

臺北市立大安高級工業職業學校專題製作競賽
「專題組」作品說明書



群別：電機與電子群

作品名稱：毛小孩管家

關鍵詞：遠端控制、訊息提醒、監控

壹、目錄

壹、 摘要.....	1
貳、 研究動機.....	1
參、 主題與課程之相關性或教學單元之說明.....	1
一、 硬體製作	1
(一)雷射切割機	1
二、 電路雕刻	2
(一)電路雕刻機	2
三、 程式撰寫	2
肆、 研究方法.....	3
一、 研究流程	3
(一)研究步驟.....	3
二、 使用材料及工具	3
(一)外殼設計	3
(二)使用材料.....	4
(三)軟體	11
伍、 研究結果.....	15
一、 硬體結構	15
(一)供給飼料部分(如圖 26)	15
(二)供給水部分(如圖 27)	16
(三)遠端視訊(如圖 28)	16
二、 操作流程	16
陸、 討論.....	21
一、 繪圖選擇	21
二、 飼料槽出飼料	21
三、 訊息傳輸	21
柒、 結論.....	21
捌、 參考資料及其他.....	22

表目錄

表 1 研究步驟.....	3
表 2HMI 規格.....	4
表 3 超音波測距傳感器規格.....	錯誤! 尚未定義書籤。
表 4 繼電器模組規格.....	5
表 5 電磁水閥規格.....	5
表 6 電源供應器規格.....	6
表 7 可調自動升降壓模組規格.....	6
表 8 Arduino mega2560 規格.....	7
表 9 Arduino 水位/液位傳感器規格.....	8
表 10 壓力感測器規格.....	8
表 11 排風風扇規格.....	9
表 12 溫度/濕度傳感器規格.....	9
表 13 ESP8266 規格.....	10
表 14 Raspberry Pi 3B+規格.....	10
表 15 Raspberry Pi 第二代 HD 攝影模組規格.....	11

圖目錄

圖 1 雷射切割機.....	2
圖 2 電路雕刻機	2
圖 3 Arduino 程式	3
圖 4 HMI.....	4
圖 5 超音波測距傳感器.....	錯誤! 尚未定義書籤。
圖 6 繼電器模組.....	5
圖 7 電磁水閥.....	5
圖 8 電源供應器.....	6
圖 9 可調自動升降壓模組.....	6
圖 10 Arduino mega2560.....	7
圖 11 Arduino 水位/液位傳感器.....	8
圖 12 壓力感測器.....	8
圖 13 排風風扇.....	9
圖 14 溫度/濕度傳感器.....	9
圖 15 ESP8266.....	10
圖 16 Raspberry Pi 3B+.....	10
圖 17 Raspberry Pi 第二代 HD 攝影模組	11
圖 18 Autodesk Inventor Professional2020	12
圖 19 機構圖.....	12
圖 20 ArduinoIDE	12
圖 21 ArduinoIDE 程式.....	13
圖 22 APP Inventor.....	13
圖 23 Python	14
圖 24 Python 程式	14
圖 25 Python 程式	15
圖 26 供給飼料.....	15
圖 27 供給水.....	16
圖 28 遠端視訊.....	16
圖 29 機構主頁面.....	17
圖 30 機構自動.....	17
圖 31 機構手動.....	18
圖 32 機構飼料量調整.....	18
圖 33 手機主頁面.....	19
圖 34 手機自動.....	19
圖 35 手機手動.....	20
圖 36 手機飼料量調整.....	20

【毛小孩管家】

壹、摘要

一、餵食

本專題提供了遠端控制及定時定量兩種出料方式。藉由遠端控制讓不在家中的飼主能利用手機 app 來操控飼料出口的開與關。飼主也能用定時定量的方式來設定出料的時間與量以避免忘記餵食的困擾。

二、提醒

利用超音波測距傳感器(HC-SR04)及 Arduino 水位/液位傳感器來檢視目前飼料或水的儲存槽中剩餘的量。當槽中剩餘的量低於一定值時，手機將會收到提醒簡訊提醒飼主前往補充飼料與水。

三、監控

在正面處我們安裝了一顆 Raspberry Pi 第二代 HD 攝影模組再利用 wifi 將其所拍攝到的畫面傳送給到網站，飼主可隨時前往該網站查看寵物刀下的動向。

貳、研究動機

在現今萬物皆漲，唯獨薪水不漲的社會中，養小孩變成了一個沉重的經濟負擔，少子化儼然成為了社會趨勢。少了小孩的陪伴隨之而來的則是一份孤獨感，而毛小孩就成為了人們最好的選擇，不僅開銷比育兒少，與生俱來的忠誠也會無時無刻守護在主人身邊。

快速的生活節奏，導致人們可能在離家前無意間忽略了毛小孩的飲食，而出門後，才陷入了遲到與毛小孩的飢餓中陷入了兩難的抉擇當中，毛小孩管家的誕生，除了解決只能在家中餵食的問題；也免去了反覆查看飼料盆中剩餘殘量的困擾，當飼料不足時會發送簡訊提醒飼主；更提供攝像頭讓步在家中的主人能夠解決其思念之情。

