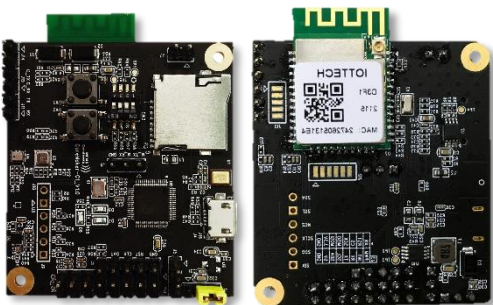


CoreMaker-01



聲控居家控制

指導單位：
 IDB
INDUSTRIAL DEVELOPMENT BANK
MINISTRY OF ECONOMIC AFFAIRS
經濟部工業局

主辦單位：
 財團法人資訊工業策進會
INSTITUTE FOR INFORMATION INDUSTRY

 物聯網智造基地
IOT SERVICE HUB

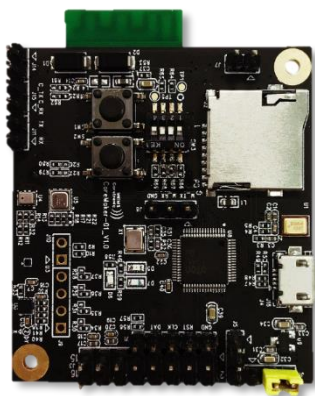
合作單位：設計者：章育銘

程式位置：<https://github.com/wildman8606/Coremaker01withIoT>

前言

- 本案將2款國產IC開發套件DSI2598+或DSI5168(二擇一)、CoreMaker-01結合，設計可由遠端連網控制之智慧居家應用示範。
- 此案例可延伸應用於智慧遠端手勢遙控家電、電燈、大門等，例如：關閉窗簾，開電視，開冷氣28度等指令。

CoreMaker-01

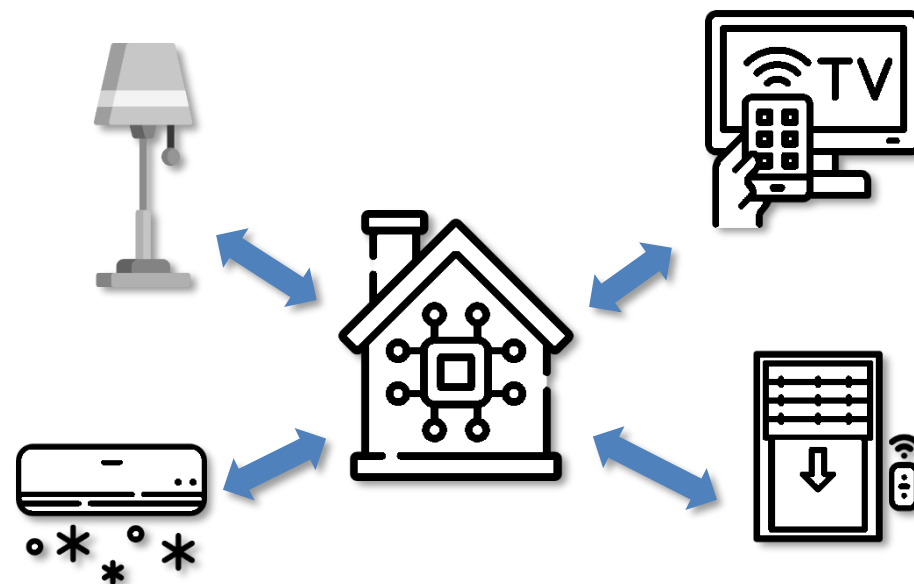


DSI 5168



OR

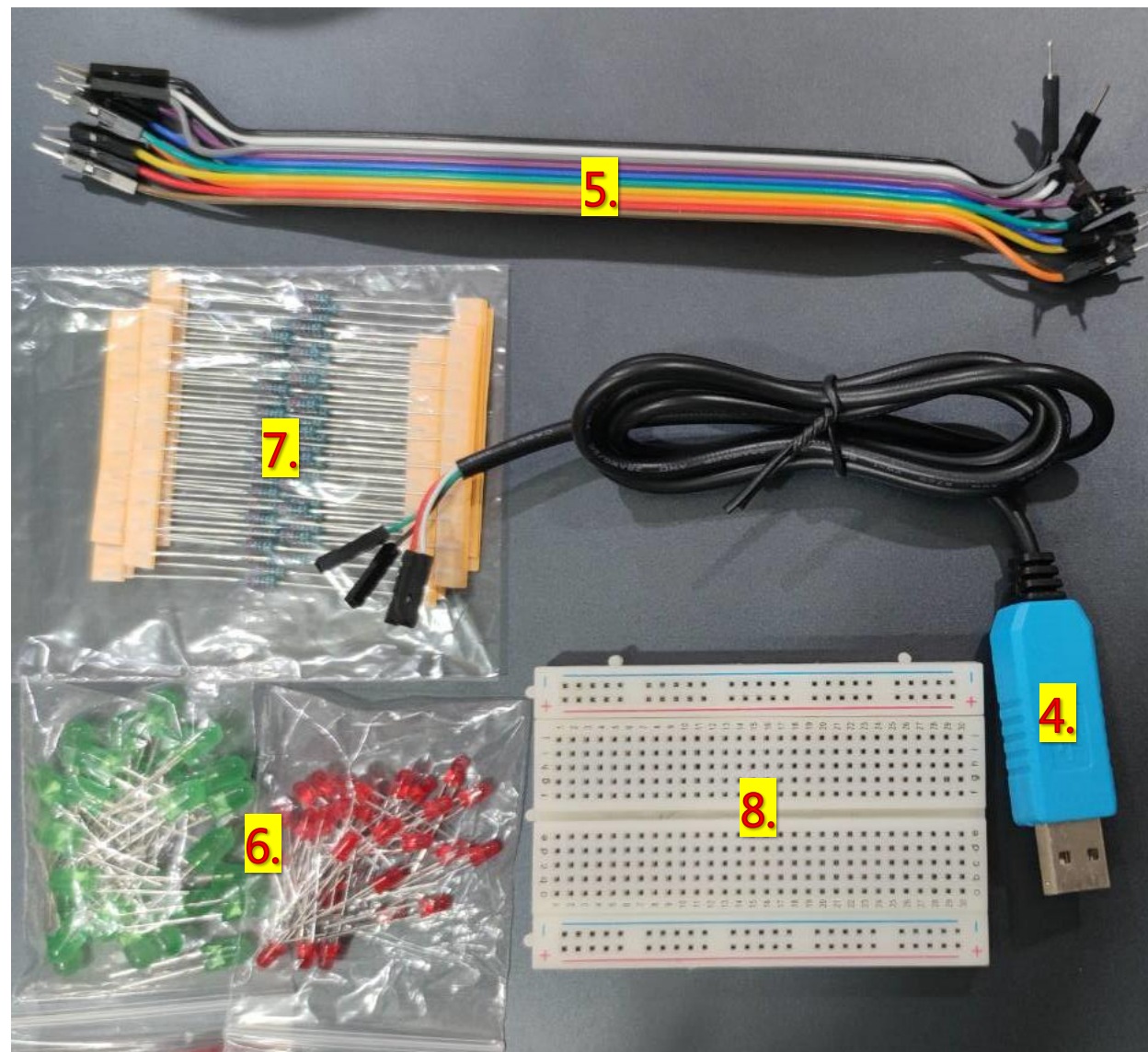
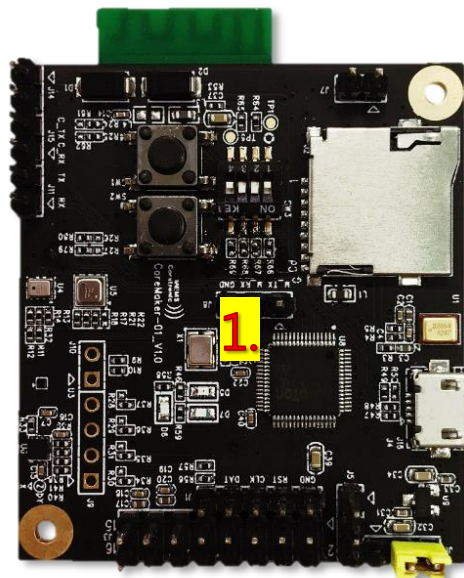
DSI 2598+



材料準備

準備材料:

1. Coremaker-01
2. DSI 5168
3. DSI 2598+
4. USB轉TTL(PL2303)
5. 杜邦線
6. LED*2色
7. 220Ω電阻
8. 麵包板
9. 電腦
10. MicroUSB傳輸線



CoreMaker-01 晶片介紹

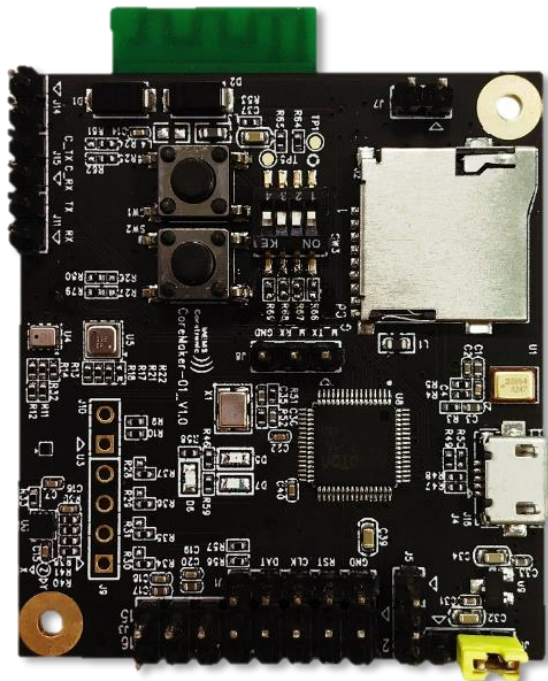
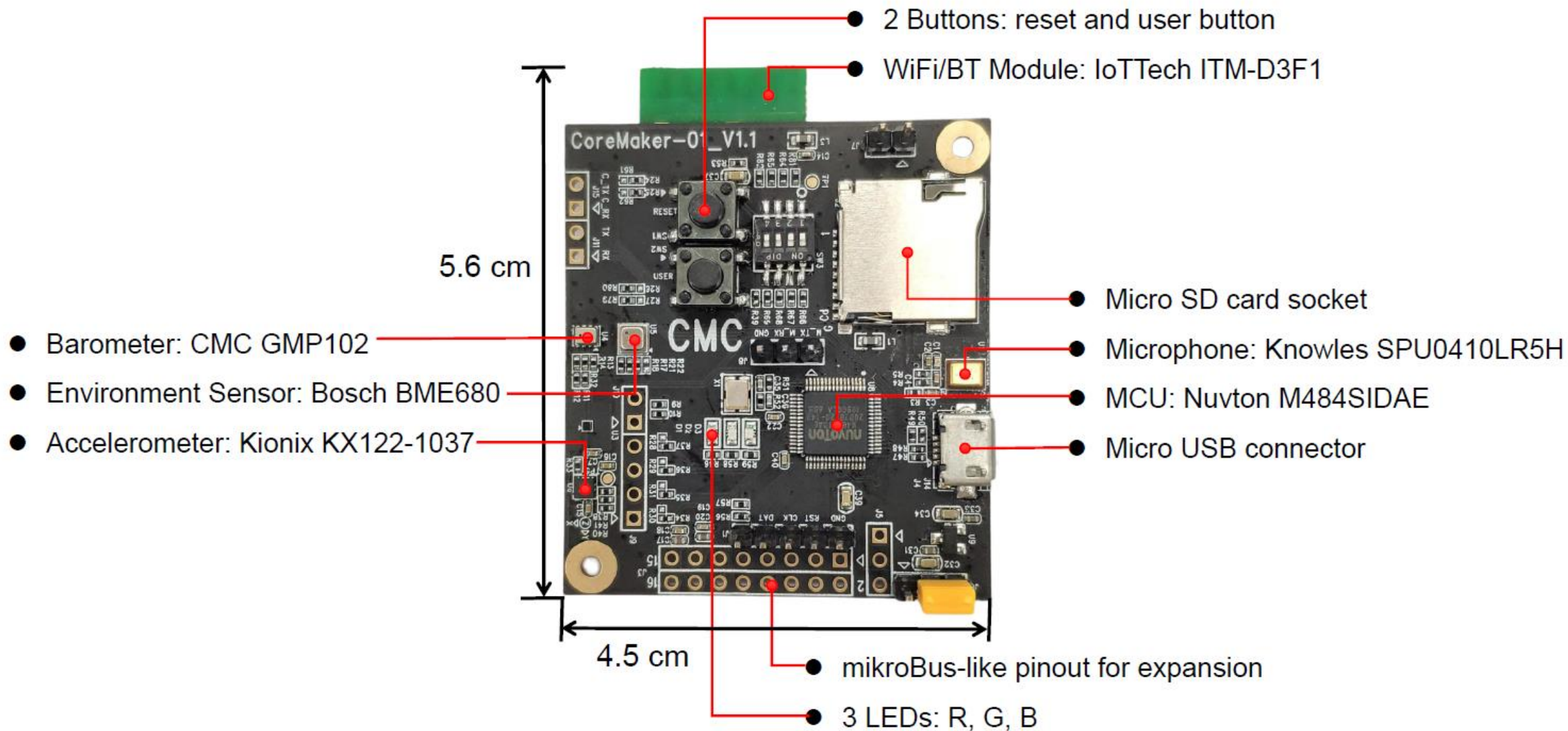


圖1、CoreMaker-01 晶片

尺寸	55mm x 45mm
特色	採用 新唐 /M484SIDAE CoreMaker-01與您一起開拓AIoT進化路
介紹	<p>晶片原廠: 中光電智能感測Coretronicmems 官網: https://www.coretronicmems.com/ CoreMaker-01使用新唐M484，以Arm® Cortex®-M4F為核心，帶有DSP指令集的高效能低功耗微控制器，支援可程設 UART / SPI / I²C。開發人員可運用ARM Mbed物聯網裝置平台，一款結合網際網路協定、資安與標準化管理的單一整合式解決方案，專門針對耗能與定價特別敏感的物聯網裝置所設計。透過多維度感測模組，以Device AI概念打造高性價比AIoT方案，客戶可自行開發或委由CMC協助開發AI模型，燒錄至CoreMaker-01 MCU，即可完成一款具AI功能之裝置。</p>

CoreMaker-01 晶片介紹



DSI5168 晶片介紹

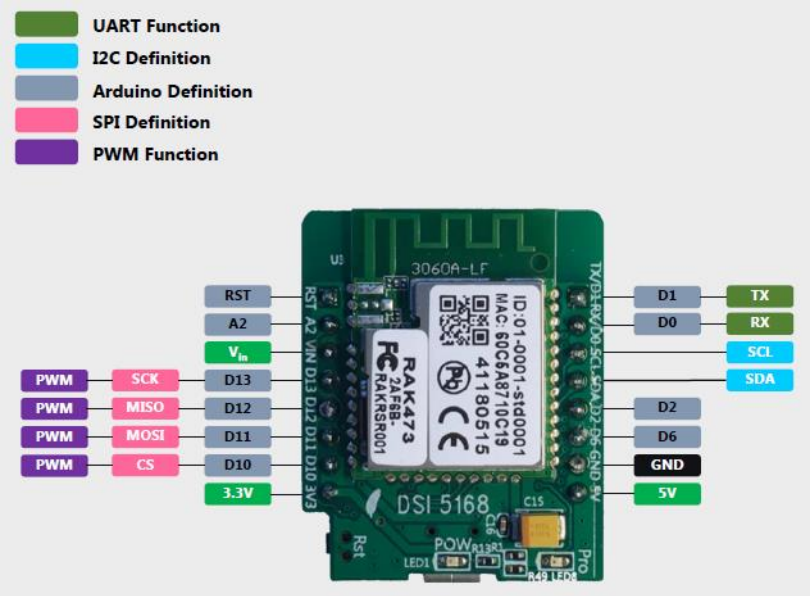


圖1、DSI 5168 晶片

尺寸	35mm x 27mm
特色	採用Ameba RTL8711AM，您的IoT產品開發 “最佳好幫手”
介紹	DSI5168對於maker來說是一項非常方便的開發工具，它包括了RTL8711AM無線模組以及集成配套式燒錄工具，完整兼容Arduino開發特性，並整合高性能MCU、Wi-Fi、Ethernet及豐富的外圍設備，提供標準化的Arduino庫，協助開發商與創客開發一套完整的物聯網解決方案。
產品案例	<ul style="list-style-type: none"> • O-take高階虛擬實境控制手套 • 無線物聯網環控系統 • 雞隻聲紋呼吸道疾病檢測儀器

DSI2598+ 晶片介紹

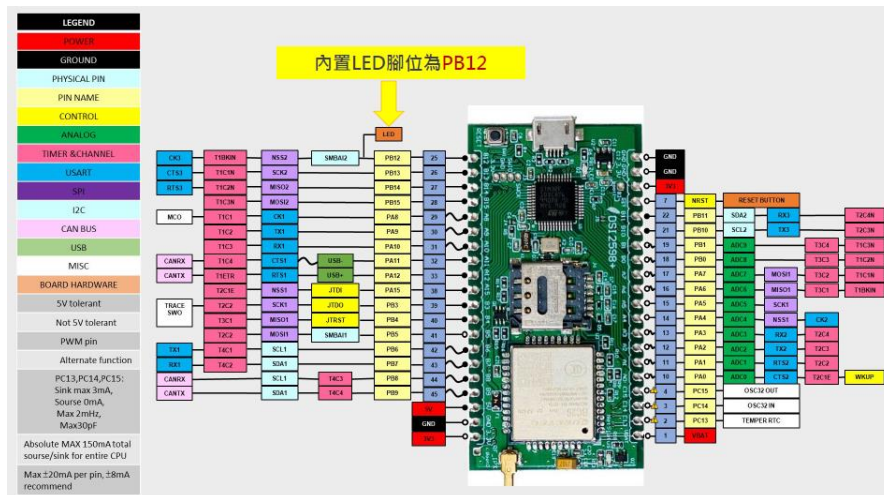


圖1、DSI 2598+ 晶片

1. STM32duino bootloader- USB燒錄
2. Serial - FTDI工具
3. STLink - STLink工具

尺寸	60mm x 30mm
特色	全台首款小型 Arduino NB-IoT開發板
介紹	DSI2598+使用 聯發科 技NB-IoT晶片-MT2625模組，搭配STM32 F103核心，有著PWM、I2C、SPI、ADC、UART等腳位功能，簡單但完整且有極佳運算能力，可讓使用者無縫接軌大部分Arduino程式庫，進行各項功能程式開發，尺寸僅6.0x3.0cm，是國內第一款小型NB-IoT開發板。
產品案例	<ul style="list-style-type: none"> • O-Care 共享機車自動消毒劑 • 發電機/空壓機/活氧機遠端監控裝置 • 社區型快扣式淹水預警系統 • 瓦斯鋼瓶液位偵測裝置 • 農業大數據擷取平台 • SimuGro水耕系統 • 家禽健康監控儀器

DSI2598+ 晶片介紹

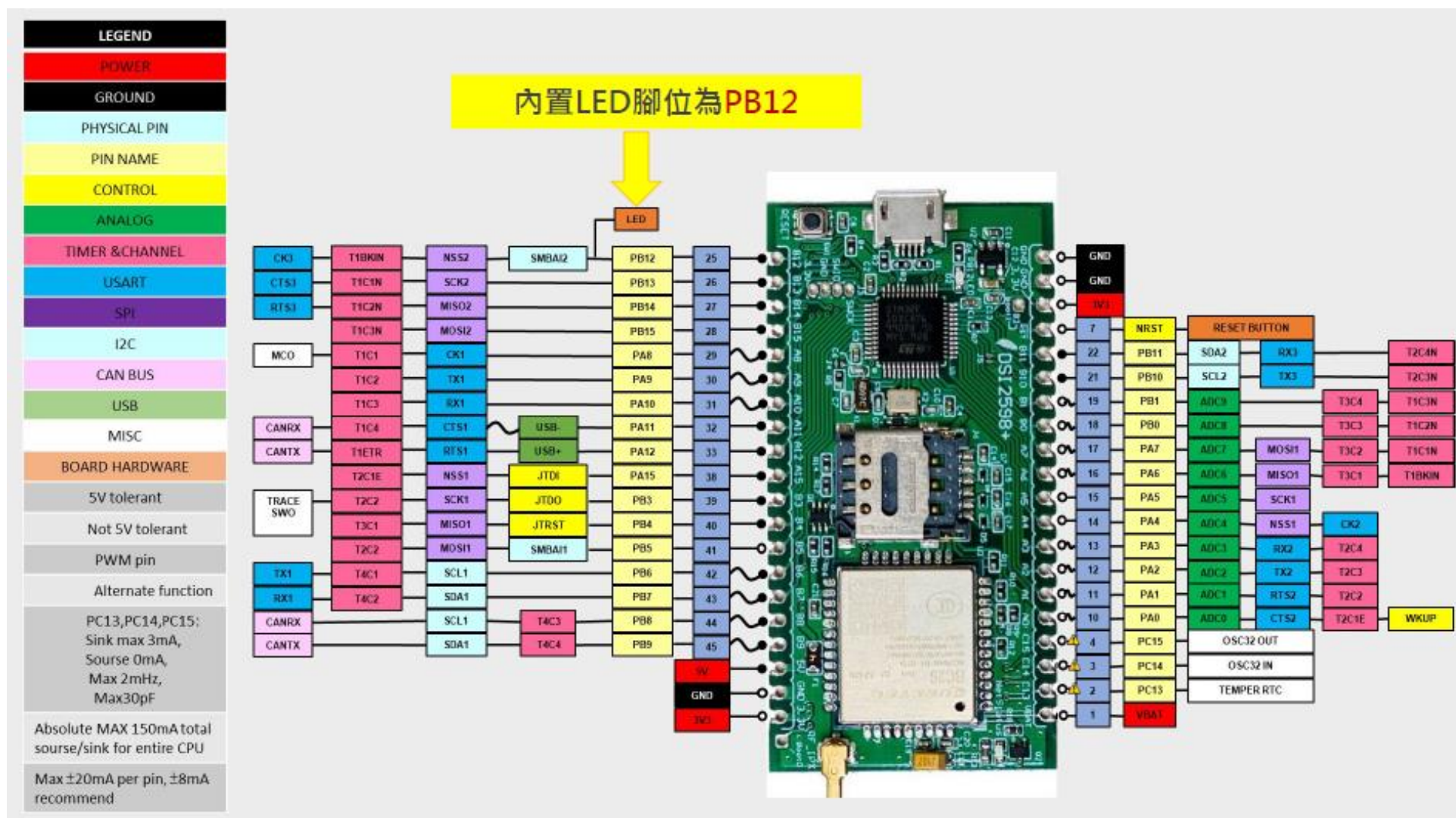


圖1 、DSI 2598+ 晶片

開始前需安裝軟體



20分鐘

安裝說明後置於附錄一

Arduino

Notepad++

Python

Tera Term



必需性:

必要

可替換

必要

可替換



說明:

國產晶片用
Arduino IDE
燒入、編譯
方便上手

慣用程式編輯器
都可，這較小、
輕量、好用。

晶片編譯需要

UART顯示結果，
可擇自己喜歡。



開始前安裝軟體二



25分鐘

CMake



git



**GNU Arm
Embedded
Toolchain**

GNU Arm
Embedded
Toolchain

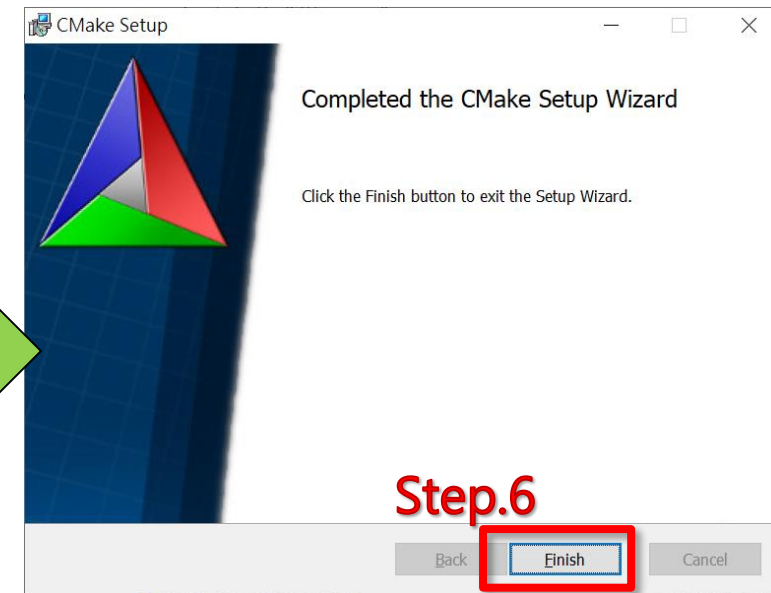
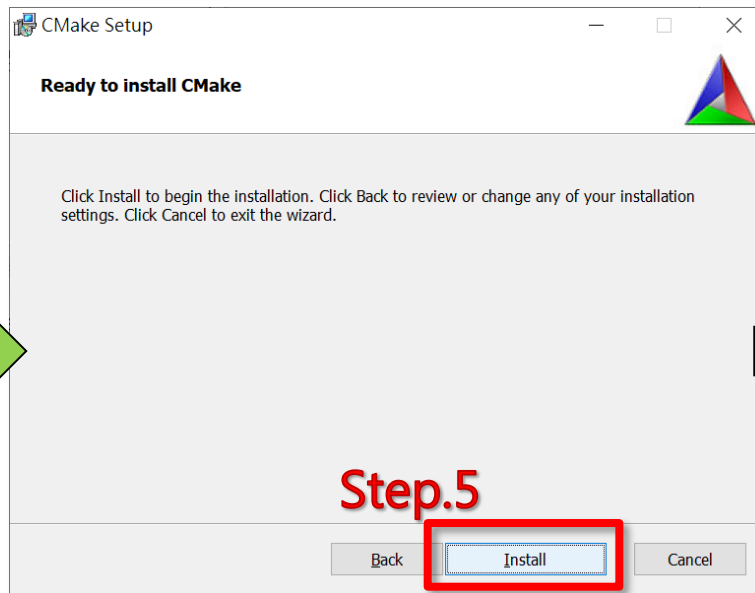
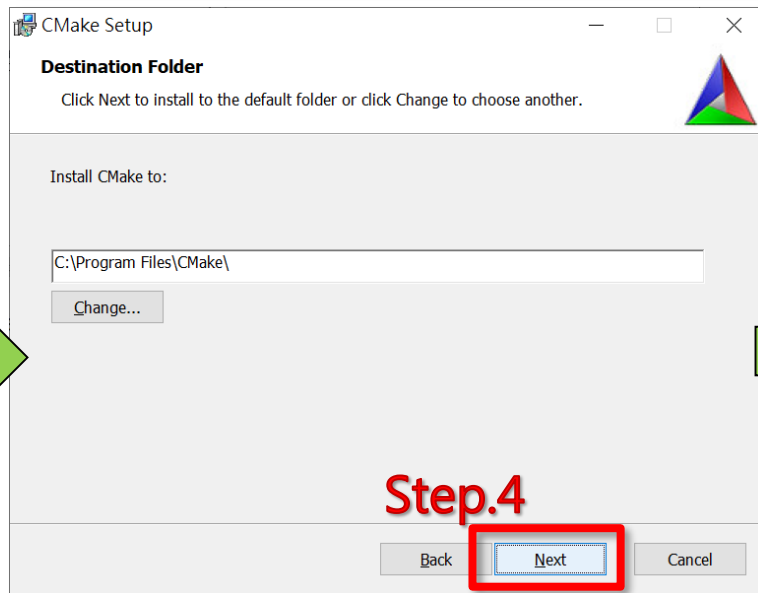
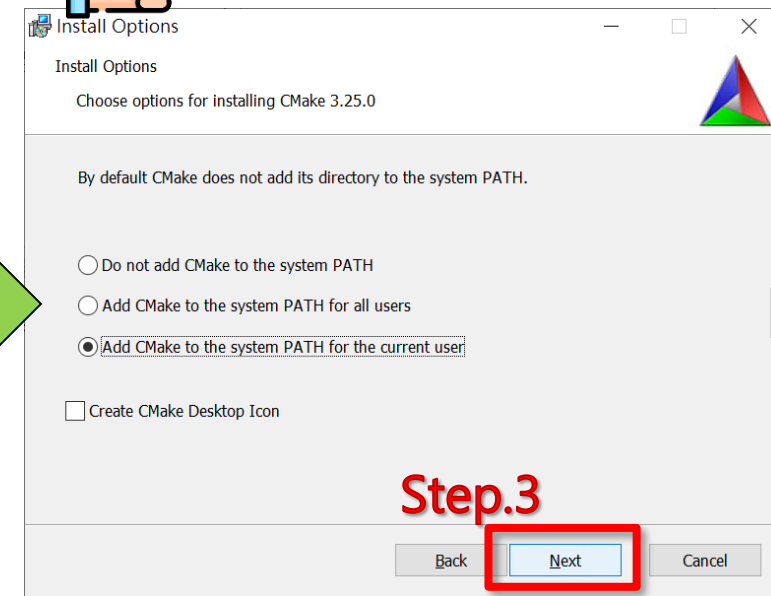
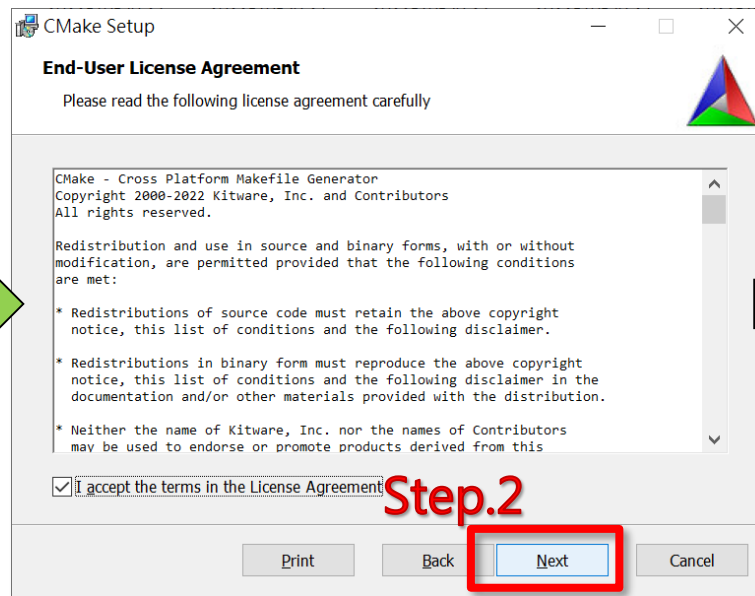
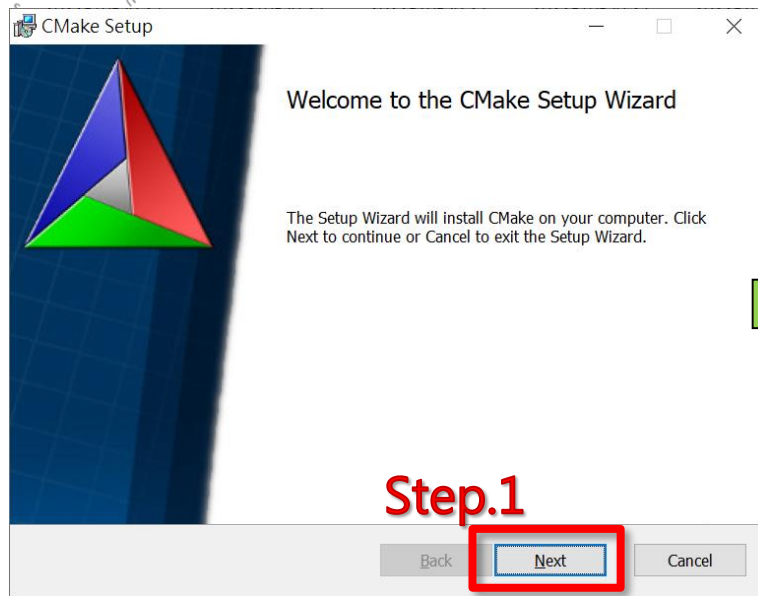
**SensiML Data
Capture Lab**



軟體安裝引導(CMake)



<https://cmake.org/download/>

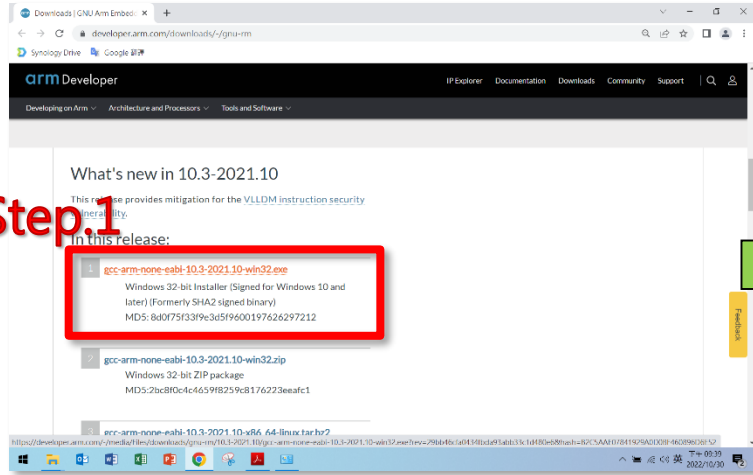


軟體安裝引導(GNU Arm Embedded Toolchain)



<https://developer.arm.com/downloads/-/gnu-rm/>

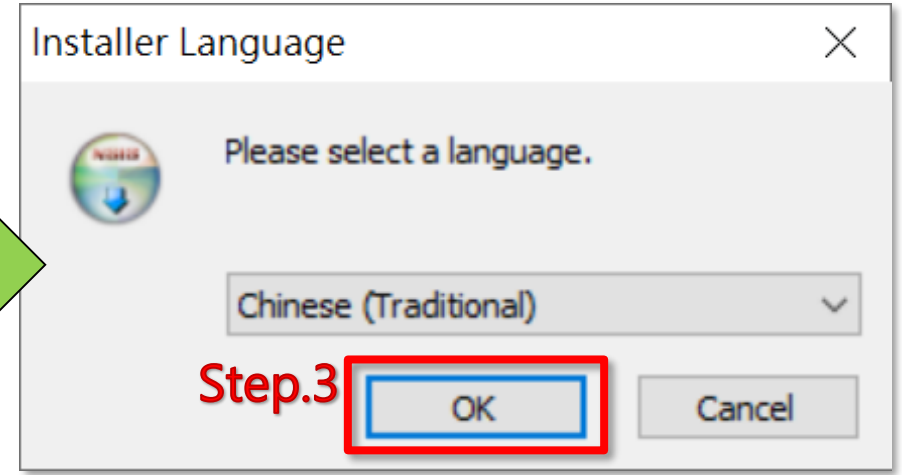
Step.1



Step.2



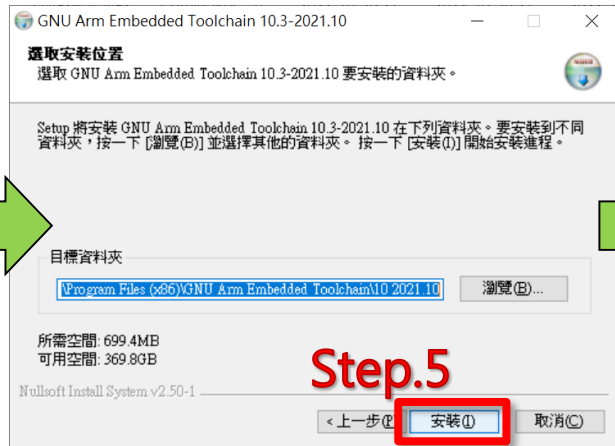
Step.3



Step.4



Step.5



Step.7



軟體安裝引導(git)

