**Lab3**

105072123黃海茵

* **MQTT運作流程**
* **角色**

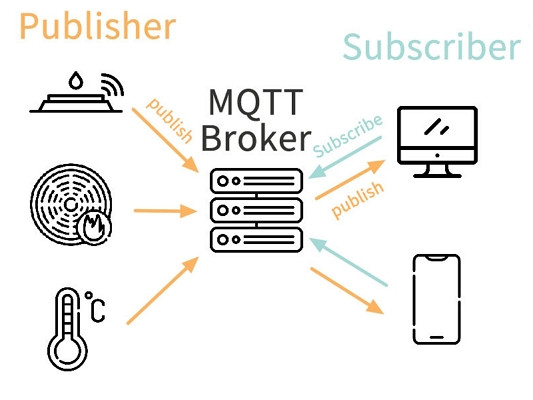
1. 分為Publisher、Subscriber、Broker三個角色
2. Publisher與Subscriber 之間有Broker當作中繼站，所以Publisher與 Subscriber不需要知道彼此的IP位址
3. Publisher與Subscriber之間透過Topic來取得需要的資訊

* **動作**

1. Publisher向Broker發送Publish
2. Subscriber向Broker發送Subscribe

* **運作方式**

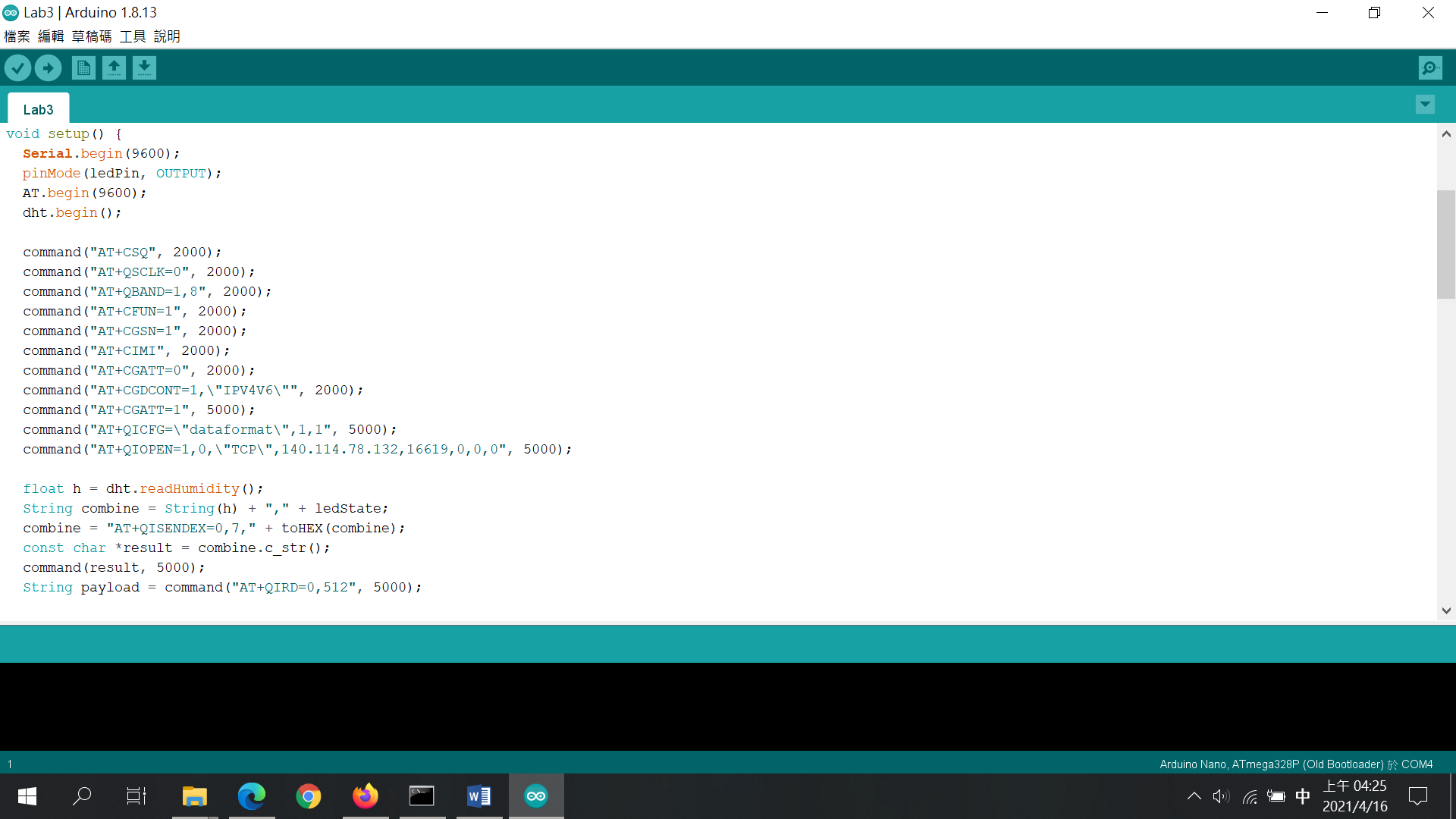
Subscriber會向Broker Subscribe特定的Topic，Broker會判斷 Publisher所發布的特定Topic是否有Subscriber訂閱此特定Topic，如有的話會將此特定Topic的訊息傳送給Subscriber



