



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

CONSTRUCCION FISICA DE UN PLC PARA AUXILIO DIDACTICO

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION

PRESENTA:

MARIA LEONOR SALCEDO UBILLA



DIRECTOR DE TESIS:

M. I. ANTONIO SALVA CALLEJA

CD. UNIVERSITARIA MEXICO D. F.

2005

m. 339869



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

AUTORIZO a la Dirección General de Bibliotecas de la UNAM a digitalizar en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: MARTA LEONOR SALCEDO UTELLA

FECHA: 12-ENERO-2005

FIRMA: [Firma manuscrita]

AGRADECIMIENTOS

A MÍ MADRE LA SRA. ROSALIA UBILLA DÍAZ
POR CUIDARME, ACONSEJARME Y AMARME.

A MÍ TÍO EL DR. JUAN UBILLA DÍAZ
QUIEN SE HA PREOCUPADO POR MÍ COMO UN PADRE.

A MÍ TÍO EL DR. SERGIO UBILLA DÍAZ
POR EL MEJOR EJEMPLO DE ÉTICA Y TRABAJO QUE HE TENIDO.

A MIS HERMANOS; ANGELICA, ROSALIA, IVON, MARTA Y MAURICIO
POR SU SOLIDARIDAD EN LOS MOMENTOS MÁS DOLOROSOS DE MI VIDA.

A SERGIO ENRIQUE ELIZARRARAS RODRÍGUEZ
POR BRINDARME SU APOYO PROFESIONAL
Y SOBRE TODO POR DARMER SU AMOR
QUE ES LO MÁS PRECIOSO QUE HAY EN LA VIDA.

AL ING. FRANCISCO J. RODRÍGUEZ RAMÍREZ
POR ANIMARME DÍA A DÍA A TERMINAR ESTE TRABAJO.
POR OTORGARME SU AMISTAD.
POR ESCUCHARME Y ACONSEJARME SIEMPRE QUE LO HE NECESITADO.

AGRADECIMIENTOS

*A LA ING. LAURA BEATRIZ REYES MARTÍNEZ
POR SU COMPAÑÍA Y AMISTAD.*

*AL M. C. EDGAR B. AGUADO CRUZ
POR RECORDARME QUE NO ESTOY SOLA.*

*AL ING. RAFAEL FLORES GARCÍA
POR SU AMENA CHARLA Y SU AFECTO.*

*AL M. I. ANTONIO SALVÁ CALLEJA
DIRECTOR DE ESTE TRABAJO
POR SU INAGOTABLE PATIENCIA.*

*A LA UNAM POR OTORGARME
LA FORMACIÓN PROFESIONAL QUE HOY TENGO.*

DEDICO ESTE TRABAJO A LAS DOS PERSONAS QUE MÁS HE AMADO EN ESTE MUNDO:

A mamá que significa para mí el amor incondicional.

Siempre que te necesito estas con migo me cuidas, me consientes y me quieres a pesar de todo lo que pueda hacer o dejar de hacer, tu mayor esfuerzo a sido para mí y mis hermanos; para lograr que seamos personas de provecho, justas y tolerantes, no puedo afirmar que tienes los mejores hijos pero por tu esfuerzo no ha quedado, olvidas en las más de las ocasiones tu persona y tus necesidades para cuidar de tus hijos y sus necesidades, por todo esto y por mucho más eres lo mejor que tengo en esta vida.

A Sergio E. Elizarraras Rodríguez.

El tiempo que Dios me concedió a tu lado pasamos por variados problemas, por muchas dudas y superamos todo paso a paso, permanecemos juntos a pesar de tus rarezas y de las mías nuestra relación se hizo más fuerte con el paso del tiempo no conocimos la decadencia del desamor, la muerte nos separo y esto es algo que yo pensaba ocurriría en 30 años o más, no en este tiempo yo deseaba partir antes que tú para no tener que soportar el dolor de perderte y me miro hoy viviendo con tu ausencia y esto me es muy difícil, a la fecha en que se imprime este trabajo te sigo amando aunque ya no estés físicamente, doy gracias a Dios por a verte conocido, a través de ti conocí lo que es amar y sentirse amada y por ese hecho soy muy afortunada y soy desdichada porque solo fueron 6 años 7 meses y 11 días los que estuviste con migo, el 26 de Julio del 2004 a las 17:45 te marchaste de este mundo, me enseñaste muchas cosas lllore por ti en muchas ocasiones, pero también me hiciste inmensamente feliz como nadie más en este mundo, soñaba con estar muchos, muchos años a tu lado; tenias muchos planes querias hacer muchas cosas yo sé que deseabas quedarte aquí de este lado, no puedo aun asimilar el que ya no estés y no sé en que momento lo pueda lograr; resuena en mi cabeza tu risa, cada día espero verte llegar a la Universidad, extraño tu abrazo, tus manos, tu mirada, tu cabello, tus pasos, tus enojos, tus caprichos y ya no hay más, simplemente no estas; me consuelo pensando que eres feliz en donde estas ahora y que en algún tiempo nuevamente te veré cuando llegue mi momento.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1	
1 ESTRUCTURA BÁSICA DEL PROGRAMADOR LÓGICO MODULAR	4
1.1 ESTRUCTURA BÁSICA DEL HARDWARE DEL PLM	4
1.1.1 Computadora Central (CC)	5
1.1.2 Bloque de Entradas (BE)	6
1.1.3 Bloque de Salidas (BS)	7
1.1.4 Bloque de Comando Local y Despliegue (BCLD)	7
1.1.5 Fuente de Alimentación (FA)	8
1.2 MÓDULOS LÓGICOS	9
1.2.1 Descripción general de los módulos lógicos	10
1.2.2 Variables booleanas en el PLM	12
CAPÍTULO 2	
2 COMPUTADORA CENTRAL, TARJETA FACIL_11	14
2.1 CARACTERÍSTICAS PRINCIPALES	14
2.2 CIRCUITERÍA	15
2.2.1 Microcontrolador 68HC11F1	16
2.2.2 Subpaginador de puertos de la CMT FACIL_11	17
2.2.3 Convertidor TTL – RS 232	18
2.2.4 Lógica M – I (Motorola – Intel)	18
2.2.5 Puertos paralelos adicionales de la FACIL_11	18
2.2.6 Memorias EPROM	19
2.2.7 Circuitería para programación de memorias EPROM	22
2.2.8 Memoria RAM	23
2.2.9 Localización de componentes en la CMT FACIL_11	25
2.3 ESPECIFICACIONES DE CONFIGURACIÓN DE MAPAS DE MEMORIA Y PUERTOS DE LA TARJETA FACIL_11	28
2.3.1 Mapas de memoria de la CMT FACIL_11 operando en modo expandido	30
2.3.2 Ejecución autónoma desde EPROM	33
2.3.3 Submapa de puertos	34
2.3.4 Submapa de puertos alterno	35
CAPÍTULO 3	
3 CIRCUITOS DE ENTRADA, DE SALIDA Y CIRCUITOS AUXILIARES	37
DISEÑO DE CIRCUITOS IMPRESOS ASOCIADOS	
3.1 DESCRIPCIÓN DE LOS CIRCUITOS DE ENTRADA	37
3.1.1 Descripción general	37
3.1.2 Descripción particular para una única entrada	38
3.1.3 Organización del bloque de entradas	44
3.2 DESCRIPCIÓN DE LOS CIRCUITOS AUXILIARES	45
3.2.1 Puertos paralelos adicionales	46

ÍNDICE

3.2.2	Bloques funcionales asociados con el BCLD	48
3.3	PRUEBA DE LOS CIRCUITOS DE ENTRADA Y AUXILIARES	54
3.3.1	Objetivo de armar los circuitos en tarjetas experimentales	55
3.4	DISEÑO DEL CIRCUITO IMPRESO PARA LA TARJETA PERIFÉRICA 1	55
3.4.1	Dimensiones de la tarjeta de circuito impreso	55
3.4.2	Integración de los componentes en la tarjeta	56
3.4.3	Selección del software para generar el dibujo de circuito para la Tarjeta Periférica 1	59
3.4.4	Elaboración del diseño del circuito impreso	60
3.5	CIRCUITOS DE SALIDA	62
3.5.1	Conceptos generales sobre relevadores	62
3.5.2	Descripción del circuito particular para una única salida	65
3.5.3	Organización de los circuitos de salida	66
3.6	PRUEBA DE LOS CIRCUITOS DE SALIDA	66
3.7	DISEÑO DEL CIRCUITO IMPRESO PARA LA TARJETA PERIFÉRICA 2	67
3.7.1	Dimensiones de la Tarjeta Periférica 2	67
3.7.2	Integración de los componentes en la tarjeta	68
3.7.3	Selección del software	71
CAPÍTULO 4		
4	GABINETE DEL PLM	72
4.1	DISEÑO DEL GABINETE PARA EL PLM	72
4.1.1	Utilidad de construcción del gabinete	72
4.1.2	Facilidad de manejo del instrumento para el usuario final	73
4.1.3	Facilidad para efectuar el ensamble del PLM	75
4.1.4	Dimensiones físicas para el diseño del gabinete	78
4.1.5	Costo de fabricación del gabinete	80
4.1.6	Costo de fabricación del prototipo.	80
CAPÍTULO 5		
5	EJEMPLO DE APLICACIÓN	82
5.1	CARACTERÍSTICAS BÁSICAS DEL LENGUAJE SIIL 1	82
5.1.1	Formato de instrucciones en SIIL 1	83
5.1.2	Características generales de la ejecución de un programa SIIL 1 en la CC del PLM	84
5.1.3	Formato de un programa fuente en SIIL	86
5.1.4	Software de generador de código objeto (SFCO)	87
5.2	DESCRIPCIÓN DEL PROCESO A SER AUTOMATIZADO UTILIZANDO EL PLM	87
5.3	SECUENCIA DE ARRANQUE PARA UN MOTOR TRIFÁSICO UTILIZANDO EL PLM	88
CONCLUSIONES		91
BIBLIOGRAFÍA		93

ÍNDICE

APÉNDICE A	
A FUENTE DE PODER	94
A.1 DISEÑO DE LA FUENTE DE PODER	94
A.1.1 La carga	95
A.1.2 Regulación de voltaje	95
A.1.3 Etapa de filtrado	97
A.1.4 Etapa de rectificación	100
A.1.5 El transformador	101
A.2 ESQUEMA DE LA FUENTE DE ALIMENTACIÓN	103
A.3 DISEÑO DEL CIRCUITO IMPRESO PARA LA FUENTE DE ALIMENTACIÓN	104
APÉNDICE B	106

INTRODUCCIÓN

El concepto de control es extraordinariamente amplio, abarcando desde un simple interruptor que gobierna el encendido de una bombilla o el grifo que regula el paso de agua en una tubería, hasta el piloto automático de un avión.

Podríamos definir el control como la manipulación indirecta de las magnitudes de un sistema denominado planta a través de otro sistema llamado sistema de control. Los primeros sistemas de control se desarrollaron con la revolución industrial a finales del siglo XIX y principios del siglo XX. Al principio, se basaron casi exclusivamente en componentes electromecánicos, básicamente engranajes, palancas, relevadores y pequeños motores, pero a partir de los años cincuenta empezaron a emplearse los semiconductores, que permitían el diseño de sistemas de menor tamaño y consumo, más rápidos y con menor desgaste.

En la década de los setenta, la complejidad y las prestaciones de los sistemas de control se incrementaron gracias al empleo de circuitos integrados y en particular los de tipo programable (sistemas basados en microprocesadores). La demanda en la industria de un sistema económico, robusto, flexible, fácilmente modificable y con mayor facilidad para tratar con tensiones y corrientes fuertes que las que tenía una computadora, hizo que se desarrollasen los controladores lógicos programables (PLC).

Un Control Lógico Programable (PLC, del inglés Programmable Logic Controller) procesa señales binarias de entrada y las convierte en señales de salida; con éstas se pueden controlar directamente secuencias mecánicas, procesos fabriles totales o parciales, etcétera.

Los posibles campos de aplicación de un PLC son casi innumerables. Los Controles Lógicos Programables se utilizan principalmente para las siguientes funciones:

Control de procesos; en esta función, el PLC se encarga de que cada paso o fase del proceso sea efectuado en el orden cronológico correcto y sincronizado.

Visualización de instalaciones; en este caso, el PLC verifica automáticamente ciertas condiciones de la instalación (por ejemplo temperaturas, presiones, niveles). Cuando en su comprobación, el control registra un exceso en los coeficientes máximos o mínimos de los

parámetros, actúa de dos formas; adopta las medidas necesarias para evitar deterioros o desperfectos, o emite señales de aviso para el personal de servicio.

Control de puesta a punto para máquinas; las máquinas herramientas modernas casi siempre están dotadas de un control numérico computarizado (CNC). El tornero o fresador ya no pone a punto su máquina ajustando manivelas y tornillos. En lugar de ello, programa un control numérico computarizado. Este se encarga entonces de realizar automáticamente los ajustes precisos para trabajar la pieza correspondiente. Pero para que el CNC y la máquina herramienta se "entiendan", es preciso integrar un PLC, que se encarga de la comunicación entre ambos equipos.

El presente trabajo fue realizado con el objetivo de realizar la construcción física de un modelo robusto para el Programador Lógico Modular (PLM), que es un prototipo de un Controlador Lógico Programable (PLC).

El alcance de este trabajo es desarrollar el hardware para el prototipo del PLM; que cuente con las siguientes características; robusto, de fácil manejo, económico y mantenimiento sencillo.

El PLM es un dispositivo electrónico basado en una computadora monotaquilla que se compone de cuatro puertos de entrada con ocho entradas por puerto y dos puertos de salida con ocho salidas por puerto, se encuentra conformado por los siguientes cinco bloques funcionales: Computadora Central (CC), Bloque de Entradas (BE), Bloque de Salidas (BS), Bloque de Comando Local y Despliegue (BCLD), y Fuente de Alimentación (FA); el PLM esta orientado a la realización de diversos bloques funcionales típicos de aplicaciones de control lógico, como son: compuertas lógicas, temporizadores, contadores de eventos y secuenciadores de estados; en la notación utilizada para el PLM se le llama módulo lógico a un bloque de los mencionados anteriormente y es realizado virtualmente por software ejecutable en el microcontrolador 68HC11F1, que gobierna el funcionamiento del dispositivo.

Para desarrollar el proyecto en primer termino se probaron en tarjeta experimental los circuitos vinculados con las entradas y salidas del sistema, empleando para ello software para desarrollo con microcontroladores diseñado por el director de tesis. Posteriormente se procedió con diseño de los circuitos impresos asociados y su liga con la arquitectura de microcontrolador que se empleó como computadora maestra del sistema. Enseguida a lo anterior se diseñó el gabinete y la

fuelle de alimentación, por último se realizó la integración mecánica de todos los componentes hasta aquí mencionados.

La organización de esta tesis es la siguiente; el trabajo está constituido por la introducción, cinco capítulos, conclusiones y dos apéndices:

En la introducción se determina el objetivo del trabajo, se expresa como se encuentra organizado el diseño del modelo y cuáles fueron los pasos que se efectuaron para llegar a presentar prototipo como un producto terminado.

En el capítulo 1, se hace una descripción general de los cinco bloques funcionales que componen el hardware del PLM, también se especifican los módulos lógicos que puede realizar el dispositivo.

En el capítulo 2, se describe la tarjeta FACIL_11 que funciona como computadora central del PLM.

En el capítulo 3, se detalla la descripción de los circuitos referentes al bloque de entradas y al bloque de salidas del PLM, así como los circuitos auxiliares, dentro de los cuales se encuentran el reloj de tiempo real, un display y los puertos auxiliares de entrada EAA y EAB, en este capítulo también se reseña el proceso que se siguió para realizar el diseño de las tarjetas de circuito impreso en donde se integran todos los componentes ya mencionados.

En el capítulo 4, se expone el proceso de diseño del gabinete del PLM así como la integración física de todos los componentes del prototipo.

En el capítulo 5 hace referencia al software empleado por el PLM, se describen las características básicas del lenguaje SIILI que es el lenguaje propio del dispositivo y se plantea un ejemplo de aplicación para demostrar el arbitraje lógico que puede realizar el PLM.

El apéndice A se refiere al diseño de la fuente de alimentación y el apéndice B trata sobre el software empleado para el diseño de los circuitos impresos que componen al PLM.

Por último se encuentran las conclusiones de este trabajo y la bibliografía empleada.

CAPÍTULO 1

ESTRUCTURA BÁSICA DEL PROGRAMADOR LÓGICO MODULAR

En este capítulo se describe de una manera general la estructura física y lógica del dispositivo denominado como Programador Lógico Modular (PLM).

Inicialmente se definen las características que corresponden al hardware del PLM planteándose la distribución de los componentes en cinco bloques funcionales, se efectúa una descripción básica de cada uno de ellos, continuando con la delineación de la lógica que puede realizar el PLM, a las diversas funciones lógicas que el dispositivo puede realizar se les denominó como Módulos Lógicos (ML), los mismos pueden ser compuertas AND, OR, temporizadores, etc. Se enumeran todos los ML que el PLC puede realizar, se efectúa una descripción generalizada para todos los Módulos Lógicos así como su forma sintáctica asociada, pasando después a la definición de los tipos de variables manejados por el PLM, concluyendo el capítulo con la mención de las características generales para la ejecución de un programa en la computadora central del PLM.

Es indispensable hacer énfasis en que el objetivo primordial de este trabajo es lo referente al hardware que constituye el PLM, sin embargo, no es posible dejar de hacer mención de varios conceptos referentes al software que emplea el dispositivo; ya que hardware y software son partes complementarias que operan conjuntamente para llevarnos a la solución de problemas relacionados con aplicaciones de control lógico.

1.1 ESTRUCTURA BÁSICA DEL HARDWARE DEL PLM.

El PLM es un dispositivo electrónico que puede usarse en el control lógico de procesos. Esta basado en una computadora monotaquilla, se compone de cuatro puertos de entrada con ocho entradas por puerto y dos puertos de salida con ocho salidas por puerto, se encuentra conformado por los siguientes cinco bloques funcionales. En la figura 1.1 se muestra la estructura a bloques del PLM.

- 1) Computadora Central (CC)
- 2) Bloque de Entradas (BE)
- 3) Bloque de salidas (BS)
- 4) Bloque de Comando Local y Despliegue (BCLD)
- 5) Fuente de Alimentación (FA)

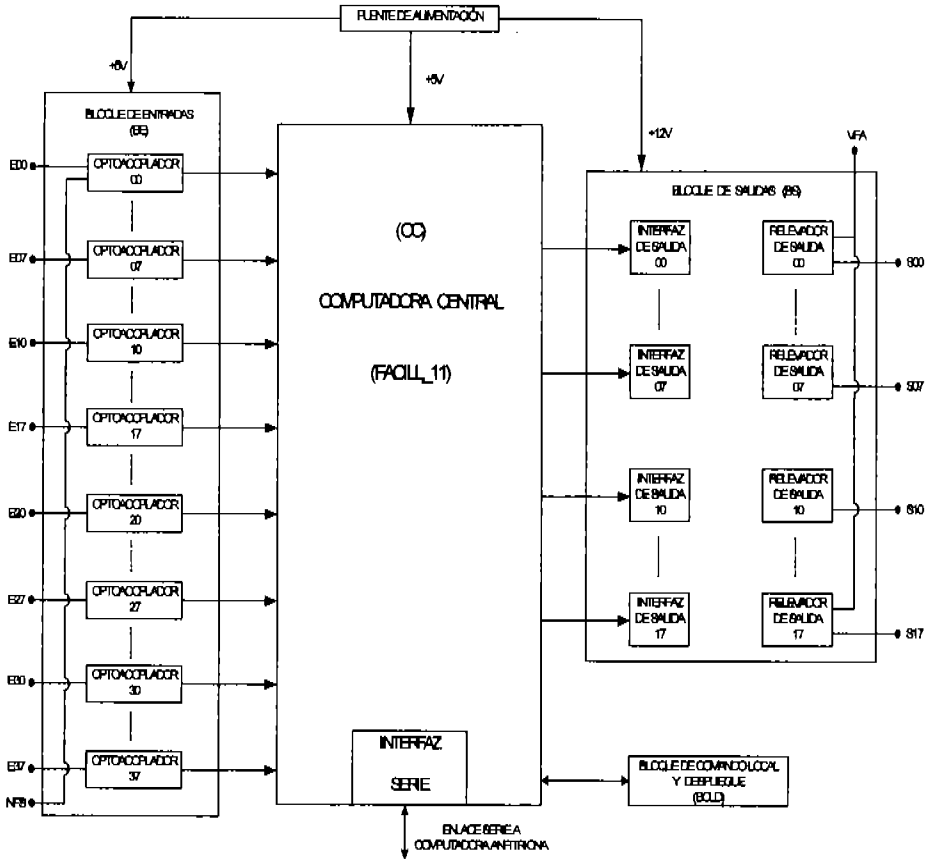


Figura 1.1.- Estructura a bloques del Programador Lógico Modular (PLM)

1.1.1 Computadora Central (CC).

La computadora central del PLM es realizada por la computadora monotañilla FACIL_11 (en la figura 1.2 se muestra la tarjeta FACIL_11) en donde el CPU es el microcontrolador 68HC11F1 fabricado por Motorola, puede operar en cualquiera de los cuatro modos asociados con el 68HC11, puede funcionar de manera autónoma o bien controlada vía serie por una computadora de tipo PC. La FACIL_11 fue desarrollada por el director de esta tesis el M. I. Antonio Salvá Calleja.

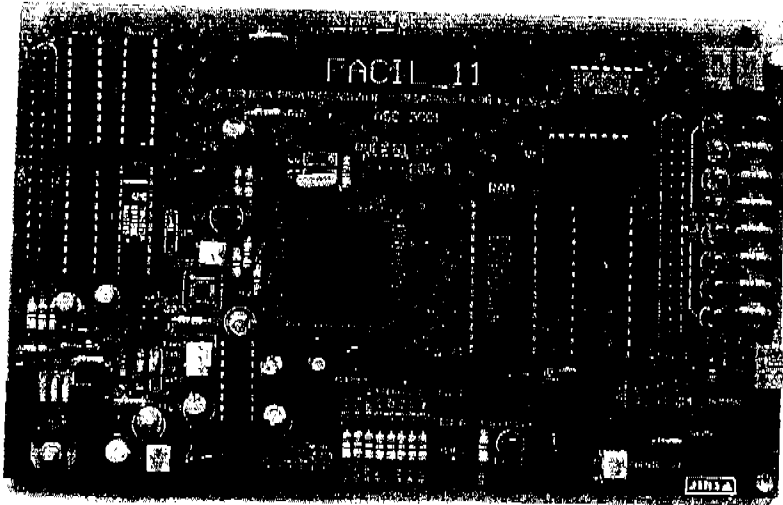


Figura 1.2.- Tarjeta FACIL_11, empleada para realizar la computadora central del PLM

La FACIL 11 como computadora central del PLM opera en el modo expandido del 6811C11, contándose en este caso con dos puertos de entrada y dos puertos de salida, con los que se realizan a nivel de la CC dos de los cuatro puertos de entradas y los dos puertos de salidas con que cuenta el PLM; para más información de la computadora monotabilla empleada, consultar el capítulo número 2 de esta tesis.

1.1.2 Bloque de Entradas (BE).

Este bloque está constituido por 32 entradas optoacopladas, distribuidas para el manejo de la computadora central en cuatro grupos de ocho entradas cada uno. Para cada bit de entrada el PLM reconoce un nivel de uno lógico cuando se presente un voltaje nominal de 24 volts medido entre la terminal correspondiente y el punto neutro de la fuente de sensores (NFS), en otro caso el nivel tomado será cero lógico. En realidad para los niveles de uno y cero lógico en las entradas corresponde un determinado intervalo de voltaje.

En el capítulo siguiente se tratara detalladamente todo lo concerniente a esta parte del hardware del prototipo.

1.1.3 Bloque de Salidas (BS).

En este bloque se concentran los dos puertos de salida con que cuenta el PLM, cada puerto está constituido por ocho bits, para cada salida se cuenta con una interfaz a un relevador de baja potencia de contactos normalmente abiertos, las terminales comunes de los 16 relevadores se encuentran conectadas al punto VFA (vivo de la fuente de actuadores) en tanto que para cada relevador el otro contacto está conectado con su correspondiente terminal de salida asociada.

Al observarse un nivel de uno lógico para una determinada salida se tiene continuidad eléctrica entre las terminales VFA y la propia correspondiente con la salida, en el caso contrario no habrá continuidad eléctrica.

1.1.4 Bloque de Comando Local y Despliegue (BCLD).

Este bloque está constituido por tres componentes, uno de ellos es la Unidad Desplegadora (UD) que maneja dos renglones de 16 caracteres, el otro es un panel que contiene cinco postes que habilitan cinco entradas binarias auxiliares, cuatro botones para comando local y un par de botones que configurarían la manera en que el PLM respondería a una reinicialización del programa del usuario, el tercer elemento que compone el bloque del BCLD es un reloj de tiempo real, que puede servir simplemente como testigo de la hora o como base de tiempo para una función especial del dispositivo, que permite generar disparos a otros módulos lógicos para horarios predeterminados por el usuario de acuerdo con las necesidades de una determinada aplicación. En la figura 1.3 se muestra esquemáticamente el bloque BCLD.

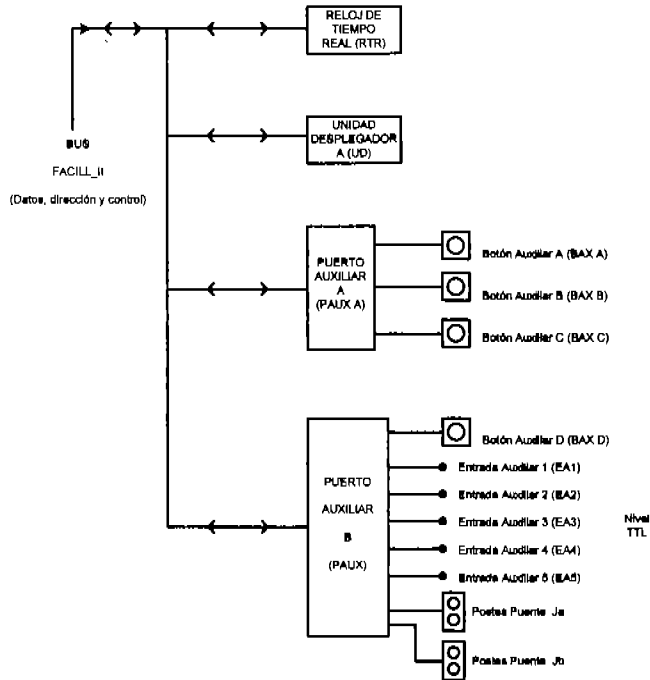


Figura 1.3.- Bloque de Comando Local y Despliegue (BCLD)

1.1.5 Fuente de Alimentación (FA).

El PLM requiere para su funcionamiento dos fuentes de voltaje, una de 12 volts para polarizar solamente los relevadores del bloque de salidas y requiere una capacidad de corriente de un Ampere y otra fuente más de 5 volts que alimenta a toda la circuitería restante del PLM y su requerimiento de corriente es 500mA.

Es necesario señalar que hasta este punto solo se hace una descripción a grandes rasgos de cada uno de los bloques ya que en capítulos siguientes se amplían los conocimientos sobre características y funcionamiento de cada una de las cinco secciones que constituyen el PLM.

1.2 MÓDULOS LÓGICOS.

El PLM, esta dirigido para realizar una variedad de bloques funcionales que suelen ser aplicados en el control lógico, como pueden ser: compuertas lógicas, temporizadores, contadores de eventos y secuenciadores de estados; dentro de la organización del PLM se denominaran como módulos lógicos (ML) a los bloques mencionados anteriormente, realizados virtualmente por software ejecutable en el microcontrolador 68HC11F1, que gobierna el funcionamiento del dispositivo.

