

COREMAKER-01

SAMPLE CODE 運行指南

目錄

WiFi Sample Code 運行指南.....	2
事前準備	2
程式碼修改.....	2
編譯.....	3
燒錄.....	4
接收訊息	5
SD Card Sample Code 運行指南	6
事前準備	6
程式碼修改.....	6
編譯.....	7
燒錄.....	8
檢查檔案	9
LED 燈效控制範例	10
程式碼修改.....	10
編譯.....	11
燒錄.....	12
檢查結果	13
Visual Studio Code 編譯環境設定說明	14
增加設定檔.....	14
安裝延伸套件	16
進行編譯	16
Document History	18

WiFi Sample Code 運行指南

● 事前準備

1. 一台 WiFi router，或是可以進行 WiFi 連線的環境。
2. NetAssist 或其他可以建立 TCP 連線的工具軟體，本文將以 NetAssist 為範例進行說明。

● 程式碼修改

1. 修改 CMakeLists.txt 第 38 行，將 target_sources 中 “main.cpp” 改為 “wifi_example.cpp”。

```
36 target_sources(${APP_TARGET}
37     PRIVATE
38         wifi_example.cpp
39         DebounceIn.cpp
40         lightEffect.hpp
41         jWrite.c
42         sensors/ADS131E.cpp
43         sensors/GMC306.cpp
44         sensors/AcousticNode.cpp
45         sensors/SensorHub.cpp
46         sensors/BME680/mbed_bme680.cpp
47         sensors/BME680/bme680.cpp
48         sensors/GMP102/mbed_gmp102.cpp
49         sensors/GMP102/gmp102.cpp
50         sensors/KX122-1037/KX122.cpp
51         sd_card/NuSDBlockDevice.cpp
52     )
```

2. 修改 mbed_app.json 第 22 及 26 行，將 wifi-ssid 的 value 改為欲連線的 WiFi 裝置名稱，wifi-password 的 value 改為欲連線的 WiFi 裝置連線密碼。

```
19     "config": {
20         "wifi-ssid": {
21             "help"                : "Enter SSID here",
22             "value"               : "\"ASUS_437\""
23         },
24         "wifi-password": {
25             "help"                : "Enter PASSWORD here",
26             "value"               : "\"5555555555\""
27         }
28     }
```

- 修改 wifi_example.cpp 第 53 行，將括號內參數改寫為欲連線的 TCP server IP address 及 port。

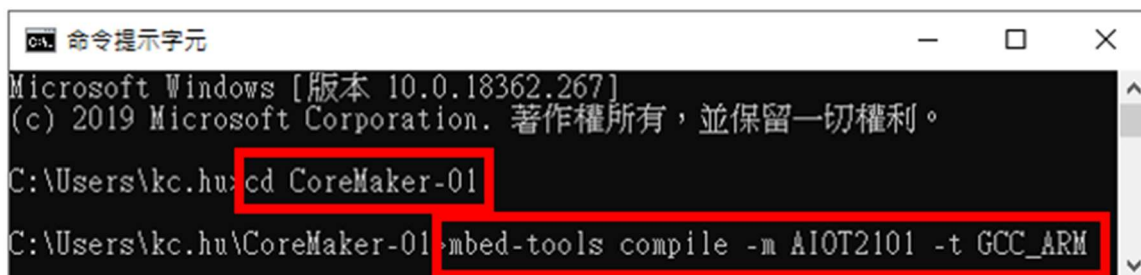
```
51  ATCmdParser *_parser;
52  OPL1000Interface wifi(PB_15, PB_14, 0);
53  SocketAddress a("192.168.50.11", 8181); //TCP server address & port.
```

● 編譯

- 點擊 “開始” ，輸入 “cmd” ，開啟 “命令提示字元” 。



- 輸入 “cd CoreMaker-01” 進入韌體程式所在的資料夾。
- 輸入 “mbed-tools compile -m AIOT2101 -t GCC_ARM” 開始編譯，若是初次編譯會需要較長的時間。

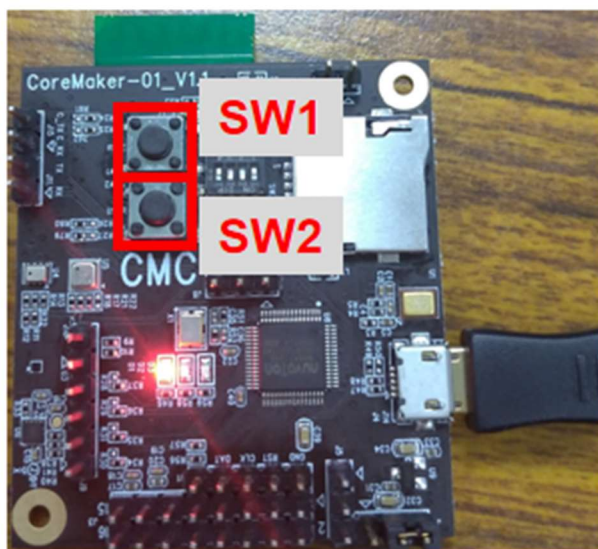


4. 編譯完成會在 CoreMaker-01\cmake_build\AIOT2101\develop\GCC_ARM\目錄下，產生 AIOT_2101.bin 檔案。

