

地球物理通論 期末報告

中央氣象署

臺灣地震與地球物理資料管理系統

介紹

中央氣象署GDMS資料庫

如何繪製地震分布圖與地震波圖？

第五組

U11210009 林欣璇 / U11210027 陳沿伊

● GDMS介紹

GDMS (Geophysical Database Management System)，臺灣地區地震與地球物理資料管理系統，由中央氣象局、中央研究院地球科學研究所負責開發

GDMS透過連線的即時紀錄來完成例行性快速的地震測報、建置地震預警系統、地震防減災，累積的資料提供了好數據給 地震物理、地球內部結構、監測環境變化 等研究

透過國際標準的交換格式與自動化的資料處理流程，對全世界開放網路資料服務，包括測站查詢、觀測資料下載等

GDMS網址 - <https://gdms.cwa.gov.tw/>

GDMIS網頁介紹



最新消息

全部	系統訊息	應用成果
2024-12-23	系統已恢復資料下載服務	
2024-12-20	系統暫時停止資料下載服務	
2024-12-09	本系統恢復資料下載服務	
2024-11-27	12/09 - 12/11 系統暫停資料下載服務	
2024-09-23	系統將於9月24日10時起至9月26日12時 暫時停止服務	

[more...](#)

顯著有感地震

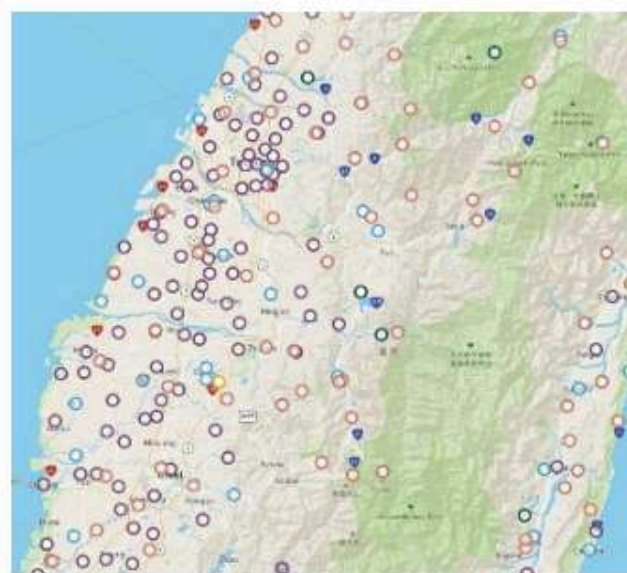
2024-08-15 23:35:53	M _L 6.30 發震時間 2024-08-15 23:35:53 資料下載
2024-04-26 18:21:24	M _L 6.10 發震時間 2024-04-26 18:21:24 資料下載
2024-04-22 18:32:49	M _L 6.30 發震時間 2024-04-22 18:32:49 資料下載
2024-04-22 18:26:51	M _L 6.00 發震時間 2024-04-22 18:26:51 資料下載
2024-04-03 02:14:35	M _L 6.20 發震時間 2024-04-03 02:14:35 資料下載

[more...](#)

GDMs網頁介紹

臺灣地震與地球物理資料管理系統

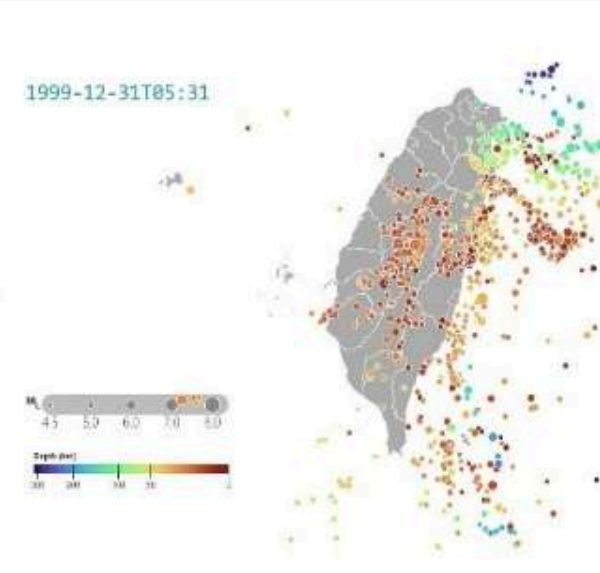
臺灣地震與地球物理資料管理系統(GDMS)是由中央氣象署與中央研究院地球科學所共同合作開發，於2020年年底正式推出對外開放測試。這個網路服務系統不僅採用了國際間標準的資料交換格式，並且透過全自動化的資料處理模組快速將高品質的地震與地球物理觀測資料呈現在使用者面前，目前開放的地震資料包括中央氣象署地震觀測網(CWASN, 2012-01-01~)和臺灣強地動觀測網(TSMIP, 2020-12-01~)的波形及其對應的儀器響應，以及1973-01-01之後氣象署所偵測到的地震目錄；而由臺灣地球物理觀測網(TGNS, 2007-01-01~)所記錄的資料則有全球導航衛星系統(GNSS)、地下水水位(GW)和地球磁場(MAGNET)等資料，由這些資料所產出的成果或產品也會陸續在教育與推廣裡推出。相信這個資料服務平台能帶來全新的體驗，歡迎各位使用與指教，來信請寄 gdms@cwa.gov.tw。



測站資訊

氣象署地震中心開放資料的所有觀測網可以透過縮放地圖及列表頂端的搜尋範圍查詢測站，點選表內或圖上的個別測站，可進一步取得詳細的測站資訊

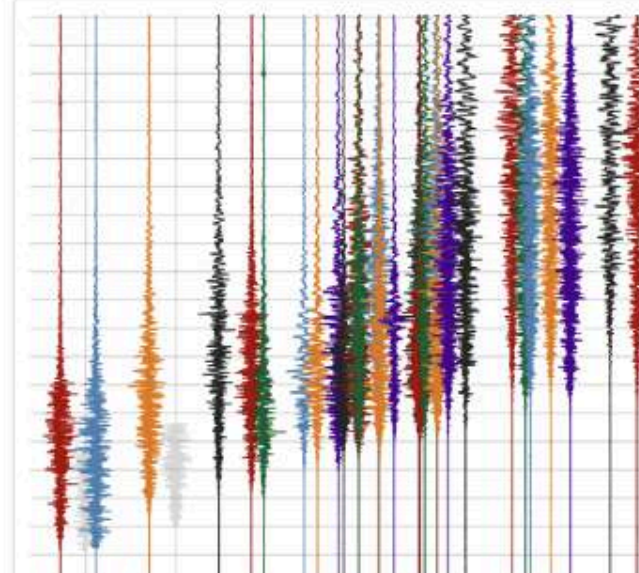
[更多資訊](#)



地震分布

這個動畫展示1990年後規模大於4.5的地震的時空分布，顏色表現震源深度，圓形的直徑與規模正相關，使用者可自行調整播放速度和播放進度

[更多資訊](#)

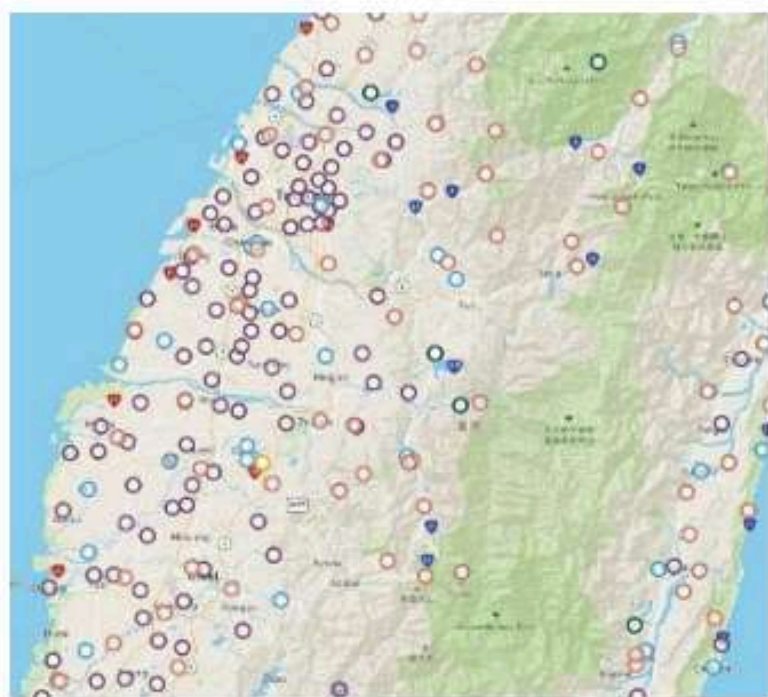


波形資料

同時索取單一地震的多個測站波形資料，可以從資料下載的多站波形資料介面取得。此圖是2024 M7.2花蓮地震，由部分TSMIP測站依震央距排列的垂直向波形紀錄。

[更多資訊](#)

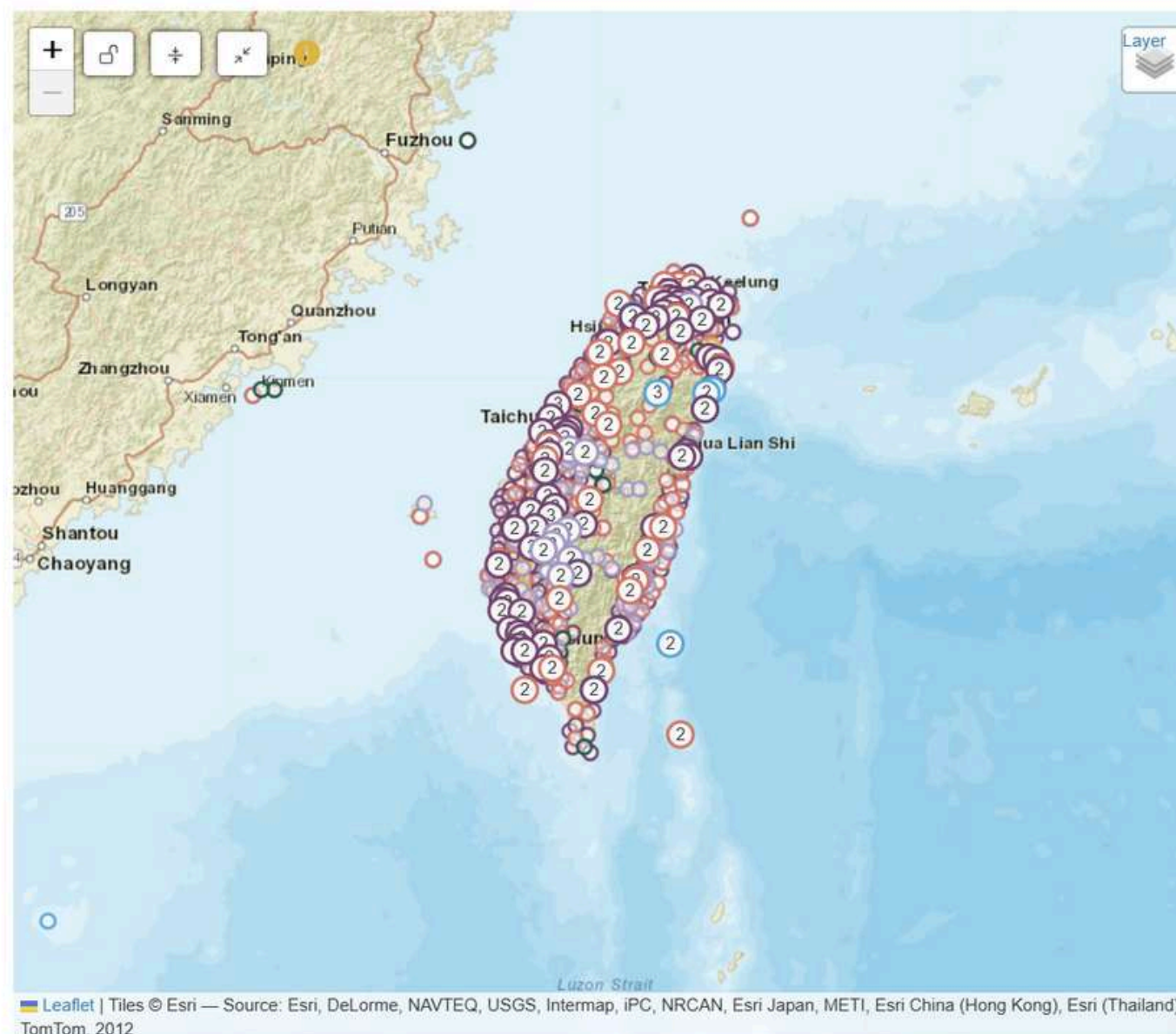
GDMS網頁介紹



測站資訊

氣象署地震中心開放資料的所有觀測網可以透過縮放地圖及列表頂端的搜尋範圍查詢測站，點選表內或圖上的個別測站，可進一步取得詳細的測站資訊

更多資訊

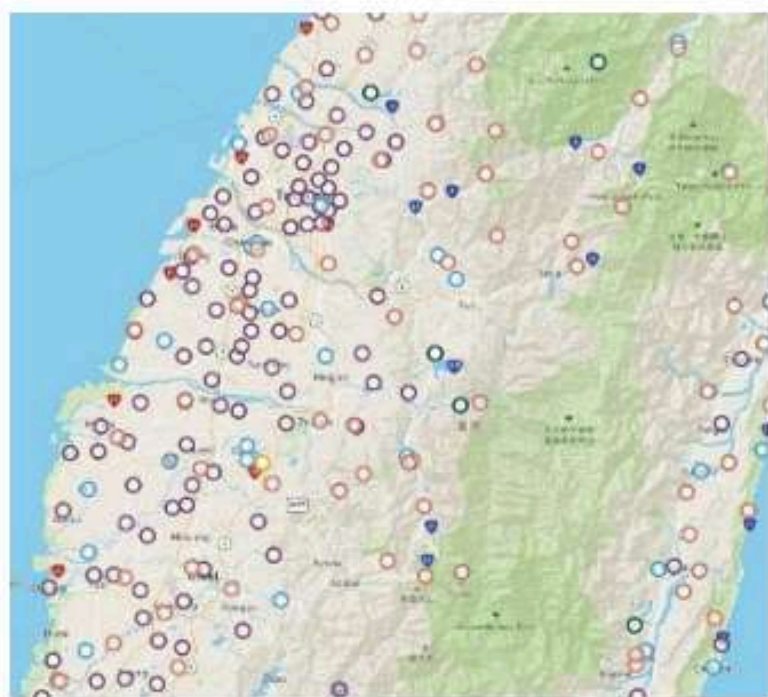


點選以下按鈕即可下載測站列表

JSON CSV Text

測網	測站代碼	緯度	經度	高程 (m)
<input type="text" value="Search"/>	<input type="text" value="Search"/>	<input type="text" value="min"/> <input type="text" value="Max"/>	<input type="text" value="min"/> <input type="text" value="Max"/>	<input type="text" value="min"/> <input type="text" value="Max"/>
CWASN	ALS	23.5084	120.8134	2417.00
CWASN	ANP	25.1828	121.5293	836.00
CWASN	BAC	24.9975	121.4418	15.00
CWASN	BGSB	24.0570	120.6106	107.00
CWASN	CHK	23.0978	121.3731	37.00
CWASN	CHKH	23.1918	121.3957	11.00

