



MICROCHIP

Regional Training Centers

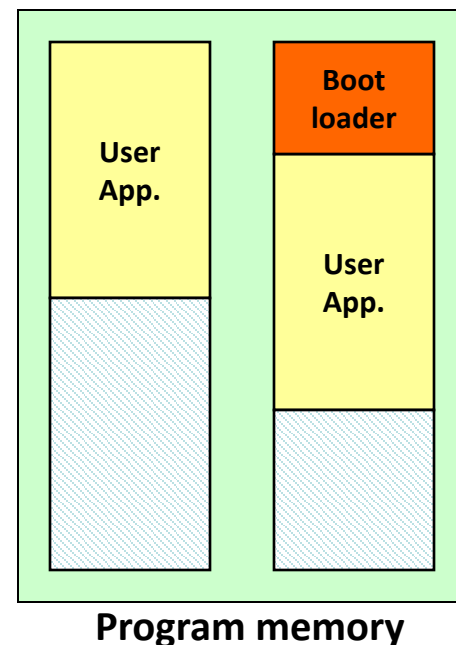
Section 6
Bootloader

What is Bootloader ?

- Bootloader 是一隻預先燒錄在MCU內的程式, 該程式可透過通訊介面, 如USB, Ethernet等與PC或其他控制器溝通, 進行自我韌體更新。
- Microchip對所有具有USB功能的MCU都有提供USB HID BootLoader的程式碼, 因此使用者可以透過Bootloader更新韌體, 不需要額外的設備或裝置。
- Microchip有提供現成的Bootloader範例程式。可以透過UART, USB, Ethernet或SD Card進行韌體更新。有興趣的學員可以參考Application Note AN1388的說明。
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en554836
- 後續將介紹如何使用最方便USB HID Bootloader。

Bootloader Introduction

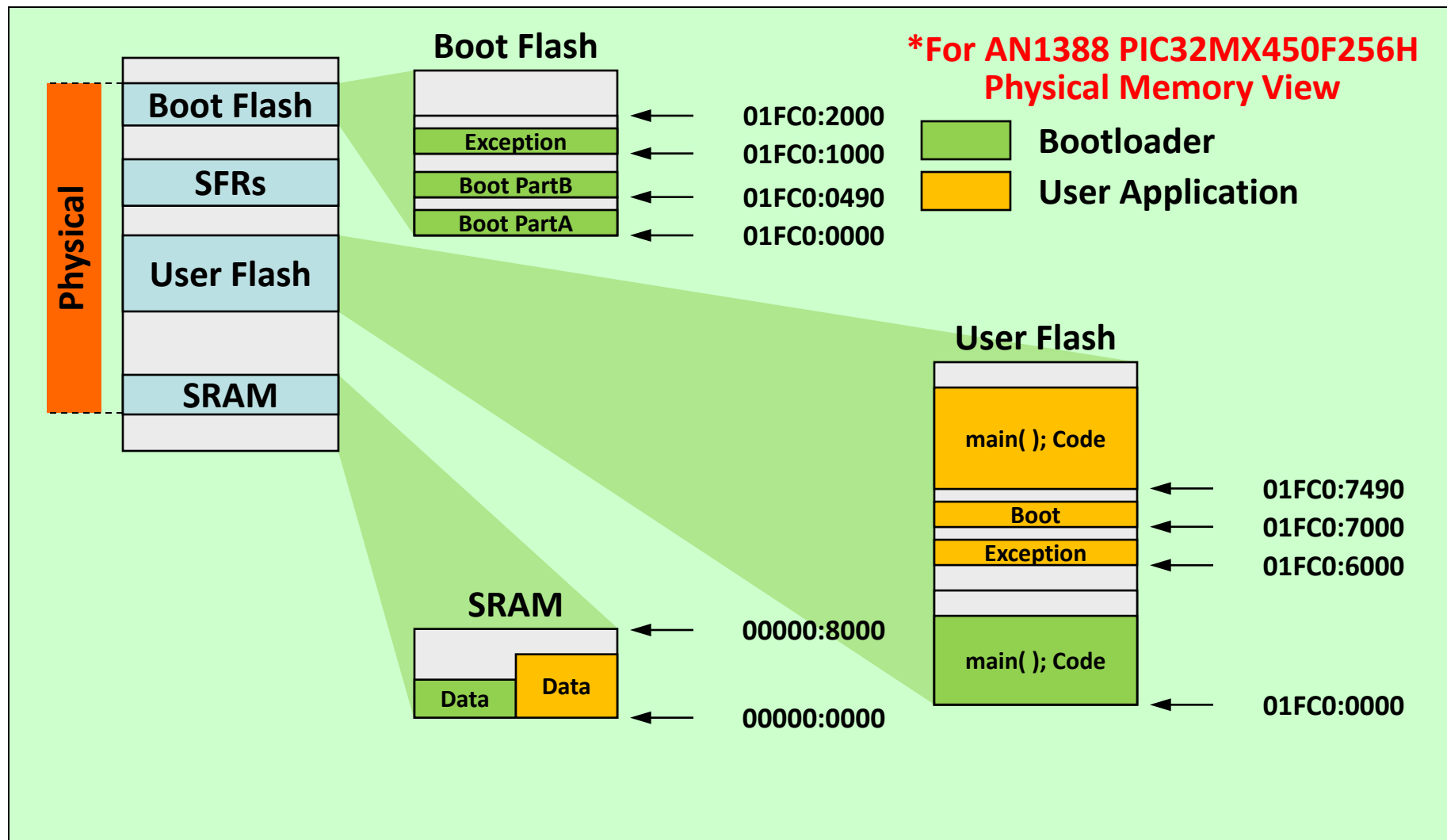
- 一般的程式記憶體的安排方式應如下圖所示, 未使用 Bootloader時安排狀況如左圖所示; 具有Bootloader時之安排狀況應如右圖所示。
- 未使用Bootloader時, 程式記憶體中, 僅有User Application, MCU 一但Reset後, 就會立即執行User Application。
- 而使用Bootloader時, 程式記憶體中, 必須同時存有User App.與Bootloader。MCU Reset後, Bootloader會透過某些機制, 如按鈕等, 確認是否進行韌體更新。或者直接將執行權交給User App.。
- 由於Bootloader跟User App.放在同一塊記憶體空間, 為了避免打架, 必須透過Linker Scripts的設定, 來將個自程式空間區隔開。



