



MICROCHIP

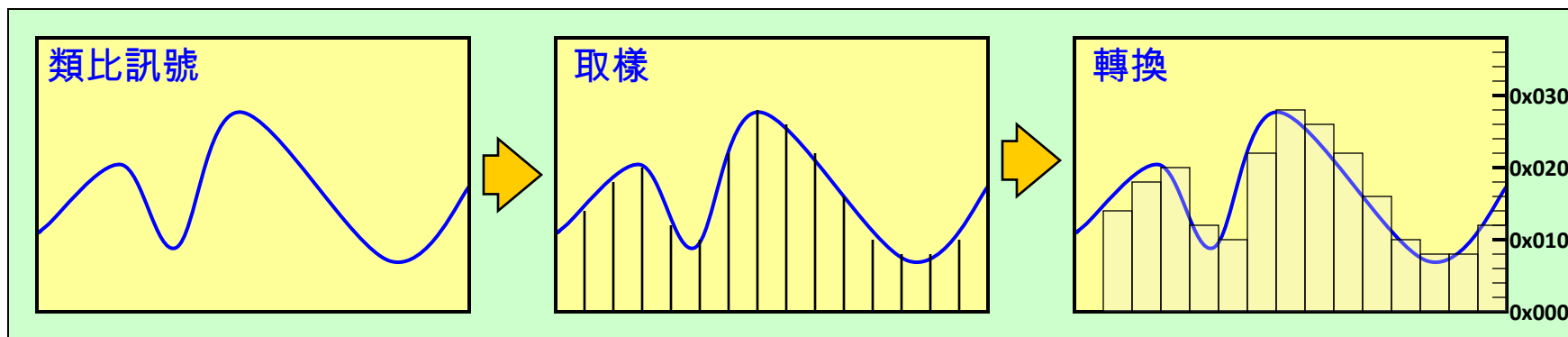
Regional Training Centers

Section 10

ADC

What's ADC ?

- ADC : Analog Digital Conversion, 類比數位轉換器。
一個可將類比訊號轉換成數位資料的模組。
- ADC的轉換過程, 可以分為兩個步驟, 如下圖所示, 首先對類比信號進行“**取樣**”, 利用外部的類比訊號對ADC內部的小電容充電, 已取得外部類比訊號的複本, 接著再將獲得的資料加以“**轉換**”, 獲得量化後的數位資料。

