



Materia: Microprocesadores y microcontroladores

Maestro: García López Jesús Adán

Alumno: Christian Alfredo Montero Martínez

Practica 7: Uso de Puertos y retardos mediante Software

Fecha: 7/04/2017

Práctica No. 7

Uso de Puertos y retardos mediante Software

Objetivo: Mediante esta práctica el alumno aprenderá la forma básica de implementar retardos por software.

Equipo:

- Computadora Personal

Teoría

Retardos por software (cálculos)

A menudo es necesario que nuestros programas usen demoras o retardos, por ejemplo, si deseamos hacer parpadear un led cada segundo evidentemente necesitaremos usar un retardo de 1s. Los retardos son prácticamente omnipresentes en nuestros programas. Los retardos por Software consisten en que el microcontrolador se quede “ciclado” durante un tiempo. Es decir, es necesario usar uno o varios contadores que deberán ser decrementados, cuando dichos contadores lleguen a 0 habrá concluido el retardo.

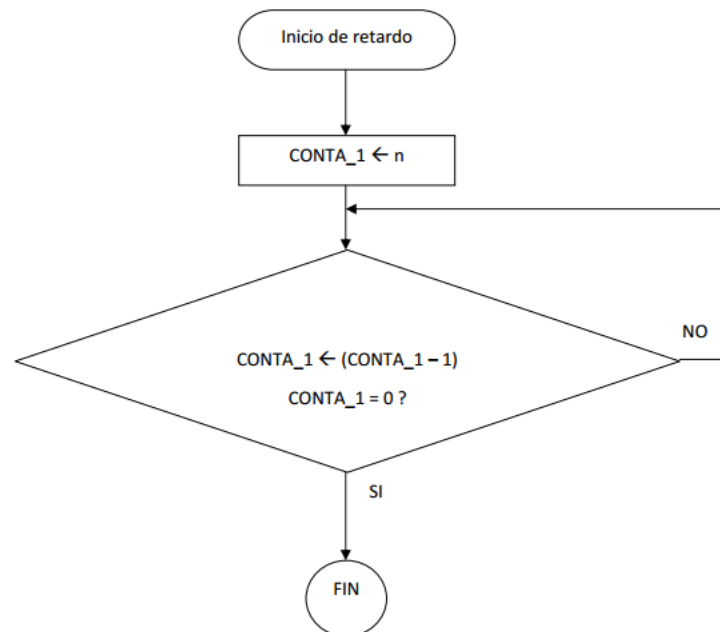


Figura No.1: Diagrama de flujo para algoritmo de retardo sencillo.

Como se puede observar en el diagrama anterior, a una variable llamada CONTA_1 se le asigna un número n , posteriormente esta variable se decrementa y se pregunta si ha llegado a 0, si no ha llegado a 0 entonces se vuelve a decrementar, y así sucesivamente hasta que llegue a 0 en cuyo caso es el FIN del retardo. El programa se quedó “perdiendo el tiempo” ciclado dando vueltas n veces.

Hasta este punto se entiende el concepto de un retardo por software, sin embargo surge una duda obvia e importante: ¿Cómo se calcula el tiempo que durará un retardo?

Para calcular el tiempo de un retardo es necesario tomar en cuenta 3 factores:

1. La frecuencia a la que opera nuestro microcontrolador.
2. Los ciclos que toma cada instrucción dentro de procedimiento retardo.
3. La cantidad de iteraciones necesarias para alcanzar el retardo.

Primer es necesario calcular el tiempo que le toma a nuestro controlador completar un ciclo de reloj, utilizando la ecuación de la frecuencia, como se muestra a continuación:

$$T = \frac{1}{16MHz} = 62.5nS$$

Ya que conocemos la cantidad de tiempo que le requiere al microcontrolador completar un ciclo de reloj, es necesario analizar el procedimiento delay de manera genérica, es decir tal como se muestra en el diagrama de flujo, únicamente con el decremento y la comparación, revisar el número de ciclos de reloj que requiere cada instrucción y determinar el tiempo que tarda en realizarse cada iteración. Después para realizar un retardo en particular, se divide el tiempo requerido entre el tiempo que tarda una iteración para determinar cuántas iteraciones se necesitan. Dependiendo la cantidad de tiempo necesario se deben anidar bucles y también se pueden utilizar instrucciones NOP para consumir más ciclos de reloj. De manera genérica se determinaron las ecuaciones para 1, 2 y 3 ciclos anidados sin operaciones NOP obteniendo los siguientes resultados:

1. $3N + 3NM$
2. $3N + 3NM + 3NMX$
3. $3N + 3NM + 3NMX + 3NMXY$

A continuación se muestra el desarrollo de las ecuaciones para lograr obtener los resultados anteriores:

a) Para un ciclo anidado sin NOP's

Para m:

$$m + 2(m - 1) + 1 = 3m - 1$$

Para n:

$$1 + 2n + n(3m - 1) + 2(n - 1) + 1$$

$$2 + 2n + 3nm - n + 2n - 2$$

$$3n + 3mn$$

b) Para dos ciclos anidados sin NOP's

Para x:

$$x + 2(x - 1) + 1$$

$$3x - 1$$

Para m:

$$2m + m(3x - 1) + 2(m - 1) + 1$$

$$3m + 3mx - 1$$

Para n:

$$1 + 2n + n(3m + 3mx - 1) + 2(n - 1) + 1$$

$$1 + 2n + 3nm + 3nm x - n + 2n - 2 - 1$$

$$3n + 3nm + 3nm x$$

c) Para tres ciclos anidados sin NOP's

Para y:

$$y + 2(y - 1) + 1$$

$$3y - 1$$

Para X:

$$2x + x(3y - 1) + 2(x - 1) + 1$$

$$3x + 3xy - 1$$

Para m:

$$2m + m(3x + 3xy - 1) + 2(m - 1) + 1$$

$$3m + 3mx + 3mxy - 1$$

Para n:

$$1 + 2n + n(3m + 3mx + 3mxy - 1) + 2(n - 1) + 1$$

$$3n + 3nm + 3nm x + 3nmxy$$

Para determinar el número de ciclos que se necesitaban ejecutar para alcanzar el retardo requerido, se dividió el tiempo solicitado correspondiente a cada retardo entre el tiempo que le toma al microcontrolador ejecutar cada ciclo como se puede apreciar a continuación:

a) 234 us

$$\frac{234\mu s}{62.5ns} = 3474 \text{ ciclos}$$

b) 5 ms

$$\frac{5ms}{62.5ns} = 80000 \text{ ciclos}$$

c) 666ms

$$\frac{666ms}{62.5ns} = 10656000$$

Con esta información y las ecuaciones previamente encontradas se procedió con la inclusión de NOP's en diferentes instancias del programa para encontrar alcanzar el retardo deseado.

Conclusiones y Comentarios

El análisis de las ecuaciones es importante antes de iniciar con la tarea de la programación, pese a que existen diversas alternativas para encontrar número de veces que debe de iterar cada ciclo, considero que la mejor manera es partiendo de la ecuación más sencilla que es aquella que no involucra operaciones NOP, posterior a esto se debe igualar la ecuación al tiempo deseado y en base a los divisores del número se deben ir asignándole valores a las variables involucradas. Para el análisis particular que realicé trataba de ir reduciendo la ecuación, dándole valor a las variables en función a los divisores del número y al llegar a la última instancia trataba de acomodar la menor cantidad de NOP's en sección del programa más conveniente.

Bibliografía

[1] Joel Oswaldo Campos Pérez. (2011). Curos básico de Retardos por Software. 05/04/2017, de INVENTRONICA Sitio web: https://campusvirtual.univalle.edu.co/moodle/pluginfile.php/161484/mod_resource/content/0/clase_3/Generar-retardos-para-PIC-en-MPLAB.pdf

[2] Juan Ignacio Huircán. (2011). Lenguaje C orientado a Microcontroladores. 05/04/2017, de. Sitio web: <http://quidel.inele.ufro.cl/~jhuircan/CCOMPILER/Lenguaje%20C%20orientado%20a%20microcontroladores%20rev%201.pdf>