嵌入式軟體

期中作業

MMA7260 加速度感測器的 SIOC 驅動程式開發

生醫所 碩一

993211011 陳柏全

目錄

一、 *MMA 7260 加速度感測器*

i. 感測原理

MMA7260 為 Freescale 半導體元件公司所出產的加速度感測器(G-sensor)，Freescale 的加速度感測器，其感測器單元，為一電容性的加速度感測細胞(G-cell)，它是利用半導體多晶矽材料，以及光罩和蝕刻製程，所製造的一種機械結構，並由彈簧、橫樑材質(Beam Masses)，和固定栓繩結構(Tether)所組成，如圖 1.1。

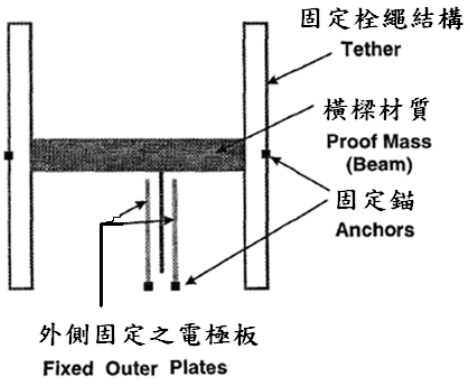


圖 1.1 Freescale 的加速度感測器構造

加速度感測細胞(G-cell)是由一組三個橫樑，所構成的一種機械結構。中間橫樑是可移動的，而兩側橫樑則是固定住。當載具系統有加速度產生時，便可利用中間移動式的橫樑，和兩側固定式橫樑的位移差，計算出加速度值。當系統維持靜止狀態，或處於等速運動時，兩側固定式橫樑栓繩，便會將中間可移動的橫樑，拉至中心位置，這種性質與彈簧的原理相同。

信號放大電路是利用切換電容技術，來量測 G-cell 的電容值，並利用兩個電容間的差值，解算出加速度加速度。信號放大電路對切換電容的輸出訊號，進行訊號條件處理後，再經過低通濾波器，產生一個輸出電壓，基本上此輸出電壓，和載具的加速度成正比。[1]

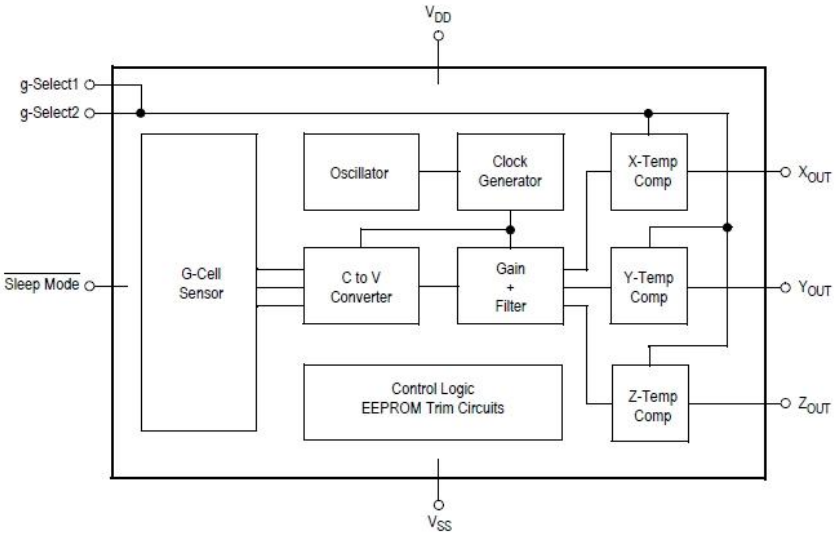


圖 1.2 加速度量測功能簡易方塊圖[2]

ii. 產品規格[2]

A. 特色

- 靈敏度可調 Selectable Sensitivity (1.5g/2g/4g/6g)
- 低工作電流 Low Current Consumption: 500 µA
- 超低功耗睡眠模式 Sleep Mode: 3 µA
- 低工作電壓 Low Voltage Operation: 2.2 V – 3.6 V
- 晶片大小 6mm x 6mm x 1.45mm QFN
- 高靈敏度 High Sensitivity (800 mV/g @ 1.5g)
- 開機快速 Fast Turn On Time
- 內建低通濾波器 Integral Signal Conditioning with Low Pass Filter
- 穩健設計、耐撞擊 Robust Design, High Shocks Survivability
- 無鉛端子設計 Pb-Free Terminations
- 環保包裝 Environmentally Preferred Package
- 價格低廉 Low Cost

