



物聯網智造基地

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

# DSI2599 開發套件教學說明

**nuvoTon**

指導單位：經濟部工業局

主辦單位：財團法人資訊工業策進會

執行單位：物聯網智造基地

合作單位：新唐科技股份有限公司

# 目錄

(一) DSI2599 介紹.....	1
1. NuMakerIoTM487 特色 .....	1
2. NuMakerIoTM487 硬體規格 .....	2
3. 腳位說明 .....	2
4. NuMakerIoTM487 搭載 MicroPython 教學 .....	3
5. 燒錄 Python 程式至 NuMakerIoTM487 .....	6
(二) IDEASCHAIN 物聯網雲端數據平台串接說明 .....	9
1. 教材說明 .....	9
2. 教材材料 .....	10
3. 接線圖 .....	10
4. 程式說明 .....	11
5. 教材成果演示 .....	14
6. 電路組裝完成圖片 .....	15

## 圖目錄

圖 1、NuMakerIoTM487 腳位圖 .....	2
圖 2、原出廠狀態的 M487 .....	4
圖 3、已將 ISW1 全部撥至「ON」的 M487 .....	4
圖 4、將 microUSB 線連接於此處 .....	4
圖 5、Python 韌體放置硬碟 .....	5
圖 6、裝置管理員中顯示的 M487 連接埠編號 .....	5
圖 7、終端模擬軟體設定畫面 .....	5
圖 8、於新唐提供的 github 網站資源的 microPython 相關韌體檔案 .....	6
圖 9、M487 中的硬碟內容 .....	6
圖 10、REPL 視窗中顯示已在 M487 安裝 MicroPython 環境 .....	6
圖 11、USB1.1 接上 USB 線 .....	7
圖 12、終端虛擬軟體視窗中提示的「StartUSBdeviceMSCclass」字樣 .....	7
圖 13、於「我的電腦」中出現的虛擬硬碟 .....	8
圖 14、Python 範例程式 .....	8
圖 15、PYBFLASH 中的 main.py 程式 .....	9
圖 16、Python 範例程式，開發板的 LED 熄滅 .....	9
圖 17、Python 範例程式，開發板的 LED 點亮。 .....	9
圖 18、NumakerIoTM487 聯網硬體架構 .....	10
圖 19、M487 與 LED 指示燈接線圖 .....	10
圖 20、程式 1~13 行 .....	11
圖 21、程式 1~13 行 .....	12
圖 22、程式 28~41 行 .....	13
圖 23、程式 43~57 行 .....	14
圖 24、程式 59~61 行 .....	14
圖 25、於 IDEASChain 平台中創建的裝置 .....	14
圖 26、已上傳至平台的數據內容 .....	15
圖 27、M487 與 LED 接線組裝 .....	15
圖 28、未上傳資料時，LED 不亮 .....	16
圖 29、上傳資料時，LED 亮起 .....	16

# 表目錄

表 1、NUMAKERIOTM487 硬體功能簡表.....	2
--------------------------------	---

物聯網智造基地與國產 IC 業者新唐攜手合作，以 NuMakerIoTM487 開發板串接物聯網數據平台(以下稱為 DSI2599)，共同推廣物聯網國產 IC 應用。

教材規劃分為三個部分：DSI2599 介紹、IDEASChain 物聯網雲端數據平台串接說明、軟硬整合應用範例。分別說明如下：

#### (一) DSI2599 介紹

物聯網應用廣泛，從基礎的端點到閘道再到雲端之間需要控制、連網與加密等技術。新唐科技推出適合物聯網使用之 NuMaker-IoT-M487 開發板，以 NuMicroM487 系列微控制器為主控核心，內建 RJ45 乙太網路與 Wi-Fi 模組，讓使用者可以有線或無線的方式連接雲端，亦可透過不同的無線子板提供 NB-IoT、6LoWPAN 等連線能力，以符合各種應用場景需求；配合 NuMicroM487 系列微控制器內建的加解密加速器，提高加密網路連線的效率。

