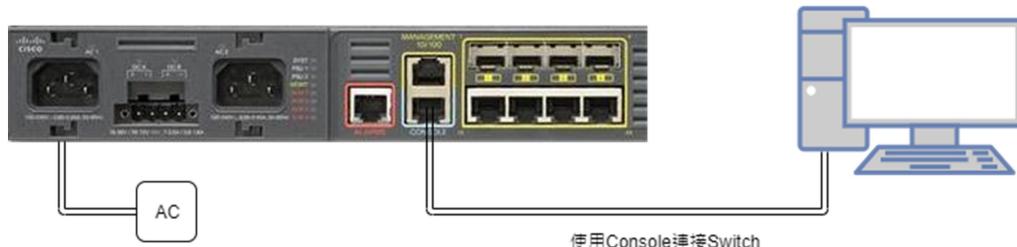


一、 備妥：

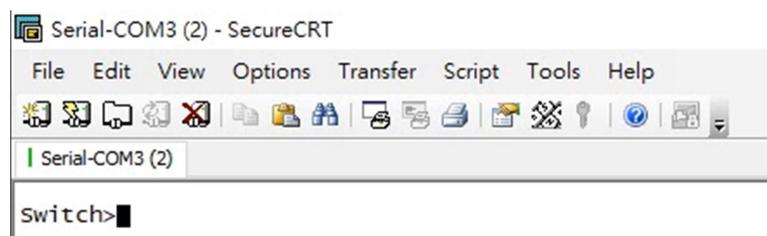
1. 電腦 1 台 (含 SecureCRT v6.2.0 以上版本)
2. Rollover Cable + RJ-45 to RS-232 轉換頭 或 Cisco 專用 Console 平行線
3. RS-232 to USB 轉換器
4. 欲還原的交換機設定檔及自動還原腳本
5. 新品交換機

