**物聯網雲端管理平台：Azure IoT**

**國立中正大學資訊工程系 黃仁竑教授**

**實驗目的**

本實驗主要是教導學生如何使用LinkIt 7688 Duo與Azure IoT平台連接，傳送溫溼度感測值。

**實驗所需軟硬體**

軟體：

* Windows 10
* Azure
* Arduino IDE
* Python

硬體：

* LinkIt 7688 Duo
* Micro-USB
* DHT 22
* 擴充板
* 杜邦線

目錄

[一、 Introduction 1](#_Toc117523227)

[1. 各元件功能 1](#_Toc117523228)

[2. 什麼是 Azure IoT 1](#_Toc117523229)

[3. Azure IoT 基本架構 2](#_Toc117523230)

[4. LinkIt Smart 7688 Duo補充 3](#_Toc117523231)

[5. 如何使用裝置連線到Azure IoT 中樞 4](#_Toc117523232)

[二、 Applying for an Azure account 4](#_Toc117523233)

[1. Join Azure Educate 4](#_Toc117523234)

[2. 填寫個人資訊 4](#_Toc117523235)

[3. Check Your E-mail 6](#_Toc117523236)

[4. E-mail Verification 7](#_Toc117523237)

[5. Successful Verification 7](#_Toc117523238)

[三、 Connect to Azure 7](#_Toc117523239)

[1. 登入Azure平台 7](#_Toc117523240)

[2. 建立IoT Hub 7](#_Toc117523241)

[3. 依序輸入資料 9](#_Toc117523242)

[4. 成功建立，完成畫面 10](#_Toc117523243)

[5. 建立IoT device 11](#_Toc117523244)

[6. 點選新增 11](#_Toc117523245)

[7. 輸入裝置識別碼(名稱)，確認認證類型為“對稱金鑰”，就可儲存。 11](#_Toc117523246)

[8. 成功建立，完成畫面 12](#_Toc117523247)

[9. 下載Visual Studio Code 13](#_Toc117523248)

[10. 開啟Visual Studio Code 13](#_Toc117523249)

[11. 連接Microsoft 帳戶使用者的Azure 13](#_Toc117523250)

[12. 單擊左下角的“ Azure IoT中心” 14](#_Toc117523251)

[13. 在菜單中單擊“Set IoT Hub Connection String”。 14](#_Toc117523252)

[14. 輸入您的IoT Hub Connection String 14](#_Toc117523253)

[15. 完成Azure IoT HUB連接，成功畫面 15](#_Toc117523254)

[16. 獲取IoT device的SAS Token 16](#_Toc117523255)

[17. 輸入SAS Token使用時數 17](#_Toc117523256)

[18. 記住SAS Token值(請留存，後面還會用到) 18](#_Toc117523257)

[19. LinkIt 7688 Duo接DHT22 18](#_Toc117523258)

[20. Arduino安裝程式庫 18](#_Toc117523259)

[21. 下載Library 19](#_Toc117523260)

[22. 匯入Library 19](#_Toc117523261)

[23. Arduino IDE端程式撰寫 19](#_Toc117523262)

[24. 安裝Mobaxterm 20](#_Toc117523263)

[25. SSH連線 20](#_Toc117523264)

[26. 成功連線，登入7688 21](#_Toc117523265)

[27. 成功登入 21](#_Toc117523266)

[28. 安裝套件 22](#_Toc117523267)

[29. 下載程式碼和憑證 22](#_Toc117523268)

[30. 將mqtt.crt和mqtt.py放入 22](#_Toc117523269)

[31. 使用vim mqtt.py查看程式 23](#_Toc117523270)

[32. 開啟Visual Studio Code監控7688所上傳的資料 24](#_Toc117523271)

[33. 執行mqtt.py 25](#_Toc117523272)

[34. 由Azure IoT HUB查看傳送數據量 26](#_Toc117523273)

[四、 參考資料 26](#_Toc117523274)

1. Introduction
2. 各元件功能

[Azure IoT Edge](https://azure.microsoft.com/services/iot-edge/)

**將雲端智慧部署到本機 IoT 邊緣裝置**

部署在雲端中建置及訓練的模型，並於內部執行。例如，如果您將預測性模型部署到工廠相機以測試品質控制，當偵測出問題時，IoT Edge 便會觸發警示，並在本機處理資料，或將其傳送至雲端進行進一步分析。

[Azure IoT 中樞](https://azure.microsoft.com/services/iot-hub/) (Azure IoT Hub)

**建置可雙向通訊的 IoT 應用程式**

使用裝置到雲端的遙測資料來了解裝置的狀態，並定義傳給其他 Azure 服務的訊息路由，而不需要撰寫任何程式碼。在雲端到裝置的訊息中，可靠地將命令及通知傳送至連接的裝置，並透過通知回條來追蹤訊息傳遞。視需要自動重新傳送裝置訊息，以便配合間歇性連線。

[Azure 串流分析](https://azure.microsoft.com/services/stream-analytics) (Azure Stream Analytics)

使用熟悉的 SQL 語法，幾分鐘內即可建置好已能在生產環境中執行的端對端分析管線，並可使用 JavaScript 和 C# 自訂程式碼進行延伸。具備彈性容量的快速延展性，可用於建置強固的串流資料管線，並以低於一秒的延遲時間分析數百萬個事件。混合式串流處理架構，並可在雲端與邊緣執行相同的查詢。具有內建復原功能的企業級可靠性，以及用於進階方案的內建機器學習功能。

[Azure 資料總管](https://azure.microsoft.com/services/data-explorer) (Azure Data Explorer)

Azure Data Explorer 是快速、完全受控的資料分析服務，可即時分析來自應用程式、網站、IoT 裝置等的大量資料流。詢問問題並即時反覆探索資料，以改善產品、增強客戶體驗、監視裝置並加速作業。快速識別您資料中的模式、異常狀況和趨勢。探索新的問題並在幾分鐘內獲得解答。多虧有最佳化的成本結構，您可以視需要執行多次查詢。

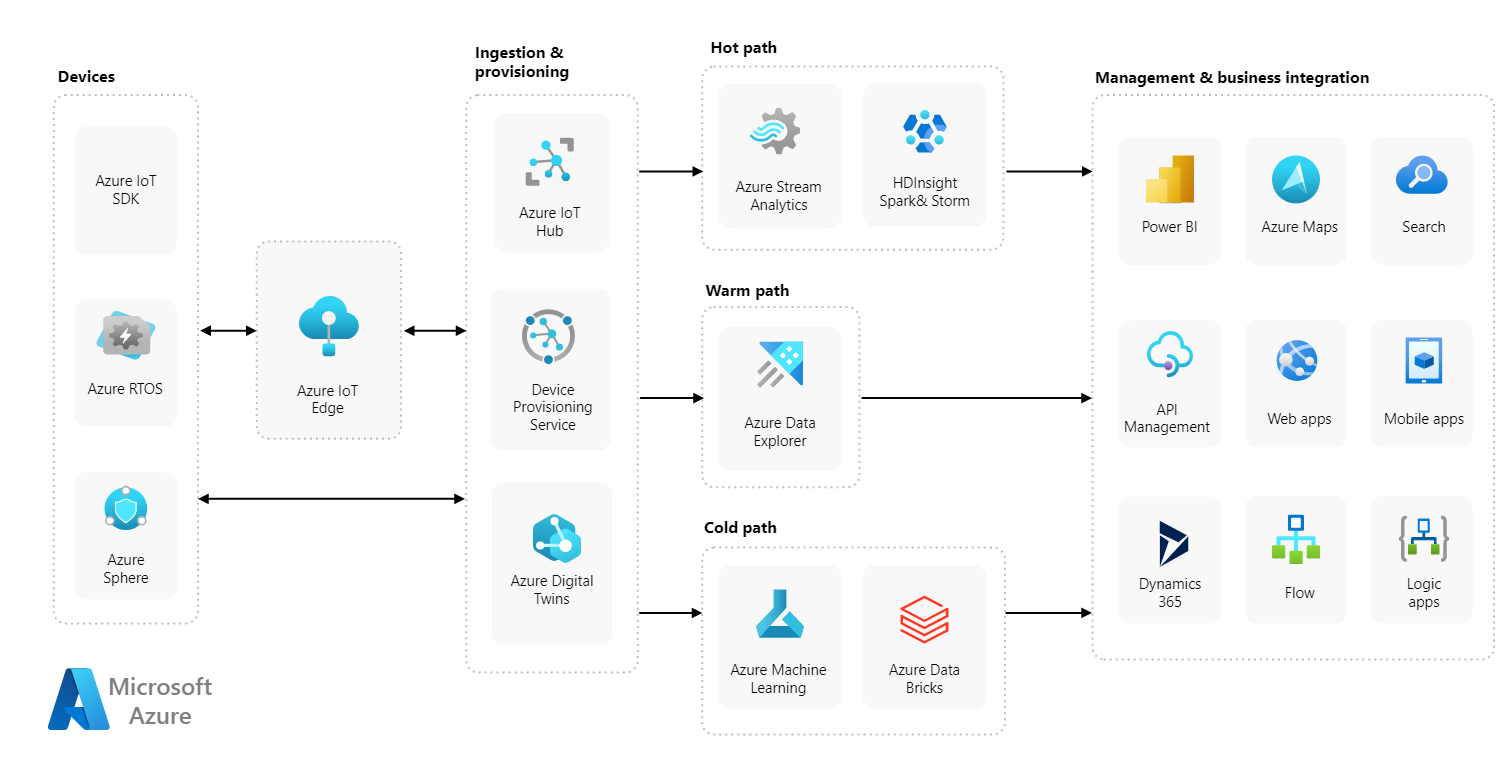
[Azure 監視器](https://azure.microsoft.com/services/monitor) (Azure Monitor)

從雲端和混合式環境中收集、分析遙測資料並對其採取動作。Azure 監視器可協助您最大限度地提高資源的效能和可用性，並主動辨識問題，從而支援您的大規模營運。

1. 什麼是 Azure IoT

Azure IoT 是跨邊緣和雲端的受控平台服務集合，可連線、監視及控制數十億個 IoT 資產。其中也包括裝置和設備的安全性與作業系統，以及可協助企業建置、部署並管理 IoT 應用程式的資料和分析。討論 Azure IoT 時，我們會考慮這些服務如何在三個元件之間搭配運作：

1. **Things**
2. **Insights**
3. **Actions**
4. Azure IoT 基本架構

****

**Devices**

Azure IoT supports a large range of devices, from microcontrollers running [Azure RTOS](https://learn.microsoft.com/en-us/azure/rtos) and [Azure Sphere](https://learn.microsoft.com/en-us/azure-sphere/product-overview/what-is-azure-sphere) to developer boards like Raspberry Pi.

