

嶺東科技大學

資訊科技系

大學部專題

智慧飲品工坊

組別：第十組

學生：羅仕勛 B 0 8 G 0 0 1

呂柏毅 B 0 8 G 0 1 2

黃筠婷 B 0 8 G 0 4 4

蔡睿家 B 0 8 G 1 3 2

林毓翔 B 0 8 G 1 9 6

指導教授：夏則智 教授

中華民國 113年12月24日



嶺東科技大學
LING TUNG UNIVERSITY

資訊科技系
專題報告審定書

學生呂柏毅、羅仕勛、林毓翔、

蔡睿家、黃筠婷

君所提之專題智慧飲品工坊

係經本委員會審議符合專題報告之標準。

指導教授：_____

中華民國 113 年 12 月 24 日

摘要

在這迅速發展的時代裡，現代人們為了享受生活和便利性，而去壓迫到其他人的生活來服務自己，但靠人力來服務自己終究有限，所以有一些人決定改善人的生活品質，發明了機械、機械動力、AI 智能，這些產品不但能減輕人的負擔，它們也有許多優點，例如：效率高、成本低、穩定、安全等等，集結了過往到現今的人們智慧的智慧結晶，因此我們決定設計（智慧飲品工坊）它是由機械手臂結合飲料機的產品，目的是為了解決飲料包裝低薪的工作沒人做、過程繁瑣等問題。有了它的出現，客人可以無所顧忌地點自己想要喝的飲料且不再因為社恐感到焦慮，這次我們機械手臂選用 Cobot280M5 的型號，程式的部分則使用 Python 來設計機械手臂的動作，最後使用了 MIT 的 AppInventor 來設計手機程式介面，進而對機械手臂下達指令。我們目前是以做出小型的（智慧飲品工坊）來進行測試，觀察它是否能達成我們的期望，且嘗試去開發出低成本且能取代人力的產品，讓生活更加便利。

關鍵字:物聯網、人工智能、機器手臂、自助化運營
Python、Android 系統、JSON、MQTT

目錄

摘要	3
第壹章 緒論	8
一、 背景探討	8
二、 現有市場分析	9
三、 研究目的	10
第貳章 文獻探討	11
一、 機械原理	11
1. 按運動形式分類	11
2. 按驅動方式分類	12
3. 綜合實例	12
4. 動力傳輸介紹與分析	14
二、 連線協議	15
1. 有線協議	15
三、 零件加工流程	16
1. 常見材質加工方法	16
(1) 金屬材料加工：	16
(2) 塑料材料加工：	16
(3) 複合材料加工：	16
2. CAD/CAM 系統應用	17
(1) SolidWorks：	17
(2) Rhino：	17
(3) AutodeskInventor：	17
(4) PTCCreo：	17
3. 軟體分析與選擇：	17
(1) Rhino：	17
(2) Autodesk Inventor：	18
(3) PTC Creo：	18
4. 3D 列印技術與快速成型	19
.....	19
第參章 研究方法	20
一、 機械手臂	21
二、 MQTT	22
三、 Python	23

四、	MITAppInventor.....	24
五、	3D 列印&SolidWorks 繪圖軟體.....	29
1.	SolidWorks 繪圖軟體.....	29
2.	FDM	29
第肆章	結論與展望	30
一、	結論.....	30
二、	成果展示	31
三、	未來展望	32
附錄	35
參考文獻	56

圖片目錄

圖 壹 - 01	1 1 2 年飲料市場統計.....	8
圖 壹 - 02	fami 販售機.....	9
圖 壹 - 03	AmazonGo 商店.....	9
圖 壹 - 04	Nespresso 咖啡機.....	10
圖 參 - 01	MyCobot 280.....	20
圖 參 - 02	AppInventor 設計介面.....	20
圖 參 - 03	流程圖.....	21
圖 參 - 04	MQTT 流程圖.....	22
圖 參 - 05	MQTT 後臺介面.....	22
圖 參 - 06	App 介面.....	24
圖 參 - 07	程式方塊.....	25
圖 參 - 08	程式方塊.....	25
圖 參 - 09	程式方塊.....	26
圖 參 - 10	程式方塊.....	27
圖 參 - 11	程式方塊.....	28
圖 肆 - 01	成果展示.....	31
圖 肆 - 02	成果影片.....	31
圖 肆 - 03	原定介面設計.....	33
圖 肆 - 04	協作型機械手臂.....	34

表目錄

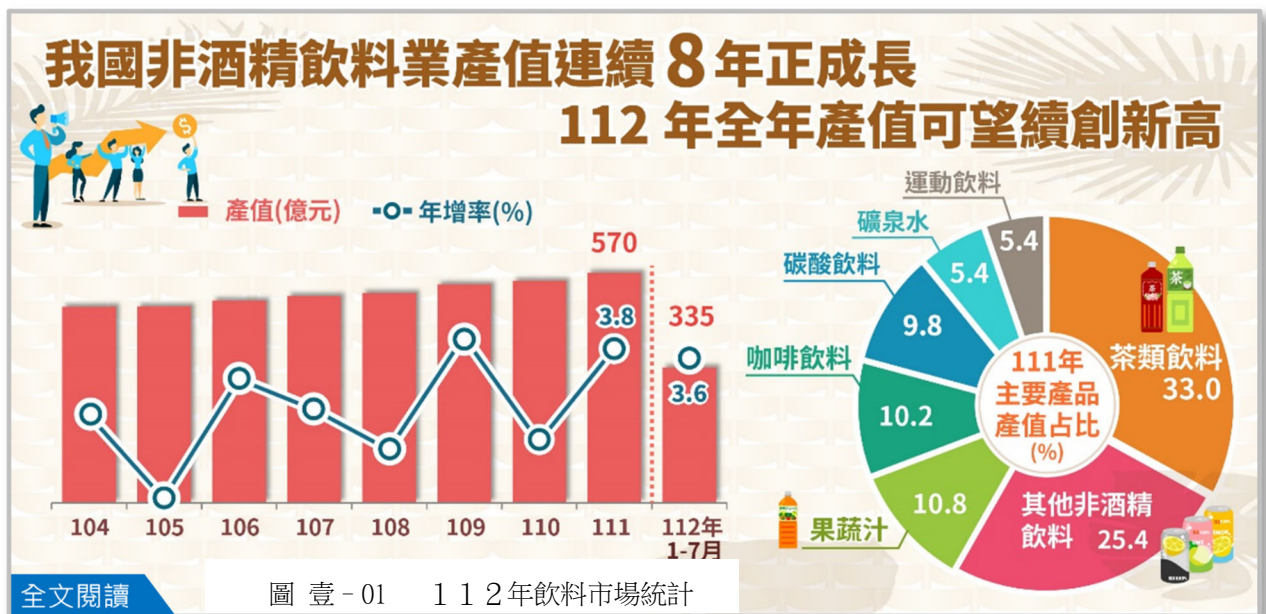
表格 貳-0 1	機械手臂運動模式分類	13
表格 貳-02	機械臂動力穿輸方式分析	15
表格 貳-03	3D 機械原理	19
表格 貳-04	3D 材料分析	19

第壹章 緒論

一、 背景探討

現今人手一杯飲料已成為常態，其中主要獲取方式如：家庭（商業用）咖啡機、飲料店、飲料機及超商，這些的被往往都是十分耗費成本，需要投入人力或是背後都是來自極大的場地去裝設機械化產線，無法實現較低成本開店的條件，飲料店市場也過於飽和，若無創新及話題性的飲品很難脫穎而出創造穩定人流量，而淡出舞台。

近年來飲料店面臨眾多巨大的挑戰，不僅是市場飽和導致品牌之間的競爭日益激烈，原物料價格日漸上漲也使得營運商家從中獲取的利潤空間被壓縮，加上消費者需求快速轉變，更多網美推薦、主打健康飲品、聯名系列等以變成消費者關注的焦點，此現象加劇了傳統飲品業者的銷售壓力。除此之外經營成本的題高，包括房租、人力與水電費等固定支出，也是許多飲料店難以忽視的負擔來源。並在這科技快速推進的時代，無人商店及自動化飲料店被受高度關注，也是許多廠商重點開發項目，這些解決方案都需要很大的場地，難以達到簡易操作及普及化的目地。



截至 1 1 2 年飲料市場統計，我國非酒精性飲品的產值年年上漲，由此可知當代台灣人對於飲料的需求只增不減，為了挖掘商機許多業界龍頭也在極力發展自動飲料機械化的產業鏈，這項技術目前尚未普及化，其中便利性及成本控制遲遲尚未得到最佳解答。

二、 現有市場分析

隨著科技與消費者需求的快速轉變，零售和服務行業呈現多元化發展趨勢，其中無人商店、自動販賣機、自動咖啡機及飲料店展現出不同的市場價值和挑戰。以無人商店為例，AmazonGo 通過感應技術和人工智能實現全自助購物，顧客可直接選購商品並自動扣款，極大地提升了購物效率，尤其適合大城市的高流量地點如機場和商業區。這類商店仍面臨技術維護成本高昂、安全性風險以及消費者購物習慣轉變等挑戰。



圖 壹 - 03 AmazonGo 商店



圖 壹 - 02 fami 販售機

再相同的大環境之下，自動販賣機也在技術升級中找到了新定位，例如富士電機推出支持觸摸屏和行動支付的智能販賣機，售賣傳統飲料、零食和即食餐，又或者是台灣已今普及的全家販賣機，接已成為學校、醫院和商業中心的新零售工具。這類販賣機的便利性對時間有限的上班族和學生有極大的吸引力，但市場上要擴展也需要面對產品線更新和技術成本控制問題。

面對飲料的需求大增但出門意願降低的大環境中，咖啡機技術普及帶動價格親民化的趨勢，咖啡機已成為家庭或現代辦公場所的標配，像 Nespresso 通過智能化設計提供遠程控制和定制化選項，能迅速製作符合個人口味的咖啡，特別在早高峰時段需求旺盛的商業大樓中受到歡迎，但這類咖啡機在提升便利性的同時，也需不斷競爭新技術與更高品質及口味獨特性。

再眾多新興飲料獲取途徑中，傳統飲料店仍然保有一席之地，像喜茶通過健康飲品和季節限定產品吸引年輕消費者，並利用外送服務擴展市場覆蓋。獨立飲料店也憑藉本地化創新和獨特店內氛圍，在競爭激烈的市場中贏得忠實顧客。整體而言，這些模式分別利用科技、便利性或體驗感滿足不同場景下的消費需求，形成零售與服務業在效率與個性化之間的微妙平衡。



