U.T.N.

F.R.S.N.

Técnicas Digitales II

Proyecto

Servir-T

Integrante: Bernárdez, Andrés

Profesor: Ing Daniel De Pauli
Ayudante: Roberto Rodríguez

Abril 2011

INDICE

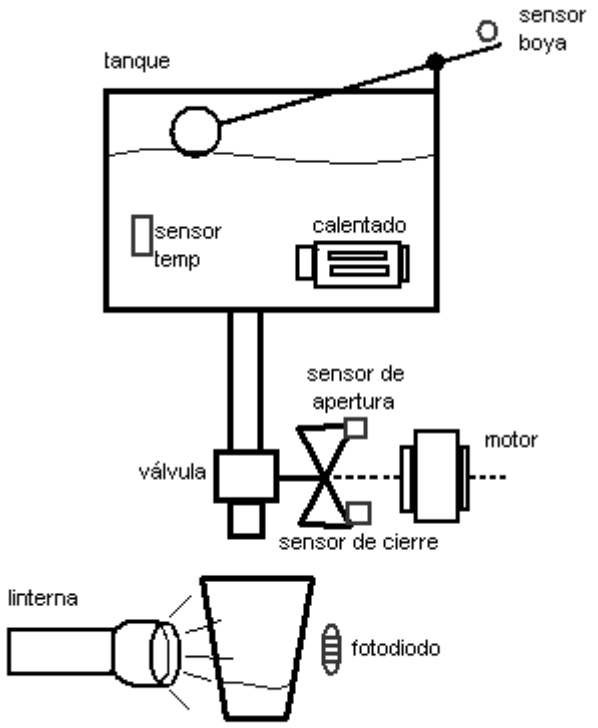
PROYECTO SERVIR-T

Servir-T

El proyecto Servir-T es realmente una máquina capaz de calentar té (o café) previamente preparado y servirlo en una tasa o vaso transparente automáticamente. Este proyecto bien podría representar un proceso químico en pequeña escala, donde se manejan niveles de líquidos y temperatura.
La máquina es además capaz de ser programada con una alarma para que realice esta operación a una hora determinada y la temperatura final puede elegirse hasta 99º C (aunque no es recomendable superar los 70ºC)

La máquina

La máquina posee un recipiente principal elevado, tanque, que está conectado directamente a una electroválvula. Debajo de esta última debe colocarse un recipiente transparente, vaso. A los costados del vaso hay una linterna y un fotodiodo. Estos están posicionados de tal manera que la luz llega al fotodiodo atravesando al vaso vacío. Además dentro del tanque estan el sensor de temperatura , el calentador y una boya para sensar nivel mínimo.



La electrovávula es de fabricación casera y consiste en un motor 220V CA y dos finales de carrera con un sensor en cada uno. El motor solo girará hacia la dirección que tenga

libre. La electroválvula solo tiene 2 estados: abierta y cerrada. A travez de los 2 sensores se sabe en que estado está.

Los sensores y la linterna son alimentados con una fuente de 5 V CC
El calentador y el motor de la electroválvula son alimentados con 220 V CA
El sensor de temperatura es el LM35 capaz de medir desde -55 hata 150ºC. La salida es lineal y equivale a 10mV/ºC. Como el sensor no está preparado para estar sumergido se lo ha recubierto de FASTIX para aislarlo de la humedad. Además el FASTIX es un aislante electrico moderado, es flexible y soporta de -18 a 180 º C. Experimentalmente se ha comprobado que es capaz de realizarse la medición bajo el agua. Es sumamente necesario que el sensor este aislado eléctricamente ya que no solo no daría una medida correcta sino que fugas de corrientes del calentador podrían destruir al sensor y al PIC.

El proceso.

Consiste en elevar la temperatura del líquido contenido en el tanque a travez del calentador hasta que alcance un límite elegido. Luego apagar el calentador, prender la linterna y abrir la electrovalvula. Esta permanecerá abierta llenando el vaso. El líquido debe ser oscuro. Cuando el líquido llegue a la posición del fotodiodo, le tapará la luz de la linterna, entonces la electrovalvula cerrará y finalizará el proceso.

Existen algunas prevenciones: El calentador no encenderá si no hay suficiente líquido en el tanque sensado por la boya. Tampoco calienta si la válvula esta abierta. El calentador se apaga después de un tiempo (45 seg.) si no se percibe variación de temperatura o luego que se supere los 9 minutos de calentamiento aunque no se haya alcanzado la temperatura final. Si cuando la electrovalvula esta abierta se supera los 45 segundos sin que el fotodiodo sense, esta se cierra automáticamente.
Todas estas prevenciones son controladas por el micro.

El microcontrolador

El micro utilizado es el pic 16f873A ya que posee conversor analógico digital, tiene varios puertos y su precio es moderado.
El oscilador es generado por un cristal de cuarzo de 4 mhz.
Se han utilizado prácticamente los 3 puertos, solo han quedado libre 2 pines del puerto A.
El puerto A es usado para medir la temperatura, prender la linterna, comandar el relé del motor de electroválvula y comandar el relé del calentador
El puerto B es usado para controlar 4 display
El puerto C es usado para 4 botones y 4 sensores

La alimentación del pic será de 5V CC proporcionados por una fuente ordinaria de computadora, la misma que alimentará a los sensores y botones.

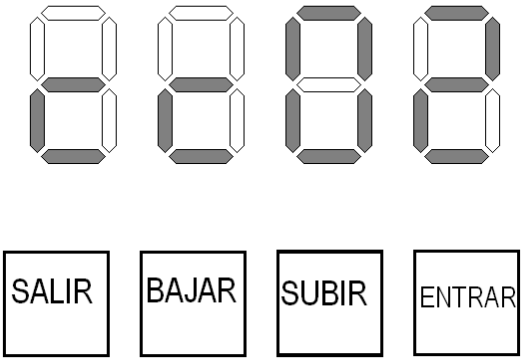
Los relés son alimentados con 12V CC proporcionados tambien por la misma fuente

La interfaz con el usuario

Para la interfaz con el usuario se usa cuatro displays de 7 segmentos y 4 botones .

Cada botón tiene una función específica.
De izquierda a derecha: "Salir", "Bajar" , "Subir" y "Entrar". Aunque esta funciones puede variar en alguna instancia del programa

Los displays son de ánodo común y estan conectados a un bcd 7447 para poder ahorrar pines y al mismo tiempo facilitar la programación. En consecuencia solo se pueden mostrar dígitos del 0 al 9 y 5 "caracteres" especiales utilizados.



Manual de usuario

Cuando se prende el equipo se puede ver en la pantalla el menú principal que se caracteriza por tener 2 "c" en los displays de la izquierda

Con los botones "subir" y "bajar" se cambia el número. Si se aprieta "entrar" se entra en la "función de usuario" seleccionada
Existen 5 funciones de usuario
1) Hora
2) Alarma
3) Ver temperatura (tensión)
4) Proceso
5) Abrir/Cerrar

Hora

A continuación se ve la pantalla mostrando las 12hs con 35 minutos

Al entrar en la opción 1 del menú se muestra la hora.
Si se aprieta el botón "Salir" se vuelve al menú principal.

Si se aprieta el borón “Subir” se ven los minutos con los segundos
Aquí se ve la pantalla mostrando 35minutos con 48 segundos

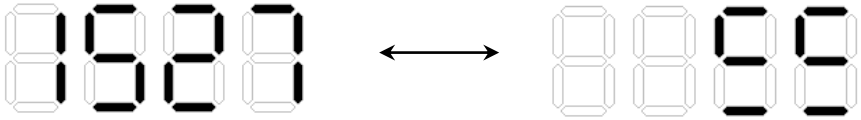
Si se aprieta el botón “Bajar”, se vuelve a mostrar la hora con los minutos
Si se aprieta el botón “Entrar” empieza a parpadear la hora
Si se vuelve a apretar “Entrar” empieza a parpadear los minutos
Cuando la hora o los minutos parpadean se pueden cambiar con los botones “Subir” o “Bajar”. Al apretar “Salir” se vuelve a la opción 1.

Alarma

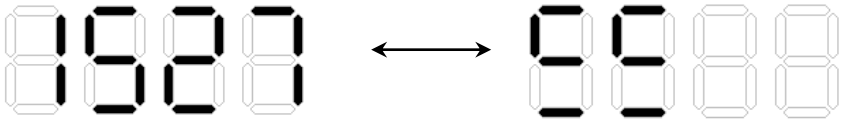
Cuando se entra en la opción 2 del menú principal empieza a parpadear la pantalla mostrando la hora y minutos para alarma.
Se sabe que está en la opción 2 del menú ya que se podrá visualizar el siguiente símbolo:

 : Símbolo que significa que se está modificando valores para alarma.

Al entrar desde el menú se verán en la pantalla la hora y los minutos para la alarma que parpadean continuamente de la siguiente manera:
Ejemplo: *hora de alarma= 15:27 entonces*



Esta es la primera pantalla de alarma.
En este caso se puede cambiar la hora de alarma con los botones “Subir” y “Bajar”
si se aprieta “Entrar” se pasa a cambiar los minutos. Para nuestro ejemplo sería:



Apretando “subir” y “bajar” se cambia los minutos de alarma. Apretando “Entrar” se pasa a modificar la temperatura de alarma final:
Supongamos 46º centigrados para temperatura de alarma:

En este caso parpadea la decena y se puede modificar de 0 a 9 con los botones “subir” y “bajar”. Al apretar “Entrar” se pasa a la siguiente pantalla:

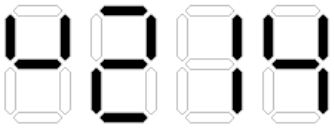
Ahora parpadea la unidad y se modifica con los botones de “subir” y “bajar”. Si apretamos “Entrar” vamos a la última pantalla de alarma que es “Activar Alarma”:

Si se aprieta "Bajar" se muestra el cero y se desactiva la alarma. Si se aprieta "Subir" se muestra el uno y se activa la alarma. Con "entrar" se vuelve a la primera pantalla de alarma y con "salir" se sale al menú principal.
En todos los otros casos el botón "salir" lleva a la primera pantalla de alarma y en caso de estar en la primera pantalla, "salir" lleva al menú principal.

>>Al estar activada la alarma, cuando la hora del dispositivo sea igual a la de alarma se activará el proceso de calentar el té para servirlo.

Ver temperatura

La opción 3 del menú lleva a la siguiente pantalla que muestra los grados centígrados que se está sensando en el tanque.
Supongamos que se miden 21,4º entonces se vería:



Si se aprieta "salir", "bajar" o "subir" se vuelve al menú principal. ("Entrar" sirve para medir tensiones superiores a 1 Volt y solo tiene aplicación para el programador.)

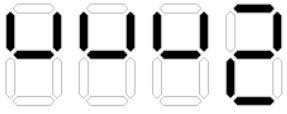
Proceso

Al elegir la opción 4 del menú se ve la temperatura límite a la que llegará el té cuando se active el proceso.
Supongamos 70ºC:

Cuando parpadea la decena se puede cambiar con "subir" y "bajar". Si se aprieta "Entrar" se hace parpadear la unidad que se cambia de la misma manera. Si se aprieta "salir" se vuelve al menú, si en cambio se vuelve a apretar "Entrar" se activa el proceso.

Entonces se verá la temperatura actual, supongamos que fuera 24,4º

Si se aprieta algún botón se vuelve a la pantalla principal de la opción 4 desactivando el proceso.
Mientras se esté en el proceso se calentará el té hasta que supere la temperatura seleccionada, en este caso 70,0º
Cuando llegue se abrirá la válvula y se mostrará la siguiente pantalla



El número indica que proceso se está realizando. Al finalizar el proceso se vuelve al menú principal.

Nota: antes de activar el proceso se debe llenar el tanque con té, de lo contrario el proceso no se activará y saldrá un error.

Abrir/Cerrar

Al entrar en la opción 5 del menú se observa la siguiente pantalla:

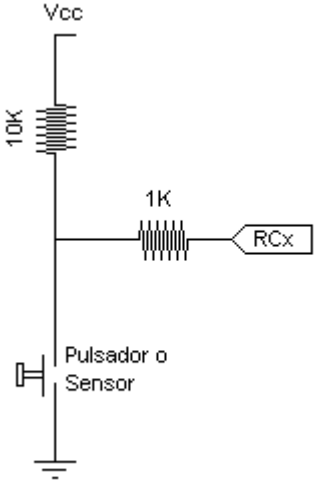
Si se aprieta el botón “entrar” se abre la válvula y el número cambia a 2.
Si se aprieta el botón “bajar” o “subir” se cierra la válvula y hasta que se cierre se muestra el 0
Si se aprieta “salir” se vuelve al menú principal
El número indica el número de proceso donde:
”0” es cerrar válvula
”1” no hacer nada
“2” abrir válvula
”3” calentar
”4” prender linterna para abrir

Pantalla de errores:
En caso de error el proceso se para y se muestra la siguiente pantalla:

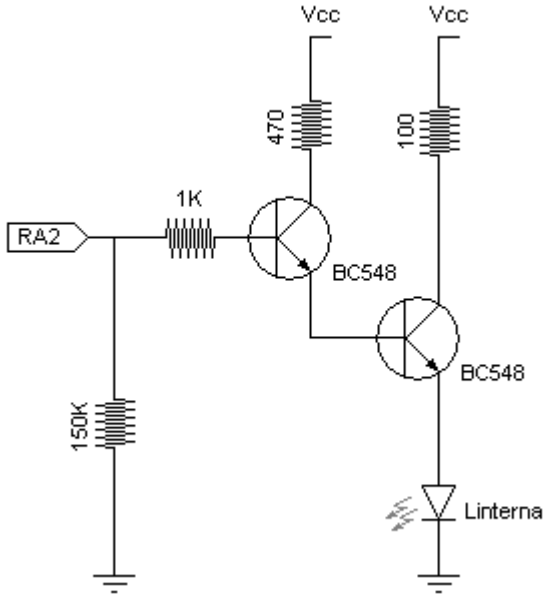
El número indica el número de error. Cualquier botón lleva al menú principal.
Los errores son explicados mas adelante (en “Errores de proceso” pag:25)

Circuitos de la máquina

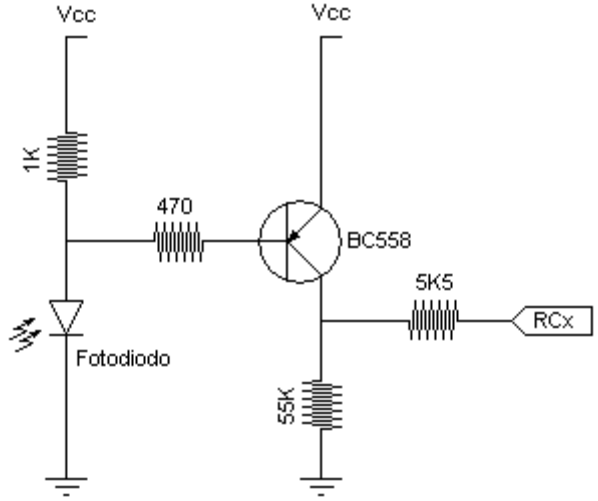
Circuito para pulsador o sensor



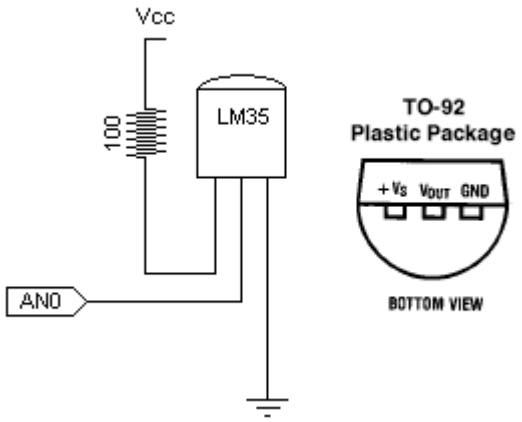
Circuito para la linterna



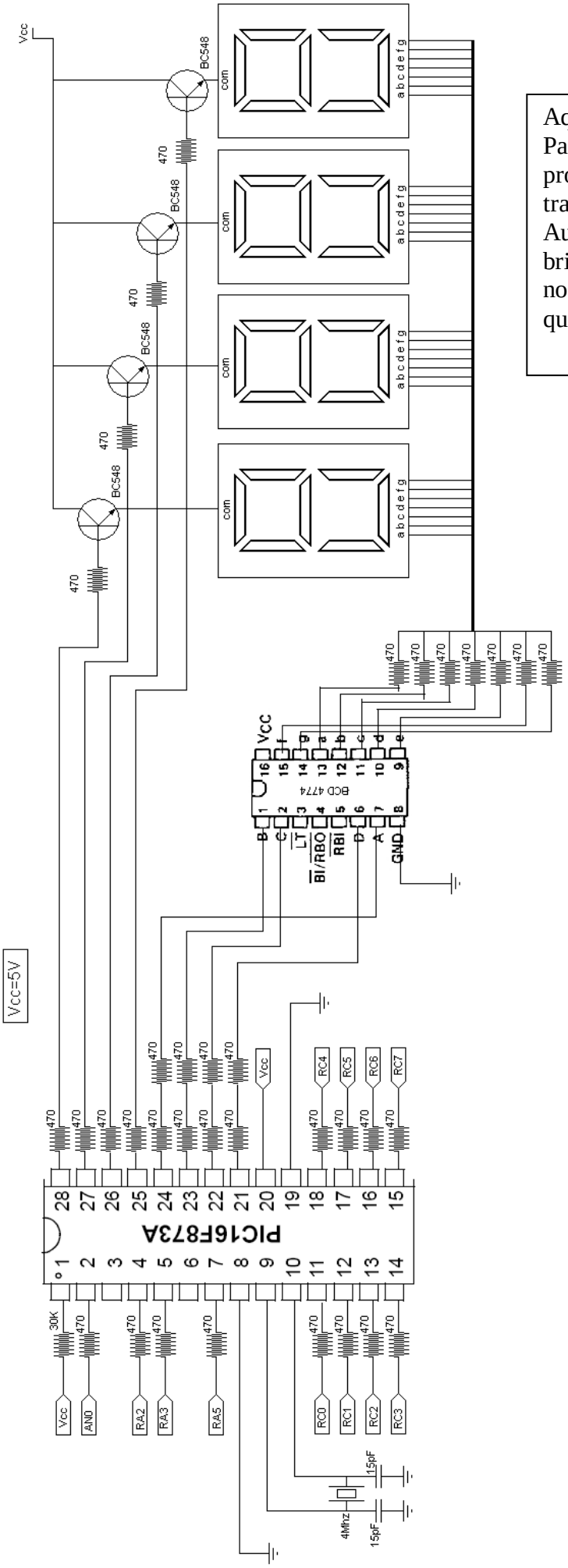
Circuito para el fotodiodo



Circuito del sensor de temperatura



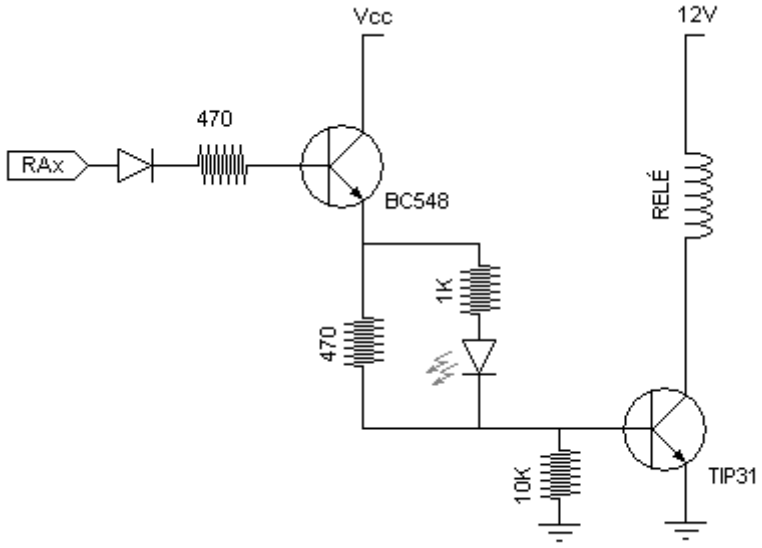
Circuito del pic16F873A y de los displays 7 segmentos



Aquí hay un error. Para no cambiar el programa se usó transistores npn. Aunque funciona, el brillo de los displays no es el mejor. Hay que usar pnp.

Circuito para relé

Este es el circuito que usa el relé para el motor controlado por el pin RA5 y el relé para el calentador controlado por el pin RA3



Estructura del programa

Interrupciones:

El tiempo y el refresco de pantalla se maneja constantemente con las interrupciones sin importar en que instancia del programa principal se encuentre.

Para refrescar los displays se toma la información de 4 punteros que indican que debe ser mostrado. Estos punteros deben haber sido cargados previamente.

Programa Principal:

El programa principal es un bucle que consiste en 5 etapas

1º revisar que parte del "proceso" a hacer
2º revisar si hay que realizar una conversión
3º revisar si se activó la alarma
4º revisar si los sensores o botones han cambiado su estado
5º ejecutar la "función actual"

1 PROCESO ACTUAL

A travez de una variable llamada "PROCESO" se elige una de las 5 etapas de proceso programadas

- 0:Cerrar válvula
- 1:No realizar nada
- 2:Abrir válvula
- 3:Calentar
- 4:Prender linterna

Si surge algún error se carga a "ERRORNUM" con el número de error correspondiente y se activa la función 30 (FUNCION=30)

2 CONVERSIÓN

La conversión se realiza a travez del bit0 del puertoA se toma como referencia directamente a V+=Vcc=5V y V-=0V . Aun cuando el LM35 solo tiene como máximo en su salida 1,5V. En consecuencia se ha sacrificado apreciación aunque se compensa por estar utilizando los 10 bits del conversor AD.
Experimentalmente se ha estimado que la medición tiene una exactitud con +-1,5 grados de error. Además debido a la capa de aislante que se ha agregado, se debe esperar un tiempo para que el sensor esté a la temperatura del líquido.

3 ALARMA:

Cuando una variable llamada "ALARMABIT" se pone a 1 se compara la hora con la hora de alarma y cuando estas son iguales se cargan parámetros, se borra a ALARMABIT y se llama "indirectamente" a la función 26 a traves de la variable "FUNCION" (FUNCION=26) para que se realice un ciclo de proceso

4 VERIFICAR BOTONES y SENSORES

Un código anti-rebote que consiste en retardos y comparaciones, verifica si algún botón o sensor ha cambiado de estado (Los botones y sensores estan conectados al Puerto C) Luego se cargan 2 variables: "SENSO" y "PETICION"

5 FUNCION ACTUAL

Para este programa se ha definido "función para usuario" o "función de usuario", como un estructurado grupo de sentencias.

Toda función de usuario consiste en:

- Actualizar los punteros para mostrar en los displays (cuando estos se refresquen)
- Atender y asignar una orden a cada botón (a través de variable "PETICION")
- Cambiar el valor de alguna/s variable/s del programa

Además indirectamente, a través de variables, las funciones pueden activar la alarma, pedir que se realice un proceso, pedir que se realice conversiones A.D., etc.

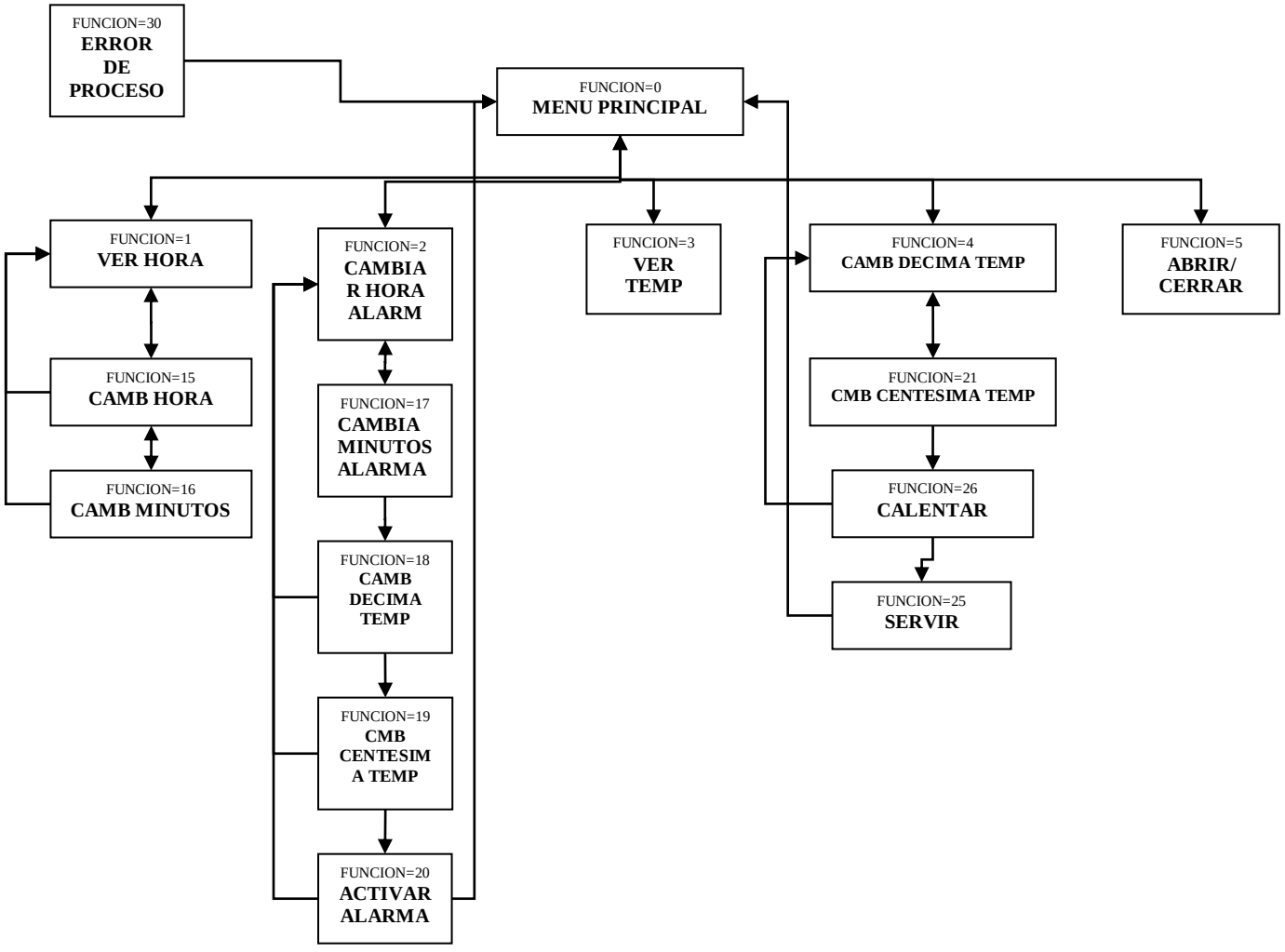
Cada función tiene un número único que lo identifica. El programa principal sabe en que función de usuario se encuentra a través del valor numérico de la variable "FUNCION"

La función 0 es el menú principal.
La función 1 sirve para ver la hora
La función 2 sirve para cambiar la hora de alarma
La función 3 sirve para ver la temperatura actual del tanque
La función 4 sirve para ajustar temperatura final
La función 5 sirve para abrir o cerrar la electorválvula
ETC…

- Estando en una función se puede acceder a otra mediante los botones.

A continuación se muestra el mapa de las funciones que el usuario puede percibir utilizando la interfaz

Mapa de las funciones de usuario



Programación:

Pic16f873A

Se utiliza la página 1 de la memoria flash, las variables auxiliar usadas están en el banco 0 de la memoria ram

Se ha aprovechado que este micro tiene mas de un timer para producir 2 interrupciones: uno para marcar el tiempo con buena precisión y otro para refrescar los displays automáticamente.

Se procuró llamar pocas funciones(usando “call”) a la vez ya que se cuenta con solo 8 niveles de stack

El tiempo:

El micro esta constantemente marcando el tiempo usando una interrupción. Esto no solo se usa para indicar la hora sino que además se aprovecha para generar pulsos y para contar tiempo mediante una variable que se incrementa cada ¼ de segundo. Así que una sola interrupción sirve como referencia para toda aplicación que trabaje con el tiempo. Para generar una buena BASE DE TIEMPO usé el módulo CCP (Capture/Compare/PWM)

Los registros en total asociados con el Capture, Compare y el timer1 son:

TABLE 8-4: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE AND TIMER1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh,8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- ---0	---- ---0
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- ---0	---- ---0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by Capture and Timer1.
Note 1: The PSP is not implemented on 28-pin devices; always maintain these bits clear.

Uso el "Compare Mode" en modo "special event trigger". Para el Timer1 sirve para crear un efectivo periodo de tiempo programable de 16bits. El timer se resetea solo cada vez que el valor del Timer1 es igual al del CCP y activa la interrupción poniendo en uno el bit CCP1F (para que la interrupción tenga efecto GIE, PEIE, CCP1IE tienen que estar en 1)

8.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated which may be used to initiate an action.

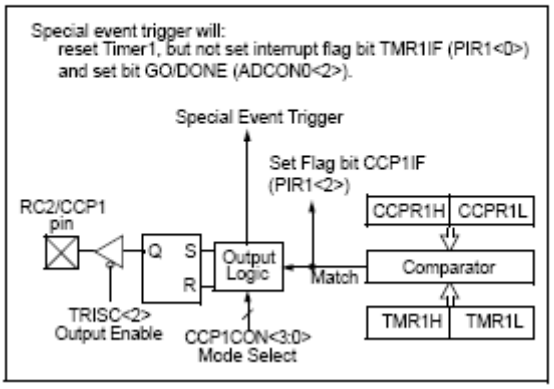
The special event trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The special event trigger output of CCP2 resets the TMR1 register pair and starts an A/D conversion (if the A/D module is enabled).

He aquí la configuración de los registros para lograr la interrupción con el special event trigger:

INTCON= b'11xx xxxx'
PIR1= b'xxxx x0xx'
PIE1= b'xxxx x1x0'
T1CON= b'xx00 0001'(tm1)
CCPR1L= 0x10
CCPR1H= 0x27 (0x2710=d'10000')
CCP1CON=b'xxxx 1011'(special event trigger)

FIGURE 8-2: COMPARE MODE OPERATION BLOCK DIAGRAM



x: es un bit que no está involucrado con el módulo
El prescaler del timer1 esta en 1:1 y funciona con el reloj interno

El periodo de tiempo de la interrupción se calcula de la siguiente manera:

Tiempo= 4*Tosc*CCPR1*Preescaler. Donde Tosc es el inverso de la frecuencia del clock del PIC (4 Mhz)

Tiempo= 4*1/(4000000)*10000*(1/1)= 0,01 segundos

O sea que habrá una interrupción por el módulo CCP cada centésima de segundo.

