

CoreMaker 操作指南

目錄

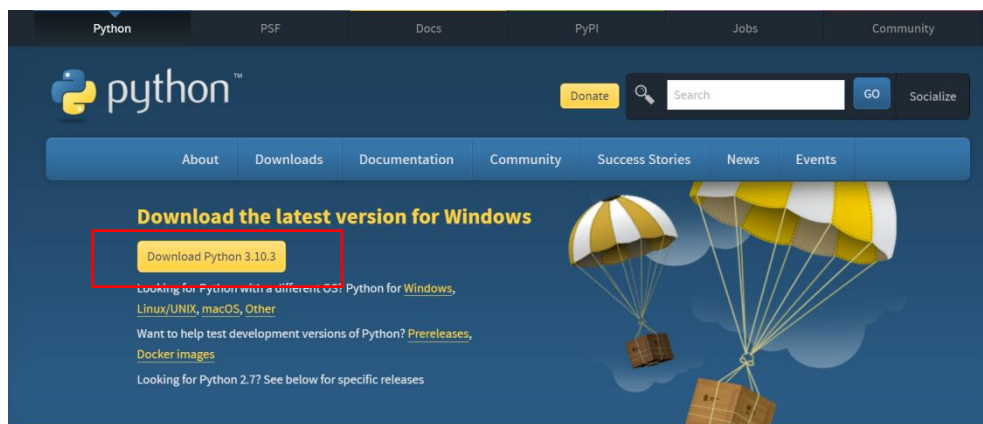
壹、 架設環境 (WINDOWS 10).....	2
一、 PYTHON.....	2
二、 GIT.....	3
三、 CMAKE.....	7
四、 NINJA.....	8
五、 MBED CLI 2.....	9
六、 GNU ARM EMBEDDED TOOLCHAIN.....	10
貳、 下載及編譯韌體原始碼.....	12
一、 下載.....	12
二、 編譯.....	12
參、 燒錄模式.....	14
肆、 加入 AI.....	16
一、 事前準備.....	16
二、 建立 DATA CAPTURE LAB 新專案.....	17
三、 資料蒐集.....	17
四、 模型訓練.....	21
五、 建立雲端檔案.....	24
六、 更新韌體.....	26
DOCUMENT HISTORY.....	29

壹、架設環境 (Windows 10)

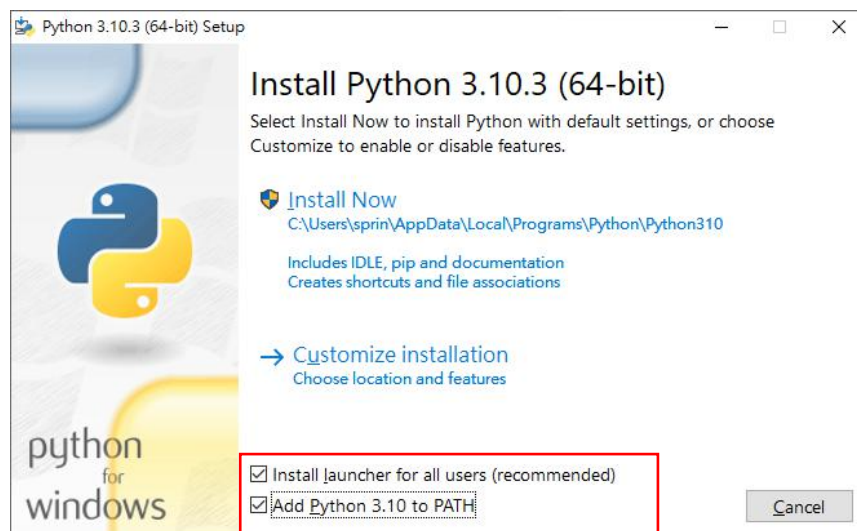
- 請參考以下附圖，完成下載及設定

一、Python

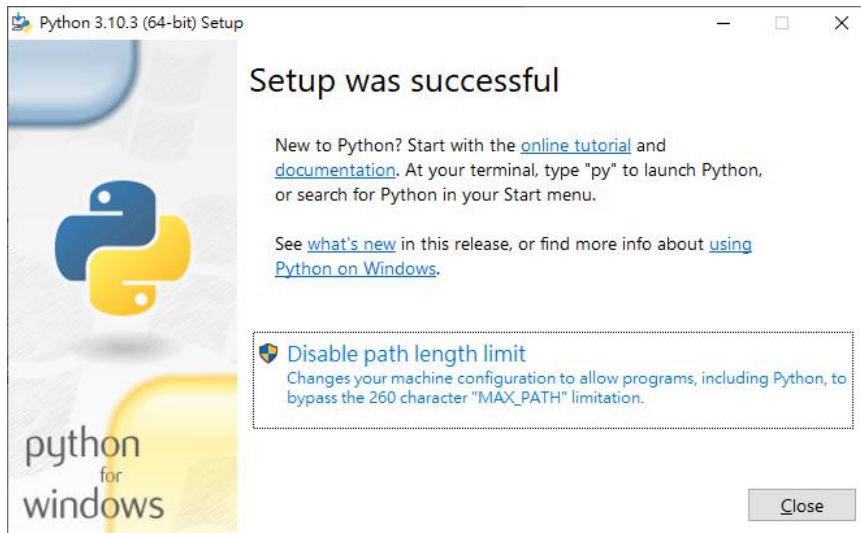
- 安裝 Python 3.6 以上版本，下載網址：<https://www.python.org/downloads/>，下載後執行安裝檔開始安裝流程



- 勾選 “Add Python 3.10 to PATH”



- 點擊 “Disable path length limit”

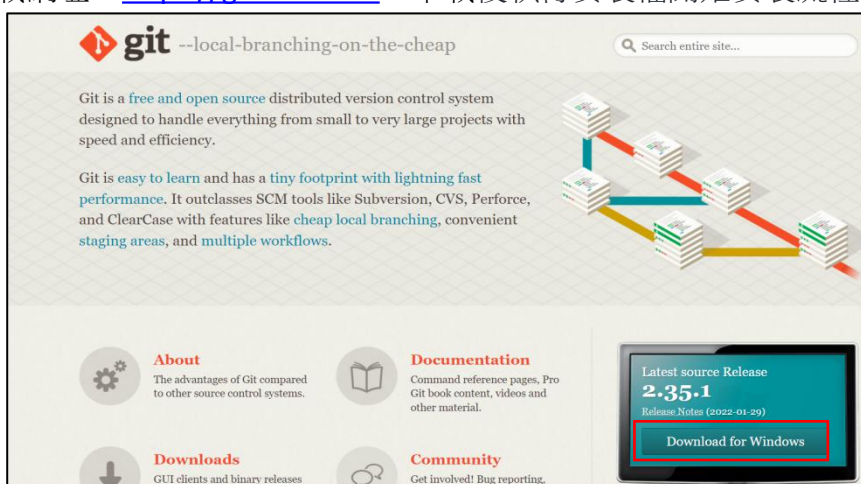


- 安裝完成可以到命令提示字元輸入 `python`，若安裝成功，則會進入 `python` 互動模式，並且顯示版本相關訊息，輸入 `"exit()"` 離開互動模式

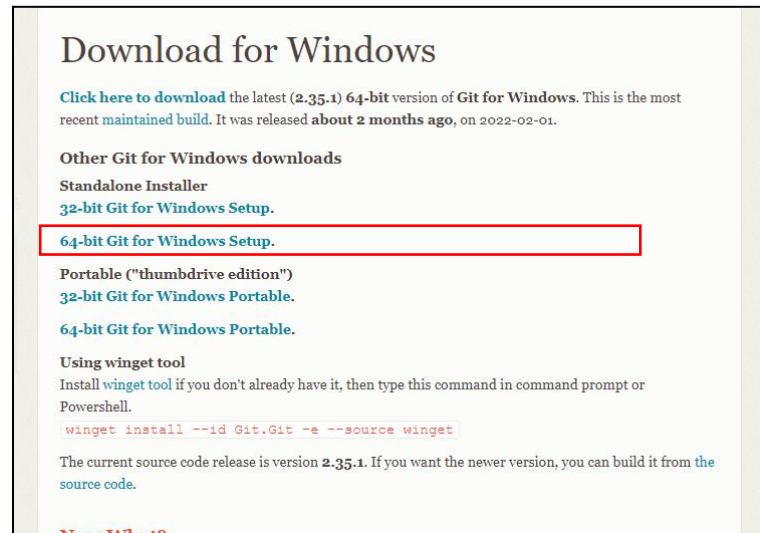


二、Git

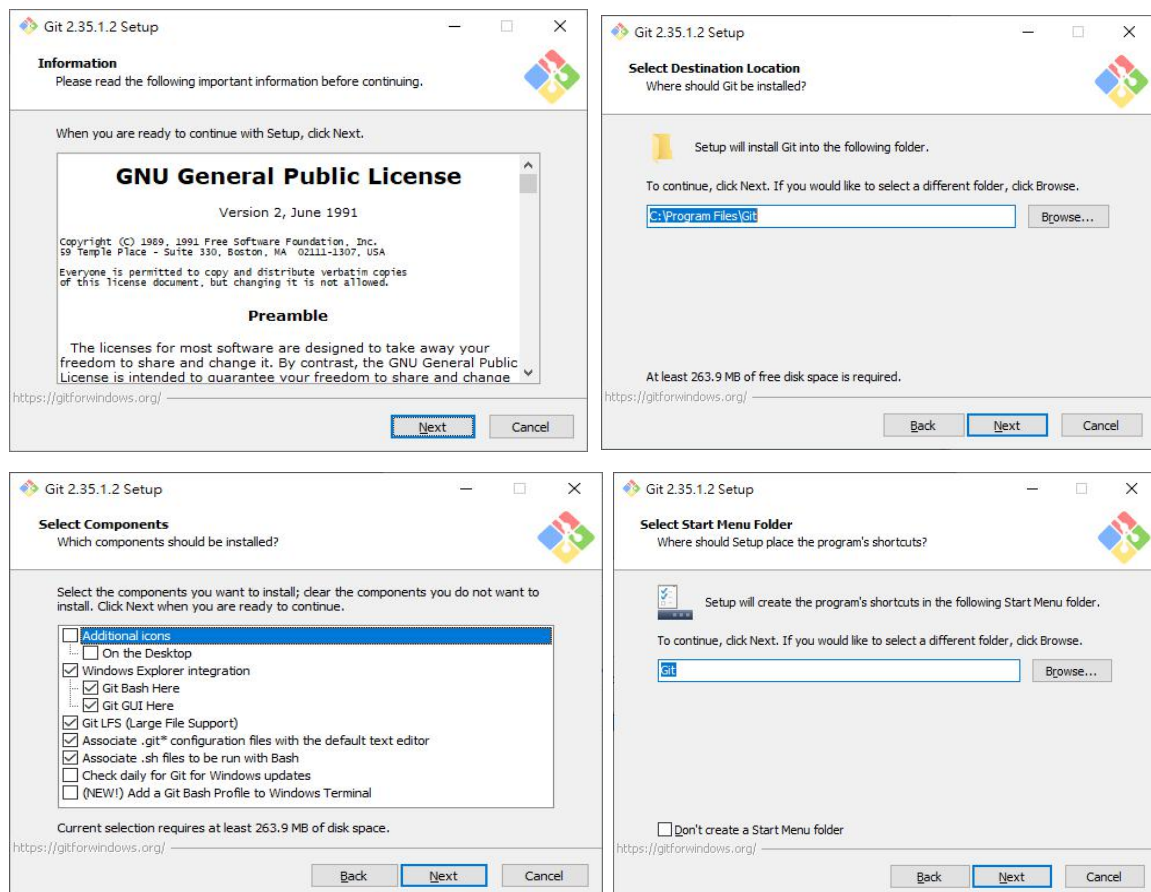
- 安裝檔下載網址：<https://git-scm.com>，下載後執行安裝檔開始安裝流程



