

carguemos un registro CONTADOR y hagamoslo decrementar para generar un retraso. Tenemos un cristal de 4 MHz, por lo tanto tenemos 1 [microseg] por cada ciclo de instrucción.

CONTADOR	EQU	08h		
	MOVLW	H'FF'	;mueve 255 al registro contador	
	MOVWF	CONTADOR	;mueve el contenido de W a contador	
RETRASO	DECFSZ	CONTADOR,1	;dcrementa CONTADOR en una unidad,almacenalo en w.	(1)
	GOTO	RETRASO	;y salta si es zero.	(2)
	END			

Calcule el tiempo de retraso

$$255 \left( 1[\mu s]_{decfsz} + 2[\mu s]_{goto} \right) = 765[\mu s]$$

Cargando el registro W con 255 vemos que se obtienen 765 micro segundos, supongamos que necesitamos 1 [ms] de retraso. La primera estrategia pensada seria hacer un retraso de 0.5 [microsegundos] y repetirlo dos veces, para completar el milisegundo.

La ecuacion queda como sigue:

$$CONTADOR \left( 1[\mu s]_{decfsz} + 2[\mu s]_{goto} \right) = 500[\mu s]$$

$$CONTADOR = \frac{500[\mu s]}{\left( 1[\mu s]_{decfsz} + 2[\mu s]_{goto} \right)}; CONTADOR = \frac{500}{3}$$

$$CONTADOR = 166.6$$

$$CONTADOR = 166_d = A6_h$$

cargando el registro contador con A6h en nuestra rutina obtenemos 500 [microsegundos]. Ahora necesitamos realizarla dos veces para obtener el tiempo de 1

[ms], así que necesitamos otro contador para que haga la cuenta doble. (pero no se puede jajaja, mejor ...)

CONTADOR1	EQU	08h		
	MOVLW	H'A6'	;mueve 255 al registro contador	
	MOVWF	CONTADOR1	;mueve el contenido de W a contador	
CALL	RETRASO		;retraso ½ ms	(2)
CALL	RETRASO		;retraso ½ ms	
	END			
RETRASO				
LAZO	DECFSZ	CONTADOR1,1	;dcrementa CONTADOR en una unidad,almacenalo en el mismo registro	(1)
	GOTO	LAZO	;y salta si es zero.	(2)
	RETURN			(2)

Recalculando:

$$2[\mu s]_{call} + 166 \left( 1[\mu s]_{decfsz} + 2[\mu s]_{goto} \right) + 2[\mu s]_{return} = 502[\mu s]$$

y como repetimos 2 veces

$2(502[\mu s]) = 1004[\mu s] = 1.004[ms]$ , el cual es un tiempo aceptable para ciertas aplicaciones.

Si necesitaramos 10[ms]:

```
MOVLW    H'0A'  
MOVWF    CONTADOR2  
  
DECFSZ   CONTADOR2,1  
CALL     RETRASO1ms    ;retraso 1 ms  (2)  
GOTO  
  
END
```