

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA.

FACULTAD DE INGENIERÍA.

**MAESTRÍA Y DOCTORADO EN CIENCIAS E
INGENIERÍA.**

INGENIERÍA ELÉCTRICA

ÁREA: CONTROL.



**MODULO DE CALIDAD CONTROLADO POR LA
TECNOLOGÍA DEVICE NET.**

PROYECTO DE TESIS

que se presenta para obtener el grado de MAESTRO EN
INGENIERÍA ELÉCTRICA.

GUILLERMO MARTÍN LIMÓN MOLINA.

**DIRECTOR DE TESIS:
M.C. VICTOR NUÑO MORENO.**

MEXICALI BC. A 27 DE OCTUBRE DEL 2007.

Resumen

Resumen de la Tesis de Guillermo Martín Limón Molina, presentada como requisito parcial para la obtención del grado de MAESTRO EN INGENIERIA ELECTRICA. Mexicali, Baja California, México. Diciembre del 2007.

MODULO DE CALIDAD CONTROLADO POR LA TECNOLOGIA DEVICE NET.

Resumen aprobado por:

M.C. Victor Nuño Moreno
Director de tesis

El objetivo principal de este proyecto es la integración de un modulo de calidad a través de adecuación de señal por puerto RS232 entre un robot FANUC 200i y una maquina de medición por Coordenadas Starret Rapid Check.

Este módulo de Calidad debe ser capaz de recibir piezas maquinadas previamente, tomando pieza por pieza para colocarlas en una máquina de medición por coordenadas. La cual realizará el proceso de medición, realizando un análisis comparativo sobre especificaciones (dimensionalmente) y ser capaz de tomar decisiones si la pieza pasó la prueba o no, para disponer del producto final. Todo el proceso debe ser controlado a través de una red industrial tipo Device Net, para lo cual se diseñaron interfaces de comunicación serial por medio de un microcontrolador.

Abstract

Abstract of the thesis, presented by Guillermo Martín Limón Molina, in order to obtain the MASTER ENGINEERING DEGREE in ELECTRIC ENGINEERING. Mexicali, Baja California, México. December 2007.

MODULO DE CALIDAD CONTROLADO POR LA TECNOLOGIA DEVICE NET.

Approved by:

M.C. Victor Nuño Moreno
Thesis Advisor

The primary target of this project is the Integration of module of quality through adjustment of signal by port RS232 between a robot FANUC 200i and a machine of measurement by Coordinates Starret Rapid Check.

This module of Quality must be able to receive machined pieces previously, taking piece by piece to place them in a machine of measurement by coordinates. Which made the measurement process, making a comparative analysis on specifications (dimensionally) and to be able to make decisions if the piece passed the test or no, to have end item. All the process must be controlled through an industrial network type Device Net, for which interfaces of serial communication by means of a microcontroller were designed.

Índice de Figuras

	<u>Página</u>
Figura 1.1. Torno HAAS HL-20.	8
Figura 1.2. Brazo Robótico GE-Fanuc LR MATE 200i.	8
Figura 1.3. Diagrama general de un Sistema Flexible de Manufactura (FMS).	10
Figura 3.1. Robot GE-FANUC-LR MATE 200i.	20
Figura 4.1. Señal de transmisión con Protocolo Hart.	37
Figura 4.2. Vista generalizada de las conexiones de red.	38
Figura 4.3. Cable plano flexible (5 hilos).	40
Figura 4.4. Vista del Kit de Iniciación 7 Device Net.	41
Figura 4.5. Red Device Net configurada en Rs-Networx.	42
Figura 5.1. Demultiplexación de Datos/Direcciones.	45
Figura 5.2. Mapa de memoria.	47
Figura 5.3. Pastillaje del 68HC11.	50
Figura 5.4. Registros del CPU del Microcontrolador 68HC11.	52
Figura 5.5. Registro de banderas CCR.	52
Figura 6.1. Algoritmo de la Red Device Net.	80
Figura 6.2. Diagrama a bloques de la conexión a red Device Net.	87

Índice de Tablas

	Página.
Tabla 4.1. Comparación entre algunos buses y protocolos.	38
Tabla 5.1. Estructura de los puertos.	51
Tabla 5.2. Registros del SCI.	66
Tabla 5.3. Registro de velocidad de transmisión.	66
Tabla 5.4. Selección de velocidades máximas de transmisión y recepción.	67
Tabla 5.5. Relaciones entre las velocidades de transmisión y recepción.	67
Tabla 5.6. Registro de control 1.	67
Tabla 5.7. Registro de Control 2.	68
Tabla 5.8. Registro de estado (SCSR \$102E).	70

Índice.

