



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

**“SISTEMA DE MONITOREO DE TEMPERATURA PARA
HORNO DE GAS”**

TESIS

**PARA OBTENER EL TÍTULO DE
INGENIERO EN ELECTRÓNICA**

PRESENTA

OSWALDO ARIAS MARTÍNEZ

DIRECTOR DE TESIS

M. C. JACOB JAVIER VÁSQUEZ SANJUAN

HUAJUAPAN DE LEÓN, OAXACA. MAYO DE 2006

Dedicatoria

Bien dicen que en cualquier proyecto intervienen más de una persona, no necesariamente de forma directa. Esa contribución es la fuerza que nos transmiten nuestros seres queridos ya sea para iniciar, continuar o terminar la meta. So pena de parecer una mención convencional y yerta, Esther y Manuel, mis padres, son los verdaderos autores de este trabajo al darme educación, casa, vestido y por sobre todo, amor y apoyo.

Injusto sería de mi parte dejar fuera a Jacob, mi asesor, quien tuvo la paciencia para guiarme en esta empresa.

Índice

Dedicatoria	iii
Índice	v
Lista de tablas	vii
Lista de figuras	ix
Resumen	xiii
Abstract	xv
Capítulo 1 - INTRODUCCION	17
1.1 Descripción general del problema	17
1.2 Antecedentes.....	18
1.3 Justificación del proyecto.....	20
1.4 Objetivos del proyecto.....	21
Capítulo 2 - EL PROCESO DE COCCIÓN CERÁMICA	23
2.1 Elementos de la quema	23
2.2 Algoritmo de la quema	28
2.3 Revisión de sistemas relacionados.....	33
Capítulo 3 - DISEÑO DEL SISTEMA	43
3.1 Consideraciones generales del diseño	43
3.2 Esquema primario del sistema.....	44
3.3 Propuesta para la etapa de control y monitoreo	45
3.4 Propuesta para la etapa de sensores y actuadores	49
3.5 Propuesta para respaldo de energía en el sistema	51
Capítulo 4 - CONSTRUCCIÓN DEL SISTEMA	53
4.1 Construcción de la etapa de control y monitoreo.....	53
4.2 Diseño final de la etapa de sensores y actuadores	71
4.3 Respaldo de energía en el sistema	77
Capítulo 5 - RESULTADOS	79
5.1 Interfaz del sistema (programa en computadora)	79
5.2 Interfaz del sistema (programa en microcontrolador)	86
5.3 Precisión en la medición de temperatura.....	87
Capítulo 6 - CONCLUSIONES	89
6.1 Conclusiones generales.....	89
6.2 Limitaciones y expansión del sistema.....	90
Apéndice A - TERMOPAR Y ELECTROVÁLVULA	93
A.1 Breve descripción del termopar.....	93

A.2 Breve descripción de la electroválvula.....	99
Apéndice B - CÓDIGO FUENTE	101
B.1 Pseudocódigo	101
B.2 Código fuente	106
Apéndice C - MANUAL DEL SISTEMA.....	125
C.1 Descripción del sistema	125
C.2 Instalación del software para computadora	128
C.3 Lógica del proceso de quema	130
C.4. Modos de operación.....	131
C.5 Especificaciones técnicas.....	141
C.6 Solución de problemas.....	143
BIBLIOGRAFÍA	145

Lista de tablas

Tabla 2.1. Resumen comparativo del proyecto con los sistemas revisados.	42
Tabla 4.1. Comparación entre varios microcontroladores.....	55
Tabla 4.2. Comparación entre códigos fuente: lenguaje C – ensamblador.	62
Tabla A.1. Coeficientes Seebeck	94
Tabla A.2. Características de los tipos de termopares	96

Lista de figuras

Figura 1.1. Taller de Cerámica de la Universidad Tecnológica de la Mixteca	17
Figura 1.2. Modelos comerciales de medición de temperatura marca TEMPCO. (a) Serie 553M, (b) Serie 555M.....	19
Figura 2.1. (a) Sancocho, o mezcla de arcilla con agua. (b) Pieza moldeada puesta a secar. (c) Pieza cocida. (d) Esmalte en polvo. (e) Pieza esmaltada.....	24
Figura 2.2. Horno de tiro invertido a base de gas, con puerta cerrada y carro en el interior.....	25
Figura 2.3. Interior del horno, riel y carro.	26
Figura 2.4. Quemadores.	26
Figura 2.5. Válvulas de regulación con sus indicadores de presión respectivos.	27
Figura 2.6. Indicador de temperatura basado en termopar.	27
Figura 2.7. Suministro del gas combustible. (a) Tanque estacionario, (b) Llave de suministro principal.	28
Figura 2.8. Carga del carro.	28
Figura 2.9. Ejemplos de conos pirométricos.	29
Figura 2.10. Diagrama de flujo simplificado del procedimiento de cocción cerámica o quema.	31
Figura 2.11. Registrador de datos Spectrum 1700	34
Figura 2.12. Software Spectrum Data Logger.....	35
Figura 2.13. Controlador AutoFire StandAlone 3000.	36
Figura 2.14. Software ContolMaster DX™.....	37
Figura 2.15. Registrador de datos Tpaq21.....	38
Figura 2.16. Software Kiln Insight. (a) Análisis, (b) Telemetría, (c) Registro de procesos.....	38
Figura 2.17. Diagrama del proyecto.	39
Figura 2.18. Controlador universal empleado en el proyecto.....	40
Figura 2.19. Detector radial de temperatura.	41
Figura 2.20. Actuador.....	41
Figura 3.1. Diagrama a bloques primario del sistema.	44
Figura 3.2. Etapa de control y monitoreo.	46
Figura 3.3. Especificación del control por microcontrolador.	47
Figura 3.4. Especificación del control por computadora.....	48
Figura 3.5. Ubicación sugerida del termopar.	50
Figura 3.6. Propuesta para etapa de sensores y actuadores.	51
Figura 3.7. Respaldo de energía en el sistema.....	52
Figura 4.1. Diagrama de conexiones del control por MC.	59
Figura 4.2. Organización de las memorias de programa y de datos.	63
Figura 4.3. Organización en la memoria en el sistema.....	64
Figura 4.4. Formato general en la memoria de datos del sistema.	64

Figura 4.5. Formato de la zona de parámetros en la memoria del sistema.....	65
Figura 4.6. Protocolo de comunicación entre PC y MC.....	71
Figura 4.7. Forma general de un termómetro basado en termopar.....	72
Figura 4.8. Interferencia eléctrica en la conexión del termopar con el circuito. (a) Representación a bloques, (b) Representación del flujo de control.....	73
Figura 4.9 Supresión de la interferencia mediante filtro pasabajas.....	73
Figura 4.10. Compensación de la unión fría del termopar.....	74
Figura 4.11. Compensación de unión fría realizada por el LT1025.....	74
Figura 4.12. Amplificación.....	75
Figura 4.13. Termómetro basado en termopar tipo K.....	76
Figura 4.14. Apariencia del termopar empleado. (a) Completo, (b) Acercamiento a la unión fría, (c) Cable de extensión para termopar tipo K.....	76
Figura 4.15. Circuito de control de electroválvula.....	77
Figura 4.16. Electroválvula S201SF02N5EG5, dos vías, normalmente cerrada.....	77
Figura 4.17. No-Break mod. Internet 3500 de Tripp-Line.....	78
Figura 5.1. Interfaz del sistema (en PC).....	80
Figura 5.2. Menús “Opción” y “Datos”.....	80
Figura 5.3. Primer aviso del sistema al operador en una quema.....	81
Figura 5.4. Aviso de error en inicialización del puerto serial.....	81
Figura 5.5. Pregunta al usuario sobre descarga de datos.....	82
Figura 5.6. (a) Indicador “VALVULA” indicando estado abierto, (b) Botón “abortar” habilitado... 82	82
Figura 5.7. Ejemplos de avisos al usuario. (a) Cuando se alcanza la inversión del cuarzo, (b) Cuando se debe incrementar la presión de L y C.....	83
Figura 5.8. Ejemplo de curva de temperatura.....	83
Figura 5.9. Guardado de datos en archivo.....	84
Figura 5.10. Aviso final después de una operación exitosa.....	84
Figura 5.11. Aviso de error durante descarga de datos.....	85
Figura 5.12. Guardado de datos recuperados.....	85
Figura 5.13. Elección del archivo a graficar e imprimir.....	86
Figura 5.14. Gráfica comparativa de la medición de temperatura entre el medidor del proyecto y el termómetro digital presente en el horno.....	88
Figura 6.1. Propuesta de colocación de termopares extra en el horno.....	91
Figura A.1. Forma general del termopar.....	93
Figura A.2. Códigos de color por país.....	97
Figura A.3. Montaje del termopar. (a) Sellado y aislado de la coraza o termopozo, (b) Sellado y aterrizado a la coraza, (c) Expuesto, (d) Expuesto con respuesta rápida.....	98
Figura A.4. Compensación de unión fría por hardware.....	98
Figura A.5. Compensación de unión fría por software.....	99
Figura A.6. Tipos de electroválvulas. (a) Dos vías, (b) Tres vías, (c) Cuatro vías.....	100
Figura C.1. Estructura general del sistema Horno Monitor v1.0.....	125
Figura C.2. Gabinete del sistema Horno Monitor v1.0.....	126
Figura C.3. Termopar tipo K.....	127
Figura C.4. Electroválvula.....	127
Figura C.5. Respaldo de energía.....	128
Figura C.6. Instalador del sistema.....	129
Figura C.7. Instalación exitosa del sistema.....	129
Figura C.8. Acceso directo del programa.....	130
Figura C.9. Interfaz del control por computadora.....	133

Figura C.10. Menú “Opción” y su contenido	133
Figura C.11. Menú “Datos y su contenido”	135
Figura C.12. Pantalla y teclado del gabinete del sistema	136
Figura C.13. Aviso en la pantalla sobre la espera de inicio de sesión	137
Figura C.14. Aviso en la pantalla sobre el control del proceso por computadora	137
Figura C.15. Alerta al usuario sobre datos en el sistema	137
Figura C.16. Confirmación de decisión	137
Figura C.17. Avisos para captura de los distintos parámetros. (a) Temperatura objetivo, (b) Hora de inicio, (c) Fecha, (d) Tasa de muestreo.....	138
Figura C.18. Despliegue de información en la pantalla durante la quema	139
Figura C.19. Ejemplo de interpretación del contenido de la pantalla del sistema	140
Figura C.20. Aviso de proceso de quema terminado.....	140
Figura C.21. Aviso de proceso de quema abortado	140
Figura C.22. Avisos relativos a la descarga de datos. (a) Mientras dura la descarga, (b) Cuando la descarga ha sido completada con éxito, (c) Cuando hubo problemas de comunicación durante la descarga	141
Figura C.23. Interior del gabinete	142
Figura C.24. Tarjeta principal	142
Figura C.25. Driver de la electroválvula	143
Figura C.26. Sensor de temperatura	143

Resumen

La Electrónica es una de las ciencias con mayor presencia en la actualidad al ofrecer herramientas de trabajo para prácticamente todos los campos de aplicaciones. La actividad orientada a la cerámica es un claro ejemplo de esta situación, pues auxilia en muchos aspectos, desde indicadores de temperatura hasta complejos sistemas de automatización de procesos.

La Universidad Tecnológica de la Mixteca cuenta con un Taller de Cerámica cuyo primer propósito es el académico. La parte final de sus procesos es la cocción – llama comúnmente *quema* – en un horno de gas de las piezas cerámicas que elaboran tanto alumnos como profesores y es ahí donde el presente trabajo cubre la necesidad de monitorear la temperatura durante tal proceso y el registro de la curva de temperatura descrito por él. Obviamente este no es el único requerimiento del taller, pero es un buen comienzo para instrumentarlo.

Electrónicamente hablando, hay varios modos de cubrir esta necesidad, siendo elegido el *microcontrol* con base en su flexibilidad en implementación para cualquier proceso. A partir de una computadora personal y del popular 8032 de Intel se diseñó y desarrolló un *Sistema de Monitoreo de Temperatura para Horno de Gas*. La naturaleza del problema dictó la arquitectura a usar, una del tipo *maestro-esclavo* con la cualidad de que el esclavo es capaz de asumir el control del monitoreo en caso de que el maestro falle. A esta cualidad se le llama *control redundante del monitoreo*. El maestro es la computadora personal, que a través de una interfaz hecha en LabView ordena a su esclavo, el microcontrolador, obtener la medición de temperatura interna del horno y así saber en qué parte del proceso de quema se encuentra.

Si se ha de medir y mostrar la temperatura (interna) del horno, la elección del transductor de temperatura fue fundamentada en su rango de medición. El termopar tipo K es el ideal, sólo fue necesario lidiar con su respuesta no lineal por medio de un compensador de unión fría que ofrece la reducción del tiempo de diseño del sensor de temperatura basado en termopar (termómetro).

Así, este sistema es una opción económica y personalizada para el Taller de Cerámica, que busca ser un punto de inicio para la optimización de sus procesos.

Abstract

Electronics is one of the most resorted sciences today since it gives powerful tools to practically any application field. Ceramics is an example of that because it is aided in many ways, since temperature indicators (thermometers) to complex automated-process systems.

The Universidad Tecnológica de la Mixteca has a Ceramics workshop whose main purpose is academic. The ultimate stage of its process is the pottery firing (made by students and professors) in a gas-based kiln. It's there where this project fulfills a workshop's need: temperature monitoring along the burn and also it registers the temperature curve depicted. Obviously this is not the only requirement of the workshop, but is a beginning for its (electronic) instrumentation.

Electronic-wisely, there are several ways for fulfilling this aspect and the chosen one is the micro-control because of its flexibility to be implemented in any process. Beginning with a personal computer and an MCS51 microcontroller, it was designed and developed a *Temperature Monitoring System for a Gas-based Kiln*. The properties of this problem lead the design of the architecture: master-slave, with the possibility that the slave can take the control of the monitoring in case of a master failure. This feature is named *redundant-control monitoring*. The computer acts as the master; it has a graphic interface developed in LabView which communicates with the slave, the microcontroller, to get the current temperature in the kiln and so, it knows exactly which part of the firing is in.

About the (internal) kiln temperature measuring, a K-type thermocouple was chosen because it covers the temperature range (up to 1250 °C). Its disadvantage is its non-linear behavior, but it was solved by using a cold-junction compensator which allows reducing the developing time of the thermocouple-based temperature sensor (thermometer).

So then, this system is an economic and personalized option for the Ceramics Workshop in the aim of being the beginning for the optimization of its processes.

Capítulo 1 - INTRODUCCION

1.1 Descripción general del problema

La Universidad Tecnológica de la Mixteca (UTM) sustenta varios talleres con propósitos académicos, donde profesores y estudiantes desempeñan alguna actividad específica con el fin de reforzar sus conocimientos a través de la práctica. Uno de estos talleres es el de Cerámica, ahí los alumnos de la carrera de Ingeniería en Diseño, los profesores encargados de la materia y ocasionalmente artesanos ajenos a la Universidad elaboran piezas cerámicas de distinta índole, tamaño y fin. Para los artesanos ceramistas, los puntos clave de una pieza son tres: materia prima, creatividad y la cocción. Este último punto, la cocción (o *quema*, como se le llama comúnmente), contempla al objeto principal del presente proyecto de tesis: un horno de gas de alta temperatura. Se califica de principal dado que toda pieza cerámica requiere de cocción y por ende, de usar el horno para cocerla. Hasta el momento el modo de operación del horno ha sido totalmente manual y basado en la experiencia del operador o mejor dicho, del responsable del taller. El operador juega un papel crucial en cada quema, pues debe estar atento al proceso la mayor parte del tiempo, cuidando que la temperatura interior del horno y la presión del gas se mantengan en un rango adecuado según la etapa en la que se encuentre la quema, tomando acciones específicas cuando algún evento se presente y llevando un registro de todos los datos que se vayan generando. Obviamente, esto lo hace aquel que tiene experiencia en el manejo del horno. De manera preliminar y superficial queda establecido que el objetivo del proyecto es asistir al operador en el proceso de cocción o quema.



Figura 1.1. Taller de Cerámica de la Universidad Tecnológica de la Mixteca

1.2 Antecedentes

1.2.1 Breve descripción de la cerámica

La cerámica es una de las primeras actividades del hombre [1] desde que se volvió sedentario, alrededor de 15 mil años a.C. Tan importante es esta actividad que los arqueólogos dependen de los tipos de cerámica para distinguir entre las culturas y establecer su orden cronológico.

Obviamente la cerámica ha evolucionado con el tiempo hasta llegar a establecerse como actividad industrial, con procesos mucho más complejos y eficientes que los primeros. En nuestros tiempos no sólo sigue siendo una actividad industrial sino que también es un campo de investigación [URL 14], especialmente en las propiedades eléctricas y moleculares de los materiales.

La cerámica es posible gracias a la propiedad física de la plasticidad (cambio de forma sin romperse al ser aplicada alguna fuerza externa y retención de dicha forma una vez retirada la fuerza), siendo la arcilla la que tiene esta propiedad. Con ello se hace fácilmente maleable una vez que es combinada con agua y pierde su maleabilidad cuando se seca y se cuece.

La realización de piezas comprende varios pasos, siendo la cocción el punto final. El método de cocción varía de acuerdo al tipo de materiales y las condiciones del medio ambiente. En el capítulo 2 se trata este tema personalizado para el taller de la universidad.

1.2.2 El entorno de la industria cerámica

Para cualquier industria, un objetivo es mejorar siempre los procesos y aumentar la productividad [1] a través de sistemas automáticos o semiautomáticos. Si ahora agregamos el factor de competencia, entonces no basta con mejorar procesos y aumentar productividad, sino también hay que incrementar rapidez en la producción y calidad del producto. Esos sistemas de control deben considerar varios factores tales como la temperatura ambiente, la humedad y presión del medio, entre varios. Con todo esto, el monitoreo de la temperatura interna del horno (u hornos) durante la cocción de las piezas de cerámica se vuelve necesario y a la postre lleva al control regulable de temperatura.

Típicamente el precio de sistemas que automatizan los procesos de producción a gran escala asciende a varios miles de dólares americanos, provocando que puedan ser adquiridos nada más por las grandes compañías y que las modestas se remitan a ejecutar sus procesos de la manera más sencilla: a través de la supervisión de un operador u operadores expertos en la materia. Entre los principales elementos de automatización se encuentran:

- Mezcladoras de materiales.
- Extractores de impurezas.
- Secadores de piezas
- Bandas transportadoras de piezas.
- Hornos de alta temperatura.
- Medidores y controles (de acuerdo al tipo del horno) de temperatura.

Del último punto se deriva el interés del presente proyecto: el monitoreo de la temperatura durante la cocción. El término “monitoreo” tiene varias acepciones similares entre sí que varían

de acuerdo al campo en que se aplica, pero todas las relacionadas con ingeniería coinciden en la detección de alguna señal o parámetro y su medición cualitativa o cuantitativa [URL 15].

1.2.3 Sistemas de medición de temperatura existentes

Actualmente existen en el mercado sistemas electrónicos para medición [2] y control de temperatura. En un principio fueron todos analógicos y tenían sus limitantes en cuanto a precisión (los indicadores eran de tipo aguja y constantemente debían calibrarse). Tiempo después, con el auge de la electrónica digital, estos sistemas mejoraron. Todavía se producen sistemas analógicos, aunque en mucha menor cantidad y variedad que los digitales.

Es pertinente hablar un poco más de este aspecto. La electrónica digital ha aportado dos dispositivos de gran versatilidad: los microcontroladores y los microprocesadores [3] (y con estos últimos, las computadoras electrónicas). En efecto, esa versatilidad abarca los sistemas de medición de temperatura, de hecho, la mayoría de ellos está basada en alguno de estos dos dispositivos. Hay un sinnúmero de modelos de microcontroladores y microprocesadores, cada uno con características particulares que en determinado contexto es preferible a los demás.

Los modelos que se van a presentar como ejemplo corresponden a la compañía TEMPCO México S.A. de C.V. [URL 1]

- Serie 553M. Control basado en microcontrolador, su método es del tipo ON-OFF, permite modificar los límites de temperatura que permiten la protección del proceso. Su factor de error en medición es de 1% sobre el rango total. Contiene una doble pantalla donde se indican la temperatura objetivo y la temperatura actual de proceso. Permite seleccionar la recepción de señales para termopares J, K, R y T. Se alimenta con 110 VCA o 220 VCA, su consumo de poder es de 2.4 VA.
- Serie 555M. Control analógico, su método es del tipo ON-OFF. Los rangos están preestablecidos. Admite termopares tipo J o K. Su factor de error en medición es de 1% sobre el rango total. Utiliza leds para indicar el estado de operación. Cuenta con protección en caso de corto en la caja y contra ruptura del termopar. Se alimenta con 110 VCA o 220 VCA, su consumo de poder de 1.8 VA.



(a)



(b)

Figura 1.2. Modelos comerciales de medición de temperatura marca TEMPCO. (a) Serie 553M, (b) serie 555M.

Para los dos ejemplos, el método ON-OFF se refiere a una simple señal electrónica que sirve para indicar el estado del proceso, es decir, por sí sola no hace el encendido o apagado del horno, es necesario añadir los dispositivos adecuados

1.3 Justificación del proyecto

La pregunta que hasta el momento podría surgir es: ¿realmente es justificable este proyecto? Si ya existen en el mercado sistemas destinados al monitoreo de temperatura, no habría la necesidad de tomar esta tarea. Además, en caso de requerir una aguda funcionalidad, sería muy difícil reproducirla por una sola persona. Para dar una idea clara de la conveniencia del proyecto se expondrá primero la situación del taller.

1.3.1 Situación del taller

En un principio se mencionó que en el Taller de Cerámica los alumnos y profesores elaboran sus prácticas, es decir, tiene un enfoque académico, haciendo que muchas de las necesidades de la industria cerámica no sean aplicables a la situación. Específicamente:

- La producción no es a gran escala ni tampoco es diaria, de hecho, típicamente el número de piezas elaboradas por quema es inferior a 50 y un máximo de diez quemadas es realizado al año.
- La máxima temperatura que soporta el horno es de 1250 °C, aunque nunca han trabajado a tal temperatura.
- La máxima presión permitida en la tubería del horno para el combustible, gas en este caso, es de 31 psi o 2.2 Kg/cm², aunque el promedio de presión por quema es de 7.8 psi o 0.6 Kg/cm².
- Únicamente se realizan dos tipos de quema: la de sancocho de barro y la de esmalte. Estos materiales no requieren un cuidado tan riguroso como el de algunos empleados en la industria (como la porcelana). En el capítulo 2 se explican estos dos términos, así como varios más involucrados con el proceso.

El encargado del taller es quien tiene la habilidad para manejar el horno. Si un percance se presenta, él es el único que sabe qué hacer, no es aconsejable que un estudiante o simplemente alguien con poca experiencia en la materia lo haga, ya que pone en peligro su integridad física al trabajar con esas temperaturas.

El proceso de la quema, a groso modo, indica que el trabajo del operador empieza por acomodar todas las piezas en un pequeño carro que es conducido a través de un riel hasta el interior del horno, no sin antes encender los quemadores que se encuentran en la parte inferior. Una vez que se ha hecho esto, las formas que tiene el operador para saber lo que sucede en el interior del horno son por medio de un indicador digital de temperatura y por medio de dispositivos conocidos como pirómetros, los cuales son vistos a través de unas mirillas ubicadas en la puerta del horno. Así, el proceso de cocción se va desarrollando de acuerdo a un método establecido para el horno del taller (será descrito en el capítulo 2) y que termina cuando se alcanza una temperatura específica. Claro está que si sucede algo que amerite el aborto del proceso de cocción, el operador lo hará cortando el suministro del gas.

Ha quedado descrita la situación actual, ahora sigue conocer lo que necesita el taller para mejorar el proceso de cocción.

1.3.2 Necesidades del taller

El Taller de Cerámica tiene básicamente cinco necesidades en el proceso de cocción:

1. Identificación de temperatura. Se refiere a tener un dispositivo que indique la temperatura neta del interior del horno.
2. Graficación de la quema. Este punto es muy importante, pues se refiere a llevar un registro de la quema. Su utilidad radica en que la curva que genere la gráfica temperatura–tiempo denota el comportamiento de la quema, los tiempos en que se alcanzan los puntos críticos de la quema. También es deseable tener un registro de la presión, mas no es imprescindible.
3. Mayor seguridad en el proceso. El hecho de trabajar con gas siempre implica un riesgo. En caso de presentarse algún incidente que amerite el corte del suministro del gas, éste debe realizarse a la mayor brevedad posible.
4. Homogenización del calor. Para que exista una buena cocción de todas las piezas que se hayan introducido en el horno, la distribución del calor debe ser equitativa en todo su interior.
5. Sistemas mecánicos que aminoren el esfuerzo humano. Se refieren a dos puntos: la manipulación del carro y el encendido automático de los quemadores. La razón estriba en que el carro es muy pesado, se requiere de al menos dos personas para meterlo y sacarlo del horno, sin considerar que al término de la quema éste se encuentra muy caliente. Respecto a los quemadores, porque estos se encienden de forma manual y existe el riesgo de sufrir quemaduras.

De manera general, se aprecia que lo que se desea es un sistema que prácticamente automatice la quema. Los tres primeros puntos pueden ser cubiertos por este proyecto ya que la identificación y la graficación pueden ser realizadas a la par por el dispositivo elegido y el aumento en la seguridad del proceso se remite a la inclusión de un dispositivo electromecánico conocido como electroválvula, que será discutido posteriormente. Pero en lo que respecta a la homogenización de la temperatura se requieren de dispositivos electromecánicos llamados servoválvulas, que no son económicamente accesibles por el autor del proyecto. Finalmente, en los mecanismos para aminorar el esfuerzo humano, el problema no sería la lógica del subproceso, sino la fabricación del mecanismo, ahí se requiere de alguien experimentado en mecánica y en consecuencia está fuera del alcance de este proyecto.

1.4 Objetivos del proyecto

Una vez expuestos todos los puntos relacionados al proyecto, es posible establecer los objetivos del mismo, estos son:

- Auxiliar al operador u operadores durante la quema. El sistema le debe avisar cuándo se ha llegado a una etapa crítica del proceso (incluyendo su finalización) o cuándo es necesario ajustar la presión de gas, etc. Incluso, al recrear el algoritmo del proceso de cocción podría servir para que gente con poca experiencia en el manejo del horno tenga una guía en el proceso, adquiriendo así la característica de estar personalizado al Taller de Cerámica.

- Realizar automáticamente la elaboración del registro y de la graficación en tiempo real de la temperatura con respecto al tiempo. Esto es, crear una base de datos para cada quema tal que el operador no tenga que estar periódicamente tomando nota ni estar graficando.
- Dar mayor seguridad al proceso. El hecho de trabajar con gas combustible obliga a tener cuidado e interrumpir su flujo mediante algún dispositivo electromecánico.
- Ofrecer una alternativa económica. Los sistemas de control de temperatura comerciales más económicos superan los dos mil quinientos pesos (a cotización del segundo trimestre del 2006) en moneda nacional (ver modelos de la compañía TEMPCO México). En este caso el concepto “económico” es engañoso, pues aunque físicamente se desarrolle el sistema desde ceros y el costo material sea inferior al de un sistema comercial, el valor agregado no se contempla en el costo del sistema.
- Por último, que el sistema requiere ser flexible. Dado que la cocción implica varias necesidades y en esta ocasión sólo se van a cubrir tres, las otras deben ser solventadas en futuros proyectos y el sistema debe darles cabida. La meta de este objetivo es que en un futuro también realice el control regulable para la homogenización de la temperatura en el interior del horno.

Con base en éstos objetivos se delimita que el proyecto tendrá las siguientes funciones:

- Monitoreo de la temperatura, que abarca la medición de la temperatura y el método o algoritmo de la cocción empleado en el Taller.
- Registro de los datos de temperatura generados durante la quema, al igual que su graficación. Esta actividad es conocida en inglés como *data log*, que se traduce literalmente como “anotación o registro de datos” y al dispositivo que lo hace se le llama *data logger* o registrador de datos e implica el uso de computadoras. La anotación de datos es muy empleada en la industria para la observación de sus procesos en aras de detectar fallas y mejorarlos [URL 16]. En el Taller también es requerida el registro de datos con el mismo propósito.
- Control del flujo del gas combustible.

Capítulo 2 - EL PROCESO DE COCCIÓN CERÁMICA

El proceso de cocción, conocido también como *quema*, es el paso final en la elaboración de piezas cerámicas. Sin una buena quema el trabajo del artesano (ceramista) se estropea por completo. Son varios los factores que intervienen y en este capítulo serán tratados.

2.1 Elementos de la quema

2.1.1 Materiales y moldeado de piezas

Para elaborar una pieza cerámica se parte de dos elementos: el material adecuado para moldear la pieza y el talento para hacerla (factor que queda totalmente fuera de la jurisdicción del sistema propuesto). De parte del artesano hay una pequeña aportación al éxito de la quema, que es la preparación correcta del material. En el capítulo anterior se mencionó que en el Taller de Cerámica de la UTM sólo se realizan dos tipos de quema, a saber, de sancocho y de esmalte [4], el primero se logra a los 850 °C y el segundo va de 940 °C hasta el límite del horno (1250 °C), dependiendo del tipo de esmalte.

Sancocho (ver figura 2.1a) es la masa, también llamada *pasta*, con la que se moldean las piezas y se compone de barro o arcilla y agua. A partir de éste momento, “barro” y “arcilla” serán referidos indistintamente dado que en el argot ceramista son sinónimos. En razón de que una de las misiones del taller y de la Universidad en general es promover el desarrollo de la Mixteca (oaxaqueña), procuran manejar el material disponible en la región, de esa forma emplean el barro rojo y arcilla caliza. Cada componente tiene propiedades físicas (humedad y factor ph entre ellas) que influyen en la quema, por ello es que su mezcla debe ser en adecuadas proporciones.

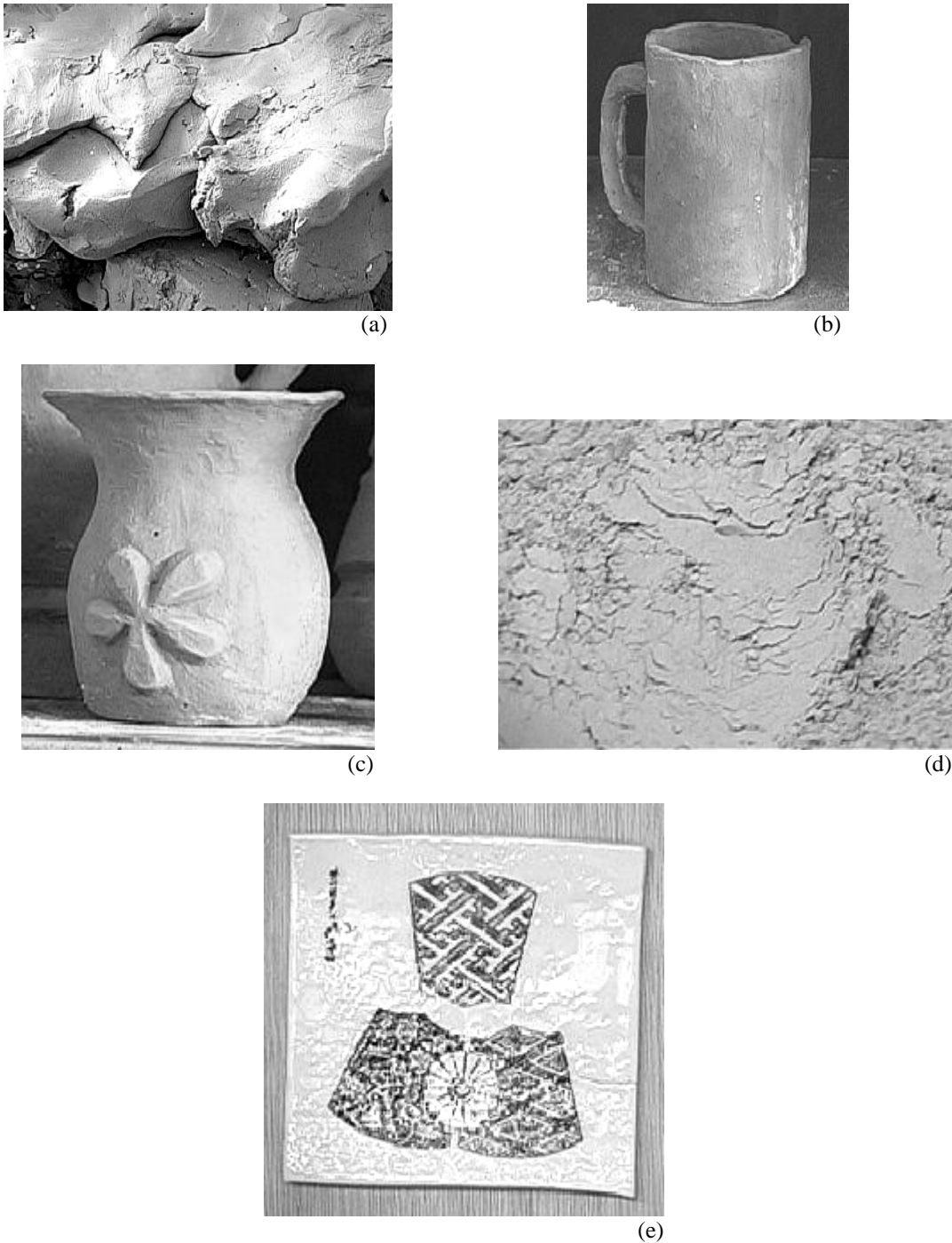


Figura 2.1. (a) Sancocho, o mezcla de arcilla con agua. (b) Pieza moldeada puesta a secar. (c) Pieza cocida. (d) Esmalte en polvo. (e) Pieza esmaltada.

Así, con la masa debidamente preparada se elabora ahora la pieza a voluntad del ceramista, pero tiene que ser secada antes de meterla al horno. Ese secado puede tardar hasta tres días, lo que sea necesario para que el sancocho fresco se deshidrate en la mayor medida posible, pues de no hacerlo, al momento de estarse cociendo el vapor de agua podría provocar la explosión de la pieza. Los experimentados en la materia dicen que cuando la pieza se ha secado y no presenta grieta alguna, significa que el sancocho fue bien preparado. Cuando se somete a cocción y ésta se lleva a cabo exitosamente, formalmente se cuenta con una pieza cerámica.

El esmalte, por su parte, es un polvo fino de estructura cristalina que se compra ya preparado y sirve esencialmente para barnizar la pieza ya cocida. Esto último indica que no se puede cocer el sancocho y el esmalte al mismo tiempo. Lo único que necesita es combinarse el esmalte con agua en proporción de 1 taza de esmalte por dos y media de agua, dejando una consistencia acuosa. La utilidad del esmalte es que evita el desgaste de la superficie del sancocho y con ello el deterioro del objeto cerámico, aparte de que evita que las personas ingieran partículas de sancocho cuando la pieza cerámica sirve como vajilla. Claro que es decisión del artesano si realiza el esmaltado o no, esto es, no es necesario que toda pieza deba ser sometida a este otro proceso.

2.1.2 El horno

El horno con el que cuenta el taller es catalogado como de *tiro invertido* o *llama invertida* [URL 2], que significa que el calor sale por la parte superior del horno, su marca es Basurto y soporta una temperatura máxima de 1250 °C, pero no se recomienda hacerlo trabajar por un tiempo prolongado a dicha temperatura, de hecho, la máxima temperatura a la que se ha trabajado en el taller es de 1050 °C. Puede funcionar a base de gas natural o de gas LP, aunque nunca se ha empleado el primero. En la figura 2.2 se muestra al horno, ahí se pueden apreciar las partes que lo componen, empezando por la puerta (a), que es por donde se introduce un carro – no mostrado en dicha figura – transportador de las piezas, lo único que se requiere para abrirla y cerrarla es correr los seguros que están a su derecha, cuenta además con dos mirillas, que sirven para ver el interior del horno; un riel (b) que sirve para meter y sacar el carro; dos válvulas (c) que sirven para regular manualmente la presión del combustible; d) cuatro hileras de 4 quemadores c/u que proveen el calentamiento del interior del horno; e) un indicador digital basado en termopar y (f) una chimenea en la parte superior que sirve para despresurizar al horno.

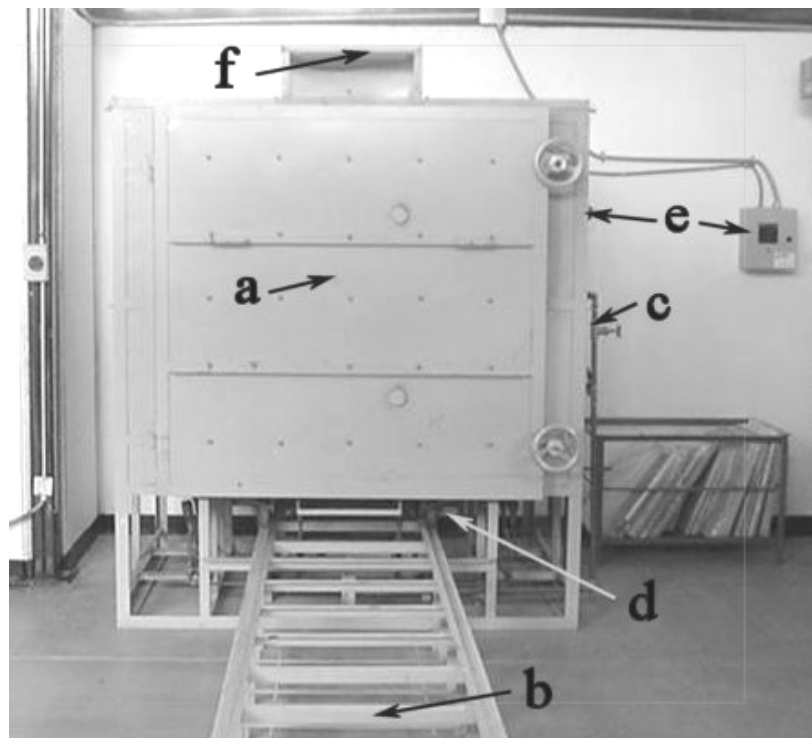


Figura 2.2. Horno de tiro invertido a base de gas, con puerta cerrada y carro en el interior.

En la figura 2.3 se muestra el interior del horno y ahí se aprecian el riel y el carro. El riel se prolonga hasta la cara posterior. El carro, por su parte, es una paleta rodada con un asa para que pueda ser jalado o empujado desde o hacia el horno. A pesar de tener ruedas es pesado y por lo general se emplean dos o tres personas para maniobrarlo, sin considerar la carga de piezas que pueda tener. El interior del horno se observa forrado con una colcha térmica cuya función es proteger las paredes metálicas y proveer la homogenización del calor interno.

En la figura 2.4 se muestran a mayor detalle los quemadores, 14 en total dispuestos en dos hileras laterales de 4 cada una y dos hileras centrales de 3 cada una. De la regulación de su flama (que es resultado de la regulación de la presión en las tuberías de los quemadores) depende el calor del interior del horno. Ésta es regulada por dos válvulas, exhibidas en la figura 2.5, una conocida como válvula lateral (o válvula L) asignada a las hileras laterales y otra conocida como válvula central (o válvula C) para las hileras centrales. Cada válvula tiene también asignada un indicador analógico de presión.



Figura 2.3. Interior del horno, riel y carro.



Figura 2.4. Quemadores.

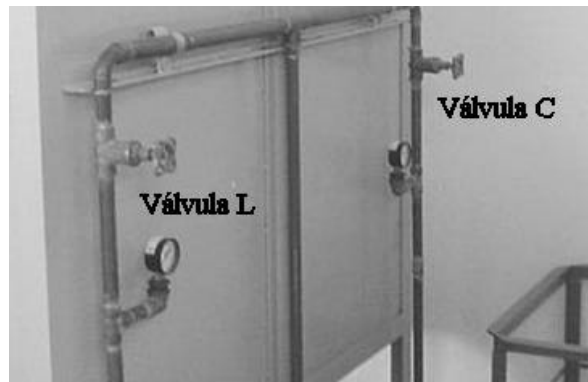


Figura 2.5. Válvulas de regulación con sus indicadores de presión respectivos.

El indicador digital de temperatura con que cuenta el horno está basado en termopar como sensor, es el modelo 2000 de ICEL S.A. de C.V. El termopar que usa es uno tipo K, suficiente para el rango de temperatura del horno. En la parte izquierda de la figura 2.6 está el termopar, situado en la cara también izquierda del horno, su termopozo (estructura que encierra y protege al termopar) tiene recubrimiento cerámico (formalmente de alúmina) y emplea cable de extensión cubierta por una manguera metálica para llegar al indicador, situado en la derecha de la misma figura.

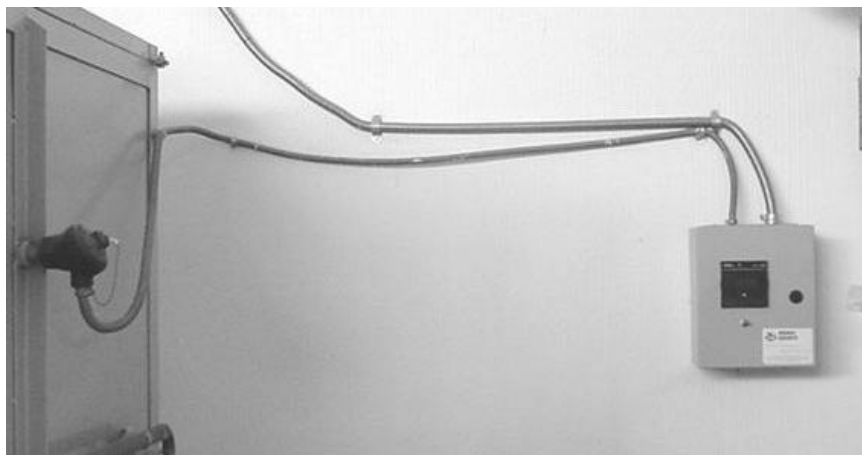


Figura 2.6. Indicador de temperatura basado en termopar.

Por otro lado, el gas combustible viene de un tanque de gas LP estacionario marca TATSA situado en las afueras del taller por razones de seguridad y se conecta a través de una tubería controlada por una llave de suministro principal (ver figura 2.7). Esta llave, cada vez que el horno es usado, se abre a su totalidad, pues la regulación de la presión del gas se realiza con las válvulas situadas a un costado del horno.

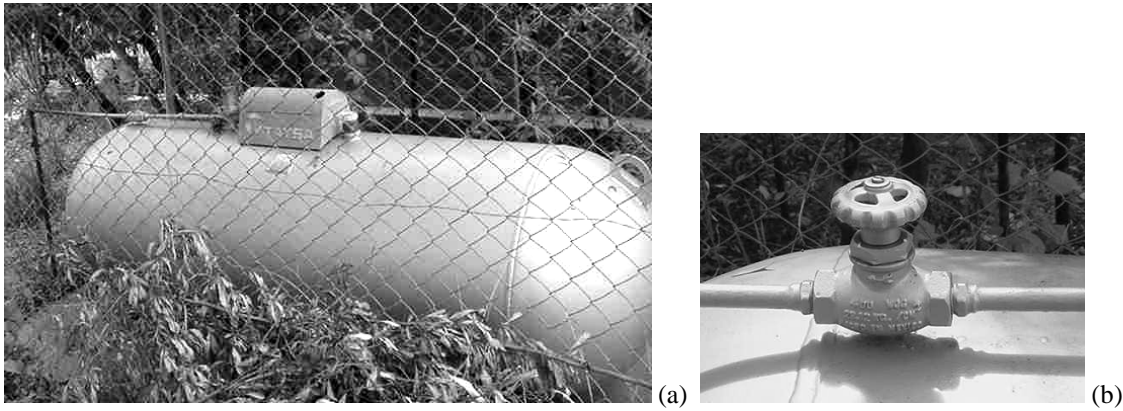


Figura 2.7. Suministro del gas combustible. (a) Tanque estacionario, (b) Llave de suministro principal.

2.2 Algoritmo de la quema

El proceso de cocción empleado en el Taller de Cerámica es el descrito a continuación. Su contenido es la transcripción verbal de la entonces responsable del taller (año 2002), la ceramista María del Carmen Soto La Marina:

“Previo al inicio, el operador tiene que cerciorarse que las válvulas L y C estén cerradas al igual que la llave de suministro en el tanque estacionario y la chimenea; la puerta, por el contrario, se abre totalmente. Por política del taller, debe anotar el nivel de combustible para así llevar un registro del número de quemas efectuadas y saber qué cantidad de gas se emplea en cada quema. Si el nivel del tanque está por debajo del 50%, significa que la presión de la fuente será baja y por lo general, el tiempo de duración de la quema aumenta. También debe revisar que los quemadores estén limpios y sin obstrucciones, como serían trozos de piezas rotas”.

“Con anterioridad el operador debió haber cargado al carro con las piezas (ver figura 2.8), para ello dispone de tablillas y tabiques especiales que soportan el calor que se pueda producir, construyendo plataformas de acuerdo al tipo de piezas. Así mismo, emplea otro indicador de calor conocido como cono pirométrico (figura 2.9), que es un pequeño dispositivo que en realidad tiene forma piramidal y su función es indicar cuando se ha alcanzado una determinada temperatura mediante el doblar de su figura. Hay conos pirométricos de distintas temperaturas, los más usados en el taller son los de 850, 930 y 1005 °C.”



Figura 2.8. Carga del carro.

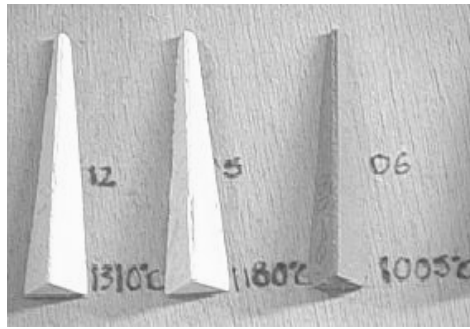


Figura 2.9. Ejemplos de conos pirométricos.

“Se abre completamente la llave de suministro principal y se espera un máximo de 20 segundos para que la tubería se llene de gas. Después se abren las válvulas L y C (referidas a partir de este momento como L y C, simplemente) a mediana capacidad (aproximadamente 0.4 Kg/cm^2) y se deja escapar el gas por un lapso máximo de 10 segundos con el objetivo de que salga un poco el gas y se evacue el aire que posiblemente tenga la tubería”.

“Se prenden con un mechero los quemadores, primero las hileras laterales, del quemador más interno del horno hasta el más externo (de acuerdo al perfil frontal del horno), es decir, de atrás hacia delante. Igual se hace con las hileras centrales”.

“Se ajusta ahora la presión de L y C inicialmente a 0.1 Kg/cm^2 . Se espera un máximo de 20 minutos (precalentado del horno) antes de introducir el carro con la carga sin cerrar la puerta, de hecho, hay que dejar pasar otros 10 minutos antes de cerrarla, pues podrían las piezas sufrir fracturas. A partir de este momento, hay que revisar periódicamente la temperatura valiéndose del indicador digital, puesto que hay dos puntos considerados críticos en la cocción: la inversión de la cristobalita ($230 \text{ }^\circ\text{C}$) y la inversión del cuarzo ($570 \text{ }^\circ\text{C}$). La cristobalita y el cuarzo son dos componentes cristalinos que se encuentran en pequeñas proporciones en el sancocho, y su inversión se refiere al momento en que sus estructuras cristalinas se maduran, con lo que se logra un precocimiento de las piezas”.

“Después de pasada una media hora, independientemente de que se hayan alcanzado las inversiones de la cristobalita y del cuarzo, se incrementa la presión de L y C en 0.05 Kg/cm^2 , es decir, ahora se tendrían 0.15 Kg/cm^2 en las válvulas. Por lo general, con tal presión y tal tiempo de espera, la inversión de la cristobalita ya debió haber sucedido, faltando la del cuarzo. Por ello se sigue esperando hasta lograrla, aumentando la presión en 0.05 Kg/cm^2 cada media hora, no importa cuanto tiempo lleve alcanzar dicha inversión”.

“Una vez alcanzada la inversión del cuarzo, la temática es ir aumentando gradualmente, en 0.05 Kg/cm^2 cada media hora, la presión de L y C. Para ello, aparte del indicador digital de temperatura, se recurre a la apreciación visual. Esta consiste en ver el color de la colcha en el interior del horno, si ésta alcanza una tonalidad de combinación de colores naranja y rojo, significa que el horno se encuentra en trabajo óptimo, entendiendo por trabajo óptimo que el horno tiene un gradiente de temperatura (idealmente) igual en todo su volumen interior”.

“Si el tipo de quema que se está haciendo es de sancocho, su cocimiento se alcanza a los $850 \text{ }^\circ\text{C}$ y siguiendo con la temática del párrafo anterior, todo es cuestión de llegar a dicha temperatura con los incrementos periódicos de la presión. Cuando se alcance dicha temperatura,

se cierra completamente la llave del suministro principal (es decir, la del tanque estacionario), dejando que todo el gas que resta en la tubería se consuma. Asimismo, se cierra la chimenea para evitar un efecto conocido como choque térmico, consistente en una rápida despresurización del horno y desestabilización de la temperatura, que puede causar un mal acabado de la quema y la ruina de las piezas. Así se da por terminada la quema de sancocho, ahora hay que dejar enfriar al horno. Si el operador lo considera prudente, puede abrir la chimenea para acelerar el enfriamiento y después verá cuando es prudente abrir la puerta y sacar el carro.”.

“Pero si la quema es de esmalte, continuar con la temática antes dicha y esperar hasta alcanzar los 900°C, temperatura a la cual el esmalte entra en un periodo conocido como reposo o maduración, cuyo objetivo es que el esmalte empiece a solidificarse y adherirse a la pieza. Este periodo tiene una duración de 15 minutos durante los cuales la temperatura del horno no debe salir del rango de 900 a 905 °C, en caso de que empiece a subir, hay que reducir la presión de L y C en 0.01 Kg/cm², en caso contrario, aumentarla en la misma cantidad“.

“A partir de estos momentos se puede recurrir a los conos pirométricos, los cuales, dependiendo de su valor, se irán doblando al alcanzar su temperatura. Se retoma la temática del aumento gradual de la presión antes descrita. El proceso continúa hasta alcanzar el punto de cocción del esmalte en cuestión, cuando llegue ese momento, se cierra completamente la llave de suministro principal y se cierra la chimenea, con el objetivo de hacer madurar al esmalte mientras el horno empieza a enfriarse. Si el operador lo considera prudente, puede abrir la chimenea para acelerar el enfriamiento y después verá cuando es prudente abrir la puerta y sacar el carro”.

Todo este texto se resume en cuatro puntos:

- Preparación del proceso. Abarca la verificación de todas las partes involucradas, la colocación de las piezas en el carro, el encendido de los quemadores el ajuste inicial de las válvulas, el precalentado del horno y la introducción del carro. A partir de este punto, la temperatura a la cual se logra la cocción ya sea del sancocho o del esmalte, será referida como temperatura objetivo.
- Primera etapa de la quema: Esta etapa durará hasta que pase una de dos cosas. Si la quema es de sancocho, la temperatura objetivo es de 850°C y al alcanzarla se interrumpen tanto la etapa como el resto del proceso al cerrarse la llave principal y dejar al horno enfriarse. Si la quema es de esmalte, la temperatura objetivo es mayor a 900°C y hay que llegar hasta la temperatura de reposo (900°C) para terminar esta etapa. Simultáneamente, cada 30 minutos se incrementa la presión de las válvulas L y C. Antes de llegar al fin de esta etapa, la inversión del cuarzo y de la cristobalita tuvieron lugar y es pertinente registrarlos.
- Segunda etapa de la quema: Durante quince minutos, mantener la temperatura en el rango de 900°C a 905°C.
- Tercera etapa: Llegar a la temperatura objetivo del esmalte en cuestión. Al igual que en la primera etapa, cada media hora se incrementa la presión de las válvulas L y C. Una vez hecho esto, cerrar la llave principal y dejar al horno enfriarse.

La figura 2.10 muestra otra forma simplificada de comprender la quema.

