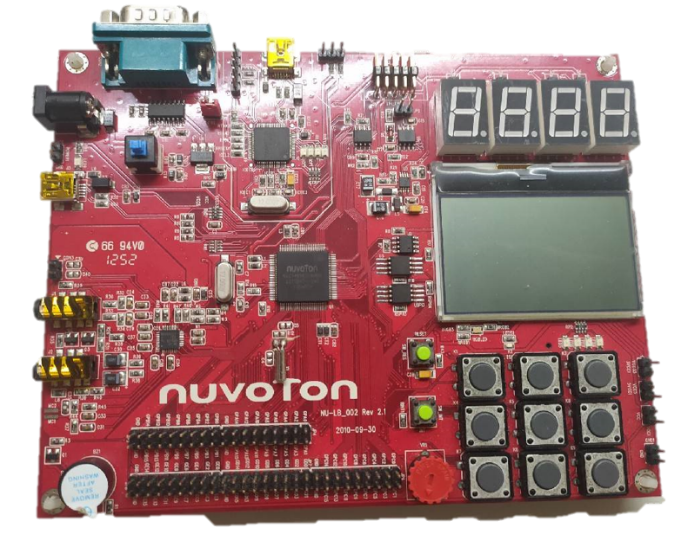
微處理機系統與介面技術 LAB 6

系所：通訊三 學號 :409430030 姓名:翁佳煌

**<實驗器材>**

NUC 140 開發板



PL2303 USB to UART線



ADXL345



**<實驗過程與方法>**

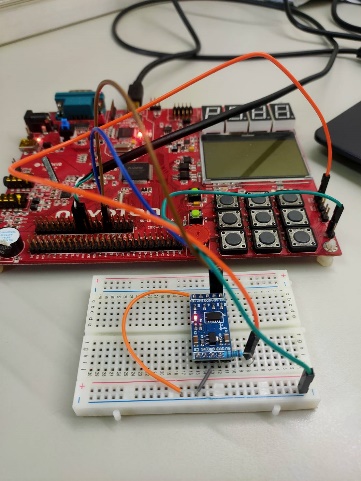
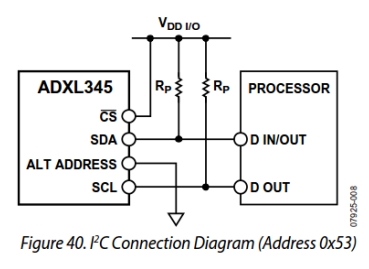
首先，先依照線路圖完成接線，如下**圖1.1**。

• SDA --------- M0 SDA(GPA8)

• SCL --------- M0 SCL(GPA9)• Don’t care: VS, INT1, INT2

• CS, SCL, SDA is already pull high for this module

• SDO is already pull low

****

▲圖1.1

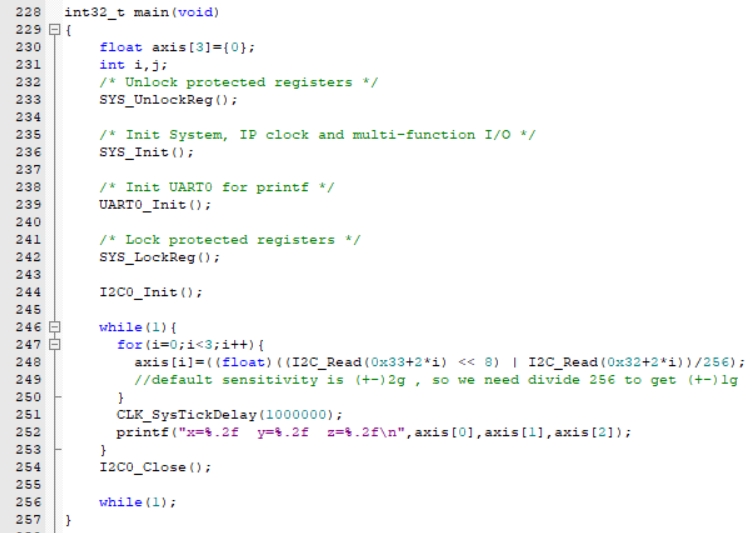
首先看到main函式，第244行為I2C的初始化，等等下半部會詳細描述，而第246行~253行的while迴圈部分是不段讀取ADXL345的x,y,z三軸檢測到的數值，讀取的腳位在ADXL的位置分別為:

• DATAX0(0x32), DATAX1(0x33)

• DATAY0(0x34), DATAY1(0x35)

• DATAZ0(0x36), DATAZ1(0x37)

其中如xaxis = ((DataX1 << 8) | DataX0)的運算是因為輸出數據為two complement，DATAx0為LSB，DATAx1為MSB。



▲圖1.2

再來看到上**圖1.2**第244行呼叫的I2C0\_Init函式，下**圖1.3**開始對ADXL的初始化，第177~179行開啟中斷的設定並決定中斷發生後會進入IRQ Handler，然後在IRQ Handler裡面會呼叫s\_I2C0HandlerFn。而s\_I2C0HandlerFn會接著呼叫I2C\_status函式。

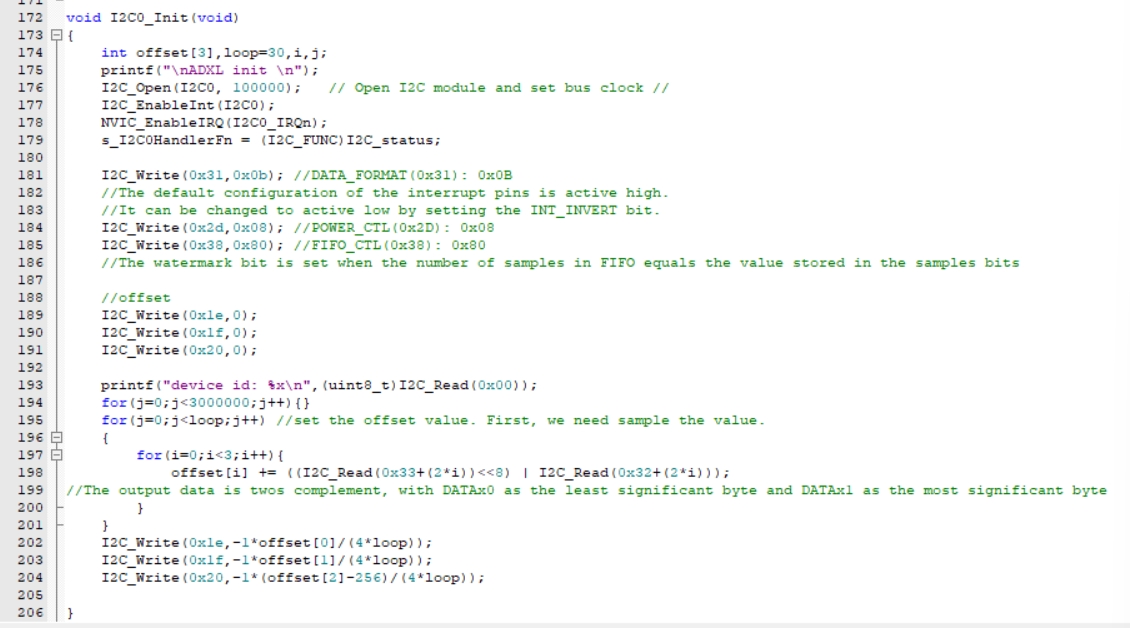
第181~186行為ADXL345的初始化:

• DATA\_FORMAT(0x31): 0x0B

• POWER\_CTL(0x2D): 0x08

• FIFO\_CTL(0x38): 0x80

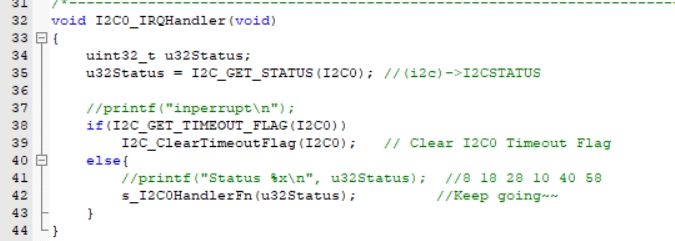
第188~204行則為根據ADXL345的datasheet設定它x,y,z三軸的offset。

