

Sensor Subsystem

Yuan Hsien Hsu

May 3, 2022

作業

1. 通訊協定分為同步(SPI, I2C)及非同步傳輸(UART)，請說明I2C及UART的通訊原理，同步/非同步傳輸差異以及優缺點
2. 說明致動器(Actuators)中，反應輪(Reaction Wheel)和磁力棒(Magnetorquer)在衛星中的用途
3. 計算題: 有一加速計的可量測範圍介於 $0 \sim +2g$ 之間，若今天設定以10bit儲存加速度資料且“1111111111”對應實際量測值代表 $+2g$ ，當輸出顯示“1001000000”對應實際量測值應該為幾g?
(四捨五入至小數點第三位)

作業

4. 請撰寫I2C程式讓MP3620輸出Z軸角速度資料並顯示在Tera Term上

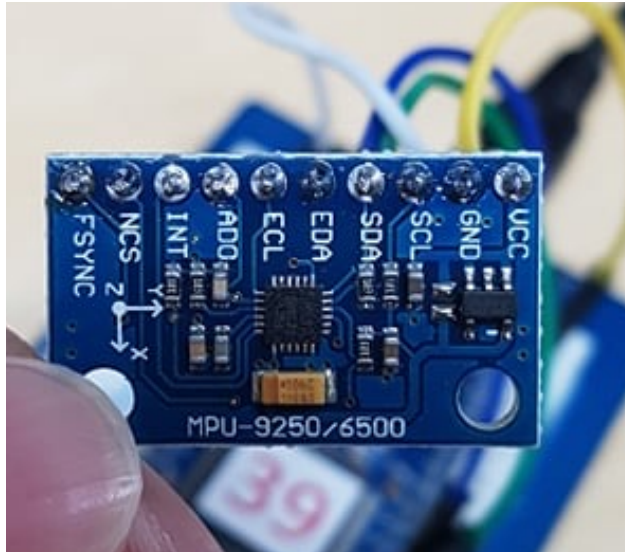
Hint!

Addr (Hex)	Addr (Dec.)	Register Name
47	71	GYRO_ZOUT_H
48	72	GYRO_ZOUT_L

Full scale range: +-250 degree/sec

Storage bit number: 16 bits

9 axis sensor- MPU9250



Power supply: 2.4-3.6V

Protocol: I2C, SPI

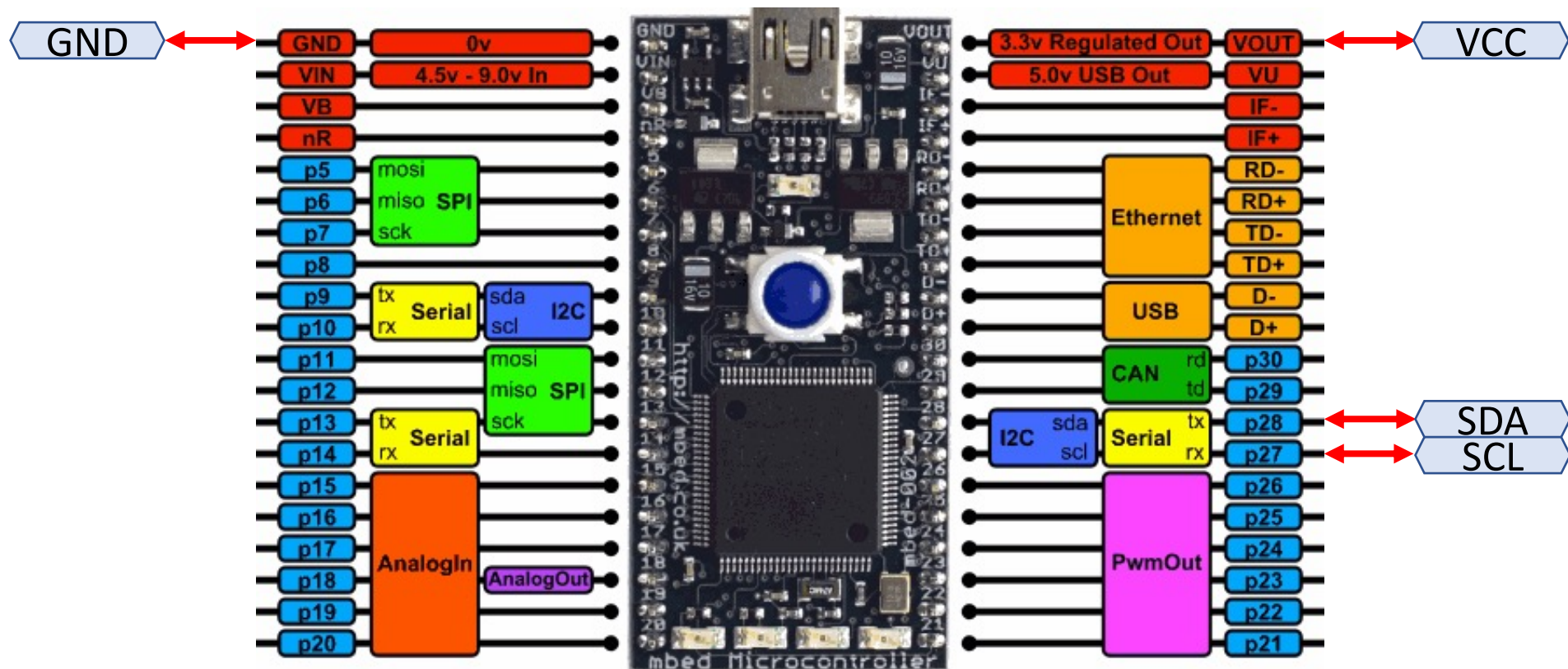
Are able to:

- 3-Axis gyroscope

- 3-Axis accelerometer

- 3-Axis magnetometer

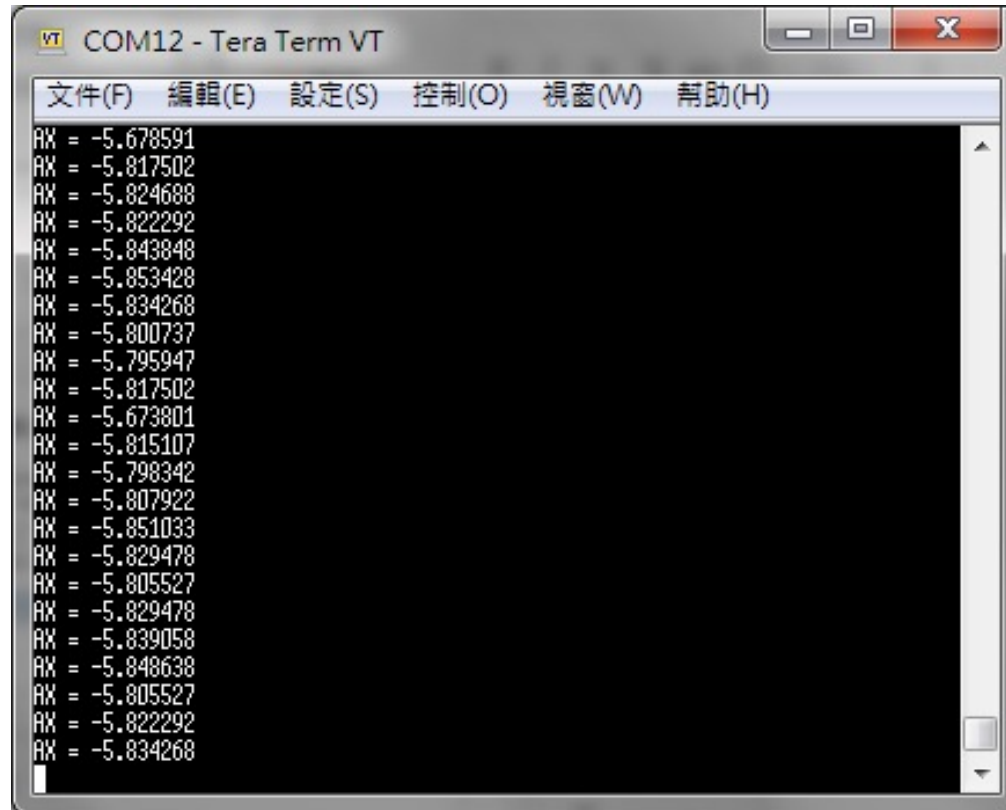
How to connect



範例程式

```
1 #include "mbed.h"
2
3 Serial pc(USBTX,USBRX);
4 I2C i2c(p28, p27);
5
6 const int addr_accel_accel = 0xD0;
7 short int xl, xh;
8 double acc_ax, AX;
9 int main()
10 {
11     while(1) {
12         //Read xh data
13         i2c.start();
14         i2c.write(addr_accel_accel);//write address
15         i2c.write(0x3b);//register address
16         i2c.start();
17         i2c.write(addr_accel_accel|0x01);//read address
18         xh = i2c.read(0);
19         i2c.stop();
20
21         //Read xl data
22         i2c.start();
23         i2c.write(addr_accel_accel);//write address
24         i2c.write(0x3c);//register address
25         i2c.start();
26         i2c.write(addr_accel_accel|0x01);//read address
27         xl = i2c.read(0);
28         i2c.stop();
29
30         acc_ax = short((xh<<8) | (xl));
31         AX = (acc_ax)*4/65535*9.81;
32         pc.printf("AX = %f\r\n",AX);
33         wait(0.5);
34     }
35 }
```

