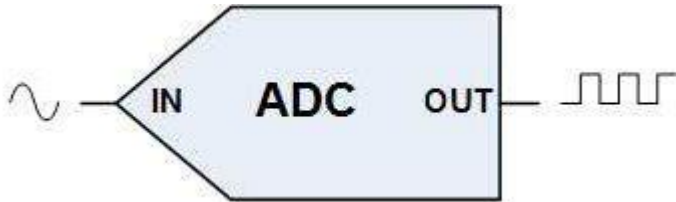


ADC Module of PIC Microcontroller

Analog to Digital Converter (ADC) is a device that converts an analog quantity (continuous voltage) to discrete digital values. This is very useful when we want to do some processing on physical quantities, which are normally analog in nature. Most of the PIC Microcontrollers have built in ADC Module. Here we are using PIC [16F877A](#) for demonstrating the working.



ADC_Symbol

ADC of PIC Microcontrollers have 5 inputs for 28 pin devices and 8 inputs for 40/44 pin devices. It is a 10-bit ADC, i.e. the conversion of analog signal results in corresponding 10-bit digital number. The positive and negative reference voltage (+Vref and -Vref) of PIC ADC is software selectable, which can be VDD, VSS, voltage at RA2 or RA3. This A/D Converter module can also operate in sleep mode in which clock is derived from its internal RC oscillator. Following points may help you to understand the concept of reference voltages.

- When the ADC input is -Vref, result will be 0000000000
- When the ADC input is +Vref, result will be 1111111111
- Resolution of ADC = $(+V_{ref} - -V_{ref}) / (2^{10} - 1)$, which is the minimum voltage required to change the ADC result by one bit.
- Here ADC Resolution = $(5 - 0) / (1024 - 1) = 5/1023 = 0.004887V$
- So, if the input is 5V, ADC Value is $5 / 0.004887 = 1023 = 11111111$ (binary)
- If the input is 0.004887V, ADC Value is $0.004887 / 0.004887 = 1 = 00000001$ (binary)

We require some hardware knowledge to program PIC ADC in Hi-Tech C. **If you don't need hardware knowledge please skip this part and go to Circuit Diagram.**

ADC Module in Detail

Registers

PIC ADC has 4 registers :

- ADCON0 – ADC Control Register 0
- ADCON1 – ADC Control Register 1
- ADRESH – ADC Result High Register
- ADRESL – ADC Result Low Register

ADCON0 Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

ADCON0 Register – PIC 16F877A

- **ADCS1** and **ADCS2** are used to select A/D Conversion Clock. It should be selected in accordance with device clock.
- **CH2**, **CH1** and **CH0** are used to select one of the analog input channel out of eight channels.
- **GO/DONE** is the A/D Conversion Status bit. Setting this bit initializes A/D Conversion and will be automatically cleared when the conversion is complete.
- **ADON** is used to switch on/off the ADC Module. When it is 1, the ADC Module turns ON and when it is 0, the ADC Module will be OFF.

ADCON1 Register

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

ADCON1 Register – PIC 16F877A

- **ADFM** is the ADC Result Format select bit. Two 8 bit register (ADRESH and ADRESL) are provided to store the 10-bit result of A/D Conversion, thus we need't use 6 bits. When ADFM is 1, the result will be right justified, ie Most Significant Bits of ADRESH will be read as 0. When ADFM is 0, the result will be left justified, ie Least Significant Bits of ADRESL will be read as zero.
- **ADCS2** is used to select A/D Conversion Clock in association with ADCS1 and ADC2 of ADCON0 register.
- **PCFG3 – PCFG0** are the A/D Port Configuration Control bits. Each pin among AN0 – AN7 is configured as analog, digital or reference voltage inputs according to the status of these configuration bits as given below.

