

5
24



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

**BIKO; UNA INTERFASE EN COMPUTADORA PARA SUPERVISION
Y CONTROL DE UN PROCESO INDUSTRIAL ESPECIFICO**

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A N
GERARDO ARROYO LOPEZ
JOSE JOEL GONZALEZ MARTINEZ
AARON PEREZ HERNANDEZ

DIRECTOR DE TESIS
ING. CARLOS TAKAHASHI FLORES



MEXICO, D.

FALLA DE ORIGEN

1990



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Nuestro más sincero agradecimiento a todo aquel que, de una u otra forma, participó en la realización del proyecto presentado en esta tesis, con especial mención a los siguientes:

- A nuestro director de tesis, Ing. Carlos Takahashi Flores, principal promotor del proyecto, cuyas enseñanzas y apoyo fueron fundamentales.
- Al Ing. Jorge Álvarez Losada, gerente general de Enterprise S.A., por la confianza depositada en nosotros durante todo el proyecto.
- A la empresa Enterprise S.A., que en todo momento nos dieron su respaldo y asesoría.
- A la empresa Forjamex S.A., donde nunca carecimos de su importante apoyo y en especial al Ing. Jesús González por las facilidades otorgadas en la planta.
- Al Banco Nacional de México S.N.C. y en especial al Ing. Alfonso Heredia, por los recursos proporcionados para la elaboración de esta tesis.
- A Alma Cárdenas y Carmen Sánchez por su invaluable colaboración para la culminación del presente documento.
- A Marilú Lopez, Lucero de León y Miguel Pérez, por sus importantes aportaciones y ayuda.
- A la Facultad de Ingeniería, a la que debemos nuestra insustituible formación profesional.
- A nosotros.

¡Muchas Gracias!

***Puedes soplar y apagar la flama de una vela
Pero no puedes soplar y orillar un incendio a ceder
Una vez que las llamas parezcan extinguirse
El viento soplará para hacerlas renacer.***

***"Biko"
Peter Gabriel
1980***

TABLA DE CONTENIDO

Objetivo.	i
Introducción.	ii
Prefacio.	vii
Capítulo 1. Conceptos básicos del control industrial.	1
Capítulo 2. Contexto del proyecto.	5
2.1 Antecedentes Históricos del Control Industrial Eléctrico.	7
2.1.1 El Control y el Transistor.	7
2.1.2 Control Distribuido de Procesos.	9
2.1.3 Data Highways y LAN's.	11
2.2 La Tecnología en el Futuro.	12

2.3 Situación Actual en México.	12
--	-----------

Capítulo 3. Presentación del Proyecto.	15
---	-----------

3.1 Automatización de Procesos Industriales.	15
3.2 Definición del Problema.	16
3.3 Alternativas y Selección de la Solución.	26
3.3.1 Subproblema I.	26
3.3.2 Subproblema II.	28

Capítulo 4. Introducción a los Controladores Lógicos Programables.	33
---	-----------

4.1 Hardware Básico de un PLC.	33
4.2 Programación de un PLC.	37

Capítulo 5. Solución Propuesta: Background.	45
--	-----------

5.1 Diseño de la Arquitectura.	45
5.1.1 Número y Tipo de Entradas y Salidas.	46
5.1.2 Distribución de Módulos y Asignación de Direcciones.	55

5.2 Programa que Goberna la Arquitectura.	72
--	-----------

Capítulo 6. Solución Propuesta: Foreground.	99
--	-----------

6.1 Requerimientos Generales.	99
6.2 Niveles de Comunicación Foreground-Background.	106
6.2.1 Nivel Físico	107
6.2.2 Nivel de Comunicación de Datos	109
6.2.3 Nivel de Comunicación de Información	109
6.3 Funciones de Comunicación del Programa C en T-35.	110
6.4 Listado del Programa BASIC de Interfase PLC -- T-35.	116

Capítulo 7. Desarrollo del Sistema BIKO.	123
---	------------

7.1 Metodologías de Desarrollo	123
7.2 Fase Inicial del Desarrollo del Sistema Biko.	126

7.3 Características del Sistema Biko	127
7.4 Descripción Semidetallada de los Módulos	130
7.5 Documentación Técnica.	132
M. Manual de Usuario.	133
M.1 Ejecución.	133
M.2 Supervisión Analógica.	136
M.2.1 Supervisión Analógica.	136
M.2.2 Tendencias.	138
M.2.3 Supervisión en Lote.	140
M.3 Monitoreo.	141
M.3.1 Temple.	141
M.3.2 Cesta.	143
M.3.3 Quemadores.	144
M.3.4 Revenido.	146
M.4 Paro.	147
M.4.1 Paro Temple.	147
M.4.2 Paro Revenido.	147

M.5 Arranque.	147
M.5.1 Arranque Temple.	147
M.5.2 Arranque Revenido.	147
M.6 Configuración Automática y Lotes.	148
M.6.1 Configuración Automática Temple.	148
M.6.2 Configuración Automática Revenido.	149
M.6.3 Carga Lote.	150
M.6.4 Tendencias Lote.	153
Conclusiones.	157
Apéndice A. PLC 5-15.	161
A.1 Especificaciones Técnicas del PLC 5-15.	161
A.2 Resumen del conjunto de instrucciones del PLC 5-15.	161
A.2.1 Almacenamiento de Datos.	161
A.2.2 Modos de Direccionamiento.	163
A.2.3 Instrucciones Lógicas de Relevador.	170
A.2.4 Instrucciones de Timers y Contadores.	171

Tabla de Contenido

A.2.5 Instrucciones Aritméticas, Lógicas y de Transferencia.	173
A.2.6 Instrucciones de Archivos.	177
A.2.7 Instrucción PID.	183
A.2.8 Instrucciones de Transferencia de Bloques.	185

Apéndice B. Módulo BASIC.

191

B.1 Características Generales.	191
---------------------------------------	------------

B.2 Características de Hardware.	191
---	------------

B.3 Descripción de Pines.	192
----------------------------------	------------

B.4 Especificaciones Técnicas.	195
---------------------------------------	------------

B.5 Resumen del Conjunto de Instrucciones.	195
---	------------

B.5.1 Guía de referencia.	196
---------------------------	-----

B.5.2 Instrucción CALL.	201
-------------------------	-----

Apéndice C. Terminal Industrial T-35.

205

C.1 Especificaciones Técnicas.	205
---------------------------------------	------------

C.2 Herramientas de Programación.	205
--	------------

Apéndice D. Tablas de Correspondencia de Datos entre PLC 5-15, Módulo BASIC y T-35.	209
D.1 Tabla de Datos del PLC 5-15 por Archivos.	209
D.2 Relación de Datos Transferidos al Módulo BASIC desde el PLC 5-15.	212
D.3 Relación de Datos Transferidos a T-35 desde Módulo BASIC.	213
D.4 Relación de Datos Transferidos de T-35 a PLC 5-15 vía 1771-DB.	213
Bibliografía.	217
Referencias.	219

OBJETIVO.

El objetivo de este documento es presentar el diseño y desarrollo de una interfase en computadora para Supervisión y Control de un proceso industrial específico.

Introducción

INTRODUCCION.

La Computación ha tenido y seguirá teniendo aplicaciones cada vez más diversas y en las áreas menos sospechadas. Así, tenemos su incursión en el área de Salud, Comunicaciones, Servicios Bancarios y Sistemas Militares de Defensa, por citar algunos ejemplos. Sin duda, la Computación es una herramienta de la cual todos pueden hacer uso y el caso de la Automatización Industrial, no podía ser la excepción.

En el ámbito industrial, el empleo de herramientas adecuadas resulta imprescindible para lograr mayor eficiencia en cualquier actividad humana. La posibilidad de emplear a la Computadora como instrumento o unidad de mando, siendo ésta una herramienta de propósito general, depende del grado de desarrollo alcanzado en el *hardware* y de la existencia de *software* que permita explotar los logros del primero.

Uno de nuestros propósitos a demostrar mediante esta tesis es que uno de los caminos de México en la Computación a corto plazo es la creación de *software* que permita el mejor aprovechamiento de los recursos proporcionadas por el *hardware*. Estamos convencidos que demostraremos lo anterior a través de un caso concreto: en México no se desarrolla, por el momento y hasta donde sabemos, *software* para Supervisión y Control de procesos industriales explotando la tecnología de las Computadoras Personales y los Controladores Lógicos Programables o PLC's (*PLC : Programmable Logic Controller*). En Estados Unidos existen varios paquetes comerciales a precios de varios miles de dólares y que, además implican unos cuantos años de dependencia, en este sentido, para los mexicanos.

En nuestra tesis desarrollamos una aplicación específica de este tipo, a un precio inferior, con la misma calidad y sobre todo, cimentamos bases para el desarrollo de un paquete nacional de aplicación general que puede competir en mercados extranjeros.

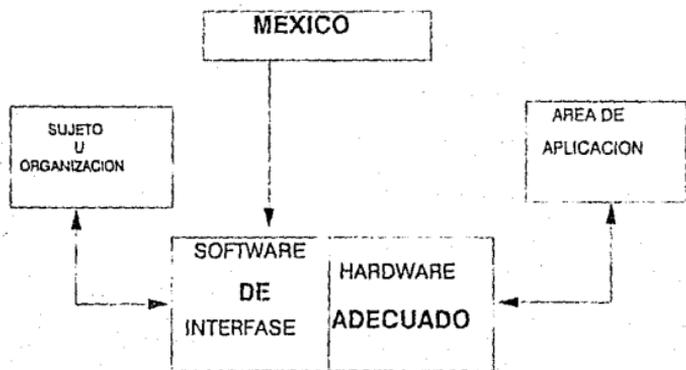


FIGURA I-1

Por lo pronto, en esta introducción describiremos, sin profundizar, el contexto bajo el cual se elaboró la tesis, así como sus características básicas. Con ello, el lector estará mejor ubicado acerca de la naturaleza del proyecto expuesto en el presente documento.

El proyecto **BIKO** tiene su aplicación dentro del área industrial y particularmente dentro del Control industrial mismo. El porqué del nacimiento de **BIKO** tiene su respuesta en una necesidad existente de la actividad empresarial: la Automatización de un proceso industrial, conservando desde el inicio la idea de satisfacer completamente los requerimientos de Control y más aún, mejorar los tradicionales sistemas de Supervisión que actualmente operan en el campo del Control Automático.

BIKO es un sistema computacional que se encuentra operando en una importante planta industrial de construcción de piezas mecánicas de la rama automotriz: FORJAMEX S.A.. Dicha empresa contrató los servicios de una segunda, llamada ENTERPRISE S.A., cuya actividad radica en prestar servicios de Automatización y Control Industrial para resolver problemas tipo al que en su oportunidad dió origen al proyecto **BIKO**. El lector deberá entender que el proyecto es propiedad privada y por lo tanto no podremos citar características detalladas del proceso fabril que

pongan en evidencia secretos industriales de las empresas mencionadas.

Como ya se citó anteriormente, **BIKO** está basado en una situación real y detrás de la cual se identifican tres partes fundamentales: la industria FORJAMEX S.A. que contrató el proyecto, la empresa ENTERPRISE S.A., la cual vendió el servicio y los tres estudiantes que conformamos el grupo de trabajo autor de la presente tesis. Los primeros fueron la semilla para el planteamiento del proyecto, los terceros son los protagonistas que realizaron el diseño e implantación del mismo, del cual hablaremos a continuación.

Para describir el proyecto comenzaremos mostrando, en la siguiente figura, un bosquejo general del entorno bajo el cual opera:

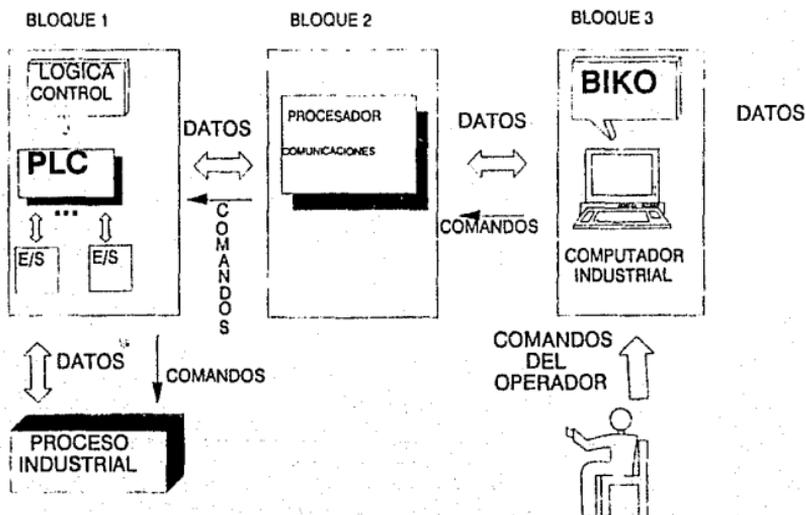


FIGURA I-2

El diagrama agrupa al sistema en tres bloques. En el primer bloque encontramos un Controlador Lógico Programable, en cuya memoria reside un conjunto de instrucciones que, al ejecutarse, activan módulos de Entrada/Salida (E/S en lo sucesivo) digitales y analógicos de acuerdo a una estrategia de Control que es particular de la aplicación. Este bloque es la parte medular del Control del proceso industrial.

El tercer bloque está formado por una computadora industrial compatible con la IBM PC tipo AT. Es en esta computadora donde cumple sus funciones el Programa de Interfase para Supervisión y Control de un proceso industrial específico, objetivo perseguido en nuestra tesis y al cual hemos bautizado como **BIKO**.

El bloque intermedio esquematiza un módulo de comunicaciones cuya función general es intercambiar datos y comandos entre los bloques 1 y 3. Este bloque está conformado físicamente por una computadora programada a través de un lenguaje BASIC extendido que resulta apropiado para dicha labor.

La figura 1-2 muestra la interrelación de tres módulos que en conjunto conforman la arquitectura del sistema, cuya finalidad podemos resumirla como sigue. Tenemos un proceso industrial específico, remarcamos una vez más la palabra específico, del cual adquirimos datos a través de módulos de entrada mismos que llegan al Controlador Lógico Programable para que éste los interprete auxiliado por una lógica de Control programada en el mismo. El programa de Control toma acciones pertinentes y realimenta al proceso con señales de salida por conducto de los módulos de salida, es decir, existe un lazo cerrado entre el proceso y el bloque 1.

Al mismo tiempo que el bloque 1 realiza adquisición de datos y Control sobre el proceso, el bloque 2 lee datos de la memoria del PLC para transmitir información al bloque 3 donde **BIKO** realiza sus labores de Supervisión y Control en la computadora personal ya mencionada.

Ahora bien, como todo sistema computacional, **BIKO** debe cumplir con ciertas características. En primera instancia, para facilitar la operación del sistema se pretendió crear un ambiente amigable al usuario a través de menús, teclas asociadas a comandos, ayudas, etc.

Como sistema de Supervisión, **BIKO** proporciona la representación gráfica de la ubicación física de los elementos del proceso en tiempo real

por medio de mímicos. De igual forma tenemos *faceplates* que muestran gráficamente el Control de lazo cerrado de las variables analógicas del proceso. Aunado a esto, tenemos la opción de desplegar e imprimir gráficas valor de variable física vs. tiempo (mejor conocidas como tendencias), sin faltar el cálculo de algunos datos estadísticos de valiosa ayuda al usuario. Otra peculiaridad relevante dentro de la Supervisión del proceso, en el aspecto de seguridad es el registro y reconocimiento de alarmas ocurridas durante el tiempo de operación.

En cuanto a la parte de Control de nuestra interfase tenemos que **BIKO** le permite al usuario elegir entre dos modalidades de Control del proceso: Automático y Manual. En ambas modalidades el operador puede intervenir en el proceso mediante ciertos comandos permisibles según sea el caso. Para la modalidad de Control Automático es posible configurar el proceso dándole una mayor gama de alternativas al operador.

A grosso modo, esto es el Sistema de Interfase para Supervisión y Control de un proceso específico al cual haremos referencia en el transcurso del texto como **BIKO**. A lo largo de la tesis se ampliará lo brevemente expuesto, tanto de la interfase como del sistema de Control con el cual interactúa, pues no podemos tratar a **BIKO** como un sistema aislado.

Prefacio

PREFACIO.

La naturaleza de nuestra tesis encierra algunos factores alrededor que, obligan por momentos a guiar al lector no familiarizado con este tipo de proyectos, por caminos que le ayuden en lo más posible, a una mejor comprensión del texto. El material contenido en este texto está enfocado para cualquier lector interesado en el tema, que posea formación académica o conocimientos en Computación y/o Control.

Es por ello que se incluye el Capítulo 1: Conceptos Básicos del Control Industrial. Tiene la finalidad de proporcionar un conjunto de conceptos e ideas fundamentales que serán retomadas a lo largo del trabajo. El lector familiarizado con este tipo de conceptos podrá avanzar, si así lo desea, al siguiente capítulo sin pérdida de continuidad.

En el capítulo 2: Contexto del Proyecto, hacemos un recorrido a los antecedentes históricos del Control Industrial, básicamente en su etapa más contemporánea. Asimismo, para redondear el contexto bajo el cual se desarrolló el proyecto exponemos brevemente cual es la situación actual que priva en México en cuanto a Control Industrial se refiere.

En el capítulo 3: Presentación del Proyecto, incluimos la definición formal del mismo, esto es, cuál es el problema a resolver, qué alternativas tenemos para solucionarlo y la selección de la mejor alternativa. A estas alturas el lector deberá tener una aceptable comprensión de lo que se pretende lograr con el proyecto.

Antes de seguir adelante con la solución propuesta, es importante incluir un apartado con información de apoyo referente a Controladores Lógicos Programables, pues el proyecto BIKO opera en conjunto con un equipo de este tipo. La idea del capítulo 4: Introducción a los Controladores Programables, es proporcionar una introducción que contempla la arquitectura y *hardware* básicos de un PLC. De igual forma se expone como se lleva a cabo su programación en términos generales. Así como en el capítulo 1, esta sección puede omitirla el lector con conocimientos previos en el tema.

En los capítulos 5 y 6, *Background* y *Foreground* de la solución propuesta respectivamente, se desgloza a mayor detalle la solución, tanto en dispositivos de *hardware* como en el *software* requerido para llevar a cabo el desarrollo e Implantación del sistema BIKO.

El capítulo 7 y último incluye todas las consideraciones para el Análisis, Diseño, Desarrollo e Implantación del sistema BIKO resumidas de todo el ciclo de vida del proyecto hasta la fecha en que redactamos las últimas líneas de esta tesis. A su vez, anexamos el Manual de Usuario del sistema BIKO donde se explica como operarlo mostrando las pantallas del sistema.

Para cerrar la exposición textual de nuestra tesis tenemos la sección de Conclusiones, que sin duda es la parte más importante de nuestro trabajo.

En la sección de Apéndices, se incluyen apartados con información técnica referente al PLC 5-15, Módulo BASIC y la Terminal Industrial T-35 que pueden resultar de interés al lector. Dada la escasa literatura sobre estos equipos, esta sección contiene material de potencial ayuda y divulgación.

1. Conceptos Básicos del Control Industrial

1 Conceptos Básicos del Control Industrial.

A continuación se presentan algunos conceptos básicos de sistemas de Control que consideramos deben comprenderse antes de seguir adelante. El lector debe tener presente que los conceptos aquí mencionados son un apoyo para este documento sin pretender ser definiciones formales.

Operación. Una operación es toda aquella acción que puede efectuar un elemento aislado. Para fines prácticos, toda operación se considera indivisible.

Proceso. Denominaremos proceso a cualquier conjunto de operaciones guiadas por el Control hacia un fin específico.

Estado del proceso. El estado de un proceso es el valor (entendiendo como valor la ubicación, posición o grado de accionamiento por ejemplo) de cada uno de sus elementos en un instante específico.

Elemento. Un elemento es todo aquel dispositivo físico que puede efectuar una operación. Para fines prácticos, todo elemento se considera indivisible.

Secuencia de operación. La secuencia de operación para un proceso, se define como una agrupación de estados de dicho proceso, que deben respetar cierto orden y tienen un propósito específico dentro del mismo.

Sistema. Un sistema es una combinación de elementos que actúan conjuntamente y cumplen determinado objetivo.

Sistema de Control de lazo cerrado. Un sistema de Control de lazo cerrado es aquel en el que la señal de salida tiene efecto directo sobre la acción de Control. Ver figura 1-1.

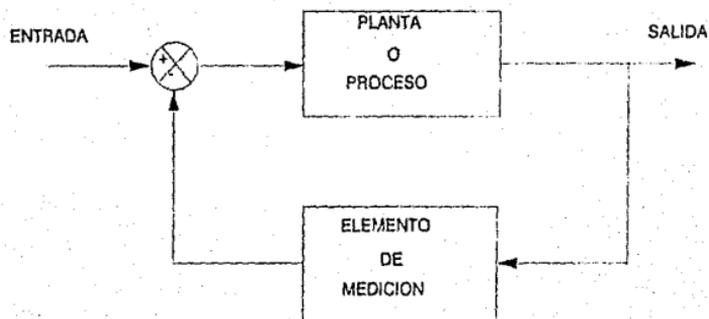


FIGURA 1-1

Para nuestros fines podemos ahondar en la definición anterior para distinguir dos modalidades de Control:

- 1) **Control Automático o en Secuencia Automática.**
- 2) **Control Manual o en Secuencia Manual.**

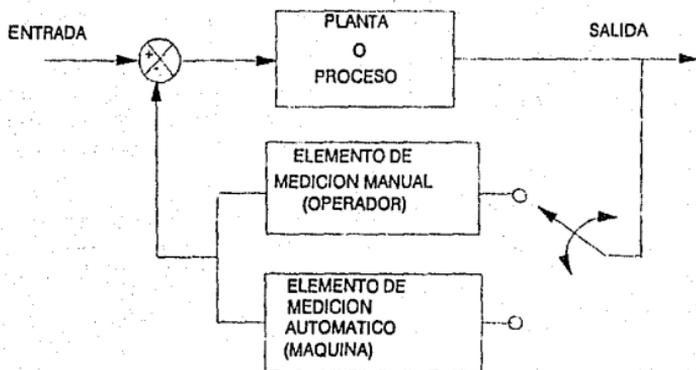


FIGURA 1-2

- 1) **Control en Secuencia Automática.** Por Control en Secuencia Automática debe entenderse un conjunto preestablecido de

estados de un proceso cualquiera que deben ejecutarse respetando un orden dado, sin intervención humana exceptuando la supervisión.

- 2) **Control en Secuencia Manual.** Por Control en Secuencia Manual debe entenderse un conjunto de todos los posibles estados de un proceso cualquiera que pueden ser ejecutados sin seguir un orden fijo, con intervención humana.

Aclaremos estas ideas. Por una parte, el conjunto de estados para el Control en secuencia automática es un subconjunto de todos los posibles estados del Control en Secuencia Manual. Por otro lado, en el primer tipo de Control la secuencia de operación es fija mientras que en el segundo, esta secuencia es variable y por tanto, la responsabilidad implícita es mucho mayor y recae sobre el responsable de la operación manual.

Software en tiempo real. El *software* que mide, analiza y Controla eventos en el mundo real tal como estos ocurren es llamado *software* en tiempo real. El *software* en tiempo real incluye: componentes de recolección de datos que transducen información proveniente de un ambiente externo, un analizador que transforma esa información como es requerida por la aplicación, elementos de Control de salida que interactúan con el ambiente externo, y un módulo de monitoreo que coordine todos los otros componentes de tal forma que la respuesta en tiempo real (típicamente en el rango de 1 milisegundo a 1 minuto) se pueda conservar. Un sistema en tiempo real debe responder dentro de límites de tiempo estrictos acorde con la aplicación.

Faceplate. Representación gráfica de los valores relacionados con el Control de una variable analógica. Esta representación está compuesta por barras proporcionales al valor del parámetro asociado, así como sus valores numéricos. En un *faceplate* se indica el estado de la Variable a Controlar, el valor de Regulación y el valor de la Variable de Realimentación. Así como su estado de operación (Manual/Automático) y las alarmas fijadas.

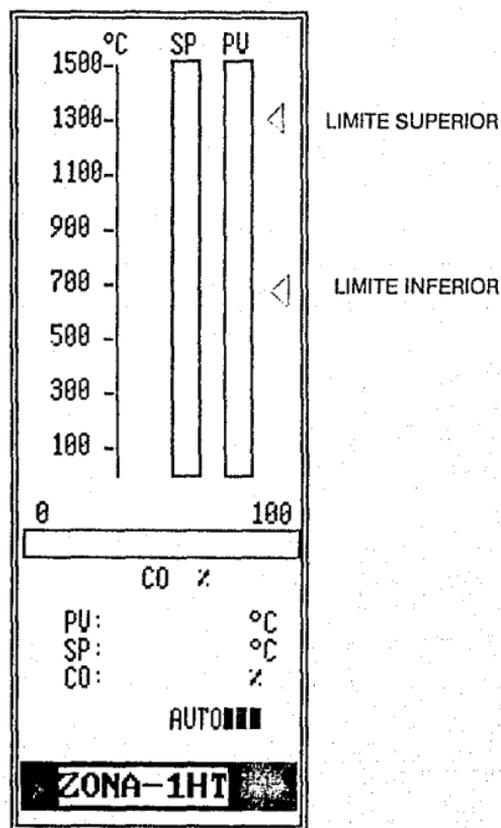


FIGURA 1-3

Alarmas. Una alarma es toda aquella indicación visual o auditiva que se activa cuando una variable o elemento del proceso ha sobrepasado su rangos de operación inferior y/o superior.

Mímico. Esquematización del estado de los diferentes elementos de un proceso con fines de Supervisión: ubicación en el espacio, valores encendido-apagado, status de operación, alarmas, etc. Generalmente son dibujos y lámparas montadas sobre paneles de acrílico.

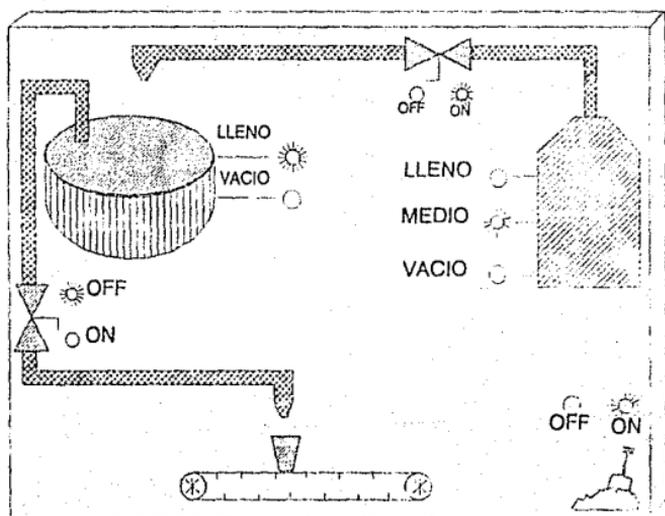


FIGURA 1-4

2. Contexto del Proyecto

2 Contexto del Proyecto.

2.1 Antecedentes Históricos del Control Industrial Eléctrico.

La historia del Control tiene antecedentes muy remotos. No nos remontaremos al siglo XVIII para hablar del regulador centrífugo para el Control de velocidad en la máquina a vapor de Watt. Nuestro estudio partirá, como lo indica el subtítulo, de los sistemas de Control contruidos en base a dispositivos eléctricos.

En los últimos 35 años el desarrollo del Control ha estado condicionado particularmente por la tecnología de la electrónica digital. Desafortunadamente, el Control a su vez, ha influido muy poco en el desarrollo tecnológico en ese tiempo. Tal vez la implacable demanda de la Automatización Industrial sobre las capacidades de las computadoras lograrán que el Control determine una pauta en el futuro. Ciertamente, la búsqueda de soluciones CIM (Manufactura Integral por Computadora) ya exige más capacidad de lo que los actuales distribuidores de computadoras de Control están ofreciendo, aunque la capacidad tecnológica ya está disponible.

Para hacer un análisis retrospectivo, consideremos la tecnología imperante en las distintas épocas para el desarrollo del Control Eléctrico.

2.1.1 El Control y el Transistor.

El transistor ya tenía 6 años cuando se introdujo el primer controlador electrónico comercial en 1954 diseñado con tubos de vacío. El primer controlador comercial transistorizado llegó cinco años después.

Pero la evolución tecnológica del Control durante los últimos 35 años, ha seguido más al desarrollo de las computadoras digitales que al desarrollo de la tecnología del transistor, aún cuando resulta claro que ambas están intrínsecamente ligadas.

Como los Controladores de Procesos, la Computadoras de Control fueron originalmente máquinas de tubos de vacío. En aquellos días, nadie reconocería los relevadores lógicos como computadoras de propósito especial. A los sistemas de Control basados en relevadores lógicos se les conoce como Sistemas de Control Alambrado.

Sin embargo, desde principios de los 50's, la necesidad de manejar una mayor precisión, señales continuas y memoria, llevaron a muchos ingenieros de Control a ver a las computadoras digitales como medios de Control dada la evidente desconfianza hacia los Controladores basados en tubos de vacío. La atención se desvió entonces a ganar la prometida confiabilidad de los transistores y por esto, sin importar lo reciente de la tecnología, para 1960 la mayoría de las computadoras de Control ya eran transistorizadas.

Computadoras como la Ramo Wooldrige RW-300 usaban circuitos de menos de 1000 transistores discretos, y se vendían en alrededor de \$100,000 dólares, incluyendo dispositivos de E/S. La memoria principal tenía 8080 palabras almacenadas en un tambor magnético, con un tiempo de acceso promedio de 8 milisegundos. No fue hasta después de 1968 cuando las minicomputadoras tuvieron disponibles 16K palabras en memoria de ferrita, con 10 microsegundos de tiempo de acceso aleatorio, con costos similares a los sistemas anteriores.

La tecnología del circuito integrado, que permitió infinita variedad de combinaciones de muchos circuitos de transistores manufacturados sobre el mismo chip de silicio, aparecieron por primera vez alrededor de 1963. Los circuitos integrados ofrecieron la promesa de una confiabilidad mucho mayor, debido principalmente al reducido número de conexiones en los circuitos impresos, reducido número de circuitos a conectarse en los sistemas, y reducido uso de energía, que disminuyó la cantidad de calor generado.

Sería hasta 1971 cuando aparecen los primeros microprocesadores de 4 bits, y antes de 1975 los microprocesadores de 16 bits costeables, se llevaron a los Controladores de Procesos y los Sistemas de Control Distribuido de procesos entraron en el mercado.

Por lo tanto, ya que el desarrollo del Control está condicionado por la tecnología de la electrónica digital, a mediados de los setentas hubo un estancamiento de aproximadamente cinco años.

Lo que pasó inadvertido para los fabricantes de computadoras y diseñadores de circuitos integrados, fué que las aplicaciones de Control industrial se encontraban operando consistentemente en un 30%-40% del número total de computadoras y esto aún continúa así. Estos porcentajes son en números de procesadores, no de dólares.

Cuando los circuitos basados en microprocesadores, que se vendían por cientos de dólares, fueron comparados con las computadoras que se vendían por \$20,000 dólares, es cuando éstos son lógicamente incorporados a los Sistemas de Control Distribuido. Así, la importancia económica del mercado de Control resultó poco atractiva para los fabricantes de computadoras y circuitos integrados. Entonces, la atención hacia los ingenieros de Control y sus necesidades fué disminuyendo y las aplicaciones de oficina atrajeron la atención de los fabricantes. Esto fué desafortunado para el Control, pues las aplicaciones de oficina no necesitan la extrema confiabilidad que los sistemas de Control requieren.

2.1.2 Control Distribuido de Procesos.

Antes de que Honeywell introdujera el primer sistema de Control Distribuido de Procesos (DCS), su sistema TDC 2000, el Control en computadoras digitales era Control Digital Directo (DDC), en el cual una sola computadora, debía llevar a cabo docenas e incluso cientos de lazos de Control simultáneamente. El DDC ha sido el punto importante desde 1962, dado que los sistemas han crecido en complejidad después de los comienzos del Control Digital.

La TDC 2000 redujo el número de lazos de Control, que los sistemas de Control basados en un sólo microprocesador tenía que ejecutar, en máximo 8 lazos en la mayoría de los casos. Serios esfuerzos fueron entonces iniciados por algunos fabricantes de equipos de Control, para alcanzar la meta de un sólo microprocesador independiente, por lazo de Control. Su objetivo fué obtener un sólo lazo fallido por microprocesador fallido, para proveer una menor degradación del desempeño y mejorar la disponibilidad completa del sistema. Este objetivo nunca ha sido completamente alcanzado en DCS dado que la creciente confianza en la

funcionalidad de los Controladores basados en microprocesadores ha hecho esto poco atractivo, económicamente hablando. En efecto, la tendencia ahora luce más orientada hacia el otro camino, esto es, hacia más lazos por procesador.

Las terminales de video a color para el operador fueron también introducidas al Control de Procesos como parte de la TDC 2000, si bien ellas ya habían sido usadas anteriormente en los cuartos de Control de distribución de potencia eléctrica.

Pero aún antes de las pantallas de tubos de rayos catódicos (CRT), los Controladores habían sido llevados a los cuartos de Control para que los operadores pudieran ver todos sus indicadores a la vez, en lugar de tener que caminar a varias unidades de proceso para registrar las lecturas y posteriormente analizarlas. Esto fué hecho primero con Controles Neumáticos montados en grandes paredes, a menudo sobre diagramas mímicos del proceso. Los esfuerzos de miniaturización vinieron con la neumática, pero fueron realmente acelerados por el advenimiento de los Controladores electrónicos. La necesidad era llevar los instrumentos lo más cerca posible del operador en consolas, para ganar precisión en las lecturas de los indicadores y ponerlos rápidamente al alcance del operador.

Otros desarrollos importantes fueron la emulación, en monitores de computadoras de diagramas mímicos del proceso y los *faceplates* de un Controlador, llegando incluso a desplegar al mismo tiempo diferentes *faceplates* con gráficas basadas en barras.

Es fácil ver como las estaciones de operador con monitores de color se ha convertido en la última expresión de la tendencia de Supervisión de Control Distribuido de Procesos.

Los ingenieros de Control de potencia eléctrica también usaron lo que ellos llamaron la arquitectura de controlador separado, en la cual los *faceplates* estaban separados de los Controladores. Los indicadores estaban en las consolas de Control y los Controladores fueron montados en *racks* con sus E/S en un cuarto cercano. La TDC 2000 también usó esta arquitectura separada, respaldando los monitores de color de Controladores Digitales con verdaderos Controladores de arquitectura separada.

Por lo tanto, bien se les podría atribuir a las estaciones de operador con monitor a color el haber sido un muy pequeño progreso hacia el

verdadero Control físicamente distribuido desde la introducción de la TDC 2000, sin importar las docenas de sistemas más recientes.

2.1.3 Data Highways y LAN's.

La tercera mayor contribución de la TDC 2000 fué el concepto de Control en Red. Hoy día reconocemos estas redes como *Data Highway* dedicadas, o bien, como Redes de Area Local (*Local Area Network*). Una diferencia básica que permanece, sin embargo, es que las *Data Highways* fueron diseñadas para satisfacer objetivos de Control muy específicos, mientras todas las LAN's de hoy, están diseñadas con motivos más generales de comunicación entre computadoras. En Control, la confiabilidad debe permanecer como objetivo principal y la utilidad universal debe ser sacrificada.

