

臺北市立大安高級工業職業學校

電子科

專題報告

個人智慧藥盒

Personal Smart Medicine Box

學生組長：陳冠霖

組員：陳柏諺

組員：黃健嘉

組員：鄭良羴

指導老師：簡靖哲

中華民國 111 年 1 月

## 中文摘要

近幾年來，在全球新冠肺炎疫情的壟罩下，病患持續增加，但醫護人員卻持續短缺，智慧醫療將是未來發展的趨勢，如何減少醫護人員與病患的交流以降低傳染風險將是一大課題。一般慢性病患者需要依照早、中、晚、睡前服用藥物，藥盒成為醫療照顧的一部份，傳統塑膠藥盒無法提醒吃藥與紀錄食用時間，若忘記吃藥，也無法及時讓家人了解狀況。經搜尋相關內容及其應用後，決定本專題以智慧醫療產品中的智慧藥盒做為主題。

透過物聯網的無線傳輸技術，病患家人可用 app 設定吃藥時間，藥盒將在吃藥時間發送訊息通知與聲響提醒，病患可利用手部感測接住掉落藥物，在病患取藥後，家人可收到取藥通知，也可利用 app 了解藥盒內的剩餘藥量情況，在不足藥量時給予通知。

**關鍵字：**智慧醫療，慢性病患者，藥盒，arduino，物聯網

## 英文摘要

Due to the pandemic medical personnel are often shorthanded, and how to reduce their risk when interacting with patients is also a problem to solve, but it looks like technology is the way.

Most people with chronic diseases usually have to consume medicine regularly (morning, noon, night, before bed), so a pill box has become a part of the patients' life. But traditional medicine box can't notify the patient to take the medications or tell his/her family whether they took the medicine or not.

Through the wireless internet transferring the data, patient's family can set the time to consume medicine with the App. The product will send a message with an alarm going off to notify the user to have the medicine. The patient can get the medicine through an IR sensor. After the patient gets the medicine, his/her family can also get the notification of him/her getting the medicine. With the App, we can know the amount of medicine that is left, and can get the notification when there's no medicine left in the product.

**Keywords : eHealth , chronic patients, medicine box, Arduino, Internet of things**

# 目錄

中文摘要.....	I
[圖]英文摘要.....	II
[圖]目錄.....	III
[圖]表目錄.....	VI
第 1 章 .....	前言
1 .....	
1-1 專題製作背景及目的.....	1
1-1-1 .....	製作背景
.....	1
1-1-2 .....	製作目的
.....	1
1-2 預期成果.....	1
第 2 章 .....	[圖]理論探討
2 .....	
2-1 軟體.....	2
2-1-1 .....	Arduino
.....	2
2-1-2 .....	MQTT
.....	2
2-1-3 .....	APP[圖]inventor
.....	3
2-1-4 .....	[圖]Onshape
.....	4
2-1-5 .....	Altium Designer
.....	4
2-2 硬體.....	4
2-2-1 .....	Arduino Mega2560
.....	4
2-2-2 .....	ESP32

.....	5
2-2-3 .....	3D 列印機
.....	5
2-2-4 .....	雷射切割機
.....	6
2-2-5 .....	步進馬達
.....	7
第 3 章 .....	專題設計
8	
3-1 系統架構圖.....	8
3-2 甘特圖.....	8
3-3 專題系統架構.....	9
3-3-1 .....	Arduino 程式
.....	9
3-3-2 .....	APP 程式
.....	9
3-3-3 .....	機構
.....	9
3-4 程式架構.....	9
3-4-1 .....	Esp32 程式
.....	9
3-4-2 .....	Arduino 程式
.....	25
3-4-3 .....	APP 程式
.....	30
3-5 機構.....	35
3-5-1 .....	藥盒設計
.....	35
第 4 章 .....	專題成果
37	
4-1 操作流程圖.....	37

4-2 前置設定流程.....	38
4-2-1 ..... 放藥與剩餘藥量	38
4-2-2 ..... 設定領藥時間	39
4-3 領藥過程.....	40
4-3-1 ..... 領藥通知	40
4-3-2 ..... 領藥動作	41
第 5 章 ..... [OBJ]結論與建議	43
5-1 結論.....	43
5-2 建議.....	43
[OBJ]參考文獻.....	44
[OBJ]附錄.....	45
附錄一 設備清單.....	45
附錄二 材料清單.....	46
附錄三 研究成員簡歷.....	47

## 圖目錄

圖 1	Arduino logo .....	2
圖 2	MQTT 架構圖 .....	3
圖 3	MQTT logo .....	3
圖 4	APP inventor logo .....	3
圖 5	onshape logo .....	4
圖 6	Altium Designer logo .....	4
圖 7	Arduino Mega2560 .....	5
圖 8	ESP32 .....	5
圖 9	3D 列印機 .....	6
圖 10	雷射切割機 .....	6
圖 11	步進馬達 .....	7
圖 12	系統架構圖 .....	8
圖 13	甘特圖 .....	8
圖 14	外觀及內部結構圖 .....	36
圖 15	操作流程圖 .....	37
圖 16	補藥通知 .....	38
圖 17	AI2 UI .....	39
圖 18	設定時間 .....	40
圖 19	手機領藥通知 .....	41
圖 20	手部深入掉藥 .....	42
圖 21	手機已取藥通 .....	42

# 第1章 前言

## 1-1 專題製作背景及目的

### 1-1-1 製作背景

在 2019 年新冠肺炎爆發至全球，確診病患越來越多，如何減少人與人的見面次數以降低確診病例便是一大課題，家中的病患因慢性病需要長期吃藥，但病患可能因年紀大，記憶力有點衰退，常常忘記自己是否已經吃過藥，家人又因為工作忙碌的關係，無法時常陪在長輩身邊和提醒他吃藥，導致病患的身體時常處於不穩定的狀態。

### 1-1-2 製作目的

為了解決病患忘記吃藥的問題，智慧醫療也逐漸普及，所以我們選擇做智慧藥盒提醒長輩吃藥，家人也能隨時了解長輩吃藥的情況。

## 1-2 預期成果

本專題預期完成一個智慧藥盒，能夠利用 APP 完成領藥時間控制、剩餘藥量顯示、WIFI 連接及接收通知訊息，外殼可看見 LCD 顯示目前時間，到設定的吃藥時間可發送訊息通知與聲響提醒，當手伸進出藥口時，藥會掉到手中且發送取藥通知。



## 第2章 理論探討

### 2-1 軟體

#### 2-1-1 Arduino

Arduino 專案始於 2003 年，作為義大利伊夫雷亞地區伊夫雷亞互動設計研究所的學生專案，目的是為新手提供一種低成本且簡單的方法，建立傳感器與環境相互作用的裝置執行器。適用於初學者愛好者。Arduino 是一個開放原始碼的單晶片微控制器，它使用了 Atmel AVR 單片機，採用了開放原始碼的軟硬體平台，建構於簡易輸出/輸入（simple I/O）介面板，並且具有使用類似 Java、C 語言的 Processing/Wiring 開發環境，也是一種開放授權的互動環境開發技術，像是冷氣的恆溫裝置，是使用感測器偵測環境溫度，進行室內溫度的自動調節，還有汽車使用的倒車雷達，過於靠近物體便會發出聲音警告駕駛者。



圖 1 Arduino logo

#### 2-1-2 MQTT

MQTT(Message Queuing Telemetry Transport)是 IBM 為了物聯網(IoT, Internet of Things)所開發的一通訊架構，此架構建構於 TCP/IP 之上，並特別針對物聯網的特性，提供輕量、且針對不穩定網路的通訊架構。MQTT 是基於 pub/sub 的架構實現，在其網路中，有一個 broker 負責作為資訊的中介層，而其他裝置可以藉由發表(publish)一個主題(topic)，傳送資訊至 broker，或是藉由定閱(subscribe)一主題，從 broker 中取得資訊。使用 MQTT 的行動式應用程式呼叫 MQTT 程式庫傳送及接收訊息。透過 MQTT 傳訊伺服器交換訊息，MQTT 用戶端及伺服器可針對行動式應用程式可靠地處理遞送訊息的複雜性，並保持低成本地進行網路管理。