程式輸出



The image shows a screenshot of a Tera Term VT window titled "COM12 - Tera Term VT". The window has a menu bar with the following items: 文件(F), 編輯(E), 設定(S), 控制(O), 視窗(W), 幫助(H). The main area of the window is black and displays a list of 20 lines of text, each starting with "AX =" followed by a numerical value. The values are: -5.678591, -5.817502, -5.824688, -5.822292, -5.843848, -5.853428, -5.834268, -5.800737, -5.795947, -5.817502, -5.673801, -5.815107, -5.798342, -5.807922, -5.851033, -5.829478, -5.805527, -5.829478, -5.839058, -5.848638, -5.805527, -5.822292, and -5.834268. A vertical scrollbar is visible on the right side of the text area.

```
AX = -5.678591
AX = -5.817502
AX = -5.824688
AX = -5.822292
AX = -5.843848
AX = -5.853428
AX = -5.834268
AX = -5.800737
AX = -5.795947
AX = -5.817502
AX = -5.673801
AX = -5.815107
AX = -5.798342
AX = -5.807922
AX = -5.851033
AX = -5.829478
AX = -5.805527
AX = -5.829478
AX = -5.839058
AX = -5.848638
AX = -5.805527
AX = -5.822292
AX = -5.834268
```

範例程式

- Line 1 ~ 8
- Serial pc(USBTX,USBRX); 功能在待會需要將輸出資料匯入電腦進行顯示，須要先定義資料傳輸的「傳輸」使用USB的Tx與「接收」使用USB的Rx。
- I2C i2c(p28,27); 功能在待會需要與IC進行I2C通訊，須要先定義資料傳輸的「資料」使用mbed的pin 28與「時脈」使用mbed的pin 27。
- 第6~9行進行的是變數的定義，使用變數前必須先對變數種類進行定義，此行為代表著要求MCU釋出足夠的記憶體空間供後續對變數的運算。
- 指向 `addr_accel_accel` 的 `0xD0` 指的是 I2C通訊位置(1101000)的16進位表示法 + 0(Write)。

I ² C ADDRESS	AD0 = 0		1101000
	AD0 = 1		1101001

Binary to Heximal

11010000 → 0xD0

範例程式

- Line 9 ~ 19
- `int main(){}` main function 為一支project的主函式，可視為一個程式的入口點。
- `while(){}` 稱為while迴圈，其通式為`while(條件判斷){程式執行}`，意指當條件判斷成立時會執行{}內的程式，反之則離開while迴圈，而條件判斷中的「1」在布林值中代表著True，表示該條件永遠成立，內部程式會無限反覆執行。
- 第13~19行: I2C通訊

Single byte read condition

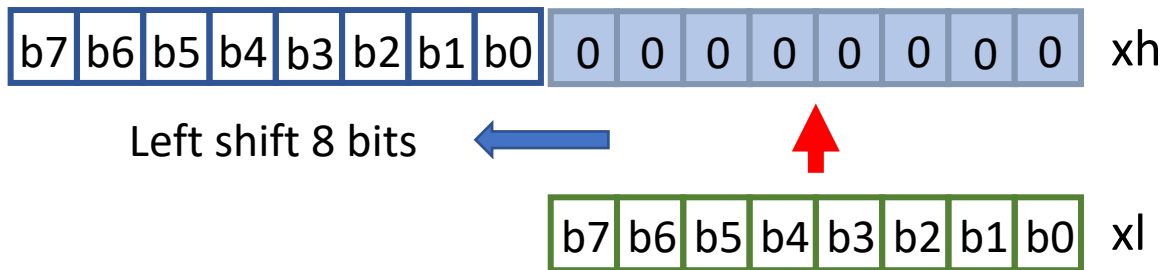
Line	13	14		15	
Master	S	AD+W		RA	
Slave			ACK		ACK
16	17		18		19
S	AD+R			NACK	P
		ACK	DATA		

7.4 I2C Terms

Signal	Description
S	Start Condition: SDA goes from high to low while SCL is high
AD	Slave I ² C address
W	Write bit (0)
R	Read bit (1)
ACK	Acknowledge: SDA line is low while the SCL line is high at the 9 th clock cycle
NACK	Not-Acknowledge: SDA line stays high at the 9 th clock cycle
RA	MPU-9250 internal register address
DATA	Transmit or received data
P	Stop condition: SDA going from low to high while SCL is high

範例程式

- Line 30 ~ 33
- 第30行功能為將收下的xh與xl資料合併，其中「<<」為位元運算子。
- 第31行: 將16位元的資料換算為實際的加速度值。
- 第32行: `printf("AX = %f\r\n", AX);` 在Terminal上印出一串字，其中的「%f」為某個浮點數值，對應「AX」。
- 第33行: 等待0.5秒。



Line 31 example



儲存空間

0~30 實際大小

$(\text{儲存的值}) * \text{實際範圍} / ((2^{\text{儲存空間}}) - 1)$

作業繳交

- 作業於**5/17(二) 23:59**前繳交至信箱 f105601043@g.ncu.edu.tw
- **作業(Group_x作業)**
- 若需借用HeptaSat套件
 - ◆ 一到五可直接到820借用，假日則在禮拜五來借
 - ◆ 表上需填寫借用人及登記