B. 規格

Table 1.1 工作特性

Unless otherwise noted: –40°C ≤ T_A ≤ 105°C, 2.2 V ≤ V_{DD} ≤ 3.6 V, Acceleration = 0g, Loaded output[1]

Characteristic	Symbol	Min	Typ	Max	Unit
Operating Range[2]					
Supply Voltage[3]	V_{DD}	2.2	3.3	3.6	V
Supply Current	I_{DD}	—	500	800	µA
Supply Current at Sleep Mode[4]	I_{DD}	—	3.0	10	µA
Operating Temperature Range	T_A	–40	—	+105	°C
Acceleration Range, X-Axis, Y-Axis, Z-Axis					
g-Select1 & 2: 00	g_{FS}	—	±1.5	—	g
g-Select1 & 2: 10	g_{FS}	—	±2.0	—	g
g-Select1 & 2: 01	g_{FS}	—	±4.0	—	g
g-Select1 & 2: 11	g_{FS}	—	±6.0	—	g
Output Signal					
Zero-g (T_A = 25°C, V_{DD} = 3.3 V)[5]	V_{OFF}	1.485	1.65	1.815	V
Zero-g[4]	V_{OFF}, T_A				mg/°C
X-axis		±2.6[6]	±0.6	±3.8[7]	
Y-axis		±5.8[6]	±5.8	±5.9[7]	
Z-axis		±1.0[6]	±0.8	±0.8[7]	
Sensitivity (T_A = 25°C, V_{DD} = 3.3 V)					
1.5g	$S_{1.5g}$	740	800	860	mV/g
2g	S_{2g}	555	600	645	mV/g
4g	S_{4g}	277.5	300	322.5	mV/g
6g	S_{6g}	185	200	215	mV/g
Sensitivity[4]	S, T_A				%/°C
X-axis		±0.02[6]	±0.02	±0.02[7]	
Y-axis		±0.01[6]	±0.01	±0.01[7]	
Z-axis		±0.01[6]	±0.00	±0.01[7]	
Bandwidth Response					
XY	f_{-3dB}	—	350	—	Hz
Z	f_{-3dB}	—	150	—	Hz

二、 *SIOC 之 Drive 設計與主要程式說明*

i. 周邊訊號處理流程

由於 MMA7260 加速度感測器之輸出為類比電壓，故採用 SIOC 內建之 ADC 與 DMA 之功能將加速度值做類比轉數位並利用 VCP 輸出至螢幕上，訊號流程如圖 2.1。

