

UNIVERSIDAD AUTÓNOMA AGRARIA ANTONIO NARRO
DIVISIÓN DE INGENIERÍA
DEPARTAMENTO DE MAQUINARIA AGRÍCOLA



CONTROL AUTOMÁTICO DE TEMPERATURA

Por:

MELBI LÓPEZ HERNÁNDEZ

TESIS

Presentada como requisito parcial para obtener el título de:

INGENIERO MÉCANICO AGRÍCOLA

BUENAVISTA, SALTILLO, COAHUILA, MÉXICO.

MAYO DEL 2010

UNIVERSIDAD AUTÓNOMA AGRARIA ANTONIO NARRO

DIVISIÓN DE INGENIERÍA

DEPARTAMENTO DE MAQUINARIA AGRÍCOLA

CONTROL AUTOMÁTICO DE TEMPERATURA

Por:

MELBI LÓPEZ HERNÁNDEZ

TESIS

Que se somete a consideración del H. Jurado Examinador como requisito parcial para obtener el título de:

INGENIERO MÉCANICO AGRÍCOLA

Aprobado por el comité de tesis

Director de Tesis

Dr. Santos G. Campos Magaña

Co-Director de Tesis

Sinodal

Dr. Martín Cadena Zapata

M.C. Juan Arredondo Valdez

Universidad Autónoma Agraria
Antonio Narro
Coordinador de la División de Ingeniería

Dr. Raúl Rodríguez García

BUENAVISTA, SALTILLO, COAHUILA, MÉXICO.

Coordinación de

MAYO DEL 2010

Agradecimientos

A **Nuestro Dios** eterno, por haberme dado las fuerzas durante todo este tiempo, no solo durante la carrera si no durante toda mi vida, que sin dudar el estuvo ahí para darme aliento e inspiración para realizar los trabajos de manera correcta y así haber salido adelante. ¡Gracias Señor!

A **Nuestra Alma Mater** que nos acobijo en sus instalaciones para adquirir el conocimiento necesario para llegar hasta aquí, de los cuales hubo momentos inolvidables, y que siempre estarán ahí para hacer memoria. Gracias **UAAAN**.

Al **Dr. Santos Gabriel Campos Magaña**, por la amistad que me brindo, sus enseñanzas, sus consejos otorgados para poder terminar satisfactoriamente la tesis.

Al **Dr. Martin Cadena Zapata**, por su amistad, por las enseñanzas que me ofreció durante la carrera, por su confianza y facilidades que me otorgo para poder trabajar en el departamento, a **Mis profesores de maquinaria**, que nos compartieron el conocimiento que han adquirido, ¡Gracias!

Al **M.C. Héctor H. Serna Fernández**, por ser más que un asesor un Amigo. ¡Muchas gracias profe!

A **Mis compañeros y amigos de la Generación CVIII**, que me brindaron su amistad, afecto y confianza, a **Obet, Nelson, Jorge Luis, Cándido, José, Ignacio, Nahúm, Patricio, Deyya, Romain, Álvaro, Darwin, Michel, Daniel, Víctor, Cesar, Samuel y Josafat**, por aquellos momentos inolvidables que jamás olvidare, ¡Dios los bendiga Siempre!.

A mis amigos que de alguna u otra manera me brindaron la mano y sin condiciones mil gracias, a todos mis compañeros de cuartos en los que habite durante la carrera ¡GRACIAS!, en especial a **Francisco, Obet, Orlando y Rubén**, por su amistad y confianza. Los admiro cuates...

DEDICATORIA

A MIS PADRES

Apolinar López Espinosa y Angelina Hernández Velasco

Por su amor, cariño, consejos y apoyo que desde toda la vida me han brindado de manera desinteresada, por los valores que me han inculcado, por su incansable labor diaria por vernos progresar en la vida, son excepcionales, “Los Amo”, admiro y respetare, este trabajo se los dedico a ustedes, sin ustedes no estaría hasta aquí, Gracias por haber creído en mi.

A MIS HERMANOS

Marín López Hernández

Anabel López Hernández

Harí Susana López Hernández

Alexander López Hernández

Por todo su cariño, amor y apoyo que me brindaron en toda la carrera, por ser más que mis hermanos, son mis mejores amigos, a mis cuñados **Claudia y Abdías**, gracias a todos por ser parte de este sueño hecho realidad,

A MIS ABUELITOS

Aquilino Hernández

Florentina Velasco †

Caralampio López Moreno †

Elía Espinosa Jiménez

Abuelito “Quicho” por su carisma y consejos que siempre me han ayudado a seguir adelante, abuelita Elia gracias por sus consejos y su cariño, a toda la familia **López y Hernández** gracias por todo, a Isain y Samuel por ser mas que primos, Amigos. Todo este trabajo va dedicado ustedes.

INDICE DE CONTENIDO

	Pagina
INDICE DE CUADROS	VII
INDICE DE FIGURAS	VIII
RESUMEN	XI
Palabras clave	XI
I INTRODUCCION	1
1.1 Objetivos	1
1.2 Hipótesis	1
II REVISION DE LITERATURA	2
2.1 Sistemas de control automático.....	2
2.2 Tipos de sistemas de control.....	4
2.2.1 Sistema en lazo abierto.....	4
2.2.2 Sistema en lazo cerrado.....	4
2.3 Microcontrolador.....	5
2.4 Sensores.....	6
2.4.1 Tipos de sensores.....	6
2.5 Sistemas de control realizados en base a diferentes microcontroladores.....	7
2.5.1 Sistema de control de temperatura para horno de gas.....	7
2.5.2 Controlador PID de temperatura de tipo didáctico.....	8
2.5.3 Sistema de calentamiento solar de agua con control electrónico de temperatura a través de un microcontrolador.....	10
2.5.4 Diseño de un sistema de control de temperatura para un horno eléctrico, por medio de un controlador ON-OFF.....	13
III MATERIALES, EQUIPOS Y METODO	16
3.1 Materiales y equipos.....	16
3.1.1 Estufa VWR, modelo 1326.....	16
3.1.2 HOBOS (4-Channel External).....	17
3.1.3 Sensores de temperatura exterior TMC6-HA.....	17
3.1.4 BoxCar 4.3.....	17
3.1.5 USB Serial Adapter RS-232.....	18
3.2 Principales componentes electrónicos de las placas de control.....	18

3.2.1 PIC16F84A.....	19
3.2.2 Moc3011.....	19
3.2.3 Relevador RAS0510.....	20
3.2.4 Sensor de temperatura 18B20 (Dallas).	20
3.2.5 Calentador.....	21
3.3 Programas utilizados en los sistemas de control.	21
3.3.1 MPLab IDE V8.10.....	21
3.3.2 Programador PICSTAR Plus.	22
3.3.3 Proteus V7.6.....	22
3.4 Método.....	22
3.4.1 Pruebas en la estufa de secado.	23
3.4.2 Simulación en software.	24
3.4.3 Pruebas de control con termostato electrónico	24
IV RESULTADOS Y DISCUSION.....	26
4.1 Pruebas realizadas en la estufa	26
4.1.1 Análisis de graficas.....	28
4.2 Simulación en tiempo real.	31
4.3 Pruebas realizadas en el calentador.....	33
4.3.1 Sistema controlado por el Relevador electrónico RAS-0510.....	34
4.3.2 Análisis de datos con el termostato electrónico.....	35
V CONCLUSIONES Y RECOMENDACIONES.....	39
VI BIBLIOGRAFIAS	40
ANEXOS	42
ANEXO 1.....	43
1.1 Entrada de datos externos y Características del HOBO 4-Channel External.	43
ANEXO 2.....	45
2.1 Especificaciones de los sensores de temperatura exterior TMC6-HA.....	45
ANEXO 3.....	46
3.1 Características del BoxCar 4.3.....	46
3.2 Pasos de Ajuste y programación.....	46
3.3 Forma de graficado	47
3.4 Requisitos del sistema	47

ANEXO 4	48
4.1 Especificaciones del USB Serial Adapter RS-232	48
4.2 Requisitos del sistema	48
ANEXO 5	49
5.1 Generalidades del PIC16F84a	49
5.2 Arquitectura interna	51
5.3 Memoria de programa	52
5.4 Memoria de datos.....	54
5.5 Descripción de las patillas o pines del microcontrolador.....	56
5.6 Otras características	57
ANEXO 6	58
6.1 Especificaciones del sensor 18B20	58
ANEXO 7	59
Procedimiento de programación y quemado del microcontrolador	59
7.1 Llamado del programa	59
7.2 Ensamblaje del programa.....	59
7.3 Programador Picstart Plus.....	60
7.4 Permitir el acceso del programador	60
7.5 Programación del microcontrolador de manera directa	61
7.6 Verificación del programa.....	61
7.7 El microcontrolador está listo para utilizarse.....	62
ANEXO 8	63
8.1 Proteus.....	63
8.1.1 Isis	63
8.1.2 Modulo VSM.....	64
8.1.3 Ares	64
ANEXO 9	65
Simulación en Proteus (ISIS)	65
9.1 Cargar el programa Proteus (ISIS)	65
9.2 Simulación.....	65
9.3 Instrucciones para el funcionamiento correcto en la simulación	66
9.4 Inicio de simulación	67
ANEXO 10	68

10.1 Funciones de la placa de control	68
10.2 Pulsador Modo	69
10.2.1 Modo "Termostato_OFF"	69
10.2.2 Modo "Termostato_Ajuste"	69
10.2.3 Modo "Termostato_ON"	69
10.3 Interrupciones	70
10.4 Funciones del termómetro DS1820	71
10.5 Resumen del programa	72
ANEXO 11.....	73
Vistas de las placas realizadas en Proteus (Ares), Placa No 2.....	73
ANEXO 12.....	74
Pruebas realizadas en la estufa.....	74
ANEXO 13.....	76
Pruebas realizadas en el calentador eléctrico	76
Anexo 14	78
Programa utilizado en los microcontroladores	78

INDICE DE CUADROS

	Pagina
Cuadro 4.1 Datos adquiridos en la estufa con el Hobo.....	30
Cuadro 4.2 Registro de ecuaciones cubicas $Y = b + bX + bX^2 + bX^3$	31
Cuadro 4.3 Control de calentador termostato digital.....	37
Cuadro 4.4 Registro de ecuaciones cubicas $Y = b + bX + bX^2 + bX^3$	38
ANEXOS	42
Cuadro 1.1 Especificaciones del hobo.....	43
Cuadro 5.1 Descripción de patillas o pines	56
Cuadro 12.1 Análisis de datos de la Prueba 20-90 (estufa).	74
Cuadro 13.1 Análisis de datos de la Prueba 35-H1.....	76

INDICE DE FIGURAS

	Página
Figura 2.1 Esquema del proceso de control automático	2
Figura 2.2 Sistema de control en lazo abierto.	4
Figura 2.3 Sistema en lazo cerrado.	5
Figura 2.4 Diferentes modelos de microcontroladores.....	6
Figura 2.5 Sensor analógico LM34.	6
Figura 2.6 Sensor digital TMC6-HA.	7
Figura 2.7 Sistema controlado por microcontrolador.....	8
Figura 2.8 Sistema de control de temperatura.....	9
Figura 2.9 Diagrama de bloque de la etapa de potencia.	10
Figura 2.10 Respuesta del sistema con el controlador PID.....	10
Figura 2.11 Calentador solar de agua bajo estudio.....	11
Figura 2.12 Placa de control con microcontrolador PIC16F877.	12
Figura 2.13 Sensor CI LM335.	13
Figura 2.14 Sistema de control del horno.	14
Figura 2.15 Diagrama del sistema ON/OFF de la alimentación del horno.....	14
Figura 3.1 Estufa VWR modelo 1326.	16
Figura 3.2 HOBO 4-channel External.....	17
Figura 3.3 Sensor TMC6-HA.....	17
Figura 3.4 Ambiente del programa Boxcar Pro 4.3.	18
Figura 3.5 Adaptador RS-232.....	18
Figura 3.6 Microcontrolador PIC16F84A.	19
Figura 3.7 Moc3011.....	20
Figura 3.8 Relevador RAS0510.	20
Figura 3.9 Sensor 18B20.	21
Figura 3.10 Calentador IH4000S	21
Figura 3.11 Programador PICSTAR Plus.....	22
Figura 3.12 Representación de combinación 30_30 (3-3).....	23
Figura 3.13 Desarrollo del Software.	25
Figura 4.1 Posición del HOBO en la estufa.....	26
Figura 4.2 Comparación de las graficas 30 [30, 60,90].....	27

Figura 4.3 Comparación de graficas 40 (30, 60, 90).....	28
Figura 4.4 Representación del ciclo 1 prueba 30_30.....	29
Figura 4.5 Simulación en Proteus (ISIS).....	32
Figura 4.6 Placa impresa realizada en Proteus (Ares).	32
Figura 4.7 Placa impresa realizada en Proteus (Ares) (Vista 3D)	33
Figura 4.8 Placas de control con diferentes actuadores.	33
Figura 4.9 Posición de los sensores en el calentador.	34
Figura 4.10 Placas sensando de forma paralela.....	35
Figura 4.11 Programación manual de las placas.....	36
Figura 4.12 Graficas de 35°, 40°, 45°, 50° y 55°.....	36
ANEXOS	42
Figura 3.1 Ajuste fácil del Logger en BoxCar Pro.	46
Figura 3.2 Exportación de BoxCAR Pro a Excel.....	47
Figura 5.1 Estructura interna del PIC16F84A.....	52
Figura 5.2 Composición de la memoria.....	53
Figura 5.3 Registros de datos.....	54
Figura 5.4 Pines del PIC16F84A.....	56
Figura 7.1 Llamado del programa.....	59
Figura 7.2 Ensamblado del programa	59
Figura 7.3 Llamado del programador a MPLab.....	60
Figura 7.4 Acceso de Picstart Plus.....	60
Figura 7.5 Programación directa	61
Figura 7.6 Verificación de la programación	61
Figura 7.7 Posición de los microcontroladores en las placas de control.....	62
Figura 8.1 Ambiente Proteus ISIS.....	63
Figura 8.2 Ambiente Proteus ARES.....	64
Figura 9.1 Cargando el circuito	65
Figura 9.2 Cargar programa para simulación	65
Figura 9.3 Comandos esenciales	66
Figura 9.4 Submenú para selección del programa.....	66
Figura 9.5 Simulación.....	67
Figura 10.1 Posición de los pulsadores en las placas de control	68
Figura 10.2 Diagrama de flujo de la subrutina Cambiar_Modo	69

Figura 10.3 Diagrama de flujo de la subrutina de atención a las interrupciones ..	70
Figura 10.4 Diagrama de flujo principal del termostato digital.....	71
Figura 11.1 Pistas de la placa No.2 (Plano).....	73
Figura 11.2 Vista en 3D, Placa No. 2.....	73
Figura 11.3 Placa impresa, Placa No 2.....	73
Figura 12.1 Prueba 20-90.....	74
Figura 12.2 Prueba 20-90, dividida en 3 Ciclos.....	75
Figura 13.1 Prueba 35-H1.	76
Figura 13.2 Prueba 35-H1, dividida en 3 Ciclos.	77

RESUMEN

Se desarrollo un sistema de control automático de temperatura, de lazo abierto (on/off), mediante el uso de microcontroladores. El tipo de controladores que se utilizaron para manejar los rangos de control automático son de la marca MICROCHIP en tipo de gama media. Para este sistema de control automático, se emplearon dos sistemas de potencia uno del tipo electrónico Triac (MOC3011) y el otro electromecánico RAS0510. El desarrollo y programación de los microcontroladores (PIC16F84A) se realizaron en los ambientes de Mplab y de PROTEUS (Isis y Ares).

Para el monitoreo y registro de temperatura se utilizaron sensores analógicos alimentando la información en un Data Logger (HOBO-H08) programados para el almacenamiento de los cambios de temperatura en el proceso de control. Para el termostato electrónico se empleo un sensor de temperatura digital el DS18B20 de la empresa Texas Instruments.

En el sistema de termostato electrónico que se diseño para el calentador y estufa mostró una variabilidad de $\pm 1^{\circ}\text{C}$ con tiempo de respuesta por ciclo de 5 minutos en comparación con el sistema propio de control de la temperatura de la estufa (BWR, modelo 1326) con una variabilidad de esta de $\pm 2^{\circ}\text{C}$ con tiempo de respuesta de 30 minutos.

Palabras clave: Microcontroladores, PIC16F84A, Sensores, control de temperatura.

I INTRODUCCION

Los sistemas de control automático en nuestro días es una de las herramientas más utilizadas por las grandes industrias a nivel mundial, en donde las mas inimaginables equipos electrónicos son controlados por pequeños microcontroladores, que pueden mantener (según su programación) desde el más pequeño sistema de control hasta el más sofisticado, dando así seguridad al propietario como al trabajador sabiendo que está funcionando de manera correcta, sin que exceda los limites.

Pero remontémonos a nuestro medio que si se analiza de manera detallada esto es tan común que en casi todos los aparatos electrónicos encontramos esta función de control, a decir verdad la mayoría de nosotros solamente adquirimos el producto sin cuestionarnos de ¿cómo funciona?, ¿cuál es su universo que lo conforma?. A lo largo de la historia, el ser humano siempre ha tratado de manipular los instrumentos de uso común de tal manera que sean más fáciles y eficientes los trabajos o labores diarias. Surgiendo así un sinfín de ideas, de los cuales muchas de ellas quedaron en el aire sin que pudieran ser probados, ¿Por qué? por los mismos factores que desde siempre ha faltado “El factor económico”, que es primordial para poder realizar proyectos en grandes escalas.

Uno de los objetivos por el cual los sistemas de control automático se han desarrollado, es para dar mayor comodidad a nuestra sociedad que cada día exige cosas diferentes, buscando más calidad y mejor precio, justo ahí es donde las grandes industrias entran en competencia buscando nuevas tecnologías para poder ser eficiente en el ámbito empresarial, analizándolo

Desde esta perspectiva las empresas que se estanca tecnológicamente, automáticamente quedan en la obsolescencia.

Los sistemas de control evolucionan sorprendentemente a nivel mundial, esto implica una evolución parcial en donde los países de alto nivel se mantienen a la vanguardia industrial, esto trae como consecuencia un alto ingreso económico, que beneficia a toda la sociedad, proporcionando así más empleos mucho mejor pagados, educación de calidad, mas estabilidad social, en si una mejor calidad de vida.

Un claro ejemplo es Japón, donde sus grandes industrias reconocidas a nivel mundial, en sus procesos automáticos la mano de obra está siendo obsoleta, evitando así el mínimo error y mucho más eficiencia y calidad

La historia nos da algunos ejemplos de cómo ha evolucionado la tecnología hasta nuestros días, trayendo pros y contras en lo económico, social y ambiental.

Ante estos retos es importante señalar la necesidad de que los egresados de la carrera de Ingeniero Mecánico Agrícola tengan el conocimiento básico del control de variables, como son la temperatura, presión, humedad, flujo, posición entre otros.

1.1 Objetivos

1. Desarrollar un sistema de control automático de temperatura de bucle abierto, con fines didácticos.
2. La simulación en tiempo real y su comprobación vía hardware.

1.2 Hipótesis

Es posible diseñar un sistema de control automático de temperatura con fines didácticos.

II REVISION DE LITERATURA

2.1 Sistemas de control automático.

Los sistemas de control automático en lo que consta a su operación, no tiene demasiada complejidad, algunos autores lo definen como algo esencial e indiscutiblemente primordial en nuestra sociedad, Navarro 2004, lo define como un conjunto formado por el proceso y el sistema de control. Para conseguir este objetivo son indispensables una serie de dispositivos adicionales, que de alguna manera lleven a cabo el control o regulación del proceso con el fin de sustituir la acción del hombre por un dispositivo llamado controlador o regulador. El conjunto que forman los componentes que llevarán a cabo el control automático de un proceso se llama sistema de control automático representado en el diagrama de la Figura 2.1.

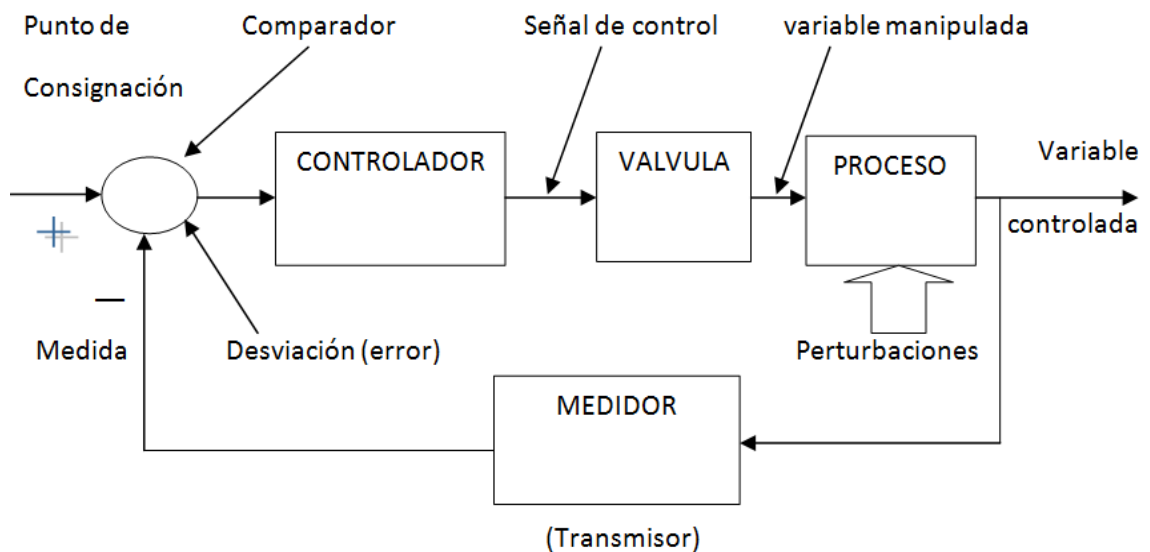


Figura 2.1 Esquema del proceso de control automático. Fuente: Navarro 2004.

Angulo y Raya 2004, nos dicen que el funcionamiento de un sistema depende de las características e interacciones entre sus elementos. Se denomina sistema de control aquel sistema constituido por un conjunto de elementos conectados de forma que permiten regular o gobernar otro sistema, denominado planta o proceso.

Acedo 2006, define a un sistema de control automático el cual mide una variable y actúa de una forma determinada para que esa variable se mantenga en un valor deseado o de referencia. Un sistema de control no automático, al actuar en lazo abierto, puede hacer cambiar el valor de la variable pero no llevarla al valor de referencia.