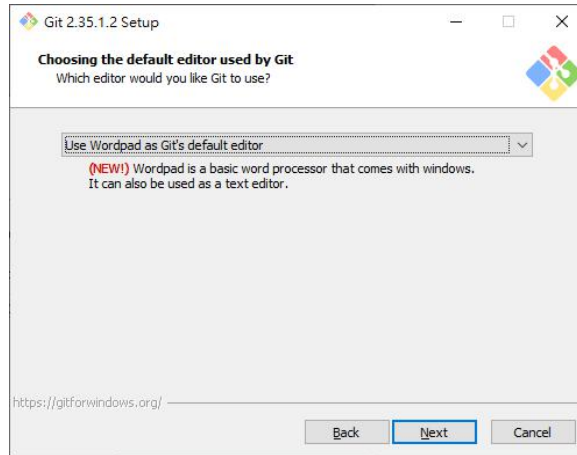
- 選擇 “64-bit Git for Windows Setup”



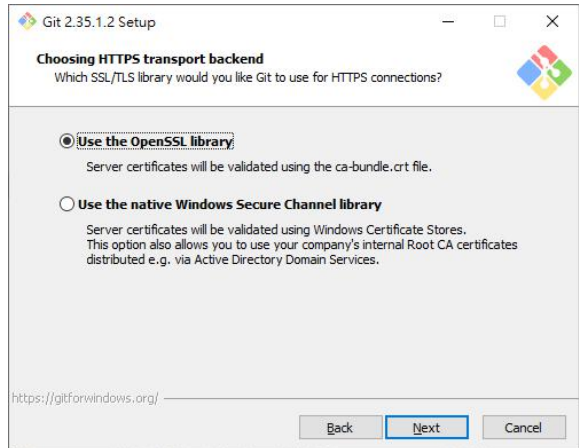
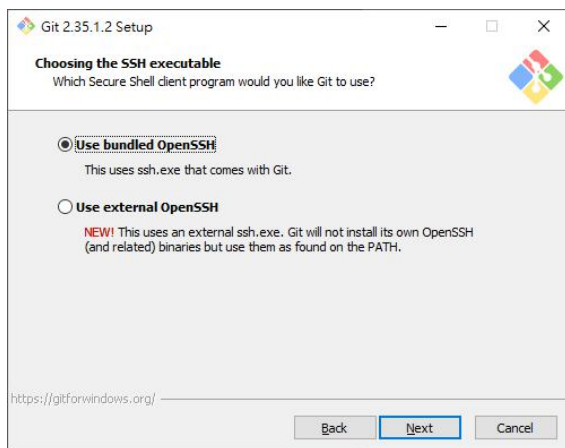
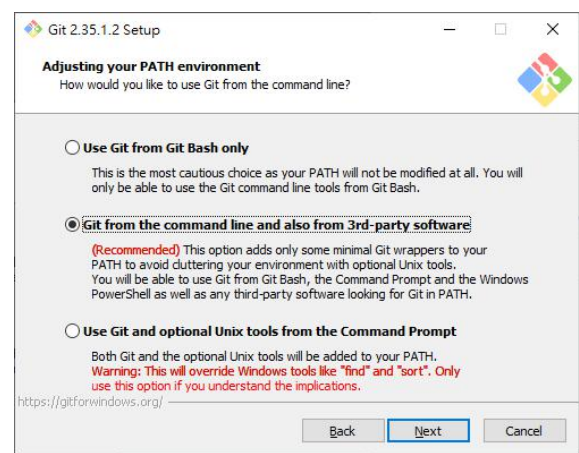
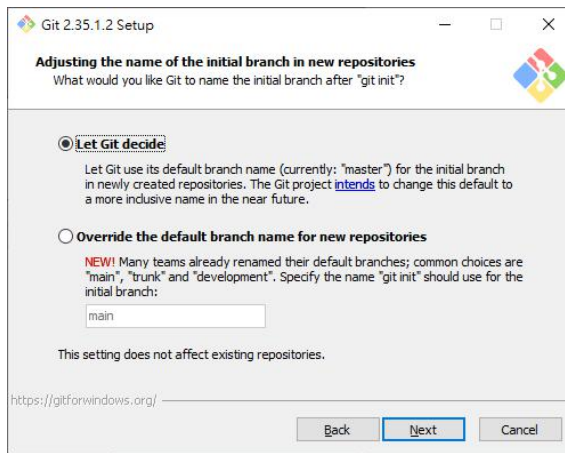
- 這邊除非有特別需求，否則使用預設值即可

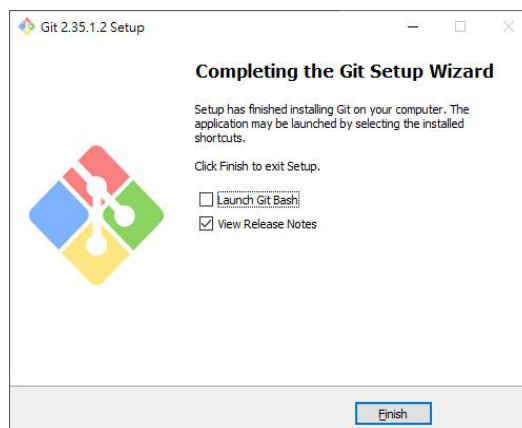
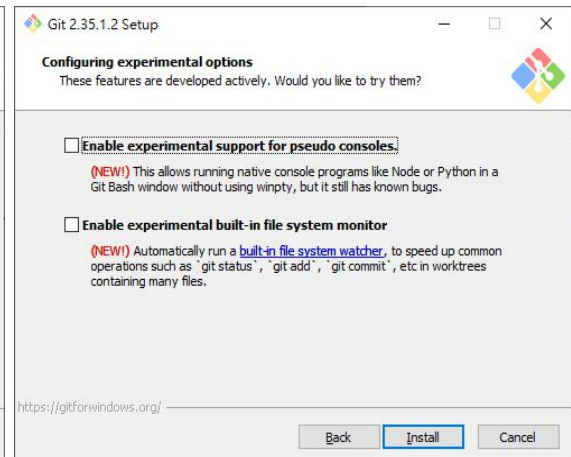
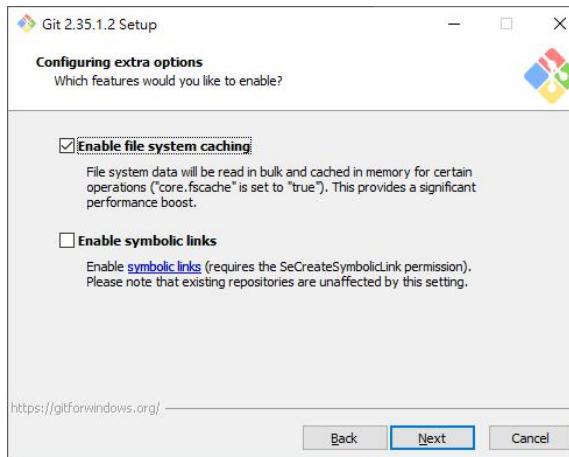
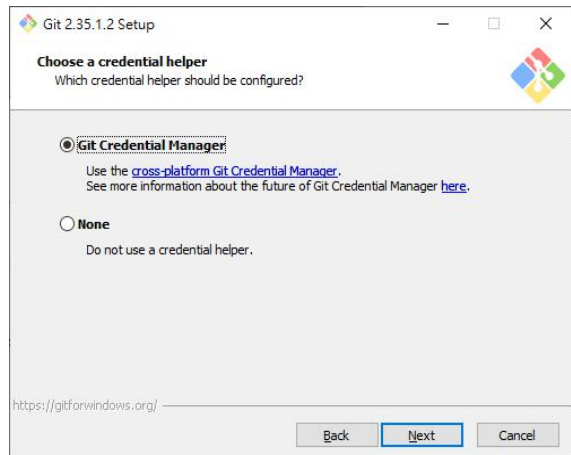
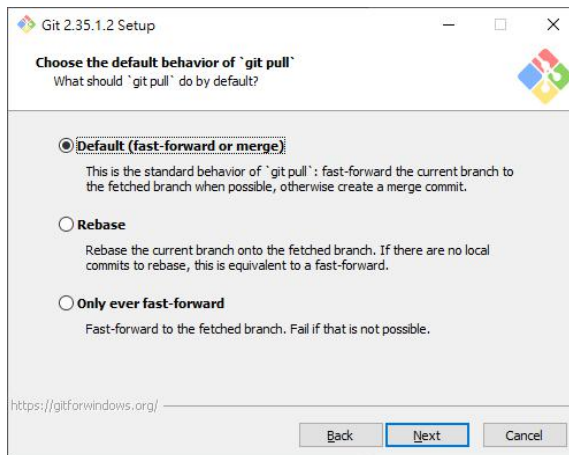
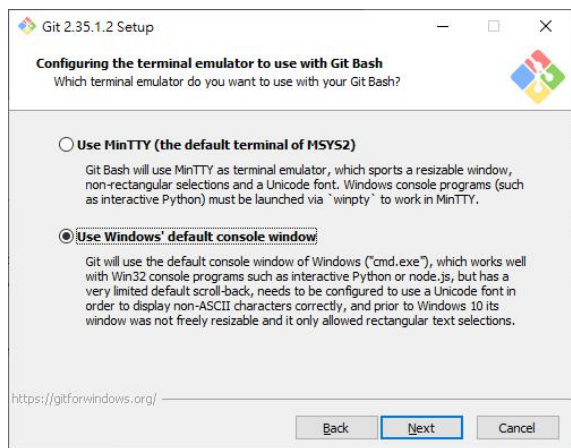
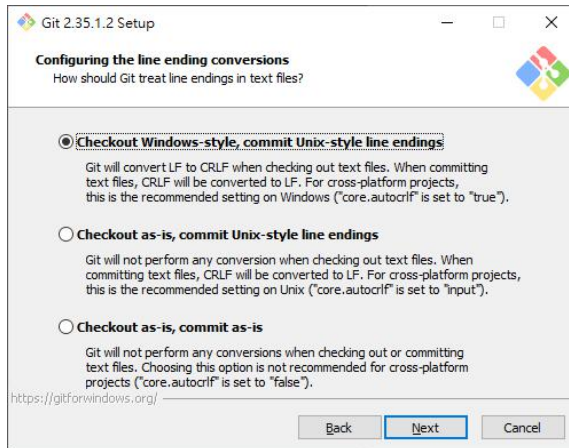


