**物聯網核心技術 雲端管理平台：Azure**

**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](#_Toc53754673)

[1. 什麼是 Azure IoT 1](#_Toc53754674)

[2. 架構 1](#_Toc53754675)

[3. 各元件功能 3](#_Toc53754676)

[4. 如何使用裝置連線到Azure IoT 中樞 4](#_Toc53754677)

[二、 Applying for an Azure account 4](#_Toc53754678)

[1. Join Azure Educate 4](#_Toc53754679)

[2. 填寫個人資訊 4](#_Toc53754680)

[3. Check Your E-mail 6](#_Toc53754681)

[4. E-mail Verification 7](#_Toc53754682)

[5. Successful Verification 7](#_Toc53754683)

[三、 Connect to Azure 7](#_Toc53754684)

[1. 登入Azure平台 7](#_Toc53754685)

[2. 建立IoT Hub 7](#_Toc53754686)

[3. 依序輸入資料 9](#_Toc53754687)

[4. 成功建立，完成畫面 10](#_Toc53754688)

[5. 建立IoT device 11](#_Toc53754689)

[6. 點選新增 11](#_Toc53754690)

[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](#_Toc53754691)

[8. 成功建立，完成畫面 12](#_Toc53754692)

[9. 下載Visual Studio Code 13](#_Toc53754693)

[10. 開啟Visual Studio Code 13](#_Toc53754694)

[11. 連接Microsoft 帳戶使用者的Azure 13](#_Toc53754695)

[12. 單擊左下角的“ Azure IoT中心” 14](#_Toc53754696)

[13. 在菜單中單擊“Set IoT Hub Connection String”。 14](#_Toc53754697)

[14. 輸入您的IoT Hub Connection String 14](#_Toc53754698)

[15. 完成Azure IoT HUB連接，成功畫面 15](#_Toc53754699)

[16. 獲取IoT device的SAS Token 16](#_Toc53754700)

[17. 輸入SAS Token使用時數 17](#_Toc53754701)

[18. 記住SAS Token值(請留存，後面還會用到) 18](#_Toc53754702)

[19. LinkIt 7688 Duo接DHT22 18](#_Toc53754703)

[20. Arduino安裝程式庫 18](#_Toc53754704)

[21. 下載Library 19](#_Toc53754705)

[22. 匯入Library 19](#_Toc53754706)

[23. Arduino IDE端程式撰寫 19](#_Toc53754707)

[24. 安裝Mobaxterm 20](#_Toc53754708)

[25. SSH連線 20](#_Toc53754709)

[26. 成功連線，登入7688 21](#_Toc53754710)

[27. 成功登入 21](#_Toc53754711)

[28. 安裝套件 22](#_Toc53754712)

[29. 下載程式碼和憑證 22](#_Toc53754713)

[30. 將mqtt.crt和mqtt.py放入 22](#_Toc53754714)

[31. 使用vim mqtt.py查看程式 23](#_Toc53754715)

[32. 開啟Visual Studio Code監控7688所上傳的資料 24](#_Toc53754716)

[33. 執行mqtt.py 25](#_Toc53754717)

[34. 由Azure IoT HUB查看傳送數據量 26](#_Toc53754718)

[四、 參考資料 26](#_Toc53754719)

1. Introduction
2. 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. 持續或間歇連線到雲端的實體物件或物聯網 (例如工業設備、裝置或感應器)；包括以下三種：

保護您的邊緣裝置和硬體 (Azure Sphere、RTO、Windows IoT)

簡化與認證裝置的連線作業 (IoT 隨插即用和 Azure 認證裝置)

將應用程式和工作負載推送至邊緣 (Azure IoT Edge、Azure Stack、Azure Stack Edge)

1. 由物聯網收集的資訊，再由人員或 AI 分析並轉換為可付諸行動的知識；包括以下三種：

彈性地連線及管理您的裝置資料 (Azure IoT 中樞和 Azure IoT Central)

轉換您的資料以進行互通 (Azure API)

建立整個環境的數位模型 (Azure Digital Twins)

1. 人們回應這些見解並將該見解連結到其商務的方式，以及所使用的系統和工具；包括以下三種：

與您的商務應用程式整合 (Power BI、Dynamics 365、CRM、Field Services)

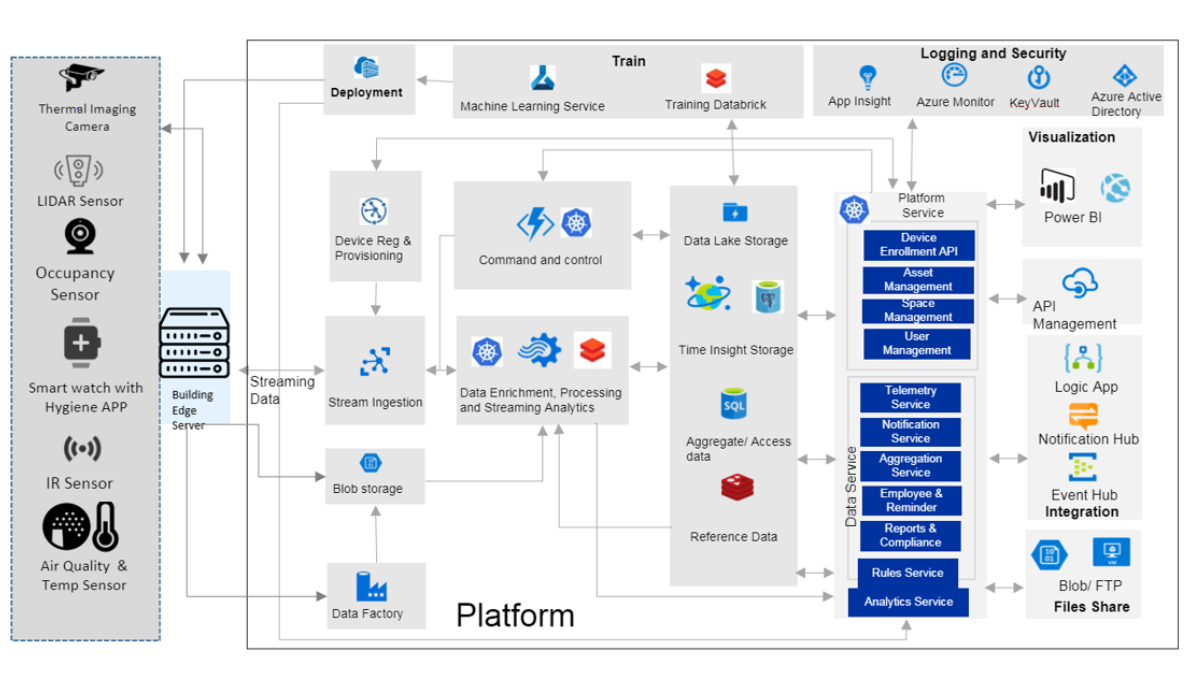
從邊緣到雲端的安全性 (Azure Defender for IoT)