	<u>Página.</u>
1. Introducción.	7
1.1 Planteamiento del Problema.	8
1.2 Antecedentes.	11
1.3 Objetivos.	12
2. Sistemas de Manufactura Flexibles.	13
2.1 Manufactura Flexible.	13
2.2 Ventajas y Desventajas de la Manufactura Flexible.	13
2.3 Elementos de los Sistemas de Manufactura Flexibles.	14
2.4 Métodos de Transporte.	16
2.5 Funciones de un Sistema de Control de FMS.	17
3. Módulo de Calidad.	18
3.1 Introducción a los Módulos de Manufactura.	18
3.2 Conceptos Básicos de Robótica.	18
3.3 Robot Fanuc 200i.	19
3.3.1 Proceso de Masterización.	20
3.3.2 Programación del Robot.	21
3.4 Metrología de Inspección.	22
3.5 Máquina de Medición de Coordenadas.	22
3.6 Funcionamiento de la Máquina de Medición de Coordenadas.	23
3.6.1 Clearance Moves.	24
3.6.2 Creación de un Programa Sencillo.	25
3.6.3 Asignación de Salida a los Resultados del Programa.	31
3.6.4 Instrucciones de Control de Flujo	31
4. Protocolos de Comunicaciones Industriales.	34
4.1 Introducción a los Protocolos de Comunicaciones Industriales.	34
4.2 Protocolo Hart.	35
4.3 Protocolo Profibus.	36
4.4 Foundation Fieldbus.	36
4.5 Modbus.	37
4.6 Device Net.	37
4.6.1 Beneficios de una red Device Net.	39
4.6.2 Kit de iniciación de Device Net.	40
5. El Microcontrolador.	43
5.1 Antecedentes de los Microcontroladores Genéricos.	43
5.2 Modos de Operación y Mapas de Memoria.	44
5.2.1. Mapas de Memoria.	46
5.3 Hardware del Microcontrolador 68HC11.	48
5.3.1 Pastillaje del MC68HC11.	48
5.3.2 Pines de Entrada/Salida.	51
5.4. Programación del Microcontrolador 68HC11.	51

5.4.1 Registros del CPU.	51
5.4.2 Modos de Direccionamiento.	53
5.4.3 Conjunto de Instrucciones.	55
5.5 Comunicación Serial.	64
5.5.2 Unidad de Transmisión y Recepción.	64
5.5.3 Registros del SCI.	66
6. Integración del Módulo de Calidad.	73
6.1 Caracterización del Robot GE-Fanuc-LR Mate 200i.	74
6.2 Caracterización de la Máquina de Medición de Coordenadas.	76
6.3 Caracterización de la Red Industrial Device Net.	79
6.4 Interfaz de Comunicaciones CMM-Robot.	80
Conclusiones.	88
Referencias.	90

Capítulo 1

Introducción

En este proyecto se propone el desarrollo de un Módulo de Calidad que forma parte de un Sistema de Manufactura Flexible mediante el uso de tecnología Device Net. Básicamente el Sistema de Manufactura consta de 4 elementos los cuales son: 3 módulos de producción (módulo de manufactura, módulo de transporte y módulo de calidad) y un módulo de control (Red industrial Device Net). Cabe mencionar que este proyecto se enfocará al módulo de calidad y de su integración a una red industrial Device Net.

El proyecto a grandes rasgos consiste en crear el sistema antes mencionado que opere en forma automatizada, supervisado y controlado en base a la tecnología Device Net. Como nota adicional tenemos que la tecnología Device Net se encuentra operando aquí en la UABC y se pretende integrar otros elementos a la red (a la fecha solo el módulo de calidad se encuentra integrado a la red).

Un Sistema de Manufactura Flexible es un sistema para programar y controlar automáticamente el proceso de una variedad de productos dependiendo del número de celdas y de los recursos de producción, el total de la operación es controlado por una computadora central.

1.1 Planteamiento del Problema

El problema en general consiste básicamente en la integración de los elementos del Sistema de Manufactura Flexible antes descritos que son: el brazo robótico GE-FANUC-LR MATE 200i, la banda transportadora, la máquina de medición de coordenadas (CMM) y el torno CNC HAAS HL-20.



Figura 1.1. Torno HAAS HL-20



Figura 1.2. Brazo Robótico GE-Fanuc LR MATE 200i

Cabe mencionar que el Sistema de Manufactura Flexible se divide en tres módulos los cuales son: el Módulo de Manufactura, Módulo de Transporte y el Módulo de Calidad. El Módulo de Manufactura consta del torno CNC HAAS HL-20 y un brazo robótico MELFA. El Módulo de Transporte consta de la banda transportadora y de otro brazo robótico. Por último el Módulo de Calidad consta de la máquina de medición de coordenadas (CMM) y del brazo robótico GE-FANUC-LR MATE 200i. El problema de este proyecto se enfoca con automatizar el Módulo de Calidad, que consta de los elementos antes mencionados.