Los módulos lógicos que el PLM puede realizar son:

- a) Compuertas AND de dos, tres y cuatro entradas.
- b) Compuertas OR de dos, tres y cuatro entradas.
- c) Compuertas NAND de dos, tres y cuatro entradas.
- d) Compuertas NOR de dos, tres y cuatro entradas.
- e) Compuertas XOR de dos, tres y cuatro entradas.
- f) Compuertas XOR negada de dos, tres y cuatro entradas.
- g) Inversores y seguidores lógicos.
- h) Cinco tipos diferentes de temporizador.
- i) Dos tipos de contadores de eventos.
- j) Secuenciadores de estado de uno a ocho bits.
- k) Flip-Flops asíncronos.

Es necesario señalar que las compuertas XOR y XOR negada se denominan respectivamente como EOR y EORN, además de que para todas las compuertas existe la posibilidad de negación para cualquiera de las entradas, contribuyendo esto a disminuir el número de módulos que se necesitan para una determinada aplicación.

El PLM puede operar de dos maneras; de manera autónoma y en modo esclavo. Al operar de manera autónoma el PLM puede realizar un sistema de control lógico, ejecutando el código correspondiente que se encontrará residente como firmware en una EPROM contenida en el mismo, en la figura 1.4 se muestra esta idea.

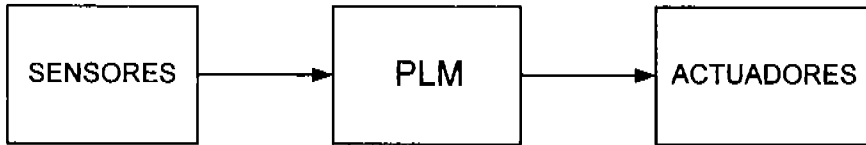


Figura 1.4.- PLM operando en forma autónoma

Cuando el PLM opera en modo esclavo el mismo se encontrará ligado vía serie a una computadora de tipo PC donde puede correrse el software que permitirá probar y depurar los programas que requiera el PLM para realizar una determinada aplicación de control lógico, en la figura 1.5 se muestra esta idea.

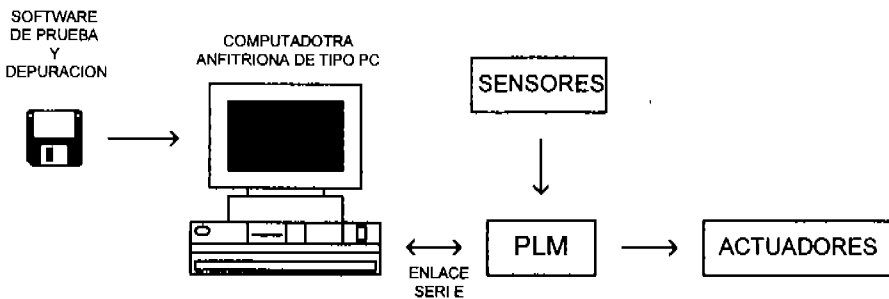


Figura 1.5.- PLM operando en modo esclavo

1.2.1 Descripción general de los módulos lógicos.

Los módulos lógicos (ML) que el PLM puede efectuar constituyen los bloques funcionales fundamentales para desarrollar aplicaciones de control lógico y pueden representarse a nivel de caja negra como se muestra en la figura 1.6; en la figura se muestra un bloque con "m" entradas y "n" salidas, los números m y n varían de acuerdo con el tipo de ML que se realice por ejemplo una compuerta AND de cuatro entradas m sería cuatro y n sería uno; en cambio para un secuenciador de estados con palabras de tamaño de cuatro bits necesita tres entradas y cinco salidas.



Figura 1.6.- Representación de un módulo lógico de m entradas y n salidas

En la figura 1.6 XE_k representa a un carácter que puede ser una de las letras E, I o S mayúscula minúscula dependiendo del tipo de variable que represente (VBE, VBI o VBS) asociada con la k-esima entrada del ML; IE_k y JE_k representan respectivamente el numero de grupo y número de bit correspondientes con la variable k-esima de entrada; XS_k representa un carácter que puede ser cualquiera de las letras I o S mayúsculas o minúsculas dependiendo del tipo de variable que este representando (VBI o VBS) asociada con la k-esima salida del ML. Por ejemplo una compuerta como la que se muestra en la figura 1.7 en donde las entradas son respectivamente las variables E01, I45 (entrada negada) y E13, la salida es la variable S02.

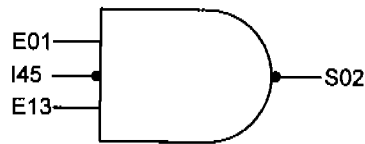


Figura 1.7.- Representación de un Módulo Lógico que realiza una compuerta NAND de tres entradas con negación en una de ellas, nótese que las entradas pueden ser de grupos diferentes

Para todos los ML que representan compuertas lógicas se tiene que responden al nivel que presentan a sus entradas; sin embargo, algunos de los ML que no son compuertas están diseñadas de modo que responden a flancos que se presenten en una o varias entradas.

Formas sintácticas asociadas con los módulos lógicos.

Cada uno de los módulos lógicos, requeridos en una determinada aplicación, deben haber sido declarados secuencialmente en un archivo de texto, para que el mismo sea procesado por una computadora anfitriona que genere el código objeto a ejecutarse en la CC de modo que los ML requeridos pueden ser realizados; además de las declaraciones asociadas con los módulos, en el

archivo mencionado se requieren colocar otras instrucciones, no relacionadas directamente con algún ML, pero necesarias para la ejecución adecuada del programa que ha de ejecutarse en el PLM.

Al conjunto de declaraciones mencionadas anteriormente se le denomina, programa fuente en lenguaje SILL1 (software de interpretación de instrucciones lógicas), que es el lenguaje propio del dispositivo, asociado con la aplicación que ha de realizar el PLM. A excepción de los módulos que requieren datos adicionales que ha de proporcionar el usuario, la forma sintáctica de las declaraciones asociadas con los mismos requiere de un solo renglón en el archivo de texto a procesar.

1.2.2 Variables booleanas en el PLM.

Las entradas y salidas que el PLM es capaz de realizar emplean variables de tipo booleano y se clasifican de la siguiente manera: variables booleanas de entrada (VBE), variables booleanas de salida (VBS), y variables booleanas intermediarias (VBI). Puesto que en el microcontrolador 68HC11 la información esta organizada en bytes, las variables aquí mencionadas están agrupadas en conjuntos de ocho variables del mismo tipo, así pues se tienen cuatro grupos de variables de entrada, dos grupos de variables de salida y por ultimo veintiún grupos de variables booleanas intermediarias; en los párrafos siguientes se plantean conceptos asociados con cada uno de los tipos de variables.

Variables Booleanas de Entrada (VBE).

Este tipo de variables se encuentran asociadas con su correspondiente entrada siendo cada una de ellas optoacoplada a la computadora central, cada terminal de entrada puede recibir una señal de voltaje (0 – 24 volts) proveniente de algún sensor que este incluido en alguna parte del sistema de control lógico que sea necesario implantar en un momento determinado. Este prototipo esta pensado como ya se ha mencionado con anterioridad para manejar 32 VBE.

Variables Booleanas de Salida (VBS).

El PLM cuenta con 16 variable booleanas de salida, cada una de ellas se encuentra ligada a un relevador de baja potencia en donde sus contactos se sierran al presentar la variable de salida correspondiente el nivel de uno lógico, si por el contrario la variable de salida presenta un nivel de

cero lógico los contactos permanecerán abiertos; las VBS están agrupadas en dos grupos de ocho salidas cada uno, en el capítulo tres de esta tesis se amplían los conceptos referentes a las salidas del PLM.

Variables Booleanas Intermediarias (VBI).

Este tipo de variables se manejan internamente y no tienen entradas o salidas físicas su función consiste en servir de enlace entre módulos lógicos cuando esto es necesario; por ejemplo, si se tiene una situación de control lógico en la que se necesitan numerosas compuertas lógicas y las salidas de algunas compuertas significan las entradas para otras; el implementar esta situación con las salidas físicas con las que contamos en el PLM sería poco práctico puesto que la cantidad con la que contamos es limitada, de ahí la necesidad de contar con otro tipo de variables, siendo éstas las variables booleanas intermedias; por supuesto que una variable booleana, que sea entrada de un módulo y salida de otro, puede ser una salida física si esto es necesario, lo que no es permitido es que una variable simultáneamente sea salida de más de un módulo.

Las VBI están organizadas en 21 grupos de ocho variables cada uno, por lo tanto el PLM tiene una capacidad de manejo de 168 variables booleanas intermedias, por otra parte la notación que se emplea para identificar este tipo de variables es la siguiente; se emplean cuatro caracteres el primero es la letra "I" que puede ser mayúscula o minúscula, el segundo y el tercer carácter representan un número entre cero y veinte denota el grupo al que pertenece la VBI y el cuarto es un dígito de cero a siete que define el número de VBI dentro del grupo; así por ejemplo, la I124 es la variable booleana intermedia que pertenece al grupo doce y es la cuarta variable dentro de su grupo.

CAPÍTULO 2

COMPUTADORA CENTRAL

TARJETA FÁCIL_11

A lo largo de este capítulo se describe la tarjeta FACIL_11, que representa el bloque funcional denominado como computadora central del PLM, iniciando con las características generales de la tarjeta, pasando después a la descripción detallada de los circuitos de la tarjeta FACIL_11 que están relacionados con el funcionamiento del PLM, terminando el capítulo con las especificaciones de configuración de mapas de memoria y puertos de la tarjeta.

2.1 CARACTERÍSTICAS PRINCIPALES.

La computadora central del PLM es realizada por la computadora monotaquilla FACIL_11 (en la figura 1.2 se muestra la tarjeta FACIL_11) en donde el CPU es el microcontrolador 68HC11F1 fabricado por Motorola, puede operar en cualquiera de los cuatro modos asociados con el 68HC11, puede operar de manera autónoma o bien controlada vía serie por una computadora de tipo PC. La FACIL_11 fue desarrollada por el director de esta tesis el M.I Antonio Salvá Calleja y tiene las siguientes características principales:

- 1) La computadora central del PLM, puede operar en cualquiera de los cuatro modos asociados con el microcontrolador 68HC11F1. Como CC del PLM opera en modo expandido, pudiéndose en este caso validar los mapas de memoria requeridos por cada una de las configuraciones de funcionamiento del PLM.
- 2) Contiene firmware interlocutor que permite manejarla, vía enlace serie, desde una computadora PC mediante la ejecución en la misma del software clásico para este fin (PCBUG11), o bien del manejador visual PUMMA_11 que corre bajo WINDOWS, que contempla los comandos de manejo más usuales además de algunos otros adicionales asociados con características propias de la tarjeta FACIL_11.
- 3) Compatibilidad con otras herramientas de software asociadas con el HC11, permitiendo esto la ejecución en la tarjeta de programas originalmente escritos en lenguaje C o ensamblador, lográndose esto mediante la carga y ejecución del archivo objeto S19 que haya sido generado por el software de ensamble o compilación respectivo.
- 4) Capacidad para configurar diversos mapas de memoria al operar en modo expandido; por ejemplo 8k de RAM y 8k de EPROM, o bien, 32k de RAM y 32k de EPROM.

- 5) Puede configurarse para ser energizada tanto con una fuente de laboratorio de cinco volts, como por un eliminador de batería.
- 6) Contiene postes para conexión de unidades desplegadas alfanuméricas comunes en la industria, así como también postes para conexión de teclados de 4x4.
- 7) Dos puertos de entrada y dos puertos de salida visibles para operar en modo expandido, además de líneas de paginación de puertos adicionales, que permiten al usuario experimentar con la conexión de dispositivos externos tales como: puertos serie o paralelos adicionales, chips de reloj, o hardware específico diseñado para una determinada aplicación.
- 8) Programador integrado de memorias EPROM, manejado por opciones especiales del software PUMMA_11. Las memorias que pueden ser programadas son: 27C64, 27C128, 27C256 y 27C512. Esta facilidad permite efectuar el desarrollo de una aplicación desde la prueba y depuración del software asociado con la misma, hasta la programación final de la memoria EPROM requerida para almacenar el código correspondiente para ejecución autónoma, todo en un solo equipo.

2.2 CIRCUITERIA.

En esta sección se hace mención detallada de los circuitos y conceptos de la FACIL_11 relacionados con el funcionamiento de la misma como CC del PLM. En la figura 2.1 se muestra un diagrama a bloques de la CMT FACIL_11.

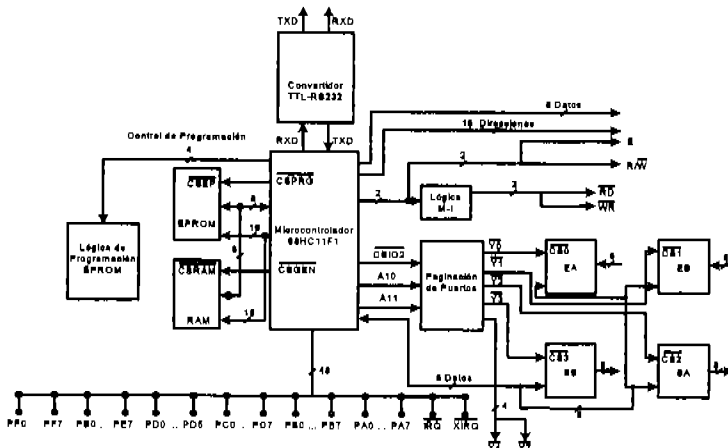


Figura 2.1.- Diagrama a bloques de la CMT FACIL_11.

2.2.1 Microcontrolador 68HC11F1.

Este componente es una computadora digital completa contenida en un solo circuito integrado, el chip contiene además periféricos de mucha utilidad en instrumentación y control.

Los componentes funcionales del 68HC11F1 son a grandes rasgos los siguientes:

- 1.- Cuatro puertos bidireccionales con capacidad de definir por software, el comportamiento, como entrada o salida, asociado con cada uno de sus bits; estos puertos se denotan con las letras A, C, D y G. Al operar en modo expandido se pierden el puerto C que pasa a ser el bus de datos y las líneas del nibble alto del puerto G que pudiera ser empleadas para paginación de memoria y puertos.
- 2.- Dos puertos de salida denotados con las letras B y F, que se pierden al operar en modo expandido, ya que pasan a ser respectivamente los bytes alto y bajo del bus de direcciones.
- 3.- Un puerto de entrada, denotado por la letra E cuyas líneas pasan a ser los ocho canales de entrada analógica asociados con el convertidor analógico digital contenido en el chip, cuando el mismo es habilitado.
- 4.- Un convertidor analógico digital de ocho bits y ocho canales, que puede ser habilitado y leído por software.
- 5.- Un puerto serie asíncrono, que puede operarse en los baudajes más comúnmente empleados, al usarse este medio de comunicación entre computadoras.
- 6.- Un puerto serie síncrono, que puede ser usado para comunicarse con periféricos que intercambian información empleando un formato serie.
- 7.- Un temporizador, que puede entre otras funciones, generar interrupciones periódicas a intervalos de repetición dependientes de la frecuencia de reloj asociada.

De los bloques y características descritas anteriormente, las más importantes para el funcionamiento de la CMT FACIL_11 como CC del PLM son: el puerto serie asíncrono, el temporizador y la capacidad de paginación de memoria y puertos; para más información acerca del 68HC11F1 se pueden consultar las hojas de datos técnicos asociados con el mismo que son proporcionados por la compañía Motorola.

2.2.2 Subpaginador de Puertos de la CMT FACIL_11.

La lógica empleada para la paginación de puertos de la CMT FACIL_11 se muestra en la figura 2.2, la misma está conformada por un decodificador de tres a ocho (74HC138) cuyas salidas Y0 a Y7, verificadas en bajo, validan sendas líneas de paginación de puerto con un intervalo asociado de 128 direcciones, así la línea Y0 decodifica al intervalo de direcciones 1800 – 187F, la línea Y1 decodifica al intervalo 1880 – 18FF, la línea Y2 decodifica al intervalo 1900 – 197F, y así sucesivamente hasta la línea Y7 que decodifica al intervalo 1B80 – 1BFF.

Las líneas de entrada al decodificador son:

1.- CSIO₂, que es una línea de control del microcontrolador que decodifica un intervalo de 2k direcciones comprendidas en el intervalo 1800 – 1FFF, esto se logra gracias a firmware que es parte de las rutinas de inicialización de la CMT FACIL_11 al operar en modo expandido.

2.- Líneas de dirección del microcontrolador comprendidas de la A7 a la A11, actuando sobre entradas de control del decodificador.

Del esquema mostrado en la figura 2.2, es fácil comprobar los intervalos de direcciones asociados con cada línea de salida del sistema de subpaginación de puertos.

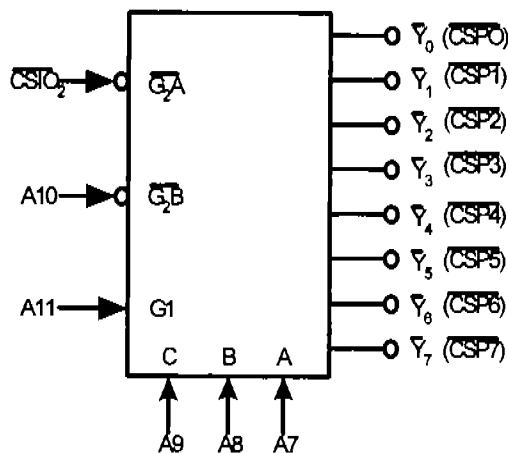


Figura 2.2.- Paginador de Puertos de la CMT FACIL_11

2.2.3 Convertidor TTL-RS 232.

Este bloque funcional está realizado con el chip MAX-232 muy popular en la industria para estos fines; la función del mismo consiste en cambiar los niveles lógicos de voltaje TTL a los niveles RS-232 (+12 volts y -12 volts) y viceversa.

2.2.4 Lógica M-I (Motorola-Intel).

Este bloque genera señalización de control de periféricos no fabricados por Motorola, para que los mismos puedan ser empleados por la CMT FACIL_11, es importante aclarar aquí que el periférico que se descara conectar no operará correctamente si la velocidad de operación del mismo no es compatible con la correspondiente a la tarjeta FACIL_11. Las señales de control generadas son WR y RD.

En la figura 2.3 se muestra el sistema lógico generador de señalización de lectura escritura de tipo Intel.

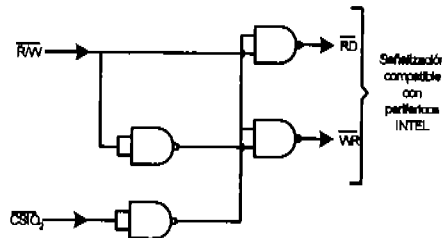


Figura 2.3.- Sistema lógico contenido en la FACIL_11 para generar la señalización de lectura escritura compatible con periféricos INTEL

2.2.5 Puertos paralelos adicionales de la FACIL_11.

La tarjeta FACIL_11 cuenta con cuatro puertos paralelos adicionales, dos de entrada y dos de salida, los puertos de entrada se denominan EA y EB, se encuentran realizados por el chip 74HC573 y los puertos de salida se llaman SA y SB que estan realizados por el circuito integrado 74HC574.

De ésta manera potencialmente se cuenta dentro de la tarjeta hasta con cuatro puertos paralelos adicionales, cuyas terminales se encuentran presentes en postes para conexionado de cable plano presentes en la tarjeta. Para fines de la CC del PLM se logran 16 entradas de las 32 entradas necesarias y 16 salidas.

A continuación se indica que puertos corresponden con cada uno de los grupos de entradas y salidas del PLM.

Entradas 00 a 07	-----	Puerto EA dirección	1800
Entradas 10 a 17	-----	Puerto EB dirección	1880
Salidas 00 a 07	-----	Puerto SA dirección	1900
Salidas 10 a 17	-----	Puerto SB dirección	1980

2.2.6 Memorias EPROM.

La tarjeta FACIL_11 incorpora en su arquitectura a un programador de memorias EPROM que requiere para su operación que la CMT FACIL_11 esté ligada vía serie con una computadora anfitriona que esté ejecutando el manejador hexadecimal PUMMA_11; la memoria que se desee programar debe ser colocada en la base que para tal fin existe en la tarjeta FACIL_11, que deberá estar configurada para operación en modo boot-strap, las EPROM que pueden ser programadas son las siguientes 27C64 o 2764, 27C128 o 27128, 27C256 o 27256 y 27C512 o 27512.

El programador puede aceptar como origen de la información a colocar en la EPROM ya sea a una lista de bytes que el usuario introduzca desde el teclado de la computadora anfitriona o a la información contenida en un archivo de tipo BLM o LEM que son los que maneja de modo natural el manejador hexadecimal PUMMA_11, si el usuario deseara programar un archivo S19 deberá antes transformarlo a un archivo BLM empleando para ello la opción dos del menú de manejo de disco del programa PUMMA_11, en caso de que el archivo a programar tenga el formato INTEL-HEX el usuario podrá pasarlo al formato S19, empleando para ello al programa HEXS19.EXE, para su posterior transformación a BLM como se ha descrito anteriormente y proceder a programarlo en la EPROM.

Para las memorias 2764, 27C64, 27128, 27C128, 27256, 27C256 cada byte es programado y verificado sucesivamente, en lo que toca a las memorias 27512 y 27C512 se debe hacer la programación completa para después proceder a hacer la verificación empleando opciones del manejador PUMA_11, esto se debe a características propias de las memorias mencionadas.

En la actualidad el voltaje más usual para programar memorias EPROM es 12.5 volts aunque para memorias fabricadas hace varios años tal voltaje podría ser mayor (de 21 a 24 volts), esto está

contemplado en el diseño del programador de la tarjeta FACIL_11, pudiendo ser configurado colocando o quitando puentes e indicándolo al manejador PUMMA_11 cuando este lo requiera.

Antes de proceder a la programación de una EPROM el usuario deberá tener dispuesto lo siguiente:

- 1) Fuente que proporcionará el voltaje de programación de la EPROM, que deberá estar calibrada a un voltaje que exceda en aproximadamente 1.5 volts al voltaje nominal requerido.
- 2) Fuente de cinco volts para polarizar la tarjeta FACIL_11.
- 3) Si la programación ha de hacerse desde disco el archivo BLM correspondiente deberá estar ya generado.

A continuación se describen los pasos a seguir para llevar a cabo la programación de una EPROM refiriéndose tanto al arreglo mostrado en la figura 2.4 como a cosas propias del manejador PUMMA_11, los pasos a seguir son los siguientes:

- a) Configurar la CMT FACIL_11 para operación en modo boot-strap (puentes J24 y J25 colocados).
- b) Con la tarjeta energizada colocar en su base la memoria a programar.
- c) Energizar con la fuente de cinco volts.
- d) Oprimir el botón de RESET en la FACIL_11.
- e) Ejecutar en la computadora anfitriona el manejador PUMMA_11.

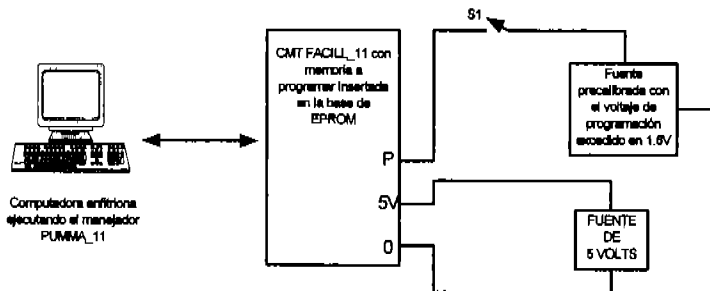


Figura 2.4.- Arreglo para programar memorias EPROM empleando a la tarjeta FACIL_11

-
- f) Una vez que PUMMA_11 esté en su menú principal y el ambiente PUMMA_11 esté ejecutándose en la tarjeta FACIL_11 pasar al menú de manejo de memoria.
- g) Una vez en el menú de manejo de memoria pasar al menú de programación de la EPROM externa. Al entrar a esta opción PUMMA_11 presentará un menú de cuatro opciones siendo estas las siguientes:
- 1) Verificar que la EPROM esté completamente borrada.
 - 2) Pasar a programar la EPROM.
 - 3) Pasar a verificar lo programado en la EPROM.
 - 4) Pasar a leer la EPROM.
- h) Verificar, antes de proceder a la programación, que la memoria esté completamente borrada de no ser así proceder a borrarla y repetir los pasos anteriores.
- i) Invocar la opción (2) del menú de programación.
- j) Indicar a PUMMA_11 el origen de la información a programar (bytes introducidos desde el teclado de la PC o un archivo de disco), en caso de que la información a programar sean bytes escritos desde el teclado PUMMA_11 procederá a pedirlos sucesivamente, en otro caso PUMMA_11 pedirá el nombre del archivo BLM o LEM a programar.
- k) Indicar a PUMMA_11 la dirección inicial de programación de los datos desde el punto de vista de la EPROM y no de su posible localización en un mapa de memoria de un sistema.
- l) Indicar a PUMMA_11 el tipo de memoria a programar tecleando los dígitos finales del número de la parte de la EPROM, por ejemplo, si se va a programar una memoria 27128 el usuario deberá teclear 128 seguido de la opresión de la tecla return.
- m) Indicar a PUMMA_11 el tipo de voltaje de programación y verificar la correcta colocación de los puentes J24 y J25 En caso de que el voltaje de programación sea 12.5 volts J24 y J25 deberán estar colocados, en otro caso deberá estar puesto solamente el puente J25.
- n) Cerrar el interruptor S1 de la figura 2.5 y después oprimir cualquier tecla, enseguida a lo anterior PUMMA_11 desplegará un letrero que dice: UN MOMENTO POR FAVOR ESTOY PROGRAMANDO. Después de terminar la programación PUMMA_11 indicará el número de errores al programar debiendo éste ser cero en caso de que la programación haya sido totalmente exitosa, si la memoria que se programó es la 27512 o 27C512 la verificación de la programación debe hacerse a posteriori, empleando para ello la opción tres del menú de programación de la EPROM.
-

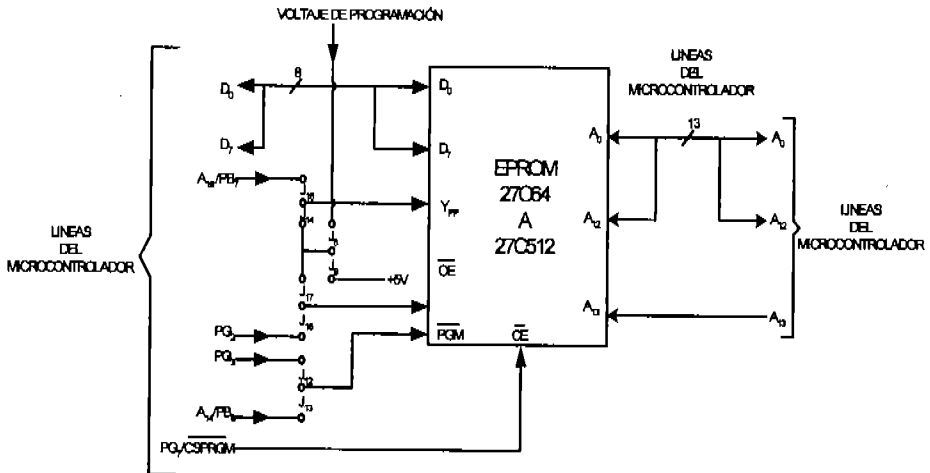


Figura 2.6.- Conexión de la memoria EPROM al bus de la tarjeta FACIL_11.

Los puentes que se aprecian en la figura 2.6 son requeridos por el hardware y software de programación de las EPROM's, además es conveniente mencionar aquí el hecho de que el firmware de restablecimiento de la tarjeta FACIL_11 reside en la memoria EEPROM interna del microcontrolador, encargándose el mismo de inicializar las líneas PG2 y PG3 de la figura 2.6, para que se pueda ejecutar código desde la memoria EPROM colocada.

2.2.8 Memoria RAM.

La tarjeta FACIL_11 puede recibir memorias RAM estáticas de 8 kb y 32 kb, que tamaño es requerido en un momento dado, depende del tipo de mapa de memoria con que se desee trabajar, en la figura 2.7 se muestra el conexionado de la RAM al bus de la tarjeta.