二、 步驟

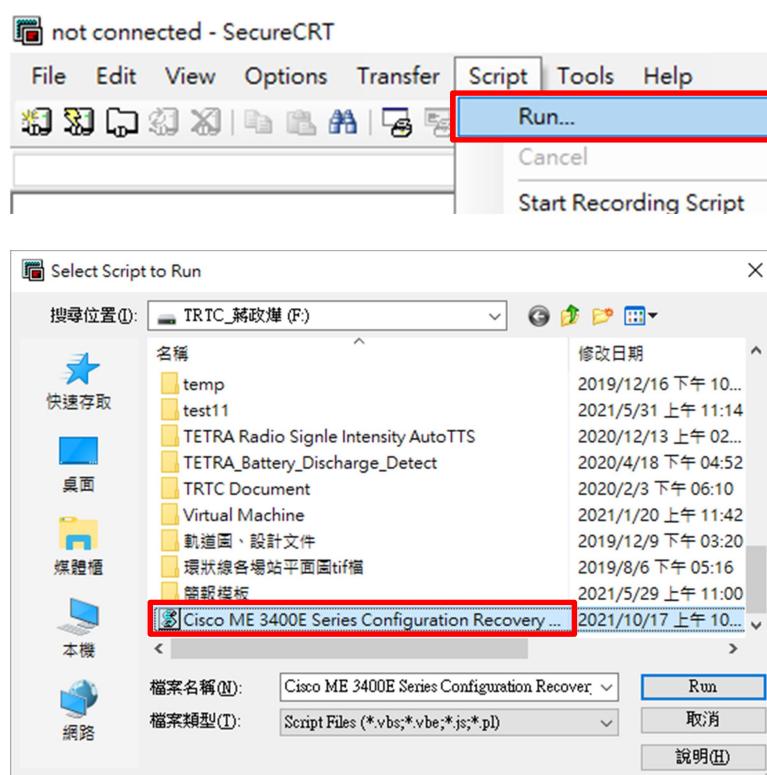
1. 將電腦以 Console 方式與交換機連接，並將交換機上電開機



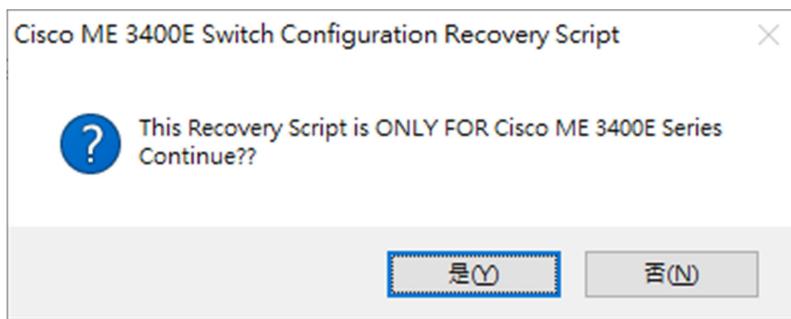
2. 使用 SecureCRT 以串列連線方式連入交換機



3. 開機完成後，畫面出現 [交換機名稱#] 或 [Switch>](空機) 提示字元時，點選功能選單 Script → Run...，並選擇自動還原腳本



4. 出現「此腳本僅適用 Cisco ME 3400E Series 交換機」提示訊息，按下「是」繼續執行，按下「否」結束腳本



5. 還原腳本將自動判斷交換機當前執行權限層級並檢查下列三項資訊是否相符

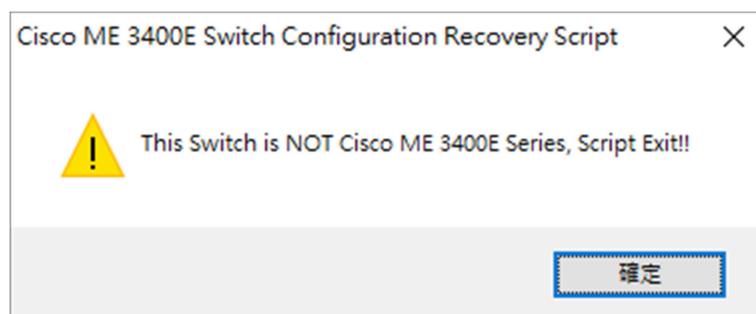
- A. 型號：ME-3400EG-12CS-M
- B. IOS 軟體系列：ME340x-METROIPACCESSK9-M
- C. IOS 檔案名稱：me340x-metroipaccessk9-mz.122-60.EZ4.bin

```

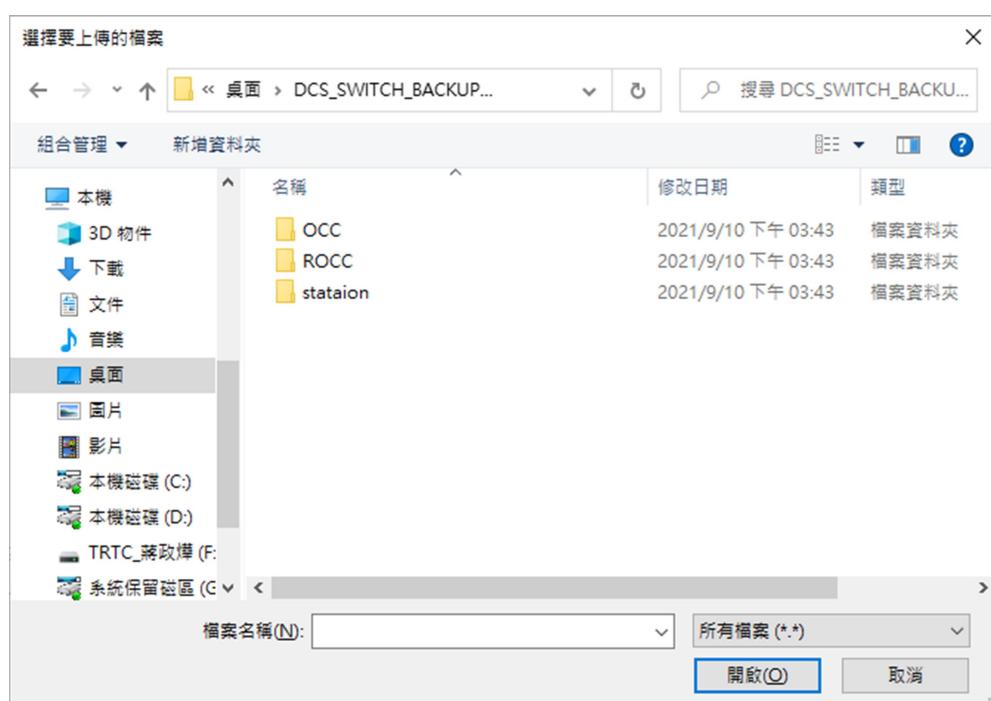
Y19-ZC-A#sh ver | inc Model number
Model number          : ME-3400EG-12CS-M
Y19-ZC-A#sh ver | inc Software
cisco IOS Software, ME340x Software (ME340X-METROIPACCESSK9-M), Version 12.2(60)EZ4, RELEASE SOFTWARE (fc3)
Y19-ZC-A#sh ver | inc image file
System image file is "flash:/me340x-metroipaccessk9-mz.122-60.EZ4.bin"