如有其他慣用git下載方法可以免安裝這個



<https://git-scm.com/downloads>

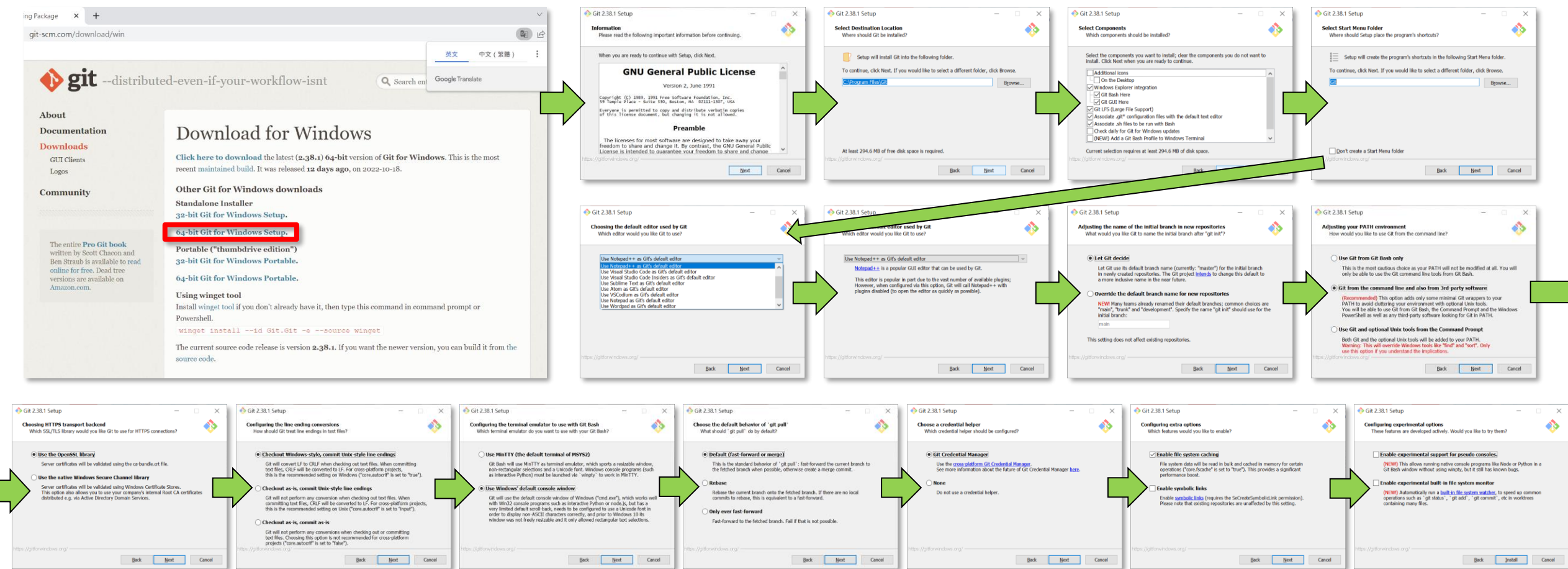


圖 1~16.git安裝引導

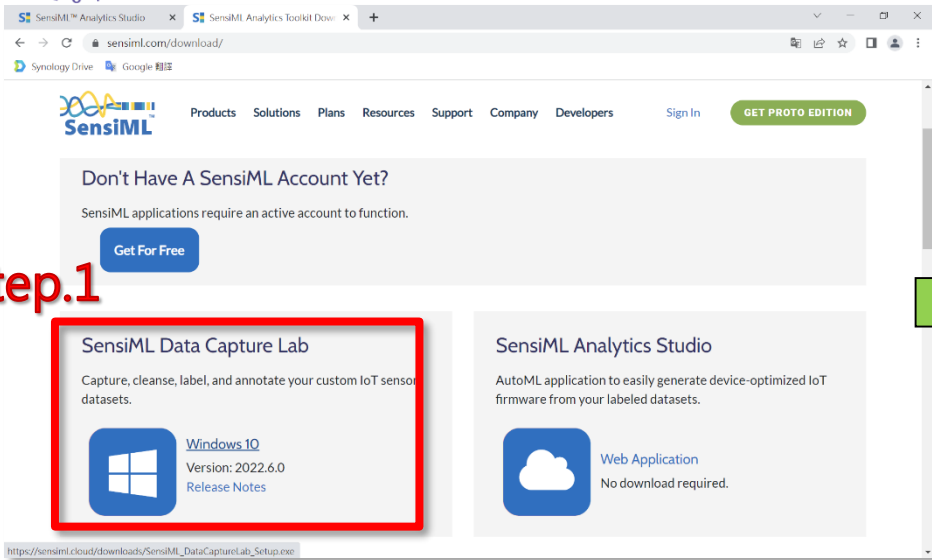


軟體安裝引導(SensiML Data Capture Lab)

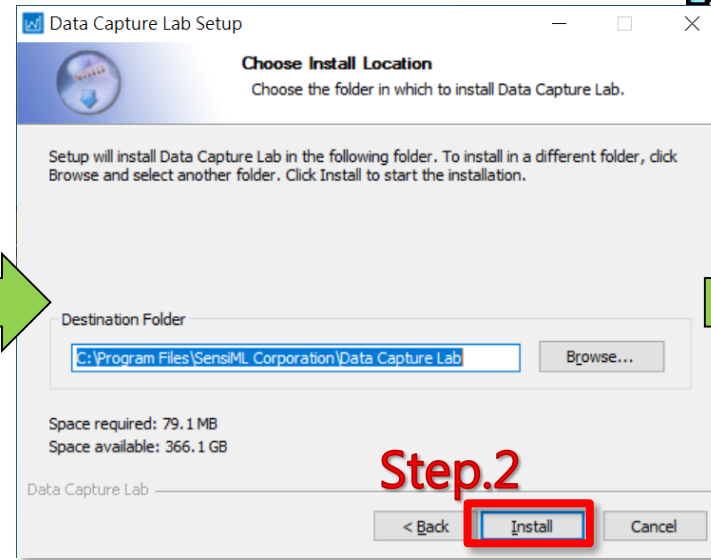


<https://sensiml.com/download/>

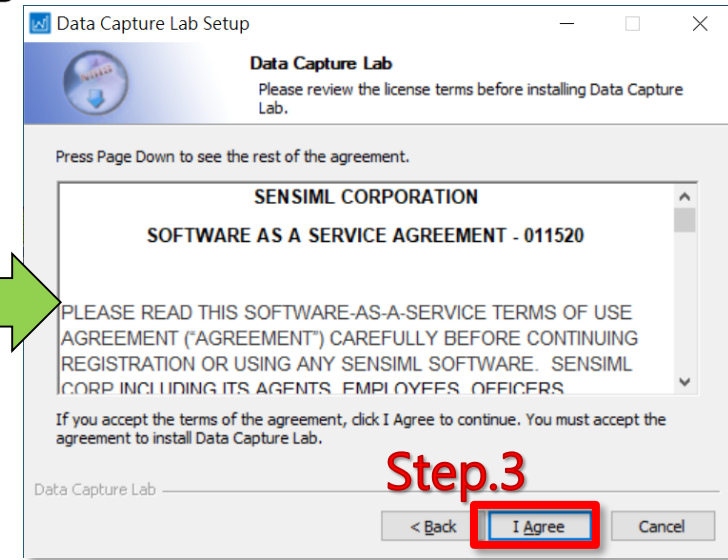
Step.1



Step.2



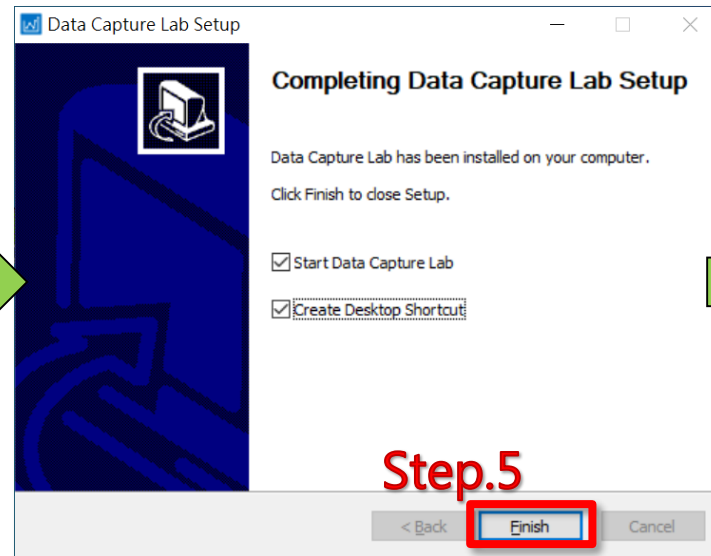
Step.3



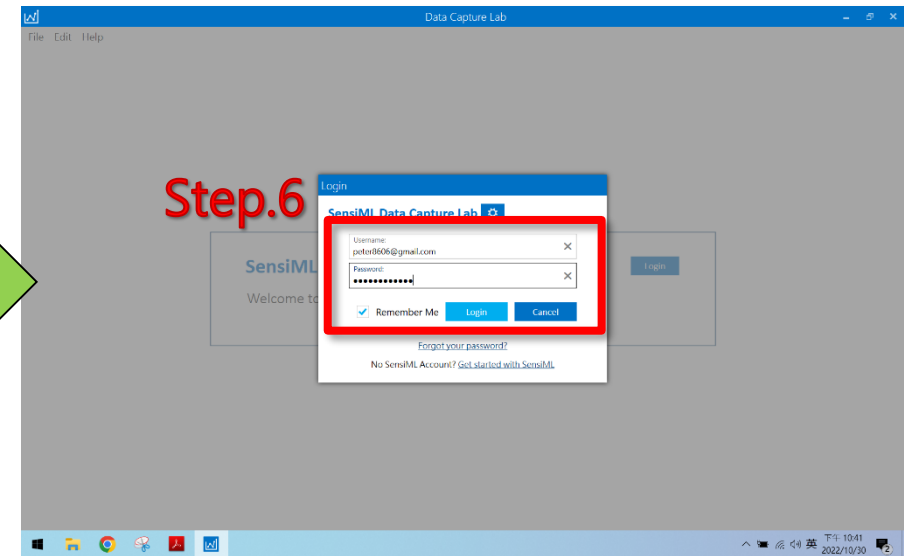
Step.4



Step.5



Step.6





製作Coremaker01映像檔所需準備



25分鐘

安裝Ninja與Mbed CLI 2套件

```
系統管理員: C:\Windows\System32\cmd.exe - python -m pip install mbed-tools
Microsoft Windows [版本 10.0.19044.2130]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\mlchai\Downloads>color a

C:\Users\mlchai\Downloads>python -m pip install ninja
Collecting ninja
  Downloading ninja-1.10.2.4-py2.py3-none-win_amd64.whl (293 kB)
----- 293.9/293.9 kB 1.4 MB/s eta 0:00:00
Installing collected packages: ninja
Successfully installed ninja-1.10.2.4

C:\Users\mlchai\Downloads>python -m pip install mbed-tools
Collecting mbed-tools
  Downloading mbed_tools-7.58.0-py3-none-any.whl (127 kB)
----- 127.3/127.3 kB 836.3 kB/s eta 0:00:00
Collecting python-dotenv
  Downloading python_dotenv-0.21.0-py3-none-any.whl (18 kB)
Collecting Click<9,>=7.1
```

Step1. 輸入指令安裝

```
> python -m pip install update
> python -m pip install ninja
> python -m pip install mbed-tools
```

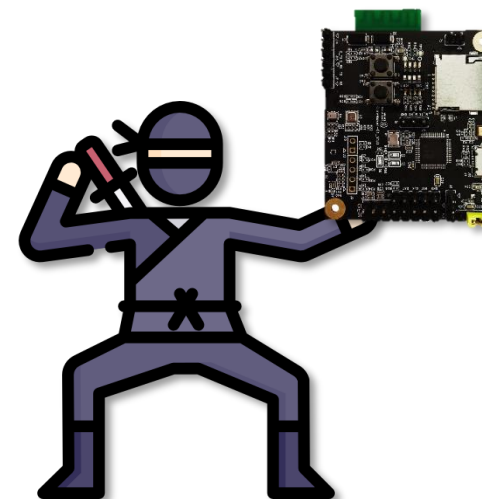


圖 1.輸入指令安裝編譯所需套件

製作Coremaker01映像檔所需準備

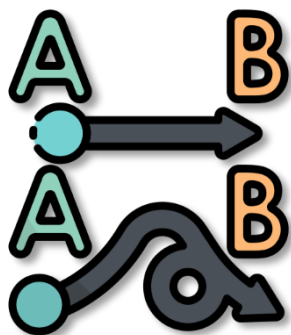


25分鐘

額外需要東西

Step2. 輸入指令安裝

```
> pip install prettytable
> pip install future
> pip install intelhex
```



講師帮助大家
排除問題

```
系統管理員: C:\Windows\System32\cmd.exe - mbed-tools compile -m AIOT2101 -t GCC_ARM
Microsoft Windows [版本 10.0.19044.2130]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

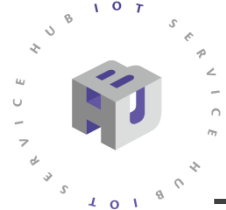
C:\Users\mlchai\Desktop\CoreMaker-01>pip install prettytable
Collecting prettytable
  Downloading prettytable-3.5.0-py3-none-any.whl (26 kB)
Collecting wcwidth
  Downloading wcwidth-0.2.5-py2.py3-none-any.whl (30 kB)
Installing collected packages: wcwidth, prettytable
Successfully installed prettytable-3.5.0 wcwidth-0.2.5

C:\Users\mlchai\Desktop\CoreMaker-01>pip install future
Collecting future
  Downloading future-0.18.2.tar.gz (829 kB)
----- 829.2/829.2 kB 2.4 MB/s eta 0:00:00
Preparing metadata (setup.py) ... done
Installing collected packages: future
  DEPRECATION: future is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
  Running setup.py install for future ... done
Successfully installed future-0.18.2

C:\Users\mlchai\Desktop\CoreMaker-01>pip install intelhex
Collecting intelhex
  Downloading intelhex-2.3.0-py2.py3-none-any.whl (50 kB)
----- 50.9/50.9 kB 199.4 kB/s eta 0:00:00
Installing collected packages: intelhex
Successfully installed intelhex-2.3.0

C:\Users\mlchai\Desktop\CoreMaker-01>mbed-tools compile -m AIOT2101 -t GCC_ARM
Configuring project and generating build system...
-- Checking for Python package intelhex -- found
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/mlchai/Desktop/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM
Building Mbed project...
[0/1] Linking CXX executable AIOT_2101.elf
```

圖 2.輸入指令安裝編譯所需套件二



下載Coremaker軟體包



25分鐘

下載Coremaker軟體包

存放路徑不能中文



```
Microsoft Windows [版本 10.0.19044.2130]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\mlchai>color a

C:\Users\mlchai>cd Desktop

C:\Users\mlchai\Desktop>git clone --recurse-submodules https://github.com/CoretronicMEMS/CoreMaker-01.git

Cloning into 'CoreMaker-01'...
remote: Enumerating objects: 625, done.
remote: Counting objects: 100% (123/123), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 625 (delta 85), reused 90 (delta 71), pack-reused 502
65 MiB | 1.92 MiB/s
Receiving objects: 100% (625/625), 23.39 MiB | 858.00 KiB/s, done.
Resolving deltas: 100% (372/372), done.
Submodule 'mbed-os' (https://github.com/CoretronicMEMS/mbed-os.git) registered for path 'mbed-os'
Cloning into 'C:/Users/mlchai/Desktop/CoreMaker-01/mbed-os'...
remote: Enumerating objects: 420174, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 420174 (delta 0), reused 0 (delta 0), pack-reused 420173
Receiving objects: 100% (420174/420174), 466.66 MiB | 4.11 MiB/s, done.
Resolving deltas: 100% (293191/293191), done.
Submodule path 'mbed-os': checked out '875ce5bfca68fd74d22ffec266e542d0ebadc0b6'
```

圖 1.輸入指令下載Coremaker軟體包

```
C:\Users\mlchai\Desktop>
C:\Users\mlchai\Desktop>
C:\Users\mlchai\Desktop>
C:\Users\mlchai\Desktop>cd CoreMaker-01

C:\Users\mlchai\Desktop\CoreMaker-01>dir
磁碟區 C 中的磁碟是 OS
磁碟區序號: 36BC-EB09

C:\Users\mlchai\Desktop\CoreMaker-01 的目錄
2022/10/30 下午 10:09 <DIR> .
2022/10/30 下午 10:09 <DIR> ..
2022/10/30 下午 10:09 44 .gitignore
2022/10/30 下午 10:09 94 .gitmodules
2022/10/30 下午 10:09 1,896 CMakeLists.txt
2022/10/30 下午 10:09 <DIR> CMC_ISP
2022/10/30 下午 10:09 611 custom_targets.json
2022/10/30 下午 10:09 2,786 DebounceIn.cpp
2022/10/30 下午 10:09 1,870 DebounceIn.h
2022/10/30 下午 10:09 <DIR> docs
2022/10/30 下午 10:09 1,751 global.h
2022/10/30 下午 10:09 22,567 jRead.c
2022/10/30 下午 10:09 5,033 jRead.h
2022/10/30 下午 10:09 14,558 jWrite.c
2022/10/30 下午 10:09 7,622 jWrite.h
2022/10/30 下午 10:09 <DIR> libsensiml
2022/10/30 下午 10:09 1,092 LICENSE.txt
2022/10/30 下午 10:09 3,243 lightEffect.hpp
2022/10/30 下午 10:09 5,080 main.cpp
```

圖 2.檢查檔案是否完整

Step1. 輸入指令GIT下載

```
> git clone --recurse-submodules https://github.com/CoretronicMEMS/CoreMaker-01.git
```

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Coremaker軟體包



25分鐘

Coremaker軟體包初次編譯

```

系統管理員: C:\Windows\System32\cmd.exe
Downloading intelhex-2.3.0-py2.py3-none-any.whl (50 kB)
----- 50.9/50.9 kB 199.4 kB/s eta 0:00:00
Installing collected packages: intelhex
Successfully installed intelhex-2.3.0

C:\Users\mlchai\Desktop\CoreMaker-01>mbed-tools compile -m AIOT2101 -t GCC_ARM
Configuring project and generating build system
-- Checking for Python package intelhex -- found
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/mlchai/Desktop/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM
Building Mbed project...
[1/1] Linking CXX executable AIOT_2101_elf
-- built: C:/Users/mlchai/Desktop/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.bin
-- built: C:/Users/mlchai/Desktop/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.hex

```

Module	.text	.data	.bss
DebounceIn.cpp.obj	676(+676)	0(+0)	0(+0)
[fill]	212(+212)	0(+0)	52(+52)
[lib]\c.a	11236(+11236)	2108(+2108)	58(+58)
[lib]\gcc.a	3528(+3528)	0(+0)	0(+0)
[lib]\m.a	3864(+3864)	0(+0)	0(+0)
[lib]\misc	188(+188)	4(+4)	28(+28)
[lib]\sensiml.a	13976(+13976)	104(+104)	4399(+4399)
[lib]\stdc++.a	0(+0)	0(+0)	0(+0)
jWrite.c.obj	630(+630)	0(+0)	0(+0)
main.cpp.obj	1938(+1938)	0(+0)	4804(+4804)
mbed-os\cmsis	8820(+8820)	168(+168)	5953(+5953)
mbed-os\connectivity	200(+200)	0(+0)	16(+16)
mbed-os\drivers	11766(+11766)	0(+0)	166(+166)
mbed-os\hal	1776(+1776)	4(+4)	59(+59)
mbed-os\platform	6280(+6280)	260(+260)	440(+440)
mbed-os\rtos	1096(+1096)	0(+0)	0(+0)
mbed-os\targets	14356(+14356)	920(+920)	321(+321)
sensors\AcousticNode.cpp.obj	638(+638)	0(+0)	0(+0)
sensors\BME680	5060(+5060)	0(+0)	8(+8)
sensors\GMC306.cpp.obj	726(+726)	0(+0)	0(+0)
sensors\GMP102	1508(+1508)	0(+0)	8(+8)
sensors\KX122-1037	854(+854)	0(+0)	0(+0)
sensors\SensorHub.cpp.obj	1988(+1988)	40(+40)	988(+988)
Subtotals	91316(+91316)	3608(+3608)	17300(+17300)
Total Static RAM memory (data + bss): 20908(+20908) bytes			
Total Flash memory (text + data): 94924(+94924) bytes			

```

C:\Users\mlchai\Desktop\CoreMaker-01>_

```

初次編譯需要較長的時間

Step1. 進入資料夾

> cd CoreMaker-01

Step2. 執行mbed工具指令

> mbed-tools compile -m AIOT2101 -t GCC_ARM

Step3. 出現'.bin'檔以及記憶體大小即為成功



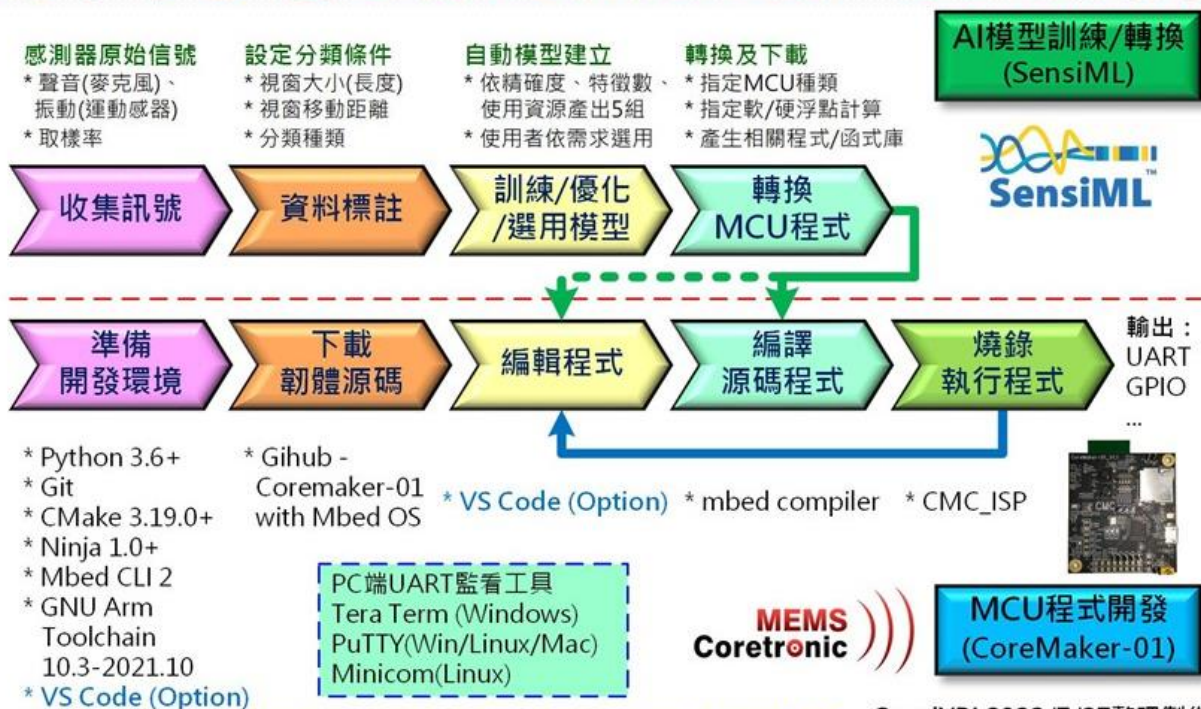
成功編譯

圖 3.編譯成功結果

Coremaker-01 操作流程

由於許博士已於網路上將Coremaker-01操作架構撰寫詳細，這邊引用歐尼克斯實境互動工作室(OmniXRI)資料：

以中光電CoreMaker-01為例



資料來源：<https://omnixri.blogspot.com/2022/07/coremaker-01-sensiml-tinyml.html> OmniXRI 2022/7/27整理製作

2022/10/14

國產IC線上分享會_AIoT與tinyML生態系國際發展趨勢與國產晶片未來方向_OmniXRI_Jack Hsu

14

資料來源:

<http://omnixri.blogspot.com/2022/10/icaiottinyml.html>



因此前兩個科目以執行完畢，於下個科目前須將支線任務做完



Coremaker01 燒入模式

Step1. 使用Micro USB to USB 轉接線，連接CoreMaker 與PC。
 Step2. 同時按下SW1 及SW2，後先放開SW1，再放開SW2，
 Step3. 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。

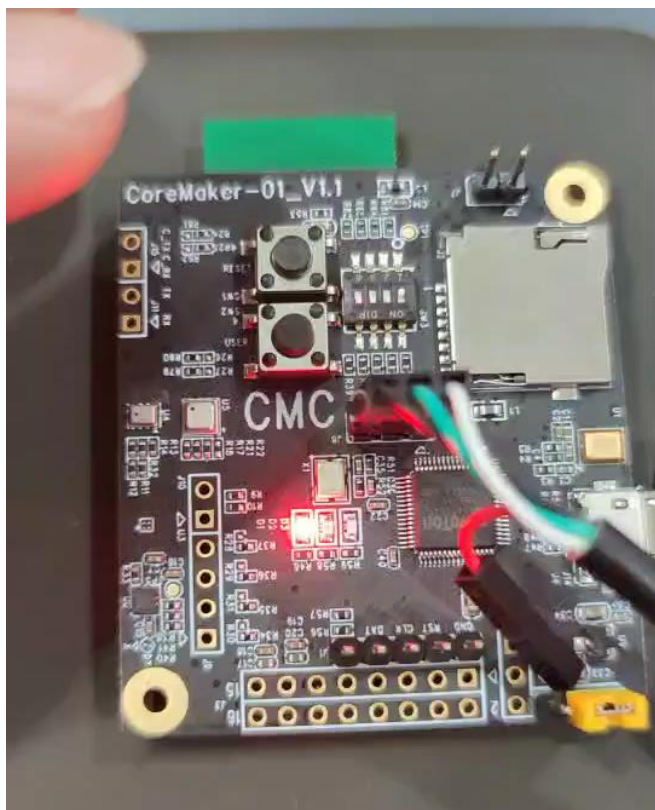


圖 1. 切換燒入模式影片

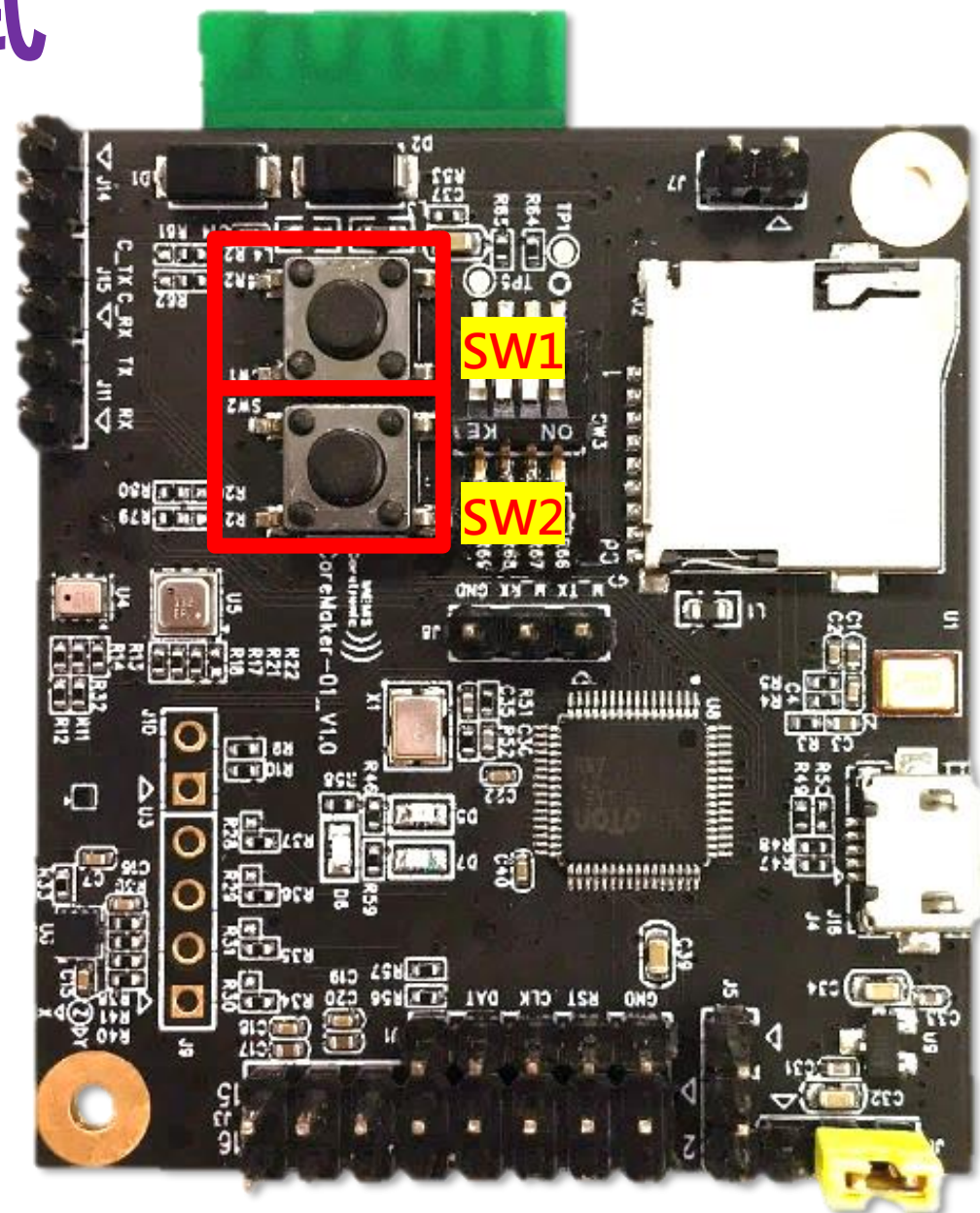


圖 2. Coremaker01 按鈕位置

Coremaker01 一般模式

SW1: reset button
SW2: user button

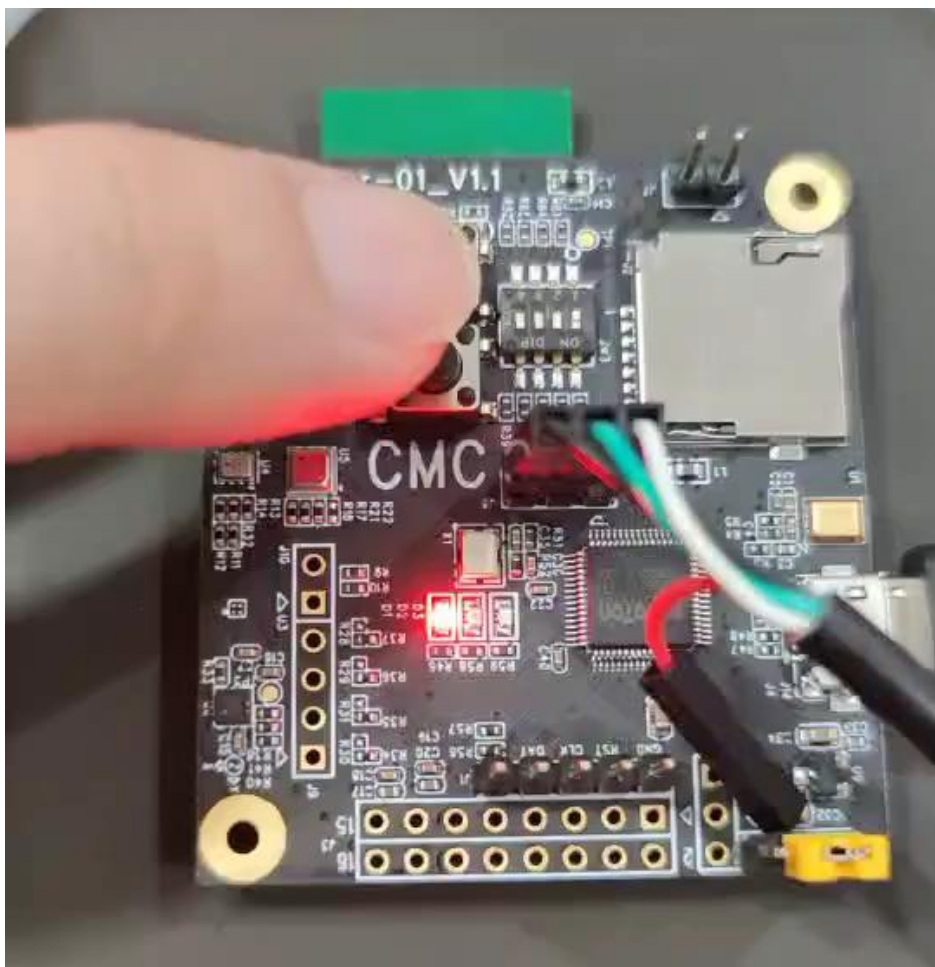


圖 1. 切換一般模式影片

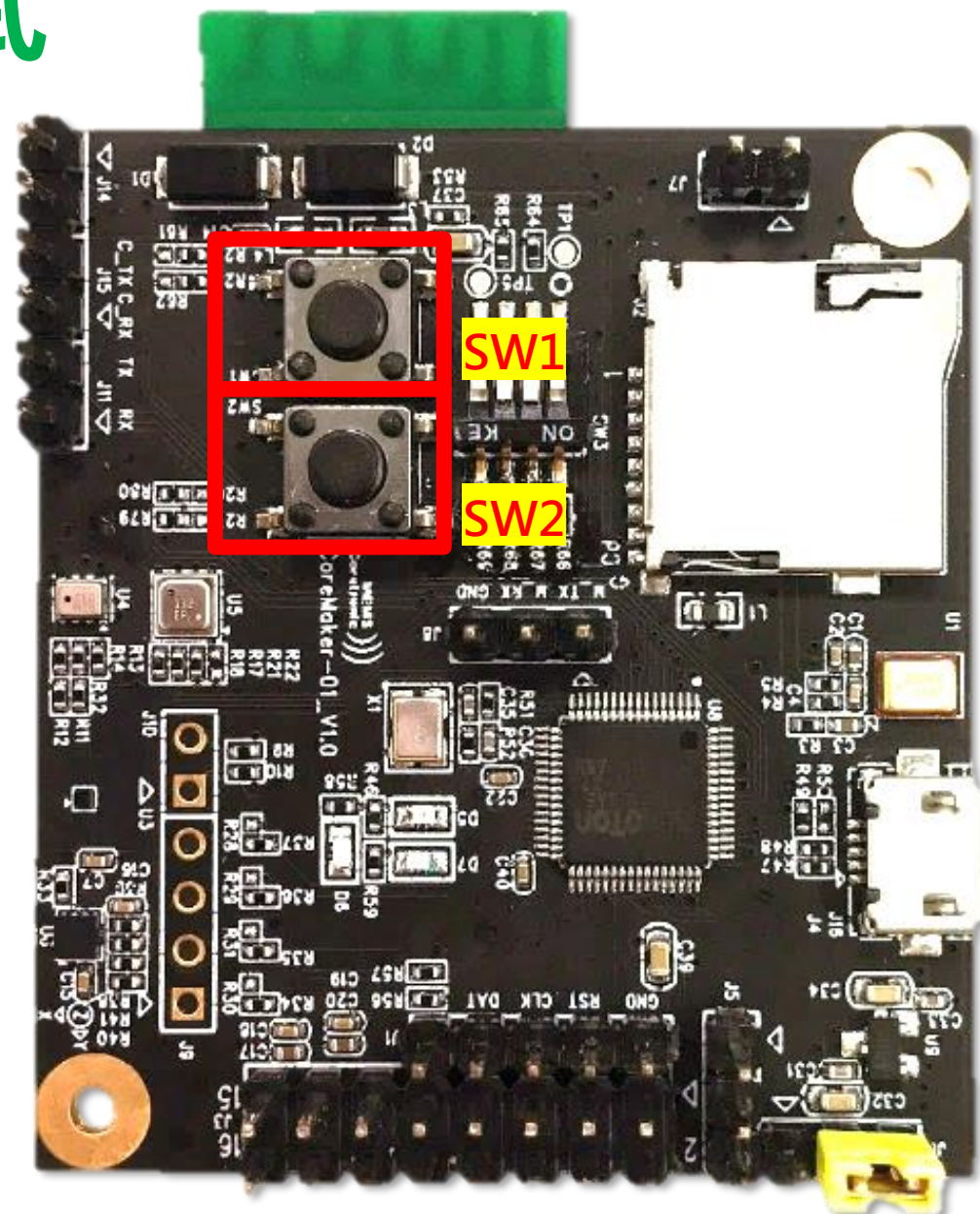
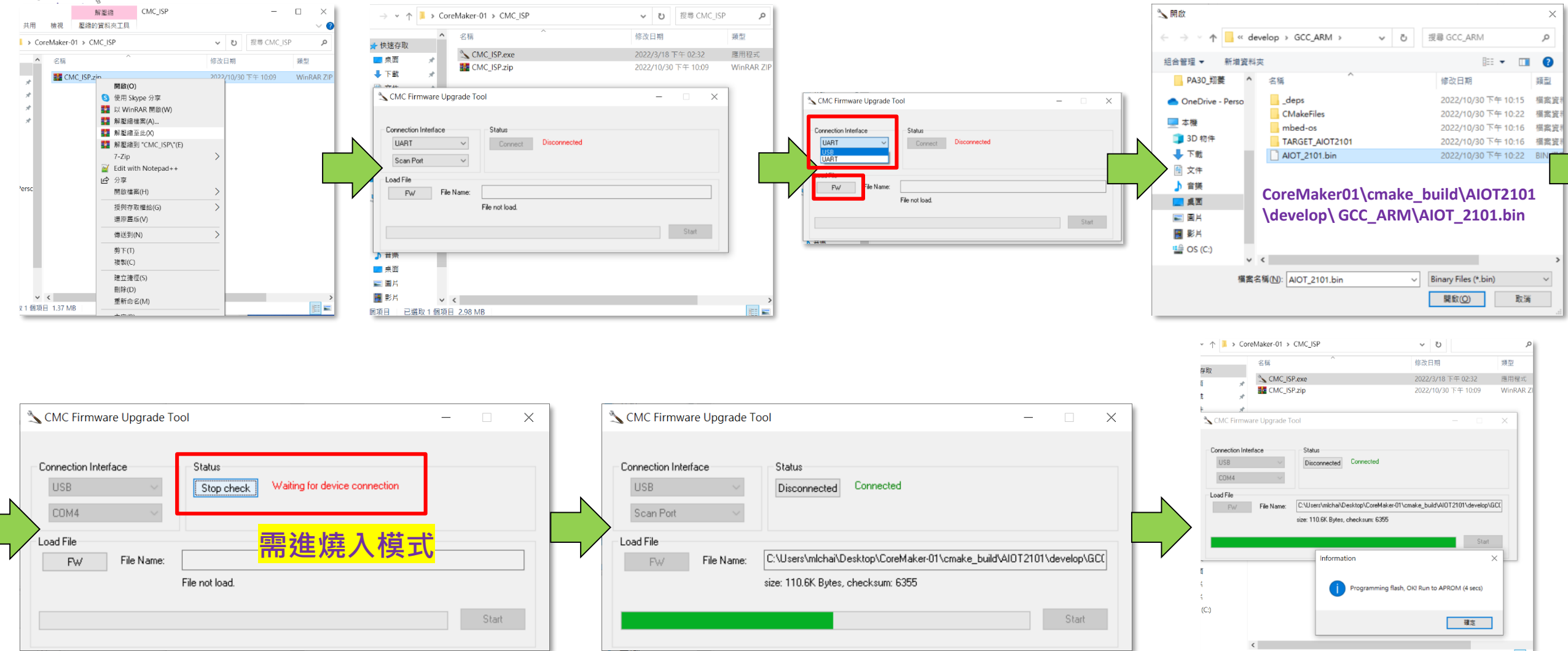