GDMS網頁介紹



測站資訊

氣象署地震中心開放資料的所有觀測網可以透過縮放地圖及列表頂端的搜尋範圍查詢測站，點選表內或圖上的個別測站，可進一步取得詳細的測站資訊

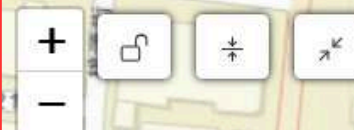
更多資訊

使用說明

🔒 / 🔓 : 鎖定 / 解除鎖定地圖

⌵ / ⌶ : 隱藏 / 顯示地圖

🔍 / 🔍 : 縮小 / 放大地圖

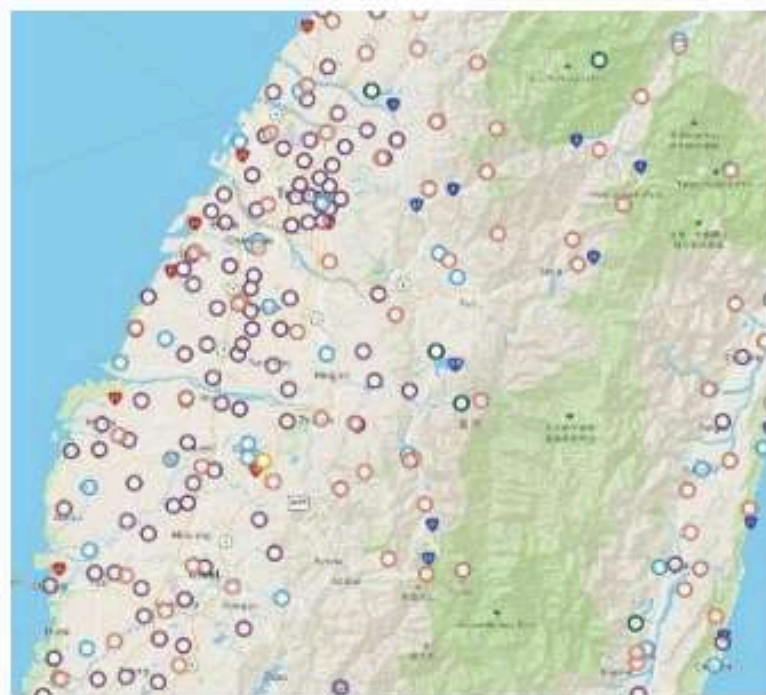


- ☒ Street Map (EN)
- ☐ Street Map
- ☐ Esri WorldImagery

CWASN TAP1

緯度: 25.0377
經度: 121.5138
起始時間: 2010-12-17
結束時間: 2018-06-27
[更多資訊](#) | [資料品質](#)

GDMIS網頁介紹



測站資訊

氣象署地震中心開放資料的所有觀測網可以透過縮放地圖及列表頂端的搜尋範圍查詢測站，點選表內或圖上的個別測站，可進一步取得詳細的測站資訊

[更多資訊](#)

CWASN TAP1

測站名稱	臺北市
位置	臺北市 中正區
經度	121.5138 °
緯度	25.0377 °
高程	15 m
起始時間	2010-12-17
結束時間	2018-06-27



可用資料

[資料品質](#)

年份

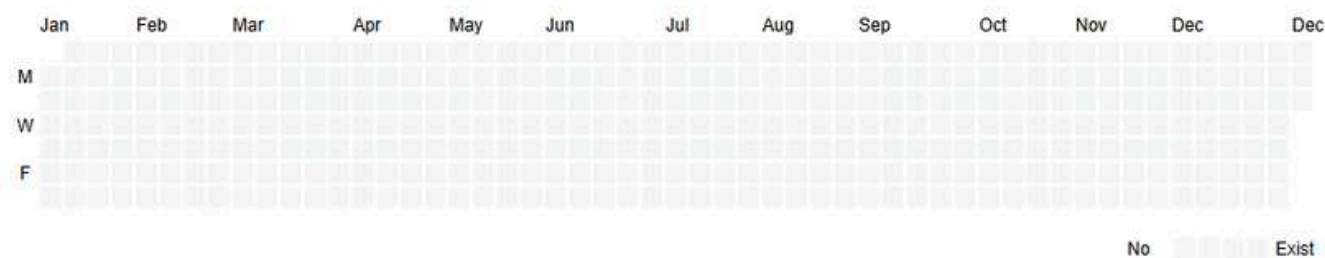
2024

位置?

00

分量?

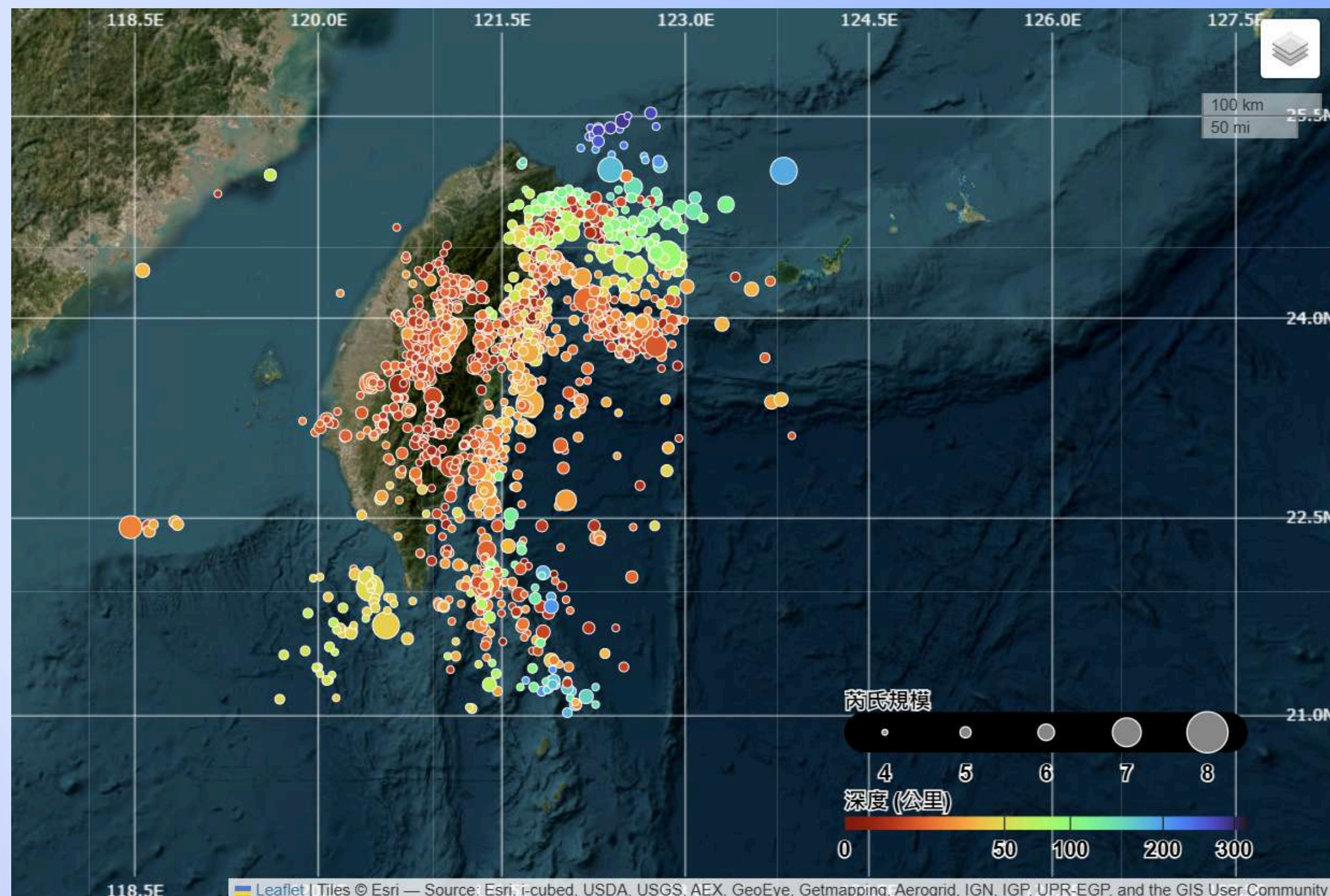
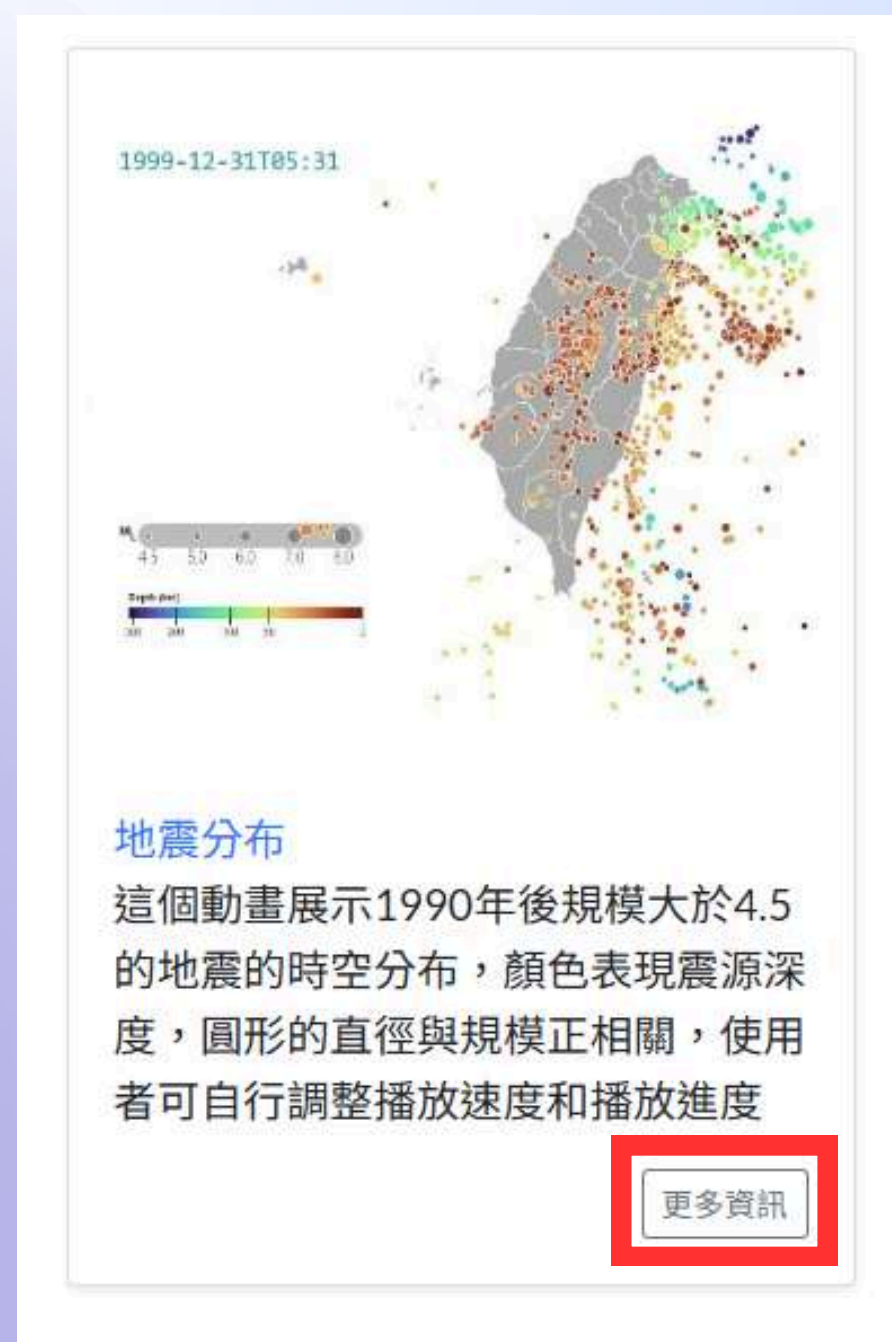
EHE



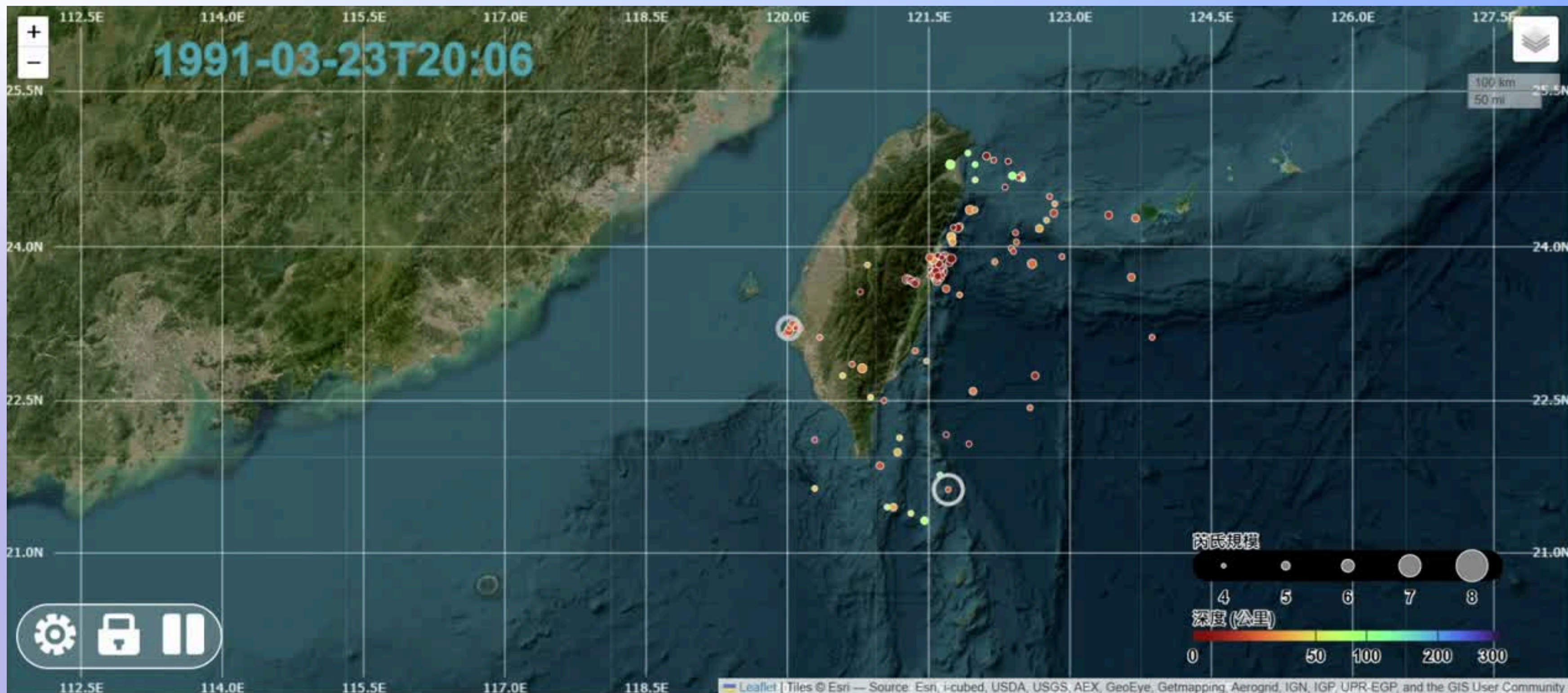
儀器資訊

分量	位置	緯度	經度	高程	深度(m)	方位角	起始時間	結束時間	地震儀	常數 (m/s/cnt. or m/s ² /cnt.)
EHE	00	25.037700	121.513800	-65.00	80.00	90.0	2010-12-17	2018-06-27	KS-54100	1.100E-10
EHN	00	25.037700	121.513800	-65.00	80.00	0.0	2010-12-17	2018-06-27	KS-54100	1.100E-10
EHZ	00	25.037700	121.513800	-65.00	80.00	0.0	2010-12-17	2018-06-27	KS-54100	1.100E-10