```

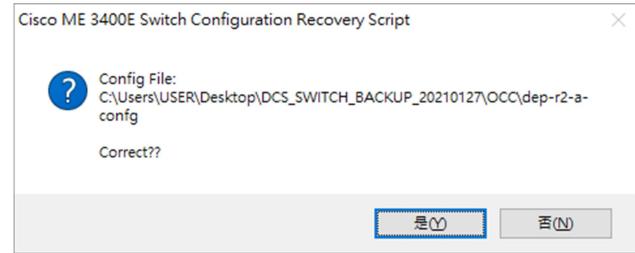
當其中任一項資訊不符時，腳本將會出現「此交換機並非 Cisco ME 3400E Series」提示訊息並結束執行



6. 通過交換機型號檢查後，會出現檔案選擇視窗，請選擇要還原的交換機設定檔



7. 因後續檔案上傳及重新啟動指令皆為自動執行，為避免誤選檔案，上傳非供 Switch 使用之設定檔，造成設備異常或無法開機。在選擇檔案後，會要求再次確認所選擇的檔案路徑是否正確



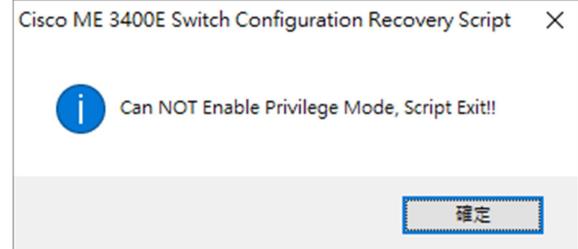
倘所選擇的檔案錯誤，請按下「否」結束執行，並回到第 3 步重新啟動腳本。
如選擇正確，請按下「是」繼續執行

8. 確認所選檔案無誤後，腳本將自動進入特權執行模式(Privilege / Enable Mode)，並清除既有 VLAN Database、Startup Config

```
ROCC-ZC-A#enable
ROCC-ZC-A#enable
ROCC-ZC-A#enable
ROCC-ZC-A#delete vlan.dat
Delete filename [vlan.dat]?
Delete flash:/vlan.dat? [confirm]
%Error deleting flash:/vlan.dat (No such file or directory)
ROCC-ZC-A#erase startup-config
Erasing the nvram filesystem will remove all configuration files! Continue? [confirm]
[OK]
Erase of nvram: complete
ROCC-ZC-A#
*Mar 1 10:56:21.616: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
```

如果無法進入特權執行模式(包含如下圖之需要輸入 Enable 密碼等錯誤狀態)，腳本將會出現「無法進入特權執行模式提示訊息並結束執行

```
Y19-ZC-A>enable
Password:
Password:
Password:
% Bad secrets
```



9. 清除完成後，將啟動 Xmodem 傳輸協定，上傳設定檔至交換機

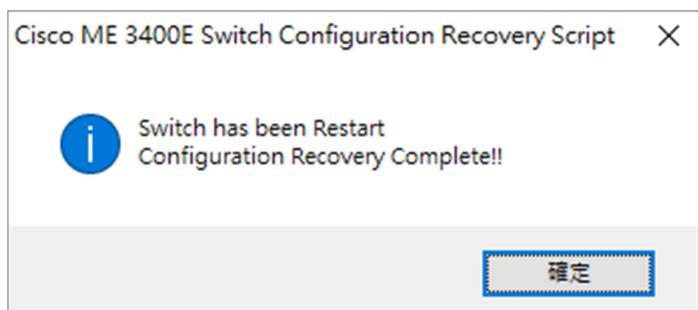
```
ROCC-ZC-A#copy xmodem: startup-config
Destination filename [startup-config]?
Begin the Xmodem or Xmodem-1K transfer now...
CC
Starting xmodem transfer. Press Ctrl+C to cancel.
Transferring y19-zc-a-config...
 100%      4 KB      0 KB/s 00:00:06      0 Errors
 100%      4 KB      0 KB/s 00:00:06      0 Errors

4864 bytes copied in 11.123 secs (437 bytes/sec)
```