Bootloader Introduction

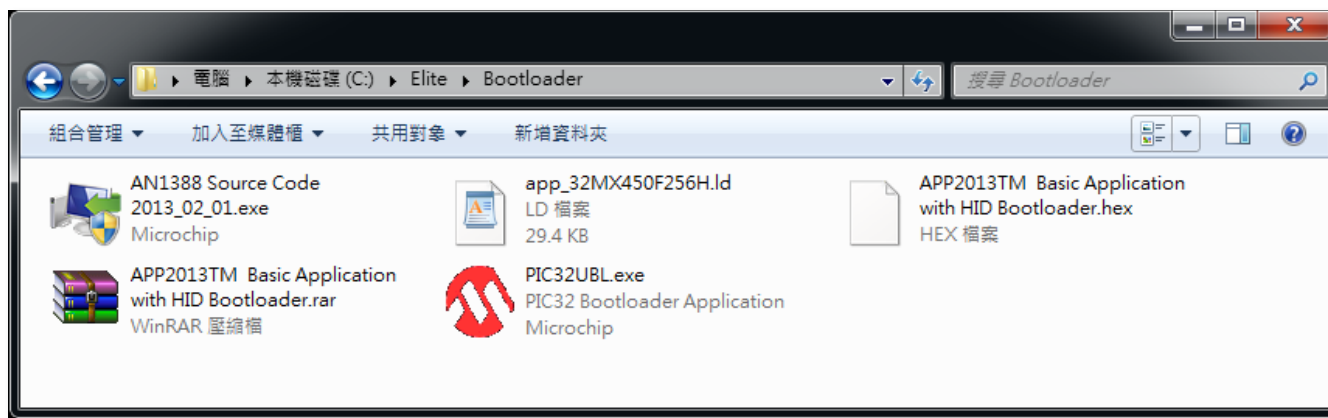
- 為了將Bootloader與User Application的記憶體空間區隔開來。AN1388提供兩個Linker Script File分別給Bootloader跟User Application用。
- Bootloader必須先用使用PICkit3/ICD3/Real ICE等工具燒錄進MCU後, 才可使用Bootloader功能, 此處已經預先將Bootloader程式燒錄完成, 學員無須煩心。
- User Application如果要透過Bootloader燒錄, 則必須透過調整過的Linker Script File(app_32MX450F256H.Id)來編譯程式, 編譯後程式碼才能被配置在正確位址。專案編譯成功後, 再經由專屬的應用程式燒錄Hex File。
- AN1388針對PIC32MX450F256H的記憶體規劃可以參考後續頁面的說明。