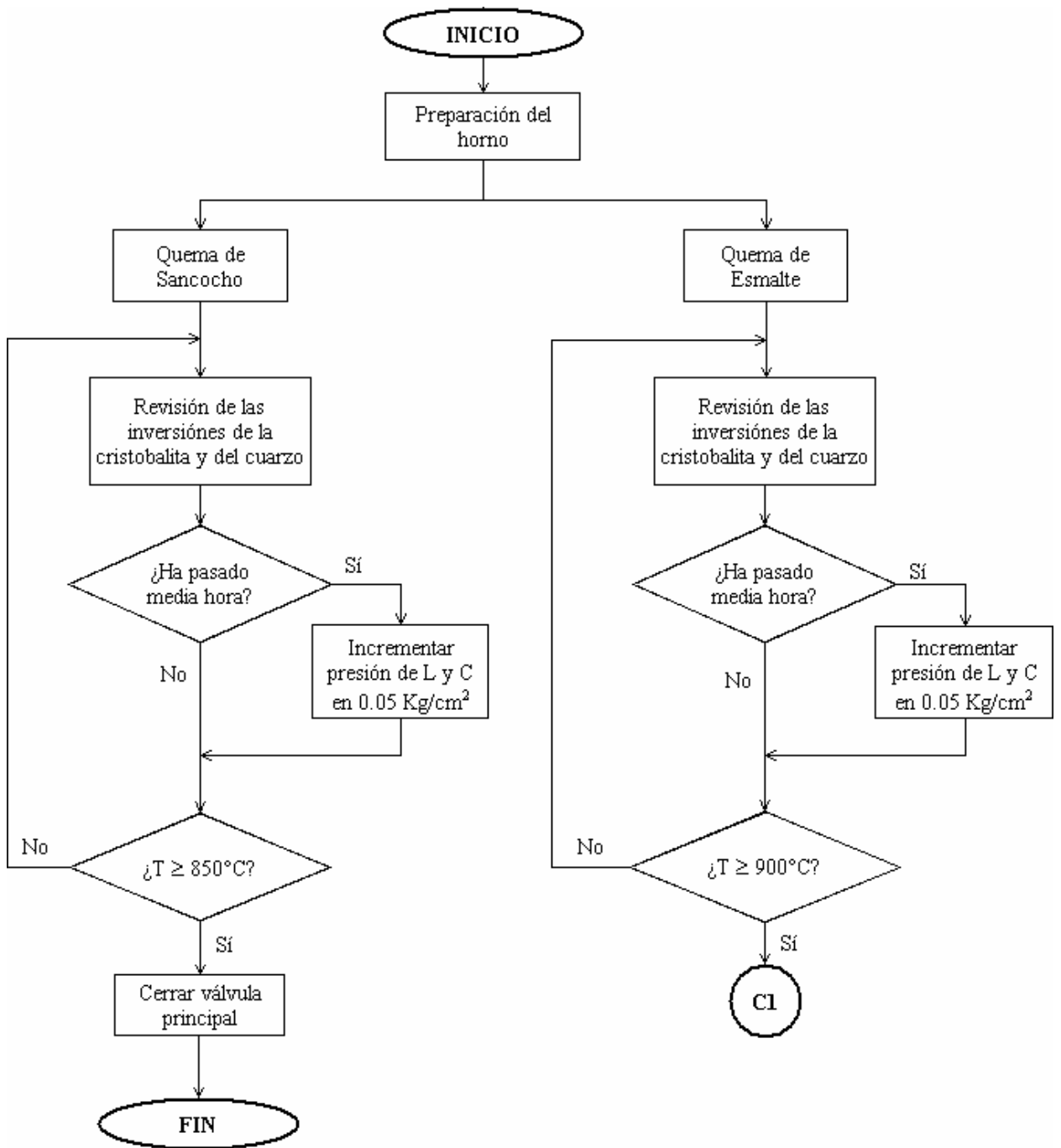


Figura 2.10. Diagrama de flujo simplificado del procedimiento de cocción cerámica o quema.

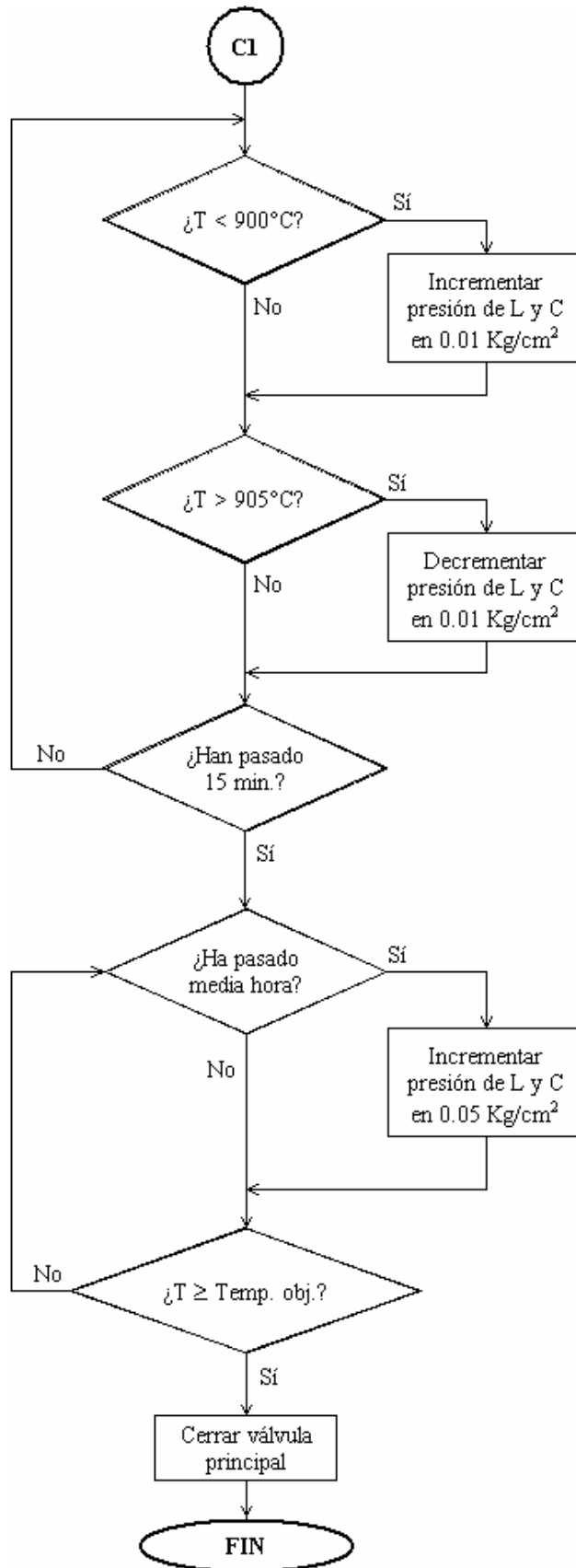


Figura 2.10. Diagrama de flujo simplificado del procedimiento de cocción cerámica o quema (continuación).

2.3 Revisión de sistemas relacionados

En el capítulo 1 se delimitó el proyecto y la forma en cómo satisface las necesidades del taller, ahora es necesario hacer un estudio de sistemas similares para determinar su eficiencia relativa. En razón a ello se adelanta la siguiente información sobre este proyecto:

- Se emplea una computadora personal y un sistema compuesto por un microcontrolador para el monitoreo del proceso de cocción.
- La computadora funge como el control principal del monitoreo y el microcontrolador es simplemente una unidad de adquisición de datos, pero puede relevar a la computadora en caso de que ésta falle, en consecuencia, el monitoreo se garantiza.
- Tanto la computadora como el microcontrolador contienen el método o algoritmo de quema empleado en el Taller (por la finalidad que tiene el Taller, sólo se emplea uno) y es posible iniciar una quema desde cualquiera de ellos.
- La computadora tiene la facultad de registrar los datos de temperatura generados durante el proceso y de graficar en tiempo real la temperatura con respecto al tiempo (cuando tiene el control) e imprimir registros de quemas anteriores.
- El microcontrolador no puede hacer las tareas anteriores, pero almacena los datos en una memoria y es posible descargarlos en la computadora una vez que termina la quema.
- La medición se hace con un termopar tipo K (cubre el rango de temperatura del termopar).
- Por seguridad en el proceso se incluye una electroválvula normalmente cerrada para permitir o detener el flujo del gas combustible. Es posible abortar el proceso si el operador lo considera necesario ya sea desde la computadora o desde el microcontrolador.
- Para prevenir cortes de energía que a su vez interrumpan el monitoreo, se incluye un sistema de respaldo o UPS.
- El proyecto tiene cabida para funciones y sensores que optimicen su desempeño en futuras expansiones.
- El costo material fue inferior a dos mil pesos a índice de precios del segundo trimestre del 2006.

En el mercado hay muchos sistemas relacionados con control y medición de temperatura y otros más relacionados con el registro de datos. Los ejemplos que se expondrán a continuación son los más parecidos al presente proyecto y están orientados a la industria, en consecuencia el número de sus funciones es mayor. No necesariamente se enfocan a hornos de gas, pero pueden adaptarse a uno.

2.3.1 Spectrum 1700

La compañía Veriteq Instruments Inc. (Canadá) se enfoca al desarrollo de herramientas de guardado de datos durante el monitoreo de temperatura y de humedad, parte de ellas se relacionan con hornos.



Figura 2.11. Registrador de datos Spectrum 1700

La serie Spectrum 1700 [URL 17] de registradores de datos (data loggers) está diseñada para trabajar con termopares de tipos J, K, T, E, R y S (se adquieren por separado), realizando por sí sola la compensación de la unión fría, acepta hasta cuatro termopares de entrada. Guarda datos de temperatura desde -240°C hasta 1760°C y los retiene un máximo de diez años gracias a su batería de larga duración. Alta precisión en la medición, acreditada por el estándar ISO 17025. Se calibra por el software Veriteq sCal y la generación de reportes es mediante el software Spectrum Data Logger. Se comunica vía puerto serial RS-232 o por USB. Su uso es sencillo y se adapta las necesidades industriales en el monitoreo de temperatura. Su costo se determina mediante cotización.

Principales campos de aplicación

Verificación de procesos térmicos, hornos de alta temperatura (como los empleados en cerámica), producción de aluminio, cualquier proceso térmico que requiera cumplir con el estándar ISO 9000/EPA/HACCP.

Operación

Para operar simplemente se selecciona el tipo de termopar previamente conectado y se marca el inicio de la grabación de datos con la periodicidad indicada. La programación es sencilla, basta con ingresar los parámetros en el software.

Software Spectrum Data Logger

Funciona en ambiente Windows, sirve para grabar información en archivo, analizar y graficar datos, cuenta con herramientas de análisis tabular y estadístico, puede combinar gráficas para estudios comparativos, exporta los datos a hojas de cálculo empleadas por programas populares como Microsoft Excel. También imprime gráficas.

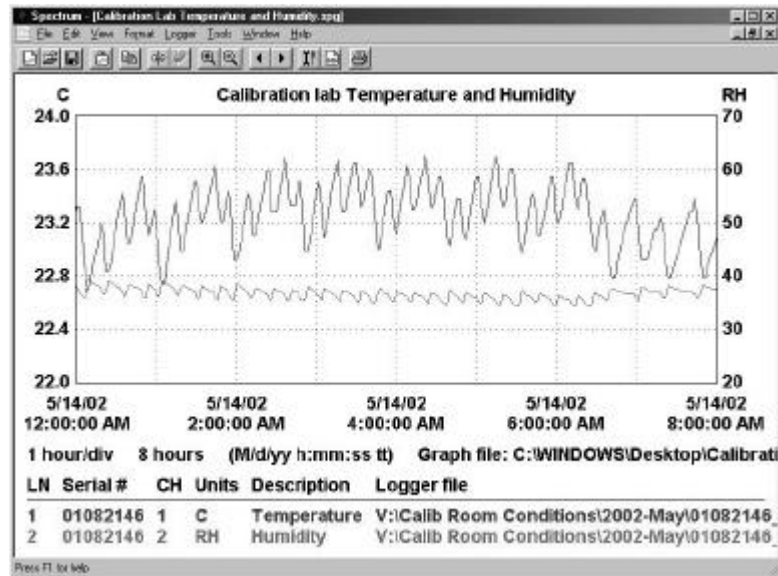


Figura 2.12. Software Spectrum Data Logger.

Comparación con el proyecto

Este ejemplo es solamente un registrador de datos, no realiza control de la quema y solo puede servir para monitorear, pero sus herramientas de generación de reportes son poderosas e incluye análisis de datos. La característica de análisis de datos es útil porque esta serie se adapta a varios campos que pueden ser totalmente diferentes a la cerámica, el proyecto no los incluye porque no los necesita, pues las actividades del taller no lo ameritan. Para adaptar el Spectrum 1700 a un horno basta con incluir el o los termopares adecuados al tipo de horno, que en el caso del taller sería un tipo K.

2.3.2 AutoFire™

La compañía Orton Foundation (EE.UU.) tiene tres áreas de producción y servicios: dispositivos de monitoreo pirométrico para procesos térmicos, diseño y construcción de instrumentos analíticos para la medición de propiedades térmicas y el servicio a empresas para la prueba de materiales (en lo que la obtención de propiedades térmicas se refiere).

Es la primera área la que se relaciona con este sistema al ofrecer los controladores de temperatura AutoFire™, diseñados para encender automáticamente hornos eléctricos, ofreciendo un control preciso de temperatura. Sus controladores tienen la característica de ser programables, con teclados fáciles de usar y con robustos programas de control. Esos controladores pueden ser usados como paneles de componentes (es decir, para hacer más de una quema a la vez en varios hornos) o como sistema solitario de control. Además, cada controlador puede ser puntual (tiene un solo sensor) o zonal (hasta tres sensores distribuidos en el horno).

Principales campos de aplicación

Quemas cerámicas, tratamiento de calor, fundición de vidrios y esmaltado.

Operación

La obtención de la temperatura viene de termopares que son colocados dentro del horno. El operador selecciona un programa preestablecido o diseña un itinerario (algoritmo) personalizado para que el controlador lo ejecute. La temperatura actual del horno es comparada con el ciclo de

quemado programado y cuando sean iguales, el controlador conmuta sus relevadores internos para apagar el dispositivo de calentamiento. De igual forma, puede regular la cantidad de energía eléctrica para mantener el itinerario deseado en la quema.

Para quemados cerámicos, los controladores AutoFire™ tienen itinerarios de quemado preestablecidos que intentan proveer la temperatura adecuada durante el tiempo exacto para determinadas mezclas de materiales.

Programación en pasos múltiples

Los itinerarios personalizados de quemado pueden requerir varios pasos para calentar y enfriar al horno a distintas temperaturas en cada uno. Es posible hacer itinerarios prolongados donde cambien los rangos de calentamiento, los rangos de enfriamiento e incluso las temperaturas de referencia. Además, es posible también que el operador modifique los rangos durante la quema, evitando tener que parar el horno y reprogramar al controlador cuando los ajustes sean necesarios. La programación se hace mediante el teclado del controlador.

Controlador AutoFire Standalone 3000

Este es un miembro de la serie AutoFire™ [URL 18], compatible con la mayoría de las marcas de hornos eléctricos y se provee con una gran variedad de cables de energía y receptáculos de hornos para una fácil instalación. Se incluyen bases de montaje en paredes diseñadas para ser colocadas cerca del horno. Los termopares se venden por separado con opción a incluir extensiones de termopar de hasta 8 pies. Sus características son:



Figura 2.13. Controlador AutoFire Standalone 3000.

- Tamaño compacto: 6x10x4 pulgadas.
- Fácil instalación.
- Bases de montaje resistentes al medio.
- Configuración para una sola zona o para varias (hasta tres termopares).
- Accesorios para montaje.
- Acepta termopares tipo K y tipo S.
- Accesorio Hand-held para operar fuera de su base.
- Opción de incluir interfaz serial RS-232 con computadora.
- Costo: 650 USD (abril de 2006).
- Garantía de dos años.

Software de monitoreo por computadora

Se puede adquirir por separado un software que permite al operador programar y monitorear remotamente al controlador desde una computadora personal. El nombre del software es ControlMaster DX™, elaborado y distribuido también por Orton Foundation. Sus características son:

- Compatible con Windows.
- Programación y monitoreo de hasta 8 hornos a la vez desde una sola computadora.
- Graficación en tiempo real de la temperatura con respecto al tiempo.
- Lectura remota de la temperatura.
- Registro de datos en archivos y su exportación.
- Herramientas de análisis de datos.
- Impresión de gráficas.
- Costo: sobre cotización.

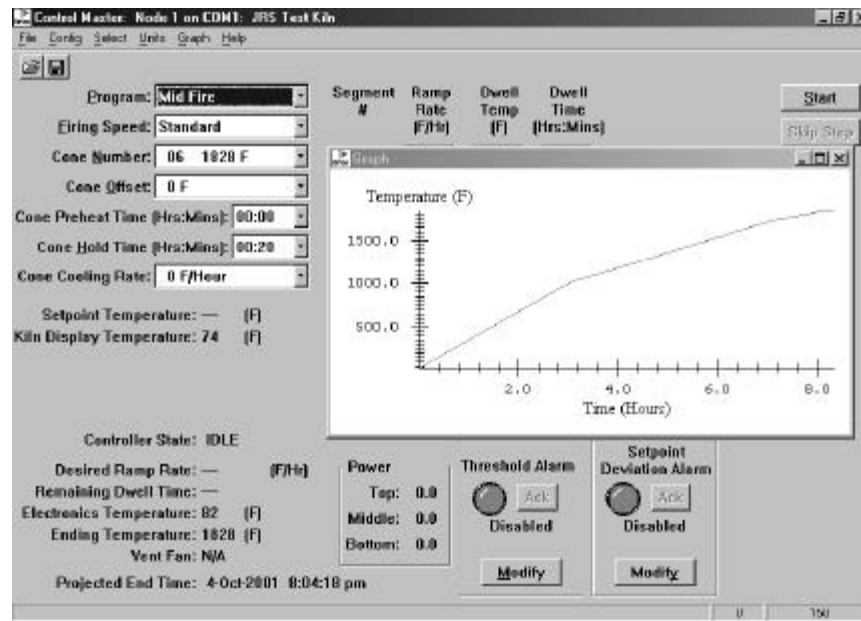


Figura 2.14. Software ControlMaster DX™.

Comparación con el proyecto

Este es el ejemplo más parecido al proyecto. Ambos pueden realizar registro, graficación e impresión de los datos generados durante la quema y el método de la quema puede configurarse de acuerdo al que se emplee en el caso.

2.3.3 Tpaq21

La compañía DataPaq [URL 19] (EE.UU.) fabrica sistemas orientados a ambientes térmicos hostiles (temperaturas extremas) para el registro de datos, análisis y monitoreo de procesos que involucren hornos y congeladores de distinto tipo.

Sus productos relacionados con el monitoreo de temperatura son de tres tipos: registradores de datos, software de registro y análisis de datos y aisladores para una correcta medición de la temperatura en el horno (es útil principalmente para la fabricación de ladrillos).

Principales campos de aplicación

Hornos de cerámica y de ladrillo.

Registrador de datos serie Tpaq21

Es un dispositivo de alta precisión y tamaño compacto (8x4x1 pulgadas), registra hasta 130 mil mediciones distribuidas en diez canales de entrada. Los intervalos de muestreo variables y la programación de eventos múltiples proveen un análisis detallado donde los procesos de tratamiento de calor comprenden varios pasos. Su costo se determina mediante cotización. Sus atributos son:

- Capacidad de hasta diez canales de termopar. Acepta los tipos K, N, R y S.
- Alta capacidad de memoria: 130,000 datos.
- Intervalos de muestro menores a 0.1 segundos. Además cuentan con la propiedad de ser variable durante el proceso
- Precisión de $\pm 0.3^{\circ}\text{C}$.
- Transmisor RF para telemetría.
- Interfaz de comunicación con computadora (USB y RS232).
- Bajo consumo de energía, puede ser alimentado con baterías.
- Gabinete resistente a ambientes hostiles (temperaturas extremas, polvo, presión, etc.).
- Botones de fácil uso.



Figura 2.15. Registrador de datos Tpaq21.

Software Kiln Insight

Este programa es quizás el más poderoso de los que se muestran en la presente sección. Abarca tres áreas: análisis, telemetría (grabado a distancia de los datos ofrecidos por el registrador de datos) y registro de procesos.

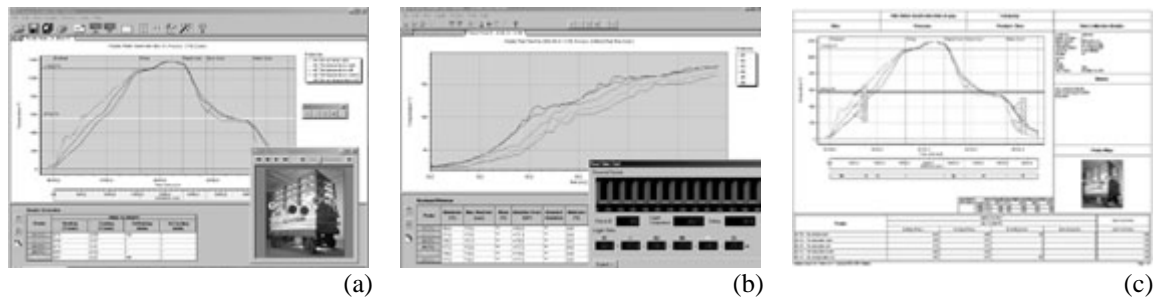


Figura 2.16. Software Kiln Insight. (a) Análisis, (b) Telemetría, (c) Registro de procesos

El análisis comprende la detección de índices programables, cálculo de pendientes, visión de datos, determinación de puntos máximos y mínimos, comparación entre registros, etc. Cuenta con tutoriales (llamados comúnmente *wizards*) para aprender a configurar el análisis de acuerdo al tipo de proceso en que se aplique. Su característica más robusta es que cuenta con modelado en tres dimensiones del horno para determinar la mejor colocación de los sensores e incluso la distribución de los elementos dentro del horno.

La telemetría, por su parte, no es más que la propiedad de estar recibiendo los datos remotamente. El registro, finalmente, es el guardado de los datos en archivo, listos para ser analizados por sus poderosas herramientas.

Comparación con el proyecto

Este ejemplo, a pesar de ser un registrador de datos, tiene más robustez que el Spectrum 1700, este sí puede implementar un algoritmo de cocción, pero no efectúa la regulación de temperatura. Su principal atributo es la poderosa herramienta de modelado en tercera dimensión para la correcta distribución de termopares y de las piezas. Esto en el taller cerámica

2.3.4 Instrumentación de horno de gas

Este es el único ejemplo no comercial e involucra un horno de gas, se trata de un proyecto efectuado en el College Park (EE.UU.) titulado *Gas & Induction Kiln Instrumentation*, o su versión en español *Instrumentación de Horno de Gas y de Inducción* [URL 20]. El trabajo muestra el funcionamiento del control y la instrumentación para un horno de gas donde es empleado un controlador comercial, que tiene grabada una secuencia de seguridad para evitar percances con el horno. El trabajo incluye también un horno de inducción pero no es objeto de interés en este caso.

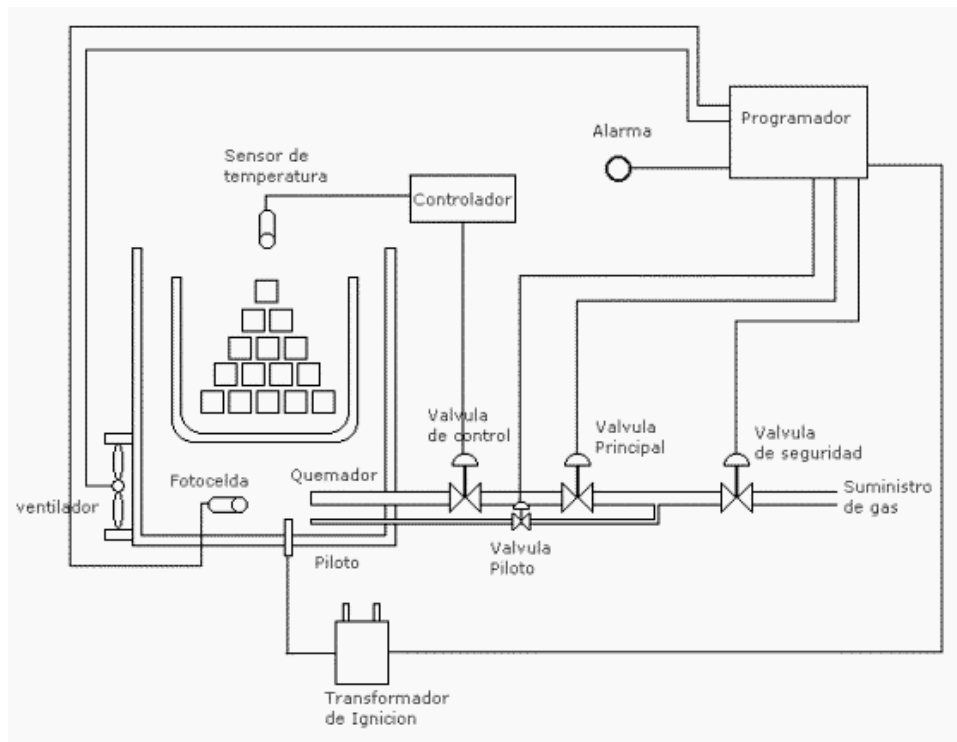


Figura 2.17. Diagrama del proyecto.

Los pasos son los siguientes:

1. Manualmente se abre la válvula de seguridad la cual a su vez está controlada por el programador, al haber algún problema el controlador cerrará la válvula automáticamente.
2. Se acciona un ventilador que funcionará por un tiempo programado para desalojar cualquier cantidad de gas en el horno que pueda ocasionar una explosión.
3. El programador abrirá la válvula del quemador piloto y al mismo tiempo creará un arco eléctrico por medio del transformador de ignición.
4. Un detector de flama determinará si existe flama en el pilo, en dado caso, se podrá pasar al siguiente paso.
5. El programador abrirá la válvula principal y ahora la regulación de la temperatura es efectuado completamente por el controlador.
6. Por medio de un sensor de temperatura se le hará llegar la medición al controlador y éste a su vez regulará el paso del gas empleando un actuador.

Los materiales en el trabajo son de la marca HoneyWell. En lo que al costo del proyecto se refiere, no aparece dicho dato en la fuente de información.

Controlador digital. Realiza el control de la temperatura, recibe la información desde un sensor de temperatura y controla una válvula. El modelo es el UCD-2000



Figura 2.18. Controlador universal empleado en el proyecto.

Detector de flama. Funciona como detector de flama, recibe la señal por medio de un amplificador de estado sólido el cual esta conectado a una fotocelda. Este dispositivo registra a cada momento la presencia de la flama. Al no detectar flama puede cerrar las válvulas de gas en un tiempo de retardo de 4 segundos y accionar una alarma. El modelo es el RA 890F.

Fotocelda. Es la que formalmente detecta la flama, activa una señal que va al detector de flama. El modelo es el C7035A.

Detector radial de temperatura. Detecta la temperatura de un horno por medio de la radiación emitida por el metal al calentarse (de acuerdo a la temperatura que mida emite una radiación que es convertida en señal eléctrica). El modelo es el STT350.



Figura 2.19. Detector radial de temperatura.

Actuador. Es un servomecanismo que regula el paso del gas al acoplarse a las válvulas, es decir, suple la acción de la mano del operador al regular la válvula. El modelo es el M7410E.



Figura 2.20. Actuador.

Comparación con el proyecto

A diferencia de los otros ejemplos, aquí no se hace el registro de datos ni la graficación, pero sí la regulación de temperatura mediante el control del paso del gas. En lugar de usar una servoválvula emplea un actuador que se alinea con la válvula de control y de acuerdo a la estimulación que recibe, abre o cierra en cierta proporción a la válvula. La desventaja de este método es que si la alineación no es correcta, la regulación va a ser mala también. Es preferible usar una servoválvula aunque su costo sea mayor que el del actuador empleado.

2.3.5 Resumen comparativo del presente proyecto con los demás sistemas.

Al describir cada uno de los ejemplos de sistemas relacionados con el proyecto efectuado se hizo la comparación por separado. La tabla 2.1 contiene el resumen comparativo de todos los sistemas y/o proyectos.

Los sistemas Spectrum 1700 y TPaq21 son registradores de datos comerciales orientados al registro de temperatura. La cantidad de funciones que contiene el programa o software de cada uno es mucho mayor al del proyecto efectuado en esta tesis, pero hay que tomar en cuenta que no son exclusivamente para hornos de cerámica, sino que pueden adaptarse a más procesos térmicos. De ahí viene la inclusión de herramientas de análisis y en el caso del Taller de Cerámica de la Universidad no son necesarias, pues sus procesos se remiten a quemas de sancocho y de esmalte con un método o algoritmo establecido. El costo de ambos sistemas se determina mediante solicitud formal de cotización.

El sistema AutoFire Standalone 3000 no solamente hace el registro de datos (solo con la adquisición por separado de su software respectivo), pero tiene mucho parecido con las funciones del proyecto de esta tesis: el sistema puede iniciar el proceso de quema desde él

mismo o desde la computadora, el método de la quema es programable y por ende da seguimiento y regula la temperatura. Aunque está diseñado para hornos eléctricos, tiene una señal de salida proporcional que podría ser adaptada para operar una servoválvula y controlar así un horno de gas. La ventaja del proyecto de esta tesis es que es más barato.

Finalmente, el proyecto realizado en el College Park realiza el control regulable de temperatura para un horno de gas. Es de observarse que al contrario de los demás sistemas, que emplean termopares para la medición de temperatura, emplea un detector radial de temperatura. Este tipo de sensor no es nuevo, pero su uso no está tan difundido, quizás porque los termopares aún son vigentes y no hay necesidad de reemplazarlos. Aunque el proyecto del College Park no está pensado precisamente para un horno de cerámica, no hay mayor problema para adaptarlo a uno, sólo le faltaría el registro de datos para cumplir con los requisitos del Taller de Cerámica. Como se mencionó en su momento, la regulación del paso del gas se hace con un actuador que se va a montar en la válvula de control, pero no es la mejor opción para la regulación, pues se debe cuidar que esté perfectamente alineado con el eje de rotación de la válvula y en caso de que llegase a desalinearse, la regulación fallará de una manera indeterminada.

Tabla 2.1. Resumen comparativo del proyecto con los sistemas revisados.

	Proyecto efectuado	Spectrum 1700	AutoFire Standalone 3000	TPaq21	Proyecto del College Park
Aplicación en tipo de horno	Gas	Cualquier tipo	Eléctrico	Cualquier tipo	Gas
Tipo de transductor empleado	Termopar tipo K	Termopares tipo J, K, T, R, E, S.	Termopar tipo K o S	Termopares tipo K, N, R y S	Detector radial de temperatura
Cobertura de temperatura	Puntual*	Puntual o zonal	Puntual o zonal	Puntual o zonal	Puntual
Control regulable de temperatura	No*	No	Sí	No	Sí
Control on/off	Sí	No	Sí	No	No
Realiza reporte de la quema	Sí	Sí	Con adquisición de software	Sí	No
Grafica en tiempo real	Sí	Sí	Con adquisición de software	Sí	No
Imprime gráfica	Sí	Sí	Con adquisición de software	Sí	No
Método o algoritmo	Fijo, personalizado al taller	No	Programable en varios pasos	Programable	Fijo
Respalda energía en caso de corte.	Sí	No	No	No	No

(*) En futuras expansiones puede ser cubierto o mejorado.

Capítulo 3 - DISEÑO DEL SISTEMA

El diseño del sistema está en función a dos factores: el cumplimiento de los objetivos expresados en el capítulo 1 y la implementación del algoritmo simplificado descrito en el capítulo 2.

3.1 Consideraciones generales del diseño

Reflexionando acerca de todas las necesidades descritas en el capítulo uno, los siguientes puntos representan los principales aspectos a considerar en el diseño.

- El manejo del sistema por el usuario debe ser sencillo. Es decir, de nada sirve desarrollarlo si la persona o las personas que van a operarlo no lo entienden. Este punto indica la necesidad de crear una interfaz amigable. Además, la inclusión de un manual de operación es ineludible. No olvidar que uno de los objetivos es el auxiliar al operador durante la quema.
- El sistema debe tener la capacidad de recobrase a infortunios. Se entiende por *infortunios* a incidentes externos tales como cortes de energía.
- La temperatura máxima que soporta el horno es de 1250 °C. Difícilmente se encontrarán sensores de tipo circuito integrado (cuya respuesta es lineal) que cubran este rango de temperatura, por ello se deben emplear otro tipo de sensores. Los basados en termopares [URL 3] – específicamente los de tipo K – son la mejor opción: son económicos con relación a otros, soportan temperaturas altas y son muy comunes. El punto malo es que su respuesta no es lineal aunque para ello existen ya dispositivos que corrigen ese problema (y siguen siendo económicos). En el Apéndice A se trata el tema de los termopares.
- El manejo del gas debe ser efectuado con precaución, hay que recordar que es un elemento inestable y que cualquier chispa eléctrica podría provocar un serio accidente. Por ello se recomienda emplear dispositivos comerciales que ofrezcan *seguridad* en el manejo del combustible. En el mercado existen dos clases de dispositivos electromecánicos apropiados para el manejo de combustibles y fluidos en general. La primera, que es la elegida en este proyecto, es la electroválvula [URL 4], conocida también como válvula solenoide, sirve como un conmutador del combustible en cuestión, es decir, sólo tiene dos estados: abierto o cerrado. Su operación básica es muy sencilla, basta con polarizar adecuadamente al solenoide y éste abre o cierra la válvula. La otra clase es representada por la servoválvula, que a diferencia de las

anteriores, tienen varios estados incluyendo el totalmente cerrado y el totalmente abierto. El problema es que son muy caras y son viables únicamente cuando la producción es a gran escala y el volumen de ventas compensaría la inversión (no olvidar que otro objetivo del proyecto es que sea económico). En el apéndice A se abarca también el tema de las válvulas solenoides o electroválvulas.

- Para realizar un registro de la quema y la respectiva gráfica, el elemento más viable es una computadora electrónica, incluso se pueden generar reportes completos y llevar el control del proceso a través de la computadora. Pero atendiendo el cuarto punto (recuperación del sistema a infortunios) se observa que un corte de energía paralizaría el proceso. La respuesta sería la utilización de un dispositivo de respaldo de energía eléctrica, conocidos como *no-break* o *UPS* [URL 5] (del inglés *uninterruptible power system* o sistema de poder sin interrupción), pero éste no puede hacerlo por mucho tiempo aunque depende, claro, de la capacidad de la batería del UPS.

3.2 Esquema primario del sistema

Tomando en consideración todos los puntos anteriores, la representación primaria del sistema es la siguiente:

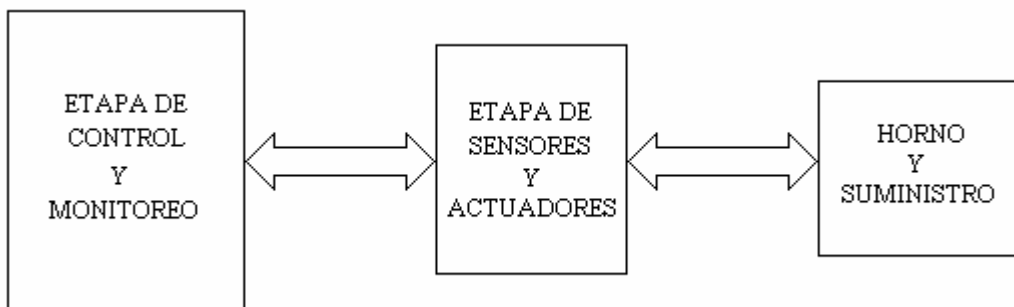


Figura 3.1. Diagrama a bloques primario del sistema.

El concepto es muy sencillo, la etapa de control y monitoreo contiene toda la lógica del proceso, conoce todos los pasos a seguir en la quema y realiza las gráficas y reportes correspondientes. Para conocer en qué momento debe efectuar una acción específica requiere de información, la cual es ofrecida por los sensores de temperatura. Además, requiere de dispositivos que realicen físicamente la acción, un nombre muy común que se les da es el de actuadores y en este caso, el actuador indicado es una electroválvula. El sensor es el que tiene relación directa con el horno, de hecho, debe estar situados ahí. El actuador, por su parte, debe estar lo más próximo al suministro de gas, específicamente, después de la llave de suministro principal de la figura 2.7.

El control que se menciona en esta sección no se refiere a un control regulador de temperatura, sino al control del proceso de monitoreo, es decir, el que lleva la lógica de acuerdo con el algoritmo simplificado de la quema. A partir de este momento, toda referencia a *control* en el presente documento se refiere a este concepto.

3.3 Propuesta para la etapa de control y monitoreo

La etapa de control y monitoreo es el corazón del sistema. Profundizando en sus características, éste lleva la lógica del proceso, lo cual implica que debe tener grabado todo el proceso y todas las acciones que deban ser tomadas ante cualquier situación que se presente. También se encarga de elaborar reportes y gráficas de la quema.

Por todos es conocido el potencial de una computadora personal electrónica (PC), de hecho, es el primer candidato. Son tan versátiles que prácticamente cualquier problema puede ser solucionado con ellas, todo es cuestión de tener los conocimientos adecuados sobre programación y adquisición de datos. Es pertinente remarcar que las computadoras típicamente no tienen *unidades de adquisición de datos* [URL 6] (también referidas como *tarjetas de adquisición de datos*), éstas se adquieren aparte. Son varias las compañías que fabrican estas unidades, ofreciendo una amplia gama de opciones para ajustarse a las características del problema a resolver.

Lamentablemente, como se expuso en la sección 3.1 (consideraciones generales del sistema), su desventaja es que deben tener garantizado el suministro de energía, situación que se lograría a través de un UPS, el cual a su vez tiene la desventaja de no poder sostener por mucho tiempo a una computadora que conste de los elementos más básicos, a saber, el gabinete con todos sus componentes internos, el teclado, el ratón y el monitor, aparte de la tarjeta de adquisición de datos. Para un corte de energía, como es bien conocido, no existe forma de determinar cuánto durará salvo que sea intencional. Si el tiempo que dure el corte es mayor al que puede soportar el dispositivo, la computadora pierde totalmente el control y el proceso se viene abajo.

El suministro de energía no es el único problema, existen más factores que podrían afectar el proceso por computadora, tales como la presencia de virus, pues podrían afectar al sistema operativo de la computadora y en consecuencia afectar al programa. Si para el día que se tenga programada una quema la computadora no está disponible, el proceso tendría que ser totalmente manual. O bien, pensemos que desafortunadamente durante el proceso otro programa que corre en la misma computadora entra en conflicto con el programa de control y monitoreo, o simplemente que por algún motivo ajeno a la voluntad del operador, la computadora se trunca en su funcionamiento y de nuevo, el proceso se interrumpe.

El segundo candidato es representado por los controladores lógicos programables [5] (abreviados como PLC, del inglés *programmable logic controller*), y al igual que con las tarjetas de adquisición de datos, son varias las compañías que los fabrican, algunos contienen en sí una unidad de adquisición de datos. En lo que al suministro de energía concierne, típicamente ocupan mucho menos que una computadora, haciendo que un UPS pueda suplir de energía al sistema de control durante más tiempo. La desventaja es que son sistemas cerrados, es decir, los programas que se implementen con ellos están limitados por sus características y difícilmente se encontrará alguno que disponga de la capacidad de realizar gráficas y reportes.

Hay un tercer candidato, el microcontrolador (MC). También es un sistema cerrado con características inferiores a los PLCs, pero que son la base de sistemas más complejos. La gran ventaja de los microcontroladores es que pueden recrear un sistema de control económico y la situación del consumo de energía es idéntica a las de un PLC.

Desde esta perspectiva parecería que no es posible cubrir los objetivos del proyecto, pero no es así: ¿por qué no hacer una combinación? Es decir, si una PC es tan susceptible de fallar por las causas señaladas pero es mucho más poderosa que un PLC o un microcontrolador, cualquiera de éstos dos últimos podría funcionar como control alternativo.

Se puede visualizar mejor este punto con un ejemplo hipotético. La implementación original del control es por PC, en ella se elabora un programa tal que permite al usuario interactuar de la manera más sencilla (obviamente, el usuario debe poseer los conocimientos básicos sobre operación de computadoras). La PC requiere de una tarjeta de adquisición de datos para obtener la información que los sensores proporcionan. Una primera alternativa es comprar esa tarjeta, pero la mayoría de las que se encuentra en el mercado dependen de la PC y si ésta falla, la tarjeta también. Es ahora cuando un PLC que posea una unidad de adquisición de datos entraría en escena, pues tanto puede tomar datos analógicos como implementar la misma estructura de programa que la PC y tomar el control cuando la computadora falle. El punto malo es un PLC tiende a ser caro y el aspecto económico del proyecto se vería afectado. Entonces nos queda la alternativa del microcontrolador. De hecho, es la que satisface tanto el control alternativo del sistema como el aspecto económico del proyecto. En efecto, con un microcontrolador es posible implementar el programa de control, la unidad de adquisición de datos – esto es, añadir un convertidor analógico a digital con un número determinado de canales analógicos– y seguir siendo una opción económica. Evidentemente no podría realizar una gráfica ni elaborar reportes (al menos con la relativa sencillez de una PC), para ello está la computadora.

La figura 3.2 muestra la etapa de control y monitoreo.

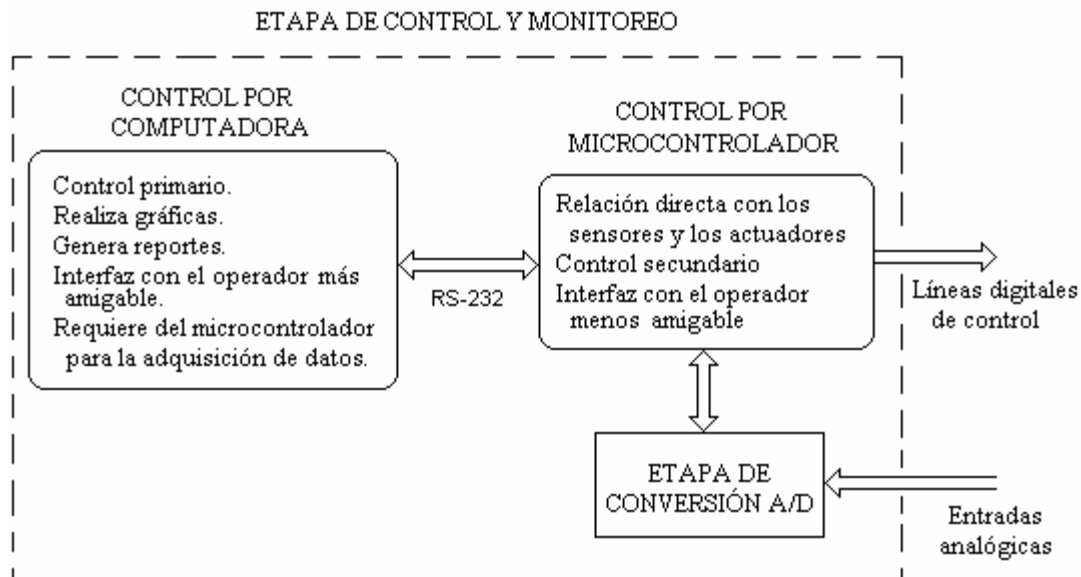


Figura 3.2. Etapa de control y monitoreo.

Es muy clara la forma en como trabajaría. La PC es el controlador primario, pero requiere de una tarjeta de adquisición de datos, así como de un protocolo de comunicación con ella. Esta tarjeta está basada en un microcontrolador y es más que una tarjeta, también es un control alternativo para el proceso cuando la computadora falle. En la figura 3.2 la etapa de conversión analógica a

digital se muestra aparte del control por microcontrolador, pero físicamente se encontrarán en la misma tarjeta.

3.3.1 Especificación para el control por microcontrolador

Ya está decidido que el control secundario será por microcontrolador (MC), ahora la pregunta es ¿cual en específico? No olvidar que se requiere que sea económico, porque también hay microcontroladores costosos y también que tenga los elementos adecuados para llevar la lógica del proceso de quema y que permitan la expansión del sistema.

Resumiendo, esta parte del control tiene las siguientes tareas:

- Dos modos de operación. El primero será como un simple esclavo de la computadora, todo lo que le ordene la PC será ejecutado por el MC, en este caso se dice que el microcontrolador fungirá como unidad de adquisición de datos. El segundo modo será como control alternativo, cuando falle la computadora el microcontrolador entrará en relevo, obviamente éste lleva el seguimiento del proceso desde el inicio aunque no tenga el control. Ambos modos de operación estarán contenidos en una memoria de programa.
- Sin importar el modo de operación, siempre manipulará al convertidor analógico a digital.
- Cuando esté en el modo de control alternativo, los datos de la quema serán guardados en un elemento conocido como memoria de datos, para que después sean descargados en la PC.
- Siguiendo en el modo de control alternativo, también debe ofrecer una interfaz al usuario lo más amigable posible o bien, que sea suficiente para que pueda operar al sistema, los elementos más simples son un teclado matricial y una pantalla de cristal líquido.
- Y por supuesto, contar con una interfaz que le permita comunicarse con la PC, optando por una serial debido a la facilidad para implementarla en un MC (muchos modelos de MCs cuentan con al menos un puerto serial). Debido a que el nivel que manejan los puertos seriales de los MCs son TTL, hay que adaptarlo al estándar RS-232 [6].

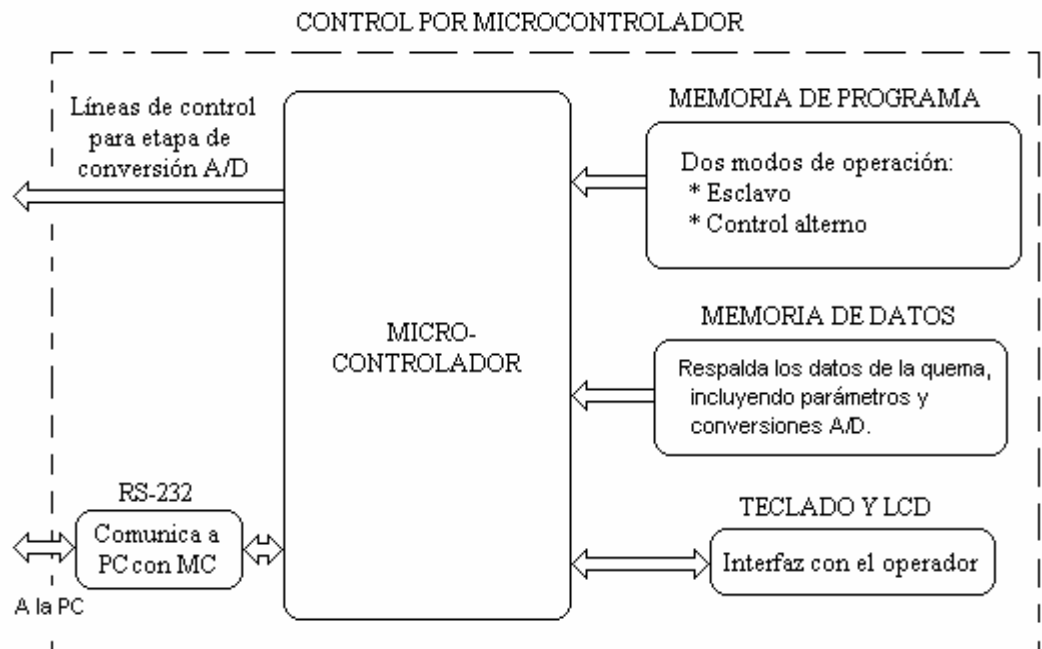


Figura 3.3. Especificación del control por microcontrolador.

3.3.2 Especificación para el control por computadora

Conceptualmente es más sencillo el control por PC. Basta con tener conocimiento de programación acerca del lenguaje que sea más apropiado para el desarrollo del sistema. Dicho lenguaje debe contar con las herramientas adecuadas para implementar la lógica del proceso, comunicar a la PC con el MC, generar los reportes y gráficas de las quemadas, ofrecer una interfaz amigable al usuario y que no interfiera con otros programas que estén activos en la computadora. En la figura 3.4 se muestra la especificación.

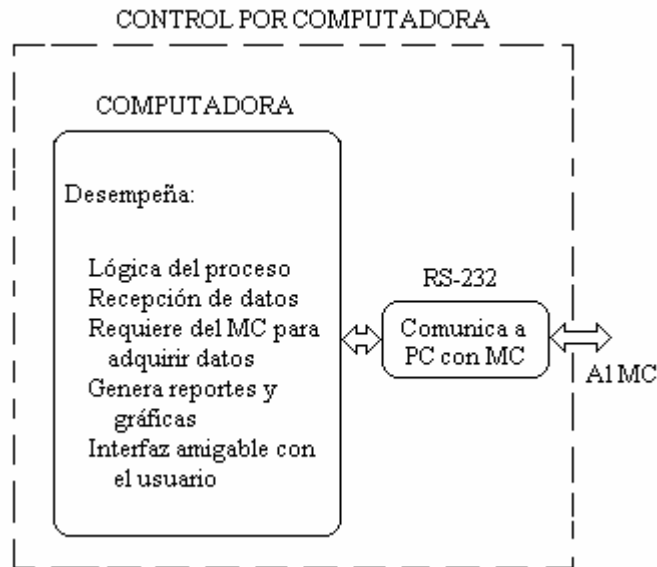


Figura 3.4. Especificación del control por computadora.

Sin embargo, realizar un programa implica hacerlo con una interfaz amigable para el usuario (operador de la quema en este caso), pues no hay que olvidar que si bien no es necesario que éste sea un experto en informática, al menos debe saber lo básico en manejo de computadoras personales y el sistema operativo que tenga cargado.

Para lograr esto existen normas para desarrollar interfaces, tal como la Interfaz Humano-Computadora o HCI [URL 7] (del inglés *Human-Computer Interface*). Esta norma tiene diez reglas fundamentales.

1- *Visibilidad del estado del sistema.* El sistema debe siempre mantener informado a los usuarios de lo que ocurre, con una correcta retroalimentación en un tiempo razonable.

2.- *Correspondencia entre el sistema y el mundo real.* El sistema debe hablar el lenguaje de los usuarios, con palabras, frases y conceptos familiares para el usuario. Esto equivale a seguir las convenciones del mundo real, haciendo que la información aparezca en forma natural y lógica.

3.- *Control y libertad del usuario.* Los usuarios frecuentemente eligen opciones por error y se necesita claramente indicar una salida para esas situaciones no deseadas sin necesidad de pasar por extensos diálogos.

4.- *Consistencia y estándares.* Los usuarios no tienen que adivinar que las diferentes palabras, situaciones o acciones significan lo mismo, por ejemplo, si por convención al hecho de duplicar algún texto se le llama “pegar”, no reemplazarlo por “unir”, aunque signifiquen lo mismo en determinados contextos.

5.- *Evitar errores.* Un diseño cuidado de tal manera que prevenga problemas en su utilización es mejor que añadir mensajes de error.

6.- *Reconocimiento versus recuerdo.* Se refiere a hacer objetos, acciones y opciones visibles. El usuario no tiene que recordar información de una parte a otra. Las instrucciones de uso del sistema deben estar visibles o fácilmente recuperables.

7.- *Flexibles y eficientes.* Diseña un sistema que pueda ser utilizado por un rango amplio de usuarios. Esto es, brindar instrucciones cuando sean necesarias para nuevos usuarios pero que no se entrometa en el camino de usuarios avanzados, permitiéndoles a éstos últimos ir directamente al contenido que buscan.

8.- *Diseño minimalista.* No mostrar información que no sea relevante. Cada pedazo de información extra compite con la información importante y disminuye su relativa visibilidad.

9.- *Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores.* Los mensajes de error deben estar escritos en lenguaje sencillo y asimilable, indicar precisamente el problema e indicar constructivamente una solución.

10.- *Ayuda y documentación.* El mejor sistema es el que se puede usar sin documentación, pero siempre es bienvenida una ayuda o documentación. Esta información debe ser fácilmente encontrada, dirigida a las tareas de los usuarios, donde se listen los pasos concretos para hacer algo y sobre todo, no ser muy extensa. Se recomienda también que sea ilustrativa, es decir, aparte de contener texto es recomendable contener también imágenes, siguiendo el conocido apotegma de que “una imagen dice más que mil palabras”.

Todos estos detalles deben ser tomados en cuenta para lograr un sistema eficiente, usable y funcional. Si bien este proyecto no tiene fines comerciales, al menos debe cubrir en lo posible estas diez reglas.

3.4 Propuesta para la etapa de sensores y actuadores

Por último queda la etapa de sensores y actuadores. Aquí están contenidos el sensor de temperatura y la electroválvula que permite el flujo del gas.

El sensor de temperatura (llamado también *termómetro*) adecuado es el basado en termopar. Con base en el rango de temperatura del horno (0 °C a 1250 °C) el tipo K es el preferido, pues cubre dicho rango y su comportamiento, aunque no es lineal, puede ser corregido por un circuito que adapte la señal a la parte de conversión analógica a digital de la etapa de control y monitoreo.