Cisco Switch 預設使用 Xmodem-CRC 進行循環冗餘校驗及檔案完整性驗證

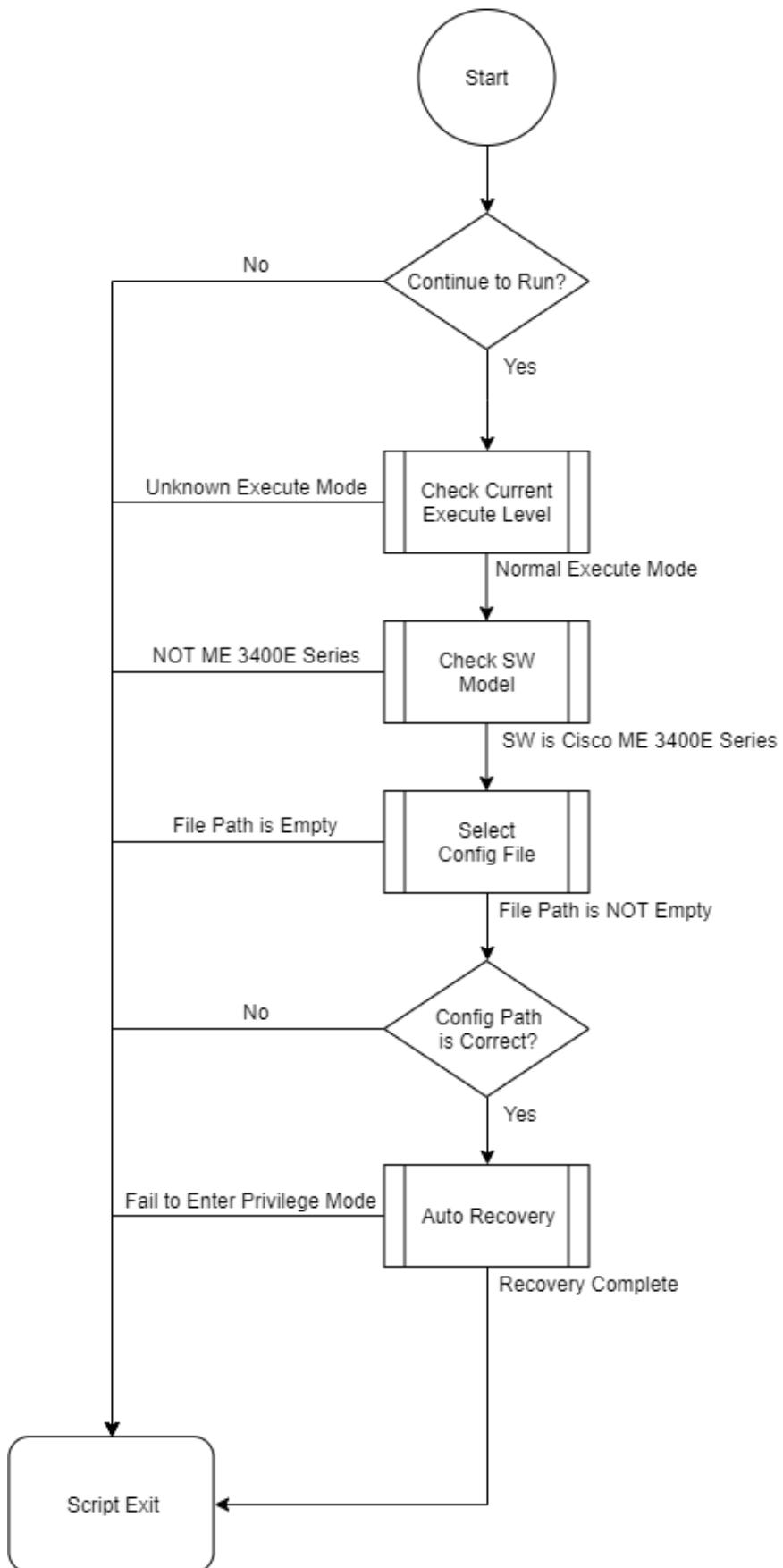
此外，SecureCRT 預設 Xmodem Block 之傳輸單位長度為 128 Bytes + 5 Bytes

