

Microcontroladores

Laboratorio Dirigido N° 2



Actividades:

- Conocer los métodos para generar retardos de tiempo.
- Implementar subrutinas de retardos básicas.
- Conocer la librería #include "retardos.inc".
- Conocer el comportamiento del contador de programa (PC) del microcontrolador mediante un código en lenguaje ASM.
- Conocer los registros y las instrucciones para el manejo de tablas en la memoria de programa.



Rutinas de Retardo



1) Rutinas de Retardo:

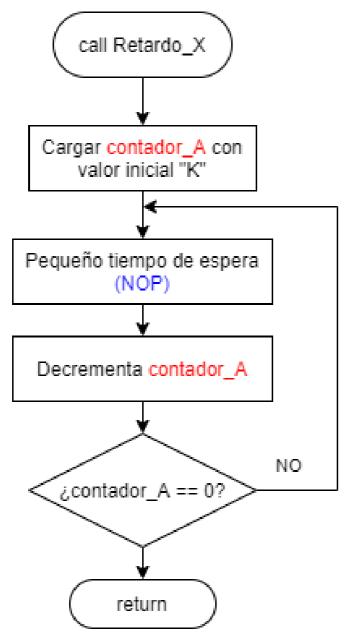
En muchas aplicaciones con microcontroladores, se requiere generar tiempos de espera muy específicos, llamados también *tiempo de retardo o delays*. Estos intervalos de tiempo pueden conseguirse mediante una *subrutina de retardos*, basada en un lazo simple para un conjunto de instrucciones que se repitan tantas veces sea necesario hasta conseguir el retardo deseado.

¿Y como hago un lazo simple para un conjunto de instrucciones que se repitan?

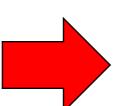




2) Rutinas de Retardo: Por lazo simple



¿Y como defino el tiempo?



Código en ensamblador del diagrama de flujo

Diagrama de flujo de una subrutina de retardo con un lazo simple.



2) Rutinas de Retardo: Por lazo simple

```
Retardo X: ;La llamada a subrutina => 2 C.I
    movlw .k ;k = literal
                                              => 1 C.I
                                         => 1 C.I
    movwf contador A ; contador A = k
Retardo:
                    ;Lazo Interno
                                         => k*C.I
                     ;No operacion
    nop
    decfsz contador A,F ;No Skip => (k-1) *C.I + Skip => 2 C.I
         Retardo ;Salta al lazo interno => (k-1)*2C.I
    goto
                      ;Fin subrutina y retorna => 2 C.I
    return
T = 2C.I + 1C.I + 1C.I + k*C.I + (k-1)*C.I + 2C.I + (k-1)*2C.I + 2C.I
                       T = 5C.I + 4kC.I
                                      \mathbf{k} = \frac{\mathbf{T} - (5C.I)}{4 * CI}
                      T = (5 + 4k)C.1
```



2) Rutinas de Retardo: Por lazo simple

Ejemplo:

Calcular el valor de la constante k, para obtener una subrutina de retardo de 500us, para un *ciclo de instrucción (C.I) = 1us*.

OJO

No se tomo en cuenta el C.I, ya que es de 1 µs, y para simplificar los cálculos, se hará todo en función de µs.

$$k = \frac{T - (5C.I)}{4C.I} = \frac{500 - 5}{4} = 123.7 \approx 123$$



$$T = (5 + 4k) = 5 + 4(123) = 497 \mu s$$

Para T
$$\approx$$
 1ms => k = 249.

Para T
$$\approx$$
 2ms => k = 498. \times

Para T
$$\approx$$
 3ms => k = 748. \times

¿Qué hago ahora si deseo mas TIEMPO?