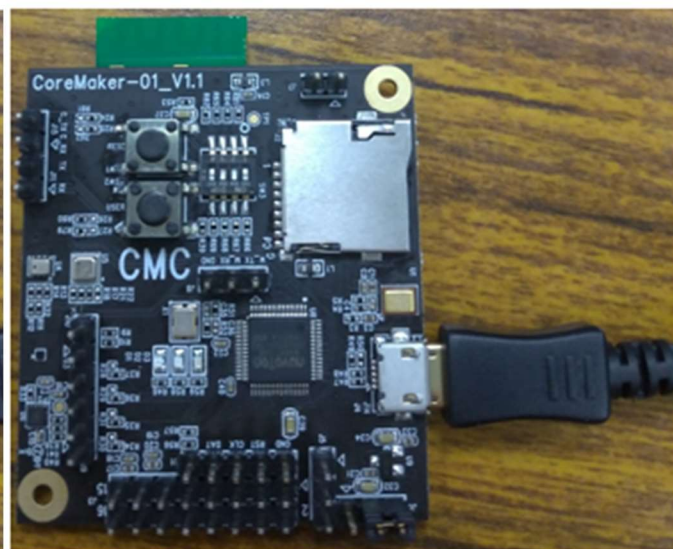
```
命令提示字元
-- built: C:/Users/kc.hu/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.bin
-- built: C:/Users/kc.hu/CoreMaker-01/cmake_build/AIOT2101/develop/GCC_ARM/AIOT_2101.hex
C:\Users\kc.hu\CoreMaker-01>
```

● 燒錄

1. 使用 Micro USB to USB 轉接線，連接 CoreMaker 與 PC。
2. 同時按下 SW1 及 SW2 按鍵。
3. 先放開 SW1，再放開 SW2。
4. 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 2 跟 3。



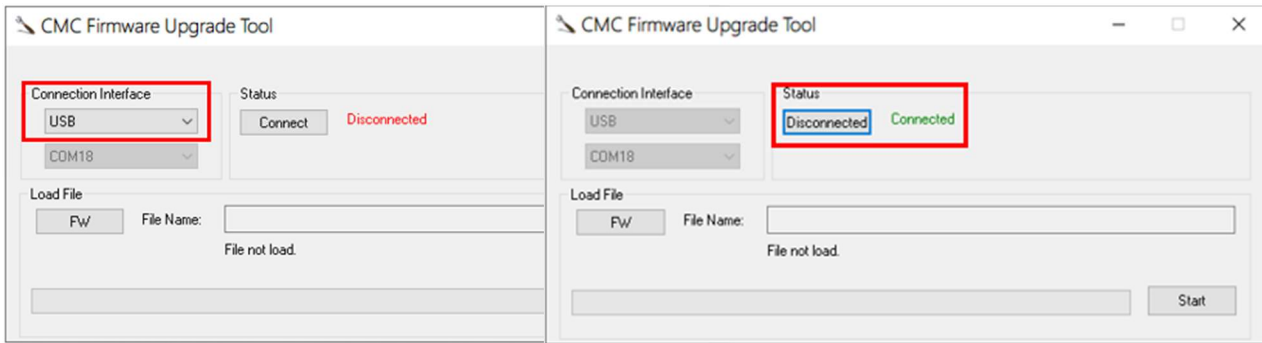
一般模式



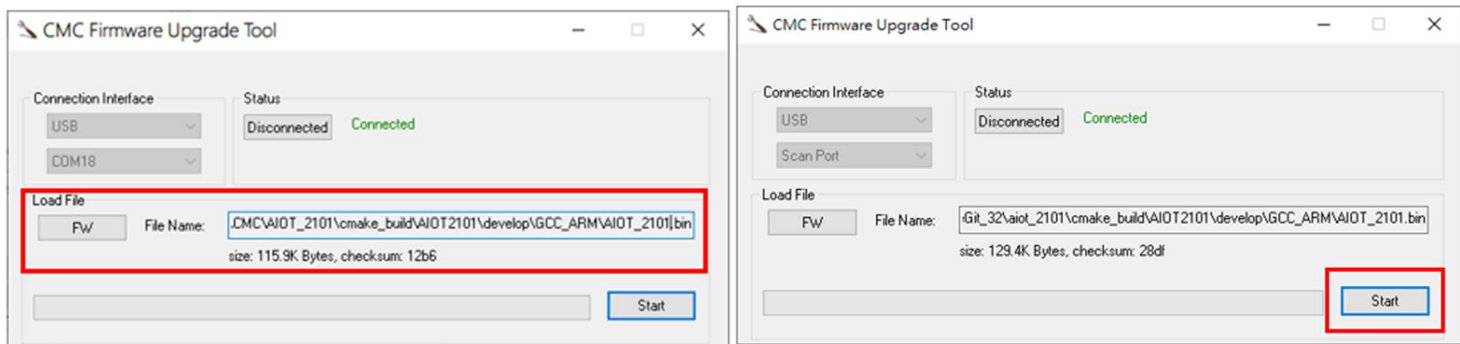
燒錄模式

5. 解壓縮 CoreMaker-01/CMC_ISP/CMC_ISP.zip，執行解開後的 CMC_ISP.exe 開啟程式。

6. 在 Connection Interface 選單選擇 “USB” ，點擊 “Connect” ，確認連線成功。



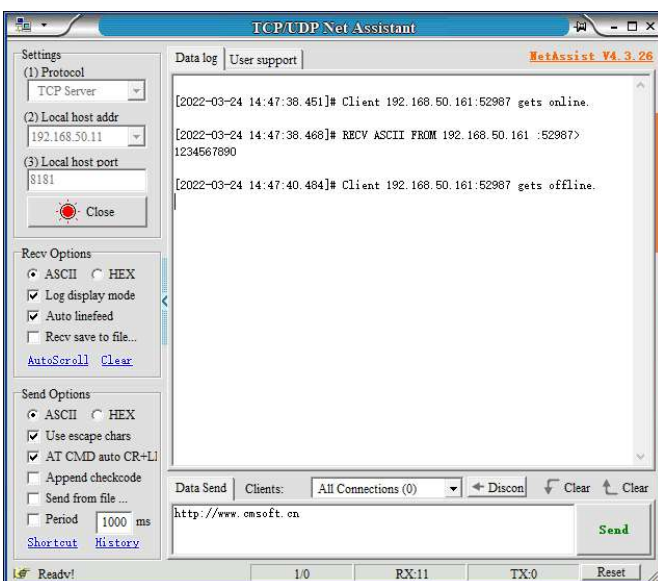
7. 選擇編譯完成的 AIOT_2101.bin 檔案，點擊 “Start” 開始燒錄。



