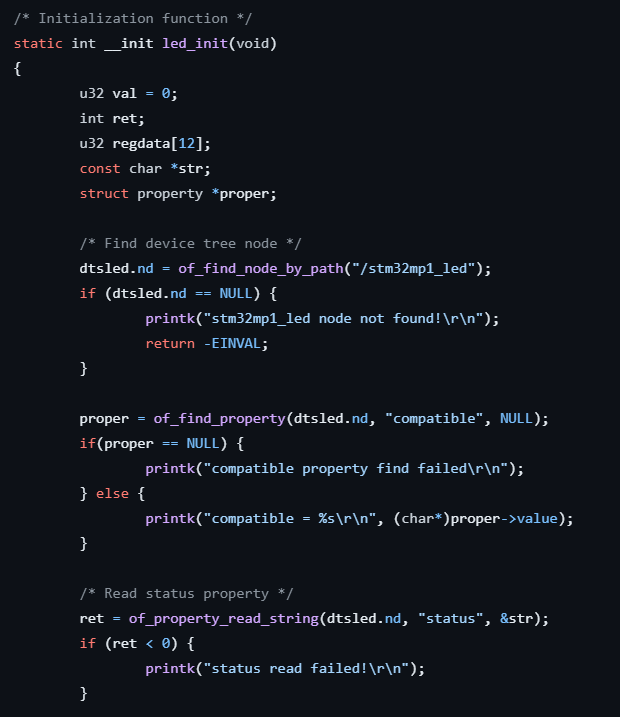
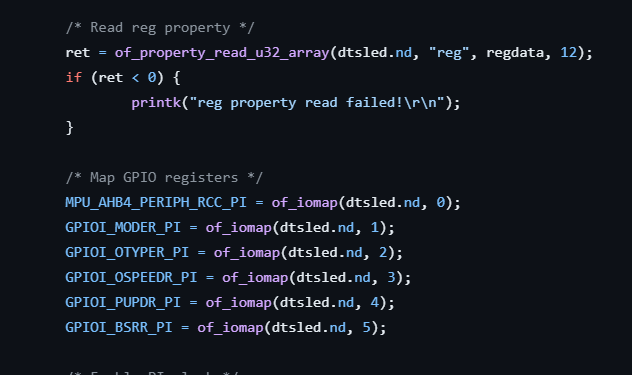
1. 專案04\_dtsled:
   1. Device Tree:由於Linux系統式開源的作業系統，所以ARM每當推出新的芯片時都會添加自己的芯片相關設備的.c和.h檔案，不僅造成整個linux source code虛胖，而且對於Kernel Developer來說是無用的訊息。因此引進了 Device Tree的概念。(功能類似USB中的descriptor)。
   2. DTS、DTB和DTC的關係:

DTS是設備樹源碼文件，DTB是將DTS編譯以後得到的二進制文件。將.c文件編譯為.o需要用到gcc編譯器，那麼將.dts編譯為.dtb需要用到DTC工具！

* 1. 設備都是以節點的形式‘掛’到設備樹上的，因此要想獲取這個設備的其他屬性信息，必須先獲取到這個設備的節點。Linux內核使用 device\_node 結構體來描述一個節點，此結構體定義在文件 include/linux/of.h
  2. 若是想在驅動程式中獲取設備節點相關訊息就必須透過of函數，又因為of系列的函數過多，本專案只舉幾個常用的為例。(如下圖一)
  3. 如圖一所示，of\_find\_node\_by\_path 函數通過路徑來查找指定的節點，這裡就是用來查找”stm32mp1\_led”的這個節點。
  4. 如圖一所示，of\_find\_property 函數用於查找指定的屬性，這裡用來查找compatible屬性。檢查Linux Kernel的驅動程式是否支援此設備。
  5. 如圖一所示，of\_property\_read\_string函數用於讀取屬性中的字符串值，此處用於檢查設備是否可操作。
  6. 如圖二所示，of\_property\_read\_u32\_array 函數用於讀取屬性中的數組數據，例如大多數的 reg 屬性都是數組數據，可以使用這個函數一次性讀取出 reg 屬性中的所有數據。此處就是用來讀取reg屬性中的所有資料。
  7. 如圖二所示，of\_iomap 是用來將設備樹中描述的物理地址映射到虛擬地址空間的一個函數。它允許驅動程式通過虛擬地址來訪問硬體設備的寄存器或記憶體。此處就是用來讀取LED屬性中的所有暫存器虛擬地址。



(圖一)



(圖二)

1. 接著後續的工作就如同前面字元設備驅動一樣的流程。完成編寫後就可以驅動LED並進行操作了。

※總結:

此專案最主要是導入Device Tree的觀念以及of函數的應用。

其餘的部分流程及內容都跟字元設備驅動的編寫大同小異。