1. 專案[03\_newchrled](https://github.com/JetWenJW/STM32MP157_Linux_Driver/tree/master/Linux_Driver/03_newchrled)

我們已經掌握了 Linux 字元設備驅動開發的基本步驟。字元設備驅動開發的重點是使用 register\_chrdev 函數註冊字元設備，當不再使用設備的時候，就使用 unregister\_chrdev 函數註銷字元設備。

register\_chrdev 和 unregister\_chrdev 這兩個函數是舊版本驅動使用的函數，現在新的字符設備驅動已經不再使用這兩個函數，而是使用 Linux Kernel推薦的新字符設備驅動 API 函數。

使用 register\_chrdev 函數註冊字符設備的時候，只需要給定一個主設備號即可，但是這樣會帶來兩個問題：

1. 需要我們事先確定好哪些主設備號沒有使用。
2. 會將一個主設備號下的所有次設備號都使用掉，比如現在設置。

解決這兩個問題最好的方法就是在使用設備號的時候向 Linux 內核申請，需要幾個就申請幾個，由 Linux Kernel分配設備可以使用的設備號。這就是我們在開發過程中最常用的方法。

1. 新字元設備的驅動註冊方法:

在 Linux 中使用 cdev 結構體表示一個字符設備，cdev 結構體定義在 include/linux/cdev.h 文件中。

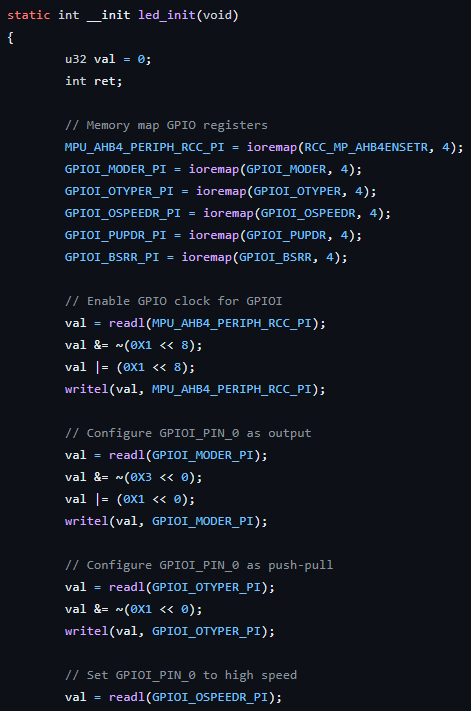
1. 首先以cdev宣告字元設備。
2. 接著，透過cdev\_init()函式對其進行初始化。
3. 最後，透過cdev\_add()函式向Linux Kernel添加此設備
4. 當要卸載模塊或不再使用此設備時可以透過cdev\_del()函式來刪除Linux Kernel中的字元設備。
5. 接著需要在cdev\_add()函式後，自動創建設備節點。那就會用的到class。

首先要創建一個 class 類，class 是個結構體，定義在文件 include/linux/device.h 裡面。

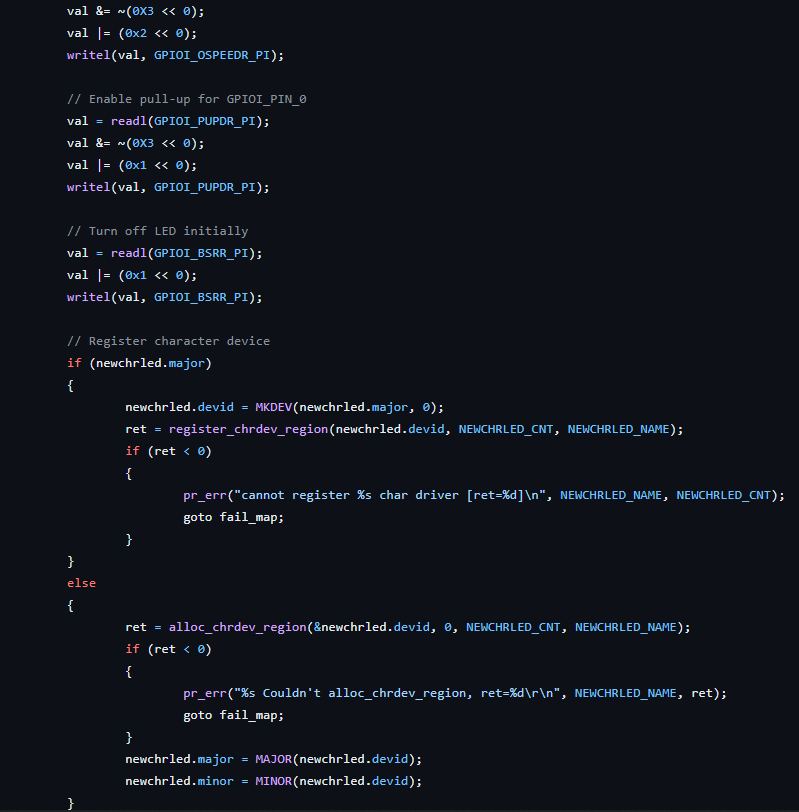
* 透過struct class \*class\_create (struct module \*owner, const char \*name)來創建class。
* 當錯誤發生或卸載模塊時可以透過void class\_destroy(struct class \*cls)來刪除class。