La idea básica de los Sistemas de Control Distribuido, era hacer posible el traslado de los Controladores, a una zona aparte de la zona del proceso. Esto disminuía los lazos de Control y con ello, se ahorra en costos de alambrado, y al mismo tiempo se harían los lazos menos vulnerables al ruido y daños físicos. El objetivo era alcanzar confiabilidad y disponibilidad en el sistema.

Pero los Controladores de lazo aún no se han logrado apartar completamente del proceso. La principal razón ha sido la ausencia de comunicaciones digitales estandar en LAN's a nivel de lazo. Las *Data Highways* afectan las condiciones de operación de lazo de una forma supervisora, pero generalmente están fuera de los lazos de Control.

2.2 La Tecnología en el Futuro.

Las computadoras cada vez son más veloces y el potencial de la Computación es más accesible día a día. Las computadoras personales están ofreciendo velocidades de reloj de 33 MHz, teniendo ya próximos los 40 y 50 MHz. Todo lo requerido por la Manufactura Integral por Computadora está al alcance tecnológicamente. Pero la solución no es con computadoras personales. Los datos en la mayoría de las industrias manufactureras y las plantas de procesamiento hoy en día están almacenados en computadoras dispersas a lo largo de las plantas. La integridad de los datos es inexistente, aún en los sistemas de Control Distribuido.

Con las tendencias del Control a incrementar despliegues en tiempo real de alta resolución, multiprocesamiento en tiempo real, múltiples usuarios, múltiples programadores y accesos en tiempo real a las bases de datos globales del sistema, la necesidad de las arquitecturas de procesamiento paralelo se hace evidente.

2.3 Situación Actual en México.

Hablar de la situación que impera en el sector industrial nacional en forma general, es prácticamente imposible dada la diversidad de actividades empresariales, tamaños y capitales invertidos, tecnologías aplicadas, etc. Aún cuando no somos expertos en el área de Control Industrial, pues debemos mencionar que nos falta mucho por aprender, el proyecto que desarrollamos como tesis, nos ha proporcionado valiosa información para visualizar el contexto que puede caracterizar, a gran parte de las empresas mexicanas en el momento actual. Durante el desarrollo de la tesis contamos con el apoyo de la empresa ENTERPRISE S.A., cuya experiencia nos ayudó a bosquejar el siguiente panorama que resulta preponderante en la Industria de la Transformación en cuanto al área de Control se refiere.

El panorama al que hacemos referencia tiene características que obedecen a un antecedente histórico, es por ello que, para poder situar correctamente el estado actual del Control en la industria nacional, realizamos previamente un breve resumen sobre la evolución del Control en el tiempo.

Una vez hecho lo anterior podemos ubicar al grueso de la industria nacional dentro de la etapa del Control Alambrado.

Según la historia que revisamos anteriormente, sabemos que existe un concepto más reciente en cuanto a tecnología de Control se refiere: PLC's, cabe entonces preguntarse ¿Por qué el uso generalizado del Control Alambrado? ¿Por qué no se ha adoptado la tecnología de los PLC's?

La primer razón es meramente cuestión cronológica, pues el Control Alambrado surge antes que el PLC. Por otro lado, el PLC no ha tenido correcta difusión en México, es necesario hacer del conocimiento de esta tecnología, a profesionistas y demás gente involucrada en el medio. Sucede comunmente que los empresarios toman una actitud temerosa al cambio. Este temor está cimentado en la falta de un adecuado soporte técnico, aún cuando existen varios fabricantes y representantes de Controladores Programables en México, hay carencia de gente lo suficientemente preparada para brindar un servicio completo en asesoría y mantenimiento sin limitarse únicamente a la venta.

Sin embargo, actualmente ya se encuentran operando algunos sistemas de Control basados en PLC's. El número es reducido pero se están dando los primeros pasos en la materia.

3. Presentación del Proyecto

3 Presentación del Proyecto.

3.1 Automatización de Procesos Industriales.

El proyecto que da pie a esta tesis es un proyecto particular de Automatización Industrial, por tal motivo abordaremos ligeramente este tema.

El proyecto **BIKO** es un proyecto computacional que tiene aplicación dentro del Control Industrial. Por lo tanto, si enfocamos la situación desde la raíz, el motivo que dió origen al proyecto es precisamente la Automatización de un proceso industrial. **BIKO** nace debido a la necesidad de automatizar y por ende, controlar procesos industriales pues, para automatizar debemos controlar. ¿Por qué es necesario automatizar y controlar procesos industriales? Esta pregunta tiene varias respuestas.

Un punto clave que se gana al automatizar es el nivel mismo de productividad. Un proceso industrial auxiliado por máquinas controladas automáticamente resultará más veloz en su operación y esto cimentará una ganancia muy considerable en productividad.

Un proceso automatizado eleva el nivel de calidad de los productos manufacturados, pues la intervención directa de máquinas y robots permite explotar la precisión y exactitud de estos. Además, con ello se optimiza el aprovechamiento de recursos materiales pues se reduce la cantidad de materia prima desperdiciada.

Otra razón importante reside en que la Automatización libera al obrero de trabajos que expongan su integridad física, pues en gran cantidad de centros de trabajo, los obreros están propensos a sufrir todo tipo de peligrosos accidentes.

Si bien, la implantación de un proyecto de Automatización Industrial implica una elevada inversión inicial, no olvidemos que una inversión correctamente guiada reflejará sus frutos tarde o temprano. La pregunta ¿cuándo automatizar y cuándo no automatizar? siempre resultará

controvertida. Si se está buscando conjugar las características expuestas en este tema debemos pensar en automatizar. Pero, la Automatización puede tener repercusiones sociales si ésta no se efectúa correctamente.

La Automatización como proceso de cambio debe ser benéfico para el hombre y no por el contrario, desplazar al hombre. Una solución a este problema lo encontraremos en la Automatización Guiada, esto es, el hombre se encarga de realizar funciones de Supervisión, alguien que deberá estar pendiente del proceso Automático en todo momento y que sea capaz de tomar decisiones en un momento dado. La Automatización Guiada libera al trabajador de trabajo pesado, riesgoso, dejándole funciones más inteligentes que actuar como máquina a lo largo de toda la jornada de trabajo.

Ahora bien, ¿qué tiene que ver la Computación con el Control Industrial? Sabemos del Capítulo 2 que el Control Industrial surge antes que los sistemas electrónicos. luego entonces, se puede realizar Control prescindiendo de las computadoras. Pero la tecnología sigue su paso y ahora la Computación puede aportar sus ventajas para diseñar sistemas de Control más eficientes y versátiles. Un PLC es un sistema digital con capacidades de programación y comunicaciones orientado a aplicaciones de Control. El proyecto **BIKO** es una interfase en computadora de aplicación particular que tiene precisamente el propósito de aumentar la eficiencia y versatilidad del sistema de Control con el cual opera conjuntamente. **BIKO** es un sistema que ataca el problema de Automatización con tecnología actualizada y con enfoque moderno.

3.2 Definición del Problema.

Como ya fué citado en la Introducción, no podemos explicar detalles fabriles del proceso por ser estos de propiedad privada. Nos limitaremos a decir que el proceso en cuestión es un proceso de Tratamiento Térmico de piezas automotrices.

Para definir el problema de Automatización del Sistema de Tratamiento Térmico, a continuación citaremos textualmente la descripción técnica del sistema tal como fué definido originalmente:

Para operar la planta proyectada, se requiere únicamente de dos hombres que coloquen el material a ser tratado en las charolas transportadoras. La planta consiste de un horno de temple, las líneas de templado con una cesta y un horno de revenido con un elevador. Todo el proceso trabajará en forma completamente automática.

El horno de temple está construido como un horno de empuje hidráulico y tiene una longitud de trabajo de aproximadamente 8.5 metros. Las charolas transportadoras tienen una superficie de 1.8 por 0.3 metros. El sistema hidráulico de empuje que funciona por presión de aceite, situado enfrente del horno de temple, empuja las charolas transportadoras hacia adentro del horno de temple desplazando a las que ya se encuentran dentro. El empuje tiene lugar en intervalos periódicos de tiempo que pueden ser regulados mediante un reloj. A la salida del horno las charolas son extraídas del interior por un equipo hidráulico de salida similar al de entrada, donde el material incandescente pasa de las charolas a la cesta. En caso de sobrecarga, 1 o 2 charolas de transporte son extraídas respectivamente y descargadas por el aparato correspondiente (descargador). El descargador empuja las piezas en la cesta de la máquina de templado. De acuerdo al número de charolas que han sido extraídas, nuevas charolas serán introducidas al horno por medio del empujador de la entrada.

La cesta de templado transporta el material incandescente rápidamente hacia el líquido de enfriamiento respectivo. Para ello se han destinado una tina de aceite y una tina de agua.

Las charolas vacías automáticamente son regresadas a la entrada del horno por el equipo de transporte y ahí pueden cargarse nuevamente. El arreglo de interruptores del equipo empujador y extractor está construido de tal forma que, después de cada secuencia extracción a la salida - empuje a la entrada de las charolas, las puertas son cerradas nuevamente. Gracias a esto, la pérdida de calor por reflexión en ambas puertas se reduce al mínimo.

La duración del tiempo de templado del material puede regularse por medio de un reloj según las circunstancias de operación.

El calentamiento del horno de temple es llevado a cabo por 21 quemadores de alto rendimiento que están útilmente disgregados en las partes baja y alta del horno. La longitud efectiva del horno está dividida en dos zonas de calentamiento que están automáticamente reguladas por dispositivos de medición y control de temperatura. Para el templado del material se han proporcionado líneas estacionarias de aceite y agua, una seguida de la otra. Ambas líneas están montadas sobre una base firme.

La cesta está construida sobre un riel fijo que es movido electromecánicamente mediante un motor. Las charolas de transporte pueden ser regresadas durante el proceso de temple constantemente, aprovechando su calor para evitar pérdidas de temperatura. Los movimientos de elevación y rotación son llevados a cabo por

cilindros diferenciales trabajando por presión hidráulica de aceite. Los componentes hidráulicos están situados encima de la cesta. Una vez terminado el proceso de templado la charola de transporte es movida rápidamente a la mesa de distribución que está colocada frente al horno de revenido. Girando la cesta hacia un lado el material resbala sobre la mesa. Las partes pesadas pueden ser colocadas directamente desde la cesta sobre las charolas transportadoras del horno de revenido.

Las charolas del horno de revenido son cargadas manualmente desde la mesa de distribución. El horno de revenido dispone de una longitud de trabajo de aproximadamente 14.590 m.

El calentamiento del horno de revenido, que está subdividido en tres grupos de calentamiento, es realizado por un total de 22 quemadores de gas automáticos de alto rendimiento. Un recirculador de aire colocado en el techo del horno garantiza calentamiento constante y uniforme del material.

Los equipos de medición y control de ambos hornos están dispuestos dentro de un gabinete. Los dispositivos eléctricos de control están situados en dos tableros de interruptores con los correspondientes contactores. Los tableros están equipados con los diagramas de conexiones.

Para aclarar un poco toda la explicación previa anexamos las figuras 3-1, 3-2 y 3-3 que son dibujos esquemáticos del proceso con cada una de las partes que lo forman.

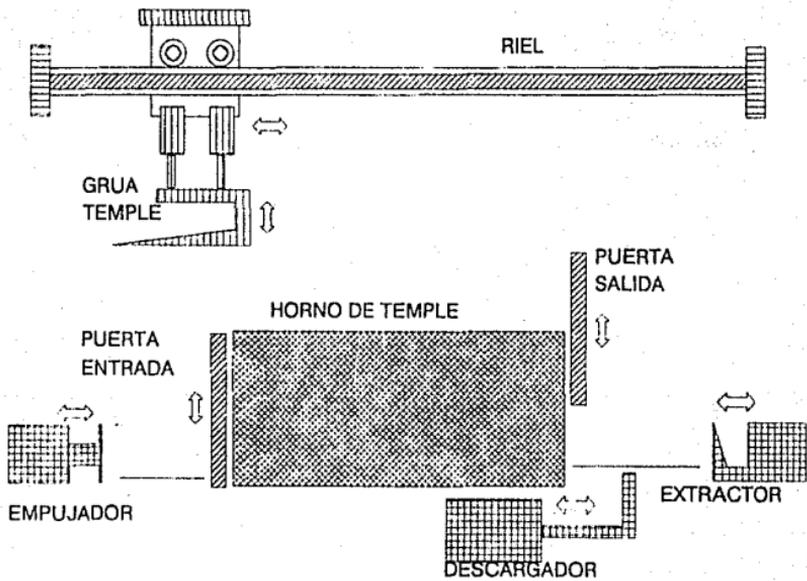


FIGURA 3-1

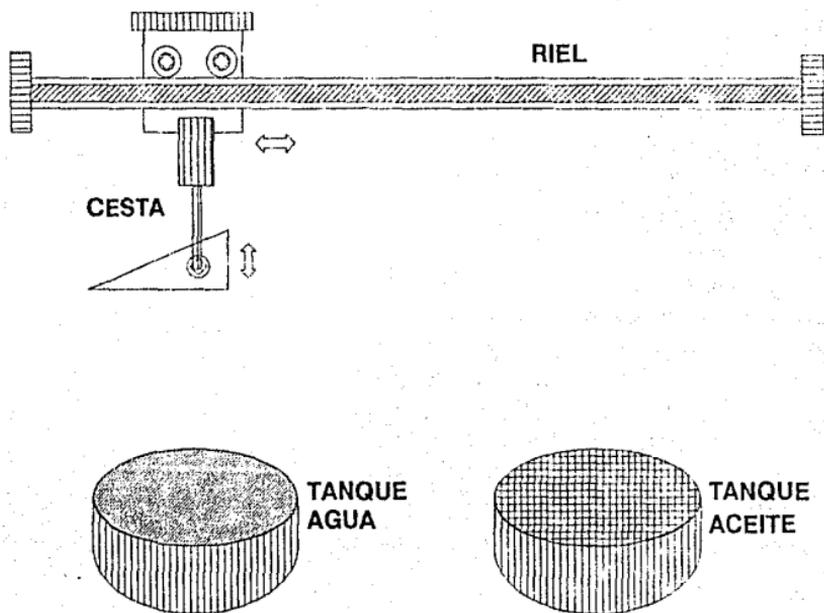


FIGURA 3-2

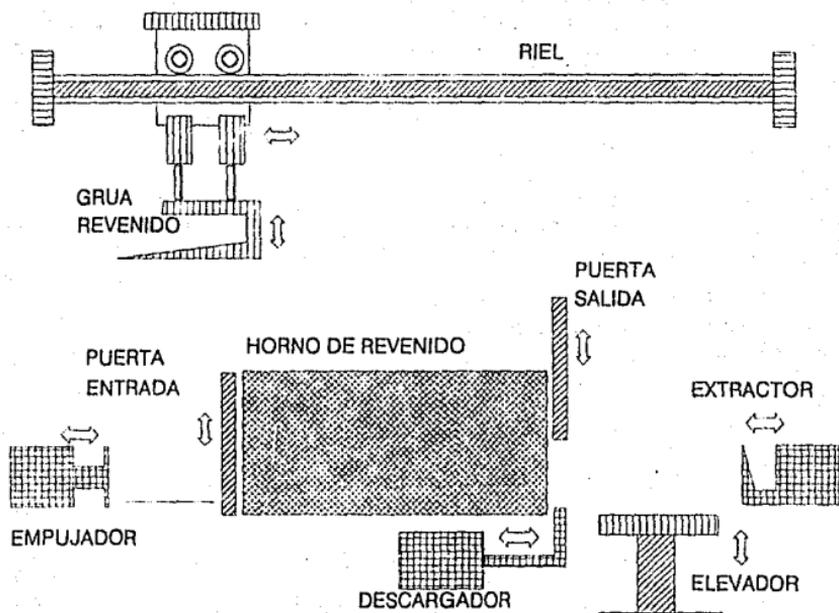


FIGURA 3-3

El problema conexas al de controlar la secuencia de movimientos de los elementos de transporte de piezas a través de los hornos es, por supuesto, el Control de temperatura en el interior de éstos. El Control de temperatura es esencial al proceso por ser inherente al Tratamiento Térmico de los materiales con el cual se modifican algunas de las propiedades físicas de estos metales y aleaciones de acuerdo al futuro uso específico de las piezas. Este problema involucra dos aspectos de diseño que implicaron un trabajo interdisciplinario con los ingenieros de ENTERPRISE S.A.:

1) El equipo de ingenieros de combustión de ENTERPRISE se encargó, a grandes rasgos de:

- Definir, junto con el usuario, las especificaciones de operación de los hornos: número de zonas de Control de temperatura, rangos de temperatura y zonas de adecuación de temperatura.
- Diseñar el sistema de combustión de acuerdo a las especificaciones anteriores:
 - Tipo, distribución y número de quemadores.
 - Distribución de tuberías de gas y aire.
 - Selección y ubicación de los dispositivos de control y seguridad de los sistemas de combustión.

El diagrama de la figura 3-4 muestra la distribución física de los quemadores del sistema de combustión del Horno de Temple. La figura 3.5 hace su parte en el horno de revenido.

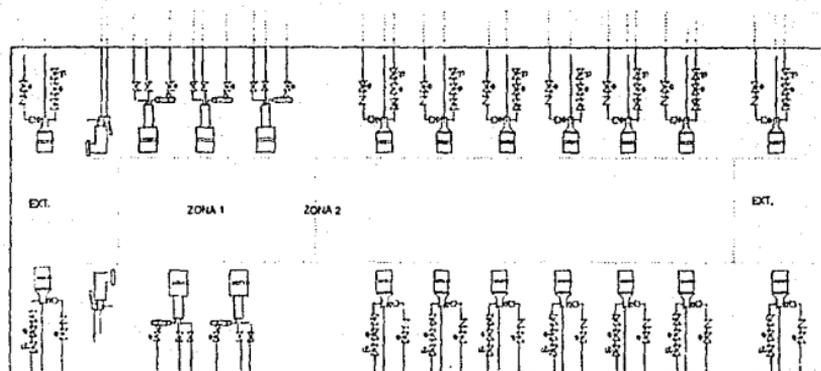


FIGURA 3-4

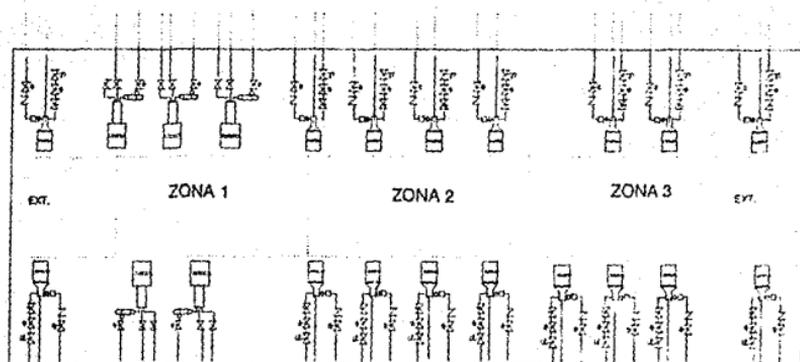


FIGURA 3-5

2) Nosotros, el equipo de desarrollo de BIKO, nos encargamos de implantar la estrategia de Control de este sistema.

A diferencia del Control de movimientos el cual es discreto, el Control de la temperatura, como variable analógica, implica la implantación de un lazo cerrado de Control para cumplir con las especificaciones del Tratamiento Térmico subyacente.

Definiremos este lazo a través de los diferentes elementos utilizados para formarlo. Cabe aclarar que existe un lazo de Control independiente para cada zona.

- Entrada. La temperatura de cada zona es monitoreada a través de un termopar y transmitida al elemento de Control a través de un convertidor analógico/digital.
- Elemento de Control: Algoritmo de Control PID (Control Proporcional Integral y Derivativo) con capacidad de operación Manual.
- Salida. Válvula proporcional que controla el flujo de aire hacia los quemadores de cada zona, permitiendo así, establecer la temperatura en ésta.

El algoritmo PID es la parte importante en el esquema de control. Para un proceso como éste, el tiempo de actualización no es demasiado crítico y esto constituye un criterio clave para elegir el dispositivo que se utilizará para llevarlo a cabo. El análisis de alternativas para dar solución a este problema será expuesto más adelante.

La descripción anterior nos ayuda a entender cuál es la secuencia de operación del proceso de Tratamiento Térmico y con qué elementos se cuenta para llegar a ella. Aún cuando ésta es breve, explica básicamente cuál es el sistema por automatizar. Como enfatizábamos al principio de este capítulo, la Automatización de dicho proceso es la causa originaria que dió motivo a nuestro proyecto de tesis y en torno a todo ello gira la aplicación del sistema BIKO.

3.3 Alternativas y Selección de la Solución.

En este capítulo hemos tocado el tema de Automatización de Procesos Industriales, seguido de la definición del problema a resolver como un problema de Automatización Industrial, citando con qué elementos se cuenta y qué debe realizar el proceso. Ahora debemos mostrar alternativas de solución para atacar el problema.

La pregunta que debemos resolver es: ¿Cómo y con qué lograr que el sistema opere en la forma especificada?

Podemos subdividir el problema en dos partes:

- 1) Control de Movimientos de los elementos de transporte de cada horno y la interfase con el operador que le permita supervisar este aspecto.
- 2) Control de la Temperatura de cada zona de los hornos y la interfase con el operador que le permita supervisar este aspecto.

Debido a que los subsistemas de Combustión y Control de movimientos de cada horno son independientes entre sí, podemos atacar cada subproblema seleccionando la mejor alternativa y después conjuntarlas para obtener la solución general. Cabe aclarar que, debido a la independencia entre ambos hornos y sus subsistemas, la interfase con el operador estará compuesta por los elementos de Supervisión y Control que proporcione cada solución. Además deberá cumplir con los requerimientos extra que llegue a solicitar el usuario.

3.3.1 Subproblema I.

El Control de Movimientos de cada horno, tiene como objetivo el transporte de las charolas a través del horno y es responsable de lograr que cada pieza sea sometida al perfil de temperatura especificado para ésta. Este Control de movimientos debe tener dos modalidades: Automática y Manual.

En la modalidad Automática, los movimientos deben seguir una secuencia predefinida que puede interrumpirse a voluntad para cambiar a Manual o continuar con la operación.

En la modalidad Manual, el operador va indicando los movimientos que desea, siempre y cuando sean permisibles.

Los requerimientos de interfase para este subproblema pueden dividirse a su vez en dos partes:

1) Requerimientos de comando.

- Definir o modificar tiempo entre ciclos.
- Definir o modificar tiempos inherentes a ciertos ciclos.
- Seleccionar el ciclo Automático.
- Cambiar modalidad Manual/Automático.
- Realizar movimientos aislados en modo Manual.

2) Requerimientos de Supervisión.

- Monitorear el estado (posición) de cada uno de los elementos.
- Indicar la modalidad de funcionamiento.

Las alternativas consideradas para solucionar este subproblema fueron:

1) Tablero de Control a base de relevadores y temporizadores.

Esta solución implica una interfase de operador en base a una estación de botones y tablero de lámparas con mímicos en acrílico.

2) Controlador Programable.

La solución implica utilizar una terminal inteligente de Supervisión (computadora).

La elección entre ambas alternativas favoreció a la que proponía al Controlador Programable, las causas se discutirán más adelante.

3.3.2 Subproblema II.

El Control de Temperatura de cada zona tiene como objetivo mantener la temperatura de cada zona a un nivel constante que fija el operador. Cada zona es independiente de las otras. Este Control tiene dos modalidades: Manual y Automático. El Control de temperatura implica el Control de encendido y apagado de los quemadores, así como su suministro de gas y aire.

En la modalidad Automática, la apertura de las válvulas proporcionales de aire debe ser controlada por un algoritmo PID (Control Proporcional Integral y Derivativo).

En la modalidad Manual, la apertura de las válvulas proporcionales de aire, es controlada y fijada por el operador.

Al igual que en el subproblema I, los requerimientos de interfase para este subproblema se dividen en:

1) Requerimientos de comando.

- Cambio de temperatura de referencia en cualquier momento para cada zona.
- Conmutación Manual/Automático para cada zona.
- Control de apertura de válvulas en modalidad Manual.
- Ajuste de límites de alarma para cada zona en cualquier momento.
- Reconocimiento de alarmas.

2) Requerimientos de Supervisión.

- Indicación de temperatura y aperturas de válvulas para cada zona.
- Indicación de límites de alarma y status de éstas.

- Registro de temperaturas y hora de cada lectura, así como representación gráfica de éstas (por zona), con cálculos estadísticos definidos por el usuario.
- Registro histórico de alarmas.

Las alternativas de solución consideradas para este subproblema fueron:

1) Controlador Programable. Implica una terminal inteligente de Supervisión como en el subproblema anterior.

2) Controladores y registradores SLC. Estos aparatos monitorean y controlan todos los parámetros correspondientes a una variable de proceso en forma Automática o Manual. Además, llevan un registro en papel de los parámetros a controlar. Esta solución implica el uso de un aparato especial de monitoreo y registro de alarmas. Cabe aclarar que los PLC's son tan eficientes como los SLC's para el Control de variables analógicas cuya rapidez de variación es media o lenta (variables que necesitan tiempos de actualización mayor a 1 seg cuando se utiliza implantación en hardware de algoritmos PID). Para esta opción, además se requiere el uso de relevadores y una estación de botones para controlar el encendido y apagado de los quemadores.

Las razones para escoger un Controlador Programable como solución al subproblema II son:

1) Económicas.

Un Controlador Programable es capaz de controlar lazos analógicos mediante un algoritmo PID. Por lo tanto, el utilizar un Controlador Programable para el Control de Movimientos (subproblema I) y para el Control de Temperatura, permite también utilizar la misma terminal para interfase con ambos aspectos del proceso y ahorrar el costo extra de los controladores SLC, registradores y monitoreo de alarmas.

2) Facilidad de Instalación.

La instalación de equipos de Control basados en PLC's es versátil, debido a que estos son sistemas modulares.

3) Centralización y Versatilidad de Supervisión.

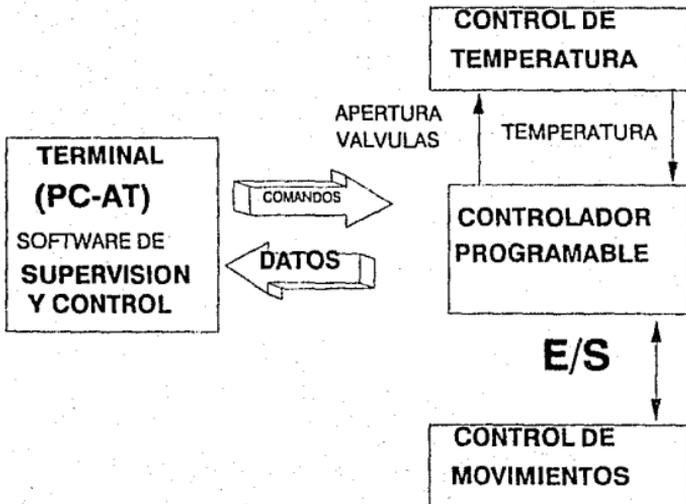
Al centralizar el Control, podemos centralizar la Supervisión, es decir, no tener disgregados diferentes tableros e indicadores de monitoreo. Tenemos mayor versatilidad en la Supervisión, al aprovechar la información digital contenida en el PLC.

Cabe aclarar, que también se consideró una alternativa más económica a la anterior: utilizar una computadora personal con las tarjetas de E/S adecuadas para el Control de Movimientos y Temperatura de los hornos. Las razones por las que esta alternativa no fué seleccionada son las siguientes:

- Carencia de un sistema operativo con multiprocesamiento. Este es un aspecto fundamental para que el programa de Supervisión comparta los recursos de la computadora con el programa de Control. Aún este esquema no hubiera permitido la operación en tiempo real por muy alta que fuese la velocidad de la máquina utilizada.
- Era obligado programar, desde el nivel más bajo, los algoritmos PID y de Control cuando éstos ya están implantados en un PLC.

En cambio, la arquitectura de la solución basada en un PLC permite un esquema de multiprocesamiento, ya que el programa de Supervisión (BIKO), se ejecuta concurrentemente con el programa de Control (el programa en escalera en el PLC). Esto, evidentemente permite una alta eficiencia y un funcionamiento en tiempo real.

Así pues, tenemos como solución propuesta la mostrada en la figura 3-5.



FIGUR A 3-5

Presentación del Proyecto

En los capítulos siguientes, ahondaremos más en estas ideas y la descripción detallada de la arquitectura seleccionada para solucionar el problema planteado.

4. Introducción a los Controladores Lógicos Programables

4 Introducción a los Controladores Lógicos Programables.

4.1 Hardware Básico de un PLC.

Sin tomar en cuenta el tamaño, la complejidad o el costo, todos los Controladores Programables contienen un conjunto básico de partes. Estas partes son elementos de *hardware* cuyo funcionamiento requiere *software* dedicado.

En su arquitectura básica, todos los PCL's deberán al menos contener lo siguiente:

- Una Interfase de Entrada
- Una Unidad de Procesamiento (CPU)
- Una Unidad de Memoria
- Un Lenguaje de Programación
- Una Herramienta de Programación
- Una Interfase de Salida

En algunos casos también pueden tener una Interfase de Comunicaciones. Ver figura 4-1.

Dentro de las características más importantes de los PLC's y los Módulos Periféricos a éstos tenemos:

- La construcción de un PLC está basada en componentes electrónicos de alta integración lo cual permite un diseño modular. Esta modularidad ofrece la ventaja de configuración de bloques según las necesidades del sistema. Esto significa que si el sistema de Control tiene requerimientos de expansión en entradas y

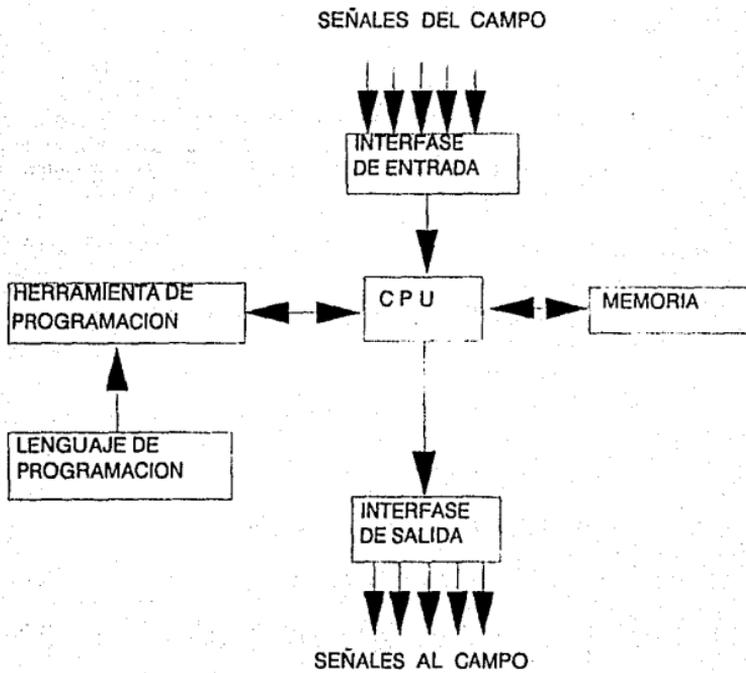


FIGURA 4-1

salidas, bastará con agregar los correspondientes módulos de E/S y programar al PLC para solucionar el problema.

- Los PLC's son sistemas orientados específicamente al Control industrial. Por ello, un PLC está construido con características tales que le permitan soportar rudos ambientes de corrosión, polvo, vibraciones, humedad y altas temperaturas. Un PLC debe satisfacer las normas industriales.
- La capacidad de programación de un PLC hace que estos equipos sean flexibles a los cambios de la lógica de Control del sistema. En forma similar a una computadora, el programa de un PLC puede editarse a través de un software especial en una terminal o computadora personal para reprogramar al PLC.
- Un PLC y un módulo de E/S ocupan tan poco espacio físico como dos carpetas en un librero. Dependiendo de la cantidad de módulos de E/S varían las dimensiones de un sistema de Control con PLC's.
- Los módulos de E/S de un PLC son interfaces directas al campo que no requieren de aislamiento eléctrico ni adecuación de los niveles de la señal en un rango muy amplio.
- La modularidad de sus componentes permite que las partes dañadas puedan intercambiarse sin afectar al resto del sistema.

La interfase de entrada proporciona una conexión con el proceso a controlar. La principal función de esta interfase es recibir y convertir las señales de campo de forma tal que puedan ser usadas por el CPU. Esto implica convertir cualquier tipo de señal a niveles de voltaje que puedan ser interpretados por el CPU. La interfase de entrada es modular por naturaleza. Generalmente puede expandirse agregando módulos que permitan mayor número de entradas, cuando la tarea por controlar o el usuario lo requieran. Los límites de esta expansión generalmente los imponen la naturaleza del CPU y el tamaño de la memoria.

La Unidad de Procesamiento (CPU) y la Unidad de Memoria son la parte inteligente del Controlador Programable. La información fundamental de operación es almacenada en la memoria como un conjunto de bits agrupados en palabras. Cada palabra en memoria puede representar una instrucción, parte de una instrucción o un conjunto de datos.

La operación del CPU y la memoria del controlador puede describirse en el siguiente ciclo:

1. Obtener información del proceso a controlar a través de la interfase de entrada.
2. Comparar la información de entrada con la información de Control proporcionada y almacenada en el programa.
3. Determinar si se requiere efectuar alguna acción de Control.
4. De ser así, ejecutar la acción de Control transmitiendo información a la interfase de salida.
5. Examinar de nuevo la información de entrada.

La interfase de salida tiene una función opuesta a la de entrada. Recibe información del CPU y la convierte a señales apropiadas para producir acciones de Control sobre dispositivos externos como válvulas, bombas, calentadores, etc. Así como la interfase de entrada, la interfase de salida también es modular, de tal forma que puedan incorporarse funciones adicionales de salida.

A la interfase de entrada y a la de salida generalmente se les nombra Unidad de E/S. Algunas funciones comunes de E/S incluyen lo siguiente:

- Detección de energía (AC/DC) de los circuitos conectados a la interfase de entrada.
- Encender o apagar señales de energía (AC/DC) de los circuitos conectados a la interfase de salida.
- Aislar los circuitos del Controlador Programable de señales de campo para evitar daños al *hardware* del PLC.
- Registrar señales de Control a niveles típicos de campo y convertirlas en señales adecuadas para el CPU.
- Registrar Señales de Instrumentación, tales como voltajes de Termopares, convirtiéndolas en señales manejables por el CPU.

- Generar señales a niveles típicos de campo para controlar dispositivos externos.
- Recibir y generar señales de comunicación en código ASCII.

El programa escrito por el usuario y almacenado en la memoria del Controlador es una representación de las acciones necesarias para producir las señales de Control de salida correctas para alguna condición dada del proceso. El programa incluye secciones de adquisición de datos, secciones de toma de decisiones y secciones de transferencia de datos de salida.