Bootloader's Memory Allocation



Lab3 – Programming from Bootloader

- 嘗試利用Bootloader將Lab2的程式燒錄進APP2013TM中。
- 在Lab2中加入Linker Script File, 達成移位功能, 再透過Bootloader所提供的應用程式更新韌體。
- 請先找到資料夾中的
 - PIC32UBL.exe : PIC32 Bootloader 的應用程式。
 - app_32MX450F256H.ld : 針對PIC32MX450F256H User Application所設計的Linker Script File。

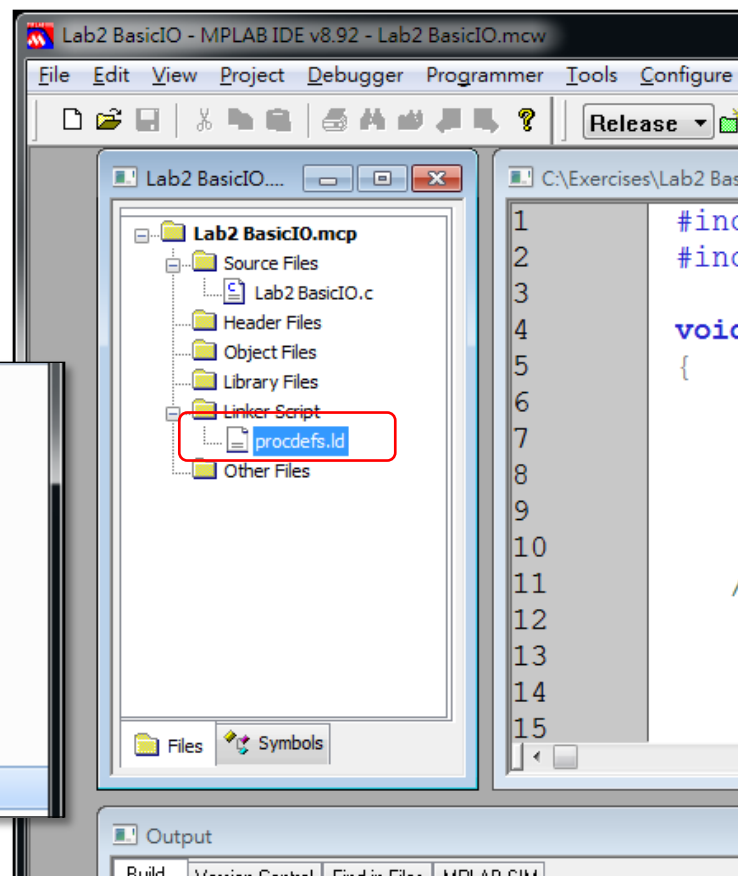
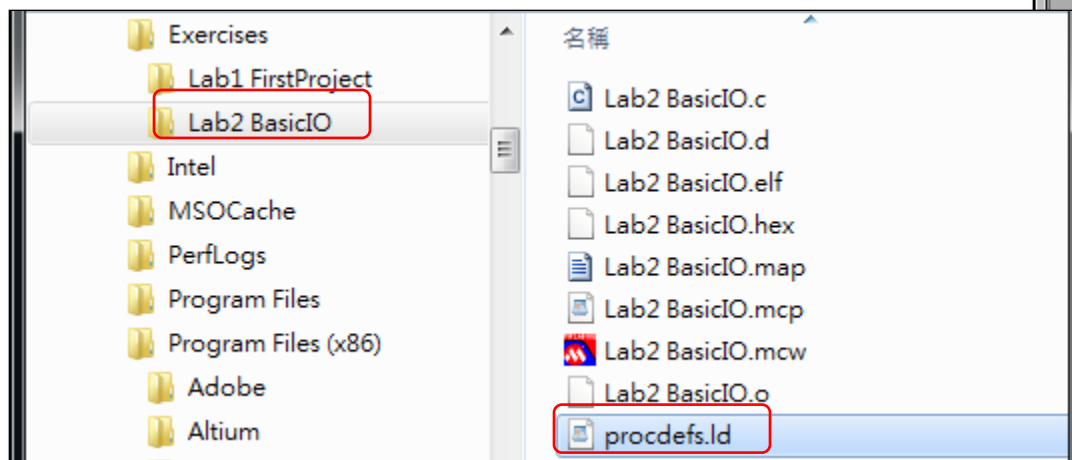


