How to run Auto Verification Tool

# Install python3 and required python libraries

AVT是用python 3編寫的腳本，請先在你的Mac上面安裝python 3

安裝檔下載：<https://www.python.org/downloads/>

可執行下面指令，查看Mac上面目前的python版本：

$ python --version

Python 3.6.0 :: Anaconda 4.3.1 (x86\_64)

安裝後，在終端機執行 $ python3 如果能夠進入python編程環境，就是安裝成功了。

$ python3

Python 3.6.0 |Anaconda 4.3.1 (x86\_64)| (default, Dec 23 2016, 13:19:00)

[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>>

然後需要安裝avt.py腳本裡面有使用到的python套件，在終端機執行pip3即可安裝！

下面是需要先被安裝的python套件（有些好像是python已經內建的）：

$ pip3 install argparse

$ pip3 install openpyxl

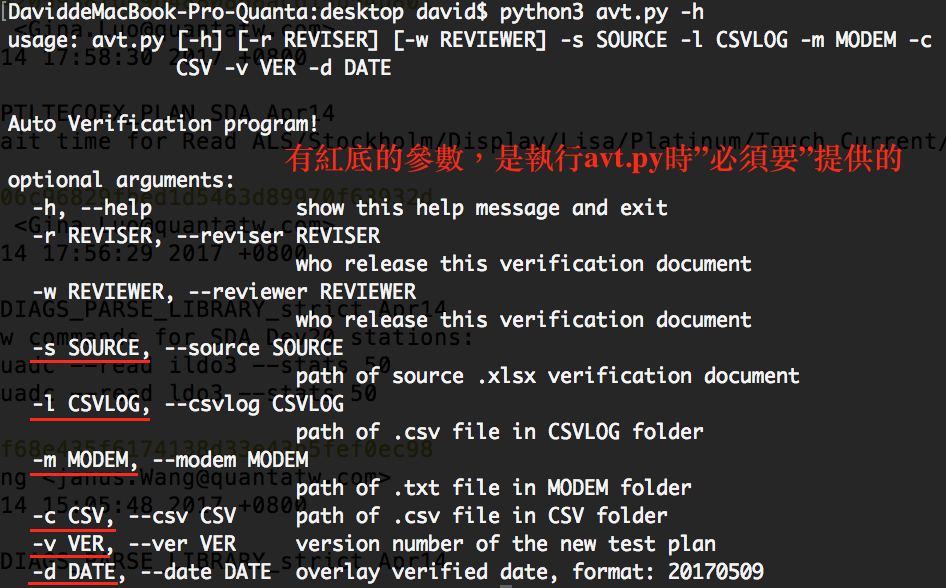
$ pip3 install csv

$ pip3 install string

$ pip3 install getpass

以上所需套件全部安裝完成之後，就可以打開終端機，切換到你放置avt.py這個腳本的路徑（例如~/desktop），然後執行下面的指令，可以瀏覽參數的使用方式了：

$ python3 avt.py -h



avt.py的運作方式很簡單，基本上就是 **–s**後面跟一個目前版本的verification document（Excel文件），然後 **–l** 後面跟一個CSVLOG文件，然後 **–m** 後面跟一個 MODEM LOG文件，然後 **–c** 後面跟一個CSV文件，-v後面接這個新版本的TestPlan版號（例如新產生的Overlay Version Name是QT0\_ChargingEfficiency20170509ver**15.0**\_JHAB\_013，那麽 **–v** 後面就給**15.0**，腳本會幫你將這個版號生成到新的Excel文件中，同時也會自動幫你將QTM的版號加1，也就是將JHAB\_**013**更新為JHAB\_**014**），最後 **–d** 後面接verify date（格式為8個數字，如20170509）。指令裡面，給參數的順序沒有固定，但參數後面必須給對指定的參數值。

另外 **–r**（--reviser）這個參數可以指定Reviser的名稱（若沒給這個參數，預設會從系統讀取你的username名稱，然後放到新產生的Excel文件裡面），而 **–w** （--reviewer）這個參數，也是提供指定名稱的功能，若沒給這個參數，Excel文件裡面，預設的Reviewer會是Doris。

下面是基本的avt.py使用範例，在終端機執行之後，腳本就會幫你自動產生新的Excel文件了（ps.如果覺得單槓的參數不好記，也可以改用雙槓參數，例如用 --source [值]，取代 –s [值]）

$ **python3 avt.py** **–s** [目前的Excel文件路徑] **-l** [csvlog文件路徑] **-m** [modemlog文件路徑] **-c** [csv文件路徑] **–v** [新的TestPlan版號] **–d** [Verify Date]



**說明：為何執行avt.py需要載入目前的verification document（Excel文件）呢？**

Excel文件裡面的“**CSV log comparison**”和“**UART Log Check**”兩個worksheet中，一般都有左右兩個區塊，右邊區塊裡面的內容是上一版本的測試資料。如果這個腳本沒有沒有載入上一版本的Excel文件，要無中生有一個新的（目前版本）Excel文件，就必須另外帶入前一個驗證版本的CSVLOG和MODEM LOG文件，這樣反而更麻煩（因為你電腦裡面必須保留上個版本的這些測試文件，且執行腳本時還要帶入一堆參數），所以我認為比較好的做法，是將上個版本的Excel文件，作為”樣板”導入到腳本中，然後產生新的Excel文件。

這樣做很大的好處是，每個worksheet裡面的所有欄位版型、欄位字型、背景色等內容，可以從目前的Excel直接套用，而用再寫程式去無中生有產生這些東西。不過在執行腳本之前，你目前的Excel文件裡面，有些欄位設定還是需要遵從一定的標準，不然執行時可能會出錯。