3) Rutinas de Retardo: Por lazos anidados

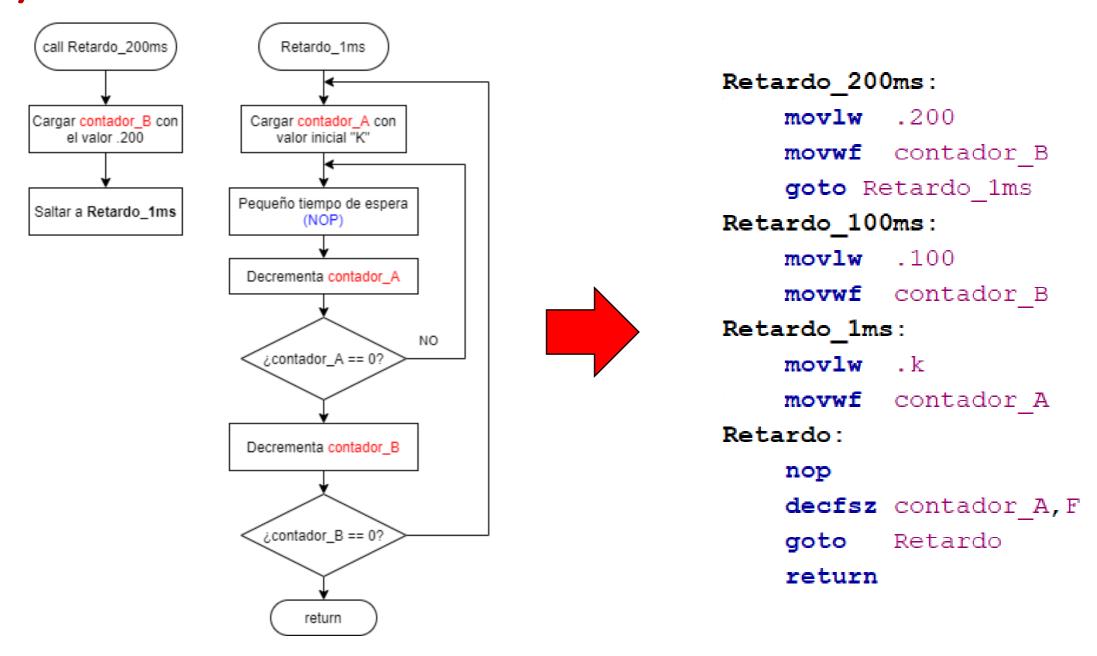


Diagrama de flujo de una subrutina de retardo con lazos anidados.

Código en ensamblador del diagrama de flujo



4) Rutinas de Retardo: Librería para Retardos

Los retardos son probablemente las subrutinas mas utilizadas dentro de los programas, es por ello importante disponer de una buena librería de subrutinas con muchos tiempos de temporización.

- Esta librería cuenta con subrutinas de retardo de 4 us hasta 20 segundos.
- Incluir esta libreria con el nombre #include "retardos.inc".



Ejemplo 1:

Ejemplo 1.1:

Conectar 8 LEDs al puerto D, realizar un código en lenguaje ensamblador para parpadear los LEDs cada 200ms, elaborar las rutinas de retardo por lazos anidados.

Ejemplo 1.2:

Realizar el ejemplo anterior pero usando la librería "retardos.inc".



Ejemplo 2: Contador de Programa(PC)

Programa para interactuar con el CONTADOR DE PROGRAMA (PC)

#include "p18f4550.inc" CONFIG FOSC = XT XT ; Oscillator Selection bits CONFIG PWRT = ON ; Power-up Timer Enable bit (PWRT disabled) CONFIG BOR = ON ; Brown-out Reset Enable bits CONFIG BORV = 3 ; Brown-out Reset Voltage bits (Minimum setting 2.05V) CONFIG WDT = OFF ; Watchdog Timer Enable bit (WDT enabled) CONFIG PBADEN = OFF ; PORTB A/D Enable bit CONFIG MCLRE = ON ; MCLR Pin Enable bit CONFIG LVP = OFF ; Single-Supply ICSP Enable bit (Single-Supply ICSP enabled) org 0x000 goto MAIN org 0x0300 MAIN: setf LATD :PC=0x300 movlw b'10000000';PC=0x302 movwf TRISD;PC=0x304 clrf 0x22;PC=0x306 movlw 0x03 ;PC=0x308 movwf PCLATH; PC=0x30A