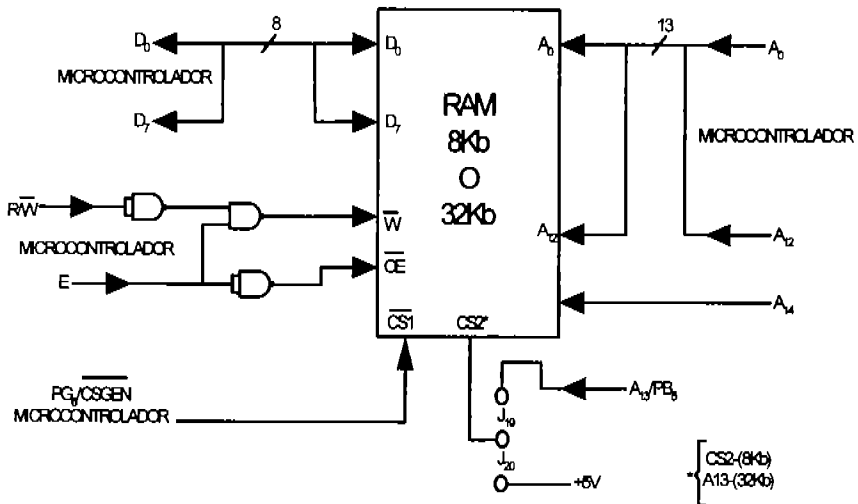


Figura 2.7.- Conexión genérica de la memoria RAM al bus de la tarjeta FACIL_11

Los diversos mapas de memoria en que puede operar la CMT FACIL_11 se configuran mediante colocación de puentes, al operar en modo expandido pueden configurarse seis mapas de memoria diferentes.

Para fines de la CC del PLM y las configuraciones de funcionamiento del mismo los mapas de memoria requeridos se muestran en la tabla 2.1

Tabla 2.1.- Configuración de los mapas EA y EB.

Configuración de Funcionamiento. PLM	Mapa Empleado	Tamaño de Memorias	Colocación de Puentes
1 y 2	EA	8 Kb RAM 8 Kb EPROM	J7, J12, J14, J16, J20
3	EB	32 Kb RAM 32 Kb EPROM	J7, J13, J14, J16, J19

2.2.9 Localización de componentes en la CMT FACIL_11.

En la figura 2.8 se muestra una vista de la plantilla de la tarjeta FACIL_11, en dicha figura se aprecia la localización de los diversos componentes que integran la arquitectura de la tarjeta.

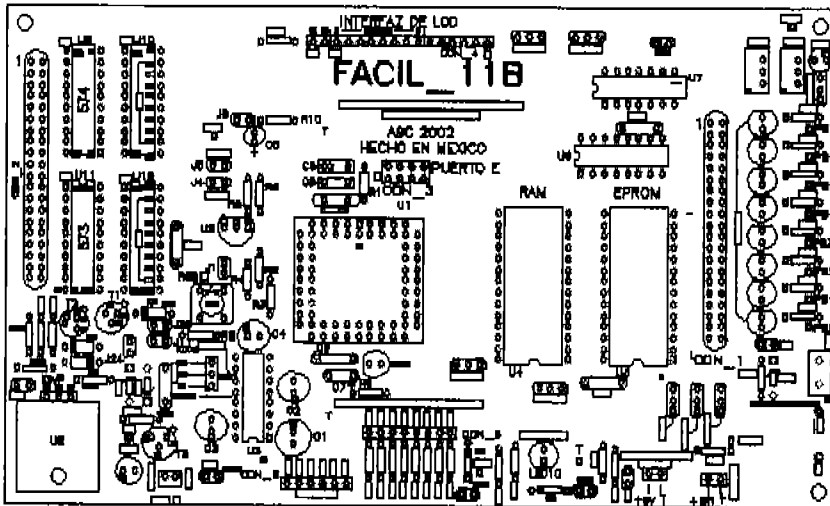


Figura 2.8.- Tarjeta FACIL_11 Computadora Central del PLM.

En la parte central se aprecia la localización del microcontrolador 68HC11F1 que es el corazón de la FACIL_11.

A la izquierda de la tarjeta se observan cuatro circuitos que constituyen cuatro puertos paralelos adicionales EA, EB, SA y SB (EA y EB son los chips U11 y U12, SA y SB son los chips U9 y U10), a la izquierda de los mismos se encuentra el conector 2 mediante el cual se tiene acceso a los cuatro puertos ya mencionados, en la figura 2.9 se muestra la descripción del conector 2. En la parte inferior izquierda se localiza un regulador de voltaje 7805 y el conector 7 utilizados para polarizar la tarjeta, además se encuentra también la parte del hardware relacionado con el programador de memorias EPROM.

A la derecha de la tarjeta aparece el conector 1 mediante éste conector el usuario tiene acceso a líneas de control de selección de puerto, líneas de polarización y líneas de puerto del propio microcontrolador; en la figura 2.9 se muestra en detalle el conector 1. A la derecha del conector 1 se encuentran los ocho leds testigos del puerto B del microcontrolador y a la izquierda del ya mencionado conector 1 se encuentra la zona donde se localiza las memorias RAM y EPROM externas, así como el subpaginador de puertos (chip U6).

A la izquierda del microcontrolador se puede observar el botón de restablecimiento manual. Bajo el botón de restablecimiento manual a la derecha se observa el chip MAX-232, bajo el microcontrolador 68HC11F1 se ve el puerto A; conector 6 (interfaz de teclado) y el conector 5 (puerto D). En la parte central superior se localiza el conector 4 utilizado para la interfaz de LCD (display) y el conector 3; puerto E. En la figura 2.10 se encuentra la descripción de los postes de los conectores 3, 4, 5 y 6.

A la derecha del chip MAX-232, se encuentra un conector de tres postes por donde ha de conectarse cable de enlace serie entre la FACIL_11 y una computadora anfitriona, en la figura 2.11 se muestra el conexionado de cable de enlace serie.

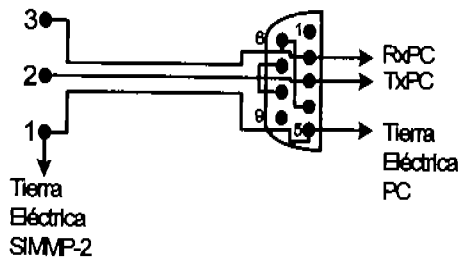


Figura 2.11 .- Conexionado del cable de enlace serie

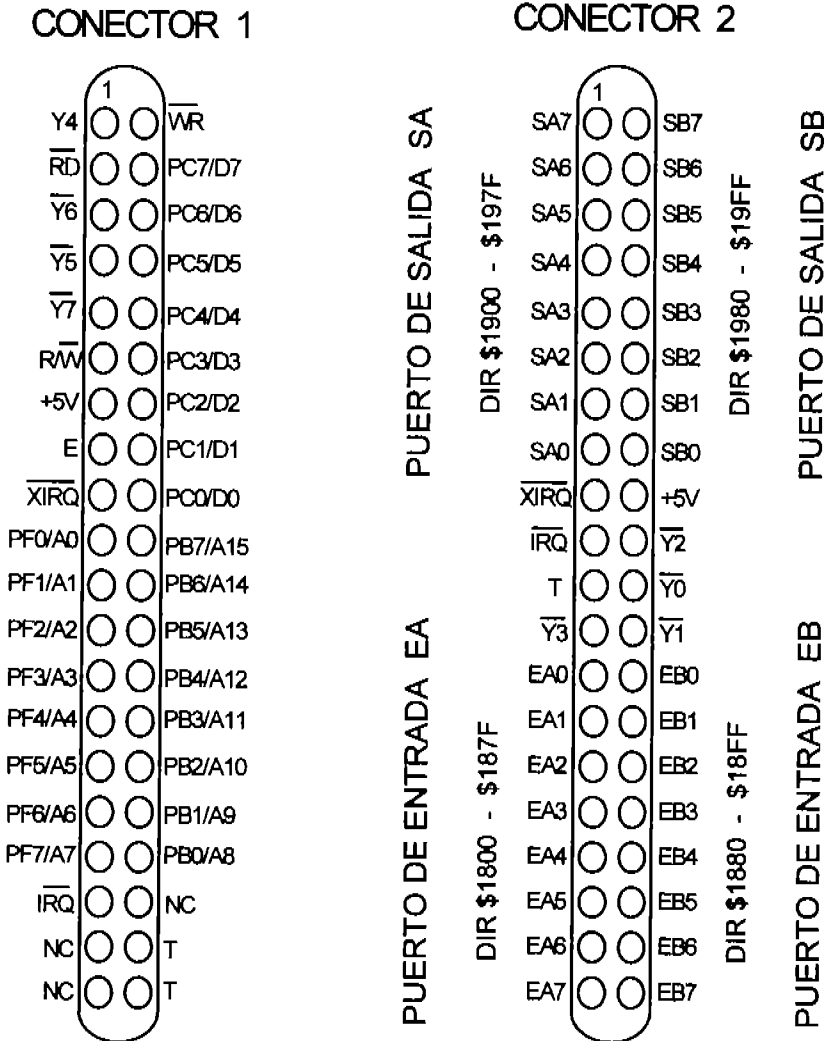


Figura 2.9.- Conectores 1 y 2 de la Computadora Central FACIL_11

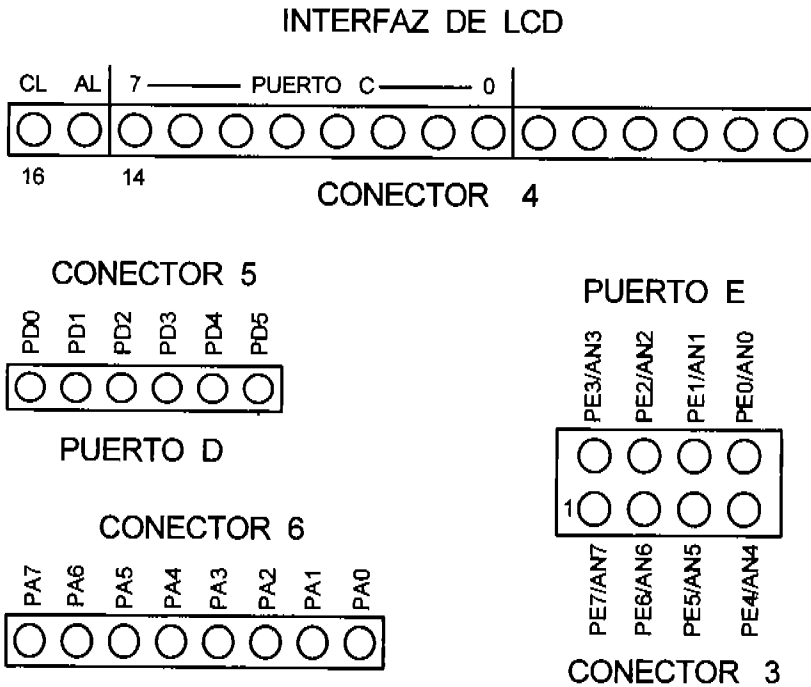
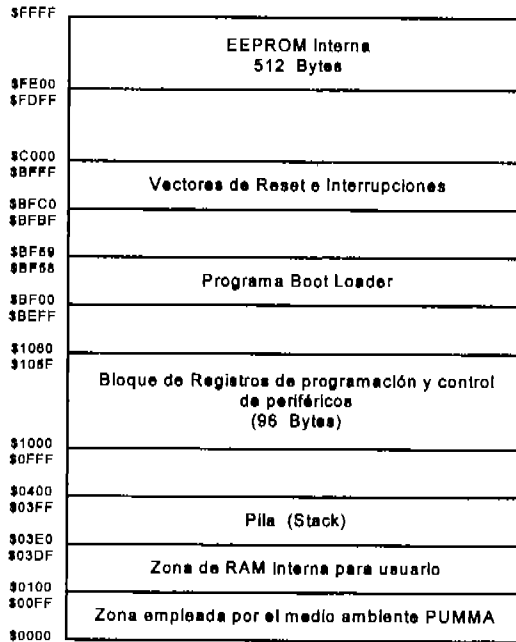


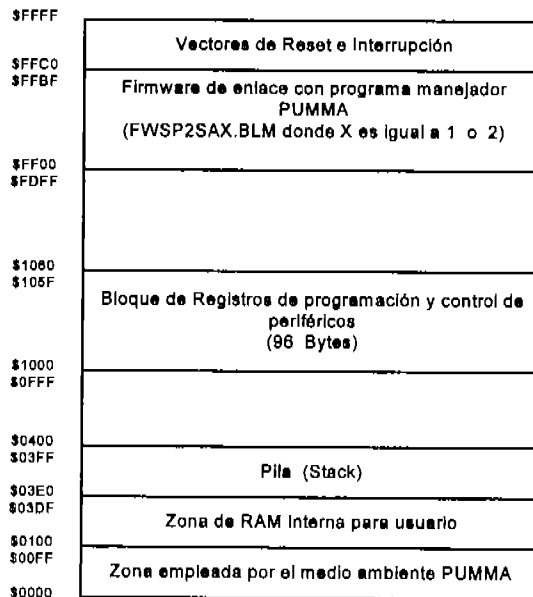
Figura 2.10 .- Conectores 3, 4, 5 y 6 de la tarjeta FACIL_11

2.3 ESPECIFICACIONES DE CONFIGURACIÓN DE MAPAS DE MEMORIA Y PUERTOS DE LA TARJETA FACIL_11.

La CMT FACIL_11 puede operar con diferentes mapas de memoria; cuando la misma opera en modo single-chip o boot-strap, los mapas correspondientes son los naturales del microcontrolador 68HC11F1, véanse las figuras 2.12 y 2.13.



Figuras 2.12 .- Mapa de memoria de la CMT FACIL_11 operando en modo Boot - Strap



2.13 .- Mapa de memoria de la CMT FACIL_11 operando en modo Single – Chip

2.3.1 Mapas de memoria de la CMT FACIL_11 operando en modo expandido.

Para la operación en modo expandido pueden configurarse dos mapas de memoria denominados como mapa EA y mapa EB; para que la tarjeta opere en expandido en forma transparente para el usuario, cuando el mismo emplea como herramienta al software PUMMA_11, el registro CONFIG (dir 103F) del microcontrolador deberá contener “unos” en los cuatro bits más significativos, esto para que el origen de la EPROM sea la dirección 103F al operar en modo expandido, un byte típico a programar en CONFIG es FF, si el usuario emplea facilidades del microcontrolador tales como el *watch dog* algunos(s) de los bits menos significativos de CONFIG deberán ser cero, para más detalles acerca del registro aquí mencionado puede consultarse el manual de referencia del HC11 editado por MOTOROLA. Dado que el registro aquí mencionado es una localidad de tipo EEPROM, para cargar un byte en el mismo se puede emplear un comando de PUMMA_11 que programe bytes individuales en la EEPROM considerando que la dirección asociada es la 103F.

Para cada uno de los mapas de memoria, EA y EB, del modo expandido deberá haberse cargado firmware asociado con los mismos a partir de la dirección FE00, que será el origen de la EEPROM, en el modo expandido; esto permite operar el modo expandido sin memoria EPROM, desarrollando las aplicaciones en forma ágil en memoria RAM.

Los pasos a seguir para configurar la tarjeta FACIL_11, de modo que la misma ejecute una aplicación en forma autónoma desde la memoria EPROM, puede verse más adelante en el tema “EJECUCIÓN AUTÓNOMA DESDE EPROM”.

A continuación se describen las características generales, así como la forma en que se configuran los mapas EA y EB.

Mapa EA.

Para este mapa se tiene el siguiente perfil: Operación en modo expandido con 8 kb de RAM externa, 1 kb de RAM interna, 7.5 kb de EPROM y 0.5 kb de EEPROM interna con el firmware FWFCLEA residente. Para configurar este mapa se deben colocar los siguientes puentes: J7, J12, J14, J16 y J20; además de que el archivo FWFCLEA2.BLM deberá haber sido programado en la

EEPROM interna a partir de su origen (FE00); el contenido del intervalo de vectores de RESET e interrupciones es el siguiente:

El vector de RESET principal apunta al origen de la EEPROM interna y todos los demás vectores de RESET e interrupción apuntan a direcciones de la página cero, siendo estas las mismas que las que corresponden a los vectores asociados con el modo boot-strap, lo anterior permite al usuario trabajar con el concepto de seudovector de interrupción al programar sus aplicaciones en modo expandido; en la figura 2.14 se aprecia el mapa de memoria EA; los requerimientos en cuanto a chips de memoria para este mapa de memoria son RAM estática de 8 kb (6264 ó equivalente) en la base correspondiente, EPROM de 8 kb (27C64) en su respectiva base.

\$FFFF	Vectores de Reset e interrupción
\$FFC0 \$FFBF	
\$FEA0 \$FE9F	Zona de EEPROM interna para usuario
\$FE00 \$DFFF	Firmware de enlace con manejador PUMMA EEPROM interna (FWFCLEA2.BLM)
\$E000 \$DFFF	EPROM externa para usuario
\$C000 \$BFFF	RAM externa para usuario
\$2000 \$1FFF	Submapa alterno de Puertos
\$1C00 \$1BFF	Submapa de Puertos
\$1800 \$17FF	
\$1080 \$105F	Registros de control y programación de periféricos
\$1000 \$0FFF	
\$0400 \$03FF	Pila (Stack)
\$03E0 \$03DF	Zona de RAM interna para usuario
\$0100 \$00FF	Zona empleada por el medio ambiente PUMMA
\$0000	

Figura 2.14.- Mapa de memoria EA (operación en modo expandido)

Mapa EB.

Para este mapa se tiene el siguiente perfil: Operación en modo expandido con 28.90625 kb de RAM externa (29600 bytes), 1 kb de RAM interna, 31.5 de EPROM y .5 kb de EEPROM interna con el firmware FWFCLEB residente. Para configurar este mapa se deben colocar los siguientes puentes: J7, J13, J14, J16 y J19; además de que el archivo FWFCLEB2.BLM deberá haber sido programado en la EEPROM interna a partir de su origen (FE00); el contenido del intervalo de vectores de RESET e interrupciones es el siguiente:

El vector de RESET principal apunta al origen de la EEPROM interna y todos los demás vectores de RESET e interrupción apuntan a direcciones de la página cero, siendo estas las mismas que las que corresponden a los vectores asociados con el modo boot-strap, lo anterior permite al usuario trabajar con el concepto de seudovector de interrupción al programar sus aplicaciones en modo expandido; en la Figura 2.15 se aprecia el mapa de memoria EB; los requerimientos en cuanto a chips de memoria para este mapa son: RAM estática de 32 kb (62256 ó equivalente) en la base correspondiente, EPROM de 32 kb (27C256) en su respectiva base. La tarjeta FACIL_11 como computadora central del PLM emplea el mapa de memoria EB. La tabla 2.2 resume la configuración de puentes asociada con los mapas EA y EB ya descritos.

8FFF	Vectores de Reset e Interrupción
8FFC0 8FFBF	Zona de EEPROM interna para usuario
8FEA0 8F89F	Firmware de enlace con manejador PUMMA EEPROM interna (FWFCLEB2.BLM)
8F800 8FDFF	EPROM externa para usuario
88000 87FFF	RAM externa para usuario
82000 81FFF	Submapa alternativo de Puertos
81C00 81BFF	Submapa de Puertos
81800 817FF	RAM externa para usuario
81080 8105F	Registros de control y programación de periféricos
81000 80FFF	RAM externa para usuario
80400 803FF	Pila (Stack)
803E0 803DF	Zona de RAM interna para usuario
80100 800FF	Zona empleada por el medio ambiente PUMMA
80000	

Figura 2.15 .- Mapa de memoria EB (operación en modo expandido)

Tabla 2.2.- Configuración de puentes y firmware residente para los Mapas de memoria EA y EB de la CMT FACIL_11.

Mapa	Puentes en CMT FACIL_11	Firmwarw Residente
EA	J7, J12, J14, J16 Y J20	FWFCLEA
EB	J7, J13, J14, J16 Y J19	FWFCLEB

2.3.2 Ejecución autónoma desde EPROM.

Una vez que el usuario ha desarrollado el código asociado con una aplicación, probando y depurando el mismo en RAM, empleando para ello al software PUMMA_11 y la tarjeta FACIL_11, si lo desea puede hacer que el software desarrollado se ejecute de manera autónoma desde la memoria EPROM, para lograr esto se puede seguir los siguientes pasos:

- 1.- Cambiar la dirección de origen del código a la que corresponda con el origen de la EPROM (E000 para el mapa EA y 8000 para el mapa EB).
- 2.- Generar, empleando algún ensamblador, el archivo S19 correspondiente.
- 3.- Programar la memoria EPROM a usar con el archivo S19 generado en el paso anterior.
- 4.- Con la EPROM en su base colocar el puente J11.

De esta forma, siempre que esté el puente J11 colocado al dar RESET en modo expandido el firmware residente hará que el código presente a partir del origen de la EPROM sea ejecutado. En la tabla 2.3 se resumen los accionamientos llevados a cabo por la tarjeta FACIL_11 después de un RESET, esto para los modos boot-strap, singlechip y expandido.

Tabla 2.3.- Acciones al restablecer (RESET) en la tarjeta FACIL_11

Modo de Operación	Puente colocado entre Las terminales Tx y Rx de FACIL_11	Puente no colocado entre Las terminales Tx y Rx de FACIL_11	Puente J11 colocado	Puente J11 no colocado
Boot-strap	Se genera un salto al origen de la EPROM interna.	Se pasa a firmware de recepción de un programa a colocarse en la página cero.	-----	-----
Single Chip	Posibilidad no permitida	-----	Se genera salto a la dirección \$FEA0	Se pasa a ejecutar firmware con el manejador PUMMA_11
Expandido	Posibilidad no permitida	-----	Se genera salto al origen de la EPROM externa	Se pasa a ejecutar firmware con el manejador PUMMA_11

2.3.3 Submapa de puertos.

El submapa de puertos está definido en un intervalo de 1 kb (de la dirección 1800 a la 1BFF) dividido en ocho subintervalos de 128 direcciones cada uno, al invocar una dirección de puerto en un subintervalo determinado se verifica en nivel bajo una de ocho líneas de habilitación de puerto, en la tabla 2.4 se detalla el submapa de puertos.

Tabla 2.4 .- Detalle del submapa de puertos de la tarjeta FACIL_11

INTERVALO DE DIRECCIONES	USO EN LA TARJETA FACIL_11	LÍNEA DE HABILITACIÓN ASOCIADA
1800 - 187F	Habilitación de puerto de entrada EA	Y0 (negada)
1880 - 18FF	Habilitación de puerto de entrada EB	Y1 (negada)
1900 - 197F	Habilitación de puerto de salida SA	Y2 (negada)
1980 - 19FF	Habilitación del puerto de salida SB	Y3 (negada)
1A00 - 1A7F	Habilitación de LCD (modo expandido)	Y4
1A80 - 1AFF	Disponible para el usuario	Y5 (negada)
1B00 - 1B7F	Disponible para el usuario	Y6 (negada)
1B80 - 1BFF	Disponible para el usuario	Y7 (negada)

Cabe señalar aquí, que aún cuando las ocho líneas de habilitación de puerto descritas en la tabla 2.4, están presentes en postes de los conectores uno y dos de la tarjeta FACIL_11, el usuario deberá tener cuidado de emplear únicamente las disponibles para él indicadas en la tabla anterior, de lo contrario se podría originar colisiones lógicas entre los puertos presentes en la tarjeta y los que el usuario colocara empleando alguna o algunas de las líneas de habilitación; sin embargo, el que alguno de los puertos no estuviera presente libera la línea de habilitación correspondiente; por ejemplo, en una tarjeta FACIL_11 que no tuviera conectado un desplegador en el conector cuatro, además de no tener colocado el chip doce (puerto de entrada EA), estarían disponibles para el usuario las líneas de habilitación Y0 (negada) y Y4.

2.3.4 Submapa de puertos alterno.

En todos los mapas de memoria aquí descritos se aprecia la existencia de un submapa de puertos alterno, definido en un intervalo de 1 kb (direcciones de la 1C00 a la 1FFF), si el usuario lo requiriera podría conectar un 74HC138 externo que generaría en sus ocho líneas de salida, habilitaciones de puerto asociadas cada una de ellas con sendos subintervalos de 128 direcciones cada uno, en la figura 2.16 se muestra como hacer esto, así como también los subintervalos que verificarían cada línea de habilitación; en la práctica se ha visto que con las líneas de habilitación propias del submapa de puertos normal (direcciones de la 1800 a la 1BFF) pueden conectarse, mediante lógica de enlace, diversos puertos externos a la FACIL_11.

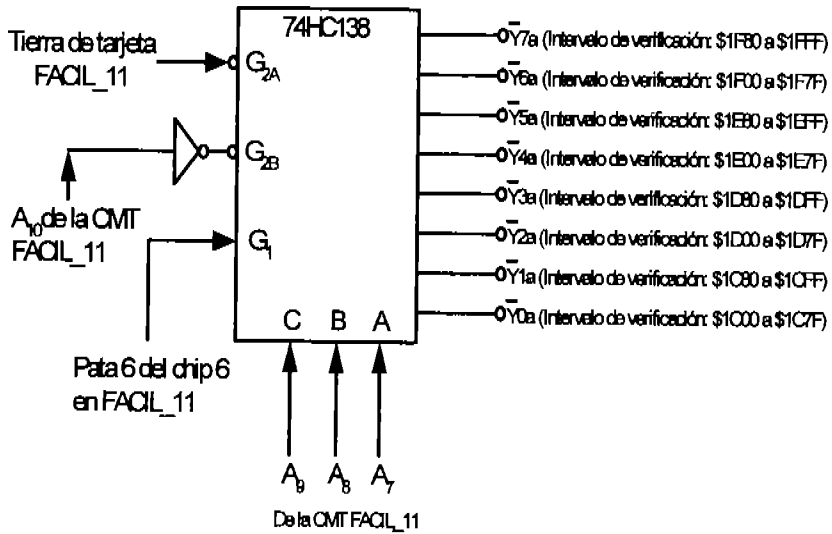


Figura 2.16.- Conexión de un decodificador 74HC138 externo para validar el submapa de puertos alterno con ocho subintervalos de verificación.

Es justo señalar que el texto que constituye el presente capítulo ha sido tomado del manual de la tarjeta FACIL_11 cuyo autor es el M.I Antonio Salvá Calleja; director de éste trabajo de tesis.

CAPÍTULO 3

CIRCUITOS DE ENTRADA, DE SALIDA Y CIRCUITOS AUXILIARES

DISEÑO DE CIRCUITOS IMPRESOS ASOCIADOS

En este capítulo se describe detalladamente la circuitería concerniente al bloque de entradas, al bloque de salidas, a los circuitos auxiliares y el diseño de los circuitos impresos correspondientes a las tarjetas periférica 1 y periférica 2.

En el capítulo inicialmente se describe el bloque de entradas integrado por 32 circuitos óptó acoplados, más tarde se detallan los circuitos auxiliares que están constituidos por el reloj, el display y los puertos adicionales, continúa el capítulo con la explicación de la función de los circuitos de salida en el PLM, pasando más tarde con los conceptos generales sobre relevadores, continuando con la organización y conexión de los circuitos de interfazado a relevadores de salida y finalizando con el diseño de los circuitos impresos asociados.

3.1 DESCRIPCIÓN DE LOS CIRCUITOS DE ENTRADA.

El bloque de entrada está constituido por circuitos electrónicos que se utilizan para adaptar o transformar las señales procedentes de emisores de señal en señales equivalentes, con un formato apropiado para ser admitidas por la unidad de entrada/salida de la computadora central.

3.1.1 Descripción general.

Los circuitos de entrada están diseñados de modo que la señal que se utiliza como entrada de datos procedente de un emisor de señales, esta compuesta por una tensión de alrededor de 24 volts de corriente continua para significar el uno lógico, en tanto que el cero lógico está asociado con un voltaje nulo.

Las señales en las entradas están aisladas eléctricamente de los puertos de entrada físicos del microcontrolador que realiza la computadora central. Esto se hace por varias razones, las dos principales son:

- 1 Evitar que cualquier sobre tensión en la entrada pueda afectar gravemente a todo el circuito electrónico del PLM.

- 2 Sustituir la señal de 24 volts por otra equivalente pero ahora de 5 volts, que es la tensión de trabajo de todo el circuito electrónico del sistema.

