

# Himax WE-I Plus



智慧顯微鏡

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

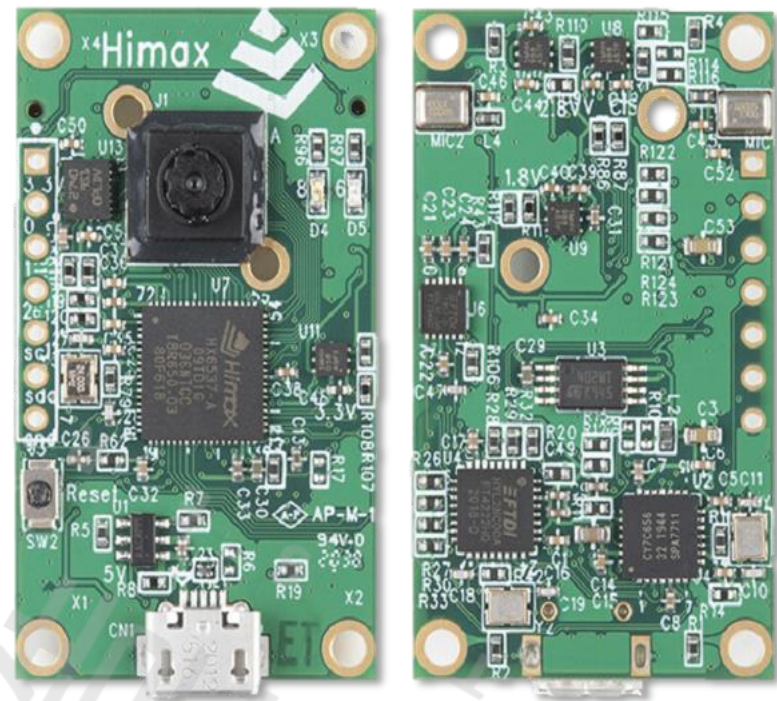
主辦單位：財團法人資訊工業策進會  
INSTITUTE FOR INFORMATION INDUSTRY

IoT SERVICE HUB  
物聯網智造基地

合作單位：章育銘

# 大綱

- 一、前言
- 二、系統介紹
- 三、AI模型製作
- 四、燒入晶片
- 五、結果展示
- 六、補充資料



# 一、前言

- 顯微鏡操做需耗費大量人力,
- 長時間下不只傷眼也耗費體力,
- 本案將國產開發晶片結合顯微鏡照片,
- 製出智慧顯微鏡打入生醫市場。

這邊找到知名數據建模和數據分析競賽平台Kaggle上的一個花粉顯微照片分類資料，甚至可分類完跟其他好手比較：

<https://www.kaggle.com/datasets/andrewmvd/pollen-grain-image-classification>

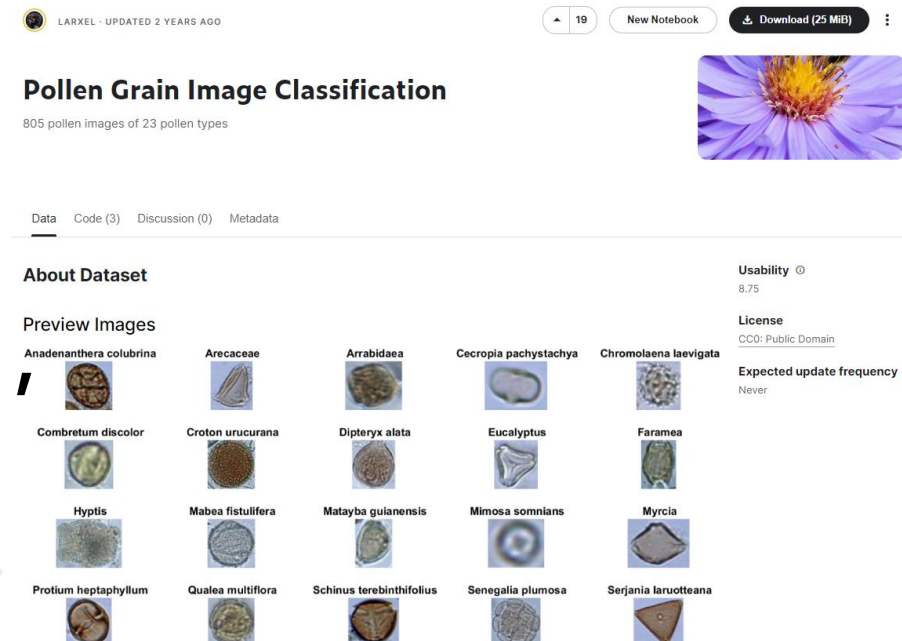


圖 1. kaggle平台上數據集

曾有一名病患就診時面露驚恐之色說「一隻眼睛突然看不到」，細問之後才得知，原來有科技人工作需要長期用單眼看顯微鏡、檢查光碟片，這些工作用眼過度且依賴光照，很容易造成黃斑部傷害。而以大多是中老年人才有的乾眼症，工程師則因常緊盯螢幕、儀器而忘了眨眼，一分鐘眨眼不到五次，特別容易引發乾眼症。

圖 2.工程師長時間看近物眼壓升高、黃斑部問題案例,  
<https://health.udn.com/health/story/5970/4080547>



## 二、系統介紹

WE-I Plus開發晶片架設於螢幕前，  
將辨識分類判斷後輸出。

由於3個GPIO輸出，故我們分類7個類別

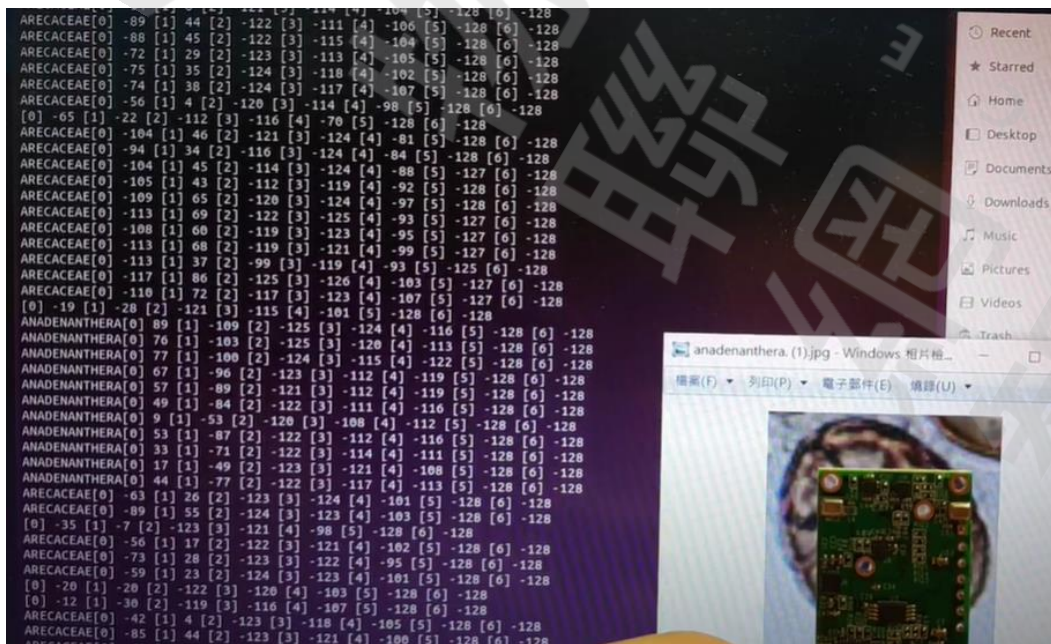


圖 1. 說明示意圖

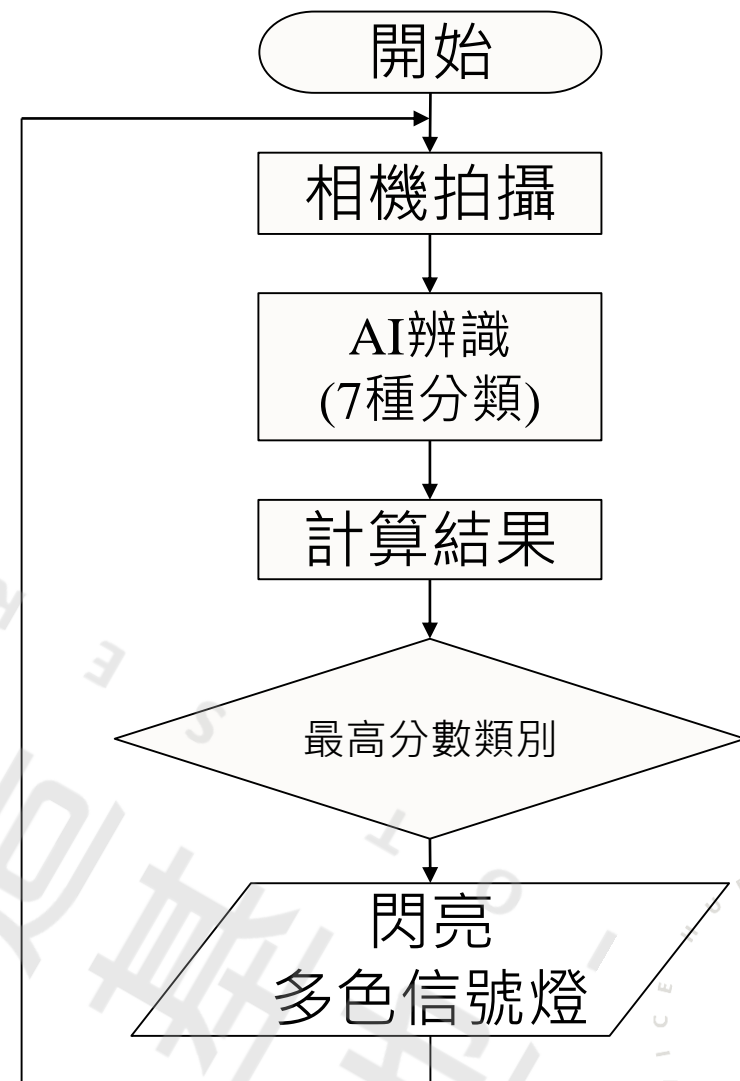


圖 2. 程式流程圖

## 二、系統介紹

準備材料:

1. 電腦
2. WE-I Plus開發板
3. LED\*3色
4. 線材
5. 麵包板

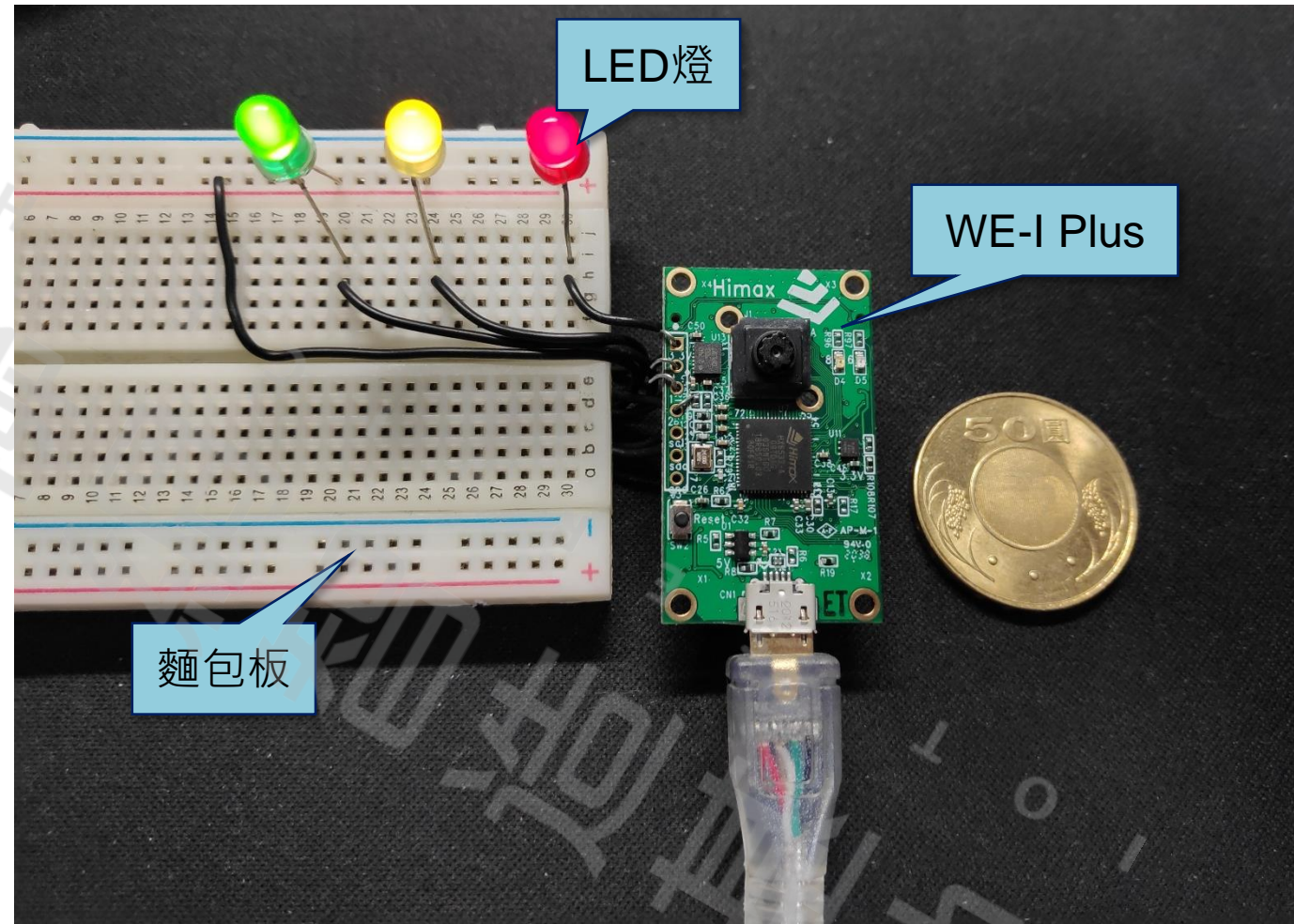


圖 1. 準備材料

# 三、AI模型製作

使用Edge Impulse快速上手，前往官方網頁：  
<https://www.edgeimpulse.com/>

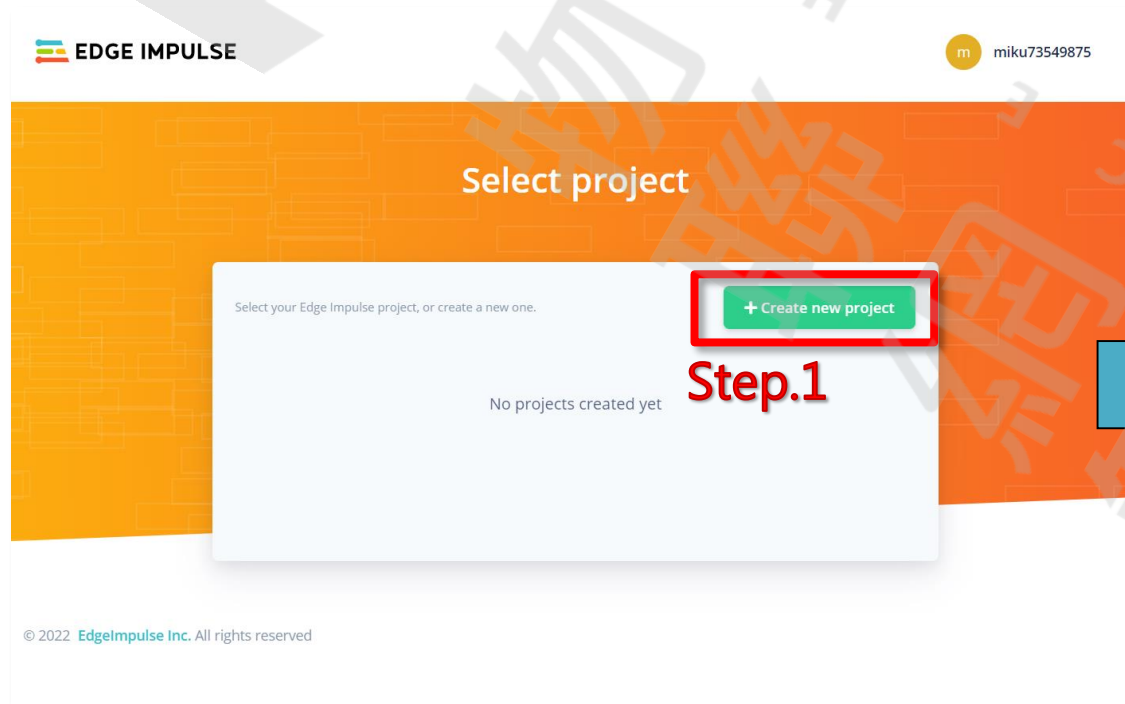


圖 1. 創建專案

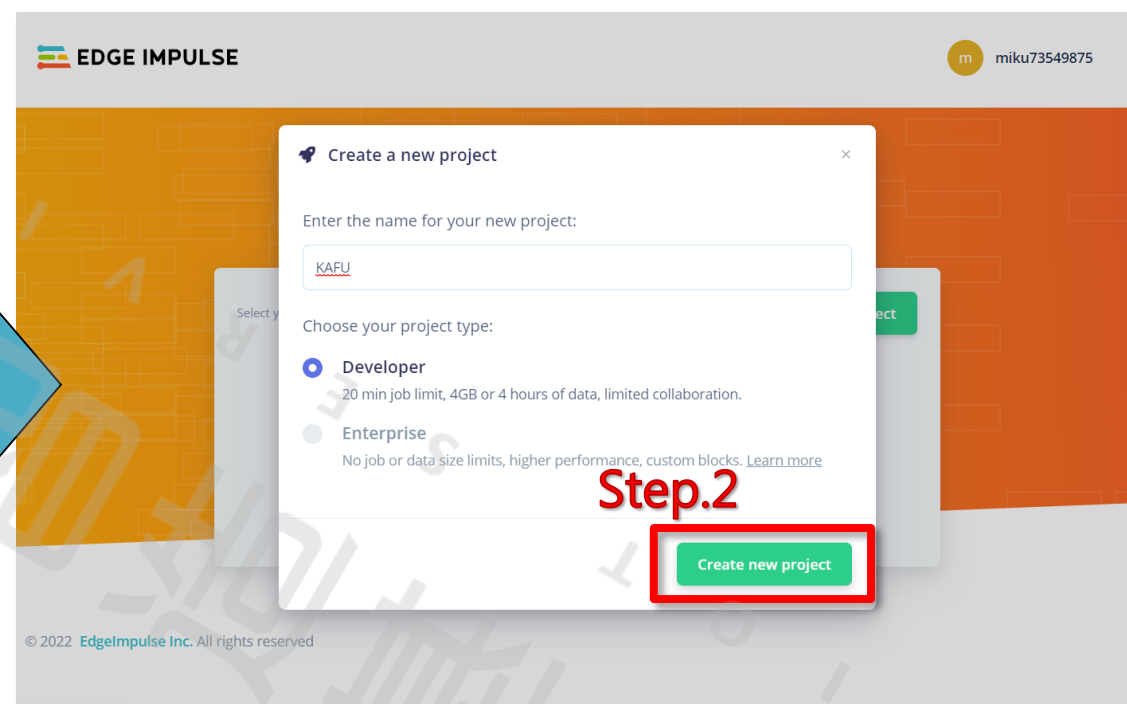


圖 2. 選擇開發模式



# 三、AI模型製作

於Edge Impulse選單，根據專案類型設定專案。

Step.3

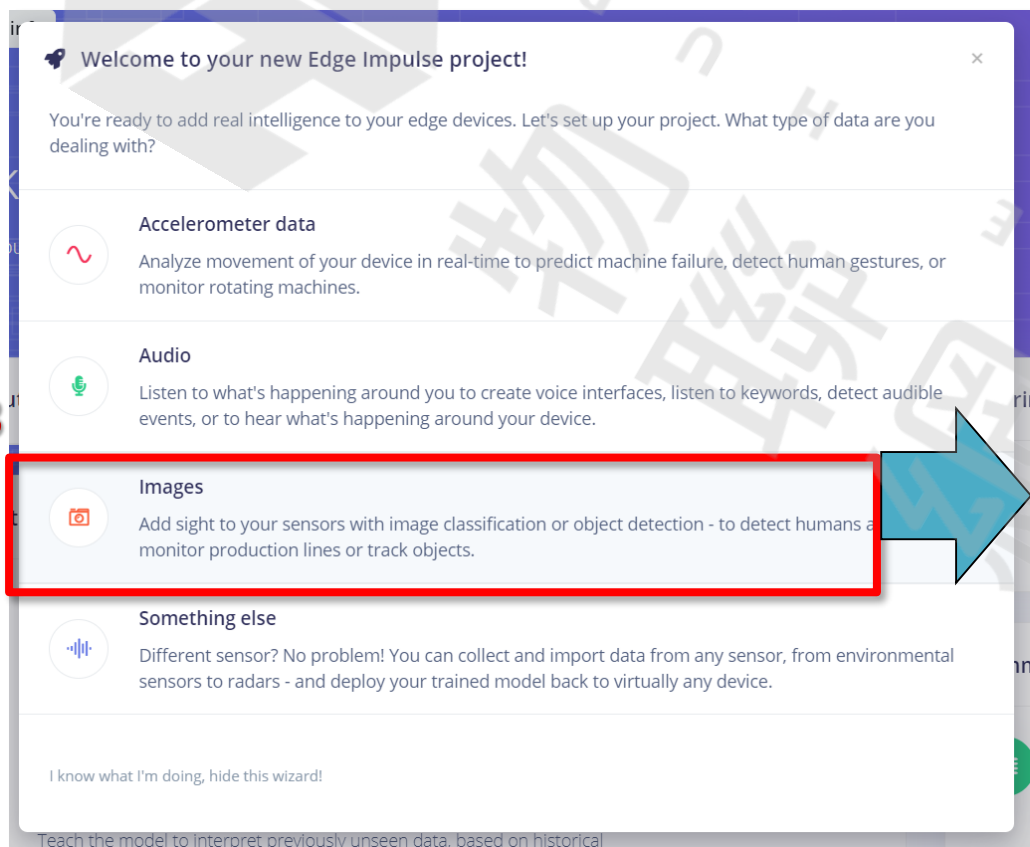


圖 1. 選擇開發類型

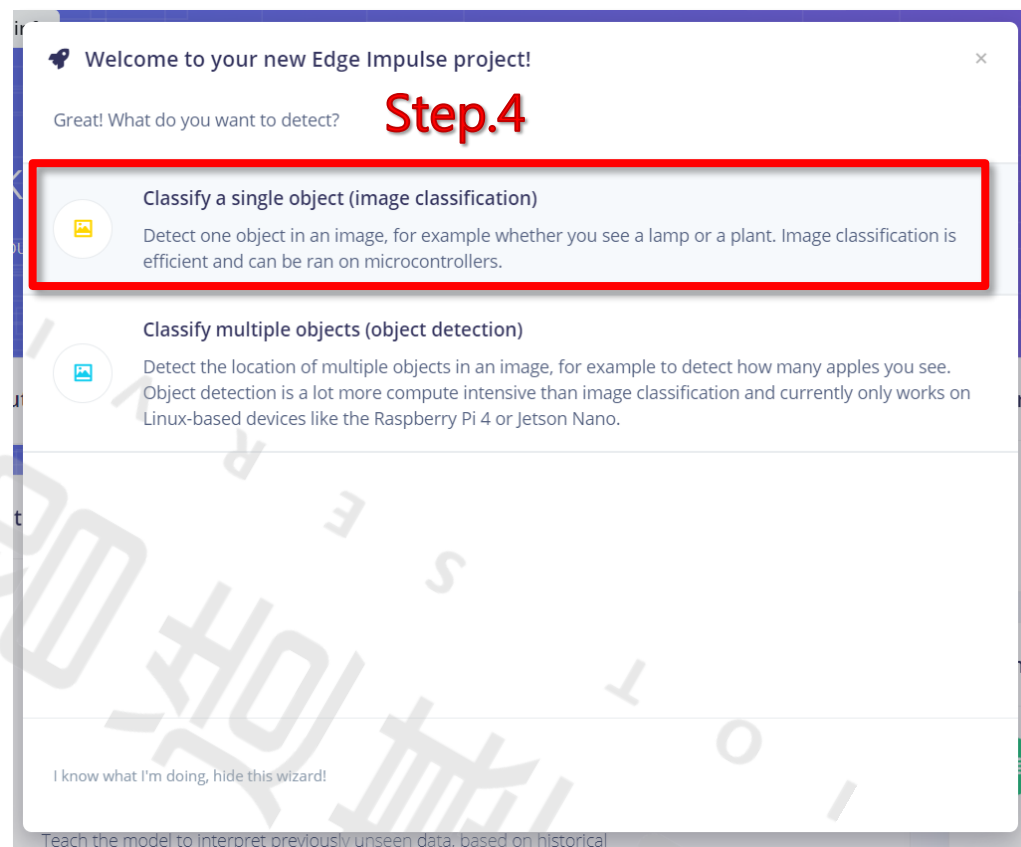


圖 2. 選擇'影像分類'

# 三、AI模型製作

於Edge Impulse選單，根據專案類型設定專案。



圖 1. 選擇開發類型

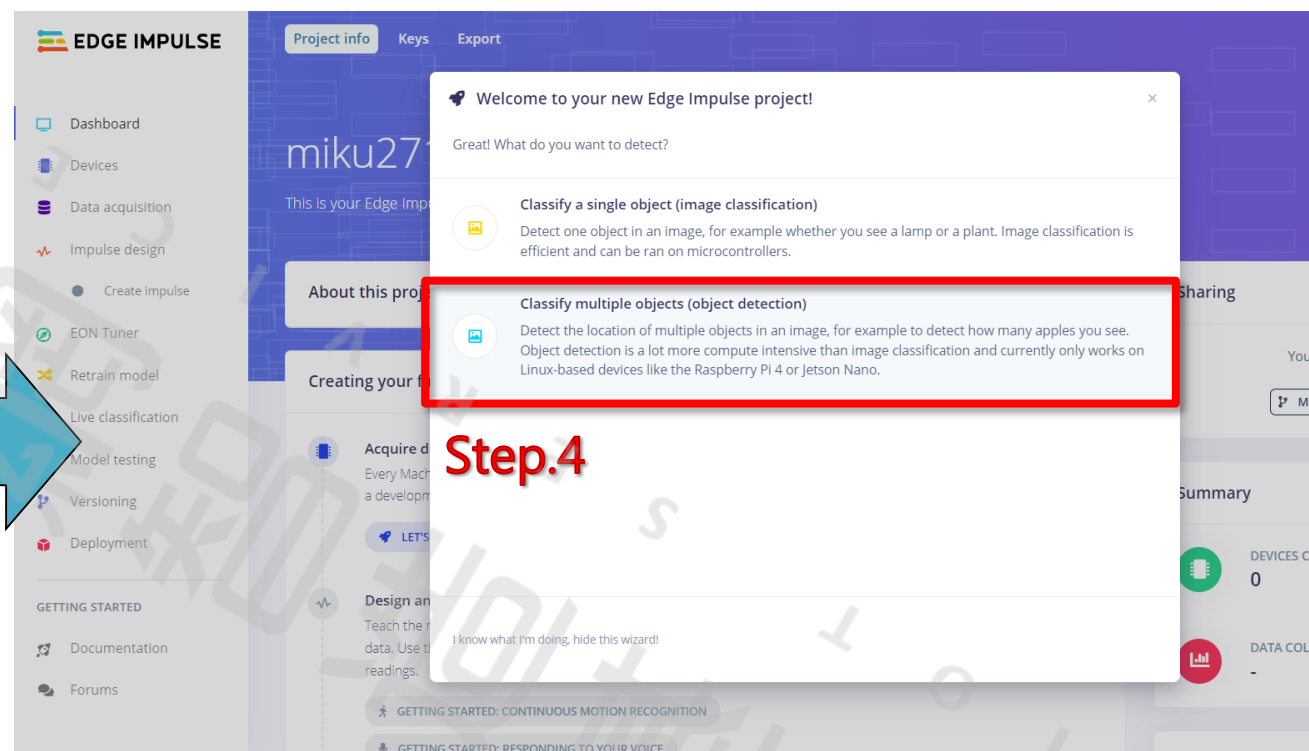
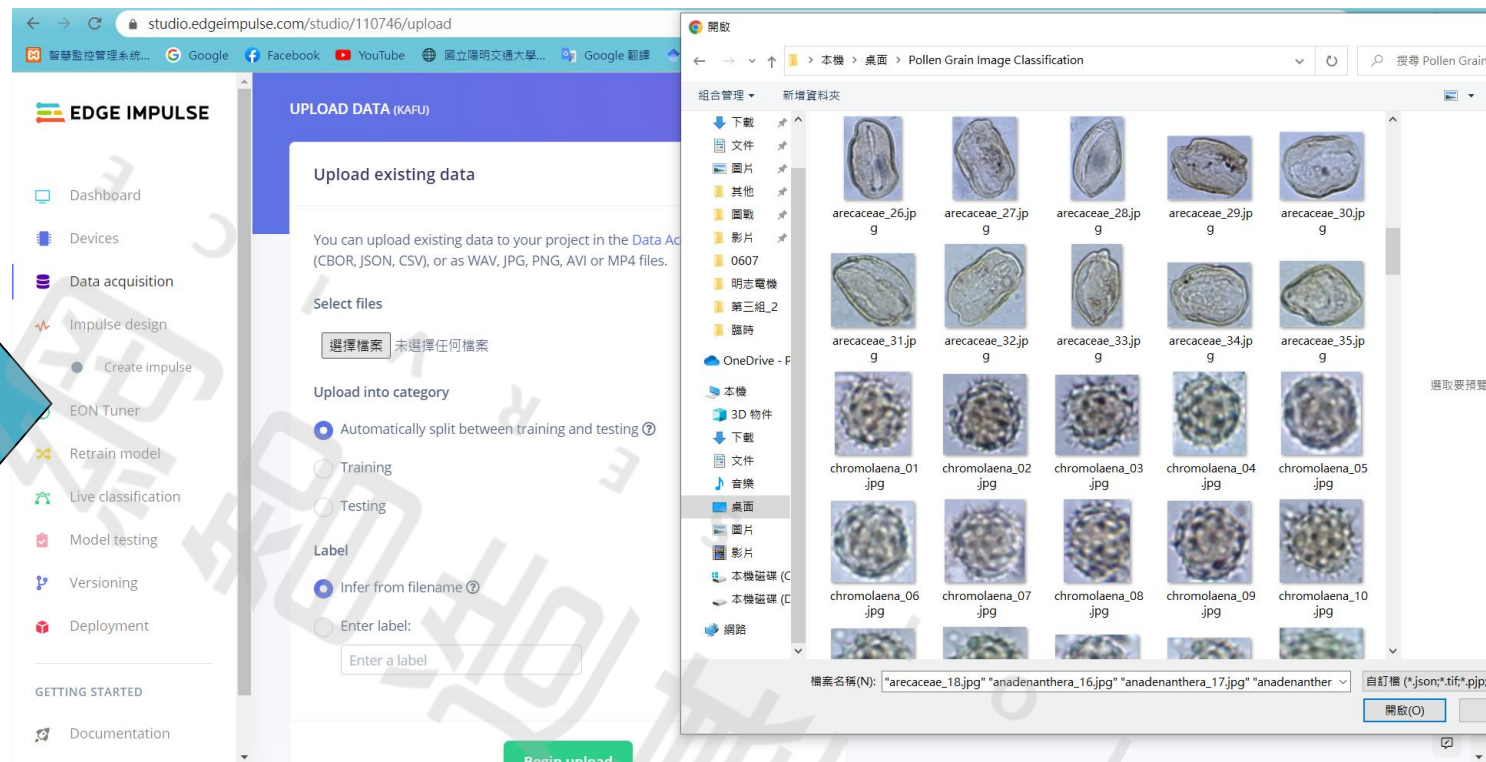


圖 2. 選擇'物件定位'



將訓練模型照片資料上傳。



### 圖 1. 點選匯入資料

### 圖 2. 'Ctrl+A'全選想要資料集

# 三、AI模型製作

將檔名先修改好, Edge Impulse 就會標好類別。



需將kaggle下載資料檔名先更改過

圖 1. 設定自動分類標籤

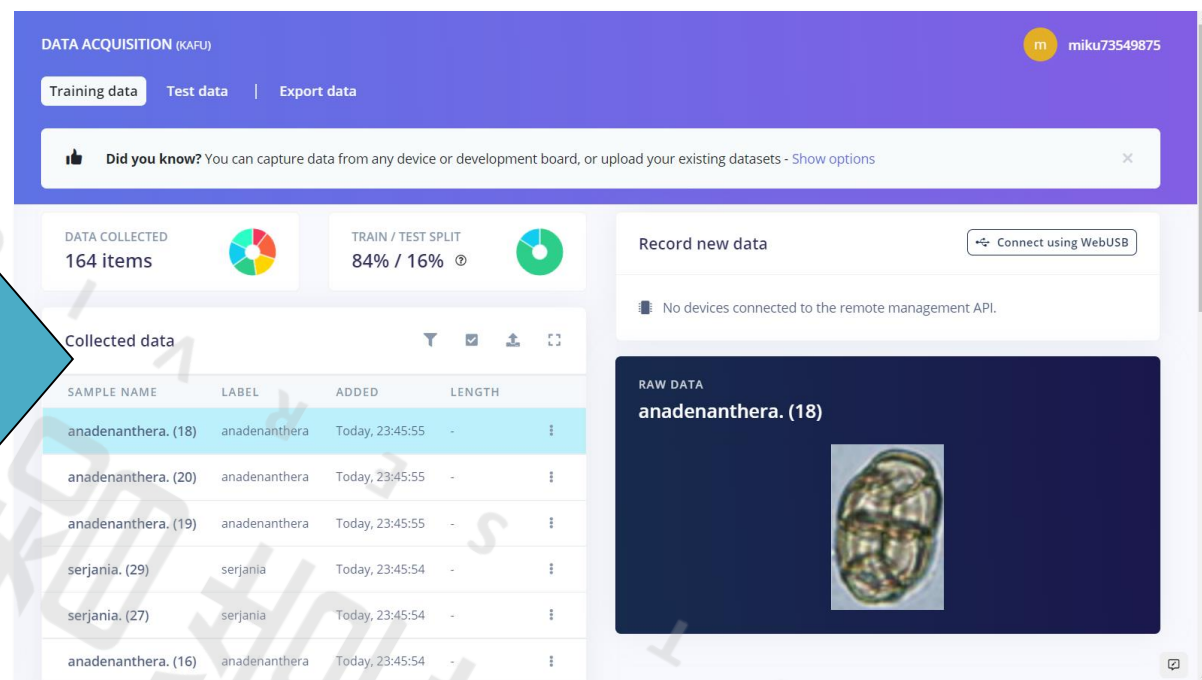


圖 2. 確認資料類別與數量

# 三、AI模型製作



圖 1. 點選'impuls design'

