**ES\_Lab1\_report**

組員：B06502141丁冠廷、B06901045曹林熹、B06901188李宗倫

**Before Lab1**

一開始要將程式download到開發板上時，我們遇到了蠻大的麻煩，就是要將程式下載至板子上時，電腦就會顯示ST-link not detected，看起來是有硬體層面的問題。

解決方式：

嘗試數種方法後，我們最後發現是沒有安裝驅動程式的問題，裝了之後程式就可以順利的導入到板子上進行實驗。

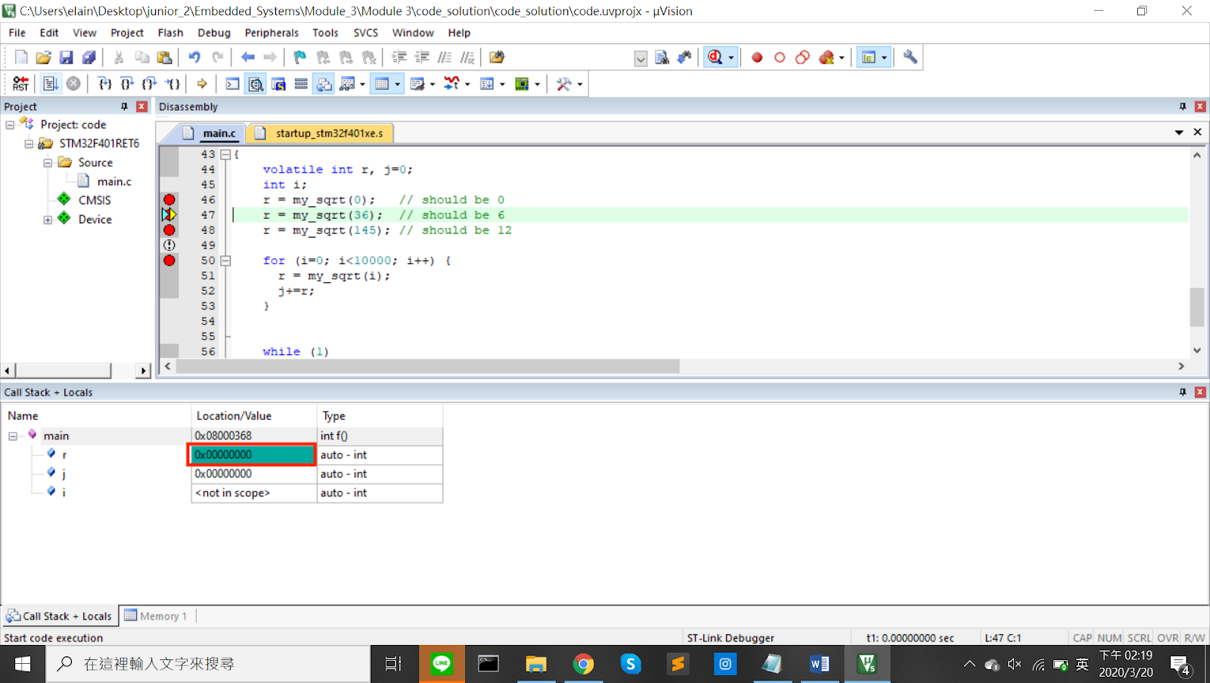
以下是我們安裝的驅動程式連結：

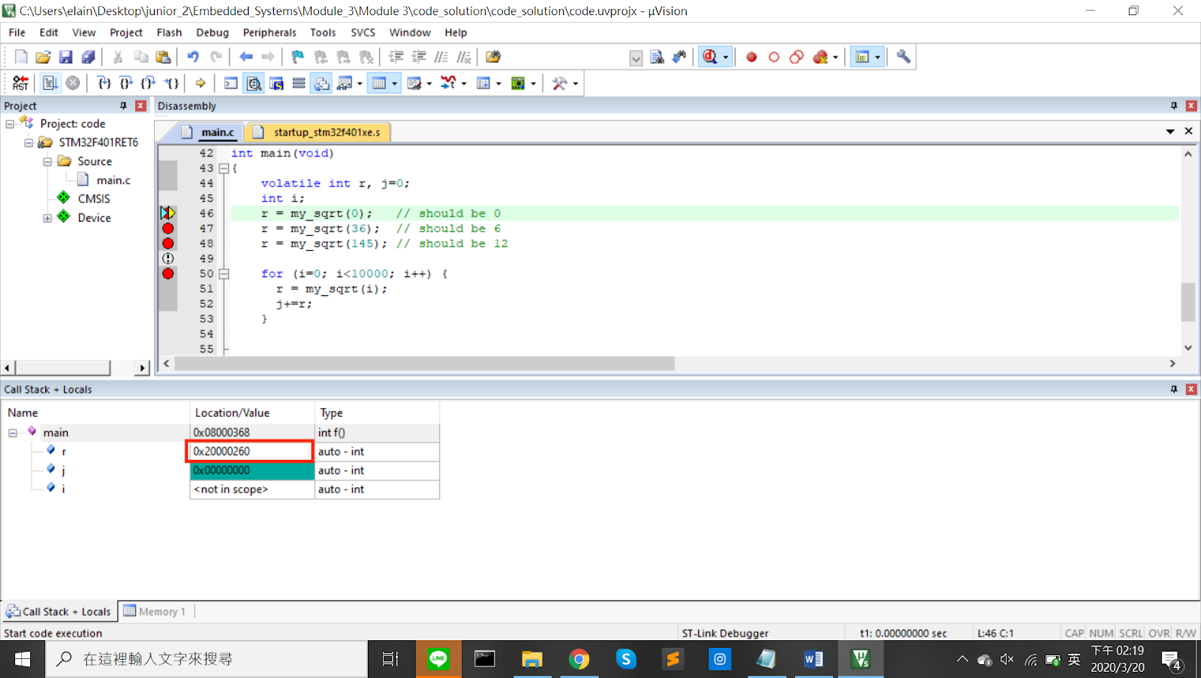
<https://www.st.com/en/development-tools/stsw-link009.html>

