

0x02. 試構想具體的設計(不需撰寫程式), 以便可以通過裝置收集 POS 系統傳送到 USB Thermal Printer 的發票列印數據, 並把收集到的數據發送到雲端上。其示意圖如下:



設計一：裝置扮演的角色

在我了解貴司目前的產品之中，我明白了一件重要的前題，就是商店不願做太多的改變，因此裝置必須無縫接軌原本的運作方式

1、不需做任何修改

2、店家只需簡單的把原本USB Thermal Printer插到裝置上，而原本留給USB Thermal Printer的USB孔則由裝置接上。

POS --- usb port --- USB Thermal Printer

POS --- usb port --- device --- usb port ---USB Thermal Printer

因此，第一個前題應該是原本USB Thermal Printer扮演的角色，要由Amazing device來扮演，以此推論下！

device有可能要實作1個Dummy USB Thermal Printer device(or USB Thermal Printer Hub)讓店家的POS機器仍將device視為一台USB Thermal Printer以達到無縫接軌，無需任何修改！

設計二：裝置與雲端的協同合作

將收集到的發票列印數據上傳到雲端

1、強固容錯的網路環境

裝置必須同時支援Ethernet/WIFI/4G等網路環境，當偵測到connection lost時，必須有能力迅速切換至其他連線

2、數據上傳到雲端必須俱備Sync同步的能力

有時因為環境問題(網路、雲端、裝置)導致裝置無法即時與雲端交換資料時，裝置必須先儲存數據在本地，待環境問題排除時，必須有能力支援續傳/同步等功能

3、周期性上傳數據(每隔2~4小時/排程店家非營業時間上傳)

設計三：裝置本身的支援度

裝置本身必須俱備容錯、強固的能力

- 1、OTA更新韌體以強化系統的穩定性
- 2、OTA更新韌體以支援更多的USB Thermal Printer Driver
- 3、周期性讓雲端Server知道自己目前的狀態(on/off line)
- 4、當連接的USB Thermal Printer有問題，裝置必須有能力指出問題
- 5、裝置本身不穩定時，必須有能力回復(watchdog/排程重開機)
- 6、裝置本身必須有更多的燈號顯示(網路狀況、印表機狀況、雲端狀況等等)
- 7、Error handling：當出現狀況時，必須有能力自行回復(重開機、recovery-mode)