Lab3 – Programming from Bootloader

Step1

- 程式要如何修改才能調整記憶體規劃？

將設計好的Linker Script File, 複製到Lab2的所在資料夾中。
更改名稱為procdefs.ld, 再重新進行編譯即可。



Lab3 – Programming from Bootloader

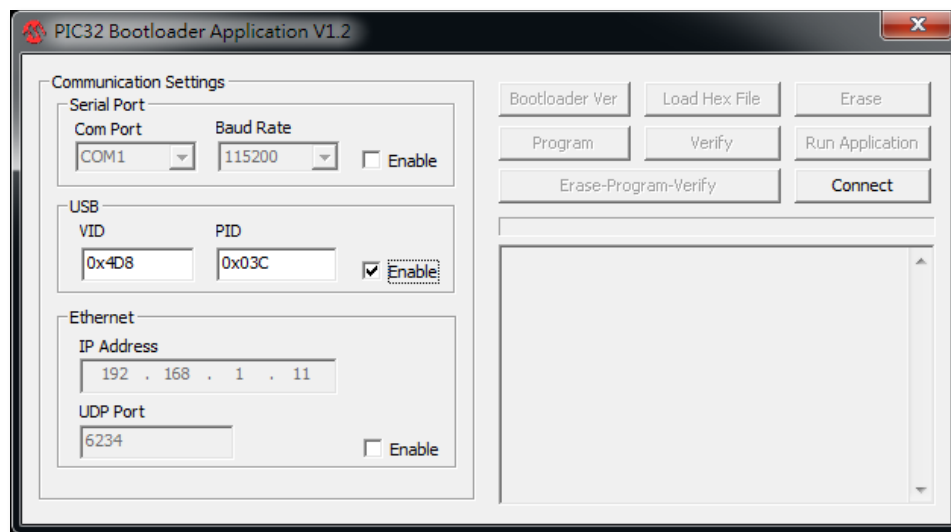
Step2

- 如何確定記憶體的位置有變化？