El procedimiento general para efectuar las entradas con arreglo a estos requisitos se basa en utilizar unos elementos electrónicos denominados *acopladores opto electrónicos*, que en este caso particular se llevara a cabo con el optoacoplador 4N26. En la figura 3.1 se muestra el circuito asociado con un bit de entrada del PLM desarrollado.

3.1.2 Descripción del circuito particular para una única entrada.

Para cada una de las entradas del PLM, se tiene un circuito de optoacoplamiento cuyas funciones son por una parte cambiar los niveles lógicos de entrada (0-24 volts) a niveles TTL propios de los puertos de entrada de la computadora central y por otra parte nos proporcionara un aislamiento eléctrico entre la circuitería asociada con los sensores y la computadora central, que opera a cinco volts. En la figura 3.1 se muestra el circuito genérico de optoacoplamiento empleado.

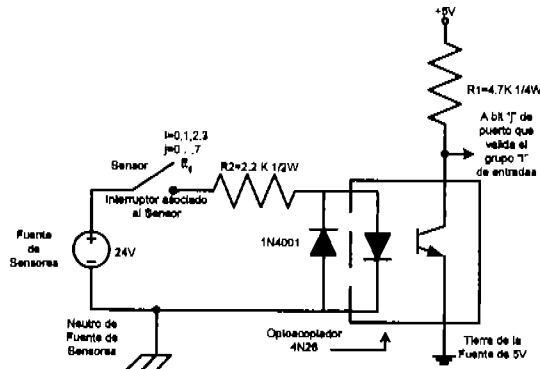


Figura 3.1.- Circuito de optoacoplamiento genérico empleado para cada una de las 32 entradas del PLM.

El componente central de nuestros circuitos de entrada es el optoacoplador debido a esta situación se desprende la necesidad de explicar el funcionamiento de este elemento que es el encargado de efectuar el aislamiento eléctrico y la obtención del nivel de TTL necesario. En la figura 3.2 se muestra la composición interior de este componente.

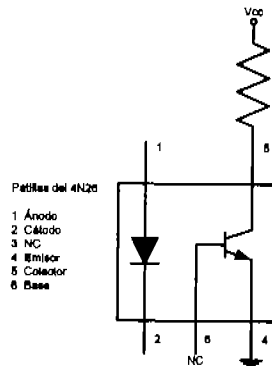


Figura 3.2 .- Composición interna del optoacoplador 4N26.

Descripción de un optoacoplador.

Un optoacoplador es un dispositivo electrónico formado por un emisor y un receptor. El emisor transformará una señal eléctrica en óptica, transmitiéndose al receptor, el cual la recogerá y la volverá a transformar a señal eléctrica.

Los optoacopladores más usuales están formados por un emisor que podrá ser un diodo led o un diodo laser, y un receptor que será un fotodiodo o un fototransistor, generalmente. Todos estos elementos se encuentran dentro de un encapsulado que por lo general es de tipo DIP.

La señal de entrada es aplicada al fotoemisor y la salida es tomada del fotorreceptor. Los optoacopladores son capaces de convertir una señal eléctrica en una señal luminosa modulada y volver a convertirla en una señal eléctrica. La gran ventaja de un optoacoplador reside en el aislamiento eléctrico que puede establecerse entre los circuitos de entrada y salida.

Funcionamiento del optoacoplador 4N26.

- a) Cuando aparece una tensión sobre los terminales del diodo (que resulta de la fuente de sensores de 24 volts), este emite un haz de rayos infrarrojo que se transmite a través de una pequeña guía de ondas de plástico o cristal hacia el fotorreceptor. La energía luminosa que incide sobre el fotorreceptor hace que este genere una tensión eléctrica a su salida.

- b) El fototransistor (parecido al que se muestra en la figura 3.3). Se encuentra conectado en configuración de emisor común donde la salida es tomada en el colector (V_c), y la entrada depende del voltaje V_d , que a su vez está en función de la operación del diodo led del optoacoplador; de manera que si el led no emite luz podemos considerar que el fototransistor se encuentra en la región de corte y el voltaje de salida del optoacoplador; tiene el valor de 5 volts, que es el valor de V_{cc} , y que se considera como un uno lógico.

$$V_{cc} = 5 \text{ volts} \quad \text{Donde } V_c = 5 \text{ volts (uno lógico)}$$

Para activar el transistor tenemos que aumentar el voltaje V_d (el diodo led emite luz) para que de esta manera el transistor entre en la región de saturación y se cumpla con:

$$V_c = 0.2 \text{ volts} \quad \text{Donde } V_c = 0.2 \text{ volts (cero lógico)}$$

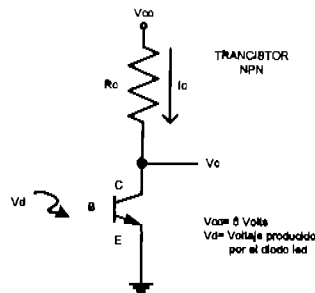


Figura 3.3.- Configuración en emisor común para el optoacoplador 4N26.

Es conveniente considerar que tanto para el uno como para el cero lógico TTL en realidad les corresponde respectivamente un rango de voltaje de V_{cc} . (Para el 1 lógico el rango de voltaje es de 2 volts a V_{cc} mínimo de 4.75 volts y V_{cc} máximo de 5.25 volts. Para el 0 lógico el rango de voltaje es 0 volts a 0.8 volts)

El optoacoplador funcionará sólo si la tensión de entrada tiene la polaridad correcta, si se conecta con la polaridad invertida, no pasara nada, solo que no funcionará, ya que el diodo no conducirá por estar alimentado en sentido inverso.

Se entiende que por el diodo circula la corriente procedente de la tensión de 24 volts, y por el transistor, la corriente procedente del circuito interior, que trabaja con 5 volts. De esta manera, la señal de 24 volts se ha convertido en una señal equivalente, pero ahora de nivel TTL. Además, la tensión de 5 volts no tiene nada que ver con la corriente de la fuente de sensores.

De esta manera se obtiene el cambio de los niveles lógicos de entrada (0 – 24 volts) a niveles TTL y el aislamiento eléctrico mencionado al principio de esta sección.

Otros elementos que componen nuestro circuito de entrada son:

- 1) Un diodo 1N4001 que proporciona protección adicional al circuito, en el caso de que la fuente de sensores se conecte erróneamente con la polaridad invertida, el 1N4001 garantizará que para el diodo led del optoacoplador solo haya 0.7 volts en polarización inversa.
- 2) Una resistencia R_2 utilizada para limitar la corriente que circula por el diodo del optoacoplador.
- 3) Una resistencia R_1 que se emplea para sujetar los valores de voltaje de salida V_c y la corriente I_c a un valor adecuado para el manejo de la lógica empleada en la computadora central del PLM.

En la figura 3.4 se muestra el circuito a partir del cual se calculó el valor de la resistencia R_2 .

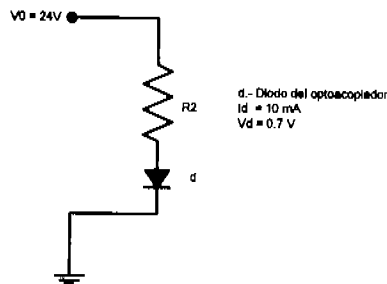


Figura 3.4.- Circuito para limitar la corriente en el diodo del optoacoplador 4N26.

En la figura 3.5 se muestra el circuito a partir del cual se calculo el valor de la resistencia R_1 . A continuación se muestran los calculos para las resistencias R_1 y R_2 .

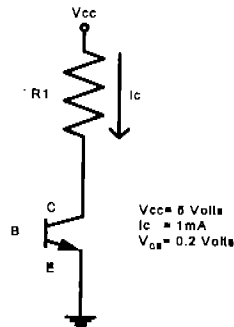


Figura 3.5 .- Circuito empleado para calcular el valor de la resistencia R_1 .

Para la resistencia R_2 :

Datos: $V_d = 0.7 \text{ V}$, $I_d = 10 \text{ mA}$, $V_o = 24 \text{ V}$

Operaciones: $V_o = R_2 I_d + V_d$, $R_2 = (V_o - V_d) / I_d$, $P = VI$

$$R_2 = 2330 \Omega \quad , \quad P = (24 \text{ volts})(10 \text{ mA}) \quad P = 240 \text{ mW}$$

Como I_d puede tener como valor máximo hasta 60 mA podemos elegir el valor comercial de resistencia más proximo al calculado, por otra parte notamos que la disipación de potencia es de 240 mW que esta en el límite del $\frac{1}{4}$ W (valor comercial), por este motivo es conveniente darle una cierta holgura en el valor para la disipación de potencia. Finalmente se selecciono el valor de la resistencia como 2.2 k Ω a $\frac{1}{2}$ W (valores comerciales)

Para calcular R_1 tenemos:

Datos: $V_{cc} = 5 \text{ volts}$, $V_{CE} = 0.2 \text{ volts}$, $I_C = 1 \text{ mA}$

Operaciones: $V_{cc} = V_{R1} + V_{CE}$, $V_{cc} = R_1 I_C + V_{CE}$, $R_1 = (V_{cc} - V_{CE}) / I_C$, $P = VI$

$$R_1 = 4.8 \text{ k}\Omega \quad , \quad P = 5 \text{ mW}$$

De acuerdo a los cálculos y tomando en cuenta que la I_C máxima es de 150 mA se eligió la resistencia

$$R_1 = 4.7 \text{ k}\Omega \text{ a } \frac{1}{4} \text{ W.}$$

Cabe señalar que se supone que la fuente de sensores debe tener una regulación adecuada, de modo que el voltaje suministrado por la misma no baje más de un 10% de su valor nominal (24 volts), en el peor caso (32 entradas conectadas con sus correspondientes interruptores de censor cerrados).

Además es necesario señalar que hay una inversión lógica en cada circuito de optoacoplamiento, esto es, en la lógica de interruptores convencionalmente se asocia un interruptor abierto a un cero lógico y a un interruptor cerrado se asocia un uno lógico, por lo tanto el circuito empleado presentara una inversión, ya que en el momento en que el interruptor asociado con la entrada se cierra tenemos un cero lógico en la salida del circuito, (el transistor se satura) el nivel de voltaje en el punto V_c (figura 3.3) será cero TTL y cuando la entrada sea cero (interruptor de sensor abierto) el nivel en el punto V_c (figura 3.3) será de uno TTL.

La inversión adicional necesaria para lograr el ajuste a la lógica de interruptores convencional, puede llevarse a cabo de dos maneras:

- a) Por hardware, colocando en el punto V_c un transistor adicional para cada entrada, esto implicaría 32 transistores más.
- b) Por software, esto es, el programa (compilador) generador de código, asociado con un programa fuente en el lenguaje SILL1, (lenguaje propio del PLM desarrollado) introduce el código que invierte los niveles lógicos asociados con los bits de entrada.

Cabe señalar que dicho compilador es objeto de otro trabajo ya que esta tesis esta enfocada a desarrollar únicamente el hardware robusto para el PLM.

En conclusión evaluando las dos posibles alternativas a la inversión de los bits de entrada, se opto por la solución vía software que representa; ahorro de componentes y en consecuencia un costo económico menor y por último simplifica la organización y distribución de elementos en el diseño del circuito impreso asociado a las 32 entradas del PLM.

3.1.3 Organización del bloque de entradas.

El bloque de entradas esta constituido por 32 entradas que el PLM reconoce con un nivel de uno lógico, cuando en determinada entrada, se presente un voltaje nominal de 24 volts medidos entre la terminal correspondiente y el punto neutro de la fuente de sensores, en otro caso el nivel tomado será cero lógico.

Las entradas están agrupadas en cuatro grupos de ocho entradas cada uno ya que la información en el microcontrolador empleado está organizada de esta manera. En la figura 3.6 se muestra la organización de tres de los cuatro grupos de entradas.

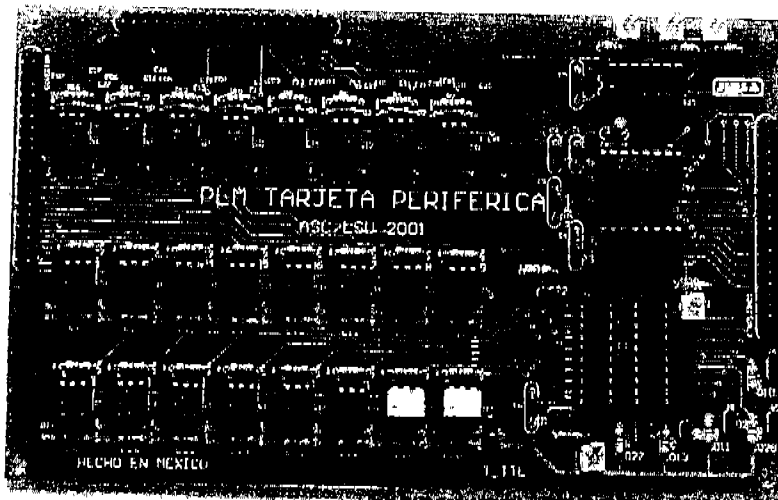


Figura 3.6 - A la izquierda de la tarjeta periférica I se muestran 24 de los circuitos de entrada, organizados en tres grupos.

Las entradas son denotadas empleando tres caracteres; el primero es una letra mayúscula que en este caso será la 'E' (por el hecho de ser entrada), el segundo y tercer simbolos son números, el primero de los cuales estará en un intervalo de valores de cero a tres para determinar el grupo al cual pertenece la entrada, y el número final se encontrara dentro de un rango de valores de cero a siete que indicara el número de 'bit' de entrada correspondiente; por ejemplo tenemos la entrada 'E16'

denotara el hecho de que tenemos una entrada que pertenece al grupo uno y que el número de bit es el seis. Para cada grupo de ocho entradas corresponde un puerto físico con una determinada dirección en el mapa de puertos.

3.2 DESCRIPCIÓN DE LOS CIRCUITOS AUXILIARES.

En el capítulo número uno se describió la estructura a bloques del PLM y uno de estos bloques es el Bloque de comando Local y Despliegue (BCLD), este bloque en conjunto con otros circuitos que funcionan como puertos de entrada adicionales constituyen lo que denominaremos en esta sección del trabajo de tesis como circuitos auxiliares, se comenzará con una descripción panorámica de los circuitos para en párrafos siguientes detallar cada una de las partes que constituyen los llamados circuitos auxiliares.

Desde el punto de vista del usuario final el BCLD está constituido por tres componentes, uno de ellos es la Unidad Desplegadora (UD) que maneja un display de dos renglones de 16 caracteres por renglón, otro es un panel que contiene un conjunto de cinco pares de postes para habilitar cinco entradas binarias auxiliares (entradas auxiliares EA1 a EA5) para este prototipo específico se tiene acceso a estas entradas solo directamente en una de las tarjetas de circuito impreso denominada como "Tarjeta Periférica 1", no hay presencia de estas entradas en el gabinete del PLM. Como una parte del BCLD tenemos un conjunto de seis botones (ver tabla 3.1) donde tres de los mismos se emplean para ajuste y puesta a tiempo del Reloj de Tiempo Real (RTR), el otro botón se emplea para probar secuencialmente los diversos mensajes que pudiera tener una aplicación, un interruptor (Ja) para controlar la reiniciación del RTR y por último otro interruptor (Jb) que controla la puesta a cero de todas las variables que se utilicen para determinada aplicación, cabe aclarar que esta serie de cuatro botones y dos interruptores, si se encuentran disponibles para el usuario en el gabinete del PLM. El tercer componente del BCLD es un Reloj de Tiempo Real (RTR) empleando para tal efecto el chip MM58274N fabricado por National.

De esta manera hacemos la presentación general del BCLD, en párrafos subsecuentes se profundiza en la descripción de cada una de las partes que constituyen el BCLD.

Adicionalmente al bloque BCLD el PLM necesita otros circuitos que cumplen la función de puertos de entrada adicionales implantados utilizando el circuito integrado 74LS573 a dichos

puertos durante el desarrollo de este trabajo los nombraremos como Entradas Auxiliares A (EAA) y Entradas Auxiliares B (EAB) en la sección siguiente se amplía la información sobre estos circuitos.

3.2.1 Puertos paralelos adicionales.

En párrafos anteriores se especifico que el PLM cuenta con treinta y dos circuitos de entrada distribuidos para su manejo en cuatro secciones con ocho entradas por sección ahora para manejar cada conjunto de entradas es necesario contar con circuitos que cumplan la función de agrupar las entradas en puertos, para satisfacer está necesidad, contamos por una parte en la tarjeta FACIL_11 (CC del PLM) con dos puertos paralelos denominados como Entradas A (EA) y Entradas B (EB) implementados con el circuito integrado 74LS573, al contar con solo dos puertos de los cuatro que se requieren es necesario adicionar los dos puertos faltantes y son estos dispositivos los que nos ocupan en está sección.

Los puertos adicionales están implementados también con el circuito con el chip 74HC573 que es un latch octal con salida de tercer estado, los puertos están denominados como EAA con la dirección en el submapa de puertos de la CC 1B80 que agrupa las entradas en el intervalo 20 a 27 y como EAB al último de los puertos paralelos con la dirección en el submapa de puertos 1B00 maneja las entradas en el rango 30 a 37.

El puerto EAA físicamente se encuentra localizado en la tarjeta denominada como Tarjeta Periférica 1 del PLM y está validado por el subpaginador de puertos alterno constituido por el circuito integrado 74HC138 por la línea denominada Y0a, verificada en bajo (localizado en la misma tarjeta) que decodifica la zona del submapa de puertos de la CC en el rango de direcciones 1B80 a 1B8F en la figura 3.7 se muestra el ensamble del puerto EAA.

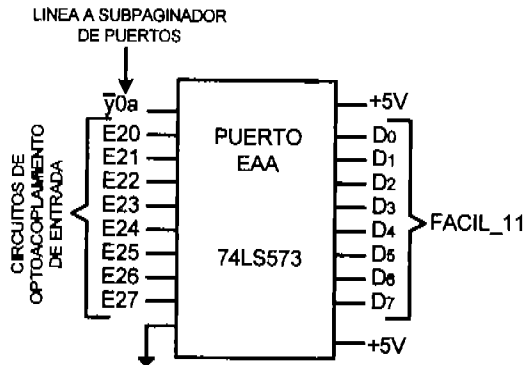


Figura 3.7.- Conexiones del puerto paralelo adicional EAA.

El puerto EAB físicamente se encuentra localizado en la tarjeta denominada como Tarjeta Periférica 2 del PLM, así como los circuitos de optoacoplamiento de entrada que habilita. El puerto EAB está validado por la línea Y6, verificada en bajo (línea que pertenece al subpágina de puertos), que se encuentra presente en las tarjetas FACIL_11, Tarjeta Periférica 1 y Tarjeta Periférica 2, en un conector de 40 postes que es común en las tres tarjetas comunicadas mediante cable plano. El puerto EAB tiene la dirección 1B00 y la línea Y6 decodifica la zona del submapa de puertos de la CC en el rango de direcciones 1B00 a 1B7F, en la figura 3.8 se muestra el ensamble del puerto EAB.

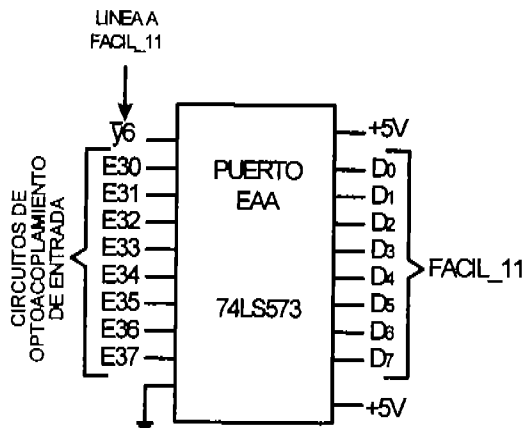


Figura 3.8.- Puerto paralelo adicional de entradas EAB.

3.2.2 Bloques funcionales asociados con el BCLD.

En el tema 3-2 del presente capítulo se describe, a nivel del usuario final, la forma en que está integrado el Bloque de Comando Local y Despliegue (BCLD), en la figura 3.9 se muestra el esquema electrónico del subpaginador de puertos requerido y el reloj de tiempo real empleado.

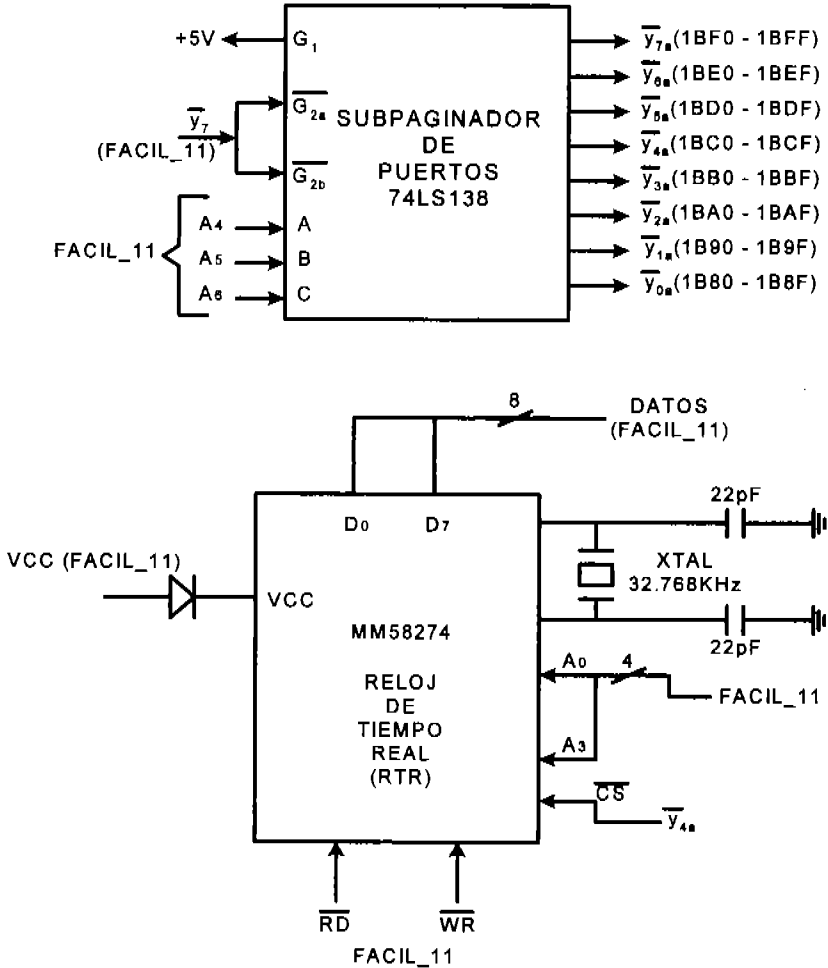


Figura 3.9.- Subpaginador de puertos y el reloj de tiempo real del BCLD.

En la figura 3.10 se muestran la conexiones de los dos puertos auxiliares y la unidad de despliegue.

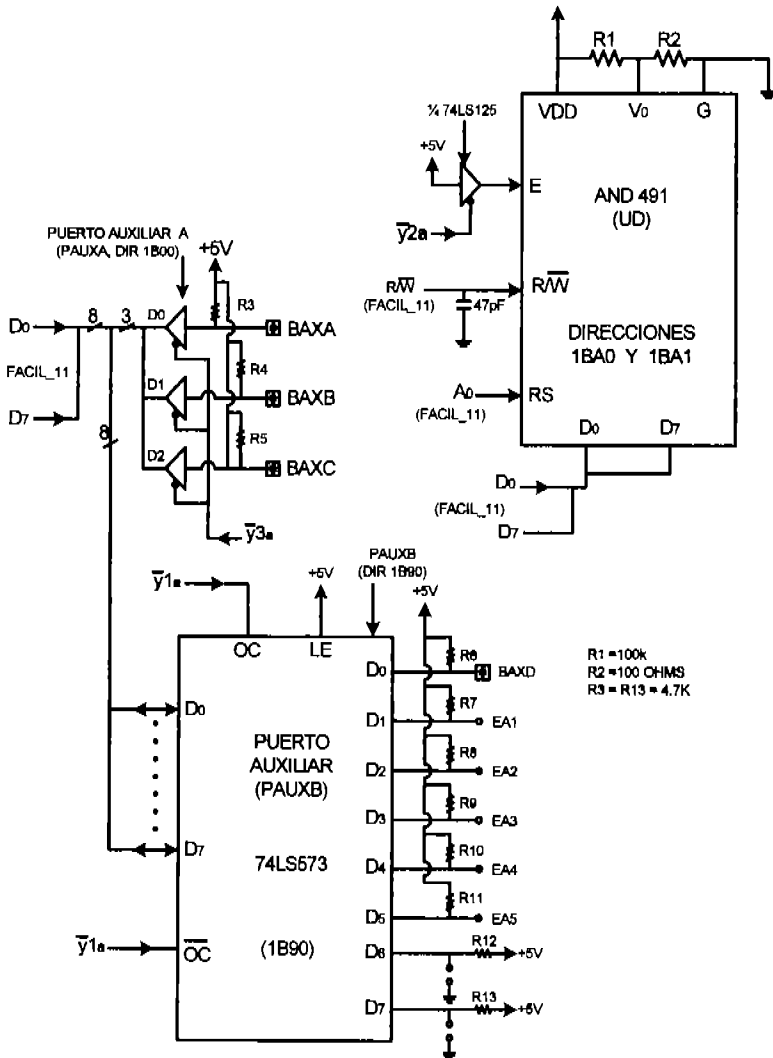


Figura 3.10 .- Puertos auxiliares y unidad desplegada del BCLD.

En la tabla 3.1 se resumen, en lo general las funciones de los botones de comando local y las entradas binarias auxiliares.

Bloque de comando local y despliegue BCLD	Uso en el PLM desde el punto de vista del usuario final
Botones BAXA, BAXB y BAXC	Ajuste y puesta a tiempo del reloj de tiempo real (RTR)
Botón BAXD	Este botón se emplea para desplegar secuencialmente mensajes priorizados en la UD
Entradas auxiliares EA1 a EA5	Reservadas para funciones futuras que pudieran requerir botones o puentes.
Puente Ja	Con Ja no colocado, al reinicializar el programa del usuario el reloj de tiempo real se pone en ceros (00:00:00), en otro caso el RTR conserva la hora al reinicializar el programa del usuario.
Puente Jb	Con Jb no colocado, al reinicializar el programa del usuario, se ponen en cero todas las variables booleanas que use la aplicación, en otro caso las variables conservan el valor que tenían antes de la reinicialización.

Subpaginador de puertos (SP).

Este bloque del BLCD está constituido por un decodificador de tres a ocho (74LS138), cada una de las ocho salidas validan sendas señales de habilitación que se verifican en bajo para un intervalo de 16 direcciones, para mejor comprensión de lo anterior, puede confrontarse la información de la tabla de funcionamiento del decodificador con las señales de entrada al mismo, mostradas en las figuras 3.9 y 3.10 con la descripción del mapa de puertos y señales de habilitación asociadas para la tarjeta FACIL_11 descrita en el capítulo dos. Es necesario aclarar que no todas las líneas de salida del subpaginador de puertos se encuentran en uso, de las ocho líneas disponibles solo se ocupan cuatro en este prototipo y estas son las siguientes:

La salida Y0a que se verifica en el intervalo de direcciones 1B80 a 1B8F, está línea es utilizada para habilitar el puerto de entradas E20 a E27, la salida Y1a que se verifica en el intervalo de direcciones 1B90 a 1B9F está línea se utiliza para habilitar el Puerto Auxiliar B, la salida Y3a que se verifica en el intervalo de direcciones 1BB0 a 1BBF está salida es utilizada para habilitar el Puerto Auxiliar A y por último la línea Y4a que se verifica en el intervalo de direcciones 1BC0 a 1BCF es utilizada para habilitar el RTR.

Puerto Auxiliar A (PAUXA).

El puerto auxiliar A es un puerto de entrada de tres bits, implantado con tres de los cuatro seguidores binarios con salida de tercer estado, contenidos en el chip 74LS125, la dirección asociada con este puerto puede ser cualquiera de las comprendidas en el intervalo 1BB0 a 1BBF, las entradas correspondientes están asociadas con los botones auxiliares A, B y C que son empleados por el software del reloj de tiempo real (RTR).