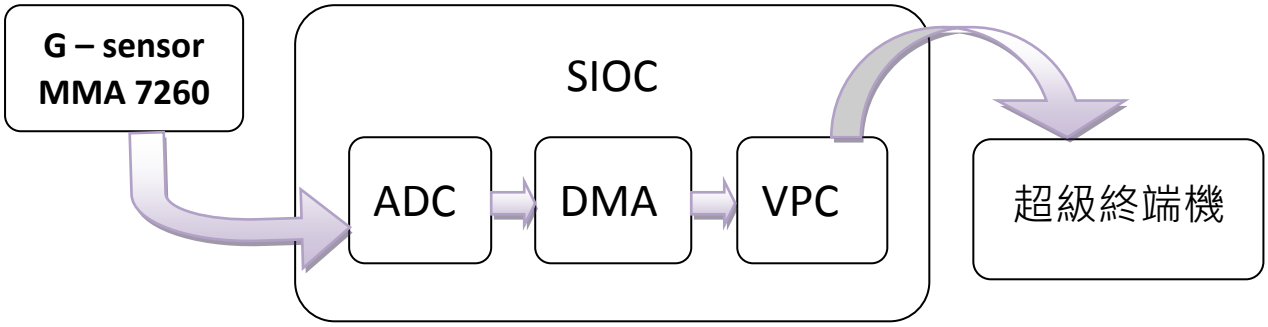


圖 2.1 訊號留程示意圖

ii. Main.c 主要程式說明

由於程式碼全部貼上來太占板面，故完整程式放在附件中，本程式碼是基於課堂中所使用之 ADC 範例程式碼編輯而成，僅於下說明不同之處。

```
/* Private define 部分 -----------------------------------------------------------*/
// vu16 ADCConvertedValue;
// 由於輸出有 x, y, z 三軸，故將轉換後數值改為含有三元素之陣列
vu16 ADCConvertedValue[3];

/*******************************************************************************
* Function Name  : DMA1 chagnel1 Configuration
* Description    : configuration the DMA
* Input          : None
* Output         : None
* Return         : None
*******************************************************************************/
void DMA_Configuration(void)
{
  DMA_InitTypeDef DMA_InitStructure;
  DMA_DeInit(DMA1_Channel1);
  DMA_InitStructure.DMA_PeripheralBaseAddr = ADC1_DR_Address;
  DMA_InitStructure.DMA_MemoryBaseAddr = (u32)&ADCConvertedValue;
```

```
    DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralSRC;
// DMA_InitStructure.DMA_BufferSize = 1;
//  由於 ADC 有三個通道在轉換，也就是說轉換後有三筆數據，故將 buffersize 增加三倍
    DMA_InitStructure.DMA_BufferSize = 3;
    DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
// DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Disable;
//  由於有多筆數據，故記憶體位置需遞增
    DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
    DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
    DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
    DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
    DMA_InitStructure.DMA_Priority = DMA_Priority_High;
    DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
    DMA_Init(DMA1_Channel1, &DMA_InitStructure);

    /* Enable DMA1 channel1 */
    DMA_Cmd(DMA1_Channel1, ENABLE);
}

/*******************************************************************************
* Function Name  : ADC1_Configuration
* Description    : configuration the ADC
* Input          : None
* Output         : None
* Return         : None
*******************************************************************************/

void ADC_Configuration(void)
{
    ADC_InitTypeDef ADC_InitStructure;
    ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
    ADC_InitStructure.ADC_ScanConvMode = ENABLE;
    ADC_InitStructure.ADC_ContinuousConvMode = ENABLE;
    ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
    ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
// ADC_InitStructure.ADC_NbrOfChannel = 1;
//  由於共有三軸需要轉換，故使用三個 channel
    ADC_InitStructure.ADC_NbrOfChannel = 3;
```

```
    ADC_Init(ADC1, &ADC_InitStructure);

    /* ADC1 regular channel 2 3 4 configuration */
// 使用 ADC1 之 Channel 2、Channel 3、Channel 4 來分別轉換 X、Y、Z 軸之加速度數據
// 分別對應到 SIOC 上的第 10~12 之接腳
    ADC_RegularChannelConfig(ADC1, 2, 1, ADC_SampleTime_55Cycles5);
    ADC_RegularChannelConfig(ADC1, 3, 2, ADC_SampleTime_55Cycles5);
    ADC_RegularChannelConfig(ADC1, 4, 3, ADC_SampleTime_55Cycles5);

    /* Enable the temperature sensor and vref internal channel */
    /* ADC_TempSensorVrefintCmd(ENABLE); */
// 將溫度感測器相關功能關閉

    /* Enable ADC1 DMA */
    ADC_DMACmd(ADC1, ENABLE);
    /* Enable ADC1 */
    ADC_Cmd(ADC1, ENABLE);
    /* Enable ADC1 reset calibaration register */
    ADC_ResetCalibration(ADC1);
    /* Check the end of ADC1 reset calibration register */
    while(ADC_GetResetCalibrationStatus(ADC1));
    /* Start ADC1 calibaration */
    ADC_StartCalibration(ADC1);
    /* Check the end of ADC1 calibration */
    while(ADC_GetCalibrationStatus(ADC1));
    /* Start ADC1 Software Conversion */
    ADC_SoftwareStartConvCmd(ADC1, ENABLE);
}

/*******************************************************************************
* Function Name  : GPIO_Configuration
* Description    : Configure the TIM3 Ouput Channels.
* Input          : None
* Output         : None
* Return         : None
*******************************************************************************/
void GPIO_Configuration(void)
{
```

