

81,301  
20 JUN 1997  
BIBLIOTECA



81301



Este libro debe ser devuelto, a más tardar en la última fecha sellada. Su retención más allá de la fecha de vencimiento, lo hace acreedor a las multas que fija el reglamento.

FECHA DEVOLUCION	FECHA DE ENTREGA
04 OCT 1997	
27 01	23 OCT 1997
03 01	03 DIC 1997

10

253-16

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY  
CAMPUS ESTADO DE MÉXICO**



**DISEÑO E IMPLANTACIÓN DE UNA INTERFAZ DE  
COMUNICACIÓN ENTRE ROBOTS INDUSTRIALES Y  
CONMUTADORES A BASE DE TRANSPUTERS.**

**TESIS QUE PARA OPTAR EL GRADO DE  
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN  
PRESENTA**

**JAIME ALARCÓN CELIS**

Asesor: Dr. JESÚS SÁNCHEZ VELÁZQUEZ

Comité de tesis: Dr. LUIS TREJO RODRÍGUEZ  
Dr. ISAAC RUDOMÍN GOLDBERG  
Dr. ROBERTO VALDIVIA BEUTELSPACHER

Jurado: Dr. LUIS TREJO RODRÍGUEZ,  
Dr. ISAAC RUDOMÍN GOLDBERG,  
Dr. JESÚS SÁNCHEZ VELÁZQUEZ,  
Dr. ROBERTO VALDIVIA BEUTELSPACHER,

Presidente  
Secretario  
Vocal  
Vocal

**Atizapán de Zaragoza, Edo. Méx., Mayo de 1997.**

11.0000-00EM

15 DIC 1997

784  
TA  
7000  
11.00

06 JUN 2000 11.0000-00EM

81,301

Quiero hacer patente mi agradecimiento al

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES  
DE MONTERREY CAMPUS TOLUCA**

En especial a:

*Ing. Roberto Rueda Ochoa*

*Ing. Hilda Catalina Cruz Solís*

*Ing. Juan Carlos Arreola Rivas*

*Ing. Manuel R. Rodríguez Abrego*

por el apoyo que me brindaron para la realización de este  
proyecto.

*Dr. Jesús Antonio Sánchez Velázquez.*

Que gracias a su apoyo y asesoría fue posible la elaboración  
de esta tesis.

*A mi esposa.*

*Kituty*

*Con todo mi amor, dedico este trabajo a mi  
adorada esposa que gracias a su apoyo y  
comprensión resultaron determinantes para la  
culminación de la Tesis.*

*Con amor Jaime.*

*A mis Hijas.*

*Nayeli e Ischel*

*Que gracias a su amor y cariño me dieron  
animos para seguir adelante y terminar mis  
estudios.*

*Las quiere su Papá.*

---

**INDICE**

LISTA DE FIGURAS, TABLAS Y FOTOGRAFÍAS.	8
Capítulo 1.	
INTRODUCCIÓN.	11
1.1.- PLANTEAMIENTO DEL PROBLEMA	14
1.2.- RESUMEN	16
Capítulo 2.	
CONTROLADORES PARA CFM	18
2.1.- ESTRATEGIAS PARA COMPARAR ALTERNATIVAS DE COMUNICACIÓN	20
2.2.- MAQUINAS Y ROBOTS DE LA CELDA DE MANUFACTURA	27
Capítulo 3.	
ARQUITECTURA DEL CONTROLADOR PROPUESTO.	31
3.1.- ARQUITECTURAS DE PROCESAMIENTO PARALELO	32
3.2.- PROCESADORES PARA SISTEMAS DE CONTROL	35
3.3.- REDES RECONFIGURABLES	37
3.4.- EL TRANSPUTER	40
3.4.1.- ARQUITECTURA INTERNA DEL TRANSPUTER	41
3.4.2.- ARQUITECTURAS BASADAS EN TRANSPUTERS	43
3.4.3.- COMUNICACIÓN ENTRE TRANSPUTERS	44
3.5.- TARJETA SMT004A	45
3.6.- COMUNICACIÓN ENTRE ELEMENTOS DE LA CELDA Y DISPOSITIVOS EXTERNOS	47
Capítulo 4.	
DISEÑO, DESARROLLO Y FABRICACIÓN DE LA INTERFAZ.	50
4.1.- DISEÑO DE LA INTERFAZ ENTRE TRANSPUTERS Y ROBOTS	53
4.1.1.- CIRCUITO MODO SALIDA	55
4.1.2.- CIRCUITO MODO ENTRADA	57
4.1.3.- CIRCUITO MODO BIDIRECCIONAL	59
4.2.- TOLERANCIA A FALLAS EN LA INTERFAZ	60
4.3.- CONEXIÓN DE LA INTERFAZ AL CONTROLADOR	63

## Capítulo 5.

PROGRAMACIÓN DE LA INTERFAZ	67
5.1.- MÓDULO DE COMUNICACIÓN TRANSPUTER - ROBOT	69
5.2.- MÓDULO DE COMUNICACIÓN ROBOT - TRANSPUTER	72
5.3.- COMUNICACIÓN BIDIRECCIONAL TRANSPUTER <-> ROBOT	74
5.4.- IMPLANTACIÓN DE LA TOLERANCIA A FALLAS	77

## Capítulo 6.

PRUEBAS Y RESULTADOS	81
6.1.- CIRCUITO PROTOTIPO (PROTO - BOARD) <-> SIMULADOR	81
6.2.- CIRCUITO PROTOTIPO (PROTO - BOARD) <-> CELDA	85
6.3.- CIRCUITO IMPRESO DEL CONTROLADOR <-> CELDA	87

## Capítulo 7.

CONCLUSIONES	90
7.1.- LIMITACIONES	91
7.2.- PROPUESTA DE MEJORAS A FUTURO	92
APÉNDICE A	94
CIRCUITO ADAPTADOR DE CANAL IMS C011 " LINK ADAPTOR "	94
APÉNDICE B	98
MICROCONTROLADOR 8751	98
ARQUITECTURA	98
MODOS DE OPERACIÓN	99
APÉNDICE C	103
INTERFASE SERIE RS232 - C	103
APÉNDICE D	104
PROGRAMAS	104
APÉNDICE E	113
DIAGRAMAS ELECTRÓNICOS	113
DIAGRAMAS DE CIRCUITO IMPRESO	117

---

APÉNDICE F	121
INFORMACIÓN TECNICA TARJETA SMT004	121
REFERENCIAS BIBLIOGRÁFICAS	128
BIBLIOGRAFÍA CONSULTADA	130

## LISTA DE FIGURAS, TABLAS Y FOTOGRAFÍAS

Fotografía (1.1)	CIM Campus Toluca	12
Figura (1.1)	Algunas topologías de redes	15
Figura (2.1)	Evolución del Hardware	19
Figura (2.2)	Elementos típicos de una red para un sistema de manufactura	20
Figura (2.3)	Rango de posibles soluciones para implementar una red	21
Fotografía (2.1)	Robots Júpiter, AS/RS, y banda transportadora	27
Fotografía (2.2)	Robot Mitsubishi	29
Fotografía (2.3)	Robot PUMA y maquinas herramientas	30
Figura (3.1)	Controlador paralelo distribuido	31
Figura (3.2)	Arquitecturas de procesamiento paralelo	33
Figura (3.3)	Arquitecturas MIMD	34
Figura (3.4)	Red estática	38
Figura (3.5)	Red dinámica	38
Figura (3.6)	Redes reconfigurables	39
Figura (3.7)	Arquitectura del IMS T800	42
Figura (3.8)	Redes de transputers	43
Figura (3.9)	Formato del enlace de datos	45
Fotografía (3.1)	Tarjeta SMT004A	46
Figura (3.10)	Módulo TRAM	46
Figura (3.11)	Interfaz RS 232-C	49
Figura (4.1)	Arquitectura del controlador	50
Figura (4.2)	Red dinámica	51
Figura (4.3)	Diagrama a bloques del controlador e interfaces	53
Fotografía (4.1)	Prototipo del controlador	54

---

Figura (4.4)	Módulo de salida	55
Figura (4.5)	Protocolo de salida del controlador de la CFM	56
Figura (4.6)	Módulo de entrada	57
Figura (4.7)	Protocolo de entrada del controlador de la CFM	58
Figura (4.8)	Módulo bidireccional	59
Figura (4.9)	Interfaz conmutada 2x1	61
Tabla (4.1)	Señales de Control del microcontrolador 8751	62
Figura(4.10)	Selección del enlace de comunicación	64
Figura(4.11)	Interfaz conmutada con caminos alternos	65
Figura(4.12)	Programación de la Tarjeta de Interfaz (Colocación de Jumpers)	66
Figura(5.1)	Comunicación Transputer-Robot	69
Figura(5.2)	Pseudo código del protocolo de salida	70
Figura(5.3)	Procesos generados por el transputer para el protocolo de salida	71
Figura(5.4)	Pseudo código del protocolo de entrada	72
Figura(5.5)	Procesos generados por el transputer para el protocolo de entrada	74
Figura(5.6)	Pseudo código del protocolo de bidireccional	75
Figura(5.7)	Procesos generados por el transputer para el protocolo bidireccional	76
Figura(5.8)	Diagrama a bloques del controlador e interfaces	77
Figura(5.9)	Pseudo código del protocolo de comunicación bidireccional para dos módulos TRAMS	79
Figura(6.1)	Sistema de prueba del prototipo	82
Fotografía(6.1)	Sistema de prueba del prototipo	82
Figura(6.2)	Diagrama a bloques del prototipo de prueba	83

---

Figura(6.3)	Prueba del prototipo con el robot PUMA	86
Fotografía(6.2)	Circuito impreso de la interfaz de la CFM	89
Figura(A.1)	Diagrama a bloques C011 modo 1	95
Figura(A.2)	IMS C011 modo 1 datos de entrada	96
Figura(A.3)	IMS C011 modo 1 datos de salida	97
Figura(B.1)	Arquitectura MCS 51	99
Figura(C.1)	Diagrama a bloques MAX 232	103
Figura(E.1)	Diagramas electrónicos de la interfaz de salida	113
Figura(E.2)	Diagramas electrónicos de la interfaz de entrada	114
Figura(E.3)	Diagramas electrónicos de la interfaz de final	115
Figura(E.4)	Diagramas Electrónicos del 8751 y el MAX232	116
Figura(E.5)	Circuito impreso de la interfaz: Leyendas	117
Figura(E.6)	Circuito impreso de la interfaz: Lado soldadura	118
Figura(E.7)	Circuito impreso de la interfaz: Lado componentes	119
Figura(E.8)	Circuito impreso de la interfaz: Capa anti-soldante	120
Figura(F.1)	Diagrama a Bloques Funcional de la Tarjeta SMT004	121
Tabla(F.1)	Mapeo del puerto de Entrada/Salida	122
Tabla(F.2)	Status de Entrada	123
Tabla(F.3)	Status de Salida	123
Tabla(F.4)	Registro de Error	124
Tabla(F.5)	Registro de Reset	124
Tabla(F.6)	Registro de Análisis	124
Tabla(F.7)	Velocidad de la Interfaz C011	125
Tabla(F.8)	Velocidad de los Enlaces TRAM	125
Tabla(F.9)	Subsistema de Control	126
Tabla(F.10)	Selección del Enlace C011	126
Tabla(F.11)	Conector de 40 vías.	127

# Capítulo 1

## INTRODUCCIÓN

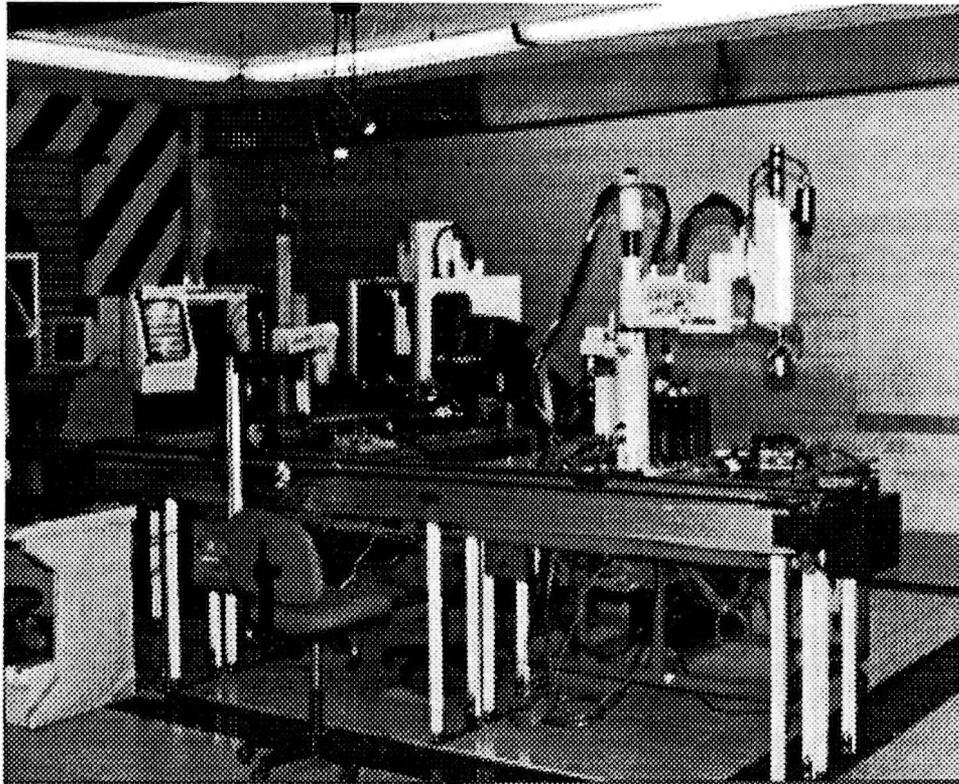
El propósito de las empresas de manufactura es convertir las materias primas en productos terminados demandados por el mercado consumidor, en el menor tiempo posible, al menor costo, con pocas pérdidas en material y mínimo uso de energía.

En muchas facetas de la manufactura moderna han surgido procesos automatizados. Las estaciones de trabajo robotizadas son cada vez más comunes y las tareas que éstas desempeñan son cada vez más variadas. El manejo de materiales, de partes, ensambles y materias primas para su procesamiento es de vital importancia en las fábricas, tal vez aún más crucial que el proceso de manufactura individual.

La Manufactura Integrada por Computadora (CIM) surge a mediados de la década pasada [2], al aparecer sistemas de cómputo que intervienen en el diseño, planeación, despacho y control de las operaciones que se realizan en la manufactura de algún producto. EL CIM es un paso en el camino dirigido hacia la automatización industrial.

Una celda de manufactura normalmente está compuesta por uno o varios manipuladores robóticos, máquinas herramientas (como tornos, fresadoras, centros de maquinado, etc.), sistemas de visión computarizada, plataformas móviles (robot-trailers), alimentadores y soporte de la materia prima (pallets) [3].

En la fotografía (1.1), se muestra un ejemplo de una celda de manufactura que corresponde al CIM que se encuentra en las instalaciones del ITESM Campus Toluca.



**Fotografía (1.1) CIM Campus Toluca**

En los sistemas de fabricación actuales es difícil producir piezas diferentes con una misma celda de manufactura, debido a que cambiar la producción de una celda resulta normalmente un proceso complicado. El advenimiento de nuevas tecnologías ha permitido desarrollar máquinas flexibles, mediante las cuales es posible fabricar diferentes piezas cambiando simplemente el programa de maquinado [1] en tornos, fresadoras y algunas otras máquinas herramientas de control numérico. Sin embargo, esta flexibilidad no existe en la actualidad, cuando se ve a la celda en conjunto.

Las celdas flexibles de manufactura son sistemas de producción computarizada que deben tener las siguientes características: flexibilidad en la planeación; reconfigurabilidad; y detección y aislamiento de fallas [13]. Una CFM normalmente es coordinada por un sistema de control distribuido el cual realiza dos funciones: el control de los robots y máquinas herramientas, y el control de las comunicaciones.

Este sistema de control permite realizar el planeamiento, temporización, órdenes de asignación, control y verificación de las tareas necesarias para fabricar un producto.

Las celdas flexibles de manufactura (CFM) implican la integración de todas las funciones de manufactura, en un sistema de procesamiento de datos distribuido. Es necesario que el sistema sea distribuido por dos razones principales: es muy difícil implementar todas las funciones en un sólo sistema; y los elementos de la celda están por lo general físicamente distribuidos en todo un piso de trabajo.

Un aspecto importante de los sistemas distribuidos es la interconexión de los elementos que constituyen la red, ya que existe una relación bien definida entre su desempeño y la topología de los sistemas, los algoritmos de ruteo, y los algoritmos de tolerancia y detección de fallas [4,5].

El propósito del proyecto de investigación en el que está incluida esta tesis [14], es la de diseñar un controlador paralelo basado en transputers que permita controlar los elementos de una celda de manufactura. Este sistema debe cumplir con las características de flexibilidad presentadas anteriormente, teniendo en cuenta que el principal problema a resolver es el de conectar eficientemente los elementos de la red (transputers), y los elementos propios de la celda de manufactura (robots).

En este trabajo de tesis se diseñó e implementó una interfaz mediante la cual se puede interconectar la red de transputers y los robots. La interfaz diseñada cumple con los requisitos de reconfigurabilidad y tolerancia a fallas del proyecto [14]. Además el diseño que se plantea es modular, logrando con ésto un sistema escalable.

El ITESM CEM tiene 5 años de experiencia trabajando en celdas de manufactura, participando principalmente en el diseño de la estructura de hardware del control de la celda, del software de operación y de la comunicación entre los elementos de la celda.

En un proyecto de investigación que se realizó con anterioridad [1] se diseñó un módulo programable para el control del CIM, en el que se utilizaba un microcontrolador para cada máquina herramienta, todos ellos interconectados a una computadora central a través de un multiplexor. En la computadora central se programaban todas las actividades del CIM, y la comunicación entre todos los procesadores se realizaba utilizando la interfaz serie RS 232 a 1200 bauds. En esta implementación se detectaron los siguientes problemas: 1) se generó un "cuello de botella" por la centralización de la comunicación, y 2) no se tuvo un sistema tolerante a fallas, ya que no se tenían caminos alternos para la comunicación entre los diferentes elementos.

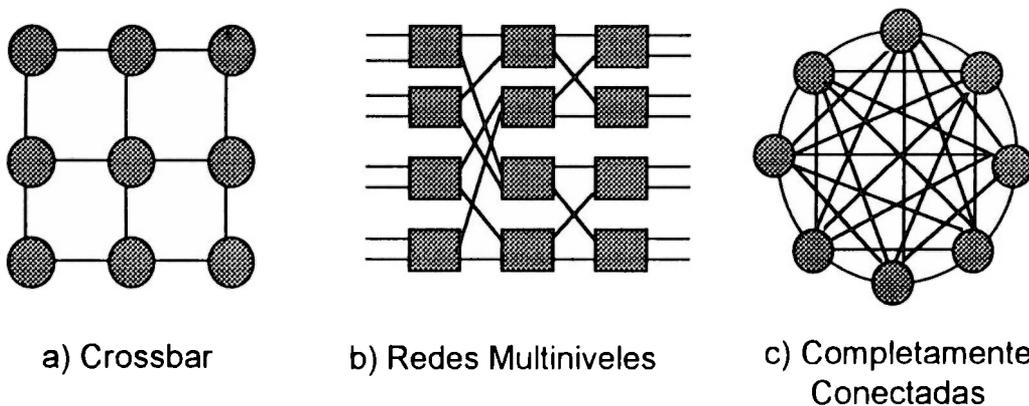
### **1.1.- PLANTEAMIENTO DEL PROBLEMA.**

Para que una CFM sea flexible, es necesario, entre otros aspectos, que la arquitectura del controlador sea reconfigurable, es decir que pueda cambiar fácilmente para adaptarse a las nuevas necesidades en la manufactura de un producto. Además esta arquitectura debe ser tolerante a fallas, por lo cual es necesario que haya redundancia en hardware y en software. En este proyecto se definirá y construirá una arquitectura que cumpla con estas características.

En lo que respecta a la tolerancia a fallas, es necesario que la arquitectura no tenga trayectorias únicas entre módulos, pues la falla en una trayectoria puede inducir al aislamiento de uno de los componentes del sistema. Para ello, se plantea una red de comunicaciones con trayectorias alternas y capaz de cambiar la topología de la red entre los elementos del controlador.

La redundancia de una red de comunicaciones puede lograrse por medio del uso de conmutadores [7] (cross-bar, figura (1.1.a)), y de canales virtuales [8]. En esta tesis se eligió explorar la primera opción.

El uso de conmutadores cross-bar permite el paso de varios mensajes a través de la red al mismo tiempo. El problema es que el número de conmutadores necesarios es proporcional a  $O(N^2)$ , donde  $N$  es el número de elementos conectados. Para reducir el costo a  $O(N \log_2 N)$  normalmente se construyen redes a varios niveles con crossbar pequeños (Multistage Interconnection Network: MIN). Una red multiniveles puede ser construida utilizando más conmutadores, lo que permite añadir caminos redundantes (figura 1.1.b) para la comunicación entre los robots, aumentando así la tolerancia a fallas.[6,16]



**Figura (1.1) ALGUNAS TOPOLOGIAS DE REDES**

Debido a los requerimientos de la aplicación, la red de control de la celda de manufactura debe funcionar en tiempo real. Ésto, aunado al requerimiento de redundancia en hardware, justifica la decisión de que el sistema sea realizado en sistemas multiprocesadores. Después de revisar varias opciones ( ver sección 3.1 ), decidimos basar la arquitectura en transputers [9].

Los cuatro canales de comunicación con que cuenta el transputer permiten la implementación de diferentes topologías como las que se muestran en la figura (1.1). Algunas de las éstas han sido propuestas e implementadas en diversas aplicaciones [9,10], algunas de ellas similares a la nuestra.

En las CFM existe la necesidad de comunicarse con dispositivos tradicionales de Entrada / Salida, tales como puertos paralelos Centronix, dispositivos con interfase RS -232 e interfaz IEEE - 488 en buses de instrumentación [10]. El diseño del controlador debe tomar en cuenta la diferencia de velocidades entre el transputer (20 Mbps) y estos dispositivos (del orden de 20 Kbps).

En esta tesis, el diseño de la topología de red para la comunicación de los elementos del controlador, se concentrará en los niveles físico y de enlace de datos, así como la interfaz con cada uno de los robots que integran la CFM.

## **1.2.- RESUMEN**

Los métodos que se proponen en esta tesis pretenden mejorar la flexibilidad y robustez de las celdas de manufactura desde el punto de vista de hardware, teniendo como objetivo contribuir a resolver los problemas de comunicación entre los elementos de una Celda Flexible de Manufactura.

Específicamente, se presenta el diseño de la arquitectura de un controlador para Celda Flexible de Manufactura (CFM). Esta arquitectura tiene un control distribuido y emplea el Transputer T805 de INMOS [9]. El transputer será el módulo principal, y tendrá como tarea controlar y enlazar cada una de las partes de la CFM.

En esta tesis se presenta en el capítulo 2, un análisis de la evolución del hardware así como las estrategias de comunicación que se utilizan en las celdas de manufactura. Se mencionan también las características de los diferentes protocolos de comunicación usados en la actualidad para este tipo de aplicaciones, para posteriormente presentar los robots con que cuenta la celda de manufactura del Campus Toluca.

En el capítulo 3 se da un panorama general de las diferentes arquitecturas de procesamiento paralelo y se presenta el Transputer, así como algunos conceptos fundamentales que se utilizan para el diseño de la interfaz que se presenta en el capítulo 4.

En los capítulos 4 y 5 se presentan los criterios de diseño y fabricación de la interfaz propuesta, tanto en hardware (capítulo 4), como en software (capítulo 5), describiendo paso a paso cada una de las etapas de diseño y cómo funcionan, hasta llegar al prototipo final en circuito impreso.

El capítulo 6 y 7 presentan los resultados y pruebas que se realizaron al sistema, llegando al final a presentar una serie de posibles mejoras en trabajos futuros.

Los apéndices muestran la información técnica de cada una de las etapas de diseño, así como los programas y circuitos desarrollados.

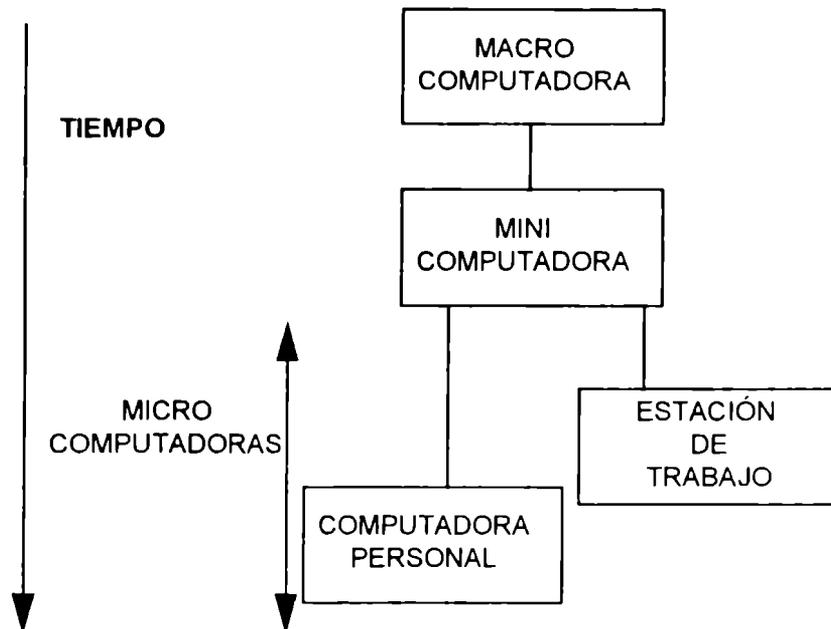
## Capítulo 2

# CONTROLADORES PARA CFM

En las últimas tres décadas, el desarrollo de las computadoras ha evolucionado rápidamente, desde microcontroladores hasta minicomputadoras (figura 2.1). Durante los años 1960, el desarrollo principal se centraba en computadoras grandes o "mainframes". Estas primeras computadoras eran demasiado grandes para los estándares actuales, estaban limitadas en capacidades de memoria y velocidad, y requerían de un gran número de especialistas para supervisar su operación y mantenimiento. Estos mainframes, permanecieron en la industria en el periodo 1960 - 1970. Sin embargo, durante este tiempo las capacidades de cómputo aumentaron al mismo tiempo que las computadoras disminuían su tamaño [15].

En 1970 se introducen las minicomputadoras, las cuales proliferan en la industria rápidamente, debido a sus capacidades de cómputo y reducido tamaño, y al proporcionar una mayor flexibilidad para interconexión en redes.

Las capacidades de hardware evolucionan rápidamente, y pronto es posible ensamblar computadoras más pequeñas llamadas microcomputadoras, las cuales pueden realizar funciones que anteriormente sólo podían hacer grandes mainframes o minicomputadoras. Estas microcomputadoras fueron desarrolladas para aplicaciones muy especiales.



**Figura (2.1) Evolución del Hardware**

La situación cambia drásticamente cuando en 1981 IBM introduce al mercado la computadora del tipo personal y crea un estándar para este tipo de sistemas. Durante ese mismo período de tiempo, pequeñas pero poderosas computadoras llamadas estaciones de trabajo ("workstation") son diseñadas para soportar aplicaciones de ingeniería y otras aplicaciones que demandan diseño asistido por computadora (CAD).

El gran desarrollo de estos equipos, permitió desplazar a los sistemas mainframes y minicomputadoras de aplicaciones que se encuentran en la industria. En lugar de utilizar terminales remotas conectadas a un mainframe, fue posible usar microcomputadoras interconectadas a través de una red.

La disponibilidad de las microcomputadoras hace posible el desarrollo de una nueva generación de controladores para la industria. Muchas compañías desechan sus antiguos controladores, para usar los nuevos y poderosos controladores basados en microcomputadoras IBM PC. La producción de nuevos equipos orientados hacia la automatización e integración de sistemas, se incrementó por la disponibilidad de estas

microcomputadoras. Al mismo tiempo, nuevo hardware y software fue desarrollado para permitir la interconexión entre computadoras. Esta evolución estimula el crecimiento del diseño asistido por computadora (CAD), como un aspecto esencial de la ingeniería. Las estaciones de trabajo llegan a ser una herramienta de diseño necesaria, utilizadas en un gran rango de aplicaciones de ingeniería y la industria, donde se requiere aumentar los niveles de productividad y calidad en productos.

Un esfuerzo significativo se está realizando en la actualidad para decidir sobre la mejor combinación costo beneficio en la configuración de computadoras para celdas de manufactura. Pero la tendencia es hacia el uso de procesamiento distribuido, con muchas computadoras interconectadas a través de una red.

## 2.1.- ESTRATEGIAS PARA COMPARAR ALTERNATIVAS DE COMUNICACIÓN

Debido a los avances que se han realizado en los sistemas de manufactura, hoy en día se requieren de redes de computadoras que interconecten diversos equipos en un sistema y permitan la comunicación con el controlador de la celda figura (2.2).

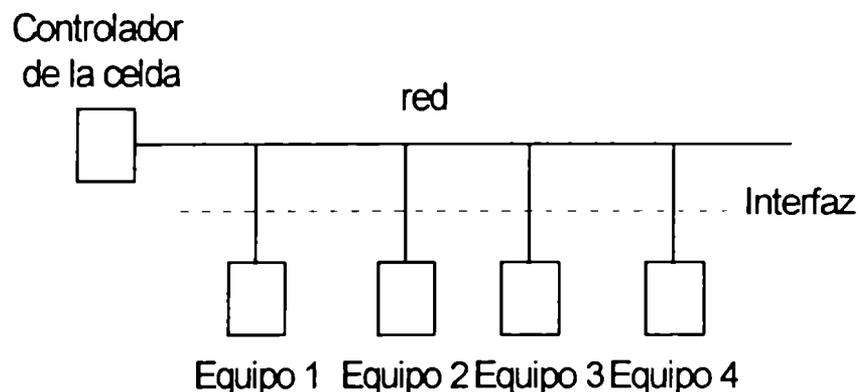


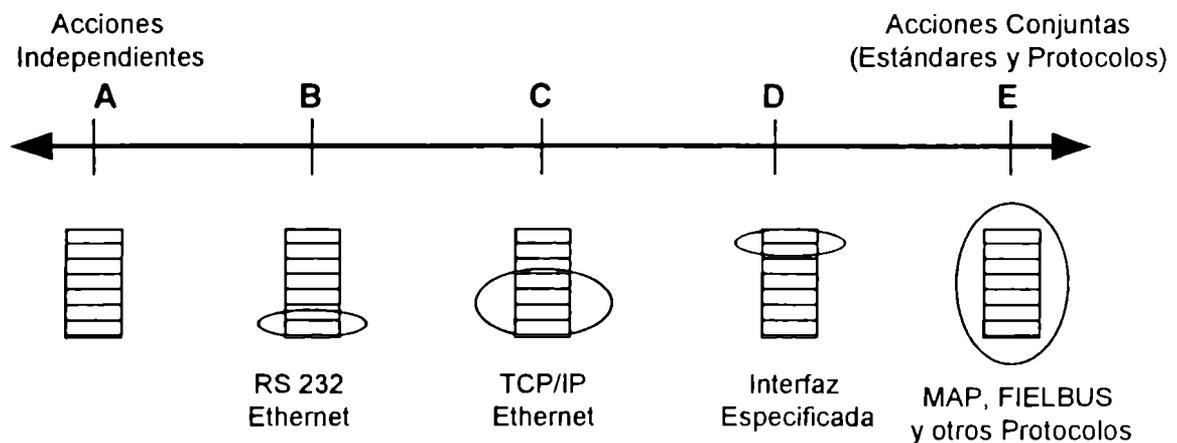
Figura (2.2) Elementos típicos de una red para un sistema de manufactura

BIBLIOTECA



La figura (2.3) representa el rango de soluciones que puede existir para este problema. El extremo izquierdo de la línea ( posición A ) corresponde a una situación en la cual cada equipo de manufactura trabaja independientemente en el desarrollo de una especificación, sin el requerimiento de comunicación entre computadoras.

En el otro extremo (posición E) se encuentran las celdas que requieren acciones conjuntas entre sus elementos, para lo que son necesarios estándares y protocolos de comunicación. Uno de los mayores problemas en el desarrollo de protocolos es el entender exactamente la naturaleza de la comunicación entre 2 ó más entes. Si el protocolo no es entendido exactamente por todos los participantes del grupo, entonces la comunicación deseada no se realizará eficientemente. Otro de los problemas que se manifiesta a veces es la interoperatividad, que se presenta cuando intervienen diversos fabricantes.



**Figura (2.3) Rango de posibles soluciones para implementar una red**

Entre los dos extremos existen otros enfoques que se han tomado en cuenta para lograr la comunicación entre computadoras. En la opción B se toman en cuenta las capas inferiores de un sistema de comunicación, para lograr la interconexión física y procesamiento de mensajes. Los enlaces de comunicación RS 232 y Ethernet son comúnmente utilizados para lograr la transferencia de información a este nivel. Muchos

equipos cuentan con estas interfaces, permitiendo que sean fácilmente integrados en un sistema.

En la opción C, las capas intermedias e inferiores del sistema de comunicación están definidas por protocolos, o estándares, como TCP/IP.

En la opción D se muestra un enfoque diferente en el que las capas de comunicación bajas y medias quedan libres para que los fabricantes las definan, mientras que los integradores de sistemas deben definir las capas superiores. Esta estrategia funciona como un puente para que los fabricantes y los integradores de sistemas trabajen juntos.

Finalmente, en la opción E, todos los aspectos de comunicación deben estar definidos a través de un estándar o protocolo. Su desempeño puede ser bueno, pero no se aplica fuertemente en la industria debido principalmente a la gran cantidad de equipo diverso que actualmente se usa.

A continuación se describe cada una de las alternativas:

### **Redes basadas en RS 232**

Una red de computadoras puede ser configurada usando conexiones RS 232 punto a punto entre ellas. Este tipo de red requiere de múltiples interfaces RS 232 (del orden de  $n^2$ ) en cada unidad de equipo.

