

Microcontroladores

Semestre: 2022-1

Profesor: Kalun José Lau Gan

Semana 7: Caso reloj, conversión A/D y Multitarea en microcontroladores

1

¿Preguntas previas?

- Si el PIC18F4550 esta entrando a la etapa de madurez. ¿Por qué no se usa el PIC18F45K50?

Part Number: PIC18F4550-I/P

Upload your code to this device for as low as \$0.23

Lead Count: 40

Package Type: PDIP

Temp Range: -40C to +85C



Standard Pricing ?

(Buy Now Price, Any Volume)

*Estimated Pricing ?

(Requires Approved Quote)

Product Details

Order Quantity

1-24	\$4.76
25-99	\$4.64
100+	\$4.54

USD per Unit

1000-4999	*\$4.17
5000+	*\$3.96

*Request Quote for Larger Quantities

Part Number: PIC18F45K50-I/P

Upload your code to this device for as low as \$0.23

Lead Count: 40

Package Type: PDIP

Temp Range: -40C to +85C



Standard Pricing ?

(Buy Now Price, Any Volume)

*Estimated Pricing ?

(Requires Approved Quote)

Product Details

Order Quantity

1-24	\$2.93
25-99	\$2.69
100+	\$2.43

USD per Unit

1000-4999	*\$2.24
5000+	*\$2.13

*Request Quote for Larger Quantities

2

¿Preguntas previas?

- Si yo consigo el PIC18F45K50, puedo usar los mismos códigos e información que usé en el PIC18F4550?
 - No, son modelos diferentes por lo que tendrán una configuración diferente para cada modelo. Los aspectos funcionales de los periféricos serán muy similares.
- ¿Dónde puedo ver acerca de los nuevos modelos de PIC18 que hay?
 - <https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus/pic-mcus>
- ¿Cuál me recomendaría para futuras aplicaciones?
 - El PIC18F45Q43

Part Number: PIC18F45Q43-I/P

Upload your code to this device for as low as \$0.23

Lead Count: 40	Package Type: PDIP
----------------	--------------------

 Standard Pricing: [?](#)
(Buy Now Price, Any Volume)

Product Details	Order Quantity	USD per Unit
	1-24	\$2.33
	25-99	\$2.15
	100+	\$1.95

3

Preguntas previas:

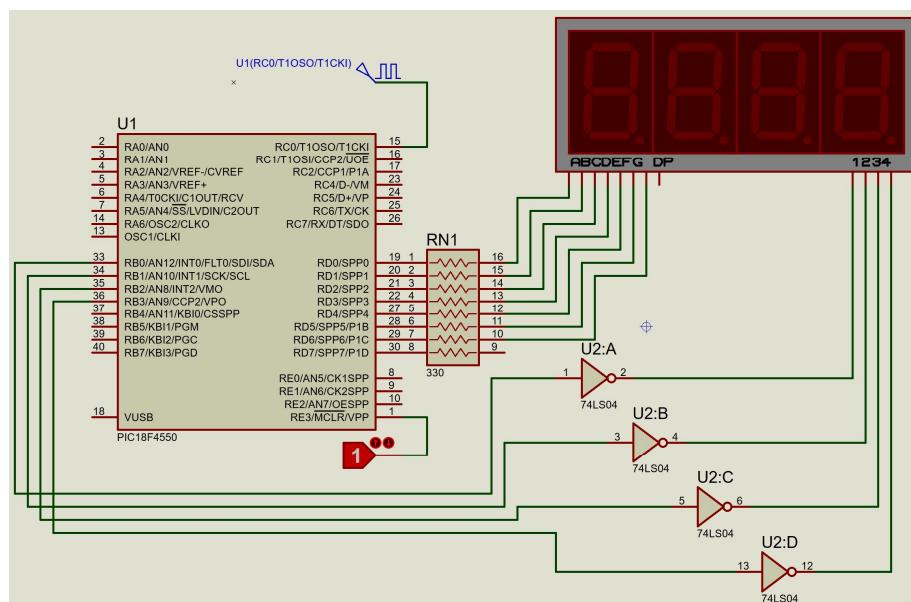
4

Agenda:

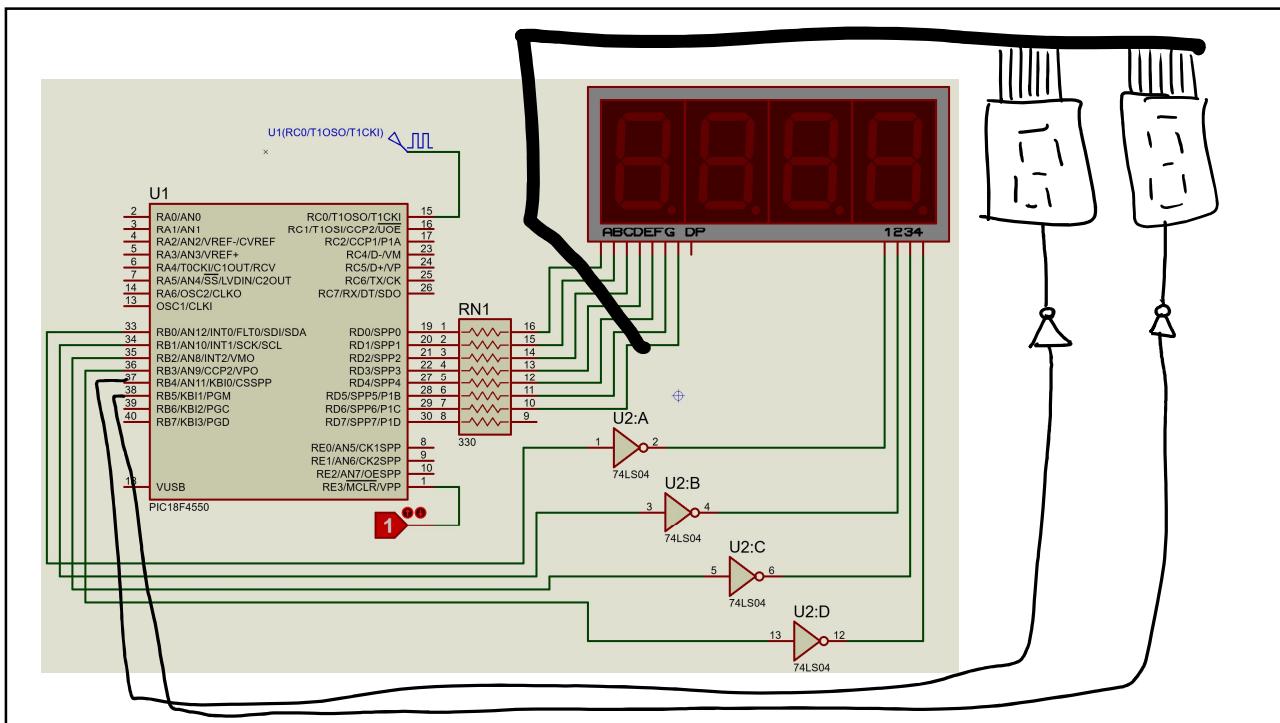
- Caso: Reloj con el Timer1 en RTC y display de cuatro dígitos de siete segmentos
- Conversión A/D
- Multitarea
- Múltiples interrupciones
- Ejemplos de multitarea