Angulo y Raya 2004. **Planta $G(s)$** , son los componentes del sistema que hay que controlar. También se denomina proceso. En la planta se produce la transformación de la energía suministrada que se debe controlar. En general, la energía eléctrica suministrada al actuador se transforma en energía mecánica (motor) en energía potencial.

Angulo y Raya 2004. **Controlador $G_c(s)$** , proporciona la excitación al proceso para alcanzar la salida deseada. Es diseñado para controlar el comportamiento global del proceso y también recibe el nombre de regulador o compensador.

Angulo y Raya 2004. **Sensor $H(s)$** , es el elemento que permite sensor la salida de la planta. En el lazo de control se suele situar en la realimentación. En ocasiones, su función de transferencia se incluye en la transmitancia del proceso $G(s)$. Los sensores se caracterizan porque transforman una observación física (nivel, caudal, velocidad, temperatura) en una señal eléctrica. Consta de dos partes, el detector y un transductor o transmisor. El detector transforma la observación física en una señal eléctrica (mA, pulsos).

Acedo 2006, nos hace referencia sobre los controladores automáticos en donde han incorporado una habilidad que corresponde al ser humano. Esa habilidad es la de <<sentir>> una condición correcta o incorrecta y actuar para corregirla. Las ventajas que se derivan porque una maquina realice algunas funciones que

hace al hombre son obvias, sobre todo cuando la maquina llega a superar al hombre en esa función. Un controlador automático no se cansa, no necesita dormir, ni sale de vacaciones; siempre está al cuidado del trabajo encomendado.

2.2 Tipos de sistemas de control.

2.2.1 Sistema en lazo abierto.

Katsuhiko 2003. Son los sistemas en los cuales la salida no tiene efecto sobre la acción de control. En otras palabras, en un sistema de control en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada. En cualquier sistema de control en lazo abierto, la salida no se compara con la entrada de referencia. Así, a cada entrada de referencia le corresponde una condición de operación fija; como resultado de ello, la precisión del sistema depende de la calibración. Ante la presencia de perturbaciones, un sistema de control de lazo abierto no realiza la tarea deseada (Figura 2.2).

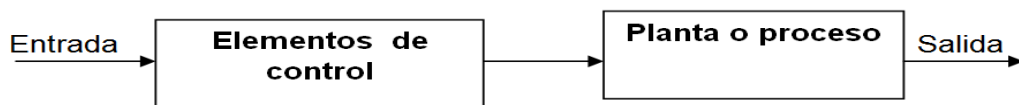


Figura 2.2 Sistema de control en lazo abierto. Fuente: Katsuhiko 2003.

2.2.2 Sistema en lazo cerrado.

Vilanova I, Geltrú 1998. La salida no se compara con la entrada de referencia, por ello cada entrada corresponderá a una operación prefijada sobre la señal de salida. Se puede asegurar entonces que la exactitud del sistema depende en gran manera de la calibración del mismo y, por tanto, la presencia de perturbaciones en la cadena (señales indeseadas) provocará que éste no cumpla la función asignada. Que en si puede fallar el sistema si no se calibra de una manera adecuada y que los sistemas de control en lazo cerrado (Figura

2.3), la señal de salida tiene efecto sobre la acción de control. La señal controlada debe realimentarse y compararse con la entrada de referencia, tras lo cual se envía a través del sistema una señal de control, que será proporcional a la diferencia encontrada entre la señal de entrada y la señal medida a la salida, con el objetivo de corregir el error o desviación que pudiera existir. La principal ventaja de los sistemas de control en lazo cerrado es que el uso de la realimentación hace al conjunto menos sensible a las perturbaciones externas y a las variaciones de los parámetros internos que los sistemas en lazo abierto.

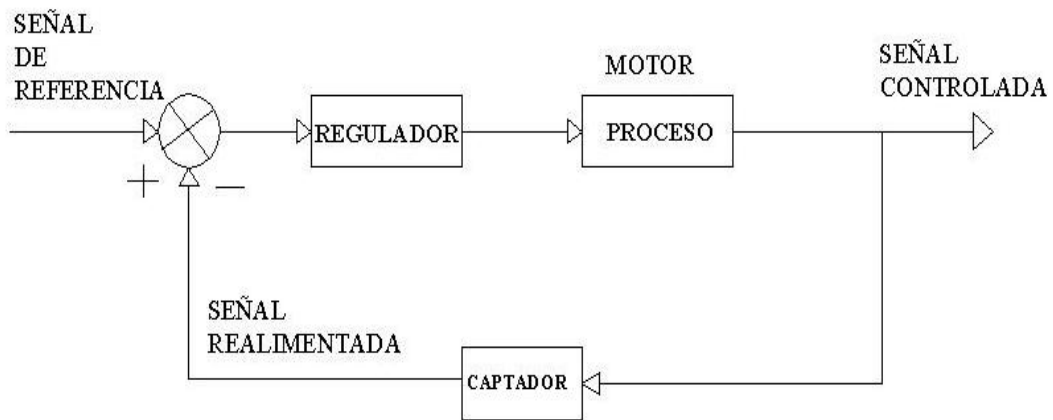


Figura 2.3 Sistema en lazo cerrado. Fuente: Vilanova I, Geltrú 1998.

2.3 Microcontrolador.

Valdés y Pallás 2007. Los microcontroladores están fabricados fundamentalmente para ser utilizados en aplicaciones puntuales, como se observa en la Figura 2.6 es decir, aplicaciones en donde el microcontrolador debe realice un pequeño número de tareas, al menor costo posible. En estas aplicaciones, el microcontrolador ejecuta un programa almacenado permanentemente en su memoria, el cual trabaja con algunos datos de forma temporal e interactúa con el exterior a través de líneas de entrada y salida de que dispone. Además es parte de la aplicación, es un controlador incrustado en la aplicación.



Figura 2.4 Diferentes modelos de microcontroladores.

Fuente: www.rev-ed.co.uk/picaxe/es/picaxe.htm.

2.4 Sensores.

Pallás 1993. Es un elemento de un sistema que lo conecta con su entorno físico, excluyendo al usuario. La función de los sensores es obtener señales electrónicas en respuesta a magnitudes de entradas no eléctricas, existen varios tipos de sensores pero los más comunes son los analógicos y digitales.

2.4.1 Tipos de sensores.

1. Pallás 2003. Analógicos, en este tipo de sensores la salida varía, a nivel macroscópico, de forma continua. La información está en la amplitud, si bien se suelen incluir en este grupo los sensores con salida en el dominio temporal como se observa en la Figura 2.5.



Figura 2.5 Sensor analógico LM34. Fuente: www.futureelectronics.com.

2. Pallás 2003. Digitales, la salida varía en forma de saltos o pasos discretos. No requiere conversión A/D (Analógico/Digital) y la transmisión de su salida es más fácil. Tienen mayor fidelidad y mayor fiabilidad, y

muchas veces mayor exactitud, pero lamentablemente no hay modelos digitales para muchas de las magnitudes físicas de mayor interés, que en muchas ocasiones son las más requeridas (Figura 2.6).



Figura 2.6 Sensor digital TMC6-HA.

2.5 Sistemas de control realizados en base a diferentes microcontroladores.

2.5.1 Sistema de control de temperatura para horno de gas.

Arias y Vásquez 2002, el sistema está basado en lo que sus autores llaman redundancia de control, una PC (Personal Computer) actúa como el control principal en este caso y cuando falla es reemplazada por un microcontrolador **80C32** de Intel. Su meta es asistir el proceso de quema a través del monitoreo en tiempo real y la generación de reportes, buscando ser una opción económica y efectiva. El sensaje de temperatura es hecho por un termopar tipo K cuyo circuito es sencillo gracias a un CI (Circuito integrado) compensador de unión fría y el flujo de gas es manejado por una electroválvula de dos vías normalmente cerrada. Como el proceso nunca debe ser detenido antes de su finalización, el sistema cuenta con un dispositivo de respaldo de energía. El tipo de control es ON/OFF, es decir, tan solo se cuenta con una válvula solenoide para permitir el flujo del combustible desde la entrada principal. Para hacer una regulación automática del gas lo apropiado es agregar servo válvulas.

El sistema tan solo muestra temperatura, no así la presión del gas combustible, que también es importante registrar. El sistema realizado se muestra en la siguiente Figura 2.7.

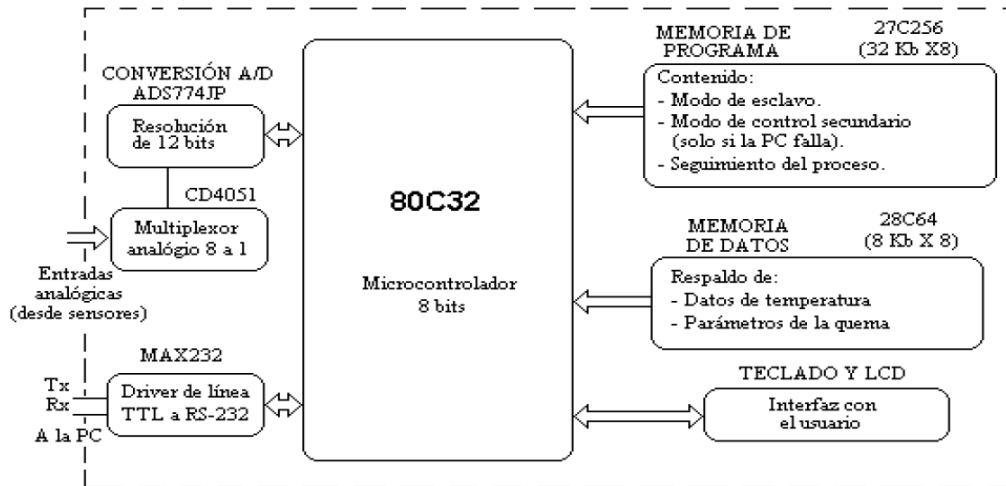


Figura 2.7 Sistema controlado por microcontrolador. Fuente: Arias y Vásquez 2002.

Resultado obtenido con el Microcontrolador 80C32.

1. Es económico en comparación con las existentes en el mercado y sin perder la eficiencia.
2. Al contener la lógica del proceso, adquiere un perfil de tutor para operadores poco experimentados.
3. Se sobrepone a cortes de energía.

2.5.2 Controlador PID de temperatura de tipo didáctico.

Bañuelos, *et al.*, 2005, menciona que es un sistema controlador PID analógico, basado en amplificadores operacionales de propósito general. El controlador está implementado de tal manera que fácilmente se distinguen cada una de las partes del controlador: sumador, acción proporcional, acción integral y acción derivativa. El sistema también cuenta con una etapa de potencia, que utiliza el método de “paquetes de onda” para regular la potencia entregada a una parrilla eléctrica. En la parrilla se coloca una tasa con agua, y la temperatura se mide

mediante un sensor de temperatura integrado (LM35) o un termistor. Considerando que se cuenta con una fuente de alimentación y un multímetro. Se utilizo además un optotriac **MOC3030** el cual al contar con un detector de cruce por cero se reduce la emisión de radiación electromagnética. En la Figura 2.8, se muestra un diagrama simplificado del sistema de control de temperatura. El sistema está constituido por un controlador PID analógico, una etapa de potencia, una parrilla eléctrica y un sensor de temperatura.

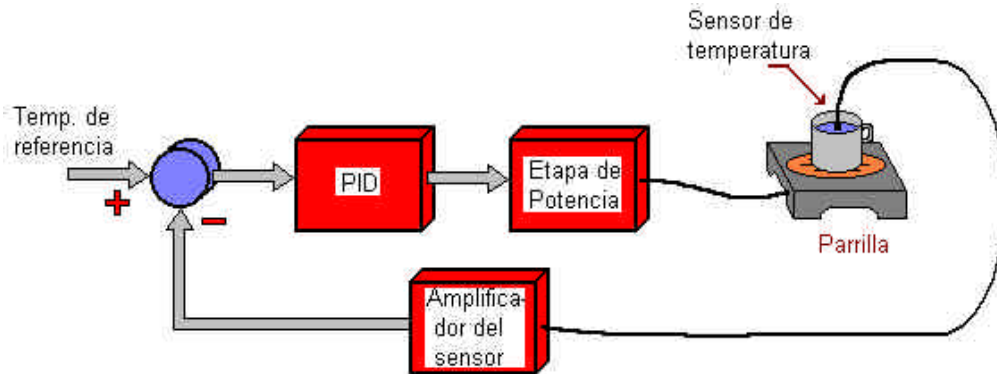


Figura 2.8 Sistema de control de temperatura. Fuente; Bañuelos, *et al.*, 2005.

En la Figura 2.9 se puede observar el diagrama de bloques de la etapa de potencia. Ésta consiste de un generador de tiempo proporcional, una etapa de aislamiento y un interruptor de estado sólido (triac). El circuito generador de tiempo proporcional recibe la señal de salida del controlador PID (señal de control) y la transforma en un pulso cuya duración es proporcional a la señal de control. La etapa de aislamiento está formada por un opto-acoplador MOC3030, el cual cuenta con un circuito detector de cruce por cero. Finalmente, la etapa de potencia cuenta con un triac, con el cual se controla el voltaje de línea que se aplica a la parrilla.

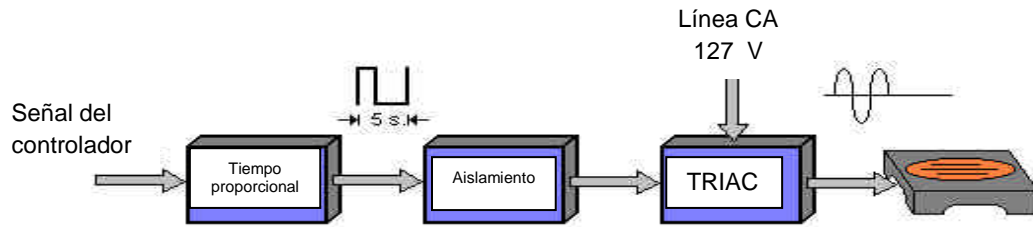


Figura 2.9 Diagrama de bloque de la etapa de potencia. Fuente; Bañuelos *et al.*, 2005.

Resultado del proyecto realizado con el MOC3030, se implementó el controlador PID analógico, y junto con el sensor y la etapa de potencia se realizó una prueba experimental, como puede observarse en la Figura 2.10, el controlador permite reducir el tiempo de asentamiento de la temperatura.

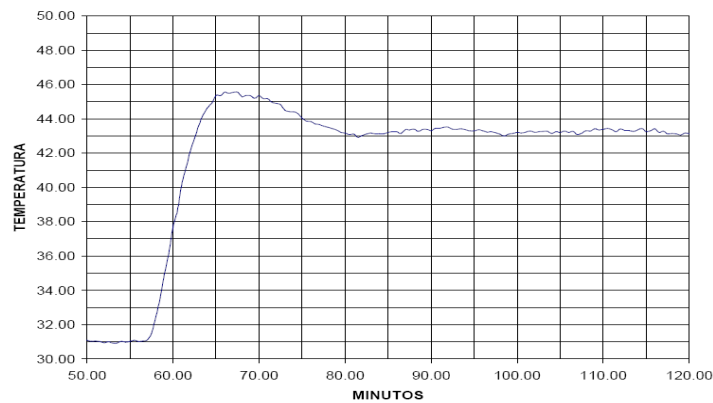


Figura 2.10 Respuesta del sistema con el controlador PID. Fuente; Bañuelos, *et al.*, 2005.

En la figura 2.10, existe un pequeño sobrepaso, el cual puede reducirse mediante un proceso adicional de sintonización manual.

2.5.3 Sistema de calentamiento solar de agua con control electrónico de temperatura a través de un microcontrolador.

López y Landaverde 2006, para el calentador solar utilizamos un control electrónico de temperatura, del tipo On-Off, basado en el dispositivo

microcontrolador **PIC16F877A** y en un sensor de temperatura LM35, el cual despliega la temperatura medida en una pantalla de LCD Hitachi HD44780.

Se fabricó una tubería con seis vueltas en caracol con tubo de cobre de $\frac{1}{4}$ " de 39 cm. de diámetro exterior y 27.5 cm. de diámetro interior, teniendo una longitud total de 3.26 m., la cual se soldó sobre una placa de acero inoxidable de 42.5 cm. X 42. 5 cm, según Figura 2.11.

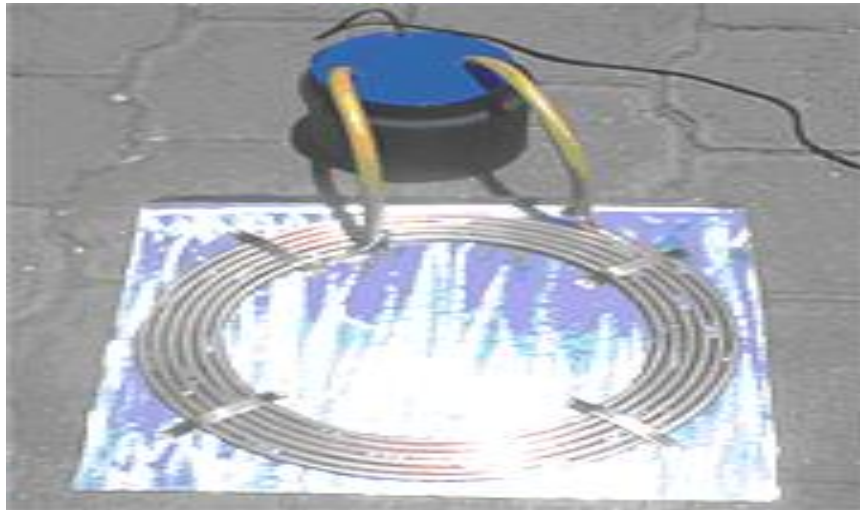


Figura 2.11 Calentador solar de agua bajo estudio. El depósito de agua caliente es el tanque de color negro. Fuente: López y Landaverde 2006.

Mediante dos mangueras flexibles, conectamos las terminales de la tubería en caracol a una bomba dentro de un tanque térmico de doble pared con aire en medio, como aislante térmico para conservar la temperatura del agua acumulada.

Utilizamos el PIC16F877 y el sensor LM 35. El primero para medir la temperatura, en un rango de 0 °C hasta 50 °C. Se conectó a un LCD el PIC16F877 con 3 botones que sirven de teclado (La conversión de datos que utiliza el PIC16F877 es de 256 bits).

Con el teclado se introducen los datos de temperatura a la cual se activará y apagará la carga del motor del sistema de bombeo de forma ascendente y descendente.

En el LCD se mostrara la temperatura introducida mediante el teclado y el incremento de temperatura del sensor LM35, además del rango de activación y desactivación de la carga (utilizando pulsadores como teclado), como se muestra en la Figura 2.12.

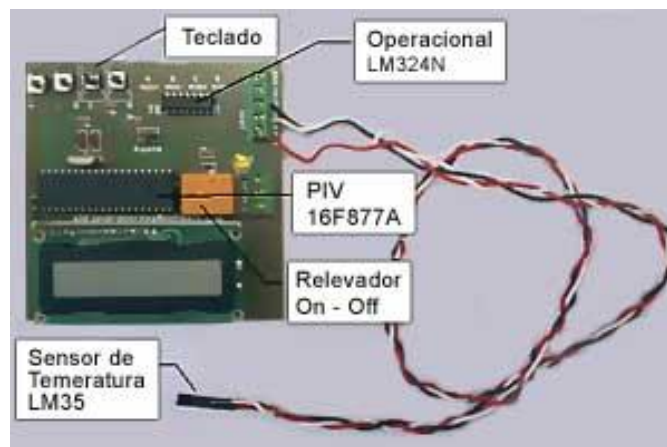


Figura 2.12 Placa de control con microcontrolador PIC16F877. Fuente: López y Landaverde 2006.

Cuando el valor de la temperatura del agua en el depósito sea igual o menor a la que el usuario digitó en el teclado como límite inferior, es decir 30° C para nuestro caso, la carga eléctrica hacia la bomba se activará. En contra parte, cuando el valor de la temperatura del agua es igual o mayor a la que el usuario digitó como límite superior, 45 °C. en nuestro caso, la carga se desactiva apagando el motor de la bomba.

Resultado obtenido con el PIC16F877, el controlador de temperatura del agua almacenada en el depósito térmico logró mantener la temperatura del líquido entre los rangos preestablecidos, cuando estuvo trabajando el sistema y mientras la intensidad de la radiación solar era suficiente para hacerlo.



Figura 2.14 Sistema de control del horno. Fuente: Hernández y Barajas (2000).

En la Figura 2.15, se muestra el diagrama general del diseño del sistema de control.

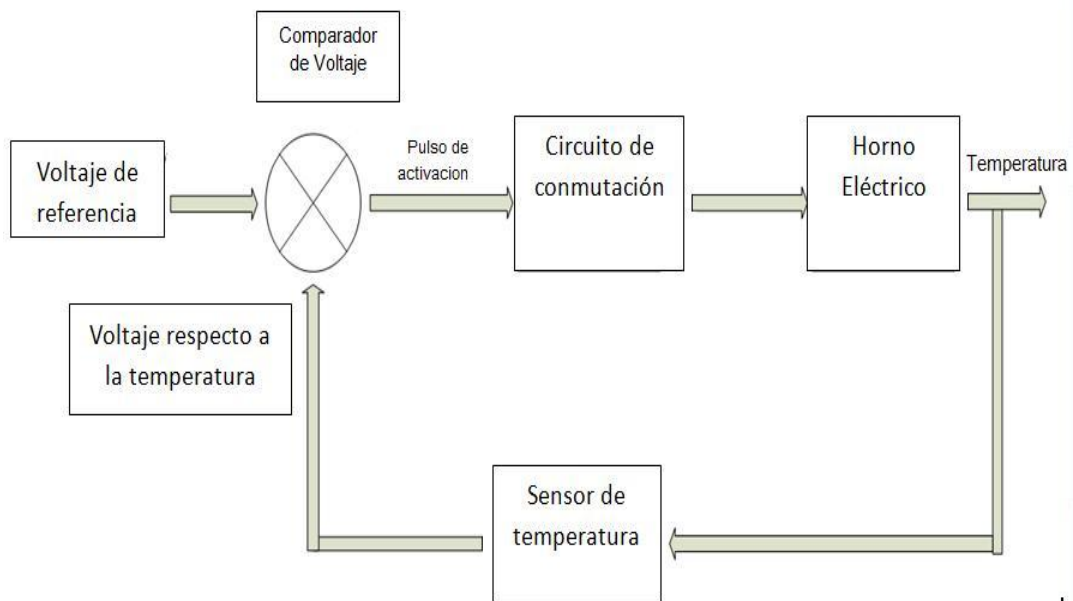


Figura 2.15 Diagrama del sistema ON/OFF de la alimentación del horno.
Hernández y Barajas 2000.