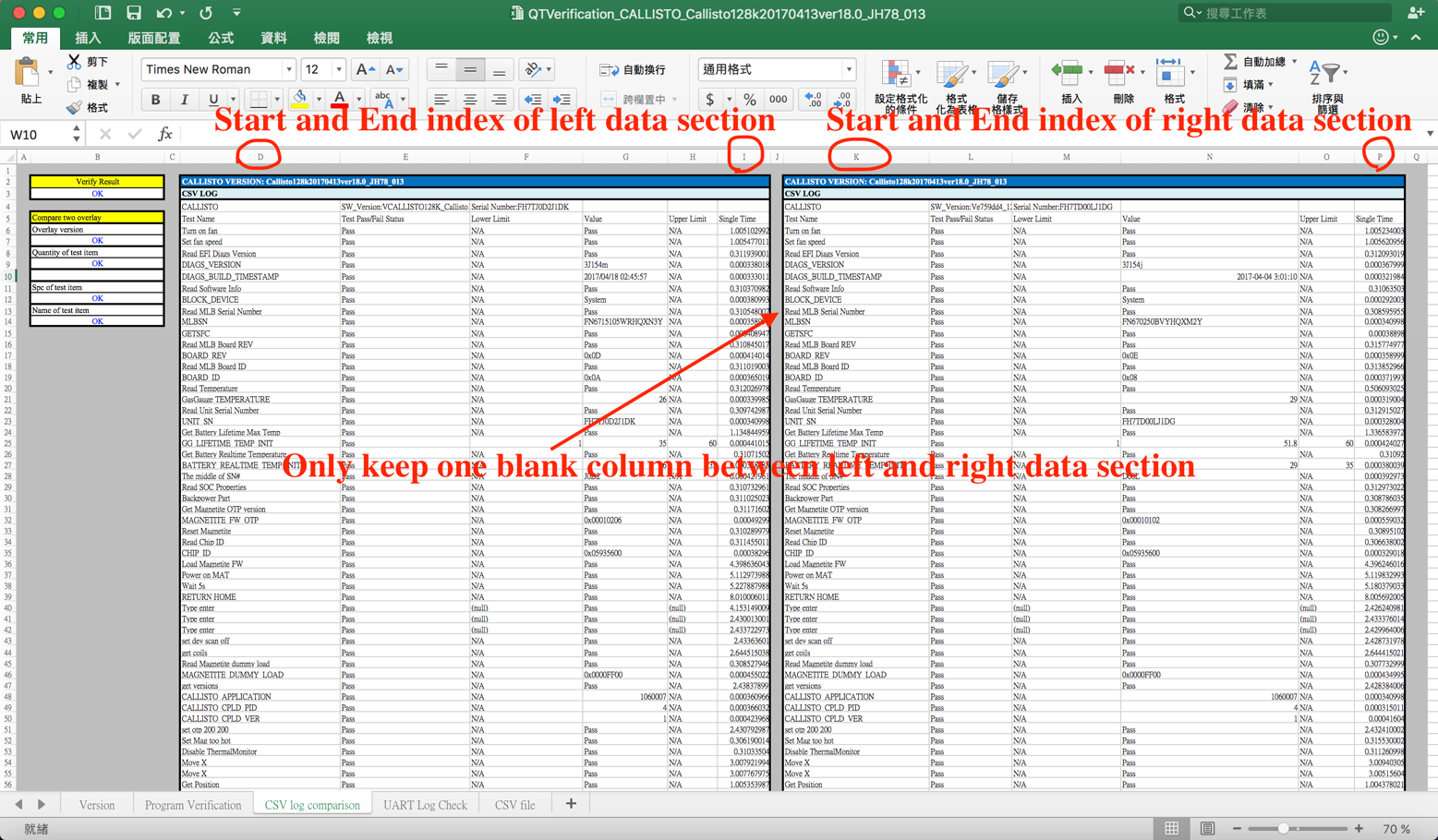
# Preparation of current Excel file

基本上avt.py腳本是根據現有的 Verification document（Excel）文件裡面的版型和設定來進行資料處理，所以基本上你不需要對現有的Excel進行什麼變動，只是有些標準需要遵守，下面為您說明執行腳本前，現有的Excel文件內容有哪些需要遵守的部分：

Excel文件的**Version、Program Verification**和**CSV file**這3個worksheet的欄位，基本上都不需要做什麼調整，因為表格的位子和欄位數量基本上是固定的。Excel文件裡需要調整的部分，應該只有**CSV log comparison**和**UART LogCheck**這2個worksheet而已。

在**CSV log comparison**這個worksheet中，因為CSVLOG.csv文件的表格欄數，基本上是不變的，所以不用做什麼變動，但是要請注意不要任意增加空白欄位。因為目前我是直接硬寫，用D欄位當作左側data cell section的起點、I欄位當作左側cell section的終點（右側data cell section的判斷方式亦同，是用K和P欄位當作起點和終點的索引），所以如果你任意在C、D或J、K欄位中間插入新的空白欄位，就會導致腳本執行時出錯。