● 接收訊息

1. 開啟 NetAssist，軟體會自動偵測 PC 的 IP address，設定完 Local host port 後，點擊 “Open” 開始運行。

2. CoreMaker 連上 TCP server 後，預設會送出 “1234567890” ，送出後中斷連線。



SD Card Sample Code 運行指南

- 事前準備

一張檔案系統為 FAT32 的 micro SD card

- 程式碼修改

修改 CMakeLists.txt 第 38 行，將 target_sources 中 “main.cpp” 改為
“sdcard_example.cpp”。

```
36 target_sources(${APP_TARGET}  
37     PRIVATE  
38     sdcard_example.cpp  
39     DebounceIn.cpp  
40     lightEffect.hpp  
41     jWrite.c  
42     sensors/ADS131E.cpp  
43     sensors/GMC306.cpp  
44     sensors/AcousticNode.cpp  
45     sensors/SensorHub.cpp  
46     sensors/BME680/mbed_bme680.cpp  
47     sensors/BME680/bme680.cpp  
48     sensors/GMP102/mbed_gmp102.cpp  
49     sensors/GMP102/gmp102.cpp  
50     sensors/KX122-1037/KX122.cpp  
51     sd_card/NuSDBlockDevice.cpp  
52 )
```

● 編譯

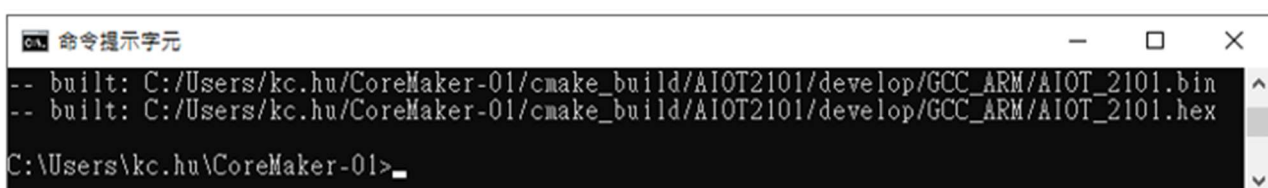
1. 點擊 “開始” ，輸入 “cmd” ，開啟 “命令提示字元” 。



2. 輸入 “cd CoreMaker-01” 進入韌體程式所在的資料夾。
3. 輸入 “mbed-tools compile -m AIOT2101 -t GCC_ARM” 開始編譯，若是初次編譯會需要較長的時間。

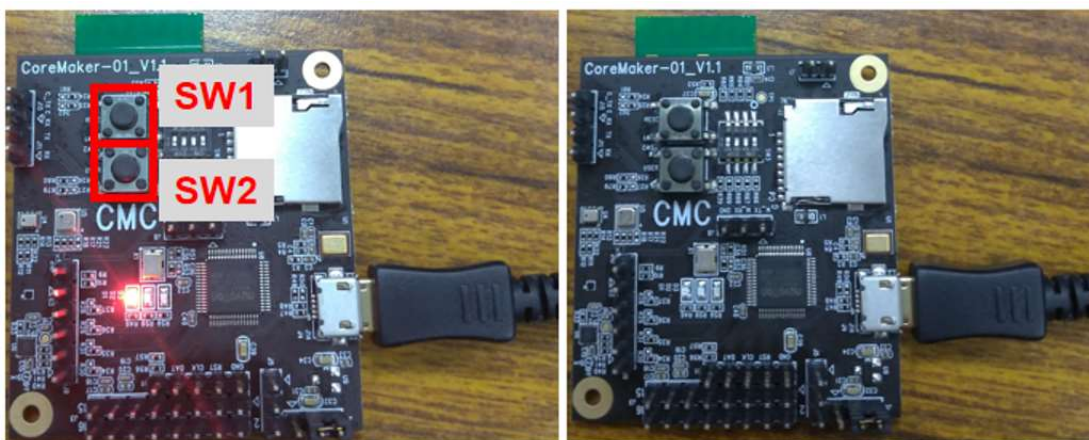


4. 編譯完成會在 CoreMaker-01\cmake_build\AIOT2101\develop\GCC_ARM\目錄下，產生 AIOT_2101.bin 檔案。



● 燒錄

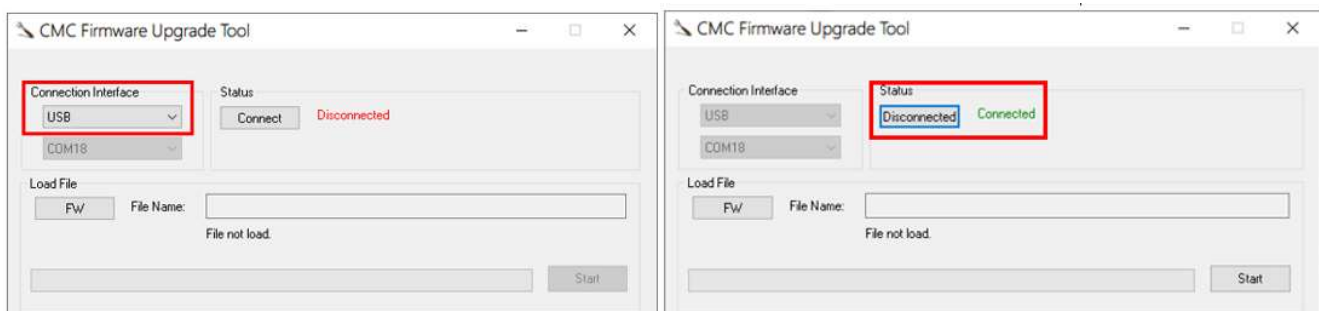
1. 使用 Micro USB to USB 轉接線，連接 CoreMaker 與 PC。
2. 同時按下 SW1 及 SW2 按鍵。
3. 先放開 SW1，再放開 SW2。
4. CoreMaker 上的紅色 LED 不再閃爍，表示進入燒錄模式，若紅色 LED 仍然閃爍，重複步驟 2 跟 3。