El estándar RS 232 describe las señales eléctricas necesarias para transferir información bit a bit de un elemento del sistema a otro. Los niveles superiores en los protocolos de comunicación no son considerados, sólo la capa física.

## Ethernet

Ethernet es el nombre de una de las redes de área local más populares hoy en día. Inventada por Xerox PARC a principios de los 70's, fue estandarizada por Xerox Corporation, Intel Corporation y Digital Equipment Corporation en 1978. La tecnología Ethernet consiste fundamentalmente en un cable coaxial llamado "ether" de aproximadamente ½ pulgada de diámetro y hasta 500 metros de longitud (figura 2.2). Cada segmento de cable puede ser interconectado a otro por medio de dispositivos llamados repetidores, que duplican señales eléctricas de un cable a otro. Sólo dos repetidores pueden ser usados entre dos computadoras, por lo que la longitud máxima de una red Ethernet es bastante moderada (1500 metros) [19].

En la mayoría de los sistemas de comunicación entre computadoras, existe un medio para controlar cuándo un nodo está transmitiendo sobre la red y cuándo está escuchando para recibir un mensaje. En Ethernet, si un nodo trata de transmitir al mismo tiempo que otro nodo lo está haciendo, existe un conflicto, por lo que ambos esperan un tiempo aleatorio antes de intentar transmitir nuevamente.

Este tipo de control de acceso para redes de computadoras es conocido como CSMA/CD (Carrier-Sense Multiple-Access with Collision Detection). Ethernet y soluciones similares son aplicadas en redes donde el tiempo máximo de espera no es limitado. Por otro lado este tipo de redes son simples de instalar, y existen una gran cantidad de productos de hardware y software para su soporte.

## TCP/IP

TCP/IP o "Transmission Control Protocol / Internet Protocol", es un protocolo de las capas de transporte y de red del modelo OSI (Open System Interconnect). TCP/IP es frecuentemente combinado con Ethernet, definiendo entre ambos los aspectos de las capas bajas y medias de un sistema. TCP/IP funciona dividiendo los mensajes en

"paquetes", y entonces enviando estos paquetes de hasta 64 kbytes a través de la red de comunicación. TCP/IP reensambla en el orden correcto los paquetes que el usuario recibe.

## **PROCOLO PARA LA AUTOMATIZACIÓN DE MANUFACTURA (MAP)**

El protocolo para la automatización de manufactura (MAP), fue desarrollado específicamente para ser utilizado en el medio ambiente de la industria, surgiendo como una necesidad al tener una gran variedad de equipos en las industrias [15]. General Motors (GM), inicia el desarrollo teniendo en cuenta la necesidad de transferir grandes volúmenes de información y proveer inmunidad al ruido, en el medio ambiente industrial.

Existen dos métodos de señalización usados por MAP en las redes de área local. En señalización tipo "broadband", el ancho de banda del medio de transmisión se subdivide en subfrecuencias para formar dos o más subcanales, donde en cada subcanal se permite la transferencia de información independientemente de los otros. En señalización "baseband", solo se transmite una señal en el medio en un momento dado. Broadband es más complejo que baseband, porque requiere que la información se transmita por medio de la modulación de una señal portadora, y por lo tanto requiere del uso de tipos especiales de modems.

Cualquier nodo MAP tiene la posibilidad de realizar el control de las comunicaciones sobre la red, y comunicarse con otro nodo dentro de un cierto tiempo de espera, debido a que MAP está implantado como un sistema de paso de "tokens". Un nodo puede realizar una comunicación solamente cuando tiene la palabra de control llamada "token". El token es utilizado para prevenir colisiones en los mensajes, y también asegura que para una configuración del sistema, el tiempo máximo de espera sea limitado.

En la selección de estrategias de comunicación, el objetivo típicamente es decidir sobre los niveles de desempeño que se requieren, al mínimo costo. MAP es un protocolo que resulta costoso en sistemas pequeños, por lo que debe ser considerado solamente en las circunstancias que lo demanden para un mejor desempeño.

## **FIELDBUS**

El desarrollo de dispositivos programables (microprocesadores) en 1980 permitió reemplazar el estándar de interfaz para controladores de lazo de corriente 4-20 mA, con un sistema de comunicación abierto, independiente de los fabricantes, bidireccional, con múltiples conexiones, conocido como Fieldbus. El Fieldbus proporciona un enlace de comunicación rápido entre sistemas de control y dispositivos digitales [17].

El protocolo Fieldbus está basado en el modelo OSI, en las capas 1,2,7. Las capas 3 a la 6 son evitadas para propósitos de tiempo real. Existe además una nueva capa de usuario en la parte superior de la capa de aplicación, donde ciertas estrategias de control distribuido se han implementado, con lo cual el Fieldbus resulta más poderoso para aplicaciones industriales. El Fieldbus se clasifica, de acuerdo al grado de complejidad del paquete requerido para comunicarse con un dispositivo, en tres categorías: Sensorbus (nivel de bit), Devicebus (nivel de byte) y Fieldbus (nivel de bloque con control de datos y procesos) .

## **COMPARACIÓN DE ESTRATEGIAS EN REDES**

La interfaz RS 232 provee un medio para crear conexiones punto a punto entre equipos. Esta interfaz define solamente la capa física de los requerimientos de comunicación y no proporciona información referente a las capas superiores.

Ethernet proporciona los medios para crear una red de área local, definiendo los dos niveles bajos de un sistema de comunicación. Ethernet requiere de una completa definición de las capas superiores del software de comunicación, y esto influye para que el costo de los sistemas sea mayor al asociado a la interfaz RS - 232.

La decisión de usar TCP/IP puede simplificar los problemas de desarrollo de software. Si en una red de área local es implementada utilizando Ethernet, a ésta se le puede agregar TCP/IP para el control de mensajes en las capas intermedias, y el desarrollo del sistema de comunicación se centraría en las capas de aplicación y presentación de un sistema de comunicación, así como a los manejadores ("drivers"), de las interfaces asociadas con los equipos.

MAP continúa con su desarrollo para ser un importante protocolo para aplicaciones en el ambiente de la celdas de manufactura. Existen versiones para industrias grandes (broadband) y pequeñas (baseband), sin embargo típicamente el costo de estas redes es alto debido principalmente a la complejidad del protocolo y a la dificultad en el desarrollo y mantenimiento del mismo.

FIELDBUS esta diseñado para conectar componentes de procesos distribuidos como computadoras, sensores inteligentes, actuadores, módulos de entrada / salida, controladores centrales. PLC's, etc., por medio de un cable de 2 o 4 conductores. El bus provee un intercambio bidireccional de información concerniente a la posición de los actuadores, límite de las válvulas, señales de realimentación, señales de falla y los datos internos del sistema [17,18]. El uso del Fieldbus con sensores y actuadores inteligentes ofrece un mejor diagnóstico para el control de procesos en comparación con las redes tradicionales. Sin embargo, pocos dispositivos de una celda requieren el nivel de desempeño de Fieldbus, por lo que los fabricantes rara vez ofrecen esta interfaz en sus equipos, como veremos en la sección siguiente.

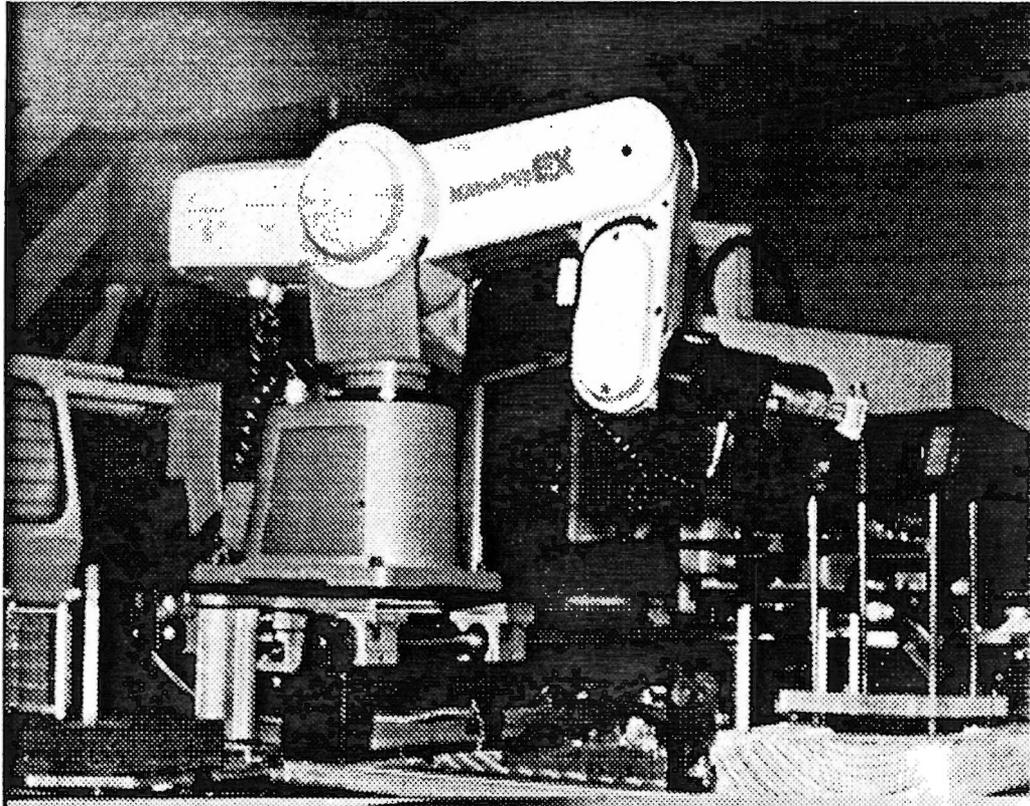
**Robot AMATROL JÚPITER:** Este robot de fabricación norteamericana, es un robot de 4 grados de libertad servocontrolados y con la característica que tiene 3 actuadores finales (dos tenazas y un destornillador). El área de trabajo tiene la forma de una rebanada de pastel por lo que se le llama de tipo "Scara". Tiene muchas aplicaciones para la inserción de componentes ya que éste robot nunca pierde la verticalidad en sus ejes (fotografía 2.1).

El controlador servo robot (899-CC) puede comunicarse con otras computadoras a través del puerto de comunicación serial RS 232-C, y la programación o control puede darse desde un telecomando (825 -TP) [20].

**Robot AS/RS:** Este robot es de la marca Amatrol, por lo que su interfaz es muy parecida a la del Júpiter, pero con sólo dos grados de libertad (dos ejes) servocontrolados. Los movimientos de girar, entrar o salir del contenedor de material y del actuador son no-servocontrolados. La función principal del AS/RS es la de proporcionar materia prima a la celda, así como el almacenar el producto terminado. Es un robot del tipo rectangular, ya que su área de trabajo es un prisma rectangular (fotografía 2.1).

**Robot MITSUBISHI:** Este robot es de fabricación japonesa y tiene cinco grados de libertad, servocontrolados con un actuador final eléctrico al que se le puede regular la presión de trabajo. La base está montada sobre un dispositivo no-servocontrolado que lleva al robot a una mesa de control de calidad; este robot también es del tipo esférico (fotografía 2.2).

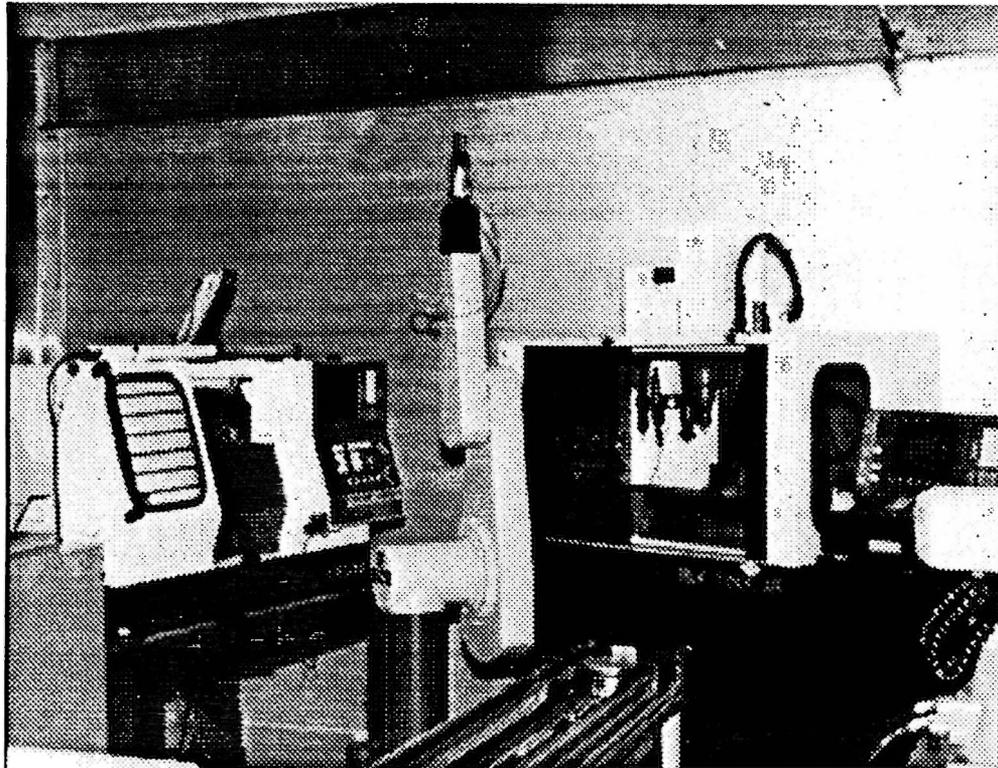
El Mitsubishi cuenta con tres diferentes tipos de interfaz de comunicación: interfaz paralela centronix, interfaz serie RS 232-C y una interfaz con equipo externo de entrada / salida [20].



**Fotografía (2.2) ROBOT MITSUBISHI**

**Robot PUMA:** Es un robot de fabricación sueca (fotografía 2.3) que tiene seis grados de libertad y es controlado por servomotores. El actuador final es una tenaza de accionamiento neumático. Este robot se clasifica como del tipo esférico ya que su área de trabajo es una esfera; las principales acciones que realiza son las de alimentar el centro de maquinado y al torno con piezas.

Los componentes periféricos para el sistema PUMA son usados para enviar y recibir información. Estos periféricos consisten de una terminal CRT y TTY, un "teach pendal" (control), disco duro, memoria adicional y módulos de entrada salida [20].



**Fotografía (2.3) ROBOT PUMA Y MAQUINAS HERRAMIENTAS**

**TORNO y FRESA:** Son máquinas herramientas de la celda que funcionan con un programa de control numérico computarizado (fotografía 2.3).

Vale la pena mencionar que los equipos CIM cuentan con dispositivos para el control de calidad como es un sistema de visión integrado por una cámara de vídeo (blanco/negro) y un sistema de digitalización de imagen. Además, los equipos cuentan con sistemas de seguridad (sensores de colisión, sensores de proximidad, sensores de máximo desplazamiento) para su programación y manejo.

Los robots de la celda de manufactura de Campus Toluca cuentan con interfaces de comunicación serie RS 232-C para su programación desde un equipo de cómputo, así como interfaces paralelas para dispositivos de entrada salida.

Para lograr construir y operar un sistema de control para la CFM será de suma importancia el multiprocesamiento, teniendo principal cuidado en seleccionar dispositivos que nos permitan resolver eficientemente los problemas de comunicación:

### **3.1.- ARQUITECTURAS DE PROCESAMIENTO PARALELO**

En la pasada década un grupo importante de centros de investigación se dieron a la tarea de desarrollar arquitecturas con un alto grado de paralelismo [6]. El avance en la investigación de tecnologías de circuitos integrados VLSI (Very Large Scale Integrated Circuit) y VHSIC (Very High Speed Integrated Circuit), hizo posible considerar como viable la construcción de arquitecturas paralelas complejas, con la interconexión de un número considerable de procesadores convencionales.

Las arquitecturas paralelas pueden utilizarse para controlar celdas flexibles de manufactura, pues los estrictos requerimientos de confiabilidad en tiempo real se pueden cumplir, al emplear redundancia en los recursos de procesamiento[4].

En esta sección mostraremos como el transputer de INMOS [7] puede ser usado como bloque básico para el diseño de tales sistemas, ya que estos circuitos proporcionan un soporte adecuado para comunicaciones y computación concurrente.

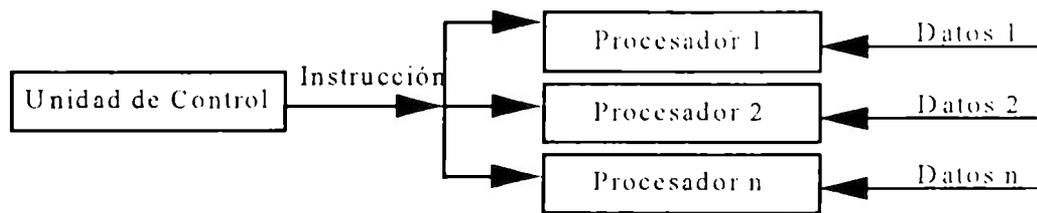
Los procesadores pueden ser conectados de diferentes formas para dar una gran variedad de arquitecturas de procesamiento paralelo. Similarmente, una variedad de métodos y paradigmas han sido propuestos para programar estas máquinas. Las arquitecturas paralelas son comúnmente clasificadas de acuerdo a la taxonomía de Flynn [9] que describimos a continuación.

**Instrucciones Únicas - Datos Múltiples (SIMD)**

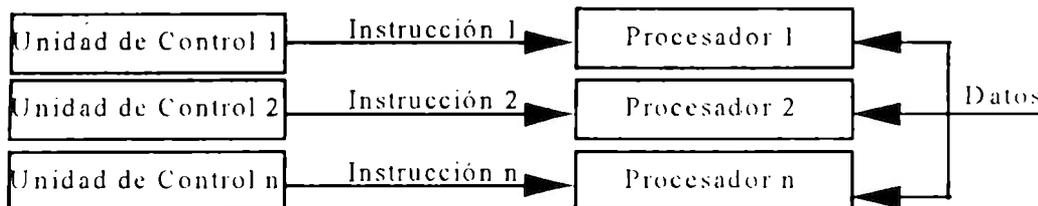
En este tipo de arquitectura, varios procesadores ejecutan simultáneamente la misma instrucción sobre múltiples secuencias de datos, como se muestra en la figura (3.2.a). Ejemplos de esta arquitectura son los arreglos de procesadores como el ICL DAP y el TMC Connection Machine [9].

**Instrucciones Múltiples - Datos Únicos (MISD)**

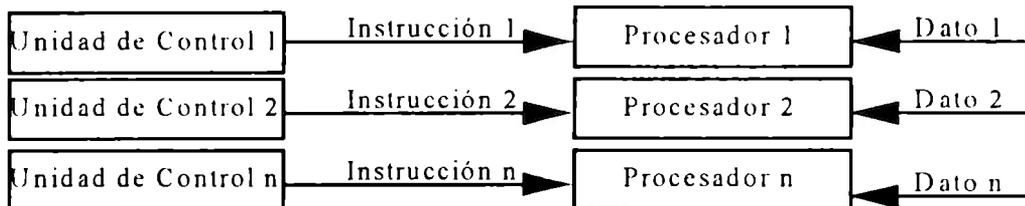
Varios procesadores simultáneamente ejecutan diferentes secuencias de instrucciones sobre una sola serie de datos, como se muestra en la figura (3.2.b). Un sistema " pipeline " es típicamente un ejemplo de una arquitectura MISD.



a) SIMD Flujo de Instrucción Simple, Flujo de Datos Múltiples



b) MISD, Flujo de Instrucciones Múltiple y flujo de datos simples



c) MIMD Flujo de Instrucciones Múltiples, Flujo de Datos Múltiples

**Figura (3.2) ARQUITECTURAS DE PROCESAMIENTO PARALELO**

### Instrucciones Múltiples - Datos Múltiples (MIMD)

En esta clase de máquinas, cada procesador puede estar simultáneamente desarrollando diferentes instrucciones sobre diferentes datos, como se muestra en la figura (3.2.c).

Los procesadores normalmente tienen una memoria local, pero cada uno puede tener acceso a una memoria global común compartida como se muestra en la figura (3.3.a).

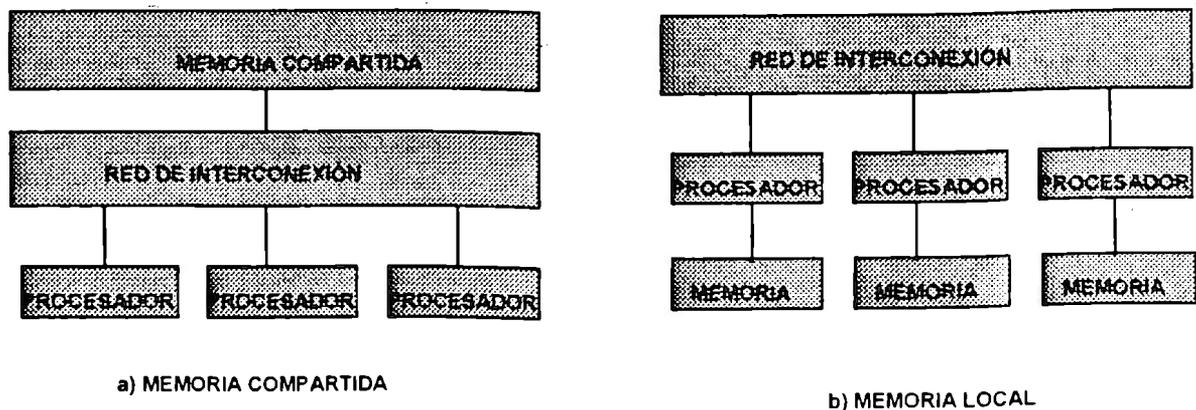


Figura (3.3) Arquitecturas MIMD

Los procesadores MIMD ofrecen la promesa de alta flexibilidad en su arquitectura y es por ello que es la arquitectura dominante en máquinas paralelas. Son frecuentemente utilizadas en un modo pseudo-SIMD (un sólo programa, datos múltiples ó SPMD) donde todos los procesadores ejecutan el mismo programa, sobre datos diferentes.

Las arquitecturas basadas en transputers pueden ser usados para construir arquitecturas MIMD con memoria local como se muestra en la figura (3.3.b).

### **3.2.- PROCESADORES PARA SISTEMAS DE CONTROL**

La selección de un microprocesador es influenciada por muchos factores, como pueden ser técnicos, económicos o circunstanciales. Las consideraciones técnicas incluyen velocidad, capacidades de entrada / salida, consumo de potencia, conjunto de instrucciones y conjunto de registros. Los factores económicos incluyen no sólo el precio unitario del dispositivo, sino la facilidad de programación, facilidad para corrección de errores (debugging), y la existencia de fabricantes alternos. La selección final frecuentemente depende de consideraciones circunstanciales, por ejemplo si el grupo de ingenieros de diseño tiene experiencia con una familia de microprocesadores en particular, y el laboratorio tiene las herramientas de diseño para estos microprocesadores [21].

Los microprocesadores son comúnmente caracterizados por el número de bits que procesan en una sola instrucción, los cuales son generalmente determinados por el tamaño del canal de datos interno. Otra distinción es entre microcomputadoras en un solo circuito integrado, que incluyen un microprocesador, memoria y puertos de entrada/ salida en el mismo circuito, y microprocesadores de propósito general, los cuales requieren de memoria externa y circuitos de entrada / salida.

Los microprocesadores han evolucionado básicamente en dos direcciones. Una dirección es en el desarrollo de microprocesadores más poderosos, con canales de datos internos más grandes (anchos) y un mejor conjunto de instrucciones; el otro es en desarrollo de sistemas con mayor integración, combinando los microprocesadores, con circuitos de memoria y circuitos de entrada / salida en el mismo circuito integrado para obtener una computadora en un solo "chip" (single chip computer). Las computadoras en un sólo circuito integrado son frecuentemente llamados *microcontroladores*, ya que generalmente son utilizados en aplicaciones de control.

Los procesadores "Bit-slice" son "rebanadas" de CPU que pueden ser utilizadas para hacer un procesador de cualquier tamaño en el canal de datos. Por ejemplo, si se tiene cuatro circuitos procesadores bit-slice de 8 bits, pueden ser utilizados para hacer un procesador de 32 bits. Los procesadores bit-slice son significativamente diferentes a los procesadores de propósito general. Una de las mayores diferencias entre los procesadores estándar y un procesador bit-slice es la localización del microcódigo, el cual le indica al microprocesador todos los pasos a detalle requeridos para ejecutar una instrucción en lenguaje máquina. En los procesadores estándar, el microcódigo se encuentra almacenado en una memoria ROM o en un arreglo lógico programable (PLA), es parte del microprocesador y no puede ser modificado por el usuario. Los procesadores Bit-slice, por el otro lado, requieren de memoria ROM externa donde se encuentra almacenado el microcódigo que el usuario debe programar. Los sistemas basados en procesadores Bit-slice, son más complicados de implementar que la mayoría de los sistemas basados en microprocesadores estándar. Los procesadores Bit-slice son utilizados en aplicaciones donde se requiere mayor velocidad de procesamiento, pero estos sistemas son necesariamente más complejos, caros y con un alto consumo de potencia [21].

Los procesadores digitales de señales (DSP) son microprocesadores diseñados para implementar filtros y otras funciones de procesamiento de señales. Un DSP incluye normalmente un conjunto de instrucciones óptimo para el procesamiento de señales (FFT, convolución, filtros, etc), y convertidores analógico / digitales, y digitales / analógicos. Los DSP son ampliamente usados en la industria por su potencial y facilidad de uso.

Sin embargo, en ciertas aplicaciones las necesidades de procesamiento son tan grandes que los procesadores no diseñados para procesamiento paralelo resultan inadecuados.

Los productos de la familia de DSP de la generación TMS320C3X avanzan en esta dirección de procesamiento paralelo, proporcionando a los usuarios dos puertos externos, cada uno con una interfaz de memoria, con lo cual se logra aumentar el ancho de banda para entradas / salidas. Los dispositivos de la siguiente generación, TMS320C4X, incorporan dentro del mismo circuito facilidades de comunicación de alta velocidad entre procesadores. Esta característica se debe principalmente a que cuenta con : seis puertos externos de alta velocidad para comunicación entre procesos; seis puertos de DMA para operaciones concurrentes; alto desempeño del CPU capaz de realizar 275 millones de operaciones por segundo (MOPS); dos canales de datos y direcciones idénticos que soportan sistemas de memoria compartida; y un módulo analizador del circuito integrado mediante el cual se puede monitorear el estado del procesamiento paralelo [22].

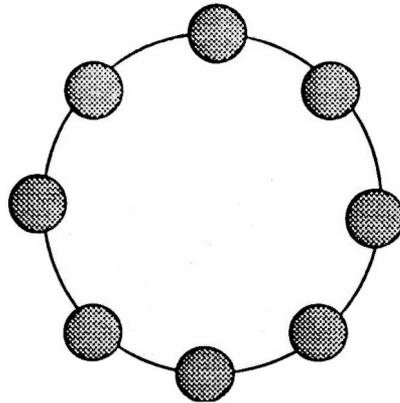
En la sección 3.4 hablaremos de otro procesador paralelo que es muy utilizado en aplicaciones de control, por las facilidades que provee para la comunicación: el transputer.

### **3.3.- REDES RECONFIGURABLES.**

La manera como los procesadores se conectan entre sí es importante para el desempeño de un sistema paralelo. Existen dos tipos principales de topologías: estáticas y dinámicas.

#### **Interconexión estática.**

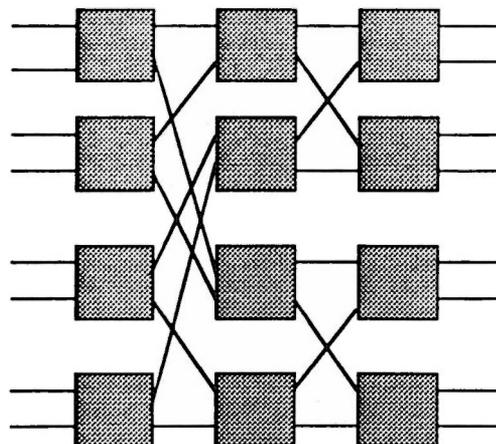
En este tipo de redes, los enlaces entre procesadores no pueden cambiar durante la ejecución de un programa. El problema con este tipo de redes es que es difícil lograr escalabilidad y reconfigurabilidad; un ejemplo se muestra en la figura (3.4).



**Figura (3.4) RED ESTÁTICA**

### **Interconexión dinámica.**

Las redes dinámicas no tienen enlaces fijos sino conmutados. Estas topologías trabajan en base a conmutadores (switches), que pueden seleccionar la trayectoria entre entradas y salidas. Las posiciones del conmutador son establecidas por software, ya sea con mensajes de control o con los mismos mensajes de datos. Esto permite tener arquitecturas reconfigurables, es decir que cambien según las necesidades del programa. Se presenta un ejemplo en la figura (3.5).



**Figura (3.5) RED DINÁMICA**

El uso de redes reconfigurables permite que el tiempo de comunicación se reduzca, al disminuir el número de procesadores intermedios a través de los cuales tiene que viajar un mensaje.

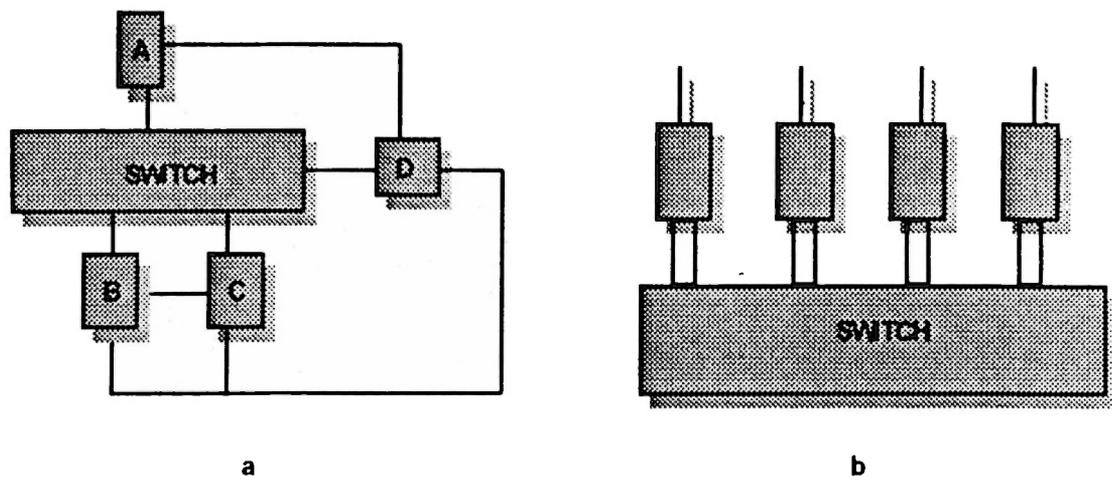


Figura (3.6) REDES RECONFIGURABLES

Las redes reconfigurables dan un cierto grado de tolerancia a fallas ya que procesadores que han fallado pueden ser evitados (bypass) en la red. Un ejemplo se muestra en la figura [3.6.a], donde existe una conexión punto a punto entre el procesador A y el procesador D. Si este enlace de comunicación falla, se tienen caminos alternos a través del conmutador, mediante el cual se pueden lograr los cambios en los enlaces de comunicación entre los procesadores, definiendo las trayectorias A -> C -> D, o bien, A -> B -> D.

La topología de la red puede ser cambiada estáticamente, quasi - estáticamente o dinámicamente. En la conmutación estática, la topología se elige antes de que los programas sean cargados y la aplicación esté corriendo.

La conmutación quasi - estática es usada cuando todos los procesadores pueden ser sincronizados en un punto predeterminado en el programa, en el cual, todas las comunicaciones cesan y todos los procesadores esperan para que la conexión de los

La sincronización es requerida cuando diferentes partes de un sistema, cada una con diferentes tiempos de ejecución, necesitan interactuar o cooperar entre ellos. Ésto se realiza en Occam cuando los procesos intercambian mensajes.

Una característica importante de la tecnología VLSI es que la comunicación interna a un dispositivo es mucho más rápida que la comunicación entre diferentes dispositivos. En una computadora, casi todas las operaciones que realiza el procesador involucran el uso de memoria. Por esta razón el transputer incluye en un sólo circuito integrado al CPU, y la memoria.

### **3.4.1.- ARQUITECTURA INTERNA DEL TRANSPUTER.**

Los miembros de la familia de transputers comprenden procesadores de diferente longitud de palabra (IMS T212 de 16 Bits, IMS T414 de 32 Bits, IMS T800 de 32 Bits con Unidad de Punto Flotante). Por esta razón, el conjunto de instrucciones usado por ellos ha sido diseñado independientemente de la longitud de palabra del procesador.

El IMS T800 es un miembro de la familia que comprende un procesador de 32 Bits, una unidad de punto flotante de 64 Bits, 4 canales de comunicación serial de alta velocidad, 4 Kbytes de memoria interna y una interfaz para memoria externa. Todo está en el mismo circuito integrado (figura 3.7), por lo que el transputer es una "computadora en un chip".

El microprocesador trabaja a 20 Mhz, y puede realizar 10 millones de instrucciones por segundo ( MIPS ). Su mínimo conjunto de instrucciones (RISC) y registros de 32 bits están diseñados para maximizar la velocidad. Cuenta con un manejador de prioridades microprogramado, mediante el cual se pueden tener dos niveles de interrupciones.

enlaces sea cambiada. Una vez que la topología ha sido cambiada, la ejecución continúa.

En la conmutación dinámica (figura 3.6.b), cualquier conexión entre dos procesadores puede ser cambiada en cualquier tiempo, con tal de que no existan comunicaciones en tránsito sobre la conexión a cambiar. La conmutación dinámica permite usar nuevos métodos de programación, tales como el balanceo dinámico de cargas.

### **3.4.- EL TRANSPUTER**

El transputer es un microprocesador para arquitecturas paralelas desarrollado por INMOS [7]. El nombre de transputer es una contracción de dos palabras: Transistor y Computadora. (**Trans** de transistor y **puter** de computer).