參、主題與課程之相關性或教學單元之說明

一、硬體製作

(一)雷射切割機(如圖 1)

在專題課中，我們學習了雷射切割機的使用方式。先在 RDWorksV8 繪製圖型之外觀及雷射時所需的功率、速度並將其存入 USB 中，再將 USB 插入雷射機讀取資料，最後調整位子按下確認即可開始切割。



圖 1 雷射切割機

二、電路雕刻

(一)電路雕刻機(如圖 2)

在高一時，我們只接觸到了麵包板接線，進行簡單的接線以及控制，但在高三時接觸到了 Altium Designer 繪製電路板讓我們在接線上更加方便大大減少了使用面積。我們在 Altium Designer 繪製電路板中，繪製了所需要的 LED 以及電阻和杜邦線所需要的接點，最後利用電路雕刻機刻出所需要電路板並進行焊接，不僅大大減少所需要的空間，更大大減少了接線錯誤的問題。

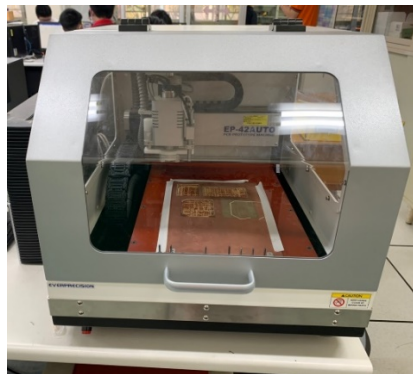


圖 2 電路雕刻機

三、程式撰寫(如圖 3)

在課堂中學習到了 arduino IDE 撰寫成控制，以及使用 app inventor 製作手機 app，在控制程式的部分使用的是 arduino mega 2560 以及接收手機 app 的 esp8266，arduino mega 2560 是作為本次專題的控制核心，esp8266 則是負責接收手機 app 訊號以及傳送訊號給 arduino mega 2560。設定 line notify 時，一開始是使用 IFTTT 系統，但是因為發現傳輸和設定困難，而且訊息也必需很固定，後來我們發現了 arduino 專屬的 line notify 程式庫，它除了好設定且傳輸快外，而且通知的訊息也較自由，所以我們最後決定放棄 IFTTT 直接使用這個程式庫。



圖 3 Arduino 程式

肆、研究方法

一、研究流程

(一) 研究步驟(如表 1)

表 1 研究步驟

	七月	八月	九月	十月	十一月	十二月	一月
製作討論							
材料購買							
程式撰寫							
機構組裝							
測試							

二、使用材料及工具

(一) 外殼設計

主體部分使用機械強度高的壓克力，不僅能防水而且方便加工，外殼機構使用 6 塊壓克力製成，前方的部分切出放置 HMI 及攝像頭的孔。

(二)使用材料

1.HMI(TJC4024K032_011RN) (如圖 4)

作為可觸控的控制面板方便我們在沒有手機時也能操作設備，利用 HMI 發出的信號來對 mega2560 開發板進行控制。編譯時可以使用 USART HMI 有專業的網站能夠方便初學者學習。規格(如表 2)

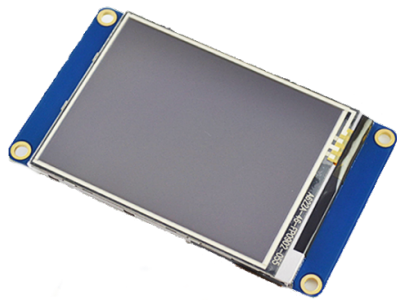


圖 4 HMI

表 2HMI 規格

尺寸	3.2 英吋
觸摸方式	電阻式觸摸
工作溫度	-20~70°C
輸入電壓	4.65~6.5V
分辨率	400×240

2.超音波測距傳感器(HC-SR04) (如圖 5)

利用超音波信號遇到障礙物會返回的特性，由 HC-SR04 發出信號遇到飼料後返回，並由內部計算距離值，來判斷內部飼料是否充足。規格(如表 3)

錯誤! 找不到參照來源。

3.繼電器模組(HL-52S V1.0) (如圖 5)

繼電器具有控制系統和被控制系統，通常應用在自動控制電路中，繼電器實際上是用較小的電流去控制較大電流的一種「自動開關」。故在電路中起著自動調節、安全保護、轉換電路的作用。規格(如表 3)

表 3 繼電器模組規格

最大電壓	AC250V 或 DC30V
最大電流	10A

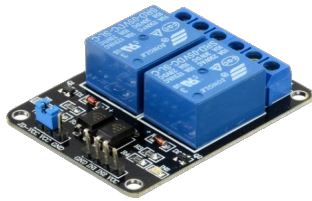


圖 5 繼電器模組

工作電壓	DC5V
------	------

4. 電磁水閥(如圖 6)

常閉式放水電磁閥，使用的是 12v 電壓，方便控制我們的水槽出水。規格(如表 4)



)

圖 6 電磁水閥

表 4 電磁水閥規格

工作電壓	DC24V
功率	4.8W
流量	300CC

5. 12V3A 電源供應器(LRS-35-12) (如圖 7)