INICIO:

movf 0x22,W;PC=0x30C

call TABLA;PC=0x30E

movwf LATD;PC=0x312

call RETARDO;PC=0x314

incf 0x22,f;PC=0x316

incf 0x22,f;PC=0x318

movlw .20;PC=0x31A

cpfseq 0x22;PC=0x31C

goto INICIO;PC=0x31E

clrf 0x22;PC=0x322

goto INICIO;PC=0x324

addwf PCL,f ;PC=0x328

retlw b'01000000';0;PC=0x32A

retlw b'01111001';1;PC=0x32C

retlw b'00100100';2;PC=0x32E

retlw b'00110000';3;PC=0x330

retlw b'00011001';4;PC=0x332

retlw b'00010010';5;PC=0x334

retlw b'000000011';6;PC=0x336

retlw b'01111000';7;PC=0x338

retlw b'000000000';8;PC=0x33A

retlw b'00011000';9;PC=0x33C

TABLA:

RETARDO: movlw .250 movwf 0x21 RET1: movlw .200 movwf 0x20 RET: nop nop nop nop decfsz 0x20,f goto RET decfsz 0x21,f goto RET1 return



Ejemplo 2: Actividad

- Explicar las rutinas que contiene el programa anterior.



Ejemplo 3: Display 7 segmentos

Conectar un display de 7 segmentos tipo ánodo común por medio de un convertidor BCD a 7segmentos (74LS47) a los 4 bits LSB del puerto D, y un pulsador en el pin RBO del puerto B, realizar un código en lenguaje ensamblador que permita incrementar un contador(0-9) por medio del pulsador y visualizar el valor del contador en el display de 7 segmentos.

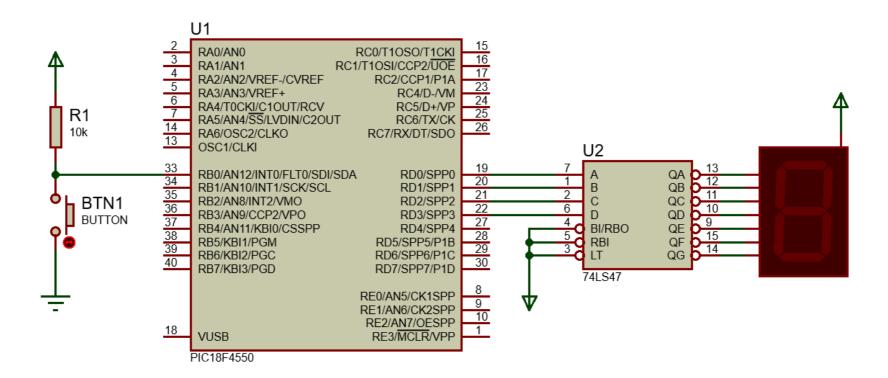


Fig.2. Circuito propuesto para el ejemplo3



Ejemplo 3: Actividades

1) Dibujar el diagrama de Flujo.



Ejemplo 3: Actividades

2) ¿Cuál es la función de la instrucción cpfslt 0x10?

3) ¿Qué cambio se haría para que el contador vaya de 0 a 5?

4) ¿Qué cambios haría para que el contador vaya de 2 a 8?



Ejemplo 4: Tabla de datos

Conectar un display de 7 segmentos tipo ánodo común al puerto D y realizar un código en lenguaje ensamblador que incremente una cuenta cada 500ms y envié la cuenta por el puerto D en formato 7 segmentos. Esta permitido el uso de la librería #include "retardos.inc".

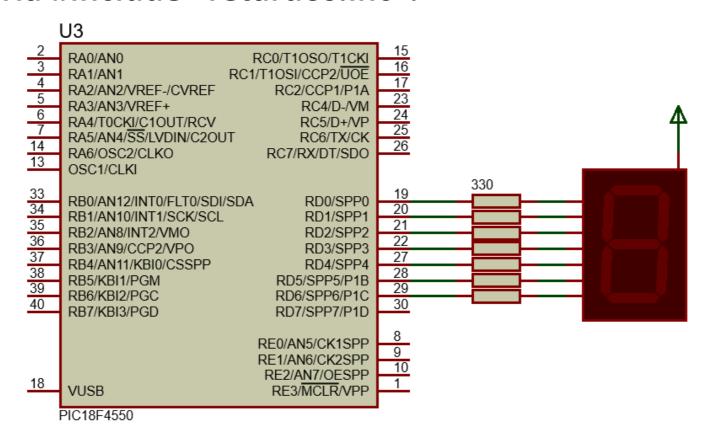


Fig.2. Circuito propuesto para el ejemplo3



Ejemplo 4: Actividades

1) Dibujar el diagrama de Flujo.



GRACIAS