El transputer es un microprocesador de propósito general que integra en un sólo circuito las facilidades de procesamiento, comunicaciones y memoria [7]. Fue desarrollado en 1980, y tiene asociado un lenguaje de programación de alto nivel, llamado Occam [9]. El transputer es una nueva generación de circuitos fabricados en tecnología VLSI, con interesantes facilidades para concurrencia y sincronización [12].

Los transputers son poderosos microprocesadores que incorporan interfaces de comunicación en su circuito integrado. La presencia de cuatro enlaces de comunicación permite teóricamente una expansión lineal en la capacidad de procesamiento de un sistema de transputers, al incorporar más nodos [8].

Un sistema paralelo puede ser construido como una colección de transputers conectados a través de enlaces de comunicación [9]. Occam, el lenguaje de programación de los transputers, permite colocar fácilmente procesos paralelos y establecer canales de comunicación entre transputers. Al ser modular el código, se puede ejecutar en uno o varios transputers, lo que permite tener una reconfiguración dinámica durante la operación del sistema.

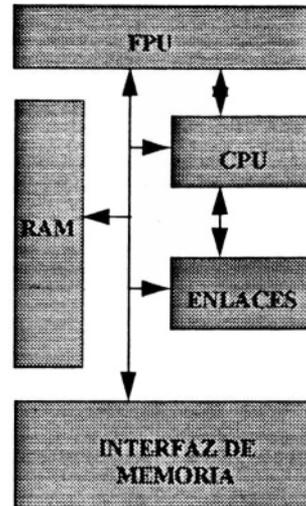


Figura (3.7) ARQUITECTURA DEL IMS T800

La unidad de punto flotante puede desarrollar operaciones aritméticas de 32 ó 64 bits. Esta unidad opera concurrentemente con el microprocesador y puede realizar tres millones de operaciones en un segundo ( MFLOPS ).

El transputer tiene 4 enlaces de comunicación para comunicarse eficientemente con otros transputers. Cada uno de los enlaces de comunicación provee una comunicación punto a punto bidireccional entre transputers y puede transferir datos a una velocidad de 20 Mbits por segundo, con un protocolo (hand-shaking) de sincronización automática en cada dirección.

Un sólo enlace de comunicación del transputer es implementado por dos canales en Occam, uno en cada dirección. Todos los enlaces operan concurrentemente con el microprocesador por medio de un DMA. Cada transputer se puede conectar hasta con otros cuatro, y se pueden implementar redes con diferente topología y tamaño. La

transferencia de datos se realiza usando un protocolo simple de un byte reconocimiento, de tal forma que se garantiza la sincronización.

### 3.4.2- ARQUITECTURAS BASADAS EN TRANSPUTERS.

Los cuatro enlaces de comunicación del transputer permiten implementar diversas topologías de interconexión, como se muestra en la figura (3.8).

Muchos algoritmos pueden ser implementados eficientemente sobre una topología "pipeline" como vemos en la figura (3.8.a). Las topologías tipo árbol son deseables en procesos jerárquicos, como se muestra en la figura (3.8.b). Arreglos de dimensiones como el de la figura (3.8.c) son deseables para las aplicaciones donde tienen estructuras de datos matriciales, tales como en el procesamiento de imágenes.

Para lograr arquitecturas reconfigurables INMOS ha diseñado el circuito que es un conmutador de 32 x 32 que conecta cada uno de los enlaces seriales de transputers con los demás [13].

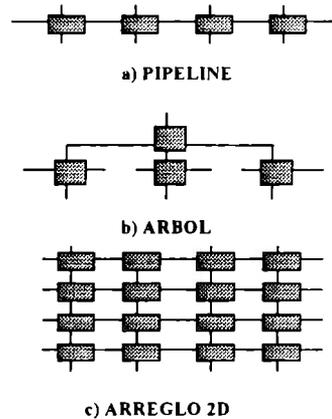


Figura (3.8) REDES DE TRANSPUTERS

Concluyendo, los transputers pueden ser empleados en una gran variedad de redes y modos de operación y son viables para diversas aplicaciones.

### **3.4.3.- COMUNICACIÓN ENTRE TRANSPUTERS.**

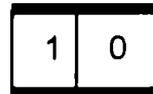
La comunicación entre procesos se realiza por medio de canales, que son la abstracción lógica del medio de comunicación. En Occam la comunicación es punto a punto, sincronizada y sin almacenar ( sin buffer ): la comunicación toma lugar cuando ambos procesos de entrada y salida se encuentran listos. Consecuentemente, el primer proceso que esté listo para la transmisión, espera a que el otro proceso lo esté a su vez.

Cuando un mensaje es enviado a través de un enlace, el procesador delega el trabajo a una interfaz, dejando al proceso en espera de una confirmación para continuar.

Para establecer la comunicación, se requiere de un protocolo para multiplexar los datos y controlar el flujo de información. Los mensajes son transmitidos como una secuencia de bytes, cada uno de los cuales requiere de un reconocimiento antes de transmitir el siguiente mensaje. Un byte de datos es transmitido iniciando con un bit de inicio y un "bit alto", seguidos por ocho bits de datos, y finalizando con un bit de paro. Un mensaje de reconocimiento es transmitido como un bit de inicio seguido por un bit de paro. El reconocimiento significa que el dato se recibió correctamente y que se está disponible para transmitir otro byte. El formato de la información se muestra en la figura (3.9).



**a) Byte de Datos**



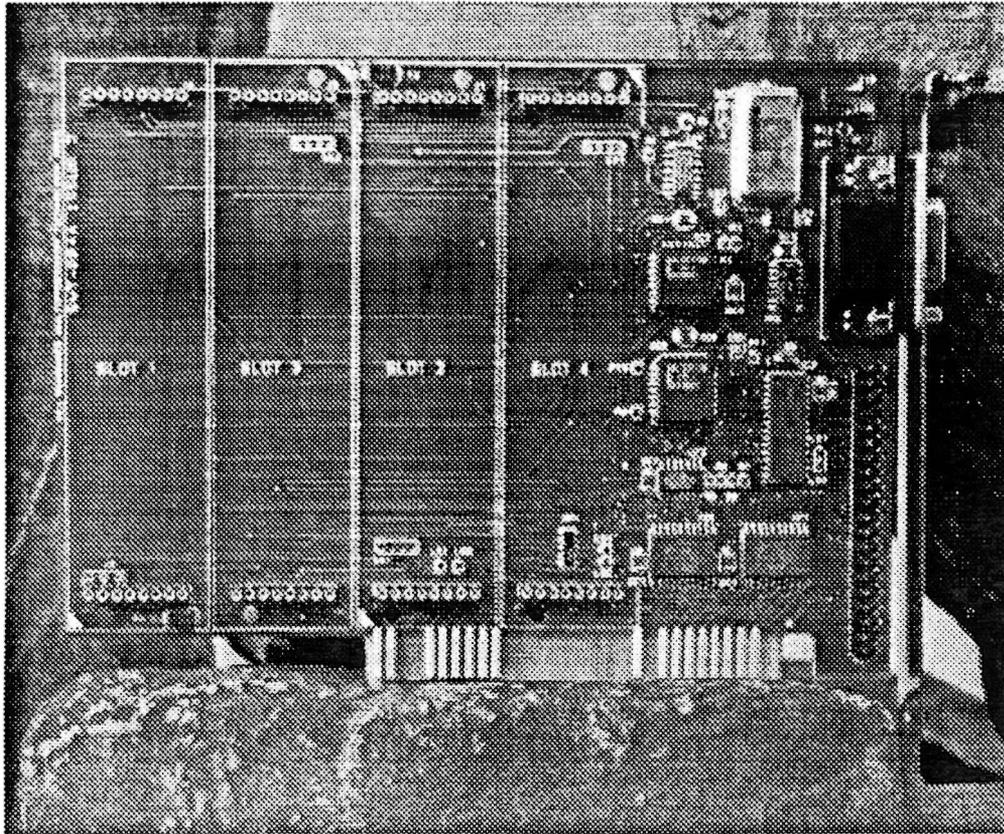
**b) Reconocimiento de mensaje**

**Figura (3.9) FORMATO DEL ENLACE DE DATOS**

### **3.5.- TARJETA SMT004A.**

Para el desarrollo de este proyecto, el controlador de la celda y la interfaz se implementaron sobre el equipo SMT004A fabricado por la compañía Sundance Multiprocessor Technology Ltd. La tarjeta principal (Motherboard) SMT004A (fotografía 3.1), es uno de los módulos básicos que permite conectar una arquitectura de hasta 4 Transputers a una computadora huésped, del tipo IBM PC o compatible. Los enlaces del 0 al 3 de cada transputer, están disponibles en un conector de 40 vías, mediante el que es posible alambrear la topología de la red de multiprocesadores.

La tarjeta principal SMT004 es una tarjeta que cuenta con cuatro ranuras (slot) para albergar hasta 4 TRAM ( TRANsputer Module ), dentro de una ranura de expansión de una computadora del tipo personal. Las dimensiones de esta tarjeta son de la mitad de la longitud estándar de una tarjeta para PC.



Fotografía [3.1] TARJETA SMT004A

Los TRAMs son pequeños módulos que cuentan con un transputer como procesador y algunos otros circuitos. El módulo puede variar en tamaño, pero todos cuentan con un conector de 16 señales para datos y señales de control y polarización. En la figura (3.10) se muestra un diagrama de un módulo TRAM.

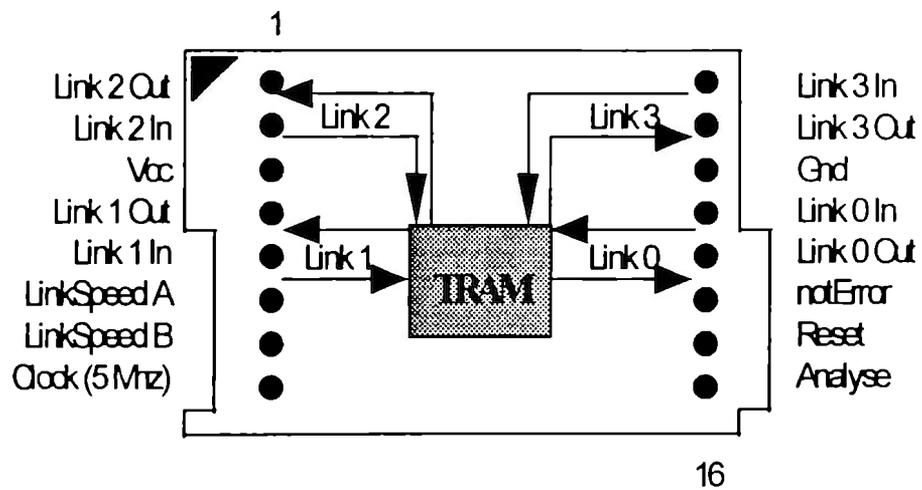


Figura (3.10) MÓDULO TRAM

El TRAM es un estándar para sistemas de procesamiento paralelo y distribuido, debido a la facilidad que brinda para usar un rango muy diverso de productos.

La SMT004 realiza la interfaz con la PC para inicializar el sistema (señal de reset), y por medio de las señales de análisis y error garantiza el correcto funcionamiento de una red de transputers. Los enlaces 0 al 3 de cada uno de los TRAMs están disponibles en un conector de 40 vías. Mediante este conector se puede configurar la tarjeta, utilizando para esto puentes ("jumpers") entre cada enlace (Apéndice F).

### **3.6.- COMUNICACIÓN ENTRE ELEMENTOS DE LA CELDA Y DISPOSITIVOS EXTERNOS**

La comunicación entre los diferentes elementos de la celda de manufactura y con dispositivos externos como puede ser un transputer, se realiza utilizando un puerto paralelo (interfaz Centronix), o bien por medio de un puerto serie (interfaz RS 232), dependiendo de lo que se encuentre disponible en las unidades de manejo de los robots.

#### **INTERFAZ CENTRONIX.**

Éste es originalmente el estándar paralelo para impresoras, establecido por Centronix Corporation. Se usa un cable con 8 bits de datos, y señales dedicadas para el control del flujo de la información. La transmisión paralela garantiza una velocidad de comunicación alta pero a distancias pequeñas de 1 a 2 metros.

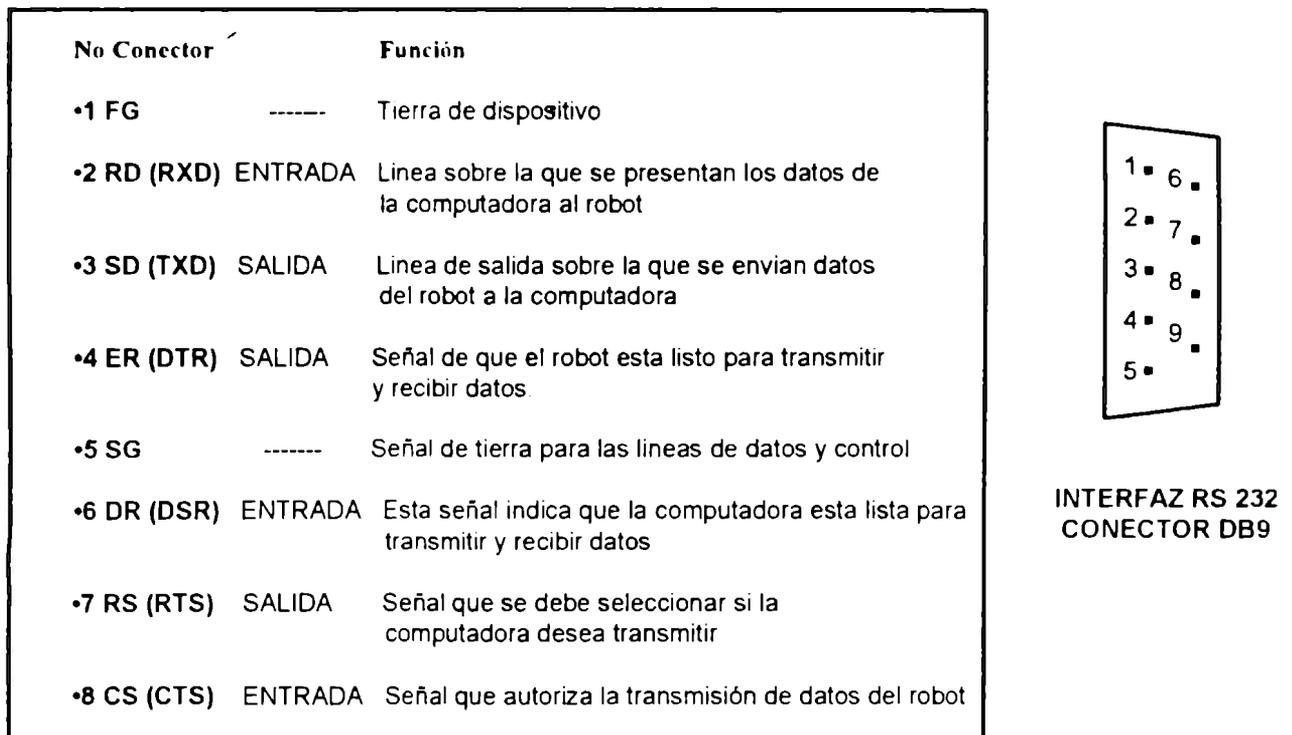
## **INTERFAZ RS 232.**

La interfaz RS 232 fue originalmente el estándar para los equipos de comunicaciones usados en líneas telefónicas y después fue adoptado como el estándar para la transmisión de datos seriales entre una computadora y su equipo periférico.

Debido a que los datos son enviados como una secuencia de bits a lo largo de un solo alambre, el tiempo que se requiere para la transmisión de información es mucho mayor que en las comunicaciones paralelas. La velocidad de comunicación (baud rate) se puede seleccionar en los robots así como en la computadora central, y es del orden de 10 Kbps.

En la comunicación serial se tiene la capacidad de realizar una transferencia de datos en forma bidireccional, y la distancia que se logra es mucho mayor que en una comunicación paralela, de 3 a 15 metros.

Para nuestra aplicación, dado que los elementos de la celda se encuentran físicamente en un piso (instalaciones del CIM), es necesario cubrir una distancia aproximada de 10 metros. Además, el controlador diseñado se basa en conmutadores y caminos múltiples (redundantes), por lo cual es mejor tener una comunicación con un número mínimo de señales de datos y control. Por las características planteadas anteriormente se eligió para el trabajo de esta tesis la interfaz serie RS 232. En la figura (3.11), se muestran la distribución de cada una de las señales en el conector DB 9 y la función de cada una de las señales que intervienen en la interfaz.



**Figura (3.11) Interfaz RS 232-C**

En el siguiente capítulo se describen las características principales de la interfaz, diseñada para que se puedan interconectar los diferentes elementos de la celda de manufactura con la red de transputers (tarjeta SMT004A).

## Capítulo 4

# DISEÑO, DESARROLLO Y FABRICACIÓN DE LA INTERFAZ

La arquitectura propuesta en [13] para el controlador de celdas flexibles de manufactura, es una máquina paralela y distribuida como se muestra en la figura (4.1). Esta arquitectura cuenta con múltiples procesadores (parte paralela, transputers) que nos garantizan procesamiento en tiempo real, reconfigurabilidad y tolerancia a fallas. La interfaz con los robots (parte distribuida), se realiza por medio del estándar RS-232C.

En este capítulo describiremos a detalle el diseño de esta arquitectura, mientras que en el siguiente describiremos el software que permite su operación.

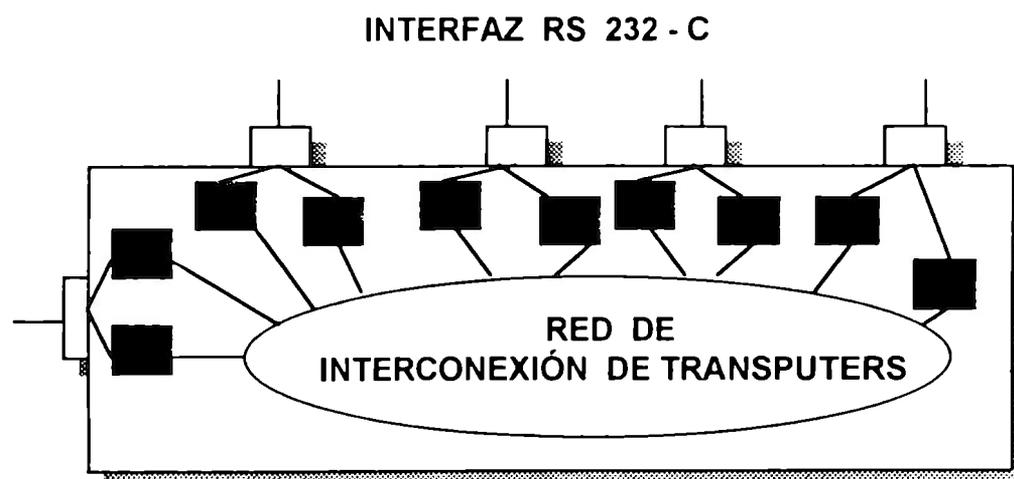


Figura (4.1) ARQUITECTURA DEL CONTROLADOR

La red de interconexión entre transputers es muy importante para el desempeño del controlador. Como se mencionó anteriormente existen dos tipos de topologías: estáticas y dinámicas.

En las redes estáticas, es difícil lograr la escalabilidad y reconfigurabilidad, debido a que sus enlaces son fijos y no pueden cambiar durante la ejecución del programa. Por eso escogimos usar redes dinámicas, donde los enlaces no son fijos sino conmutados, de forma que se conecta cualquier entrada con cualquier salida del conmutador. La ruta de enlace del conmutador se establece por software, ya sea por medio de mensajes de control o bien con el mismo mensaje de datos.

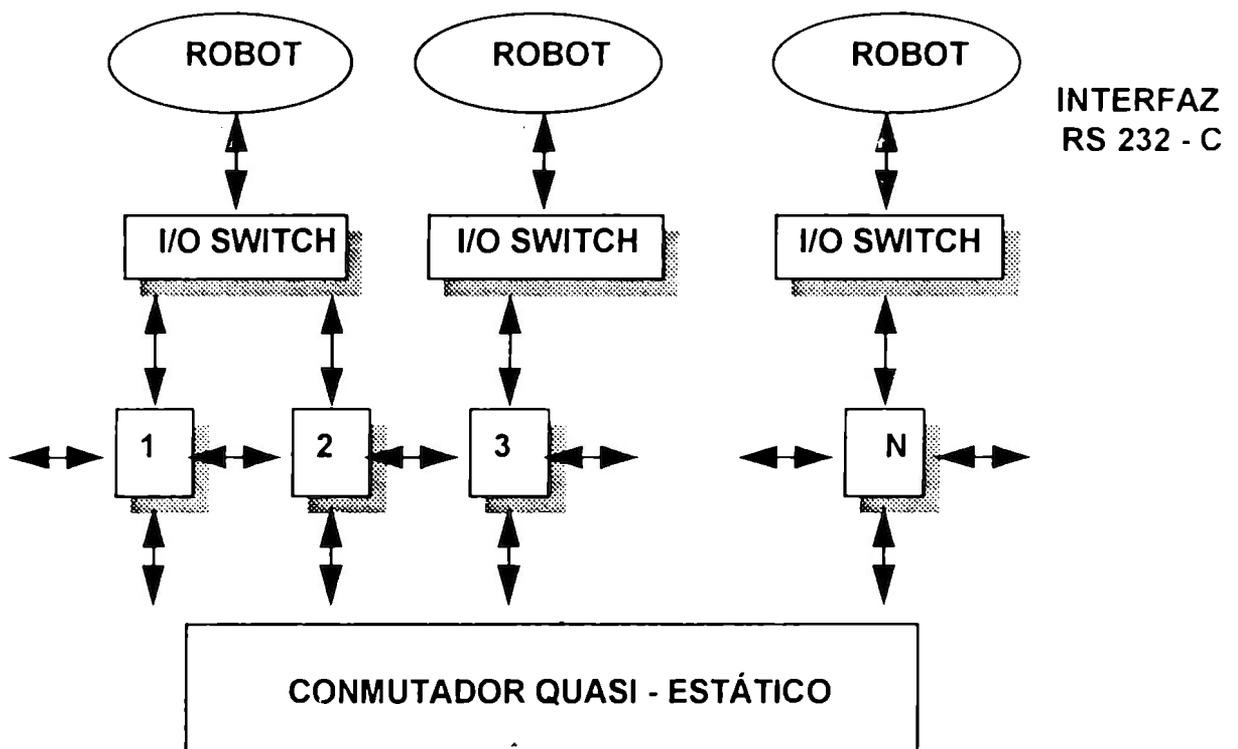


Figura (4.2) RED DINÁMICA

En nuestro controlador, la topología de transputers se puede ver como quasi - estática (figura 4.2) debido a que los conmutadores deben programarse en algún

momento de la ejecución para definir las conexiones, como se describe en el apéndice F.

Las redes dinámicas (y las quasi-estáticas, en menor medida) tienen dos ventajas principales sobre las estáticas para la aplicación deseada: permiten implementar la reconfigurabilidad, y el aislamiento y recuperación de fallas.

Una falla de un sistema computacional, es el estado erróneo de hardware o software causado por fallas de componentes, interferencia o diseño incorrecto. Un error es la manifestación de la falla. La tolerancia a fallas es la habilidad de un sistema para seguir funcionando correctamente aún en presencia de fallas.

La técnica básica de tolerancia a fallas es la redundancia, es decir la replicación del hardware, software y/o datos. Para el controlador propuesto se utiliza la técnica de redundancia en hardware activa, aprovechando las características de paralelismo de los transputers. Si un enlace de comunicación falla, se pueden tomar caminos alternos; si un procesador falla, otro toma a su cargo el trabajo que se estaba realizando.

Para incrementar la tolerancia a fallas del controlador se implementa un conmutador dinámico (I/O Switch en la figura 4.2) entre transputers y robots, el cual describiremos en la sección 4.2. El control de este conmutador es por software y se envía junto con los mismos datos.

La reconfigurabilidad de la red se logra al alambrar por hardware caminos alternos entre los transputers, por medio del conmutador quasi-estático descrito en el apéndice F. Al hacer uso de caminos redundantes entre los elementos de la celda, aumenta la tolerancia a fallas del controlador, ya que un enlace o un transputer puede ser aislado y evitado.

La redundancia también se implementa en las conexiones entre transputers y los elementos de la celda de manufactura, ya que se usa un circuito con conmutadores.

La función principal de este circuito, que describiremos en la siguiente sección, es la de servir como interfaz entre el controlador (la red de transputers) y los robots de la celda de manufactura. Al describir esta interfaz, también mostraremos como se solucionó el problema de interconectar dispositivos con diferente velocidad de comunicación como son los enlaces de transputers (20 Mbits/s) y la interfaz serial de los robots (9600 bauds).

#### 4.1.- DISEÑO DE LA INTERFAZ ENTRE TRANSPUTERS Y ROBOTS.

La interfaz que mostramos en esta sección es una arquitectura heterogénea, debido a que se interconectaron diferentes dispositivos procesadores. El elemento base que nos permitió trabajar con el transputer es el circuito adaptador de enlace C011 (apéndice A), el cual recibe en su entrada serial la información de uno de los enlaces de salida del transputer (a 20 Mbits por segundo) y la transforma en una señal de 8 bits, para que sea procesada digitalmente.

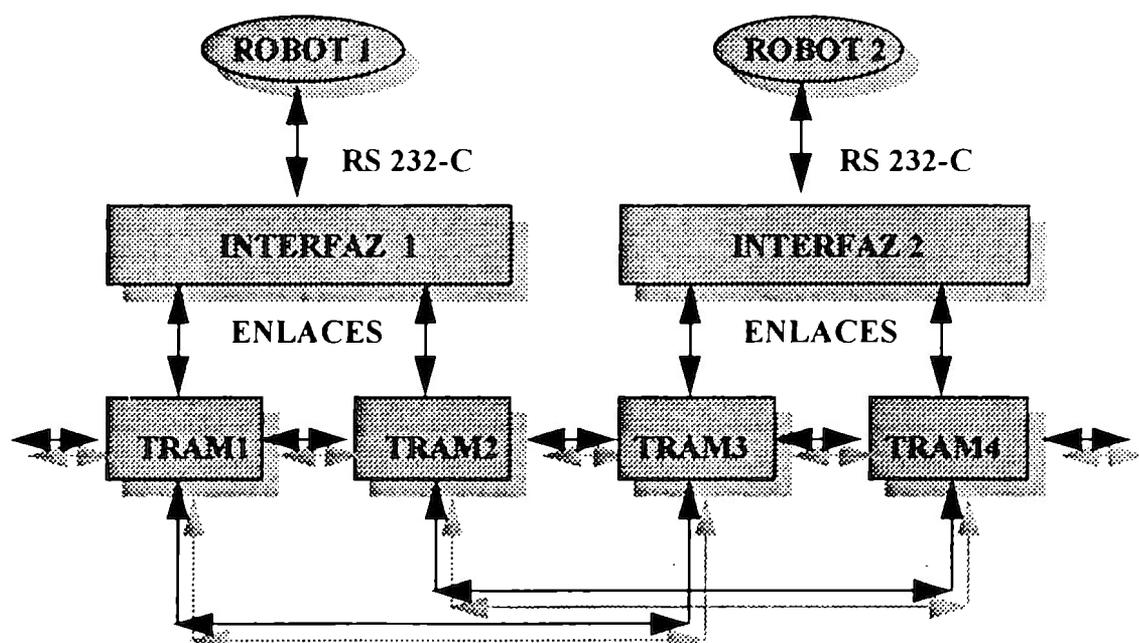
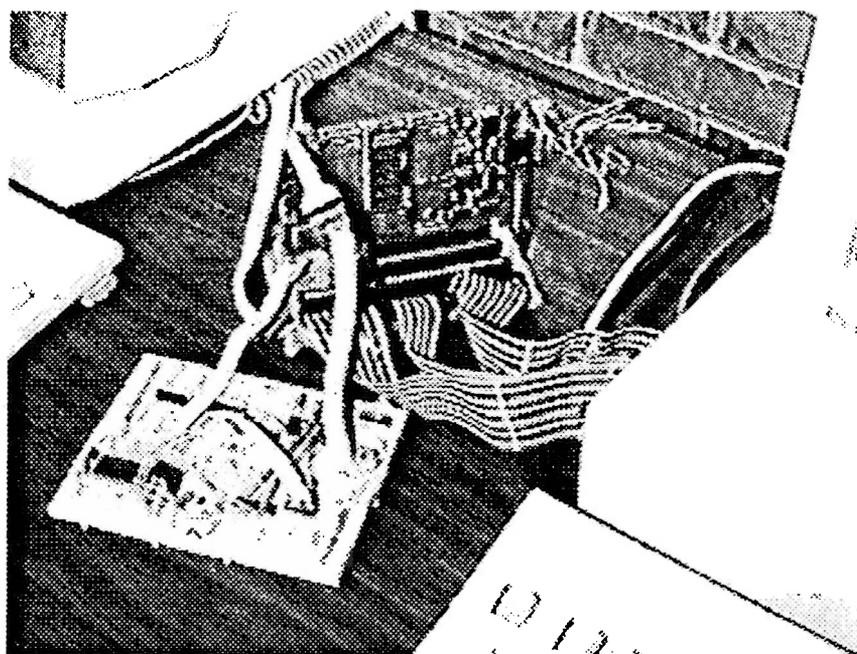


Figura (4.3) DIAGRAMA A BLOQUES DEL CONTROLADOR E INTERFACES

Esta interfaz puede interconectarse a cualquier módulo TRAM (sección 3.7), ya que se diseñó utilizando el estándar de INMOS. Un diagrama a bloques del controlador, junto con la interfaz diseñada, se muestra en la figura (4.3). En esta figura puede verse que nuestro circuito permite interconectar dos enlaces de comunicación a una interfaz serie RS 232, lo cual puede ser usado para implementar tolerancia a fallas.

El controlador de celdas flexibles de manufactura mostrado en la figura (4.3) es escalable, pues se pueden interconectar varios módulos y conmutadores a la red de transputers. Esta característica permite que varios transputers puedan controlar un mismo robot, por lo que se tienen caminos alternos que permitirán implantar un sistema tolerante a fallas.

Para la interfaz, se diseñó un circuito bidireccional que funciona tanto para entrada como salida de datos. Para llegar a este diseño, se estudiaron los diagramas de tiempo del circuito C011 y se desarrollaron módulos independientes de salida y de entrada, para posteriormente diseñar e implementar la interfaz final.



Fotografía (4.1) PROTOTIPO DEL CONTROLADOR

En los apéndices A, B y C se describen las características de los dispositivos usados en esta interfaz: el adaptador de canal C011, el microcontrolador 8751 y el circuito MAX 232 respectivamente. En la fotografía (4.1) pueden verse los circuitos electrónicos del controlador, los cuales detallaremos a continuación.

#### 4.1.1.- CIRCUITO MODO SALIDA

La interfaz de salida permite al transputer enviar información de control a dispositivos de la celda de manufactura equipados con interfaz RS 232. Se muestra su diagrama a bloques en la figura (4.4).

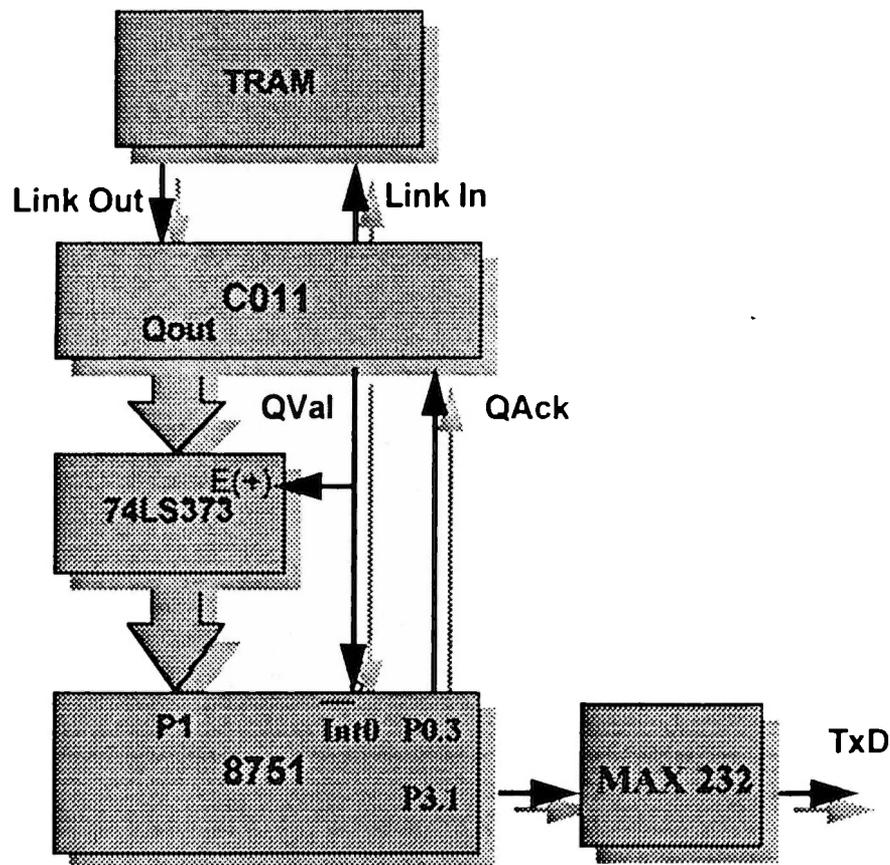


Figura (4.4) MODULO DE SALIDA

Cada módulo TRAM envía la información de la celda hacia los robots a través de uno de sus enlaces de salida (Link Out). Esta información es adquirida por el C011, quien procede entonces a poner en un nivel alto la señal QVal. Esta señal se utiliza para indicar al circuito 74LS373 que almacene la información en su registro. QVal después es negada para generar una solicitud de interrupción (Int0) al microcontrolador 8751.

El circuito 74LS373 es un registro de 8 bits con control de tercer estado. Este circuito es indispensable en nuestra aplicación ya que nos permitirá establecer una configuración de bus bidireccional. Por medio del control de la señal de tercer estado QVal, se puede indicar el sentido del flujo de información.