建置連線的生態系統



1. 架構

Azure大致提供的服務及流程，如下圖所示，我們會依序做介紹。



1. 不同的裝置會收集資料：熱映射攝影機、LIDAR 感應器、佔用感應器、智慧監看與防護應用程式、IR 感應器，以及空氣品質和溫度感應器。 該資料會流入建築物的 IoT Edge 伺服器。
2. 伺服器會將資料摘要至 Azure IoT 中樞、Azure Data Factory 和 Azure Blob 儲存體。
3. Azure IoT 中樞會透過 Azure Kubernetes Service (AKS) 來傳送資料。
4. AKS 會將資料路由傳送，讓 Azure 串流分析和 Azure Databricks 可以分析並充實它。
5. 系統本身會將處理過的資料傳送至各種資料存放區：
   * Data Lake-Azure Data Lake Storage
   * 時間深入解析-Azure Cosmos DB 和適用於 PostgreSQL 的 Azure 資料庫
   * 匯總/存取資料-Azure SQL Database
   * 參考資料-Azure Cache for Redis
6. 從該處，不同的服務和 Api 會使用資料：
   * 資料服務-這些服務會取用資料、進行處理，並將它傳遞給下游服務，以通知使用者有建築物偵測到的任何異常狀況。
   * 平臺服務-這些服務會分析裝置、資產、空間和使用者度量，並將資訊傳遞給 Power BI 的 web 應用程式實例。 Power BI 會以有意義且有用的方式來顯示統計資料。
   * 定型-Azure Databricks 和 Azure Machine Learning 接受資料並更新深度學習模型。建築物會使用 Azure Container Registry 將模型散發至 IoT Edge 伺服器和資料服務。
   * 命令和控制項 AKS 會將平臺服務的資料傳遞給 Azure Functions。
7. 各元件功能
   * [Azure IoT Edge](https://azure.microsoft.com/services/iot-edge/) 可讓您透過標準容器，部署雲端工作負載，以在物聯網 (IoT) Edge 裝置上執行。 人工智慧、Azure 和協力廠商服務，或您自己的商務邏輯等工作負載。
   * [Azure IoT 中樞](https://azure.microsoft.com/services/iot-hub/) 可讓您的 IoT 應用程式與其管理的裝置之間進行高度安全且可靠的通訊。
   * [Azure Kubernetes Service](https://azure.microsoft.com/services/kubernetes-service/) 使用完全受控的 Kubernetes 服務，更輕鬆地部署和管理容器化應用程式。
   * [Azure 串流分析](https://azure.microsoft.com/services/stream-analytics/) 是一種便於使用的即時分析服務，專為任務關鍵性工作負載而設計。
   * [Azure Databricks](https://azure.microsoft.com/services/databricks/) 從您的所有資料中解除見解，並建立人工智慧 (AI) 解決方案。 幾分鐘內即可設定您的 Apache Spark 環境、自動調整，以及在互動式工作區中的共用專案上共同作業。
   * [Azure Data Lake Storage](https://azure.microsoft.com/services/storage/data-lake-storage/) -適用于海量資料分析的高度可擴充且符合成本效益的 data Lake 解決方案。
   * [Azure Cosmos DB](https://azure.microsoft.com/services/cosmos-db/) -完全受控的 NoSQL 資料庫服務，適用于新式應用程式開發。 Azure Cosmos DB 保證單一位數毫秒的回應時間。 它也保證99.999% 的可用性受 Sla、自動化和立即的擴充性，以及適用于 MongoDB 和 Cassandra 的開放原始碼 Api 所支援。
   * [適用於 PostgreSQL 的 Azure 資料庫](https://azure.microsoft.com/services/postgresql/) 著重于應用程式創新，而不是資料庫管理，具備完全受控且智慧型的適用於 PostgreSQL 的 Azure 資料庫。
   * [Azure SQL Database](https://azure.microsoft.com/services/sql-database/) -針對雲端打造的智慧型、可擴充的關係資料庫服務。 它是長時間，而且一律是最新的，具備 AI 技術和自動化的功能，可為您優化效能和持久性。
   * [Azure Cache For Redis](https://azure.microsoft.com/services/cache/) -完全受控且符合開放原始碼規範的記憶體內部資料存放區，可提供快速、可擴充的應用程式。
   * [Azure Machine Learning](https://azure.microsoft.com/services/machine-learning/) 讓開發人員和資料科學家能夠更快地建立、定型及部署機器學習模型。 連續重新訓練、更新及改善資料收集和分析，可讓模型在一段時間後變得更好。
   * [Azure 監視器](https://azure.microsoft.com/services/monitor/) 會從您的 Azure 和內部部署環境收集、分析及採取遙測資料的動作。
   * [Power BI](https://powerbi.microsoft.com/) 可讓小組成員找出資料中隱藏的見解。
   * [Azure Functions](https://azure.microsoft.com/services/functions/) 可讓您在本機建立和偵錯工具，而不需要在雲端中進行額外的安裝、部署和操作，以及使用觸發程式和系結來整合服務。
   * [AZURE ACTIVE DIRECTORY](https://azure.microsoft.com/services/active-directory/)、 [Azure API 管理](https://azure.microsoft.com/services/api-management/)、 [azure Blob 儲存體](https://azure.microsoft.com/services/storage/blobs/)、 [Azure Container Registry](https://azure.microsoft.com/services/container-registry/)、 [Azure Data Factory](https://azure.microsoft.com/services/data-factory/)、 [Azure 事件中樞](https://azure.microsoft.com/services/event-hubs/)、 [Azure Key Vault](https://azure.microsoft.com/services/key-vault/)、 [azure 邏輯應用程式](https://azure.microsoft.com/services/logic-apps/)、 [azure 通知中樞](https://azure.microsoft.com/services/notification-hubs/)、 [azure vm](https://azure.microsoft.com/services/virtual-machines/)、 [azure Webapp Service](https://azure.microsoft.com/services/app-service/web/)…..等等。
8. 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

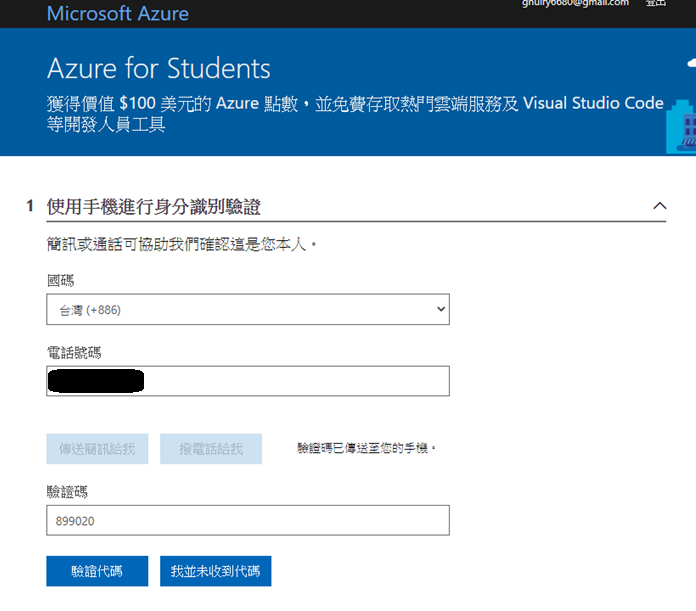
到Azure Educate網址註冊<https://azure.microsoft.com/zh-tw/free/students/>。



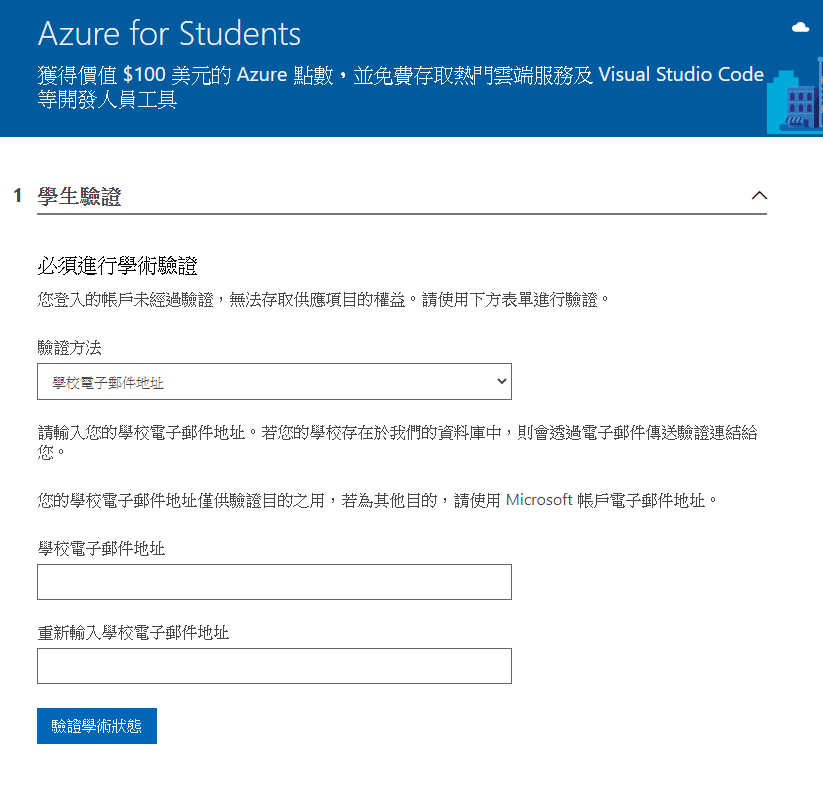
1. 填寫個人資訊

範例：

國碼：台灣(+886)