圖 壹 - 0 4 Nespresso 咖啡機

三、 研究目的

在這迅速發展的時代裡，現代人們為了享受生活和便利性，而去壓迫到其他人的生活來服務自己，但靠人力來服務自己終究有限，因此我們希望採用異整套完整的系統與半機械化製作流程來取代先今，發明了機械、機械動力、AI 智能，這些產品不但能減輕人的負擔，它們也有許多優點，例如：效率高、成本低、穩定、安全等等，集結了過往到現今的人們智慧的智慧結晶，因此我們決定設計（智慧飲品工坊）它是由機械手臂結合飲料機的產品，目的是為了解決飲料包裝低薪的工作沒人做、過程繁瑣等問題。有了它的出現，客人可以無所顧忌地點自己想要喝的飲料且不再因為社恐感到焦慮。

第貳章 文獻探討

一、 機械原理

機械手臂作為現代自動化領域不可或缺的核心設備，憑藉其高效、精準和穩定的運行能力，廣泛應用於工業製造、醫療、服務及科研領域。在工業 4.0 和智能製造快速發展的背景下，機械手臂已成為推動生產自動化、提高效率和降低人力成本的重要技術手段。

根據機械手臂的設計特性及實際應用，其可以根據運動形式和驅動方式進行分類，並在不同領域展現出卓越的性能。

1. 按運動形式分類

- **關節型機械手臂：**

具備多個旋轉關節，靈活度高，適合複雜工序和多角度操作。應用於飛機機翼製造與汽車生產線中，關節型機械手臂能靈活進行焊接與鉚接。例如，某國際知名汽車製造商導入多關節型機械手臂進行自動化焊接，使焊接精度提升 30%，生產效率提升 20%。主要使用 ABB 和 KUKA 品牌機械手臂。

- **直角坐標型機械手臂：**

依據直線軸移動，結構簡單，精度高，適用於搬運、堆疊或簡單的裝配作業。應用於電子廠裝配線或物流分揀中心。例如，某電子代工廠使用直角坐標型機械手臂對電路板進行精確組裝，確保生產品質一致性。代表品牌包括 Yamaha 和 IAI。

- **圓柱坐標型機械手臂：**

以圓柱坐標系為基準，適合圓周內的取放操作，常見於物流與倉儲系統。應用於自動化倉儲和貨物搬運。例如，亞馬遜倉庫內的自動分揀設備，配備圓柱型機械手臂進行高速分揀和包裹搬運。主要使用三菱電機和川崎重工設備。

- **極坐標型機械手臂：**

基於極坐標系移動，能在較大範圍內進行操作，適合大型設備維護及材料搬運。應用於石油平台和航空維修。例如，某航空公司導入極坐標型機械手臂進行飛機引擎維修，縮短維修時間 40%。代表品牌包括 KUKA 和 Comau。

2. 按驅動方式分類

- **液壓驅動機械手臂：**

承載能力強，適用於重型設備和高負荷操作。應用於建築工地與鋼材搬運。例如，建築工地上使用液壓驅動手臂搬運重型鋼材，提升施工效率。代表品牌為 Caterpillar 和 Kawasaki。

- **電動驅動機械手臂：**

操作精度高，能耗較低，適合精細加工和電子組裝。

應用於醫療設備與電子產品製造。例如，某醫療科技公司使用電動機械手臂進行人工關節的精密製造，使產品合格率提升至 98%，主要使用 FANUC 機械手臂。

- **氣動驅動機械手臂：**

結構簡單，成本低，適合快速輕量級操作。應用於食品加工與包裝生產線。例如，食品加工廠利用氣動驅動手臂快速完成包裝流程。代表廠商如 SMC 和 Festo。

3. 綜合實例

- **電子產品裝配：**

某電子代工廠使用 Yaskawa 的 MotomanGP7 電動驅動機械手臂進行晶片精密組裝。該機械手臂具有 7kg 負載能力和高速多軸控制系統，透過視覺檢測系統確保精準度，良品率達到 99.8%。

- **物流中心：**

某電商企業的自動倉儲系統配備 Omron 的 i4L 直角坐標型機械手臂，每小時分揀超過 5000 件包裹。手臂採用 3 軸設計，結合 AI 識別技術，自動篩選並搬運包裹，使物流成本下降 15%。

- **醫療領域：**

達文西 Xi 手術系統由 IntuitiveSurgical 開發，醫生透過遠端操作四臂機械手臂完成複雜微創手術。該手臂具備高精度內視鏡控制系統，操作靈敏度極高，有助於縮短病患康復時間。

- **科研探索：**

NASA 的 Canadarm2 太空機械手臂安裝於國際太空站，主要負責設備維修和衛星部署。該機械手臂長約 17 米，負載達 11677 公斤，透過遙控與自動程式進行操作，提升太空任務的安全性和準確性。

機械手臂運動模式分類					
分類方式	類型	特點	優點	缺點	應用場景
按運動形式	直角坐標機械手臂	三個直線運動軸，運動簡單	精度高，穩定，重載能力強	運動範圍有限，靈活性較差	CNC機床、自動焊接、物料搬運
	圓柱坐標機械手臂	一個旋轉軸+兩個直線運動軸	結構簡單，成本低	適用空間有限，精度中等	包裝線、裝配、自動取放
	極坐標機械手臂	旋轉+俯仰+伸縮運動	運動範圍廣，覆蓋空間大	控制難度高	採礦設備、搬運機械
	關節型機械手臂（多關節）	類似人手臂，多旋轉關節	靈活性高，適應多自由度運動	結構複雜，成本高	汽車製造、焊接、醫療手術
按驅動方式	液壓驅動	高承載力，大功率	動力強，抗衝擊能力強	結構複雜，維護成本高	重型裝卸、採礦設備
	電動驅動	電機驅動，中小型手臂	精度高，控制靈活，易編程	動力有限，不適合超重負載	電子裝配、生產線檢測、自動焊接
	氣動驅動	壓縮空氣推動，適合輕載	動作迅速，結構簡單，成本低	精度較低，動力小	包裝、自動分揀、灌裝設備
按應用場景	工業機械手臂	重載能力強，適應多種工況	高負載、高強度	成本高，維修要求高	自動焊接、物料搬運、工業塗裝
	協作機械手臂（Cobot）	與人協作，安全性高	操作簡單，靈活，易佈署	負載能力有限	電子裝配、檢測、3C產品加工
	醫療機械手臂	精密操作，運動平穩	高精度，穩定性好	環境要求高，成本高	手術機械、康復輔助
	服務型機械手臂	日常生活或服務行業應用	成本低，操作簡單	精度和負載能力有限	送餐機器人、餐飲服務、護理手臂

表格 貳-01 機械手臂運動模式分類

4. 動力傳輸介紹與分析

● 齒輪傳動

特點與應用：

精確性高，適合大扭矩傳動，可實現較小的速度比誤差。

承載能力強，適用於高負載和長期運行的設備。

可用於正反轉運行，運行穩定，壽命長。

常見於汽車變速箱、工業機床、渦輪減速機等。

選型要點：

速比要求高且空間受限時，優先選用齒輪傳動。

適用於傳遞較大功率或要求運行精度高的場合。

需要考慮齒輪模數、齒數比及材料（如 45 號鋼、合金鋼等）。

● 皮帶傳動

特點與應用：

結構簡單，成本低廉，安裝維護方便。

傳動平穩，具有緩衝吸振作用，能有效保護機械設備。

適合長距離傳動，可允許兩軸間有一定的位移誤差。

常見於風扇、洗衣機、農業機械與小型機床。

選型要點：

需要靜音運行或有振動的設備適合選擇皮帶傳動。

適用於速度較高但負載較輕的傳動系統。

選擇時需考慮皮帶的類型（如 V 型皮帶、同步帶）及帶輪直徑。

● 鏈條傳動

特點與應用：

能夠傳遞較大功率，適合中等距離的傳動系統。

傳動效率高，打滑現象少，速比穩定。

結構緊湊，尺寸小，適用於重載且低速的工況。

廣泛應用於自行車、摩托車、輸送設備與農業機械。

選型要點：

適用於需要高效且空間受限的傳動場合。

鏈條傳動適合高負載和低速場合，適合油污環境下運行。

選擇時需要考慮鏈條的節距、鏈輪齒數及鏈條材料。

機械臂動力穿輸方式分析			
傳動方式	優點	缺點	適用場合
齒輪傳動	高精度、高效率、壽命長	製造成本高，對潤滑要求高	高扭矩、精密機械
皮帶傳動	成本低，緩衝吸震，結構簡單	易打滑，效率低，壽命較短	輕載、高速、長距離
鏈條傳動	承載能力大，不易打滑	噪音大，潤滑保養要求高	重載、中低速

表格 貳-02 機械臂動力穿輸方式分析

二、 連線協議

通訊協議是確保設備與設備之間能夠順利交換數據的規則和標準，廣泛應用於工業、消費電子、物聯網（IoT）等領域。隨著連線需求的增加，有線與無線通訊協議在不同場景中發揮關鍵作用。

1. 有線協議

- **Ethernet（乙太網）**

Ethernet 是目前最常見的有線通訊技術，主要應用於局域網（LAN）和工業網路環境。隨著技術演進，Ethernet 已從傳統的 10Mbps 發展到 1Gbps 甚至更高。工業乙太網的出現使得乙太網在惡劣環境下也能穩定運行，並支援實時數據傳輸。應用於工廠自動化設備之間的數據傳輸、機器人控制系統等。

- **RS-485**

RS-485 是一種串行通訊標準，適用於長距離和高干擾環境的數據傳輸。它支援多點連接，最多可連接 32 個設備，常用於工業自動化領域。應用於電梯控制系統、感測器數據收集與傳輸。

- **Modbus**

Modbus 是一種工業自動化領域廣泛使用的協議，允許設備之間進行主從式通訊。Modbus 有多種形式，包括 ModbusRTU（串行）和 ModbusTCP（基於以太網）。應用於 PLC（可程式邏輯控制器）與感測器之間的數據交換，能源管理系統。

