



CENTRO DE CIENCIAS BÁSICAS
DEPARTAMENTO DE SISTEMAS ELECTRÓNICOS
ORGANIZACIÓN COMPUTACIONAL
4° "A"

PRÁCTICA 5

M. en CC. Juan Pedro Cisneros Santoyo

Alumno: Joel Alejandro Espinoza Sánchez

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

Práctica 5

Objetivo

La manipulación básica del microcontrolador 8051 (en cualquiera de sus variantes).
Utilizar el software Keil μ Vision para realizar el código en lenguaje ensamblador.
Uso del timer

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 un timer?

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 μ Vision.
El software Proteus 8.8.

Método (Variables)

Dependiente: El encendido y apagado del led.

Independiente: El proceso de ejecución del timer.

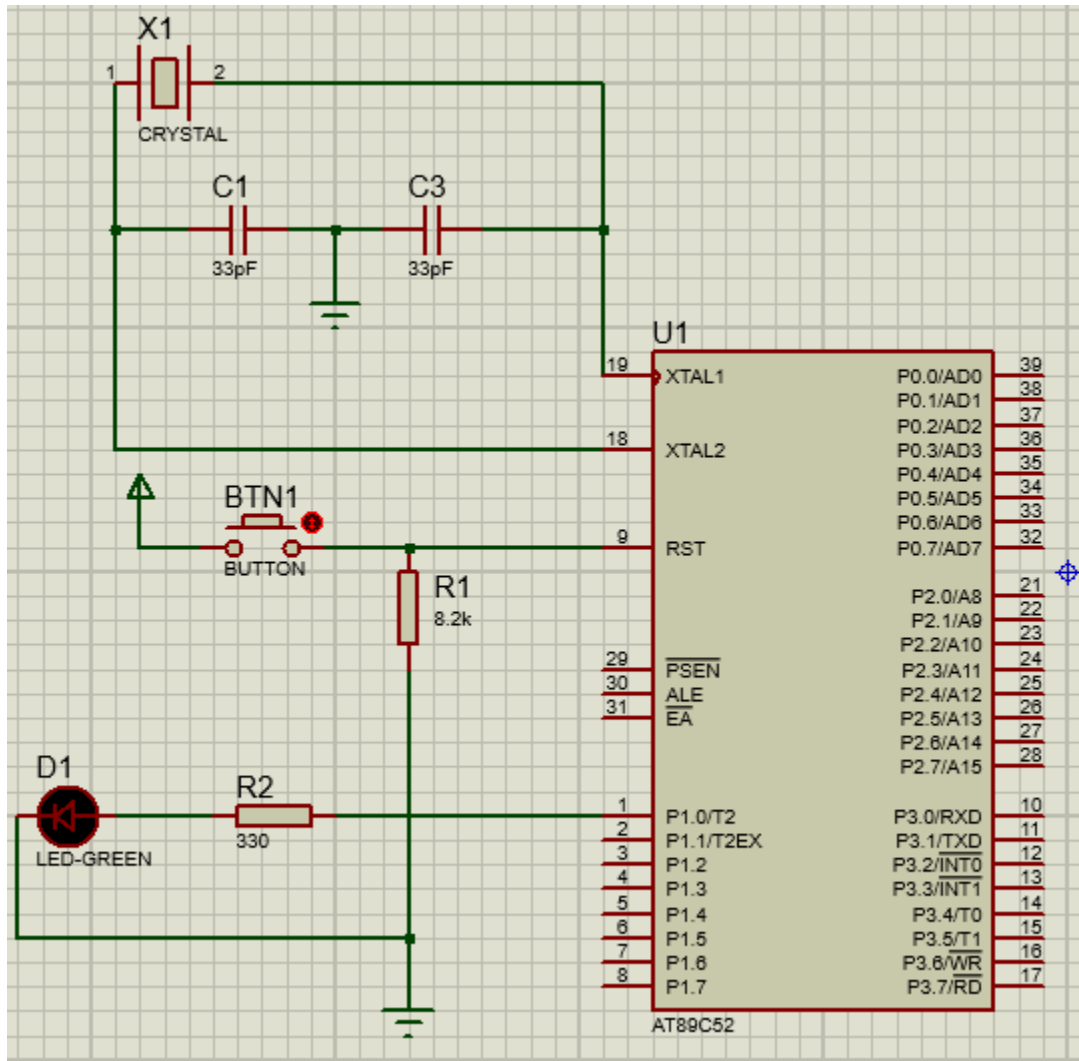
Controlada: El circuito elaborado y el programa en ensamblador.

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 hacía uso del Timer del 8052 usando como modo el modo 2 (auto recarga) donde el programa hacía toggle (encendido – apagado) en un led y el delay buscado que se generó con el Timer es de 500 ms.

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

Obtención y Procesamiento de Datos

Se comenzó con un circuito realizado previamente, realizando algunos cambios principalmente para el cambio al modo 2 del timer 1. También tuvo que cambiarse un cálculo, pues este temporizador trabaja con ocho bits. Para ello conocemos que este modo trabaja con 8 bits, es decir, tenemos 255 transiciones; esto multiplicado por los ciclos máquina obtenemos el máximo delay posible:

$$\begin{aligned} \text{delay} &= (255)(1 \times 10^{-6}) \\ &= 2.55 \times 10^{-4} \end{aligned}$$

$$= 0.255 \text{ ms}$$

Es decir que el retraso máximo que se puede hacer es de 0.255 milisegundos. Para no complicar las cosas, se decidió hacer un delay de 0.2 milisegundos. Para ello haremos un cálculo del contador:

$$\begin{aligned} count &= (0.2 \times 10^{-3})(1 \times 10^{-6}) \\ &= 184.3318 \\ &\approx 184 \end{aligned}$$

Finalmente, con estos datos, obtendremos el valor a cargar, el cual realizaremos con la siguiente fórmula:

$$\begin{aligned} valor &= (255 - 184) + 13 \\ &= 84 \\ &= 54H \end{aligned}$$

Que es lo que se carga al temporizador. Sin embargo, nosotros necesitamos un retraso de 500 milisegundos y aquí se ha realizado uno de 0.2 milisegundos. Para con estos 0.2 milisegundos se puedan conseguir 500 milisegundos, se necesitaría repetir este pequeño delay 250 veces. Para ello se cargará este valor a un registro que se repetirá en un ciclo hecho con la instrucción DJNZ (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>
- Pahrani, B. (2005). Arquitectura de Computadoras. México: McGraw Hill.
- Wackerly, J. (2008). Diseño Digital. Principios y prácticas. México: Pearson.

Anexos

Anexo 1: Código del programa.

led EQU P1.0

ORG 0000H

LJMP inicio

ORG 0030H

inicio:

MOV TMOD,#0x20 ;Activamos el modo 2 de ambos timers

ciclo:

CPL led

MOV R0,#250

ciclo1:

ACALL tim1_delay

DJNZ R0,ciclo1

AJMP ciclo

tim1_delay:

MOV TH1,#54H

SETB TR1

JNB TF1,\$

CLR TF1

CLR TR1

RET

END