motor2.angle() - 517) <= 20:
201         motor2.stop()
202         run_tep = run_tep + 1
203     else:
204         motor2.run_target(100, 517, Stop.COAST, False)
205 elif run_tep == 8: # 打開貓糧盒
206     if abs(motor3.angle() - 2200) <= 20:
207         motor3.stop()
208         run_tep = run_tep + 1
209     else:
210         motor3.run_target(400, 2200, Stop.COAST, False)
211 elif run_tep == 9: # 上升到頂
212     if Limit2 == 0:
213         motor2.stop()
214         run_tep = run_tep + 1
215     else:
216         motor2.run_target(200, 0, Stop.COAST, False)

```

探討大眾對於寵物餵食器之看法與自製寵物餵食器

餵食動作：
(詳細動作
請見黃框
中的綠字)

```
217  ▾ elif run_tep == 10: # 到貓糧盒蓋回收區
218  ▾     if abs(motor1.angle() + 1630) <= 20:
219  ▾         motor1.stop()
220  ▾         run_tep = run_tep + 1
221  ▾     else:
222  ▾         motor1.run_target(500, -1630, Stop.COAST, False)
223  ▾ elif run_tep == 11: # 鬆開貓糧盒蓋
224  ▾     if Limit3 == 0:
225  ▾         motor3.stop()
226  ▾         run_tep = run_tep + 1
227  ▾     else:
228  ▾         motor3.run_target(400, 0, Stop.COAST, False)
229  ▾ elif run_tep == 12: # 結束餵食
230  ▾     run_tep = run_tep + 1
231  ▾     ev3.speaker.beep()
```

馬達校正：
234~274。
用於對馬達歸
零操作。根據
homeset_step
變量的值，逐
步執行不同機
構的歸零動作。

```

234 def motor_homeset():
235     global PID
236     global KB
237     global RunMode
238     global Limit1
239     global Limit2
240     global Limit3
241     global Limit4
242     global homeset_step
243     global initMotors
244
245     if PID != 0 or homeset_step > 3:
246         motor1.stop()
247         motor2.stop()
248         motor3.stop()
249         return
250
251     if homeset_step == 0:
252         if Limit3 == 1:
253             motor3.run(-400)
254         else:
255             motor3.stop()
256             motor3.reset_angle(0)
257             homeset_step = homeset_step + 1
258     elif homeset_step == 1:
259         if Limit2 == 1:
260             motor2.run(-200)
261         else:
262             motor2.stop()
263             motor2.reset_angle(0)
264             homeset_step = homeset_step + 1
265     elif homeset_step == 2:
266         if Limit1 == 1:
267             motor1.run(500)
268         else:
269             motor1.stop()
270             motor1.reset_angle(0)
271             homeset_step = homeset_step + 1
272     elif homeset_step == 3:
273         homeset_step = homeset_step + 1
274         initMotors = True

```

手動操作：
282~325。
對馬達進行手
動操作。根據
PID 和 *KB*變量
的值，來決定
是對旋轉馬達
、升降馬達還
是抓取馬達進
行哪種操作，
例如旋轉、上
升、下降、抓
取或放鬆。

```

282 def motor_jog():
283     global PID
284     global KB
285     global Limit1
286     global Limit2
287     global Limit3
288     global Limit4
289     global initMotors
290
291     if PID == 2: # 2: 旋轉馬達
292         if KB == 3: # 右璇
293             if motor1.angle() < -1630 and initMotors:
294                 motor1.stop()
295             else:
296                 motor1.run(-500)
297         elif KB == 2: # 左璇
298             if Limit1 == 0:
299                 motor1.stop()
300             else:
301                 motor1.run(500)
302         else:
303             motor1.stop()
304
305         motor2.stop()
306         motor3.stop()
307         return
308
309     if PID == 3: # 3: 升降馬達
310         if KB == 3: # 上升
311             if Limit2 == 0:
312                 motor2.stop()
313             else:
314                 motor2.run(-200)
315         elif KB == 2: # 下降
316             if motor2.angle() > 517 and initMotors:
317                 motor2.stop()
318             else:
319                 motor2.run(200)
320         else:
321             motor2.stop()
322
323         motor1.stop()
324         motor3.stop()
325         return
    