10. 設定檔上傳完成後，交換機將自動重啟，並會出現還原完成之提示訊息

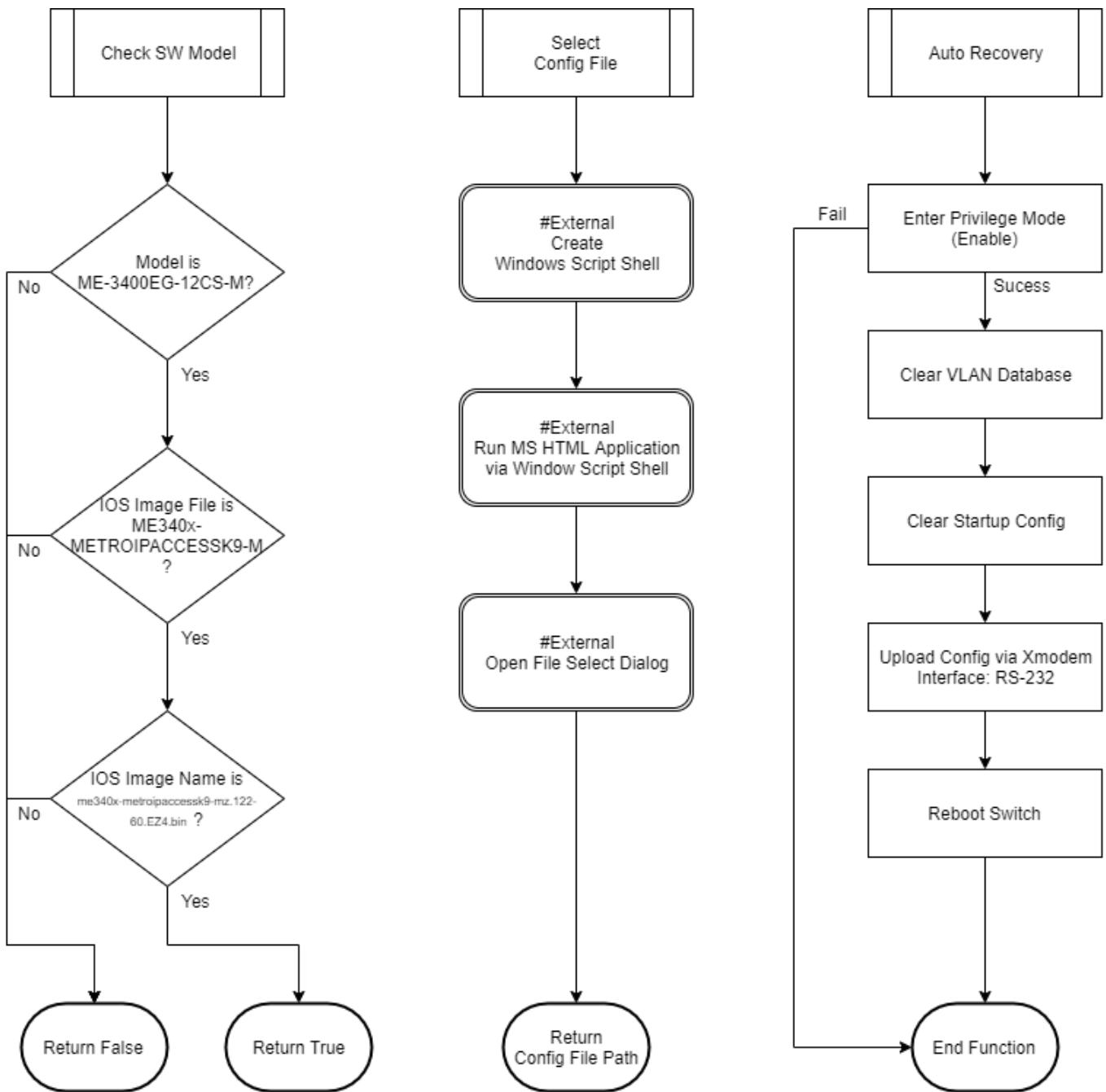


三、腳本設計流程圖

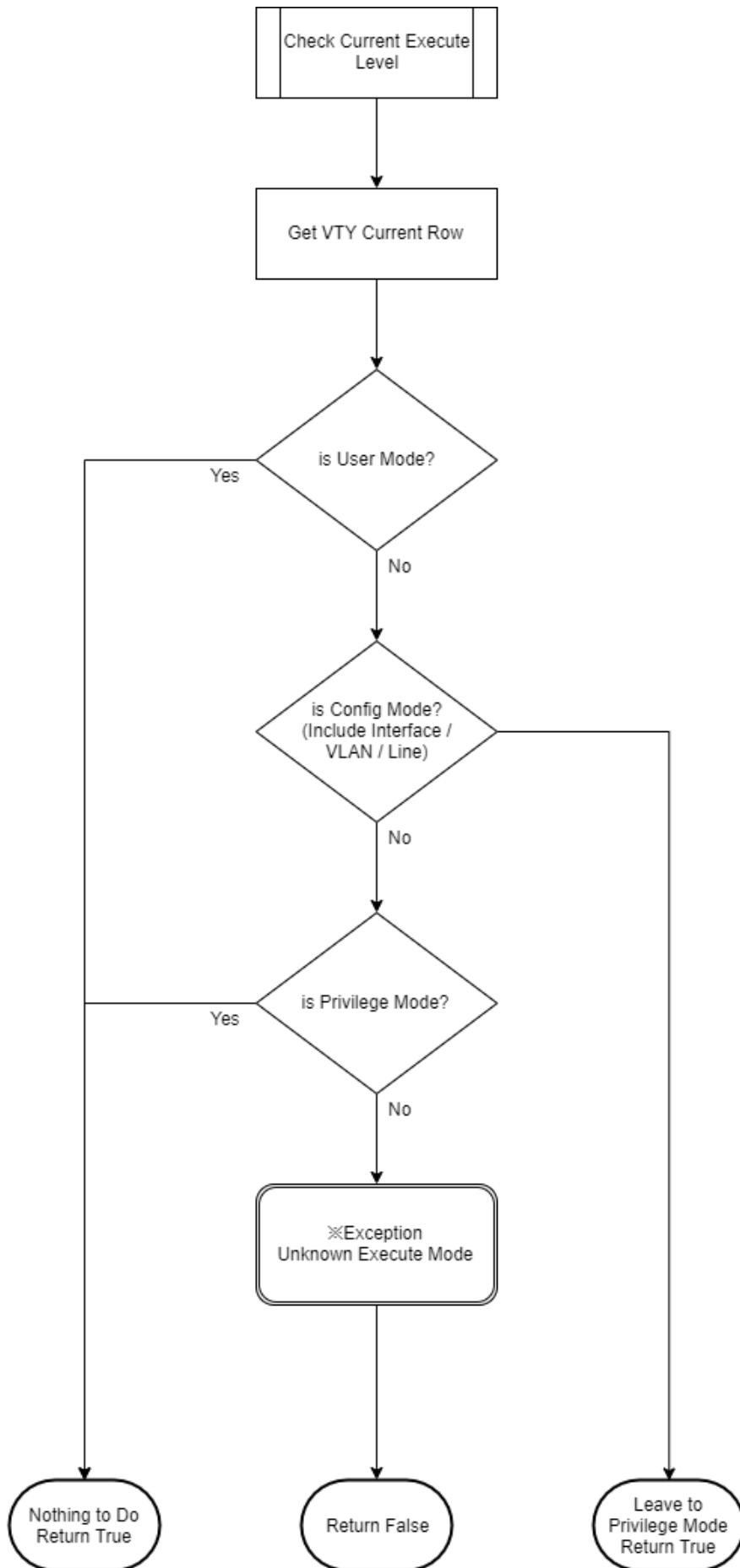
主架構流程圖



模組流程圖 - 1



模組流程圖 - 2



四、Xmodem 傳輸協定

Xmodem 協定開發於 1977 年，為串列通訊中一種古老的泛用型非同步文件傳輸協定，可以做為 Cisco 設備上傳設定檔、IOS 作業系統或救援模式的傳輸協定。

在 Xmodem 標準模式下，一個完整的資料塊(Block)長度固定為 132 Bytes，其中包括 128 Bytes 的數據區塊及 4 Bytes 的控制區塊，封包格式如下：

SOH	Block No.	Block No.(反碼)	Data Block	Checksum
1 Bytes	1 Bytes	1 Bytes	128 Bytes	1 Bytes

其中資料塊均以 SOH 控制碼(^A, 0x01)為開頭，並附上區塊編號(0x01 ~ 0xFE)、反向區塊編號(0xFE ~ 0x01)。最後為校驗和，其值為數據區塊內容參照 ASCII Table 之 Hex 值總和除以 256 的餘數。

需注意的是，不論傳送資料的大小，每個資料塊中的數據區塊長度均固定為 128 Bytes。意即，當傳輸至資料末端時，數據長度可能會不滿足 128 Bytes，此時協定將以控制碼 Ctrl-Z(^Z, 0x1A)填補剩餘空間。

Xmodem 協定亦有其他增強型變體，如 Xmodem-CRC 及 Xmodem-1K，以及改良協定如 Ymodem、Zmodem 及 Kermit 等。

Cisco 設備即預設使用 Xmodem-CRC 格式進行傳輸。