La pregunta es: ¿cuántos termopares hay que incluir? En el capítulo 2 se habló del proceso de quema existente en el taller de cerámica. Ahí se mencionó que entre más termopares haya en el horno, más precisa será la lectura de la temperatura. Pero en términos económicos, la inclusión de más termopares y sus respectivos circuitos aumentaría el costo, además de que realizar un promedio de las lecturas se haría más complejo. Por ello de momento se empleará únicamente un termopar, que no será el que ya contiene el horno sino uno adicional (la razón de ello es para conservar el sistema de medición original del horno intacto). Otro punto es la ubicación de este nuevo termopar. En la figura 3.5 se muestra una ubicación más adecuada para el termopar, que es en sustitución de la mirilla superior de la puerta del horno, necesitando de fijarlo con un pedazo de masa de arcilla. La razón de la colocación del termopar es que típicamente las piezas

se conglomeran en la parte media del horno, difícilmente se colocan a los costados dado el espacio que se tiene que respetar para la maniobra del carro.

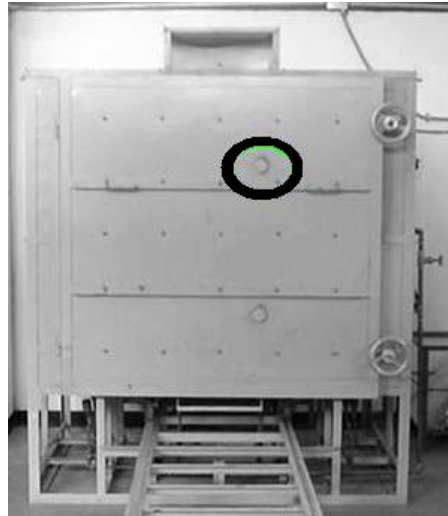


Figura 3.5. Ubicación sugerida del termopar.

En la sección 1.3.2 se dijo que es deseable registrar la presión del gas, para ello se requiere de sensores de ese tipo, uno para la válvula de los quemadores centrales y otro para la válvula de los quemadores laterales. El único requisito es que soporten el rango de presión de 0 psi a 15 psi, que es el límite de la tubería del horno. Pero de momento no se incluirán por dos motivos: tiempo de desarrollo y costo. En futuros proyectos se habrán de incluir.

Y respecto a los actuadores, se sabe que las opciones son las servoválvulas o las electroválvulas (o en su defecto, válvulas neumáticas). El inconveniente es que las servoválvulas son muy caras, por ello, este proyecto prescindirá de ellas y se remitirá a permitir el flujo de gas desde el contenedor principal, cubriendo de esa forma la necesidad del taller de interrumpir la quema cuando algún incidente (que lo amerite) se presente. Esa tarea se cumple mediante una electroválvula, ésta funciona ya sea con corriente alterna o con corriente directa y lo indicado es que sea del tipo normalmente cerrado aunque eso signifique que deba tener una batería de respaldo para cualquier corte de energía que se pueda presentar.

La interacción que esta etapa tendrá con la de control y monitoreo se da a través de las líneas de entrada analógica, provenientes del circuito del termopar; y de las líneas de control digital que van hacia el circuito de la electroválvula.

En la figura 3.6 se muestra la propuesta para esta etapa. Ahí se aprecia que las señales y líneas de control van a su bus correspondiente, la razón estriba en que si en un futuro llegan a añadirse más elementos a esta etapa, las señales y líneas de control de ellos irán a sus buses respectivos.

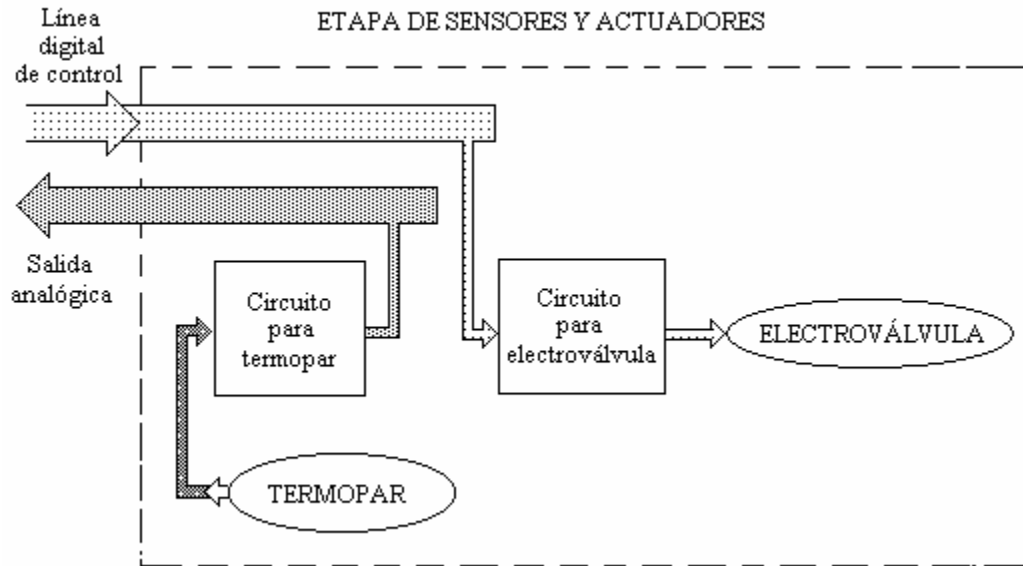


Figura 3.6. Propuesta para etapa de sensores y actuadores.

3.5 Propuesta para respaldo de energía en el sistema

Todo lo que se ha dicho acerca del respaldo de energía se ha remitido a la inclusión de un sistema que lo haga. Este es conocido como *no-break* que literalmente se traduce “sin corte” de energía y también como UPS (del inglés *uninterruptible power system* o sistema de poder sin interrupción). Este sistema provee de corriente alterna a una carga en específico.

Si todos los elementos del sistema fuesen respaldados con el UPS, el soporte de éste sería muy corto, especialmente porque la PC consume mucha energía y porque la electroválvula estará activa durante todo el proceso. Si privamos a la computadora del respaldo no habría tanto problema para la continuidad del proceso, pues entraría el microcontrolador como control secundario y los datos se siguen guardando. Además, la electroválvula consume menos energía que la computadora y por tanto el UPS podrá soportar toda la carga durante más tiempo.

Formalmente, la propuesta es que el respaldo de energía contemple a todo el sistema excepto la computadora. En la figura 3.7 se muestra el esquema correspondiente al respaldo de energía.

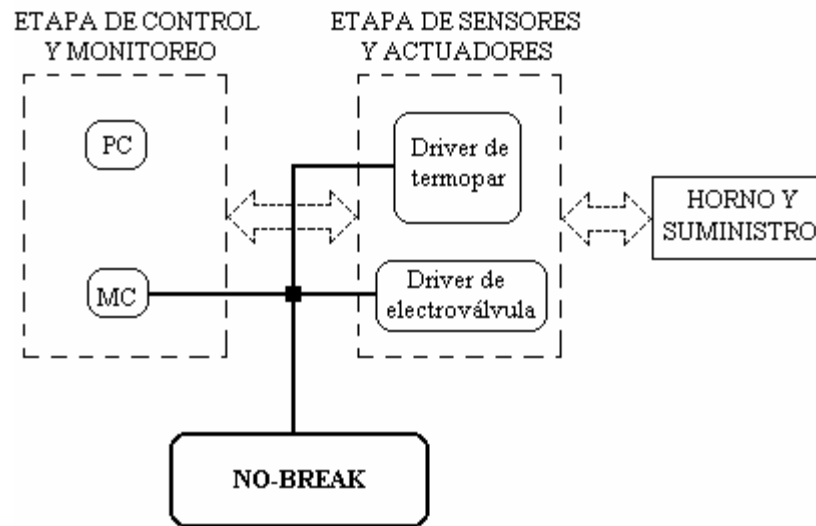


Figura 3.7. Respaldo de energía en el sistema.

Capítulo 4 - CONSTRUCCIÓN DEL SISTEMA

En el capítulo anterior se mencionó la clase de dispositivos se habían de emplear para realizar el proyecto. En éste se definen exactamente los modelos que habrán de emplearse y cómo se relacionan entre sí.

4.1 Construcción de la etapa de control y monitoreo

Es pertinente recordar las características de esta etapa (ver 3.3):

- Redundancia en el control. El control primario será a través de una computadora, la cual será relevada por un microcontrolador cuando ésta falle.
- Tanto la computadora como el microcontrolador deben tener una interfaz de comunicación.
- La computadora es la que ofrecerá una interfaz amigable con el usuario, sólo ella tiene la facultad de realizar gráficas y reportes. La interfaz del microcontrolador será mucho más limitada.
- El microcontrolador constituye la unidad de adquisición de datos y el respaldo de los datos generados durante la quema.
- Es importante que el consumo de energía sea el menor posible, para que el respaldo de energía sea mayor en tiempo.
- El algoritmo a implementar en ambos dispositivos es el mismo y está descrito en el capítulo 2. De él se desprende que al inicio de la operación es necesario que el operador provea ciertos datos de entrada o parámetros de la quema. El principal e indispensable es la temperatura objetivo a la cual el sistema ha de terminar de operar. Otros parámetros son la hora de inicio, fecha de la quema y el periodo de almacenamiento de datos.

4.1.1 Construcción del control por microcontrolador

En nuestro país está disponible una gran variedad de microcontroladores que pueden cubrir por completo el algoritmo y también satisfacer la característica de ampliación del proyecto. Como ejemplos están las familias 68HC11 de Motorola, MCS-51 [URL 8] de Intel, AVR de Atmel, PIC de Microchip, etc.

Atendiendo a los aspectos establecidos en la sección 3.3.2 y al algoritmo del proceso (ver capítulo 2), las características deseadas en el microcontrolador son:

- Una memoria de programa suficientemente grande para implementar el algoritmo y sus futuras ampliaciones, 32 KBytes es más que suficiente para implementar el algoritmo de la quema. Por limitaciones en cuanto a tecnología de fabricación de circuitos impresos, el encapsulado del MC debe ser DIP (*dual in-line package* o encapsulado de línea dual) o en su defecto, PLCC (*plastic leaded chip carrier* o portador de chip de terminales plásticas).
- Contar con memoria de acceso aleatorio (RAM o *random access memory*) para el manejo de las variables del programa a desarrollarse y de la pila. Típicamente los MCs en sus versiones de encapsulado DIP incluyen memorias de datos de entre 128 hasta 2 KBytes, suficientes para la declaración de variables. En lo que concierne al respaldo de los datos generados durante el proceso, no es aconsejable usar RAM para almacenarlos, dado que al interrumpir la energía eléctrica, tales datos se pierden. Lo recomendable es que el MC cuente con líneas de acceso a memoria externa.
- Una interfaz de comunicación con la computadora. La mayoría de las basadas en la arquitectura IBM y compatibles incluyen un puerto de comunicación serial RS-232 que soporta el protocolo de transmisión-recepción asíncrono universal [URL 9] (UART, por sus siglas en inglés). Este mismo protocolo se encuentra en algunos microcontroladores permitiendo así la comunicación de los dispositivos. Otra opción es el puerto paralelo de la PC, con la desventaja de que ocuparía más líneas del MC y por propósitos de expansión del sistema, esas líneas podrían ser necesarias para ello.
- Una interfaz de comunicación con el operador, pues si el proceso llega a ser controlado por el MC, requerirá de comunicarse con el usuario. El medio más efectivo es un teclado, para introducción de los datos y una pantalla de cristal líquido (*liquid crystal display* o *LCD*) para mostrarlos. Es por ello que se necesitará de otras líneas para manejar estos dispositivos.
- Dado que el algoritmo del proceso se basa en manejar tiempos, se requiere de temporizadores internos programables. La longitud de los temporizadores depende del periodo-máquina del MC dado que según el algoritmo, cada treinta minutos se incrementa la presión de las válvulas. Por ejemplo, si el periodo-máquina fuese de un microsegundo, se requeriría de un temporizador de 32 bits para cubrir los treinta minutos.
- Respecto a la adquisición de datos, lo más cómodo es que tenga un convertidor analógico a digital (*analog to digital converter* o ADC) integrado, en caso contrario habrá que encontrar uno que sea compatible con el MC. La resolución del ADC depende del rango de temperatura del horno. Si dicho rango es de 1250 °C, un convertidor de 8 bits daría una resolución de aproximadamente 5 °C por LSB, lo cual no es deseable porque una diferencia de esa magnitud puede influir en la etapa de reposo de esmalte (ver capítulo 2). Con un ADC de 10 bits la resolución sería de 1.22 °C que si bien es mejor que el anterior, es deseable tener una menor resolución. Lo mejor es un ADC de 12 bits, que ofrece una resolución de aproximadamente 0.3 °C.
- Disponibilidad de herramientas de desarrollo. Cada familia de microcontroladores debe de contar con sus compiladores y manuales respectivos, que son provistos por los

mismos fabricantes y la mayoría de las veces son gratuitos pero solo en su versión de lenguaje ensamblador. Otras compañías se dedican a desarrollar compiladores en lenguaje de nivel más alto (C, Basic, Pascal), con su correspondiente costo por concepto de compra.

- Debe ser económico pero cumpliendo en la mayor medida los puntos anteriores.
- Debe conseguirse fácilmente con distribuidores locales y nacionales, pues una importación incrementa el costo.

En la tabla 4.1 se muestra la comparación entre cuatro candidatos: 68HC11E1CFN2 de Motorola, P80C32SBPN de Phillips (con licencia de Intel), ATmega8515L (antes AT90S8515) de Atmel y PIC16F877 de Microchip.

Tabla 4.1. Comparación entre varios microcontroladores.

	68HC11E1CFN2	P80C32SBPN	ATmega8515L	PIC16F877
Memoria de programa interna (EPROM, FLASH) [bytes]	-	-	8 K	8 K
Memoria de datos interna (RAM) [bytes]	512	256	512	368
Memoria EEPROM interna [bytes]	2048	-	512	256
Direccionamiento a memoria externa [bytes]	Hasta 64 K en paralelo (programa y datos)	Hasta 64 K en paralelo (programa y datos)	Hasta 64 K en paralelo (datos)	-
Puerto de comunicación serial	1 x UART	1 x UART	1 x USART	1 x USART
Número de líneas digitales de entrada/salida	38	32	35	33
Temporizadores/Contadores internos programables (incluyendo pines de captura/comparación)	4 x 16 bits	3 x 16 bits	1 x 16 bits 1 x 8 bits	1 x 16 bits 2 x 8 bits
Convertidor A/D integrado	8 bits (8 canales)	-	-	10 bits (8 canales)
Interrupciones externas	2	2	3	1
Otros periféricos	1 x SPI 1 x WDT	-	3 x PWM 1 x SPI 1 x WDT 1 x Comparador analógico	2 x PWM 1 x SPI 1 x I ² C 1 x PSP
Ejemplos de compiladores y/o herramientas de desarrollo libres	Buffalo (ensamblador) AS11ASM (ensamblador)	ASM51 (ensamblador) Small Device C (lenguaje C)	AvrStudio4	MPLAB IDE
Costo* (m.n.)	121.99	32.93	72.00	96.99
Disponibilidad**	Nacional	Nacional	Nacional	Nacional

(*) Cotización al inicio del segundo trimestre del 2006.

(**) Distribuidor: AG Electrónica S.A. de C.V. (México, D.F.)

Nota: Todos los modelos tienen en su encapsulado la opción DIP y PLCC.

Elección del microcontrolador

Todas las familias mencionadas anteriormente y los modelos que la componen cubren la mayoría de estos requisitos, en mayor o menor medida unas respecto a las otras. Ante esta situación, los tres últimos criterios fueron decisivos para la elección.

Se optó por el microcontrolador P80C32SBPN o simplemente 80C32 [7] (ese chip es producido por Phillips con licencia de Intel). Es fácil de encontrar en México, puede acceder a memoria externa tanto de programa como de datos para lectura y escritura, su longitud de dato es de 8 bits, cuenta con tres temporizadores programables, contiene un módulo UART, etc. Un aspecto importante es que el 80C32 no tiene un ADC integrado, entonces hay que buscar uno adecuado para el proyecto.

Elección de dispositivos periféricos (basado en los detalles a considerar).

- Al momento de activar la energía, el 80C32 debe sufrir un pulso de reset para iniciar su operación. Ese pulso debe durar al menos 1 milisegundo. Un simple circuito RC compuesto por un resistor de 100 ohmios con un condensador de 10 microfaradios basta para ello.
- El 80C32 no tiene memoria de programa interna. Eso obliga a agregar dispositivos externos. Una EPROM (*erasable/programmable read only memory* o memoria de sólo lectura programable/borrable) de 32 Kbytes es adecuada pensando que el proyecto debe ser susceptible a expandirse. Esto también lleva a demultiplexar el puerto de datos/direcciones del MC. Es necesario agregar un latch, el 74LS573 es suficiente.
- La RAM del 80C32 es muy pequeña, dispone de 256 bytes que alcanzan para declarar variables y atender a la pila del programa. Si el objetivo es almacenar datos de la quema es mejor usar una EEPROM (*electrically erasable/programmable read only memory* o memoria de sólo lectura programable/eléctricamente borrable) externa. La ventaja de la EEPROM es que retiene los datos aún cuando la energía se retire del sistema. La desventaja de la EEPROM es el número límite de ciclos de escritura, dado que una vez pasado dicho límite el contenido del chip ya no puede ser modificado. Se entiende por *ciclo de escritura* cuando se han escrito todas las celdas del chip. Para determinar el tamaño de la memoria hay que pensar en cuatro variables: la tasa de muestreo o almacenamiento de datos, el número de sensores, la extensión de los datos y la duración de la quema. Para empezar, cada dato es de 12 bits pero dado que la mayoría de las EEPROM tiene una longitud de dato de 8 bits, es mejor emplear dos locaciones continuas para dar cabida al dato y todavía sobrarían 4 bits; luego, acorde al algoritmo descrito en el capítulo 2, no es necesario almacenar datos a cada momento, de hecho la máxima tasa de muestreo sería de una muestra cada minuto; por otra parte, de momento nada más hay un sensor, pero pensando en las futuras expansiones del sistema, un máximo de ocho sensores sería suficiente; finalmente, la quema más rápida que se ha realizado en el horno ha sido de siete horas. Considerando el peor de los casos, es decir, contar con 8 sensores y tener una tasa de almacenamiento de 1 muestra cada minuto y la duración de la quema de 7 horas, se requerirían de 6720 bytes para respaldar la quema. Una EEPROM de 8 Kbytes es suficiente, incluso alcanza para almacenar los parámetros de la quema. El modelo elegido fue el AT28C64 [8] de Atmel con base en su compatibilidad con el 80C32. La interfaz es paralela y puede ocuparse el puerto de dirección y datos del MC para

acceder a los datos contenidos en la EEPROM. En cuanto al almacenaje de los datos, basta con que el MC haga una rutina de escritura externa y después espere un milisegundo para que la EEPROM tenga tiempo de guardar el dato. En lo que al número de ciclos de escritura respecta en el AT28C64, éste es de cien mil iteraciones y considerando que el número de quemas es pequeño (ver capítulo 1, sección 1.3), un chip alcanza para varios años de operación.

- En lo que a la interfaz MC–Operador concierne, si se ocupa un teclado matricial habrá que incluir un decodificador adecuado; el circuito integrado LM74C922 es una buena opción. En el caso del LCD, existen módulos que facilitan su uso, el AND491 [9] (16 columnas x 2 filas) y compatibles también es una buena opción.
- Debido a que sólo se dispone de dos puertos más en el 80C32, las líneas de éstos serían insuficientes para controlar más dispositivos. Se requiere de hacer mapeo en memoria para esos dispositivos tales como teclado, LCD y el mismo ADC. Hay que incluir un decodificador de líneas de control. La propuesta es usar un dispositivo lógico programable (PLD, por sus siglas en inglés). Su ventaja es que puede modificarse acorde a las necesidades que vayan surgiendo al momento de expandir el proyecto. El dispositivo elegido fue la GAL22V10 [10] de Lattice.
- El 80C32 tiene una interfaz UART que le permitirá comunicarse serialmente con la PC. Para adaptar la interfaz al RS232 de la PC se recurre al integrado MAX232 [11].
- La comunicación serial lleva al concepto de *baud rate* (tasa de baudios o tasa de transferencia de bytes por segundo) y éste a su vez al reloj de trabajo del MC. Con un oscilador basado en cristal de 12 MHz se puede trabajar a velocidades casi exactas de 2400 y 4800 baudios por segundo (bps). Además, la temporización del algoritmo y de la EEPROM previamente seleccionada son sencillas gracias a que con ese cristal la velocidad de bus es de 1 microsegundo.
- La búsqueda de un ADC (de 12 bits) es tarea fácil y difícil a la vez. Fácil porque hay un sinnúmero de ejemplares, difícil porque no todos esos ejemplares se encuentran con facilidad en México y porque hay que cuidar que ese dispositivo no se discontinúe a corto plazo (en su producción). Después de analizar varias opciones se optó por el ADS774 [12] de Texas Instruments. La principal ventaja que ofrece es una total compatibilidad con el 80C32 debido a que sus diagramas de tiempo para conversión y lectura de datos son iguales, pero requiere estar mapeado en memoria. Su desventaja es que cuenta con un canal analógico, misma que se soluciona agregando un multiplexor analógico. El CD4051 de Motorola es adecuado para este caso, que es un multiplexor analógico de 8-1 canales. El inconveniente de este multiplexor es que sus líneas digitales de selección de canal no soportan la velocidad de las líneas de direccionamiento del 80C32. Una opción sería disponer de tres líneas del puerto 1 (no consideramos al puerto 3 dado que las líneas de comunicación serial, lectura y escritura externa e interrupción externa ya están ocupadas), pero hay que recordar que para propósitos de expansión del proyecto podría ser necesario tener íntegro dicho puerto. En vista de que se requieren de tres líneas para multiplexar los 8 canales, una buena opción es usar un flip-flop de 4 bits (como el 74173) también mapeado en memoria para que guarde la combinación de las líneas de selección del CD4051.

- La conversión analógico a digital implica resolver un problema muy común: el ruido eléctrico debido a desacoplo de tierras. Una técnica para eliminar ese problema es el diseño de un plano especial de tierra analógica [13] en el circuito impreso, mismo que es complicado de hacer. Pero también es posible incluir un transformador de acoplo entre la tierra analógica y la tierra digital, en lo único que repercute es en el tamaño del circuito impreso.
- Retomando el aspecto de expansión del proyecto, es posible que se emplee el mapeo en memoria para otros dispositivos. El punto es que tendrían que utilizar el puerto 0 para transferir datos y como es bien conocido, se corre el riesgo de atenuar la señal del puerto. Entonces es pertinente reforzarlo mediante búferes. El integrado 74HC245 es una opción viable.
- Por último, si se desea que el circuito completo consuma la menor cantidad posible de potencia, es pertinente que cada uno de los dispositivos que lo compongan sean de tecnología CMOS o incluso CHMOS, desde el microcontrolador hasta los flip-flops.

Finalmente se muestra el diagrama de conexiones del control por MC.

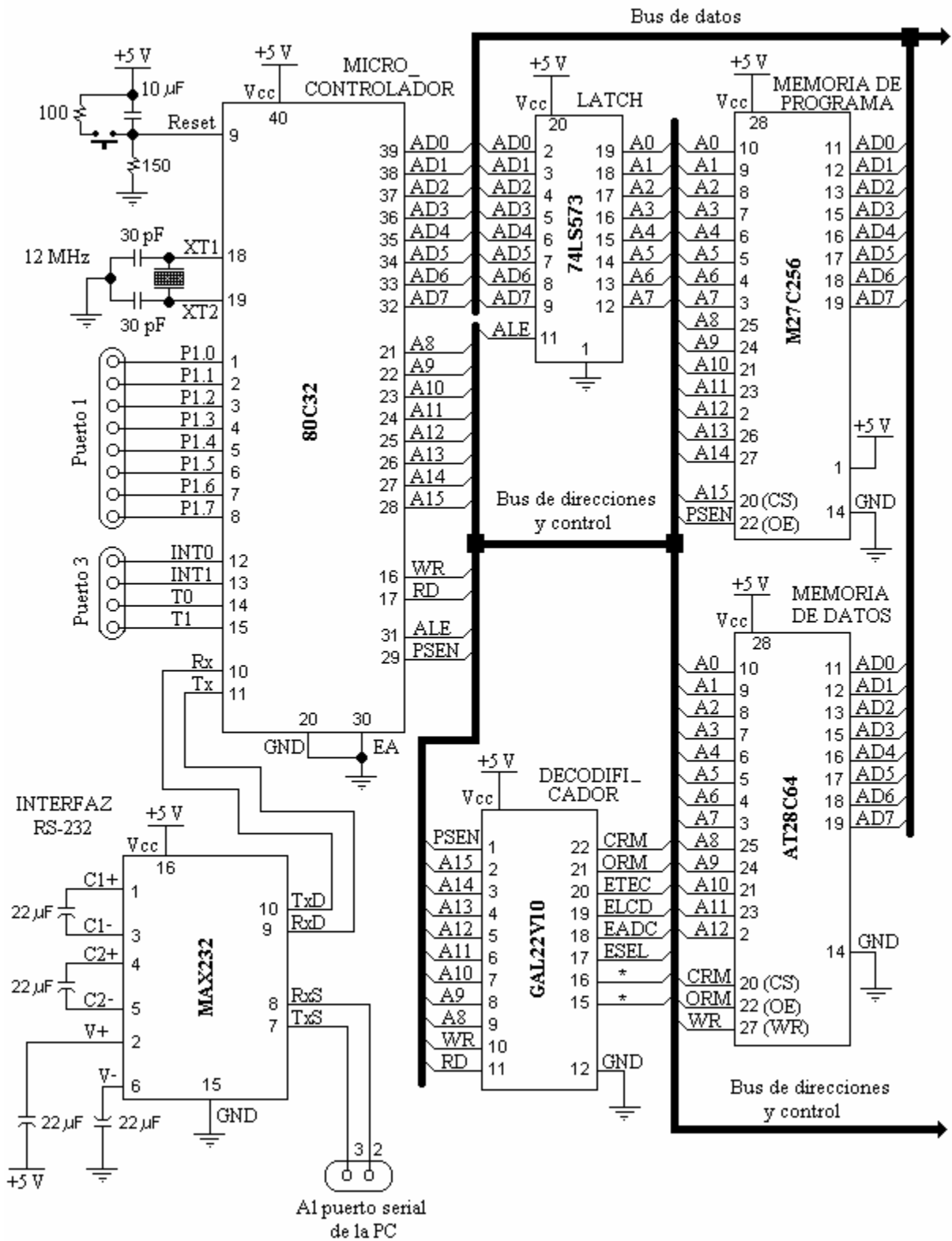


Figura 4.1. Diagrama de conexiones del control por MC.

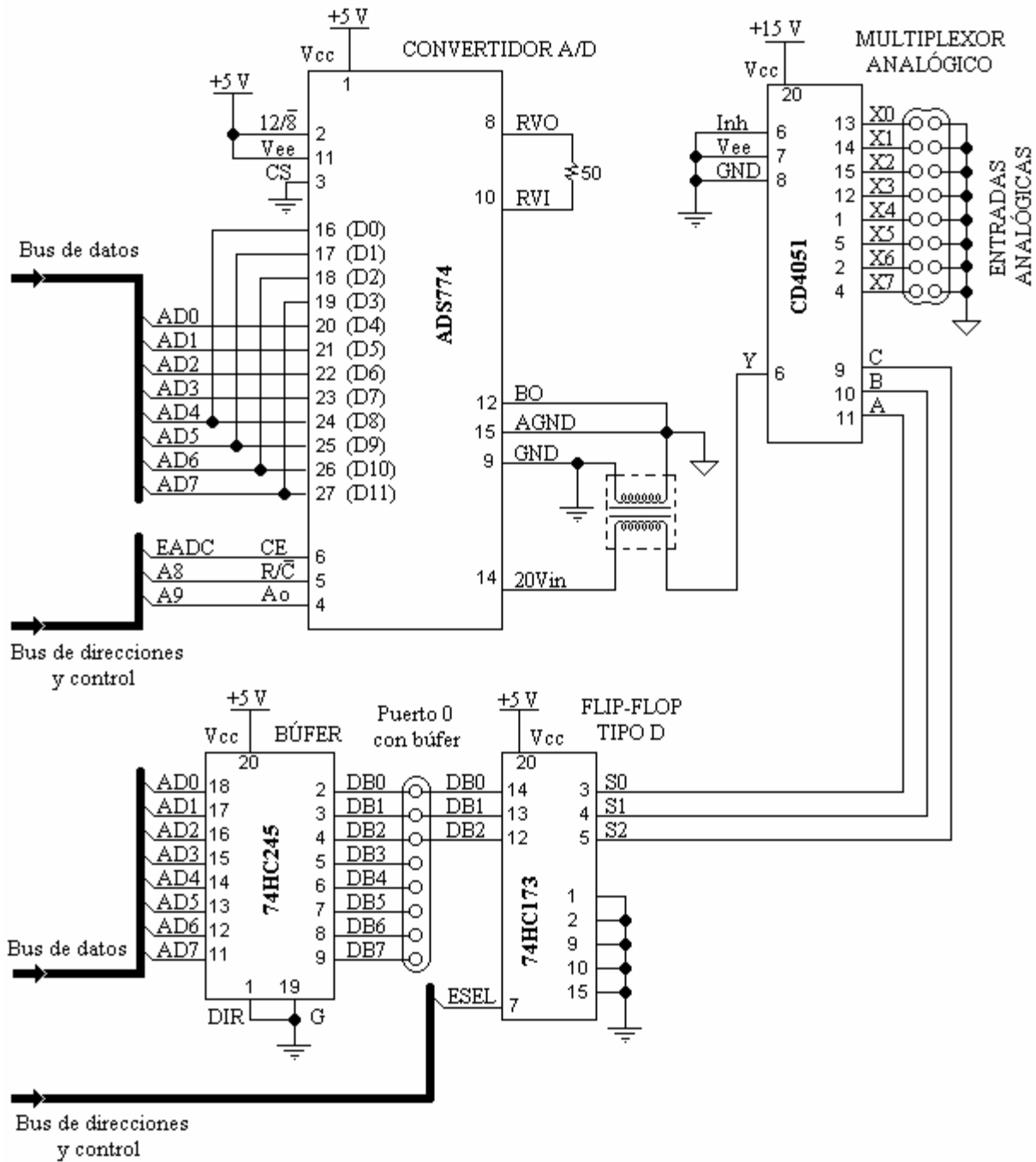


Figura 4.1. Diagrama de conexiones del control por MC (continuación)

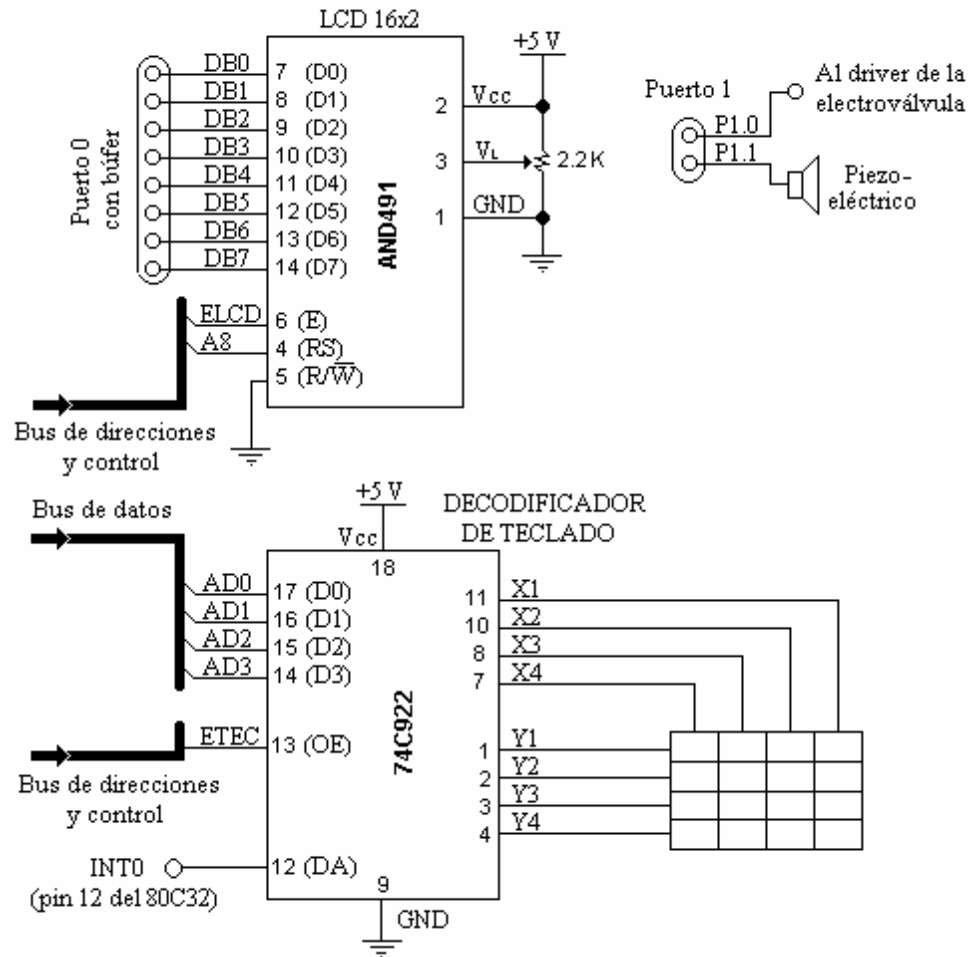


Figura 4.1. Diagrama de conexiones del control por MC (continuación)

De acuerdo con la figura anterior, las líneas de salida del dispositivo GAL22V10 fueron programadas de la siguiente manera:

$$\text{CRM} = \neg A15 \text{ (dirección: } 0x8000\text{)}$$

$$\text{ORM} = \text{PSEN} * \text{RD}$$

$$\text{EDEC} = \neg(\neg A15 * A14 * A13 * A12) + \text{RD} \text{ (dirección: } 0x7000\text{)}$$

$$\text{ELCD} = \neg A15 * A14 * A13 * \neg A12 * \neg \text{WR} \text{ (dirección: } 0x6000\text{)}$$

$$\text{EADC} = \neg A15 * A14 * \neg A13 * A12 * \neg \text{RD} \text{ (dirección: } 0x5000\text{)}$$

$$\text{ESEL} = \neg A15 * A14 * \neg A13 * \neg A12 * \neg \text{WR} \text{ (dirección: } 0x4000\text{)}$$

Cabe mencionar que los pines 15 y 16 del dispositivo GAL22V10 de momento no están programados, se incluyeron en el bus de direcciones y control por si llega a expandirse el sistema y se requiera de mapear más dispositivos en memoria. De esa manera se dispone de dos líneas para habilitar tales dispositivos.

Consideraciones en la programación del microcontrolador.

Existen varias herramientas de desarrollo para la familia MCS-51, algunos de licencia libre y otros más de licencia comercial, incluyen compiladores de lenguaje ensamblador de la familia y a veces compiladores de lenguajes de nivel más alto (obviamente para el microcontrolador en cuestión), como Basic, C y Pascal. La herramienta de desarrollo elegida para la programación del 80C32 fue el Proview32 versión 3.3 de Franklin Software Inc. [URL 10], éste incluye compilador en C, compilador en ensamblador, simulador, depurador y emulador (estos últimos sólo con tarjeta de desarrollo de la compañía). Respecto al costo que genera el uso de un compilador comercial, éste fue suprimido al ser facilitado por el asesor de la presente tesis.

De esta herramienta se optó por el compilador en C por una razón: facilita la estructuración del código fuente y esto hace más fácil de entender por programadores con experiencia en C, no obstante la gran popularidad con la que aún tiene dicho lenguaje, además cuenta con librerías de funciones complejas y emula operaciones aritméticas con variables de tipos distintos (enteros con y sin signo, flotantes y dobles). La desventaja es que aumenta el tamaño del código hexadecimal (el que se graba directamente en la memoria de programa del sistema) en comparación a que si se hubiera hecho el código fuente en lenguaje ensamblador y aumenta mucho más al usar las librerías, de ahí la previsión de dar una memoria de programa grande al sistema.

Vale la pena ahondar en este punto, un ejemplo sencillo dará claridad a la idea. El compilador cuenta con una librería llamada STDIO.H, del mismo nombre que el de los compiladores C/C++ para microprocesadores 8086 de Intel. Ahí está incluida la función *scanf*, que sirve para capturar datos desde puerto serial. Bien, debido a todas las opciones con que cuenta la función hace que al momento de compilarla ésta ocupe por sí sola alrededor de 1000 bytes en formato hexadecimal y si tan sólo se usó para capturar un valor del tipo *unsigned char* (caracter sin signo, longitud de 8 bits), significa que se derrochó espacio en la memoria de programa, de hecho, con un máximo de 5 líneas en ensamblador se obtendría el mismo resultado, las cuales al compilarse ocupan alrededor de 12 bytes. La diferencia es considerable, en la tabla 4.2 se muestra el ejemplo.

Tabla 4.2. Comparación entre códigos fuente: lenguaje C – ensamblador.

Lenguaje C	Ensamblador
Líneas de código fuente: 1 Tamaño en bytes después de la compilación: alrededor de 1000	Líneas de código fuente: 5 Tamaño en bytes después de la compilación: alrededor de 12
<code>scanf ("%c", caracter);</code>	Obtener_caracter: JNB RI, Obtener_caracter MOV A, SBUF CLR RI RET

Luego entonces, se prefiere no utilizar las librerías especiales del Proview32, sino hacer librerías propias.

Organización de la memoria en el microcontrolador.

La forma en como se va a disponer de los dispositivos de memoria está en función del direccionamiento que realiza el 80C32. Este microcontrolador puede direccionar hasta 64 KB de memoria externa, ya sea de programa, de datos o combinada.

Por diseño, en lo que respecta a la memoria de programa, el compilador es el que se encarga de distribuir todo el código hexadecimal que vaya generando el compilador, simplemente no debe exceder los límites establecidos en la figura 4.2, que va desde la locación 0x0000 hasta la 0x7FFF, equivalente a 32 Kbytes.

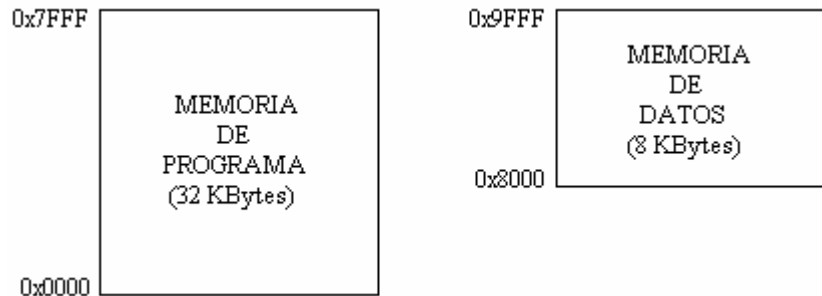


Figura 4.2. Organización de las memorias de programa y de datos.

La memoria de datos tiene a su vez una organización más compleja. Sus límites van desde la locación 0x8000 hasta la 0x9FFF, equivalentes a 8 Kbytes. Dado que la extensión de cada dato es de 2 bytes, se le referirá como *palabra*. El objetivo de guardar datos es para hacer un registro de la quema con toda la información necesaria.

Básicamente se subdivide en tres partes.

1. Parámetros de la quema. Se refiere a la temperatura objetivo, la hora de inicio, la fecha, etc. La extensión de esta parte es de 32 palabras.
2. Acciones varias. Cualquier acción de los actuadores durante el proceso será guardada en esta zona de 100 palabras de extensión. Esta servirá cuando en futuras expansiones del proyecto se incluyan servoválvulas u otra clase de actuadores. Si bien actualmente se cuenta con una electroválvula como actuador, sus acciones no se registran aquí porque son sólo dos: su activación al inicio de la quema y su desactivación al fin de la quema.
3. Zona de datos. Aquí se guardarán los datos recogidos por los sensores durante el proceso. Dado que la unidad de adquisición de datos soporta hasta 8 canales de entrada analógica, esta zona tiene cabida para todos ellos. Cada sensor tiene asignado un espacio de 450 palabras, mismo que a una tasa de 1 muestra por minuto respalda una quema con duración de siete horas y media.

Todavía queda un espacio libre de 364 palabras, quizás en un futuro se le dé un uso.

En la figura 4.3 se muestra la estructura de la memoria en el sistema.

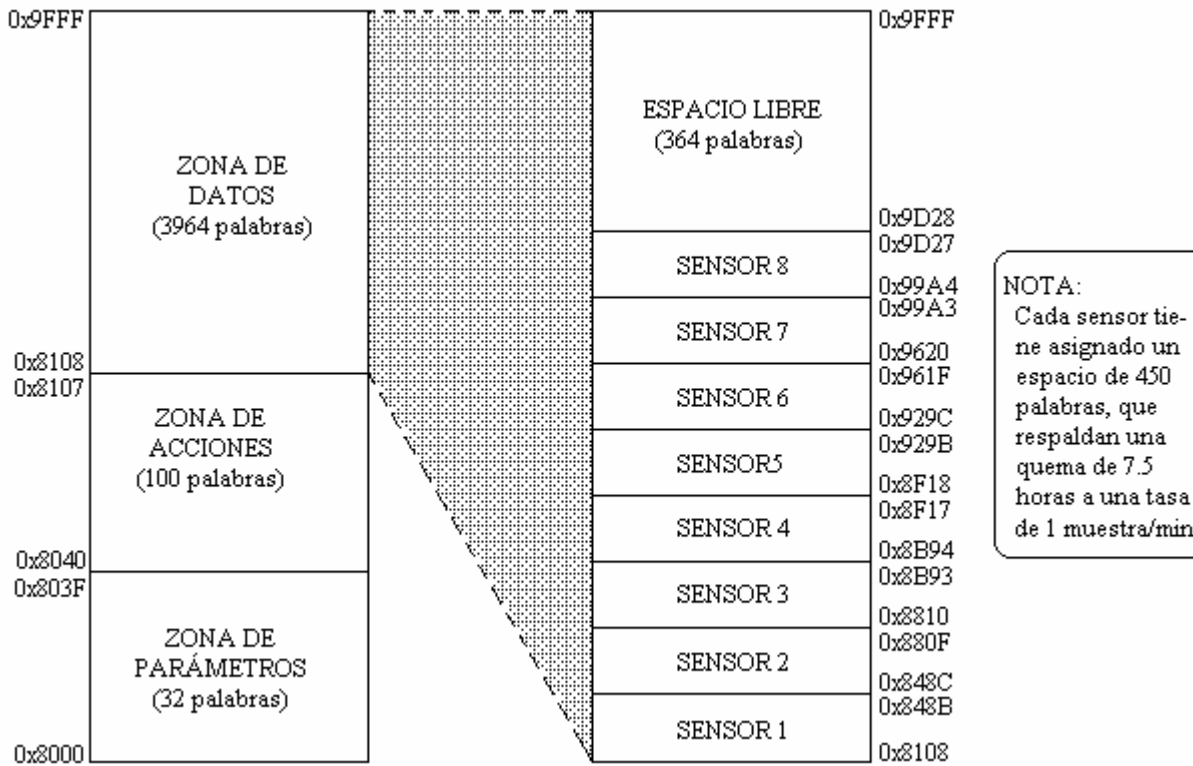


Figura 4.3. Organización en la memoria en el sistema.

De aquí se desprende otro detalle: el formato para almacenar los datos en cada una de las zonas de la memoria.

De manera general, el formato es el mostrado en la figura 4.4.

FORMATO GENERAL DE LOS DATOS EN LA MEMORIA DEL SISTEMA

D/A	X	X	X	X	X	X	X	X	H
	X	X	X	X	X	X	X	X	L

Si D/A = 0, la palabra indica un dato

0	S2	S1	S0	D11	D10	D9	D8	H	
	D7	D6	D5	D4	D3	D2	D1	D0	L

Donde: S2 - S0 = Identificador del sensor
D11 - D0 = Dato

Si D/A = 1, la palabra indica una acción

1	-	-	-	-	d2	d1	d0	H	
	-	-	A5	A4	A3	A2	A1	A0	L

Donde: d2 - d0 = Identificador del dispositivo
A5 - A0 = Acción u orden

Figura 4.4. Formato general en la memoria de datos del sistema.

El bit más significativo (el número 15) de la palabra sirve para indicar si el dato es propiamente una lectura desde algún dispositivo o una acción efectuada por un actuador. De esa manera, cuando el bit 15 es igual a cero, indica un dato y significa que está en la zona de datos

del sistema, por su parte, los bits 14 al 12 (S2 a S0) indican a qué dispositivo corresponde dicho dato, el los bits 11 a 0 (D11 a D0) son propiamente la lectura de 12 bits de resolución, efecto directo de que el convertidor analógico a digital es de dicha resolución. Pero si el bit 15 es igual a uno, indica una acción y significa que está en la zona de acciones del sistema, por su parte, los bits 10 a 8 (d2 a d0) indican qué actuador hizo la acción a especificar por los bits 5 a 0 (A5 a A0). Se hace recordar que de momento la zona de parámetros no tiene uso, se diseñó así para dar cabida a futuras expansiones.

Este formato general sirve también como la base de la comunicación entre PC y MC, esto es, representa el formato de la trama serial, que es de longitud de 2 bytes. Cuando la PC envía una orden al MC, el bit 15 es igual a 1 y ahora los bits 5 al 0 (A5 a A0) indican qué acción en específico se ha de realizar. Si la acción involucra a un dispositivo, los bits 10 al 8 (d2 a d0) especifican a cuál. Por ejemplo, si la orden es tomar la lectura de un termopar (el único que hay de momento), los bits A5 a A0 contienen la orden de tomar lectura y los bits d2 a d0 tienen el identificador del termopar.

Cuando la PC o el MC envíen un dato en específico, simplemente el bit 15 es igual a 0, los bits 11 a 0 (D11 a D0) contienen el valor correspondiente al sensor identificado por los bits 14 a 12 (S2 a S0).

Regresando al tema del formato en la memoria del sistema, el de la zona de parámetros varía con respecto al general, como se aprecia en la figura 4.5.

FORMATO EN LA ZONA DE PARÁMETROS DE LA MEMORIA DE DATOS										
TEMPERATURA	-	-	-	-	D11	D10	D9	D8	H	Donde: D11 - D0 = Valor del parámetro en 12 bits
	D7	D6	D5	D4	D3	D2	D1	D0	L	
HORA	hd3	hd2	hd1	hd0	hu3	hu2	hu1	hu0	H	Donde: hd3 - hd0 = Decenas de horas hu3 - hu0 = Unidades de horas md3 - md0 = Decenas de minutos mu3 - mu0 = Unidades de minutos
	md3	md2	md1	md0	mu3	mu2	mu1	mu0	L	
FECHA	a4	-	-	d4	d3	d2	d1	d0	H	Donde: d4 - d0 = Día del mes m3 - m0 = Mes a4 - a0 = Año en dos dígitos
	m3	m2	m1	m0	a3	a2	a1	a0	L	
TASA DE MUESTREO	s15	s14	s13	s12	s11	s10	s9	s8	H	Donde: s15 - s0 = Valor del parámetro en 16 bits
	s7	s6	s5	s4	s3	s2	s1	s0	L	

Figura 4.5. Formato de la zona de parámetros en la memoria del sistema.

El peculiar formato responde a la necesidad de adaptarlos a la extensión de la palabra empleada, es decir a 16 bits.

4.1.2 Algoritmo de quema en el microcontrolador

El algoritmo implementado se basa en el diagrama de flujo de la figura 2.10.

Consideraciones

La implementación neta del programa en el microcontrolador tuvo que contemplar varios aspectos:

- El MC tiene función doble, primero como esclavo y segundo como maestro.
- Mientras actúe como esclavo, puede economizar sus recursos, esto es, dado que la computadora es quien mide los tiempos y decide qué acción tomar, el MC puede funcionar manteniendo activo su puerto de comunicación serial, sin necesidad de activar los temporizadores (junto con sus respectivas interrupciones). A lo más se requeriría habilitar una interrupción externa para dar la posibilidad de abortar el proceso cuando sea necesario.
- Dado que la comunicación es serial, la forma de saber si se ha perdido la comunicación es medir el tiempo en que la computadora tarda en enviar información, sea ésta un comando o datos. Si el tiempo sobrepasa un límite determinado, hacer la conmutación de esclavo a maestro. Esta acción debe estar activa en todo momento que el microcontrolador espere información desde la computadora.
- En la sección 4.1.3 se verá que LabView (el software elegido para realizar el programa por PC) emplea el puerto serial para transferencia de bytes en formato BCD (*binary coded to decimal* o binario codificado a decimal). Entonces, es necesario que las funciones encargadas en el MC de transferir datos vía serial conviertan el formato BCD al hexadecimal y viceversa.
- Básicamente, mientras el MC actúa como esclavo, sólo recibe órdenes de adquirir y enviar datos de temperatura vía puerto serial, almacenar datos de temperatura en la EEPROM y abrir o cerrar la electroválvula (estos dos últimos son recibidos una vez por quema).
- Si surge la necesidad de actuar como maestro, ahí se tendrán que activar temporizadores y sus vectores de interrupción. Sin embargo, hay que considerar que originalmente éstos no estaban activados y por ende no tendría noción de la etapa ni del momento de la etapa en la que se encuentra. Por ello la computadora debe enviar en periodos de 1 minuto dicha información.
- El hecho de que el MC haya tenido que tomar el control del proceso implica que no hay seguridad que la computadora tenga todos los datos generados. Es necesario contar con una bandera que se encargue de indicarlo. Cuando ésta tenga un valor específico, significaría que todos los datos almacenados en la memoria deben ser descargados antes de la próxima quema para que no se pierda el registro actual.
- Como consecuencia del punto anterior, también es necesario guardar los *parámetros* de la quema, de ahí la justificación de la organización de la memoria de datos, pues con ellos se complementa el respaldo de la información.

- Dentro de los parámetros de la quema, uno muy importante, que es la tasa de muestreo. Este parámetro determina la frecuencia con que se almacenan los datos y repercute en la duración del respaldo de la quema, es decir, acorde con el espacio reservado para los datos en la memoria, el tiempo que puede estar guardando la información. Por ejemplo, si la tasa es de 1 muestra cada minuto y el espacio reservado para cada dispositivo es de 450 palabras (16-b), significa que puede respaldar una quema de 450 minutos (7 horas y media), y si la tasa fuese de 1 muestra cada 2 minutos, la duración se dobla a 900 minutos (15 horas).

Descripción breve del programa por microcontrolador

Aún con el uso del lenguaje C, la cantidad de código fuente es grande e incluso la del pseudo código. Ambos son mostrados completamente en el apéndice B. Lo que se muestra enseguida es un esbozo del pseudocódigo.

1. Declaración de funciones, variables y constantes, así como la configuración del puerto serial, del LCD, los temporizadores y los vectores de interrupción necesarios. El pin que controla la válvula solenoide debe tenerlo en estado apagado.

2. Esperar a que el usuario inicie sesión con el sistema ya sea desde el teclado integrado o vía serial desde el programa por PC.

3. Cuando sea la PC la que inicia la sesión, hay dos solicitudes que puede hacer: iniciar el proceso o descargar datos de la memoria. Si se trata de descargar, se activa la función correspondiente y después de ello, regresar al punto 1. Si se trata de continuar, revisar primero si hay datos disponibles en la memoria de datos que no hayan sido registrados por la PC, en caso de haberlos, dar el aviso correspondiente en espera de dos cosas: que el usuario decida continuar o que prefiera declinar para descargar después los datos. Si aún así decidió continuar pese a perder datos, resta recibir los parámetros de la quema en orden riguroso: la temperatura objetivo, la hora y fecha de inicio y la tasa de muestreo. Después de ello se abre la electroválvula y se salta al paso 5. Pero si decidió no seguir, simplemente ir al paso 1.

4. Cuando sea el teclado el que inició la sesión, verificar si hay datos disponibles en la memoria que no hayan sido registrados por la PC y hacer el mismo procedimiento de confirmación de respuesta del paso anterior. Cuando se dé el caso de seguir a pesar de perder datos, capturar los parámetros en el mismo orden de la versión serial e ir al paso 6.

5. Este es el control por PC. Es un ciclo indefinido donde el MC simplemente espera un comando con límite de tiempo, el cual, si se llega a vencer, indica que se ha perdido la comunicación y ahora el MC tiene el control y se salta hasta el paso 6. En caso contrario, se envía un acuse de recibo del comando, se ejecuta y se dispone a esperar el siguiente comando. Cuando el comando recibido sea uno que indique la finalización del proceso, se rompe este ciclo y se va al paso 7. Si llega a recibirse la orden de abortar el proceso, se cierra automáticamente la válvula y se termina formalmente este proceso, indicando el tipo de fin que tuvo. Al principio de este paso el pin que controla la válvula solenoide debe estar en estado encendido.

