CENTRO DE CIENCIAS BÁSICAS

DEPARTAMENTO DE SISTEMAS ELECTRÓNICOS

ORGANIZACIÓN COMPUTACIONAL

4° "A"

PRÁCTICA 2

M. en CC. Juan Pedro Cisneros Santoyo

Alumno: Joel Alejandro Espinoza Sánchez

Fecha de Entrega: Aguascalientes, Ags.**, 11** de mayo de 2020

Práctica 2

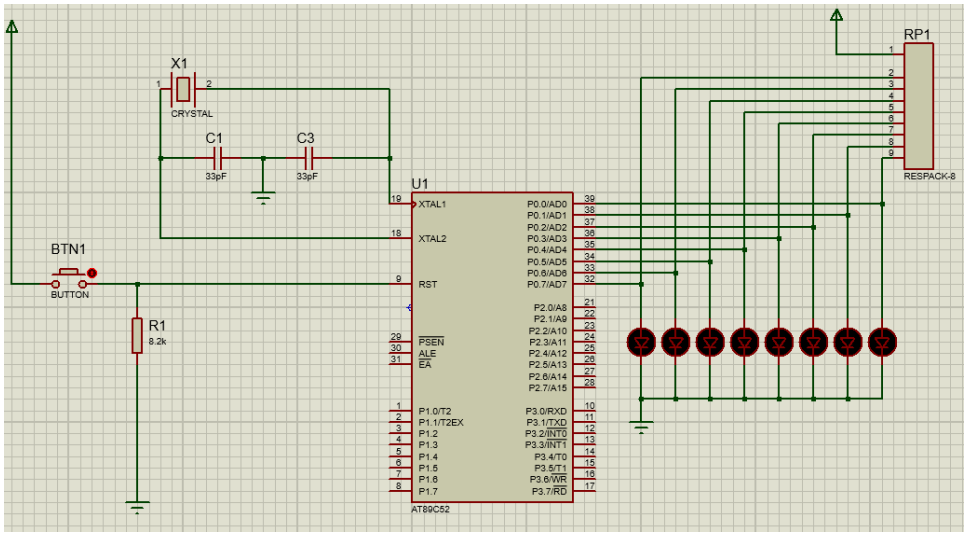
Objetivo

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

Antecedentes

Previamente se trabajó usando el mismo microcontrolador con un grupo de leds, en los que se simulaba el led que caracterizaba a la serie de televisión Knight Rider (o El Auto Increíble en Latinoamérica) emitida en la década de 1980.

Por lo que se elaboró el siguiente circuito a modo de simulación:



Sin embargo, se había planteado un código largo para el recorrido de las luces, pues tenía que comenzar de un lado, recorrer todos los leds, llegar al otro lado y regresar, para realizar este ciclo continuamente. El código propuesto tenía más de 70 líneas, donde alrededor de 40 eran para hacer el recorrido de los leds (véase anexo 1) y se pedía reducir dicho segmento de código en esta práctica.

Pregunta de Investigación

¿De qué manera se puede reducir el código dado de modo que el recorrido de leds se haga de una forma automática?

Predicción

Creo yo que la instrucción RL A y RR A que vimos en la teoría de clases será importante, así como probablemente SWAP y ADD o ANL.

Materiales

Una computadora con ciertas especificaciones.

El software Keil µVision.
El software Proteus 8.8.

Método (Variables)

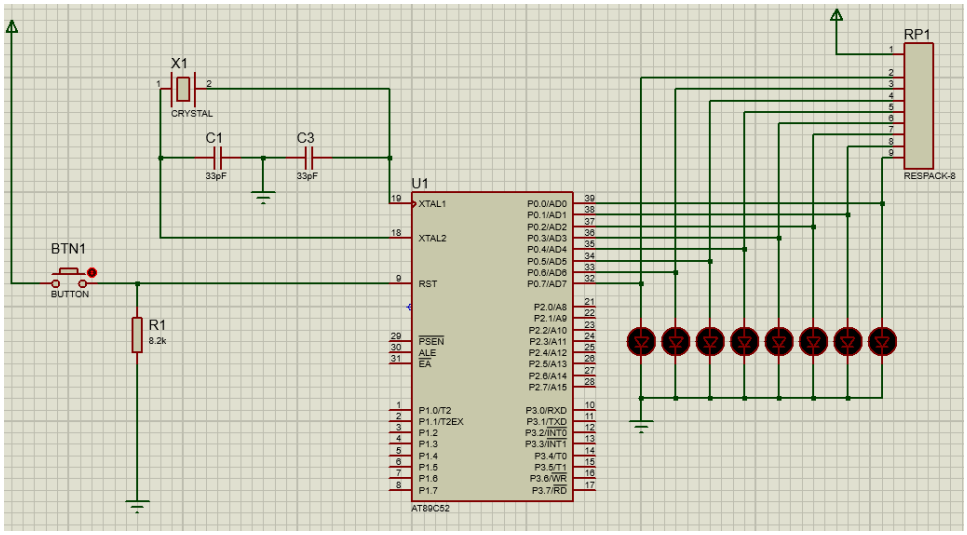
Dependiente: El recorrido de leds mostrado en el circuito.
Independiente: Las instrucciones en lenguaje ensamblador del circuito.
Controlada: El circuito elaborado.