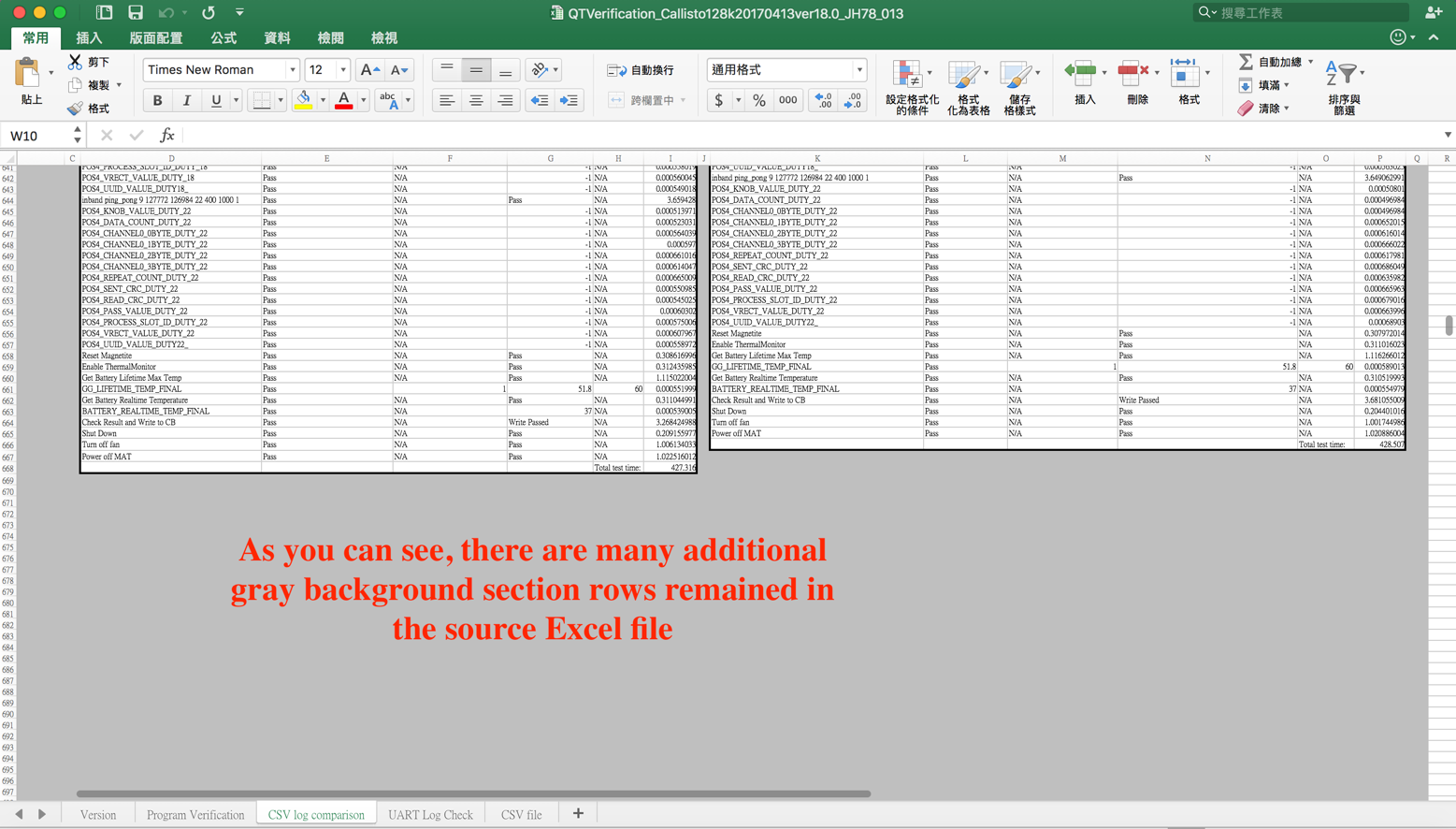
因為目前CSVLOG.csv的格式是固定的，所以目前這種方法是可以正常work的。如果日後CSVLOG.csv的欄位數量會變動，之後也可以將腳本改為自動去判斷左右兩個data cell section的起點和終點欄位索引。但目前正常可以work，就暫時先不改了。



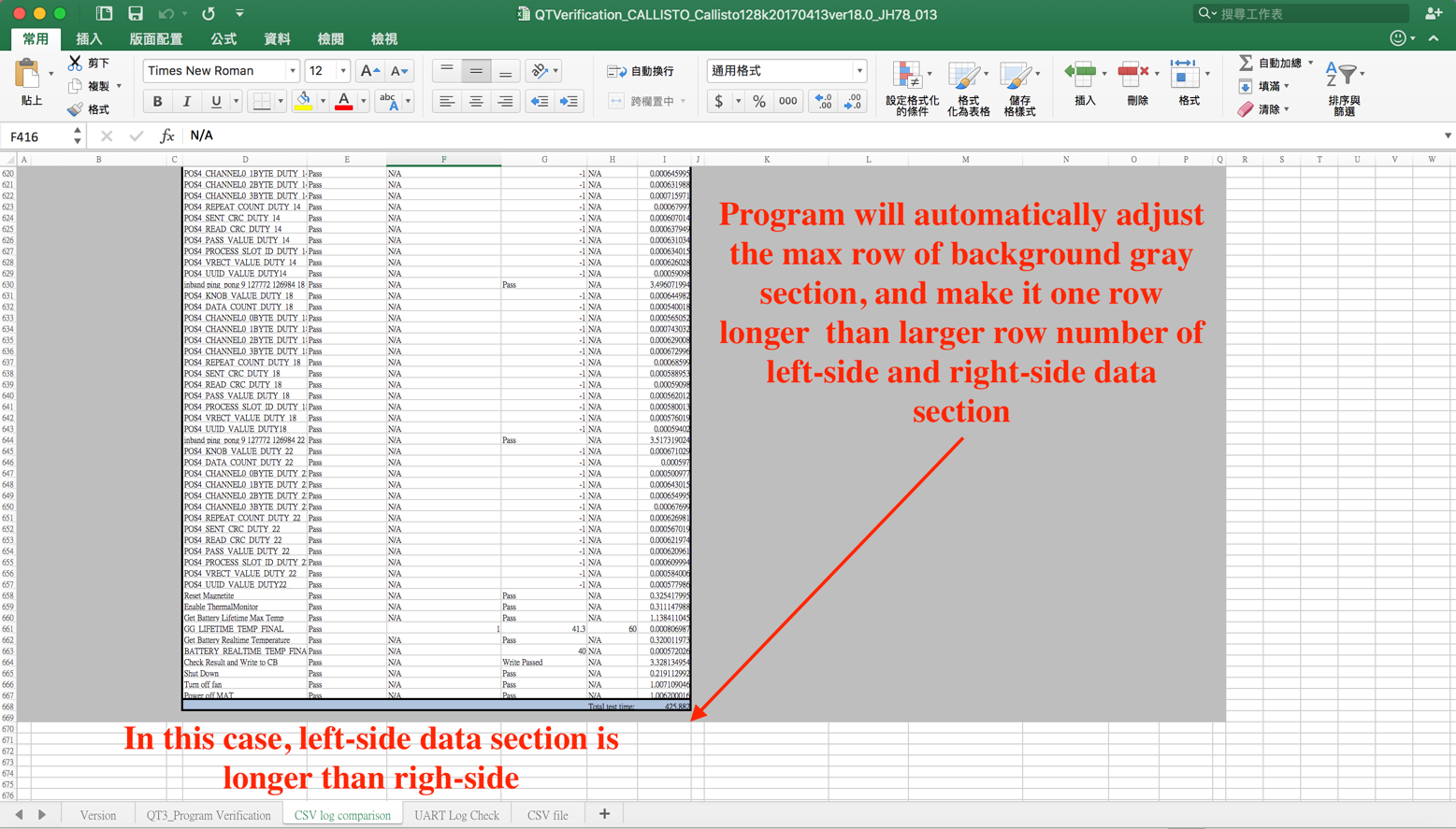
然後灰色背景高度的部分，你不需要特地去做變動，avt.py腳本會自動幫你將處理資料後，多餘的灰色背景部分刪掉，刪到剩下比左、右側資料欄位高度較高的那一邊，高出一個row的高度。例如資料處理之後，左側資料欄位有30個row，右側資料欄位有25個row，那麼最後灰色背景的高度就會是到31個row。