Para habilitar el poder poner a tiempo el RTR del PLM, existen dos tipos de módulos auxiliares (Existen módulos para el PLM denominados como auxiliares, que sirven para manejar facilidades que no son propiamente funciones lógicas, para varios de ellos la declaración correspondiente no requiere operandos MA) denominados como RTRA y RTRC; la diferencia entre ambos radica en el hecho de que para el primero el estado del RTR (hora y fecha) es visible en la UD, ocupando la fecha las columnas nueve a la dieciséis del primer renglón y la hora las mismas columnas en el renglón dos, véase la figura 3.11, mientras que para el MA RTRC el estado del RTR no es visible en la UD; ambos MA contienen código que permite el poder poner a tiempo el RTR empleando para ello únicamente los tres botones BAXA, BAXB y BAXC, ligados al puerto auxiliar A del bloque de comando local y despliegue (BCLD) del PLM.

										D	0	0	4	0	4	0	4
										1	9	:	1	5	:	3	7

Figura 3.11 .- Aspecto que tendría el despliegue, del estado del RTR en la UD, cuando la fecha y hora son los indicados, al ejecutarse en el PLM un programa que contenga la declaración del MA RTRA.

Para poner a tiempo el RTR se deben llevar a cabo los siguientes pasos:

- 1 Oprimir el botón "PA" (BAXA), esto hace que aparezca en la UD el estado del RTR al instante de la opresión sin mostrarse avance en el tiempo, en todos los siguientes pasos salvo el último, el botón PA (puesta a tiempo) deberá permanecer oprimido.
- 2 Oprimir el botón "A" (BAXB), esto hace que aparezcan sucesivamente en la posición de los caracteres que denotan el año, los siguientes pares de caracteres: SE, indicando que al

oprimir el botón "D" (BAXC) los dígitos indicadores de los segundos avanzan en forma natural; MI, indicando que al oprimir el botón D los dígitos indicadores de los minutos avanzan en forma natural; HO, indicando que al oprimir el botón D los dígitos indicadores de las horas avanzan en forma natural; DS, indicando que al oprimir el botón D los caracteres indicadores del día de la semana avanzan en forma natural; DM, indicando que al oprimir el botón D los dígitos indicadores del día del mes avanzan en forma natural; ME, indicando que al oprimir el botón D los dígitos indicadores del mes avanzan en forma natural; repitiéndose lo anterior en forma cíclica mientras el botón A permanezca oprimido. Por defecto, el avance que queda habilitado al oprimirse el botón PA es el correspondiente al par de dígitos que denota el año, no habiendo testificación de avance como las indicaciones en el párrafo anterior.

- 3 Soltar el botón A cuando en las columnas quince y dieciséis del renglón uno de la UD, aparezcan los caracteres que denotan al par de dígitos que se desea ajustar; por ejemplo si se desea ajustar los dígitos que indican "horas", el botón A deberá soltarse cuando en la esquina superior derecha de la UD aparezcan el par de letras "HO", véase la figura 3.12.
- 4 Oprimir el botón "D" (BAXC), hasta que el par de dígitos seleccionado llegue al valor deseado.
- 5 Repetir los pasos dos, tres y cuatro, para cada par de dígitos indicadores del estado del RTR que se desee ajustar.
- 6 Soltar el botón "PA", esto hace que la hora y fecha indicada por el usuario sea copiada al RTR.

								D	O	0	4	0	4	H	O
								1	9	:	2	5	:	4	6

Figura 3.12.- Aspecto de la pantalla de la UD, indicando que el par de dígitos indicadores de las "horas", se incrementarán al oprimirse el botón "D"

Puerto Auxiliar B (PAUXB).

Este puerto de entrada está implantado por el circuito integrado 74LS573, que es un latch octal con salida de tercer estado, las direcciones asociadas con este puerto son las comprendidas en el intervalo 1B90 a 1B9F, observándose que el bit menos significativo está ligado con el botón auxiliar D, que se usa para desplegar secuencialmente mensajes priorizados, que pudieran haber

sido programados por el usuario en el programa fuente en SILL de acuerdo con necesidades propias de la aplicación que el PLM estuviera realizando en un momento dado, para habilitar esto el MA debe declararse como parte del subprograma temporizado; así, al estarse ejecutando el programa y oprimirse sucesivamente el botón BAXD del BCLD se mostrarán, uno a la vez, los diferentes textos asociados con los módulos de tipo mensajero que se hubieren declarado, especificando el usuario el número máximo de módulo mensajero implicado, los textos irán apareciendo en el orden ascendente, al oprimirse BAXD cuando se esté desplegando el mensaje cuyo módulo asociado tiene el valor máximo definido, se regresa al despliegue del mensaje correspondiente al módulo mensajero número cero, repitiéndose el ciclo siempre que el usuario continúe oprimiendo el botón BAXD.

Las líneas de entrada uno a la cinco de este puerto están ligadas con las entradas auxiliares EA1 a EA5, por último las entradas seis y siete están asociadas con los interruptores Ja y Jb respectivamente en la tabla 3.1 se describe la finalidad de estos últimos.

Unidad de Despliegue (UD).

La unidad de despliegue, tiene una capacidad de dos renglones de 16 caracteres cada uno y está realizada por el display modelo TM162AAC6-1 fabricado por la corporación TIANMA Microelectronics Co; Ltd, en la figura 3.10 se aprecia el conexionado de la UD a la tarjeta FACIL_11. La UD del PLM puede realizar módulos denominados de tipo mensajero, con capacidad para generar texto, predefinido por el usuario. El módulo mensajero es deshabilitado siempre que el usuario tenga oprimido el botón auxiliar "A" (BAXA), el cual es empleado para habilitar el poder poner a tiempo el RTR mediante los botones auxiliares BAXB y BAXC, presentes en el BCLD del PLM.

Reloj de Tiempo Real (RTR).

El reloj de tiempo real del PLM es implantado con el chip MM58274 fabricado por National; este chip está pensado para interfazarse con microcontroladores o microprocesadores, siendo la base de tiempo del mismo generada mediante el auxilio de un cristal externo de 32768 Hertz, este CI tiene capacidad para manejar un formato de tiempo en horas, minutos, segundos, décimas de segundo, además de manejar días, meses y años en la figura 3.9 se puede observar el conexionado del RTR a la tarjeta FACIL_11.

3.3 PRUEBA DE LOS CIRCUITOS DE ENTRADA Y AUXILIARES.

Con fines experimentales se hizo la conexión en tarjetas protoboard de los 32 circuitos de optoacoplamiento de entrada; se armaron cuatro puertos de ocho entradas y ocho salidas, los puertos fueron implementados con los circuitos integrados 74LS573, cabe señalar que en la tarjeta de circuito impreso final solo se armaron dos de los puertos, ya que la tarjeta FACII_11 utilizada como la CC del PLM cuenta con dos puertos paralelos para manejar 16 de los circuitos de entrada. Para realizar el direccionamiento de estos cuatro puertos se empleo como paginador un circuito integrado 74LS138, que es un decodificador de tres a ocho el cual no es necesario en el prototipo final, se armaron también los puertos auxiliares PAUXA y PAUXB y para coordinar el funcionamiento de todo lo anterior se armo el CI que en el PLM sería el subpaginador de puertos. Para comunicar toda este circuito con la CC se armaron dos conectores con cable plano uno de cuarenta postes, mediante este conector se tiene acceso a líneas de puerto del microcontrolador, y otro conector llamado conector auxiliar de expansión mediante el cual se tiene acceso a líneas de control de selección de puerto y polarización; Es necesario aclarar que para fines experimentales en este circuito se utilizo como CC la tarjeta SIMMP-2, no siendo esta la tarjeta utilizada en el dispositivo final, sin que esto significara en el futuro ningún problema con la operación del PLM. El circuito armado en tarjetas protoboard se muestra en la figura 3.13.

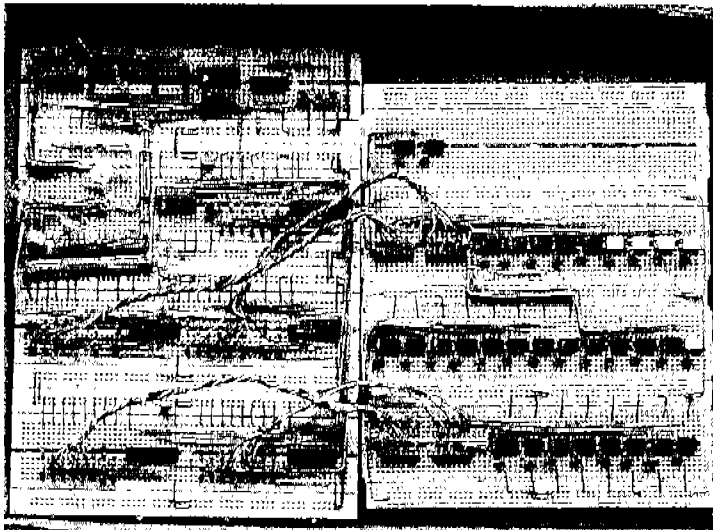


Figura 3.13 .- Fotografía del alambrado de los circuitos de entrada y circuitos auxiliares del PLM en tarjetas protoboard.

3.3.1 Objetivos de armar los circuitos en tarjetas experimentales.

Los objetivos de armar en tarjetas experimentales todos los circuitos que constituyen el PLM (circuitos de entrada, auxiliares y salidas) son primordialmente los siguientes:

- 1 Verificar que toda la lógica funcione de acuerdo a las expectativas
- 2 Hacer pruebas sencillas; como medir el consumo de corriente, dato que posteriormente es útil para realizar el diseño de las fuentes de poder que alimentan los circuitos de entrada y salida.
- 3 Conocer el espacio físico que ocupan los componentes, lo cual es de utilidad para hacer la planeación del diseño de las tarjetas de circuito impreso.
- 4 Obtener una primera aproximación del costo de manufactura del PLM.

3.4 DISEÑO DEL CIRCUITO IMPRESO PARA LA TARJETA PERIFÉRICA 1.

El diseño de un proyecto electrónico o prototipo se puede resumir en las siguientes etapas: la prueba del circuito armado en protoboard, el diseño esquemático del circuito, el diseño del circuito impreso, la fabricación del circuito impreso, el ensamble de componentes y el diseño del gabinete.

Hasta aquí ya se han cubierto las dos primeras etapas de diseño del proyecto, para desarrollar la fase correspondiente al diseño del circuito impreso, como primer paso se plantearon las siguientes preguntas:

- ¿Qué dimensiones físicas serían las apropiadas para la tarjeta?
- ¿Cuáles serían los componentes que se montarían en el circuito impreso y de que manera se organizarían sobre la superficie disponible en la tarjeta?
- ¿Cuál sería el software que se emplearía para desarrollar el dibujo del circuito impreso?

En los siguientes párrafos se responde a cada una de las preguntas planteadas anteriormente.

3.4.1 Dimensiones de la tarjeta de circuito impreso.

Para definir las medidas físicas que nos resultarían adecuadas en el diseño de la tarjeta de entradas, se tomaron como referencia las dimensiones de la tarjeta FACIL_11 que funcionaría como

CC del PLM. Se concluyo que lo más adecuado sería que la tarjeta dispuesta para albergar los circuitos de entrada, a sí como las tarjetas de circuitos de salida y la diseñada para contener las fuentes de alimentación tendrían las mismas dimensiones que la FACIL_11; esto nos es de utilidad para facilitar la integración de las tarjetas de manera paralela en el interior del gabinete (sobre el diseño del gabinete se dará información en el capítulo cuatro), por último las medidas de la Tarjeta Periférica 1 son las siguientes 7.1 pulgadas por 4.5 pulgadas.

3.4.2 Integración de los componentes en la tarjeta.

Ya que se ha definido la superficie con la que se cuenta para distribuir los componentes, procedemos a decidir cual es la mejor manera de utilizar el espacio.

El primer paso para decidir como se efectuará la distribución de los componentes sobre la tarjeta, es definir cuales serán los elementos que se colocaran sobre la superficie.

Se decidió incluir en la Tarjeta Periférica 1 los siguientes elementos; 24 del total de los 32 circuitos de optoacoplamiento (por razones de falta de espacio fue necesario desplazar el último de los cuatro puertos de entrada hacia la Tarjeta Periférica 2), los puertos EAA, PAUXA, PAUXB, el RTR, tres conectores de 40 postes cada uno y por último un conector molex de dos postes que es utilizado para habilitar la sección de la fuente de poder +5V_{cc} que alimenta todos los CI de la tarjeta.

La organización de los circuitos presentes en la Tarjeta Periférica 1 se realizo pensando en cumplir con la necesidad de obtener una rápida identificación de los sectores funcionales con que cuenta la tarjeta.

A continuación se hace la descripción de la localización de los componentes en la tarjeta. En la figura 3.14 se muestra una vista de la plantilla de la Tarjeta Periférica 1, en dicha figura se aprecia la localización de los diversos componentes que integran la arquitectura de la tarjeta.

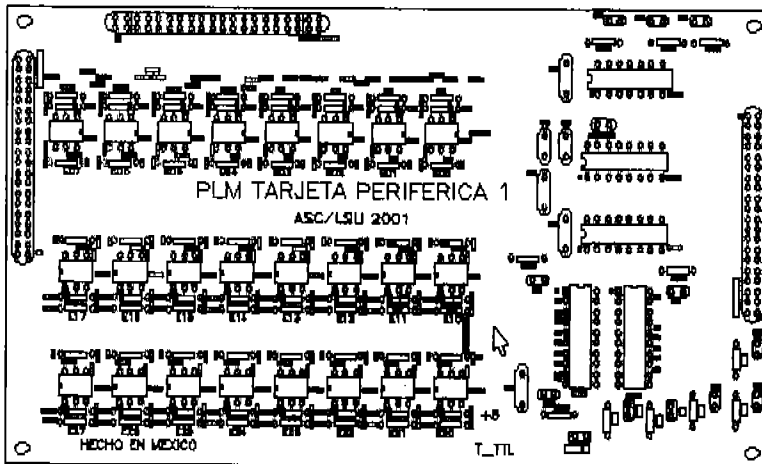


Figura 3.14 .- Vista de la Tarjeta Periférica 1.

A la izquierda de la tarjeta aparecen tres líneas verticales con ocho circuitos de optoacoplamiento por línea que son los asociados a tres de los cuatro puertos de entrada del PLM, dichos puertos manejan las siguientes entradas: E00 a E07, E10 a E17 y el E20 a E27. En la parte superior de la tarjeta, exactamente sobre las líneas de los puertos, ya mencionados se encuentra un conector de 40 postes, que en la Tarjeta Periférica 1 está nombrado como conector 2; mediante éste se tiene acceso a las entradas del PLM para la conexión de los sensores que una aplicación requiera, en éste conector se encuentra también la línea de tierra de la fuente de sensores de entrada. Este conector lo encontramos también en la Tarjeta Periférica 2 aunque recibe otro nombre (conector_3). En la figura 3.15 se observa el esquema del conector 2.

A la izquierda de las líneas de circuitos de optoacoplamiento se localiza el conector 1, en este conector también de 40 postes, se localizan las entradas TTL de los optoacopladores lo encontramos también en la Tarjeta Periférica 2 y en la FACIL_11, en la figura 3.15 se muestran las señales que maneja este conector.

En el lado derecho de la Tarjeta Periférica 1 se localizan los siguientes circuitos; en la parte inferior, primeramente se observa un conector de polarización de dos terminales, en donde se ha de conectar la fuente de $5V_{ca}$ (fuente que polarizara todos los CI de la Tarjeta Periférica 1), a la derecha

de éste conector encontramos una serie de 8 pares de postes y ocho resistencias, en donde cinco pares de postes corresponden a cinco entradas binarias auxiliares (entradas auxiliares EA1 a EA5) y los restantes 3 pares de postes corresponden; un par al botón BAXD, otro par al interruptor Ja y el último al interruptor Jb; todos estos componentes se encuentran coordinados por el puerto PAUXB. Hacia arriba de estos componentes se encuentran 2 circuitos 74HC573 que funcionan como los puertos EAA y PAUXB. Hacia el centro del sector derecho de la Tarjeta Periférica 1 se encuentran el RTR (CI MM58274) y el subpaginador de puertos (CI 74HC138). En la parte superior derecha de la tarjeta se localiza el puerto PAUXA (CI 74HC125) y a la derecha de éste componente se encuentran 3 resistencias y 3 pares de postes que son utilizados para implementar los botones BAXA, BAXB y BAXC; que son coordinados por el puerto PAUXA. En la parte extrema derecha se encuentra el conector 3 (conector de 40 patas), éste conector es común en las tarjetas Periférica 1, Periférica 2 y FACIL_11, en éste conector se tiene acceso a líneas de puerto del microcontrolador HC11F1, a líneas de control de selección de puerto de la tarjeta FACIL_11 y líneas de polarización. En la figura 3.15 se muestra en detalle las líneas asociadas con éste conector.

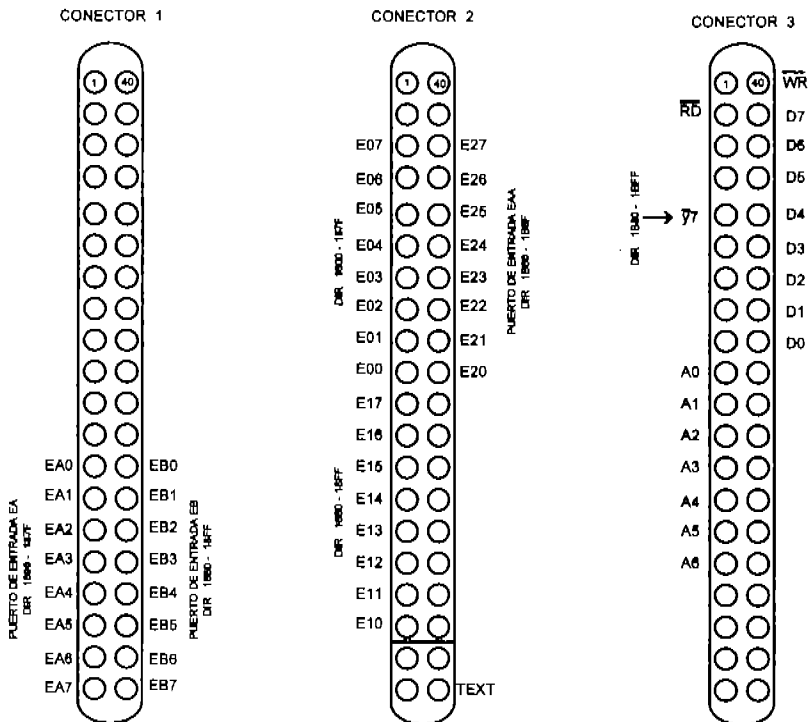


Figura 3.15.- Conectores incluidos en la Tarjeta Periférica 1.

3.4.3 Selección del software para generar el dibujo del circuito para la Tarjeta Periférica 1.

Antes de la aparición de los paquetes de diseño, los diseñadores solo contaban con su ingenio y un buen equipo de dibujo para transportar al papel sus ideas. Es quizás por éste motivo, por el que los primeros paquetes de diseño surgieron como replica a éstos buenos dibujantes, conforme el hardware evolucionaba y disminuían los costos de los equipos, los programas eran más rápidos, apareció un fenómeno de insatisfacción en los usuarios, un buen programa de dibujo no bastaba, era necesario un sistema que diseñara el producto desde el principio (dibujar el esquema) hasta el final (placa de circuito impreso terminada), siguiendo unas reglas de diseño.

Como consecuencia de éstas necesidades surgieron los paquetes de Ingeniería Asistida por Computadora (sus siglas en ingles son CAE), en general un sistema CAE para el diseño de circuitos impresos entre otras se emplean las reglas de diseño siguientes:

1. Capturas de dibujos esquemáticos
2. Diseño de circuitos analógicos y digitales
3. Simulación lógica y analógica de dichos circuitos
4. Análisis térmico
5. Diseño del PCB (dibujo de circuito impreso)
6. Proceso de electromecánica

La etapa dentro del sistema de software CAE que nos ocupa es el diseño del PCB, para éste punto existe software diseñado para la creación de circuitos impresos, algunos de los paquetes que realizan éste trabajo son:

CAE Auto Trax
Win Board
Protel Advanced PCB
Workbench Layout
OrCAD/PCB
Ez-Route
Smart Work
Specetra
Tango PCB de Accel

Dentro de los paquetes de software que se listaron anteriormente el software que se utilizó para crear el diseño de la placa de circuito impreso es el paquete Tango PCB de Accel Technologies Inc, fue seleccionado por los siguientes motivos; de inicio se contaba ya físicamente con el programa, por otra parte el software es de un sencillo manejo de comandos, se puede manejar por teclado y por ratón y por último cuenta con muy buena presentación.

Tango/PCB es un programa popular en el diseño de circuitos. Es un flexible y completo paquete de diseño de circuitos, fácil de usar que se apoya en un menú que maneja comandos para la ayuda de creación, edición, impresión, respaldo y trazado de diagramas de conexión de circuitos, desarrollado para correr en PC IBM y compatibles. Éste paquete elimina la necesidad de un hardware especial o equipo fuera de lo standard (La versión de Tango que se utilizó fue la 1.6).

Éste programa que auxilia a crear, editar y respaldar hojas de trabajo de PCB, cuenta con una serie de menús y submenús en los cuales maneja variados comandos (para ampliar la información sobre este programa consultar el apéndice A). Están incluidas en Tango/PCB algunas librerías que usan dispositivos de los más comúnmente utilizados en la industria.

Ejemplo de estas librerías son:

TTL

Componentes discretos

Componentes analógicos

Microcontroladores

3.4.4 Elaboración del diseño del circuito impreso.

Para el diseño de la Tarjeta Periférica 1 del PLM (ver la figura 3.16) se trabajó de la siguiente manera:

Las caras para elaborar la placa del circuito impreso que se utilizaron fueron:

Top Layer: Cara superior para el trazado de pistas.

Bottom Layer: Cara inferior para el trazado de pistas.

Board Layer: Cara para la delimitación del área de la placa.

Top Silk: Cara para componentes normales en encapsulado tipo dip y colocación de etiquetas.

El ancho mínimo para el trazado de pistas es de 12 mils (sobre la unidad mil se hace referencia dentro de ésta misma sección en párrafos siguientes). La distancia mínima entre las pistas, pads (Pad; agujero utilizado para insertar las patas de variados componentes) y vias (Via; agujero utilizado para conectar las caras de conexionado Top y Bottom) es de 12 mils. Las medidas para los pads son 58 mils para el diámetro exterior y 36 mils para el diámetro interior, para las vias las medidas son las siguientes; para el diámetro exterior 50 mils y para el diámetro interno 32 mils.

Las medidas que se utilizaron para las pistas, pads y vias, así como las distancias entre los mismos elementos, se establecieron por recomendación del fabricante de la tarjeta de circuito impreso. Los parámetros de medida para los distintos elementos (pads, vias y pistas) que se utilizaron en la elaboración del dibujo se pueden modificar; pero si se reducen las medidas y distancias entre los componentes del dibujo los costos de producción se elevan al hacerse más complicada la fabricación de la tarjeta.

La unidad de medida que se maneja en Tango para el diseño de la placa de circuito impreso fue la llamada mils. El sistema mil está basado en milésimas de pulgada, la separación entre dos agujeros la placa de intersección (protoboard) es exactamente de 100 mils (100 mils = 1 agujero), ésta unidad es sumamente práctica, ya que permite utilizar la placa de protoboard como patrón de medida para conocer cual es el espacio que ocupara en el dibujo casi cualquier componente. Para los dispositivos en los que no era posible utilizar éste sencillo método de medida se empleo un vernier. Éste sistema de medida nos permite obtener un ajuste perfecto al momento de montar los componentes sobre la superficie de la tarjeta.

Tomando en cuenta todo lo expuesto en los párrafos anteriores se elaboró el dibujo del diseño de la Tarjeta Periférica 1, y con esto concluimos la etapa de diseño del circuito impreso, sigue la etapa de fabricación, esta fase es efectuada por una empresa dedicada a tal propósito, a continuación se cumple con la etapa de ensamble de componentes y de esta manera obtenemos la placa final. En la figura 3.16 se muestra la Tarjeta Periférica 1 ya ensamblada.

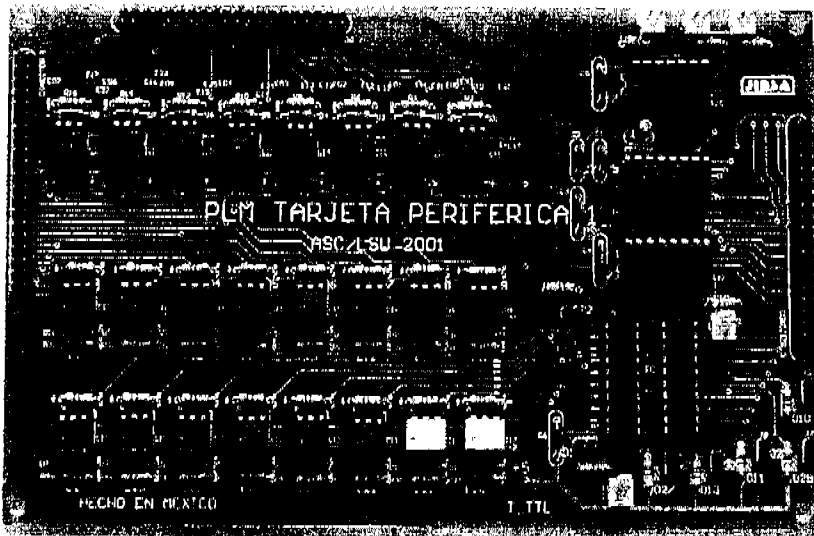


Figura 3.16.- Fotografía de la Tarjeta Periférica I terminada.

3.5 CIRCUITOS DE SALIDA.

Los circuitos de salida son circuitos electrónicos, capaces de transformar las señales procedentes de la CC en señales eléctricas utilizables por el usuario.

Hay varios sistemas para la transformación de las señales de la CC, como son las salidas por relevador, por triac o por transistor. Para el PLM se seleccionó el más usado de estos métodos que es el que implementa las salidas con pequeños relevadores electromecánicos.

3.5.1 Conceptos generales sobre relevadores.

Como ya se menciona en el segmento anterior las salidas del PLM se implementaran con relevadores, por tal motivo en esta sección del capítulo se presentan algunos conceptos sobre relevadores.

Sabemos que cuando una corriente recorre un conductor rectilíneo, el movimiento de las cargas es responsable de la aparición de un campo magnético que tiene la misma naturaleza que el y que se produce con una barra de imán permanente que puede actuar sobre objetos ferromagnéticos, atrayéndolos. En el caso del campo producido por una corriente en el conductor no solo tenemos el control de su intensidad si no también podemos intervenir en la geometría del sistema, dándole formas y disposiciones mediante las que se puede aumentar, dirigir y difundir las líneas de fuerza y el campo según se desee. Hay varias maneras de lograr eso, lo que nos lleva a la elaboración de distintos dispositivos de aplicación en electrónica.

Uno de estos dispositivos que es comúnmente utilizado es el relevador, el cual podemos definirlo como un dispositivo de conmutación eléctrica controlado por señal eléctrica.

El relevador esta formado por las siguientes partes básicas:

- 1.- El electroimán.
- 2.- Contacto común.
- 3.- Contactos móviles.

En las cercanías de un electroimán se coloca un juego de contactos electrónicos, cuando no circula la corriente por el solenoide (bobina) los contactos permanecen en un estado normal (el normalmente cerrado permanece cerrado y el normalmente abierto permanece abierto), pero si llegase a aplicarse una tensión en las terminales de la bobina los contactos móviles sufrirán un cambio; el cerrado se abrirá y el abierto se cerrará.

En la figura 3.17 se presenta el croquis de un relevador típico; estos relevadores se construyen para ser utilizados con una tensión de mando (para la bobina), comprendida entre los que trabajan con 2 volts, hasta los que trabajan con una tensión de 220 volts. La tensión puede ser de corriente continua o alterna e incluso los hay que sirven para las dos formas de corriente. La tensión para la que se ha construido siempre viene reflejada en las características del relevador, que el fabricante facilita siempre, ya sea, escritas en el propio relevador o bien en hoja aparte o en el catálogo.

