



VM1010+Apollo3 voice switch 交接文件

Advisor : Tay-Jyi Lin and Chingwei Yeh
Student : 陳憲億

Outline

1. 介紹

研究目的、行為概念

2. 使用元件

介紹 麥克風 (VM1010)
與 電路板 (Apollo3 blue)

3. 技術需求

語音識別、關鍵字檢測、低功耗

4. 階層式喚醒流程

5. 詳細流程圖

6. 實際操作

- Debugging method
- 藉由麥克風取得 PCM data
- Burning NN model

7. 參考資料

介紹

此研究目的為 ”使**聲控裝置**以極低的功耗運作” 讓此裝置結合其他產品時可以很省電，

因此讓系統平時處於功耗極低的睡眠狀態，

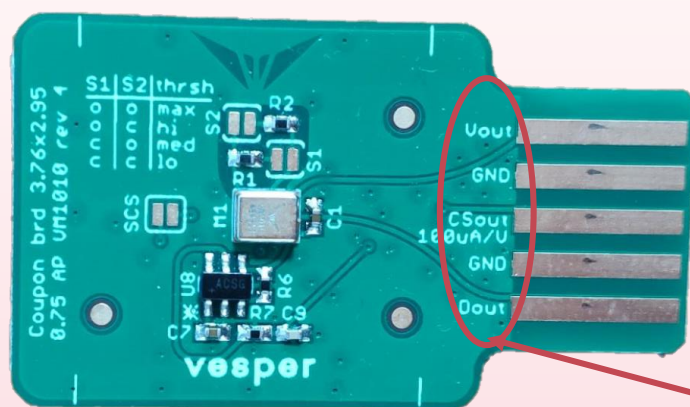
並透過**①**睡眠、**②**人聲判斷、**③**關鍵字辨識 **漸進式開啟** 來降低功耗



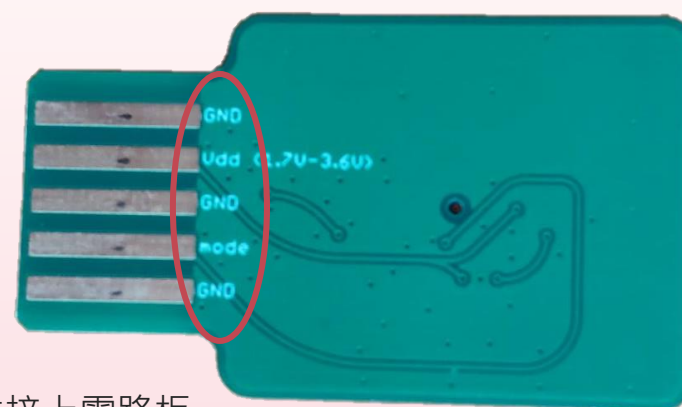
使用元件 (1/2)

Vesper VM1010 (壓電式麥克風)

正面



反面



後面會依照此腳位接上電路板
實作跳至 [p.14](#)

專有名詞解釋：

壓電式傳感器：基於某些介質材料的壓電效應，是典型的有源傳感器。當材料受力作用而變形時，其表面會有電荷產生，從而實現非電量測量

VM1010 介紹

麥克風平時會是睡眠模式(sleep)，所需電流為 10 μ A

喚醒方式為當音量超過 89dB 時，麥克風就會被喚醒為工作模式(normal)，

此時所需電流為 85 μ A

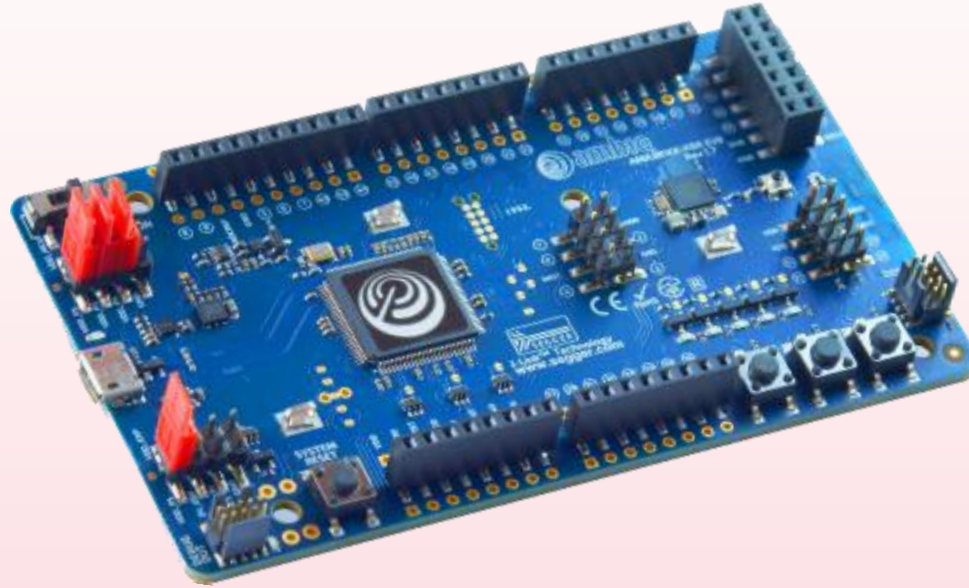
Mic 模式	電流大小 (μ A)
Sleep	10
Normal	85

備註：

89dB 的音量大小像是：繁忙的城市街道、通過的摩托車、狗叫聲

使用元件 (2/2)

Ambiq Apollo3 Blue MCU



此為使用的電路版

專有名詞解釋：

MCU 微控制器 (Microcontroller Unit)：一顆完整的MCU，會將CPU、RAM、ROM、I/O 或A/D等周邊關於記憶與運算功能整合在一起，簡單來說，MCU已經是一個微型的電腦

Apollo3介紹

具有 1MB flash 與 384kB SRAM 足以提供我們儲存音訊資料及 DNN 參數

並且支援 DMA，將錄得的音訊資料直接存至記憶體，無須依賴 CPU

內建 ADC 功能，可將麥克風收到的類比訊號做處理

而 Apollo3 在不同模式下的電流大小為

Apollo3 模式	電流大小 (μA)
Sleep	1
Normal	288
Turbo (超頻)	576

專有名詞：

SRAM 靜態隨機存取記憶體 (Static Random Access Memory)：此記憶體只要保持通電，裡面儲存的資料就可以恆常保持

DMA 直接存取記憶體 (Direct Memory Access)：可以獨立地直接讀寫系統記憶體，而不需中央處理器 (CPU) 介入處理

ADC (Analog-to-digital converter)：類比數位轉換器

類比訊號：可將聲音大小轉換成電壓大小，並由連續的訊號表示

數位訊號：由 0 與 1 兩種不連續訊號所組成

技術需求

1. 人聲語音識別 (VAD)

- 訓練方式是以 DNN 模型為基礎實作 VAD

2. 關鍵字檢測 (KWS)

- 先辨識是否為 VAD 後，才做關鍵字辨識
- 同樣使用 DNN 訓練模型

3. 低功耗 (語音辨識運算量大，所以相當耗電)