NuMaker-IoT-M487 開發板支援 Arm®Mbed™OS，整合常用元件，內建多軸感測器，提供相容 ArduinoUNO 與 mikroBUS™介面，可與 GPIO、UART、ADC、PWM、SPI 或 I2C 等介面連接；內建新唐積木式概念的感測器模組 NuMakerBrickI2C 介面；提供獨特的模擬影像感測器介面，連接 CMOS 影像感測器，可用於截取影像進行分析；並結合 NAU88L25 音頻編解碼器，提供高音質的錄放音功能；外掛 SPIFlash 及提供 microSD 卡插槽，增加額外的資料存取空間。

NuMaker-IoT-M487 提供多元連網方式與多種介面，並提供影像感測、錄放音功能以及彈性擴充資料存取空間，不論是端點裝置或輕量型的閘道應用，NuMaker-IoT-M487 都是踏入物聯網應用的適切選擇。

#### 1. NuMakerIoTM487 特色

NuMaker-IoT-M487 是使用 NumicroM487JIDAEMCU 的開發板，其中包括 ARMCortex-M4 內核和豐富的外圍設備。它還具有板載運動傳感器和 Wi-Fi 連接以及大量參考資料，可讓用戶快速開發 IoT 設備應用程序。此外，該板還提供了幾個用於擴展的接口和連接器。有 ArduinoUno 兼容接口，mikroBUS™接口，NuMakerBrickI2C 連接器和特定的 I/O 連接器，用於模擬例如圖像傳感器接口。不僅可以直接使用各種 Arduino 或 mikroBUS 子板，而且還方便用戶連接其他所需的組件。

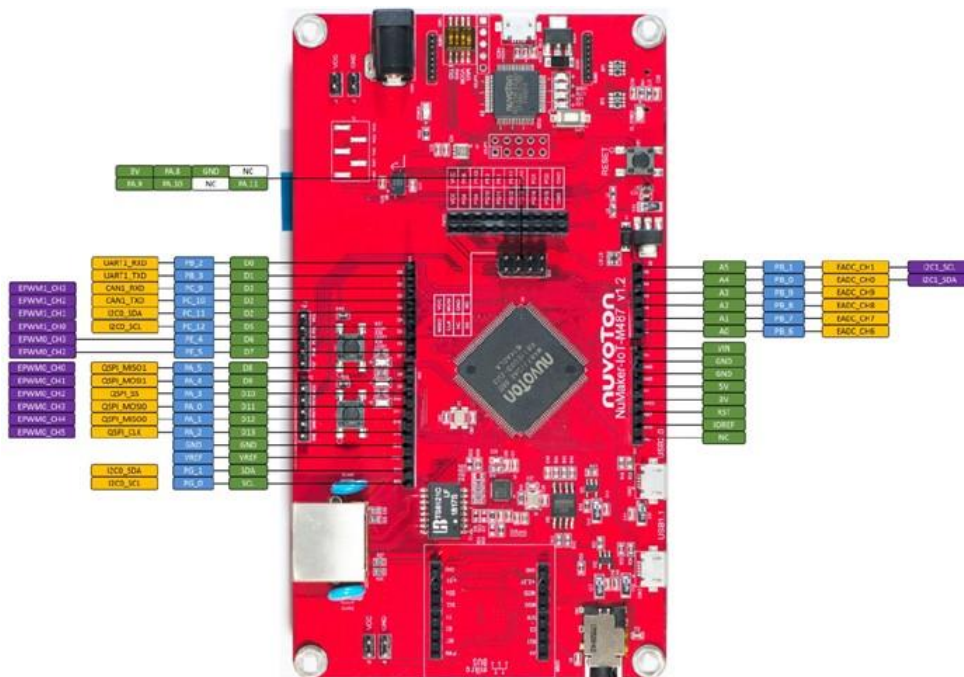
## 2. NuMakerIoTM487 硬體規格

表 1、NuMakerIoTM487 硬體功能簡表

硬體功能	NuMakerIoTM487
MCU	32-bitArmCortex®-M4M487JIDAE
Frequency	192MHz
I/O	16
SRAM	160KB
Flashmemory	512KB
Ethernet	1
SPI	2
UART	1
I2C	2
Audio	1
PWM	12