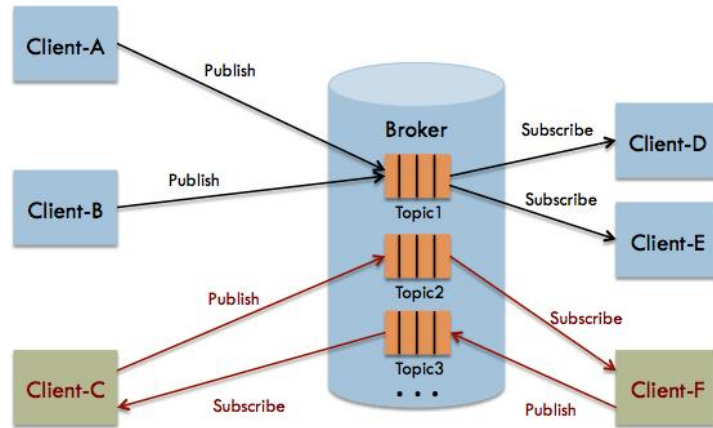


圖 2 MQTT 架構圖



圖 3 MQTT logo

### 2-1-3 APP inventor

App Inventor 是一個完全線上開發的 Android 程式環境，拋棄複雜的程式碼而使用樂高積木式的堆疊法來完成 Android 程式，適合無 Java 基礎的初學朋友，操作概念很類似 scratch，且全雲端，所有作業都在瀏覽器完成，除此之外它也正式支援樂高 NXT 機器人，因為對於想要用手機控制機器人的使用者而言，他們不大需要太華麗的介面，只要使用基本元件例如按鈕、文字輸入輸出即可。



圖 4 APP inventor logo

#### 2-1-4 Onshape

Onshape 是一套線上 3D 繪圖軟體，介面簡單，所以進入門檻低，而且免費跟方便，不用對檔案做其它的處理。Onshape 是用規劃圖來建立 2D 的概念，而不是單一斷面，表示使用者可以在一張 2D 圖上看出各個面向的設計，Onshape 也是一個協同設計平台，主要的概念是互相支援協助，大家可在平台上提供自己的想法或專長，使概念更完整，這部份 Onshape 甚至提供可在平板或手機上使用。



圖 5 onshape logo

#### 2-1-5 Altium Designer

Altium Designer 是原 Protel 軟體開發商 Altium 公司推出的一體化的電子產品開發系統，主要功能有原理圖設計、印刷電路板設計、FPGA 的開發、嵌入式開發、3D PCB 設計以及封裝庫設計，主要運行在 Windows 作業系統。這套軟體通過把原理圖設計、電路仿真、PCB 繪製編輯、拓撲邏輯自動布線、信號完整性輸出和設計輸出等技術的完美融合，使設計者可以輕鬆進行設計。



圖 6 Altium Designer logo

## 2-2 硬體

#### 2-2-1 Arduino Mega2560

Arduino Mega 2560 微控制板是一款基於 ATmega2560 的微控制器版。他

有 54 個數字輸入/輸出引腳(其中 15 個可用作 PMW 輸出)、16 個模擬輸入、4 個 UART(硬件串行端口)、一個 16MHz 晶體振盪器、1 個 USB 連接、1 個電源插座、1 個 ICSP 頭和 1 個復位按鈕。它包含了支持微控制器的一切；只預通過 USB 電纜將其連至計算機或者通過 AC-DC 適配器或電池為其供電即可開始。



圖 7 Arduino Mega2560

### 2-2-2 ESP32

ESP32 是一系列由中國上海樂鑫科技(Espressif)推出的晶片、模組、開發板，採用 Tensilica Xtensa LX6 微處理器，包括雙核心和單核變體，支援 Arduino 開放性架構，內建 WiFi 及低功率藍芽 BLE，可用腳位也擴增到約 26 根，大部分腳位同時支援類比及數位，內建霍爾、溫度、觸控感測器，主頻高達 240MHz 的雙核心 CPU，傳輸界面 SPI、I2C、UART 等強大的功能。



圖 8 ESP32

### 2-2-3 3D 列印機

3D 列印又稱為積層製造或加法製造，以 3D 圖檔為基礎，透過多種方式達到堆疊列印成型，且能協助企業縮短產品開發流程，降低最終大量生產成本，主要有三種原理，Ployjet 聚合物噴射成型、FDM 熱熔擠製成型與 LOM 層壓物

體製造成型，以我們使用的 FDM 熱熔擠製成型方式來說，其 3D 列印原理是像擠牙膏方式，將線狀材料透過加熱噴頭擠出在預定路徑與列印範圍內層層堆疊、冷卻並固化完成，且材料多具耐用、堅固的工程性能，如 ABS、PC、PLA 等。



圖 9 3D 列印機

#### 2-2-4 雷射切割機

雷射切割 (Laser Cutting) 這項技術，是透過一束經過聚焦後的高功率雷射光來燒熔、燒蝕物體，以對物體進行切割或雕刻。因為雷射光束的能量與密度相當高，而能在材料上產生細緻的缺口，所謂的雷射切割機其實就是一種加熱切割的方式，常用於各種生產、加工、學術領域。雷射主要有氣體、固體、半導體三種雷射源及打標、切割、焊接、除鏽、雕刻、美容六種功能。



圖 10 雷射切割機

### 2-2-5 步進馬達

步進馬達（Step motor）是直流無刷電動機的一種，為具有如齒輪狀突起（小齒）相契合的定子和轉子，可藉由切換流向定子線圈中的電流，以一定角度逐步轉動的馬達。步進馬達的特徵是採用開迴路控制（Open-loop control）處理，不需要運轉量感測器（sensor）或編碼器，且切換電流觸發器的是脈衝信號，不需要位置檢出和速度檢出的回授裝置，所以步進電機可正確地依比例隨脈衝信號而轉動，因此達成精確的位置和速度控制，且穩定性佳。



圖 11 步進馬達

## 第3章 專題設計

### 3-1 系統架構圖

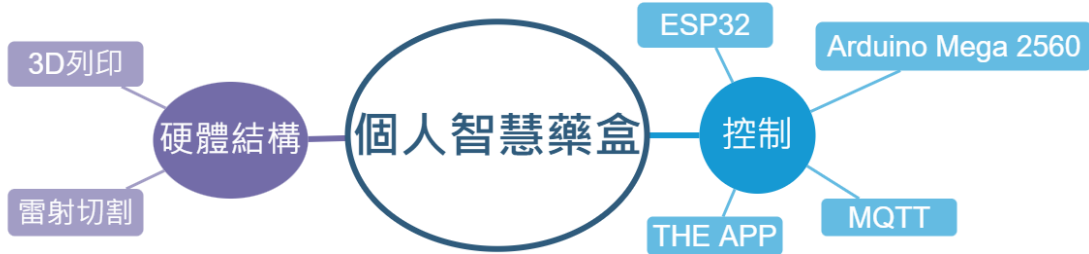


圖 12 系統架構圖

### 3-2 甘特圖

週次	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	負責成員	
外觀與內部設計	■	■																		全員
3D 列印			■	■	■	■	■	■	■	■	■	■	■	■	■					25,32
程式設計			■	■	■	■	■	■	■	■	■	■	■	■	■					32,36
物聯網製作			■	■	■	■	■	■	■	■	■									36,32
Line notify 連接						■	■	■	■											29
APP INVENTOR 製作	■	■	■	■									■	■	■	■	■			36,29
調整外觀										■	■	■	■	■						全員
功能測試											■	■	■	■						全員
整合														■	■	■	■	■		全員
製作期末報告														■	■	■	■	■		全員
口頭進度報告						■		■		■				■		■		■		全員
預定進度	10	20	25	25	30	35	40	50	60	65	70	80	85	90	93	95	98	100	累積百分比%	

圖 13 甘特圖

### 3-3 專題系統架構

我們將專題分成 Arduino 程式、app 程式、機構這三個部份。

#### 3-3-1 Arduino 程式

負責處理藥倉公轉、藥倉內隔板轉動、LCD 顯示、蜂鳴器聲音控制、傳送通知訊息、按鈕控制及紅外線感測。當領藥時間一到，則發出提醒聲響，LCD 顯示目前時間，按下按鈕後 app 上的剩餘藥量則顯示為剩餘七天份，紅外線感測到手後，藥倉轉動使藥物掉落且發送已取藥通知，藥量也隨之減一。

