

天氣預報 AI 機器人

研究者：陳諒昕、沈書賢 指導老師：曾義祥

壹、研究動機

我們在資優班教室裡看到一台 ai 聊天機器人，有基本的語音互動功能，為了增加生活的便利性，我們打算讓它擁有天氣預報的功能。這隻機器人必須搭配一台手機才能使用，但不是人人都有一台閒置的手機，於是我們想用別的硬體設備來取代，也想利用雷射切割機來為它打造一個全新又更堅固的外觀。

貳、研究目的

讓 AI 聊天機器人能和真人互動，用語音的方式達到天氣預報的效果，這樣比手動打字搜尋的方式更快。我們也加入螢幕，來顯示天氣預報詳細資料。同時我們也要用木板和雷射雕刻機來打造一個全新的外觀，以取代原本容易受潮的紙製外殼。

參、過程

操作影片



軟體程式

第一版	模組是 PCM5102(語音輸出) 和 LD3320(語音輸入)，但語音辨識成功率太低並且會延遲。
第二版	改用劉正吉老師發布的吉哥積木，語音部份順利完成，但遇到供電問題。
第三版	更換電源供應模組，並加入電量顯示

外觀設計

第一版	使用厚紙板製作，但在各方面的效果皆不理想
第二版	改採電腦繪製
第三版	使用雷射切割木板的方式來製作

肆、結論

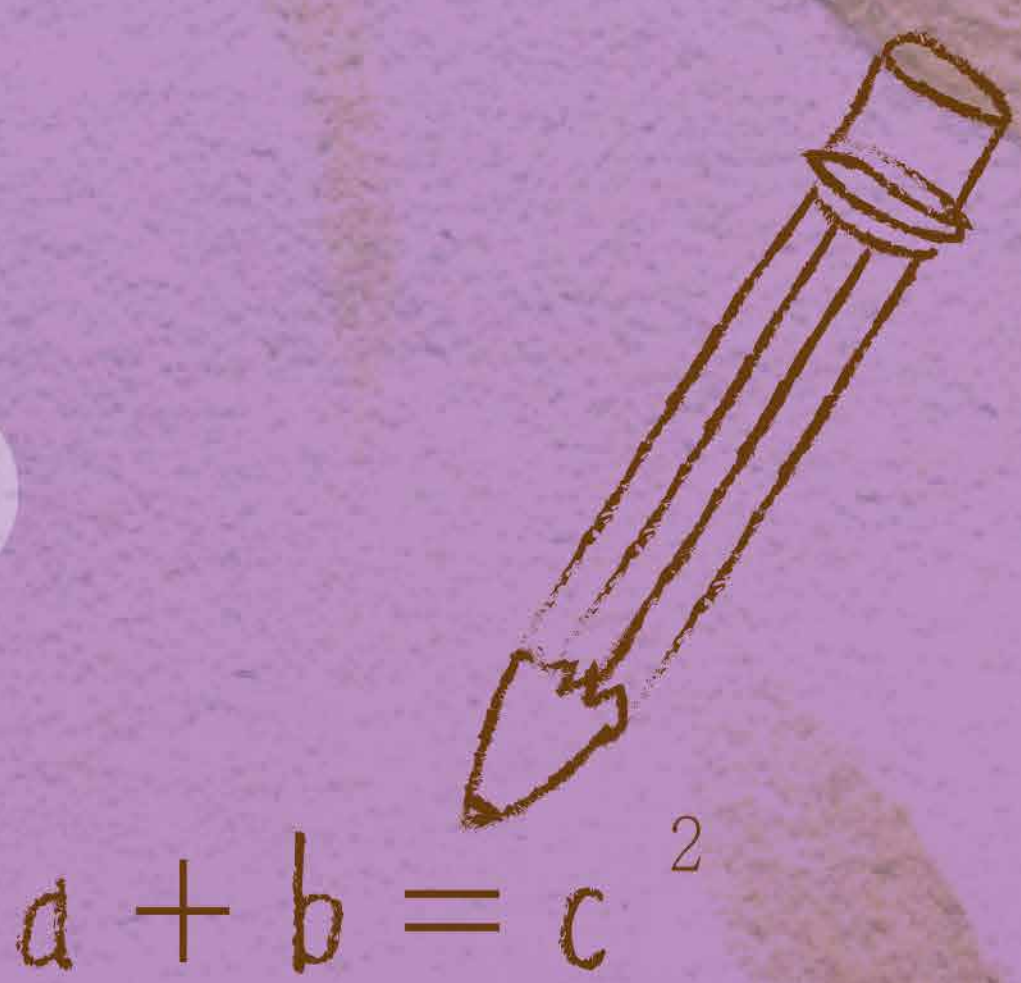