資料來源：本計畫彙整

## 3. 腳位說明



資料來源：本計畫彙整

圖 1、NuMakerIoTM487 腳位圖

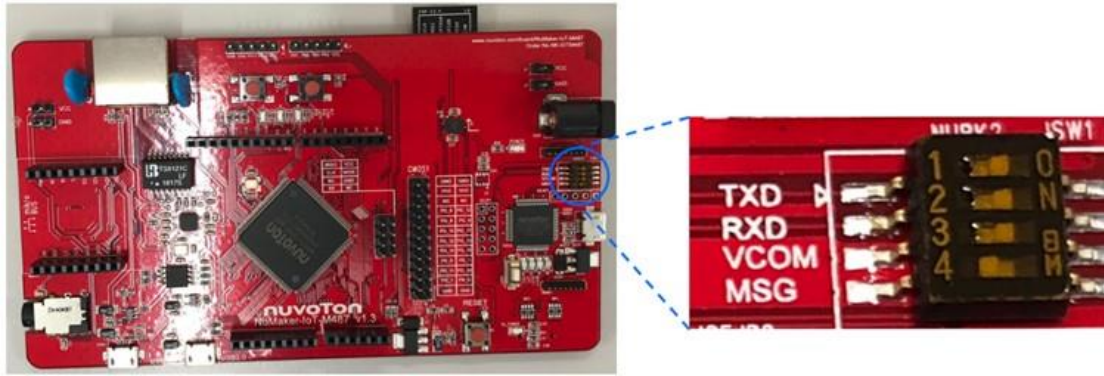
#### 4. NuMakerIoTM487 搭載 MicroPython 教學

(1) 首先將固件刻錄到 MCU 中，啟動後固件將執行用戶的 Python 代碼，Python 代碼的執行支持 REPL 模式和/或從存儲中執行 Python 代碼。固件首先嘗試從存儲執行 Python 代碼，最後進入 REPL 模式，使用者可以在 REPL 模式下測試其 Python 代碼，最後將 Python 代碼放入存儲中。

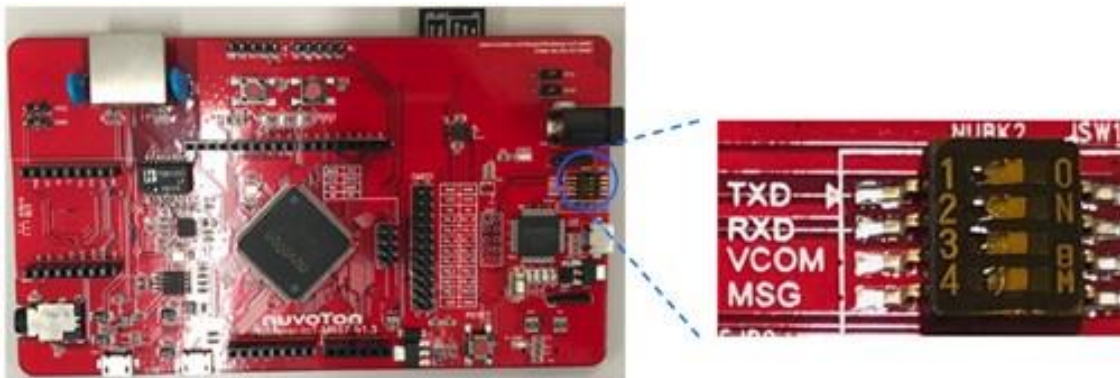
(2) NuMicroPy 將 MCU 的嵌入式閃存分為兩個分區，一個分區是固件分區，另一個分區是數據分區。固件分區用於放置固件二進制代碼。數據分區將是 FAT 文件系統的分區。固件將在執行開始時嘗試安裝數據分區。如果掛載失敗，它將強制將數據分區格式化為 FAT 文件系統，並生成兩個空白的 Python 文件（main.py 和 boot.py）。UART、I2C、CAN、Ethernet 您可以強制固件進入 USB 大容量存儲模式，然後將 Python 代碼寫入這些文件。