- 在 Choosing the default editor used by Git 步驟，可選擇 PC 中有且慣用的編輯器



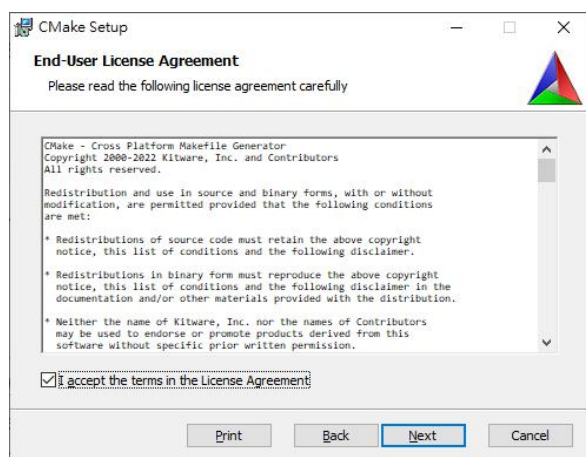
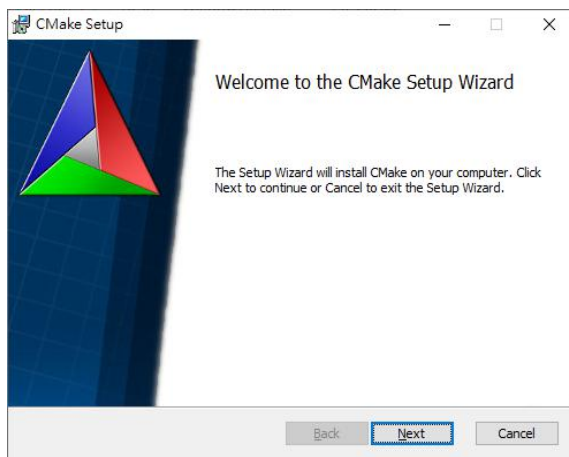
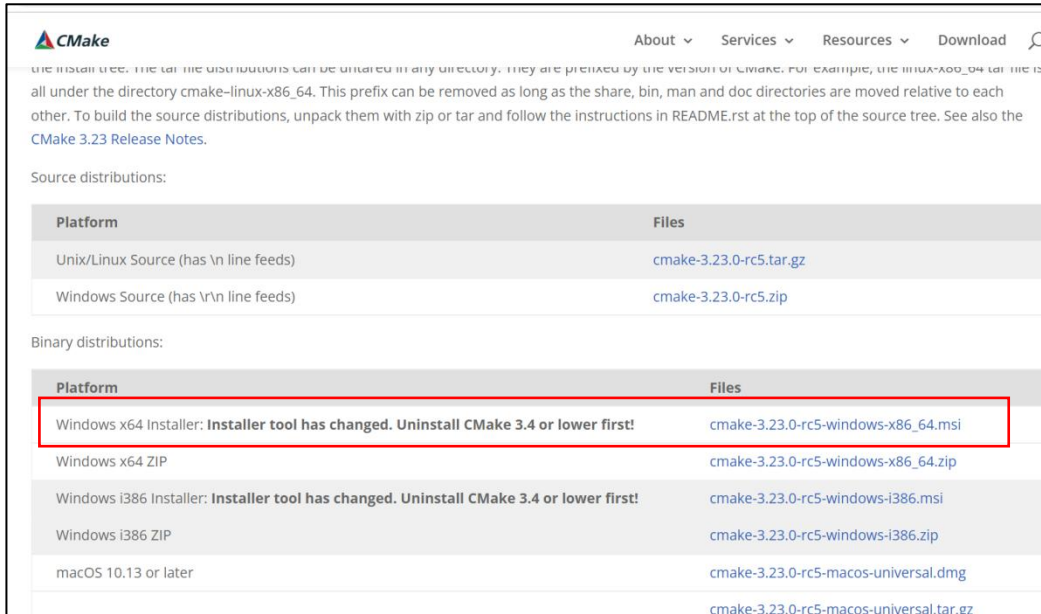
- 後續步驟請參考以下圖示中選擇的選項進行設定



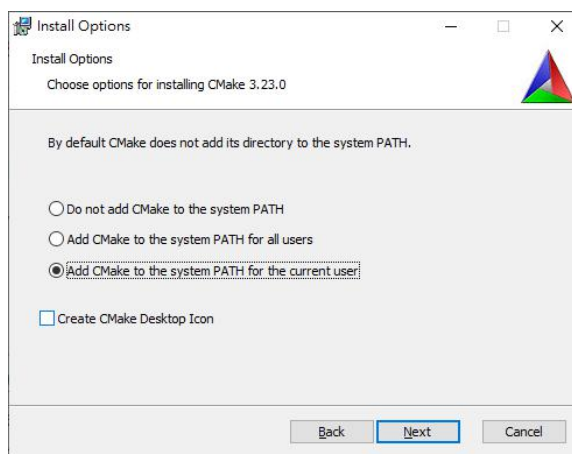


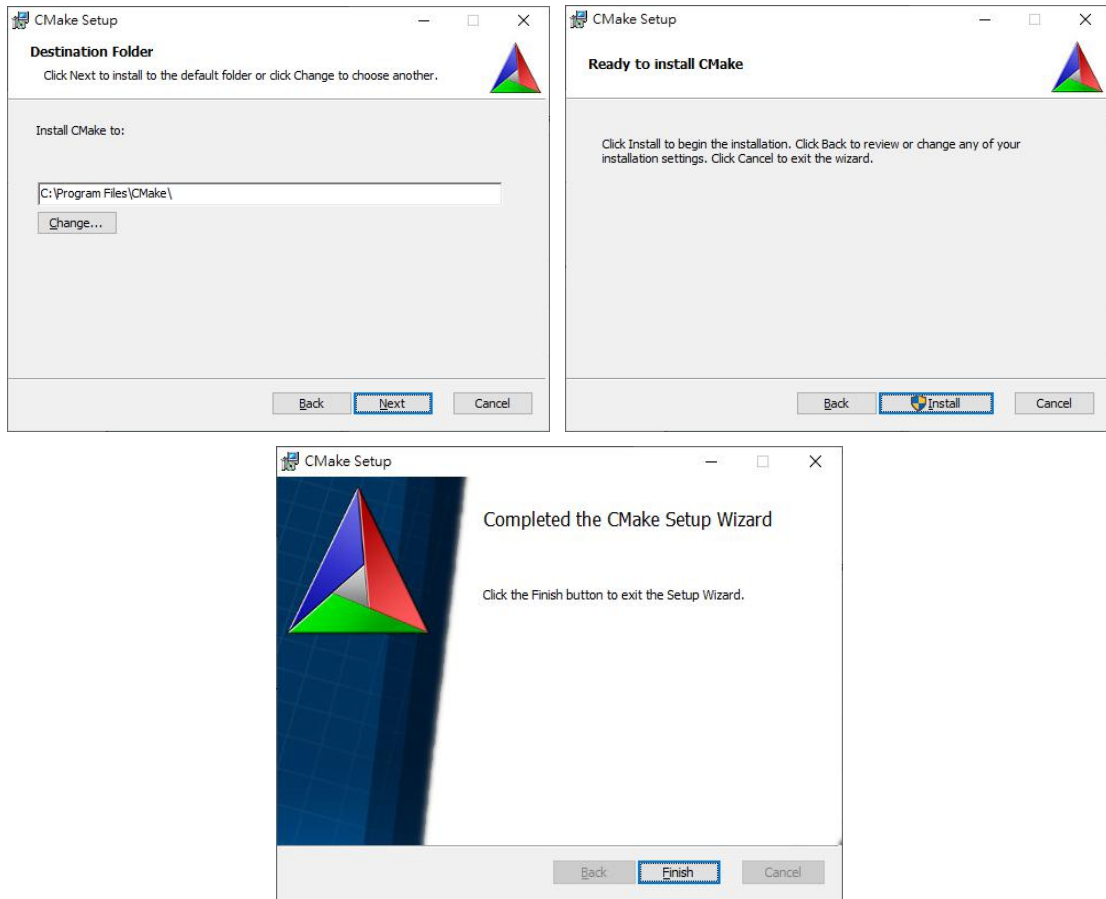
三、CMake

- 安裝 CMake 3.19.0 以上版本，安裝檔下載網址：<https://cmake.org/download/>，下載後執行安裝檔開始安裝流程



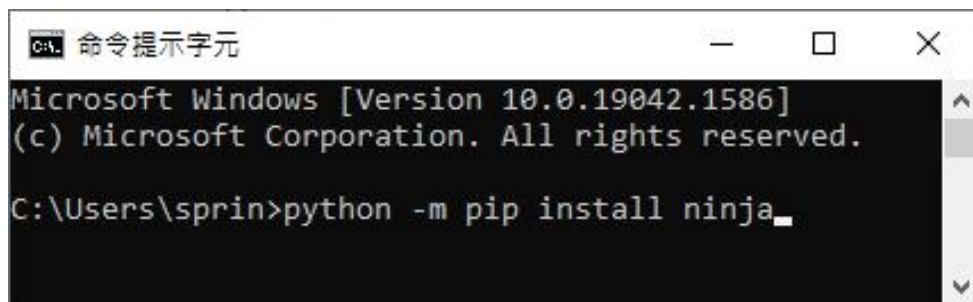
- 此處選則 “Add CMake to the system PATH for the current user”



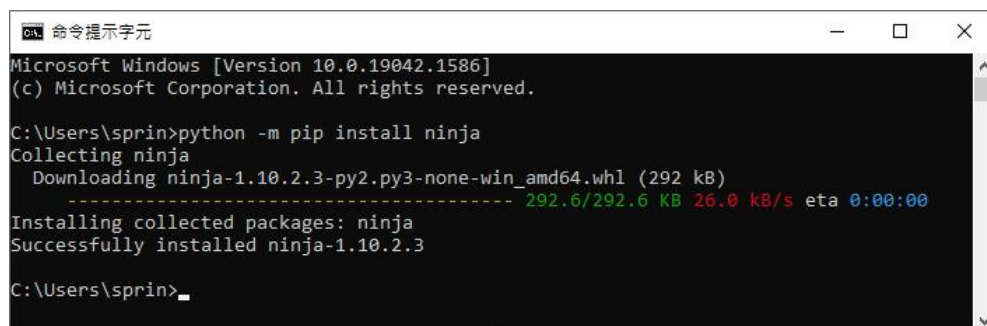


四、Ninja

- 安裝 Ninja 1.0 以上版本，開啟命令提示字元，輸入 “python -m pip install ninja”

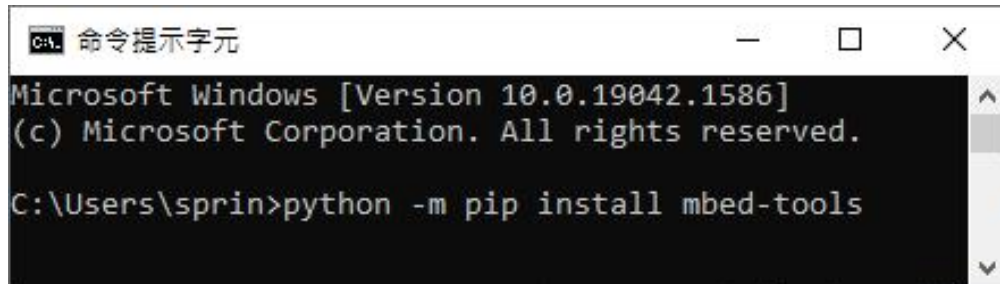


- 安裝完成後會出現如下圖的訊息



五、Mbed CLI 2

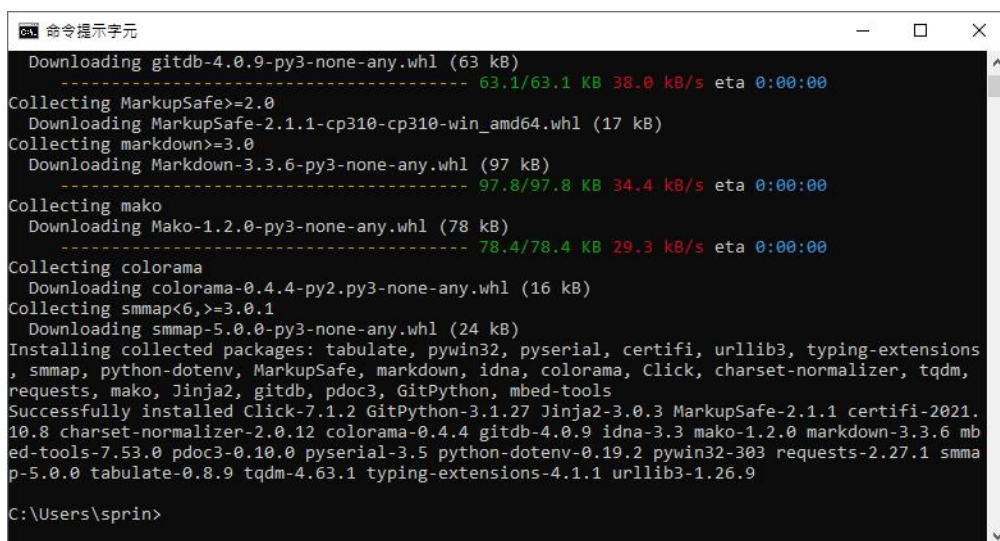
- 開啟命令提示字元，輸入 “python -m pip install mbed-tools”



```
命令提示字元
Microsoft Windows [Version 10.0.19042.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sprin>python -m pip install mbed-tools
```