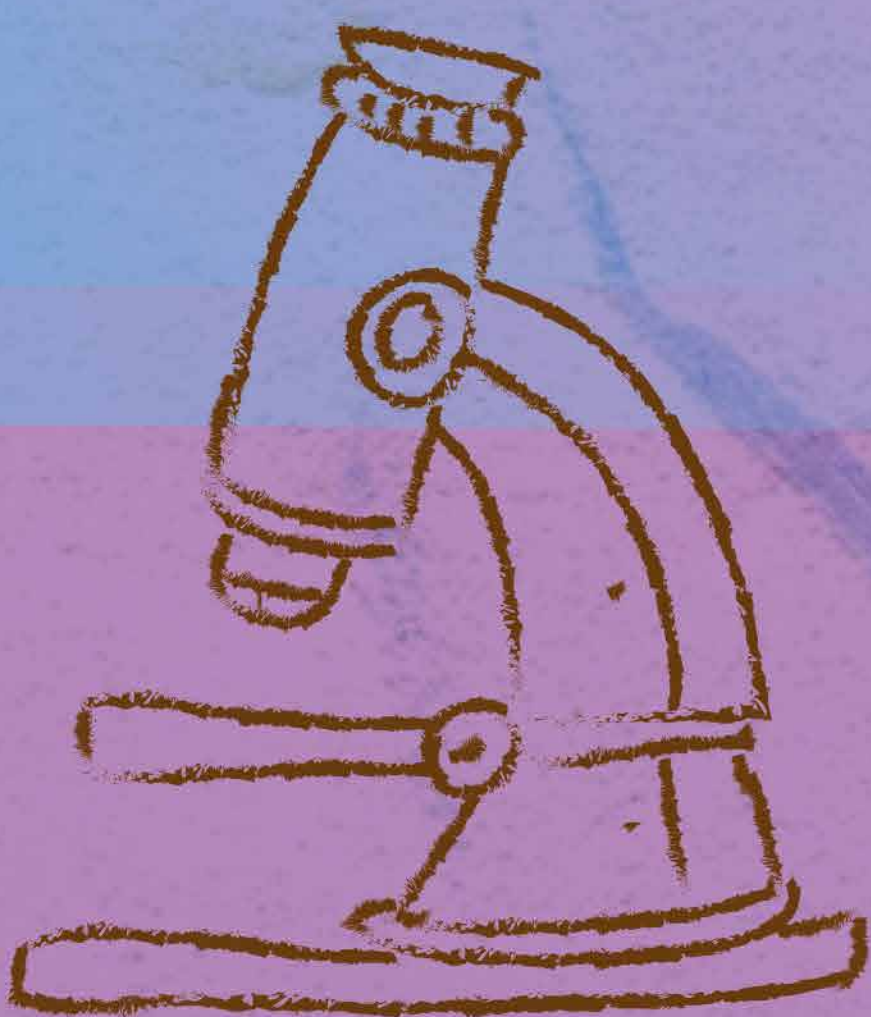
一、電路

我用 C++ 程式配合 Arduino 程式編輯器來寫程式，因為我們用了許多不同的 arduino 模組，因此要學會許多技巧，才能好好運用這些模組。語音辨識的部分 (microbit) 則是使用 python 來寫出程式。語音輸出部分則是使用吉哥積木裡的程式庫搭配 Arduino 編輯器來完成。

二、外觀

我在 inkscape 軟體把設計圖完成，這個軟體能自動做出卡榫，等所有的設計圖畫完後，再匯入 cubio2 程式即可，但是雷射切割的深度問題需要再想辦法解決。

三、因為我們的資料來源是國外的網站，調閱資料的步驟比較多，還要等一段時間，資料才進得來，而且常常無法讀取。在程式中，我們使用 json 程式庫來解決這幾個問題。



天氣預報AI機器人

研究者:陳諒昕、沈書賢

指導老師:曾義祥老師

壹、緒論

一、研究動機

我們在資優班教室裡看到一台ai聊天機器人，有基本的語音互動功能，為了增加生活的便利性，我們打算讓它擁有天氣預報的功能。這隻機器人必須搭配一台手機才能使用，但不是人人都有一台閒置的手機，於是我們想用別的硬體設備來取代，也想利用雷射切割機來為它打造一個全新又更堅固的外觀。