能將交流或是直流電轉換成低壓且穩定的直流電，提供給我們的 Arduino Mega2560、電磁水閥、除溼風扇使用。規格(如表 5)



圖 7 電源供應器

表 5 電源供應器規格

輸出電壓	12V
輸出電流	3A
輸出功率	36W
外型尺寸	99×82×30mm

6.可調自動升降壓模組(buck-boost zk-4kx) (如圖 8)

有別於傳統變壓器輸出的電壓較為固定且單一，buck-boost zk-4kx 能夠藉由旋鈕來調整輸出的電壓和電流，能使輸出電壓更為精確。規格(如表 6)



圖 8 可調自動升降壓模組

表 6 可調自動升降壓模組規格

輸入電壓	5~30V
輸出電壓	0.5~30V
輸出電流	3A

輸出功率	35W
電壓顯示分辨率	0.01V
電流顯示分辨率	0.001A
工作頻率	180KHZ
外型尺寸	79×43×26mm

7.Arduino mega2560(如圖 9)

Arduino Mega 2560 為一塊以 ATmega2560 為核心的微控制器開發板，擁有 54 組數位 I/O 端(其中 14 組支援 PWM)，16 組類比 I/O 端。ATmega 2560 內建的 4 路 UART 可以與外部實現串列埠通訊，並預置了 Bootloader 程式，因此可以通過 Arduino 軟體直接下載至 Mega2560 中。我們運用此開發板於控制箱及主體中，達成收放訊號及控制各項負載的功能。規格(如表 7)

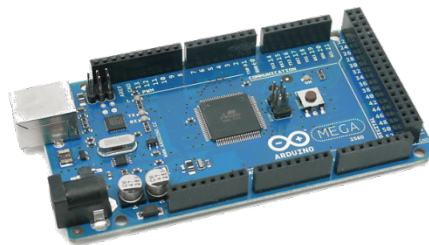


圖 9 Arduino mega2560

表 7 Arduino mega2560 規格

外接輸入電壓	7~12V
工作電壓	5V
數位 I/O 接腳	54(其中 15 支提供 PWM 輸出)
頻率	16MHZ
閃存空間	256KB
主控晶片	AT mega2560
外型尺寸	101×54mm
重量	36g

8.Arduino 水位/液位傳感器(如圖 10)

水位傳感器基於所測液體之靜壓和高度成比例的原理來產生出信號，我們用於偵測水箱內水位的高低，當水箱內水量不足時會發送訊息提醒飼主補水。

外殼設計。規格(如表 8)

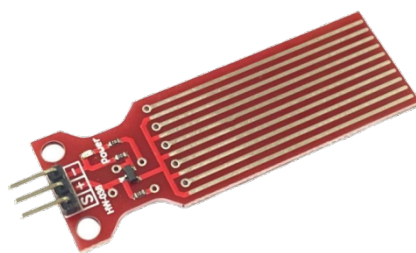


圖 10 Arduino 水位/液位傳感器

表 8 Arduino 水位/液位傳感器規格

工作電壓	DC3~5V
工作電流	0~20Ma
工作濕度	10%~90%無凝結
工作溫度	10~30°C
檢測面積	40×16mm
外型尺寸	62×20×8mm
重量	3.5g

9. 壓力感測器(FSR-402) (如圖 11)

FSR402 是一款重量輕，體積小，感測精度高，超薄型電阻式壓力傳感器。這款壓力傳感器是將施加在 FSR 傳感器薄膜區域的壓力轉換成電阻值的變化，從而獲得壓力信息。壓力越大，電阻越低。其允許用在壓力 100g-10kg 的場合。我們將壓力感測器放置於底部的飼料盤上，用來偵測底下飼料剩餘的量。規格(如表 9)

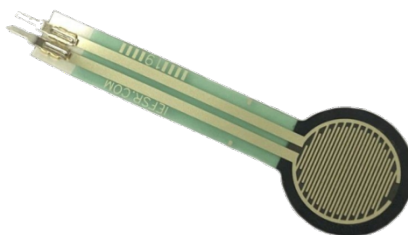


圖 11 壓力感測器

表 9 壓力感測器規格

外型尺寸	2.375"×0.75"×0.018"
感測區域	12.7mm
感測值	0.1~100 Nt
環境溫度	-30~+70°C

10.排風風扇(GD128025HS) (如圖 12)

利用 12 伏特直流風扇來處理飼料的通風問題對飼料來說潮濕且悶的環境會使飼料軟化甚至發霉。我們利用排風風扇來將飼料槽內的空氣排出達到通風和除溼的效果。規格(如表 10)



圖 12 排風風扇

表 10 排風風扇規格

輸入電壓	7~13.2V
工作電壓	12V
轉速	2900rpm
重量	88g
噪音	33.5 分貝

11.溫度/濕度傳感器(DHT22) (如圖 13)

一款含有已校準數字信號輸出的溫濕度復合傳感器。使用了專用的數字模塊採集技術和溫濕度傳感技術，確保產品具有極高的可靠性與卓越的長期穩定性。規格(如表 11)