圖 2. Coremaker01 按鈕位置



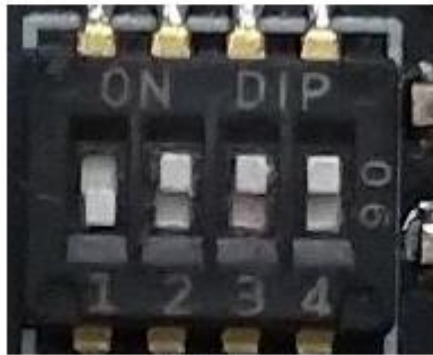
coremaker01 燒入操作

(亦可先進燒入模式再選燒入檔)

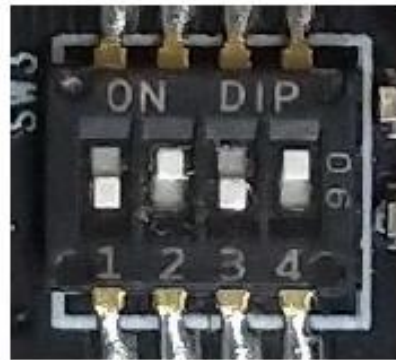


Coremaker-01蒐集資料設定

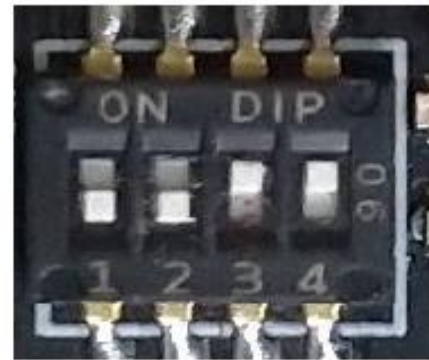
於晶片上須設定使用感測器，本案例需選聲音類別:



聲音



加速度

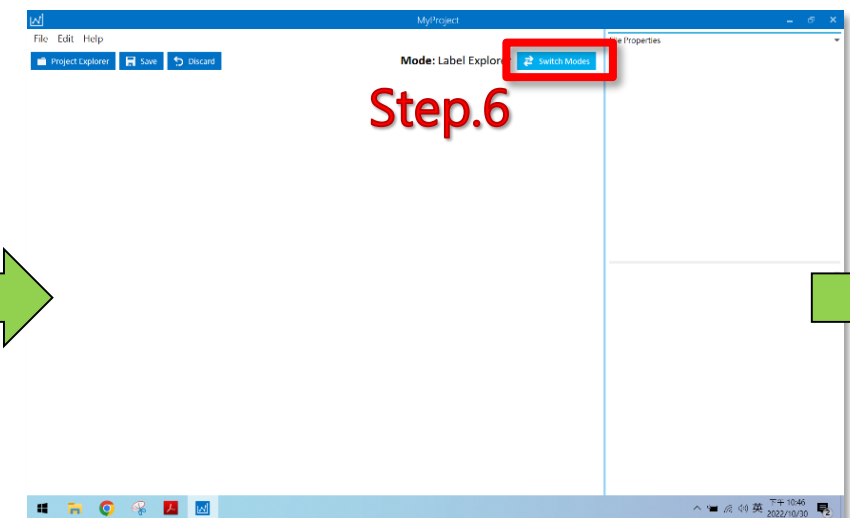
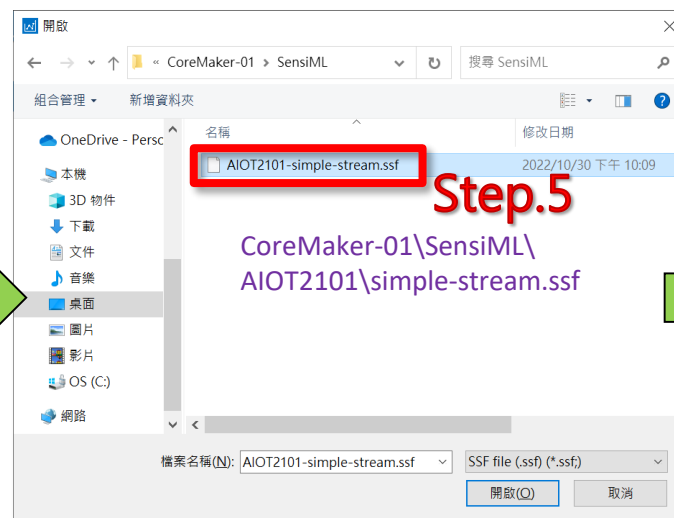
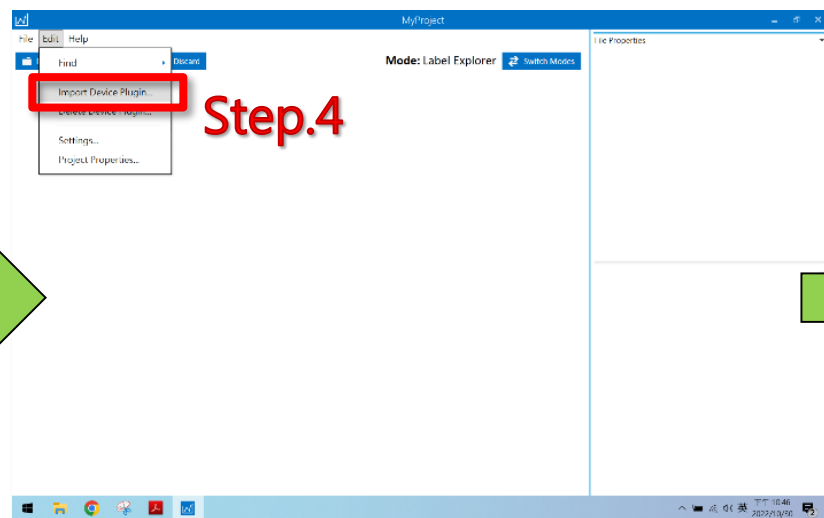
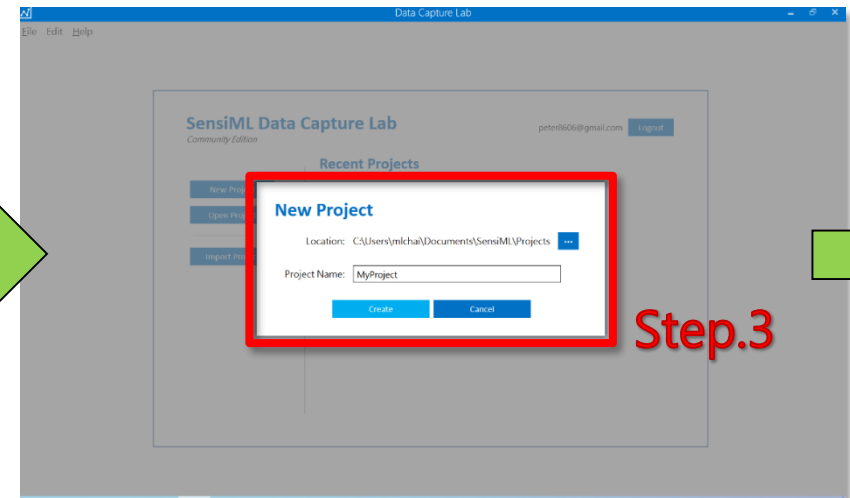
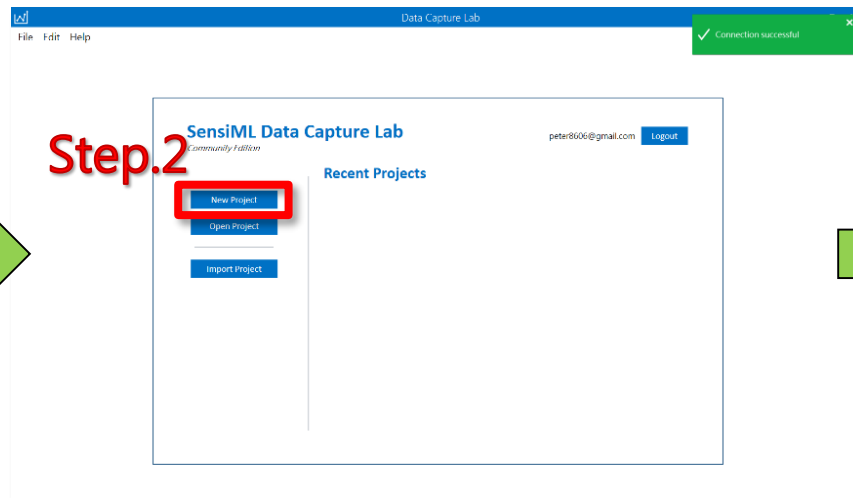
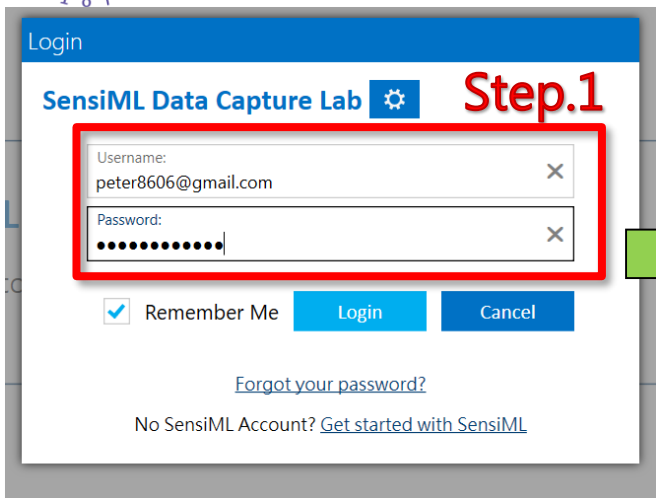


環境



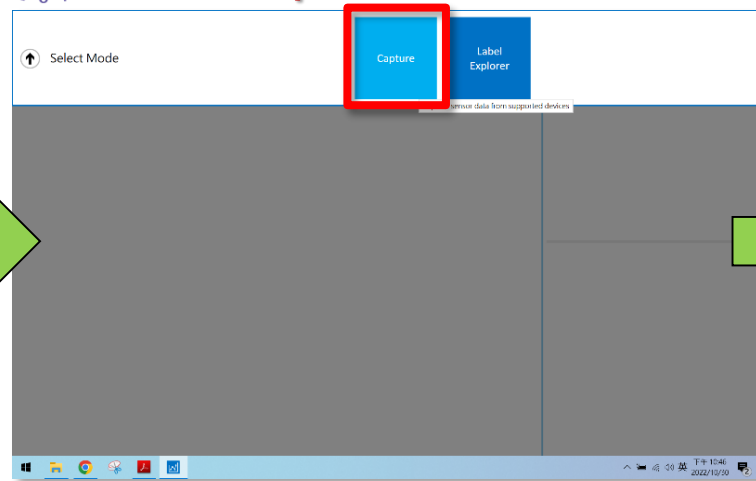
壓力

蒐集數據1

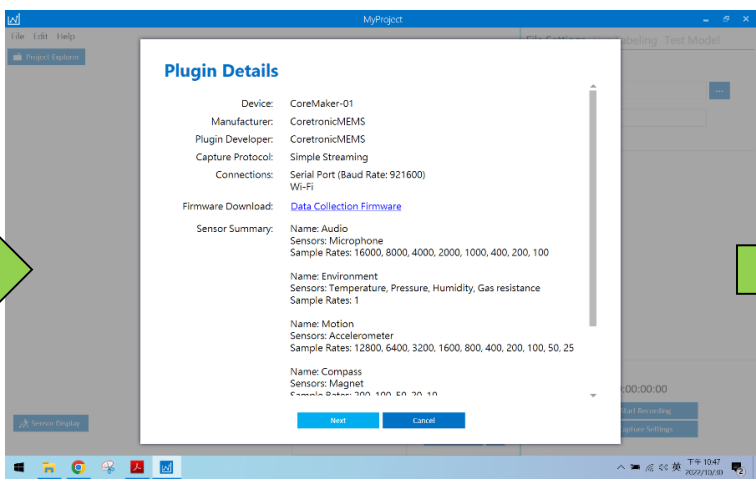
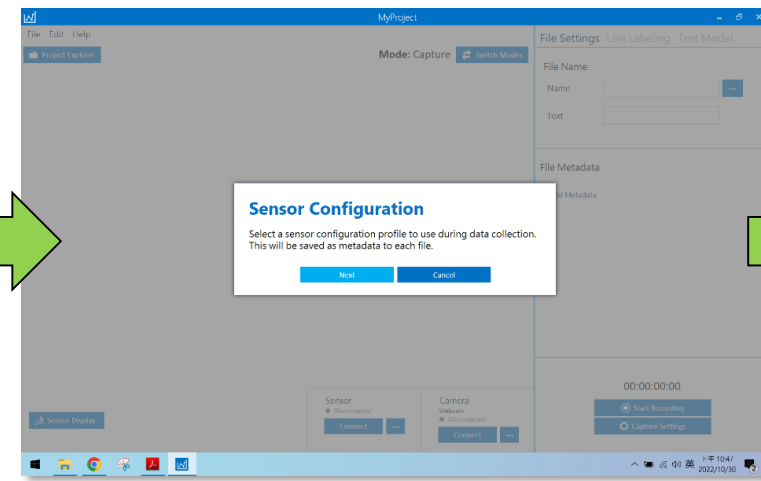
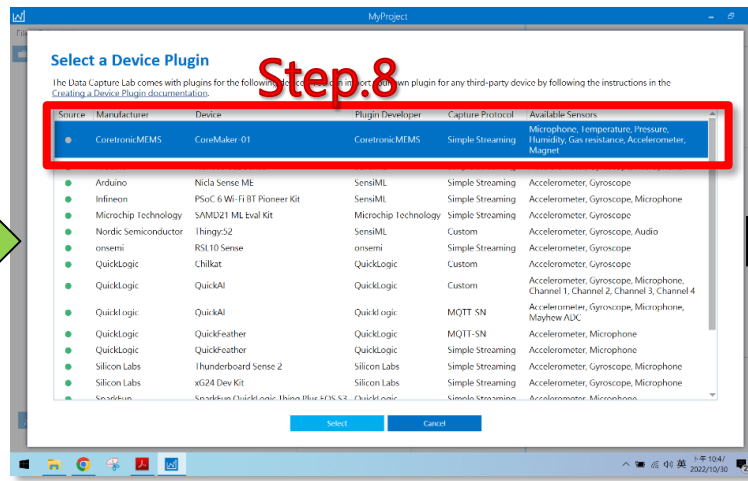


蒐集數據2

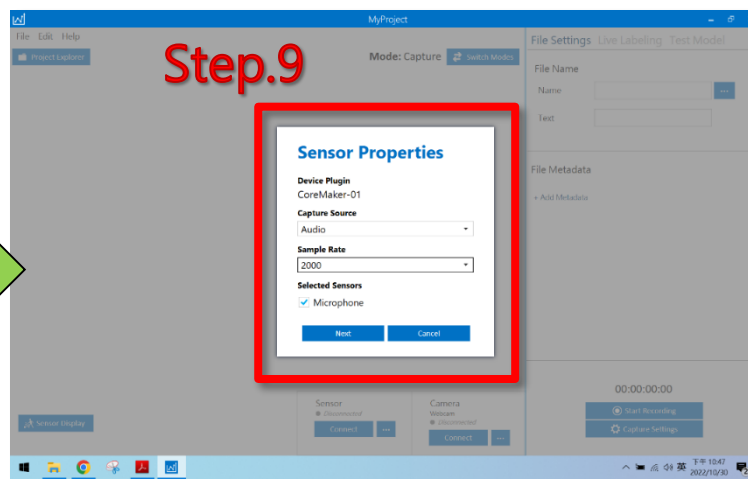
Step.7



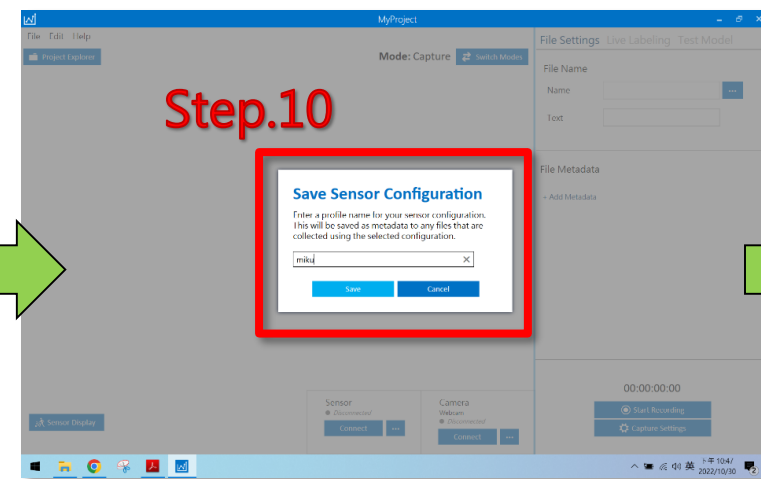
Step.8



Step.9



Step.10



蒐集數據3

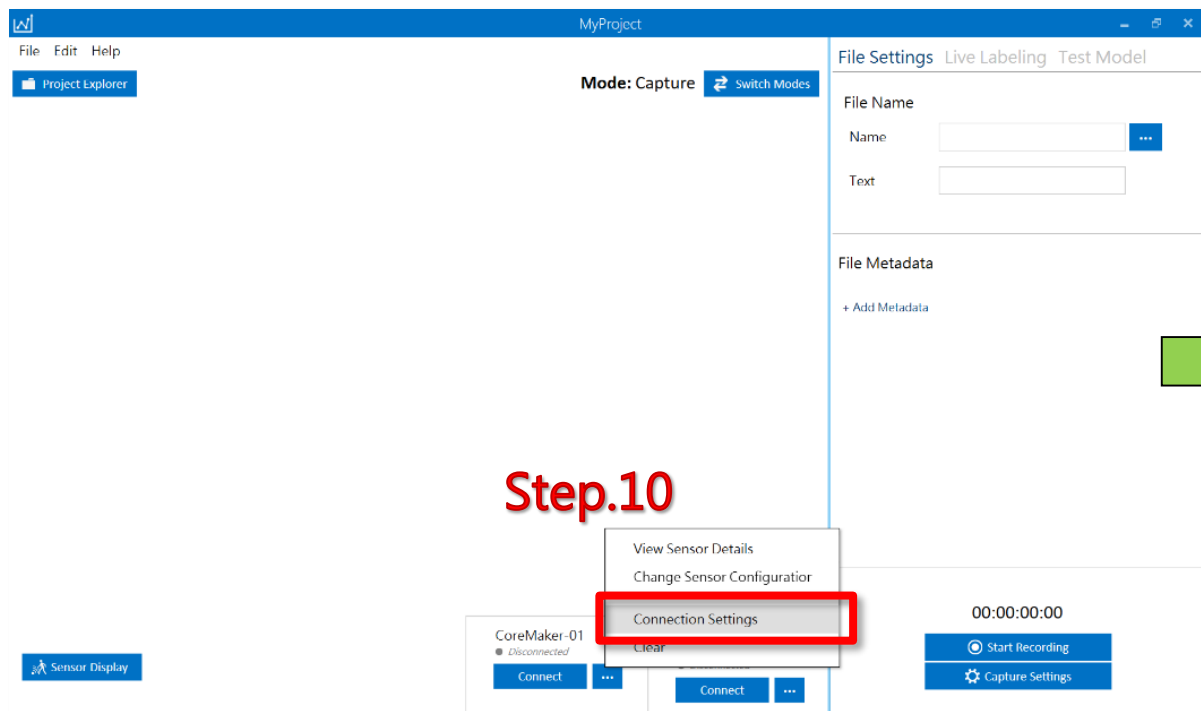


圖 1.點選連線設定'Connection Settings'

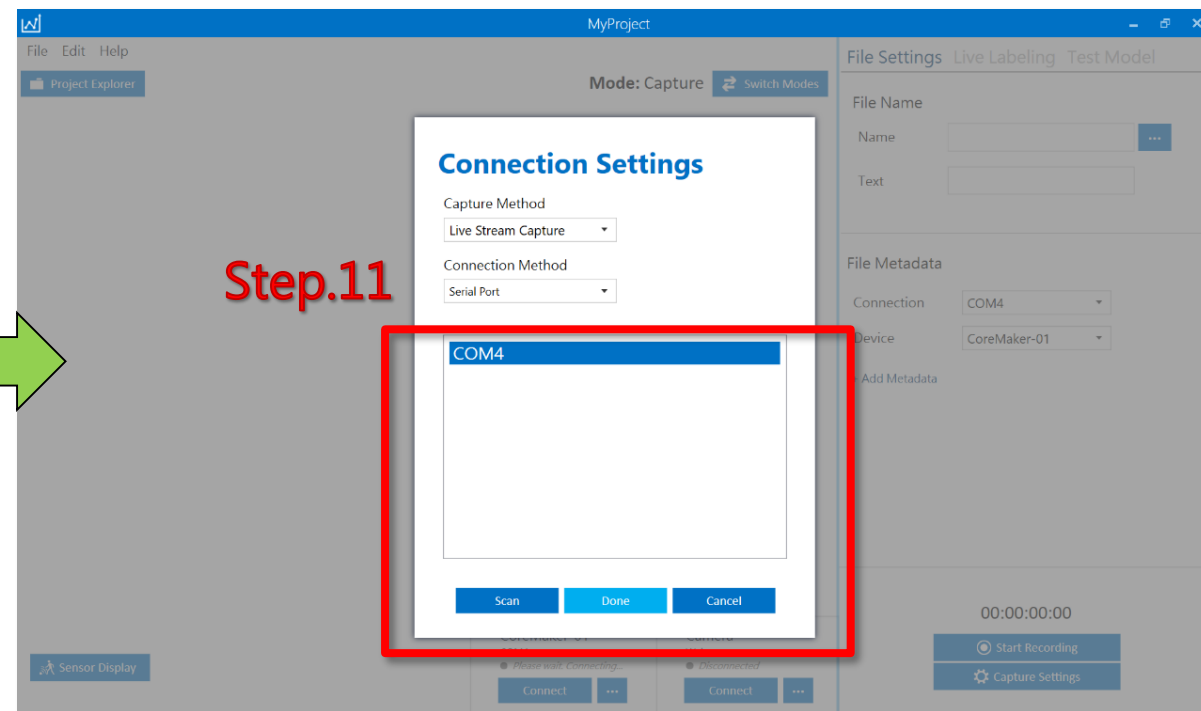


圖 2. 先按下'Scan'找到序列埠選之後再按'Done'

蒐集數據4

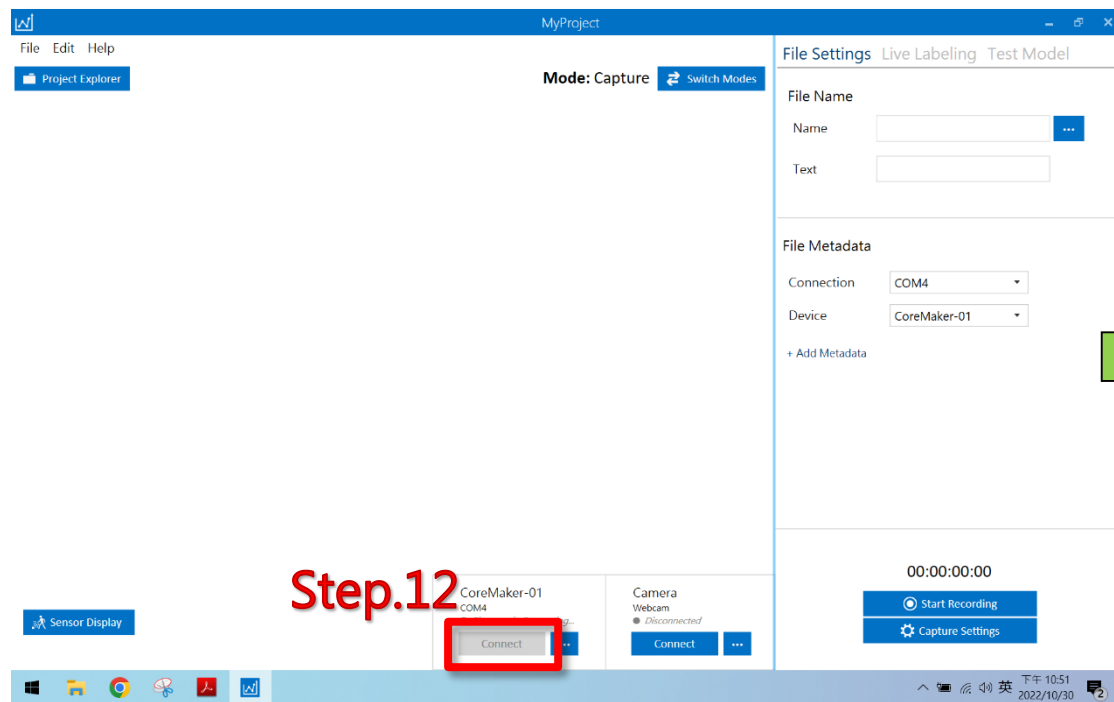


圖 1.點選連線設定'Connection'

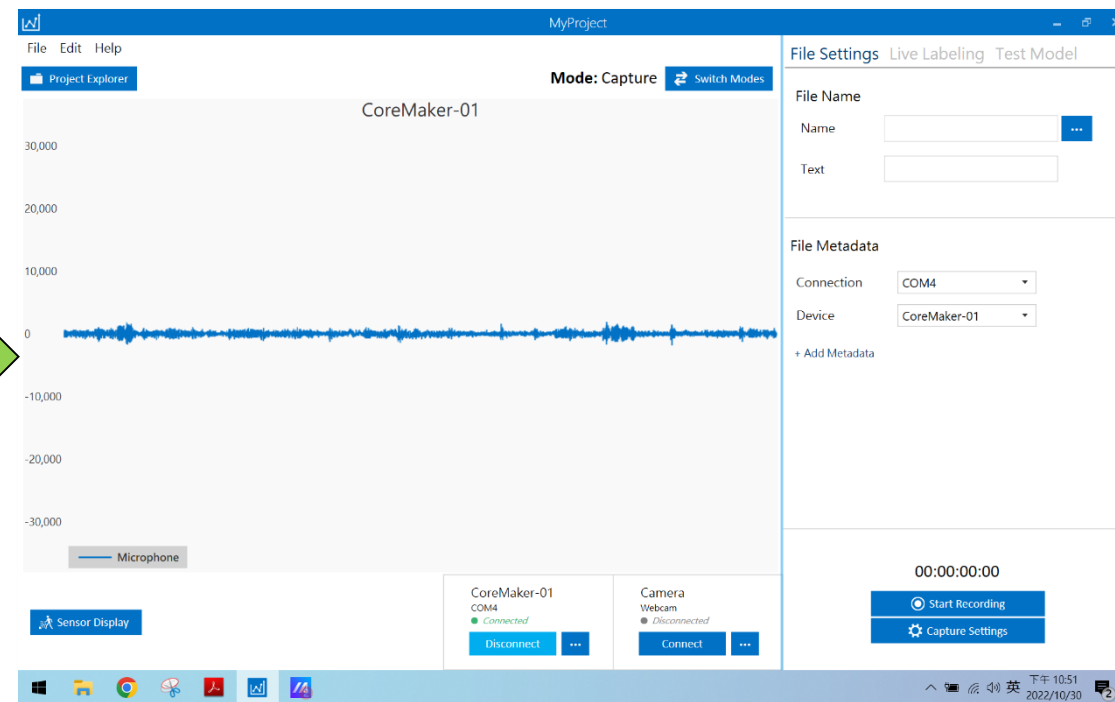
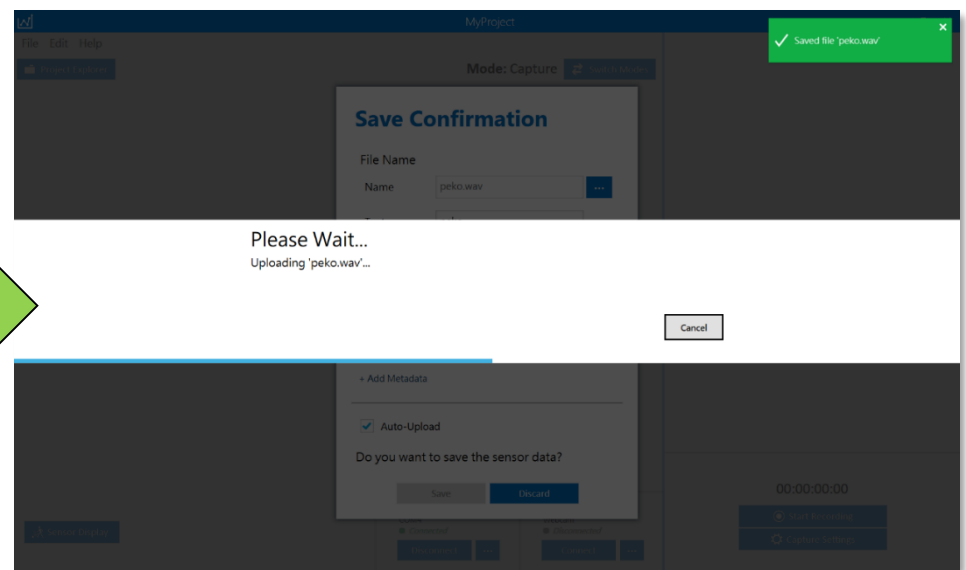
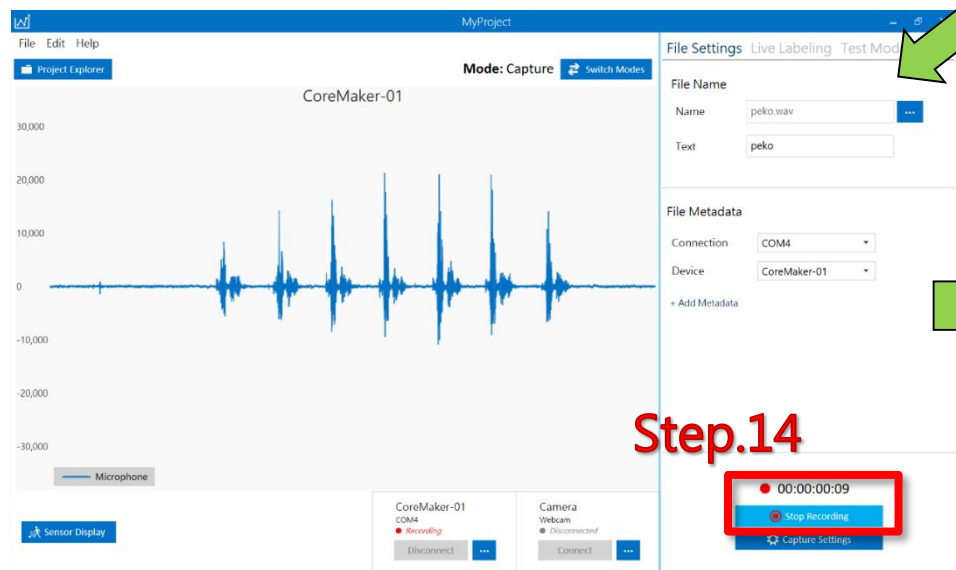
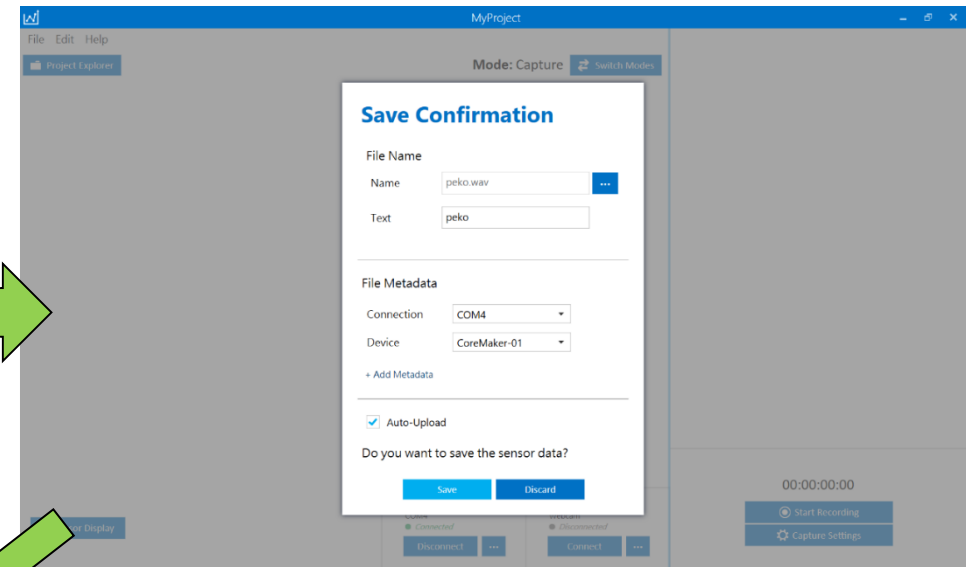
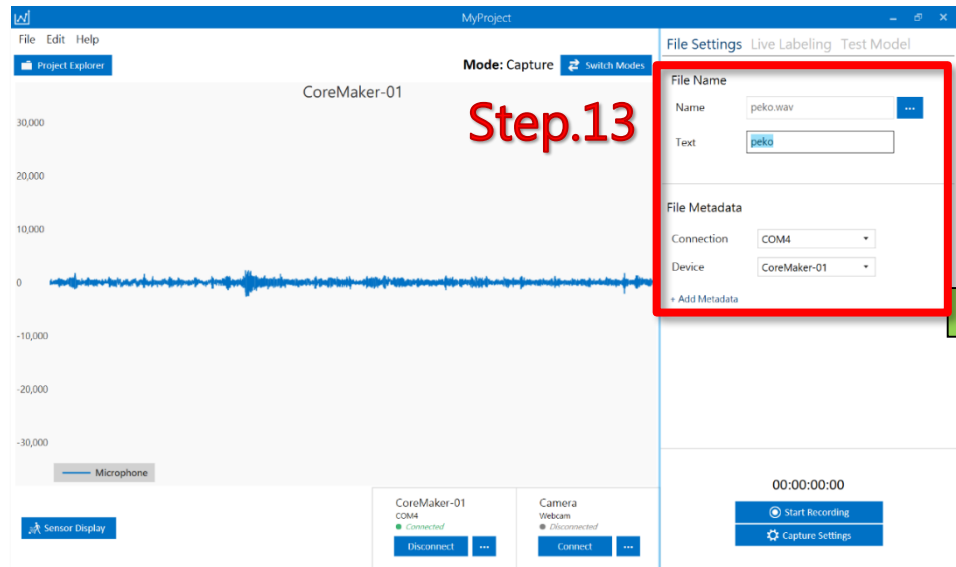


