

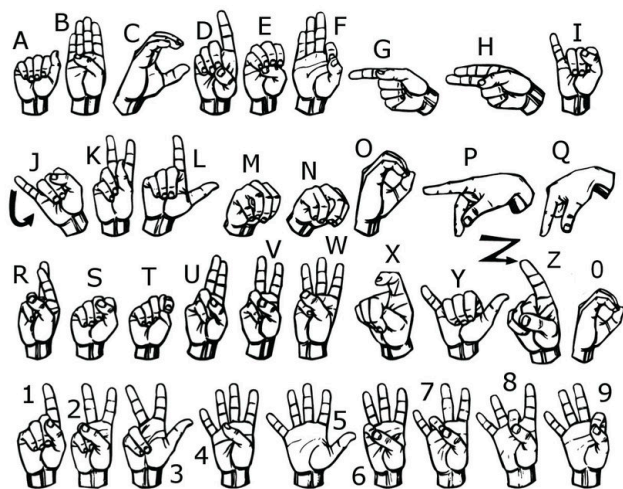
電工實驗 生醫工程 期末專題報告

組別:G1 組員: 李柏宇 簡敦賢 林禹融 蕭銜甫

一、研究動機 (為什麼想做這個題目、臨床需求、價值...)

現今市面上存在許多方便的翻譯軟體，支持各種語言，且準確度高，使用不同語言的人也能交流無礙。然而跟據統計，全世界約有7000萬人用手語，但他們的126種語言卻沒有很好的即時翻譯功能。這除了令日常溝通不方便以外，對於緊急事故的場合，如醫院、派出所，無法即時有效溝通可能導致危險或遺憾發生。

因此在這次專題中，我們想要開發一個穿戴式裝置，穿在手上可以將手語翻譯，並呈現在手機上，來解決瘖啞人士生活的不便。我們選擇翻譯的手語是 American sign language，一種廣泛被採用的手語。我們希望能透過不同手勢每隻手指彎曲程度以及手掌方向的不同，準確翻譯26個字母透過藍芽傳輸翻譯結果，並結合MIT App Inventor 將結果呈現在手機App上，並希望能做到至少80%的準確度。這樣的裝置不僅滿足瘖啞人士日常的溝通需求，還在特殊場景中具有不可忽視的價值，是一個具有實際應用潛力的專題。



圖一、American sign language

二、研究方法 (流程圖、發想方法、解決方法...)

裝置硬體介紹

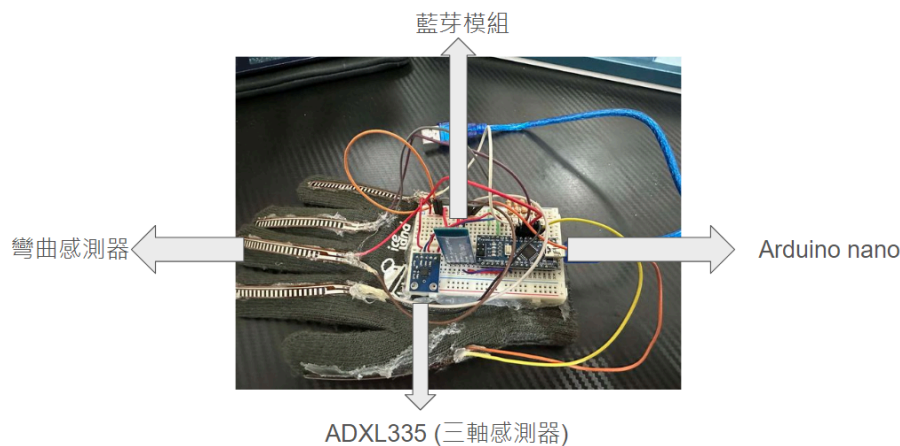
我們使用了Arduino Nano、彎曲感應器、藍牙模組、ADXL 335三軸加速度感測器來開發此專題，以下是它們的功能介紹：