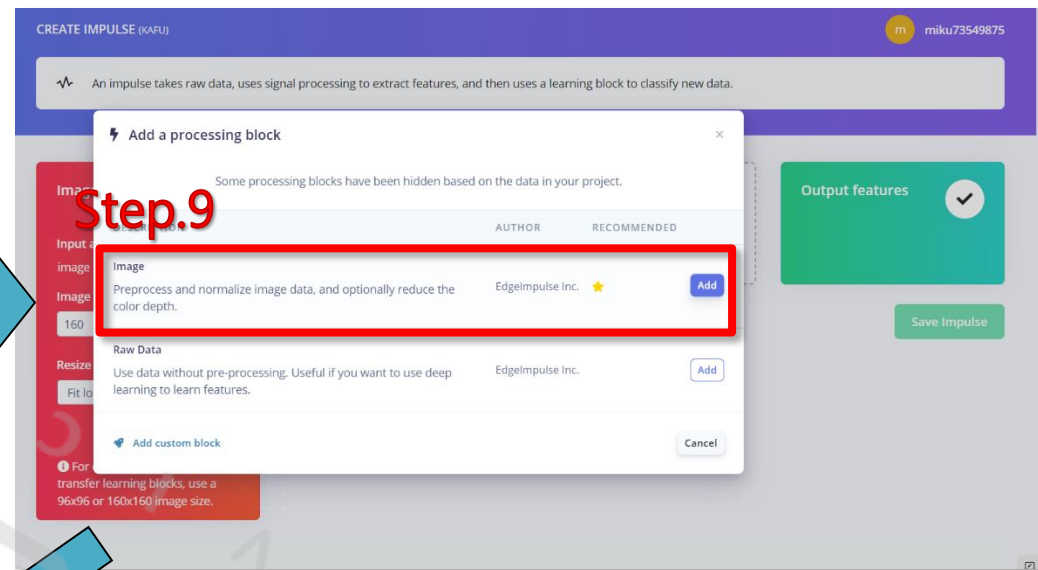


圖 2. 新增照片方塊

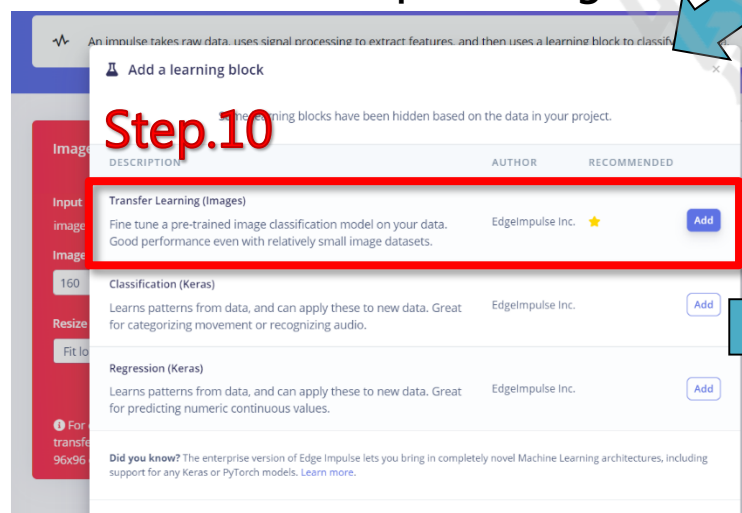


圖 3. 點選遷移學習會比傳統分類佳

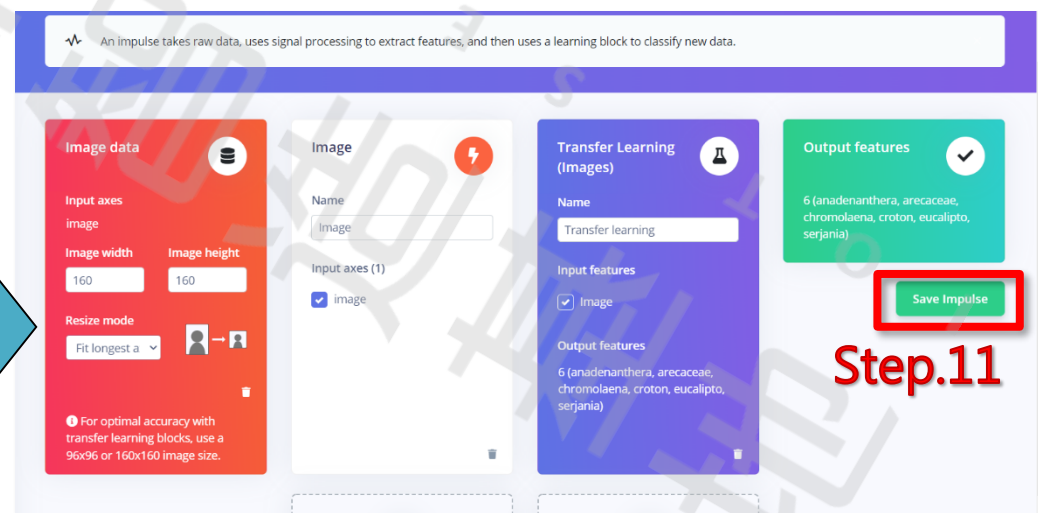


圖 4. 儲存模型



# 三、AI模型製作

## 資料增強

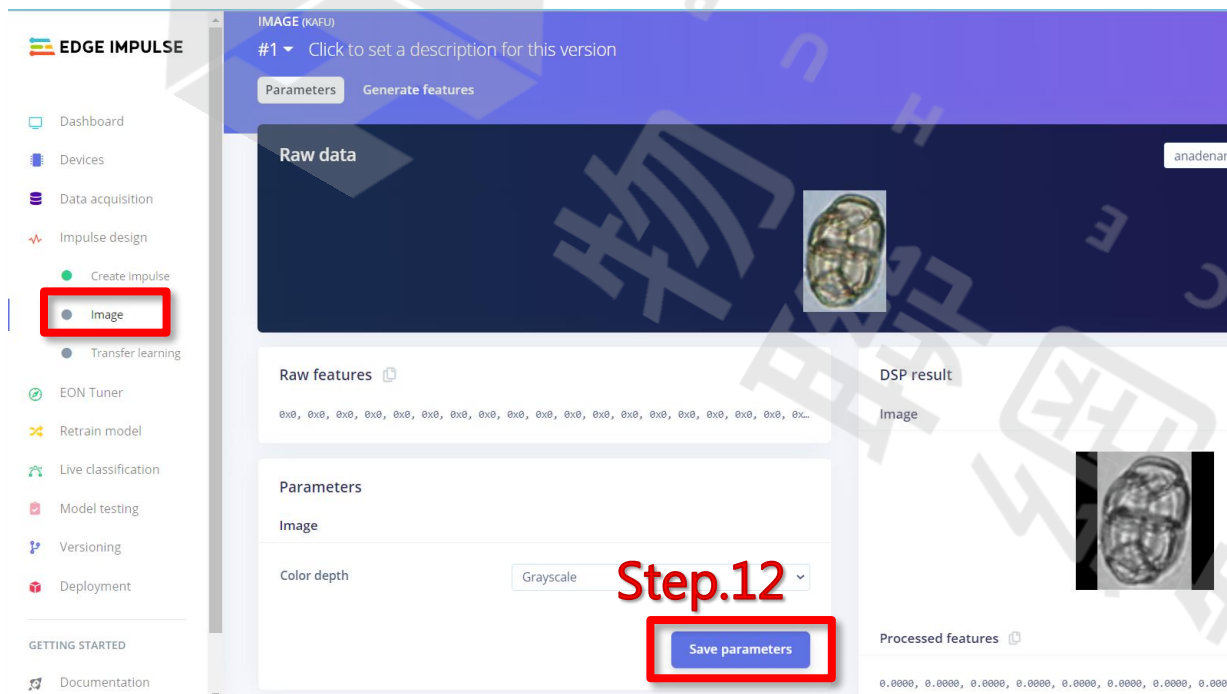


圖 1. 選擇輸入單通道灰階圖片

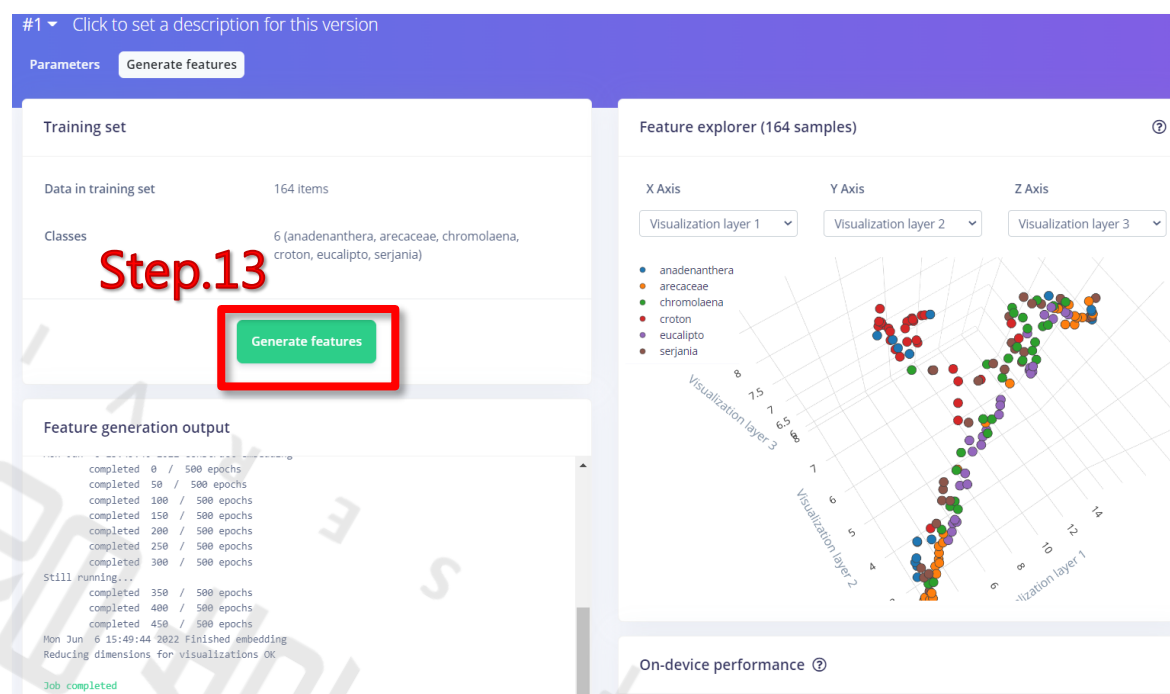
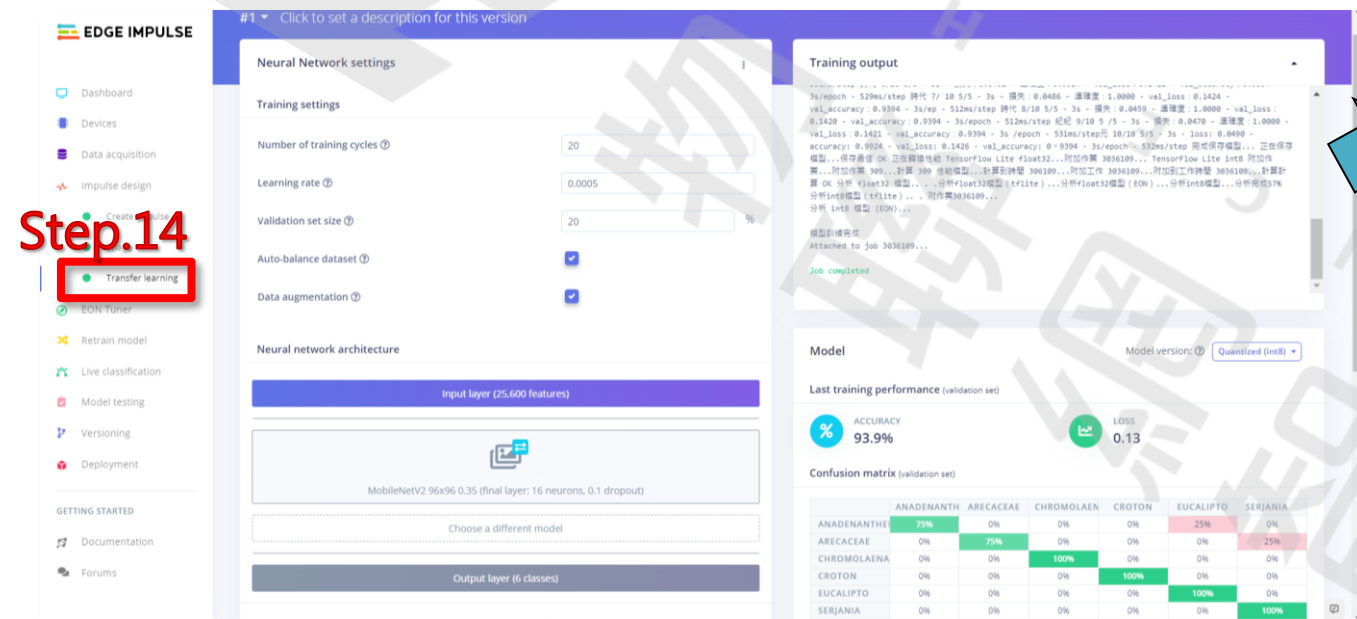


圖 2. 資料增強產生

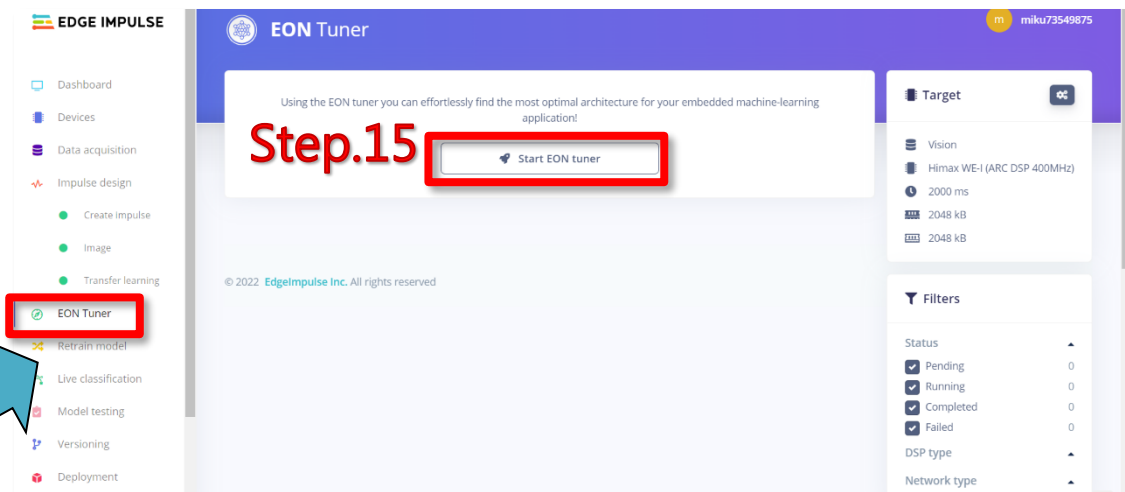
資料增強 ( Data Augmentation ) 是一種通過讓有限的資料產生更多的等價資料來人工擴充套件訓練資料集的技術。它是克服訓練資料不足的有效手段，目前在深度學習的各個領域中應用廣泛。但是由於生成的資料與真實資料之間的差異，也不可避免地帶來了噪聲問題。