- 挑選了功耗低的 Apollo3 板子
- 使用階層式喚醒的方式降低待機時耗電問題

專有名詞解釋：

VAD 語音活性檢測 (Voice Actication Detection)：是一項用於語音處理的技術，目的是檢測語音訊號是否存在

DNN 深度神經網路 (Deep Neural Networks)：對資料進行表徵學習的演算法

KWS 關鍵字識別 (Keyword spotting)：像是 "hey Siri" 一樣的功能

階層式喚醒流程

1. 初始設定 (init)

2. 睡眠模式 (sleep)

```
if(麥克風偵測到聲音){  
    進入 recording  
}
```

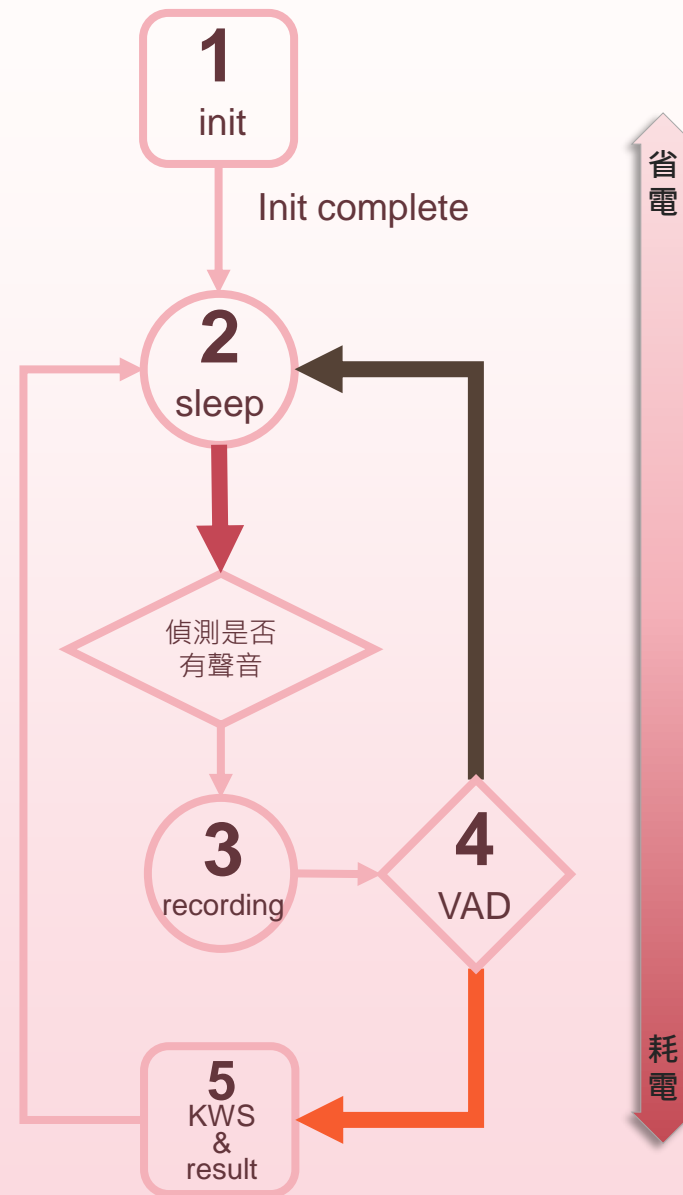
3. 錄音 (recording) 並萃取聲音特徵(錄音到 120ms 時進入步驟 4)

4. 人聲判別

```
if(是人聲){  
    就繼續錄音滿 1s 並到第 5 步驟  
}
```

```
else(不是人聲){  
    回到第 2 步驟  
}
```

5. 辨識關鍵字並用 LED 顯示結果



詳細流程圖 (1/4)

1. 初始設定 (開機後)
2. 睡眠模式 (常駐，等待中斷 GPIO_ISR)

```
if(麥克風偵測到聲音大於 89 dB){  
    進入第 3 步驟  
}
```

3. 錄音並萃取聲音特徵 (錄音到 120ms 時進入步驟 4)

取樣時以 16kHz(16000Hz) 進行 ADC 取樣，得到的數值由 DMA 存進 SRAM 的緩衝區，且要錄滿 120ms 才會進到下一步驟。
因為每 40ms 取樣一次，所以在進入到步驟4之前會取樣 3 次

```
if(緩衝區取得一個 40ms 的音框){  
    觸發中斷程式 (ADC_ISR)  
    進行直流濾除  
    MFCC 擷取特徵  
}
```

我們引用 Github 專案 Hello Edge 中的 MFCC 函式，Hello Edge 是在微控制器上實作 KWS 的 Github 專案

專有名詞解釋：

GPIO 通用型之輸入輸出 (General-purpose input/output)：接腳可以供使用者由程式控制自由使用

ISR 中斷服務 (interrupt service routine)：負責處理中斷的一種專用服務程式

16kHz：1 Hz 代表每秒取樣一次，而 16000 Hz 就是一秒取樣 16000 次，換句話說，就是把一秒分割成 16000 點

MFCC 梅爾頻率倒譜系數 (Mel-Frequency Cepstral Coefficients)：由梅爾倒頻譜的係數組成

梅爾頻率倒譜：一個可用來代表短期音訊的頻譜

詳細流程圖 (2/4)

呈上頁的判斷式

```
if(緩衝區取得一個 40ms 的音框){
```

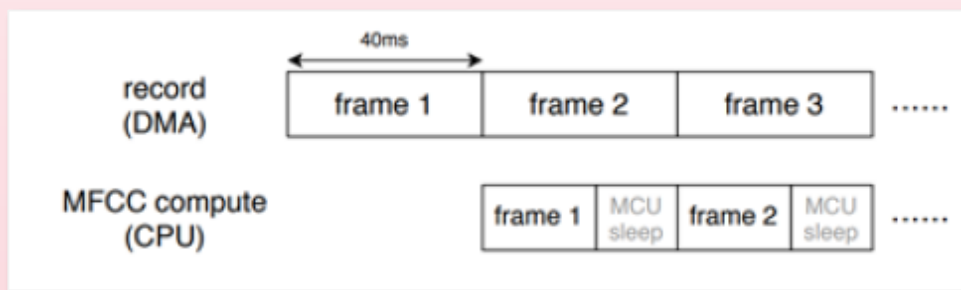
觸發中斷程式 (ADC_ISR)

進行直流濾除

MFCC 擷取特徵

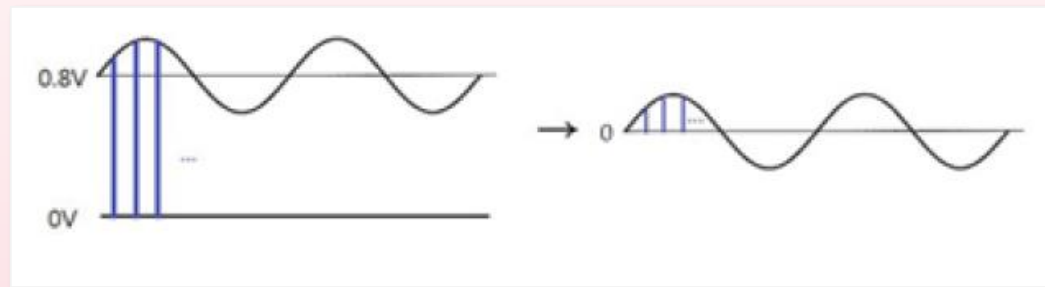
```
}
```

錄完 40ms 並完成上述步驟後，此時 SRAM 的緩衝區會存有一個音框 (時長 40ms) 的 PCM 音訊資料，並將這個音框進行 MFCC 計算後會得到 10 個特徵，而要進到第四步驟，需錄滿 120ms，因此會得到 30 個特徵值