Al recibir el microcontrolador la interrupción Int 0, procede a leer la información que se encuentra en el 74LS373 y la coloca en su registro interno SBUF. Posteriormente, empieza a transmitir esta información en serie a una velocidad de 9600 bauds al circuito MAX 232, hasta que se activa la señal interna TI que indica el fin de la transmisión. El circuito MAX 232 recibe la información en niveles de voltaje TTL y la envía hacia el robot en los niveles de voltaje requeridos por el estándar RS 232.

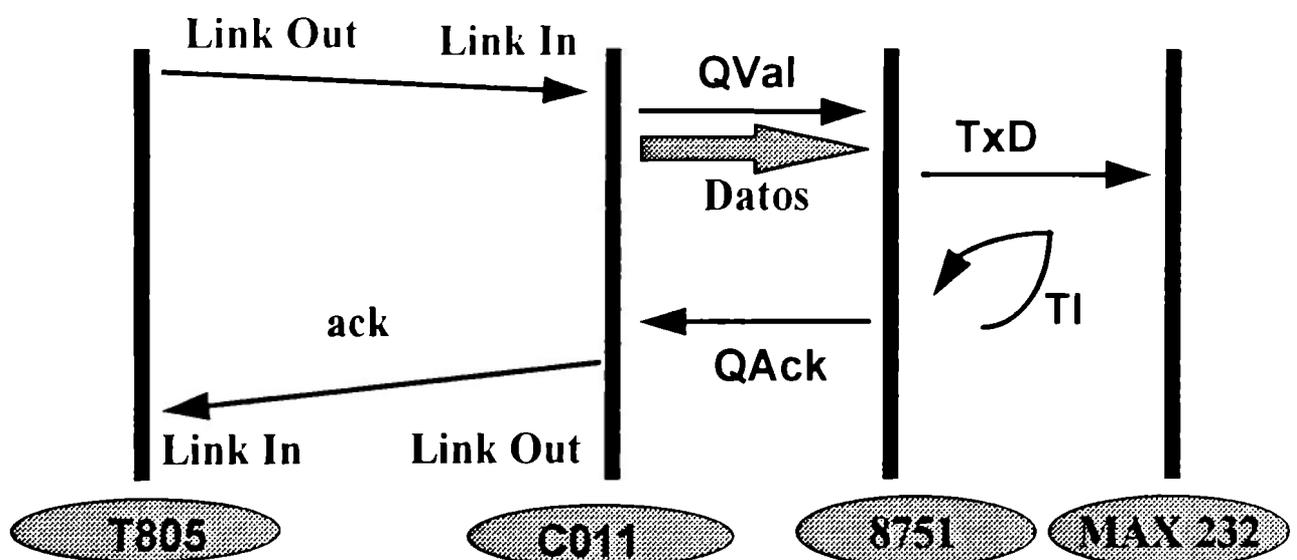


Figura (4.5) PROTOCOLO DE SALIDA DEL CONTROLADOR DE LA CFM.

#### 4.1.2.- CIRCUITO MODO ENTRADA

Este circuito permite que los elementos de la celda transmitan información al controlador por su interfaz RS 232. El diagrama a bloques del circuito de entrada se muestra en la figura (4.6).

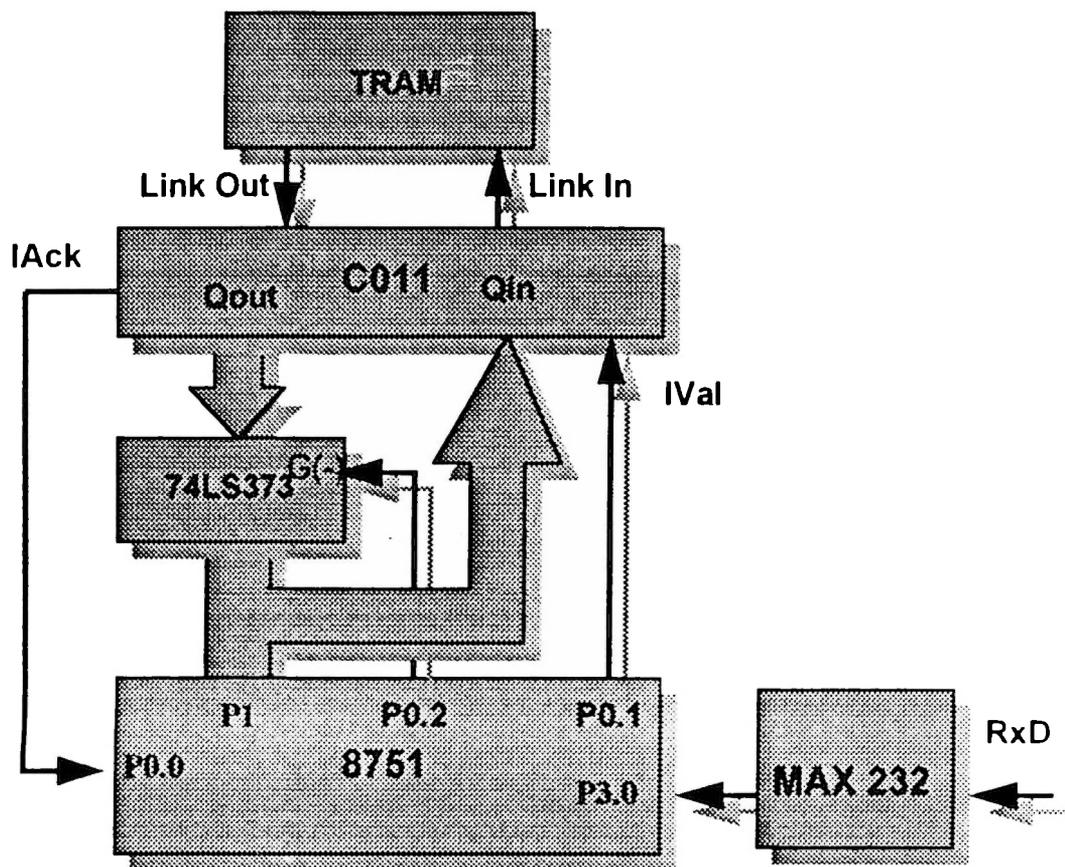


Figura (4.6) MODULO DE ENTRADA

El funcionamiento de la interfaz de entrada inicia cuando alguno de los robots envía información a través del MAX 232 hacia el controlador. Esta información se va almacenando en el registro interno SBUF del microcontrolador 8751. En el momento en que éste se llena, se genera una interrupción (RI) para iniciar la comunicación con el transputer. El microcontrolador genera entonces una señal de control G(-) a la salida del puerto P0.2, la cual coloca en estado de alta impedancia al circuito 74LS373. Ésto

permite que el flujo de información sea del puerto de salida del microcontrolador, hacia el registro de entrada del adaptador de enlace C011. Después de activar G(-), el microcontrolador genera la señal IVal en P0.1 para indicar al C011 que tiene un dato válido a la entrada de su registro paralelo. Esta señal permanece activa durante el tiempo en que el C011 transmite la información en forma serial hacia uno de los enlaces del TRAM.

Después de recibir la información, el módulo TRAM responde con una señal de Ack, lo que hace que el C011 responda a su vez con una señal de IAck al puerto P0.0 del microcontrolador.

Cuando el microcontrolador recibe la señal de IAck, pone en estado inactivo la señal de IVal, con lo cual se da por concluida la transferencia de información. Todo este proceso se muestra en la figura (4.7).

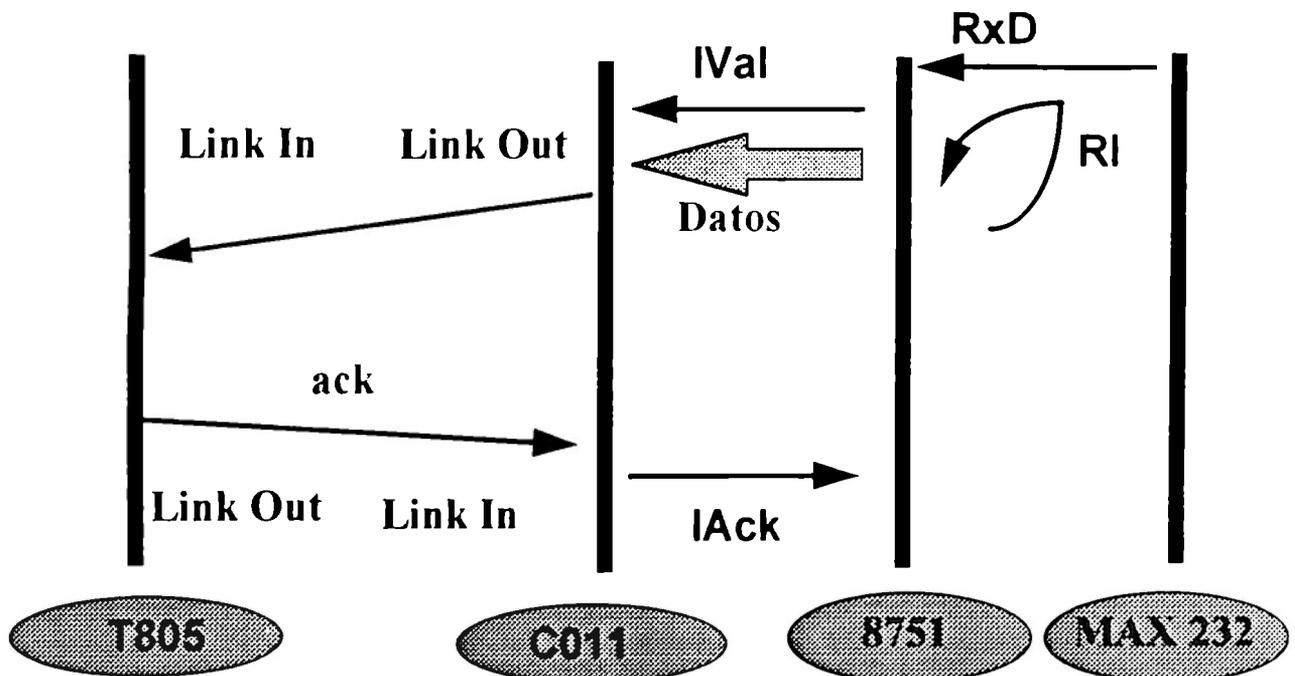


Figura (4.7) PROTOCOLO DE ENTRADA DEL CONTROLADOR DE LA CFM

### 4.1.3.- CIRCUITO MODO BIDIRECCIONAL

Para el diseño final de la interfaz se unieron los dos modos anteriormente descritos en un circuito bidireccional. En la figura (4.8) se muestra el diagrama a bloques así como las señales de control, indicando el sentido del flujo de la información. En esta figura se muestra el control para un módulo TRAM, y la comunicación hacia uno de los robots por medio del circuito MAX 232.

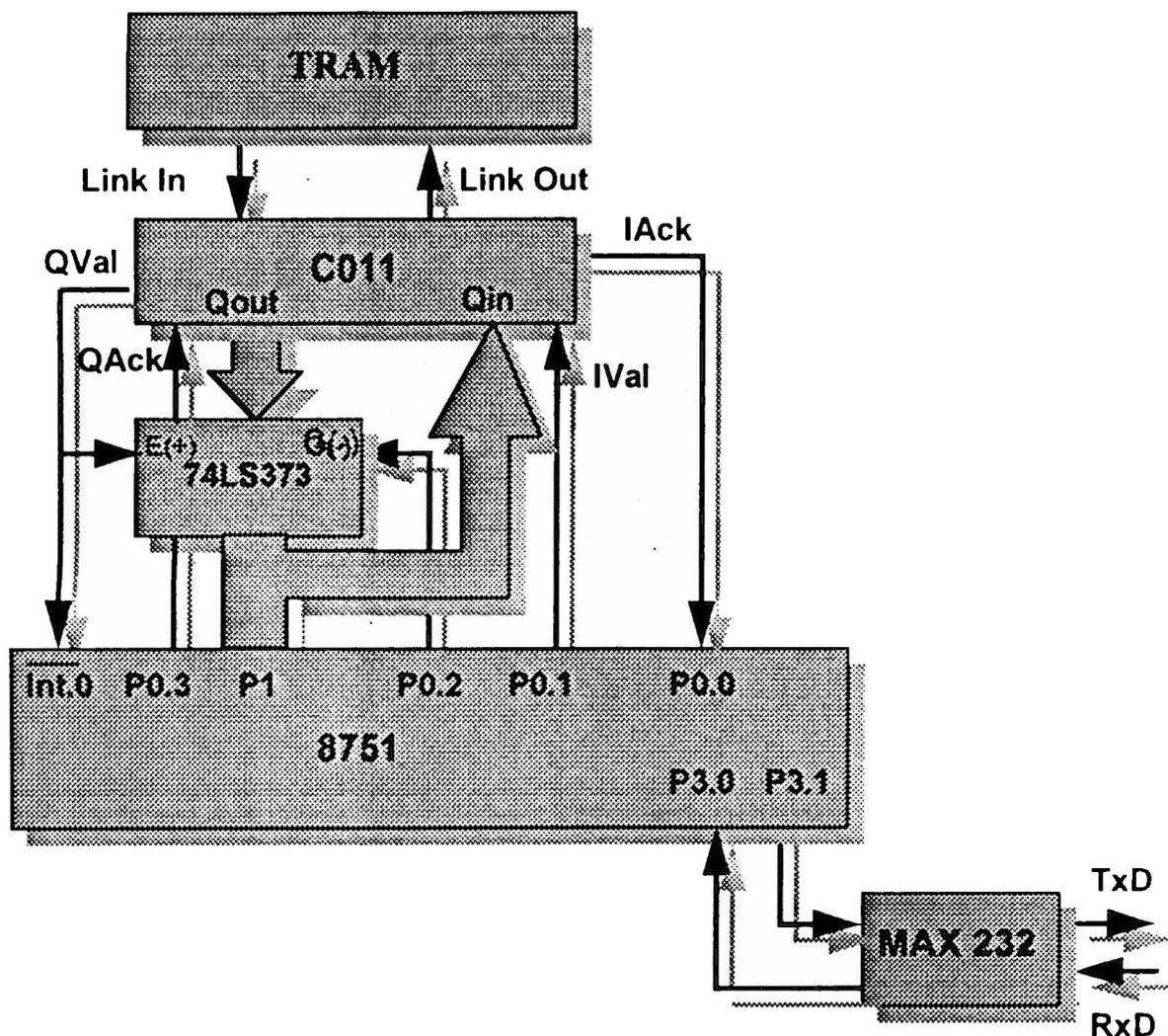


Figura (4.8) MODULO BIDIRECCIONAL

Las señales de control que intervienen en la comunicación hacia el robot son QVal, para indicar que existe un dato en el puerto de salida, y QAck, para indicar que el dato fue enviado hacia el robot. Cuando la información va hacia el controlador las señales que intervienen son IVal e IAck. El puerto P0 del microcontrolador es usado para albergar estas señales, mientras que el puerto P1 es usado para transmitir información en forma paralela.

El puerto P3 del microcontrolador es usado para comunicarse con el robot en forma serial. Los datos se envían a través de la línea P3.1 en niveles TTL y se interconecta a la entrada transmisora T2 in del MAX 232 (Apéndice C) para lograr en la salida T2 Out una señal dentro de los niveles RS 232. La recepción de información proveniente de los robots se recibe en la entrada R1 in del MAX 232, quien entrega una señal de salida R1 out en niveles TTL, requeridos para interconectarla a la terminal ("pin") receptora de datos del microcontrolador en el puerto P3.0.

Dadas las características del circuito bidireccional, el método de comunicación implementado es "half- duplex": la comunicación con un robot se puede realizar en cualquier dirección (entrada ó salida), pero no en ambas direcciones al mismo tiempo. Esto será suficiente para la operación normal de la celda pues los elementos de la celda usualmente están esperando comandos, y solo envían información al controlador como respuesta a uno de éstos.

## **4.2.- TOLERANCIA A FALLAS EN LA INTERFAZ**

Como se mencionó anteriormente, las redes dinámicas permiten implementar de forma natural la reconfigurabilidad necesaria en celdas flexibles de manufactura y nos ayudan al aislamiento y recuperación de fallas.

La reconfigurabilidad está presente en el controlador de dos formas:

- 1.- En la red quasi - estática de transputers.

2.- En la interfaz entre transputers y robots.

En el capítulo anterior (y Apéndice F) se habló de la red quasi-estática. En esta sección describiremos cómo se implementa la reconfigurabilidad en la interfaz entre transputers y robots.

Para lograr la reconfigurabilidad de la interfaz, se conectaron los enlaces de comunicación de los transputers a conmutadores 2X1. Se muestra un ejemplo, de configuración que se puede lograr con nuestra interfaz en la figura (4.3). En este ejemplo, si se tiene redundancia en hardware y también en software, se pueden tener los mismos procesos en los cuatro módulos TRAM para el control de dos robots diferentes.

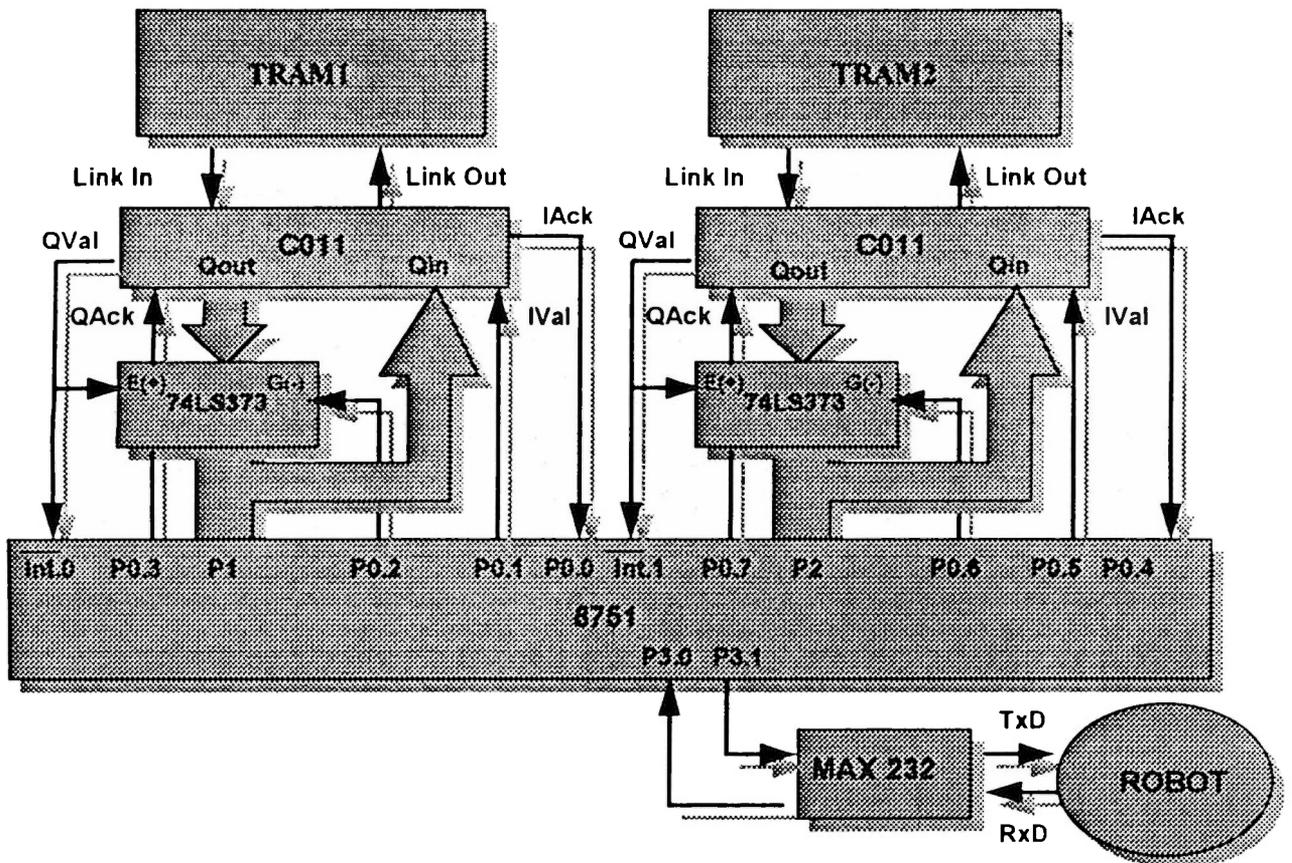


Figura (4.9) INTERFAZ CONMUTADA 2 X 1

Construimos un conmutador 2X1 colocando dos circuitos como los descritos en la sección anterior, sobre un mismo microcontrolador, como se muestra en la figura (4.9).

Para realizar el control de la interfaz conmutada se utilizan las señales generadas por los transputers, el adaptador de enlace C011, el microcontrolador 8751 y el circuito MAX 232, de la manera que se describe a continuación.

### LÍNEAS DE DATOS ( C011 -> Microcontrolador )

Para la comunicación de datos de y hacia los elementos de la celda, se utilizan los puertos paralelos P1 y P2 del microcontrolador 8751, los cuales se interconectan a través de circuitos C011, con la red de Transputers.

### LÍNEAS DE CONTROL ( C011 -> Microcontrolador )

Para generar las señales de control del protocolo de comunicación con el C011, se utiliza el puerto P0. Las señales P0.0 - P0.3 controlan la comunicación del puerto P1, mientras que las señales P0.4 - P0.7 controlan la comunicación del puerto P2. La tabla (4.1) muestra cada una de estas señales:

P0.0	IAck	Datos aceptados (entrada)	Puerto 1
P0.1	IVal	Datos válidos	Puerto 1
P0.2	G	Control de Flujo de Información	Puerto 1
P0.3	QAck	Datos aceptados (salida)	Puerto 1
P0.4	IAck	Datos aceptados (entrada)	Puerto 2
P0.5	IVal	Datos válidos	Puerto 2
P0.6	G	Control de Flujo de Información	Puerto 2
P0.7	QAck	Datos aceptados (salida)	Puerto 2

Tabla (4.1) Señales de Control del microcontrolador 8751

### **LÍNEAS DE DATOS ( Microcontrolador <-> Robots )**

Para realizar la transferencia de información con cada uno de los robots de la celda de manufactura se utiliza una interfaz serie con el estándar RS 232. El microcontrolador realiza la función de convertir en serie los datos que se reciben en paralelo de los transputers y viceversa. Para recibir o enviar los datos seriales de o hacia el robot, se utilizan las líneas P3.0 (RxD) y P3.1 (TxD) del puerto P3.

### **LÍNEAS DE CONTROL ( Microcontrolador - Interfaz RS 232 )**

Para realizar el control de la comunicación serial se utilizan las banderas internas del microcontrolador. Las banderas utilizadas son TI (Transmit Interrupt ), y RI ( Recibe Interrupt). La función de estas señales es la siguiente:

- TI se pone en un estado activo, cuando el registro interno SBUF ( Serial Buffer ) se encuentra vacío, al terminar de enviar el último bit de un carácter.
- RI se pone en un estado activo y genera una interrupción al microcontrolador, cuando SBUF está lleno, al recibir el último bit de un carácter.

### **4.3.- CONEXIÓN DE LA INTERFAZ AL CONTROLADOR**

El controlador diseñado es reconfigurable y escalable, pues se pueden interconectar varios módulos y conmutadores a la red de transputers con diferentes topologías, como se muestra en la figura (4.3).

Como se mencionó anteriormente, el diseño de la interfaz fue hecho teniendo en cuenta el estándar de los módulos TRAM. La interfaz se implementó en una tarjeta de circuito impreso que se inserta en una ranura de la PC, y los enlaces con el controlador

se realizan físicamente por medio de conductores entre la tarjeta SMT004 (sección 3.7) y la interfaz. Esta conexión se selecciona por medio de "jumpers" como se muestra en la figura (4.10).

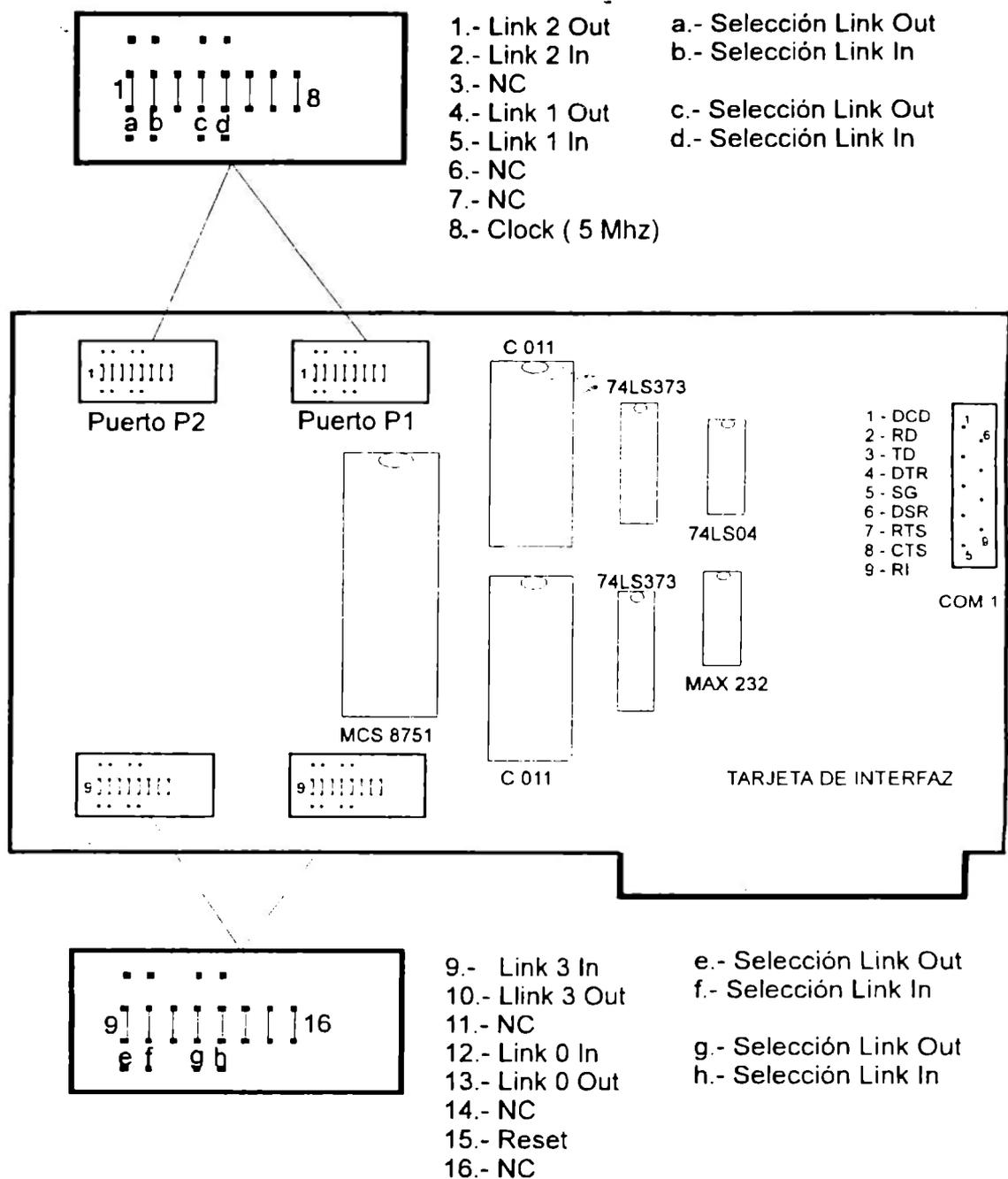
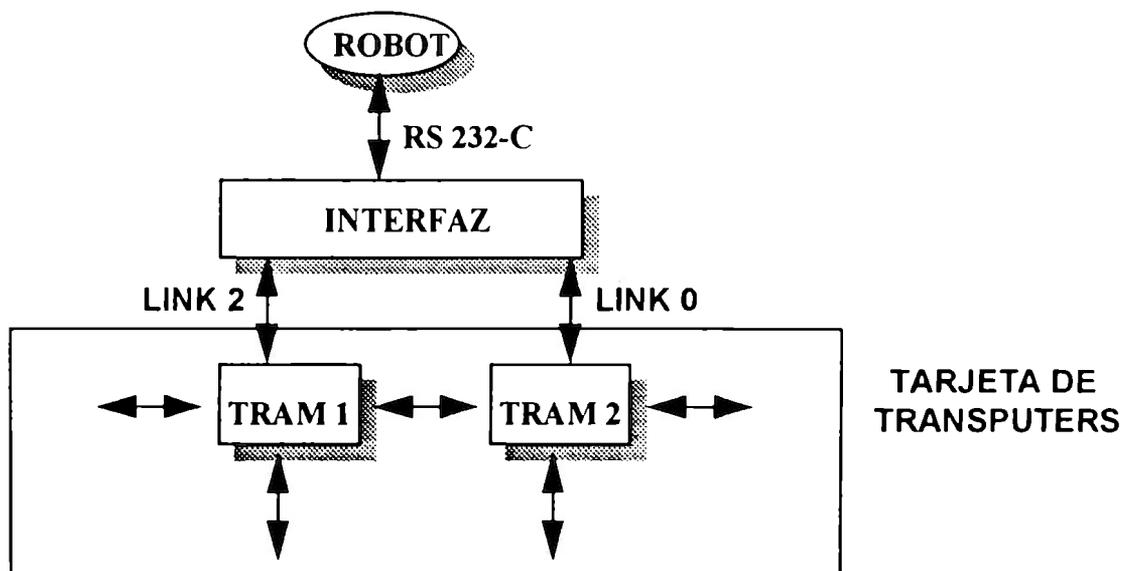


Figura (4.10) SELECCIÓN DEL ENLACE DE COMUNICACIÓN

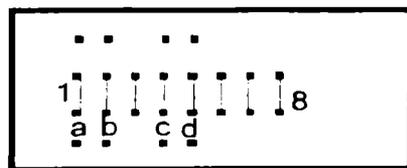
La interfaz cuenta con dos bancos de conectores de 16 "pines" cada uno. En los conectores (1-16) se encuentran las señales que provienen de la red de transputers: la señal de reset, las señal de reloj y cuatro enlaces de datos de entrada/salida para cada uno de los módulos TRAM. El otro banco de conectores tiene "jumpers" marcados con las letras (a) a la (h), que permiten mapear un enlace en cada uno de los puertos del microcontrolador.



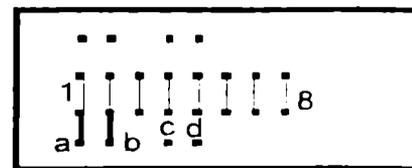
**Figura (4.11) INTERFAZ CONMUTADA CON CAMINOS ALTERNOS**

Al implementar este conmutador se logró que la interfaz sea escalable ya que se puede implementar un conmutador de n entradas a m salidas con módulos básicos de 2x1. Veamos por ejemplo en el caso de que se desee programar la interfaz para que el módulo TRAM 1 controle a un robot a través de su enlace 2, y al mismo tiempo exista un camino alternativo a través del enlace 0 del módulo TRAM 2 para el control del mismo robot, como se muestra en la figura (4.11). El usuario tendrá que realizar la conexión física de los módulos TRAM a la interfaz utilizando los conectores de cable plano con que cuenta la tarjeta diseñada.

En la tarjeta de interfaz se seleccionan los enlaces de comunicación por medio de los "jumpers" que se colocan entre el número de enlace de entrada / salida del TRAM y el enlace de entrada / salida de la tarjeta, como se muestra en la figura (4.12) para este ejemplo en particular.



Puerto P 2  
 Selección del enlace 0  
 Jumpers 12<->g, 13<->h



Puerto P 1  
 Selección del enlace 2  
 Jumpers 1<->a, 2<->b

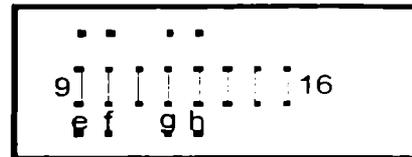
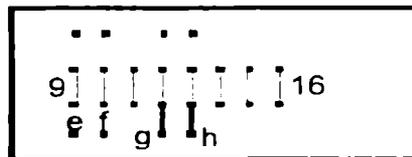


Figura (4.12) Programación de la Tarjeta de Interfaz ( Colocación de Jumpers)

## Capítulo 5

# PROGRAMACIÓN DE LA INTERFAZ

Para el control de la interfaz diseñada, se desarrollaron programas en el lenguaje ensamblador del microcontrolador 8751, los cuales se encuentran almacenados en la memoria EPROM del mismo.

Los programas diseñados se realizaron de acuerdo a las etapas presentadas anteriormente, para posteriormente realizar una integración de todos los módulos en la interfaz final.

Para realizar la sincronización del microcontrolador con la información que envía el circuito C011, en un principio se pensó en un esquema de programación basado en poleo (polling). En este esquema, el microcontrolador siempre está supervisando una bandera que le indica cuándo existe información válida sobre el puerto para ser leída y procesada. Cuando se trata de un sólo TRAM, este método de sincronización es adecuado, pero cuando se realiza la comunicación con 2 ó más TRAM éste ya no es eficiente, por lo que preferimos realizar la sincronización por medio de interrupciones (ver sección 4.2).

El número de interrupción identifica el transputer que desea comunicarse con el mundo exterior a través del microcontrolador. Cuando este último recibe la interrupción, establece la comunicación y deshabilita las demás interrupciones para evitar conflictos en la transferencia de información. Al finalizar la comunicación, se habilitan de nuevo

las interrupciones, permitiendo que cualquier transputer pueda solicitar una comunicación.

Los programas para el control de dos módulos TRAM y la interfaz serie, se encuentran en la memoria interna del microcontrolador. Estos programas, junto con el hardware, son los encargados de establecer el protocolo de comunicación (en la capa física), entre la red de transputers y los elementos de la celda de manufactura. Otros trabajos están siendo desarrollados [14], para implantar las capas superiores del controlador.

Para el desarrollo e implementación de estos programas se utilizaron diferentes herramientas de diseño, como son simuladores, ensambladores y ligadores de la compañía INTEL, mediante las cuales fue posible comprobar la correcta operación de cada uno de los módulos.

La programación sobre el microcontrolador se hizo en lenguaje ensamblador, usando un sistema de cómputo PC 486 como sistema de desarrollo. Después de comprobar el correcto funcionamiento de los programas, los grabamos en el banco de memoria interna (EPROM) del microcontrolador.