電話：900-000-000

學校信箱：g0\*\*10\*\*\*@ccu.edu.tw



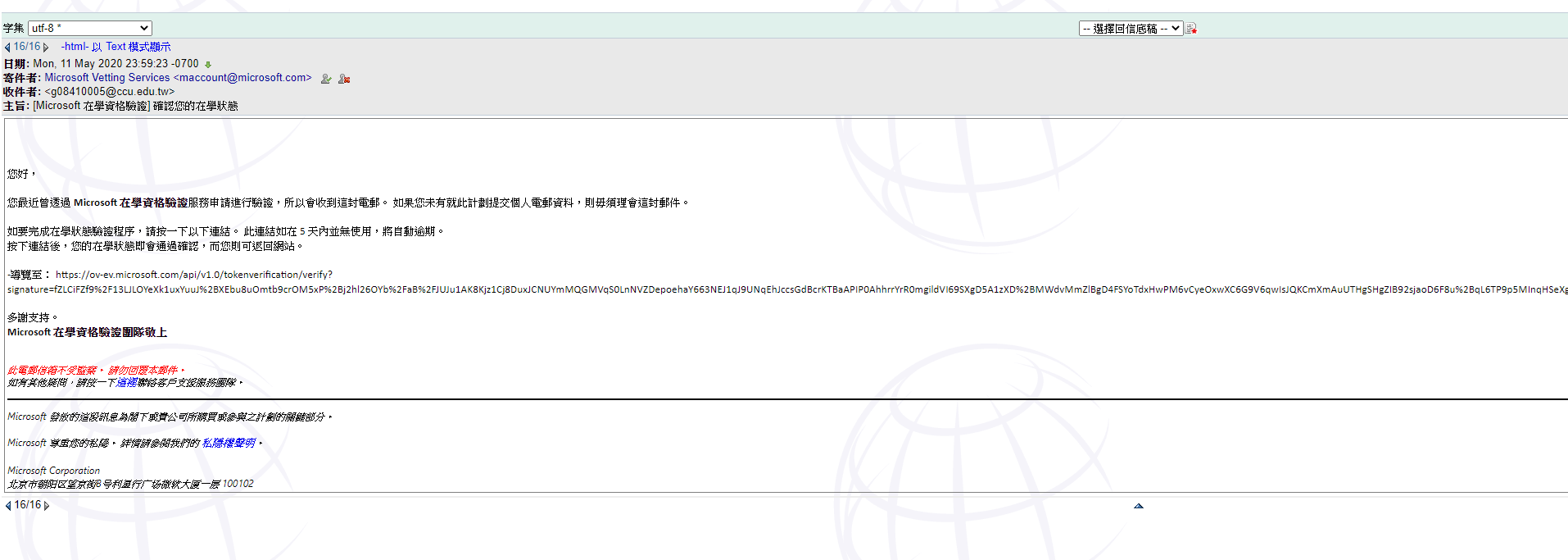
1. Check Your E-mail

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



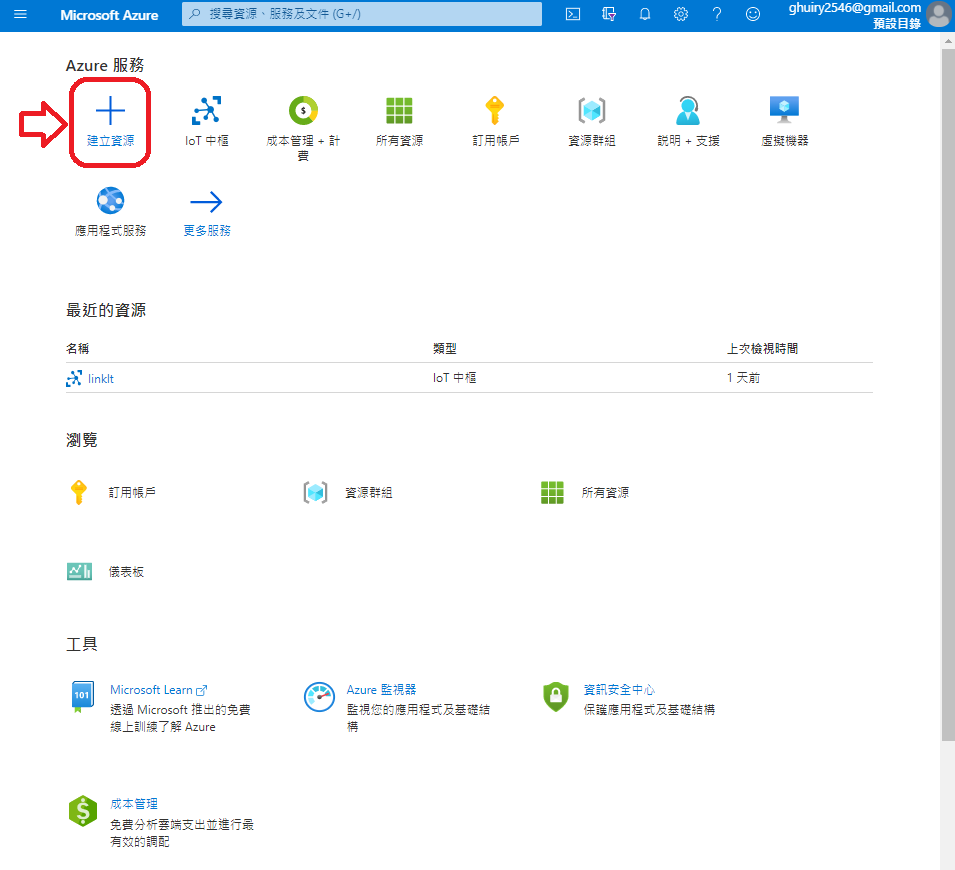
1. E-mail Verification

請點選信件中的網址。



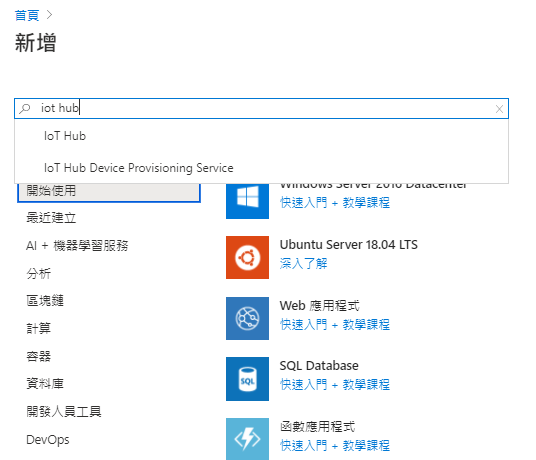
1. Successful Verification
2. Connect to Azure
3. 登入Azure平台

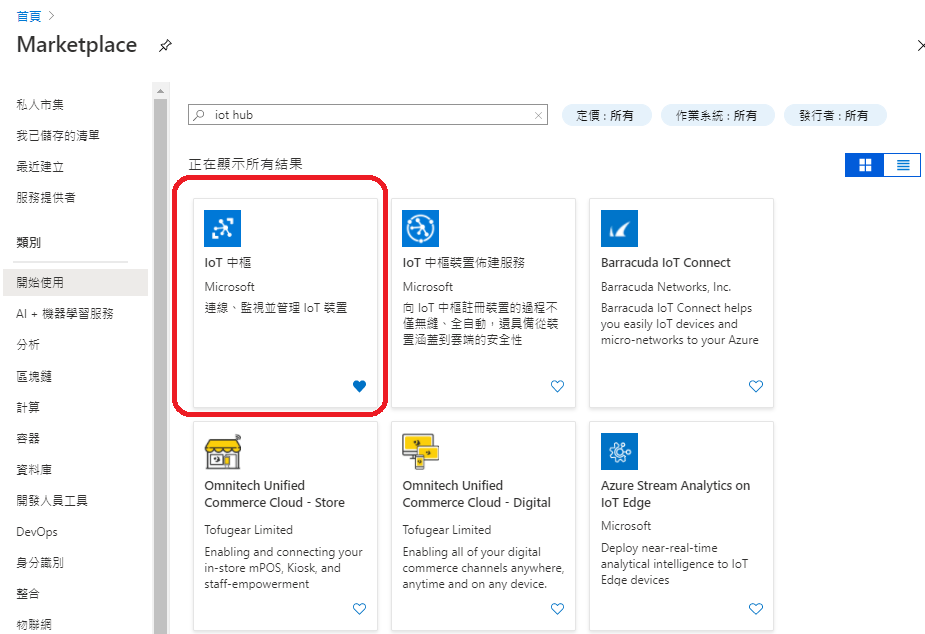
到 <https://portal.azure.com/#home> ，點選“建立資源”



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”

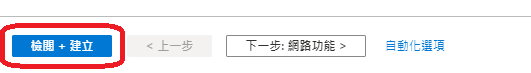
定價與級別層：請選“F1；免費層”

名稱：值 → “都可：都可”