Azure IoT also supports smart server gateways capable of running custom code. Devices might perform some local processing through a service such as [Azure IoT Edge](https://learn.microsoft.com/en-us/azure/iot-edge), or just connect directly to Azure so that they can send data to and receive data from the IoT solution.

When devices are connected to the cloud, there are several services that assist with ingesting data.

* [Azure IoT Hub](https://learn.microsoft.com/en-us/azure/iot-hub) is a cloud gateway service that can securely connect and manage devices.
* [Azure IoT Hub Device Provisioning Service (DPS)](https://learn.microsoft.com/en-us/azure/iot-dps/about-iot-dps) enables zero-touch, just-in-time provisioning that helps to register a large number of devices in a secure and scalable manner.
* [Azure Digital Twins](https://learn.microsoft.com/en-us/azure/digital-twins) enables virtual models of real world systems.

**Insights**

Once devices are connected to the cloud, you can process and explore their data to gain customized insights about their environment.

At a high level, there are three ways to process data: **hot path**, **warm path**, and **cold path**.

The paths differ in their requirements for **latency** and **data access**.

* **Hot path**
  + analyzes data in near-real-time as it arrives.
  + telemetry must be processed with very low latency.
  + typically uses a stream processing engine.
* Consider using services such as [Azure Stream Analytics](https://learn.microsoft.com/en-us/azure/stream-analytics) or [Azure HDInsight](https://learn.microsoft.com/en-us/azure/hdinsight).
* **Warm path**
  + analyzes data that can accommodate longer delays for more detailed processing.
* Consider [Azure Data Explorer](https://learn.microsoft.com/en-us/azure/data-explorer) for storing and analyzing large volumes of data.
* **Cold path**
  + performs batch processing at longer intervals, like hourly or daily.
  + typically operates over large volumes of data, which can be stored in [Azure Data Lake Storage](https://learn.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction).
  + Results don't need to be as timely as in the hot or warm paths.
* Consider using [Azure Machine Learning](https://learn.microsoft.com/en-us/azure/machine-learning) or [Azure Databricks](https://learn.microsoft.com/en-us/azure/databricks) to analyze cold data.

**Actions**

You can use the insights you gather about your data to manage and control your environment.

Business integration actions might include:

* + Storing informational messages.
  + Raising alarms.
  + Sending email or SMS messages.
  + Integrating with business applications such as customer relationship management (CRM) and enterprise resource planning (ERP).

1. LinkIt Smart 7688 Duo補充

LinkIt Smart 7688 Duo有2個核心：

第一個是以Arduino的MCU (Micro Processor Unit)為主控，提供與Yun類似的橋接函式庫(Bridge Library)。

第二個則是Wi-Fi、嵌入式Linux的MPU (Micro Processing Unit)。

7688 Duo 上的 MT7688 可以提供 wifi 給沒有 wifi 的 Arduino chip (32U4) 使用。但是 Duo 預設的運算上以 MPU (MT7688) 為主 MCU (Arduino) 為輔。若要轉變成 MCU (Arduino) 為主 MPU (MT7688) 為輔的話，需要把 Yunbridge 做開啟的動作。

1. 如何使用裝置連線到Azure IoT 中樞

首先我們需要先確保裝置具備Python或C語言，並針對特定裝置功能來認證 IoT 裝置，有列於 Azure 認證裝置目錄中 (<https://devicecatalog.azure.com/>)。

第二步，我們使用Arduino IDE將DHT的溫溼度借由Bridge Library從MPU傳到MCU。

接下來我們使用MQTT協定來傳輸其裝置sensor所測量到的值，同時使用Visual Studio Code直接讀取7688傳輸到IoT 中樞的資料。

所以我們會需要IoT Hub Connection String先讓IoT 中樞與Visual Studio Code相連，再Python程式中會使用IoT Device的SAS Token和DigiCert Baltimore 根憑證，將數據傳輸到IoT Device中。如果想使用Azure其他服務可以參考上方各元件功能進行使用。

1. Applying for an Azure account
2. Join Azure Educate

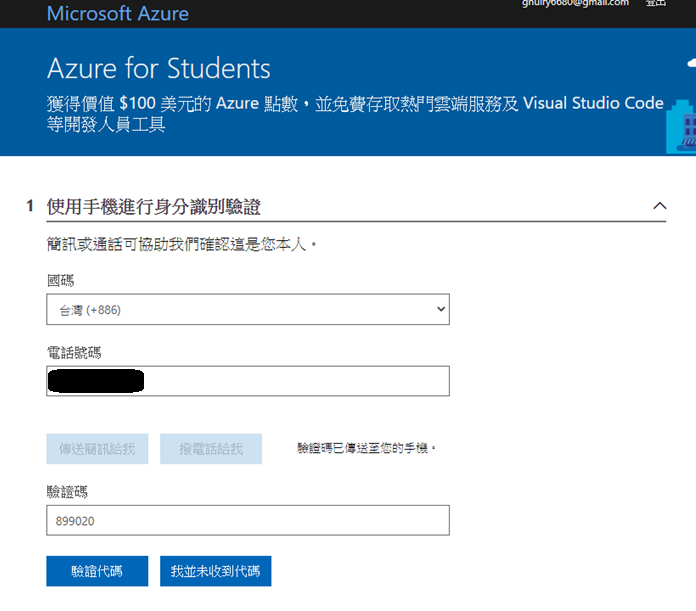
到Azure Educate網址註冊<https://azure.microsoft.com/zh-tw/free/students/>。



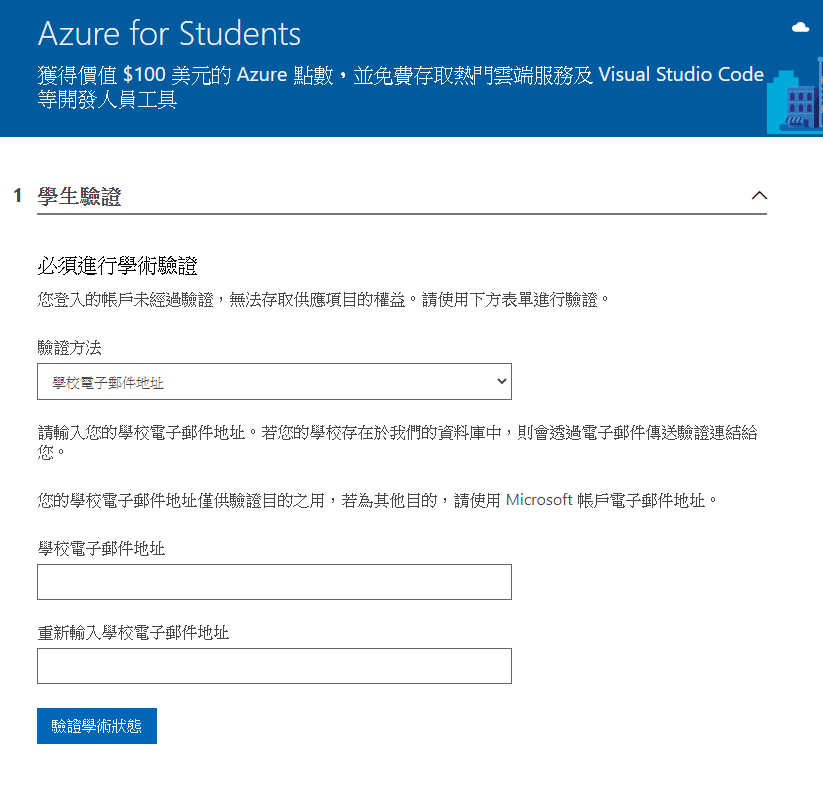
1. 填寫個人資訊

範例：

國碼：台灣(+886)