Los lenguajes de programación de PLC's tienen muchas formas. Las primeras versiones estaban restringidas para apegarse a la convención de la lógica de relevador que dió origen al PLC. Esta consiste de diagramas de escalera que representan contactos y bobinas. Este tipo de programas consisten de una representación del esquema de Control mediante lógica de relevador. El PLC opera para generar las salidas que normalmente se producirían por el correspondiente circuito de Control. Este lenguaje escalera-relevador es aún muy popular. En muchos casos, sus rangos han sido extendidos y la programación ha sido enormemente simplificada introduciendo funciones de bloques, tales como programar temporizadores de cierta complejidad, lo cual reduce grandemente la carga de entrada al programa. Algunos lenguajes alternativos usan una representación booleana de estos esquemas de Control como base para la representación en computadora. Desarrollos subsecuentes han llevado a la introducción de una variedad de lenguajes de alto nivel. Algunos de estos están especializados según la aplicación mientras otros son más generales.

4.2 Programación de un PLC.

Hasta el momento no se ha detallado acerca de la programación de un PLC, es por ello que incluimos este apartado para proporcionar una idea global de cómo se programa lógica de Control en equipos de esta naturaleza. No es finalidad nuestra hacer llegar al lector una guía de

aprendizaje de programación de PLC's ni mucho menos. Quien esté interesado encontrará información detallada en el apéndice A y en los manuales de referencia de los fabricantes de Controladores Programables (ver Referencias).

La forma de programar un PLC es mediante un lenguaje especial denominado comunmente **Lenguaje Escalera**.

Dicho lenguaje es un conjunto de intrucciones que, agrupadas sintácticamente, pueden ser interpretadas por el PLC en el momento de ejecución.

El Lenguaje Escalera tiene ciertas características muy propias. En primera instancia, es un lenguaje orientado a interpretarse de forma análoga a un diagrama eléctrico. Es por ello que tiene una apariencia gráfica para poder representar, en base a dibujos, los símbolos empleados en la nomenclatura eléctrica: botones, contactos, lámparas, etc.

Para aclarar lo anterior observemos la siguiente figura que representa una instrucción típica en Lenguaje Escalera:



FIGURA 4-2

su diagrama eléctrico equivalente sería:

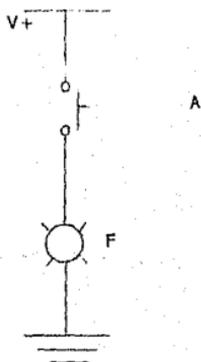


FIGURA 4-3

La interpretación a ambas figuras debe coincidir en lo siguiente:

Si el contacto A está cerrado, energiza la salida F

La figura 4-2 es una instrucción para el procesador del PLC. Esta instrucción puede constituir por sí misma un programa en Escalera, aunque éste sería un caso muy sencillo de Control. Para seguir ejemplificando el concepto de programación en escalera supongamos el siguiente caso en nomenclatura eléctrica:

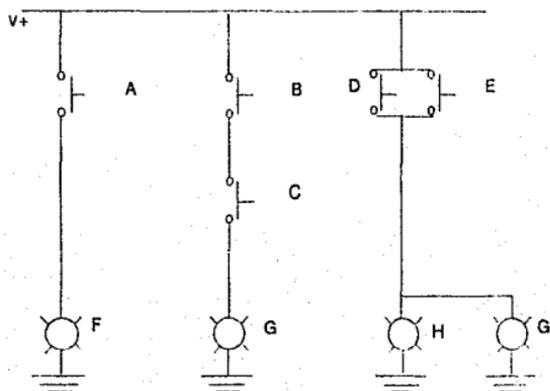


FIGURA 4-4

que traducido a escalera quedaría como sigue:

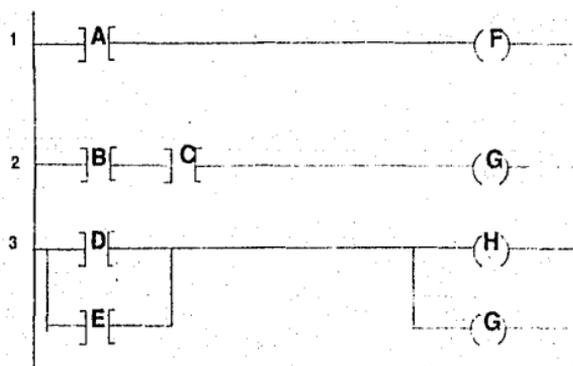


FIGURA 4-5

El procesador interpreta este programa de la siguiente forma:

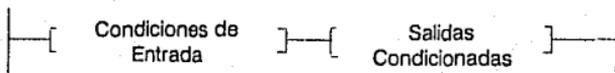
Si el contacto A está cerrado, energiza la salida F

Si el contacto B y el contacto C están cerrados, energiza la salida H

Si el contacto D o el contacto E están cerrados, energiza las salidas H y G

El procesador ejecuta el programa instrucción por instrucción, en el orden secuencial en el que aparecen, esto es, efectúa un barrido de arriba a abajo a todo lo largo del programa. Cada instrucción constituye un escalón dentro del programa. Si observamos la figura 4-5 con un poco de imaginación, encontraremos semejanza con una escalera de tres escalones. Ahora podemos entender el porqué se le denomina a este lenguaje con tan singular nombre.

Podemos separar un escalón en dos partes fundamentales:



En la parte izquierda del escalón tenemos un conjunto de condiciones de entrada, en la parte derecha residen las instrucciones de salida. Las condiciones de entrada se evalúan en el momento de ejecución y si el resultado es verdadero, se efectúan las instrucciones de salida:

SI <CONDICIONES DE ENTRADA> ENERGIZA < SALIDAS CONDICIONADAS >

Esto es similar a una instrucción :

IF (expresión condicional) THEN (bloque de Instrucciones)

de algún Lenguaje de programación, por lo que, si el resultado del escalón es falso, es decir, la evaluación de las condiciones de entrada, no se ejecutarán las instrucciones de salida.

Dentro de las condiciones de entrada no sólo existen intrucciones lógicas como las mostradas en los ejemplos. Dichas intrucciones se conocen como intrucciones de relevador. También existen intrucciones aritméticas, de comparación, para manejo de *bits* y archivos, así como contadores, temporizadores y otras intrucciones que pueden combinarse como condiciones de entrada para un escalón.

Un programa en escalera en su nivel más elemental, como puede apreciarse, tiene la característica de no tener bifurcaciones por decisión, saltos a etiquetas ni mucho menos estructuras de Control inmersas en el lenguaje. La ejecución del programa es puramente secuencial, instrucción por instrucción, es decir, escalón por escalón. ¿Qué sucede cuando termina de ejecutar el último escalón? El programa no finaliza, sucede que se ejecuta nuevamente el primer escalón y demás consecutivos. En otras palabras, existe un ciclo inherente en la ejecución de un programa en escalera, un ciclo infinito que permanece iterando durante todo el momento de ejecución del programa. A este ciclo se le denomina **SCAN**. El término **SCAN**, entonces, debe interpretarse como un barrido secuencial y cíclico de arriba a abajo a lo largo del programa en escalera. Esto se ilustra en la siguiente figura:

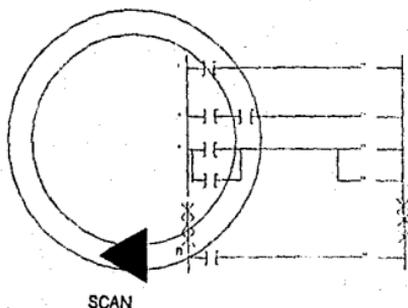


FIGURA 4-6

Para ejercer Control sobre la secuencia de ejecución de un programa es necesario emplear instrucciones especiales para alterar la secuencia normal del SCAN pues éste se realiza en todo momento mientras esté en ejecución el programa. Un programador generalmente está acostumbrado a tener Control absoluto sobre la secuencia de ejecución en cualquier lenguaje, pero en este caso no resulta así. En un programa en escalera debe considerarse que cada escalón se ejecutará inevitablemente a menos que se llame a una instrucción especial. Esta situación, como mencionamos, inicialmente puede causar desconcierto en programadores acostumbrados a lenguajes de alto nivel, por lo que debe tenerse en mente un enfoque especial al programar en escalera.

El lenguaje en escalera es sin duda un caso especial dentro de los lenguajes de programación. El lector podría preguntarse por qué este lenguaje busca apegarse a los diagramas y nomenclatura eléctricos para representar lógica. Si bien el Lenguaje Escalera está muy lejos de ser un lenguaje de alto nivel, aun cuando podemos catalogarlo dentro de los lenguajes de tipo imperativo, su propósito es acercarse en la medida de lo posible a representar lógica alamburada que sea de fácil y rápida identificación al personal con conocimiento técnico que lleva a cabo la programación. Es decir, un ingeniero o técnico que diseña una lógica de Control como en el caso de la figura 4-4, requiere de un lenguaje que le permita traducir ese esquema a un conjunto dado de instrucciones, evitando en lo más mínimo la pérdida de información visualizada en el diagrama eléctrico original. El lenguaje Escalera está diseñado con este fin.

Resulta evidente, que el lenguaje en escalera está orientado a la programación de lógica de Control como si ésta estuviese alamburada. Para reafirmar esta idea, pensemos en un sistema digital construido en base a compuertas lógicas, como el mostrado en la figura 4-7, que desempeñe la misma función que el programa de la figura 4-5:

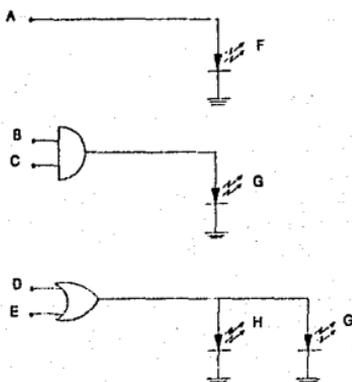


FIGURA 4-7

¿Cuál es la diferencia entre este sistema y el programa de la figura 4-4? Aparentemente ninguna, pero ¿qué sucedería si desearamos modificar nuestra lógica de Control? Sucedería simplemente que tendríamos que deshechar este circuito y diseñar y armar uno nuevo, en cambio, el PLC sólo requiere ser programado con las instrucciones apropiadas. Visto como una caja negra, el PLC es algo así como un sistema digital que puede ser realambrado cada vez que se requiera.

Hasta aquí hemos tratado la idea básica de cómo se efectúa la programación en Escalera que es el Lenguaje de programación de un PLC. En el apéndice A se amplía con más detalle este tema.

5. Solución Propuesta: Background

5 Solución Propuesta: Background.

A la organización y distribución de los diferentes elementos de *hardware* de la estrategia de Control propuesta le denominaremos Arquitectura. Al nivel compuesto por la Arquitectura y el programa de Control de ésta para actuar sobre el proceso le nombraremos **Background**. Por otro lado, al nivel compuesto por la terminal de Supervisión y Control y el *software* que permite esta operación le llamaremos **Foreground**. Si revisamos la figura 1-2 tenemos que el bloque 1 representa el *Background*, en tanto, el bloque 3 representa al *Foreground*. El bloque 2 es una vía de comunicación entre el primero y el último.

De esta forma, el *Background* es el responsable de controlar el proceso y su falla implica una falla en este último. Una falla en el *Foreground* se traduce en una falla para supervisar y enviar comandos al proceso. El *Foreground* sólo constituye la interfase para el operador. En este sentido, no suele ser crítico para el Control del proceso. El *Background* y el *Foreground* son independientes.

Una Arquitectura basada en Controladores Programables puede estar compuesta por uno o más PLC's, uno o más *racks* y uno o varios Módulos de E/S. La descripción del tipo de PLC's, el número de *racks* y el número, tipo y distribución de módulos de E/S se definen en base al número y tipo de entradas y salidas que involucra el proceso.

Esta información se obtiene ya sea de los planos eléctricos de Control o bien al momento de diseñar el proceso. En nuestro caso, la información se obtuvo de los planos eléctricos del Sistema de Control anterior.

Otro aspecto importante para definir la Arquitectura es la distribución de las cargas de salida en los Módulos de Salida. Esta distribución debe ser tal que en ningún caso se deben demandar más de 8 Amps por módulo. A su vez, es importante definir el modo de direccionamiento utilizado, esto es, el método que utilizará el procesador del PLC para efectuar la correspondencia entre direcciones lógicas y ubicaciones físicas.

5.1 Diseño de la Arquitectura

En esta sección describiremos a detalle los elementos de *hardware* de la solución propuesta.

5.1.1 Número y Tipo de Entradas y Salidas.

La fuente de esta información son los planos eléctricos del sistema de Control anterior proporcionados por FORJAMEX S.A.

ENTRADAS DEL HORNO DE REVENIDO

No	TIPO	USO
1	Analógica 4-20 mA	Cálculo de gasto
37	Discretas 120 VCA	<i>Push-Button</i>
2	Discretas 120 VCA	Selectores
12	Discretas 120 VCA	Dispositivos de levas
15	Discretas 120 VCA	Switch de límite
5	Discretas 120 VCA	Interruptores de presión y válvulas de suministro
3	Analógicas Termopar mV	Temperatura de las zonas
22	Discretas 120 VCA	Detector de falla de flama

Subtotales: 93 discretas (120 VCA)

1 analógica (4-20 mA)

3 analógicas (Termopar mV)

ENTRADAS DEL HORNO DE TEMPLE

No	TIPO	USO
1	Analógica 4-20 mA	Cálculo de gasto
30	Discretas 120 VCA	<i>Push-Button</i>
3	Discretas 120 VCA	Selectores
12	Discretas 120 VCA	Dispositivos de levas
14	Discretas 120 VCA	Switch de límite
5	Discretas 120 VCA	Interruptores de presión y válvulas de suministro
2	Analógicas Termopar mV	Temperatura de las zonas
21	Discretas 120 VCA	Detector de falla de flama

Subtotales: 85 discretas (120 VCA)

1 analógica (4-20 mA)

2 analógicas (Termopar mV)

ENTRADAS DE LA CESTA

No	TIPO	USO
15	Discretas 120 VCA	<i>Push-Button</i>
18	Discretas 120 VCA	Switch de límite

Subtotales: 33 discretas (120 VCA)

ENTRADAS DEL TANQUE DE AGUA

No	TIPO	USO
6	Discretas 120 VCA	Selectores
2	Discretas 120 VCA	Termostatos
1	Discretas 120 VCA	Detector de nivel

Subtotales: 9 discretas (120 VCA)

ENTRADAS DEL TANQUE DE ACEITE

No	TIPO	USO
6	Discretas 120 VCA	Selectores
2	Discretas 120 VCA	Termostatos
1	Discretas 120 VCA	Detector de nivel

Subtotales: 9 discretas (120 VCA)

TOTAL DE ENTRADAS

No	TIPO	USO
229	Discretas 120 VCA	15 módulos 1771- IAD AB (16 entradas 120 VCA c/u)
2	Analógicas 4-20 mA	1 módulo 1771- IFE AB (8 - 16 canales)
5	Termopar mV	1 módulo 1771- IXE AB (8 canales)

NOTA: En la implantación todas las entradas discretas de 120 VCA correspondientes a los *push-buttons* y selectores quedaron vacantes debido a que el usuario decidió operar los movimientos desde la terminal de Supervisión. Por lo tanto todas las entradas correspondientes se encuentran disponibles actualmente.

SALIDAS DEL HORNO DE REVENIDO

No	TIPO	USO
21	Discretas 120 VCA	Arrancadores de motores
20	Discretas 120 VCA	Válvulas solenoides
49	Discretas 120 VCA	Lámparas
22	Discretas 120 VCA	Controles de falla de flama
3	Analógicas 4-20 mA	Válvulas proporcionales

Subtotales: 112 discretas (120 VCA)

3 analógicas (4-20 mA)

SALIDAS DEL HORNO DE TEMPLE

No	TIPO	USO
11	Discretas 120 VCA	Arrancadores de motores
18	Discretas 120 VCA	Válvulas solenoides
45	Discretas 120 VCA	Lámparas
21	Discretas 120 VCA	Controles de falla de flama
2	Analógicas 4-20 mA	Válvulas proporcionales

Subtotales: 95 discretas (120 VCA)

2 analógicas (4-20 mA)

SALIDAS DE LA CESTA

No	TIPO	USO
7	Discretas 120 VCA	Arrancadores de motores
4	Discretas 120 VCA	Válvulas solenoides
29	Discretas 120 VCA	Lámparas

Subtotales: 40 discretas (120 VCA)

SALIDAS DEL TANQUE DE ACEITE

No	TIPO	USO
6	Discretas 120 VCA	Arrancadores de motores
1	Discretas 120 VCA	Válvulas solenoides
5	Discretas 120 VCA	Lámparas

Subtotales: 12 discretas (120 VCA)

SALIDAS DEL TANQUE DE AGUA

No	TIPO	USO
6	Discretas 120 VCA	Arrancadores de motores
1	Discretas 120 VCA	Válvulas solenoides
5	Discretas 120 VCA	Lámparas

Subtotales: 12 discretas (120 VCA)

TOTAL DE SALIDAS

No	TIPO	USO
271	Discretas 120 VCA	18 módulos 1771- OAD AB (16 salidas 120 VCA c/u)
5	Analógicas 4-20 mA	2 módulos 1771- OFE2 AB (4 canales c/u)

NOTA: En la implantación todas las salidas discretas de 120 VCA correspondientes a las lámparas quedaron vacantes en los Módulos de E/S debido a que el usuario decidió monitorear el estado y ubicación de los elementos del proceso desde la Terminal de Supervisión. Por lo tanto todas las salidas correspondientes actualmente se encuentran disponibles para expansión.

Para especificar el *hardware* necesario, según las características demandadas, debemos consultar los catálogos correspondientes en los Apéndices A, B y C.

De los totales anteriores, considerando 16 Entradas o Salidas por Módulo, se observa que se requieren 37 módulos de E/S. Si a estos se agregan un módulo de interfase que permita la comunicación entre el PLC y la terminal supervisora se tiene:

No	TIPO DE MODULOS	No RACKS
37	Entrada/Salida	4 (con capacidad para 16 módulos, considerando espacio para expansión)
1	Interfase	

El hecho de que exista esta cantidad de entradas y salidas más la necesidad de controlar 5 variables analógicas y comunicarse con 4 racks de E/S, así como una terminal de Supervisión vía un módulo de interfase, nos lleva a decidir que el PLC necesario para esta aplicación es el PLC 5-15 de Allen-Bradley (más los Módulos correspondientes). Este PLC puede manejar hasta cuatro racks de E/S, 512 entradas y salidas y 32 lazos de Control Analógico.

La selección de un PLC Allen-Bradley recae sobre 3 aspectos fundamentales:

- La marca Allen-Bradley es de reconocido prestigio y tiene representación en México.
- El equipo de desarrollo, es decir, los autores de esta tesis, poseen conocimientos teóricos de ésta línea de Controladores Programables.
- Fué la alternativa propuesta por la Empresa ENTERPRISE S.A.

Resumiendo, el número y tipo de módulos necesarios para esta aplicación es:

No	MODELO	USO
1	PLC 5-15 1785-LT	Controlador Programable
4	1771-ASB	Adaptador de Rack
4	LP7	Fuente de poder
15	1771-IAD	16 Ent. discretas 120 VCA
18	1771-OAD	16 Sal. discretas 120 VCA
2	1771-OEF2	4 Sal. analógicas 4-20 mA
1	1771-IFE	16 ó 8 canales de entrada analógica
1	1771-IXE	8 canales de termopar
4	1771-A4B	Racks de 16 Slots

5.1.2 Distribución de Módulos y Asignación de Direcciones.

La distribución de módulos se hizo de acuerdo a tres criterios:

- 1) Direccionamiento de dos *slots*. Este esquema define 8 grupos modulares por cada *rack* y cada grupo modular está compuesto de dos *slots* adyacentes: el *slot* de la izquierda se utiliza para entradas y el de la derecha para salidas.

- 2) Procurar que los racks agrupen los módulos de acuerdo a las áreas operativas del proceso: el rack 0 está dedicado a la comunicación con la Terminal de Supervisión, el Control de Temperatura y medición de gasto. El rack 1 se dedica a la operación del horno de temple, el rack 2 a la operación del horno de revenido y el rack 3 al Control de la cesta y los tanques.
- 3) Procurar una distribución balanceada de módulos por rack para permitir la posibilidad de una expansión equilibrada.

Los 15 módulos de entrada se colocaron de 5 en 5 en los 3 racks que controlan los subsistemas del proceso. Lo mismo sucedió con los 18 módulos de salida: se repartieron de 6 en 6 en los 3 racks. Las figuras 5-1, 5-2, 5-3 y 5-4 muestran esta distribución física.

Para finalizar el diseño de la arquitectura, se muestra a continuación la asignación de direcciones para cada uno de los elementos de entrada y salida del proceso. Esta asignación se hizo en base a los criterios siguientes:

- 1) Distribuir los elementos de entrada y salida de cada subsistema, de tal forma que cada uno ocupe un rack, esto es, el horno de temple ocupa el rack 2, el horno de revenido el rack 3 y la cesta y los tanques el rack 4.

ELEMENTO	CORRIENTE DE OPERACION (mA)
Arrancador de motor	2.0
Válvula solenoide	0.3
Lámpara piloto	0.2
Relevador para control de falla de flama	0.5

2) Distribuir los elementos de salida procurando que en el peor de los casos (todos los elementos de salida de un módulo estén activos), la corriente demandada por módulo no exceda 8 Amps. Para hacer esta distribución, se consideró que los diferentes elementos de salida utilizados en el proceso consumen las corrientes siguientes:

En las figuras 5-5 a la 5-14 se muestran las asignaciones de direcciones a nivel de *bits* en la tabla de datos del PLC.

PAGE 4 OF _____
 DATE _____
 DESIGNER _____
 RACK NO. 1540 4 SIZE 1040 3

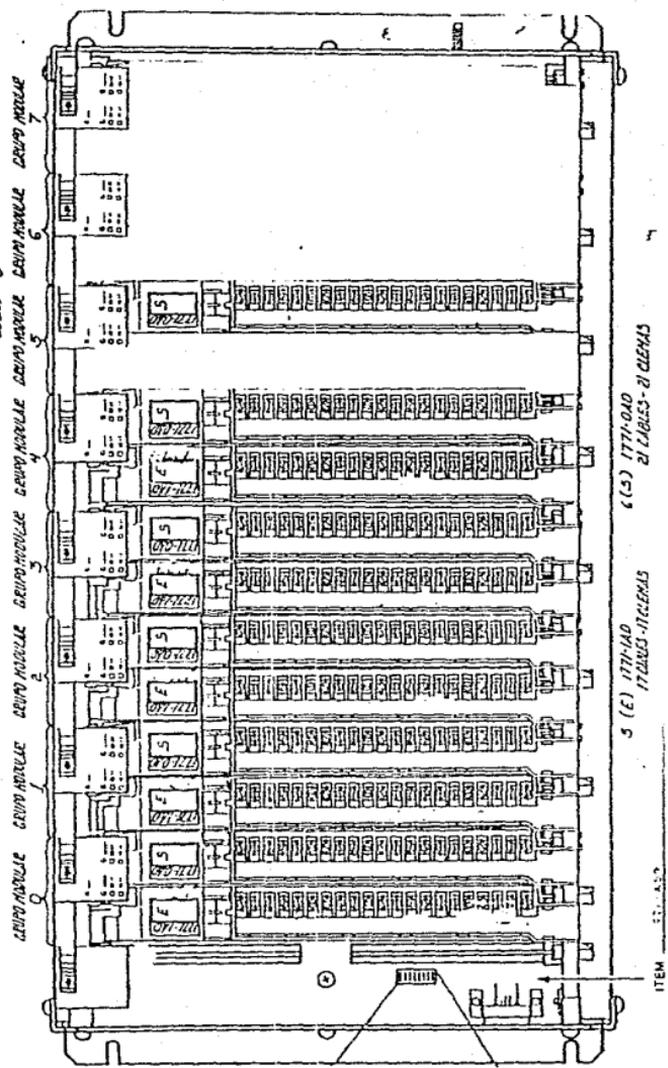


FIGURA 5-4

ALLEN-BRADLEY
PLC-5 Family
Programmable Controller

PAGE 1 OF 1

DATA TABLE BIT ASSIGNMENTS ADDRESS _____ TO _____

PROJECT NAME FORJAMEK PROCESSOR NAME PLC-5/15
 DESIGNER ARI DATA FILE NO 1 TYPE 1

ELEMENT	BIT	DESCRIPTION	ELEMENT	BIT	DESCRIPTION
1:01	0 0		0:02	0 0	
	0 1	PT1 +		0 1	MM1 +
	0 2	PT1 -		0 2	MM4 -
	0 3	PT2 +		0 3	MM5 +
	0 4	PT2 -		0 4	MM5 -
	0 5			0 5	SPACE
	0 6			0 6	SPACE
	0 7			0 7	
	0 8			0 8	
	0 9			0 9	
	1 0			1 0	
	1 1			1 1	
	1 2			1 2	
	1 3			1 3	
	1 4			1 4	
	1 5			1 5	
	1 6			1 6	
1 7		1 7			
0:01	0 0		1:03	0 0	0
	0 1	MM1 +		0 1	1
	0 2	MM1 -		0 2	2
	0 3	MM2 -		0 3	3
	0 4	MM2 +		0 4	4
	0 5	MM3 +		0 5	5
	0 6	MM3 -		0 6	5
	0 7			0 7	7
	0 8			0 8	
	0 9			0 9	
	1 0			1 0	
	1 1			1 1	
	1 2			1 2	
	1 3			1 3	
	1 4			1 4	
	1 5			1 5	
	1 6			1 6	
1 7		1 7			

NOTE: For I/O files, skip bits 08 and 09. For all other files, skip bits 16 and 17

Comments _____

FIGURA 5-5

ALLEN-BRADLEY
PLC-5 Family
Programmable Controller
DATA TABLE BIT ASSIGNMENTS ADDRESS 1:10/00 TO 1:13/17

PAGE 1 OF 4

PROJECT NAME FORJAMEX PROCESSOR NAME PLC-5/15
DESIGNER ABT DATA FILE NO 1 TYPE 1

ELEMENT	BIT	DESCRIPTION	ELEMENT	BIT	DESCRIPTION
1:10	0 0	1b1 P.B. ENCENDIDO TV2	1:12	0 0	1b27 L.S. EMPUJADOR ←
	0 1	1b2 T. APAGADOR		0 1	1b28 P.R. EMPUJADOR
	0 2	1b3 P.B. BOMBA MID. 1		0 2	1b29
	0 3	1b4		0 3	2b2 DISPOSITIVO DE LEVAS
	0 4	1b5 P.B. BOMBA MID. 2		0 4	2b3 GANCHO
	0 5	1b6		0 5	2b4
	0 6	1b7 P.B. BOMBA MID. 3		0 6	2b5 L.S. GANCHO
	0 7	1b8		0 7	2b6 L.S. GANCHO
	0 8			0 8	
	0 9			0 9	
	1 0	DISPOSITIVO DE LEVAS		1 0	2b7 L.S. GANCHO
	1 1	1b9 DEL EXTRACTOR		1 1	2b8 "
	1 2			1 2	2b9 "
	1 3	1b10 P.B. EXTRACTOR		1 3	2b10 "
	1 4	1b11		1 4	2b11 P.B. GANCHO
	1 5	DISPOSITIVO DE LEVAS		1 5	2b12 "
	1 6	1b12 DE LA PUERTA DE SALIDA		1 6	2b13 "
	1 7			1 7	2b14 "
1:11	0 0	1b13 P.B. PUERTA DE SALIDA	1:13	0 0	2b15 "
	0 1	1b14		0 1	2L16 "
	0 2	1b15 L.S. DESCARGADOR →		0 2	2b17 "
	0 3	1b16 L.S. DESCARGADOR →		0 3	2b18 "
	0 4	1b17 L.S. DESCARGADOR ←		0 4	3b5 SEL. ON/OFF CONTADOR
	0 5	1b18 L.S. DESCARGADOR ←		0 5	3b6 CON LLAVE CONTADOR
	0 6	1b19 P.B. DESCARGADOR		0 6	4b P.R.
	0 7	1b20		0 7	4b P.R.
	0 8			0 8	
	0 9			0 9	
	1 0	DISPOSITIVO DE LEVAS		1 0	b10 P.B.
	1 1	1b21 DE LA PUERTA DE ENTRADA		1 1	b11 P.B.
	1 2			1 2	b12 P.B.
	1 3	**1b22* P.B. PUERTA DE ENTRADA		1 3	b13 SEL. ON/OFF
	1 4	1b23		1 4	0E1
	1 5	1b24 L.S. EMPUJADOR →		1 5	0E2 4 ENTRADAS CONTROL
	1 6	1b25 L.S. EMPUJADOR →		1 6	0E3 FALLA DE FLAJA
	1 7	1b26 L.S. EMPUJADOR ←		1 7	0E4 (LA QUEHADRES)

NOTE For I/O files, skip bits 08 and 09. For all other files, skip bits 16 and 17.

Comments HORNO DE TEMPLE DE: 1:10/0 A: 1:20/5
P.B. - PUSH BOTTON L.S. - LIMIT SWITCH SEL. ON/OFF - SELECTOR ON/OFF
* 1b22 NO APARECE EN EL PLANO, SIN EMBARGO, SE APARTA 1:11/13

FIGURA 5-6

Background

ALLEN-BRADLEY
PLC-5 Family
Programmable Controller
DATA TABLE BIT ASSIGNMENTS ADDRESS 1:14/0 TO 1:22/17

PAGE 2 OF 4

PROJECT NAME FORJAMEX PROCESSOR NAME PLC-5/15
DESIGNER ARI DATA FILE NO 1 TYPE L

ELEMENT	BIT	DESCRIPTION	ELEMENT	BIT	DESCRIPTION
1:14	0 0	001	1:21	0 0	5b19 P.B. EXTRACTOR
	0 1	002		0 1	DISPOSITIVO DE LEVAS
	0 2	003 8 ENTRADAS CONTROL		0 2	5b20 DE LA PUERTA DE SALIDA
	0 3	004 FALLA DE FLAMA		0 3	
	0 4	005 (8 QUEMADORES)		0 4	5b21 P.B. PUERTA DE SALIDA
	0 5	006		0 5	5b22
	0 6	007		0 6	5b23 L.S. EXTRACTOR
	0 7	008		0 7	5b24
	0 8			0 8	
	0 9			0 9	
	1 0	009		1 0	5b25 L.S. EXTRACTOR
	1 1	010		1 1	5b26
	1 2	011		1 2	5b27 SEL. ON/OFF EXTRACTOR
	1 3	012 9 ENTRADAS CONTROL		1 3	5b28 P.B. DESCARGADOR
	1 4	013 FALLA DE FLAMA		1 4	5b29
	1 5	014 (9 QUEMADORES)		1 5	DISPOSITIVO DE LEVAS
	1 6	015		1 6	5b30 DE LA PUERTA DE ENTRADA
1 7	016	1 7			
1:20	0 0	017	1:22	0 0	5b31 P.B. PUERTA DE ENTRADA
	0 1	LPS 2 (LOW PRESSURE SWITCH)		0 1	5b32
	0 2	YS8A2 (VAL. SEG. MANUAL)		0 2	5b33 L.S. EMPUJADOR
	0 3	YS8A2 (VAL. SEG. AUTOMATICA)		0 3	5b34
	0 4	HPS2 (HIGH PRESSURE SWITCH)		0 4	5b35
	0 5	SP2 (SWITCH FLUJO DE AIRE)		0 5	5b36
	0 6	5b11 P.B. BOMBA HID. 1		0 6	5b37 P.B. EMPUJADOR
	0 7	5b12		0 7	5b38
	0 8			0 8	
	0 9			0 9	
	1 0	5b13 P.B. BOMBA HID. 2		1 0	6b2 DISPOSITIVO DE LEVAS
	1 1	5b14		1 1	6b3 GANCHO
	1 2	5b15 P.B. BOMBA HID. 3		1 2	6b4
	1 3	5b16		1 3	6b5 L.S. GANCHO
	1 4	DISPOSITIVO DE LEVAS		1 4	6b6
	1 5	5b17 DEL EXTRACTOR		1 5	6b7
	1 6			1 6	6b8
1 7	5b18 P.B. EXTRACTOR	1 7	6b9		

NOTE: For I/O files, skip bits 08 and 09. For all other files, skip bits 16 and 17

Comments HORNO DE REVENIDO DE : 1:20/6 A: 1: 31/5
EL SIGNIFICADO DE LAS ADEVIACIONES ES EL MISMO QUE EL DE LA HOJA NO.1

FIGURA 5-7

ALLEN-BRADLEY
PLC-5 Family
Programmable Controller
DATA TABLE BIT ASSIGNMENTS

PAGE 3 OF 4

ADDRESS 1:23/0 TO 1:31/17

PROJECT NAME FORJAMEX

PROCESSOR NAME PLC-5/15

DESIGNER ARI

DATA FILE NO 1 TYPE 1

ELEMENT	BIT	DESCRIPTION	ELEMENT	BIT	DESCRIPTION
1:23	0 0	6b10 P.B. GANCHO	1:30	0 0	QR5
	0 1	6b11 "		0 1	QR7
	0 2	6b12 "		0 2	QR8
	0 3	6b13 "		0 3	QR9 8 ENTRADAS CONTROL
	0 4	6b14 "		0 4	QR10 FALLA DE FLAMA
	0 5	6b15 "		0 5	QR11 (8 QUEMADORES ZONA 2)
	0 6	6b16 "		0 6	QR12
	0 7	6b17 "		0 7	QR13
	0 8			0 8	
	0 9			0 9	
	1 0	6b20 P.B. ELEVADOR		1 0	QR14
	1 1	6b21 L.S. ELEVADOR		1 1	QR15 5 ENTRADAS CONTROL
	1 2	6b22 P.B. ELEVADOR		1 2	QR16 FALLA DE FLAMA
	1 3	6b23 L.S. ELEVADOR		1 3	QR17 (5 QUEMADORES ZONA 3)
	1 4	5b7 P.B. RECIRCULADOR 1		1 4	QR18
	1 5	5b4		1 5	QR19
	1 6	5b5 P.B. RECIRCULADOR 2		1 6	QR20 4 ENTRADAS CONTROL
	1 7	5b6		1 7	QR21 FALLA DE FLAMA
1:24	0 0	5b7 P.B. RECIRCULADOR 3	1:31	0 0	QR22 (4 QUEMADORES EXTREMOS)
	0 1	*5b8		0 1	IPS7
	0 2	5b1 RECIRCULADOR 4		0 2	VSMAT
	0 3	5b2		0 3	VSAAL
	0 4	(1)		0 4	HPS1
	0 5			0 5	SF1
	0 6	b14 SEL.ON/OFF		0 6	3b1 P.B.
	0 7	b15 P.B.		0 7	3b2 P.B.
	0 8			0 8	
	0 9			0 9	
	1 0	b16 P.B.		1 0	3b3 P.B.
	1 1	b17 P.B.		1 1	3b4 P.B.
	1 2	b19 P.B.		1 2	3b5 L.S.
	1 3	QR1		1 3	3b6 L.S.
	1 4	QR2 5 ENTRADAS CONTROL		1 4	3b7 L.S.
	1 5	QR3 FALLA DE FLAMA		1 5	3b8 L.S.
	1 6	QR4 (5 QUEMADORES ZONA 1)		1 6	3b11 L.S.
	1 7	QR5		1 7	3b12 L.S.

NOTE For I/O files, skip bits 08 and 09. For all other files, skip bits 16 and 17.