1. 成功建立，完成畫面



1. 建立IoT device

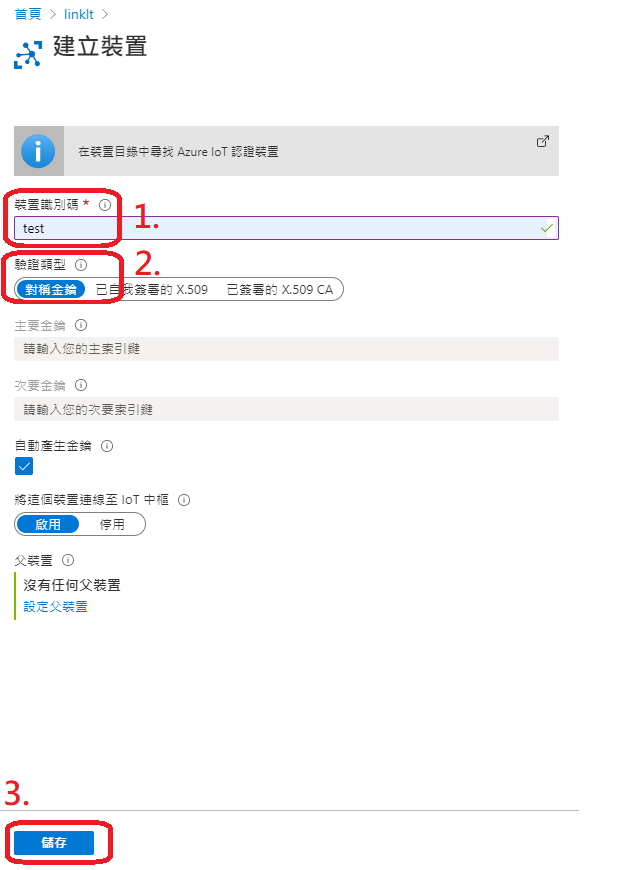
點選IoT裝置



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。

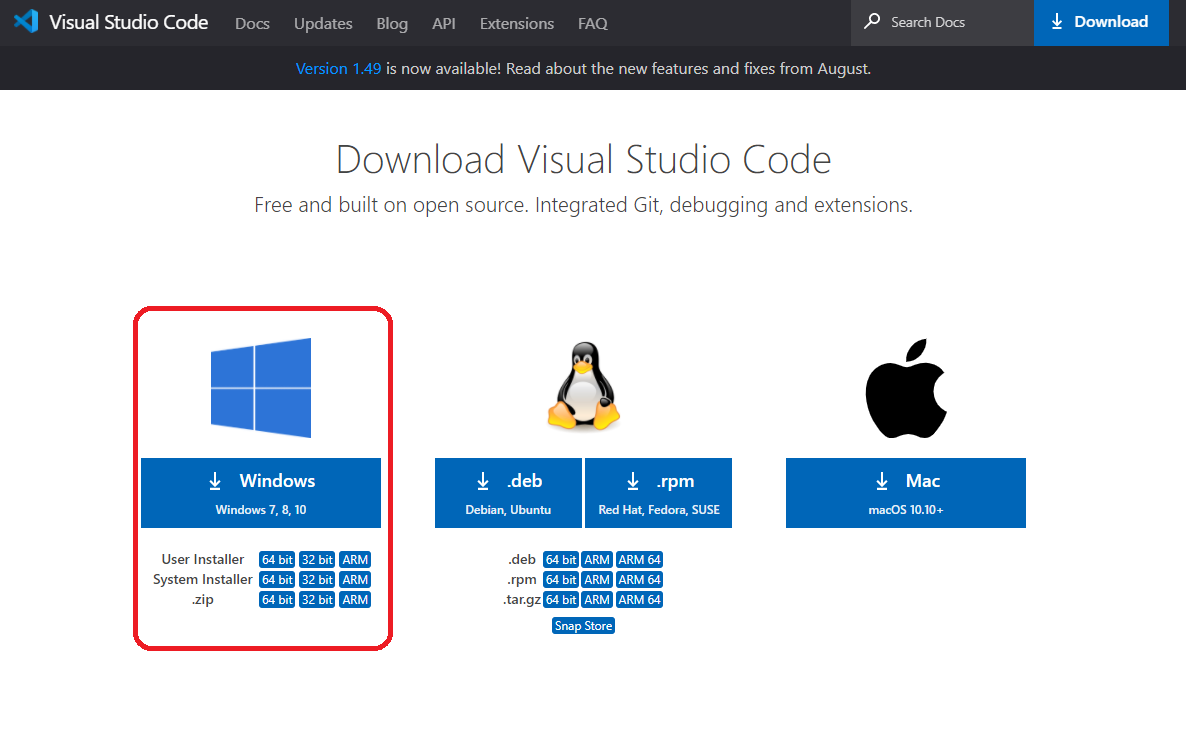


1. 成功建立，完成畫面

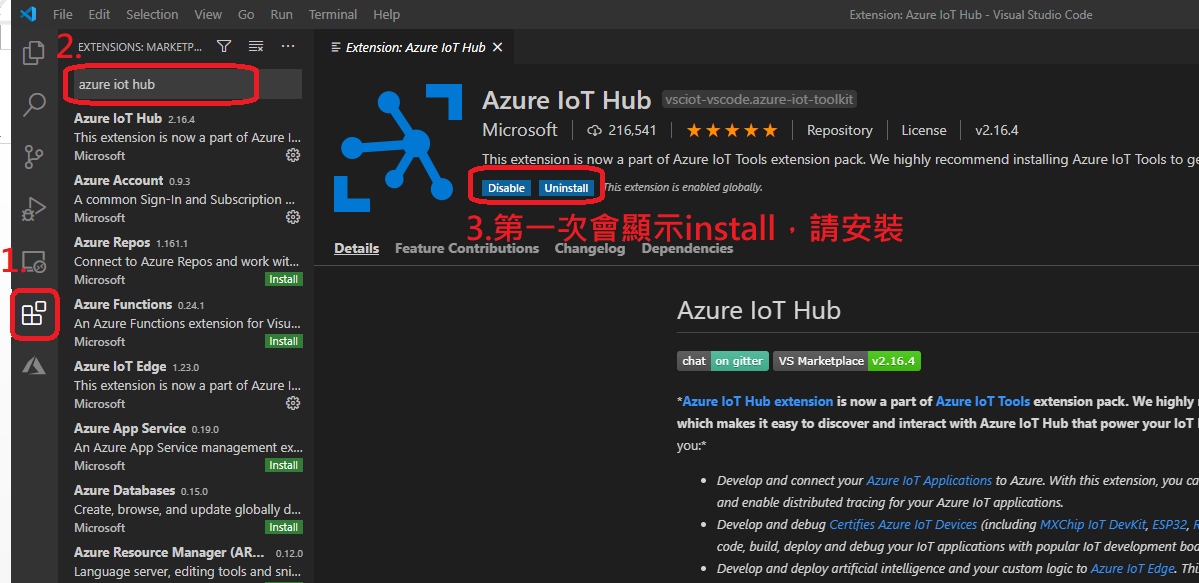


1. 下載Visual Studio Code

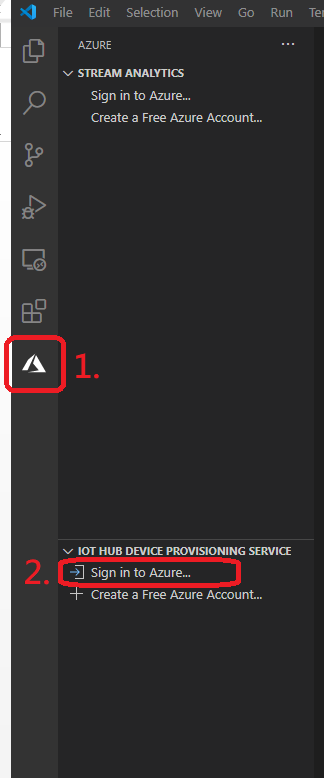
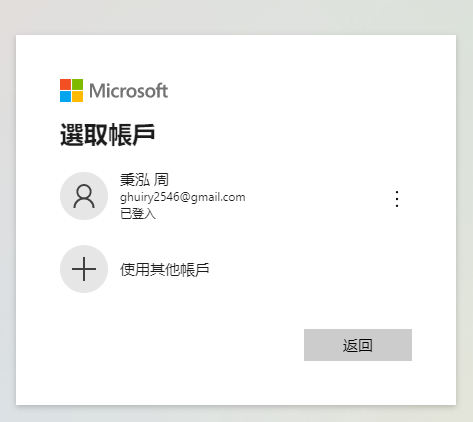
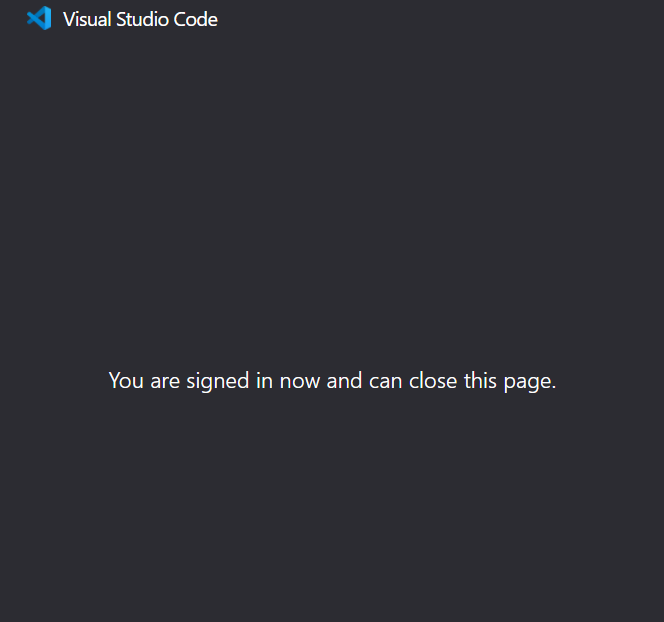
(<https://code.visualstudio.com/download>)，下載windows版本



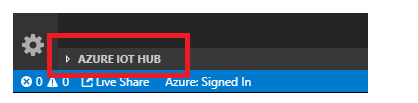
1. 開啟Visual Studio Code



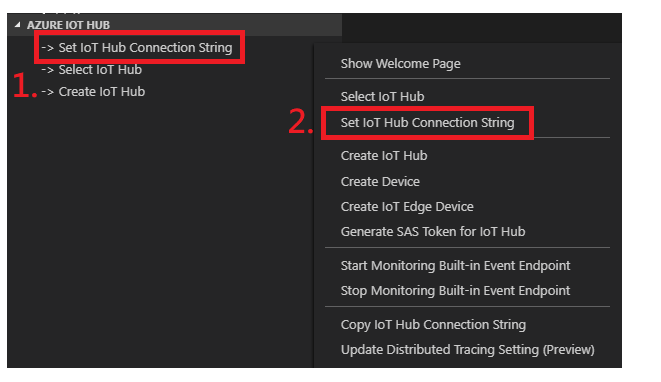
1. 連接Microsoft 帳戶使用者的Azure

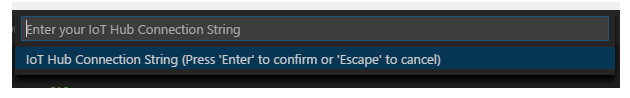
1. 單擊左下角的“ Azure IoT中心”