- **USB（通用序列匯流排）**

USB 是廣泛應用於消費電子與工業設備的通訊協議，支援高速數據傳輸與即插即用功能。USB 技術不僅簡化了設備連接，還能同時供電與數據傳輸。隨著 USB3.0 及更高版本的普及，其在工業控制與數據收集領域也越來越常見。應用於工業相機數據傳輸、設備韌體升級、感測器數據擷取。

三、 零件加工流程

1. 常見材質加工方法

(1) 金屬材料加工：

- 車削、銑削：常見於鋼、鋁、銅、鈦等金屬的精密加工。
- 電火花加工(EDM)：對硬度較高的金屬如工具鋼、模具鋼進行精密加工。
- 激光切割與水切割：用於金屬板材的精密切割，特別適合鋼材和不鏽鋼。

(2) 塑料材料加工：

- 塑膠射出成型：適用於高產量生產塑料零件，常用於聚丙烯(PP)、聚氯乙烯(PVC)、聚碳酸酯(PC)等塑料。
- 旋轉成型(Rotational Molding)：通常用於製造大型中空物體，如塑料桶、儲水槽等。
- 3D 列印：用於小批量或定制零件，常見的材料有 ABS、PLA、尼龍等。
- 車削與銑削：在塑料加工中也可使用，常見於製作精密零件或小批量生產。

(3) 複合材料加工：

- 碳纖維與玻纖材料：使用數控銑床或切割機進行加工，並可進行激光切割、磨削等處理。
- 模壓成型：用於大規模生產複合材料結構件。
- 陶瓷材料加工：
 - 磨削與拋光：陶瓷材料一般通過磨削加工，以達到所需的尺寸和表面光潔度。
 - 放電加工(EDM)：適用於某些高硬度陶瓷材料的精密加工。
- 橡膠材料加工：
 - 注射成型：用於生產各種橡膠零件，特別是在汽車、電子和消費品領域。
 - 壓力成型：常見於較大尺寸的橡膠零件。
- 木材與其他軟質材料加工：
 - 車削與銑削：木材通常使用這些方法進行形狀加工。

- 鋸切與切割：適用於木材、塑料和某些軟性材料。
- 激光切割：精確切割木材、塑料或布料。

2. CAD/CAM 系統應用

3D 繪圖與設計技術：使用先進的 3D 建模軟體來進行零件和組件的設計。這些軟體可幫助設計師創建複雜的三維模型並進行虛擬檢查。常見的 3D 繪圖軟體包括以下

(1) SolidWorks：

擁有強大的參數化設計和組合設計功能，適用於工程和機械設計。它的直觀界面和強大的模擬功能，使其成為許多機械設計師的首選。特別適用於機械零件、組件設計以及進行強度分析。

(2) Rhino：

以其在處理自由曲面和非參數化設計方面的優勢而著名。Rhino 在工業設計、珠寶設計和建築設計領域中有著廣泛應用。由於其靈活性，它特別適合需要創造複雜和有機形狀的設計，並且支持多種插件進行功能擴展。

(3) AutodeskInventor：

這是一款專為機械設計開發的 3DCAD 軟體，提供強大的參數化建模、協作設計功能。它擅長創建詳細的裝配設計和精密的機械零件，適用於需要高精度和複雜裝配設計的領域。

(4) PTCCreo：

這款軟體在高精度機械設計和複雜組件設計中表現出色，尤其在產品生命周期管理（PLM）方面有顯著優勢。Creo 擅長於創建可重複使用的設計，並有強大的數據管理和協作功能。

3. 軟體分析與選擇：

SolidWorks：適用於機械工程、產品設計等領域，強大的參數化建模、裝配設計以及分析工具，使其非常適合需要精確控制和設計標準的工程師。它的優勢在於易學易用和多功能的集成，並且有豐富的生態系統支持。

(1) Rhino：

適合需要自由曲面和創意設計的領域，如建築、珠寶、工業設計等。Rhino 具有極高的靈活性，能夠處理不規則和複雜的形狀，非常適合藝術性與設計感較強的項目。其開放的插件支持也擴展了它的應用範圍。

(2) Autodesk Inventor :

強調製造業中機械設計的精密度，特別適合需要詳細工程設計的使用者。它的優勢在於對裝配設計的支持，適合有較大、複雜組件的機械設計。

(3) PTC Creo :

在多領域（如航空、汽車、重型機械等）中廣泛應用，對於處理高難度工程設計和結構分析非常出色。適用於需要協同工作並進行複雜分析的大型團隊。

選擇合適的 3D 繪圖軟體：選擇合適的 3D 設計軟體應根據特定的需求和設計環境進行。例如，若需求是快速創建複雜的自由曲面或藝術性設計，則 Rhino 可能是最好的選擇；如果是進行精確的機械設計並且強調製造支持，則 SolidWorks 或 Autodesk Inventor 可能更為合適。總的來說，選擇應根據設計的複雜性、預算、團隊協作需求及軟體的擴展性等因素來做出決策。

4. 3D 列印技術與快速成型

技術	原理	優點	缺點
FDM	通過射喇叭將熱塑性材料(如PLA或ABS)加熱融化，並依迴歸路徑將材料堆疊。	設備便宜，適合個人和小型企業使用。材料選擇多樣，包括各種小型增展材料。採用開放檔佈格，便於編輯和分享。	列印速度較慢，並且在大型作品中有高序列印的難度。面層緊膨，需要後處理以完善外觀。
SLA	使用紅外線光注射流體光效檔材，進行層層光固化。	能夠對複雜設計進行高精度的出圖。列印品面光滑，適合用於精緻的藝術或模型製作。	光效檔材的選擇不多，並且文件思維很依賴檔光成影趨。為了除去基材，後處理需要充滿積極性和技術級。
SLS	通過激光燒結粉末材料，將材料堆疊成型。每層粉末被激光添加圓項導般的形狀，最終作為層壓新材料。	力層力層很列是甚自據用模程和時導磁活性定型供。如補刪增官裁。	外觀結構需要額外處理以完成。粉末材料經常需要重複利用，以提高成本效益。
DLP	利用數位光源快速固化光效檔材，通過光注射致全面為一平面固化，進行最精度的成型。	成型速度快，適合製作試測或快速出圖。細節精膩，全體成析能力強。	設備成本高，並且光效檔材適用範圍限制。不合適用於力學需求高的成析功能。
MJF	通過多射流技術將粘合劑和粉末材料混合，利用熱能進行逐層燒結，實現高精度的成型。	能夠快速生產功能性部件，適合用於小批量製造。成型後的部件表面細膩，具備良好的力學性能。材料利用率高，減少浪費。	設備成本和運營成本較高。需要專業的後處理設備以完成部件的清理和精加工。

表格 貳-03 3D 機械原理

材料類型	常用材料	特性	應用範圍
熱塑性塑料	PLA	生物可降解，易加工，低收縮率	原型設計、玩具、教具
	ABS	高韌性，耐熱，抗衝擊	工業零件、機械部件
	PETG	強度高，韌性好，耐化學性	容器、食品接觸產品
工程塑料	尼龍（PA）	高強度，耐磨損，柔韌性高	齒輪、承重部件
	PC	高透明度，耐高溫，抗衝擊性高	光學零件、電氣外殼
合成樹脂	光敏樹脂	高精度，表面光滑	模型製作、珠寶設計
金屬材料	鋁合金	重量輕，導熱性高，耐腐蝕	工業部件、航空航天
	不銹鋼	高強度，耐腐蝕	醫療器械、建築結構
	鈦合金	高強度比，生物相容性好	航空航天、醫療植入物
複合材料	碳纖維	輕質，高強度，耐熱性高	汽車部件、航空航天
	木材纖維	模仿木材質感，環保	室內裝飾、藝術品

表格 貳-04 3D 材料分析

第參章 研究方法

在大學期間學習到了很多有關資訊方面的知識，我們善用學校的資源希望能在現有的資源中做一個專題，且我們選用機械手臂 MyCobot280M5 來做為我們專題主要的運作中心，並融合在校期間學到各門的技術。

經過多次討論後，最終選定用手機連動機械來至作專屬飲品來作為本次研究主題，本次採用自由度較高的 Android 平台去開發手機介面程式，智慧飲品工坊主要是用我們設計的 app，做一個點餐的動作然後接收由 app 傳送的資料，再將接收到的資料儲存到 Drink，再由 Drink 傳送到 MQTT，當 Python 接收到關於 MQTT 的資料時，會開始分析傳送的資料去執行接收到的飲料跟佐料的位子，當動作都完成的時候 App 會顯示完成的畫面。