Ps. 目前這個腳本是用CSVLOG文件裡面的**Total test time**字串，來判斷這份CSVLOG的row數量，如果CSVLOG文件裡面沒有這個字串，執行腳本時就會出錯。不確定是不是每個station產生的CSVLOG裡面都有這個字串，如果有例外情況請通知我，我會再改寫程式，用其他方式來判斷CSVLOG文件的最大row數量。

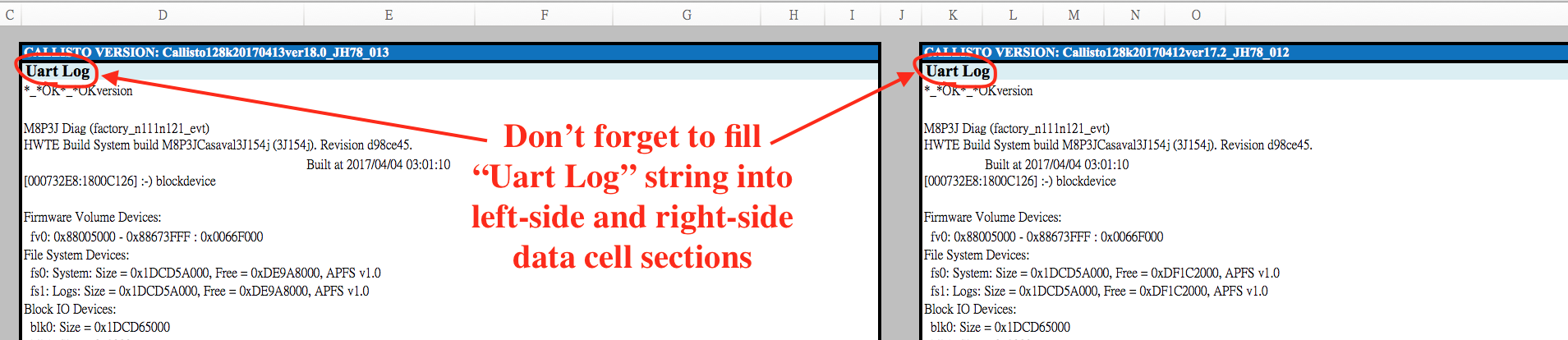
下圖是一個Excel範例，原始的Excel裡面，有很多忘了被刪掉的多餘灰色背景區塊：