Xmodem-CRC 與標準模式相差不多，僅將資料塊最後的「校驗和」改為「循環冗餘校驗」，資料塊長度則固定為 133 Bytes。

其中 16bit CRC 可拆分為高、低位元組，意即 CRC-H (8bit)、CRC-L (8bit)。

SOH	Block No.	Block No.(反碼)	Data Block	16bit CRC
1 Bytes	1 Bytes	1 Bytes	128 Bytes	2 Bytes

該模式在 VTY 虛擬終端畫面上的顯式特徵為，於啟動傳輸時，Rx 會向 Tx 發送字母 C(0x43)，藉以向傳送端表示將使用 CRC 循環冗餘校驗。在標準校驗和模式下則改為發送控制碼 NAK(^U, 0x15)

```
ROCC-ZC-A#copy xmodem: startup-config
Destination filename [startup-config]?
Begin the Xmodem or Xmodem-1K transfer now...
CC
Starting xmodem transfer. Press Ctrl+C to cancel.
Transferring y19-zc-a-configure...
 100%      4 KB      0 KB/s 00:00:06      0 Errors
 100%      4 KB      0 KB/s 00:00:06      0 Errors
4864 bytes copied in 11.123 secs (437 bytes/sec)
```

Xmodem-1K 模式強制採用 CRC 循環冗餘校驗，數據區塊長度由 128 Bytes 提升至 1024 Bytes，資料塊表頭以控制碼 STX (^B, 0x02) 取代控制碼 SOH (^A, 0x01)。資料塊總長為 1029 Bytes。因為數據區塊長度的增加，Xmodem-1K 的傳輸速率較 CRC 及標準模式來的有效率。