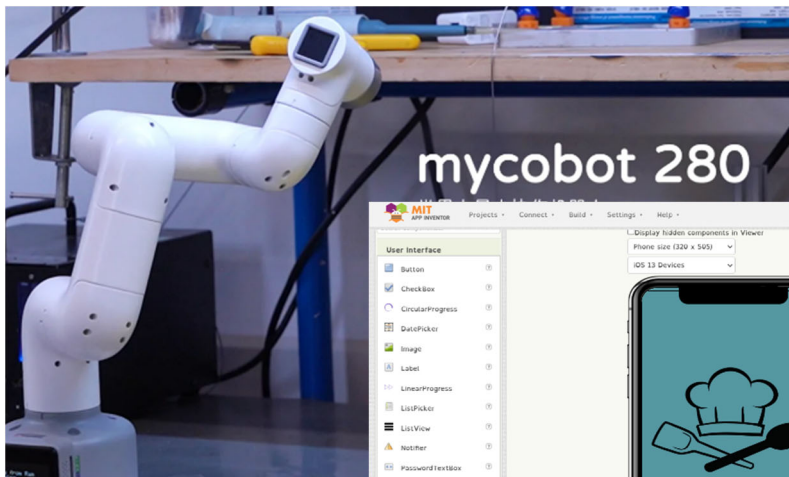


圖 參 - 01 MyCobot 280

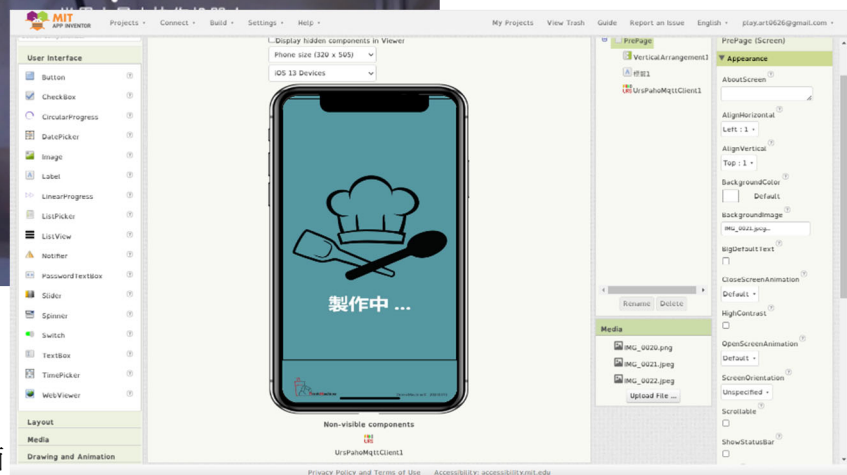
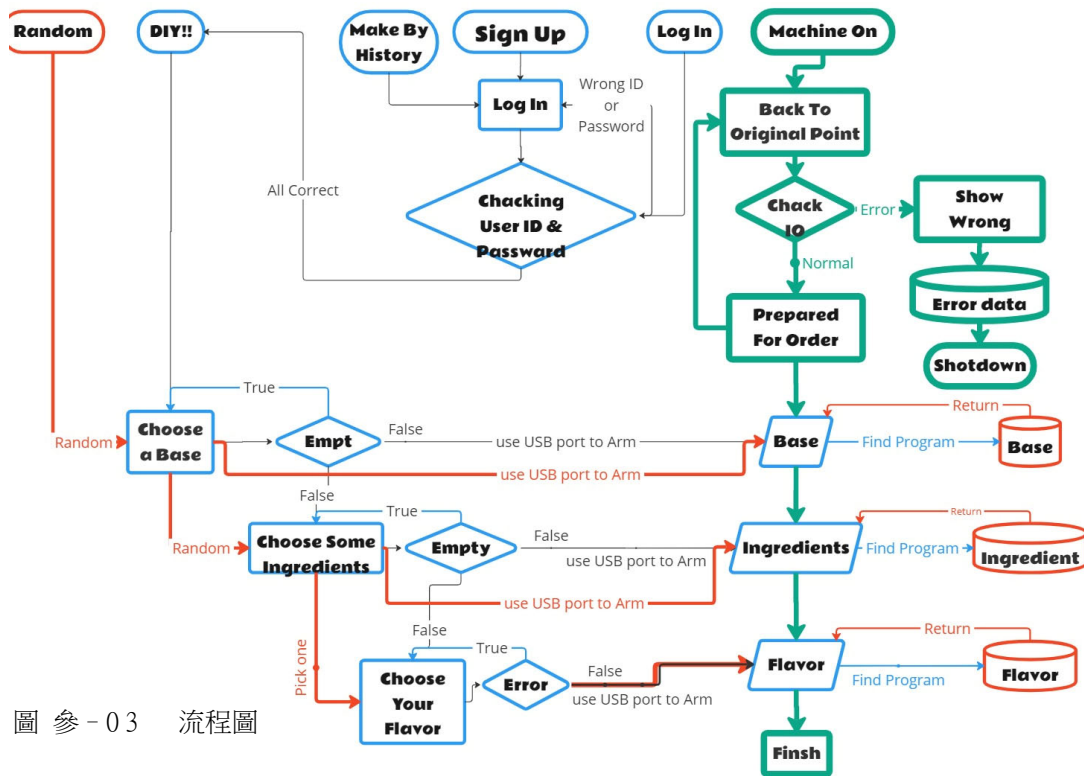


圖 參 - 02 AppInventor 設計介面

另外我們希望智慧飲品工坊能夠帶給人們更多的便利性與品質的保證，運用智慧飲品工坊能夠提高產品的生產效率，也能讓品質保持在一種高品質且一致的狀態上，更能減少人力成本的支出，畢竟在現今社會越來越多人不願意從事這種快速又需要大量製作還要兼顧品質的工作。

如果是使用人力來製作可能會有食安問題的產生，也可能會因為一些人為因素而導致產品的品質下降或不穩定，如果使用機器來製作這些問題都能一併被解決。包括口味調配、比例控制和創新研發，確保更高的衛生標準和製作的一致性，減少人為因素導致的品質波動，因此我們使用智慧飲品工坊來取代人力，創造出更高品質和高效率的製作模式。



圖參-03 流程圖

一、機械手臂

初始拿到 MyCobot 這隻手臂的時候是先閱讀由飊機器人賦予的書籍去研究如何操作手臂，我們第一個使用香蕉派(BananaPi)微電腦來做測試，再用香蕉派裡內建的 Python 來控制手臂，之後使用內建的 node-red 來測試 MQTT，在測試的中途發現他無法跟從香蕉派的 MQTT 傳送到 Windows 端的 MQTT，這時候因為我們還沒有研究透徹所以不了解其時程式跟 MQTT 可以直接在 Windows 端跑，到最後決定使用 Windows 來撰寫寫程式，因為我們取料時需要即時跑到指定的點位去取料，在抓點位時非常的繁瑣，需要先將手臂馬達都停止後再把手臂移動到我們想要的位子讓程式去抓取我們要設定的角度跟點位，在經過多次調整跟測試我們終於可以完成我們想要去執行動作的點位，在研究手臂的過程我們學到了如何使用 Python 去控制手臂，也為未來應用類似技術積累了寶貴的基礎。

二、MQTT

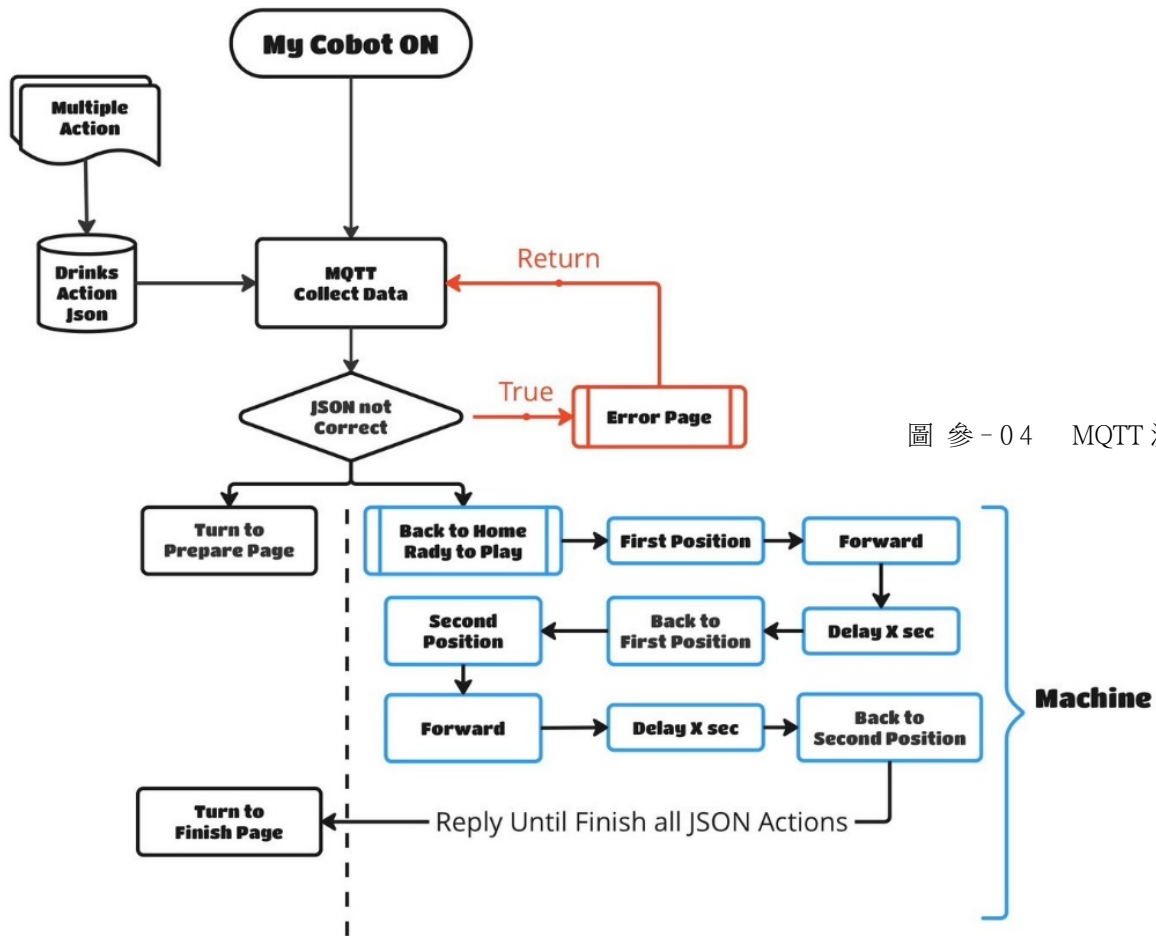


圖 參-04 MQTT 流程圖

再開始知道這個通訊協議時還不知道是什麼作用是啥也不知道，所以我們在手臂賦予的教材裡跟網站去研究，在研究過程中發現了一個軟體 MQTTBOX 是可以接收 MQTT 發出來的訊息，首先我們必須設定參數，MQTT 有一個公共的伺服器 (broker.hivemq.com) 在這個伺服器內所以人都可以使用，但是在設定參數的時候需要注意有一項參數不予許跟任何人重複不然會互相衝突到，每個人都有著自己獨特訂閱的參數，我們使用 MQTTBOX 去觀察 APP 是否有傳輸資料到 MQTT，到最後終於成功利用 MQTTBOX 進行了訊息的監測與確認，發現手臂的訊息能夠順利傳送並被接收，在這個研究的過程我們深刻的理解 MQTT 協議在物聯網中有多某的方便，並學會了如何設定獨特的訂閱參數來避免訊息衝突。

圖 參-05 MQTT 後臺介面



三、 Python

在眾多程式語言中我們使用 Python 作為主要的程式語言去設計機械手臂的程式，它可以在多平台的執行非常方便開發與測試而且它的語法淺顯易懂目前也是程式開發的主流，還有非常多的函示庫也可配合 github 來達到程式要的完整性，在這期間我們藉著書上的範例程式來知道機器手臂的各個角度了解到最大限度是多少以及它的活動範圍，之後在求位置時再配合一個程式可以快速讀取到目前手臂的各個關節角度抓個大概再進行微調，後會先弄一個前置點位然後再往前推反覆的進行模擬和調整以便達到我們想要的運動軌跡經過這段時間的嘗試和調整我們一步步讓機械手臂按照預期動起來從最初的範例程式到後來的調整與優化，雖然過程中有不少卡關的時候現在也沒辦法很順利地執行我們理想中的動作，希望未來可以解決這些問題然後有更好的突破。

