微處理機系統與介面技術 LAB 2

系所: 電機系　　學號: 609415194　　姓名: 劉家豪

<實驗器材>

NUC 140 V2.0 開發板

<實驗過程與方法>

剛開始首先必須 rx 與 tx 對接，剛開始實作時就一度接錯，讓板子毫無反應，那這次功課主要分為兩部分，一個是偵測到 enter(0x0D)才能 output 東西。那較困難的地方也是在這邊，一但這邊做出來後面關於 led 開關就並不那麼困難了。作業的部分主要在做有無觸發 0x0D。

以下是我的 main function code:

```
int main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Init System, peripheral clock and multi-function I/O */
    SYS_Init();

    /* Lock protected registers */
    SYS_LockReg();

    /* Init UART0 for printf and testing */
    UART0_Init();

       //GPIO setting
     GPIO_SetMode(PA, 12, GPIO_PMD_OUTPUT); // initial GPIO pin GPA 12 to output mode
     GPIO_SetMode(PA, 13, GPIO_PMD_OUTPUT); // initial GPIO pin GPA 13 to output mode
     GPIO_SetMode(PA, 14, GPIO_PMD_OUTPUT); // initial GPIO pin GPA 14 to output mode
    /*---------------------------------------------------------------------------------------------------------*/
    /* SAMPLE CODE                                                                                             */
    /*---------------------------------------------------------------------------------------------------------*/

    printf("\n\nCPU @ %dHz\n", SystemCoreClock);

    printf("\n\nUART Sample Program\n");

    /* UART sample function */
    UART_FunctionTest();

    while(1);

}
```

主要是在 set gpio mode 的部分作改動。

那其他 code 的改動主要是在 handle 的部分

```
if(u32IntSts & UART_ISR_RDA_INT_Msk)
{
    if(g_u32comRtail == 0)
    {
        printf("\nInput:");
    }
    /* Get all the input characters */
    while(UART_IS_RX_READY(UART0))
    {
                /* Get the character from UART Buffer */
                u8InChar = UART_READ(UART0);

                //printf("%c ", u8InChar);

        if(u8InChar == '0')
        {
            g_bWait = FALSE;
        }

        /* Check if buffer full */
        if(g_u32comRbytes < RXBUFSIZE)
        {
                if(u8InChar != 0x0D)
                {
            /* Enqueue the character */
            g_u8RecData[g_u32comRtail] = u8InChar;
            g_u32comRtail = (g_u32comRtail == (RXBUFSIZE - 1)) ? 0 : (g_u32comRtail + 1);
            g_u32comRbytes++;
                }
        }
    }
    //printf("\nTransmission Test:");
}
```

這個 part 是 read 的部分，但主要改動是 write 的部分(下一頁)。

```
if(u32IntSts & UART_ISR_THRE_INT_Msk)
{       uint32_t i;

                if(u8InChar == 0x0D)
                {
                    if(g_u32comRhead != g_u32comRtail)
                    {
                        for(i=0;i<g_u32comRtail;i++)
                        {
                        UART_WRITE(UART0, g_u8RecData[i]);
                        }
                    }
                    if ( !strcasecmp(g_u8RecData, "blue on") )// compare
                    {
                        PA12 = 0; // GPA12 = Blue,  0 : on, 1 : off
                    }
                    if ( !strcasecmp(g_u8RecData, "blue off") )// compare
                    {
                        PA12 = 1; // GPA12 = Blue,  0 : on, 1 : off
                    }
                    if ( !strcasecmp(g_u8RecData, "green on") )// compare
                    {
                        PA13 = 0; // GPA12 = Blue,  0 : on, 1 : off
                    }
                    if ( !strcasecmp(g_u8RecData, "green off") )// compare
                    {
                        PA13 = 1; // GPA12 = Blue,  0 : on, 1 : off
                    }
                    if ( !strcasecmp(g_u8RecData, "red on") )// compare
                    {
                        PA14 = 0; // GPA12 = Blue,  0 : on, 1 : off
                    }
                    if ( !strcasecmp(g_u8RecData, "red off") )// compare
                    {
                        PA14 = 1; // GPA12 = Blue,  0 : on, 1 : off
                    }
                g_u32comRtail = 0;
                g_u32comRbytes = 0;

                for(i=0; i<RXBUFSIZE; i++)
                {
                    g_u8RecData[i] = 0;
                }
            }
}
```

那關於 write 的部分主要是著重於當偵測到 0x0D 時去做 write 的動作並對 buffer data 做初始化以便進行下次的資料寫入。

那在 led on 與 off 的部分，主要是做 string compare 那我是引入 string.h 的 strcasecmp 去做比對，但助教說有可能會遇到 bug 所以這個部份我在空閒時會再做修正。

<心得與收穫>

在這次的實作中，我聊解了所謂的 uart 裡包了什麼涵式，也充分地了解中斷以及 uart 如何的利用 rx 以及 tx 進行傳輸，雖然中間遇到了一些不該式瓶頸的瓶頸 XD，但是整體而言收穫很多。