二、研究目的

讓ai聊天機器人能和真人互動，用語音的方式達到天氣預報的效果，這樣比手動打字搜尋的方式更快。我們也加入螢幕，來顯示天氣預報詳細資料。同時我們也要用木板和雷射雕刻機來打造一個全新的外觀，以取代原本容易受潮的紙製外殼。

三、研究問題

1. 用C++程式語言讓機器人有語音輸入與輸出的功能
2. 以雷射雕刻機製作機器人外殼
3. 如何從網路上取得天氣資料

貳、文獻探討

一、主機板

1. Esp8266

ESP8266大致可以扮演兩個角色，第一個是當作arduino的擴充，就像是幫arduino裝了WIFI擴充版；第二個是扮演獨立運作的Web Client角色，就是當成一個伺服器，例如接一個溫溼度感測器，並將資料傳至網頁，使用者就可以連接。我們是讓它當成一個用戶端，上網取得資料，並顯示在螢幕上。

2. Esp32

ESP32可用於IOT物聯網，例如：-相機的視頻流 -OTT機頂盒設備 -家庭自動化 -WIFI語音識別設備-工業無限控制 -健康醫療 -運動檢測和防丟報警器 -可穿戴電子產品：智能手環，智能手錶我們則是它當作用戶端，取得網路上的資料播出聲音(PCM5102)。

二、語音數入及輸出

1. EF04108:

此模組可用於需要語音辨識的環境，它可以輕易辨識出一些關鍵字，裡如開燈、關燈等，也可以學習辨識其他的自訂關鍵字，而且可以辨識中文，我們把它用在啟動語音播放的方面，這樣就可以實現語音控制的效果。

2. max98357a:

此模組可接擴音機，並播出高品質音效，用於耳機等小功率設備。

三、天氣資料取得

由於天氣預報站需要天氣資料，原本想說直接取用中央氣象局的資料就好了，但是程式無法辨識資料，因此換成國外的openweathermap網站

Weather	Current weather and forecast	Developer plan	Hourly forecast: 4 days Daily forecast: 16 days Calls per minute: 3,000 3 hour forecast: 5 days	view
History	Historical data	Medium plan	Calls per day: 50,000	view

取得資料的方法也很多，可以直接輸入網址取得，但要放入程式裡有點困難，因此選擇直接用現有的程式庫來取得資料，進而顯示在螢幕上。

四、超圖解Arduino互動設計入門

打造微電腦互動作品，需要融合不同領域的軟、硬體基礎知識。本書的目標是讓沒有電子電路基礎，對微電腦、電子DIY 及互動裝置有興趣的人士，也能輕鬆閱讀，進而順利使用Arduino 控制板完成互動應用。因此，本書以手繪圖解的方式，說明基本電子學和程式設計入門。

五、超圖解esp32深度實作

ESP32 的獨特硬體架構需要專門的程式庫和指令才能釋放它的威力，例如，低功耗藍牙（BLE）無線通訊、可輸出高品質數位音效的 I2S(序列音訊介面)、DAC(數位類比轉換器)、Mesh(網狀) 網路、HTTPS 安全加密連網... 等。

六、雷射雕刻

1. 定義

雷射雕刻機是利用極高強度的雷射光，連續照射到所要雕刻的物品上，讓物體局部達到熔點，向下軟化、切割，再配合高壓的氣流將物體熔化或氣化的過程中產生的廢氣吹走，達到比刀模更精準的切割效果。相較於傳統的手工雕刻方式，雷射雕刻可以將雕刻效果做到相當細膩，絲毫不亞於手工雕刻的工藝水平。

2. cubiio 2 功能

可切穿多種材質厚達5mm，直接雕刻不銹鋼等多項堅硬材質，性能毫不妥協。自動對焦、預覽對位、開蓋模式、Wi-Fi遠端連線遙控、OTA無線韌體更新、相容多種軟體格式、雷射免對光路免保養、內建空氣濾芯、機內照明、社群共享。

