

INFORME DE LABORATORIO 2: USO DEL CONVERSOR ANÁLOGO - DIGITAL

Luis Miguel Rincon Pinilla
e-mail: lmrinconp@itc.edu.co
Cristian Camilo Pianda Rodríguez
e-mail: ccpiandar@itc.edu.co

RESUMEN: En este informe se detalla los procedimientos seguidos y los resultados que se obtuvieron como parte del segundo laboratorio con el entorno de programación MPLAB IDE. En el cual se adquirieron las destrezas necesarias para la conversión análogo-digital usando el microcontrolador PIC18F4550 por medio de la modificación de sus registros de configuración.

PALABRAS CLAVE: Análogo, C++, Compilador, Microcontrolador, Oscilador, Programación, Registro.

1 INTRODUCCIÓN

El medio de conexión para monitorear o controlar el hardware externo a los microcontroladores son las entradas y salidas analógicas o digitales con las que cuenta. Por lo cual es necesario comprender la manera en que se configuran cada uno de los puertos y registros con los que cuenta el PIC, especialmente los registros de configuración para la conversión analógica-digital.

La conversión analógica-digital nos permite procesar en el microcontrolador las distintas señales analógicas provenientes del entorno a través de sensores.

2 OBJETIVO

Utilizar el conversor Análogo – Digital para procesar señales del ambiente y obtener la respuesta deseada en las salidas del microcontrolador

3 ELEMENTOS UTILIZADOS

Se Utilizaron los siguientes elementos durante el laboratorio:

- DIP Switch. (Simulador)
- Displays de 7 segmentos de cátodo común. (Simulador)
- Fuente de voltaje D.C. (Simulador)
- Integrados 74LS48. (Simulador)
- LEDs. (Simulador)
- Microcontrolador PIC18F4550. (Simulador)
- MPLAB X IDE 5.40.
- Potenciómetro. (Simulador)
- Proteus 8.9.
- Resistencias varias. (Simulador)
- Sensor de temperatura LM35. (Simulador)

4 DESARROLLO

A continuación, se describe el proceso de diseño y programación del sistema que da solución a cada uno de los problemas planteados.

4.1 CONVERSIÓN POTENCIOMETRO

Para el primer sistema se requería configurar el conversor análogo – digital para obtener la señal desde un potenciómetro en un rango de 0 a 5v, y procesar dicha señal para obtener el valor de voltaje en un display 7 segmentos doble con una cifra de precisión. Además, el microcontrolador tomará esta conversión para compararla con el equivalente a 2,5V. Si la entrada es mayor a 25 voltios un led debe ser encendido y si es menor el led debe estar apagado.

4.1.1 MONTAJE DEL CIRCUITO

Primeramente, se realizó el montaje del circuito con un microcontrolador PIC18F4550 (Fig. 1) con las siguientes características:

- Se conecta un potenciómetro al pin 2 (RA0).
- Para la conversión BCD a 7 segmentos se usaron dos integrados 74LS48 conectados al puerto B.
- Se conectaron dos displays 7 segmentos uno a cada uno de los 74LS48.
- Se conectó 1 Led (Indicador de Comparación) al pin 8 (RE0).

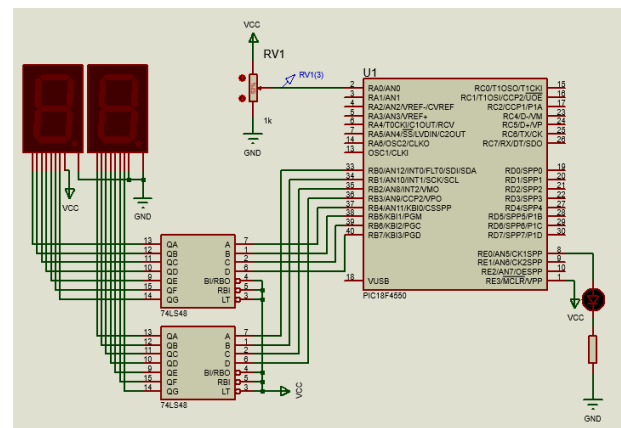


Figura 1. Montaje del Circuito 1.

4.1.2 REGISTROS DE CONFIGURACIÓN

Para habilitar la entrada análoga en AN0 era necesario modificar el registro **ADCON1** (Fig. 2) que se configuró de la siguiente manera:

- Bits 6 y 7 no es necesaria su modificación pues no están implementados.
- Bit 5 en 0 para usar Vss en vez de RA2 como Vref-.
- Bit 4 en 0 para usar Vdd en vez de RA3 como Vref+.
- Bits 3-0 como 1110 para habilitar AN0 como la única entrada análoga.