程式與註解，請看附錄

四、 MITAppInventor

因為在歷年授課的時候有學習過所以在操作方面不太會有問題，在還有要使用 AppInventor 之前我們是使用 androidStudio 來製作我們要點餐的畫面，是在嘗試要讓 androidStudio 傳送訊息給 MQTT 的時候發現比較繁瑣所以到最後我們選擇了 APPInventor 來作為我們點餐 APP 的軟體，我們想要使用 APPInventor 做出來的 APP 來作為點餐的 APP，在 APP 裡面我們有設定當選取完飲料和佐料時會利用 MQTT 協議傳送客人所選取的飲料和佐料的訊息給後端的 Python，Python 接收到訂單後再由程式這邊去判辦由 MQTT 傳送過來的訊息去控制手臂到指定點位取料，取完料會到取餐區則 APP 會顯示請想要的頁面，在經過這些設計跟實作中，我們成功把 MQTT 協議利用在點餐 APP 裡，使用 APPInventor 製作的頁面非常簡單明讓客戶可以簡單的選擇飲料跟佐料，這樣一來我們實現了自動化的點參與取餐過程，也可以讓客戶覺得有趣，未來可以再把自動化多元化，讓服務過程更加高效簡潔。

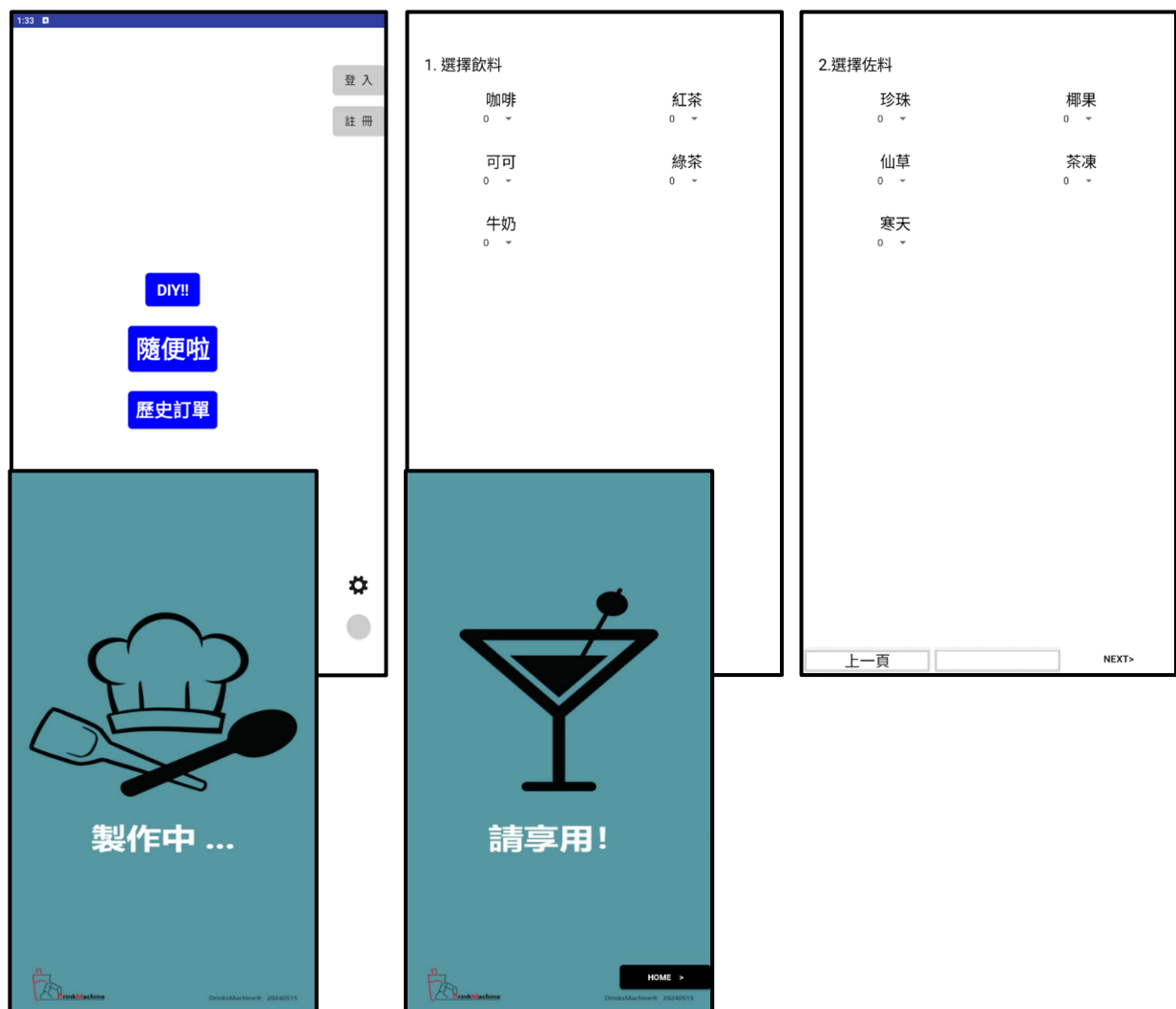


圖 參-06 App 介面

以下為 APPinventor 的積木程式方塊解說：



圖 參 - 07 程式方塊

當 Screen1 初始化時呼叫 MQTT 軟件對我們所設定的伺服器連線。

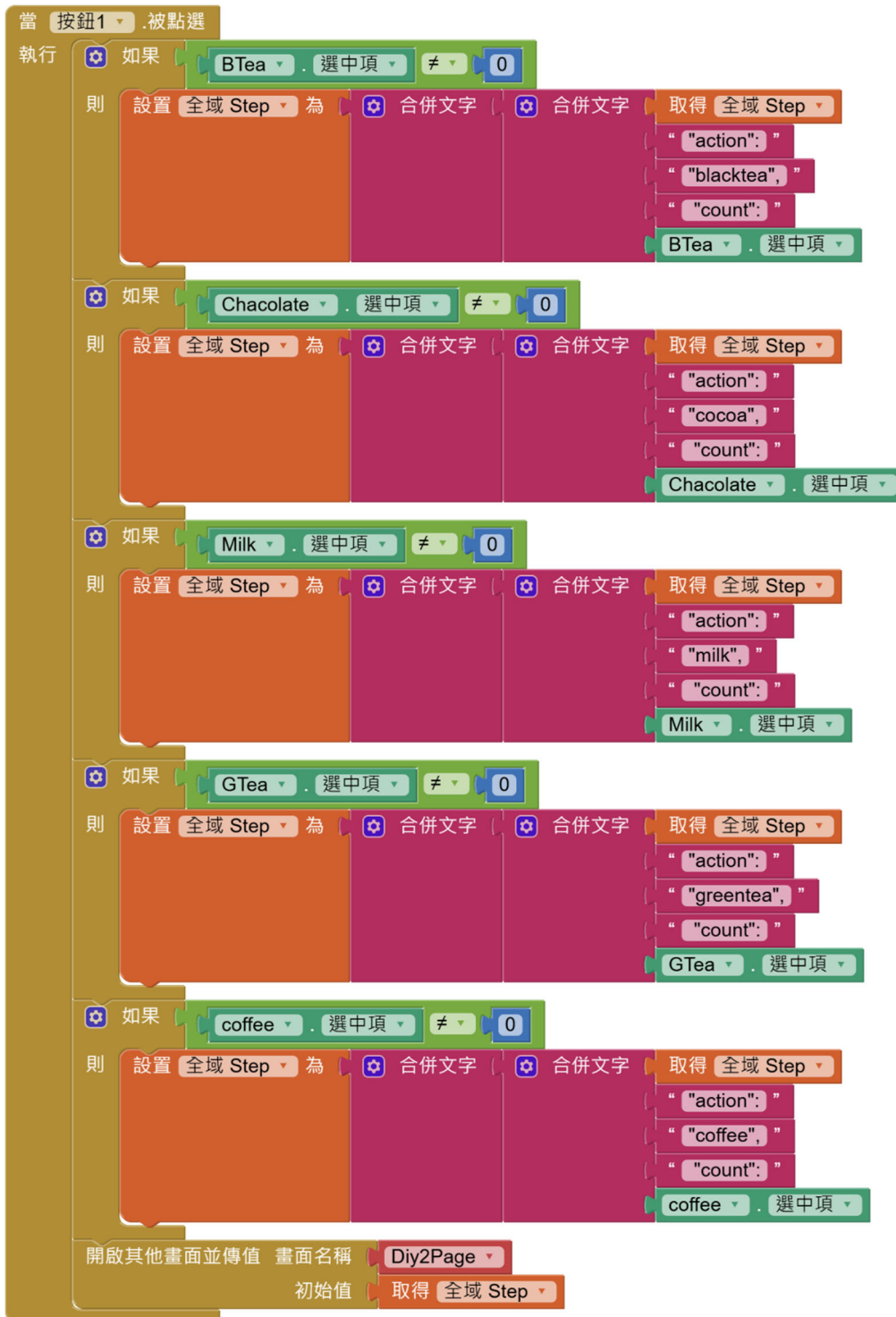
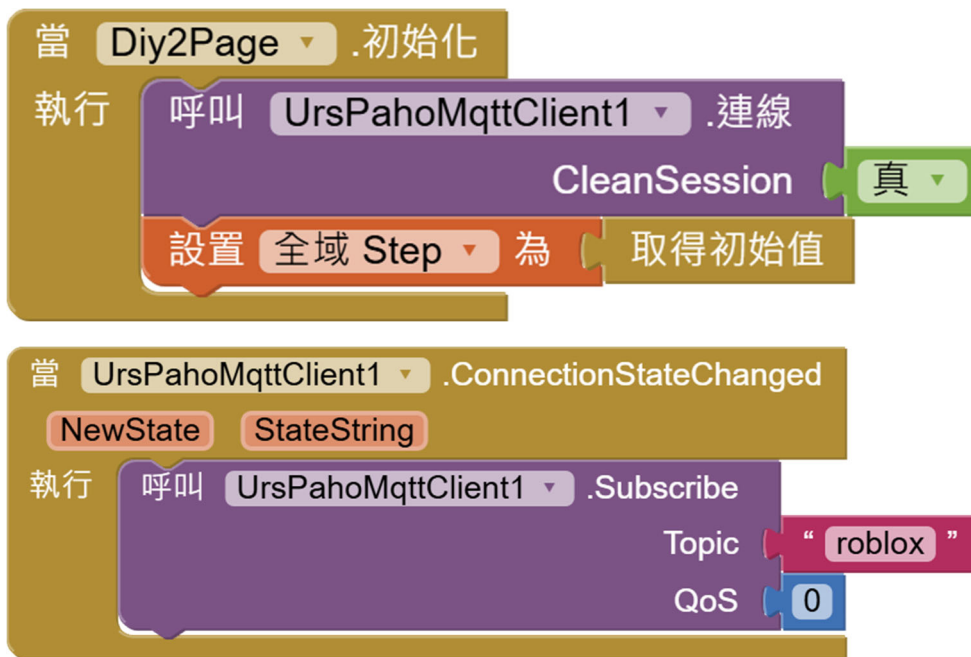


