

Análogo Digital ADC pic18f4550

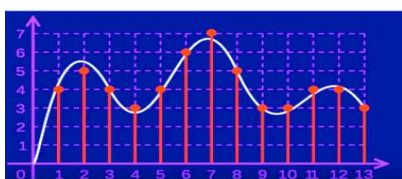
Es un circuito que convierte un valor analógico de voltaje a su correspondiente combinación en binario.



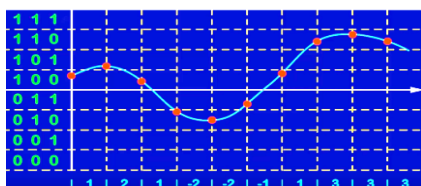
El Mc tiene incorporado un módulo de conversión de señal analógico a señal digital de 13 canales.

Pasos para la conversión.

Muestreo de señal implica tomar muestras de la señal cada periodo (T seg).

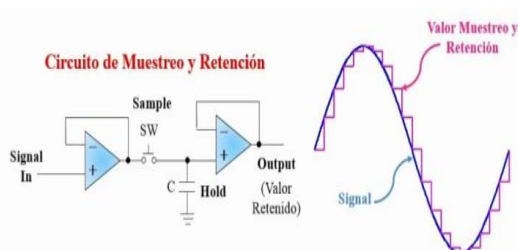


Cuantificación: Una vez que se tiene una muestra su amplitud puede tomar infinitos valores. Estos valores van a depender del número de bis que se vaya a almacenar para cada muestra.



Si se toman n bits se forman 2^n combinaciones por tanto 2^n distintos valores.

El modulo que utiliza el MC hace un muestreo y retención (sample and hold), con un condensador y luego utiliza el módulo de conversión, hasta encontrar un valor digital que iguale la tensión entregada por el conversor A/D y la tensión de entrada.



CARACTERISTICAS

- 13 canales analógicos pic18f4550 y 10 pic18f2550
- Resolución programable por software en modo de 8 bits o 10 bits.
- Señal de reloj configurable
- Tacq tiempo de adquisición configurable (0 a 20 Tad)
- TAD tiempo de conversión configurable
- Rango de tensión de conversión configurable mediante tensión de referencia.

Pines a ser configurador como Entradas Analógicas

- PORTA(5)=> AN0:RA0, AN1:RA1, AN2:RA2
AN3:RA3, AN4:RA5
- PORTB(5)=> AN8:RB2, AN9:RB3, AN10:RB1,
AN11:RB4, AN12:RB0
- PORTE(3)=> AN5:RE0, AN6:RE1, AN7:RE2

Registros asociados al móduloADC

- ADCON0: Registro de control
- ADCON1: Registro de control
- ADCON2: Registro de control
- ADRESH: Contiene la parte alta dela conversión.
- ADRESL: Contiene la parte baja de la conversión

ADCON0: Encargado de seleccionar los canales A/D y activar la conversión.

ADCON0: REGISTRO DE CONTROL A/D 0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON		

Bit5-2 CHS3:CHS0: Bits Selectores del canal Analógico.

- 0000 = Canal 0 (AN0).
- 0001 = Canal 1 (AN1).
- 0010 = Canal 2 (AN2).
- 0011 = Canal 3 (AN3).
- 0100 = Canal 4 (AN4).
- 0101 = Canal 5 (AN5).
- 0110 = Canal 6 (AN6).
- 0111 = Canal 7 (AN7).
- 1000 = Canal 8 (AN8).
- 1001 = Canal 9 (AN9).
- 1010 = Canal 10 (AN10).
- 1011 = Canal 11 (AN11).
- 1100 = Canal 12 (AN12).

Bit1 GO/DONE: Bit A/D del estado dela Conversión.

- 1= Conversión A/D en marcha.
- 0= A/D en reposo.

Bit0 ADON: Bit de permiso de A/D.

- 1= Módulo A/D permitido.
- 0= Módulo A/D no permitido.

ADCON1: Encargado de configurar las entradas como analógicas o digitales y fijar la tensión de referencia del convertor.

ADCON1: REGISTRO DE CONTROL A/D 1							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	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	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

Bits de configuración para definir los puertos como digitales / analógicos. Una vez configurados, un canal puede ser seleccionado mediante los bits CHS3:CHS0 del registro ADCON 0

ADCON2: Registro de control del ADC encargado de configurar el tiempo de adquisición, tiempo de adquisición

ADCON2: REGISTRO DE CONTROL A/D 2							
ADFM	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	Bit0
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7 ADFM: Bit de Justificación del Formato A/D.
 1= Justificación Derecha (modo-10bits).
 0= Justificación Izquierda (modo-8bits).