Para resolver este problema se propone el estudio de cada elemento por separado en base a la información con que se cuenta disponible en la UABC, esta información con la que se cuenta son manuales de operación, manuales de instalación y en algunos casos como en el caso del robot y de la máquina de medición de coordenadas (CMM).

Posteriormente se realizaron estudios y pruebas en base a la recientemente introducida tecnología Device Net, en la UABC. En estos momentos se cuenta con kit de iniciación que nos permitirá llevar a cabo este objetivo.

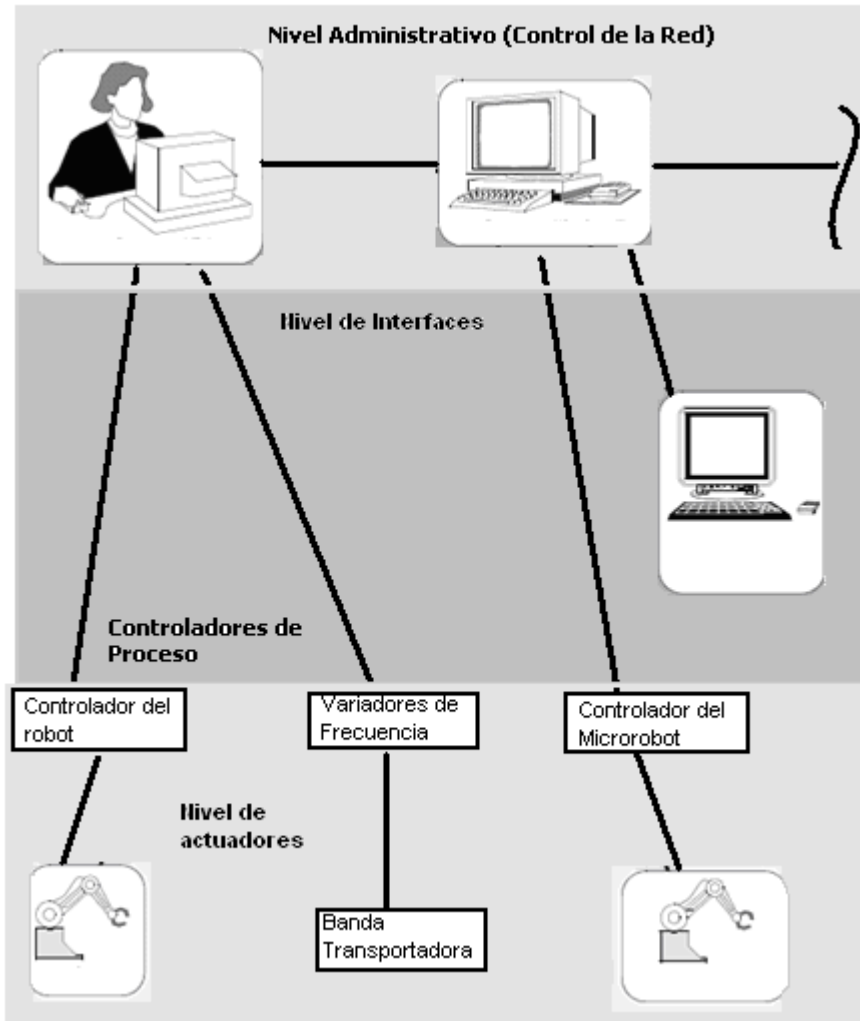


Figura 1.3. Diagrama general de un Sistema Flexible de Manufactura (FMS)

1.2 Antecedentes

Una de las demandas del gobierno del estado es el desarrollo de industria de alta tecnología especialmente en el sector electrónico (televisión) y el sector metal-mecánico (área aeroespacial y automotriz). Es así es como el desarrollo de un Sistema Flexible de Manufactura vendría a apoyar los procesos de automatización de las líneas de producción de este tipo de empresas, en donde se generarán propuestas innovadoras para mejorar la calidad, los tiempos de proceso y reducción de costos.

La necesidad de procesos automatizados para cumplir altos requerimientos y estándares internacionales en este sector requiere el desarrollo y dominio de tecnología de automatización y la robótica. Con esto se pretende elaboración de sistemas con alta tecnología en el estado y fomentar el uso de otras tecnologías para lograr dicho objetivo.

La UABC cuenta con toda una amplia infraestructura que puede dar soporte tecnológico tanto en el desarrollo como en la transferencia de tecnología a la industria manufacturera regional. En este punto cabe aclarar que se cuenta con personal altamente calificado en el soporte de tecnologías.

1.3 Objetivos Específicos

- Caracterizar el brazo robótico GE-FANUC-LR MATE 200i de tal manera que se puedan hacer subrutinas que hagan funciones de transferencia y disposición de producto final.
- Caracterizar la máquina de medición de coordenadas (CMM) para que verifique una pieza determinada dimensionalmente.
- Probar y caracterizar cada uno de los elementos de la tecnología Device Net.
- Integrar los elementos antes mencionados y hacer pruebas piloto de comunicación y control.