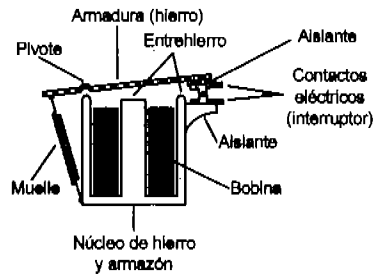


Figura 3.17.- El relevador se representa con un contacto simple, es decir; que cuando está en reposo no existe contacto alguno, pero al alimentar la bobina se efectúa el contacto. Estos suelen ser para cargas fuertes, y se llaman CONTACTORES. Suelen tener adosados o se les puede adosar también contactos conmutados. Hay relevadores en los que una sola bobina puede accionar 1, 6 y aún 8 conmutadores totalmente independientes y aislados entre sí.

Dentro de las características del relevador facilitadas por el fabricante, tienen que estar las características de los platinos: corriente máxima (los contactos soportan hasta 6 Amperes y la bobina consume 15.625 mA) que pueden soportar. Es conveniente no sobrepasar nunca lo indicado por el fabricante, ni siquiera llegar al máximo, pues la vida del relevador se acorta considerablemente en estas situaciones.

El relevador de la figura 3.17 se ve en la posición desconectada. Cuando circula una corriente por la bobina, la armadura es atraída por el núcleo de hierro. Esto cierra un juego de contactos que pueden completar otro circuito eléctrico. Cuando la corriente cesa en la bobina, el muelle empuja a la armadura y abre los contactos.

Una de las principales ventajas de un relevador sobre un simple interruptor es que permite operación remota. Un suministro a bajo voltaje y baja corriente puede controlar la bobina del relevador. Entonces el relevador con sus contactos puede controlar un circuito de alto voltaje y /o elevada corriente.

3.5.2 Descripción del circuito particular para una única salida.

Las salidas del PLM se manifiestan físicamente mediante la apertura (salida cero lógico) o cerradura (salida uno lógico) de los contactos de sendos relevadores con voltaje de activación de +12 Volts, para la bobina de los mismos. En la figura 3.18 se muestra el circuito genérico empleado para una salida del PLM.

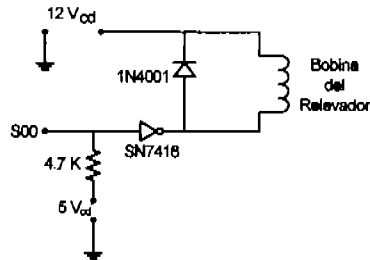


Figura 3.18 .- Circuito genérico para una salida del PLM.

La señal que viene de la CC del PLM (S00) se invierte en el buffer SN7416 y la señal a la salida de dicho circuito determina si hay o no paso de corriente en la bobina del relevador. Si a la salida del buffer tenemos un TTL la diferencia de potencial existente entre los extremos de la bobina no es suficiente para provocar la conmutación en los platinos del relevador y por lo tanto como resultado final para esta única salida tendremos un cero lógico, si por el contrario a la salida del buffer tenemos un cero TTL, esto tiene como consecuencia que la diferencia de potencial entre los extremos de la bobina permite la conmutación de los platinos (el contacto normalmente abierto cierra) y a si se obtiene como resultado un uno lógico para esta salida.

El diodo 1N4001 se coloca como dispositivo de protección para el transistor localizado en las salidas del chip SN7416. La protección se coloca porque en la situación en que la bobina del relevador pase de un estado de circulación de corriente a través del devanado a un estado de reposo se provoca un aumento transitorio muy importante de voltaje que pasaría directamente al transistor del buffer ocasionando su posible daño o disminución de vida útil, pero al encontrarse el diodo; la bobina se descarga a través de él impidiendo de esta manera el daño del transistor.

3.5.3 Organización de los circuitos de salida.

Como salidas del PLM tenemos 16 circuitos que emplean relevadores con un voltaje de activación de +12 V_{cd} y una corriente máxima en sus contactos de 6 Amperes. Las 16 salidas están agrupadas en dos conjuntos de ocho salidas.

Las salidas son denotadas empleando tres caracteres; el primero es una letra mayúscula que en este caso será la S (por el hecho de ser salida), el segundo y tercer símbolos son números, el primero de los cuales será cero o uno para determinar el grupo al cual pertenece la entrada, y el número final se encontrará dentro de un rango de valores de cero a siete que indica el número de 'bit' de salida correspondiente; por ejemplo tenemos la salida 'S07' denotará el hecho de que tendremos una salida que pertenece al grupo cero y que el número de bit es el siete. Para cada grupo de ocho salidas corresponde un puerto físico con una determinada dirección en el mapa de puertos.

3.6 PRUEBA DE LOS CIRCUITOS DE SALIDA.

Para fines experimentales se armaron en tarjeta protoboard únicamente seis circuitos de salida, esto con la finalidad de estudiar el desempeño del CI SN7416 y definir si dicho chip resultara ser adecuado para el manejo de las señales de salida del PLM.

Inicialmente se probó el chip solo para el estudio de una única salida, en la figura 3.18 se muestra la conexión entre una salida del CI SN7416 y el circuito compuesto por un diodo (1N4001) y la bobina del relevador.

Se midió la corriente que consumía el relevador al hacer la conmutación entre los dos platinos, esta corriente fue de 15.625 mA. Consultando las hojas de especificaciones para el SN7416, tenemos que soporta hasta 40 mA por cada una de sus salidas, por lo tanto el consumo que hace el relevador se encuentra dentro del rango que soporta el chip, posteriormente se conectaron las seis salidas con que cuenta el SN7416 y los correspondientes circuitos de relevador, esto para efectuar la prueba del dispositivo en la situación extrema de accionamiento simultáneo de los seis relevadores y un lapso de tiempo prolongado de trabajo. Por espacio de 5 horas se monitoreó el funcionamiento del chip; no presentándose ningún problema durante el tiempo de estudio.

Como resultado de estos sencillos experimentos se decidió emplear el dispositivo SN7416 para el interfazado de circuitos de salida en el prototipo del PLM.

Después de hacer esta sencilla prueba al chip SN7416, no se considero necesario armar en protoboard las 16 salidas y se continuo el trabajo con el diseño del circuito impreso que albergara los dos grupos de circuitos de interfazado de salida asociados con los puertos de salida del PLM, dicha placa de circuito impreso lleva el título de Tarjeta Periférica 2.

3.7 DISEÑO DEL CIRCUITO IMPRESO PARA LA TARJETA PERIFÉRICA 2.

En el tema 3.4 de este trabajo se hizo referencia a las etapas en que se puede dividir el diseño de un proyecto electrónico, para el caso de la Tarjeta Periférica 2 que se reporta en esta sección la etapa que corresponde reseñar es la tercera de las cinco en que se fracciona el diseño de un prototipo.

Tomando como referencia el tema 3.4 se necesita definir lo siguiente:

- 1 Dimensiones de la tarjeta.
- 2 Componentes que albergara el circuito impreso y su organización en la superficie de la placa.
- 3 Software que se emplea para dibujar el circuito impreso.

3.7.1 Dimensiones de la Tarjeta Periférica 2.

Las dimensiones de la Tarjeta Periférica 2 son las mismas que se emplearon en, la Tarjeta Periférica 1 (tarjeta de circuitos de entrada y auxiliares), tarjeta FACIL_11 (CC del PLM) y Tarjeta Periférica 3 (fuente de alimentación). Las razones por las cuales se adoptaron estas medidas ya se expusieron en temas anteriores. Las medidas de la tarjeta son: 7.1 pulgadas de largo por 4.5 pulgadas de alto.

3.7.2 Integración de los componentes en la tarjeta.

Se decidió incluir en la Tarjeta Periférica 2 los siguientes elementos: 16 circuitos de interfazado a relevadores de salida, 8 circuitos de optoacoplamiento que son las entradas E30 a E37, el puerto EAB, tres conectores de 40 postes cada uno (para conexión de cable plano), dos conectores molex de nueve postes y un conector molex de tres postes utilizado para habilitar las dos secciones de la fuente de alimentación (sección de +5 V_{cd} y sección de +12 V_{cd}).

La organización de los circuitos presentes en la tarjeta se hizo pensando en cumplir con la necesidad de disponer de una rápida identificación de los sectores funcionales con que cuenta la tarjeta.

A continuación se hace la descripción de los componentes que se montaran en la tarjeta. En la figura 3.19 se muestra una vista de la plantilla de la Tarjeta Periférica 2, en dicha figura se aprecia la localización de los diversos componentes que integran la arquitectura de la tarjeta.

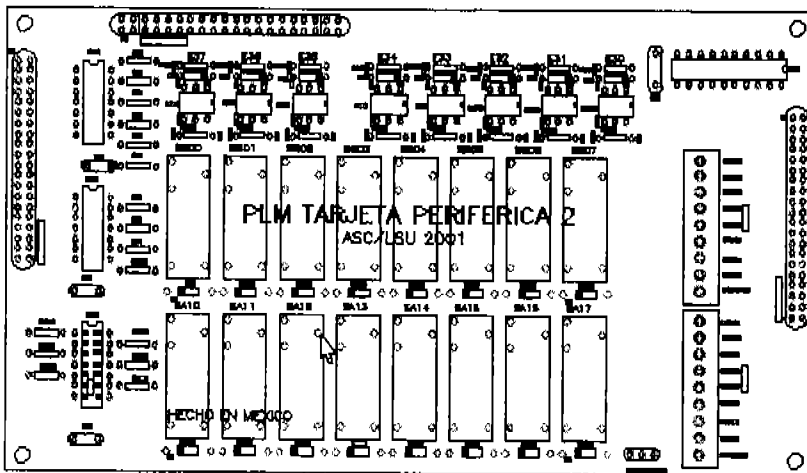


Figura 3.19 .- Plantilla de la Tarjeta Periférica 2.

En el borde superior izquierdo de la tarjeta se localiza el Conector_2; en este conector se encuentran las líneas de control de los dos puertos de salida y los dos puertos de entrada con que cuenta la tarjeta FACIL_11, de manera que este conector se localiza también en las tarjetas FACIL_11 y Tarjeta Periférica 1. En la figura 3.20 se muestran las líneas que la Tarjeta Periférica 2 ocupa de este conector. En la parte central de la placa y ocupando la mayoría de la superficie se localizan dos líneas horizontales de ocho circuitos con relevadores que aunados a los tres chips SN7416 situados a la izquierda de los relevadores y los dos conectores moles colocados a la derecha de los relevadores constituyen los circuitos asociados a los dos puertos de salida con que cuenta el PLM. En el borde superior de la placa se puede ver el Conector_3, mediante este conector se tiene acceso a las entradas del PLM para la conexión de los sensores que una aplicación requiera, en este conector se encuentra también la línea de tierra de la fuente de sensores de entrada. Este conector se encuentra también en la Tarjeta Periférica 1, solo que con el nombre de Conector 2 en la figura 3.20 se muestra el dibujo del Conector_3 con las líneas que se utilizan para la Tarjeta Periférica 2. Justo bajo el Conector_3 se encuentra una línea horizontal con ocho circuitos de optoacoplamiento que son los circuitos asociados al puerto de entrada EAB y dicho puerto (CI 74HC573) se localiza en la esquina superior derecha de la tarjeta. En el borde izquierdo de la tarjeta se puede ver el Conector_1 a través de este conector tenemos acceso a líneas de puerto del microcontrolador HC11F1, líneas de control de selección de puerto de la Tarjeta FACIL_11 y líneas de polarización, este conector es común también en la tarjeta FACIL_11 y la Tarjeta Periférica 1, en la figura 3.20 se puede ver el dibujo de este conector con las líneas que se ocupan solo para la Tarjeta Periférica 2. Finalmente se observa un conector de tres postes que sirve para conectar la fuente de poder, este dispositivo se localiza en el borde inferior de la placa hacia la esquina inferior derecha.

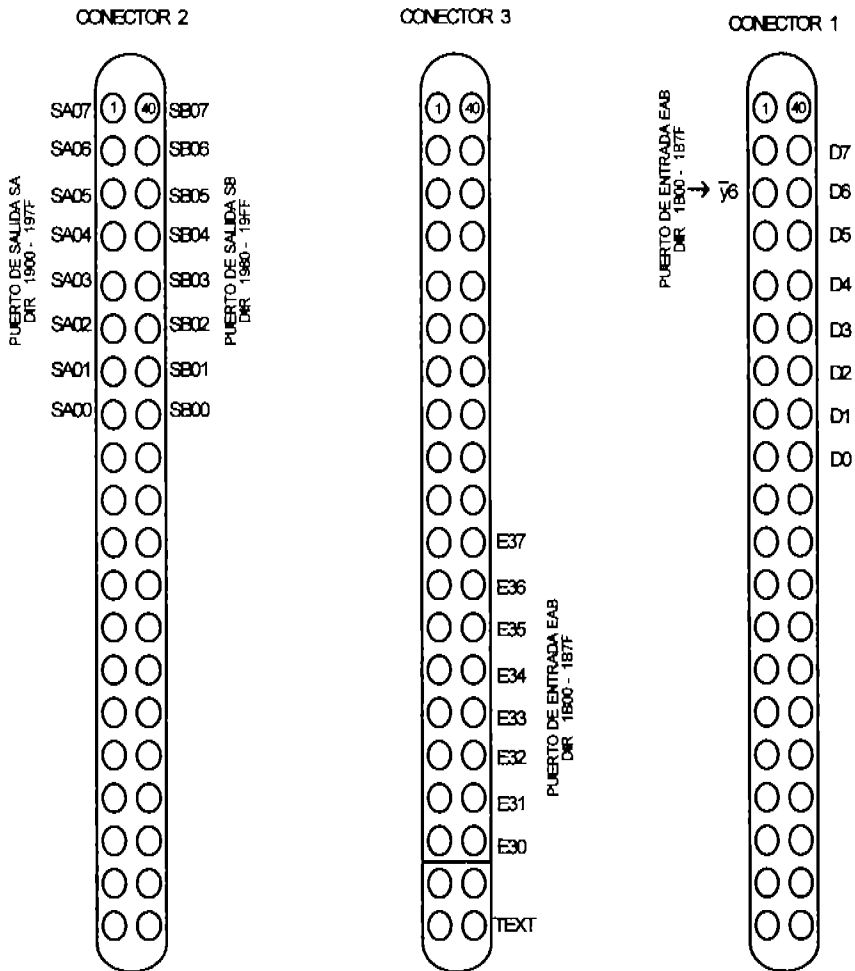


Figura 3.20 .- Conectores 1, 2 y 3 de la Tarjeta Periférica 2 del PLM.

3.7.3 Selección del software.

En el punto 3.4.3 se hizo una amplia descripción y justificación sobre la elección del software que se utilizó para realizar el diseño de todos los dibujos PCB necesarios para completar la elaboración del prototipo del PLM, desde las tres placas nombradas como: Tarjeta Periférica 1, Tarjeta Periférica 2, Fuente de Alimentación, hasta las pequeñas placas en donde se montan los bornes para los circuitos de entradas y salidas del PLM, que se colocan en la parte externa del gabinete del PLM, todo se elaboró utilizando el programa TANGO PCB VERSIÓN 1.6. En la figura 3.21 se muestra la fotografía de la Tarjeta Periférica 2 ya terminada.

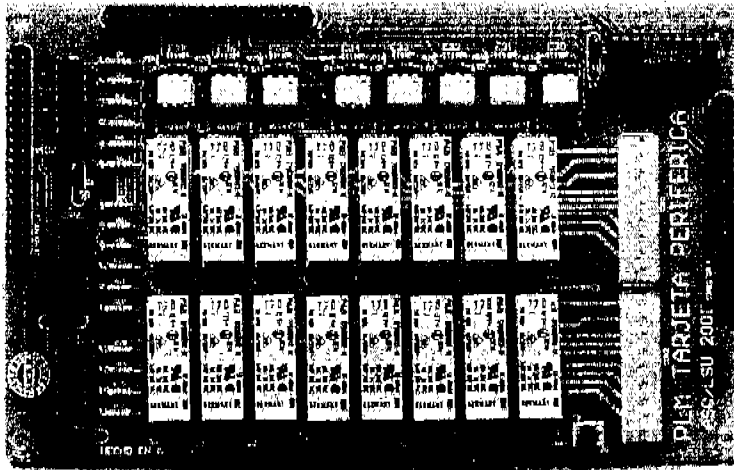


Figura 3.21.- Fotografía de la Tarjeta Periférica 2 del PLM.

CAPÍTULO 4

GABINETE DEL PLM

En el presente capítulo se hace la descripción general del diseño y construcción del gabinete que contiene al dispositivo denominado PLM, se inicia el capítulo con la enumeración y exposición de los requerimientos considerados para el diseño del gabinete y concluye el capítulo con la reseña sobre la elaboración del gabinete.

4.1 DISEÑO DEL GABINETE PARA EL PLM.

En este punto del trabajo se contaba físicamente con todos los componentes del PLM, las pruebas individuales y de conjunto para el dispositivo estaban ya concluidas de manera exitosa, es entonces el momento de decidir como se ha de presentar el PLM, esto es diseñar el medio que ha de permitir al usuario relacionarse con el instrumento, para este propósito inicialmente se definieron cuales serían las condiciones que servirán de apoyo para llevar a cabo el diseño del gabinete.

Los requisitos que se consideraron para el diseño del gabinete son los siguientes:

- 1.- Utilidad de construcción del gabinete.
- 2.- Facilidad de manejo del instrumento para el usuario final.
- 3.- Facilidad para el efectuar el ensamblaje del PLM.
- 4.- Dimensiones físicas para el diseño del gabinete.
- 5.- Costo de fabricación.

4.1.1 Utilidad de la construcción del gabinete.

Una de las utilidades del gabinete es la de proteger al instrumento del medio ambiente hay ambientes de trabajo con condiciones extremas dichas condiciones pueden ser de humedad, temperatura, presión, etc. En este caso no es objetivo del proyecto que el dispositivo se vea expuesto a ninguna exigencia extraordinaria, en cambio el diseño del gabinete tendrá las siguientes características; medios de ventilación apropiados al instrumento, protección contra el polvo así como protección en

caso de un uso poco cuidadoso por parte del usuario, el gabinete proporcionara también una manera práctica de transporte, se pretende que sea un medio para obtener una buena organización de todos los componentes del aparato, esto es, obtener del diseño tanto una buena disposición interna de las tarjetas y el cableado, así como una favorable distribución de los componentes externos del dispositivo y por último como característica deseable se ambiciona que el instrumento tenga una apariencia agradable a la vista. En las figuras 4.1 y 4.2 se muestran fotografías del P.I.M terminado por completo.

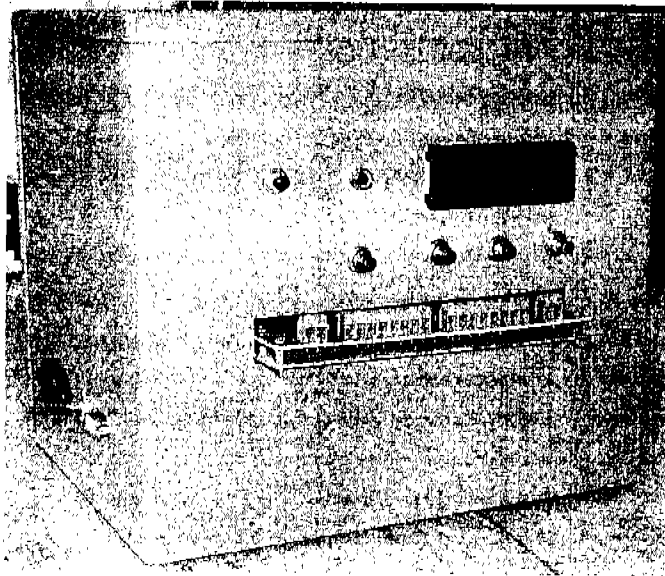


Figura 4.1.- Fotografía del P.I.M en donde se muestran la UD y los dos puertos de salida.

4.1.2 Facilidad de manejo del instrumento para el usuario final.

Para este punto se considero la armonía que debía establecerse entre los elementos del dispositivo con el fin de obtener una aceptación adecuada a las expectativas de los usuarios, se busco que existiera en el diseño una compatibilidad geométrica en la disposición del espacio que ocupan todos los elementos que componen al prototipo; para lograr este propósito se tomaron en consideración

tres condiciones la primera definir cuales serian los componentes que debían distribuirse en la superficie del gabinete y que son los siguientes; 36 terminales para los puertos de entrada, 8 terminales de tierra para la fuente de sensores, 16 terminales para los puertos de salida, 4 terminales para tierra de la fuente de actuadores, un led indicador de funcionamiento del microcontrolador, un conector a la línea de energía, un conector DB9 ala PC, un interruptor de encendido ON - OF, la Unidad Desplegadota, 3 botones de ajuste y puesta a tiempo del reloj de tiempo real (RTR), un botón empleado para desplegar mensajes priorizados en la UD y por último un botón de reinicio para el microcontrolador, la siguiente condición tomada fue la selección del material que se utilizaría para los elementos mencionados anteriormente y la última consideración fue definir como se distribuirían los componentes en el gabinete y se decidió utilizar tres de las caras del gabinete para colocar los componentes de la siguiente manera; en una de las caras que se consideraría como el frente del PLM se colocarían la UD, el led indicador y el botón de reinicio del microcontrolador en la primera línea componentes, en una segunda línea y justo debajo de la UD se colocarían los botones de ajuste del reloj y el botón para despliegue de mensajes, en la siguiente línea de componentes se colocaron 16 terminales de salida para los puertos SA y SB con cuatro terminales de tierra para la fuente de actuadotes, dos para cada puerto. En la cara posterior del gabinete se instalarían 16 terminales de entrada en una línea, para los puertos EA y EB con dos terminales de tierra por cada puerto, por último se colocaría una última línea de componentes donde se acomodarían los puertos de entrada EAA y EAB con sus cuatro terminales de tierra para la fuente de sensores, finalmente en la cara lateral izquierda del gabinete se acomodarían el conector ala línea de 127 V de AC, el interruptor ON - OF y el conector DB9 para el enlace al puerto serial de la PC.

En la figura 4.3 se muestra el dibujo de las tres caras descritas anteriormente con las dimensiones utilizadas y en las figuras 4.1 y 4.2 se muestran las fotografías de las mismas caras del gabinete en el proyecto ya concluido.

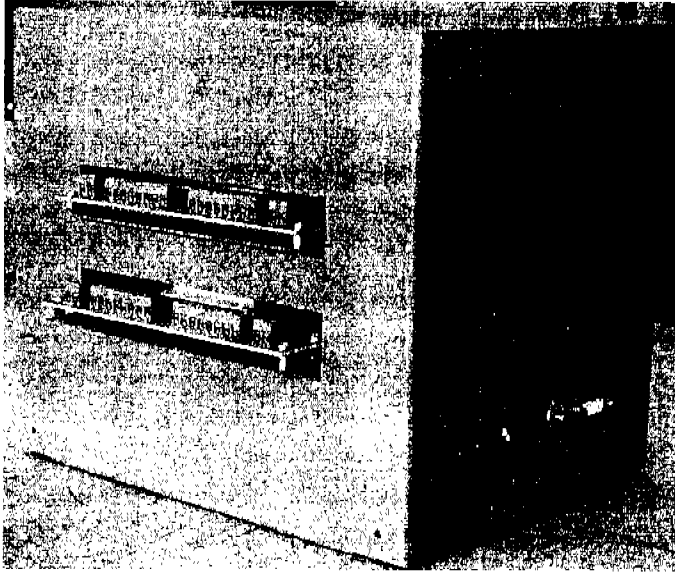


Figura 4.2 .- Fotografía del PLM en donde se muestran los cuatro puertos de entrada.

4.1.3 Facilidad para efectuar el ensamble del PLM.

El diseño del gabinete esta planeado para presentar una gran facilidad ya sea para ensamblar o desmontar el modelo, esta característica es útil en primer termino para permitir una manufactura poco complicada. pues para la mayoría de las piezas que constituyen el gabinete solo se requiere de cortes y perforaciones en material de fabricación. Por otra parte el fácil desmontaje del gabinete permite el acceso a todas las tarjetas que integran al PLM, y de esta manera se podrán hacer pruebas individuales para cada tarjeta en caso de reparación o simplemente para darle mantenimiento al instrumento. En el párrafo siguiente se describe la distribución de las tarjetas en el interior del gabinete.

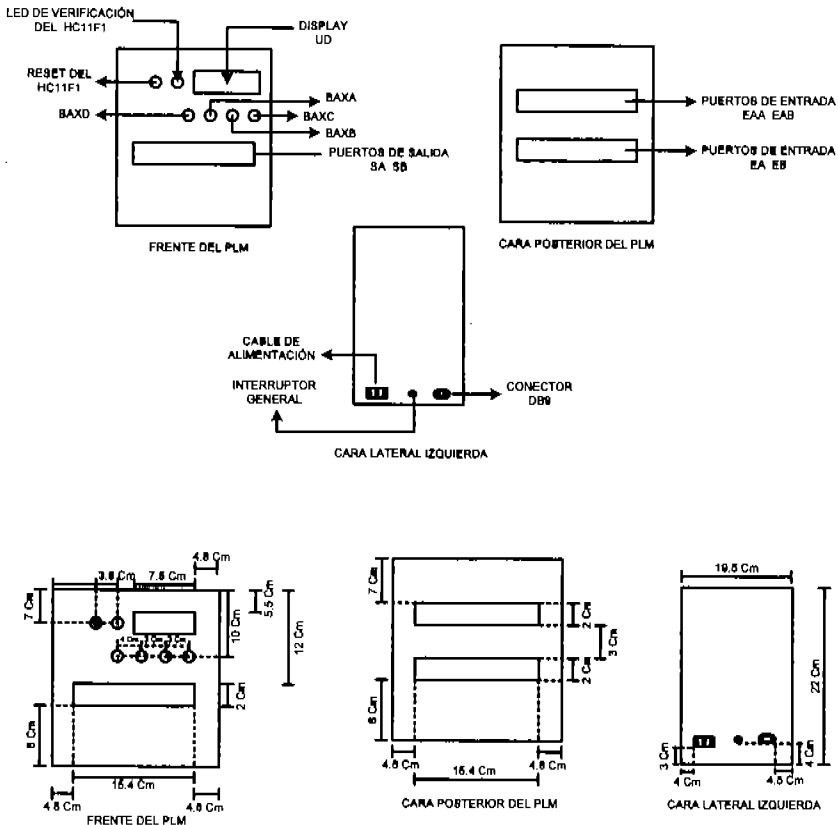


Figura 4.3.- En este dibujo se muestran alguna de las especificaciones utilizadas en el diseño del gabinete del PLM.

La tarjeta con la fuente de alimentación se localiza en el fondo del gabinete por ser esta la más pesada, luego se acomodan la Tarjeta Periférica 2 y posteriormente la Tarjeta Periférica 1 ya que su interconexión es más práctica empleando este orden, en la parte superior del gabinete se localiza la tarjeta FACILL_11 pues es necesario tener un acceso inmediato a la tarjeta para poder efectuar cambios en la programación de la memoria EPROM del microcontrolador, en la figura 4.4 se muestra

como se colocaron las tarjetas en el interior del gabinete. En las siguientes líneas se hace la descripción de las piezas que integran al gabinete.

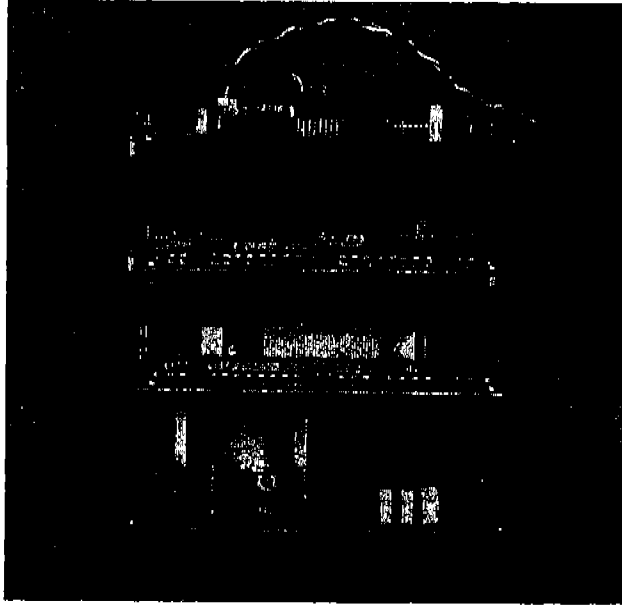


Figura 4.4 .- Fotografía del PLM en donde se muestra como se hizo la distribución de las tarjetas en el gabinete.