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

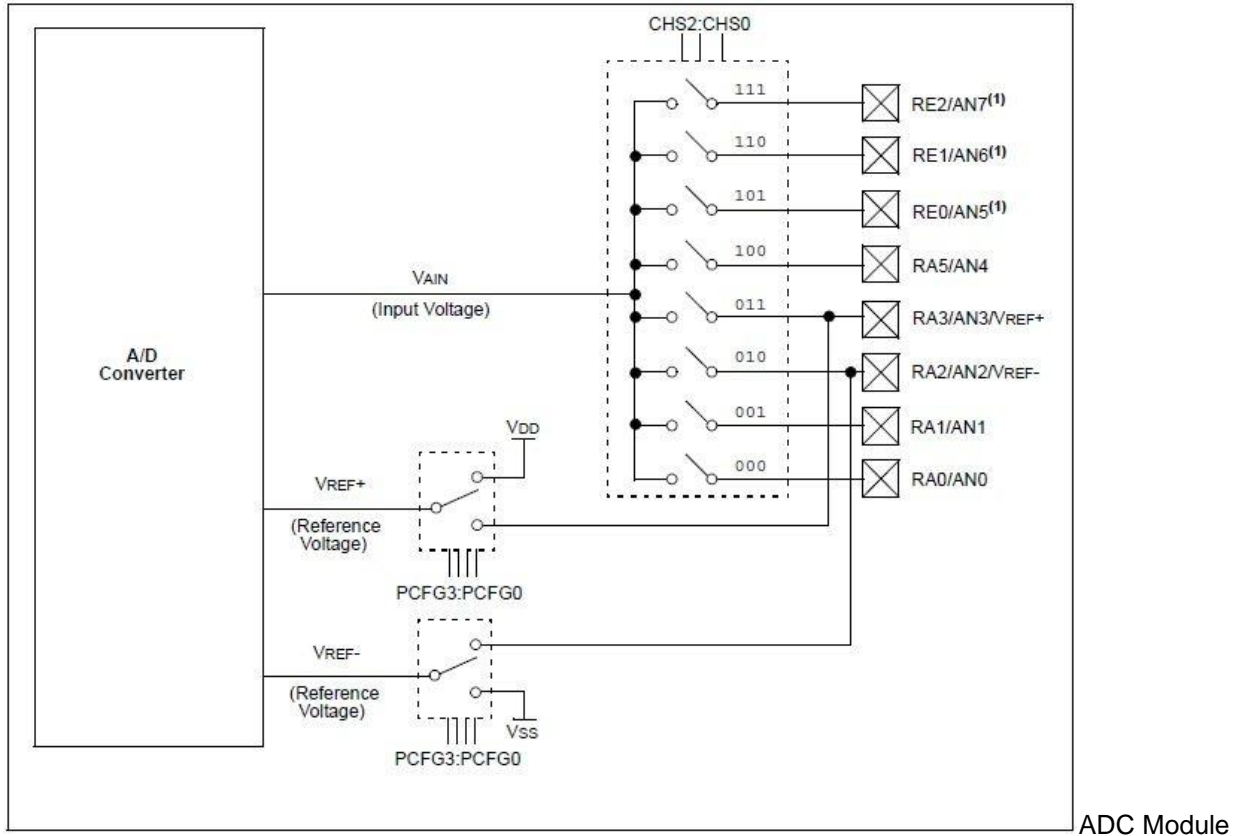
A = Analog input D = Digital I/O

C/R = # of analog input channels/# of A/D voltage references

PIC ADC Port Configuration Bits

Note : TRIS Registers of Analog inputs must be configured as input for the proper operation.

ADC Block Diagram



Block Diagram – PIC16F877A

Selection of A/D Conversion Clock

The time for A/D Conversion per bit is defined as T_{AD} and it requires minimum $12T_{AD}$ to generate the 10-bit result. The time T_{AD} is determined by the A/D Conversion Clock which is software selectable to following options.