### 三、AI模型製作

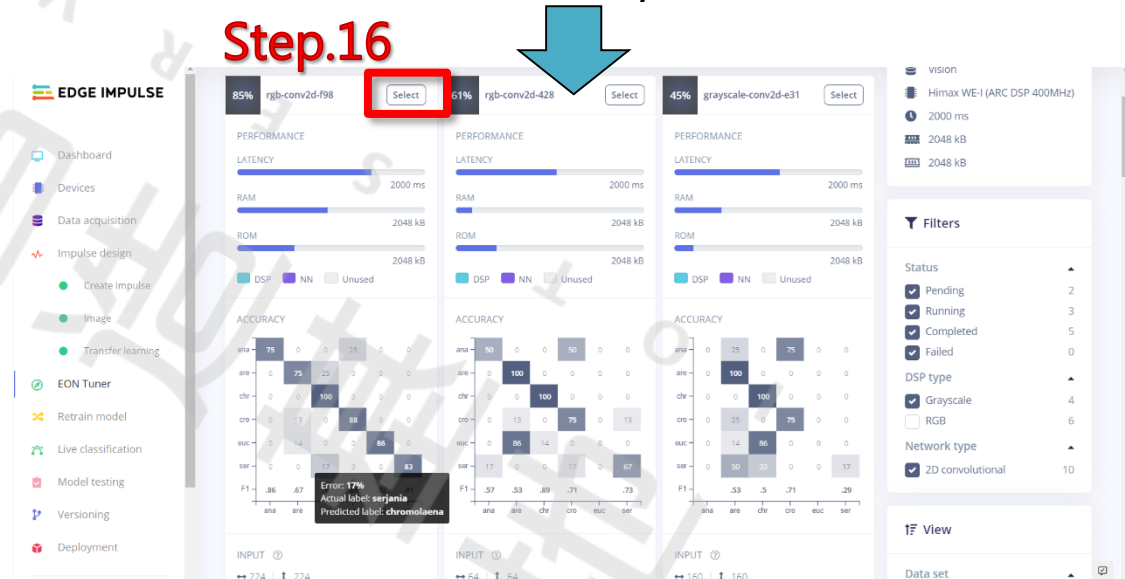
## 訓練模型



### 圖 1. 訓練模型



## 圖 2.使用 EON 調諧器, 找尋更佳模型



### 圖 3.選擇最佳模型

# 三、AI模型製作

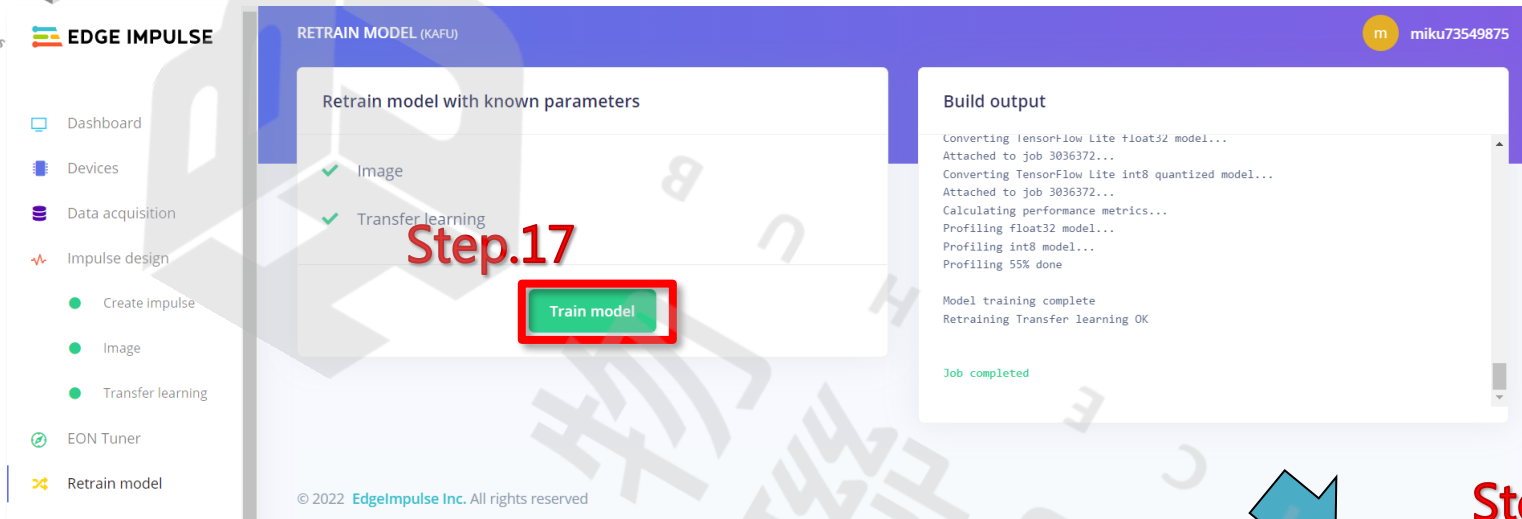


圖 1. 再次訓練模型

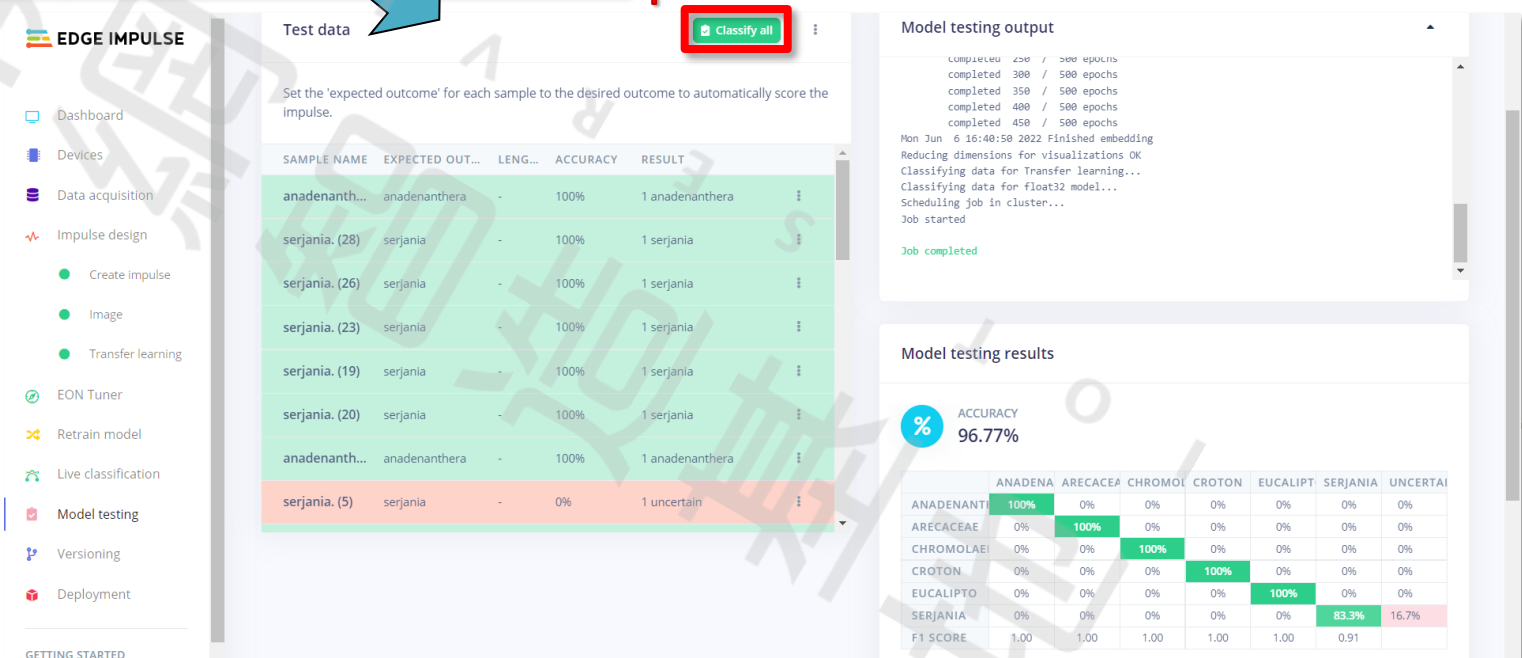


圖 2. 觀看測試集結果



# 三、AI模型製作

## 模型匯出。

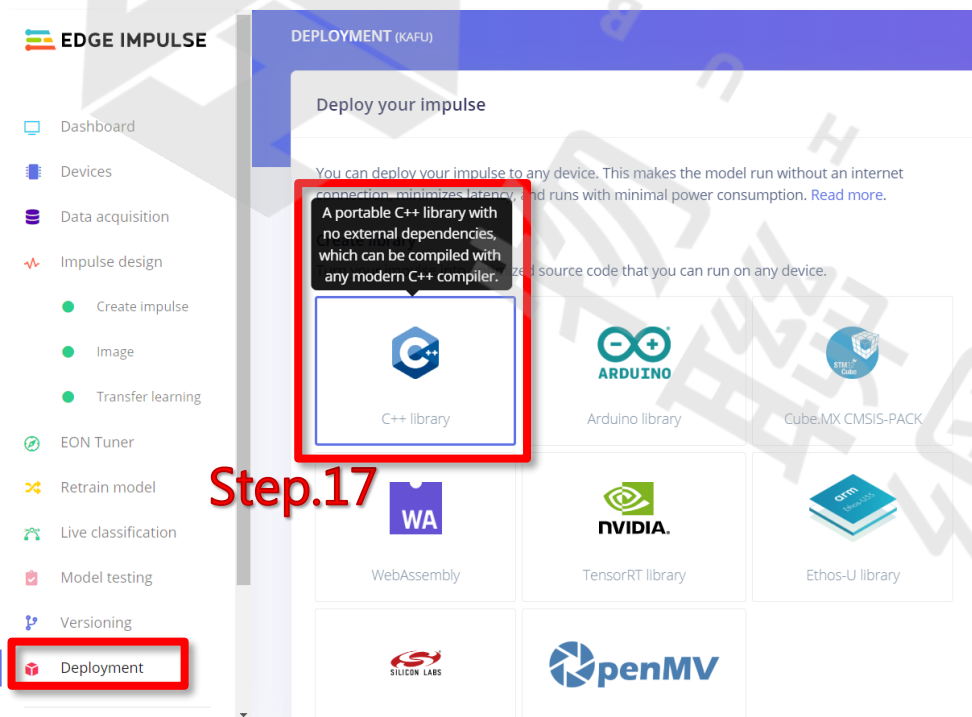
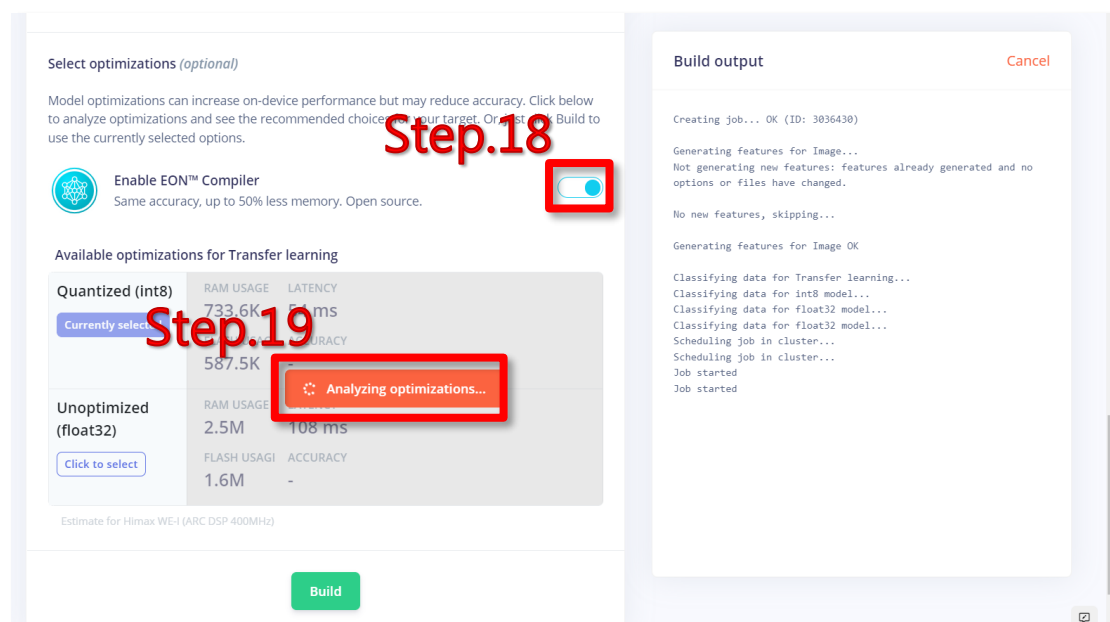


圖 1.選擇C++庫



Step.18

Step.19

圖 2.INT8優化



Step.20

圖 3. 建立並匯出

## 四、燒入晶片

將訓練好的AI模型燒入進WE-I Plus 裡。

由於需要使用到Linux系統,故WINDOWS系統使用者這邊先從建構虛擬電腦方式帶領。



圖 1. 進入VirtualBox官方頁面下載-> <https://www.virtualbox.org/>



圖 2. 安裝點選下一步

# 迅速帶過安裝步驟。



圖 1. 點選下一步



圖 2. 點選安裝

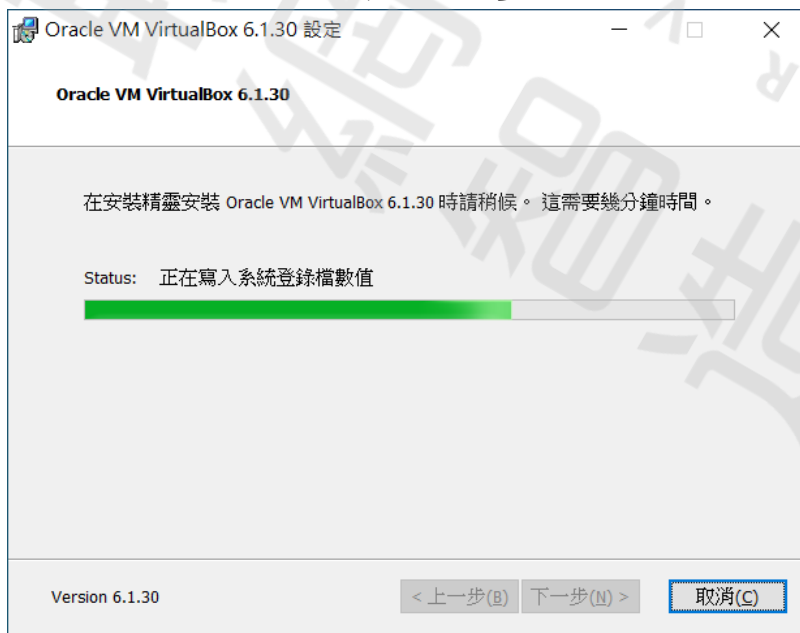


圖 3. 等待安裝



圖 4. 安裝所需驅動



## 四、燒入晶片



圖 1. 安裝成功後開啟VirtualBox管理員設定, 點選'新增'

### 名稱和作業系統

請為新的虛擬機器選擇描述性名稱和目的地資料夾，並選取要在其上安裝的作業系統類型。您選擇的名稱將在整個VirtualBox 中使用，以標識這部電腦。

名稱:

機器資料夾:

類型(T):

版本(V):

### Step.7

專家模式(E)  取消

圖 2. 輸入虛擬機器名稱

### 記憶體大小

選取配置到虛擬機器的記憶體量 (RAM)，單位 MB。

建議的記憶體大小為 1024MB。

4 MB  MB

8192 MB

### Step.8

取消

圖 3. 設定資源

### 硬碟

如果您希望能加入虛擬硬碟到新的機器。可以建立新的硬碟檔或從清單選取一個或使用資料夾圖示選取另一個位置。

如果需要更多複雜存放裝置設定，可以略過此步驟，並在機器建立時進行變更機器設定。

建議硬碟的大小為 10.00 GB。

- ☐ 不加入虛擬硬碟(D)
- ☒ 立即建立虛擬硬碟(C)
- ☐ 使用現有虛擬硬碟檔案(I)

空的

### Step.9

取消

圖 4. 設定內容

### 硬碟檔類型

請選擇新的虛擬硬碟希望使用的檔案類型。如果不需要用在其它虛擬化軟體，您可以保留這個設定不變更。

- ☒ VDI (VirtualBox 磁碟映像)
- ☐ VHD (虛擬硬碟)
- ☐ VMDK (虛擬機器磁碟)