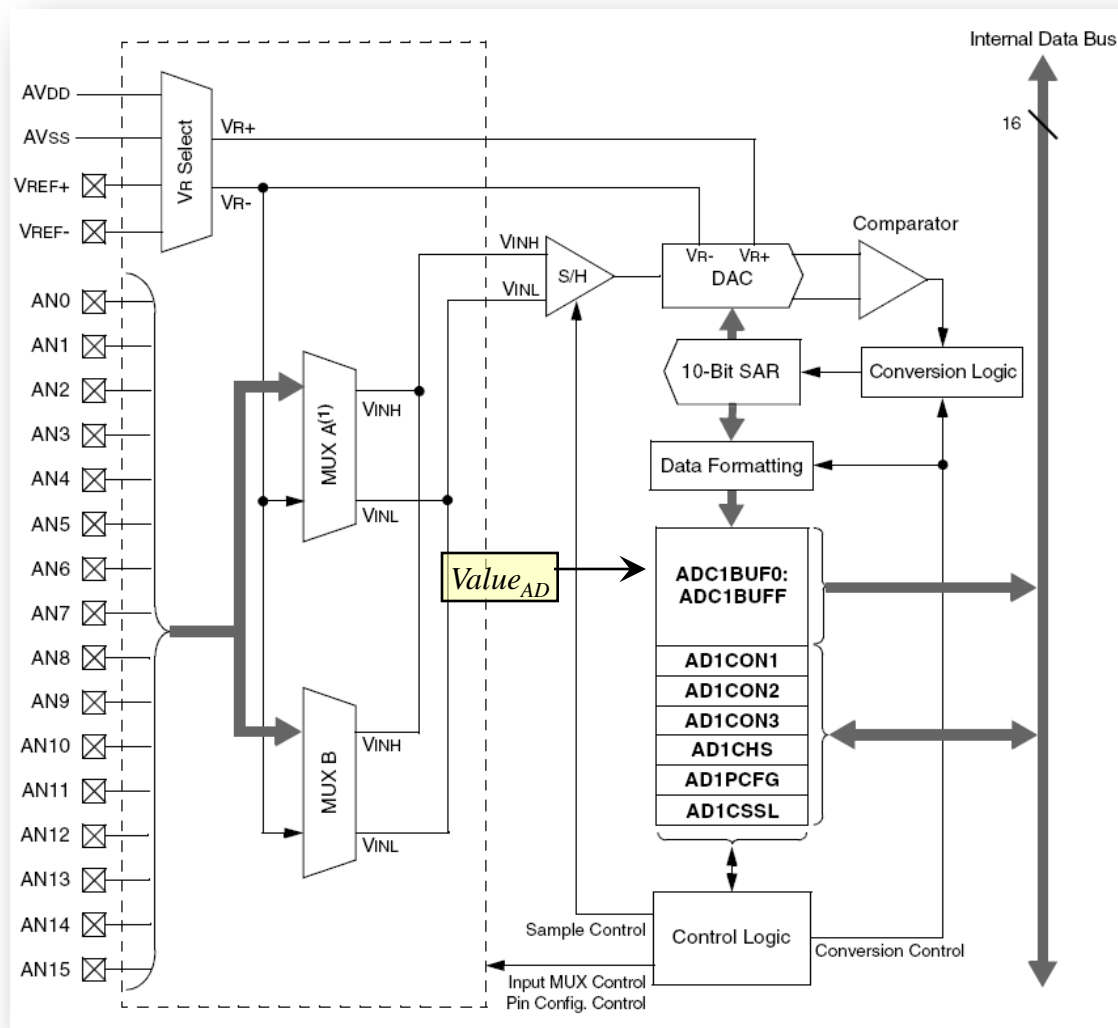


PIC24F ADC Arch.

- PIC24F具有一組採用SAR(連續近似法)的ADC。搭配16對1之類比多工器,達成多通道轉換功能。
- 具兩組多工器,可交替使用,多工器A支援通道掃描轉換功能。
- 類比信號轉換結果為

$$Value_{AD} = \frac{V_{AD} - V_{R-}}{V_{R+} - V_{R-}} \times 2^n$$

- V_{R+} , V_{R-} 可使用 V_{ref+} , V_{ref-} 或 AV_{DD} , AV_{SS} 。



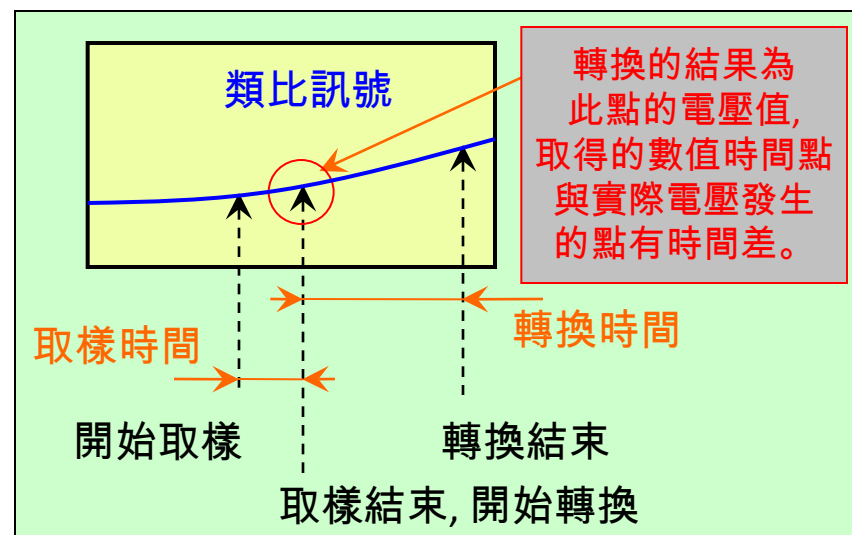
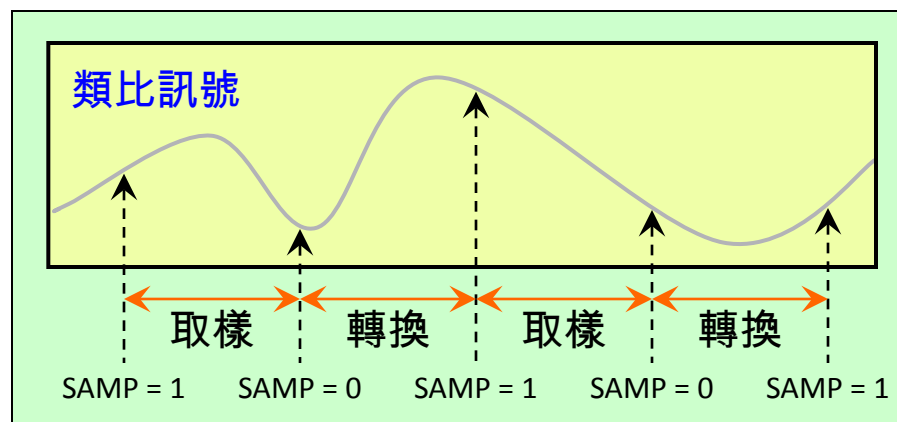
Sample and Conversion Sequence