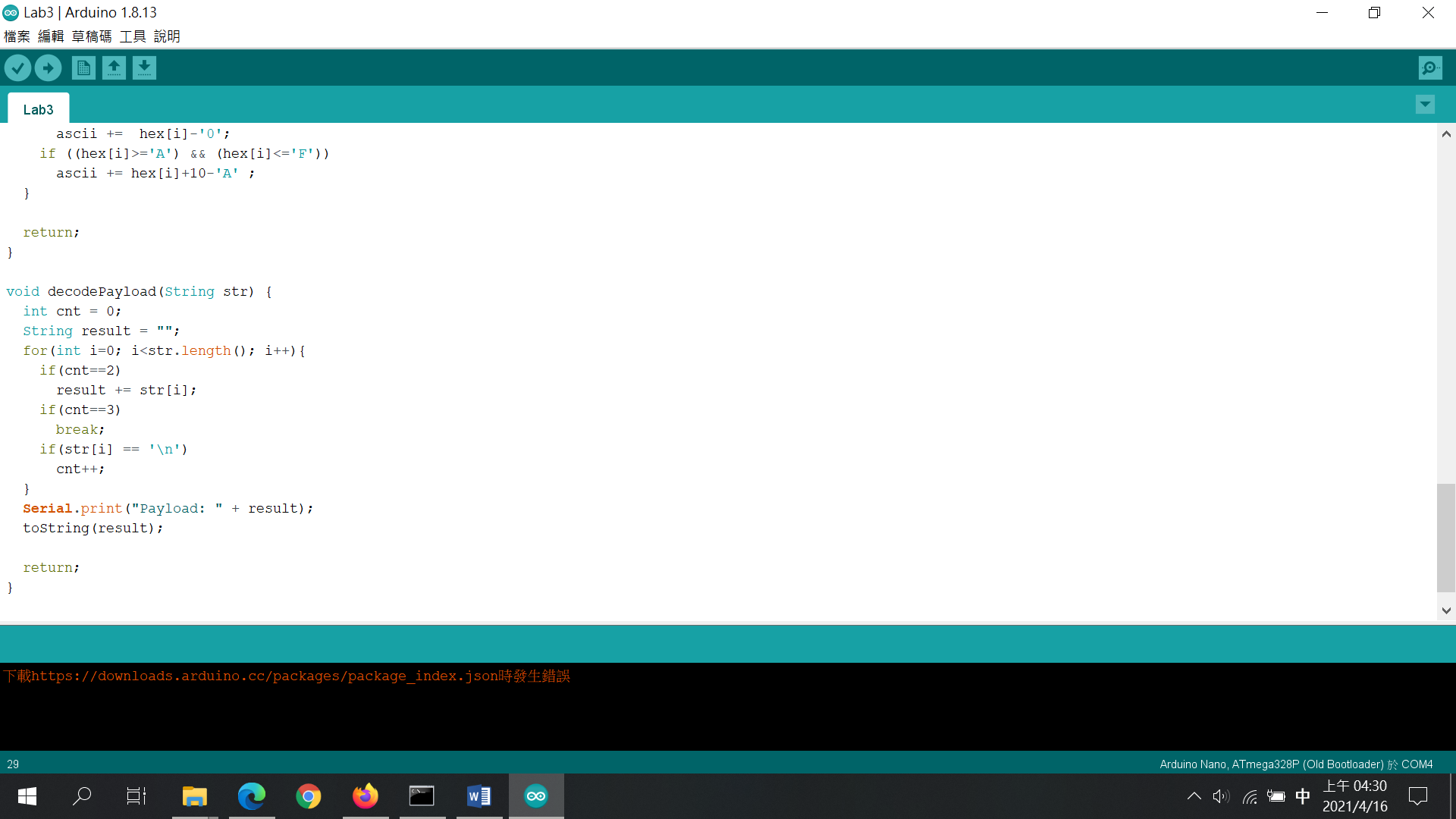
* **QoS參數**
* **0： 最多一次傳送（只負責傳送，發送過後就不管數據的傳送情況）**  
  當Publisher在發送訊息時，指定本次發送Qos為0，則Broker可能會沒收到訊息；而Broker若有接到訊息，發佈給Subscriber時，Subscriber也可能沒收到訊息。
* **1： 至少一次傳送（確認數據交付）**  
  當 Publisher 指定 Qos 為 1，Broker 收到訊息後會回傳 PUBACK Message給Publisher。Publisher在發佈訊息後，若沒收到回傳的 PUBACK Message，一定時間後會再重新傳一次相同內容的訊息給 Broker (不論 Broker 是否有到訊息)，也因此可能造成Subscriber 收到的訊息重複。
* **2： 正好一次傳送 (保證數據交付成功)**  
  當Publisher指定Qos為2，Broker為了避免發生Qos為1時，收到重複訊息的情況，Broker與Publisher會進行兩次的來回確認才將訊息傳遞下去。