5

Diseño: Reloj en el PIC18F4550



6



7

Diseño: Reloj en el PIC18F4550

- El display es del tipo multiplexado de 4 dígitos de siete segmentos cátodo común.
 - Revisar rutina para la decodificación de números en siete segmentos (se recomienda el empleo de TBLPTR).
 - Revisar rutina de multiplexación en displays de siete segmentos.
 - Revisar rutina de individualización de dígitos de un registro de 8 bits.
- El módulo Timer1 en modo RTC.
 - Contemplar el uso del cristal 32.768KHz (en implementación) o fuente de reloj de 32.768KHz (en simulación en Proteus)
 - Revisar rutina de interrupción para asegurarse de que se cumpla la carga de cuenta inicial dentro de los 15.25μs.
- Sistema de cuenta del reloj.
 - Considerando Sistema 24H: 00:00:00 – 23:59:59
 - Opción de ingreso de la hora actual (ej: Interrupciones externas)
 - Estrategia para poder visualizar el formato completo de la hora

8

Avance del código:

Avance del código:

```
1      PROCESSOR 18F4550
2      #include "cabecera.inc"
3
4      PSECT reloj, class=CODE, reloc=2, abs
5
6      ;Etiquetas a los GPR
7      horas      EQU 000H
8      minutos    EQU 001H
9      segundos   EQU 002H
10     digbyte_temporal EQU 003H
11     digbyte_centena  EQU 004H
12     digbyte_decena   EQU 005H
13     digbyte_unidad   EQU 006H
14
15     reloj:
16         ORG 00000H
17         goto configuro
18
19         ORG 00008H
20         goto TMRI_ISR
21
22         ORG 00018H
23         goto TMRO_ISR
24
25         ORG 00400H
26
27         tabla7s:  db 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x67
28
29         ORG 00020H
30         configuro:
31             movlw 0x80
32
33             movwf TRISD      ;RD6-RD0 comc
34             movlw 0xF0        ;RB3-RB0 comc
35             clrf LATB        ;Condicion ir
36             movlw 10
37             movwf horas
38             movlw 12
39             movwf minutos
40             movlw 34
41             movwf segundos ;Configuracicn
42             movwf HIGH tabla7s
43             movwf TBLPTRH
44             movlw LOW tabla7s
45             movwf TBLPTRL ;TBLPTR apunt
46
47             loop:
48                 movff minutos, digbyte_temporal
49                 call digbyte
50                 movf digbyte_decena, 0
51                 call decodifica
52                 bsf LATB, 0
53                 nop
54                 bcf LATB, 0
55                 movf digbyte_unidad, 0
56                 call decodifica
57                 bsf LATB, 1
58                 nop
59                 bcf LATB, 1
60                 movff segundos, digbyte_temporal
61                 call digbyte
62                 movf digbyte_decena, 0
63                 call decodifica
64                 bsf LATB, 2
65                 nop
66                 bcf LATB, 2
67                 movf digbyte_unidad, 0
68                 call decodifica
69                 bsf LATB, 3
70                 nop
71                 bcf LATB, 3
72
73             cirlf TBLPTRL
74             goto loop
75
76             decodifica:
77                 movwf TBLPTRL
78                 TBLRD*
79                 movwf TABLAT, LATD
80                 return
81
82             digbyte:
83                 clrf digbyte_centena
84                 clrf digbyte_decena
85                 clrf digbyte_unidad
86                 movf digbyte_temporal, 0
87                 movlw 100
88
89             otro1:
90                 incf digbyte_centena, 1
91                 subwf digbyte_temporal, 1
92                 btfsr STATUS, 0 ;Pregunto
93                 goto otro1
94                 decf digbyte_centena, 1
95                 addwf digbyte_temporal, 1
96
97             otro2:
98                 movlw 10
99                 incf digbyte_decena, 1
100                subwf digbyte_temporal, 1
101                btfsr STATUS, 0 ;Pregunto
102                goto otro2
103                decf digbyte_decena, 1
104                addwf digbyte_temporal, 1
105                movwf digbyte_temporal, digbyte_unidad
106                return
107
108             TMRI_ISR:
109                 retfie
110
111             TMRO_ISR:
112                 retfie
113
114             end reloj
```

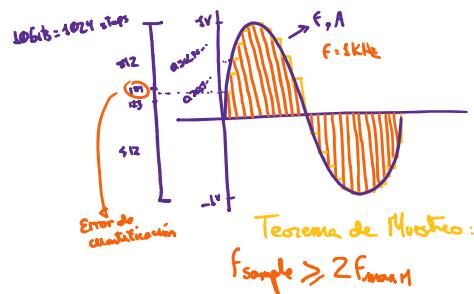
9

Repaso de conocimientos:

¿Tipos de señales? → Continuas
Discretas

¿Por qué digitalizamos series?

preservar (almacenar) } Mejor que
procesar ✓ en enclógico
transferir



Audios: $200\text{Hz} \sim 20\text{kHz}$ \rightarrow Calidad CD en audio digital
 $f_s = 44100\text{Hz}$ 16 bits

Conversor A/D

Revisar Capítulo 21 de la hoja técnica del PIC18F4550

- Resolución:
- Cantidad de canales analógicos:
- Tiempo de adquisición:
- Rango de voltaje de entrada:
- ¿Cuáles son los valores límites de Vref+ y Vref-?
- Proceso de adquisición de una señal analógica
- ¿Interviene el teorema de muestreo?
- ¿Hay interrupciones?

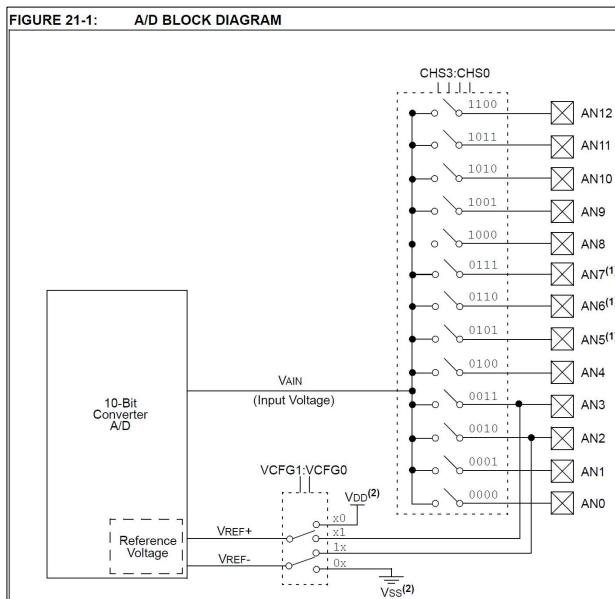
11

Conversor A/D

- Resolución: 10bits (ADRESH:ADRESL)
 - Posee un bit ADFM (justificación del resultado)
- Cantidad de canales analógicos: 13
- **Se lee un canal analógico a la vez**
- ¿Interviene el teorema de muestreo? Si.
- Tiempo de adquisición: se configura en reg. ADCON2
- Rango de voltaje de entrada: 0-5V? (Vref+ = VDD, Vref- = VSS)
- Proceso de adquisición de una señal analógica (ver datasheet)
- ¿Hay interrupciones? Si. ADIE, ADIF

12

Conversor A/D: Diagrama de bloques



13

Registros de configuración para el A/D:

14

Registros de configuración para el A/D:

REGISTER 21-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0
Legend:							
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'					
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown				
bit 7	ADFM: A/D Result Format Select bit						
	1 = Right justified						
	0 = Left justified						
bit 6	Unimplemented: Read as '0'						
bit 5-3	ACQT2:ACQT0: A/D Acquisition Time Select bits						
	111 = 20 TAD						
	110 = 16 TAD						
	101 = 12 TAD						
	100 = 8 TAD						
	011 = 6 TAD						
	010 = 4 TAD						
	001 = 2 TAD						
	000 = 0 TAD ⁽¹⁾						
bit 2-0	ADCS2:ADCS0: A/D Conversion Clock Select bits						
	111 = FRC (clock derived from A/D RC oscillator) ⁽¹⁾						
	110 = Fosc/64						
	101 = Fosc/16						
	100 = Fosc/4						
	011 = FRC (clock derived from A/D RC oscillator) ⁽¹⁾						
	010 = Fosc/32						
	001 = Fosc/8						
	000 = Fosc/2						