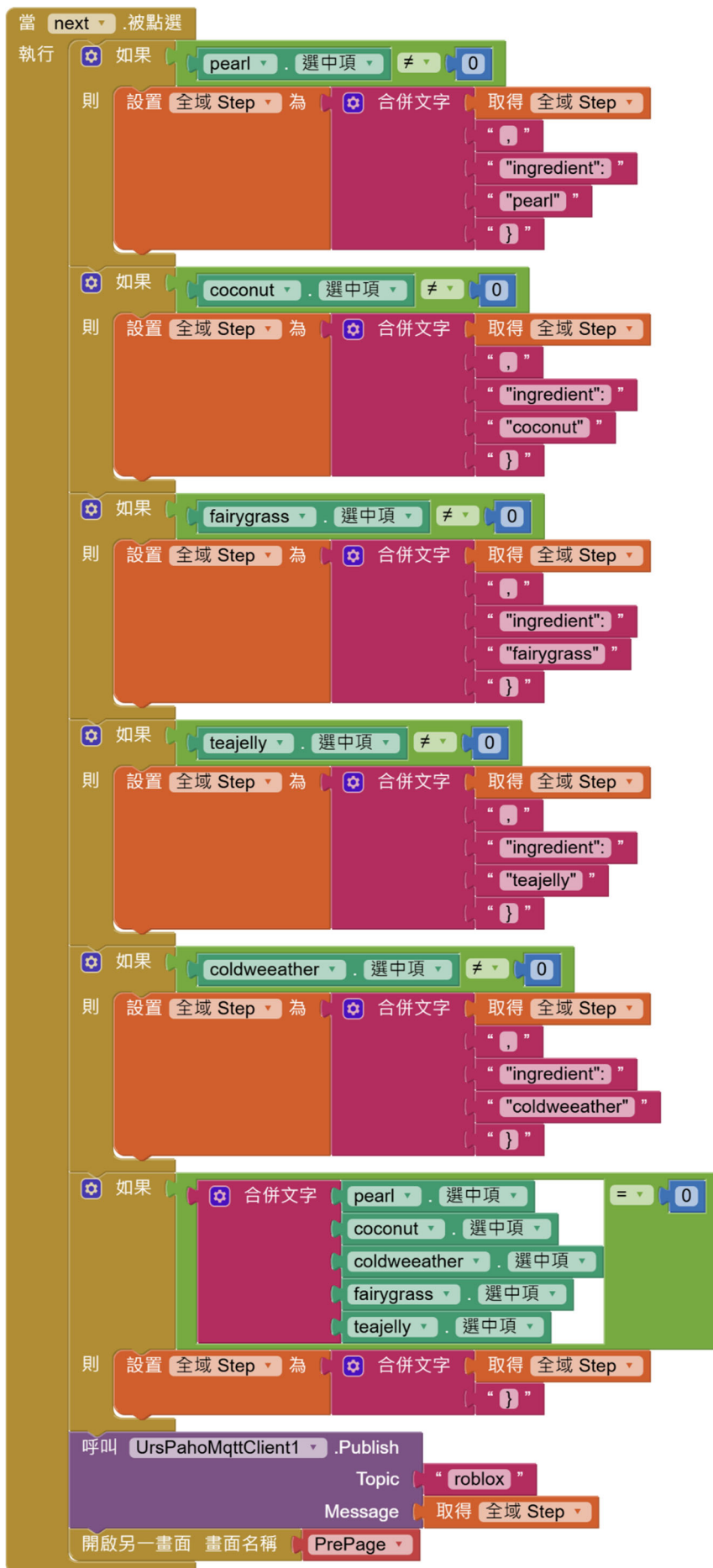
圖 參 - 08 程式方塊

當下一頁的按鈕被點選開始執行後面的動作，程式會隨著使用者選擇的飲料選項來設定一個叫 Step 的變數，紅茶(BTea)假如紅茶的數量不為 0，就將 Step 設定為紅茶+選擇的數量，其他飲料也是如此，當選擇完成時會打開另一個畫面名稱 Diy2Page，同時把剛剛設定的 Step 資料傳送過去。



當 DiyPage2 初始化時呼叫 MQTT 軟件對我們所設定的伺服器連線，然後再將上一個頁面的資料抓過來。

圖 參 - 09 程式方塊



我們按下 next 按鈕時，程式會檢查各個配料的選擇況，然後把選到的東西組合來，最後送出資料給 MQTT 並切換畫面。

檢查配料選擇珍珠(pearl)，如果有選擇程式就會把 "ingredient": "pearl" 加進去。其他配料以此類推。

再來就是檢查有沒有選擇配料如果上面那些配料通通沒選，程式會重置 Step 讓它變成空值 {}。

Topic:roblox 就像寄信時的收件地址；Message 則把剛剛辛苦整理的 Step 資料送給 MQTT 伺服器，讓它知道你選了什麼配料。

再把整理好的 Step 資料發送出去，接著跳到 PrePage，準備讓下一個畫面接手處理。

圖 參 - 10 程式方塊

當 UrsPahoMqttClient1 收到訊息時

這裡代表 MQTT 客戶端接收到了來自某個主題(Topic)的訊息。

檢查 Message 的內容

程式進一步判斷接收到的 Message 是否等於{"action":"Finish"}。

條件成立時

如果接收到的訊息內容符合條件也就是 action 的值為"Finish"，則執行後續動作。

切換到另一個畫面

此時程式就會開啟另一畫面，畫面的名稱是 FinishPage 通常用來顯示任務完成的相關內容。



圖 參 - 11 程式方塊

五、 3D 列印&SolidWorks 繪圖軟體

1. SolidWorks 繪圖軟體

SolidWorks 是一款功能強大的設計工具主要用於產品設計和工程製圖，它能夠幫助設計師和工程師快速創建、模擬和測試各種產品極大地提高設計效率，這款軟體支援 2D 草圖設計使用者可以透過簡單的拉伸或旋轉等操作將 2D 草圖轉換為 3D 模型達到設計者對零件或組件的需求，我們利用 SolidWorks 繪製需要的零件完成設計後將文件輸出為 STL 格式並用 3D 列印機將其打印出來，由於我們設計的零件尺寸較小列印過程中通常會反覆打印多次，藉此排除尺寸誤差以及公差確保零件符合我們的需求在列印完成後我們還會檢查零件的外觀是否平滑如果在組裝測試時發現零件無法順利滑動或存在卡頓現象，我們會使用砂紙將表面進一步打磨確保其能夠順暢地運作提升整體品質。

2. FDM

我們會選擇 FDM 的理由是因為，相較於 SLA 列印機，FDM 的維護成本低，操作介面對新手比較友善，比較適合作為教育用和快速原型開發工具，歷年授課也有教學關於 FDM 列印機要如何操作跟維護，還有考一張 3D 列印工程師的證照，多少有幫助我們更了解 3D 列印機，在我們研究的過程中我們發現 FDM 可以應用的範圍很多，在材料上也很多樣化像是我們這種小零件教學用就使用 PLA，ABS 大多都使用在摩托車的車殼等等，在操作 FDM 列印機時發現在調整一下參數時都會影響到列印的品質，例如：填充率、壁厚、噴嘴直徑、列印產品擺放的方向種種因數都會影響到列印的品質，這些經驗幫助我們提升專業能力，也可以在未來時幫助他人教學更多人，讓更多人了解 3D 列印上多某的有趣。

第肆章 結論與展望

一、 結論

當代生活追求便利與高效，飲品製作也不例外我們透過 MyCobot280M5 機械手臂、MQTT 通訊協議、Python 程式語言及 MITAppInventor 打造了智慧飲品工坊讓飲品調製邁向自動化。不僅是技術的創新，更是為日常生活帶來全新的體驗

透過智慧飲品工坊達到

提升效率：

取代人力節省時間，不僅加快了流程，也讓工作變得更輕鬆高效，機械化的運作能全天候穩定執行，避免了人力因疲勞或疏忽造成的延誤，讓每一杯飲品快速送到顧客手中，滿足高峰時段的需求。

品質保證：

確保每一杯飲品在比例、口感和製作流程上都保持一致，不管是繁忙時段還是特殊訂單，都能穩定呈現最高品質，消除了因人為因素可能導致的差異與不穩定，讓顧客每次喝到的都是熟悉又安心的好味道，讓體驗更放心、更滿意。

安全衛生：

由機械手臂取代人力全程在無接觸在乾淨的環境中製作飲料完全避免了人為的玩笑因素為顧客提供了更加安心的飲品品質，讓每一口都喝得放心

降低成本：

因為我們使用了半自動化機械手臂的操作，減少了人工的需求，不僅降低人力成本也避免了因人員變動或操作失誤所帶來的額外費用，這樣的運作模式不僅提升了效率，還讓整體成本更具彈性和可控性，讓我們能夠更好地應對市場變化。

二、 成果展示



圖 肆 - 01 成果展示

成果測試影片：



圖 肆 - 02 成果影片

三、 未來展望

我們希望能夠簡化使用者的操作流程，讓使用者不會因為繁瑣的点餐系統而退卻，用最簡單最便利的方式讓使用者一目瞭然。提供 24 小時的服務，讓使用者隨時都能滿足想喝飲料的慾望，在飲料店打烊時間也能喝上飲料，設立多個智慧飲品工坊不限地區不限時間讓各地的使用者都能因為智慧飲品工坊在生活上帶來便利。