Comments *5b8 NO ESTA EN EL PLANO ORIGINAL POSIBLEMENTE FALTE Y SE APARTA SU LUGAR

(1) NO HAY BOTONES DE ENCENDIDO PARA TVI EN EL PLANO, SE APARTA UN LUGAR POR SI FALTA.

FIGURA 5-8

ALLEN-BRADLEY
PLC-5 Family
Programmable Controller PAGE 4 OF 4
DATA TABLE BIT ASSIGNMENTS ADDRESS I:32/0 TO I:34/17

PROJECT NAME FORJAHET PROCESSOR NAME PLC-5/15
DESIGNER ARI DATA FILE NO 1 TYPE L

ELEMENT	BIT	DESCRIPTION	ELEMENT	BIT	DESCRIPTION
I:32	0 0	3b13 L.S.	I:34	0 0	b2A DETECTOR DE NIVEL
	0 1	3b14 L.S.		0 1	T2 TERMOSTATO
	0 2	3b15 L.S.		0 2	T4 TERMOSTATO
	0 3	3b16 L.S.		0 3	b21 MAN. SEL. ON/OFF
	0 4	3b17 L.S.		0 4	b21 AUT. SEL. ON/OFF
	0 5	3b18 L.S.		0 5	4b2 CALENTADOR ON/OFF
	0 6	3b19 L.S.		0 6	4b4 RECIRCULADOR 1
	0 7	3b20 L.S.		0 7	410 RECIRCULADOR 2
	0 8			0 8	
	0 9			0 9	
	1 0	3b21 L.S.		1 0	4b6 SEL. ON/OFF
	1 1	3b22 L.S.		1 1	4b9 SEL. ON/OFF BOMBA DE RESE.
	1 2	3b23		1 2	
	1 3	3b24 L.S.		1 3	
	1 4	3b25 P.B.		1 4	
	1 5	3b26 P.B.		1 5	
	1 6	3b27 P.B.		1 6	
1 7	3b28 P.B.	1 7			
I:33	0 0	3b29 P.B.	0 0		
	0 1	3b30 P.B.	0 1		
	0 2	3b31 P.B.	0 2		
	0 3	3b32 P.B.	0 3		
	0 4	3b33 P.B.	0 4		
	0 5	3b34 P.B.	0 5		
	0 6	3b35 P.B.	0 6		
	0 7	b1A DETECTOR DE NIVEL	0 7		
	0 8		0 8		
	0 9		0 9		
	1 0	T1 TERMOSTATO	1 0		
	1 1	T3 TERMOSTATO	1 1		
	1 2	b20 MAN. SEL. ON/OFF	1 2		
	1 3	b20 AUT. SEL. ON/OFF	1 3		
	1 4	4b1 SEL. ON/OFF (CAL. ON)	1 4		
	1 5	4b3 RECIRCULADOR 1 SEL. ON/OFF	1 5		
	1 6	4b7 RECIRCULADOR 2 SEL. ON/OFF	1 6		
1 7	4b5 SEL. ON/OFF	1 7			

NOTE: For I/O files, skip bits 08 and 09. For all other files, skip bits 16 and 17.

Comments CESTA DE : I:31/6 A: I:33/6
TANQUE DE AGUA DE : I:33/7 A: I:33/17
TANQUE DE ACEITE DE : I:34/0 A: I:34/10

FIGURA 5-9

ALLEN-BRADLEY
PLC-5 Family
Programmable Controller
DATA TABLE BIT ASSIGNMENTS

PAGE 4 OF 8

ADDRESS _____ TO _____

PROJECT NAME FORJAMEX

PROCESSOR NAME PLC-5/15

DESIGNER ARI

DATA FILE NO 0 TYPE 0

ELEMENT	BIT	DESCRIPTION	ELEMENT	BIT	DESCRIPTION
0:10	0 0	1C1 ENCENDIDO TV2	0:12	0 0	2C3 GANCHO LENTO
	0 1	1C1E		0 1	2C4
	0 2	0E1		0 2	2C5 GANCHO RAPIDO
	0 3	0E2		0 3	2C6
	0 4	0E3 4 RELEVADORES		0 4	005
	0 5	0E4 (QUEMADORES DE EXTREMOS)		0 5	006 4 RELEVADORES
	0 6	YSM2		0 6	007
	0 7	VV2		0 7	008
	0 8			0 8	
	0 9			0 9	
	1 0	YSA2		1 0	155 ←
	1 1	h4		1 1	156 →
	1 2	h5		1 2	157 ↑
	1 3	h6		1 3	158 ↓
	1 4	h7		1 4	1h10
	1 5	1h1		1 5	1h11
	1 6	1h2		1 6	1h12
	1 7	1h3		1 7	1h13
0:11	0 0	2C1 GANCHO	0:13	0 0	MS3 EXTREMOS
	0 1	2C2		0 1	MS4
	0 2	00		0 2	009
	0 3	001 4 RELEVADORES		0 3	Q10 4 RELEVADORES
	0 4	002		0 4	Q11
	0 5	003		0 5	Q12
	0 6	1S1 →		0 6	1S9 →
	0 7	1S2 ←		0 7	1S10 ←
	0 8			0 8	
	0 9			0 9	
	1 0	1S3 ↓		1 0	1h14
	1 1	1S4 ↓		1 1	2h1
	1 2	1h4		1 2	2h2
	1 3	1h5		1 3	2h3
	1 4	1h6		1 4	2h4
	1 5	1h7		1 5	2h5
	1 6	1h8		1 6	2h6
	1 7	1h9		1 7	1h15

NOTE: For I/O files, skip bits 08 and 09. For all other files, skip bits 16 and 17.

Comments _____

FIGURA 5-10

ALLEN-BRADLEY
PLC-5 Family
Programmable Controller
DATA TABLE BIT ASSIGNMENTS

PAGE 5 OF 8
ADDRESS 0-14/00 TO 0-21/17

PROJECT NAME FORJANE1
DESIGNER ARI

PROCESSOR NAME PLC-5/15
DATA FILE NO. 0 TYPE 0

ELEMENT	BIT	DESCRIPTION	ELEMENT	BIT	DESCRIPTION	
0:14	0 0	IC1 B.h1	0:20	0 0	3h6	
	0 1	IC2 B.h2		0 1	5C1	TV1 ON
	0 2	Q13		0 2	5C1A	
	0 3	Q14		0 3	5S1	←
	0 4	Q15		0 4	5S2	→
	0 5	Q16		0 5	5S3	↑
	0 6	1S11		0 6	5S4	↓
	0 7	1S12		0 7	5S5	←
	0 8			0 8		
	0 9			0 9		
	1 0	1S13		1 0	5S6	→
	1 1	1h16		1 1	5S7	↓
	1 2	1h17		1 2	5S8	↓
	1 3	1h18		1 3	5S9	→
	1 4	1h19		1 4	5S10	←
	1 5	1h20		1 5	6S14	↑
	1 6	1h21		1 6	6S15	↓
1 7	1h22	1 7	5h1			
0:15	0 0	IC3 B.h3	0:21	0 0	6C1	GANCHO
	0 1	7d1 CIGARRA		0 1	6C2	
	0 2	Q17		0 2	MS1	
	0 3	spare		0 3	MS2	
	0 4	1h23		0 4	5S11	
	0 5	1h24		0 5	5S12	
	0 6	2h7		0 6	5S13	
	0 7	2h8		0 7	VSM1	
	0 8			0 8		
	0 9			0 9		
	1 0	2h9		1 0	VV1	
	1 1	2h10		1 1	VSA1	
	1 2	2h11		1 2	QR21	
	1 3	2h12		1 3	QR22	3 RELEVADORES
	1 4	2h13		1 4	QR1	
	1 5	3h1		1 5	5h2	
	1 6	3h4		1 6	5h3	
1 7	3h5	1 7	5h4			

NOTE: For I/O files, skip bits 08 and 09. For all other files, skip bits 16 and 17.

Comments

FIGURA 5-11

ALLEN-BRADLEY
PLC-5 Family
Programmable Controller PAGE 6 OF B
DATA TABLE BIT ASSIGNMENTS ADDRESS 0:22/0 TO 0:25/17

PROJECT NAME FORJAMES PROCESSOR NAME PLC-5/15
DESIGNER ART DATA FILE NO. 1 TYPE 1

ELEMENT	BIT	DESCRIPTION	ELEMENT	BIT	DESCRIPTION
0:22	0	SC3 LENTO	0:24	0	5C20
	1	SC4		1	5C21
	2	SC5 RAPIDO		2	SC4
	3	SC6		3	QR10 4 RELEVADORES
	4	SC2		4	QR11
	5	QR		5	QR12
	6	QR 4 RELEVADORES		6	QR13
	7	QR		7	5h21
	8			8	
	9			9	
	10	QR5		10	5h22
	11	5h5		11	5h23
	12	5h6		12	5h24
	13	5h7		13	5h25
	14	5h8		14	5h26
	15	5h9		15	5h27
	16	5h10		16	5h28
17	5h11	17	6h1		
0:23	0	5C18	0:25	0	5C7
	1	5C19		1	5C7.1
	2	SC1		2	5C5
	3	QR 4 RELEVADORES		3	QR14 4 RELEVADORES
	4	QR		4	QR15
	5	QR6		5	QR16
	6	QR		6	QR17
	7	5h12		7	6h2
	8			8	
	9			9	
	10	5h13		10	6h3
	11	5h14		11	6h4
	12	5h15		12	6h5
	13	5h16		13	6h6
	14	5h17		14	6h7
	15	5h18		15	6h8
	16	5h19		16	6h9
17	5h20	17	6h10		

NOTE: For I/O files, skip bits 08 and 09. For all other files, skip bits 16 and 17.

Comments _____

FIGURA 5-12

ALLEN-BRADLEY
PLC-5 Family
Programmable Controller

PAGE 2 OF 8

DATA TABLE BIT ASSIGNMENTS ADDRESS 0:10/00 TO 0:32/17

PROJECT NAME F02JAMEX PROCESSOR NAME PLC-5/15
DESIGNER ARI DATA FILE NO 1 TYPE I

ELEMENT	BIT	DESCRIPTION	ELEMENT	BIT	DESCRIPTION
0:30	0 0	SCR	0:37	0 0	3C2 MOTOR 2
	0 1	SCR 1		0 1	3C6 MOTOR 4
	0 2	SCR		0 2	3C7
	0 3	OR18		0 3	3h11
	0 4	OR19		0 4	3h12
	0 5	OR 20		0 5	3h13
	0 6	SPARE		0 6	3h14
	0 7	6h11		0 7	3h15
	0 8			0 8	
	0 9			0 9	
	1 0	6h12		1 0	3h16
	1 1	3h13		1 1	3h17
	1 2	6h20		1 2	3h17A
	1 3	6h21		1 3	3h18
	1 4	6h22		1 4	3h18A
	1 5	6h23		1 5	3h19
	1 6	h10		1 6	3h20
	1 7	h11		1 7	3h21
0:31	0 0	h12	0:33	0 0	3C8
	0 1	h13		0 1	3S1 ↑
	0 2	3C1		0 2	3S2 ↓
	0 3	3C3		0 3	3S3 →
	0 4	3C4		0 4	3S4 ←
	0 5	M1		0 5	3h22
	0 6	3h1		0 6	3h23
	0 7	3h2		0 7	3h24
	0 8			0 8	
	0 9			0 9	
	1 0	3h2A		1 0	3h25
	1 1	3h4		1 1	3h26
	1 2	3h5		1 2	3S1
	1 3	3h6		1 3	3S2
	1 4	3h7		1 4	4C1
	1 5	3h8		1 5	4C2
	1 6	3h9		1 6	4C5
	1 7	3h10		1 7	4C6

NOTE: For I/O files, skip bits 08 and 09. For all other files, skip bits 16 and 17.

Comments _____

FIGURA 5-13

ALLEN-BRADLEY
PLC-5 Family
Programmable Controller
DATA TABLE BIT ASSIGNMENTS

PAGE 3 of 3

ADDRESS 0:34/00 TO 0:35/17

PROJECT NAME FORJANEX

PROCESSOR NAME PLC-5/15

DESIGNER API

DATA FILE NO 1 TYPE 1

ELEMENT	BIT	DESCRIPTION	ELEMENT	BIT	DESCRIPTION
0:34	0	4C3 T.AGUA B2		0	
	1	4C3.1		1	
	2	h20		2	
	3	h1		3	
	4	h1A		4	
	5	h1		5	
	6	h5		6	
	7	h7		7	
	8			8	
	9			9	
	10	1C4		10	
	11	7C1		11	
	12	4C7		12	
	13	h5		13	
	14	C1		14	
	15			15	
	16			16	
0:35	0	4C4 T.ACETE B2		0	
	1	4C4.1		1	
	2	h21		2	
	3	h2		3	
	4	h2A		4	
	5	h4		5	
	6	h6		6	
	7	h8		7	
	8			8	
	9			9	
	10	1C0		10	
	11	7C2		11	
	12	C2		12	
	13			13	
	14			14	
	15			15	
	16			16	
17			17		

NOTE For I/O files, skip bits 08 and 09 For all other files, skip bits 16 and 17.

Comments

FIGURA 5-14

5.2 Programa que Governa la Arquitectura.

La parte más importante del *Background* es el programa que toma las decisiones a partir de la información del proceso y ordena las acciones adecuadas. Si el *Background* está constituido por un Controlador Programable, como es el caso de nuestra tesis, este programa será una secuencia de instrucciones simbólicas en Lenguaje Escalera que residen en la memoria RAM (respaldada por batería) del procesador. Este programa será ejecutado de acuerdo a la secuencia de ejecución denominada SCAN, concepto que ya fué tratado con anterioridad.

Para el lector interesado, a continuación mostramos la estructura del programa de Control junto con algunos escalones que ejemplifican la programación en Lenguaje Escalera del PLC 5-15.

1. Comunicación con módulo BASIC.

OPERACION HORNO DE TEMPLE FORJAMEX 15/NOV/89 15 March 1990 Page 1
 Ladder Listing Processor File: HORNNOTE.ACH Rung 2:0

Rung 2:0

Transmisión de datos de status a la terminal supervisora (T-35) v/a
 módulo BASIC.

```

| W7:531 W7:536                                *BTW-----*
| --J/[-----J/[-----]BLOCK TRNSFR WRITE |-(EM)-| |
|   15   15                                     |Rack      0|
|                                               |Group     7|-(DM)|
|                                               |Module    1|
|                                               |Control Block W7:531|-(ER)|
|                                               |Data file   W7:223|
|                                               |Length     64|
|                                               |Continuous  W|
|-----|-----|-----|-----|-----|

```

Rung 2:1

Recepción de datos y comandos de la terminal supervisora (T-35) v/a
 módulo BASIC.

```

| W7:536 W7:531                                *BTR-----*
| --J/[-----J/[-----]BLOCK TRNSFR READ  |-(EM)-| |
|   15   15                                     |Rack      0|
|                                               |Group     7|-(DM)|
|                                               |Module    1|
|                                               |Control Block W7:536|-(ER)|
|                                               |Data file   W7:287|
|                                               |Length     64|
|                                               |Continuous  W|
|-----|-----|-----|-----|-----|

```

2. Configuración módulo de termopares.

Rung 2:5

Configuración del módulo de termopares.

```

| W7:511 W7:516                                *BTW-----*
| --J/[-----J/[-----]BLOCK TRNSFR WRITE |-(EM)-| |
|   15   15                                     |Rack      0|
|                                               |Group     0|-(DM)|
|                                               |Module    0|
|-----|-----|-----|-----|-----|

```

```

| Control Block  N7:511 |-(ER)|
| Data file     N7:194 |
| Length       27 |
| Continuous   N |
|-----|

```

3. Lectura de temperaturas.

Lectura de temperaturas y status del módulo de termopares.

```

| N7:516 N7:511 |
| -J/(-) | | |
| 15 15 | | |
|-----|
| *BTR-----* |
| BLOCK TRNSFR READ | |-(EN)-|
| Rack              0 |
| Group             0 |-(DN)|
| Module            0 |
| Control Block     N7:516 |-(ER)|
| Data file         N7:211 |
| Length            12 |
| Continuous        N |
|-----|

```

4. Transferencia de temperatura a la T-35.

Rung 2:7.

Transmisión de temperaturas a la T-35.

```

|-----|
| Temp. |
| ZONA-1HT |
| *MOV-----* |
| |MOVE | |
| |Source N7:220 |
| | 794 |
| |Dest N7:255 |
| | 794 |
|-----|
| Temp. |
| ZONA-2HT |
| *MOV-----* |
| |MOVE | |
| |Source N7:219 |
| | 790 |
| |Dest N7:256 |
| | 790 |
|-----|
| Temp. |
| ZONA-1HR |
| *MOV-----* |

```



```

N7:506 N7:501          *BTR-----*
--]/[---]/[-----|BLOCK TRANSFR READ |-(EN)-|
  15  15             |Rack          0|
                    |Group          1|-(DN)|
                    |Module         0|
                    |Control Block  N7:506|-(ER)|
                    |Data file     N7:152|
                    |Length         6|
                    |Continuous    N|
                    *-----*

```

7. Transferencia de presiones a T-35.

Rung 2:4

Transmisión de los datos de presión a la T-35.

```

                    *MOV-----*
                    |MOVE          |-*|
                    |Source       N7:156|
                    |              0|
                    |Dest        N7:257|
                    |              0|
                    *-----*
                    *MOV-----*
                    |MOVE          |-*|
                    |Source       N7:157|
                    |              0|
                    |Dest        N7:268|
                    |              0|
                    *-----*

```

8. Control de apertura de válvulas.

Rung 2:9

Transferencia de datos de apertura a las válvulas proporcionales vía módulo de salidas analógicas. Válvulas: MM1, MM2 y MM3.

```

N7:526                *BTW-----*
--]/[---]/[-----|BLOCK TRANSFR WRITE |-(EN)-|
  15                  |Rack          0|
                    |Group          1|-(DN)|
                    |Module         1|
                    |Control Block  N7:526|-(ER)|
                    |Data file     N7:171|
                    *-----*

```

Length	13
Continuous	N

Rung 2:10

Transferencia de datos de apertura a las válvulas proporcionales vía módulo de salidas analógicas. Válvulas: MM4 y MM5.

N7:521	*BTW-----*
--1/(-	BLOCK TRANSF WRITE -(EN)-
15	Rack 0
	Group 2 -(DN)
	Module 1
	Control Block N7:521 -(ER)
	Data file N7:158
	Length 13
	Continuous N

Rung 2:20

Transferencia del grado de apertura (0X al 100%) de las válvulas MM1 y MM2 a la T-35.

	Apertura válvula MM1.
	MOV-----
	MOVE
	Source N7:171
	570
	Dest N7:260
	570
	Apertura válvula MM2.
	MOV-----
	MOVE
	Source N7:172
	2070
	Dest N7:261
	2070

10. Purga del horno de Revenido y del horno de Revenido
 12. Control de temperatura del Horno de Temple y Revenido

Rung 2:178

Inicio de la secuencia de encendido de los quemadores del horno de revenido.
 Encendido del turboventilador:



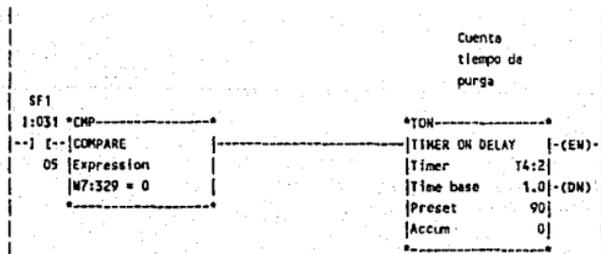
Rung 2:179

El arrancador auxiliar permanece activado hasta que transcurra el tiempo de purga y se alcance la presión necesaria de trabajo.



Rung 2:180

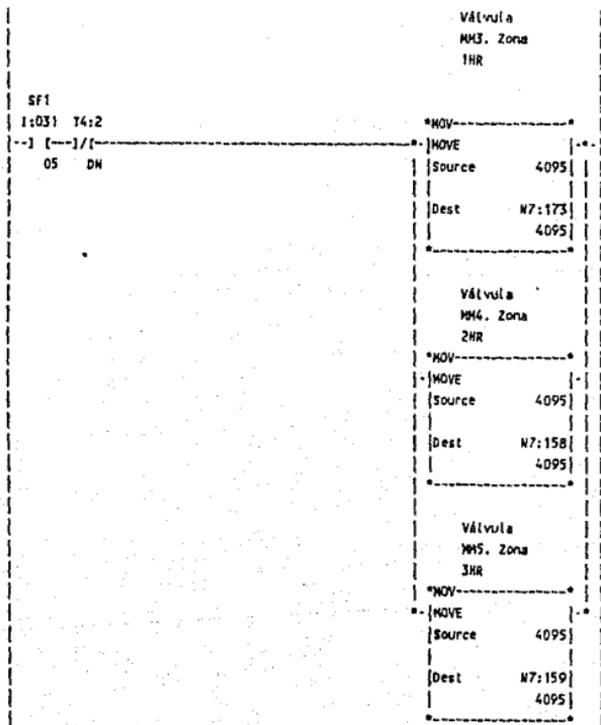
Iniciá el tiempo de purga a partir de que hay presión de aire en la línea.
 (HR).



Background

Rung 2:181

Apertura al 100% de las válvulas proporcionales, durante el tiempo de purga.
 Las válvulas son: MM3, MM4 y MM5.



Rung 2:182

Consultación manual/automático de las zonas de control de temperatura 1HR, 2HR y 3HR. Los comandos son enviados por la I-35.



zona 1HR	
T00-----	
[TO BCD]
[Source	N7:306
	900
[Dest	N7:188
	2304
-----*	
Lim. inf. de temp. zona 1HR	
T00-----	
[TO BCD]
[Source	N7:307
	600
[Dest	N7:187
	1536
-----*	
Set point zona 2HR	
MOV-----	
[MOVE]
[Source	N7:308
	650
[Dest	N7:25
	650
-----*	
Lim. sup. de temp. zona 2HR	
T00-----	
[TO BCD]
[Source	N7:309
	950
[Dest	N7:190
	2384
-----*	
Lim. inf. de temp. zona 2HR	
T00-----	
[TO BCD]
[Source	N7:310
	400

```

| [Dest      N7:189] |
| | 1024 |
| *-----* |
| | |
| | Set point |
| | zona 3HR |
| *NOV-----* |
| | NOVE | |
| | Source  N7:311 |
| | 600 |
| | Dest     N7:48 |
| | 600 |
| *-----* |
| | |
| | Lim. sup. |
| | de temp. |
| | zona 3HR |
| *TOO-----* |
| | TO BCD | |
| | Source  N7:312 |
| | 950 |
| | Dest     N7:192 |
| | 2384 |
| *-----* |
| | |
| | Lim. inf. |
| | de temp. |
| | zona 3HR |
| *LOO-----* |
| | TO BCD | |
| | Source  N7:313 |
| | 400 |
| | Dest     N7:191 |
| | 1024 |
| *-----* |

```

Rung 2:185

Instrucciones de control de temperatura para las zonas 1HR, 2HR y 3HR.
 La ley de control es proporcional con opción a implementar un algoritmo
 de control proporcional, integral y derivativo.

```

| | | | Control P | |
| | | | zona 1HR |
| | | | *PID-----* |
| T4:3 | | | PID | |
| | | | |
| | | | [Control block N7:0] |
| | | | |

```

```

| Process variable N7:353 |
| Tieback N7:316 |
| Control variable N7:173 |
|-----|
| Control P |
| zona 2HR |
|-----|
| *PID-----* |
| PID |
| Control block N7:23 |
| Process variable N7:354 |
| Tieback N7:317 |
| Control variable N7:158 |
|-----|
| Control P |
| zona 3HR |
|-----|
| *PID-----* |
| PID |
| Control block N7:46 |
| Process variable N7:355 |
| Tieback N7:318 |
| Control variable N7:159 |
|-----|

```

13. Movimientos del Horno de Temple y Revenido
 14. Control automático de movimientos de ambos hornos

```

Rung 2:215
| Paro de |
| emergencia |
| horno de |
| revenido |
| movs. mec. |
|-----|
| I:024 | ARR_C2 |
|-----| O:035 |
|-----] ( )-----|
| 12 | 12 |

```

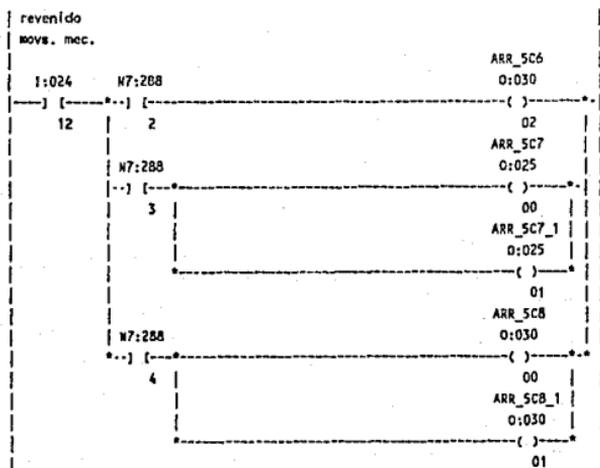
Rung 2:216

Activación de las bombas hidráulicas del horno de revenido (vía T-35).

```

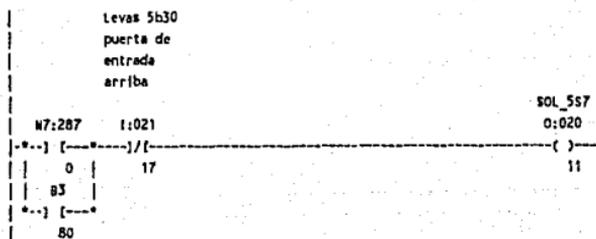
| Paro de |
| emergencia |
| horno de |

```



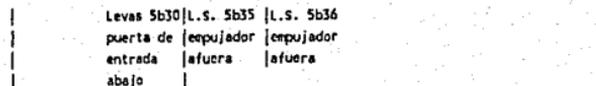
Rung 2:217

Apertura de la puerta de entrada del horno de revenido. (MAN/AUTO).



Rung 2:218

Cierre de la puerta de entrada (HR). (MAN/AUTO).



```

|                                     SOL_558 |
| N7:287 1:021 1:022 1:022 0:020 |
| *--] [---*--] [---*--] [---*--] [---*--] |
| | 1 | 15 04 05 12 | | | |
| | B5 | | | | |
| | *--] [---*-- |
| | 81 | | | | |

```

Rung:219

Introducción del empujador (HR). (MAN/AUTO).

```

|                                     SOL_559 |
| L.S. 5b33 Levas 5b30 |
| Empujador puerta de |
| adentro entrada |
| arriba |
|                                     |
| N7:287 1:022 1:021 0:020 |
| *--] [---*--] [---*--] [---*--] |
| | 2 | 02 17 13 |
| | | L.S. 5b34 |
| | | Empujador |
| | | adentro |
| | B5 | 1:022 |
| | *--] [---*--] [---*--] |
| | 82 03 |

```

Rung:220

Extracción del empujador (HR). (MAN/AUTO).

```

|                                     SOL_5610 |
| L.S. 5b35 |
| empujador |
| afuera |
|                                     |
| N7:287 1:022 0:020 |
| *--] [---*--] [---*--] |
| | 3 | 04 14 |
| | | L.S. 5b36 |
| | | empujador |
| | | afuera |
| | B3 | 1:022 |
| | *--] [---*--] [---*--] |
| | 83 05 |

```

Rung:221

Background

Extracción del extractor (MR). (MAN/AUTO).

	Levas 5b17	Levas 5b20	L.S. 6b23	
	extractor	puerta de	Elevador	
	afuera	salida	abajo	
		arriba		
				SOL_552
N7:287	1:020	1:021	1:023	0:020
7	16	03	13	04
83				
87				

Rung 2:225

Apertura automática de la válvula de alivio 5s13

SOL_557		SOL_553
0:020		0:021
11		06
SOL_558		
0:020		
12		
SOL_559		
0:020		
13		
SOL_563		
0:020		
05		
SOL_564		
0:020		
06		
SOL_552		
0:020		
04		
SOL_551		
0:020		
03		

Rung 2:226

Apertura automática de las válvulas de alivio 5S11 y 5S12.

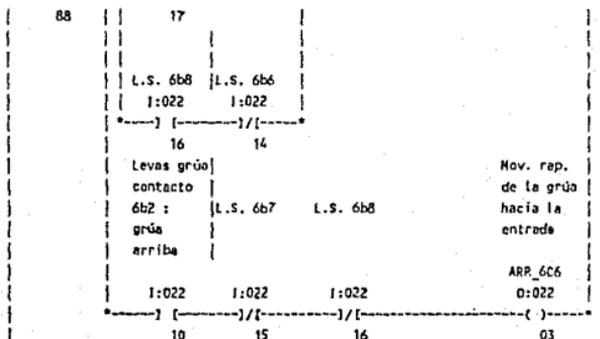
SOL_5S9		SOL_5S11
0:020		0:021
13		04
SOL_5S10		SOL_5S12
0:020		0:021
14		05
SOL_5S5		
0:020		
07		
SOL_5S6		
0:020		
10		
SOL_6S14		
0:020		
15		
SOL_6S15		
0:020		
16		

Rung 2:227

Movimiento, lento y rápido, de la grúa hacia la entrada. (HR).

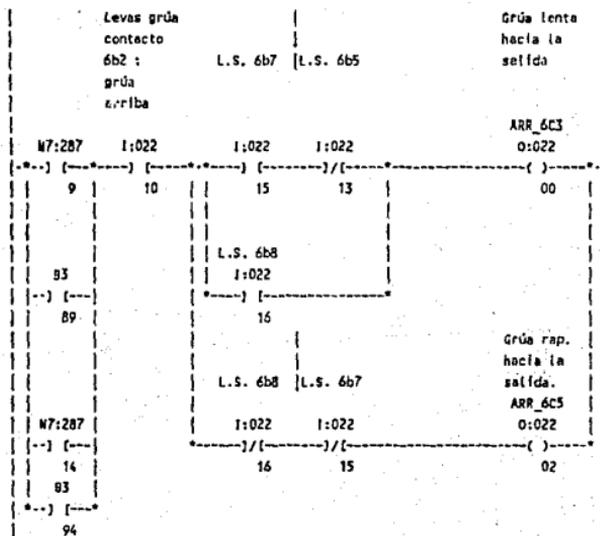
	Levas grúa				Mov. lento
	contacto				de la grúa
	6b2 :	L.S. 6b7			hacia la
	grúa				entrada
	arriba				
					ARR_6C4
H7:2B7	1:022	1:022			0:022
8	10	15			01
		L.S. 6b9			
B3		1:022			

Background



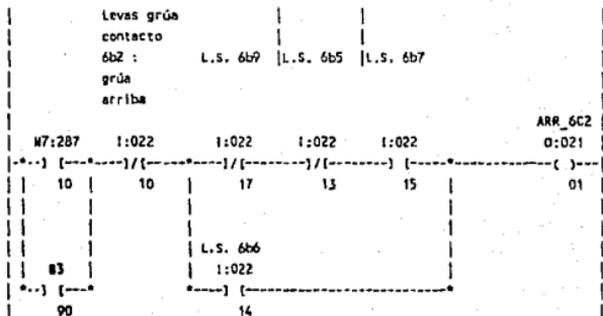
Rung 2:228

Movimiento rápido y lento de la grúa hacia la salida. (NR).



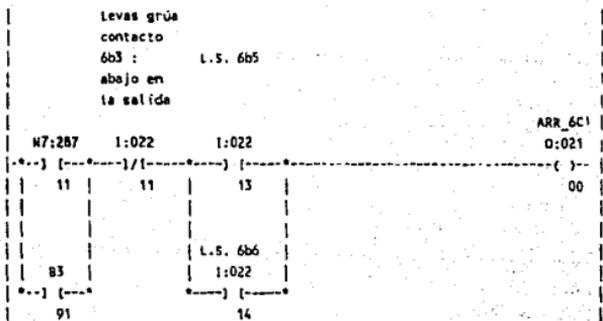
Rung 2:229

Subir grúa. (HR).



Rung 2:230

Bajar grúa. (HR).



Rung 2:231

Subir elevador. (HR).



Background

			SOL_6514
N7:287	I:023	I:022	0:020
12	11	10	15
83			
92			

Rung 2:232

Bajar elevador. (HR).

			SOL_6515
L.S. 6b23			
Elevador			
abajo			
N7:287	I:023		0:020
13	13		16
83			
93			

Rung 2:233

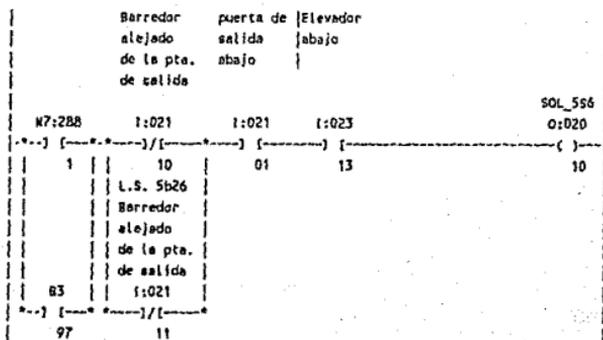
Colocar descargador (barredor) sobre la puerta de salida. (HR).

			SOL_555
L.S. 5b23			
barredor			
sobre la			
puerta			
Levas 5b20			
puerta de			
salida			
abajo			
N7:288	I:021	I:021	0:020
0	06	01	07
	L.S. 5b24		
	barredor		
	sobre la		
	puerta		
83	I:021		
96	07		

Rung 2:234

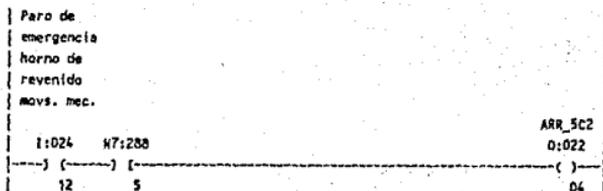
Atejar descargador (barredor) de la puerta.

L.S. 5b25	Levas 5b20	L.S. 6b23
-----------	------------	-----------



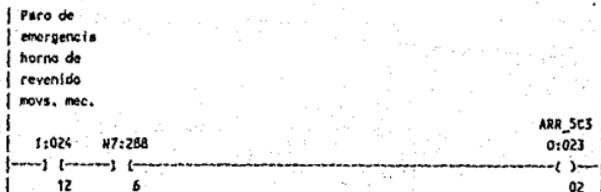
Rung 2:235

Activación del recirculador de aire No.1 (NR).



Rung 2:236

Activación del recirculador de aire No.2 (NR).



Rung 2:237

Activación del recirculador de aire No.3 (HR).

	Paro de	
	emergencia	
	horno de	
	revenido	
	movs. mec.	
		ARR_SCA
	1:024 N7:288	0:024
	-----	-----
	12 7	02

Rung 2:238

Activación del recirculador de aire No.4 (HR).

