微算機 Lab9 - A/D Converter

> 參考自去年教材

ADC簡介

什麼是ADC

主要功能：把輸入的類比訊號轉成數位數值

本次Lab會把可變電阻輸入的電壓轉成數值形式

更多參考 ADC - 成大資工 (http://wiki.csie.ncku.edu.tw/embedded/ADC)

VREF與resolution

VREF+：上界的參考電壓

VREF-：下界的參考電壓

Resolution：ADC的解析度

> e.g., VREF- = 0V, VREF+ = 10V, Resolution : 10bits (range = [0,1023])
> 0V→ 0
> 5V → 511
> 10V → 1023

T_{AD}

T_{AD} : A/D Clock period, the time required to convert one bit

T_{AD} 越小越好，但要大於0.7μs

TABLE 26-25: A/D CONVERSION REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
130	T_{AD}	A/D Clock Period	PIC18**F**XXXX	0.7	25.0(1)	μs	T_{OSC} based, $V_{REF} \geq 3.0V$
			PIC18**LF**XXXX	1.4	25.0(1)	μs	V_{DD} = 2.0V; T_{OSC} based, V_{REF} full range
			PIC18**F**XXXX	—	1	μs	A/D RC mode
			PIC18**LF**XXXX	—	3	μs	V_{DD} = 2.0V; A/D RC mode
131	T_{CNV}	Conversion Time (not including acquisition time) **(Note 2)**		11	12	T_{AD}	
132	T_{ACQ}	Acquisition Time **(Note 3)**		1.4	—	μs	-40°C to +85°C
135	T_{SWC}	Switching Time from Convert → Sample		—	**(Note 4)**		
TBD	T_{DIS}	Discharge Time		0.2	—	μs	

Note 1: The time of the A/D clock period is dependent on the device frequency and the T_{AD} clock divider.

2: ADRES register may be read on the following T_{CY} cycle.

3: The time for the holding capacitor to acquire the "New" input voltage when the voltage changes full scale after the conversion (V_{DD} to V_{SS} or V_{SS} to V_{DD}). The source impedance (R_S) on the input channels is 50Ω.

4: On the following cycle of the device clock.

T_{AD} 設定

TABLE 19-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock Source (TAD)		Maximum Device Frequency	
Operation	ADCS<2:0>	PIC18F2X20/4X20	PIC18LF2X2X/4X20[4]
2 TOSC	000	2.86 MHz	1.43 kHz
4 TOSC	100	5.71 MHz	2.86 MHz
8 TOSC	001	11.43 MHz	5.72 MHz
16 TOSC	101	22.86 MHz	11.43 MHz
32 TOSC	010	40.0 MHz	22.86 MHz
64 TOSC	110	40.0 MHz	22.86 MHz
RC[3]	x11	1.00 MHz[1]	1.00 MHz[2]

Note 1: The RC source has a typical TAD time of 1.2 μs.
2: The RC source has a typical TAD time of 2.5 μs.
3: For device frequencies above 1 MHz, the device must be in Sleep for the entire conversion or the A/D accuracy may be out of specification.
4: Low-power (PIC18LFXXXX) devices only.

REGISTER 19-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified
0 = Left justified

bit 6 **Unimplemented:** Read as '0'

bit 5-3 **ACQT<2:0>:** A/D Acquisition Time Select bits
111 = 20 TAD
110 = 16 TAD
101 = 12 TAD
100 = 8 TAD
011 = 6 TAD
010 = 4 TAD
001 = 2 TAD
000 = 0 TAD[1]

bit 2-0 **ADCS<2:0>:** A/D Conversion Clock Select bits
111 = FRC (clock derived from A/D RC oscillator)[1]
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC (clock derived from A/D RC oscillator)[1]
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

Note 1: If the A/D FRC clock source is selected, a delay of one TCY (instruction cycle) is added before the A/D clock starts. This allows the `SLEEP` instruction to be executed before starting a conversion.