在製作這個專題的過程中，陸續搜尋到的資料顯示，今年來我們的專題題材不再是新興產業鏈，但我們始終堅決製作這項題目的原因，我們希望找到最佳的解決方案，並降低其中的成本及營運困難點。希望從中加入手機端的操作能夠隨時隨地預約自己喜歡的訂單，並且加入會員機制可以使得點餐更為便捷，發揮其商業模式的最大利益及便利性，以下是我們原定計劃的 App 介面排版，與時間的關係用了 AppInventor 作為開發製作了較為簡易的版本，希望在未來的以後能夠真正在 Android 平台或 iOS 平台上架這樣一套完整銷售平台。



圖 肆 - 03 原定介面設計

我們計劃將現有的機械手臂升級為更大的協作型機械手臂，這項改變不僅能提升操作的精度，還能有效降低運行過程中發生錯誤的可能性。大型的機械手臂不僅擁有更廣泛的運作範圍，還能進一步擴展物料存放區域的覆蓋範圍，從而大幅提升工作效率與靈活性。由於存放空間的增加，我們將能容納更多的飲品與配料種類，這不僅有助於提供更為多樣化的商品選擇，還能根據客戶的需求實現更個性化、定制化的服務內容。



圖 肆 - 04 協作型機械手臂

在軟體設計方面，我們在規劃初期就特別考量了市場需求，參考了台灣知名飲料品牌「多多茶坊」的多樣化飲品特色，並基於此創造出更具差異化的商品選擇。這樣的設計思路讓我們的系統能夠靈活地迎合顧客的不同口味與需求。更有趣的是，當顧客對於選擇飲品感到困惑，或想嘗試全新的飲料搭配時，我們的系統能通過隨機算法，利用機械手臂調製出獨一無二的創意飲品。這種創新功能不僅為顧客帶來驚喜，還有助於提升品牌的特色與吸引力。

附錄

機械手臂運行 python 程式碼：

```
frompymycobot.mycobotimportMyCobot
frompymycobot.genreimportAngle
frompymycobot.genreimportCoord
importpaho.mqtt.clientasmqtt
importjson
importtime
mc=MyCobot('com6','115200')
mc.send_angles([5,62,125,-94,0,-3],50)
mc.set_gripper_state(1,70)
MQTT_BROKER="broker.hivemq.com"
MQTT_PORT=1883
MQTT_TOPIC="roblox"
messang={
    "action":"Finish"
}
messang_json=json.dumps(messang)

defon_connect(client,userdata,flags,rc):#MQTT 連接回調函數
print(f"Connectedwithresultcode{rc}")
client.subscribe(MQTT_TOPIC)
defon_message(client,userdata,msg):#MQTT 消息回調函數
command_data=json.loads(msg.payload.decode())
print(f"Receivedcommanddata:{command_data}")
action=command_data.get("action","")
ingredient=command_data.get("ingredient","")#默认为空字符串
count=command_data.get("count",1)#默认为 1

print(f"Action:{action},Ingredient:{ingredient},Count:{count}")

if"action"incommand_data:
action=command_data["action"]
print(f"Action:{action}")
if"count"incommand_data:
count=int(command_data["count"])#次數
```

```

print(f"Count:{count}")
if "ingredient" in command_data:
    ingredient = command_data["ingredient"]
    print(f"Ingredient:{ingredient}")
    if action == "coffee":
        coffee(count, ingredient)
    elif action == "blacktea":
        blacktea(count, ingredient)
    elif action == "greentea":
        greentea(count, ingredient)
    elif action == "cocoa":
        cocoa(count, ingredient)
    elif action == "milk":
        milk(count, ingredient)
    else:
        print("Unknown action")

def coffee(count, ingredient):
    print(f"Making coffee with ingredient {ingredient}...")
    for i in range(count):
        print(f"Performing action {i+1} of {count}")
        mc.send_angles([28, 77, 55, -41, -0, -28], 40)
        time.sleep(5)
        mc.send_angles([25, 90, 28, -26, -4, -25], 40)
        time.sleep(5)
        mc.send_angles([28, 77, 55, -41, -0, -28], 40)
        time.sleep(5)
        if ingredient:
            print(f"正在添加配料：{ingredient}。")
            if ingredient.lower() == "coldweather":
                print("添加寒天")
                time.sleep(2)
            mc.send_angles([28, 77, 55, -41, -0, -28], 40)
            time.sleep(2)
            mc.send_angles([28, 77, 55, -41, -0, -28], 50)
            time.sleep(3)
            mc.send_angles([25, 90, 28, -26, -4, -25], 50)

```

```

time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="pearl":
print("添加珍珠")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([3,64,100,-76,-2,-3],50)
time.sleep(3)
mc.send_angles([4,71,85,-71,-3,-7],50)
time.sleep(3)
mc.send_angles([3,64,100,-76,-2,-3],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="coconut":
print("添加椰果")

```

```

time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([-24,79,86,-88,0,19],50)
time.sleep(3)
mc.send_angles([-23,76,81,-75,-2,19],50)
time.sleep(3)
mc.send_angles([-24,79,86,-88,0,19],50)
time.sleep(3)
mc.send_angles([-30,62,124,-101,-0,28],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="fairygrass":
print("添加仙草")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([-45,81,64,-65,3,42],50)
time.sleep(3)
mc.send_angles([-42,81,56,-54,2,40],50)
time.sleep(3)
mc.send_angles([-45,81,64,-65,3,42],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)

```

```

mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="teajelly":
print("添加茶凍")
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],40)
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],50)
time.sleep(3)
mc.send_angles([25,90,28,-26,-4,-25],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

else:
print(f"未找到指定的配料：{ingredient}")
time.sleep(1)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)

```



```

mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

defblacktea(count,ingredient):
print(f"Makingblackteawithingredient{ingredient}...")
foriinrange(count):
print(f"Performingaction{i+1}of{count}")
mc.send_angles([3,64,100,-76,-2,-3],40)
time.sleep(5)
mc.send_angles([4,71,85,-71,-3,-7],40)
time.sleep(5)
mc.send_angles([3,64,100,-76.,-2,-3],40)
time.sleep(5)
ifingredient:
print(f"正在添加配料：{ingredient}。")
ifingredient.lower()=="coldweeather":
print("添加寒天")
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],40)
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],50)
time.sleep(3)
mc.send_angles([25,90,28,-26,-4,-25],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)

```

```

time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="pearl":
print("添加珍珠")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([3,64,100,-76,-2,-3],50)
time.sleep(3)
mc.send_angles([4,71,85,-71,-3,-7],50)
time.sleep(3)
mc.send_angles([3,64,100,-76,-2,-3],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="coconut":
print("添加椰果")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([-24,79,86,-88,0,19],50)
time.sleep(3)
mc.send_angles([-23,76,81,-75,-2,19],50)
time.sleep(3)
mc.send_angles([-24,79,86,-88,0,19],50)
time.sleep(3)
mc.send_angles([-30,62,124,-101,-0,28],50)

```

```

time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="fairygrass":
print("添加仙草")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([-45,81,64,-65,3,42],50)
time.sleep(3)
mc.send_angles([-42,81,56,-54,2,40],50)
time.sleep(3)
mc.send_angles([-45,81,64,-65,3,42],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="teajelly":
print("添加茶凍")

```

```

time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],40)
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],50)
time.sleep(3)
mc.send_angles([25,90,28,-26,-4,-25],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

else:
print(f"未找到指定的配料：{ingredient}")
time.sleep(1)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

defgreentea(count,ingredient):
print(f"Makinggreenteawithingredient{ingredient}...")
foriinrange(count):
print(f"Performingaction{i+1}of{count}")

```

```

mc.send_angles([-24,79,86,-88,0,19],40)
time.sleep(5)
mc.send_angles([-23,76,81,-75,-2,19],40)
time.sleep(5)
mc.send_angles([-24,79,86,-88,0,19],40)
time.sleep(5)
mc.send_angles([-30,62,124,-101,-0,28],50)
time.sleep(3)
if ingredient:
    print(f"正在添加配料：{ingredient}。")
    if ingredient.lower()=="coldweather":
        print("添加寒天")
        time.sleep(2)
        mc.send_angles([28,77,55,-41,-0,-28],40)
        time.sleep(2)
        mc.send_angles([28,77,55,-41,-0,-28],50)
        time.sleep(3)
        mc.send_angles([25,90,28,-26,-4,-25],50)
        time.sleep(3)
        mc.send_angles([5,62,125,-94,0,-3],50)
        time.sleep(5)
        mc.send_angles([168,62,125,-94,0,-3],50)
        time.sleep(3)
        mc.send_angles([153,70,12,-11,0,-6],50)
        time.sleep(3)
        mc.send_angles([153,70,13,-10,-100,0],50)
        time.sleep(5)
        mc.send_angles([5,62,125,-94,0,-3],50)
        time.sleep(5)
        client.publish(MQTT_TOPIC,messang_json)

    elif ingredient.lower()=="pearl":
        print("添加珍珠")
        time.sleep(2)
        mc.send_angles([5,62,125,-94,0,-3],40)
        time.sleep(2)
        mc.send_angles([3,64,100,-76,-2,-3],50)
        time.sleep(3)

```

```

mc.send_angles([4,71,85,-71,-3,-7],50)
time.sleep(3)
mc.send_angles([3,64,100,-76,-2,-3],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="coconut":
print("添加椰果")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([-24,79,86,-88,0,19],50)
time.sleep(3)
mc.send_angles([-23,76,81,-75,-2,19],50)
time.sleep(3)
mc.send_angles([-24,79,86,-88,0,19],50)
time.sleep(3)
mc.send_angles([-30,62,124,-101,-0,28],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)

```

```

time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="fairygrass":
print("添加仙草")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([-45,81,64,-65,3,42],50)
time.sleep(3)
mc.send_angles([-42,81,56,-54,2,40],50)
time.sleep(3)
mc.send_angles([-45,81,64,-65,3,42],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="teajelly":
print("添加茶凍")
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],40)
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],50)
time.sleep(3)
mc.send_angles([25,90,28,-26,-4,-25],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)

```

```

time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

else:
print(f"未找到指定的配料：{ingredient}")
time.sleep(1)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

defcocoa(count,ingredient):
print(f"Makingcocoawithingredient{ingredient}...")
foriinrange(count):
print(f"Performingaction{i+1}of{count}")
mc.send_angles([-45,81,64,-65,3,42],40)
time.sleep(5)
mc.send_angles([-42,81,56,-54,2,40],40)
time.sleep(5)
mc.send_angles([-45,81,64,-65,3,42],40)
time.sleep(5)
ifingredient:
print(f"正在添加配料：{ingredient}。")
ifingredient.lower()=="coldweather":
print("添加寒天")