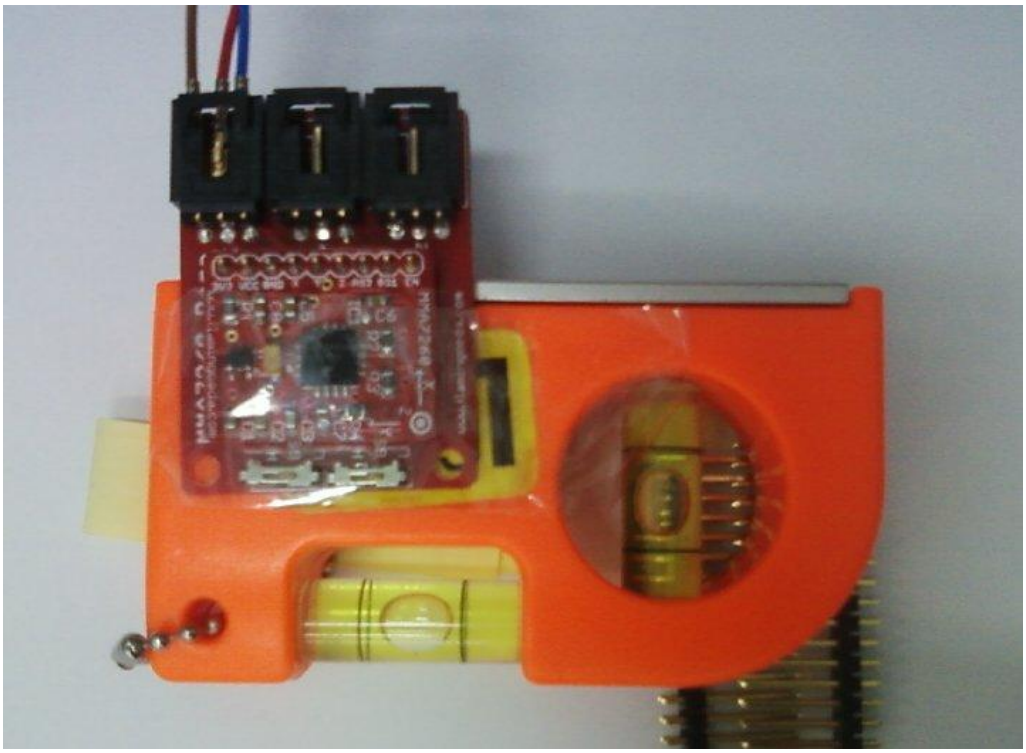
```
    // 將 PA2、PA3、PA4 接腳開啟並初始設定

    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &GPIO_InitStructure);
}
/*******************************************************************************
* Function Name  : Count_Acc
* Description    : Count_Acc
* Input          : None.
* Output         : None
* Return         : None
*******************************************************************************/
void Count_Acc(void)
{
        float acc_x=0,acc_y=0,acc_z=0;
        int x=0, y=0, z=0;

        x = ADCConvertedValue[0];       // 取 X 軸數據
        acc_x = (1.0*x-1700)/1000;      // 將加速度單位轉為 g (9.8m/s2)
        y = ADCConvertedValue[1];       // 取 y 軸數據
        acc_y = (1.0*y-2100)/1000;      // 將加速度單位轉為 g (9.8m/s2)
        z = ADCConvertedValue[2];       // 取 z 軸數據
        acc_z = (1.0*z-2070)/1000;      // 將加速度單位轉為 g (9.8m/s2)
        printf("%.3f, %.3f, %.3f\n\r",acc_x,acc_y,acc_z);
        Delay(100000);
```

} // 校正公式之差異原因為 0 g 時電壓基準值不同

iii. 校正方法

由於 SIOC 之輸出電壓非恰好 3.3V (實測約 2.92V)，且 MMA7260 之使用手冊提到各軸之基準電壓也不盡相同，故在使用前先做簡單校正，方法如下(以 X 軸為例)：

1. 將精準度選擇至 800mV/g
2. 將板子固定在水平儀上並調整至水平，如下圖 2.2



3. 此時輸出應為 0g，記錄轉換後數值為 1700
4. 將水平儀順時鐘旋轉 90 度，並調整至水平
5. 此時輸出應為 1g，記錄轉換後數值為 2700
6. 將水平儀翻轉 180 度，，並調整至水平
7. 此時輸出應為 -1g，記錄轉換後數值為 700