STX	Block No.	Block No.(反碼)	Data Block	16bit CRC
1 Bytes	1 Bytes	1 Bytes	1024 Bytes	2 Bytes

與標準模式相同，當資料塊中的數據區塊長度不足 1024 Bytes 時，協定將以控制碼 Ctrl-Z(^Z, 0x1A) 進行填補，對於交換機或嵌入式單板電腦等具有記憶體容量限制之設備而言，此模式在特定情形下具有較高程度的記憶體浪費。

下表為 Xmodem 協定的控制碼定義：

控制碼	ASCII Hex Code	跳脫字元	用途	發送 角色
SOH	0x01	^A	資料塊表頭，用以表示數據區塊為 128 Bytes	Tx
STX	0x02	^B	資料塊表頭，用以表示數據區塊為 1024 Bytes	Tx
ETX	0x03	^C / Ctrl-C	Tx 手動終止傳輸，Rx 不須再傳送 ACK	Tx
EOT	0x04	^D	傳輸結束	Tx
ACK	0x06	^F	資料塊校驗無誤，可發送下一個資料塊	Rx
NAK	0x15	^U	1. 在標準模式下等待接收，通知 Tx 使用校驗和 2. 資料塊校驗失敗，通知 Tx 重送	Rx
CAN	0x18	^X	Rx 無條件終止傳輸，Tx 不須再傳送 EOT	Rx
SUB	0x1A	^Z / Ctrl-Z	數據區塊末端填充用控制碼	Tx
字母 C	0x43	N/A	等待接收，通知 Tx 使用 CRC 循環冗餘校驗	Rx

Xmodem 傳輸過程以 Rx 為主體，下列為 Xmodem-CRC 應答流程範例：

Tx	方向	Rx	說明
	<<	C	以 CRC 模式等待接收
SOH 01 FE Data[128] CRC-H CRC-L	>>		以 128 Bytes 長度傳送
	<<	ACK	校驗無誤
SOH 02 FD Data[128] CRC-H CRC-L	>>		
	<<	ACK	
SOH 03 FC Data[128] CRC-H CRC-L	>>		
	<<	NAK	校驗失敗
SOH 03 FC Data[128] CRC-H CRC-L	>>		重送資料塊
	<<	ACK	
SOH 04 FB Data[100] 0x1A[28] CRC-H CRC-L	>>		資料末端，以 Ctrl-Z(^Z, 0x1A) 填充
	<<	ACK	
EOT	>>		Tx 傳輸結束
	<<	ACK	Rx 傳輸結束

五、Xmodem 資料塊傳輸之驗證處理

標準校驗和模式資料塊範例：

表頭	Block No.	Block No.(反碼)	Data Block	Checksum
1 Bytes	1 Bytes	1 Bytes	128 Bytes	1 Bytes
SOH, 0x01	0x03	0xFC	0x41[50] 0x61[50] 0x1A[28]	0x7C

Rx 在接收到以控制碼 SOH / STX 為開頭的資料塊後，將進行下列幾個步驟藉以確認資料傳輸完整性(以上述資料塊為計算範例)：

A. 確認資料塊序號完整性：

Block No. + Block No.(反碼)，其值應為 0xFF。

$$0x03 + 0xFC = \textcolor{red}{0xFF}$$

Block No. NOR Block No.(反碼)，其值應為 0x00

$$0x03 \text{ NOR } 0xFC = 0xFFFF FFFF FFFF FF00 \text{ (OverFlow)}$$

因資料塊序號僅佔 1 個位元組，故其計算產生之溢位位元不計入驗證結果，最終值為 $\textcolor{red}{0x00}$

倘上述兩項檢查不符預期，則傳送控制碼 NAK(^U, 0x15)請求 Tx 重送資料塊

B. 檢查資料塊序號是否為期望值：

依上述資料塊範例，其前一資料塊序號為 0x02，故次一序號期望值應為 0x03，倘收到非序號 0x03 之資料塊時，Rx 將發出控制碼 CAN(^X, 0x18)終止傳輸。