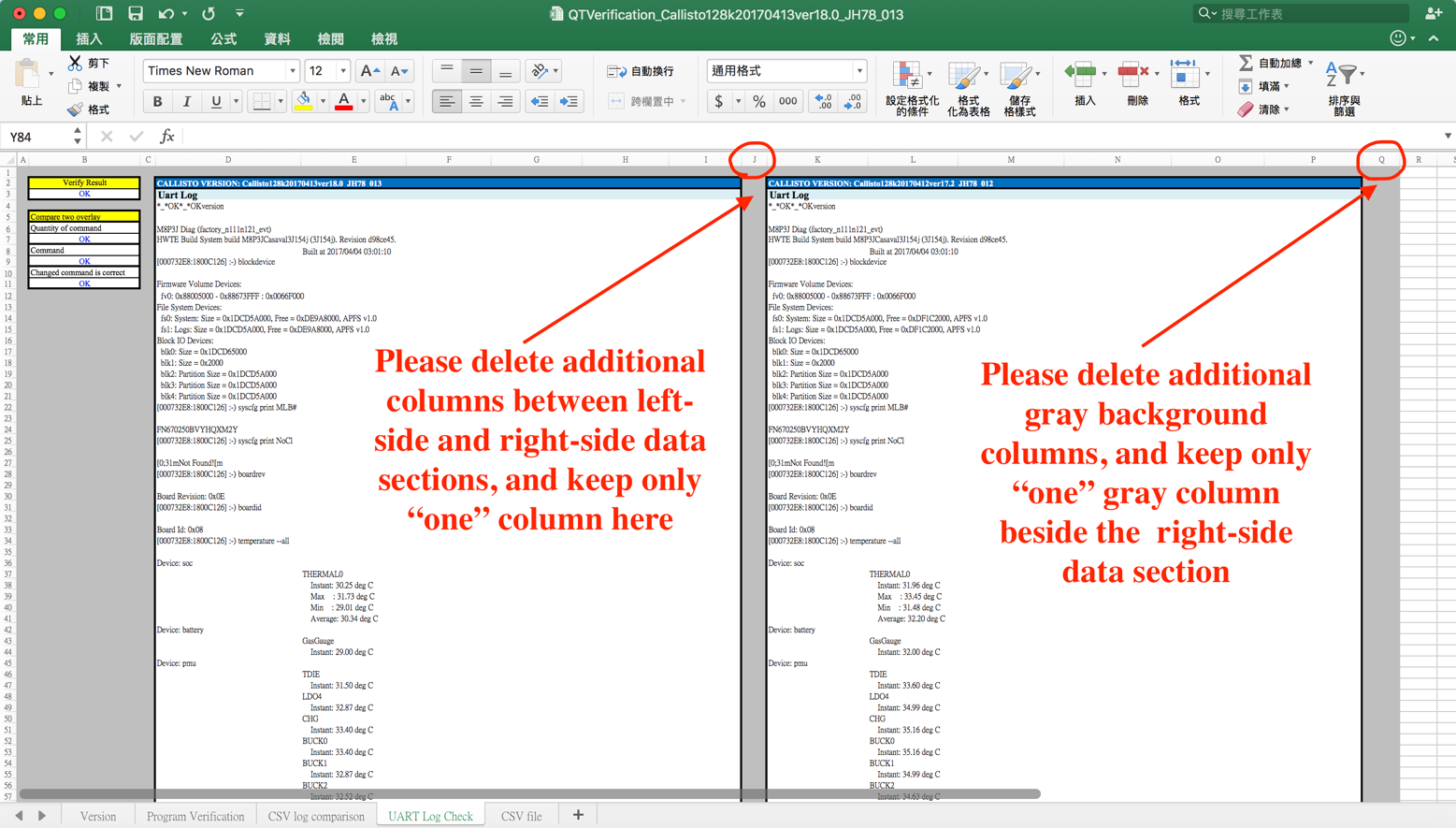
最後產生的新的Excel文件，灰色背景的高度設定會如下圖所示：



接下來是**UART Log Check**這個worksheet的部分，首先請確認左右兩側的data section裡面，都有**Uart Log**這個字串，因為目前的腳本是用這個字串，來偵測右側data cell section的起始欄位索引，如果你把這個字串從右側data cell section裡面刪掉，腳本處理時就會發生錯誤。

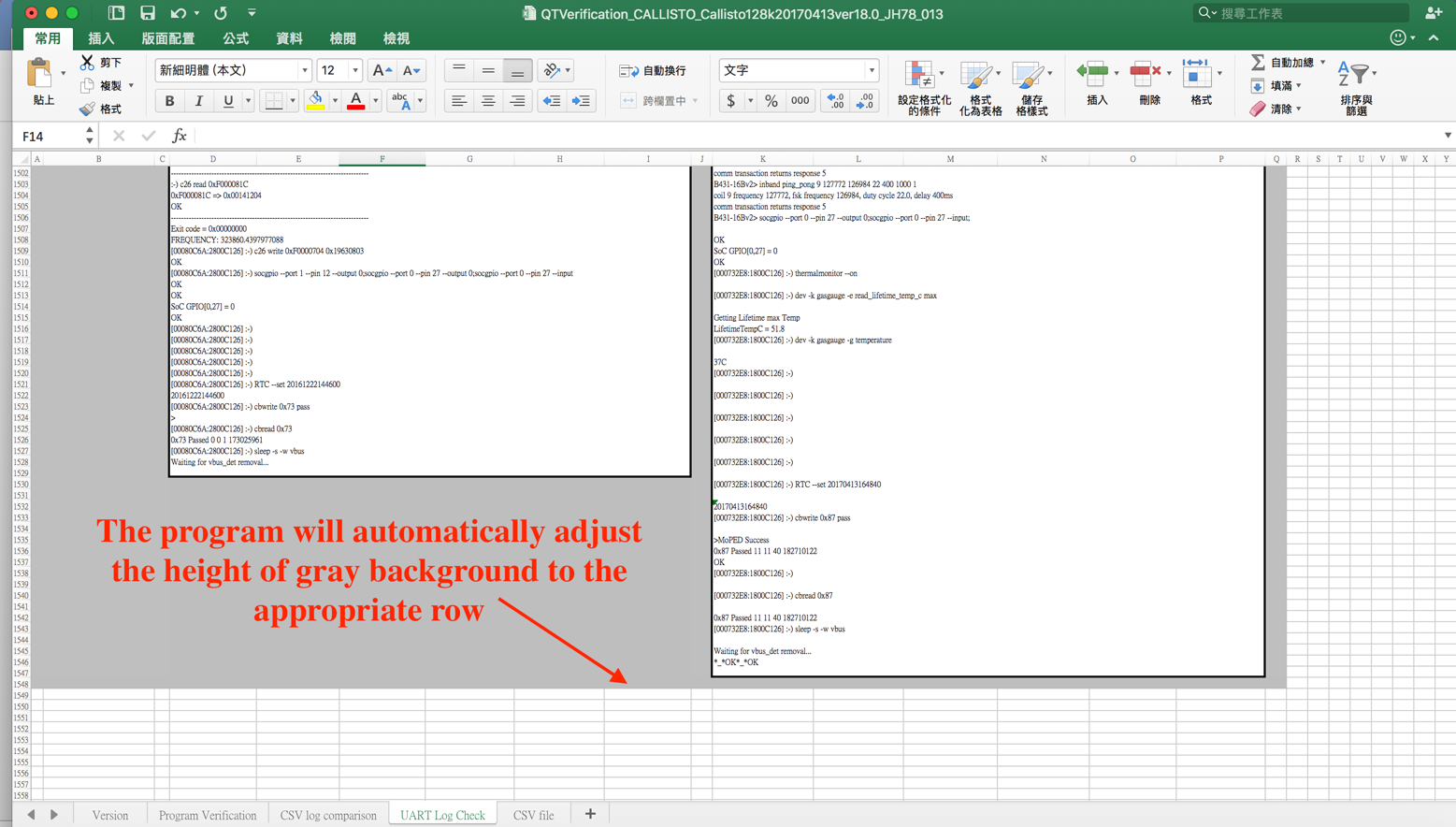


接下來請將左右兩個data cell section中間，多餘的灰色column刪除（只保留一個就好），然後同樣請將最右側的灰色背景column刪到剩下一欄就好。如果有多的灰色背景欄位，執行腳本時會出錯。