錄音與MFCC計算的時序圖

觸發 ADC_ISR 中斷程式，也就是打斷 ADC 休息的狀態，開始執行 ADC，左下藍色直線為取樣結果，而音訊若要以 PCM 格式儲存，需轉換成右下，而濾除前後差異在基準值 (0) 不同



直流成分濾除前後示意圖 (左為濾除前，右為濾除後)

專有名詞：

PCM 脈波編碼調變 (Pulse-code modulation)：是一種類比訊號的數位化方法

詳細流程圖 (3/4)

4. 人聲判別 (VAD)

```
if(是人聲){  
    就繼續錄音滿 1s 並到第 5 步驟  
}  
else(不是人聲){  
    回到第 2 步驟  
}
```

DNN 架構設定：

參數	設定值
Input Layer	30 節點
Hidden Layer	16 x 16 x 16
Output Layer	語音、非語音
Training data : test data	33615 : 5026 (6 : 1)
Learning rate	0.0005

- 語音活性偵測 (VAD)：

我們利用 DNN 實作，此 DNN model 能判斷一段 120ms 的音訊屬於語音或非語音，沒有判斷一秒是否為語音是為了及早判斷音訊是否為語音，以免浪費功耗，而此 DNN model 的 input 是被 MFCC 解析完的 120ms 音訊，為 30 個特徵值，output 為語音、非語音。

- DNN 資料：

資料定義：將語音定義為一個人以正常朗讀或談話的語調對著麥克風發出的聲音，而其餘皆判斷為非語音 (包含餐廳的吵雜人聲)

資料集來源：語音資料庫來源分為 GSCD 以及 MS-SNSD 皆可劃分為語音及非語音兩個部分，GSCD 提供單詞語音，MS-SNSD 則提語句，兩資料集共提供數十種環境聲音，包含「白雜訊」、「粉紅雜訊」、「日常噪音」、「雨水等自然聲音」等

專有名詞：

GSCD (Google Speech Commands Dataset)：音訊資料集

MS-SNSD (Microsoft Scalable Noisy Speech Dataset)：資料集

詳細流程圖 (4/4)

5. 辨識關鍵字並用 LED 顯示結果

DNN 架構設定：

參數	設定值
Input Layer	250 節點
Hidden Layer	144 x 144 x 144
Output Layer	4 種關鍵詞 + unknown
Training data : test data	90% : 10%
Training steps	10000, 10000, 10000
Learning rate	0.0005, 0.0001, 0.00002

在第 4 步驟確認是人聲後，會錄滿 1s 並進入第 5 步驟，而 MFCC 會將每 40ms 的音訊產生 10 個特徵值 ($40 \times 25 = 1000$)，所以總共會有 250 個特徵值當作 input 輸入

關鍵字檢測，自定義了 4 種 (KWS)：

我們也是利用 DNN model 實作，也參考至 Github 專案 Hello Edge，它能判斷 1s 的音訊是否屬於我們自定義的關鍵詞 (人聲)「up」、「down」、「left」、「right」，若是人聲但為未定義單詞則顯示「unknown」

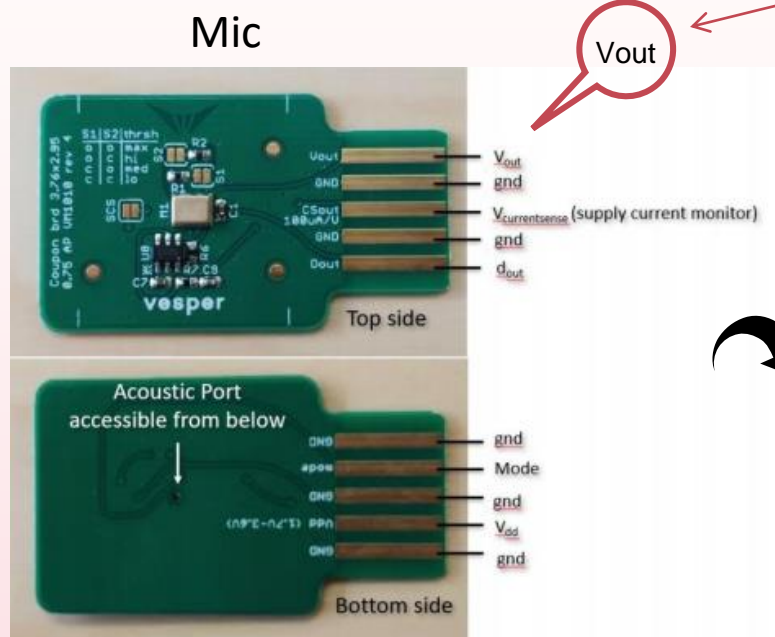
資料集來源：

與 VAD 使用同一個資料集 GSCD，其中包含「up」、「down」、「left」、「right」、「go」、「stop」等共 35 種關鍵詞，每個字約有 3000 筆錄音，由不同男女錄製而成

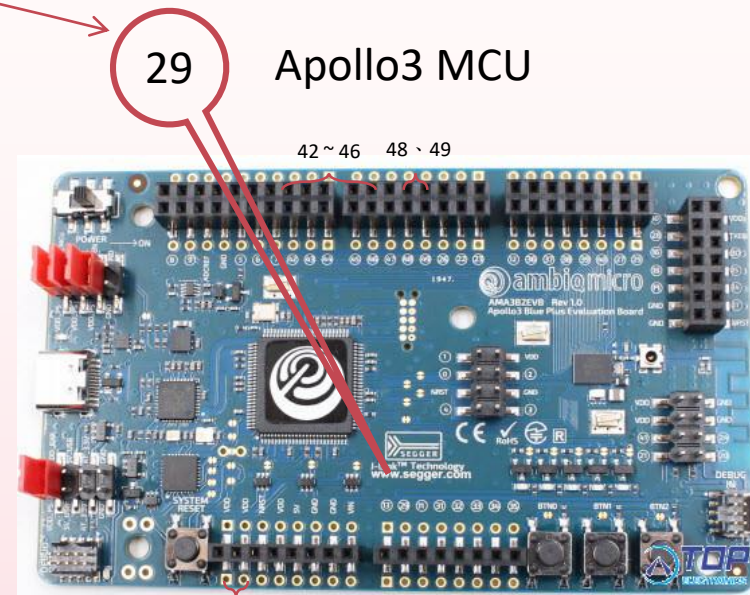
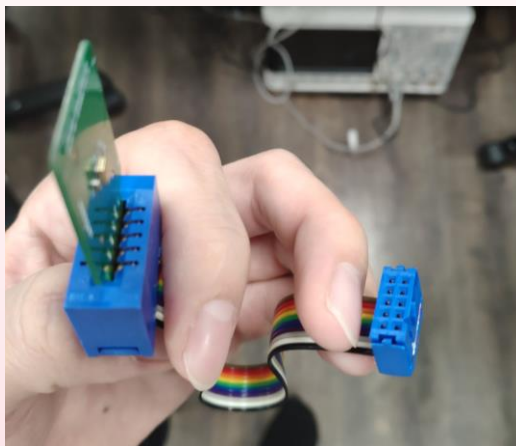
燈輸出顯示結果：

