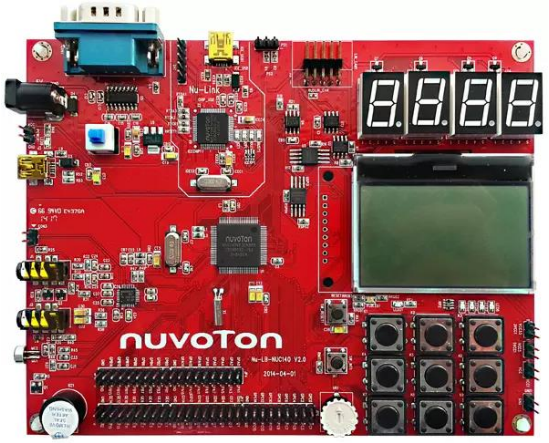
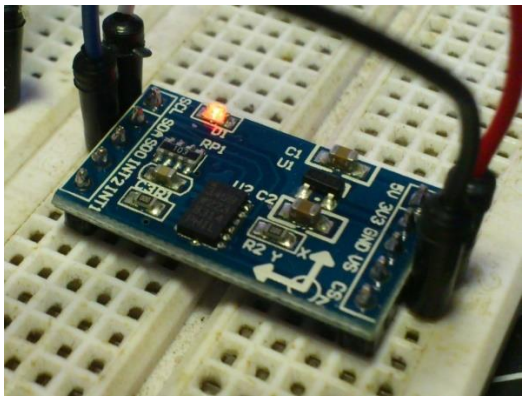
微處理機系統與介面技術 LAB6

系所: 通訊三　　學號:409430043　　姓名: 高司玹

<實驗器材>

Nu-LB-NUC 140 V2.0 開發板　　PL2303TA





<實驗過程與方法>

I2C

這次的實驗主題是 I2C，I2C 在傳輸資訊前要使用一個 7 位元長度的位址空間，還有一位元的 W/R。

有個比較重要的點是，在讀取 adxl 的資料時，是要先 write 再 read，我覺得這是比較需要注意的，一開始 wirte 要讀的 rigister 位置，然後才開始 read 要讀的資訊。

另外有一個重點是在 nuc140 的 datasheet 有提到每當線路中的 status 改變時，則會觸發中斷，因此會在 sample code 內看到一個判斷式傳送或讀取的函式，他是用一個函式指標指著一個函式。

```
    }
    else
    {
        if(s_I2C0HandlerFn != NULL){
            //printf("next state\n");
            s_I2C0HandlerFn(u32Status);
        }
    }
}
```

並在讀取或寫入的時候分別進入不同的動作

```
void I2C_MasterTx(uint32_t u32Status)
{
    if(u32Status == 0x08)                       /* START has been transmitted */
    {
        printf("u32Status == 0x08\n");
        I2C_SET_DATA(I2C0, g_u8DeviceAddr << 1);    /* Write SLA+W to Register I2CDAT */
        I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_SI);    //我猜是清flag
    }
    else if(u32Status == 0x18)                  /* SLA+W has been transmitted and ACK has been received */
    {   printf("u32Status == 0x18\n");
        I2C_SET_DATA(I2C0, g_au8MstTxData[g_u8MstDataLen++]);
        I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_SI);
    }
    else if(u32Status == 0x20)                  /* SLA+W has been transmitted and NACK has been received */
    {
        printf("u32Status == 0x20\n");
        I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_STA_STO_SI);
    }
    else if(u32Status == 0x28)                  /* DATA has been transmitted and ACK has been received */
    {
        printf("u32Status == 0x28\n");
        if(g_u8MstDataLen != data_num_i_want_to_send)
        {
            printf("g_u8MstDataLen != data_num_i_want_to_send\n");
            I2C_SET_DATA(I2C0, g_au8MstTxData[g_u8MstDataLen++]); //給data
            I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_SI);
        }
        else
        {   printf("g_u8MstDataLen = data_num_i_want_to_send\n");
            I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_STO_SI);
            g_u8MstEndFlag = 1;
        }
    }
    else
    {
        /* TO DO */
        printf("Status 0x%x is NOT processed\n", u32Status);
    }
}
```

1. write 的 function

```
void I2C_MasterRx(uint32_t u32Status)
{
    if(u32Status == 0x08)                       /* START has been transmitted and prepare SLA+W */
    {
        //printf("u32Status == 0x08\n");
        I2C_SET_DATA(I2C0, (g_u8DeviceAddr << 1));    /* Write SLA+W to Register I2CDAT */
        I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_SI);     //何時flag被clear???
    }
    else if(u32Status == 0x18)                  /* SLA+W has been transmitted and ACK has been received */
    {
        //printf("u32Status == 0x18\n");
        I2C_SET_DATA(I2C0, g_au8MstTxData[g_u8MstDataLen++]);     //rigister address
        I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_SI);
    }
    else if(u32Status == 0x20)                  /* SLA+W has been transmitted and NACK has been received */
    {
        //printf("u32Status == 0x20\n");
        I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_STA_STO_SI);
    }
    else if(u32Status == 0x28)                  /* DATA has been transmitted and ACK has been received */
    {
        //printf("u32Status == 0x28\n");
        if(g_u8MstDataLen != data_num_i_want_to_send) //原本是 !=2
        {
            //printf("g_u8MstDataLen != data_num_i_want_to_send\n");
            I2C_SET_DATA(I2C0, g_au8MstTxData[g_u8MstDataLen++]);
            I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_SI);
        }
        else
        {
            //printf("g_u8MstDataLen = data_num_i_want_to_send\n");
            I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_STA_SI);
        }
    }
    else if(u32Status == 0x10)                  /* Repeat START has been transmitted and prepare SLA+R */
    {
        //printf("u32Status == 0x10\n");
        I2C_SET_DATA(I2C0, ((g_u8DeviceAddr << 1) | 0x01));   /* Write SLA+R to Register I2CDAT */
        I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_SI);
    }

    else if(u32Status == 0x40)                  /* SLA+R has been transmitted and ACK has been received */
    {
        //printf("u32Status == 0x40\n");
        I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_SI);
    }
    else if(u32Status == 0x58)                  /* DATA has been received and NACK has been returned */
    {
        //printf("u32Status == 0x58\n");
        g_u8MstRxData = (unsigned char) I2C_GET_DATA(I2C0);
        I2C_SET_CONTROL_REG(I2C0, I2C_I2CON_STO_SI);
        g_u8MstEndFlag = 1;
    }
    else
    {
        /* TO DO */
        printf("Status 0x%x is NOT processed\n", u32Status);
    }
```

2. Read 的 funtion

他會不斷的進入 IRQ 並在這些 status 內跑，實現 datasheet 上的 flowchart。

<遇到的問題>

這次整個實驗的 code 一樣都是自己寫的，在整個假日兩天讀完 adxl 和 nuc140 的 datasheet 後，對於整個概念已經沒有什麼問題，比較有的問題是還是沒有很懂為什麼資料最後要進行除 256 這樣的計算。

$$\text{Result} = (\text{Raw data} \pm \text{offset})/(256 \pm \text{offset})$$

<心得與收穫>

這次的 lab 做的雖然很久，但很扎實，完完全全靠著讀文章、datasheet，一點一滴的肝出來，很開心自己的實力又提升了，也學到了很多。