- ADC作動分為“取樣”及“轉換”兩步驟。

取樣:外部訊號對ADC內部的電容器充電,取得外部電壓值。

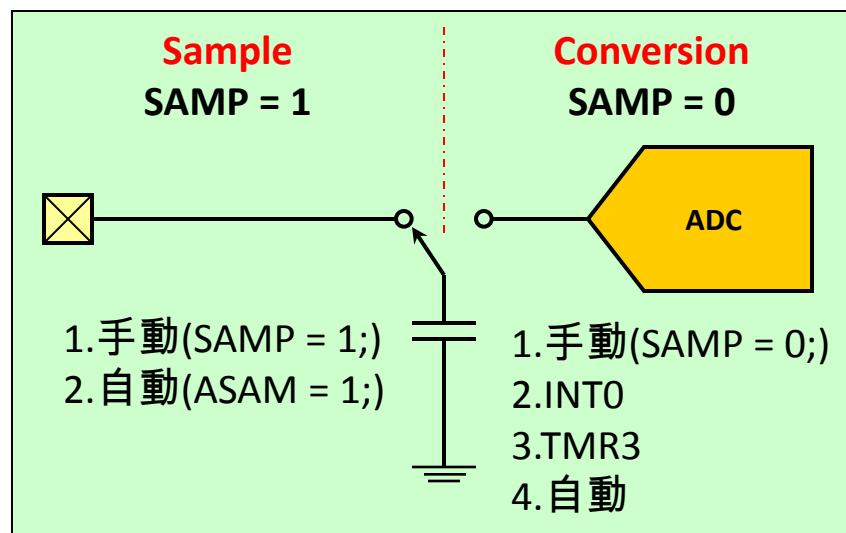
轉換:依據取得的電壓值換算出結果。

- ADC的取樣時間與轉換時間都有最短需求時間的規範,設計時必須滿足才能確保轉換結果正確。時間規範可查詢Datasheet電氣特性章節。
- PIC24系列的取樣時間至少需要 $1T_{AD}$,轉換時間需要 $12T_{AD}$ 。

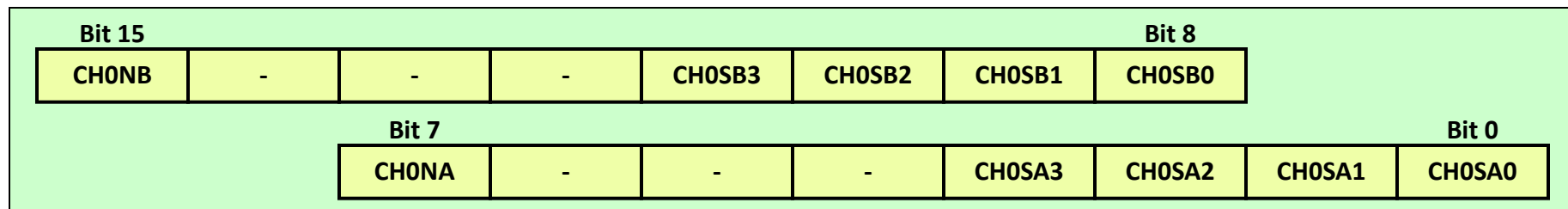


ADC Sample, Conv. Trigger Source

- ADC的轉換其實一點都不複雜,所有動作其實都圍繞在SAMP位元(A/D Sample Enable bit), **SAMP=1時進行取樣, SAMP=0則進行轉換。**
- 手動取樣:SAMP完全由軟體控制,硬體不干涉。
- 自動取樣:SAMP可由硬體控制,在每次轉換完成後,自動將SAMP設為1。(透過將ASAM設為1可達成)
- 要進入轉換狀態,將SAMP清除為0即可。
- PIC24提供手動(軟體清除),自動,外部中斷或Timer3 Match等方式清除SAMP,觸發轉換。