Sabiendo esto se crea una variable "CENTISEGUNDO" que incrementa en cada interrupción y se resetee cada vez que llegue a 100. Cuando este llegue a 100 habrá pasado 1 segundo, entonces se incrementa otra variable llamada "SEGUNDO". Habrá otra para minutos y horas.
Pero el tiempo debe ser mostrado en forma decimal y los registros del pic se guardan en forma hexadecimal. Entonces si se guardara en forma hexadecimal habría que luego hacer una transformación. Para evitar esto directamente se guarda en forma decimal agregando nuevas variables para la unidad y la decena. O sea las variables que incrementarán a su respectivo turno son:

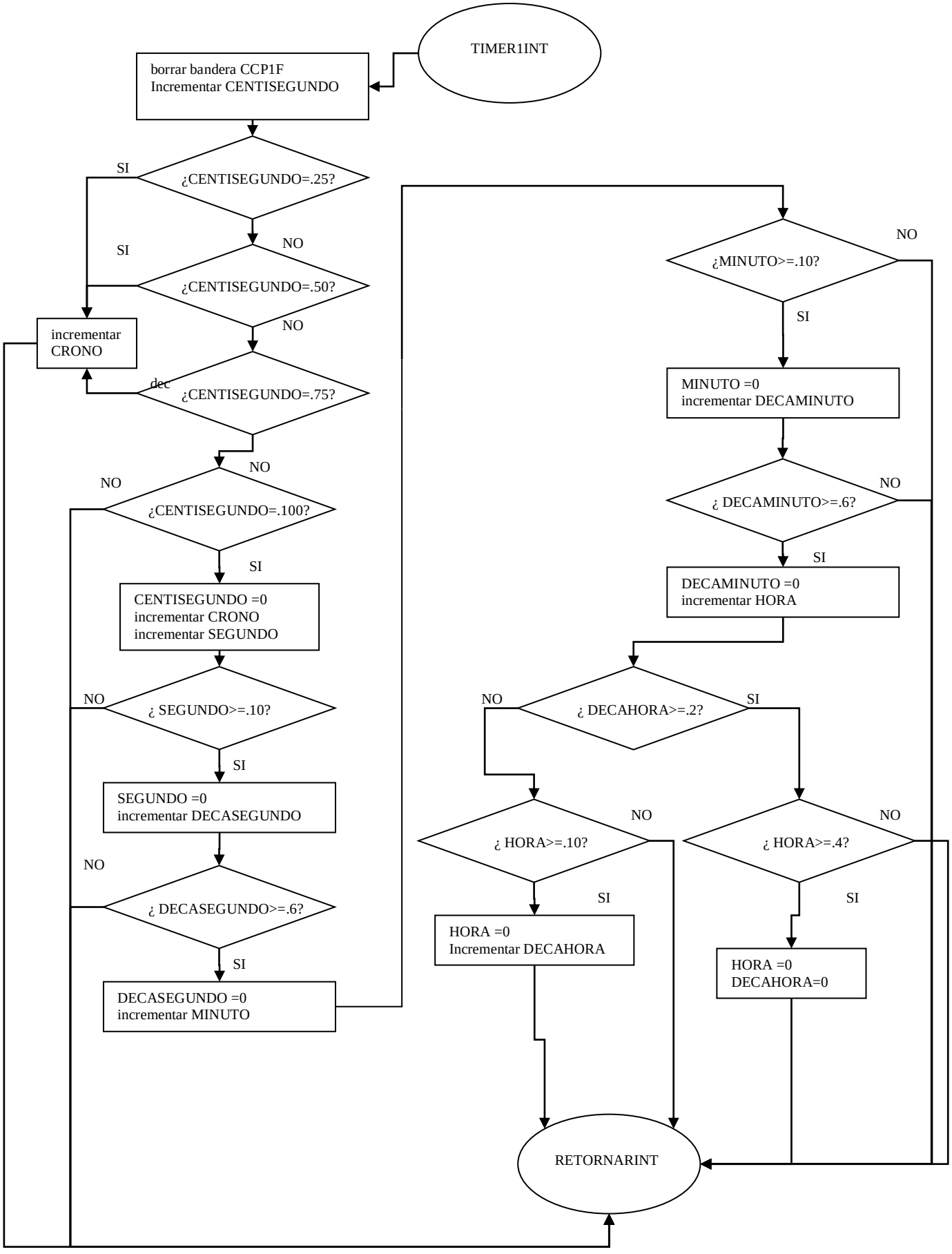
CENTISEGUNDO: Se incrementa cada centésima de segundo y se resetea cada vez que llega a d'100'.
SEGUNDO: Se incrementa cada segundo y se resetea cuando supera 9
DECASEGUNDO: se incrementa cada 10 segundos y se resetea cuando llega a 6
MINUTO: se incrementa cada minuto y se resetea cuando supera 9
DECAMINUTO: se incrementa cada 10 minutos y se resetea cuando llega a 6
HORA: se incrementa cada hora
si DECAHORA es menor a 2 y se resetea cuando llega a 10.
Si DECAHORA es 2 entonces se resetea cuando llega a 4.
DECAHORA: se incrementa cada 10 horas y se resetea cuando DECAHORA es 2 y HORA es 4.

Además se agrega la variable CRONO que se incrementa si CENTISEGUNDO es d'25', d'50', d'75' o d'100'. Esta variable es usada por aplicaciones externas las cuales resetean y checkean su valor, logrando así contar de a ¼ de segundo.

Estas variables pueden directamente cargarse en los displays sin necesidad de hacer otra conversión.
Por ejemplo si se quiere mostrar la hora y los minutos
Display1= DECAHORA
Display2=HORA
Display3=DECAMINUTO
Display4=MINUTO

Nota: La manera en que se cargan los displays se muestra mas adelante.

Diagrama de flujos de Interrucpción CCP1



Refresco de pantalla:

Para mostrar información al usuario se usan 4 displays conectados en paralelo a un conversor bcd y cada uno a un transistor que habilita la alimentación.
El Pic usa 4 pines para mandar un número binario al conversor y este deja prender los segmentos correspondiente para formar un número digital. La base de los transistores estan conectados cada uno a un pin del pic. En total están involucrados 8 pines y elegí los del PUERTO B
El puerto B es configurado como salida con el registro TRISB=0x00
Los primeros 4 bits son usados para mandar al conversor y los otros 4 para controlar cada uno un transistor.

En cada refresco se alimentan los displays de a uno de izquierda a derecha. Cada display permanece un tiempo encendido a travez de un retardo, luego se apaga y se pasa a encender el siguiente. Nunca hay dos displays siendo alimentados al mismo tiempo.

Cuando un display se enciende, el bcd debe habilitar los segmentos que corresponden para ese display.

Haciendo el ciclo de refresco rápido y con una buena frecuencia, se logra dar la sensación de que los cuatro estan siendo alimentados al mismo tiempo.

Para lograr que el refresco tenga una buena frecuencia se usa una interrupción, generada por el timer 0.

TIMER 0
Los registros asociados al TIMER 0 son:

TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
01h,101h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

Se usa el timer 0 con reloj interno y sin preescaler o sea 1:2. Se lo hace contar desde 00 hasta 0xFF. Entonces la interrupción se activa cada 512 ciclos de instrucción. O sea que cada vez que el programa principal realiza aproximadamente 512 ciclos de instrucción se refresca la pantalla.

Para el modo interrupción timer0 los registros son configurados de la siguiente manera:

INTCON= 1x1x x0xx
OPTION_REG=xx0x 1xxx
x: Bit cuyo valor no afecta a la interrupción que se usará

Hay que considerar que ya hay una interrupción que cuenta cada 10000 ciclos de instrucción y que tiene la misión de marcar el tiempo.

Cuando un timer llega a su límite activa su respectiva bandera. Para desactivar la bandera debe hacerse por software. Si el bit GIE esta activado y se activa una bandera

de interrupción, inmediatamente se llama a la interrupción y se deshabilita GIE hasta que se salga de la interrupción.

Puede pasar, y de hecho pasa constantemente, que el otro timer llegue también a su límite sin haber salido de la primera interrupción. El segundo timer activa su bandera. Cuando se salga de la primera interrupción y se habilite el GIE inmediatamente al estar la segunda bandera activada se vuelve a llamar a interrupción pero esta vez ocacionada por el otro timer.

El problema puede aparecer si una interrupción por un timer tarda tanto que el otro timer llega a su límite mas de una vez. Porque al salir de la primera interrupción se leerá un solo ravasamiento por el segundo timer, perdiéndose información.
Esto es crítico si se quiere marcar una hora precisa. El refresco no puede tardar mas de 10000 ciclos porque se perdería de contar alguna centésima de segundo.

Si bien con 10000 ciclos se pueden ejecutar hasta 10000 instrucciones lo cual parece demasiado, el refresco tiene retardos importantes. Son los 4 retardos de cada display los principales consumidores de tiempo.

La mayoría de las instrucciones del PIC consumen 1 ciclo y las otras solo llegan a consumir 2 ciclos.

El bucle en cuestión que se repite 4 veces es el siguiente

```
MOVLW 0XFF;
MOVWF RETARDOM

ESPERA1:      NOP ; (1 ciclo)
NOP; (1 ciclo)
NOP; (1 ciclo)
DECFSZ RETARDOM ; (1 ciclo a menos que se salte una intrucción)
GOTO ESPERA1; ( 2 ciclo)
```

El total de ciclos reloj consumido en cada ciclo de bucle es: 6
6 ciclo reloj/ciclo bucle* 256ciclo bucle =1536 ciclo reloj
1536ciclo reloj/diplay*4 displays= 6144 ciclos reloj

Los 4 retardos consumen aproximadamente 6144 ciclos en total dejando un margen de mas de 3000 ciclos. Mas que suficiente para el resto de las instrucciones si consideramos que el programa completo solo tiene aprox. 2700 líneas.
Con esto se asegura que no se perderá de registrar cada vez que el TIMER 1 llegue a 10000.

La pantalla:
Como puede observarse el PIC es el encargado de que el conjunto de displays cumpla la función de pantalla. Esto le demanda un porcentaje de tiempo importante de procesamiento ocupado solo en la pantalla(aprox. 90% del tiempo). Gracias a que el tiempo de respuesta del resto de programa no es crítico y las tareas a realizar son relativamente fáciles, se tolera este desperdicio de tiempo.
Lo que vincula al programa principal con el dispositivo de pantalla es la función "MOSTRAR"y las variables OPCIONMOS, PARPADEOBIT, LETRADER Y LETRAIZQ

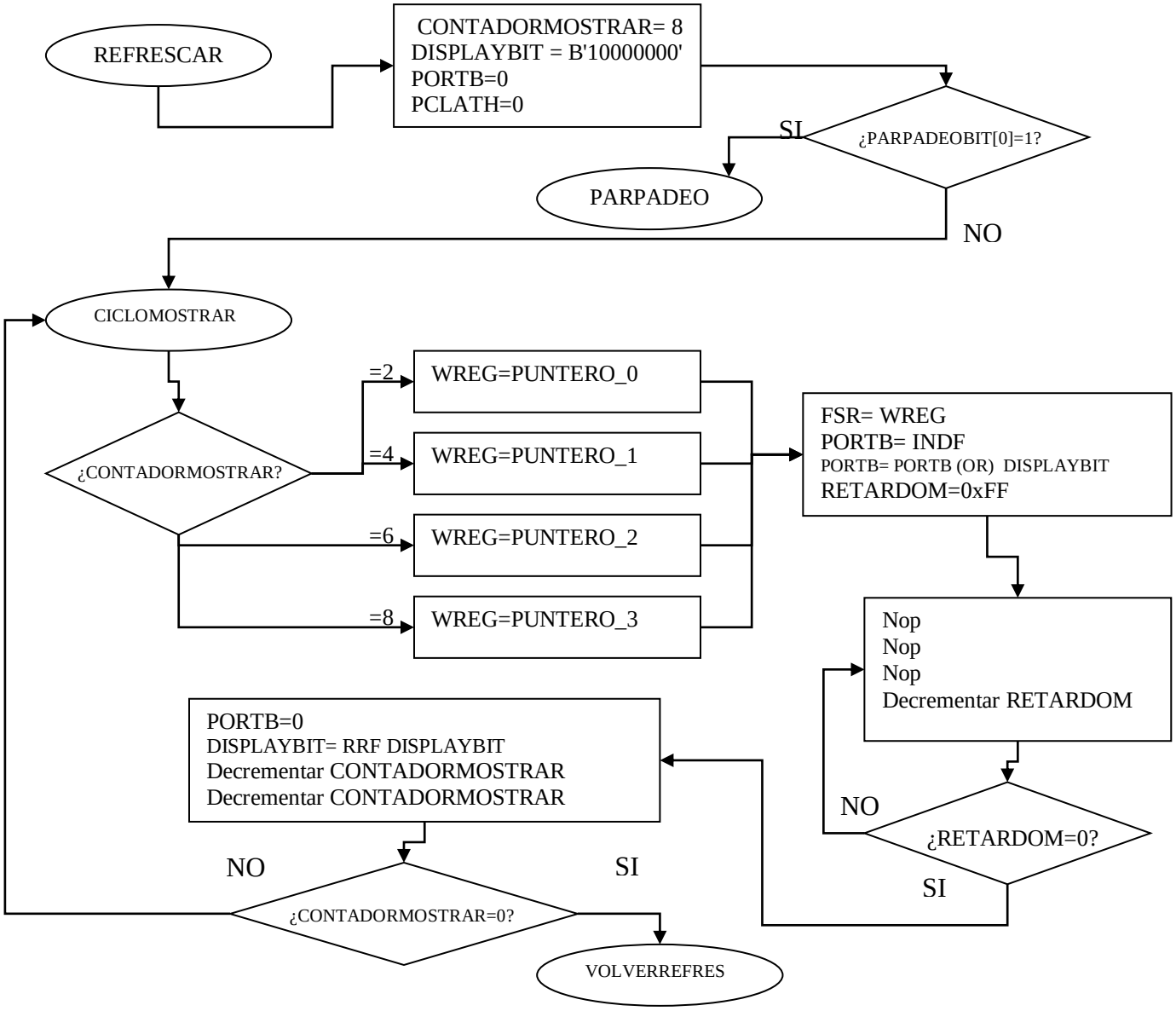
El programa principal carga a OPCIONMOS y llama a la función MOSTRAR con un CALL.
La Función MOSTRAR carga 4 punteros con los que trabaja el refresco: PUNTERO_0, PUNTERO_1, PUNTERO_2 Y PUNTERO_3.
Por ejemplo si se quiere mostrar la hora entonces los punteros deberían cargarse de la siguiente manera.
PUNTERO_3= dirección (DECAHORA)
PUNTERO_2= dirección (HORA)
PUNTERO_1= dirección (DECAMINUTO)
PUNTERO_0= dirección (MINUTO)

A traves de OPCIONMOS la función MOSTRAR elige que 4 direcciones hay que cargar. Esto está previamente programado.

Al producirse un refresco y usando los punteros, los display se cargan de a uno por vez.

Las interrupción REFRESCAR y la función MOSTRAR serían como los "drivers" de la pantalla.

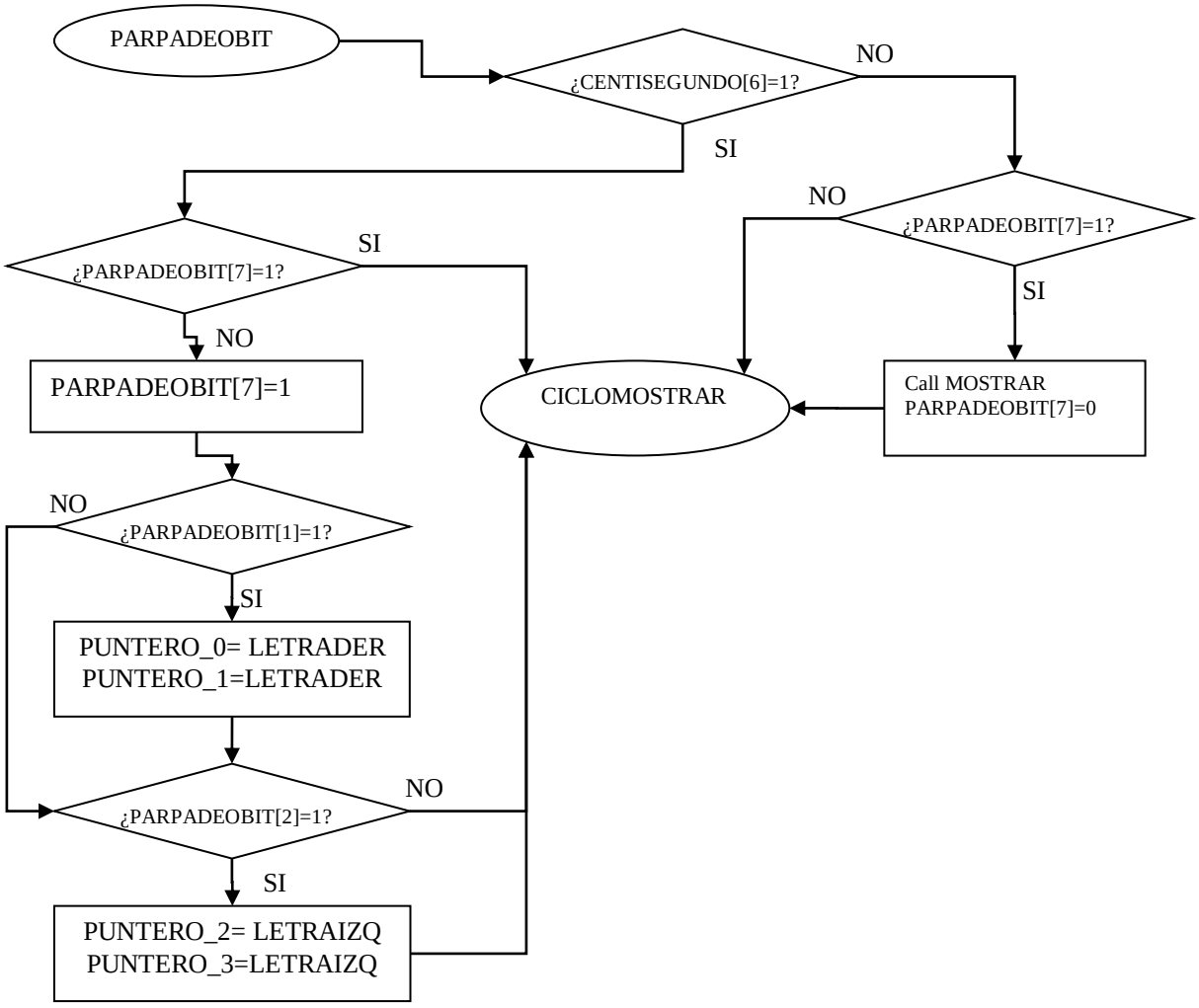
Diagrama de flujo de Refrescar Pantalla



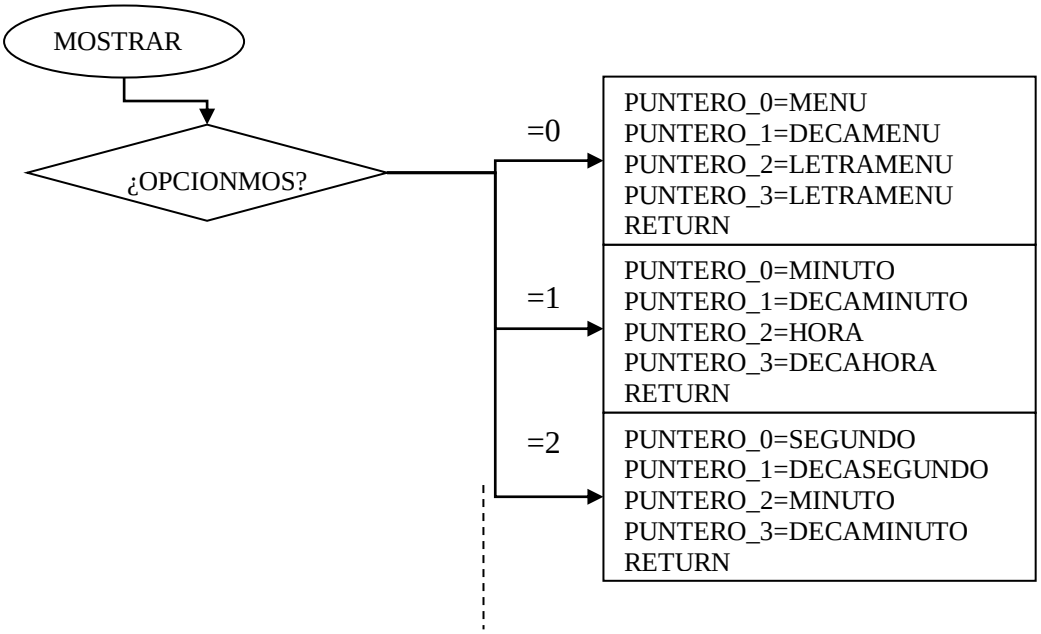
PARPADEOBIT es una variable de 8 bits usada para producir el efecto de parpadeo.
El bit 0 indica si esta activado el parpadeo
El bit 1 indica que parpadean los 2 displays de la derecha
El bit 2 indica que parpadean los 2 displays de la izquierda
El bit 7 es un bit que usa la función PARPADEO para saber en que fase del parpadeo se encuentra.
Aprovechando el conteo de tiempo por el timer1 se puede observar que los bits del registro CENTISEGUNDO varían de forma cíclica. El bit 6 se pone en 1 a las 64 centésimas de segundo luego de haberse reseteado y se pone en 0 cuando se vuelve a resetear o sea cuando CENTISEGUNDO llega a .100 o sea cuando se cumpla 1 segundo. Por lo tanto se puede producir un parpadeo en cada segundo. El parpadeo consiste en intercambiar el valor que se está mostrando en un display por otro cíclicamente. Así se puede lograr que prenda y apague cargando el valor 0x0F en el bcd. Pero tambien se usa con otros caracteres especiales del bcd estos son los números hexadecimales: A, B, C, D y E. Entonces lo que mostrará el parpadeo se guarda en 2 variables: LETRAIZQ Y LETRADER. Para el parpadeo se han realizado 3

posibilidades. Que parpadeen los 2 displays de la izquierda, los 2 de la derecha o los 4 a la vez

Diagrama de flujo de "Parpadeo"



Función MOSTRAR:

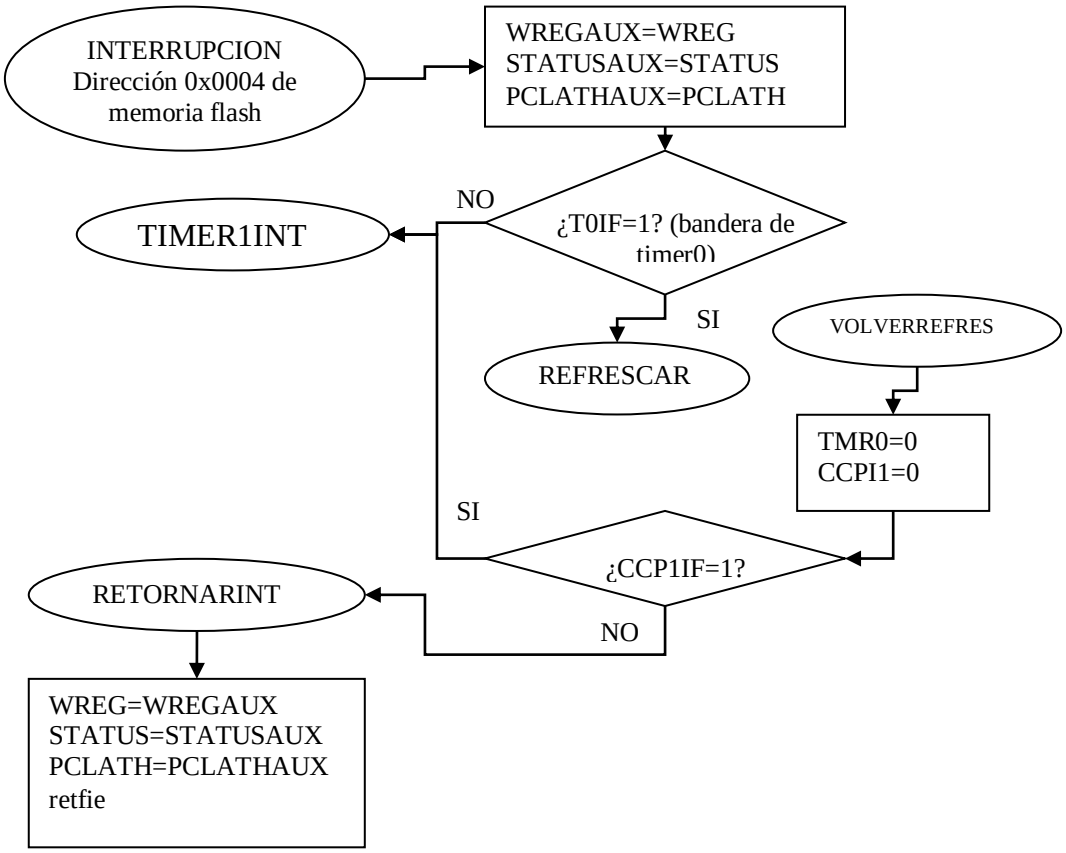


De la misma forma se generan en total 9 pantallas diferentes.

INTERRUPCIONES

Al existir 2 interrupciones cuando se la llame esta debe decidir cual de las 2 hay que atender. Como el stack del PIC solo guarda la dirección del PC (Contador de Programa), es necesario guardar además los registros importantes en algunos registros auxiliares. Los registros a guardar son: WREG, STATUS y PCLATH. Al finalizar la interrupción hay que recargar estos registros.

Diagrama de flujo de Interrupciones



Primero se revisa si la bandera de timer0 es 1 sino se deduce que la interrupción fue llamada por el módulo CCP(timer1). Cuando se vuelve del refresco, el cual tarda bastante, es muy probable que el timer1 ya haya llegado a su límite por lo que se verifica la bandera CCP1IF

Función PROCESO_ACTUAL:

Es una función creada en la memoria 0x0700 el cual controla todas las etapas del proceso. Para controlar el motor, la linterna y el calentador solo se necesitan cambiar 3 bits del puerto A. Se le ha excluido al programa principal el control de estos bits directamente. Lo único que el programa principal puede hacer es apagar el calentador. Para realizar otra operación debe pedirlo cargando la variable PROCESO. Cuando el

bucle principal llame a la función PROCESO_ACTUAL esta realizará la tarea encargada.

PROCESO_ACTUAL no solo es capaz de cambiar de estado los bits de control sino que es la encargada de la SEGURIDAD del proceso.

Chequea que no se cometan errores de proceso. Hay que tener en cuenta que se trabaja con líquido caliente que puede ocacionar accidentes electricos, quemaduras y suciedad, y sobre todo hay que tener en cuenta que el calentador tiene la posibilidad de generar un incendio si no se maneja apropiadamente.

Errores de proceso

Cuando un error surge, PROCESO_ACTUAL tiene como misión llevar la máquina a un estado seguro y señalar el error.
El estado seguro es: Válvula cerrada y calentador apagado. Por eso es que el programa principal puede apagar el calentador, ya que lo acerca a un modo seguro. Pero no puede encenderlo sin permiso.

Los errores posibles que fueron concebidos son:

ERROR 0= ¡¡¡error con muy alto riesgo!! se esta calentando y no hay variación de temperatura por mas de 45 segundos; riesgo de incendio, posible error de sensor, controle el nivel de liquido a calentar, posible error o daño en el calentador.

ERROR 1= ¡¡¡error con alto riesgo !! Se ha intentado cerrar la valvula por mas de 10 segundos sin éxito. Posible error en el sensor de cierre, posible fallo del motor. Riesgo de revalsamiento, quemaduras y accidentes eléctricos

ERROR 2= ¡¡¡error con riesgo!!! el proceso de abrir valvula siguió activado por mas de 30 segundos; es posible que no haya recipiente debajo de la válvula. Riesgo por derramar líquido caliente.

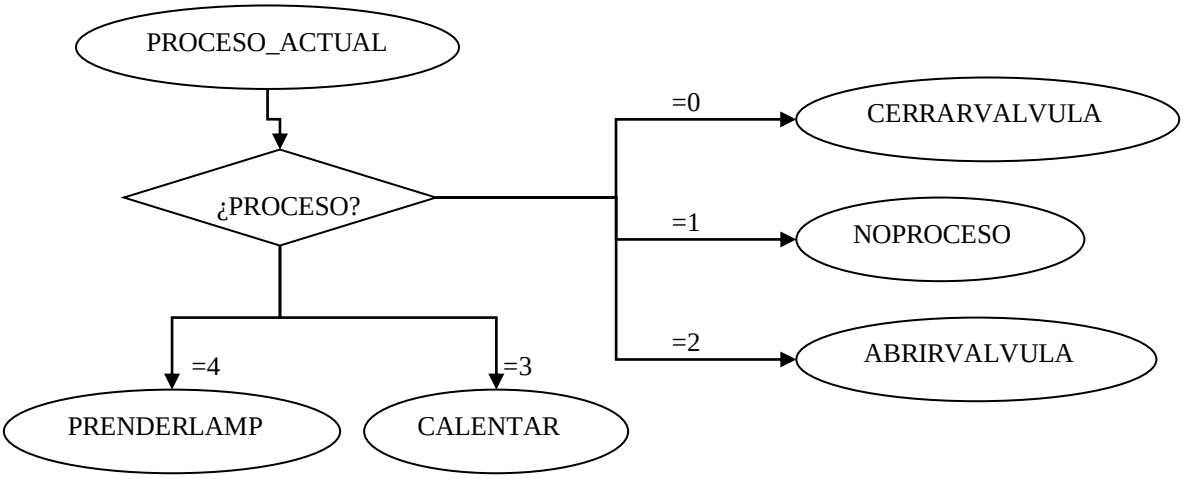
ERROR 3= ¡¡¡error con riesgo moderado!!! se ha energizado el motor pero la valvula se detectó cerrada luego de 3 segundos. Es muy probable que el motor esté fallando. Es posible que haya falla en sensores.