	Paro de	
	emergencia	
	horno de	
	revenido	
	movs. Mec.	
		ARR_SCS
	1:024 N7:288	0:025
	-----	-----
	12 8	02

Rung 2:239

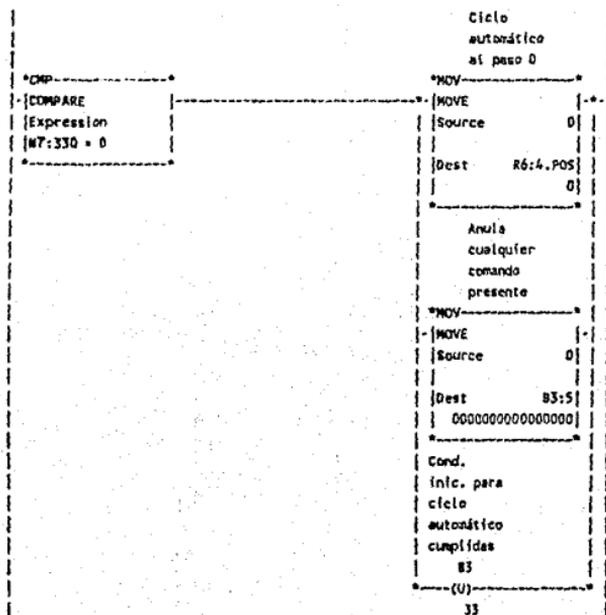
Establece condiciones iniciales para iniciar la operación en ciclo automático.

	Levas 5b17		Levas 5b20		L.S. 5b23		L.S. 5b24		Levas 5b30		L.S. 5b35		L.S. 5b36	
	extractor		puerta de		barredor		barredor		puerta de		empujador		empujador	
	afuera		salida		sobre la		sobre la		entrada		afuera		afuera	
			abajo		puerta		puerta		abajo					
	1:020		1:021		1:021		1:021		1:021		1:022		1:022	
	-----		-----		-----		-----		-----		-----		-----	
	16		01		06		07		15		04		05	

	Levas grúa		L.S. 6b23		Cond.	
	contacto		Eleveador		inic. para	
	6b2 :		L.S. 6b5		abajo	
	grúa				ciclo	
	arriba				automático	
	< 1:022		1:022		1:023	
	83					
	<-----		-----		-----	
	< 10		13		13	
					33	

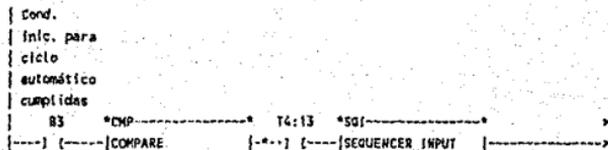
Rung 2:240

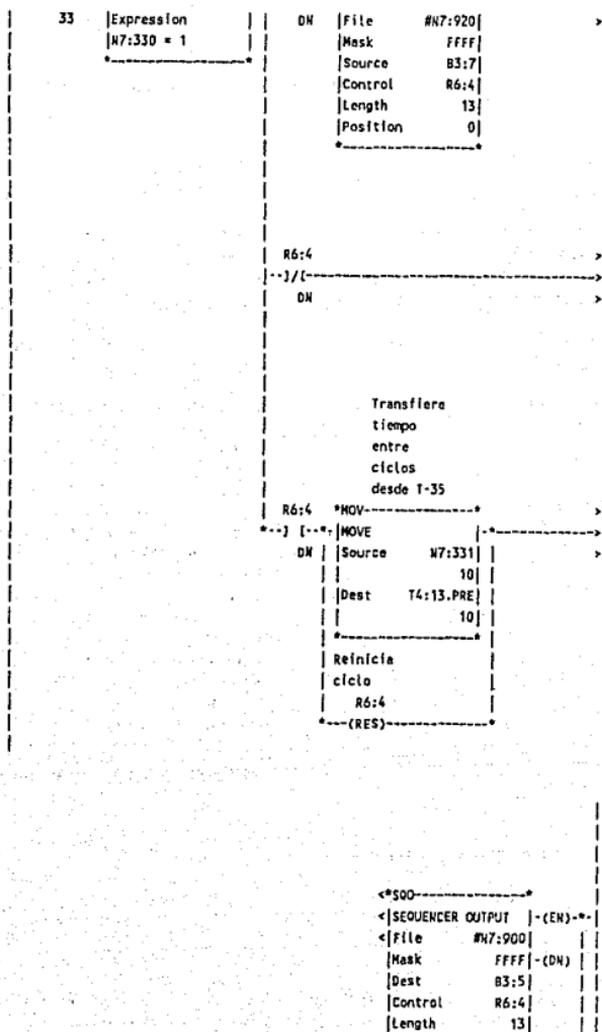
Comutación de la operación automática del horno de revenido a la operación manual.



Rung 2:241

Control de la operación en ciclo automático. (RR).





```
|Position      0|
+-----+
      Cuenta
      tiempo
      entre
      ciclos
<---TON----->
<|TIMER ON DELAY  |-(EN)-|
<|Timer      T4:13|
|Time base   1.0|-(DN)|
|Preset      10|
|Accum       0|
+-----+

<
<----->
<
```

6. Solución Propuesta : Foreground

6 Solución Propuesta: Foreground.

En el capítulo anterior determinamos con qué y cómo vamos a dar solución al problema planteado. La solución para el *Background* está constituida por un Controlador Programable y por supuesto, su programa en Escalera. Por medio del programa en Escalera el Controlador Programable es capaz de dirigir en forma automática la secuencia de encendido de los quemadores de ambos hornos, regular la temperatura de éstos, supervisar la operación del sistema de combustión e implantar las medidas de seguridad sugeridas por ENTERPRISE S.A. que a su vez fueron adecuadas por el cliente.

El Controlador Programable está programado para dirigir los movimientos de elementos de transporte de charolas y piezas en forma automática o Manual. Se debe hacer notar, sin embargo, que todas estas habilidades implantadas a través del *Background* sirven únicamente para ejecutar decisiones humanas sobre la producción. Luego entonces, surge la necesidad de un operador que indique al Controlador Programable las operaciones que quiere realizar. En otras palabras, se presenta el problema de cómo lograr que el *Background* interactúe con un operador.

Resumiendo, el *Background* resuelve el problema de Control de los elementos de campo y adquisición de datos del proceso, sin embargo existe una nueva necesidad: una interfase de operador.

6.1 Requerimientos Generales

Para atacar el problema citado, consideremos ahora los requerimientos para la interfase de operador del proyecto en cuestión:

1) Control y Supervisión del sistema mecánico de cada horno:

- Modalidad de operación Manual o Automática.
- Seleccionar cualesquiera de 8 ciclos de operación Automática.
- Establecer los tiempos de permanencia en tanques y espera entre cada ciclo, asociados a cada una de las modalidades de operación Automática.
- Cambios aleatorios entre operación Manual y Automática
- Control Manual e independiente de cada uno de los Movimientos de los elementos de transporte de charolas y piezas.
- Indicar la posición de cada elemento y la modalidad de operación: Manual o Automática.

2) Control y Supervisión del sistema de combustión de cada horno:

- Arranque y paro del sistema en forma arbitraria según necesidades del operador.
- Control Automático del tiempo de purga, secuencia de encendido de quemadores y apertura de las válvulas de seguridad.
- Apagado Automático de quemadores y cierre Automático de válvulas de seguridad al violarse las condiciones mínimas de seguridad en la operación.
- Control Automático o Manual de las válvulas proporcionales para el Control de la temperatura en cada zona de cada horno.
- Establecer valor de referencia para regulación (*set point*) y límites de alarma para temperatura.
- Indicar la modalidad de operación para cada zona de temperatura.

- Indicar valores de *set point*, temperatura, límites de alarma y el porcentaje de apertura de la válvula proporcional en cada zona.
- Indicar el estado de cada quemador: apagado o encendido.

Construir esta interfase de operador por los medios tradicionales implica permanecer en el nivel del *Background*, esto es, identificar las entradas y los elementos de *hardware* asociados para que el operador ordene al Controlador Programable y por su lado, identificar las salidas y los elementos de *hardware* asociados para que el Controlador Programable proporcione información al operador.

Si traducimos lo anterior a una relación más concreta tendríamos lo siguiente:

ENTRADAS	ELEMENTOS DE HARDWARE
Selección de modalidades y control de movimientos.	Selectores y <i>Push-Button</i> .
Establecimiento de valores y parámetros de Control.	Potenciómetros y <i>Thumb-Wheels</i> .

SALIDAS	ELEMENTOS DE HARDWARE
<p>Posición y ubicación de los elementos mecánicos y estado de los quemadores.</p>	<p>Lámparas en un mímico.</p>
<p>Valores de parámetros y variables de proceso.</p>	<p>Pantallas con presentación analógica o digital.</p>
<p>Registro de alarmas.</p>	<p>Monitores de alarmas.</p>
<p>Indicación de gasto de gas en cada horno.</p>	<p>Totalizadores de flujo.</p>
<p>Registro de tendencias de variables.</p>	<p>Registadores de multipunto.</p>

Esta alternativa presenta las siguientes ventajas y desventajas inherentes:

Ventajas:

- Facilidad de diseño.
- Instalación según manuales.
- Independencia de funciones y por lo tanto, menor probabilidad de falla total.

Desventajas:

- Implica el diseño de tableros voluminosos de operador
- Mayor consumo de potencia
- Escasa versatilidad y flexibilidad.
- Menor confiabilidad debido al tendido de cables extra que requiere.
- Costos mayores en elementos de marcas diferentes
- Mayor número de proveedores y por lo tanto, mayores problemas para servicio y mantenimiento.

Además presenta una desventaja mayor para los criterios emergentes con las ventajas que trae consigo la Automatización y el uso de Controladores Programables: desaprovecha la capacidad de adquisición de datos y centralización de información inherente a un Controlador Programable.

En efecto, el Controlador Programable tiene en su tabla de datos toda la información asociada al proceso que esta controlando. Además, proporciona los medios adecuados para modificar esta tabla y por lo tanto, de influir directamente sobre los elementos de campo. De lo anterior y considerando que esta información es digital, podemos concluir que la interfase de operador que cumple con los requisitos de este proceso en particular y seguramente de muchos otros, es aquella combinación de *hardware* y *software* capaz de recibir e interpretar la información digital del Controlador Programable y transmitir comandos

para modificar esta información. En este punto estamos hablando de que la interfase de operador será implantada por la herramienta que ha demostrado su utilidad para comunicar información digital y procesarla: una computadora.

En la selección del tipo de computadora debe considerarse lo siguiente:

- Debe ser un equipo capaz de soportar los rigores del ambiente industrial: humedad, altas temperaturas, vibraciones, polvo, etc
- Debe ser un equipo de bajo costo.

Los equipos que se cumplen con estas dos características son:

- Una Terminal de Programación de PLC's
- Una Computadora Personal (PC) adaptada como Terminal Industrial.

Existen a su vez tres alternativas considerando el enfoque de emplear, ya sea una terminal de programación o una computadora personal, como elemento de *hardware* para la interfase de operador:

1) Utilizar una Terminal de Programación de PLC's y el software necesario para supervisar, a nivel de *bits*, el estado del proceso, esto es, realizar labores de Supervisión y Control en el nivel más bajo.

2) Utilizar una Terminal Industrial que se comunique a través del puerto serie con el PLC, empleando un dispositivo intermedio como interfase de comunicación.

3) Utilizar una PC que se comunique con el PLC a través del *bus* del *rack* (PC integrada al Controlador Programable).

La alternativa 1) fue descartada porque una Terminal de Programación requiere un conocimiento técnico especializado que no es importante para el operador. La alternativa 3) se descartó porque esta solución existe únicamente para el *VMEbus* y este no es el *bus* utilizado en el proyecto. Así pues, la solución para implantar la interfase de operador requerida es utilizar una PC (tipo AT) con los elementos defensivos adecuados contra el ambiente hostil que hace que se le denomine Terminal Industrial.

Concretamente, la Terminal Industrial utilizada en el proyecto es el modelo T-35 de Allen-Bradley. Consultar el Apéndice C para detalles sobre las características de la Terminal Industrial modelo T-35.

La Terminal Industrial, y lo más importante, el *software* de Supervisión y Control (BIKO) a instalarse en ésta, constituyen un nivel superior al *Background* que se denomina *Foreground*. Los detalles del desarrollo de este nivel se tratará más adelante en el capítulo 7.

Para comprender con mayor claridad los dos niveles de la solución, presentaremos como trabajan conjuntamente el *Background* y el *Foreground*.

Tenemos un sistema donde distinguimos dos partes: *Foreground* y *Background*. En este sistema, el *Background* se comporta como un Sistema Operativo en tiempo real que se encarga de controlar y administrar los recursos a bajo nivel del sistema: los elementos de campo. Por su lado, el *Foreground* viene a ser un programa de aplicación que se encarga de atender al usuario en cualquier momento y traducir la información proveniente del PLC a una forma entendible para el usuario, y viceversa, interpretar las órdenes del usuario para comunicarlás al PLC.

Visto como un sistema global, es decir, como una caja negra, podríamos ver al *Background* y al *Foreground* como procesos concurrentes. Pero, dado que éstos se están ejecutando en procesadores independientes, el programa de Control en el PLC y el programa de Supervisión en la T-35, en realidad estamos hablando de procesos en paralelo.

Sigamos una orden del operador hacia un elemento de campo:

- El operador da una orden al programa de Supervisión y Control a través del teclado de la Terminal T-35.
- El programa traduce esta tecla a una orden concreta y la transmite, vía el puerto serie, al dispositivo intermedio de interfase entre la T-35 y el PLC.
- El Módulo intermedio de interfase almacena la orden y la traduce para transmitirla vía el *bus* del *rack*, sobre el cuál esta montado, hacia el PLC.

- El PLC interpreta el contenido de la información recibida según el programa de escalera y ejecuta la tarea según su *SCAN*.

En tanto, la información de campo hacia el usuario:

- El PLC registra el estado de entradas y salidas del proceso según su *SCAN*, y de acuerdo a su programa de escalera, transmite la información vía el *bus* del *rack* al Módulo intermedio de interfase.
- El Módulo intermedio de interfase recibe la información, la traduce y la transmite vía su puerto periférico.
- La terminal recibe la información vía interrupción y la almacena.
- El software de Control y Supervisión traduce estos datos y los presenta como información gráfica y numérica al operador.

Hasta ahora estos detalles nos sirven para recalcar cómo el *Foreground* hace peticiones al *Background* y que el programa clave para la administración de los recursos es el *Background*. Cabe llamar la atención sobre el grado de paralelismo existente entre el *Foreground* y el *Background*. El *Background* es un Sistema Operativo de tiempo real con capacidad de ejecutarse en paralelo con un programa de aplicación: *Foreground*.

6.2 Niveles de Comunicación Foreground-Background.

El utilizar una Terminal Industrial como interfase de operador para un Controlador Programable implica otro problema. Debido a que la Terminal Industrial y el PLC no comparten el mismo *bus* se presenta la necesidad de establecer una comunicación entre éstos, es decir, emplear un dispositivo intermedio como interfase (no confundirse con la interfase de Supervisión), en cuyo protocolo se identifican los siguientes niveles para establecer dicha comunicación:

- Nivel físico.
- Nivel de comunicación de datos.

- Nivel de comunicación de información.
- Nivel de presentación de información.
- Nivel de interpretación de la información.

En seguida presentamos los diferentes niveles para establecer la comunicación entre el PLC y la Terminal Industrial. El dispositivo intermedio de interfase para la comunicación entre estos dos elementos es un Módulo BASIC 1771-DB de Allen-Bradley. Ver el Apéndice B para mayor información concerniente al Módulo BASIC.

La comunicación entre el PLC y la T-35 es bidireccional y se efectúa de la siguiente forma:

Comunicación T-35 – PLC:

- La T-35 transmite vía puerto serie a Módulo BASIC.
- El Módulo BASIC transmite al PLC mediante transferencia de bloques de datos a través del *bus* del *rack*.

Comunicación PLC – T-35:

- El PLC hace una transferencia de bloques de datos vía el *bus* del *rack* al Módulo BASIC.
- El Módulo BASIC transmite a través de su puerto periférico a la T-35.

El esquema de comunicación anterior se implanta a través de los siguientes niveles:

6.2.1 Nivel físico

Las conexiones entre el puerto serie de la T-35 y el puerto periférico del Módulo BASIC son:

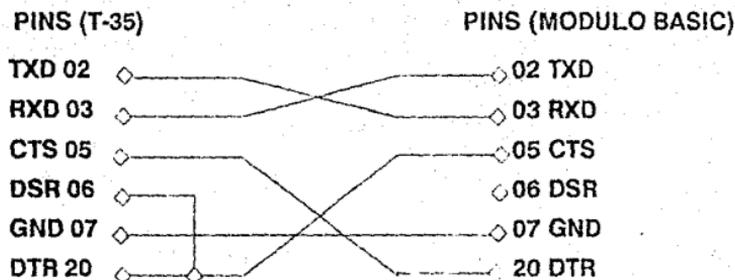


FIGURA 6-1

Con esta conexión se logra que el Módulo BASIC transmita datos únicamente cuando la T-35 se lo permite (cuando no se encuentra atendiendo una interrupción) y que la T-35 transmita cuando la línea de espera de caracteres en el *buffer* de entrada del Módulo BASIC es menor a 223. Además de implantar este *handshaking*, el esquema permite un modo de comunicación *half-duplex*.

6.2.2 Nivel de comunicación de datos.

El formato para transmitir datos es el siguiente:

- 8 bits de datos.
- 1 bit de paro.
- Sin bit de paridad.
- Velocidad de transmisión de 9600 bauds.

Este formato se programa por software en la T-35 y en el Módulo BASIC. En la T-35, se efectúa a través de la función `com_int()` de la cual se anexa un listado al final del capítulo. En el Módulo BASIC, la programación se logra a través de un CALL especial y la velocidad de transmisión se establece con un JUMPER.

6.2.3 Nivel de comunicación de información

El formato de transmisión de la T-35 al Módulo BASIC es el siguiente:

<Índice> <ENTER> <dato> <ENTER>

La T-35 transmite vía las funciones `Tx_car()`, `Tx_v1()` que igualmente se listan al término de éste capítulo. Por su lado, el programa BASIC recibe a través de la rutina 400 del programa BASIC. El programa BASIC completo también está contenido al final de éste capítulo.

El formato de transmisión del módulo BASIC a la T-35 es el siguiente:

0 o más espacios <índice> <espacio> <espacio> <valor> 0 o más espacios

El Módulo BASIC transmite vía la declaración **P.# (PRINT #)**. La T-35 recibe información vía la rutina de atención de interrupción programada cuando la función `lee_com1()` recibe un caracter.

6.2.4 Niveles de presentación e interpretación de la información

Los niveles de presentación e interpretación de la información se presentarán a través de la correspondencia de datos entre las tablas de datos del PLC, el Módulo BASIC y la T-35 contenidas en el apéndice D de este documento.

Alrededor de estas tablas y de las rutinas mencionadas anteriormente gira la construcción del software para la interfase de operador llamada **BIKO**. Los detalles del desarrollo son tema del capítulo siguiente.

6.3 Funciones de Comunicación del Programa C en T-35.

```
-----/
/- Objetivo. Inicializar el puerto de comunicación con los-/
/- valores adecuados para los parámetros de la -/
/- transmisión y recepción de datos. Además instala la -/
/- interrupción. -/
-----/

void com_int(void)
{
    _dos_setvect(0xC, lee_com1); /* Instala la rutina de atención de
                               interrupciones cuando recibe un caracter*/

    /* UART 8250 de la PC a 9600 bauds */
    outp(0x3FB, 0x80);
    outp(0x3F9, 0x0);
    outp(0x3F8, 0xC);
}
```

```
/* UART 8250; 8 bits de datos, 1 bit de paro y sin bit de paridad */
outp(0x3FB,0x3);

/* Señal DTR activa y permitir que las interrupciones sean */
/* transmitidas al 8259 */
outp(0x3FC,0x9);

/* Interrumpir cada que se recibe un caracter */
outp(0x3F9,0x1);

/* Habilita interrupciones del puerto serial en el 8259 */
outp(0x21,inp(0x21)&0xef);
}

/*-----*/
/*--- Objetivo: Núcleo del sistema de interfaz ya que ---*/
/*--- recibe la información en cualquier momento y ---*/
/*--- opera en forma transparente al usuario. ---*/
/*-----*/
void interrupt lee_com1(void)
{
    int c,j;

    static char dato[10]; /* Almacena los caracteres que constituyen un dato. */
    static short i,no_datos; /* i, no. de caracteres en dato */
    static int indice;
    static unsigned minant,alar_ant; /* Minuto y alarma anterior */

    /* Impide la recepción de otro caracter, DTR=0 */
    outp(0x3fc,inp(0x3fc)&0xfe);
}
```

Foreground

```
/* Sincronización para la recepción de datos */  
  
if ( sincron ) {  
    i = 0;  
    no_datos = 0;  
    sincron = 0;  
}  
  
/* Almacena dígitos únicamente */  
if ( isdigit(c=inp(0x3f8)) )  
    dato[i++] = c;  
  
/* Si hay un espacio convierte la cadena en dato a un número */  
Else if ( isspace(c) ) {  
    if( (i9)&&(i=1) ) {  
        no_datos++;  
        switch (no_datos) {  
            case 1 : dato[i] = '\0'; /* El 1er dato es el índice */  
                    indice = atoi(dato);  
                    i = 0;  
                    break;  
            case 2 : dato[i] = '\0'; /* El 2do dato es la información */  
                    i = 0;  
                    break;  
        }  
    }  
  
/* Almacena información válida */  
if ( (indice=1) || (indice=64) ) {  
    tabla_datos[indice-1] = (unsigned) atoi(dato);  
    no_datos = 0;  
    i = 0;  
}  
}
```

```
/* Registro de alarmas: tipo y hora */
if ( tabla_datos[46] != alar_ant ) {
    if ( ind_alar = 100 )
        ind_alar = 0;
    alarmas[ind_alar].rec = 0;
    alarmas[ind_alar].tip = tabla_datos[46];
    alarmas[ind_alar].dia = tabla_datos[47];
    alarmas[ind_alar].mes = tabla_datos[48];
    alarmas[ind_alar].ano = tabla_datos[49];
    alarmas[ind_alar].hor = tabla_datos[50];
    alarmas[ind_alar].min = tabla_datos[51];
    alar_ant = tabla_datos[46];
    ind_alar++;
}
/* Registra informaci3n sobre temperatura */
/* cada minuto. */
if ( minant != tabla_datos[51] ) {
    if ( itfps = 580 ) {
        itfps = 0;
        for ( j = 0 ; j < 6 ; j++ )
            fyhref[j] = tabla_datos[47+j];
        for ( j = 0 ; j < 5 ; j++ ) {
            isps[j] = 1;
            (sps[j][0]).hor = tabla_datos[50];
            (sps[j][0]).min = tabla_datos[51];
            (sps[j][0]).sp = fpls[j].sp;
        }
    }
}
```

```

    }

}

for ( j = 0 ; j < 5 ; j++ ) {
    (pvs[j][itfps]).pv = tabla_datos[32+j];
    (pvs[j][itfps]).hor = tabla_datos[50];
    (pvs[j][itfps]).min = tabla_datos[51];
}

minant = tabla_datos[51];
itfps++;
}

break;
default: break;
}
}

/* Habilita la recepción de datos, DTR=1 */
outp(0x3fc,inp(0x3fc)|0x1);

/* Indica al 8259 que la interrupción ha sido atendida */
outp(0x20,0x20);
}

```

```
/*-----*/  
/*--- Objetivo: Transmitir un caracter al módulo BASIC. ---*/  
/*-----*/  
void Tx_car(c)  
char c;  
{  
short i;  
for (i=0;i<500;i++); /* Tiempo de espera para transmitir caracteres */  
    outp(0x3f8,c); /* Transmite el caracter */  
}  
/*-----*/  
/*--- Objetivo: Transmitir un comando al PLC vía BASIC. ---*/  
/*-----*/  
void Tx_vl(tip,ind,val)  
char tip;  
char *ind;  
char *val;  
{  
Tx_car(tip); /* Transmite identificación de transacción */  
do { Tx_car(*ind++); } while (*ind); /* Transmite el índice */  
Tx_car(0xd); /* Transmite ENTER */  
do { Tx_car(*val++); } while (*val); /* Transmite el valor */  
Tx_car(0xd); /* Transmite ENTER */  
}
```

6.4 Listado del Programa BASIC de Interfase PLC - T-35.

* Declara 20 bytes para usar cadenas de 8 caracteres

10 STRING 20,8

* TX() arreglo que contiene los datos del PLC hacia la
T-35

* TR() arreglo de referencia para el arreglo TX

20 DIM TX(64),TR(64)

* FT gasto de gas del horno de temple

* FR gasto de gas del horno de revenido

* TA variable de referencia para FT

* RA variable de referencia para FR

22 FT=0 : FR=0 : TA=0 : RA=0

* Ejecutar la rutina de interrupción en la línea 600 cada
10 minutos (600 segundos)

24 TIME=0 : CLOCK 1 : ONTIME 600,600

* Borra el contenido de los buffers asociados al puerto
periférico del Módulo

30 PUSH 2 : CALL 37

* Define el número de palabras a transferir entre el
Módulo BASIC y el PLC (64)

40 PUSH 64 : CALL 4

50 PUSH 64 : CALL 5

* Transfiere los datos del PLC al Módulo BASIC

60 CALL 3 : POP SR

* Convierte los datos binarios del PLC a números que se
almacenan en TX()

70 FOR I= 1 TO 47

80 PUSH I : CALL 11 : POP TX(I)

90 NEXT I

* Transfiere los datos recibidos por el Módulo BASIC desde
la T-35 al PLC

100 CALL 2 : POP ST

* X=0; No transmitir información a la T-35

* X0; Transmitir información a la T-35

* El valor de X es fijado por la T-35

110 IF X=0 THEN 160

* Determina fecha para transmitir a T-35

111 CALL 44

112 POP TX(48),TX(49),TX(50)

* Determina hora para transmitir a T-35

113 CALL 46

114 POP TX(51),TX(52),TX(53)

* Determina el número de caracteres recibidos desde T-35 y
almacena el valor en LI

115 PUSH 0 : CALL 36 : POP LI

* push 0 : call 37 borra el *buffer* de entrada del Módulo

BASIC

116 IF LI200 THEN PUSH 0 : CALL 37

* Envía información a T-35...

120 FOR I=1 TO 52

```
130 IF TX(I)TR(I).AND.I45.AND.I46 THEN PRINT
    #I,TX(I)
140 TR(I)=TX(I)
150 NEXT I
152 IF TAFT THEN PRINT #45,FT : TA = FT
154 IF RAFR THEN PRINT #46,FR : RA = FR
```

* Lee el caracter enviado desde la T-35

```
160 CALL 35 : POP C
```

* Recibe datos si el caracter es 'T'

```
170 IF C84 THEN 220
180 GOSUB 400
182 IF IN39 THEN 200
184 PUSH 16384 : PUSH 4 : CALL 21
186 PUSH 0 : PUSH 7 : CALL 21
200 PUSH VA : PUSH IN : CALL 21
210 GOTO 60
```

* Retransmite datos a T-35 cuando el caracter es 'I'

```
220 IF C73 THEN 290
```

```
230 X = 1
240 FOR I = 1 TO 52
250 TR(I) = 0
260 NEXT I
270 PUSH 0 : CALL 37
280 GOTO 60
```

* Suspende transmisión a T-35 cuando el caracter es 'F'

```
290 IF C70 THEN 60
300 X = 0
310 GOTO 60
```

* Rutina de recepción y conversión de datos transmitidos
por la T-35 hacia el Módulo BASIC.

\$(0); almacena el valor del índice

\$(1); almacena el valor asociado al índice

```
400 J = 1
410 DO
420 CALL 35 : POP C

430 ASC$(0),J) = C
440 J = J + 1
```

```
450 UNTIL C = 13
460 J = 1
470 DO
480 CALL 35 : POP C
490 ASC($ (1), J) = C
500 J = J + 1
510 UNTIL C = 13
520 PUSH 0 : CALL 63
530 POP V : POP IN
540 PUSH 1 : CALL 63
550 POP V : POP VA
560 RETURN
```

* Rutina de interrupción para cálculo del gasto cada 10 minutos.

```
600 TIME = 0 : ONTIME 600,600
610 FT = FT + INT(11.15*SQR(TX(45)))
620 FR = FR + INT(11.15*SQR(TX(46)))
650 RETI
```

7. Desarrollo del Sistema Biko

7 Desarrollo del Sistema BIKO.

7.1 Metodologías de Desarrollo de Sistemas.

Con la finalidad de guiar el desarrollo de un proyecto de *software*, existen algunas metodologías que, bajo ciertas circunstancias, no siempre son aplicables en su totalidad, puesto que éstas son de propósito general. Dependiendo de la naturaleza de cada proyecto se debe elegir una determinada metodología o adecuar cualquiera de las ya existentes. Lo importante de estas metodologías no es seguirlas estrictamente, sino encontrar un apoyo para el diseño y desarrollo de proyectos de *software*.

Estas metodologías identifican una secuencia general de pasos dentro del proceso de desarrollo:

- 1) Especificación y Análisis de requerimientos.
- 2) Evaluación de Alternativas.
- 3) Diseño.
- 4) Implantación.

En el primer paso se identifican las necesidades y problemas por resolver, así como las posibles alternativas de solución.

En el segundo paso se analizan las alternativas de solución identificadas y en base a los criterios de evaluación pertinentes, se selecciona la más adecuada.

En el paso de diseño, se define la arquitectura del sistema y en el último paso se efectúa la codificación y pruebas del *software* así como procesos de documentación y entrenamiento a usuarios finales.

La metodología propuesta, en muchas ocasiones, no concuerda con la práctica. Las razones de esto, pueden ser variadas según el caso del sistema a desarrollar. Particularmente, nuestro proyecto de tesis se vio

envuelto por una serie de circunstancias que impidieron seguir la metodología. Tales circunstancias se pueden resumir a lo siguiente:

1. Falta de definición detallada en las especificaciones del proyecto. Es labor del diseñador de un proyecto obtener una definición detallada y completa antes de comenzar el desarrollo, pero este no fué el caso. El equipo de diseño en labor conjunta con los usuarios llegamos a conformar las primeras bases del sistema requerido, pero la planta atravesaba por una etapa de reconstrucción física que inequívocamente sería factor para futuros cambios.
2. Susceptibilidad a cambios imprevistos. Esta característica está entrelazada con la anterior, pues a medida que avanzaba la reconstrucción de la planta, ocasionalmente las circunstancias obligaban modificaciones sobre el software ya desarrollado.
3. Nuevos requerimientos. Además de surgir cambios al software ya desarrollado, se presentaban nuevas necesidades en el sistema conforme seguía avanzando la reconstrucción de la planta.
4. Pruebas modulares sobre la marcha. Una vez más, el proceso de reconstrucción adquiere importancia pues era necesario efectuar pruebas por etapas conforme iban siendo desarrolladas. Sólo si éstas arrojaban resultados favorables se podía avanzar a las etapas subsecuentes.
5. El factor tiempo. Prácticamente todo el proceso de diseño y desarrollo estuvo bajo presiones de tiempo como suele suceder en los proyectos reales.

Todos estos factores condujeron a centrar la atención sobre la arquitectura en vez de seguir una metodología. La idea básica de dicha arquitectura se apoya fuertemente en un diseño modular, capaz de crecer conforme los requerimientos lo indiquen. Lo que se hizo fué bosquejar una arquitectura básica, a partir de la cual se modularizó el sistema según una organización lógica y se comenzaron a desarrollar las primeras partes. Conforme seguía avanzando el proyecto surgían nuevos requerimientos que se iban agregando al rompecabezas. Generalmente cada uno de esos requerimientos era analizado al momento para determinar su factibilidad de realización, pues el usuario necesitaba prontas respuestas.

Mientras se efectuaba el desarrollo nacían nuevas características en el sistema, mismas que se atacaban a la par. En otras palabras, seguimos un proceso iterativo de desarrollo, continuando con el esquema de la arquitectura inicial y adecuando los nuevos requerimientos. El diagrama siguiente ayudará a entender esta idea.

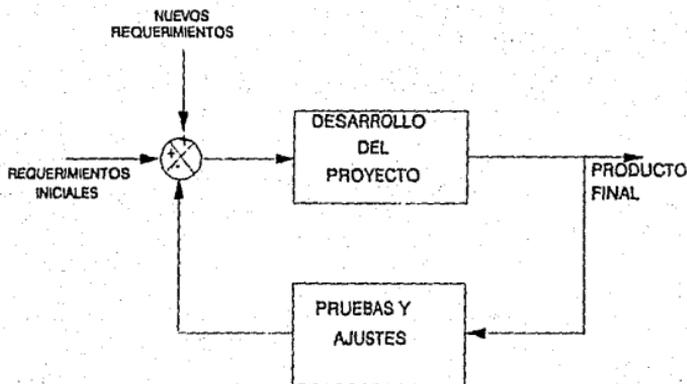


FIGURA 7-1

La modularidad planteada desde el comienzo del proyecto resultó ser clave y de gran beneficio, ya que de no haber sido así, el citado método iterativo habría acarreado enormidad de problemas.

7.2 Fase Inicial del Desarrollo del Sistema BIKO.

En la etapa inicial del sistema, recabamos la información mínima necesaria en base a documentos escritos e información verbal para bosquejar un esquema base. Esta búsqueda inicial pretendía identificar las características más importantes a partir de las cuales estructuramos una primera arquitectura. Estas características serán expuestas más adelante.

Las características de esa arquitectura dieron pie a seleccionar las herramientas de *software* apropiadas para una aplicación de tal naturaleza. La selección del lenguaje de programación es un punto de relevante importancia. Un lenguaje de alto nivel con capacidad de programar interfases gráficas era la elección idónea. Las posibles opciones nos inclinaron a pensar en lenguajes del tipo de C o PASCAL, que son estructurados y de los cuales existen diversos compiladores y utilerías en el mercado. De nuestras experiencias y predilecciones personales optamos por emplear el Lenguaje C. Además C ofrece características favorables para desarrollos de este tipo que requieren facilidades de acceso directo al *hardware*.

En cuanto a la disponibilidad de recursos, aspecto siempre importante para el tiempo de desarrollo, se contó en todo momento con tiempo de máquina durante los horarios asignados para desarrollo. Es muy difícil calcular un estimado del tiempo de desarrollo de la primer versión probada en la planta industrial, no obstante, el tiempo de desarrollo aproximado resultó ser de 6 meses invirtiendo un promedio de 12 horas-hombre diarias repartidas en un equipo de 3 personas.

7.3 Características del Sistema BIKO.

La arquitectura base fué presentada repetidas ocasiones a los usuarios con fines de retroalimentar dicho esquema (método iterativo). Finalmente llegamos a definir un conjunto completo de características requeridas:

1. Ambiente amigable de operación.
 - 1.1 Manejo de menús.
 - 1.2 Ayudas.
 - 1.3 Validación de datos.
 - 1.4 Guía dinámica y continua de operación y captura.
 - 1.5 Mensajes de status de operación.
 - 1.6 Cajas de diálogo.
 - 1.7 Retroalimentación al usuario.
2. Representación gráfica en tiempo real del proceso.
 - 2.1 *Faceplates* (Control de lazo cerrado).
 - 2.2 Mímicos (representación de la ubicación física de los elementos del proceso).
 - 2.3 Gráficas valor de variable física vs. tiempo.
3. Modalidades de Control del proceso.
 - 3.1 Manual.
 - 3.2 Automática.
4. Estadísticas.