#### 3-3-2 APP 程式

負責處理領藥時間設定、發送領藥通知、時間及剩餘藥量顯示，可利用 APP 完成以下動作：了解當下時間與剩餘藥量，當設定完領藥時間後，MQTT 將記錄領藥時間，領藥時間一到，則發送領藥通知。

#### 3-3-3 機構

使用 onshape 來設計藥盒外殼、藥倉與藥倉隔板，外殼分為四部份，分為三個無裝飾物的外殼與一個放置 LCD 與按鈕並且有出藥口的外殼，內部則有四個藥倉，分為早、中、晚與睡前，藥倉內用隔板分成八格，分為出藥口與七天份。

使用雷射切割來完成藥盒上蓋，選擇透明壓克力板作為材料，以方便查看內部運作情況。

### 3-4 程式架構

#### 3-4-1 Esp32 程式

**Esp32 會用到的函式庫：**

```
// http://dl.espressif.com/dl/package_esp32_index.json prefer direction
// download PubSubClient / line notify
#include <WiFi.h> //Wifi library
#include <PubSubClient.h> //library for MQTT server
#include "time.h" //esp32's default RTC library
#include <LiquidCrystal.h> //library for LCD fuction
#include <WiFiClient.h> //Wifi library2
#include <TridentTD_LineNotify.h> //library for Line Notify
fuction
```



```

#define LINE_TOKEN
"0XD31b2oE6QG6qtkFhD9IO2hRhSWY04D12teQbUqN80" //line group's token
//NodeMCU-32S
//接到 ESP32 RXTX
//WiFi

```

### 用來儲存函示庫資料的全域變數:

```

unsigned long timenow = 0; //a variable to save
the current time
const char *ssid = "Justin"; // Enter your WiFi name
const char *password = "36285251"; // Enter WiFi password
const int e18_sensor = 36; //紅外線,5V 旁的 GND 是 3.3V, pin36 = SUN
// MQTT Broker
const char *mqtt_broker = "broker.emqx.io"; //MQTT server's
direction
const char *topic0 = "medicineproject2021daan"; //Seperate to 5
topic, topic0 is to let esp32 save the time to have medicine
const char *topic1 = "medicineproject2021daan1"; //topic1 to set how
much left is in morning's medicines
const char *topic2 = "medicineproject2021daan2"; //noon's medicines
const char *topic3 = "medicineproject2021daan3"; //night's medicines
const char *topic4 = "medicineproject2021daan5"; //before sleep's
medicines
const char *mqtt_username = "ninjasalad"; //username for
MQTT client
const char *mqtt_password = "medicineproject"; //user's password
const int mqtt_port = 1883; //1883 means a
public port
const char* ntpServer = "pool.ntp.org"; //a server to get the
RTC's data(Real Time Clock)
const long gmtOffset_sec = 28800;
const int daylightOffset_sec = 3600;
int pushButton = 34; // 按鈕 腳位 7

```

```
int buttonState = 0; //假設為按鈕狀態，先設定 0
int beforeState = 0; //假設之前的狀態
```

### **LCD 腳位:**

```
LiquidCrystal My_LCD(15, 4, 5, 18, 19, 21);
// (RS, EN, D4, D5, D6, D7)
```

### **載入網路以及網路伺服器客戶端:**

```
WiFiClient espClient;
PubSubClient client(espClient);
```

### **拿來運算程式的變數:**

```
int s = 0;
int s1, s2;
String hour1, hour2, text, minute1, minute2, message, Top2,
Time0, Time1, Time2, Time3, hourcombine, minutecombine, remindhour,
remindtime0, remindtime1, remindtime2, remindtime3,
currenttime, currenthour, currentminute, currentsecond, endhour, endtime0, endtime1,
endtime2, endtime3;
String second = "00";
int a, b, c, d, e, i, a1, b1, c1, d1; //a1~d1 is for sensor command to morning ~ sleep
(a1= 1:accept the command, else)
int state = 1;
struct tm timeinfo;
int t1, t2, t3, t4, t5, t6, t7, t8;
```

### **利用 RTC 功能獲取現在時間以及將之顯示在 LCD 上:**

```
void printLocalTime()
{
    struct tm timeinfo;
    if(!getLocalTime(&timeinfo)){
        return;
    }
}
```

```

}
//Serial.println(&timeinfo, "%A, %B %d %Y %H:%M:%S");
currenthour = timeinfo.tm_hour;
currentminute = timeinfo.tm_min;
currentsecond = timeinfo.tm_sec;
endhour = timeinfo.tm_sec + 20;
remindhour = timeinfo.tm_sec + 10;
if ((int)timeinfo.tm_hour <=9){
    currenthour = "0" + currenthour;
}
if ((int)timeinfo.tm_min <=9){
    currentminute = "0" + currentminute;
}
if ((int)timeinfo.tm_sec <=9){
    currentsecond = "0" + currentsecond;
}
currenttime = currenthour + currentminute + currentsecond;
//Serial.println(&timeinfo,"%T");
My_LCD.setCursor(0,0);
My_LCD.print(&timeinfo,"%F"           );
My_LCD.setCursor(0,1);
My_LCD.print(&timeinfo,"%T"           );
}

```

**主程式-設定鮑率、LCD 設定、紅外線感測器腳位、按鈕腳位以及  
網路和 MQTT 伺服器連接:**

```

void setup() {
Serial.begin(115200);
pinMode(pushButton, INPUT); //將 34 腳位 定為輸入
pinMode (e18_sensor, INPUT);
My_LCD.begin(16, 2);
My_LCD.clear();
//Serial2.begin(9600,SERIAL_8N1, RXp2, TXp2);

```

```

// connecting to a WiFi network
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
printLocalTime();
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    //Serial.println("Connecting to WiFi..");
}
// Serial.println("Connected to the WiFi network");
//connecting to a mqtt broker
client.setServer(mqtt_broker, mqtt_port);
client.setCallback(callback);
while (!client.connected()) {
    String client_id = "esp32-client-";
    client_id += String(WiFi.macAddress());
    //Serial.printf("The client %s connects to the public mqtt broker\n",
client_id.c_str());
    if (client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
        //Serial.println("Public emqx mqtt broker connected");
    } else {
        //Serial.print("failed with state ");
        //Serial.print(client.state());
        delay(2000);
    }
}
// client.publish(topic, "string");
client.publish(topic0, "connected!");
client.subscribe(topic0);
}

```

**將從 APP 傳送過來的時間儲存起來：**

```

void callback(char *topic0, byte *payload, unsigned int length) {

```

```

for (int i = 0; i < 1; i++){          //Saving the first "string"
    Top2 = (char) payload[0];
}
for (int i = 1; i < length; i++){    //sending the time *the String after the third
one
    hour1 = (char) payload[1];
    hour2 = (char) payload[2];
    minute1 = (char) payload[4];
    minute2 = (char) payload[5];
}
message = hour1 + hour2 + minute1 + minute2 + second; // no :

```

判斷儲存起來的時間是早上、中午、晚上或睡前並分別存起來:

```

if (Top2 == "1"){                    //checking if it's time for morning
    Time0 = message;
}
if (Top2 == "2"){                    //checking if it's time for noon
    Time1 = message;
}
if (Top2 == "3"){                    //checking if it's time for night
    Time2 = message;
}
if (Top2 == "4"){                    //checking if it's time for sleep
    Time3 = message;
}
}
}