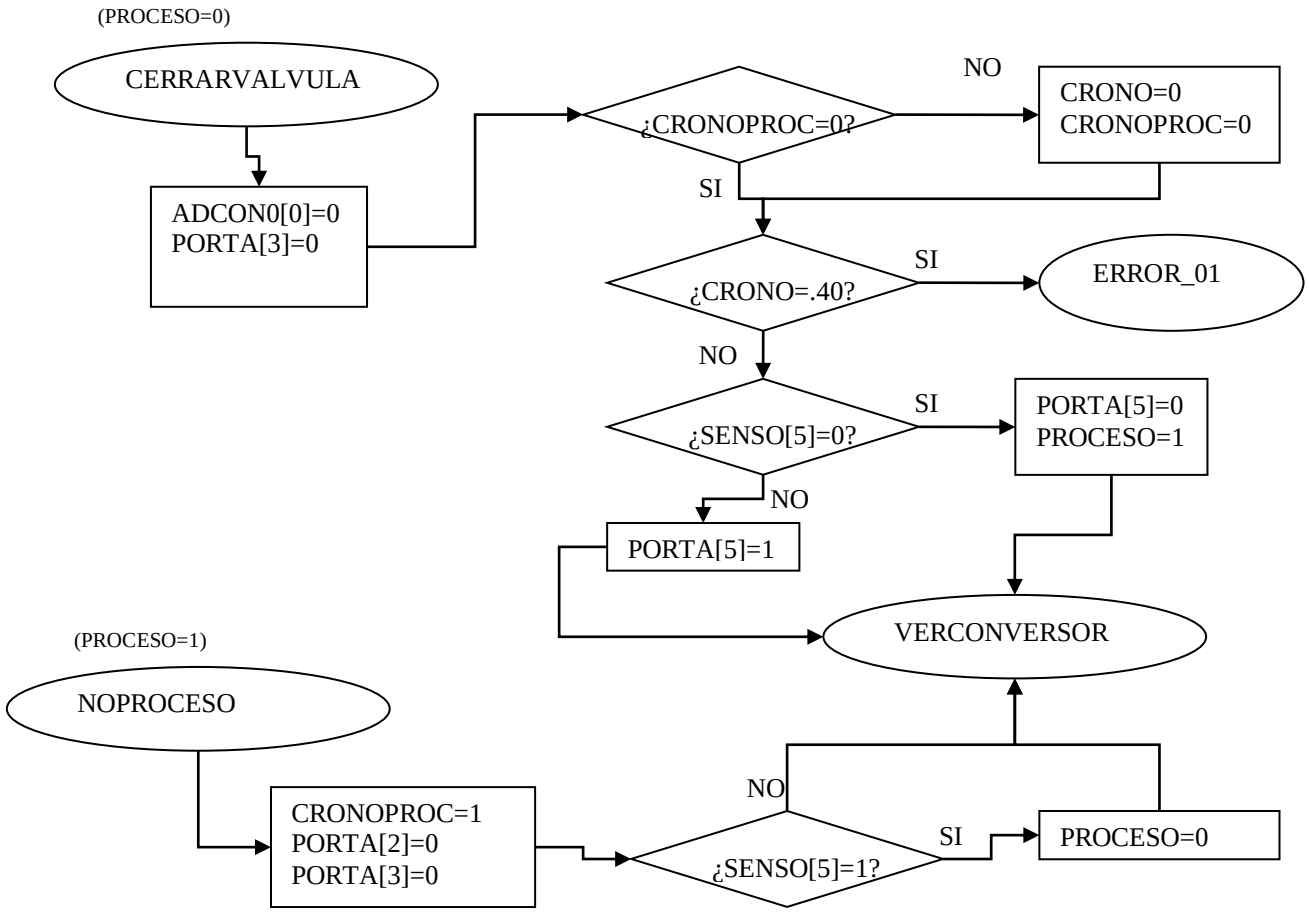
ERROR 4= ;¡¡¡peligro!!! se ha intentado calentar con la valvula abierta. No se permite calentar con la válvula abierta. Es posible no se haya dado tiempo a que se termine de cerrar la válvula. Es posible que se haya forzado mecánicamente la apertura de la misma mientras se estaba calentando. Posible extraña y repentina falla del sensor.

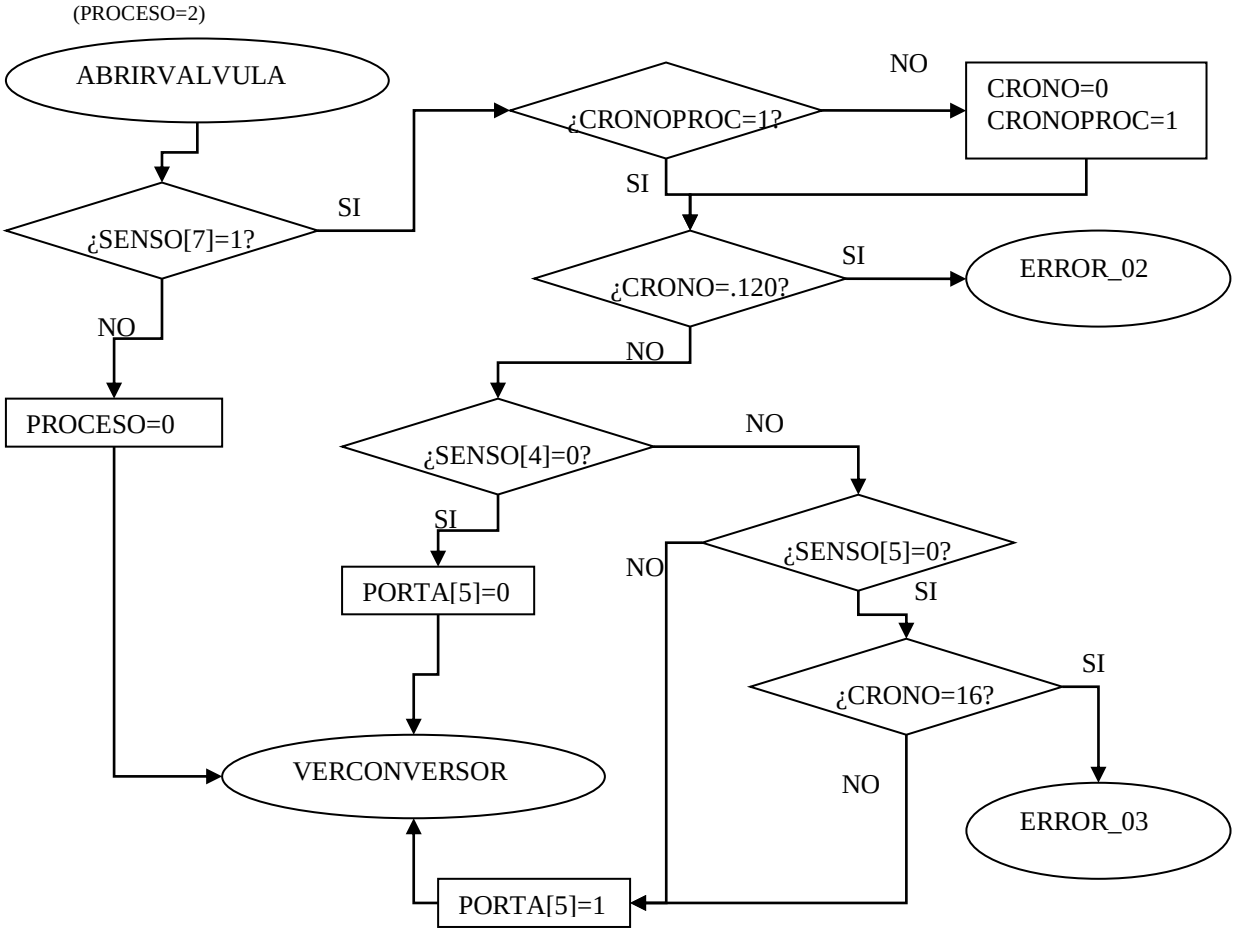
ERROR 5= ¡¡¡error con muy alto riesgo!! ha pasado mas de 9 minutos (y 15 seg de precalentamiento) y el proceso de calentar no ha terminado. Riesgo de incendio. Riesgo de sobrepresion en el líquido y salpicaduras. Es posible que se haya exigido llegar a una temperatura elevada desde una muy baja y que haya demasiado líquido. Es posible que haya falla en el sensor. Controle el nivel del líquido con respecto al sensor y calentador.

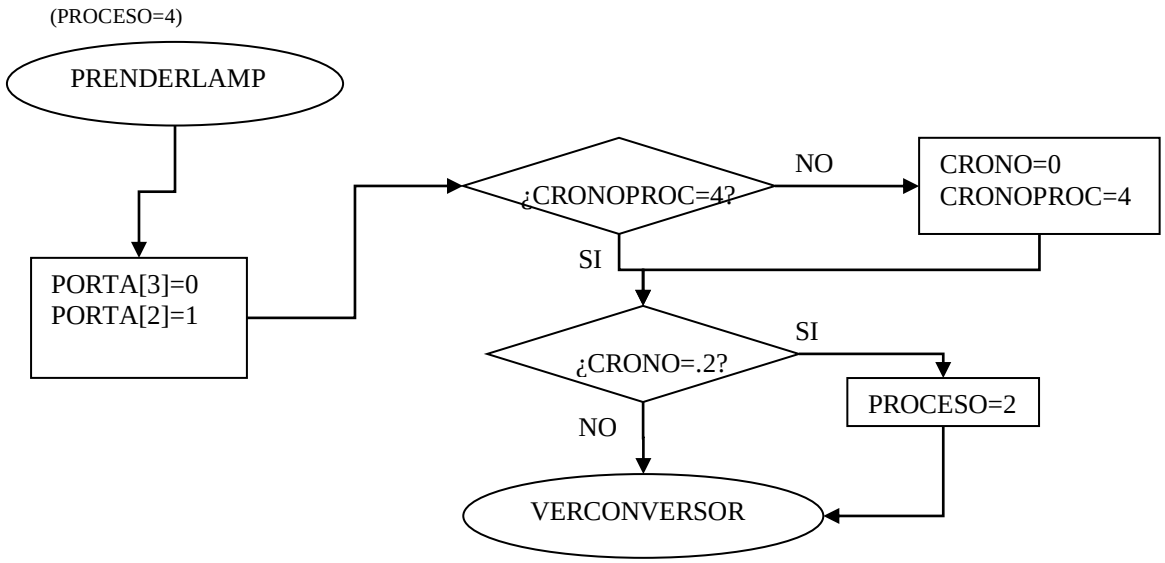
ERROR 6=¡¡¡peligro!!! Se ha intentado calentar sin líquido y la operación ha sido abortada. El calentador siempre debe estar sumergido. Revise el nivel de líquido, es posible que haya falla en el sensor de la boya.

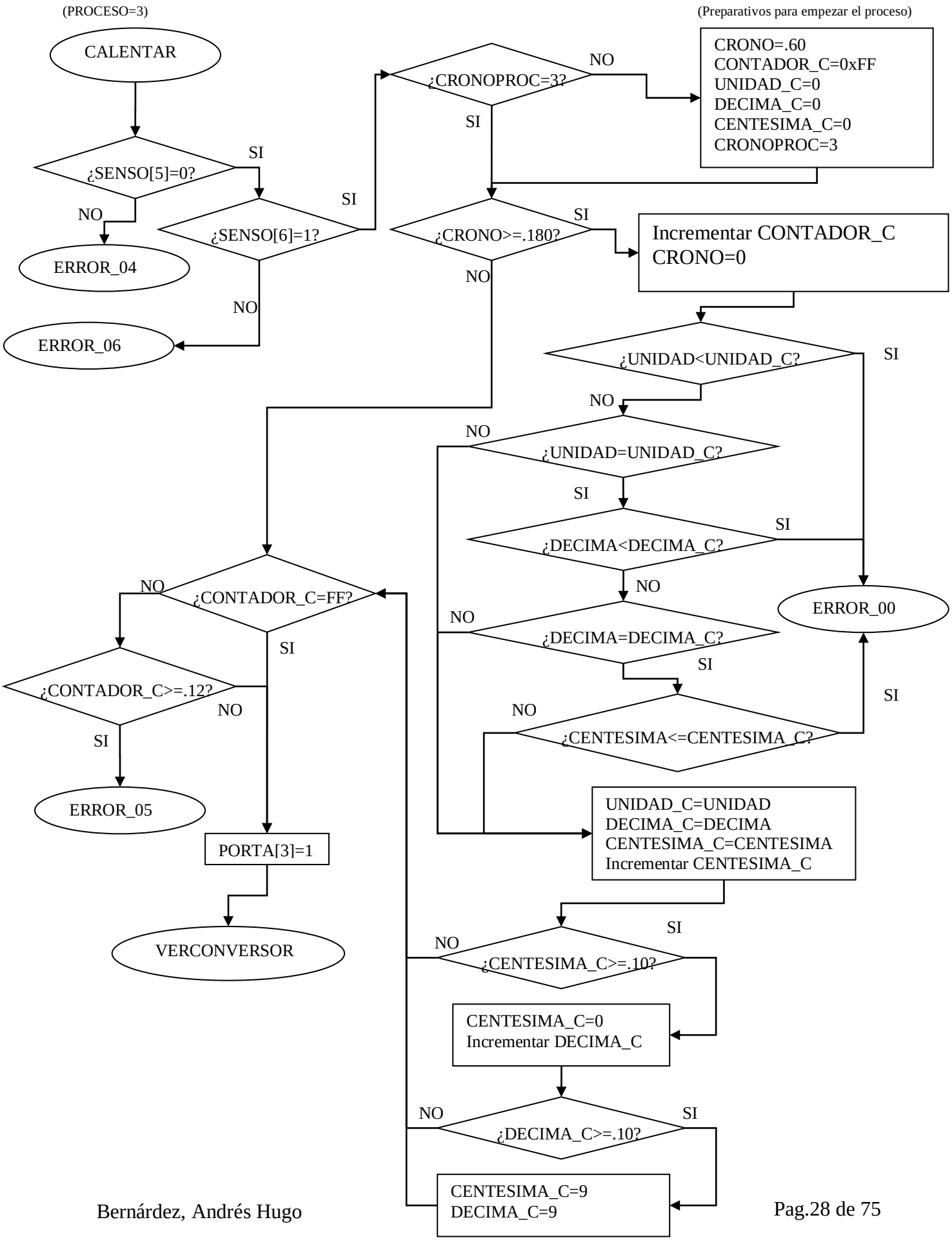
Diagramas de flujo de Proceso Actual



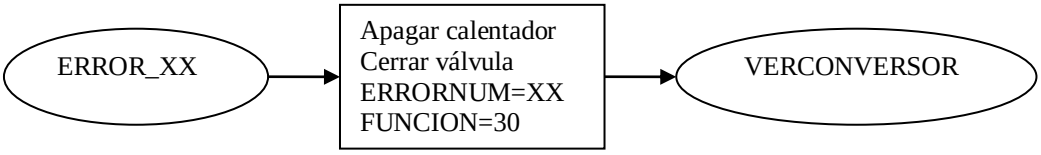



(PROCESO=2)
ABRIRVALVULA
¿SENSO[7]=1?
SI
NO
PROCESO=0
¿CRONOPROC=1?
NO
CRONO=0
CRONOPROC=1
SI
¿CRONO=.120?
SI
ERROR_02
NO
¿SENSO[4]=0?
NO
SI
PORTA[5]=0
¿SENSO[5]=0?
NO
SI
¿CRONO=16?
SI
NO
ERROR_03
VERCONVERSOR
PORTA[5]=1


(PROCESO=4)
PRENDERLAMP
PORTA[3]=0
PORTA[2]=1
¿CRONOPROC=4?
NO
CRONO=0
CRONOPROC=4
SI
¿CRONO=.2?
SI
PROCESO=2
NO
VERCONVERSOR


(PROCESO=3)
CALENTAR
¿SENSO[5]=0?
SI
NO
ERROR_04
¿SENSO[6]=1?
SI
NO
ERROR_06
¿CRONOPROC=3?
NO
SI
(Preparativos para empezar el proceso)
CRONO=.60
CONTADOR_C=0xFF
UNIDAD_C=0
DECIMA_C=0
CENTESIMA_C=0
CRONOPROC=3
¿CRONO>=.180?
SI
NO
Incrementar CONTADOR_C
CRONO=0
¿UNIDAD<UNIDAD_C?
SI
NO
¿UNIDAD=UNIDAD_C?
NO
SI
¿DECIMA<DECIMA_C?
SI
NO
ERROR_00
¿DECIMA=DECIMA_C?
NO
SI
¿CENTESIMA<=CENTESIMA_C?
NO
SI
UNIDAD_C=UNIDAD
DECIMA_C=DECIMA
CENTESIMA_C=CENTESIMA
Incrementar CENTESIMA_C
¿CENTESIMA_C>=.10?
NO
SI
CENTESIMA_C=0
Incrementar DECIMA_C
¿DECIMA_C>=.10?
NO
SI
CENTESIMA_C=9
DECIMA_C=9
¿CONTADOR_C=FF?
NO
SI
¿CONTADOR_C>=.12?
SI
NO
ERROR_05
PORTA[3]=1
VERCONVERSOR

La estructura de los errores de los errores de proceso



El conversor:

El PIC 16F873A tiene un conversor de tensión analógica digital integrado con una resolución de 10 bits. En este programa solo se usa el pin 0 del puerto A como AD.

Los registros involucrados en el conversor son

TABLE 11-2: REGISTERS/BITS ASSOCIATED WITH A/D

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on MCLR, WDT
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	00-- 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
89h(1)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits			0000 -111	0000 -111
09h(1)	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.
Note 1: These registers are not available on 28-pin devices.

Los registros quedan se configuraron de la siguiente manera:
ADCON0=B'01000000'(channel 0, 8Tosc)
ADCON1=B'00001110' (todos pines digitales menos el PORTA[0])
TRISA=B'00xx xxx1'(pin 0 de puerto A como entrada)

El sensor tiene por salida 10 mV por grado centigrado y llega a medir 150 grados sin destruirse. Por lo tanto la tensión máxima que puede tener en su salida es 1,5 V. Por otro lado solo se medirá a partir de temperatura ambiente que en esta zona suele ser por arriba de cero. Así que la tensión mínima considerada que marcará el sensor es 0V. sin embargo para no complicar la circuitería ni ocupar mas de un pin del puertoA se tomará la configuración que toma a V+=Vdd=5V y V-=Vss=0V. Por supuesto que se tendría mejor resolución si se bajara V+ pero ya que el sensor tiene un error de 0.5ºC y

que el proceso no requiere tanta precisión simplemente aprovecho los 10 bits del conversor y con eso logro suficiente precisión.

$Apreciación = \frac{5V}{2^{10}} = 4{,}8828mV$ que equivale a 0,48828 grados centígrados en la escala del sensor.
La apreciación es de 0.488 grados y el error es de 0.5 grado. Por estos motivos se puede tener un error máximo de 1 grado.

La conversión se hace con los 8 bits mas significativos en el registro ADRESH y los otros en ADRESL.
La siguiente tabla muestra que valor puede aportar un bit de estos registros al valor total y que valor sumarle a cada variable auxiliar. Luego de haberse chequeado todos los bits y de haber cargado las variables auxiliares hay que acomodarlas.

TENSION A CONVERTIR (Vmax-Vmin)	**5**

	VALOR DE TENSION QUE APORTA CADA BIT	EJEMPLO CON BITS	APORTE PARA CADA CIFRA						
BIT			UNIDAD	DECIMA	CENTESIMA	MILESIMA	DECMIL	CENMIL	BIT
ADRESH,7	2,5	**0**	**2**	**5**	**0**	**0**	**0**	**0**	9
ADRESH,6	1,25	**1**	**1**	**2**	**5**	**0**	**0**	**0**	8
ADRESH,5	0,625	**0**	**0**	**6**	**2**	**5**	**0**	**0**	7
ADRESH,4	0,3125	**0**	**0**	**3**	**1**	**2**	**5**	**0**	6
ADRESH,3	0,15625	**1**	**0**	**1**	**5**	**6**	**2**	**5**	5
ADRESH,2	0,078125	**1**	**0**	**0**	**7**	**8**	**1**	**2**	4
ADRESH,1	0,0390625	**0**	**0**	**0**	**3**	**9**	**0**	**6**	3
ADRESH,0	0,01953125	**0**	**0**	**0**	**1**	**9**	**5**	**3**	2
ADRESL,7	0,00976563	**1**	**0**	**0**	**0**	**9**	**7**	**6**	1
ADRESL,6	0,00488281	**1**	**0**	**0**	**0**	**4**	**8**	**8**	0
	ADRESH:	76							
	ADRESL:	192							
	RESPUESTA:	1,49902344							

APRECIACION	0,00488281

Como cada cifra es decimal, no pueden tener mas de un valor de 9, si la sumatoria total de todos los aportes dio más hay que pasar el exceso a la siguiente cifra mas significativa. Esto se hace agregando el resultado de la división entera por diez en la siguiente cifra y reemplazando la cifra actual por el resto de dicha división.

Utilizando registros auxiliares se guarda la unidad, décima, centésima y milésima de volt para que puedan ser mostrados directamente en los displays.
Como el líquido es a base de agua evapora aprox. a los 100ºC y el líquido no aumentará su temperatura mas que eso. Entonces solo se permitirá calentar hasta 99ºC o sea 0,99Volt pero no se recomienda llegar hasta esa temperatura ya que la máquina es solo un prototipo.
En consecuencia cuando se esté midiendo temperatura la unidad tendría que ser cero. La décima y la centésima de Volt representan respectivamente la decena y la unidad de grado centígrado.

Cuando se activa la conversión, esta no se hace constantemente, ya que no se requiere. Se ha elegido hacer 12 veces por segundo para que el usuario pueda ver la variación rápidamente.
Observando la variable CENTISEGUNDO se concluye que el bit 2 cambia de cero a uno 12 veces por segundo. Aprovechando esto se genera un pulso de referencia.

CENTISEGUNDO= 4= B'0000 0**1**00'
=8= B' 0000 1**0**00'
=.12=B'0000 1**1**00'
= .16=B'0001 0**0**00'

El bit2 cambia cada 4 centésimas de segundo o sea tiene un ciclo de 80 mseg. Que equivaldría a 12.5 hz pero cuando CENTESIMA llega a .100 se tiran todos los bits a cero y el ciclo se trunca justo antes producirse un nuevo pulso. Por eso solo hay 12 pulsos cuadrados por segundo.

El registro ALARMA_T es utilizado para saber cuando el valor de Tensión leído por el conversor supera a un valor de tensión seleccionado previamente. Esto se hace inmediatamente después de la conversión si la alarma de tensión está activada

Promedio de la entrada:

En los registros ADRESH y ADRESL del micro se guarda la conversión. Esta no es directamente transformada a temperatura ya que es un valor que varía constantemente, especiamente en las cifras menos significativas. Si se transformara la lectura instantanea, se vería que los valores de los displays cambiarían constantemente, lo que es molesto para su lectura. Por otro lado, no es necesario que el proceso sense la temperatura instantánea. Lo que resulta mas útil es llevar un promedio de los últimos valores que se han sensado. Además el promedio es mas representativo que la lectura de un valor instantáneo que pudo haber sido el producto de algun ruido.

En un principio la idea es promediar los últimos 10 valores ingresados. Esto es:

$$ADPROM_k = \frac{ADRES_k + ADRES_{k-1} + ADRES_{k-2} + ... + ADRES_{k-8} + ADRES_{k-9}}{10}$$

Esto se puede escribir como:

$$ADPROM_k = \frac{\sum_{i=0}^{9}(ADRES_{k-i})}{10}$$

Esto tiene la desventaja de guardar y sumar 10 números de 10bits que fácilmente demanda 20 registros de 1byte.

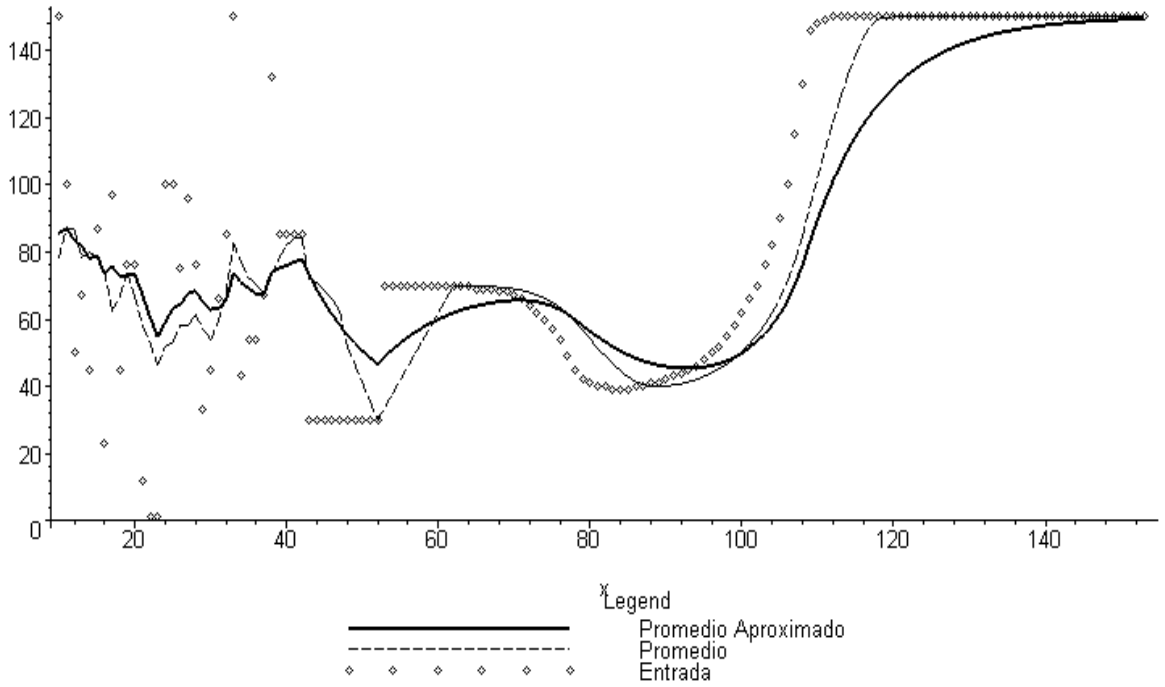
Fácilmente se ve que el promedio es igual al promedio anterior mas el 10% del nuevo valor menos el 10% del valor ingresado hace 11 conversiones.

$$ADPROM_k = ADPROM_{k-1} + \frac{ADRES_k}{10} - \frac{ADRES_{k-10}}{10}$$

sin embargo aún necesitamos saber cual fue el valor hace 11 conversiones.
Aproximándolo se puede decir que ese valor es igual al promedio. De esta manera no hay que guardar ningún valor anterior mas que el promedio.
El promedio aproximado queda como:

$$ADPROM_k \cong ADPROM_{k-1} + \frac{ADRES_k}{10} - \frac{ADPROM_{k-1}}{10}$$

O sea al promedio se le saca un 1/10 y se le agrega el de la nueva lectura dividido 10.
He aquí una gráfica que muestra como trabajan el promedio de los últimos 10 números y el promedio aproximado.



Si la entrada se estabiliza, ambos promedios convergen a ese valor. Como puede verse el promedio aproximado es mas suave que el promedio real. Esto resulta conveniente ya que lo que se pretende es que la lectura varíe con suavidad.

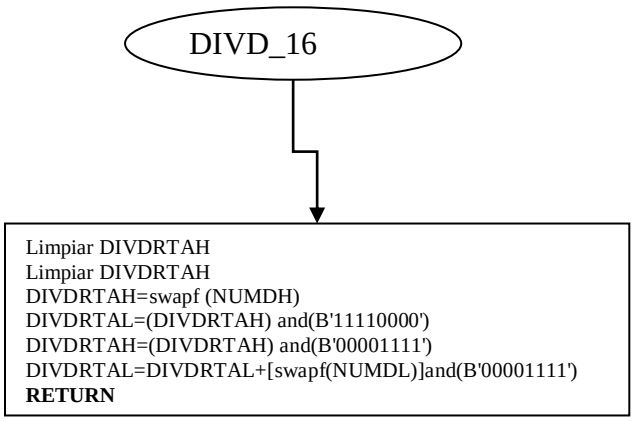
Para promediar los registros ADRESH:ADRESL se hará sumando, restando y dividiendo números de 16 bits.
Como son números de base binaria o incluso de base hexadecimal, en vez de dividir por 10 resulta mejor dividir por números múltiplos de 2. Ya que un número binario dividido 2^n solo debe correrse *n* bits a la derecha. Se elige entonces dividir por 16. De esta manera al dividir un número por 16 solo se corren 4 bits a la derecha o sea 1 nibble a la derecha. La fórmula de promedio aproximado queda así:

$$ADPROM_k \cong ADPROM_{k-1} + \frac{ADRES_k}{16} - \frac{ADPROM_{k-1}}{16}$$

Tambien se pudo elegir "8" pero la idea es lograr suavidad en la convergencia. Además como existe la instrucción SWAPF que invierte los nibbles, es mas conveniente para programar dividir por 16.
Además de estas ventajas existe otra razón: dividir por .10 en el sistema hexadecimal trae problemas por ejemplo:
$\frac{130_{16}}{A_{16}} = 1E,666666666...{}_{16}$ Se necesita guardar mas información para lograr exactitud

en cambio:
$\frac{130_{16}}{10_{16}} = 13_{16}$
Como lo mas fácil es hacer divisiones enteras conviene tratar de evitar obtener cifras después de la coma.

Diagrama de flujo de "DIV_16"(división de número de 2 bytes por 16)



Es notable lo sencillo que es dividir por 16.

Por ser una división entera, se perderá información. Esto se compensa por el hecho que usamos 10 de los 16 bits entonces. Los 10 bits mas significativos(a la izquierda) son los que traen información, el resto de los bits del promedio sirven para guardar la parte "no entera".

Esto en números decimales sería algo así como en vez de trabajar con el 62,15, hacerlo con el 6215 y terminar usando solo las primeras dos cifras.

