CENTRO DE CIENCIAS BÁSICAS

DEPARTAMENTO DE SISTEMAS ELECTRÓNICOS

ORGANIZACIÓN COMPUTACIONAL

4° "A"

PRÁCTICA 4

M. en CC. Juan Pedro Cisneros Santoyo

Alumno: Joel Alejandro Espinoza Sánchez

Fecha de Entrega: Aguascalientes, Ags.**, 14** de junio de 2020

Práctica 4

Objetivo

La manipulación básica del microcontrolador 8051 (en cualquiera de sus variantes). Utilizar el software Keil µVision para realizar el código en lenguaje ensamblador. Interrupciones externas

Pregunta de Investigación

¿De qué manera se puede construir el código que ejecute las instrucciones dadas de modo que pueda usarse correctamente la implementación de las interrupciones externas?

Predicción

Creo que lo más conveniente es basarse en los códigos prueba de clases y adecuarlo a actividades previamente realizadas.

Materiales

Una computadora con ciertas especificaciones.

El software Keil µVision.
El software Proteus 8.8.

Método (Variables)

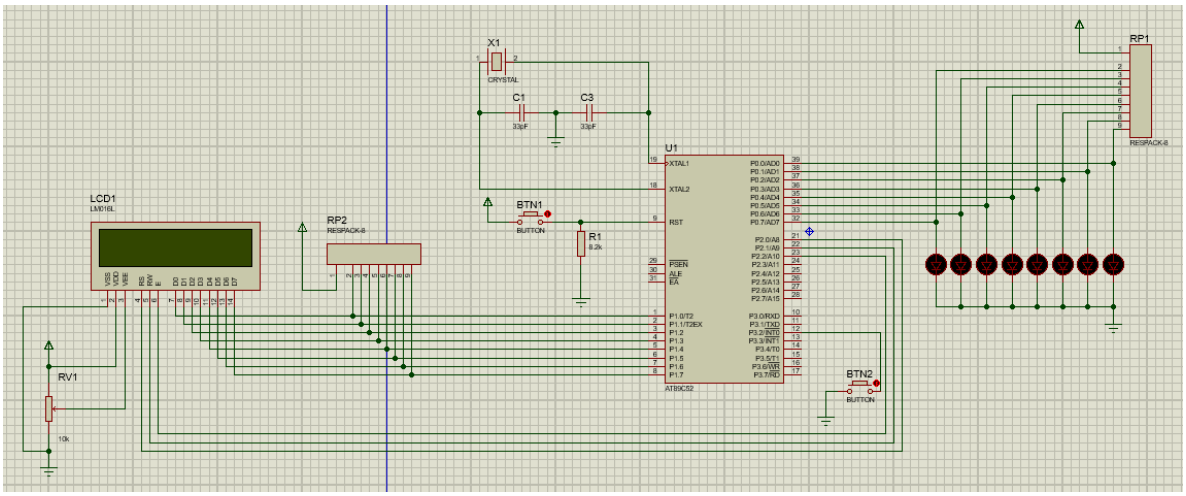
Dependiente: El LCD mostrando los mensajes y los leds en movimiento.
Independiente: La interrupción activada.
Controlada: El circuito elaborado y el programa en ensamblador.

Seguridad

No existen riesgos físicos en la elaboración de esta práctica (a excepción de las medidas de precaución con el uso de una computadora).

Procedimiento

1.- Se realizó en el software Proteus 8.8 el siguiente circuito:



2.- Se realizó en lenguaje ensamblador un programa que haga uso de las interrupciones externas del micro, con las siguientes funciones.

- El programa deberá estar corriendo el péndulo (los leds en movimiento) como programa principal siempre en dirección de derecha izquierda, cuando este llegue al MSB se deberá regresar a encender el LSB. (La lcd deberá mostrar un mensaje que diga “Principal”).
- Cuando se presione la interrupción externa el programa deberá cambiar de dirección (de izquierda a derecha, salvando la posición en la que este haya quedado para correr desde ahí) cuando este llegue al LSB se deberá estar ahí 2 segundos para después regresar al programa principal (cuando la interrupción este en uso se deberá mostrar un mensaje en la LCD que diga INTx, donde x es el número de interrupción usada).

3.- Se cargó el programa de lenguaje ensamblador dentro del microcontrolador y se probó el circuito y el programa.

Obtención y Procesamiento de Datos

Primeramente se tomó el código del péndulo donde se modificó para que el tratamiento del circuito fuera en una sola dirección. Seguido a esto se agregaron los aspectos de inicialización de la interrupción y del LCD. También se añadió un delay que se había usado en otras prácticas.