經過移位後的程式, 已經無法透過軟體模擬驗證。唯一的方法就是直接將程式透過Bootloader來燒錄, 再直接驗證。

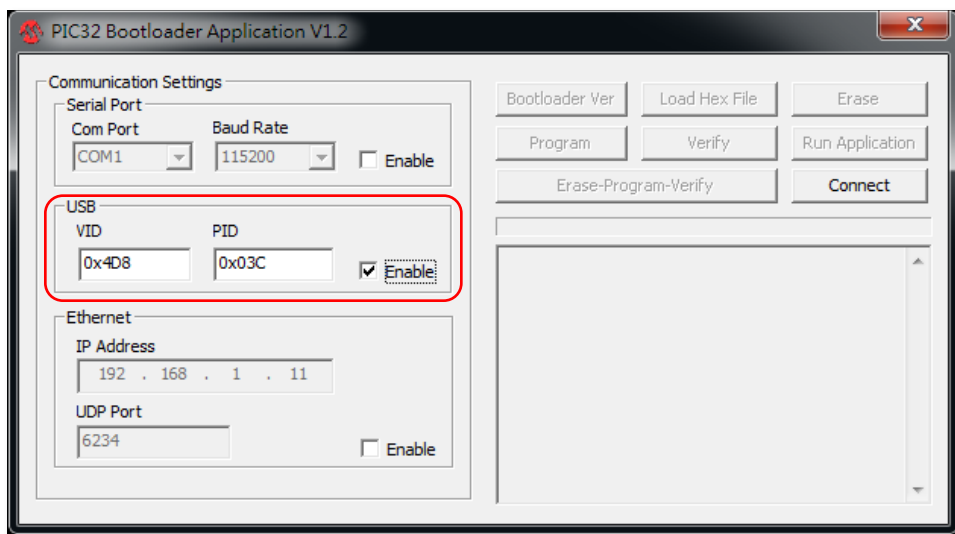
- 如何透過Bootloader燒錄？

執行PIC Bootloader Application, 載入Hex Files。再進行燒錄。

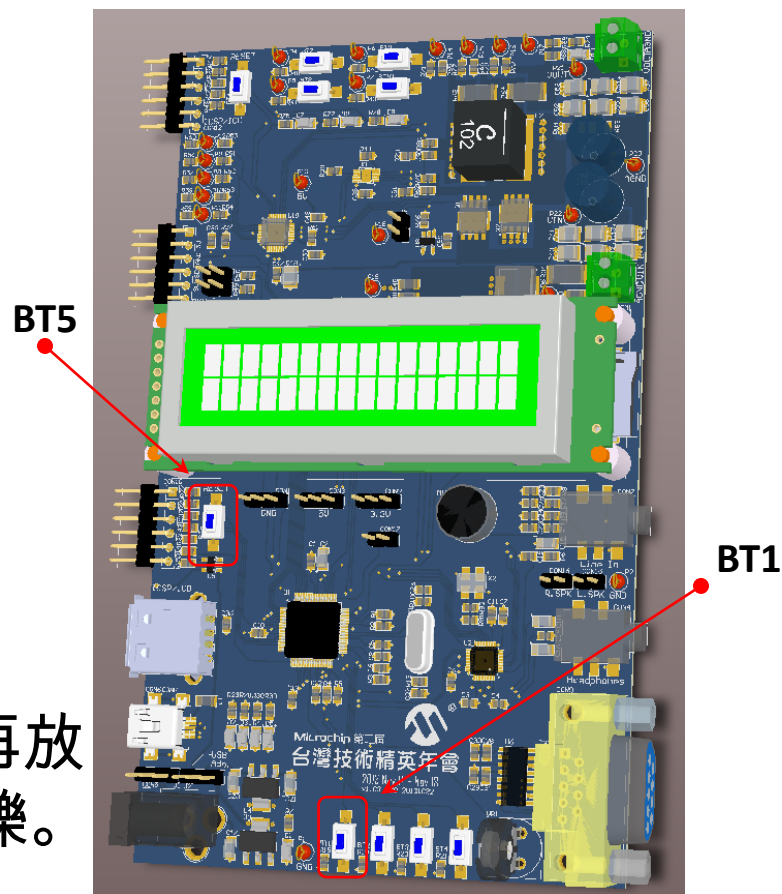


Firmware Update Operation Setp1

- 先找到PIC32UBL.exe, 並且執行該程式。勾選USB Enable。