則是使用 5 顆燈泡作為表示，分別代表「up」、「down」、「left」、「right」，以及是否為人聲「VAD」，故如果今天說“yes”則單亮 VAD 一顆燈泡，如果說“up”則亮 VAD + up 兩顆燈泡，亮暗規則是說一聲關鍵字亮起，第二聲則關閉

實際操作 (1/8)



排線左邊對應左邊，右邊對應右邊



板子上有許多 VDD 與 GND

VM1010 對應腳位：

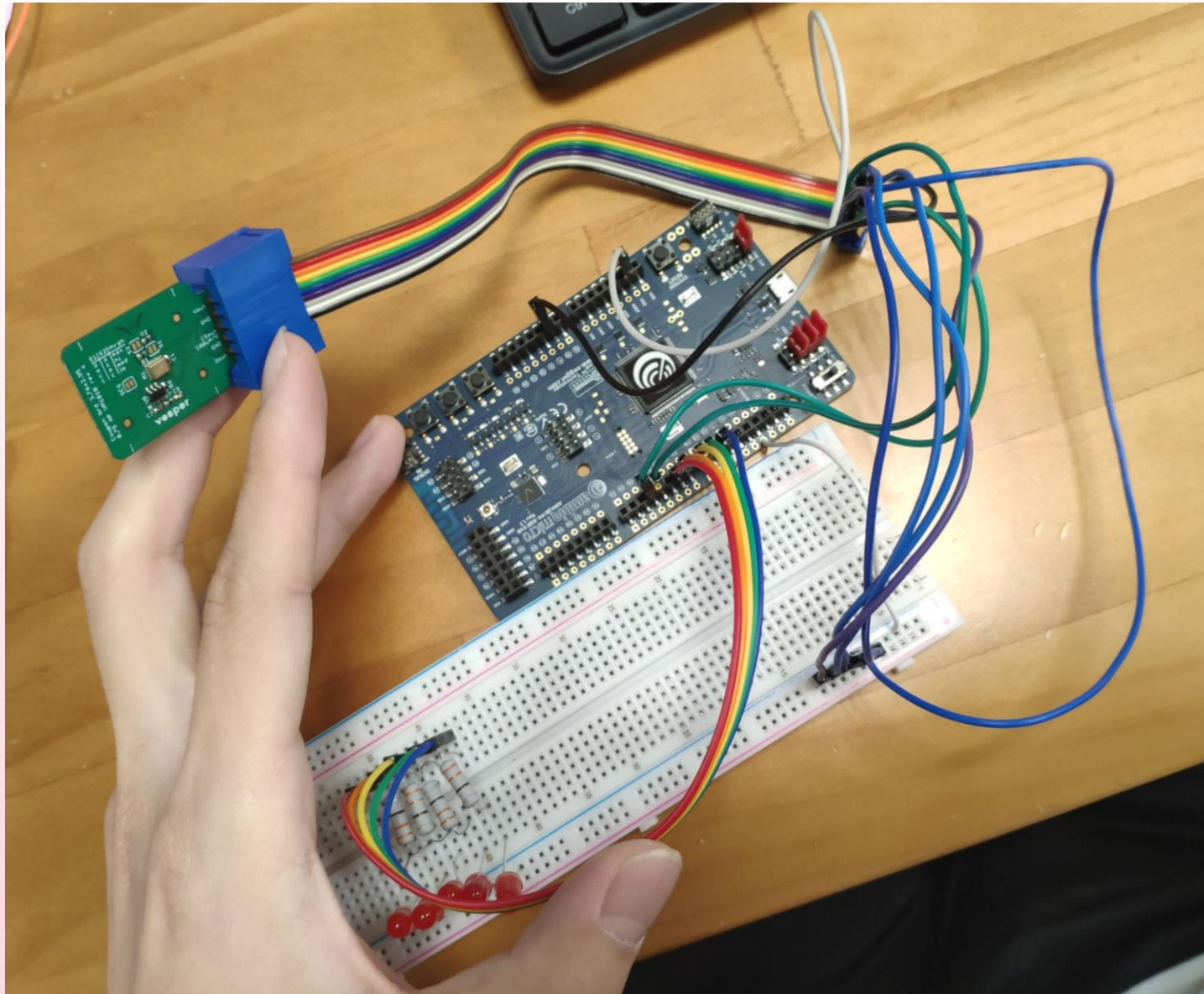
Mic	Apollo3 MCU
VDD	VDD
GND 接地	GND 接地
ex : Vout	29
Mode	48
Dout	49
CSout	不用接

LED 燈對應腳位：

LED 燈	Apollo3 MCU
up	42
down	43
left	44
right	45
VAD	46



實際操作 (2/8)



完成圖



實際操作 (3/8)

接著點選連結，下載 Keil，Keil 是微控制器軟體的開發平臺，可編譯、執行等功能

<https://www.keil.com/demo/eval/arm.htm#/DOWNLOAD>

Product Information

Software & Hardware Products

- Arm Development Tools
- C166 Development Tools
- C51 Development Tools
- C251 Development Tools
- Debug Adapters
- Evaluation Boards
- Product Brochures
- Newsletters

Device Database®

Device List

Compliance Testing

ISO/ANSI Compliance

Validation and Verification

Distributors

Overview

Home / Product Downloads

MDK-ARM

MDK-ARM Version 5.33
Version 5.33

Problems were detected with the information submitted. Please correct these issues and re-submit the form.

- Please select your country/region.

Enter Your Contact Information Below

First Name:

Last Name:

E-mail:

Company:

Job Title:

Country/Region:

Phone:

☐ Send me e-mail when there is a new update.

NOTICE:
If you select this check box, you will receive an e-mail message from Keil whenever a new update is available. If you don't wish to receive an e-mail notification, don't check this box.

Which device are you using?
(eg, STM32)

Arm will process your information in accordance with the Evaluation section of our [Privacy Policy](#).

☐ Please keep me updated on products, services and other relevant offerings from Arm. You can change your mind and unsubscribe at any time.

arm KEIL

Home / Product Downloads

MDK-ARM

MDK-ARM Version 5.33
Version 5.33

- Review the [hardware requirements](#) before installing this software.
- Note the [limitations of the evaluation tools](#).
- Further installation instructions for MDK5

(MD5:1c06594006dd0bde9e492f9f1e2cf3bd)

To install the MDK-ARM Software...