Arduino Nano：作為主控板，負責整合感測器數據、進行運算處理，以及將手勢翻譯為字母。

彎曲感應器:測量每根手指的彎曲程度,用以判斷手勢中各手指的角度變化,作為字母辨識的主要數據來源。

藍牙模組:將翻譯結果透過無線方式傳輸到手機,實現即時顯示與溝通功能。

ADXL 335三軸加速度感測器:感測手掌的空間方位和動作方向,輔助判斷手勢中的方向性元素,提升翻譯準確度。



圖二、裝置圖

裝置運作流程

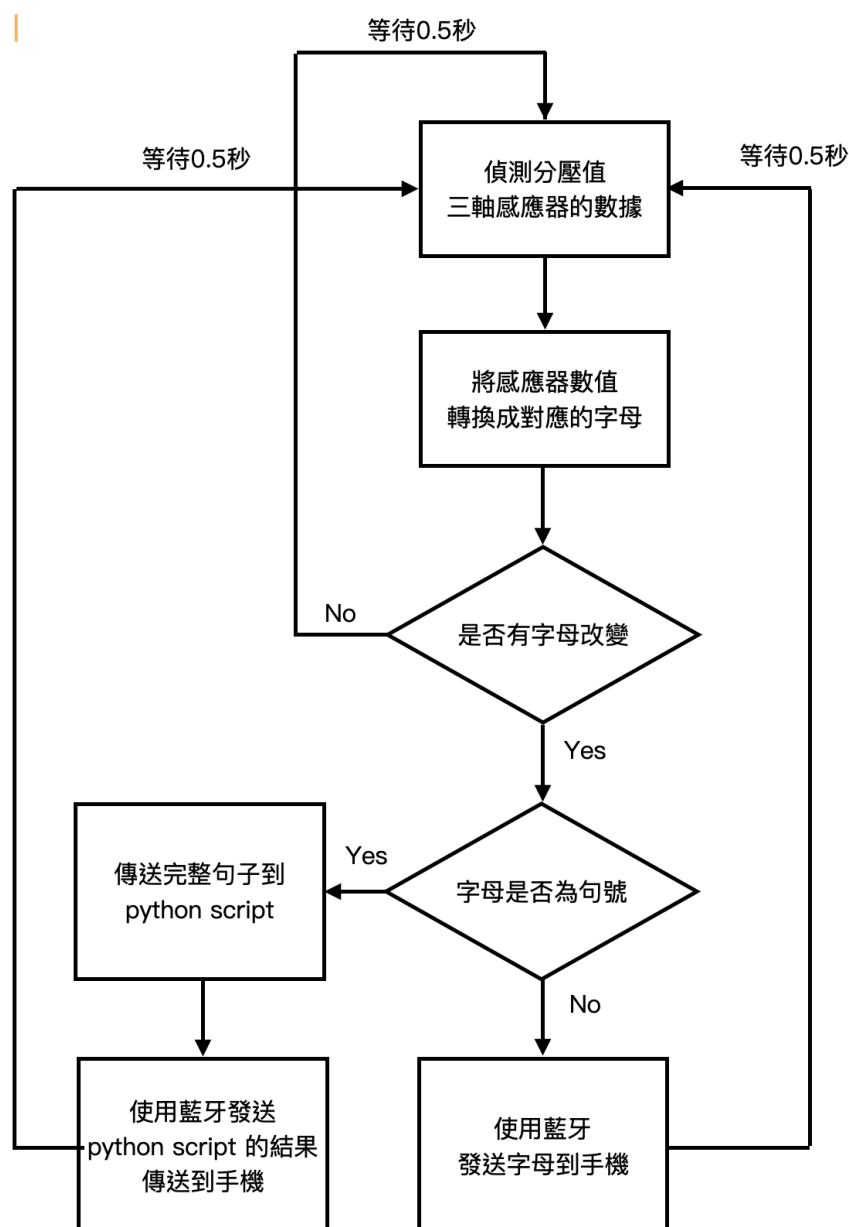
1. 利用arduino 偵測彎曲感應器的電壓值, 並且將五隻手指的電壓值轉換成它們分別的彎曲程度(0~100)。
2. 利用五隻手指的彎曲角度去對應它到這個手勢應該對應到的手語字母。
3. 如果此時手勢代表的字母和前一個字母不同, 就利用藍牙傳送這個新字母到手機上顯示。如果相同, 則等待0.5秒後, 再從第一步開始執行。
4. 如果手勢代表的是句號, 將完整的句子傳送到python script, 在其中有串接 Gemini 的api 可以將我們的句子做一次錯誤的偵測和改進, 利用藍牙將最後 python script 的輸出結果傳送到手機上顯示。
5. 重複執行第一點到第四點。