Errores de la medición

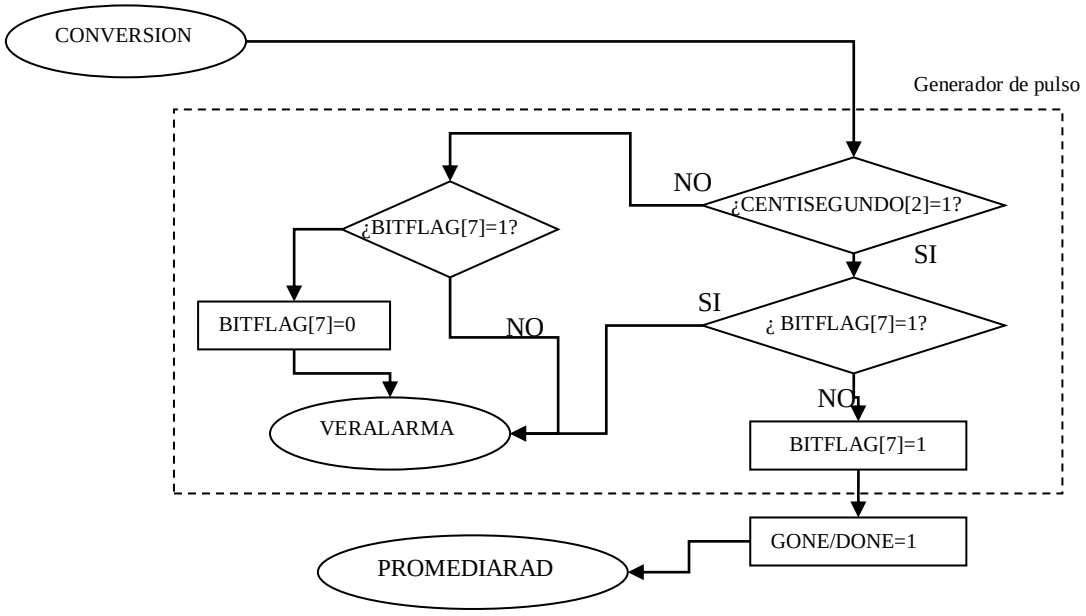
Experimentalmente con un multímetro y un termómetro se ve que:

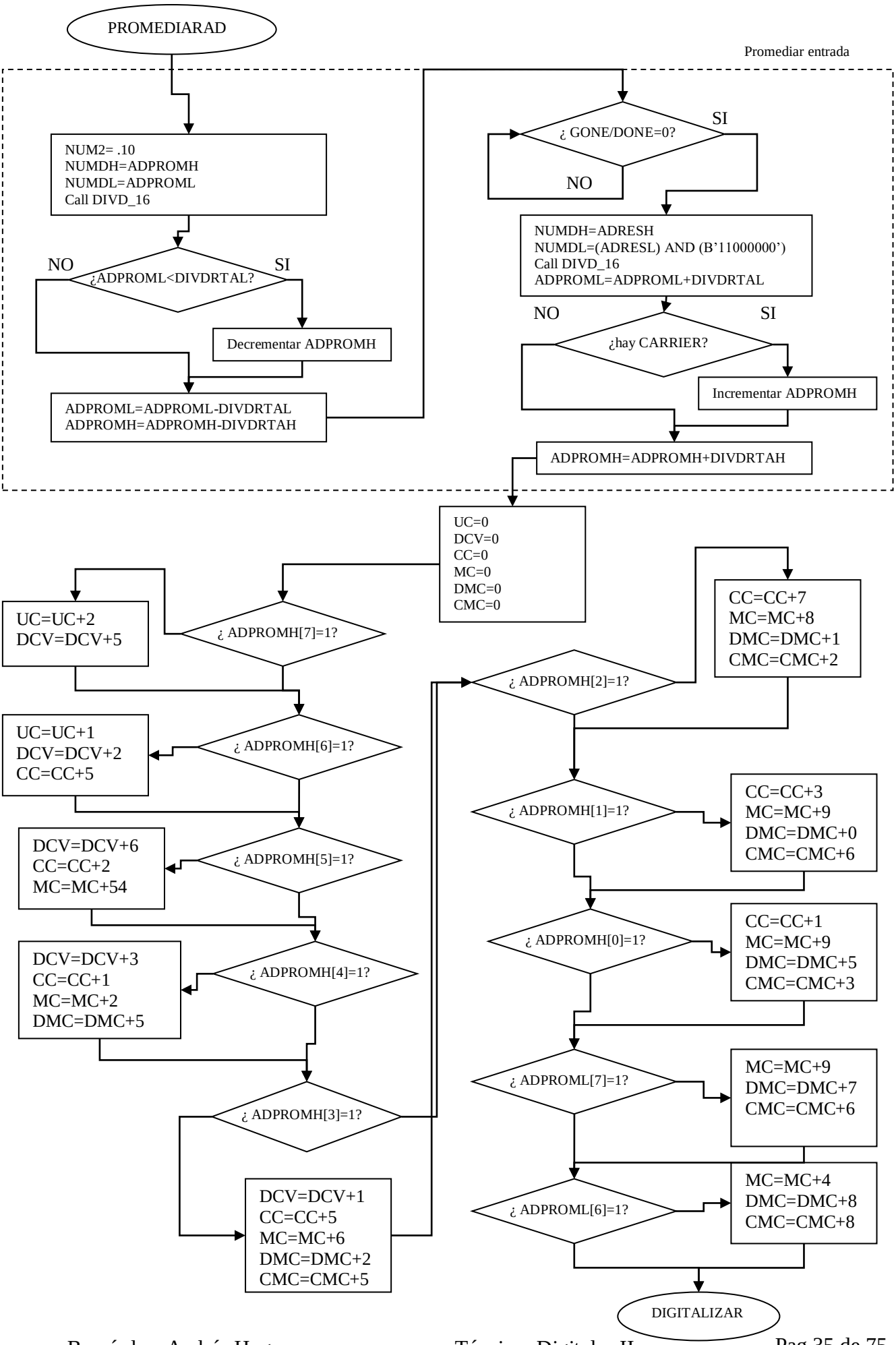
- El valor de tensión que se obtienen con el programa es menor al del multímetro en 10mV.
- Cuando la temperatura se estabiliza. El valor de temperatura medido por el dispositivo es 2,5 grados menos que el termómetro cuando se mide temperaturas cercanas a 50 ºc y 1,5 grados menos cuando se mide temperaturas cercanas a los 10ºc.
  o

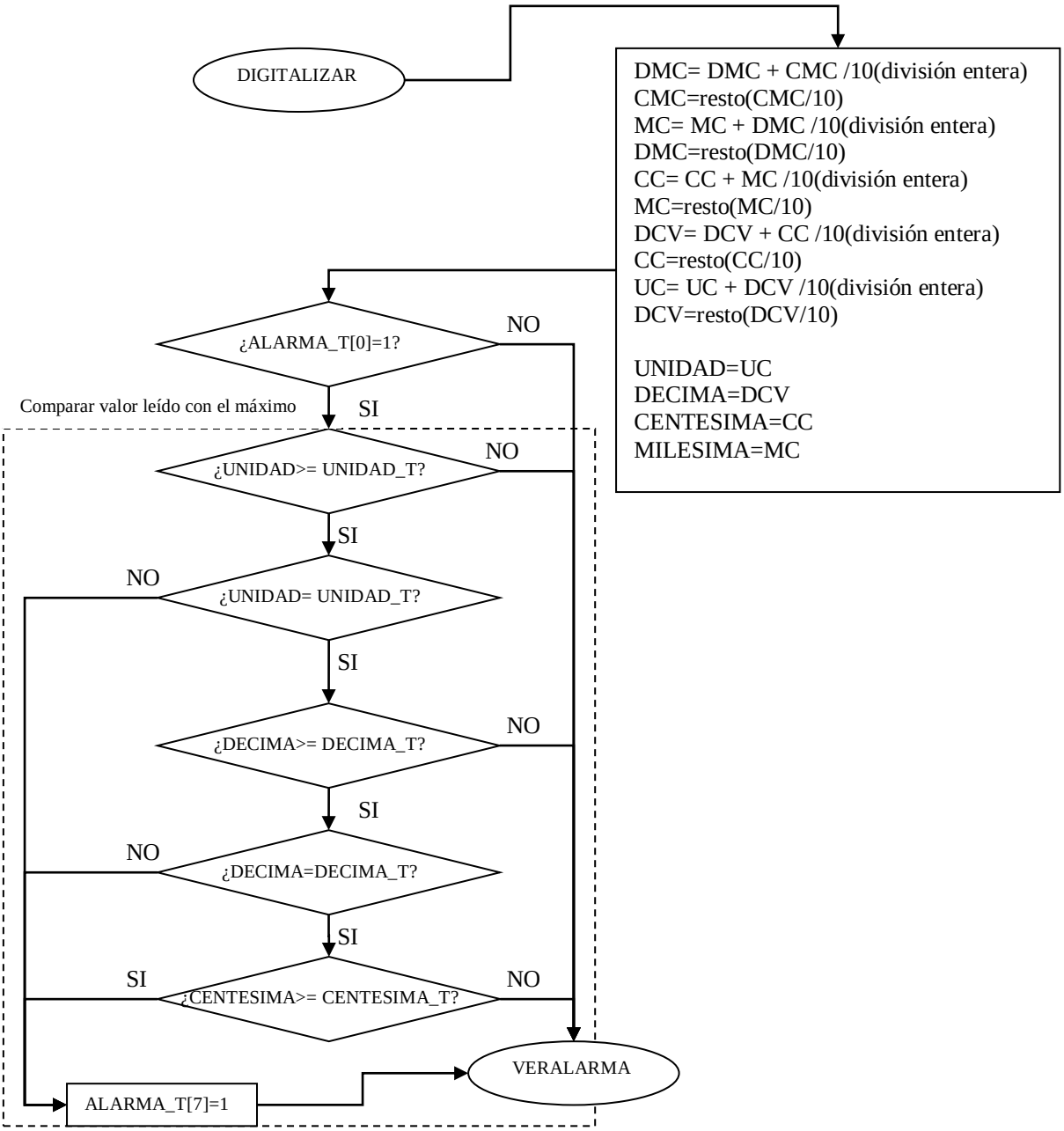
De cualquier manera ninguno de los instrumentos utilizados ha sido calibrado. Ninguno está certificado. Con estas mediciones se puede estimar un error máximo de -3ºc.

Nota:*Véase foto 12 en página 44*

Diagrama de flujo del conversor



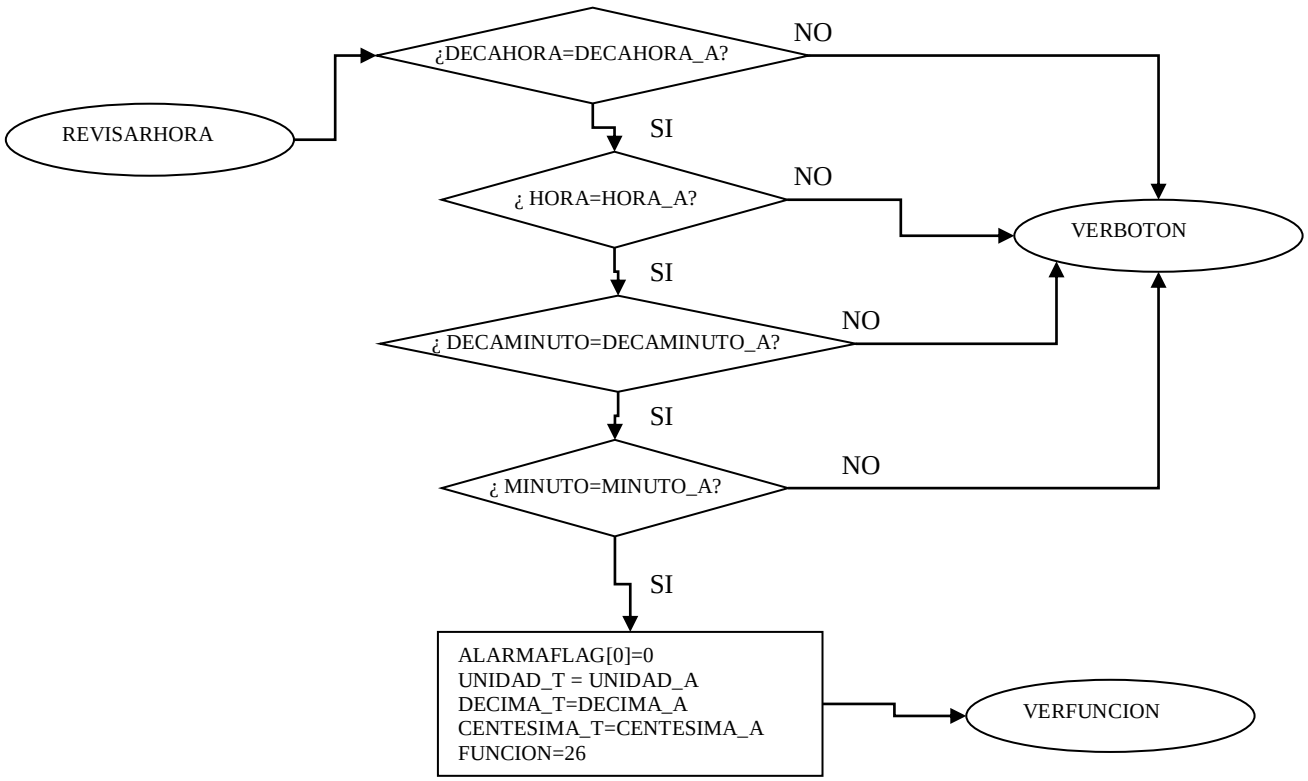

PROMEDIARAD
Promediar entrada
NUM2= .10
NUMDH=ADPROMH
NUMDL=ADPROML
Call DIVD_16
NO
¿ADPROML<DIVDRTAL?
SI
Decrementar ADPROMH
ADPROML=ADPROML-DIVDRTAL
ADPROMH=ADPROMH-DIVDRTAH
¿ GONE/DONE=0?
SI
NO
NUMDH=ADRESH
NUMDL=(ADRESL) AND (B'11000000')
Call DIVD_16
ADPROML=ADPROML+DIVDRTAL
NO
¿hay CARRIER?
SI
Incrementar ADPROMH
ADPROMH=ADPROMH+DIVDRTAH
UC=0
DCV=0
CC=0
MC=0
DMC=0
CMC=0
¿ ADPROMH[7]=1?
UC=UC+2
DCV=DCV+5
¿ ADPROMH[6]=1?
UC=UC+1
DCV=DCV+2
CC=CC+5
¿ ADPROMH[5]=1?
DCV=DCV+6
CC=CC+2
MC=MC+54
¿ ADPROMH[4]=1?
DCV=DCV+3
CC=CC+1
MC=MC+2
DMC=DMC+5
¿ ADPROMH[3]=1?
DCV=DCV+1
CC=CC+5
MC=MC+6
DMC=DMC+2
CMC=CMC+5
¿ ADPROMH[2]=1?
CC=CC+7
MC=MC+8
DMC=DMC+1
CMC=CMC+2
¿ ADPROMH[1]=1?
CC=CC+3
MC=MC+9
DMC=DMC+0
CMC=CMC+6
¿ ADPROMH[0]=1?
CC=CC+1
MC=MC+9
DMC=DMC+5
CMC=CMC+3
¿ ADPROML[7]=1?
MC=MC+9
DMC=DMC+7
CMC=CMC+6
¿ ADPROML[6]=1?
MC=MC+4
DMC=DMC+8
CMC=CMC+8
DIGITALIZAR



Revisar alarma

Si la alarma está activada una de las misiones del bucle principal es comparar la hora con la hora de alarma. La alarma está activada si el bit 0 de ALARMAFLAG es 1 Entonces simplemente se hace lo siguiente:

Diagrama de flujo de Revisar Alarma



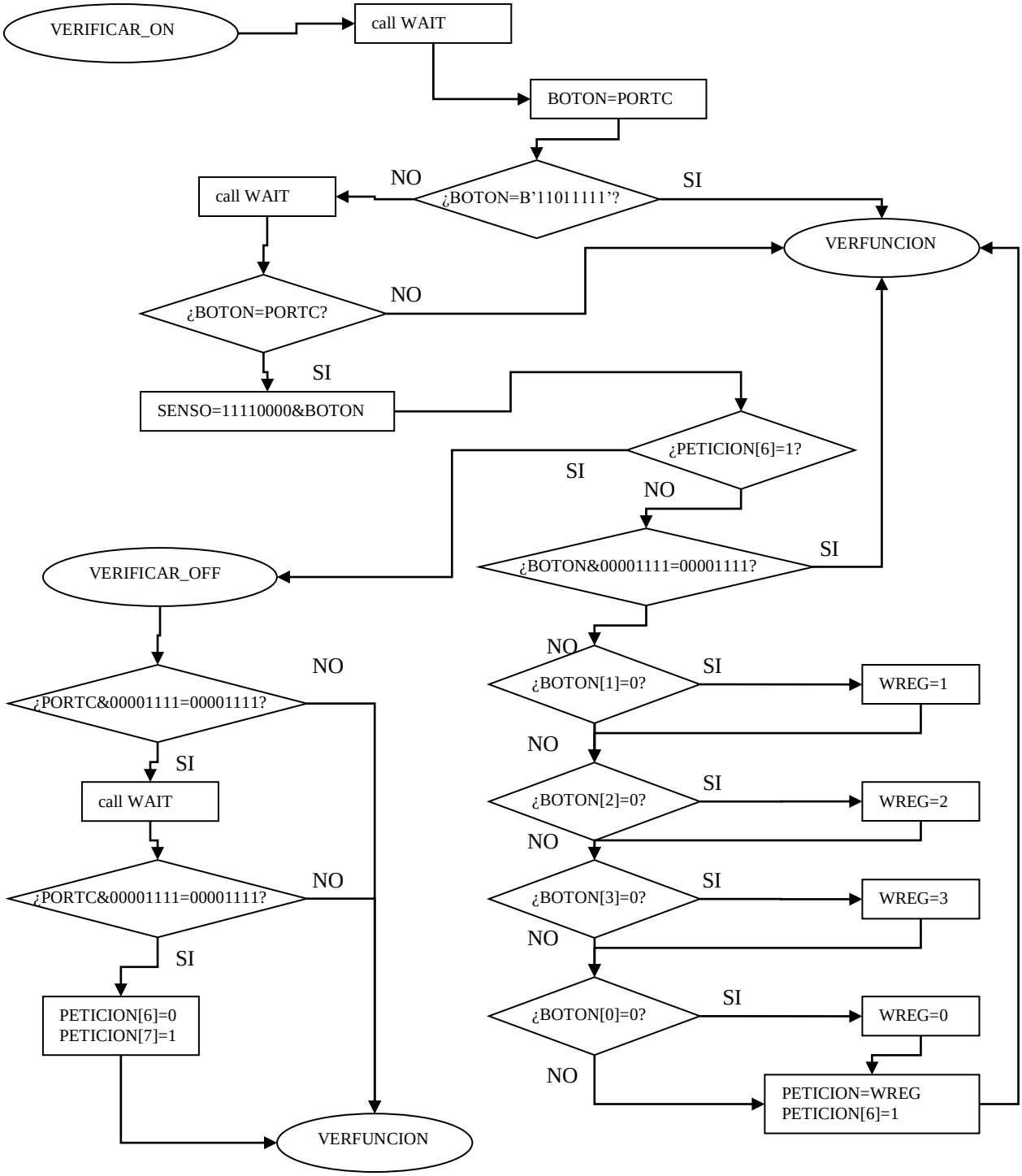
Botones y Sensores:

Los 4 botones y 4 sensores han sido conectados al puerto C.
Para evitar el efecto rebote se lo hace mediante software. Los botones y sensores son chequeados en el programa principal, en el bucle principal.
El bit 7 corresponde al fotodiodo, el bit 6 al sensor de boya, 5 al sensor de válvula cerrada y el 4 al sensor de válvula abierta. El bit 3 es "ENTRAR", el 2 es "ARRIBA", el 1 es "ABAJO" y el 0 es "SALIR".

Los sensores cargan una variable llamada SENSO que es requerida cuando se realiza el proceso.
Los botones cargan a la variable PETICION. El bit 7 de PETICION esta activado cuando se ha presionado un botón recientemente y se desactiva cuando una función del programa atiende a la petición.
Los botones necesitan ser soltados para que el programa los tome en cuenta

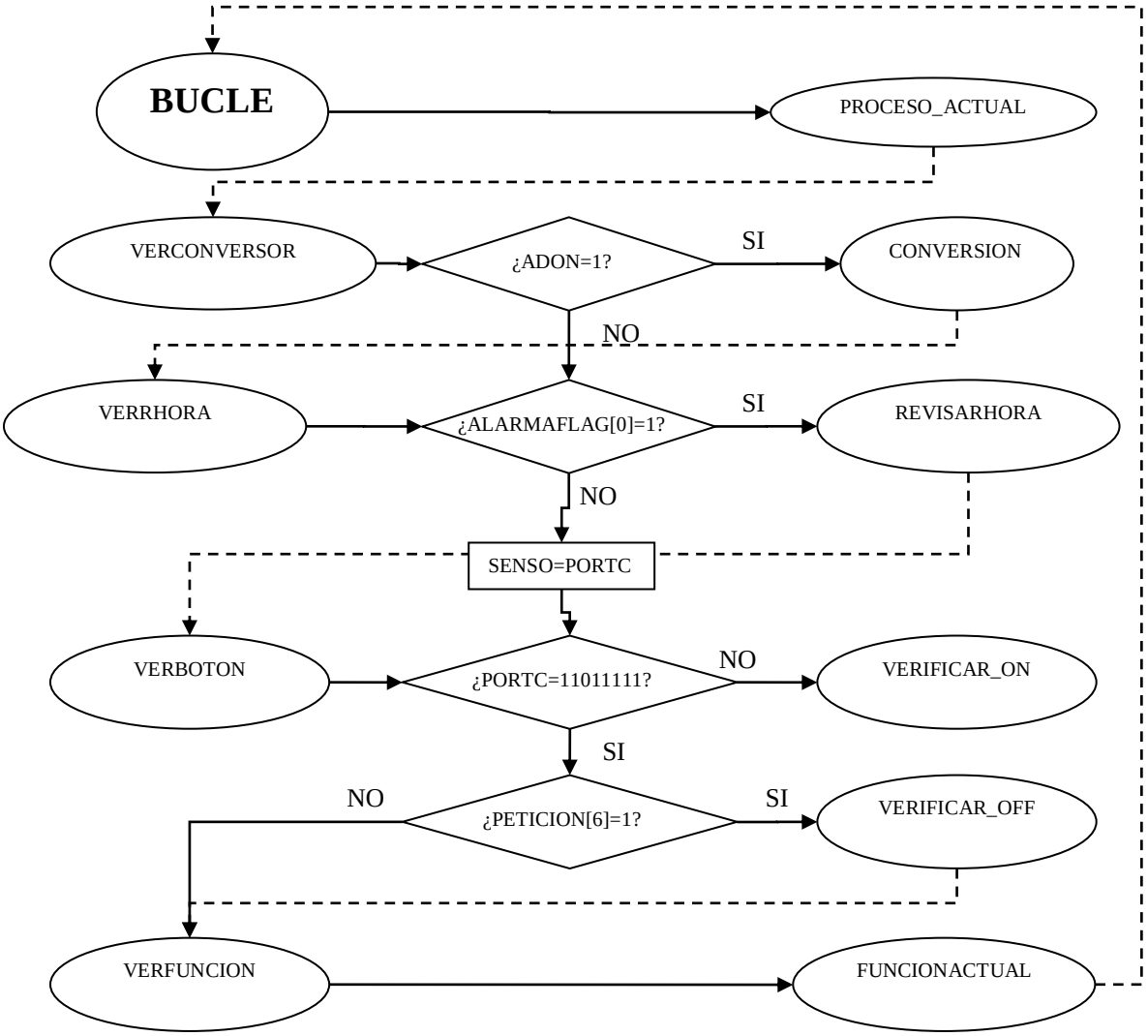
Se ha creado una función llamada WAIT que no hace otra cosa que perder tiempo en un retardo.
El puerto C es configurado como entrada con el registro TRISC=0xFF

Diagrama de flujo de botones y sensores (verificar on)



EL BUCLE PRINCIPAL:

Este bucle siempre esta activado y es el encargado de llamar a realizar las tareas activadas.



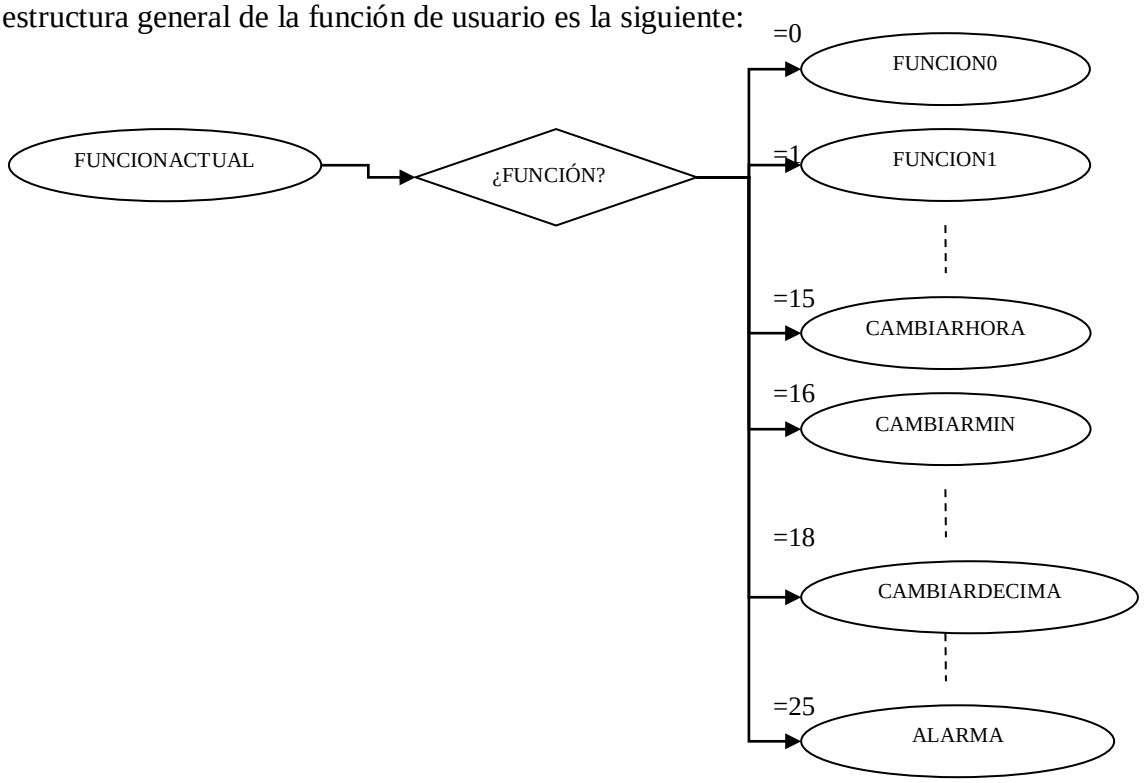
Las líneas punteadas indican a donde retornará una subfunción luego de haber realizado su tarea.
El estado base del puerto C es b'11011111' (donde el 0 significa el sensor de válvula cerrada). Por lo tanto para optimizar el funcionamiento general, no se verifica el encendido de los botones o sensores si está en ese estado ya que la verificación tiene retardos inútiles.
Tampoco se hacen conversiones si no son necesarias y no se verifica la hora si la alarma no está activada.

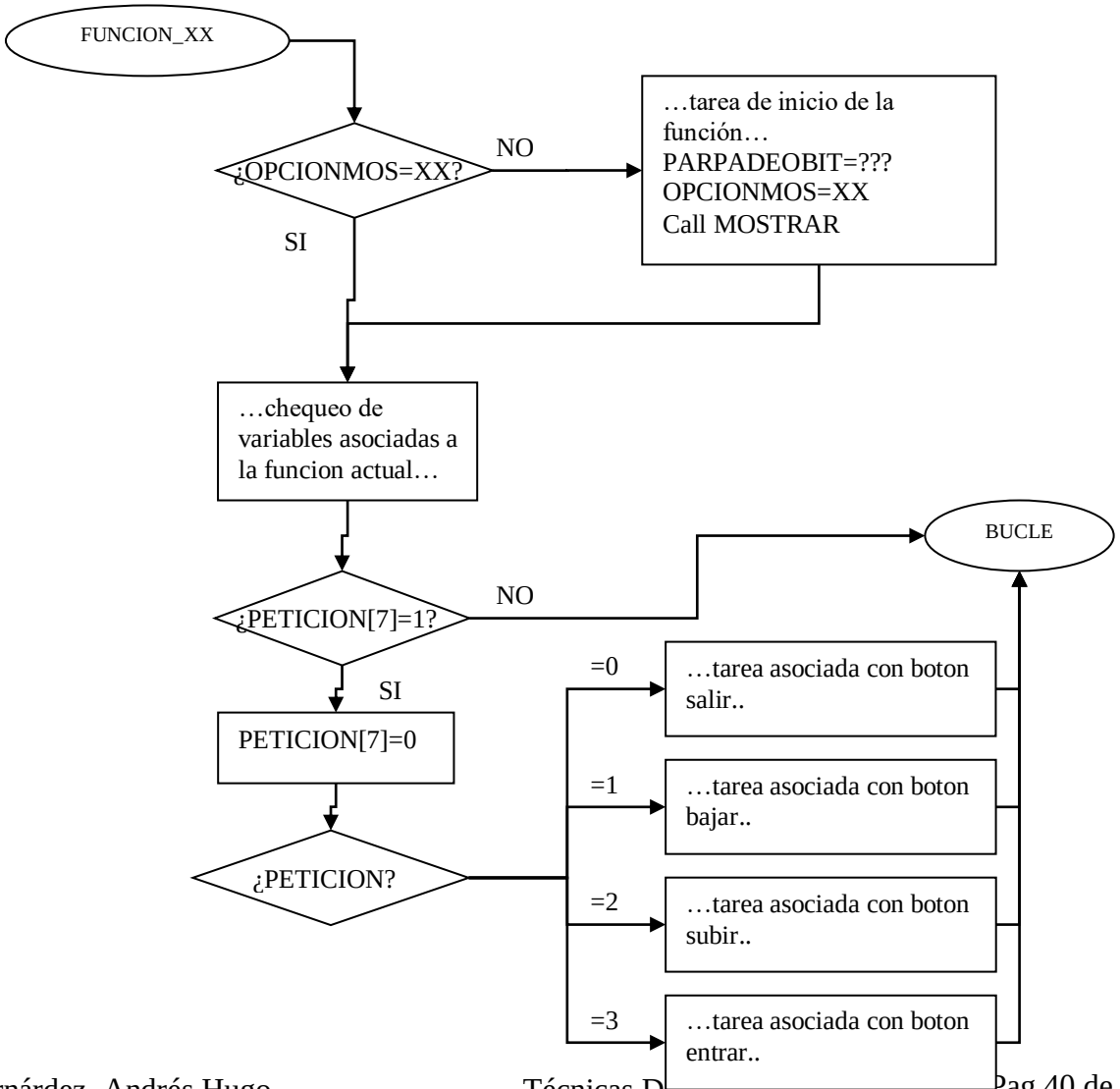
FUNCION ACTUAL:

Por último queda explicar como funcionan las que he denominado "funciones de usuario"o "funciones para usuario". A traves de las funciones de usuario el usuario puede acceder a las distintas pantallas y realizar las tareas programadas utilizando los botones.