* **程式流程**
* **Arduino**

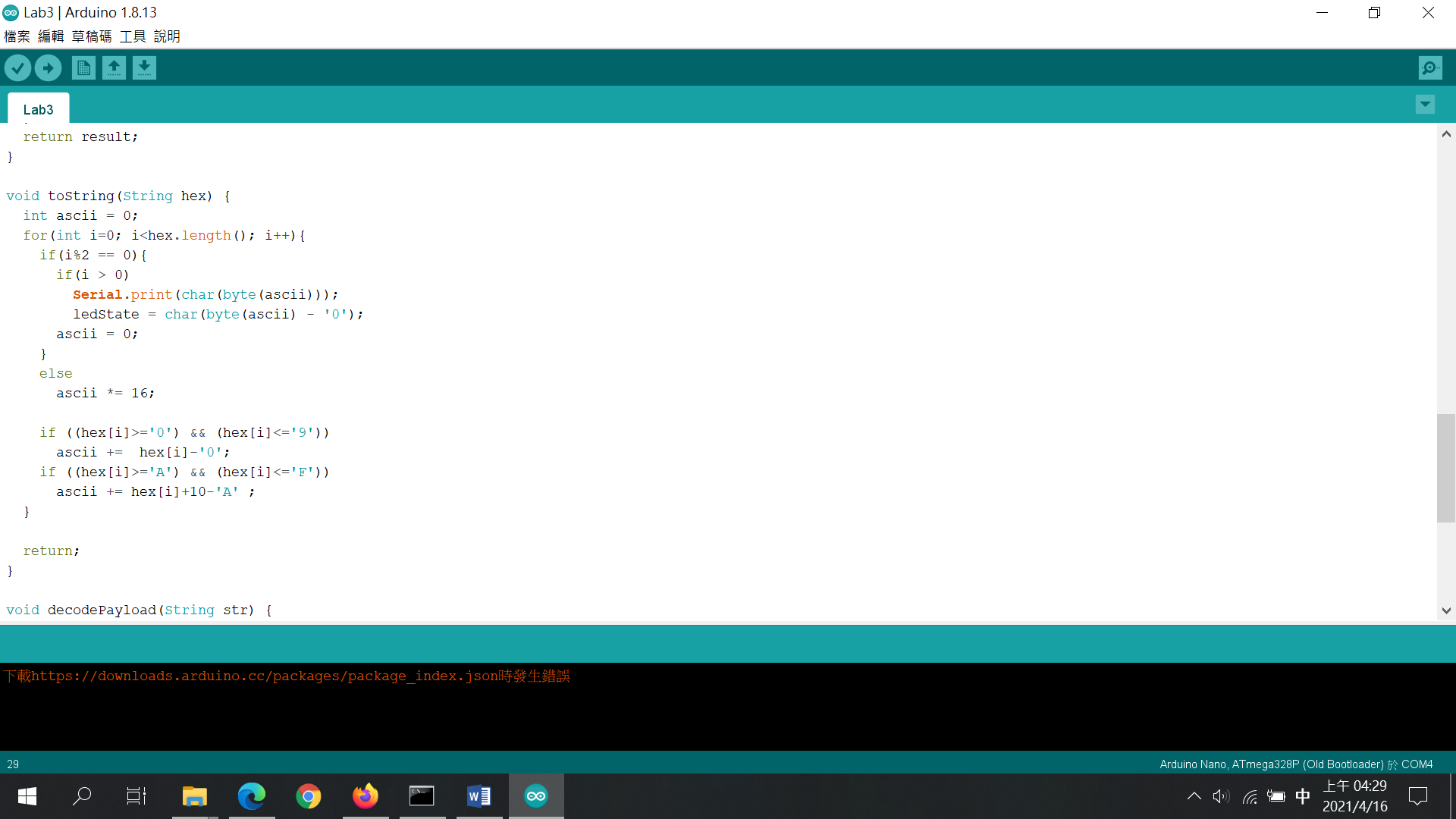


首先是和server建立連線，並且傳濕度和led