### Step.10

專家模式(E)  取消

圖 5. 設定內容

# 四、燒入晶片

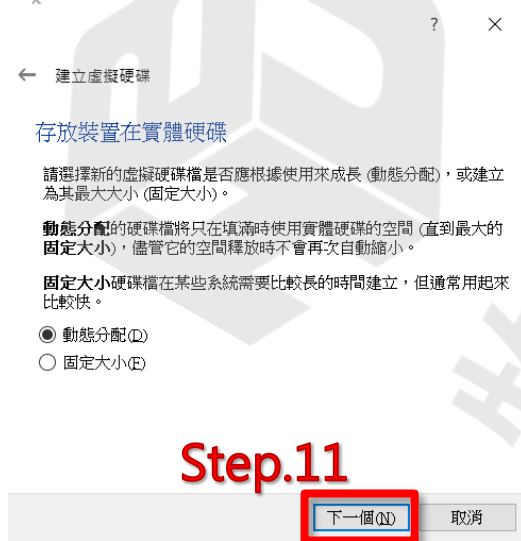


圖 1. 選擇設定

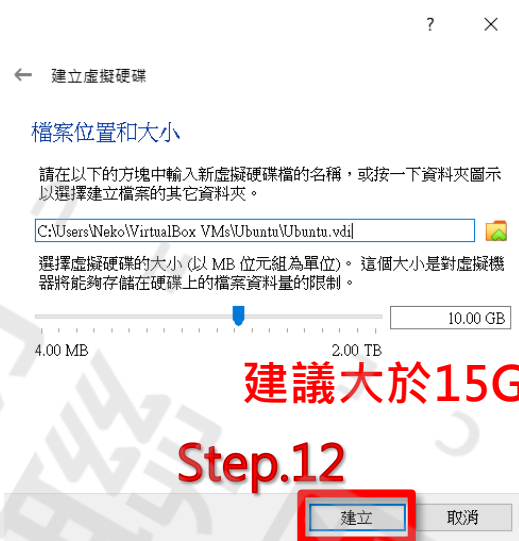


圖 2. 點選建立



圖 3. 點選 設定

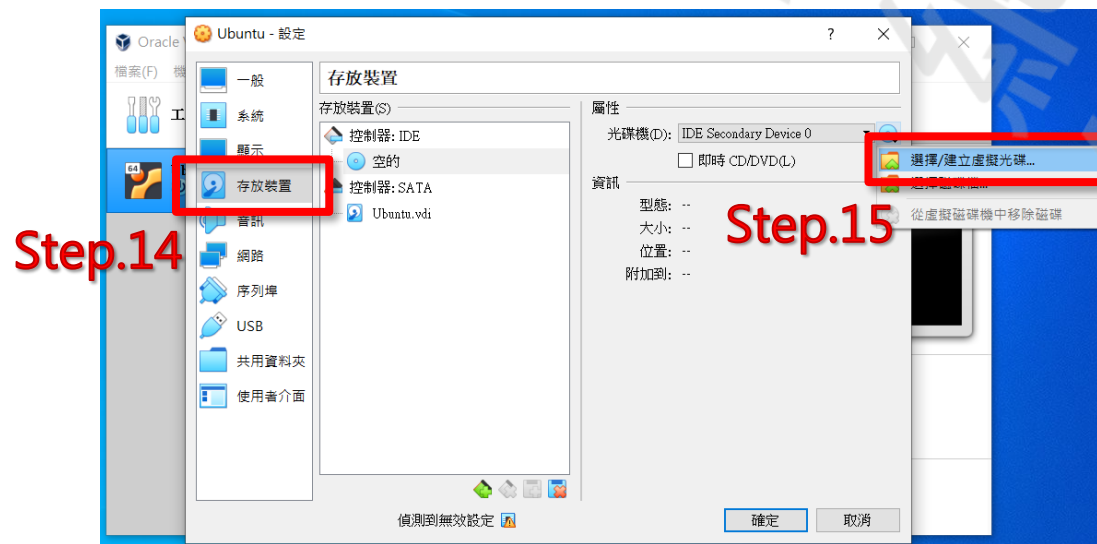


圖 4. 選擇映像檔

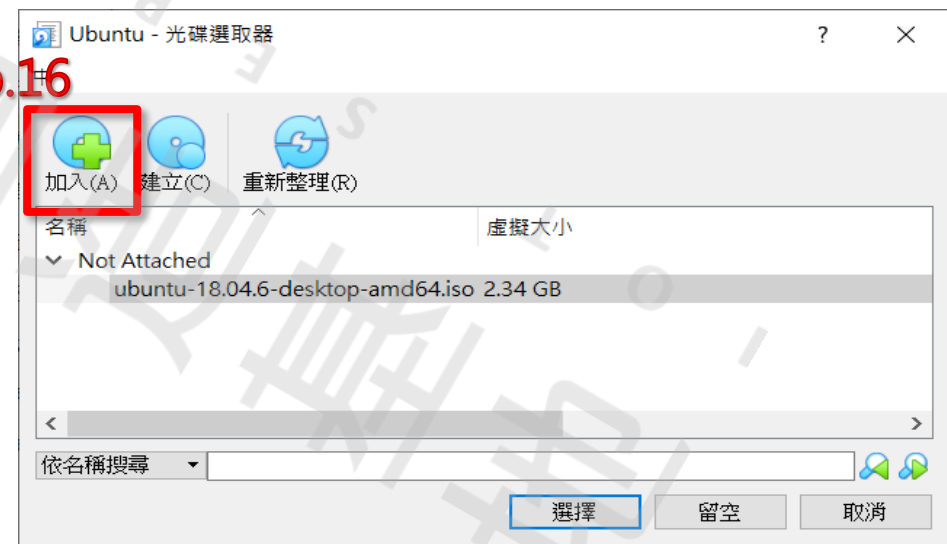


圖 5. 新增映像檔

# 四、燒入晶片

開始安裝Ubuntu，經一連串步驟後，安裝如圖2即為成功。

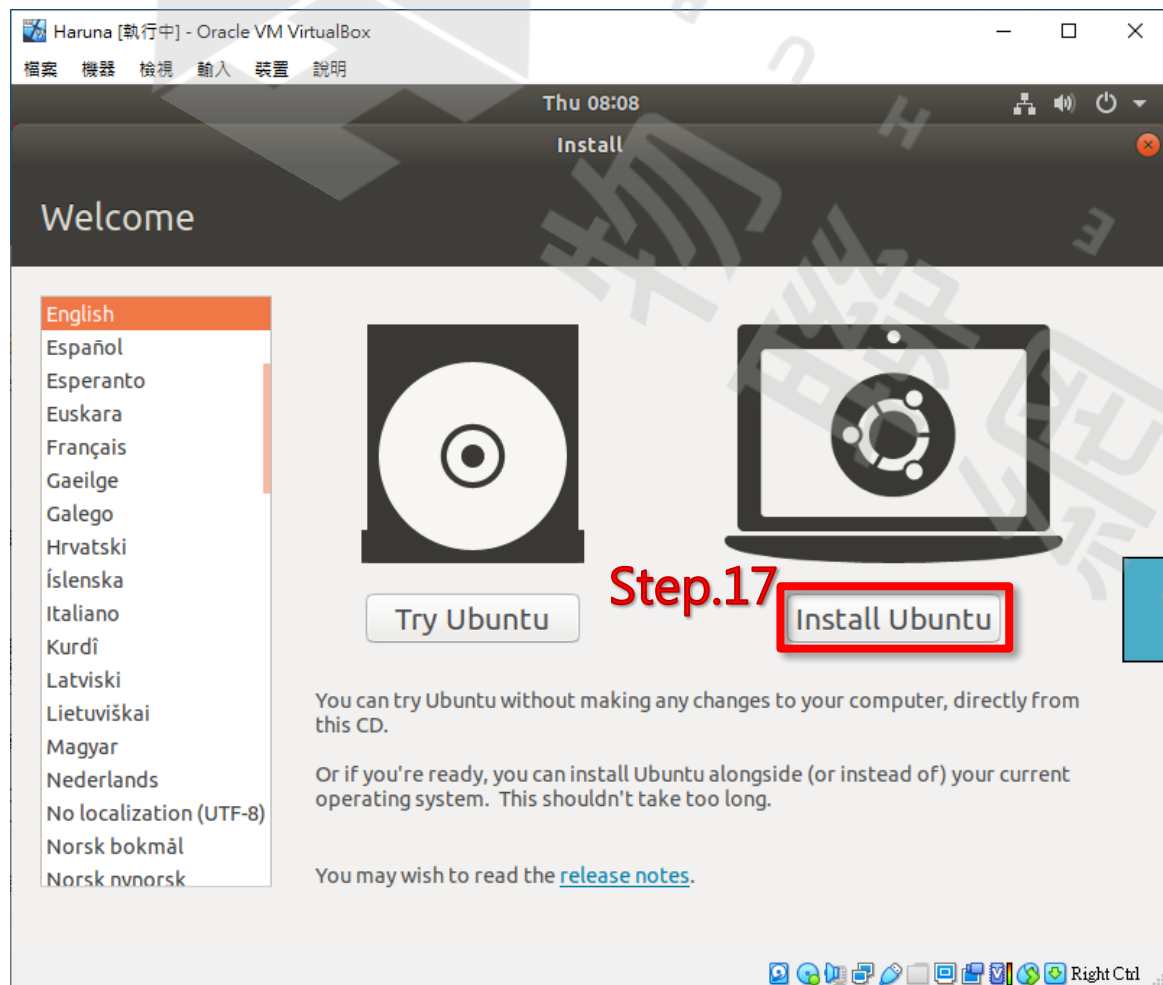


圖 1. 開始安裝Ubuntu

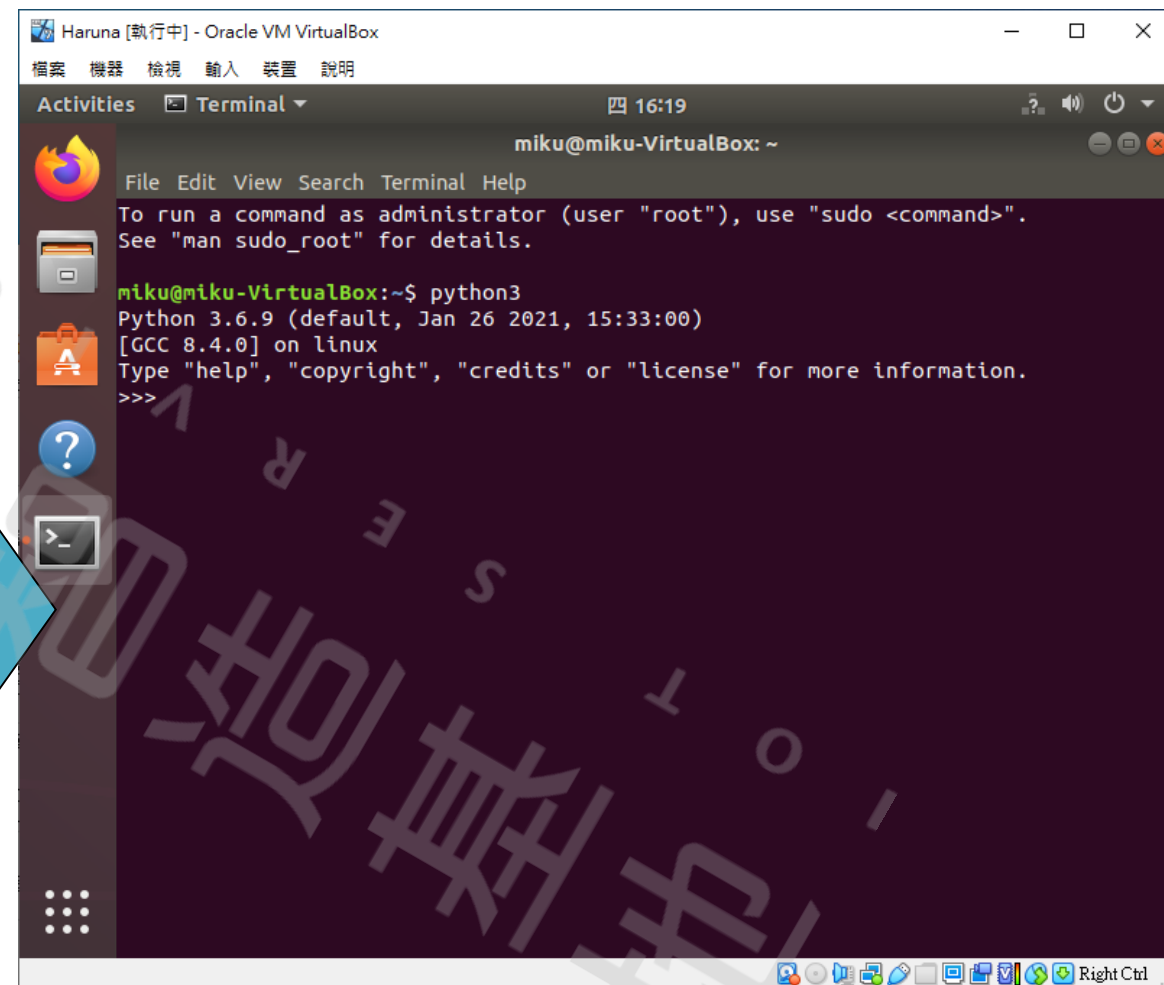


圖 2. 安裝成功頁面



## 四、燒入晶片

編輯C/C++ 程式碼後，以 ARC GNU tool compile project，使用 make 指令產生.elf 檔案，再用 make 指令產生 img 檔案，後續再將 img 檔透過 minicom 燒入

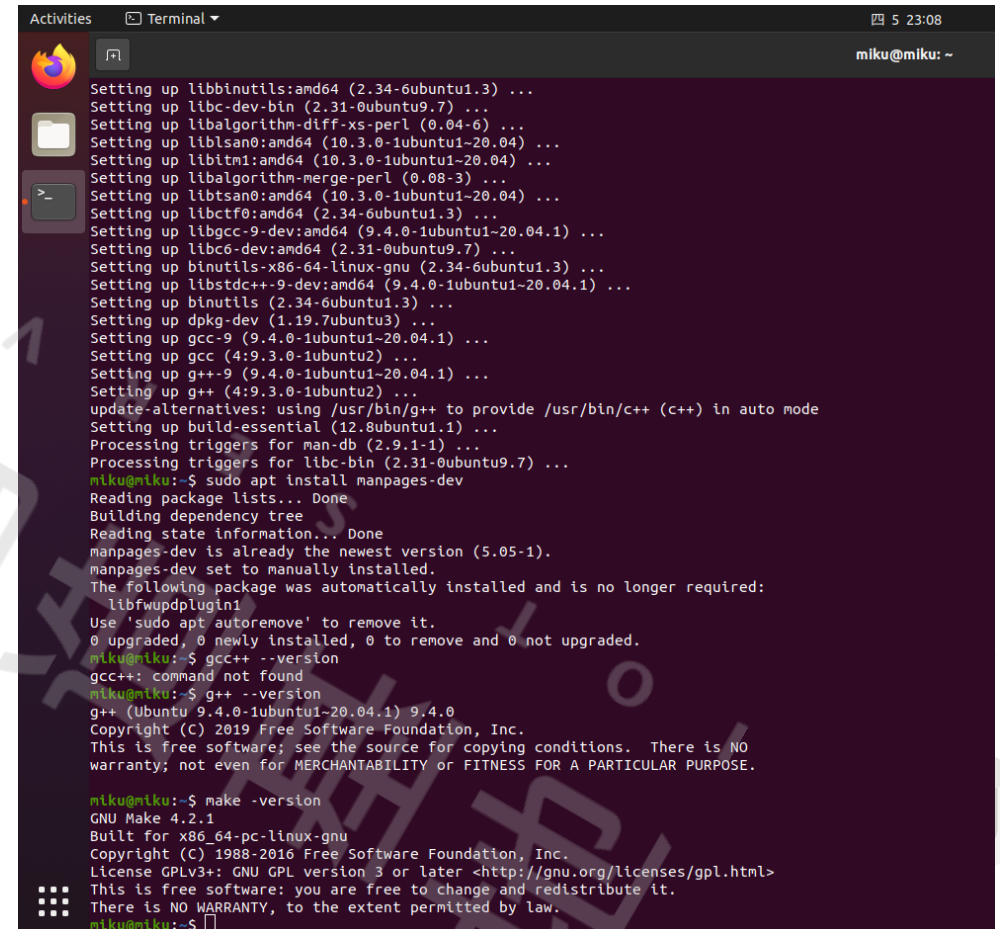
Step1. 於Ubuntu呼叫出終端機

Step2. 輸入指令安裝

```
$ sudo apt update
$ sudo apt install build-essential
$ sudo apt get install manpages-dev
```

Step3. 輸入指令確認安裝是否成功

```
$ gcc --version
$ g++ --version
$ make --version
```



```
Setting up libbinutils:amd64 (2.34-6ubuntu1.3) ...
Setting up libc-dev-bin (2.31-0ubuntu9.7) ...
Setting up libalgorithm-diff-xs-perl (0.04-6) ...
Setting up liblsan0:amd64 (10.3.0-1ubuntu1-20.04) ...
Setting up libitm1:amd64 (10.3.0-1ubuntu1-20.04) ...
Setting up libalgorithm-merge-perl (0.08-3) ...
Setting up libtsan0:amd64 (10.3.0-1ubuntu1-20.04) ...
Setting up libctf0:amd64 (2.34-6ubuntu1.3) ...
Setting up libgcc-9-dev:amd64 (9.4.0-1ubuntu1-20.04.1) ...
Setting up libc6-dev:amd64 (2.31-0ubuntu9.7) ...
Setting up binutils-x86-64-linux-gnu (2.34-6ubuntu1.3) ...
Setting up libstdc++-9-dev:amd64 (9.4.0-1ubuntu1-20.04.1) ...
Setting up binutils (2.34-6ubuntu1.3) ...
Setting up dpkg-dev (1.19.7ubuntu3) ...
Setting up gcc-9 (9.4.0-1ubuntu1-20.04.1) ...
Setting up gcc (4:9.3.0-1ubuntu2) ...
Setting up g++-9 (9.4.0-1ubuntu1-20.04.1) ...
Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.8ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.7) ...
miku@miku:~$ sudo apt install manpages-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
manpages-dev is already the newest version (5.05-1).
manpages-dev set to manually installed.
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
miku@miku:~$ gcc --version
gcc++: command not found
miku@miku:~$ g++ --version
g++ (Ubuntu 9.4.0-1ubuntu1-20.04.1) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

miku@miku:~$ make --version
GNU Make 4.2.1
Built for x86_64-pc-linux-gnu
Copyright (C) 1988-2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
miku@miku:~$
```

圖 1.輸入指令確認安裝是否成功

## 四、燒入晶片

### 下載ARC GNU Toolchain

Step1. 點選或git clone 下載套件壓縮檔案

```
https://
github.com/foss for synopsys drc arc
processors/toolchain/releases/download/arc 2020.09
release/arc_gnu_2020.09_prebuilt_elf32_le_linux_install.tar.gz
```

Step2. 解壓縮檔案

```
$ sudo tar xzvf
arc_gnu_2020.09_prebuilt_elf32_le_linux_install.tar.gz
```

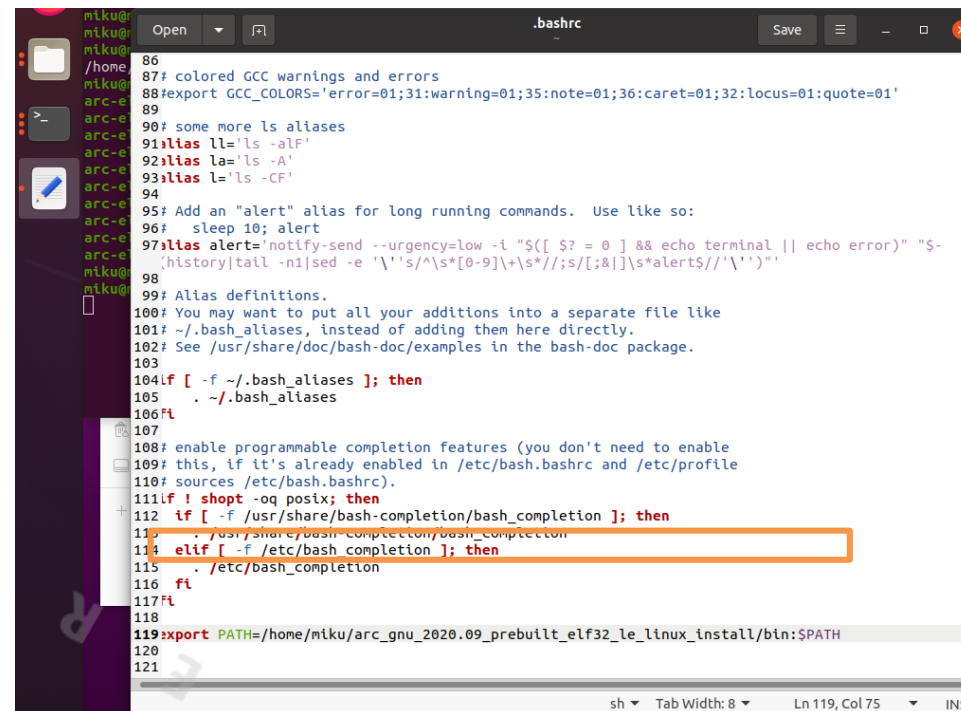
Step3. 更改環境變數

```
$ gedit ~/.bashrc
```

將下列加入檔案之中，並儲存

```
export PATH=/
home/miku/arc_gnu_2020.09_prebuilt_elf32_le_linux_install /bin :$PATH
```

```
$ source ~/.bashrc
```



```
.bashrc
86
87# colored GCC warnings and errors
88#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'
89
90# some more ls aliases
91alias ll='ls -alF'
92alias la='ls -A'
93alias l='ls -CF'
94
95# Add an "alert" alias for long running commands. Use like so:
96# sleep 10; alert
97alias alert='notify-send --urgency=low -i "${?} = 0" && echo terminal || echo error)' "$-
(history|tail -n1|sed -e '\s/\s*[0-9]\+\s*//;s/;/&|]\s*alert$//'\''"'
98
99# Alias definitions.
100# You may want to put all your additions into a separate file like
101# ~/.bash_aliases, instead of adding them here directly.
102# See /usr/share/doc/bash-doc/examples in the bash-doc package.
103
104if [ -f ~/.bash_aliases ]; then
105  . ~/.bash_aliases
106fi
107
108# enable programmable completion features (you don't need to enable
109# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110# sources /etc/bash.bashrc).
111if ! shopt -oq posix; then
112  if [ -f /usr/share/bash-completion/bash_completion ]; then
113    . /usr/share/bash-completion/bash_completion
114  elif [ -f /etc/bash_completion ]; then
115    . /etc/bash_completion
116  fi
117fi
118
119export PATH=/home/miku/arc_gnu_2020.09_prebuilt_elf32_le_linux_install/bin:$PATH
120
121
```

圖 1. Gedit Editing Tools

黃底紅字部分須改自己的路徑變數

## 四、燒入晶片

安裝git、curl、minicom與lrzsz

Step1. 輸入指令安裝

```
$ sudo apt update && sudo apt install git
$ sudo apt install curl
```

Step2. 輸入指令確認安裝是否成功

```
$ git --version
$ curl
```

Step3. 安裝minicom與lrzsz

```
$ sudo apt get install minicom
$ sudo apt get install lrzsz
```