Bit5-3 ACQT2:ACQT0: Bit Selectores del Tiempo de Adquisición.
 111 = 20 T_{AD}
 110 = 16 T_{AD}
 101 = 12 T_{AD}
 100 = 8 T_{AD}
 011 = 6 T_{AD}
 010 = 4 T_{AD}
 001 = 2 T_{AD}
 000 = 0 T_{AD}

Bit2-0 ADCS2:ADCS0 Bits Selectores del Reloj de Conversión.
 111 = F_{RC} (Reloj de A/D deriva del Oscilador RC)
 110 = F_{RC}/64
 101 = F_{RC}/16
 100 = F_{RC}/4
 011 = F_{RC} (Reloj de A/D deriva del Oscilador RC)
 010 = F_{RC}/32
 001 = F_{RC}/8
 000 = F_{RC}/2

INTCON: Registro de control de INT

INTCON: REGISTRO DE CONTROL DE INTERRUPTONES							
GIE	PEIE	TMROIE	INTOIE	RBIE	TMROIF	INTOIF	RBIF
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7 GIE: Bit de permiso global de Interrupciones.
 1= Habilita todas las interrupciones globales.
 0= Deshabilita todas las interrupciones globales.

Bit6 PEIE: Bit de permiso de Interrupciones Periféricas.
 1= Habilita las interrupciones periféricas.
 0= Deshabilita las interrupciones periféricas.

Resolución

El rango de tensión para el ADC, preestablecidas es de 0 a 5v.

Cuando el Mc tiene configurado los voltajes de referencia con la misma alimentación (V_{SS}, V_{DD}), se asume que V_{ref+}=5v, V_{ref-}=0v.

$$Res = 1Lsb = \frac{(V_{ref+})-(V_{ref-})}{1023} = 4.89mv$$

La resolución cambia si se cambia la tensión de referencia, los voltajes a convertir siempre son positivos.

Para configurar los Vref

- VSS_VDD //Rango 0 a VDD
- VREF_VREF // Rango (Vref-) a (Vref+)
- VREF_VDD // Rango (Vref-) a Vdd
- VSS_VREF // Rango 0 a (Vref+)

Tiempo de adquisición (Tacq)

El módulo A/D del PIC incorpora un condensador de muestreo, antes de la conversión A/D debemos asegurarnos de que el condensador haya sido cargado con la tensión suministrada en el canal elegido, esto viene a ser el tiempo de adquisición (Tacq), este debe ser mayor a 2.4us

Tiempo de adquisición programable de (0 a 20 Tad)

Tiempo de Conversión (Tad)

Tiempo necesario para convertir un bit la señal analógica de entrada. Este depende de la señal de reloj que se seleccione para la conversión. Este tiempo está entre 0,8us - 25us, basado en el periodo del oscilador T_{osc} y V_{ref} 3v

Tad=1us, máximo basado en el módulo RC interno.

La forma de configurar este parámetro es mediante la siguiente sentencia.

SETUP_ADC (modo);

Modo puede ser.

- ADC_OFF
- ADC_CLOCK_DIV_2 Fosc/2
- ADC_CLOCK_DIV_8 Fosc / 8
- ADC_CLOCK_DIV_16
- ADC_CLOCK_DIV_32 Fosc / 32
- ADC_CLOCK_DIV_64
- ADC_CLOCK_INTERNAL //sincronizado con el reloj interno 32 khz

Configura Entradas Analógicas

SEUTP_ADC_PORTS (cte);

Esta función trabaja en conjunto con la operación "OR", con la configuración de voltajes de referencia, valores de la cte.

```
NO_ANALOGS 0x0F // None
ALL_ANALOG 0x00 // A0 A1 A2 A3 A5 E0 E1 E2 B2 B3 B1 B4 B0
ANO_TO_AN11 0x03 // A0 A1 A2 A3 A5 E0 E1 E2 B2 B3 B1 B4
ANO_TO_AN10 0x04 // A0 A1 A2 A3 A5 E0 E1 E2 B2 B3 B1
ANO_TO_AN9 0x05 // A0 A1 A2 A3 A5 E0 E1 E2 B2 B3
ANO_TO_AN8 0x06 // A0 A1 A2 A3 A5 E0 E1 E2 B2
ANO_TO_AN7 0x07 // A0 A1 A2 A3 A5 E0 E1 E2
ANO_TO_AN6 0x08 // A0 A1 A2 A3 A5 E0 E1
ANO_TO_AN5 0x09 // A0 A1 A2 A3 A5 E0
ANO_TO_AN4 0x0A // A0 A1 A2 A3 A5
ANO_TO_AN3 0x0B // A0 A1 A2 A3
ANO_TO_AN2 0x0C // A0 A1 A2
ANO_TO_AN1 0x0D // A0 A1
ANO 0x0E // A0
```