電話：900-000-000

學校信箱：g0\*\*10\*\*\*@ccu.edu.tw



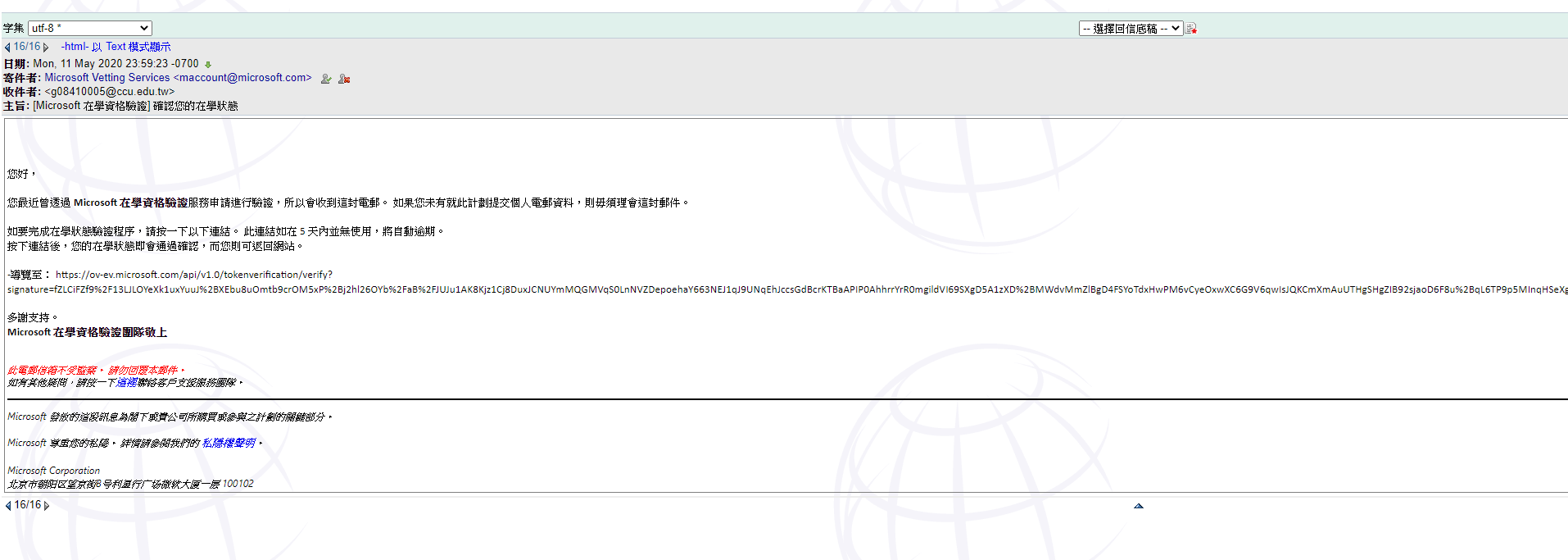
1. Check Your E-mail

(<https://webmail.ccu.edu.tw/cgi-bin/owmmdir2/openwebmail.pl>)，請到信箱收確認信。



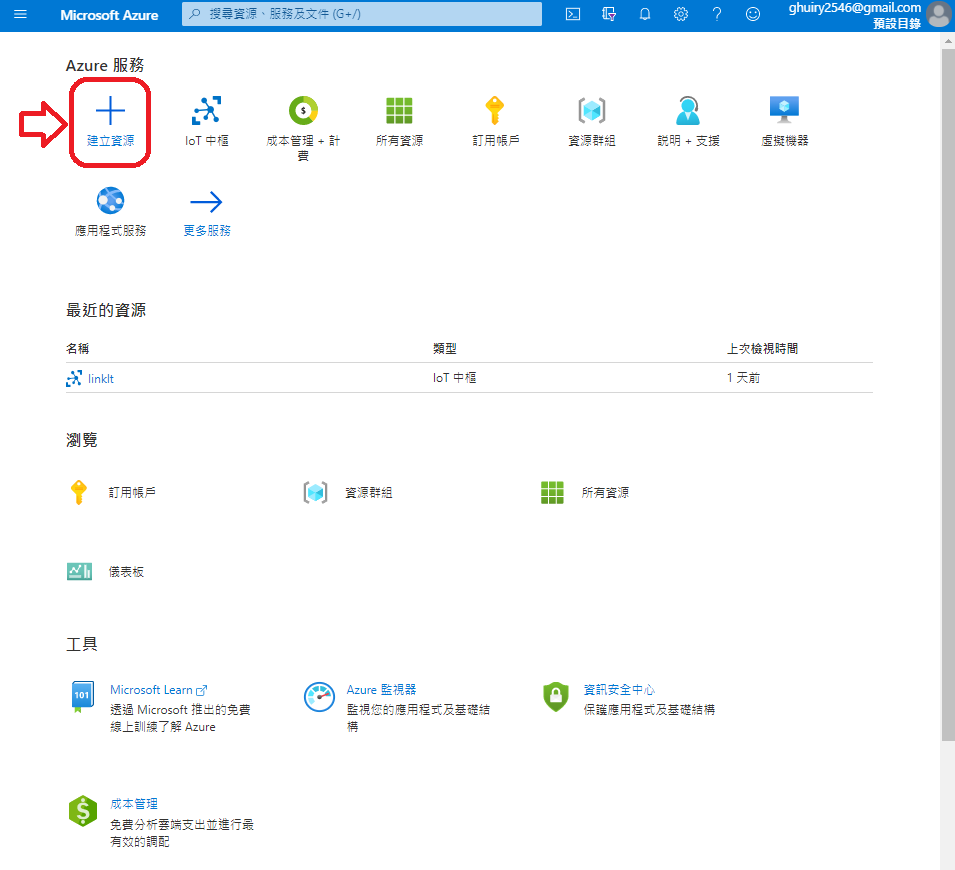
1. E-mail Verification

請點選信件中的網址。



1. Successful Verification
2. Connect to Azure
3. 登入Azure平台

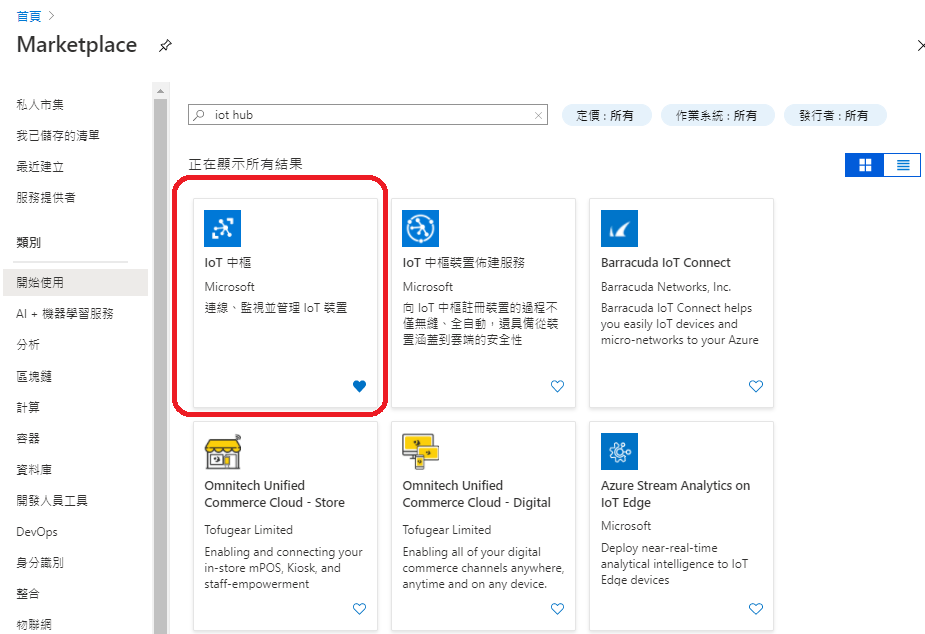
到 <https://portal.azure.com/#home> ，點選“建立資源”



1. 建立IoT Hub

輸入“ IoT Hub ”，中文版選“ IoT中樞”，英文版選“ IoT Hub”。





1. 依序輸入資料

訂閱：“Azure for Students Starter”

資源群組：請新增一個 “隨便名稱”

區域：建議使用 “東亞”

IoT中樞名稱：即為此中樞取名“都可”

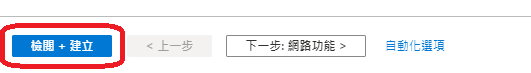
定價與級別層：請選“F1；免費層”

名稱：值 → “都可：都可”