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

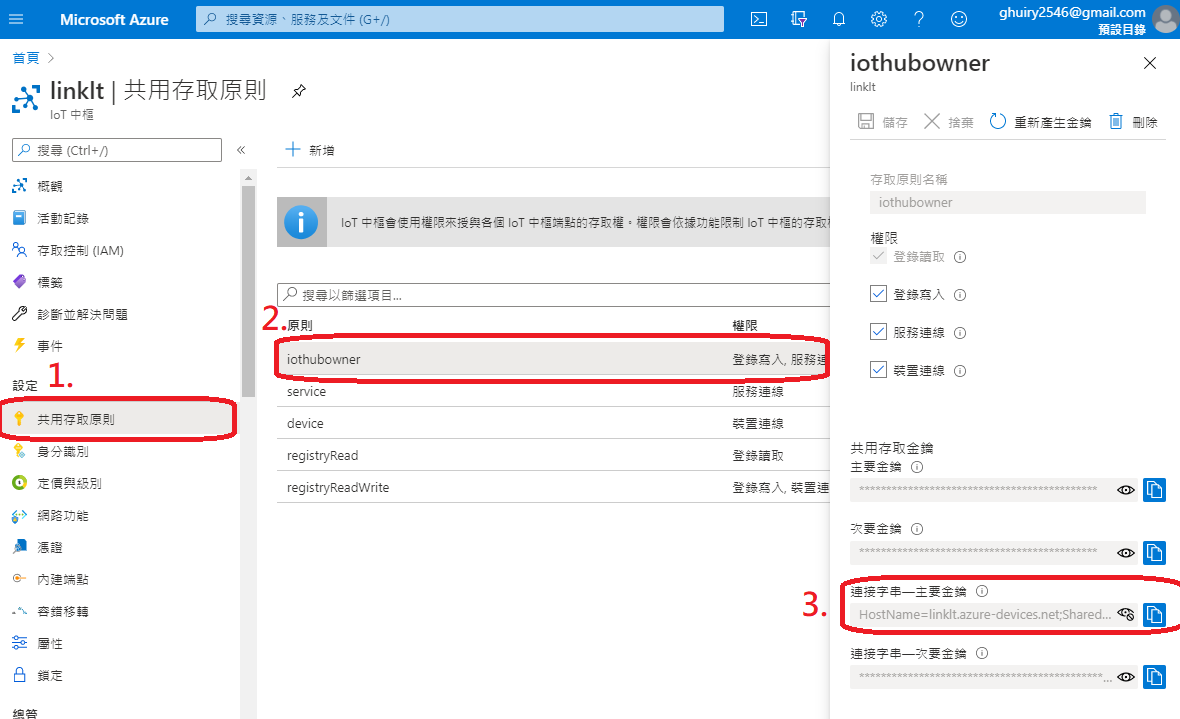


1. 輸入您的IoT Hub Connection String

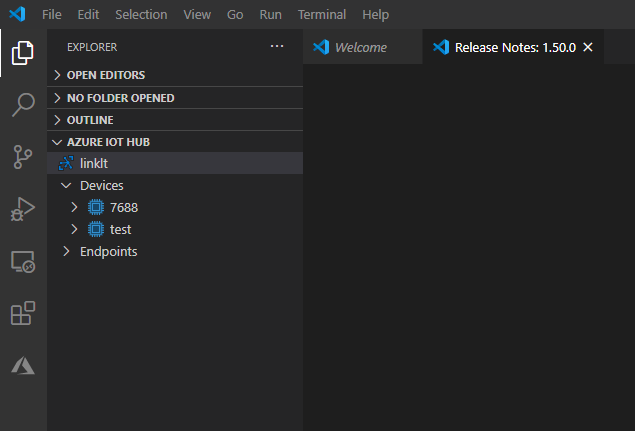


請到IoT中樞的共用存取原則裡拿取。

(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>。

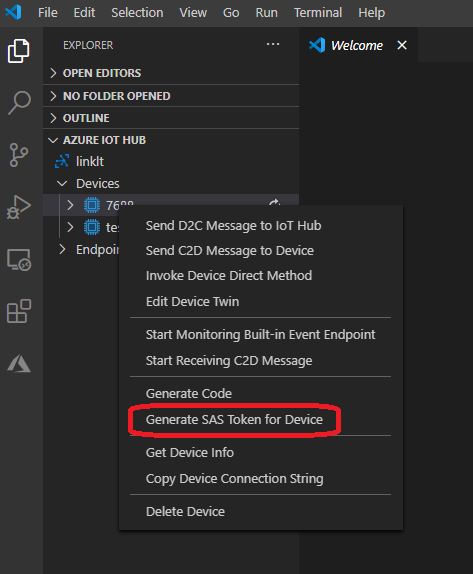
0

1. 完成Azure IoT HUB連接，成功畫面



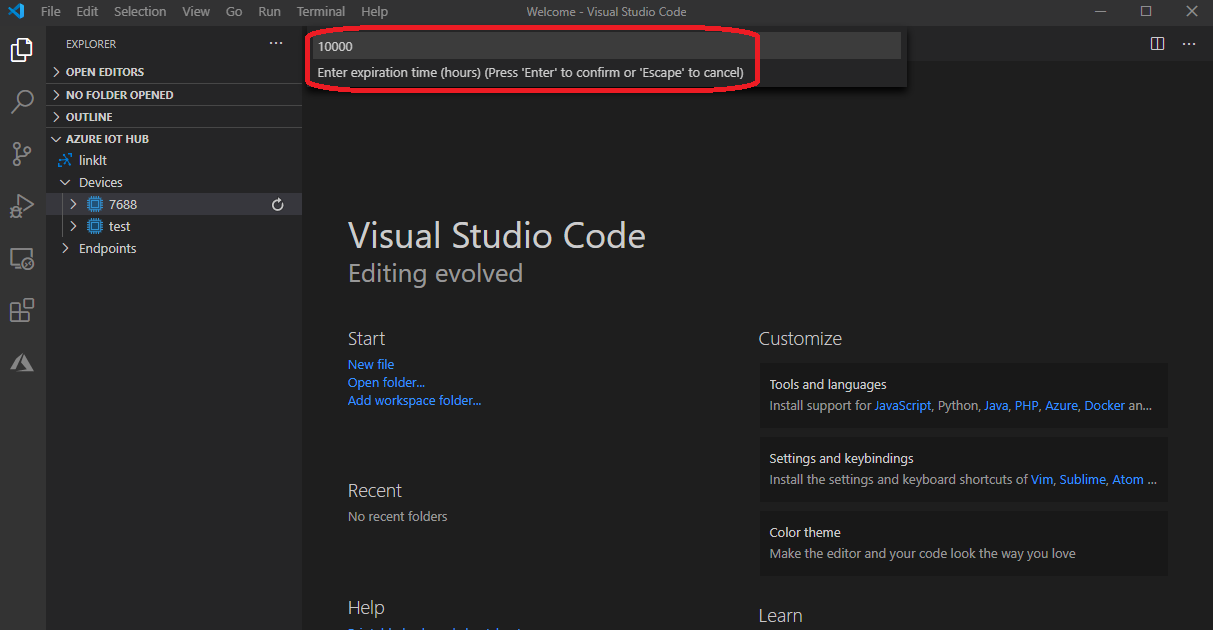
1. 獲取IoT device的SAS Token

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

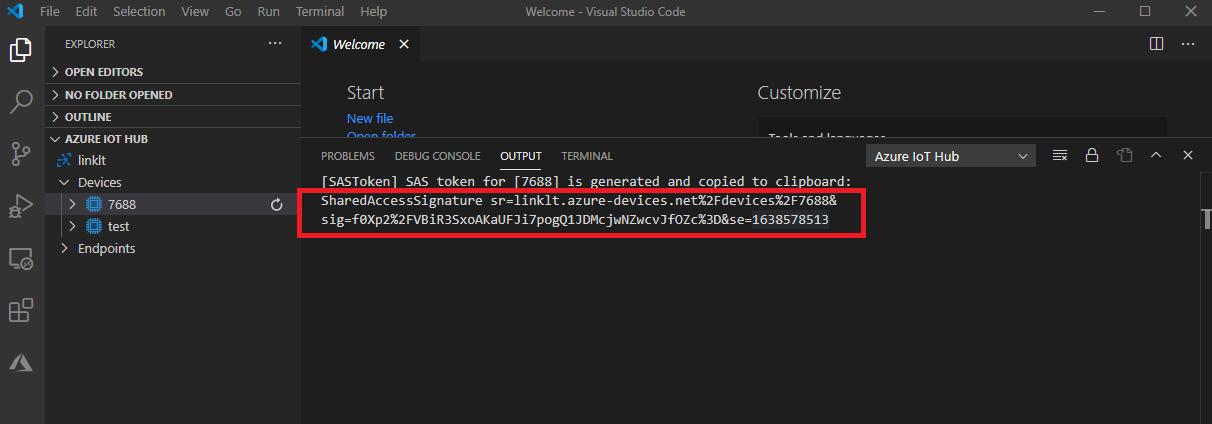


1. 輸入SAS Token使用時數

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)