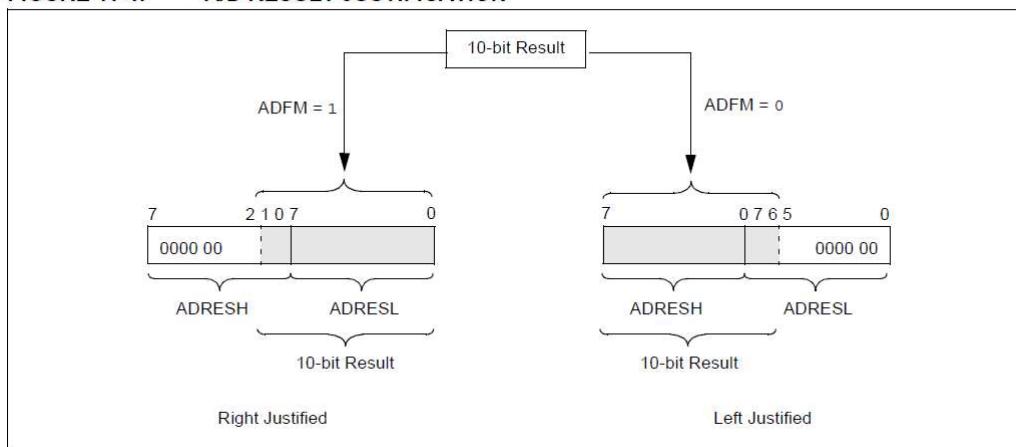
Procedimiento para adquirir una muestra de una señal analógica:

- Configure the A/D module:
 - Configure analog pins, voltage reference and digital I/O (ADCON1)
 - Select A/D input channel (ADCON0)
 - Select A/D acquisition time (ADCON2)
 - Select A/D conversion clock (ADCON2)
 - Turn on A/D module (ADCON0)
- Configure A/D interrupt (if desired):
 - Clear ADIF bit
 - Set ADIE bit
 - Set GIE bit
- Wait the required acquisition time (if required).
- Start conversion:
 - Set GO/DONE bit (ADCON0 register)
- Wait for A/D conversion to complete, by either:
 - Polling for the GO/DONE bit to be cleared
OR
 - Waiting for the A/D interrupt
- Read A/D Result registers (ADRESH:ADRESL); clear bit ADIF, if required.
- For next conversion, go to step 1 or step 2, as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 3 TAD is required before the next acquisition starts.

15

Para obtener el resultado de la conversión A/D:

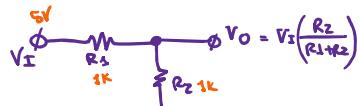
FIGURE 11-4: A/D RESULT JUSTIFICATION



16

Ejemplo, leer en AN0 el valor de voltaje de un potenciómetro configurado como divisor de tensión y el resultado en 8 bits emitirlo por el puerto D

Recordando:



$$V_O = \frac{V_I}{2} = 2.5V$$

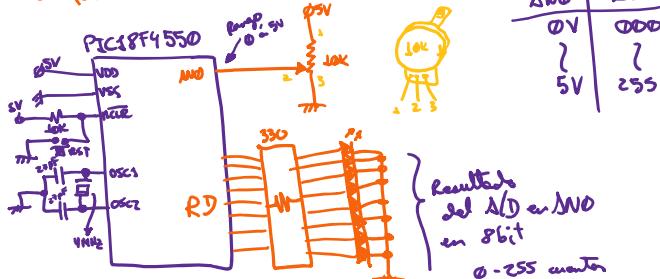


Diagrama de flujo:



17

Configuración para el A/D para AN.0:

REGISTER 21-1: ADCON0: A/D CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS0	CHS1	CHS2	CHS3	GO/DONE	ADON
bit 7							1
							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 7-6 Unimplemented: Read as '0'

bit 5-2 CHS3:CHS0: Analog Channel Select bits

- 0000 = Channel 0 (AN0)
- 0001 = Channel 1 (AN1)
- 0010 = Channel 2 (AN2)
- 0011 = Channel 3 (AN3)
- 0100 = Channel 4 (AN4)
- 0101 = Channel 5 (AN5)^(1,2)
- 0110 = Channel 6 (AN6)^(1,2)
- 0111 = Channel 7 (AN7)^(1,2)
- 1000 = Channel 8 (AN8)
- 1001 = Channel 9 (AN9)
- 1010 = Channel 10 (AN10)
- 1011 = Channel 11 (AN11)
- 1100 = Channel 12 (AN12)
- 1101 = Unimplemented⁽²⁾
- 1110 = Unimplemented⁽²⁾
- 1111 = Unimplemented⁽²⁾

bit 1 GO/DONE: A/D Conversion Status bit

When ADON = 1:
 1 = A/D conversion in progress
 0 = A/D Idle

bit 0 ADON: A/D On bit

1 = A/D converter module is enabled ✓
 0 = A/D converter module is disabled

REGISTER 21-2: ADCON1: A/D CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 7-6 Unimplemented: Read as '0'

bit 5 VCFG1: Voltage Reference Configuration bit (VREF- source)

1 = VREF- (AN2)

0 = VSS

bit 4 VCFG0: Voltage Reference Configuration bit (VREF+ source)

1 = VREF+ (AN3)

0 = VDD

bit 3-0 PCFG3:PCFG0: A/D Port Configuration Control bits:

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7 ⁽²⁾	AN6 ⁽²⁾	AN5 ⁽²⁾	AN4	AN3	AN2	AN1	AN0
0000 ⁽¹⁾	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111 ⁽¹⁾	D	D	D	D	A	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	A	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	A	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	A	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = Analog input

D = Digital I/O

18

Conf. del A/D para AN0: (tiempo de adq y just res)

REGISTER 21-3: ADCON2: A/D CONTROL REGISTER 2							
R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ACQT2	ACQT1	ACQT0	ACCS2	ACCS1	ACCS0	
bit 7							bit 0
Legend:							
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'					
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown				
bit 7	ADFM: A/D Result Format Select bit						
1	= Right justified						
0	= Left justified ✓						
bit 6	Unimplemented: Read as '0'						
bit 5-3	ACQT2:ACQT0: A/D Acquisition Time Select bits						
111	= 20 TAD						
110	= 16 TAD						
101	= 12 TAD						
100	= 8 TAD ✓						
011	= 6 TAD						
010	= 4 TAD						
001	= 2 TAD						
000	= 0 TAD ⁽¹⁾						
bit 2-0	ACCS2:ACCS0: A/D Conversion Clock Select bits						
111	= FRC (clock derived from A/D RC oscillator) ⁽¹⁾						
110	= Fosc/64						
101	= Fosc/16 ✓						
100	= Fosc/4						
011	= FRC (clock derived from A/D RC oscillator) ⁽¹⁾						
010	= Fosc/32						
001	= Fosc/8						
000	= Fosc/2						