Diagrama de flujo de Función Para Usuario

La estructura general de la función de usuario es la siguiente:





En la primera etapa de una función de usuario se chequea si OPCIONMOS es lo que se tiene que mostrar. En caso que no sea significa que es el primera ciclo en que se entra a la función por lo tanto hay que configurar los vectores a mostrar, si hay que hacer parpadear algún display y si hay que encender algún proceso y/o el conversor.

Luego en caso que se esté realizando un proceso, se chequea si el proceso ha cambiado o en caso de la función ALARMATENSION se chequea si ALARMA_T[7]=1, para realizar una tarea.

Las tareas comunes que se pueden realizar con los botones son:

- Cambiar de función de usuario mediante el cambio del valor del registro FUNCION
- Cambiar el valor de algún registro como: HORA, MINUTO, CENTESIMA_T, etc.
- Realizar alguna parte del proceso mediante el cambio del valor del registro PROCESO

La función FUNCION3(ver tensión/temperatura) y ALARMATENSION activan el conversor. Por lo tanto al salir de estas funciones, deben asegurar apagarlo.

Fotos de la máquina:

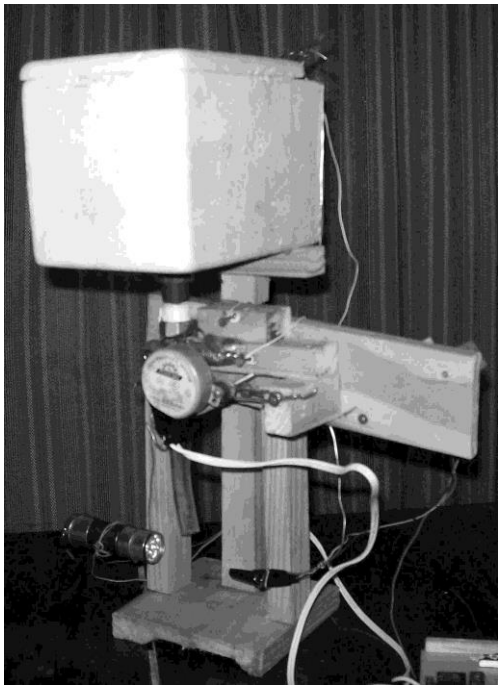
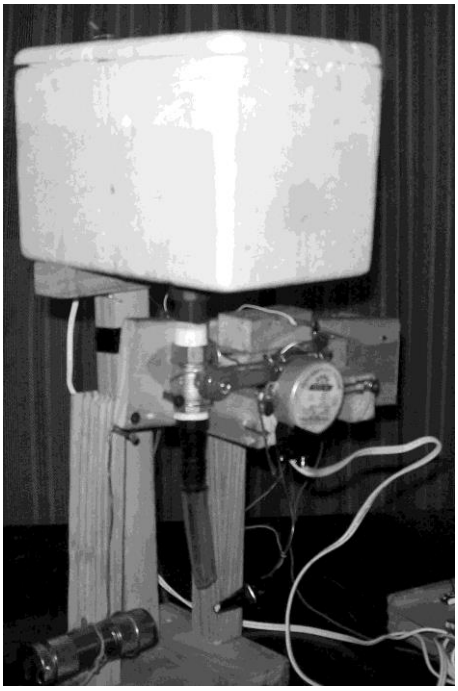



Foto1 y foto 2: Aquí se puede apreciar el tanque, el motor con la válvula y la linterna con el fotodiodo(recubierto)

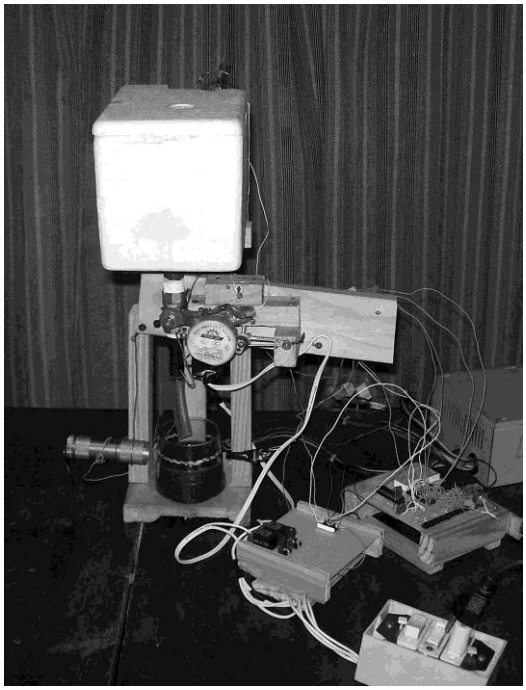
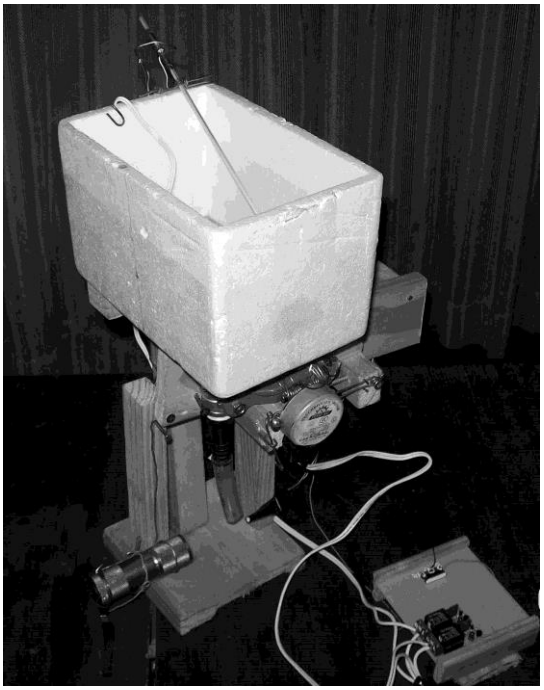
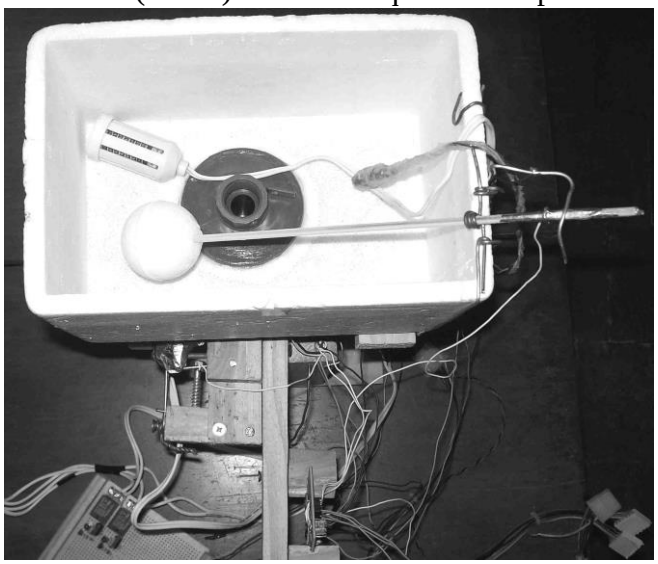



Foto3 y Foto4:
En la foto de la izquierda (foto 3) está la máquina completa con una tasa. En la foto de la derecha(foto 4) se ve el tánque sin la tapa.



.

Foto5: Aquí se puede ver el interior del tanque: Adentro se ve el calentador, el sensor lm35 recubierto de fastix y la boya que activa un sensor

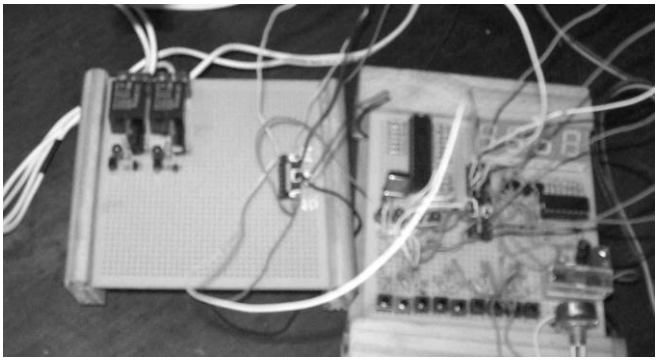


Foto6: Aquí se ven los 2 relés a la izquierda y el pic, displays y botones a la derecha

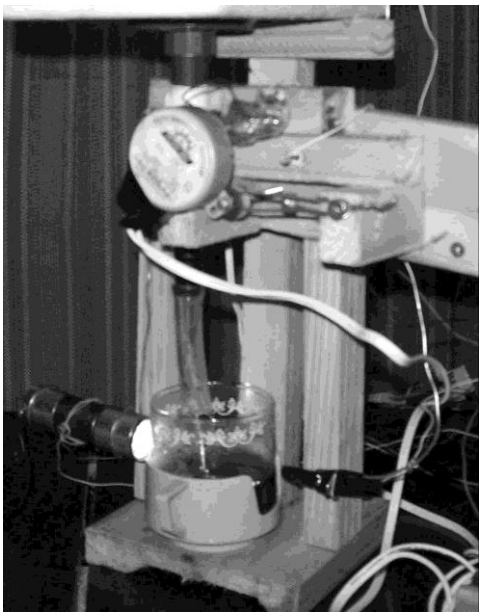
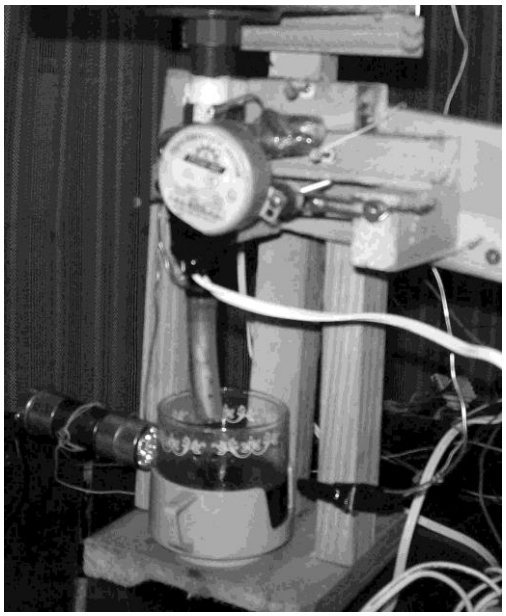



Foto 7 y foto 8:
A la izquierda se ve como se sirve el té a la derecha cuando terminó de servirlo. Nótese como se abre y cierra la válvula. Cuando esta sirviendo la linterna está prendida.

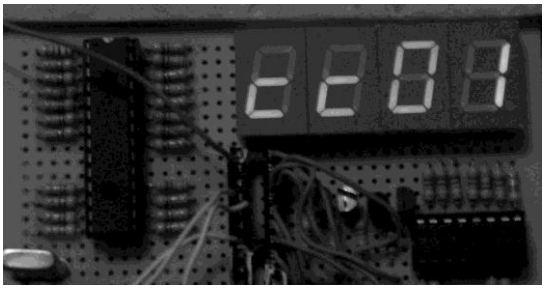



Foto 9 y foto 10
Foto 9(izquierda): Aquí se ve la pantalla de displays en el menú principal
Foto 10: La pantalla en el momento que se cambia la unidad de temperatura final de alarma (función 2)

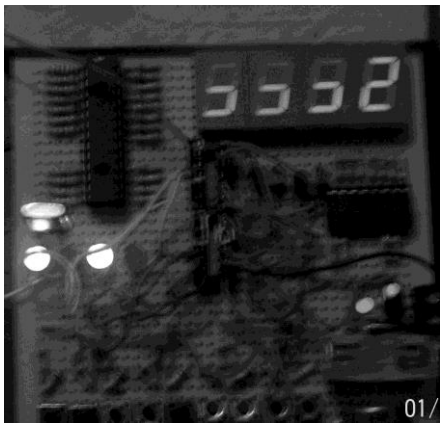


Foto 11: Pantalla que se ve cuando se abre la válvula en la función 5. Los leds encendidos simulan el relé del motor y la linterna

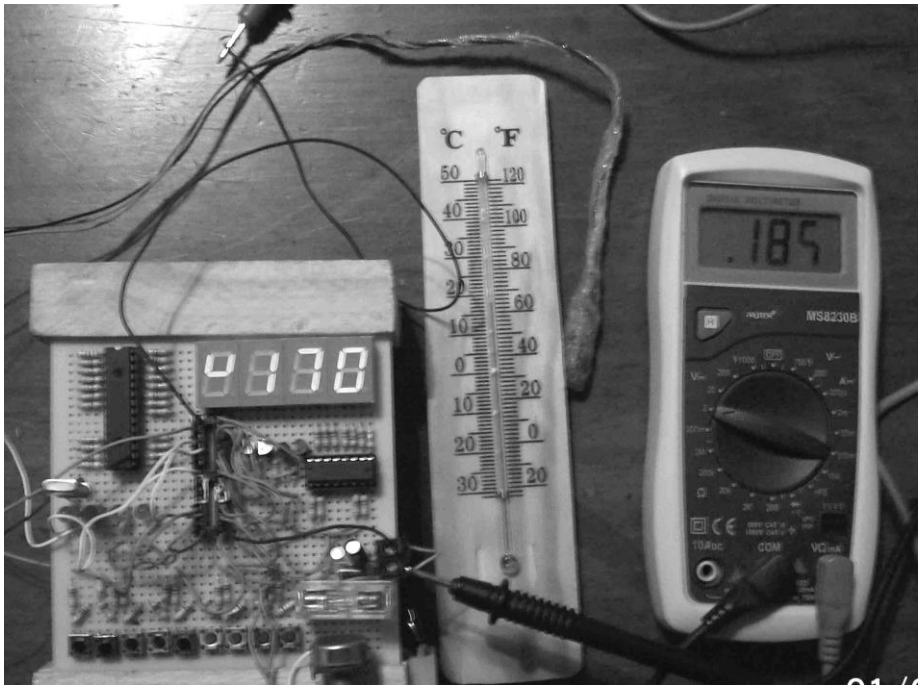


Foto 12:
Medición de la temperatura ambiente: En los displays se aprecia 17ºC esto es porque se está midiendo 170mV. La misma tensión medida con el multímetro muestra 185mV. El termómetro indica 18ºC. Entre el termómetro y el multímetro está el sensor LM35 (recubierto y fuera del tanque).

Código fuente en assembler(programado con MPLAB)