Un análisis de las características a cubrir por el sistema BIKO nos llevó al diseño de un modelo a bloques de los principales módulos que conformarían el sistema. La finalidad buscada en este modelo era identificar los puntos medulares a partir de los cuales se cimentaría el desarrollo. El diagrama de este modelo se muestra en la figura 7-2.

Un paso importante para llegar a este modelo fué la debida identificación de las entradas y salidas del sistema. La figura 7-2 muestra las entradas en la parte superior y las salidas en la parte inferior de la hoja.

Las entradas se subdividen en dos módulos fundamentalmente:

- Entrada de comandos del operador a través del teclado.
- Entrada de datos del proceso industrial vía el puerto serie.

La información adquirida por el bloque de módulos de entrada es procesada por un segundo bloque de módulos (indicado en la parte media de la figura). Dentro de este bloque están catalogados los módulos de:

- Validación de comandos.
- Interpretación lógica de comandos.
- Interpretación física del proceso.
- Cálculo de estadísticas.

El bloque de módulos encargado de generar las salidas agrupa:

- Despliegue y bitácora de alarmas.
- Representación gráfica del proceso.
- Despliegue de gráficas variable-tiempo y estadísticas.
- Ayudas y retroalimentación al operador.

Cada módulo, a su vez, está formado por un conjunto de funciones. No llegaremos al detalle de estas pero explicaremos brevemente cada uno de los módulos del modelo.

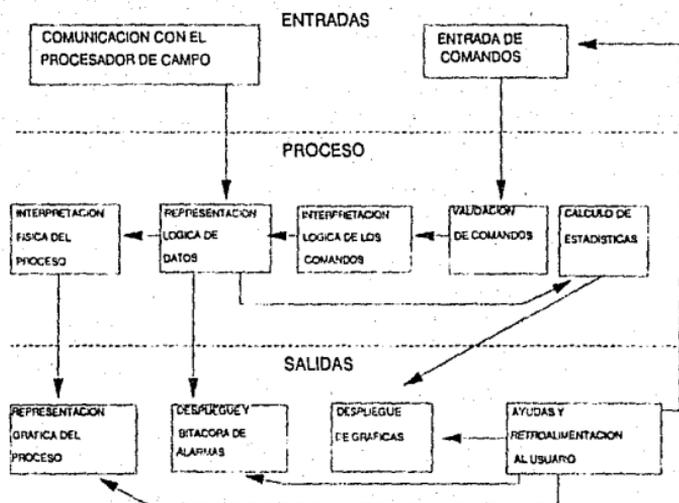


FIGURA 7-2

7.4 Descripción Semidetallada de los Módulos.

Ya identificamos dos módulos que conforman las entradas al sistema BIKO. Por su parte, las entradas vía el puerto serie, son un conjunto de datos provenientes del procesador de comunicaciones conectado al PLC (Módulo BASIC). El módulo de comunicaciones encargado de esta tarea es el canal de acceso a la información de campo, es decir, del proceso industrial. La rutina de comunicación es invocada automáticamente cuando el puerto serie recibe un carácter, manteniendo un enlace permanente con el campo.

Por otro lado, las entradas por medio del teclado son un conjunto de comandos que el operador ordena según sus necesidades. Los comandos de entrada van desde instrucciones específicas de Supervisión hasta activación de botones que ejecutan acciones de Control sobre el proceso. Entonces, el módulo de entrada de comandos se encarga del manejo del teclado bajo un ambiente amigable y de fácil uso para el operador.

Para conservar la seguridad del sistema los comandos deben ser validados. De esto se encarga el módulo de validación de comandos. El operador será retroalimentado con mensajes de ayuda en todo momento antes de efectuar cualquier instrucción de Control para prevenir daños sobre el equipo o personal de la planta.

El módulo de interpretación lógica de los comandos, es una agrupación de funciones encargada de traducir la información de entrada, a una serie de instrucciones que llevarán a cabo la petición del operador.

El módulo de cálculo de estadísticas efectúa operaciones aritméticas sobre los datos adquiridos del campo para obtener cifras de relevante importancia para los usuarios. El módulo de despliegue de gráficas toma parte de sus entradas del módulo anterior y el resto de los datos desplegados provienen directamente del campo. Dicho despliegue contempla gráficas de la variación con respecto al tiempo de las variables de interés del proceso y datos estadísticos de alta precisión.

La interpretación física del proceso es un módulo cuya función radica en determinar el estado físico del proceso industrial en todo momento, empleando la información de campo. Este módulo realiza una función similar al SCAN del PLC, pues continuamente efectúa un barrido cíclico sobre los datos detectando cualquier cambio de estado.

Intimamente ligado con el módulo anterior se encuentra el módulo de representación gráfica del proceso. Este se encarga de representar mediante mímicos los diferentes estados del proceso. Este conjunto de funciones dependen de las invocaciones del módulo de interpretación física del proceso.

Era imprescindible un módulo encargado de advertir al operador de situaciones delicadas en mayor o menor grado que puedan llegar a representar peligros extremos: despliegue y bitácora de alarmas. Además del despliegue visual, este módulo lleva un registro en disco de las últimas alarmas presentadas durante el proceso.

Un último módulo, ayuda y retroalimentación al usuario tiene la función, como su nombre lo indica, de auxiliar al operador con mensajes de apoyo para el manejo del sistema, pues no debemos olvidar que BIKO es una interfase gráfica para Supervisión y Control y el aspecto interactivo es de suma importancia.

Al centro de toda esta arquitectura de módulos fuente, se encuentra un vector de datos que alimenta a todo el sistema: `tabla_datos[]`. Podemos recalcar que dicho conjunto de datos ocupa tan sólo 64 palabras de la memoria principal de la computadora. El vector `tabla_datos[]` junto con las funciones que lo actualizan son el punto medular del modelo.

El modelo anterior describe las partes que conforman al sistema BIKO desde un panorama muy general y al mismo tiempo proporciona información sobre el flujo de datos en el mismo.

Es una idea conceptual del sistema que sirvió como herramienta para iniciar el desarrollo de código fuente, es decir, la implantación, y sí bien, el modelo parece sencillo a primera instancia, la realización consumió una extensión considerable de programas fuente codificados en lenguaje C.

7.5 Documentación Técnica

Las características que el sistema BIKO requiere para su ejecución se listan en seguida:

- Una computadora personal 100% compatible con la IBM-PC/AT
- 512 KB de memoria RAM
- Monitor color
- Tarjeta gráfica EGA
- Disco duro 10 MB
- Puerto RS-232C
- Sistema operativo MS-DOS 3.2 o versiones subsecuentes

Debe considerarse que el ambiente industrial bajo el cual operan este tipo de sistemas es un ambiente hostil (corrosión, polvo, altas temperaturas, vibraciones, etc) por lo que se recomienda ampliamente el empleo de una computadora de tipo industrial.

Manual de Usuario

MANUAL DE USUARIO

M.1 Ejecución.

Para ejecutar el sistema de interfase para Supervisión y Control: **BIKO**, basta con restablecer la computadora T-35, con lo cual **BIKO** se cargará automáticamente pues éste se encuentra invocado desde el programa **AUTOEXEC**, o simplemente se puede invocar como cualquier programa ejecutable desde **DOS** si ya se encuentra encendida la computadora. El usuario deberá esperar la pantalla principal de **BIKO**, desde la cual se puede acceder cualesquiera de los servicios que ofrece el sistema. Esta pantalla es mostrada en la figura M-1. Para utilizar los diferentes servicios de **BIKO** se cuenta con un menú del tipo *pull-down* que tiene la característica de manipularse muy fácilmente mediante las teclas de dirección (flechas) y la tecla **<ENTER>**.

FORJAMEX	
SISTEMA DE SUPERVISION Y CONTROL:	HORNOS DE TEMPLE Y REVENIDO

BIKO	
SUP. ANALOG MONITOREO PAÑO ENCENDIDO CONF. AUTOMATICA Y LOTES	

Biko: * ALG.
* J ² GM.
* APH
Seleccione con flechas y presione la tecla de <ENTER>

FIGURA M-1

El menú principal de **BIKO** cuenta con cinco opciones básicas, cada una de las cuales tiene asociado un submenú que se activa presionando <ENTER> y se desactiva mediante la tecla <ESC>. La opción seleccionada se encontrará sombreada de tal manera que sea fácilmente identificable. Así pues, para utilizar una de las partes de **BIKO** solo tenemos que posicionarnos mediante las flechas en la opción deseada y presionar la tecla <ENTER>.

La figura M-2 muestra un ejemplo de un submenú.

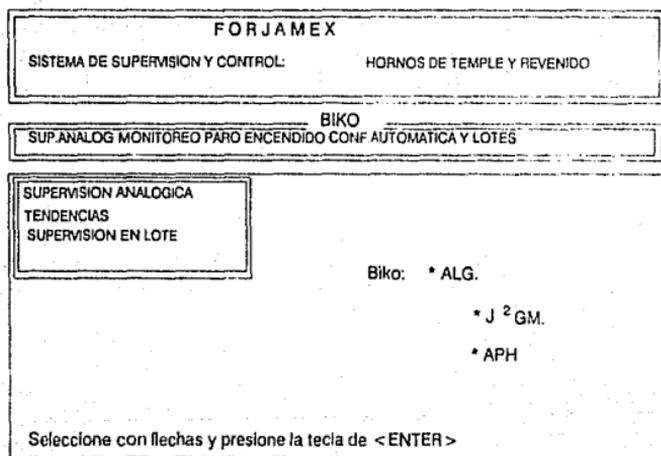


FIGURA M-2

Las opciones del menú principal y submenús de **BIKO** son las siguientes:

1. Supervisión analógica.
 - 1.1 Supervisión analógica.
 - 1.2 Tendencias.
 - 1.3 Supervisión en lote.
2. Monitoreo.
 - 2.1 Temple.
 - 2.2 Cesta.
 - 2.3 Quemadores.
 - 2.4 Revenido.
3. Paro.
 - 3.1 Paro temple.
 - 3.2 Paro revenido.
4. Encendido.
 - 4.1 Arranque temple.
 - 4.2 Arranque revenido.
5. Configuración automática y lotes.
 - 5.1 Configuración automática temple.
 - 5.2 Configuración automática revenido.

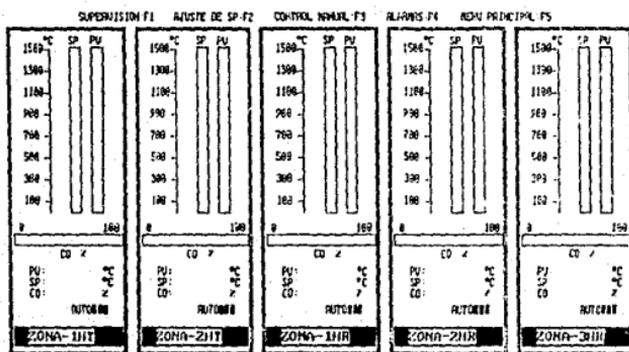
5.3 Carga lote.

5.4 Tendencias lote.

M.2 SUPERVISION ANALOGICA.

M.2.1 SUPERVISION ANALOGICA.

Mediante esta opción se tiene acceso al monitoreo y Control de *faceplates* para las distintas zonas definidas en ambos hornos, tanto temple como revenido. Al acceder esta opción se presenta la pantalla mostrada en la figura M-3.



OPCIÓN F5 PARA SELECCIONAR LA OPCIÓN RECIBIDA Y DESPUÉS OPRIR EL BOTÓN

FIGURA M-3

desde la cual tenemos un menú con cinco opciones:

- F1.- Permite el monitoreo de las cinco zonas distribuidas en ambos hornos, mostrando los valores de *Set Point (SP)*: Temperatura en grados centígrados, *Variable de Proceso (PV)*: Temperatura en grados centígrados y *Valor de la Salida del lazo de Control en porcentaje (CO)*: Porcentaje de apertura de la válvula que controla el flujo de aire a los quemadores. El despliegue es digital y gráfico.
- F2.- Permite ajustar el valor del *Set Point* al valor deseado por el usuario, ya sea capturando directamente el dato o incrementándolo a través de las flechas del teclado.
- F3.- Es la opción equivalente a la anterior para el Valor de la Salida del lazo de Control en porcentaje.
- F4.- Alarmas. Esta opción lleva al usuario a un submenú con las siguientes opciones:
 - *F1.- Reconocimiento de alarmas. Permite al operador indicarle a **BIKO** que ya tomó las acciones necesarias, para ajustar los parámetros de operación definidos que originaron la alarma indicada.
 - *F2.- Últimas alarmas. Presenta al usuario las últimas 20 alarmas ocurridas en la operación del sistema.
 - *F3.- Archivo de alarmas. Mediante esta opción se presenta al usuario el archivo que contiene todas las alarmas ocurridas desde la operación del horno.
 - *F4.- Ajuste de límites. Permite establecer los límites superior e inferior de alarma para cada zona.
- F5.- Es la salida al menú principal.

M.2.2 TENDENCIAS.

Muestra una gráfica con las tendencias de la temperatura vs. tiempo en cada una de las cinco zonas definidas. Al acceder esta opción lo primero que aparece es una pantalla de captura donde se solicita el número de la zona que se desea observar, esta pantalla se muestra en la figura M-4.

The screenshot displays a graphical user interface for the FORJAMEX system. At the top, a header box contains the text "FORJAMEX" and "SISTEMA DE SUPERVISION Y CONTROL: HORNOS DE TEMPLE Y REVENIDO". Below this, a central box prompts the user to "INTRODUZCA EL NUMERO DE FACEPLATE..." with the number "1" entered. To the right of the input box, the text "Biko: * ALG." is visible. Below the input area, there are two options: "* J 2 GM." and "* APH". At the bottom of the screen, a instruction reads "Seleccione con flechas y presione la tecla de <ENTER>".

FIGURA M-4

Después de capturarse el número deseado se tiene acceso a la pantalla de tendencias, la cual se muestra en la figura M-5.

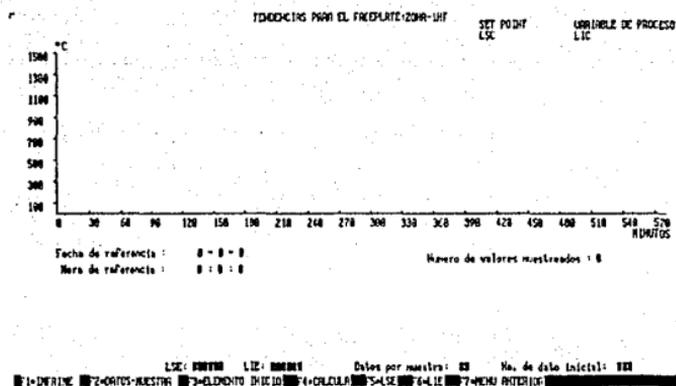


FIGURA M-5

Al entrar a esta pantalla el usuario tiene una visión completa de las tendencias, ya que tiene en una gráfica el comportamiento del *set point*, de la variable de proceso y del límite superior e inferior de Control. Cada gráfica es representada con un color diferente.

Dentro de esta pantalla se tiene un menú con las siguientes opciones:

- F1.- Imprime. Permite al usuario obtener una impresión de la gráfica de tendencias.
- F2.- Datos-muestra. Es el número de datos por muestra para realizar un cálculo estadístico.
- F3.- Elemento inicio. Indica el número del elemento a partir del cual se desea efectuar la gráfica.
- F4.- Calcula. Obtiene el cálculo de algunos valores estadísticos en base a los datos de temperatura y tiempo almacenados.
- F5.- LSE. Define el límite superior especificado.
- F6.- LIE. Define el límite inferior especificado.
- F7.- Menú anterior. Es la salida al menú principal.

M.2.3 SUPERVISION EN LOTE.

Esta opción permite al usuario tener una visión general de lo que ocurre en todo el proceso. Al seleccionar este servicio se muestra cíclicamente cada una de las pantallas de Supervisión durante un período de 30 segundos, de forma tal que, sin necesidad de activar ninguna otra tecla, el usuario monitorea todo lo que ocurre en cada una de las diferentes partes del proceso.

Es conveniente aclarar que durante el tiempo en que una pantalla permanece activa (30 seg) el usuario no puede llevar a cabo ninguna acción de Control, ya que esta opción solo permite monitorear las diferentes partes del proceso. Si el usuario desea llevar a cabo cualquier

acción de Control debe salir de esta opción presionando la tecla <ESC> y acceder la opción adecuada del menú principal.

M.3 MONITOREO.

M.3.1 TEMPLE.

Este servicio permite supervisar y controlar todos los movimientos mecánicos que se realizan en la zona del horno de temple. La pantalla de esta opción se presenta en la figura M-6.

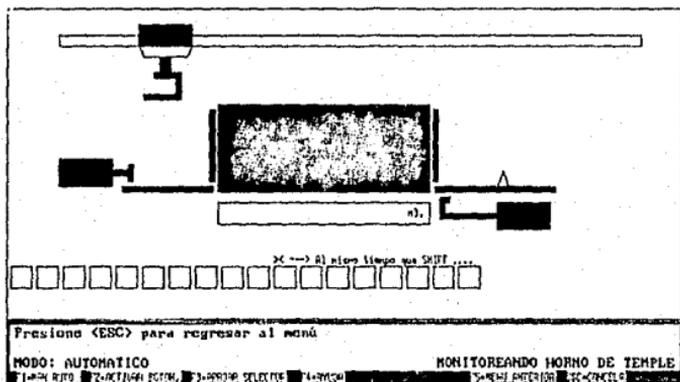


FIGURA M-6

Por default este servicio se encuentra siempre monitoreando lo que sucede en la zona del horno de temple, representándolo en la pantalla, de tal forma que el usuario conoce en todo momento el estado de todos los elementos del horno.

Además del monitoreo que se lleva a cabo dentro de este servicio, el usuario puede realizar algunas otras acciones más, las cuales son accesibles mediante un menú que se encuentra en la parte inferior de la pantalla. Para acceder a este menú, puesto que **BIKO** se encuentra monitoreando el horno en ese momento, debe presionarse primero la tecla de <ESC> para abandonar temporalmente el monitoreo y después presionar la tecla de la función deseada.

Las opciones posibles dentro del monitoreo del horno de temple son:

- **F1 - MAN AUTO.** Permite al usuario cambiar el modo en el que se encuentra operando el horno, ya sea Manual o Automático. En secuencia automática los movimientos son gobernados por la lógica de Control permitiéndole al usuario únicamente el monitoreo de los mismos, mientras que en secuencia Manual los movimientos son controlados por el usuario, permitiéndosele a éste el Control mediante botones simulados en la pantalla.
- **F2 - ACTIVAR BOTON.** Cuando el sistema se encuentra operando en modo Manual, mediante este menú el usuario puede activar cualquiera de los diferentes botones que controlan el movimiento de los elementos mecánicos. Estos botones pueden ser de dos tipos: *push-buttons* o selectores. Los botones que son equiparables con *push-buttons* permanecerán activos hasta que se presione <ESC> para "soltarlos", mientras que aquellos que son selectores permanecerán en su último estado como sucede con los selectores reales.
- **F3 - APAGAR SELECTOR.** Permite al usuario apagar algún selector que haya activado mediante la tecla F2.
- **F4 - AYUDA.** Muestra al usuario una pantalla de ayuda, la cual le indica los diferentes botones que están habilitados para cada sección así como el elemento físico que activan.
- **F5 - MENU ANTERIOR.** Regresa al usuario al menú principal de **BIKO**.

- **ESC - CANCELA.** Tiene dos funciones básicas, la primera es avisarle al sistema BIKO, cuando éste se encuentra monitoreando, que el usuario desea utilizar cualquiera de los servicios del menú. La segunda función es de cancelación de cualquier dato o acción erróneos. En algunos casos, como en activar botón, tiene alguna utilidad especial pero cualquiera que esta sea se le indica al usuario mediante un mensaje en la pantalla.

M.3.2 CESTA.

Esta opción y la opción de monitoreo del horno de revenido (2.4) son prácticamente idénticas a la anterior, pero cada una monitoreando su zona del proceso correspondiente. Así pues, tomando en cuenta que la forma de operarlos es muy similar sólo se detallarán las pequeñas diferencias.

La figura M-7 muestra la pantalla de monitoreo de la zona de cesta, en esta se puede observar que el menú es prácticamente igual al del horno de temple.

Obviamente los botones activan elementos diferentes y la pantalla de ayuda es también diferente, pero la forma de operarlo es la misma que para el monitoreo del horno de temple.

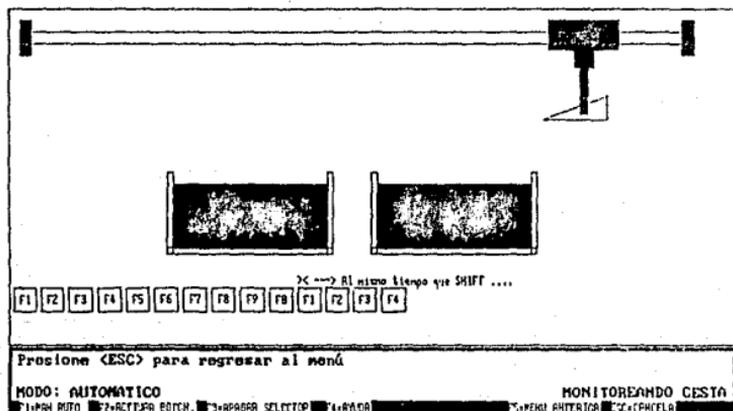


FIGURA M-7

M.3.3 QUEMADORES.

Esta opción presenta al usuario una pantalla en la cual se representan cada uno de los quemadores que se encuentran en ambos hornos representados mediante lámparas simuladas. Estas lámparas están distribuidas en la pantalla de acuerdo a las zonas definidas en los hornos de temple y revenido para que al usuario le sea fácil asociar a cada lámpara con la posición física del quemador al que representa.

En esta pantalla los colores son importantes. El usuario puede conocer el estado de los quemadores mediante su color, esto es, cuando una lámpara (quemador) se encuentra de color verde indica que el quemador asociado se encuentra encendido y trabajando sin problemas, mientras que si éste se encuentra en color rojo significa que el quemador se encuentra apagado por alguna causa.

Esta pantalla es de mucha utilidad pues mediante ella, el usuario puede determinar si todos los quemadores de ambos hornos están trabajando correctamente o si existe alguno de ellos que sea necesario revisar.

En esta opción las únicas opciones habilitadas son el <ESC> que sirve para interrumpir el monitoreo y accesar el menú y la tecla <F5> que regresa al usuario al menú principal.

La figura M-8 muestra la pantalla de monitoreo de quemadores.

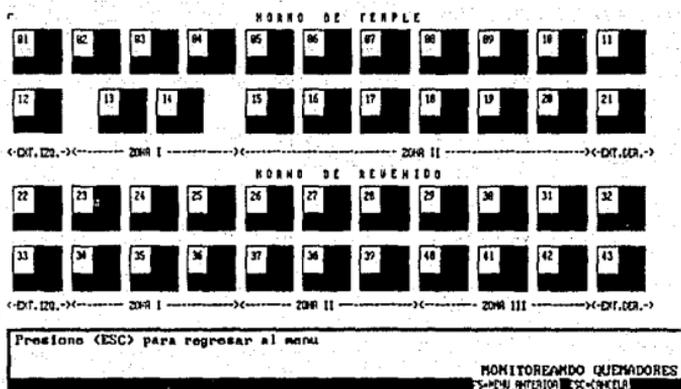


FIGURA M-8

M.3.4 REVENIDO.

Esta opción funciona de la misma manera que el monitoreo del horno de temple, la única diferencia entre estas dos opciones es la zona que monitorean y controlan. La figura M-9 muestra la pantalla de monitoreo del horno de revenido.

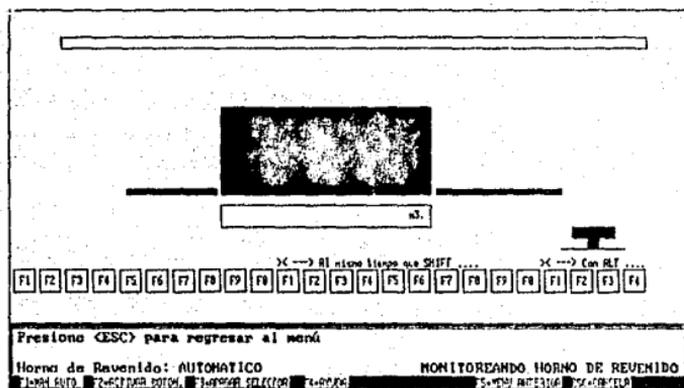


FIGURA M-9

M.4. PARO

M.4.1 PARO TEMPLE.

Esta opción se accesa directamente desde el menú principal, su función es apagar todos los quemadores del horno de temple junto con sus dispositivos asociados.

M.4.2 PARO REVENIDO.

Esta tiene la misma finalidad que la anterior pero para el horno de revenido, esto es, su función es apagar todos los quemadores del horno de revenido junto con sus dispositivos asociados.

M.5 ARRANQUE

M.5.1 ARRANQUE TEMPLE.

Las dos opciones de arranque son el complemento de las opciones descritas en el punto M.4 y como es evidente, su función es encender todos los quemadores del horno de temple. No existe ningún submenú dentro de esta opción.

M.5.2 ARRANQUE REVENIDO.

A través de esta opción el usuario puede arrancar todos los quemadores dentro del horno de revenido.

M.6 CONFIGURACION AUTOMATICA Y LOTES.

M.6.1 CONFIGURACION AUTOMATICA TEMPLE.

Por medio de esta opción el usuario puede determinar las características del ciclo del horno de temple a operar. En la zona de la cesta encontramos dos tinajas las cuales contienen agua y aceite respectivamente. El tratamiento para cada conjunto de piezas es diferente. Por ejemplo puede ser que en algunos casos no sea necesario que las piezas pasen por el tanque de aceite y permanezcan más tiempo dentro del tanque de agua. Todas estas indicaciones pueden llevarse a cabo aquí. Al entrar a esta opción aparece la pantalla de captura mostrada en la figura M-10.

FORJAMEX	
SISTEMA DE SUPERVISION Y CONTROL:	HORNO DE TEMPLE Y REVENIDO

PERMANENCIA EN TANQUE 1 (SEG.) :	
PERMANENCIA EN TANQUE 2 (SEG.) :	
TIEMPO ENTRE CICLOS (SEG.) :	
CICLO AUTOMATICO (1,2,3 o 4) :	
1:NO TANQUES 2:TANQUE 1 3:TANQUE 2 4:AMBOS TANQUES	

Seleccione con flechas y presione la tecla de <ENTER >

FIGURA M-10

Dentro de esta pantalla se capturan cuatro datos:

- Permanencia en tanque 1. Este es el tiempo (en segundos) que la cesta va a permanecer dentro del tanque de agua.
- Permanencia en tanque 2. Este es el tiempo (en segundos) que la cesta va a permanecer dentro del tanque de aceite.
- Tiempo entre ciclos. Es el tiempo en segundos que el controlador espera entre la última operación del ciclo anterior y la primera operación del siguiente ciclo.
- Ciclo Automático. Se presentan cuatro diferentes posibilidades: 1 si se desea que la cesta no pase por ningún tanque; 2 si se desea que la cesta pase sólo por el tanque de agua; 3 si se desea que la cesta pase sólo por el tanque de aceite y 4 si se desea que la cesta pase por ambos tanques.

M.6.2 CONFIGURACION AUTOMATICA REVENIDO.

Esta opción realiza la misma función que la anterior, con una restricción: en esta solamente se configura el tiempo entre ciclos para el horno de revenido. Esta pantalla se muestra en la figura M-11.

FORJAMEX	
SISTEMA DE SUPERVISION Y CONTROL:	HORNOS DE TEMPLE Y REVENIDO

TIEMPO ENTRE CICLOS (SEG.) :

Blko: * ALG. * J 2 GM. * APH

Seleccione con flechas y presione la tecla de <ENTER >

FIGURA M-11

M.6.3 CARGA LOTE.

Le permite al usuario capturar un conjunto de datos pertenecientes a la dureza que un conjunto de piezas haya adquirido durante el tratamiento

térmico, con el fin de obtener estadísticas y gráficas a partir de ellos. La pantalla se muestra en la figura M-12.

FORJAMEX	
SISTEMA DE SUPERVISION Y CONTROL:	HORNO DE TEMPLE Y REVENIDO

PROPORCIONE LAS DUREZAS (1.0 - 7.0):
PARA TERMINAR OPRIMA <ESC>. PARA CAMBIAR LOTE < + >

Biko: * ALG.
* J ² GM.
* APH

Seleccione con flechas y presione la tecla de <ENTER>

FIGURA M-12

Dentro de esta opción se puede capturar el conjunto de datos (durezas) para el lote actual o bien mediante la tecla de < + > cambiar de lote. Al oprimir esta tecla aparece la pantalla mostrada en la figura M-13.

FORJAMEX	
SISTEMA DE SUPERVISION Y CONTROL:	HORNO DE TEMPLE Y REVENIDO
IDENTIFICACION DEL LOTE (NUM):	
TIPO DE PIEZAS DEL LOTE (NUM):	
Biko: * ALG.	
* J 2 GM.	
* APH	
Seleccione con flechas y presione la tecla de <ENTER>	

FIGURA M-13

Esta pantalla permite capturar dos números correspondientes a la identificación del nuevo lote y el tipo de las piezas del lote.

M.6.4 TENDENCIAS LOTE.

Esta opción realiza la misma función que la opción de tendencias mencionada en el inciso M.2.2 con la única diferencia de que esta opción obtiene la gráfica y datos estadísticos para el conjunto de durezas capturado en el inciso anterior.

Para abandonar el sistema es suficiente presionar <ESC> desde el menú principal con lo cual el sistema confirmará la petición de salir mediante la pantalla M-14 :

FORJAMEX

SISTEMA DE SUPERVISION Y CONTROL: HORNOS DE TEMPLE Y REVENIDO

DESEA SALIR DEL SISTEMA? SI = ENTER/NO = ESC

Biko: * ALG.
 * J 2 GM.
 * APH

Seleccíone con flechas y presíone la tecla de <ENTER>

FIGURA M-14

Si el usuario decide realmente abandonar el sistema, **BIKO** checa el modo actual en que se encuentran los movimientos mecánicos de ambos hornos, si alguno de ellos se encuentra en modo Automático se preguntará al usuario si desea regresarlos a Manual antes de abandonar el sistema. Lo anterior se lleva cabo mediante dos pantallas mostradas en las figuras M-15 y M-16.

FORJAMEX	
SISTEMA DE SUPERVISION Y CONTROL:	HORNOS DE TEMPLE Y REVENIDO

LOS MOVIMIENTOS MECANICOS DEL HORNO DE TEMPLE SE ENCUENTRAN EN AUTOMATICO, DESEA REGRESARLOS A MANUAL? SI = ENTER/NO = ESC	
---	--

Biko: * ALG.
 * J² GM.
 * APH

Seleccione con flechas y presione la tecla de < ENTER >

FIGURA M-15.

FORJAMEX	
SISTEMA DE SUPERVISION Y CONTROL:	HORNOS DE TEMPLE Y REVENIDO

LOS MOVIMIENTOS MECANICOS DEL HORNO DE ENCUENTRAN EN AUTOMATICO, DESEA REGRESARLOS A MANUAL?	REVENIDO SE
--	-------------

Biko: * ALG.
 * J² GM.
 * APH

Seleccione con flechas y presione la tecla de <ENTER >

FIGURA M - 16

Conclusiones



CONCLUSIONES

La Automatización Industrial está abriendo camino a más y más aplicaciones de Computación cada día. Basta con hojear revistas de Control tales como *InTech*, *Industrial Computing*, *Control Engineering*, entre otras, para visualizar el extenso campo de acción que puede abarcar la Computación dentro del Control Industrial.

En México, como en el resto de los países en vías de desarrollo, el *boom* de la Computación Industrial parece inminente, tal como está sucediendo en los países industrializados. Es evidente, que el rezago tecnológico es uno de los obstáculos que todo país debe superar para su crecimiento. Hoy día, se habla mucho sobre la modernización de la industria mexicana y demás factores que involucran al sector empresarial. De alguna forma, el proyecto **BIKO** es uno de los primeros pasos en la materia, no precisamente por su complejidad e importancia a nivel nacional, sino por ser, hasta donde sabemos, el primero en su especie diseñado y desarrollado en México.

BIKO es un proyecto práctico y como tal, no persigue conclusiones teóricas, sino resultados que se apeguen a las especificaciones del funcionamiento buscado. En este sentido, podemos concluir que el *Background* y el *Foreground* diseñado a lo largo de esta tesis, funcionan de acuerdo a lo especificado y a satisfacción del cliente. Este hecho, por sí solo habla de la eficacia del sistema diseñado pero ¿qué hay de su eficiencia? Contestar esta pregunta requiere definir parámetros, en relación a los cuales se va a determinar la eficiencia del mismo. Sin embargo, más que un cálculo detallado de eficiencia, en estas conclusiones queremos aportar datos significativos, ideas y experiencias concretas acerca de nuestro trabajo.

- **BIKO es un sistema de bajo costo y corto tiempo de desarrollo.** Como todo proyecto real, **BIKO** debería ser un sistema costeable. Este objetivo fué alcanzado exitosamente ya que el monto global aproximado del sistema no rebasa los \$ 4,000 dólares que se encuentra muy alejado de los \$9,000 dólares de un *software* extranjero equiparable con **BIKO**.

- **BIKO cumple con ser un sistema en tiempo real ya que su tiempo de respuesta es de 1 a 2 seg.** Como Sistema de Supervisión y Control de un Proceso Industrial, **BIKO** debería ser un sistema en tiempo real de respuesta rápida. El tiempo de respuesta mínimo para que un sistema se considere en tiempo real depende de la aplicación. En el caso de **BIKO**, el tiempo de respuesta está en el rango de 1 a 2 seg que, para la aplicación en particular, resulta muy eficiente.
- **Los sistemas de Control basados en Controladores Programables son altamente eficientes.** Cabe resaltar que un punto importante para el corto tiempo de respuesta de **BIKO** es el tiempo de *SCAN* del programa de Escalera y éste resultó ser de 30 mseg, esto es, en 80 mseg el PLC ha adquirido los datos de todo el proceso (236 Entradas) y ha efectuado todas las acciones de Control (276 Salidas) correspondientes a la lógica de Control. Este tiempo de *SCAN* tan corto habla de la eficiencia del Sistema de Control que permite un esquema basado en Controladores Programables.
- **Viabilidad de desarrollar software en áreas poco exploradas en México.** Sin duda, **BIKO** es un ejemplo que demuestra que en México, se tiene la capacidad de crear aplicaciones competitivas en el extranjero y más aún, en áreas con amplio campo por explorar, como es el caso del Control Industrial. Para apoyar esta afirmación citaremos el ejemplo de un proyecto similar, desarrollado en Estados Unidos recientemente.

En la publicación del mes de marzo/1990 de *INTECH*, aparece un artículo titulado *PC's AND PLC's JOIN FORCES IN BATCH REACTOR CONTROL*, donde se describe un sistema de Control y Supervisión basado en un PLC 5-15 de Allen-Bradley, con los módulos de Entrada/Salida correspondientes, el cual controla un total de:

16 entradas analógicas.

48 entradas discretas.

8 salidas analógicas.