1. 成功建立，完成畫面



1. 建立IoT device

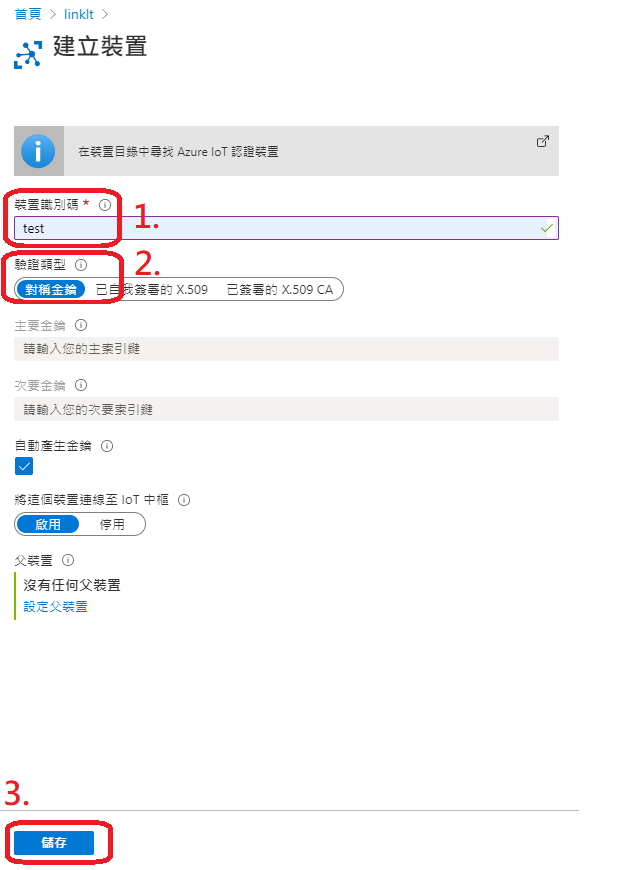
點選IoT裝置



1. 點選新增



1. 輸入裝置識別碼(名稱)，確認認證類型為“對稱金鑰”，就可儲存。

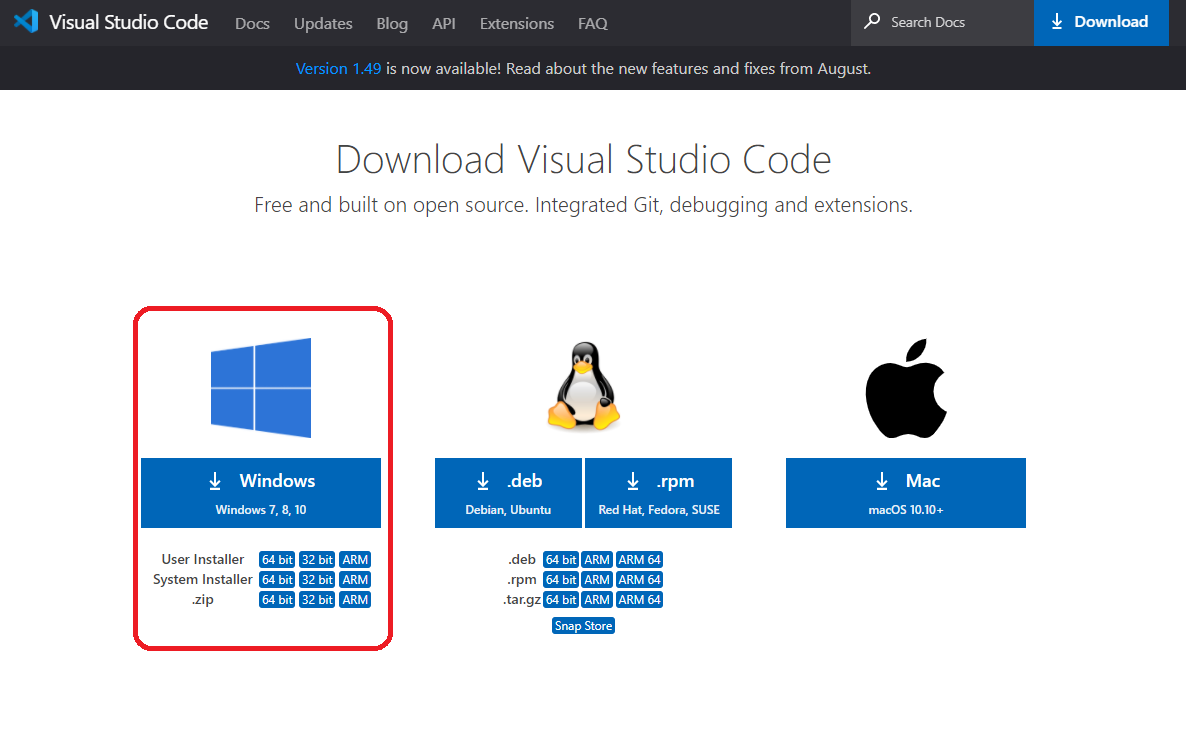


1. 成功建立，完成畫面

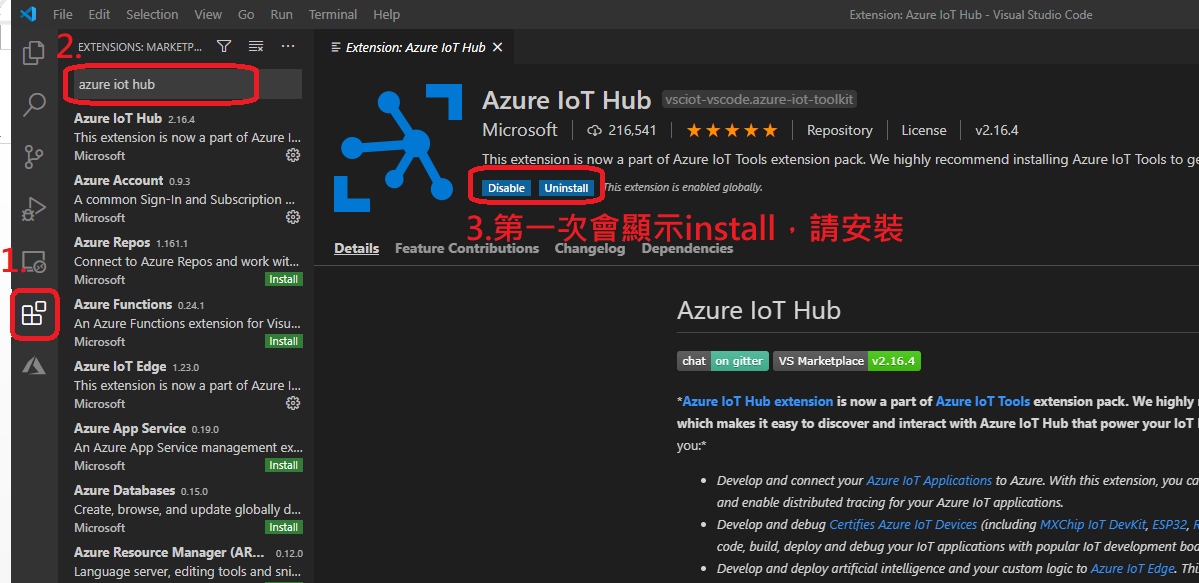


1. 下載Visual Studio Code

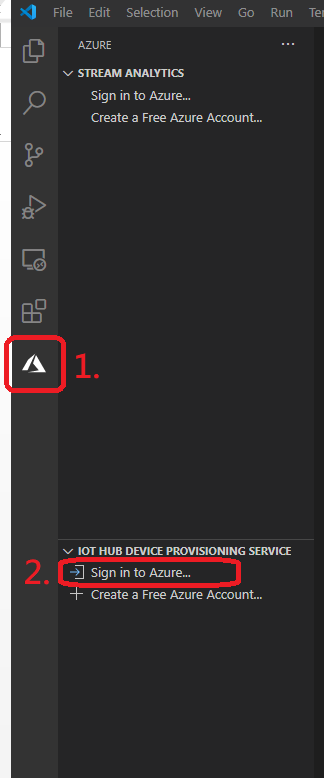
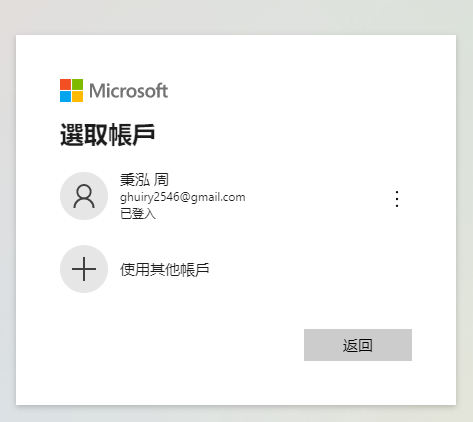
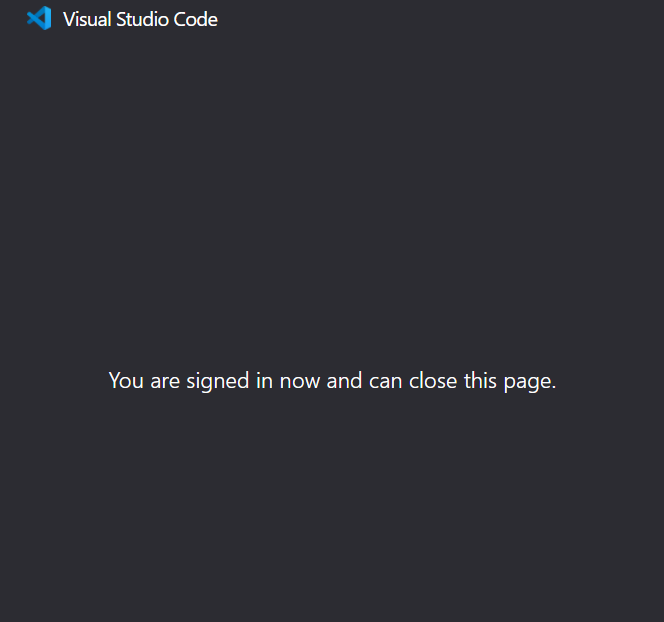
(<https://code.visualstudio.com/download>)，下載windows版本



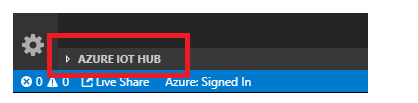
1. 開啟Visual Studio Code



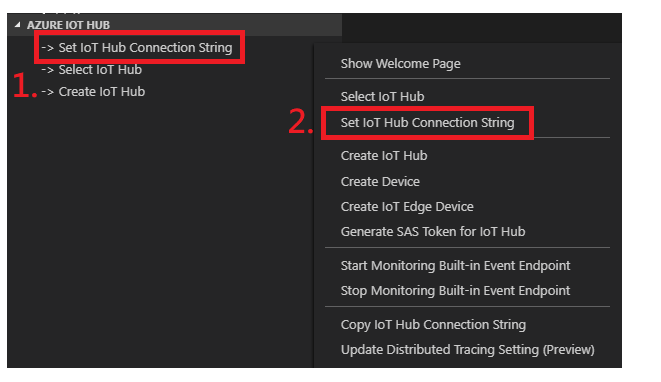
1. 連接Microsoft 帳戶使用者的Azure

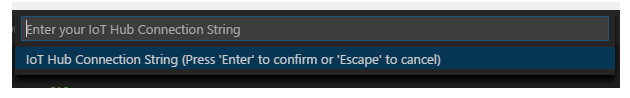
1. 單擊左下角的“ Azure IoT中心”