Luego de configurar y seleccionar el canal del módulo ADC, se puede iniciar y leer la conversión por medio de la sentencia:

READ_ADC ();

Funciones para configura ADC

1. Encabezado

```
#device ADC = 10 //ADC de 10 bits
```

2. Programa principal

```
setup_adc_ports(ANO); //Config. puertos analógicos
setup_adc(modos); // Config. Tipo de reloj, Osc
                    // interno o externo
set_adc_channel(0); //Canal ADC que se va a leer
valor = read_adc(); //Leemos el valor del ADC,
                    // valor Numero entero de 16 bits
```

El retardo de 20 microsegundos es necesario cada vez que configuramos el canal que se va a leer del ADC, es importante recordar que si estamos trabajando con el ADC de 10 bits, la variable donde se almacena el valor leído del ADC (en este caso) deberá ser un entero de 16 bits.

Para convertir este valor almacenado en voltaje necesitamos realizar una regla de 3 como se muestra a continuación:

$$\begin{aligned} \text{Si } 5\text{v} &\rightarrow 1023 \\ \text{voltaje} &\rightarrow \text{valor_digital} \\ \text{voltaje} &= \text{valor_digital} * 5.0 / 1023.0 \end{aligned}$$

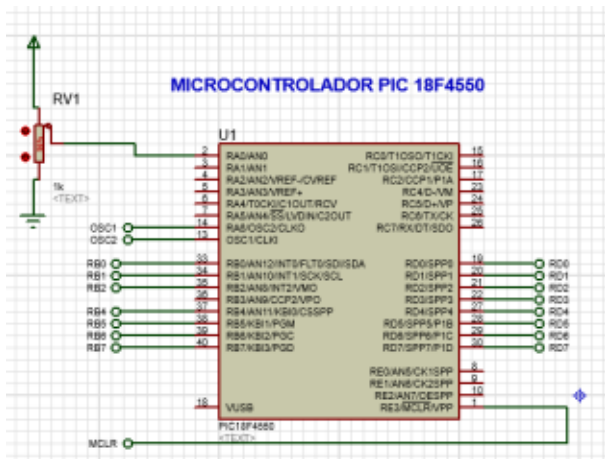
Voltaje deberá ser una variable de tipo flotante para que almacene el valor exacto de la conversión.

Ejemplo: leer la tensión de entrada controlada por el potenciómetro y mostrar su correspondiente valor binario por el portd.

```
#include <18F4550.h>
#device ADC=10 // 10 bits
#FUSES NOWDT //No Watch Dog Timer
#FUSES NOBROWNOUT //No brownout reset
#FUSES NOLVP //No low voltage prgming,
#byte portd=0xf83
#use delay(crystal=4MHz)
void main()
{
    int16 tension;
    int8 valor;
    set_tris_d(0x00);

    setup_adc_ports(ANO);
    setup_adc(adc_clock_internal);
    set_adc_channel(0);
    delay_us(10);

    while(TRUE)
    {
        tension=read_adc();
        delay_ms(1);
        valor=tension/127; //1023/8= 127 (8 leds)
        switch(valor)
        {
            case 0: portd=0b00000000;
                    break;
            case 1: portd=0b00000001;
                    break;
            case 2: portd=0b00000011;
                    break;
            case 3: portd=0b00000111;
                    break;
            case 4: portd=0b00001111;
                    break;
            case 5: portd=0b00011111;
                    break;
            case 6: portd=0b00111111;
                    break;
            case 7: portd=0b01111111;
                    break;
            case 8: portd=0b11111111;
                    break;
            default:break;
        }
    }
}
```



Bibliografía

Título: Microcontroladores PIC: Diseño práctico de aplicaciones /por José María, Angulo Usategui y Ignacio, Angulo Martínez.

Compilador C CCS y Simulador PROTEUS para Microcontroladores PIC /Eduardo García Breijo

www.microchip.com