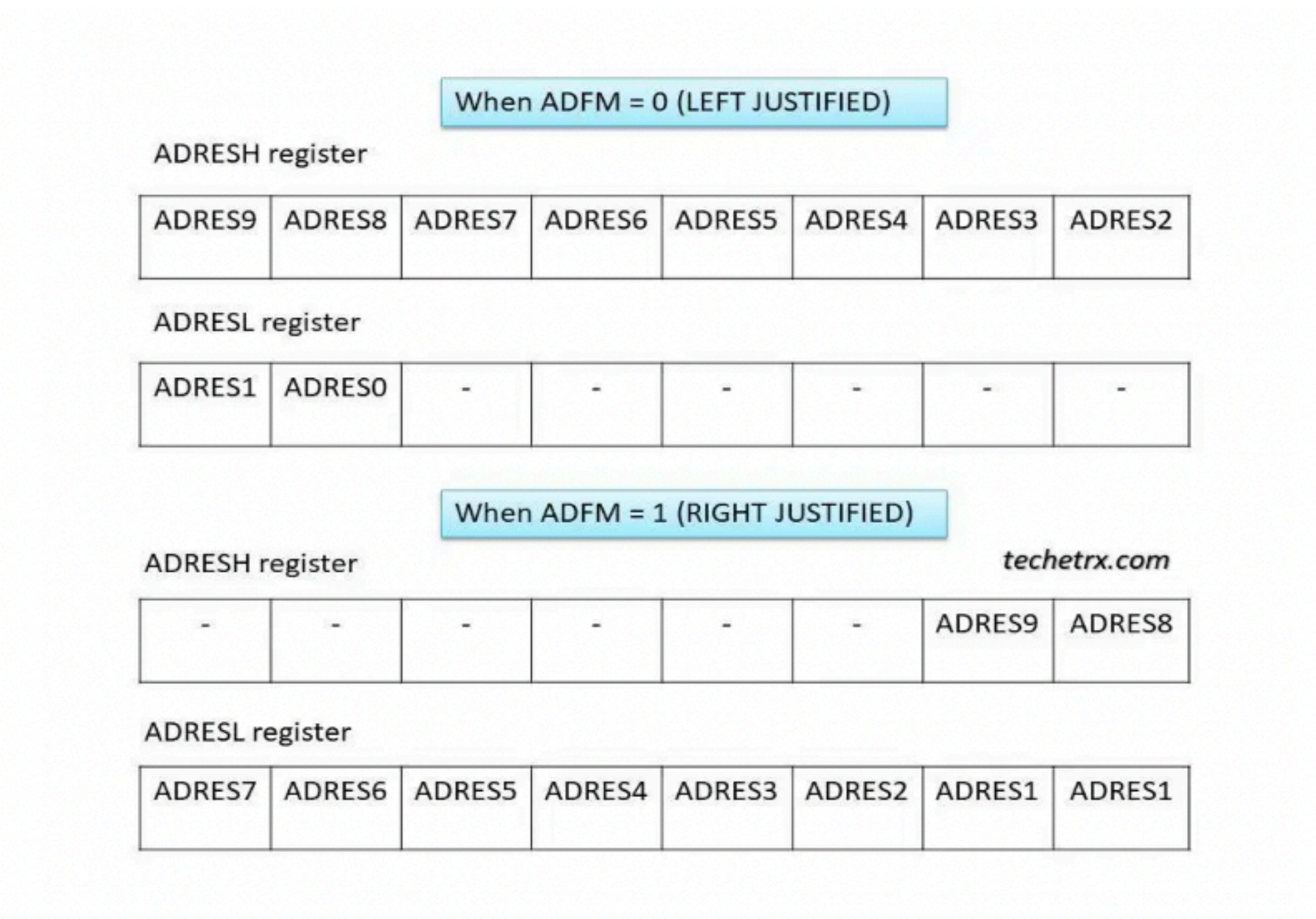
透過查表設定ADCS(ADCON2)