倘收到之資料塊序號與前一資料塊序號相同，則 Rx 將忽略重複發送的 Block，並向 Tx 發送控制碼 ACK(^F, 0x06)。

C. 進行校驗和(Checksum)驗證

對數據區塊進行驗證，以標準模式校驗和(Checksum)為例，其計算方式如下所述：

Data Block (128 Bytes)	
數據內容	50 個大寫 A、50 個小寫 a、28 個末端控制碼 Ctrl-Z(^Z, 0x1A)
ASCII Hex Code	0x41[50] 0x61[50] 0x1A[28]

$$\begin{aligned} \text{其數據總和值} &= (0x41 * 50) + (0x61 * 50) + (0x1A * 28) \\ &= 0x0CB2 + 0x12F2 + 0x02D8 \\ &= 0x227C \end{aligned}$$

$$\text{總和值餘除 256 之 Checksum 為：} 0x227C \bmod 256 = \textcolor{red}{0x7C}$$

最終結果應與資料塊中最後一個位元組相同。倘驗證結果相同則傳送控制碼 ACK(^F, 0x06)，不同則傳送控制碼 NAK(^U, 0x15)。

六、 Xmodem 數據區塊末端 Ctrl-Z 填充

Xmodem 數據區塊在所乘載之資料未滿 128 Bytes 時(1K 模式為 1024 Bytes)，會以控制碼 Ctrl-Z(^Z, 0x1A)進行填補，其附加長度計算方式如下：

標準模式： $128 - [\text{FileSize(Bytes)} \bmod 128]$ = 控制碼填補長度
1K 模式： $1024 - [\text{FileSize(Bytes)} \bmod 1024]$ = 控制碼填補長度

以 Y19-ZC-A 之 Configuration File 為例，原始檔案長度為 4829 Bytes，透過 Xmodem 上傳至 Cisco 交換機後，其檔案末端將會被附加上 35 個控制碼 Ctrl-Z(^Z, 0x1A)。

128 - [4829 Mod 128]	描述:	y19-zc-a-config
= 128 - 93	位置:	F:\cisco setting upload\自動腳本
= 35 個	大小:	4.71 KB (4,829 位元組)

另於 Cisco 交換機所顯示的傳輸資訊內，亦可檢視附加填充控制碼後的總資料長度

```
ROCC-ZC-A#copy xmodem: startup-config
Destination filename [startup-config]?
Begin the Xmodem or Xmodem-1K transfer now...
CC
Starting xmodem transfer. Press Ctrl+C to cancel.
Transferring y19-zc-a-cfg...
 100%      4 KB      0 KB/s 00:00:06      0 Errors
 100%      4 KB      0 KB/s 00:00:06      0 Errors
```

※總資料長度必為 128 之整數倍，

4864 bytes copied in 11.123 secs (437 bytes/sec)

於 Cisco 交換機檢視 Startup Config，其檔案長度與傳輸長度相符

```
Y19-ZC-A#sh startup-config
using 4864 out of 524288 bytes
!
! Last configuration change at 12:06:40 UTC Tue Apr 21 2020 by gemadmin
! NVRAM config last updated at 12:06:41 UTC Tue Apr 21 2020 by gemadmin
!
version 12.2
no service pad
```

亦可於末端看到相等數量之 $\wedge z$ (此範例為 35 個 $\wedge z$)

```
login local
transport input all
!
monitor session 1 source interface Gi0/3 - 4
monitor session 1 destination interface Gi0/5 - 6 ingress untagged vlan 30
ntp server 172.27.1.32 prefer
ntp server 172.27.1.160
end
```

控制碼 Ctrl-Z(^Z, 0x1A)雖被填充至 Startup Config 末端，但對於 Cisco IOS 而言，並不影響其 Bootloader、POST、初始化及讀取 Startup Config 至 Running Config 之過程，其原因如下：

- A. 在 Startup Config 內，填充控制碼前的最後一行指令為「end」。對於 Cisco IOS 而言，該指令為退出設定層級模式(Config Mode)。
- B. Cisco IOS 為運作於 IBM PowerPC 處理器之 Unix-like 網路作業系統。在原生 Unix 作業系統中，Ctrl-Z 被用於退出當前正在執行的互動式執行緒。
- C. 對 Cisco IOS 而言，Ctrl-Z 亦為退出設定層級模式(Config Mode)之快速鍵。

```
Y19-ZC-A#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Y19-ZC-A(config)#  
Y19-ZC-A(config)#  
Y19-ZC-A(config)#
```

- D. 此外，在同屬 Unix-like 的 CP/M 作業系統(1974-1983)內，^Z(0x1A)被視為檔案末端及填充剩餘檔案空間(與 Xmodem 原理相同)

不論上傳之 Configuration File 內附加了多少個 Ctrl-Z 控制碼，或是否包含「end」指令。基於上述 A. 及 C. 項，Cisco IOS 在開機過程只要讀取到「end」或 Ctrl-Z 控制碼兩者其一(不論先後順序)，皆會視為 Startup Config 結束，故其末端填充行為不影響 Cisco 設備之正常運作。

原始程式碼版控儲存庫：

<https://github.com/cathay2218/Cisco-ME3400E-Switch-Configuration-AutoRecovery>
<https://git.io/JillK>