- 安裝完成後會出現如下圖的訊息

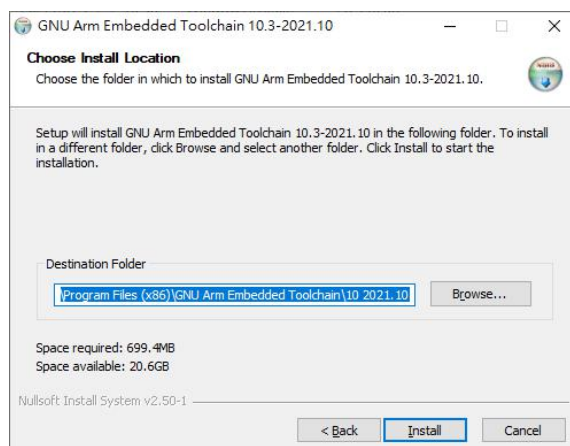
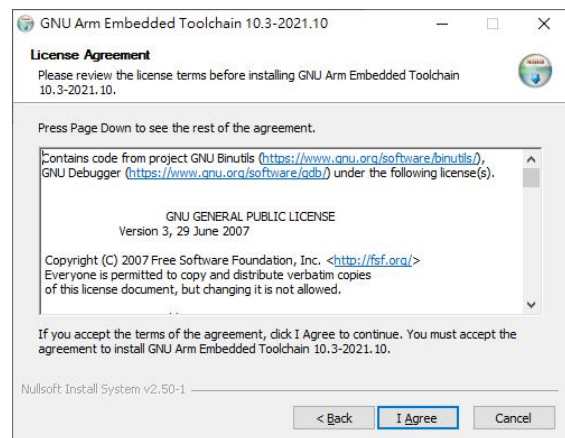
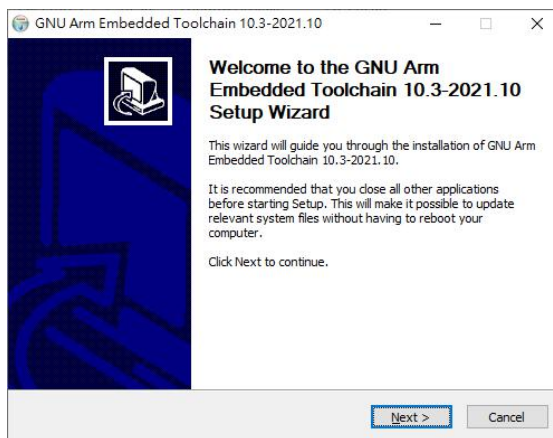
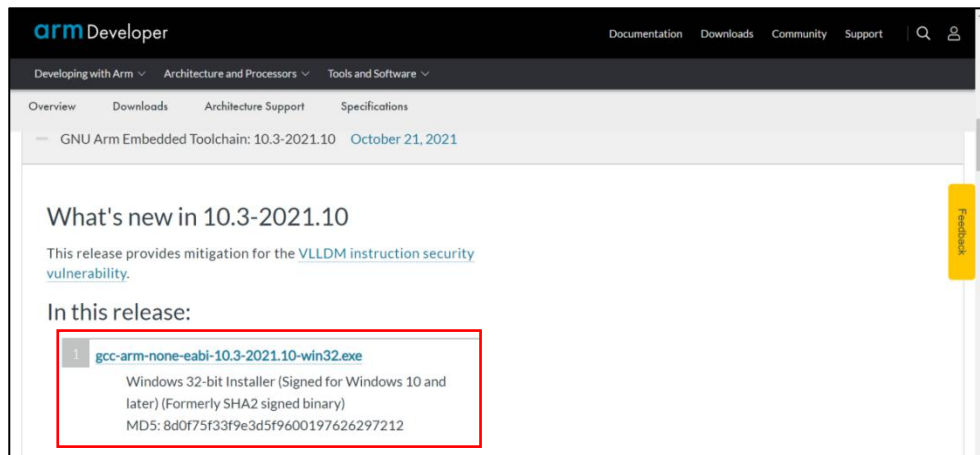


```
命令提示字元
Downloading gitdb-4.0.9-py3-none-any.whl (63 kB)
----- 63.1/63.1 KB 38.0 kB/s eta 0:00:00
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.1-cp310-cp310-win_amd64.whl (17 kB)
Collecting markdown>=3.0
  Downloading Markdown-3.3.6-py3-none-any.whl (97 kB)
----- 97.8/97.8 KB 34.4 kB/s eta 0:00:00
Collecting mako
  Downloading Mako-1.2.0-py3-none-any.whl (78 kB)
----- 78.4/78.4 KB 29.3 kB/s eta 0:00:00
Collecting colorama
  Downloading colorama-0.4.4-py2.py3-none-any.whl (16 kB)
Collecting smmap<6,>=3.0.1
  Downloading smmap-5.0.0-py3-none-any.whl (24 kB)
Installing collected packages: tabulate, pywin32, pyserial, certifi, urllib3, typing-extensions, smmap, python-dotenv, MarkupSafe, markdown, idna, colorama, Click, charset-normalizer, tqdm, requests, mako, Jinja2, gitdb, pdoc3, GitPython, mbed-tools
Successfully installed Click-7.1.2 GitPython-3.1.27 Jinja2-3.0.3 MarkupSafe-2.1.1 certifi-2021.10.8 charset-normalizer-2.0.12 colorama-0.4.4 gitdb-4.0.9 idna-3.3 mako-1.2.0 markdown-3.3.6 mbed-tools-7.53.0 pdoc3-0.10.0 pyserial-3.5 python-dotenv-0.19.2 pywin32-303 requests-2.27.1 smmap-5.0.0 tabulate-0.8.9 tqdm-4.63.1 typing-extensions-4.1.1 urllib3-1.26.9

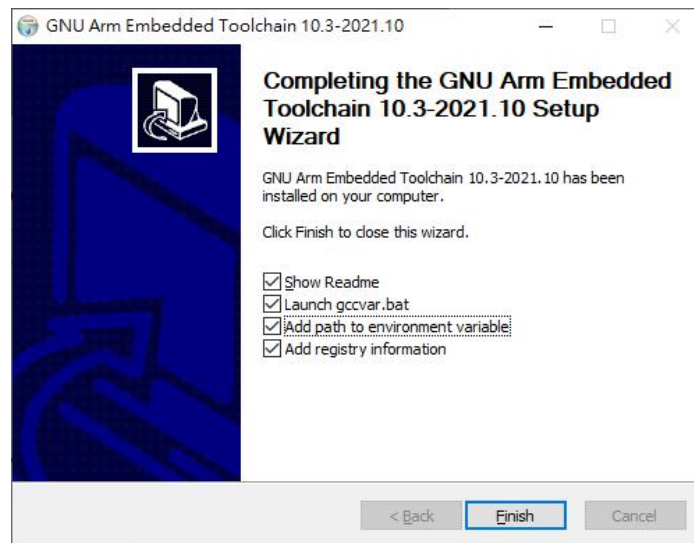
C:\Users\sprin>
```

六、GNU Arm Embedded Toolchain

- 安裝檔下載網址：
<https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads>
- 根據網頁說明，選擇下載 “gcc-arm-none-eabi-10.3-2021.10-win32.exe”



- 勾選 “Add path to environment variable”

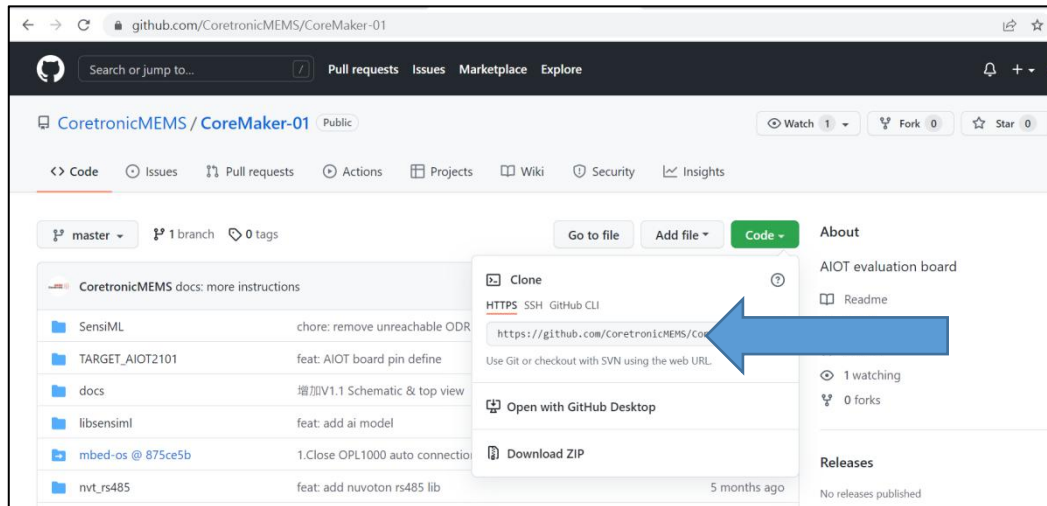


- 以上套件安裝完成後，需要重開機

貳、下載及編譯韌體原始碼

一、下載

- CoreMaker 韌體原始碼位置：<https://github.com/CoretronicMEMS/CoreMaker-01>



- 進入命令提示字元，輸入

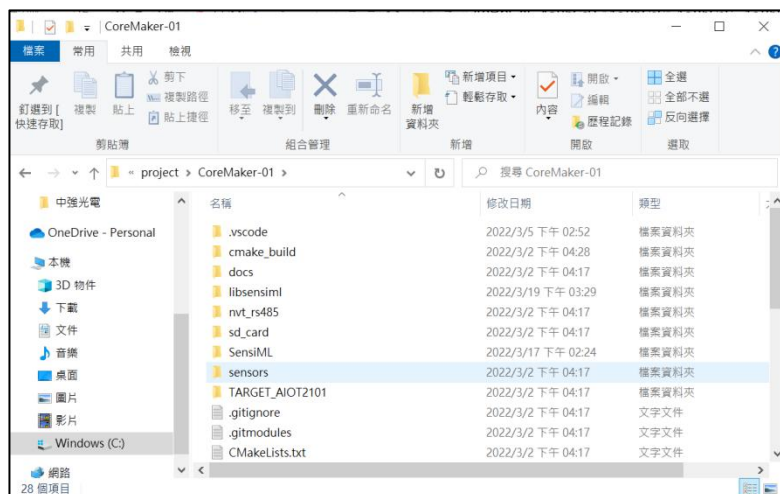
“git clone --recurse-submodules https://github.com/CoretronicMEMS/CoreMaker-01.git”

，指令後方網址即上圖箭頭處複製的內容

註：請勿直接使用 GitHub 網頁上的下載功能，因為會缺少 Mbed-OS 相關資料庫檔案



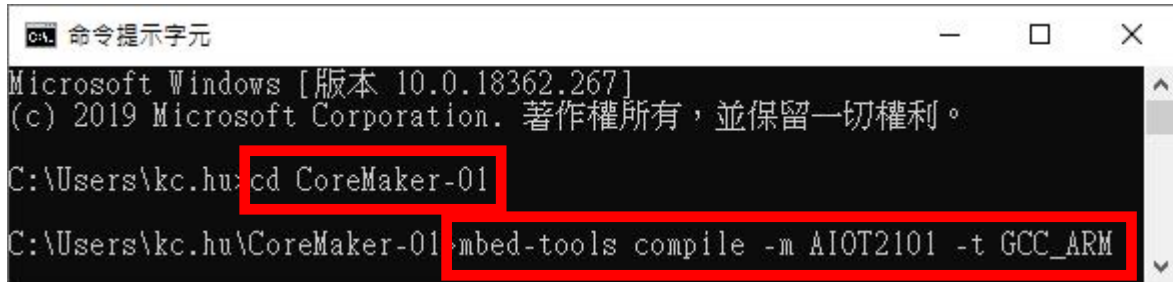
- 下載完成後，會產生 CoreMaker-01 資料夾，資料夾內容如下圖所示



二、編譯

- 在命令提示字元中，輸入 “cd CoreMaker-01” 進入韌體程式所在的資料夾

- 輸入 “mbed-tools compile -m AIOT2101 -t GCC_ARM” 開始編譯，初次編譯需要較長的時間

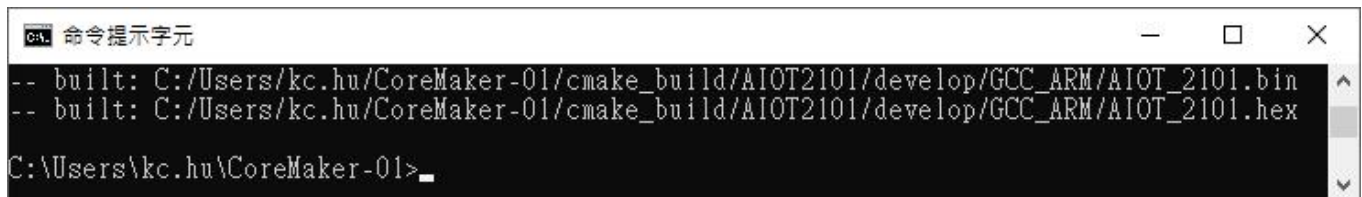


```

Microsoft Windows [版本 10.0.18362.267]
(c) 2019 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\kc.hu>cd CoreMaker-01
C:\Users\kc.hu\CoreMaker-01>mbed-tools compile -m AIOT2101 -t GCC_ARM
  