R/W-0	R/W-0	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾								
VCFG0	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0								
bit 0													
bit 7-6	Unimplemented: Read as '0'												
bit 5	VCFG0: Voltage Reference Configuration bit (Vref- source) 1 = Vref- (AN2) 0 = Vss												
bit 4	VCFG0: Voltage Reference Configuration bit (Vref+ source) 1 = Vref+ (AN2) 0 = VDD												
bit 3-0	PCFG3:PCFG0: 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							

Figura 2. Configuración ADCON1 (Pág. 262)

Para el registro ADCON0 se usarán las siguientes configuraciones:

- **CHS3:CHS0**: para este grupo de bits usaremos el valor 0 ya que solamente usaremos AN0 como la entrada analógica.
- **GO/DONE**: para este bit usaremos 1 para iniciar la conversión A/D.
- **ADON**: este bit se usará en 1 para habilitar el módulo de conversión A/D.

Para el registro ADCON2 se usarán los siguientes parámetros:

- **ADFM**: para este de bit usaremos el valor 0 para justificar hacia la izquierda ya que solamente usaremos los 8 bits más significativos de la conversión.
- **ADCS**: para este grupo de bits usaremos el valor 1 debido a:

$$T_{osc} = \frac{1}{8 \text{ MHz}} = 125 \text{ ns}$$

$$T_{AD} = 0,8 \mu\text{s}$$

$$T_{AD} = 0,8 \mu\text{s} * \frac{T_{osc}}{125 \text{ ns}} = 6,4 T_{osc} \approx 8 T_{osc}$$

- **ACQT**: este grupo de bits lo configuraremos a 4 debido a:

$$T_{AD} = 8 * 125 \text{ ns} = 1 \mu\text{s}$$

$$T_{ACQ} = 2,45 \mu\text{s}$$

$$T_{ACQ} = 2,45 \mu\text{s} * \frac{T_{AD}}{1 \mu\text{s}} = 2,45 T_{AD} \approx 4 T_{AD}$$

4.1.3 PROGRAMACIÓN

El código de programación en C que creamos para el microcontrolador es el siguiente:

```
#include <xc.h>
#define _XTAL_FREQ 8000000
#pragma config FOSC = INTOSCIO_EC

void main(void) {
    int conv;
    OSCCONbits.IRCF=0b111;
    ADCON1=0b00001110;
    TRISAbits.RA0=1;
    TRISB=0b00000000;
    TRISEbits.RE0=0;
    PORTAbits.AN0=0;
    PORTB=0b00000000;
    PORTEbits.RE0=0;
    ADCON0bits.ADON=1;
    ADCON2bits.ACQT=4;
    ADCON2bits.ADCS=1;
    ADCON2bits.ADFM=0;

    while (1){
        ADCON0bits.CHS=0x00;
        ADCON0bits.GO_DONE=1;
        conv = (ADRESH*50)/255;
        PORTB = ((conv/10)*16)+(conv%10);

        if(conv>25){
            PORTEbits.RE0=1;
        }
        else{
            PORTEbits.RE0=0;
        }
    }
    return;
}
```

4.2 CONVERSIÓN SENSOR DE TEMPERATURA LM35

Para el segundo sistema se requería configurar el conversor análogo – digital para obtener la señal desde un sensor LM35 (temperatura) el cual tiene una sensibilidad de 10mV/°C. Además, el microcontrolador tomará esta conversión para compararla con el equivalente a 176°F. Si la entrada es mayor a 176°F un led debe ser encendido y si es menor el led debe estar apagado.

Finalmente, se debe utilizar lógica combinacional agregando 3 entradas y una salida adicional que cumplan lo siguiente:

- Si la temperatura es mayor a 176°F y las entradas 1 y 2 están activadas la salida debe ser 1.

- Si la temperatura es menor a 176°F y las entradas 1 y 3 están activadas la salida debe ser 1.
- Si la temperatura es menor a 176°F y la entrada 2 está activada la salida debe ser 1.
- Si la temperatura es mayor a 176°F y la entrada 3 está activada la salida debe ser 1.
- En cualquier otro caso la salida debe ser 0.

4.2.1 MONTAJE DEL CIRCUITO

Primeramente, se realizó el montaje del circuito con un microcontrolador PIC18F4550 (Fig. 3) con las siguientes características:

- Se conecta la salida del sensor LM35 al pin 2 (AN0).
- Se conecto un módulo de 3 DIP Switch a los pines 3-5 (RA1-RA3) cada uno conectado a una resistencia de pull-down para garantizar el valor lógico.
- Se conectaron 2 Leds a los pines 8-9 (RE0-RE1).