MicroPython 是使用 ANSIC 來打造可在微控器上運行的 Python 直譯器、Python 核心庫和底層硬體介面模組。借助 MicroPython，使用者可以使用 Python 語言來實現微控器底層硬體控制，例如：LED 的控制，ADC 的讀取等。另外 MicroPython 還提供了 REPL 模式(Read-Eval-Print-LoopPrompt)，使用者可以直接在命令列上直接下達 Python 程式碼，無需經編譯器，直接在微控器上執行。NuMicroPy 是 MicroPython 在 NuMicro 微控器上的移植方案。移植了直譯器和核心庫外，也移植了底層硬體介面模組，如：UART、I2C、CAN、Ethernet 等。另外也提供 USBMassStorage 的方式來更新使用者的 Python 程式碼，僅需一條 USB 線便可以在電腦上，使用複製貼上方式將使用者的程式碼寫入到微控器。目前 NuMicroPy 支援了 NuMaker-PFM-M487、NuMaker-IOT-M487 和 NuMaker-M263KI 開發板，只需按照 NuMicroPy 網站的說明，就可以在這三張板子上快速開發 Python 的應用。以下為在 NuMakerIoTM487 中搭載 NuMicroPy 的步驟：

- 步驟 1：安裝終端模擬軟體(如 Teraterm、PuTTY 等)，以下範例將使用 PuTTY 作為示範軟體。
- 步驟 2：檢查開發板上，ISW1 開關全數撥到「ON」側。(如下圖)



資料來源：本計畫彙整  
圖 2、原出廠狀態的 M487



資料來源：本計畫彙整  
圖 3、已將 ISW1 全部撥至「ON」的 M487

- 步驟 3：將 USB 透過開發板右方的 USB 孔與電腦連接，連接後在「我的電腦」中會出現開發板的虛擬硬碟(如下圖)。



資料來源：本計畫彙整  
圖 4、將 microUSB 線連接於此處

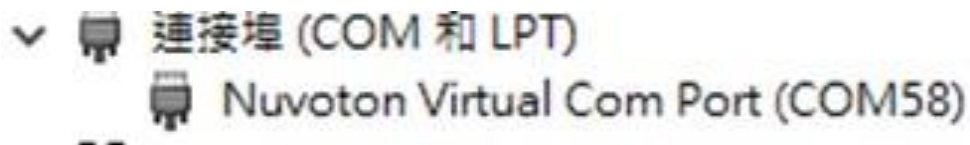




資料來源：本計畫彙整

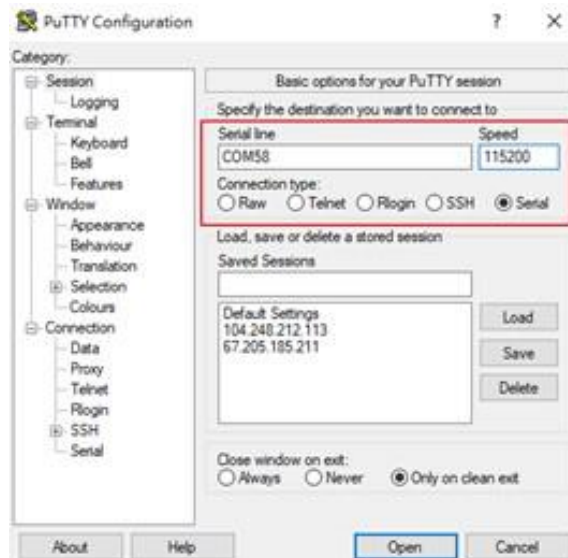
圖 5、Python 韌體放置硬碟

- 步驟 4：先到裝置管理員查看開發板的連接埠號碼(如圖 6)，再開啟 PuTTY 進行設定，並開啟其序列串端口(如圖 7)。開發板的 baudrate 必須設定為 115200，否則序列串端口會無法開啟。



資料來源：本計畫彙整

圖 6、裝置管理員中顯示的 M487 連接埠編號



資料來源：本計畫彙整

圖 7、終端模擬軟體設定畫面

- 步驟 5：至(t.ly/2YIH)下載新唐提供的檔案，並將 M487 的 Python 韌體設定檔案複製到 NumicroMCU 的虛擬硬碟中，完成後終端模擬軟體的視窗會出現 microPython 字樣為首的提示句，表示韌體設定完成。