假設頻率 F_{OSC} 是 2.86 MHz, 則周期 (T_{OSC})會是 $\frac{1}{2.86\times10^6}\approx 0.35\mu s$，為了滿足最低 A/D Clock period ($0.7\mu s$)，要把T_{AD}設成兩倍的T_{OSC}，Operation欄位中的數值即為T_{AD}

e.g.,假設頻率 (Fosc) 是 1 MHz，透過查表得知ADCS要設成000，
而Operation欄位是2×Tosc=2×$\frac{1}{1MHz}=2\mu s$，因此T_{AD} 為2μs

Left/Right justified

ADC轉換的結果放在ADRES register裡，存放的方式分為left justified與right justified，可依據使用需求設置
e.g., 需要8 bits resolution，設定為left justified，取ADRESH數值；需要10 bits resolution，則設定為right justified，將ADRESH前兩bits與ADRESL結合



When ADFM = 0 (LEFT JUSTIFIED)

ADRESH register

ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2

ADRESL register

ADRES1	ADRES0	-	-	-	-	-	-

When ADFM = 1 (RIGHT JUSTIFIED)

ADRESH register

techetrx.com

-	-	-	-	-	-	ADRES9	ADRES8

ADRESL register

ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES1

Source (https://techetrx.com/pic-microcontroller/adc-in-pic16f887-microcontroller/)

ADC流程

1. Acquisition：採樣輸入電壓
2. Conversion：將電壓轉換成數值
3. Discharge：釋放電壓

Acquisition

採樣輸入電壓，需要時間

依據data sheet的推導(p. 228)，acquisition time最少會花$2.4\mu s$

EQUATION 19-1: ACQUISITION TIME

$$
\begin{aligned}
T_{ACQ} &= \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient} \\
&= T_{AMP} + T_C + T_{COFF}
\end{aligned}
$$

EQUATION 19-2: A/D MINIMUM CHARGING TIME

$$V_{HOLD} = (V_{REF} - (V_{REF}/2048)) \cdot (1 - e^{(-T_C/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$

or

$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S) \ln(1/2048)$$

EQUATION 19-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$$
\begin{aligned}
T_{ACQ} &= T_{AMP} + T_C + T_{COFF} \\
T_{AMP} &= 0.2\ \mu s \\
T_{COFF} &= (\text{Temp} - 25°C)(0.02\ \mu s/°C) \\
&\quad (85°C - 25°C)(0.02\ \mu s/°C) \\
&\quad 1.2\ \mu s
\end{aligned}
$$

Temperature coefficient is only required for temperatures > 25°C. Below 25°C, $T_{COFF} = 0\ \mu s$.

$$
\begin{aligned}
T_C &= -(C_{HOLD})(R_{IC} + R_{SS} + R_S) \ln(1/2047)\ \mu s \\
&\quad -(25\ pF)(1\ k\Omega + 2\ k\Omega + 2.5\ k\Omega) \ln(0.0004883)\ \mu s \\
&\quad 1.05\ \mu s \\
T_{ACQ} &= 0.2\ \mu s + 1\ \mu s + 1.2\ \mu s \\
&\quad \boxed{2.4\ \mu s}
\end{aligned}
$$

REGISTER 19-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **ADFM:** A/D Result Format Select bit
- 1 = Right justified
- 0 = Left justified

bit 6 **Unimplemented:** Read as '0'

bit 5-3 **ACQT<2:0>:** A/D Acquisition Time Select bits
- 111 = 20 TAD
- 110 = 16 TAD
- 101 = 12 TAD
- 100 = 8 TAD
- 011 = 6 TAD
- 010 = 4 TAD
- 001 = 2 TAD
- 000 = 0 TAD(1)

bit 2-0 **ADCS<2:0>:** A/D Conversion Clock Select bits
- 111 = FRC (clock derived from A/D RC oscillator)(1)
- 110 = FOSC/64
- 101 = FOSC/16
- 100 = FOSC/4
- 011 = FRC (clock derived from A/D RC oscillator)(1)
- 010 = FOSC/32
- 001 = FOSC/8
- 000 = FOSC/2