A nivel físico, se estudiaron las señales que el fabricante INMOS proporciona en el adaptador Serie - Paralelo C011 para lograr la sincronización correcta entre los transputers y el microcontrolador, por medio de una comunicación punto a punto.

La información que se envía de y hacia la red de transputers, permanece almacenada temporalmente en el microcontrolador hasta que se realiza la lectura de ésta por el C011 o los robots. Ésto permite resolver uno de los principales problemas al que nos enfrentamos: la diferencia de velocidad entre los enlaces de los transputers (Mb/s) y los canales de entrada de los robots (Kb/s). El manejo y generación de las señales de control necesarias para esta sincronización las proporciona el microcontrolador a través del puerto P0, como se describió en la sección 4.1.

Después de esta explicación general, a continuación describiremos detalladamente los programas de comunicación residentes en el microcontrolador y el transputer.

### 5. 1.- MÓDULO DE COMUNICACIÓN TRANSPUTER - ROBOT.

Para la comunicación entre la red de transputers y los robots, el circuito C011 de INMOS (Apéndice A) trabaja en modo salida (ver figura 4.5). El C011 recibe los datos provenientes del módulo TRAM en serie y los coloca en sus registros de salida en paralelo. El microcontrolador 8751 los lee por uno de sus puertos paralelos y los envía a la interfaz serie RS 232 (figura 5.1).

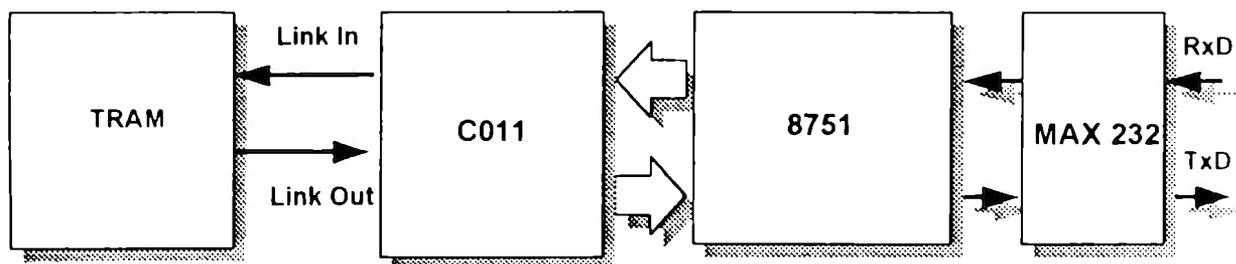


Figura (5.1) COMUNICACIÓN TRANSPUTER - ROBOT

El pseudo - código del protocolo de la capa física, entre el adaptador de enlace y el microcontrolador, se muestra en un diagrama a bloques en la figura (5.2). El listado de este programa se localiza en el Apéndice D.1.

Este programa se encuentra en la memoria EPROM del microcontrolador y se inicializa en el momento de "arrancar" el sistema. Las funciones que se ejecutan son las siguientes: el microcontrolador inicializa su operación y declara el uso de las interrupciones INT0 y Serie. En la etapa de inicialización, el microcontrolador declara la velocidad de comunicación a 9600 bits/seg (bauds) y la longitud del dato como 8 bits por caracter.

Para iniciar la operación de la interfaz, el microcontrolador envía a un nivel bajo las salidas del puerto P0, garantizando con esto que las señales QVal e IVal se encuentran en su estado normal de operación. La comunicación principia cuando se genera una interrupción externa (INT0), como se explicó en la sección 4.1.1

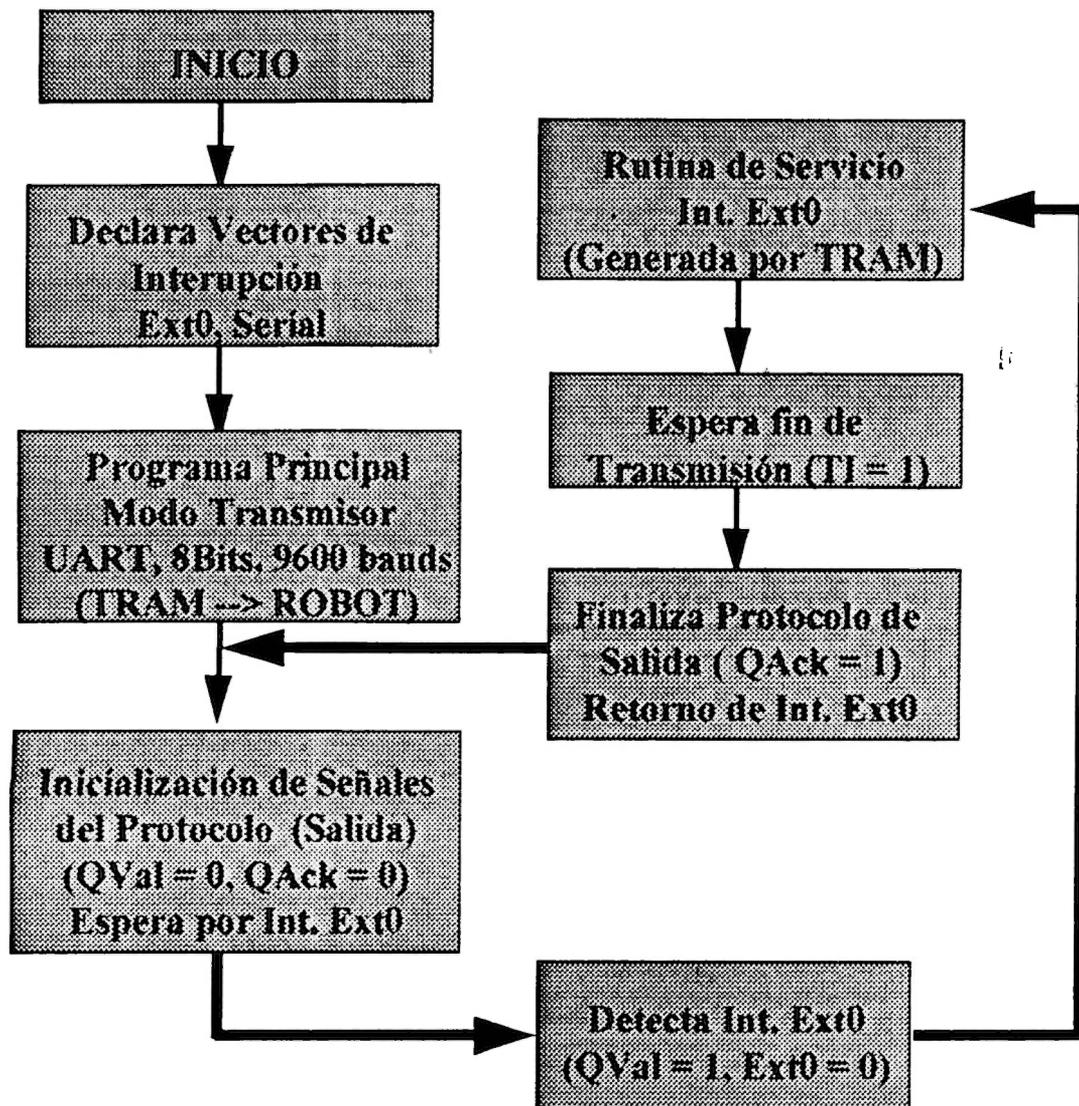


Figura (5.2) PSEUDO CÓDIGO DEL PROTOCOLO DE SALIDA

En el momento en que se detecta la interrupción INT0, el microcontrolador transfiere el control a la rutina de servicio de interrupción. Esta rutina se encarga de leer la información paralela del puerto del microcontrolador y la convierte en serie para ser

### 5.2.- MÓDULO DE COMUNICACIÓN ROBOT - TRANSPUTER

La transferencia de información de un robot hacia el controlador da inicio cuando la interfaz serie recibe una comunicación del robot, generando así una solicitud de interrupción al microcontrolador, quien a su vez genera las señales de control del protocolo de comunicación mostrado en la figura 4.7. Los datos que se reciben son almacenados temporalmente en el registro SBUF del microcontrolador, quien posteriormente los coloca en un puerto paralelo que se interconecta al bus de entrada del circuito C011. El adaptador de canal C011 recibe los datos en paralelo provenientes del microcontrolador, y los transforma en serie, para ser enviados a través un enlace al módulo TRAM.

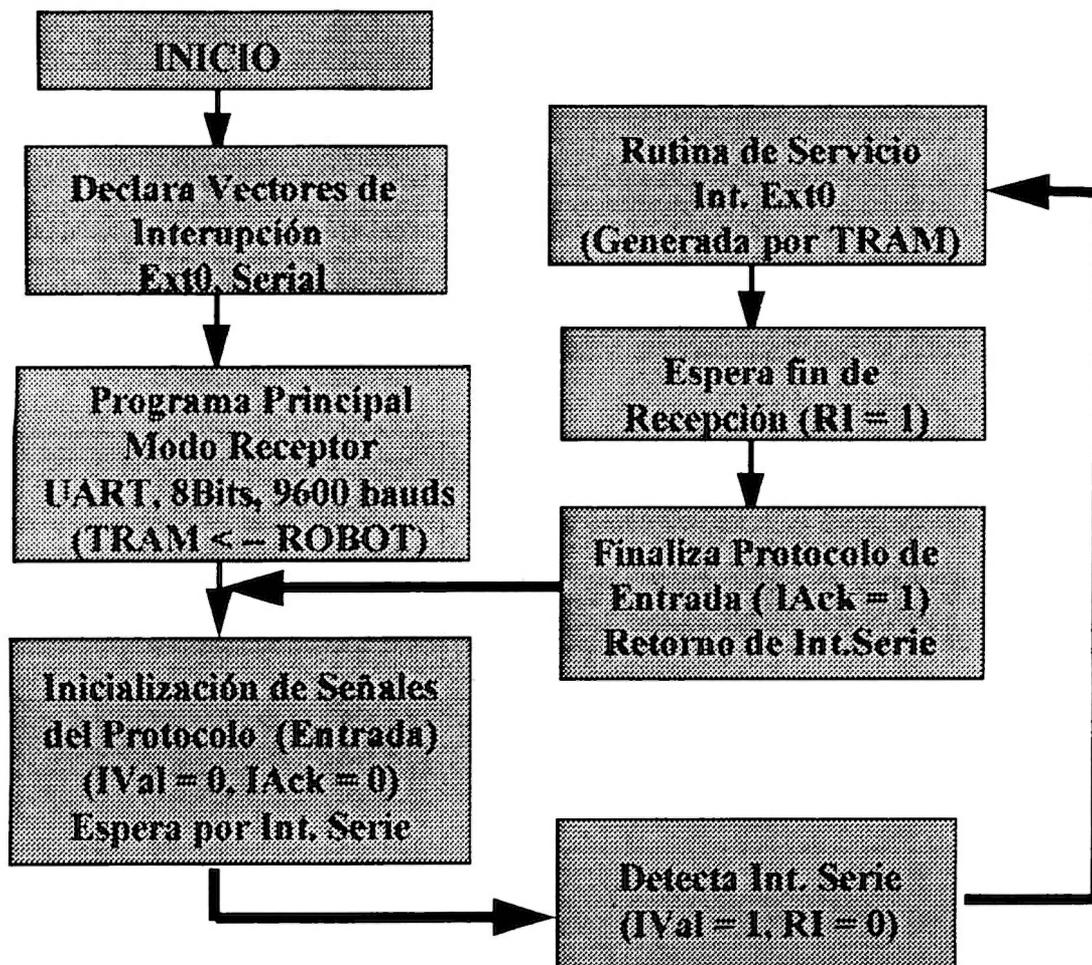


Figura (5.4) PSEUDO CÓDIGO DEL PROTOCOLO DE ENTRADA

El programa con el protocolo de la capa física entre el adaptador de enlace y el microcontrolador, se encuentra en el Apéndice D.2. En la figura 5.4 se muestra un diagrama a bloques del pseudo código.

Este programa se encuentra almacenado en la memoria interna (EPROM) del microcontrolador, y al momento de iniciar el sistema inicializa los vectores de interrupción (Interrupción serie) mediante los cuales se realiza la solicitud de comunicación entre el robot y el transputer. Para efectuar la transferencia de información se programa la interfaz serie con una velocidad de 9600 bauds y una longitud de carácter de 8 bits. El microcontrolador está listo para iniciar el protocolo de la figura 4.7, cuando coloca las salidas del puerto P0 a un nivel bajo.

Inicia la recepción de información cuando se detecta la señal de interrupción serie (RI); en ese momento el microcontrolador transfiere el control a la rutina de servicio de interrupción quien recibe la información a través de su registro serie y la transforma en paralelo para enviarla hacia uno de los puertos paralelos del microcontrolador. La comunicación finaliza cuando el transputer responde con una señal de reconocimiento (Iack), la cual se utiliza para regresar el control al programa principal donde se inicializan nuevamente las señales que intervienen en el protocolo.

De la misma manera como se hizo con el protocolo de salida, se hizo un programa para probar el envío de información del robot al transputer. Este programa, descrito a detalle en [23], inicia su operación ejecutando un proceso en la capa de presentación, otro en la capa de enlace de datos y uno más en la capa física.

Cuando la capa física recibe información serial del robot, el microcontrolador activa la señal IVa1 para que las capa de enlace de datos, y la de aplicación reciban información en un transputer del controlador. Posteriormente, se envía de las capas superiores a la capa física la señal de reconocimiento (Ack) para indicar que la transferencia se realizó correctamente en cada una de las capas de comunicación.

Finalmente, la capa física entre el transputer y el controlador genera la señal (IAck) para dar término a la transferencia de información entre el controlador y la red de transputers. Se muestra en la figura (5.5), un resumen de los procesos que intervienen en la comunicación robot - transputer. Puede consultar el código de este protocolo en [23].

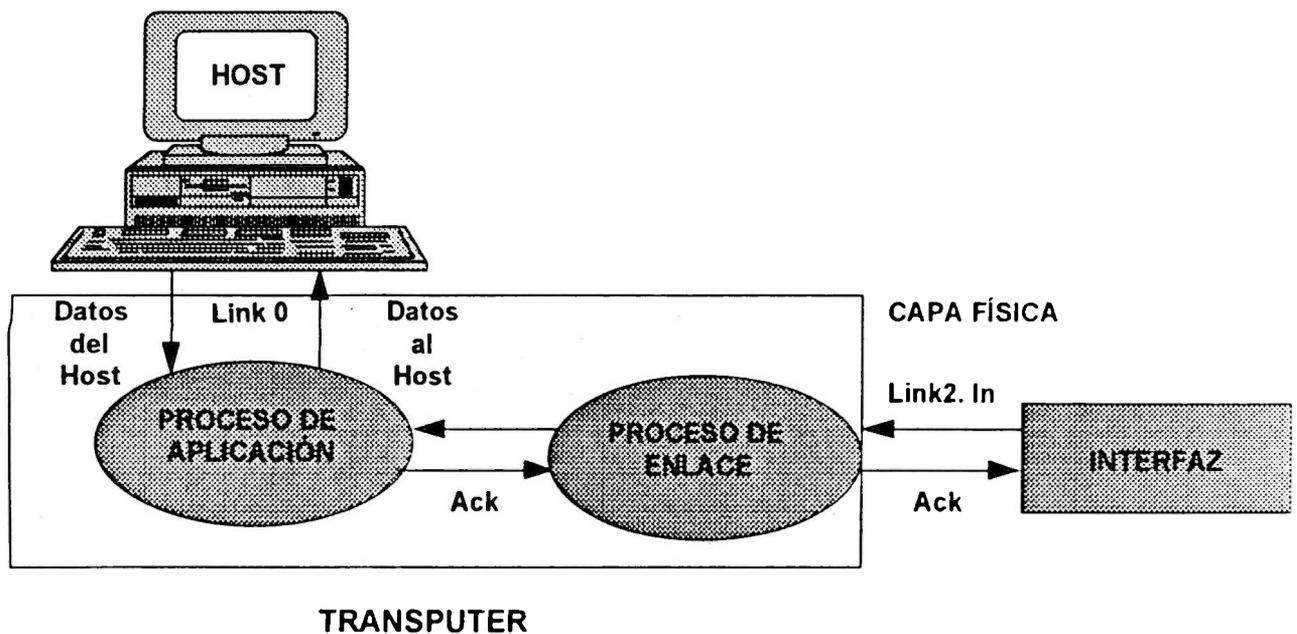


Figura (5.5) PROCESOS GENERADOS POR EL TRANSPUTER PARA EL PROTOCOLO DE ENTRADA

### 5.3.- COMUNICACIÓN BIDIRECCIONAL TRANSPUTER <->ROBOT

Para realizar la comunicación entre el transputer y el robot en modo bidireccional, se implementaron en forma conjunta los modos de salida y entrada descritos anteriormente. El programa se encuentra almacenado en la memoria EPROM del microcontrolador y puede consultarse en el Apéndice D.3. El diagrama a bloques de este módulo se muestra en la figura (5.6). Las rutinas de servicio de interrupción

ocupan localidades diferentes en el mapa de memoria y se direccionan por medio del vector de interrupciones.

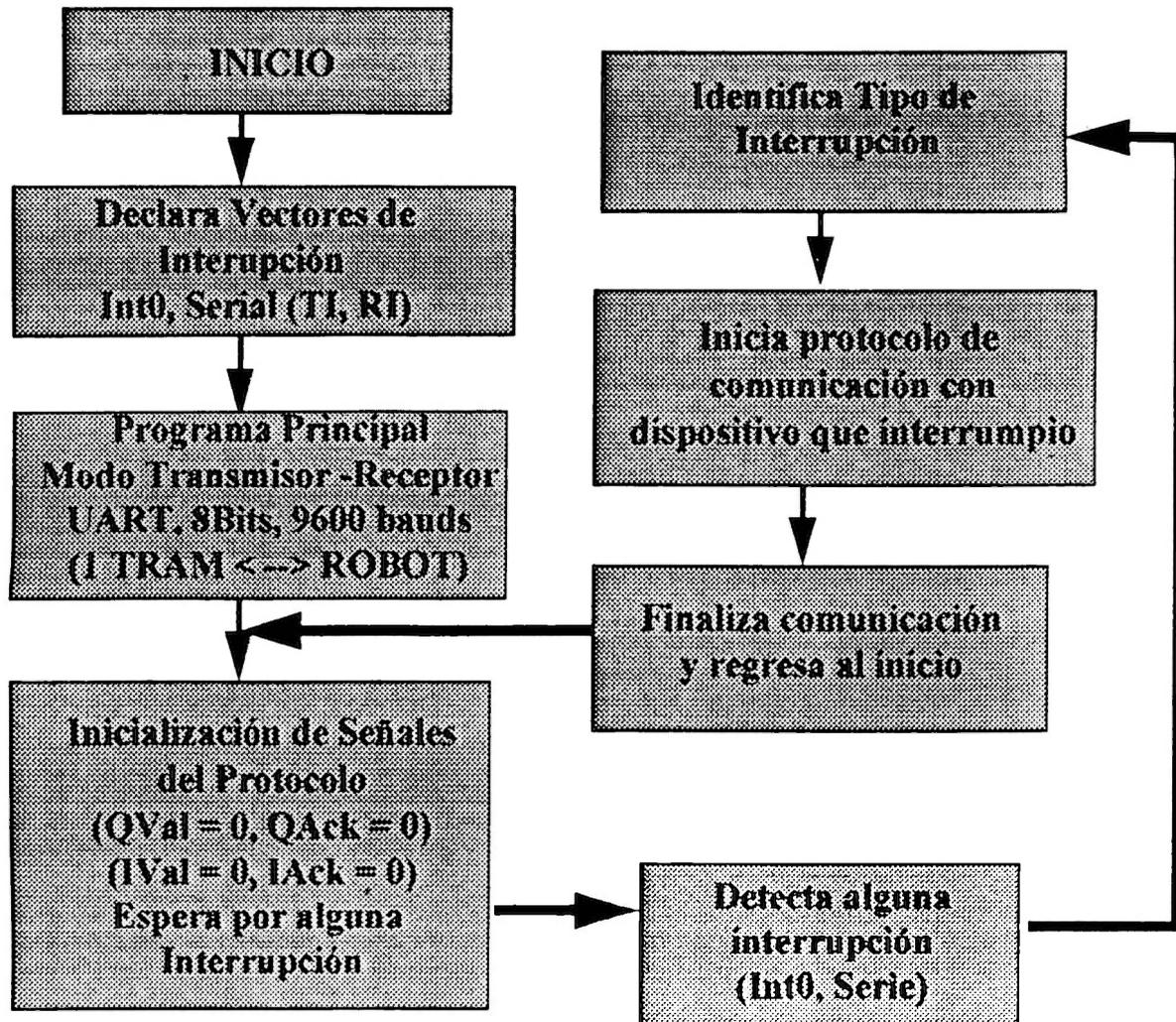
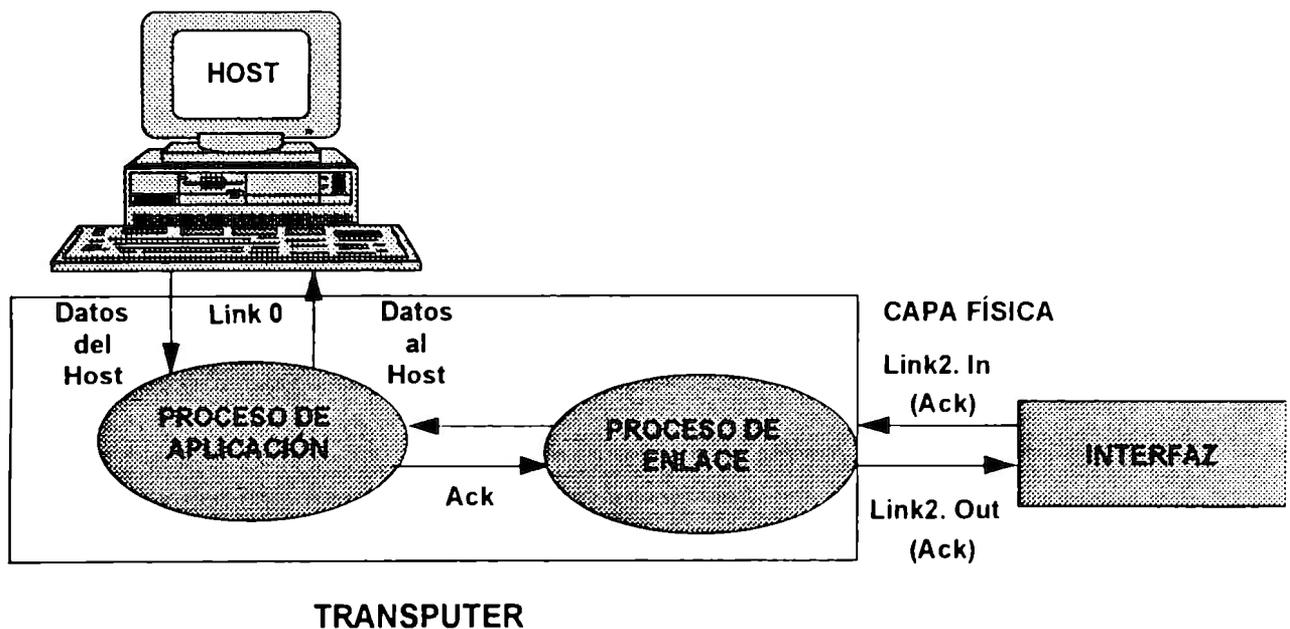


Figura (5.6) PSEUDO CÓDIGO DEL PROTOCOLO DE BIDIRECCIONAL

El funcionamiento de la comunicación bidireccional principia cuando el microcontrolador declara los vectores de interrupción (INT0) y serie (TI, RI), con parámetros de comunicación: half duplex, de 8 bits, velocidad de 9600 bauds. La comunicación es half duplex debido a que se realiza en ambos sentidos (transmisor y receptor) pero no al mismo tiempo.

En el diagrama a bloques de la figura (5.6), se muestra un bloque en el que se espera por alguna de las interrupciones para iniciar la comunicación, ya sea del controlador (INT0) o del Robot (RI). Si la interrupción es INT0, el microcontrolador ejecuta la rutina de servicio de salida descrita en la sección 5.1 y al terminar de ejecutar esta rutina de servicio regresa nuevamente al programa principal para inicializar nuevamente las señales del protocolo.

Si estando en el programa principal se genera una interrupción proveniente del robot, se identifica que la interrupción es (RI). En ese caso, el microcontrolador ejecuta la rutina de servicio de entrada descrita en la sección 5.2 y al finalizar, regresa al programa principal.



**Figura (5.7) PROCESOS GENERADOS POR EL TRANSPUTER PARA EL PROTOCOLO BIDIRECCIONAL**

Para probar esta rutina se utilizó un programa de prueba para el envío y recepción de información del transputer al robot y del robot al transputer, descrito a detalle en [23]. Este programa principia su operación iniciando en paralelo un proceso en la capa

de presentación, otro en la capa de enlace de datos y uno más en la capa física. Cada uno de estos procesos realiza las operaciones de transmisión y recepción de información que se describieron en la sección 5.1 y 5.2, en la figura (5.7) se muestra un diagrama a bloques de los procesos utilizados.

#### 5.4.- IMPLANTACIÓN DE LA TOLERANCIA A FALLAS

Como se recordará de la descripción de la interfaz en la sección 4.2, en nuestro diseño se implementa una redundancia de caminos de la celda a cada robot como se muestra en la figura (4.3), que se repite a continuación. En esta sección describimos el software que permite utilizar esta redundancia en la interfaz.

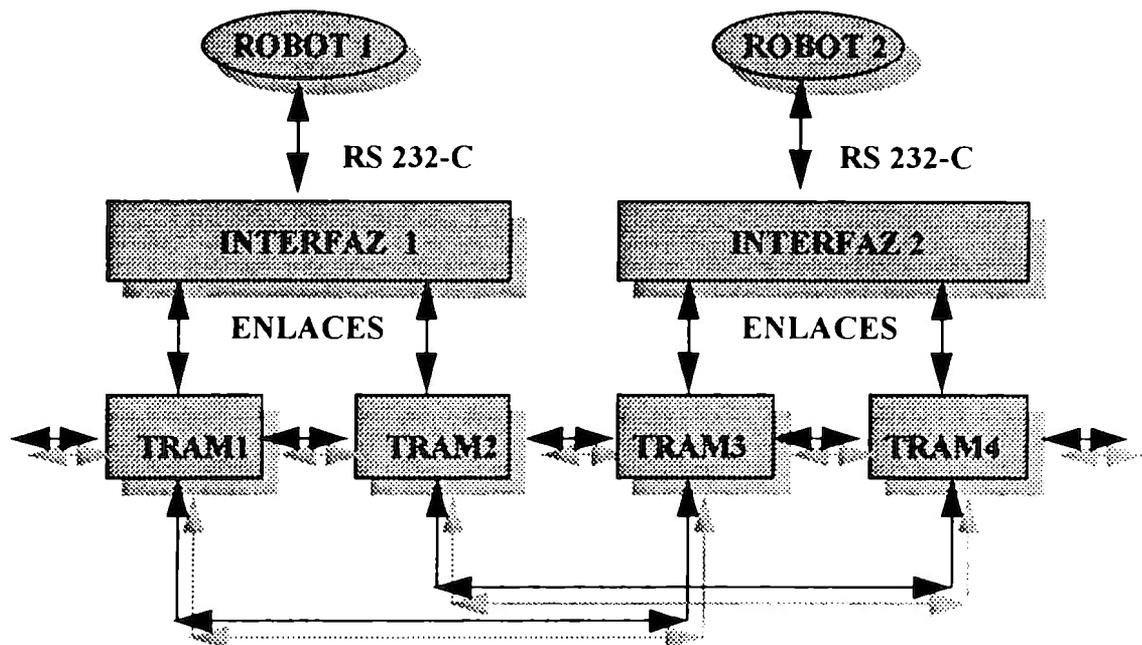


Figura (5.8) DIAGRAMA A BLOQUES DEL CONTROLADOR E INTERFACES

La comunicación de los módulos TRAM con la interfaz es asíncrona, es decir que puede presentarse en cualquier instante, por lo que la interfaz tiene que sincronizarse para establecer una correcta transferencia de información. Esta sincronización se logra

utilizando las señales de interrupción externa (una para cada TRAM) con que cuenta el microcontrolador ( INT0, INT1 ).

El método de selección se realiza con ayuda de las interrupciones (INT0, INT1) generadas por los transputers a través de los circuitos C011 (sección 4.11), de tal forma que solamente uno de ellos tendrá bajo su control a un robot. Esta selección permanecerá durante toda la operación de la CFM a menos que se requiera configurar el sistema, o se detecte una falla en los enlaces de comunicación de los transputers, y se decida por lo tanto utilizar un camino (ó transputer) alterno.

La exclusión mutua se realiza a través de una interrupción. El numero de interrupción determina el módulo TRAM que hizo la solicitud de comunicación. Este enlace permanecerá activo hasta que finalice la transferencia de información entre Transputer y Robot.

El programa desarrollado se presenta en el diagrama a bloques de la figura (5.9). Cada uno de los bloques representa en pseudo-código la función que realiza, y que describiremos a continuación. El listado del programa puede consultarse en el Apéndice D.4.

Primeramente, se inicializan los vectores de interrupciones externas (INT0, INT1), para cada TRAM y el vector de la interrupción serie para la comunicación con el robot (TI, RI) con 8 bits por carácter a una velocidad de 9600 bauds.

En el momento en que algún módulo TRAM o el robot requieren realizar una comunicación, se genera una interrupción. El microcontrolador deshabilita entonces las interrupciones para realizar, una exclusión mutua que deja el control de la interfaz a uno de los dispositivos.

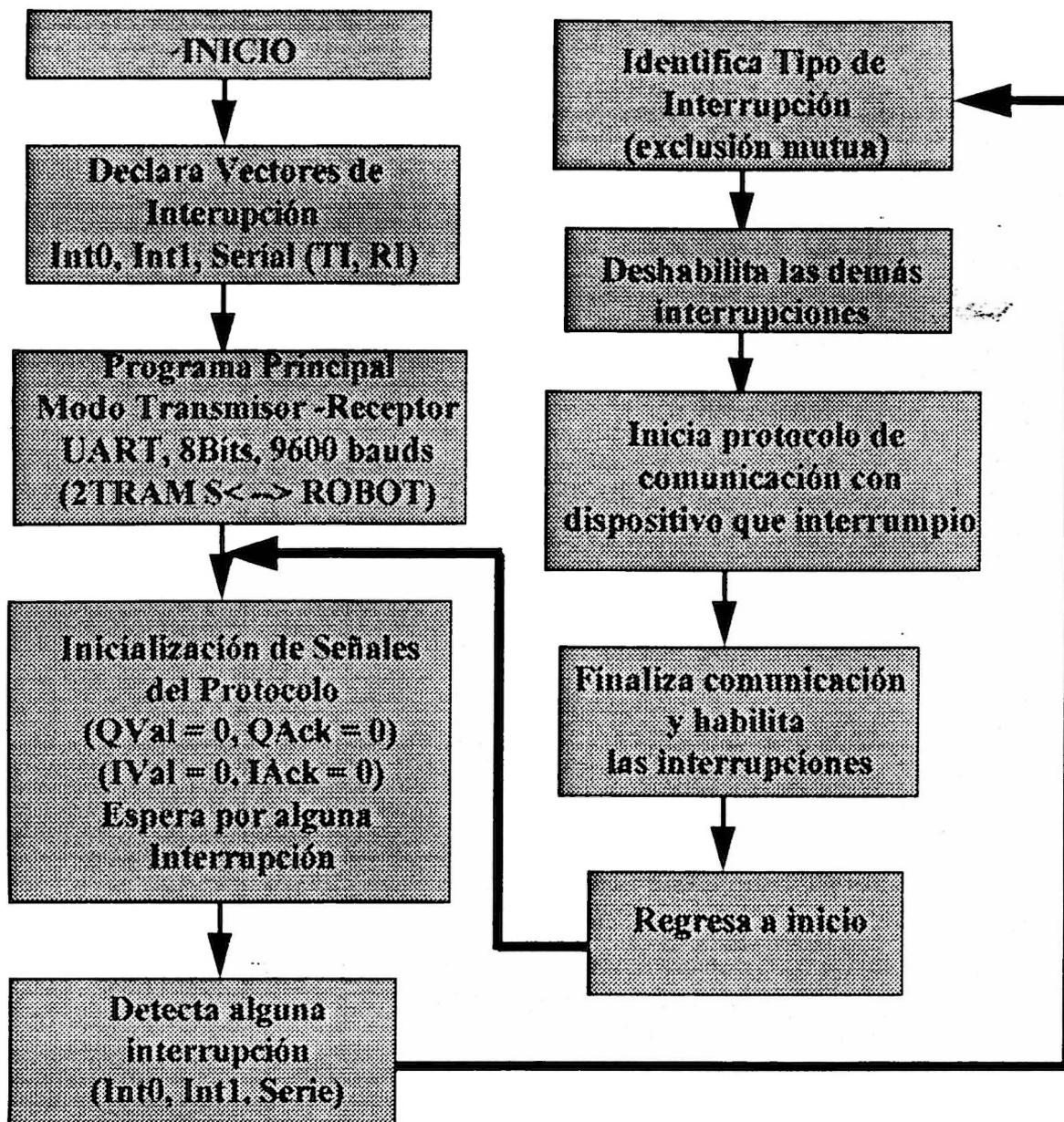


Figura (5.9) PSEUDO CÓDIGO DEL PROTOCOLO DE COMUNICACIÓN BIDIRECCIONAL PARA DOS MÓDULOS TRAMS.

El numero de interrupción identifica el TRAM y al mismo tiempo el puerto del microcontrolador por el cual se llevara a cabo la comunicación, es decir, INTO corresponde al puerto P1 e INT1 al puerto P2.

Si el robot que esté conectado a la interfaz inicia la comunicación, ésta se llevará con el modulo TRAM que estableció el control anteriormente. Al principio de la operación del controlador se inicializa la comunicación para que esta ocurra por "default" a través del puerto P1 del microcontrolador.