```

如果到了吃藥時間，會傳送 LINE 訊息，以及傳送指令給

**Arduino**，使 **Arduino** 讓蜂鳴器動作，發出提醒聲響。半小時後再傳送一樣的訊息以及聲響提醒病患，一個小時吃藥時間過期，不再接收吃藥要求:

```

void distinguish(){

```

```

if (Time0 == currenttime){
    LINE.setToken(LINE_TOKEN);
    LINE.notify("\n 早安~該吃藥囉~");
    Serial.print("buzzer");          //a1 = make a command for buzzer
    for(i=0;i<=0;i++){
        endtime0 = currenthour + currentminute + endhour;
        remindtime0 = currenthour + currentminute + remindhour;
    }
    a = 1;
    b = 1;
    a1 = 1;
    s2 = 0;
    s = 0;
    s1 = 0;
}
if (currenttime == remindtime0 && s == 0 && s1 == 0) {
    Serial.print("buzzer");
    LINE.setToken(LINE_TOKEN);
    LINE.notify("\n 請提醒病患吃藥");
    s1 = 1;
}
if (currenttime == endtime0){
    a = 2;
    b = 0;
    a1 = 0;
    s = 0;
    s1 = 0;
}
if ((currenttime == endtime0) &&(s == 0) && (s2 == 0)){
    a = 2;
    b = 0;
    a1 = 0;
    Serial.print("center");
}

```

```

    s2 = 1;
    LINE.setToken(LINE_TOKEN);
    LINE.notify("\n 錯過一次用藥，請提醒下次記得吃");
}
if (Time1 == currenttime){
    LINE.setToken(LINE_TOKEN);
    LINE.notify("\n 午安~該吃藥囉~");
    Serial.print("buzzer");
    for(i=0;i<=0;i++){
        endtime1 = currenthour + currentminute + endhour;
        remindtime1 = currenthour + currentminute + remindhour;
    }
    a = 1;
    b = 1;
    b1 = 1;
    s2 = 0;
    s = 0;
    s1 = 0;
}
if (currenttime == remindtime1 && s == 0 && s1 == 0) {
    Serial.print("buzzer");
    LINE.setToken(LINE_TOKEN);
    LINE.notify("\n 請提醒病患吃藥");
    s1 = 1;
}
if (currenttime == endtime1){
    a = 2;
    b = 0;
    b1 = 0;
    s = 0;
    s1 = 0;
}
if ((currenttime == endtime1) &&(s == 0) && (s2 == 0)){

```

```

    a = 2;
    b = 0;
    b1 = 0;
    Serial.print("center");
    s2 = 1;
    LINE.setToken(LINE_TOKEN);
    LINE.notify("\n 錯過一次用藥，請提醒下次記得吃");
}
if (Time2 == currenttime){
    LINE.setToken(LINE_TOKEN);
    LINE.notify("\n 晚安~該吃藥囉~");
    Serial.print("buzzer");
    for(i=0;i<=0;i++){
        endtime2 = currenthour + currentminute + endhour;
        remindtime2 = currenthour + currentminute + remindhour;
    }
    a = 1;
    b = 1;
    c1 = 1;
    s2 = 0;
    s = 0;
    s1 = 0;
}
if (currenttime == remindtime2 && s == 0 && s1 == 0) {
    Serial.print("buzzer");
    LINE.setToken(LINE_TOKEN);
    LINE.notify("\n 請提醒病患吃藥");
    s1 = 1;
}
if (currenttime == endtime2){
    a = 2;
    b = 0;
    c1 = 0;

```



```

    s = 0;
    s1 = 0;
}
if ((currenttime == endtime2) &&(s == 0) && (s2 == 0)){
    a = 2;
    b = 0;
    c1 = 0;
    Serial.print("center");
    s2 = 1;
    LINE.setToken(LINE_TOKEN);
    LINE.notify("\n 錯過一次用藥，請提醒下次記得吃");
}
if (Time3 == currenttime){
    LINE.setToken(LINE_TOKEN);
    LINE.notify("\n 睡覺前記得吃藥喔!");
    Serial.print("buzzer");
    for(i=0;i<=0;i++){
        endtime3 = currenthour + currentminute + endhour;
        remindtime3 = currenthour + currentminute + remindhour;
    }
    a = 1;
    b = 1;
    d1 = 1;
    s2 = 0;
    s = 0;
    s1 = 0;
}
if (currenttime == endtime3){
    a = 2;
    b = 0;
    d1 = 0;
    s = 0;
    s1 = 0;
}

```

```

}
if (currenttime == remindtime3 && s == 0 && s1 == 0) {
  Serial.print("buzzer");
  LINE.setToken(LINE_TOKEN);
  LINE.notify("\n 請提醒病患吃藥");
  s1 = 1;
}
if ((currenttime == endtime3) &&(s == 0) && (s2 == 0)){
  a = 2;
  b = 0;
  d1 = 0;
  Serial.print("center");
  s2 = 1;
  LINE.setToken(LINE_TOKEN);
  LINE.notify("\n 錯過一次用藥，請提醒下次記得吃");
}
}
}

```

**若在吃藥時段內去領藥，紅外線會感測到，且發送指令給 Arduino 使藥盒運作並傳送 LINE 告知家屬病患已吃藥：**

```

void sensor(){
  if (a == 1 && a < 2){
    if (state == 0 && b == 1 && a1 == 1){
      Serial.print("a1");
      LINE.setToken(LINE_TOKEN);
      LINE.notify("\n 藥物已被領取!"); //早
    }
  }
  a1 = 0; //check if it's in time
  t1 = t1 + 1; //see the left medicine
  t5 = 1; //stop looping the mqtt message
  s = 1;
  s2 = 1;
}

```

```

    }
    if (state == 0 && b == 1 && b1 == 1){
        Serial.print("b1");
        LINE.setToken(LINE_TOKEN);
        LINE.notify("\n 藥物已被領取!");
//中午
        b1 = 0;
        t2 = t2 + 1;
        t6 = 1;
        s = 1;
        s2 = 1;
    }
    if (state == 0 && b == 1 && c1 == 1){
        Serial.print("c1");
        LINE.setToken(LINE_TOKEN);
        LINE.notify("\n 藥物已被領取!");
        c1 = 0;
        t3 = t3 + 1;
        t7 = 1;
        s = 1;
        s2 = 1;
    }
    if (state == 0 && b == 1 && d1 == 1){
        Serial.print("d1");
        LINE.setToken(LINE_TOKEN);
        LINE.notify("\n 藥物已被領取!");
        d1 = 0;
        t4 = t4 + 1;
        t8 = 1;
        s = 1;
        s2 = 1;
    }
}
}

```

```
}
```

### 讓 APP 裡的剩餘藥量數量改變：

```
void mqttsend(){
    if(t1 == 1 && t5 == 1){
        client.publish(topic1, "剩六天");
        t5 = 0;
    }
    if(t1 == 2 && t5 == 1){
        client.publish(topic1, "剩五天");
        t5 = 0;
    }
    if(t1 == 3 && t5 == 1){
        client.publish(topic1, "剩四天");
        t5 = 0;
    }
    if(t1 == 4 && t5 == 1){
        client.publish(topic1, "剩三天");
        t5 = 0;
    }
    if(t1 == 5 && t5 == 1){
        client.publish(topic1, "剩兩天");
        t5 = 0;
    }
    if(t1 == 6 && t5 == 1){
        client.publish(topic1, "剩一天");
        t5 = 0;
    }
    if(t1 == 7 && t5 == 1){
        client.publish(topic1, "沒藥了");
        LINE.notify("\n 早上要吃的藥沒了!!!請補藥。");
        t1 = 0;
    }
}
```

//早上剩餘藥量 start

```

    t5 = 0;
}
//早上剩餘藥量 end
if(t2 == 1 && t6 ==1){
    client.publish(topic2, "剩六天");
    t6 = 0;
}
//中午剩餘藥量 start
if(t2 == 2 && t6 ==1){
    client.publish(topic2, "剩五天");
    t6 = 0;
}
if(t2 == 3 && t6 ==1){
    client.publish(topic2, "剩四天");
    t6 = 0;
}
if(t2 == 4 && t6 ==1){
    client.publish(topic2, "剩三天");
    t6 = 0;
}
if(t2 == 5 && t6 ==1){
    client.publish(topic2, "剩二天");
    t6 = 0;
}
if(t2 == 6 && t6 ==1){
    client.publish(topic2, "剩一天");
    t6 = 0;
}
if(t2 == 7 && t6 ==1){
    client.publish(topic2, "沒藥了");
    LINE.notify("\n 中午要吃的藥沒了!!!請補藥。");
    t6 = 0;
    t2 = 0;
}
//中午剩餘藥量 end