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:

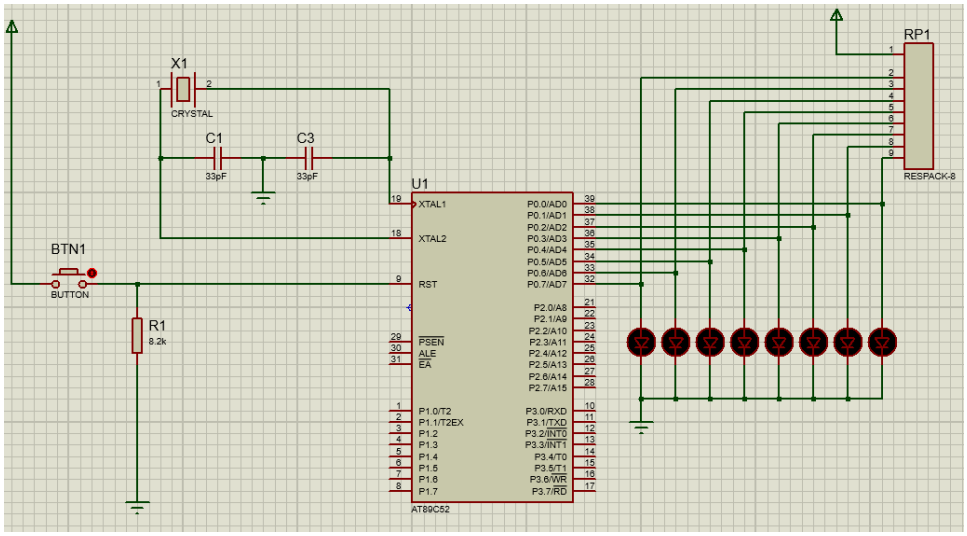


2.- Se escribió un programa en lenguaje ensamblador en Keil µVision para encender y apagar el led del circuito con ciertas modificaciones para que exista un retardo de un segundo como tiempo de encendido y apagado.

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 realizó el circuito pedido en el software Proteus, quedando de la siguiente manera:



Después se escribió el programa en Keil, el cual en un principio fue el siguiente:

```
MOV A,#01H

izquierda:
MOV leds,A
RLC A
ACALL delay
JC derecha
SJMP izquierda

derecha:
MOV leds,A
RRC A
ACALL delay
JC izquierda
SJMP derecha
```

Pues se inspiró este código de otro en clase en el cual se debía encender y apagar un led, pues se creaban dos subrutinas, una para el estado encendido y otra para el estado apagado del led.

Asimismo, se realizó una ligera modificación a la subrutina “`delay`” la cual se pedía que tuviera un retraso de 0.25 segundos. Usando de igual forma el set de instrucciones como prácticas anteriores, se redujo el delay de la práctica anterior, dejando una instrucción `DJNZ` que consumía la mitad de este tiempo, de modo que se llamó dos veces a la subrutina para generar los 250 milisegundos buscados.

Se pensó que sería análogamente similar, pues un “estado” sería cuando los leds deben avanzar a la izquierda y otro “estado” cuando ellos deben ir a la derecha.

Primeramente, se comienza asignando al acumulador el bit que será de utilidad para mover la luz a través de los leds. Después se crean estas dos subrutinas en las cuales se usan las instrucciones RRC y RRL para recorrer a la derecha y a la izquierda respectivamente el bit y finalmente, se encontró en el set de instrucciones otorgado en la misma clase que la instrucción JC permitiría realizar el salto entre subrutina y subrutina. Por lo que el propósito del programa estaba hecho en aproximadamente 10 líneas.

Conclusiones

Este programa nos fue de utilidad para poder investigar un poco más usando el set de instrucciones. Tuvimos el ejemplo en clase y con algunas pistas de cómo desarrollar esta práctica y con un poco de indagación conseguimos terminar esta práctica.

Como propuesta de mejora he de reconocer que el circuito realizar una pequeña pausa sin luz, la cual no he sabido cómo eliminar, pero se ha conseguido reducir el código, pese a este pequeño error.

Referencias

- Anónimo. (2005). Knight Rider. Mayo 7, 2020, de Wikipedia Sitio web: https://es.wikipedia.org/wiki/Knight_Rider
- 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 expuesto en clase en ensamblador

```
leds equ P0
ORG 0000H
     inicio:

     MOV A,#01H ;00000001B
     MOV leds,A
     ACALL delay
     MOV A,#02H ;00000010B
     MOV leds,A
     ACALL delay
     MOV A,#04H ;00000100B
     MOV leds,A
     ACALL delay
     MOV A,#08H ;00001000B
     MOV leds,A
     ACALL delay
     MOV A,#10H ;00010000B
     MOV leds,A
     ACALL delay
     MOV A,#20H ;00100000B
     MOV leds,A
     ACALL delay
     MOV A,#40H ;01000000B
     MOV leds,A
     ACALL delay
     MOV A,#80H ;10000000B
     MOV leds,A
     ACALL delay
     MOV A,#40H ;01000000B
     MOV leds,A
     ACALL delay
     MOV A,#20H ;00100000B
     MOV leds,A
     ACALL delay
     MOV A,#10H ;00010000B
     MOV leds,A
     ACALL delay
     MOV A,#08H ;00001000B
     MOV leds,A
     ACALL delay
     MOV A,#04H ;00000100B
     MOV leds,A
```

```
        ACALL delay
        MOV A,#02H ;00000010B
        MOV leds,A
        ACALL delay
        SJMP inicio


        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
```

Anexo 2: Código del programa propuesto en ensamblador

```
leds equ P0
ORG 0000H

      MOV A,#01H

      izquierda:
      MOV leds,A
      RLC A
      ACALL delay
      ACALL delay
      JC derecha
      SJMP izquierda

      derecha:
      MOV leds,A
      RRC A
      ACALL delay
      ACALL delay
      JC izquierda
      SJMP derecha

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

END
```