83 salidas discretas.

y una PC-AT utilizada como terminal que monitorea el proceso empleando animación, tendencias, alarmas, etc. Un comando de Control al proceso tiene un tiempo de respuesta de 2 seg.

La revista menciona que el equipo de trabajo que desarrolló este proyecto tiene 17 años de experiencia en el campo del Control.

- **Esbozamos una metodología para desarrollar interfaces de operador para Supervisión y Control Industrial.** Aún cuando BIKO es una aplicación particular, nos ha permitido bosquejar una metodología general de desarrollo para sistemas de este tipo, y lo que es mejor, ha dado bases para el futuro desarrollo de un Sistema Configurable de Propósito General para Supervisión y Control de Procesos Industriales.
- **Experimentamos la importancia del vínculo escuela-Industria.** Es de suma importancia que cualquiera que sea el tema desarrollado como tesis, éste no se limite a un desarrollo teórico, sino por el contrario busque su aplicación práctica.

Un proyecto de Automatización de un proceso industrial no es lo mismo que un proyecto de Automatización de una oficina administrativa, no es lo mismo actualizar una Base de Datos que mover una grúa electromecánica. El riesgo involucrado es incomparable, pero alguien debe tomar ese riesgo.

El desarrollo de todo el proyecto ha arrojado un conjunto de experiencias de significativo valor para nosotros. Sin lugar a dudas, estas experiencias nos servirán como soporte para futuros proyectos dentro del área del Control Industrial.

Apéndice A. PLC 5-15

Apéndice A. PLC 5-15

A.1 Especificaciones Técnicas del PLC 5-15.

En el renglón correspondiente al PLC 5-15 de la tabla A-1, *Programmable Controller Processors*, encontrará las especificaciones de este PLC.

Las tablas A-2, A-3 y A-4, al final del apéndice, conforman un catálogo de referencia de módulos para Controladores Programables de Allen-Bradley.

Para mayor información referirse a los manuales correspondientes.

A.2 Resumen del Conjunto de Instrucciones del PLC 5-15.

Este apéndice contiene, a manera de resumen, el conjunto de instrucciones básicas del PLC 5-15 de Allen-Bradley, agrupadas por tipo para facilitar la lectura. Es una guía rápida del set de instrucciones. El capítulo 6 de esta tesis es un marco de referencia introductorio a los PLC's, mientras que este apéndice contiene información más detallada. La descripción completa de cada instrucción se encuentra en el Manual de procesador de dicho PLC.

A.2.1 Almacenamiento de Datos.

El procesador del PLC 5-15 soporta hasta 1000 archivos de datos de varios tipos. Un archivo de datos (*data file*) en un PLC 5-15 no es un medio de almacenamiento secundario. El término archivo de datos se refiere a la organización o mapa de la memoria principal del PLC 5-15. Por lo que se debe tener cuidado y no confundir los conceptos. Pero para

conservar congruencia con las referencias técnicas, manejaremos el término archivo de datos bajo esta premisa.

Cada tipo de archivo puede almacenar hasta 1000 elementos. Un elemento emplea una, dos o tres palabras dependiendo del tipo de archivo.

La figura A-1 Muestra la organización de la memoria del archivo de datos para la familia de procesadores PLC 5-15:

ORGANIZACION DE LA MEMORIA DE LA FAMILIA 5-15

No MAX ELEMENTOS	TIPO DE ARCHIVO	No. DE ARCHIVO	PALABRAS POR ELEMENTO
32	Imagen de salida	0	1
32	Imagen de entrada	1	1
32	Status	2	1
1000	Bit	3	1 (16 bits)
1000	Timer	4	3
1000	Control	6	3
1000	Entero	7	1
1000	Punto flotante	8	2
...
1000	Asignar tipo de archivo según se requiera	9 - 999	

FIGURA A-1

El tipo de archivo está designado según el tipo de dato que va a almacenar. No se pueden mezclar tipos de datos en un archivo.

El direccionamiento lógico de los datos se puede llevar a cabo con direccionamiento directo, direccionamiento indirecto y direccionamiento indexado.

A.2.2 Modos de Direccionamiento.

Direccionamiento Lógico.

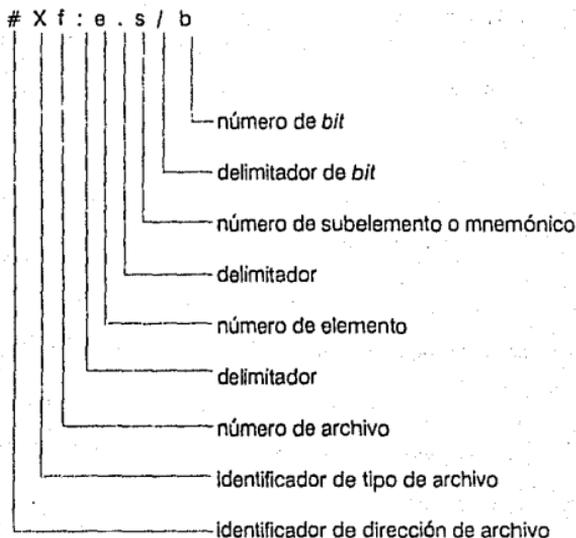
Los modos de direccionamiento lógico en el PLC 5-15 son 3:

- Directo.
- Indirecto.
- Indexado.

En este Apéndice sólo será tratado el direccionamiento directo.

Direccionamiento Lógico Directo.

Este modo de direccionamiento permite acceder un *bit*, un elemento o un archivo de datos. El formato de la dirección varía, dependiendo del tipo de dirección. El formato y símbolos generalizados de este modo de direccionamiento son:



Cada símbolo identifica a lo siguiente:

#: Identificador de dirección de archivo.

X: Tipo de archivo

B = *bit* C = contador

F = punto flotante I = entrada

N = entero O = salida

R = Control S = status

T = *timer* A = ASCII

D = BCD

f: Número de archivo

0 = salida 1 = entrada

2 = status 3 = *bit*

4 = *timer* 5 = contador

6 = Control 7 = entero

8 = punto flotante 9 - 999 para archivos adicionales

: Separa números de archivo y elemento.

e: Número de elemento

0-37 octal para archivos de E/S

0-31 decimal para el archivo de status

0-999 para todos los demás tipos de archivo

. Delimitador para subelemento en contadores, *timers* y archivos de Control.

s: Mnemónico para subelemento en *timers*, contadores y archivos de Control.

Direcciones de Palabras en Archivos de Timers, Contadores y Control.

Usar el mnemónico y el delimitador (.) según el tipo de archivo como sigue:

Mnemónico en archivos T ó C	Mnemónico en archivos R
.PRE (preset)	.LEN (longitud)
.ACC (Accumulated)	.POS (posición)

Por ejemplo, C5:8.ACC, según el formato general Xf:e.s, identifica el subelemento valor acumulado del archivo 5 de contadores, elemento 8.

/: Este delimitador separa números de *bit* de números de elemento o subelemento

b: Número de *bit*

0-17 octal para archivos de E/S

0-15 decimal para todos los demás tipos de archivos.

Direcciones de bit en Archivos de Timers, Contadores y Control.

Sólo se accesan los siguientes números de bit o mnemónicos:

bit	timer	counter	control
15	EN enable	CU up enable	EN enable
14	TT timing	CD down enable	EU unload enable
13	DN done	DN done	DN done
12	-	OV overflow	EM empty
11	-	UN underflow	ER error
10	-	-	UL unload
09	-	-	IN inhibit
08	-	-	FD found

Direccionamiento de E/S:

El formato para direccionamiento de E/S difiere del formato general. La forma de direccionar palabras o *bits* en archivos de E/S es la siguiente:

O:rg/00-17

I:rg/00-17

donde: I = entrada

O = salida

r = número de *rack* asignado

g = número de grupo de E/S

00-17 = número de *bit*

Direccionamiento del Archivo de Status

El formato para el direccionamiento del archivo de status también difiere del formato general. Se direccionan palabras y *bits* en el archivo de status como sigue:

S:e/b

donde: S = status

e = elemento

b = número de *bit*

Ejemplos de Direccionamiento Directo.

1. Formato para Direcciones de Bit

B3:15/5 Archivo binario 3, elemento 15, *bit* 5.

I:02/10 Ejemplo para E/S y *bits* de status que tienen asignados los números de archivo 1,0, y 2,

respectivamente; *rack* 0, grupo de E/S 2, *bit* de entrada 10.

C5:6.DN Archivo de contadores número 5, elemento 6, *bit* 13 (*bit done*).

N7:64/00 Archivo de enteros 7, elemento 46, *bit* 0.

2. Formato de Dirección de Subelemento por Mnemónico.

T4:12.ACC Ejemplo de formato para direccionar una palabra en un archivo de *timers*; valor acumulado del *timer* 12 en el archivo 4.

3. Formato de Dirección por Elemento.

S:3 Ejemplo de formato para direccionar una palabra en un archivo de status; elemento 3 del archivo de status.

F8:58 Elemento 58 del archivo número 8 de punto flotante.

A.2.3 Instrucciones Lógicas de Relevador

Las instrucciones lógicas de relevador y sus mnemónicos encerrados en paréntesis son:

Examine If Closed (XIC) { examinar si el contacto está cerrado }

Examine If Open (XIO) { examinar si el contacto está abierto }

Output Energize (OTE) { energizar salida }

Output Latch (OTL) { energizar salida y conservarla así }

Output Unlatch (OTU) { desenergizar salida }

Estas instrucciones se emplean para monitorear el status de los *bits* en la tabla de datos, tales como *bits* de entrada o *bits* de la palabra de Control de un *timer*. De igual forma, se pueden utilizar para Control de los *bits* en la tabla de datos como son los *bits* de salida.

Examine If Closed (XIC) -] [-

Cuando un dispositivo de entrada cierra su circuito, la terminal de entrada conectada al dispositivo indicará un estado ON. Este estado ON se verá reflejado en la memoria en el *bit* correspondiente. Cuando el procesador encuentra una instrucción **XIC**, determina si el dispositivo de entrada con esa dirección esta ON y de ser así, asigna VERDADERO al valor lógico de la instrucción. De lo contrario, el procesador asigna un valor FALSO a la instrucción.

Examine If Open (XIO) -] [-

La instrucción **XIO** es la versión negada de XIC. Cuando el procesador encuentra una instrucción **XIO**, determina si el dispositivo de entrada con la dirección asociada esta OFF, de ser así, asigna VERDADERO y viceversa.

Output Energize (OTE) -() -

Usese la instrucción OTE para fijar un *bit* particular en memoria. Si la dirección del *bit* en la instrucción OTE corresponde a la dirección de una

terminal de un módulo de salida, el dispositivo conectado a esa terminal será energizado.

Una instrucción **OTE** sustituiría un relevador en una lógica alamburada. La instrucción es controlada por las instrucciones precedentes en el escalón que, entre otras pueden ser **XIC** y **XIO**.

Output Latch (OTL) -(L)- ; Output Unlatch (OTU) -(U)-

OTL y **OTU** son instrucciones de salida retentivas. Generalmente son usadas en pares para cada *bit* de la tabla de datos que controlen.

La diferencia entre la instrucción **OTE** y la instrucción **OTL** es que, la segunda mantiene el estado de la salida constante hasta que es desenergizada con una instrucción **OTU**. Mientras que la primera sólo dura un instante de tiempo suficiente para activar la salida.

A.2.4 Instrucciones de Timers y Contadores.

Las instrucciones de *timers* y contadores y sus mnemónicos incluyen:

Timer On-delay (TON) { *timer On* }

Timer Of-delay (TOF) { *timer Off* }

Retentive Timer (RTO) { *timer retentivo* }

Count Up (CTU) { contador ascendente }

Count Down (CTD) { contador descendente }

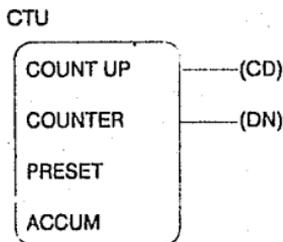
Counter or Timer Reset (RES) { Reset de *timer* o contador }

Estas instrucciones permiten muchas de las capacidades de los relevadores de tiempo o de los *timers* y contadores de estado sólido.

Estas son instrucciones de salida que se pueden condicionar por instrucciones de entrada tales como **XIC** y **XIO**. Los *timers* cuantifican

intervalos de tiempo y los contadores cuantifican eventos, según sea determinado por la lógica del programa de aplicación.

El formato de una instrucción de contador es la siguiente:



Cada instrucción de *timer* o contador tiene dos valores asociados a ella. Estos valores son:

PRESET. Este es el valor determinado de fijación (*set point*). Se introduce este valor para gobernar el cronometraje o conteo de la instrucción. Cuando el valor acumulado es igual o mayor al valor de *preset*, un *bit* de status es cambiado. Se puede emplear este *bit* para controlar un dispositivo de salida.

ACCUM. Este es el número corriente de unidades de tiempo que han sido cuantificadas por una instrucción de *timer* o para una instrucción de contador, el número de eventos que han ocurrido.

Las instrucciones de *timer* y contador requieren de tres palabras de la tabla de datos, cada una para:

- valor acumulado
- valor de *preset*
- palabra de Control

Los datos en estas palabras son almacenados bajo el formato de enteros.

Tanto el *preset* como el valor acumulado para los *timers* tienen un rango de 0 a +32,767; en el caso de los contadores el rango es de -32,768 a +32,767.

Si el *preset* o valor acumulado de un *timer* es un número negativo, ocurrirá un error en tiempo de ejecución y causará una falla en el procesador.

A.2.5 Instrucciones Aritméticas, Lógicas y de Transferencia.

Las instrucciones aritméticas, lógicas y de transferencia se pueden ejecutar sobre palabras (de datos) individuales. Estas instrucciones son instrucciones de salida. Las operaciones y sus mnemónicos se listan a continuación:

Aritméticas.

En el formato de una instrucción aritmética se incluyen generalmente las direcciones fuente (operandos) y la dirección destino donde se registrará el resultado. La lista incluye:

Add (ADD)	{ sumar dos datos }
Subtract (SUB)	{ sustraer dos datos }
Multiply (MUL)	{ multiplicar dos datos }
Divide (DIV)	{ dividir dos datos }
Square Root (SQR)	{ raíz cuadrada de un dato }
Negate (NEG)	{ cambio de signo de un dato }

Clear (CLR)	{ asignar valor cero a un dato }
Convert to BCD (TOD)	{ convertir un dato a BCD }
Convert from BCD (FRD)	{ convertir de BCD un dato }

Lógicas.

Permiten efectuar operaciones a nivel de *bits* sobre los datos (*bitwise*):

And (AND)	{ and lógico }
Or (OR)	{ or lógico }
Exclusive Or (XOR)	{ or exclusivo }
Not (NOT)	{ not lógico }

Transferencia.

Estas instrucciones permiten transferir (mover) datos de una localidad (dirección) a otra, pudiéndose emplear opcionalmente una máscara para filtrar el dato a transferir, es decir, se efectúa una operación **AND bit a bit** antes de la transferencia. Las instrucciones de este grupo son:

Move (MOV)	{ mueve el contenido de una localidad a otra }
Masked Move (MMV)	{ mueve un dato a través de una máscara }

Por ejemplo. Si se deseara transferir el contenido de la dirección C a la dirección P empleando la máscara F0F0 hexadecimal, la instrucción sería:

MVM

MASKED MOVE	
SOURCE	C
MASK	F0F0
DEST	P

Si los contenidos antes de la instrucción son:

C = 0101010101010101

P = 1111111111111111

F0F0(hex) = 1111000011110000

el destino después de ejecutada la instrucción será:

P = 0101111010111111

Instrucciones de Comparación.

Las instrucciones de comparación permiten comparar valores entre dos datos. Son instrucciones de entrada. Las operaciones que se pueden efectuar dentro de este grupo y sus mnemónicos son:

Equal (EQU)	{ igual que }
Not Equal (NEQ)	{ no igual que }
Less Than (LES)	{ menor que }

Limit Test (LIM) { prueba que un valor esté dentro de un rango }

Less Than or Equal (LEQ) { menor o igual que }

Greater Than (GRT) { mayor que }

Greater Than or Equal (GEQ) { mayor o igual que }

Masked Comparision for Equal (MEQ) { comparación con máscara }

Los parámetros que se introducen a estas instrucciones son constantes de programas o directamente las direcciones lógicas de los valores a comparar.

Está permitido comparar valores de diferentes tipos de datos como punto flotante y enteros. Sin embargo, sólo se deben emplear valores BCD y ASCII para despliegues.

También se pueden efectuar estas operaciones con la instrucción **Compare (CMP)** excepto **MEQ** y **LIM**, pues el formato de la instrucción **CMP** recibe una expresión como $A = B$.

Por ejemplo, el formato de la instrucción **EQU** es:

EQU

EQUAL
SOURCE A
SOURCE B

Cuando los valores fuente A y B son iguales, la instrucción es lógicamente verdadera. Si estos valores no son iguales, la instrucción es lógicamente falsa.

A.2.6 Instrucciones de Archivos.

Instrucción File Arithmetic and Logic (FAL)

Esta instrucción de salida efectúa las siguientes operaciones aritméticas y lógicas:

Aritméticas:

adición $A + B = C$

substracción $A - B = C$

multiplicación $A * B = C$

división $A / B = C$

raíz cuadrada $SQR A = C$

cambio de signo $- A = C$

asignar cero $0 = C$

mover $A = C$

convertir a BCD $TOD A = C$

convertir de BCD $FRD A = C$

Lógicas:

AND A AND B = C

OR A OR B = C

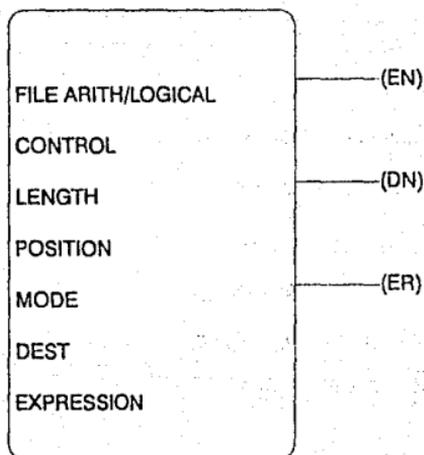
XOR A XOR B = C

NOT NOT A = C

La operación a efectuar se determina a través del campo correspondiente a la expresión en el formato de la instrucción.

El formato de esta instrucción es:

FAL



La instrucción **FAL** ejecuta transferencias de datos en una de tres formas, dependiendo del tipo de dirección que se emplee para la expresión y el destino:

- 1) archivo - archivo
- 2) elemento - elemento
- 3) archivo - elemento

Es decir, una instrucción de operación sobre archivos **FAL** es una extensión de las instrucciones elementales aritméticas y lógicas, pero con la facilidad de manipular archivos de datos:

Los parámetros de la instrucción son los siguientes:

CONTROL.- Es la dirección del elemento constituido por tres palabras que controla la operación de la instrucción. El elemento almacena el status de la instrucción, longitud del archivo y posición del elemento actuado en cualquier instante dado.

15	13	11	00	BITS
EN	DN	ERR	STATUS	
LONGITUD				
POSICION				

El status (palabra 0 del elemento de Control) indica el status de la instrucción como sigue:

EN (bit 15). El *bit enable* (habilitado) se activa en la transición de falso a verdadero del escalón e indica que la instrucción esta habilitada. Continúa activado hasta que la instrucción es completada independientemente de

la condición del escalón en modo numérico o total. En el modo incremental, sigue a la condición del escalón.

DN (bit 13). El *bit done* (efectuado) es activado una vez que ha sido completada la operación sobre todo el archivo. Es desactivado cuando el escalón cambia su valor a falso.

ER (bit 11). El *bit de error* es activado cuando el procesador detecta un error aritmético o lógico en la operación. La instrucción detiene la operación hasta que el programa de escalera desactiva este *bit*.

LONGITUD. (Palabra 1 del elemento de Control) es la longitud de los archivos de la instrucción sobre los cuales se va a trabajar. El máximo es 1000 elementos.

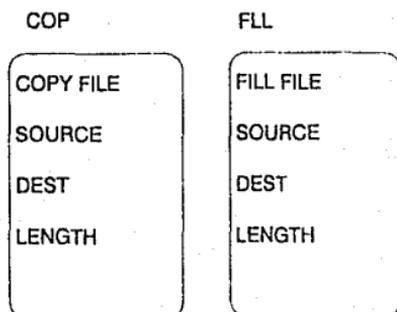
POSICION. (palabra 2 del elemento de Control) almacena la posición del elemento sobre el cual se está operando en cualquier instante dado. La operación es completada cuando el valor de la posición se iguala al valor de la longitud.

MODO. - determina el número de elementos de archivo ejecutados por SCAN en cada la instrucción.

EXPRESION. - Determina la operación ejecutada en la instrucción. La expresión generalmente consiste de dos direcciones, un símbolo aritmético o expresión lógica.

Instrucciones File Copy (COP) y File Fill (FLL)

Estas instrucciones tienen el mismo formato:



La instrucción **File Copy** es una instrucción de salida de alta velocidad que opera en forma similar a una instrucción **MOV** de archivo a archivo operando en modo total (*ALL MODE*). Copia datos de una localidad a otra.

Los parámetros de la instrucción son:

SOURCE.- es la dirección del archivo a copiar.

DEST.- es la dirección destino donde la instrucción almacena la copia.

LENGTH.- es el número de elementos en el archivo que van a ser copiados.

Los elementos son copiados del archivo fuente especificado al archivo destino especificado cada *SCAN* que el escalón es verdadero. Los elementos son copiados en orden ascendente sin transformación alguna de los datos. Los elementos son copiados hasta que se alcanza el número especificado o hasta que se alcanza el último elemento del archivo, según suceda primero.

La instrucción **File Fill** es similar a la instrucción **File Copy**. Inicializa elementos de un archivo ya sea con una constante de programa o un valor de una dirección directa o indirecta.

Instrucción File Search and Compare (FSC)

La instrucción **FSC** es una instrucción de salida que puede ejecutar las siguientes operaciones:

Search Equal **A = B**

Search Not Equal **A <> B**

Search Less Than **A < B**

Search Less Than or Equal **A <= B**

Search Greater than **A > B**

Search Greater than or Equal **A >= B**

La operación a efectuar se determina por la expresión dada en el formato general de la instrucción, que es igual al formato de la instrucción **FAL**.

Instrucciones FIFO (FFL,FFU) y Bit Shift (BSL,BSR).

Las instrucciones **FIFO** (*First Input First Output*) **Fifo Load (FFL)** y **Fifo Unload (FFU)**, permiten la facilidad de manipular archivos de forma similar a estructuras de datos **FIFO**, también conocidas como Colas. La instrucción **FFL** almacena un dato en la cola mientras que la instrucción **FFU** extrae un dato de la cola.

Por su parte, las instrucciones **Bit Shift Left BSL** y **Bit Shift Right BSR** permiten manipular archivos de datos como arreglos de *bits* de mucha capacidad y efectuar sobre ellos operaciones de corrimiento a la izquierda **BSL** y corrimiento a la derecha **BSR**.

Instrucciones de secuenciador (SQO,SQI,SQL).

Las instrucciones de secuenciador permiten efectuar las siguientes funciones:

Sequencer Output (SQO) transfiere palabras de 16 *bits* hacia direcciones específicas para Control de operaciones secuenciales de máquinas.

Sequencer Input (SQI) compara palabras de 16 *bits* contra palabras almacenadas para monitorear las condiciones de operación de máquinas o para propósitos de diagnóstico.

Sequencer Load (SQL) carga palabras de 16 *bits* en archivos de datos para capturar condiciones de referencia.

Quando la aplicación requiera múltiplos de 16 *bits* de Entrada/Salida, coloque en paralelo instrucciones de salida de secuenciador o coloque en serie instrucciones de entrada de secuenciador en el programa de escalera.

A.2.7 Instrucción PID.

La instrucción **PID** permite al procesador el monitoreo y lazos de Control para variables como presión, temperatura, flujo o nivel de un fluido. Es seleccionable el tipo de ecuación de Control **PID** (ganancias independientes o ISA) y los parámetros de Control en cualquier combinación proporcional, integral y/o derivativa.

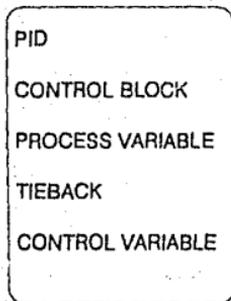
Las características de la instrucción incluyen:

- Ecuación **PID** expresada en ISA o ganancias independientes.
- Rangos de entrada/salida de 0 a 4095.
- Escalamiento de entradas en unidades de ingeniería.
- Banda muerta en cruce por cero.

- Término derivativo (puede actuar sobre la variable de Control PV o el error).
- Operación en modo Manual.
- Perturbación Artificial (Feedforward).
- Despliegue y monitoreo de variables PID.

El formato de la instrucción PID es:

PID



Los parámetros de la instrucción son:

CONTROL BLOCK.- es la dirección entera del bloque de Control que almacena el status PID y los bits de Control, constantes y variables PID, y otros parámetros asociados con la instrucción PID.

PROCESS VARIABLE.- es la dirección del elemento de la cual la instrucción PID obtiene el valor de entrada de la variable del proceso.

TIEBACK.- es la dirección de un elemento de entrada, empleado para efectuar transferencias sin saltos cuando una estación Manual de Control

es empleada. El *tieback* es habilitado al introducirse la dirección de la estación de Control Manual en la instrucción **BTR**.

OUTPUT- (*Control Variable*) es la dirección de un elemento al cual la instrucción **PID** envía el valor de salida calculado.

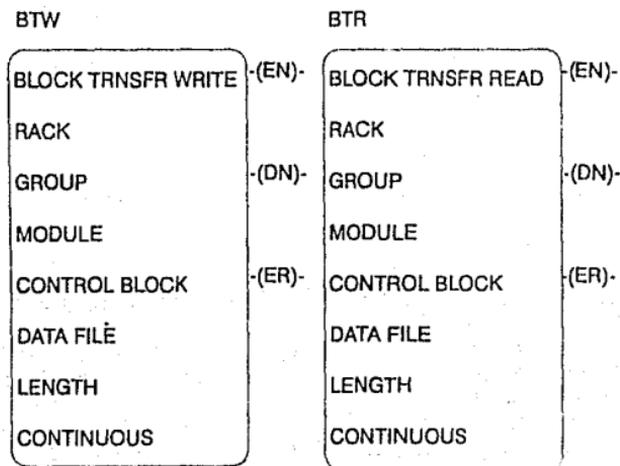
A.2.8 Instrucciones de Transferencia de Bloques (BTR,BTW).

Estas instrucciones ejecutan las siguientes funciones:

Block-Transfer-Write (BTW) transfiere hasta 64 palabras a la vez del procesador hacia un módulo inteligente de E/S en un chasis local o remoto.

Block-Transfer-Read (BTR) transfiere hasta 64 palabras a la vez de un módulo inteligente de E/S en un chasis local o remoto hacia el procesador.

La mayoría de los módulos de entrada con los cuales se comunica el procesador son diseñados para transferencias de bloque bidireccionales usando una instrucción (**BTR**) y una instrucción (**BTW**), las cuales son instrucciones de salida.



Los parámetros de la instrucción son los siguientes:

RACK.- es el número de *rack* asignado al chasis de E/S en el cual se monta el módulo de E/S. El rango en el PLC 5-15 es de 0-3.

GROUP.- es el número de grupo de E/S que especifica la posición del módulo de Entrada/Salida direccionado en el chasis de E/S. El rango es de 0-7

MODULE.- es el número de *slot* dentro del grupo de E/S.

CONTROL BLOCK.- es un archivo de 5 palabras que controla la operación de la instrucción.

DATA FILE.- es la dirección del archivo de transferencia de bloques hacia el cual escribe o del cual lee el procesador.

LENGTH.- es el número de palabras que el módulo transfiere.

CONTINUOUS.- determina el modo de ejecución de la instrucción.

Discrete I/O Modules

Module/Catalog No.	Voltage	No. Points	Backplane Current	Current Per Point	Maximum Current Per Module	Input Delay Time	Slots	Applications	Spring Arm, 1771-
		Inputs Output							
AC/DC 1771-A	120V	8	0.075A		200 mA	200 ms	1	General purpose I/O inputs	WA
AC High Density 1771-AQ	120V	16	0.25A		400 mA on 40 ms off	200 ms	1	120V inputs requiring primary protection	WA
Isolated AC/DC 1771-Q	120V	8	0.025A		200 mA	200 ms	1	Isolated 120V inputs	AD
AC/DC 1771-AJ	120V	8	0.075A		200 mA	200 ms	1	120V outputs requiring primary protection	WA
AC/DC 1771-AM	220/240V	8	0.075A		200 mA	200 ms	1	General purpose 220/240V inputs	WA
High Density Input 1771-AD	220V AC	16	0.25A		50 mA on 100 ms off	200 ms	1	General purpose 220/240V AC inputs	WA
AC 1771-AB	24V	8	0.085A		15 ± 10 mA	10 ms	1	Low voltage AC devices	WA
DC 1771-AC	12-24V	8	0.075A		15 ± 7 mA	10 ms	1	Low voltage DC devices	WA
DC High Density 1771-AD	10-30V	16	0.25A		5 mA ± 2	10 ms	1	Low voltage DC inputs	WA
32 of High True Input 1771-AN	10-30V DC	32	0.25A		5 mA ± 2	10 ms	1	Sourcing input devices	WA
DC High Density 1771-AD	20-40V	16	0.15A		5 ± 2 mA on 20 ± 1 ms off	10 ms	1	Medium voltage DC devices	WA
DC 1771-AL	24-48V	8	0.075A		15 ± 7 mA	10 ms	1	Medium voltage DC devices	WA
DC Sensitive Input 1771-AQ	5-30V	8	0.150A		1 ± 10 mA	10 ms	1	Sourcing or sinking device	AD
Fast Response DC 1771-AL	12-24V	8	0.075A		1 mA	10 ms	1	Inputs requiring maximum fast speed	WA
DC Logic Driver 1771-AL	12-24V	8	0.075A		15 ± 7 mA	10 ms	1	Current sourced DC inputs	WA
32 of Low True Input 1771-AN	10-30V DC	32	0.25A		5 mA ± 2	10 ms	1	Sinking input devices	WA
RL 1771-AG	24V AC	8	0.125A		1 mA	10 ms	1	RL inputs	WA
24V AC High Density Input 1771-AD	24V AC	16	0.25A		10 mA on 30 ms off	10 ms	1	Low voltage 24V AC inputs	WA
AC 1771-QA	120V	8	0.125A	1.5A 5A	5A	200 ms	1	General purpose 120V AC outputs	WA
AC High Density 1771-QAQ	120V	16	0.2A	2A 5A	5A	200 ms	1	120V AC outputs	WA
Isolated AC 1771-QD	120V	8	0.255A	2A 5A	5A	200 ms	1	Isolated 120V AC outputs	AD
AC 1771-QH	200-240V	8	0.125A	1.5A 5A	5A	200 ms	1	200-240V AC loads	WA
120V AC High Density Output 1771-QAQ	184-252V AC	16	0.18A	2A 5A	5A	200 ms	1	120V AC outputs	WA
AC 1771-QV	24V	8	0.255A	2A 5A	5A	200 ms	1	Low voltage 24V outputs	WA
Protected AC 1771-QP	120V	4	0.25A	2A 5A	5A	200 ms	1	Center-tapped outputs	WA
Protected AC 1771-QR	220V	8	0.255A	2A 5A	5A	200 ms	1	Isolated 220V AC outputs	AD
DC Drive 1771-OB	12-24V	8	3.105A	1.5A 5A	5A	200 ms	1	Low voltage DC outputs	AA
DC 1771-OC	48V	4	0.355A	1.5A 5A	5A	200 ms	1	48V DC outputs	WA
DC High Density 1771-QAQ	10-60V	16	0.2A	2A 4A	4A	200 ms	1	10-60V DC outputs	WA
32 of High True Output 1771-QAN	10-32V AC	32	0.33A	0.5 5A	5A	200 ms	1	Sinking input devices	WA
Isolated DC 1771-QD	24V	8	0.215	2.5A 5A	5A	200 ms	1	Isolated 24V DC loads	AD
32 of Low True Output 1771-QAN	10-30V DC	32	0.33A	0.5 5A	5A	200 ms	1	Sourcing input devices	WA
RL 1771-OG	1V	8	0.158	20mA 0.05A 1mA	0.05A	200 ms	1	RL loads, source or sink	AA
8 of Sensitive Contact 1771-OW	12-136V AC 5-12V DC	8	0.3A	70 mA voltage sensitive	120V AC 300mA DC	200 ms	1	High contact sensitive outputs	AD
High Power Contact 1771-OW	250V	4	0.35A	2A 5A 120V AC 300mA DC	120V AC 300mA DC	200 ms	1	250V AC 500mA AC/DC 300mA DC	AD
Contact Output 1771-OW	0-120V	8	0.2A	0.25 A ± 10% voltage sensitive 120V AC 300mA DC	120V AC 300mA DC	200 ms	1	2-120V AC/DC 300mA DC	AD
Converter Output 1771-OW	0-120V	8	0.35A	0.25 A ± 10% voltage sensitive 120V AC	120V AC 300mA DC	200 ms	1	4-120V AC/DC 300mA DC	AD

* 10 ± 4 ms on AC 20 ± 9 ms on DC 24 ± 10 ms AC

TABLA A-2

Racks

Catalog No.	Description	Number I/O Module Slots	Maximum Backplane Current	Dimensions	Comments
1771-A1B	I/O Chassis Assembly	4 Slots	16 amps max.	9" W 12.41" H 6.75" D	
1771-A2B	I/O Chassis Assembly	8 Slots	16 amps max.	14" W 12.41" H 6.75" D	
1771-A3B	I/O Chassis Assembly	12 Slots	16 amps max.	19" W rack mount 12.41" H 6.75" D	
1771-A4B	I/O Chassis Assembly	16 Slots	16 amps max.	24" W 12.41" H 6.75" D	
1771-P5C	Power Supply Chassis	4 Slots	16 amps max. supplied to an I/O Chassis	8" W 12.23" H 7.1" D	Supports only Slot Power Supplies and Communication Modules

Power Supplies

	1771-P1 ¹ , 1771-P2	1771-P3 ²	1771-P4 ²	1771-P5	1771-P7	1775-P1
Nominal Input Voltage	120 VAC or 220/240 VAC ³	120 VAC	120 VAC	24 VDC	120 VAC or 220 VAC ¹	120 VAC or 220/240 VAC ¹
Input Voltage Range	98-132 VAC or 196-250 VAC ³	97-132 VAC	97-132 VAC	20.5-30 VDC	97-132 VAC or 195-264 VAC ¹	97-132 VAC or 194-264 VAC ³
Input Power	75 VA	38 W	79 W	72 W	300 VA	600 W
Output Current	6.5A	3A	8A	8A	16A	60A 6A @ ±15 VDC
Frequency	60/50 Hz	47 to 63 Hz	47 to 63 Hz		47 to 63 Hz	47 to 63 Hz
Fuse	1A, 3AG, slow-blow or 0.5A, 3AB, slow-blow ¹	1A, slow-blow	1.5A, slow-blow	5A, 32 V, slow blow	3A, 250V, normal blow	10A (aircut breaker)
Dimensions	4.56 x 11.25 x 7.16 (WHD)	1 chassis slot	2 chassis slots	2 chassis slots	4.53 x 12.41 x 6.25	6.5 x 20 x 14.5
Weight	13 lb (5.89 kg)	1.6 lb (0.73 kg)	2.3 lb (1.04 kg)	2.3 lb (1.16 kg)	4.3 lb (1.95 kg)	37 lb (16.8 kg)
Module Location	chassis side, panel	1771 I/O chassis, single slot	1771 I/O chassis, two slots	1771 I/O chassis	chassis side or panel mount	standalone
Keying	NA	12 and 14 18 and 20	12 and 14 20 and 22	12 and 14 22 and 24	NA	NA