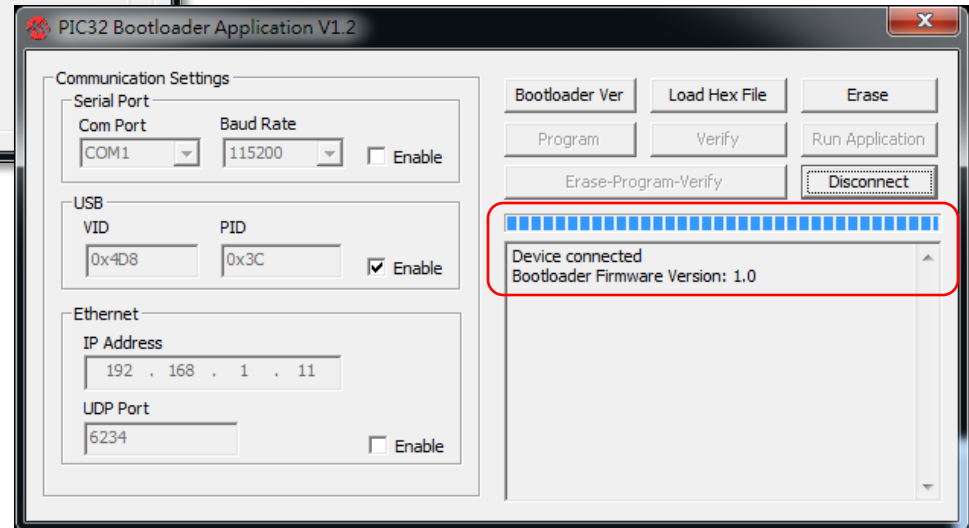
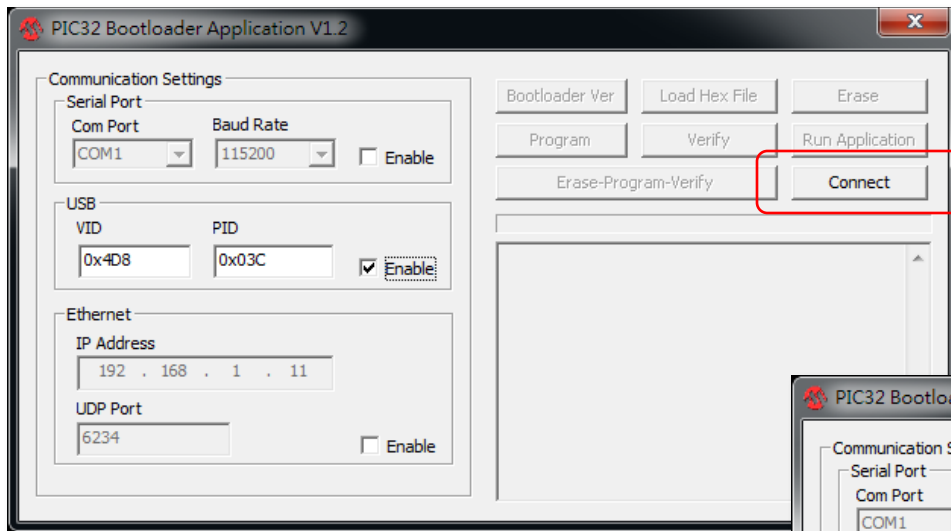


- 切換程式進入Bootloader Mode。
同時按住BT5跟BT1, 接著放開BT5, 再放開BT1。成功的話會看到D2持續閃爍。



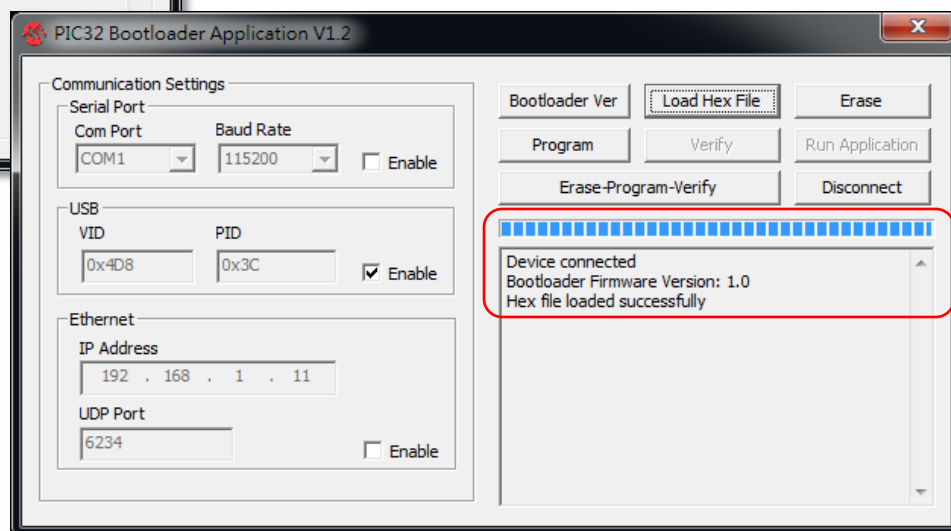
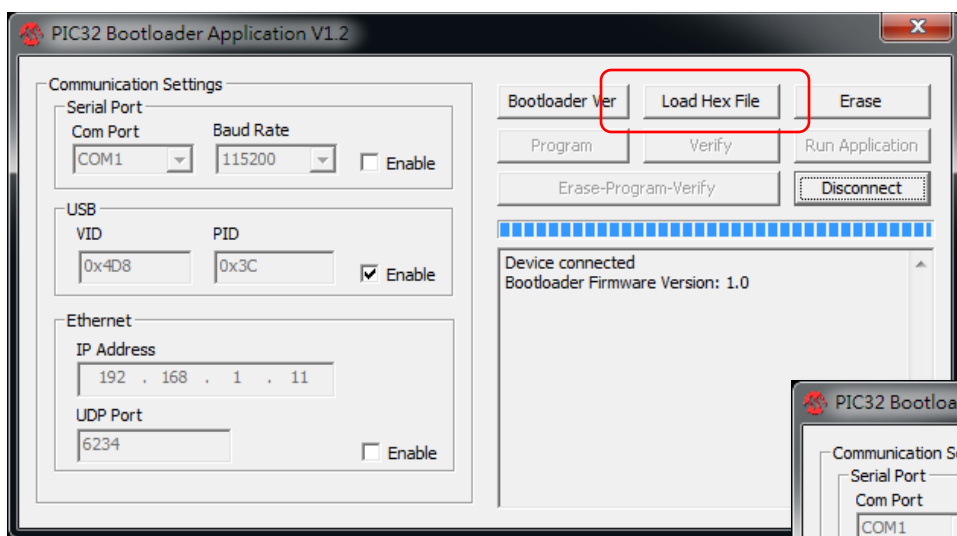
Firmware Update Operation Setp2