Note 1: If the A/D FRC clock source is selected, a delay of one TCY (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

依據T_{AD}的時間決定ACQT，若T_{AD}為$2\mu s$，則ACQT要設成001，也就是$2T_{AD}$= $4\mu s$>$2.4\mu s$

Conversion and Discharge

將採樣電壓轉換成數值，需要時間

依據data sheet，conversion需要花11到12個T_{AD}

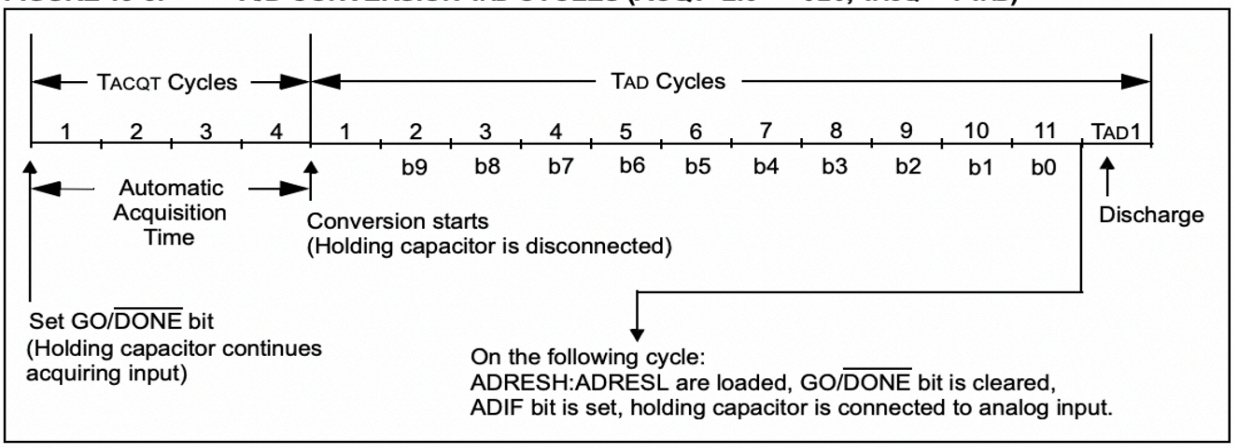
TABLE 26-25: A/D CONVERSION REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
130	TAD	A/D Clock Period	PIC18**F**XXXX	0.7	25.0[1]	μs	TOSC based, VREF ≥ 3.0V
			PIC18**LF**XXXX	1.4	25.0[1]	μs	VDD = 2.0V; TOSC based, VREF full range
			PIC18**F**XXXX	—	1	μs	A/D RC mode
			PIC18**LF**XXXX	—	3	μs	VDD = 2.0V; A/D RC mode
131	TCNV	Conversion Time (not including acquisition time) **(Note 2)**		11	12	TAD	
132	TACQ	Acquisition Time **(Note 3)**		1.4	—	μs	-40°C to +85°C
135	TSWC	Switching Time from Convert → Sample		—	**(Note 4)**		
TBD	TDIS	Discharge Time		0.2	—	μs	

Note 1: The time of the A/D clock period is dependent on the device frequency and the TAD clock divider.
2: ADRES register may be read on the following TCY cycle.
3: The time for the holding capacitor to acquire the "New" input voltage when the voltage changes full scale after the conversion (VDD to VSS or VSS to VDD). The source impedance (*RS*) on the input channels is 50Ω.
4: On the following cycle of the device clock.