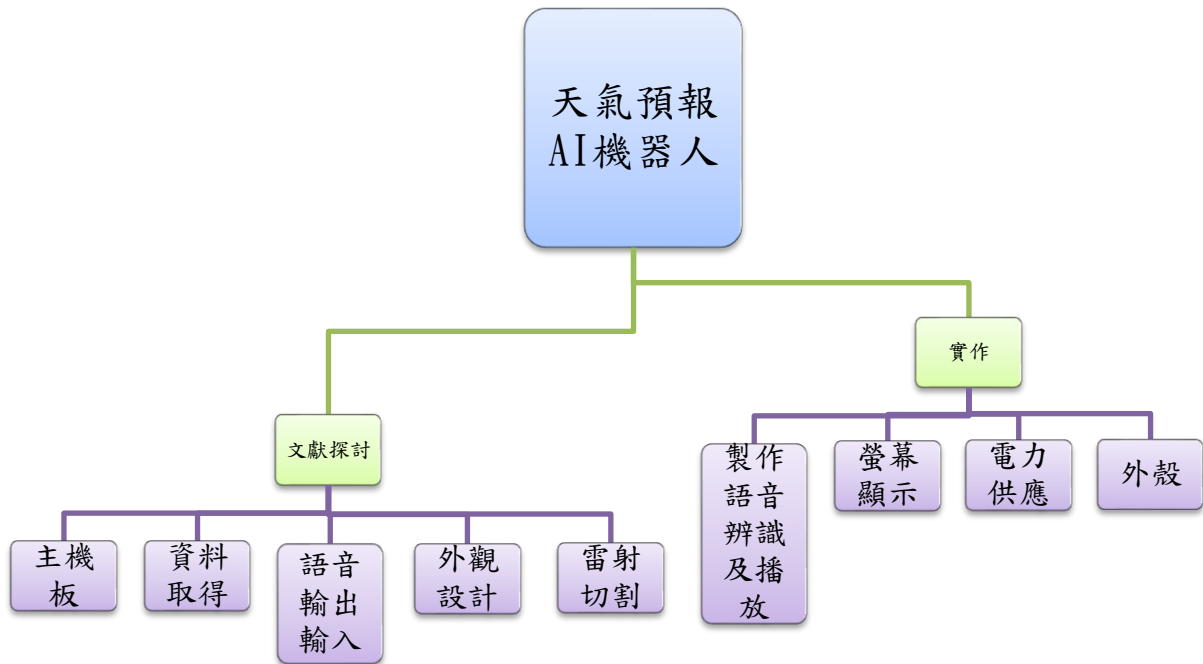
七、AI聊天機器人

『聊天機器人』原本是一種網路程式，擅長面對重複率高、順序變化小的語音或文字問題，給予即時且有效的回覆。另外，聊天機器人還可以扮演語音小助理，利用『網路爬蟲』的能力，將你所問的問題，搜尋網路找到對應的答案，想問天氣、算數學、找電影、找美食，通通不成問題。



參、研究方法與設計

一、研究架構



二、研究流程

1. 擬定主題
2. 查詢文獻
3. 了解硬體的運作原理
4. 繪製設計圖
5. 討論與購買材料
6. 軟硬體及外觀製作
7. 運作測試並排除錯誤
8. 撰寫研究報告

三、研究設計

1. 製作語音辨識及播放

我用對於arduino比較好用的模組來製作，並加入不同的函數來讓程式變簡單、易讀。語音播放因為還要寫入許多文字轉語音的資料，因此我決定用運算速度比較快的主板來執行這個程式。

2. 螢幕顯示

因為要在螢幕上顯示天氣預報的詳細資料，於是我們使用彩色螢幕來顯示。

3. 外殼設計

目標是能容納所有零件又能兼顧美觀，取代原本的紙材質，採用木板。

肆、研究結果

一、AI設計

1. 第一版

我一開始用的模組是PCM5102(語音輸出)和LD3320(語音輸入)，不只是因為它們比較便宜，而且我也知道使用方法，因此選用這兩個模組。

PCM5102	LD3320
	

當我寫完這些程式之後，語音輸入和輸出有效果，但是使用時發現這兩個模組有太多問題要解決，例如語音辨識成功率太低，我們對機器人講了好多話，叫它好幾次但不是每次都有反應。還有語音輸出速度太慢，有時候甚至延遲了快半分鐘才報出天氣。