三、 周邊接線圖

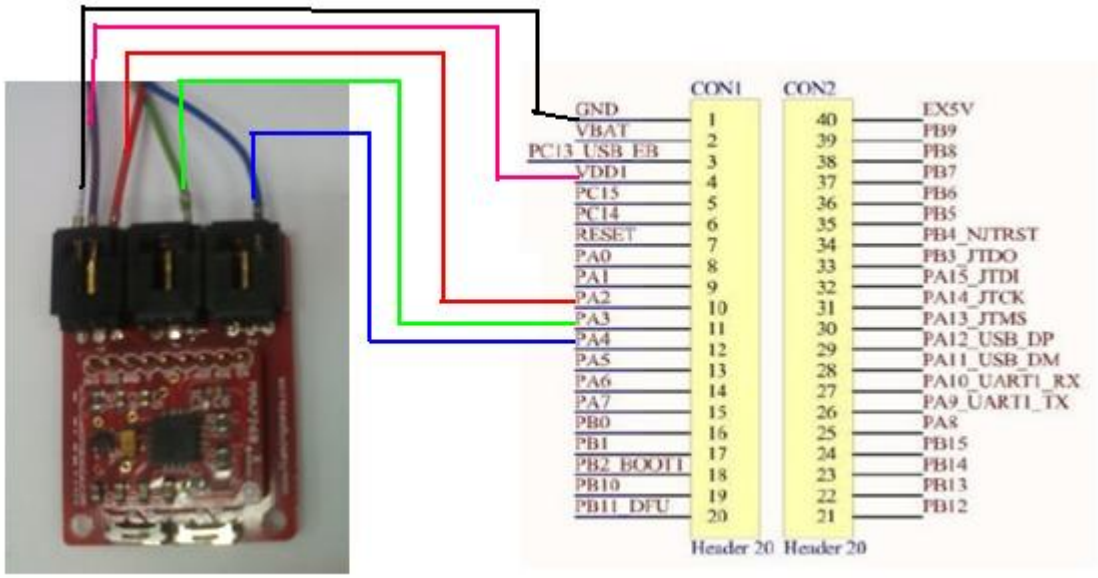


圖 3.1 周邊接線圖

四、 *輸出驗證-使用超級終端機*

沿 X 軸做往復運動，輸出如圖 4.1

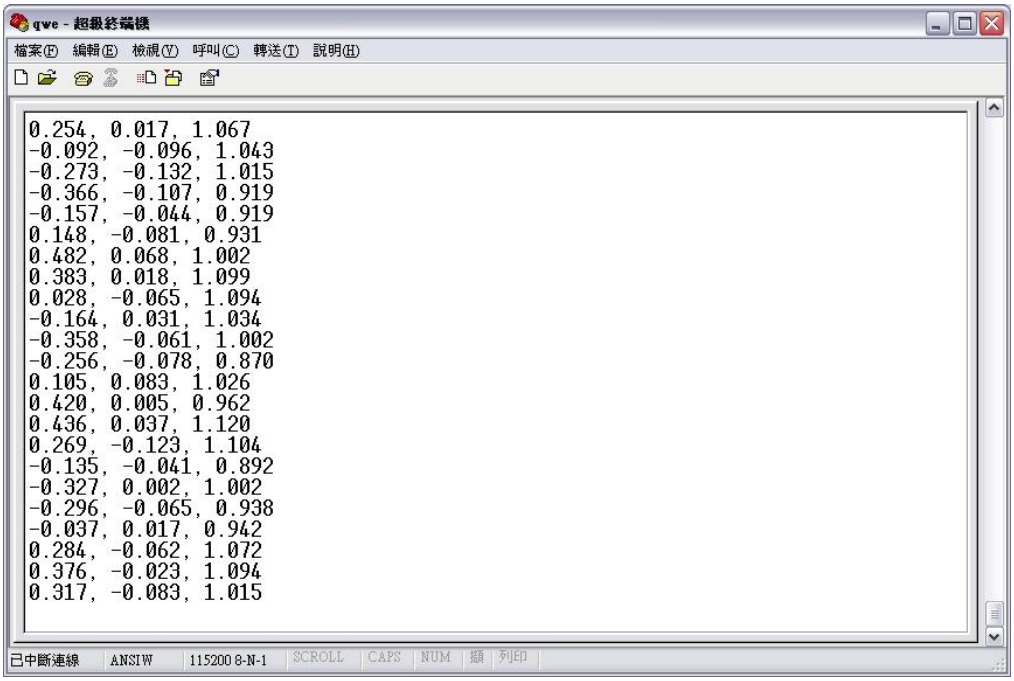


圖 4.1 超級終端機輸出圖

可看出除 X 軸有遞增遞減之現象，其餘兩軸改變甚小，周邊實驗成功。

五、 *參考資料*

[1]「飛思卡爾(Freescale)加速度感測器」技術報告

[2]Freescale Semiconductor MMA 7260 Technical Data