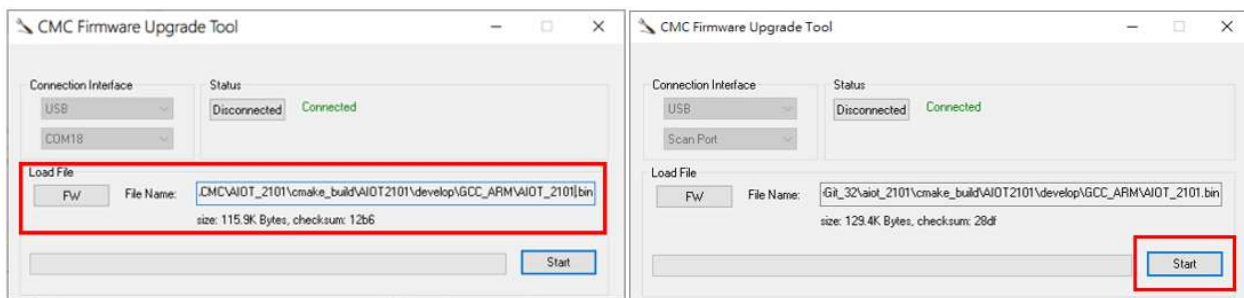
一般模式

燒錄模式

5. 解壓縮 CoreMaker-01/CMC_ISP/CMC_ISP.zip，執行解開後的 CMC_ISP.exe 開啟程式。
6. 在 Connection Interface 選單選擇 “USB”，點擊 “Connect”，確認連線成功。



7. 選擇編譯完成的 AIOT_2101.bin 檔案，點擊 “Start” 開始燒錄。



● 檢查檔案

使用讀卡機或是其他可以讀取 micro SD card 的方式，查看是否有產生一個名為 “number.txt” 的文字檔，且檔案中包含數字 0 到 9。若有，則 SD card 記錄功能正常。

LED 燈效控制範例

CoreMaker 提供三種顏色的 LED，分別是紅色、綠色、藍色，以下將以控制綠色 LED 作為範例。

● 程式碼修改

1. 使用編輯器開啟 CoreMaker-01/main.cpp。
2. 在第 102 行增加 “led_g.Flash(100);”。此處的 led_g 是控制綠色 LED 的物件，Flash 是閃爍功能，括號內的數字表示閃爍的間隔，單位為毫秒 (millisecond, ms)，所以此行程式表示綠色 LED 每 0.1 秒閃爍一次。

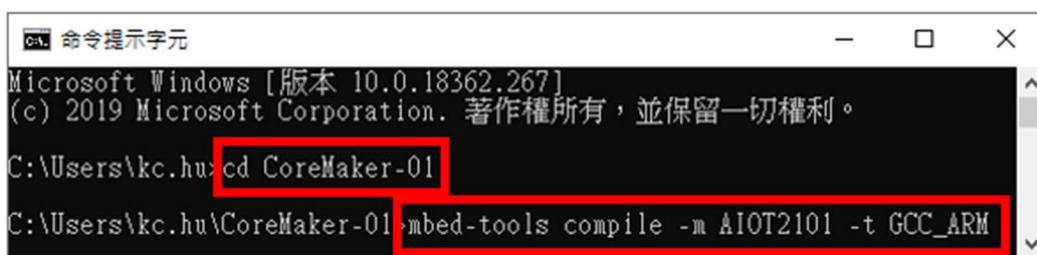
```
87     while (1)
88     {
89         uint flags = mainEvent.wait_any(0xFFFF, 1000);
90         if (!(flags & osFlagsError))
91         {
92             if (flags & SW3_EVENT)
93                 SwitchChanged();
94             if (flags & SW2_EVENT)
95                 sensorHub.ButtonPress();
96             if (flags & UART_EVENT)
97                 onSerialReceived();
98         }
99         else if (flags == osFlagsErrorTimeout) // No event
100         {
101             }
102         led_g.Flash(100);
103     }
104
105     return 0;
106 }
```

● 編譯

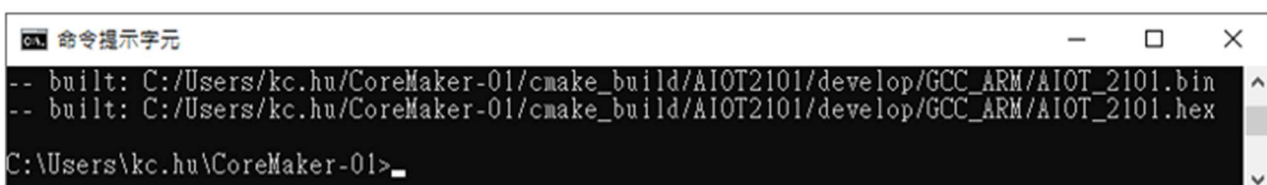
1. 點擊 “開始” ，輸入 “cmd” ，開啟 “命令提示字元” 。



2. 輸入 “cd CoreMaker-01” 進入韌體程式所在的資料夾。
3. 輸入 “mbed-tools compile -m AIOT2101 -t GCC_ARM” 開始編譯，若是初次編譯會需要較長的時間。