下圖是當初設計的程式：

LD3320程式(初始化)
<pre>{ UBYTE nAsrResCount = 0; ucRegVal = LD_ReadReg(0x2B); ucHighInt = LD_ReadReg(0x29); // interrupt enable flag ucLowInt = LD_ReadReg(0x02); // interrupt enable flag LD_WriteReg(0x29, 0); // interrupt disable LD_WriteReg(0x02, 0); // interrupt disable if(nLD_Mode == LD_MODE_ASR_RUN) { Serial.print("-----ASR-----\r\n"); //The interruption caused by speech recognition //(There is sound input, and there is interruption whether if((ucRegVal & 0x10) && LD_ReadReg(0xb2)==0x21 && LD_ReadReg(0xb2) != 0x21) { nAsrResCount = LD_ReadReg(0xba); if(nAsrResCount>0 && nAsrResCount<=4) { Serial.print("ASR SUCCESSFUL \r\n"); nAsrStatus = LD_ASR_FOUNDDOK; } else { Serial.print("ASR UNSUCCESSFUL \r\n"); nAsrStatus = LD_ASR_FOUNDDZERO; } } else { Serial.print("No ASR \r\n"); nAsrStatus = LD_ASR_FOUNDDZERO; } } LD_WriteReg(0x2b, 0); LD_WriteReg(0x1c, 0); return; }</pre>

PCM5102程式(主要部分)

```
void setup() {
    Serial.begin(115200);
    pinMode(inputpin, INPUT);
    pinMode(LED, OUTPUT);
    digitalWrite(LED, HIGH);
    pwm.begin();
    pwm.setPWMFreq(FREQUENCY);
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, pwd);
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(500);
    }
    Serial.println("\nWi-Fi已連線!");
    audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);
    audio.setVolume(21);
}

void loop() {
    bool State = digitalRead(inputpin);
    http.begin(url);
    http.end();
    payload = http.getString();
    if (State != 0)
    {
        nod();
        activate = true;
        digitalWrite(LED, LOW);
        DynamicJsonDocument doc(1024);
        deserializeJson(doc, payload);
        JsonObject weather = doc["weather"][0];
        JsonObject main = doc["main"];
        int temp = (float)main["temp"] - 273.15;
        char* msg = first + temp;
        audio.connecttospeech(msg, "zh");
    }

    audio.loop();
}
```

2. 第二版

就在上學期期末的時候，劉正吉老師在網路上發布了「吉哥積木20220110版」，我發現這些程式積木也許可以處理我們當前遇到的問題，於是改成microbit主機板搭配EF04108(語音輸入)和ESP32主機板搭配MAX98357A(語音輸出)。程式寫好之後，語音指令可以自訂了，語音輸出時也可以即時播放不再有延遲的狀況，因此最後我們決定用這兩個模組來製作機器人，我們的程式也全面改成用吉哥積木裡的程式庫來使用。

程式語言的部分完成了，但是在硬體設備方面遇到了電量供應的問題，就是語音輸出到一半就會斷掉，整台機器人斷電，於是我開始搜尋如何解決這個問題。

MAX98357A



EF04108



取得語音資料

```
void getVoiceFromGoogle(String myTalk,String tl)
{
    myTalk=URLEncode(myTalk.c_str());
    ttsDone=false;
    httpDone=true;
    mp3Done=true;
    dacPlayType="TTS";
    ttsContent=myTalk;
    myTalk="http://translate.google.com/translate_tts?ie=UTF-8&client=tw-ob&tl="+tl+"&q="+myTalk;
    saveTTStoFile(myTalk,"/TTS/tts.mp3",2);
    getVoiceFromFile("/TTS/tts.mp3",2);
}
```