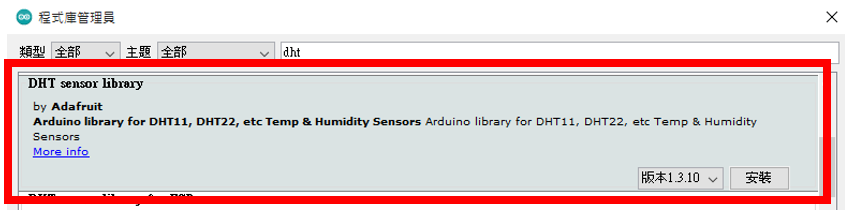
1. LinkIt 7688 Duo接DHT22

記得接上Micro USB讓它開機，並確認有連到同一個網路底下



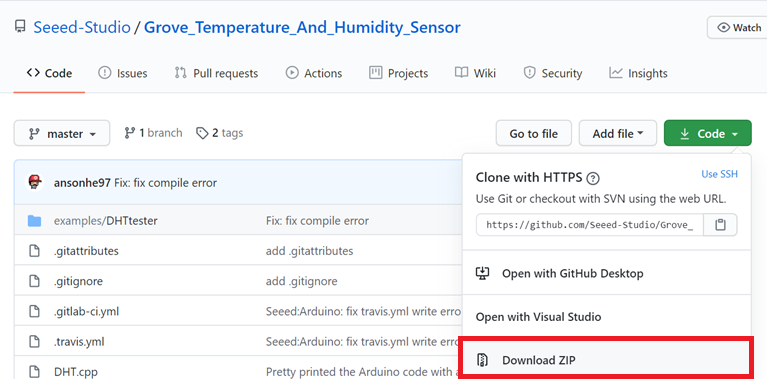
1. Arduino安裝程式庫

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。

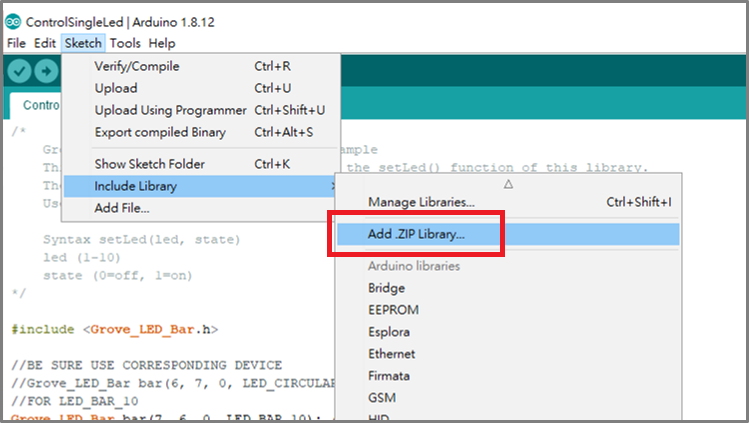


1. 下載Library

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

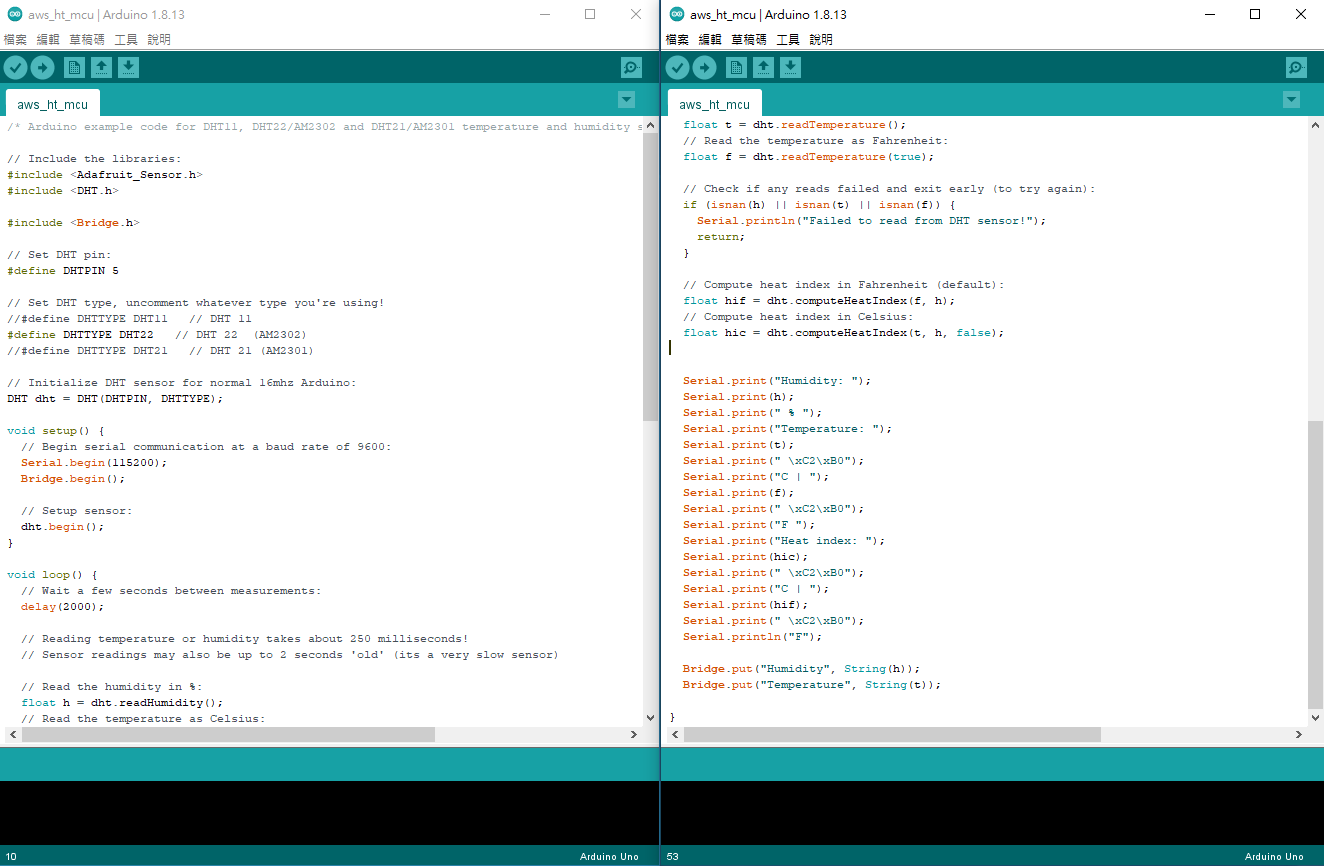
點擊File→New並輸入程式，接著點擊編譯並上傳。

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

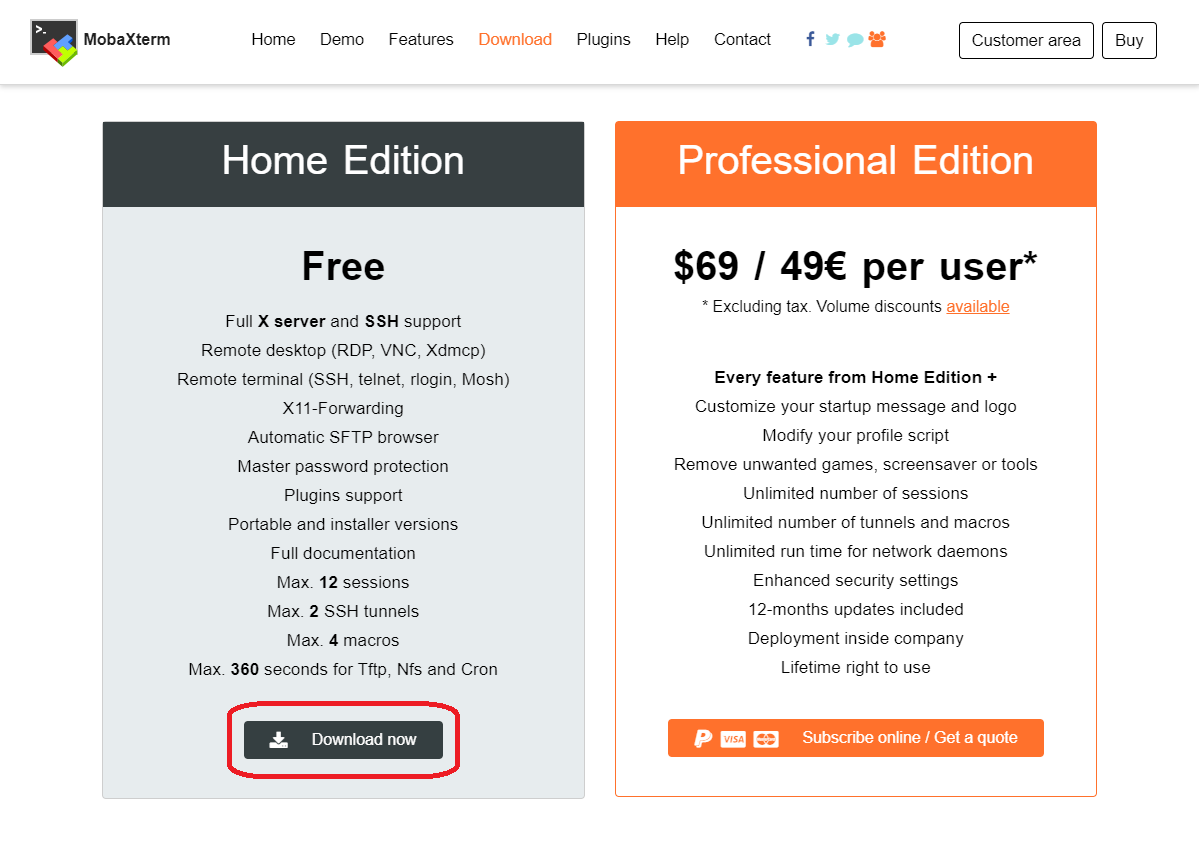
Serial.print(h);

Serial.print(t);)