ADCON2 = 0X24

19

Código en MPASM

```

1 ;Este es un comentario, se le antecede un punto y coma
2      list p=18f4550      ;Modelo del microcontrolador
3      #include <p18f4550.inc>    ;Llamada a la librería de nombre de
4
5      ;Directivas de preprocesador o bits de configuración
6      CONFIG PLLDIV = 1          ; PLL Prescaler Selection bits (N)
7      CONFIG CPUDIV = OSC1_PLL2 ; System Clock Postscaler Selection bits (M)
8      CONFIG FOSC = XT_XT      ; Oscillator Selection bits (XT or X)
9      CONFIG PWRT = ON         ; Power-up Timer Enable bit (PWRT)
10     CONFIG BOR = OFF        ; Brown-out Reset Enable bits (BOR)
11     CONFIG WDT = OFF        ; Watchdog Timer Enable bit (WDT)
12     CONFIG CCP2MX = ON       ; CCP2 MUX bit (CCP2 input/output)
13     CONFIG PBADEN = OFF     ; PORTB A/D Enable bit (PORTB<4:0>)
14     CONFIG MCLRE = ON       ; MCLR Pin Enable bit (MCLR pin enable)
15     CONFIG LVP = OFF        ; Single-Supply ICSP Enable bit (LVP)
16
17     org 0x0000
18     goto init_conf
19
20     org 0x0020
21 init_conf: clrf TRISD      ;RD como salidas
22             movlw 0x24
23             movwf ADCON2    ;8TAD, FOSC/4 ADFM=0
24             movlw 0x0E
25             movwf ADCON1    ;AN0 habilitado
26             movlw 0x01
27             movwf ADCON0    ;AN0 seleccionado y AD funcionando
28 loop:    bsf ADCON0, 1      ;Inicio la conversion en AN0
29 otro:   btfss ADCON0, 1    ;Pregunto si ya termino de convertir
30         goto otro
31         movff ADRESH, LATD
32         goto loop
33         end

```

20

Código en XC8 PIC Assembler

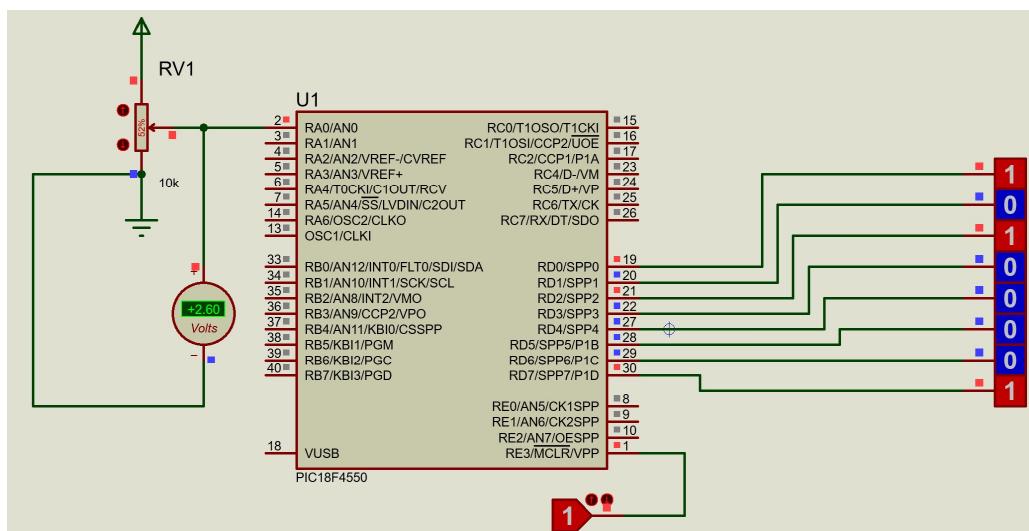
```

1      PROCESSOR 18F4550
2      #include "cabecera.inc"
3
4      PSECT principal, class=CODE, reloc=2, abs
5
6      principal:
7          ORG 0000H
8          goto configuro
9          ORG 0020H
10
11     configuro:
12         clrf TRISD
13         movlw 24H
14         movwf ADCON2
15         movlw 0EH
16         movwf ADCON1
17         movlw 01H
18         movwf ADCON0
19
20     loop:
21         bsf ADCON0, 1
22
23     otro:
24         btfsc ADCON0, 1
25         goto otro
26         movff ADRESH, LATD
27         goto loop
28
29     end principal

```

21

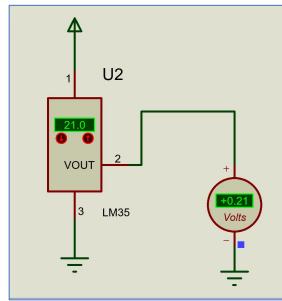
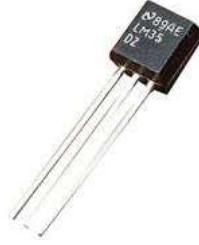
Simulación en Proteus



22

El sensor de temperatura LM35

- Sensor de temperatura de rampa lineal
- Pendiente de 10mv/°C
- Alimentación 5V
- Reacción lenta ante cambios bruscos de temperatura



1 Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full -55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates From 4 V to 30 V
- Less Than 60- μ A Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only $\pm\frac{1}{4}$ °C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load

2 Applications

- Power Supplies
- Battery Management
- HVAC
- Appliances

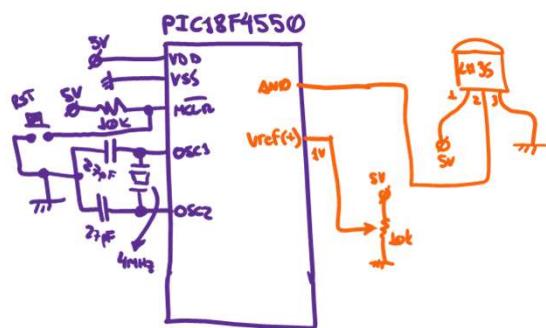
23

Consideraciones para usar el LM35 con el microcontrolador PIC18F4550

- Recortar la escala para una mayor facilidad en el tratamiento numérico. Ej 0°C a 100°C
- Si recortamos a ese rango (recordando 10mv/°C):
 - 0°C = 0V
 - 100°C = 1V
- El conversor A/D se deberá ajustar el Vref+ a un valor de 1V para que la señal adquirida obtenga todo el rango del A/D.

24

Circuito de interface del PIC18F4550 con el LM35

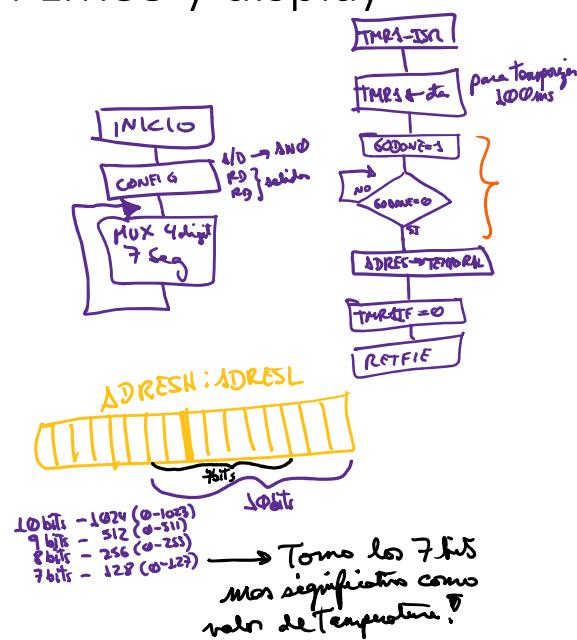
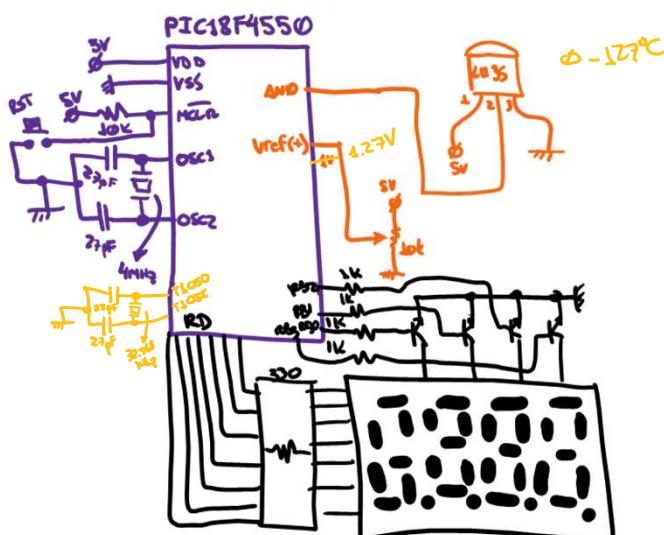