Los bits de

```
;
;                   Proyecto= Servir-T
;                   TÉCNICA DIGITALES II
;                   FRSN UTN
;                   2011
;
;               PROGRAMADOR: BERNÁRDEZ ANDRÉS HUGO



            list p=16F873A
            include "p16f873a.inc"
            __CONFIG 0x3F39  ; OSCILADOR XT, SIN WDT


CENTISEGUNDO    EQU 0x20; VARIABLES DE TIEMPO
SEGUNDO         EQU 0x21
DECASEGUNDO     EQU 0x22
MINUTO          EQU 0X23
DECAMINUTO      EQU 0X24
HORA            EQU 0X25
DECAHORA        EQU 0X26
DIA             EQU 0X27
MINUTO_A        EQU 0X28   ;VARIABLES DE ALARMA
DECAMINUTO_A    EQU 0X29
HORA_A          EQU 0X2A
DECAHORA_A      EQU 0X2B
ALARMAFLAG      EQU 0X2C ; INDICA CUANDO ESTA ACTIVADA LA ALARMA
BITFLAGS        EQU 0X2D ; BITS DE PROPOSITOS GENERALES, 7:AUXILIAR PAR FREC.DE CONVERSION
```

```
MILESIMA       EQU 0X2F; VARIABLES DE MEDICION
CENTESIMA      EQU 0X30;
DESIMA         EQU 0X31
UNIDAD         EQU 0X32
DECENA         EQU 0X33

OPCIONMOS      EQU 0X35; VARIABLES DE MOSTRAR (4 DISPLAYS 7SEG CON BCD Y REFRESCO CON SOFTWARE)
CONTADORMOSTRAR EQU 0X36;  CONTADOR EN FUNCION MOSTRAR
DISPLAYBIT     EQU 0X37;  ESTE BIT ACTIVA UN DISPLAY POR VEZ EN EL PUERTO B
PUNTERO_0      EQU 0X38  ; VALORES QUE APUNTAN LO QUE DEBEN MOSTRAR LOS DISPLAY
PUNTERO_1      EQU 0X39
PUNTERO_2      EQU 0X3A
PUNTERO_3      EQU 0X3B
RETARDOM       EQU 0X3C; RETARDO QUE SE USA CUANDO SE PRENDE UN DISPLAY
PCLATHAUX1     EQU 0X3D; REGISTRO AUXILIAR DE PCLATHAUX USADO EN FUNCION MOSTRAR


RETARDO        EQU 0x40; VARIABLE DE RETARDO SIN TIMER


                ;VARIABLES DE INTERFAZ
BOTON          EQU 0X41; VARIABLE DE PRESION DE BOTON(GUARDA QUE SE APRETO)
PETICION       EQU 0X42; VARIABLE QUE INTERACTUA CON LA FUNCION ACTUAL CUANDO EL USUARIO APRIETA UN BOTON
SENSO          EQU 0X43; VARIABLE DE DETECCION DE ESTADO DE LOS SENSORES


MENU           EQU 0X44; INDICA QUE FUNCION ESTA SELECCIONADA PARA SER USADA CON ENTER
DECAMENU       EQU 0X45; //     //  //
FUNCION        EQU 0X46;  INDICA CUAL ES LA FUNCION ACTUAL QUE SE ESTA USANDO

PROCESO        EQU 0X48;  PROCESO ACTUAL QUE SE REALIZA
CRONOPROC      EQU 0X49;  AVISA QUE SE HA COMENZADO A CRONOMETRAR UN PROCESO
CRONO          EQU 0X4A; CRONOMETRA EL TIEMPO DE ALGUN PROCESO (PRECISION DE +0.25 SEGUNDOS)
                ;CADA 4 UNIDADES ES UN SEGUNDO. CUENTA MAXIMA 63.75 SEGUNDOS
ERRORNUM       EQU 0X4B;  DESIGNA CON UN NUMERO A UN ERROR QUE HAYA OCURRIDO
                ;( POR DESBORDAMIENTO DE TIEMPO ESTIMADO PARA UN PROCESO)

WREGAUX        EQU 0X50; VARIABLES AUXILIARES DE INTERRUPCION
STATUSAUX      EQU 0X51
PCLATHAUX      EQU 0X52

                ; VARIABLES PARA EL ANÁLISIS DE LA CONVERSION
UC             EQU 0X53;
DCV            EQU 0X54;
CC             EQU 0X55;
MC             EQU 0X56;
DMC            EQU 0X57
CMC            EQU 0X58
NUM1           EQU 0X59;      PARA DIVIDIR
NUM2           EQU 0X5A
ARIT_AUX       EQU 0X5B


LETRAMENU      EQU 0X5C  ;VARIABLES QUE SOLO SIRVE PARA MOSTRAR ALGO EN EL DISPLAY
LETRAIZQ       EQU 0X5D
LETRADER       EQU 0X5E
PARPADEOBIT    EQU 0X5F  ; //BIT7:ESTADO DE PARPADEO//2:PARPADEOIZQ//1:PARPADEODER//0:PARPADEO ON

UNIDAD_T       EQU 0X60   ; VARIABLES DE TENSION FINAL
DESIMA_T       EQU 0X61
CENTESIMA_T    EQU 0X62
ALARMA_T       EQU 0X63   ; ALARMA DE TENSION
UNIDAD_A       EQU 0X64   ; VARIABLES DE TENSIÓN FINAL DE ALARMA
DESIMA_A       EQU 0X65
CENTESIMA_A    EQU 0X66
UNIDAD_C       EQU 0X67  ; CONTROL DEL "ERROR_00"( NO SE DETECTA  VARIACION POR MAS DE UN MINUTO)
DESIMA_C       EQU 0X68
CENTESIMA_C    EQU 0X69
CONTADOR_C     EQU 0X70



ADPROMH        EQU 0X71
ADPROML        EQU 0X72

NUMDH          EQU 0X73  ; VARIABLES PARA FUNCION "DIVDOBLE"
NUMDL          EQU 0X74
DIVDRESTOH     EQU 0X75
DIVDRTAH       EQU 0X76
DIVDRTAL       EQU 0X77
```

```
;*****************************************************************

        org 0X00
        NOP
        GOTO INICIO;  EMPEZAR EL PROGRAMA
        GOTO INICIO;  REDUNDANCIA
;_____________________________________________________
;-------------------------------------------------¬|
        ORG 0X04   ;INTERRUPCION     ___<|

        MOVWF WREGAUX   ; GUARDAR REGISTROS IMPORTANTES: PCLATH STATUS Y WREG
        MOVF STATUS,0
        MOVWF STATUSAUX
        MOVF PCLATH,0
        MOVWF PCLATHAUX


        BTFSS INTCON,2 ; ¿QUE INTERRUPCION? ¿POR TIMER0?
        GOTO TIMER1INT ; NO
        GOTO REFRESCAR ; SI

VOLVERREFRES:   CLRF TMR0;
        BCF INTCON,2; HABILITAR TIMER0
        BTFSS  PIR1,2
        GOTO RETORNARINT


TIMER1INT:      BCF PIR1,2;HABILITAR BANDERA COMPARADOR TIMER1


;-----------------------RUTINA DE INCREMENTAR TIEMPO Y LLENAR VARIABLES CON VALOR CORRESPONDIENTE.....
;----------------

        INCF CENTISEGUNDO  ; INCREMENTAR UN CENTISEGUNDO Y ACOMODAR HORA...

        MOVLW .25
        SUBWF CENTISEGUNDO,0  ; ¿ES CENTISEGUNDO=.25?
        BTFSC STATUS,Z
        INCF CRONO ;  SI: INCREMENTAR CRONO  (CRONO SIRVE PARA CRONOMETRAR CON ERROR MAX DE 0.25 SEGUNDOS,
                ; 4 UNIDADES DE CRONO ES UN SEGUNDO)

        MOVLW .50
        SUBWF CENTISEGUNDO,0  ; ¿ES CENTISEGUNDO=.50?
        BTFSC STATUS,Z
        INCF CRONO ;

        MOVLW .75
        SUBWF CENTISEGUNDO,0  ; ¿ES CENTISEGUNDO=.75?
        BTFSC STATUS,Z
        INCF CRONO ;


SIGUERELOJ:     MOVLW .100
        SUBWF CENTISEGUNDO,0  ; ¿ES CENISEGUNDO >=.100?
        BTFSS STATUS,C
        GOTO RETORNARINT      ; NO: IR A RETORNARINT

        CLRF CENTISEGUNDO     ;SI: CENTISEGUNDO=0

        INCF CRONO            ; INCREMENTAR CRONO

        INCF SEGUNDO          ; INCREMENTAR SEGUNDO
        MOVLW 0X0A            ; ¿ES SEGUNDO >=0X0A?
        SUBWF SEGUNDO,0
        BTFSS STATUS,C
        GOTO RETORNARINT      ; NO: IR A RETORNARINT

        CLRF SEGUNDO          ;SI: SEGUNDO=0, INCREMENTAR DECASEGUNDO
        INCF DECASEGUNDO
        MOVLW 0X06
        SUBWF DECASEGUNDO,0   ;¿ES DECASEGUNDO>= 6?
        BTFSS STATUS,C
        GOTO RETORNARINT      ;NO: IR A RETORNARINT

        CLRF DECASEGUNDO      ;SI: DECASEGUNDO=0, INCREMENTAR MINUTO
        INCF MINUTO
        MOVLW 0X0A            ;¿ES MINUTO>=10?
        SUBWF MINUTO,0
        BTFSS STATUS,C
        GOTO RETORNARINT      ;NO: IR A RETORNARINT

        CLRF MINUTO
        INCF DECAMINUTO       ;SI: MINUTO=0, INCREMENTAR DECAMINUTO
        MOVLW 0X06
```

```
        SUBWF DECAMINUTO,0      ; ¿ES DECAMINUTO>= 6?
        BTFSS STATUS,C
        GOTO RETORNARINT        ;NO: IR A RETORNARINT

        CLRF DECAMINUTO         ;SI...

            ;-------------------LLENAR LAS HORAS TIENE MAS COMPLICACION...
        INCF HORA               ; INCREMENTAR HORA...


        MOVLW 0X02              ; ¿ DECAHORA>=2 ?
        SUBWF DECAHORA,0
        BTFSS STATUS,C
        GOTO HORA1              ;NO: IR A HORA1

        MOVLW 0X04              ; SI:¿ HORA>=4?
        SUBWF HORA,0
        BTFSS STATUS,C
        GOTO RETORNARINT            ;NO: IR A RETORNARINT

        CLRF HORA               ;ENTONCE DECAHORA,HORA= 00HS
        CLRF DECAHORA
        GOTO DIA1               ;IR A DIA

HORA1:      MOVLW 0X0A              ; VERIFICAR SI HORA= 0X0A??
        SUBWF HORA,0
        BTFSS STATUS,C
        GOTO RETORNARINT            ;NO: IR A RETORNARINT

        CLRF HORA               ;SI: HORA=00// DECAHORA++
        INCF DECAHORA
        GOTO RETORNARINT

            ;-----------------------------FIN LLENAR HORAS
DIA1:
        INCF DIA                ;¿DIA>=0X0A?
        MOVLW 0X0A
        SUBWF DIA,0
        BTFSS STATUS,C
        GOTO RETORNARINT        ;NO: IR A RETORNARINT

        CLRF DIA                ;SI: DIA=0
        GOTO RETORNARINT; IR A RETORNARINT
;-----------------
;---------------------------------------FIN DE RUTINA DE LLENAR TIEMPO


RETORNARINT:    MOVF STATUSAUX,0; RECUPERAR STATUS
        MOVWF STATUS;
        MOVF PCLATHAUX,0
        MOVWF PCLATH;  RECUPERAR REGISTRO PCLATH

        MOVF WREGAUX,0; RECUPERAR REGISTRO W
        RETFIE

;---------------------FIN INTERRUPCION    ||
;_____________________________________________||


;-------------------FUNCION REFRESCAR.....

REFRESCAR:
        MOVLW 0X08
        MOVWF CONTADORMOSTRAR   ; CONTADORMOSTRAR=8

        MOVLW B'10000000'
        MOVWF DISPLAYBIT ;      ;DISPLAYBIT=B'10000000'
        CLRF PORTB;             PORTB=0

        CLRF PCLATH;            ;PCLATH=0

        BTFSC PARPADEOBIT,0 ; ¿¿ESTA ACTIVADO EL PARPADEO??
        GOTO PARPADEO;          SI ESTA ACITVADO IR A PARPADEO
CICLOMOSTRAR:
        MOVF CONTADORMOSTRAR,0 ; NO

        ADDWF PCL,1
        NOP             ; CONTADORMOSTRAR=??
        NOP
        MOVF PUNTERO_0,0    ; =2-> WREG=PUNTERO_0
        GOTO CARGARPORTB
        MOVF PUNTERO_1,0    ; =4-> WREG=PUNTERO_1
        GOTO CARGARPORTB
        MOVF PUNTERO_2,0    ; =6-> WREG=PUNTERO_2
        GOTO CARGARPORTB
```

```
        MOVF PUNTERO_3,0    ; =8-> WREG=PUNTERO_3

                ;DIRECCIONAMIENTO INDIRECTO!!!
CARGARPORTB:    MOVWF FSR  ; DIRECCION DONDE ESTA EL DATO A LEER
        MOVF INDF,0; LEER DATO AL QUE APUNTA FSR


        MOVWF PORTB;  PORTB DEBE SER UN NÚMERO DEL 0X0 AL 0xF
        MOVF DISPLAYBIT,0; EJEMPLO SI DISPLAYBIT=00010000=>W=00010000
        IORWF PORTB,1 ; EJEMPLO PORTB=(00010000 OR 0000XXXX)=0001XXXX

        MOVLW 0XFF;  RETARDO PRENDIDO
        MOVWF RETARDOM

ESPERA1:        NOP
        NOP
        NOP;          SON UN TOTAL DE RETARDO1*4 CICLOS   NOP=1 CICLO GOTO=2CICLO DECFSZ=1 CICLO CUANDO NO SALTA
        DECFSZ RETARDOM ;; 4DISPLAYS QUE ESTARAN 256*4 CICLOS SON UN TOTAL DE 6144 USEG QUE ES MENOR
                ; A LOS 10000 DEL TMR1 TODAVIA HAY TIEMPO ANTES DE SER NECESARIO ACTIVAR
                ;LA INTERRUPCION X TMR1
        GOTO ESPERA1

        CLRF PORTB ;SE APAGA EL PORTB
        NOP
        NOP       ; SE ESPERA UN POCO PARA QUE SE APAGUE EL DISPLAY
        NOP      ;APROVECHANDO LAS INSTRUCCIONES DE ESTA FUNCION
        NOP

        RRF DISPLAYBIT,1;  SE ROTA EL BIT A LA DER. DE DISPLAYBIT

        DECF CONTADORMOSTRAR   ; DECREMENTAR CONTADORMOSTRAR
        DECFSZ CONTADORMOSTRAR ; VOLVER A DECREMENTAR  ¿ES CERO?
        GOTO CICLOMOSTRAR      ; AUN NO
        GOTO VOLVERREFRES   ; SI
;-----------------------------------FIN FUNCION RESFRESCAR


PARPADEO:
        BTFSC CENTISEGUNDO,6; ESTE BIT SE PRENDE A LOS 640 MSEG Y SE APAGA CUANDO SE CUMPLE EL SEGUNDO
        GOTO PARPADEAR      ; SI BIT6 DE CENTISEGUNDO ESTA PRENDIDO IR A PARPADEAR

        BTFSS PARPADEOBIT,7  ; ¿¿BIT 7=ON??  BIT 7 DE PARPADEO BIT SE PRENDE CUANDO EMPIEZA EL PARPADEO
                             ;Y SE APAGA CUANDO TERMINA
        GOTO CICLOMOSTRAR       ;NO: VOLVER...

                ; FIN DE PARPADEO
        CALL MOSTRAR;      ; LA FUNCIÓN MOSTRAR ACTUALIZA LOS VECTORES SEGÚN OPCIONMOS
                ; EXCEPTO FUNCION0 QUE NO PARPADEA ASI QUE NO HAY PROBLEMA
        BCF PARPADEOBIT,7    ; LIMPIAR BIT 7 DE PARPADEOBIT: FIN DE PARPADEO
        GOTO CICLOMOSTRAR    ; VOLVER

PARPADEAR:      BTFSC PARPADEOBIT,7 ;  BIT7=OFF
        GOTO CICLOMOSTRAR   ; SI ESTA PRENDIDO ES PORQUE YA EMPEZO EL PARPADEO ENTONCES SE VUELVE

        BSF PARPADEOBIT,7  ; SI ESTA APAGADO HAY QUE PRENDERLO E INICIAR EL PARPADEO

PARPADEO_DER:   BTFSS PARPADEOBIT,1; SI ESTA PRENDIDO EL BIT 1 PARPADEAR DERECHO: PARPADEO DERECHO SIGNIFICA
                ; QUE LOS DOS DISPLAYS DE LA DERECHA INTERCAMBIA CICLICAMENTE SU VALOR POR OTRO
        GOTO  PARPADEO_IZQ

        MOVLW LETRADER   ;LETRADER (Y LETRAIZQ)DEBERÍA SER UN NUMERO DEL 0X0 AL 0XF
        MOVWF PUNTERO_0
        MOVLW LETRADER
        MOVWF PUNTERO_1


PARPADEO_IZQ:   BTFSS PARPADEOBIT,2; SI ESTA PRENDIDO EL BIT 2 PARPADEAR IZQUIERDO: SIGNIFICA QUE LOS DOS DISPLAYS
                ; DE LA IZQUIERDA INTERCAMBIA CICLICAMENTE SU VALOR POR OTRO
        GOTO  CICLOMOSTRAR

        MOVLW LETRAIZQ
        MOVWF PUNTERO_2  ; NÓTESE QUE SI SE PRENDE EL BIT 1 Y 2 PARPADEAN LOS 4 DISPLAYS AL MISMO TIEMPO
        MOVLW LETRAIZQ
        MOVWF PUNTERO_3

        GOTO CICLOMOSTRAR




;*****************************************************************************************
;*****************************************************************************************
;
;>>>>>>>>>>>>>>INICIO DEL PROGRAMA AQUI!!!<<<<<<<<<<<<<<<<<<<<<<<<_____________________________
```

```
;-------------------------------------------------------------------------------------------

INICIO:     CLRF INTCON; DESACTIVAR INTERRUPCIONES

        BCF STATUS, IRP; LOS DATOS SUELEN ESTAR EN BANCO OO Ó O1 POR ESO SE ELIGE IRP=0 PARA DIRECCIONAMIENTO INDIRECTO
        BCF STATUS, RP1;
        BSF STATUS, RP0; BANCO 01
                            ;---------CONFIGURACION PORTA EN BANCO 1
        MOVLW B'00010011'                   ;---CONVERSOR A/D
        MOVWF TRISA;  PORTA,5; PORTA,2 Y PORTA,3 COMO SALIDAS EL RESTO ENTRADAS PORTA,0 SERA ENTRADA ANALÓGICA
        MOVLW B'00001110';
        MOVWF ADCON1; ADCON1:LEFT JUSTIFIED// RA0 ES ANALOGICO EL RESTO DIGITAL// RELOJ FOSC/16


        CLRF TRISB; PORTB COMO SALIDA       ;----------CONFIGURACION PORTB Y PORTC EN BANCO 1
        MOVLW 0XFF;
        MOVWF TRISC; PORTC COMO ENTRADA


                            ;-----------CONFIGURACION TIMERS EN BANCO 1
        BCF OPTION_REG,5;  TIMER0 ACTIVADO CON FUENTE INTERNA SIN PRESCALER
        CLRF PIE1;
        BSF PIE1,2; HABILITAR INTERRUPCION POR CCP DE TIMER1




        BCF STATUS, RP0; BANCO 00

                ;---------CONFIGURACION PORTA EN BANCO 0
        MOVLW B'01000000'  ; ADCON0: 8TOSC--- CHANNEL 0--- SIN OPERACION--- CONVERSOR MODO: APAGADO!
        MOVWF ADCON0;

                ;----------CONFIGURACION DE PORTB EN BANCO 0
        CLRF PORTB; SALIDA PORTB=0



        ; CONFIGURACION DEL CCP PARA TIMER1

        MOVLW B'00001011'
        MOVWF CCP1CON;  ; MODO SPECIAL EVENT TRIGGER:CUANDO TMR1 LLEGA A CCPR1 RESETEA E INTERRUMPE
        MOVLW 0X10;     0X2710 CICLOS=10000US=0.01SEG
        MOVWF CCPR1L;    CARGAR CCPR1 PARA QUE INTERRUMPA CADA 0.01 SEGUNDO
        MOVLW 0X27;
        MOVWF CCPR1H;
        CLRF T1CON;
        BSF T1CON,0;  SIN PRESCALER-> TME1 1:1, TIMER1 PRENDIDO, CLOCK INTERNO
        CLRF PIR1; TODAS LAS BANDERAS INTERRUPCION PIR1 APAGADAS



        CLRF CENTISEGUNDO ; LIMPIAR VARIABLES TIEMPO
        CLRF SEGUNDO
        CLRF DECASEGUNDO
        CLRF MINUTO
        CLRF DECAMINUTO
        CLRF HORA
        CLRF DECAHORA
        CLRF DIA
        CLRF MINUTO_A
        CLRF DECAMINUTO_A
        CLRF HORA_A
        CLRF DECAHORA_A
        CLRF BITFLAGS
        CLRF ALARMAFLAG

        MOVLW 0X07
        MOVWF DESIMA_T; X DEFECTO 700mV PARA EL MAXIMA TEMPERATURA REPRESENTA 70 GRADOS
        CLRF CENTESIMA_T
        CLRF UNIDAD_T
        CLRF ALARMA_T
        CLRF DESIMA_A
        CLRF CENTESIMA_A
        CLRF UNIDAD_A

        MOVLW 0X01      ; PROCESO 1= NOPROCESO
        MOVWF CRONOPROC
        MOVWF PROCESO

        BCF PORTA,2
        BCF PORTA,3     ; CALENTADOR, LINTERNA Y MOTOR APAGADOS
        BCF PORTA,5
```

```
        CLRF PETICION ; PREPARANDO VARIABLES DE LA INTERFAZ
        CLRF FUNCION
        CLRF MENU
        BSF MENU,0 ; MENU=1
        CLRF DECAMENU
        CLRF OPCIONMOS; OPCIONMOS=0
        CLRF PARPADEOBIT;
        MOVLW 0X0A
        MOVWF LETRAMENU; CODIGOS PARA MOSTRAR EN EL DISPLAY

        MOVLW 0XFF    ; SENSO Y BOTON SE ASUMEN QUE EMPIEZAN EN ESTADOS NEUTROS
        MOVWF SENSO
        BCF SENSO,5
        MOVWF BOTON



        CALL MOSTRAR



        CLRF TMR1H   ; LIMPIAR TIMERS
        CLRF TMR1L
        CLRF TMR0

        MOVLW B'11100000' ;ACTIVACION INTERRUPCIONES (TIMER 1 Y TIMER0)
        MOVWF INTCON
;-------------------------------------------------------------
;-------------------------------------------------------------

 ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>BUCLE PRINCIPAL<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>

BUCLE:      GOTO PROCESO_ACTUAL; EN DIRECCION FLASH 0X0700


VERCONVERSOR:   BTFSC ADCON0,0; VER SI ESTA ENCENDIDO EL CONVERSOR
        GOTO CONVERSION;

VERALARMA:    BTFSC ALARMAFLAG,0; VER SI ESTA ENCENDIDA LA ALARMA
        GOTO REVISARHORA;



VERBOTON:     MOVF PORTC,0
        MOVWF SENSO; DEBE ESTAR CONSTANTEMENTE ACTUALIZADO
        SUBLW B'11011111';CHEQUEAR SI PORTC ES DISTINTO DE 11011111
        BTFSS STATUS,Z
        GOTO VERIFICAR_ON; SI ES DISTINTO ES PORQUE SE APRETO UN BOTON O SE SENSO ALGO DISTINTO AL ESTADO DE REPOSO

        BTFSC PETICION,6;
        GOTO VERIFICAR_OFF   ; HASTA QUE EL BOTÓN NO SE SUELTE NO HABRÁ PETICIÓN


VERFUNCION:    GOTO FUNCIONACTUAL; EN FUNCIONACTUAL SE PREGUNTA QUE TAREA SE DEBE REALIZAR EN ESE MOMENTO




;-------------------------------------------------
;-------------------------------------------------



;-----------------------------FUNCION DE RETARDO GENERAL
WAIT:       MOVLW 0X7F;  W=0X7F

WAIT_PARAMETRO: MOVWF RETARDO; RETARDO=W  SI SE CARGA PREVIAMENTE A W SE HARAN W CICLOS

ESPERA:     NOP             ;SE HARAN 4*CICLOS DE INSTRUCCIÓN POR CICLO DE RETARDO
        DECFSZ RETARDO
        GOTO ESPERA

        RETURN
;---------------------------FUNCION DE ENTRADAS EN PUERTOC

VERIFICAR_ON:   CALL WAIT     ;ESPERAR
                ; NOTA: NO OLVIDAR QUE LOS BOTONES SON TIPO PULL UP
                ; CUANDO ESTA APRETADO ES CERO Y CUANDO ESTA SUELTO ES UNO

        MOVF PORTC,0    ; VER PORTC
        MOVWF BOTON      ; GUARDAR EL VALOR EN "BOTON"
        SUBLW B'11011111'
        BTFSC STATUS,Z   ; VERIFICAR QUE ESTE FUERA DEL ESTADO DE REPOSO
```

```
        GOTO VERFUNCION


        CALL WAIT      ; ESPERAR

        MOVF PORTC,0
        SUBWF BOTON,0
        BTFSS STATUS,Z    ;VERIFICAR QUE SE MANTIENE EL VALOR DE PORTC CON LO GUARDADO
        GOTO VERFUNCION



        MOVLW B'11110000'  ; AQUI SE GUARDAN LOS 4 VALORES DE LOS SENSORES. 1:DESACTIVADO//0:ACTIVADO
        ANDWF BOTON,0
        MOVWF SENSO

        BTFSC PETICION,6;
        GOTO VERIFICAR_OFF   ; HASTA QUE EL BOTÓN NO SE SUELTE NO HARÁ EFECTO


        MOVLW B'00001111'
        ANDWF BOTON,0
        SUBLW B'00001111'
        BTFSC STATUS,Z   ; ¿HAY CAMBIO EN LOS PRIMEROS 4 BITS?
        GOTO  VERFUNCION;     ; NO: IR A BUCLE


        BTFSS BOTON,1     ; SEGUN EL BOTON QUE SE APRIETE SE LLENA OPCION
        MOVLW 0X01        ; SI SE APRIETA MAS DE UNO. SOLO UNO ES REGISTRADO
        BTFSS BOTON,2     ; LA PRIORIDAD ES RC0, RC3, RC2, RC1
        MOVLW 0X02
        BTFSS BOTON,3
        MOVLW 0X03
        BTFSS BOTON,0
        MOVLW 0X00
        MOVWF PETICION;

        BSF PETICION,6;     ACTIVAR BIT DE VERIFICACION DE APAGADO(OFF) DE BOTON.
        GOTO VERFUNCION;

;-------------------------------------------------------------------------

VERIFICAR_OFF:  MOVLW B'00001111'  ;VERIFICAR QUE PORTC ESTE SIN APRETAR
        ANDWF PORTC,0
        SUBLW B'00001111'
        BTFSS STATUS,Z
        GOTO VERFUNCION ; HASTA  QUE NO SE SUELTEN LOS BOTONES LA PETICIÓN NO ES REALIZADA

        CALL WAIT

        MOVLW B'00001111'  ;VERIFICAR QUE PORTC SIGA SIN APRETAR
        ANDWF PORTC,0
        SUBLW B'00001111'
        BTFSS STATUS,Z
        GOTO VERFUNCION ; SINO VOLVER AL BUCLE PRINCIPAL


        BCF PETICION,6;     DESACTIVAR BIT DE VERIFICAR OFF
        BSF PETICION,7;     ACTIVAR BIT DE PETICIÓN
        GOTO VERFUNCION ;    EL BOTON HA SIDO APRETADO CORRECTAMENTE, HAY UNA PETICION PENDIENTE!
;-----------------------------------------


REVISARHORA:    MOVF DECAHORA_A,0
        SUBWF DECAHORA,0     ; ¿DECAHORA=DECAHORA_A?
        BTFSS STATUS,Z
        GOTO VERBOTON       ; NO:


        MOVF HORA_A,0       ;SI
        SUBWF HORA,0
        BTFSS STATUS,Z;       ¿HORA=HORA_A?
        GOTO VERBOTON        ;NO

        MOVF DECAMINUTO_A,0    ;SI
        SUBWF DECAMINUTO,0    ;¿DECAMINUTO=DECAMINUTO_A?
        BTFSS STATUS,Z
        GOTO VERBOTON        ;NO

        MOVF MINUTO_A,0      ;SI
        SUBWF MINUTO,0       ;¿DECAMINUTO=DECAMINUTO_A?
        BTFSS STATUS,Z
        GOTO VERBOTON        ;NO

                ;SI

        BCF ALARMAFLAG,0; SE APAGA LA ALARMA
```

```
            MOVF UNIDAD_A,0       ; UNIDAD_T=UNIDAD_A// DEIMA_T=DESIMA_A//CENTESIMA_T=CENTESIMA_A
            MOVWF UNIDAD_T
            MOVF DESIMA_A,0
            MOVWF DESIMA_T
            MOVF CENTESIMA_A,0
            MOVWF CENTESIMA_T

            MOVLW .26  ;  FUNCION = ALARMA TENSION
            MOVWF FUNCION
            GOTO VERFUNCION
;---------------------------------------


;---------------DIVIDIR

DIV:         CLRF ARIT_AUX;  WREG= NUM1 /NUM2(DIVISION ENTERA)..Resto en NUM1
            MOVF NUM2,0;

BUCLEDIV:     INCF ARIT_AUX
            SUBWF NUM1,1
            BTFSC STATUS,C;
            GOTO BUCLEDIV

            ADDWF NUM1,1
            DECF ARIT_AUX
            MOVF ARIT_AUX,0

            RETURN



;-------------------------------------------------------------------------
*************************************************************************************************************************FUNCION MOSTRAR
;----------MOSTRAR: FUNCION QUE DEBE ADAPTARSE AL DISPOSITIVO A USAR PARA VISUALIZAR LAS SALIDAS----
                  ;ESTE CASO ES PARA 4 DISPLAYS 7 SEGMENTOS

            ORG 0X0150  ;
MOSTRAR:
            MOVF PCLATH,0
            MOVWF PCLATHAUX1;  SE ALMACENA EL PCLATH ACTUAL

            CLRF PCLATH;
            BSF PCLATH,0; PCLATH=1

            MOVF OPCIONMOS,0 ; WREG=OPCIONMOS.. OPCIONMOS ES LA VARIABLE QUE INDICA QUE DEBE APARECER EN PANTALLA

            ADDWF PCL,1   ;  SEGUN LO QUE SEA OPCIONMOS SE ELIGE
            GOTO MOSTRAR0
            GOTO MOSTRAR1
            GOTO MOSTRAR2
            GOTO MOSTRAR3
            GOTO MOSTRAR4
            GOTO MOSTRAR5
            GOTO MOSTRAR6
            GOTO MOSTRAR7
            GOTO MOSTRAR8
            GOTO MOSTRAR9

;CARGAR PUNTERO_0, PUNTERO_1, PUNTERO_2, PUNTERO_3 QUE SON LOS QUE UTILIZA EL REFRESCO DEL DISPLAY


;ATENCION!!: ES IMPORTANTE QUE LOS VALORES SEAN DEL 0X00 AL 0X0F YA QUE NO HAY NINGUN FILTRO,
; OTRO VALOR PUEDE HACER QUE SE PRENDA MAS DE UN DISPLAY POR VEZ



MOSTRAR0:     MOVLW MENU        ; OPCIONMOS=0  MOSTRAR MENU
            MOVWF PUNTERO_0
            MOVLW DECAMENU
            MOVWF PUNTERO_1
            MOVLW LETRAMENU
            MOVWF PUNTERO_2
            MOVLW LETRAMENU
            MOVWF PUNTERO_3
            GOTO FINMOSTRAR

MOSTRAR1:     MOVLW MINUTO       ; OPCIONMOS=1  MOSTRAR HORAS Y MINUTOS
            MOVWF PUNTERO_0
            MOVLW DECAMINUTO
            MOVWF PUNTERO_1
            MOVLW  HORA
            MOVWF PUNTERO_2
            MOVLW DECAHORA
            MOVWF PUNTERO_3
```

```
        GOTO FINMOSTRAR

MOSTRAR2:   MOVLW SEGUNDO   ;  OPCIONMOS=2    MOSTRAR MINUTOS Y SEGUNDOS
        MOVWF PUNTERO_0
        MOVLW DECASEGUNDO
        MOVWF PUNTERO_1
        MOVLW MINUTO
        MOVWF PUNTERO_2
        MOVLW DECAMINUTO
        MOVWF PUNTERO_3
        GOTO FINMOSTRAR

MOSTRAR3:   MOVLW MINUTO_A      ; OPCIONMOS=3  MOSTRAR HORA DE ALARMA
        MOVWF PUNTERO_0
        MOVLW DECAMINUTO_A
        MOVWF PUNTERO_1
        MOVLW HORA_A
        MOVWF PUNTERO_2
        MOVLW DECAHORA_A
        MOVWF PUNTERO_3
        GOTO FINMOSTRAR
;----------------------------------------
MOSTRAR4:   MOVLW ALARMAFLAG
        MOVWF PUNTERO_0
        MOVLW LETRAIZQ
        MOVWF PUNTERO_1    ; ACTIVAR ALARMA SI/NO
        MOVLW LETRAIZQ
        MOVWF PUNTERO_2
        MOVLW LETRAIZQ
        MOVWF PUNTERO_3
        GOTO FINMOSTRAR

MOSTRAR5:   MOVLW PROCESO
        MOVWF PUNTERO_0
        MOVLW LETRADER  ;   ALARMA SONANDO!!
        MOVWF PUNTERO_1
        MOVLW LETRADER
        MOVWF PUNTERO_2
        MOVLW LETRADER
        MOVWF PUNTERO_3
        GOTO FINMOSTRAR

MOSTRAR6:   MOVLW MILESIMA  ; MOSTRAR TENSION
        MOVWF PUNTERO_0
        MOVLW CENTESIMA
        MOVWF PUNTERO_1
        MOVLW DESIMA
        MOVWF PUNTERO_2
        MOVLW LETRAIZQ
        MOVWF PUNTERO_3
        GOTO FINMOSTRAR;

MOSTRAR7:   MOVLW LETRADER ; MOSTRAR TENSION FINAL
        MOVWF PUNTERO_0
        MOVLW CENTESIMA_T
        MOVWF PUNTERO_1
        MOVLW DESIMA_T
        MOVWF PUNTERO_2
        MOVLW LETRADER
        MOVWF PUNTERO_3
        GOTO FINMOSTRAR;

MOSTRAR8:   MOVLW LETRADER ; MOSTRAR TENSION FINAL PARA ALARMA RELOJ
        MOVWF PUNTERO_0
        MOVLW CENTESIMA_A
        MOVWF PUNTERO_1
        MOVLW DESIMA_A
        MOVWF PUNTERO_2
        MOVLW LETRAIZQ
        MOVWF PUNTERO_3
        GOTO FINMOSTRAR;

MOSTRAR9:   MOVLW LETRADER ; ERRORES DE PROCESO
        MOVWF PUNTERO_0
        MOVLW LETRADER
        MOVWF PUNTERO_1
        MOVLW ERRORNUM
        MOVWF PUNTERO_2
        MOVLW LETRADER
        MOVWF PUNTERO_3
        GOTO FINMOSTRAR;



FINMOSTRAR:   MOVF PCLATHAUX1,0
```

```
            MOVWF PCLATH
            RETURN


;------------------------------


;-----------------------------------------------*************************************************************************
;*************************************************************************************************FUNCIONES PRINCIPALES
'           ORG 0X01FE
FUNCIONACTUAL:              ; LAS FUNCIONES DE USUARIO (FUNCIONES PARA EL USUARIO) QUE ESTEN ENTRE LA MEMORIA 0X0200 HASTA 0X02FF
                    ; NO NECESITAN MODIFICAR EL PCLATH.
                       ; LAS SUBFUNCIONES DE LAS FUNCIONES SE ENCUENTRAN A PARTIR DE LA MEMORIA 0X0400.
                       ; EL REGISTRO PCLATH COMPLEMENTA AL PCL.