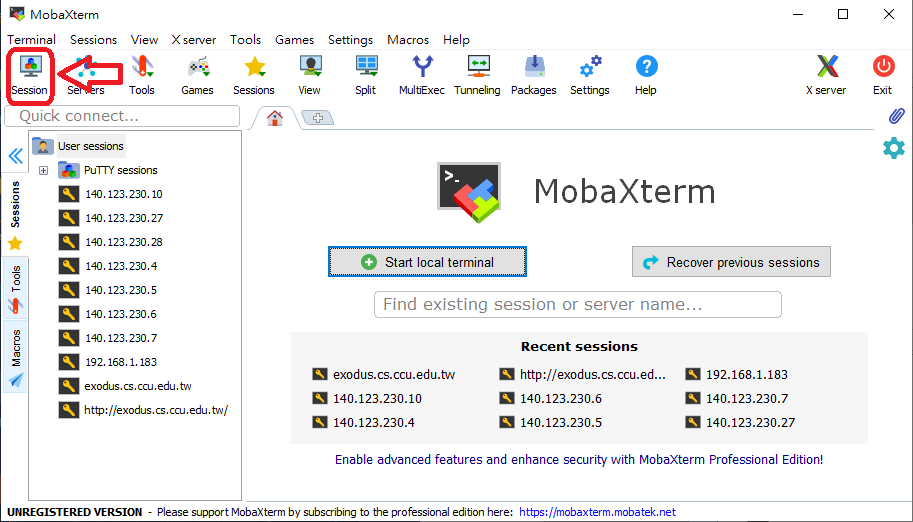


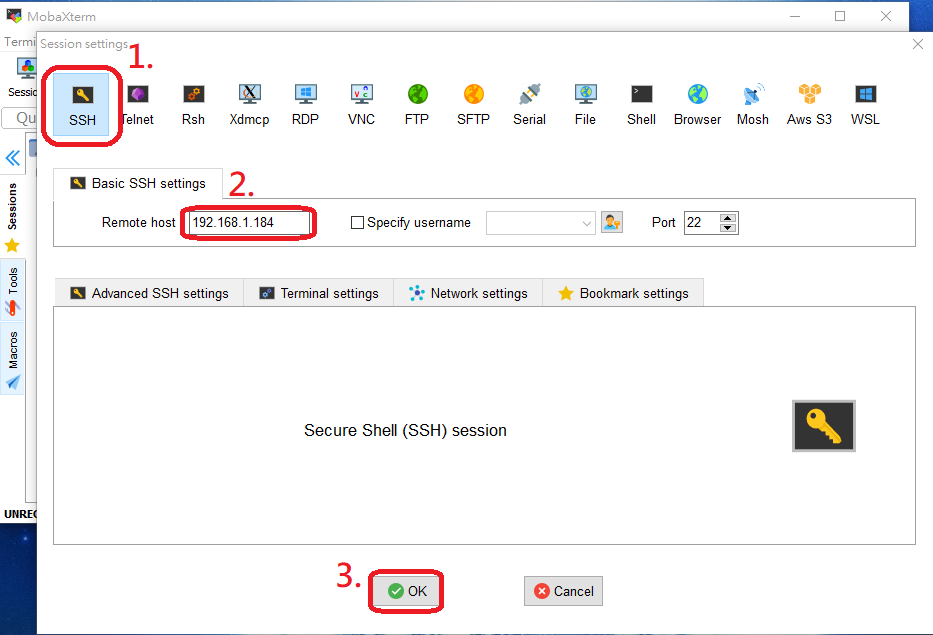
1. 安裝Mobaxterm

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



1. SSH連線

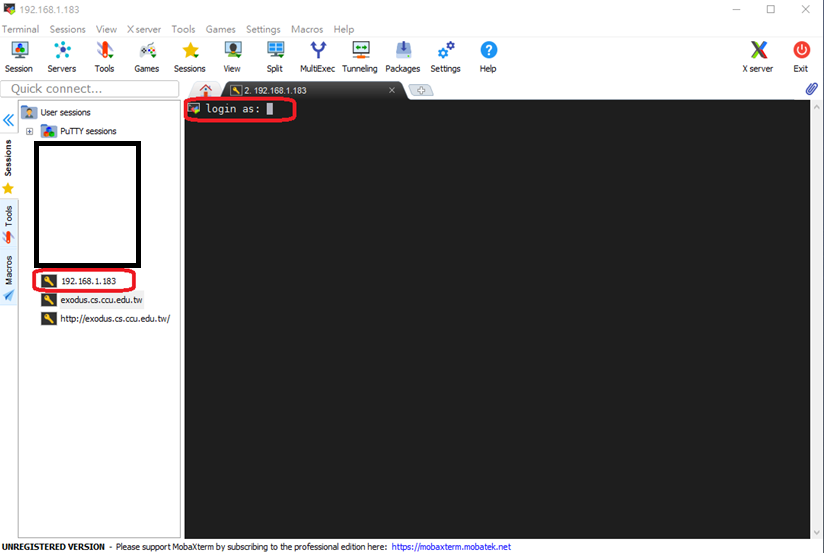




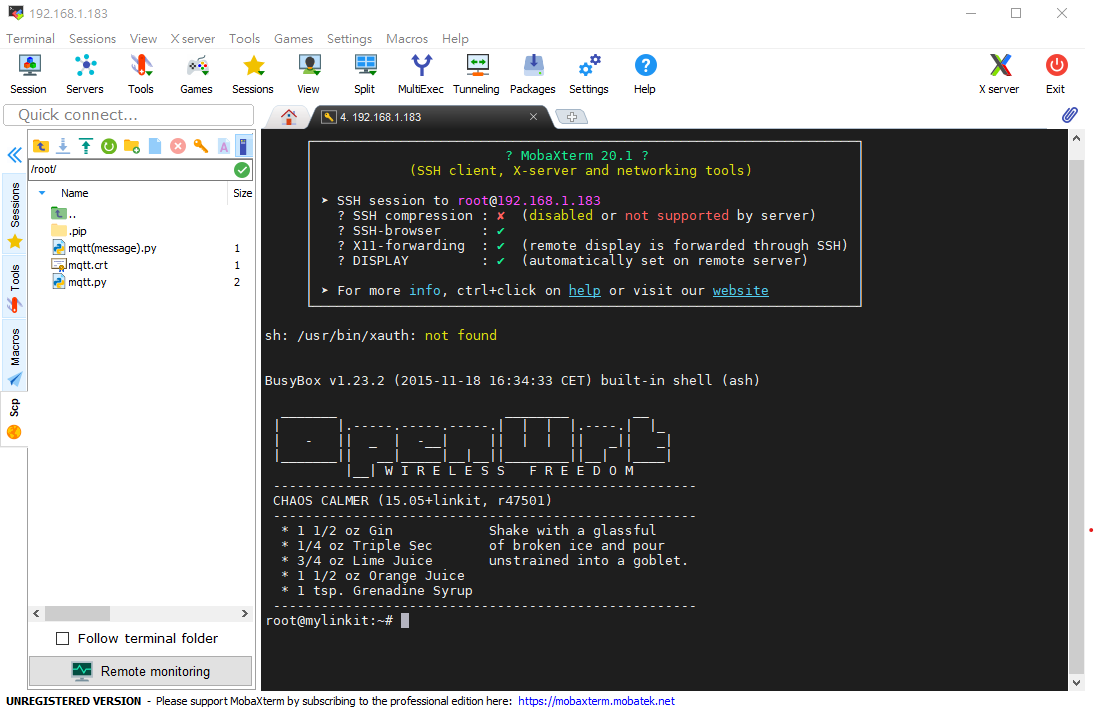
1. 成功連線，登入7688

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

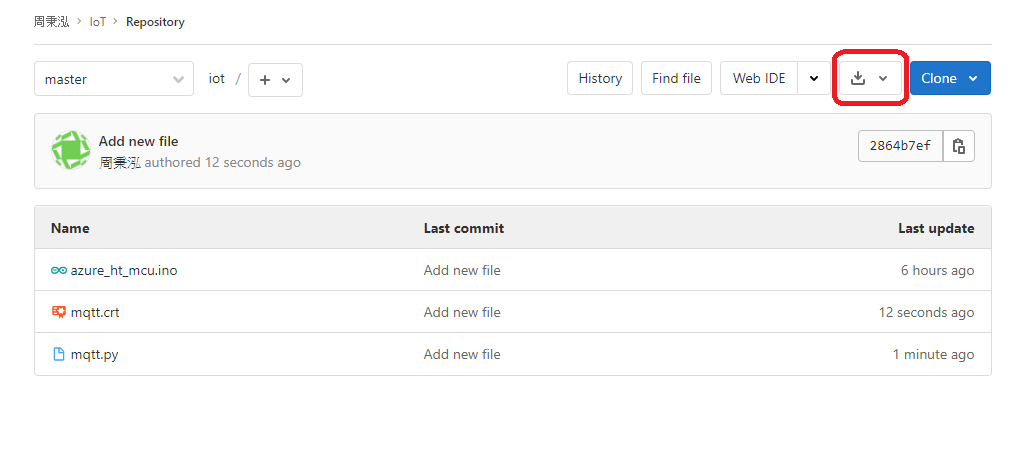
uci set yunbridge.config.disabled=0

uci commit

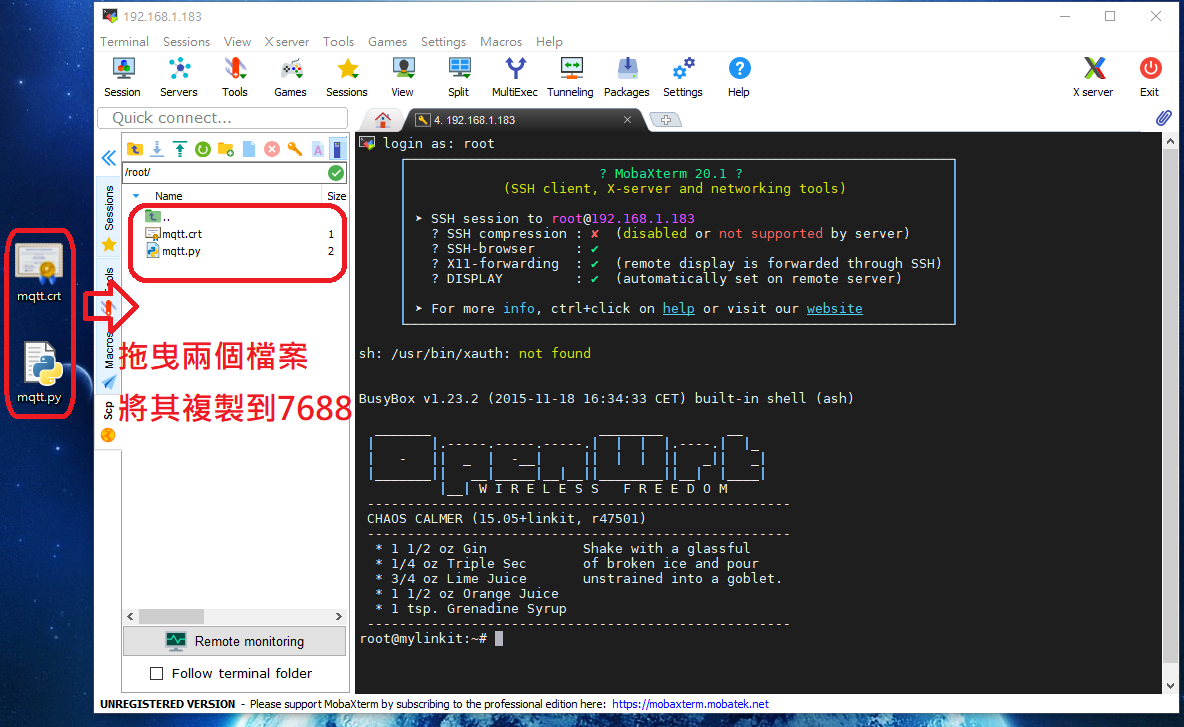
reboot

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

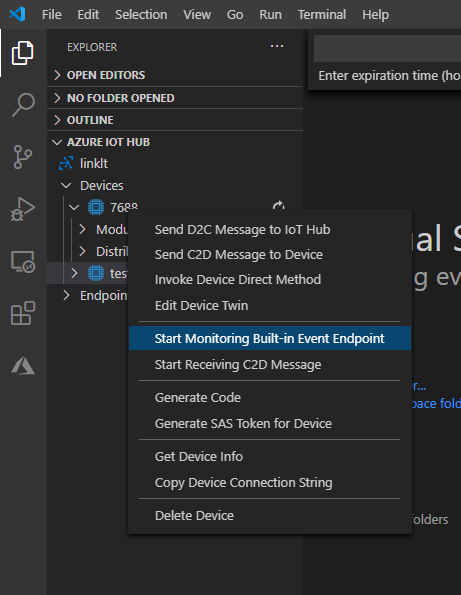
sas\_token = "SharedAccessSignature sr=linklt.azure-devices.net%2Fdevices%2F7688&sig=cF9yvJiFt0nErXoeDxbgzZ%2FANBdeQCbzdl7jaUyZADM%3D&se=1633243410"