Step3. 開啟minicom確認安裝是否成功

```
$ sudo minicom -s
```

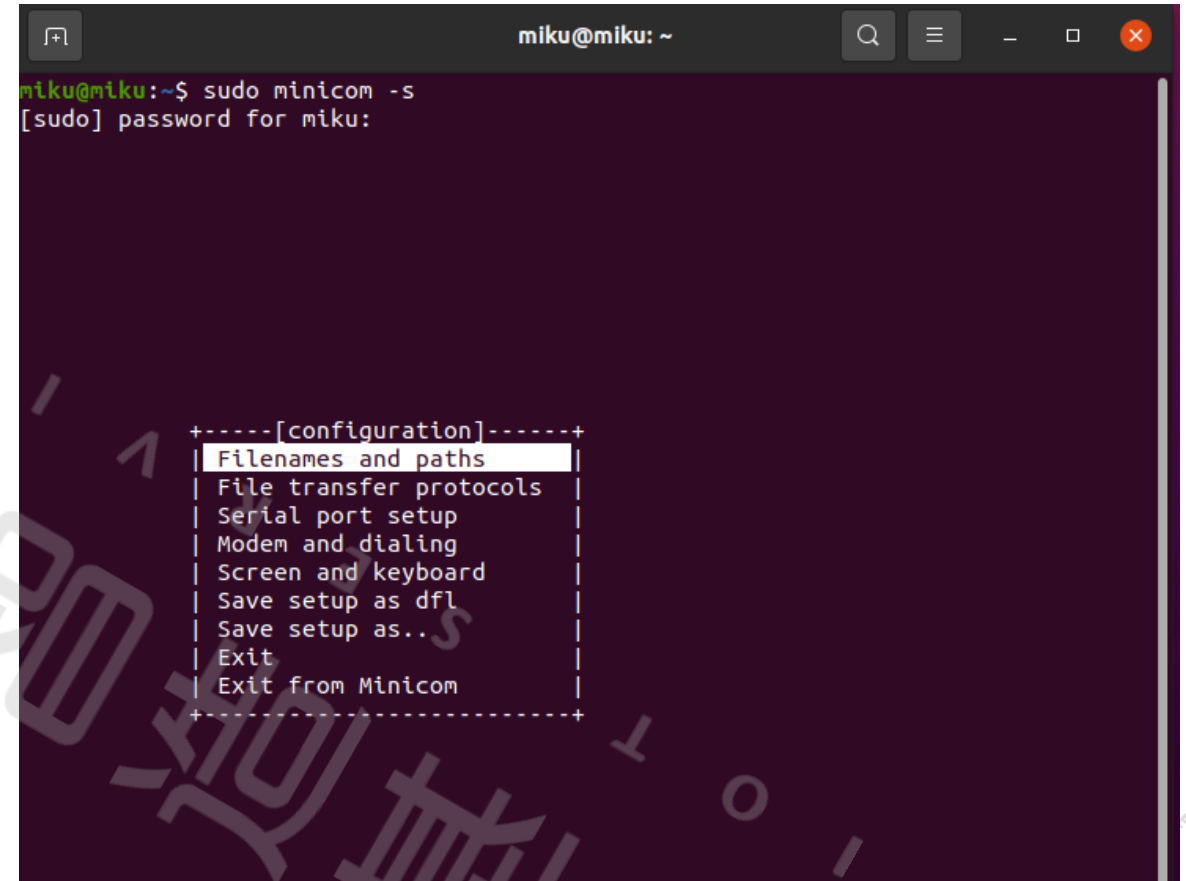


圖 1. 成功進入minicom串口通信工具

## 四、燒入晶片

連接WE-I Plus開發版

Step1. 將開發板接上電腦

Step2. 虛擬電腦設定輸入

Step3. 查詢Device ID 為 ttyUSBXXX

```
$ ls /sys/bus/usb-serial/devices/ -ltrah
```



圖 1. 兩個裝置皆要勾選

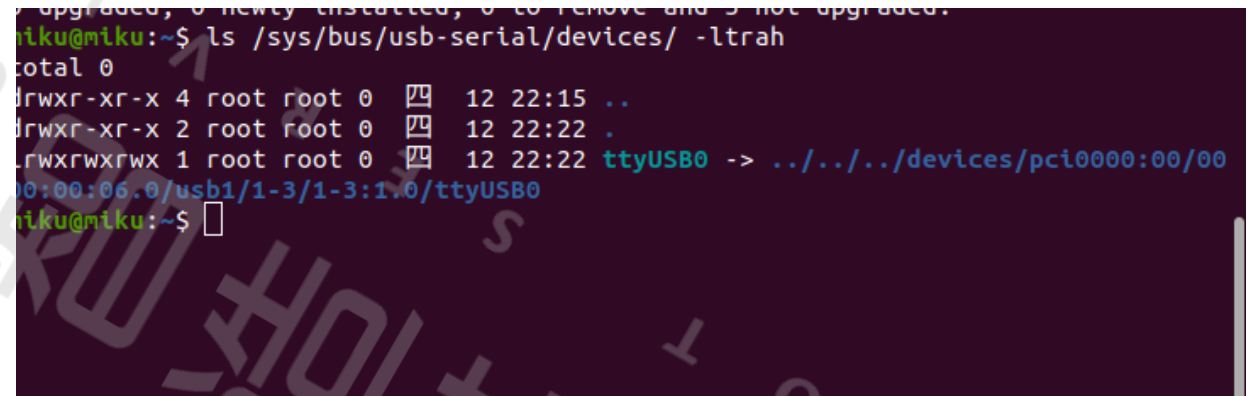


圖 2. 虛擬主機裡檢查是否有讀取到設備





## Minicom操作與設定

- Step1. 連線設定
- Step2. 更改設備id
- Step3. (可跳過)建議儲存紀錄,  
方便下次連線使用.
- Step4. 離開

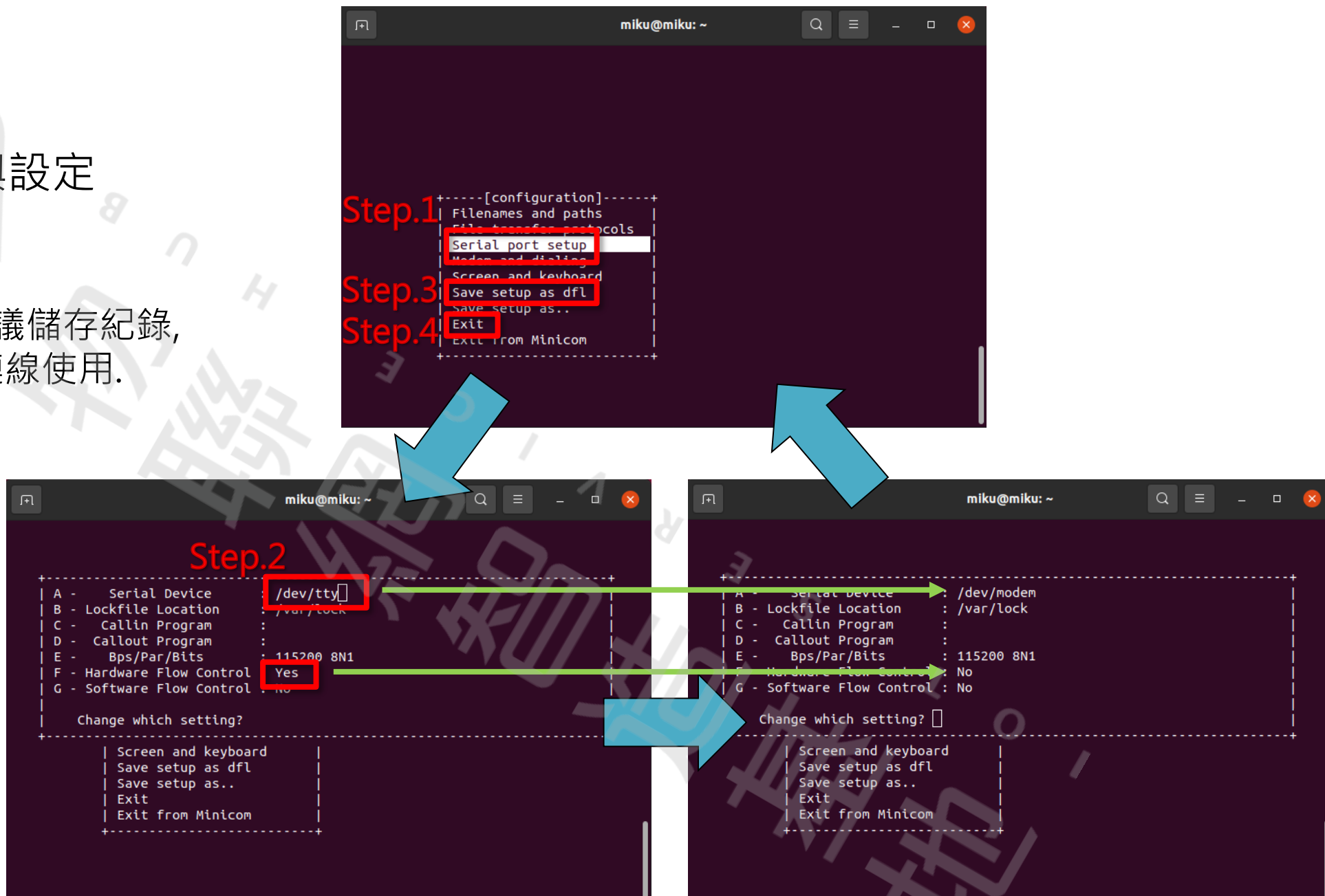


圖 1. Minicom設定

## 四、燒入晶片

將檔案燒入開發板中。

Step1. 下載檔案

```
$ git clone https://github.com/HimaxWiseEyePlus/bsp_tflu
```

Step2. 將第三章所製作出模型放入該資料夾(也可創立資料夾)

Step3. 下載相關第三方資料

```
$ make -C ../.. download
```

解壓縮第三章.zip檔案, 複製model-parameters、edge-impulse-sdk 和 tflite-model folders

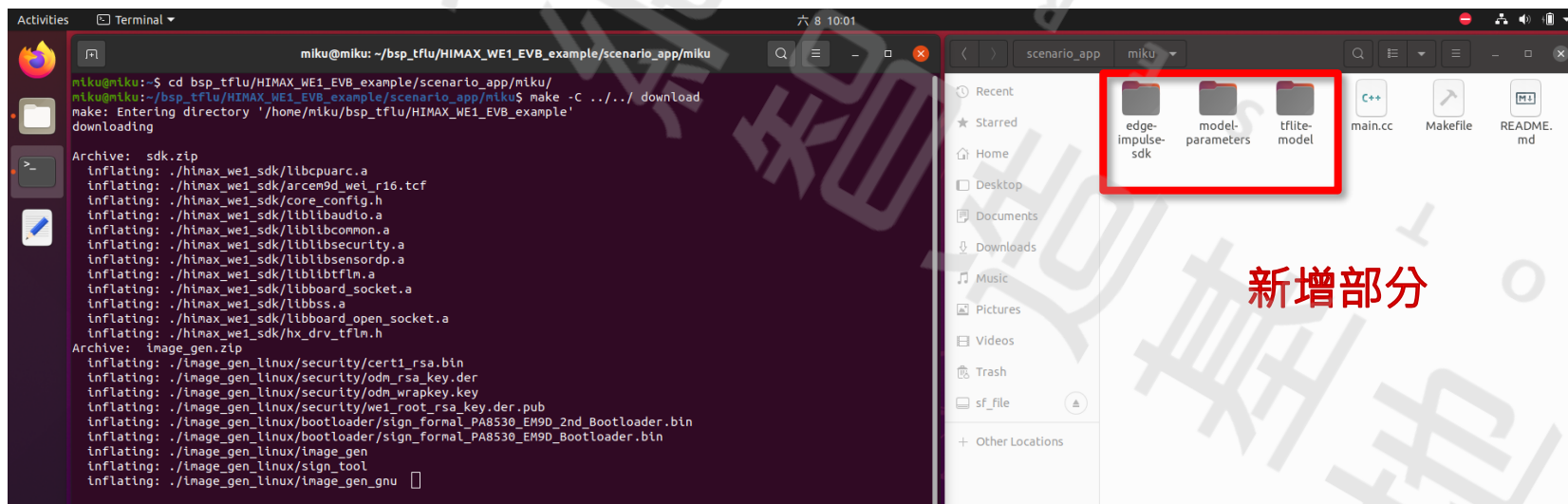


圖 1. 下載相關第三方資料

## 將編譯flash 映像檔

## Step1. cd 進入資料夾

## Step2. make 編譯

- \$ make clean
- \$ make all
- \$ make flash



## 四、燒入晶片

### 燒入WE-I Plus開發版

Step1.(如圖1)按下開發板上的重置鍵，並不斷按鍵盤的'1'鍵，會出現boot loader 版本訊息與

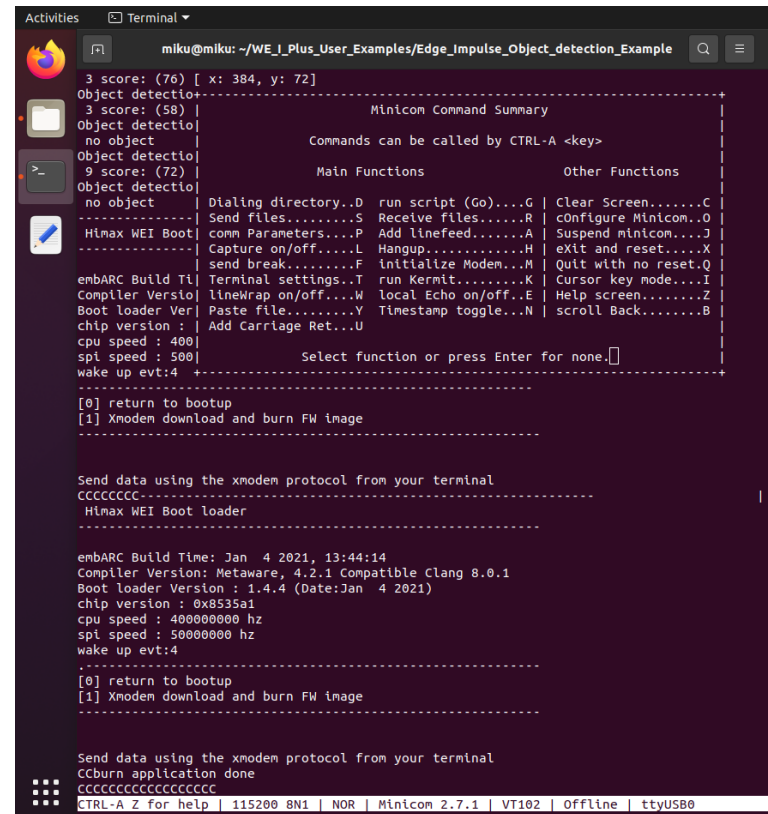
<<Send data using the xmodem protocol from your terminal>>

Step2.按下Ctrl+A 叫出 minicom menu

Step3.再按下'S'鍵跳出選單選擇'xmodem',  
按下'Enter'鍵選擇之。

Step4.使用 '空白' 鍵與 'Enter' 鍵選擇映像檔。

Step5.(如圖2)燒入完成後，按'Ctrl+A'，再Z跳出說明  
選擇'X' 'exit and reset'。



```

miku@miku: ~/WE_I_Plus_User_Examples/Edge_Impulse_Object_detection_Example
3 score: (76) [ x: 384, y: 72]
Object detection:
3 score: (58)
Object detection:
no object
Object detection:
9 score: (72)
Object detection:
no object
Main Functions
Dialing directory..D
Send files.....S
comm Parameters....P
Capture on/off....L
send break.....F
embARC Build Tl
Compiler Versio
Boot loader Ver
chip version :
cpu speed : 400
spi speed : 500
wake up evt:4
Other Functions
run script (Go)....G
Receive files.....R
Add linefeed.....A
Hangup.....H
Initialize Modem...M
run Kermit.....K
local Echo on/off..E
Timestamp toggle...N
Clear Screen.....C
cOnfigure Minicom..O
Suspend minicom...J
eXit and reset.....X
Quit with no reset.Q
Cursor key mode....I
Help screen.....Z
scroll Back.....B
Select function or press Enter for none.

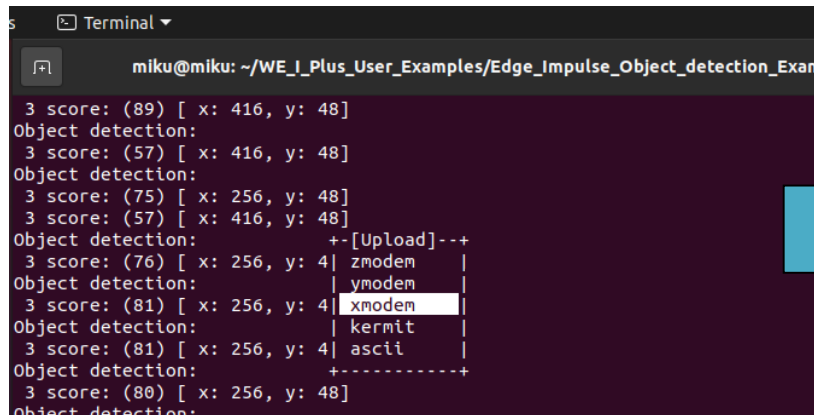
[0] return to bootup
[1] Xmodem download and burn FW image

Send data using the xmodem protocol from your terminal
CCCCCCCCCCCCCCCC
Himax WEI Boot loader

embARC Build Time: Jan  4 2021, 13:44:14
Compiler Version: Metaware, 4.2.1 Compatible Clang 8.0.1
Boot loader Version : 1.4.4 (Date:Jan  4 2021)
chip version : 0x8535a1
cpu speed : 400000000 hz
spi speed : 50000000 hz
wake up evt:4
[0] return to bootup
[1] Xmodem download and burn FW image

Send data using the xmodem protocol from your terminal
CCCCCCCCCCCCCCCC
CTRL-A Z for help | 115200 BN1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0
  
```

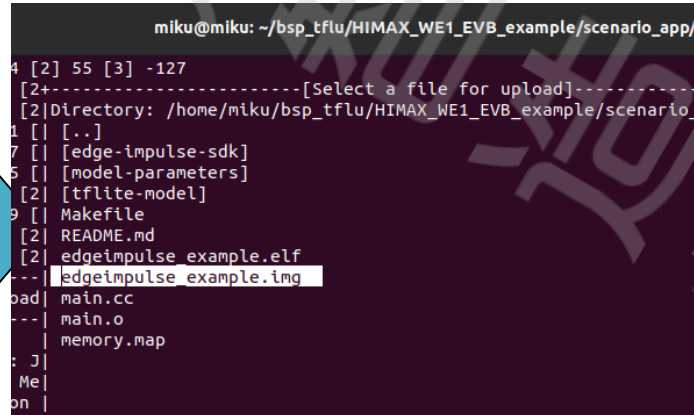
圖 1. Ctrl+A 後再按'Z'可進入說明選單



```

miku@miku: ~/WE_I_Plus_User_Examples/Edge_Impulse_Object_detection_Example
3 score: (89) [ x: 416, y: 48]
Object detection:
3 score: (57) [ x: 416, y: 48]
Object detection:
3 score: (75) [ x: 256, y: 48]
3 score: (57) [ x: 416, y: 48]
Object detection:
++[Upload]++
3 score: (76) [ x: 256, y: 4]
Object detection:
3 score: (81) [ x: 256, y: 4]
Object detection:
3 score: (81) [ x: 256, y: 4]
Object detection:
3 score: (80) [ x: 256, y: 48]
Object detection:
  
```

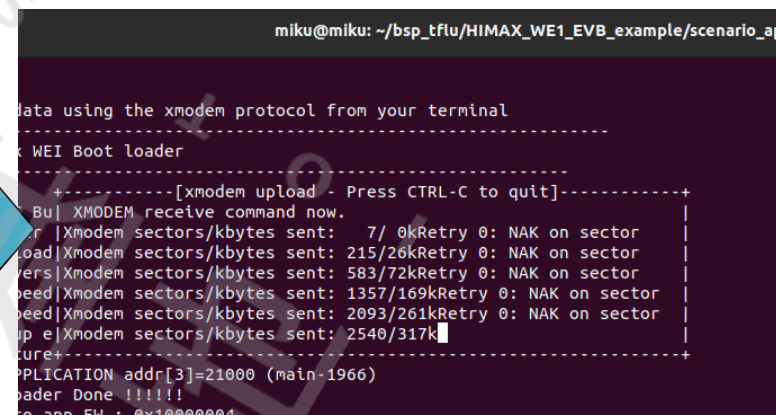
圖 2. 選擇'xmodem'



```

miku@miku: ~/bsp_tflu/HIMAX_WE1_EVB_example/scenario_app/n
[2] 55 [3] -127
[2+]-----[Select a file for upload]-----
[2]Directory: /home/miku/bsp_tflu/HIMAX_WE1_EVB_example/scenario_
1 [ [..]
7 [ [edge-impulse-sdk]
5 [ [model-parameters]
[2] [tflite-model]
9 [ Makefile
[2] README.md
[2] edgeimpulse example.elf
---| edgeimpulse example.img
load| main.cc
---| main.o
| memory.map
: J]
Me|
on |
  
```

圖 3. 使用空白與'Enter'鍵選擇.img



```

miku@miku: ~/bsp_tflu/HIMAX_WE1_EVB_example/scenario_app
data using the xmodem protocol from your terminal
-----
x WEI Boot loader
-----
++-----[xmodem upload - Press CTRL-C to quit]-----++
Bu| XMODEM receive command now.
r |Xmodem sectors/kbytes sent: 7/ 0kRetry 0: NAK on sector
oad|Xmodem sectors/kbytes sent: 215/26kRetry 0: NAK on sector
vers|Xmodem sectors/kbytes sent: 583/72kRetry 0: NAK on sector
peed|Xmodem sectors/kbytes sent: 1357/169kRetry 0: NAK on sector
eed|Xmodem sectors/kbytes sent: 2093/261kRetry 0: NAK on sector
p e|Xmodem sectors/kbytes sent: 2540/317k
ture+
APPLICATION addr[3]=21000 (main-1966)
loader Done !!!!!
  