我之後可能會更新avt.py腳本，讓程式用更聰明的方式去判斷左右兩個data cell section的起始及終止欄位的位置索引，但是你也不用擔心，因為刪除多餘欄位的這個動作，你只需要做一次而已。因為在這之後，avt.py腳本每次都會自動幫你產生相同的樣板格式，使用自動化腳本的好處是，可以讓不同station的Excel文件，都能產生具有一致性的樣板格式。

然後跟是**CSV log comparison**這個worksheet一樣，**UART Log Check**這個worksheet裡面的灰色背景的row數量，你可以不用去做刪除的動作，因為腳本會在做完資料處理之後，判斷左右兩個data cell section的row數量，找出較大的那一邊，然後自動將灰色背景的高度設為row數量較多的那一側的row值加1。



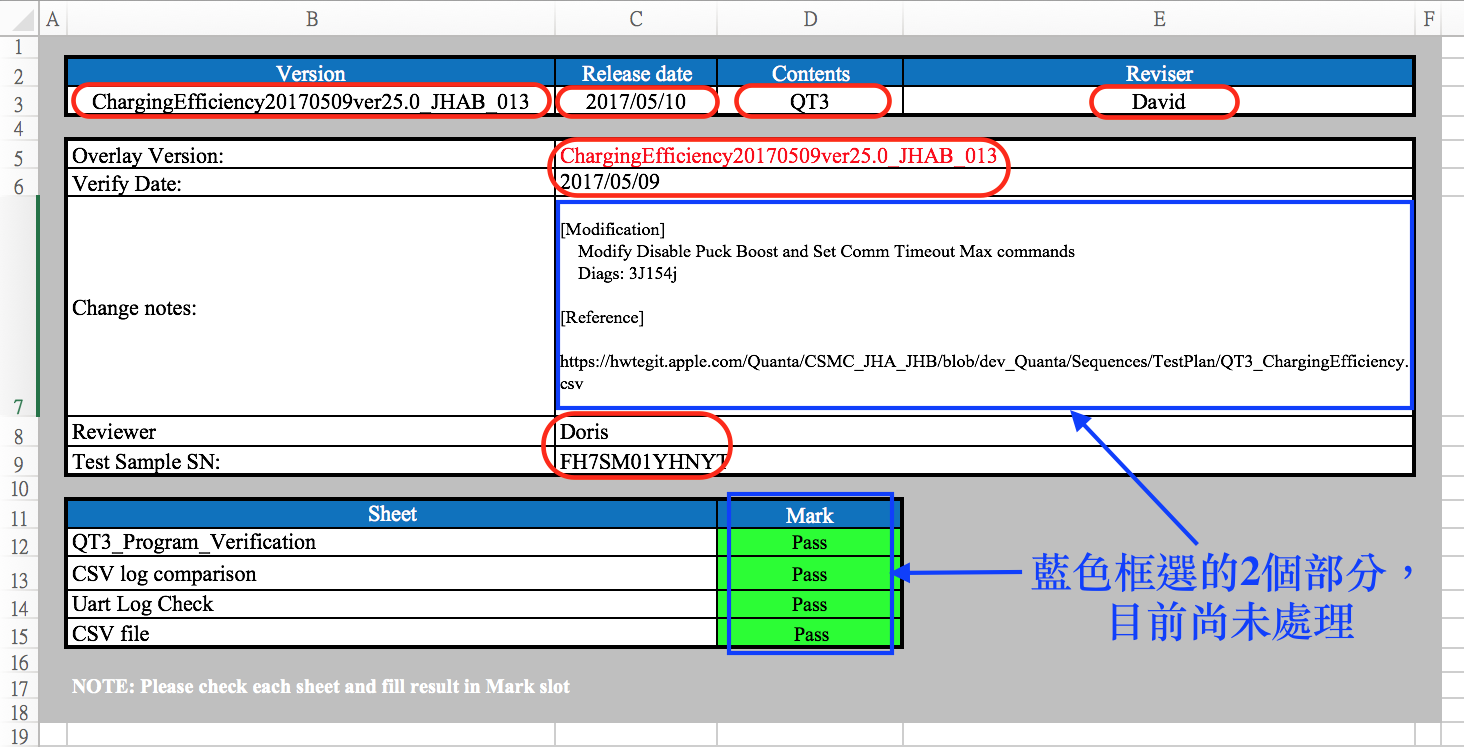
MODEM log跟CSVLOG不太一樣，因為每次的文件內容長度和寬度都是不一樣的，但是你不用擔心，avt.py腳本會計算左右兩側的資料長度和寬度，自動調整灰色背景的高度，並將左右兩個data cell section的寬度size設為相同，以及將每一個資料欄位的寬度設為相同的值（目前腳本中預設為30，這個值可以調整）。