```

手動操作：
282~325。
對馬達進行手動操作。根據PID和KB變量的值，來決定是對旋轉馬達、升降馬達還是抓取馬達進行哪種操作，例如旋轉、上升、下降、抓取或放鬆。

```

327     if PID == 4: # 4: 抓取馬達
328         if KB == 3: # 抓取
329             if motor3.angle() > 2200 and initMotors:
330                 motor3.stop()
331             else:
332                 motor3.run(400)
333         elif KB == 2: # 放鬆
334             if Limit3 == 0:
335                 motor3.stop()
336             else:
337                 motor3.run(-400)
338         else:
339             motor3.stop()
340
341     motor1.stop()
342     motor2.stop()
343     return
    
```

歸零、測試和自動運行：
349~374。
控制機器的運行模式。根據PID和RunMode變量的值，決定是進行的模式

```

349 def machine_ctrl():
350     global PID
351     global KB
352     global oldKB
353     global RunMode
354     global homeset_step
355     global initMotors
356     global run_tep
357     if PID == 0: # 自動模式
358         if RunMode == 0: # 歸零模式
359             motor_homeset()
360             if oldKB != KB and KB == 1: # 按下開始鍵(紅鍵), 進行歸零
361                 homeset_step = 0
362
363         if RunMode == 1: # 測試模式
364             machine_run(False)
365             if oldKB != KB and KB == 1: # 按下開始鍵(紅鍵), 進行測試
366                 run_tep = 0
367
368         if RunMode == 2: # 運行模式
369             machine_run(True)
370             if oldKB != KB and KB == 1: # 按下開始鍵(紅鍵), 進行全自
371                 run_tep = 0
372
373     elif PID >= 2 and PID <= 4: # 手動模式 2: 旋轉馬達 3: 升降馬達 4:
374         motor_jog()
    
```

待機狀態：
用read_command()
讀取命令，用write_command()
來發送命令，若已連接，執機器控制函數machine_ctrl()

```

377 while True:
378     read_command()
379     write_command()
380     if connected:
381         machine_ctrl()
382
383     wait(100)
    