圖 13 溫度/濕度傳感器

表 11 溫度/濕度傳感器規格

工作電壓	3.0V~5.5V
輸出形式	數位訊號
溫度測量範圍	-40°C~80°C
溫度精準度	0.5°C
濕度測量範圍	0~100%RH
濕度精準度	2%RH

尺寸	28.2mm x 13.1mm x 5.5mm
重量	6g

12.ESP8266(如圖 14)

ESP8266 開發板用來當信息和信號傳遞的中繼站，協助 Arduino mega2560 和手機 app 溝通，當 app 傳送的信號送到 ESP8266 後，就會藉由邏輯準位調整器輸出到 Arduino mega2560 進行控制。規格(如表 12)

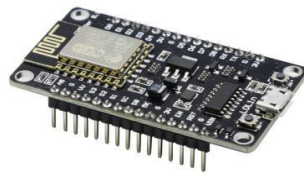


圖 14 ESP8266

表 12 ESP8266 規格

輸入電壓	4.5~10V
工作電壓	5V
接腳數	10(全部可以做為 PWM 引腳)
重量	5g

13.Raspberry Pi 3B+(如圖 15)

Raspberry Pi 開發板用來和它專屬的攝像頭連接，為手機提供即時影像，Raspberry Pi 和電腦一樣內建連網功能，所以不需要額外的網路模組。規格(如表 13)



圖 15 Raspberry Pi 3B+

表 13 Raspberry Pi 3B+規格

輸入電壓	7~24V
尺寸	86 x 56 x 17mm
重量	5g
記憶體	1GB
處理器	雙核 GPU 的 Broadcom 升級 BCM2837B0，ARM 的 Cortex-A8(A53)四核處理器
接口	USB 2.0 接口 x 4
插針	40 pin

14.Raspberry Pi 第二代 HD 攝影模組(如圖 16)

Raspberry Pi HD 攝影模組除了擁有高畫質外，還擁有紅外線攝影功能，可以讓我們就算在黑暗中也可以清楚的了解狀況，只是唯一的缺點是他只能連接於 Raspberry Pi 開發板。規格(如表 14)



圖 16 Raspberry Pi 第二代 HD 攝影模組

表 14 Raspberry Pi 第二代 HD 攝影模組規格

圖像傳感器	SONY IMX219
相片解析度	3280 x 2464
影片解析度	1080p30, 720p60 640x480p90
尺寸	25mm x 23mm x 9mm

(三)軟體

1.Autodesk Inventor Professional2020(如圖 17)

我們利用Inventor來繪製專題基礎草圖及製作製作動畫，包括爆炸圖及成品動作等等。機構圖(如圖 18)



圖 17 Autodesk Inventor Professional2020

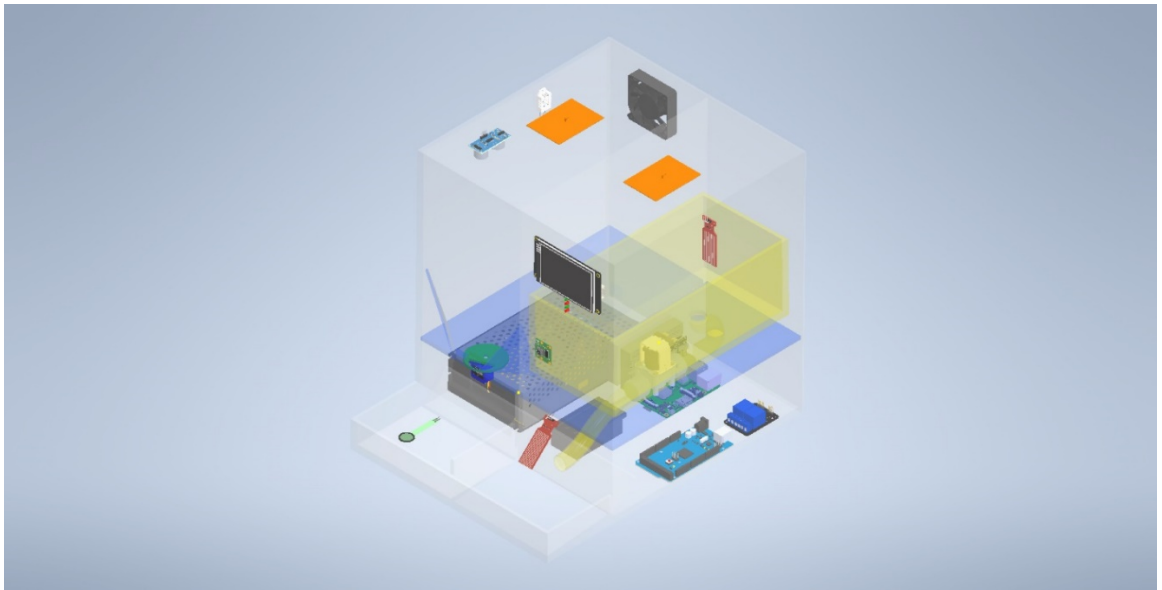


圖 18 機構圖

2.ArduinoIDE(如圖 19)

ArduinoIDE 是用來編寫程式和把程式燒錄到 arduino 開發板的電腦程式，因程式編寫容易和相對開發板較廉價因而讓我們決定使用。程式(如圖 20)