6. Este es el control por MC. A este paso se puede llegar directamente desde el paso 4 o bien, cuando originalmente la PC tenía el control (periódicamente la PC envió información sobre la etapa en la que se encuentra el proceso) y se perdió la comunicación. Aquí es donde se indica que la PC no tiene registrados todos los datos y es lo que se revisa en los pasos 3 y 4. Al principio de este paso el pin que controla la válvula solenoide debe estar en estado encendido. Para abortar el proceso en cualquier momento, el operador debe recurrir al teclado del sistema y oprimir una combinación de dos teclas, la combinación debe ser tal que no sea fácil accionarla

por error. Básicamente el control por microcontrolador se trata de las tres etapas mencionadas en la sección 2.2:

6.1. Primera etapa. Es un ciclo indefinido que durará hasta ocurrir uno de dos casos, alcanzar una temperatura de 850°C y en caso de la temperatura objetivo sea esa (se trata de una quema de sancocho), terminar el proceso apagando la válvula solenoide y dejando al horno enfriarse, ir al paso 7. El otro caso es que la temperatura objetivo es mayor a los 850°C, típicamente arriba de 950°C (se trata de quema de esmalte), por lo que este ciclo se rompe al llegar a los 900°C para saltar hasta 6.2. Durante este ciclo se va almacenando la información de temperatura en la memoria de datos de acuerdo a la tasa de muestreo, de igual forma, cada media hora se enviará un aviso al usuario de incrementar la presión mediante el LCD y una señal sonora mediante el buzzer o piezoeléctrico.

6.2. Segunda etapa. Para llegar a esta etapa la quema es forzosamente de esmalte y consiste en un ciclo de 15 minutos de duración durante los cuales hay que mantener la temperatura en un rango de 900°C a 905°C, si está arriba del rango, se envía un aviso al operador de disminuir ligeramente la presión del gas, o de incrementarla en caso de estar abajo del rango. Esta espera es conocida como maduración del esmalte. Al igual que en 6.1, la temperatura se registra de acuerdo a la tasa de muestreo.

6.3. Tercera etapa. Una vez madurado el esmalte, simplemente hay que llegar a la temperatura objetivo y al alcanzarla, se termina el proceso en la misma forma que en 6.1, al igual que el almacenamiento de temperatura y los avisos de incremento periódico de la presión. Una vez terminada la esta etapa, ir al paso 7.

7. Enfriamiento del horno. Para enriquecer el contenido de la gráfica, se sigue graficando la temperatura cuando ésta descienda a los 600°C, después de ello, terminar formalmente el proceso.

4.1.3 Construcción del control por PC

La arquitectura de la computadora debe ser compatible con IBM-Intel, ya que predomina en el mercado mexicano junto con el sistema operativo Microsoft Windows y debe tener también un puerto de comunicación serial RS-232.

La mayoría de los lenguajes de programación y sus distintas herramientas de desarrollo para este tipo de computadoras son aptas para implementar el algoritmo. Los más comunes son el ensamblador, C/C++, Basic y Pascal y a recientes fechas, los lenguajes gráficos han adquirido popularidad, un ejemplo de ello es LabView [14] de National Instruments, que de hecho, fue el elegido. Este programa funciona en casi todas las plataformas de Microsoft Windows haciendo portátil al programa que se elabore con base en él (ocasionado por la fuerte presencia comercial de dicho sistema operativo). Además, posee herramientas que hacen sencilla la tarea de graficar datos, la generación de reportes, la validación de datos de entrada y en especial, si la expansión del proyecto llegase a requerir de funciones matemáticas complejas, cuenta con funciones adecuadas para ello. Su punto débil es que no ofrece muchas facilidades para elaborar un programa estructurado, pues carece de saltos de línea, o en este caso, de secuencias.

Consideraciones

Con LabView es posible hacer un programa de acuerdo con las normas especificadas en la HCI, descrita en 3.3.2. Tomando en cuenta las diez reglas ahí mencionadas, la interfaz se va desarrollando al considerar los siguientes puntos:

1. El objetivo del programa por PC es mostrar gráficamente el comportamiento de la quema y hacer un registro de la misma. Esto viene de la necesidad del usuario por conocer la forma en

cómo se está comportando el proceso de cocción. Hay que tomar en cuenta que en esencia, el algoritmo de la quema es el mismo que en el MC, la diferencia radica en que la PC está programada siempre para ser el maestro o control primario del proceso.

2. Las funciones que debe realizar son, primeramente, la conexión de la computadora con el microcontrolador, pues éste funge como unidad de adquisición de datos. También debe capturar los parámetros iniciales de la quema (temperatura objetivo, hora y fecha de inicio, etc.), obtener a intervalos específicos la lectura desde el termopar, la conmutación de la válvula solenoide, el despliegue de la temperatura interna actual del horno y la exposición gráfica de la curva que genera la relación temperatura-tiempo. También es necesario que se muestre un historial de eventos de la quema, tales como el inicio de sesión y el alcance de puntos críticos (como la inversión de la cristobalita y del cuarzo). Por ello, los elementos que debe contar son indicadores digitales (para la temperatura), un indicador gráfico (para la curva temperatura-tiempo), cajas de texto para la introducción de datos iniciales por el usuario e indicadores de texto para mostrar el historial de la quema. Al último pero no menos importante, un botón de aborto de la quema.

3. Ahora, la forma en como se van a disponer los elementos visuales debe ser tal que el usuario tenga en una sola pantalla todos los indicadores necesarios para saber qué temperatura está registrándose en el interior del horno, pero deben estar estratégicamente distribuidos para evitar que se pierda interés en la interfaz.

4. El guardado del registro de la quema será a base de archivos. Si la operación concluye sin complicaciones, entonces es necesario guardar los datos. Por ello debe aparecer al usuario un aviso en pantalla que le pida hacer dicho guardado exactamente después de terminada la quema. Es pertinente que el sistema proponga un nombre al archivo basándose en la fecha de la quema, pero también hay que dar opción al operador de cambiar el nombre a un formato que él ya conozca.

5. Otro aspecto deseable en el programa es tener la posibilidad de imprimir la gráfica, de hecho, sería mejor que inmediatamente después de guardar los datos en el archivo se le pregunte al usuario si desea imprimir la gráfica.

6. Aunado a los dos puntos anteriores, también es deseable registrar e imprimir gráficas de quemas anteriores. Esto implica la necesidad de incluir menús de opciones en la forma tradicional del ambiente Windows (situados en la parte superior izquierda de la interfaz).

8. Hay que proveer la opción de recuperar datos de la última quema efectuada. Es prudente recordar que si por algún motivo la PC perdió contacto con el MC, éste toma el control de la quema y almacena los datos en su memoria, por ende, es posible recuperar los datos posteriormente. Esto obliga a que el programa por PC envíe al MC la orden de descargar los datos disponibles, para inmediatamente después guardarlos en un archivo y si lo desea, graficar e imprimir. Aquí también se requiere de incluir un menú de opciones para acceder a la descarga de datos.

Descripción breve del programa por computadora

Al igual que con el programa por microcontrolador, hay una secuencia de pasos que implementan el algoritmo de quema del taller. Obviamente la secuencia en este caso es ligeramente diferente y se detalla a continuación.

1. Declaración de controles (aquellos que sirven para que el operador o usuario ingrese valores de distinta clase) e indicadores (aquellos que muestran información al operador, entre ellos un área de graficación). También se configura al puerto serial y se añaden los menús de opciones que incluyen tres operaciones: inicio de quema, descarga de datos y graficación e impresión de quemas anteriores.

2. Esperar a que el usuario elija una opción. En caso de que se trate de una descarga de datos o una graficación en impresión de quemas anteriores, se realiza la función en cuestión y después de eso se termina el programa.

3. Cuando la opción elegida sea la de inicio de quema, enviar un identificador o token de inicio de sesión al MC y la respuesta será una de dos posibles: la confirmación del inicio de sesión o una alerta sobre datos no registrados. Si se recibe la confirmación, se salta al paso 4. Si se recibió la alerta, significa que los datos de la última quema no están registrados en archivo alguno y lo recomendable es que se detenga el programa para realizar primero la descarga de tales datos. El usuario tiene un tiempo límite para contestar (medio minuto) que caso de ser rebasado, se asume que la elección fue detener el programa. Pero si el operador eligió continuar pese a perder los datos, ir al paso 4.

4. Enviar los parámetros de la quema en orden riguroso: la temperatura objetivo, la hora y fecha de inicio y la tasa de muestreo. Después de ello se abre la electroválvula y se salta al paso siguiente.

5. Desarrollo de las etapas de la quema. Se hace de manera idéntica a la versión del programa en el microcontrolador, también se incluye el enfriamiento del horno. La única diferencia es que la computadora siempre envía los comandos u órdenes y espera los datos en caso de solicitarlos. Durante el desarrollo de las etapas envía periódicamente (cada minuto) al MC la información sobre la etapa y el momento de la etapa en que se encuentra. El aborto del proceso se hace mediante la activación del botón correspondiente y para evitar que se haga por error, es preferible que se active mediante una combinación de teclas en lugar de un clic o pulsado del ratón.

4.1.4 Protocolo de comunicación de la PC con el MC.

Con anterioridad se mencionó que la relación de PC con MC es la de *maestro – esclavo*. En la figura 4.4 se mostró el formato o trama de la comunicación (que también es el formato de almacenamiento de datos en la memoria de datos del sistema). Ahora se muestra la forma del protocolo.

1. PC y MC se encuentran en un estado inicial, donde el primero está listo para enviar una orden (comando) y el segundo está listo para recibirla.
2. El MC está en espera de la orden, pero tiene un tiempo de espera determinado, el cual, en caso de ser superado, activa en el microcontrolador la orden de conmutarse como maestro del proceso.
3. Si la orden fue recibida dentro del tiempo de espera, lo primero que el microcontrolador hará es enviar un acuse de recibo que indica a la computadora que se ha recibido bien el comando y que se está ejecutando.
4. El MC ejecuta la orden y después de terminarla envía un aviso conocido como *comando ejecutado* hacia la PC, la cual está a la espera de dicho aviso.

5. Finalmente, tanto PC como MC están listos para enviar y recibir un nueva instrucción respectivamente.

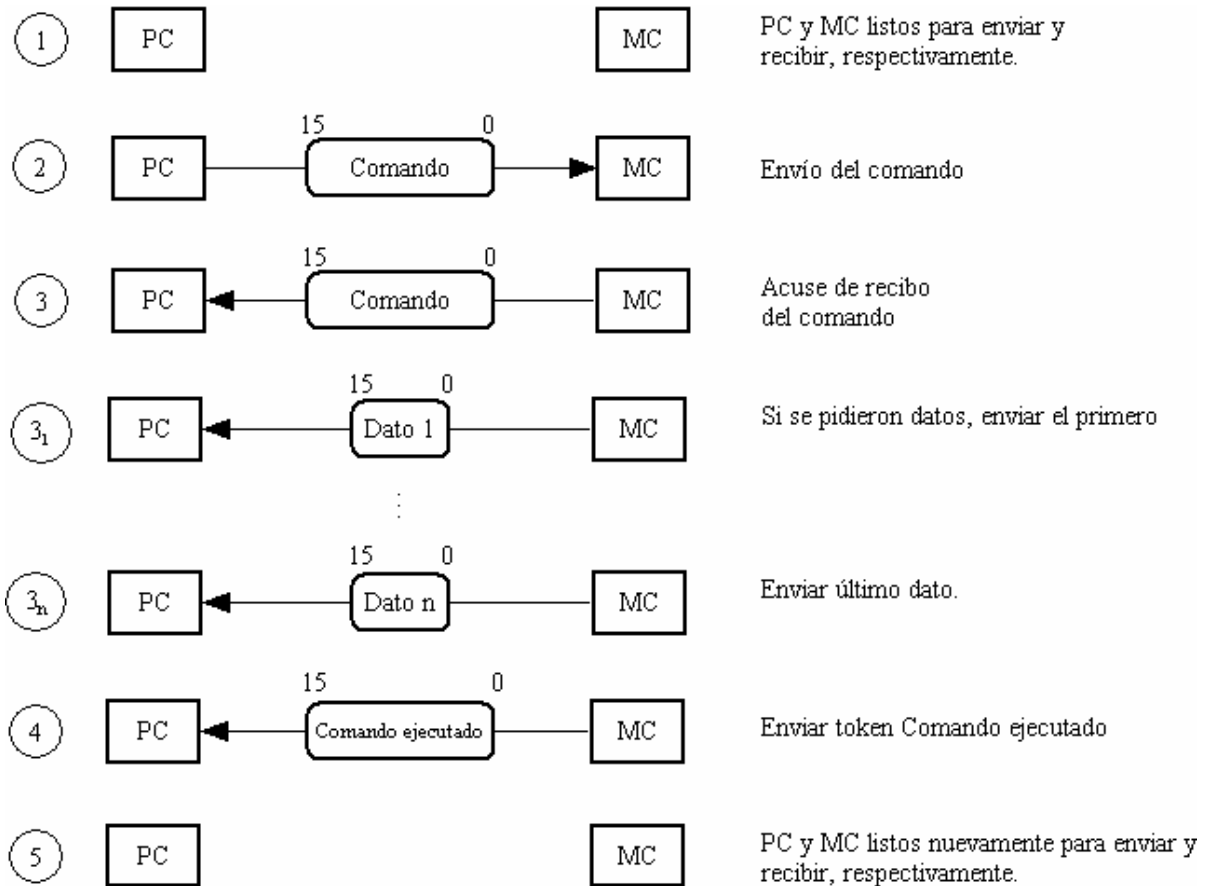


Figura 4.6. Protocolo de comunicación entre PC y MC.

4.2 Diseño final de la etapa de sensores y actuadores

Con base en los puntos vistos en la sección 3.4, la construcción esta etapa se muestra a continuación.

4.2.1 Construcción del termómetro o sensor de temperatura

Esta parte contiene varios conceptos relacionados con la teoría del termopar. En caso de que el lector tenga dudas sobre este dispositivo, se recomienda que vea primero el apéndice A, donde el transductor es detallado lo suficiente como para comprender lo que a continuación será presentado.

Consideraciones

- El termopar tiene un comportamiento no lineal.
- Se elige el tipo K debido a que es el que presenta una no linealidad menor en comparación con los demás tipos y cubre el rango de temperatura del horno.

- El voltaje proporcional a la temperatura que entrega (el termopar tipo K) es muy pequeño, del orden de microvoltios para temperaturas debajo de 100 °C y de unos cuantos milivoltios cuando se encuentra alrededor de los 1000 °C.
- Durante la cocción del sancocho, el agua que contiene se evapora al igual que pequeñas porciones de minerales contenidas en el material (como hierro y carbono). Esos vapores pueden ocasionar tanto oxidación del termopar como acumulación de partículas minerales que en determinado momento podrían formar uniones no deseadas en el transductor. La solución es muy simple, una carcasa con recubrimiento de porcelana. El efecto que tiene sobre la capacidad sensitiva del termopar es nula a temperaturas arriba de los 100 °C.
- Hay que poner atención en las conexiones del dispositivo con el resto del circuito, pues en ellas se forman termopares no deseados, aparte de que pueden servir como antena receptora de ondas electromagnéticas (interferencia eléctrica). También hay que poner atención a la influencia de la temperatura ambiente. Todos estos sucesos se dan en la unión fría del termopar, mismos que hay que compensar.
- El rango de temperatura del horno es de 0 a 1250 °C, pero sólo interesan las temperaturas arriba de 200 °C ya que no hay pieza cerámica alguna que se cueza abajo de tal punto. Esto hace un poco más sencillo al circuito del termómetro, pues en el rango de 0 a 150 °C es donde típicamente se presenta la mayor parte de la no linealidad del termopar (tipo K), ocasionando que no sea tan rigurosa la compensación de la unión fría, mas no prescindible.
- Es pertinente contar con un cable de extensión para el termopar, con fin de que el transductor y el resto del circuito se encuentren alejados lo suficiente como para evitar que el calor generado por el horno afecte el funcionamiento de los componentes electrónicos del circuito.

Con base en los puntos anteriores, la estructura general de un termómetro basado en termopar [15] debe estar compuesta por las siguientes partes: *adaptación de señal*, donde los efectos físicos de interferencia eléctrica se anulan; *compensación de unión fría*, cuyas funciones son cancelar el termopar formado en la conexión de la unión fría con el resto del circuito y regular su comportamiento no lineal y finalmente, la *amplificación* de la señal, para que el dato pueda someterse a conversión analógica-digital. La figura 4.7 muestra esta estructura.

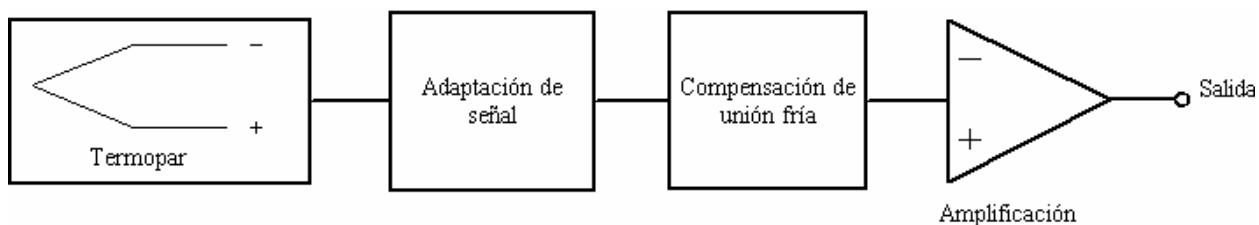


Figura 4.7. Forma general de un termómetro basado en termopar.

De acuerdo con el diseño del circuito, las etapas no necesariamente deben tener ese orden, la configuración puede variar de acuerdo con los resultados deseados.

Descripción de la adaptación de señal

El objetivo es eliminar la interferencia eléctrica en la conexión del termopar con el resto del circuito (ver figura 4.8). En este caso, la interferencia puede ser por señales de radio frecuencia que circulan por el medio y por ruido eléctrico en general. Ciertamente ellas son del orden de microvoltios, pero desafortunadamente el voltaje que entrega el termopar también es de unos cuantos microvoltios a temperatura ambiente.

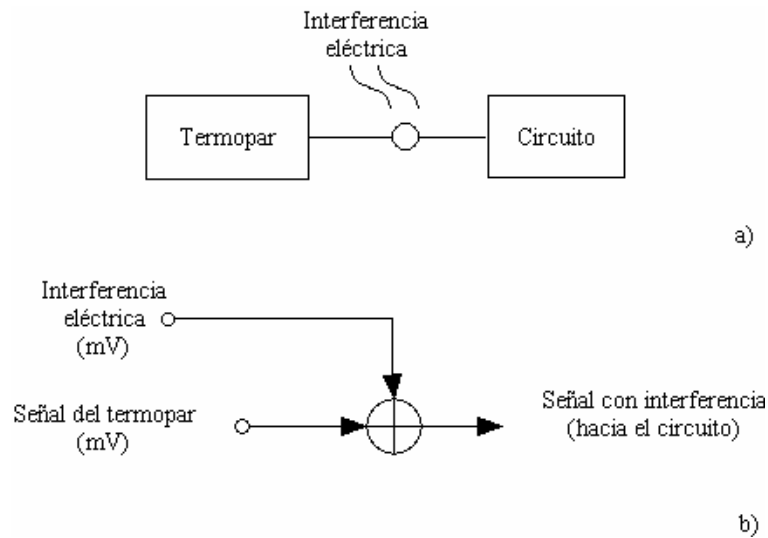


Figura 4.8. Interferencia eléctrica en la conexión del termopar con el circuito: a) Representación a bloques, b) Representación del flujo de control.

La solución es emplear un filtro pasabajas.



Figura 4.9 Supresión de la interferencia mediante filtro pasabajas.

Descripción de la compensación de la unión fría.

Para efectuar la cancelación del termopar formado en la conexión del termopar con el circuito hay que considerar que en condiciones ideales ese termopar no deseado mide la temperatura ambiente, luego entonces, se debe contar con otro sensor de temperatura que genere una señal de referencia pero de signo contrario tal que se anulen entre sí, siendo preferente que el sensor extra trabaje en la escala Celsius. Por su parte, para quitar la parte no lineal (referida aquí como *codigo*) se requiere básicamente de un generador de voltaje que regule la parte en cuestión.

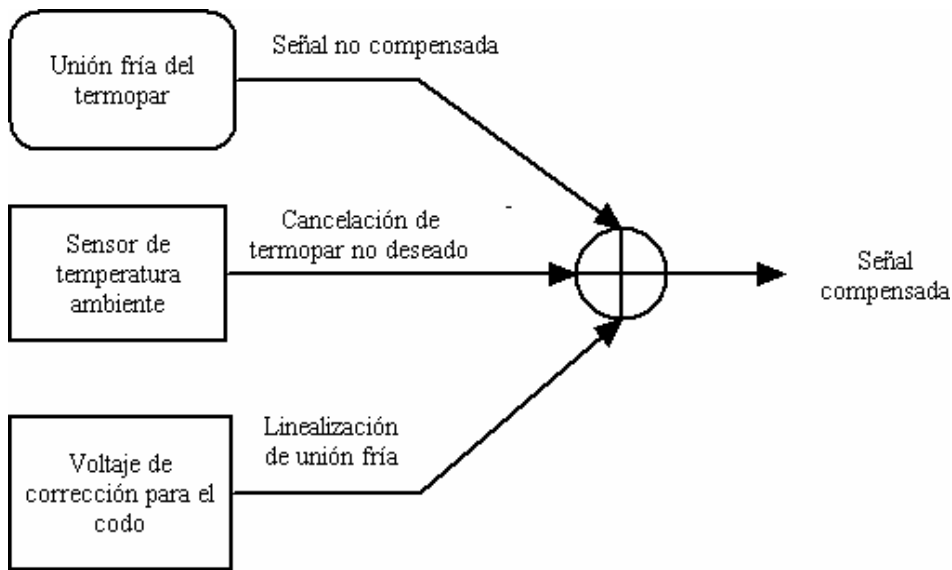


Figura 4.10. Compensación de la unión fría del termopar.

Construir este tipo de circuito con dispositivos discretos es complicado y requiere de constantes calibraciones. Hoy día se cuenta con circuitos integrados que realizan por sí solos la compensación de la unión fría.

El circuito empleado es el LT1025A [16] de Linear Technologies Inc., puede trabajar con los tipos más comunes de termopares (incluido el tipo K, que es el adecuado para este sistema), realiza la cancelación del termopar no deseado, corrige el codo o zona no lineal y además tiene un lazo de realimentación interna que ofrece mayor estabilidad al circuito, así como una red de selección de termopares. Su error es de medio grado centígrado en condiciones normales de trabajo, es decir, con una alimentación de voltaje y una temperatura de operación especificadas en sus hojas técnicas.

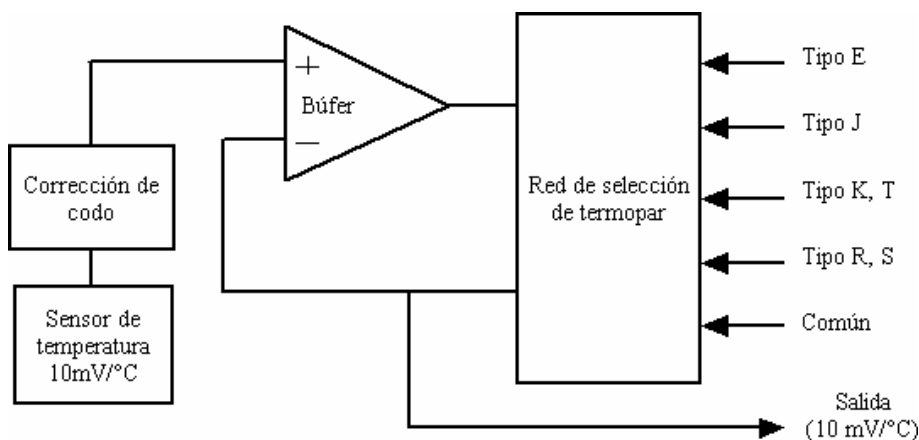


Figura 4.11. Compensación de unión fría realizada por el LT1025.

Es de observar que su terminal de salida tiene como referencia una resolución de 10 mV/°C, es decir, proveer 10 mV por cada grado centígrado. En la figura 4.11 se observa cómo está

constituido el circuito integrado; se parte de un sensor de temperatura seguido de una etapa de corrección de codo (parte no lineal), la señal resultante va a un búfer en donde la señal es procesada junto con una señal de retroalimentación. Dicha señal viene de una red de selección de termopar, donde se aprecia una terminal común que se encarga de trabajar la señal proveniente del termopar. Esta terminal se usa principalmente cuando la amplificación de la señal se da antes de la compensación de la unión fría. Por otra parte, el voltaje con que se alimenta al circuito depende del rango de temperatura a medir. Se recomienda que sea al menos 2 V arriba del factor *temperatura * resolución*, o bien, $1250^{\circ}\text{C} * 10 \text{ mV}/^{\circ}\text{C} = 12.5\text{V}$ más los 2 V de reserva dan un total de 14.5 V, el cual se redondea a 15 V.

Descripción de la amplificación.

El hecho de que la señal del termopar es pequeña hace necesario amplificarla. Se podría optar ya sea por un amplificador de instrumentación o por un amplificador operacional de precisión, el punto es que ambos deben ser capaces de manejar esos niveles de voltaje tan bajos. Por razones de comodidad y economía se prefiere usar un amplificador operacional, específicamente, el modelo LT1012, de la misma marca del LT1025. Su costo es menor en comparación con los otros y además se adecua perfectamente al compensador de unión fría de manera tal que con una sencilla configuración es posible desarrollar el termómetro.

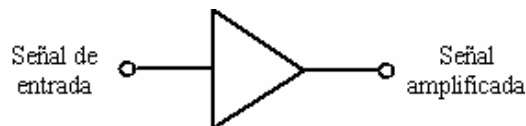


Figura 4.12. Amplificación.

Circuito del termómetro basado en termopar

Finalmente, en la figura 4.13 se muestra al circuito construido. La configuración mostrada del transductor se conoce como *termopar aterrizado*, su utilidad es que permite hacer una amplificación en modo común (lo que da pie a utilizar un amplificador operacional de precisión en vez de un amplificador de instrumentación).

Se aprecia que la adaptación se hace con un simple circuito RC, ahí se eliminan los componentes de alta frecuencia que pudieran adherirse a la señal para ir inmediatamente después a un amplificador no inversor, donde se multiplica por un factor de 255 a 260, dependiendo del resistor variable de 100Ω . El condensador que está presente en el lazo de retroalimentación sirve para compensar en frecuencia. De esa forma, la señal amplificada y sin interferencia va al LT1025, específicamente a los pines Gnd y Com, ahí sufre el proceso de compensación de unión fría, dando como resultado una salida final de aproximadamente $10 \text{ mV}/^{\circ}\text{C}$.

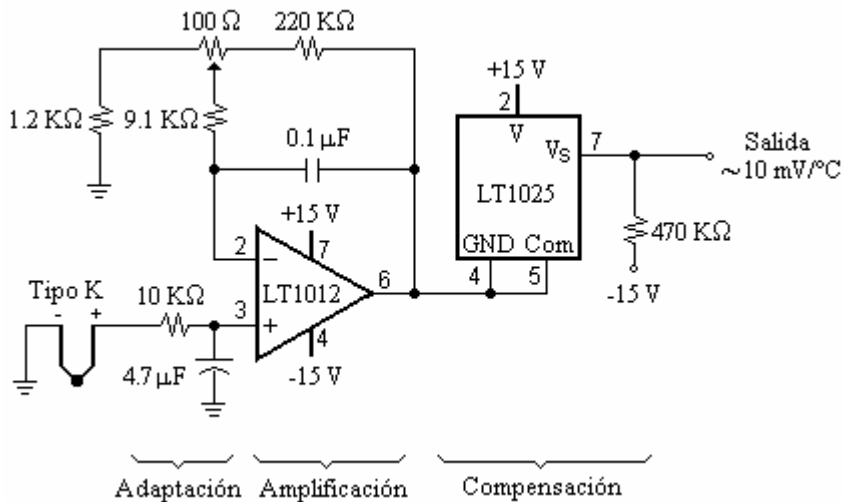


Figura 4.13. Termómetro basado en termopar tipo K.

Acerca del termopar, en la figura 4.14 se muestra su forma física, con su termopozo de recubrimiento cerámico (alúmina, para ser precisos) de aproximadamente 46 centímetros de largo. El montaje del termopar en el termopozo es de tipo sellado-aislado (ver la figura A-3 del apéndice A).

Es pertinente profundizar en el tema del termopozo. Su interés radica en que está hecho para el rango de temperaturas que se alcanzan en el horno, sólo que para ello necesita estar a la temperatura del horno, esto es, formalmente lo que mide el termopar es la temperatura del termopozo. Al ser de alúmina, y dependiendo de la longitud, va a tener un tiempo de respuesta (calentamiento, en este caso) más grande con respecto a otros tipos de termopozo, especialmente los metálicos.

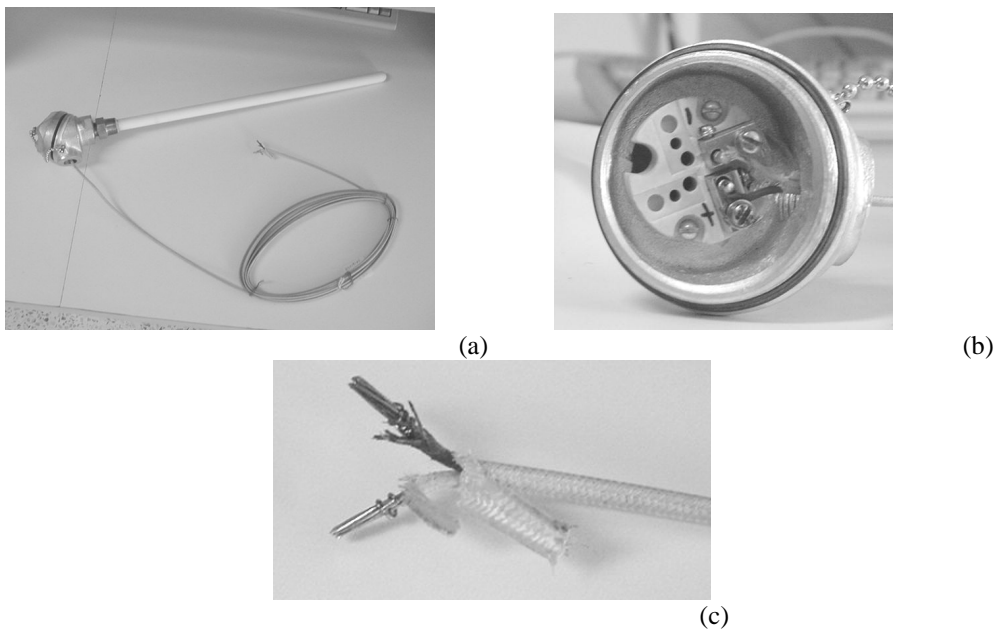


Figura 4.14. Apariencia del termopar empleado. (a) Completo, (b) Acercamiento a la unión fría, (c) Cable de extensión para termopar tipo K.

4.2.2 Construcción del circuito para la electroválvula.

Respecto a la electroválvula, el modelo con el que se cuenta es el S201GF02N5EG5 de GC Valves [URL 11], dos vías, tipo normalmente cerrada, bajo consumo de energía (150 mA nominales en corriente alterna, 173.5 mA reales), soporta presión de fluidos de hasta 130 psi y trabaja con gas natural, LP y aire. Dado que emplea corriente alterna, su circuito consiste en un triac modelo TIC226, un optoacoplador MOC3031 y tres resistores. Se añade una etapa de supresión de picos y un fusible de 1 A que protegen a la válvula de sobrecargas en la línea eléctrica.

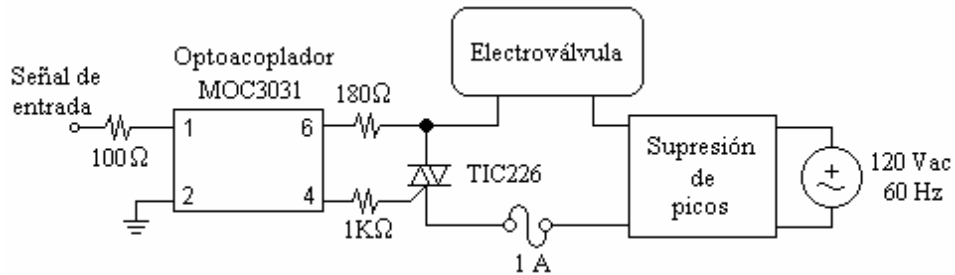


Figura 4.15. Circuito de control de electroválvula.

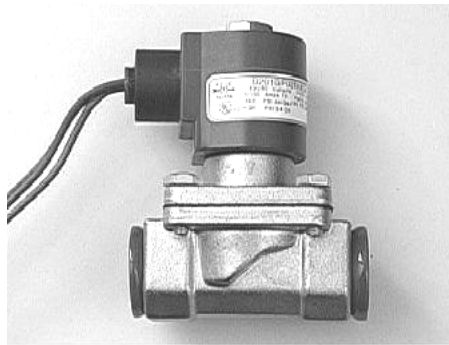


Figura 4.16. Electroválvula S201SF02N5EG5, dos vías, normalmente cerrada.

4.3 Respaldo de energía en el sistema

El no-break o UPS incluido es el modelo INTERNET 350U de la marca Tripp-Lite [URL 12], mostrado en la figura 4.16. Tiene una capacidad de 350 VA y el fabricante indica que a plena carga puede respaldar por un máximo de 2.5 minutos. Para hacer un estimado del tiempo que puede soportar al sistema es necesario considerar lo siguiente: El amperaje medido en el circuito entero, exceptuando la PC y la electroválvula, es de 485 mA rms. La electroválvula junto con circuito consume un máximo de 250 mA rms, dando un total de 735 mA rms que ajustando a VA, es igual a 8.933 VA. Haciendo una simple división se tiene que $(350 / 8.933) \times 2.5$ minutos da un resultado de 93.95 minutos, o bien, 1.63 horas de respaldo, suficiente para que se restablezca la energía.

Hay que aclarar algo sobre este punto: esta relación de energía se verá afectada cuando se agreguen más dispositivos al sistema, por lo que en determinado momento será necesario aumentar la capacidad del UPS. Cuando llegue el momento, se recomienda adquirir el modelo

INTERNET 750U (750VA) de la misma marca para duplicar el tiempo de respaldo o bien, si se desea tener poco más del triple, hay que adquirir el modelo TE1200 (1200VA), con un significativo aumento en precio (alrededor de 599 dólares americanos a precio del primer trimestre del 2005).



Figura 4.17. UPS modelo Internet 3500 de Tripp-Lite.

Capítulo 5 - RESULTADOS

Una vez que el sistema está construido, el siguiente paso son las pruebas y resultados del mismo. Debido a sus características, las pruebas contemplan dos aspectos: la facilidad de su uso por parte del operador del proceso de quema (lo cual involucra a la interfaz tanto en la computadora como en el microcontrolador) y la eficiencia en la medición de factores físicos, en este caso, la temperatura.

5.1 Interfaz del sistema (programa en computadora)

Se aborda este aspecto en primera instancia porque es con lo que tendrá contacto el usuario la mayoría de las veces.

El resultado final se muestra en la figura 5.1, la cual representa la pantalla que el usuario se encuentra exactamente al ejecutar el programa. Se puede apreciar que la interfaz es sencilla pero completa, es decir, tiene los elementos suficientes para que el usuario (o mejor dicho, el operador) comprenda lo que está pasando durante la quema. Los elementos que se encuentran en la interfaz son:

- Cajas de texto donde se introducen los parámetros de la quema (temperatura objetivo, nivel de presión inicial en el tanque de gas estacionario y la tasa de muestreo. Nótese que no se pide la hora ni tampoco la fecha, pues el sistema los determina automáticamente.
- Un área donde se grafica el historial de la temperatura interna del horno, cuyo periodo de actualización depende de la tasa de muestreo.
- Una barra de menús, donde hay dos tipos principales: “Opción” y “Datos”. El primero tiene dos campos, “Iniciar nueva quema” y “Salir”. El segundo también tiene dos campos, “Recuperar datos desde MC” y “Graficar datos de quemas anteriores”.
- Un indicador digital de la temperatura en el interior del horno.
- Un indicador del estado de la electroválvula. En la figura indica un estado “apagado”, mostrando un color verde oscuro, el estado “encendido” lo muestra con un color verde brillante. Nótese que en la figura se le refiere simplemente como “VALVULA”.
- Una caja de texto denominada “Estado de la operación” que va reseñando la quema.
- Y finalmente, un botón para abortar la operación cuando sea necesario. Para evitar que sea accidentalmente accionado, su activación es con la combinación de las teclas SHIFT + F12. En la figura 5.1 dicho botón está deshabilitado.

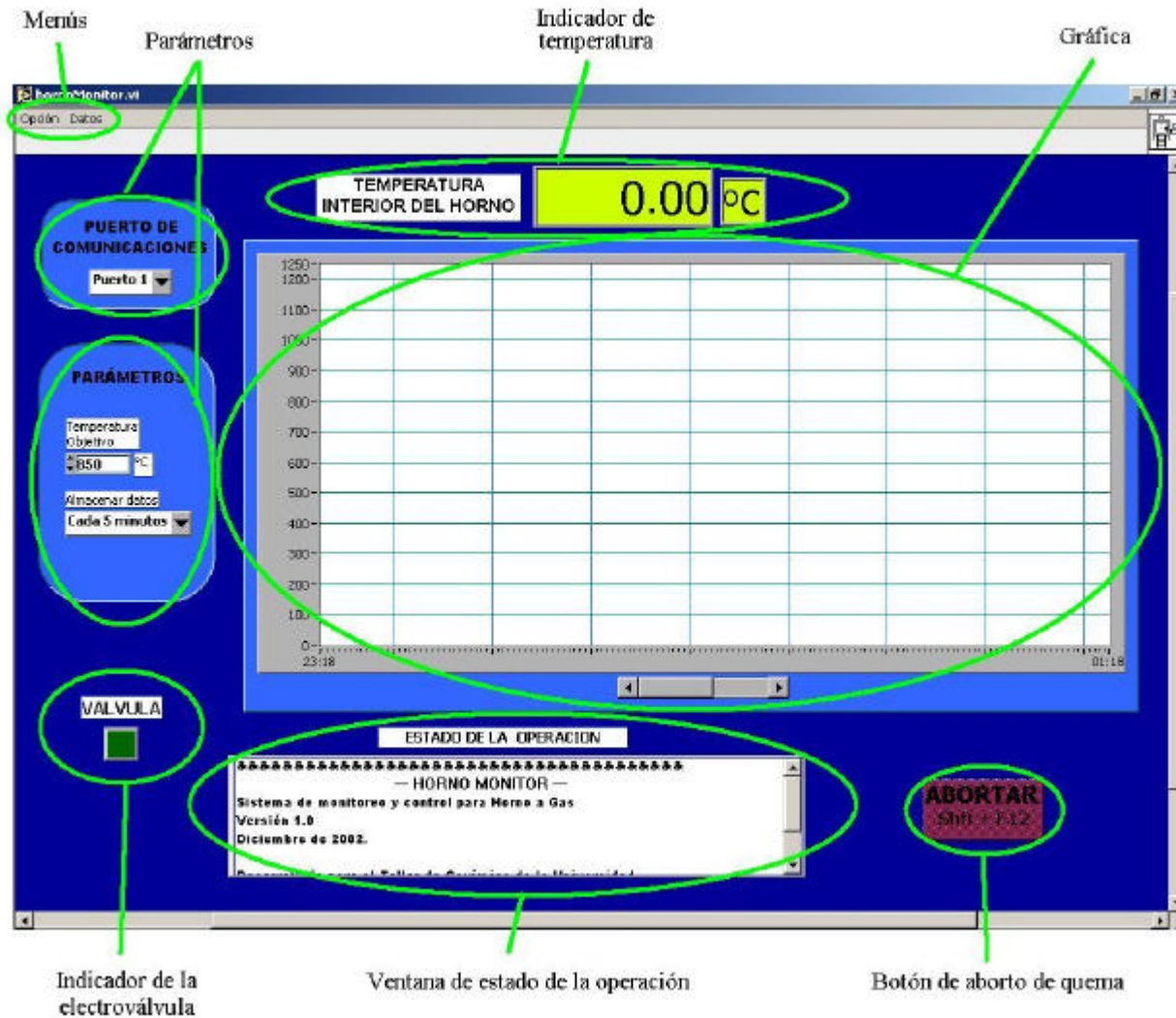


Figura 5.1. Interfaz del sistema (en PC).



Figura 5.2. Menús “Opción” y “Datos”.

En cuanto a colores, se eligió el azul marino para el fondo y el azul rey para los paneles en razón de ofrecer un contraste suave (en atención a las reglas del HCI). El indicador de la temperatura interior del horno tiene un fondo verde cuyo contraste con el resto de la interfaz hace que el usuario ubique a primera vista al indicador.

Son tres las tareas que realiza el programa.

5.1.1 Opción “Iniciar nueva quema”

Al hacer clic en esta opción se inicia propiamente el proceso de quema. Lo primero que aparece en pantalla es una caja de mensaje que pide al usuario cerciorarse de que la válvula del contenedor de gas esté totalmente abierta. Después hay que hacer clic en “CONTINUAR” para que el sistema siga haciendo los preparativos. En operación normal, esta ventana sería la única que aparece al usuario durante el proceso, el resto de avisos se le comunican a través del indicador “ESTADO DE LA OPERACIÓN”.



Figura 5.3. Primer aviso del sistema al operador en una quema.

Inmediatamente después quedan deshabilitados los menús “Opción” y “Datos” para evitar que inconscientemente el usuario inicie otro proceso que interrumpa el actual.

El sistema entonces prosigue inicializando el puerto serial elegido. Si llegase a haber un problema con el puerto serial, se envía otro aviso al operador donde se le explica que debido a fallas en el puerto serial, el programa debe cerrarse y que debe probar con otro puerto.

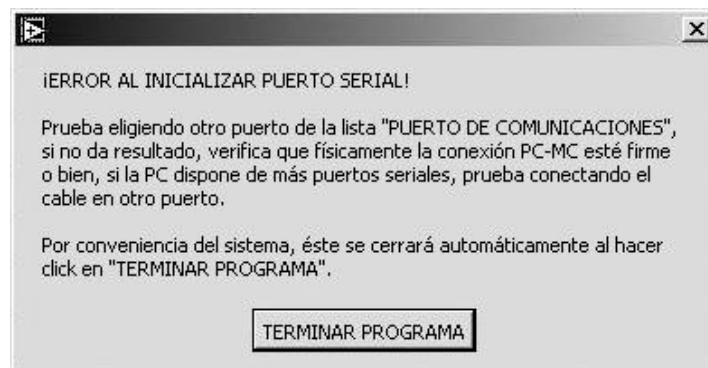


Figura 5.4. Aviso de error en inicialización del puerto serial.

En la sección 4.1.3 se habló de la limitación de LabView para realizar un programa estructurado al no incluir saltos incondicionales a líneas, lo cual es apreciado con este detalle: ante un evento que implique una interrupción en la secuencia del programa es preferible cerrarlo y reiniciar después. Igual sucede cuando la comunicación se pierde durante la quema, primero el programa trata de restablecer automáticamente la comunicación y en caso de fallar, vuelve a enviar un aviso a través de una ventana parecida a la de la figura 5.4 al operador diciéndole que debe cerrarse el sistema, pero con la ventaja es que el MC adoptaría el control de la quema y los datos pueden ser recuperados después.

Pero en caso de que no hubiese tal error, lo siguiente es verificar que el parámetro “Temperatura Objetivo” no exceda los límites del horno, 1250°C , pero tampoco estar por debajo de los 573°C , que sería la inversión de la cristobalita, pues no hay pieza cerámica que se cueza por debajo de esta temperatura. De nueva cuenta, si el usuario ingresó un valor no permitido, se le comunica del error y se cierra el programa.

Antes de accionar la electroválvula para iniciar formalmente la quema, el programa enviará un identificador o token al MC diciéndole que este quiere iniciar una nueva sesión. El MC le responderá de dos formas: regresándole el mismo token indicando que va a iniciar la quema o regresándole el token *descargar*, el cual da a entender que en la última quema hubo una interrupción en la comunicación PC-MC y que hay datos disponibles en la memoria de datos, los cuales no es seguro tenga registrado la PC. El programa entonces mostrará al operador una caja de mensaje donde le explica tal situación y da opción a elegir entre dos opciones: continuar con el proceso con el inconveniente de que se van a perder los datos anteriores o bien, cerrar el programa y después elegir la opción “Descargar datos desde MC”. Todo esto se ilustra en la figura 5.5.

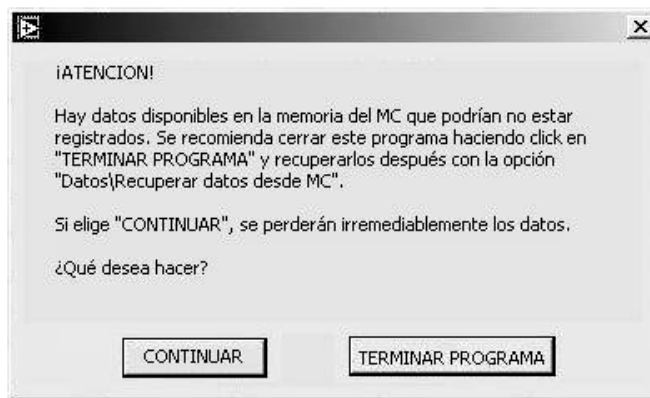


Figura 5.5. Pregunta al usuario sobre descarga de datos.

Una vez hecho esto, el sistema manda al MC abrir la electroválvula y lo muestra en su pantalla cambiando el color del indicador “VALVULA” y después habilita al botón “ABORTAR”, el cual se mantendrá así hasta que termine la quema (figura 5.6).

A partir de este momento, todo aviso al operador será a través del indicador “ESTADO DE LA OPERACION” acompañado de una señal audible para captar su atención. En la figura 5.7 se muestran como ejemplos, el aviso cuando se ha alcanzado la inversión del cuarzo y el aviso de incrementar la presión de las válvulas L y C en 0.05 Kg/cm^2 una vez que ha transcurrido el conteo de 30 minutos.



Figura 5.6. (a) Indicador “VALVULA” indicando estado abierto, (b) Botón “abortar” habilitado.

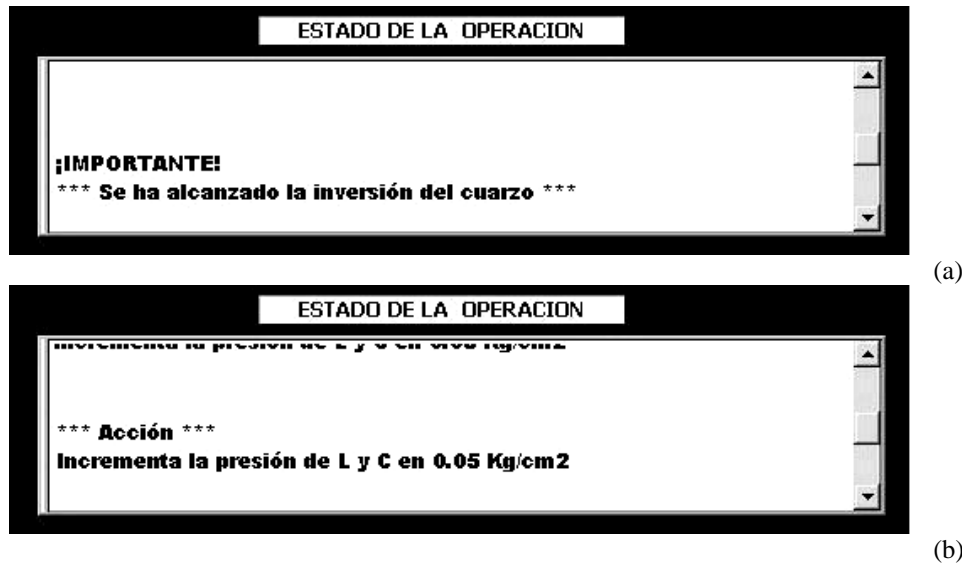


Figura 5.7. Ejemplos de avisos al usuario. (a) Cuando se alcanza la inversión del cuarzo, (b) Cuando se debe incrementar la presión de L y C.

Por otra parte, en el indicador “TEMPERATURA INTERIOR DEL HORNO” se muestra el valor actual de temperatura, mientras que en zona de graficación se van imprimiendo los datos a la tasa de muestreo marcada por el parámetro “Almacenar Datos”. La figura 5.8 muestra la curva de temperatura de una quema en la que la última medición es de 573 °C. Ahí la curva se nota accidentada, pero corresponde a una simple prueba con el horno vacío, donde además se varió de manera arbitraria con la presión de las válvulas (el objetivo de esa prueba fue simplemente llegar lo más pronto posible a los 573°C).

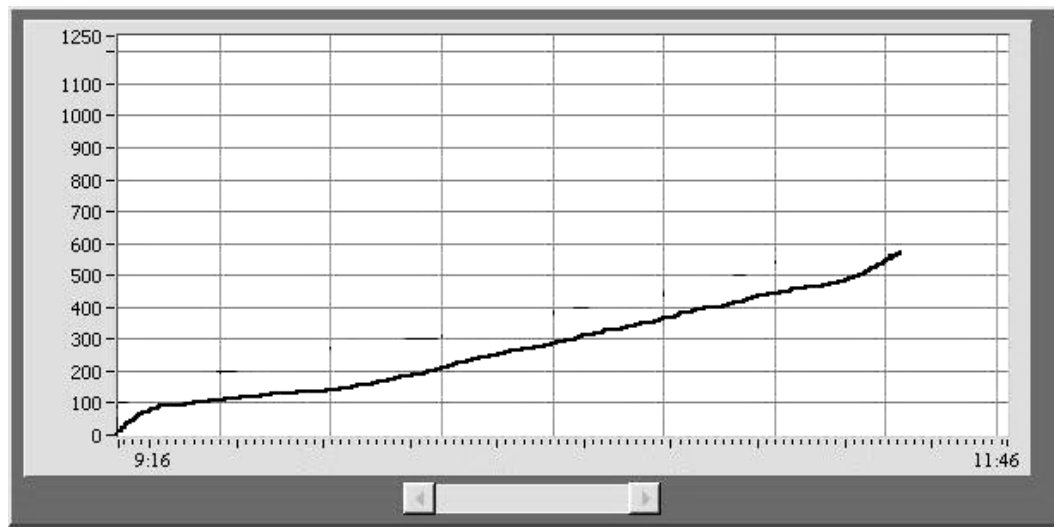


Figura 5.8. Ejemplo de curva de temperatura.

De esa manera, el sistema seguirá monitoreando la temperatura y graficándola hasta llegar a la temperatura objetivo y esperar el enfriamiento del horno (la temperatura tiene que descender a 600 °C para terminar formalmente la quema).

Finalmente, el sistema pide al usuario que ingrese un nombre para el archivo donde se guardarán los datos. En la figura 5.9 se observa cómo el sistema propone automáticamente un nombre basándose en la fecha en que se realizó la quema, pero el usuario es libre de cambiárselo. Este paso es obligatorio, no se permite cancelar el guardado de los datos. El registro de la quema es guardado en formato XLS, que es compatible con el software Microsoft Excel (versión 97 como mínimo), esto le permite al operador realizar sus propias gráficas, claro que para ello debe tener experiencia con dicho programa.



Figura 5.9. Guardado de datos en archivo.

Después de ello se le pregunta al usuario (a través de una caja de mensaje) si desea imprimir la gráfica; en dado caso, utiliza la impresora por omisión configurada (si es que hay alguna instalada en la computadora) y su resultado es simplemente la misma zona de la gráfica en la interfaz, ajustada en tiempo conforme a la duración total de la quema y ocupando la máxima área posible de la hoja (depende de la configuración que tenga la impresora).

Y formalmente se da por concluida la quema y su registro. De nueva cuenta, se da un aviso a través de una caja de mensaje (ver figura 5.10) donde se avisa que todo el proceso ha concluido exitosamente y que es necesario cerrar el programa.

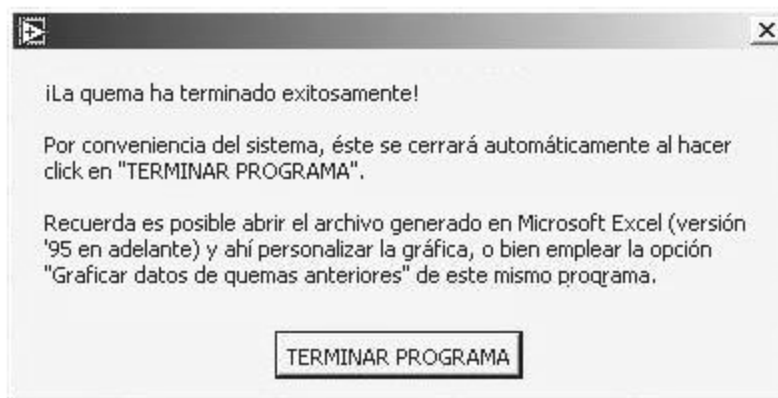


Figura 5.10. Aviso final después de una operación exitosa.