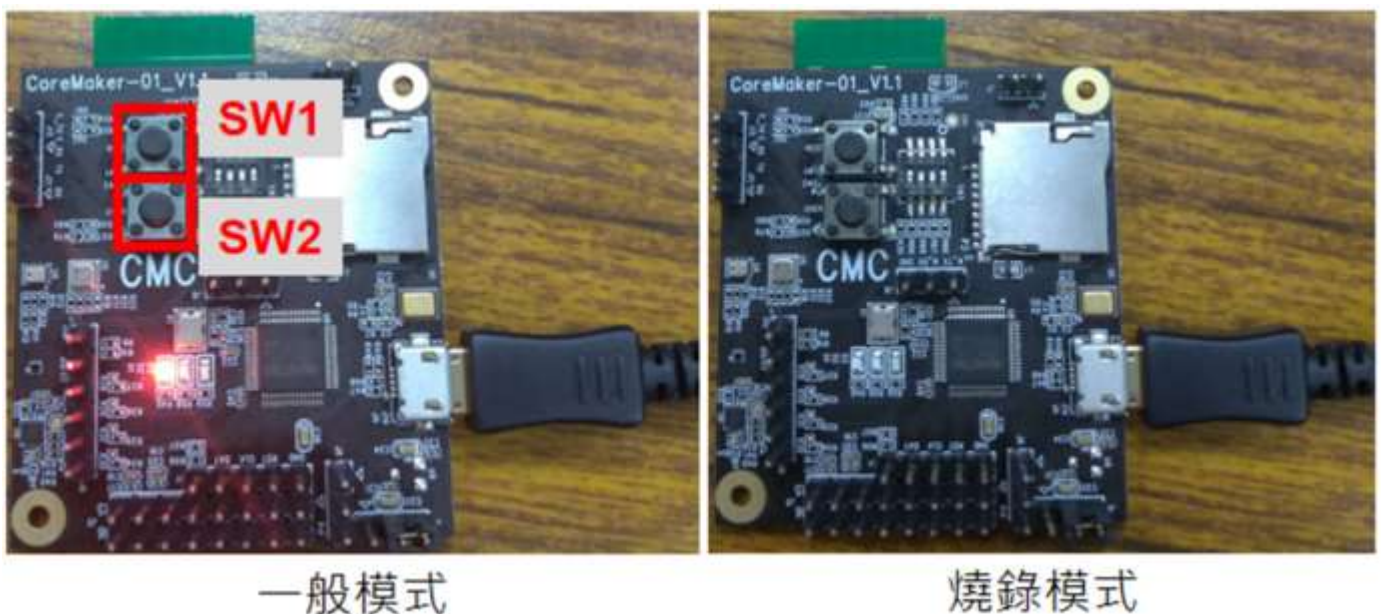


4. 編譯完成會在 CoreMaker-01\cmake_build\AIOT2101\develop\GCC_ARM\目錄下，產生 AIOT_2101.bin 檔案。

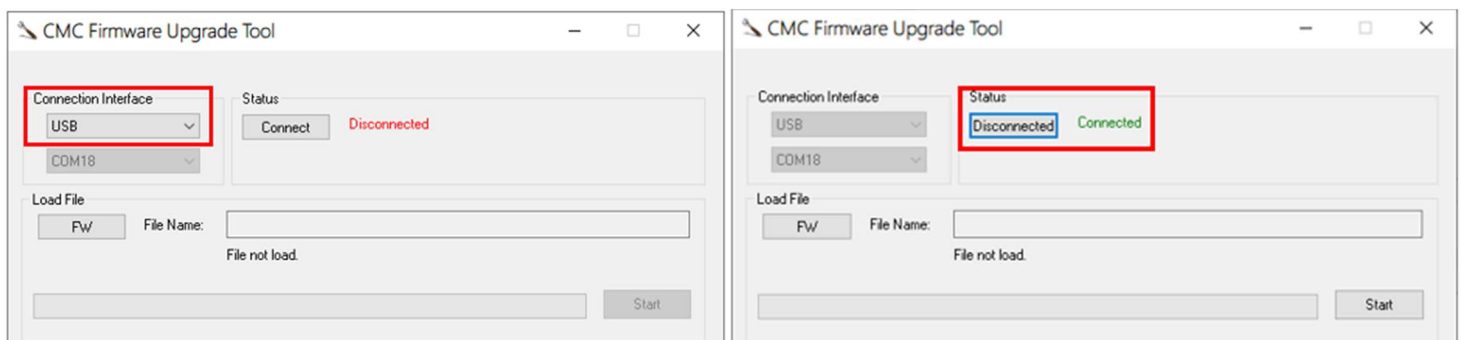


● 燒錄

1. 使用 Micro USB to USB 轉接線，連接 CoreMaker 與 PC。
2. 同時按下 SW1 及 SW2 按鍵。
3. 先放開 SW1，再放開 SW2。
4. 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 2 跟 3。