Al observar detalladamente el diagrama anterior, muestra que el sistema no es totalmente eficiente, el sistema está controlado por el incremento y decremento

de la temperatura, este sistema sería muy eficiente en los sistemas de más alto rango de temperatura.

Resultados obtenidos con el sensor LM335, el circuito funciona correctamente aunque con algunas variantes, durante el proceso de elaboración no se tuvieron percances, ya que el sensor LM335 tiene buena linealidad, además adaptamos el circuito de tal manera que no solo pueda controlar la temperatura del horno, si no que puede controlar la temperatura de cualquier otro dispositivo.

III MATERIALES, EQUIPOS Y METODO

3.1 Materiales y equipos.

Los principales componentes utilizados en los sistemas de control evaluados fueron.

3.1.1 Estufa VWR, modelo 1326.

Fabricada por Sheldon Manufacturing Inc. 2000, estas unidades son de carácter general y puede utilizarse tanto profesional, industrial o educativamente. Donde la preparación y las pruebas de materiales se realizan a presión atmosférica por la cabina que tiene. La estufa esta en el laboratorio de Mecánica de Suelos del Departamento de Maquinaria de la UAAAN.

Sus especificaciones son, modelo No. 1326 de la Marca Shellab (VWR), Con N° de parte 9071194, con capacidad de 9 amperes-115 volts-60 HZ, el rango de operación de temperatura es de + 5 ° C a 200 ° C., obsérvese en la Figura 3.1.



Figura 3.1 Estufa VWR modelo 1326.

3.1.2 HOBOS (4-Channel External).

En la página www.microdaq.com, nos dice que el HOBO mostrado en la Figura 3.2, acepta sensores externos de temperatura, es un registrador que midió y registro los cambios de temperatura en los sistemas, AC (Corriente Alterna) voltios de corriente. Detalles de sus especificaciones en Anexo 1.



Figura 3.2 HOBO 4-channel External. Fuente: www.microdaq.com.

3.1.3 Sensores de temperatura exterior TMC6-HA.

Este sensor puede observarse en la Figura 3.3 y es el que se conecta al HOBO H08, se pueden conectar directamente a las tomas de entrada externa de HOBO H08 y son registradores de datos para medir la temperatura en el aire, el agua o el suelo. Especificaciones generales en Anexo 2.

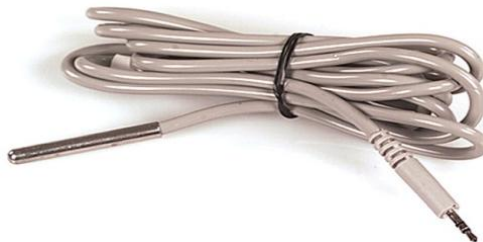


Figura 3.3 Sensor TMC6-HA. Fuente: www.microdaq.com.

3.1.4 BoxCar 4.3.

Es una aplicación de Windows para registrar y gestionar datos utilizando registradores de datos HOBO. BoxCar Pro 4.3, en la Figura 3.4 podemos visualizar su ambiente. Detalles en Anexo 3. (www.microdaq.com).

3.2.1 PIC16F84A.

El motor de las placas de control se debe principalmente a este microcontrolador, Figura 3.6, que lleva la programación necesaria para controlar el sistema, con instrucciones lógicas, programadas previamente.



Figura 3.6 Microcontrolador PIC16F84A. Fuente: www.steren.com.

Microchip Technology Inc 2001, nos especifican su composición exterior de 8 bits, 18 pines, y un set de instrucciones RISC muy amigable para memorizar y fácil de entender. Especificaciones en Anexo 5.

- Memoria Flash de programa (1K x 14).
- Memoria EEPROM de datos (64 x 8).
- Memoria RAM (68 registros x 8).
- Varios puertos de entrada-salida (13 pines en dos puertos, 5 pines el puerto A y 8 pines el puerto B).
- HS se utiliza un cristal de cuarzo o resonador cerámico (Hasta 10 Mz)
- XT Cristal o resonador hasta 4 Mhz.

3.2.2 Moc3011.

Este dispositivo funciona como un triac, Figura 3.7, están diseñadas para interactuar entre controles electrónicos y triacs de potencia para controlar cargas resistivas e inductivas en operaciones de 115/240 VAC. Este dispositivo se utilizo en la placa de control 1. (www.steren.com).



Figura 3.7 Moc3011. Fuente: www.steren.com.

3.2.3 Relevador RAS0510.

En la Figura 3.8, se muestra un relevador compacto de 1 polo, 2 tiros (SPDT) y bobina de 5 Vcc. ideal para equipos y proyectos electrónicos. Es un dispositivo electromecánico, que funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes, se utilizo en la placa no. 2. (www.steren.com).

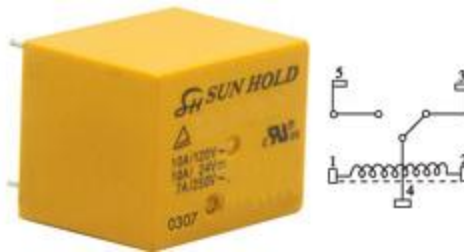


Figura 3.8 Relevador RAS0510. Fuente: www.steren.com.

3.2.4 Sensor de temperatura 18B20 (Dallas).

Dallas semiconductor 2008, el DS18B20 representado en la Figura 3.9, es un pequeño sensor de temperatura de 3 pines que se asemeja a un transistor. Cada uno tiene un único "código de 64 bits de serie" y pueden comunicarse a través de un "1 Wire (alambre)" estándar, para conectarse al microcontrolador y utilizarlos en los sistemas de control. Este sensor se conecto a las diferentes placas de control, especificaciones en Anexo 6.

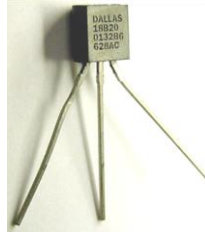


Figura 3.9 Sensor 18B20. Fuente: Dallas semiconductor 2008.

3.2.5 Calentador.

Para poder realizar las pruebas en las placas de control automático se utilizó un calentador Marca Travel Aire, modelo IH4000S, potencia 1200 Watts, voltaje 110, corriente 10,9 Amperes, frecuencia 60 Hz. Figura 3.10.



Figura 3.10 Calentador IH4000S.

3.3 Programas utilizados en los sistemas de control.

3.3.1 MPLab IDE V8.10

Es un programa de programación y simulación de sistemas previamente realizados, con este programa se programó los microcontroladores. En anexo 7 se menciona el quemado del programa al PIC16F84A.

3.3.2 Programador PICSTAR Plus.

En su página www.microchip.com, es una herramienta complementaria de MPLab, para programar los microcontroladores, el cual utilizamos, para la programación del microcontrolador se describe en el Anexo 7. Este dispositivo es conectado por vía usb como se muestra en la Figura 3.11.



Figura 3.11 Programador PICSTAR Plus.

3.3.3 Proteus V7.6.

Proteus se divide en Ares e Isis, lo cual en Isis se simulo el programa en tiempo real, para lograr realizar las pistas se utilizo Ares. Más especificaciones en Anexo 8.

3.4 Método.

El proceso de desarrollo de este sistema de control automático de lazo abierto se dividió en tres fases. Las cuales a continuación se enuncian y posteriormente se describen.

- Pruebas en la estufa de secado.
- Simulación en software.
- Pruebas de control con termostato electrónico.

3.4.1 Pruebas en la estufa de secado.

1. Primeramente se realizaron pruebas de comportamiento de la estufa VWR modelo 1326, como se explico en el apartado de Figura 3.1, se utilizaron las siguientes combinaciones de los controles de la misma 30 (30, 60, 90), 40 (30, 60, 90) realizando tres repeticiones por cada arreglo. (Figura 3.12).

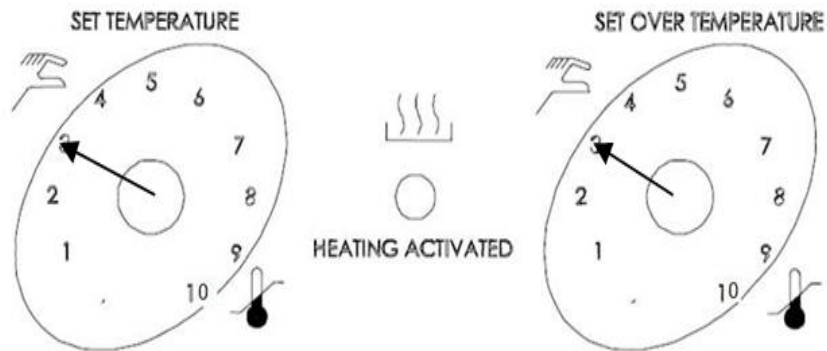


Figura 3.12 Representación de combinación 30_30 (3-3).

Para lo cual se introdujo el sensor **TMC6-HA** de temperatura y se registro en el data logger HOB0. El tiempo de registro para cada corrida fue de 4 a 7 horas. Tiempo suficiente para mostrar al menos tres ciclos de respuesta del control interno de la estufa. Posterior a la obtención de datos se determinaron las ecuaciones de respuesta de cada arreglo mediante análisis de regresión empleando el paquete estadístico de Minitab 14.

Además se determino el error, tiempo de respuesta y el periodo de cada ciclo de trabajo.

3.4.2 Simulación en software.

Se analizo su estructura, haciendo modificaciones hasta tener la programación deseada.

1. Análisis del programa (Anexo 14), tomado del libro “Microcontrolador PIC16F84, Desarrollo de Proyectos” (Palacios *et al.*, 2008).
2. Se programo en MPLab, corriéndolo y probando su funcionamiento correcto evitando que tenga errores. El método de programación se describe en el Anexo 7.
3. Programación en Proteus (ISIS) para su análisis en tiempo real. Para la devida y fácil programación pueden observarse de manera detallada en el Anexo 9.
4. Viendo que el sistema simulado en Proteus (ISIS), funciono de manera correcta se realizaron las pistas de las placas en Proteus (ARES), checando nuevamente de que no existieran pistas montadas una sobre otra para evitar cortos circuitos.

3.4.3 Pruebas de control con termostato electrónico.

En este apartado se menciona de manera detallada el proceso de quemado de las placas.

1. Impresión de las mallas para el quemado e impresión de las pistas de conexión sobre las placas fenolicas.
2. Cuando las placas ya estaban impresas se procedió a montar las piezas electrónicas, checando que efectivamente las piezas electrónicas funcionaran de manera correcta.
3. Antes de montar el Relevador RAS0510 y el MOC3011, de manera independiente se simularon en el proto board.
4. Verificación del funcionamiento correcto de los circuitos electrónicos.
5. Las placas de control se montan sobre una placa de metal, posterior a esto se procedió a conectar todos los cables de las placas, evitando cortos circuitos.

Teniendo el microcontrolador programado correctamente, se verifico la programación del microcontrolador, referenciado en el diagrama que se muestra en la Figura 3.13.

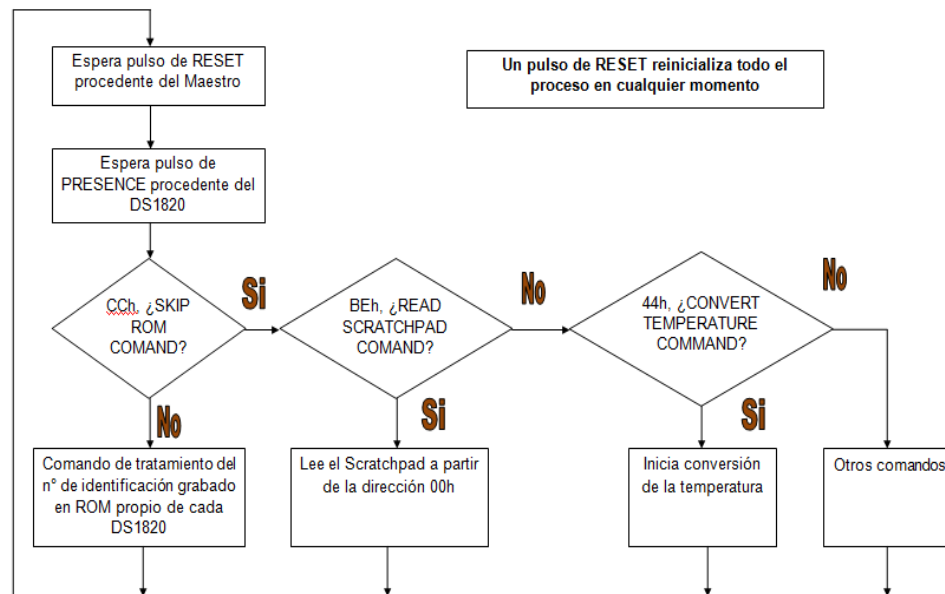


Figura 3.13 Desarrollo del Software. Fuente: Palacios *et al.*, 2008.

Las pruebas realizadas a cada uno de sistemas de control se mencionan a continuación:

Especificaciones del funcionamiento de las placas de control Anexo 10, se realizaron 5 pruebas, con intervalos de 5 °C cada uno (35, 40, 45, 50 y 55 °C). Con tiempos desde 2 hasta 3 horas por prueba con tres repeticiones por combinación. Para el termostato electrónico se empleo el sensor digital de temperatura DS18B20, cuyas características se encuentran en el Anexo 6.

Para el registro de temperatura se utilizo el sensor **TMC6-HA** y el data logger HOBO modelo H08-006-04. Posterior a la obtención de datos se determinaron las ecuaciones de respuesta de cada arreglo mediante análisis de regresión empleando el paquete estadístico de Minitab. Determinando además: el error, tiempo de respuesta y el periodo de cada ciclo de trabajo.

IV RESULTADOS Y DISCUSION

4.1 Pruebas realizadas en la estufa

Como se menciona en el capítulo anterior se realizaron pruebas a la estufa VWR modelo 1326, se analizó el tiempo de respuesta del sistema de control interno que tiene, con la ayuda del HOBO (Figura 4.1) modelo H08-006-04, se programó para registrar los cambios de temperatura cada segundo.



Figura 4.1 Posición del HOBO en la estufa

Para evitar cambios de temperatura en el horno se realizó una prueba por día,

aunque estas fueron distintas en tiempo, desde 4 hasta 7 horas por corrida, esto fue con la finalidad de que el sistema establezca la temperatura, y muestre como funciona su sistema de control. Como cito en el capitulo anterior se realizaron combinaciones de temperatura en la estufa por medio de 2 perillas mostradas en la Figura 3.12, que tienen una escala de 1 hasta 10 cada una.

Los datos obtenidos se representan en la grafica de la Figura 4.2, en donde se representan las combinaciones de las corridas de 30 (30, 60 y 90), donde se puede observar el comportamiento de control de temperatura de la estufa.

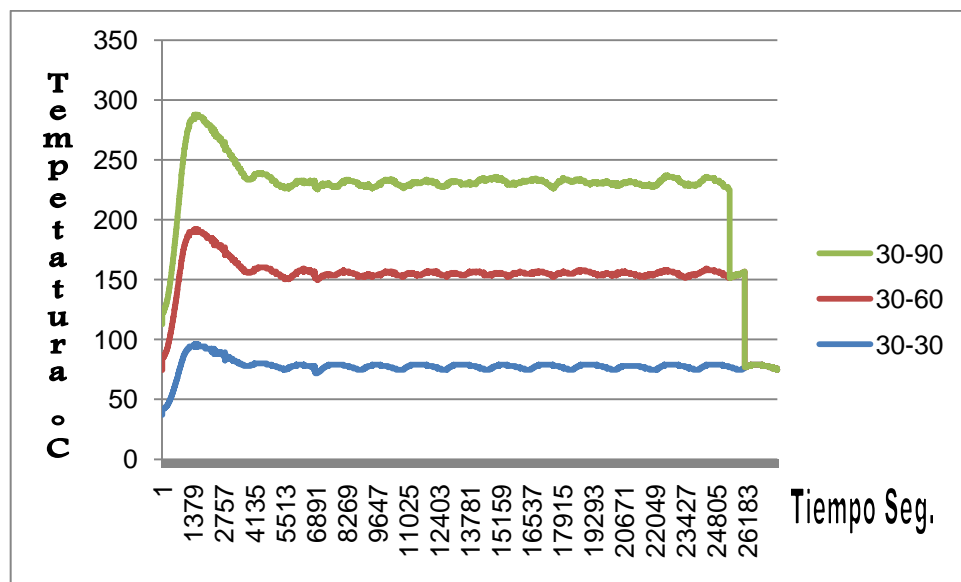


Figura 4.2 Comparación de las graficas 30 [30, 60,90].

Al analizar de manera detallada las graficas podemos ver muy claramente, la temperatura de disparo, el error y el tiempo de estabilidad que tiene el horno, donde los ciclos que se muestran tuvieron un periodo de hasta 40 minutos.

La grafica representada en la Figura 4.3, con rangos de control de temperatura de 40 (30, 60, 90), muestra una estabilidad similar a la anterior.

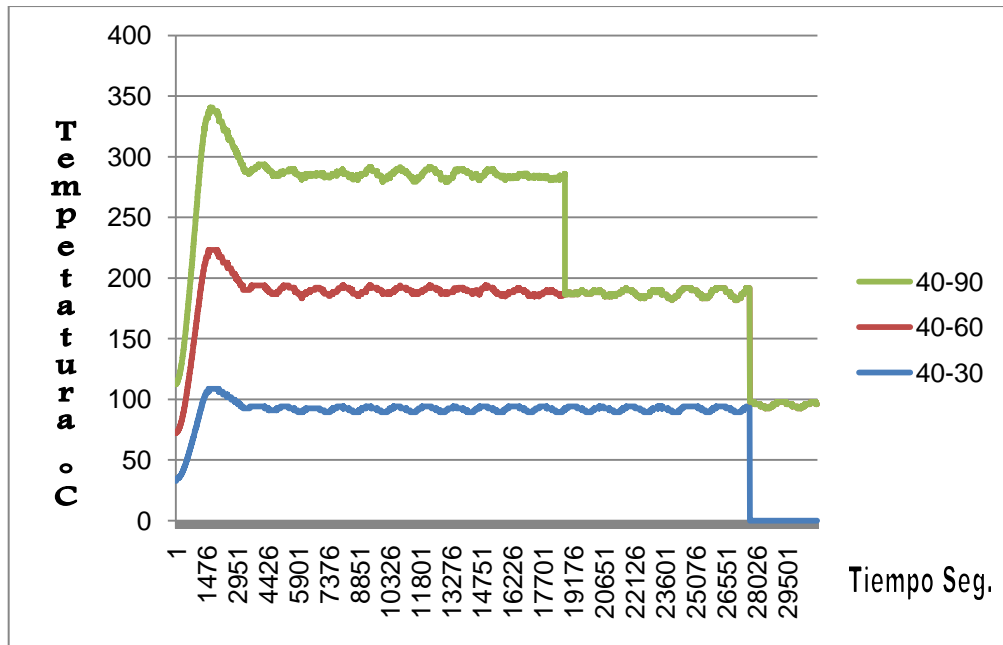


Figura 4.3 Comparación de graficas 40 (30, 60, 90).

Si se analizan las pruebas de la Figura 4.3, estas tienen un tiempo de respuesta en el arranque de 40 minutos. Diferenciado entre ellas por la temperatura de disparo de 16°, 18° y 21° con respecto a la temperatura fijada para los arreglos 40-30, 40-60 y 40-90 respectivamente.

4.1.1 Análisis de graficas

1. Se aprecia que al inicio de las grafica cuando comienza a registrar los datos se tiene una temperatura de disparo hasta en un 20%, para que el sistema tenga una temperatura elevada y así disminuirla de manera gradual hasta estabilizarse.
2. La estabilidad del horno es un poco lenta como se menciona y tiene un rango de tiempo de espera bastante alto.

3. Una característica que se observó, es que el sistema mostró más estabilidad al tener una mayor temperatura.
4. Además, se puede mencionar que las combinaciones posibles fueron 100, pero no se pudieron hacer algunas pruebas con temperaturas más altas ya que el sensor **TMC6-HA** empleado, no soporta altas temperaturas.

Para el análisis de las gráficas, de los datos obtenidos con el control original de la estufa, se tomaron 3 ciclos de cada uno de ellos los cuales se especifican en el Cuadro 4.1 en donde se muestran la descripción estadística de la temperatura registrada de cada ciclo de trabajo.

Se realizó un análisis de regresión, de los datos registrados, empleando el programa estadístico Minitab 14. La ecuación que demostró un comportamiento bastante aceptable con los datos fue con una ecuación cúbica, como se muestra en la Figura 4.4,

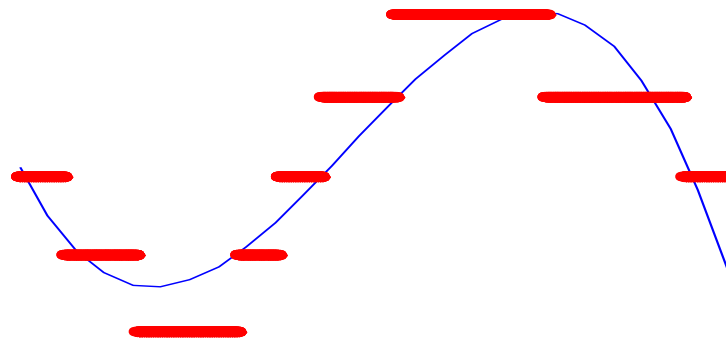


Figura 4.4 Representación del ciclo 1 prueba 30_30.

Como se observa en la gráfica anterior el sistema es bastante aceptable en su comportamiento, dando un comportamiento oscilatorio, dando a entender que el sistema se estabiliza en un determinado tiempo.

Cuadro 4.1 Datos adquiridos en la estufa con el Hobo

SET I	SET II	Datos adquiridos (Segundos)	Periodo de respuesta (Minutos)	Desviación estándar	Varianza	Coeficiente de Variación	Temperatura ° Centígrados		
							Mínimo	Media	Máximo
20	90	7571	42.06	1.38	1.90	2.16	61.29	64.24	65.79
30	30	5695	31.64	1.52	2.29	1.96	74.69	77.83	78.94
	60	6149	34.16	1.80	3.26	2.31	74.69	77.83	80.09
	90	4570	25.39	1.19	1.41	1.57	73.70	75.71	76.76
40	30	4110	22.83	1.71	2.91	1.86	89.30	92.44	94.11
	60	4738	26.32	1.84	3.37	1.92	92.44	95.87	97.72
	90	4931	27.39	1.82	3.32	1.90	92.44	95.87	97.72
50	30	7428	41.27	8.44	71.31	6.56	114.35	128.99	139.24

En el cuadro 4.2 se analizan las formulas de los diferentes ciclos y su comportamiento con las graficas evaluadas con la regresión cubica.