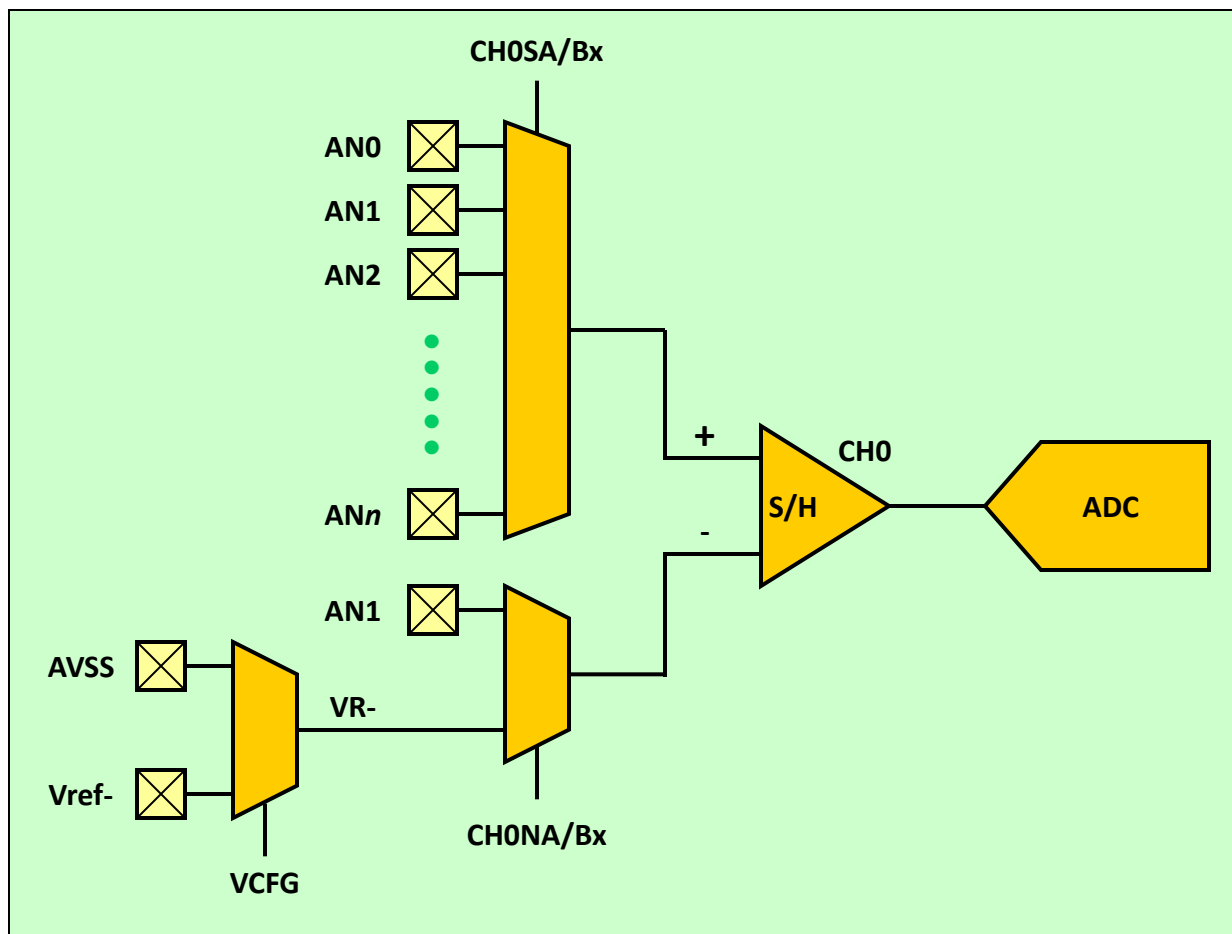
類比通道選擇



- ADC透過類比多工器,連接到多個類比接腳。到底要將哪個接腳的訊號,送給ADC處理。必須設定ADxCDn暫存器來指定。
- CH0NB:多工器B負端輸入選擇。
1 = AN1, 0 = V_{R-}
- CH0NA:多工器A負端輸入選擇。
1 = AN1, 0 = V_{R-}
- CH0SB<3:0>:多工器B正端輸入選擇。
1111 = AN15, 01110 = AN14, ..., 0000 = AN0
- CH0SA<3:0>:多工器A正端輸入選擇。
1111 = AN15, 01110 = AN14, ..., 0000 = AN0