```

```

if(t3 == 1 && t7 ==1){
    client.publish(topic3, "剩六天");
    t7 = 0;
}
if(t3 == 2 && t7 ==1){
    client.publish(topic3, "剩五天");
    t7 = 0;
}
if(t3 == 3 && t7 ==1){
    client.publish(topic3, "剩四天");
    t7 = 0;
}
if(t3 == 4 && t7 ==1){
    client.publish(topic3, "剩三天");
    t7 = 0;
}
if(t3 == 5 && t7 ==1){
    client.publish(topic3, "剩二天");
    t7 = 0;
}
if(t3 == 6 && t7 ==1){
    client.publish(topic3, "剩一天");
    t7 = 0;
}
if(t3 == 7 && t7 ==1){
    client.publish(topic3, "沒藥了");
    LINE.notify("\n 晚上要吃的藥沒了!!!請補藥。");
    t7 = 0;
    t3 = 0;
}
//晚上剩餘藥量 end
if(t4 == 1 && t8 ==1){
    client.publish(topic4, "剩六天");

```

```

    t8 = 0;
}
if(t4 == 2 && t8 == 1){
    client.publish(topic4, "剩五天");
    t8 = 0;
}
if(t4 == 3 && t8 == 1){
    client.publish(topic4, "剩四天");
    t8 = 0;
}
if(t4 == 4 && t8 == 1){
    client.publish(topic4, "剩三天");
    t8 = 0;
}
if(t4 == 5 && t8 == 1){
    client.publish(topic4, "剩二天");
    t8 = 0;
}
if(t4 == 6 && t8 == 1){
    client.publish(topic4, "剩一天");
    t8 = 0;
}
if(t4 == 7 && t8 == 1){
    client.publish(topic4, "沒藥了");
    LINE.notify("\n 睡前要吃的藥沒了!!!請補藥。");
    t8 = 0;
    t4 = 0;
}
//睡前剩餘藥量 end
}

```

重複執行副程式，且當按鈕被按下，將會重製藥物藥艙及剩餘藥量:

```
void loop() {
```

```

    if (millis() - timenow >= 1000) {
state = digitalRead(e18_sensor);
//Serial.println(digitalRead(e18_sensor));
//Serial.println(state);
client.loop();
sensor();
printLocalTime();    //getting the RTC
distinguish();
mqttsend();
    }
buttonState = digitalRead(pushButton); //從 7 號腳位偵測 有按或沒按 1 或 0 把
結果給 buttonState
if(buttonState==1 and beforeState==0){
    client.publish(topic1, "剩七天");
    client.publish(topic2, "剩七天");
    client.publish(topic3, "剩七天");
    client.publish(topic4, "剩七天");
}
beforeState=buttonState;
}

```

### 3-4-2 Arduino 程式

所需要使用到的函示庫(蜂鳴器、馬達):

```

#define buzzerPin 7
#include <Stepper.h>

```

自訂義全域變數:

```

const int stepsPerRevolution = 256;           //藥倉馬達步數
const int centralstepsPerRevolution = 512;   //中心軸馬達步數
unsigned long timenow = 0;

```

全域變數(指令及運算變數):



```
String morning, noon, night, sleep;
int Mo = 0;
int No = 0;
int Ni = 0;
int Sl = 0;
int central = 0;
char Top2;
String messenge;
```

### **馬達腳位:**

```
Stepper morningStepper(stepsPerRevolution,22 , 24, 26, 28);
Stepper noonStepper(stepsPerRevolution,30 , 32, 34, 36);
Stepper nightStepper(stepsPerRevolution,38 , 40, 42, 44);
Stepper sleepStepper(stepsPerRevolution,46 , 48, 50, 52);
Stepper centralStepper(stepsPerRevolution,47 , 49, 51, 53);
```

### **主程式(設定鮑率、蜂鳴器接腳、馬達轉動速度):**

```
void setup() {
  Serial.begin(115200);
  pinMode(buzzerPin,OUTPUT);
  morningStepper.setSpeed(60);
  noonStepper.setSpeed(60);
  nightStepper.setSpeed(60);
  sleepStepper.setSpeed(60);
  centralStepper.setSpeed(60);
}
```

### **副程式(4 個時段的藥倉及中心軸的馬達運轉):**

```
void motor1(){
  morningStepper.step(-stepsPerRevolution);
  centralStepper.step(centralstepsPerRevolution);
```

```

        central = central + 1;
        Mo = Mo + 1;
    }
void motor2(){
    noonStepper.step(-stepsPerRevolution);
    centralStepper.step(centralstepsPerRevolution);
    central = central + 1;
    No = No + 1;
}
void motor3(){
    nightStepper.step(-stepsPerRevolution);
    centralStepper.step(centralstepsPerRevolution);
    central = central + 1;
    Ni = Ni + 1;
}
void motor4(){
    sleepStepper.step(-stepsPerRevolution);
    centralStepper.step(centralstepsPerRevolution);
    central = central + 1;
    Sl = Sl + 1;
}
void centralmotor(){
    if(messenge == "center"){
        centralStepper.step(512);
        central = central + 1;
    }
}

```

### 副程式(重製馬達):

```

void rst(){
    if (Mo == 7){
        morningStepper.step(2048);
        Mo = 0;
    }
}

```

```

}
if (No == 7){
    noonStepper.step(2048);
    No = 0;
}
if (Ni == 7){
    nightStepper.step(2048);
    Ni = 0;
}
if (Sl == 7){
    sleepStepper.step(2048);
    Sl = 0;
}
if (central == 4){
    centralStepper.step(-2048);
    central = 0;
}
}

```

### 副程式(使蜂鳴器運作):

```

void trigger(){
    tone(buzzerPin, 2000, 1000);
}

```

### 重複程式(接收 Esp32 指令並運作該副程式):

```

void loop() {
    messenge=Serial.readString();
    Serial.print(messenge);
    if (messenge == "buzzer"){
        trigger();
    }
    if (messenge=="a1") {
        motor1();
    }
}

```

```
        rst();
    }
    if (message=="b1") {
        motor2();
        rst();
    }
    if (message=="c1") {
        motor3();
        rst();
    }
    if (message=="d1") {
        motor4();
        rst();
    }
    if (message=="center") {
        centralmotor();
    }
    delay(1000);
}
```

### 3-4-3 APP 程式



APP 螢幕開啟時運作功能及連接網路伺服器:

當 Screen1 初始化

執行

- 設置 全域 提前領藥 為 建立清單
  - "早上"
  - "中午"
  - "晚上"
  - "睡前"
- 設 計時器1 . 啟用計時 為 真
- 設 藍牙客戶端1 . 啟用安全連線 為 真
- 設 UrsPahoMqttClient1 . Broker 為 "broker.emqx.io"
- 設 UrsPahoMqttClient1 . Port 為 1883
- 設 UrsPahoMqttClient1 . Protocol 為 "TCP"
- 設 UrsPahoMqttClient1 . UserName 為 "ninjasalad"
- 設 UrsPahoMqttClient1 . UserPassword 為 "medicineproject"
- 呼叫 UrsPahoMqttClient1 . 連線
- CleanSession 為 真
- 設 時間選擇器\_早上 . 文字 為 呼叫 微型資料庫1 . 取得數值
  - 標籤 "早上設定時間"
  - 無標籤時之回傳值 "選擇早上吃藥時間"
- 設置 全域 早上設定時間 為 時間選擇器\_早上 . 文字
- 設 時間選擇器\_中午 . 文字 為 呼叫 微型資料庫1 . 取得數值
  - 標籤 "中午設定時間"
  - 無標籤時之回傳值 "選擇中午吃藥時間"
- 設置 全域 中午設定時間 為 時間選擇器\_中午 . 文字
- 設 時間選擇器\_晚上 . 文字 為 呼叫 微型資料庫1 . 取得數值
  - 標籤 "晚上設定時間"
  - 無標籤時之回傳值 "選擇晚上吃藥時間"
- 設置 全域 晚上設定時間 為 時間選擇器\_晚上 . 文字
- 設 時間選擇器\_睡前 . 文字 為 呼叫 微型資料庫1 . 取得數值
  - 標籤 "睡前設定時間"
  - 無標籤時之回傳值 "選擇睡前吃藥時間"
- 設置 全域 睡前設定時間 為 時間選擇器\_睡前 . 文字
- 如果 是否為空 呼叫 微型資料庫1 . 取得數值
  - 標籤 "提醒訊息"
  - 無標籤時之回傳值 "0"
  - 為 真
- 則 呼叫 微型資料庫1 . 儲存數值
  - 標籤 "提醒訊息"
  - 儲存值 "該吃藥咯"
- 設 早上剩餘藥量 . 文字 為 呼叫 微型資料庫1 . 取得數值
  - 標籤 "早"
  - 無標籤時之回傳值 "0"
- 設 中午剩餘藥量 . 文字 為 呼叫 微型資料庫1 . 取得數值
  - 標籤 "中"
  - 無標籤時之回傳值 "0"
- 設 晚上剩餘藥量 . 文字 為 呼叫 微型資料庫1 . 取得數值
  - 標籤 "晚"
  - 無標籤時之回傳值 "0"
- 設 睡前剩餘藥量 . 文字 為 呼叫 微型資料庫1 . 取得數值
  - 標籤 "睡"
  - 無標籤時之回傳值 "0"