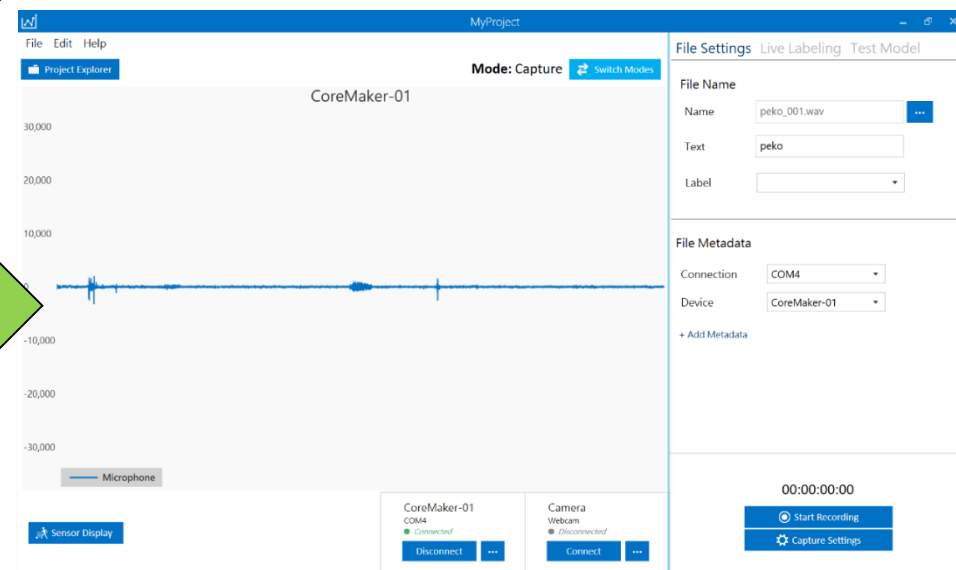
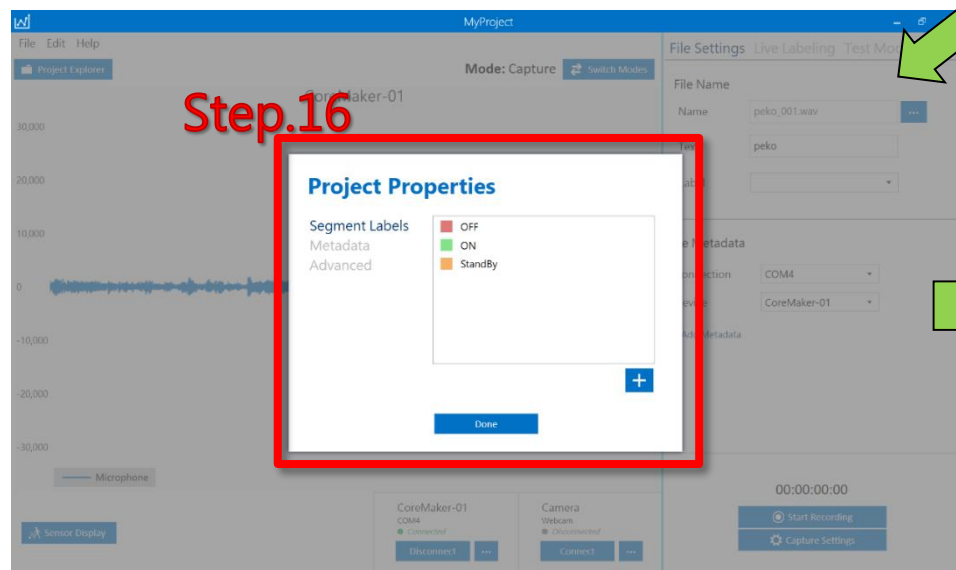
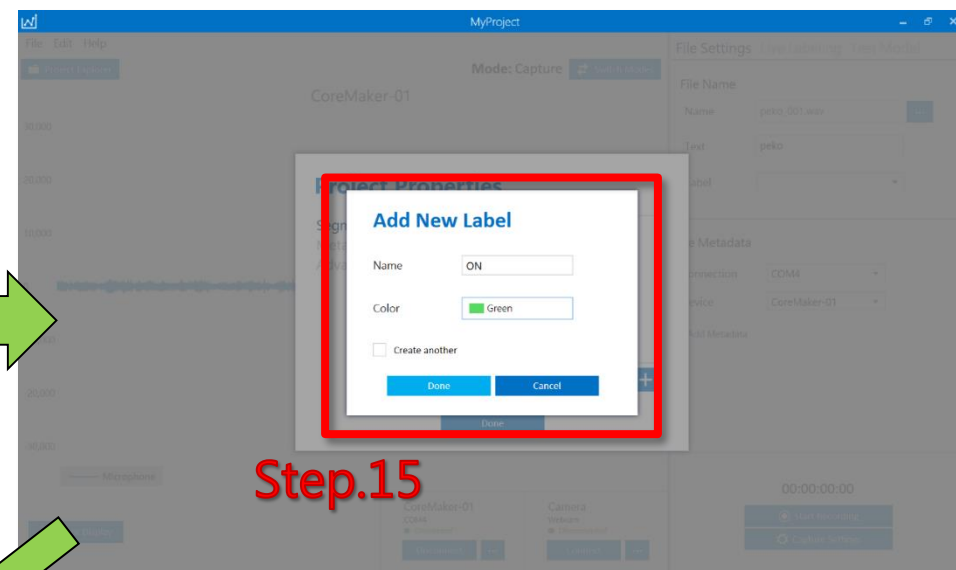
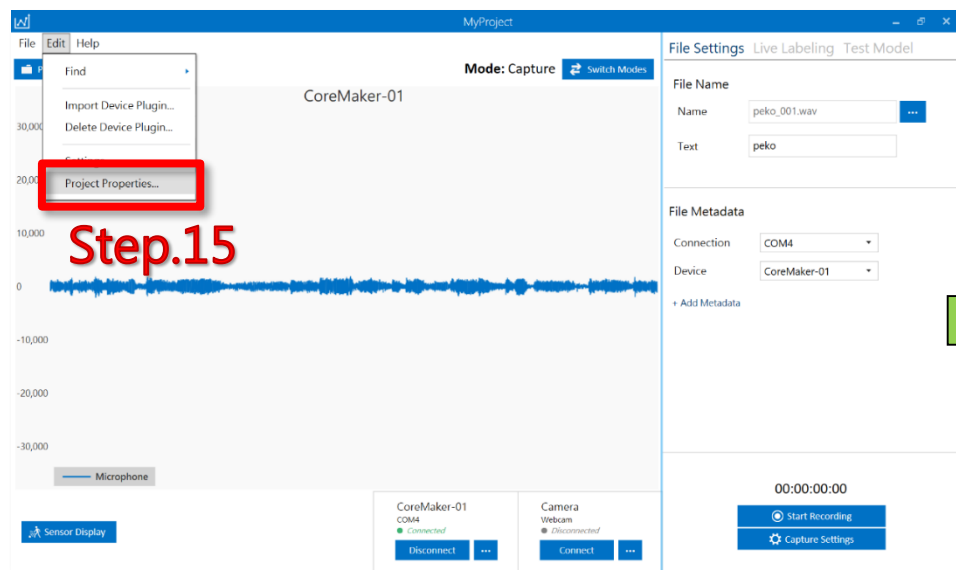
圖 2. 成功連線

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。

蒐集數據5



蒐集數據6

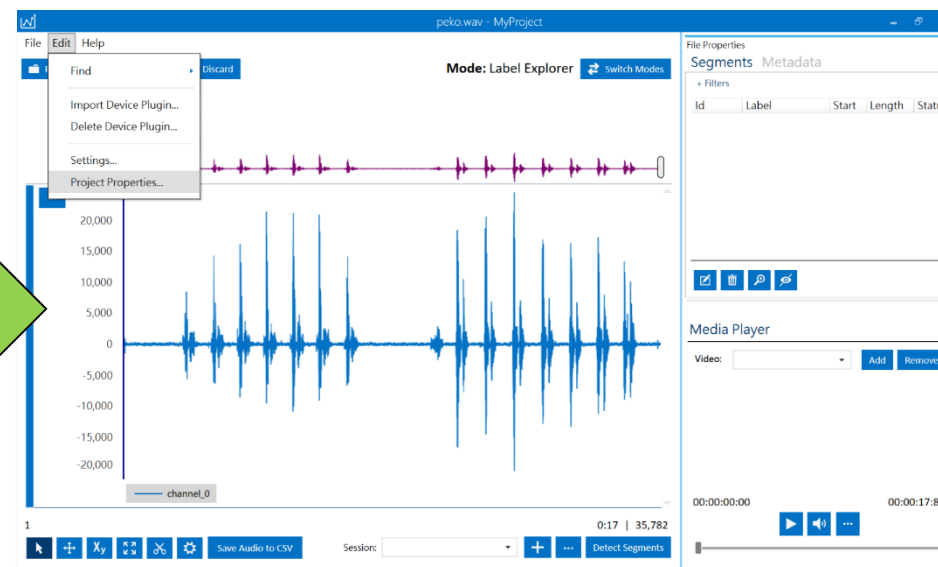
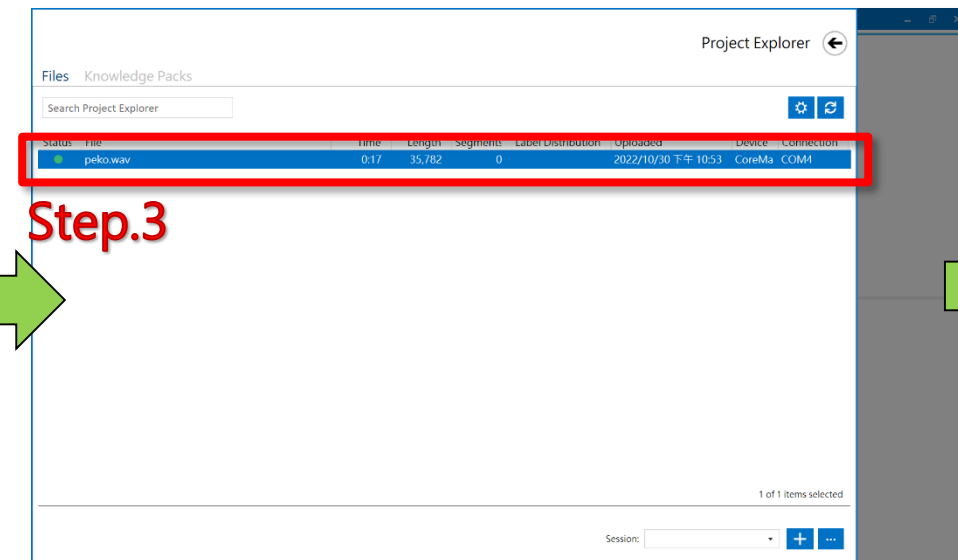
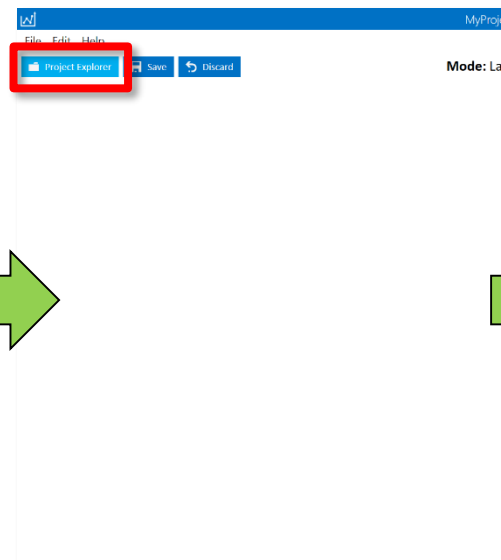
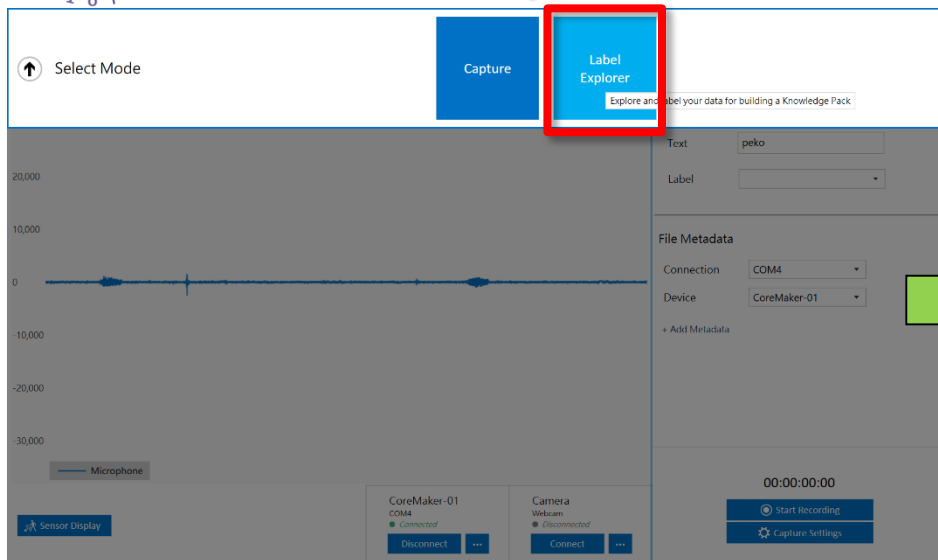


分類標註1

Step.1

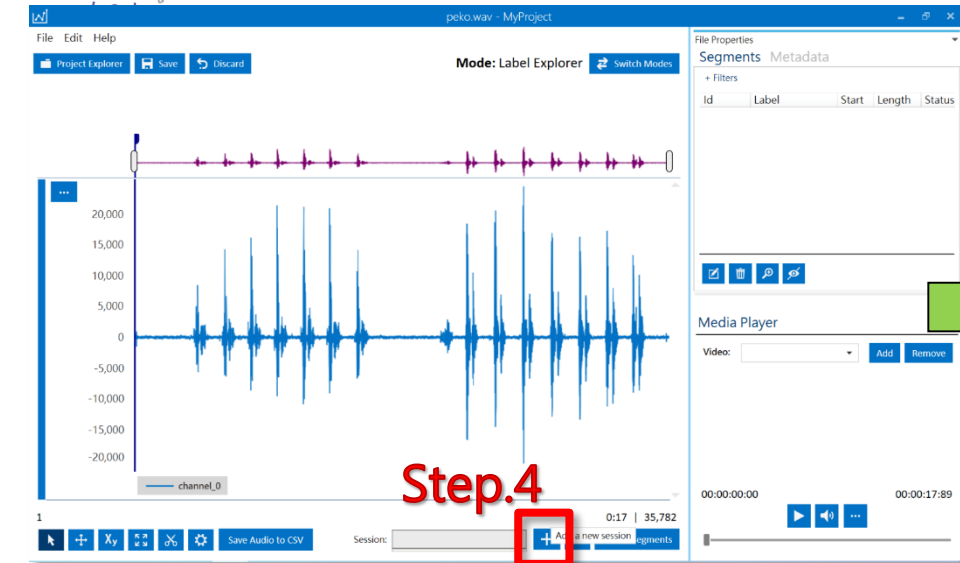
Step.2

Step.3



分類標註2

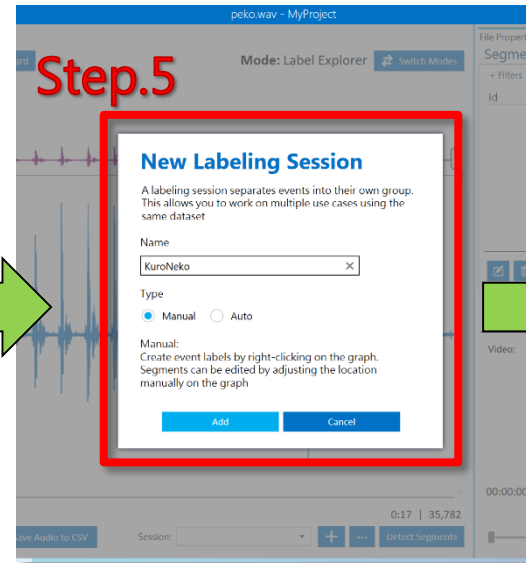
Step.4



The screenshot shows the 'Mode: Label Explorer' window. The 'Add a new session' button is highlighted with a red box. The 'Segments' table is empty.

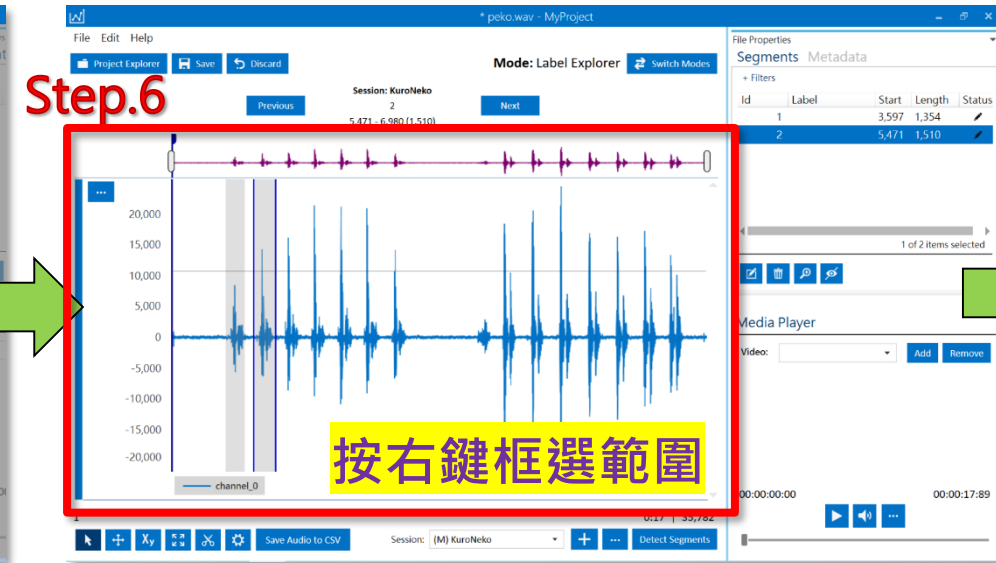
Id	Label	Start	Length	Status
----	-------	-------	--------	--------

Step.5



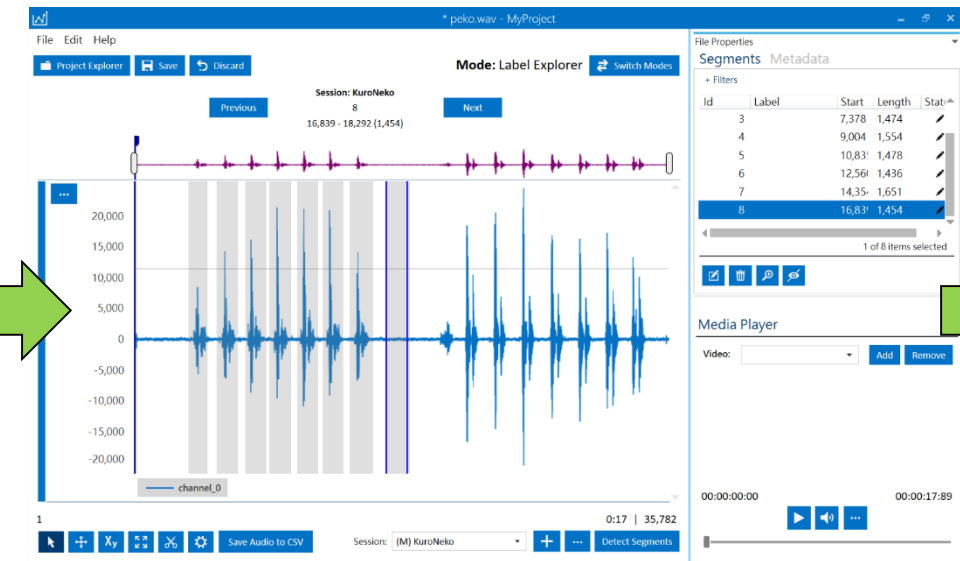
The 'New Labeling Session' dialog box is shown. The 'Name' field is 'KuroNeko'. The 'Type' is 'Manual'. The 'Manual' option is selected. The 'Add' button is highlighted.

Step.6



The screenshot shows the waveform with a range selection. A yellow box highlights the text '按右鍵框選範圍' (Click right button to select range). The 'Segments' table shows two items.

Id	Label	Start	Length	Status
1		3,597	1,354	
2		5,471	1,510	



The screenshot shows the 'Detect Segments' button highlighted with a red box. The 'Segments' table shows 8 items.

Id	Label	Start	Length	Status
3		7,378	1,474	
4		9,004	1,554	
5		10,831	1,478	
6		12,561	1,436	
7		14,351	1,651	
8		16,839	1,454	

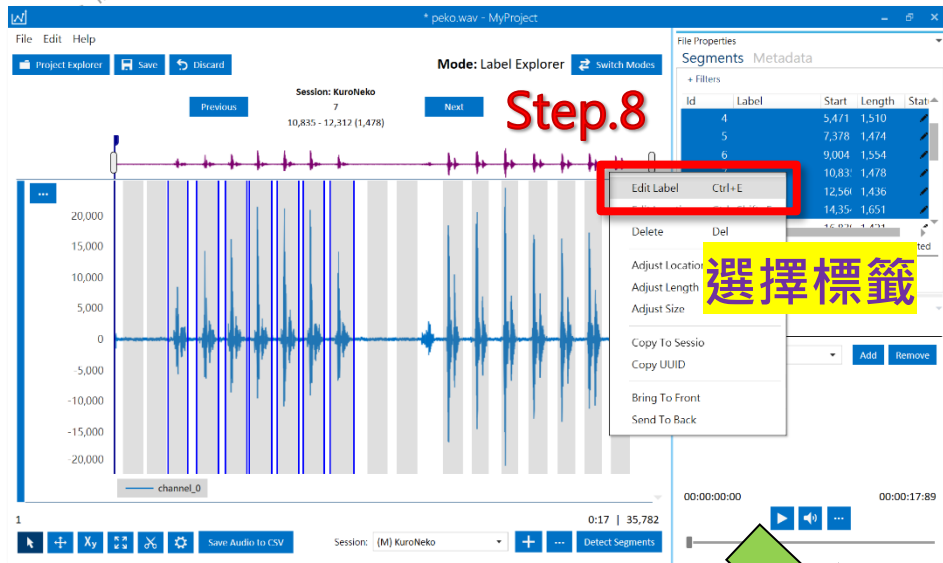
Step.7

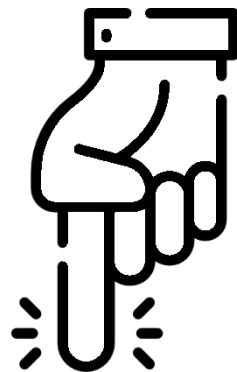


The screenshot shows the 'Detect Segments' button highlighted with a red box. The 'Segments' table shows 9 items.

Id	Label	Start	Length	Status
4		5,471	1,510	
5		7,378	1,474	
6		9,004	1,554	
7		10,831	1,478	
8		12,561	1,436	
9		14,351	1,651	

分類標註3



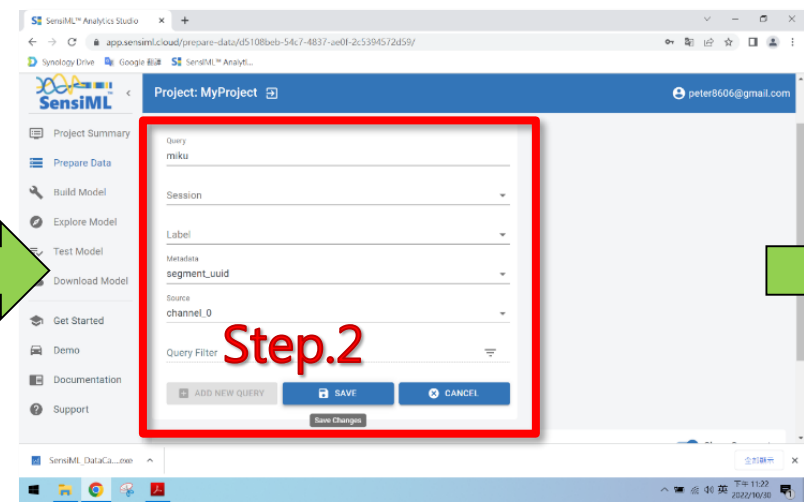
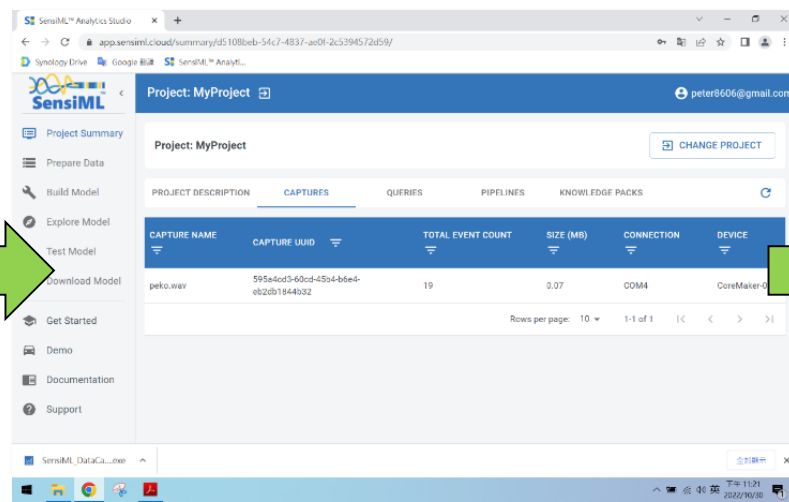
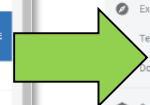
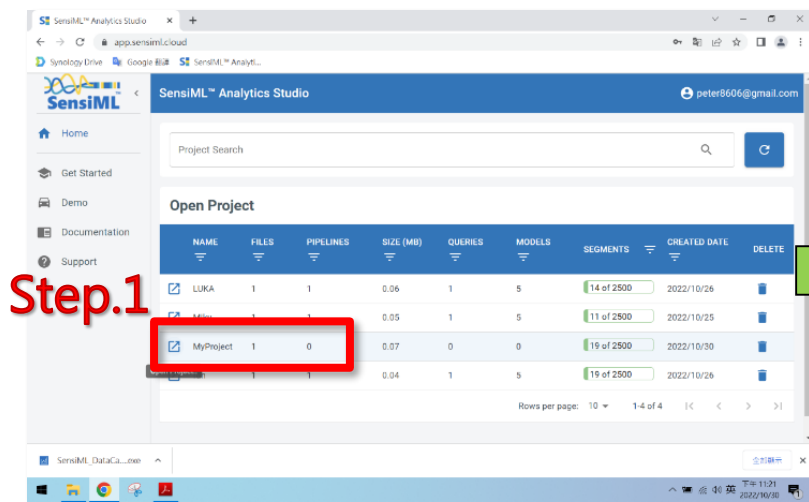


<https://app.sensiml.cloud>

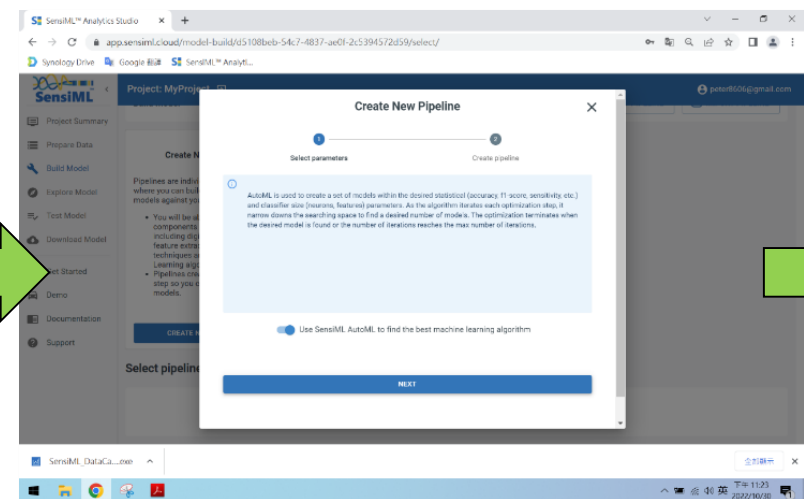
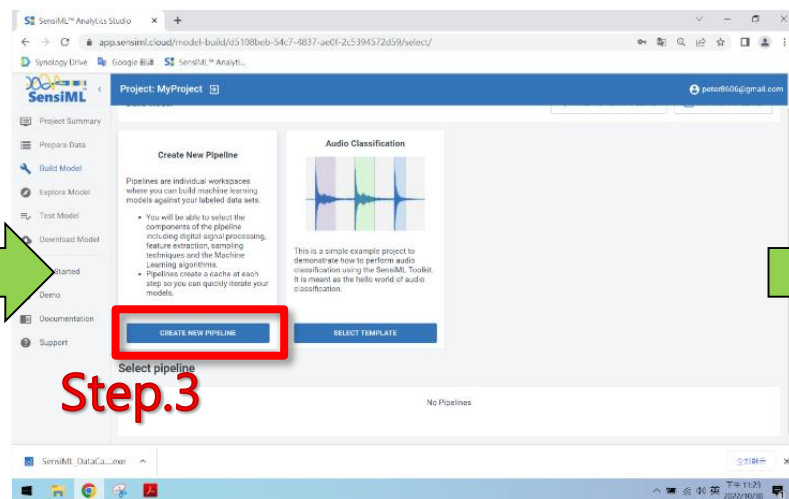
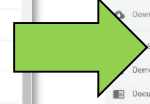
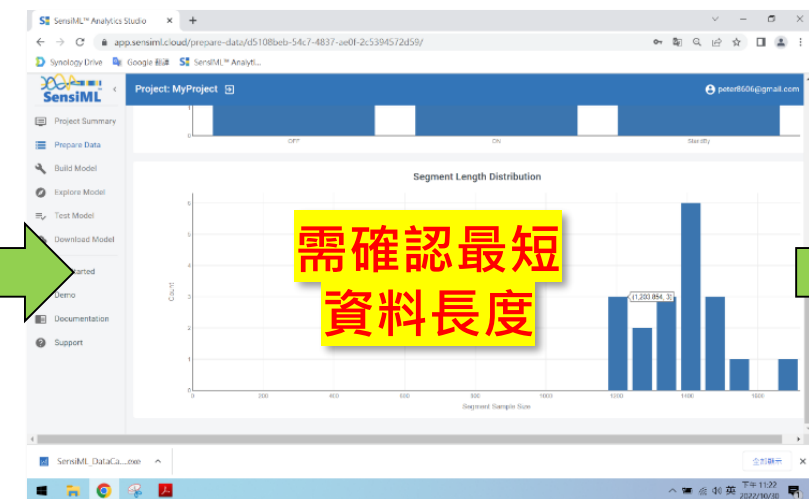
前往SensiML網站 ➔

模型訓練1

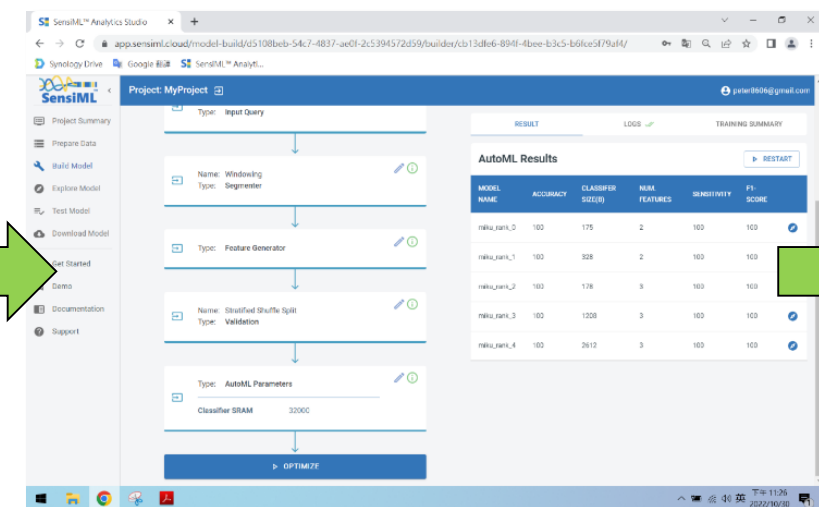
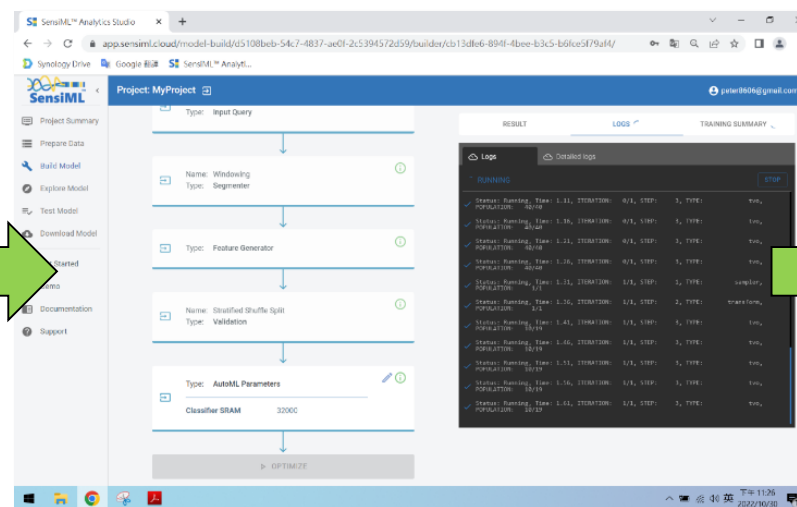
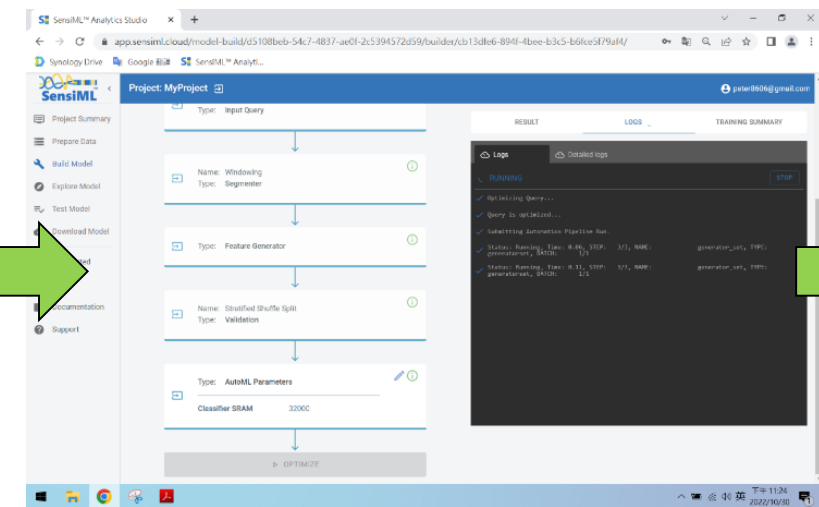
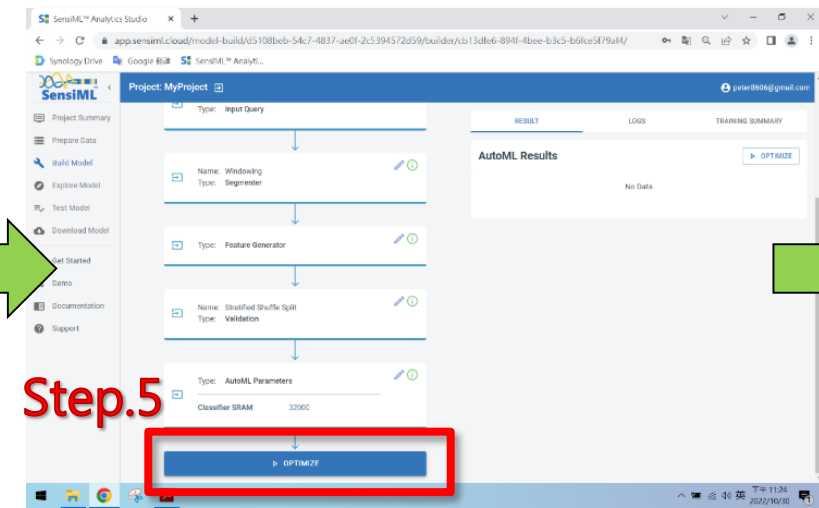
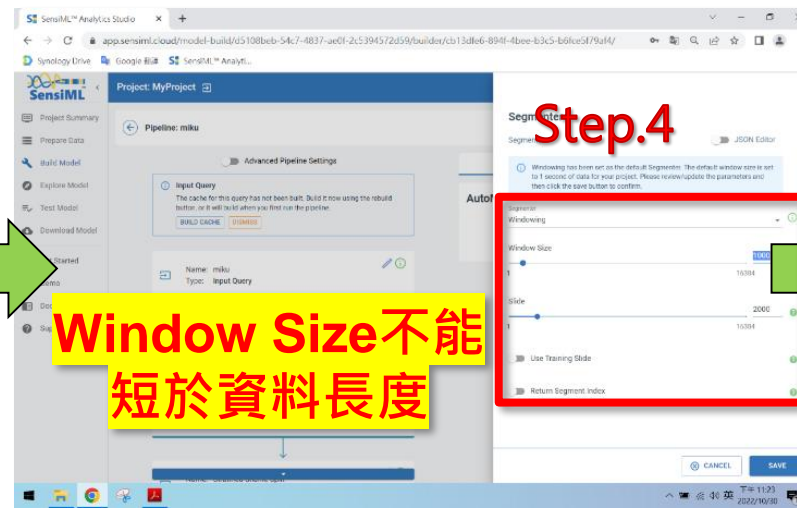
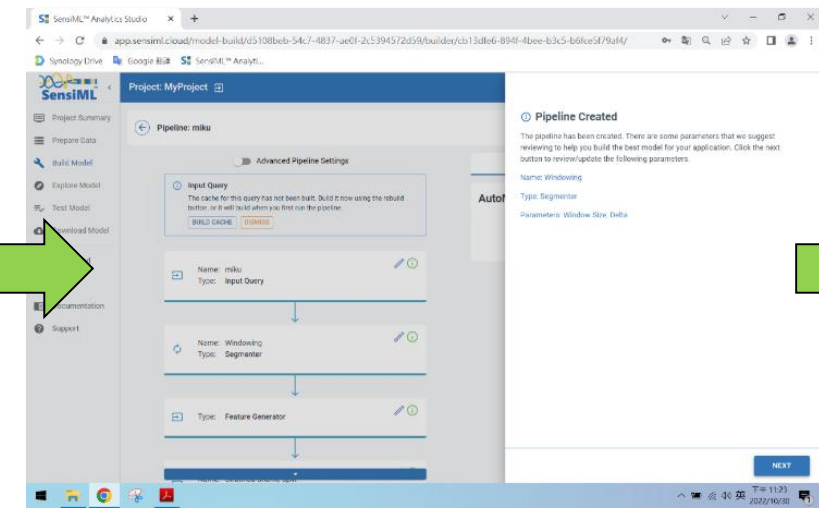
Step.1