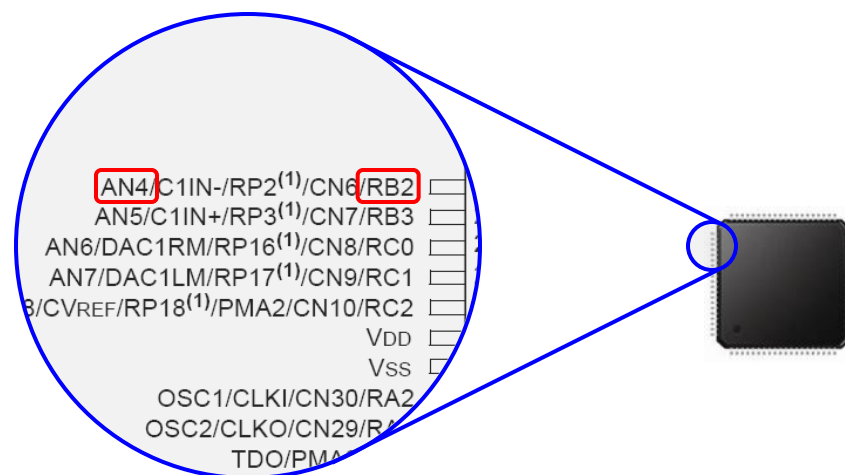
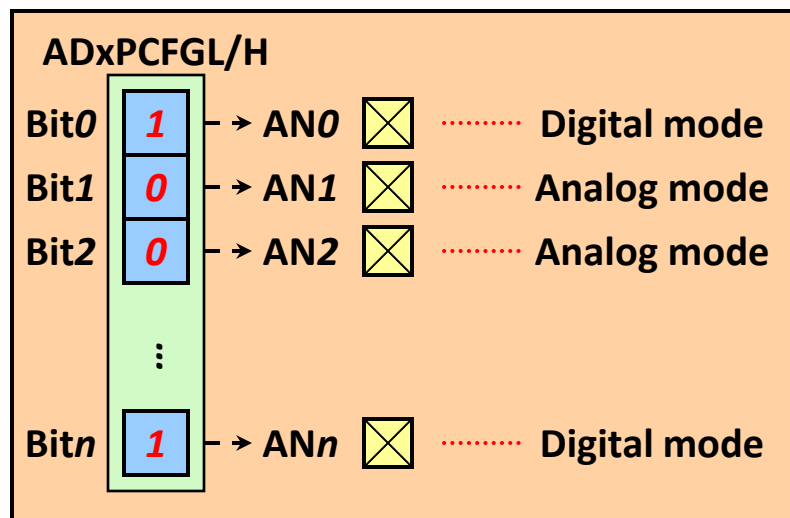
類比通道選擇

- 正端可以選擇 AN15~AN0,負端則可選擇VR-或AN1。
- 一般負端都是直接連接VR-。
- 負端如果連接 AN1,稱為“單端差動”模式。正負兩端的電壓會相減後才送入ADC,類似“差動”的概念,但兩者有相同的地(GND)。



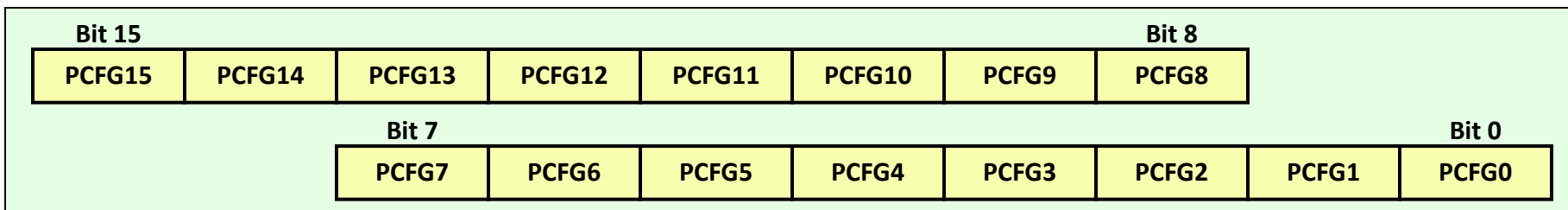
Pin's Analog or Digital Mode

- 回憶下IO章節的內容。某些 IO接腳(RX_n)的功能跟類比輸入(AN_n)共用的。這類接腳必須明確指定要設定為何模式。
- 作為ADC的輸入接腳時,必須設為Analog Mode。要做IO Port則要設為Digital Mode。
- 這類接腳,在MCU Reset/Power On後的預設值為Analog mode,無法做為IO Port使用。在使用時必須先設定Mode。



Analog or Digital Mode Control

ADxPCFGL



- 在PIC24中,透過ADxPCFGL來設定接腳的模式。
- 每一類比接腳AN n 都可以單獨設定,設成Analog Mode後,讀取IO的結果為"0"。MCU編號不同,AN n 接腳數量也不同。

MPLAB C30對ADC的支援

- MPLAB C30提供許多ADC Function可供使用。要使用ADC時跟Timer相同,必須含入(include)ADC的標頭檔。
Ex:#include <ADC.h>
- 在C:\ProgramFiles\Microchip\mplabc30\vx.x\docs\periph_lib\Microchip PIC24F Peripheral Library.chm 檔案中可以找到詳細的使用說明。
- 使用ADC Function前,最好先閱讀下Function的說明。



MPLAB C30 ADC Function

MPLAB C30中可用的ADC Function:

OpenADCx(); // 啟用ADCx,設定ADCx工作模式。

SetChanADCx(); // 手動切換ADCx的取樣通道。

ConvertADCx(); // 手動觸發轉換(SAMP -> 0)。

BusyADCx(); // ADC是否忙碌中?