Capítulo 2

Sistemas de Manufactura Flexibles

En este capítulo se hablará de conceptos básicos que son necesarios para el entendimiento de esta tesis. Entre los conceptos a definir se hablará de la manufactura flexible, de sus ventajas y desventajas, de los elementos que la integran entre otros conceptos.

2.1 Manufactura Flexible

Manufactura Flexible (FM) es un término que se utiliza para describir una actividad de manufactura que es controlada por computadora para que automáticamente se procesen un número o diferentes tipos de productos.¹

La flexibilidad en el concepto Manufactura Flexible proviene del hecho de que es un sistema automatizado en el cual la mayoría de los elementos son programables; de que esta totalmente equipada y programada para procesar una variedad de productos, a diferencia de una automatización dedicada que solo produce un producto a la vez y que solo es capaz de crear una sola familia de productos. También depende de que es controlado y supervisado por computadora con lo cual se interpreta todo el programa de producción y crea un programa de un producto en específico, dándole la respectiva prioridad y cantidad. Por último la flexibilidad del concepto se debe a que es capaz de crear cualquier producto independientemente de la prioridad que este tenga y la cantidad requerida.

La Manufactura Flexible se clasifica en tres formas diferentes: celdas (FMC), sistemas (FMS) y líneas de transferencia flexibles. La diferencia radica básicamente en el

¹ Administración y Dirección de Operaciones, Chase Aquilano. Mc Graw Hill.

número de partes (productos) que puede manejar, del tamaño y del sistema de transporte a utilizar.

Celdas de Manufactura Flexibles (FMC). Una celda es un grupo de 2 a 6 máquinas que son controladas por una computadora dentro o adyacentes a una celda.

Sistemas de Manufactura Flexibles (FMS). Es un sistema para programar y controlar automáticamente el proceso de una variedad de productos dependiendo del número de celdas y de los recursos de producción, el total de la operación es controlado por una computadora central.

Líneas de Transferencia Flexibles. Son aplicaciones de FMS de alto volumen de producción y son utilizadas en la industria automotriz.

2.2 Ventajas y Desventajas de la Manufactura Flexible

La Manufactura Flexible tiene las siguientes ventajas: gran flexibilidad en el cambio de productos, prioridades y cantidades; incremento de la productividad comparado con el uso individual de máquinas CNC, menos tiempo de muerto por cambio herramientas. Con lo anterior produce una utilización de la maquinaria a un 90 %.

Sus desventajas van del hecho de que cuentan con: alto costo del capital, entre mas grande y complejo mayor es el costo; la inversión en una FM es generalmente una planeación de largo plazo, para obtener una recuperación de la inversión después de 4 o 5 años, su introducción requiere de mucho tiempo de especialistas y gerentes, incluye nuevas prácticas de trabajo y de organización. Y por último las habilidades de uso no se encuentran dentro de la organización, solo pocos saben como utilizar la FM.

El empleo de los Sistema de Manufactura Flexible permite flexibilidad productiva, gestión en tiempo real y acelerado nivel de automatización general, así que una celda en línea es en resumen aceptar el ingreso de materia prima y sacar productos listos para ser ensamblados.

2.3 Elementos de los Sistemas de Manufactura Flexibles

Los Sistemas de Manufacturas Flexibles cuentan con varios elementos entre los que destacan las estaciones de trabajo que son los lugares donde se elabora, se ensambla o subensambla un producto determinado.

También cuentan con pallets, accesorios, cargadores, herramientas entre otros elementos que se definen a continuación:

Elemento	Uso
Estaciones de trabajo	Unidades básicas del sistema.
Pallets	Para cargar materia prima mientras es procesada.
Accesorios	Para sostener subensambles mientras son procesados.
Cargadores	Para cargar pallets y accesorios.
Herramientas	Para sostener herramientas para cada estación de trabajo
Vehículos de transporte	Para mover pallets, accesorios y materiales mientras son procesados.
Almacenes temporales	Se almacenan pallets, accesorios y productos terminados o subensambles.
Estaciones de inspección	Para la medición y pruebas de los productos.

Limpieza	Para limpieza de los productos.
----------	---------------------------------

Computadora de supervisión	Controla el sistema entero.
----------------------------	-----------------------------

Haciendo mención en el último elemento con el objeto de expandir su concepto, se puede comentar que la computadora es elemento principal de un Sistema de Manufactura Flexible, de hecho este elemento es que hace posible su automatización.

Cabe mencionar que un Sistema de Manufactura Flexible puede estar completamente descentralizado, es decir que se pueden tener varios controladores o computadoras individuales de cada proceso, configuradas como maestro, con lo que es posible hacer que un módulo independiente trabaje por si solo, sin la necesidad de que todo el sistema se encuentre operando.