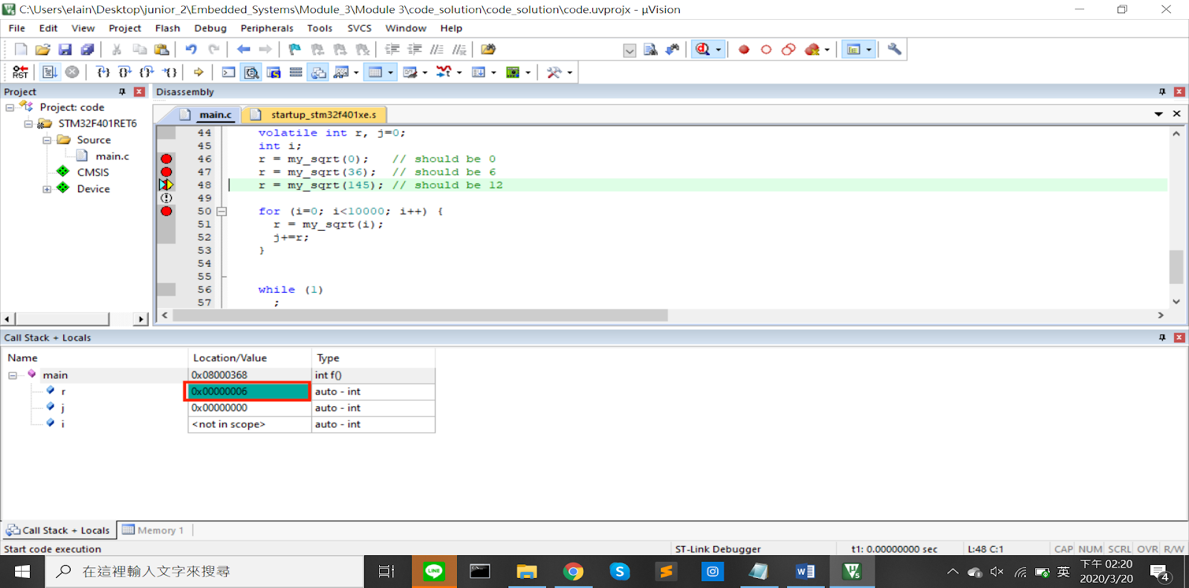
驅動程式官網

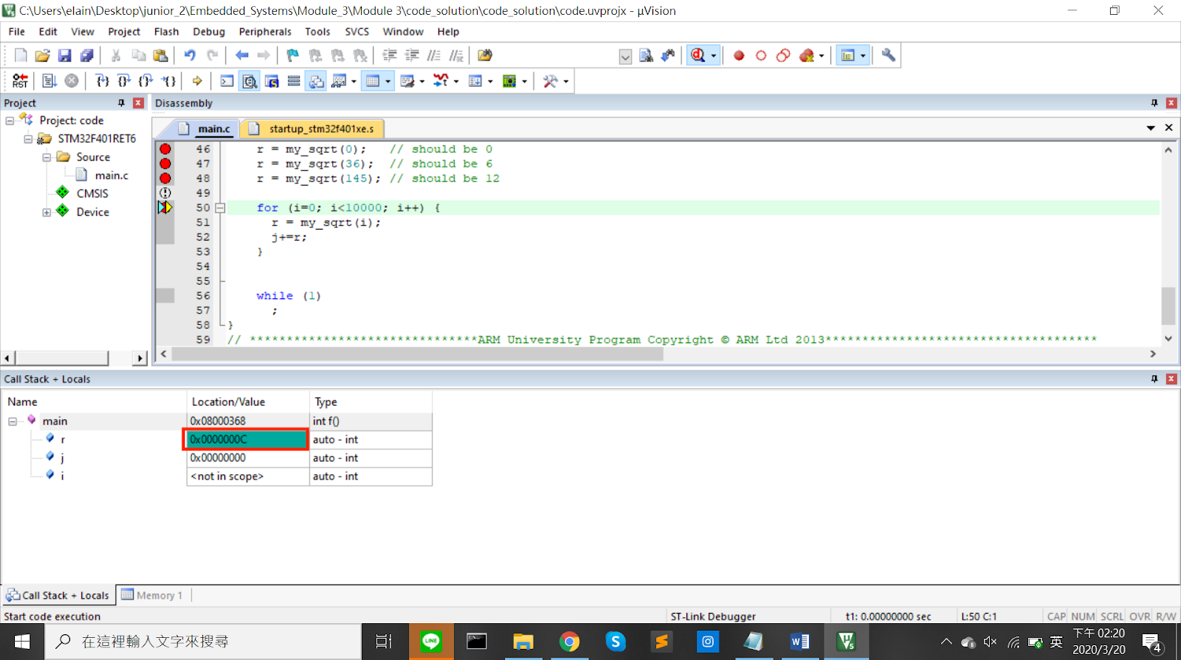
**1. Square Root Approximation**

以下是我們進行第一個實驗的截圖，從圖中我們可以看到，程式將正確的平方根算出來

****

****

****

****

**2. Digital