ReadADCx(); // 讀取ADCx轉換的結果。

CloseADCx(); // 關閉ADCx。

ConfigIntADCx(); // 致能ADCx的中斷,並設定中斷優先權。

- MPLAB C30中可用的ADC Macro:

EnableIntADCx(); //開啟ADCx的中斷。

DisableIntADCx; //關閉ADCx的中斷。

SetPriorityIntADCx(); //設定ADCx中斷優先權。

Ex:SetPriorityADC1(7);

ADC10 Open Example

- ADC10的初始化範例:

```
void OpenADC10_v2( unsigned int config1 , unsigned int config2 ,  
                  unsigned int config3 ,  
                  unsigned int configport_low , unsigned int configport_high ,  
                  unsigned int configscan_low , unsigned int configscan_high );
```

config1 , *config2* , *config3* : ADC10的工作模式,

configport_low , *configport_high* : 設定類比通道的運作模式(Digital Mode / Analog Mode),

configscan_low , *configscan_high* : 設定開啟通到掃描模式時, 要掃描的通道。

Ex:

```
OpenADC10_v2( ADC_MODULE_ON & ... & ADC_CLK_MANUAL &  
              ADC_AUTO_SAMPLING_ON & ADC_SAMP_OFF ,  
              ADC_VREF_AVDD_AVSS & ADC_SCAN_OFF & ... ,  
              ADC_SAMPLE_TIME_31 & ADC_CONV_CLK_SYSTEM & ... ,  
              ENABLE_AN2_ANA , 0 ,  
              DISABLE_ALL_INPUT_SCAN , 0 );
```

ADC10 Select Channel Example

- ADC1手動切換取樣通道的範例:

void SetChanADC10(unsigned int *channel0*);

channel0 : 設定Channel0的多工器A,B的正負端輸入,

Ex:

```
SetChanADC10(ADC_CH0_POS_SAMPLEA_AN1 &  
              ADC_CH0_NEG_SAMPLEA_VREFN &  
              ADC_CH0_POS_SAMPLEB_AN0 &  
              ADC_CH0_NEG_SAMPLEB_VREFN );
```

Lab7 ADC Single

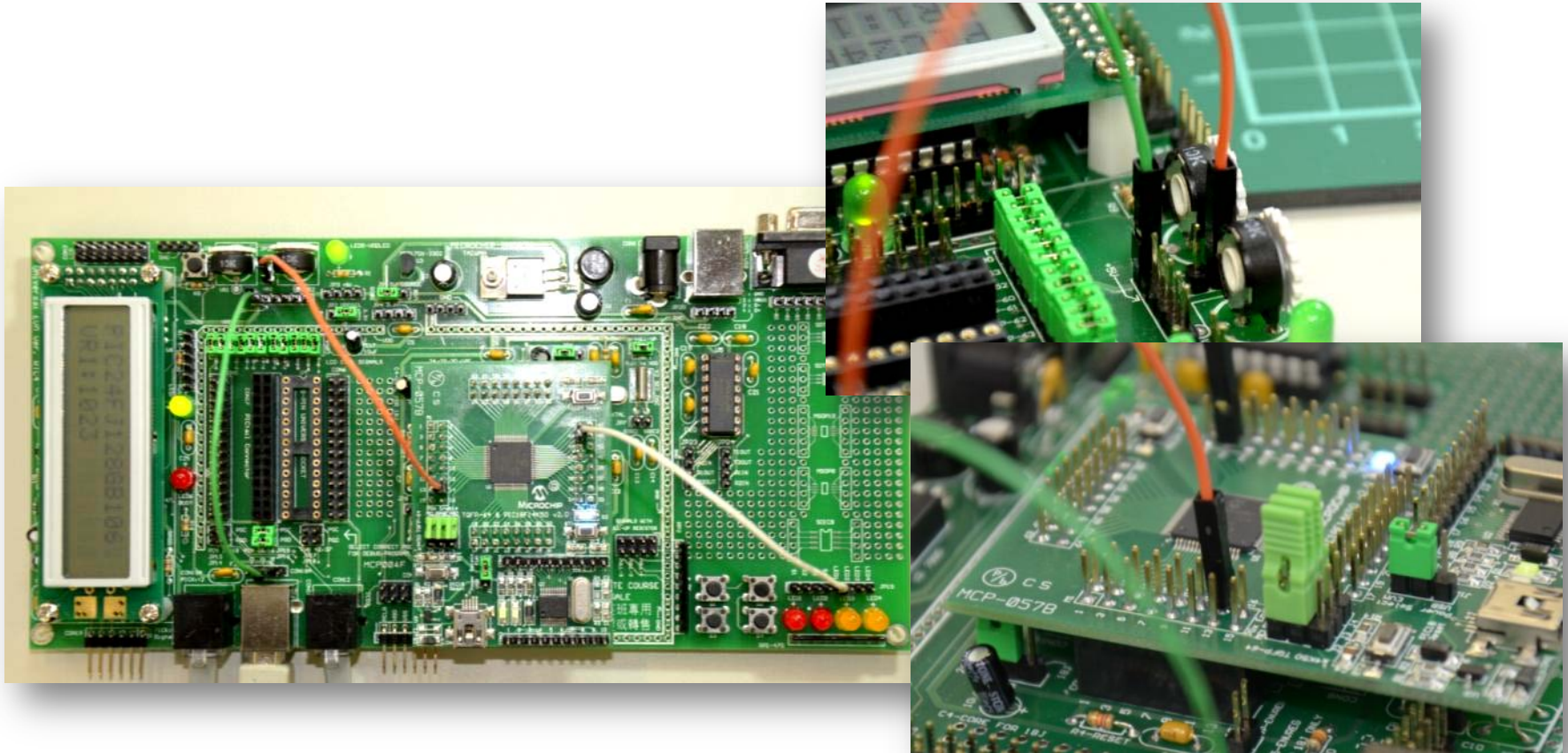
- 在Lab5的程式基礎上,嘗試加入ADC的程式片斷。利用ADC10來取得VR1的類比電壓值,並顯示在LCD Module上。
記住前面所提過的,要使用ADC時必須先含入(include)ADC的標頭檔。