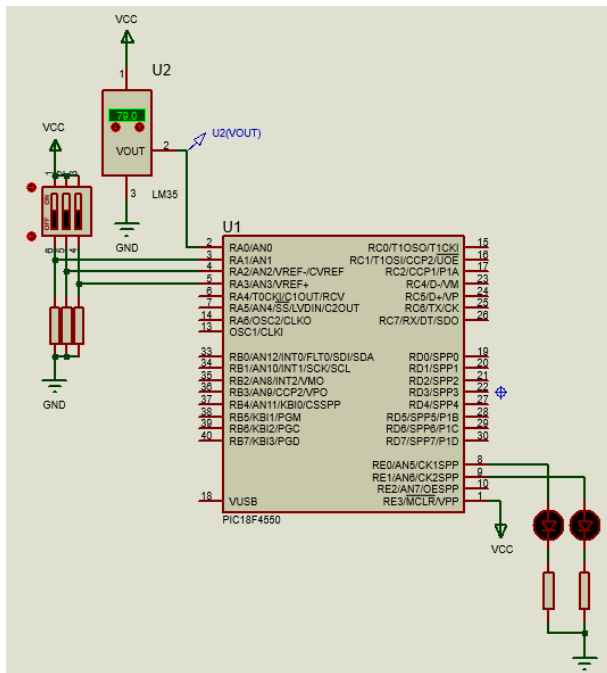


Figura 3. Montaje del Circuito 2.

4.2.2 REGISTROS DE CONFIGURACIÓN

Para este circuito utilizamos la misma configuración que en el primero con la siguiente excepción:

- **ADFM:** para este de bit usaremos el valor 1 para justificar hacia la derecha usaremos los 10 bits resultantes de la conversión.

4.2.3 PROGRAMACIÓN

El código de programación en C que creamos para el microcontrolador es el siguiente:

```
#include <xc.h>
#define _XTAL_FREQ 8000000
#pragma config FOSC = INTOSCIO_EC

void main(void) {
    int conv;
    int c;
    int f;
    OSCCONbits.IRCF=0b111;
    ADCON1=0b00001110;
    TRISA=0x0f;
    TRISEbits.RE0=0;
    TRISEbits.RE1=0;
    PORTA=0x00;
    PORTEbits.RE0=0;
    PORTEbits.RE1=0;
    ADCON0bits.ADON=1;
    ADCON2bits.ACQT=4;
    ADCON2bits.ADCS=1;
    ADCON2bits.ADFM=1;

    while (1){
        ADCON0bits.CHS=0x00;
        ADCON0bits.GO_DONE=1;
        conv = (ADRESH*256)+ADRESL;
        c = (((conv*100)+0.0)/1023)*5;
        f = ((c*9)/5)+32;

        if(f>176){
            PORTEbits.RE0=1;
        }
        else{
            PORTEbits.RE0=0;
        }

        if((f<176)&&(PORTAbits.RA1==0)&&(PORTAbits.RA2==1)&&(PORTAbits.RA3==0)){
            PORTEbits.RE1=1;
        }
        else if
        ((f<176)&&(PORTAbits.RA1==1)&&(PORTAbits.RA2==0)&&(PORTAbits.RA3==1)){
            PORTEbits.RE1=1;
        }
        else if
        ((f>176)&&(PORTAbits.RA1==0)&&(PORTAbits.RA2==0)&&(PORTAbits.RA3==1)){
            PORTEbits.RE1=1;
        }
        else if
        ((f>176)&&(PORTAbits.RA1==1)&&(PORTAbits.RA2==1)&&(PORTAbits.RA3==0)){
            PORTEbits.RE1=1;
        }
        else {
            PORTEbits.RE1=0;
        }
    }
    return;
}
```

5 Enlace del Video

El video de la simulación se encuentra en:
<https://youtu.be/5Gn3-AsRkv0>

6 CONCLUSIONES

Lo expuesto en este informe de laboratorio permite arribar a la siguiente conclusión:

- Con la configuración de los registros asignados al conversor análogo-digital (**ADCON0:ADCON2**), se logro implementar una interfaz de adquisición y análisis de información, obtenida del ambiente a través de un sensor, la cual convierte una señal voltaica variable a una palabra (byte) que es procesada por un dispositivo digital (microcontrolador), facilitando la posterior interacción de esta con otros medios digitales por medio de diversas interfaces de comunicación (Paralela, USB, TCP/IP, Bluetooth).