



**UNIVERSIDAD
NACIONAL AUTÓNOMA DE
MÉXICO**



FACULTAD DE INGENIERÍA

LABORATORIO DE MICROCOMPUTADORAS

PRACTICA #9

**“PROGRAMACIÓN EN C, CONVERTIDOR A/D E
INTERRUPCIONES”**

EQUIPO:

- **BUSTOS RAMÍREZ LUIS ENRIQUE**
- **EGUIARTE MORETT LUIS ANDRÉS**

SEMESTRE: 2017-2

Desarrollo

1.- Programa el cual obtenga una señal analógica a través del canal de su elección, se realice la conversión y el resultado de esta, la muestre en un puerto paralelo y a su vez lo trasmita al puerto serie.

```
#include <16f877.h>
#device adc=8
#fuses HS,NOPROTECT,
#use delay(clock= 20000000)
#use rs232(baud = 9600, xmit = PIN_C6, rcv=PIN_C7)
#org 0x1F00, 0x1FFF void loader16F877(void){}

int conv;

void main(){
    float a;
    setup_port_a(ALL_ANALOG);
    setup_adc(ADC_CLOCK_INTERNAL);
    set_adc_channel(a);
    delay_us(20);
    while(1){
        a= input_a();
        conv = read_adc();
        output_b(conv);
        printf("%X,%u, %2.2f \n",conv, conv, conv/51.00);
    }
}
```

Algoritmo

```
Inicio
Configurar puertos
Mientras 1 = 1 hacer
    Lee puerto a
    Lee convertidor A/D
    Manda señal a puerto B
    Imprime en pantalla los valores
Fin mientras
Fin
```

2.- Utilizando la interrupción del TIMER0, realizar un programa que transmita el resultado de la conversión cada 10 segundos.

```
#include <16f877.h>
#device ADC=8
#fuses HS,NOPROTECT,
#use delay(clock=20000000)
```

```
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#org 0x1F00, 0x1FFF
void loader16F877(void) {}
```

```
float resultado;
int contador=0;
```

```
#int_rtcc
clock_isr(){
    contador++;
    if(contador==770){
        output_b(resultado);
        printf(" valor= %2.2f \n\r",resultado);
        contador=0;
    }
}
void main(){
    setup_port_a(ALL_ANALOG);
    setup_adc(ADC_CLOCK_INTERNAL);
    set_adc_channel(0);
    set_timer0(0);
    setup_counters(RTCC_INTERNAL,RTCC_DIV_256);
    enable_interrupts(INT_RTCC);
    enable_interrupts(GLOBAL);
    delay_us(20);
    while(1){
        resultado=read_adc();
        delay_us(20);
    }
}
```

Algoritmo

```
Inicio
Configura puertos
Inicio interrupción
    Variable entera = Variable entera + 1
    Si variable_entera es igual a 770 entonces
        Imprime "Resultado del convertidor A/D"
        Variable_entera = 0
    Fin de si
Fin de interrupción
Mientras 1 = 1 hacer
    Leer Resultado del convertidor A/D
Fin Mientras
Fin
```

3.- Realizar un programa el cual constantemente transmita el resultado de la conversión a la terminal, y cada 30 segundos interrumpa la ejecución de este y envíe el siguiente texto "Laboratorio de Microcomputadoras"

```
#include <16f877.h>
#device adc=8
#fuses HS,NOPROTECT,
#use delay(clock= 20000000)
#use rs232(baud = 9600, xmit = PIN_C6, rcv=PIN_C7)
#org 0x1F00, 0x1FFF void loader16F877(void){}
long int contador;
#int_rtcc
clock_isr(){
    contador++;
    if(contador == 2166){
        printf("LABORATORIO DE MICROCOMPUTADORAS");
        delay_ms(3000);
        contador = 0;
    }
}

void main(){
    float a;
    int conv;
    contador = 0;
    set_timer0(0);
    setup_counters(RTCC_INTERNAL,RTCC_DIV_256);
    setup_port_a(ALL_ANALOG);
    setup_adc(ADC_CLOCK_INTERNAL);
    set_adc_channel(a);
    delay_us(20);
    enable_interrupts(GLOBAL);
    enable_interrupts(INT_RTCC);
    while(1){
        a= input_a();
        conv = read_adc();
        output_b(conv);
        printf("%X,%u, %2.2f \n",conv, conv, conv/51.00);
    }
}
```

Algoritmo

Inicio

Configura puertos

Inicio de Interrupción

Si variable_entera == 2166

Imprime "Laboratorio de microcomputadoras"

```

        Variable_entera = 0
    Fin de Si
Fin de Interrupción
Mientras 1 = 1 hacer
    Lee puerto a
    Lee convertidor A/D
    Manda señal a puerto B
    Imprime en pantalla los valores
Fin mientras
Fin

```

4.- Utilizando la interrupción por cambio de nivel del puerto paralelo, realizar un programa que reconozca un flanco positivo en los pines PB4, PB5, PB6 o PB7 del puerto B, y cuando se presente, envíe a la terminal el siguiente texto; de acuerdo a la entrada en la que ha ocurrido el evento.

```

#include <16f877.h>
#fuses HS,NOPROTECT,
#use delay(clock= 20000000)
#use rs232(baud = 9600, xmit = PIN_C6, rcv=PIN_C7)
#org 0x1F00, 0x1FFF void loader16F877(void){}

#int_rb
port_rb(){
    if(input(PIN_B4))
        printf("Interrupcion PB4 Activada\n");
    else
        printf("Pulso de bajada PB4\n");
    if(input(PIN_B5))
        printf("Interrupcion PB5 Activada\n");
    else
        printf("Pulso de bajada PB5\n");
    if(input(PIN_B6))
        printf("Interrupcion PB6 Activada\n");
    else
        printf("Pulso de bajada PB6\n");
    if(input(PIN_B7))
        printf("Interrupcion PB7 Activada\n");
    else
        printf("Pulso de bajada PB7\n");
}

void main(){
    set_timer0(0);
    setup_counters(RTCC_INTERNAL,RTCC_DIV_256);
    enable_interrupts(GLOBAL);
    enable_interrupts(INT_RB);
    while(1){}
}

```

Algoritmo

Inicio

Configurar puertos

Inicio de Interrupción

Si PIN_B4 = 1 entonces

Imprime "Interrupción PB4 prendida"

En caso contrario

Imprime "Pulso de bajada PB4"

Fin de Si

Si PIN_B5 = 1 entonces

Imprime "Interrupción PB5 prendida"

En caso contrario

Imprime "Pulso de bajada PB5"

Fin de Si

Si PIN_B6 = 1 entonces

Imprime "Interrupción PB6 prendida"

En caso contrario

Imprime "Pulso de bajada PB6"

Fin de Si

Si PIN_B7 = 1 entonces

Imprime "Interrupción PB7 prendida"

En caso contrario

Imprime "Pulso de bajada PB7"

Fin de Si

Fin de Interrupción

Mientras 1 = 1 hacer

Fin mientras

Fin

Conclusiones

Bustos Ramírez Luis Enrique: Esta práctica me ayudo a reforzar mis conocimientos de interrupciones y programación en C, aparte de aprender a manejar el microcontrolador y sus componenetes.