Escalamiento de medidas:

T	LM35	A/D (10bits)
0°C	0V	1023
100°C	1V	1023
127°C	1.27V	1023 (ADRESH: ADRESL)

25

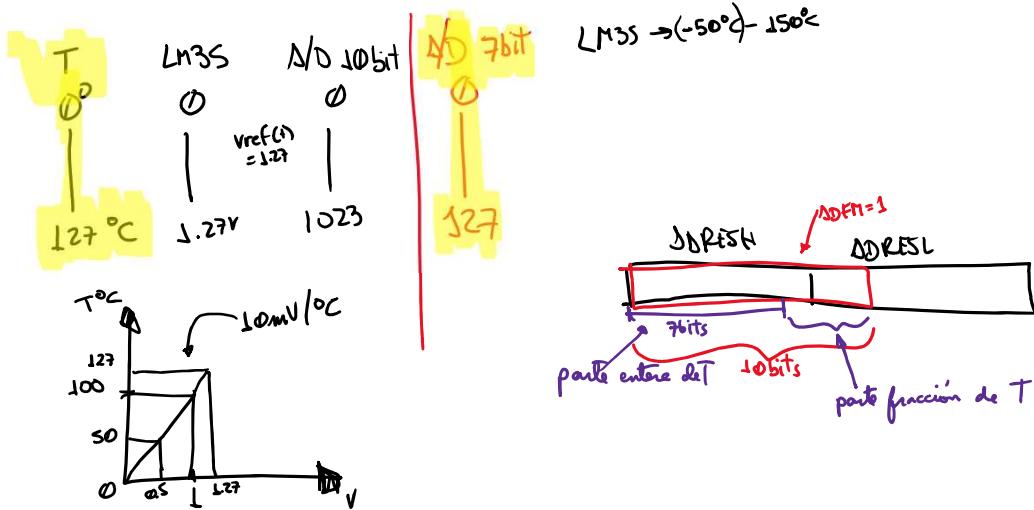
Ejemplo: Termómetro con LM35 y display multiplexado de 4 dígitos



26

¿Cómo hago para hacer el escalamiento de medidas sin hacer uso de operaciones de división y/o multiplicación?

- Ajustando el Vref para que tenga un valor de peso binario:



27

(cont...)

- El resultado del A/D en 7 bits nos da el valor de la temperatura sin necesidad de hacer cálculos adicionales si es que ajustamos el Vref+ a 1.27V
- Se requiere de un algoritmo (digbyte) que me permita obtener los dígitos centena, decena y unidad de manera independiente para que en el proceso de multiplexación los tome y se visualice.
 - Digbyte.inc deberá ser incluido en la carpeta Header Files del árbol del proyecto
- Para poder simular el ejemplo en Proteus:
 - Desactivar el T1OSCEN para que el PIC18F4550 reciba una señal cuadrada en T13CKI de 32.768KHz y tener así el RTC

```

1  digbyte MACRO arg0
2    LOCAL   Exit0
3    LOCAL   Exit1
4    LOCAL   Exit2
5
6    clrf    Dig0
7    clrf    Dig1
8    clrf    Dig2
9
10   movf   arg0, w
11   movwf  Digttemp
12   movlw  .100
13   Exit2
14   incf    Dig2, f
15   subwf  Digttemp, f
16   btfsc  STATUS, C
17   goto   Exit2
18   decf    Dig2, f
19   addwf  Digttemp, f
20   Exit1
21   movlw  .10
22   incf    Dig1, f
23   subwf  Digttemp, f
24   btfsc  STATUS, C
25   goto   Exit1
26   decf    Dig1, f
27   addwf  Digttemp, f
28   Exit0
29   movf   Digttemp, w
30   movwf  Dig0
31   ENDM

```

28

(cont...)

• Código en MPASM:

```

1 ;Este es un comentario, se le antecede un punto y coma
2 list p=18f4550           ;Modelo del microcontrolador
3 #include <p18f4550.inc>    ;Llamada a la librería de nombre de los
4 include "digibyte.inc"     ;Llamada a la librería de obtención de
5
6 ;Directivas de preprocesador o bits de configuración
7 CONFIG_PLLDIV = 1          ; PLL Prescaler Selection bits (No p
8 CONFIG_CFDIV = OSC1_PLL2   ; System Clock Postscaler Selection
9 CONFIG_FOSC = XT_XT        ; Oscillator Selection bits (XT oscil
10 CONFIG_FWRT = ON           ; Power-up Timer Enable bits (FWRT en
11 CONFIG_BOR = OFF           ; Brown-out Reset Enable bits (Brown-
12 CONFIG_WDT = OFF           ; Watchdog Timer Enable bit (WDT dis
13 CONFIG_CCP2MX = ON          ; CCP2 MUX bit (CCP2 input/output is
14 CONFIG_PORSEN = OFF         ; PORTB A/D Enable bit (PORTB<4:0> p
15 CONFIG_MCLR = ON            ; MCLR Pin Enable bit (MCLR pin enab
16 CONFIG_LVP = OFF            ; Single-Supply ICSP Enable bit (Sinc
17
18 cblock 0x0000
19 temporal                   ;almacenamiento temporal del resultado
20 Dig0
21 Dig1
22 Dig2
23 Digtemp
24 var_lms_x
25 var_lms_y
26 endc
27
28 ;Valores de la tabla de búsqueda para el siete segmentos
29 org 0x0400
30 tabla_7s db 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x67
31
32 org 0x0000                 ;vector de reset
33 goto init_conf
34
35 org 0x0008                 ;vector de interrupción
36 goto TMR1_ISR
37
38 org 0x0020                 ;zona de programa de usuario

```

```

40 ;Configuración inicial
41 init_conf: clrf TRISD
42           moviw 0xF0
43           movwf TRISE
44           moviw 0x24
45           movwf ADCON2
46           moviw 0x1E
47           movwf ADCON1
48           moviw 0x01
49           movwf ADCON0
50           moviw 0x04
51           movwf TBLPTRH
52           clrf TBLPTRL
53           moviw 0x0F
54           movwf TICON
55           bcf TICON, TIOSCEN
56           moviw 0x00
57           movwf THRIH
58           clrf THRL
59           bcf PIEL, TMRLIE
60           baf INTCON, PEIE
61           baf INTCON, GIE

```

```

63 ;Rutina principal
64 loop: digibyte temporal
65           movff Digi2, TBLPTRL
66           TBLRD*
67           movff TABLAT, LATD
68           baf LATB, 0
69           call delay_mux
70           bcf LATB, 0
71           movff Digi1, TBLPTRL
72           TBLRD*
73           movff TABLAT, LATD
74           baf LATB, 1
75           call delay_mux
76           bcf LATB, 1
77           movff Digi0, TBLPTRL
78           TBLRD*
79           movff TABLAT, LATD
80           baf LATB, 2
81           call delay_mux
82           bcf LATB, 2
83           moviw 0x39
84           movwf LATD
85           baf LATB, 3
86           call delay_mux
87           bcf LATB, 3
88           goto loop