Cuadro 4.2 Registro de ecuaciones cubicas $Y = b + bX + bX^2 + bX^3$

SET I	SET II	Temperatura de referencia °C.	Ecuación	S-Rq. %
30	30	77.83	$= 76.91 - 0.01038 T1 + 0.000019 T1^2 - 0.0 T1^3$	83.5
	60	77.83	$= 75.81 - 0.002688 T2 + 0.000010 T2^2 - 0.0 T2^3$	80.9
	90	75.71	$= 74.13 - 0.003444 T3 + 0.000014 T3^2 - 0.0 T3^3$	85
40	30	92.44	$= 94.11 - 0.02652 T4 + 0.000046 T4^2 - 0.0 T4^3$	89.8
	60	95.87	$= 97.24 - 0.02215 T5 + 0.000035 T5^2 - 0.0 T5^3$	88.2
	90	95.87	$= 96.94 - 0.02047 T6 + 0.000035 T6^2 - 0.0 T6^3$	86.7

En el análisis de regresión para una ecuación cubica se logro obtener el mejor coeficiente de correlación con un máximo de 89.8%, que para ser un sistema de control en lazo abierto es bastante aceptable, demostrando un funcionamiento correcto para las placas de control.

4.2 Simulación en tiempo real.

El programa de termostato digital, como se muestra en la Figura 4.5, empleado para esta simulación fue el desarrollado por Palacios *et al.*, 2008, así mismo el programa una vez verificado y ensamblado con la programación deseada se procedió a evaluar su funcionamiento en el ambiente de ISIS de Proteus.

Los resultados obtenidos en las corridas realizadas mostraron excelente estabilidad en el sistema, aunque las variantes pueden variar, ya teniendo las placas de control impresas.

En la Figura 4.7 se muestra una vista en 3D para tener una vista de cómo va a quedar las placas con sus respectivos componentes.

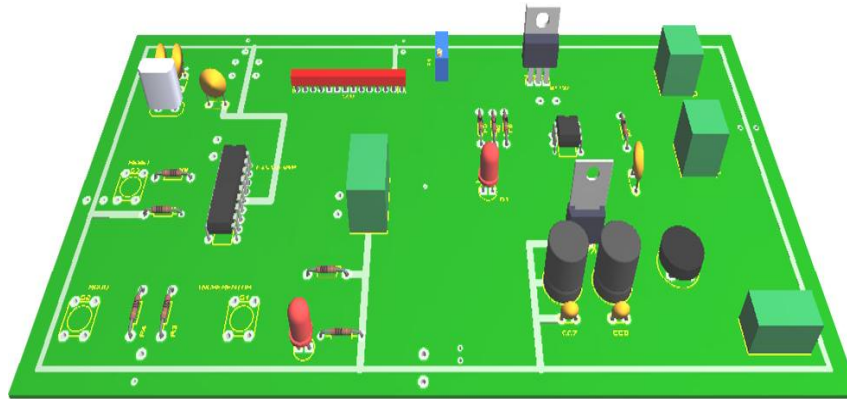


Figura 4.7 Placa impresa realizada en Proteus (Ares) (Vista 3D).

4.3 Pruebas realizadas en el calentador

El sistema como se menciona con anterioridad es para un control automático de temperatura utilizando dos placas de control mostradas en la Figura 4.8.

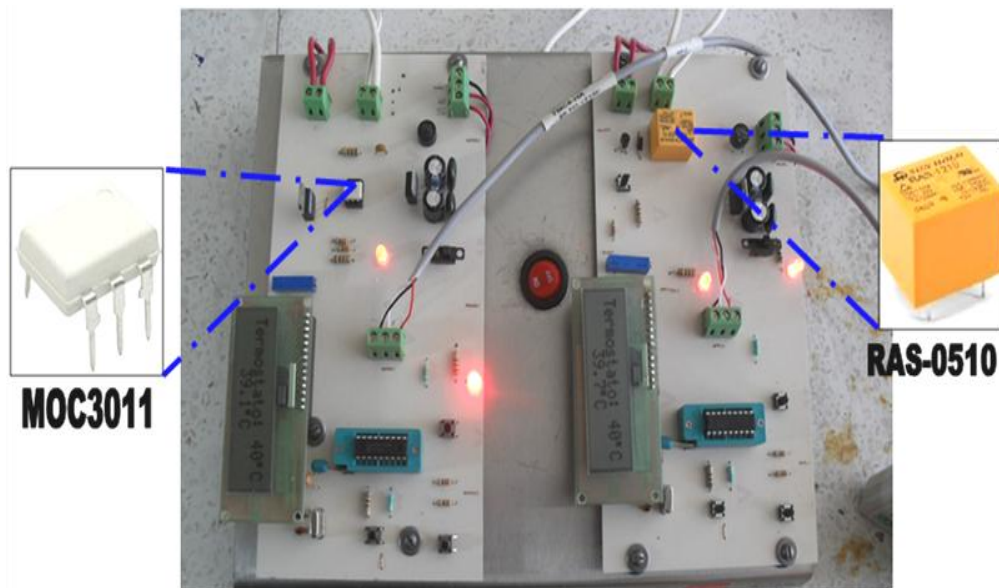


Figura 4.8 Placas de control con diferentes actuadores.

Realizadas con sistemas de control similares, haciéndolas trabajar paralelamente, trayendo consigo un control optimo para ambos sistemas evaluados. Los dos sistemas de potencia empleados fueron el MOC-3011 y el RAS-0510 ambos sistemas funcionan de manera similar ya que el MOC-3011(capacidad de 5 amp) es un optotriac mientras que el RAS-0510 es un relevador (capacidad de 10 amp.).

4.3.1 Sistema controlado por el Relevador electrónico RAS-0510.

En este sistema, el RAS-0510 funciona como un interruptor el cual es comandado por el microcontrolador PIC16F84A dándole las instrucciones necesarias para mantener estable por medio del sensor **DS18B20** que mandaba la señal directamente al microcontrolador y de ahí al relevador para que encienda o apague el calentador.

El sensor de este sistema fue colocado directamente en una placa de metal que se le adapto al calentador, con el fin de que no esté en contacto directo el sensor con la estufa. Así mismo los sensores **TMC6-HA** de los HOBOS se pegaron a la placa como se muestra en la Figura 4.9, con la finalidad de registrar la temperatura.

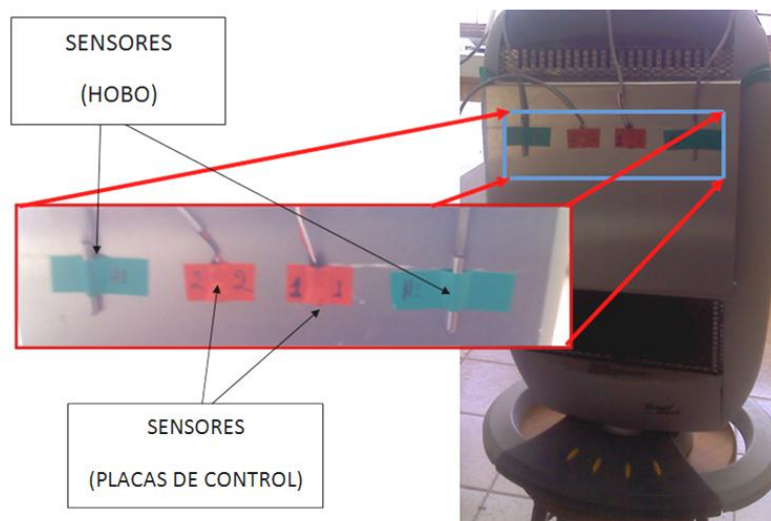


Figura 4.9 Posición de los sensores en el calentador.

Los sistemas fueron programados para que trabajaran de manera paralela. En la Figura 4.10, se muestra la ubicación de los sensores tanto para el relevador electrónico (empleando el sensor DS18B20) como para el registro de datos empleando el data logger HOB0 H08.

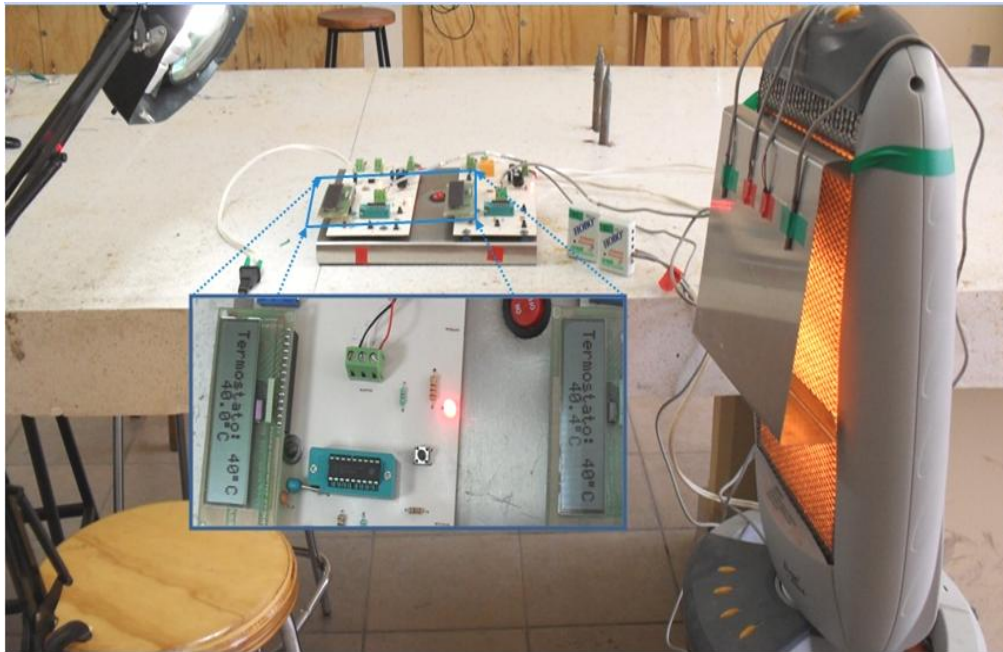


Figura 4.10 Placas sensando de forma paralela.

4.3.2 Análisis de datos con el termostato electrónico.

Para poder comparar el comportamiento de los sistemas de control se hicieron pruebas de 2 a 3 horas cada corrida para evaluar su estabilidad, descrita en la sección 3.4.3, posteriormente se dejaba reposar 2 horas para que la placa se enfriara, a diferencia de la estufa que se dejaba hasta 16 horas para que enfriara (por el tipo de materiales que cada uno lo conforman), posterior a las dos horas nuevamente se realizaba otra corrida demostrada en la Figura 4.11.

Así mismo el data logger HOB0 H08, se programaban nuevamente para registrar los datos correspondientes a esa prueba.



Figura 4.11 Programación manual de las placas.

En la Figura 4.12 se muestran las graficas de resultados con los registros de temperatura de 35°C, 40°C, 45°C, 50°Cy 55°C.

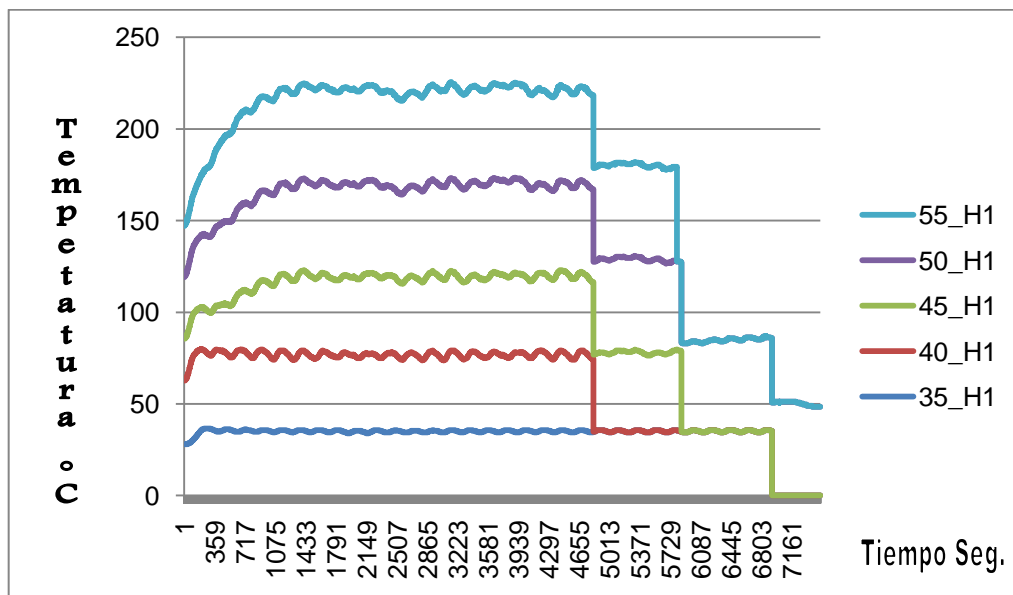


Figura 4.12 Graficas de 35°, 40°, 45°, 50° y 55°.

Los datos registrados de temperatura se presentan en el Cuadro 4.3.

Cuadro 4.3 Control de calentador termostato digital (Datos adquiridos con el hobo)

Prueba °C	HOBO	Datos adquiridos (Segundos)	Periodo de respuesta (Minutos)	Desviación estándar	Varianza	Coeficiente de Variación	Temperatura ° Centígrados		
							Mínimo	Media	Máximo
35	1	733	4.07	0.53	0.28	1.51	34.01	35.27	35.7
	2	672	3.73	0.46	0.21	1.37	33.59	34.43	34.85
40	1	739	4.11	1.32	1.76	3.2	39.22	41.52	43.42
	2	759	4.22	1.29	1.66	3.28	37.44	39.67	41.05
45	1	2106	11.70	0.71	0.51	1.65	42.46	43.42	44.4
	2	1039	5.77	0.69	0.48	1.68	40.59	41.52	42.46
50	1	5059	28.11	1.03	1.07	2.07	48.49	50.66	51.79
	2	2184	12.13	0.95	0.91	2	46.4	47.96	49.02
55	1	4426	24.59	0.34	0.12	0.67	51.22	51.79	52.37
	2	4708	26.16	0.35	0.12	0.78	44.89	45.39	46.4

Como se comento anteriormente se corrieron los datos en el programa estadístico Minitab 14, buscando la ecuación de respuesta que más se adapto a los datos obtenidos y nuevamente la ecuación de tipo cubica mostrados en el Cuadro 4.4, fue la que más se acoplo.

Con los registros estadísticos de las graficas representadas en la Figura 4.12, mostradas anteriormente. El termostato eléctrico presenta una variación de la temperatura programada de $\pm 1^\circ\text{C}$.

Cuadro 4.4 Registro de ecuaciones cubicas $Y = b + bX + bX^2 + bX^3$

Pruebas	Hobbo	Temperatura de referencia °C	Ecuación	R-Sq %
35	1	35.27	$= 34.98 - 0.03584 T1 + 0.000454 T1^2 - 0.000001 T1^3$	89.5
	2	34.43	$= 33.84 - 0.01010 T2 + 0.000269 T2^2 - 0.000001 T2^3$	85.1
40	1	41.52	$= 42.26 - 0.09831 T3 + 0.001122 T3^2 - 0.000003 T3^3$	93.0
	2	39.67	$= 39.75 - 0.08872 T4 + 0.001055 T4^2 - 0.000003 T4^3$	92.4
45	1	43.42	$= 43.65 - 0.01774 T5 + 0.000068 T5^2 - 0.0 T5^3$	94.9
	2	41.52	$= 41.61 - 0.03038 T6 + 0.000234 T6^2 - 0.0 T6^3$	96.2
50	1	50.66	$= 49.06 - 0.002702 T7 + 0.000010 T7^2 - 0.0 T7^3$	91.6
	2	47.96	$= 46.92 - 0.008098 T8 + 0.000039 T8^2 - 0.0 T8^3$	94.2
55	1	51.79	$= 51.96 - 0.000254 T9 + 0.0 T10^2 - 0.0 T10^3$	38.3
	2	45.39	$= 45.01 + 0.000266 T9 - 0.0 T9^2 - 0.0 T9^3$	11.4

En el Cuadro 4.4 podemos ver de manera clara que tiene un mayor porcentaje de confiabilidad en el sistema, a excepción de la prueba 55°C en donde las graficas muestran una linealidad, esto podría deberse a que las ecuaciones de ajuste sean las menos indicadas. Las graficas y cuadros de cada prueba realizadas en Anexo 13.

V CONCLUSIONES Y RECOMENDACIONES

Se logro desarrollar los sistemas de control tanto el software como hardware para los circuitos electrónicos y al mismo tiempo se deja al departamento de Maquinaria Agrícola, un sistema para uso didáctico.

Para los resultados obtenidos en lo que consta a la comparación de los dos sistemas analizados (Estufa y calentador), se llego a la conclusión de que ambos sistemas son buenos pero, con diferentes capacidades de respuesta de control.

En lo que corresponde a la estufa tiene un sistema de control con una respuesta aparentemente lenta (en sus combinaciones analizadas), ya que conforme se iban incrementado las combinaciones la estufa demostró un control mucho más rápido.

En el sistema de termostato electrónico que se diseño para el calentador demostró efectividad con las pruebas de inicio con una reacción de control bastante aceptable, aunque por las limitaciones del calentador, en las pruebas finales hizo falta que incrementara su radiación de calor.

Como recomendaciones, se propondría que se utilizara sensores con más resistencia a temperaturas altas, para así poder realizar pruebas con temperaturas más elevadas (Estufa). Para las placas de control programarse un pulsador de "RETROCESO" para tener un sistema más completo. Aunque lo esencial seria que sea un sistema de control en Lazo Cerrado.

VI BIBLIOGRAFÍAS

- 1) Acedo Sánchez, José (2006). "Instrumentación". Ediciones Díaz de Santos. España.
- 2) Angulo Bahón (2004). Cecilio, Raya Giner Cristóbal. "Tecnología de sistemas de control". Ediciones UP. Barcelona (España).
- 3) Arias Martínez, Oswaldo, Vásquez Sanjuan, Jacob J. (2002). "Sistema de control de temperatura para horno de gas". Artículo publicado por la Universidad Tecnológica de la Mixteca, Huajuapán de León. Oaxaca. (México).
- 4) Bañuelos S. M., Castillo H. J., Rayo L. G., Quintana T. S., Damián Z. R., Pérez S. J. (2005). "Controlador PID de temperatura de tipo didáctico". Proyecto realizado por El Centro de Instrumentos de la Universidad Nacional Autónoma de México (UNAM). México D. F.
- 5) Dallas Semiconductor, DS18B20, Programmable Resolution, 1-Wire Digital Thermometer (2008). (datasheets.maxim-ic.com/en/ds/DS18B20.pdf).
- 6) Hernández Trujillo, Israel, Barajas Martínez, Álvaro (2000). "Diseño de un sistema de control de temperatura para un horno eléctrico, por medio de un controlador ON-OFF". Artículo publicado por la Universidad de Colima (Facultad de ingeniería mecánica y eléctrica). Colima, Colima.
- 7) Kuo, Benjamín C. (1996), Sistemas de control automático 7ª Edición., Prentice-Hall Hispanoamericana S. A., Naucalpan de Juárez, Edo. De México.
- 8) López Campos, Carlos Enrique, Landaverde Muñoz, Octavio (2006). "Sistema de calentamiento solar de agua con control electrónico de temperatura a través de un microcontrolador". Proyecto realizado por La Universidad del Valle de México (UVM) (Campus Querétaro). Querétaro, Querétaro.

- 9) Microchip Technology Inc. (2001). PIC16F84A, Data Sheet, 18-pin Enhanced FLASH/EEPROM, 8-bit Microcontroller. (<http://ww1.microchip.com/downloads/en/devicedoc/35007b.pdf>).
- 10) Ogata, Katsuhiko (2003). "Ingeniería de control moderna". Pearson Educación, S. A. Madrid España. Pags. 3-7.
- 11) Palacios, Enrique, Remiro, Fernando y López, Lucas J. (2008). "Microcontrolador PIC16F84A, Desarrollo de Proyectos". Ediciones Alfaomega, Ra-ma. México D.F. Pags.271-288.
- 12) Pallás Areny, Ramón (1993). "Adquisición y distribución de señales", Marcombo S. A. Barcelona (España).
- 13) Pallás Areny, Ramón (2003). "Sensores y acondicionadores de señal 4ª Edición". Marcombo S. A., Barcelona (España).
- 14) Rina Navarro (2004). "Ingeniería de control analógica digital" Ediciones McGraw-Hill. México D.F.
- 15) Sheldon Manufacturing Inc. "Installation and operation manual", P.O. Box 627 Cornelius, Oregon 97113 (<http://www.Shellab.com/~Shellab>).
- 16) Valdés Pérez, Fernando E., Pallás Areny, Ramón (2007). "Microcontroladores: fundamentos y aplicaciones con PIC", Marcombo S. A. Barcelona (España).
- 17) Vilanova i la Geltru (1998). "Teoría del control electrónico". Ediciones UPC. México D.F.
- 18) www.microdaq.com
- 19) www.steren.com

ANEXOS

ANEXO 1

1.1 Entrada de datos externos y Características del HOBO 4-Channel External.

- Acepta cables y sensores externos de entrada de temperatura, corriente AC, 0-2.5 Voltios DC y 4-20 mA.
- El usuario puede seleccionar Intervalo de muestreo: 0,5 segundos a 9 horas.
- Hora de inicio programable / Fecha.
- Memoria no volátil EEPROM retiene datos aunque la batería se produce un error.
- Parpadeo de luz LED Confirma la operación.
- Indicación de batería de nivel en el lanzamiento.
- Requiere Kit de Software BoxCar para la operación y manipulación de datos.