1. 在菜單中單擊“Set IoT Hub Connection String”。

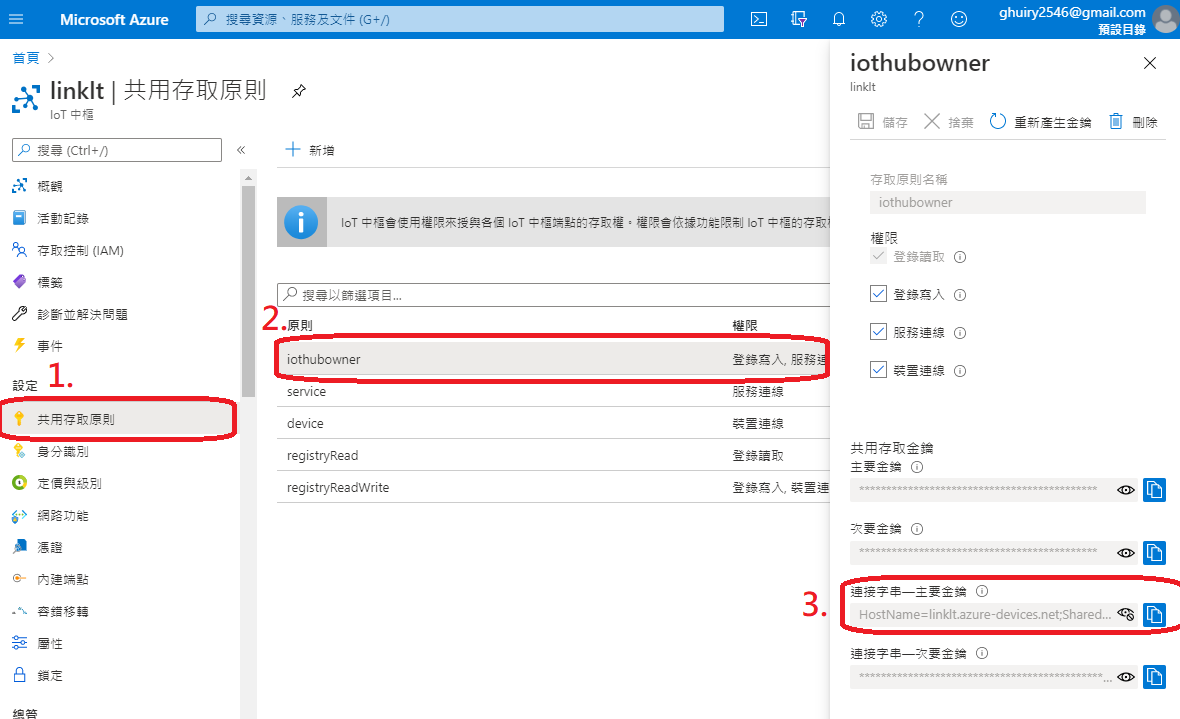


1. 輸入您的IoT Hub Connection String

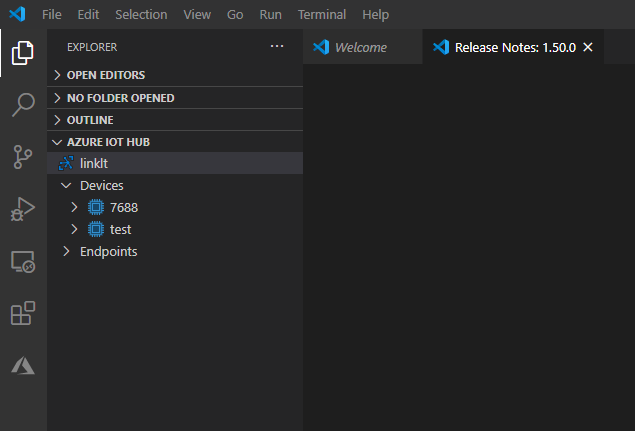


請到IoT中樞的共用存取原則裡拿取。

(這是一次性配置，請確保它是**IoT Hub Connection String**而不是 **Device Connection String)**。格式為HostName=<my-hub>.azure-devices.net;SharedAccessKeyName=<my-policy>;SharedAccessKey=<my-policy-key>。

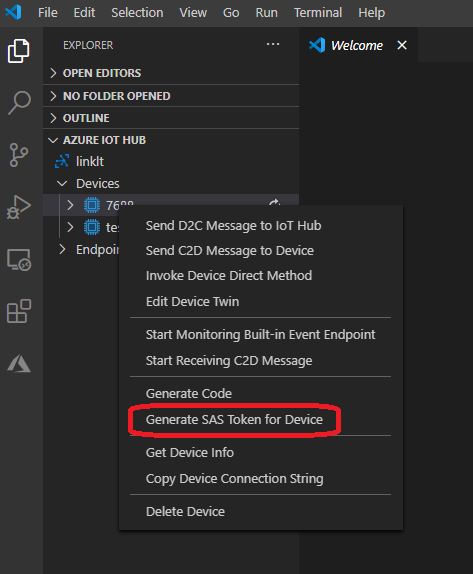
0

1. 完成Azure IoT HUB連接，成功畫面



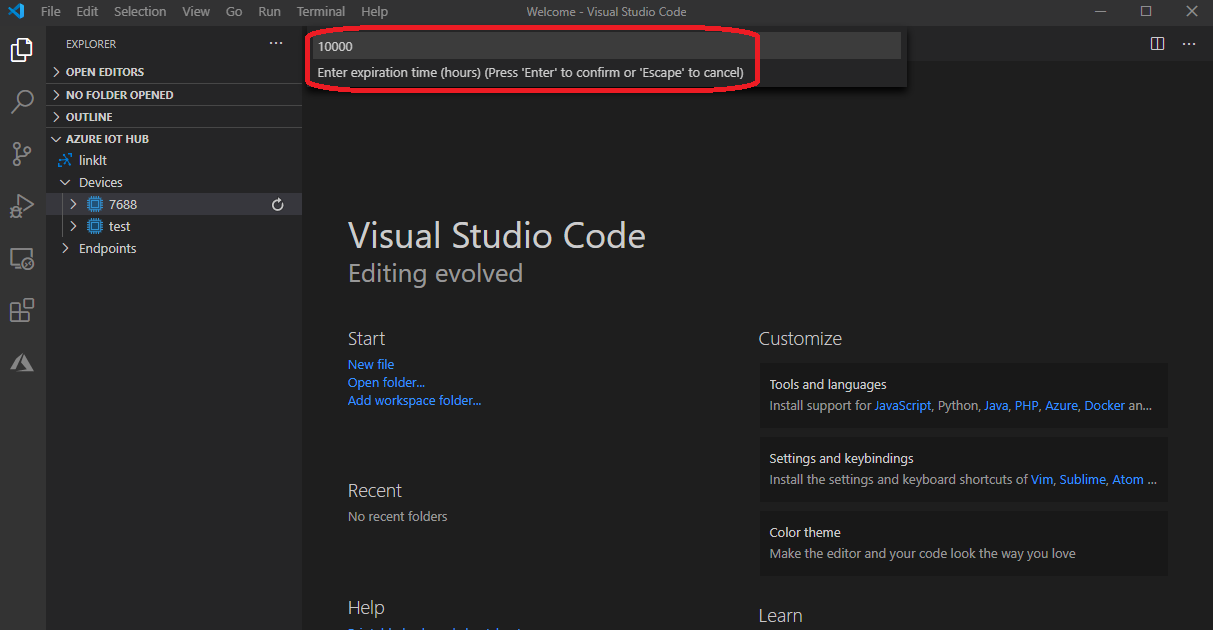
1. 獲取IoT device的SAS Token

右鍵要用的IoT device，選取“Generate SAS Token for Device”

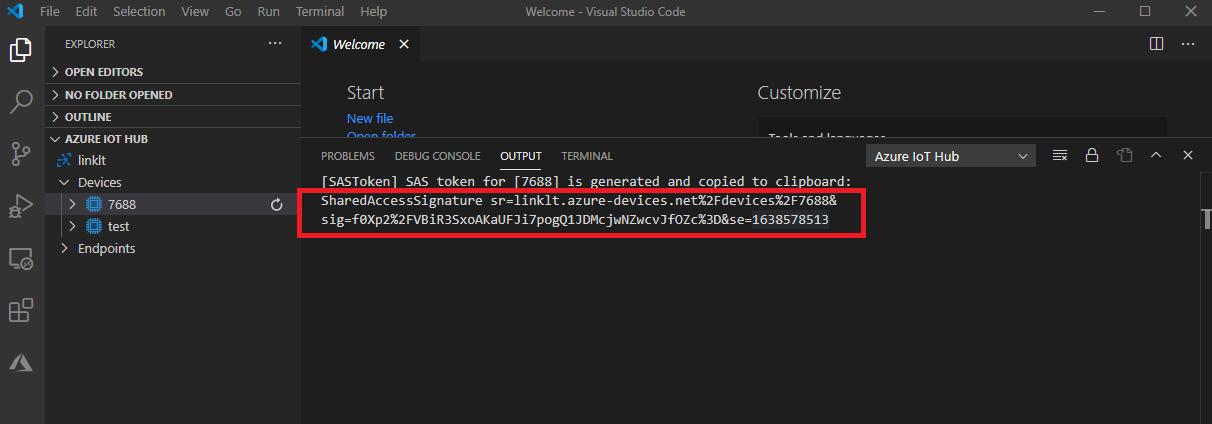


1. 輸入SAS Token使用時數

建議輸入10000 hr (一年為8700小時)

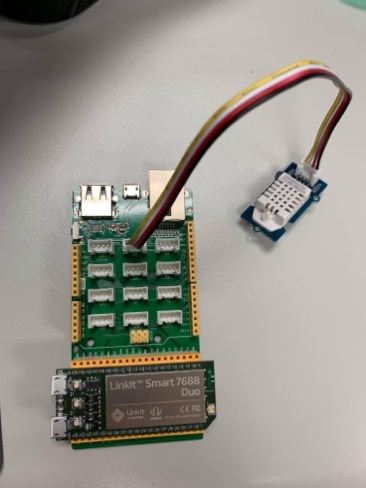


1. 記住SAS Token值(請留存，後面還會用到)