資料來源：本計畫彙整

圖 8、於新唐提供的 github 網站資源的 microPython 相關韌體檔案



資料來源：本計畫彙整

圖 9、M487 中的硬碟內容



資料來源：本計畫彙整

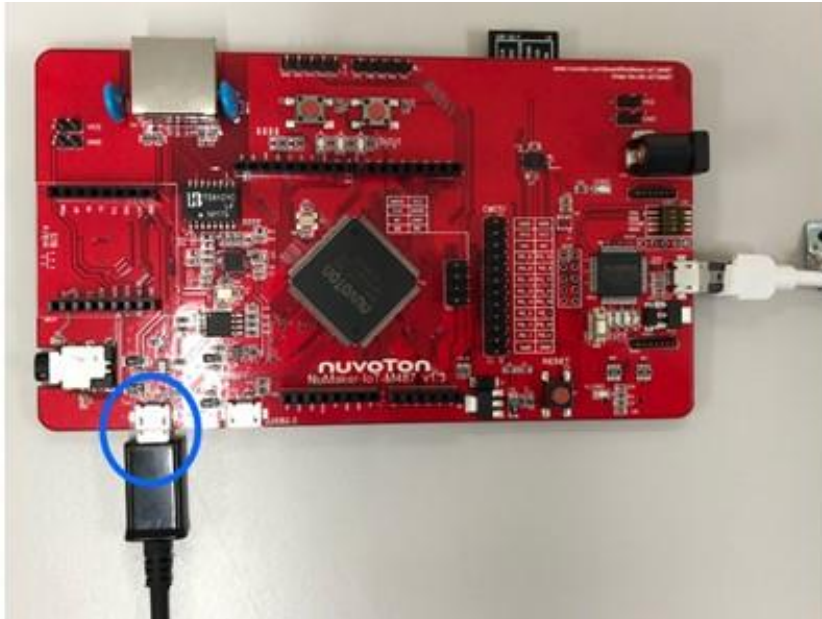
圖 10、REPL 視窗中顯示已在 M487 安裝 MicroPython 環境

- 完成至此步驟，便能透過終端虛擬軟體，使用 REPL 進行簡單的 Python 程式撰寫與執行。

## 5. 燒錄 Python 程式至 NuMakerIoTM487

完成上述步驟後，便可以將 Python 的程式檔(.py)燒錄到 M487 開發板中，讓開發板在日後通電，便能自己執行內部的程式檔案。

- 步驟 1：承上節步驟 5，並保持終端虛擬軟體開啟，用 microUSB 將 M487 的 USB1.1 與電腦連接(如圖 11)。



資料來源：本計畫彙整

圖 11、USB1.1 接上 USB 線

- 步驟 2：先長按開發板上的 SW2，再觸發 RESET 鍵，若終端虛擬軟體出現「StartUSBdeviceMSCclass」字樣(如圖 12)，或是「我的電腦」中出現「PYBFLASH」的虛擬硬碟，則 SW2 可放開(如圖 13)。

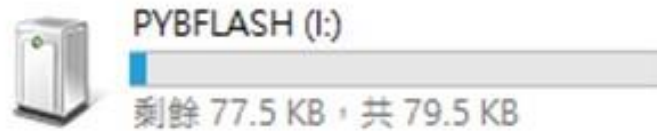
A screenshot of a terminal window titled 'COM58 - PuTTY'. The terminal displays the output of a MicroPython program. The text is as follows:

```
MicroPython 56de7d8-dirty on 2019-11-06; NuMaker-IOT-M487 with Nuvoton-M48x
>>> +-----+
|           Micropython Code for Nuvoton M48x           |
+-----+
MicroPython 56de7d8-dirty on 2019-11-06; NuMaker-IOT-M487 with Nuvoton-M48x
>>> +-----+
|           Micropython Code for Nuvoton M48x           |
+-----+
Start USB device MSC class
```

The text 'Start USB device MSC class' is highlighted with a red rectangular box.

資料來源：本計畫彙整

圖 12、終端虛擬軟體視窗中提示的「StartUSBdeviceMSCclass」字樣



資料來源：本計畫彙整

圖 13、於「我的電腦」中出現的虛擬硬碟

- 步驟 3：開啟文字編輯軟體(如記事本、Notepad++等)，並開啟 Python 程式碼。任何程式內容，必須將其檔名儲存為 main.py。使用的範例程式為開發板上的 LED 閃爍控制示範。

```
main.py [x]
1  import pyb
2  from pyb import LED
3
4  led = LED('led0')
5  while(1):
6      led.off()
7      pyb.delay(1000)
8      led.on()
9      pyb.delay(1000)
```