Por otra parte, si el proceso fue abortado, de todos modos se le pide al usuario guardar los datos, añadiendo ahora la leyenda “(abortado)” en el nombre propuesto por el sistema para

indicar el tipo de fin que tuvo el proceso (y otra vez, hay plena libertad de cambiar el nombre) y se procede a cerrar el programa, con su caja de mensaje correspondiente.

5.1.2 Opción “Recuperar datos desde MC”

Esta opción realiza la función de enviar al MC la orden de descargar todos los datos que tenga disponible en su memoria (de datos obviamente) correspondientes a la última quema efectuada (haya sido exitosa o abortada). Esta operación sólo se puede realizar al inicio de la sesión con el MC.

Al hacer clic aparece en el indicador “ESTADO DE LA OPERACION” un aviso al operador sobre la acción que solicitó. Inmediatamente después busca inicializar el puerto serial indicado por la lista “PUERTO DE COMUNICACIONES” y en caso de error, termina el programa. Si no hubo error, envía la orden de descargar datos al MC y comienza el trabajo.

Si durante la descarga ocurre algún error en la comunicación, se le envía un aviso al usuario ilustrado en la figura 5.11 donde una vez más, se cierra automáticamente el sistema al hacer clic en el botón “TERMINAR PROGRAMA”.

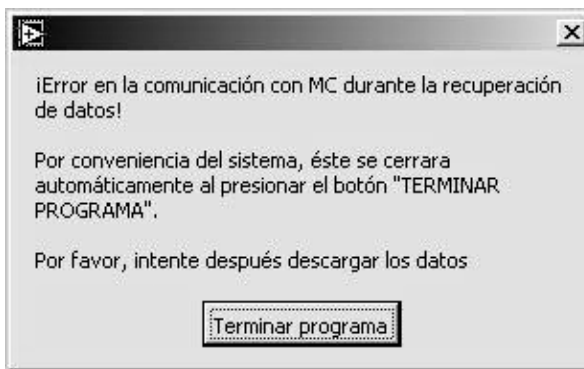


Figura 5.11. Aviso de error durante descarga de datos.

En caso contrario, el programa continuará su tarea hasta obtener el último dato. Después de esto, se le pedirá al usuario que guarde los datos en un archivo, de forma idéntica al de la figura 5.9. Aquí también el sistema propone un nombre por omisión al cual se añade la leyenda “(recuperado)”, pero dando libertad al usuario de cambiarlo.



Figura 5.12. Guardado de datos recuperados.

Una vez guardado el archivo se le pregunta al usuario si desea graficar e imprimir los datos, independientemente de su respuesta, se envía después un último aviso donde se explica que todo el proceso (de descarga) ha concluido satisfactoriamente y que es necesario cerrar el programa. Estos dos últimos pasos iguales a los de la opción “Iniciar nueva quema”.

5.1.2 Opción “Graficar datos de quemas anteriores”

Esta tercera opción realiza la función de graficar e imprimir el registro de alguna quema anterior en la impresora por omisión del sistema operativo de la computadora, donde procura utilizar toda el área posible de la hoja configurada en la impresora. Para ello el sistema generó, junto con el archivo XLS, otro archivo de extensión RQ cuyo formato es conocido por el mismo sistema y lo emplea este fin. Una vez finalizada esta tarea, se envía un aviso idéntico al operador donde se explica que es necesario terminar el programa.



Figura 5.13. Elección del archivo a graficar e imprimir.

5.2 Interfaz del sistema (programa en microcontrolador)

Hablar de la interfaz del programa en el MC se remite a que el usuario verá los datos en una pequeña pantalla de cristal líquido y que, en caso de que sea el microcontrolador quien lleve el seguimiento del proceso desde el comienzo, el usuario ingresará los parámetros de la quema por medio de un teclado numérico. Obviamente esta interfaz no es capaz de graficar ni generar reportes, pero al menos puede mostrar el usuario la temperatura interior del horno y emitir señales audibles cuando se le requiera para incrementar la presión de las válvulas L y C o para avisarle si algún punto crítico de la quema se ha alcanzado.

Por las dimensiones de la pantalla, no es posible colocar párrafos completos sobre lo que pasa en el momento, por lo que se recurre a mostrar palabras cortas que el usuario irá interpretando de acuerdo al manual provisto junto con el sistema y que está anexo a este documento en el apéndice C.

5.3 Precisión en la medición de temperatura

Obviamente, de poco sirve una interfaz amigable si la medición de la temperatura no es precisa. La referencia que se tomó para comparar la efectividad del medidor es el termómetro digital que tiene originalmente el horno por razones obvias: tiene exactamente el mismo tipo de termopar (tipo K), que el empleado en este proyecto y electrónicamente está adaptado para desplegar las temperaturas que alcanza el horno. La única diferencia que tiene con el empleado en el sistema es que es aproximadamente 15 centímetros más corto el termopozo (y por ende, el termopar), lo cual afecta el tiempo de respuesta.

En las pruebas hechas, los cuidados que se tuvieron al comparar las mediciones fueron básicamente el tener relativamente cerca ambos termopares uno del otro, mantener la puerta y la chimenea del horno cerradas para evitar que éste se despresurice y realizar la prueba sin carga en el carro del horno, es decir sin piezas; se hizo así porque las piezas cerámicas tienden a absorber calor mientras se cuecen y conservarlo una vez cocidas y podrían hacer variar la lectura del sensor (dependiendo de qué tan cerca estén de él). Además, las mediciones se tomaron apagados los quemadores del horno. La justificación de ello es que mientras los quemadores estén encendidos, las válvulas L y C deben tener la misma presión, pues eso se traduce en el grosor de la flama despedida y con ello, el calentamiento del interior del horno; si no se colocan a la misma presión simplemente van a generar varias zonas de calor distintas unas de otras. Retomando la figura 2.4 se aprecia que los quemadores están aparentemente distribuidos equidistantemente en el horno y a la misma altura, pero en realidad su distribución varía por escasos centímetros y hay dos quemadores que no están a la misma altura de los demás. Los puntos anteriores indican que mientras estén encendidos los quemadores no es fácil tener un calentamiento homogéneo del horno, para ello se requerirían sensores digitales de presión y servoválvulas que hagan un poco más uniforme las flamas de los quemadores y si se considera que no están a la misma altura los quemadores, es aún más difícil hacer la comparación.

Por ello, primero se calentó el horno hasta los 1080°C (usando el indicador de temperatura original del taller) procurando que la presión en las válvulas L y C fuese la misma para después cerrarlas, dejando enfriar al horno. De las leyes (la primera, específicamente) de la termodinámica se puede inferir que el calor tiende a distribuirse uniformemente en un espacio cerrado una vez que ya se ha llegado a un equilibrio térmico [URL 13], aplicándolo a este caso, se espera que los dos termopares tengan la misma lectura al llegar a ese equilibrio, e irá descendiendo en valor hasta alcanzar la temperatura ambiente. Por *equilibrio térmico* se ha de entender que cuando los quemadores están encendidos y la presión no se ha modificado por un tiempo prolongado, llega el momento en que ya no aumenta la temperatura e incluso se llega a tener una temperatura uniforme en el interior del horno. Se recuerda que estas pruebas fueron hechas sin piezas a cocer por la razón antes expuesta.

Esto no quiere decir que en uso normal el medidor fallará, simplemente que esas pruebas se optaron por hacerlas con todas esas características para ver la proximidad de ambos medidores.

En la figura 5.14 se muestra la gráfica comparativa de los dos medidores. La curva del medidor de este proyecto se denota por puntos y la del termómetro digital presente en el horno se delimita con línea interrumpida. La forma de la gráfica es meramente ilustrativa, se usa para

mostrar cuán cercana es la lectura de los dos medidores, mas no es una curva característica de una quema normal. Se aclara que los puntos de la curva del medidor del proyecto se tomaron convirtiendo el voltaje de salida del circuito del termopar (ver figura 4.12) a grados centígrados (cada 10 mV corresponde a un grado centígrado).

Se observa que las curvas son muy semejantes, con diferencia típica de 0.5°C , lo cual indica una buena precisión del medidor del proyecto. Se detectan tres zonas donde la diferencia es un poco mayor que en el resto de la curva: la primera zona es de 650 a 670°C con una diferencia que va de 2 a 3 grados centígrados, la segunda zona va de 895 a 910°C con una diferencia de hasta 3.2 grados centígrados y la última va de 1040 a 1050°C con una diferencia de hasta 2 grados centígrados.

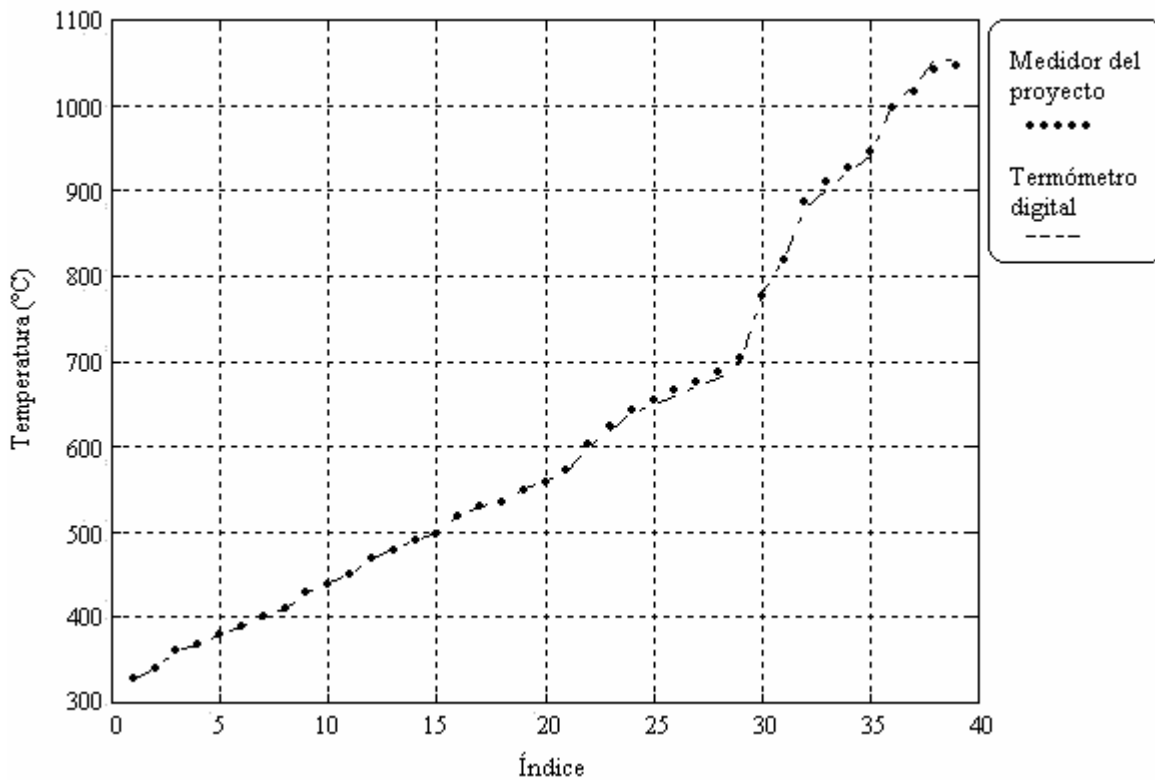


Figura 5.14. Gráfica comparativa de la medición de temperatura entre el medidor del proyecto y el termómetro digital presente en el horno.

Capítulo 6 - CONCLUSIONES

De los resultados expuestos en el capítulo 5, es momento de dar las impresiones finales de este proyecto. De antemano se está consciente de que este sistema no ha cubierto todas las necesidades del Taller de Cerámica, por lo que también se expondrán los puntos de expansión del sistema

6.1 Conclusiones generales

En primer lugar, se cumplieron los objetivos planteados 1.4:

- Satisface una de las principales necesidades del Taller de Cerámica: grafica en tiempo real y genera reportes de la quema. La gráfica representa el registro de la quema (el cual tiende a ser llamado *bitácora* por los encargados del taller) y los reportes son el almacenaje del registro en un medio electrónico, además, da la oportunidad de imprimir desde el mismo programa o si el usuario lo prefiere, exportar el archivo al popular programa Microsoft Excel (a través de Microsoft Graph) y ahí personalizar la graficación de los datos.
- A través de la electroválvula se ofrece más seguridad a la quema, pues permite abortar el proceso en el momento que sea necesario ya sea a través del programa por PC (con la combinación de teclas *Shift + F12*) o por medio del teclado del sistema. Como se explicó en el capítulo 4, esto provoca que la electroválvula se cierre inmediatamente.
- Es personalizado para el taller, esto es, se ajusta a sus necesidades y evita tener características innecesarias para el mismo. De hecho, los sistemas de control de temperatura comerciales (ver 1.2.2) en su mayoría se remiten a simplemente indicar de manera digital o analógica la temperatura y a dar una señal tipo ON/OFF que active o desactive algún actuador, pero no grafican ni generan reportes del proceso.
- Como complemento del punto anterior, también sirve como auxiliar del operador al ahorrarle el trabajo de graficar y al avisarle visual y audiblemente cuándo se ha llegado a un punto crítico en el proceso e incluso puede servir para ir entrenando a gente inexperta en este tema.

- Es económico comparado con el costo material de los sistemas comerciales. El neto costo por materiales fue de 1613.00 pesos en moneda nacional a precios del primer semestre del 2002, mientras que los sistemas comerciales se encuentran arriba de dos mil quinientos pesos (sin incluir actuador ni sensor). Por otra parte, el termopar (junto con termopozo, cabeza y extensión) costó aproximadamente mil pesos y la electroválvula costó mil quinientos pesos. Si bien es cierto que se requiere de una computadora personal para hacerle trabajar, no se contempla el costo de ésta dado que en el taller cuentan con una, asignada por la misma Universidad.
- El diseño del control por MC está preparado para expandirse y cubrir posteriormente el resto de las necesidades del taller.
- Gracias al UPS, el sistema se sobrepone a cortes de energía y garantiza el seguimiento de la quema.

En segundo lugar, la experiencia de trabajar con LabView enseña que es posible realizar con facilidad aplicaciones que satisfagan problemas de este tipo. No hay que olvidar que LabView está orientado inicialmente a la instrumentación virtual, razón por la cual posee herramientas poderosas de cálculo y procesamiento de datos y señales. Como se indicó en su momento, sus funciones (o mejor dicho, instrumentos virtuales) para graficar son muy flexibles y ofrecen también los medios de realizar una interfaz amigable, aunque no dan muchas facilidades para realizar un programa estructurado. Dado que se trabajó con la versión 5.1, quizás en nuevas versiones incluya herramientas que faciliten aún más la elaboración de aplicaciones de este tipo.

En tercer lugar y escudriñando en aspectos de instrumentación, el manejo de termopares para medir la temperatura se minimizó con el empleo del circuito integrado LT1025A pues éste realiza el trabajo más difícil: la compensación de la unión fría y con ello reduce el espacio del circuito. Si bien, su grado de error es de hasta dos grados centígrados en el rango de 0 a 100 °C, la naturaleza de este problema hace intrascendente ese detalle.

6.2 Limitaciones y expansión del sistema.

Básicamente las limitaciones de este sistema son las necesidades del taller en lo que al proceso de cocción se refiere y que no fueron cubiertas y que a su vez representan los puntos de expansión de este proyecto.

6.2.1 Cobertura de temperatura

Al utilizarse un único termopar no se garantiza que su lectura sea correspondiente a la temperatura neta del interior del horno, es decir, ese termopar puede cubrir solamente una zona del horno y por ello, la lectura que entregue al sistema aplica a las piezas que más cerca estén del termopar. Esto es comprensible, cada quemador no entrega exactamente el mismo calor que los demás ocasionando que el calentamiento sea desproporcionado en la medida en que sea desproporcionada la flama de cada quemador.

La propuesta es expandir a cuatro termopares colocados estratégicamente como lo muestra la figura 6.1, incluyendo los dos que existen actualmente. De esa forma, la cobertura en el horno es casi total, pues se ocupan las zonas donde típicamente se colocan las piezas a ser cocidas.

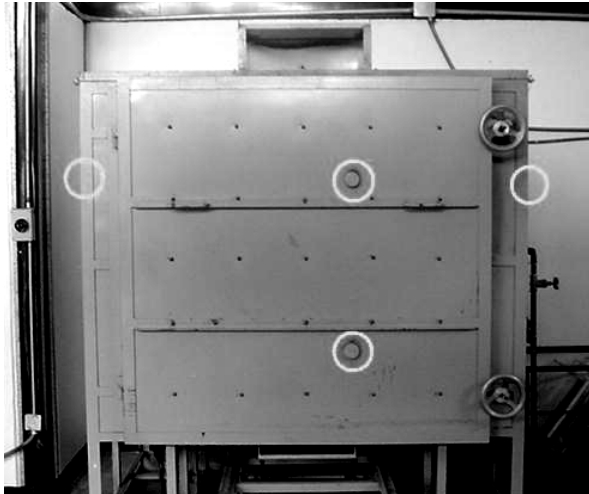


Figura 6.1. Propuesta de colocación de termopares extra en el horno.

6.2.3 Medición de presión

No se hace ni la medición ni el registro de presión en las válvulas lateral y central. Actualmente cada una tiene un indicador analógico para regular manualmente la presión, lo deseable es contar con sensores de este tipo que indiquen la presión y que además ésta se incluya en el registro.

La propuesta es emplear dos ejemplares, uno para cada válvula. Una buena opción sería el modelo 24FC15PS, marca Honeywell, que tiene un rango de medición de 15 psi, suficiente para satisfacer el problema. Éstos se colocarían físicamente lo más próximo posible a las válvulas y no se recomienda retirar los indicadores analógicos que ya existen en el horno.

En razón a estos dos primeros puntos se justifica la distribución de la memoria en el sistema (ver 4.1.1) y la existencia de ocho canales de entrada en la etapa de conversión analógica a digital, al igual que el tamaño de memoria de programa sobrante.

6.2.4 Homogenización de temperatura

Otro aspecto que no cubre el sistema es homogenizar la temperatura interior del horno. Para lograr esto de manera automática se requieren de dispositivos electromecánicos (o neumáticos en su defecto) conocidos como *servoválvulas* o en otras palabras, reguladores de presión. Si una electroválvula hace un control tipo ON/OFF, una servoválvula hace un control proporcional. En el mercado existen varios productores de estos dispositivos y uno de ellos es MOOG Inc., que ofrece toda una gama de modelos que se ajustan a los requerimientos del consumidor. El problema es que el costo de esos dispositivos, puede llegar a rebasar el precio de mil dólares americanos (que al tipo de cambio actual con el peso mexicano deja muy pocas posibilidades de ser adquirido para este proyecto). El tiempo dirá si es posible y factible incorporarlos al proyecto.

Hay un detalle interesante: para realizar plenamente la homogenización de temperatura no bastaría con agregar dos servoválvulas que reemplazaran las válvulas manuales existentes, sino que en realidad serían necesarias 16 servoválvulas que regulen la presión de cada quemador. De esa manera y con un adecuado control de estos dispositivos se lograría el propósito. Pero el costo

del proyecto se dispararía irracionalmente, sin contar el costo de los sensores de presión extras, uno para cada quemador, lo cual también implica hacer cambios drásticos en la tubería. Esto, desde luego, es incosteable e innecesario.

6.2.5 Mejora en la seguridad

La inclusión de la electroválvula ofrece un nivel de seguridad mayor al proceso y como ya se ha explicado, su activación se da exclusivamente al inicio del monitoreo de la quema y su desactivación puede ser realizada en cualquier momento ya sea mediante la computadora o mediante el teclado del microcontrolador. Si bien esto representa una pequeña ventaja al ahorrarle tiempo al usuario en el cierre de la llave principal de gas, tampoco es una solución total, pues también es deseable que el sistema sea capaz de analizar la situación de la presión, tal que si detecta una anomalía, mande cerrar la electroválvula de inmediato y de esa forma, dar una seguridad aún mayor. Por “anomalía” se ha de entender la despresurización repentina de las válvulas, provocadas seguramente por una obstrucción en alguno de los quemadores. En otras palabras, darle al sistema un perfil de inteligencia, la suficiente como para tomar decisiones trascendentes basadas en datos desde los sensores.

Cada uno de los puntos anteriores es posible agregarlo al sistema, pues sus capacidades lógicas y físicas lo permiten. Por ello es que se procuró economizar los puertos de Entrada/Salida al igual que la memoria de datos y la memoria de programa en el microcontrolador. Dado que el código hexadecimal del sistema ocupa aproximadamente 6.3 KBytes, todavía restan 25.7 KBytes para las expansiones.

Lo importante es que al agregar los puntos de expansión, la línea de operación del sistema no se modifique mucho para que el operador no tenga problemas en su uso. En otras palabras, actualmente el usuario conoce una forma de trabajar con el sistema, conoce un algoritmo; si la expansión es tal que modifica drásticamente dicho algoritmo, hará que el usuario pierda identificación con el sistema y que le tome más tiempo regresar a un ritmo normal de trabajo.

Apéndice A - TERMOPAR Y ELECTROVÁLVULA

A.1 Breve descripción del termopar

Un **termopar** es un transductor empleado para medir temperatura y se basa en el concepto de fuerzas electromotivas (EMF, por sus siglas en inglés) generadas por efecto de la temperatura, o bien, *efecto termoeléctrico*. Tiene un costo relativamente bajo y un amplio rango de medición, gracias estas dos características ellos son empleados en múltiples aplicaciones industriales como uno de los métodos preferidos para medir temperaturas medias y altas, son inmunes a vibración mecánica, simples de manufactura y no requieren de voltaje de excitación. Pero sus desventajas son que producen una relativamente baja señal de salida (orden de microvoltios para temperatura ambiente) y su comportamiento es no lineal, esto es, la relación EMF – temperatura presenta elementos logarítmicos. Esto hace que requiera un dispositivo de medición sensible y estable que sea capaz de compensar estas anomalías y que haga lineal al sensor. Por otra parte, la baja señal de salida demanda el aislamiento del ruido eléctrico en el circuito de medición

Constitución del termopar

El termopar se forma a partir de dos conductores eléctricos, cada uno de diferente aleación metálica, a uno de ellos se le llamará *material positivo* y al otro, *material negativo*. Típicamente son construidos a manera de cable con funda resistente a la temperatura y a menudo se le integra una coraza. En uno de los extremos del cable, los dos conductores están eléctricamente empalmados (corto circuito) ya sea mediante un trenzado o mediante soldadura; precisamente esta terminación es la que se conecta térmicamente al objeto a ser medido, por lo que recibe el nombre de *unión caliente* (*hot junction*) cuya temperatura es mucho mayor que la del otro extremo. El nombre “unión caliente” es un convencionalismo, dado que también se pueden medir temperaturas negativas, en ese caso, la relación de voltajes entre las terminales se invierte.

El otro extremo es conectado a un dispositivo de medición, esto es, el circuito que procesa su señal con el fin de determinar la temperatura cercana a la unión caliente y recibe el nombre de *unión fría* o *unión de referencia*.

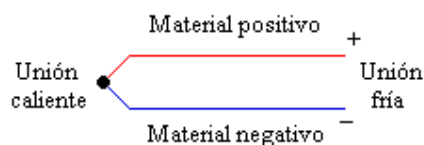


Figura A.1. Forma general del termopar

De manera formal, se dice que los termopares manipulan el hecho de que una EMF entre dos metales disimilares es producto de la diferencia sus temperaturas (conocido como gradiente), habiendo tres efectos implicados al momento de realizar circuitos electrónicos con el transductor. Estos efectos son el efecto Seebeck, el efecto Peltier y el efecto Thompson. El primero describe la EMF existente entre dos materiales disimilares, donde se define al *coeficiente Seebeck* (o también como *sensitividad termoeléctrica*) como el cambio del EMF material con respecto a un cambio en temperatura, el cual es una función no lineal de la temperatura. El segundo efecto se refiere al EMF que es reversible y se asocia con cambios en temperatura. El tercero relaciona el gradiente térmico reversible y a la EMF en un conductor homogéneo. De estos tres efectos el Seebeck es el de mayor peso al momento de diseñar circuitos basados en termopar.

Voltaje Seebeck

Los metales generan un voltaje en circuito abierto conocido como voltaje Seebeck, proporcional a la diferencia entre las temperaturas de la unión fría y la unión caliente.

$$V_{\text{Seebeck}} = \alpha (T_{\text{uc}} - T_{\text{uf}}) \quad (\text{Ec. A.1})$$

Donde T_{uc} es la temperatura de unión caliente, T_{uf} es la de la unión fría y α es el coeficiente Seebeck. De esta fórmula se deduce que si se conoce V_{Seebeck} y T_{uf} es posible determinar T_{uc} , por ende, un termómetro basado en termopar debe controlar la temperatura de la unión fría y conocer al coeficiente de Seebeck, el cual indica la sensibilidad termoeléctrica del material y cuyas unidades están dadas en $\text{mV}/^{\circ}\text{C}$ (medidas en la unión fría a una temperatura de 0°C). La tabla A-1 muestra los coeficientes Seebeck para varios materiales.

Tabla A.1. Coeficientes Seebeck.

Material	Coef. Seebeck	Material	Coef. Seebeck	Material	Coef. Seebeck
Aluminio	3.5	Oro	6.5	Rodio	6.0
Antimonio	47	Hierro	19	Selenio	900
Bismuto	-72	Plomo	4.0	Silicio	440
Cadmio	7.5	Mercurio	0.60	Plata	6.5
Carbono	3.0	Nichrome	25	Sodio	-2.0
Constantano	-35	Níquel	-15	Tantalio	4.5
Cobre	6.5	Platino	0	Telurio	500
Germanio	300	Potasio	-9.0	Tungsteno	7.5

En términos prácticos se considera que la unión caliente es la fuente del voltaje Seebeck, pero en realidad se genera a lo largo del alambre, de forma que si dicho alambre tuviera la misma temperatura en toda su extensión, no habría voltaje alguno generado. De forma similar se considera que los voltajes de la unión son generados en la terminal fría entre el alambre del termopar y el contacto con el circuito (que por lo general es de cobre), haciendo necesario medir la temperatura de la unión fría, sin embargo, esa temperatura es el punto de referencia para medir la diferencia de temperaturas a lo largo del circuito del termopar.

Consideraciones

Todo lo anteriormente dicho, al momento de llevarlo a la práctica hay que tomar en cuenta cinco implicaciones:

1. Un tercer metal puede ser introducido para conectarlo a un circuito electrónico que procese la señal (en este caso, el voltaje Seebeck) el cual no tendrá impacto negativo alguno sobre el termopar siempre y cuando ambas terminales estén a la misma temperatura.
2. La señal de salida es generada por un gradiente de temperatura a lo largo de los alambres, por lo que es importante que la calidad de los alambres sea mantenida aislándolos por medio de corazas. La calidad puede estar comprometida por la contaminación del ambiente en que opere. Para temperaturas debajo de 400 °C, en un ambiente no corrosivo, no representa mayor problema y el aislamiento no necesita ser muy alto, pero a temperaturas arriba de 1000 °C sí lo es.
3. El voltaje generado es una función de la diferencia de temperaturas entre las uniones caliente y fría, pero eso era tradicionalmente cuando la unión fría era mantenida a una temperatura de 0 °C por medio de un baño de hielo, técnica que es impráctica y ha sido reemplazada por un circuito de compensación de unión fría. Esto se logra midiendo la temperatura exclusivamente de dicha unión con un sensor alternativo de temperatura y aplicar un voltaje de corrección a la medición del termopar. Esta corrección puede ser hecha eléctricamente en hardware o matemáticamente en software en los modernos sistemas de medición, el cual además permitiría la corrección de la no-linealidad del termopar.
4. El bajo nivel de salida requiere que se evite la interferencia eléctrica y magnética de motores, cables de poder y transformadores. Para ello existen técnicas como el trenzando los alambres (por ejemplo, una trenza cada diez centímetros) para reducir la interferencia magnética, o el uso de cable blindado para reducir interferencia eléctrica.
5. Como inferencia del punto 3, el ambiente en que opere el sensor debe ser considerado, pues la exposición a atmósferas oxidantes a altas temperaturas puede degradar significativamente a algunos tipos de termopar, especialmente a los que contienen rodio (B, R, S),

Tipos de Termopares

Como se ha de deducir, puede haber muchos tipos de termopares (tantos como el número de combinaciones de aleaciones lo permita) pero sólo un pequeño número de ellos ha sido estandarizado al punto de tener tablas de calibración, códigos de color y una designación por letras. Para el continente Americano, el estándar E230 de la Asociación Americana para Pruebas y Materiales (ASTM, por sus siglas en inglés) provee todas las especificaciones antes mencionadas.

De esa manera, hay cerca de trece tipos de termopar estandarizados, de los cuales ocho han conseguido una designación internacional. Cada tipo tiene características que le hacen ser elegible entre otros según la aplicación, entre ellas, por supuesto, el rango de temperaturas que pueden medir. En la tabla A-2 se muestran los trece tipos de termopares estandarizados, donde se muestra la constitución principal de cada aleación, su porcentaje de error mínimo, su rango de temperatura total en °C y su rango de aplicación útil (esto es, en el que tiene un comportamiento más estable).

Tabla A.2. Características de los tipos de termopares

Tipo	Material Positivo	Material Negativo	Error	Rango total (°C)	Rango útil (°C)	Comentario
B	Pt, 30%Rh	Pt, 6%Rh	0.5% @ >800°C	50 a 1820	1370 a 1700	Bueno para altas temperaturas, donde no requiere compensación de unión fría.
C	W, 5%Re	W, 26%Re	1% @ >425°C	0 a 2315	1650 a 2315	Para su uso a muy altas temperaturas, pero es quebradizo.
D	W, 3%Re	W, 25%Re	1% @ >425°C	0 a 2315	-	Idéntico al tipo C.
E	Ni, 10%Cr	Cu, 45%Ni	0.5% o 1.7°C	-270 a 1000	95 a 900	Para medianas y bajas temperaturas
G	W	W, 26%Re	1% @ >425°C	0 a 2315	-	Idéntico al tipo C.
J	Fe	Cu, 45%Ni	0.75% o 2.2°C	-210 a 1200	95 a 760	Para altas temperaturas con ambientes reductores
K*	Ni, 10%Cr	Ni, 2%Al 2%Mn 1%Si	0.75% o 2.2°C	-270 a 1372	95 a 1260	Para altas temperaturas en ambientes oxidantes.
M	Ni	Ni, 18%Mo	0.75% o 2.2°C	-50 a 1410	-	-
N*	Ni, 14%Cr 1.5%Si	Ni, 4.5%Si 0.1%Mg	0.75% o 2.2°C	-270 a 1300	650 a 1260	Un tipo relativamente nuevo, reemplaza al tipo K a temperaturas medias.
P	Platinol** II	Platinol** II	1.0%	0 a 1395	-	Un más estable pero caro sustituto para los tipos K y N.
R	Pt, 13%Rh	Pt	0.25% o 1.5°C	-50 a 1768	870 a 1450	De gran precisión a alta temperatura
S	Pt, 10%Rh	Pt	0.25% o 1.5°C	-50 a 1768	980 a 1450	Idéntico al tipo R.
T*	Cu	Cu, 45%Ni	0.75% o 1.0°C	-270 a 400	-200 a 350	Para bajas temperaturas, tolerante a humedad.

(*) Tipos más usados

(**) Marca registrada de Englehard Industries.

Simbología de materiales: Al = Aluminio, Cr = Cromo, Cu = Cobre, Fe = Hierro, Mg = Magnesio, Mo = Molibdeno, Ni = Níquel, Pt = Platino, Re = Renio, Rh = Rodio, Si = Silicio, W = Tungsteno

Códigos de color

Los colores empleados varían de acuerdo al país, en la figura A-2 se muestran los códigos para distintos países (México adopta el de EE.UU.)

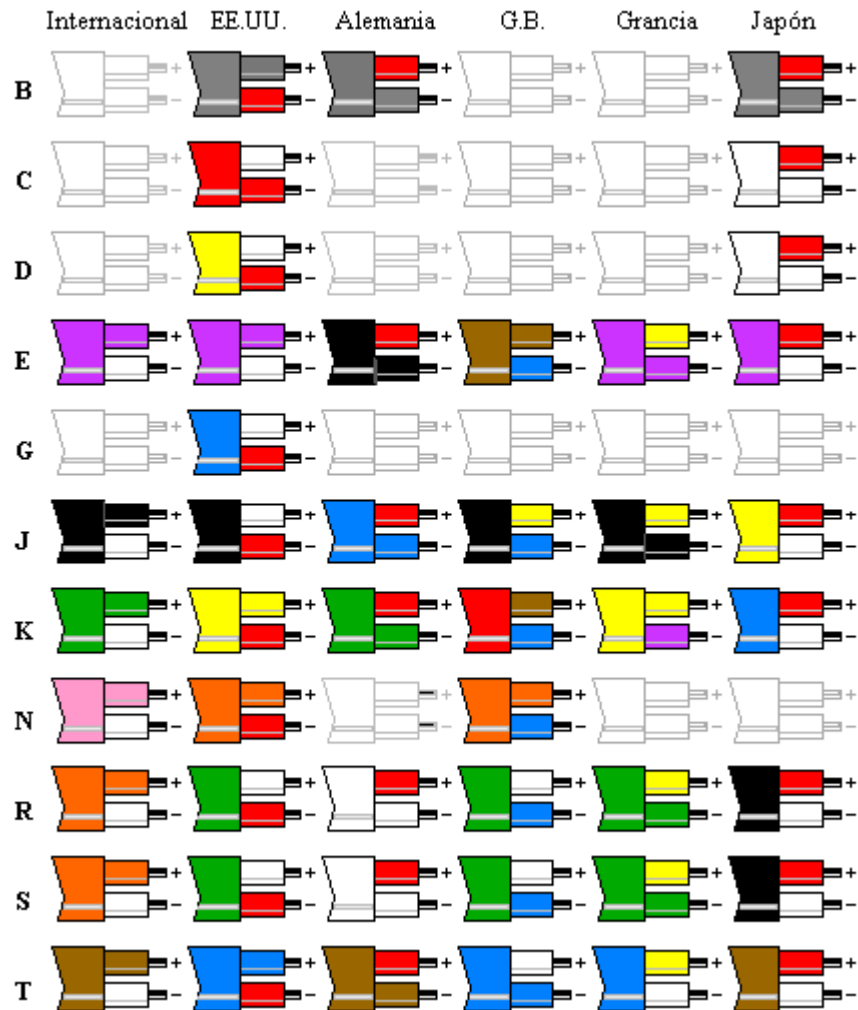


Figura A.2. Códigos de color por país.

Montaje del termopar

Hay cuatro modos de montar un termopares en su termopozo ya sean de acero sin estaño, cerámicos (alúmina) y finalmente, con o sin aislamiento eléctrico a través de óxidos minerales. La importancia del termopozo radica en que protege al termopar de los probables agentes oxidantes del medio.

El primero es el *sellado y aislado del termopozo*, el cual es relativamente confiable en su lectura, pero su tiempo de respuesta inicial (para la medición) es muy largo (puede tomar hasta 75 segundos). El segundo es el *sellado y aterrizado al termopozo*, el cual tiene un tiempo de respuesta inicial menor (alrededor de 40 segundos) pero tiene la desventaja de que inyecta ruido eléctrico al termopar, el cual tiene que ser compensado en la unión fría. El tercero es el *expuesto*, que tiene respuesta inicial rápida (menos de 15 segundos), pero ofrece poca protección al termopar. El último es el *expuesto de respuesta rápida*, cuyo tiempo es 2 segundos, pero ofrece aún menos protección al termopar. Estos tiempos aumentan o disminuyen de acuerdo a la longitud del termopozo.

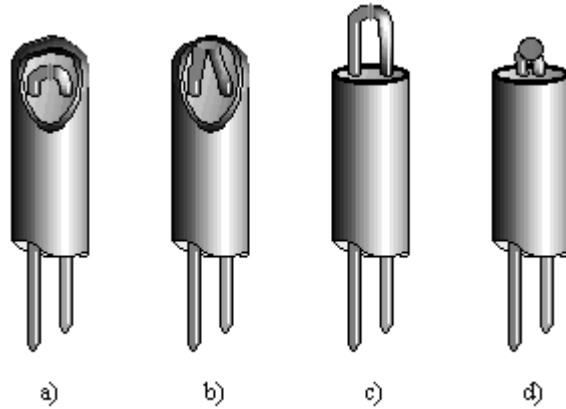


Figura A.3. Montaje del termopar. (a) Sellado y aislado del termopozo, (b) Sellado y aterrizado al termopozo, (c) expuesto, (d) Expuesto con respuesta rápida.

Compensación y linealización del termopar

Como se dijo anteriormente, el comportamiento del termopar es no lineal. De los termopares estandarizados, los tipos J y K son los que menor no-linealidad presentan, pero aún así es necesario someter su señal a tratamiento ya sea mediante hardware o software. La compensación, por su parte, se refiere a eliminar efectos físicos que se presentan cuando el termopar se une a un circuito, como lo son la interferencia eléctrica (que se supone ya debió haber sido al menos minimizada con un buen aislamiento) y la formación de un “nuevo termopar”, producto de unir los metales del sensor con las terminales de cobre (el material más común) del medidor, de hecho, la temperatura que mide ese nuevo termopar es la ambiental.

También se dijo con anterioridad que la compensación puede ser por hardware o por software. El principio es el mismo, basta con adherir un voltaje de corrección proporcional a la temperatura de la unión fría en la salida del termopar en el punto donde se une con el cobre del circuito electrónico, siendo que esta área debe ser isotérmica.

La solución por hardware es conceptualmente simple pero su implementación no lo es tanto, pues el circuito necesita ser diseñado para el termopar en específico. En la figura A-4 se observa la presencia del sensor de temperatura para la unión fría y un amplificador que lo escale para que pueda sumarse a la señal de salida. Lo que resta es hacer otra amplificación para adecuarlo al nivel de voltaje que maneje el circuito y la corrección de la no-linealidad.

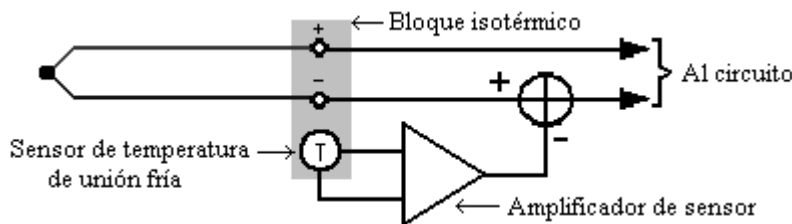


Figura A.4. Compensación de unión fría por hardware.

Por su parte la solución por software simplifica los requerimientos del circuito al eliminar el amplificador del sensor y el sumador, pero requiere que el algoritmo del software sea cuidadosamente escrito. Obviamente las tablas de calibración del termopar son importantes en este caso, pues con ellas se puede hacer la linealización del sensor.

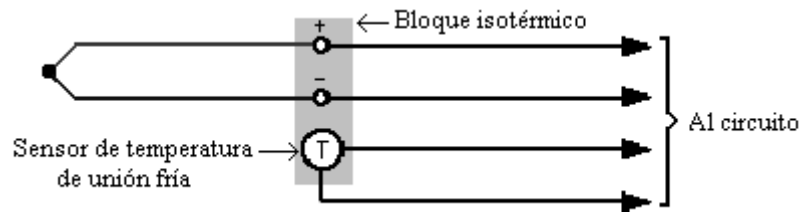


Figura A.5. Compensación de unión fría por software.

Cable de extensión para termopar

El hecho de que la temperatura de la unión fría y el dispositivo de lectura estén a la misma temperatura implica que físicamente deben estar cerca uno del otro. Cuando esto no es posible se emplea un cable de extensión para termopar cuya función es “recorrer” la unión, esto es, que el punto de medición puede estar lejos del termopar y por ende, de la atmósfera de medición. Los materiales con que se fabrican los extensores deben ser los mismos del correspondiente tipo de termopar.

A.2 Breve descripción de la electroválvula

La electroválvula (llamada también válvula solenoide) es un dispositivo electromecánico cuya función es permitir o interrumpir el paso de un fluido, como agua, gas, aire y combustibles líquidos. Emplea un solenoide para conmutar el estado de una válvula ya sea en abierto o en cerrado cuando éste se magnetiza al excitarse eléctricamente. En el mercado existen electroválvulas que operan tanto con corriente continua como con alterna, ajustables en medidas para las tuberías estándar y para distintos rangos de presión.

Clases y tipos de electroválvulas

Al igual que en conmutadores, hay dos clases de electroválvulas: las normalmente cerradas y las normalmente abiertas, aunque la mayoría de las existentes son de la primera clase. Por tipos se ha de entender cuántos puertos de conexión (llamados vías) tiene y cómo están dispuestos (a esto le llaman “geometría” de la válvula), hay una gran cantidad de ellos pero los más comunes son los siguientes:

1. Electroválvula (EV) de dos vías. Cuenta con dos puertos tal que cuando está energizado el solenoide deja pasar el fluido de un puerto al otro y cuando se retira la energía, el fluido cesa.
2. EV de tres vías. En un extremo tiene dos puertos y el restante está en el otro extremo. Cuando está energizado el solenoide, uno de los puertos está conectado a un puerto común y la presión fluye a través de él para extender un cilindro neumático. Cuando no está energizado, el otro puerto es ahora el que se conecta al puerto común y la presión es liberada para restablecer al cilindro. Esto vendría a ser el equivalente neumático de un relevador eléctrico de un polo con un tiro.
3. EV de cuatro vías. Tiene dos puertos en extremos opuestos. Cuando está energizado el solenoide, uno de los puertos está conectado de manera directa a uno de los puertos del extremo contrario, pero cuando se suspende la energía, la conexión se revierte.

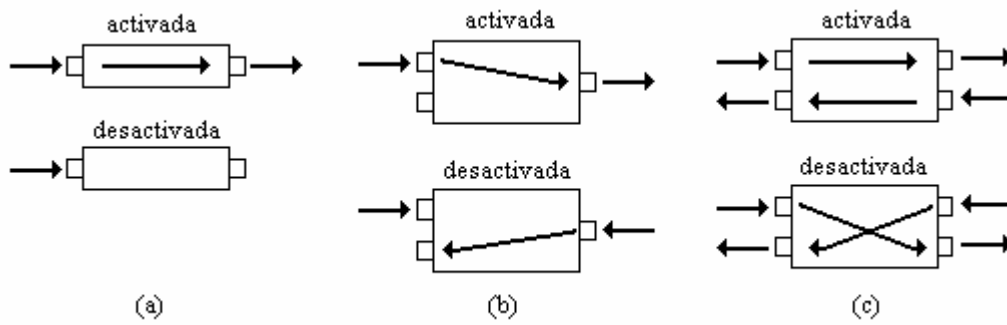


Figura A.6. Tipos de electroválvulas. (a) Dos vías, (b) Tres vías, (c) Cuatro vías.

Código de alambrado

En los EE.UU. el código de colores para el alambrado de electroválvulas que operan con corriente alterna (110 V) es el siguiente: si son tres alambres, dos de ellos deben ser rojos representando la fase y el neutro, el restante es verde y representa la tierra física; si son dos alambres, son de color rojo y de nueva cuenta, uno es la fase y otro es el neutro.

Apéndice B - CÓDIGO FUENTE

En vista de que el programa en PC está desarrollado en lenguaje gráfico, su inclusión en el presente documento no es pertinente, por ello se describirá únicamente el programa en MC.

B.1 Pseudocódigo

1. Zona de Reset (Inicio del programa):
 - 1.1 Declarar variables globales, entre ellos un pin dedicado a controlar la electroválvula al que será nombrado como VALVPIN ('0' = cerrado, '1' = abierto).
 - 1.2 Colocar a VALVPIN en estado cerrado.
 - 1.3 Inicializar el puerto serie (con temporizador 1) y el LCD.
 - 1.4 Configurar a los temporizadores 0 y 2 a 16 bits sin recarga.
 - 1.5 Configurar al vector de interrupción externa 0 para dar soporte al teclado.
 - 1.6 Habilitar interrupciones serial y externa 0.
 - 1.7 Habilitar la bandera de interrupción global e ir al paso 2.
2. Ciclarse indefinidamente hasta que una y sólo una de las interrupciones entre en acción, en dado caso, romper el ciclo e ir al paso 3.
3. Si la interrupción que entró en acción fue la del puerto serial, significa que la PC está iniciando el proceso y que toma el control del mismo. Entonces,
 - 3.1 Verificar una bandera llamada *Descargar?* cuya función es indicar si la PC desea descargar los datos existentes en la memoria del sistema y que corresponden a la última quema efectuada. Si la bandera está en 1, significa que la PC solicitó la descarga de los datos, por lo que se procederá a ejecutar la función DESCARGAR_DATOS, después de ello, regresar al paso 1. Si la bandera está en 0, irse al paso 3.2.
 - 3.2 Verificar si de la quema anterior hay datos disponibles que no es seguro tenga la computadora, esto lo indicará una variable llamada CLASEFIN cuando su valor sea diferente de 0 ('1' = El proceso fue terminado por el MC, '2' = El proceso fue abortado).
 - 3.2.1 Si los hay, avisar a la PC que hay datos disponibles en la memoria del sistema, los cuales no es seguro que tenga la PC. Formalmente se le pregunta a la PC si desea continuar con el inconveniente de perder los datos, o si prefiere terminar el programa para después descargar los datos. Si la PC responde que desea

- continuar, irse al paso 5. Si la PC responde que no desea continuar porque prefiere descargar los datos después, regresar al paso 1.
- 3.2.2 Si no hubo datos disponibles, ir al paso 5.
4. Si la interrupción que entró en acción fue la externa 0, significa que el mismo MC está iniciando el proceso y que toma el control del mismo. Entonces,
 - 4.1 Verificar si de la quema anterior hay datos disponibles en la memoria del sistema los cuales no es seguro tenga registrados la computadora (esto se presentará si en la quema anterior la comunicación entre PC y MC se interrumpió y por ende, fue necesario que el MC tomara el control del proceso). Si los hay, desplegar en el LCD un aviso de la situación y que si desea continuar el proceso con el riesgo de perder los datos. Esperar la respuesta del operador. Si el operador responde que no desea continuar, irse a paso 1. Si el operador responde que desea continuar a pesar de que se pierdan los datos, ir al paso 6.
 - 4.2 Si no hubo datos disponibles, ir al paso 6.
 5. Captura de parámetros de la quema vía serial.
 - 5.1 Deshabilitar la interrupción externa 0 y la interrupción serial, pues no es conveniente que se interrumpa la captura de datos.
 - 5.2 Apuntar al inicio de la zona de parámetros en la memoria de datos del sistema.
 - 5.3 En un ciclo indefinido, esperar dato desde la PC.
 - 5.3.1 Una vez recibido, almacenarlo en su localidad correspondiente de memoria y apuntar a la localidad siguiente. Nota: Se tiene ya establecido un orden de los parámetros, a saber, primero la temperatura objetivo, seguido por la presión inicial del tanque, hora de inicio, fecha y tasa de muestreo.
 - 5.3.2 En caso de que la espera del dato sobrepase el límite, irse al paso 6.
 - 5.3.3 Si el dato recibido es el token *Fin de parámetros*, romper ciclo e irse al paso 5.4.
 - 5.3.4 Si no es el token mencionado con anterioridad, regresar al paso 5.3
 - 5.4 Mediante una función, preparar al sistema para iniciar formalmente el proceso de quema.
 - 5.5 Habilitar la interrupción externa 0 e indicar en una variable llamada *Maestro* que la PC tiene el control del proceso. La razón de activar a la interrupción externa 0 es porque a través de ella y el teclado que soporta, el operador puede hacer uso de un botón de emergencia cuando éste lo crea conveniente, y su función es cerrar la electroválvula que alimenta de gas al horno. En otras palabras, es un botón de Aborto de Proceso.
 - 5.6 Ir al paso 7.
 6. Captura de parámetros de la quema vía teclado
 - 6.1 Deshabilitar la interrupción serial y habilitar la interrupción externa 0, pues no es conveniente que la interrupción serial entre en acción durante la captura de parámetros.
 - 6.2 A través del LCD y del teclado, pedir al operador que ingrese los parámetros de la quema en el siguiente orden, temperatura objetivo, nivel inicial del tanque estacionario, hora de inicio, fecha y tasa de muestreo.
 - 6.3 Mediante una función, preparar al sistema para iniciar formalmente el proceso de quema. En este caso, como el microcontrolador debe llevar la cuenta del tiempo, hay que configurar también a los temporizadores y a sus respectivos vectores de interrupción.

- 6.4 Habilitar la interrupción serial, mantener activada la interrupción externa 0 e indicar en la variable *Maestro* que el microcontrolador tiene el control del proceso. La razón de activar la interrupción serial es por si el operador decide que la PC retome el control del proceso; en dado caso, la PC hace uso de esta interrupción para avisar al MC del cambio de controlador. Por su parte, la razón de mantener activada a la interrupción externa 0 es, de nueva cuenta, por si el operador requiere de abortar la quema (idéntico que en 5.5).
7. Bifurcación del control.
 - 7.1 Si la variable *Maestro* indica que la computadora tiene el control, entonces hay que habilitar el teclado y deshabilitar los temporizadores y sus respectivos vectores de interrupción e irse al paso 8.
 - 7.2 En otro caso, *Maestro* indica que el microcontrolador tiene el control, entonces hay que habilitar la interrupción serial, la interrupción externa 0, los temporizadores y sus respectivos vectores de interrupción e irse al paso 9.
8. Control por computadora.
 - 8.1 Esperar hasta que la computadora envíe una orden. Si el tiempo de espera rebasa el límite establecido, indicar en la variable *Maestro* que el microcontrolador tendrá ahora el control, también habrá de indicar en la variable *Tipo de Fin del Proceso* que se ha interrumpido la comunicación con la PC (esto ayudará para indicar la siguiente sesión que hay datos disponibles en la memoria de datos, los cuales no es seguro que estén registrados en la PC). Irse al paso 7, donde la bifurcación se irá al paso 9.
 - 8.2 Si no hubo problemas, enviar de regreso la misma orden para que sirva como acuse de recibido.
 - 8.3 Decodificar y ejecutar la orden. Las órdenes se reflejan en funciones, las más importantes son dos: la primera es TOMAR_MEDICIONES, la cual captura un dato a través del convertidor analógico a digital de cada uno de los dispositivos que estén presentes en el sistema; la segunda es ALMACENAR_DATOS, que guarda las mediciones hechas en la zona de datos del sistema.
 - 8.4 Enviar el token *Comando ejecutado*, que indica la ejecución (exitosa) del comando y que se está listo para recibir la siguiente orden.
 - 8.5 Si la orden actual es una que indique el fin del proceso, colocar a 0 a VALVPIN e ir al paso 10. Si indica el aborto, colocar en la memoria de datos el token *enddata*, que especifica el fin de datos registrados, colocar a VALVPIN a 0, dar a CLASEFIN un valor de 2 indicando que el proceso fue abortado y por último, entrar en un ciclo infinito de inactividad, a la espera de un reset del MC o bien, del apagado del sistema.
 - 8.6 Si la orden actual no indica ni el fin ni el aborto del proceso, regresar al paso 8.1.
9. Control por microcontrolador. Para todos y cada uno de los pasos de este paso y sus subpasos está habilitada la interrupción serial, la cual, en caso de entrar en acción, será para regresar el control del proceso a la computadora. Lo que sucederá es que se ejecuta la función CAMBIO_DE_CONTROL, si esta función regresa un valor que indique la ejecución exitosa, regresar al paso 7, donde el control se bifurcará hacia el paso 8, pero si el valor devuelto indica que hubo problemas, habrá que continuar en el punto donde se haya dado la petición de cambio de control. Por otro lado, hay un proceso paralelo, ejecutado por uno de los temporizadores, el cual hace una llamada a la función ALMACENAR_DATOS, que consiste en almacenar el datos de cada dispositivo en un

periodo determinado por el parámetro tasa de muestreo, el cual se desactivará si el control del proceso regresa a la computadora. Otro proceso paralelo hace que cada media hora se envíe un aviso al operador mediante el LCD y mediante el buzzer para que incremente en una medida de 0.05 Kg/cm^2 la presión en las válvulas L y C (no aplica para el paso 9.2). Cada uno de los siguientes sub-pasos tiene un identificador llamado *Etapa*, el cual hace que se sepa la etapa exacta en la que se encuentra el proceso. Su utilidad radica en que si por ejemplo, la PC llevó el control hasta la sexta etapa y de repente perdió la comunicación con el MC, al darse el cambio de controlador y llegar al paso 7 (bifurcación de control), no es conveniente que el MC inicie desde la primera etapa, sino tiene que llegar a la etapa exacta en que se quedó la PC. Por otro lado, la interrupción externa 0 (el teclado) se habilita para que, si es necesario, el usuario oprima la secuencia de teclas 'A', '*' que indica el aborto de la operación, haciendo las mismas cosas descritas en 8.5.