時間表

FIGURE 19-5: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



PIC18 ADC register introduction

ADCON0

CHS：設定analog input 輸入腳位

GO/DONE：設為1時(ADCON0bits.GO = 1)開始做ADC，轉換完後 GO/DONE會自動設為0

ADON：開啟ADC功能

REGISTER 19-1: ADCON0: A/D CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-2 **CHS<3:0>:** Analog Channel Select bits
- 0000 = Channel 0 (AN0)
- 0001 = Channel 1 (AN1)
- 0010 = Channel 2 (AN2)
- 0011 = Channel 3 (AN3)
- 0100 = Channel 4 (AN4)
- 0101 = Channel 5 (AN5)(1,2)
- 0110 = Channel 6 (AN6)(1,2)
- 0111 = Channel 7 (AN7)(1,2)
- 1000 = Channel 8 (AN8)
- 1001 = Channel 9 (AN9)
- 1010 = Channel 10 (AN10)
- 1011 = Channel 11 (AN11)
- 1100 = Channel 12 (AN12)
- 1101 = Unimplemented(2)
- 1110 = Unimplemented(2)
- 1111 = Unimplemented(2)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
<u>When ADON = 1:</u>
- 1 = A/D conversion in progress
- 0 = A/D Idle

bit 0 **ADON:** A/D On bit
- 1 = A/D Converter module is enabled
- 0 = A/D Converter module is disabled

Note 1: These channels are not implemented on 28-pin devices.
2: Performing a conversion on unimplemented channels will return a floating input measurement.

ADCON1

VCFG1：設定下界參考電壓

VCFG0 :設定上界參考電壓

PCFG：設定ANx PORT為類比還是數位，使用 ADC 的同時若發現其他 PORT 的 input 值怪怪

的也許是誤把那些 PORT 設成 analog input

REGISTER 19-2: ADCON1: A/D CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-q[1]	R/W-q[1]	R/W-q[1]
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **VCFG1:** Voltage Reference Configuration bit (VREF- source)
1 = VREF- (AN2)
0 = VSS

bit 4 **VCFG0:** Voltage Reference Configuration bit (VREF+ source)
1 = VREF+ (AN3)
0 = VDD

bit 3-0 **PCFG<3:0>:** A/D Port Configuration Control bits:

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7[2]	AN6[2]	AN5[2]	AN4	AN3	AN2	AN1	AN0
0000[1]	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111[1]	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = Analog input D = Digital I/O

Note 1: The POR value of the PCFG bits depends on the value of the PBADEN Configuration bit. When PBADEN = 1, PCFG<2:0> = 000; when PBADEN = 0, PCFG<2:0> = 111.

2: AN5 through AN7 are available only on 40/44-pin devices.

ADCON2

ADFM：設定justified

ADCS：選擇conversion clock

ACQT：選擇acquisition time要幾個T_{AD}

REGISTER 19-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified
0 = Left justified

bit 6 **Unimplemented:** Read as ‘0’