state過去，這部分和上次lab大同小異，就不贅述。



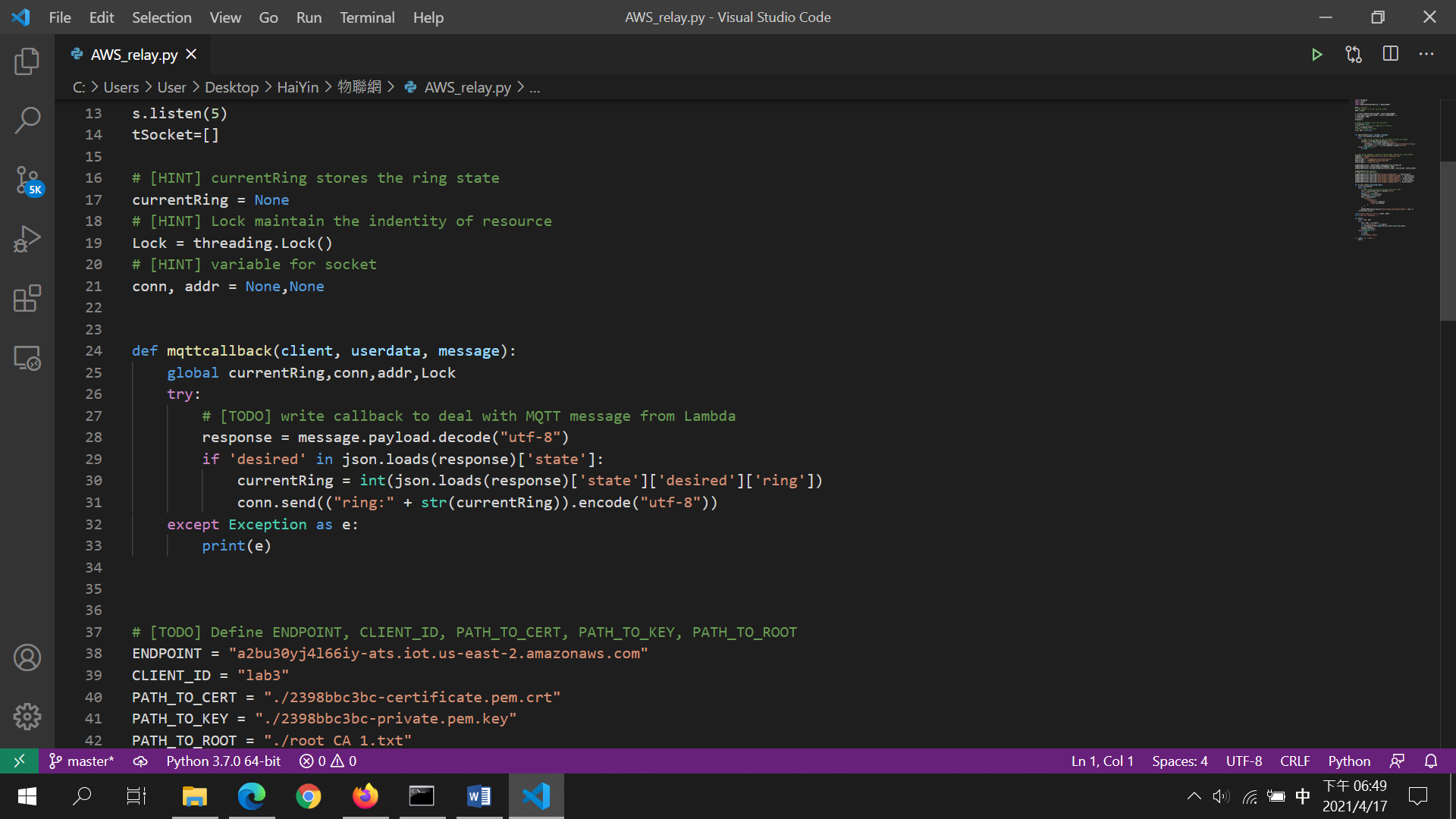
這次主要就是多寫了兩個function，decodePayload和toString。