Finalmente se ingresó la interrupción, con la dirección de memoria 0003H para la interrupción y las acciones de la interrupción como se pide, con lo que daba por finalizado el código (véase anexo 1).

Referencias

- Anónimo. (2007). Microcontrolador. Mayo 2, 2020, de Wikipedia Sitio web: https://es.wikipedia.org/wiki/Microcontrolador
- Anónimo. (2013). Microcontrolador. Mayo 3, 2020, de EcuRed Sitio web: https://www.ecured.cu/Microcontrolador
- Pahrami, B. (2005). Arquitectura de Computadoras. México: McGraw Hill.
- Wackerly, J. (2008). Diseño Digital. Principios y prácticas. México: Pearson.

Anexos

Anexo 1: Código del programa.

```
leds EQU P0
led EQU P0.0
en equ P2.2
rs equ P2.0
rw equ P2.1
datos equ P1
ORG 0000H ;Reset
     LJMP inicio

ORG 0003H ;Interrupción externa 0
     ;Código para la interrupción
     CPL led

     derecha:
     ACALL print_INTX
     MOV leds,A
     RR A
     ACALL delay
     CJNE A,#01H,derecha
     LJMP wait

     wait:
     MOV leds,A
     ACALL delay
     ACALL delay
     ACALL delay
     ACALL delay

     RETI

ORG 0030H
     inicio:
     ACALL init_LCD
     ACALL init_config
     MOV A,#01H
     izquierda:
     ACALL print_principal
     MOV leds,A
     RL A
     ACALL delay
     SJMP izquierda
```

```
aqui:
AJMP aqui

print_principal:
ACALL en_h
ACALL rs_h
MOV datos,#50H ;P
ACALL en_l

ACALL en_h
ACALL rs_h
MOV datos,#52H ;R
ACALL en_l

ACALL en_h
ACALL rs_h
MOV datos,#49H ;I
ACALL en_l

ACALL en_h
ACALL rs_h
MOV datos,#4EH ;N
ACALL en_l

ACALL en_h
ACALL rs_h
MOV datos,#43H ;C
ACALL en_l

ACALL en_h
ACALL rs_h
MOV datos,#49H ;I
ACALL en_l

ACALL en_h
ACALL rs_h
MOV datos,#50H ;P
ACALL en_l

ACALL en_h
ACALL rs_h
MOV datos,#41H ;A
ACALL en_l
```

```
ACALL en_h
ACALL rs_h
MOV datos,#4CH ;L
ACALL en_l
RET

print_INTX:
ACALL en_h
ACALL rs_h
MOV datos,#49H ;I
ACALL en_l

ACALL en_h
ACALL rs_h
MOV datos,#4EH ;N
ACALL en_l

ACALL en_h
ACALL rs_h
MOV datos,#54H ;T
ACALL en_l

ACALL en_h
ACALL rs_h
MOV datos,#30H ;0
ACALL en_l
RET

init_config:
SETB EX0
SETB EA
SETB led
RET

init_LCD:
ACALL rw_l
ACALL rs_l

;Comienza la configuración de encendido de la LCD
ACALL en_h
MOV A,#38H
MOV datos,A
ACALL en_l
```

```
ACALL en_h
MOV A,#38H
MOV datos,A
ACALL en_l

ACALL en_h
MOV A,#38H
MOV datos,A
ACALL en_l

ACALL en_h
MOV A,#38H
MOV datos,A
ACALL en_l

;Apaga la pantalla
ACALL en_h
MOV A,#08H
MOV datos,A
ACALL en_l

;Limpiar pantalla
ACALL en_h
MOV A,#01H
MOV datos,A
ACALL en_l

;Modo de entrada
ACALL en_h
MOV A,#06H
MOV datos,A
ACALL en_l

;Encender la pantalla
ACALL en_h
MOV A,#0FH
MOV datos,A
ACALL en_l
RET

en_l:
CLR en
RET
```

```
      en_h:
      SETB en
      RET

      rs_l:
      CLR rs
      RET

      rs_h:
      SETB rs
      RET

      rw_l:
      CLR rw
      RET

      rw_h:
      SETB rw
      RET

      delay:
      MOV R6,#0FAH
      d1:
      MOV R7,#0F9H
      NOP
      NOP
      NOP
      NOP
      NOP
      d2:
      NOP
      NOP
      NOP
      NOP
      NOP
      NOP
      DJNZ R7,d2
      DJNZ R6,d1
      RET
END
```