- 9.1 Primera etapa. Es un ciclo que durará hasta que pase una de dos cosas: si la quema es de sancocho, al llegar a los 850°C se cerrará la electroválvula dando a VALVPIN un valor de 0 y se avisará al operador mediante el LCD y el buzzer que se ha llegado a la temperatura objetivo, después de ello se irá al punto 9.4. Si la quema es de esmalte, este ciclo se romperá al llegar a los 900°C y se irá al paso 9.2 Durante esta etapa forzosamente tuvieron que suceder dos temperaturas importantes, la inversión de la cristobalita (230°C) y la inversión del cuarzo (570°C); cuando se alcancen tales temperaturas se avisará mediante LCD y buzzer al operador.
- 9.2 Segunda etapa: Es un ciclo de 15 minutos de duración, donde la temperatura debe conservarse en el rango de 900 a 905°C . Si superase el rango, indicar mediante LCD y buzzer al operador que es necesario decrementarlo en 0.01 Kg/cm^2 la presión de las válvulas L y C, o bien, incrementarlas en la misma proporción si el caso es contrario.
- 9.3 Tercera etapa: Es un ciclo que durará hasta alcanzar la temperatura objetivo. Al llegar a ella se cerrará la electroválvula dando a VALVPIN un valor de 0 y se avisará al operador mediante el LCD y el buzzer que se ha llegado a la temperatura objetivo, después de ello se irá al punto 9.4.
- 9.4 Enfriamiento del horno: Una vez apagada la electroválvula, hay que seguir graficando a la misma tasa de muestreo hasta que la temperatura descienda a 600°C . Su utilidad es solamente para enriquecer el contenido de la gráfica.
10. Finalmente ciclarse indefinidamente hasta que el usuario apague el sistema.

Descripción de funciones principales.

Función TOMAR_MEDICIONES:

1. En un ciclo definido por el total de dispositivos a leer, hacer lo siguiente:
 - 1.1. Tomar la lectura del dispositivo haciendo uso del convertidor analógico a digital, el cual genera un dato de 12 bits, por lo que será necesario adaptarlo a 16 bits.
 - 1.2. Asignar una máscara al dato para saber después a qué dispositivo pertenece.
 - 1.3. Enviar el dato vía serial.
2. Terminar la función regresando un indicador de ejecución exitosa

Función ALMACENAR_DATOS:

1. Declarar un puntero a la memoria de datos del sistema
2. En un ciclo definido por el total de dispositivos a leer, hacer lo siguiente:
 - 1.4. Tomar la lectura del dispositivo haciendo uso del convertidor analógico a digital, el cual genera un dato de 12 bits, por lo que será necesario adaptarlo a 16 bits.
 - 1.5. Asignar una máscara al dato para saber después a qué dispositivo pertenece.
 - 1.6. Cada dispositivo tiene asignado un espacio en la memoria de datos del sistema, apuntar a su espacio y escribir el dato leído.
3. Terminar la función regresando un indicador de ejecución exitosa

Función DESCARGAR_DATOS:

1. Declarar un apuntador a la memoria de datos del sistema.
2. Apuntar primero a la zona de parámetros en su localidad de inicio
 - 2.1 En un ciclo delimitado por el total de parámetros, leer el dato señalado por el apuntador y enviarlo vía serial hacia la computadora.
 - 2.2 Para verificar que el dato fue recibido, esperar respuesta serial y comparar datos, en caso de que no sean iguales o bien, se exceda el tiempo límite de espera, interrumpir la función y regresar un valor que indique que hubo problemas. Si no hubo problemas en la verificación, apuntar a la siguiente localidad y regresar al paso 2.1.
 - 2.3 Enviar un valor que indique que se han mandado todos los parámetros.
3. Apuntar a la zona de acciones en su localidad inicial.
 - 3.1 En un ciclo indefinido, leer el dato señalado por el apuntador. Si dicho dato es un indicador del fin de la zona de acciones, romper el ciclo e ir al paso 3.3. Si no lo es, enviarlo vía serial hacia la computadora.
 - 3.2 Para verificar que el dato fue recibido, esperar respuesta serial y comparar datos, en caso de que no sean iguales o bien, se exceda el tiempo límite de espera, interrumpir la función y regresar un valor que indique que hubo problemas. Si no hubo problemas en la verificación, apuntar a la siguiente localidad y regresar al paso 3.1.
 - 3.3 Enviar un valor que indique que se han mandado todas las acciones.
4. Apuntar a la zona de datos en su localidad inicial.
 - 4.1 De esa zona, apuntar al inicio del espacio asignado al termopar.
 - 4.1.1 En un ciclo indefinido, leer el dato señalado por el apuntador. Si dicho dato es un indicador del fin del espacio asignado al termopar, romper el ciclo e ir al paso 4.2. Si no lo es, enviarlo vía serial hacia la computadora.
 - 4.1.2 Para verificar que el dato fue recibido, esperar respuesta serial y comparar datos, en caso de que no sean iguales o bien, se exceda el tiempo límite de espera, interrumpir la función y regresar un valor que indique que hubo problemas. Si no hubo problemas en la verificación, apuntar a la siguiente localidad y regresar al paso 4.1.4.
 - 4.2 Enviar un valor que indique que se han enviado todos los datos. Con el propósito de que en un futuro el sistema se expanda, el mismo esquema del punto 4.1 se puede repetir hasta 8 veces, de acuerdo a la estructuración de la memoria de datos.
5. Terminar la función regresando un indicador de ejecución exitosa de la función.

Función CAMBIO_DE_CONTROL

1. Declarar un puntero a la memoria de datos del sistema
2. Enviar un valor que indique la etapa en la que se encuentra el proceso. Para verificar si el dato recibido, esperar respuesta serial y comparar datos, en caso de que no sean iguales o

bien, se exceda el tiempo límite de espera, interrumpir la función y regresar un valor que indique que hubo problemas.

3. Enviar el momento exacto de la etapa en curso, es decir, el valor del temporizador. Para verificar el dato recibido, esperar respuesta serial y comparar datos, en caso de que no sean iguales o bien, se exceda el tiempo límite de espera, interrumpir la función y regresar un valor que indique que hubo problemas.
4. Terminar la función regresando un indicador de ejecución exitosa

B.2 Código fuente

El código fuente del programa en el MC fue desarrollado con la plataforma Proview32 versión 3.3 y tiene la siguiente estructura:

- Nombre del proyecto: Horno.prj
- Archivos cabecera: Sistema.h. Contiene definiciones de variables globales y funciones.
- Archivos componentes:
 - Main.c. Es el cuerpo principal del programa, incluye expresamente el algoritmo de la quema tratado en la sección 4.1.2.
 - Rutina1.c. Contiene funciones menores, tales como escritura a puerto serial, escritura de mensajes a LCD, etc.
 - Rutina2.c. Contiene otras funciones menores, tales como adquisición de datos desde el convertidor A/D, escritura a EEPROM, etc.
- Tamaño de código hexadecimal: 6.3 KBytes

Contenido de Sistema.h

```

/*****
/*****
/****  S I S T E M A . H      ****
/**** Archivo que contiene todas las definiciones del sistema
/****
/****
// **** Zona de declaración de cadenas ****

#define XBYTE ((unsigned char *) 0x20000L) // Para usar el DPT8

unsigned char code m_wcome[17] = {"Bienvenidos",0};

unsigned char code m_tquema[17] = {"Tipo de Quema",0};
unsigned char code m_tquema1[17] = {" Sch Esm",0};

unsigned char code m_temp[17] = {"Temperatura",0};
unsigned char code m_temp1[17] = {" 0000 °C",0};
unsigned char code m_pres[17] = {"Presion inic.",0};
unsigned char code m_pres1[17] = {" 00.00 Kg/cm2",0};
unsigned char code m_time[17] = {"Hora inic.",0};
unsigned char code m_time1[17] = {" 00:00 hrs",0};
unsigned char code m_date[17] = {"Fecha",0};
unsigned char code m_date1[17] = {"dd/mm/aa",0};
unsigned char code m_sam[17] = {"Muestreo",0};
unsigned char code m_sam1[17] = {" 00 min",0};
unsigned char code m_int[17] = {" 0000 ",0};
unsigned char code m_float[17] = {" 00.00",0};

unsigned char code m_hora[17] = {" 00:00",0};
unsigned char code m_fecha[17] = {"dd/mm/aa",0};
unsigned char code m_indice[17] = {" 00",0};
unsigned char code m_OK[4] = {"OK",0};
unsigned char code m_proc[17] = {" Proceso",0};
unsigned char code m_fin[17] = {" Terminado",0};
unsigned char code m_abort[17] = {"Abortado (E2)",0};
unsigned char code m_invcu[17] = {"Inv. cuarzo",0};
unsigned char code m_invcb[17] = {"Inv. cristob.",0};
unsigned char code m_invcv[17] = {"Inc. valv. C",0};
unsigned char code m_incvl[17] = {"Inc. valv. L",0};
unsigned char code m_downl[17] = {"M1: DATOS DISP.",0};
unsigned char code m_downl1[17] = {"Esp. Cont.",0};
unsigned char code m_yesdw [17] = {"DESCARGANDO...",0};
unsigned char code m_yesdw1[17] = {" HECHO!",0};
unsigned char code m_nodw [17] = {"M2: SEGURO?...",0};
unsigned char code m_nodw1[17] = {"SI NO",0};
unsigned char code m_espera[17] = {"Esperando...",0};
unsigned char code m_pasook[17] = {"ACCION oK",0};
unsigned char code m_error[8] = {"ERROR!",0};
unsigned char code m_nocomm[13] = {"E1: NO COM.",0};
unsigned char code menspasos[ 17 ] = {"Realizar pasos",0};
unsigned char code mens8910[ 17 ] = {" 8, 9 y 10.",0};
unsigned char code menspaso11[17] = {" Paso 11",0};
unsigned char code menspaso12[17] = {" Paso 12",0};
unsigned char code menspaso13[17] = {" Paso 13",0};

```

```

//*****
//*** DECLARACIÓN DE FUNCIONES
//*****
// *****
// Estas son las funciones contenidas en el archivo RUTINA1.CPP
// *****
void INITSER (void); // Inicializa el puerto serial
void INITLCD(void); // Inicializa el LCD 16X2
// Imprime en LCD una cadena void
IMPRIME_LCD(unsigned char code *);
void LCD_COM(unsigned char ); // Envía un comando al LCD 16x2
void LCD_DATA(unsigned char ); // Envía un dato al LCD 16x2
void DELAY100(void); // Realiza un retardo de 100 ms
void DELAY1MS(void); // Realiza un retardo de 100 ms
// Escribe un dato 16-b en la EEPROM
void WREEPROM16(unsigned int , unsigned char xdata * );
// Lee un dato 16-b en la EEPROM
unsigned int RDEEPROM16(unsigned char xdata *);
// Escribe una palabra en la zona de datos del sistema
void WR_PAR16(unsigned int );
void EMIT_SOUND(void ); // Señal audible a 300 Hz
void SENDWORD(unsigned int ); // Envía una palabra vía serial
// Recibe una palabra vía serial con tiempo de espera
unsigned int RXWORDTO(unsigned char);
// Realiza la conversión de float a BCD y lo imprime en el LCD,
//formato %.2f
void FLT2BCDPRN (unsigned int);
// Captura un dato desde teclado.
unsigned int SCANF2 (unsigned char );

// *****
// Estas son las funciones contenidas en el archivo RUTINA2.CPP
// *****
// Realiza una conversión A/D
unsigned int ADC_CN(unsigned char);
// Descarga de datos desde memoria de datos
unsigned int DOWNLOAD(void);
// Rutina para cambio de controlador durante el proceso
unsigned char SWITCHMTR(void);
// Decodificación y ejecución de algún comando desde PC
void DECINSTR(unsigned int);
// Almacena información en la memoria de daos del sistema
void STRAMDATA (void);
void SENDSAMPLE(void); // Envía información a la PC
// Prepara al sistema para iniciar el proceso
void INITPROCESS(void);
// Prepera al sistema para terminar el proceso en forma normal
void ENDPROCESS (void);
// Prepera al sistema para terminar el proceso en forma anormal
void ABORTPROCESS (void);
// Obtiene lectura de temperatura desde termopar y la
//imprime en el LCD. Solo se emplea cuando el MC
//tiene el control del proceso.
unsigned int GET_TEMP(void);

//*****
//*** VARIABLES Y BITS GLOBALES
//*****
//***** VARIABLES ***
// Variable que contiene la temperatura objetivo del proceso
unsigned int TEMPOBJETIVO;
// Dato serial. Solo se emplea cuando el vector de
//interrupción SC está activo
unsigned char SCDAT;
// Registro 16-b para temporización (cuando el MC tiene el
//control)
unsigned int TIMER;
// Registro 16-b para temporización de almacenamiento de
//datos (cuando el MC tiene el control)
unsigned int TIMERSM;
// Contiene el código de la tecla presionada (atendida en el
//vector de interrupción 0)
unsigned char TECLA;

// Esta variable contiene el estado o parte del proceso en
//que se encuentra.
unsigned char ETAPA;

// *** BITS ***
// Bandera bit que indica el inicio de la operación
bit INI_FLAG;
// Bandera bit que indica quién tiene el control de la operación
// tal que: 0=PC, 1=MC.
bit MASTER;
// Bandera que indica la presión de una tecla o bien, la
//disponibilidad de un dato 16-b en el buffer serial
bit CAPFLAG;
// Bandera bit que indica la finalización de un retardo de x-
//minutos
bit TIMERFLAG;

// (Nota: Todis estos bits pertenecen al semento de direccionamiento por
bits de la RAM interna del 80C32)

//////// PINES EXTERNOS //////////
// Aquí se declaran bits que tienen relación directa con un pin físico
//de algunos de los puertos del 80C32

sbit SOUNDPIN = P1 ^ 7;
sbit VALVPIN = P1 ^ 6;
//*****

//***** DEFINICIONES GLOBALES
//*****
// -----
// Direccionamiento en memoria de parámetros de la quema
// -----
// Dirección inicial para almacenamiento de parametros
#define STPAR 0x8000
// Dirección inicial para almacenamiento de acciones
#define STACC 0x8040
// Dirección donde se almacena la temperatura objetivo
//del sistema
#define TEMPOBJ 0x8000
// Dirección donde se almacena el nivel de presión inicial
//del sistema
#define NIVPRES 0x8002
// Dirección donde se almacena la hora de inicio
#define HORAINI 0x8004
// Dirección donde se almacena la fecha de la quema
#define FECHA 0x8006
// Dirección donde se almacena la tasa de muestreo del
//CAD
#define TASAMT 0x8008
//Dirección donde se almacena la indicación de qué
//clase de finalización tuvo la quema. Los valores son:
// 0 - Finalización exitosa, donde la PC recabó todos los
//datos
// 1 - Finalización exitosa, donde no se asegura que la
//PC haya recabado todos los datos
// 2 - Finalización no exitosa, donde el proceso tuvo que
//ser abortado
#define CLASEFIN 0x800A
// Dirección donde se almacena el puntero de memoria
//Nota: este parámetro es invisible para la PC.
#define POINTER 0x800C

// -----
// Definicion de direcciones para dispositivos a ser leídos por el
// CAD
// -----
// Dirección RAM para zona de datos del termopar 0
#define tc0addr 0x8810
//NOTA: En futuras expansiones del sistema se irán ocupando
//otras locaciones. Para ese momento, la diferencia entre cada
//dirección es de 450 palabras

```

```
// -----
// Constantes en general
// -----

#define abort_key 0x07 //Decodificación de la tecla Abortar
#define tot_param 0x06 //Total de parámetros a almacenar
#define tot_disp 0x01 //Total de dispositivos en el sistema
#define trefmx 0x04E2 // Referencia máxima para
//temperatura objetivo
#define trefmn 0x00E6 // Referencia mínima para
//temperatura objetivo
#define quarzinv 573 // Valor en °C referente a la inversión
//del cuarzo
#define cristoinv 273 // Valor en °C referente a la inversión
//de la cristobalita
#define sancocho 850 // Valor en °C referente a la
//temperatura de cocción del
//sancocho
#define tempreposo 0x0384 // Valor en °C referente a la
//temperatura reposo
#define reposomax 900 // Valor en °C referente al valor
//máximo de la temp. de reposo
#define reposomin 904 // Valor en °C referente al valor
//mínimo de la temp. de reposo
#define min30 0x62E2 // Valor para temporización de 30
//min.
#define min15 0x3171 // Valor para temporización de 15
//min.
#define abierto 1 // Estado abierto para la electroválvula
#define cerrado 0 // Estado cerrado para la electroválvula
#define tm_disp 0x2D // Para la temporización de
//despliegue en LCD (cuando MC
//tiene el control)
#define ini_dnw 0x32 // En el inicio de sesión, para revisar
//si la PC envió una orden de
//descarga de datos.

// -----
// Valores de retorno durante comunicación serial y constantes
//relacionadas
// -----
#define endaction 0xFFFD // Para indicar el fin de datos en
//zona de acciones
#define endpar 0xFFFC // Para indicar el fin de datos en zona
#define enddata 0xFFFB // Para indicar el fin de datos en zona
//de datos respaldados.
#define pfcomok 0xFFFE // Referencia para PC - indica la
//ejecución exitosa del comando
#define ER_COMM 0xFFFF // Referencia para MC - indica
//cuando se ha perdido la
//comunicación con la PC
#define bd2400 230 // Baud Rate de 2400 bps (cristal de
//12 MHz)

// -----
// Máscaras y referencias
// -----
#define tipcmask 0x8000 // Para usarse con ANL en dato serial
#define dispmsk 0x0700 // Para usarse con ANL en dato serial
#define dispmski 0x00F0 // Para usarse con ANL en dato serial
//(invertido)
#define accionmsk 0x003F //Para usarse con ANL en SCDAT
//(5 bits)
#define accionmk 0x8000 //Para usarse con ORL en SCDAT (5
//bits)
// Nota: Estas referencias para la acción a realizar (byte bajo del
//dato serial) deben ser filtradas con accmsk (cuando la PC tiene el
//control)

#define initmision 0x00 // Para iniciar la sesión
#define abortproc 0x1F // Para finalizar el proceso
#define endproc 0x1E // Para abortar el proceso
#define muestra 0x01 // Para tomar muestras y enviarlas a la
//PC
```

```
#define newstage 0x04 // Para indicar que se ha pasado a la
//próxima etapa
#define download 0x06 // Para descargar datos del proceso
//anterior
#define numstage 0x07 // Para preguntar por la etapa en
//desarrollo.
#define openvalve 0x08 // Para abrir la válvula
#define closevalve 0x0A // Para cerrar la válvula
// Nota: Estas referencias para la acción a realizar (byte bajo del
//dato serial) Deben ser filtradas con accmsk

// Pero cuando el MC tiene el control este valor se utiliza íntegro:
#define changemtr 0x39 // Para indicar el cambio de maestro
//de parámetros

//*****
//*** DEFINICIONES PARA LA CAPTURA DE DATOS DESDE
//TECLADO (SCANF)
//*****

// *** Direcciones
#define LCDDAT 0x6100 // Dirección XRAM del LCD para envío
//de datos
#define LCDCOM 0x6000 //Dirección XRAM del LCD para envío
//de comandos
#define TECLADO 0x7000 //Dirección XRAM del teclado para
//lectura

// *** Constantes
#define up_key 0x0E // Decodificación de la tecla Up
#define bkwd_key 0x0D // Decodificación de la tecla Backward
#define fcap_key 0x0B // Decodificación de la tecla Fin de
//captura (u OK)
#define er_key 0x10 // Decodificación de la tecla Erase
#define integer 0x00 // Bandera que indica tipo de captura
//entera
#define floating 0x01 // Bandera que indica tipo de captura
//flotante
#define time 0x02 // Bandera que indica tipo de captura
//time
#define date 0x03 // Bandera que indica tipo de captura
//date
#define yesnomb 0x04 // Bandera que indica tipo de captura
//para master
#define yesno 0x05 // Bandera que indica tipo de captura
//para master
#define index 0x06 // Bandera que indica tipo de captura
//para tipo índice
#define day 0x07 // Bandera que indica tipo de captura
//day
#define month 0x08 // Bandera que indica tipo de captura
//month
#define year 0x09 // Bandera que indica tipo de captura
//year

#define pos0 0xC7 //Posición 0 en el LCD
#define pos1 0xC6 //Posición 1 en el LCD
#define pos2 0xC5 //Posición 2 en el LCD
#define pos3 0xC4 //Posición 3 en el LCD
#define pos4 0xC3 //Posición 4 en el LCD
#define pos5 0xC2 //Posición 5 en el LCD
#define pos6 0xC1 //Posición 6 en el LCD
#define pos7 0xC0 //Posición 7 en el LCD
#define borrarlcd 0x01 //Orden de borrar el LCD
#define yes 0 //Opción sí (Cuando el tipo de captura sea
//"yesnomb")
#define no 1 //Opción no "
#define maybe 2 //Opción "tal vez". "

//*****
//*** DEFINICIONES PARA LA OPERACION DEL ADS774
// (TEXAS INSTRUMENT)
//*****
/////// Direcciones /////
#define adtsel 0x4000 // Mapeo del multiplexor analógico
#define adconv 0x5000 // Mapeo para orden de conversión
#define adcrdh 0x5100 // Mapeo para lectura del
#define adcrdl 0x5300 // Mapeo del multiplexor analógico
```

```
// Referencias para los dispositivos que se van a muestrear
//(canales analógicos)
#define termopar0 0x05 //Canal 0
#define canal1 0x03 //Canal 1 Disponible
#define canal2 0x01 //Canal 2 Disponible
#define canal3 0x07 //Canal 3 Disponible
#define canal4 0x00 //Canal 4 Disponible
```

```
##define canal5 0x06 //Canal 5 Disponible
##define canal5 0x02 //Canal 6 Disponible
##define canal7 0x04 //Canal 7 Disponible

// Máscaras para los dispositivos (para usarse con ORL)
#define termopar0msk 0x2000 //Máscara para termopar
```

B.2 Contenido de Main.c

```
#####
//*** M A I N . A S M ***
//***
//*** PROGRAMA PRINCIPAL DEL SISTEMA DE MONITOREO Y
// CONTROL) PARA EL
// PROCESO DE QUEMA DEL HORNO DE GAS DE ALTA
// TEMPERATURA

//
//*** DISPOSITIVO UTILIZADO: MC 80C32 (INTEL), DE LA
// FAMILIA MCS-51
// Versión del software: 1.3
#####
//#####
#include <reg52.h>
#include "sistema.h"
#include "rutina1.c"
#include "rutina2.c"

//#####
//#####
//#####
//#####
void main (void)
{
    unsigned int parametro=0;
    unsigned char indice=0, aux1=0, aux2=0 ;
    unsigned char xdata *DIRDAT; // Puntero general a memoria

    //*** ZONA DE RESET ***

    VALVPIN = cerrado;
    INI_FLAG=0; // La bandera bit de inicio se coloca a 0
    DNW_FLAG=0; // La bandera bit de descarga se coloca a 0
    INITSER(); // Inicializa la comunicación serial
    INTLCD(); // Se inicializa al LCD 16x2
    TMOD = TMOD & 0xF0; // Configuración del Timer 0 para
    TMOD = TMOD | 0x01; //operar como timer de 16-b
    T2CON = 0x00; // El timer2 se configura como timer de 16-b

    // Nota: El timer0 se encarga del conteo de media hora y el timer2
    //se encarga de temporizar el almacenamiento de datos en la
    //EEPROM

    EX1 = 0; // Se deshabilita la interrupción externa 1.
    ET0 = 0; // Se deshabilita la interrupción ov del timer 0
    ET2 = 0; // Se deshabilita la interrupción ov del timer 2
    TR0=0; // Se detiene al Timer 0
    TR2 = 0; // Se detiene al Timer 2
    IT0=1; // Se configura INT0 por flanco de bajada
    RI = 0; // Se limpia la bandera RI
    TI = 0; // Se limpia la bandera TI

    //Esto es temporal. Se indica que no hay acciones
    WREEPROM16(endaction, STACC);
    EA =1; // Se habilita la interrupción global

    // Recordar que la característica del sistema es que el controlador
    //primario es la PC, teniendo al MC como unidad de adquisición de
    //datos y accionador, con la posibilidad de actuar como controlador
    //secundario en caso de que la PC falle.
```

```
// Lo primero que hay que hacer es colocarse en modo de espera
//hasta que una de dos cosas pasen:
// 1. Que la PC establezca comunicación vía serial.
// 2. Que el operador inicie el proceso directamente con el MC vía
//interrupción externa 1.
// La espera consiste en revisar el valor de la bandera INI_FLAG,
//la cual indica que se ha iniciado la sesión.
main0_0: //Etiqueta de referencia

ES=1; // Se habilitan interrupciones serial y externa 0
EX0=1; //con el fin de esperar una respuesta

while (INI_FLAG==0) // Mientras la sesión no sea iniciada, el
{ ; } //valor de INI_FLAG es 0. Cuando se
//inicie, la bandera cambia su valor a 1. A
//PARTIR DE ESTE MOMENTO,
//INI_FLAG NO DEBE CAMBIAR DE
//VALOR HASTA QUE SE TERMINE LA
//SESIÓN DE FORMA REGULAR.

/* Ahora revisamos la bandera DNW_FLAG, si ésta es igual a
1, significa que la PC pidió descargar datos. */

if (DNW_FLAG==1)
{
    LCD_COM(0x80); // Se despliega en LCD que se inicia la
//descarga de datos
    IMPRIME_LCD(m_yesdw);
    SENDWORD (0x8006); // Se envía token de descarga.
    parametro = DOWNLOAD(); // === Función de descarga de
//datos ===

// Si la función devuelve algo diferente
//de 0, hubo problemas en la comm.

if (parametro != 0)
{ LCD_COM(0x80); // En dado caso, indicamos al usuario
//por LCD y por alarma.
    WREEPROM16(0x01, CLASEFIN); // Colocamos a la
//bandera CLASEFIN el valor de 0.
    IMPRIME_LCD(m_error); // Imprimimos mensaje de error
    LCD_COM (0xC0);
    IMPRIME_LCD(m_nocomm); }
else
{ LCD_COM(0x80); // Una vez terminado, imprimimos
//mensaje de descarga terminada en el LCD.
    IMPRIME_LCD(m_yesdw1);
    WREEPROM16(0x00, CLASEFIN); // Colocamos a la
//bandera CLASEFIN el valor de 0.
    INI_FLAG = 0; } // Reseteamos el valor de INI_FLAG.
    DNW_FLAG=0; // Igualmente de DNW_FLAG
    goto main0_0; } // Y nos vamos a goto main0_0

// Si MASTER == 0, el control lo tiene la PC
if (MASTER==0) // Revisamos la bandera CLASEFIN, en caso
//de valer distinto de 0, indica
{ ES = 0; //que de la sesión anterior posiblemente la
//PC no tiene todos los datos.
    EX0 = 0; // De momento nos esperamos
    parametro = RDEEPROM16(CLASEFIN);
    if (parametro != 0x0000)
    { //inicio de if parametro!=0
        parametro = download | accionmk;
        SENDWORD(parametro); // Entonces, lo primero que
//enviaremos a la PC es el token "download"
```

```

//indicando que hay datos disponibles para descargarse
parametro = RXWORDTO(2); // Y esperamos la
//respuesta de la PC con timeout de 1 minuto
if (parametro == ER_COMM) // Si recibimos un token
//ER_COMM como respuesta, significa que se
//cortó la comunicación,
{
INI_FLAG = 0; //Entonces reseteamos bandera
//INI_FLAG, imprimimos mensaje
LCD_COM (borrarlcd); //correspondiente en el LCD
//y nos regresamos a etiqueta
LCD_COM (0x80); //main0_0.
IMPRIME_LCD(m_error);
LCD_COM (0xC0);
IMPRIME_LCD(m_nocomm);
goto main0_0; }
/* Pero si no hubo problemas en la recepción y si
recibimos como respuesta el token initmision,
simplemente desechamos los datos. Brincando a la
etiqueta PARAM_SER */
parametro = parametro & accionmsk;
if (parametro == initmision)
goto PARAM_SER;
else
goto main0_0; // Si el token no fue initmision,
//brincamos a main0_0
} // Fin de if parametro!=0

/* En caso de que el tipo de fin de la sesión anterior haya
sido exitoso, iniciamos normalmente la sesión (enviando
el token InitMision) y brincamos a etiqueta
PARAM_SER.*/
else
{ parametro = initmision | accionmk;
SENDWORD(parametro);
goto PARAM_SER; }
// Fin del "if MASTER==0"

/* Ahora bien, que si MASTER es igual a 1, significa que el que inició
la sesión fue el mismo MC y por ende, tiene el control, entonces lo
primero que se hará es deshabilitar la interrupción serial y habilitar
la interrupción externa 0 (teclado). */

if (MASTER == 1)
{ ES = 0;
EX0=1;
LCD_COM(borrarlcd);
parametro = RDEEPROM16(CLASEFIN);
/* De nueva cuenta revisamos el valor de la bandera
CLASEFIN. Si tiene valor distinto de 0, desplegamos en
el LCD un aviso de que hay datos de respaldo
disponibles que posiblemente no tenga la PC. Entonces
pedimos elegir entre esperar comunicación con PC para
descargar o continuar con la sesión. Si respondió sí,
imprimimos el mensaje de espera y brincamos a la
etiqueta "main0_0", no sin antes regresar la bandera
INI_FLAG a su valor original. Si respondió no, brincar a
etiqueta PARAM_TEC. */
if (parametro != 0x00)
{ LCD_COM(0x80);
IMPRIME_LCD(m_downl);
LCD_COM(0xC0);
IMPRIME_LCD(m_downl1);
parametro = SCANF2(yesno);
if (parametro = yes)
{ LCD_COM(borrarlcd);
LCD_COM(0x80);
IMPRIME_LCD(m_espera);
INI_FLAG = 0;
goto main0_0; }
else
goto PARAM_TEC; // Ir a la captura de parámetros de
//la quema por teclado (MC).
}
else // Pero si CLASEFIN=0, simplemente brincamos a
goto PARAM_TEC; //PARAM_TEC
// fin del "if MASTER==1"

```

```

/*
En esta zona parte se procederá a capturar los parámetros de la
quema desde teclado (vía interrupción externa 0)
*/
PARAM_TEC:
ES=0; // Se deshabilita interrupción serial temporalmente
DELAY100());

partec_to: // Comenzamos pidiendo la temperatura objetivo
LCD_COM(borrarlcd); // Se despliega el mensaje
LCD_COM(0x80); // correspondiente
IMPRIME_LCD (m_temp);
LCD_COM(0xC0);
IMPRIME_LCD (m_temp1);
parametro = SCANF2(integer); // Obtenemos dato.
/* Ahora verificaremos que el dato, temperatura objetivo,
esté dentro de los límites del horno. Si lo está, lo
almacenamos en su localidad correspondiente. Si no lo
está, volvemos a pedir la captura del dato, regresando
a la etiqueta partec_to */
if (parametro > 573 && parametro < 1200)
WREEPROM16(parametro, TEMPOBJ);
else
goto partec_to;

partec0_2: // Ahora pediremos el nivel actual del tanque.
LCD_COM(borrarlcd); // despliega el mensaje correspondiente
LCD_COM(0x80);
IMPRIME_LCD (m_pres);
LCD_COM(0xC0);
IMPRIME_LCD (m_pres1);
parametro = SCANF2(floating); // Obtenemos dato
WREEPROM16(parametro, NIVPRES); // Y lo almacenamos en
//su localidad correspondiente

// Ahora pedimos la hora de inicio.
LCD_COM(borrarlcd); // Desplegar el mensaje correspondiente.
LCD_COM(0x80);
IMPRIME_LCD (m_time);
LCD_COM(0xC0);
IMPRIME_LCD (m_time1);
parametro = SCANF2(time);
WREEPROM16(parametro, HORAINI);

/* Luego la fecha por partes, primero se pide el día, luego el
mes y por último el año.*/
LCD_COM(borrarlcd);
LCD_COM(0x80);
IMPRIME_LCD (m_date);
LCD_COM(0xC0);
IMPRIME_LCD (m_dia);
parametro = SCANF2(index);
indice = parametro; //día
indice = indice & 0x1F;
LCD_COM(borrarlcd);
LCD_COM(0x80);
IMPRIME_LCD (m_date);
LCD_COM(0xC0);
IMPRIME_LCD (m_mes);
parametro = SCANF2(index);
aux1 = parametro; //mes
aux1 = aux1 & 0x0F;
LCD_COM(borrarlcd);
LCD_COM(0x80);
IMPRIME_LCD (m_date);
LCD_COM(0xC0);
IMPRIME_LCD (m_year);
parametro = SCANF2(index);
aux2 = parametro; //año
aux2 = aux2 & 0x1F;

// Y adaptamos los valores a un dato único de 16 bits
if (aux1 >= 0x07) //si mes es mayor que 0x07
indice = indice | 0x80; //Colocamos a 1 su MSB. "Indice" tiene
//la parte alta del parámetro
aux1 = aux1 << 0x05 ;
aux2 = aux2 | aux1; // aux2 tiene la parte baja del parámetro
parametro = indice;
parametro = parametro << 0x08;

```

```

parametro = parametro | aux2;
// Y lo almacenamos en su correspondiente locación de
//memoria.
WREEPROM16(parametro, FECHA);

// Y por último pedimos la tasa de muestreo. Como el datos se
captura a modo de índice. Se verifica que el dato esté en el rango de
1 a 10, si lo está, se almacena en su respectiva locación de memoria,
si no lo está, se vuelve a pedir la tasa de muestreo)
partec_sm:
LCD_COM(borrarlcd); // Se manda borrar el LCD
LCD_COM(0x80);
IMPRIME_LCD (m_sam);
LCD_COM(0xC0);
IMPRIME_LCD (m_sam1);
parametro = SCANF2(index);

if (parametro>0 && parametro <20)
{ if (parametro == 1)
  parametro = 0x0393;
  if (parametro == 2)
  parametro = 0x0726;
  if (parametro == 3)
  parametro = 0x0AB9;
  if (parametro == 4)
  parametro = 0x0E4C;
  if (parametro == 5)
  parametro = 0x11DF;
  if (parametro == 6)
  parametro = 0x1572;
  if (parametro == 7)
  parametro = 0x1905;
  if (parametro == 8)
  parametro = 0x1C98;
  if (parametro == 9)
  parametro = 0x202B;
  if (parametro == 10)
  parametro = 0x23BE;
  WREEPROM16(parametro, TASAMT); }
else
  goto partec_sm;

CAPFLAG = 0;
MASTER = 1;
INITPROCESS(); // Ahora se preparan al sistema para
//iniciar formalmente el proceso.
goto PREINICIO; // Y después de ello se brinca a la etiqueta
//PREINICIO

/*
Por otro lado, si el usuario envió la señal desde la PC, asumimos que
ésta última tienen el control, por tanto, el procedimiento de captura
será acorde a los recursos del ordenador.
Nota: En el programa de PC debe existir un pequeño retardo para
compensar el tiempo que se tarda el microcontrolador en llegar a esta
parte del programa.
*/
PARAM_SER:
EX0=0; // Se deshabilita INT0 y la interrupción serial
ES = 0;
DIRDAT = STPAR; // Se apunta al inicio de la zona de
//parámetros en el sistema

/*En un ciclo definido por la recepción serial del token endpar, se
esperará el dato correspondiente con un timeout de 5 segundos, si el
dato no llegó en ese tiempo, significa que se perdió la comunicación
con la PC y entonces se brincaré a la etiqueta PARAM_TEC, para
realizar la captura desde teclado e iniciar el proceso con el MC. Si no
se presenta problemas en la comunicación con la PC, se envía el
mismo dato a la PC como acuse de recibido y los parámetros se van
almacenando en su respectiva locación de memoria */
for(;;)
{
  parametro = RXWORDTO(2);
  if (parametro == ER_COMM)
  { LCD_COM (borrarlcd);
    LCD_COM (0x80);
    IMPRIME_LCD(m_error);
    LCD_COM (0xC0);
    IMPRIME_LCD(m_nocomm);

```

```

MASTER = 1;
goto PARAM_TEC; }
SENDWORD (parametro); //Enviar acuse de recibido
if (parametro== endpar) // Si el dato recibido fue el token
break; //endpar, rompemos el ciclo
/* Se almacena el dato. DIRDAT es un puntero
a la zona de parámetros del sistema y es el que
lleva el direccionamiento.*/
WREEPROM16(parametro, DIRDAT);
DIRDAT++;
DIRDAT++;
} // Aquí termina la captura de parámetros desde PC (vía serial)

CONTROL_PC:
parametro = RXWORDTO(1); // Esperamos respuesta de la PC
//con timeout de 5 segundos
if (parametro==ER_COMM) // Si se perdió comunicación,
{ MASTER=1; //la bandera MASTER se coloca a 1,
//indicando que el nuevo maestro es el MC
WREEPROM16(0x01, CLASEFIN); // Con una sola vez que
//pase esto, la PC no tiene la certeza de
//tener un registro completo, es por ello que
//CLASEFIN se coloca a 1.
EMIT_SOUND(); // Se da una señal audible
LCD_COM (borrarlcd); // Y se avisa por medio del LCD
LCD_COM (0x80); //
IMPRIME_LCD(m_error);
LCD_COM (0xC0);
IMPRIME_LCD(m_nocomm);
gotoPREINICIO; }

SENDWORD(parametro); // Si no hubo problemas,
//reenviaremos el comando como acuse de
//recibido
DECINSTR(parametro); // Se manda llamar rutina de
//decodificación y ejecución de instrucciones
SENDWORD(pfcomok); // Después de decodificada y
//ejecutada, se envía el token "pfcomok", que
//notifica a la PC el suceso y que se
//encuentra lista para el próximo comando.

parametro = parametro & accionmsk;

/* Si el comando actual es el de abortproc o el de
endproc, romper este ciclo brincando a la etiqueta
fin_PC. En otro caso, regresamos a la etiqueta
CONTROL_PC.*/
if (parametro == abortproc || parametro == endproc)
goto fin_PC;
else
goto CONTROL_PC;

fin_PC:
// Hay que incluir un mensaje en el LCD para el operador
EMIT_SOUND();
EMIT_SOUND();
EMIT_SOUND();
goto cicloinf;

```

```

/*
PROCESO DE QUEMA NETO CONTROLADO POR MC
*/

```

Por diseño del programa, éste fue dividido en nueve etapas, mismas que son invisibles para el operador. En la mayoría de las etapas se realizan tres tareas "simultáneas": el sensaje de la temperatura, un conteo de media hora al término del cual, se le envía al operador un aviso de que es necesario incrementar la presión en las válvulas L y C en 0.05 kg/cm² y el almacenamiento periódico de datos (en este caso, la temperatura) en la memoria de datos del sistema. La primer tarea se realiza con una periodicidad de dos segundos, la segunda, como ya se dijo, cada media hora y la última con la periodicidad marcada por la tasa de muestreo. El timer0 es el que se encarga de la segunda tarea y el timer2 de las tareas 1 y 3

La identificación de las etapas se da por medio de la variable ETAPA.

Nota: No olvidar que al dejarse activa la interrupción serial se da oportunidad a que la PC retome el control del proceso, por lo que en cada etapa se revisa si dicha interrupción ha provocado el cambio de controlador mediante el cambio de valor de la variable MASTER a 0.

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
*/
CONTROL_MC:

```

```

//*****
//** ETAPA UNO: Espera de 30 minutos, en donde se puede
//dar la inversión de la cristobalita.
//*****

```

```

if (ETAPA == 1)
{ // Principio Etapa 1
  TH0 = 0x00; // Se hacen los ajustes necesarios para
  TL0 = 0x00; //la temporización del retardo de media
  // hora a cargo del timer0
  TIMER=0x0000; //Reseteamos la variable TIMER, que
  //lleva la temporización del conteo de la
  //media hora
  TIMERDP=0x0000; // Se resetea la variable TIMERDP, que
  //lleva la temporización de lectura de
  //temperaturas cada dos segundos.
  TIMERFLAG = 0; // Bandera bit que indica cuándo se ha
  //cumplido el conteo de media hora
  TR0 = 1; // Activamos el timer 0

```

/*En esta etapa, como solo dura media hora, no se lleva el conteo del que se habló al principio.*/

```

while (TIMERFLAG==0)
{ // inicio de while mc0
  // ¿Se ha cumplido la temporización de 2 seg para
  //mostrar temperatura en LCD?
  if ( TIMERDP == tm_disp )
  { //inicio del if mc1
    TIMERDP = 0x00; //Primero reseteamos a variable
    //TIMERDP

    parametro = GET_TEMP(); // Obtiene la temperatura
    //y la despliega en LCD
    if (parametro >= cristoinv) //¿Es igual o mayor a la inv.
    { //de la cristovalita?
      LCD_COM(borrarlcd); // Se manda borrar el LCD
      LCD_COM(0x80);
      IMPRIME_LCD(m_invcb);
      EMIT_SOUND();
      STRAMDATA(); // Y que se almacene esta lectura
      //en la EEPROM
    }
  } //fin del if mc1

```

```

if (MASTER==0) // Si MASTER = 0, el control lo tiene
goto PREINICIO; //el MC. Brincamos a PREINICIO.

```

```

} //Fin del while mc0
TIMERFLAG = 0; // Se limpia la bandera de temporización
TR0 = 0; // Paramos al timer 0...
TIMER = 0x0000; // Se resetea a TIMER
TIMERDP = 0x0000; //También a TIMERDP
ETAPA = 0x02; // Actualizamos ETAPA a 2.
} // Fin Etapa 1

```

```

//*****
//** ETAPA DOS: Inmediatamente después que pasaron los
//30 min, indicar al operador por merdio del LCD que es
//necesario abrir la chimenea, cerrar la puerta y aumentar la
//presión de ambas válvulas en 0.05 kg/cm2.
//*****

```

```

if (ETAPA == 2)
{ // Principio Etapa 2
  LCD_COM(borrarlcd); // Se manda borrar el LCD
  LCD_COM(0x80);
  IMPRIME_LCD(m_etapa2);

```

```

LCD_COM(0xC0);
IMPRIME_LCD(m_etapa2_1);
EMIT_SOUND();
ETAPA = 0x03; // Una vez que se han desplegado los
//mensajes, pasamos a la etapa 3

```

```

if (MASTER==0) // Estas dos líneas sirven para saber
goto PREINICIO; //si la PC ha retomado el control del
//proceso. En dado caso, irse a PREINICIO.

```

```

} // Fin de Etapa2

```

```

//*****
//** ETAPA TRES: Ahora nos esperaremos indefinidamente
//hasta que se presente la inversión del cuarzo. Cuando eso se
//dé, avisamos al operador. Aquí sí se hace el conteo de media
//hora descrito anteriormente
//*****

```

```

if (ETAPA == 3)
{ // Principio Etapa3
  TH0 = 0x00; // Se hacen los ajustes necesarios para la
  TL0 = 0x00; // temporización del retardo de media hora.
  TIMERFLAG = 0;
  TIMERDP = 0x0000;
  TR0 = 1; // Activamos el timer 0
  for (;;)
  { // inicio de for indefinido 2

```

// ¿Se ha cumplido la temporización de 2 seg para

//mostrar temperatura en LCD?

```

if ( TIMERDP == tm_disp )
{ //inicio del if mc3_1
  TIMERDP = 0x00; //Primero reseteamos a variable
  //TIMERDP
  parametro = GET_TEMP();
  if (parametro = quazinv)
  { LCD_COM(borrarlcd); // Se manda borrar el LCD
    LCD_COM(0x80);
    IMPRIME_LCD(m_invcu);
    LCD_COM(0xC0);
    STRAMDATA(); // Y que se almacene esta lectura en
    //la EEPROM
    EMIT_SOUND();
    break; }
} // Fin del if mc3_1

```

```

} // Fin del if mc3_1

```

/* Aquí se hace el conteo de media hora. */

```

if (TIMERFLAG==1)
{ LCD_COM(borrarlcd); // Se manda borrar el LCD
  LCD_COM(0x80);
  IMPRIME_LCD(m_incrm1);
  EMIT_SOUND();
  TH0 = 0x00; // Se hacen los ajustes necesarios
  //para la temporización
  TL0 = 0x00; //del retardo de media hora.
  TIMER = 0x0000; //-
  TIMERFLAG = 0; //-
  TR0 = 1; // Activamos el timer 0
}

```

```

if (MASTER==0) // Si MASTER = 0, el control lo tiene
goto PREINICIO; //el MC. Brincamos a PREINICIO

```

```

} // Fin del for indefinido 2
TIMERFLAG = 0; // Se limpia la bandera de temporización
TR0 = 0; // Detenemos al timer 0
TIMER = 0x0000; //-
ETAPA = 0x04; // Una vez que se ha llegado a la inv. del
//cuarzo, pasamos a la etapa 4
} // Fin de Etapa 3

```



```

//*****
//** ETAPA CUATRO: Ahora esperaremos indefinidamente
//hasta que se alcance la temperatura de sancocho, entonces,
//se dará una de dos cosas: si la quema es de sancocho,
//terminamos el proceso, si es de esmalte, pasamos
//a la siguiente etapa.
// Aquí también se hace el conteo de media hora.
//*****

if (ETAPA == 4)
{ // Principio Etapa 4
  TH0 = 0x00; // Se hacen los ajustes necesarios para la
  TL0 = 0x00; // temporización de la media hora.
  TIMERFLAG = 0; //
  TIMERDP = 0x00; // reseteamos a variable TIMERDP
  TR0 = 1; // Activamos el timer 0

  for (;;)
  { // inicio de for indefinido 4
    if ( TIMERDP == tm_disp )
    { //inicio del if mc1_4
      TIMERDP = 0x00; //Primero reseteamos a variable
      //TIMERDP
      parametro = GET_TEMP();
      if (parametro >= sancocho )
      { //inicio del if 4_1
        if ( parametro >= RDEEPROM16(TEMPOBJ))
        {
          ETAPA = 8;
          ENDPROCESS();
          STRAMDATA(); // Y que se almacene esta lectura en
          //la EEPROM
          goto mc_etapa8; }
        else
        { STRAMDATA(); // Y que se almacene esta lectura en
          //la EEPROM
          break; }
      } // fin del if 4_1
    } // fin del if mc1_4

    if (TIMERFLAG == 1)
    { LCD_COM(borrarlcd); // Se manda borrar el LCD
      LCD_COM(0x80);
      IMPRIME_LCD(m_incr1);
      EMIT_SOUND();
      TH0 = 0x00; // Se hacen los ajustes necesarios para
      //la temporización
      TL0 = 0x00; //del retardo de media hora.
      TIMER = 0x0000; //
      TIMERFLAG = 0; //
      TR0 = 1; // Activamos el timer 0
    } // fin del if2

    if (MASTER==0)
      goto PREINICIO;

  } // fin de for indefinido 4

  TH0 = 0x00; //
  TL0 = 0x00; //
  TIMER = 0x0000; //
  TIMERFLAG = 0; //
  TR0 = 0; // Detenemos el timer 0
  ETAPA = 0x05; // Una vez que se ha llegado a la inv. del
  //cuarzo, pasamos a la etapa 5
} // Fin Etapa 4
//*****
//** ETAPA CINCO: Al llegar a esta etapa ya sabemos algo,
//la temperatura objetivo no es la de sancocho, sino una de
//esmalte, entonces, esperamos indefinidamente
//hasta que lleguemos a los 900 °C
//*****

if (ETAPA == 5)
{ // Principio Etapa 5
  for (;;)
  { // inicio de for indefinido 5

    if ( TIMERDP == tm_disp )
    { //inicio del if mc1_5

      TIMERDP = 0x00; //Primero reseteamos a variable
      //TIMERDP

      parametro = GET_TEMP();
      if (parametro >= tempreposo)
      { STRAMDATA(); // Y que se almacene esta lectura en
        //la EEPROM
        break; }
    } // fin del if mc1_5

    if (TIMERFLAG == 1) // En esta sección se hace el
    { //conteo de 30 minutos, al término
      LCD_COM(borrarlcd); //del cual se envía mensaje al
      //operador de subir la presión de
      LCD_COM(0x80); //ambas válvulas en 0.05 Kg/cm2.
      IMPRIME_LCD(m_incr1);
      EMIT_SOUND();
      TH0 = 0x00; //
      TL0 = 0x00; //
      TIMER = 0x0000; //
      TIMERFLAG = 0; //
      TR0 = 1; // Activamos al timer 0
    }

    if (MASTER==0)
      goto PREINICIO;
  } // fin de for indefinido 5

  TIMER = 0x0000; //
  ETAPA = 0x06; // Una vez que se ha llegado a los 900 °C,
  //pasamos a la etapa 6
} // Fin Etapa 5

//*****
//** ETAPA SEIS: Una vez alcanzado los 900 °C el cuerpo
//de quema tiene que pasar a un estado conocido como
//reposo. Esta etapa tiene duración de 15 minutos.
// Aquí, la temperatura tiene que permanecer en el rango de
//900 °C a 904 °C, en caso de estar mas allá de los 904°C, el
//usuario es avisado que debe reducir la presión en
//0.01kg/cm2 en ambas válvulas, o subirla en 0.01 kg/cm2 en
//caso contrario, o hacer nada si la temperatura se mantiene en
//el rango.
//*****

if (ETAPA == 6)
{ // Principio Etapa 6
  TH0 = 0x00; // Se
  TL0 = 0x00; //
  TIMERFLAG = 0; //
  TIMERDP = 0x0000;
  TR0 = 1; // Activamos el timer 0

  while (TIMERFLAG==0)
  { // inicio de while 6

    if ( TIMERDP == tm_disp )
    { //inicio del if mc1_6
      TIMERDP = 0x00; // Primero reseteamos a variable
      //TIMERDP
      parametro = GET_TEMP();
      if (parametro >= reposomax) // Si la temperatura es
      { LCD_COM(borrarlcd); //mayor a 904°C,
        LCD_COM(0x80); //mandar bajar la presión en
        IMPRIME_LCD(m_incr2); //0.01 kg/cm2.
        EMIT_SOUND(); }
      else
      if (parametro <= tempreposo) // Si la temperatura es
      { //menor a 900°C,
        LCD_COM(borrarlcd); //mandar subir la presión
        LCD_COM(0x80); //en 0.01 kg/cm2
        IMPRIME_LCD(m_dec1);
        EMIT_SOUND(); }
    } //fin del if mc1_6

    if (MASTER==0)
      goto PREINICIO;
  }
}

```

```

} // fin de while 6
TH0 = 0x00; //
TL0 = 0x00; //
TIMER = 0x0000; //
TIMERFLAG = 0; //-
TR0 = 0; // Detenemos el timer 0
ETAPA = 0x07; // Una vez que se ha terminado el tiempo
//de reposo, pasamos a la penúltima
//etapa del proceso
} // Fin Etapa 6

//*****
//** ETAPA SIETE: Nos esperamos indefinidamente a que la
//temperatura llegue a su valor objetivo, cuidando que la
//presión sea constante. En caso de que la presión se
//desbalancee, enviamos un mensaje al operador de ajustar la
//presión. Durante este tiempo indefinido, se hace un conteo de
//30 min., al término del cual se pide al usuario aumentar la
//presión de ambas válvulas en 0.05 Kg/cm2.
//*****

if (ETAPA == 7)
{ // Principio Etapa 7
  TH0 = 0x00; //
  TL0 = 0x00; //
  TIMER = 0x0000; //
  TIMERFLAG = 0; //-
  TR0 = 1; // Activamos al timer 0

  for (;;)
  { // inicio de for 7

    if (TIMERDP == tm_disp)
    { //inicio del if mc1_7
      TIMERDP = 0x00; //Primero reseteamos a variable
      //TIMERDP
      parametro = GET_TEMP();
      if (parametro == RDEEPROM16(TEMPOBJ))
      { LCD_COM(borrarlcd);
        LCD_COM(0x80);
        STRAMDATA(); // Y que se almacene esta lectura en la
        //EEPROM
        EMIT_SOUND();
        ENDPROCESS();
        break; } // En caso de alcanzar la temperatura objetivo,
        //rompemos este ciclo.
      } // fin del if mc_1_7

    if (TIMERFLAG == 1) // En esta sección se hace el
    { //conteo de 30 minutos, al término
      LCD_COM(borrarlcd); //del cual se envía mensaje al
      //operador de subir la presión de
      LCD_COM(0x80); //ambas válvulas en 0.05 Kg/cm2.
      IMPRIME_LCD(m_incr1);
      EMIT_SOUND();
      TH0 = 0x00; //
      TL0 = 0x00; //
      TIMER = 0x0000; //
      TIMERFLAG = 0; //-
      TR0 = 1; } // Activamos al timer 0

    if (MASTER==0)
      goto PREINICIO;