Cuadro 1.1 Especificaciones del hobo

Capacidad almacenamiento de datos	de	32.520 muestras de 8-bit / Lecturas.
Frecuencia muestreo	de	0,5 segundos a 9 horas (seleccionable).
Rango funcionamiento	de	a) <u>Temperatura</u> : -20 ° C a 70 ° C (-4 ° F a 158 ° F). b) <u>Humedad</u> : 0 a 95% HR sin condensación, no se empañen.
Modos funcionamiento	de	a) Stop Cuando completa. b) Wrap-Around Cuando completa.
Entrada externa		a) <u>Rango</u> : -4 ° F a 158 ° F (-20 ° C a 70 ° C). b) <u>Rango Fuera de caso</u> : -40 ° F a 248 ° F (-40 ° C a 120 ° C). c) <u>Precisión</u> : ± 1,27 ° F (± 0,7 ° C) a 70 ° F. d) <u>Resolución</u> : 0,7 ° F (0,4 ° C) a 70 ° F. e) <u>Tiempo de respuesta (sensor dentro de la caja)</u> : 15 min. f) <u>Tiempo de respuesta (sensor Fuera de caso)</u> :

	1 min.
Compatible tipos de entrada Externa	<ul style="list-style-type: none"> a) Sensores de temperatura (# TMCX-Hx). b) Split TC básicos para corriente alterna (# CTV-X). c) Transmisores de tensión (# T-CON-ACT-X). d) 4-20mA de entrada de cable (# CABLE-4-20mA). e) 0 a 2,5 voltios de corriente continua Cable de entrada (# CABLE-2-5-STEREO). f) Dióxido de Carbono para el monitor Telaire (# CABLE-CO2).
Externos de entrada 2.5V	<ul style="list-style-type: none"> a) <u>Jack de 2,5 mm</u>: Planta de entrada externa, de entrada, salida conmutada 2,5 voltios. b) Conexión de entrada a tierra externa no es lo mismo que la interfaz de conexión a tierra de PC y no puede conectar a cualquier tierra externo. c) <u>rango de entrada</u>: 0 a 2,5 voltios DC. d) <u>Precisión</u>: $\pm 10 \text{ mV} \pm 1\%$ de la lectura. e) <u>Resolución</u>: 10mV (8-bits). f) <u>Potencia de salida</u>: $\pm 2,5 \text{ voltios DC @ } 2 \text{ mA}$, activo únicamente durante las mediciones.
Drop Prueba	Hasta 5 pies.
Exactitud del tiempo	± 1 minuto por semana ($\pm 100 \text{ ppm}$ a 20° C o 68° F).
De Almacenamiento	<u>Temperatura</u> : -40° C a 75° C (-40° F a 167° F).
Batería	CR 3-V-2032 Batería de litio (reemplazables por el usuario).
Duración de la batería	Por lo general 1 año.
Cumplimiento de los estándares	CE.
Peso	29 gramos (1 onza).
Dimensiones	61mm x 48mm x 20 mm (2,4 "x 1,9" x 0,8 ").

ANEXO 2

2.1 Especificaciones de los sensores de temperatura exterior TMC6-HA.

- Rango: -40 ° a 50 ° C (-40 ° a 122 ° F) en el agua o el suelo, -40 ° a 100 ° C (-40 ° a 212 ° F) en el aire
- La precisión con los madereros H8: $\pm 0,5$ ° a 20 ° C ($\pm 0,9$ ° a 68 ° F), ver diagrama a continuación A1 (0,9 introduzca la sonda "mín.)
- Drift: $<0,1$ ° C (0,2 ° F) por año
- Resolución con los madereros H8: 0,41 ° a 20 ° C (0,7 ° F a 68 ° F).
- Tiempo de respuesta en el aire: 3 minutos típica a 90% en el aire que se mueve 1 m / seg (2,2 mph)
- El tiempo de respuesta en agitó el agua: 1 minuto, típica del 90%
- Disponible en 1,8 m, 6,1 m, y 15.2 m longitud de los cables pies (6, 20 y 50 pies)
- Encapsulado: punta de acero inoxidable del sensor.
- Dimensiones de la sonda: 0,5 cm x 2,5 cm (0,20 "x 1,0").
- Peso: 37g (1,3 oz), 125 g (4,4 oz), o 280g (9,9 onzas).

La punta del detector y la inmersión en agua dulce de cable de hasta 50 ° C (122 ° F) durante un año; escudo de radiación altamente recomendable para su uso en la luz del sol.

ANEXO 3

3.1 Características del BoxCar 4.3

Es una aplicación de Windows para registrar y la gestionar datos utilizando registradores de datos HOBO o polizón. BoxCar Pro 4.3, es una versión mejorada que ofrece características adicionales para la representación gráfica, análisis de datos, exportación de datos y la gestión simultánea de múltiples registradores.

3.2 Pasos de Ajuste y programación

- Selección de los intervalos de muestreo predefinidas (0.5 segundos a 9 horas) o el programa se auto programa.
- Establezca el tiempo de inicio y los modos de memoria.
- Verifique la operación del registrador antes de programar.
- Sincronizar de los relojes y el reloj de la computadora. Para el inicio de registro de datos
- Los controles estado de la batería, mostrada en Figura 3.1.

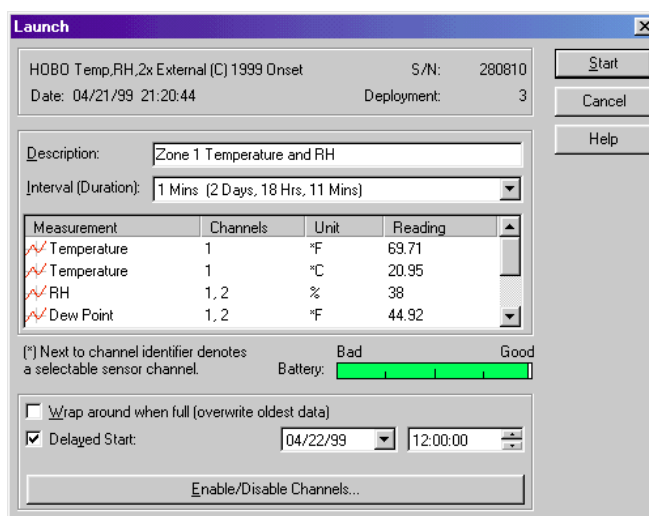


Figura 3.1 Ajuste fácil del Logger en BoxCar Pro.

3.3 Forma de graficado

Permiten que se pueda comparar varios parámetros en un gráfico, incluidos los datos de los despliegues sucesivos. A continuación, utilice el zoom y el control de las herramientas de eje para centrarse en los datos de interés.

- Añadir nuevas series de datos de los archivos almacenados o arrastrar y soltar archivos desde una parte a otra, múltiples eje de valores en un gráfico, como la temperatura y humedad relativa.
- Superposición de datos de diferentes implementaciones, para comparar mes a mes, o antes y después, con el cursor se puede mostrar valores específicos en una lista independiente.
- Control ejes, series y propiedades leyenda, los registra por intervalo de tiempo Y número de eventos. El control de la barra de herramientas para Microsoft Excel de exportación (. TXT). (Figura 3.2).

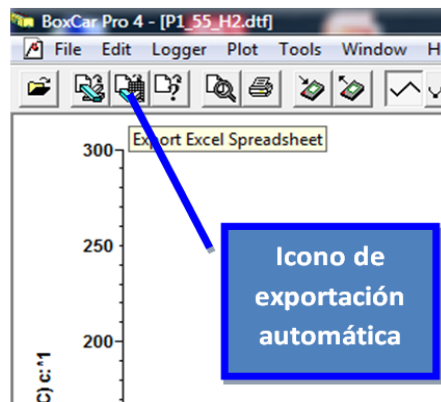


Figura 3.2 Exportación de BoxCAR Pro a Excel.

3.4 Requisitos del sistema

El requisito mínimo es un sistema PC con Windows ® XP/2000/Me/98/Vista. El sistema debe tener al menos un puerto serie disponible COM, una pantalla mínima de color de 16-bit y 10 MB de espacio disponible en disco.

ANEXO 4

4.1 Especificaciones del USB Serial Adapter RS-232

- Se conecta a cualquier puerto USB disponible en un PC con Windows.
- Proporciona un puerto RS-232 DB9 macho para la conexión directa a dispositivos de serie.
- Llama la energía de la conexión USB - no requiere adaptador de energía.
- Fácil de llevar - pesa sólo 2 onzas (57 gramos).
- 0 a 40 ° C (32 a 104 ° F), 0-95% RH rango de operación.

4.2 Requisitos del sistema

- PC con Windows 98SE/ME/2000/XP/VISTA.
- Al menos un puerto USB disponible.

Este dispositivo es una conexión entre el HOBO y la PC, cabe mencionar que este mismo adaptador se utiliza en Picstart Plus para poder programar los microcontroladores.

ANEXO 5

5.1 Generalidades del PIC16F84a

El microcontrolador tiene un Repertorio de 35 Instrucciones.

- Todas las instrucciones se ejecutan en un solo ciclo excepto las de salto que necesitan dos.
- Versiones para bajo consumo (16LF84A), de 4 MHz (PIC16F84A-04) y 20 MHz (PIC16F84A-20). Un ciclo máquina del PIC son 4 ciclos de reloj, por lo cual si tenemos un PIC con un cristal de 4 MHz, se ejecutarán 1 millón de instrucciones por segundo.
- Memoria de programa Flash de 1 K x 14 bits.
- Memoria RAM dividida en 2 áreas: 22 registros de propósito específico (SFR) y 68 de propósito general (GPR) como memoria de datos.
- 15 registros de funciones especiales.
- Memoria de datos RAM de 68 bytes (68 registros de propósito general).
- Memoria de datos EEPROM de 64 bytes.
- Contador de programa de 13 bit (lo que en teoría permitiría direccionar 4 KB de memoria, aunque el 16F84 solo dispone de 1KB de memoria implementada).
- Pila con 8 niveles de profundidad.
- Modos de direccionamiento directo, indirecto y relativo.
- ALU de 8 bits y registro de trabajo W del que normalmente recibe un operando que puede ser cualquier registro, memoria, puerto de Entrada/Salida o el propio código de instrucción.
- 4 fuentes de interrupciones:
 - A través del pin RB0/INT.
 - Desbordamiento del temporizador TMR0.
 - Interrupción por cambio de estado de los pins 4:7 del Puerto B.
 - Completada la escritura de la memoria EEPROM.
- 1.000.000 de ciclos de borrado/escritura de la memoria EEPROM.
- 40 años de retención de la memoria EEPROM.

- 13 pins de E/S con control individual de dirección.
- PortA de 5 bits <RA0:RA4>.
- PortB de 8 bits <RB0:RB7>.
- Contador/Temporizador TMR0 de 8 bits con divisor programable.
- Power-on Reset (POR).
- Power-up Timer (PWRT).
- Oscillator Start-up Timer (OST).
- Watchdog Timer (WDT).
- Protección de código.
- Modo de bajo consumo SLEEP.
- Puede operar bajo 4 modos diferentes de oscilador.
- Programación en serie a través de dos pins.
- Tecnología de baja potencia y alta velocidad CMOS Flash/EEPROM.
- Características eléctricas máximas (no deben ser superadas y de mantenerse por un tiempo en algún máximo puede dañarse al PIC)
 - Temperatura ambiente máxima para funcionamiento de -55°C to $+125^{\circ}\text{C}$.
 - Tensión máxima de VDD respecto a VSS de $-0,3$ a $+7,5\text{V}$.
 - Tensión de cualquier patilla con respecto a VSS (excepto VDD, MCLR, y RA4) de $-0,3\text{V}$ a $(\text{VDD} + 0.3\text{V})$.
 - Tensión en MCLR con respecto a VSS $-0,3$ a $+14\text{V}$.
 - Tensión en RA4 con respecto a VSS $-0,3$ a $+8,5\text{V}$.
 - Disipación de potencia total de 800 mW .
 - Máxima corriente de salida a VSS 150 mA .
 - Máxima corriente de salida de VDD 100 mA .
 - Máxima corriente del puerto "A" como fuente, 50 mA .
 - Máxima corriente del puerto "A" como sumidero, 80 mA .
 - Máxima corriente del puerto "B" como fuente, 100 mA .
 - Máxima corriente del puerto "B" como sumidero, 150 mA .
 - Máxima corriente que puede suministrar una sólo salida como fuente o sumidero, 25 mA .

- Rango de alimentación:
 - 16LF84A: de 2 a 5,5 V en configuración de oscilador XT, RC y LP.
 - 16F84A:
 - de 4 a 5,5 v en configuración de oscilador XT, RC y LP.
 - de 4,5 a 5.5 v en configuración de oscilador HS.
- Consumo típico:
 - 16LF84A:
 - de 1 a 4 mA en configuración de oscilador RC y XT (FOSC=2 MHz, VDD=5,5V).
 - de 15 a 45 μ A en configuración de oscilador LP (FOSC=32kHz, VDD=2V, WDT deshabilitado).
 - 16F84A:
 - de 1,8 a 4.5 mA en configuración de oscilador RC y XT (FOSC=4 MHz, VDD=5,5V).
 - de 3 a 10 mA en configuración de oscilador RC y XT durante la programación de la FLASH (FOSC=4MHz, VDD=5,5V).
 - 16F84A-20: de 10 a 20 mA en configuración de oscilador HS (FOSC=20 MHz, VDD=5,5V).

5.2 Arquitectura interna

Las altas prestaciones de los microcontroladores PIC derivan de las características de su arquitectura. Están basados en una arquitectura tipo Harvard que posee buses y espacios de memoria por separado para el programa y los datos, lo que hace que sean más rápidos que los microcontroladores basados en la arquitectura tradicional de Von Neuman.

Otra característica es su juego de instrucciones reducido (35 instrucciones) RISC, donde la mayoría se ejecutan en un solo ciclo de reloj excepto las instrucciones de salto que necesitan dos. Posee una ALU (Unidad Aritmético

Lógica) de 8 bits capaces de realizar operaciones de desplazamientos, lógicas, sumas y restas. Posee un Registro de Trabajo (W) no direccionable que usa en operaciones con la ALU. (Figura 5.1)

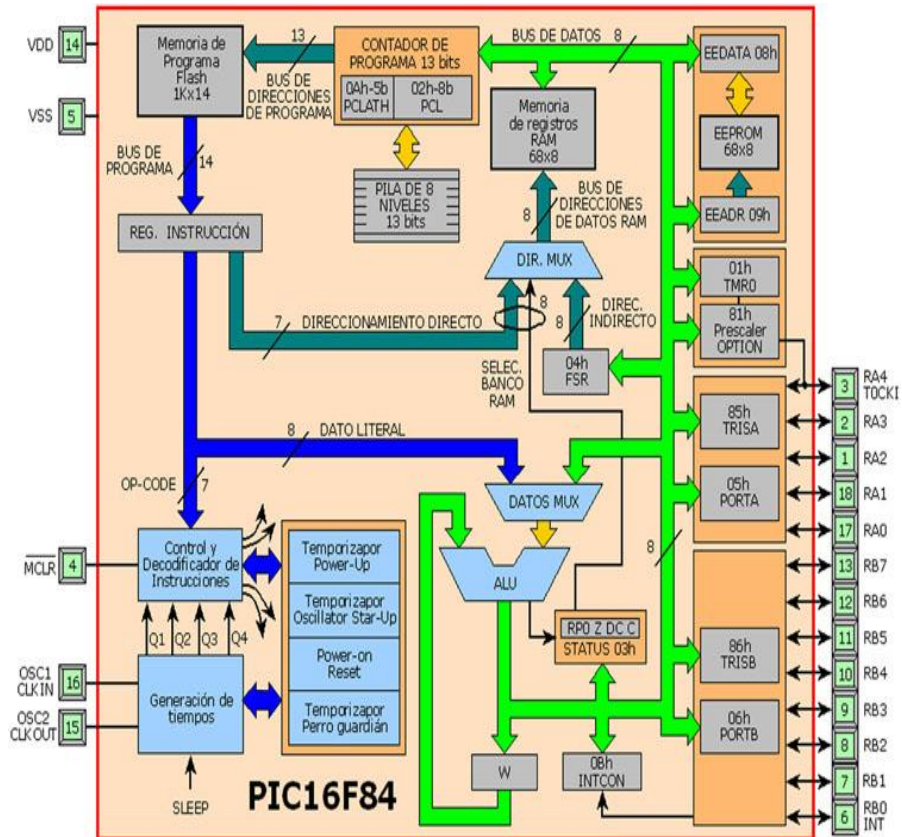


Figura 5.1 Estructura interna del PIC16F84A

5.3 Memoria de programa

La memoria de programa está organizada con palabras de 14 bits con un total de 1 K, del tipo Flash, que durante el funcionamiento es de solo lectura. Sólo se ejecutará el código contenido en esta memoria, pudiendo almacenar en ella una cantidad limitada de datos como parte de la instrucción RETLW. En una sola palabra se agrupa el código de la instrucción y el operando o su dirección esto puede visualizarse en la Figura 5.2.

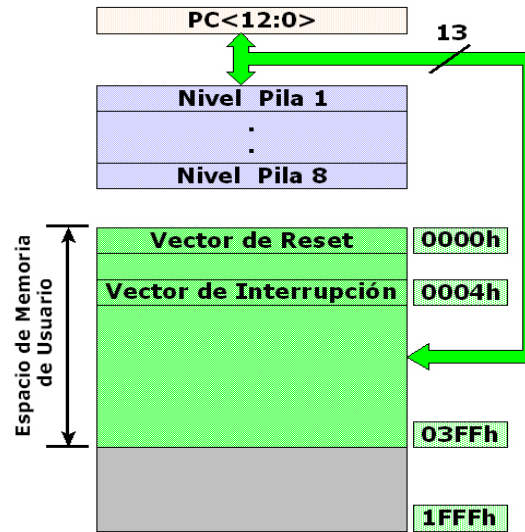


Figura 5.2 Composición de la memoria

El tipo de memoria utilizada en este microcontrolador, podrá ser grabada o borrada eléctricamente a nuestro antojo desde el programador. La memoria tipo Flash tiene la característica de poderse borrar en bloques completos y no podrán borrarse posiciones concretas o específicas. Este tipo de memoria no es volátil, es decir, no pierde los datos si se interrumpe la energía, la memoria para almacenar el programa nos resultará perfecta para realizar pruebas y experimentos, además de para la programación "on-board" o "in-circuit", esto es, nos permite la programación del dispositivo o actualización del programa sin necesidad de retirarlo del circuito donde va montado, existen varias versiones de memoria de programa para los PIC16f84A:

- **Versión Flash**, Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. A diferencia de la memoria de tipo ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM. Esta versión es idónea para la enseñanza y la Ingeniería de diseño.
- **Versión OTP** ("One Time Programmable") "Programable una sola vez", Sólo se puede grabar una vez por el usuario sin la posibilidad de borrar

lo que se graba. Resulta mucho más económica en la implementación de prototipos y pequeñas series.

- **Versión QTP**, Es el propio fabricante el que se encarga de grabar el código en todos los chips que configuran pedidos medianos y grandes.
- **Versión SQTP**, El fabricante solo graba unas pocas posiciones de código para labores de identificación, número de serie, palabra clave, checksum, etc.

5.4 Memoria de datos

Está organizada en dos páginas o bancos de registro, banco 0 y banco 1. Para cambiar de página se utiliza un bit del registro STATUS (RP0).

Cada banco se divide a su vez en dos áreas: RFS (Registros de Funciones Especiales) y RGP (Registros de Propósito General)

En la Figura 5.3 nos podemos hacer una idea de cómo están distribuidos:

Dir. de registro	BANCO 0	BANCO 1	Dir. de registro
00h	Dir. Ind. ¹	Dir. Ind. ¹	80h
01h	TMRO	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	-	-	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2 ¹	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	68	MAPEADOS (ACCESO) EN EL BANCO 0	8Ch
4Fh	REGISTROS DE PROPÓSITO GENERAL		CFh
50h			D0h
7Fh			FFh

Localización de memoria no implementada, se lee como '0'

Nota 1: No es un registro físico

Figura 5.3 Registros de datos

La primera es la de RFS (Registros de Funciones Especiales) que controlan el funcionamiento del dispositivo. Estos se emplean para el control del funcionamiento de la CPU y de los periféricos.

El segundo área (68 bytes SRAM) es la de RGP (Registros de Propósito General), y puede accederse a ellos tanto directa como indirectamente haciendo uso del registro FSR.

Banco 0:

- Este banco está formado por 80 bytes, desde la posición 00 hasta la 4Fh (de la 0 a la 79).
- El área RFS consta de 12 registros que serán utilizados por funciones especiales del microcontrolador. Comienza en la dirección 00h y termina en la 0Bh, es decir, de la 0 a la 11.
- El Área RGP consta de 68 registros de memoria RAM que serán utilizados para almacenar datos temporales requeridos por los programas. Comienza en la dirección 0Ch y termina en la posición 4Fh (de la 12 a la 79). Esta parte es la memoria de registros de propósito general.

Banco 1:

- Este banco tiene las mismas dimensiones que el anterior, pero su uso es menor, ya que no tiene banco para registros de propósito general. Solamente tiene una sección de registros especiales que van de la posición 80h a la 8Bh (de la 128 a la 139).

La memoria RAM así como algunos registros especiales son los mismos en los dos bancos del mapa de memoria del PIC. La anchura de los bytes en la memoria es de 8 bits.

5.5 Descripción de las patillas o pines del microcontrolador (Tabla 5.1)

Disposición de patillas para encapsulado DIL 18 (Figura 5.4).

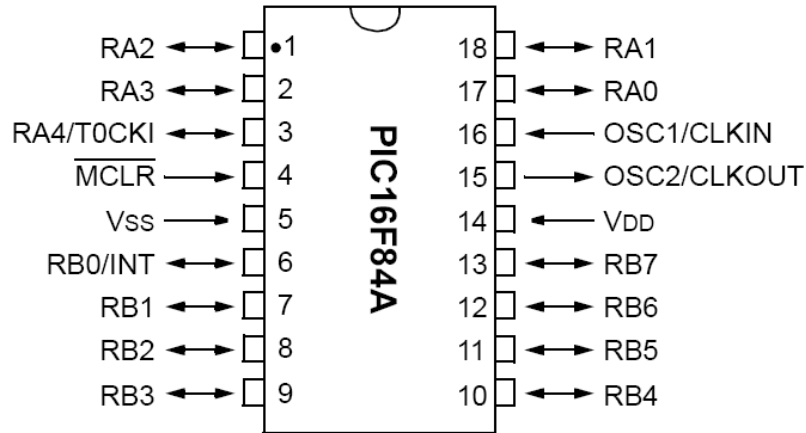


Figura 5.4 Pines del PIC16F84A

Cuadro 5.1 Descripción de patillas o pines

Nombre	Nº	Tipo	Descripción
OSC1/CLKIN	16	I	Entrada del oscilador a cristal/Entrada de la fuente de reloj externa
OSC2/CLKOUT	15	O	Salida del oscilador a cristal. En el modo RC, es una salida con una frecuencia de $\frac{1}{4}$ OSC1
MCLR	4	I/P	Reset/Entrada del voltaje de programación.
RA0	17	I/O	Puerto A bidireccional, bit 0
RA1	18	I/O	Puerto A bidireccional, bit 1
RA2	1	I/O	Puerto A bidireccional, bit 2
RA3	2	I/O	Puerto A bidireccional, bit 3
RA4/T0CKI	3	I/O	También se utiliza para la entrada de reloj para el TMR0
RB0/INT	6	I/O	Puerto B bidireccional, bit 0 Puede seleccionarse para entrada de interrupción externa

RB1	7	I/O	Puerto B bidireccional, bit 1
RB2	8	I/O	Puerto B bidireccional, bit 2
RB3	9	I/O	Puerto B bidireccional, bit 3
RB4	10	I/O	Puerto B bidireccional, bit 4 Interrupción por cambio de estado
RB5	11	I/O	Puerto B bidireccional, bit 5 Interrupción por cambio de estado
RB6	12	I/O	Puerto B bidireccional, bit 6 Interrupción por cambio de estado
RB7	13	I/O	Puerto B bidireccional, bit 7 Interrupción por cambio de estado
Vss	5	P	Tierra de referencia
Vdd	14	P	Alimentación

5.6 Otras características

Dentro del microcontrolador hay unos registros especiales que determinan algunas de las características notables del microcontrolador:

- Temporizador/Contador TMR0
- Perro guardián watch Dog (WD)
- Interrupciones.
- Reset (Reinicio del sistema).