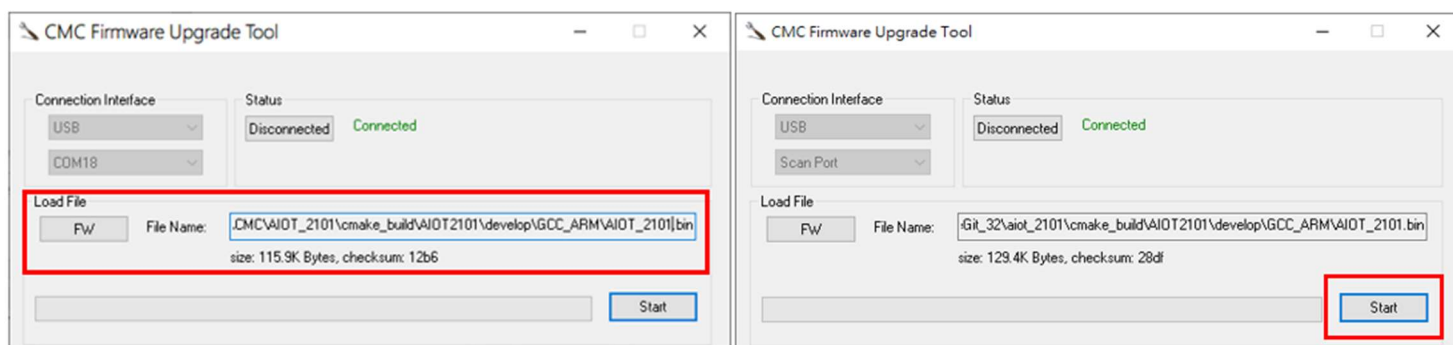


5. 解壓縮 CoreMaker-01/CMC_ISP/CMC_ISP.zip，執行解開後的 CMC_ISP.exe 開啟程式。



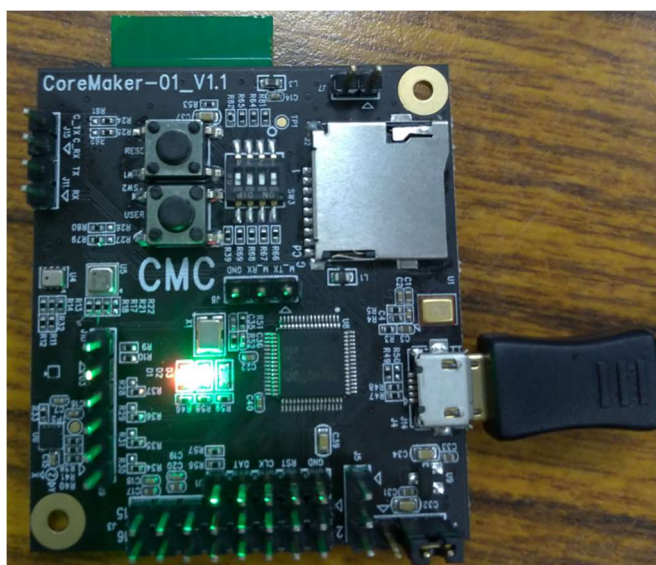
6. 在 Connection Interface 選單選擇 “USB”，點擊 “Connect”，確認連線成功。

7. 選擇編譯完成的 AIOT_2101.bin 檔案，點擊 “Start” 開始燒錄。



● 檢查結果

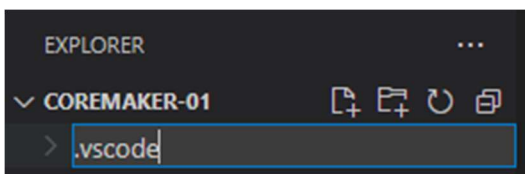
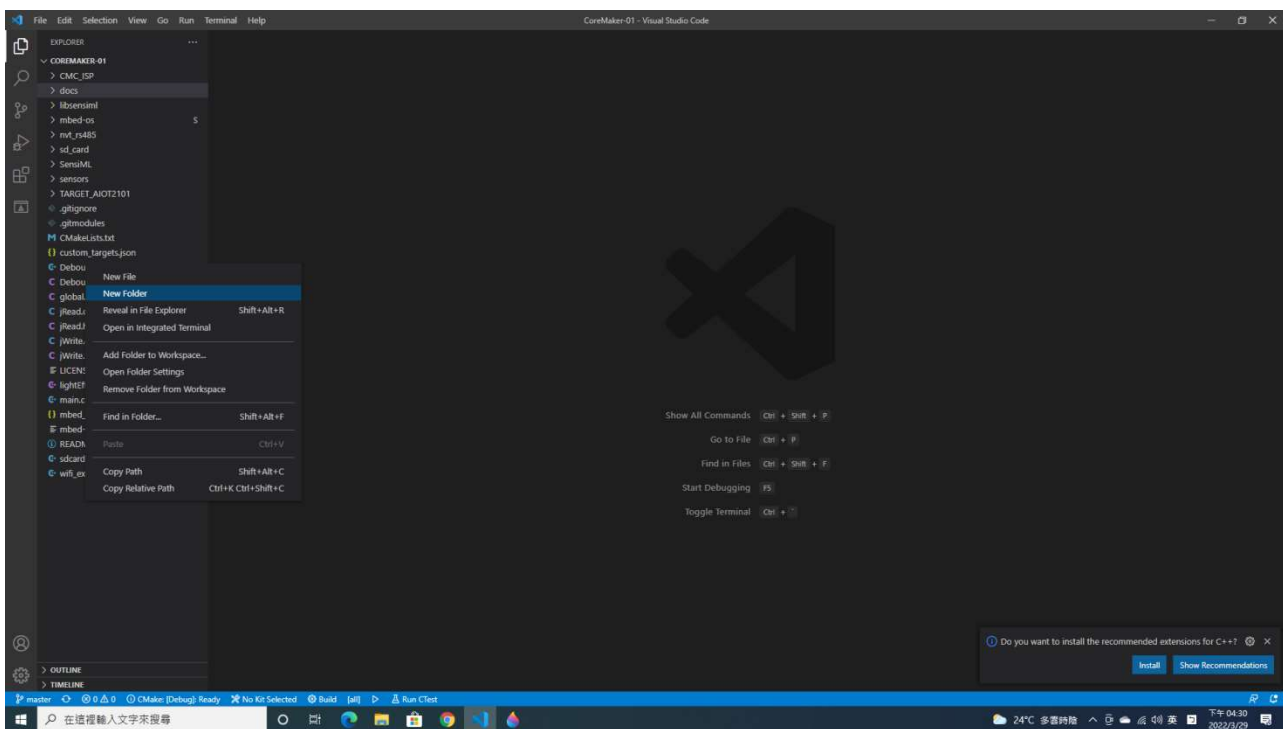
燒錄成功後，CoreMaker 會自動回復為一般模式，此時可以發現 LED 綠燈每 0.1 秒會閃爍一次。