```

圖 4. 成功燒入中畫面



# 五、結果展示

上述可以115200鮑率輸出至下位控制器

格式為:

```
("[0] %d [1] %d [2] %d [3] %d [4] %d [5] %d [6] %d\n", score[0], score[1], score[2], score[3], score[4], score[5], score[6]);
```

[標籤] 分數 [標籤] 分數 [標籤] 分數

```
THERA[0] 44 [1] -77 [2] -122 [3] -121 [4] -108 [5] -128 [6] -128
E[0] -63 [1] 26 [2] -123 [3] -124 [4] -101 [5] -128 [6] -128
E[0] -89 [1] 55 [2] -124 [3] -123 [4] -103 [5] -128 [6] -128
[1] -7 [2] -123 [3] -121 [4] -98 [5] -128 [6] -128
E[0] -56 [1] 17 [2] -122 [3] -121 [4] -102 [5] -128 [6] -128
```



圖 1. 序列傳出輸出結果至電腦

圖 2. 打印傳出部分

下位控制器可依照結果判斷使用

## 五、結果展示

### Demo時間



#### Demo影片:

<https://www.youtube.com/watch?v=S76xsM41OJ0>

<https://www.youtube.com/watch?v=DD7FauRRzos>

#### 程式位置:

<https://github.com/wildman8606/-WE-I-Plus-Smart-microscope/blob/main/main.cc>

#### 講師聯絡:

<https://www.facebook.com/peter8606/>

### 電路圖

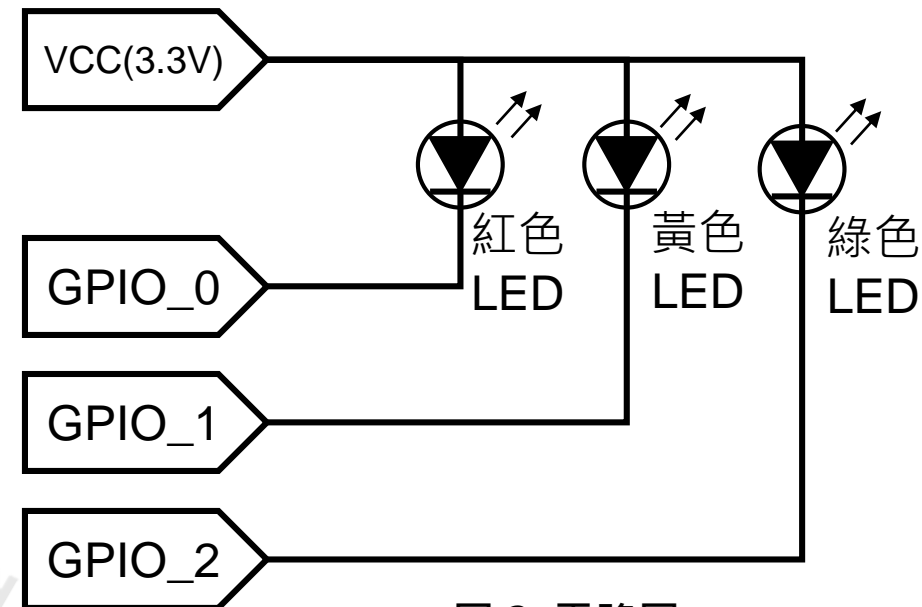


圖 2. 電路圖

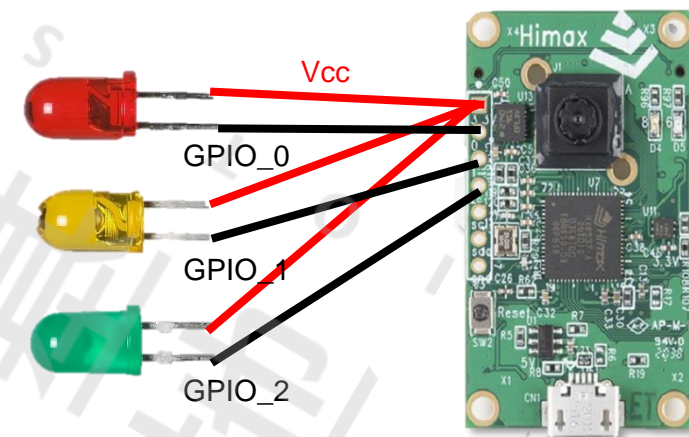


圖 3. 實體圖



## 六、結果展示

或是我們這邊改寫範例

每個標籤打印出來  
並顯示不同燈號

```

72 if (maxvalue > 0){
73     switch (maxindex)
74     {
75     case 0:
76         // ANADENANTHERA
77         gpio_config.gpio_pin = HX_DRV_PGPI0_0;
78         gpio_config.gpio_data = 1;
79         hx_drv_gpio_set(&gpio_config);
80         hx_drv_uart_print("ANADENANTHERA");
81         break;
82     case 1:
83         // ARECACEAE
84         gpio_config.gpio_pin = HX_DRV_PGPI0_1;
85         gpio_config.gpio_data = 1;
86         hx_drv_gpio_set(&gpio_config);
87         hx_drv_uart_print("ARECACEAE");
88         break;
89     case 2:
90         // CHROMOLAENA
91         gpio_config.gpio_pin = HX_DRV_PGPI0_0;
92         gpio_config.gpio_data = 1;
93         hx_drv_gpio_set(&gpio_config);
94         gpio_config.gpio_pin = HX_DRV_PGPI0_1;
95         gpio_config.gpio_data = 1;
96         hx_drv_gpio_set(&gpio_config);
97         hx_drv_uart_print("CHROMOLAENA");

```

```

*main.cc
~/bsp_tflu/HIMAX_WE1_EVB_example/scenario_app/edgeimpulse-example
Save

1 #include "hx_drv_tflm.h"
2 #include "tflite-model/trained_model_compiled.h"
3
4 namespace
5 {
6     hx_drv_sensor_image_config_t g_pimg_config;
7     hx_drv_gpio_config_t gpio_config;
8     int input_width = 160;
9     int input_height = 160;
10    int input_channels = 1;
11 }
12
13 TfliteStatus GetImage(int image_width, int image_height, int channels, int8_t *image_data)
14 {
122     gpio_config.gpio_data = 1;
123     hx_drv_gpio_set(&gpio_config);
124     hx_drv_uart_print("SERJANIA");
125     break;
126
127     case 6:
128         // UNCERTAIN
129         gpio_config.gpio_pin = HX_DRV_PGPI0_2;
130         gpio_config.gpio_data = 1;
131         hx_drv_gpio_set(&gpio_config);
132         gpio_config.gpio_pin = HX_DRV_PGPI0_1;
133         gpio_config.gpio_data = 1;
134         hx_drv_gpio_set(&gpio_config);
135         gpio_config.gpio_pin = HX_DRV_PGPI0_0;
136         gpio_config.gpio_data = 1;
137         hx_drv_gpio_set(&gpio_config);
138         hx_drv_uart_print("UNCERTAIN");
139         break;
140     }
141
142     // show the scores to UART
143     hx_drv_uart_print("[0] %d [1] %d [2] %d [3] %d [4] %d [5] %d [6] %d\n", score[0], score[1], score[2], score[3],
144
145 int main(void)
146 {
147     // initial uart and gpio
148     hx_drv_uart_initial(UART_BR_115200);
149     gpio_config.gpio_pin = HX_DRV_PGPI0_0;