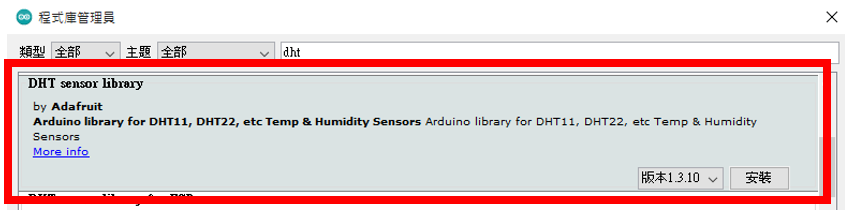
1. LinkIt 7688 Duo接DHT22

記得接上Micro USB讓它開機，並確認有連到同一個網路底下



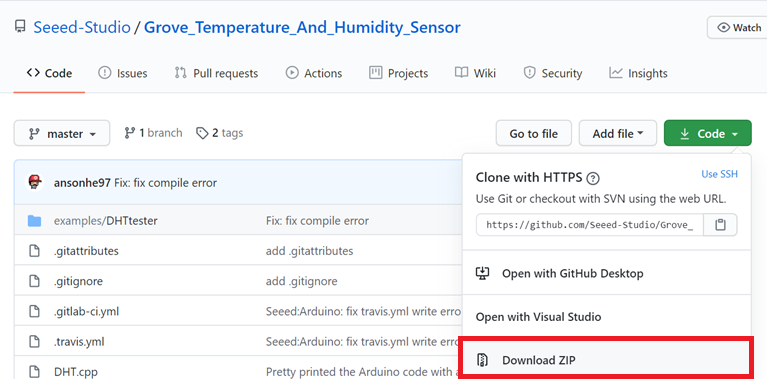
1. Arduino安裝程式庫

開啟Arduino IDE，草稿碼→匯入程式庫→管理程式庫→輸入dht→DHT sensor library安裝。

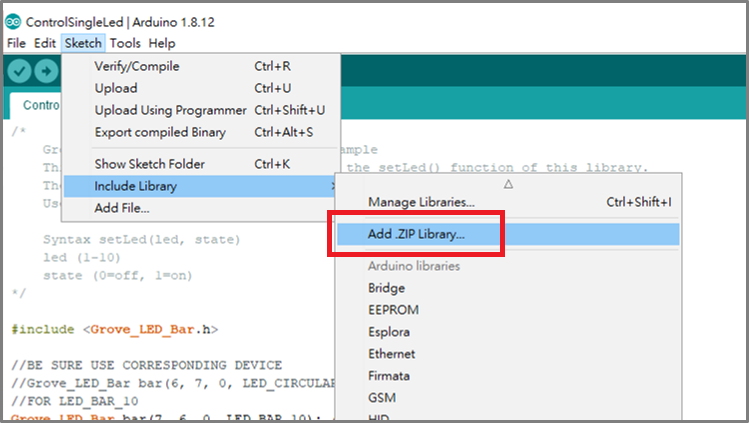


1. 下載Library

下載網址：<https://github.com/Seeed-Studio/Grove_Temperature_And_Humidity_Sensor>。



1. 匯入Library



1. Arduino IDE端程式撰寫

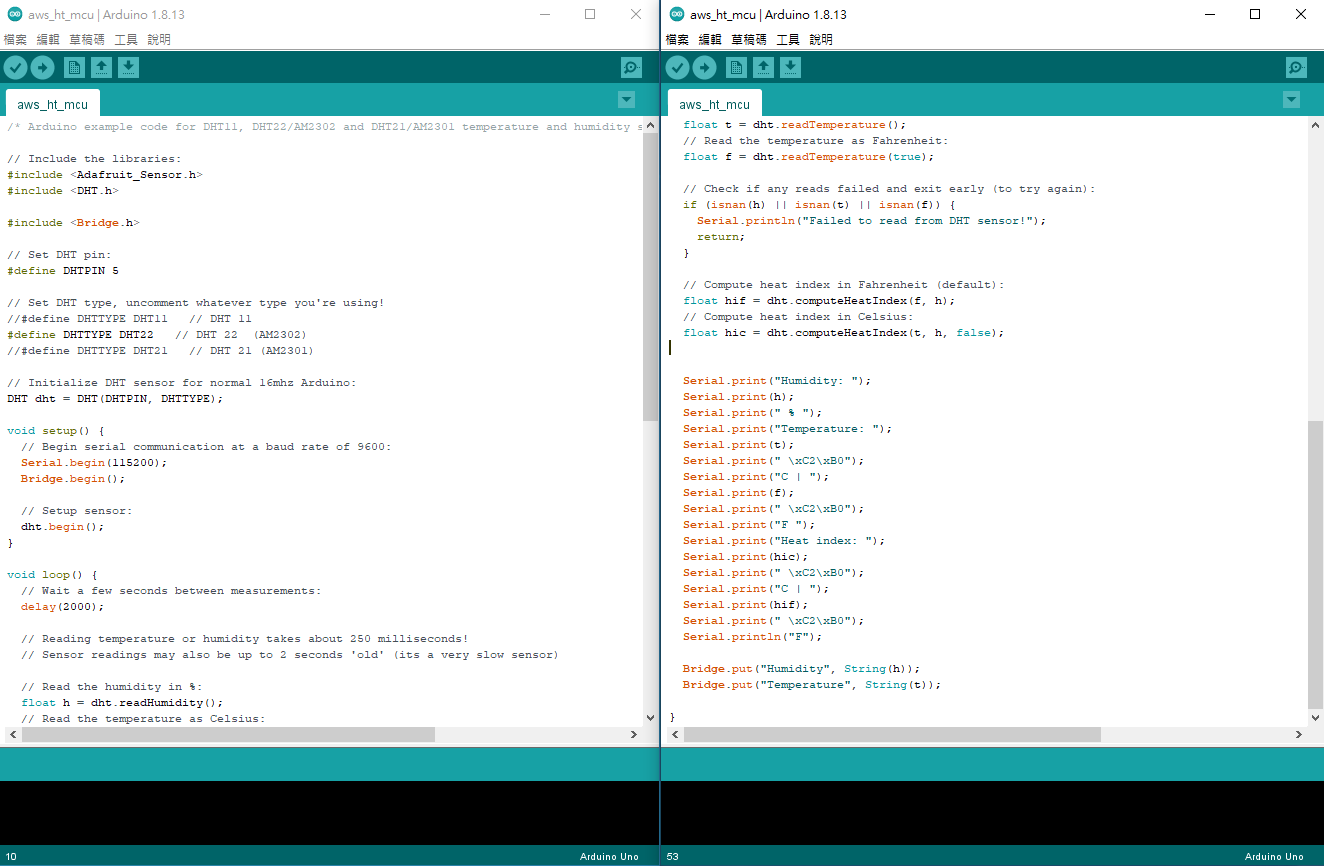
點擊File→New並輸入程式，接著點擊編譯並上傳。

MCU程式碼: 請至以下https://gitlab.com/ghuiry2546/iot/-/blob/master/azure\_ht\_mcu.ino複製。

(若想看7688是否有正確收到sensor的值，可以加上

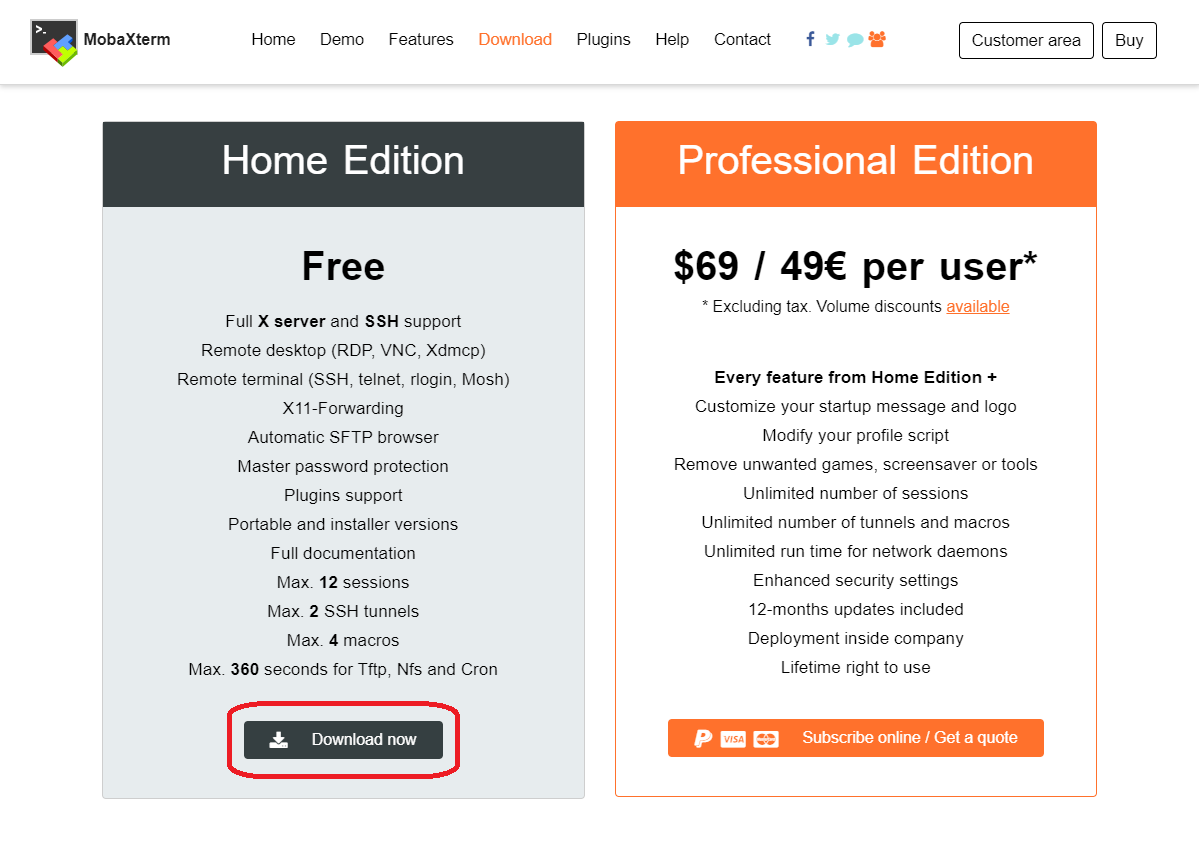
Serial.print(h);

Serial.print(t);)