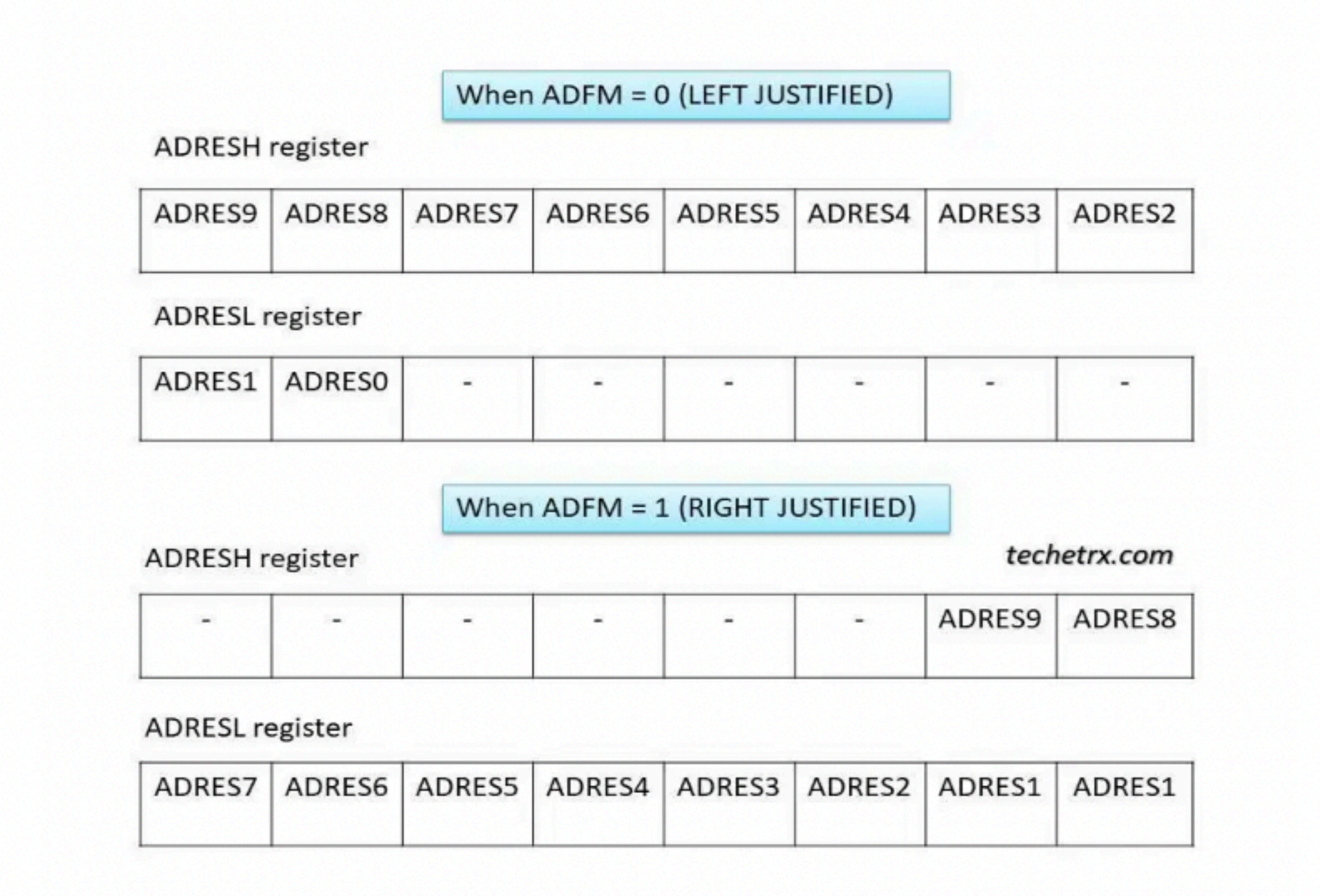
bit 5-3 **ACQT<2:0>:** A/D Acquisition Time Select bits
111 = 20 TAD
110 = 16 TAD
101 = 12 TAD
100 = 8 TAD
011 = 6 TAD
010 = 4 TAD
001 = 2 TAD
000 = 0 TAD(1)

bit 2-0 **ADCS<2:0>:** A/D Conversion Clock Select bits
111 = FRC (clock derived from A/D RC oscillator)(1)
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC (clock derived from A/D RC oscillator)(1)
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

Note 1: If the A/D FRC clock source is selected, a delay of one TCY (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

ADRESH、ADRESL

result of conversion



[Source (https://techetrx.com/pic-microcontroller/adc-in-pic16f887-microcontroller)](https://techetrx.com/pic-microcontroller/adc-in-pic16f887-microcontroller)

workflow of ADC using interrupt I/O

step-1

Configure the ADC module:

- Select VREF (ADCON1.VCFG0, ADCON1.VCFG1)
- Select A/D port control(ADCON1.PCFG)
- Select A/D input channel (ADCON0.CHS)
- Select A/D conversion clock (ADCON2.ADCS)
- Select A/D acquisition time (ADCON2.ACQT)
- Select justified method (ADCON2.ADFM)