圖 19 ArduinoIDE


```

#include "Nextion.h" //dht 3.3v 4
#include "DHT.h" //relay 12
#include <Servo.h> //wts A0up A1down
#include <Ultrasonic.h>
#define DHTPIN 4 //gled 28w 30f
#define DHTTYPE DHT22
#define fsr_pin A3
//rled 24w 26f
const int fanPin = 22; //sg ora 10
const int ewaterPin = 28; //hmi trx 2
const int newwaterPin = 24;
const int efoodPin = 30;
const int nefoodPin = 26;
const int watergatePin = 12;
Ultrasonic ultrasonic(6, 7); //(trig,echo)

Servo servofeed;
Servo servowater;
DHT dht(DHTPIN, DHTTYPE);
// Declare your Nextion objects - Example (page id = 0, component id = 1, component name = "b0")
int hr,minute,sec,rhr,rhrmin,rhrsec,rmin,rminsec,rsec,serveamount,waterserveamount,hfood,hwater,thr,ohr,tmin,omin,tsec,osec,atmode,mnmo
NexButton atpage = NexButton(0, 2, "atpage");

```

圖 20 ArduinoIDE 程式

3.APP Inventor(如圖 21)

APP Inventor 為麻省理工學院所製作用來編寫 app 的軟體，因為他易於編寫且方便下載，所以我們決定用它來編寫我們的 app。



圖 21 APP Inventor

4.Python(如圖 23)

Python 支援多種程式設計範式，包括函數式、指令式、反射式、結構化和物件導向程式設計，所以我們使用他來撰寫樹梅派的的程式以控制攝像頭。程式(如圖 23 圖 24)



圖 22 Python

```
1 import io
2 import picamera
3 import logging
4 import socketserver
5 from threading import Condition
6 from http import server
7
8 PAGE="""\
9 <html>
10 <head>
11 <title>Raspberry Pi - Surveillance Camera</title>
12 </head>
13 <body>
14 <center><h1>Raspberry Pi - Surveillance Camera</h1></center>
15 <center></center>
16 </body>
17 </html>
18 """
19
20 class StreamingOutput(object):
21     def __init__(self):
22         self.frame = None
23         self.buffer = io.BytesIO()
24         self.condition = Condition()
25
26     def write(self, buf):
27         if buf.startswith(b'\xff\xd8'):
28             # New frame, copy the existing buffer's content and notify all
29             # clients it's available
30             self.buffer.truncate()
31             with self.condition:
32                 self.frame = self.buffer.getvalue()
33                 self.condition.notify_all()
34             self.buffer.seek(0)
35             return self.buffer.write(buf)
36
37 class StreamingHandler(server.BaseHTTPRequestHandler):
38     def do_GET(self):
39         if self.path == '/':
40             self.send_response(301)
41             self.send_header('Location', '/index.html')
42             self.end_headers()
43         elif self.path == '/index.html':
44             content = PAGE.encode('utf-8')
45             self.send_response(200)
46             self.send_header('Content-Type', 'text/html')
47             self.send_header('Content-Length', len(content))
```

圖 23 Python 程式

```
47 self.send_header('Content-Length', len(content))
48 self.end_headers()
49 self.wfile.write(content)
50 elif self.path == '/stream.mjpg':
51     self.send_response(200)
52     self.send_header('Age', 0)
53     self.send_header('Cache-Control', 'no-cache, private')
54     self.send_header('Pragma', 'no-cache')
55     self.send_header('Content-Type', 'multipart/x-mixed-replace; boundary=FRAME')
56     self.end_headers()
57     try:
58         while True:
59             with output.condition:
60                 output.condition.wait()
61                 frame = output.frame
62                 self.wfile.write(b'--FRAME\r\n')
63                 self.send_header('Content-Type', 'image/jpeg')
64                 self.send_header('Content-Length', len(frame))
65                 self.end_headers()
66                 self.wfile.write(frame)
67                 self.wfile.write(b'\r\n')
68     except Exception as e:
69         logging.warning(
70             'Removed streaming client %s: %s',
71             self.client_address, str(e))
72 else:
73     self.send_error(404)
74     self.end_headers()
75
76 class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
77     allow_reuse_address = True
78     daemon_threads = True
79
80 with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
81     output = StreamingOutput()
82     # Uncomment the next line to change your Pi's Camera rotation (in degrees)
83     #camera.rotation = 90
84     camera.start_recording(output, format='mjpeg')
85     try:
86         address = ('', 8080)
87         server = StreamingServer(address, StreamingHandler)
88         server.serve_forever()
89     finally:
90         camera.stop_recording()
```

圖 24 Python 程式

伍、研究結果

一、硬體結構

毛小孩管家大部分皆由壓克力組成，以下分為供給飼料、水以及遠端視訊三部分說明，說明如下：

(一)供給飼料部分(如圖 25)

我們製作飼料槽方便儲存飼料，搭配 SG90 進行控制，不僅能自訂所需要的時間進行出飼料，還能方便使用者進行控制。



圖 25 供給飼料

(二)供給水部分(如圖 26)