            MOVLW 0X02
            MOVWF PCLATH ; PCLATH ES 0X02

            MOVF FUNCION,0;
            ADDWF PCL,1

            GOTO FUNCION0;   MENU INICIAL
            GOTO FUNCION1;   HORA
            GOTO FUNCION2;   ALARMA   (CABIAR HORA DE ALARMA)
            GOTO FUNCION3;   TENSION ACTUAL/
            GOTO FUNCION4;   SETEAR TENSION FINAL Y LUEGO ACTIVAR ALARMATENSION
            GOTO FUNCION5;   ABRIR/CERRAR
            NOP;6
            NOP;7          RESERVADAS PARA FUTURAS FUNCIONES
            NOP;8
            NOP;9
            NOP;10
            NOP;11
            NOP;12
            NOP;13
            NOP;14
            ;-------------------SUBFUNCIONES
            GOTO CAMBIARHORA;15 ; DE FUNCION1
            GOTO CAMBIARMIN;16
            GOTO CAMBIARMINALARM;17 DE FUNCION2
            GOTO CAMBDESIMA_A;18
            GOTO CAMBCENTESIMA_A;19
            GOTO ACTIVARALARMA;20
            GOTO CAMBCENTESIMA_T;21 DE FUNCION4
            NOP;GOTO ACTIVAR_T;22 (OBSOLETA)
            NOP;23
            NOP;24
            GOTO ALARMA;25  DE FUNCION4 Y ALARMA PRINCIPAL
            GOTO ALARMATENSION;26  DE FUNCION4
            NOP;27
            NOP;28
            NOP;29
            GOTO ERRORDEPROCESO;30  GENERAL DE PROCESO



            CLRF FUNCION;
            GOTO BUCLE



;------------------------------FUNCION=0   MENU INICIAL

FUNCION0:
             MOVLW 0X00
             SUBWF OPCIONMOS,0;    VERIFICAR QUE OPCIONMOS SEA EL CORRESPONDIENTE AL DE LA FUNCION ACTUAL
             BTFSC STATUS,Z
             GOTO F0PETICION

             CLRF PARPADEOBIT
             CLRF OPCIONMOS      ; OPCIONMOS=0 (MOSTRAR MENU)
             CALL MOSTRAR

F0PETICION:     BTFSS PETICION,7; VER SI HAY PETICION
             GOTO BUCLE;       VOLVER AL BUCLE PRINCIPAL

             BCF PETICION,7; LIMPIAR BIT DE PETICION
```

```
        MOVF PETICION,0

        ADDWF PCL,1 ;           ¿¿¿QUE HACER???>>>>>>

        GOTO BUCLE;             PETICION=0 NO HACE NADA
        GOTO RESTARMENU         ;PETICION=1  SUBE OPCION DEL MENU
        GOTO SUMARMENU          ;PETICION=2  BAJA OPCION DEL MENU
        MOVF MENU,0;            PETICION=3 IR A LA FUNCION MOSTRADA POR EL MENU
        MOVWF FUNCION           ; FUNCION=MENU
        GOTO BUCLE




;------------------------------FUNCION=1     VER HORA
FUNCION1:       MOVLW 0X01
        SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X01 (MOSTRAR HORA Y MINUTOS)
        BTFSC STATUS,Z
        GOTO F1PETICION

        CLRF PARPADEOBIT
        MOVLW 0X02
        SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X02 (MOSTRAR MINUTOS Y SEGUNDOS)
        BTFSC STATUS,Z
        GOTO F1PETICION
        GOTO F1_1

F1PETICION:     BTFSS PETICION,7; VER SI HAY PETICION
        GOTO BUCLE

        BCF PETICION,7; LIMPIAR BIT DE PETICION


        MOVF PETICION,0  ; ¿PETICION?
        ADDWF PCL, 1

        GOTO F1_0       ;=0: VOLVER A LA FUNCION0(VOLVER AL MENU PRINCIPAL)
        GOTO F1_1       ;=1: VER HORAS Y MINUTOS
        GOTO F1_2       ;=2: VER MINUTOS Y SEGUNDOS

        MOVLW .15       ;=3: IR A FUNCION15 (CAMBIAR HORA)/ FUNCION=.15
        CLRF OPCIONMOS; BORRAR OPCIONMOS PARA QUE LA SIGUIENTE FUNCION TENGA QUE ACTUALIZARLA
        MOVWF FUNCION
        GOTO BUCLE



;------------------------------ FUNCION=15       CAMBIAR HORA
CAMBIARHORA:
        MOVLW 0X01
        SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X01
        BTFSC STATUS,Z
        GOTO F15PETICION

        MOVLW 0X0F
        MOVWF LETRAIZQ
        MOVLW B'00000101'; ACTIVAR PARPADEO IZQUIERDO
        MOVWF PARPADEOBIT

        GOTO F1_1

F15PETICION:    BTFSS PETICION,7; VER SI HAY PETICION
        GOTO BUCLE

        BCF INTCON,7; DESHABILITAR INTERRUCIONES

        BCF PETICION,7; LIMPIAR BIT DE PETICION

        MOVF PETICION,0


        ADDWF PCL, 1
        GOTO F15VOLVER;VOLVER A FUNCION1
        GOTO RESTAHOR; DECREMENTAR HORA
        GOTO SUMAHOR; INCREMENTAR HORA

        MOVLW .16    ; IIR A FUNCION CAMBIAR MINUTO (FUNCION=16)
        MOVWF FUNCION
        BSF INTCON,7 ;HABILITAR INTERRUPCION
        CLRF OPCIONMOS; BORRAR OPCIONMOS PARA QUE LA SIGUIENTE FUNCION TENGA QUE ACTUALIZARLA
        GOTO BUCLE
```

```
;---------------------- FUNCION=16  CAMBIAR MINUTOS
CAMBIARMIN:
        MOVLW 0X01
        SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X01
        BTFSC STATUS,Z
        GOTO F16PETICION


        MOVLW 0X0F
        MOVWF LETRADER
        MOVLW B'00000011'; ACTIVAR PARPADEO DERECHO
        MOVWF PARPADEOBIT
        GOTO F1_1

F16PETICION:    BTFSS PETICION,7; VER SI HAY PETICION
        GOTO BUCLE

        BCF INTCON,7; DESHABILITAR INTERRUCIONES

        BCF PETICION,7; LIMPIAR BIT DE PETICION


        MOVF PETICION,0


        ADDWF PCL, 1

        GOTO F15VOLVER;VOLVER A FUNCION1
        GOTO RESTAMIN; DECREMENTAR MINUTO
        GOTO SUMAMIN; INCREMENTAR MINUTO
        MOVLW .15    ;IR A FUNCION CAMBIAR HORA (FUNCION=15)
        MOVWF FUNCION
        CLRF OPCIONMOS;  BORRAR OPCIONMOS PARA QUE LA SIGUIENTE FUNCION TENGA QUE ACTUALIZARLA
        BSF INTCON,7 ;HABILITAR INTERRUPCION
        GOTO BUCLE



;----------------------FUNCION=2 CAMBIAR HORA DE ALARMA

FUNCION2:       MOVLW 0X03
        SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X03
        BTFSC STATUS,Z
        GOTO F2PETICION


        MOVLW 0X0D
        MOVWF LETRADER
        MOVLW 0X0F
        MOVWF LETRAIZQ

        MOVLW B'00000111'; ACTIVAR PARPADEO DERECHA
        MOVWF PARPADEOBIT
        GOTO F2_A

F2PETICION:     BTFSS PETICION,7; VER SI HAY PETICION
        GOTO BUCLE


        BCF PETICION,7; LIMPIAR BIT DE PETICION

        MOVF PETICION,0


        ADDWF PCL, 1
        GOTO F2VOLVER_A;VOLVER A FUNCION0
        GOTO RESTAHOR_A; DECREMENTAR HORA
        GOTO SUMAHOR_A; INCREMENTAR HORA

        MOVLW .17    ; IR A FUNCION CAMBIAR MINUTO ALARMA (FUNCION=17)
        MOVWF FUNCION
        CLRF OPCIONMOS; BORRAR OPCIONMOS PARA QUE LA SIGUIENTE FUNCION TENGA QUE ACTUALIZARLA
        GOTO BUCLE


;----------------------FUNCION=17  CAMBIAR MINUTOS DE ALARMA
CAMBIARMINALARM:
        MOVLW 0X03
        SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X03
        BTFSC STATUS,Z
        GOTO F17PETICION

        MOVLW 0X0D
        MOVWF LETRAIZQ
        MOVLW 0X0F
```

```
            MOVWF LETRADER
            MOVLW B'00000111'; ACTIVAR PARPADEO IZQUIERDO
            MOVWF PARPADEOBIT
            GOTO F2_A

F17PETICION:    BTFSS PETICION,7; VER SI HAY PETICION
            GOTO BUCLE


            BCF PETICION,7; LIMPIAR BIT DE PETICION


            MOVF PETICION,0


            ADDWF PCL, 1

            GOTO F18VOLVER;VOLVER A FUNCION2(CAMBIAR HORA ALARMA)
            GOTO RESTAMIN_A; DECREMENTAR MINUTO
            GOTO SUMAMIN_A; INCREMENTAR MINUTO
            MOVLW .18    ;IR A FUNCION CAMBIAR TENSION_A TENSION PARA LA ALARMA (FUNCION= 18)
            MOVWF FUNCION
            CLRF OPCIONMOS; BORRAR OPCIONMOS PARA QUE LA SIGUIENTE FUNCION TENGA QUE ACTUALIZARLA

            GOTO BUCLE

;-------------------------FUNCION=20  ACTIVAR ALARMA
ACTIVARALARMA:

            MOVLW 0X04
            SUBWF OPCIONMOS,0;    ¿eS OPCIONMOS=4?
            BTFSC STATUS,Z
            GOTO F20PETICION      ;SI: IR A F20PETICION

            CLRF PARPADEOBIT
            MOVLW 0X0D
            MOVWF LETRAIZQ
            MOVLW 0X04
            MOVWF OPCIONMOS     ;NO: OPCIONMOS=04 (MOSTRAR ACTIVACION ALARMA)
            CALL MOSTRAR

F20PETICION:    BTFSS PETICION,7; VER SI HAY PETICION
            GOTO BUCLE;       VOLVER AL BUCLE PRINCIPAL

            BCF PETICION,7; LIMPIAR BIT DE PETICION

            MOVF PETICION,0
            ADDWF PETICION,0

            ADDWF PCL,1 ;             ¿¿¿QUE HACER???>>>>>>


            CLRF FUNCION;             ;PETICION=0   VOLVER A MENU
            GOTO BUCLE;

            BCF ALARMAFLAG,0          ;PETICION=1  APAGA ALARMAFLAG
            GOTO BUCLE

            BSF ALARMAFLAG,0          ;PETICION=2  PRENDE ALARMAFLAG
            GOTO BUCLE

            MOVLW 0X02              ;PETICION=3 IR A LA FUNCION2
            MOVWF FUNCION             ; FUNCION=MENU
            GOTO BUCLE

;-----------------------------------------


;-------------------------FUNCION=3:   VERSION ANALOGICA A DIGITAL:

FUNCION3:
            MOVLW 0X06
            SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X06
            BTFSC STATUS,Z
            GOTO F3PETICION

            BSF ADCON0,0; HABILITAR CONVERSIONES
            CLRF ADPROMH
            CLRF ADPROML

            MOVLW 0X0C
            MOVWF LETRAIZQ
            CLRF PARPADEOBIT
            MOVLW 0X06
            MOVWF OPCIONMOS     ;NO: OPCIONMOS=06 (MOSTRAR TENSION ACTUAL)
            CALL MOSTRAR
```

```
F3PETICION:     BTFSS PETICION,7; VER SI HAY PETICION
                GOTO BUCLE


                BCF PETICION,7; LIMPIAR BIT DE PETICION
                MOVF PETICION,0
                ADDWF PCL,1 ;           ¿¿¿QUE HACER???>>>>>>

                NOP;    0:VOLVER AL MENU
                NOP;    1:VOLVER AL MENU
                GOTO F3VOLVER;  2:VOLVER AL MENU
                GOTO F3COMPLETO; 3: VER LA PANTALLA COMPLETA

F3VOLVER:       CLRF FUNCION; IR A MENU

                BCF ADCON0,0; DESHABILITAR CONVERSIONES
                GOTO BUCLE


;-------------------------FUNCION=4  TENSION PARA ALARMATENSION  CAMBIAR UNIDAD Y DESIMA
FUNCION4:
                MOVLW 0X07
                SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X07
                BTFSC STATUS,Z
                GOTO F4PETICION    ; SI: IR A F4PETICION

                MOVLW 0X0F      ;NO: LETRAIZQ=LETRADER=0XF, PARPADEOBIT=00000101, OPCIONMOS=7,
                MOVWF LETRAIZQ
                MOVWF LETRADER
                MOVLW B'00000101'
                MOVWF PARPADEOBIT
                MOVLW 0X07
                MOVWF OPCIONMOS     ;NO: OPCIONMOS=07 (MOSTRAR TENSION PARA ALARMATENSION)
                CALL MOSTRAR

F4PETICION:     BTFSS PETICION,7; VER SI HAY PETICION
                GOTO BUCLE


                BCF PETICION,7; LIMPIAR BIT DE PETICION

                MOVF PETICION,0
                ADDWF PCL,1 ;           ¿¿¿QUE HACER???>>>>>>

                GOTO F4VOLVER;    0:VOLVER AL MENU
                GOTO RESTARDEC_T; 1:INCREMENTAR DESIMA_T
                GOTO SUMARDEC_T;  2:DECREMENTAR DESIMA_T

                MOVLW .21
                MOVWF FUNCION    ;3:FUNCIÓN=21
                CLRF OPCIONMOS   ; LIMPIAR OPCIONMOS PARA QUE LA SIGUIENTE FUNCION DEBA ACTUALIZARLA
                GOTO BUCLE
;-------------------------

;-------------------------------FUNCION=21 CAMBIAR CENTESIMA PARA ALARMATENSION
CAMBCENTESIMA_T:
                MOVLW 0X07
                SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X07
                BTFSC STATUS,Z
                GOTO F21PETICION

                MOVLW 0X0F      ;NO: LETRAIZQ=LETRADER=0XF, PARPADEOBIT=00000011, OPCIONMOS=7,
                MOVWF LETRADER
                MOVLW B'00000011'
                MOVWF PARPADEOBIT
                MOVLW 0X07
                MOVWF OPCIONMOS     ;NO: OPCIONMOS=07 (MOSTRAR TENSION PARA ALARMATENSION)
                CALL MOSTRAR


F21PETICION:    BTFSS PETICION,7; VER SI HAY PETICION
                GOTO BUCLE

                BCF PETICION,7; LIMPIAR BIT DE PETICION

                MOVLW 0X03
                MOVWF PCLATH


                MOVF PETICION,0
                ADDWF PCL,1 ;           ¿¿¿QUE HACER???>>>>>>

                GOTO F4VOLVER;    0:VOLVER AL MENU
                GOTO RESTARCEN_T; 1:DECREMENTAR CENTESIMA_T
                GOTO SUMARCEN_T;  2:INCREMENTAR CENTESIMA_T
```

```
            MOVLW .26
            MOVWF FUNCION    ;3: VOLVER A FUNCION=26
            GOTO BUCLE




;-------------------------FUNCION=26 VER  ALARMA DE TENSION


ALARMATENSION:
            MOVLW 0X06
            SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X06
            BTFSC STATUS,Z
            GOTO F26CHECK

            BSF ADCON0,0; HABILITAR CONVERSIONES
            BSF ALARMA_T,0; PRENDER BIT DE ALARMA POR TENSION
            CLRF ADPROMH
            CLRF ADPROML;  BORRAR ADPROML YA QUE GUARDA EL PROMEDIO DE LA ULTIMA CONVERSION


            MOVLW 0X03
            MOVWF PROCESO; CALENTAR

            MOVLW 0X0F
            MOVWF LETRAIZQ
            CLRF PARPADEOBIT
            MOVLW 0X06
            MOVWF OPCIONMOS      ;NO: OPCIONMOS=06 (MOSTRAR TENSION/TEMPERATURA ACTUAL)
            CALL MOSTRAR

F26CHECK:       BTFSS ALARMA_T,7; ESTE BIT SE ACTIVA CUANDO EL VALOR DE TENSION ES SUPERIOR O IGUAL AL DEL SETEADO
            GOTO F26PETICION


            BCF ADCON0,0; DESHABILITAR CONVERSIONES
            BCF PORTA,3; APAGAR CALENTADOR
            MOVLW 0X01;
            MOVWF PROCESO; TERMINAR PROCESO

            MOVLW .25
            MOVWF FUNCION
            CLRF ALARMA_T; APARGAR BITS DE ALARMA POR TENSION

            GOTO BUCLE



F26PETICION     BTFSS PETICION,7; VER SI HAY PETICION
            GOTO BUCLE


            BCF PETICION,7; LIMPIAR BIT DE PETICION

            BCF ADCON0,0; DESHABILITAR CONVERSIONES

            BCF PORTA,3; APAGAR CALENTADOR
            MOVLW 0X01;
            MOVWF PROCESO; TERMINAR PROCESO


            MOVLW 0X04
            MOVWF FUNCION            ; FUNCION=4
            CLRF ALARMA_T
            GOTO BUCLE


 ;-------------------------FUNCION=25 ALARMA ACTIVADA!!!!!<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
ALARMA:         MOVLW 0X05
            SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X05
            BTFSC STATUS,Z
            GOTO F25PETICION


            MOVLW 0X0C
            MOVWF LETRADER
            CLRF PARPADEOBIT
            MOVLW 0X05
            MOVWF OPCIONMOS
            CALL MOSTRAR
            MOVLW 0X04
```

```
        MOVWF PROCESO; PROCESO=4            PRENDER LINTERNA PARA LUEGO ABRIR VALVULA


F25PETICION:    MOVLW 0X00
        SUBWF PROCESO,0
        BTFSC STATUS,Z;     ¿(PROCESO=0)?
        CLRF FUNCION        ;SI => FUNCION=00 (SALIR DE ALARMA)

                ;NO:....
        BTFSS PETICION,7; VER SI HAY PETICION
        GOTO BUCLE

        BCF PETICION,7; LIMPIAR PETICION

        CLRF PROCESO; PROCESO=0

        CLRF FUNCION

        GOTO BUCLE

;---------------------------------FUNCION=18  CAMBIAR DESIMA Y UNIDAD PARA ALARMA TENSION CON ALARMA RELOJ
CAMBDESIMA_A:
        MOVLW 0X08
        SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X08
        BTFSC STATUS,Z
        GOTO F18PETICION    ; SI: IR A F18PETICION

        MOVLW 0X0D
        MOVWF LETRADER
        MOVLW 0X0F
        MOVWF LETRAIZQ
        MOVLW B'00000101'
        MOVWF PARPADEOBIT
        MOVLW 0X08
        MOVWF OPCIONMOS      ;NO: OPCIONMOS=08 (MOSTRAR TENSION PARA ALARMATENSION)
        CALL MOSTRAR

F18PETICION:    BTFSS PETICION,7; VER SI HAY PETICION
        GOTO BUCLE


        BCF PETICION,7; LIMPIAR BIT DE PETICION

        MOVLW 0X03
        MOVWF PCLATH

        MOVF PETICION,0
        ADDWF PCL,1 ;            ¿¿¿QUE HACER???>>>>>>

        GOTO F18VOLVER;   VOLVER A FUNCION2
        GOTO RESTARDEC_A;
        GOTO SUMARDEC_A;

        MOVLW .19         ;IR A CAMBCENTESIMA_A
        MOVWF FUNCION
        CLRF OPCIONMOS
        GOTO BUCLE
;----------------------------

;---------------------------------FUNCION=19 CAMBIAR CENTESIMA PARA ALARMATENSION CON ALARMA RELOJ!!
CAMBCENTESIMA_A:
        MOVLW 0X08
        SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X08
        BTFSC STATUS,Z
        GOTO F19PETICION

        MOVLW 0X0F
        MOVWF LETRADER
        MOVLW 0X0D
        MOVWF LETRAIZQ

        MOVLW B'00000011'
        MOVWF PARPADEOBIT
        MOVLW 0X08
        MOVWF OPCIONMOS      ;NO: OPCIONMOS=08 (MOSTRAR TENSION PARA ALARMATENSION)
        CALL MOSTRAR

F19PETICION:    BTFSS PETICION,7; VER SI HAY PETICION
        GOTO BUCLE

        BCF PETICION,7; LIMPIAR BIT DE PETICION

        MOVLW 0X03
        MOVWF PCLATH

        MOVF PETICION,0
```

```
        ADDWF PCL,1 ;           ¿¿¿QUE HACER???>>>>>>

        GOTO F18VOLVER;   VOLVER A FUNCION2
        GOTO RESTARCEN_A;
        GOTO SUMARCEN_A;

        MOVLW .20;  IR A ACTIVAR ALARMA RELOJ
        MOVWF FUNCION
        GOTO BUCLE



;-----------------------------------FUNCION=5          ABRIR/CERRAR VALVULA


FUNCION5:       MOVLW 0X05
        SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X05
        BTFSC STATUS,Z
        GOTO F5PETICION


        MOVLW 0X0B
        MOVWF LETRADER
        CLRF PARPADEOBIT
        MOVLW 0X05
        MOVWF OPCIONMOS
        CALL MOSTRAR


F5PETICION:     BTFSS PETICION,7; VER SI HAY PETICION
        GOTO BUCLE

        BCF PETICION,7; LIMPIAR BIT DE PETICION

        MOVLW 0X03   ;  MEMORIA A LA ALTURA DEL 0X0300
        MOVWF PCLATH

        MOVF PETICION,0
        ADDWF PETICION,0  ; SALTAR DE ADOS

        ADDWF PCL,1 ;           ¿¿¿QUE HACER???>>>>>>

        CLRF FUNCION
        GOTO BUCLE;  PETICON=0: VOLVER A MENU


        CLRF PROCESO;  PETICON=1: PROCESO=0 (CERRAR VALVULA)
        GOTO BUCLE;

        CLRF PROCESO;  PETICON=2: PROCESO=0 (CERRAR VALVULA)
        GOTO BUCLE;

        MOVLW 0X04;     PETICION=3: PROCESO=4 (PREPARAR LINTERNA PARA ABRIR Y LUEGO HACERLO )
        MOVWF PROCESO


        GOTO BUCLE

;-------------------------------FUNCION=30  ERROR DE PROCESO
ERRORDEPROCESO:  MOVLW 0X09
        SUBWF OPCIONMOS,0;   CHEQUEAR SI OPCIONMOS ES 0X01
        BTFSC STATUS,Z
        GOTO F30PETICION

        BCF ADCON0,0; DESHABILITAR CONVERSIONES

        MOVLW 0XE
        MOVWF LETRADER
        CLRF PARPADEOBIT
        MOVLW 0X09
        MOVWF OPCIONMOS
        CALL MOSTRAR


F30PETICION:     BTFSS PETICION,7; VER SI HAY PETICION
        GOTO BUCLE
        BCF PETICION,7; LIMPIAR PETICION

        CLRF FUNCION
        GOTO BUCLE
```

```
;.............**************************************.....................................**


;----.............**************************************************.....................
;*******************************************************************************FUNCIONES AUXILIARES




;-------------------------------------------------------.
;---------------------- CODIGOS AUXILIARES DE LAS FUNCIONES
            ORG 0X400


;--------------------FUNCION=0 FUNCION0 AUXILIAR           (MENU)
SUMARMENU:      INCF MENU           ;SUMAR 1 AL MENU
            MOVLW 0X06
            SUBWF MENU,0        ; COMPARAR CON 0X06
            BTFSS STATUS,Z
            GOTO DECAMENU1      ; SI LLEGO A 0X06 VOLVER A 0X01
            CLRF MENU
            BSF MENU,0
            GOTO DECAMENU1


RESTARMENU:     DECF MENU           ;RESTAR 1 AL MENU
            MOVLW 0X00
            SUBWF MENU,0        ; COMPARAR CON 0
            BTFSS STATUS,Z
            GOTO  DECAMENU1     ; SI ES CERO LLEVARLO A 0X05

            MOVLW 0X05
            MOVWF MENU

DECAMENU1:      CLRF DECAMENU;      ; DECAMENU ES CERO YA QUE SOLO HAY 3 OPCIONES
            GOTO BUCLE;



;-----------------------FUNCION=1 AUXILIARES FUNCION1       (VER HORA)
F1_0:           CLRF FUNCION;    FUNCION=0
            GOTO BUCLE

F1_1:           CLRF OPCIONMOS
            BSF OPCIONMOS,0; OPCIONMOS=1  MOSTRAR HORAS Y MINUTOS
            CALL MOSTRAR
            GOTO BUCLE

F1_2:           CLRF OPCIONMOS
            BSF OPCIONMOS,1  ; OPCIONMOS=2  MOSTRAR MINUTOS Y SEGUNDOS
            CALL MOSTRAR
            GOTO BUCLE



;----------------------FUNCION=15 CAMBIARHOR AUXILIAR   (CAMBIAR HORA)

F15VOLVER       CLRF OPCIONMOS;
            MOVLW 0X01
            MOVWF FUNCION   ; FUNCION=1
            BSF INTCON,7    ; SE HABILITAN LAS INTERRUPCIONES
            GOTO BUCLE      ;VOLVER AL BUCLE PRINCIPAL


SUMAHOR:        INCF HORA;      ; INCREMENTAR HORA
            MOVLW 0X02
            SUBWF DECAHORA,0  ; SI DECAHORA ES 2 ENTONCES VERIFICAR QUE NO SE HAYA LLEGADO A LA HORA 24
            BTFSC STATUS,Z
            GOTO  SUMAHOR1    ; SI ES 2 IR A SUMAHOR1

            MOVLW 0X0A        ; SI DECAHORA NO ES 2 ENTONCES VEMOS SI HORA LLEGO A 0X0A
            SUBWF HORA,0
            BTFSS STATUS,Z
            GOTO FINHOR       ; SI NO ES DIEZ NO HAY PROBLEMA

            CLRF HORA         ; SI ES DIEZ SE INCREMENTA DECAHORA Y SE LIMPIA HORA(=0)
            INCF DECAHORA
```

```
            BSF INTCON,7
            GOTO BUCLE

SUMAHOR1:   MOVLW 0X04      ; VERIFICAR QUE HORA NO SEA 4
            SUBWF HORA,0
            BTFSS STATUS,Z
            GOTO FINHOR

            CLRF HORA       ; SI ES 4 ENTONCES SON LAS 24 O SEA LAS 00HS
            CLRF DECAHORA   ; LIMPIAR DECAHORA Y LIMPIAR HORA
            BSF INTCON,7    ; HABILITAR INTERRUPCIONES Y VOLVER A BUCLE
            GOTO BUCLE



RESTAHOR:   DECF HORA       ; SE DECREMENTA HORA
            MOVLW 0XFF      ; SI HORA ES= 0XFF ES PORQUE YA DIO LA VUELTA HAY QUE MOSTRAR 9
            SUBWF HORA,0
            BTFSS STATUS,Z
            GOTO FINHOR     ; SI NO ES 0XFF NO HAY PROBLEMA

            DECF DECAHORA   ; SI ES 0XFF HAY QUE PONER 9 EN HORA Y DECREMENTAR DECAHORA
            MOVLW 0X09
            MOVWF HORA

            MOVLW 0XFF      ; ¿DECAHORA PASO A SER 0XFF?
            SUBWF DECAHORA,0
            BTFSS STATUS,Z
            GOTO FINHOR     ;NO ENTONCES VOLVER

            MOVLW 0X03      ;SI ENTONCES MOSTRAR 23HS=> DECAHORA=2 HORA=3
            MOVWF HORA
            MOVWF DECAHORA
            DECF DECAHORA


FINHOR:     BSF INTCON,7; HABILITAR INTERRUPCIONES
            GOTO BUCLE

;-------------------FUNCION=16   CAMBIARMIN AUXILIAR        (CAMBIAR MINUTOS)
SUMAMIN:    INCF MINUTO         ;INCREMENTAR HORA
            MOVLW 0X0A          ; ¿ HORA ES 0XA?
            SUBWF MINUTO,0
            BTFSS STATUS,Z
            GOTO FINMIN         ;NO: NO HAY PROBLEMA VOLVER A BUCLE

            CLRF MINUTO         ;SI: MINUTO=0 INCREMENTAR DECAMINUTO
            INCF DECAMINUTO

            MOVLW 0X06          ; ¿DECAMINUTO ES 6?
            SUBWF DECAMINUTO,0
            BTFSS STATUS,Z
            GOTO FINMIN         ;NO: VOLVER
            CLRF DECAMINUTO     ;SI: LIMPIAR DECAMINUTO
            GOTO FINMIN

RESTAMIN:   DECF MINUTO         ;DECREMENTAR MINUTO
            MOVLW 0XFF
            SUBWF MINUTO,0      ;ES MINUTO 0XFF? (DIO VUELTA??)
            BTFSS STATUS,Z
            GOTO FINMIN         ;NO: VOLVER

            MOVLW 0X09          ;SI: MINUTO=9 DECREMENTAR DECAMINUTO
            MOVWF MINUTO
            DECF DECAMINUTO

            MOVLW 0XFF
            SUBWF DECAMINUTO,0  ;¿ES DECAMINUTO 0XFF?
            BTFSS STATUS,Z
            GOTO FINMIN         ; NO: VOLVER
            MOVLW 0X05          ; SI: DECAMINUTO=5
            MOVWF DECAMINUTO

FINMIN:     BSF INTCON,7
            GOTO BUCLE
;-----------------------------------------------


;---------------------FUNCION2:                   (CAMBIAR HORA ALARMA)
             ;-----      NOTA: ESTE TRAMO ES IGUAL AL DE CAMBIARHOR CON MODIFICACIONES EN LAS VARIBLES Y LOS LABELS
F2_A:

            MOVLW 0X03
            MOVWF OPCIONMOS; OPCIONMOS=3  MOSTRAR HORAS Y MINUTOS ALARMA
            CALL MOSTRAR
            GOTO BUCLE
```

```
F2VOLVER_A
        CLRF FUNCION   ; FUNCION=0
        GOTO BUCLE     ;VOLVER AL BUCLE PRINCIPAL


SUMAHOR_A:      INCF HORA_A;      ; INCREMENTAR HORA
        MOVLW 0X02
        SUBWF DECAHORA_A,0  ; SI DECAHORA_A ES 2 ENTONCES VERIFICAR QUE NO SE HAYA LLEGADO A LA HORA 24
        BTFSC STATUS,Z
        GOTO  SUMAHOR1_A    ; SI ES 2 IR A SUMAHOR1

        MOVLW 0X0A        ; SI DECAHORA_A NO ES 2 ENTONCES VEMOS SI HORA LLEGO A 0X0A
        SUBWF HORA_A,0
        BTFSS STATUS,Z
        GOTO FINHOR_A       ; SI NO ES DIEZ NO HAY PROBLEMA

        CLRF HORA_A         ; SI ES DIEZ SE INCREMENTA DECAHORA Y SE LIMPIA HORA_A(=0)
        INCF DECAHORA_A
        GOTO BUCLE

SUMAHOR1_A:       MOVLW 0X04      ; VERIFICAR QUE HORA_A NO SEA 4
        SUBWF HORA_A,0
        BTFSS STATUS,Z
        GOTO FINHOR_A

        CLRF HORA_A         ; SI ES 4 ENTONCES SON LAS 24 O SEA LAS 00HS
        CLRF DECAHORA_A     ; LIMPIAR DECAHORA_A Y LIMPIAR HORA_A
        GOTO BUCLE



RESTAHOR_A:     DECF HORA_A         ; SE DECREMENTA HORA_A
        MOVLW 0XFF        ; SI HORA ES= 0XFF ES PORQUE YA DIO LA VUELTA HAY QUE MOSTRAR 9
        SUBWF HORA_A,0
        BTFSS STATUS,Z
        GOTO FINHOR_A       ; SI NO ES 0XFF NO HAY PROBLEMA

        DECF DECAHORA_A     ; SI ES 0XFF HAY QUE PONER 9 EN HORA Y DECREMENTAR DECAHORA_A
        MOVLW 0X09
        MOVWF HORA_A

        MOVLW 0XFF        ; ¿DECAHORA_A PASO A SER 0XFF?
        SUBWF DECAHORA_A,0
        BTFSS STATUS,Z
        GOTO FINHOR_A         ;NO ENTONCES VOLVER

        MOVLW 0X03          ;SI ENTONCES MOSTRAR 23HS=> DECAHORA_A=2 HORA_A=3
        MOVWF HORA_A
        MOVWF DECAHORA_A
        DECF DECAHORA_A


FINHOR_A:       GOTO BUCLE
;--------------------FUNCION=17     (CAMBIARMIN_A PARA ALARMA) AUXILIAR        (CAMBIAR MINUTO ALARMA)
SUMAMIN_A:      INCF MINUTO_A         ;INCREMENTAR MINUTO
        MOVLW 0X0A          ; ¿ MINUTO ES 0XA?
        SUBWF MINUTO_A,0
        BTFSS STATUS,Z
        GOTO FINMIN_A           ;NO: NO HAY PROBLEMA VOLVER A BUCLE

        CLRF MINUTO_A           ;SI: MINUTO_A=0 INCREMENTAR DECAMINUTO
        INCF DECAMINUTO_A

        MOVLW 0X06          ; ¿DECAMINUTO_A ES 6?
        SUBWF DECAMINUTO_A,0
        BTFSS STATUS,Z