ANEXO 6

6.1 Especificaciones del sensor 18B20

- Único 1-Wire[®] interfaz requiere un solo pin del puerto para la Comunicación
- Cada dispositivo tiene un único código de serie-Bit 64 almacenados en sobre una Junta-ROM.
- Capacidad Multidrop Simplifica la temperatura distribuida de detección de aplicaciones, no requiere componentes externos
- Puede suministrar energía de línea de datos.
- La fuente de alimentación Gama es 3.0V a 5.5V
- Medidas temperaturas de -55 ° C a +125 ° C (-67 ° F a 257 ° F)
- ± 0,5 ° C Precisión de -10 ° C a +85 ° C
- Termómetro de la Resolución es seleccionable por el usuario 9 a 12 Bits
- Temperatura Convierte a Bit Digital Palabra-12 en 750ms (Max)
- Comando de búsqueda de alarma identifica
- Trata los productos cuya temperatura es programada Límites exterior (condición de alarma de temperatura)
- Software compatible con el DS1822
- Las aplicaciones que se pueden beneficiar de esta característica son los controles de climatización ambiental, sistemas de control de la temperatura interior de los edificios, equipos o maquinaria, y el seguimiento de procesos en sistemas de control.

ANEXO 7

Procedimiento de programación y quemado del microcontrolador

7.1 Llamado del programa que se utilizara (primero abrir MPLab) para la programación (ya teniendo el programa corriendo) (Figura 7.1).

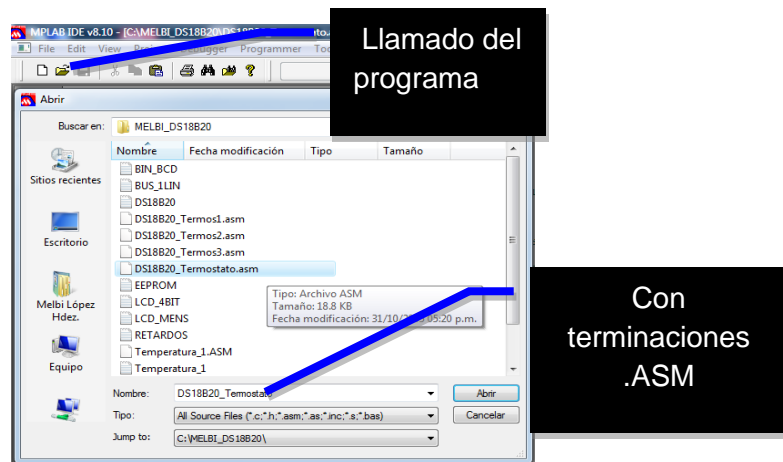


Figura 7.1 Llamado del programa con terminación .ASM

7.2 Ensamblaje del programa representado en la Figura 7.2, debe estar sin errores (No muestra mensajes de errores), si nuestro programa muestra error de ensamblaje (mensaje en rojo en MPLab), checar nuevamente el programa.

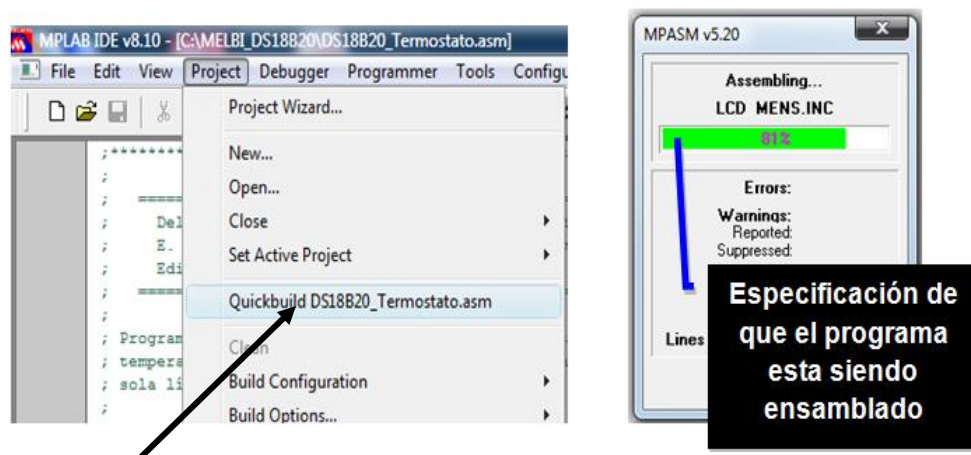


Figura 7.2 Ensamblado del programa

7.3 Programador Picstart Plus, para poder enlazar el programador con la PC se selecciona un comando como se muestra en la Figura 7.3, pero antes debe colocarse el microcontrolador en el ensamblador Picstart Plus.

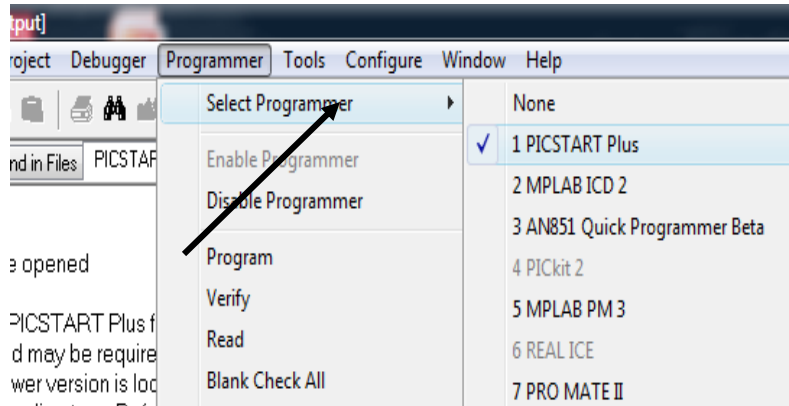


Figura 7.3 Llamado del programador a MPLab

7.4 Permitir el acceso del programador para que se pueda utilizar de manera correcta, comando mostrado Figura 7.4, de otra manera no lee el programador, e inversamente se puede quitar la conexión. Ya depende del usuario.

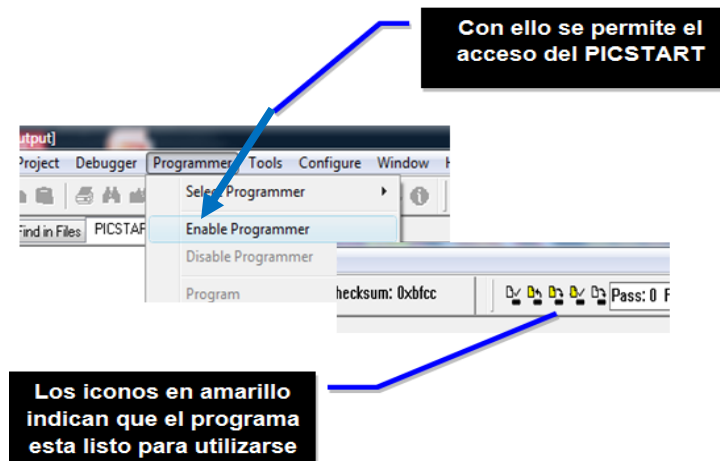


Figura 7.4 Acceso de Picstart Plus

7.5 Programación del microcontrolador de manera directa. (Figura 7.5), este icono es una manera directa para la programación del microcontrolador. Al activarlo, en el Picstart Plus muestra una luz verde que indica que esta encendido y cuando esta programando, tiene otro led que empieza a parpadear, esto indica que esta siendo programado el microcontrolador.

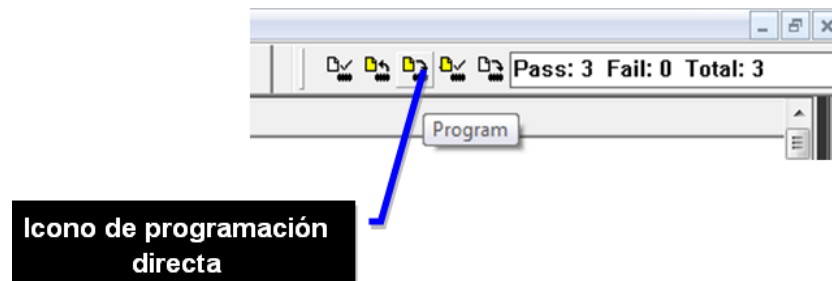


Figura 7.5 Programación directa

7.6 Verificación del programa. Si quedo correctamente ensamblado en el PIC16F84A, puede verse el programa en pantalla, Figura 7.6.

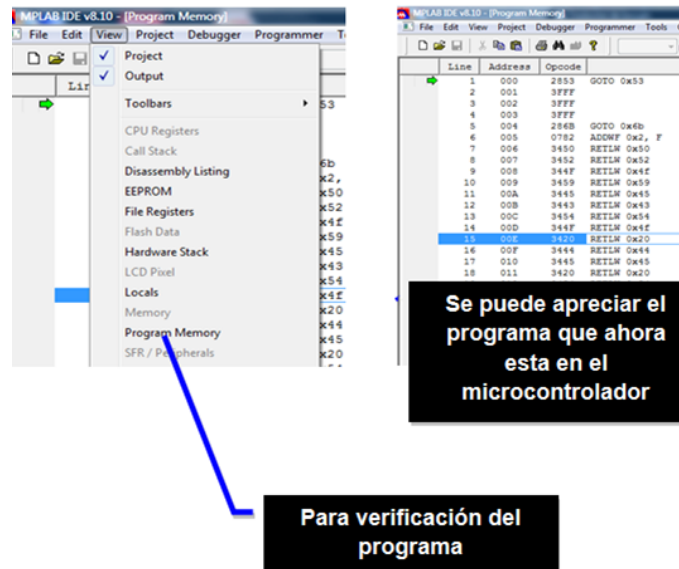


Figura 7.6 Verificación de la programación

7.7 El microcontrolador está listo para utilizarse

Como se observa en la Figura 7.7 puede observarse la posición de los microcontroladores en las placas de control, previamente ensamblados en MPLab y programados en el PicStar Plus.



Figura 7.7 Posición de los microcontroladores en las placas de control

Después de haber ensamblado el programa en el microcontrolador están listos para realizar su función en las placas de control.

Nota importante: Algo indispensable que se debe tomar en cuenta es que el programa debe estar guardado en la "unidad de disco C", de otra manera no correrá el programa.

ANEXO 8

8.1 Proteus

Es un software de diseño electrónico desarrollado por Labcenter Electronics que consta de dos módulos: Ares e Isis y que incluye un tercer módulo opcional denominado Electra.

8.1.1 Isis

Mediante este programa podemos diseñar el circuito que deseemos con componentes muy variados, desde una simple resistencia hasta algún que otro microprocesador o microcontrolador, incluyendo fuentes de alimentación, generadores de señales y muchas otras prestaciones (Figura 8.1).

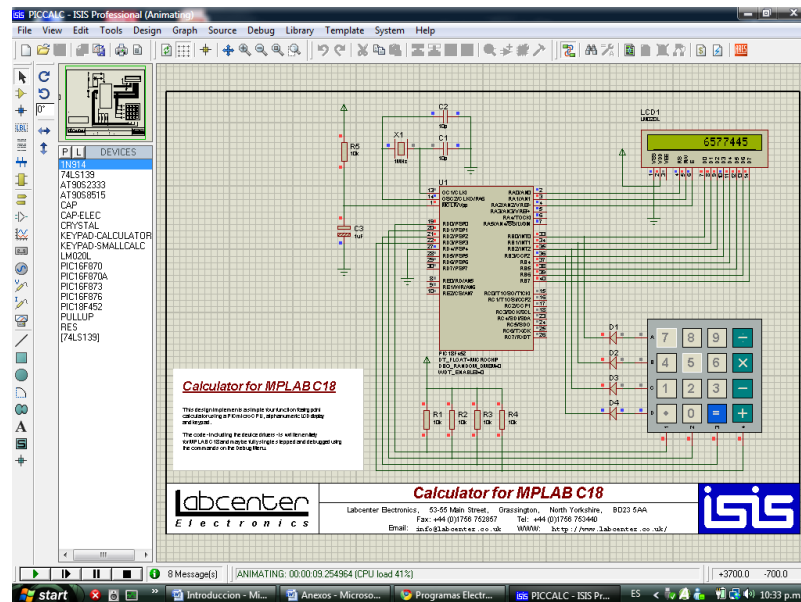


Figura 8.1 Ambiente Proteus ISIS

Los diseños realizados en Isis pueden ser simulados en tiempo real.

8.1.2 Modulo VSM

Una de las prestaciones de Proteus, integrada con ISIS, es VSM, el Virtual System Modeling (Sistema Virtual de Modelado), una extensión integrada con ISIS, con la cual se puede simular, en tiempo real, con posibilidad de más rapidez; todas las características de varias familias de microcontroladores, introduciendo nosotros mismos el programa que controlará el microcontrolador y cada una de sus salidas, y a la vez, simulando las tareas que queramos que lleve a cabo con el programa. Se pueden simular circuitos con microcontroladores conectados a distintos dispositivos, como motores, lcd's, teclados en matriz, etc. Incluye, entre otras, las familias de PIC's PIC10, PIC12, PIC16, PIC18, PIC24 y dsPIC33. ISIS es el corazón del entorno integrado PROTEUS. Es mucho más que un simple programa de dibujo de esquemas electrónicos.

8.1.3 Ares

Ares o Advanced Routing and Editing Software (*Software de Edición y Ruteo*) es la herramienta de rutado de Proteus, se utiliza para la fabricación de placas de circuito impreso

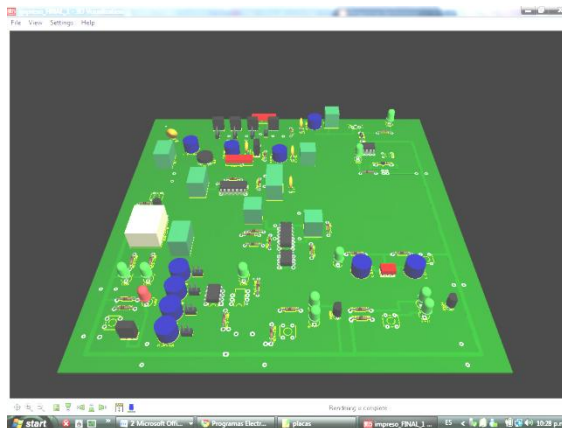


Figura 8.2 Ambiente Proteus ARES

ANEXO 9

Simulación en Proteus (ISIS)

9.1 Cargar el programa Proteus (ISIS), previo de haber abierto ISIS, se carga el circuito (Figura 9.1), para tenerlo en pantalla como se muestra en la Figura 9.2.

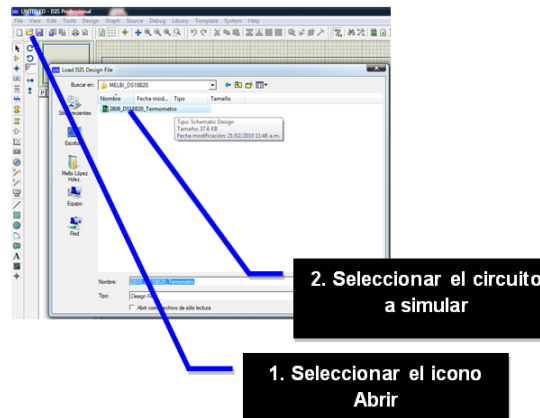


Figura 9.1 Cargando el circuito

9.2 Simulación, ya teniendo el circuito electrónico en la pantalla se llama un programa externo (que con anterioridad se probó en MPLab), a continuación se muestra como llamar el programa en la Figura 9.2. Después de seleccionarlo nos aparece un submenú (Figura 9.3).

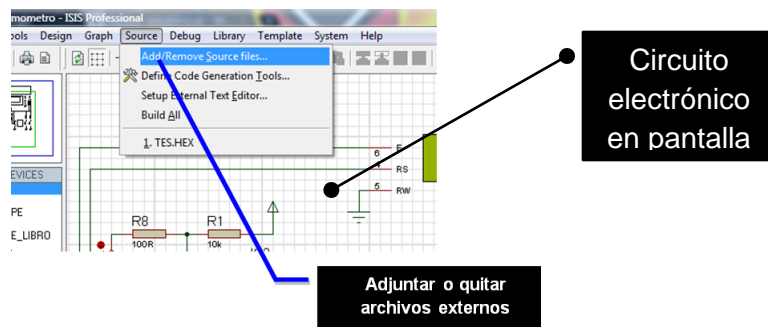


Figura 9.2 Cargar programa para simulación

9.3 Instrucciones para el funcionamiento correcto en la simulación

Al seleccionar new, aparece el submenú (Figura 9.4), se selecciona el programa a utilizar, posteriormente se da abrir, al tener ya el programa seleccionado y con las instrucciones correctas se da Ok como se representa en la Figura 9.3.



Figura 9.3 Comandos esenciales.

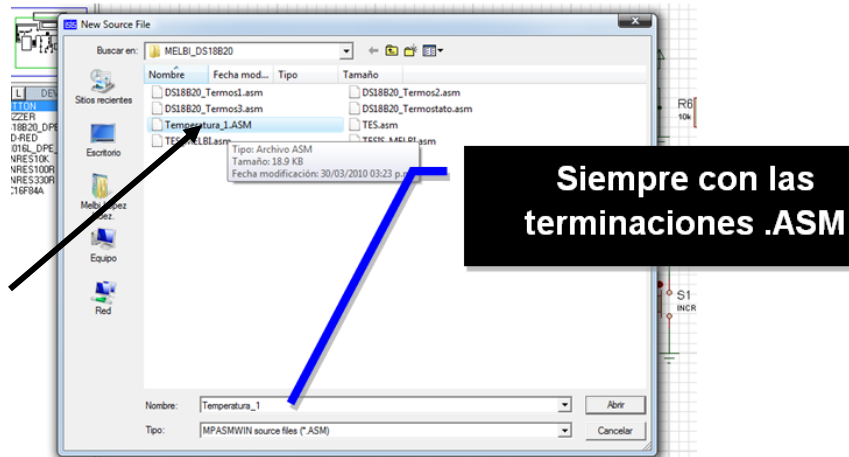


Figura 9.4 Submenú para selección del programa

Así automáticamente se carga el programa al circuito a simular.

9.4 Inicio de simulación, como se muestra en la Figura 9.5, este icono inicia automáticamente la simulación, además pueden observarse otros controles de pausa, pausa-play y stop (detiene la simulación).

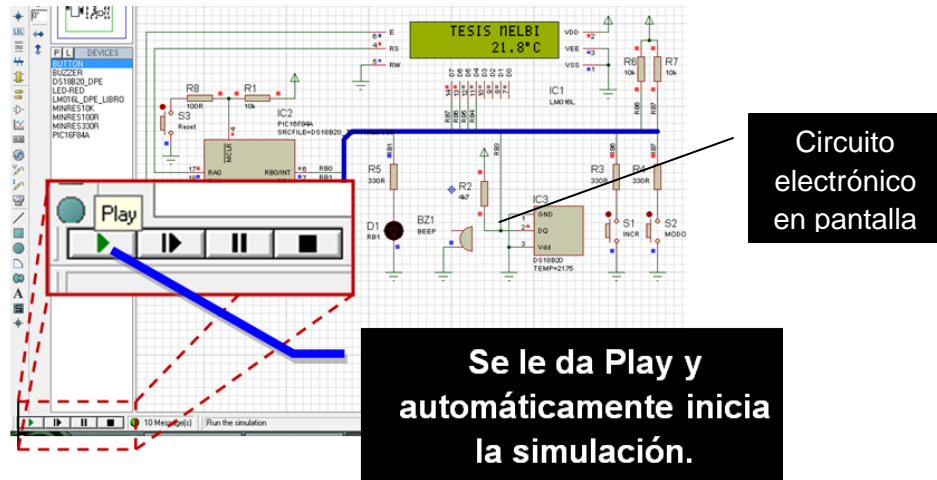


Figura 9.5 Simulación

Nota: Si la simulación no comienza y muestra error, realizar los pasos mencionados anteriormente o verificar el programa en MPLab (Anexo 7). Posterior a esto intentar de nuevo.

ANEXO 10

10.1 Funciones de la placa de control

Para comenzar a explicar las funciones de la placa nos vamos directamente al programa que se utilizó en el microcontrolador.

El programa de control para un termómetro y termostato digital. Utiliza el sensor de temperatura DS18B20 que transmite la información vía serie a través de un bus de una sola línea según un protocolo del fabricante de semiconductores Dallas Semiconductors.

El ajuste de la temperatura a la que conmuta el termostato se logra mediante tres pulsadores como se ve en la Figura 10.1 y son: "MODO" e "INCREMENTAR", además un pulsador de "RESET" para reinicio del programa, que se conectan a pines del Puerto B y cuyo funcionamiento se basa en interrupción por cambio en la línea del Puerto B.

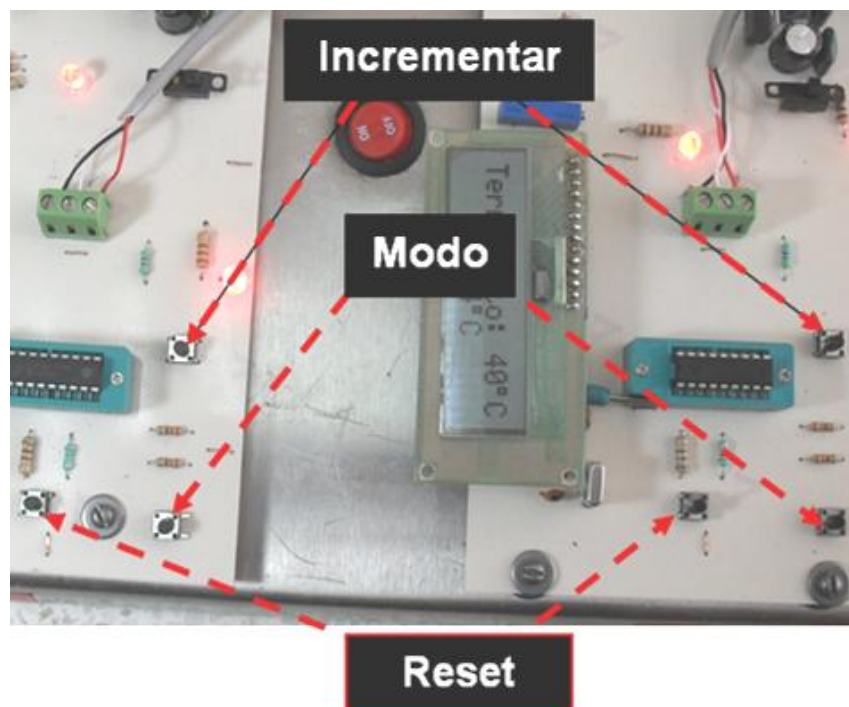


Figura 10.1 Posición de los pulsadores en las placas de control

10.2 Pulsador Modo

En esta sección se especifica el funcionamiento del pulsador modo, este se basa en tres funciones, con subrutinas mostrada en la Figura 10.2.