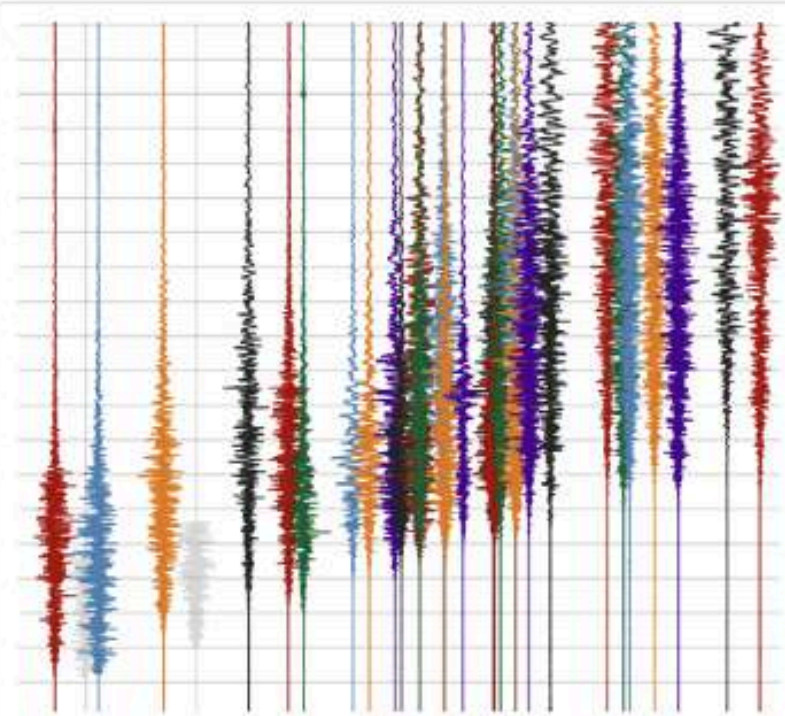
GDMS網頁介紹



● GDMS網頁介紹



GDMS網頁介紹



波形資料

同時索取單一地震的多個測站波形資料，可以從資料下載的多站波形資料介面取得。此圖是2024 M7.2花蓮地震，由部分TSMIP測站依震央距排列的垂直向波形紀錄。

[更多資訊](#)

地震資料

[多站波形資料](#)[連續波形資料](#)[多筆事件波形資料](#)[地震波形繪圖](#)[儀器響應資料](#)[地震目錄](#)

地球物理資料

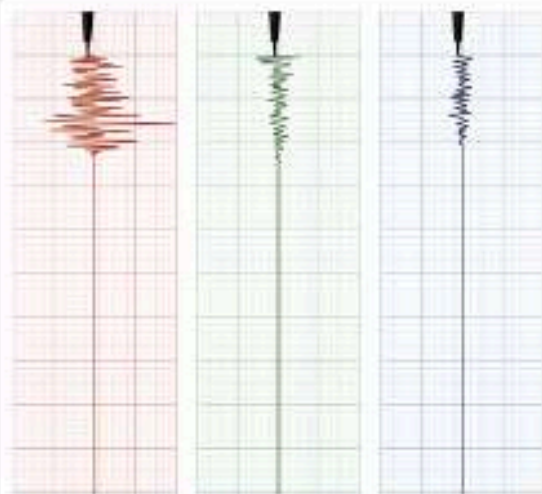
[地球物理資料](#)

GDMS網頁介紹

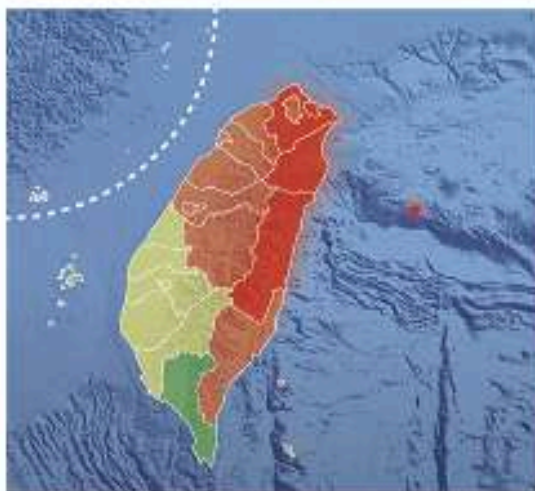
[首頁](#)[測網介紹](#)[測站查詢](#)[資料下載](#)[教育推廣](#)[使用說明](#)[相關連結](#)

地震觀測網

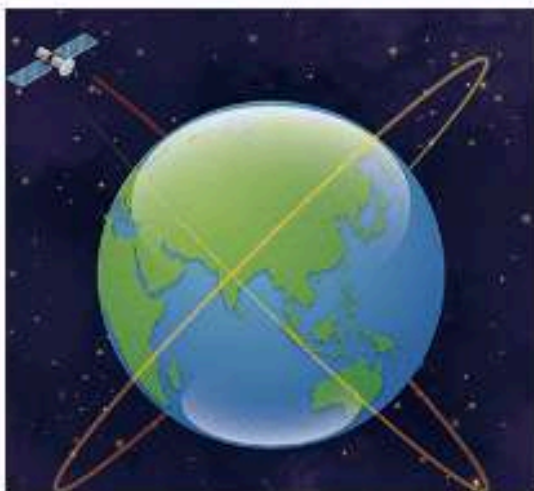
STOP



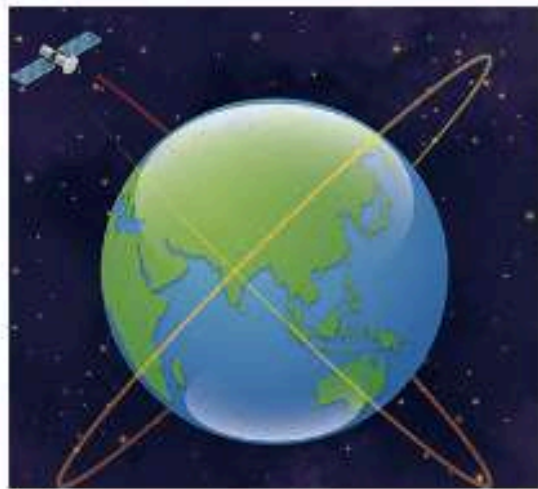
CWASN
中央氣象署地震觀測網
Central Weather Administration
Seismographic Network
[更多資訊](#)



TSMIP
臺灣強地動觀測網
Taiwan Strong Motion
Instrumentation Program network
[更多資訊](#)



TGNS
臺灣地球物理觀測網
Taiwan Geophysical Network for
Seismology
[更多資訊](#)



OTHERS
協力提供下載觀測網
Cooperation Network
[更多資訊](#)

GDMMS網頁介紹



首頁



測網介紹



測站查詢



資料下載



教育推廣



使用說明



相關連結

氣象署數位科普網

地震百問

臺灣的地震特性

臺灣的地震頻率如何？

臺灣的地震分布

臺灣的地震活動(動畫)

地震看的見



中央氣象署數位科普網

網站總覽

文章區

兒童區

熱門區

影音區

漫畫區

互動區



氣象

地震

海象

天文

防災教育

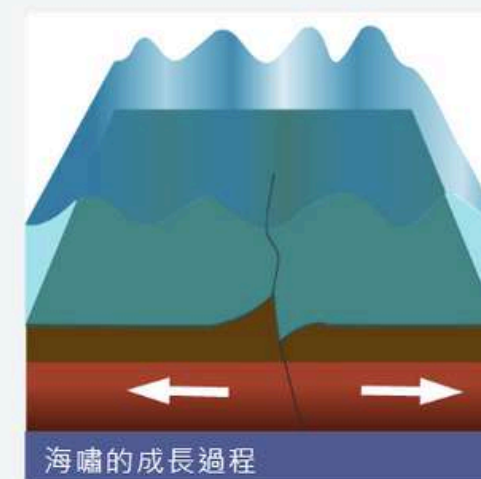
首頁 > 地震

熱門文章

更新時間

分享次數

依年份



海嘯的成長過程



日本東北311大地震 模最大地震！



重新搞懂板塊這件事



唐山大地震



莫氏不連續面的發現者



地牛翻身新指標—震度分級知多少

GDMIS網頁介紹



首頁



測網介紹



測站查詢



資料下載



教育推廣



使用說明



相關連結

氣象署數位科普網

地震百問

臺灣的地震特性

臺灣的地震頻率如何？

臺灣的地震分布

臺灣的地震活動(動畫)

地震看的見

地震資訊

海嘯資訊

中心簡介

教育宣導

觀測網介紹

技術報告

地震話題

告警資訊

地震百問

⋮ [首頁](#) / [教育宣導](#) / [地震百問](#) / 一、地球概論

一、地球概論

二、地震

三、地震災害與測報

四、地震災害預防

五、地震資料服務

上一筆

下一筆

一、地球概論

- 1. 地球是什麼？
- 2. 地球的內部構造為何？
- 3. 地球各部分構造的質量與體積如何？
- 4. 地球的常數如何？
- 5. 地球的重力如何？
- 6. 地球之磁力如何？
- 7. 地球之溫度變化？

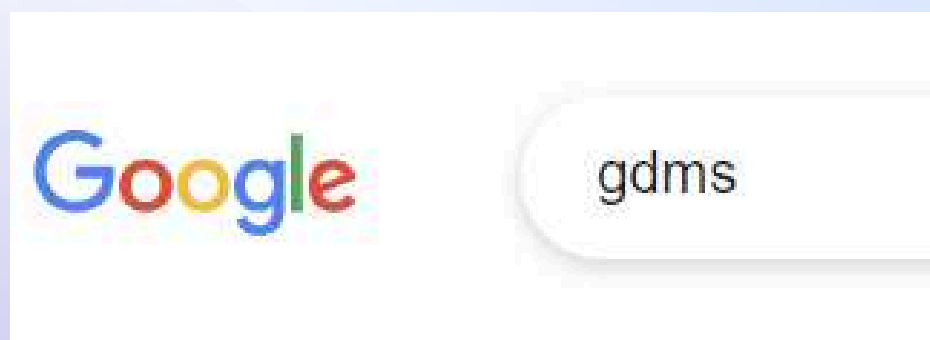
上一筆

下一筆

GDMIS網頁介紹

首頁	測網介紹	測站查詢	資料下載	教育推廣	使用說明	相關連結
相關連結						
中央氣象署			https://www.cwa.gov.tw/			
中央氣象署地震測報中心			https://scweb.cwa.gov.tw/			
中央研究院地球科學研究所			https://www.earth.sinica.edu.tw/			
台灣地震科學中心			https://tec.earth.sinica.edu.tw/			
國家地震工程研究中心			https://www.ncree.narl.org.tw/			
台灣寬頻地震觀測網			https://bats.earth.sinica.edu.tw/			
FDSN			https://www.fdsn.org/			
GEOFON, Germany			https://geofon.gfz-potsdam.de/			
IRIS, USA			https://www.iris.edu/			
NIED, Japan			https://www.bosai.go.jp/			
ORFEUS, EU			https://www.orfeus-eu.org/			

GDMS註冊



Google搜尋GDMS：

A screenshot of a Google search results page for the query 'gdms'. The search bar at the top contains 'gdms' and is highlighted with a red box. A red arrow points from this box to the search bar in the inset image on the left. Below the search bar, the '全部' (All) tab is selected. The first search result is for '地球物理資料管理系統' (Geophysical Data Management System) with the URL 'https://gdms.cwa.gov.tw'. This result is highlighted with a red box. Below the title, the text reads: 'GDMS 臺灣地震與地球物理資料管理系統. 臺灣地震與地球物理資料管理系統(GDMS)是由中央氣象署與中央研究院地球科學所共同合作開發，於2020年年底正式推出對外開放測試。... 相信 ...'. Below this, there are four links: '地震目錄' (Earthquake Catalog), '使用說明' (User Guide), 'TSMIP', and '登入' (Login). Each link has a brief description below it.

Google

gdms

全部 圖片 購物 新聞 影片 地圖 網頁 更多 工具

分析 原理 Software CCTV Camera Analysis 輝光 放電 質譜 Meaning

地球物理資料管理系統
https://gdms.cwa.gov.tw

GDMS
臺灣地震與地球物理資料管理系統. 臺灣地震與地球物理資料管理系統(GDMS)是由中央氣象署與中央研究院地球科學所共同合作開發，於2020年年底正式推出對外開放測試。... 相信 ...

地震目錄
gdms.cwa.gov.tw. Taiwan Seismological and Geophysical ...

使用說明
這四行分別表示：(1)讀入SAC檔、(2)移除平均值及線性趨勢、(3)兩端 ...

TSMIP
臺灣強地動觀測網. 1991年氣象署開始逐步於臺灣本島建置大量的自 ...

登入
地球物理資料. 地球物理資料下載，GNSS全部測站資料下載，選取 ...

點選網頁右上角的登入之後進行會員註冊並點選同意：



請登入

帳號

Email

密碼

密碼

☐ 手動驗證(本站採自動驗證，若自動驗證失敗可開啟)

登入

[會員註冊](#)

[忘記密碼?](#)

您要如何退出、刪除或修改您已經提供給我們的資訊?

您可連絡我們(gdms@cwa.gov.tw)完全刪除您的個人資料，或者登入本網站後進入個人資料網頁進行修

我們會向第三方揭露嗎?

我們不會販售、交易或以其他方式轉移您的個人可辨識資訊給其他單位，除非我們有進一步通知並獲得遵守法律、強化本網站隱私政策或保護本網站及其他使用者的權益及安全的考量下，我們有可能釋出個部門(例如法院或政府單位)提出合理要求，否則我們不會在違反個人意願下轉移您的個人資料。

您個人資料的保留

我們只會在符合這個隱私權政策下的時間內，保留您的個人資料，並且合法、合理地使用這些資訊來解策。

自註冊申請時至您選擇退出本系統或刪除，或至本系統服務終止為止，如本系統服務終止，亦將妥善銷

聯絡我們

如果您有任何關於這個隱私權政策的疑問，您可以透過email與我們聯繫: gdms@cwa.gov.tw

同意

不同意

輸入各項資料驗證並送出：

* 必填

電子郵件信箱 (即登入帳號) *

可用學校信箱

注意: 認證信有可能在垃圾信件匣內或被電子郵件伺服器阻擋，若收不到信建議使用其他信箱申請(例如: 學校或公務單位信箱)

單位 *

臺北市立大學

姓名 *

國家或地區 *

台灣

Taiwan

過往研究 *

無 填無即可

資料類型 (不影響之後搜尋及資料下載服務) *

☒ CWASN

☒ TSMIP

☒ GNSS

☒ GW

☒ MAGNET

☒ GNSS_IES

☒ GNSS_ETEC

密碼 * (需包含英文大寫、小寫、數字及特殊符號(~!@#\$%^&*()-_+=)，且長度至少需12碼)

.....