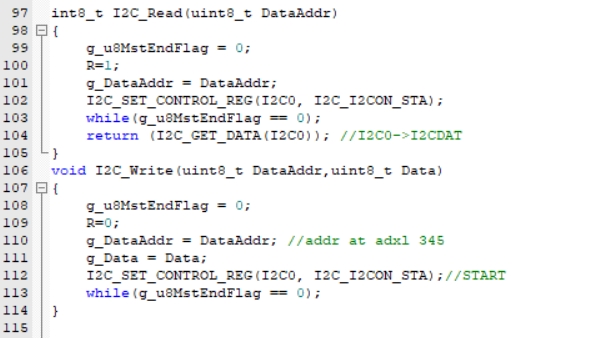


▲圖1.3

接下來看到下**圖1.4**，I2C\_Read和I2C\_Write，設有g\_u8MstEndflag，此變數代表是否終止的Flag，若1則代表終止。

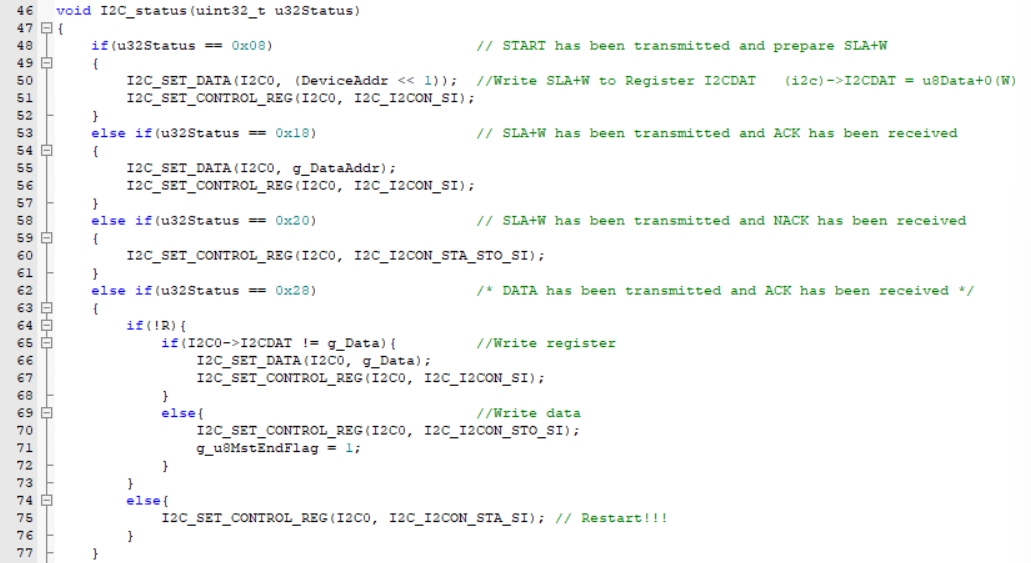
不管事I2C\_Read或是I2C\_Write都會送一個start的訊號，之後會進入中斷，並在第35行利用u32Status得到目前的狀態，第38行判斷是否超時中斷，如果是就清除旗標，如果不是則會在第42行再一次呼叫函示I2C\_status。