最後是**CSV file**這個worksheet，這個不需要做任何變更，腳本會自動從新的csv file讀取資料，並將資料儲存到這個worksheet裡面。

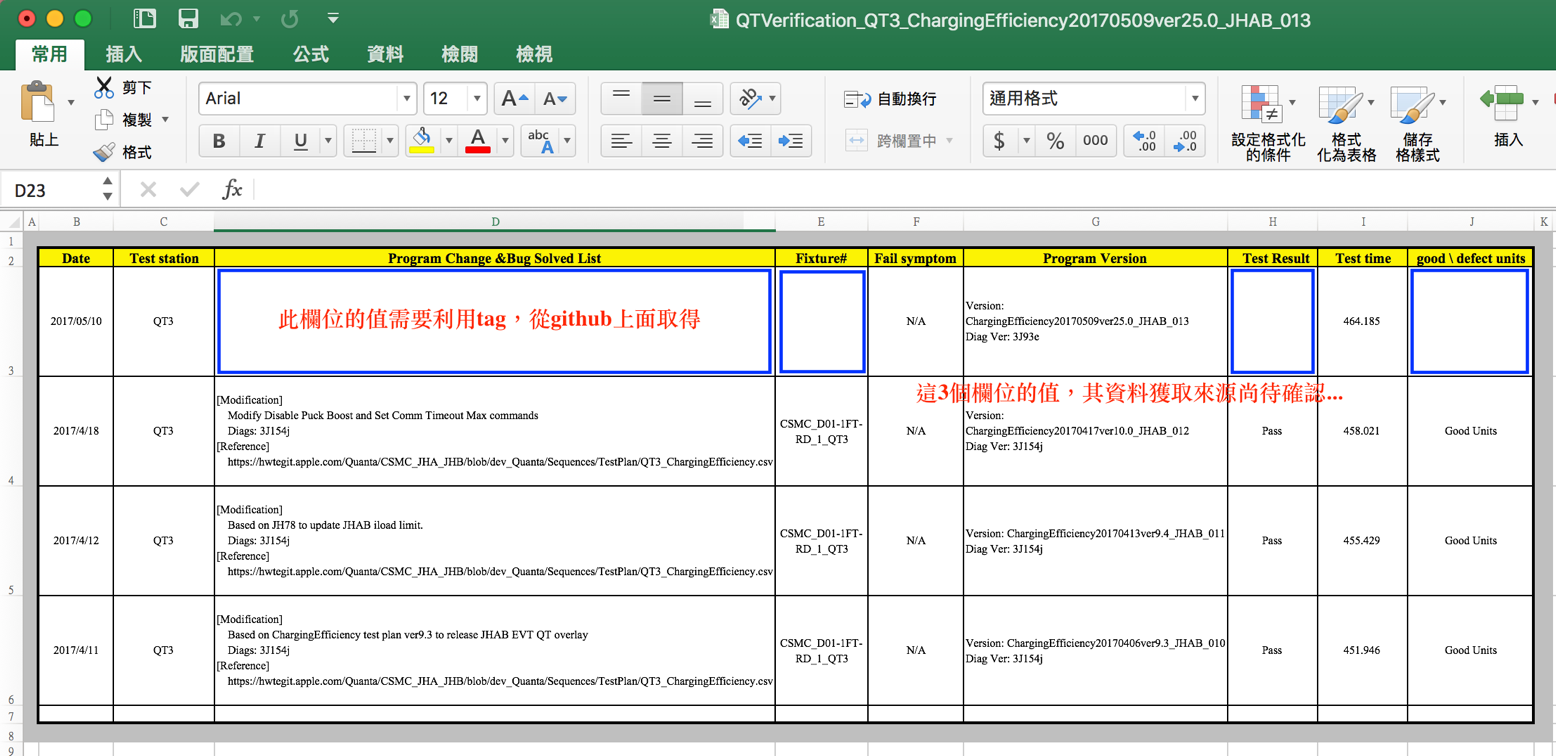
# Further development（TODO List）

目前avt.py腳本，針對處理資料的部分已經有了一定程度的完善，但是針對Excel裡面的資料，還有還是有一些尚未處理到的部份，這些部分需要再進一步的完善。以下說明一下尚未處理到的部份：

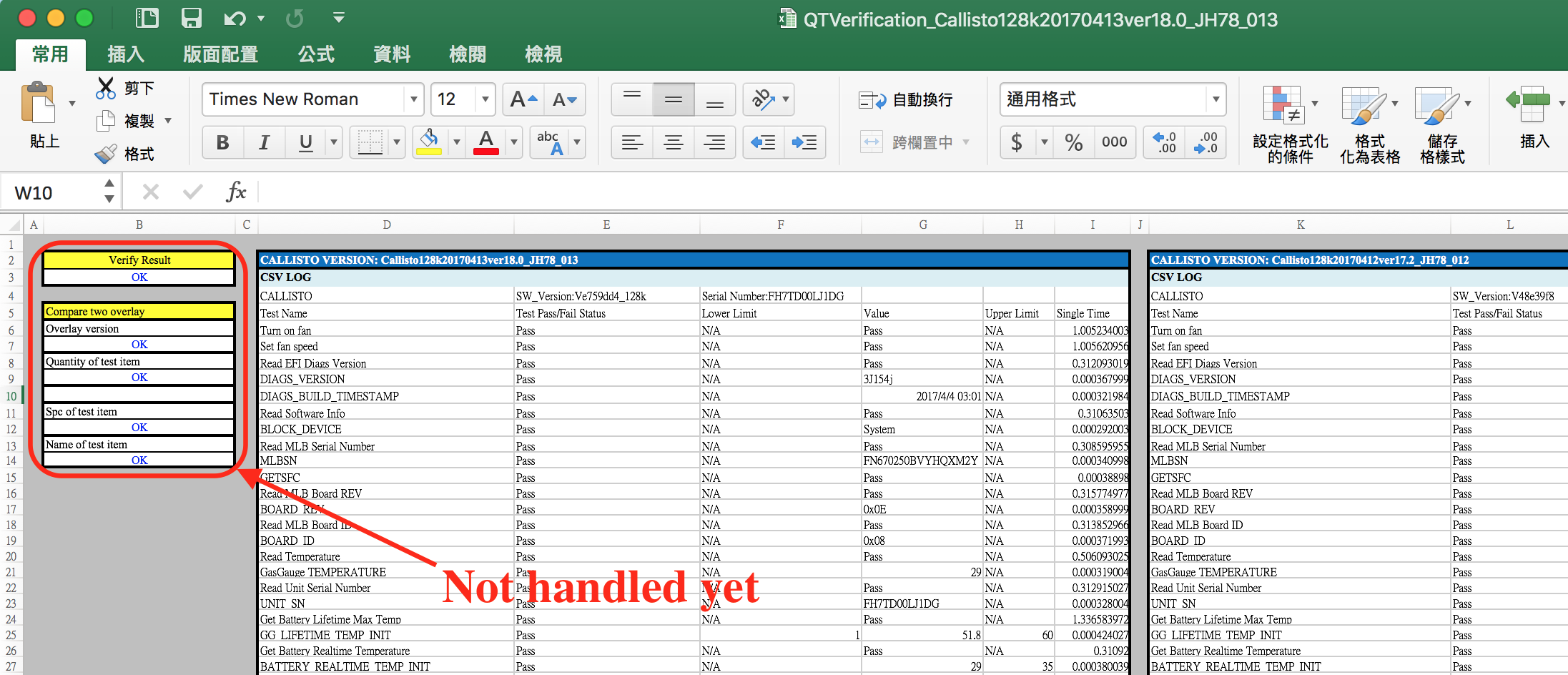
**1. Version** worksheet裡面，如下圖中，紅色框選的部分，目前都能自動產生資料，但還有2個地方還沒處理好（藍色框選的部分）：**Change note**的部份，需要整合git指令來抓資料，但github上，tag對應到release note的內容，仍有資料一致性的問題，目前尚待確認處理。而下方**Mark**的部分，目前也不知道這些值（Pass或Fail）要從什麼來源取得，一樣需要再作進一步討論（或者就是保留Pass，有需要的話，再去手動修改內容）