Como la comunicación que se realiza es de entrada y salida, una vez que un módulo TRAM toma el control de la interfaz, se realiza la transferencia de información por medio de los puertos paralelos P1 ó P2 y el puerto serie P3.0 (RxD) y P3.1 (TxD). Al finalizar la transferencia de información las interrupciones se habilitan, teniendo así la posibilidad de realizar una nueva comunicación.

## Capítulo 6

# PRUEBAS Y RESULTADOS

Durante el desarrollo de esta tesis se realizaron pruebas a cada uno de los circuitos que integran al controlador diseñado. Para realizar dichas pruebas se implantaron diversos circuitos, con el propósito de probar las señales requeridas para establecer un protocolo de comunicación entre los diferentes elementos del controlador.

En este capítulo describiremos a continuación los circuitos y pruebas implementados.

### **6.1.- CIRCUITO PROTOTIPO (PROTO - BOARD) <-> SIMULADOR**

Durante las etapas de diseño del controlador, se implementó en "proto - board" el prototipo de los módulos de salida y entrada descritos en el capítulo anterior. Este prototipo tiene la capacidad de controlar un robot por medio de un enlace de un transputer y un circuito MAX 232. Para comprobar el funcionamiento de este circuito antes de conectarlo al robot, se utilizó un equipo de cómputo del tipo IBM PC, como se muestra en la figura [6.1].

El equipo de cómputo auxiliar simula al robot enviando una serie de mensajes de inicialización hacia el controlador, esperando a que éste posteriormente envíe comandos para iniciar una secuencia de operación. La computadora auxiliar simula solo el protocolo de comunicación del robot, no su comportamiento funcional. Un simulador funcional del robot está siendo implementado por otro equipo de trabajo del proyecto [14].

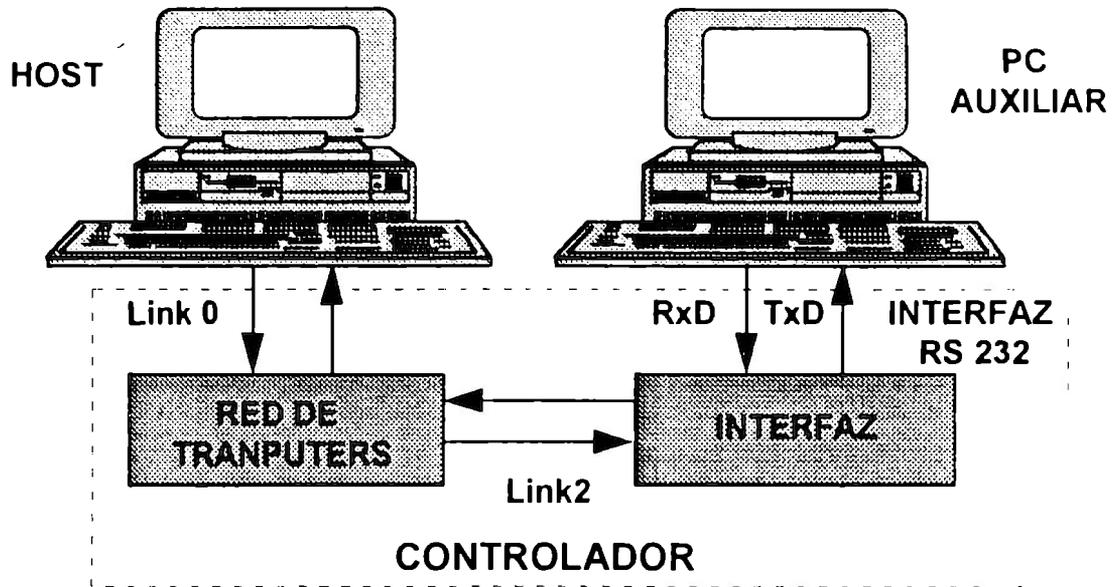
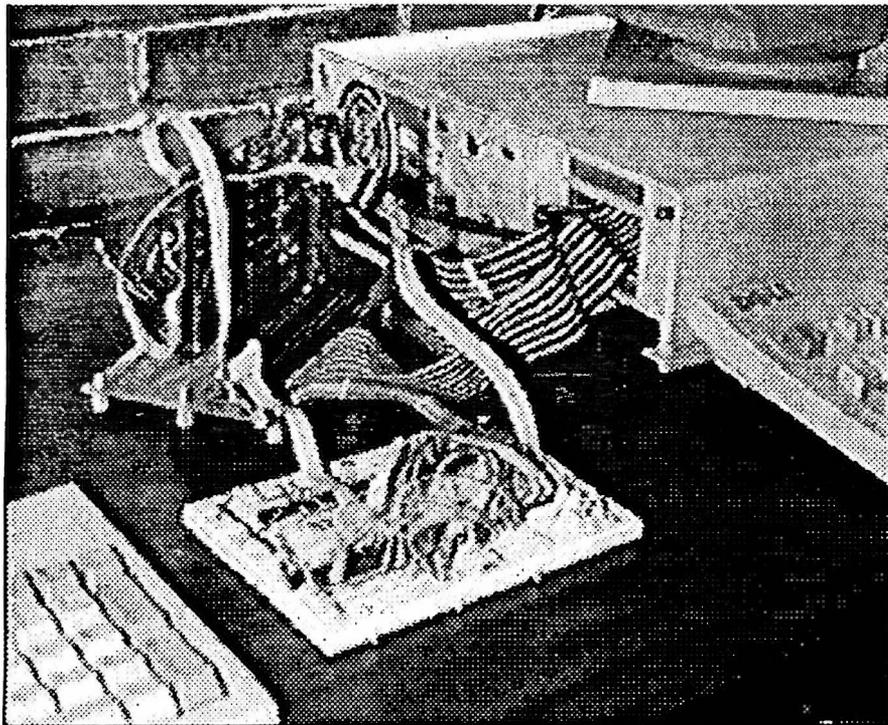


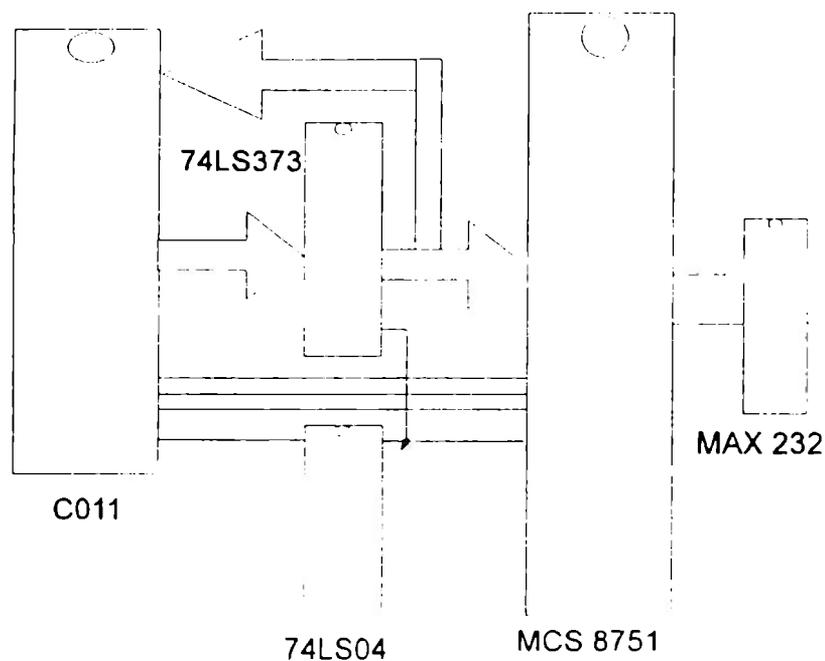
Figura 6.1 Sistema de prueba del prototipo

La comunicación con el controlador se estableció utilizando el puerto de comunicación serie de la computadora auxiliar, el cual cumple con el estándar RS232C.



Fotografía 6.1 Sistema de prueba del prototipo

Para el desarrollo del prototipo se utilizó, un microcontrolador Intel 8751, un registro de 8 bits 74LS373, un circuito adaptador de canal C011 y un circuito MAX 232. Junto con estos circuitos se utilizaron resistencias de "pull up", requeridas en los puertos de entrada, para visualizar temporalmente los datos a la salida del registro se utilizaron led's y para simular el envío de información se utilizó un micro-switch. Todos estos circuitos se interconectaron en un proto-board, como se muestra en el diagrama a bloques de la figura (6.2). El diagrama eléctrico de este circuito se localiza en el apéndice E.



**Figura 6.2 Diagrama a bloques del prototipo de prueba**

La conexión se hizo utilizando una ranura de expansión como se observa en la fotografía (6.1). La conexión del módulo TRAM hacia el prototipo se hizo utilizando dos cables planos a través de sus conectores.

Se muestra a continuación el listado del programa de prueba que se utilizó en la computadora auxiliar para realizar la comunicación con el transputer.

```

/*****
Este programa es para la comunicación serie de entrada / salida,
a través del puerto COM1 con estándar RS 232- C.
La función del BIOSCOM permite que se usen los servicios que ofrece la interrupción
INT14 del BIOS. Los parámetros de comunicación se deben inicializar
*****/

#include <bios.h>
#include <conio.h>
#include <stdio.h>
#include <dos.h>

/* DEFINICIONES */

#define COM1    0
#define DATA_READY 0x100
#define TRUE    1
#define FALSE   0

/*****
Parámetros de comunicación:( baud rate | paridad | bit_paro | long_dato)
para simular la comunicación del robot (9600 bauds | sin paridad | 1 bit | 8 bits)
*****/

#define SETTINGS ( 0xE0 | 0x0 | 0x0 | 0x03 )

int main(void)
{
    int in, out, status, DONE=FALSE ;
    clrscr();

/*****
Inicialización del puerto de comunicación
*****/

    bioscom(0, SETTINGS, COM1);
    printf("\n\nEl dato recibido es: \n");
    while (!DONE)

/* Lee el status del puerto */
    {
        status = bioscom(3, SETTINGS, COM1);
        if (status & DATA_READY)

/*****
Pregunta si existe dato a la entrada si es si entonces
Estas en modo RECEPTOR , recíbelo y desplégalo en pantalla

```

```

*****/
    if ((out = bioscom(2, SETTINGS, COM1) & 0xFF) != 0)
        printf("\%X",out);
    if (kbhit())

/*****
Tienes un dato para enviar, estas en modo TRANSMISOR
el dato es [ESC], termina comunicación, si no entonces envíalo
*****/
    {
        if ((in = getch()) == '\x1B')
            DONE = TRUE;
            bioscom(1, in, COM1);
            printf("\%X",in);
        }
    }
    return 0;
}

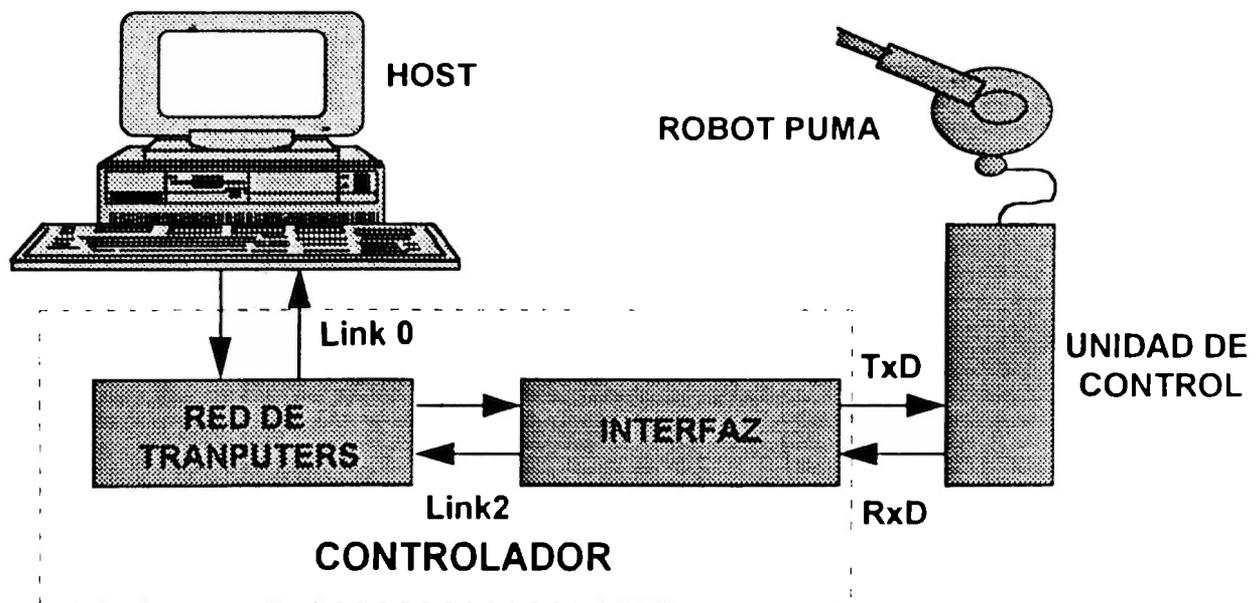
```

Las pruebas realizadas en laboratorio con este prototipo, el cual se muestra en la fotografía 6.1, nos permitieron comprobar la operación correcta del protocolo de comunicación establecido en la capa física, entre transputers y el robot simulado.

## **6.2.- CIRCUITO PROTOTIPO (PROTO - BOARD) <-> CELDA**

Después de haber realizado las pruebas anteriormente descritas, el prototipo en "proto - board" se interconectó a la celda de manufactura del Campus Toluca.

Esta conexión se hizo a través de la unidad de control del robot PUMA, ya que ésta dispone de una interfaz RS 232 para la comunicación, como se muestra en la figura (6.3).



**Figura 6.3 Prueba del prototipo con el robot PUMA**

Las pruebas del prototipo consistieron en el envío de información en tiempo real hacia el robot. Primeramente se envía información para inicializar el sistema y calibrar el robot. Posteriormente se envían una serie de puntos para posicionar el robot. La serie de comandos enviados al robot es mostrada en la siguiente sección.

En cada uno de los procesos de comunicación se siguió el protocolo "half duplex" descrito en la sección 4.2.

Mediante estas pruebas se comprobó en un ambiente industrial real el funcionamiento de los módulos de salida, entrada y la interfaz serie RS 232, así como de los programas residentes en la memoria EPROM del microcontrolador encargados del protocolo de comunicación.

### **6.3.- CIRCUÍTO IMPRESO DEL CONTROLADOR <-> CELDA**

Las pruebas finales del sistema se hicieron en el laboratorio de electrónica del Campus Toluca. Con ayuda del equipo de medición se comprobaron los niveles de voltaje, los niveles de ruido y los tiempos de respuesta de cada una de las señales que intervienen en la interfaz ya implementada en circuito impreso. Después de realizar estas mediciones, se utilizó la computadora auxiliar descrita en la sección 6.1 (simulador del robot), para revisar el comportamiento del protocolo de comunicación.

Finalmente, se reemplazó la terminal del robot PUMA por nuestro controlador y la interfaz en circuito impreso. La tarjeta se diseñó en un circuito impreso de doble lado (soldadura y componentes), capa anti-soldante, capa de leyendas y con las dimensiones (16.3 cm de largo x 10.3 cm de alto) requeridas para ser insertada en una ranura de expansión de una computadora de tipo PC. La información correspondiente para la fabricación de esta interfaz se localiza en el apéndice E.

Para la interconexión con la tarjeta de transputers se respetó el estándar de los módulos TRAM. En la fotografía [6.2] se muestra el circuito impreso de la interfaz. La interconexión de la tarjeta de transputers con la tarjeta de interfaz se realiza por medio de dos cables planos que se conectan a la salida del módulo TRAM y a la interfaz, con los conectores descritos en la sección 4.3.

Después de realizar las conexiones físicas, se realizó el control del robot mediante el envío y recepción de comandos en tiempo real. Se comprobó el correcto funcionamiento del sistema, ya que el robot ejecutó correctamente los comandos que se le enviaron, y al mismo tiempo respondió con las señales de reconocimiento necesarias para el control del protocolo de comunicación (sección 4.2). Los parámetros de control que se enviaron durante esta prueba se muestran a continuación. Mediante estos comandos se puede inicializar, calibrar y localizar puntos, para posteriormente recorrerlos de acuerdo a la secuencia determinada por el usuario.

**USUARIO (CONTROLADOR)**

**PUMA**

(Power ON)

LOAD VAL II FLOPPY TO DISK (Y)

n (RESPONDE NO)

INICIALIZATE (Y)

n (RESPONDE NO)

\*\*\*\*\*

Aqui empiezan los comandos del PUMA  
 los envia el usuario y los ejecuta el robot

\*\*\*\*\*

.ca (comando de calibración)

Are you sure (Y)

y (RESPONDE SI)

Ejecuta el comando que calibra al robot

.do ready (comando para iniciar)

Ejecuta comando

.do here id (comandos para el punto id)

Ejecuta comando

.move id (comando para posicionar  
 en el punto id)

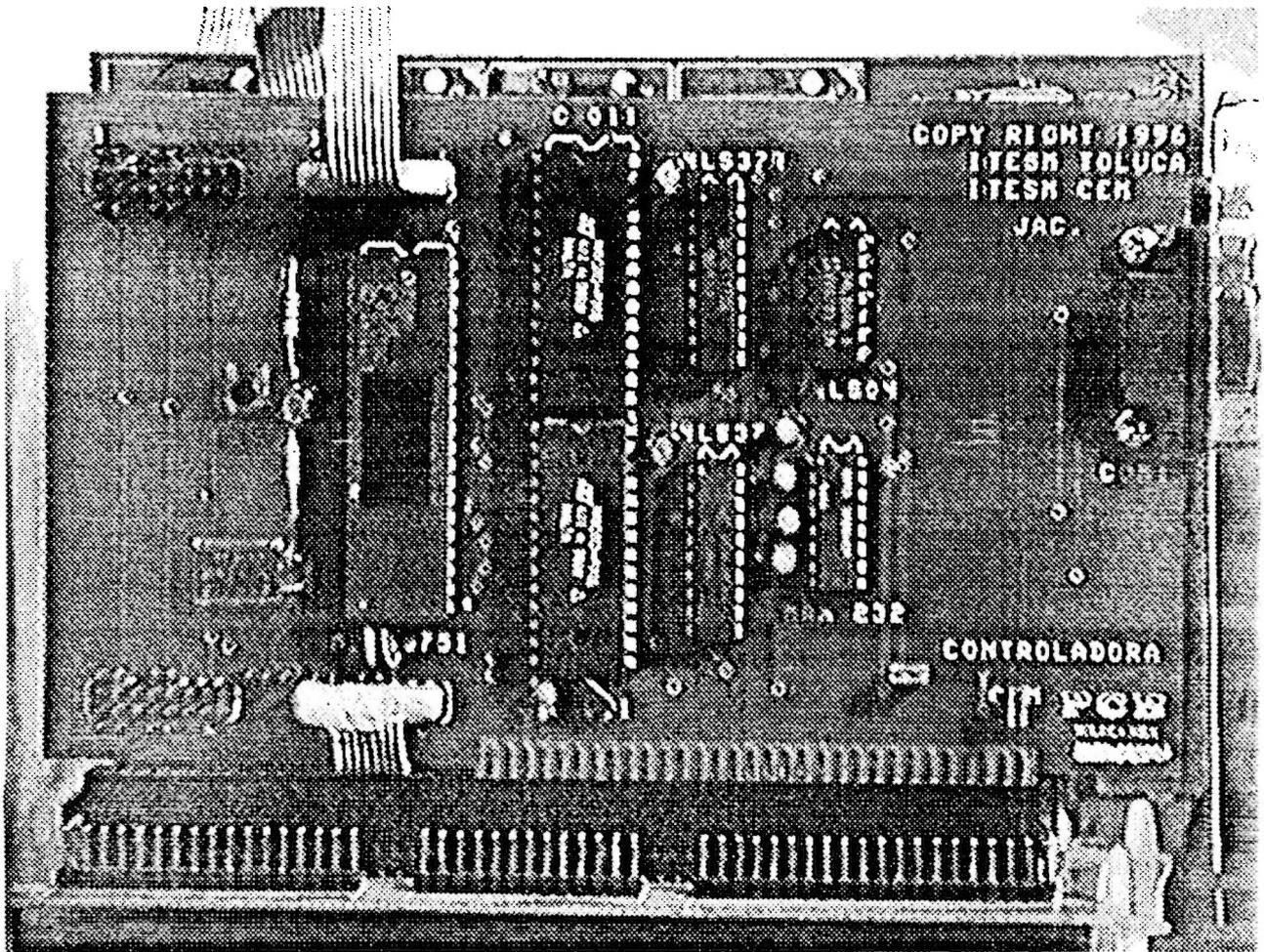
Ejecuta comando

.do openi (comando para abrir mano)

Ejecuta comando

.do closei (comando para cerrar mano)

Ejecuta comando



Fotografía 6.2 Circuito Impreso de la Interfaz de la CFM

## Capítulo 7

# CONCLUSIONES

El análisis que se realizó acerca de los transputers, elementos de procesamiento paralelo y con enlaces de alta velocidad, nos permite garantizar que se pueden establecer redes de procesadores reconfigurables y tolerantes a fallas para controlar celdas flexibles de manufactura. Si se usa una red de transputers se tiene una redundancia en hardware, por lo que se pueden declarar caminos alternos. Además, el hecho de tener un controlador paralelo nos permite tener redundancia en software, al poder implantar una réplica de los programas de control en diferentes procesadores.

Para lograr los objetivos planteados en esta tesis fue necesario el desarrollo de un sistema completo, integrado por hardware y software. El controlador es una máquina paralela compuesta por transputers, y diseñada para lograr la reconfigurabilidad que nos permite tener flexibilidad en la celda de manufactura. La interfaz diseñada en esta tesis resolvió el problema fundamental de la diferencia de velocidades de comunicación entre los elementos de la red de transputers y los elementos de la celda de manufactura, permitiendo una conexión simple entre ambos por medio del circuito impreso que se implementó.

Para realizar la interconexión del controlador con los diferentes elementos de la celda de manufactura, se eligió la interfaz serie con estándar RS 232-C, ya que ésta cumple con los requerimientos de comunicación de una celda (9600 bauds y una

distancia hasta de 30 metros), a bajo costo. Además, todos los elementos de la celda con la que trabajamos (CIM Campus Toluca) cuentan con esta interfaz.

En esta tesis se obtuvo como resultado principal el diseño, desarrollo y fabricación de un prototipo de interfaz entre un controlador a base de transputers y una celda de manufactura. Esta interfaz, además de cumplir con los requerimientos de la aplicación, cumple con los estándares de INMOS, el fabricante de transputers. El diseño fue modular, de tal forma que este dispositivo permite que el controlador sea escalable, es decir pueda crecer de acuerdo a las necesidades y posibilidades de los usuarios.

Una ventaja muy importante de este prototipo es el costo de la tarjeta de interfaz, de aproximadamente \$180.00 Dlls, comparada con el costo de una tarjeta comercial de INMOS con interfaz RS 232, con un costo 10 veces superior. Este costo se logró gracias a que en el diseño se emplearon diferentes tecnologías de circuitos integrados y microprocesadores de bajo costo, logrando con ésto desarrollar una arquitectura heterogénea barata.

Como resultado colateral a esta investigación se creó en el ITESM Campus Toluca una infraestructura con "tecnología de punta", que nos permite seguir investigando en el área de procesamiento paralelo.

### **7.1.- LIMITACIONES**

El sistema se diseñó en base a una tarjeta básica de transputers (SMT400A), que no cuenta con un circuito conmutador para la interconexión entre ellos. Ésto resulta una limitante, ya que la reconfiguración de la red es establecida mediante el alambrado de trayectorias como se muestra en el apéndice F.

El dispositivo microcontrolador que se utilizó cumple con los requerimientos mínimos de este proyecto. Cuenta con un banco de memoria de programa (EPROM interna de 4

Kbytes) suficiente para almacenar los programas de control para la comunicación en la capa física, pero su banco de memoria de datos (RAM interna de 128 bytes) es muy pequeño, lo que puede en un momento dado implicar un problema para una implantación más complicada.

## **7.2.- PROPUESTA DE MEJORAS A FUTURO**

Para una implantación más avanzada del controlador, se recomienda utilizar un modelo más poderoso de tarjeta principal en la que se incluya un circuito conmutador como el (C004) de INMOS. Ésto permitiría tener la posibilidad de reconfigurar dinámicamente la red de transputers. La interconexión del C011 con la interfaz diseñada en esta tesis no representa ningún problema, ya que ésta cumple con los estándares de los TRAMS de INMOS.

Para el diseño del circuito impreso de la interfaz, se tomaron las dimensiones de una tarjeta que abarca únicamente la mitad de la longitud del contenedor de una computadora (half card computer), limitando estas dimensiones a incluir únicamente los circuitos necesarios para la interfaz con un robot. Se recomienda que el diseño del circuito impreso de la interfaz sea al tamaño completo de los contenedores de las computadoras del tipo personal. De esta manera se podría incluir en una sola tarjeta el doble de circuitos, y por lo tanto el control para 2 robots y la interfaz para cuatro enlaces de comunicación ( cuatro TRAMS).

Para la comunicación entre la interfaz y el robot se puede pensar que la comunicación full-duplex es la más adecuada. Este modo de comunicación lo soporta el microcontrolador empleado en esta tesis, así como también un incremento en la velocidad hasta de 19,200 bauds. Por lo que ésta es otra oportunidad de desarrollo para trabajos futuros, siempre y cuando se requieran transmitir grandes volúmenes de información a una velocidad mayor, y los elementos de la celda de manufactura lo soporten.

---

Claramente se observa que existe una parte sensible a fallas en nuestro sistema, que es la computadora huésped. Por razones de costo se evitó la redundancia de está, pero físicamente es posible conectar dos computadoras huésped, cada una con transputers, usando nuestra interfaz.

# APÉNDICE A

## CIRCUITO ADAPTADOR DE CANAL.

El adaptador de canal IMS C011, es un circuito integrado diseñado por INMOS para establecer enlaces de comunicación de alta velocidad entre el transputer y otros dispositivos.

Todos los productos de INMOS que utilizan los enlaces de comunicación, sin importar el tipo de dispositivo, soportan la frecuencia de comunicación estándar de 10 Mbits/seg; la mayoría de los productos soporta también 20 Mbits/seg. El IMS C011 puede operar a cualquiera de estas frecuencias. La comunicación de datos que se establece es asíncrona, independiente de la fase del reloj.

El adaptador de canal puede operar en dos modos. En el modo 1, el IMS C011 convierte la información proveniente de un enlace de comunicación del Transputer, en dos canales de comunicación de entrada y salida paralela y líneas de control, que permiten establecer un protocolo handshake con otros dispositivos. Este modo de operación puede ser utilizado para que diferentes dispositivos periféricos paralelos se comuniquen con el transputer. En el modo 2 el C011 proporciona la interfaz entre un enlace de comunicación de un transputer y un bus de un sistema basado en microprocesadores, por ejemplo de una computadora del tipo personal.

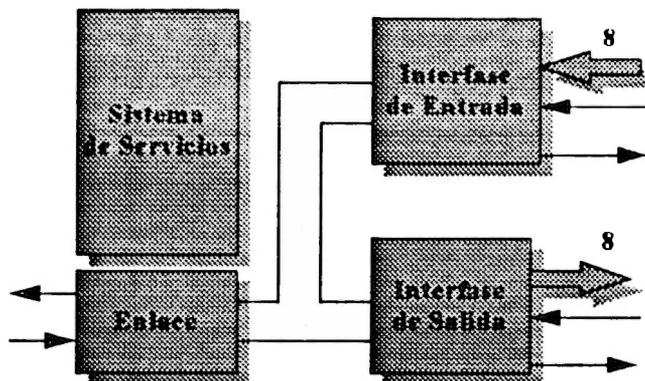
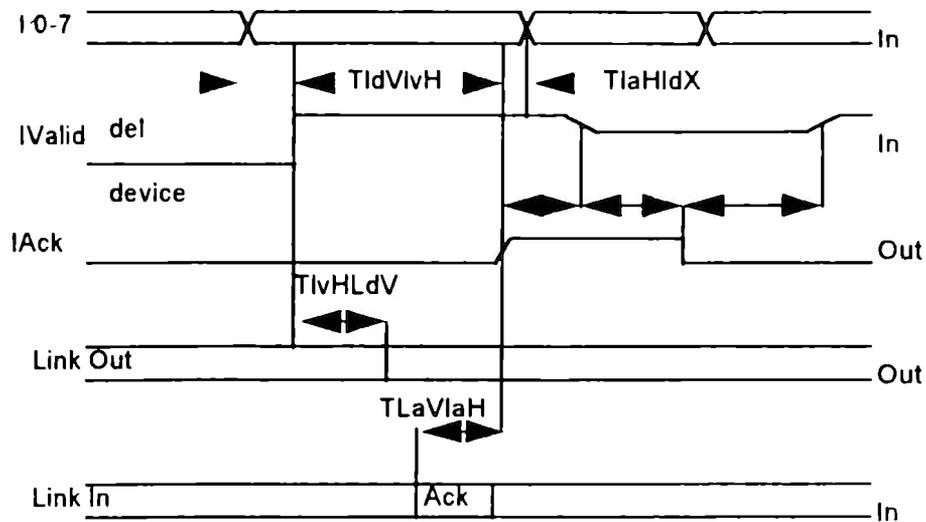


Figura (A.1) DIAGRAMA A BLOQUES C011 MODO 1

A continuación describiremos el funcionamiento de los puertos de entrada y salida del modo 1 usado para nuestra aplicación.

**Puerto de Entrada.**

Los 8 bits paralelos del puerto de entrada I0 - I7 pueden ser leídos por el transputer a través de uno de sus enlaces enlace de comunicación serie. Las señales de control IValid y IACK proporcionan la manera de establecer un protocolo handshake con este puerto. El protocolo de comunicación se establece de la manera que se muestra en la figura (A.2). Cuando el dato es válido sobre I0 - I7, la señal IVal es colocada en un nivel alto por el dispositivo periférico, para iniciar el handshake. El circuito C011 transmite entonces el dato presentado a su puerto de entrada hacia el transputer. Después de que la transmisión se realizó, espera sobre el enlace de entrada un reconocimiento de que la transmisión se realizó, y el IMS C011 coloca en un nivel alto la señal IACK. Para terminar el handshake, el dispositivo periférico debe regresar la señal de IVal a un nivel bajo. El C011 regresa entonces la señal de IACK a un nivel bajo para iniciar nuevamente otro ciclo de transmisión de información.



**Figura (A.2) IMS C011 MODO 1 DATOS DE ENTRADA**

**Puerto de Salida.**

Los 8 bits del puerto de salida pueden ser controlados por el enlace de comunicación del transputer. Las señales QVal y QAck proporcionan el control de handshake necesario para el manejo del puerto.

Un paquete de datos que es recibido sobre el enlace de un transputer es transferido sobre las salidas Q0 - Q7. En ese momento la señal QVal es colocada en un nivel alto por el C011, para iniciar el protocolo de comunicación. Después de que el dato ha sido leído, el dispositivo periférico coloca la señal de QAck en un nivel alto. El C011 envía un reconocimiento al transputer a través del enlace de salida, para indicar que la transacción se ha completado y coloca la señal de QVal en un nivel bajo para completar el handshake como se muestra en la figura (A.3).

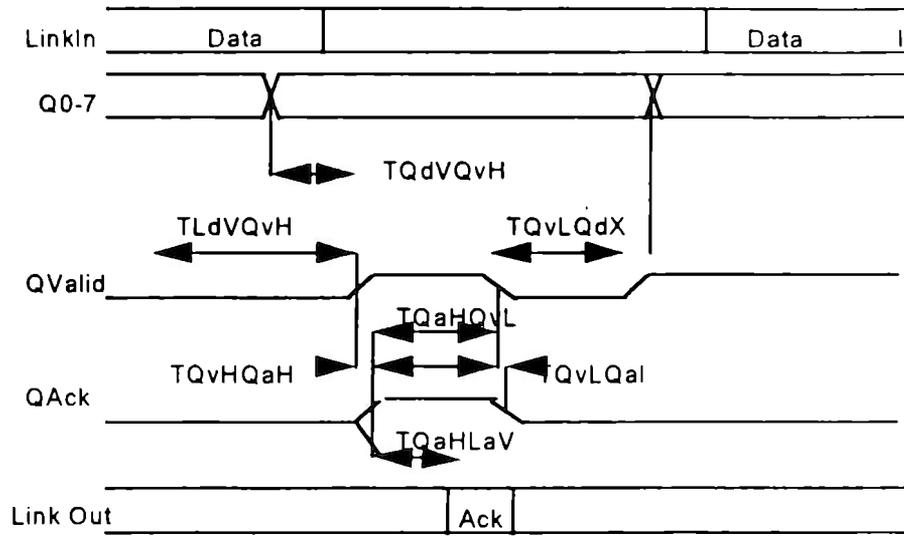


Figura (A.3) IMS C011 MODO 1 DATOS DE SALIDA

## APÉNDICE B

### MICROCONTROLADOR 8751.

El MCS-51 es una familia de circuitos integrados diseñado y desarrollado por INTEL Corporation. El microcontrolador 8751 incluye en un sólo circuito integrado, como se muestra en la figura (B.1), bancos de memoria RAM, ROM, interfaz serie y paralela, temporizadores y circuitos manejadores de interrupciones. Los microcontroladores se recomiendan en aplicaciones de control o en diseños de dispositivos de entrada/salida donde se requiere un mínimo de componentes y el poder de un sistema de procesamiento de información.