語音播放

```
void loop()
{
    checkDACrunning();
    checkTTS();
    checkMP3();
    checkHttpMP3();
    if (digitalRead(2) == HIGH) {
        u8g2.firstPage();
        do {
            u8g2.setCursor(0, 0);
            u8g2.print(String("台北偵測!").c_str());

            u8g2.sendBuffer();
        } while (u8g2.nextPage());
        dacStop();
        fetchWeatherInfo("1665148","567cc66d5b10f39b046e1e3862925c69");
        checkDACrunning();
        checkTTS();
        checkMP3();
        checkHttpMP3();
        getVoiceFromGoogle((String("台北目前溫度")+String((doc["main"] ["temp"]).as<float>
        u8g2.clearDisplay());
    } else if (digitalRead(15) == HIGH) {
        u8g2.firstPage();
        do {
            u8g2.setCursor(0, 0);
            u8g2.print(String("台中偵測!").c_str());

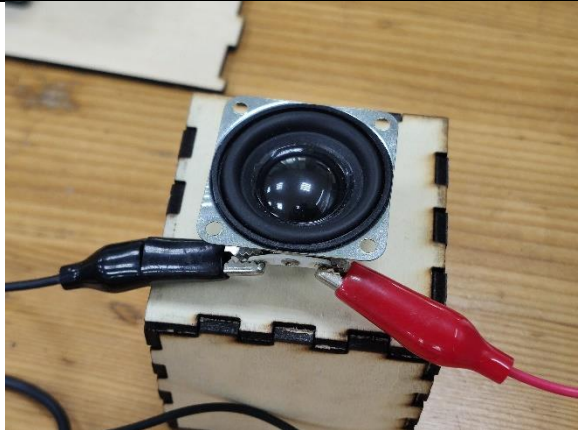
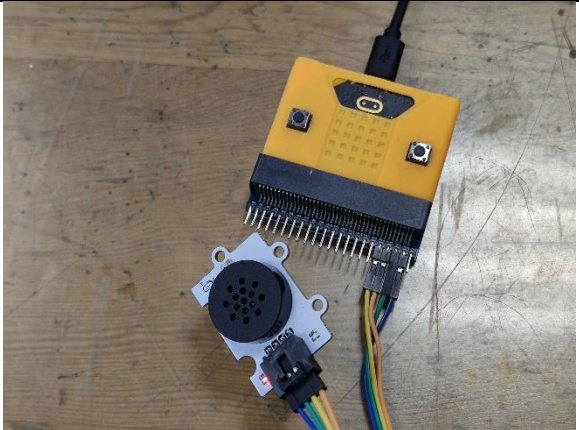
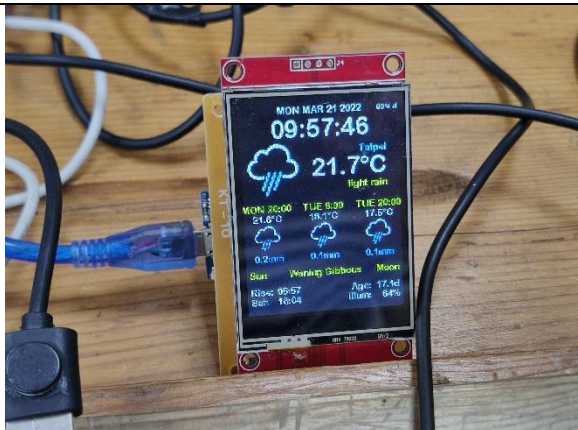
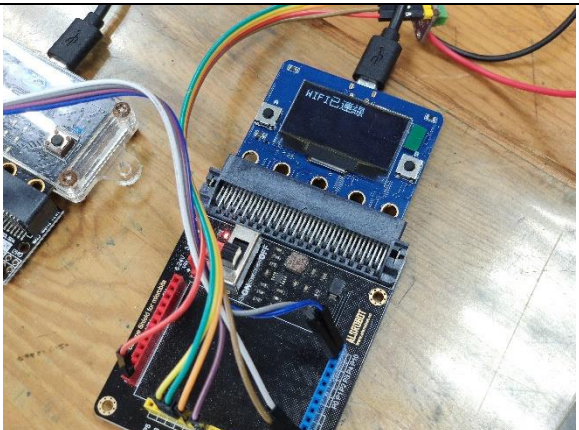
            u8g2.sendBuffer();
        } while (u8g2.nextPage());
        dacStop();
        fetchWeatherInfo("1668399","567cc66d5b10f39b046e1e3862925c69");
        checkDACrunning();
        checkTTS();
        checkMP3();
        checkHttpMP3();
        getVoiceFromGoogle((String("台中目前溫度")+String((doc["main"] ["temp"]).as<float>
        u8g2.clearDisplay());
    } else if (digitalRead(4) == HIGH) {
        u8g2.firstPage();
        do {
            u8g2.setCursor(0, 0);
            u8g2.print(String("嘉義偵測!").c_str());

            u8g2.sendBuffer();
        } while (u8g2.nextPage());
        dacStop();
        fetchWeatherInfo("1668347","567cc66d5b10f39b046e1e3862925c69");
        checkDACrunning();
        checkTTS();
        checkMP3();
        checkHttpMP3();
        getVoiceFromGoogle((String("嘉義目前溫度")+String((doc["main"] ["temp"]).as<float>
        u8g2.clearDisplay());
    }
}
```