- Turn on A/D module (ADCON0.ADON)

> Note : The port pins needed as analog inputs must have their corresponding TRIS bits set (input).

step-2

Configure the ADC interrupt:

- Enable A/D interrupt (PIE1.ADIE)
- Clear A/D interrupt flag bit (PIR1.ADIF)
- Enable peripheral interrupt (INTCON.PEIE)
- Set GIE bit (INTCON.GIE)

step-3

Start conversion:

- Set GO/DONE bit (ADCON0.GO)

step-4

Conversion completed:

- Go to ISR
- Read value of ADRES register
- Do things you want
- Clear ADC interrupt flag bit (PIR1.ADIF)

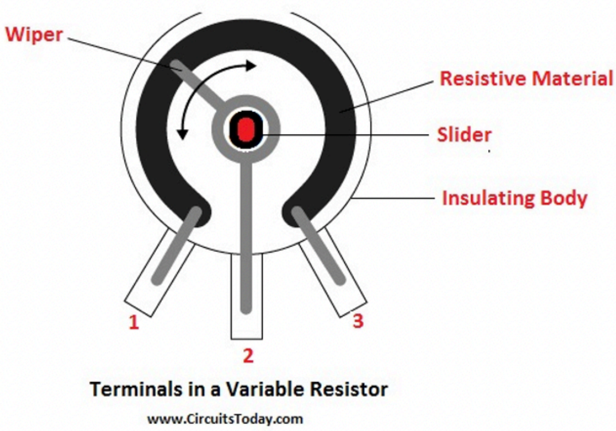
step-5

Next conversion(if required) :

- You need to have a minimum wait of 2 T_{AD} before next acquisition start, then go back to step 3.

Variable resistor

左邊接 5V，右邊接地，中間接 Analog 輸入



範例code

```
#include <xc.h>
#include <pic18f4520.h>
#include <stdio.h>

#pragma config OSC = INTIO67 // Oscillator Selection bits
#pragma config WDT = OFF     // Watchdog Timer Enable bit
#pragma config PWRT = OFF    // Power-up Enable bit
#pragma config BOREN = ON    // Brown-out Reset Enable bit
#pragma config PBADEN = OFF  // Watchdog Timer Enable bit
#pragma config LVP = OFF     // Low Voltage (single -supply) In-Circute Seria
#pragma config CPD = OFF     // Data EEPROM?Memory Code Protection bit (Data

void __interrupt(high_priority)H_ISR(){

    //step4
    int value = ADRESH;


    //do things


    //clear flag bit
    PIR1bits.ADIF = 0;
```

```
    //step5 & go back step3
    /*
    delay at least 2tad
    ADCON0bits.GO = 1;
    */

    return;
}

void main(void)
{
    //configure OSC and port
    OSCCONbits.IRCF = 0b100; //1MHz
    TRISAbits.RA0 = 1;       //analog input port

    //step1
    ADCON1bits.VCFG0 = 0;
    ADCON1bits.VCFG1 = 0;
    ADCON1bits.PCFG = 0b1110; //AN0 為analog input,其他則是 digital
    ADCON0bits.CHS = 0b0000;  //AN0 當作 analog input
    ADCON2bits.ADCS = 0b000;  //查表後設000(1Mhz < 2.86Mhz)
    ADCON2bits.ACQT = 0b001;  //Tad = 2 us acquisition time設2Tad = 4 > 2.4
    ADCON0bits.ADON = 1;
    ADCON2bits.ADFM = 0;    //left justified


    //step2
    PIE1bits.ADIE = 1;
    PIR1bits.ADIF = 0;
    INTCONbits.PEIE = 1;
    INTCONbits.GIE = 1;


    //step3
    ADCON0bits.GO = 1;

    while(1);

    return;
}
```