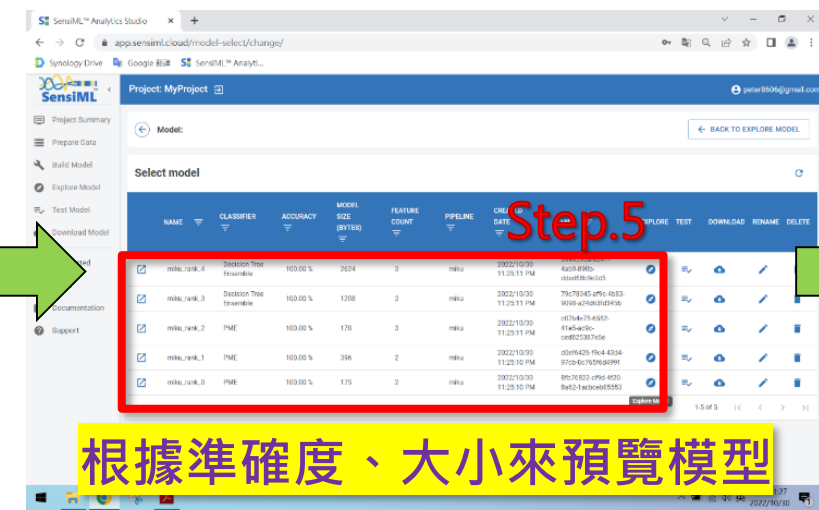
需確認最短
資料長度



模型訓練2

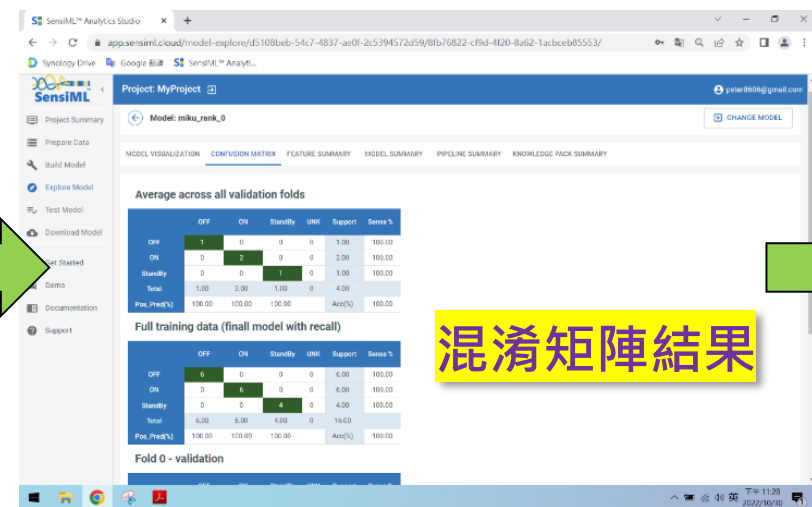
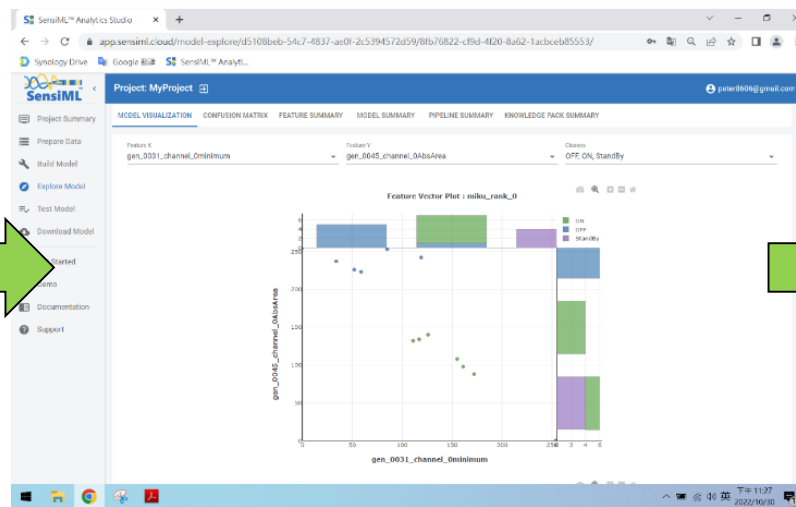
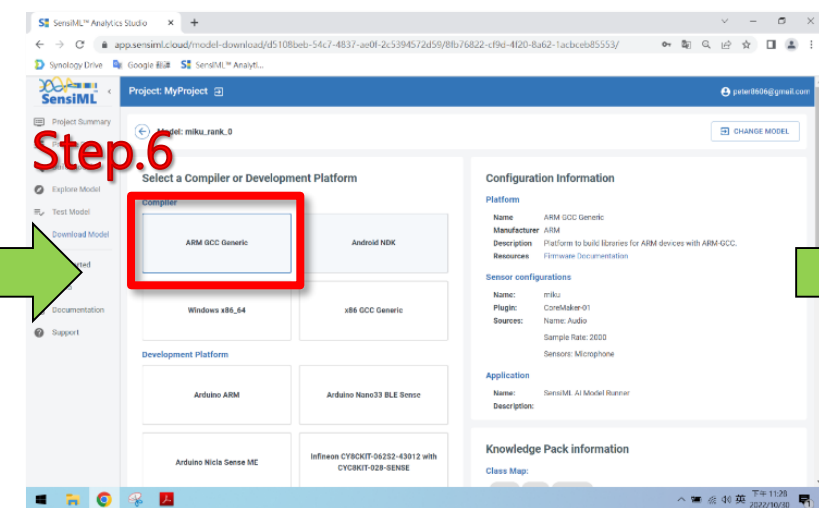


模型訓練3

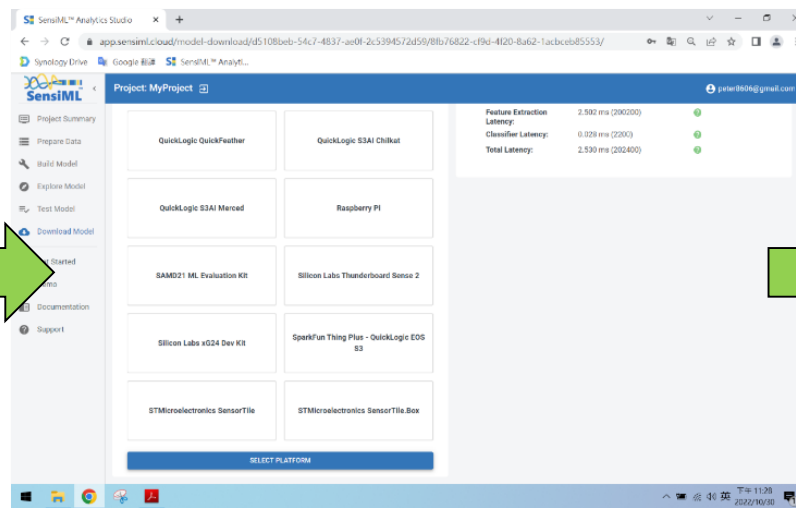
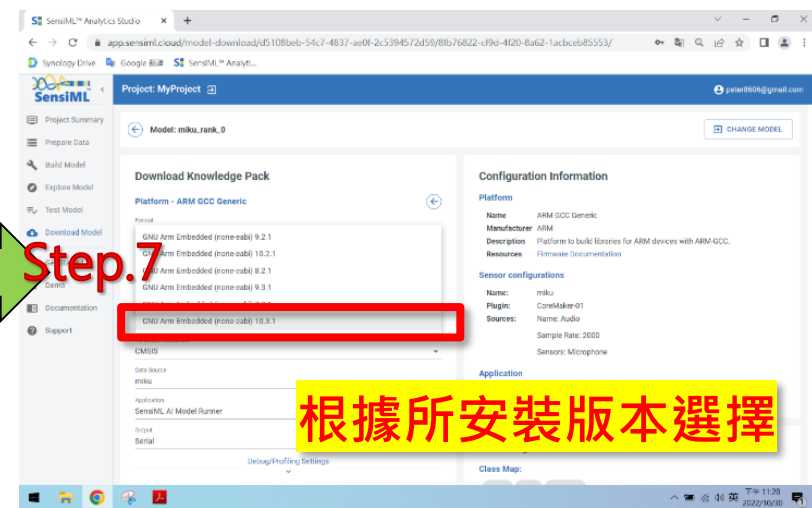


Step.5

根據準確度、大小來預覽模型

Step.6

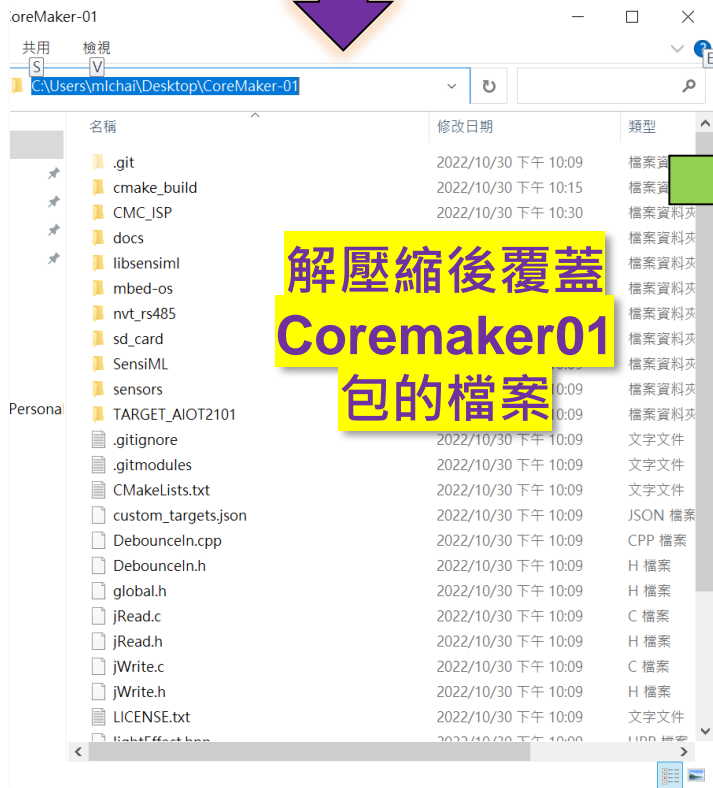
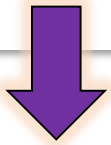
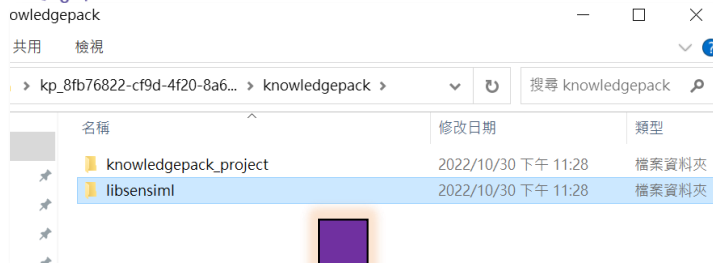



Step.7

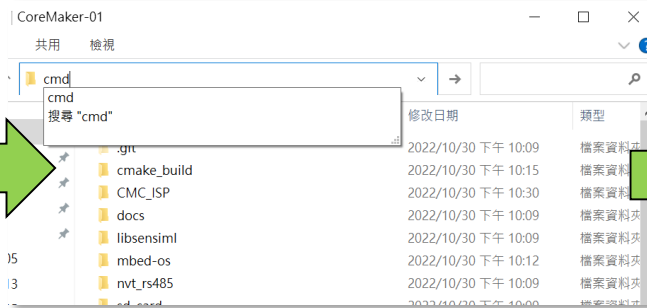
根據所安裝版本選擇



Coremaker-01 燒入



解壓縮後覆蓋
Coremaker01
包的檔案



```
C:\Windows\System32\cmd.exe
Downloading intelhex-2.3.0-py2.py3-none-any.whl (50 kB)
----- 50.9/50.9 kB 199.4 kB/s eta 0:00:00
Installing collected packages: intelhex
Successfully installed intelhex-2.3.0

C:\Users\mlchai\Desktop\CoreMaker-01>mbed-tools compile -m AIOT2101 -t GCC_ARM
Configuring project and generating build system...
-- Checking for Python package intelhex -- found
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/mlchai/Desktop/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM
Building Mbed project...
[1/1] Linking CXX executable AIOT_2101.elf
-- built: C:/Users/mlchai/Desktop/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.bin
-- built: C:/Users/mlchai/Desktop/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.hex
```

Module	.text	.data	.bss
DebounceIn.cpp.obj	676(+676)	0(+0)	0(+0)
[fill]	212(+212)	0(+0)	52(+52)
[lib]C.a	11236(+11236)	2108(+2108)	58(+58)
[lib]gcc.a	3528(+3528)	0(+0)	0(+0)
[lib]m.a	3864(+3864)	0(+0)	0(+0)
[lib]misc	188(+188)	4(+4)	28(+28)
[obj]stdc++.a	13976(+13976)	104(+104)	4399(+4399)
Write.c.obj	630(+630)	0(+0)	0(+0)
main.cpp.obj	1938(+1938)	0(+0)	4804(+4804)
mbed-os\cmsis	8820(+8820)	168(+168)	5953(+5953)
mbed-os\connectivity	200(+200)	0(+0)	16(+16)
mbed-os\drivers	11766(+11766)	0(+0)	166(+166)
mbed-os\hal	1776(+1776)	4(+4)	59(+59)
mbed-os\platform	6280(+6280)	260(+260)	440(+440)
mbed-os\rtos	1096(+1096)	0(+0)	0(+0)
mbed-os\targets	14356(+14356)	920(+920)	321(+321)
sensors\AcousticNode.cpp.obj	638(+638)	0(+0)	0(+0)
sensors\BME680	5060(+5060)	0(+0)	8(+8)
sensors\GMC306.cpp.obj	726(+726)	0(+0)	0(+0)
sensors\GMP102	1508(+1508)	0(+0)	8(+8)
sensors\KX122-1037	854(+854)	0(+0)	0(+0)
sensors\SensorHub.cpp.obj	1988(+1988)	40(+40)	988(+988)
Subtotals	91316(+91316)	3608(+3608)	17300(+17300)
Total Static RAM memory (data + bss): 20908(+20908) bytes			
Total Flash memory (text + data): 94924(+94924) bytes			

C:\Users\mlchai\Desktop\CoreMaker-01>

重新編譯

> mbed-tools compile -m AIOT2101 -tGCC_ARM

Coremaker-01燒入

- Step1.至CoreMaker-01\CMC_ISP
- Step2.CoreMaker-01進入燒入模式
- Step3.選擇'USB'連線
- Step4.按下'Connect'連線
- Step5.選擇.bin檔案位置
- Step6.開始燒入

CoreMaker01\cmake_build\AIOT2101\develop\
GCC_ARM\AIOT_2101.bin

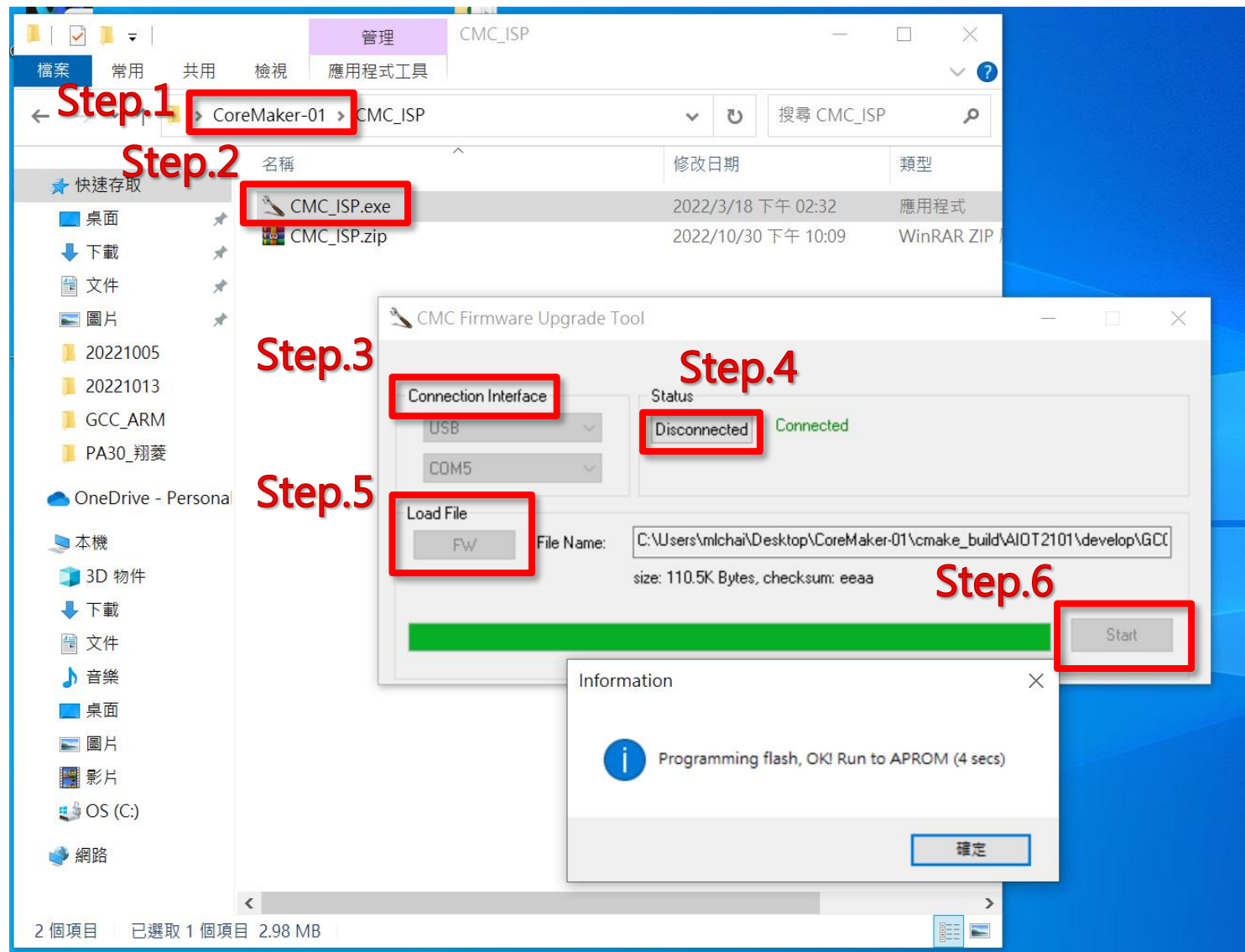


圖 1. Coremaker-01燒入介面操作

Coremaker-01 燒入



圖 1. 打開Tera Term設定鮑率115200

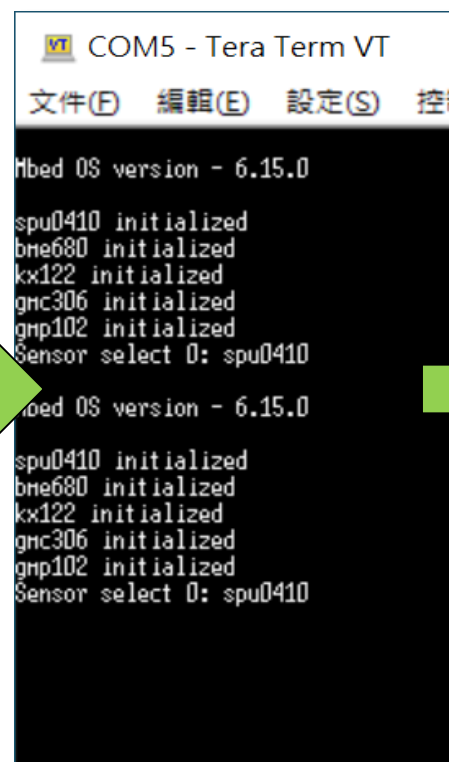


圖 2. 成功連上初始化資料

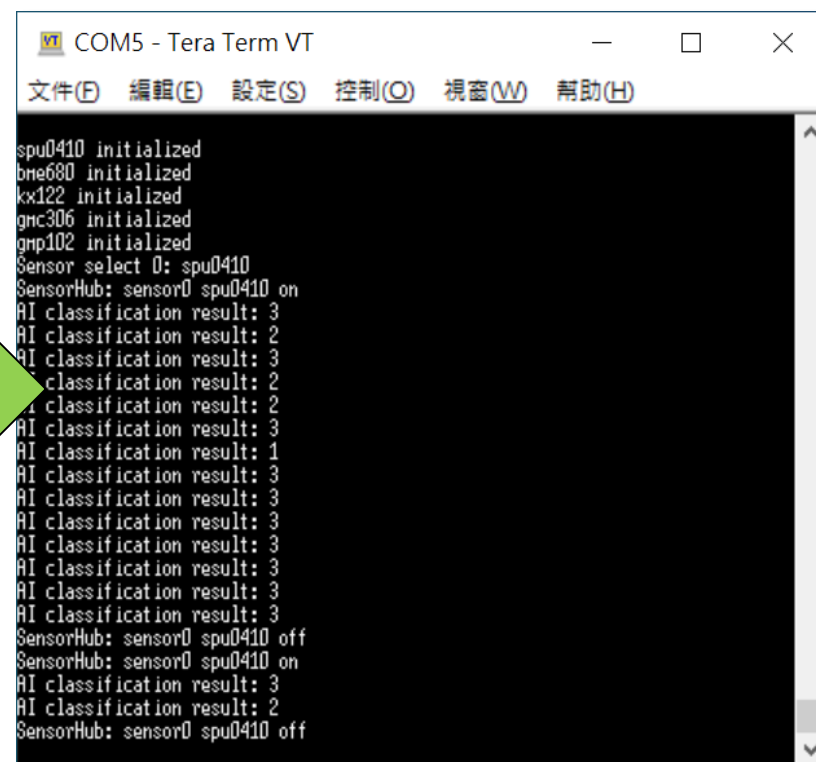


圖 3. 按下'SW2'開始辨識



更改Coremaker-01輸出

Step1. 至 CoreMaker-01/sensors/SensorHub.cpp 的260行

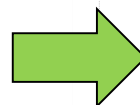
Step2. 將print內改為想要輸出方式

Step3. 重新編譯後燒入

```
> mbed-tools compile -m AIOT2101 -t GCC_ARM
```

```
// if(i == SENSOR_GMP102)
//     gmp102.PrintFormattedData(m_dataBuffer);

if(m_DCLStatus == DCL_CONNECTED)
{
    serial.send((uint8_t*)m_dataBuffer, m_dataLen);
}
else if(run_ai_model[i])
{
    int ret = run_ai_model[i]((SENSOR_DATA_T*)m_dataBuffer, m_dataLen/sizeof(short), 0);
    if (ret > -1)
    {
        printf("AI classification result: %d\n", ret);
        kb_reset_model(0); // Reset running model to initial state.
    }
    else if (ret == -2)
        printf("This segment has been filtered.\n");
    else if (ret < -2)
        printf("AI error: %d\n", ret);
}
```



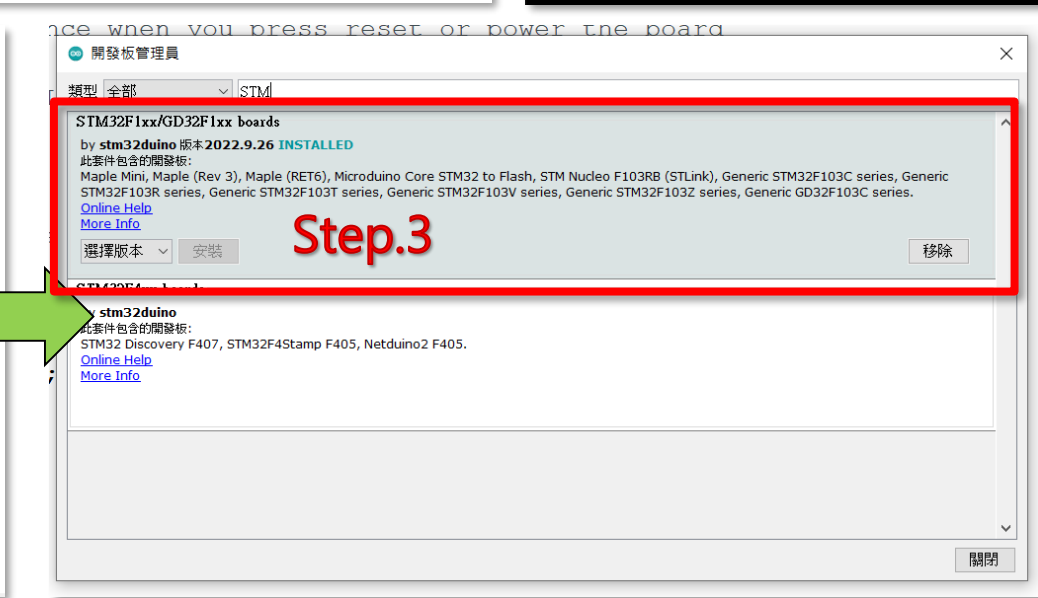
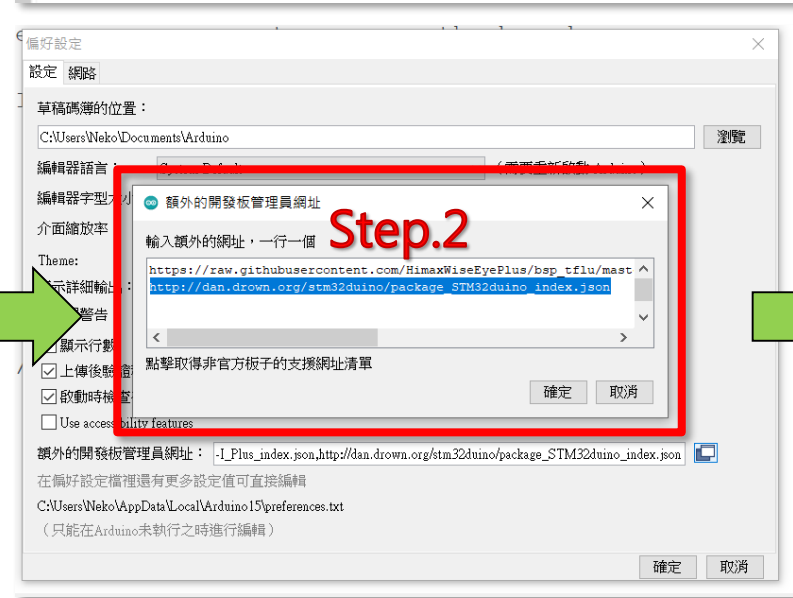
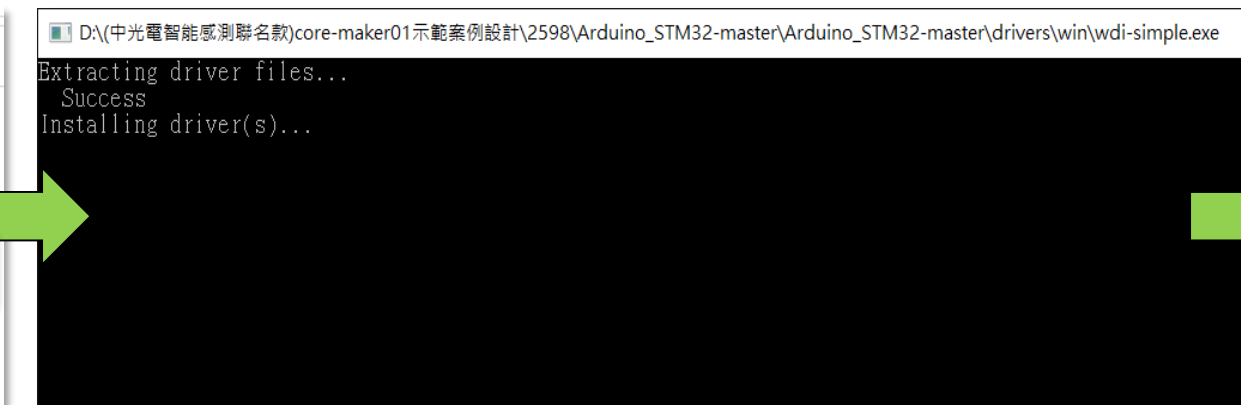
```
// if(i == SENSOR_GMP102)
//     gmp102.PrintFormattedData(m_dataBuffer);

if(m_DCLStatus == DCL_CONNECTED)
{
    serial.send((uint8_t*)m_dataBuffer, m_dataLen);
}
else if(run_ai_model[i])
{
    int ret = run_ai_model[i]((SENSOR_DATA_T*)m_dataBuffer, m_dataLen/sizeof(short), 0);
    if (ret > -1)
    {
        printf("%d\n", ret);
        kb_reset_model(0); // Reset running model to initial state.
    }
    else if (ret == -2)
        printf("This segment has been filtered.\n");
    else if (ret < -2)
        printf("AI error: %d\n", ret);
}
```

2598 + 設定

Step1. 下載並安裝驅動程式

https://github.com/rogerclarkmelbourne/Arduino_STM32



Step3.在Arduino IDE 功能列的'工具'中選擇'開發板管理員', 連網狀態下安裝 "STM32F1xx/GD32F1xx boards" 確認開發板'STM32F1xx'並且按下更新。

Step2.於額外的開發板管理員網址中輸入：
http://dan.drown.org/stm32duino/package_STM32duino_index.json

2598+ 設定

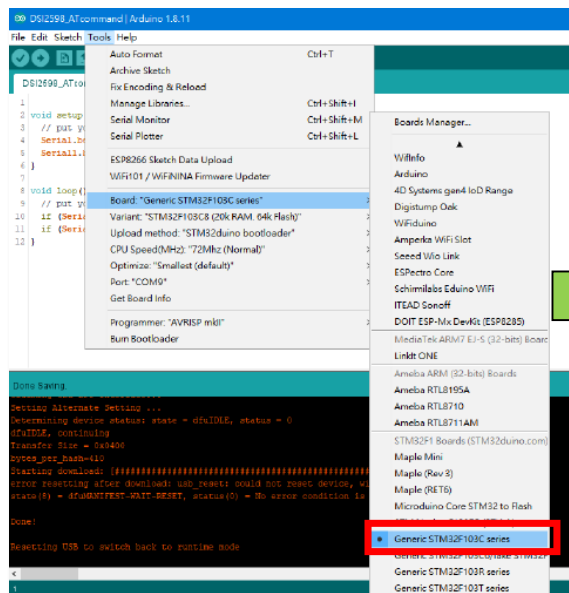


圖 1.選擇開發板: "Generic STM32F103C series"

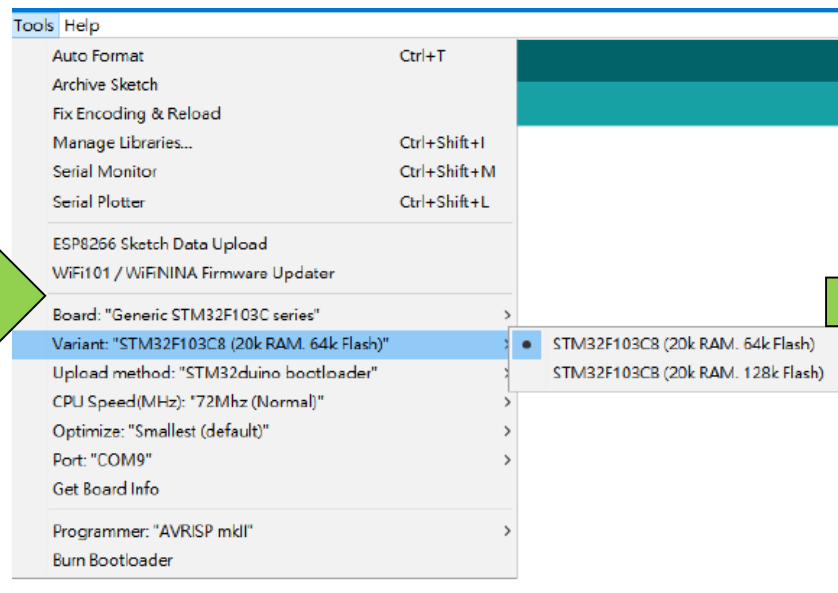


圖 2.選擇64k 的Flash 版本

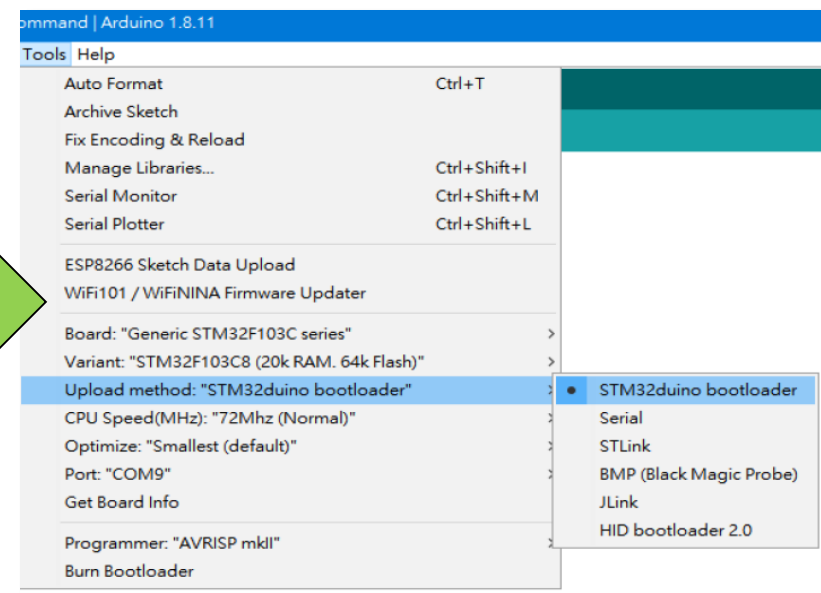


圖 3.預設使用DSI 2598+上的USB，配合已經先預載的bootloader來燒錄。

2598+ 燒入測試

確認可操作2598+燒入以及驗證該晶片運作正常

2598_Blink | Arduino 1.8.18

檔案 編輯 草稿碼 工具 說明



2598_Blink

```
#define pin_led PB12
```

```
// the setup function runs once when you press reset or power the board
```

```
void setup() {
```

```
    // initialize digital pin LED_BUILTIN as an output.
```

```
    pinMode(pin_led, OUTPUT);
```

```
}
```

```
// the loop function runs over and over again forever
```

```
void loop() {
```

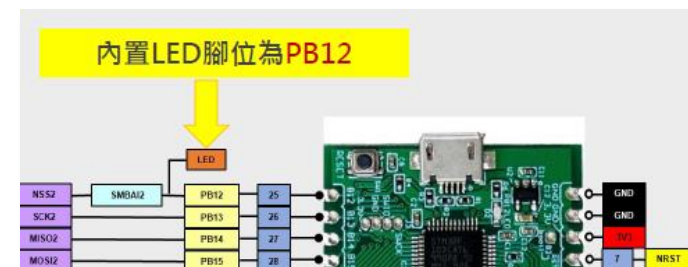
```
    digitalWrite(pin_led, HIGH);    // turn the LED on (HIGH is the voltage level)
```

```
    delay(1000);                    // wait for a second
```

```
    digitalWrite(pin_led, LOW);     // turn the LED off by making the voltage LOW
```

```
    delay(1000);                    // wait for a second
```

```
}
```



程式會使內建LED (PB12)每秒閃爍

程式位置:

https://github.com/wildman8606/Coremaker01withIoT/tree/main/2598/2598_Blink

2598+ 串接Coremaker-01

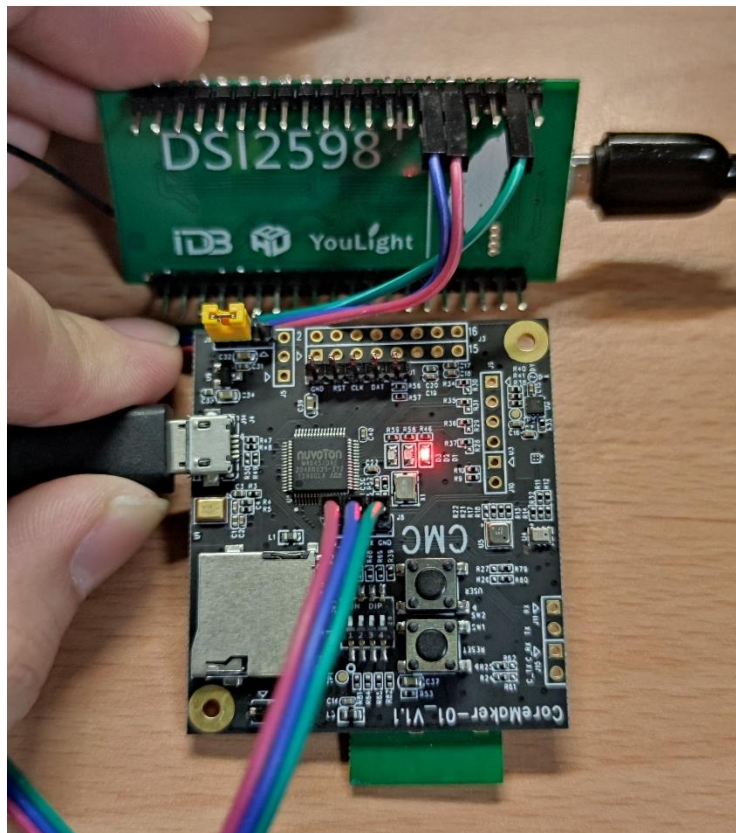


圖 1. 串接實體

DSI-2598+ RX3/B11 → CoreMaker-01 TX
 DSI-2598+ TX3/B10 → CoreMaker-01 RX
 DSI-2598+ GND → CoreMaker-01 GND