- 點選Connect, 會看到連接成功的提示訊息。



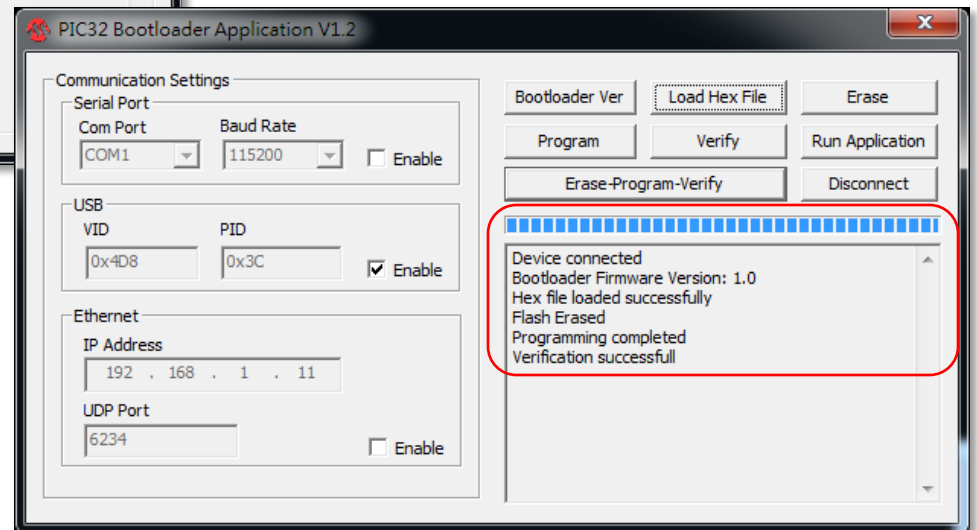
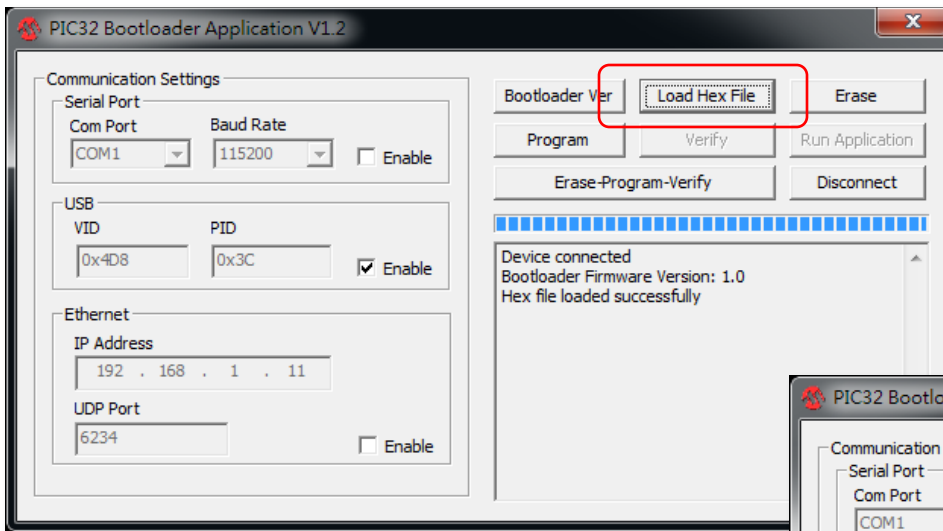
Firmware Update Operation Setp3