設定全域變數:

初始化全域變數 目前時間 為 "0"

初始化全域變數 判定瞬間 為 "0"

初始化全域變數 早上設定時間 為 "0"

初始化全域變數 中午設定時間 為 "0"

初始化全域變數 晚上設定時間 為 "0"

初始化全域變數 睡前設定時間 為 "0"

初始化全域變數 提醒訊息 為 "0"

初始化全域變數 提前領藥 為 建立空清單

### 擷取現在時間：

The image shows a Scratch script for extracting the current time. It starts with a '當計時器1計時' (When timer 1 starts) event. The script then sets the '日期' (Date) variable to a concatenated string of year, month, and date. It then extracts the hour, minute, and second from the current time using a series of '如果' (If) and '否則' (Else) blocks. Each time component is extracted by calling the '取得' (Get) block of the timer and comparing it to a threshold (0 for hours, 9 for minutes and seconds). The final step is to set the '全域目前時間' (Global current time) variable to a concatenated string of the extracted hour, minute, and second.

## 設定 4 個時段的吃藥時間:

當 時間選擇器\_早上 . 時間設定完成

執行

```

    設 時間選擇器_早上 . 文字 為 合併文字
    如果 ( 時間選擇器_早上 . 小時 > 9 )
    則 時間選擇器_早上 . 小時
    否則 合併文字 "0"
    "0"
    如果 ( 時間選擇器_早上 . 分鐘 > 9 )
    則 時間選擇器_早上 . 分鐘
    否則 合併文字 "0"
    時間選擇器_早上 . 分鐘

    呼叫 微型資料庫1 . 儲存數值
    標籤 "早上設定時間"
    儲存值 時間選擇器_早上 . 文字
    設置 全域 早上設定時間 為 呼叫 微型資料庫1 . 取得數值
    標籤 "早上設定時間"
    無標籤時之回傳值 "0"
  
```

當 時間選擇器\_中午 . 時間設定完成

執行

```

    設 時間選擇器_中午 . 文字 為 合併文字
    如果 ( 時間選擇器_中午 . 小時 > 9 )
    則 時間選擇器_中午 . 小時
    否則 合併文字 "0"
    時間選擇器_中午 . 小時
    "0"
    如果 ( 時間選擇器_中午 . 分鐘 > 9 )
    則 時間選擇器_中午 . 分鐘
    否則 合併文字 "0"
    時間選擇器_中午 . 分鐘

    呼叫 微型資料庫1 . 儲存數值
    標籤 "中午設定時間"
    儲存值 時間選擇器_中午 . 文字
    設置 全域 中午設定時間 為 呼叫 微型資料庫1 . 取得數值
    標籤 "中午設定時間"
    無標籤時之回傳值 "0"
  
```

當 時間選擇器\_晚上 . 時間設定完成

執行

```

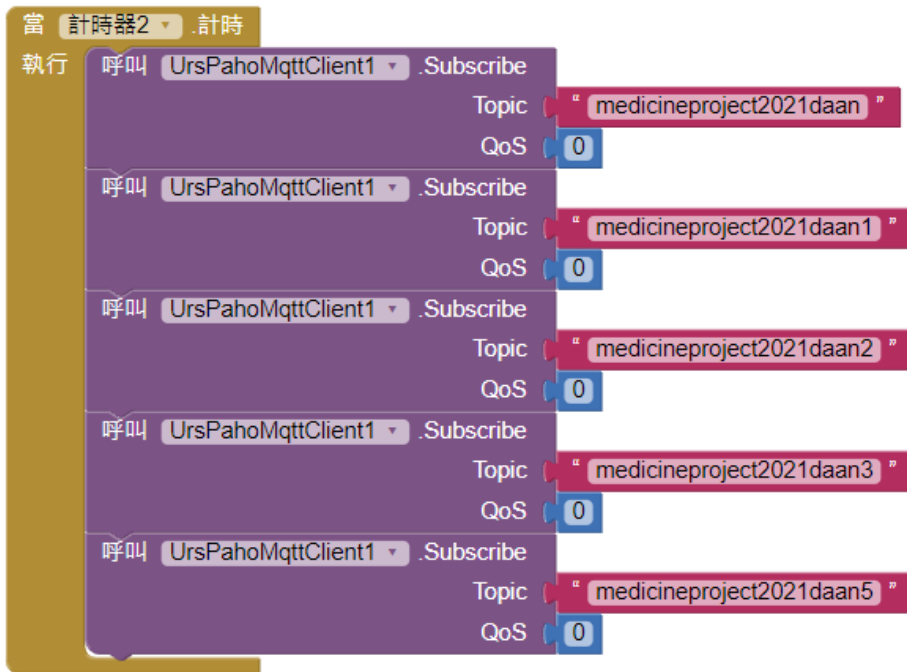
    設 時間選擇器_晚上 . 文字 為 合併文字
    如果 ( 時間選擇器_晚上 . 小時 > 9 )
    則 時間選擇器_晚上 . 小時
    否則 合併文字 "0"
    時間選擇器_晚上 . 小時
    "0"
    如果 ( 時間選擇器_晚上 . 分鐘 > 9 )
    則 時間選擇器_晚上 . 分鐘
    否則 合併文字 "0"
    時間選擇器_晚上 . 分鐘

    呼叫 微型資料庫1 . 儲存數值
    標籤 "晚上設定時間"
    儲存值 時間選擇器_晚上 . 文字
    設置 全域 晚上設定時間 為 呼叫 微型資料庫1 . 取得數值
    標籤 "晚上設定時間"
    無標籤時之回傳值 "0"
  
```





訂閱 MQTT(網路伺服器)主題:



時間設定完按下確認鍵(傳送 MQTT 訊息):



### 3-5 機構

#### 3-5-1 藥盒設計

1. 無裝飾物外殼：以拼圖式拼接法連接四個外殼
2. 出藥口外殼：放置 LCD 與按鈕
3. 中心柱與馬達支架：支撐起四個藥倉與馬達，使四個藥倉可公轉
4. 藥倉與隔板：外部設計卡榫以方便與中心軸連接