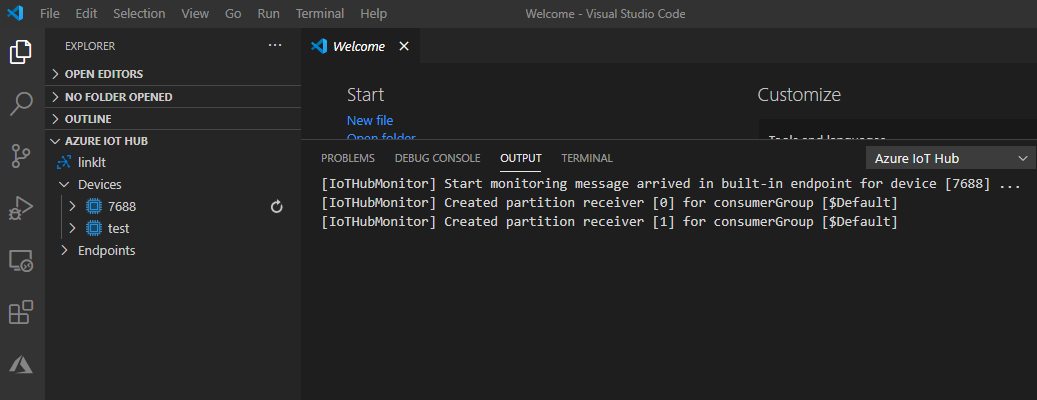
# IoT HUB名稱

iot\_hub\_name = "linklt"



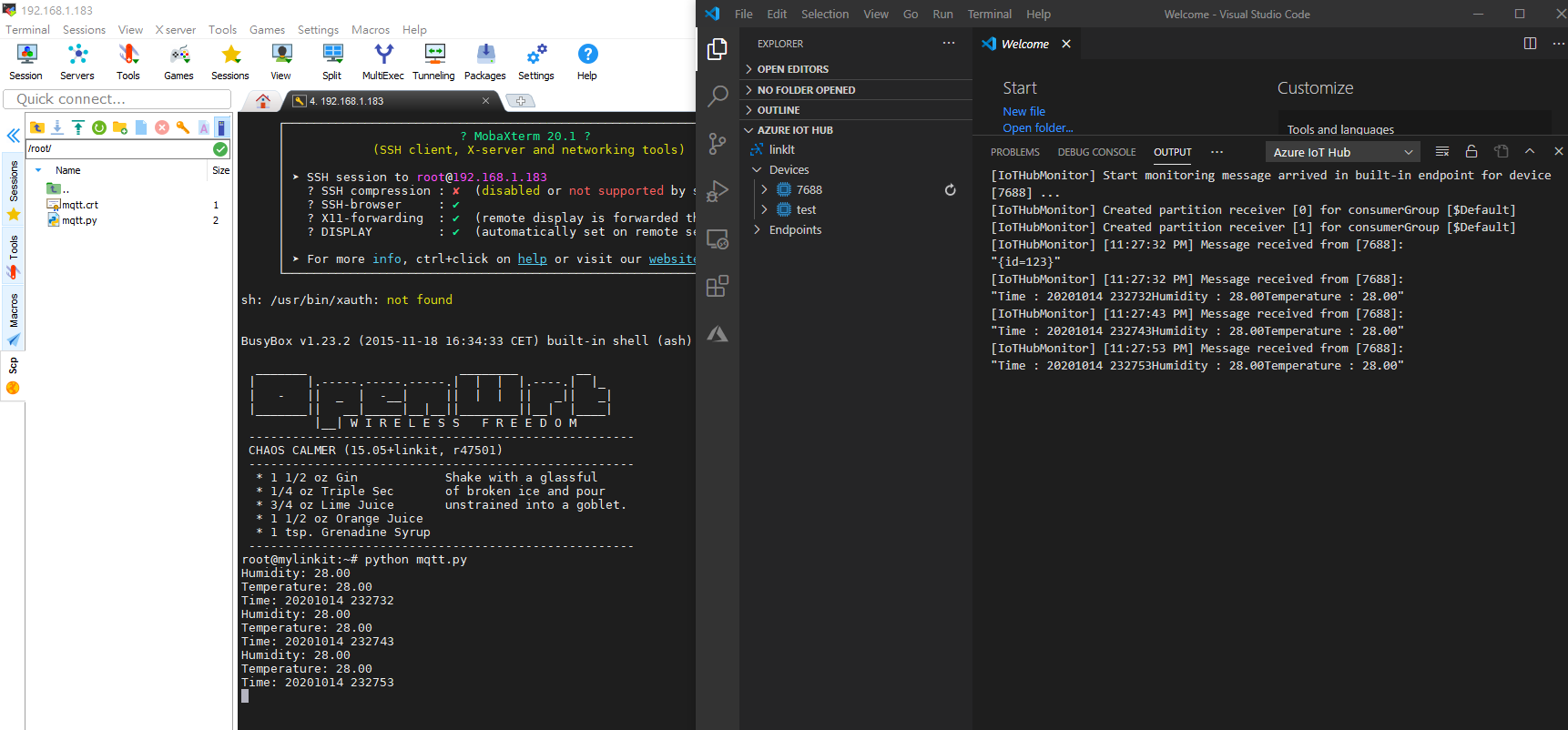
1. 開啟Visual Studio Code監控7688所上傳的資料



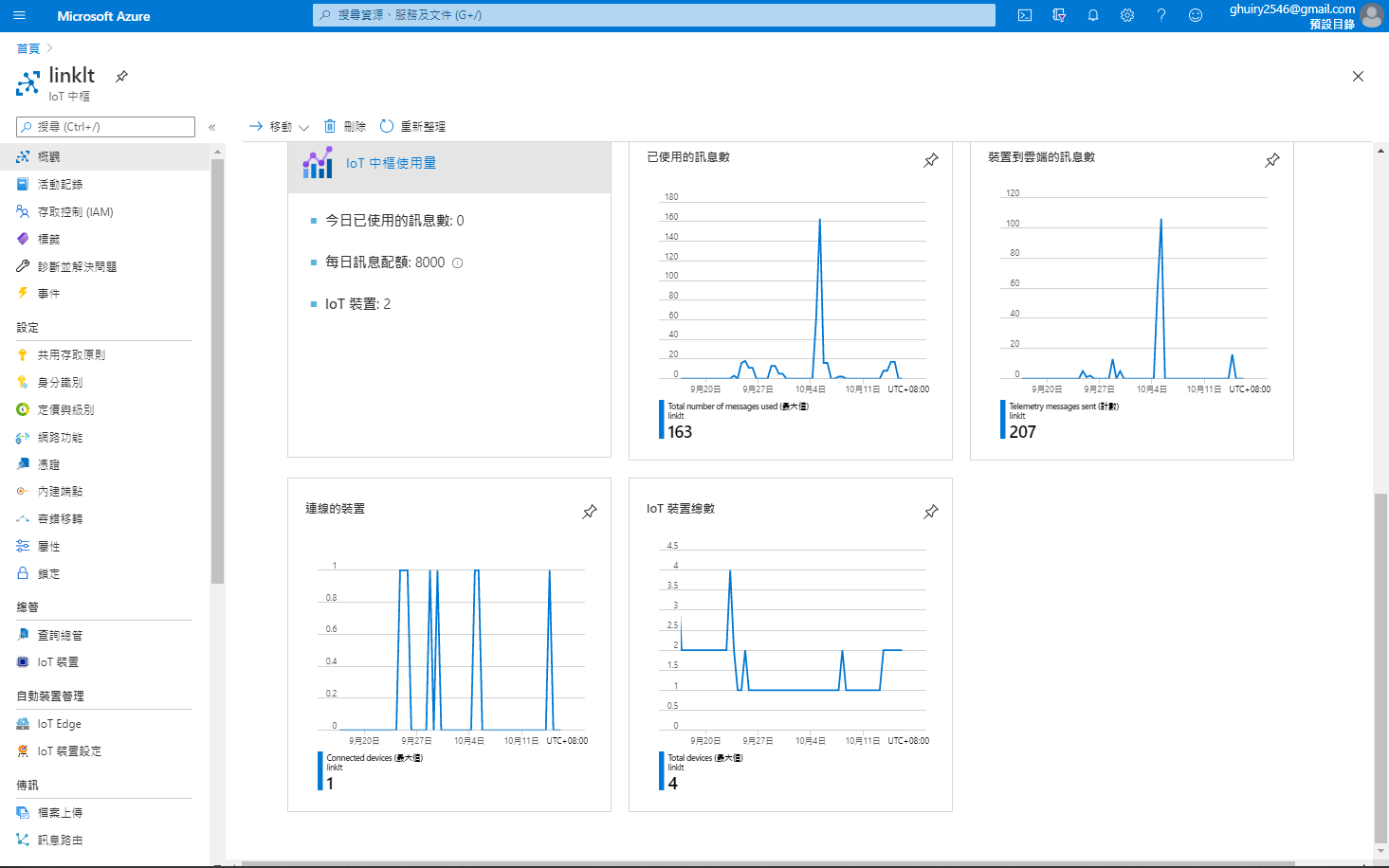


1. 執行mqtt.py

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)