10.2.1 Modo "Termostato_OFF", donde funciona como termómetro normal sin termostato. Se reconoce por el flag Termostato_OFF.

10.2.2 Modo "Termostato_Ajuste", donde se ajusta la temperatura deseada cuando funcione como termostato. Se reconoce por el flag Termostato_Ajuste.

10.2.3 Modo "Termostato_ON", donde funciona como termómetro normal con termostato. Se reconoce por el flag Termostato_ON.

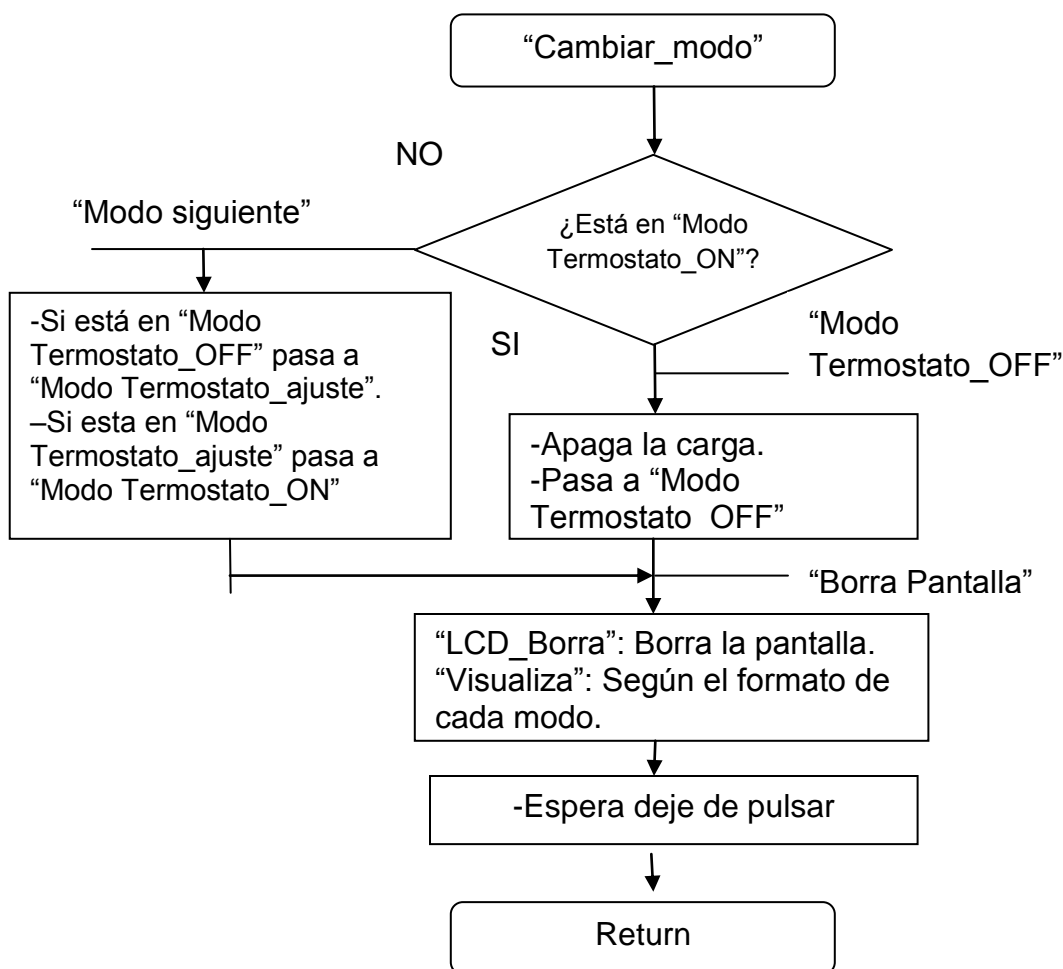


Figura 10.2 Diagrama de flujo de la subrutina Cambiar_Modo

10.3 Interrupciones: Esta subrutina lee y visualiza el termómetro cada 2 segundos aproximadamente, como se observa en el diagrama de la Figura 10.3, se ejecuta debido a la petición de interrupción del Timer 0, cada 50 milisegundo. Para conseguir una temporización de 2 s, habrá que repetir 40 veces el lazo de 50 milisegundo ($40 \times 50\text{ms} = 2000\text{ms} = 2\text{s}$).

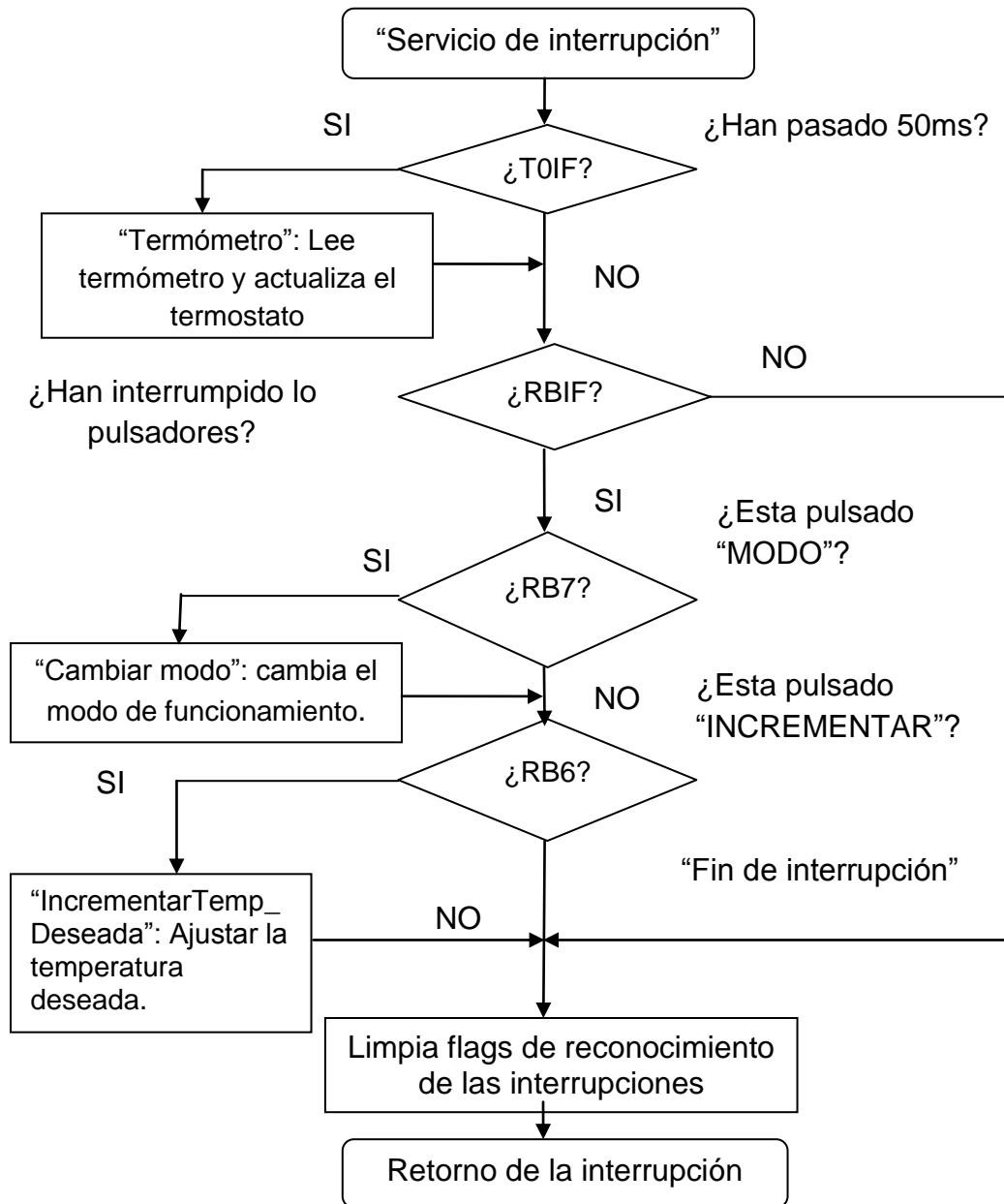


Figura 10.3 Diagrama de flujo de la subrutina de atención a las interrupciones

10.4 Funciones del termómetro DS1820, en el siguiente diagrama mostrado en la Figura 10.4, se resume su funcionamiento, conjuntamente con el servicio de interrupciones (Figura 10.3).

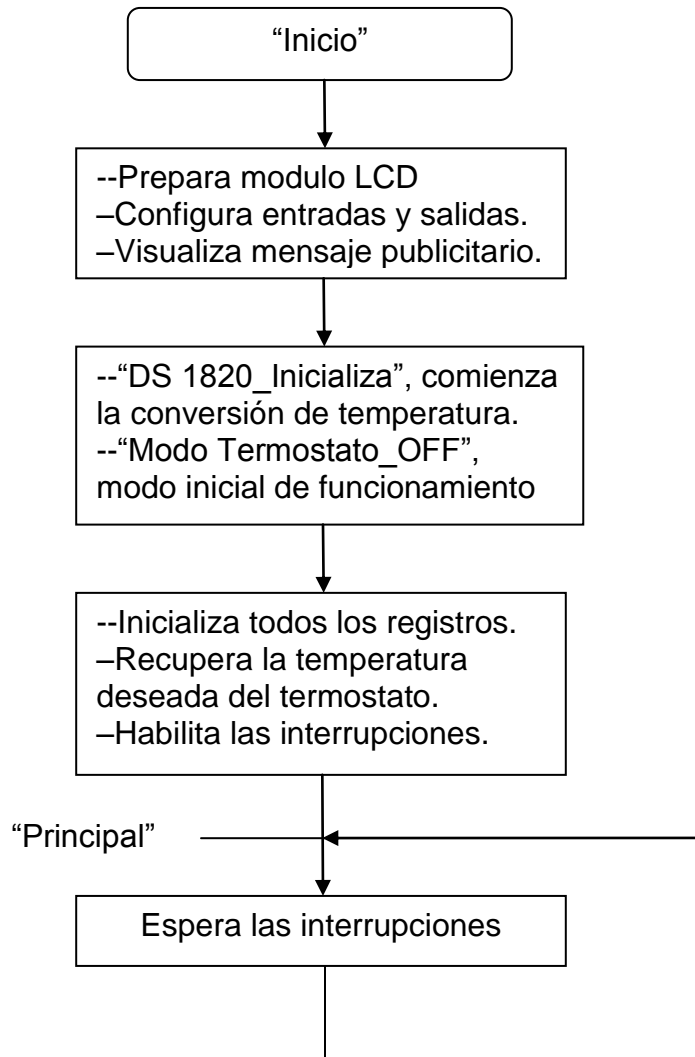


Figura 10.4 Diagrama de flujo principal del termostato digital.

El programa de manera resumida podría decirse que los controles principales son los del MODO e INCREMENTO, por el tipo de programación, que a base de

ellos podía regularse los diferentes arreglos para las pruebas, aunque el sistema no tiene un pulsador de "REDUCCION".

10.5 Resumen del programa, el autor lo resume en 4 pasos:

1. En estado de reposo funciona sólo como termómetro. Aparece la temperatura en pantalla del módulo LCD. La salida del termostato está apagada.
2. Pulsa "MODO" y se ajusta la temperatura deseada mediante el pulsador "INCREMENTAR".
3. Vuelve a pulsar "MODO", se activa el termostato. Si la temperatura medida es menor que la deseada enciende la carga, que puede ser un calefactor. Si la temperatura medida es mayor que la deseada, apaga la carga.
4. Si se vuelve a pulsar "MODO", apaga la carga y pasa a funcionar sólo como termómetro.

ANEXO 11

Vistas de las placas realizadas en Proteus (Ares), Placa No 2.

En este apartado se muestran las tres vistas de la placa número 2, tanto en el software representado en las Figuras 11.1 y 11.2, y la placa ya impresa sobre la placa fenolica mostrada en la Figura 11.3.

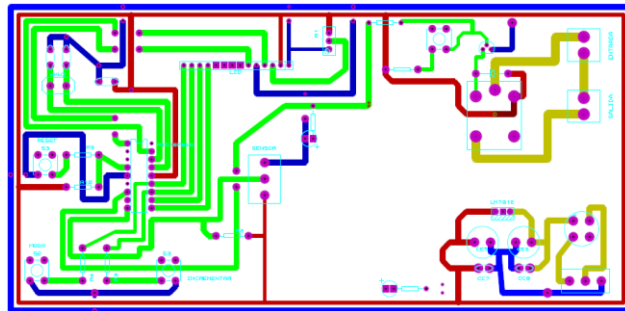


Figura 11.1 Pistas de la placa No.2 (Plano)

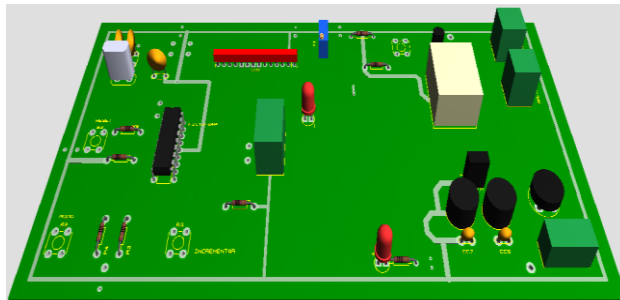


Figura 11.2 Vista en 3D, Placa No. 2

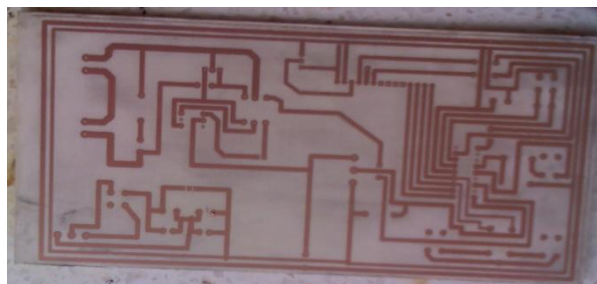


Figura 11.3 Placa impresa, Placa No 2

ANEXO 12

Pruebas realizadas en la estufa

A continuación se muestran ciclos de las diferentes pruebas que se adquirieron en la estufa cabe mencionar que las graficas solamente se registraron 3 ciclos de cada corrida realizada para ver la estabilidad generada en cada uno de ellos:

Pruebas de la estufa:

Cuadro 12.1 Análisis de datos de la Prueba 20-90 (estufa).

Ciclos	Total de datos	Amplitud de respuesta °C	Temperatura Mínima °C	Temperatura Máxima °C	Periodo de respuesta Hr.
1	2494	3.72	61.29	65.01	0.69
2	2671	4.5	61.29	65.79	0.74
3	2409	3.72	61.29	65.01	0.67
				Total horas	2.10

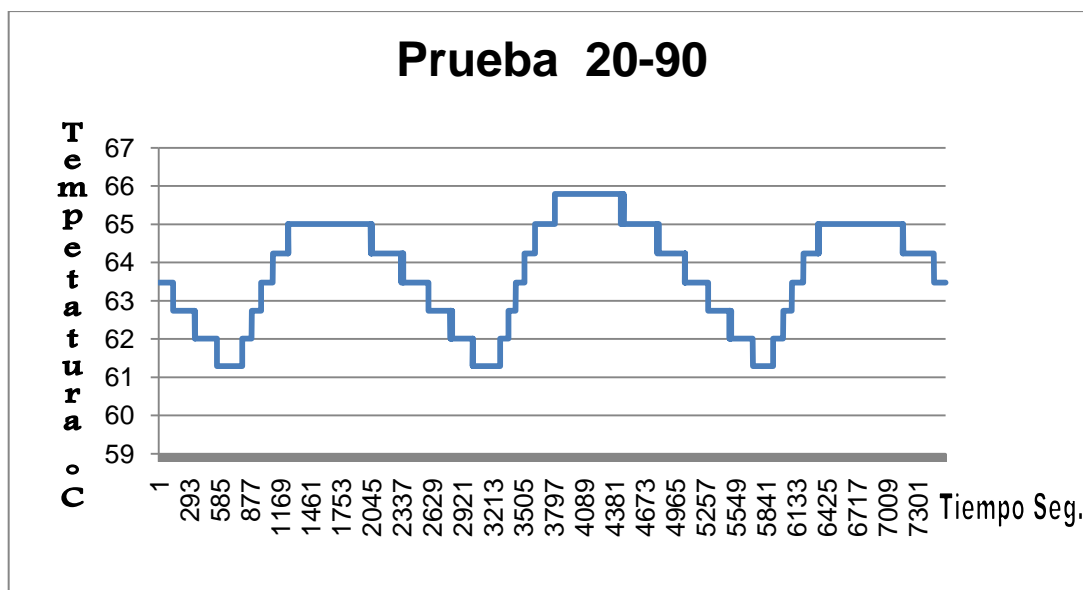


Figura 12.1 Prueba 20-90.

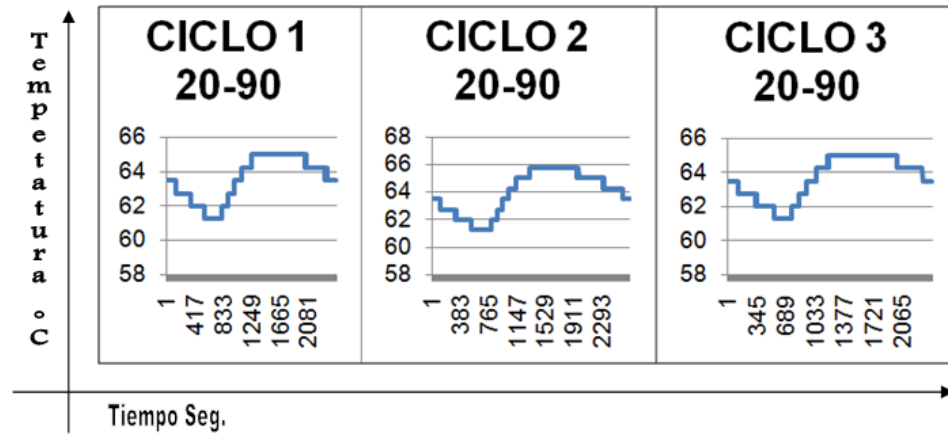


Figura 12.2 Prueba 20-90, dividida en 3 Ciclos.

Las demás pruebas se analizaron de igual manera.

ANEXO 13

Pruebas realizadas en el calentador eléctrico

Cuadro 13.1 Análisis de datos de la Prueba 35-H1.

Ciclos	Total de datos seg.	Amplitud de respuesta °C	Temperatura Mínima °C	Temperatura Máxima °C	Periodo de respuesta min.
1	269	1.27	34.43	35.7	4.48
2	229	1.26	34.01	35.27	3.81
3	238	1.69	34.01	35.7	3.96
				Total min.	12.26

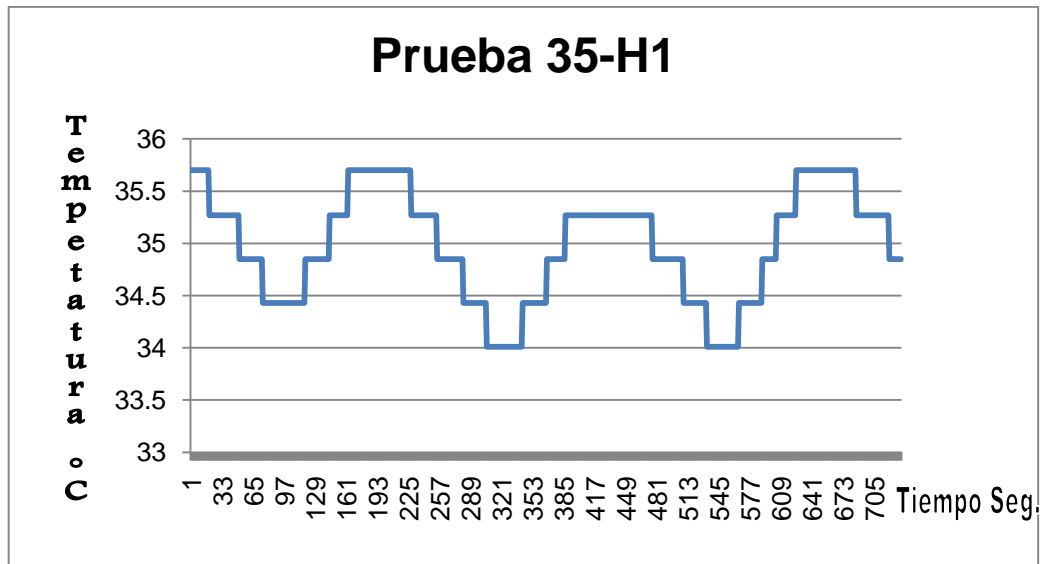


Figura 13.1 Prueba 35-H1.

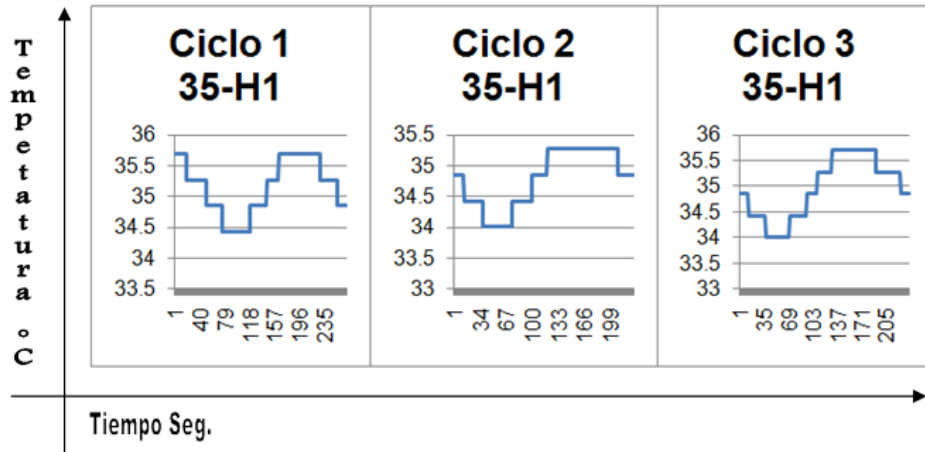


Figura 13.2 Prueba 35-H1, dividida en 3 Ciclos.