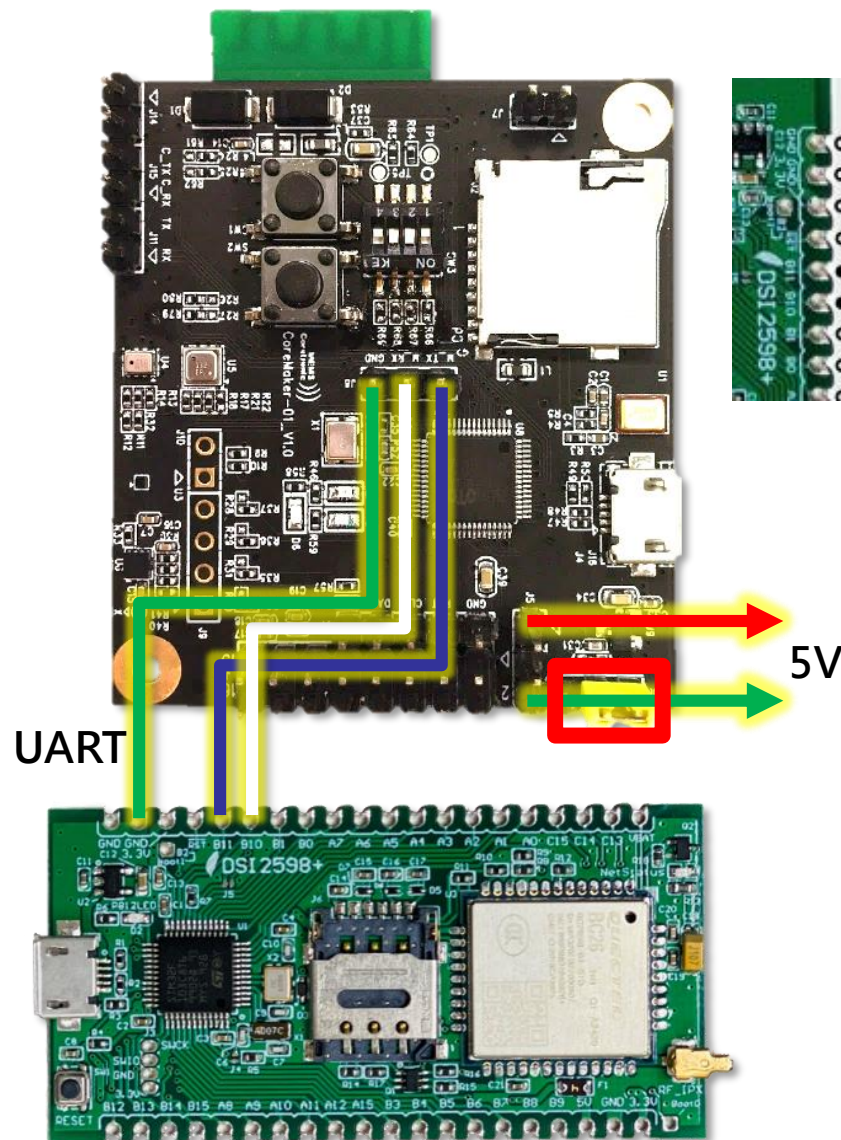


圖 2. 串接線路圖(額外供電)

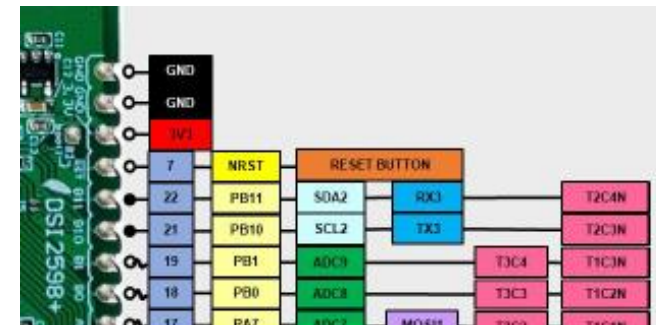


圖 3. 接點放大圖

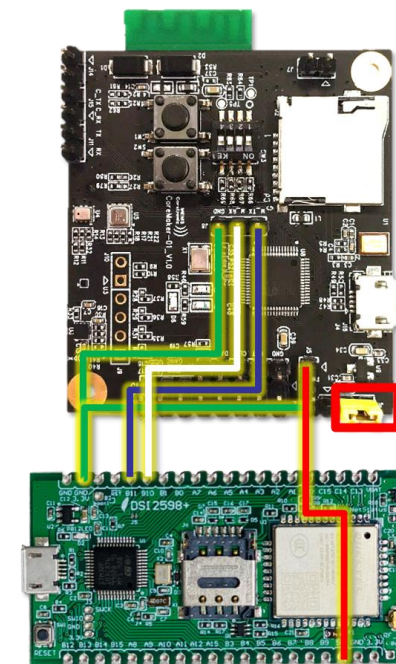


圖 4. 串接線路圖(共同供電)



2598+ 串接Coremaker-01

2598_Byte | Arduino 1.8.18

檔案 編輯 草稿碼 工具 說明



2598_Byte

```
void setup() {  
  Serial.begin(115200);  
  Serial3.begin(115200);  
}  
  
void loop() {  
  // read from port 0, send to port 1:  
  if (Serial.available()) {  
    int inByte = Serial.read();  
    Serial3.print(inByte);  
  }  
  // read from port 1, send to port 0:  
  if (Serial3.available()) {  
    int inByte = Serial3.read();  
    Serial.print(inByte);  
  }  
}
```

會收到來自Coremaker-01byte訊號，
需經ASCII轉換

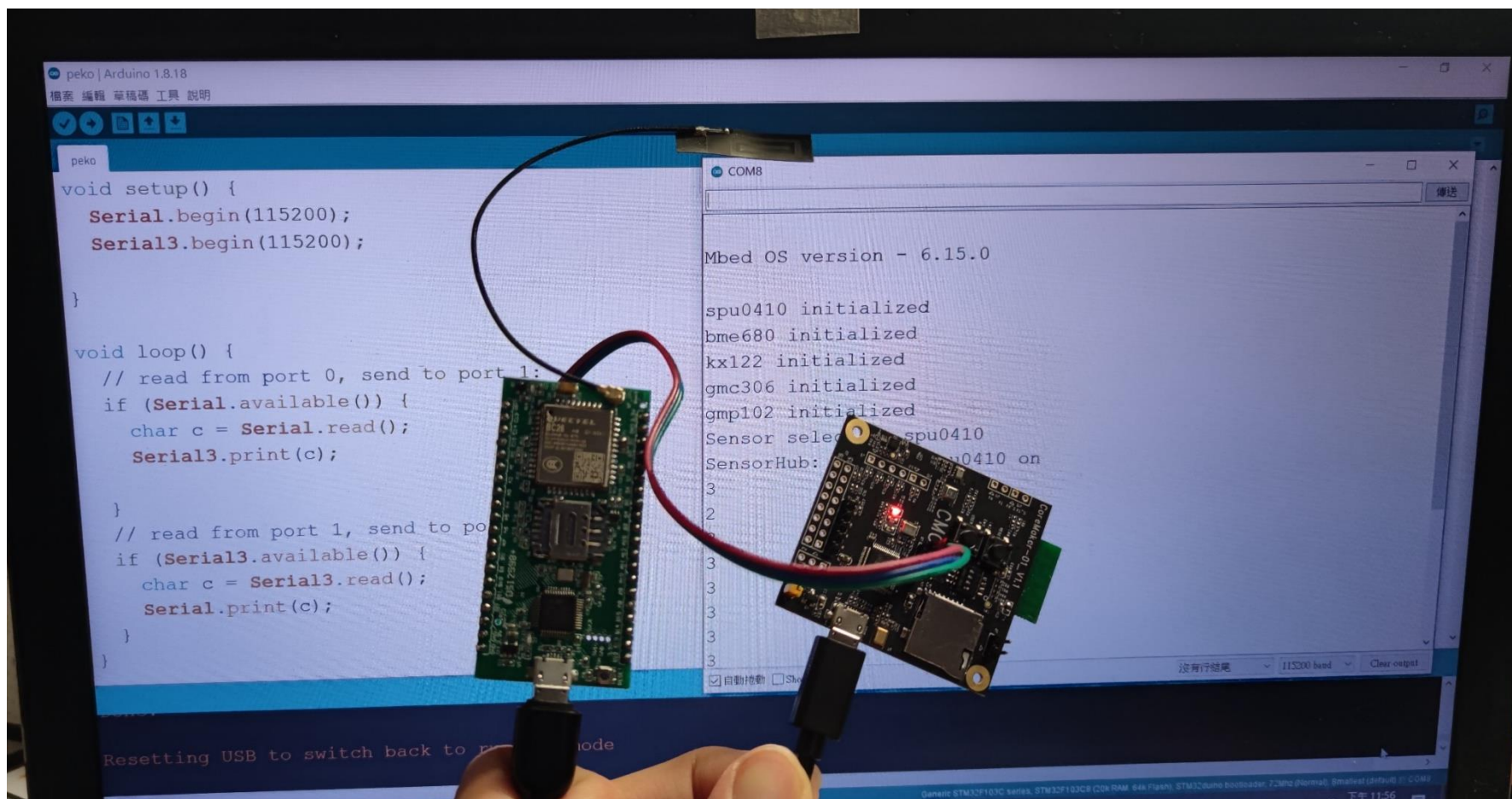
ASCII補充資料: <https://zh.wikipedia.org/wiki/ASCII>

程式位置:

https://github.com/wildman8606/Coremaker01withIoT/tree/main/2598/2598_Byte

2598+ 串接Coremaker-01

經轉換後應可以從2598+收到Coremaker-01資料



程式位置:

https://github.com/wildman8606/Coremaker01withIoT/tree/main/2598/2598_char

2598+ 串接Coremaker-01動作流程圖

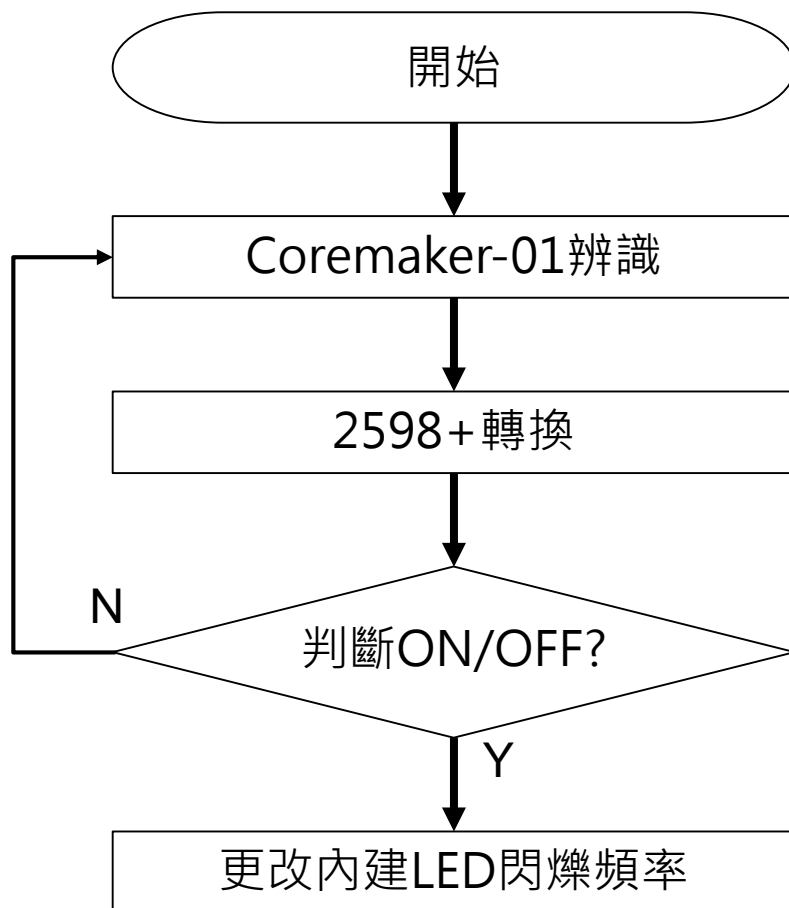


圖 1. 流程圖

Knowledge Pack information

Class Map:

1 - OFF

2 - ON

3 - StandBy

Resource Estimates

Estimated Memory Usage

SRAM Used: 4194 Bytes



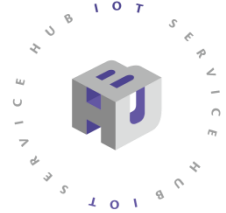
Stack Size: 1040 Bytes



Flash Used: 15661 Bytes



圖 2. 於SensiML網站輸出標籤說明



2598 + 串接Coremaker-01

收到ON/OFF訊號
後會改變閃爍頻率

2598_Coremaker01 | Arduino 1.8.18

檔案 編輯 草稿碼 工具 說明



2598_Coremaker01 \$

```
#define pin_led PB12
/*間隔 1000ms / 1秒*/
long interval = 10;
int ledState = LOW; // LED I/O 狀態
unsigned long previousMillis = 0;

void setup() {
  Serial.begin(9600);
  Serial3.begin(115200);
  pinMode(pin_led, OUTPUT);
}

void loop() {
  // read from port 3, send to port 0:
  if (Serial3.available()) {
    char c = Serial3.read();
    Serial.print(c);
    if (c=='1'){
      interval = 250;
    } else if (c=='2'){
      interval = 1000;
    }
  }

  unsigned long currentMillis = millis();
  if(currentMillis - previousMillis >= interval ){
    previousMillis = currentMillis;
    ledState == LOW ? ledState = HIGH : ledState = LOW;
    digitalWrite(pin_led, ledState);
  }
}
```

收到來自Coremaker-01byte訊號

判斷並更改閃爍時間參數

持續閃爍

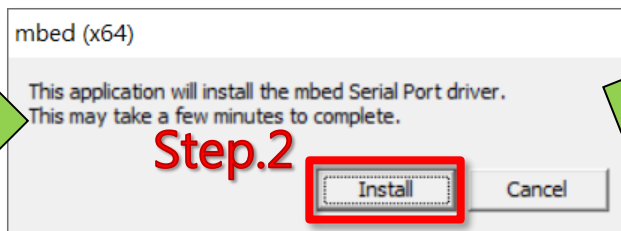
程式位置:

https://github.com/wildman8606/Coremaker01withIoT/tree/main/2598/2598_Coremaker01

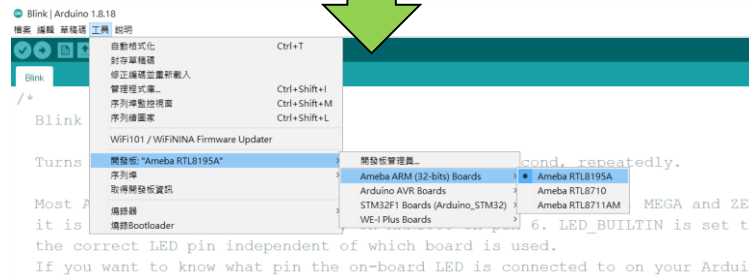
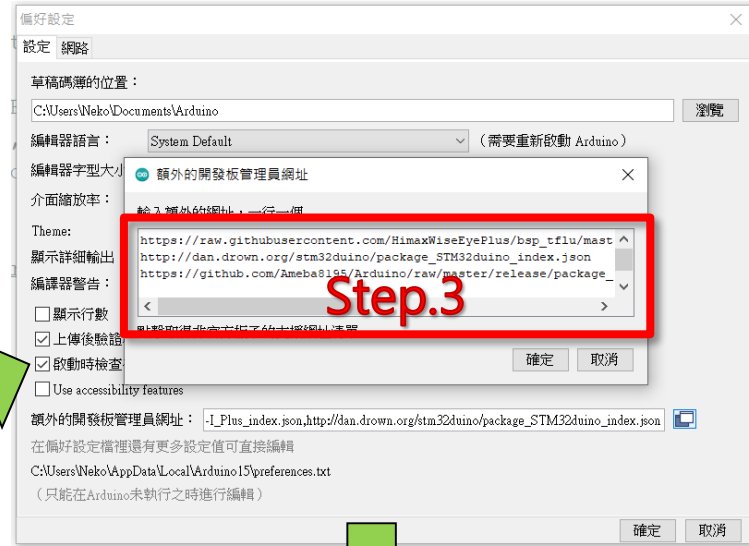
5168設定

<https://os.mbed.com/handbook/Windows-serial-configuration>

- Step.1**
1. Download the mbed Windows serial port driver
Download the installer to your PC, e.g. your desktop.
Download latest driver
 2. Run the installer



Step2. 安裝之



Step1. 前往arm mbed官網下載驅動程式

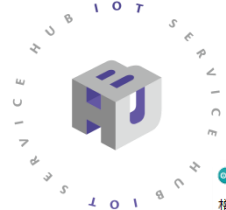
https://os.mbed.com/handbook/Windows-serial-configuration/media/downloads/drivers/mbedWinSerial_16466.exe

Step2. 開啟Arduino 程式, 選擇【檔案】→【偏好設定 (Preference)】, 設定額外的開發板管理員網址：

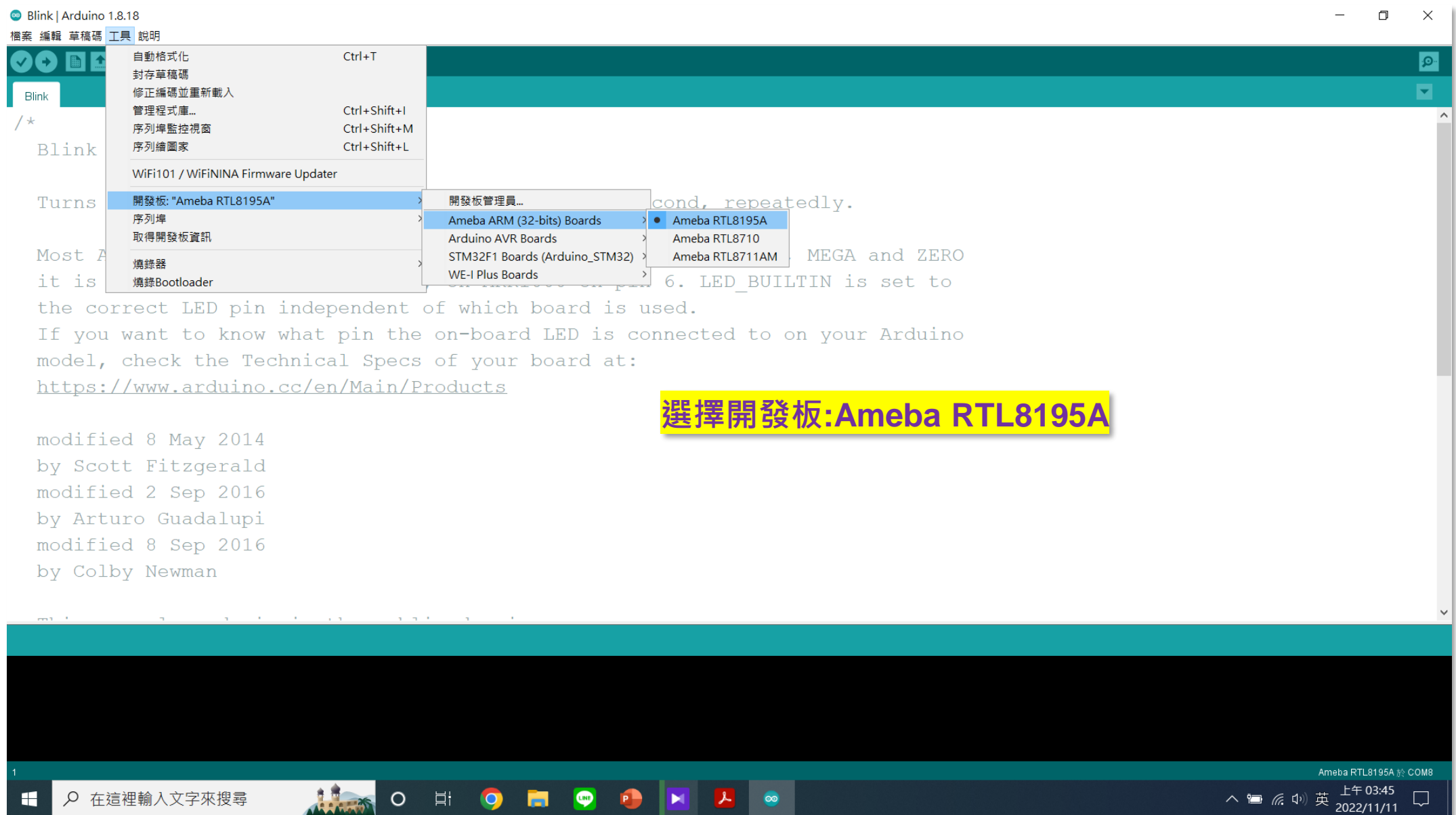
https://github.com/Ameba8195/Arduino/raw/master/release/package_realtek.com_ameba_index.json

Step3. 在Arduino 功能列選擇【工具】→【開發版】→【開發版管理者】，在上方輸入列輸入 Realtek, 在畫面上即可看到 Realtek Ameba Boards (32-bits ARM Cortex-M3) 等字樣, 可選擇最新版按下其右下角的安裝 (install)。

資策會官網補充資料: <https://forum.ideaschain.com.tw/forum/devtool/board.do?board=1>

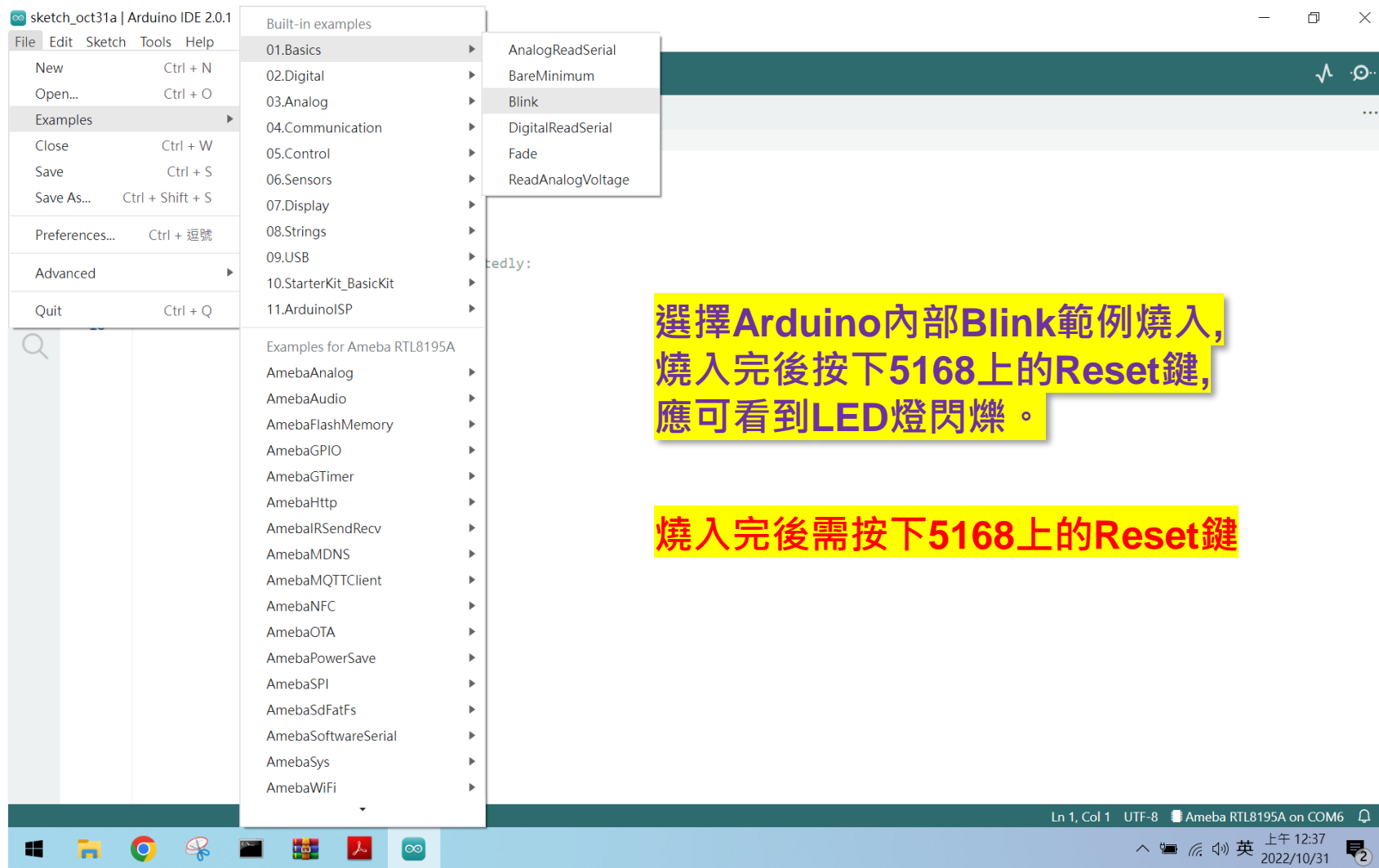


5168設定



5168 燒入測試

確認可操作5168燒入以及驗證該晶片運作正常



5168 串接Coremaker-01

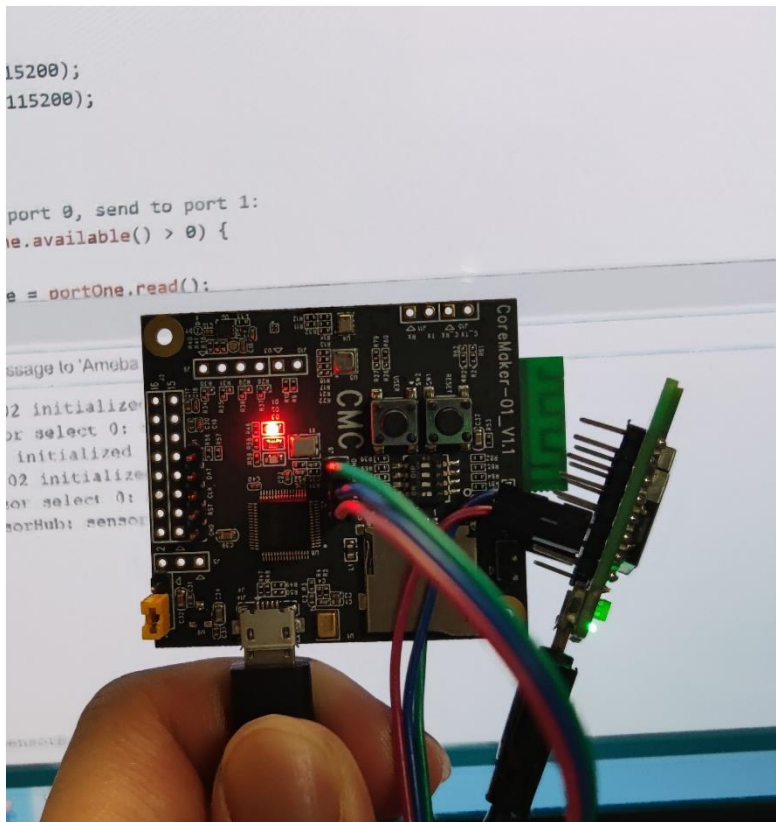


圖 1. 串接實體

DSI-2598+ RX3/D11	→	CoreMaker-01 TX
DSI-2598+ TX3/D10	→	CoreMaker-01 RX
DSI-2598+ GND	→	CoreMaker-01 GND

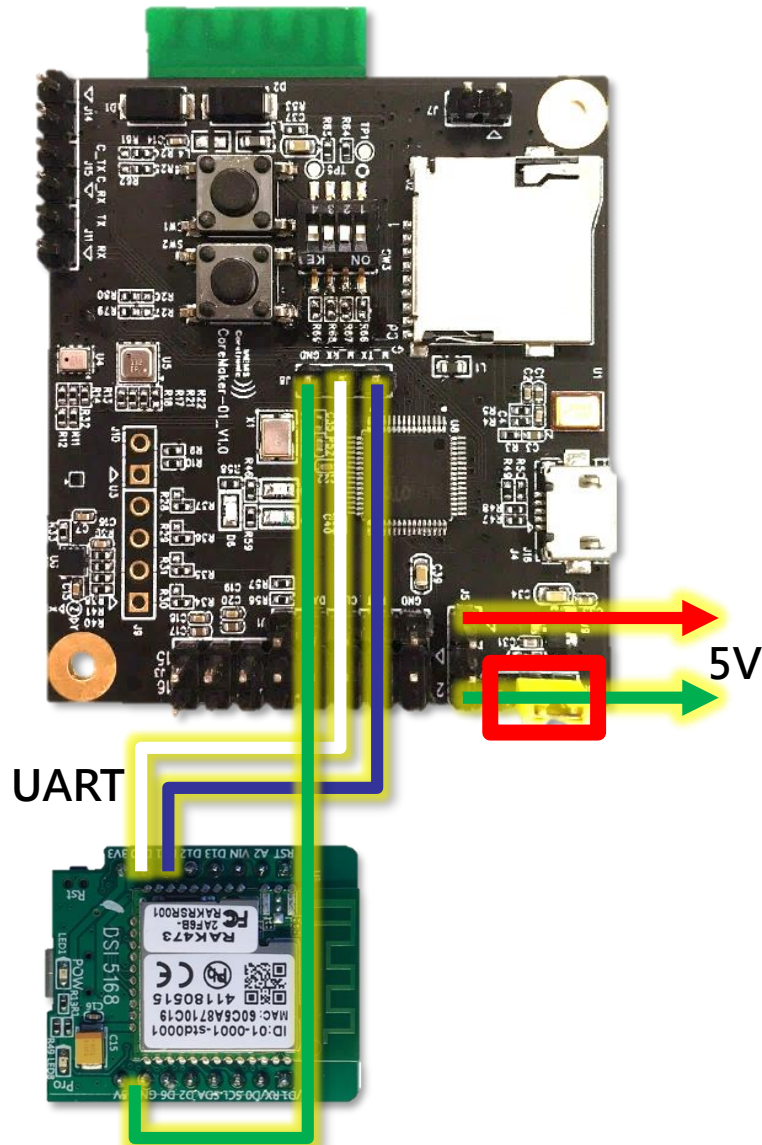


圖 2. 串接線路圖(額外供電)

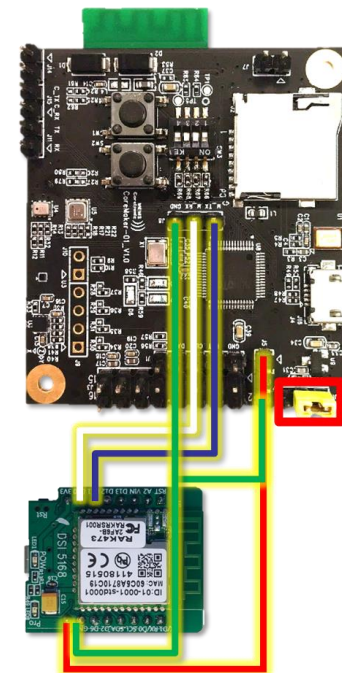


圖 3. 串接線路圖(共同供電)



5168 串接Coremaker-01

5168_Serial2 | Arduino 1.8.18

檔案 編輯 草稿碼 工具 說明



5168_Serial2 \$

```
#include <SoftwareSerial.h>
// software serial #1: RX = digital pin 10, TX = digital pin 11

SoftwareSerial SoftwareSerial2(10, 11);
```

```
void setup() {
  Serial.begin(9600);
  SoftwareSerial2.begin(115200);
}

void loop() {
  while (SoftwareSerial2.available() > 0) {
    char inByte = SoftwareSerial2.read();
    Serial.write(inByte);
  }
}
```

由於5168預設只有1個UART,
故使用<SoftwareSerial.h>函式庫,
可新增多組做通訊

儲存完畢

程式位置:

https://github.com/wildman8606/Coremaker01withIoT/tree/main/5168/5168_Serial2

系統架構圖

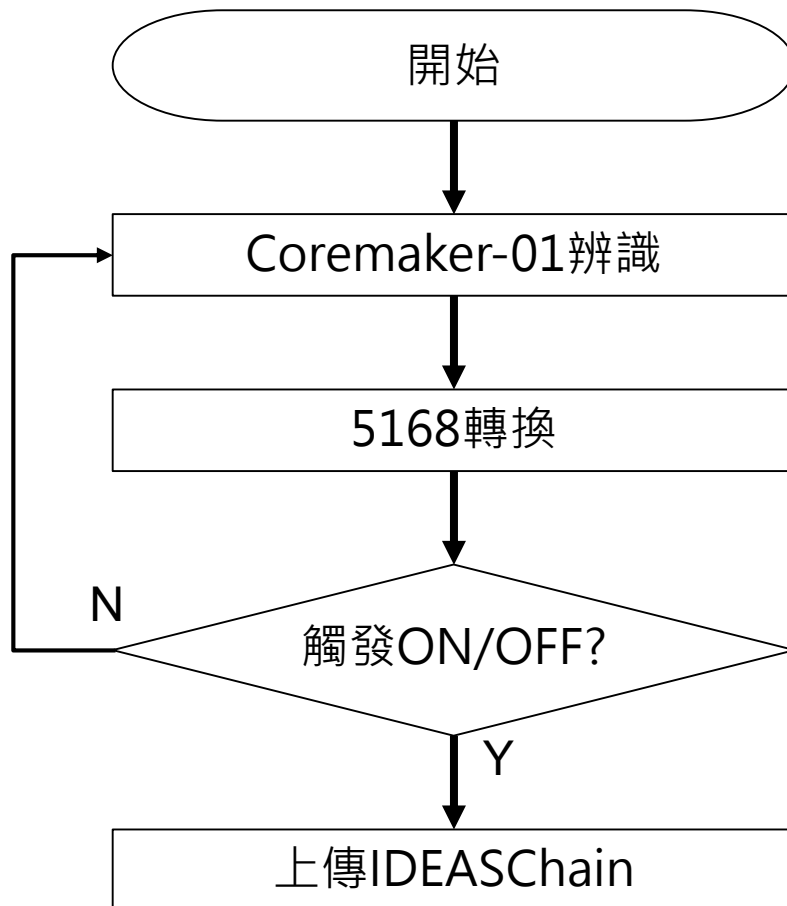


圖 1. 流程圖

Knowledge Pack information

Class Map:

1 - OFF

2 - ON

3 - StandBy

Resource Estimates

Estimated Memory Usage

SRAM Used: 4194 Bytes



Stack Size: 1040 Bytes



Flash Used: 15661 Bytes



圖 2. 於SensiML網站輸出標籤說明

程式位置:

https://github.com/wildman8606/Coremaker01withIoT/tree/main/5168/5168_MQTT

MQTT ?