- Right-click on **MDK533.EXE** and save it to your computer.
- PDF files may be opened with Acrobat Reader.
- ZIP files may be opened with PKZIP or WINZIP.

MDK533.EXE (945.880K)
Wednesday, November 18, 2020

- If you are evaluating the tools, be sure to [request a quote](#) for the full version of the tools.

Products

- Development Tools
- Arm
- C166
- C51
- C251
- µVision IDE and Debugger

Hardware & Collateral

- ULINK Debug Adaptors
- Evaluation Boards
- Product Brochures
- Device Database
- Distributors

Downloads

- MDK-Arm
- C51
- C166
- C251
- File downloads

Support

- Knowledgebase
- Discussion Forum
- Product Manuals
- Application Notes

Contact

- Distributors
- Request a Quote
- Sales Contacts

Cookie Settings | Terms of Use | Privacy | Accessibility | Trademarks | Contact Us | Feedback

Copyright © 2005-2019 Arm Limited (or its affiliates). All rights reserved.

arm

此為網址點開頁面，填完資料後按下 Submit

(Next Page)
點下 EXE 檔進行下載



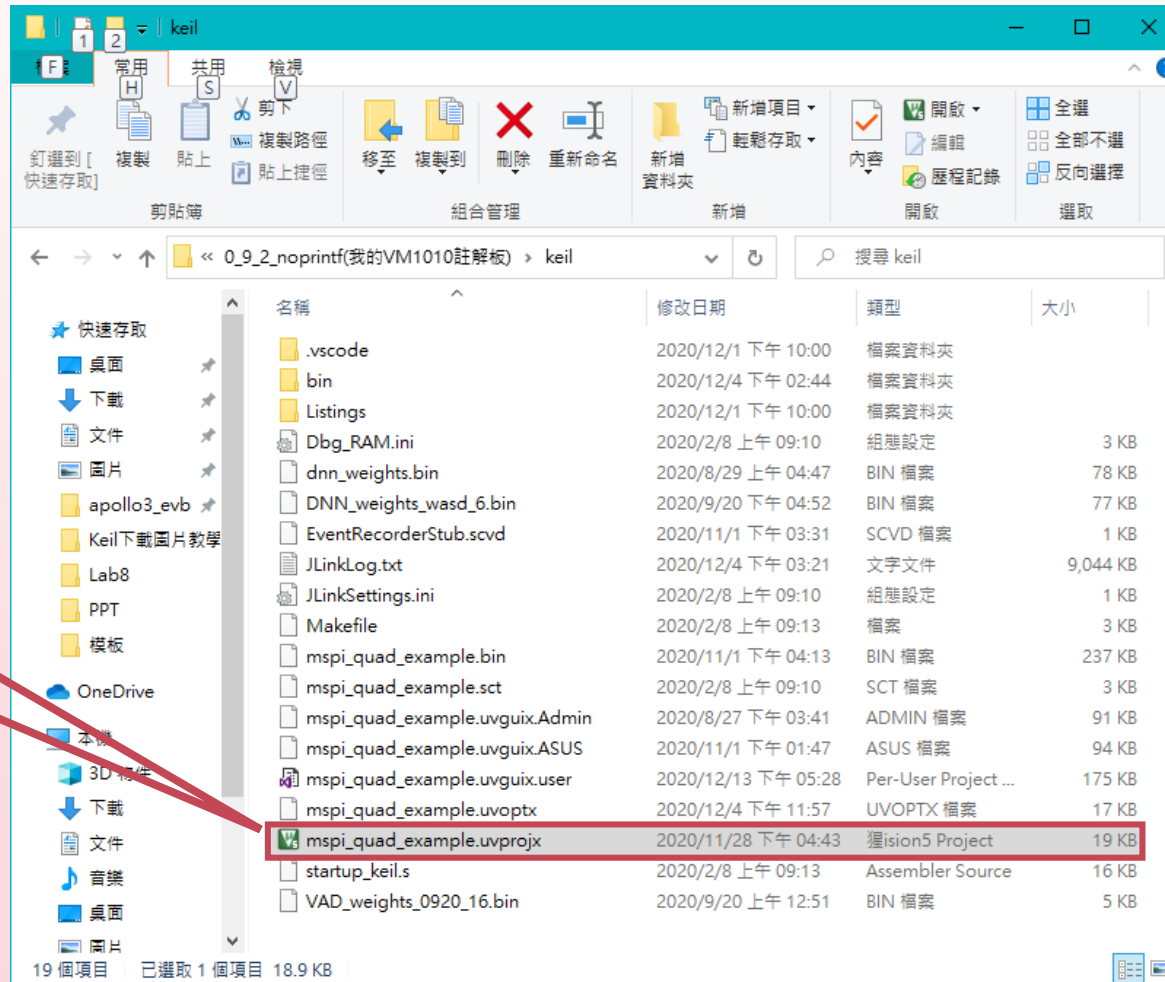
實際操作 (4/8)

接著點開資料夾裡的專案檔

Voice_Switch_20201124 → AmbiqSuite-SDK-Version-2.4.2 → AmbiqSuite-R2.4.2 → boards
→ apollo3_evb → examples → 0_9_2_noprintf (選擇VM1010) → keil → mspi_quad_example.uvprojx