```

更改成自己的模型  
輸入圖片大小

將尾端打印結果改成  
自己標籤數量

# 六、補充資料

第三章Ubuntu安裝於這詳細補充給第一次使用朋友 PEKO



圖 1. Choose a language and click Install Ubuntu

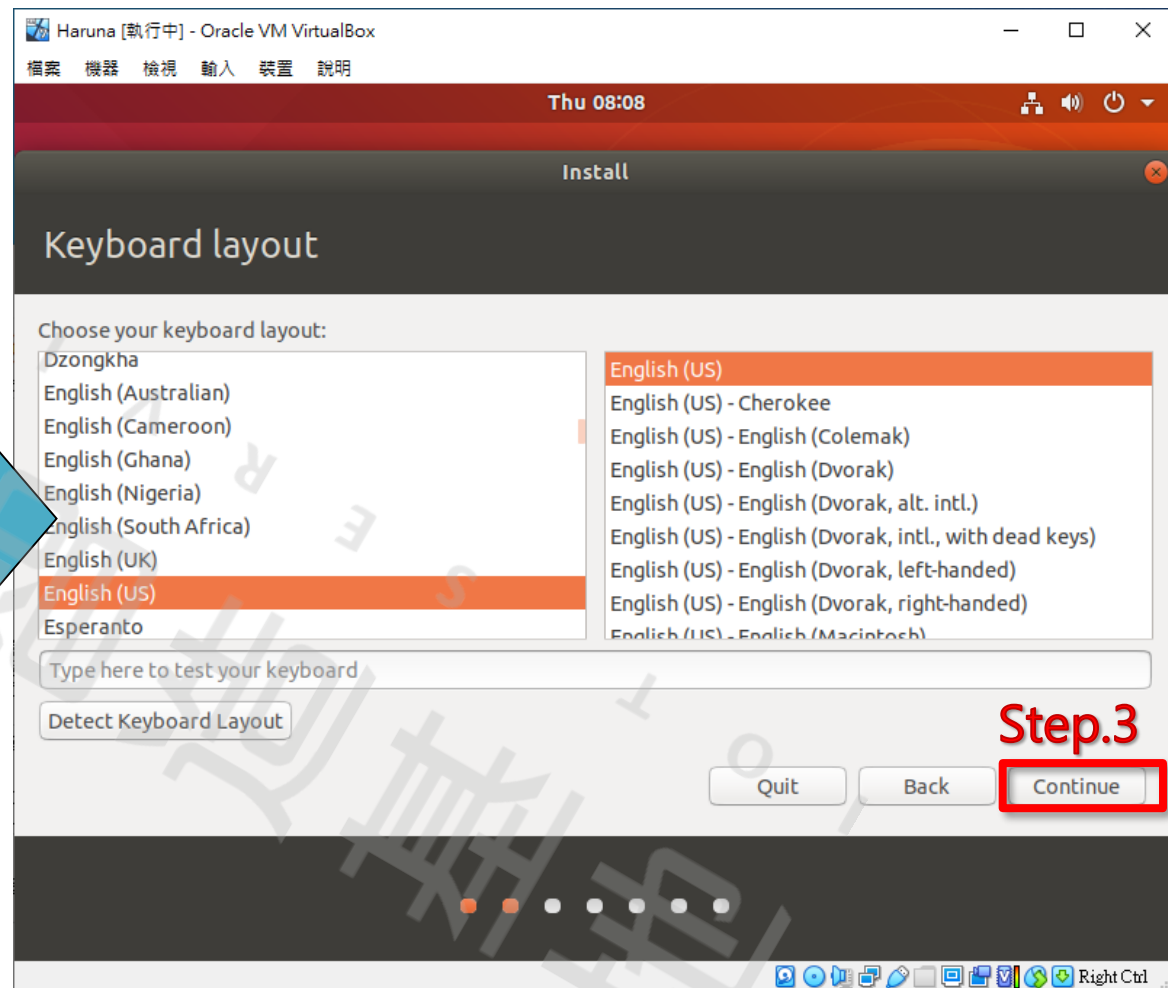


圖 2. Click Continue



## 六、補充資料

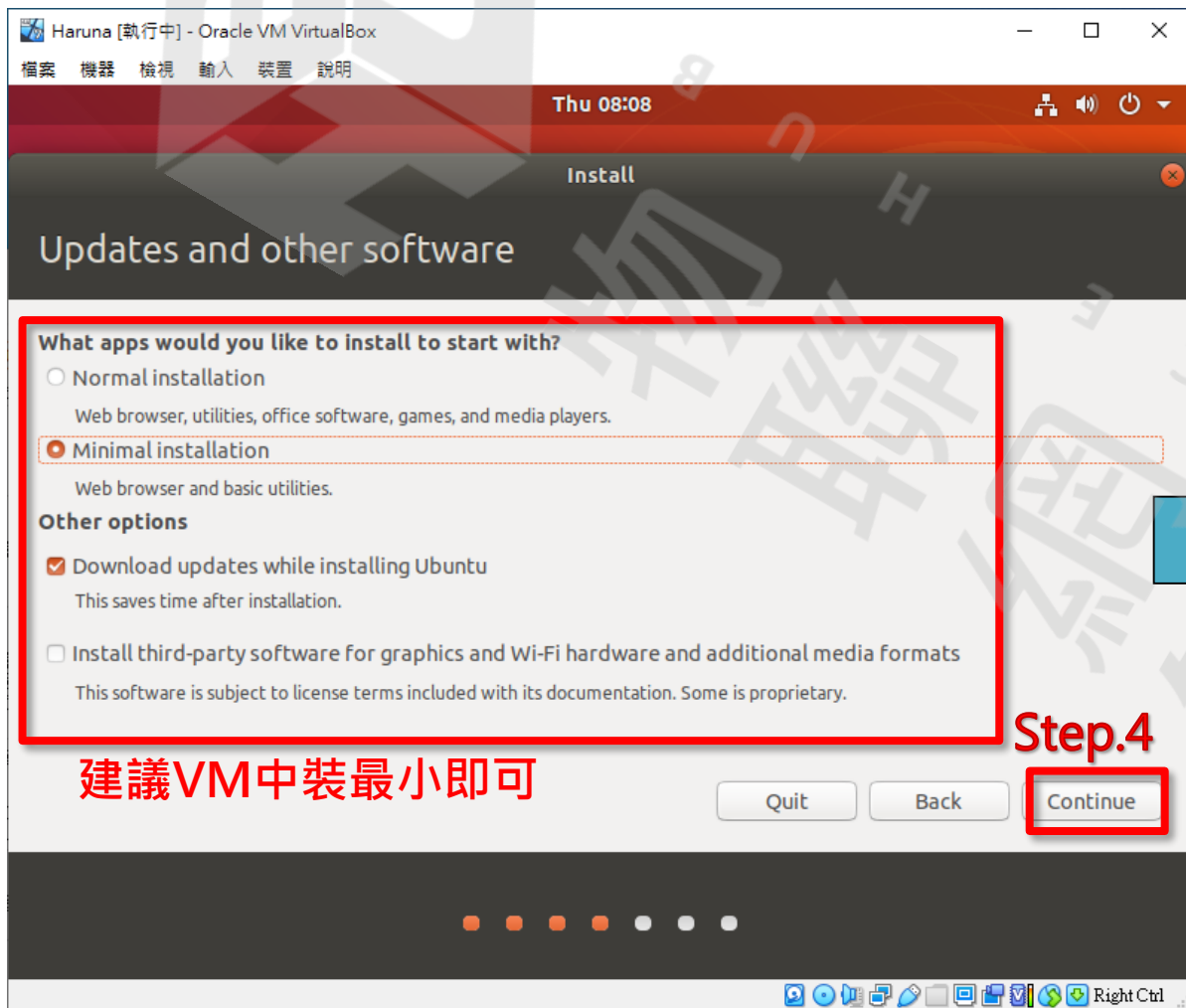


圖 1. Setting and click Continue

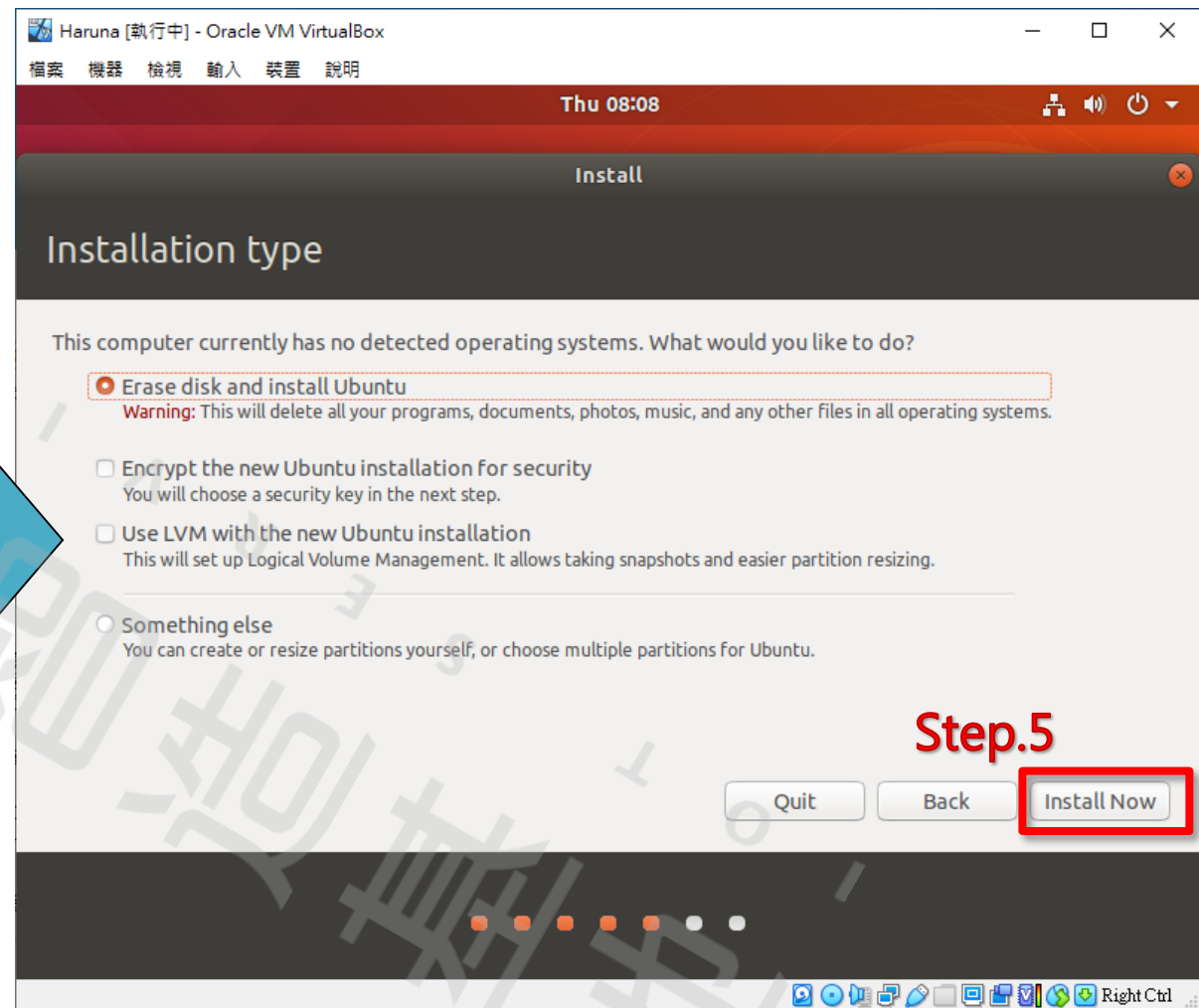
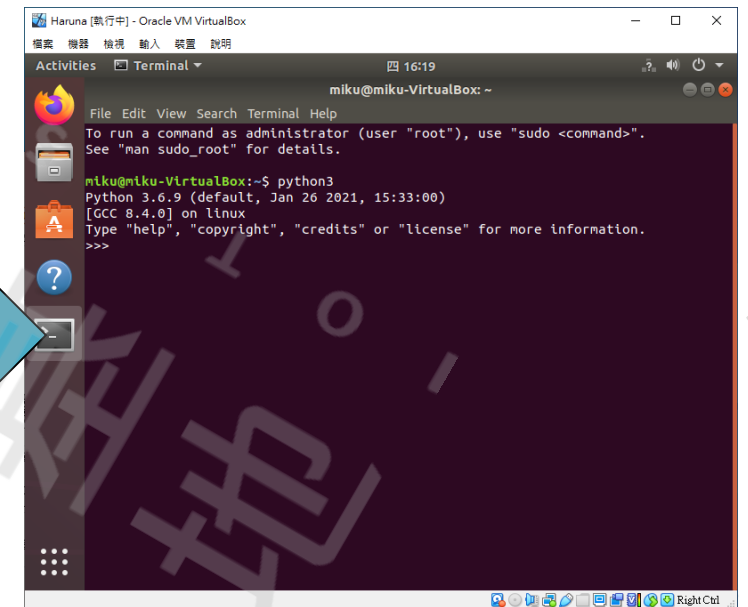
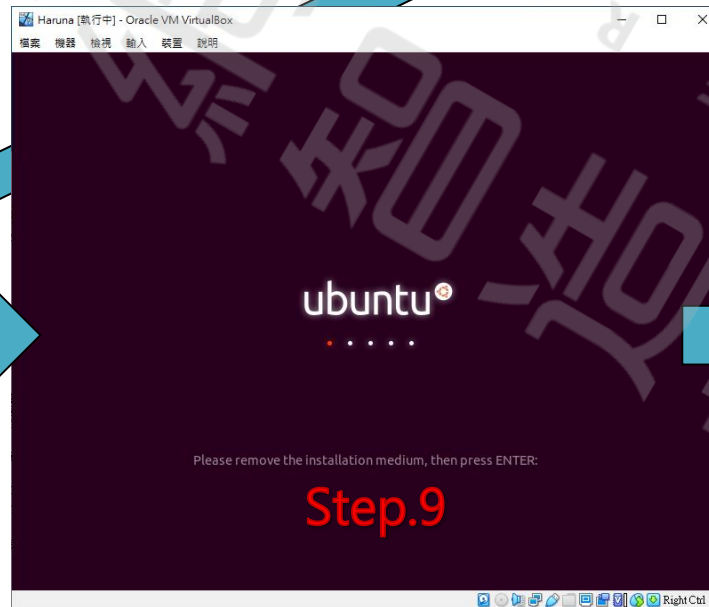
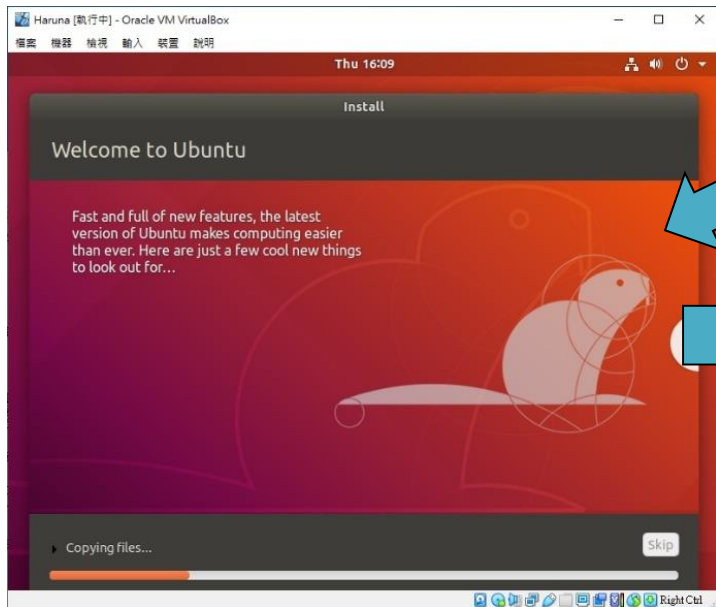
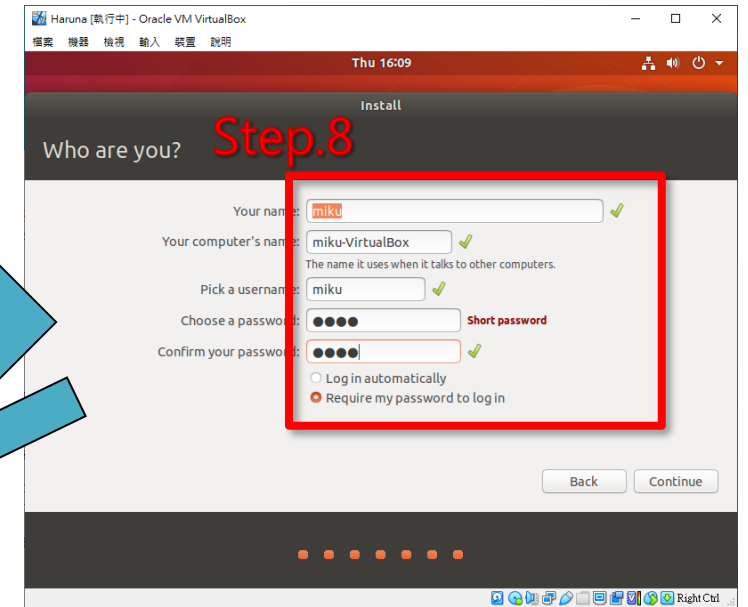
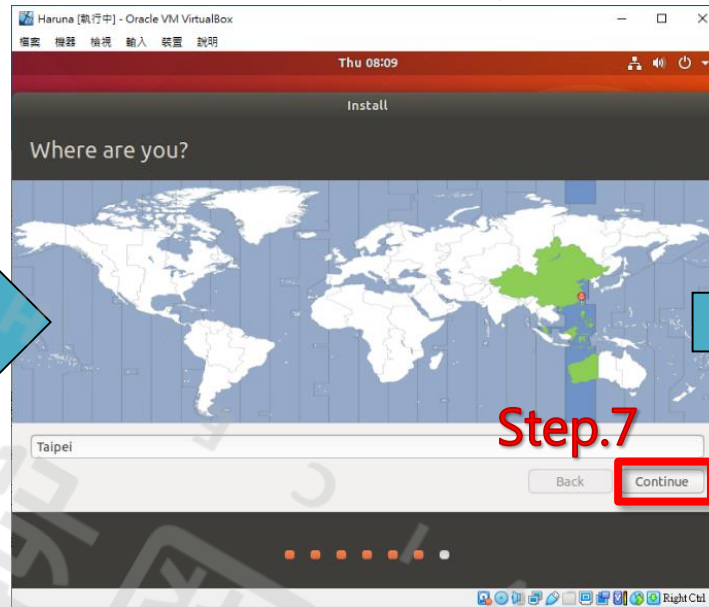
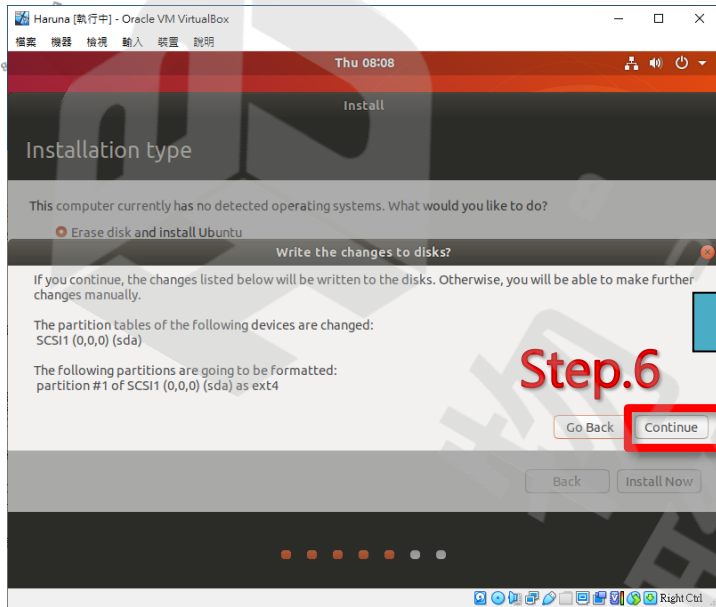


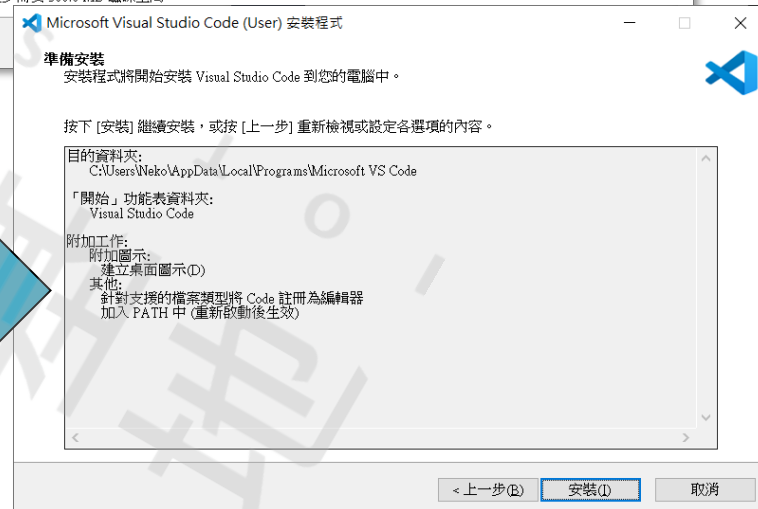
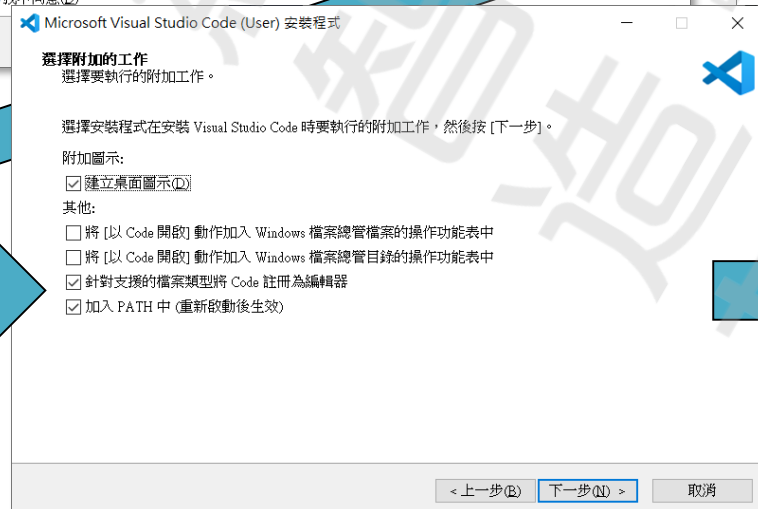
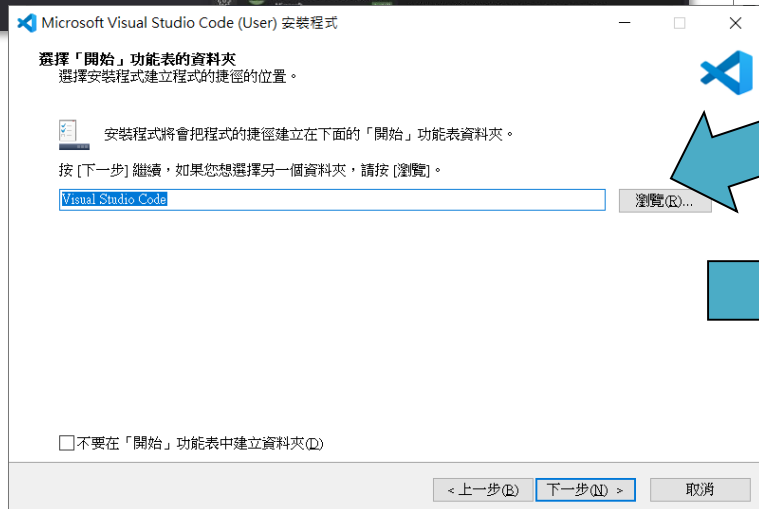
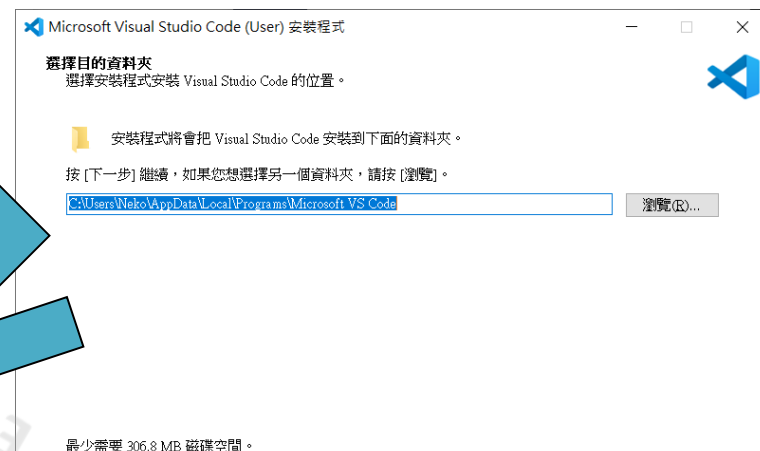
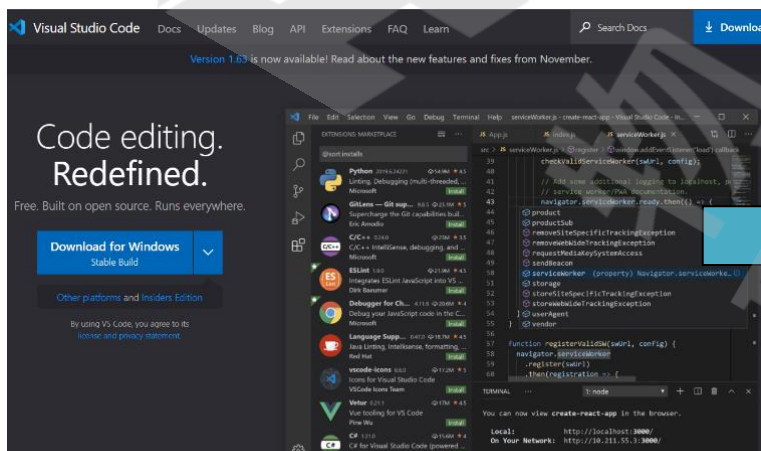
圖 2. Setting and click Install Now

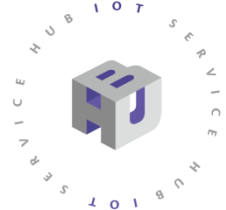
## 六、補充資料



# 六、補充資料

程式編輯可選Visual Studio Code 或 NotePad++, 甚至有TXT大法  
這邊補充較常用的VScode安裝





更多  
國產IC解決方案



Thank you



物聯網智造基地

IOT SERVICE HUB