Ex:#include <ADC.h>

- ADC的工作模式設定為,自動取樣,手動轉換,類比通道設定為AN2,無號整數格式,不開啟通道掃描。
- ADC的設定比Timer複雜很多,設定上要更細心,先閱讀ADC Function的說明文件,了解ADC Function的始用方法。至少必須了解OpenADCx(),SetChanx(), ConvertADCx();
BusyADCx(),ReadADCx(),用法。

Lab7 ADC Single Step1

- 將PIC24FJ128GB106的AN2(Pin14)接到VR1的Pin上(JP22)。
- 可利用LED (RD0)與LCD Module觀察程式運作結果。



Lab7 ADC Single Step2

- 程式ADC的工作模式, 通道選擇要如何設定?

觀察OpenADC10_2()與SetChanADC10()裡面缺漏的參數, 然後查閱Microchip PIC24F Peripheral Library.chm文件中有關ADC的說明, 將缺漏的部分補齊。

- 如何觸發ADC開始轉換?

ADC被設定為手動觸發轉換模式, 所以必須自行呼叫來觸發ADC開始轉換ConvertADC10()。

程式中利用Timer1產生100mS的中斷事件, 請在Timer中斷中加入ConvertADC10(), 讓ADC每100mS會進行一次轉換。

Lab7 ADC Single Step3

- 如何取得ADC轉換的結果？

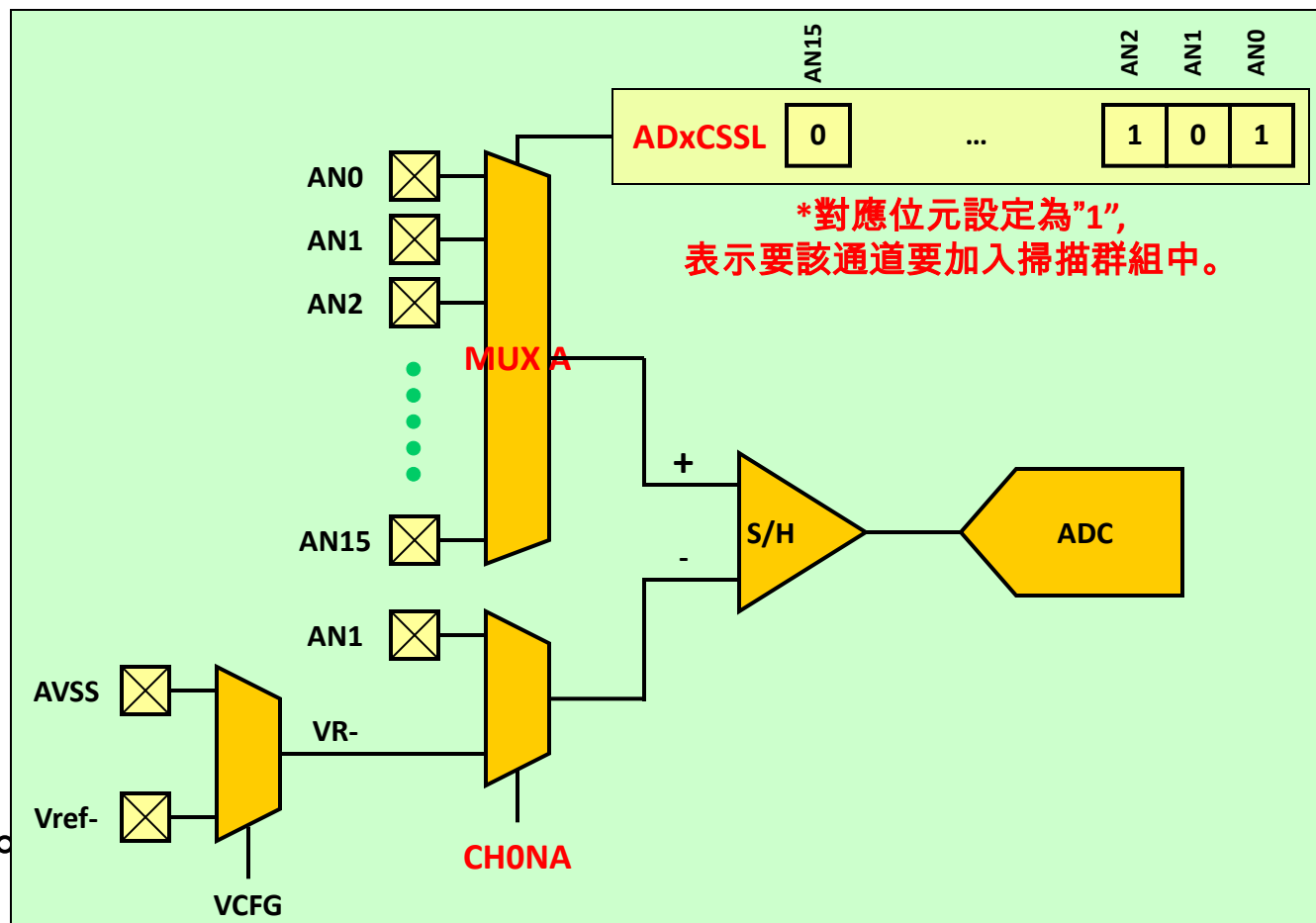
可以透過ReadADC10(*n*)取得ADCBUF*n*裡面的資料。
讀取資料前必須先判斷ADC中斷旗標或Busy Flag來確定ADC已經完成轉換, 不然無法取得正確的結果。