El gabinete esta constituido de la siguiente manera; consta de cuatro placas que se acomodaran en el interior del prototipo de manera horizontal, estas placas albergan a cada una de las tarjetas que forman al PLM (Tarjeta de las fuentes de poder, Tarjeta Periférica 1, Tarjeta Periférica 2 y Tarjeta FACILL-11), las dos placas centrales cuentan con soportes pala colocar los circuitos impresos que contienen los cuatro puertos de entrada del PLM y los dos puertos de salida del mismo dispositivo. El cuerpo del gabinete esta formado por dos placas una ellas es una lamina que por medio de dobleces en el material proporciona tres de las paredes del gabinete, la segunda lamina esta doblada en un ángulo de noventa grados y proporciona dos paredes para el gabinete, una de estas paredes funciona como tapa

del modelo y en ella se localiza una rejilla de ventilación para el instrumento, la cara restante funciona como la cara posterior del PLM. En la figura 4.5 se muestra una vista del PLM.



Figura 4.5 .- Fotografía que muestra el interior gabinete del PLM.

4.1.4 Dimensiones físicas para el diseño del gabinete.

Las medidas del gabinete son 25 centímetros de largo, 22 centímetros de ancho y 19.5 centímetros de profundidad, para decidir que estas eran las dimensiones adecuadas para el diseño se tomo en consideración lo siguiente; en primer termino las medidas de las cuatro tarjetas que componen al PLM que son de 18 centímetros por 11.1 centímetros, también se tomo en consideración el espacio ocupado por las conexiones que deben hacerse entre todas las tarjetas, además de la facilidad que debía proporcionar el modelo para acceder manualmente cualquiera de sus partes para proporcionar al dispositivo mantenimiento, ventilación y reparaciones en caso de fallas, todas dimensiones que se consideraron para el diseño del gabinete se muestran en la figura 4.6

Es posible reducir en algunos centímetros las dimensiones del PLM; esto se podría lograr eliminando la tarjeta que contiene las fuentes de alimentación sustituyéndola por fuentes de poder externas (eliminadores de baterías), esta modificación podría reducir en 8 centímetros la altura del dispositivo y daría al PLM sensiblemente más ligero, en versiones posteriores del instrumento esta podría ser una opción a considerar.

4.1.5 Costo de fabricación del gabinete.

El costo de fabricación es sin duda un factor determinante en la elección del material en que debe ser fabricarse el gabinete del instrumento. Para decidir el material con el que debía producirse el modelo se consideraron las siguientes posibilidades; lamina de acero y acrílico; en el caso del acrílico el costo era del doble que la opción en lamina (2000.00 \$ en acrílico, 1000.00 \$ en lamina), aunado a este inconveniente se tenía que la resistencia del gabinete fabricado en acrílico resultaría ser menor que la alternativa en lamina, ya que en acrílico las paredes del modelo se mantendrían unidas con pegamento, además el costo de manufactura no se reduciría en nada si se solicitaba la fabricación de un número mayor de piezas, el costo permanecería idéntico en la manufactura ya sea de una pieza, diez o cien; por estos motivos se eliminó la opción en acrílico.

La opción en lamina resultó ser la más práctica ya que nos proporciona más protección para los componentes del PLM, un precio más accesible, mayor facilidad de ensamblaje y la certeza de que en la producción de un número mayor de piezas los costos bajarían en la medida en que aumentara el número de piezas solicitadas.

4.1.6 Costo de fabricación del prototipo.

El costo de manufactura del PLM lo he dividido en cinco aspectos, estos son; el costo de los componentes electrónicos, el costo de las tarjetas de circuito impreso, el costo del gabinete, el costo de diseño y el costo de ensamblaje. Para los tres primeros puntos no hay ningún problema para determinar la suma que hay que invertir, para establecer el costo del ensamble tendría que considerar el tiempo que se requiere para armar el modelo y cuanto se pagaría por unidad de tiempo empleada y por último el aspecto más difícil de valorar es el costo de diseño que por supuesto es el más elevado. En la tabla 6.1 se muestra la descripción de los recursos empleados para manufacturar el prototipo.

Tabla 4.1 Costo de manufactura del PLM (Las cifras incluyen IVA)

CONCEPTO	COSTO
Componentes electrónicos	2090.60 \$
Tarjetas de circuito impreso	3680.00 \$
Gabinete	1000.00 \$
Total	6770.60 \$

CAPÍTULO 5

EJEMPLO DE APLICACIÓN

La mayor parte de este reporte se ha planteado en torno al hardware que complementa al PLM, siendo esto el objetivo central de este trabajo, para completar el trabajo es conveniente plantear un ejemplo de aplicación con el propósito de mostrar el potencial del prototipo en la automatización de procesos, dicho ejemplo tiene una finalidad puramente ilustrativa.

El capítulo inicia con la descripción de las características básicas del lenguaje SILL1, se incluye el formato para la instrucción en el lenguaje SILL1, los elementos y un ejemplo de instrucción, las características generales para la ejecución de un programa, el formato de programas fuente en SILL1, más adelante se describe como ejemplo de aplicación la realización de la lógica requerida por un arrancador para un motor trifásico efectuado con el PLM, el ejemplo está integrado por la descripción del proceso, la caracterización de los módulos lógicos empleados para determinar la secuencia de arranque del motor y por último se muestra el programa fuente en SILL1 para efectuar el arranque del motor.

5.1 CARACTERÍSTICAS BÁSICAS DEL LENGUAJE SILL1

Una determinada aplicación de automatización con el PLM requiere de la interconexión de varios módulos lógicos estando las entradas de algunos de ellos ligadas con los sensores binarios empleados, existiendo además módulos en donde las salidas estarán ligadas con los actuadores binarios utilizados.

Al desarrollar una aplicación con el PLM, cada módulo lógico debe ser declarado por medio de uno o varios renglones de texto que expresan en alguna forma el tipo de módulo y características del mismo, al conjunto de instrucciones mencionadas en el párrafo anterior se le denomina *programa fuente en lenguaje SILL1*, asociado con la aplicación que se desea realice el PLM.

5.1.1 FORMATO DE INSTRUCCIÓN EN SILL1

A excepción de los módulos tales como los secuenciadores de estado que requieren datos adicionales que no cabrían en un renglón, la forma sintáctica de las declaraciones asociadas cabe en un renglón, ésta se ilustra a continuación:

CODM#N E1,.. En,..S1,..Sm, D1,..Dq, CADBI;

Elementos de instrucción.

CODM#N E1,.. En,..S1,..Sm, D1,..Dq, CADBI;

CODM es una cadena de caracteres que simboliza la función efectuada por el módulo.

N es el número asociado con el módulo.

E1 a **En** son las designaciones asociadas con las n variables de entrada que el módulo requiera.

S1 a **Sm** son las designaciones asociadas con las m salidas que pudiera tener el módulo.

D1 a **Dq** son datos auxiliares que pudieran ser requeridos por algunos módulos, estos podrían ser entre otros: el tiempo asociado con la duración de un pulso generado por un temporizador o bien el número de estados que ha de presentar un secuenciador, etc.

Para los módulos que requieren de estos datos, q es un número que está comprendido entre cero y tres.

CADBI es una cadena formada por unos y ceros que especifica diversas características de funcionamiento como podrían ser:

Cuales entradas a una compuerta van a tener negación implícita, a que tipo de flanco responde una entrada de algún otro tipo de módulo, etc.

Ejemplo de instrucción en SILL1.

Compuerta AND de dos entradas, para la cual se desea que las entradas E0 y E1 y la salida S sean respectivamente las VB E01, I24, y S13; requiriéndose que la entrada E0 tenga preinversión y que el número de asignación sea 2, la declaración asociada es:

•AND2#2 E01, I24, S13, 10;

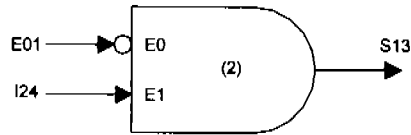


Figura 5.1 .- Compuerta AND de dos entradas.

5.1.2 CARACTERÍSTICAS GENERALES DE LA EJECUCIÓN DE UN PROGRAMA EN SIII1 EN LA CC DEL PLM.

Al correr un programa en SIII1 en la CC del PLM, el código asociado con cada ML es ejecutado cíclicamente siguiendo la siguiente secuencia:

- 1 Se copian en un buffer de entrada (BE) en RAM el estado que guardan los cuatro puertos asociados con los cuatro puertos asociados con las 32 entradas físicas VBE.
- 2 Se ejecuta uno a uno el código asociado con cada uno de los ML que el usuario haya declarado en el programa fuente correspondiente, tomándose del BE las entradas que cada módulo requiera, las salidas que se fueran generando son colocadas en un buffer de salida (BS) en RAM; si el ML emplea una o varias VBI como entradas los valores asociados con las mismas son tomados de un buffer intermediario (BI) en RAM, en caso de que haya en el ML una o varias salidas de tipo VBI los valores correspondientes son escritos en el BI.
- 3 Se copia el estado del BS en los puertos físicos asociados con las VBS.
- 4 Se regresa al paso uno.

De lo anterior se aprecia que el código asociado con cada módulo es ejecutado repetitivamente, variando el intervalo de repetición de acuerdo con el número de módulos que contenga el programa; esto es a mayor número de módulos crece el periodo de repetición.

Existen módulos que requieren que el periodo de repetición de la ejecución de su código asociado sea constante (10 ms), tal es el caso por ejemplo de los temporizadores, para hacer esto

posible el código asociado es colocado en una rutina de servicio de interrupción que es invocada con una periodicidad de 10 ms, empleándose para ello facilidades de temporización con que cuenta la CC del PLM.

En consecuencia, el código asociado con un programa en SIIL1 está dividido en dos partes, una de ellas es la que se ejecuta de acuerdo con los cuatro pasos descritos un párrafo anterior, a esta parte se le llama *subprograma principal (SPP)*, la otra parte está construida por el código cuya ejecución es temporizada y se denomina *subprograma temporizado (SPT)*. En la figura 5.2 se ilustra esta idea. Cabe señalar aquí que todo programa en SIIL1 debe tener un subprograma principal; sin embargo, puede haber programas que no contengan un subprograma temporizado.

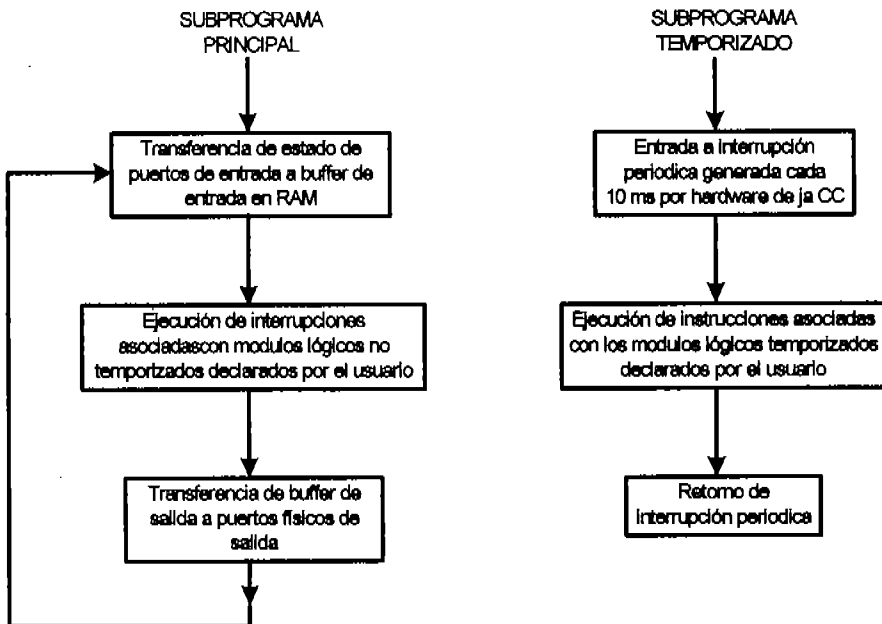


Figura 5.2.- Ejecución, en la CC del PLM, de los dos subprogramas que integran un programa en SIIL1

5.1.3 FORMATO DE PROGRAMAS FUENTE EN SIIL1

En general el formato de un programa fuente en SIIL1 deberá presentar la siguiente forma:

INPROG; delimitador de inicio del SPP.
 Sentencias asociadas con declaraciones
 de módulos que deben estar en el SPP.
 FINPP; delimitador de fin de SPP.
 INMODI; delimitador de inicio de SPT.
 Sentencias asociadas con declaraciones
 de módulos que deben estar en el SPT.
 FINMODI; delimitador de fin de SPT.

Ejemplo de un programa en SIIL1

```

INPROG;
AND2#1 E00, E01, I00, 11;
FINPP;
INMODI;
TEMPOC#1 I00, E03, S00, 00:02:00.00, 100;
FINMODI;

```

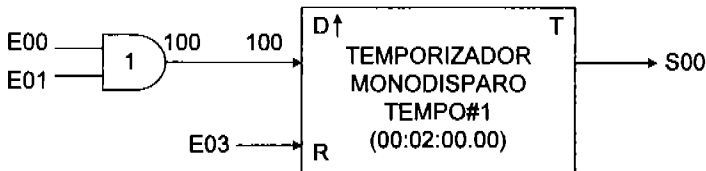


Figura 5.3 .- Representación de los módulos lógicos; compuerta AND y Temporizador que sirven como ejemplo para un programa SIIL1.

5.1.4 SOFTWARE GENERADOR DE CÓDIGO OBJETO (SGCO)

El SGCO procesa renglón por renglón el archivo de texto que contiene el programa fuente y genera uno o dos archivos de salida, el primero es un archivo de texto que contiene un reporte de los errores de sintaxis encontrados. En caso de no haber errores se genera un archivo binario, que contendrá el código objeto y las direcciones inicial y final de carga del mismo en el procesador del PLM.

Para cada tramo de código en un programa en SIIL1, correspondiente a una determinada acción efectuada por el mismo (módulo lógico, módulo auxiliar, código de inicialización, etc.), existe un código esqueleto normalizado (CEN) que es empleado por el SGCO, colocando para ello los bytes que correspondan en el CEN y localizando el mismo en la posición de memoria que corresponda.

Cada CEN se generó empleando un ensamblador de código fuente 68HC11, de modo que el código hexadecimal generado se origine en la dirección 0000, de esta manera cada CEN tiene asociada una tabla de asignación de bytes (TAB), donde se especifica que bytes han de ser modificados por el SGCO de modo que el módulo asociado tenga las características especificadas originalmente por el usuario en el programa fuente; de esta manera, en cada uno de los CEN que maneja el software de traducción, el valor de los bytes que ha de cambiar el mismo podrá parecer absurdo, ya que dichos bytes tendrán su valor correcto después de que el SGCO los haya reemplazado para obtener el tramo de código objeto requerido por el módulo que el usuario haya declarado en el programa fuente en SIIL1, que corresponda con la aplicación que este desarrollándose.

5.2 DESCRIPCIÓN DEL PROCESO A SER AUTOMATIZADO UTILIZANDO EL PLM.

Para aclarar ideas se expone aquí un ejemplo de aplicación, que consiste en realizar con el PLM la lógica requerida por un arrancador de voltaje reducido basado en auto transformador con transición de circuito abierto, para un motor de inducción trifásico; en la industria existen arrancadores de este tipo implantados con lógica alambrada. En la figura 5.4 se ilustra el esquema del conexionado al motor de los elementos actuadores de potencia (contactores), se supone que la secuencia de arranque requiere

que los contactores 1A y 2A se cierran al oprimirse el botón de arranque "A" y permanezcan así por tres segundos para abrirse una vez que ha transcurrido este tiempo, después de esto debe haber un tiempo de espera de medio segundo para cerrar el contactor M (marcha); como es usual en este tipo de sistemas se cuenta con un sensor de sobrecarga (OL) que abre sus contactos asociados al detectarse esta condición, además de que se debe contar con un botón de paro (P) de modo que al oprimirse el mismo, el motor sea desconectado del suministro trifásico que lo alimenta, se supone que el tap seleccionado para los auto transformadores es el adecuado para lograr el par de arranque requerido por la carga mecánica que mueve el motor.

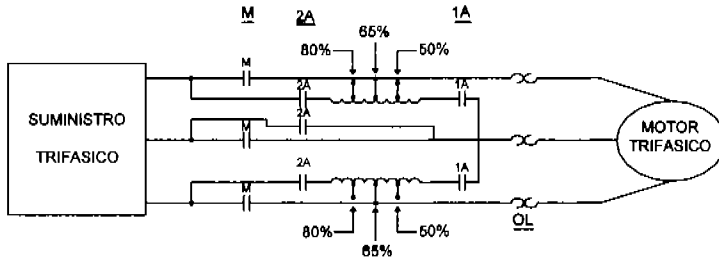


Figura 5.4 .- Conexión de arrancador de voltaje reducido basado en auto transformador cuya secuencia de arranque se ha de implantar con el PLM.

5.2 SECUENCIA DE ARRANQUE PARA UN MOTOR TRIFÁSICO UTILIZANDO EL PLM.

Una forma de realizar la secuencia de arranque con módulos lógicos realizables por el PLM se ilustra en la figura 5.5, apreciándose en la misma el empleo de cinco módulos los cuales son: una compuerta AND de dos entradas, un Flip-Flop RS con prioridad al RESET, un temporizador monodisparo (one-shot) con salida verificada en alto y duración de tres segundos con disparo por flanco de subida, un temporizador con retardo de activación (ON DELAY) con retardo de 3.5 segundos y un seguidor lógico; en la misma figura se muestran las variables booleanas empleadas y con que elementos físicos están relacionadas, en la figura 5.6 se ilustra el conexionado al PLM de los sensores y actuadores requeridos en esta aplicación.

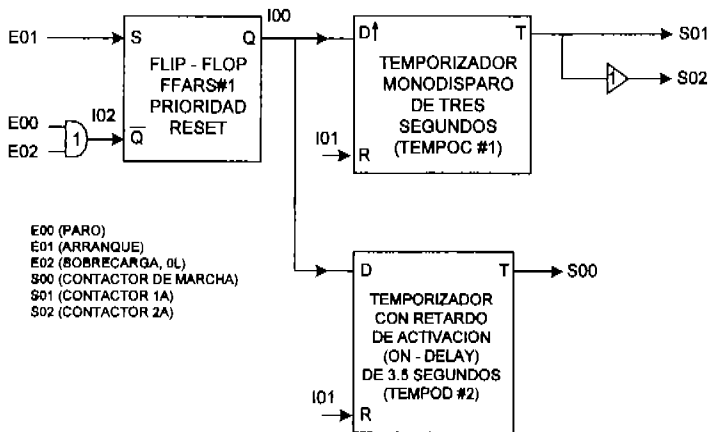


Figura 5.5.- Implantación de la secuencia de arranque requerida, empleando módulos realizables por el PLM.

Para un mismo tipo de módulo el lenguaje SHIL1 permite especificar entre otras cosas, los niveles de verificación de las entradas y salidas que el mismo pudiera contener; de esta manera, los niveles de verificación para las entradas "S" y "R" del Flip - Flop se especificaron como alto y bajo respectivamente, que es lo adecuado para este caso, el nivel de verificación de las entradas de RESET para los dos temporizadores se definió como alto, el nivel de verificación para las salidas de los mismos se definió como alto, nótese el empleo de la VBI I01 como señal de RESET para ambos temporizadores, cabe señalar aquí que siempre que se inicia la ejecución de un programa en el PLM, los buffers asociados con las variables booleanas son inicializados con ceros, de esta manera, una VBI que no sea empleada como salida por algún módulo tendrá siempre un nivel de cero lógico.

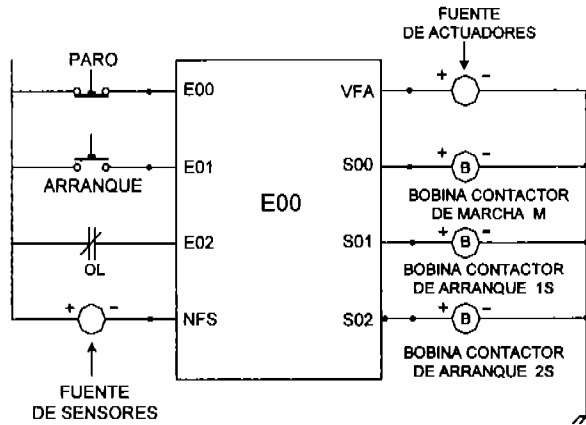


Figura 5.6 .- Conexionado al PLM de sensores y actuadores asociados con el arrancador aquí Mostrado.

Programa fuente en SIIL para implementar el arranque del motor con el PLM.

INPROG; Inicio de subprograma principal.

* Declaración de la compuerta AND.

AND2#1 E00, E02, I02, 11;

* Declaración del seguidor empleado.

SEG#1 S01, S02;

* Declaración del Flip-Flop empleado.

FFARS#1 E01, I02, I00, 1000;

FINPP; Declaración de fin de subprograma principal.

* Inicio de subprograma temporizado.

INMODI;

* Declaración de temporizador monodisparo.

TEMPOC#1 I00, I01, S01, 00:00:03.00, 101;

* Declaración del temporizador con retardo a la activación.

TEMPOD#2 I00, I01, S00, 00:00:03.50, 10;

FINMODI; Declaración de fin de subprograma temporizado.

CONCLUSIONES

La finalidad de este trabajo fue la de obtener como resultado la creación de un prototipo robusto para el dispositivo denominado Programador Lógico Modular (PLM), partiendo del diseño previo, realizado por el M. I. Antonio Salvá Calleja que es el director de este trabajo; este objetivo se ha cumplido al presentar como producto terminado el PLM. Partiendo de este modelo se tiene la pauta para la manufactura del dispositivo con mucha mayor facilidad y menor costo de lo que represento la construcción del primer prototipo.

A lo largo del desarrollo del trabajo se presentaron varias interrogantes como son: ¿que software se utilizaría para elaborar los circuitos impresos? , ¿cuántas tarjetas serían necesarias para contener el PLM? , ¿cómo se organizarían los circuitos en las tarjetas? , ¿cuál sería la mejor manera de presentar al usuario final el prototipo? , ¿qué material sería el adecuado para elaborar la construcción del gabinete? , además de muchos otros detalles, todas las interrogantes fueron resueltas de la mejor manera posible con los recursos que estuvieron a disposición.

El desarrollo del prototipo se realizó en torno a la tarjeta FACIL_11 que funciona como CC del PLM y que emplea como microcontrolador al 68HC11, esto se hizo con la finalidad de aprovechar la experiencia y infraestructura creada previamente por el M. I. Antonio Salvá Calleja, de no hacerlo así habría implicado mucho más trabajo; desde elegir otro microcontrolador y adicionar en su entorno todo lo necesario para hacer el enlace con el hardware restante del PLM y realizar todas las pruebas que se consideraran apropiadas.

La presentación del prototipo del PLM podría hacerse más reducida; esto sería posible si se sustituyera la tarjeta que contiene a la fuente de alimentación por un eliminador de baterías como fuente externa, dicho eliminador tendría que proporcionar los voltajes de $12V_{cd}$ 1 A y $5V_{cd}$ 0.5 A, esto reduciría las dimensiones y el peso del PLM; la altura del modelo que es de 22 Cm se reduciría en 8 Cm y el peso del dispositivo se reduciría aproximadamente en una tercera parte ya que los dos elementos más pesados en el prototipo son la fuente de alimentación y el gabinete (debido al material en el que fue elaborado), el lograr que se reduzcan las dimensiones del PLM haría más práctico su transporte.

El desarrollar este trabajo me deja una buena experiencia en cuanto al proceso que se sigue para implementar un diseño partiendo primero de la idea de que hacer para después pasar a la búsqueda de cómo hacerlo, iniciando con la prueba de los circuitos en tablillas experimentales hasta llegar a presentarlo como un producto terminado.

Hardware y software son complementarios en este trabajo a mí me ha tocado únicamente la parte concerniente a la implementación hardware con lo que cumplido de la mejor manera que he podido brindar. El prototipo del PLM es resultado del esfuerzo de dos personas básicamente el director de esta tesis y quien presenta a consideración del jurado este trabajo.

Al ya contar con el modelo terminado yo consideraría que el paso siguiente es implementar su aplicación en el campo didáctico; esto podría hacerse diseñando algunas prácticas para mostrar a los alumnos la utilidad de los PLC's con un dispositivo diseñado en su facultad.

BIBLIOGRAFÍA

Victoriano Angel Martínez Sánchez
Automatización Industrial Moderna
Alfaomega Ra – Ma
2001

Robert Boylestad
Louis Nashelsky
Electrónica Teoría de Circuitos
Prentice Hall Hispanoamericana, S. A.
Segunda edición en español 1989

Adel S. Sedra
Kenneth C. Smith
Circuitos Microelectrónicas
Cuarta edición 1999

Antonio Salvá Calleja
Programador Lógico Modular
Noviembre de 1998

Joseph Balcells
José Luis Romeral
Autómatas Programables
Alfaomega marcombo
1998

R. Ackermann, J. Franz, T. Hartmann
A. Hopf, M. Kantel, B. Plagemann
Controles lógicos programables
1988

Timothy J. Maloney
Electrónica industrial moderna
Prentice Hall
Tercera edición

APÉNDICE A

FUENTE DE PODER

En esta sección del trabajo se describe paso a paso el diseño de la fuente que alimentara a todos los circuitos incluidos en el PLM, comenzando con el planteamiento de condiciones generales sobre fuentes de poder, siguiendo con la descripción de cada una de las etapas en que se divide el diseño, continuando con el dibujo esquemático del dispositivo y finalizando con la implementación de la fuente de poder en una placa de circuito impreso.

A.1 DISEÑO DE LA FUENTE DE PODER.

En la figura A.1 se muestra el esquema a bloques que constituye las partes de una fuente de poder y los tipos de voltajes en los diversos puntos de la fuente. El voltaje de ca, por lo general de 127 volts rms, se conecta a un transformador el cual reduce el nivel de voltaje para obtener la salida de cd deseada. Un rectificador con diodos proporciona más tarde un voltaje rectificado de media onda o, más generalmente, de onda completa que se aplica a un filtro para suavizar la variación de la señal. A menudo un simple filtro de capacitor es suficiente para proporcionar esta acción de suavizamiento. El voltaje resultante de cd con algún rizo o variación de voltaje de ca se proporciona después como entrada a un regulador de circuito integrado (CI) que produce como salida un nivel de voltaje de cd bien definido, con un voltaje de rizo extremadamente bajo sobre un intervalo de carga.

Para realizar el diseño de la fuente que alimentara al PLM es necesario definir cada una de las etapas que se muestran en la figura A.1

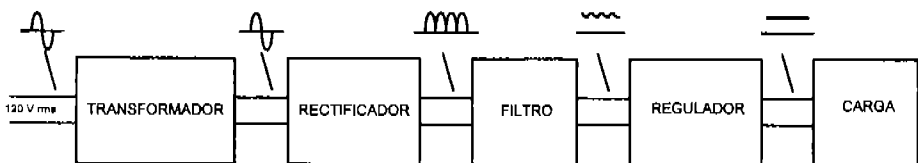


Figura A.1 .- Etapas de la fuente de alimentación.

A.1.1 La carga.

Puede parecer un poco extraño iniciar por la última etapa, en el diseño de la fuente de poder, pero la carga nos marca las necesidades que debe cubrir el dispositivo, para cumplir con los requerimientos de alimentación.