  } // fin de for 7

  ETAPA = 0x08; // Una vez que se ha alcanzado la
  //temperatura objetivo, pasamos a la etapa 9
  TH0 = 0x00; // Detenemos al timer0
  TL0 = 0x00; //
  TIMER = 0x0000; //
  TIMERFLAG = 0; //-
} // Fin Etapa 7

```

```

//*****
//** ETAPA OCHO: Ahora la válvula del gas debe ser
//cerrada, entonces se iniciará mandando la señal al relé para
//cerrarla, y como mera documentación se seguirá graficando
//la temperatura hasta que descienda a 600°C.
// Esta etapa se podría llamar "de enfriamiento".
//*****

mc_etapa8:
if (ETAPA = 0x08)
{ // Principio de Etapa8
  VALVPIN = cerrado; // Se cierra la válvula

  for (;;) // Se inicia el ciclo indefinido donde se sigue capturando
  { //la temperatura hasta que ésta descienda a los 600°C
    // inicio de for 8
    parametro = GET_TEMP();
    if (parametro <= 600)
    { LCD_COM(borrarlcd);
      LCD_COM(0x80);
      IMPRIME_LCD (m_cerrars);
      break; } // En caso de alcanzar la temperatura
      //objetivo, terminamos proceso

    if (MASTER==0)
      goto PREINICIO;

  } // fin de for 8

  ET2 = 1; // Se deshabilita la interrupción ov del timer 2
  TR2 = 1; // Se deshabilita al Timer 2 (Conteo de tasa de
  //almacenamiento)
  ETAPA = 0x09; // Una vez que se ha alcanzado la
  //temperatura objetivo, pasamos a la etapa 9

} // Fin de Etapa 8

//*****
//** ETAPA NUEVE: Esta no es una etapa en sí, pero sirve
//para que el MC sepa que ya pasó la etapa de enfriamiento (8)
//Aquí lo único que se hace es avisar al operador que el
//proceso ha terminado formalmente. Se envían tres avisos
//sonoros.
// Después se envía a un ciclo infinito que solo puede ser
//roto apagando el MC o bien, reseteándolo.
//*****

FINAL_OP:
ES = 0;
EX0 = 0;
LCD_COM(borrarlcd);
LCD_COM(0x80);
IMPRIME_LCD(m_proc);
LCD_COM(0x00);
IMPRIME_LCD(m_fin);
MASTER = 1;
ENDPROCESS(); // Finalmente terminamos la operación
EMIT_SOUND();
EMIT_SOUND();
EMIT_SOUND();

cicloinf:
goto cicloinf;

//*****
}
//** FIN DEL MAIN *
//*****

```

```

/* Y en lo que respecta a los vectores de interrupción, éstos se verán a
continuación*/
//-----
//**** VECTOR DE INTERRUPCIÓN EXTERNA 0
//**** ETIQUETA: CAPTURA_INT
//**** Este vector sirve para recoger datos desde el teclado, ya
// sean parámetros de la quema u órdenes (como abortar
// proceso, etc).
//-----
void INTCAP (void) interrupt 0
{
  if (INI_FLAG==0) // Si la bandera bit de inicio está a 0,
  {
    //significa que el sistema está por arrancar
    INI_FLAG=1; //Entonces, INI_FLAG se coloca a 1 y nos
    //vamos a break_tec
    MASTER=1; //Nota: INI_FLAG no debe ser regresado a
    //cero durante la operación.
    WREEPROM16 (0x01, CLASEFIN);
    goto salir_intcap; }

  TECLA = XBYTE[TECLADO]; // Se lee la tecla pulsada por el
    //usuario (el teclado está
    //mapeado en memoria
  TECLA = TECLA & 0x1F; // Se guarda en la var. global
    //TECLA

  if (TECLA==abort_key) // Si la tecla que se oprimió fue ABORT.
  { WREEPROM16(0x02, CLASEFIN); // Como se tuvo que
    //abortar, se almacena en la localidad
    //correspondiente de
    //memoria de datos la indicación apropiada
    ABORTPROCESS(); //y entonces se llama a la función
    //correspondiente

    kbcicinf: // Y una vez hecho eso, nos ciclamos infinitamente,
    //es forzoso que el operador apague o resetee al MC
    goto kbcicinf;
  }

  salir_intcap:
  CAPFLAG=1; // Se activa la bandera de dato disponible
} // FIN DE INTERRUPCION 0
//-----
//**** VECTOR DE INTERRUPCIÓN DE SOBREFLUJO PARA
// TIMER 0
//**** ETIQUETA: SFLUJO_INT
//**** Nota: Este vector de interrupción es el que mayor importancia
// tiene, pues es el que temporiza la actividad de contar
// media hora..
//-----

void OVERFLOW0 (void) interrupt 1
{
  if (MASTER==0)
    goto sfint_pc; // Si MASTER==1, entonces el sobreflujo se
    //manejará acorde al MC

  else
    goto sfint_mc; // Pero si MASTER==0, se manejará acorde
    //a la PC

    // ++++++
sfint_pc: // ++++++ CONTROL POR PC ++++++
    // ++++++

  goto salirovfl;

    // ++++++
sfint_mc: // ++++++ CONTROL POR MC ++++++
    // ++++++

  if (ETAPA == 6)
  { if (TIMER == min15)
    { TIMERFLAG = 1; // Si se cumplieron los 15 minutos
    //(sólo para etapa 6), reseteamos
    TIMER = 0x00; //todas las variables y reiniciamos el
    //conteo

```

```

    TLO = 0x00;
    TH0 = 0x00;
    TR0 = 0; }
  else
    TIMER++; // Si no se ha cumplido tan solo
    //incrementamos TIMER
  }
  else
  {
    if (TIMER == min30)
    { TIMERFLAG = 1; // Si se cumplieron los 30 minutos,
    //reseteamos
    TIMER = 0x00; //todas las variables y reiniciamos el
    //conteo
    TLO = 0x00;
    TH0 = 0x00;
    TR0 = 0; }
  else
    TIMER++;
  }
  salirovfl:
  ;
}
//-----
//**** VECTOR DE INTERRUPCIÓN SERIAL
//**** ETIQUETA: SERIAL_INT
//**** Este vector realiza tres funciones, indica si el que inicia la
// sesión es la PC; recibe petición de descarga de datos (activa
// la bandera DNW_FLAG); y recibe la petición de cambiar el
// maestro a la PC (cuando el MC tiene el control).
//-----
void SERIAL_INT (void) interrupt 4
{
  unsigned char dato=0;

  if (TI == 1) goto salir_scint;

  /* Al inicio de la sesión, tiene la tarea de revisar si la PC es quien
  inicia la sesión y en dado caso, si lo que solicita es la descarga de
  datos*/
  if (INI_FLAG==0) // Si la bandera bit de inicio está a 0,
  {
    //significa que el sistema está por arrancar
    INI_FLAG=1; //Entonces, INI_FLAG se coloca a 1 y nos
    //vamos a break_tec
    MASTER=0; //Nota: INI_FLAG no debe ser regresado a
    //cero durante la operación.

    dato = SBUF;
    if (dato = ini_dnw) // Se revisa si la PC envió una orden de
    DNW_FLAG=1; //descarga de datos. (0x32)
    // En dado caso, se activa la bandera DNW_FLAG
    goto salir_scint;
  }

  /* Durante el proceso regular, este vector de interrupción sólo está
  activo cuando el MC tiene el control, su función será la de verificar si
  el dato serial implica un cambio de maestro*/

  SCDAT = SBUF;
  if (SCDAT == changemtr) // Revisamos si lo que recibimos fue .
  { //petición de cambiar de maestro
    // Principio del if 1
    dato = SWITCHMTR(); // Mandamos a llamar a la función
    //de cambio de maestro, puede
    //tardar hasta 10 seg.
    if (dato == 0xFF) // Si la función retornó un 0xFF
    //significa que se volvió a interrumpir
    goto salir_scint; //la comunicación y que es necesario que el
    //MC siga siendo el maestro.
    MASTER = 0; // Pero si no, el control retorna a la PC
  } // Fin del if 1
  else
    MASTER = 1; // El control retorna al MC

  salir_scint:
  TI = 0; // Se limpia la bandera Rx
  RI = 0; // Se limpia la bandera Tx
} // FIN DE INTERRUPCION 4
//-----

```

```

//-----
//**** VECTOR DE INTERRUPCIÓN DE SOBREFLUJO PARA
// TIMER 2
//**** ETIQUETA: OVERFLOW2
//**** Nota: Este vector de interrupción es el de segunda mayor
// importancia, pues temporiza el almacenamiento de datos en
// la memoria de datos del sistema. Solo está activo cuando
// MC tiene el control
//-----
void OVERFLOW2(void) interrupt 5
{
  if (MASTER==0)
    goto salirovf2; // Si MASTER==0, el control es por PC,
                  //por ende, no hacemos uso de este
                  //vector de interrupción.

  // ++++++
  // ++++++ CONTROL POR MC ++++++
  // ++++++

```

```

if (TIMERSM==RDEEPROM16(TASAMT)) // ¿El periodo de
{ //muestreo se ha cumplido?
  TR2 = 0; // En dado caso, detenemos
          //momentaneamente el timer 2
  STRAMDATA(); // Mandamos almacenar datos
  TIMERSM = 0x00; // Después de eso, limpiamos todas las
  TL2 = 0x00; //variables involucradas con el conteo
  TH2 = 0x00; //en específico.
  TR2 = 1; // Y colocamos nuevamente al timer 2 para
          //temporizar.
}
else
  TIMERSM++; // En caso contrario, todavía falta para el
            //siguiente almacenamiento

  TIMERDP++; //También se incrementa el contador de
            //despliegue en display

salirovf2:
;
}
//-----

```

B.3 Contenido de Rutina1.c

```

//*****
//*****
//***** R U T I N A 1 . C *****
//**** ARCHIVO QUE CONTIENE LA PRIMERA PARTE DE LAS
//RUTINAS NECESARIAS EN EL SISTEMA. ****
//*****
//-----
//--- RUTINA PARA LA INIC. DE COMUNICACIÓN SERIAL ---
//--- NOMBRE: INITSER
//--- REFERENCIAS: Utiliza al Timer1 como generador de Baud Rate,
// el cual es a 4800 bps
//-----
void INITSER (void)
{
  PCON = PCON | 0x80; // SMOD=1
  TMOD=0x20; // Timer1 en autocarga
  TH1=230; // Baud Rate = 2400 bps
  SCON=0x50; // (trata después con 70H)
  TR1=1; // Se habilita Timer1

  //Nota:SMOD debe ser estrictamente = 1
}
//-----

//-----
//--- RUTINA DE INICIALIZACIÓN DE LCD
//-----
void INITLCD(void)
{
  DELAY100();
  // Se envía tres veces e1 primer comando
  // 2 líneas, resol. de 5x7 (3C=2|n+res5x10)
  LCD_COM(0x38); //0x30
  LCD_COM(0x30);
  LCD_COM(0x30);
  LCD_COM(0x38);
  // Display ON + cursor en forma de guión
  LCD_COM(0x0E); //0x0E
  // Display clear
  LCD_COM(0x01);
  // Entry set mode :Incremento, sin corrimiento
  LCD_COM(0x06); //0x06
  // DD RAM
  // LCD_COM(0x80); // Se coloca el cursor
}
//-----

```

```

//-----
//--- RUTINA DE IMPRESIÓN DE MENSAJE ---
//--- NOMBRE IMPRIME_LCD
//-----
void IMPRIME_LCD(unsigned char code *cadena)
{
  unsigned char indice = 0, TEMP = 0;
  unsigned int i=0;

  for (;;)
  {TEMP = cadena[indice];
  if(TEMP==0x00)
    break;
  XBYTE[LCDDAT]=TEMP; // Se envían los bits y ENABLE=high

  for (i=0xABFE; i>0; i--) // Generamos un retardo de 40 ms aprox.
  {;}

  indice++;}
}
//-----

//-----
//--- RUTINA DE ESCRITURA DE COMANDO PARA EL AND491
//--- NOMBRE: LCD_COM
//--- REFERENCIAS: El registro TEMP contiene la orden
//-----
void LCD_COM(unsigned char TEMP)
{unsigned int retardo=0;

  XBYTE[LCDCOM]=TEMP; // Dirección del LCD para comando
  for (retardo=0xFF00; retardo>0; retardo++)
  {;}

}
//-----

//-----
//--- RUTINA DE ESCRITURA DE DATO PARA EL AND491 ---
//--- NOMBRE: LCD_DATA
//--- REFERENCIAS: El registro TEMP contiene el dato
//-----
void LCD_DATA(unsigned char TEMP)
{unsigned int retardo=0;

  XBYTE[LCDDAT]=TEMP; // Se envían los bits y ENABLE=high
  for (retardo=0xFF00; retardo>0; retardo++)
  {;}

}
//-----

```

```

//-----
//--- RUTINA DE RETARDO DE 100 ms ---
//--- NOMBRE: DELAY100
//-----
void DELAY100(void)
{ unsigned int i = 0;

    for (i=0x00; i<0xFFFF; i++) ;
}
//-----

//-----
//--- RUTINA DE RETARDO DE 1 ms ---
//--- NOMBRE: DELAY1MS
//-----
void DELAY1MS(void)
{ unsigned int i = 0;

    for (i=0x420; i>0; i-- ) ;
}
//-----

//-----
//--- RUTINA PARA ESCRITURA EN LA EEPROM EXTERNA
//--- NOMBRE: WREEPROM16
//-----
void WREEPROM16(unsigned int valor, unsigned char xdata
*direccion)
{ unsigned int i = 0;

    *direccion = valor >> 0x08; // Primero se escribe la parte alta
    for (i=0x0420; i>0; i--) // Se hace un retardo de 1 ms
    {;} //para dar tiempo a la EEPROM de escribir el dato
    direccion++;
    *direccion = valor; // Ahora la parte baja del valor
    for (i=0x0420; i>0; i--) // Se hace un retardo de 1 ms
    {;} //para dar tiempo a la EEPROM de escribir el dato
}
//-----

//-----
//--- RUTINA PARA ESCRITURA EN LA EEPROM EXTERNA
//--- NOMBRE: RDEEPROM16
//-----
unsigned int RDEEPROM16(unsigned char xdata *direccion)
{ unsigned char c = 0;
  unsigned int dato=0;

  c = *direccion;
  dato = c;
  dato = dato << 0x08;
  direccion++;
  c = *direccion;
  dato = dato | c;
  return (dato);
}
//-----

//-----
//--- RUTINA PARA ENVIAR UNA PALABRA VÍA SERIAL A LA PC
//--- NOMBRE: SENDWORD
//--- DETALLES: Simplemente envía un dato 16-b en formato BCD
// hex hacia la PC, siempre en cuatro bytes.
//-----
void SENDWORD(unsigned int palabra)
{ unsigned char aux=0,i=0;
  // El orden de envío será en formato BCD, de MSN a LSN
  for (i=0; i<4; i++)
  { //Inicio del for1
    switch(i) {
      case 0: aux = palabra >> 0x08;
              aux = aux >> 0x04; // MSN
              break;
      case 1: aux = palabra >> 0x08;
              aux = aux & 0x0F; // 3er. nibble
              break;
      case 2: aux = palabra;
              aux = aux >> 0x04; // 2o. nibble
              break;
      case 3: aux = palabra;
              aux = aux & 0x0F; // LSN
              break;
      default: break; ;
    }
  }

  if (aux < 10) // Si aux<10, se convierte a BCD
    aux = aux+0x30; //con sumarle 0x30(48d)
  else // Si aux > 10, se convierte a BCD
    aux = aux+0x37; // con sumarle 0x37(55d)

  SBUF = aux;
  while(TI==0)
  {;}
  TI = 0;
} // Fin del for1

salir_SNDW:
TI = 0; // Esta línea es de relleno
}
//-----

//-----
//--- RUTINA PARA RECIBIR UNA PALABRA VÍA SERIAL DESDE LA
// PC con TIMEOUT
//--- NOMBRE: RXWORDTO
//--- DETALLES: Simplemente recibe un dato de 32-b desde la
//PC. Si ésta tarda más de la cuenta en responder, se asume que
//se ha interrumpido la comunicación y regresamos un 0xFFFF.
//Los cuatro bytes que recibe están en formato BCD
//-----
unsigned int RXWORDTO(unsigned char tipo)
{
  unsigned int palabra=0;
  unsigned long int contador=0;
  unsigned char aux1=0, aux2=0, i=0, aux3=0, aux4=0;

  ES = 0; // Inhabilitar temporalmente int. serial.

  for (i=0; i<4; i++)
  { // inicio del for 1
    switch (tipo) {
      case 0: contador = 0x1C2050; // Espera por dos segundos
              break;
      case 1: contador = 0x462156; // Espera por cinco segundos
              break;
      case 2: contador = 0x24BCDD3; // Espera por un minuto
              break;
      default: contador = 0x1E84F0;
    }

    while(RI==0) // Mientras recibimos el dato, nos ciclamos
    {contador--;
      if (contador==0x00) //Contador se decremente
      { RI=0;
        return (0xFFFF); } // Si se vence, regresamos un 0xFFFF
    }
    RI = 0;
    aux1 = SBUF;

    if (aux1<0x3A)
      aux1 = aux1 - 0x30;
    else
      aux1 = aux1 - 0x37;

    switch(i) { // Inicio del switch1
      case 0:aux4 = aux1; // En la primer iteración, obtenemos el
                    break; //byte más significativo
      case 1:aux3 = aux1;
                    break;
      case 2:aux2= aux1;
                    break;
      case 3: // En la última iteración obtenemos el
                    break; // byte menos significativo
      default: goto salir_RXW;
    } //Fin del switch1
    ES=0;
  } // fin del for 1
}

```

```

// Ahora juntaremos las partes
// Con esto formamos el byte bajo
aux2 = aux2 << 0x04;
aux2 = aux2 | aux1; // ---> 'aux2' tiene el byte bajo
// Con esto formamos el byte alto
aux4 = aux4 << 0x04;
aux4 = aux4 | aux3; // ---> 'aux4' tiene el byte alto

palabra = aux4;
palabra = palabra << 0x08;
palabra = palabra | aux2; // Finalmente 'palabra' tiene el dato

salir_RXW:
RI = 0;
return (palabra);
}
//-----

//--- RUTINA PARA EMITIR UNA SEÑAL AUDIBLE DE ALERTA
//--- NOMBRE: EMITSOUND
//--- DETALLES: Genera una señal de 10 Hz aprox. durante 3
//segundos.
//-----
void EMIT_SOUND(void)
{
    unsigned char i=0;

    for (i=0;i<0xA0;i++)
    { SOUNDPIN = 0;
      DELAY100();
      SOUNDPIN = 1;
      DELAY100(); }
    SOUNDPIN=0;
}
//-----

//--- RUTINA DE CONVERSIÓN DE UN NÚMERO A BCD Y SU
// IMPRESIÓN EN DISPLAY
//--- NOMBRE FLTBCDPRN
//--- DETALLES: EL dato que llega aquí tiene notación 10^2, por lo
//que simplemente se recorre el punto.
//-----
void FLTBCDPRN (unsigned int dato)
{ unsigned char indice=0;
  unsigned int cociente=0;

  if (dato == 0x00)
  { for (indice = pos0; indice<pos4 ; indice --)
    { LCD_COM(indice); // Se posiciona el cursor en el LCD
      if (indice==pos2)
        LCD_DATA (0x24); // Valor assci del '.'
      else
        LCD_DATA (0x30); // Valor assci del dígito '0'
      goto salirconv; } }

  indice = pos0;
  while (dato!=0)
  { LCD_COM(indice); // Se posiciona el cursor en el LCD
    if (indice==pos2)
      LCD_DATA (0x24); // Valor assci del '.'
    else
    { cociente = dato / 0x0A;
      dato = dato - (cociente * 0x0A);
      LCD_DATA(dato);
      indice--;
      dato = cociente; }
    }
  LCD_COM(indice);
  LCD_DATA(dato);

  salirconv:
  cociente = 0;//Línea de relleno
}
//-----

```

```

//-----
//--- RUTINA PARA CAPTURAR UN NÚMERO DESDE TECLADO //
// NOMBRE: SCANF2
//--- REFERENCIAS: Se requiere que INT0 esté, habilitado, esto
//es, cerciorarse de que IE.0 esté a 1. Esta función está hecha
//para teclado matricial y usando el LM74C923 ó LM74C922.
// De la misma forma hay que cerciorarse de que la configuración
//inicial del LCD establece que el corrimiento es a la derecha con
//sobreescritura.
//-----
unsigned int SCANF2 (unsigned char TIPOCAP)
{ unsigned char POSICION=0, NUM0=0, NUM1=0, NUM2=0;
  unsigned char NUM3=0;
  unsigned int NUMERO=0;

  /* Lo primero que se hará es acomodar al puntero de acuerdo al tipo
  de captura.*/
  if (TIPOCAP == integer)
  { LCD_COM (pos3);
    POSICION = pos3; }
  if (TIPOCAP == floating || TIPOCAP == time)
  { LCD_COM (pos4);
    POSICION = pos4; }
  if (TIPOCAP == date)
  { LCD_COM (pos7);
    POSICION = pos7; }
  if (TIPOCAP == index)
  { LCD_COM (pos1);
    POSICION = pos1; }
  if (TIPOCAP == yesnomb || TIPOCAP == yesno)
  { LCD_COM (pos0);
    POSICION = pos0; }

  CIC0: EX0=1; // Se habilita INT0
  CAPFLAG=0;
  while (CAPFLAG==0) // Recordar que en el vector de int. ext. 0
  //ésta bandera se coloca a 1
  { ; } // Sólo hasta que haya dato disponible se procesará.
  EX0=0; //Se deshabilita INT0
  CAPFLAG=0;

  /* De acuerdo con la configuración del teclado matricial, se hará un
  reajuste de valores para algunas teclas.*/
  if (TECLA==0x0C) TECLA=0;
  if (TECLA==0x00) TECLA=1;
  if (TECLA==0x01) TECLA=2;
  if (TECLA==0x02) TECLA=3;
  if (TECLA==0x08) TECLA=7;
  if (TECLA==0x09) TECLA=8;
  if (TECLA==0x0A) TECLA=9;

  /* Ahora se verificará que la tecla oprimida tenga una función
  específica*/
  if (TECLA==er_key)
  goto ERASE;
  if (TECLA==bkw_key)
  goto BACKWARD;
  if (TECLA==acc_key)
  goto FIN_CAP;
  if (!(TECLA<=9))
  goto NUMBER;

  goto CIC0;

  //--- Borrar dato---
  ERASE:
  if (TIPOCAP==yesnomb || TIPOCAP==yesno)
  goto CIC0;
  NUM0=0;
  NUM1=0;
  NUM2=0;
  NUM3=0;
  LCD_COM(0x80);
  if (TIPOCAP == integer)
  { IMPRIME_LCD(m_int);
    LCD_COM (pos3);
    POSICION = pos3; }
  if (TIPOCAP == floating)
  { IMPRIME_LCD(m_float);

```

```

LCD_COM (pos4);
POSICION = pos4;}
if (TIPOCAP == time)
{IMPRIME_LCD(m_hora);
LCD_COM (pos4);
POSICION = pos4;}
if (TIPOCAP == date)
{IMPRIME_LCD(m_fecha);
LCD_COM (pos7);
POSICION = pos7;}
if (TIPOCAP == index)
{IMPRIME_LCD(m_indice);
LCD_COM (pos1);
POSICION = pos1;}
goto CIC0;

/-- Número simple -----
NUMBER:
if (TIPOCAP==yesnomb || TIPOCAP==yesno)
{if (TECLA == 1) NUMERO = yes;
if (TECLA == 2) NUMERO = no;
if (TECLA == 3 && TIPOCAP==yesnomb) NUMERO = maybe;
else
goto CIC0;
goto salir_scanf;}

switch(POSICION) {
case pos0: if ( TIPOCAP==date)
{NUM1 = NUM1*10 + TECLA;
POSICION = pos0;}
else
{NUM0 = TECLA;
POSICION = pos0;}
break;
case pos1: if (TIPOCAP==integer || TIPOCAP==floating)
{NUM1 = TECLA;
POSICION = pos0;}
if (TIPOCAP==time)
{if (TECLA < 5)
goto CIC0;
NUM1 = TECLA;
POSICION = pos0;}
if ( TIPOCAP==date)
{if (TECLA>2)
goto CIC0;
NUM1 = TECLA;
POSICION = pos0;}
break;
case pos2: if ( TIPOCAP == integer)
{NUM2 = TECLA;
POSICION = pos1;}
break;
case pos3: if (TIPOCAP==integer)
{NUM3 = TECLA;
POSICION = pos2;}
if (TIPOCAP==floating)
{NUM2 = TECLA;
POSICION = pos1;}
if ( TIPOCAP== time)
{if (NUM3 == 2 && TECLA >4)
goto CIC0;
if (NUM3 < 2)
NUM2 = TECLA;
POSICION = pos1;}
if ( TIPOCAP==date)
{if (NUM1==1 && TECLA>2)
goto CIC0;
NUM2 = NUM1*10 + TECLA; // Cálculo del mes
if (NUM2==0)
goto CIC0;
if (NUM2==2 && NUM3>28) // Revisión del mes de febrero
goto ERASE;
if (NUM2==4 || NUM2==6 || NUM2==9 || NUM2==11)
{if (NUM3>30) //los meses de abril, junio,
goto ERASE; } //septiembre y noviembre.
POSICION = pos1;}
break;
case pos4: if ( TIPOCAP== floating)
{NUM3 = TECLA;
POSICION = pos3;}
if ( TIPOCAP== time)
{if (TECLA>2)
goto CIC0;
NUM3 = TECLA;
POSICION = pos3;}
if ( TIPOCAP==date)
{if (TECLA>1)
goto CIC0;
NUM1 = TECLA;
POSICION = pos3;}
break;
case pos6: if (NUM1==3 && TECLA>1)
goto CIC0;
NUM3 = NUM1*10 + TECLA;
if (NUM3==0)
goto CIC0;
POSICION = pos4;
break;
case pos7: if (TECLA>3)
goto CIC0;
NUM1 = TECLA;
POSICION = pos6;
break;
default: break;
} // Fin del switch1
wr_numero: //Se va a escribir el número
LCD_DATA(TECLA+0x30);
LCD_COM(POSICION);

goto CIC0;

BACKWARD:
if (TIPOCAP==yesnomb || TIPOCAP==yesno)
goto CIC0;
switch(POSICION) {
case pos0: if (TIPOCAP == integer || TIPOCAP == floating ||
TIPOCAP == time || TIPOCAP == index)
POSICION = pos1;
if (TIPOCAP==date)
goto CIC0;
break;
case pos1: if (TIPOCAP==integer)
POSICION = pos2;
if (TIPOCAP==floating || TIPOCAP==time)
POSICION = pos3;
if (TIPOCAP==index)
goto CIC0;
if (TIPOCAP==date)
POSICION = pos4;
break;
case pos2: if (TIPOCAP==integer)
POSICION = pos3;
break;
case pos3: if (TIPOCAP==integer)
goto CIC0;
if (TIPOCAP==time || TIPOCAP==floating)
POSICION = pos4;
if (TIPOCAP==date)
goto CIC0;
break;
case pos4: if (TIPOCAP==time || TIPOCAP==floating)
goto CIC0;
if (TIPOCAP==date)
POSICION = pos7;
break;
case pos6: goto CIC0;
break;
case pos7: goto CIC0; // El único tipo de captura que ocupa es
break; //el "date".
default : break;
} // Fin del switch2
wr_posi:
LCD_COM(POSICION);
goto CIC0;

```

```

//--- OK -----
FIN_CAP:
//Ahora vamos a convertir la cadena en el dato correspondiente
NUMERO = 0x00;
switch(TIPOCAP) {
case integer:
if (NUM3==1)
NUMERO = 0x03E8;
else
NUMERO = 0x00;
NUMERO = NUMERO + (NUM2*100) + (NUM1*10) + NUM0;
break;
case floating:
NUMERO = (NUM2*100) + (NUM1*10) + NUM0;
break;
case time:
NUM3 = NUM3 * 10;
NUM3 = NUM3 + NUM2;
NUMERO = NUM3;
NUMERO = NUMERO << 0x08;
NUM1 = NUM1 * 10;
NUM1 = NUM1 + NUM0;
NUMERO = NUMERO | NUM1;
break;

```

```

case date:
NUM2 = NUM2 << 0x04;
NUM0 = NUM2;
NUM0 = NUM0 & 0x80;
NUM3 = NUM3 | NUM0;
NUM2 = NUM2 << 0x01;
NUM1 = NUM1 | NUM2;
NUMERO = NUM3;
NUMERO = NUMERO << 0x08;
NUMERO = NUMERO | NUM1;
break;
case index:
NUMERO = NUM1*10 + NUM0;
break;
default: NUMERO = 0xFFFF;
} // Fin del switch 3

salir_scanf:
CAPFLAG = 0; // Se limpia bandera
EX0 = 1; // Se reactiva la interrupción externa 0.
return(NUMERO); // AQUÍ SE ENVÍA EL DATO CAPTURADO
DESDE TECLADO
}
//-----

```

B.4 Contenido de Rutina2.c

```

//*****
//*****
//***** R U T I N A 2 . C *****
//*** ARCHIVO QUE CONTIENE LA SEGUNDA PARTE DE LAS
//RUTINAS NECESARIAS EN EL SISTEMA.
//*****
//-----
//--- RUTINA PARA ENVIAR COMANDOS AL ADS774 (TI)
//--- NOMBRE: ADC_CN
//--- REFERENCIAS:
//--- CE = P0.7; R/C = P0.6; A0 = P0.5
//--- adtsel (0x4000) -> Para sel. de señal en el mux analógico
//--- adcconv(0x5000) -> Para enviar orden de conversión
//--- adcrdh(0x5100) -> Para leer los 8 MSB's
//--- adcrdl(0x5300) -> Para leer los 4 LB's
//-----
unsigned int ADC_CN(unsigned char selec)
{
unsigned int ADCOUT=0;
unsigned char TEMP=0,aux=0;

// Lo primero que haremos será seleccionar la señal a convertir
XBYTE[adtsel] = selec;

aux++; // Esta línea es de relleno, para dar tiempo a que en el
//mux analógico se presente la señal

// Ahora enviamos el pulso que indica el inicio de conversión
TEMP = XBYTE[adcconv];

/* Ahora realizamos un retardo de 13 us para dar tiempo al ADC de
terminar la conversión*/
for (aux=0x0D;aux>0;aux--)
{;}

TEMP = XBYTE[adcrdh]; // Ahora leeremos los 8 MSB's
aux = XBYTE[adcrdl]; // Luego los 4 LSB's

// Y los adaptamos a un dato de 16-b
ADCOUT = TEMP;
ADCOUT = ADCOUT << 0x08;
ADCOUT = ADCOUT | aux;
ADCOUT = ADCOUT >> 0x04;
ADCOUT = ADCOUT & 0x0FFF;

```

```

return (ADCOUT);
}
//-----
//-----
//--- RUTINA PARA DECODIFICAR Y EJECUTAR
//INSTRUCCIONES DESDE LA PC
//--- NOMBRE: DECINSTR
//--- REFERENCIAS: Entradas:
// comando (variables unsg. char universales)
// NOTA: SCDAT1 contiene en su nibble bajo el tipo de
// acción a realizar y SCDATO de momento (en esta primer
// versión) no tiene aplicación.
//-----
void DECINSTR(unsigned int comando)
{ unsigned char TEMP;

TEMP = comando & accionmsk;
switch (TEMP)
{ case muestra:
SENDSAMPLE(); break;
case abort_end:
ABORTENDPROCESS(); break;
case alarma:
EMIT_SOUND();
break;
case initmision:
INITPROCESS();
break;
case stdata:
STRAMDATA();
break;
case opvalve:
VALVPIN = abierto;
break;
case closevalve:
VALVPIN = cerrado;
break;
case newstage:
comando = comando & dispmsk;
ETAPA = comando >> 0x08;
break;
default: break;
} // Fin del switch
}
//-----

```



```

//-----
//--- RUTINA PARA ENVIAR LAS MUESTRAS VÍA SERIAL
//--- NOMBRE: SENDSAMPLE
//--- DETALLES: Realiza una serie de conversiones A/D, una de
//c/u de los dispositivos y los almacena en su locación
//correspondiente de memoria de datos. Esta rutina
//-----
void SENDSAMPLE(void)
{ unsigned char i=0,aux=0;
  unsigned int resultado=0;

  for (i=0;i<tot_disp;i++)
  { // Inicio de for 1

    switch(i) { // En este switch se indica el dispositivo a leer.
      case 0: aux = termopar0; // Primero será el termopar0 en su
        break; // respectivo canal
      default: break; // Los otros canales se irán ocupando después
    } //fin del switch1

    resultado = ADC_CN(aux); // Se manda hacer la conversión
    //de la señal, obteniendo un dato de 12-b
    switch (aux){ // Nos disponemos a colocar el identificador en
    //el valor añadiendo la máscara correspondiente.
      case termopar0: resultado = resultado | termopar0msk;
        //Añadimos la máscara correspondiente.
        break;
      default: break;
    } //Fin del switch2

    SENDWORD(resultado); // Y después enviaremos a la PC el
    //dato
  } // Fin de for 1
}
//-----
//--- RUTINA PARA ALMACENAR UNA SERIE DE MUESTRAS EN
LA MEMORIA DE DATOS
//--- NOMBRE: STRAMDATA
//--- DETALLES: Realiza una serie de conversiones A/D, una de
//c/u de los dispositivos y los almacena en su locación
//correspondiente de memoria de datos. Esta rutina SI ACTUALIZA
//los valores de DIRDATA y de DIRDATL.
//-----
void STRAMDATA (void)
{ unsigned char i=0,aux=0;
  unsigned int resultado=0;
  unsigned char xdata *ptr=0;

  for (i=0;i<tot_disp;i++)
  { // Inicio del for1
    switch(i) { // En este switch se indica el dispositivo a leer.
      case 0: aux = termopar0; // Primero será el termopar0 en su
        //respectivo canal
        ptr = tc0addr; // Y apuntamos a su dirección
        break;
      default: break; // Los otros canales se irán ocupando después
    } //fin del switch1

    resultado = RDEEPROM16(POINTER);
    ptr = ptr + resultado;
    resultado = ADC_CN(aux); // Se realiza la conversión y el dato
    //es de 12-b

    /* Nos disponemos a colocar el identificador en el valor y al puntero
    en la locación adecuada*/
    switch (aux){
      case termopar0: resultado = resultado | termopar0msk;
        //Añadimos la máscara correspondiente.
        break;
      default: break;
    } //Fin del switch2

    WREEPROM16(resultado, ptr);
    SENDWORD (resultado); // Y después se envía el dato a la PC
  } // Fin del for 1

  resultado = RDEEPROM16(POINTER);
  resultado++;
  WREEPROM16(resultado, POINTER); //Y actualizamos el valor
  //de POINTER en la zona de parámetros de la
  //memoria de datos del sistema
}
//-----
//--- RUTINA PARA HACER EL CAMBIO DE MAESTRO
//--- NOMBRE: SWITCHMTR
//--- DETALLES: Antes de hacer formalmente el cambio de
//maestro, hay que enviar los siguientes datos a la PC: etapa en la
//que se encuentra, conteo del timer y parámetros de la quema.
//-----
unsigned char SWITCHMTR (void)
{ unsigned int xdata *puntero=0;
  unsigned int dato=0,aux=0;
  unsigned char i=0;

  // Lo primero enviaremos los parámetros de la quema
  puntero = STPAR; // Apuntamos a la zona de parámetros.
  for (i=0;i<tot_param;i++)
  { dato = *puntero;
    SENDWORD(dato); // Enviamos el dato
    aux = RXWORDTO(1); // Se espera como respuesta el mismo
    //dato
    if (dato != aux || dato == 0xFFFF) // En caso contrario,
    //terminamos rutina
    return(0xFF); //e indicamos que hubo error
    puntero++;
  }

  // Segundo, enviaremos la etapa en la que nos encontramos.
  aux = ETAPA;
  SENDWORD(aux);
  dato = RXWORDTO(1);
  if (dato != aux || dato == 0xFFFF) // En caso contrario, terminamos
  //rutina
  return(0xFF);

  // Finalmente, enviaremos el valor del timer.
  SENDWORD(TIMER);
  dato = RXWORDTO(1);
  if (dato != TIMER || dato == 0xFFFF) // En caso contrario,
  //terminamos rutina
  return(0xFF);

  return(0x00);
}
//-----
//--- RUTINA PARA DESCARGAR DATOS EN LA PC
//--- NOMBRE: DOWNLOAD
//--- DETALLES: Cuando todo un proceso no ha sido registrado
//completamente por la PC, la memoria de datos del MC entra
//como respaldo. Esta rutina se encarga de enviar todo el
//contenido de la memoria de datos del sistema hacia la PC.
//-----
unsigned int DOWNLOAD(void)
{
  unsigned int dato=0,aux=0;
  unsigned int xdata *ptr=0;
  unsigned char i=0;

  ptr = STPAR; // Primero se enviarán los parámetros de la
  //quema.
  for (i=0;i<tot_param;i++)
  { // inicio del for 1
    dato = *ptr;
    SENDWORD(dato);
    aux = RXWORDTO(1);
    if (aux != dato || aux == ER_COMM)
      break;
    ptr++;
  } // Fin del for 1
}

```

```

SENDWORD(endpar);
aux = RXWORDTO(1);
if (aux == 0xFFFF)
return (0xFFFF);

ptr = STACC; // Luego se enviarán las acciones de la quema.
for (;;)
{ // inicio del for 2 indefinido
dato = *ptr;
if (dato == endaction)
break;
SENDWORD(dato);
aux = RXWORDTO(1);
if (aux != dato || aux == ER_COMM)
return(0xFFFF);
ptr++;
} // Fin del for 2 indefinido

SENDWORD(endaction);
aux = RXWORDTO(1);
if (aux == ER_COMM)
return (0xFFFF);

/* EN ESTE ULTIMO PUNTO DE LA RUTINA, ENVIAMOS LOS
DATOS ALMACENADOS*/
// ---- Iniciaremos con el termopar 0 ----
ptr = tc0addr;
for (;;) // Por último las lecturas de los dispositivos
{ // inicio del for 3
aux = RDEEPROM16(POINTER); //Lee el valor de pointer, que
//tiene la última localidad de la memoria de datos
dato = ptr; // En "dato" se guarda la localidad actual de "ptr".
if (aux == dato) // Si "aux" y "dato" son iguales, rompemos el
break; //ciclo.
dato = *ptr;
SENDWORD(dato);
aux = RXWORDTO(1);
if (aux != dato || aux == ER_COMM)
return (0xFFFF);
ptr++;
} // fin del for 3

/* Cuando se incluyan más sensores, se colocará su respectivo bucle
for.*/

SENDWORD(enddata);
aux = RXWORDTO(1);
if (aux == 0xFFFF)
return (0xFFFF);

return(0);
}
//-----
//--- RUTINA PARA OBTENER LA TEMPERATURA;
//--- NOMBRE: GET_TEMP
//--- DETALLES: Realiza conversiones A/D a los termopares,
//hace una estimación de la temperatura neta en el horno.
//-----
unsigned int GET_TEMP(void)
{ unsigned int aux1=0;
aux1 = ADC_CN(termopar0);
aux1 = aux1 >> 0x01; // Como el valor recibido es la mitad
//de la temperatura real, dividimos entre 2, o
//lo que es igual, desplazar 1 bit a la izquierda

//Ahora nos disponemos a desplegar en LCD
LCD_COM(borrarlcd);
LCD_COM(0x82); //Posicionamos en celda dos de la primera
//línea al cursor
IMPRIME_LCD(m_temp); // Mandamos imprimir palabra
//"Temperatura"
FLT2BCDPRN(aux1); // E imprimimos el valor de la temperatura
//medida
LCD_COM(0xCC); // Posicionamos en celda trece de la segunda
//línea al cursor
LCD_DATA(0xCF); //Imprimimos el caracter 'o'
LCD_DATA(0x43); //Imprimimos el caracter 'C' -> ' °C '

```

```

return (aux1);
}
//-----
//--- RUTINA PARA INICIAR EL PROCESO
//--- NOMBRE: INITPROCESS
//--- DETALLES: Todos los subsistemas son inicializados: el adc,
//los puentes H, etc.
//-----
void INITPROCESS(void)
{
WREEPROM16(0x000,POINTER); // Se limpia el puntero a
//memoria de datos
TIMER = 0x0000; // Se limpia el temporizador de eventos
TIMERSM = 0x0000; // Se limpia el temporizador de
//almacenamiento de datos.

/* De acuerdo con el que haya iniciado la sesión, la bandera
CLASEFIN tomará el valor correspondiente .*/

if (MASTER==0)
WREEPROM16(0x00, CLASEFIN);
else // Si CLASEFIN==0, la PC inició el proceso y va por
//buen camino, en caso
WREEPROM16(0x01, CLASEFIN); //contrario, los datos
//requieren de ser almacenados

ETAPA = 0; //La bandera-byte ETAPA se coloca a 0 para
//indicar desde donde se iniciará el proceso.

//*****
// NOTA: Esta bandera no debe ser vuelta a 0 hasta que el
proceso termine o sea abortado en su defecto
//*****
}
//-----
//--- SUBROUTINA PARA FIN DE PROCESO
//--- NOMBRE: ENDPROCESS
//--- DETALLES:
//-----
void ENDPROCESS (void)
{
/* Lo primero que se hará en esta rutina es cerrar la válvula
principal, con ello se corta el suministro de gas */
VALVPIN = cerrado;

EMIT_SOUND(); // Emitimos alarma de terminación
LCD_COM(borrarlcd); // .
LCD_COM(0x80); // .
IMPRIME_LCD(m_proc); // .
LCD_COM(0xC0); // Con estas líneas
IMPRIME_LCD(m_fin); //líneas avisamos al operador

if (MASTER == 0) // Ahora revisaremos quién terminó la
{ //operacion

/* Si fue la PC quien la terminó e incluso quien siempre llevó
la operación, no hay problema, los datos siempre estuvieron
guardados.*/
if (RDEEPROM16(CLASEFIN) == 0x00)
WREEPROM16(0x00, CLASEFIN);
}
else // Pero si quien terminó el proceso fue el mismo MC,
{WREEPROM16(0x01, CLASEFIN); //Indicamos en la
//variable CLASEFIN

}

salir_abend;
;
}
//-----

```

```
//-----  
//--- SUBROUTINA PARA ABORTO DE PROCESO  
//--- NOMBRE: ABORTPROCESS  
//--- DETALLES:  
//-----  
void ABORTPROCESS (void)  
{  
  VALVPIN = cerrado; // Lo primero que se hará en esta rutina  
                    //es cerrar la válvula principal, con ello se  
                    //corta el suministro de gas.  
  
  LCD_COM(borrarlcd);  
  LCD_COM(0x80);  
  
  IMPRIME_LCD(m_proc);  
  LCD_COM(0xC0);  
  IMPRIME_LCD(m_abort);  
  
  WREEPROM16(0x02, CLASEFIN);  
  
  EMIT_SOUND();  
  EMIT_SOUND();  
  EMIT_SOUND();  
  