資料來源：本計畫彙整

圖 14、Python 範例程式

- 步驟 4：將此範例程式，取代 PYBFLASH 中原本的 main.py(如圖 15)。



資料來源：本計畫彙整

圖 15、PYBFLASH 中的 main.py 程式

- 步驟 5：按下 RESET 鍵，此時開發板會執行程式內容。



資料來源：本計畫彙整

圖 16、Python 範例程式，開發板的 LED 熄滅



資料來源：本計畫彙整

圖 17、Python 範例程式，開發板的 LED 點亮。

## (二) IDEASChain 物聯網雲端數據平台串接說明

### 1. 教材說明

NuMakerIoTM487 本身便配有一顆 Wi-Fi 的連線模組，就算不使用其他的無線連線模組，只要放置環境具備可連線的 Wi-Fi 熱點，開發板便能與雲端平台建立連線(如下圖)。



資料來源：本計畫彙整

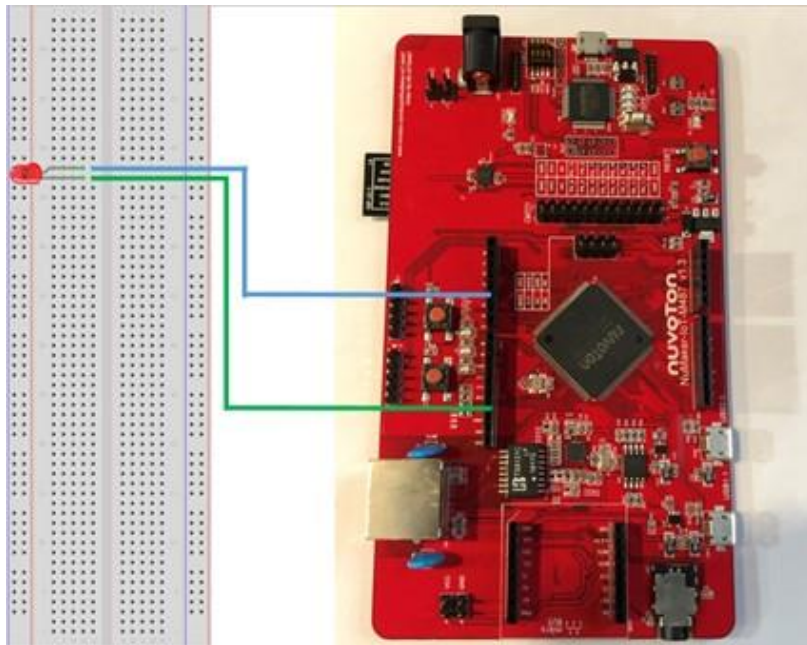
圖 18、NumakerIoTM487 聯網硬體架構

本段內容為說明 NuMakerIoTM487 如何透過 Wi-Fi 與 IDEASChain 平台連接，並將數據資料上傳到平台。上傳的數據為教學測試用，本身並無任何意義，使用者可根據自身產品的應用內容賦予數據意義。

## 2. 教材材料

- NuMakerIoTM487 開發板 1 個
- LED1 個
- 杜邦線 2 條

## 3. 接線圖



資料來源：本計畫彙整

圖 19、M487 與 LED 指示燈接線圖

#### 4. 程式說明

請至(t.ly/YH0f)下載範例程式，並解壓縮取得程式檔案。以下為範例程式之說明內容：

第 1-4 行：導入程式需要用到的 library。

第 6 行：設定控制 LED 腳位為 OUTPUT，腳位為開發板上的 D4。

第 7 行：讓腳位初始輸出電壓為 0，也就是 LOW。

第 9 行：宣告連接 Wi-Fi 的 SSID 內容。

第 10 行：宣告連接 Wi-Fi 的密碼內容。

第 11 行：連接的雲端平台伺服器區域名稱。

第 12 行：宣告後續 POST 格式要用到的平台網址名稱。

第 13 行：宣告要傳送到雲端平台的存取權限碼。

```
1 import pyb
2 from pyb import Pin
3 import network
4 import usocket as socket
5
6 get_out = Pin(Pin.board.D4, Pin.OUT)
7 get_out.value(0)
8
9 SSID = "ST67_fat_2.4G"
10 PASS = "0988719859"
11 HOST = "ideaschain.com.tw"
12 API_URL = "iiot.ideaschain.com.tw"
13 DEVICE_KEY = "N0aQ40M9R8PbGN6ucZLz"
```