```

- 編譯完會在 CoreMaker-01\cmake_build\AIOT2101\develop\GCC_ARM\，產生 AIOT_2101.bin



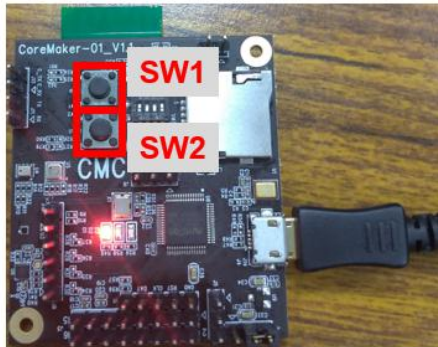
```

-- built: C:/Users/kc.hu/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.bin
-- built: C:/Users/kc.hu/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.hex

C:\Users\kc.hu\CoreMaker-01>_
  
```

叁、燒錄模式

- 使用 Micro USB to USB 轉接線，連接 CoreMaker 與 PC。
- 同時按下 SW1 及 SW2，後先放開 SW1，再放開 SW2，CoreMaker 上的紅色 LED 不再閃爍，表示進入燒錄模式，若紅色 LED 仍然閃爍，重複步驟。

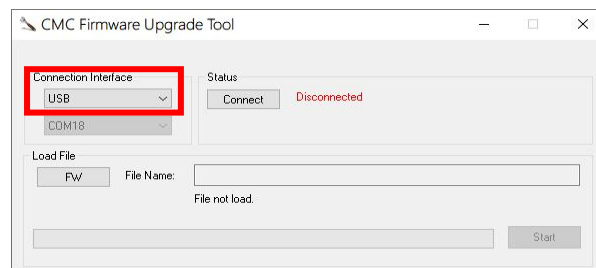


一般模式

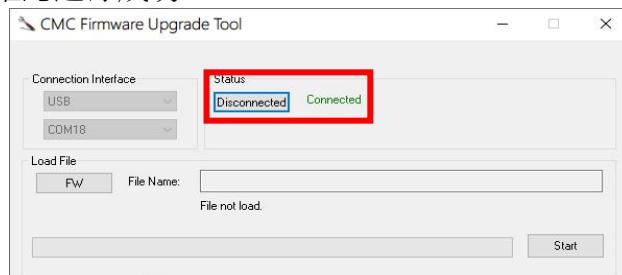


燒錄模式

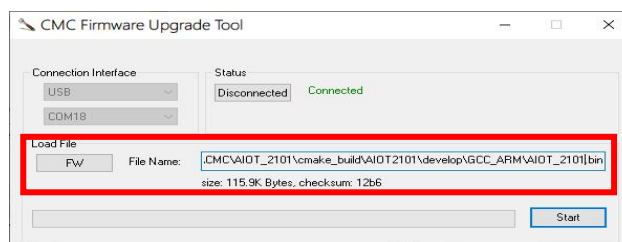
- 解壓縮 CoreMaker-01/CMC_ISP/CMC_ISP.zip，執行解開後的 CMC_ISP.exe 開啟程式，在 Connection Interface 選單選擇 “USB”



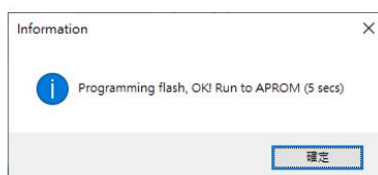
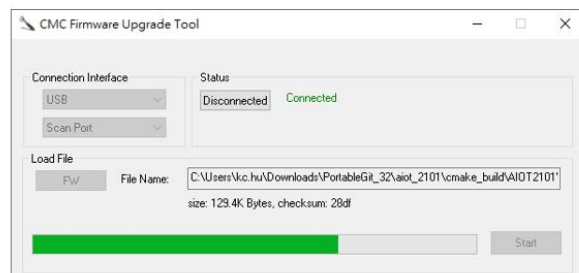
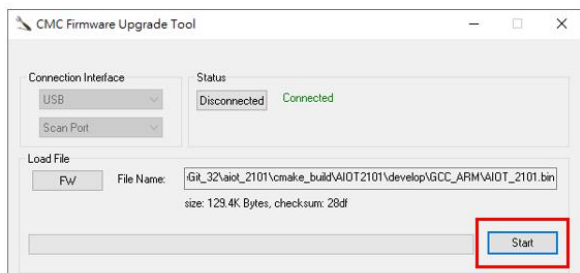
- 點擊 “Connect”，確認連線成功



- 選擇欲燒錄的 bin 檔案，預設檔案路徑為
CoreMaker01\cmake_build\AIOT2101\develop\GCC_ARM\AIOT_2101.bin



- 點擊 “start” 開始燒錄，燒錄成功會出現成功訊息

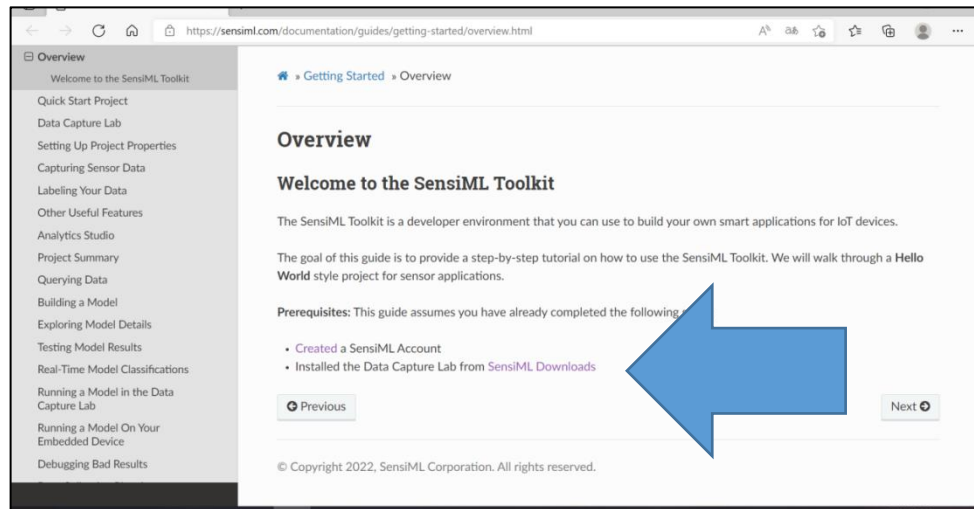


肆、加入 AI

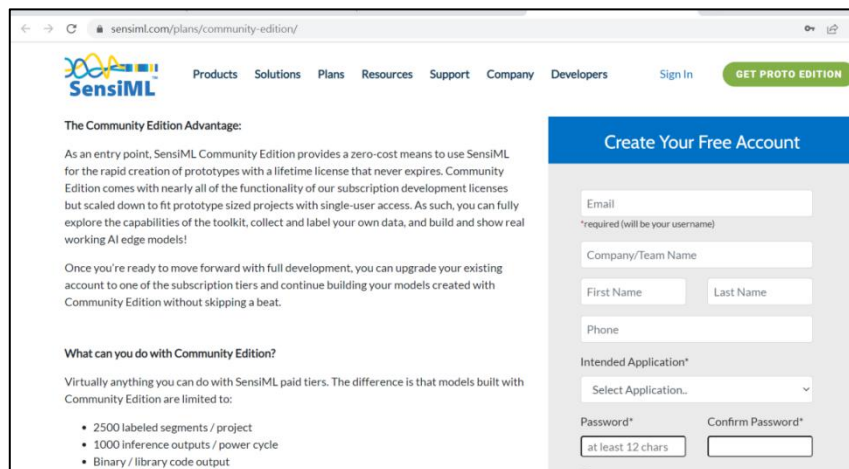
一、事前準備

- 參考網頁內容進行以下步驟：

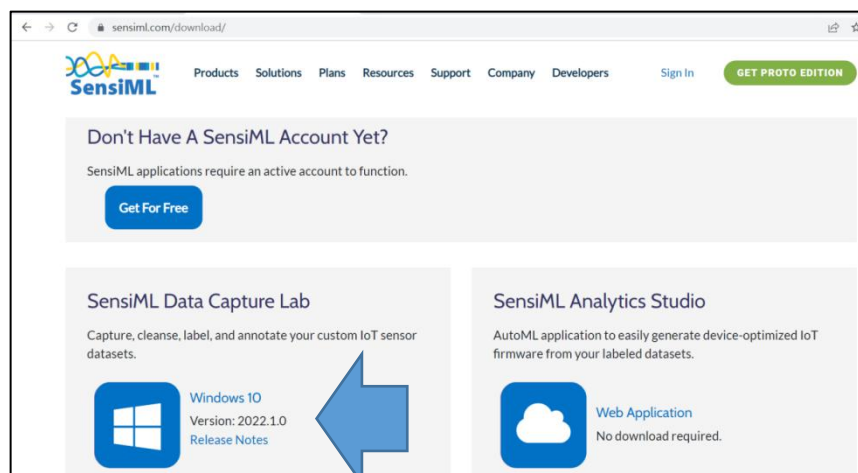
<https://sensiml.com/documentation/guides/getting-started/overview.htm>



- 建立 SensiML 帳號(註冊後，需到郵件確認帳號)。



- 進入下載頁面，點選下圖箭頭處，下載並安裝 Data Capture Lab 工具。

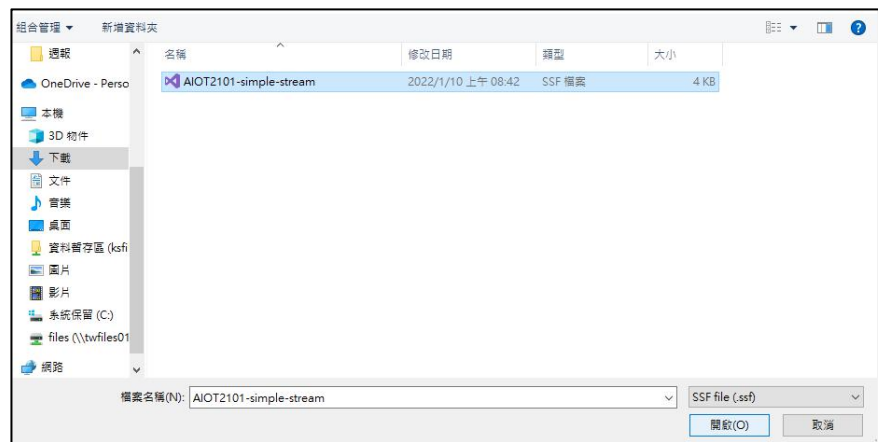
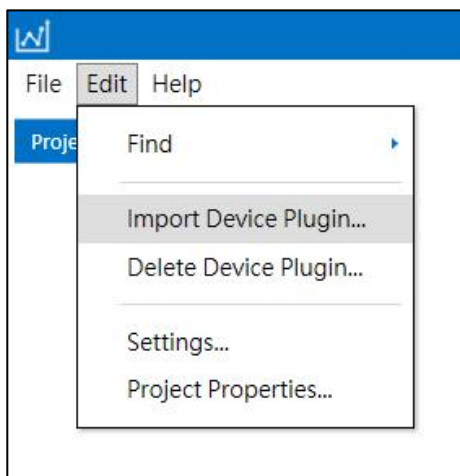