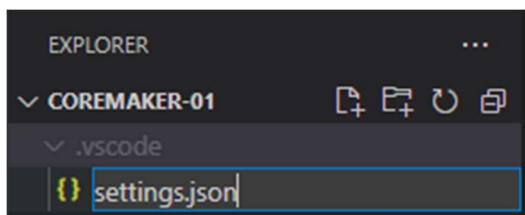
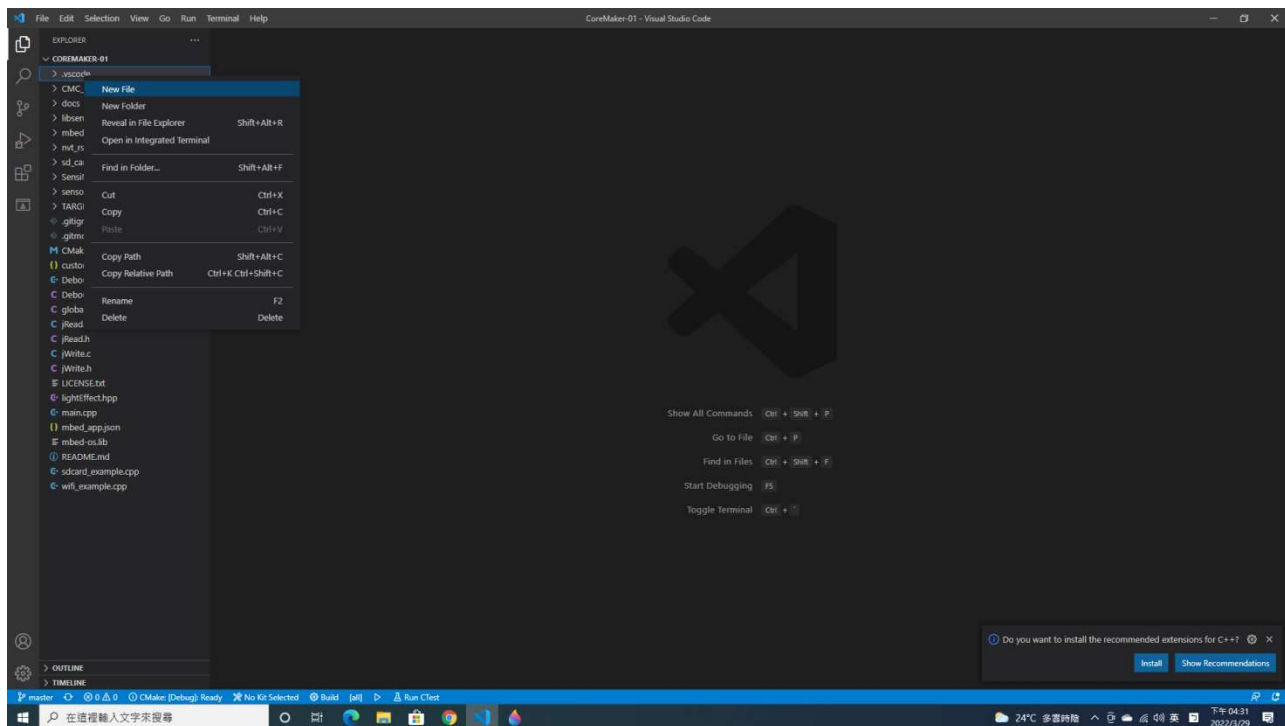
Visual Studio Code 編譯環境設定說明

- 增加設定檔

1. 點選視窗最左方功能列的 “檔案總管 (Explorer) ”，在展開欄位內的空白處點擊滑鼠右鍵，選擇 “新資料夾 (New Folder)” ，建立名為 “.vscode” 的資料夾。