資料來源：本計畫彙整

圖 20、程式 1~13 行

第 14-23 行：自定義的 def 函數內容，Wi-Fi 連線上若出現錯誤，可保持繼續嘗試連線而不會因出現錯誤而當機。

第 15 行：當程式正常運作時，會執行第 13~15 行的內容。

第 16 行：在終端模擬程式顯示「connectingtoWi-Fi」的訊息。

第 17 行：設定聯網的對象為 WLAN，即為無線網路。

第 18 行：開始連線到該 Wi-Fi 熱點。

第 20 行：若程式運作途中出現問題，則執行 21~23 行內容。

第 21 行：在終端模擬程式顯示「Wi-Fimoduleinitialerror, reconnecting.....」的訊息，表示目前運作途中出現問題。

第 22 行：延遲 1000 毫秒=1 秒。

第 23 行：回去重複執行此函數內容，直到連到該 Wi-Fi 熱點為止。

第 25 行：開始執行 Wi-Fi()函數內容。

第 26 行：成功連上 Wi-Fi 後，在終端模擬程式顯示「Wi-Ficonnect」的訊息，表示已經連上設定的 Wi-Fi 熱點。

```
14 def wifi():
15     try:
16         print("connecting to wifi")
17         wlan = network.WLAN()
18         wlan.connect(SSID, PASS)
19
20     except:
21         print("Wifi module initial error, reconnecting.....")
22         pyb.delay(1000)
23         wifi()
24
25 wifi()
26 print("Wi-Fi connect")
```

資料來源：本計畫彙整

圖 21、程式 1~13 行

第 28 行：宣告一字符串內容，其內容為要上傳到雲端平台的訊息內容。其中，key1 為物件名稱，value1 為該物件的內容。例：上傳目前溫度為 28 度，key1 可改為「temperature」，value 改為「28」。

第 29 行：計算此字符串長度。

第 31 行：宣告一 socket 類別的變數，其內容為要連接的伺服器區域名稱與連接端口。

第 32 行：顯示該伺服器的 IP 訊息。

第 34-39 行：宣告一變數，組成後面上傳訊息需要用的字符串。

第 41 行：顯示字符串內容，讓開發者檢查字符串組合是否有錯誤。



```

28 param_data = "{\n \"key1\": \"value1\"\n}"
29 param_length = str(len(param_data))
30
31 addr = socket.getaddrinfo(HOST, 80)[0][-1]
32 print(addr)
33
34 request_str = ""\
35 POST /api/v1/"+"+DEVICE_KEY+""/telemetry HTTP/1.1\r
36 Content-Type: application/json\r
37 Content-Length: "+"param_length+"\r
38 Host: "+"API_URL+"\r
39 \r\n"+"param_data
40
41 print(request_str);

```

資料來源：本計畫彙整

圖 22、程式 28~41 行

第 43-57 行：自定義的 def 函數內容，伺服器連線上若出現錯誤，可保持繼續嘗試連線而不會因出現錯誤而當機。

第 44 行：當程式正常運作時，會執行第 45~54 行的內容。

第 45 行：建立 socket 通訊的參數。

第 46 行：顯示「connectingto server」，讓開發者知道已經開始與伺服器進行連線。

第 47 行：socket 開始進行 server 連線。

第 48 行：延遲 1000 毫秒=1 秒。

第 49 行：顯示「serverisconnected」，讓開發者知道已經進行連線上。

第 50 行：上傳資料前，控制 LED 亮起。

第 51 行：上傳資料。

第 52 行：顯示「senddone」，讓開發者知道已結束資料上傳。

第 53 行：將 LED 燈熄滅。

第 54 行：關掉與伺服器的連線，避免佔據連接端口。

第 55 行：若發生無法連線到伺服器的狀況，造成意外當機，則執行 56~57 行的內容。

第 56 行：顯示「senddone」，讓開發者知道上傳過程中出現問題，並結束此函數動作，待迴圈回來時再從頭執行。

第 57 行：延遲 10000 毫秒=10 秒。

```

43 =def go():
44 =     try:
45         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
46         print("connecting to server")
47         s.connect(addr)
48         pyb.delay(1000)
49         print("server is connected")
50         get_out.value(1)
51         s.send(bytes(request_str, 'utf8'))
52         print("send done")
53         get_out.value(0)
54         s.close()
55 =     except:
56         print("except")
57         pyb.delay(1000)