為了準確判斷代表手語的手勢, 我們設計了一套多感測器融合的系統, 結合 Arduino、五個手指上的 Flex Sensor 和三軸感應器 (ADXL 335), 提升手勢判斷的精準度。Flex Sensor 負責測量手指的彎曲程度, 將五隻手指的電壓值轉換為彎曲程度, 進而得出手勢的基本資訊。同時, 三軸感應器提供手掌角度與朝向的數據, 確保可以判斷的出手掌呈現的是水平或鉛直, 在某些手勢中至關重要, 如字母I和J。透過將五指角度與手掌朝向的數據進行結合, 我們構建了一個手勢映射表, 使系統能準確判斷每個手勢所代表的手語字母。

在識別過程中，當前手勢的字母會與上一個字母進行比較，若兩者不同，系統會即時透過藍牙模組將新字母傳送至手機顯示；若相同，則系統等待 0.5 秒後重新進行偵測，避免重複傳送。當手勢代表的字母為句號(.)時，表示句子的結束，系統會將完整句子透過藍牙傳送至 Python Script，並藉由串接 Gemini API 對句子進行錯誤檢測與改進。經修正後的句子會回傳至 Arduino，最終透過藍牙顯示於手機端。

此外，為了提升數據穩定性與系統效率，我們設定了合理的角度容差範圍，容忍微小的手指或手掌偏移，避免誤判。這樣的設計不僅提升了手勢識別的準確性，還能在實際應用中提供更流暢的用戶體驗，實現手語翻譯的即時性與高可靠性。

裝置運作流程圖：



實驗步驟

步驟一、初始化與準備工作

1. 硬體設置：

- 使用 Arduino 接收五隻手指的電壓值：在手套上的彎曲感測器 (Flex Sensor) 的電阻會隨著手指彎曲的程度而改變，再藉由和一個固定電阻做分壓，可以得到和手指彎曲角度有關的電壓值。
- 利用三軸感應器去監測手掌的角度和朝向，用於提升後續手勢判斷的精準度。
- 配置藍牙模組 (HC-05) 作為數據傳輸工具。
- 確認藍牙與手機的配對成功。

2. 軟體初始化：

- 在 Arduino 中設置電壓-角度的線性對應，將感測到的電壓值轉換為彎曲程度。
- 將手勢和對應的手語字母定義成映射表 (例如，A 對應的五指角度組合)。
- 初始化變數存儲當前手勢與前一次手勢。