語音辨識

```
1 asr.onASR(asr.vocabularyList.Learning_entry_3, function () {
2     basic.clearScreen()
3     radio.sendNumber(2)
4     basic.showString("Chiayi")
5     basic.clearScreen()
6 })
7 asr.onASR(asr.vocabularyList.Learning_entry_1, function () {
8     basic.clearScreen()
9     radio.sendNumber(0)
10    basic.showString("taipei")
11    basic.clearScreen()
12 })
13 input.onButtonPressed(Button.A, function () {
14     asr.setASRLearn()
15 })
16 asr.onASR(asr.vocabularyList.Hi_Shaun, function () {
17     basic.clearScreen()
18     basic.showIcon(IconNames.Happy)
19 })
20 input.onButtonPressed(Button.B, function () {
21     radio.sendNumber(0)
22     basic.showNumber(0)
23 })
24 asr.onASR(asr.vocabularyList.Learning_entry_2, function () {
25     basic.clearScreen()
26     radio.sendNumber(1)
27     basic.showString("Taichung")
28     basic.clearScreen()
29 })
30 radio.setGroup(6)
31 radio.setTransmitPower(7)
32
```

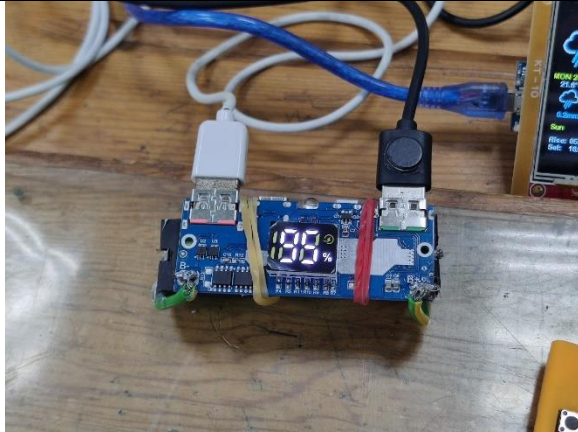

使用的硬體設備還有

小型喇叭	語音辨識模組
	
螢幕	esp32
	

3. 第三版

針對電源供應的問題，我去電子材料行看看有沒有電源供應模組，有看到一個之前沒看過的，於是買回家測試，沒想到竟然成功了!仔細一看，原來是因為變壓模組有兩個，因此解決了供電不穩以及斷電的問題。

也因為電量供應已經穩定了，這次還加入了「電量顯示」的功能，可以讓使用者知道當前的電量還有多少。

電源供應模組	
	

二、外觀

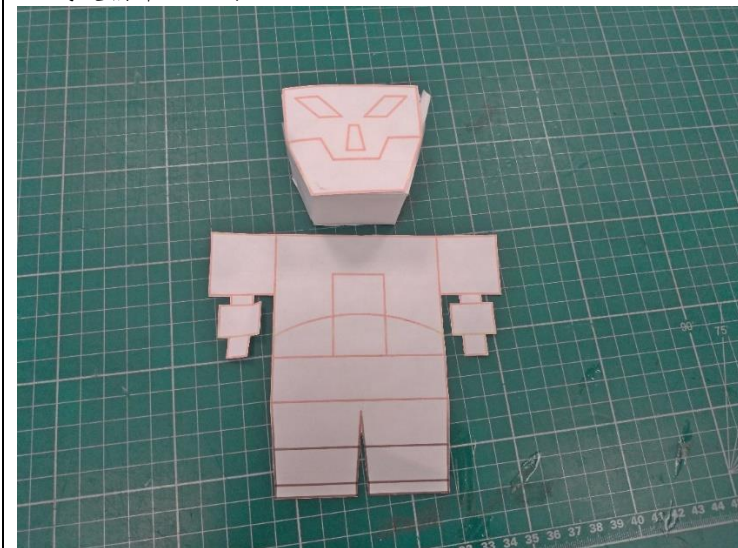
1. 第一版

一開始，我參考鋼彈來製作頭部，身體為了方便收納主機板，設計成長方形，因為是第一次做，以後還會做更改，所以都先用厚紙板製作，做到一半發現頸部可能會無法承受頭部的重量，因此決定將頭改為比較小型的八面體。



2. 第二版

在製作八面體的頭部之後，我們覺得角度及長度不精準，所以想試試看用電腦來繪圖，經過我搜尋查資料之後，嘗試以<https://www.geogebra.org/>網站來繪製，外觀比之前手繪的方式更精準一點了。