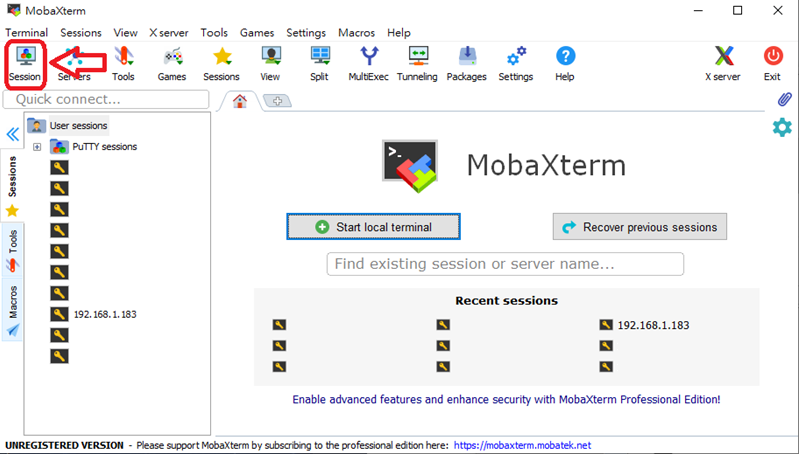


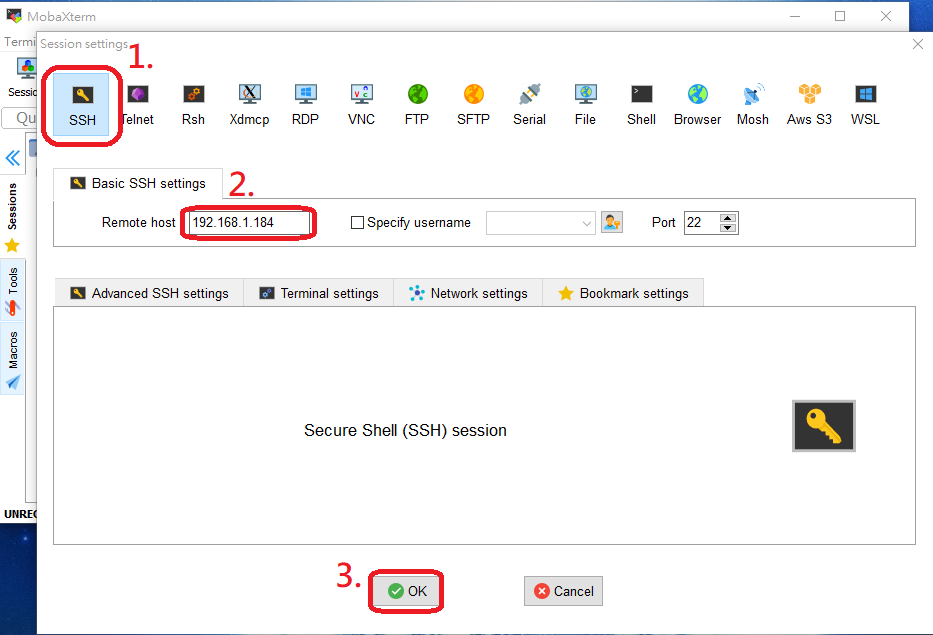
1. 安裝Mobaxterm

<https://mobaxterm.mobatek.net/download.html>



1. SSH連線

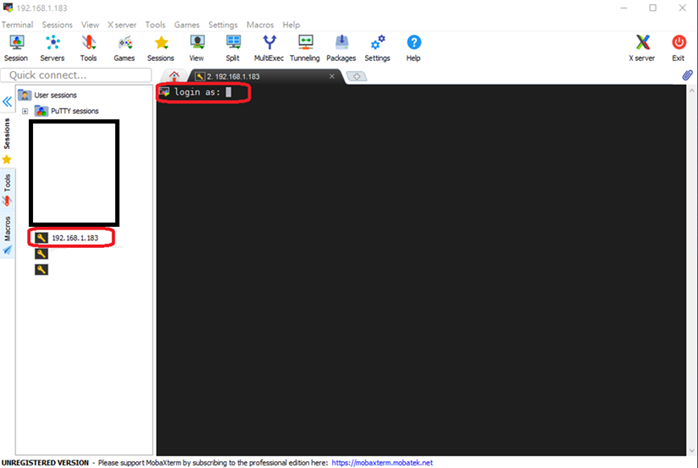




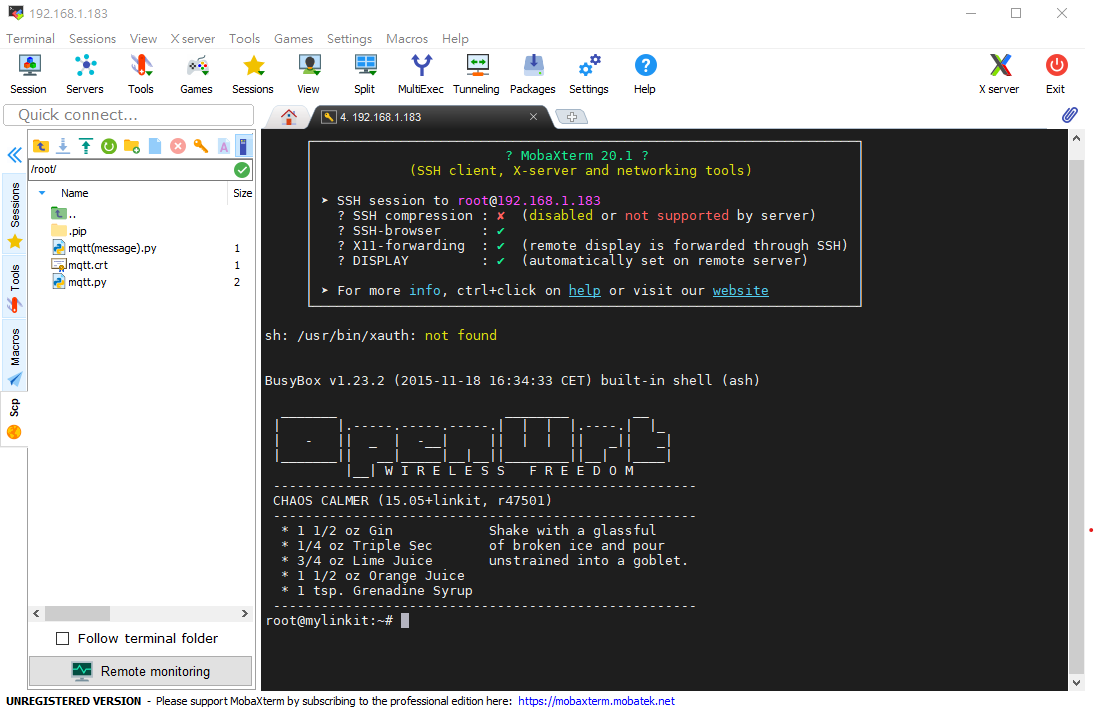
1. 成功連線，登入7688

預設帳號：root

預設密碼：lab331



1. 成功登入



1. 安裝套件

接著進行安裝更新，請直接輸入以下指令：

opkg update

pip install paho-mqtt

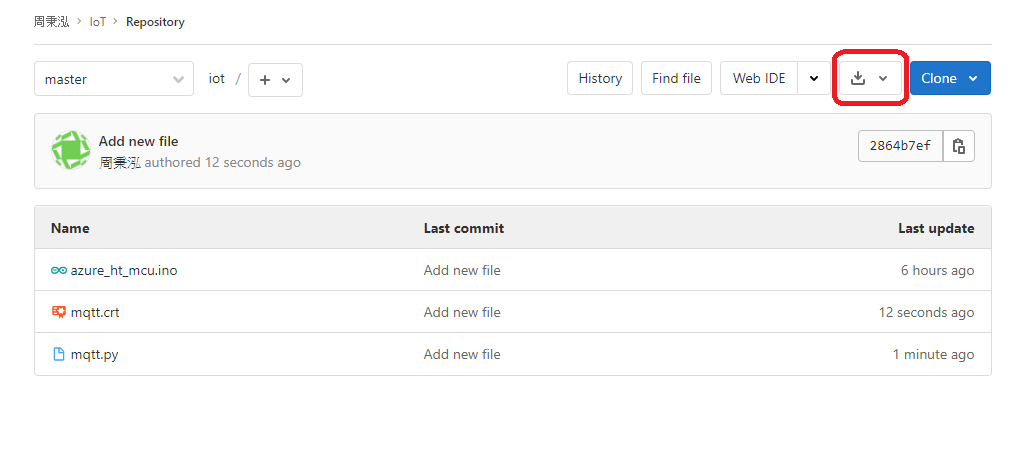
uci set yunbridge.config.disabled=0

uci commit

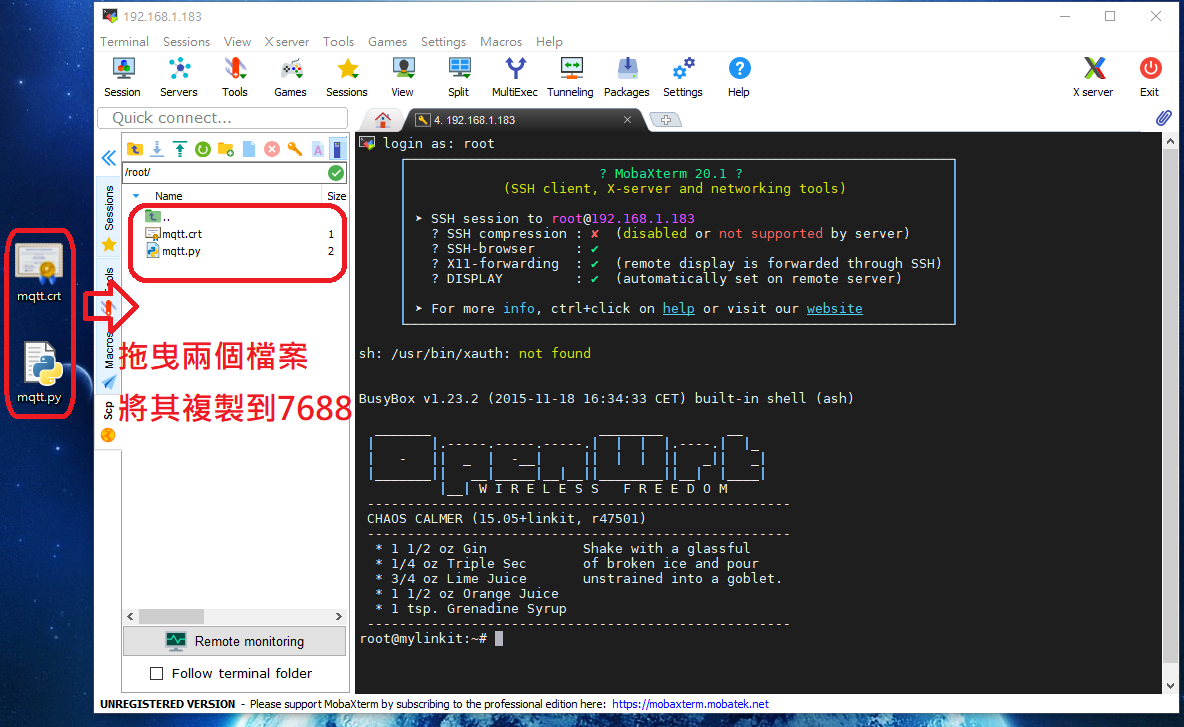
reboot

1. 下載程式碼和憑證

到<https://gitlab.com/ghuiry2546/iot/-/tree/master>下載zip檔



1. 將mqtt.crt和mqtt.py放入



1. 使用vim mqtt.py查看程式

需要修改的有

#為30步所複製的mqtt.crt路徑位置

path\_to\_root\_cert = "/root/mqtt.crt"

#IoT HUB 中IoT device的名稱

device\_id = "7688"

#為18步所複製SAS Token

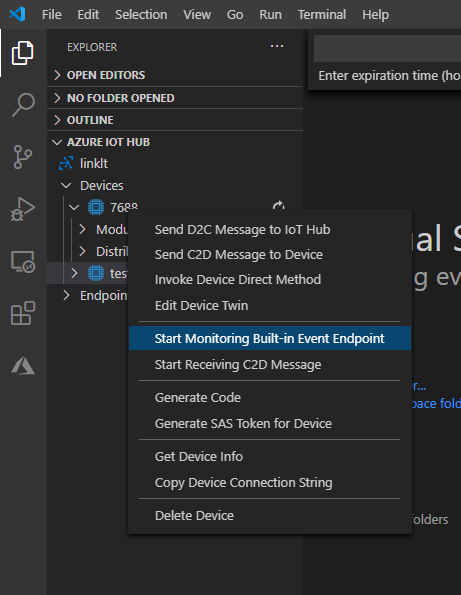
sas\_token = "SharedAccessSignature sr=linklt.azure-devices.net%2Fdevices%2F7688&sig=cF9yvJiFt0nErXoeDxbgzZ%2FANBdeQCbzdl7jaUyZADM%3D&se=1633243410"