步驟二、偵測手指彎曲程度並識別手語字母

1. 偵測電壓值並轉換為彎曲程度：

- 讀取五隻手指的電壓值，透過校準公式將其轉換為彎曲程度。

2. 手勢識別：

- 利用五指的彎曲程度以及三軸感應器的測量值查找映射表，判斷當前手勢對應的手語字母。

```
float flexADC1 = analogRead(FLEX_PIN1);
flexADC1 = constrain(flexADC1, sensorMin1, sensorMax1);
float angle1= 100 - map(flexADC1, sensorMin1, sensorMax1, 0, 100);
float flexADC2 = analogRead(FLEX_PIN2);
flexADC2 = constrain(flexADC2, sensorMin2, sensorMax2);
float angle2= 100 - map(flexADC2, sensorMin2, sensorMax2, 0, 100);
float flexADC3 = analogRead(FLEX_PIN3);
flexADC3 = constrain(flexADC3, sensorMin3, sensorMax3);
float angle3= 100 - map(flexADC3, sensorMin3, sensorMax3, 0, 100);
float flexADC4 = analogRead(FLEX_PIN4);
flexADC4 = constrain(flexADC4, sensorMin4, sensorMax4);
float angle4= 100 - map(flexADC4, sensorMin4, sensorMax4, 0, 100);
float flexADC5 = analogRead(FLEX_PIN5);
flexADC5 = constrain(flexADC5, sensorMin5, sensorMax5);
float angle5= 100 - map(flexADC5, sensorMin5, sensorMax5, 0, 100);
xadc = analogRead(xpin);
yadc = analogRead(ypin);
bool horizontal = ((xadc>=330) && (xadc<=400)) && ((yadc>=270) && (yadc<=300));
bool vertical = ((xadc>=270) && (xadc<=300)) && ((yadc>=340) && (yadc<=400));
bool equilibrium = (((xadc>=330) && (xadc<=400)) && ((yadc>=330) && (yadc<=400)));
```

圖三、讀取數值轉換

```

if(compare(flexADC1,460,40) && compare(flexADC2,540,40) && compare(flexADC3,600,40) && compare(flexADC4,580,40) && compare(flexADC5,600,40)){
    letter = "b";
}

if(compare(flexADC1,565,40) && compare(flexADC2,440,40) && compare(flexADC3,510,40) && compare(flexADC4,510,40) && compare(flexADC5,540,40) && !vertical){
    letter = "c";
}

if(compare(flexADC1,475,40) && compare(flexADC2,550,40) && compare(flexADC3,452,40) && compare(flexADC4,465,40) && compare(flexADC5,465,40)){
    letter = "d";
}

if(compare(flexADC1,515,40) && compare(flexADC2,350,40) && compare(flexADC3,605,40) && compare(flexADC4,590,40) && compare(flexADC5,605,40)){
    letter = "f";
}

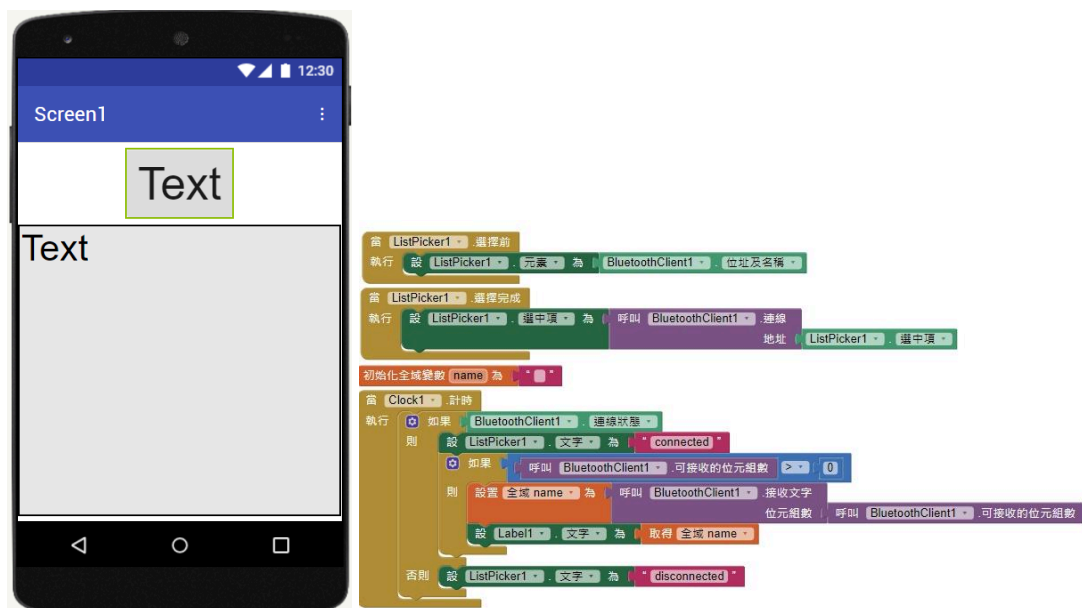
```