Input/Output and GPIO**

在這個項目中，我們要實踐的功能是按下開發板上的藍色按鈕(user button)就可以讓LED燈發亮，放開按鈕即會熄滅。（若要持續發亮就必須不斷按著）

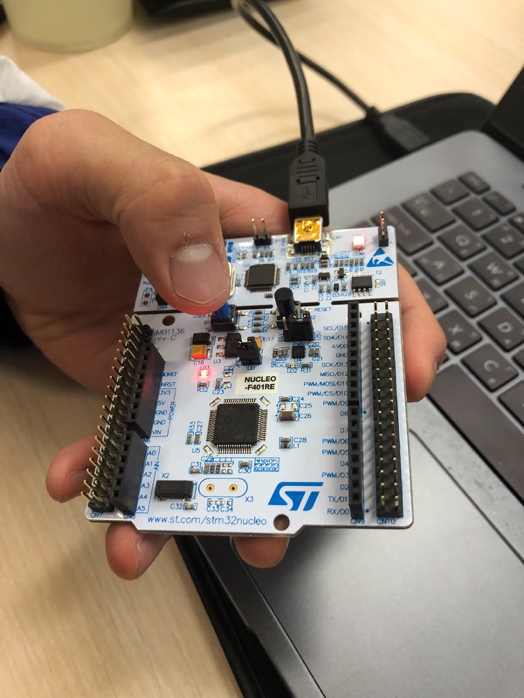
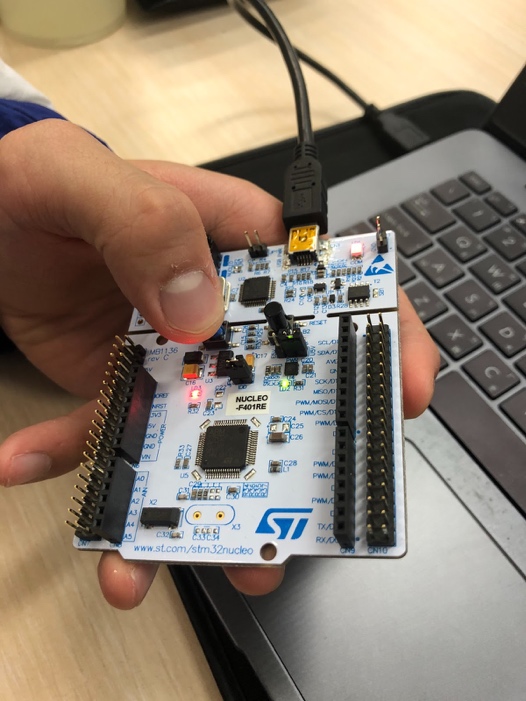
為了讓開發板正確達成這個功能，我們必須執行以下操作：