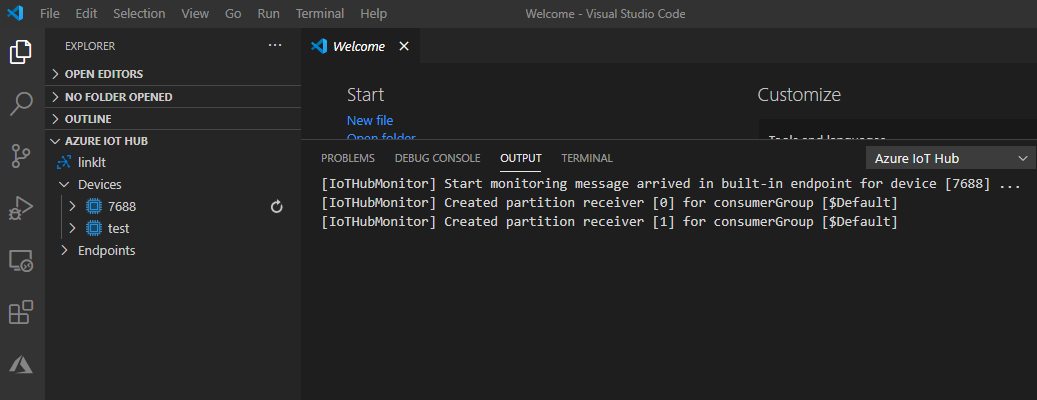
# IoT HUB名稱

iot\_hub\_name = "linklt"



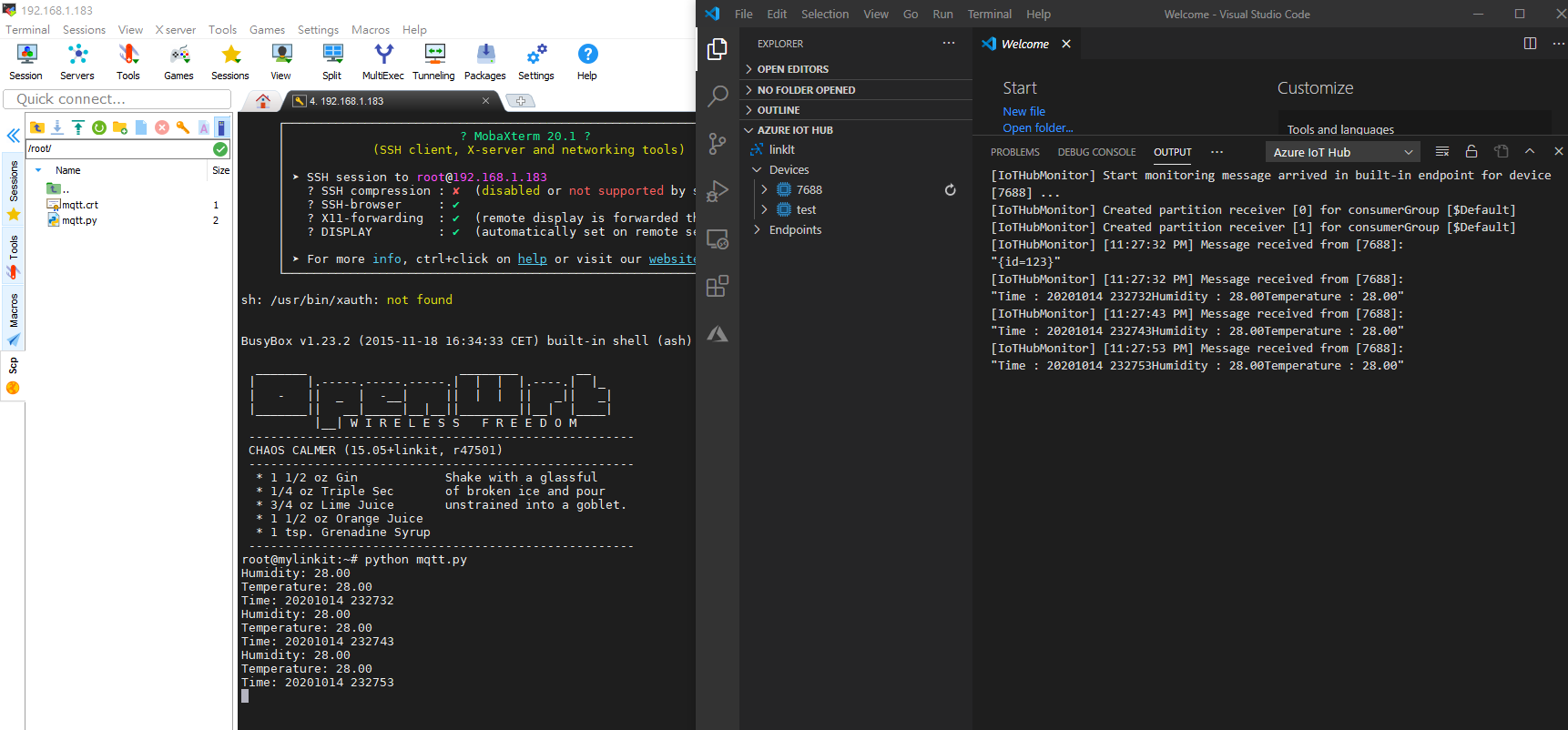
1. 開啟Visual Studio Code監控7688所上傳的資料



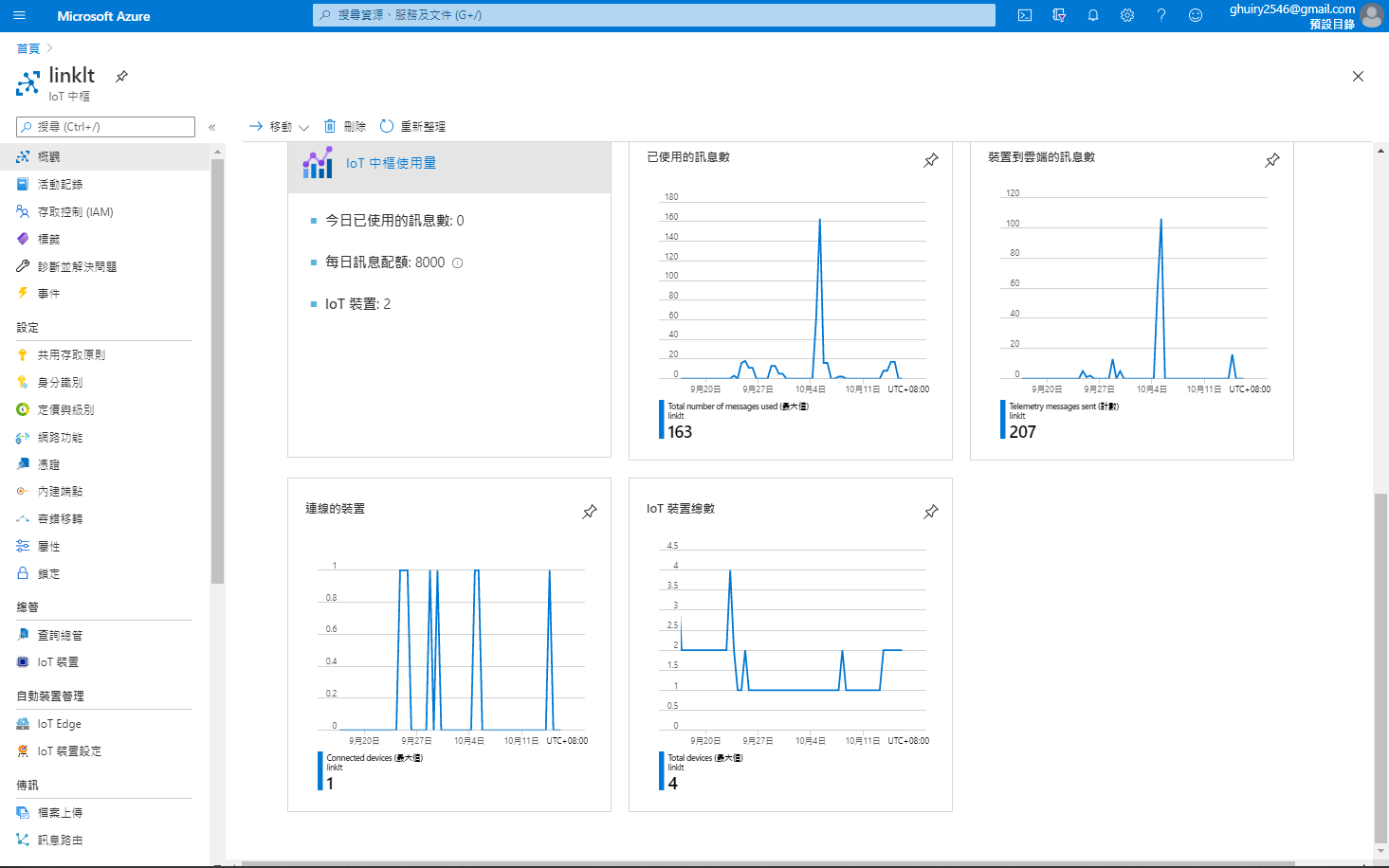


1. 執行mqtt.py

便可成功看到7688傳輸到IoT HUB的DHT資料



1. 由Azure IoT HUB查看傳送數據量



1. 參考資料

* [**Azure IoT Hub**](https://marketplace.visualstudio.com/items?itemName=vsciot-vscode.azure-iot-toolkit)
* [**使用 MQTT 通訊協定來與 IoT 中樞通訊**](https://docs.microsoft.com/zh-tw/azure/iot-hub/iot-hub-mqtt-support)
* [**azure-iot-sdk-c/certs/certs.c**](https://github.com/Azure/azure-iot-sdk-c/blob/master/certs/certs.c)