2.4 Métodos de Transporte

Existen dos métodos de enlazar los centros de trabajo a distancia que son más extensos que un brazo robótico son los RGVs y AGVs.

Los RGVs son montados en dos rieles paralelos que corren en línea recta por encima de la superficie del piso. El sistema de rieles son generalmente utilizados muy cerca de los centros de trabajo, y normalmente solo uno o dos vehículos son utilizados debido a que sus movimientos se limitan en una sola trayectoria.

Los AGVs corren directamente en el piso y son guiados por un control que sigue una guía alámbrica enterrada en el piso. Son utilizados en instalaciones lineales y mayormente utilizados en aplicaciones a sistemas de gran escala que se extienden a una gran área de trabajo.

2.5 Funciones de un Sistema de Control de FMS

El programa de control reside en la computadora central y sus funciones son:

1. Aceptar los planes de producción del programa de manufactura principal, como los MRP.
2. Agendar el sistema y establecer las ordenes de trabajo de la manera mas apropiada.
3. Capturar y almacenar el rendimiento, el estado actual y la generación de reportes.
4. Supervisar los sistemas subsidiarios para que todo el proceso se desarrolle adecuadamente.
5. Ejecutar modelos de simulación de la carga de trabajo propuesta.
6. Revisar la red de comunicaciones y las terminales locales del sistema.

Capítulo 3

Módulo de Calidad

3.1 Introducción a los Módulos de Manufactura

El Sistema de Manufactura Flexible (FMS) que se desea implementar en el laboratorio se divide en tres módulos los cuales son: el Módulo de Manufactura, Módulo de Transporte y el Módulo de Calidad. El Módulo de Manufactura consta del torno CNC HAAS HL-20 y un brazo robótico MELFA. El Módulo de Transporte consta de la banda de transportadora y de otro brazo robótico. Por último el Módulo de Calidad consta de la máquina de medición de coordenadas (CMM) y del brazo robótico GE-FANUC-LR MATE 200i. El problema de este proyecto se enfoca con automatizar el Módulo de Calidad, que consta de los elementos antes mencionados.

3.2 Conceptos Básicos de Robótica

Un robot es un dispositivo mecánico capaz de realizar tareas que pueden desempeñar seres humanos², pero en la mayoría de los casos un robot es capaz de realizar dichas funciones con mayor una mayor eficiencia.

Los robots tienen diferentes clasificaciones en general dependen del tipo operación a realizar, otro tipo de clasificación es por el tipo de generación a la que pertenecen, también se pueden clasificar por su arquitectura generacional, por su nivel de inteligencia, por el nivel de control hasta por el lenguaje de programación que utilizan. La clasificación que fue más conveniente en base a su aplicación es la siguiente:

- **Robots manipuladores:** Estos robots están diseñados para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos variados, programados para la ejecución de distintas tareas. Estos tipos de robots se

² <http://www.quizma.cl/servicios/robotica/robot.htm>

caracterizan por tener un funcionamiento repetitivo, preciso, rápido pero cuentan con una percepción limitada.

- **Robots móviles y de servicios:** Son robots que cuentan con un incremento de autonomía: sistema de navegación automática. Generalmente estos robots perciben el medio, planifican y actúan con mínima o una nula intervención humana.
- **Plataforma móvil + manipulador:** Estos robots cuentan con gran autonomía, estos cuentan con una plataforma y un manipulador. Ejemplo de estos se utilizan en grúas, transporte, asistencia y minería.
- **Telerobots:** A este tipo de robots el hombre los manipula es decir realiza su percepción y su planificación. No se les considera como un tipo de robots.³

3.3 Robot Fanuc 200i

Consiste en un brazo robótico que tiene movimientos en 6 ejes principales. En cada uno de los 6 ejes puede realizar diferentes tipos de movimientos, la combinación entre ellos nos da como resultado que el brazo robótico pueda moverse en tres dimensiones dependiendo de cada aplicación. En el último eje es de la pinza en este nada mas se puede abrir y cerrar la pinza.

El brazo robótico cuenta dispositivos de entrada y salida, los cuales pueden ser utilizados para distintas funciones. Estas funciones para las entradas por ejemplo pueden incluir sensores que restrinjan ciertos movimientos del robot y para las salidas se pueden incluir indicadores o alarmas cuando se cumplan ciertas condiciones, al terminar un ciclo normal de operación por ejemplo, entre muchas otras aplicaciones.

³[http://icaro.eii.us.es/descargas/RI%202005%20transparencias%20tema%201\(4%20por%20pag\).pdf](http://icaro.eii.us.es/descargas/RI%202005%20transparencias%20tema%201(4%20por%20pag).pdf)

Además el brazo robótico cuenta con un “teach pendant” o panel del operador que es un dispositivo que despliega las opciones del software que lo controla llamado Handling Tool.