```

4.設備外殼設計及線路安裝

在設計機構、組裝、撰寫程式和測試後，確保餵食動作能正常運行後，要為主體製作外殼，同時也能隔絕灰塵和外界的干擾。關於外殼的材質，我選擇壓克力，在網上訂製後，組裝成一個罩子，並連結底板，請見圖 23 和 24。

為了使餵食過程更加順利，我在左右兩側各開了一個小門，用於放置餵食盒和取出餵食盒的蓋子。並在正前方搭建斜坡和門，使餵食盒掉落後會順勢滑出，而門則可防止寵物破壞餵食器，請見圖 25、26 和 27。關於遙控器和其他設備，我用樂高積木為其製作外殼，請見圖 28 和 29。

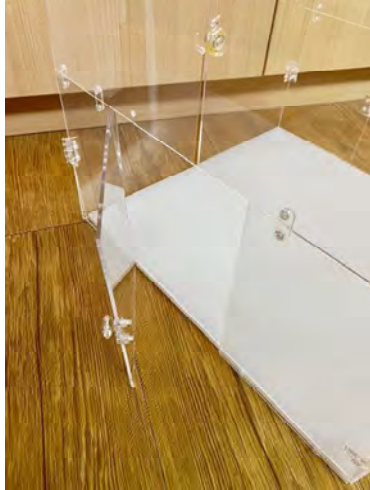


▲ 圖 23 壓克力罩子的平面圖



▲ 圖 24 壓克力罩子的完成圖

探討大眾對於寵物餵食器之看法與自製寵物餵食器



▲ 圖 25 側門 1(餵食盒入口)



▲ 圖 26 側門 2(蓋子出口)



▲ 圖 27 正門及斜坡



▲ 圖 28 遠端遙控設定器的外殼



▲ 圖 29ESP32 的外殼

伍、研究結果與討論

一、寵物餵食器的設計標準根據之來源

一開始，我先挑選四種市售常見的寵物餵食器整理出不同的產品功能，包括：餵食方式、食物存放、外觀設計、安全性與保護、供電方式和其他功能六項類別，發現市售普遍的餵食器多體現於食物存放，並經由 SWOT 分析，瞭解並評估寵物餵食器在市場中的競爭程度，以便更好的確立製作寵物餵食器的規劃方向，讓我製作出的餵食器能符合大眾的需要。

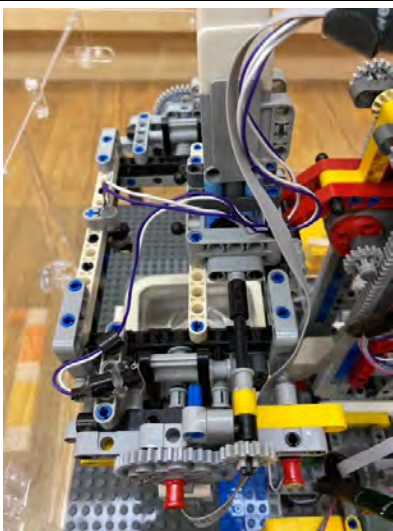
接著，根據以上的產品功能，製作餵食器調查問卷，瞭解大眾對於寵物餵食器的看法和需求，後製作標準表，確定製作寵物餵食器的方向及功能。

二、寵物餵食器的整體及功能展示

我已根據「三、寵物餵食器原型製作」中的標準表，製作出寵物餵食器的原型。接下來，我將針對機構設計、實際功能及操作方式這三個類別介紹我的餵食器，另外也附上實際操作影片的 QRcode 和網址。

三、機構設計

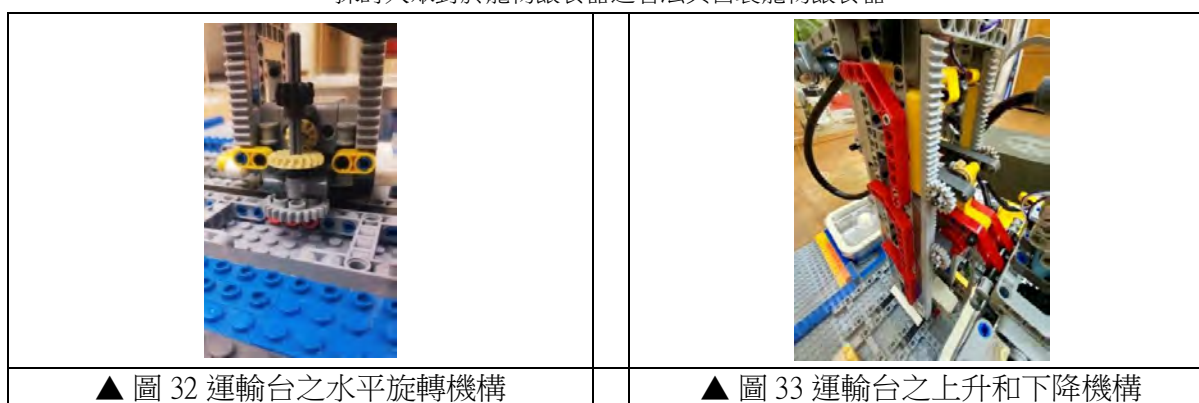
我所製作的寵物餵食器共分為四個部分：解真空機構、開蓋子機構、餵食碗儲存匣和運輸台，我將逐一介紹這些機構的設計原因及運作原理：解真空機構用於破壞餵食盒的真空狀態，我參考機械手臂的機構，使用螺桿機構帶動四根削尖的竹筷，戳進餵食盒和個子的縫隙中來達成(請見圖 30)；開蓋子機構用於將餵食盒的蓋子打開，我發現解真空機構也可同時達成此目的，因此我採用同樣的機構(請見圖 30)；餵食碗儲存匣用於讓餵食器能在固定的範圍中夾取餵食盒，因此我搭建一個圍欄來達成(請見圖 31)；關於運輸台，我則搭建一個能上升、下降和水平旋轉的機構，並將機械夾爪固定在上面(請見圖 32 和 33)。



▲ 圖 30 解真空機構和開蓋子機構



▲ 圖 31 餵食碗儲存匣



四、實際功能及特色

根據製作完成的寵物餵食器，共有以下 12 種功能及特色，分別是：可定時、定量、自動和臨時手動餵食，並且可以真空保鮮、分開存放不同種類食物、能隔絕細小灰塵和蟲類、能避免寵物自行打開餵食器、能緊急停止動作(被動)、防止寵物破壞餵食器，也能方便清潔，詳細的功能敘述請見下表。

表 21 實際功能及特色


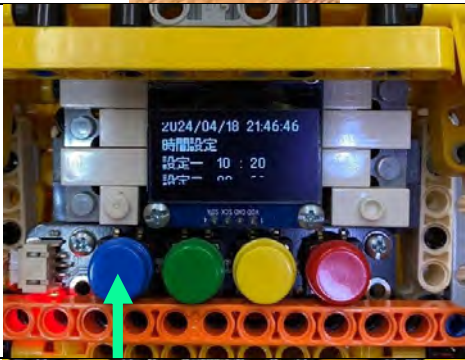

項目	實際功能	功能簡介	類別
1	可定時餵食	能按照設定的時間餵食寵物，一天可設定 2 次餵食時間，並且不用經過設定每天就會重複餵食。	餵食方式