¹ Provides battery backup
² P3 and P4s can be networked together to provide 6, 11 or 16 amps
³ User selectable

TABLE A-4

Analog I/O

Catalog No.	Module Description	Number of Inputs/Outputs	Backplane Current Draw	External Power Required	Channel Update Resolution	Ranges		Processor Data Formats	Slots	Sling Arm
						Voltage	Current			
1771-E	Analog Input	8 Single-ended	400mA	-5V DC ±15V DC	2.5mS/8 chns 8 bit resolution	0-5V DC 1-5V DC 0-10V DC ±10V DC	4-20ma 0-20ma ±20ma	BCD 000 to 255	1	1771-WB
1771-EF	Analog Input	8 Single-ended (with Expanders up to 64 inputs)	1.3A	-5V DC ±15V DC	No Expanders 5mS/6 chns w/expanders 1.3mS/chns 12 bit resolution (IF and all expanders)	0-5V DC 1-5V DC 0-10V DC ±10V DC (E, E1, E2)	4-20ma 0-20ma ±20ma	BCD 000 to 999 Binary 0000 to 4095 (IF and all expanders)	2	1771-WB
1771-E1	Input Expander	8 Single-ended	150mA	±15V DC					1	1771-WF
1771-E2	Input Expander	6 Differential	150mA	±15V DC					1	1771-WF
1771-E3	Input Expander	6 two-wire sourcing	150mA	±15V DC			4-20ma 0-20ma		1	1771-WF
1771-EF	Analog Input	8 Differential 16 Single-ended Selectable	750mA	None	12.5mS/8 chns 25mS/16 chns 12 bit resolution	0-5V DC 1-5V DC ±5V DC ±10V DC	4-20ma 0-20ma ±20ma	Binary or BCD 0-4095, ±4095 Scaled BCD ±9999	1	1771-WG
1771-EL	Isolated Analog Input	8 Isolated Differential (1000 Volts)	1.3A	None	500mS/6 chns 12 bit resolution	0-5V DC 1-5V DC ±5V DC 0-10V DC ±10V DC	4-20ma 0-20ma ±20ma	Binary or BCD 0-4095, ±4095 Scaled BCD ±9999	1	1771-WF
1771-EX	Thermocouple Input	8 Floating Differential	1.2A	None	Type R,S 8 chns/sec Type E,J,K,T 16 chns/sec 1°C, 1°F or 10µV resolution	Type E,J,K,R,S ±99.99 mV		Binary or BCD Actual temperature or mV readings	1	1771-WF
1771-IR	RTD Input	6 RTD Inputs (Three wire)	1A	None	Platinum 330mS/chnl Copper ± 1.5mS/6 chnl 1°C, 2°F, 1 Ohm (plat) or other 3°C, 5°F, 1 Ohm (copper)	100 Ω Platinum 10 Ω Copper		Binary or BCD Actual temperature or Ohm readings	1	1771-WF
1771-OF	Analog Output	4 Common Ground (With Expanders up to 60 outputs)	1.4A	-5V DC ±15V DC	No Expanders 0.35mS/chnl w/Expanders 0.67mS/chnl 12 bit resolution (OF and E4)	0-5V DC 1-5V DC 0-10V DC ±10V DC	4-20ma 0-20ma ±20ma	BCD 000 to 999 Binary (OF and E4)	2	1771-WB
1771-E4	Output Expander	4 Common Ground	165mA			±5V DC (±100mV)			1	1771-WB
1771-OFE1, OFE2	Analog Output	4 Isolated outputs (1000 volts)	1.5A	None	8.6 msec/5 chns (BCD) 1.6 msec/8 chns (Binary) 12 bit resolution	1-5V DC 0-10V DC ±10V DC (OFE 1)	4-20ma (OFE 2)	Binary or BCD 0-4095, ±4095 Scaled BCD ±9999	1	1771-WC
1771-PD	PID Control Module	2 PV Inputs 2 Feedback Inputs 2 Outputs	1.2A	±15V DC	100mS (Loop Update) 12 bit resolution	1-5V DC	4-20ma	Scaled BCD ±9999	2	1771-WF

TABLA A-3

Analog I/O

Catalog No.	Module Description	Number of Inputs/Outputs	Backplane Current Draw	External Power Required	Channel Update Resolution	Ranges		Processor Data Formats	Slots	Swing Arm
						Voltage	Current			
1771-E	Analog Input	8 Single-ended	400mA	+5V DC ±15V DC	2.5mS/8 chns 8 bit resolution	0-5V DC 1-5V DC 0-10V DC ±10V DC	4-20ma 0-20ma ±20ma	BCD 000 to 255	1	1771-WB
1771-F	Analog Input	8 Single-ended (with Expanders up to 64 inputs)	1.3A	-5V DC ±15V DC	No Expanders 5mS/8 chns w/expanders 1.3mS/chns 12 bit resolution (F and all expanders)	0-5V DC 1-5V DC 0-10V DC ±10V DC	4-20ma 0-20ma ±20ma (F, E1, E2)	BCD 000 to 999 Binary 0000 to 4095 (F and all expanders)	2	1771-WB
1771-E1	Input Expander	8 Single-ended	150mA	±15V DC					1	1771-WF
1771-E2	Input Expander	6 Differential	150mA	±15V DC					1	1771-WF
1771-E3	Input Expander	6 two-wire sourcing	150mA	±15V DC			4-20ma 0-20ma		1	1771-WF
1771-EF	Analog Input	8 Differential 16 Single-ended Selectable	750mA	None	12.5mS/8 chns 25mS/16 chns 12 bit resolution	0-5VDC 1-5VDC ±5V DC 0-10V DC	4-20ma 0-20ma ±20ma	Binary or BCD Binary Scaled BCD ±9999	1	1771-WG
1771-IL	Isolated Analog Input	8 Isolated Differential (1000 Volts)	1.3A	None	500mS/8 chns 12 bit resolution	0-5V DC 1-5V DC ±5V DC 0-10V DC ±10V DC	4-20ma 0-20ma ±20ma	Binary or BCD Actual 0-4095, ±4095 Scaled BCD ±9999	1	1771-WF
1771-IX	Thermocouple Input	8 Floating Differential	1.2A	None	Type R,S 8 chns/sec Type E,J,K,T 16 chns/sec 1°C, 1°F or 10V resolution	Type E,J,K,R,S ±9999 mV		Binary or BCD Actual temperature or mV readings	1	1771-WI
1771-IR	RTD Input	6 RTD Inputs (Three wire)	1A	None	Platinum 330mS/chns Copper = 1 Sec/6 chns 1°C, 1°F, 1 Ohm (ptal) 3°C, 5°F, 1 Ohm (copper)	100 Ω Platinum or other temperature or Ohm readings		Binary or BCD Actual temperature or Ohm readings	1	1771-WF
1771-OF	Analog Output	4 Common Ground (With Expanders up to 60 outputs)	1.4A	-5V DC, ±15V DC	No Expanders 0.33mS/4 chns w/expanders 0.61mS/chns 12 bit resolution (OF and E4)	0-5V DC 1-5V DC 0-10V DC ±10V DC	4-20ma 0-20ma ±20ma (OF and E4)	BCD 000 to 9999 Binary 0-4095 (OF and E4)	2	1771-WB
1771-E4	Output Expander	4 Common Ground	165mA			±5V DC (- above)			1	1771-WB
1771-OFE1, OFE2	Analog Output	4 Isolated outputs (1000 volts)	1.5A	None	8.0 msec/6 chns (BCD) 1.0 msec/6 chns (Binary) 12 bit resolution (OFE 1)	1-5V DC 0-10V DC ±10V DC	4-20ma (OFE 2)	Binary or BCD 0-4095, ±4095 Scaled BCD ±9999	1	1771-WC
1771-PD	PID Control Module	2 PV Inputs 2 Feedback inputs 2 Outputs	1.2A	±15V DC	100mS (Loop Update) 12 bit resolution	1-5V DC	4-20ma	Scaled BCD ±99990	2	1771-WF

TABLA A-3

Apéndice B. Módulo BASIC

Apéndice B. Módulo BASIC.

B.1 Características Generales.

El módulo BASIC 1771-DB proporciona la capacidad de realizar funciones matemáticas, generación de reportes y lenguaje BASIC para cualquier procesador Allen-Bradley que se comunique con el sistema de E/S 1771 usando transferencia de bloques. El módulo BASIC contiene:

- Programación BASIC empleando el lenguaje BASIC-52 de Intel.
- Funciones matemáticas consistentes con la definición de BASIC-52.
- Dos puertos seriales independientemente configurables con capacidad de conexión a varios dispositivos de usuario.
- Reloj de tiempo real con 5 mseg de resolución accesible al usuario.
- Reloj-Calendarario con 1 seg de resolución accesible al usuario.
- Generación y edición de programa por medio de una terminal de programación.
- Capacidad de comunicación de transferencia de bloques desde un Procesador de familia PLC-2, PLC-3 o PLC-5.

B.2 Características de Hardware.

- 13 KBytes de **RAM** respaldada por batería para los programas del usuario.

PIN	NOMBRE	DESCRIPCIÓN
4	RTS	<p>RTS es una línea de salida de handshaking por hardware. Esta línea cambia a estado 1 cuando el módulo BASIC tiene datos en la cola de salida y está solicitando permiso para transmitir al DCE.</p>
5	CTS	<p>CTS es una línea de entrada de handshaking por hardware. Esta línea debe estar en estado 1 para que el módulo BASIC transmita al puerto periférico.</p>
6	DSR	<p>DSR es una línea de entrada compatible con RS-423A de propósito general. El módulo BASIC transmite o recibe en estado 1 o 0.</p>
7*, 9,		
10	SEÑAL COMUN	<p>Use los pines de señal común para referenciar todas las señales compatibles RS-423A/RS-422.</p>

PIN	NOMBRE	DESCRIPCION
20	DTR	DTR es una línea de salida de handshaking por hardware. Esta línea cambia a estado 0 cuando la cola de entrada del módulo BASIC ha acumulado más de 223 caracteres. La línea DTR cambia a estado 1 cuando la cola de entrada contiene menos de 127 caracteres.

*Pines del puerto de programación.

B.4 Especificaciones Técnicas.

Ver la Tabla B-1 al final de este apéndice.

B.5 Resumen del Conjunto de Instrucciones.

MNEMONICO	DESCRIPCION
DIM	Asigna memoria a un arreglo de variables.
DO	Establece iteración para instrucciones WHILE o UNTIL.
END	Termina la ejecución del programa.
EXP()	Exponencial del argumento.
FOR - TO	Establece iteración para el siguiente paso.
GET	Lectura de consola.
GOSUB	Ejecuta subrutina.
GOTO	Salto incondicional a una línea de programa.
IF - THEN - ELSE	Prueba condicional.
INPUT	Entrada de un string o variable entera.
INT()	Entero.
LD@	Carga variable.
LEN	Obtiene el tamaño en bytes del programa presente en memoria.
LET	Asigna valor a una variable o string.
LIST	Lista el programa a la consola.
LIST#	Lista el programa a la impresora serial.
LOG()	Logaritmo natural.

MNEMONICO	DESCRIPCION
MTOP	Lee la última dirección válida en memoria.
NEW	Borra el programa almacenado en RAM.
NEXT	Prueba la siguiente condición iterativa FOR - NEXT.
NOT()	Complemento a uno.
NULL	Establece cuenta nula después del <i>Carriage Return-Line Feed</i> .
ONERR	En caso de error salta a otra línea de programa.
ON GOTO	GOTO condicional.
ON GOSUB	GOSUB condicional.
ONTIME	Genera una interrupción cuando el tiempo es mayor ó igual al argumento de la función.
PH0	Imprime en modo hexadecimal con supresión de cero.
PH1	Imprime en modo hexadecimal sin supresión de cero.
PH0 . #	PH0 a la impresora en línea.
PH1 . #	PH1 a la impresora en línea.
PI	Constante PI (3.1415926).

MNEMONICO	DESCRIPCION
POP	Operación POP sobre el stack de argumentos.
PRINT	Imprime variables, strings ó literales.
PRINT #	Imprime al puerto serial.
PRINT CR	PRINT I,CR
PRINT SPC(#)	PRINT "A",SPC(5),"B"
PRINT TAB(#)	PRINT TAB(5),"X"
PRINT USING (Fx)	Imprime usando formato de punto flotante.
PRINT USING (#,#)	Imprime usando formato entero-fracción.
PROG	Salva el programa en EPROM.
PROG1	Salva la velocidad en BAUDS en EPROM.
PROG2	Salva la velocidad en BAUDS en EPROM y ejecuta el programa después del RESET.
PUSH	Operación PUSH sobre el stack de argumentos.
RAM	Invoca modalidad RAM.
READ	Lee datos de la instrucción DATA.
REM	Remarca.
RESTORE	Restablece el apuntador de lectura.
RETI	Regresa de interrupción.
RETURN	Regresa de subrutina.

MNEMONICO	DESCRIPCION
RND	Número aleatorio.
ROM#	Invoca modalidad ROM.
RUN	Ejecuta un programa.
SGN	Signo.
SIN()	Regresa el seno del argumento.
SQR()	Raíz cuadrada.
ST@	Almacena variable.
STOP	Detiene la ejecución del programa.
STRING#,#	Asigna memoria para strings.
TAN()	Regresa la tangente de argumento.
TIME	Obtiene y/o asigna valor al reloj de tiempo real.
UNTIL	Prueba la condición iterativa de la instrucción DO.
WHILE	Prueba la condición iterativa de la instrucción DO.
XFER	Transfiere un programa desde ROM/EPROM hacia RAM.
+	Adición.
/	División.

MNEMONICO	DESCRIPCION
**	Exponenciación.
*	Multiplicación.
---	Sustracción.
.AND.	AND lógico.
.OR.	OR lógico.
.XOR.	OR exclusivo.

*El conjunto de CALLs de mayor relevancia será expuesto en la siguiente sección.

B.5.2 Instrucción CALL.

La instrucción **CALL** [entero] se emplea para llamadas a programas de aplicación especialmente escritos para el módulo BASIC. Las llamadas específicas por número de mayor importancia son las siguientes.

CALL 3: Update-Block-Transfer-Write.

Esta rutina transfiere el *buffer* de escritura de transferencia de bloques (**BTW**) del procesador auxiliar en el módulo BASIC hacia el *buffer* **BTW** del BASIC. Si no existe transferencia de bloques dentro de dos segundos, la rutina regresa al BASIC sin nuevos datos.

CALL 4: Block-Transfer-Write-Length.

Establece el número de palabras (1-64) a transferir entre el módulo BASIC y el procesador *host*.

CALL 5: Block-Transfer-Read-Length.

Establece el número de palabras (1-64) a transferir entre el módulo BASIC y el procesador *host*.

CALL 11: 16 BIT Binary to Internal Floating Point.

El argumento de entrada es el número (1-64) de la palabra en el *buffer* de transferencia de bloques a ser convertido de 16 *bits* a formato interno. El argumento de salida sobre el *stack* es el valor convertido a formato interno.

CALL 21: Internal Floating Point to 16-Bit Unsigned.

Esta rutina toma un valor entre 0-65535, lo convierte a su representación binaria y lo almacena en una palabra en el *buffer* de transferencia de bloques de lectura.

CALL 35: Get Numeric Input Character From Peripheral Port.

Obtiene el carácter presente en el *buffer* de entrada de 255 caracteres del puerto periférico. Regresa la representación decimal de los caracteres recibidos como argumento de salida. El puerto periférico recibe los datos transmitidos por el dispositivo al cual esta conectado y los almacena en este *buffer*.

CALL 36: Get the Number of Characters in the Peripheral Port Buffer.

Obtiene el número de caracteres en el *buffer* seleccionado como argumento de salida. Se debe efectuar un *PUSH* sobre el *stack* para seleccionar el *buffer* a ser examinado.

PUSH 1 para el *buffer* de entrada.

PUSH 0 para el *buffer* de salida.

Se requiere efectuar un *POP* para obtener el número de caracteres.

CALL 37: Clear the Peripheral Ports Input or Output Buffer.

Esta rutina limpia cualquiera de los *buffers* del puerto periférico, entrada o salida.

PUSH 0 para limpiar el *buffer* de salida.

PUSH 1 para limpiar el *buffer* de entrada.

PUSH 2 para limpiar ambos *buffers*.

CALL 44: Date Retrieve Numeric.

Regresa la fecha actual en el *stack* de argumentos como tres números. No hay argumento de entrada para esta rutina. La fecha se obtiene efectuando *POPs* para el día, mes y año.

CALL 46: Time Retrieve Numeric.

Regresa la hora del reloj en el *stack* de argumentos como tres números. No hay argumento de entrada para esta rutina. La hora se obtiene efectuando *POPs* para la hora, minutos y segundos respectivamente.

CALL 63: String To Number Conversion.

Convierte el primer número decimal encontrado en el *string* especificado a su valor numérico.

Intelligent I/O Modules

Peripheral Interface Modules

Catalog No.	Module Description	Electrical Interface	Baud Rate Compatibility	Backplane Power	Slot
1771-DA	ASCII Interface Module	1 Port Configurable for: - RS-232-C - A-B Long Line - Current Loop	110/300/600 1200/2400/4800/9600	1.3 A	1
1771-DB	BASIC Module	1 Program Port Configurable for: - RS423-A/232 Compatible 1 Peripheral Port Configurable for: - RS422 or - RS423-A/232 Compatible	300/600/1200/2400 4800/9600/19200	1.5 A	1

TABLA B-1

**Apéndice C. Terminal
Industrial T-35**

Apéndice C. Terminal Industrial T-35.

C.1 Especificaciones Técnicas.

En la tabla C-1, *Operator Interfase*, ver la columna correspondiente a la información técnica de la Terminal T-35.

C.2 Herramientas de Programación.

El programa fuente de la Interfase para Supervisión y Control, BIKO, fué codificado en lenguaje C. A continuación se listan el compilador y utilerías empleadas para su desarrollo:

1. Microsoft C Optimazer Compiler Ver. 5.1

Microsoft Corp.

2. C Utility Library Ver. 4.0

Essential Software Inc.

1988.

3. Screen-Maker Ver. 1.0

MultiSystem Development Inc.

1988.

	T30 Operator Interface Series A		T35 Terminal	Advisor PC	Advisor 2-	T50 Terminal
PC SYSTEM COMMUNICATIONS						
PC User PLC Program Upload/Download	N/A	N/A	Options - KTK+ or RS-232 to KX2 Series B w/ Software	N/A	N/A	KTK1 (PC), Options RS-232 to KX2 Series B w/ Software
Data Transfer	Option of F30F to 1770-K2 Series B	Optional F30F to 1770-K2 Series B	Options - KTK1 or RS-232 to KX2 Series B w/ User Software	Software Support RS-232 to KX2 Series B	RS-232 port to KX2 Series B	Options - KTK1 or RS-232 to KX2 w/ Software
On-line Programming & Upload/Download	N/A	N/A	Option w/PTS	N/A	N/A	w/AD200 Software & KTK
Parallel Ports	F30C Option 1 5-24V DC (DC I/O only)	F30C Option 1 5-24V DC (DC I/O only)	6174-KX5 - 1 port 6174-MX5 - 2 ports Centronics Parallel Standard	6174-KX5 - 1 port 6174-MX5 - 2 ports Centronics Parallel Standard	N/A	Centronics Standard for Printer
Data Highway Diagnostic Display	N/A	N/A	USD	Yes	Yes	USD
Non-Networking PLC System Connects for Data Transfer	F30A/F30B RS232 to 1771-06, DA or 1775-GA F30C Parallel port to 1771-B/O8 or 3D/O8D F30D Direct Remote I/O F30F Direct Data Highway-Peer Comm Link		N/A	N/A	N/A	N/A
SOFTWARE FEATURES						
Archiving of Alarms to Disk	N/A	N/A	USD	Yes to Hard Disk	Yes M10 Disk	USD
Archiving Operator Commands	N/A	N/A	USD	Yes to Hard Disk	Yes M10 Disk	USD
Background Alarming	No	No	USD	256 Standard or 1000 w/2 Mb DM81 board	1000 Points	USD
Alarm Summary	User Application Program	User Application Program	USD	240 Alarms	240 Alarms	USD
Windows (Zooming)	No	No	USD	5 View Ports/ Screens, 4 Levels/ View Port	5 View Ports/ Screens, 4 Levels/ View Port	USD
Process Facets/ Library	No	No	USD	Yes	Yes	USD
Process Graphical Symbol Library	31 Symbols	36 Symbols	USD	Yes	Yes	USD
Max No. Active Points Per Display	Not Limited	Not Limited	USD	300	300	USD
Data Logging	No	No	USD	6171: DL Software	Yes	USD
USER SYSTEM CONFIGURATION SOFTWARE						
Number User Display Screens	255	255	USD	255	255	USD
No. Personnel Identif. (Security Passwords)	1	1	USD	32	32	USD
No. Access Levels for PIN Codes	No	No	USD	15	5	USD
Number of Database Tags	N/A	N/A	USD	4000	8000	USD
Number of Colors	Monochrome	Monochrome or 8 Basic Colors	16 w/mouse 8 w/out mouse	16 w/mouse 8 w/out mouse	8 colors	USD
ALTERNATE SUPPORTING SOFTWARE						
A-B/CAN/M 6200 Software Advisor PC	N/A	N/A	Yes Yes Yes	N/A	N/A	Yes Yes Yes

USD = User Software Dependent
 USD = User Hardware Dependent

TABLE C-1

Operator Interface

	130 Operator Interface Series A	130 Operator Interface Series B	135 Terminal	Advisor PC	Advisor 3+	150 Terminal
SYSTEM HARDWARE & SOFTWARE						
Operator Control Panel	NEMA 4 Front Panel, 25 Position Keyboard	Same as Series A with 10 added function keys, Option 1-EVMA 4x	BM AT Compatible, Optional 6171-CPI4	6172-CPI4 Serial or AT Compatible	6172-CPI4 Serial or 6173-CPI2 Parallel	Industrial BM AT Compatible
Multi-tasking User Background Programs	No	No	USD	Yes	Yes	User dependent hardware & software
CPI Character Resolution	25 Rows x 80 Columns	25 Rows x 80 Columns	75 Rows x 80 Columns	8M EGA High Res. (43 lines x 80 columns)	51 Lines	25 Rows x 80 Columns
CPI Size	12" Amber 9" Green	12" Amber 12" Color	12" Color	13" or 19"	19" or 19"	9" Amber
Touchscreen	No	No	No	Option	Option	No
Auxiliary Monitor	No	No	RGB Output	USD	Yes	Composite Video & RGB Outputs
Operating System	Dedicated	Dedicated	MS-DOS 3.2	ChronOS/DOS	CPM	MS-DOS Rev. 3.2
B & W Printer	Serial RS-232 Program Listing Messages	Serial RS-232 Program Listing Messages	Centronics Parallel or Serial RS-232	Supplies Centronics Parallel	Serial RS-232	Serial RS-232 Centronics compatible
Color Printer	No	No	IBM Color Jet	BM Color Jet	Printron 57024C	BM Color Jet
Program Archiving	SA or SB Recorder or 150 (w/Cross Talk)	SA or SB Recorder or 150 (w/Cross Talk)	3.5" floppy & Hard Disk	User Hard Disk	PRGM or M10	3.5" floppy & Hard Disk
SYSTEM MEMORY						
User Screen Memory	40K bytes or 128K bytes	40K bytes or 128K bytes	1 Meg RAM & Hard Disk	User Hard disk	256K bytes thru 2.5 M bytes PCMCIA	1 Meg RAM & Hard Disk
Database Memory	500 data registers	User configurable data registers	1 Meg RAM & Hard Disk	256K RAM Max.	512K RAM	1 M Bytes RAM
Hard Disk Support	No	No	20 MB Hard Disk	Same as User DOS	22.5 MB Hard Disk	20 MB Hard Disk
Floppy Diskette Support	No	No	On-Board Drive 720 K Bytes	Same as User DOS	supported on M10 Drive	On-Board Drive 3.5/720 KB
PROGRAM DEVELOPMENT						
Program Development Keyboard	WAA Keyboard	1784-WAA Keyboard or BM-XT Keyboard	WAB or BM-AT or -XT Keyboard	BM-AT Compatible	6172-KEY2	AT Keyboard
Graphic Language	Video Interfacing Programming (VIP)	Video Interfacing Programming (VIP)	EGA	GRAPHIC and High Resolution Option	GRAPHIC	USD
Mouse Graphics	No	No	EGA	Optional 6171-M5 Software	No	USD
Pixel (Circle/Ellipse) or Character Based	Character	Character	640 x 350 pixels (EGA)	Character or w/ mouse 640 x 350 pixels	Character	Character or 640 x 200 Pixels with EGA option
PC SYSTEM COMMUNICATIONS						
Serial Port (Baud Rate)	Integral RS-232 300 - 19.2K baud	Integral RS-232 300 - 19.2K baud	Integral RS-232 300 - 9,600 baud	6174-MX5 & Ports 6174-MX8 & Ports	4 Ports, 19.2K baud	Integral RS-232 300 to 19.2K baud
Data Highway User PLC Program Upload/Download	N/A	N/A	RS-232 to KE/KF & 6200 Software	N/A	N/A	RS-232 to KE/KF & 5203 Software
Data Transfer	Optional F30F to 1771-KE/KF/KG 1770-KF2 1775-KA (Modem)	Optional F30F to 1771-KE/KF/KG 1770-KF2 1775-KA (Modem)	RS-232 to KE/KF & User Software	Software Supports 1771-KE/KF/KG 1770-KF2 1775-KA (Modem) thru RS-232	Software Supports 1771-KE/KF/KG 1770-KF2 1775-KA (Modem) thru RS-232	RS-232 to KE/KF & User Software
Peer Communication Line (PCL) Port	Optional F30F to K2 Series B	Optional F30F to K2 Series B	Option w/PT5 or KTM	RS-232 to Optional K2 Series B	RS-232 to Optional K2 Series B	Integral Port

USD = User Software Dependent
 USD = User Hardware Dependent

TABLE C-1

**Apéndice D. Mapas de
Correspondencia entre las
Tablas de Datos del PLC 5-15,
Módulo Basic 1771-DB y la
Terminal T-35**

Apéndice D. Mapas de Correspondencia Entre las Tablas de Datos del PLC 5-15, Módulo BASIC 1771-DB y la Terminal T-35.

Con efectos de establecer la comunicación entre el PLC y la Terminal Industrial, se ha establecido una correspondencia de datos entre los tres elementos de *hardware* involucrados: El PLC 5-15, El Módulo BASIC y la Terminal Industrial T-35.

Observaciones: La tabla de datos correspondiente al Módulo BASIC es el arreglo TX() del programa BASIC. La tabla de datos correspondiente a la Terminal T-35 es el arreglo tabla_datos[] definido en el programa fuente C.

D.1 Tabla de Datos del PLC 5-15 por Archivos.

ARCHIVO 0-0

No DE PALABRAS	RANGO DE DIRECCIONES	USO
32	11:00/00 - 11:37/17	Imagen de entradas

ARCHIVO N - 7

No DE PALABRAS	RANGO DE DIRECCIONES	USO
23	N7:0 - N7:22	PID Zona 1HR
23	N7:23 - N7:45	PID Zona 2HR
23	N7:46 - N7:68	PID Zona 3HR
23	N7:69 - N7:91	PID Zona 1HT
23	N7:92 - N7:114	PID Zona 2HT
37	N7:115 - N7:151	Config. 1771-IFE (PT1 y PT2)
6	N7:152 - N7:157	Datos del 1771-IFE
13	N7:158 - N7:170	Config. y datos del 1771-OFE2 (0:02), zonas: 2HR y 3HR
13	N7:171 - N7:183	Config. y datos del 1771-OFE2 (0:01), zonas: 1HT, 2HT y 1HR
27	N7:184 - N7:210	Config. 1771-IXE
12	N7:211 - N7:222	Status y datos 1771-IXE
64	N7:223 - N7:286	Datos a T-35 vía 1771-DB y BTW del PLC 5-15
64	N7:287 - N7:350	Datos de T-35 vía 1771-DB y BTR del PLC 5-15
5	N7:501 - N7:505	BTW (Config. del 1771-IFE)

CONTINUACION ARCHIVO N - 7

No DE PALABRAS	RANGO DE DIRECCIONES	USO
5	N7:511 - N7:515	BTW (Configuración del 1771-IFE)
5	N7:516 - N7:520	BTR (Datos del 1771-IXE)
5	N7:521 - N7:525	BTW (Configuración y datos del 1771-OFE2) (0:02)
5	N7:526 - N7:530	BTW (Configuración y datos del 1771-OFE2) (0:01)
5	N7:531 - N7:535	BTW (Hacia T-35 vía 1771-DB)
5	N7:536 - N7:540	BTR (De T-35 vía 1771-DB)

ARCHIVO B-3

No DE PALABRAS	RANGO DE DIRECCIONES	USO
1	B3:0 (B3/0 - B3/15)	Comandos auto. HT
1	B3:1 (B3/16 - B3/31)	Comandos auto. Cesta
1	B3:3 (B3/48 - B3/63)	Status posiciones HT
1	B3:4 (B3/64 - B3/79)	Status posiciones Cesta
1	B3:5 (B3/80 - B3/95)	Comandos auto. HR
1	B3:7 (B3/112 - B3/127)	Status posiciones HR

D.2 Relación de Datos Transferidos al Módulo BASIC desde el PLC 5-15

RELACION MODULO BASIC - PLC

No. de palabras	Datos	Dirección en PLC 5-15	Datos en Módulo BASIC
32	Status de entrada	N7:223 - N7:254	TX(1) - TX(32)
5	Temperaturas	N7:255 - N7:259	TX(33) - TX(37)
5	Apertura de válvulas	N7:260 - N7:264	TX(38) - TX(42)
2	Contadores	N7:265 - N7:266	TX(43) - TX (44)
2	Consumo de gas	N7:267 - N7:268	TX(45) - TX(46)
1	Tipo de alarma	N7:269	TX(47)

D.3 Relación de Datos Transferidos a T-35 desde Módulo BASIC

En la T-35 existe un arreglo tabla_datos[] que corresponde al arreglo TX() en el Módulo BASIC según la relación siguiente:

El dato TX(i) corresponde al dato tabla_datos[i-1]

El criterio de transmisión de BASIC a T-35 es:

$Tx(\text{dato}) | TX(i)TR(i) = Tx(i, TX(i))$

Adicionalmente BASIC transmite:

TX(48): Día

TX(49): Mes

TX(50): Año

TX(51): Hora

TX(52): Minutos

D.4 Datos Transferidos de T-35 a PLC 5-15 via 1771-DB.

La terminal T-35 transmite mediante el comando Tx_v1(indice,dato) y el Módulo BASIC recibe el dato mediante la subrutina 400 para, después, hacer una llamada que almacena el dato en el *buffer* BTR y en la posición indicada por el índice (CALL 21). Posteriormente este *buffer* (BTR) es transferido al PLC-5/15 mediante la instrucción CALL 2.

RELACION MODULO BASIC - PLC

Indice	Dirección en PLC	Uso
1-3	N7:287 - N7:289	Comandos manuales HR
4-6	N7:290 - N7:292	Comandos manuales HT
7	N7:293	Comandos manuales Cesta
8	N7:294	Comandos manuales Tanque 1 y Tanque 2
9-12	N7:295 - N7:298	Disponibles para expansión
13-27	N7:299 - N7:313	SP, HL, y LL para cada Faceplate
28-32	N7:314 - N7:318	Valores de apertura manual de válvulas (1HT - 3HR)
33	N7:319	Comutación MAN/AUTO zona 1HT
34	N7:320	Comutación MAN/AUTO zona 2HT
35	N7:321	Comutación MAN/AUTO zona 1HR
36	N7:322	Comutación MAN/AUTO zona 2HR
37	N7:323	Comutación MAN/AUTO zona 3HR
38	N7:324	Apagado del sistema de combustión Horno de Temple
39	N7:325	Comutación MAN/AUTO de los

CONTINUACION RELACION MODULO BASIC - PLC

Indice	Dirección en PLC	Uso
41	N7:327	Tiempo de inmersión en Tanque 1
42	N7:328	Tiempo de inmersión en Tanque 2
43	N7:329	Apagado del sistema de combustión del Horno de Revenido
44	N7:330	Conmutación MAN/AUTO de los movimientos del Horno de Temple
45	N7:331	Tiempo entre ciclos del Horno de Revenido

BIBLIOGRAFIA

1. Ogata, Katsuhiko.

INGENIERIA DE CONTROL MODERNA

Prentice-Hall Hispanoamericana

1980

2. Gilbert and Llewellyn.

PROGRAMMABLE CONTROLLERS, PRACTICES AND CONCEPTS

Industrial Training Corporation

1985

3. Kompass, E.J.

CHANGING CONTROL TECHNOLOGY: THE PAST AND FUTURE

Control engineering magazine, Vol. 36 No. 12

Octubre 1989

4. Ledgerwood, B.K.

THE BUSINESS OF CONTROL

Control engineering magazine, Vol. 36 No. 12

Octubre 1989

5. Fairley, Richard.

INGENIERIA DE SOFTWARE

Mc Graw Hill de México.

1987

6. Pressman, Roger S.

SOFTWARE ENGINEERING

Mc Graw Hill de México.

1988

7. Kernighan, Brian W.

EL LENGUAJE DE PROGRAMACION C

Prentice Hall Hispanoamericana.

1986

REFERENCIAS

1. Allen-Bradley

PLC 5-15 FAMILY PROGRAMMABLE CONTROLLERS

PROCESSOR MANUAL

Allen-Bradley

1989

2. Allen-Bradley

BASIC MODULE, USER'S MANUAL

Allen-Bradley

1989

3. Microsoft Corp.

MICROSOFT C VER. 5.1

OPTIMIZING COMPILER, USER'S GUIDE

Microsoft Corp.

1989

4. Microsoft Corp.

MICROSOFT C VER. 5.1

OPTIMIZING COMPILER, LANGUAGE REFERENCE

Microsoft Corp.

1989

5. Microsoft Corp.

MICROSOFT C VER. 5.1

OPTIMIZING COMPILER, RUN-TIME LIBRARY REFERENCE

Microsoft Corp.

1989

6. Microsoft Corp.

MICROSOFT C VER. 5.1

OPTIMIZING COMPILER, MIXED LANGUAGE PROGRAMMING
GUIDE

Microsoft Corp.

1989

7. Essential Software, Inc.

C UTILITY LIBRARY, VER. 4.0

USER'S MANUAL

Essential Software, Inc.

1988

8. MultiSystem Development, Inc.

SCREENMAKER-T USER'S MANUAL

MultiSystem Development, Inc.

1987