我們製作了水箱，搭配電磁水閥進行控制出水，將原本的所要用的虹吸出水，改為電動出水控制，大大增強了我們的控制性能。



圖 26 供給水

(三)遠端視訊(如圖 27)

在進行遠端監控時，我們怕程式所寫出的視訊畫質及連線會不好，所以我們選擇價錢較貴的樹梅派以及專屬攝像頭進行控制

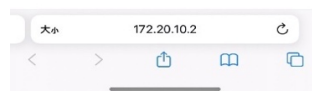
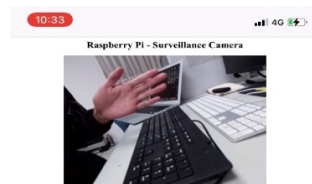


圖 27 遠端視訊

二、操作流程

流程主要有機構和手機兩部份，且兩部份各分為四個小部份，機構部份有主頁面(如圖 28)，自動(如圖 29)，手動(如圖 30)，飼料量調整(如圖 31)。手機部份也有主頁面(如圖 32)，自動(如圖 33)，手動(如圖 34)，飼料量調整(如圖 35)。機

構部份利用 hmi 來對整個系統進行控制，定時部份設定好時間後，系統就會在設定的時間到進行供食，而手動則是可自行控制兩個閥門的開關來進行供食，雖然必需要頻繁的在 hmi 前控制，但可以較精準的控制量，手機部份是利用 app，除了上述的功能外，還有視訊功能使主人可以看到即時影像。