2	可定量餵食	將要餵食的食物放置在真空盒內，份量可依寵物的需求自行調整，但因為要抽真空，不能裝濕食	
3	可自動餵食	透過機器，能實現全自動餵食。	
4	可臨時手動餵食	可以在非設定的時間餵食寵物。	
5	具真空保鮮的功能	用真空餵食盒裝食物，可達到此目的。	食物存放
6	分開存放不同種類食物	可將不同種類食物存放在真空餵食盒(僅限乾食)	
7	食物要能隔絕細小灰塵和蟲類	要餵食的食物存放在真空餵食盒，可預防此問題。	
8	能避免寵物自行打開餵食器	餵食盒放置在有壓克力罩的餵食器，並且因為餵食盒經抽真空，寵物不易打開盒子。	
因餵食器用於展示，故採用透明壓克力作為外殼，較容易觀察到內部構造。			外觀設計
9	能緊急停止動作(被動)	當要緊急停止動作，可暫停 EV3 的程序，或切斷電源。	安全性與保護
10	具防卡設計	餵食器在有壓克力罩，並且 EV3 馬達會在馬達遭遇的阻力時控制功率。	
11	防止寵物破壞餵食器	已在寵物餵食器安裝壓克力罩，若不是大型貓犬，基本上皆能防止寵物破壞餵食器。	
12	能方便清潔	此餵食器只需清潔存放食物的容器(餵食盒)，即是方便清潔。	其他功能

關於供電方式因涉不同而有所異同：EV3 可自性放入電池(6 顆 4 號 AAA)或採用 LEGO EV3 鋰電池電池，以充電的方式供電；ESP32 可由 EV3 供電，或自行插電(Type-C)；而遙控器也可自行插電(Type-C)。



五、操作方式

操作方式在參考問卷調查及設計思考的方式，將餵食分為手動及自動餵食兩種，以下將以表格介紹此寵物餵食器自動和手動餵食的操作方式。

1. 自動餵食

步驟	執行事項	圖片
1	將 EV3、ESP32 和遙控器接上電源並啟動。 將餵食盒放置在指定位置。	無
2		
3	按藍色按鍵切換到下一頁，並按紅色按鍵設定餵食時間(黃色按鍵用於增加時間；綠色按鍵用於減少時間)。	
4	按藍色按鍵回到首頁，再按黃色或綠色按鍵選取「運轉」，最後按紅色按鍵執行。	
5	等待時間到，餵食器就會開始餵食，並在結束後自動歸位，準備下一次餵食(一天進行2次餵食，並且每天會重複餵食。)	無

2. 手動餵食

步驟	執行事項	圖片
1	將 EV3、ESP32 和遙控器接上電源並啟動。	無
2	將餵食盒放置在指定位置。	
3	來到首頁，按黃色或綠色按鍵選取「測試」，最後按紅色按鍵執行(再動作完成後，不會馬上歸零，但在下次餵食後會線歸零再餵食)。	
4	等待餵食動作開始並完成，即完成手動餵食。	無

以下為實際操作演示影片(手動餵食)的網址和 QRcode。

實際操作演示影片
https://www.youtube.com/watch?v=ZsYrNIBhxSM


陸、研究結論與建議

寵物餵食器餵食的方法與功能多樣，為了製作出符合市場需求的餵食器，我整理市面上的功能及特點，透過比對市面上的餵食器後製作調查問卷來了解相關需求及選定目標方向。選擇大眾較為重視的功能，並且加入我的個人需求，篩選較為重要的餵食器功能作為製作的目的。有別於一般市售的寵物餵食器，我設計類似於機械手臂的機構來達成，使其不僅可以分開存放食物，也能達到「真空」保鮮的效果。

為了使此餵食器可以有更符合現代生活的趨勢，更貼近實際大眾的需要，未來我建議可以往以下三個方向來進行：

1. 如同 MoScoW 優先級排序法中的最後一個階段，可使手機能連接寵物餵食器，遠程監控和餵食寵物；或是新增其他遠端操控功能，例如：手機應用程式連接、網路攝像頭、通知功能和語音指令等。