二、建立 Data Capture Lab 新專案

- 開啟 Data Capture Lab，建立新的專案

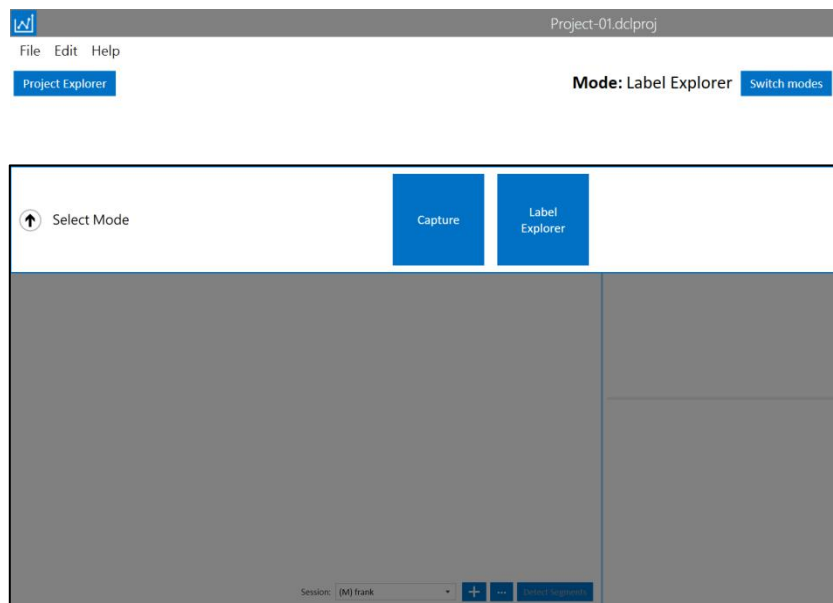


- 匯入裝置設定檔，位置為：CoreMaker-01\SensiML\AIOT2101\simple-stream.ssf。

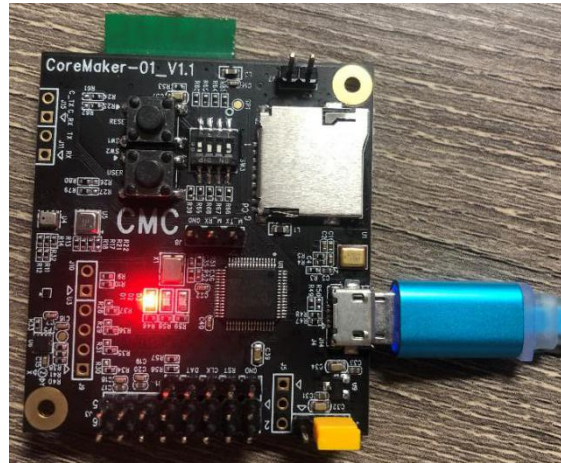


三、資料蒐集

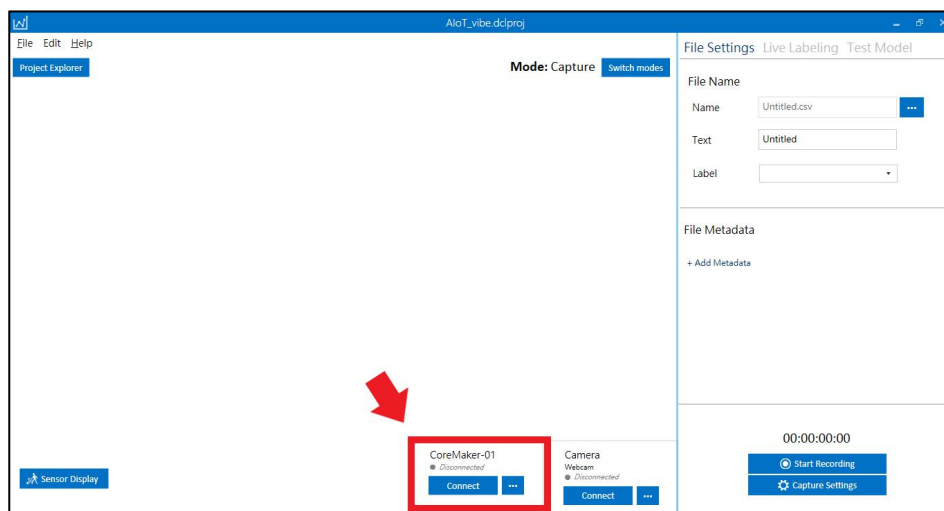
- 點選 Switch modes 按鈕，選擇 Capture。



- Micro USB to USB 的線，連接 CoreMaker-01。



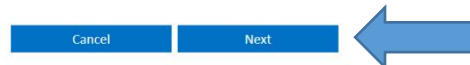
- 進入 Capture 頁面後，點選頁面下方 CoreMaker-01 的 Connect 按鈕。



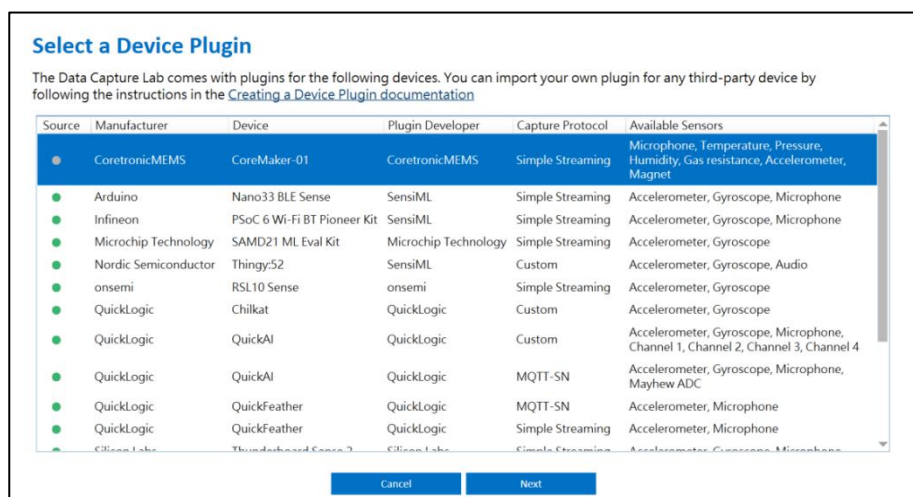
- 設定 sensor 參數
 - 按下 CoreMaker-01 的 Connect 按鈕後，出現畫面。

Sensor Configuration

Select a sensor configuration profile to use during data collection.
This will be saved as metadata to each file.



- 點選 CoretronicMEMS (Manufacture)、CoreMaker-01 (Device)選項。



- 顯示 Plugin Detail。

Plugin Details

Manufacturer: CoretronicMEMS
 Plugin Developer: CoretronicMEMS
 Capture Protocol: Simple Streaming
 Firmware Download: [Data Collection Firmware](#)

Sensor Summary:

- Name: Audio
Sensors: Microphone
Sample Rates: 16000, 8000, 4000, 2000, 1000, 400, 200, 100
- Name: Environment
Sensors: Temperature, Pressure, Humidity, Gas resistance
Sample Rates: 1
- Name: Motion
Sensors: Accelerometer
Sample Rates: 12800, 6400, 3200, 1600, 800, 400, 200, 100, 50, 25
- Name: Compass
Sensors: Magnet
Sample Rates: 200, 100, 50, 20, 10

Back Next

- 選擇 sensor (依照各需求選擇 sensor，此次先用 Audio 作為示範)。

Sensor Properties

Device Plugin
CoreMaker-01

Capture Source

- Audio
- Environment
- Motion
- Compass
- ☒ Microphone

Back Next

- 選擇 Sample rate (選擇 2000)。

Sensor Properties

Device Plugin
CoreMaker-01

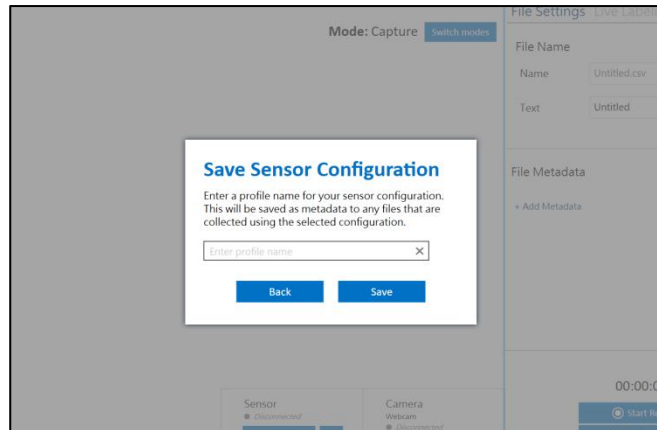
Capture Source
Audio

Sample Rate

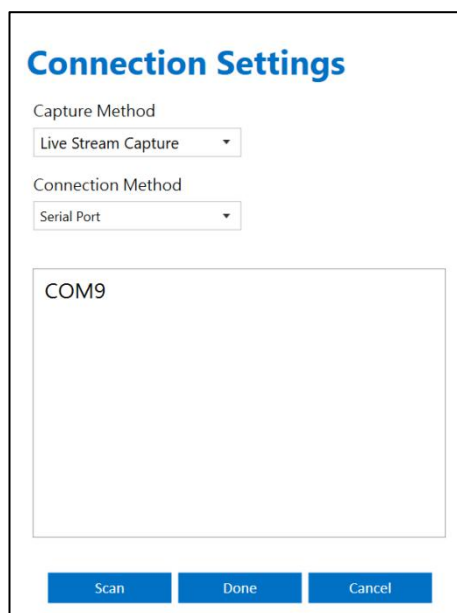
- 2000
- 16000
- 8000
- 4000
- 2000
- 1000
- 400
- 200
- 100

Back Next

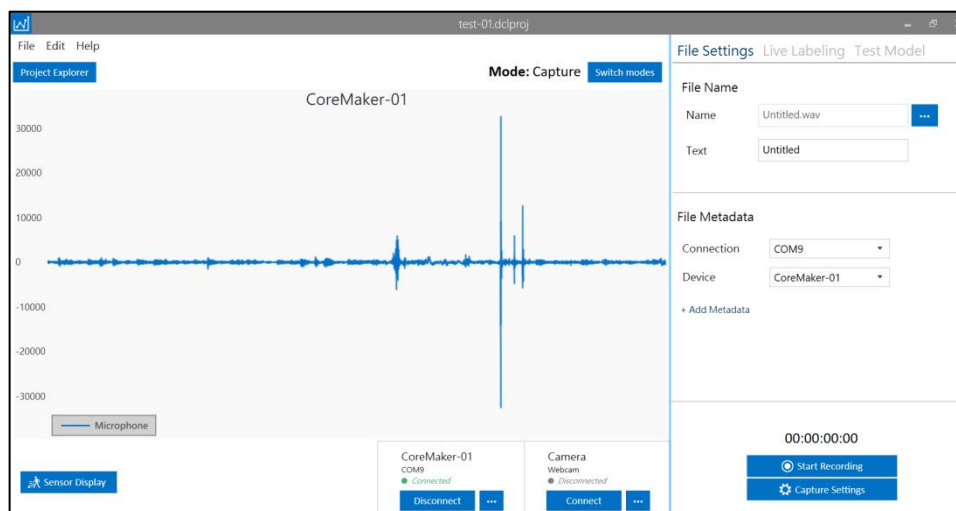
■ 檔名設定



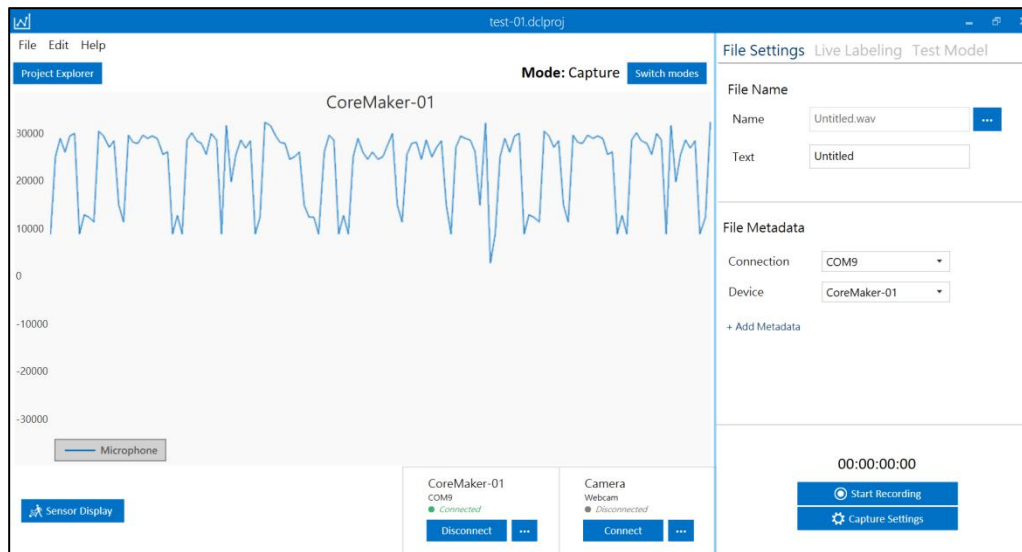
■ 按下 Scan，選擇訊號源(Micro USB to USB)。



- 再次點選頁面下方 CoreMaker-01 的 Connect 按鈕，當狀態列出現 Retrieving Configuration 時，按壓 CoreMaker-01 上的 SW2 按鍵即可連線。連線成功後，頁面中間會顯示輸入資料的波型圖。