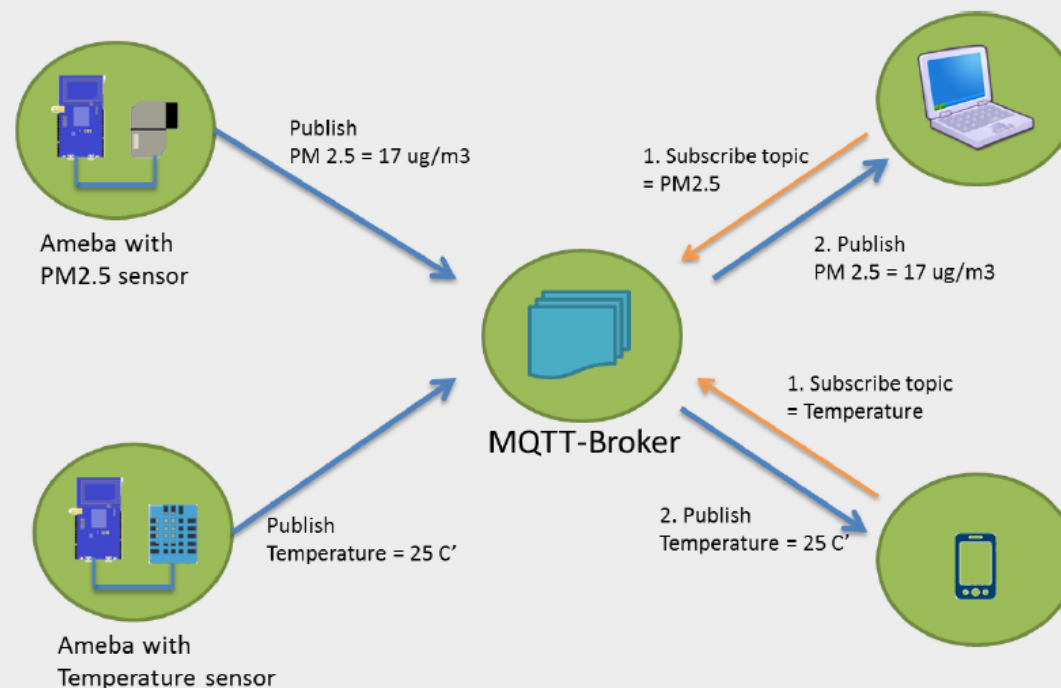
國產IC開發套件 DSI 5168

MQTT基本架構

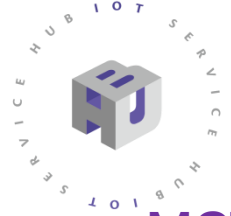
使用MQTT server測試工具，目的是為了要測試你所使用的server，可以順利地透過你所設定的topic來subscribe和publish你所要傳送的payload。

Subscribe
平台向Server端訂閱MQTT訊息

Publish
Sensor透過DSI5168發布MQTT訊息，訊息格式需為Json格式



(圖片來源：amebaiot.com)



MQTT 測試

MQTT的測試工具，如MQTTBox、MQTTLens等

<https://chrome.google.com/webstore/detail/mqttbox/kaajoficamnjjhkeomgfljpicifbkaf?hl=zh-TW>

Clientname: 任意指定

Username: 平台的存取權杖(可參考教學之page23)

MQTT Client id: 指定一個獨一無二的名稱

Host: <https://ideaschain.com.tw>

(使用IDEAS Chain作為Server及數據平台)

Topic: 為IDEAS Chain所指定之路徑

v1/devices/me/telemetry

Payload: 需為Json格式

設定完成後先按下Subscribe再按下Publish，若有成功訂閱到payload訊息，表示成功透過server收發MQTT訊息



5168 串接 Coremaker01 再上IDEASChain



<https://iiot.ideaschain.com.tw/home>

The screenshot shows the IDEASChain website interface. At the top, there's a navigation bar with links for 數據平台 (Data Platform), 論壇 (Forum), 應用案例 (Application Cases), 開發工具 (Development Tools), 技術支援 (Technical Support), and 登出 (Logout). The main content area features the IDEASChain logo and the title "物聯數據創新應用分享" (IoT Data Innovation Application Sharing). Below this, it says "IDEAS Chain 簡單、快速導入雲端數據應用" (IDEAS Chain Simple, Fast Import Cloud Data Application). A prominent blue button labeled "加入我們" (Join Us) is visible. The background of the page is a large illustration depicting a smart home environment with various IoT devices like a smartwatch, a house, a washing machine, a refrigerator, a laptop, and a smartphone, all interconnected by a network of dashed lines. A "Scroll" label with a downward arrow is positioned near the bottom left of the illustration.



5168 串接 Coremaker01 再上IDEASChain



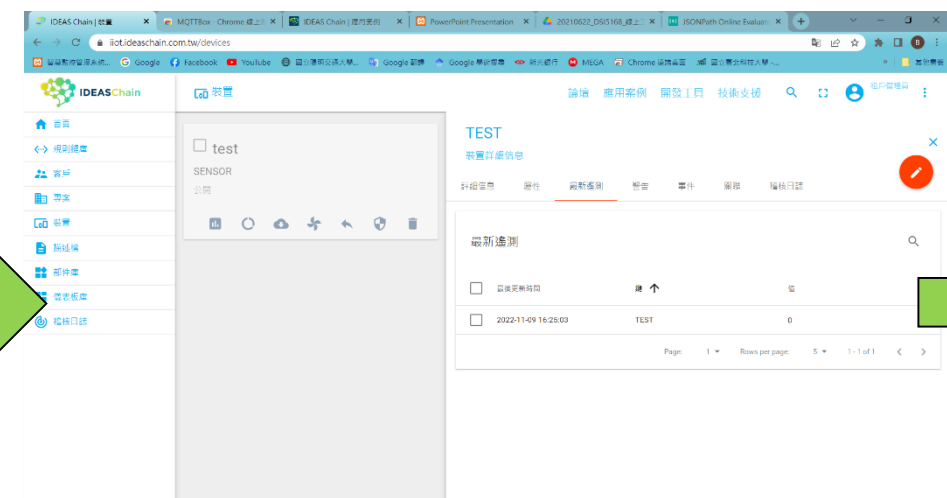
<https://iiot.ideaschain.com.tw/home>

Step.1





5168 串接 Coremaker01 再上IDEASChain





5168 串接 Coremaker01 再上IDEASChain

Step.7

MQTT CLIENT SETTINGS

MQTT Client Name: miku

MQTT Client Id: d0fe9db0-3d0c-40d4-b211-2c15a8060735

Protocol: mqtt / tcp

Host: iot.ideaschain.com.tw

Username: 5qmCDay5NMRVlp6xeK8Y

Password: Password

Reconnect Period (milliseconds): 1000

Connect Timeout (milliseconds): 30000

Will - Topic: Will - Topic

Will - QoS: 0 - Almost Once

Append timestamp to MQTT client id? ☒ Yes

Clean Session? ☒ Yes

Reschedule Pings? ☒ Yes

KeepAlive (seconds): 10

Will - Retain: ☐ No

Broker is MQTT v3.1.1 compliant? ☒ Yes

Auto connect on app launch? ☒ Yes

Queue outgoing QoS zero messages? ☒ Yes

Will - Payload: {test:0}

Save

Delete

先使用MQTTBox連線測試,
再換5168。

Step.8

MQTT CLIENT SETTINGS

Topic to publish: v1/devices/me/telemetry

QoS: 0 - Almost Once

Retain: ☐

Payload Type: Strings / JSON / XML / Characters

Payload: {test:0}

Publish

Topic to subscribe: v1/devices/me/telemetry

QoS: 0 - Almost Once

Subscribe



5168 串接 Coremaker01 再上IDEASChain

The screenshot displays the IDEASChain web interface. On the left is a sidebar with navigation links: 首頁, 規則鏈庫, 客戶, 專案, 裝置, 描述檔, 部件庫, 儀表板庫, and 稽核日誌. The main area shows a dashboard titled 'test' with a 'New State Chart' graph. The graph has a y-axis with 'On' and 'Off' states and an x-axis with timestamps from 16:27:30 to 16:28:20. A blue line labeled 'TEST' shows the device state over time. On the right, an MQTT configuration modal is open, showing settings for publishing and subscribing to the topic 'v1/devices/me/telemetry'. The 'Publish' section includes fields for Topic, QoS (set to 0 - Almost Once), Retain (unchecked), Payload Type (Strings / JSON / XML / Characters), and Payload ({"TEST": "1"}). A 'Publish' button is visible. The 'Subscribe' section also shows the same topic and QoS, with a 'Subscribe' button.

程式位置:

https://github.com/wildman8606/Coremaker01withIoT/tree/main/5168/5168_MQTT

5168 串接 LineNotify



程式位置:

https://github.com/wildman8606/Coremaker01withIoT/tree/main/5168/5168_line

5168 串接 LineNotify

5168_line | Arduino 1.8.18

檔案 編輯 草稿碼 工具 說明

```
/*
  IDEASCHAIN MQTT example for DSI5168
*/
#include <SoftwareSerial.h>
// software serial #1: RX = digital pin 10, TX = digital pin 11
SoftwareSerial SoftwareSerial2(10, 11);

#include <WiFi.h>
// WPA/WPA2 SSID and password
char ssid[] = "您的WiFi帳號"; // your network SSID (name)
char pass[] = "您的WiFi密碼"; // your network password
String Linetoken = "您的權杖";

//至
int status = WL_IDLE_STATUS; // the Wifi radio's status
char host[] = "notify-api.line.me";

WiFiSSLClient client;

void setup()
{
  Serial.begin(9600);
  SoftwareSerial2.begin(115200);
}
```

儲存完畢

5168_line | Arduino 1.8.18

檔案 編輯 草稿碼 工具 說明

```
void loop()
{
  String message = "PEKO";
  if (client.connect(host, 443)) {
    int LEN = message.length();
    //1. 傳遞網站
    String url = "/api/notify"; //Line API網址
    client.println("POST " + url + " HTTP/1.1");
    client.print("Host: "); client.println(host); //Line API網站
    //2. 資料表頭
    client.print("Authorization: Bearer "); client.println(Linetoken);
    //3. 內容格式
    client.println("Content-Type: application/x-www-form-urlencoded");
    //4. 資料內容
    client.print("Content-Length: "); client.println( String((LEN + 8)) ); //訊息長度
    client.println();
    client.print("message="); client.println(message); //訊息內容
    //等候回應
    delay(2000);
    String response = client.readString();
    //顯示傳遞結果
    Serial.println(response);
    client.stop(); //斷線
  }
  else {
    //傳送失敗
    Serial.println("connected fail");
  }
  delay(5000);
}
```

儲存完畢

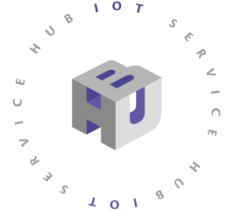
Step.6

每5秒會發送文字，
可再串接Coremaker-01

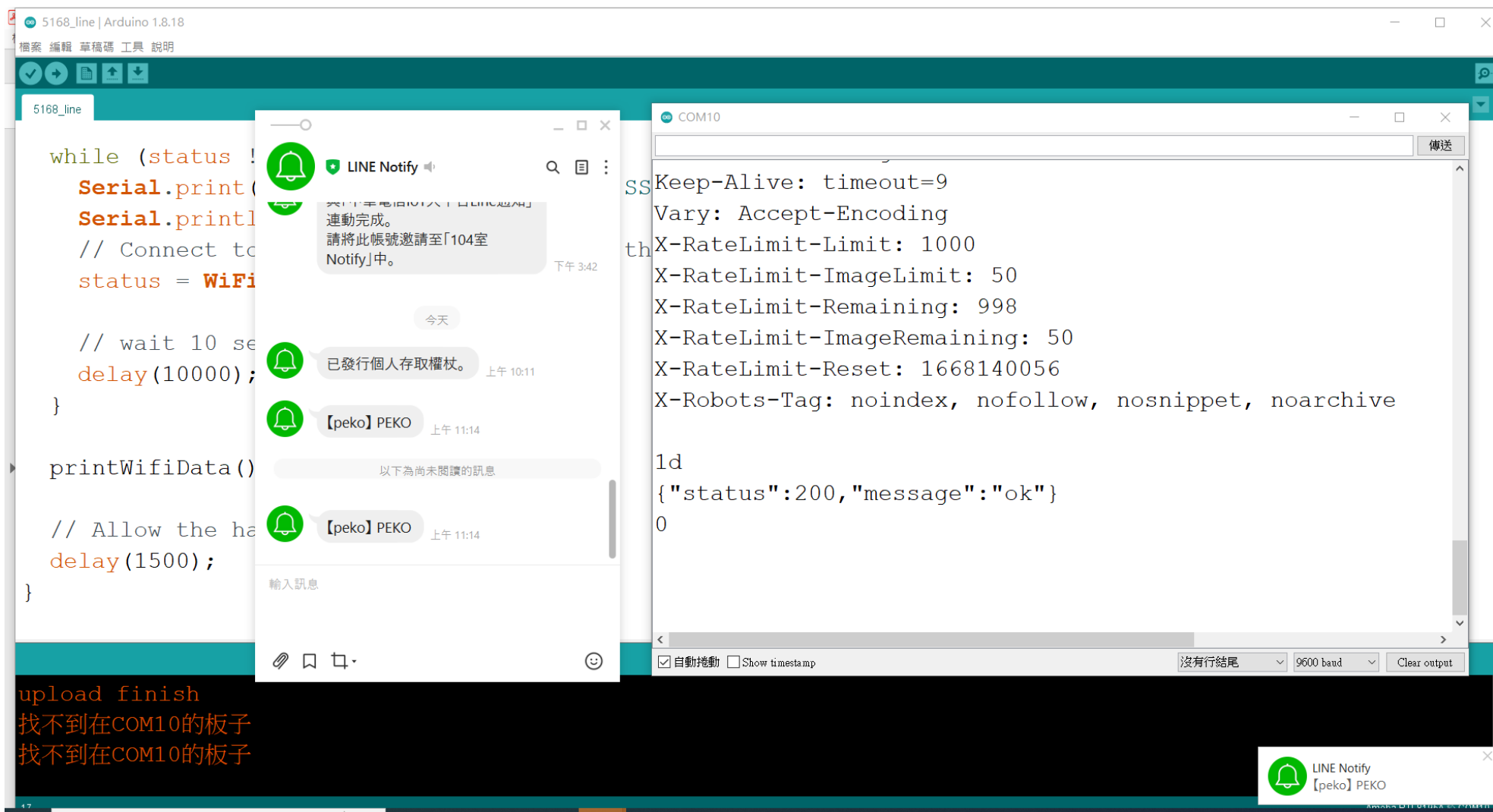
Step.5

程式位置:

https://github.com/wildman8606/Coremaker01withIoT/tree/main/5168/5168_line



5168 串接 Coremaker-01 及LineNotify



程式位置:

https://github.com/wildman8606/Coremaker01withIoT/tree/main/5168/5168_line



補充資料

需安裝軟體

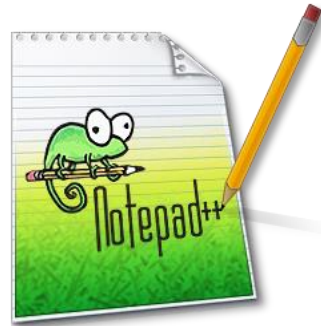


20分鐘

Arduino



Notepad++



Python



Tera Term



必需性:

必要

可替換

必要

可替換



說明:

國產晶片用
Arduino IDE
燒入、編譯
方便上手

慣用程式編輯器
都可，這較小、
輕量、好用。

晶片編譯需要

UART顯示結果，
可擇自己喜歡。

軟體安裝引導(Arduino)



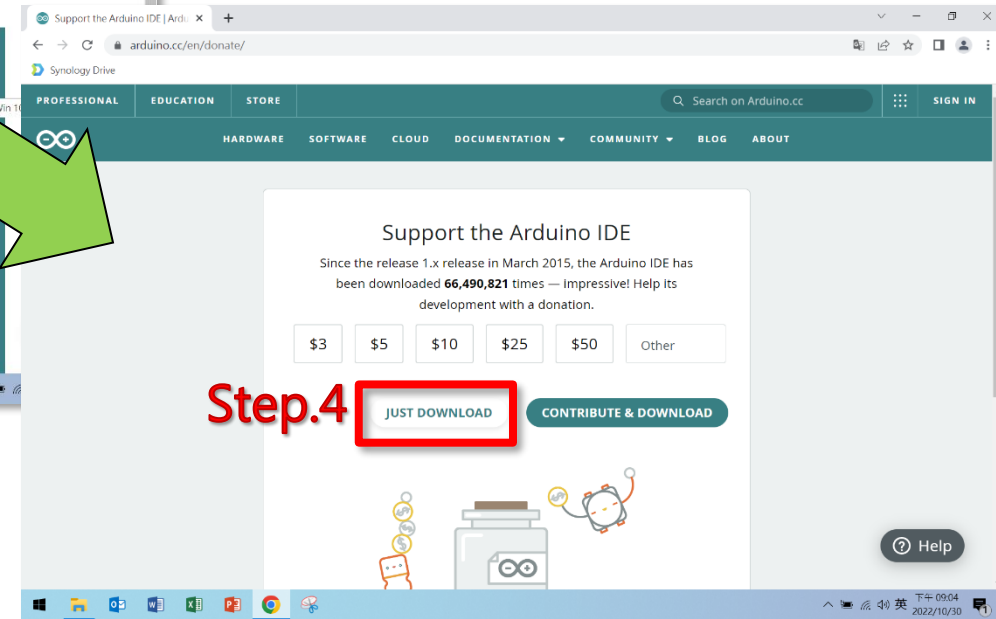
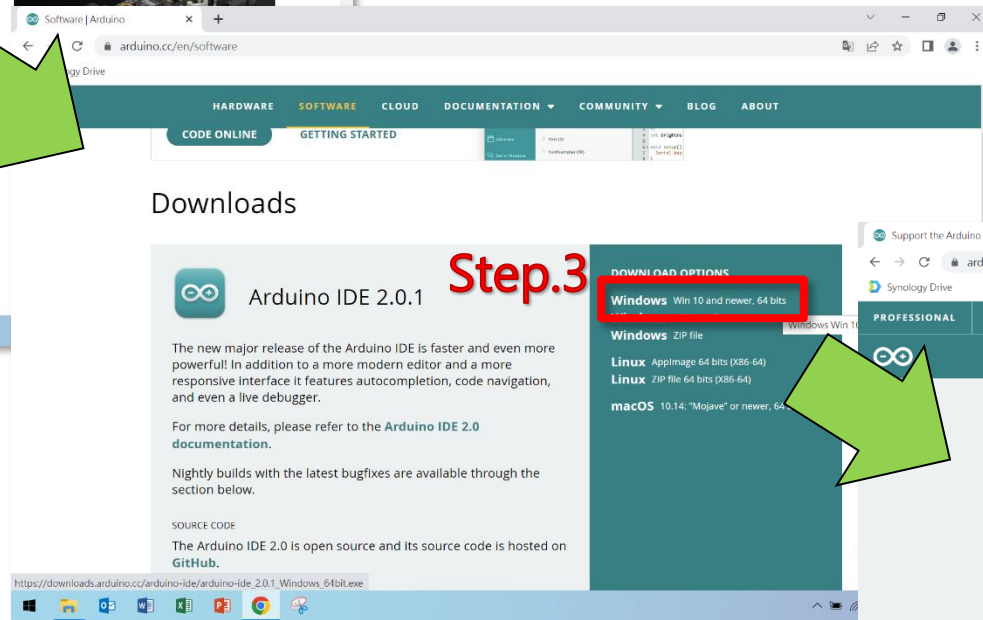
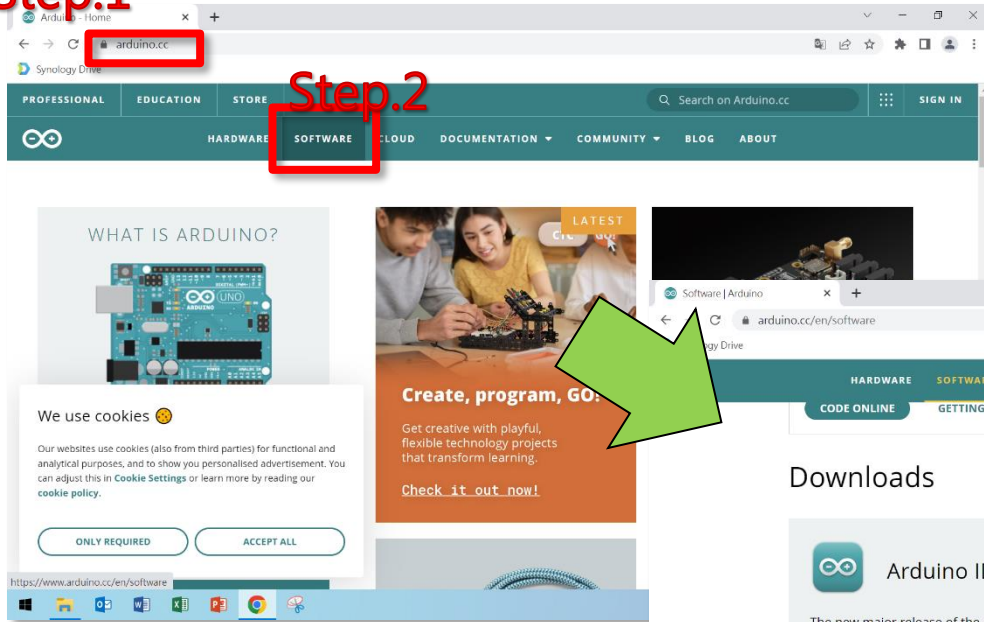
<https://www.arduino.cc/>

Step.1

Step.2

Step.3

Step.4



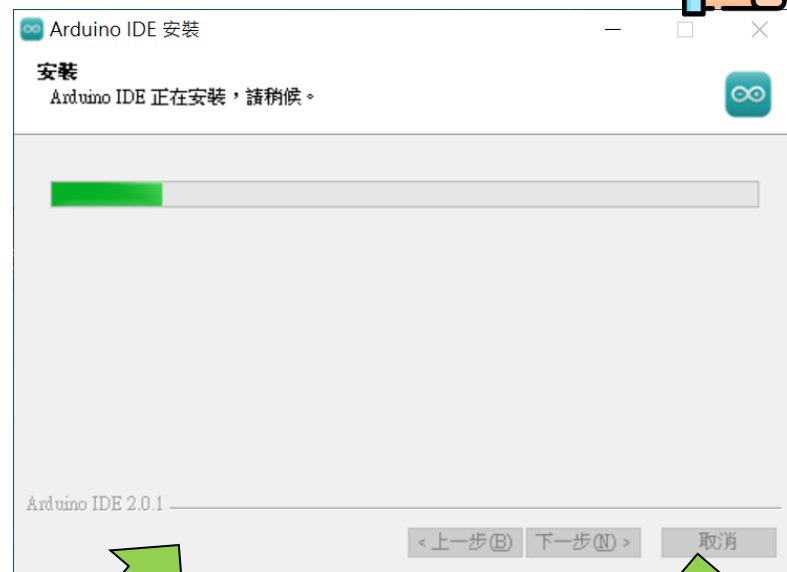
軟體安裝引導(Arduino)



<https://www.arduino.cc/>



Step.5

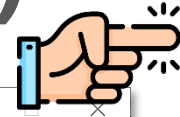


Step.6

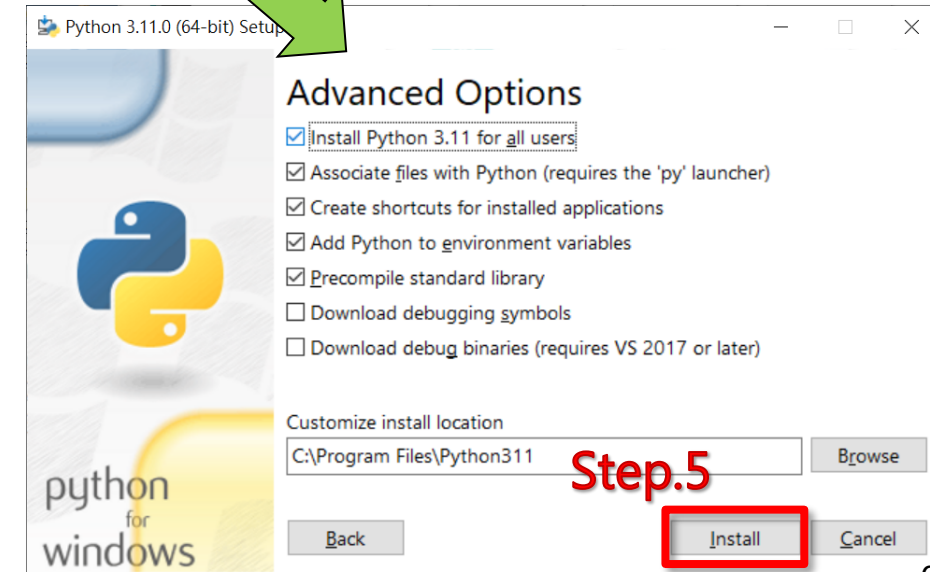
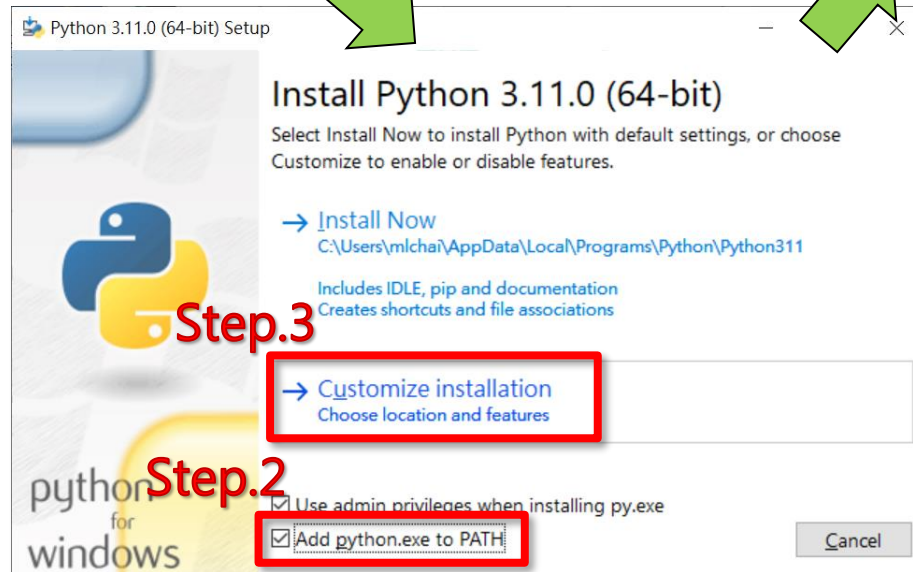
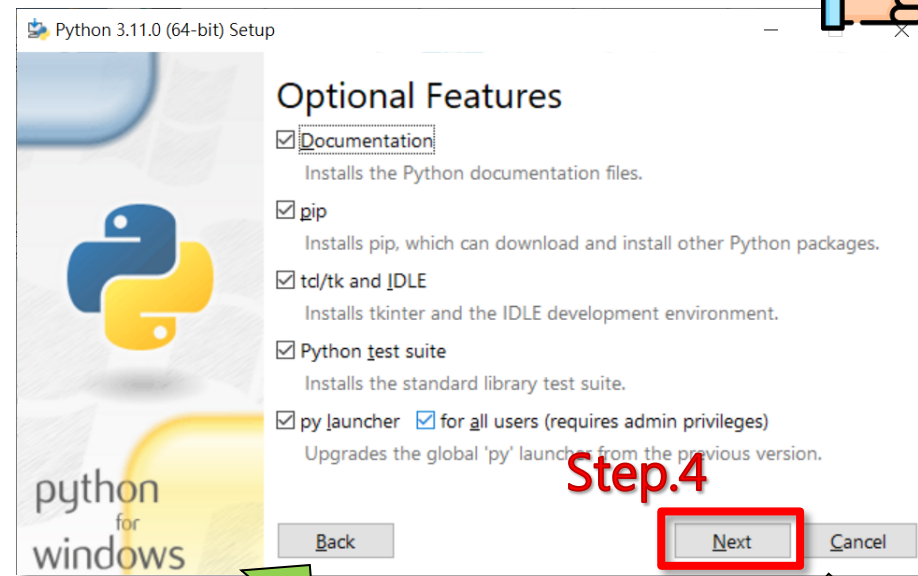
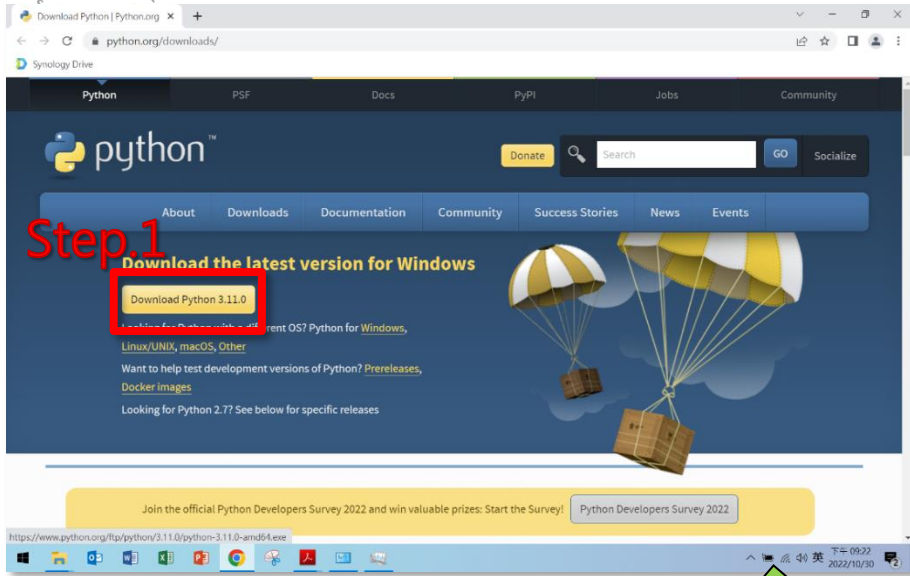


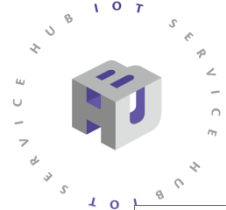
Step.7

軟體安裝引導(Python)



<https://www.python.org/>





軟體安裝引導(Python)



<https://www.python.org/>

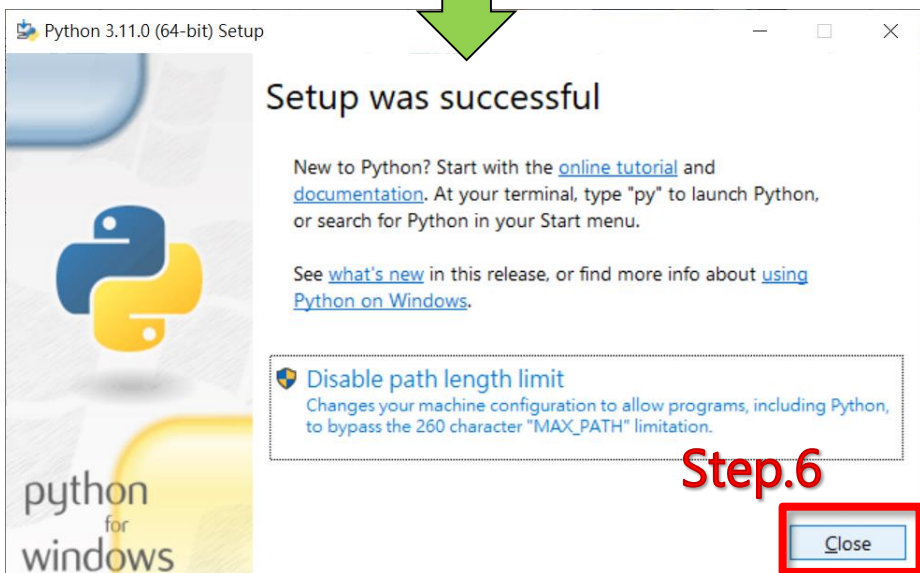
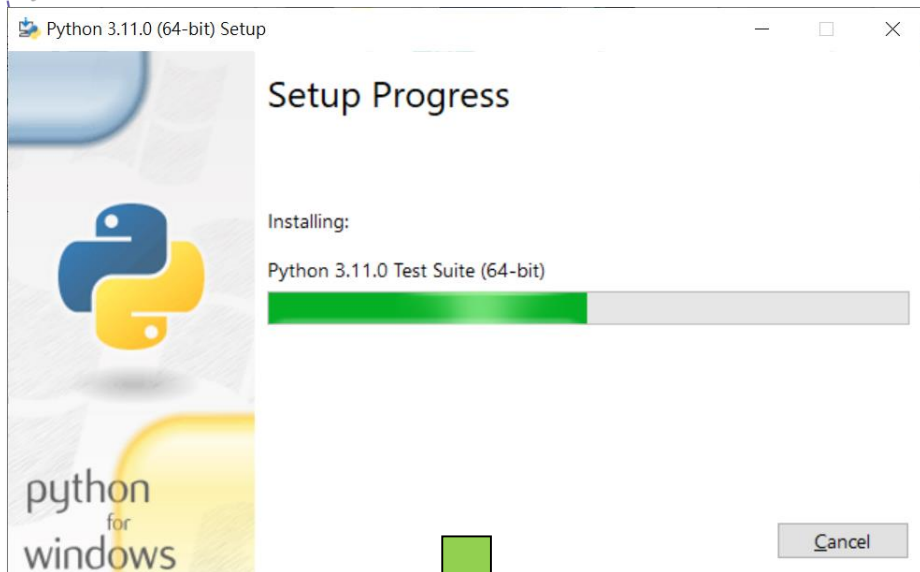