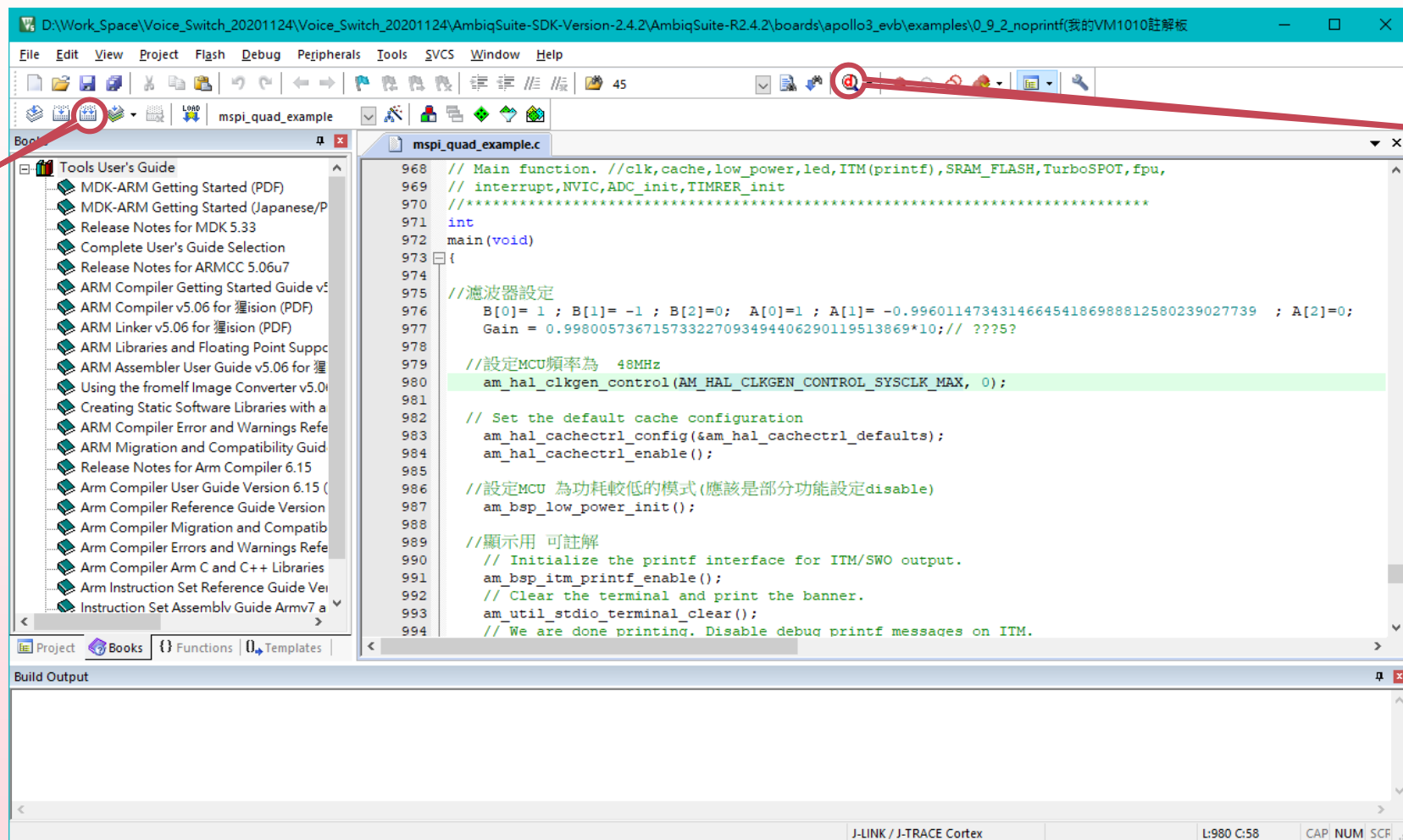
keil



實際操作 (5/8)

點開專案檔後

燒錄全部檔案



Debug 用

<https://www.jb51.net/softs/566199.html#downintro2>

跟著教學安裝破解版



實際操作 (6/8)

```
File Edit Selection View Go Run Terminal Help mspi_quad_example.c
C VC.c C mspi_quad_example.c X
D:\Work_Space> Voice_Switch_20201124 > Voice_Switch_20201124 > AmbiqSuite-SDK-Version-2.4.2 > AmbiqSuite-R2.4.2 > boards > apollo3_evb > exam
845 int LED_state;
846 switch (max_ind){
847     case 2:
848         am_hal_gpio_output_toggle(42); //up
849         LED_state = am_hal_gpio_output_read(42);
850         //am_util_stdio_printf ("up %d\n",LED_state);
851         break;
852     case 3:
853         am_hal_gpio_output_toggle(43); //down
854         LED_state = am_hal_gpio_output_read(43);
855         //am_util_stdio_printf ("down %d\n",LED_state);
856         break;
857     case 4:
858         am_hal_gpio_output_toggle(44); //left
859         LED_state = am_hal_gpio_output_read(44);
860         //am_util_stdio_printf ("left %d\n",LED_state);
861         break;
862     case 5:
863         am_hal_gpio_output_toggle(45); //right
864         LED_state = am_hal_gpio_output_read(45);
865         //am_util_stdio_printf ("right %d\n",LED_state);
866         break;
```

```
910 if(voice == 0 )
911 {
912     ///////////////////////////////////////////////////turn off mic////////////////////////////////////
913     am_hal_gpio_output_set(48); //mode = 1
914     am_devices_led_off(am_bsp_pLEDs, 4);
915 }
```

```
1037 am_hal_gpio_pinconfi( 29 , gpio_fg_t_ain); //Vout
1038
1039 //2. 麥克風 喚醒訊號 (數位輸入 ((中斷信號)))
1040 am_hal_gpio_pinconfi( 49 , gpio_fg_t_in); //dout
1041 //GPIO中斷信號設置
```

開啟程式碼後，可以看到 [p.14](#) 頁所插的腳位在程式裡都有設定

VM1010 對應腳位：

Mic	Apollo3 MCU
VDD	VDD
GND 接地	GND 接地
Vout	29
Mode	48
Dout	49
CSout	不用接

LED 燈對應腳位：

LED 燈	Apollo3 MCU
up	42
down	43
left	44
right	45
VAD	46

實際操作 (7/8)