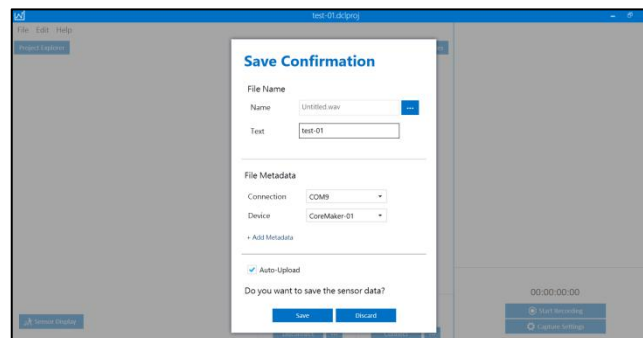


- 若連線失敗，或是波形圖異常 (靜止狀態但波型變化劇烈)，可按壓 SW1 (Reset) 按鈕，重新執行前步驟連線。

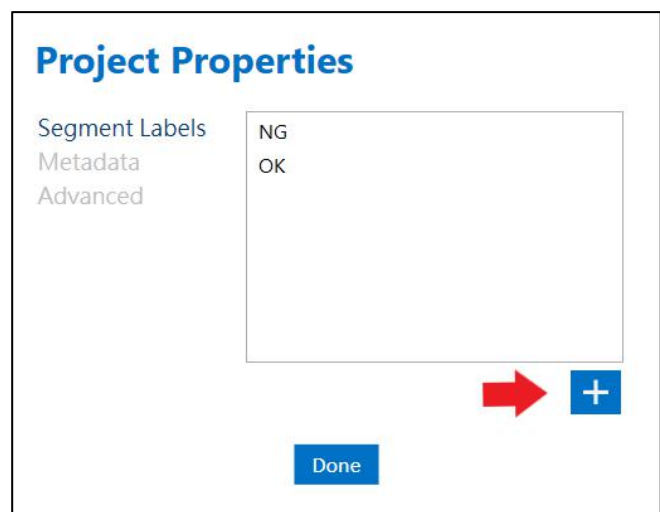
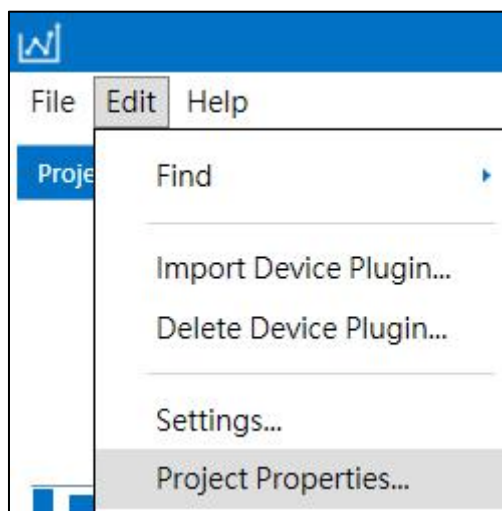


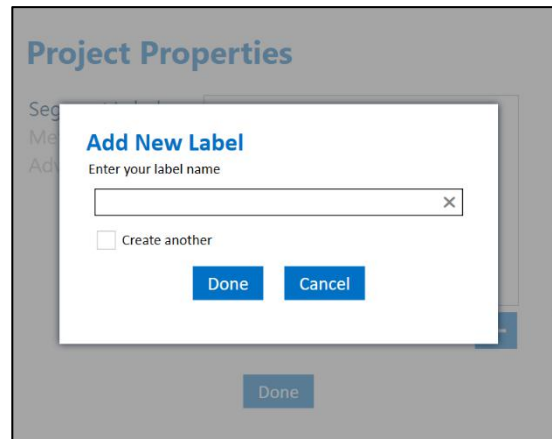
四、模型訓練

1. 點選 Start Recording 按鈕開始蒐集資料，完成後按下 Stop Recording，以.csv 檔案儲存，檔名在附圖中的對話框 TEXT 設定。

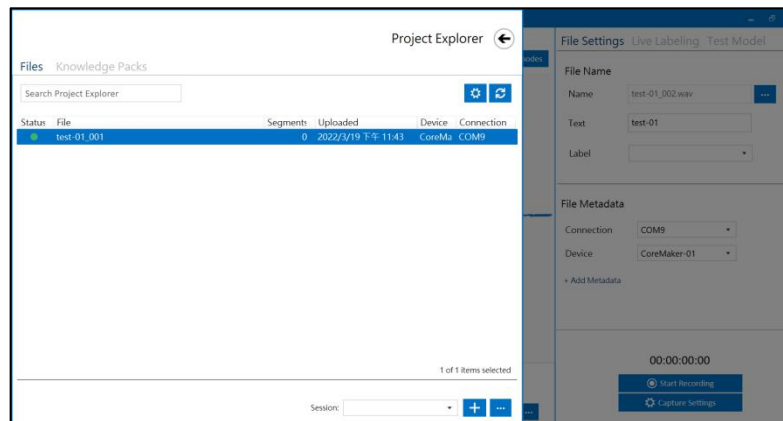


2. 建立標籤

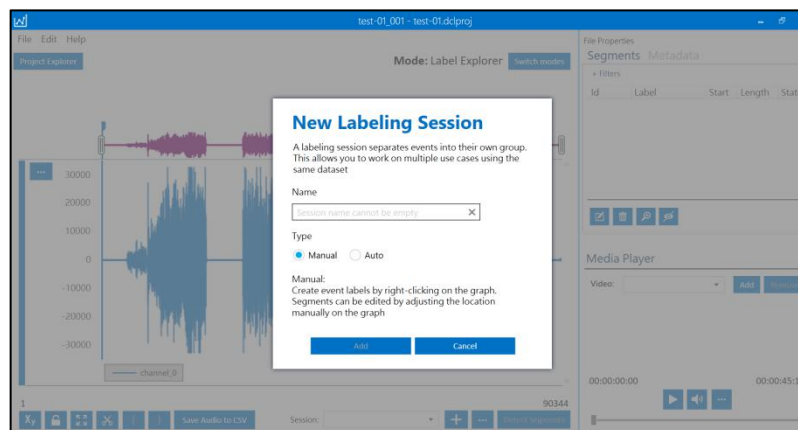




3. 點選 **Project Explorer** 按鈕，雙擊檔案名稱開啟，開啟後在波形圖上點擊滑鼠右鍵，此時會出現紅色及藍色的垂直長線，以滑鼠拖動這兩條線來設定標記範圍，點選編輯按鈕編輯標記。標記完成後，點選 **Save Changes** 按鈕儲存。



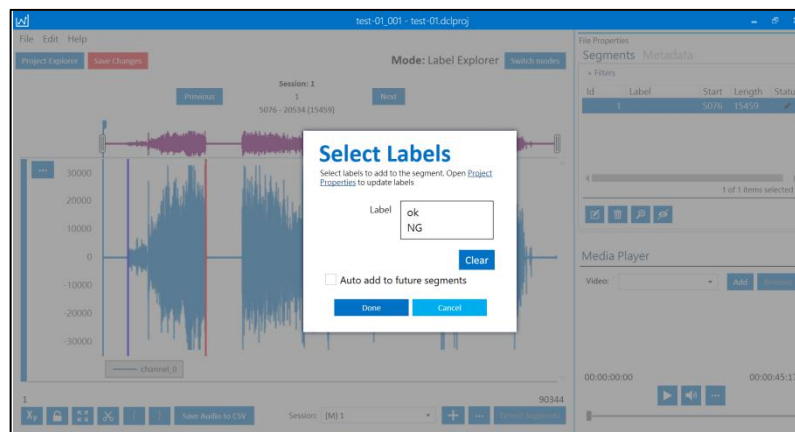
- 在波形圖上點擊滑鼠右鍵，此時會出現以下附圖，建立 **Labeling session** 名稱，Type 選擇 **Manual**。



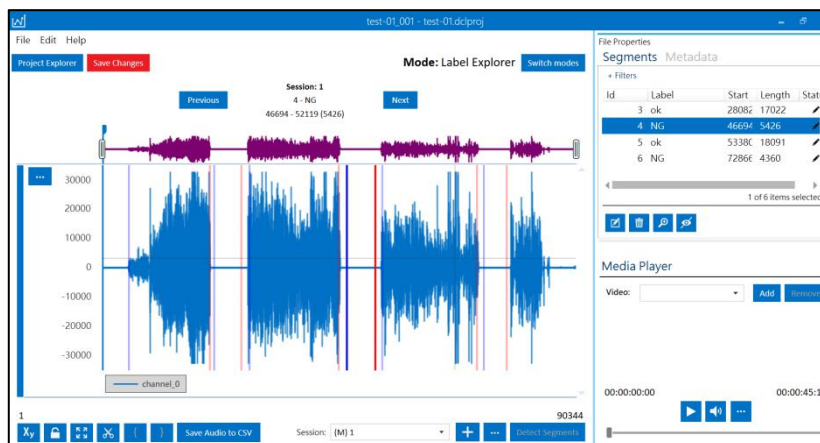
- 建立後會出現紅色及藍色的垂直長線，以滑鼠拖動這兩條線來設定標記範圍，點選編輯按鈕編輯標記。



- 按下編輯標籤後，選擇當初建立標籤時的種類 (ex: 有收集到資訊=ok、無收集資料=NG)。

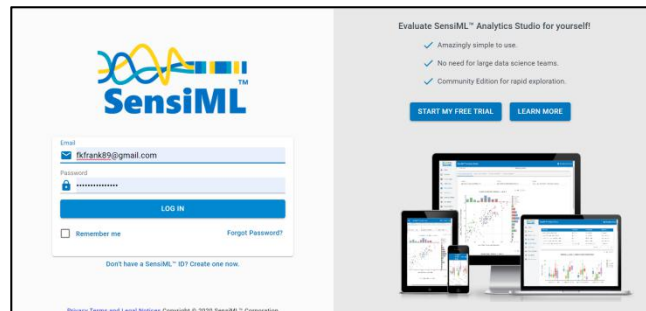


- 標記完成後，點選 Save Changes 按鈕儲存。



五、建立雲端檔案

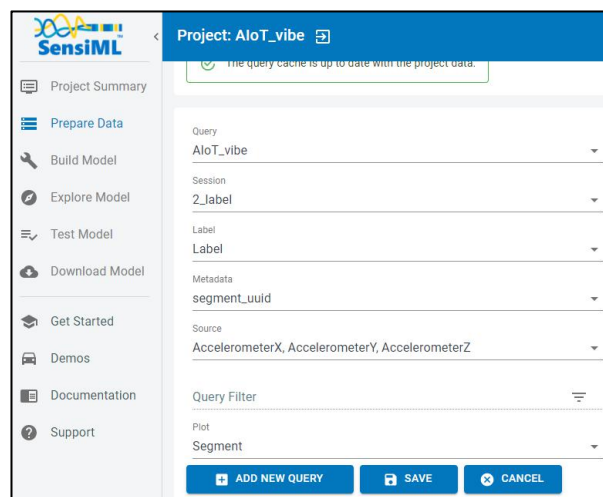
- 進入 SensiML Analytics Studio 網頁 (<https://app.sensiml.cloud/auth/login/>)，登入帳號