decodePayload用來把server回傳的payload擷取出需要的部分，然後丟進toString中轉成字元。

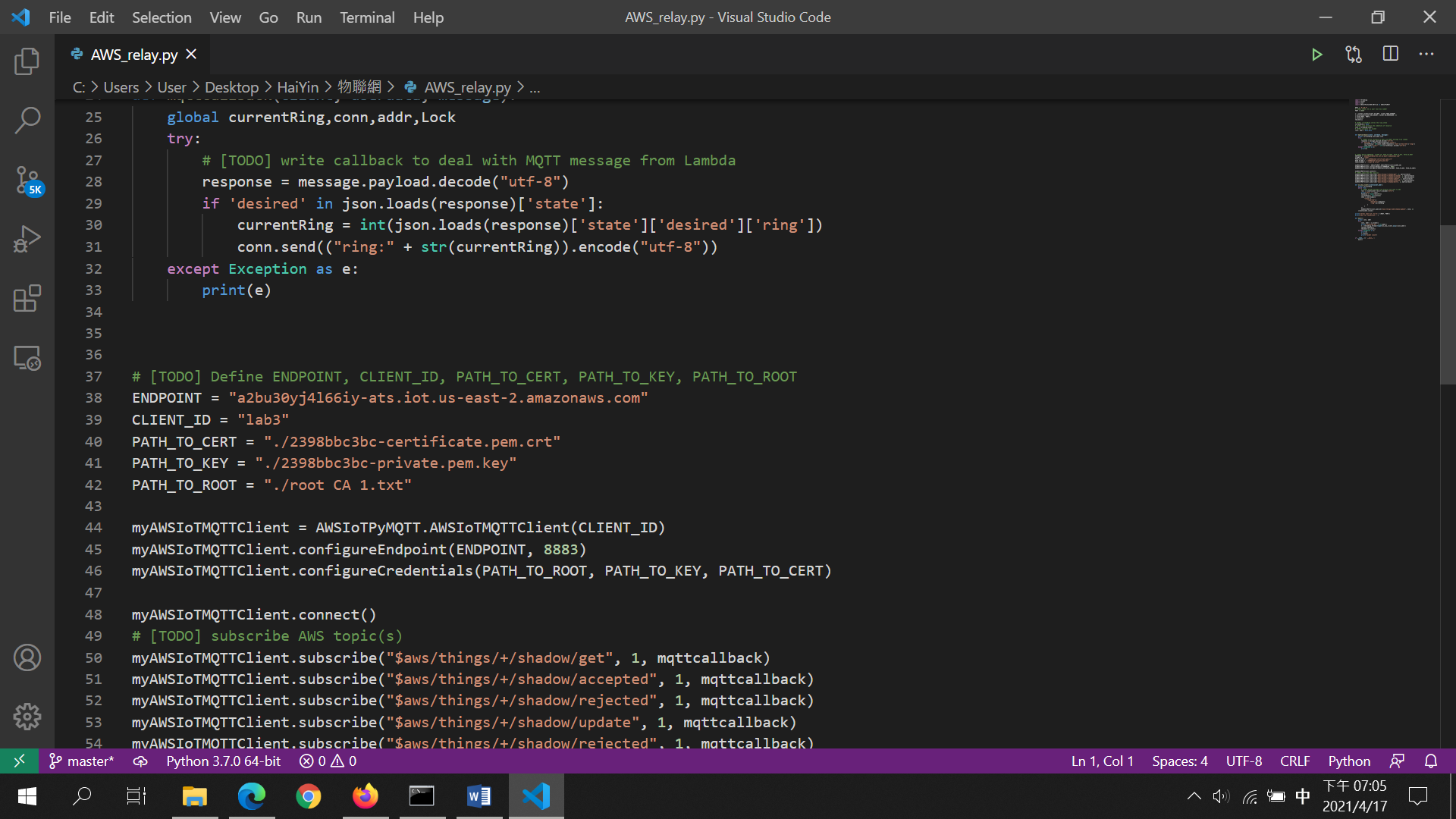