Lab8 ADC Single Timer3 Trigger

- 利用Lab7的程式基礎,將ADC1的轉換觸發來源由手動觸發改成TMR3觸發。
- 建立ADC1的中斷服務常式,並開啟ADC1的中斷,設定優先權。優先權設定為預設值”4”。
- 將原先Polling AD1IF的模式,改成由中斷服務常式完成。
- 閱讀ADC Function的說明文件,了解ADC Function的始用方法。了解如何使用OpenADCx()改變”轉換”的觸發來源,跟ConfigIntADCx(); 的用法。

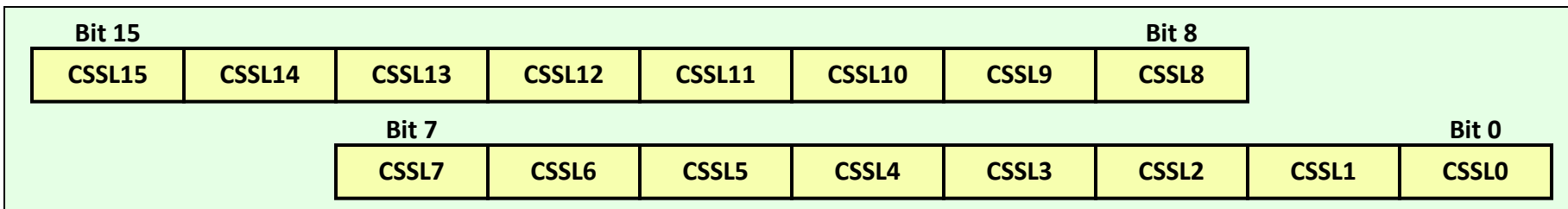
Auto Channel Scan

- 多工器A可支援自動通道掃描模式,透過設定($CSCNA = 1$), 可開啟通道掃描。
- ADC會自動的切換通道,取得類比電壓,進行掃描。
- 搭配轉換次數($SMPI < 3:0 >$)的設定可以規劃轉換資料存在Buffer中的排列方式。



Channel Scan Select

ADxCSSL



- 開啟通掃描模式後(ADxCON2bits.CSCNA = 1), 必須指定哪些通道要進行掃描。指定方式就是設定ADxCSSL暫存器。被設定為"1"的通道,就會列入掃描。
- ADC在每次轉換的時候,會依通道編號,由小至大自行切換通道,轉換結果。

Lab9 ADC Scan Timer3 Trigger

- 利用Lab8的程式基礎, 開啟自動掃描功能, 讓ADC1可以依序的取得VR1, VR2的結果。
- 將VR1, VR2的結果, 透過LCD Module顯示。
- 閱讀ADC Function的說明文件, 了解ADC Function的始用方法。了解如何使用OpenADCx()開啟自動掃描功能。

Lab9 ADC Scan Timer3 Trigger Step

- 將PIC24FJ256GB106的AN2(Pin14)接到VR1的Pin上(JP22) ,的AN3(Pin13)接到VR2的Pin上(JP22) 。
- 可利用LED (RD0)與LCD Module觀察程式運作結果。