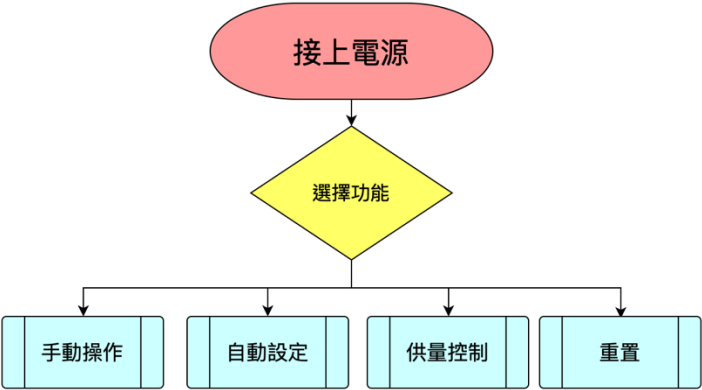


圖 28 機構主頁面

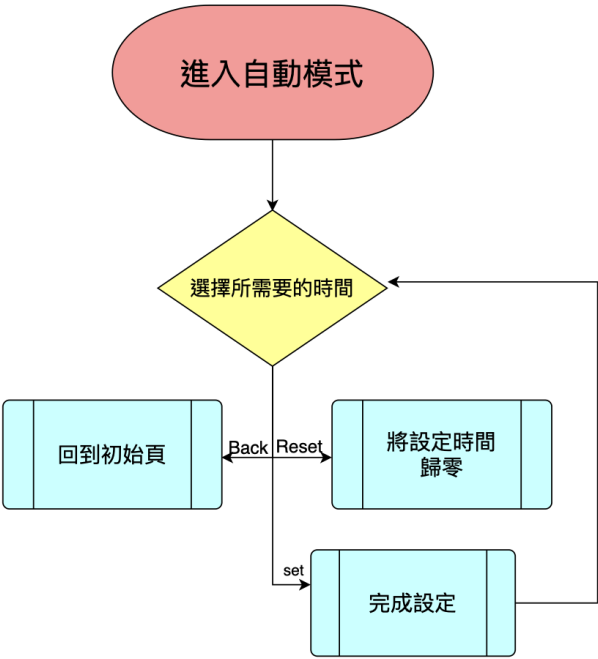


圖 29 機構自動

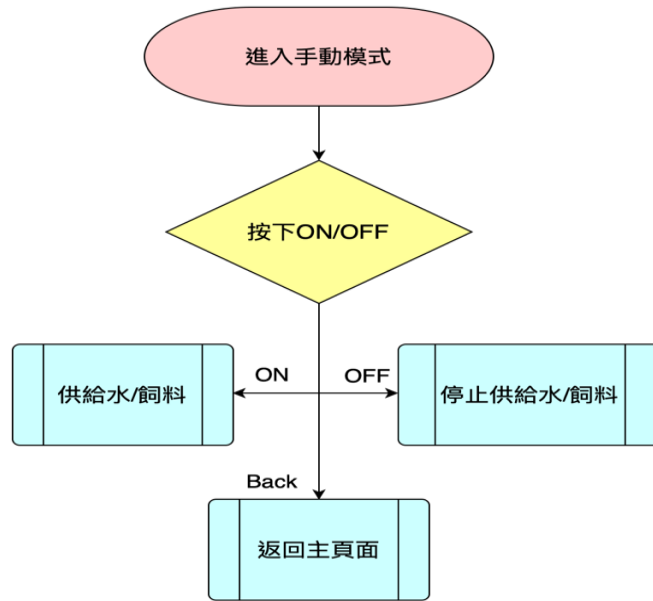


圖 30 機構手動

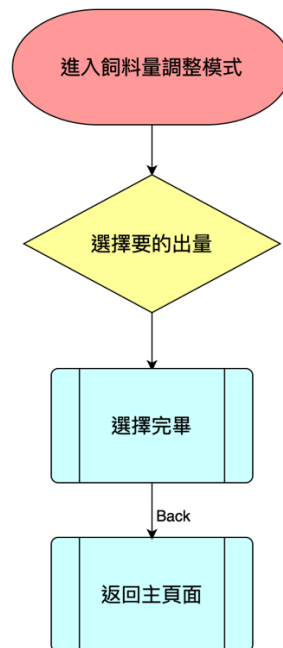


圖 31 機構飼料量調整

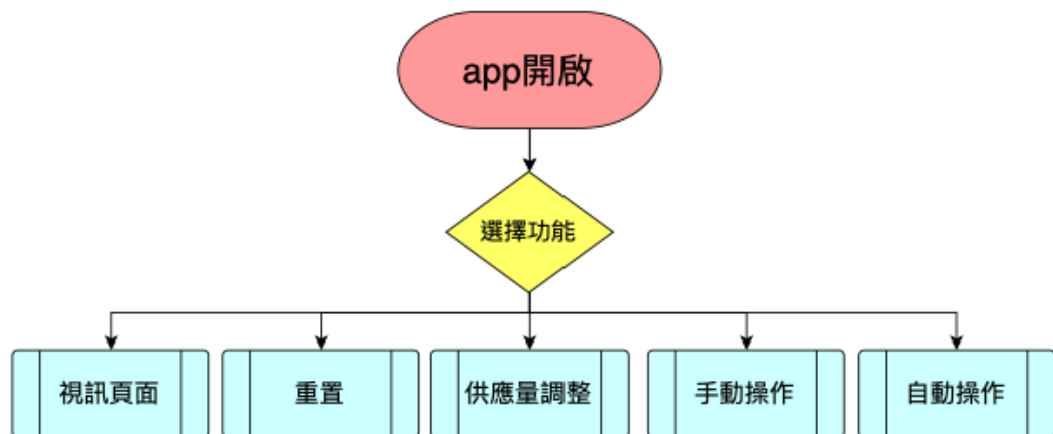


圖 32 手機主頁面

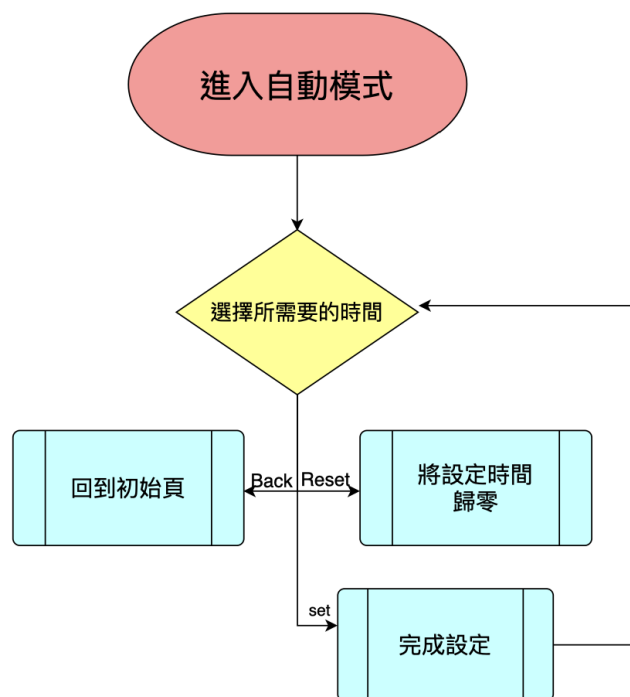


圖 33 手機自動

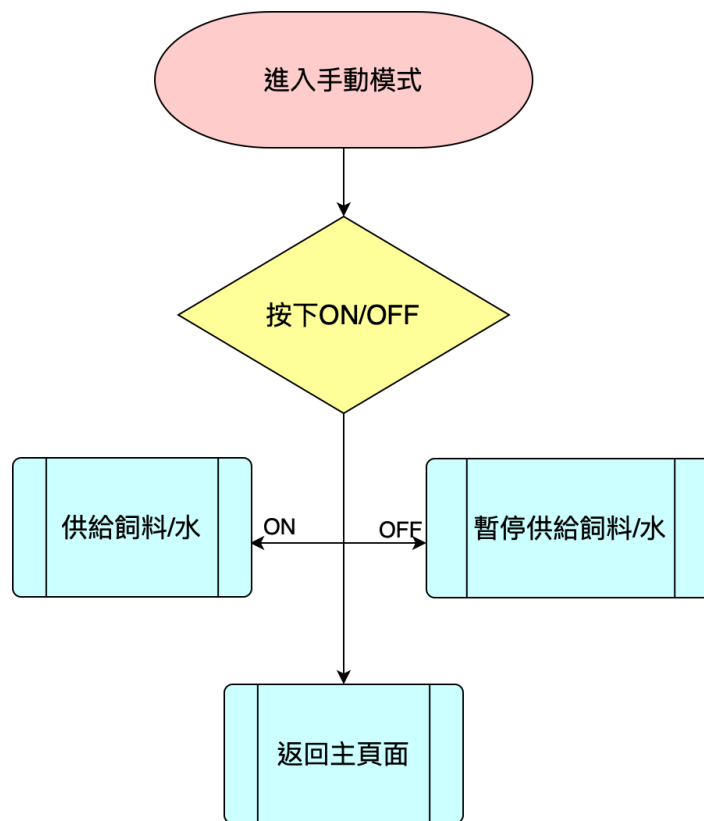


圖 34 手機手動

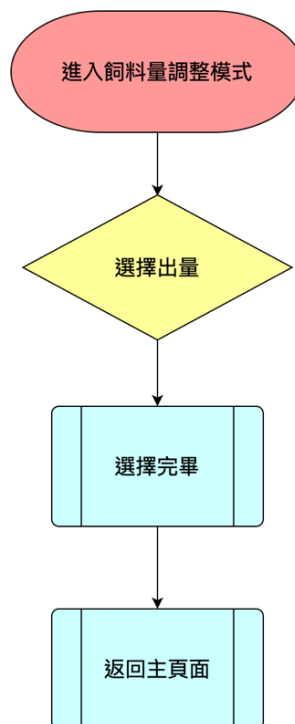


圖 35 手機飼料量調整

陸、討論

一、繪圖選擇

一開始在設計機構草圖時，我們是使用 Tinkercad，但後來發現遇到問題時學校能給的資源較少，因此後來我們採用跨科學習的 Inventor。雖然 Inventor 的繪製相較於 Tinkercad 較為複雜，但遇到問題時能夠隨時向製圖科裡的老師們求助。

二、飼料槽出飼料

前期製作飼料槽時，使用的是 SG90 以及飼料轉盤，在實際測試時飼料轉盤會出現卡料以及馬達無法支撐飼料重量得情況，所以我們在馬達的四周圍進行加強結構讓馬達足以支撐飼料的重量，在轉盤的部分，我們將轉盤貼合飼料槽避免了卡料的情況操控方法。

三、訊息傳輸

設定 line notify 時，一開始是使用 IFTTT 系統，但是因為發現傳輸和設定困難，而且訊息也必需很固定，後來我們發現了 arduino 專屬的 line notify 程式庫，它除了好設定且傳輸快外，而且通知的訊息也較自由，所以我們最後決定放棄 IFTTT 直接使用這個程式庫。

柒、結論

首先毛小孩管家和我們起初的外觀想法完全不一樣，原先我們想使用 3D 列印製作出可防水且堅固的外觀，但想到 3D 列印的材料費不僅昂貴，而且在製作時非常浪費時間，所以我們才將原料改為壓克力，不僅方便切割，而且防水又堅硬。在控制方法時則是原本只想使用手機 APP 作為主要控制的方法，但想說如果使用者目前沒有手機時該如何處理，所以我們在機構上加上了 HMI 觸控面板方便使用者在沒有手機時也能進行設定以及控制。