LED 規則說明

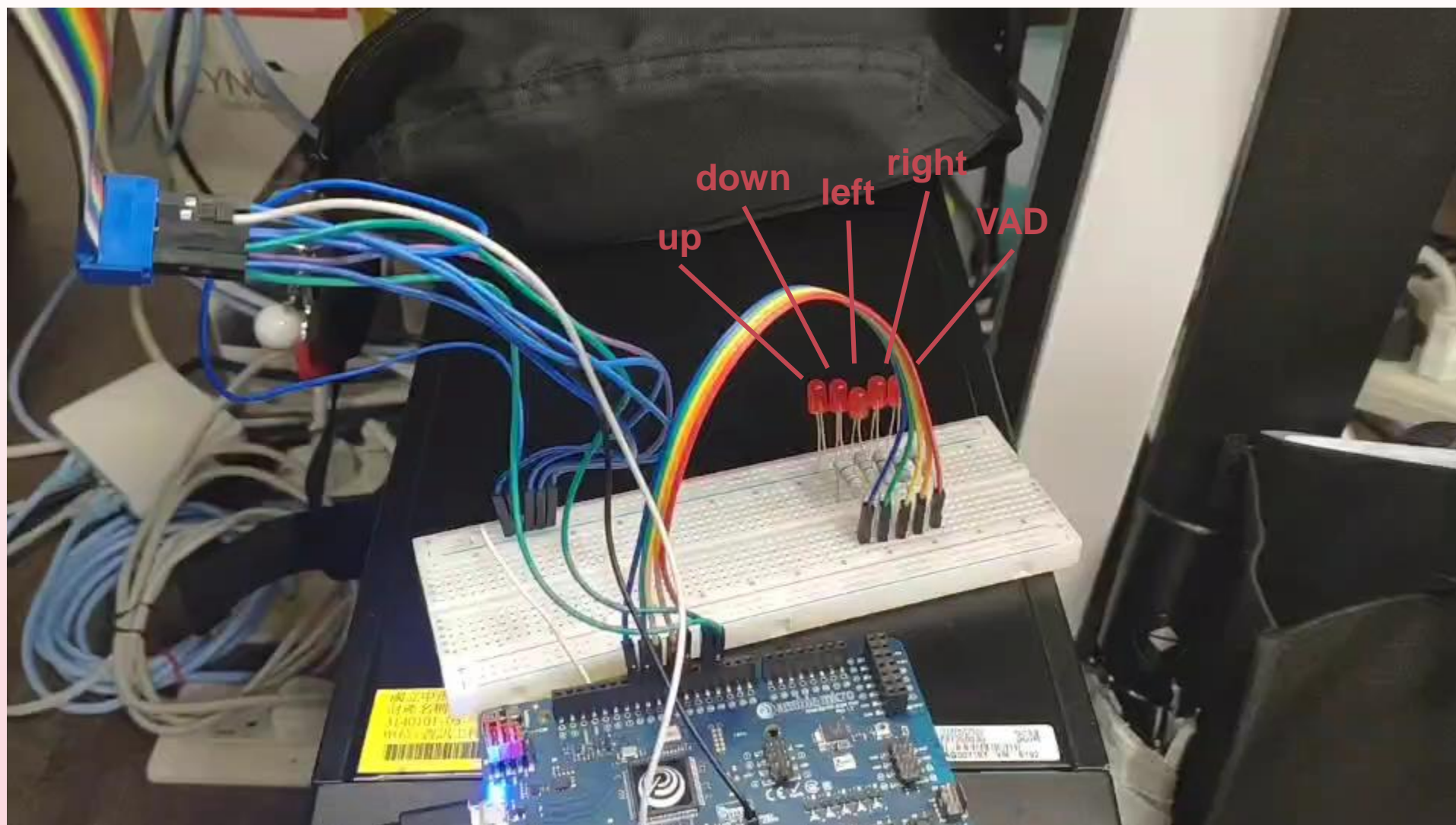
LED 燈輸出顯示結果：

則是使用 5 顆燈泡作為表示，分別代表「up」、「down」、「left」、「right」，

以及是否為人聲「VAD」，故如果今天說“yes”則單亮 VAD 一顆燈泡，如果說“up”則

亮 VAD + up 兩顆燈泡，亮暗規則是說一聲關鍵字亮起，第二聲則關閉

實際操作 (8/8)



其中 left 的辨識率最差



Debugging method

除錯對象：杜邦線、LED 燈、VM1010

- **杜邦線**：將三用電表調至蜂鳴檔 (如圖)，連接兩端直到發出 “逼” 聲，檢查是否短路。
- **LED 燈**：利用 MCU 上的 VDD 針腳，給予 LED 電源，檢測是否正常。
- **VM1010**：
 1. 利用**電源供應器**給予對應電壓，量測值是否與 datasheet 相同，使用時再請教學長。
 2. 利用**邏輯分析儀**量測 Mode、Dout、Vout 值，查看麥克風運作行為是否正常，需要使用時一樣請教學長。
 3. 利用程式將讀入音檔存成 PCM data 並直接聆聽，跳至 [藉由麥克風取得 PCM data](#)。



ELECTRICAL SPECIFICATIONS						
All specifications are at 25°C, VDD = 1.8 V unless otherwise noted						
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Units
Normal Mode Specifications						
Supply Voltage	VDD		1.6	1.8	3.6	V
Supply Current	IDD	VDD ≤ 1.8 V, Mode = Low		85	100	μA
Power Supply Rejection Ratio	PSRR	VDD = 1.8, 1kHz, 200mV _r Sine wave		53		dB
Power Supply Rejection	PSR	VDD = 1.8, 217Hz, 100mV _r square wave, 20 Hz – 20kHz, A-weighted		-85		dB(A)
Output Impedance	ZOUT			1		kΩ
Output DC Offset				0.8		V
Wake on Sound Specifications						
Supply Voltage	VDD		1.6	1.8	3.6	V
Supply Current	IDD	Mode = High	7	10	14	μA

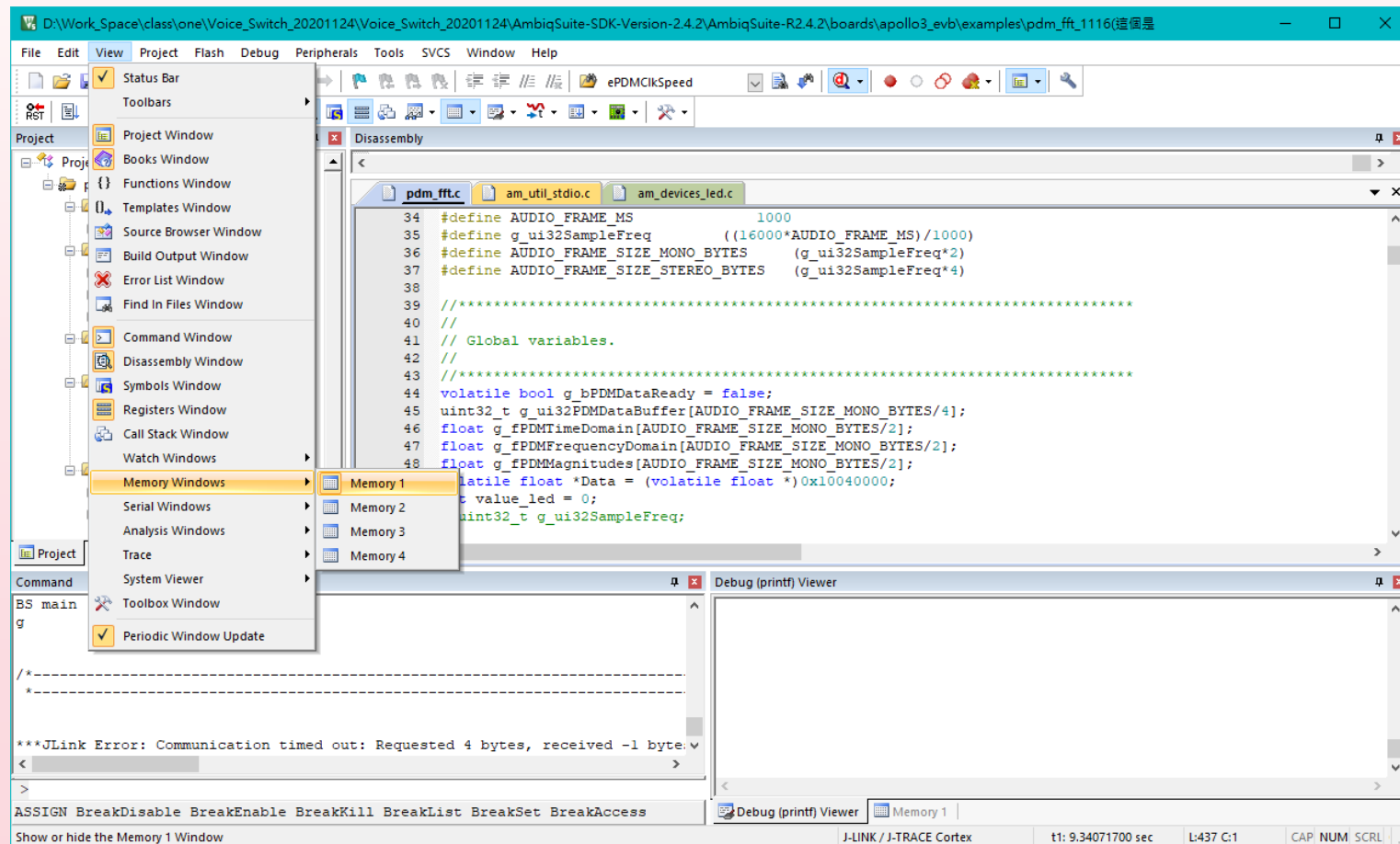
[VM1010_Datasheet](#)