圖四、將角度跟方向轉成對應的文字

步驟三、傳送新字母到手機

1. 字母判斷與傳送：

- 比較當前手勢代表的字母與前一次的字母：
 - 如果不同，透過藍牙模組傳送新字母到手機顯示。
 - 如果相同，則等待 0.5 秒 再次進行偵測。



圖五、手機畫面與app inventor 程式，若連線成功，上方的text就會從 diconneceted變成connected。下方的text則會顯示收到的數值。

步驟四、判斷句點並進行文字處理

1. 檢測句點：

- 當手勢對應的字母為句號 (.) 時，將完整句子存儲並準備傳送到 Python Script。

2. 文字錯誤檢測與改進：

- 使用藍牙傳送完整句子到電腦端的 Python Script。
 - Python Script 串接 Gemini API:
 - 將句子進行錯誤檢測與改進。
 - 將最終結果回傳給 Arduino。
3. 回傳修正後句子到手機:
- 將改進的句子透過藍牙傳送到手機顯示。

```
def chat_with_gemini(message):  
    try:  
        api_key = 'Your_API_key'  
        genai.configure(api_key = api_key)  
  
        model = genai.GenerativeModel('gemini-1.5-flash')  
        response = model.generate_content(message)  
        return response.text  
    except Exception as e:  
        print(f"Unexpected error: {e}")  
        return "An error occurred while generating a response."
```

圖六、串接gemini, 回傳修改後的句子。

步驟五、重複流程

返回到步驟二，持續進行手勢識別與處理。

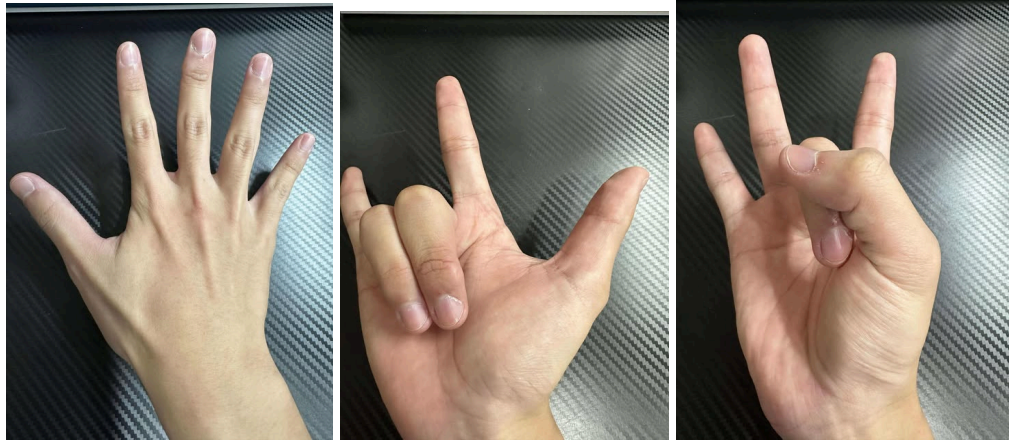
問題與解決方法:

問題一：此裝置需要即時翻譯，而EMG等上課中使用的生醫訊號無法即時，使用自製的電路卻體積過於龐大。

解決方法：經過討論我們認為此裝置並不適合採用EMG等複雜的生醫訊號下去分析，而適合一些較簡單直接的生醫訊號，最後是採用手指彎曲和手掌方向。

問題二：除了American sign language的手勢，還需要一些斷句的手勢，而且因為我們感應的是讀取字母是否有變化，若有兩個重複的字母(如ntuee中的ee)，就無法判斷這是兩個字母。