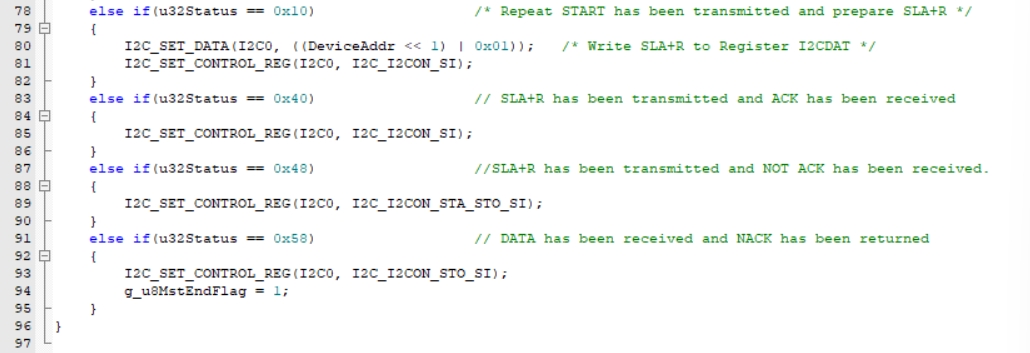


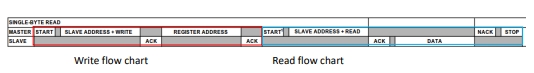
****

▲圖1.4

再來下**圖1.5**則是和sample code一樣，主要再負責判斷當前狀態與相對應要做的事情，依照著最下方圖中的資料格式做傳輸。

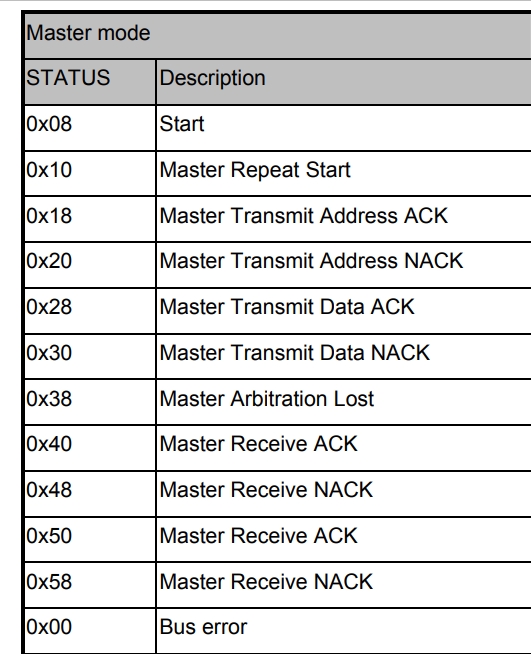
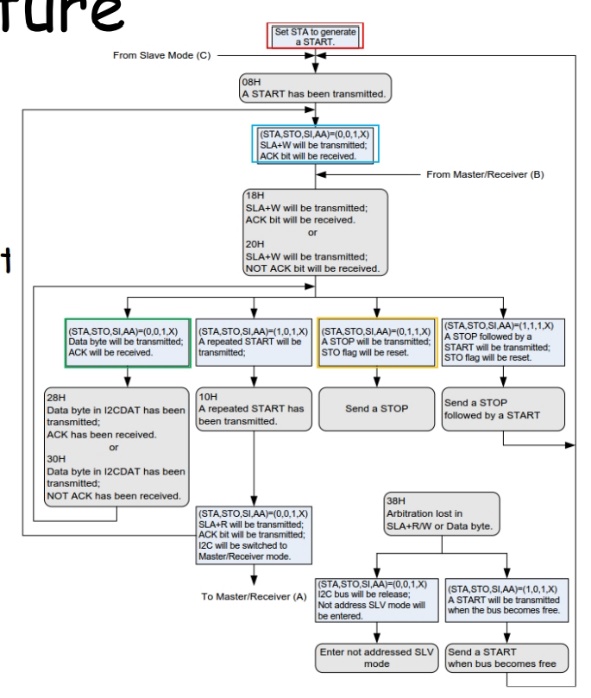






▲圖1.5

下**圖1.6**左邊為對應的狀態流程圖，右邊為datasheet中各個數值代表的狀態和對應的描述。

****

▲圖1.6

**<心得與收穫>**

這次的LAB真的如助教所言特別的困難阿!我連續好幾天嘗試到半夜都無法單獨寫出來，直到最後去問了各個大神才了解CODE如何撰寫，我這才發現雖然明白I2C的原理是很簡單的，只需要2條線就能實現資料傳輸的方便功能，但在實作CODE上真的是燒壞了我的腦袋，這次的實驗雖然不是靠我自己獨立完成的，但大家一起花時間討論出來的結果卻令人雀躍不已。還剩下最後一次LAB為SPI，希望能把握好每次學習的機會，讓自己更加進步。