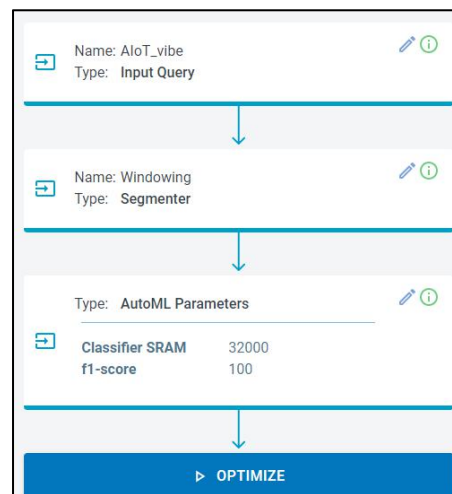
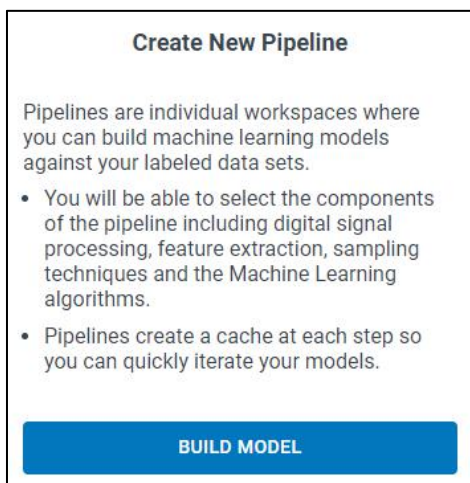
- 打開 project

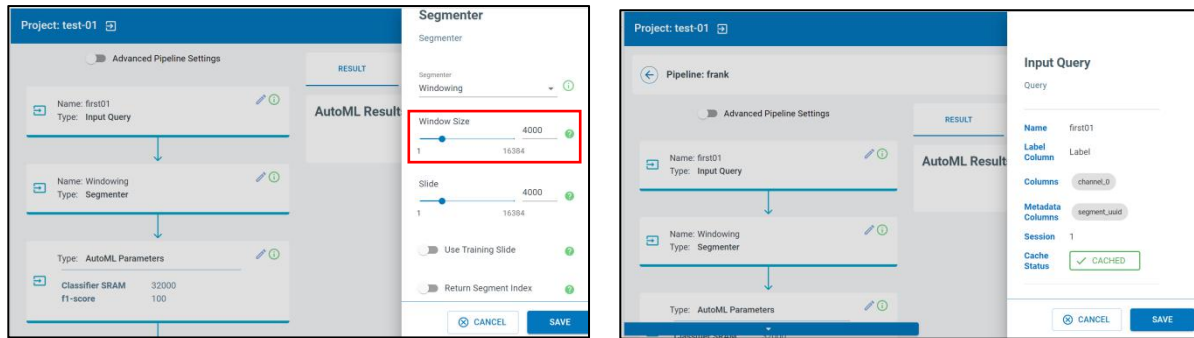


- 在 Prepare Data 頁面中建立新的 Query。



- 在 Build Model 頁面中建立模型 (關於建立模型的詳細設定，請參考 <https://sensiml.com/documentation/guides/getting-started/index.html>)。





✧ 上圖雲端中的 window Size 數值不能超過 Local 端 (Date Capture Lab) Segments-Length 之值。

File Properties

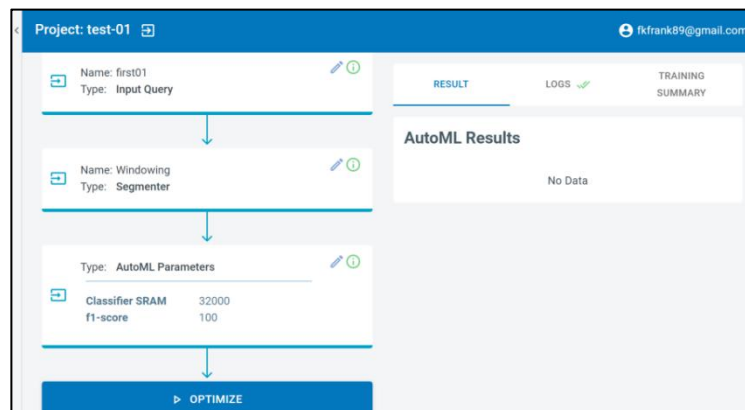
Segments Metadata

+ Filters

	Label	Start	Length	Status
1	ok	5076	15459	●
2	NG	21374	5179	●
3	ok	28082	17022	●
4	NG	46694	5426	●
5	ok	53380	18091	●

1 of 6 items selected

- 按下 OPTIMIZE (下圖為成功執行)



Project: test-01

AutoML Results

MODEL NAME	ACCURACY	CLASSIFIER SIZE(B)	NUM. FEATURES	SENSITIVITY
frank_rank_0	100	158	2	100
frank_rank_1	100	160	3	100
frank_rank_2	100	162	4	100
frank_rank_3	100	162	4	100
frank_rank_4	100	162	4	100

六、更新韌體

- 模型建立完成後，在 Download Model 頁面中，選擇合適的設定，點選 DOWNLOAD 下載模型。

Download Knowledge Pack

Target Device Options:

HW Platform
ARM GCC Generic

Processor
ARM Cortex M4

Float Options
Soft FP

Compiler
GNU Arm Embedded (none-eabi) 10.3.1

Format
Library

Data Source
AIoT_vibe

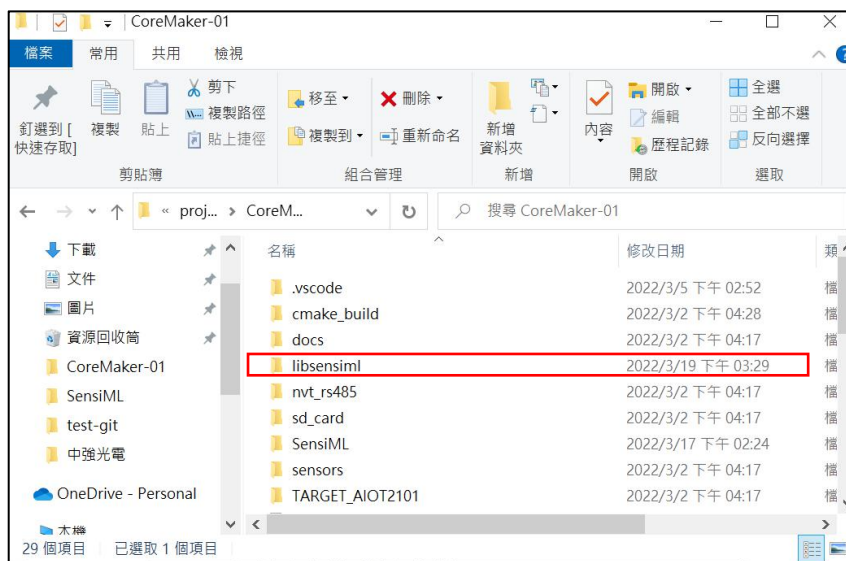
Application
SensiML AI Model Runner

Output
Serial

[Advanced Settings](#)

DOWNLOAD

- 解壓縮下載的檔案，用裡面的 libsensiml 資料夾覆蓋 CoreMaker 的韌體原始碼的資料夾下的 libsensiml 資料夾。



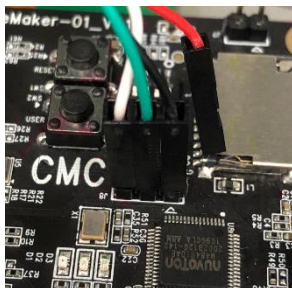
- 在 CoreMaker-01 的韌體原始碼的資料夾下，執行 `mbed-tools compile -m AIOT2101 -t GCC_ARM` 編譯 (成功編譯如下圖)。

```

命令提示字元
68 | if (!(expr)) {
    | ^~~~~
In file included from c:\project\coremaker-01\mbed-os\drivers\include\drivers\spi.h:40,
    | from C:/project/CoreMaker-01/mbed-os/mbed.h:66,
    | from C:/project/CoreMaker-01/sd_card/NuSDBlockDevice.h:23,
    | from C:/project/CoreMaker-01/sd_card/NuSDBlockDevice.cpp:20:
C:/project/CoreMaker-01/mbed-os/platform/include/platform/CThunk.h: In instantiation of 'CThunk<T>::CThunk(T*, CThunk<T>::CCallbackSimple) [with T = NuSDBlockDevice; CThunk<T>::CCallbackSimple = void (NuSDBlockDevice::*)()]:
C:/project/CoreMaker-01/sd_card/NuSDBlockDevice.cpp:143:55: required from here
C:/project/CoreMaker-01/mbed-os/platform/include/platform/CThunk.h:70:24: warning: cast between incompatible pointer to
member types from 'CThunk<NuSDBlockDevice>::CCallbackSimple' {aka 'void (NuSDBlockDevice::*)()'} to 'CThunk<NuSDBlockDev
ice>::CCallback' {aka 'void (NuSDBlockDevice::*)(void*)'} [-Wcast-function-type]
70 |     init(instance, (CCallback)callback, NULL);
    |                   ~~~~~^~~~~~
[809/809] Linking CXX executable AIOT_2101.elf
-- built: C:/project/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.bin
-- built: C:/project/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.hex
C:\project\CoreMaker-01>

```

- 進入燒錄模式 (可參考第參章)
- 使用 UART to USB 轉接線，連接 CoreMaker J8 區針腳



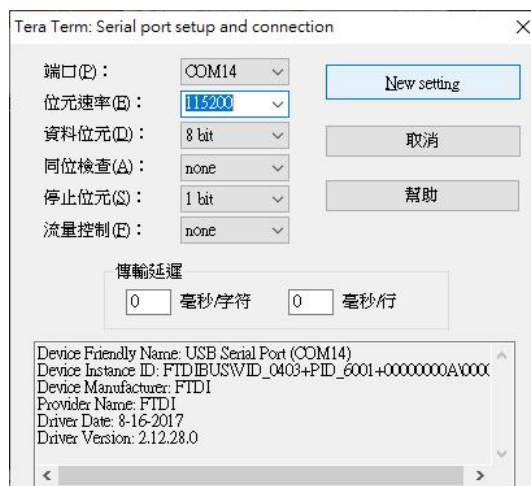
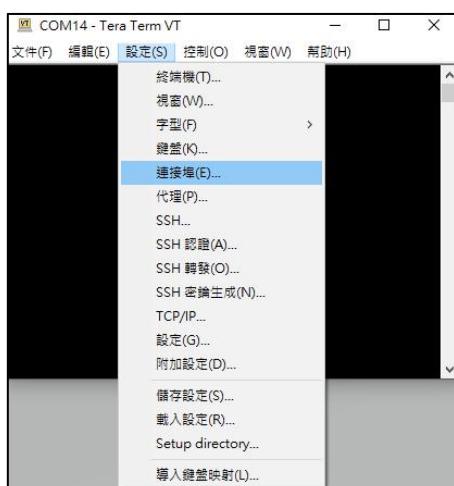
✧ 連接方式

UART GND -> CoreMaker-01 GND
 UART TX -> CoreMaker-01 M_RX
 UART RX -> CoreMaker-01 M_TX

- 開啟連線程式，如 Tera Term (Windows)、串口調試助手 (Windows)、Minicom (Linux) 等，以下將以 Tera Term 作為範例
- 選擇連接埠



- 設定 Baud Rate 為 115200



- 上電後，用 UART 連接 Debug Pins 查看模型辨識結果。

```
SensorHub: sensor0 spu0410 on  
AI classification result: 8  
AI classification result: 1  
AI classification result: 1  
AI classification result: 1
```

Document History

[illegible]