```

```

89 ;Rutina pequeño retraso
90 delay_mux: moviw .80
91           movwf var_lms_x
92           call anidl
93           aun_no1: call anidl
94           decfsz var_lms_x, f
95           goto aun_no1
96           return
97           anidl: moviw .10
98           movwf var_lms_y
99           aun_no2: nop
100          decfsz var_lms_y, f
101          goto aun_no2
102          return

```

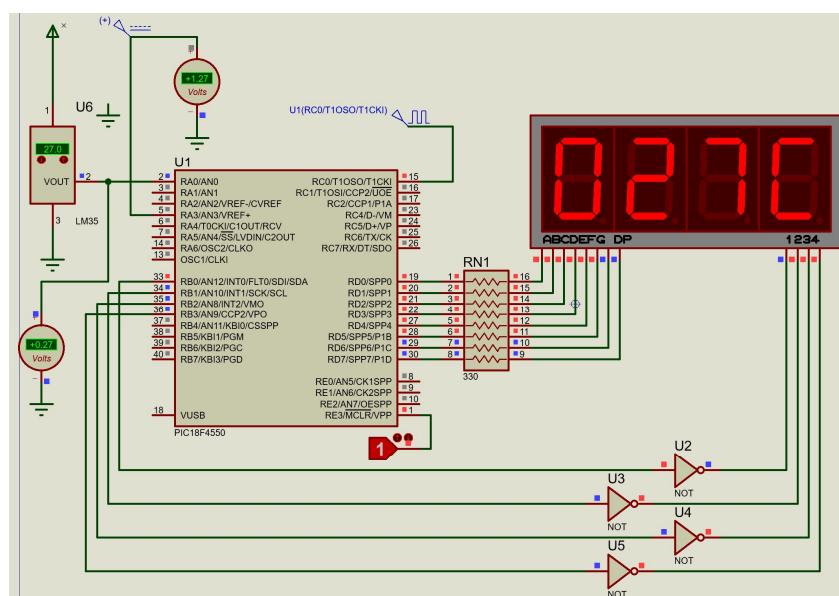
```

103 ;Rutina de interrupción
104 TMR1_ISR: moviw 0xC0
105           movwf THRIH
106           clrf THRL
107           bsf ADCONO, GO
108           otro9: btfss ADCONO, DONE
109           goto otro9
110           rrof ADRESH, W
111           andiw 0x7F
112           movwf temporal
113           bcf PIR1, TMRLIF
114           retfie
115
116
117 end

```

29

Circuito en Proteus



30

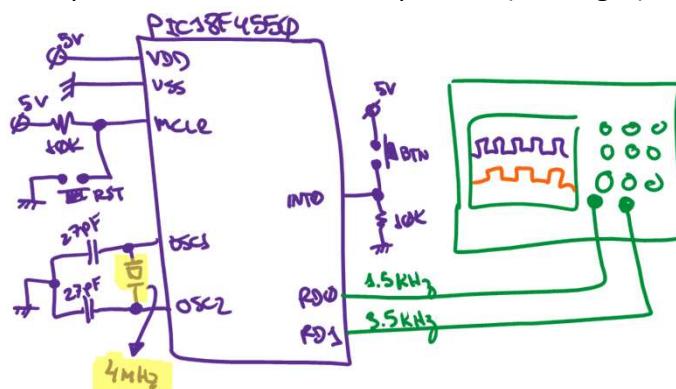
¿Multitarea?

- Ejecución de varias tareas a la vez en el microcontrolador
- Sistemas RTOS (Real Time Operating System), relacionado con lenguajes de alto nivel generalmente.
- En ensamblador la multitarea está relacionada con el uso de interrupciones.
- Ejecución de una instrucción ≠ ejecución de una tarea

31

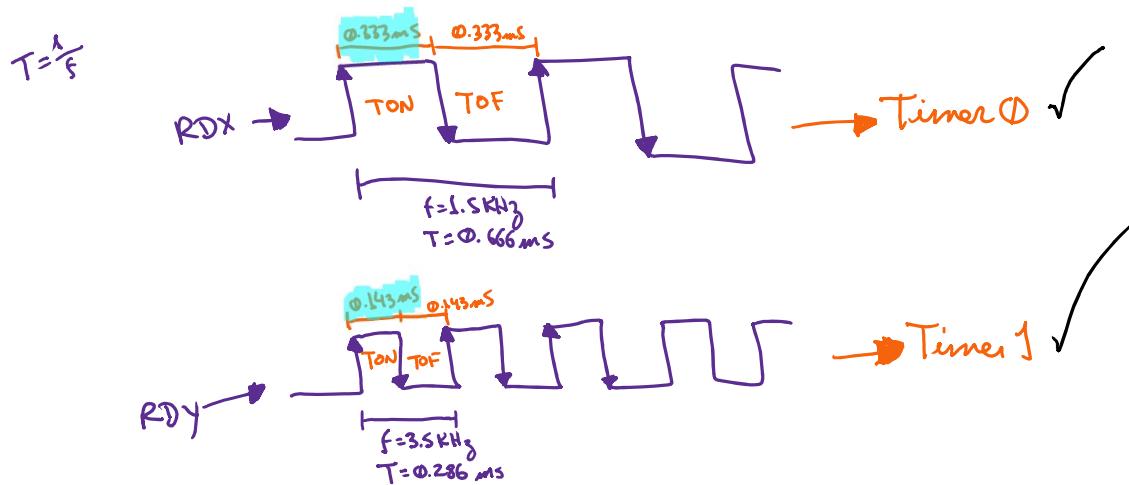
Ejemplo: Generación de dos ondas cuadradas de 1.5KHz y 3.5KHz (asíncronas)

- Se usarán dos temporizadores: Timer0 (señal 1.5KHz) y Timer1 (señal 3.5KHz)
- Al presionar BTN (INT0) se intercambiarán (swap) la salida de las señales.
- Se activarán interrupciones por desborde de ambos temporizadores y la INT0
- Se emplearán prioridades en las interrupciones (a escoger)



32

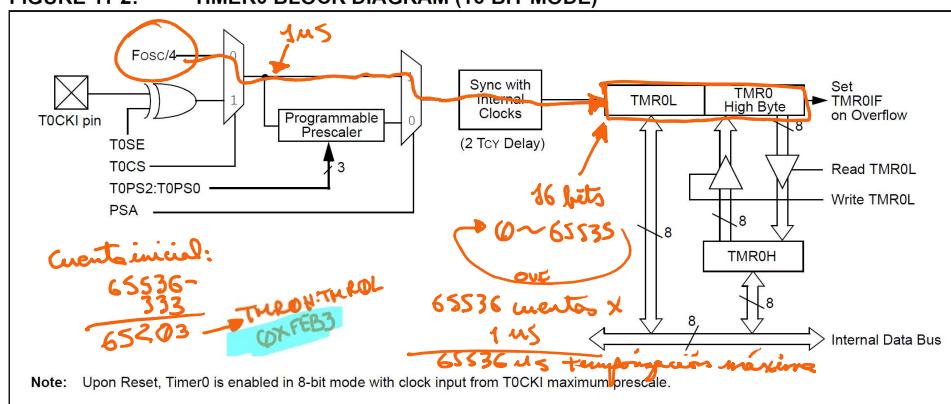
Análisis de las señales a reproducir (50% DC):



33

Configuración del Timer0 para temporizar 0.333ms

FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)