確認密碼 *

.....

☒ 手動驗證(本站採自動驗證，若自動驗證失敗可開啟)

驗證碼

g487k5

6487k5

Reload

送出

自動驗證如果失敗可以利用手動驗證，輸入驗證碼

註冊成功會跳出此對話框：



至填寫的信箱中點選申請連結之後即可登入使用：

申請成功，您已可登入

● GDMIS地震相關資料下載及應用



● 地震分布圖資料下載

地震資料

多站波形資料

連續波形資料

多筆事件波形資料

地震波形繪圖

儀器響應資料

地震目錄

地球物理資料

地球物理資料

透過地震目錄填寫各項資料(日期、經緯度等)查詢地震資料：

地震目錄

可用資料：1973-01-01 至 2024-11-30

(因地震資料龐大，優先處理規模2以上地震，目前全部處理完成的地震目錄統計到2019年10月。)

可依條件搜尋地震目錄

* 必填

起始日期 *

起始時間 (UTC) *

結束日期 *

結束時間 (UTC) *

2024-04-03

00:00:00

2024-04-04

00:00:00

查詢方式

方框 box

地圖工具

最小經度

最大經度

最小緯度

最大緯度

118

124

20

27

min M_L *

Max M_L *

最小深度

最大深度

4

10

0

1000

送出

設定經緯度範圍

設定想找的時間

選擇需要的檔案資料下載(.json/.csv/.txt)：

點選以下按鈕即可下載地震目錄 [help](#)

資料總數：244

Show 10 rows

JSON

CSV

Text

此處可選擇需要下載的檔案類型

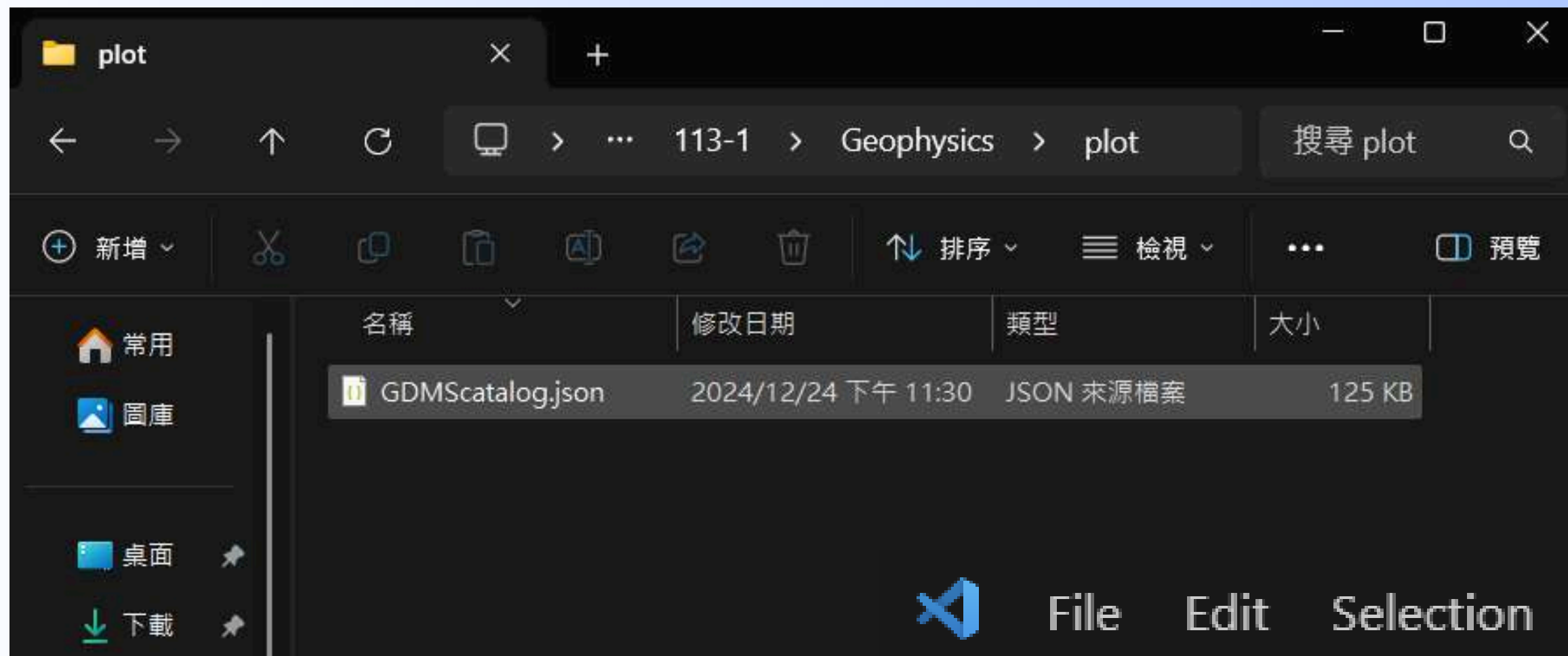
- .json：適合儲存結構化數據，常見於網路數據傳輸。
- .csv：用於簡單的表格數據，適合數據導入與導出。
- .txt：適合存儲非結構化文本內容，簡單但缺乏數據結構。

date	time	lat	lon	depth	ML	nstn	dmin	gap	trms	ERH	ERZ	fixed	nph	quality
2024-04-03	00:00:19.33	23.8075	121.5847	30.72	5.71	77	3.10	176	0.31	0.30	0.20	F	135	C
2024-04-03	00:01:51.02	24.0083	121.6590	37.02	5.41	99	5.90	107	0.29	0.20	0.20	F	254	B
2024-04-03	00:02:23.81	24.1727	121.6590	33.30	5.08	80	17.70	94	0.25	0.20	0.20	F	144	B
2024-04-03	00:02:54.06	24.0487	121.5540	26.06	4.52	19	10.20	117	0.40	1.20	0.80	F	26	C
2024-04-03	00:03:47.62	23.9670	121.6628	32.96	5.06	89	5.10	181	0.37	0.20	0.20	F	169	D
2024-04-03	00:04:06.83	24.0868	121.6447	19.87	4.33	17	12.80	176	0.23	0.40	0.70	F	34	C
2024-04-03	00:04:14.77	24.1188	121.6818	26.17	5.18	88	17.78	188	0.38	0.18	0.18	F	281	C
2024-04-03	00:07:22.50	23.8295	121.5682	19.81	4.45	61	4.70	162	0.29	0.10	0.10	F	109	C
2024-04-03	00:07:35.39	24.0082	121.5403	18.25	4.22	20	6.70	92	0.30	1.10	0.90	F	26	B

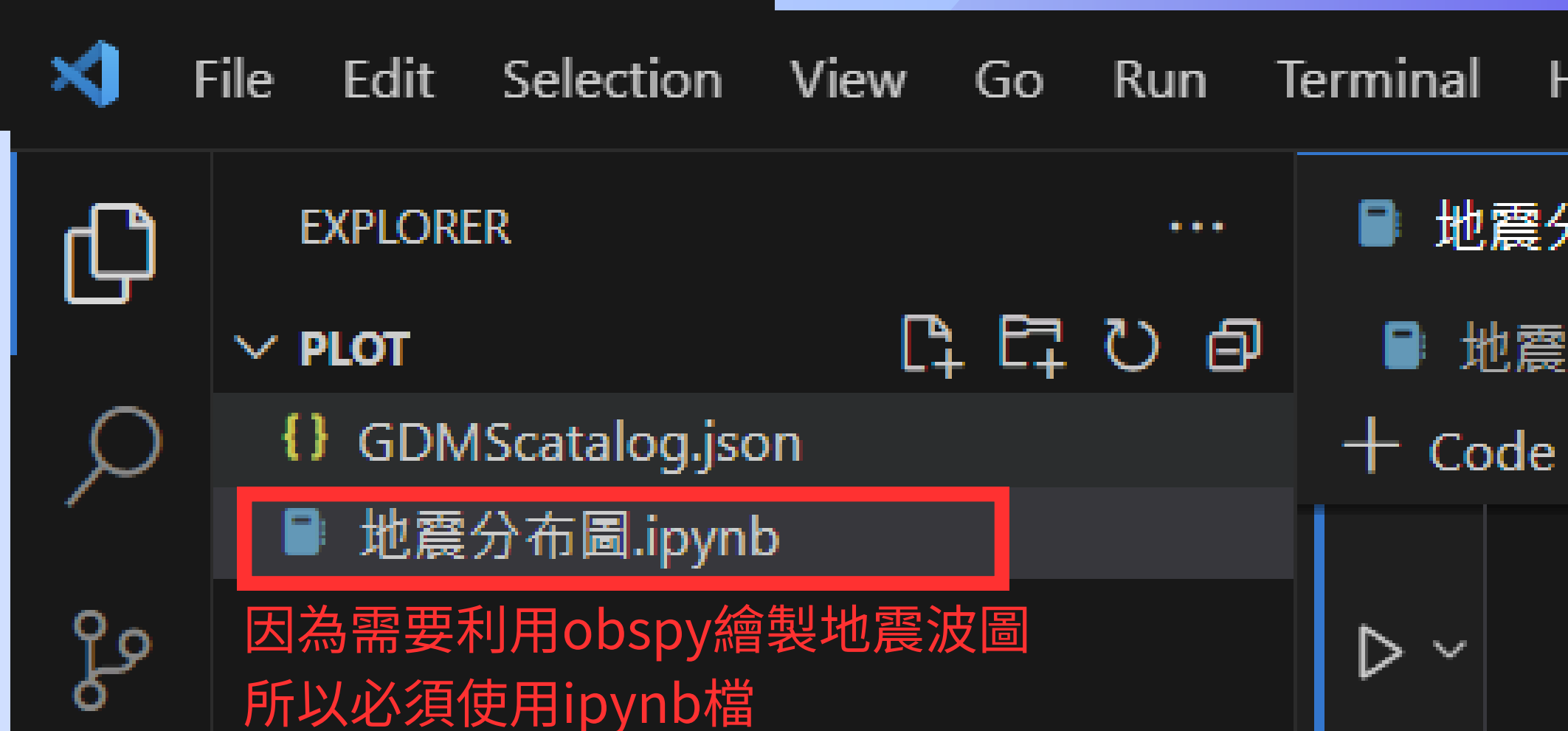
• ML: 規模 • nstn: 使用測站數量 • Dmin: 最近站震央距(F5.1) • Gap: 相鄰測站之最大間隙角度 • RMS: 時間殘值之方均根誤差值
• ERH: 震央之(水平)標準差(公里) • ERZ: 震源深度之(垂直)標準差(公里) • Fixed: 深度控制(F: 未限制,X:限制深度) • nph: 到時相位數量

透過篩選條件所選出的資料內容

繪製地震分布圖



建立一個.ipynb檔：




```
import obspy
import json
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy.feature as cfeature

# 讀取 JSON 檔案
file_path = "GDMScatalog.json" # 替換為您的檔案路徑
with open(file_path, 'r') as f:
    data = json.load(f)

# 提取地震數據
header = data["header"]
body = data["body"]

# 解析經緯度
lat_index = header.index("lat")
lon_index = header.index("lon")
magnitude_index = header.index("ML")

lats = [float(row[lat_index]) for row in body]
lons = [float(row[lon_index]) for row in body]
magnitudes = [float(row[magnitude_index]) for row in body]

# 繪製地震分布圖
plt.figure(figsize=(12, 8))
ax = plt.axes(projection=ccrs.PlateCarree())

# 設定地圖範圍 (可根據您的需求調整)
ax.set_extent([119, 123, 21, 26], crs=ccrs.PlateCarree())

# 添加地圖特徵
ax.add_feature(cfeature.LAND, edgecolor="black")
ax.add_feature(cfeature.OCEAN)
ax.add_feature(cfeature.COASTLINE)
ax.add_feature(cfeature.BORDERS, linestyle=":")

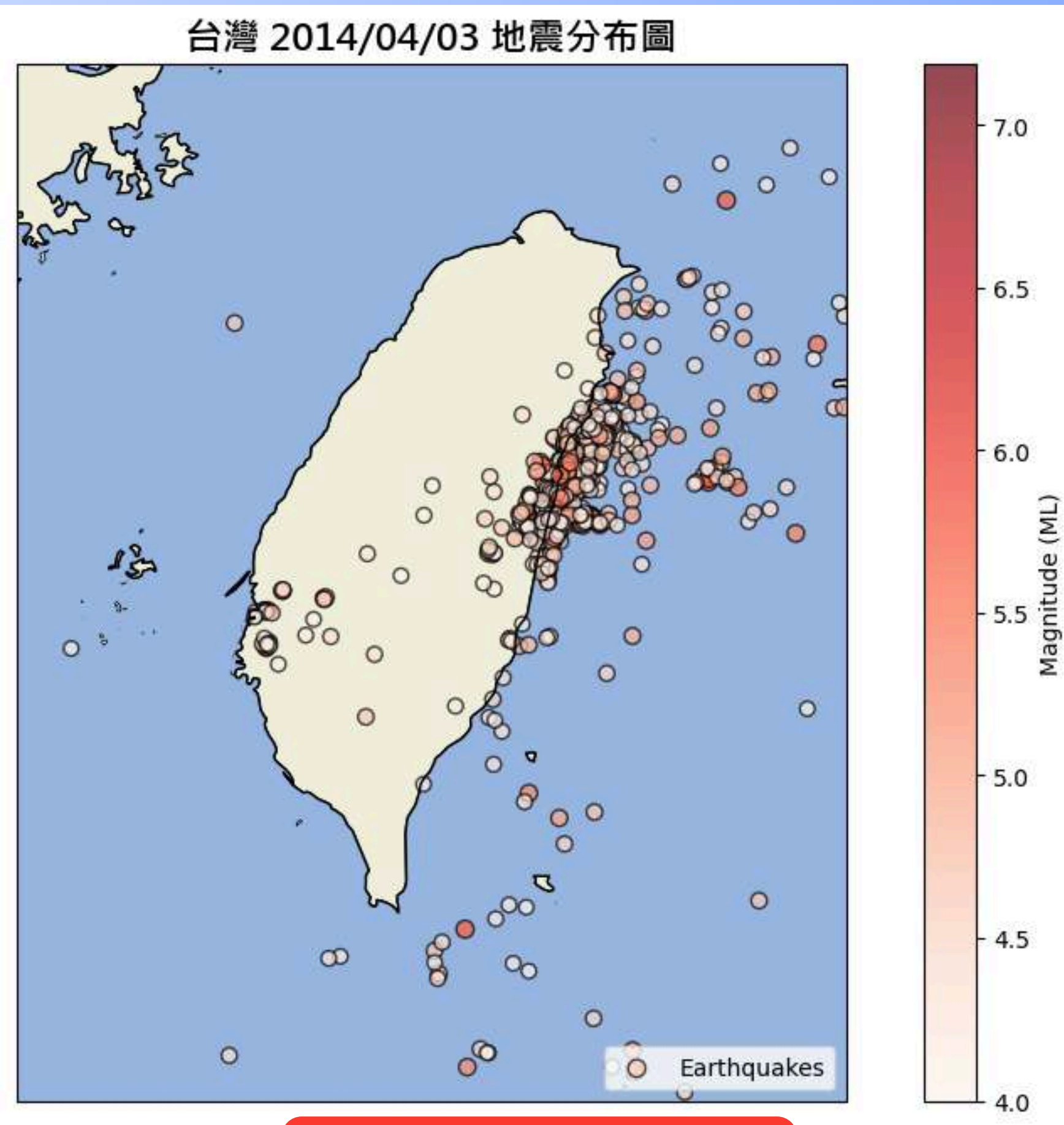
# 繪製地震事件，大小按震級設定
scatter = ax.scatter(lons, lats, c=magnitudes, cmap="Reds", s=[mag * 10 for mag in magnitudes],
                    transform=ccrs.PlateCarree(), label="Earthquakes", edgecolor="black", alpha=0.7)

# 添加顏色條
cbar = plt.colorbar(scatter, ax=ax, orientation="vertical", pad=0.05)
cbar.set_label("Magnitude (ML)")

# 添加繁體標題
plt.title("台灣 2014/04/03 地震分布圖", fontsize=16, fontweight="bold", family="Microsoft JhengHei")

# 顯示圖例
plt.legend(loc="lower right")
plt.show()
```