2. 因普遍台灣大眾的居住空間較小，此餵食器尺寸較大，可能會佔用太多空間。建議可將此餵食器當作雛型，利用其他不同材料及工具製造更精密的機構，將尺寸減少。
3. 如果此寵物餵食器要商品化，面對市面上各種不同構造的真​​空盒，該如何設計機構讓餵食器能適應各種真空盒？或是面對更廣大的客戶群，該如何改良餵食器，增加可擴展性？這類問題值得思考。

柒、參考文獻

- 一、黎人瑜(2023)。【獸醫監製】2023 最新 11 款貓咪自動餵食器推薦 2023 排行榜(取自：【獸醫監製】2023 最新 11 款貓咪自動餵食器推薦排行榜 | mybest (my-best.com))
- 二、陳志柏(民 104)。應用 ISM 與 AHP 分析法於寵物產品設計之概念發展。(取自：10013604.doc (cyut.edu.tw))
- 三、佚名(2014)。MoSCoW 優先級排序法。(取自：MoSCoW 優先級排序法 - Praxis Framework)
- 四、親子天下編輯部、台灣大學創新設計學院、DFC 台灣團隊(2017)。設計思考實踐手冊。臺北市：大和書報圖書股份有限公司。
- 五、耐撕 BaBa(2018)。【每日頭條】想讓你的娃學習 STEM 技能？以下十種機器人套件也許能幫你(取自：<https://kknews.cc/education/blvrr39.html>)
- 六、翔太創新科技股份有限公司(2023)。3D 列印可以做什麼？認識 3D 列印機應用、原理和常見技術(取自：<https://toseyinnovation.com/3d-printing/3d-printing/>)
- 七、五十川芳仁(2016)。樂高動力組創意搭建指南機械結構篇。中國：人民郵電出版社。
- 八、五十川芳仁(2016)。樂高動力組創意搭建指南車輛裝置篇。中國：人民郵電出版社。
- 九、DANIELE BENEDETTELLI(2016)。樂高機器人 EV3 創意實驗室。中國：人民郵電出版社。
- 十、PETLIBRO。Granary WIFI Feeder(取自：<https://petlibro.com/products/petlibro-5g-wifi-automatic-pet-feeder?variant=40972430639151>)
- 十一、嘉澎塑膠(2023)。塑膠材質比較，1 張表格秒懂各種塑膠材質種類(取自：<https://www.chia.com.tw/news/4/95>)

附錄

親愛的受訪者：

感謝您參與這份有關寵物餵食器商品的調查，這份問卷將有助於研究者更深入瞭解大眾對於寵物餵食器這類產品的看法和需求，本結果僅供學術參考用，決不會對外公開，請安心作答。

一、寵物的相關問題

1-1 是否曾經擁有寵物？（若否，請跳至 2-4 繼續作答）

是

否

1-2 您曾經擁有的寵物種類是什麼？（可複選）

狗

貓

鳥類

小型哺乳動物（例如兔子、倉鼠等）

其他（請說明）_____

1-3 您與寵物的基本習慣調查（可複選）

寵物需要飲食均衡

寵物需要定量的餵食

寵物需要固定時間的餵食

寵物餵食器要能方便清潔

寵物餵食器的價格是考量的因素

1-4 您的寵物通常每天進食幾餐？

1 餐

2 餐

3 餐

4 餐

5 餐或 5 餐以上

無固定時間，寵物通常在

表現出飢餓行為

運動後

其它：_____的時候餵食

二、對寵物餵食器的使用現況及認知

2-1 您是否有使用過寵物餵食器？（若否，請跳至 2-4 繼續作答）