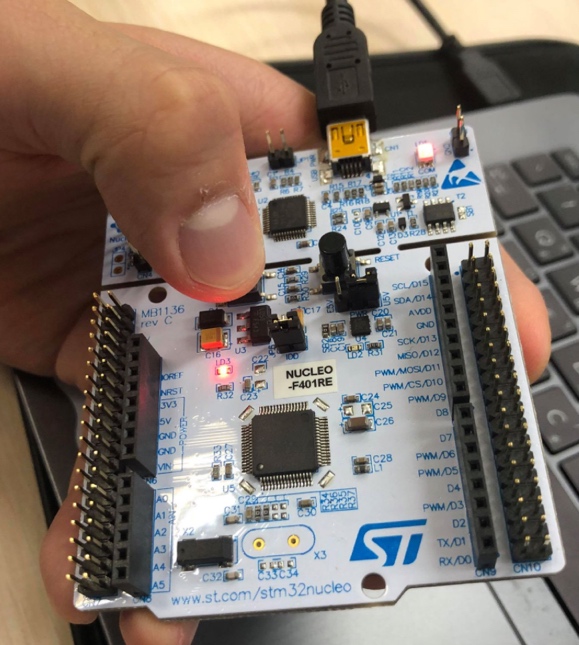
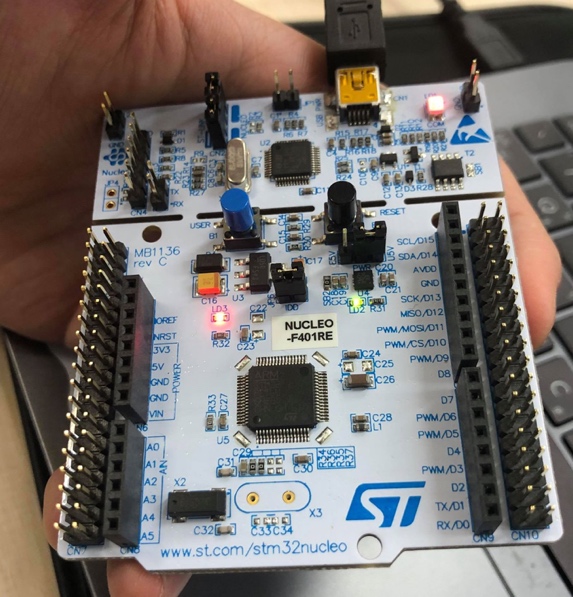
1.初始化user button(GPIO port C, bit 13)。

2.初始化LED(GPIO port A, bit 5)。

3.將操作LED亮、暗所需的設定寫成function。

4.在while(1)無窮迴圈內利用if-else判斷user button是否被按下，若是則亮燈，反之則不亮。

**    **

** **

此外我們也實踐了當按件放開時，燈會亮著，而將按鍵按著時，燈則會熄滅的應用。

**心得**：

這裏使用的語言還在CMSIS-CORE的階層上，所以在使用時需要去閱讀arm所提供的文件，我們才能了解到要控制哪一個輸入或者是輸出，這樣的寫法其實蠻麻煩的，除了更改程式時需要確定輸入輸出外，也比較難上手。