- $2 T_{Osc}$
- $4 T_{Osc}$
- $8 T_{Osc}$
- $16 T_{Osc}$
- $32 T_{Osc}$
- $64 T_{Osc}$
- Internal RC oscillator of ADC Module

T_{Osc} is the time period of the device clock usually provided by the crystal oscillator. Care should be taken while selecting the A/D Conversion Clock, such that the clock should provide the minimum T_{AD} ($1.6\mu S$) required for the correct A/D Conversion. So refer the following table before setting the ADC clock.

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS2:ADCS1:ADCS0	
2 Tosc	000	1.25 MHz
4 Tosc	100	2.5 MHz
8 Tosc	001	5 MHz
16 Tosc	101	10 MHz
32 Tosc	010	20 MHz
64 Tosc	110	20 MHz
RC ^(1,2)	×11	(Note 1)

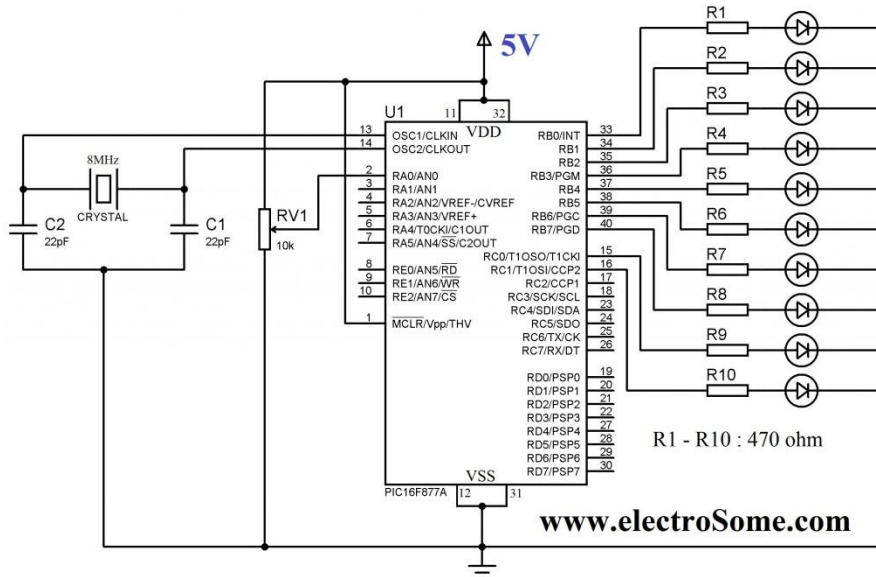
Note 1: The RC source has a typical TAD time of 4 μ s but can vary between 2-6 μ s.

Note 2: When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for Sleep operation.

ADC Clock

Selection Table – PIC 16F877A

Circuit Diagram



```
#include<pic.h>
```

```
#define _XTAL_FREQ 8000000
```

```
void ADC_Init()
```

```
{
    ADCON0 = 0x41; //ADC Module Turned ON and Clock is selected
    ADCON1 = 0xC0; //All pins as Analog Input
    //With reference voltages VDD and VSS
}
```

```
unsigned int ADC_Read(unsigned char channel)
```

```
{
    if(channel > 7) //If Invalid channel selected
        return 0; //Return 0
}
```

```

    ADCON0 &= 0xC5; //Clearing the Channel Selection Bits
    ADCON0 |= channel<<3; //Setting the required Bits
    __delay_ms(2); //Acquisition time to charge hold capacitor
    GO_nDONE = 1; //Initializes A/D Conversion
    while(GO_nDONE); //Wait for A/D Conversion to complete
    return ((ADRESH<<8)+ADRESL); //Returns Result
}

void main()
{
    unsigned int a;
    TRISB = 0x00; //PORTB as output
    TRISC = 0x00; //PORTC as output
    TRISA = 0xFF; //PORTA as input
    ADC_Init(); //Initializes ADC Module

    do
    {
        a = ADC_Read(0); //Reading Analog Channel 0
        PORTB = a; //Lower 8 bits to PORTB
        PORTC = a>>8; //Higher 2 bits to PORTC
        __delay_ms(100); //Delay
    }while(1); //Infinite Loop
}

```