解決方法：我們另外設計了空白、句號、複製前一個字母的手勢來完善我們的系統，手勢如圖七。



圖七、新增的手勢，左到右分辨為空白、複製前一個字母、句號。

問題三:對於一些相似的手勢，在轉換過程中容易誤觸。

解決方法: 我們一開始僅設置好請語言模型修正拼字的命令而已，如圖八，但發現將手勢比出來的結果傳進去後，得到的回答與我們預期的相差甚遠，如主詞想要打出"it"但中間多一個"i" 或 "c"，則容易生成"ice"、be動詞想要打出"is"但中間多一個"a" 或 "r"，則容易生成"are"。後來經過不斷地測試，我們最終想出一個較完美且回答幾乎與預期一致的prompt，如圖九。我們制定一些規則，將一個主詞、be動詞容易混淆的部分清楚告訴語言模型，並在最後面舉一些實際轉換的例子，繃著語言模型更好掌握我們的input，並讓我們得到正確的回答。

```
problem = """Help me break the following series of letters into correct sentences and correct spelling errors.
Don't change the subject arbitrarily. Just give me the corrected answer and don't answer me."""
```

圖八、初始命令

```
rule = """Analyze the given text to identify the subject and the appropriate form of the "be" verb. Follow these rules:
1. If the subject is a pronoun, determine the most likely pronoun based on its length, structure, and context:
- Single-letter subjects are most likely "I".
- Only two or three letters and "i" appears without "t", it is most likely "I"
- Three to Five-letter subjects could be "he", "she", or "it", depending on the gender, singularity or spelling in the context.
- Subjects like "they" suggest a plural entity.
2. If the "be" verb appears in the text:
- If it contains "i" and "s", it is most likely "is".
- If it is associated with a plural subject, it is most likely "are".
- Match the verb form to the subject's grammatical number and person (singular/plural).

Examples:
- Input: "ci_likse_arpbbls."
- Output: "I like apples."
- Input: "hse iaxs fat."
- Output: "He is fat."
This is the input:
"""
```

圖九、設計的prompt

三、成果 (DEMO影片網址、結果、討論、總結、未來展望...)

DEMO影片網址: <https://youtu.be/xW549KzdgJQ>

結果: 我們自行選擇了一些目標句子進行測試, 結果如下表。

目標句子	未經python script	經python script
I need help.	i tneed khstselp.	I need help.
I need help. (in the demo video)	i noeed help.	I need help.
Today is cold.	cteodtaty is ceold.	Today is cold.
How are you.	how tsare yovu	How's are you?
I like apple.	ci lirkvkse applse.	I like apples.
Cat is eating.	cat is oeoatitg.	The cat is eating.
Thank you.	atuhatnkvk yovu.	Thank you.
Can you give me a book?	catn ycorvu givue nme a boovk.	Can you give me a book?
Can you give me a book? (in the demo video)	ocan you givue tmes a boovk.	Can you give me a book?
This is biomedical engineering lab.	cthits biomoedicatl setmngireeoevuritnmg lab	It is in the biomedical engineering laboratory.
This is biomedical engineering lab. (in the demo video)	thitst its biotsedioal setgineerintg lab	This is its biomedical engineering lab.

電腦終端顯示的結果如下(擷取部分):

```
before: How's are you?cat_is_oeoatitg.  
after:  How are you? The cat is eating.
```

```
before:  atuhatnkvk_yovu.  
after:  Thank you.
```

```
before:  how_tsare_yovu.  
after:  How's are you?
```

```
before:  Thank you.catn_ycorvu_givue_nme_a_boovk.  
after:  Thank you. Can you give me a book?
```

```
before:  Thank you.Thank you. Can you give me a book?cthits_biomoedicatl_setmngireeoevuritnmg_lab.  
after:  Thank you. Thank you. Can you give me a book? It is in the biomedical engineering laboratory.
```


討論：

根據上述及其他的測試結果，我們發現經python script所得到的結果，能完美修飾我們一開始得到的文句，但在一些地方仍有些微誤差，不過最終都能翻譯正確，達到一開始80%準確度的目標。