圖 14 外觀及內部結構圖

## 第4章 專題成果

### 4-1 操作流程圖

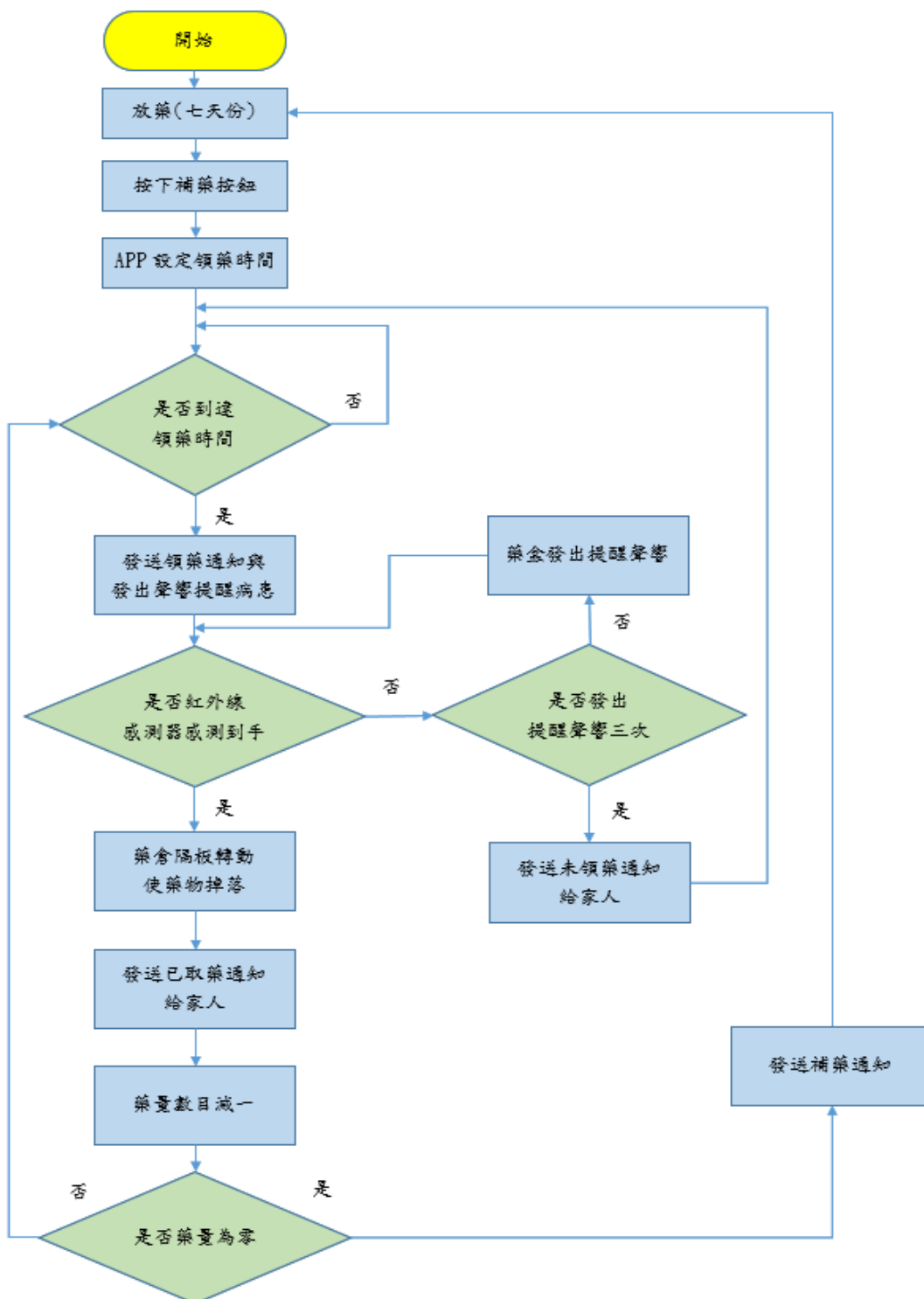


圖 15 操作流程圖

## 4-2 前置設定流程

### 4-2-1 放藥與剩餘藥量

將透明壓克力板上蓋打開後，即可在藥盒內的藥倉隔板之中放入各時段的藥物。放完藥後連續按兩下 LCD 旁的按鈕後，APP inventor 內的剩餘藥量將顯示剩餘七天份。