#### B.1.- ARQUITECTURA.

Las características del microcontrolador 8751 son:

- 4K bytes de memoria EPROM
- 128 bytes RAM
- 4 puertos de entrada/salida de 8 bits
- 2 timers de 16 bits
- Interfase serial
- 64 K para espacio de memoria de programa externa
- 64 K para espacio de memoria de datos externos
- Procesador booleano
- 4 microsegundos en operaciones de multiplicación / división

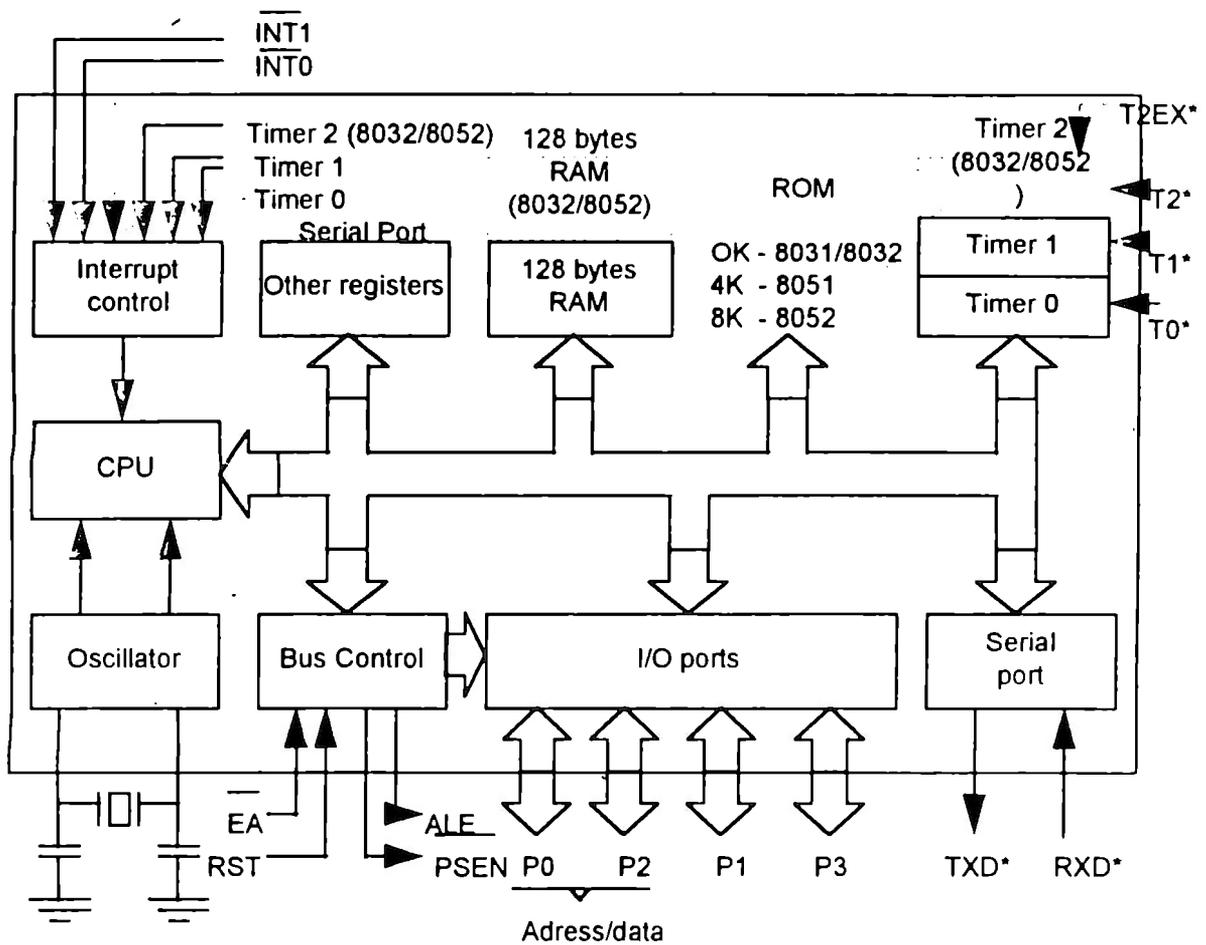


Figura (B.1) ARQUITECTURA MCS 51

## B.2.- MODOS DE OPERACIÓN.

### PUERTO 0

El puerto 0 es un puerto de doble propósito. En diseños de componentes mínimos, es usado como un puerto para salidas / entradas de propósito general. Para diseños más grandes con memoria externa se convierte en un bus con direcciones y datos multiplexados.

## PUERTO 1

El puerto 1 está dedicado a entradas / salidas paralelas. Los " pines " están disponibles para interconectar dispositivos externos que se requieran. Ninguna otra función tiene asignada el puerto 1, por lo que sólo es usado como interfaz externa.

## PUERTO 2

Es un puerto con doble objetivo: para salidas / entradas de propósito general, o como la parte alta del bus de direcciones para diseños con memoria de programa externa o más de 256 bytes de memoria de datos externa.

## PUERTO 3

El puerto 3 tiene dos objetivos. Realiza entradas/ salidas de propósito general y además sus pines son multifuncionales, cada uno teniendo su propósito alternativo relacionado con funciones específicas del 8051:

BIT NOM	BIT	FUNCTION ALTERNATIVA
P3.0	RXD B0H	Dirección del Puerto Serial
P3.1	TXD B1H	Transmitir Datos al Puerto Serial
P3.2	-INT0 B2H	Interrupción Externa 0
P3.3	-INT1 B3H	Interrupción Externa 1
P3.4	T0 B4H	Temporizador/contador 0 de entrada externa
P3.5	T1 B5H	Temporizador/contador 1 de entrada externa
P3.6	-WR B6H	Strobe de escritura para memoria de datos externa
P3.7	-RD B7H	Strobe de lectura para memoria datos externa
P1.0	T2 90H	Temporizador/contador 2 de entrada externa
P1.1	T2EX 91H	Temporizador/contador 2 de captura/repetición

El MCS 51 cuenta con 4 señales dedicadas a el control y operación del dispositivo. Estas señales son: " Program Store Enable ", " Address Latch Enable", " External Access" y "Reset".

### **PROGRAM STORE (PSEN)**

Es una señal de control que habilita la memoria de programa externa. Usualmente se conecta al " pin " de habilitación de salida (-OE) de un EPROM para permitir la lectura de bytes de programa.

La señal PSEN manda un pulso bajo durante la búsqueda de una instrucción. Los códigos binarios de un programa (opcodes) son leídos desde la EPROM, viajan a través del bus de datos y son decodificados en el registro de instrucciones del 8051. Cuando se ejecuta un programa desde la ROM interna (8051/8052), -PSEN permanece inactiva (alta).

### **ADRESS LATCH ENABLE (ALE)**

El bus 8051 usa la señal de ALE para demultiplexar el bus de direcciones y datos. Cuando el puerto 0 es usado en su modo alternativo (como el bus de datos y como la parte baja del bus de direcciones), ALE es la señal que mantiene la dirección en un registro externo durante la primera mitad de un ciclo de memoria. Realizado ésto, las líneas del puerto 0 son habilitadas para la entrada o salida de datos durante la segunda mitad del ciclo de memoria, cuando la transferencia de datos se realiza.

La señal ALE pulsa a una sexta parte de la frecuencia del oscilador del circuito integrado y puede ser usada como un reloj de propósito general por el resto del sistema.

## **EXTERNAL ACCES (EA)**

Sí la señal de entrada -EA es alta, el 8051/8052 ejecuta programas del ROM interno en los primeros 4KBytes de memoria.

Si es baja, los programas se ejecutan desde memoria externa sólo si además PSEN es baja, la ROM interna es deshabilitada y los programas se ejecutan desde EPROM externa.

## **RESET (RST)**

Cuando la entrada RESET sobre el " pin " 9 se pone en alto por al menos dos ciclos de máquina, los registros internos del 8051 son ocupados por valores apropiados para la reinicialización correcta del sistema.

## **ENTRADAS DEL OSCILADOR**

El 8051 trabaja con un oscilador interno que es controlado típicamente por un cristal conectado a los " pines " 18 y 19. La frecuencia nominal es de 12 MHz para la mayoría de los chips de la familia MCS-51.

## **CONEXIONES DE ALIMENTACIÓN**

El 8051 opera con una fuente simple de +5 Volts. La conexión para Vcc está sobre el pin 40, y Vss (tierra) está en el pin 20.

## APÉNDICE C

### INTERFAZ SERIE ( Estándar RS 232 ) con el circuito MAX 232.

Para la comunicación con los robots se envía y recibe información en forma serie. Para cumplir con el estándar de la interfaz RS 232 se utiliza un circuito MAX 232, el cual tiene como función la de intercambiar los niveles de las señales TTL / CMOS a niveles de la interfase RS - 232 o viceversa. En la figura [C.1], se muestra el diagrama a bloques del circuito.

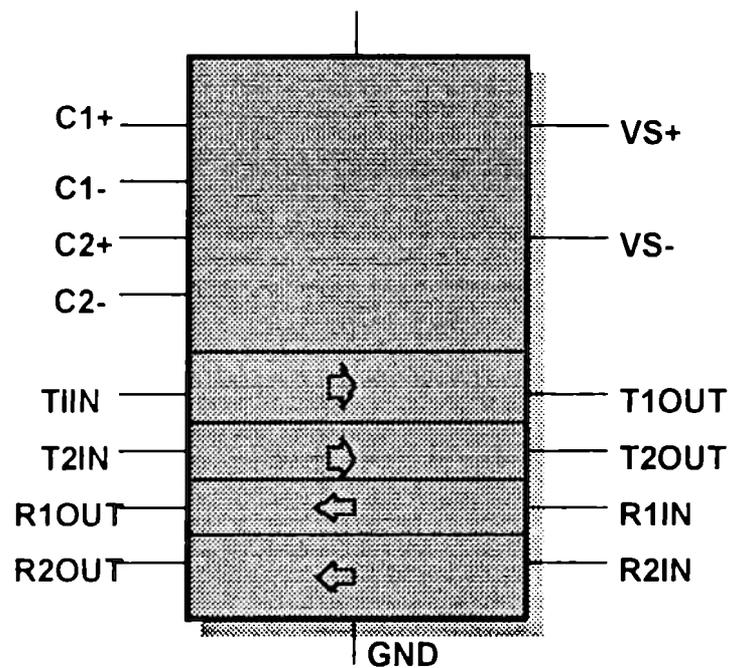


Figura (C.1) DIAGRAMA A BLOQUES MAX 232

El MAX 232 es un circuito integrado de tecnología BiCMOS, en el que se encuentran dos transmisores / receptores seriales. Tiene un generador de voltaje por medio de capacitores, el cual le permite generar pulsos que cumplan con los requeridos por la EIA - 232-C, a partir del voltaje de polarización de 5 volts.

## APÉNDICE D.1

### PROGRAMA DEL MICROCONTROLADOR: PROTOCOLO DE SALIDA

\*\*\*\*\*

\*

\*           PROGRAMA DE COMUNICACIÓN  
\*           TRANSPUTER -- ROBOT

\*

\*\*\*\*\*

\*\*\*\*\*

\*

\*           PRINCIPIA MEMORIA DE PROGRAMA  
\*           LOCALIDAD 0000H

\*\*\*\*\*

\*\*\*\*\*

\*

\*           DECLARACIÓN DE LOS VECTORES DE LAS  
\*           RUTINAS DE INTERRUPCION  
\*           INTERRUPCIÓN EXTERNA 0 GENERADA POR EL  
\*           TRANSPUTER

\*

\*\*\*\*\*

```
org 0h
jmp main
org 03h
jmp interT1 ;rutina de interrupcion para el T1
org 30h
```

\*\*\*\*\*

\*

\*           PROGRAMA PRINCIPAL  
\*           INICIALIZA MODO DE OPERACION DEL REGISTRO SERIE  
\*           CON EL SIGUIENTE FORMATO: LONG DATOS 8 BITS,

\* 9600 BAUDS, MODO ASINCRONO

\*\*\*\*\*

```
main:  mov SCON,#50h ;puerto serial modo 1 (uart 8bits)
       mov IE,#91h  ;habilita inter. ext1 y serial
       mov IP,#0Dh  ;maxima prioridad de interrupcion
       mov TMOD,#20h ;Timer1 modo autocarga
       mov TH1,#0FDh ;baud rate 9600
       setb TR1     ;arranca contador para el baud rate
```

\*\*\*\*\*

\*  
\* INICIALIZACIÓN DE LAS SEÑALES DEL PROTOCOLO  
\* SALIDA DE DATOS HABILITA 373, QVal, QAck, IVal, Iack,  
\* ESPERAN POR UNA INTERRUPCIÓN DEL TRAM  
\*

\*\*\*\*\*

```
stand: clr P0.2     ;habilita al 373
       clr P0.3     ;baja el nivel de la sepal de Qack
       clr P0.0     ;pone en cero el pin que recibe respuesta del Iack
       clr P0.1     ;
       sjmp stand   ;
```

\*\*\*\*\*

\*  
\* RUTINA DE SERVICIO DE INTERRUPCIÓN EXTERNA  
\* GENERADA POR EL TRAM PARA ENVIAR DATOS  
\* AL ROBOT, LA ENTRADA A ESTA RUTINA ES POR  
\* LA INTERRUPCIÓN 0 , GENERADA POR QVal  
\*

\*\*\*\*\*

```
interT1: clr TI     ;llego la señal de QVal
         mov A,P1    ;carga el contenido del puerto 0, datos a transmitir
         mov SBUF,A  ;envia los datos al puerto serial
inter1:  jnb TI,inter1 ;espera a que el registro SBUF se vacie
         setb P0.3   ;los datos se transmitieron, envia QAck positivo al TRAM
         clr TI
         RETI        ;regresa a stand, y espera otra solicitud de envio
```

## APÉNDICE D.2

### PROGRAMA DEL MICROCONTROLADOR: PROTOCOLO DE ENTRADA

\*\*\*\*\*

\*           PROGRAMA DE COMUNICACIÓN  
\*           ROBOT - TRANSPUTER

\*\*\*\*\*

\*\*\*\*\*

\*           PRINCIPIA MEMORIA DE PROGRAMA  
\*           LOACALIDAD 0000H

\*\*\*\*\*

\*\*\*\*\*

\*  
\*           DECLARACIÓN DE LOS VECTORES DE LAS  
\*           RUTINAS DE INTERRUPCION  
\*           INTERRUPCIÓN EXTERNA 0 GENERADA POR EL  
\*           TRANSPUTER  
\*           INTERRUPCIÓN SERIE GENERADA POR EL  
\*           ROBOT

\*\*\*\*\*

```
org 0h
jmp main
org 03h
jmp interT1 ;rutina de interrupcion para el T1
org 23h
jmp interT3 ;rutina de interrupcion para el max232
org 30h
```

\*\*\*\*\*

\*  
\*           PROGRAMA PRINCIPAL  
\*           INICIALIZA MODO DE OPERACION DEL REGISTRO SERIE  
\*           CON EL SIGUIENTE FORMATO: LONG DATOS 8 BITS,  
\*           9600 BAUDS, MODO ASINCRONO

\*\*\*\*\*

```
main:  mov SCON,#50h ;puerto serial modo 1 (uart 8bits)
      mov IE,#91h  ;habilita inter. ext1 y serial
      mov IP,#0Dh ;maxima prioridad de interrupcion
```

```

mov TMOD,#20h ;Timer1 modo autocarga
mov TH1,#0FDh ;baud rate 9600
setb TR1      ;arranca contador para el baud rate
*****
*
*           INICIALIZACIÓN DE LAS SEÑALES DEL PROTOCOLO
*           SALIDA DE DATOS HABILITA 373, QVal, QAck, IVal, Iack,
*           ESPERAN POR UNA INTERRUPCIÓN DEL TRAM
*
*****
stand:  clr P0.2      ;habilita al 373
        clr P0.3      ;baja el nivel de la señal de Qack
        clr P0.0      ;pone en cero el pin que recibe respuesta del Iack
        clr P0.1      ;
        sjmp stand    ;

*****
*
*           RUTINA DE SERVICIO DE INTERRUPCIÓN EXTERNA
*           GENERADA POR EL TRAM PARA ENVIAR DATOS
*           AL ROBOT
*           LA ENTRADA A ESTA RUTINA ES POR MEDIO DE LA
*           INTERRUPCIÓN 0, GENERADA POR QVal
*****
interT1: clr TI        ;llego la señal de QVal
         mov A,P1      ;carga el contenido del puerto 0, datos a transmitir
         mov SBUF,A    ;envia los datos al puerto serial
inter1:  jnb TI,inter1 ;espera a que el registro SBUF se vacie
         setb P0.3     ;los datos se transmitieron, envia QAck positivo al TRAM
         clr TI
         RETI          ;regresa a stand, y espera otra solicitud de envio
*****
*
*           RUTINA DE SERVICIO DE INTERRRUPCIÓN SERIE
*           GENERADA POR EL ROBOT CUANDO QUIERE
*           ENVIAR DATOS AL MODULO TRAM
*****
interT3: clr RI        ;limpia RI cuando ocurre la interrup.
         mov A,SBUF    ;mueve el dato serial al acumulador
         clr TI        ;limpia TI para la transmision
         setb P0.2     ;inhabilita al 373, selecciona puerto de entrada del C011
         mov P1,A      ;pone el dato serial en el puerto 1
         setb P0.1     ;genera la señal de control IVal
ciclo1: jnb P0.0,ciclo1 ;espera la señal de IAck respuesta del C011
         RETI          ;regresa a stand

```

## APÉNDICE D.3

### PROGRAMA DEL MICROCONTROLADOR: PROTOCOLO BIDIRECCIONAL

```

*****
*
*          PRINCIPIA MEMORIA DE PROGRAMA
*          LOCALIDAD 0000H
*
*****
*****
*
*          DECLARACIÓN DE LOS VECTORES DE LAS
*          RUTINAS DE INTERRUPCION
*          INTERRUPCIÓN EXTERNA 0 GENERADA POR EL
*          TRANSPUTER 1
*          INTERRUPCIÓN SERIE GENERADA POR EL
*          ROBOT
*
*****
org 0h
  jmp main
  org 03h
  jmp interT1    ;rutina de interrupcion para el T1
  org 23h
  jmp interT3    ;rutina de interrupcion para el max232
  org 30h
*
*****
*
*          PROGRAMA PRINCIPAL
*          INICIALIZA MODO DE OPERACION DEL REGISTRO SERIE
*          CON EL SIGUIENTE FORMATO: LONG DATOS 8 BITS,
*          9600 BAUDS, MODO ASINCRONO
*
*****
main:  mov  SCON,#50h    ;puerto serial modo 1 (uart 8bits)
      mov  IE,#91h     ;habilita inter. ext0 y serial
      mov  IP,#11h     ;maxima prioridad de interrupcion
      mov  TMOD,#20h   ;Timer1 modo autocarga
      mov  TH1,#0FDh   ;baud rate 9600
      setb TR1         ;arranca contador para el baud rate

```



## APÉNDICE D.4

### PROGRAMA FINAL DE LA INTERFAZ : CONMUTADOR 2X1

\*\*\*\*\*

\*

\*           PRINCIPIA MEMORIA DE PROGRAMA  
\*                                   LOCALIDAD 0000H

\*\*\*\*\*

\*

\*           DECLARACIÓN DE LOS VECTORES DE LAS  
\*           RUTINAS DE INTERRUPCION  
\*           INTERRUPCIÓN EXTERNA 0 GENERADA POR EL  
\*           TRANSPUTER 1  
\*           INTERRUPCIÓN EXTERNA 1 GENERADA POR EL  
\*           TRANSPUTER 2  
\*           INTERRUPCIÓN SERIE GENERADA POR EL  
\*           ROBOT

\*\*\*\*\*

```
org 0h
  jmp main
  org 03h
  jmp interT1      ;rutina de interrupcion para el T1
  org 13h
  jmp interT2      ;rutina de interrupsion para el T2
  org 23h
  jmp interT3      ;rutina de interrupcion para el max232
  org 30h
```

\*\*\*\*\*

\*

\*           PROGRAMA PRINCIPAL  
\*           INICIALIZA MODO DE OPERACION DEL REGISTRO SERIE  
\*           CON EL SIGUIENTE FORMATO: LONG DATOS 8 BITS,  
\*           9600 BAUDS, MODO ASINCRONO

\*\*\*\*\*

```

main:  mov  SCON,#50h    ;puerto serial modo 1 (uart 8bits)
      mov  IE,#95h     ;habilita inter. ext0, int ext1 y serial
      mov  IP,#11h     ;maxima prioridad de interrupcion
      mov  TMOD,#20h   ;Timer1 modo autocarga
      mov  TH1,#0FDh   ;baud rate 9600
      setb TR1        ;arranca contador para el baud rate

```

```

stand: mov  P0,#00h
      clr  RI
      clr  TI
      sjmp stand

```

\*\*\*\*\*

```

*           RUTINA DE SERVICIO DE INTERRUPCIÓN EXTERNA
*           GENERADA POR EL TRAM 0 PARA ENVIAR DATOS
*           AL ROBOT
*           LA ENTRADA A ESTA RUTINA ES POR MEDIO DE LA
*           INTERRUPCIÓN 0, GENERADA POR QVal

```

\*\*\*\*\*

```

interT1: clr  TI
      mov  A,P1        ;carga el contenido del puerto 1
      mov  R0,#01h
      cjne A,#0Dh,sigue1
      mov  SBUF,#0Dh   ;envia cr al puerto serial

```

```

inter1:  jnb  TI,inter1
      setb P0.3       ;envia ack positivo al Trans
      clr  TI
      clr  RI
      setb IE.4
      RETI

```

```

sigue1: clr  IE.4.....;deshabilita int. serie(RI)

```

```

      mov  SBUF,A
inter1a: jnb  TI,inter1a
      setb P0.3
      clr  TI
      clr  RI
      RETI           ;regresa a stand

```

\*\*\*\*\*

```

*           RUTINA DE SERVICIO DE INTERRUPCIÓN EXTERNA
*           GENERADA POR EL TRAM 2 PARA ENVIAR DATOS
*           AL ROBOT
*           LA ENTRADA A ESTA RUTINA ES POR MEDIO DE LA
*           INTERRUPCIÓN 1, GENERADA POR QVal

```

\*\*\*\*\*

```

interT2: clr TI
          mov A,P2
          mov R0,#02h
cjne A,#0Dh,sigue2
          mov SBUF,#0Dh      ;envia cr al puerto serial
inter2:  jnb TI,inter2
          setb P0.7
          clr TI
          RETI

```

```

sigue2:  clr IE.4
          mov SBUF,A
inter2a: jnb TI,inter2a
          setb P0.7
          clr TI
          clr RI
          RETI      ;regresa a stand

```

\*\*\*\*\*

```

*
*           RUTINA DE SERVICIO DE INTERRUPCIÓN SERIE
*           GENERADA POR EL ROBOT CUANDO QUIERE
*           ENVIAR DATOS A LOS MODULOS TRAM 0 Y TRAM 1
*

```

\*\*\*\*\*

```

interT3: clr RI      ;limpia RI cuando ocurre la interrup.
          clr TI
          mov A,SBUF  ;mueve el dato serial al acumulador
          clr TI      ;limpia TI para la transmision
          cjne R0,#01h,tram2
          setb P0.2    ;inhabilita al 373
          mov P1,A     ;pone el dato serial en el puerto 1
          setb P0.1    ;envia lval
ciclo1:  jnb P0.0,ciclo1
          clr P0.1
          RETI      ;regresa a stand

```

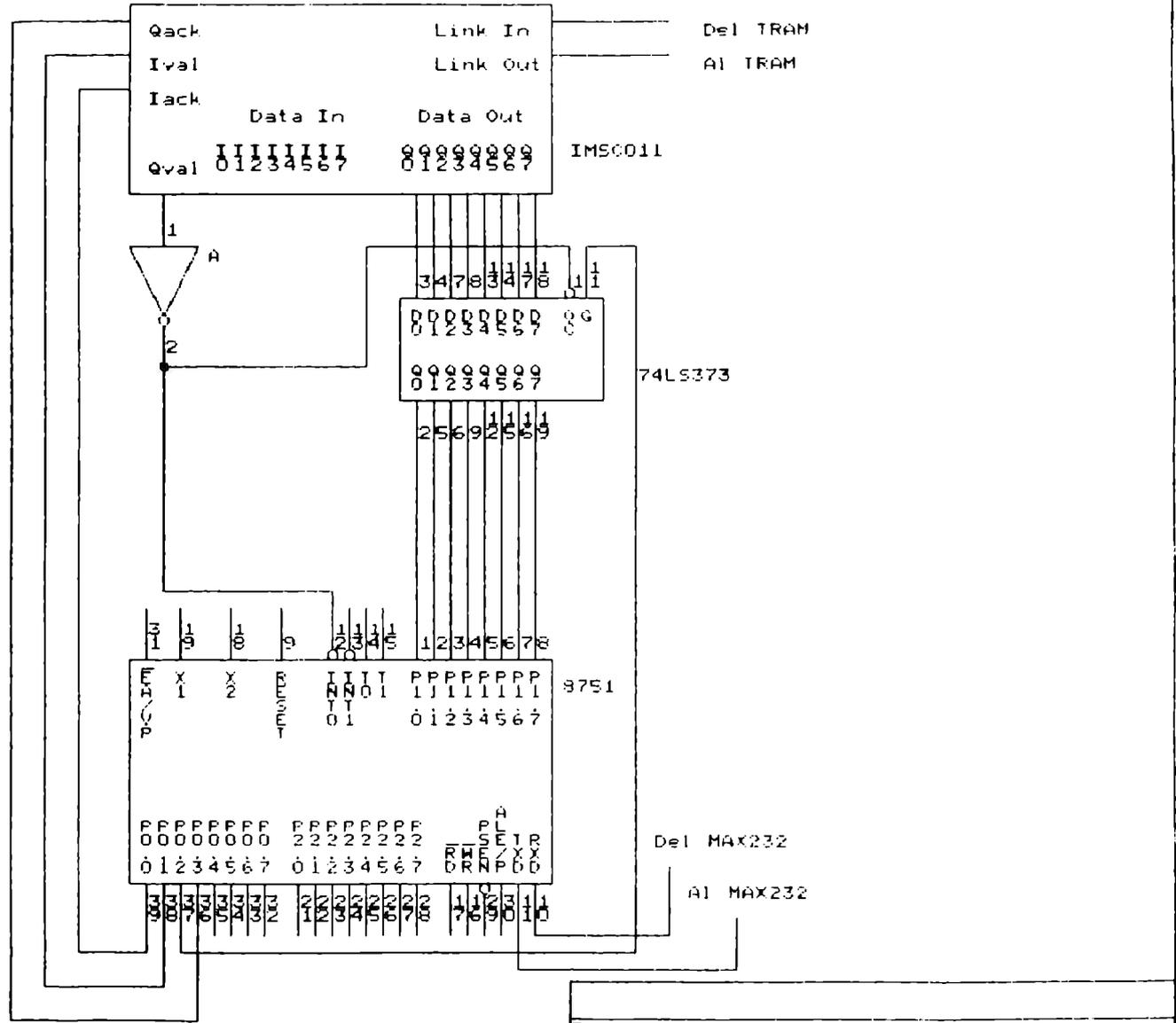
```

tram2:   setb P0.6
          mov P2,A
          setb P0.5
ciclo2:  jnb P0.4,ciclo2
          clr P0.5
          RETI

```

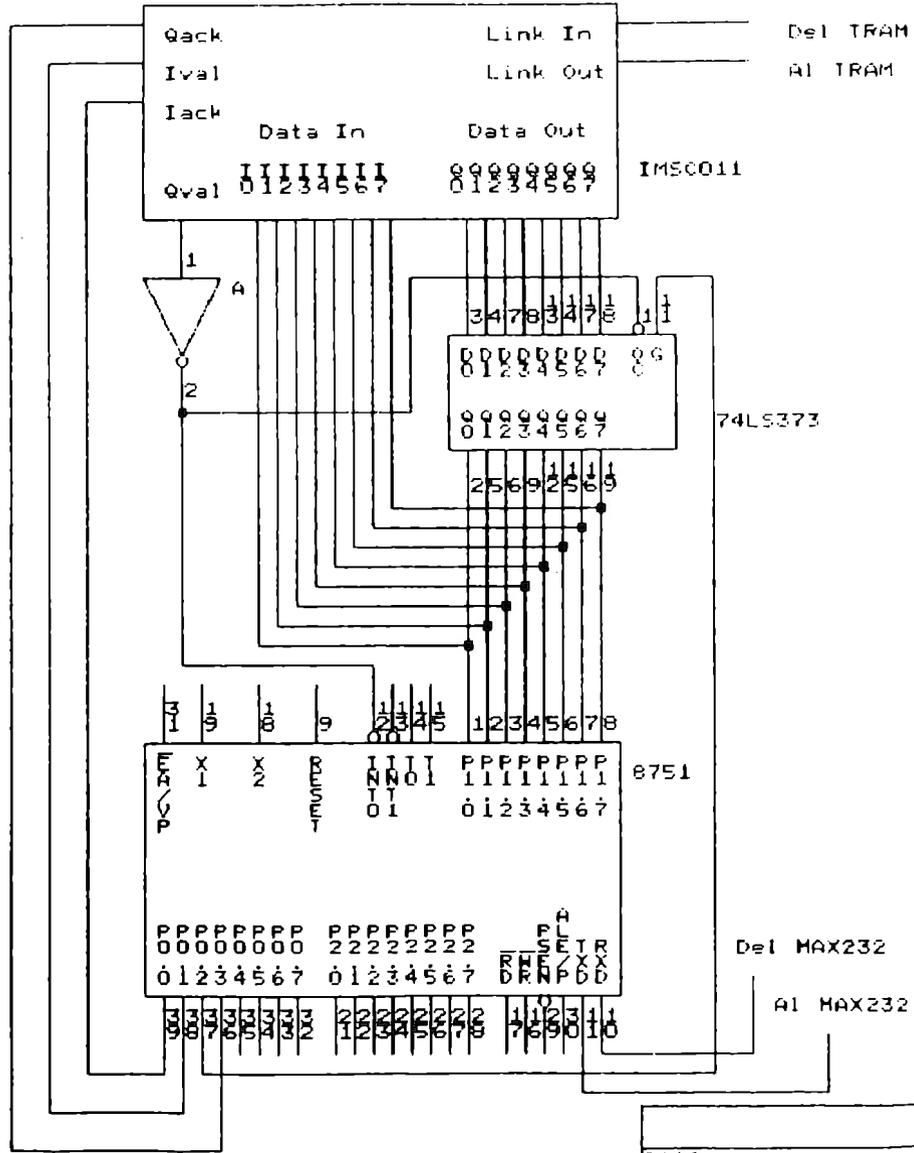
**APÉNDICE E**

**DIAGRAMAS ELECTRONICOS DE LA INTERFAZ DE SALIDA**



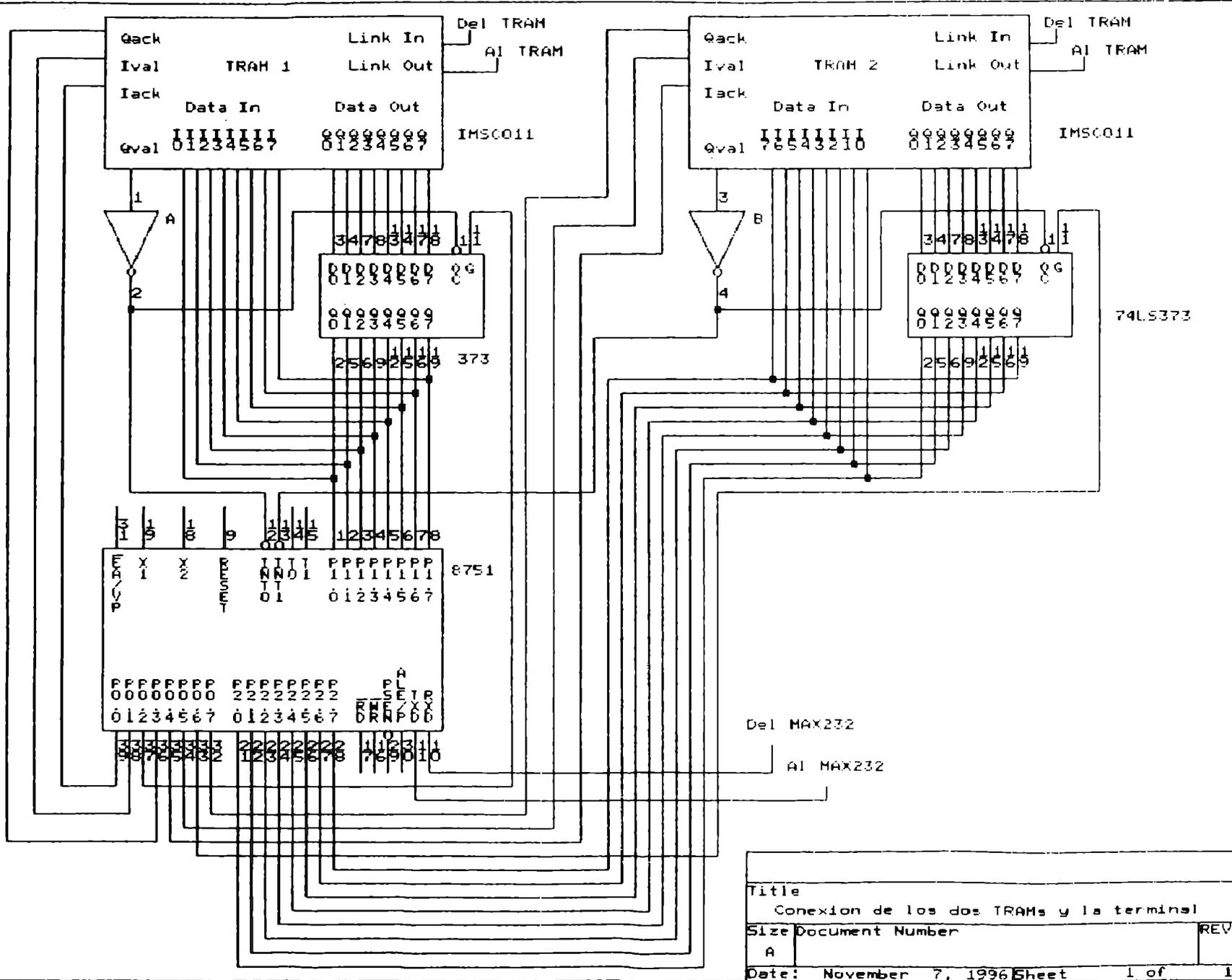
Title		Conexion del TRAM hacia la terminal	
Size Document Number			REV
A			
Date: November 6, 1996	Sheet	2 of	3