1. 創建好類以後還不能實現自動創建設備節點，我們還需要在這個類下創建一個設備。

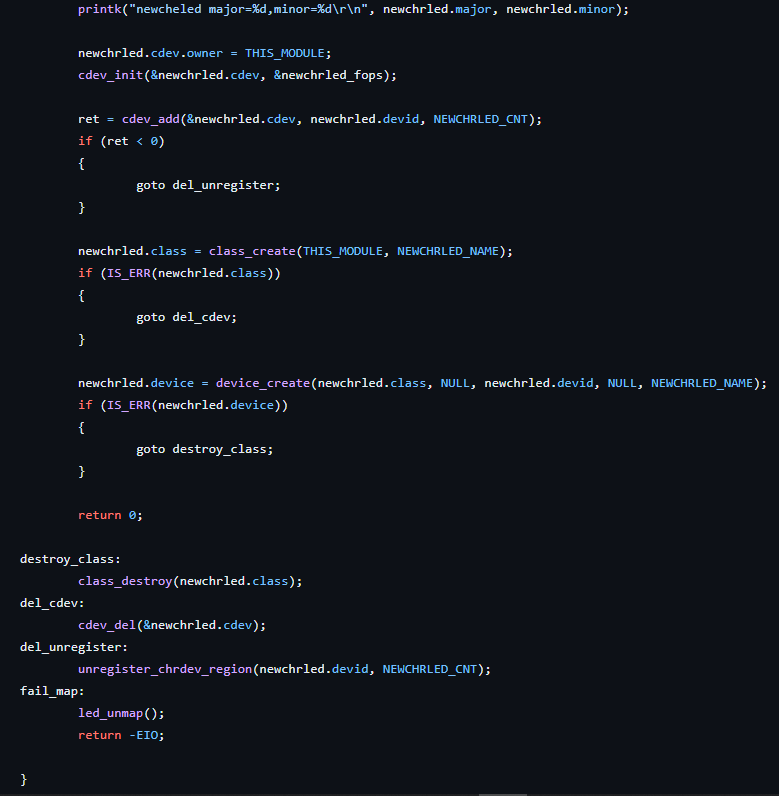
* 可以透過struct device \*device\_create()在class下面創建device。
* 當錯誤發生或卸載模塊時可以透過void device\_destroy來刪除class。



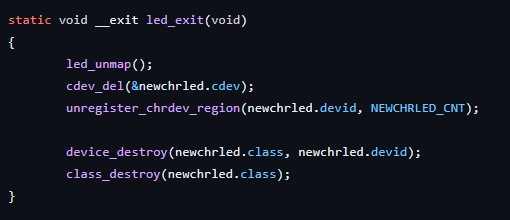
(圖一)



(圖二)



(圖三)



(圖四)

※總結:

圖一到圖三是led\_init()函式實際內容，而圖四是led\_exit()函式實際內容

可以看到led\_init()函式就包含了register\_chrdev\_region()，cdev\_init()，cdev\_add()，class\_create()，device\_create()…等函式使用。且當錯誤發生時則以cdev\_del()，class\_destroy()，unregister\_chedev\_region()做相應處理。

此外觀察圖四在卸載模塊時，也會使用cdev\_del()，class\_destroy()，unregister\_chedev\_region()，device\_destroy()進行操作。