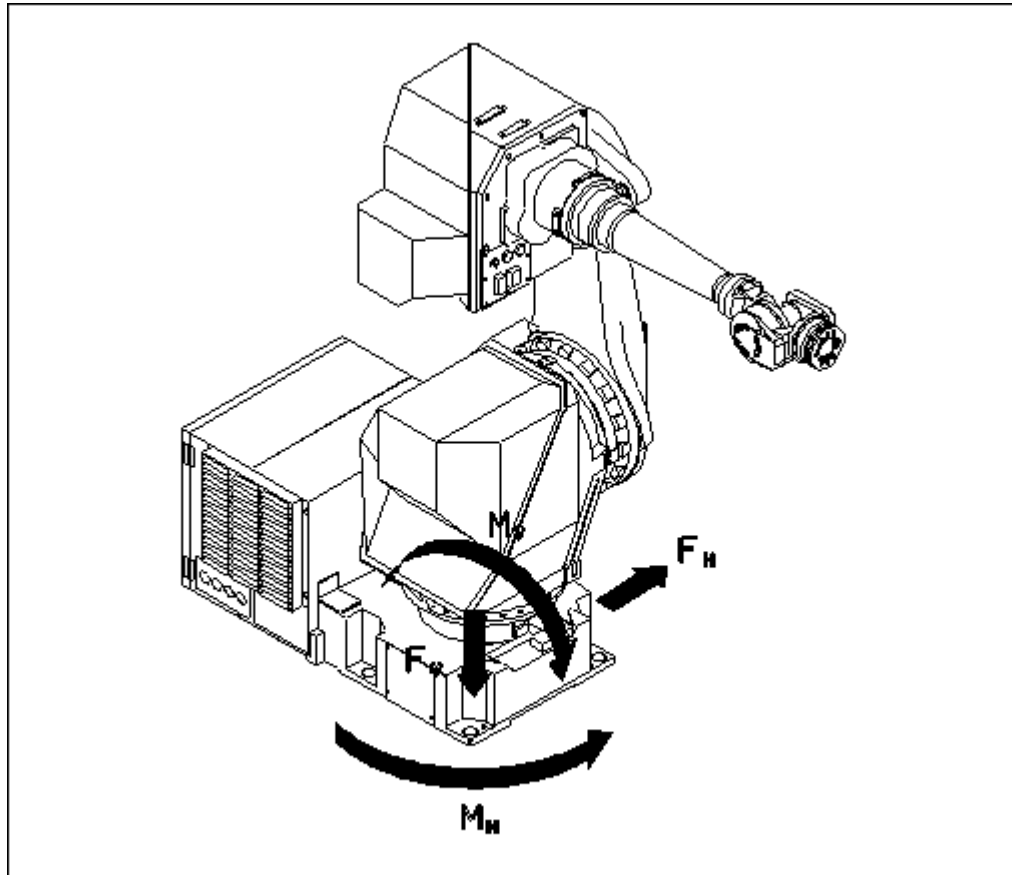


Figura 3.1. Robot GE-FANUC-LR MATE 200i

Se usa para mover el robot, crear, editar y probar programas, revisar el estado del sistema y realizar funciones manuales. Cabe mencionar que la única manera de establecer una rutina de movimiento es a través de la creación de un programa, por este medio, o por medio de software a través de la computadora.

3.3.1 Proceso de masterización

Consiste en ajustar los ceros del Robot Fanuc, esto se logra con mover los 6 ejes del robot a cada una de las marcas que indican el origen en cada eje respectivo.

Este proceso, se lleva a cabo cuando se adquirió el Robot Fanuc y es necesario, repetirse cada vez que las baterías del sistema son reemplazadas. Esto significa que cada vez que al Robot se le descargan las baterías, se pierden los datos del origen de cada uno de los ejes.

Cuando no se realiza el proceso de masterización, no es posible realizar ninguna operación con el Robot Fanuc, es por tal razón que es tan importante.

3.3.2 Programación del robot

Un programa de aplicación es una combinación de instrucciones que, cuando se ejecutan en secuencia, completarán una tarea. El software Handling Tool permite crear y modificar un programa que consta de:

- **Encabezado:** Incluye el nombre del programa, comentarios y el tipo de programa.
- **Instrucciones de movimiento:** Son comandos que le dicen al robot como y donde moverse.
- **Instrucciones de programa:** Para usar las entradas y salidas, funciones de salto, de espera, y funciones avanzadas.

Una instrucción de movimiento dirige al robot para que se mueva de una manera específica hacia una determinada posición dentro del área de trabajo a una velocidad dada. Dicha instrucción de movimiento incluye: el tipo de movimiento, información posicional, tipo de terminación y la velocidad. (Para ver un ejemplo de esto, ver el capítulo de los resultados).