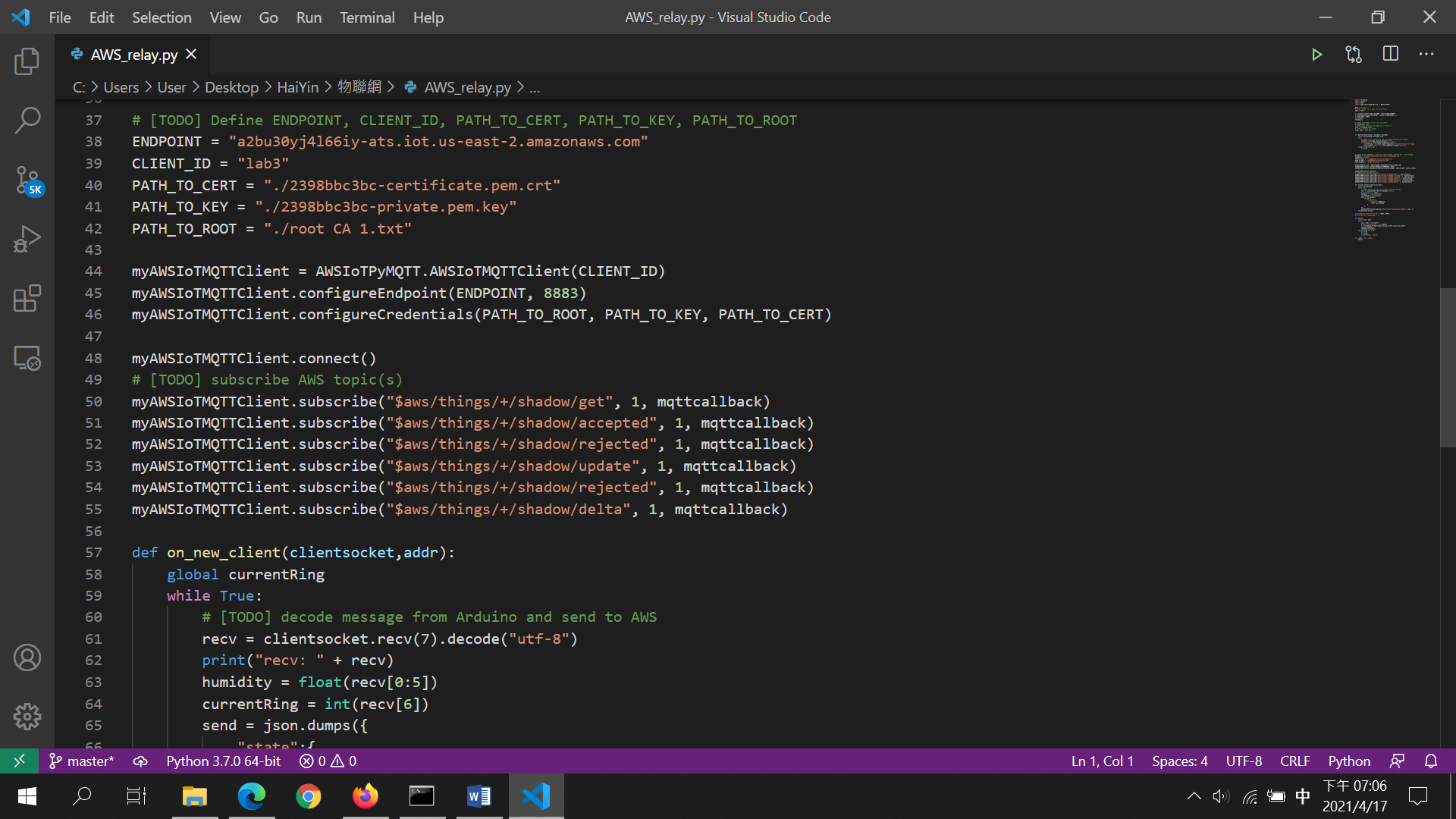
toString用來把十六進位的ASCII碼轉回字元，再存到ledState中。