本次專題的難點是在，如何寫出樹莓派的控制功能，這個東西在實習課程中老師並沒有教，所以我們必須自己查詢相關文件，自己學習製作出這次的樹莓派攝像頭。在這次的專題中，我們覺得不只是需要製作各種的程式以及電路板，在組員之間的溝通也相對重要，若是一個人無法在即時的時間內完成所需要的程式或者結構，那必定會拖累其他的隊員，而且在這次的合作中我了解到了，其實組員之間不應該吝嗇於自己該做什麼，或者只做什麼，應該互相幫忙，互相理解，才能完成。

捌、參考資料及其他

2021/11/20。結構參考。取自

<https://www.khpetbuy.com/products/p56221d1i00ni2423>

2021/11/20。樹莓派攝像頭程式參考。取自

<https://www.youtube.com/watch?v=nTVI9YBZoFg>

2021/11/20。Python。取自

<https://zh.m.wikipedia.org/wiki/Python>

2021/11/20。攝像頭程式。取自

https://github.com/RuiSantosdotme/Random-Nerd-Tutorials/blob/master/Projects/rpi_camera_surveillance_system.py

2021/11/20。零件購買。取自

<https://www.jin-hua.com.tw>

2021/11/20。APP Inventor 參考資料。取自

<http://www.appinventor.tw/ai2>

2021/11/20。APP Inventor 視訊參考資料。取自

http://www.appinventor.tw/ntut_app/feeding-machaine

2021/11/20。Python 初學資料。取自

<https://youtu.be/zdMUJJKFdsU>

2021/11/20。HMI 面板程式。取自

<http://wiki.tjc1688.com/doku.php?id=2.%E4%B8%8A%E4%BD%8D%E8%BD%AF%E4%BB%B6%E4%B8%8B%E8%BD%BD:1.%E6%9C%80%E6%96%B0%E7%89%88%E6%9C%AC%E4%B8%8B%E8%BD%BD>

2021/11/20。Word 製作教學。取自

<https://youtu.be/X6xT-foxEmM>