Existen tres tipos de movimientos los cuales son el articulado (joint), circular y lineal. El movimiento articulado mueve todos los ejes a la vez y mueve el robot hasta llegar a la posición destino. El movimiento circular, describe un arco cuando se traslada de un punto a otro (si se usan varios de estos es posible describir una circunferencia. Y por último el movimiento lineal, describe una línea recta desde el punto inicial hasta el punto destino.

Por otro lado las instrucciones del programa permiten activar o desactivar señales de salida, así como habilitar una lectura de un canal de entrada. También dentro de este grupo se encuentran las condicionales (if), así como las instrucciones de salto y etiquetado.

3.4 Metrología de Inspección

La medición es un procedimiento en el cual una cantidad desconocida es comparada a un estándar conocido, utilizando un sistema de unidades aceptado y consistente. Se involucran mediciones de longitud, fuerza y tensión.

La Metrología es la ciencia de la medición. Involucra 7 cantidades básicas las cuales son: longitud, masa, tiempo, corriente eléctrica, temperatura, intensidad luminosa y materia.

De las cantidades mencionadas, se derivan otras cantidades físicas como: área, volumen, velocidad, aceleración, fuerza, voltaje, energía, etc.

En la metrología de manufactura, lo que queremos medir del producto es su longitud, anchura, profundidad, diámetro, dureza, rugosidad de superficie, etc.⁴

3.5 Máquina de Medición por Coordenadas

La Máquina de Medición por Coordenadas (CMM) puede ser definida como “una máquina que emplea tres componentes móviles que se trasladan a lo largo de guías con recorridos ortogonales, para medir una pieza por determinación de las coordenadas X, Y y Z de los puntos de la misma con un palpador de contacto o sin él y sistema de medición del desplazamiento (escala), que se encuentran en cada uno de los ejes”. Como las mediciones están representadas en el sistema tridimensional, la CMM puede efectuar diferentes tipos de medición como: dimensional, posicional, desviaciones geométricas y mediciones de contorno.

Los procedimientos de medición y procesamiento de datos de las CMM, poseen una serie de características que se describen a continuación: Primeramente se tiene un sistema

⁴ <http://www.intelmax.com/ensanluis/images/capitulo23.ppt>

de posicionamiento que provoca que el palpador alcance cualquier posición en X, Y o Z; este sistema de posicionamiento es accionado a través de unos motores, que a su vez, poseen unos codificadores ópticos rotatorios, los que producirán una señal adecuada para activar un contador que incrementa su número en relación a la posición del eje con respecto de su origen.

En este sistema como en otros es de primordial importancia la existencia de un origen para poder determinar la posición.

El sistema dispondrá además de un palpador que al ser accionado, hará que los datos del contador del sistema de posicionamiento sean trabajados por la unidad principal de la CMM y sean transformados en coordenadas X, Y y Z y además se apliquen las fórmulas programadas para después desplegar los datos en una pantalla de cristal líquido.

El sistema también posee una palanca de control que acciona directamente los servomotores provocando un desplazamiento manual de cada uno de los ejes.

Este sistema CMM en particular posee teclado para introducción de datos, un monitor que proporciona la visualización de ellos ya sea que se introduzcan o se generen por la CMM.

Como se mencionó anteriormente el palpador que se encuentra en el extremo inferior del eje Z, se acciona al toque de la pieza que se desea medir.⁵

3.6 Funcionamiento de la Máquina de Medición de Coordenadas

La CMM consta de una serie de movimientos que caracterizan su funcionamiento. El primero de ellos, y uno de los más importantes es el movimiento de **Auto Home**, el cual consiste en la búsqueda automática de los ceros físicos de la máquina.

⁵<http://www.comflyer.com/strg/docs/j19ho55dcnq.pdf>

Esto se logra por medio de unos sensores magnéticos que se activan cuando se ha alcanzado el fin de carrera en cada eje respectivo, una vez alcanzado este fin de carrera, la CMM esta lista para usarse.

Otros de los movimientos importantes se denominan **Clearance Moves**, que simplemente mueven la máquina en una posición segura cada vez que se realice una medición y así se asegura que la CMM no choque en un punto no deseado. A continuación se explica más a detalle en que consiste este tipo de movimientos.

3.6.1 Clearance moves

Al realizar un programa el software QC-5000 crea caminos directos entre una posición y otra. Este camino no considera que haya obstrucciones entre las trayectorias. Por tal motivo es deber del operador el determinar si hay obstáculos en el camino de la sonda y programar un camino seguro alrededor de dichos obstáculos.

Para lograr esto es necesario usar el botón de **goto here** para crear clearance moves o caminos en un programa. Se requiere crear este tipo de movimientos cuando existan obstáculos en cada trayectoria definida. A veces es necesario que se creen más de un clearance move para desplazarse entre algunos obstáculos.