34

Configuración del Timer0 para temporizar 0.333ms

REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMROON	0BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7	bit 0	bit 0	X	bit 1	X	X	bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 7	TMROON: Timer0 On/Off Control bit 1 = Enables Timer0 0 = Stops Timer0
bit 6	T0BIT: Timer0 8-Bit/16-Bit Control bit 1 = Timer0 is configured as an 8-bit timer/counter 0 = Timer0 is configured as a 16-bit timer/counter
bit 5	T0CS: Timer0 Clock Source Select bit 1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (Fosc/4) → Fosc/4
bit 4	T0SE: Timer0 Source Edge Select bit 1 = Increment on high-to-low transition on T0CKI pin 0 = Increment on low-to-high transition on T0CKI pin
bit 3	PSA: Timer0 Prescaler Assignment bit 1 = Timer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler. 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.
bit 2-0	T0PS2:T0PS0: Timer0 Prescaler Select bits 111 = 1:256 Prescale value 110 = 1:128 Prescale value 101 = 1:64 Prescale value 100 = 1:32 Prescale value 011 = 1:16 Prescale value 010 = 1:8 Prescale value 001 = 1:4 Prescale value 000 = 1:2 Prescale value

$$T0CON = \emptyset X X 88$$

35

Configuración del Timer1 para temporizar 143us

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMROON
bit 7	bit 0	bit 0	bit 0	X	X	bit 0	bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 7	RD16: 16-Bit Read/Write Mode Enable bit 1 = Enables register read/write of Timer1 in one 16-bit operation 0 = Enables register read/write of Timer1 in two 8-bit operations
bit 6	T1RUN: Timer1 System Clock Status bit 1 = Device clock is derived from Timer1 oscillator 0 = Device clock is derived from another source
bit 5-4	T1CKPS1:T1CKPS0: Timer1 Input Clock Prescale Select bits 11 = 1:8 Prescale value 10 = 1:4 Prescale value 01 = 1:2 Prescale value 00 = 1:1 Prescale value
bit 3	T1OSCEN: Timer1 Oscillator Enable bit 1 = Timer1 oscillator is enabled 0 = Timer1 oscillator is shut off The oscillator inverter and feedback resistor are turned off to eliminate power drain.
bit 2	T1SYNC: Timer1 External Clock Input Synchronization Select bit When TMR1CS = 1: 1 = Do not synchronize external clock input 0 = Synchronize external clock input When TMR1CS = 0: This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.
bit 1	TMR1CS: Timer1 Clock Source Select bit 1 = External clock from RC0/T1OSO/T13CKI pin (on the rising edge) 0 = Internal clock (Fosc/4)
bit 0	TMROON: Timer1 On bit 1 = Enables Timer1 0 = Stops Timer1

$$T1CON = \emptyset X \emptyset 1$$

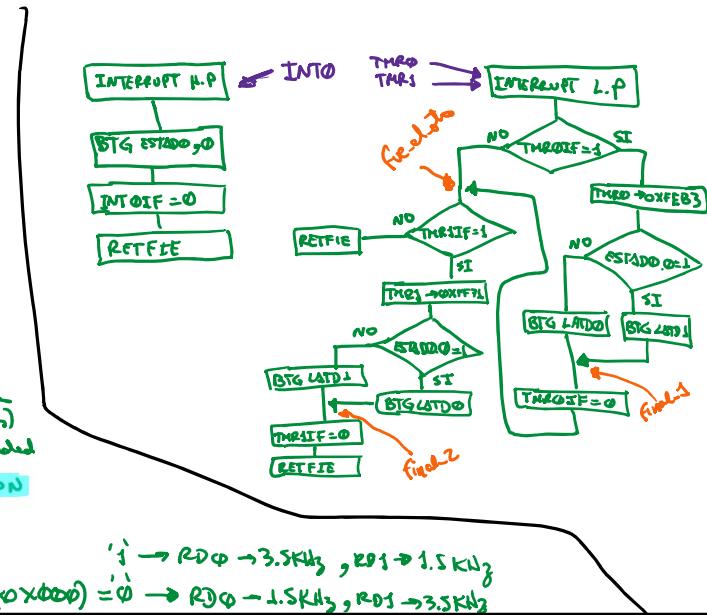
36

Desarrollo del algoritmo en diagrama de flujo

5º Hay tres fuentes de interrupción:

- INT0 → alta prioridad
 - Timer0 → baja prioridad
 - Timer1 → baja prioridad

2º Diagrama de flujo:



37

Código en MPASM:

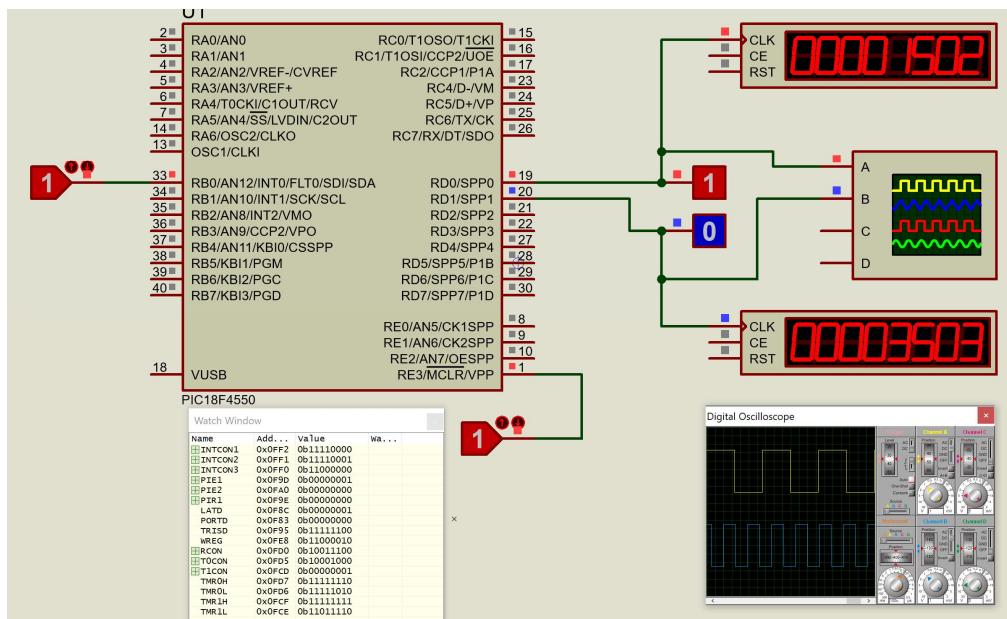
```

60 TMR0_TMR1_ISR:
61     btfss INTCON, TMROIF      ;Pregunto si ocurrio TMRO
62     goto fue_el_otro
63     movlw 0xFF
64     movwf TMR0H
65     movlw 0xC2
66     movwf TMR0L
67     btfss ESTADO, 0           ;Cargo cuenta inicial en TMRO
68     goto estado_cero_a
69     btg LATD, 1               ;Basculo RD1
70     goto final_1
71
72     estado_cero_a:
73         btg LATD, 0           ;Basculo RD0
74
75     final_1:
76         bcf INTCON, TMROIF  ;Bajamos bandera de TMRO
77
78     fue_el_otro:
79         btfss PIR1, TMRLIF   ;Pregunto si ocurrio TMRL
80         retfie
81         movlw 0xFF
82         movwf TMR1H
83         movlw 0x80
84         movwf TMR1L
85         btfss ESTADO, 0           ;Cargo cuenta inicial en TMRL
86         goto estado_cero_b
87         btg LATD, 0               ;Basculo RD0
88         goto final_2
89
90     estado_cero_b:
91         btg LATD, 1           ;Basculo RD1
92
93     final_2:
94         bcf PIR1, TMRLIF   ;Bajo bandera de TMRL
95         retfie
96         end