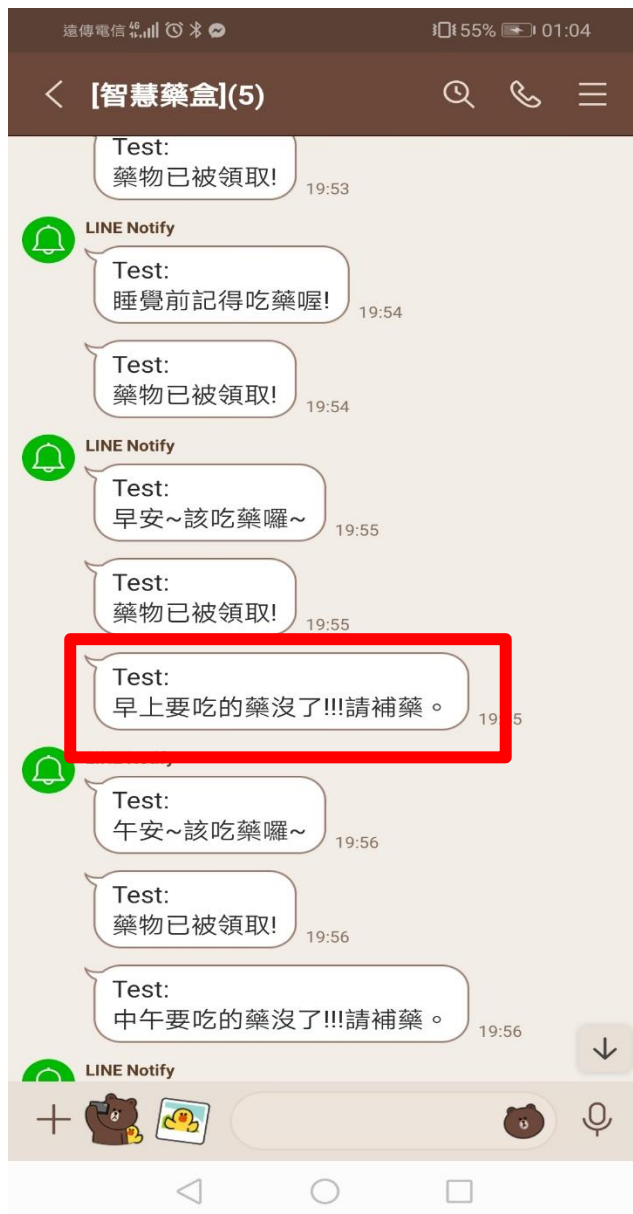


圖 16 補藥通知

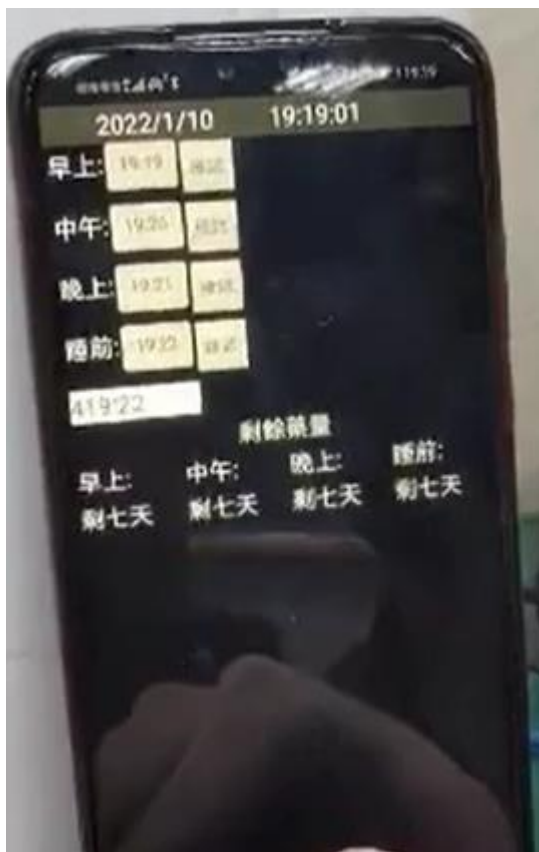


圖 17 AI2 UI

#### 4-2-2 設定領藥時間

可利用 APP inventor 設定之後的藥在什麼時間領取，到達所設定時間後，將會發送領藥通知與聲響通知病患取藥。



圖 18 設定時間

## 4-3 領藥過程

### 4-3-1 領藥通知

到達所設定的領藥時間後，蜂鳴器則發出提醒聲響且發送領藥通知到病患的 LINE 中提醒病患吃藥。

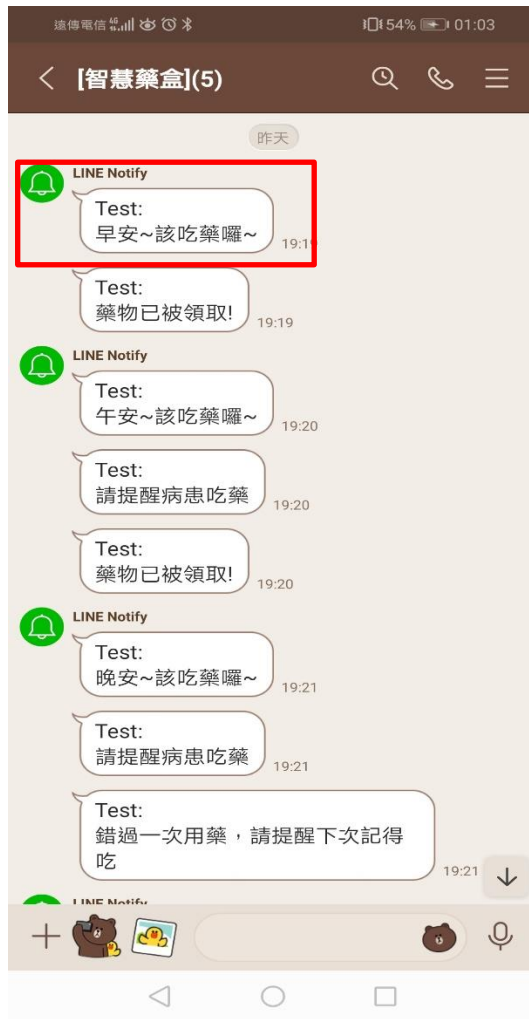


圖 19 手機領藥通知

#### 4-3-2 領藥動作

當紅外線感測器感測到手後，藥倉隔板便會轉動，使藥物掉落至手中。病患領完藥物後，則發送已取藥通知至病患家屬的 LINE 中提醒病患已取藥。





圖 20 手部深入掉藥



圖 21 手機已取藥通

## 第5章 結論與建議

### 5-1 結論

在製作專題的過程中，一路上我們跌跌撞撞，但也因為這次的專題，我們知道團結合作的重要性，在討論過程中也會互相發表意見並討論，並不會有人固執己見，也因為這樣，我們嘗試了許多方法也查詢了許多資料，在程式和物聯網的製作上，我們學到了物聯網的架構和原理，以及微處理機之間資料的傳遞，邏輯上的分析也更進一步，硬體上則學會 3D 列印機及雷射切割機的使用，也知道了要如何配置結構才能使結構堅固，文書處理及報告方面我們也學到要如何製作文書及報告，好讓報告時能抓住觀眾的目光，讓他人更快速理解我們想要傳達的訊息，同時也能有更好的口條，書面上則能完整呈現我們所有的資料，讓他人一看就清楚，有了這次專題的經驗，讓我們以後製作類似這種大型報告的時候，能夠製作得更佳完善。

### 5-2 建議

1. 利用 MQTT 時主題要相同，避免收不到資料
2. 一次列印多個 3D 物件時，記得開啟同時列印
3. 燒錄 esp32 時，RX、TX 等接腳記得拔除，避免燒錄失敗

## 參考文獻

- [1] 蜂鳴器 <https://www.circuspi.com/index.php/2021/09/11/arduino-passive-buzzer/>
- [2] RTC [http://www.4k8k.xyz/article/Naisu\\_kun/115627629](http://www.4k8k.xyz/article/Naisu_kun/115627629)
- [3] 串列傳輸 <https://www.programmingboss.com/2021/04/esp32-arduino-serial-communication-with-code.html>
- [4] 紅外線感測 <https://www.yiboard.com/thread-1602-1-1.html>
- [5] [ESP32 - 維基百科，自由的百科全書 \(wikipedia.org\)](https://zh.wikipedia.org/wiki/ESP32)

## 附錄

### 附錄一 設備清單


類別	設備、軟體名稱	應用說明
軟體	Onshape	繪製 3D 圖檔
軟體	Arduino1.8.13	撰寫程式控制藥盒
軟體	App inventor	製作手機 APP
軟體	Line notify	建立與 LINE notify 的連結
軟體	Cura	切片 3D 圖檔
軟體	Microsoft Office	撰寫計畫書及簡報
軟體	LaserBox	控制雷射切割機
硬體	Atom 3	做出外殼、藥倉、中心柱與馬達支架
硬體	雷射切割機	切出藥盒上蓋
硬體	電腦	寫程式與報告
硬體	隨身碟	儲存資料
硬體	手機	使用 APP，收取提醒通知

## 附錄二 材料清單


類別名稱	材料名稱	單位	數量	應用說明	備註
步進馬達	28BYJ-48	個	4	轉動中心軸及隔板	
按鈕	輕觸開關	個	1	重置剩餘藥量	
顯示器	1602LCD	個	1	顯示現在時間	
電阻	可變電阻	個	1	調整顯示對比	
開發版	mega2560	個	1	硬體控制	
模組	esp32	個	1	物聯網	
模組	E18-D80NK	個	1	手部感測	
線材	公對公排線	條	N	內部線路連接	
線材	公對母排線	條	N	內部線路連接	
線材	母對母排線	條	N	內部線路連接	
蜂鳴器	無源蜂鳴器	個	1	發出提醒聲響	
板子	電路板	片	2	電路配置	
板子	壓克力板	片	1	上蓋	
銅管	銅管	根	4	支撐上蓋	
電源	行動電源	個	1	提供電源	

### 附錄三 研究成員簡歷

姓名	陳冠霖	班級	電子三乙	
曾修習專業科目	<ol style="list-style-type: none"> <li>1. 基本電學實習</li> <li>2. 電子學實習</li> <li>3. 數位邏輯實習</li> <li>4. 微處理機實習</li> <li>5. 程式設計實習</li> <li>6. 微電腦實質</li> <li>7. 汽車電子應用黃習</li> <li>8. CPLD 邏輯實習</li> <li>9. Altium Designer 電路圖與印刷電路板設計</li> </ol>			
參與專題工作項目	<p>組長            整體結構設計            3D 繪圖與列印電路設計</p>			
經歷簡介	<ol style="list-style-type: none"> <li>1. 電子丙級技能檢定考合格</li> <li>2. 擔任 108 學年工廠安全</li> <li>3. 擔任 108 學年班長</li> <li>4. 參加 108 學年度英文字彙比賽第一名</li> <li>5. 擔任 109 年工廠 領班</li> <li>6. 參加 109 學年度英文字彙比賽第一名</li> <li>7. 擔任 110 學年衛生股長</li> <li>8. 參加 110 學年度英文字彙比賽第一名</li> <li>9. 參加第 17 屆技術創造力競賽</li> <li>10. 參加 110 學年度英文簡報比賽</li> </ol>			

姓名	陳柏諺	班級	電子三乙	
曾修習專業科目	<ol style="list-style-type: none"> <li>1. 基本電學實習</li> <li>2. 電子學實習</li> <li>3. 數位邏輯實習</li> <li>4. 微處理機實習</li> <li>5. 程式設計實習</li> <li>6. 微電腦實質</li> <li>7. 汽車電子應用黃習</li> </ol>			

	8.CPLD 邏輯實習 9.Altium Designer 電路圖與印刷電路板設計	
<b>參與專題 工作項目</b>	外殼設計 IFTTT 製作 App Inventor 製作 文書處理	
<b>經歷簡介</b>	1.工業電子丙級技能檢定考合格 2.擔任 108 學年副班長 3.擔任 109 學年總務股長	

姓名	黃健嘉	班級	電子三乙
<b>曾修習 專業科目</b>	1 基本電學實習 2.電子學實習 3.數位邏輯實習 4.微處理機實習 5.程式設計實習 6.微電腦實質 7.汽車電子應用黃習 8.CPLD 邏輯實習 9.Altium Designer 電路圖與印刷電路板設計		
<b>參與專題 工作項目</b>	外殼設計 程式設計 物聯網設計 3D 繪圖與列印		
<b>經歷簡介</b>	1.工業電子丙級技能檢定考合格		

姓名	鄭良弼	班級	電子三乙	
曾修習專業科目	1.基本電學實習 2.電子學實習 3.數位邏輯實習 4.微處理機實習 5.程式設計實習 6.微電腦實質 7.汽車電子應用黃習 8.CPLD 邏輯實習 9.Altium Designer 電路圖與印刷電路板設計			
參與專題工作項目	外殼設計 程式設計 物聯網設計 App Inventor 製作			
經歷簡介	1.工業電子丙級技能檢定考合格			