有的是語言模型給出的答案有文法錯誤，如"How are you?"變成"How's are you?"。有的是輸入造成的混淆，如我們輸入"I like apple."，手語轉換的結果為"ci lirkvkse applse."，最後卻因為"apple"不小心多了一個s，使受詞變成複數、輸入"This is biomedical engineering lab."，但因為手語判斷出現太多偵測失誤，使得其結果變成"This is its biomedical engineering lab."。

最大的誤差為我們輸入"This is biomedical engineering lab."後，經python script變成"It is in the biomedical engineering laboratory."，我們認為是手語判斷的結果與目標句子相差太多，而語言模型為了得到符合原本輸入及正確文法的句子才得出這樣的結果。

總結：

這次的實作，我們結合了手部特征、嵌入式系統、AI，共同完成了能將手語轉換成文字的裝置，是一個跨領域結合AI的產品。這能彌補啞啞人士打字時沒有語音輸入的缺憾，提供更方便、快速的文字呈現方式。

製作過程中，除了上述的「問題與解決方法」外，我們還遇到一個瓶頸，同時也是花最多時間的地方，便是測試比出每個字母的手部彎曲程度。我們完成初步設計後，對每個字母設定了一個基準範圍，而後便開始給身邊的同學、朋友嘗試使用，並根據他們的手勢彎曲程度來做取平均值並微調，最後對每個字母設計出最合適的判斷範圍。

未來展望：

儘管我們透過AI來幫我們修正輸入的文句，一些字母手勢仍然非常相似，在判斷上很容易辨識錯誤，在一些比較極端的情況下，若是我們的裝置持續判斷錯誤，丟進語言模型中潤飾仍可能會得到錯誤的結果。我們未來希望能綜合每個測試者的彎曲數值，將其整理完後丟進一些數學、統計模型，精準掌握每個字母的手勢彎曲角度，加強判斷能力，使得精確度能更高。

此外，在判斷較長、較複雜的單字時，若手語判斷出現些許失誤，也容易產生判斷錯誤。我們未來會強化prompt的設計，增強生成的判斷力，使我們遇到較長、較複雜的單字時也能正確判斷。

四、參考資料、文獻引用、分工

參考資料：

<https://newsroom.ucla.edu/releases/glove-translates-sign-language-to-speech>

文獻引用：

A. J. Abougarair and W. A. Arebi, "Smart glove for sign language translation," International Robotics & Automation Journal, vol. 8, no. 3, pp. 109–117, Dec. 2022. DOI: 10.15406/iratj.2022.08.00253.

H. Shaheen and T. Mehmood, "Talking Gloves: Low-Cost Gesture Recognition System for Sign Language Translation," 2018 IEEE Region Ten Symposium (Tensymp), Sydney, NSW, Australia, 2018, pp. 219-224, doi: 10.1109/TENCONSpring.2018.8692069.

J. Wu, L. Sun and R. Jafari, "A Wearable System for Recognizing American Sign Language in Real-Time Using IMU and Surface EMG Sensors," in IEEE Journal of Biomedical and Health Informatics, vol. 20, no. 5, pp. 1281-1290, Sept. 2016, doi: 10.1109/JBHI.2016.2598302.

R. Ambar, C. Kar Fai, M. H. A. Wahab, M. M. Abdul Jamil, and A. A. Ma'radzi, "Development of a wearable device for sign language recognition," Journal of Physics: Conference Series, vol. 1019, no. 1, p. 012017, 2018. DOI: 10.1088/1742-6596/1019/1/012017.

分工：

李柏宇：裝置設計實作、裝置測試、App inventor 程式、報告一、二部分

簡敦賢：裝置設計實作、實驗數據測量、報告二、三部分

林禹融：裝置設計實作、實驗數據測量、軟體處理

蕭銜甫：裝置設計實作、api程式、報告三部分

五、附件 (程式碼(github url)...)

github: https://github.com/Jackbear8868/BME_Final_G1

demo: <https://youtu.be/xW549KzdgJQ>