}  
//-----
```


Apéndice C - MANUAL DEL SISTEMA

En este apéndice muestra el manual operativo del sistema. Se omitieron la portada y el índice para evitar una confusión con las referencias a número de página. De igual forma, los números de página originales se han suprimido y la numeración de las figuras y las secciones se han adaptado para este documento.

C.1 Descripción del sistema

Introducción

HORNO MONITOR v1.0 es un sistema para monitorear la temperatura en un horno a base de gas, diseñado especialmente para el Taller de Cerámica de la Universidad Tecnológica de la Mixteca.

HORNO MONITOR tiene la siguiente representación:

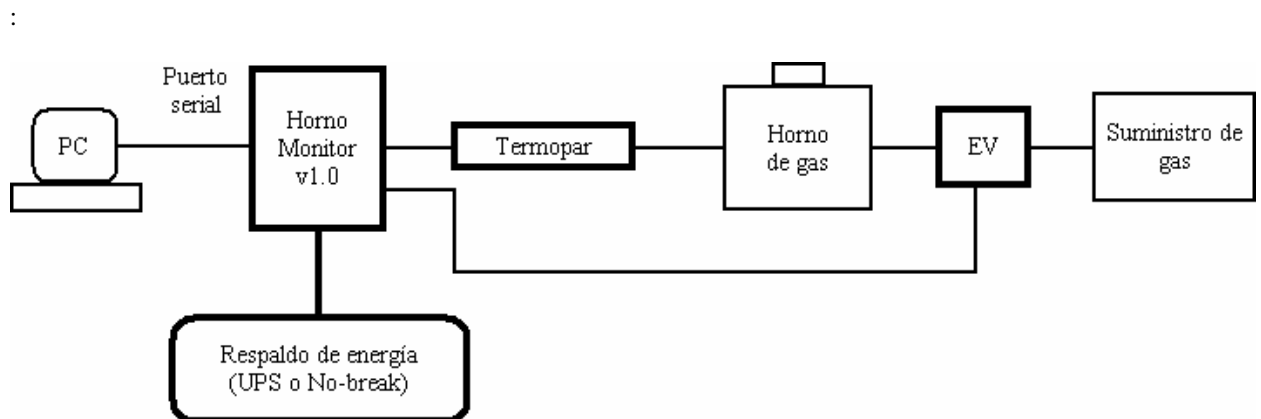


Figura C.1. Estructura general de Horno Monitor v1.0.

Componentes del sistema

De la figura C.1, los elementos con borde grueso son los componentes del sistema Horno Monitor v1.0.

1. **Horno Monitor v1.0.** Es un gabinete donde se conectan el termopar, la electroválvula y también el cable de comunicación serial. Tiene integrados una pantalla alfanumérica y un teclado numérico para desplegar información y capturar datos (cuando aplique).

El gabinete es de forma prismática y en la figura C.2 se muestra su perfil frontal. Cuenta con un botón de encendido localizado en la parte inferior derecha. Justo arriba del botón de encendido hay un interruptor configurado para un tomacorriente de 115 Vac, el cual no debe ser cambiado de posición. Arriba del interruptor hay un conector para cable estándar de corriente alterna con tierra, todo es cuestión de conseguir el cable por separado.

El conector serial está situado en su costado izquierdo a media altura, su tipo es el estándar DB9 de género macho, lo que indica que el cable que lo conecte con una computadora debe tener ambos extremos de género hembra.

El conector del termopar es OST hembra para tipo K.

Las terminales para la electroválvula van a una hilera interna de 4 puntos de conexión con doble tornillo (conocidos también como hileras de conectores para corriente alterna con doble tornillo o bien, *2-screw VAC terminal strips* en inglés). Viendo de frente a la hilera y de izquierda a derecha, las terminales son E1, E2, Vca y Gnd. Las dos primeras son simplemente la fase y el neutro de la electroválvula (ambos en color rojo). La siguiente es Vca y en realidad es un cable con un conector estándar (color negro) de corriente alterna que irá a una salida del respaldo de energía. Finalmente, Gnd es la tierra física reconocida (color verde) e irá al cable del mismo color de la electroválvula.

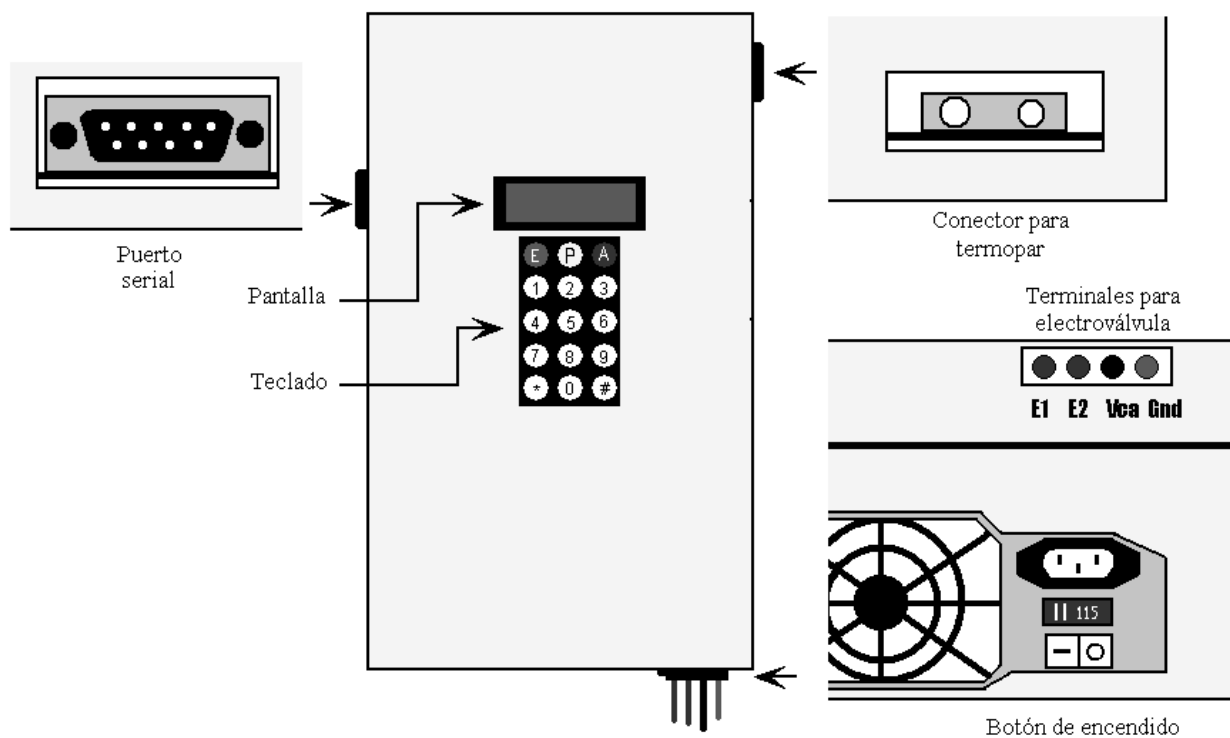


Figura C.2. Gabinete del sistema Horno Monitor v1.0

2. **Termopar.** Es el transductor empleado para medir la temperatura. El termopar provisto con el sistema es de tipo K y tiene un termopozo de alúmina con longitud de 43 cm. (ver figura C.3), adecuado para hornos de alta temperatura a base de gas. Para conectarse al gabinete, tiene un

conector OST macho para tipo K. Si el usuario desea cambiar el termopar, debe cuidar que sea estrictamente de tipo K, el termopozo y la longitud del mismo es a elección del usuario.

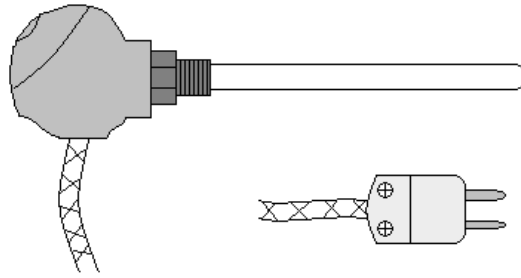


Figura C.3. Termopar tipo K.

En el conector se aprecia que un pin o pata es más grueso que el otro. De igual forma el conector para termopar en el gabinete hay un agujero más grande que el otro. Eso indica que existe una particular forma de enchufar este dispositivo y nunca se debe forzar a los conectores.

3. **EV o Electroválvula.** Permite/detiene el flujo del gas, el diámetro de la tubería que acepta es de $\frac{3}{4}$ de pulgada. Tiene tres alambres, dos de color rojo que irán a las terminales E1 y E2 del gabinete (el orden no importa) y una más de color verde, que irá a la terminal Gnd en el gabinete.

La electroválvula provista es el modelo S201GF02N5EG5 de la marca GC Valves (ver figura C.4), funciona con corriente alterna (115Vca, 60 Hz). Por especificaciones del fabricante, cada 1000 horas de uso se le debe dar mantenimiento de acuerdo al manual propio y anexo a este documento. Se recomienda que ésta se instale lo más cercano posible al suministro de gas.

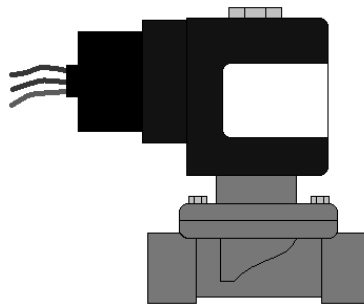


Figura C.4. Electroválvula.

4. **Respaldo de energía (no-break o UPS).** Es un dispositivo que, en caso de un corte de energía eléctrica, puede proveer hasta por 1.6 horas la energía al sistema. La computadora NO DEBE estar conectada al sistema. El modelo provisto es el INTERNET 350U de la marca Tripp-Lite (ver figura 1.5), su capacidad es de 350 VA. Si desea el usuario incrementar el tiempo de respaldo de energía, tiene que aumentar la capacidad del dispositivo. En dado caso, se recomienda el modelo INTERNET 750U para aumentar el tiempo a 3.2 horas o bien, el TE1200 para casi 15 horas, ambos de la misma marca.

De la figura C.5 se observa que hay 8 tomas de corriente eléctrica en dos hileras de cuatro cada una. Una de las hileras tiene dos símbolos, un rayo y una batería, son las que están

respaldadas; aquí deben ir conectados los cables de corriente directa del gabinete (el que está arriba del botón de encendido y el de la electroválvula). La otra hilera tendrá corriente eléctrica mientras el UPS esté conectado a una toma-corriente y éste tenga energía a su vez.

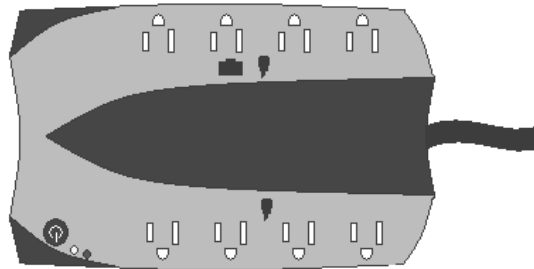


Figura C.5. Respaldo de energía.

Características

- El sistema está basado en un microcontrolador (MC).
- El control principal del monitoreo reside en un programa para computadora y ésta emplea al sistema (MC en este caso) para adquirir información sobre la temperatura actual del horno a través del puerto serial.
- Muestra en tiempo real la temperatura interior del horno (control por computadora solamente).
- Grafica en tiempo real la temperatura (control por computadora solamente).
- Guarda los datos de la quema en archivo (control por computadora) y en un módulo de memoria del sistema (control por sistema o MC).
- Garantiza el seguimiento del proceso: si la computadora falla por cualquier motivo, el mismo sistema puede continuar y terminar el proceso gracias al respaldo de energía (conocido como UPS o también como no-break).

Requisitos de la computadora

- Procesador Intel 80486 o superior (o un equivalente a Intel).
- Sistema operativo Microsoft Windows 95/98/2000/XP (el operador debe tener conocimientos básicos sobre este ambiente).
- Microsoft Excel 97/2000/XP
- RAM mínima de 32 MB (independiente de la memoria requerida por el sistema operativo).
- Espacio en disco duro de 3 MB (independiente del sistema operativo).
- Puerto serial RS-232. En caso de no contar con él pero sí tener un puerto USB disponible, comprar un adaptador USB-Serial.

C.2 Instalación del software para computadora

En el disco de instalación provisto con este manual, buscar y ejecutar el siguiente archivo:

<Unidad de CD>:\Instalacion\Setup.exe

Con ello aparecerá la siguiente pantalla:

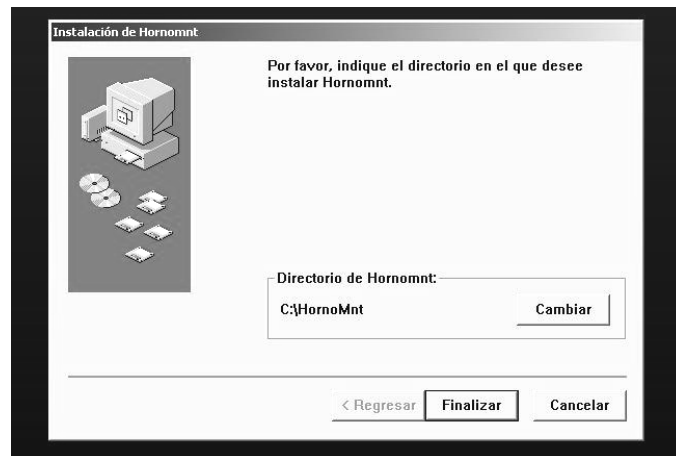


Figura C.6. Instalador del sistema

El programa del sistema se instalará en un directorio por omisión, pero el usuario es libre de cambiar a un directorio más conveniente para él. Para cambiar el directorio basta con hacer clic en el botón “Cambiar”.

Una vez bien definido el directorio de instalación, oprimir el botón “Finalizar”, con lo que se procederá formalmente a instalar el sistema en la computadora o si lo prefiere, haga clic en “Cancelar” para salir del instalador.

Si optó por “Finalizar”, el instalador seguirá su tarea hasta terminarla, en ese momento aparecerá un aviso como la de la figura C.7. Para que la instalación se lleve a cabo exitosamente, el usuario debe reiniciar el sistema operativo.

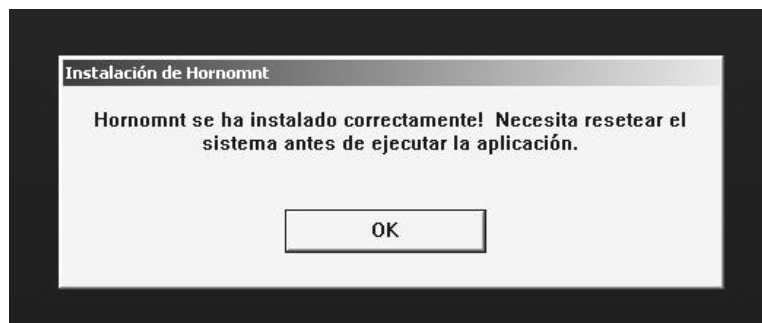


Figura C.7. Instalación exitosa del sistema.

Con esto, se crea un icono de acceso directo en el menú Inicio\Programas.

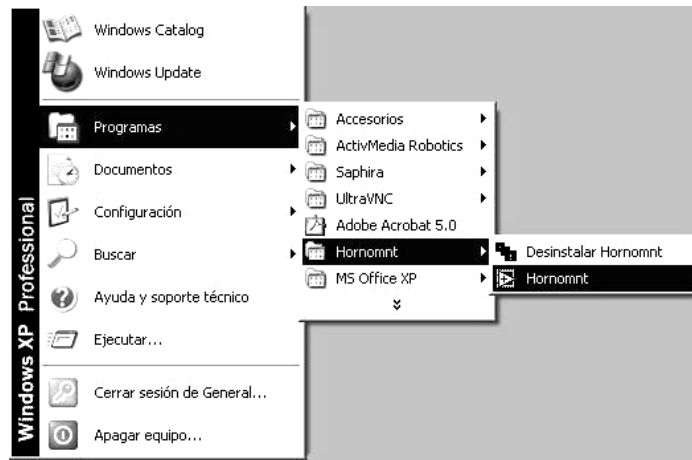


Figura C-8. Acceso directo del programa.

C.3 Lógica del proceso de quema

La lógica de la quema empleada implica que el operador proporcione al sistema los siguientes parámetros:

- Temperatura objetivo de la quema (min. 573°C, máx. 1250°C).
- Hora de inicio de la quema (en formato de 24 hrs.).
- Fecha de la quema (en formato dd/mm/aa).
- Tasa de almacenamiento o muestreo de datos (min. 1 minuto, máx. 10 minutos). Este parámetro se refiere a la periodicidad con la que se guardará la temperatura del horno en el sistema.

Una vez conocidos tales datos, inicia propiamente el proceso de quema abriendo la electroválvula. Obviamente el operador debió previamente hacer los preparativos del horno y el último paso de esos preparativos será el encender al horno una vez que el sistema haya abierto la electroválvula.

El proceso abarca tres etapas:

1a. Etapa

La meta de esta etapa es llegar a una de dos temperaturas: 850°C o 900°C. Si la temperatura objetivo es mayor o igual a 850°C y menor a 900°C, significa que la quema es de sancocho, entonces, al llegar a dicha temperatura, se corta el suministro de gas y el horno comienza a enfriarse, el proceso seguirá funcionando hasta que la temperatura interior del horno haya descendido a 600°C. Si la temperatura objetivo es mayor a 900°C, significa que la quema es de esmalte y una vez alcanzados los 900°C se da paso a la segunda etapa. Durante esta primera etapa se deben alcanzar la inversión de la cristobalita y la del cuarzo, lo único que se hace es avisarle de ello al usuario.

De acuerdo al valor de la tasa de almacenamiento o muestreo, cada determinado número de minutos se guarda el valor de la temperatura, tanto durante el calentamiento del horno como durante el enfriamiento del mismo. De igual forma, cada media hora se avisa al operador que es necesario aumentar la presión del gas combustible en 0.05 Kg/cm².

2a. Etapa

La meta de esta etapa es vigilar que la temperatura del horno esté en el rango de 900°C a 905°C durante quince minutos para que el esmalte madure. Una vez pasados los quince minutos se accede a la tercera y última etapa. Durante ese tiempo se avisa al operador que aumente la presión del gas en 0.01 Kg/cm² si la temperatura desciende del rango, o a disminuirla en caso contrario. El guardado de datos es igual al que se hace en la primera etapa.

3a. Etapa

La meta de esta etapa es llegar a la temperatura objetivo. Una vez logrado aquello, se corta el suministro del gas mediante la electroválvula y se espera el enfriamiento del horno de la misma manera que en la primera etapa. El guardado de los datos y el incremento de la presión del gas son, nuevamente, iguales en forma a la primera etapa.

Para hacer una gráfica más ilustrativa, el sistema sigue graficando en la etapa de enfriamiento, hasta que la temperatura desciende a 600°C.

C.4. Modos de operación

Horno Monitor v1.0 tiene dos modos de operación: control del monitoreo por computadora y por MC (sistema). Ambos siguen la misma lógica del proceso, lo único que cambia es la interfaz que le presentan al operador. Mientras que la computadora tiene una interfaz gráfica totalmente amigable, el MC presenta una limitada pero suficiente para realizar el monitoreo de la quema.

C.4.1 Control por computadora

En el control por computadora se pueden hacer tres tareas, estos se acceden mediante los menús del programa y son:

- Iniciar nueva quema. Es realizar desde el comienzo e íntegramente (salvo algún percance) un proceso de cocción. En este tipo de control es posible ver en tiempo real la gráfica temperatura-tiempo generada por la quema. Al final de ésta, se pide al operador guardar en un archivo los datos de la quema.
- Recuperar datos desde MC (sistema). La última quema realizada tiene aún respaldados los datos en el sistema o MC y es posible recuperarlos con esta opción guardándolos en archivo e incluso imprimiéndolos directamente en la impresora por omisión de la computadora.
- Graficar datos de quemas anteriores. Esto es simplemente elegir un archivo de cualquier quema realizada e imprimir la gráfica.

Conociendo la interfaz

La interfaz del programa es la mostrada en la figura C.9 y está compuesta por los siguientes elementos:

1. Menús del programa: Contienen todas las tareas que es posible hacer en el programa.
2. Parámetros de la quema: Definen cómo va a ser la quema. La temperatura objetivo es la temperatura a la cual se va a realizar la quema y el operador debe introducir en su caja de texto correspondiente un valor que esté en el rango expresado en la sección C.3, de lo contrario, el sistema le pedirá al operador que rectifique. La tasa de muestreo aparece etiquetada como “Almacenar datos” y se representa como una persiana que tiene valores preestablecidos desde 1 hasta 10 minutos. No se pregunta la hora de inicio ni tampoco la fecha porque el mismo programa los toma desde el sistema operativo de la computadora (esto implica que tales datos deben estar bien configurados). Se advierte que hay un parámetro extra: el puerto de comunicación. Este compete solamente al programa por computadora e indica el número de identificación del puerto presente en la computadora. Por omisión se tiene el puerto 1 en vista de que la mayoría de las computadoras personales tienen solamente uno (en modelos recientes ya no está presente dicho puerto serial, especialmente en computadoras portátiles). En caso de tener más, verificar qué puerto es al que conectaron el cable.
3. Indicador de temperatura: Muestra la temperatura actual del horno, ésta se refresca cada 2 segundos. Indicador del estado de la electroválvula. Cuando el sistema lo activa, el color del indicador es un verde intenso y cuando lo desactiva el color es un verde ocre.
4. Botón de aborto de quema. Si el operador cree que es necesario abortar, tiene que oprimir la combinación Shift + F12. Se opta por este medio para evitar un accionamiento accidental del botón con el ratón.
5. Ventana de estado de la operación. Cualquier situación que se presente, como la inversión del cuarzo y la de la cristobalita, la necesidad de incrementar la presión del gas, el aviso de la etapa en la que se encuentra el proceso, etc., se muestran en esta ventana, acompañándose de una señal audible.
6. Gráfica de la quema. De acuerdo con la tasa de muestreo, irá apareciendo un punto en la gráfica cada determinado número de minutos.

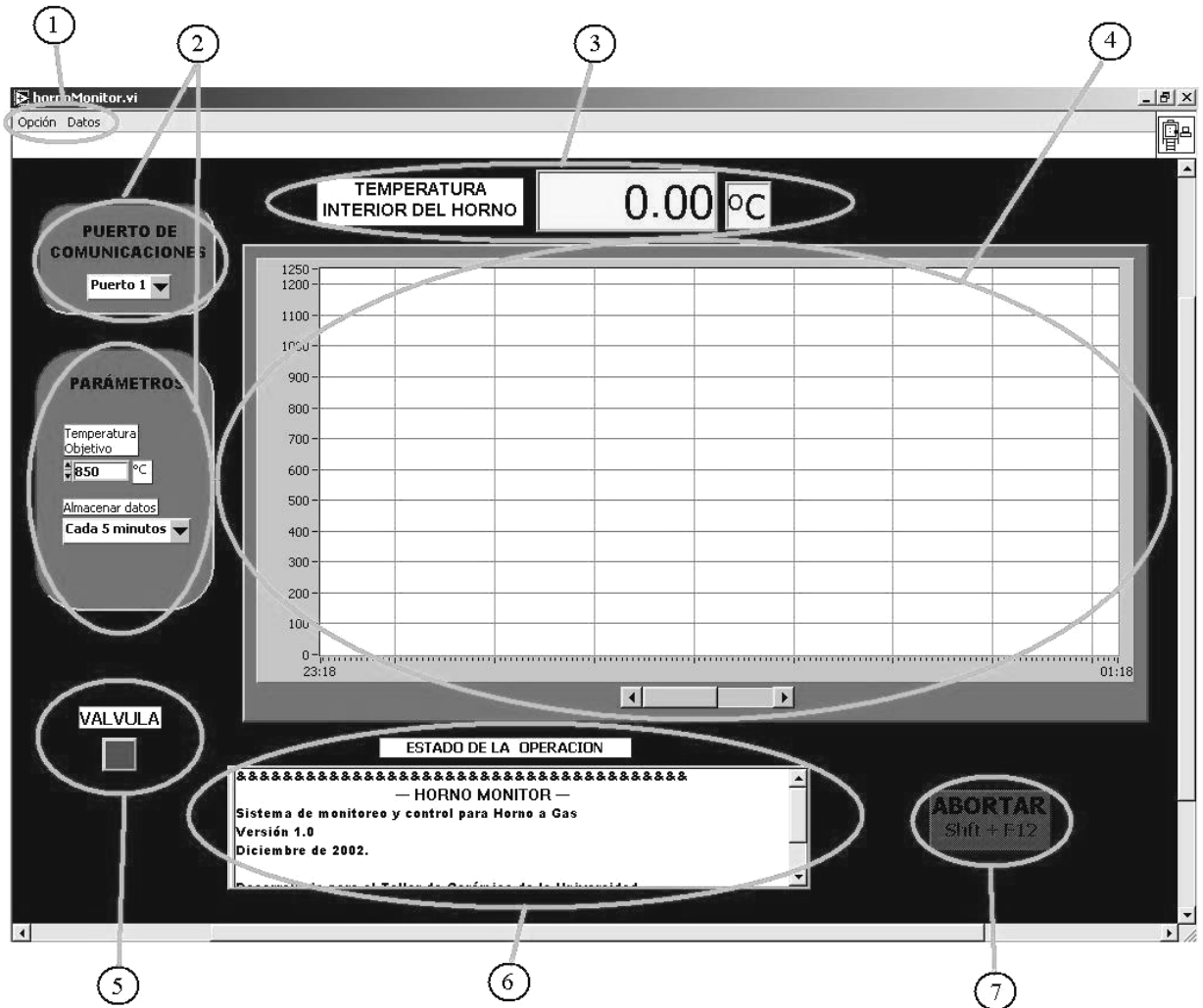


Figura C.9. Interfaz del control por computadora.

Realizando una quema

Para iniciar una nueva quema, elegir la opción “Iniciar nueva quema” en el menú “Opción”, como se ve en la figura C.10. Previamente el operador debió haber encendido al sistema usando el botón de encendido en el gabinete.



Figura C.10. Menú “Opción” y su contenido.

Después de esto, el sistema revisará el estado de la última quema, si considera que no hay un registro (sea completo o parcial) de la citada quema, le dará un aviso emergente al operador sobre la situación, presentándole dos opciones: CONTINUAR o TERMINAR PROGRAMA. Si

eligió TERMINAR PROGRAMA, la quema no se inicia y se termina el programa, esperando que la próxima vez el operador elija la opción “Descargar datos desde MC” en el menú “Datos”.

Si eligió CONTINUAR, la quema iniciará normalmente y los datos anteriores se perderán. De inmediato activa la electroválvula y lo hace saber a través del indicador correspondiente en la interfaz y el operador debe encender inmediatamente después los quemadores y ajustar la presión del gas a su criterio. A partir de este momento, el botón ABORTAR de la interfaz del programa estará habilitado para que, si el operador lo considera apropiado, se aborte el proceso de quema. Si se llega a dar el caso, el sistema cierra la electroválvula (con la indicación respectiva en la interfaz) y prepara el registro de los datos en un archivo, donde le pedirá al operador que asigne un nombre al archivo o en su defecto, acepte el nombre por omisión que ofrece el programa.

En condiciones normales, el proceso se irá cumpliendo y conforme pase el tiempo y la temperatura se vaya incrementando, la gráfica irá aumentando en puntos. Por su parte, la ventana del estado de la operación irá mostrando eventos de la quema, tales como la inversión de la cristobalita, la necesidad de incrementar la presión del gas, y demás, junto con una señal audible para que el usuario tenga noticia de ello.

Cuando el proceso llegue a su fin, aparecerá un aviso emergente avisándole al operador que la quema ha concluido sin problemas y de inmediato generará un archivo con los datos de la quema, no sin antes pedirle al operador que le asigne un nombre o bien, que acepte el nombre por omisión que otorga el programa. También le pregunta si desea imprimir la gráfica, en caso de responder “sí”, imprime la gráfica de la interfaz en la impresora configurada por omisión en el sistema operativo de la computadora.

NOTA: Se entiende por “aviso emergente” a toda ventana que aparezca durante el proceso y que requiera que el operador haga clic en algún botón. Los avisos que se dan en la ventana “Estado de la operación” no interrumpen de manera alguna el procedimiento, pero es pertinente que el operador se entere de su contenido.

¿Qué pasa si se pierde la comunicación durante la quema? En caso de que se llegase a interrumpir la comunicación entre computadora y el MC, aparece un aviso emergente en el programa indicándole el problema (antes de ello, el sistema agotó todas las posibilidades de reestablecer la conexión) e inmediatamente después el programa se cierra. El operador no tiene que preocuparse, la electroválvula no se cerrará y por ende, el proceso no se interrumpirá, pues el MC asumirá el control del monitoreo y le dará seguimiento hasta terminarlo (o abortarlo, según sea el caso). El único detalle es que ahora la única forma de ver lo que pasa durante la quema es mediante la pantalla del sistema que tiene una presentación limitada en comparación con la del programa.

Recuperando datos desde MC

Para iniciar una nueva quema, elegir la opción “Recuperar datos desde MC” en el menú “Datos”, como se ve en la figura C.11.

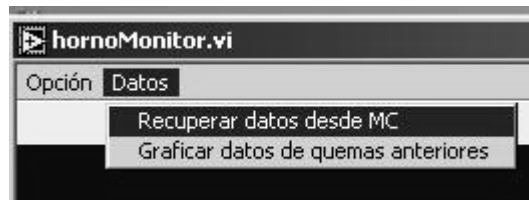


Figura C.11. Menú “Datos” y su contenido.

De nueva cuenta, el parámetro “Puerto Serial” en la interfaz debe estar bien configurado para que se realice la descarga de datos. Si ningún problema relacionado con este parámetro ha ocurrido, de inmediato se inicia la descarga, esperando que no haya problemas en la comunicación.

NOTA: La descarga puede tardar varios minutos, dependiendo de la cantidad de datos almacenada en la última quema, de manera tal que el operador no debe asumir que la comunicación se ha perdido, salvo que el mismo sistema lo llegue a indicar.

En condiciones normales, al terminar de descargar los datos el programa dará el aviso emergente correspondiente al operador, inmediatamente después le pedirá que guarde los datos en un archivo, proponiéndole un nombre por omisión, que el operador tiene total libertad de cargar. También le pregunta si desea imprimir la gráfica, en caso de responder “sí”, imprime la gráfica de la interfaz en la impresora configurada por omisión en el sistema operativo de la computadora.

Graficando datos de quemas anteriores

Para iniciar una nueva quema, elegir la opción “Graficar datos de quemas anteriores” en el menú “Datos” (ver figura C.11). El sistema le pide al operador que elija el archivo y lo imprime en la impresora por omisión del sistema operativo de la computadora. El tipo de gráfica que el mismo programa por PC logra generar es un tanto limitado en comparación con los que otros programas logran.

Formato de los archivos

Horno Monitor v1.0 genera dos archivos por cada quema:

- Archivo con extensión “Rq”: Es un archivo en formato texto que se usa cuando el operador elige la opción “Graficar datos de quemas anteriores”, contiene una estructura tal que el operador no debe modificar, pues ya no podrá graficar ni imprimir la información de ese archivo.
- Archivo con extensión “XLS”: Su contenido es idéntico al RQ, pero está hecho para ser manipulado en el popular software Microsoft Excel (versión 97 en adelante). Esto remedia la limitación de la graficación que hace el sistema, pues en Excel se tienen más opciones para personalizar la gráfica. Lo anterior implica que el operador debe tener conocimiento del software en cuestión.

C.4.2 Control por MC (Sistema)

A diferencia del control (del monitoreo) por computadora, el MC o sistema sólo puede realizar quemas, no grafica ni imprime, aunque sí carga los datos de quemas anteriores en la computadora. La interfaz que ofrece al operador es limitada en comparación con su contraparte. Aquí se hace uso de la pantalla y del teclado presentes en el gabinete del sistema, como se muestran en la figura C.12.

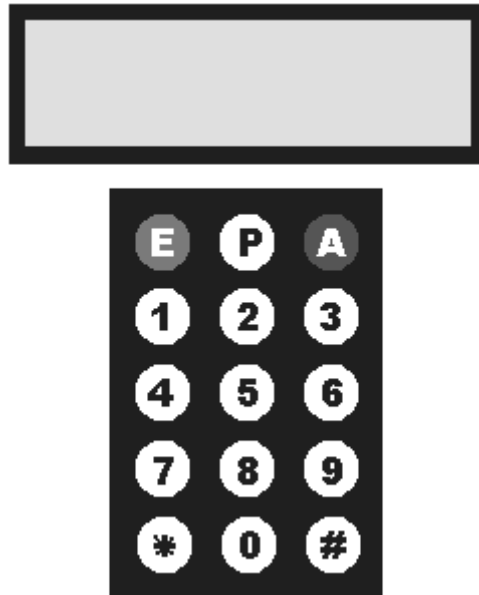


Figura C.12. Pantalla y teclado del gabinete del sistema

La pantalla es alfanumérica, es decir, muestra números y letras en dos líneas de 16 caracteres cada una, lo que significa que no es posible mostrar avisos o mensajes largos y se recurre a mensajes cortos e incluso abreviados. El teclado, por su parte, tiene un total de 15 teclas que incluyen los dígitos del 0 al 9, el símbolo del asterisco (*), el símbolo del número (#) y tres teclas especiales: 'E', que sirve para borrar el valor de algún parámetro de la quema; 'P', que sirve para ingresar el valor del parámetro al sistema (es el equivalente al 'enter', 'entrar' o 'intro' de un teclado de computadora) y finalmente 'A', que seguida de la tecla asterisco sirve para abortar al proceso de quema cuando el operador lo considere pertinente.

NOTA: Las teclas de los símbolos asterisco y número no tienen aplicación alguna en este sistema, por lo que su pulsado no afecta en ningún momento al sistema ni al proceso de quema.

Iniciando una quema con el MC o sistema

Las figuras de esta sub-sección que son relativas a los avisos en la pantalla del sistema muestran un tipo de letra que no corresponde a la real debido a que en este documento no se dispone de un tipo de letra que sea semejante. Se hace recordar que el MC sigue la misma lógica del procedimiento de quema, la diferencia es la forma en como lo inicia.

El primer paso es, obviamente, encender al sistema con el botón de encendido en el gabinete. Algo que no se mencionó en C.4.1 (control por computadora) es que al activar el botón de encendido, la pantalla del sistema muestra el siguiente aviso:



Figura C.13. Aviso en la pantalla sobre la espera de inicio de sesión.

Este aviso indica que el sistema está a la espera del inicio de sesión (que a la postre incluirá el inicio de la quema y su monitoreo). Si el operador optó por usar la PC, realiza el procedimiento explicado en C.4.1 y la pantalla se remitirá a mostrar el siguiente aviso:



Figura C.14. Aviso en la pantalla sobre el control del proceso

Que durará mientras la computadora tenga el control del proceso de monitoreo.

Pero si el operador opta por iniciar el proceso desde el teclado, lo único que debe hacer es pulsar cualquier tecla del teclado del gabinete. Entonces el sistema revisará el estado de la última quema realizada. Si considera que es posible que los datos no hayan sido registrados en archivo, mostrará el siguiente aviso:



Figura C.15. Alerta al usuario sobre datos en el sistema

El operador entonces tendrá que tomar una decisión: si pulsa '1', elige la opción "Esp" o "Espera", que le indicará al sistema que esperará el proceso de quema, haciendo que éste vuelva a un estado original y muestre nuevamente el aviso de la figura C.13, el operador por su parte, debe iniciar el proceso de descarga de datos explicado en C.4.1. Pero si pulsa '2', elige la opción "Cont" o "Continuar", que le indicará al sistema que desea continuar e iniciar la quema, a pesar de que los datos anteriores se perderán.

En cualquier caso, el sistema pedirá al operador que confirme su decisión mostrándole el siguiente aviso:



Figura C.16. Confirmación de decisión

Si el operador elige ‘1’, simplemente confirma su decisión; pero si elige ‘2’, el sistema regresa al estado de la figura C.15.

Captura de parámetros desde teclado

Suponiendo ahora que de cualquier manera el proceso va a iniciarse desde el mismo sistema, lo que sigue es la captura de parámetros. En cada uno de ellos, el sistema verifica que el valor ingresado esté en un rango permisible, de no ser así, borra el valor y seguirá así hasta que el operador ingrese un número válido. El orden en que pide los datos es:

- *Temperatura objetivo* (en °C). Valor mínimo de 573, valor máximo de 1200g. En la figura C.17(a) se observa que el texto dice “Temp. obj. (gC)”, donde “gC” significa “°C” debido a que la pantalla no puede desplegar el símbolo de grados.
- *Hora de inicio*. Usa el formato de 24 horas, no admite un valor mayor a 23:59. Ver figura C.17(b)
- *Fecha*. Se ingresan de manera consecutiva, el día (mín. 1, máx. 31), el mes (mín. 1, máx. 12) y el año (00 al 99). En la figura C.17(c) se muestra la captura del día, pero los otros dos son idénticos. Este es el único parámetro que no se verifica totalmente, pues no se consideran años bisiestos ni que hay meses con menos de 31 días, por lo que el operador debe ser cuidadoso al introducir los datos.
- *Tasa de muestreo* (en minutos). Valor mínimo de 1, valor máximo de 10. Ver figura C.17(d)

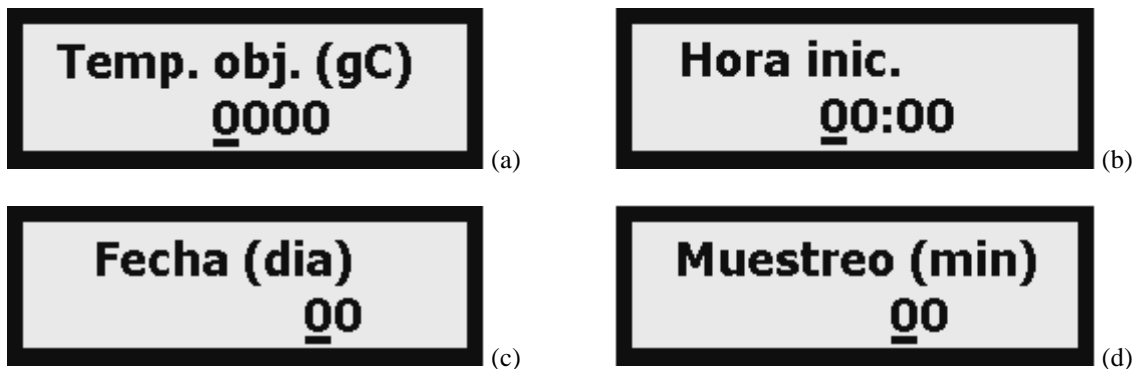


Figura C.17. Avisos para captura de los distintos parámetros. (a) Temperatura objetivo, (b) Hora de inicio, (c) Fecha, (d) Tasa de muestreo.

Cómo se ingresan los valores desde el teclado

Para cualquier parámetro, se observa que en todo momento uno de los dígitos que componen el valor del parámetro tiene una pequeña raya debajo de él. Esa raya es el cursor y si el operador pulsa una tecla de número, cambia el valor actual por el número pulsado. El orden de corrimiento del cursor es de izquierda a derecha, o lo que es igual, del dígito más significativo al menos significativo.

Tomando como ejemplo a la temperatura objetivo, si el operador desea ingresar el valor 1050, tiene que oprimir en orden, ‘1’, ‘0’, ‘5’, ‘0’. Si desea ingresar 850, el orden sería ‘0’, ‘8’, ‘5’, ‘0’. Para ingresar el valor tiene que pulsar la tecla ‘P’ (color amarillo). Este proceso es igual para los demás parámetros. Si el operador ha cometido un error y desea borrar el dato, debe pulsar la tecla ‘E’ (color verde) y el valor adquiere un valor igual a cero.

Nota: Todos los valores capturados son del tipo entero.

Seguimiento del proceso con el sistema

Después de la captura de los datos, el proceso del monitoreo de la quema ha iniciado propiamente, para ello el sistema activa la electroválvula y el operador debe encender los quemadores del horno y realizar los ajustes en la presión que crea convenientes. Como se ha declarado anteriormente, la lógica de la quema es la expresada en la sección C.3 y en cuanto al despliegue de la temperatura interior del horno se refiere, la pantalla muestra la siguiente información.



Figura C.18. Despliegue de información en la pantalla durante la quema.

La primera línea de la pantalla muestra el último evento de la quema y será discutida más adelante. La segunda línea tiene dos elementos: la temperatura actual (sin la leyenda “gC” o “°C”) en formato de punto flotante con un decimal de precisión (la precisión es de 0.5) y hasta cuatro dígitos después del punto (recordar que la temperatura máxima permitida por el sistema es de 1200°C); el otro elemento es la etapa en que se encuentra el proceso, es decir, ‘Y’ tiene un valor de 1, 2 o 3 (ver sección 3).

NOTA: A partir de que la quema ha iniciado, la tecla ‘A’ (color rojo) en el teclado del sistema servirá para abortar el proceso. El operador no debe pulsarla a menos que lo considere prudente.

Avisos de eventos durante la quema.

Los avisos de eventos a los que se refiere la figura C.18 no son otra cosa más que puntos de la quema tales como la inversión del cuarzo, la solicitud de incrementar o decrementar presión, etc. En cualquier caso, el aviso va acompañado de una señal audible para que el operador se entere a tiempo. Lo que se quiere dar a entender es que si por ejemplo, es necesario que se incremente la presión de las válvulas, de no haber señal audible el operador podría no enterarse, pues no es posible que esté todo el tiempo atento a la pantalla del sistema.

A continuación se describen los avisos mostrados y su significado.

- “Inv. cristob.” ➔ Se ha alcanzado la inversión de la cristobalita.
- “Inv. cuarzo” ➔ Se ha alcanzado la inversión del cuarzo.
- “Temp. sanc.” ➔ Se ha alcanzado la temperatura de cocción del sancocho.
- “Rep. esmalte” ➔ Se ha alcanzado la temperatura de reposo o maduración del esmalte.
- “Inc. presion” ➔ Solicitud de incrementar la presión del gas en ambas válvulas.
- “Dec. presion” ➔ Solicitud de decrementar la presión del gas en ambas válvulas.
- “Enfriando...” ➔ Se ha apagado la electroválvula y el horno comienza a enfriarse.

Así, por ejemplo, si en determinado momento el operador escucha un sonido proveniente del gabinete del sistema y al revisar la pantalla ésta contiene:



Figura C.19. Ejemplo de interpretación del contenido de la pantalla del sistema

Significaría que el evento reciente es una solicitud de incrementar la presión del gas de acuerdo a la etapa en que se encuentra la quema, que la temperatura actual del horno es de 645.0°C y que el proceso se encuentra en la primera etapa,

Terminando el proceso con el sistema

Cuando el proceso termine, el sistema emitirá una señal audible más prolongada de lo normal y en la pantalla aparecerá lo siguiente.



Figura C.20. Aviso de proceso de quema terminado

Internamente se prepara al sistema para que la próxima vez que el operador intente hacer una quema, se le avise que hay datos no registrados en archivo, para que primero los descargue por el procedimiento ya explicado. Después de ver este aviso, el operador debe apagar al sistema.

Abortando el proceso con el sistema

De manera idéntica a la terminación del proceso, aparece un aviso semejante en la pantalla y el sistema hace los preparativos para la próxima quema. El sonido emitido es más prolongado de lo normal. Después de ver este aviso, el operador debe apagar al sistema.



Figura C.21. Aviso de proceso de quema abortado.

Descargando datos

Cuando se trató el tema del control del proceso por computadora, se mencionó la tarea de descargar los datos desde el MC, para ello obviamente hay que encender al sistema y lo que aparece en primera instancia en la pantalla es lo mostrado en la figura C.13. Al elegir el operador la opción "Descargar datos desde MC", la pantalla muestra el aviso de la figura C.22(a), si todo concluye satisfactoriamente, se despliega lo mostrado en C.22(b) y la próxima vez que el operador intente hacer una quema, ya no le saldrá el aviso de los datos. Si hubo problemas

durante la descarga, el programa en computadora da su aviso propio al operador y en la pantalla del sistema aparece el aviso mostrado en C.22(c). En este caso, la próxima vez que el operador quiera hacer una nueva quema, el aviso de los datos volverá a salir.

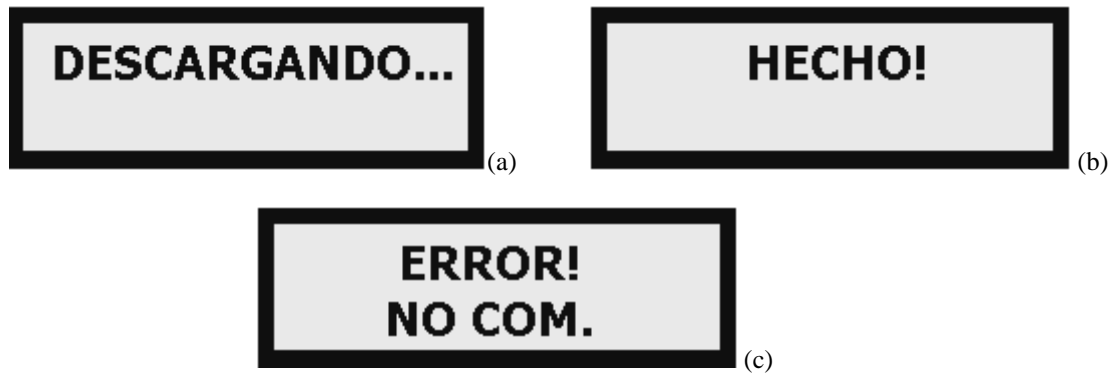


Figura C.22. Avisos relativos a la descarga de datos. (a) Mientras dura la descarga, (b) Cuando la descarga ha sido completada con éxito, (c) Cuando hubo problemas de comunicación durante la descarga.

C.5 Especificaciones técnicas

Físicamente, el sistema tiene las siguientes especificaciones:

- Dimensiones del gabinete (Largo \times Ancho \times Alto): 405 \times 180 \times 170 mm.
- Consumo aproximado de potencia eléctrica: 8.9 VA
- Respaldo aproximado de energía por el UPS (completamente cargado): 1.63 horas.
- Especificación del termopar: Tipo K con termopozo de alúmina de aproximadamente 430 mm de largo.
- Tipo de electroválvula: Modelo S201GF2B5EG5, marca GC Valves, de dos vías, normalmente cerrada, compatible con tuberías de $\frac{3}{4}$ de pulgada, presión máxima de 150 psi para gas, funciona con corriente alterna.

Organización del gabinete

Cuando la situación lo amerite, hay ciertas partes del gabinete que pueden ser revisadas por el operador para reemplazar dispositivos dañados.

Al abrir el gabinete, éste tiene la forma mostrada en la figura C.23. Ahí, los cuadros blancos representan las tres tarjetas más importantes del gabinete y que en determinado momento pueden sufrir daños en algún o algunos elementos de ellas. Esos elementos son dispositivos electrónicos que en determinado momento podrían descomponerse, pero su reemplazo es sencillo. Posteriormente se dará una breve descripción de las tarjetas más importantes y se indicarán los elementos a reemplazar y su posición en la tarjeta.

Por cierto, los cuadros negros de la misma figura también representan tarjetas electrónicas, pero el deterioro de sus elementos es mucho menos probable que en los cuadros blancos.

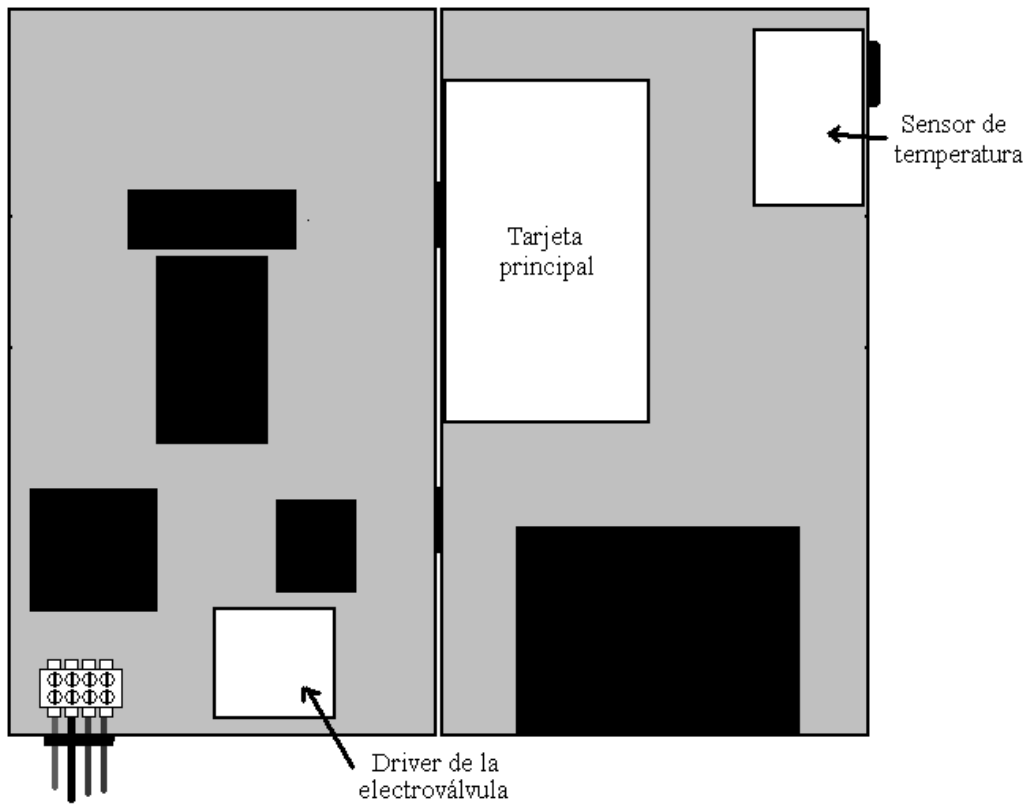


Figura C.23. Interior del gabinete

Tarjeta principal

Contiene al microcontrolador que representa al sistema y varios dispositivos electrónicos más, el único elemento que podría en determinado momento dañarse es el marcado como U1, se trata de un multiplexor analógico y tiene efecto directo en el despliegue de la temperatura del horno. En la tarjeta, U1 = CD4051 de Fairchild Semiconductor.

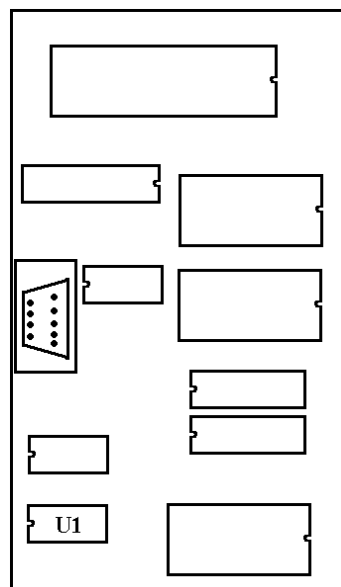


Figura C.24. Tarjeta principal

Driver de la electroválvula

Se encarga de accionar la electroválvula. Los dos dispositivos que podrían dañarse son el fusible marcado como F1 y el circuito integrado U2 que se trata de un optoacoplador. En la tarjeta, F1 = fusible de 1A estándar europeo, U2 = MOC3031 de Freescale Semiconductor.

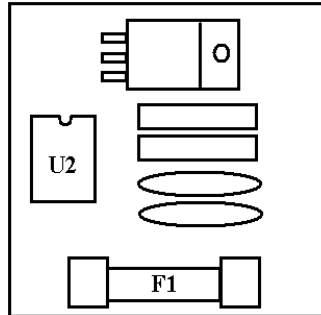


Figura C.25. Driver de la electroválvula

Sensor de temperatura

Se encarga de adaptar la medición del termopar en una señal eléctrica que sea convertida a temperatura. Los elementos que podrían dañarse son U3 y U4, que son circuitos integrados que hacen la adaptación mencionada. En la tarjeta, U3 = LT1025ACN8 y U4 = LT1012, ambos de Linear Technologies.

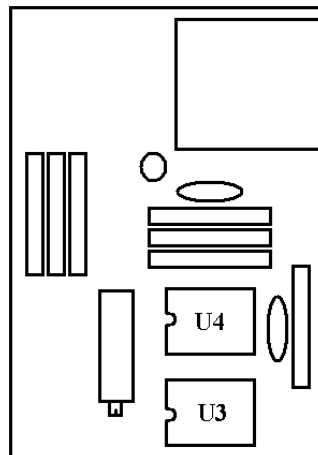


Figura C.26. Sensor de temperatura

C.6 Solución de problemas

A continuación se abordarán los problemas más comunes que podrían presentarse con el sistema y su solución respectiva.

P. *El programa por PC me indica que se ha perdido la comunicación en cualquier intento de inicio de quema o descarga de datos. ¿Cuál será el problema?*

S. La primera posibilidad es que el puerto de comunicación no es el indicado, revisar en la computadora cuantos puertos hay y verificar que sea el elegido en el programa. La otra posibilidad es que el cable esté defectuoso y necesita reemplazarse.

P. *La computadora no tiene puerto serial, pero sí varios puertos USB. ¿Es imposible utilizar el sistema?*

S. No. En la actualidad se pueden adquirir dispositivos conocidos como adaptadores USB-serial. Son varias las compañías que los producen y hay de varios precios.

P. *El programa por computadora está fallando, pues no se ejecuta o se cierra con frecuencia.*

S. Dado que este programa corre en ambiente Microsoft Windows, lo primero a hacer es desinstalar el programa mediante la aplicación “Quitar o agregar programas” en el Panel de Control de Windows. Después se recomienda reiniciar la computadora para volver a instalar el programa en cuestión. Si hubo problemas durante la instalación o el programa sigue teniendo el mismo comportamiento lo más seguro es que la computadora necesita mantenimiento. Acudir en dado caso con un técnico en mantenimiento de computadoras.

P. *El indicador de temperatura nunca cambia pese a que la temperatura notoriamente lo está haciendo.*

S. El multiplexor analógico en la tarjeta principal se ha dañado y es necesario reemplazarlo. En la sección 5 se indica la ubicación del chip y el modelo que debe ser adquirido para su reemplazo.

P. *La temperatura mostrada tiene valores que no pueden ser considerados como válidos.*

S. Alguno de los chips de la tarjeta del sensor de temperatura se ha dañado y hay que reemplazarlo. En la sección 5 se indica la ubicación de los chips y los modelos que deben ser adquiridos para su reemplazo.

P. *La electroválvula no se activa.*

S. Primero hay que revisar la tarjeta de su driver. Revisar el fusible y si está en buenas condiciones, la otra explicación es que el optoacoplador se ha dañado. Reemplazar el o los dispositivos adecuados. En la sección 5 se indica la ubicación del chip y del fusible y los modelos que deben ser adquiridos para su reemplazo. Si esto no funciona, existe una tercera opción: la electroválvula se ha dañado y hay que adquirir una nueva (el modelo se menciona en la sección 1).

BIBLIOGRAFÍA

- [1] Rado, Paul; “Introducción a la Tecnología de la Cerámica”; Ediciones Omega; 1990; España.
- [2] Enciclopedia; “La Electrónica en la Industria”; Volumen 2; Ediciones Orbis/Marcombo; 1986, España.
- [3] Peatman, John B.; “Design with Microcontrollers”; McGraw-Hill; 1988; U.S.A.
- [4] Schackelford, James F.; “Ciencia de materiales para ingenieros”; Pearson Education, 2001, México.
- [5] Webb, John W. & Reis, Ronald A.; “Programmable Logic Controllers: Principles and Applications”; Prentice-Hall, 1999, U.S.A.
- [6] Axelson, Jan; “Serial Port Complete: Programming and Circuits for RS-232 and RS-485 lines and networks”; LakeView Research; 1998; U.S.A.
- [7] Nota de aplicaciones “MCS-51 Microcontroller Family User’s Manual”; Intel Corp.; 1994; U.S.A.
- [8] Referencia técnica “AT28C64B: 64K Parallel EEPROM with Page Write and Software Data Protection”; Atmel Corp.; 2002; U.S.A.
- [9] Nota de aplicaciones “Intelligent Alphanumeric Application Notes”; Pardy Electronics Corp.; 1999; U.S.A.
- [10] Referencia técnica “GAL22V10: High Performance E²CMOS PLD Generis Array Logic”; Lattice Semiconductor Corp.; 2002; U.S.A.
- [11] Referencia técnica “+5V-Powered Multichannel RS-232 Drivers/Receivers”; Maxim Integrated Products; 2002; U.S.A.
- [12] Referencia técnica “ADS774: Microprocessor-Compatible Sampling CMOS Analog-to-Digital Converter”; Texas Instrument Inc.; 1995, U.S.A.
- [13] Nota de aplicaciones “Analog-to-Digital Converter Grounding Practices Affect System Performance”; Texas Instrument Inc.; 1994.
- [14] Manual de referencia para plataforma de desarrollo de aplicaciones LabView; “G Programming Reference Manual”; National Instruments Corp.; 1998, U.S.A.
- [15] Ibrahim, Dogan; “Microcontroller Based Temperature Monitoring and Control”; Newnes Press; 2002, England.
- [16] Referencia técnica “LT1025: Micropower Thermocouple Cold Junction Compensator”; Linear Technologies Corp.; 2003, U.S.A.

Sitios de Internet

- [URL 1] <http://www.tempcomexico.com>
Tempco México S.A. de C.V. 2006,
Diseño y fabricación de resistencias eléctricas calentadoras, sensores y controles de temperatura. Última visita al URL: febrero de 2006.
- [URL 2] <http://www.xtec.es/~aromero8/ceramica/horno.htm>
Cerámica artística y acuarela, 2006,
Sitio dedicado al arte de la cerámica. Incluye tutoriales. Última visita al URL: febrero de 2006.
- [URL 3] <http://www.capgo.com/Resources/Temperature/Thermocouple/Thermocouple.html>
Capgo Pty. Ltd., 2006,
Empresa dedicada a proveer soluciones a la industria y a la investigación. Última visita al URL: febrero de 2006.
- [URL 4] http://wolfstone.halloweenhost.com/TechBase/pnuvlv_SolenoidValves.html
Explicación sobre electroválvulas o válvulas solenoides". Última visita al URL: febrero de 2006.
- [URL 5] http://en.wikipedia.org/wiki/Uninterruptible_power_supply
Wikipedia, 2006,
Enciclopedia on-line. Última visita al URL: febrero de 2006.
- [URL 6] http://en.wikipedia.org/wiki/Data_acquisition_system
Wikipedia, 2006,
Enciclopedia on-line. Última visita al URL: febrero de 2006.
- [URL 7] http://www.albertolacalle.com/hci_principios.htm
AlbertoLacalle.com, 2006,
Diseño de tecnología para personas. Última visita al URL: febrero de 2006.
- [URL 8] <http://www.intel.com/design/mcs51>
Intel Corp., 2006,
Fabricante de la familia MCS-51. Última visita al URL: febrero de 2006.
- [URL 9] http://www.freebsd.org/doc/en_US.ISO8859-1/articles/serial-uart.
The freeBSD Project, 2006,
Sistema operativo avanzado para procesadores x86 y compatibles. Última visita al URL: febrero de 2006.
- [URL 10] <http://www.fsinc.com>
Franklin Software Inc., 2006,
Compiladores C para diversas familias de microcontroladores. Última visita al URL: febrero de 2006.
- [URL 11] <http://www.gcvalves.com/agAir.htm>
GC Valves, 2006,
Fabricante de electroválvulas para la industria. Última visita al URL: febrero de 2006.
- [URL 12] <http://www.tripplite.com/>
Tripp-Lite Corp., 2006,
Fabricante de sistemas respaldo de energía. Última visita al URL: febrero de 2006.

- [URL 13] http://omega.ilce.edu.mx:3000/sites/ciencia/volumen1/ciencia2/11/htm/sec_20.html
Instituto Latinoamericano de la Comunicación Educativa, 2006,
Explicación sobre leyes de la termodinámica. Última visita al URL: febrero de 2006.
- [URL 14] <http://www.ceramicindustry.com>
La voz global exclusiva de los negocios de cerámica y vidrio y su manufactura, 2006,
Sitio web dedicado a la difusión de información de la industria cerámica y del vidrio.
Última visita al URL: abril de 2006.
- [URL 15] http://www.google.com.mx/search?hl=es&lr=&defl=en&q=define:Monitoring&sa=X&oi=glossary_definition&ct=title
Google TM, 2006,
Búsqueda de definición de la palabra monitoring o su equivalente en español,
monitoreo. Nota: se puede obtener el mismo resultado ingresando en la caja de
búsqueda de Google la siguiente frase: *define:monitoring*. Última visita al URL: abril
de 2006.
- [URL 16] [http://www.ceramicindustry.com/CDA/Archives/
f3c67184a6ac7010VgnVCM100000f932a8c0_____](http://www.ceramicindustry.com/CDA/Archives/f3c67184a6ac7010VgnVCM100000f932a8c0_____)
Ensayo “Conexión de Hornos: Interpretando los resultados de los Data Loggers”, 2006,
Ensayo sobre cómo los registros de quemas pueden ayudar a mejorar procesos en la
industria cerámica. Última visita al URL: abril de 2006.
- [URL 17] <http://www.veriteq.com/thermocouple-data-logger/>
Veriteq Instruments Inc., 2006,
Fabricante de data loggers. Última visita al URL: abril de 2006.
- [URL 18] <http://www.ortonceramic.com/controllers/standalone3000.shtml>
Orton Foundation, 2006,
Fabricante y distribuidor de instrumentos de análisis para la industria cerámica. Última
visita al URL: abril de 2006.
- [URL 19] [http://www.datapaq.com/Home.htm?url=http://www.datapaq.com/additional_info/
temperature_monitoring.htm](http://www.datapaq.com/Home.htm?url=http://www.datapaq.com/additional_info/temperature_monitoring.htm)
DataPaq, 2006,
Sistemas de monitoreo de temperatura para industrias. Última visita al URL: abril de
2006.
- [URL 20] <http://www.ilustrados.com/publicaciones/EEkyyFEkpEELyzhaeW.php#superior>
Instrumentación de Horno de Gas y de Inducción, 2005,
Proyecto de instrumentación para dos tipos de hornos. Última visita al URL: abril de
2006.