        GOTO FINMIN_A           ;NO: VOLVER
        CLRF DECAMINUTO_A       ;SI: LIMPIAR DECAMINUTO_A
        GOTO FINMIN_A

RESTAMIN_A:       DECF MINUTO_A           ;DECREMENTAR MINUTO_A
        MOVLW 0XFF
        SUBWF MINUTO_A,0        ;ES MINUTO_A 0XFF? (DIO VUELTA??)
        BTFSS STATUS,Z
        GOTO FINMIN_A           ;NO: VOLVER

        MOVLW 0X09          ;SI: MINUTO_A=9 DECREMENTAR DECAMINUTO_A
        MOVWF MINUTO_A
        DECF DECAMINUTO_A

        MOVLW 0XFF
        SUBWF DECAMINUTO_A,0    ;¿ES DECAMINUTO_A 0XFF?
        BTFSS STATUS,Z
        GOTO FINMIN_A           ; NO: VOLVER
        MOVLW 0X05          ; SI: DECAMINUTO_A=5
        MOVWF DECAMINUTO_A
```

```
FINMIN_A:       GOTO BUCLE
;-------------------------------------------------

;---------------------------AUXILIAR FUNCION3  (MOSTRAR TENSION)

F3COMPLETO:     MOVLW UNIDAD
            MOVWF PUNTERO_3
            GOTO BUCLE;




;---------------------------AUXILIAR FUNCION4 (CAMBIAR DESIMA_T Y UNIDAD_T)


F4VOLVER:         CLRF FUNCION
            GOTO BUCLE


RESTARDEC_T:      DECF DESIMA_T   ; DECREMENTAR DESIMA_T

            MOVLW 0XFF
            SUBWF DESIMA_T,0 ; CUANDO DESIMA_T LLEGUE A 0XFF
            BTFSS STATUS,Z  ; HAY QUE PONER 9 EN DESIMA_T Y DECREMENTAR UNIDAD_T
            GOTO BUCLE
            MOVLW 0X09
            MOVWF DESIMA_T

            ;DECF UNIDAD_T
            ;MOVLW 0XFF
            ;SUBWF UNIDAD_T,0
            ;BTFSS STATUS,Z
            ;GOTO BUCLE     ; CUANDO UNIDAD_T LLEGUE A 0XFF HAY QUE PONER 4 EN UNIDAD_T
            ;MOVLW 0X04
            ;MOVWF UNIDAD_T
            GOTO BUCLE


SUMARDEC_T:       INCF DESIMA_T   ; DECREMENTAR DESIMA_T

            MOVLW 0X0A
            SUBWF DESIMA_T,0  ; CUANDO DESIMA_T LLEGUE A 0X0A
            BTFSS STATUS,Z  ; HAY QUE PONER 0 EN DESIMA_T E INCREMENTAR UNIDAD_T
            GOTO BUCLE
            CLRF DESIMA_T


                  ;DESHABILITADO YA QUE SE PUSO MÁXIMO DE 0,99 mV EQUIVALENTE A 99ºC
            ;INCF UNIDAD_T
            ;MOVLW 0X05
            ;SUBWF UNIDAD_T,0
            ;BTFSS STATUS,Z
            ;GOTO BUCLE     ; CUANDO UNIDAD_T LLEGUE A 0X05 HAY QUE PONER 0 EN UNIDAD_T
            ;CLRF UNIDAD_T

           ; UNIDAD_T NO INCREMENTA YA QUE SE MEDIRÁ DE 0 A 99 ADEMÁS LA TEMPERATURA MAXIMA DEBERÍA MANTENERSE A 100ºC
            GOTO BUCLE
;-----------------------------------AUXILIAR FUNCION 21   (CAMBIAR CENTESIMA_T)
RESTARCEN_T:      DECF CENTESIMA_T   ; DECREMENTAR DESIMA_T

            MOVLW 0XFF
            SUBWF CENTESIMA_T,0  ; CUANDO DESIMA_T LLEGUE A 0XFF
            BTFSS STATUS,Z  ; HAY QUE PONER 9 EN DESIMA_T Y DECREMENTAR UNIDAD_T
            GOTO BUCLE
            MOVLW 0X09
            MOVWF CENTESIMA_T
            GOTO BUCLE


SUMARCEN_T:       INCF CENTESIMA_T   ; DECREMENTAR DESIMA_T

            MOVLW 0X0A
            SUBWF CENTESIMA_T,0  ; CUANDO DESIMA_T LLEGUE A 0X0A
            BTFSS STATUS,Z  ; HAY QUE PONER 0 EN DESIMA_T E INCREMENTAR UNIDAD_T
            GOTO BUCLE
            CLRF CENTESIMA_T
            GOTO BUCLE



;---------------------------AUXILIAR FUNCION18 (CAMBIAR DESIMA_A Y UNIDAD_A)   PARA ALARMA RELOJ
```

```
F18VOLVER:
            MOVLW .2
            MOVWF FUNCION
            CLRF OPCIONMOS
            GOTO BUCLE


RESTARDEC_A:      DECF DESIMA_A   ; DECREMENTAR DESIMA_T

            MOVLW 0XFF
            SUBWF DESIMA_A,0 ; CUANDO DESIMA_T LLEGUE A 0XFF
            BTFSS STATUS,Z  ; HAY QUE PONER 9 EN DESIMA_T Y DECREMENTAR UNIDAD_T
            GOTO BUCLE
            MOVLW 0X09
            MOVWF DESIMA_A


                  ;DESHABILITADO YA QUE SE PUSO MÁXIMO DE 0,99 mV EQUIVALENTE A 99ºC
            ;DECF UNIDAD_A
            ;MOVLW 0XFF
            ;SUBWF UNIDAD_A,0
            ;BTFSS STATUS,Z
            ;GOTO BUCLE     ; CUANDO UNIDAD_T LLEGUE A 0XFF HAY QUE PONER 4 EN UNIDAD_T
            ;MOVLW 0X04
            ;MOVWF UNIDAD_A
                  ; UNIDAD_T NO INCREMENTA YA QUE SE MEDIRÁ DE 0 A 99 ADEMÁS LA TEMPERATURA MAXIMA DEBERÍA MANTENERSE A 100ºC
            GOTO BUCLE


SUMARDEC_A:       INCF DESIMA_A   ; DECREMENTAR DESIMA_T

            MOVLW 0X0A
            SUBWF DESIMA_A,0  ; CUANDO DESIMA_T LLEGUE A 0X0A
            BTFSS STATUS,Z  ; HAY QUE PONER 0 EN DESIMA_T E INCREMENTAR UNIDAD_T
            GOTO BUCLE
            CLRF DESIMA_A

            ;INCF UNIDAD_A
            ;MOVLW 0X05
            ;SUBWF UNIDAD_A,0
            ;BTFSS STATUS,Z
            ;GOTO BUCLE     ; CUANDO UNIDAD_T LLEGUE A 0X05 HAY QUE PONER 0 EN UNIDAD_T
            ;CLRF UNIDAD_A
            GOTO BUCLE
;-----------------------------------AUXILIAR FUNCION 19   (CAMBIAR CENTESIMA_A)   PARA ALARMA RELOJ
RESTARCEN_A:      DECF CENTESIMA_A   ; DECREMENTAR DESIMA_T

            MOVLW 0XFF
            SUBWF CENTESIMA_A,0  ; CUANDO DESIMA_T LLEGUE A 0XFF
            BTFSS STATUS,Z  ; HAY QUE PONER 9 EN DESIMA_T Y DECREMENTAR UNIDAD_T
            GOTO BUCLE
            MOVLW 0X09
            MOVWF CENTESIMA_A
            GOTO BUCLE


SUMARCEN_A:       INCF CENTESIMA_A   ; DECREMENTAR DESIMA_T

            MOVLW 0X0A
            SUBWF CENTESIMA_A,0  ; CUANDO DESIMA_T LLEGUE A 0X0A
            BTFSS STATUS,Z  ; HAY QUE PONER 0 EN DESIMA_T E INCREMENTAR UNIDAD_T
            GOTO BUCLE
            CLRF CENTESIMA_A
            GOTO BUCLE

;-------------------------------------------------------------------------------




           GOTO BUCLE;




;------------------FUNCION DIVD_16
      ; ESTA FUNCIÓN DIVIDE UN NÚMERO DE 2 BYTES POR .16 QUE EN HEXADECIAMAL SE ESCRIBE 0X10.
      ; AL IGUAL QUE EN EL SISTEMA DECIMAL, DIVIDIR POR LA BASE OCACIONA EL CORRIMIENTO DE LOS DÍGITOS A LA DERECHA.



DIVD_16:      CLRF DIVDRTAH
          CLRF DIVDRTAL
```

```
        MOVF NUMDL,0
        SWAPF NUMDH,0
        MOVWF DIVDRTAH  ;  DIVDRTAH=SWAPF (NUMDH)
        ANDLW B'11110000'
        MOVWF DIVDRTAL  ;   DIVDRTAL=(DIVDRTAH) AND(B'11110000')
        MOVLW B'00001111'
        ANDWF DIVDRTAH,1; DIVDRTAH=(DIVDRTAH) AND(B'00001111')
        SWAPF NUMDL,0
        ANDLW B'00001111'
        ADDWF DIVDRTAL   ; DIVDRTAL=DIVDRTAL+[SWAPF(NUMDL)] AND(B'00001111')
        RETURN




;*****************************UNA ACCION IMPORTANTE!!!! CONVERSION ANALÓGICA A DIGITAL*******************************
CONVERSION:
        BTFSC CENTISEGUNDO,2; ESTE BIT SE PRENDE Y SE APAGA UNA VEZ CADA 8 MSEG ESTO SERÍA UNA FRECUENCIA DE 12.5 HZ
                        ;( EN REALIDAD SOLO HAY 12 CONVERSIONES POR SEGUNDO)
        GOTO CONVERTIR    ; SI BIT 2 DE CENTISEGUNDO ESTA PRENDIDO IR A CONVERTIR


        BTFSC BITFLAGS,7  ; ¿¿BIT 7=OFF??
        BCF BITFLAGS,7      ; NO:  APAGAR BITFLAGS,7
        GOTO VERALARMA      ; SI:VOLVER


CONVERTIR:     BTFSC BITFLAGS,7 ;  ¿BIT7=OFF?
        GOTO VERALARMA   ; NO: ESTA PRENDIDO SI YA SE REALIZO LA CONVERSION ENTONCES SE VUELVE

        BSF BITFLAGS,7  ; SI: HAY QUE PRENDERLO Y REALIZAR LA CONVERSION


        BSF ADCON0,2;  PEDIR CONVERSION!!!!!!<<<<<<<<<<<<<<<<<<<<




PROMEDIARAD:
                        ;AL PROMEDIO TOTAL SE LE SACA UN 10% Y SE LE AGREGA LA NUEVA LECTURA DIVIDIDA 10
                        ; O SEA:
                        ; ADPROM[K]=ADPROM[K-1]-ADPROM[K-1]/10 +ADRES[K]/10

                          ;PASO1:   RESTARLE 1/16 AL PROMEDIO

        MOVLW 0X0A   ; NUM2=.10 QUE ES USADO COMO DIVISOR DE 1 BYTE
        MOVWF NUM2
        MOVF ADPROMH,0
        MOVWF NUMDH
        MOVF ADPROML,0
        MOVWF NUMDL


        CALL DIVD_16
                        ; (RESTA DE DOBLE-BYTE)
        MOVF DIVDRTAL,0;
        SUBWF ADPROML,1; ADPROML=ADPROML-DIVDRTAL
        BTFSS STATUS,C; SI CARRIER ES 0 ES PORQUE ADPROML<DIVDRTAL HAY QUE SACARLE 1 A ADPROMH
        DECF ADPROMH;
        MOVF DIVDRTAH,0;
        SUBWF ADPROMH,1; ADPROMH= ADPROMH-DIVDRTAH
                        ; POR SER DIVRTA(H:L) EL 10% DE ADPROM(H:L)  LA RESTA DA UN NUMERO POSITIVO

ESPERAR_AD:    BTFSC ADCON0,2; LA CONVERSION FUE PEDIDA CON ANTERIORIDAD YA DEBERÍA HABERSE LOGRADO.
        GOTO ESPERAR_AD


                        ;PASO2:  LEER REGISTROS DE CONVERSION Y DIVIDIRLOS EN 16

        BSF STATUS, RP0 ; (IR AL BANCO 1)
        MOVF ADRESL,0;
        BCF STATUS, RP0 ; (VOLVER AL BANCO 0)

        ANDLW B'11000000'; SE FILTRA POR LAS DUDAS, SOLO SON UTILES LOS BITS 7 Y 6
        MOVWF NUMDL

        MOVF ADRESH,0
        MOVWF NUMDH

        CALL DIVD_16

                        ;PASO3: SUMARSELO AL PROMEDIO
                         ;(SUMA DOBLE-BYTE)
```

```
        MOVF DIVDRTAL,0;
        ADDWF ADPROML,1; ADPROML= ADPROML+DIVDRTAL
        BTFSC STATUS,C; SI HAY CARRIER HAY QUE SUMARLE 1 A
        INCF ADPROMH;
        MOVF DIVDRTAH,0;
        ADDWF ADPROMH,1; ADPROMH= ADPROMH+DIVDRTAH



LEER_AD:

        CLRF UC; LIMPIAR UNIDAD DE CONVERSOR
        CLRF DCV; LIMPIAR DECIMA DE CONVERSOR
        CLRF CC; LIMPIAR CENTESIMA DE CONVERSOR
        CLRF MC; LIMPIAR MILESIMA DE CONVERSOR
        CLRF DMC; LIMPIAR DECIMA DE MILESIMA DE CONVERSOR
        CLRF CMC; LIMPIAR CENTESIMA DE MILESIMA DE CONVERSOR



BIT9:   BTFSS ADPROMH,7
        GOTO BIT8

        MOVLW .2
        ADDWF UC,1

        MOVLW .5
        ADDWF DCV,1


BIT8:   BTFSS ADPROMH,6
        GOTO BIT7

        MOVLW .1
        ADDWF UC,1

        MOVLW .2
        ADDWF DCV,1

        MOVLW .5
        ADDWF CC



BIT7:
        BTFSS ADPROMH,5
        GOTO BIT6

        MOVLW .6
        ADDWF DCV,1

        MOVLW .2
        ADDWF CC

        MOVLW .5
        ADDWF MC,1


BIT6:
        BTFSS ADPROMH,4
        GOTO BIT5


        MOVLW .3
        ADDWF DCV,1

        MOVLW .1
        ADDWF CC

        MOVLW .2
        ADDWF MC,1

        MOVLW .5
        ADDWF DMC,1



BIT5:   BTFSS ADPROMH,3
        GOTO BIT4
```

```
        MOVLW .1
        ADDWF DCV,1

        MOVLW .5
        ADDWF CC

        MOVLW .6
        ADDWF MC,1

        MOVLW .2
        ADDWF DMC,1

        MOVLW .5
        ADDWF CMC,1


BIT4:   BTFSS ADPROMH,2
        GOTO BIT3


        MOVLW .7
        ADDWF CC

        MOVLW .8
        ADDWF MC,1

        MOVLW .1
        ADDWF DMC,1

        MOVLW .2
        ADDWF CMC,1


BIT3:   BTFSS ADPROMH,1
        GOTO BIT2


        MOVLW .3
        ADDWF CC

        MOVLW .9
        ADDWF MC,1

        MOVLW .0
        ADDWF DMC,1

        MOVLW .6
        ADDWF CMC,1

BIT2:   BTFSS ADPROMH,0
        GOTO BIT1


        MOVLW .1
        ADDWF CC

        MOVLW .9
        ADDWF MC,1

        MOVLW .5
        ADDWF DMC,1

        MOVLW .3
        ADDWF CMC,1


BIT1:   BTFSS ADPROML,7
        GOTO BIT0


        MOVLW .9
        ADDWF MC,1

        MOVLW .7
        ADDWF DMC,1
```

```
        MOVLW .7; (SE HA REDONDEADO PARA ARRIBA)
        ADDWF CMC,1



BIT0:       BTFSS ADPROML,6
        GOTO  DIGITALIZAR


        MOVLW .4
        ADDWF MC,1

        MOVLW .8
        ADDWF DMC,1

        MOVLW .8
        ADDWF CMC,1


DIGITALIZAR:            ; EL DIVISOR ES NUM2 QUE FUE PREVIAMENTE CARGADO CON .10
        MOVF CMC,0
        MOVWF NUM1; NUM1=CMC
        CALL DIV    ; NUM1/NUM2=W O SEA W=CMC/10  NUM1=RESTO
        ADDWF DMC,1  ; DMC= DMC+CMC/10
        MOVF NUM1,0
        MOVWF CMC    ; CMC=RESTO

        MOVF DMC,0
        MOVWF NUM1; NUM1=DMC
        CALL DIV    ; NUM1/NUM2=W O SEA W=DMC/10  NUM1=RESTO
        ADDWF MC,1  ; MC= MC+DMC/10
        MOVF NUM1,0
        MOVWF DMC    ; DMC=RESTO

        MOVF MC,0
        MOVWF NUM1; NUM1=CMC
        CALL DIV    ; NUM1/NUM2=W O SEA W=MC/10  NUM1=RESTO
        ADDWF CC,1  ; CC= CC+MC/10
        MOVF NUM1,0
        MOVWF MC    ; MC=RESTO

        MOVF CC,0
        MOVWF NUM1; NUM1=CMC
        CALL DIV    ; NUM1/NUM2=W O SEA W=CC/10  NUM1=RESTO
        ADDWF DCV,1  ; DCV= DCV+CC/10
        MOVF NUM1,0
        MOVWF CC    ; CC=RESTO

        MOVF DCV,0
        MOVWF NUM1; NUM1=CMC
        CALL DIV    ; NUM1/NUM2=W O SEA W=DCV/10  NUM1=RESTO
        ADDWF UC,1  ; UC= UC+DCV/10
        MOVF NUM1,0
        MOVWF DCV    ; DCV=RESTO


TRANSFERIR:
        MOVF UC,0;      UNIDAD=UC //DESIMA=DCV // CENTESIMA=CC // MILESIMA=MC
        MOVWF UNIDAD;
        MOVF DCV,0;
        MOVWF DESIMA;
        MOVF CC,0;
        MOVWF CENTESIMA;
        MOVF MC,0;
        MOVWF MILESIMA;


TEST_T:     BTFSS ALARMA_T,0    ;¿ALARMA_T = 1?
          GOTO VERALARMA      ; NO

          MOVF UNIDAD_T,0     ;SI
          SUBWF UNIDAD,0
          BTFSS STATUS,C;  ¿UNIDAD>=UNIDAD_T?
          GOTO VERALARMA ;  NO
          BTFSS STATUS,Z;   ¿UNIDAD=UNIDAD_T?
          GOTO ACT_ALARMA_T  ;NO: ENTONCES UNIDAD>UNIDAD_T ACTIVAR ALARMA

          MOVF DESIMA_T,0 ;SI: UNIDAD=UNIDAD_T
          SUBWF DESIMA,0
          BTFSS STATUS,C;  ¿DESIMA>=CENTESIMA_T?
          GOTO VERALARMA
          BTFSS STATUS,Z;  ¿DESIMA =DESIMA_T?
          GOTO ACT_ALARMA_T ;NO
```

```
            MOVF CENTESIMA_T,0 ;SI
            SUBWF CENTESIMA,0
            BTFSS STATUS,C;  ¿CENTESIMA>=CENTESIMA_T?
            GOTO VERALARMA; NO
ACT_ALARMA_T:
            BSF ALARMA_T,7 ;SI




          GOTO VERALARMA

;*************************************
;*************************************ACCIONES DE COMANDO DE DISPOSITIVOS EXTERNOS******************
                ; **FUNCIONES RELACIONADAS:
                ; "BUCLE"//"ALARMA_TENSION"(FUNCION=26)//"ALARMA"(FUNCION=25)//ERRORESPROC

                ; **VARIABLES USADAS:
                ;ENTRADAS:
                ; SENSO/ BIT7= LUZ NIVEL //BIT6= SENSOR BOYA//BIT5=SENSOR CERRADO //BIT4=SENSOR ABIERTO
                ;SALIDAS:
                ; PORTA:  BIT5=MOTOR VALVULA// BIT3=CALENTADOR//BIT2=LINTERNA//  ADCON0,0
                ;ENTRADAS/SALIDAS:
                ; PROCESO//CRONO//CRONOPROC//ERRORNUM//FUNCION//UNIDAD_C//DESIMA_C//CENTESIMA_C
                ;CONTADOR_C//AUXILIAR_C

                ;**POSIBILIDAD A SALTOS EXTERNOS:
                ;   "VERCONVERSOR"(EN BUCLE PRINCIPAL)

                ;COMENTARIO: ES UNA FUNCION IMPORTANTE POR ESO ES LLAMADA EN CADA BUCLE DEL BUCLE PRINCIPAL

            ORG 0X700
PROCESO_ACTUAL:

            MOVLW 0X07
            MOVWF PCLATH

            MOVF PROCESO,0

            ADDWF PCL,1
            GOTO CERRARVALVULA; PROCESO=0
            GOTO NOPROCESO; PROCESO=1  (SI NO HAY QUE REALIZAR PROCESO DEBE ESTAR EN "1")
            GOTO ABRIRVALVULA; PROCESO=2
            GOTO CALENTAR;    PROCESO=3
            GOTO PRENDERLAMP; PROCESO=4
            ;GOTO APAGARCALOR   EL APAGADO SE HACE DIRECTAMENTE AL SELECCIONAR OTRO PROCESO O EN UNA FUNCION
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP

            GOTO CERRARVALVULA; DEFAULT

;-----------------------------------PROCESO=0  CERRAR VALVULA

CERRARVALVULA:
            BCF ADCON0,0; APAGAR CONVERSIONES
            BCF PORTA,2; APAGAR LINTERNA
            BCF PORTA,3; FORZAR EL CIERRE DEL CALENTADOR (SI TODO ESTA BIEN EL CALENTADOR YA DEBERÍA HABERSE APAGADO ANTES)

            MOVLW 0X00
            SUBWF CRONOPROC,0 ;  ¿CRONOPROCESO=0X00?
            BTFSC STATUS,Z
            GOTO CONTAR0; SI

            CLRF CRONO;   ;N0: RESETEAR CRONOMETRO PARA PROCESO ACTUAL
            CLRF CRONOPROC; CRONOPROCESO= NUMERO DE PROCESO ACTUAL



CONTAR0:       MOVLW .40; 40/4= 10 SEGUNDOS
            SUBWF CRONO,0
            BTFSC STATUS,C; ¿CRONO>=.40?
            GOTO ERROR_01  ; SI

                        ;NO...



            BTFSC SENSO,5;  ¿SENSO[BIT5]?
            GOTO CERRARYA;   BIT 5=1 => NO SE DETECTA QUE LA VALVULA ESTE CERRADA. SE ASUME QUE ESTA ABIERTA
                        ;  BIT5=0 LA VALVULA ESTA CERRADA...
```

```
        BCF PORTA,5; APAGAR MOTOR DE VALVULAS O FORZAR APAGADO.  NO SE DEBE ABRIR LA VALVULA SI EL VASO ESTA LLENO!!
        MOVLW 0X01;
        MOVWF PROCESO; TERMINAR PROCESO

        MOVLW 0XFF    ; PONIENDO UN NUMERO NO USADO EN OPCIONMOS HACE QUE CUALQUIER FUNCIÓN INICIALICE X PRIMERA VEZ
        MOVWF OPCIONMOS; ACTUALIZAR FUNCION ACTUAL PARA QUE VUELVA A ACTIVAR ADCON0,0 POR EJEMPLO
        GOTO VERCONVERSOR; VOLVER


CERRARYA:   BSF PORTA,5; ENCENDER MOTOR DE VALVULA(ES UN MOTOR SINCRONICO QUE GIRA EN ALGUNA SENTIDO HASTA CHOCAR
                    ;CON ALGUN TOPE A LO QUE INVIERTE SU SENTIDO)
        GOTO VERCONVERSOR


;-----------------------------------PROCESO=1   NO HAY PROCESO ACTIVO

NOPROCESO:  CLRF CRONOPROC;
        INCF CRONOPROC;
        BCF PORTA,3; FORZAR EL CIERRE DEL CALENTADOR



        BTFSS SENSO,5;  ¿SENSO[BIT5]?
        GOTO VERCONVERSOR;   BIT5=0 LA VALVULA ESTA CERRADA...

        CLRF PROCESO; PROCESO=0 CERRAR VALVULA
        GOTO VERCONVERSOR;


;-----------------------------------PROCESO=2  ABRIR VALVULA


ABRIRVALVULA:
        BTFSC SENSO,7;  SI EL RECIPIENTE ESTA LLENO HAY QUE CERRAR LA VALVULA.
        GOTO VERCRONO02 ; AÚN ESTA VACÍO
        CLRF PROCESO;  ESTA LLENO HAY QUE CERRAR VALVULA
        GOTO VERCONVERSOR




VERCRONO02:     MOVLW 0X02
        SUBWF CRONOPROC,0 ;  ¿CRONOPROCESO=0X02?
        BTFSC STATUS,Z
        GOTO CONTAR2; SI

        CLRF CRONO;   ;N0: RESETEAR CRONOMETRO PARA PROCESO ACTUAL
        MOVLW 0X02
        MOVWF CRONOPROC; CRONOPROCESO= NUMERO DE PROCESO ACTUAL


CONTAR2:    MOVLW .180; 180/4= 45 SEGUNDOS
        SUBWF CRONO,0
        BTFSC STATUS,C; ¿CRONO>=.100?
        GOTO ERROR_02  ; SI


                ;NO...



        BTFSC SENSO,4;  ¿SENSO[BIT4]?
        GOTO ABRIRYA;   BIT 4=1 => NO SE DETECTA QUE LA VALVULA ESTE ABIERTA. SE ASUME QUE ESTA CERRADA
                ;  BIT4=0 LA VALVULA ESTA ABIERTA...
        BCF PORTA,5; APAGAR MOTOR DE VALVULAS O FORZAR APAGADO.  NO SE DEBE ABRIR LA VALVULA SI EL VASO ESTA LLENO!!


        GOTO VERCONVERSOR; VOLVER


ABRIRYA:
        BTFSC SENSO,5;    ¿SENSO[BIT5]?
        GOTO ABRIRYA1     ;BIT=1//NO

                ;SI:BIT=0
        MOVLW .16; 16/4= 4 SEGUNDOS
        SUBWF CRONO,0
        BTFSC STATUS,C; ¿CRONO>=.12?
        GOTO ERROR_03  ; SI
                ;NO
ABRIRYA1:   BSF PORTA,5; ENCENDER MOTOR DE VALVULA(ES UN MOTOR SINCRONICO QUE GIRA EN ALGUNA SENTIDO
                ;HASTA CHOCAR CON ALGUN TOPE A LO QUE INVIERTE SU SENTIDO)
        GOTO VERCONVERSOR
```

```
;-----------------------------PROCESO=3 CALENTAR



CALENTAR:       BTFSC SENSO,5   ; SI LA VALVULA *NO ESTA CERRADA*(ABIERTA) NO TIENE QUE CALENTAR
                GOTO ERROR_04
                BTFSS SENSO,6   ; SI EL CONTENEDOR PRINCIPAL ESTA VACÍO, SE ACTIVA EL SENSOR Y NO SE DEBE CALENTAR
                GOTO ERROR_06


                MOVLW 0X03
                SUBWF CRONOPROC,0 ;  ¿CRONOPROCESO=0X00?
                BTFSC STATUS,Z
                GOTO CONTAR3; SI

                MOVLW .60; N0: RESETEAR VARIABLES PARA PROCESO ACTUAL  (se da 180-60=120 => 120/4=30segundos de precalentamiento
                        ; por experimento se ve que al principio cuesta mas variar la temperatura)
                MOVWF CRONO;   SE PREPARA CRONO 30 SEGUNDOS ANTES DE QUE HAGA  PASAR A CONTADOR_C A CERO Y HAGA
                                ;REFRESCAR LAS VARIABLES POR PRIMERA VEZ

                CLRF CONTADOR_C; RESETEAR CONTADOR_C
                DECF CONTADOR_C;


                CLRF UNIDAD_C
                CLRF DESIMA_C
                CLRF CENTESIMA_C
                MOVLW 0X03
                MOVWF  CRONOPROC; CRONOPROCESO= NUMERO DE PROCESO ACTUAL


CONTAR3:        MOVLW .180; (.180; 180/4= 45 SEGUNDO)
                SUBWF CRONO,0
                BTFSS STATUS,C; ¿CRONO>=CUMPLIO .45 SEG
                GOTO CALENTAR_YA;NO
                INCF CONTADOR_C;SI
                CLRF CRONO;

                        ;HA PASADO 45 SEGUNDOS SI TODO MARCHA BIEN ENTONCES EL VALOR FORMADO  CON UNIDAD,
                         ; DESIMA Y CENTESIMA TENDRÍA QUE HABER INCREMENTADO
                MOVF UNIDAD_C,0
                SUBWF UNIDAD,0;       ¿UNIDAD>=UNIDAD_C ?
                BTFSS STATUS,C;
                GOTO  ERROR_00  ; NO: ERROR
                BTFSS STATUS,Z;       ¿UNIDAD=UNIDAD_C?
                GOTO REFRESCAR_C; : entonces si no es menor ni es igual: es mayor

                MOVF DESIMA_C,0       ; SI
                SUBWF DESIMA,0;     ¿DESIMA>=DESIMA_C  ?
                BTFSS STATUS,C;
                GOTO ERROR_00;   NO
                BTFSS STATUS,Z;     ¿DESIMA=DESIMA_C?
                GOTO  REFRESCAR_C  ;  NO: entonces es mayor

                MOVF CENTESIMA,0
                SUBWF CENTESIMA_C,0;    ¿CENTESIMA=<CENTESIMA_C?
                BTFSC STATUS,C;
                GOTO ERROR_00  ;   SI


REFRESCAR_C:    MOVF UNIDAD,0
                MOVWF UNIDAD_C;       UNIDAD_C=UNIDAD
                MOVF DESIMA,0
                MOVWF DESIMA_C;     DESIMA_C=  DESIMA
                MOVF CENTESIMA,0
                MOVWF CENTESIMA_C;   CENTESIMA_C= CENTESIMA

                INCF CENTESIMA_C;  SE SUMA 1 A CENTESIMA_C PARA QUE EL PROXIMO VALOR TENGA QUE SER SI O SI MAYOR AL ACTUAL
                MOVLW 0X0A
                SUBWF CENTESIMA_C,0;  CENTESIMA_C>=0X0A??
                BTFSS STATUS, C
                GOTO CALENTAR_YA;  NO

                CLRF CENTESIMA_C  ;SI...
                INCF DESIMA_C;
                MOVLW 0X0A
                SUBWF DESIMA_C,0;  DESIMA_C>=0X0A??
                BTFSS STATUS, C
                GOTO CALENTAR_YA;  NO

                MOVLW 0X09  ; SI: SI LLEGA A 99 SE DEBE MANTENERSE, YA QUE ES EL MÁXIMO VALOR
                MOVWF DESIMA_C
```

```
            MOVWF CENTESIMA_C


CALENTAR_YA:    MOVLW 0XFF;
            SUBWF CONTADOR_C,0;
            BTFSC STATUS,Z;¿CONTADOR_C=0XFF?(CONDICION INICIAL)
            GOTO FIN_C ; SI
                        ;NO
            MOVLW .12;
            SUBWF CONTADOR_C,0;  CUANDO EL CONTADOR LLEGA A .12 HABRÁN PASADO 9 MINUTOS
            BTFSC STATUS,C; ¿CONTADOR_C>=.12?
            GOTO ERROR_05


FIN_C:        BSF PORTA,3; ENCENDER CALENTADOR
            GOTO VERCONVERSOR


           ; NOTA: PARA CALENTAR SE UTILIZA LA FUNCION ALARMATENSION(FUNCION=26)

;-----------------------------------PROCESO=4  PRENDER LAMPARA PARA DETECTAR NIVEL DE LÍQUIDO EN RECIPIENTE MEDIANTE FOTODIODO

PRENDERLAMP:
            BCF PORTA,3; FORZAR EL CIERRE DEL CALENTADOR (SI TODO ESTA BIEN EL CALENTADOR YA DEBERÍA HABERSE APAGADO ANTES)
            BSF PORTA,2; PRENDER LINTERNA!!

            MOVLW 0X04
            SUBWF CRONOPROC,0 ;  ¿CRONOPROCESO=0X04?
            BTFSC STATUS,Z
            GOTO CONTAR4; SI

            CLRF CRONO;   ;N0: RESETEAR CRONOMETRO PARA PROCESO ACTUAL

            MOVLW 0X04
            MOVWF CRONOPROC; CRONOPROC= PROCESO ACTUAL



CONTAR4:       MOVLW .1; 1/4= 1/4 SEGUNDO  (SE DA UN TIEMPO ANTES DE ABRIR PARA QUE EL FOTODIODO DETECTE LA LUZ CORRECTAMENTE)
            SUBWF CRONO,0
            BTFSS STATUS,C; ¿CRONO>=.2?
            GOTO VERCONVERSOR;  NO: ESPERAR

            MOVLW 0X02; SI: PROCESO=2 (LISTO PARA ABRIR VALVULA ENTONCES IR A ABRIR VALVULA)
            MOVWF PROCESO;
            GOTO VERCONVERSOR;




;*****************************************ERRORES
ERROR_00:    ; ERROR FATAL!! SE ESTA CALENTANDO Y NO HAY VARIACION DE TEMPERATURA POR MAS DE 45 SEGUNDOS;

        BCF PORTA,3; PARAR DE CALENTAR
        CLRF PROCESO; PROCESO=0X00 CERRARVALVULA
                ; ANTE CUALQUIER ERROR SE BUSCA QUE LA VALVULA ESTE CERRADA Y EL CALENTADOR APAGADO
        MOVLW 0X00
        MOVWF ERRORNUM
        MOVLW .30
        MOVWF FUNCION
        GOTO VERCONVERSOR


ERROR_01:   MOVLW 0X01; ERROR FATAL!! LA VALVULA ESTA SIN CERRAR POR MAS DE 10 SEGUNDOS Y EL NIVEL LLEGO A SU LIMITE
        MOVWF ERRORNUM
        MOVLW .30
        MOVWF FUNCION
        CLRF CRONO
        GOTO VERCONVERSOR; VOLVER AL BUCLE PRINCIPAL



ERROR_02:       ; ERROR!!! EL PROCESO DE ABRIR VALVULA  SIGUIÓ ACTIVADO POR MAS DE 45 SEGUNDOS; ES POSIBLE QUE NO HAYA RECIPIENTE
        BCF PORTA,5; PARAR MOTOR
        CLRF PROCESO; PROCESO=0X00 CERRARVALVULA
        MOVLW 0X02
        MOVWF ERRORNUM
        MOVLW .30
        MOVWF FUNCION
        GOTO VERCONVERSOR


ERROR_03:      ; ERROR !!! SE HA ENERGIZADO EL MOTOR PERO LA VALVULA ESTA SIN ABRIR POR MAS DE 4 SEGUNDOS
        BCF PORTA,5; PARAR MOTOR
        CLRF PROCESO; PROCESO=0X00 CERRARVALVULA
        MOVLW 0X03
```

```
        MOVWF ERRORNUM
        MOVLW .30
        MOVWF FUNCION
        GOTO VERCONVERSOR


ERROR_04    ;PELIGRO!!! SE HA INTENTADO CALENTAR CON LA VALVULA ABIERTA
        BCF PORTA,3; FORZAR LA PARADA DEL CALENTADOR
        CLRF PROCESO; PROCESO=0X00 CERRARVALVULA
        MOVLW 0X04
        MOVWF ERRORNUM
        MOVLW .30
        MOVWF FUNCION
        GOTO VERCONVERSOR

ERROR_05    ; ERROR GRAVE!! HA PASADO MAS DE 9 MINUTOS Y EL PROCESO DE CALENTAR NO HA TERMINADO
        BCF PORTA,3; PARAR DE CALENTAR
        CLRF PROCESO; PROCESO=0X00 CERRARVALVULA  ;
        MOVLW 0X05
        MOVWF ERRORNUM
        MOVLW .30
        MOVWF FUNCION
        GOTO VERCONVERSOR


ERROR_06        ; PELIGRO!!! SE HA INTENTADO CALENTAR SIN LÍQUIDO
        BCF PORTA,3; FORZAR LA PARADA DEL CALENTADOR
        CLRF PROCESO; PROCESO=0X00 CERRARVALVULA
        MOVLW 0X06
        MOVWF ERRORNUM
        MOVLW .30
        MOVWF FUNCION
        GOTO VERCONVERSOR




         ORG 0X00F00;
                ; ERROR DEL PROGRAMA, POR ALGUNA EXTRAÑA RAZON PCL HA LLEGADO A UNA DIRECCIÓN PROHIBIDA

        BCF STATUS,IRP; BANCO 0 PARA DIRECCIONAMIENTO INDIRECTO
        BCF STATUS,RP1; IR AL BANCO 0
        BCF STATUS,RP0;
        CLRF PCLATH;

        MOVLW 0X09
        MOVWF ERRORNUM
        MOVLW .30
        MOVWF FUNCION
        GOTO BUCLE;
        END
```