

Forum: [8-bit PIC](#)

Topic: [pic16f690 ADC UART](#)

Subject: Re: [pic16f690 ADC UART](#)

作者: Ryang

2020年05月15日 11:22:06

基本上 ADC 讀取到的值是 Hex 的格式，需要再轉成 BCD（十進制的格式）後再轉成 ASCII Code 後經 UART 送出。

前面的轉碼比較辛苦一點，你可以參考一下教育訓練光碟下-- 8-bit MCU 教材下的：W301 Advance PICC Application

"Lab6 顯示輸入電壓值" 這個章節及範例參考看看是怎樣轉換的。

底下為其中一段讀取 ADC 值再轉成 10 進制的範例。

程式裡顯示電壓值到 mV 是怎樣做到的？

$Disp_Temp = (((unsigned\ long)\ Temp_Buff) * 4883) / 100;$

5V / 1024 = 4.883 mV，要顯示 4.9mV 的精度

不要使用浮點數運算，只要用整數運算即可

```
void Show_ADC_Voltage(void )
{
    //          Disp_Count = 20;                // 設定每 100mS 顯示電壓一次
    Clear_LCD( );

    do
    {
        Disp_Count==0)                // 100mS 到了嗎？本程式會每 100mS 更新 LCD 上的 VR2 電
壓一次
    {
        Read_ADC();                    // 讀取 ADC (VR2) 的轉換值
        Temp_Buff= ADC_Result.Word & 0x3FF;    // 將無用位元遮銷 0
        Disp_Temp=(((unsigned long)Temp_Buff)*4883)/100;    // 10-bit 解析度, 1
LSB=4.883mV

        DisplayDigit1(Disp_Temp/10000);    // 顯示 V LCD 第一數字上
        Disp_Temp=Disp_Temp%10000;        // 取出餘數

        DisplayDigit2(Disp_Temp/1000 );    // 顯示 100mV
        Disp_Temp=Disp_Temp%1000;

        DisplayDigit3(Disp_Temp/100);    // 顯示 10mV
        Disp_Temp=Disp_Temp%100;
```

```

    DisplayDigit4(Disp_Temp/10);           // 顯示 mV
    Disp_Temp=Disp_Temp%10;

    DisplayDigit5(Disp_Temp);             // 顯示 0.1mV

    SEG_4DP=1 ;                           // 點小數點

    SEG_m = !SEG_m;                        // 顯示 mV 的符號
    SEG_V= !SEG_V;
//          SEG_BAT=!SEG_BAT;

    Disp_Count = 20;                       // 設定每 100ms 顯示電壓一次
}
} while( Input == 0);                     // 迴圈內檢查是否有新按鍵按下? 若無則繼續顯示電
壓                                         // 若有則跳出顯示電壓函數

Clear_LCD( );
}

void Read_ADC(void)
{
    unsigned char i;

    GO_nDONE = 1 ;
//    NOP( );
//    while(GO_nDONE);
    for(i=0;i<30;i++) ;                   // Delay 42uS for Conversion

    ADC_Result.Byte[0] = ADRESL ; // Get 10-bit ADC Result
    ADC_Result.Byte[1] = ADRESH ;
}

void Initialize_ADC(void)
{
    TRISB2 = 1;                          // Set RB2 as Input pin
    ANSB2 = 1;                            // Set RB2/AN8 as Analog input function pin
    ADCON0 = 0b00100001;                 // ADC Enable, Select AN8 for VR2 input (RB2/AN8)
    ADCON1 = 0b11100000;                 // Right Justified, Fosc/64, Vref- to Vss, Vref+ to Vdd
    ADIF = 0;
}

```