藉由麥克風取得 PCM data

資料夾中有個文件檔為 **把pcm資料存至PC.txt**

程式中音檔儲存位置為 0x10040000，資料型態為 float

點選 keil 工具列：View -> Memory window -> Memory 1 輸入 0x10040000 即可查看記憶體值



藉由麥克風取得 PCM data

並按照資料夾 把pcm資料存至PC.txt 的步驟操作

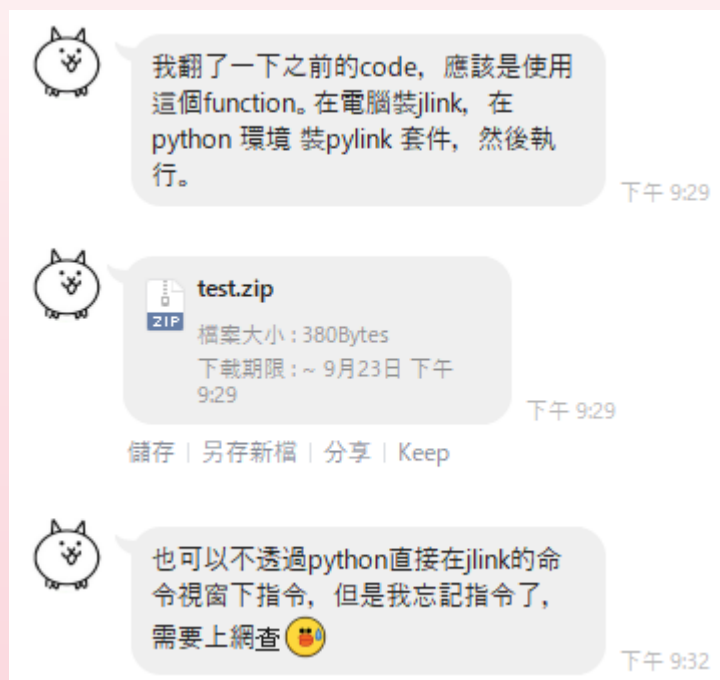
開啟同樣在資料夾中一個叫做 Jlink_commander 的資料夾，開啟 Jlink.exe，並按步驟操作

1. SWOSpeed
2. 選擇板子型號(AMA3B1KK-KBR) or 按 Enter 選擇預設
3. 1000 (Keil 與板子連接的速率？可仔細看 Keil 中 Command 視窗會顯示)
4. 輸入以下指令儲存至指定位置
 - SAVEBIN 指定位置\檔名.pcm 音訊資料位置, 儲存大小
 - e.g. SAVEBIN D:\0918.pcm 0x10040000, 0x10000

Burning NN model

燒錄方法：(這邊因為是冠良與泰翔學長操作，我沒有參與，因此無細部內容，請自行操作)

1. 參考網址：[J-Flash的Hex/Bin文件的烧录_DylanHe215的博客-CSDN博客](#)
2. 使用下面專題生給予的方法，檔案需要再跟我說



參考資料

1. D:/Work_Space/Voice_Switch_20201124/Voice_Switch_20201124/第貳章_專題_含分工(專題生報告).pdf
2. https://ambiq.com/wp-content/uploads/2020/09/Apollo3_Blue_MCU_Data_Sheet_v0_13_0.pdf
3. https://vespermems.com/wp-content/uploads/2019/03/VM1010_Datasheet-1.pdf
4. <https://drive.google.com/drive/folders/16VUDzmlrKOHxPmfLPQpWJqyssviEFTpJ>
5. <https://kknews.cc/zh-tw/news/mrxyq32.html>
6. <https://zh.wikipedia.org/wiki/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0>
7. <https://zh.wikipedia.org/wiki/%E5%99%AA%E5%A3%B0%E7%9A%84%E9%A2%9C%E8%89%B2#%E7%B2%89%E7%BA%A2%E5%99%AA%E5%A3%B0>
8. <https://ithelp.ithome.com.tw/articles/10204032>
9. <https://luckylong.pixnet.net/blog/post/6512881>
10. <https://zh.wikipedia.org/wiki/%E9%A1%9E%E6%AF%94%E6%95%B8%E4%BD%8D%E8%BD%89%E6%8F%9B%E5%99%A8>
11. <https://zh.wikipedia.org/wiki/%E6%A2%85%E5%B0%94%E9%A2%91%E7%8E%87%E5%80%92%E8%B0%B1%E7%B3%BB%E6%95%B0>
12. <https://zh.wikipedia.org/wiki/%E8%84%88%E8%A1%9D%E7%B7%A8%E7%A2%BC%E8%AA%BF%E8%AE%8A>





MERRY
CHRISTMAS
&
HAPPY NEW YEAR

謝謝大家

附錄 (兩周間的差異)

將使用到的元件放在一起講，整體比較連貫



附錄 (兩周間的差異)

原本的介紹頁面講解不清楚，加上了小圖示與粗體表示

介紹

此研究目的為如何讓聲控裝置以極低的功耗運作，

從而能不受限地安裝在各種產品上，

因此讓系統平時處於功耗極低的睡眠狀態，


並透過睡眠、人聲判斷、關鍵字辨識階層式喚醒機制來降低功耗

介紹

此研究目的為 "使**聲控裝置**以極低的功耗運作" 讓此裝置結合其他產品時可以很省電，

因此讓系統平時處於功耗極低的睡眠狀態，

並透過睡眠、人聲判斷、**關鍵字辨識** **漸進式開啟** 來降低功耗

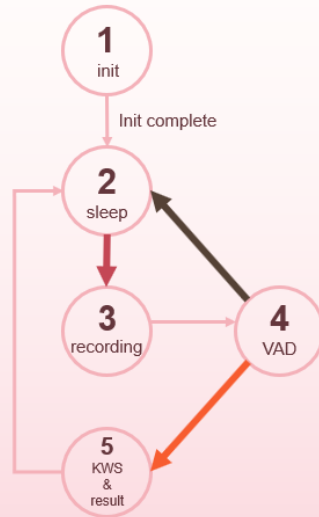


附錄 (兩周間的差異)

更改階層式流程圖，並補上耗電差異
從第 12 頁開始為新增頁數

MCU階層式喚醒流程

1. 初始設定
2. 睡眠模式
 - if(麥克風偵測到聲音){
進入第 3 步驟
3. 錄音並萃取聲音特徵(錄音到 120ms 時進入步驟 4)
4. 人聲判別
 - if(是人聲){
就繼續錄音滿 1s 並到第 5 步驟
 - else(不是人聲){
回到第 2 步驟
5. 辨識關鍵字並用 LED 顯示結果



階層式喚醒流程

1. 初始設定 (init)
2. 睡眠模式 (sleep)
 - if(麥克風偵測到聲音){
進入 recording
3. 錄音 (recording) 並萃取聲音特徵(錄音到 120ms 時進入步驟 4)
4. 人聲判別
 - if(是人聲){
就繼續錄音滿 1s 並到第 5 步驟
 - else(不是人聲){
回到第 2 步驟
5. 辨識關鍵字並用 LED 顯示結果



9