使用pygmt環境 以及
2024/04/03 00:00-23:59
的地震波數據



顏色越深，表示震度越大

● 地震波圖資料下載

透過多站波形資料填寫各項資料詢波形資料：

地震資料

[多站波形資料](#)

[連續波形資料](#)

[多筆事件波形資料](#)

[地震波形繪圖](#)

儀器響應資料

[儀器響應資料](#)

地震目錄

[地震目錄](#)

地球物理資料

[地球物理資料](#)

多站波形資料

多站波形資料下載，索取時間不可超過1天(86400秒)。

*** 必填**

輸出格式 *

MiniSEED

測網 *

CWASN

測站 *

☐ 全部測站

位置? *

ALS,

分量? *

10

起始時間 (UTC) *

2024-04-03T00:00:00

結束時間 (UTC) *

2024-04-03T23:59:59

輸出檔名

0403eq

送出

通常使用MiniSEED(.mseed)或SAC binary(.sac)

等待檔案生成後即可下載：

此處可刷新確認資料是否生成完成

Reload

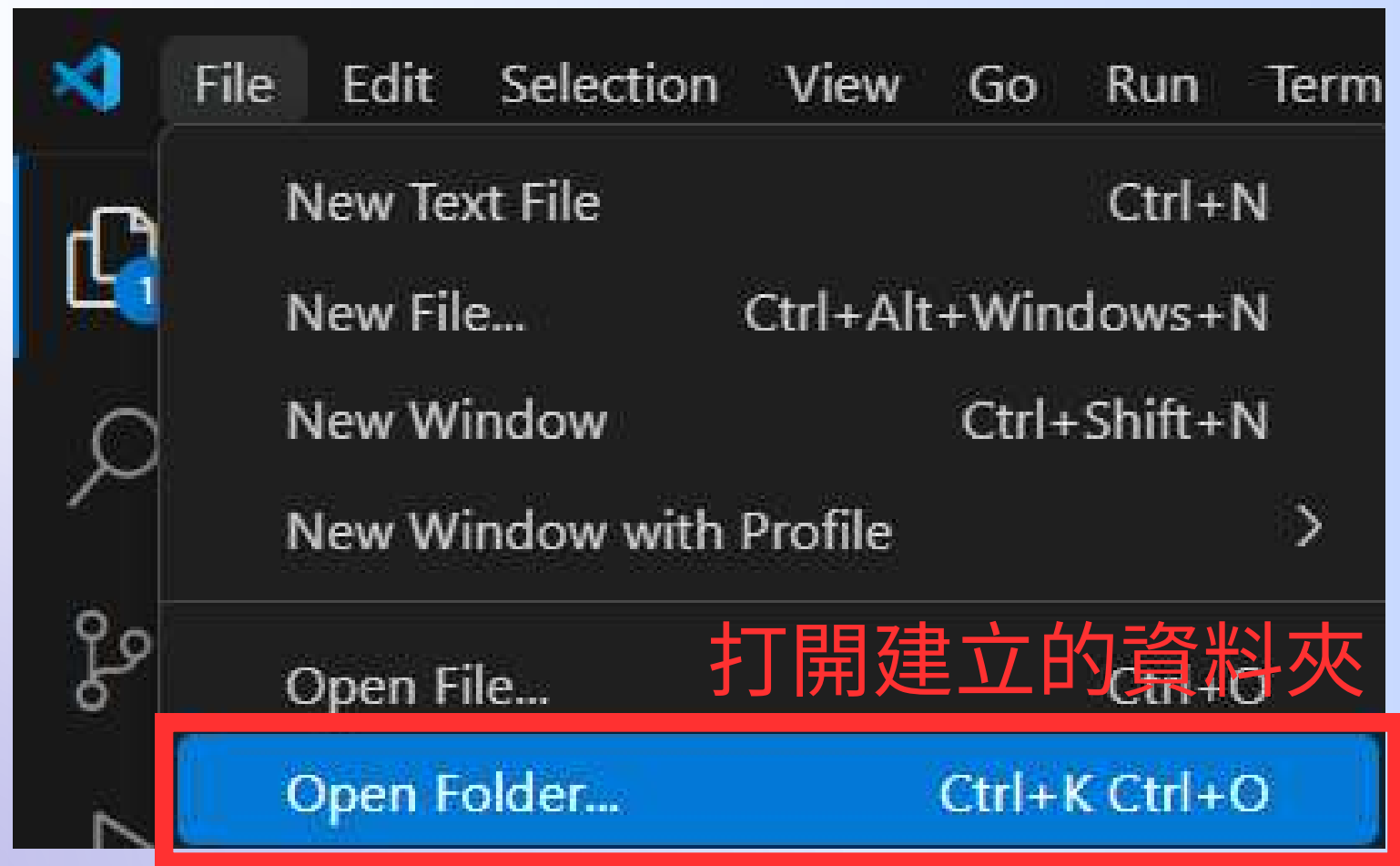
更新時間：2024/10/29 下午9:12:19

資料類型	索取時間 (UTC+8)	索取內容	下載檔案？
地震 (資料)	2024-10-29 21:10:41	內容	0403eq.mseed

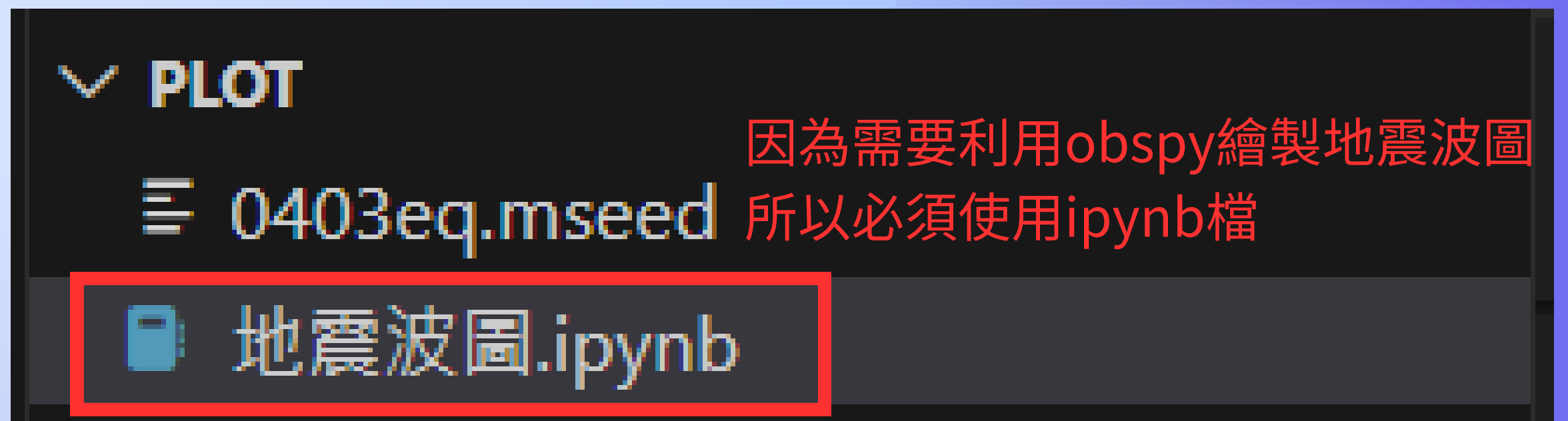
點此可看所篩選的資料內容

點此下載檔案

繪製地震波圖



建立一個ipynb檔：




```
import obspy
```

[7] ✓ 0.0s



```
from obspy import read
```

```
st = read("0403eq.mseed")  
st.plot()
```

[8] ✓ 0.2s

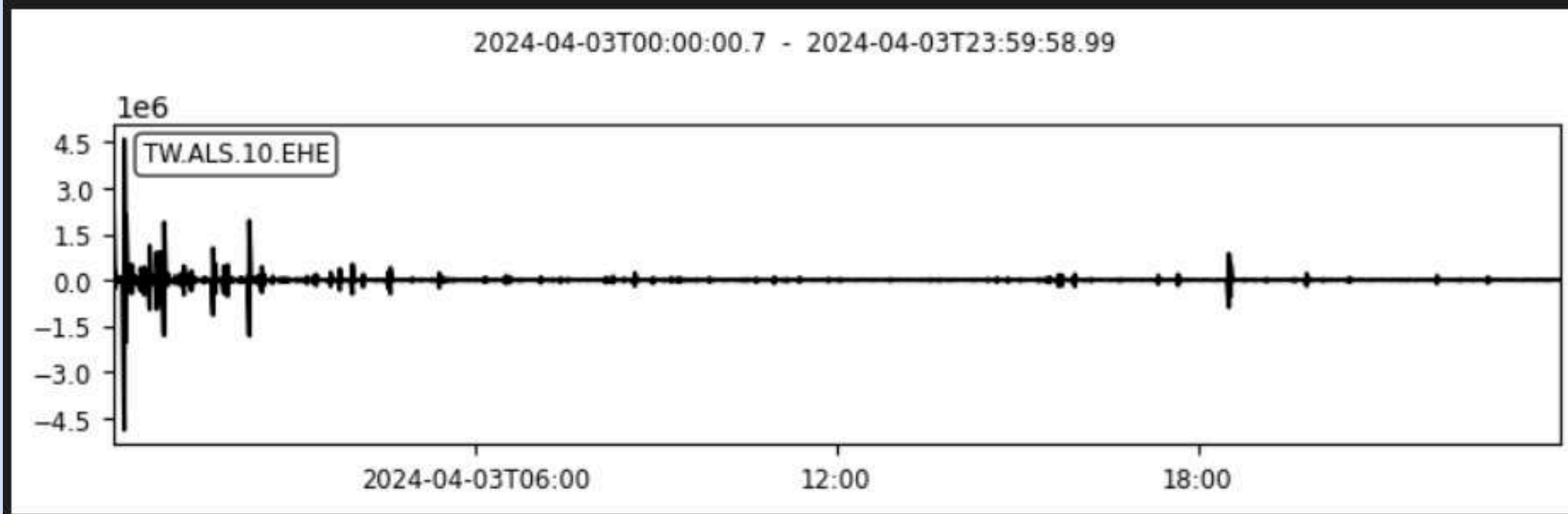
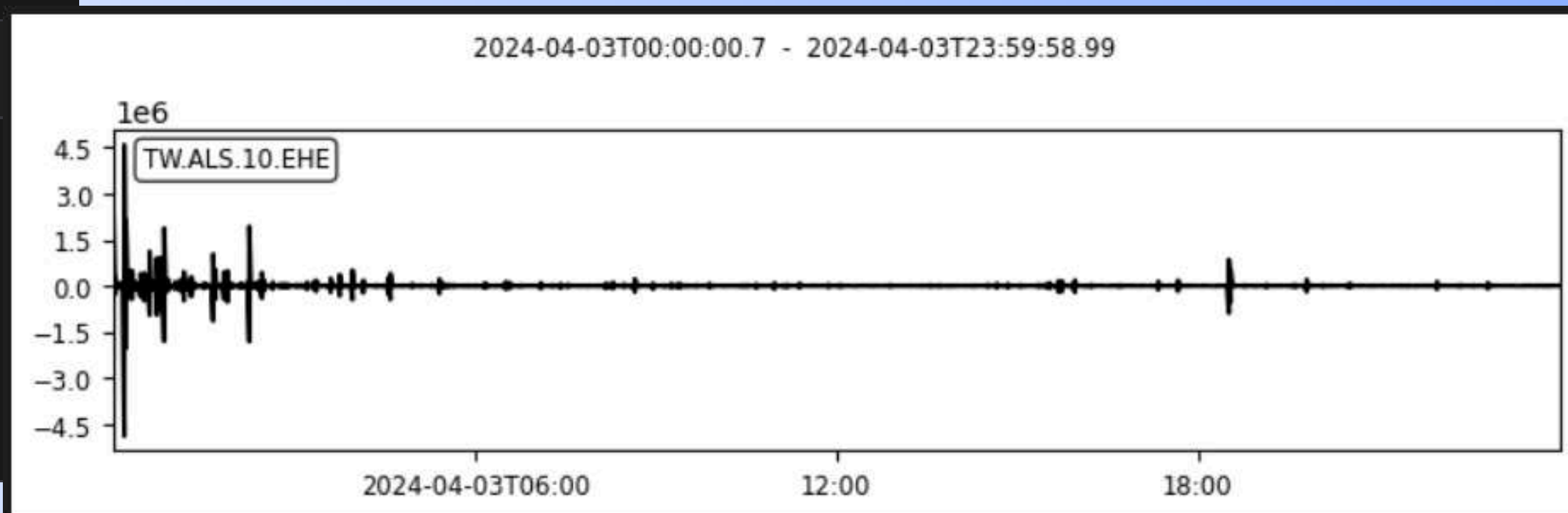
輸入程式碼並輸出：

此處使用

pygmt環境 以及

2024/04/03 00:00-23:59

的地震波數據



自製Raspberry Pico地震監測儀

C:\Windows\System32>ipconfig

確認IP位置

Windows IP 設定

乙太網路卡 乙太網路 2:

連線特定 DNS 尾碼 :
 連結-本機 IPv6 位址 : fe80::281b:4db1:fc09:6093%4
 IPv4 位址 : 192.168.56.1
 子網路遮罩 : 255.255.255.0
 預設閘道 :

無線區域網路介面卡 區域連線* 1:

媒體狀態 : 媒體已中斷連線
 連線特定 DNS 尾碼 :

無線區域網路介面卡 區域連線* 2:

媒體狀態 : 媒體已中斷連線
 連線特定 DNS 尾碼 :

無線區域網路介面卡 Wi-Fi:

連線特定 DNS 尾碼 :
 IPv6 位址 : 2001:b400:e25e:eb32:27a3:7fe:8b82:d5b1
 臨時 IPv6 位址 : 2001:b400:e25e:eb32:8870:6cb:7d5a:9b5f
 連結-本機 IPv6 位址 : fe80::de8b:40aa:bd7:25e4%7
 IPv4 位址 : 192.168.115.117
 子網路遮罩 : 255.255.255.0
 預設閘道 : fe80::7894:cbff:fe39:f550%7
 192.168.115.209