```



```

time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],40)
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],50)
time.sleep(3)
mc.send_angles([25,90,28,-26,-4,-25],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="pearl":
print("添加珍珠")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([3,64,100,-76,-2,-3],50)
time.sleep(3)
mc.send_angles([4,71,85,-71,-3,-7],50)
time.sleep(3)
mc.send_angles([3,64,100,-76,-2,-3],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)

```

```

mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="coconut":
print("添加椰果")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([-24,79,86,-88,0,19],50)
time.sleep(3)
mc.send_angles([-23,76,81,-75,-2,19],50)
time.sleep(3)
mc.send_angles([-24,79,86,-88,0,19],50)
time.sleep(3)
mc.send_angles([-30,62,124,-101,-0,28],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="fairygrass":
print("添加仙草")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([-45,81,64,-65,3,42],50)
time.sleep(3)
mc.send_angles([-42,81,56,-54,2,40],50)
time.sleep(3)

```

```

mc.send_angles([-45,81,64,-65,3,42],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="teajelly":
print("添加茶凍")
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],40)
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],50)
time.sleep(3)
mc.send_angles([25,90,28,-26,-4,-25],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

else:
print(f"未找到指定的配料：{ingredient}")
time.sleep(1)

```

```

mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

def milk(count,ingredient):
    print(f"Making milk with ingredient {ingredient}...")
    for i in range(count):
        print(f"Performing action {i+1} of {count}")
        mc.send_angles([28,77,55,-41,-0,-28],40)
        time.sleep(5)
        mc.send_angles([25,90,28,-26,-4,-25],40)
        time.sleep(5)
        mc.send_angles([28,77,55,-41,-0,-28],40)
        time.sleep(5)
        if ingredient:
            print(f"正在添加配料：{ingredient}。")
            if ingredient.lower()=="coldweather":
                print("添加寒天")
                time.sleep(2)
                mc.send_angles([28,77,55,-41,-0,-28],40)
                time.sleep(2)
                mc.send_angles([28,77,55,-41,-0,-28],50)
                time.sleep(3)
                mc.send_angles([25,90,28,-26,-4,-25],50)
                time.sleep(3)
                mc.send_angles([5,62,125,-94,0,-3],50)
                time.sleep(5)
                mc.send_angles([168,62,125,-94,0,-3],50)
                time.sleep(3)
                mc.send_angles([153,70,12,-11,0,-6],50)

```

```

time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="pearl":
print("添加珍珠")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([3,64,100,-76,-2,-3],50)
time.sleep(3)
mc.send_angles([4,71,85,-71,-3,-7],50)
time.sleep(3)
mc.send_angles([3,64,100,-76,-2,-3],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="coconut":
print("添加椰果")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([-24,79,86,-88,0,19],50)
time.sleep(3)
mc.send_angles([-23,76,81,-75,-2,19],50)

```

```

time.sleep(3)
mc.send_angles([-24,79,86,-88,0,19],50)
time.sleep(3)
mc.send_angles([-30,62,124,-101,-0,28],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="fairygrass":
print("添加仙草")
time.sleep(2)
mc.send_angles([5,62,125,-94,0,-3],40)
time.sleep(2)
mc.send_angles([-45,81,64,-65,3,42],50)
time.sleep(3)
mc.send_angles([-42,81,56,-54,2,40],50)
time.sleep(3)
mc.send_angles([-45,81,64,-65,3,42],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)

```

```

client.publish(MQTT_TOPIC,messang_json)

elif ingredient.lower()=="teajelly":
print("添加茶凍")
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],40)
time.sleep(2)
mc.send_angles([28,77,55,-41,-0,-28],50)
time.sleep(3)
mc.send_angles([25,90,28,-26,-4,-25],50)
time.sleep(3)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

else:
print(f"未找到指定的配料：{ingredient}")
time.sleep(1)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
mc.send_angles([168,62,125,-94,0,-3],50)
time.sleep(3)
mc.send_angles([153,70,12,-11,0,-6],50)
time.sleep(3)
mc.send_angles([153,70,13,-10,-100,0],50)
time.sleep(5)
mc.send_angles([5,62,125,-94,0,-3],50)
time.sleep(5)
client.publish(MQTT_TOPIC,messang_json)

```

```
client=mqtt.Client()  
client.on_connect=on_connect  
client.on_message=on_message  
client.connect(MQTT_BROKER,MQTT_PORT,60)#連接到 MQTTBroker  
client.loop_forever()#開始 MQTT 循環
```


參考文獻

Amazon.(無日期).AmazonGo:Ourcheckout-freeshoppingexperience.擷取自

<https://www.amazon.com/b?ie=UTF8&node=16008589011>

Analysys.(無日期).Chinaunmannedretailmarketdata.擷取自 <https://www.analysys.cn/>

calaCoco.(無日期).日本的販售機 CokeON.擷取自

<https://apps.apple.com/jp/app/cokeon%E3%82%B3%E3%83%BC%E3%82%AF%E3%82%A%E3%83%B3/id1088184021>

CoCoBar.(無日期).CoCoCafe 無人旗艦商用咖啡機.擷取自

<https://www.cocoi.com.tw/ai3000-feature/machine-introduction/>

CoffeeBI.(無日期).Professionalanalysisofthecoffeeindustry.擷取自 <https://coffeebi.com/>

Community.AppInventorMIT.(無日期).MQTTclientextension.擷取自

<https://community.appinventor.mit.edu/t/mqtt-client-extension/17301>

Foundation.Eclipse.(無日期).EclipsePahoMQTTPythonClient.擷取自

<https://www.eclipse.org/paho/index.php?page=clients/python/index.php>

FoundationSoftwarePython.(無日期).PythonDeveloper'sGuide.擷取自

<https://devguide.python.org/>

IBISWorld.(無日期).Professionalindustryreports.擷取自 <https://www.ibisworld.com/>

Inventor.App.(無日期).JSONTextDecodeandJSONTextDecodeWithDictionaries.擷取自

<https://appinventor.mit.edu/explore/ai2/json>

Inventor.App.(無日期).ListViewdocumentation.擷取自

<https://appinventor.mit.edu/explore/ai2/listview>

JSON.org.(無日期).JSONofficialdocumentation.擷取自

<https://community.appinventor.mit.edu/t/how-to-parse-json-data-in-app-inventor/17459>

Markets.andResearch.(無日期).Globalindustrymarketresearch.擷取自

<https://www.researchandmarkets.com/>

NOOONE 無人製茶.(無日期).NOOONE 無人製茶.擷取自 <https://nooone.1shop.tw/>

Robotics.Elephant.(無日期).ElephantRoboticsofficialdocumentation.擷取自
<https://www.elephantrobotics.com/document/>

Robotics.Elephant.(無日期).PyMyCobotSDK(PyPI).擷取自
<https://pypi.org/project/pymycobot/>

Statista.(無日期).Industrystatisticsandmarketresearchreports.擷取自
<https://www.statista.com/>

Times.JapanThe.(無日期).Japanesevendingmachinecultureandapplications.擷取自
<https://www.japantimes.co.jp/>

Watch.MarketVending.(無日期).Vendingmachineindustrymarketandtrends.擷取自
<https://www.vendingmarketwatch.com/>

雀巢專業餐飲.(無日期).NescafeFTP30V2.0.擷取自
<https://www.nestleprofessional.tw/product/nescafe-ftp30-v20-0>

鄭芝珊.(無日期).百睿達 AI 手搖飲機台.擷取自
<https://money.udn.com/money/story/5635/8083366>

曜比科技股份有限公司.(無日期).線上點餐.擷取自 <https://pos.qlieer.com/mobileorder/>

Ethernet Alliance. (n.d.). Ethernet technology overview. Retrieved from
<https://ethernetalliance.org>

Modbus Organization. (n.d.). Modbus protocol specifications. Retrieved from
<https://modbus.org>

USB-IF. (n.d.). USB developer resources. Retrieved from <https://usb.org>

Wi-Fi Alliance. (n.d.). Wi-Fi technology overview. Retrieved from <https://wi-fi.org>

Zigbee Alliance. (n.d.). Zigbee protocol overview. Retrieved from <https://zigbeealliance.org>

LoRa Alliance. (n.d.). LoRaWAN specifications and documents. Retrieved from <https://loralliance.org>

3rd Generation Partnership Project (3GPP). (n.d.). 5G specifications and technical documentation. Retrieved from <https://3gpp.org>

OPC Foundation. (n.d.). OPC UA information model. Retrieved from <https://opcfoundation.org>

MQTT.org. (n.d.). MQTT protocol overview. Retrieved from <https://mqtt.org>

Intuitive Surgical. (n.d.). da Vinci Surgical System. Retrieved from <https://www.intuitive.com/en-us/products-and-services/da-vinci>

NASA. (n.d.). Canadarm2: The Space Station's Robotic Arm. Retrieved from https://www.nasa.gov/mission_pages/station/structure/elements/canadarm2.html

Omron. (n.d.). i4L SCARA Robot. Retrieved from <https://industrial.omron.us/en/products/i4l>

Yaskawa. (n.d.). Motoman GP7 Industrial Robot. Retrieved from <https://www.yaskawa.com/products/robots/gp-series/gp7>

Dassault Systèmes. (n.d.). SolidWorks. Retrieved from <https://www.solidworks.com>

McNeel & Associates. (n.d.). Rhinoceros 3D. Retrieved from <https://www.rhino3d.com>

Autodesk Inc. (n.d.). Inventor. Retrieved from <https://www.autodesk.com/products/inventor/overview>

Autodesk Inc. (n.d.). Fusion 360. Retrieved from <https://www.autodesk.com/products/fusion-360/overview>

CNC Software, LLC. (n.d.). Mastercam. Retrieved from <https://www.mastercam.com>

Hexagon AB. (n.d.). Edgecam. Retrieved from <https://www.hexagon.com/products/edgecam>

PTC Inc. (n.d.). Creo. Retrieved from <https://www.ptc.com/en/products/cad/creo>

GibbsCAM. (n.d.). GibbsCAM. Retrieved from <https://www.gibbscam.com>