```

38

Simulación:

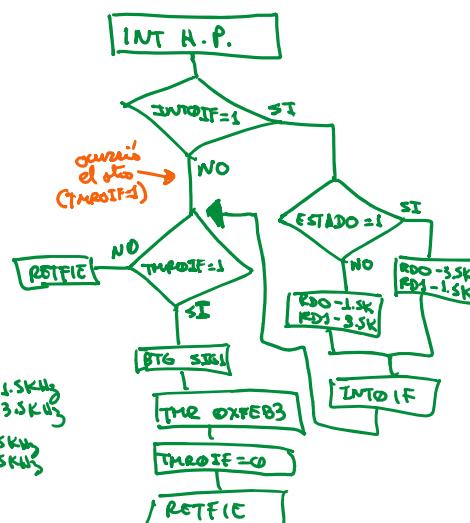
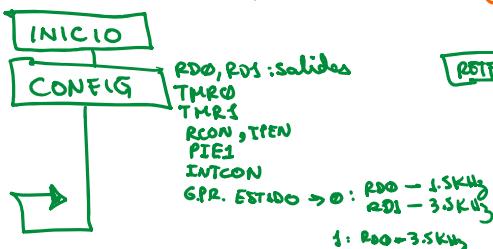


39

Modificación al ejemplo: Cambiando las prioridades de las fuentes de interrupción

Fuentes de interrupción:

- INT0 \leftarrow alta prioridad
 - Timer0 \leftarrow alta prioridad
 - Timer1 \leftarrow baja prioridad



40

Código en MPASM: Configuración y rutina principal

```

1      ;Este es un comentario, se le antecede un
2      list p=18f4550      ;Modelo del micro
3      #include <p18f4550.inc>      ;Llamada
4
5      ;Directivas de preprocesador o bits de
6      CONFIG PLLEDIV = 1          ; PLL P
7      CONFIG CPUDIV = OSC1_PLL2    ; System
8      CONFIG FOSC = XT_XT        ; Oscill
9      CONFIG EWRT = ON           ; Power
10     CONFIG BOR = OFF           ; Brown
11     CONFIG WDT = OFF           ; Watch
12     CONFIG CCP2MX = ON          ; CCP2
13     CONFIG PBADEN = OFF         ; PORTB
14     CONFIG MCLRE = ON           ; MCLR
15     CONFIG LVP = OFF            ; Singl
16
17     ;Aqui va el cblock o declaración de nombre
18     cblock 0x0000
19     ESTADO
20     endc
21
22     org 0x0000                ;Vector de RE
23     goto init_conf
24
25     org 0x0008                ;Vector de in
26     goto INTO_TMRO_ISR
27
28     org 0x0018                ;Vector de in
29     goto TMRI_ISR
30
31     org 0x0020                ;Zona de
32     init_conf:
33     bcf TRISD, 0
34     bcf TRISD, 1
35     movlw 0x88
36     movwf TOCON               ;Configu
37     movlw 0x01
38     movwf TICON                ;Configu
39     bsf RCON, IPEN             ;Activo
40     bcf IPR1, TMRLIP           ;Mandar
41     bsf PIE1, TMRLIE           ;Activo
42     movlw 0xF0
43     movwf INTCON               ;Activo
44     clrf ESTADO                ;Forzam
45     loop:
46     nop
47     nop
48     goto loop

```

41

Código en MPASM: Rutinas de interrupción

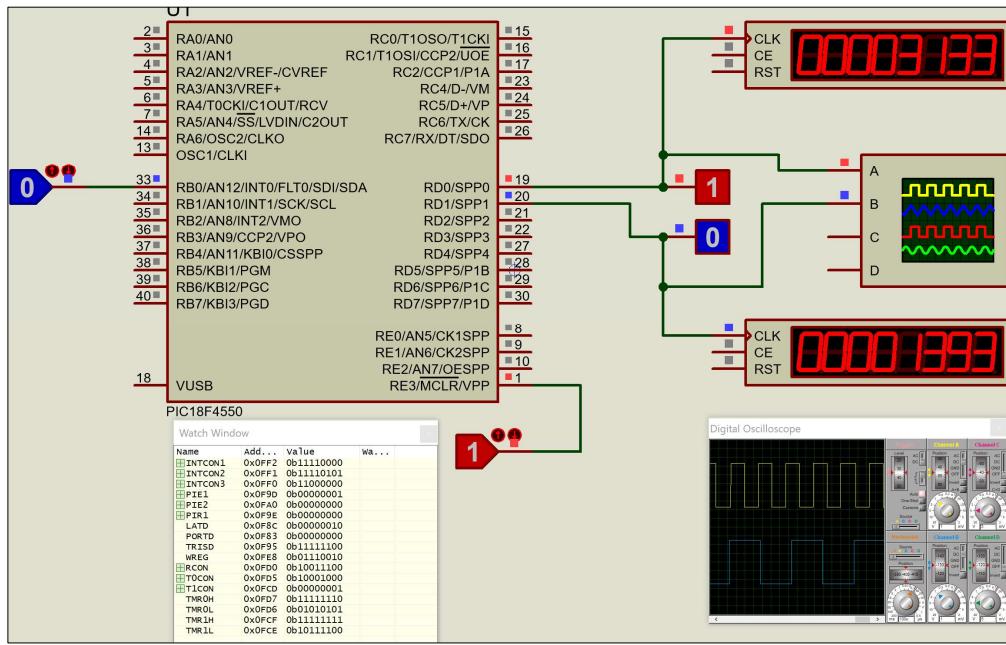
```

50    INTO_TMRO_ISR:          ;Rutina de interrupcion high priority
51    ;   bcf INTCON, GIEH      ;Apagamos temporalmente el interrupcion GIEH para
52    btfss INTCON, INTOIF    ;Pregunto si ocurrio INTO
53    goto el_otro
54    btg ESTADO, 0           ;Basculo el estado
55    bcf INTCON, INTOIF    ;Bajo la bandera de INTO
56    el_otro:
57    btfss INTCON, TMROIF    ;Pregunto si se desbordo TMRO
58    goto nada_masi
59    btfss ESTADO, 0          ;Pregunto a que puerto debe de salir la señal SIG1
60    goto para_RD0
61    btg LATD, 1
62    goto nada_masi
63    para_RD0:
64    btg LATD, 0
65    nada_masi:
66    movlw 0xFF
67    movwf TMROH
68    movlw 0xB3
69    movwf TMROL
70    bcf INTCON, TMROIF    ;Cargamos cuenta inicial en TMRO
71    ;   bsf INTCON, GIEH      ;Bajamos la bandera de TMRO
72    retfie                 ;Encendemos nuevamente GIEH
73
74    TMRI_ISR:              ;Rutina de interrupcion low priority
75    btfss ESTADO, 0
76    goto para_RD1
77    btg LATD, 0
78    goto nada_mas2
79    para_RD1:
80    btg LATD, 1
81    nada_mas2:
82    movlw 0xFF
83    movwf TMRIH
84    movlw 0x72
85    movwf TMRI
86    bcf IPR1, TMRLIF       ;Cargamos cuenta inicial en TMRI
87    retfie                 ;Bajamos la bandera de TMRI
88    end

```

42

Simulación en Proteus



43

Cuestionario:

- Modificar el ejemplo anterior para que se pueda modificar el duty cycle de ambas ondas de manera independiente empleando un teclado matricial para ello.

44

Fin de la sesión