本機IP 也是Broker的IP

設定連接轉發

C:\Windows\System32>netsh interface portproxy add v4tov4 listenaddress=192.168.115.117 listenport=1883 connectaddress=127.0.0.1 connectport=1883

C:\Windows\System32>netsh interface portproxy show all

接聽 ipv4:

連線到 ipv4:

位址	連接埠	位址	連接埠
192.168.115.117	1883	127.0.0.1	1883
10.1.206.232	1883	127.0.0.1	1883

查看已設定的轉送規則

MQTT Broker(代理人)-轉發設定:

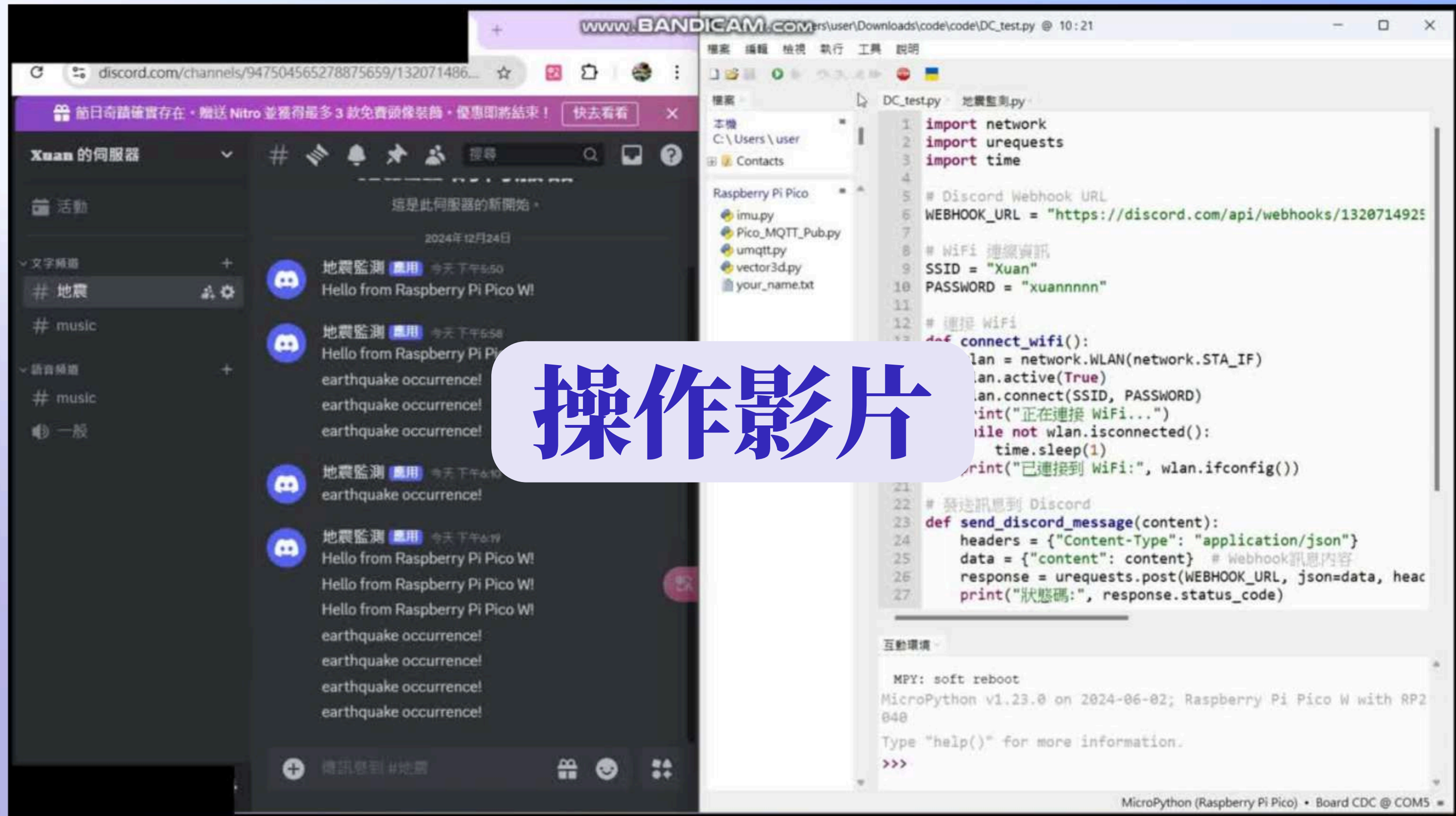
Discord webhook連結

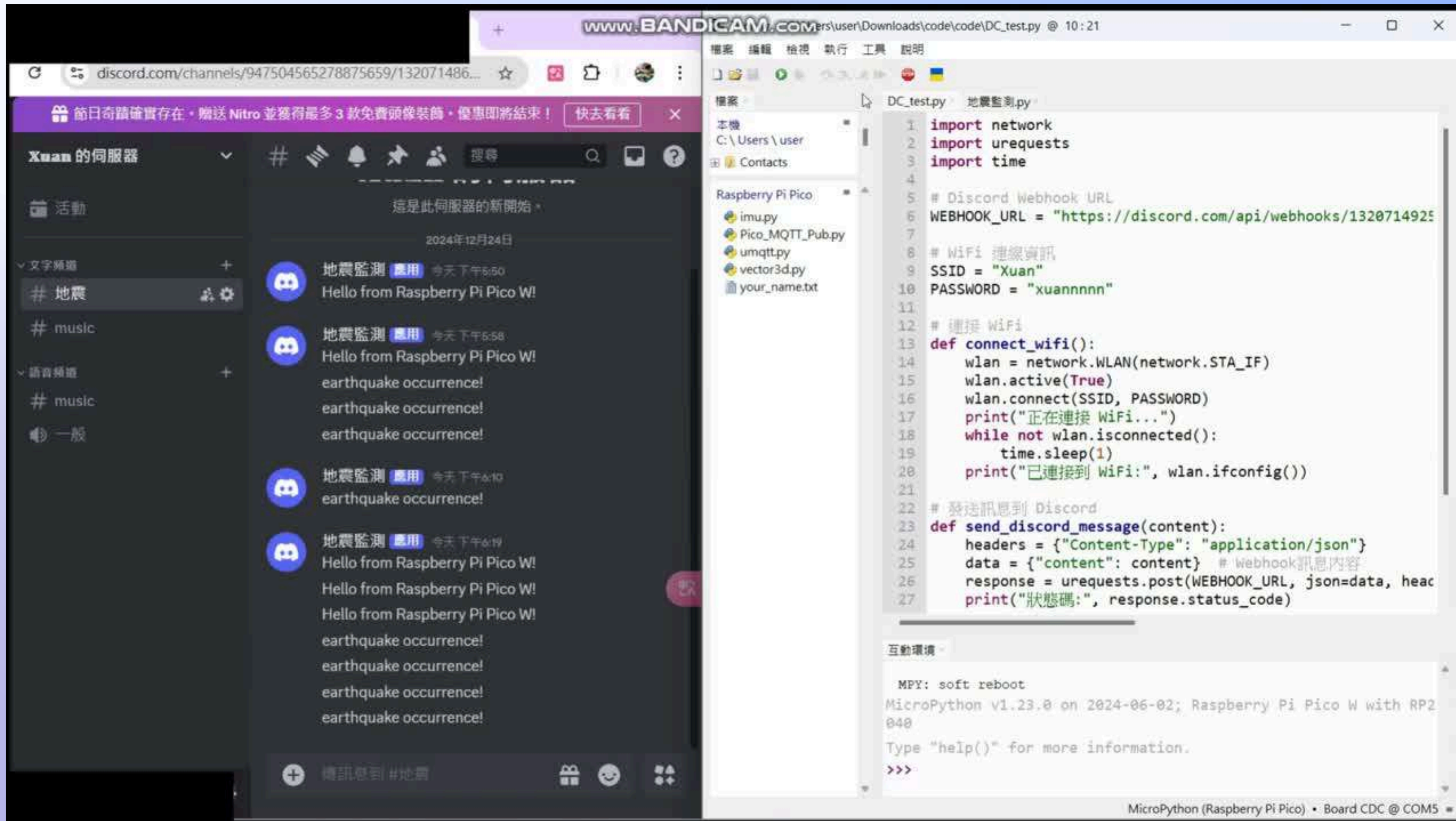
```
1 import network
2 import urequests
3 import time
4
5 # Discord Webhook URL
6 WEBHOOK_URL = "https://discord.com/api/webhooks/1320714925114396763/g7qXTkXqg5Tr81bDKCg9QJtBesf1w4nWQ_CxtKxd2SALH-4s_stgiZs1xeVDRu0UWi00"
7
8 # WiFi 連線資訊
9 SSID = "Xuan"
10 PASSWORD = "xuannnnn"
11
12 # 連接 WiFi
13 def connect_wifi():
14     wlan = network.WLAN(network.STA_IF)
15     wlan.active(True)
16     wlan.connect(SSID, PASSWORD)
17     print("正在連接 WiFi...")
18     while not wlan.isconnected():
19         time.sleep(1)
20     print("已連接到 WiFi:", wlan.ifconfig())
21
22 # 發送訊息到 Discord
23 def send_discord_message(content):
24     headers = {"Content-Type": "application/json"}
25     data = {"content": content} # Webhook訊息內容
26     response = urequests.post(WEBHOOK_URL, json=data, headers=headers)
27     print("狀態碼:", response.status_code)
28     response.close()
29
30 # 主程式
31 try:
32     connect_wifi()
33     send_discord_message("Hello from Raspberry Pi Pico W!")
34 except Exception as e:
35     print("出錯了:", str(e))
```

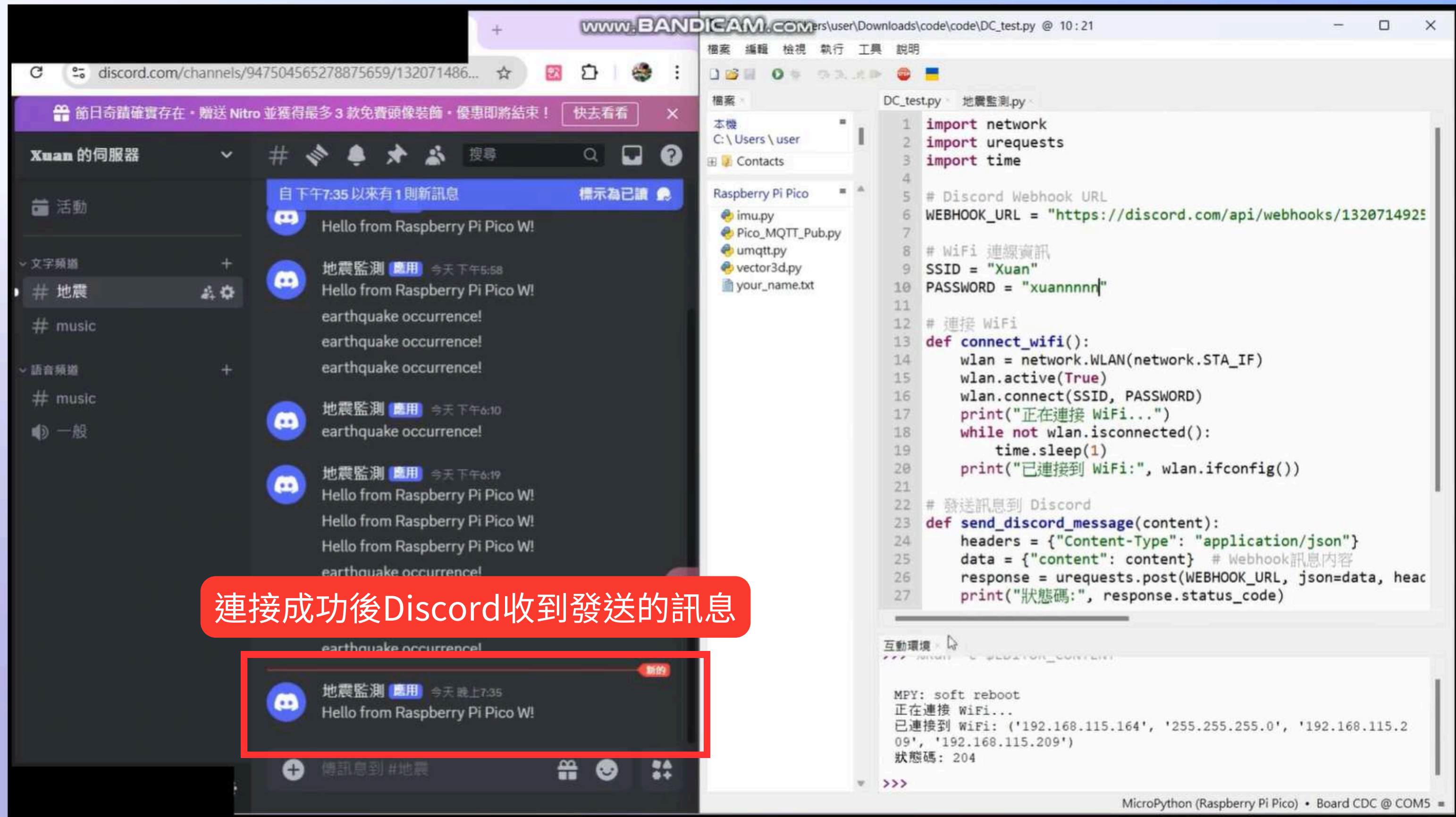
Pico連接的行動網路名稱與密碼

連接網路與Discord webhook:

網路連接成功之後發送到Discord的訊息內容

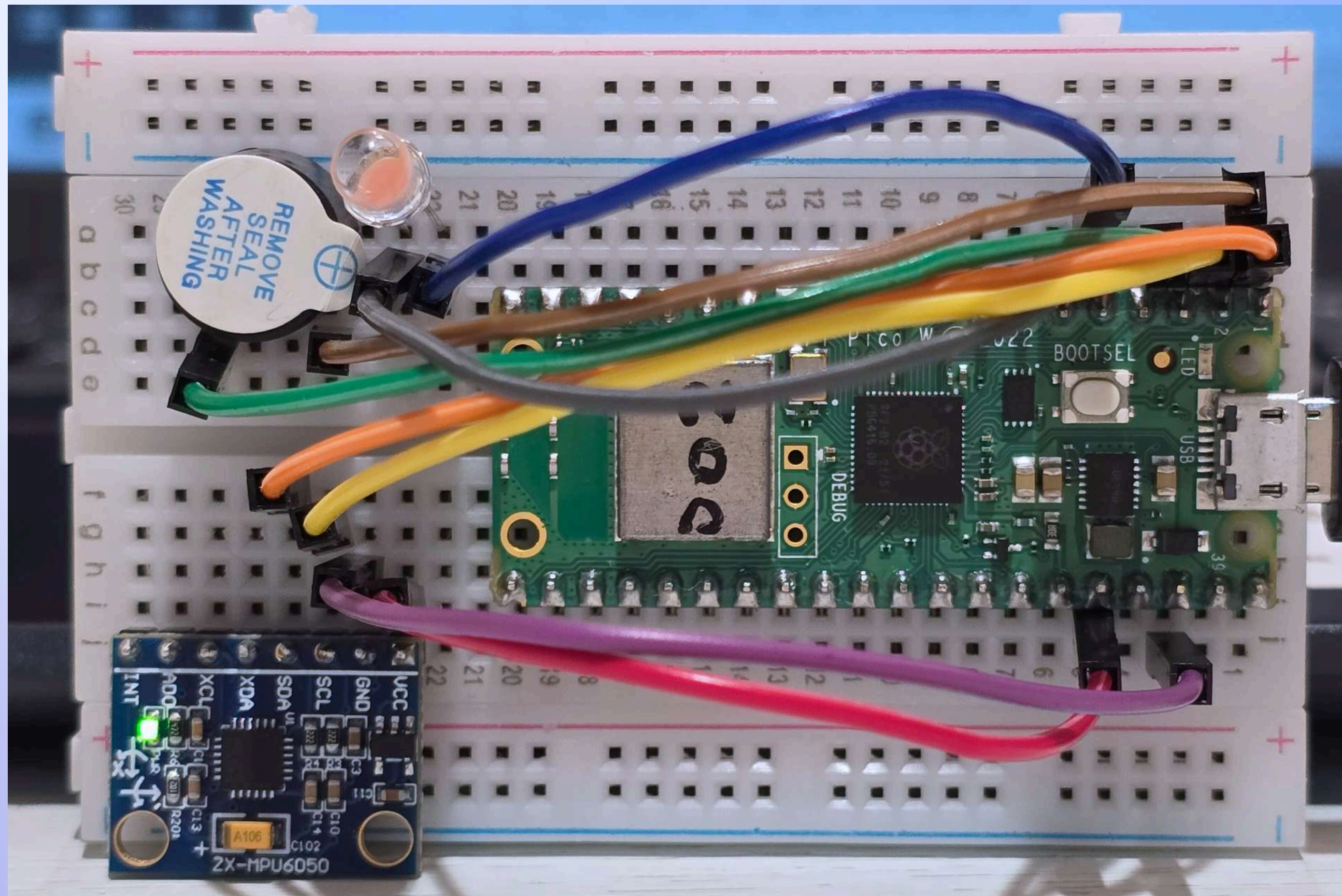




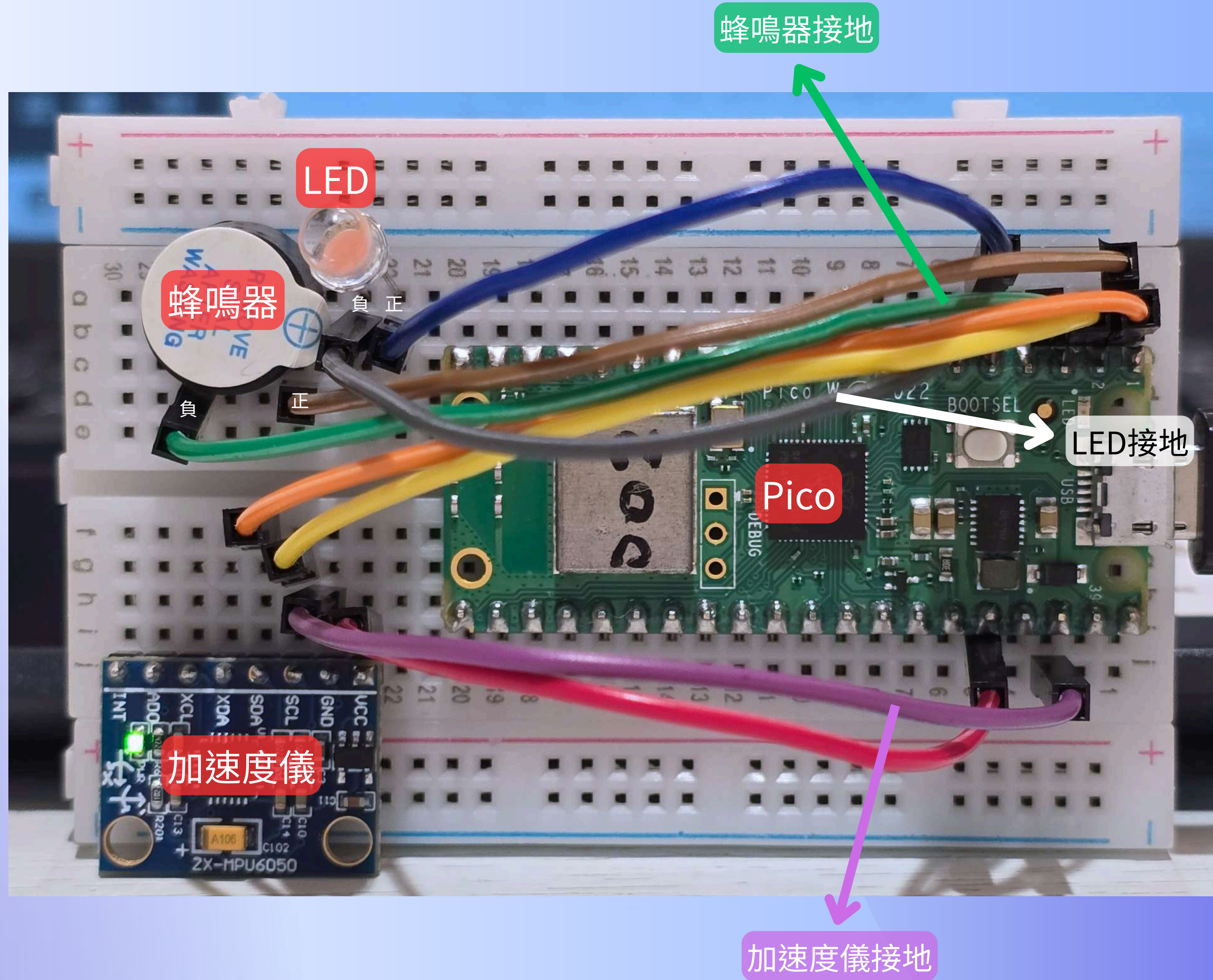


連接成功後Discord收到發送的訊息

Pico連接:



Pico連接:



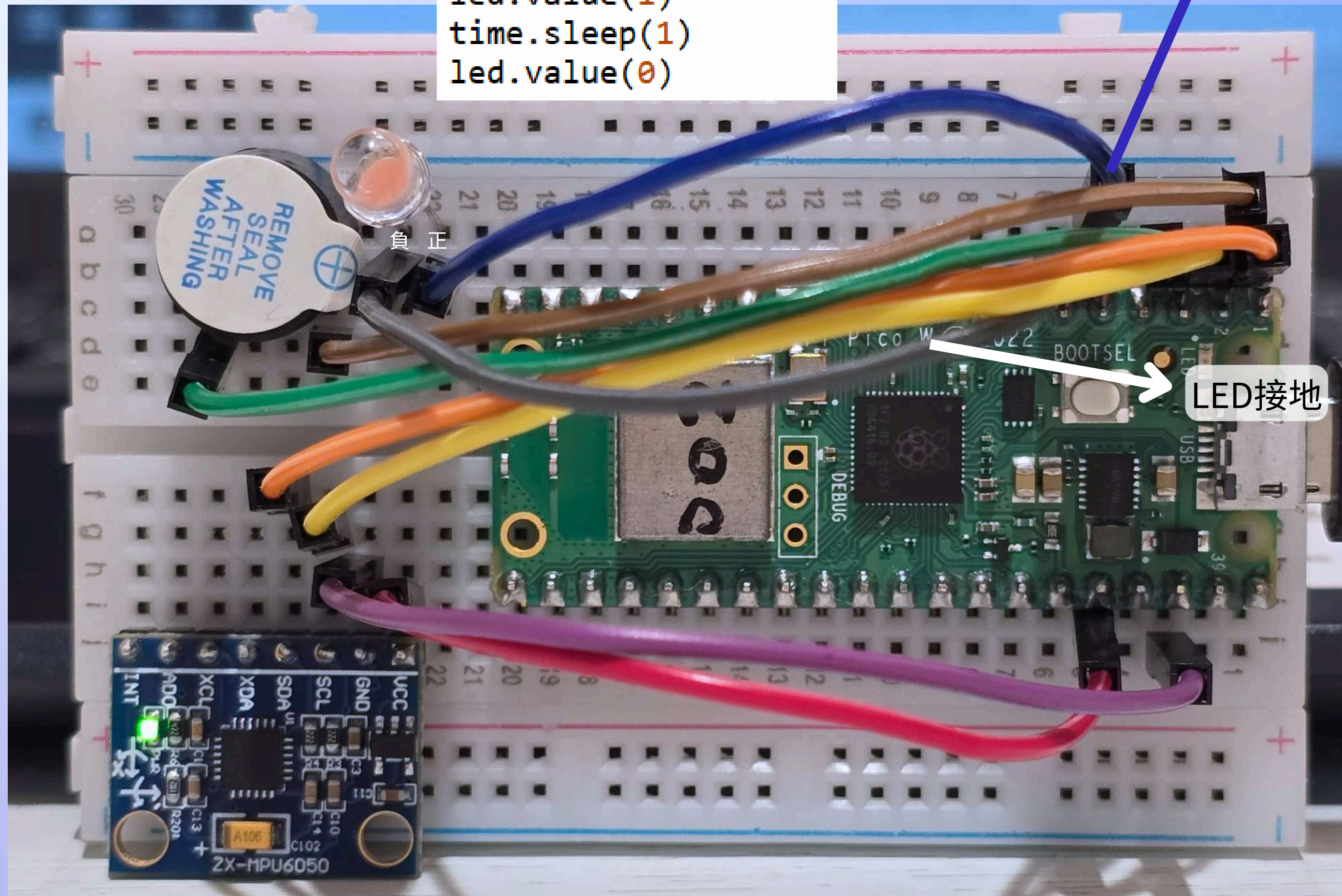
Pico連接:

LED

```
led = Pin(3,Pin.OUT)
led.value(1)
time.sleep(1)
led.value(0)
```

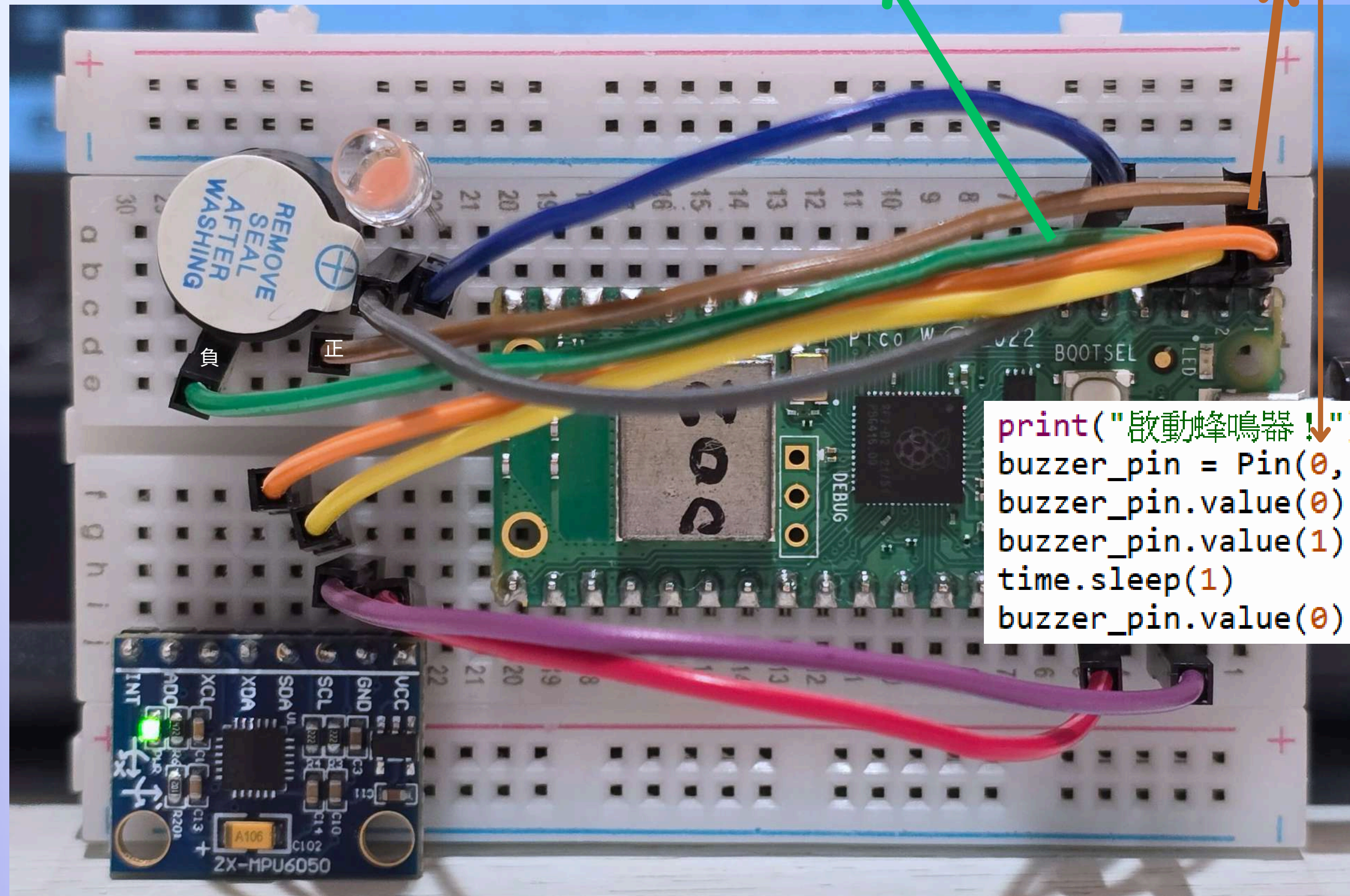
接GP3

LED接地



Pico連接:

蜂鳴器



```
print("啟動蜂鳴器!")  
buzzer_pin = Pin(0, Pin.OUT)  
buzzer_pin.value(0)  
buzzer_pin.value(1)  
time.sleep(1)  
buzzer_pin.value(0)
```


Discord webhook連結

```
16 # Discord Webhook URL
17 webhook_url = "https://discord.com/api/webhooks/1320714925114396763/g7qXTkXqg5Tr8lbDKCg9QJtBesf1w4nWQ_CxtKxd2SALH-4s_stgiZslxeVDRu0UWi00"
18
19 # 定義發送警報訊息到 Discord
20 def notify_alert(message, webhook_url):
21     payload = {"content": "earthquake occurrence!"}
22     try:
23         response = requests.post(webhook_url, json=payload)
24         if response.status_code == 204:
25             print("成功發送通知到 Discord。")
26         else:
27             print(f"發送通知失敗，狀態碼：{response.status_code}")
28     except Exception as e:
29         print(f"發送通知時出現錯誤：{e}")
```

地震發生時發送到Discord的文字內容

成功發送會顯示「成功發送通知到 Discord。」

發送失敗會顯示「發送通知失敗」以及狀態碼

*完整程式碼見附件

```

34 # 無限循環
35 while True:
36     try:
37         # 讀取加速度數據
38         ax = imu.accel.x
39         ay = imu.accel.y
40         az = imu.accel.z
41
42         # 計算總加速度（不考慮方向）
43         total_acceleration = (ax**2 + ay**2 + az**2)**0.5
44
45         # 偵測地震
46         if total_acceleration > THRESHOLD:
47             consecutive_count += 1
48             if consecutive_count >= CONSECUTIVE_COUNT and not earthquake_detected:
49                 print("⚠ 確認地震發生！")
50                 # 發送警報訊息到 Discord
51                 alert_message = f"⚠ 確認地震發生！\n總加速度: {total_acceleration:.2f} g"
52                 notify_alert(alert_message, webhook_url)
53                 earthquake_detected = True # 設置標誌為已偵測到地震
54
55                 led = Pin(3, Pin.OUT)
56                 led.value(1)
57                 time.sleep(1)
58                 led.value(0)
59
60                 print("啟動蜂鳴器！")
61                 buzzer_pin = Pin(0, Pin.OUT)
62                 buzzer_pin.value(0)
63                 buzzer_pin.value(1)
64                 time.sleep(1)
65                 buzzer_pin.value(0)
66
67         else:
68             consecutive_count = 0 # 若未超過閾值，重置計數

```

計算加速度值

判斷加速度值是否超過設定的閾值

地震發生LED發亮

地震發生蜂鳴器鳴叫

*完整程式碼見附件

↑ 加速度未達設定值，不進行反應

實際操作影片

www.BANDICAM.COM

discord.com/channels/947504565278875659/13207...