**DIAGRAMAS ELECTRONICOS DE LA INTERFAZ DE ENTRADA**



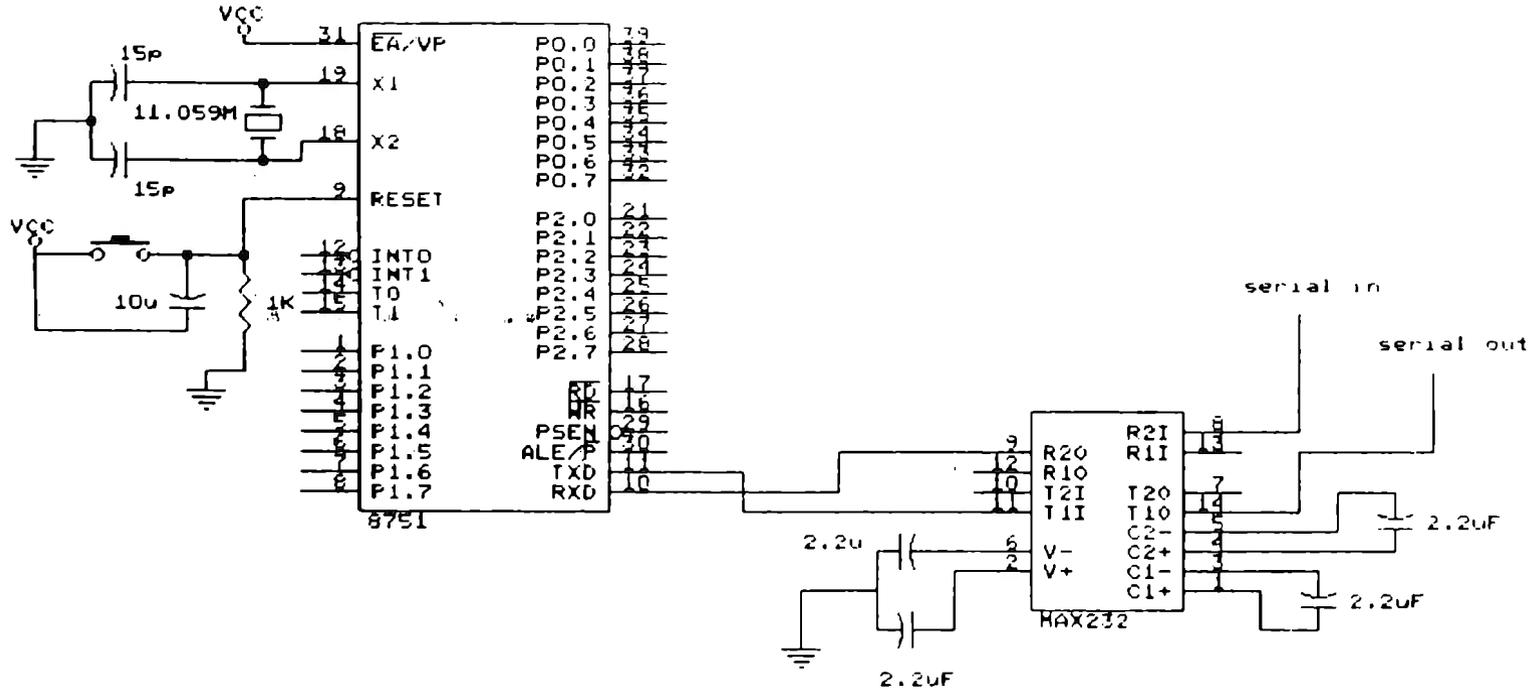
Title	
Conexion de la terminal hacia el TRAM	
Size Document Number	REV
A	
Date: November 6, 1996	Sheet 3 of 3

**DIAGRAMAS ELECTRONICOS DE LA INTERFAZ DE FINAL**



Title		
Conexion de los dos TRAMs y la terminal		
Size	Document Number	REV
A		
Date: November 7, 1996		Sheet 1 of 1

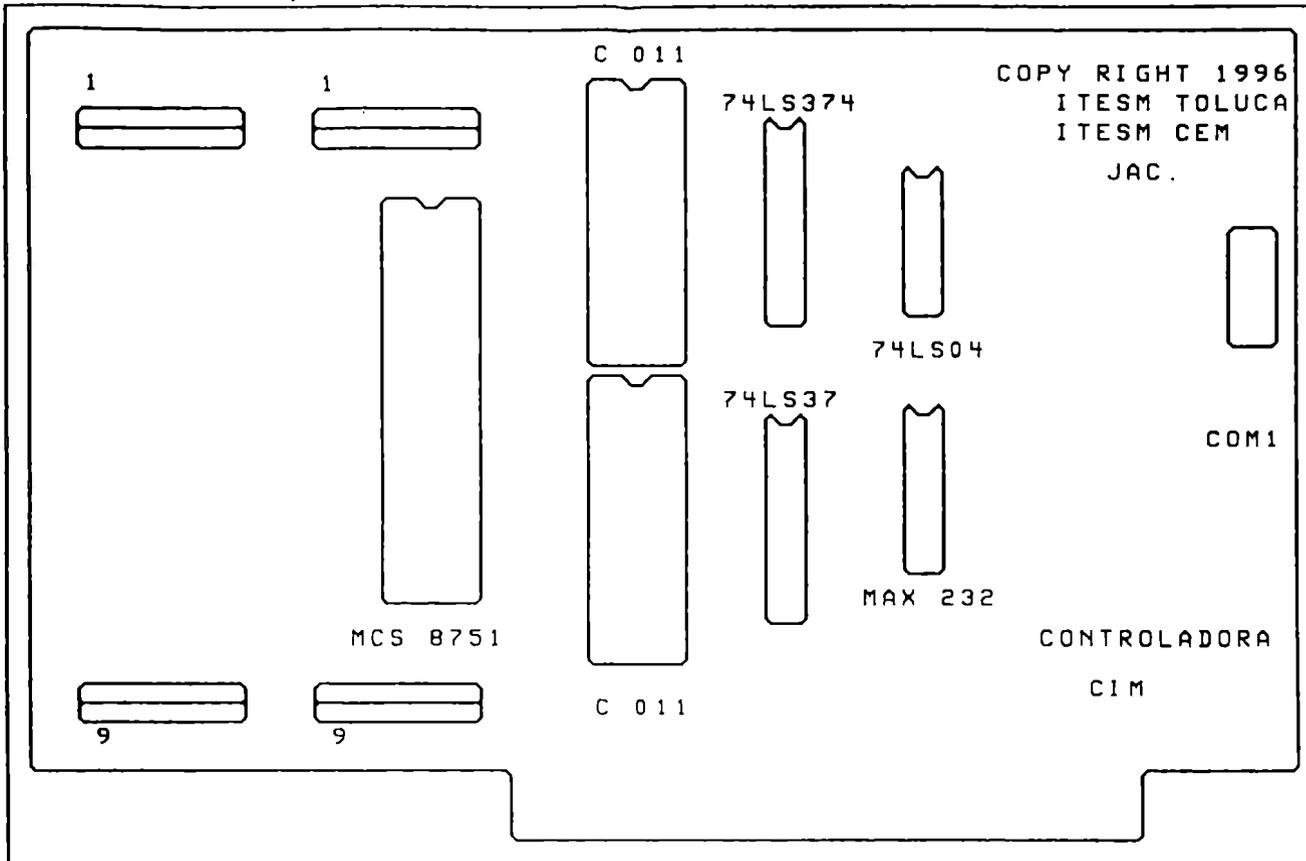
**DIAGRAMAS ELECTRONICOS DEL 8751 Y EL MAX232**



Title		
Conexion entre el 8751H y el MAX232		
Size	Document Number	REV
A		
Date:	November 5, 1996	Sheet 1 of 3

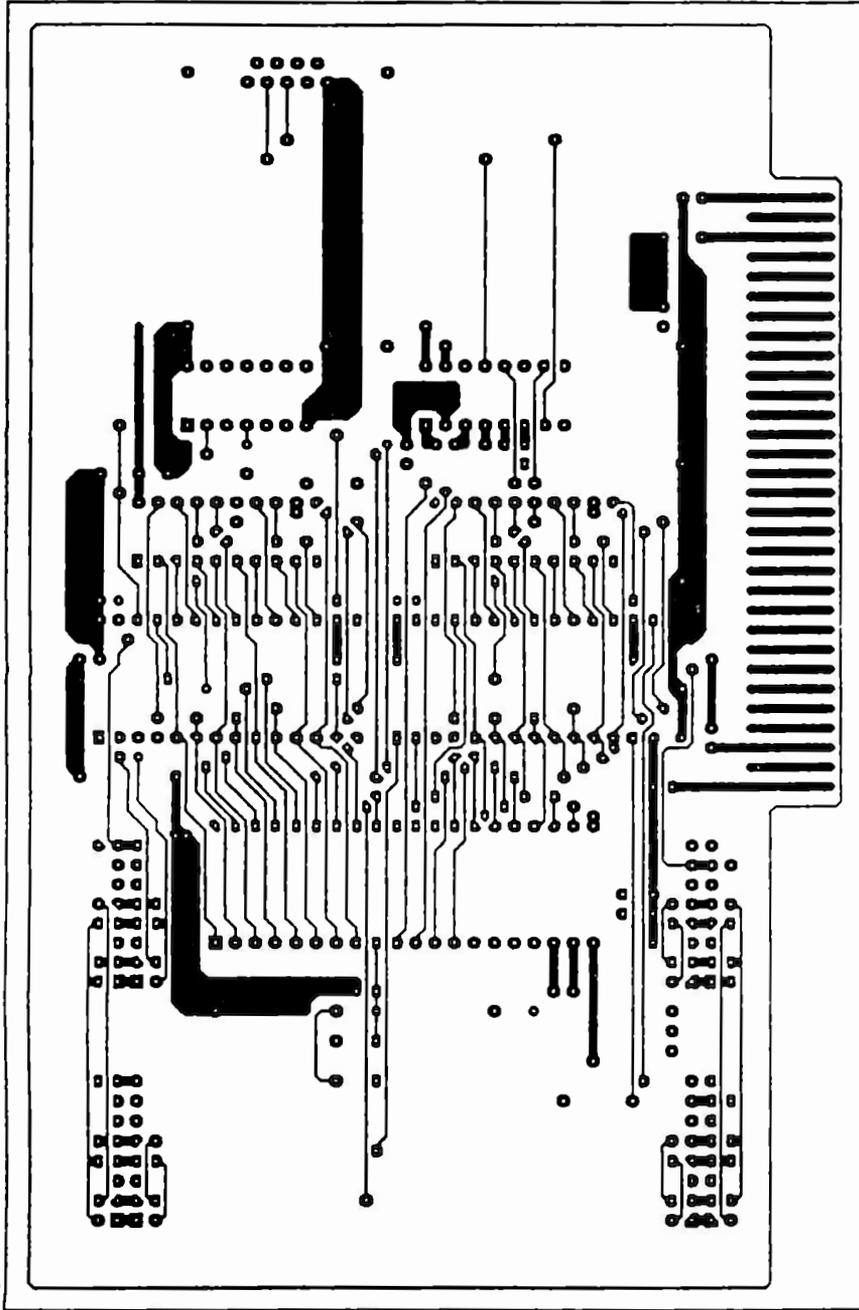
**CIRCUITO IMPRESO DE LA INTERFAZ: LEYENDAS**

2X artwork 22 Oct 98 16:36:48  
tra04.pcb  
v1.3 r4 holes: 440 silkscreen  
approximate size: 6.50 by 4.15 inches



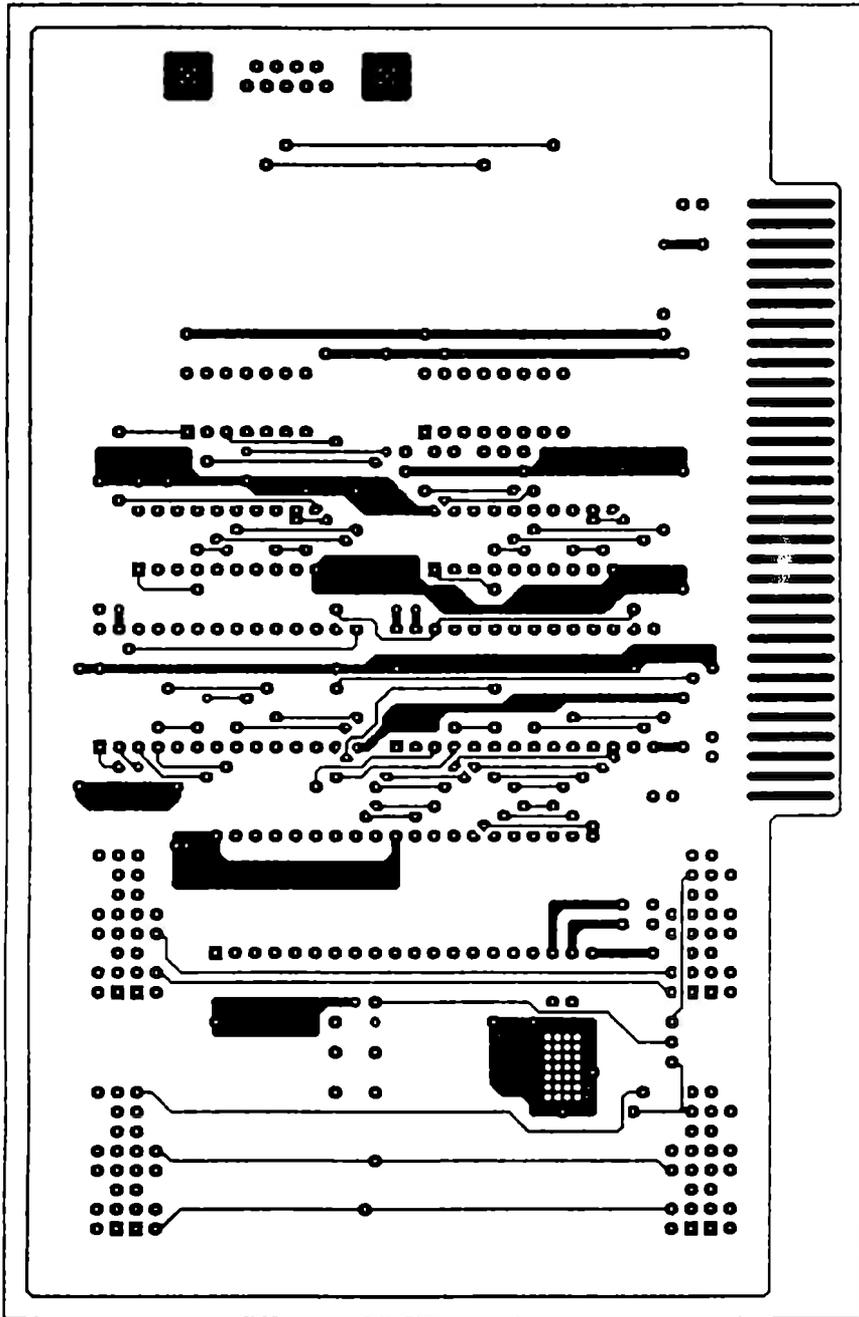
# CIRCUITO IMPRESO DE LA INTERFAZ: LADO SOLDADURA

2X artwork      22 Oct 96    16 15:26  
tra04.pcb  
v1.3 r4 holes: 440    solder side  
approximate size: 6.50 by 4.15 inches



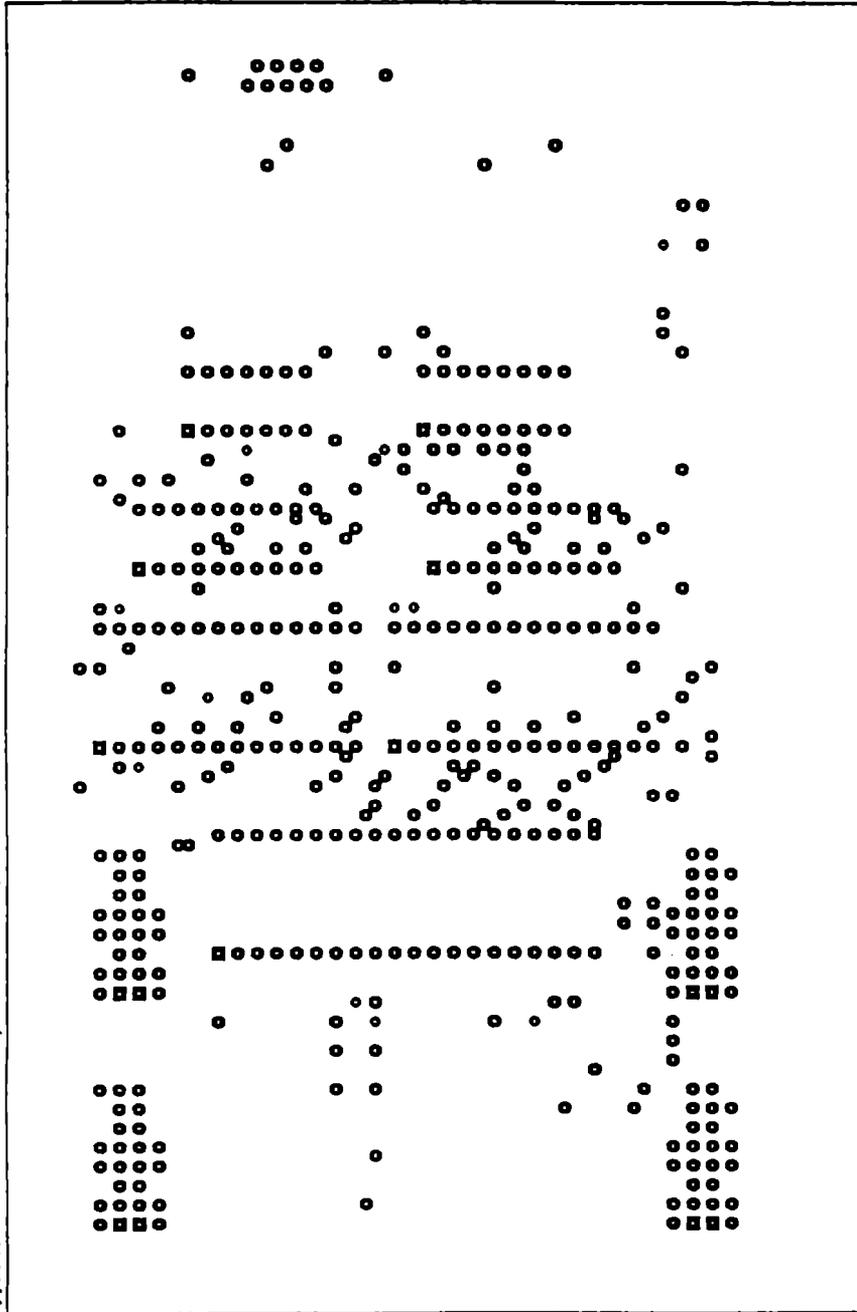
# CIRCUITO IMPRESO DE LA INTERFAZ: LADO COMPONENTES

2X artwork 22 Oct 98 16:24:36  
tre04.pcb  
v1.3 r4 holes: 440 component aids  
approximate size: 6.50 by 4.15 inches



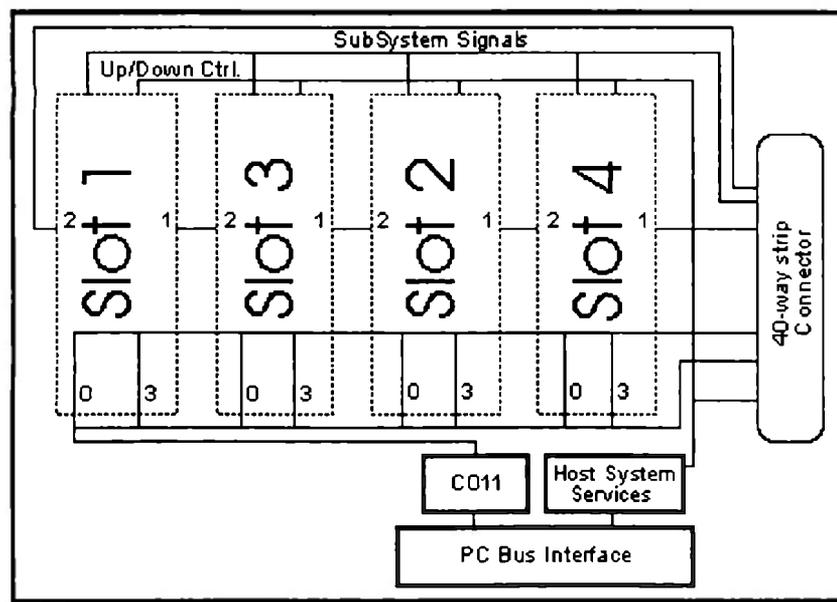
# CIRCUITO IMPRESO DE LA INTERFAZ: CAPA ANTI-SOLDANTE

2X artwork 22 Oct 98 16:48:30  
tra04.pcb  
v1.3 r4 holes: 440 padmaster  
approximate size: 6.50 by 4.15 inches



## APÉNDICE F

La tarjeta SMT004 está diseñada para acomodar hasta 4 módulos de transputers dentro de una ranura (slot) de expansión de una computadora del tipo personal. Normalmente ocupa la mitad de la longitud estándar de un slot, como se muestra en el diagrama a bloques funcional de la figura (F.1).



**Figura (F.1) Diagrama a Bloques Funcional de la Tarjeta SMT004**

La tarjeta SMT004 cuenta con una interfaz para una computadora PC, que le permite que la computadora inicie (reset), analice y se comuniquen con una red de transputers. El enlace 0 y el enlace 3 de cada módulo TRAM se localizan en un conector de 40 vías.

Las señales de Reset / Analyse / Error de cada TRAM pueden ser conectadas a la señales de otro Transputer o conectadas al bus de la PC. El Slot 1 tiene un puerto para el control de un subsistema, y estas señales pueden ser utilizadas para el control de módulos TRAM remanentes. Las señales Up, Down y Subsystem se encuentran disponibles en el conector de 40 vías. La dirección #150 de la PC es usada como una interfaz con el bus de 8 bits.

## F.1.-REGISTROS

### Registros Programables

Sobre la tarjeta SMT004 existen un grupo de registros que se programan para realizar el control de varias funciones requeridas en los enlaces de comunicación. Estos registros permiten controlar y determinar el status de una red de transputers externa.

Los registros son mapeados como un puerto de entrada / salida a partir de la dirección base #150.

<b>Dirección</b>	<b>Registro</b>
Base+0	Registro de Entrada de Datos (ro)
Base+1	Registro de Salida de Datos (wo)
Base+2	Registro de Status Entrada (rw)
Base+3	Registro de Status Salida (rw)
Base+16	Registro de Error (ro)
Base+16	Registro de Reset (wo)
Base+17	Registro de Análisis (wo)

**Tabla F1 Mapeo del Puerto de Entrada/Salida**

## Registros de Entrada / Salida de Datos

Estos dos registros se encuentran localizados en el circuito integrado C011. La función de estos registros es la de enviar y recibir datos. Cuando el registro de status indica que se encuentra listo para recibir un dato válido, o cuando contiene un dato válido, entonces el dato es escrito o leído a / o desde estos registro.

## Registros de Entrada / Salida de Status

Estos registros (tablas F2 y F3) indican el status de la entrada / salida de datos.

rd	Bit 0	0	No datos de Entrada
rd	Bit 0	0	No datos de Entrada
		1	
wr	Bit 1	0	int. Entrada deshabilitada
		1	int. Entrada habilitada

**Tabla F2 - Status de Entrada.**

rd	Bit 0	0	Salida Ocupada
		1	Salida Lista
wr	Bit 1	0	int Salida deshabilitada
		1	int. Salida habilitada

**Tabla F3 - Status de Salida.**

El origen y destino de las señales de Reset, Analyse y Error esta determinada por la presencia de las resistencias LS4 y LS5.

## Registro de Error

El registro de error almacena el status de la red de transputers externa. La señal de error puede ser originada en el slot 1 o bien por lo línea de error sobre el conector de 40 vías.

rd	Bit 0	0	Error Falso
		1	Error Verdadero

**Tabla F4 - Registro de Error**

## Registro de Reset

Este registro se puede usar para controlar la red de transputers externa. Normalmente esta señal es realimentada al SMT004 para el control de cualquier TRAM. Se puede escribir en el registro para controlar el estado de la línea PCNotReset localizada en el conector de 40 vías.

wr	Bit 0	0	Reset Falso
		1	Reset Verdadero

**Tabla F5 Registro de Reset**

## Registro de Análisis

Se puede escribir a este registro para controlar el estado de la línea de Análisis en la tarjeta de transputers, ó en la línea PCAnalyse en el conector de 40 vías.

wr	Bit 0	0	Análisis Falso
		1	Análisis Verdadero

**Tabla F6 - Registro de Análisis**

## F.2.- Control de la Configuración

El SMT004 es una tarjeta de bajo costo en la que todas las opciones son pre-configuradas de fábrica utilizando resistencias de montaje superficial. Estas opciones pueden ser cambiadas por el usuario o por el fabricante si es requerido.

### Velocidad de los Enlaces

Los enlaces de INMOS son capaces de trabajar a dos velocidades diferentes. La tarjeta tiene dos grupos de resistencias, mediante las cuales el usuario puede seleccionar la velocidad de los enlaces de comunicación de los TRAM y la velocidad de la interfaz con el bus, independientemente.

LS1	LS2	Velocidad	Comentario
Salida	Salida	N/A	No funciona.
Salida	Entrada	10MBit/s	De Fabrica (Default)
Entrada	Salida	20MBit/s	
Entrada	Entrada	N/A	No funciona.

**Tabla F7 - Velocidad de la Interfaz C011**

LS6	LS7	Velocidad	Comentario
Salida	Salida	20MBit/s	
Salida	Entrada	N/A	No funciona
Entrada	Salida	N/A	No funciona
Entrada	Entrada	10MBit/s	De Fabrica (Default).

**Tabla F8 - Velocidad de los Enlaces TRAM**

## Control de Subsistemas

El circuito impreso está diseñado para que los TRAM de los slots 1, 2, 3 y 4 puedan ser controlados por una variedad de fuentes. Para el slot 0 existe la opción de que el control se establezca por la interfaz con la PC o bien por medio de las líneas de control Up que se localizan el conector de 40 vías. La selección se realiza a través de las resistencias LS4 y LS5. Para los otros tres slots la selección del Reset y Analiza se realiza por medio del puerto del subsistema sobre el primer TRAM o las líneas de control Down.

<b>Res.</b>		<b>Comentario</b>
LS4	Entrada	Subsistema Activo -De Fábrica (Default)
	Salida	Abajo No Activo
LS5	Entrada	Interfaz PC Activa
	Salida	Arriba Interfaz no Activa

**Tabla F9 - Subsistema de Control**

## C011 Selección del Enlace

La resistencia LS3 es usada para seleccionar la fuente y el destino de la información que pasa a través del adaptador de enlace C011. Ésta puede ser colocada para que el envío y recepción de información sea por medio del conector de 40 vías, o por medio del TRAM del slot1. Se tiene que seleccionar una de las dos opciones.

<b>Res.</b>		<b>Comentario</b>
LS3	Entrada	40 - Vías Conector
	Salida	Slot 1, De Fabrica (Default)

**Tabla F10 - Selección del Enlace C011**

**Conector de 40 Vías.**

Para usar la conexión pipe el IC7 debe estar instalado (74BC125). LS3 selecciona la conexión con el C011 (PL1 o TRAM enlace 0).

<b>Pata</b>	<b>Señal</b>	<b>Pata</b>	<b>Señal</b>
1	Gnd	21	TRAM 4 Enlace 7 Out
2	UpNot Reset	22	TRAM 4 Enlace 7 In
3	UpNot Analyse	23	Gnd
4	UpNot Error	24	C011 Enlace In
5	NC	25	Pipe Head Enlace Out
6	NC	26	Pipe Head Enlace In
7	TRAM 1 Enlace 3 Out	27	Subsystem Not Reset
8	TRAM 1 Enlace 3 In	28	NC
9	Gnd	29	Subsystem Not Analyse
10	TRAM 2 Enlace 2 Out	30	Subsystem Not Error
11	TRAM 2 Enlace 2 In	31	Pie Tail Enlace Out
12	TRAM 2 Enlace 3 Out	32	Pipe Tail Enlace In
13	TRAM 2 Enlace 3 In	33	NC
14	TRAM 3 Enlace 4 Out	34	NC
15	TRAM 3 Enlace 4 In	35	Down Not Reset
16	Gnd	36	Down Not Analyse
17	TRAM 3 Enlace 5 Out	37	Down Not Error
18	TRAM 3 Enlace 5 In	38	NC
19	TRAM 4 Enlace 6 Out	39	NC
20	TRAM 4 Enlace 6 In	40	C011 Enlace Out

**Tabla F11 - Conector de 40 vías**

## **BIBLIOGRAFÍA**

### **REFERENCIAS BIBLIOGRÁFICAS**

[1] R. Bottle

" Modulo de Interfase Programable"

Tesis de Maestría en Sistemas de Manufactura .ITESM.CEM. 1993

[2] G.Lefranc

" La manufactura integrada por computador: Un tutorial "

Automatica e Innovación Vol 1 No 2

[3] Gastón Lefranc H. Guillermo Vidal M. Grupo de Robótica e Inteligencia Artificial.

'SIMULACION DE LA GENERACION DE SECUENCIAS DE ENSAMBLADO EN UNA CELDA FLEXIBLE DE PRODUCCION "

Escuela de Ingeniería Eléctrica. Universidad Católica de Valparaíso, Chile.

[4] A Sinha, A Chaudhuri and P K Das.

" TRANSPUTER IMPLEMENTATION OF FAUL TOLERANT DISTRIBUTED ARCHITECTURE FOR CRITICAL REAL - TIME APPLICATION "

Jadaypur University, India

[5] Tse-yun Feng

" A survey of Interconnection Networks "

The Ohio State University. COMPUTER 1981 IEEE.

[6] Sudhakar Yalamanchili and J. K. Aggarwal.

" Reconfiguration Strategies for Parallel Architectures ".

The University of Texas at Austin. COMPUTER,1985 IEEE.

[7]\_Products overview.

"The T9000 transputer"

SGS Thompson, 1991.

[8] D. Pountain.

"Virtual channels: the next generation of transputers".

Byte, 1990, pp 3-12.

[9] Edited by Gordon Harp;

"Transputer Applications"

Pitman Publishing 1989. Computer Systems Series

[10] Lt Gregory G. Nordstrom, USAF

I / O Interfacing for the transputer.

Arnold Engineering Development Center, Tennessee, IEEE 1991.

[11] INMOS Ltd

" The transputer instruction set: a compiler write's guide "

Prentice Hall 1988

[12] INMOS Ltd

" Introduccion Occam and the Transputer "

Prentice Hall 1988.

[13] Jesús Sánchez

" PARDICO: Un controlador paralelo/distribuido para celdas flexibles de manufactura "

II TALLER INTERNACIONAL DE PROCESAMIENTO PARALELO: UNAM 1996.

[14] Jesús Sánchez

" TECNOLOGÍAS DE COMUNICACIÓN AVANZADAS EN ROBOTICA Y CELDAS FLEXIBLES DE MANUFACTURA "

Propuesta CONACYT- NSF ITESM/CEM/CMOR/UNIVERSIDAD OF TEXAS, 1995

[15] A. Valezano, C. Demartini, L. Ciminiera.

"MAP AND TOP COMMUNICATIONS Standards and Applications " Primera Edición.

[16] Jesús Sánchez Velázquez

" Cómputo Paralelo: un camino hacia la tolerancia a fallas."

2º Simposium Internacional de Informática y Sistemas Computacionales. Hermosillo Sonora, mayo 1996.

[17] Fieldbus Forum

Página de la Red Internet " <http://hallen.ele.k...willi/Fieldbus.html> "

[18] P-NET in General

Página de la Red Internet " <http://www.infoside...a/pnet/p-nete01.htm> "

[19] L. A. Trejo

" Redes de Área Local Apuntes del curso "

ITESM Campus Estado de México. DIC Ciencias Computacionales. Nov 1995

[20] Edgar Morales Sardaneta

" DISEÑO DE INTERFAZ Y CONMUTADOR DE COMUNICACION ENTRE CONTROLADOR Y ELEMENTOS DE UNA CELDA DE MANUFACTURA FLEXIBLE "

Propuesta de Tesis MCC. ITESM CEM 1996.

[21] MICHEL SLATER

" MICROPROCESSOR - BASED DESIGN "

Prentice Hall 1989

[22] TMS320C4x

" User's Guide "

TEXAS INSTRUMENTS

[23] S. S. Mota G.

" PROTOCOLO DE COMUNICACIÓN EN UNA ARQUITECTURA DE TRANSPUTERS PARA CONTROLAR UNA CELDA FLEXIBLE DE MANUFACTURA "

Tesis de Maestría en Ciencias Computacionales .ITESM.CEM/ Campus Toluca.

Mayo 1997

### **BIBLIOGRAFIA CONSULTADA**

- SGS - THOMSON

MICROELECTRONICS

DATA MANUAL 1992.

- INTEL

MICROCONTROLLER

USER MANUAL 1992

- THE TTL

DATA BOOK

VOLUME 2 1990.

- TRANSPUTER EDUCATION KIT

Occam and Transputer a WorkbookCompuert System Architects 1990

- David Krawchik, Delbert Tesar, Pedro Grasa

The Suitability of Multiplexing for Reconfigurable Interfaces in Manufacturing Systems

- Q. Wang and A M S Zalzala

A TRANSPUTER INTERFACE FOR THE PUMA 560 MANIPULATOR

- Dr. Jesús Sánchez Velázquez

Comunicacion en Celdas Flexibles de Manufactura.

ITESM - CEM, Centro de Tecnología en Informática

Apuntes 13 de febrero de 1996

- D.A. Nicole, E.K. Lloyd, J.S. Ward

Transputer link reconfiguration: switching networks for 4 valent graphs.

IEEE PROCEEDINGS, Vol 137, Pt. E, No. 4, JULY 1990.

- ALAN CLEMENTS

Multiprocessors systems

ELECTRONICS and WIRELESS WORLD

The School of Information Engineering at Teesside Polytechnic.

- H. Lecocq and Z. Zheng

Parallel Processing for Real - Time Robot Manipulator Control

Robotique et Automatisation, University of Liege, Rue E. Solvay, 21 - C3

BELGIUM