是

否

2-2 您使用寵物餵食器的主要原因是什麼？（可複選）

方便定時餵食

避免忘記餵食

寵物有特殊飲食需求

超過一天無人在家

其他（請說明）_____

2-3 您是否曾經遇到使用寵物餵食器的問題或困擾？（可複選）

1.餵食過程

食物堵塞或卡住

寵物偷取機器內的食物

食物存儲及保鮮問題

儲存食物的容器容量不足

2.設計和材料

餵食器故障

不耐用材料

有安全疑慮

噪音問題

定時錯誤

電池或電源問題

餵食器安裝問題

清潔或維護餵食器困難

3.遠程控制和技術

網絡連接問題：智能餵食器可能需要網絡連接，斷線可能影響遠程控制。

無法通知問題：某些餵食器可以通知飼主，但可能無法正確通知。

維護困難：清潔或維護餵食器可能會很困難。

4.您是否還曾遇到使用寵物餵食器的其他問題或困擾（如果有，請簡單描述）

否

是，其它：_____

2-4 您認為一款理想的寵物餵食器的價格應該在多少範圍內？

對價格較無概念（不確定）

有明確價格範圍概念：

500 元以下

500 元至 1000 元

1000 元至 2000 元

2000 元至 3000 元

3000 元至 4000 元

4000 元至 5000 元

5000 元以上

2-5 為確保這份問卷為有效問卷，請勾選「沒有」

是

否

沒有

三、商品功能及重視程度

3-1 您對寵物餵食器各功能的重視程度為何？1 代表非常重視，2 代表重視程度僅次於 1……依此類推，5 代表非常不重視。請依照您對各功能的重視程度，在框中打勾。

(每個小題，每個重視程度的選項，只能勾選一次)

餵食方式的重視程度		1 非常重視	普通	5 非常不重視		
編號	項目	1	2	3	4	5
1.	可定時餵食					
2.	可定量餵食					
3.	可自動餵食					
4.	可臨時手動餵食					
食物存放的重視程度		1 非常重視	普通	5 非常不重視		
編號	項目	1	2.	3	4	5
1.	具真空保鮮的功能					
2.	分開存放不同種類食物					
3.	食物要能隔絕細小灰塵或蟲類					
4.	能避免寵物自行打開餵食器					
外觀設計的重視程度		1 非常重視	普通	5 非常不重視		
編號	項目	1	2	3	4	5
1.	尺寸造型					
2.	整體質感					
3.	色彩規劃					
安全性及保護的重視程度		1 非常重視	普通	5 非常不重視		
編號	項目	1	2	3	4	5
1.	能緊急停止動作					
2.	具防卡設計 (即使寵物踩進碗內，正巧碰上轉盤運作時也不會因此受傷)					
3.	防止寵物破壞餵食器					

探討大眾對於寵物餵食器之看法與自製寵物餵食器

供電方式的重視程度		1 非常重視	普通	5 非常不重視		
編號	項目	1	2	3	4	5
1.	自行裝入電池					
2.	充電 (內建電池)					
3.	插座直接供電					
其他功能的重視程度		1 非常重視	普通	5 非常不重視		
編號	項目	1	2	3	4	5
1.	要能方便清潔					
2.	透過手機應用程式遠端遙控					
3.	具視訊功能 (能遠程觀看寵物在家中的狀況)					
4.	具錄音及播放功能					

感謝您參與這份關於寵物餵食器的問卷！我們非常感激您提供的意見和建議，這將有助於我們的研究。如果您有任何其他意見或反饋，請在下方提供：

附加意見或建議：

再次感謝您的參與！

以上是問卷的所有內容！