* **server**

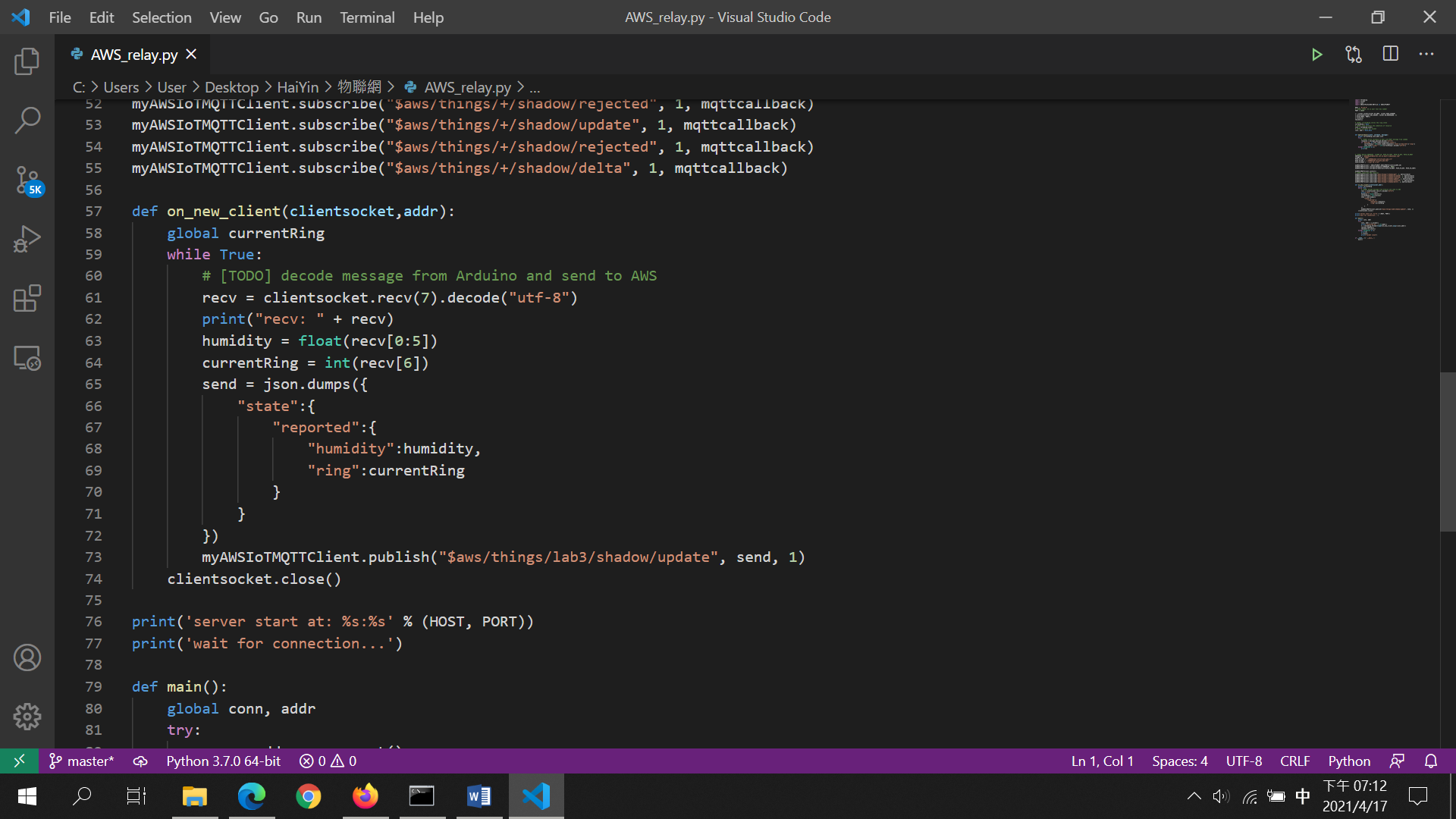


這個function實作server收到AWS的狀態，decode之後，取出字串中有用的部分，再encode並轉傳到BC20。





這兩部分把原本直接在MQTT中手動輸入的資料，自動執行。



這個function實作server收到arduino傳來的資料，decode後轉成正確的格式，再送到AWS。

* **遇到困難及解決方式**

我覺得最近這兩次的lab，最麻煩也弄比較久的部分都是在字串轉換、處理，每次想要嘗試找一些內建的東西來做，然後都找不到，最後都直接自己寫一個function，用最原始的方式跑，不知道有沒有更好的方法（至今還沒找到 ☹）

* **操作截圖**