ASUS VivoBook

DC_test.py · 地震監測.py · blink.py · GPIO_LED.py

```
1 from machine import I2C, Pin
2 import time
3 from imu import MPU6050
4 import requests
5
6 # 初始化 I2C 和 MPU6050
7 i2c = I2C(0, scl=Pin(1), sda=Pin(0), freq=400000)
8 imu = MPU6050(i2c)
9
10 # 設定地震偵測參數
11 THRESHOLD = 1.0 # 加速度閾值 (單位 g)
12 INTERVAL = 0.5 # 每次讀取的時間間隔 (秒)
13 CONSECUTIVE_COUNT = 3 # 持續超過閾值的次數判定地震
14 count = 0
15
16 # 發送消息到 Discord
17 def alert(message, webhook_url):
18     payload = {"content": message}
19     try:
20         response = requests.post(webhook_url, json=payload)
21         if response.status_code == 204:
22             print("成功發送通知到 Discord。")
23         else:
24             print(f"發送通知失敗，狀態碼：{response.status_code}")
25     except:
26         print("發送通知失敗")
27
28 # 主程式
29 while True:
30     # 讀取加速度數據
31     accel = imu.get_acceleration()
32     # 檢查是否超過閾值
33     if accel > THRESHOLD:
34         count += 1
35         if count == CONSECUTIVE_COUNT:
36             alert("earthquake occurrence!", "https://discord.com/api/webhooks/1320714925...")
37             count = 0
38     time.sleep(INTERVAL)
```

Hello from Raspberry Pi Pico w!

earthquake occurrence!

earthquake occurrence!

傳訊息到 #地震

互動環境

程式中斷，關閉蜂鳴器。

MPY: soft reboot

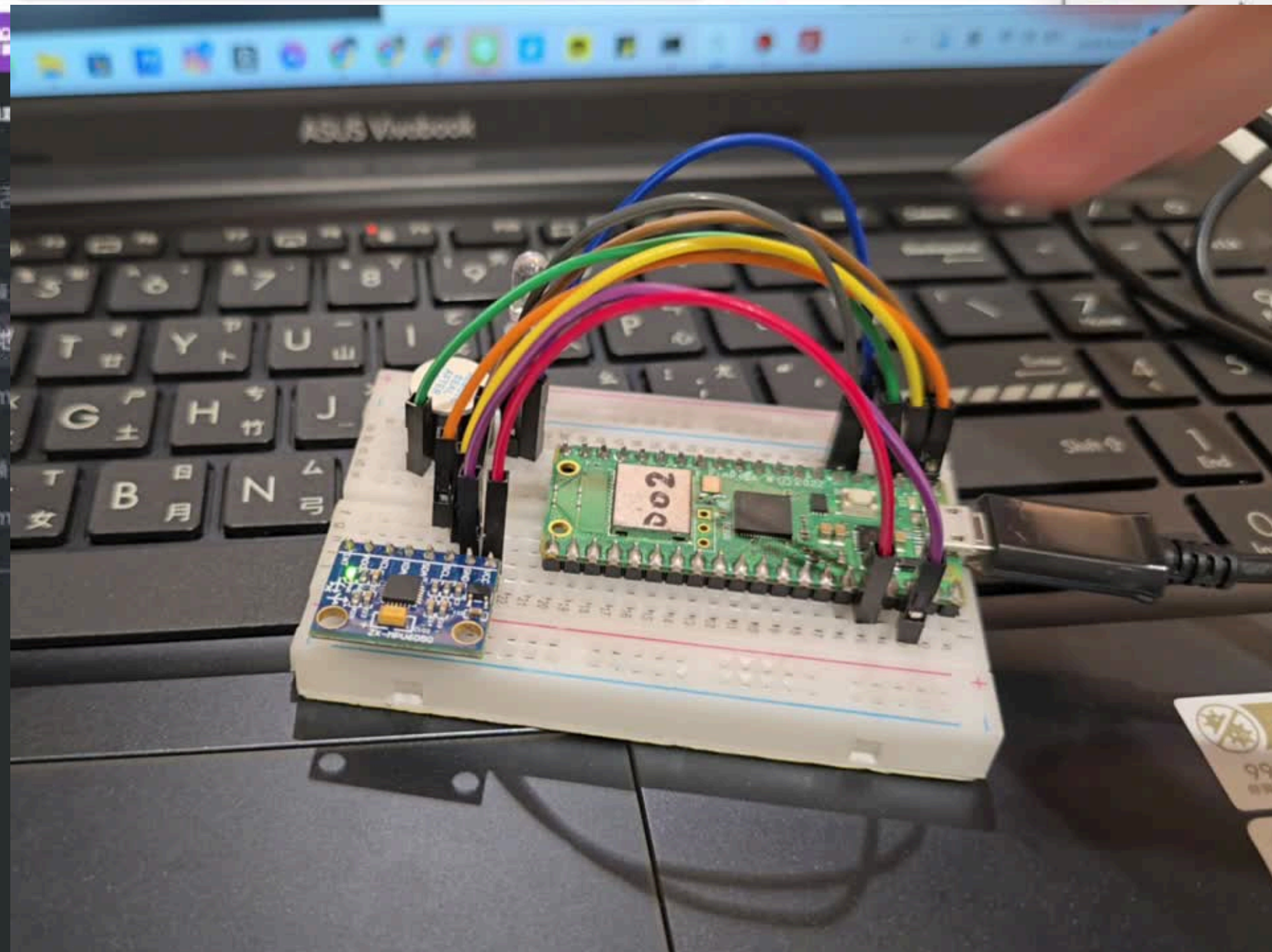
MicroPython v1.23.0 on 2024-06-02; Raspberry Pi Pico W with RP2040

Type "help()" for more information.

>>>

MicroPython (Raspberry Pi Pico) • Board CDC @ COM5

discord.com/channels/947504565278875659/13207...



DC_test.py 地震監測.py blink.py GPIO_LED.py

```
1 from machine import I2C, Pin
2 import time
3 from imu import MPU6050
4 import requests
5
6 # 初始化 I2C 和 MPU6050
7 i2c = I2C(0, scl=Pin(1), sda=Pin(0), freq=400000)
8 imu = MPU6050(i2c)
9
10 # 設定地震偵測參數
11 THRESHOLD = 1.0 # 加速度閾值 (單位 g)
12 INTERVAL = 0.5 # 每次讀取的時間間隔 (秒)
13 CONSECUTIVE_COUNT = 3 # 持續超過閾值的次數判定地震
14 consecutive_count = 0
15
16 # Discord Webhook URL
17 webhook_url = "https://discord.com/api/webhooks/1320714925
18
19 # 定義發送警報訊息到 Discord
20 def notify_alert(message, webhook_url):
21     payload = {"content": "earthquake occurrence!"}
22     try:
23         response = requests.post(webhook_url, json=payload)
24         if response.status_code == 204:
25             print("成功發送通知到 Discord。")
26         else:
27             print(f"發送通知失敗，狀態碼：{response.status_c
```

互動環境

MicroPython (Raspberry Pi Pico) • Board CDC @ COM5

```
ax: 0.42 g, ay: 2.00 g, az: 1.47 g
gx: 250.13 °/s, gy: 250.13 °/s, gz: 250.13 °/s
Temperature: 22.34 °C
總加速度: 2.52 g
```

⚠ 確認地震發生！

偵測到地震之後LED燈亮起

偵測到地震之後Discord接收到訊息



地震監測 應用 今天 晚上8:20
earthquake occurrence!

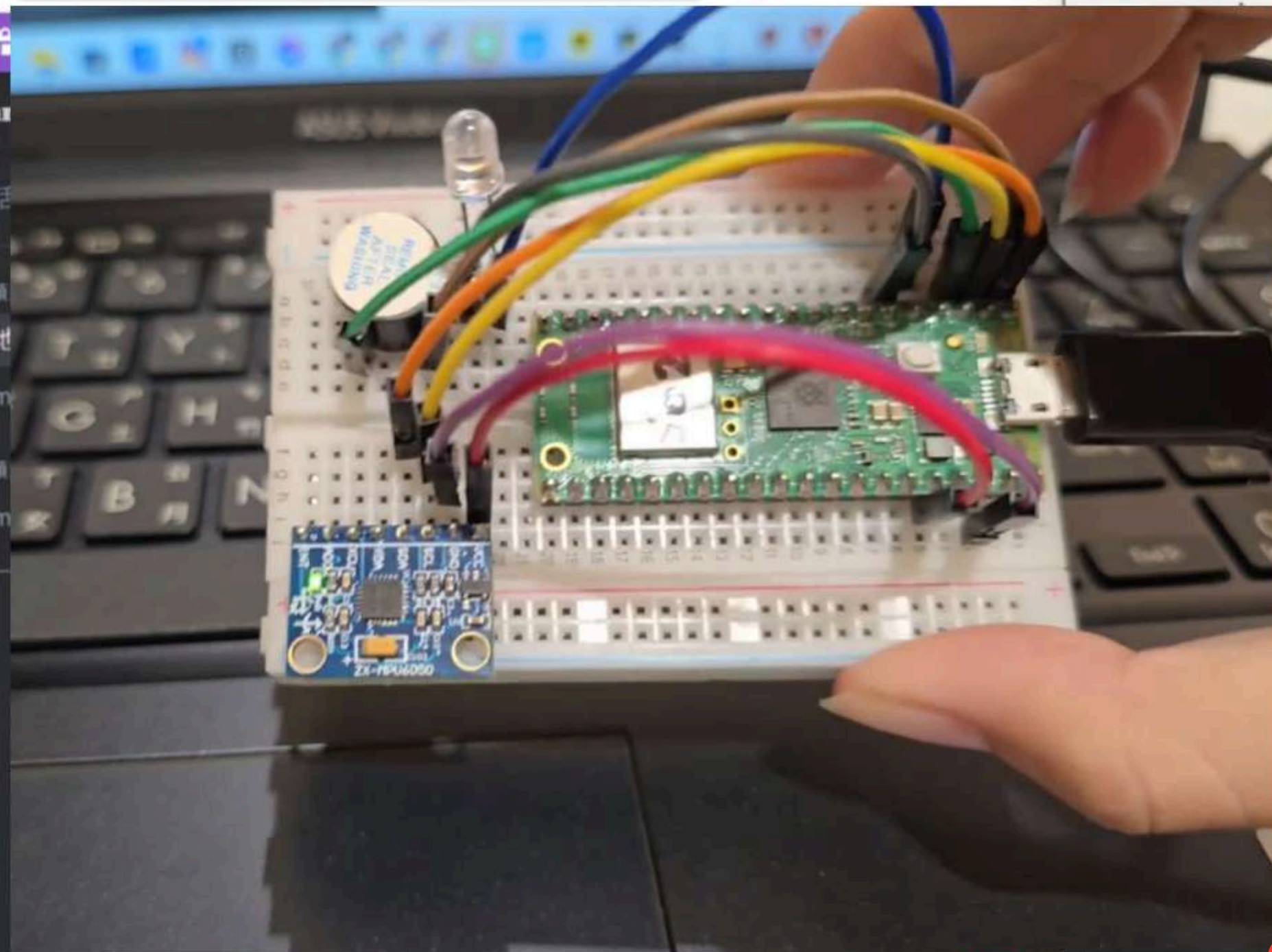
```
DC_test.py 地震監測.py blink.py GPIO_LED.py
1 from machine import I2C, Pin
2 import time
3 from imu import MPU6050
4 import requests
5
6 # 初始化 I2C 和 MPU6050
7 i2c = I2C(0, scl=Pin(1), sda=Pin(0), freq=400000)
8 imu = MPU6050(i2c)
9
10 # 設定地震偵測參數
11 THRESHOLD = 1.0 # 加速度閾值 (單位 g)
12 INTERVAL = 0.5 # 每次讀取的時間間隔 (秒)
13 CONSECUTIVE_COUNT = 3 # 持續超過閾值的次數判定地震
14 consecutive_count = 0
15
16 # Discord Webhook URL
17 webhook_url = "https://discord.com/api/webhooks/1320714925
18
19 # 定義發送警報訊息到 Discord
20 def notify_alert(message, webhook_url):
21     payload = {"content": "earthquake occurrence!"}
22     try:
23         response = requests.post(webhook_url, json=payload)
24         if response.status_code == 204:
25             print("成功發送通知到 Discord。")
26         else:
27             print(f"發送通知失敗，狀態碼：{response.status_c
```

互動環境

```
ax: 0.42 g, ay: 2.00 g, az: 1.47 g
gx: 250.13 °/s, gy: 250.13 °/s, gz: 250.13 °/s
Temperature: 22.34 °C
總加速度: 2.52 g
```

⚠ 確認地震發生！
成功發送通知到 Discord。

discord.com/channels/947504565278875659/13207...



DC_test.py 地震監測.py blink.py GPIO_LED.py

```
1 from machine import I2C, Pin
2 import time
3 from imu import MPU6050
4 import requests
5
6 # 初始化 I2C 和 MPU6050
7 i2c = I2C(0, scl=Pin(1), sda=Pin(0), freq=400000)
8 imu = MPU6050(i2c)
9
10 # 設定地震偵測參數
11 THRESHOLD = 1.0 # 加速度閾值 (單位 g)
12 INTERVAL = 0.5 # 每次讀取的時間間隔 (秒)
13 CONSECUTIVE_COUNT = 3 # 持續超過閾值的次數判定地震
14 consecutive_count = 0
15
16 # Discord Webhook URL
17 webhook_url = "https://discord.com/api/webhooks/1320714925
18
19 # 定義發送警報訊息到 Discord
20 def notify_alert(message, webhook_url):
21     payload = {"content": "earthquake occurrence!"}
22     try:
23         response = requests.post(webhook_url, json=payload)
24         if response.status_code == 204:
25             print("成功發送通知到 Discord。")
26         else:
27             print(f"發送通知失敗，狀態碼：{response.status_c
```

確認蜂鳴器有正常啟動

⚠ 確認地震發生！
成功發送通知到 Discord。
啟動蜂鳴器！