**3. Interrupt and Low Power Features**

接續上一個項目，並做了一些效能改進。在此項目，只要按一次user button後放開，就可以讓LED燈持續亮著，若要熄滅LED燈，則只要再按一次即可。

為了做到此功能，我們必須執行以下操作：

1.初始化user button(GPIO port C, bit 13)。

2.初始化LED(GPIO port A, bit 5)，除了將操作LED亮、暗所需的設定寫成function外，必須再新增一個切換LED現在狀態的function。

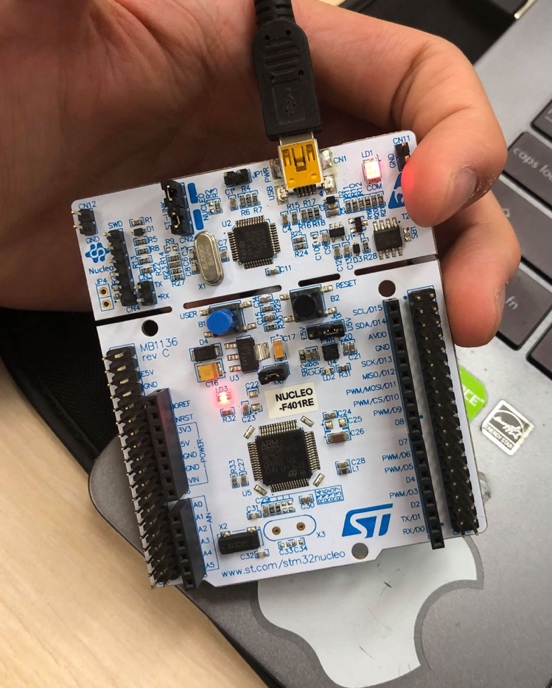
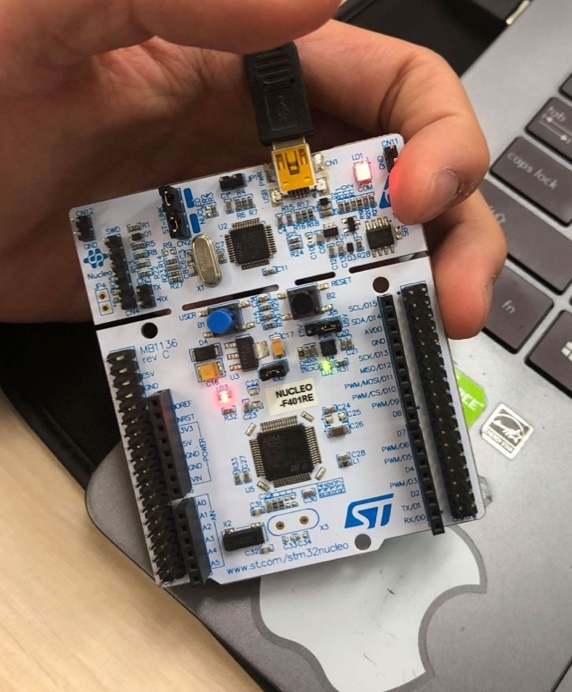
3.**初始化中斷(interrupt)設定**

4.在main.c中完成interrupt handler，設定造成中斷的input，當中斷發生切換LED狀態，清除interrupt status register。