La carga para la fuente; en este caso, es la totalidad de los circuitos que componen el PLM, la carga esta dividida en dos partes; por un lado tenemos la sección que corresponde a las salidas del PLM, que están constituidas por circuitos que emplean relevadores de baja potencia, que deben ser alimentados con +12 volts de cd y que tienen un consumo de corriente total de 250 mA, aun siendo el consumo de corriente de $\frac{1}{4}$ de Ampere, la fuente se diseño con una capacidad de carga de 1A.

Por otra parte tenemos los circuitos de entrada, los circuitos auxiliares y la CC del PLM que deben ser alimentados por una fuente de +5 volts de cd y que tienen un consumo de corriente extremo de 100 mA, la fuente se diseño con un voltaje en la salida de +5V_{cd} y una corriente de 500 mA.

Las especificaciones de corriente mencionadas son para los casos extremos en que las 32 entradas y las 16 salidas son uno lógico, se consideraron factores de seguridad para cada caso.

A.1.2 Regulación de voltaje.

Existe una amplia variedad de configuraciones de circuitos que son capaces de efectuar la regulación de voltaje o corriente, pero la forma que nos resulta más adecuada y sencilla para realizar la etapa de regulación de voltaje para el diseño de la fuente de poder que en este capítulo nos ocupa es la de utilizar unidades que contienen los circuitos para la fuente de referencia, el amplificador de error, el dispositivo de control y la protección a sobrecarga todo esto en un sólo chip de circuito integrado que cuenta con tres terminales.

Para hacer una buena elección de los dos reguladores que se usan dentro del diseño de la fuente de poder, es necesario determinar lo siguiente:

1. Voltaje y corriente a los cuales se necesita regular.
2. Valor del voltaje no regulado mínimo que se requiere a la entrada del CI para que éste opere adecuadamente.
3. Cuales son los CI reguladores que se utilizarán en el diseño.

Para cubrir el primer punto recordaremos que en el capítulo uno, en el párrafo que se refiere a la fuente de alimentación se estableció que son necesarias dos fuentes, una que operara $+12V_{cd}$ a corriente de un máximo de 1 A y otra fuente de $+5V_{cd}$ a una corriente máxima de 500 mA.

Para definir el segundo y tercer punto existe un grupo de reguladores de voltaje positivo fijo, es la serie 78XX, los dos dígitos después del prefijo 78 indican el voltaje de salida del regulador, estos dispositivos brindan voltajes fijos desde $+5V_{cd}$ hasta $+24V_{cd}$.

Para una unidad de CI particular, las especificaciones del dispositivo señalan un intervalo de voltaje dentro del cual el voltaje de entrada puede variar para mantener el voltaje de salida regulado, dentro de un intervalo de corriente de carga. Debe mantenerse un voltaje diferencial de salida-entrada para que el CI opere, lo que significa que la variación de voltaje de entrada debe siempre mantenerse lo suficientemente grande para conservar una caída de voltaje a través del CI que permita la operación adecuada del circuito interno. Las especificaciones del dispositivo también incluyen la cantidad del cambio de voltaje de salida, que resulta de los cambios en la corriente de carga (regulación de carga) y también de los cambios en el voltaje de entrada (regulación de línea). De la serie de reguladores 78XX se seleccionaron los dos que se presentan en la tabla A.1.

Tabla A.1.- Muestra los reguladores que se seleccionaron para el diseño de la fuente de poder, los dos reguladores se eligieron para trabajar con una corriente de 1A, ya que para el CI 7805 no hay presentación del chip para la corriente de 500 mA.

Número de parte del CI	Voltaje positivo Regulado (V)	Voltaje de entrada mínimo (V)	Corriente de Operación (A)	Encapsulado
7805	+ 5	7.3	1	TO-220
7812	+ 12	14.6	1	TO-220

En la figura A.2 muestra cómo se conectan los reguladores del tipo 78XX. Un voltaje v_{cd} no regulado, rectificado y filtrado es la entrada (V_{ent}) para la terminal 1 del CI regulador. Los capacitores conectados de la entrada y la salida a tierra ayudaran a mantener el voltaje v_{cd} y a filtrar adicionalmente cualquier variación de voltaje de alta frecuencia. El voltaje de salida desde la terminal 2 está entonces disponible para conectarlo a la carga. La terminal 3 es la referencia o tierra del circuito integrado.

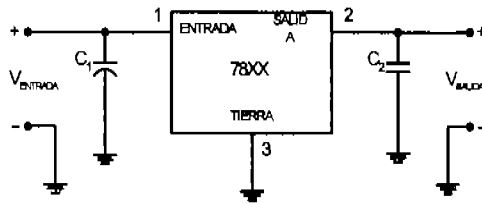


Figura A.2.- Regulador de voltaje 78XX empleado en la fuente de alimentación del PLM.

A.1.3 Etapa de filtrado.

En el párrafo correspondiente a las consideraciones generales sobre fuentes de poder se establece la necesidad de una etapa de filtrado, en donde la finalidad del filtro es suavizar las variaciones de voltaje de la señal de entrada al circuito regulador. El filtro para este caso estará constituido únicamente por un capacitor, el cual será un dispositivo capaz de proporcionar el ya nombrado suavizamiento de la señal, es importante una buena elección del valor de capacitancia a fin de obtener un voltaje de cd suficiente del capacitor, así como un valor de voltaje de rizo que garanticen una buena operación del CI regulador de voltaje.

El voltaje que se obtiene de la etapa de rectificación no es un voltaje filtrado, contiene una componente de cd y una componente de rizo, la componente de rizo es la parte de la señal que el filtro nos ayudara a suavizar. El capacitor está conectado a través de la salida del rectificador y se dispone de voltaje de cd de salida a través del capacitor. La figura A.3a muestra el voltaje de salida rectificado de onda completa antes de filtrar la señal. La figura A.3b muestra la forma de onda resultante después de que el capacitor se conecta a través de la salida del rectificador. Como se muestra, este voltaje filtrado tiene un nivel de cd con cierto factor de rizo superpuesto en él.

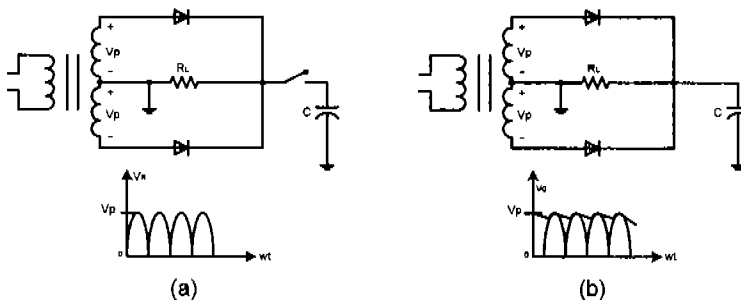
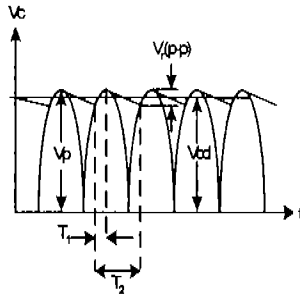


Figura A.3.- Voltaje rectificado de onda completa (a), Voltaje de salida filtrado (b).

En la señal rectificada de onda completa que se indica en la figura A.4 se señalan dos intervalos de tiempo. T_1 es el tiempo durante el cual un diodo del rectificador de onda completa conduce y carga el capacitor hasta el voltaje de salida pico del rectificador (V_p). T_2 es el tiempo en el que el voltaje del rectificador desciende por debajo del voltaje pico y el capacitor se descarga a través de la carga ($V_{ent\ min}$).



T_1 : Es el tiempo durante el cual un diodo rectificador de onda completa conduce y carga el capacitor hasta el voltaje de salida pico del rectificador (V_p).
 T_2 : Es el tiempo en el que el voltaje del rectificador desciende por debajo del voltaje pico, y el capacitor se descarga a través de la carga.
 $V_{rd} = V_p - V(p-p)/2$
 $V(p-p)$: Voltaje de rizo

Figura A.4.- Voltaje de salida aproximado de un circuito filtro con capacitor.

Si el capacitor se descarga sólo ligeramente (debido a una carga pequeña), el voltaje promedio estaría muy cerca del valor óptimo de V_p . La cantidad de voltaje de rizo sería también pequeña para una carga de bajo valor. Para cargas más grandes, el voltaje de cd será más pequeño con rizados más grandes.

Después de lo expresado en los párrafos anteriores sólo nos restaría decidir el valor del capacitor correspondiente para las dos secciones de la fuente de poder, así como el cálculo del voltaje de rizo y el voltaje que se proporcionara a la entrada de la etapa de regulación en cada caso.

1.- Fuente de $+5V_{cd}$ a 500 mA.

Corriente de carga: $I_{cd} \approx 100$ mA

Valor del capacitor propuesto: $C = 2200$ μ F

Voltaje proporcionado por la bobina secundaria: $V_{rms} = 7$ V

$$V_p = \sqrt{2}(V_{rms})$$

$$V_p = \sqrt{2}(7)$$

$$V_p = 9.8994$$

Voltaje de rizo (rms)

$$V_r(rms) = (2.4 I_{cd})/C ; \text{ Donde } I_{cd} \text{ esta en mA y } C \text{ en } \mu\text{F}$$

$$V_r(rms) = (2.4(100))/2200$$

$$V_r(\text{rms}) = 0.1091$$

$$\text{Voltaje de rizo (pico): } V_r(\text{pico}) = \sqrt{3} V_r(\text{rms})$$

$$V_r(\text{pico}) = (1.7305)(0.1091)$$

$$V_r(\text{pico}) = 0.1889$$

Nivel de cd de voltaje a través del capacitor $C = 2200 \mu\text{F}$ es:

$$V_{cd} = V_p - V_r(\text{pico})$$

$$V_{cd} = 9.89 - 0.1889$$

$V_{cd} = 9.7011$ (Cuando se emplea para cargas bajas, el valor de V_{cd} es sólo un poco menor que V_p .)

Voltaje de entrada mínimo al regulador 7805:

$$V_{ent \text{ min}} = V_p - 2V_r(\text{pico})$$

$$V_{ent \text{ min}} = 9.89 - 2(0.1889)$$

$$V_{ent \text{ min}} = 9.5122$$

Las especificaciones del dispositivo 7805 señalan el valor V_{ent} requerido para mantener la regulación en forma adecuada, este valor es de 7.3 V para la terminal 1 del chip. El voltaje que se mantiene en el capacitor para la carga de 100mA es de 9.5122 V, que es mayor a 7.3 V, por lo tanto el 7805 mantendrá perfectamente la regulación a $+5 V_{cd}$.

2.- Fuente de $+12V_{cd}$ a 1A.

Corriente de carga: $I_{cd} = 250 \text{ mA}$

Valor del capacitor propuesto: $C = 2200 \mu\text{F}$

Voltaje proporcionado por la bobina terciaria: $V_{rms} = 14 \text{ V}$

$$V_p = \sqrt{2}(V_{rms})$$

$$V_p = \sqrt{2}(14)$$

$$V_p = 19.7989$$

Voltaje de rizo (rms)

$V_r(\text{rms}) = (2.4 I_{cd})/C$; Donde I_{cd} esta en mA y C en μF

$$V_r(\text{rms}) = (2.4(250))/2200$$

$$V_r(\text{rms}) = 0.2727$$

Voltaje de rizo (pico): $V_r(\text{pico}) = \sqrt{3} V_r(\text{rms})$

$$V_r(\text{pico}) = (1.7305)(0.2727)$$

$$V_r(\text{pico}) = 0.4723$$

Nivel de cd de voltaje a través del capacitor $C = 2200 \mu\text{F}$ es:

$$V_{cd} = V_p - V_r \text{ (pico)}$$

$$V_{cd} = 19.7989 - 0.4723$$

$V_{cd} = 19.3266$ (Cuando se emplea para cargas bajas, el valor de V_{cd} es sólo un poco Menor que V_p .)

Voltaje de entrada mínimo al regulador 7812:

$$V_{ent \text{ min}} = V_p - 2V_r \text{ (pico)}$$

$$V_{ent \text{ min}} = 19.7989 - 2(0.4723)$$

$$V_{ent \text{ min}} = 18.8543$$

Las especificaciones del dispositivo 7812 señalan el valor V_{ent} requerido para mantener la regulación del chip en forma confiable este valor es de $14.6 V_{cd}$. El voltaje más bajo que se mantiene en el capacitor es de $18.8543 V_{cd}$, que es mayor a $14.6 V_{cd}$, por lo tanto el 7812 mantendrá perfectamente la regulación a $+12 V_{cd}$.

A la salida de los correspondientes reguladores de voltaje, de las secciones de la fuente de poder ($+5 V_{cd}$ y $+12 V_{cd}$), se hace la conexión entre la terminal 2 y tierra de un capacitor de $100 \mu\text{F}$ para filtrar el ruido de alta frecuencia.

A.1.4 Etapa de rectificación.

En la etapa de rectificación de la señal tenemos dos opciones; hacer una rectificación de media onda o efectuar una rectificación de onda completa.

La señal rectificada de onda completa tiene menos porcentaje de rizo y es en consecuencia una señal mejor rectificada que la señal de media onda, la cantidad del factor de rizo de la señal rectificada de onda completa es aproximadamente 2.5 veces más pequeña que la de la señal rectificada de media onda y proporciona una señal mejor filtrada.

Por los motivos expuestos anteriormente, se decide tomar la opción de rectificación de onda completa y el dispositivo que se emplea para tal efecto es el puente rectificador DB105 que se presenta en encapsulado de tipo Dip.

A.1.5 El transformador.

1.- Conceptos generales sobre transformadores

El transformador es un dispositivo que convierte energía eléctrica alterna de un cierto nivel de voltaje, en energía alterna de otro nivel de voltaje, por medio de la acción de un campo magnético. Está constituido por dos o más bobinas de alambre, aisladas entre sí eléctricamente por lo general y arrolladas alrededor de un mismo núcleo de material ferromagnético. La única conexión entre las bobinas la constituye el flujo magnético común que se establece en el núcleo.

Uno de los devanados del transformador se conecta a una fuente de potencia de corriente alterna, y el segundo (y a veces el tercero) devanado entrega la potencia eléctrica a las cargas. Al arrollamiento que se conecta a la fuente suele llamársele devanado primario o devanado de entrada, y al arrollamiento que se conecta a la carga se le designa devanado secundario o devanado de salida. Si llegará a existir un tercer arrollamiento en el transformador, a este se le denominaría devanado terciario.

Las relaciones entre los voltajes de entrada y salida, y entre las corrientes de entrada y de salida, están dadas por dos ecuaciones sencillas. Los transformadores ideales tienen N_p espiras de alambre en su primario, y N_s espiras de alambre en su lado secundario. La relación entre el voltaje $V_p(t)$ aplicado al primario del transformador, y el voltaje $V_s(t)$ inducido en el secundario es:

$$V_p(t)/V_s(t) = N_p/N_s = a$$

$$a = N_p/N_s$$

Donde a se define como relación de espiras del transformador.

La relación entre las corrientes $i_p(t)$ del primario e $i_s(t)$ del secundario del transformador es.

$$N_p i_p(t) = N_s i_s(t)$$

$$I_p(t)/i_s(t) = 1/a$$

2.- Selección del transformador

Como se ha descrito en repetidas ocasiones, son necesarios dos niveles diferentes de voltaje positivo y de corrientes también distintas; por este motivo se requiere un transformador con tres devanados; una bobina primaria conectada a la línea de 127 V_{rms} , una bobina secundaria que proporcione un valor de V_{rms} que resulte adecuado para alimentar el regulador 7812 a una corriente de 1A y por último un devanado terciario que proporcione una corriente de 500 mA con un valor de V_{rms} suficiente para obtener una buena operación del CI 7805 que se utiliza para la sección de la fuente de poder de +5 V_{cd} .

Para elegir el transformador adecuado para cada caso, se considero lo siguiente:

- 1 En la fuente de +5 V_{cd} , la etapa de regulación de voltaje trabajara con el circuito 7805 que necesita un valor mínimo de 7.3 V_{cd} en la entrada

El voltaje que alimenta la entrada del regulador es suministrado por un capacitor que funciona como filtro, la magnitud de este voltaje se puede ser calculado con la ecuación siguiente:

$$V_{ent\ min} = V_p - 2V_r \text{ (pico)}$$

Si consideramos un voltaje de rizo pequeño, tendremos que $V_{ent\ min} \cong V_p$, entonces podemos considerar lo siguiente:

$$V_p = 7.3$$

$$V_p = \sqrt{2}V_{rms}$$

$$V_{rms} = V_p/\sqrt{2}$$

$$V_{rms} = 5.1618$$

El transformador en el devanado secundario debe proporcionar 5.1618 V_{rms} y 500 mA, estos serian los parámetros para obtener el mínimo de voltaje en la entrada del regulador, sin embargo para la fabricación del transformador se solicito que la bobina secundaria proporcionara un voltaje

de $7 V_{rms}$ y 500 mA, este cambio no representa ningún problema para el buen desempeño del regulador 7805.

- 2 En la fuente de $+12 V_{cd}$ la etapa de regulación trabaja con el chip 7812 que necesita en la entrada un valor mínimo de voltaje de $14.6 V_{cd}$ para trabajar correctamente.

De la misma manera que se expone en el inciso anterior, se calcula el voltaje rms mínimo necesario en el devanado terciario.

$$V_p = 14.6$$

$$V_p = \sqrt{2} V_{rms}$$

$$V_{rms} = V_p / \sqrt{2}$$

$$V_{rms} = 10.3237$$

El transformador en el devanado terciario debe proporcionar $10.3237 V_{rms}$ y 1 A, estos son los parámetros para obtener el mínimo de voltaje en la entrada del regulador, sin embargo para la fabricación del transformador se pidió que la bobina terciaria proporcionara un voltaje de $14 V_{rms}$ y 1 A, que como ya se expuso anteriormente no ocasionara ningún inconveniente.

En resumen el transformador utilizado en la fuente de poder del PLM en su bobina de entrada trabaja con el voltaje de línea de $127 V_{rms}$. Del devanado secundario se obtienen $7 V_{rms}$ a 500 mA y por último en el devanado terciario tenemos $14 V_{rms}$ y una corriente de 1A.

A.2 ESQUEMA DE LA FUENTE DE ALIMENTACIÓN.

En esta sección se presenta el dibujo esquemático de la fuente de poder del PLM. En la figura A.5, se muestra la distribución de los componentes del dispositivo, se agregan a los elementos ya mencionados en los párrafos superiores tres fusibles dos de un ampere y uno de quinientos mili amperes; el primer fusible se localiza en la entrada del devanado primario del transformador, los otros dos fusibles se encuentran después de la etapa de filtrado de los devanados secundario y terciario del transformador.

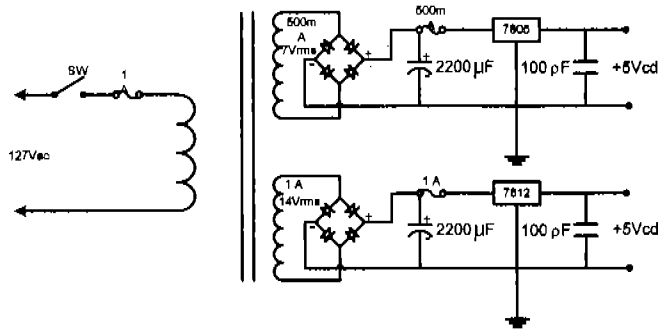


Figura A.5 .- Esquema de la fuente de alimentación para el PLM.

El siguiente paso en el diseño de la fuente es pasar del dibujo esquemático, al diseño del circuito impreso plasmado en un dibujo PCB generado como las placas anteriores (Tarjeta Periférica 1 y Tarjeta Periférica 2) en el software Tango.

A.3 DISEÑO DEL CIRCUITO IMPRESO PARA LA FUENTE DE ALIMENTACIÓN.

La placa empleada para contener la fuente de poder conserva las medidas de las tarjetas Periférica 1 y Periférica 2 (7.1 pulgadas por 4.5 pulgadas), dimensiones que son prácticas para efectuar el diseño del gabinete que contendrá al PLM. En la figura A.6 se muestra el dibujo PCB de la fuente de alimentación.

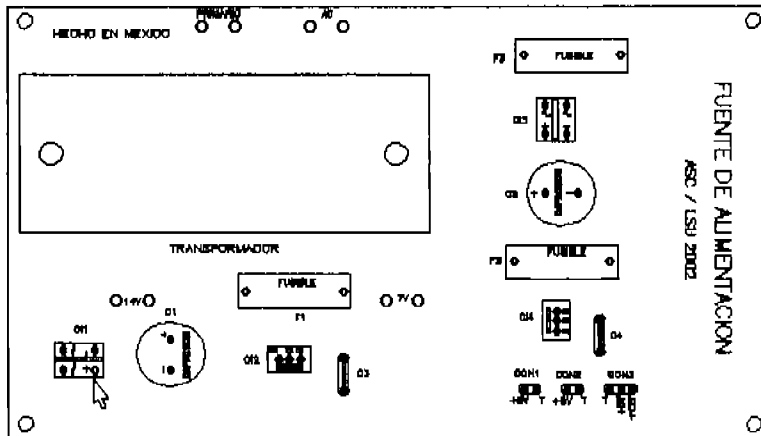


Figura A.6 .- Dibujo PCB para la fuente de alimentación para el PLM.

Contando con una amplia superficie solo resta decidir la distribución de los componentes la cual es la siguiente:

En la zona izquierda de la tarjeta tenemos; en la parte superior el transformador de 127 V_{ca} a 7V_{ca} y 14V_{ca}, bajo el transformador se localizan todos los componentes que constituyen la sección de la fuente de +12V_{cd}, de izquierda a derecha tenemos, el puente rectificador DB105, el filtro (un capacitor de 2200 μ F a 25V), un fusible de 1A, el regulador 7812 y un capacitor de 100 ρ F.

En la zona derecha de la placa, en la parte superior se localiza un fusible de 1A que esta dispuesto para la bobina primaria del transformador, bajo este dispositivo se encuentran los componentes que forman la sección de la fuente de +5V_{cd}, de arriba hacia abajo tenemos; el puente rectificador DB105, el filtro (un capacitor de 2200 μ F a 25V), un fusible de 500mA, un regulador 7805 y un capacitor de 100 ρ F y por último en la esquina inferior derecha tenemos tres conectores molex que proporcionan las salidas de la fuente hacia las tarjetas FACIL_1, Periférica 1 y Periférica 2. En la figura A.7 se puede observar la tarjeta correspondiente a la fuente de alimentación.

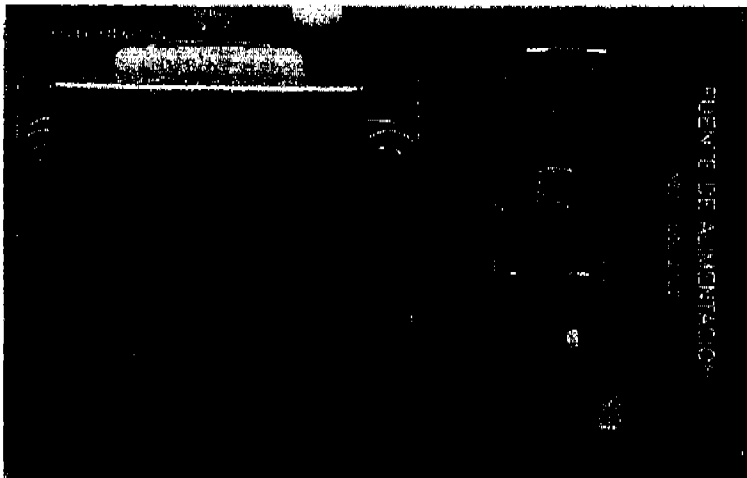


Figura A.7.- Tarjeta de la fuente de alimentación para el PLM terminada.

Es necesario mencionar que adicionalmente a las Tarjeta Periférica 1, Tarjeta Periférica 2 y Fuente de Alimentación, son necesarias tres pequeñas placas más; dos de ellas albergan cada una 16 bornes para las correspondientes 16 señales de los sensores de entrada al PLM y la tercera placa contiene los 16 bornes para los actuadores del PLM.

APÉNDICE B

GUÍA PARA EL USO DEL SOFTWARE TANGO/PCB DE ACCEL VERSIÓN 1.6

Tango/PCB es un programa popular en el diseño de circuitos. Es un flexible y completo paquete de diseño de circuitos, fácil de usar que se apoya en un menú que maneja comandos para la ayuda de creación, edición, impresión, respaldo y trazado de diagramas de conexión de circuitos, desarrollado para correr en PC IBM y compatibles. Éste paquete elimina la necesidad de un hardware especial o equipo fuera de lo Standard.

El menú principal de Tango/PCB cuenta con los siguientes comandos:

- Current
- Delete
- Edit
- File
- Jump
- Library
- Move
- Nets
- Output
- Place
- Setup
- Undo
- Zoom

Todos los comandos cuentan con un submenú de comandos. En las siguientes líneas se describe en forma general la función de los comandos más utilizados.

Current: Mediante éste comando se fijan los parámetros que tomaran los pads (Agujeros que se utilizan para insertar las patas de los componentes en la placa del circuito), vias (Agujeros que se utilizan para comunicar las caras de la tarjeta), las pistas (líneas de conexión entre componentes).

El comando `current` también sirve para determinar en que cara de la placa se trabaja (Top Layer o Bottom Layer).

El submenú del comando `current` es el siguiente:

Layer
Pad
String
Track
Via

Delete: Se utiliza para borrar componentes como pistas (Tracks), pads, vias, bloques de componentes etc. El submenú del comando `delete` es el siguiente:

Arc
Block
Component
Fill
Highlight
Pad
String
Track
Via

Edit: Se utiliza para modificar en forma particular los parámetros establecidos en `current` para una determinada pista, via, pad etc. El submenú para `edit` es el que sigue:

Arc
Component
Pad
String
Track
Via

File: Sirve para el manejo de archivos; carga, borrado, almacenamiento etc. Los subcomandos de file son los siguientes:

Clear
Dos
Load
Quit
Save

Library: Permite crear, borrar, renombrar y modificar librerías, los subcomandos de library son:

Add
Browse
Delete
Merge
Rename

Move: Permite mover de posición pads, vias, tracks, componentes, etc el submenú correspondiente es:

Arc
Block
Component
End point
Fill
Pad
Reroute
String
Track
Via

Nets: Entre otras funciones en éste comando encontramos el subcomando highlight que sirve para checar todas las conexiones que tiene un pad o una via con las demás vias o pads, este subcomando es de gran utilidad para evitar fallas en el conexionado de componentes. El submenú de éste comando es el siguiente:

- Clear
- Display
- Generate
- Highlight
- Identify
- Load
- Optimize
- Route
- Verify

Output: Permite elaborar reportes sobre diversos aspectos de la placa de PCB, por ejemplo, la impresión del archivo tomando solo la cara L Top o solo la cara L Bottom y otros tipos de reportes más. El submenú correspondiente es el siguiente:

- Apertures
- Cam
- Plot/Print
- Reports

Place: Se utiliza para plasmar en el dibujo, los componentes, pads, strings, etc. El submenú con que cuenta el comando place es el siguiente:

- Arc
- Block Copy
- Component
- Designator

Fill
Pad
String
Track
Via

Setup: Configura el puerto (os) seriales, asigna los colores, a las capas de la placa, borde, pads, pistas etc. Sirve también para establecer el avance del cursor sobre el grid etc. El submenú de comandos es el que sigue:

Communications
DRC
Display
Grids
Options
Palette

Undo: Se utiliza para deshacer la última operación ejecutada. No cuenta con submenú.

Zoom: Se utiliza para dibujar y obtener varias formas de visualizar el dibujo del circuito impreso. El submenú de comandos es:

All
Board
Center
In
Out
Redraw
Window

Cuando se quiere seleccionar de algún comando, se puede realizar de dos modos distintos; uno es moviendo el cursor hasta la posición del comando elegido y pulsando enseguida el botón izquierdo del ratón, a continuación se desplegará el submenú del comando y repitiendo la operación ya descrita se puede seleccionar alguno de los subcomandos, otra forma de seleccionar es pulsando la letra que se encuentra subrayada en el comando, nuevamente se desplegará el submenú del comando y repitiendo el procedimiento de pulsar la letra subrayada se elige el subcomando deseado.