圖3 輸入'CMD'或用其他方法打開命令提示字元

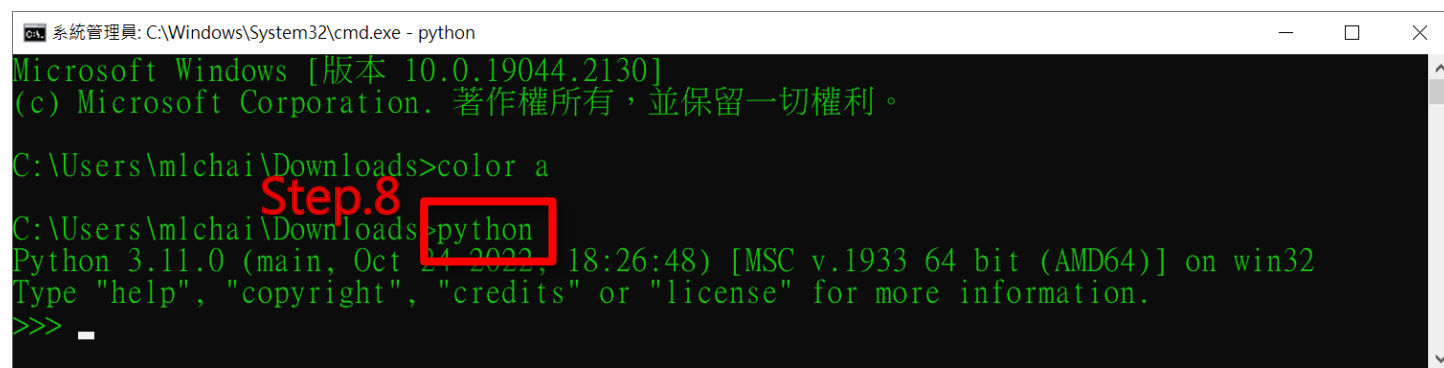


圖4 輸入'python'確認環境變數有設定成功



需安裝軟體二



25分鐘

CMake



git



**GNU Arm
Embedded
Toolchain**

GNU Arm
Embedded
Toolchain

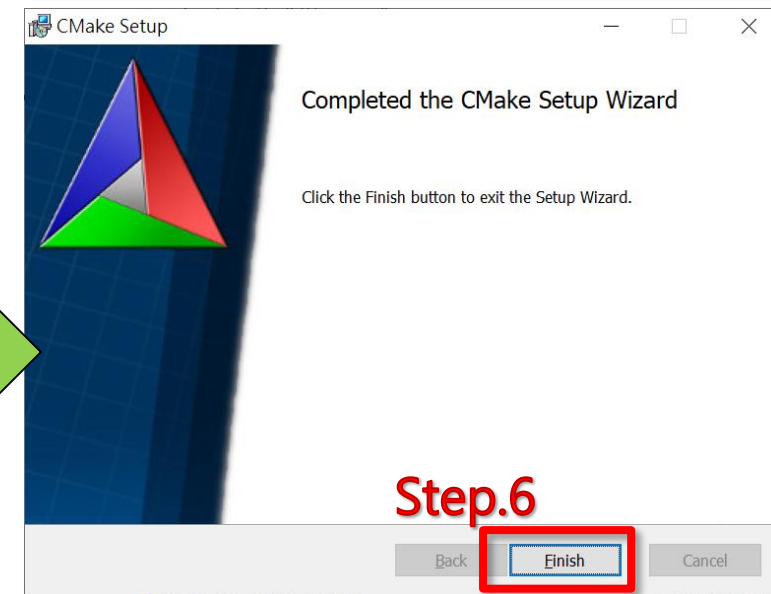
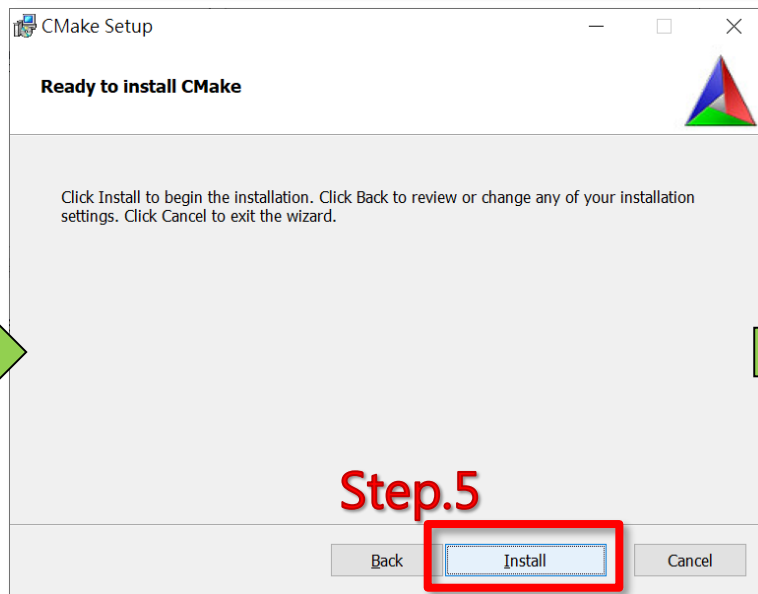
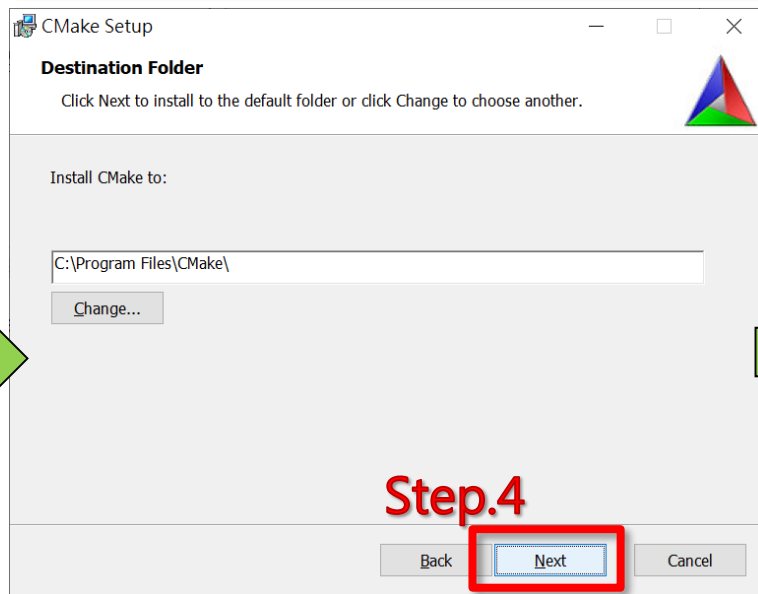
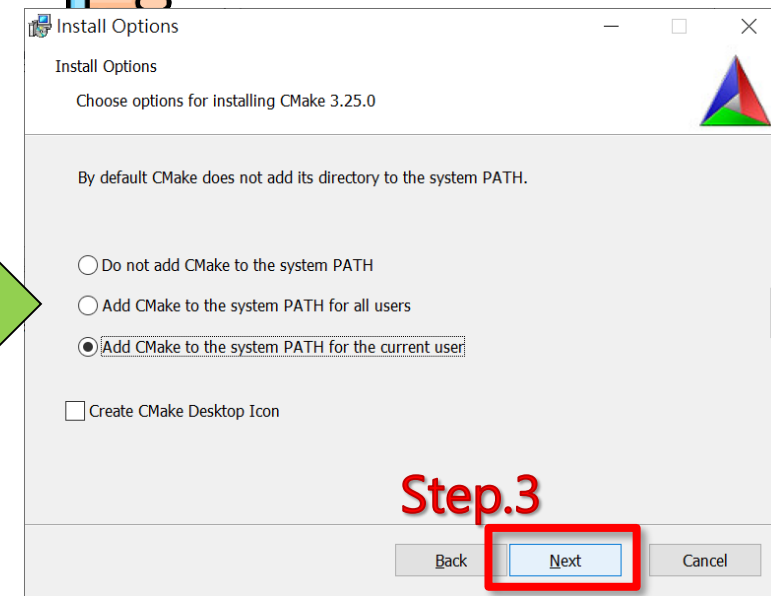
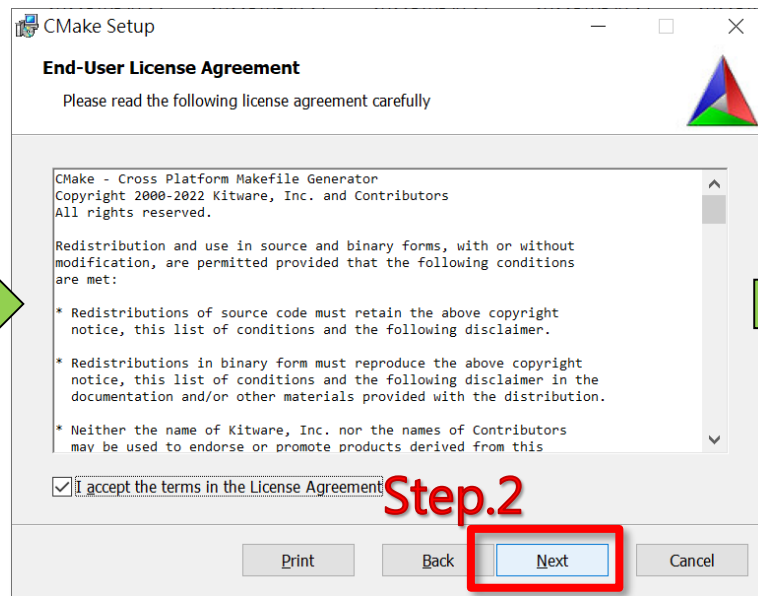
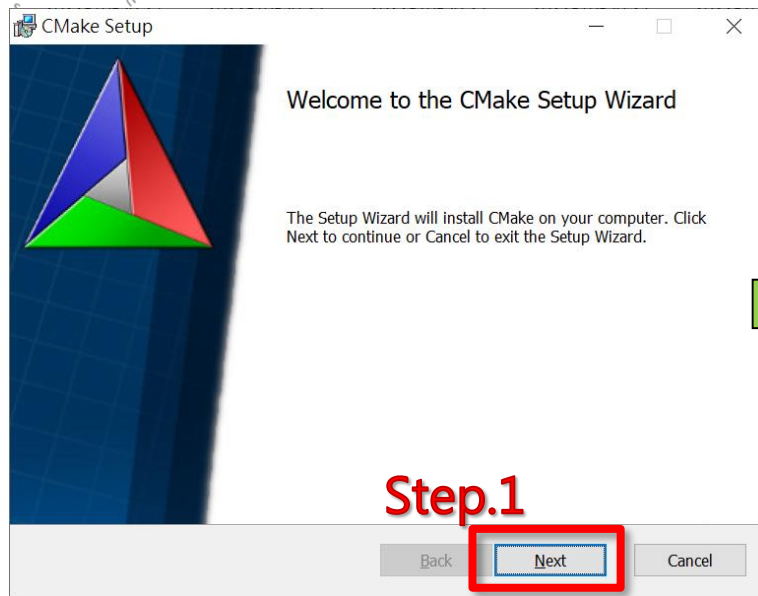
**SensiML Data
Capture Lab**



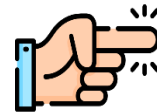
軟體安裝引導(CMake)



<https://cmake.org/download/>

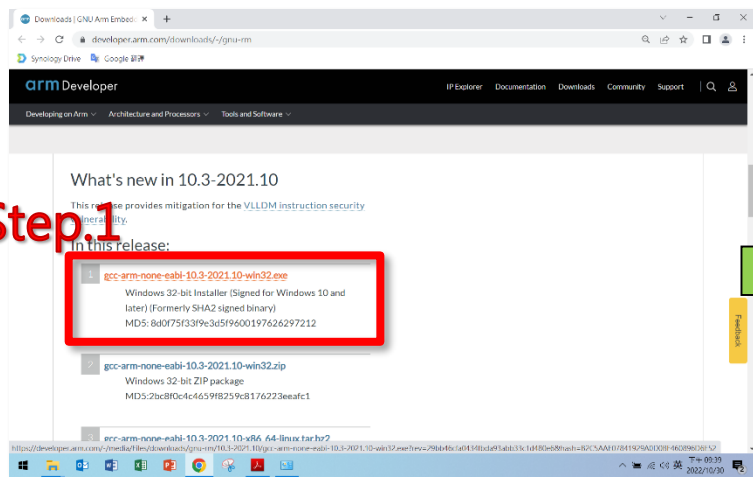


軟體安裝引導(GNU Arm Embedded Toolchain)



<https://developer.arm.com/downloads/-/gnu-rm/>

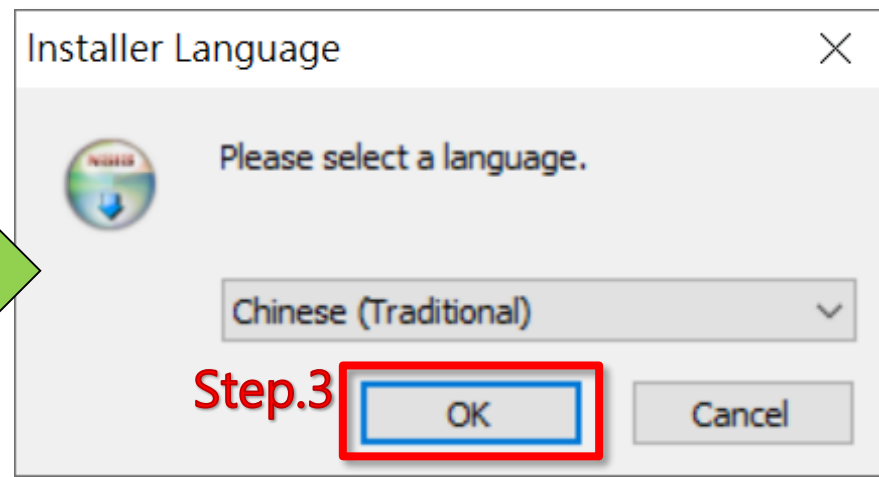
Step.1



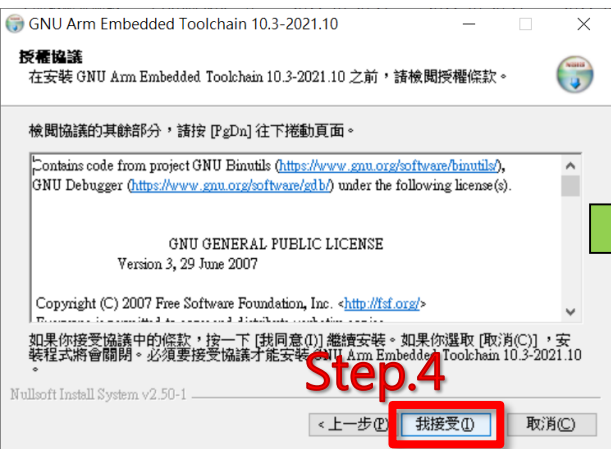
Step.2



Step.3



Step.4



Step.5



Step.7



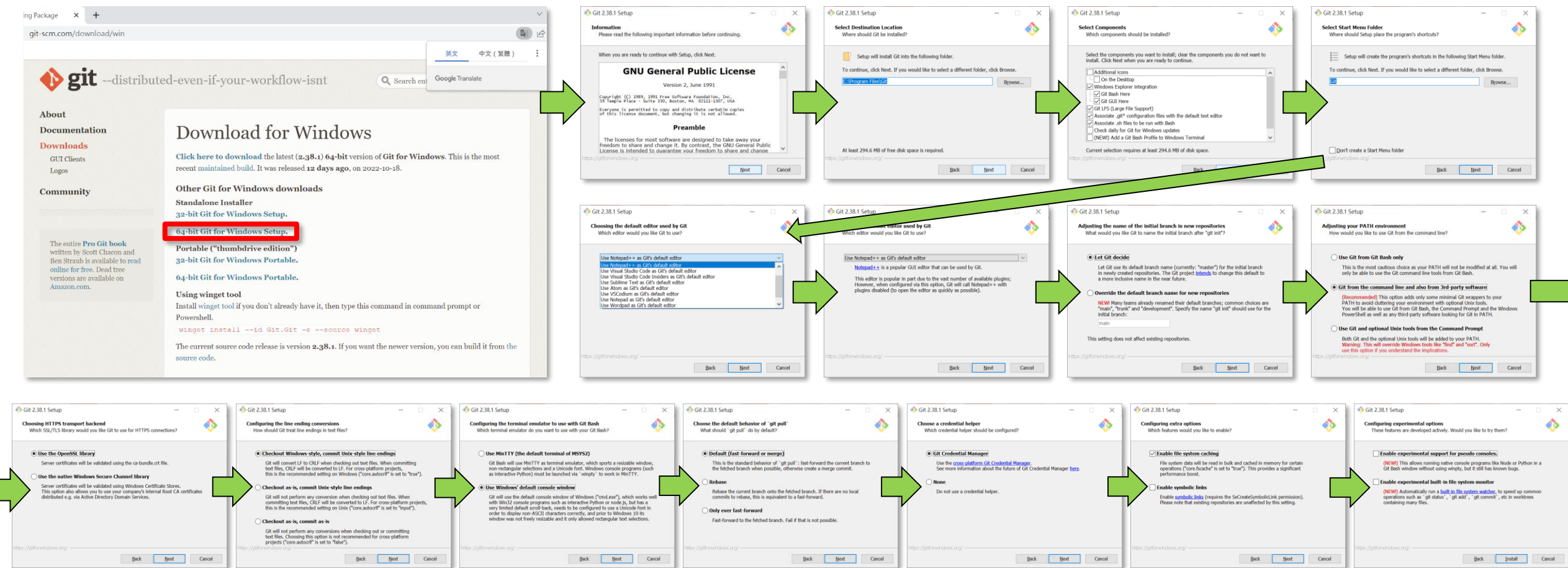
安裝完後需要重新啟動

軟體安裝引導(git)

如有其他慣用git下載方法可以免安裝這個



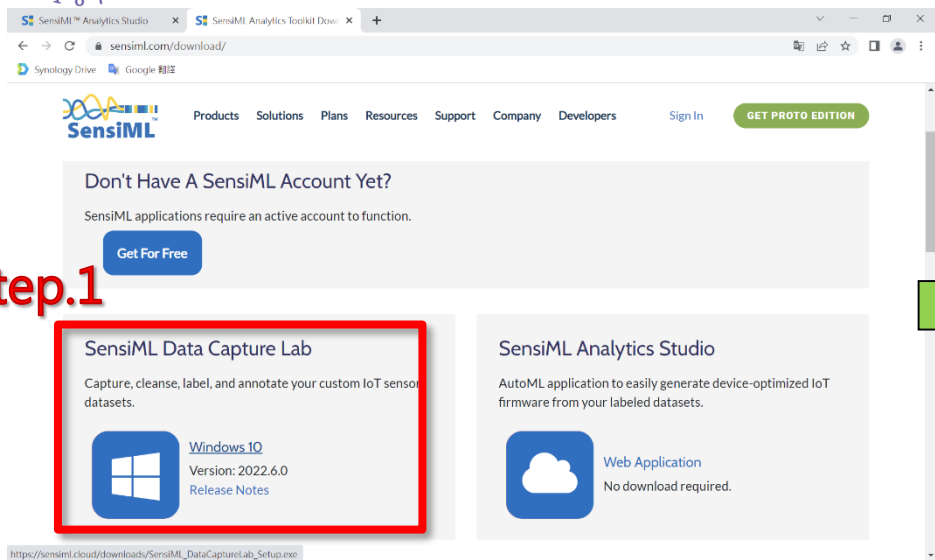
<https://git-scm.com/downloads>



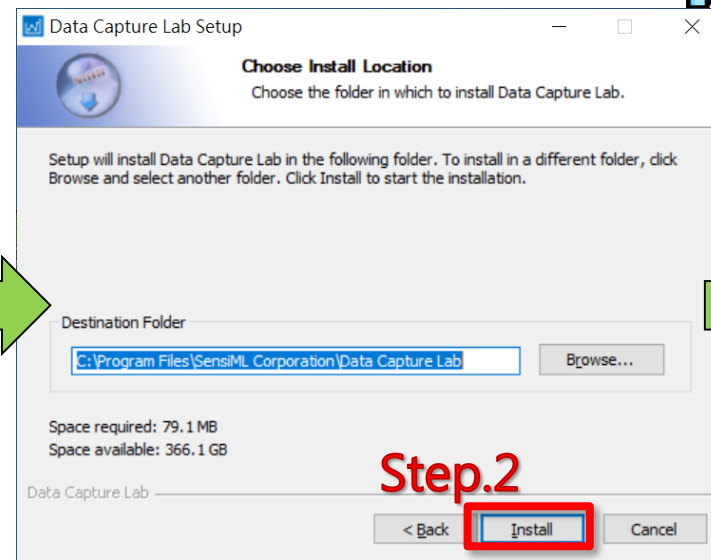
軟體安裝引導(SensiML Data Capture Lab)

<https://sensiml.com/download/>

Step.1



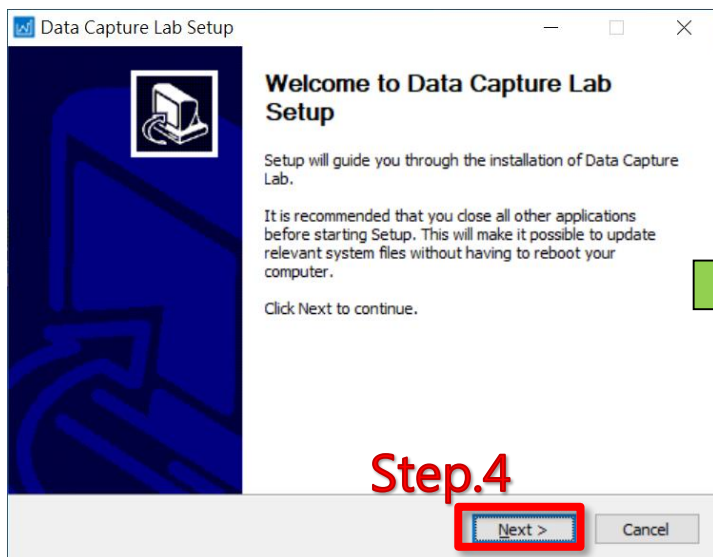
Step.2



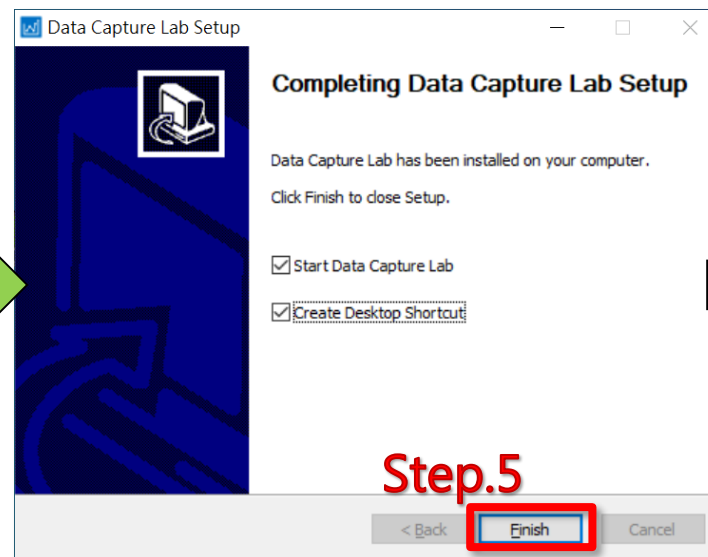
Step.3



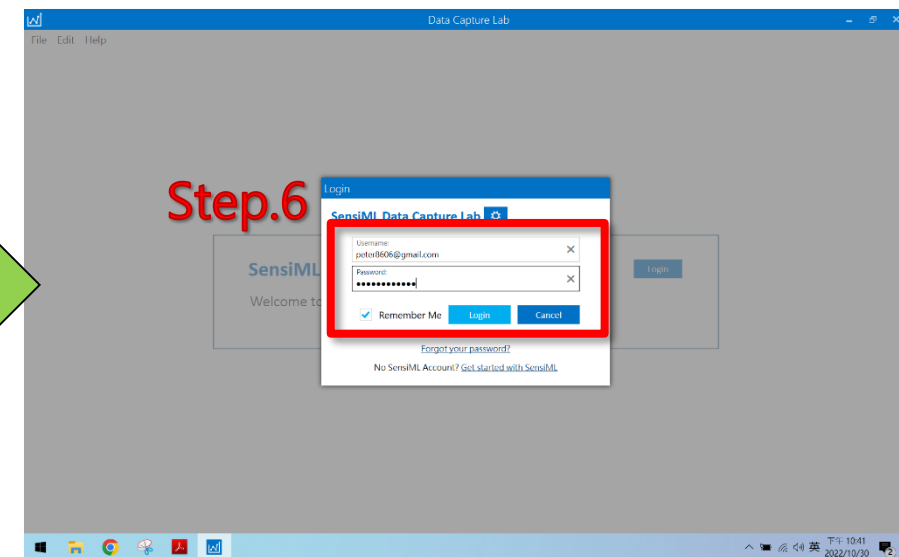
Step.4



Step.5



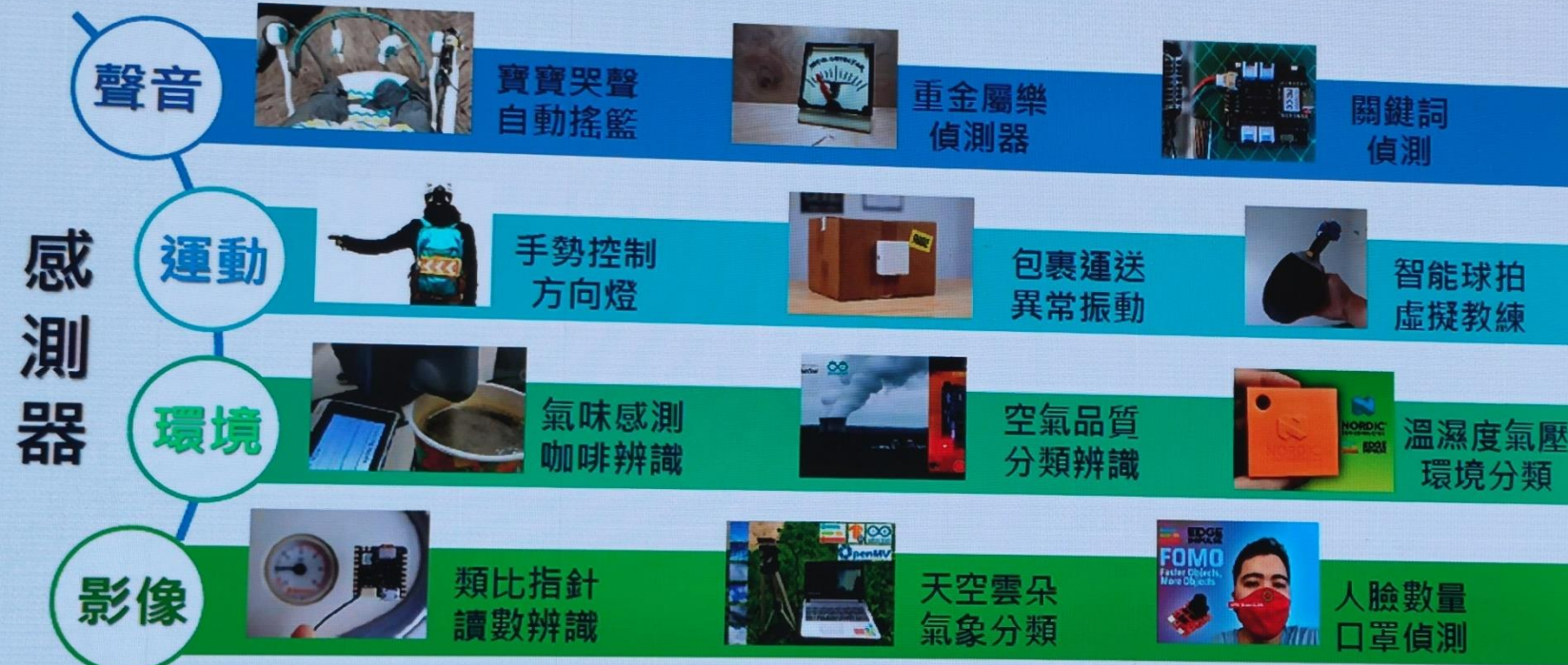
Step.6



tinyML 運用案例

歐尼克斯實境互動工作室(OmniXRI)

tinyML案例分享 (以技術分類)



資料來源：https://hackmd.io/@OmniXRI-Jack/tinyML_projects

2022/11/10

台灣IC智造年會_AIoT與tinyML生態系國際發展趨勢與國產IC未來方向_OmniXRI_Jack Hsu

18



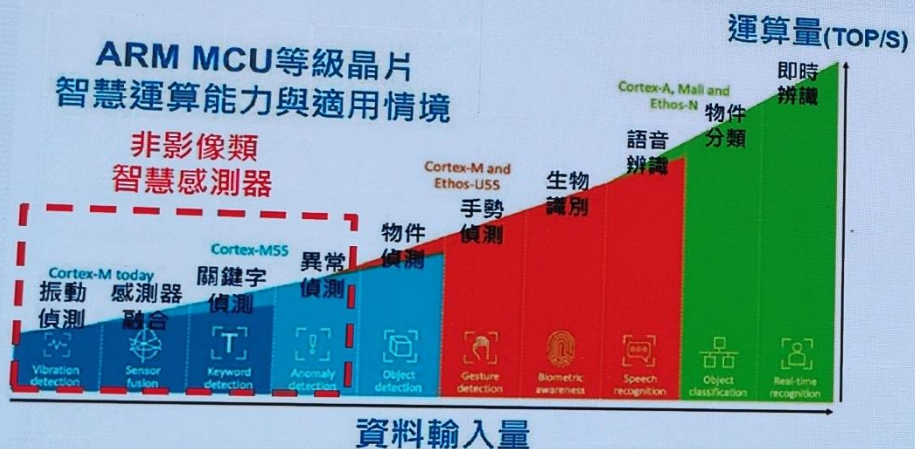
<https://www.youtube.com/watch?t=1049&v=S3UudNBkZyo>

76

tinyML 運用案例

歐尼克斯實境互動工作室(OmniXRI)

tinyML常見應用及限制



MCU等級跨度大
以Arm Cortex-M為例
M0+, M3, M4, M7, M55

指令速度從數十MHz到數百MHz
程式碼儲存空間從數KB到數MB
SRAM從數KM到數MB

圖片來源：<https://www.arm.com/blogs/blueprint/ai-for-iot-devices>

2022/11/10

台灣IC智造年會_AIoT與tinyML生態系國際發展趨勢與國產IC未來方向_OmniXRI_Jack Hsu

9

MCU等級tinyML

優點

- 低單價、功耗
- 低延時(反應快)
- 高隱私(免上網)
- 易連接各式感測器及通訊模組

缺點

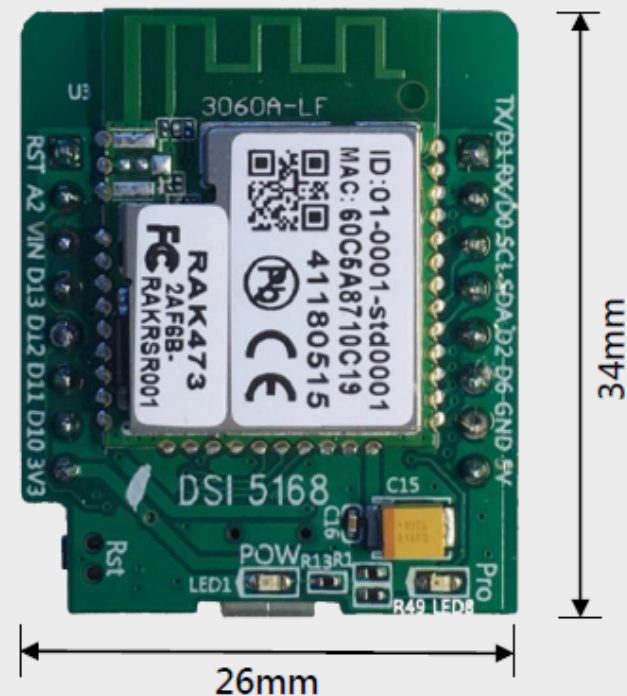
- 速度、算力不足
- 記憶體不足
- 儲存能力小
- 難以在線訓練



<https://www.youtube.com/watch?t=1049&v=S3UudNBkZyo>

DSI5168 晶片介紹

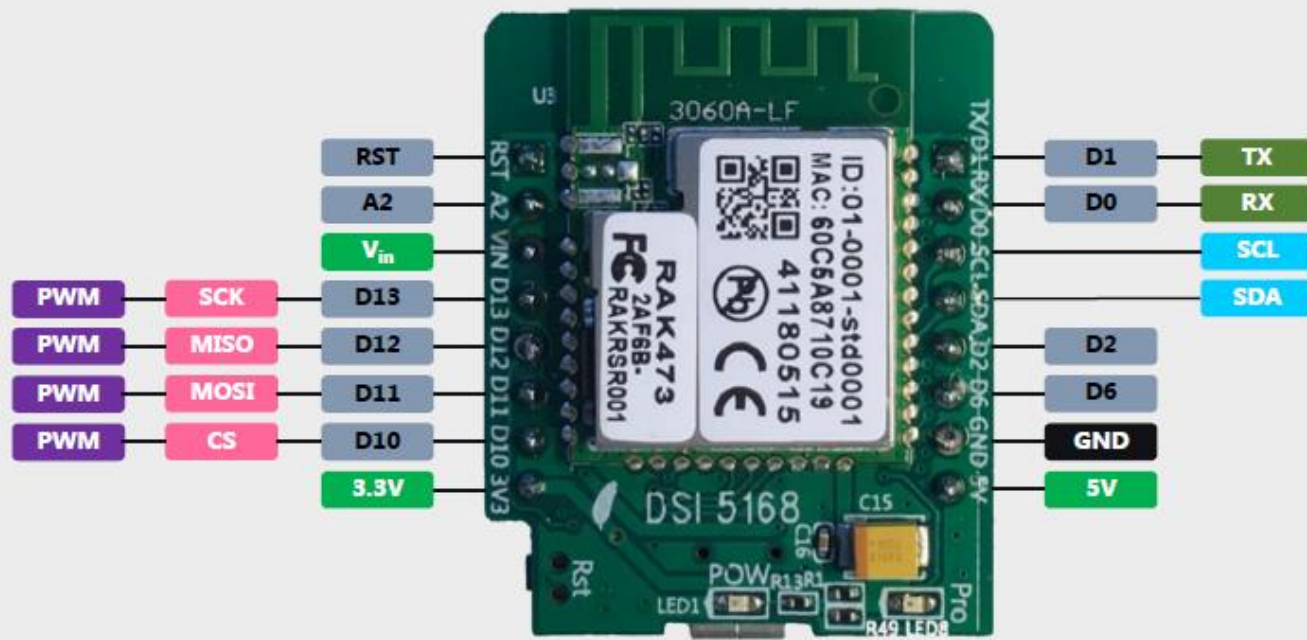
- 採用Realtek Ameba RTL8711AM晶片
- 以ARM架構Cortex M3為核心
- 支援Arduino、IAR、mbed等開發環境
- 擁有2MB Flash 並具 Wi-Fi 功能
- 支援UART、SPI、I2C、PWM等功能
- 擁有8個GPIO腳位



DSI5168 晶片介紹

硬體功能	規格
Chipset	RTL8711AM
MCU	ARM M3/166MHz
I/O	12
ROM	1MB
SRAM	512KB
Internal Flash	N/A
External Flash	2MB
ADC	1
SPI	1
UART	1
I2C	1
I2S	N/A
PWM	4
SSL	Support

- UART Function
- I2C Definition
- Arduino Definition
- SPI Definition
- PWM Function



DSI5168 晶片介紹

- **CPU**
 - ARM®Cortex™-M3 (up to 166MHz)
- **Memory**
 - 1MB embedded ROM
 - 2.5MB embedded RAM
- **Wi-Fi**
 - 2.4GHz 1T1R 802.11b/g/n up to 150Mbps
- **Security**
 - Wi-Fi WEP, WPA, WPA2, WPS
- **Peripheral Interface**
 - NFC tag (read/write)
 - Maximum two PCM with 8/16KHz sample rate
 - UART x 2 (HS-UART x 1, log UART x 1)
 - SPI interface x 1
 - Maximum I2S interface x 1
 - Maximum I2C interface x 3
 - ADC x 1
 - PWM x 4
 - Maximum 19 GPIOs

RTL8711AM



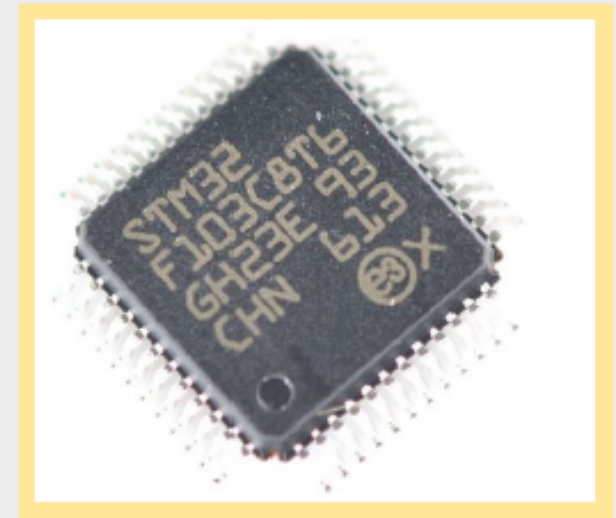
DSI2598+ 晶片介紹

- NB-IoT使用MTK MT2625晶片
- STM32 F103 32 bit核心
- 相容Arduino IDE開發環境
- Keil C / STM32Cube 開發環境
- 多種韌體燒錄方式
- 更多功能腳位，12 bit ADC解析度
- 郵票式電路板設計



DSI2598+ 晶片介紹

- STM32F103C8T6 ARM Cortex M3
- 72 MHz maximum frequency
- 64 Kbytes of Flash memory
- 20 Kbytes of SRAM
- 8 MHz system crystal
- 32.768 KHz RTC crystal
- 2x SPI, 3x USART, 2x I2C, 1x CAN
- USART1 for NB-IoT & firmware upload (with boot0 = 1, JP4 connect to 3.3V)



DSI2598+ 晶片介紹

本開發板設計，可提供多種程式燒錄方式，一般出廠設定為方便使用，免去外接燒錄設備，採用bootloader燒錄方式，如此一來只需接上開發板的USB即可透過Arduino IDE選擇STM32duino bootloader選項，直接燒錄。

另有使用FTDI工具的Serial燒錄方式與STLink工具的STLink燒錄方式

三種上傳方法

STM32duino
bootloader

Serial

STLink

USB直接燒錄

FTDI工具

STLink工具



補充資料

Ideaschain應用案例分享(附程式碼):

<https://iforum.ideaschain.com.tw/iforum/techmatch/list.do>

SensiMI應用案例:

https://www.youtube.com/watch?v=4b_cMLDofO8

各類國產晶片資源:

https://www.ideas-hatch.com/mem_evb.jsp

如何焊好新購開發板

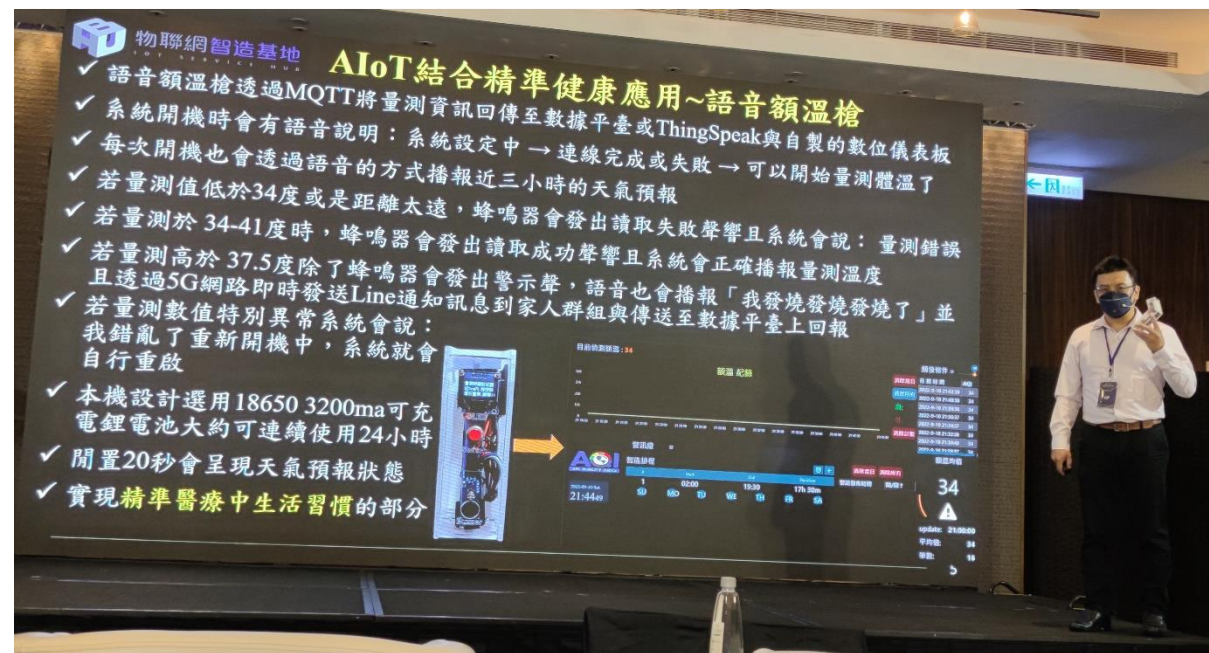


圖 1. 新買開發板排針與主板分開



圖 2. 使用麵包板固定後再焊

5168其他應用案例



<https://iforum.ideaschain.com.tw/iforum/techmatch/list.do>

<https://www.youtube.com/watch?t=1049&v=S3UudNBkZyo>

國產晶片相關消息



IDEAS Hatch:

<https://www.ideas-hatch.com/>

<https://www.facebook.com/iiiideashatch>

https://www.ideas-hatch.com/mem_evb.jsp

講師介紹

講師: 章育銘

- # 退役競賽獎金獵人
- # 門薩成員
- # 資訊工業策進會 講師
- # 工業技術研究院 副工程師
- # 物聯網策略解決專家
- # 甲種電匠
- # 交大電控博士休學中
- ... +10



專長: 自動化、系統整合、人工智慧...





Thank you



物聯網智造基地

I O T S E R V I C E H U B