5.在while(1)無窮迴圈中，利用\_\_wfi()函式，讓processor在沒有中斷程序發生時維持睡眠狀態。若有則執行中斷程序，並在結束後回到睡眠狀態。

**心得：**

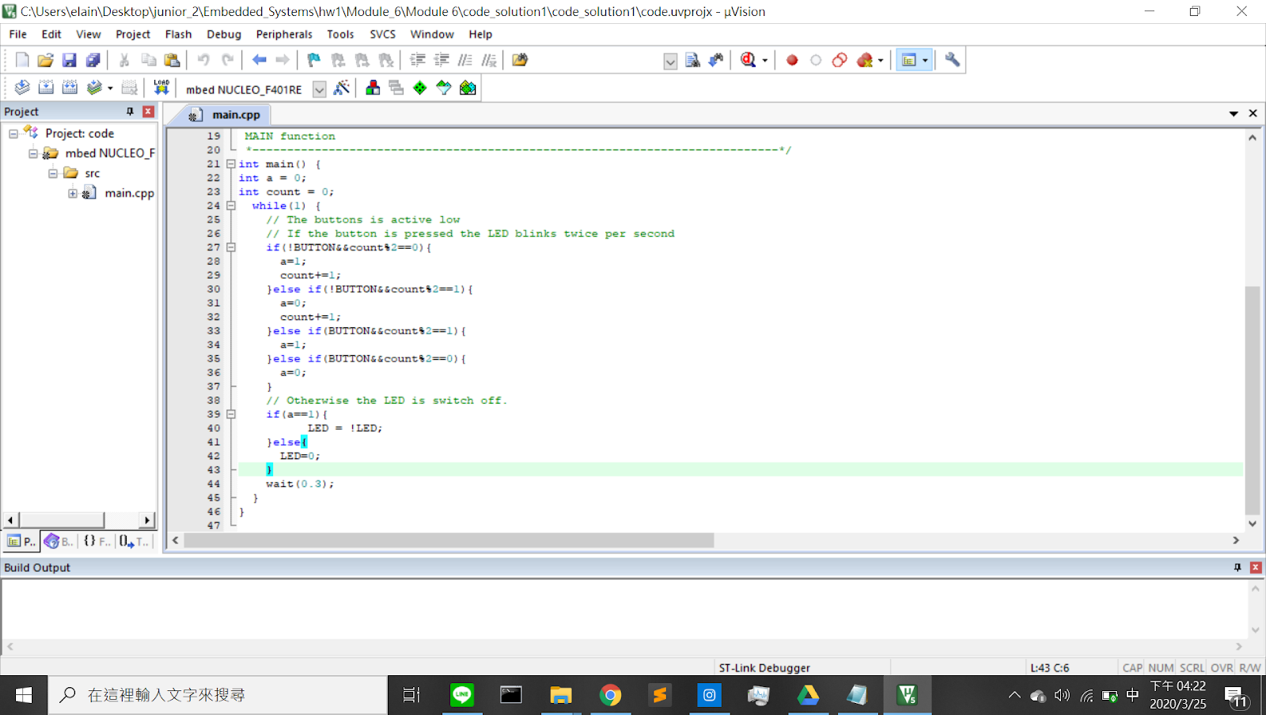
若將此項目與上個項目做比較，可以發現兩者**最大的不同在於功率的消耗**。此項目利用中斷程序的設定讓processor在大多時候能保持睡眠，直到觸發中斷（按下user button），則執行LED狀態改變，爾後又回到睡眠。如此一來，就能顯著減少功率的消耗。

** **

**4. Programming Using mbed API**

延續上一個項目，一樣使用 LED 作為觀察的項目，但這裡使用到更高階的mbed api來實作我們的程式。

這邊我們修改了原先的範例檔，範例檔的應用為手指按壓按鈕則會連續閃爍亮燈，我們修改為按下一次按鈕連續閃爍亮燈，再按一次後則會停止閃爍。

以下將逐步介紹 main function：****

1. 我們首先設定參數 a (判斷按鈕的狀態)以及參數 count (判斷按鈕按下幾次)

初始態 a = 0, count = 0

1. 第一個判斷

進入 while 迴圈，判斷 !BUTTON == true為按下且 count 為偶數，設定此時 a =1，count+=1，因此此時 a == 1，保持發亮。

1. 第二個判斷

判斷 !BUTTON == true為按下且 count 為奇數，此時要從量變成不亮，設定此時 a =0，count+=1，因此此時 a == 0，保持不亮。

1. 第三個判斷

判斷 BUTTON == true為按下且 count 為奇數，此時 LED 狀態是發亮，但是 BUTTON 並沒有指令，因此此時 a == 1，保持發亮。

1. 第四個判斷

判斷 BUTTON == true為按下且 count 為偶數，此時 LED 狀態是不發亮，但是 BUTTON 並沒有指令，因此此時 a == 0，保持不亮。

1. 而最後的 wait(0.3)，代表每 0.3 秒會發亮一次，也可自行調整

發亮頻率

**心得：**

使用mbed api來實作後，有發現真的如教授上課所說，會比較容易上手，其實蠻像從組合語言一直到C/C++更高階層的語言來控制這塊開發版的感覺，但相對的，如果使用越高階的語言，能控制的細節也越會有限，所以這就要看要應用或開發的東西是什麼來決定怎麼寫程式了。