```

資料來源：本計畫彙整

圖 23、程式 43~57 行

第 59 行：反覆執行 60~61 行的內容。

第 60 行：執行 go() 函數。

第 61 行：延遲 10000 毫秒=10 秒。

```

59 =while True:
60     go()
61     pyb.delay(1000)

```

資料來源：本計畫彙整

圖 24、程式 59~61 行

## 5. 教材成果演示



資料來源：本計畫彙整

圖 25、於 IDEASChain 平台中創建的裝置

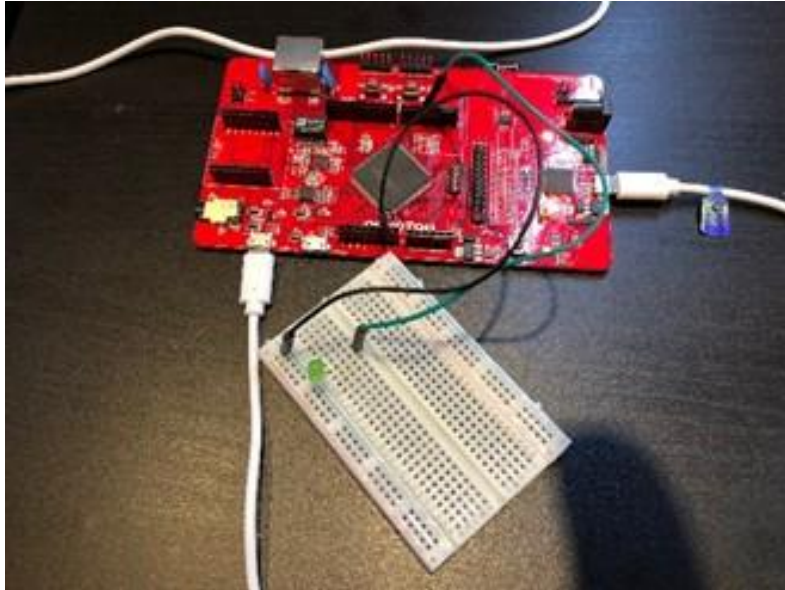


資料來源：本計畫彙整  
圖 26、已上傳至平台的數據內容

#### 6. 電路組裝完成圖片

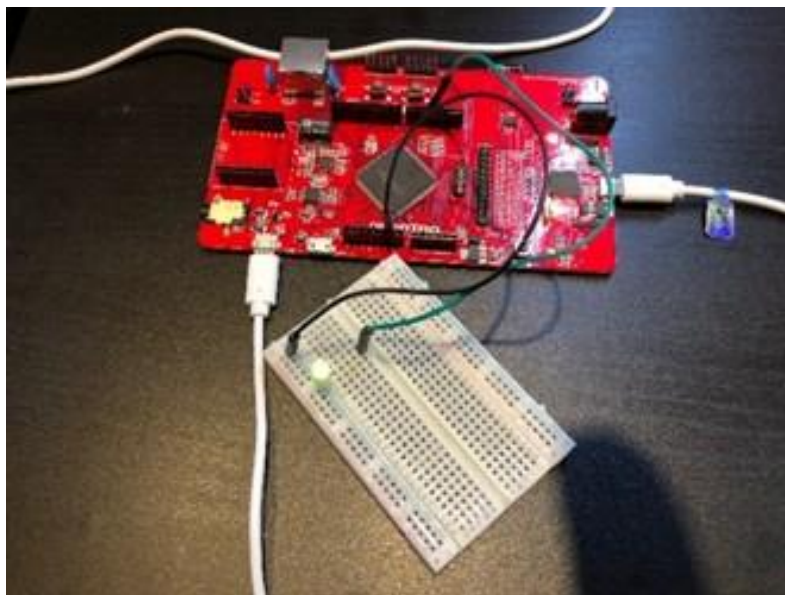


資料來源：本計畫彙整  
圖 27、M487 與 LED 接線組裝



資料來源：本計畫彙整

圖 28、未上傳資料時，LED 不亮



資料來源：本計畫彙整

圖 29、上傳資料時，LED 亮起