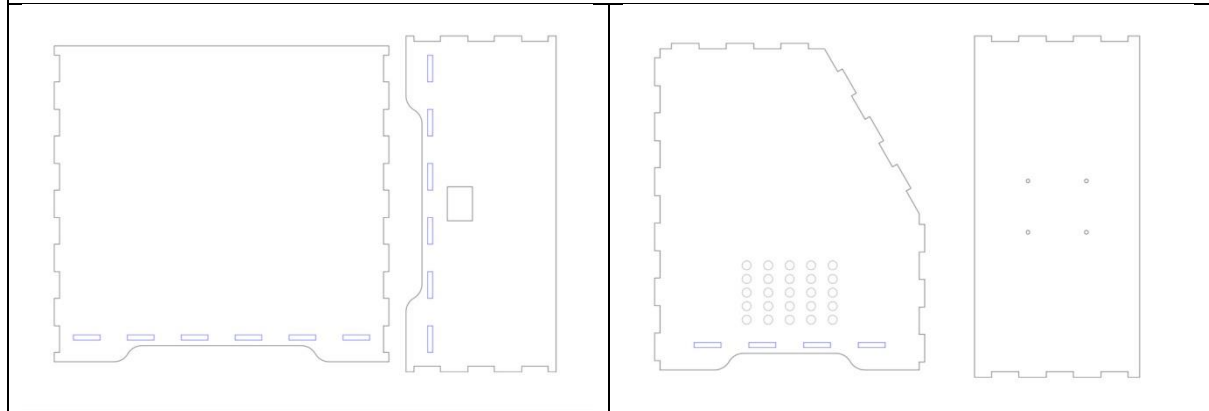


3. 第三版

紙製的外觀看起來還是太簡陋，所以在寒假的時候我去找了一些資料，這次想試試看用木板搭配雷射切割來製作機器人的外觀。

於是我先從<https://zh.makercase.com>網站製作零件。但發現這軟體只能切正方體或正五邊形，所以最後改用inkscape來製作。

用inkscape製作的零件展開圖



當我畫完平面圖之後發現很難把檔案轉成雷射切割可以讀取的格式。

為了解決這些問題，我從cubiio的官方網站找到軟體「laserweb」來轉檔。

成功讀取到檔案之後，開始進行切割工作。於是有了以下幾種版本

這是雷射切割的第一個版本



為了讓使用者更容易看到天氣資訊，將螢幕這一面改成斜的



這過程也遇到了切割的問題，雷射頭的功率用太高的話，會切太深傷到底板，所以用低功率分多次來切割。光是切一個6 X 6 X 6的立方體展開圖，一次就要大約20分鐘，切完之後拿起來檢查，發現切不夠深要再放回去繼續切，但是這樣就會有一點偏移，幾次反覆切割，木板就變得焦黑，切的深度也不一致。這個部分，可能要多幾次實驗才能找到完美的流程。

最後我們的機器人長這樣



實際操作效果



<https://youtu.be/THsSeR79A0E>

伍、結論與建議

一、研究結論

1. 電路

我用C++程式配合Arduino程式編輯器來寫程式，因為我們用了許多不同的arduino模組，因此要學會許多技巧，才能好好運用這些模組。

語音辨識的部分(microbit)則是使用python來寫出程式。

語音輸出部分則是使用吉哥積木裡的程式庫搭配Arduino編輯器來完成。

2. 外觀

我在inkscape軟體把設計圖完成，這個軟體能自動做出卡榫，等所有的設計圖畫完後，再匯入cubiio2程式即可，但是雷射切割的深度問題需要再想辦法解決。

3. 因為我們的資料來源是國外的網站，調閱資料的步驟比較多，還要等一段時間，資料才進得來，而且常常無法讀取。在程式中，我們使用json程式庫來解決這幾個問題。

二、研究建議

1. 我們剛開始想做跟模型機器人一樣漂亮的外觀，結果花了很多時間在做紙模型，建議以後的研究者可以直接用電腦版的inkscape軟體製作，並且用laserweb從svg轉成txt(g-code)檔，才能讓雷射切割機讀取。

2. 如果雷射切割要一次切到位，把零件一次切好，建議先切一塊小型的圖作為測試，找到更適合的功率，節省等待時間。

3. 寫程式時可以加入註解，以後修改時才會比較方便找到要修的地方，程式也會較容易閱讀。

4. 要盡量讓程式的執行效率加快，不然的控制板可能會過載。例如刪除不必要的指令，想辦法減少行數，不要太長，否則會不方便維護。

5. 使用國內的網站，資料取得可能會比較方便與準確，但是程式會變複雜，如果要這樣做，語音就會有延遲。

6. 為了增加趣味性，建議使用扭力大於3KG-cm的伺服馬達，才能讓機器人的頭動起來。