- 點選Load Hex File, 載入Hex file。 (Lab2 BasicIO\Lab2 BasicIO.hex)



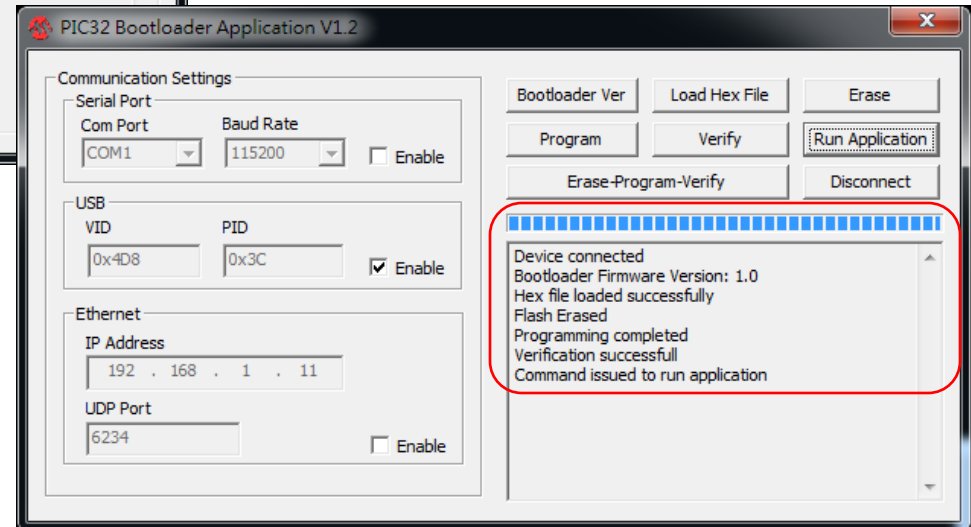
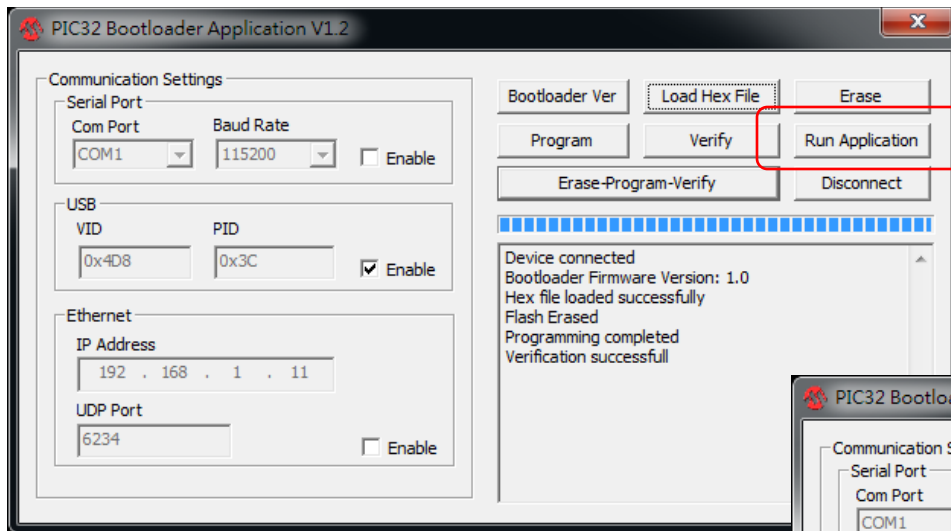
Firmware Update Operation Setp4

- 點選Erase-Program-Verify, 燒錄Hex File。



Firmware Update Operation Setp5

- 點選Run Application or 按一下BT5, 程式就會開始執行了!



Lab3 – Programming from Bootloader

Step3

- 觀察實際的狀態LED(D5)閃爍頻率好像快很多, 軟體模擬的結果應該是1Hz的頻率, 為何會這樣?

因為Bootloader將系統頻率調整到80MHz。

而Lab2模擬時是用8MHz, 所以速度快了10倍!

- 為何會如此及該如何修正?

Oscillator & Configuration Bits章節會詳細說明, 別著急!

但有一點要先記住, 在使用Bootloader的情況下,

Configuration Bits是以Bootloader的設定為準。