Las demás pruebas se realizaron de manera similar.

Anexo 14

Programa utilizado en los microcontroladores

```

;
; =====
; Del libro "MICROCONTROLADOR PIC16F84. DESARROLLO DE PROYECTOS"
; E. Palacios, F. Remiro y L. López.          www.pic16f84a.com
; Editorial Ra-Ma. www.ra-ma.es
; =====
;
; Programa de control para un termómetro y termostato digital. Utiliza el sensor de
; temperatura DS18B20 que transmite la información vía serie a través de un bus de una
; sola línea según un protocolo del fabricante de semiconductores Dallas Semiconductors.
;
; El ajuste de la temperatura a la que conmuta el termostato se logra mediante dos
; pulsadores: "MODO" e "INCREMENTAR", que se conectan a pines del Puerto B y cuyo
; funcionamiento se basa en interrupción por cambio en la línea del Puerto B.
;
; Se maneja de la siguiente forma:
; - En estado de reposo funciona sólo como termómetro. Aparece la temperatura en pantalla
; del módulo LCD. La salida del termostato está apagada.
; - Pulsa "MODO" y se ajusta la temperatura deseada mediante el pulsador "INCREMENTAR".
; - Vuelve a pulsar "MODO", se activa el termostato. Si la temperatura medida es menor
; que la deseada enciende la carga, que puede ser un calefactor. Si la temperatura
; medida es mayor que la deseada, apaga la carga.
; - Si se vuelve a pulsar "MODO", apaga la carga y pasa a funcionar sólo como termómetro.
;
; Así pues, en el circuito se distinguen tres modos de funcionamiento que se identifican
; mediante tres flags:
; A) Modo "Termostato_OFF", donde funciona como termómetro normal sin termostato. Se
; reconoce por el flag F_Termostato_OFF.
; B) Modo "Termostato_Ajuste", donde se ajusta la temperatura deseada cuando funcione
; como termostato. Se reconoce por el flag F_Termostato_Ajuste.
; C) Modo "Termostato_ON", donde funciona como termómetro normal con termostato. Se
; reconoce por el flag F_Termostato_ON.

```

```

;
; El programa consigue que esté activado uno solo de los flags anteriores.
;
; Al apagar el sistema debe conservar el valor de la temperatura deseada en el termostato
; para la próxima vez que se encienda.
;
; ZONA DE DATOS *****

```

```

LIST          P=16F84A
INCLUDE       <P16F84A.INC>
__CONFIG     _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

```

```

CBLOCK 0x0C

```

```

    TemperaturaDeseada

```

```

    Registro50ms                ; Guarda los incrementos cada 50 ms.

```

```

    FlagsModos                  ; Guarda los flags para establecer los

```

```

    ENDC                        ; modos de trabajo.

```

```

    ORG    0x2100                ; Corresponde a la dirección 0 de la zona
                                ; EEPROM de datos. Aquí se va a guardar el

```

```

    DE    .24                    ; la temperatura deseada. En principio 24 °C.

```

```

#define SalidaTermostato      PORTB,1        ; Carga controlada por el termostato.

```

```

#define Zumbador              PORTB,2        ; Aquí se conecta el zumbador.

```

```

#define ModoPulsador          PORTB,7        ; Los pulsadores se conectan a estos

```

```

#define IncrementarPulsador    PORTB,6        ; pines del puerto B.

```

```

#define F_Termostato_ON       FlagsModos,2   ; Flags utilizados en el ajuste de la

```

```

#define F_Termostato_Ajuste    FlagsModos,1   ; temperatura del termostato.

```

```

#define F_Termostato_OFF       FlagsModos,0

```

```

TMR0_Carga50ms EQU    d'256'-d'195'        ; Para conseguir interrupción cada 50 ms.

```

```

CARGA_2s          EQU    d'40'              ; Leerá cada 2s = 40 x 50ms = 2000ms.

```

```

; ZONA DE CÓDIGOS *****

```

```

ORG    0
goto   Inicio
ORG    4
goto   ServicioInterrupcion

```

Mensajes

```
addwf  PCL,F
```

MensajePublicitario

```
DT "IES. ISAAC PERAL", 0x00
```

MensajeTermostato_ON

```
DT "Termostato: ", 0x00
```

MensajeTermostato_Ajuste

```
DT "Temper. deseada", 0x00
```

MensajeGradoCentigrado

```
DT "°C ", 0x00 ; En pantalla LCD: "°C "
```

Inicio

```

call   LCD_Inicializa
bsf    STATUS,RP0
movlw  b'10000111' ; Prescaler de 256 para el TMR0.
movwf  OPTION_REG
bsf    ModoPulsador ; Se configuran como entrada.
bsf    IncrementarPulsador
bcf    SalidaTermostato ; Se configuran como salida.
bcf    Zumbador
bcf    STATUS,RP0
call   LCD_Linea1 ; Se sitúa al principio de la primera línea.
movlw  MensajePublicitario
call   LCD_Mensaje
call   DS18B20_Inicializa ; Comienza la conversión del termómetro y pone
call   ModoTermostato_OFF ; este modo de funcionamiento.
movlw  TMR0_Carga50ms ; Carga el TMR0 en complemento a 2.
movwf  TMR0

```

```

movlw  CARGA_2s                ; Y el registro cuyo decremento contará los 2 s.
movwf  Registro50ms
clrw                                       ; Lee la posición 0x00 de memoria EEPROM de datos
call   EEPROM_LeeDato           ; donde se guarda la temperatura deseada de la última
movwf  TemperaturaDeseada      ; vez que se ajustó.
movlw  b'10101000'            ; Activa interrupción del TMR0 (T0IE), por cambio de
movwf  INTCON                   ; líneas del Puerto B (RBIE) y la general (GIE)

```

;

; La sección "Principal" es mantenimiento. Sólo espera las interrupciones.

; No se puede poner en modo de bajo consumo porque la instrucción "sleep" detiene el Timer 0.

Principal

```

goto   Principal

```

; -----Subrutina "ServicioInterrupcion" -----

;

; Detecta qué ha producido la interrupción y ejecuta la subrutina de atención correspondiente.

ServicioInterrupcion

```

btfsc  INTCON,T0IF             ; Si es una interrupción producida por el Timer 0
call   Termometro              ; lee el termómetro y actualiza termostato.
btfss  INTCON,RBIF             ; Si es una interrupción RBI lee los pulsadores.
goto   FinInterrupcion
btfss  ModoPulsador            ; ¿Está presionado el pulsador de "AJUSTE"?
call   CambiarModo             ; Sí. Ajusta la temperatura deseada en el termostato.
btfss  IncrementarPulsador     ; ¿Pulsado "INCREMENTAR"?
call   IncrementarTempDeseada  ; Sí, pasa a incrementar la temperatura deseada.

```

FinInterrupcion

```

bcf    INTCON,RBIF             ; Limpia los flags de reconocimiento.
bcf    INTCON,T0IF
retfie

```

; -----Subrutina "Termometro" -----

;

; Esta subrutina lee y visualiza el termómetro cada 2 segundos aproximadamente. Se ejecuta
 ; debido a la petición de interrupción del Timer 0, cada 50 ms. Para conseguir una
 ; temporización de 2 s, habrá que repetir 40 veces el lazo de 50 ms (40x50ms=2000ms=2s).

;
 ; También actúa sobre la salida del termostato posicionándola adecuadamente.

Termometro

```

    movlw   TMR0_Carga50ms
    movwf   TMR0                ; Recarga el TMR0.
    decfsz  Registro50ms,F      ; Decrementa el contador.
    goto    FinInterrupcion     ; No han pasado 2 segundos, por tanto sale.
    movlw   CARGA_2s           ; Repone este contador nuevamente.
    movwf   Registro50ms
    call    DS18B20_LeerTemperatura ; Lee la temperatura.
    call    DS18B20_Inicializa   ; Comienza conversión para la siguiente lectura.
    call    Termostato          ; Actúa sobre el termostato.
;    call   Visualiza           ; Como esta subrutina se escribe a continuación
;    return                    ; se ahorra estas dos instrucciones y ahorra
;                               ; también espacio en la pila.

```

; -----Subrutina "Visualiza" -----
 ;
 ;

; Visualiza el termómetro en tres formatos posibles:

; A) Con el termostato desactivado, modo "Termostato_OFF". Por ejemplo:

```

;                               "IES. Isaac Peral" (Primera línea)
;                               " 24.5°C " (Segunda línea).

```

; Donde en la primera línea se visualiza un mensaje publicitario y en la
 ; segunda línea la temperatura medida actual.

; B) Ajuste del termostato, modo "Termostato_Ajuste". Por ejemplo:

```

;                               "Temper. deseada" (Primera línea)
;                               " 25°C " (Segunda línea).
;                               Donde en la segunda línea visualiza la temperatura que se desea ajustar.

```

; C) Con el termostato activado, modo "Termostato_ON". Por ejemplo:

```

;                               "Termostato: 25°C" (Primera línea)
;                               " 23.5°C " (Segunda línea).

```



```
; Donde en la primera línea se visualiza la temperatura que se desea
; ajustar y en la segunda línea la temperatura medida actual.
```

```
Visualiza
```

```
    btfsc    F_Termostato_OFF
    goto     VisualizaTermometro
    btfsc    F_Termostato_Ajuste
    goto     VisualizaTermostato_Ajuste
    btfsc    F_Termostato_ON
    goto     VisualizaTermostato_ON
    return
```

```
; -----"VisualizaTermostato_ON" -----
```

```
;
; Visualiza el valor de la temperatura deseada en la primera línea y el valor de la
; temperatura medida en la segunda línea.
```

```
;
```

```
VisualizaTermostato_ON
```

```
    call     LCD_Linea1
    movlw    MensajeTermostato_ON
    call     LCD_Mensaje
    call     VisualizaTemperaturaDeseada
    call     VisualizaTemperaturaMedida
    return
```

```
; -----"VisualizaTermostatoAjuste" y "VisualizaTemperaturaDeseada" -----
```

```
;
```

```
; Visualiza en la pantalla el formato propio de este modo.
```

```
;
```

```
; Entradas:      (TemperaturaDeseada) temperatura ajustada en la subrutina Incrementar.
```

```
VisualizaTermostato_Ajuste
```

```
    call     LCD_Linea1                ; Se sitúa al principio de la primera línea.
    movlw    MensajeTermostato_Ajuste ; Visualiza mensaje en la primera línea.
    call     LCD_Mensaje
```

```

movlw .6 ; Se coloca para centrar visualización en la
call LCD_PosicionLinea2 ; segunda línea.

```

VisualizaTemperaturaDeseada

```

movf TemperaturaDeseada,W
call BIN_a_BCD ; La pasa a BCD.
call LCD_Byte ; Visualiza, apagando los ceros no significativos.
movlw MensajeGradoCentigrado ; En pantalla aparece "°C ".
call LCD_Mensaje
return

```

----- "VisualizaTermometro" y ""VisualizaTemperaturaMedida" -----

;

; En la primera línea se visualiza un mensaje publicitario y en la segunda línea la

; temperatura medida

;

; Entradas:

; - (DS18B20_Temperatura), temperatura medida en valor absoluto.

; - (DS18B20_TemperaturaDecimal), parte decimal de la temperatura medida.

; - (DS18B20_Signo), registro con el signo de la temperatura. Si es igual a

; b'00000000' la temperatura es positiva. Si es b'11111111' resulta que

; la temperatura es negativa.

;

VisualizaTermometro

```

call LCD_Linea1 ; Se sitúa al principio de la primera línea.
movlw MensajePublicitario
call LCD_Mensaje

```

VisualizaTemperaturaMedida

```

movlw .5 ; Se coloca para centrar visualización en la
call LCD_PosicionLinea2 ; segunda línea.
btss DS18B20_TemperaturaSigno,7 ; ¿Temperatura negativa?
goto TemperaturaPositiva ; No, es positiva.

```

TemperaturaNegativa:

```

movlw '-' ; Visualiza el signo "-" de temperatura negativa.
call LCD_Character

```

TemperaturaPositiva

```
    movf    DS18B20_Temperatura,W
    call    BIN_a_BCD                ; La pasa a BCD.
    call    LCD_Byte                ; Visualiza apagando los ceros no significativos.
    movlw   '.'                     ; Visualiza el punto decimal.
    call    LCD_Caracter
    movf    DS18B20_TemperaturaDecimal,W ; Visualiza la parte decimal.
    call    LCD_Nibble
    movlw   MensajeGradoCentigrado  ; En pantalla LCD aparece "°C ".
    call    LCD_Mensaje
    return
```

; -----Subrutina "Termostato" -----

;

; Controla una carga en función del valor de la temperatura medida respecto de la temperatura

; deseada. Para evitar inestabilidad en la salida, tendrá un pequeño ciclo de histéresis.

; Así por ejemplo, si la temperatura deseada es 24 °C la carga se activará cuando la

; temperatura baje o sea igual a 23,5 °C y se apagará cuando la supere o sea igual a 25°C.

; Si la temperatura medida está entre esos márgenes (23,5 y 25°C), se queda en el estado

; anterior, tanto si está encendida como apagada.

;

; Para temperaturas negativas la salida se debe activar siempre.

;

; Entradas:

; - (DS18B20_Temperatura), temperatura medida en valor absoluto.

; - (TemperaturaDeseada), temperatura a partir de la cual se tomarán

; decisiones sobre la salida.

; - (DS18B20_Signo), registro con el signo de la temperatura medida. Si es cero

; la temperatura es positiva y todos sus bits son "1", es negativa.

;

; Salida: - Su funcionamiento:

; - Estando apagada, si la temperatura medida desciende por debajo de la

; temperatura deseada la salida se activará.

; - Estando encendida, si la temperatura medida supera la deseada la

```

;                               salida se apagará.
;
;   -   Si las temperaturas medidas y deseada son iguales se queda en estado
;
;                               anterior, tanto si está encendida como si está apagada.
;
;   -   Para temperaturas negativas la salida se debe activar siempre.
Termostato
    btfss    F_Termostato_ON          ; Si el termostato no está activado salta a
    goto     ApagaCarga                ; apagar la carga.
    btfsc    DS18B20_TemperaturaSigno,7 ; Con temperaturas negativas pasa a activar
    goto     EnciendeCarga              ; la carga.
    btfss    SalidaTermostato          ; Comprueba el estado actual de la salida para
    goto     SalidaEstabaApagada        ; actuar en consecuencia.
SalidaEstabaActivada                ; Pasa a comprobar si tiene que apagar la carga.
    movf     DS18B20_Temperatura,W
    subwf    TemperaturaDeseada,W      ; (W)=(TemperaturaDeseada)-(DS18B20_Temperatura).
    btfsc    STATUS,C                  ;
    ¿(TemperaturaDeseada)<(DS18B20_Temperatura)?
    goto     FinTermostato              ; Sí, por tanto, lo deja encendido y sale.
    call     Pitido                      ; Pitido cada vez que conmuta la carga.
ApagaCarga
    bcf      SalidaTermostato           ; Apaga la salida y sale.
    goto     FinTermostato
SalidaEstabaApagada                ; Pasa a comprobar si tiene que encender la carga
    movf     TemperaturaDeseada,W
    subwf    DS18B20_Temperatura,W    ; (W)=(DS18B20_Temperatura)-(TemperaturaDeseada).
    btfsc    STATUS,C                  ;
    ¿(DS18B20_Temperatura)<(TemperaturaDeseada)?
    goto     FinTermostato              ; Sí, la deja apagada y sale.
EnciendeCarga
    call     Pitido                      ; Pitido cada vez que activa la carga.
    bsf      SalidaTermostato
FinTermostato
    return

;----- Subrutinas "CambiarModo" y "ModoTermostato_OFF" -----
;

```

```

; Subrutina de atención a la interrupción producida por el pulsador "MODO" que cambia el modo
; de funcionamiento. Cada vez que pulsa pasa por los modos "Termostato_Ajuste", "Termostato_ON",
; "Termostato_OFF" y vuelta repetir.
;
; El ajuste de la temperatura deseada en el termostato se logra mediante dos pulsadores: "MODO"
; e "INCREMENTAR" conectados a pines del Puerto B.

; Al principio aparecerá sólo el termómetro y el termostato estará desactivado: modo
; "Termostato_OFF"
;
; Para comprender el funcionamiento de esta subrutina, hay que saber que el registro FlagsModos
; contiene 3 flags que permiten diferenciar cada uno de los modos de funcionamiento:
; A)      Modo "Termostato_OFF", donde funciona como termómetro normal sin termostato. Se
;         reconoce por el flag F_Termostato_OFF, que es el bit 0 del registro FlagsModos.
; B)      Modo "Termostato_Ajuste", donde se ajusta la temperatura deseada cuando funcione
;         como termostato. Se reconoce por el flag F_Termostato_Ajuste, que es el bit 1 del
;         registro FlagsModos.
; C)      Modo "Termostato_ON", donde funciona como termómetro normal y, además, como termostato.
;         Se reconoce por el flag F_Termostato_ON, que es el bit 2 del registro FlagsModos.
;
; Así pues, el contenido del registro (FlagsModos) identifica los siguientes modos de
; funcionamiento:
; - (FlagsModos)=b'00000001'. Está en el modo "Termostato_OFF".
; - (FlagsModos)=b'00000010'. Está en el modo "Termostato_Ajuste".
; - (FlagsModos)=b'00000100'. Está en el modo "Termostato_ON".

; Pueden darse dos casos:
; -      Que pulse "AJUSTE" estando en el modo más alto, "Termostato_ON",
;        (FlagsModos)=b'00000100'. En este caso debe pasar al modo inicial
;        "Termostato_OFF" poniendo (FlagsModos)=b'00000001'.
; -      Que pulse "AJUSTE" estando ya en cualquiera de los otros dos modos, en cuyo caso debe
;        pasar al siguiente modo. Esto lo hace mediante un desplazamiento a izquierdas. Así, por
;        ejemplo, si antes estaba en modo "Termostato_OFF", (FlagsModos)=b'00000001', pasará a
;        (FlagsModos)=b'00000010' que identifica al modo "Termostato_Ajuste".

```

```

;
CambiarModo
    call    Retardo_20ms           ; Espera a que se estabilicen niveles de tensión.
    btfsc  ModoPulsador           ; Si es un rebote, sale fuera.
    goto   FinCambiarModo
    call   PitidoCorto            ; Cada vez que pulsa se oye un pitido.
    btfss  F_Termostato_ON       ; Detecta si está en el último modo.
    goto   ModoSiguiente         ; Si no, pasa al modo siguiente.

```

```

ModoTermostato_OFF
    call   Pitido                 ; Pitido cada vez que conmuta la carga.
    bcf    SalidaTermostato       ; Apaga la carga.
    movlw  b'00000001'           ; Actualiza el registro FlagsModos pasando al
    movwf  FlagsModos            ; modo inicial "Termostato_OFF".
    goto   BorraPantalla

```

```

ModoSiguiente                               ; Desplaza un "1" a la izquierda del registro
    bcf    STATUS,C              ; FlagsModos para ajustar secuencialmente
    rlf    FlagsModos,F          ; cada uno de los modos de funcionamiento.

```

```

BorraPantalla
    call   LCD_Borra             ; Borra la pantalla anterior.

```

```

FinCambiarModo
    call   Visualiza
    btfss  ModoPulsador         ; Ahora espera a que deje de pulsar.
    goto   FinCambiarModo
    return

```

```

; ----- Subrutina "IncrementarTempDeseada" -----

```

```

;
; Subrutina de atención a la interrupción por cambio de la línea RB6 a la cual se ha conectado
; el pulsador "INCREMENTAR". Estando en el modo "Termostato_Ajustar" incrementa el valor de
; la temperatura deseada entre unos valores máximo y mínimo.
;
; Al final debe guardar el valor de la temperatura deseada en memoria EEPROM de datos para
; preservar su valor en caso que desaparezca la alimentación.
;

```

TemperaturaMinima EQU .20

TemperaturaMaxima EQU .36

IncrementarTempDeseada

```
call    Retardo_20ms           ; Espera a que se estabilicen niveles de tensión.
btfsc  IncrementarPulsador     ; Si es un rebote sale fuera.
goto   FinIncrementar
btfss  F_Termostato_Ajuste     ; Si no está en modo "Termostato_Ajuste" sale
goto   FinIncrementar         ; fuera.
call   PitidoCorto              ; Pitido cada vez que pulsa.
incf   TemperaturaDeseada,F     ; Incrementa el valor de la temperatura deseada.
movlw  TemperaturaMaxima       ; ¿Ha llegado a la temperatura máxima de ajuste?.
subwf  TemperaturaDeseada,W     ; (W) = (TemperaturaDeseada) - TemperaturaMaxima.
btfss  STATUS,C                ;
¿(TemperaturaDeseada)>=TemperaturaMaxima?
goto   VisualizaIncremento     ; No, pasa a visualizarlo.
movlw  TemperaturaMinima       ; Sí, entonces inicializa el registro.
movwf  TemperaturaDeseada
```

VisualizaIncremento

```
call   Visualiza               ; Visualiza mientras espera a que deje
call   Retardo_200ms           ; de pulsar.
btfss  IncrementarPulsador     ; Mientras permanezca pulsado,
goto   IncrementarTempDeseada  ; incrementa el dígito.
clrw
; Salva el valor de la temperatura deseada en la
movwf  EEADR                   ; posición 00h de la EEPROM de datos. Se conserva
movf   TemperaturaDeseada,W    ; aunque se apague la alimentación.
call   EEPROM_EscribeDato
```

FinIncrementar

```
return
```

; -----Subrutina de pitidos -----

;

PitidoLargo

```
bsf    Zumbador
```

```
call Retardo_500ms
```

```
Pitido
```

```
bsf Zumbador
```

```
call Retardo_200ms
```

```
PitidoCorto
```

```
bsf Zumbador
```

```
call Retardo_20ms
```

```
bcf Zumbador
```

```
return
```

```
;
```

```
INCLUDE <DS18B20.INC> ; Subrutinas de control del termómetro digital.
```

```
INCLUDE <BUS_1LIN.INC> ; Subrutinas de control del bus de 1 línea.
```

```
INCLUDE <RETARDOS.INC>
```

```
INCLUDE <BIN_BCD.INC>
```

```
INCLUDE <LCD_4BIT.INC>
```

```
INCLUDE <LCD_MENS.INC>
```

```
INCLUDE <EEPROM.INC>
```

```
END
```