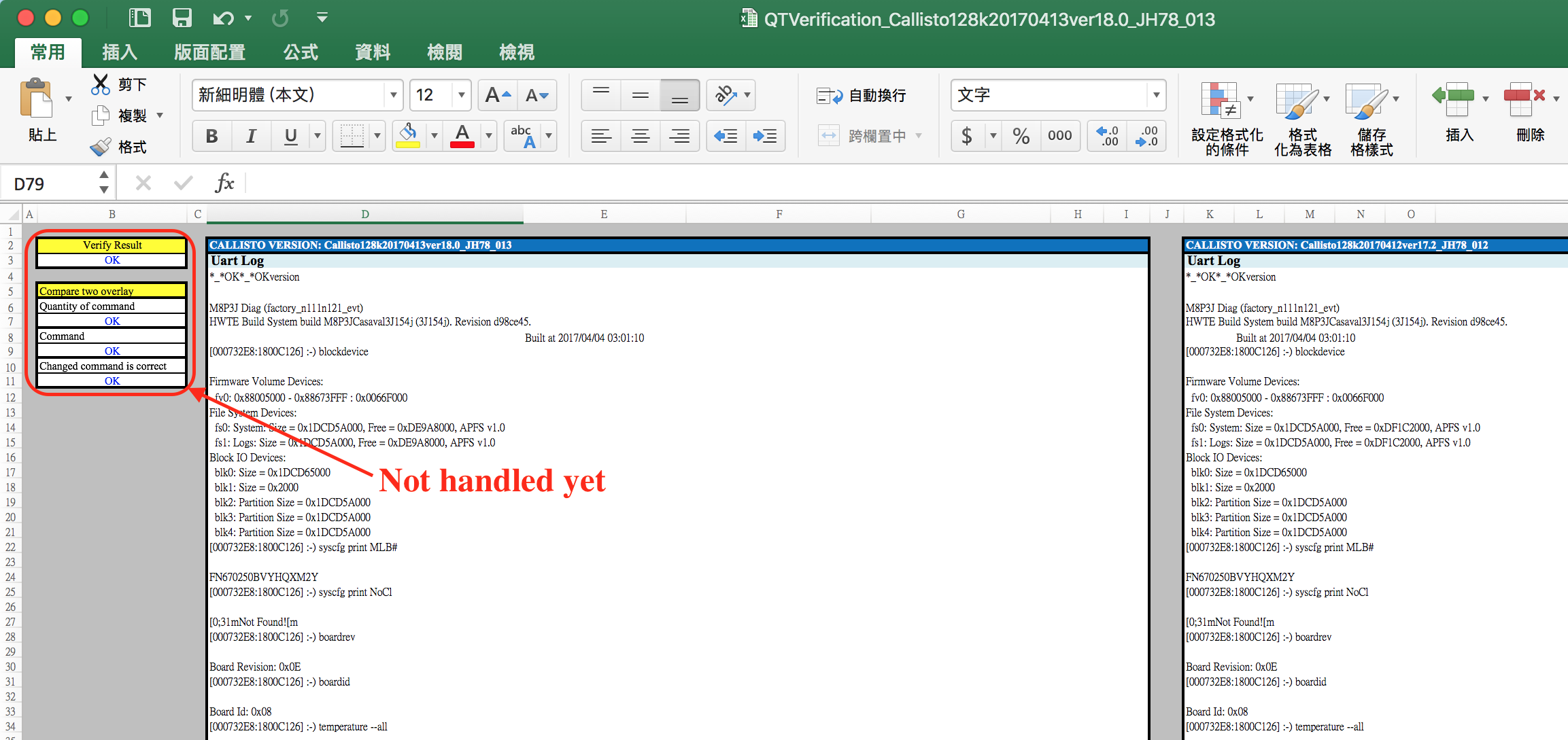
**2. Program Verification**裡面有4個部分，目前還沒處理到（判斷這些值的來源＆方式？），其中**Program Change &Bug Solved List**欄位的值，跟Version worksheet一樣，內容需要從github上面取得（目前尚待確認處理）。而**Test result**和**good \ defect units**兩個欄位的值，其資料來源和取得方式，也需要再作進一步討論（或者可以將值，預設為跟上一筆資料相同）。

****

**3. CSV log comparison**裡面有1個部分，目前還沒處理到（判斷這些值的來源＆方式？）



**4. UART Log Check**裡面有1個部分，目前還沒處理到（判斷這些值的來源＆方式？）



目前資料取得有困難的大概就是以上這4張截圖說明的部分，如有任何意見，歡迎feedback