2. 在 .vscode 資料夾上點擊滑鼠右鍵，選擇 “新增檔案 (New File)” ，建立名稱為 “settings.json” 的檔案。

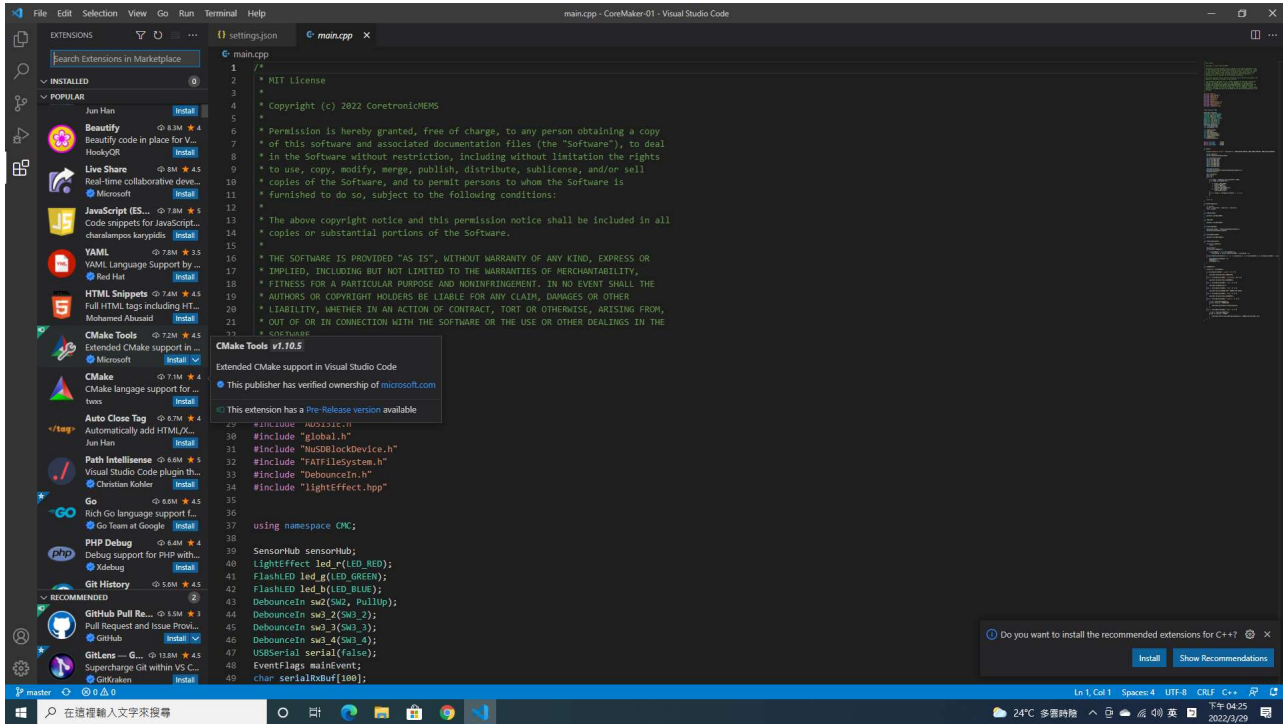


3. 在 settings.json 內，輸入以下內容後儲存：

```
{  
  "cmake.generator": "Ninja",  
  "cmake.buildDirectory": "${workspaceFolder}/cmake_build/AIOT2101/develop/GCC_ARM"  
}
```

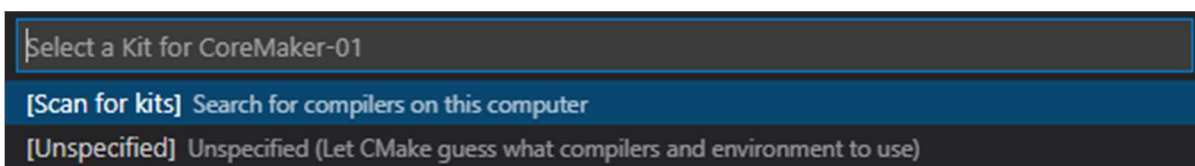
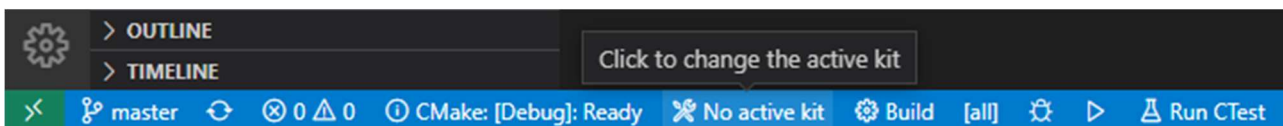
● 安裝延伸套件

點選視窗最左方功能列的 “延伸模組 (Extensions)” ，搜尋 “CMake Tools” ，點擊 CMake Tools 欄位右下角的 “安裝 (Install)” 。

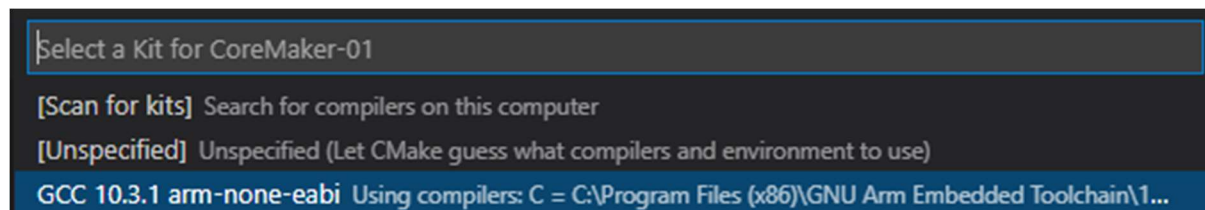


● 進行編譯

1. 點擊視窗最下方藍色區域的交叉工具圖示處 (一開始會顯示 “No active kit”) ，在視窗上方出現的下拉式選單中，選擇 “Scan for kits” 。



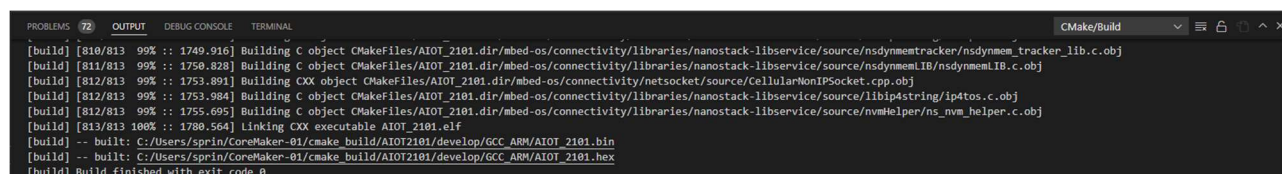
- 再次點擊視窗最下方藍色區域的交叉工具圖示處，此時在視窗上方出現的下拉式選單中，會出現找到的編輯器列表，本範例選擇 “GCC 10.3.1 arm-none-eabi”。



- 點擊視窗最下方藍色區域的齒輪圖示處（文字顯示 “Build”）開始編譯，若是初次編譯會需要較長的時間。



- 編譯完成後，在 Output 欄位會顯示如下圖的資訊。



Document History

[illegible]