嵌入式系統實驗

I2C & ADXL345

電機四　謝宜展　B03901166
電機四　高佑豪　B03901136

Experiment 1: I2C

※Modify the communication program for RPiand Arduino using the asynchronous IO techniques using C++ and Python

兩種語言皆是以教授在Asynchronous inputs in Python and C的講義中的範例程式碼去做延伸，我們在RPi額外使用的pin為GPIO 26(pin 37)

Python

在 code 中加入 GPIO.add_event_detect(26, GPIO.BOTH, callback=my_callback, bouncetime=300)來另外開一個 thread 偵測 arduino 的回覆，偵測的 pin 為 GPIO 26，而偵測的變化則為 rise 和 fall 都有。

C

在 C 中則是使用 wiringPiISR(26,INT_EDGE_BOTH,my_callback)來另外開 thread，PIN 一樣為 GPIO 26，偵測的變化也跟前面一樣是 BOTH

Experiment 2: ADXL345

※Read the 3D accelerator value periodically in 200 Hz from an ADXL345 module to RPi using C++ and Python

下列為 ADXL345 的 register 中，我們會用到的部份以及其內容(資料來源為 ADXL345 的 datasheet)

input

Address (Hex)	內容	設置
0x2C	Bandwidth	0000 1011 (0x0C)
0x2D	Powermode	0000 1000 (0x08)
0x31	Data Format	0000 1000 (0x08)

output

Address (Hex)	內容	Data type
0x32, 0x33	X-axis data	以二進位表示，LSB 的單位為 4mg，address 中的前者為較低位數，後者為較高位數。
0x34, 0x35	Y-axis data	
0x36, 0x37	Z-axis data	

Python

Python 中我們使用 smbus 這個 package 來實現對 I2C 的操作，使用以下兩個 function：

1. write_byte_data((<I2C address>, <Register address>, <value>)
2. read_i2c_block_data(<I2C address>, <Register address>, <block size>)

C

在 C 中實現 I2C 的傳輸則比較麻煩。有點像是建立一個存放兩個 char，第一個是 address，第二個則是 value。使用以下兩個 function 來讀寫：

1. write(<file>,<buffer[2]>,<length>);
2. read(<file>,<buffer[]>,<length>);