- **Para crear un clearance move**

Paso 1

Retraiga la sonda de fuera de la zona del obstáculo y déjelo justo arriba de este. Asegúrese que el eje Z este lo suficientemente elevado para permitir que la sonda viaje libremente por los ejes X y Y. Presione el botón **goto here** en la barra de herramientas de program.

Paso 2

Posicione la sonda sobre el siguiente obstáculo. Presione el botón de **goto here** en la barra de herramientas de program.

Paso 3

Continué con la inspección.

3.6.2 Creación de un programa sencillo

Un programa es la forma de definir de manera automática una medición de una pieza específica. Por tal motivo si se requiere automatizar una medición es necesario entender como se efectúa un programa. A continuación se describe como se elabora un programa sencillo.

Paso 1

Posicionamiento de la sonda sobre la parte deseada. Utilice el joystick para mover la sonda en una posición sobre la parte deseada. Seleccione ***new*** de ***file pull-down menu***. Seleccione ***part*** del submenú. Presione el botón ***record/edit*** en la barra de herramientas de program.

Paso 2

Presione el botón ***goto here*** en la barra de herramientas de program.

Paso 3

Medición del plano principal. Presione el botón de ***primary plane*** en la barra de herramientas de datum. Sondee cuatro puntos en la superficie del plano principal. Un mínimo de tres puntos son requeridos para medir un plano. Presione ***OK*** en el cuadro de diálogo después de sondear el cuarto punto. Utilice el joystick para desplazar la sonda en una posición segura cerca de la línea secundaria. (Plano secundario, parte frontal de la pieza). Agregue clearance moves si son necesarios.

Paso 4

Medición de la línea secundaria (plano secundario o parte frontal). Presione el botón de ***secondary line*** en la barra de herramientas de datum. Sondee al menos dos puntos a lo largo de la línea secundaria. Presione ***OK*** en el cuadro de dialogo después del último punto. Agregue clearance moves si es necesario.

Paso 5

Medición de una línea. (Para medir la parte lateral de la pieza). Presione el botón de **line button** en la barra de herramientas de measure. Sondee al menos dos puntos para medir una línea. Agregue clearance moves de ser necesario.

Paso 6

Construcción del punto cero. Seleccione **secondary line** y **line 3** en la ventana de part view o features list. Presione el botón de **zero point** en la barra de herramientas de datum. Marque **auto zero** en el cuadro de dialogo y luego presione **OK**. Con esto el marco de referencia queda establecido. Presione en el botón **save reference frame** en la barra de herramientas de datum.

Al guardar el marco de referencia agrega el paso de **registration complete** en el programa. Este paso permite que la CNC se pueda mover una vez que el marco de referencia ha sido establecido si y solo si una part fixture permanente no esta siendo utilizada.

Teclee un nombre para el marco de referencia en el cuadro de dialogo. Ahora el cuadro de referencia ha sido grabado.

Paso 7

Medición de un cilindro utilizando la medición automática. Presione el botón de **cylinder** en la barra de herramientas de measure. Presione el botón de **create** en el cuadro de dialogo. Presione el botón de **auto** en la ventana de create cylinder. No es necesario agregar datos en esta ventana. Posicione la sonda sobre la parte superior del cilindro y presione **OK** en el cuadro de dialogo.

Utilice el teclado para acceder valores nominales de X, Y y Z en la parte que dice **axis start point** y en **axis end point** del cilindro. Accese el diámetro nominal en el cuadro de texto de **size**. Presione en el botón de la izquierda para cambiar de diámetro a radio. Seleccione **measure as a bore**.

Accese el número deseado de puntos en el cuadro de point to probe. Una slice incluye todos los puntos (normales al eje) encontrados alrededor de un punto en el eje de la figura. Accese el número de slices en el cuadro de texto **number of slices**. Esto significa puntos que serán sondeados alrededor de los puntos en el eje X del cilindro. Ejemplo de lo anterior si se tienen 5 puntos por slice y 3 slices en total se sondearan 15 puntos.

Agregue la longitud de la sonda a utilizar en el cuadro de texto de *usable tip length*. Esto se usa para determinar la profundidad que se puede sondear un cilindro sin ninguna interferencia de la cabeza del mecanismo. No sondee más allá de esta profundidad.

Ingrese la profundidad de inicio deseada para la medición del cilindro en el cuadro de texto de *depth*. Esta cantidad representa el punto de inicial por encima del eje nominal de *axis end point* o por debajo del eje nominal de *axis start point*. Presione **OK** en el cuadro de dialogo.

QC-5000 posicionará la sonda para realizar la medición. Presione **OK** en el cuadro de dialogo. Después de esto el programa preguntará que si es seguro continuar, de no haber obstrucciones en el camino, presione **Yes**. Con esto el cilindro será medido automáticamente. Agregue clearance moves que sean necesarios para continuar con el programa.

Paso 8

Selección del cilindro en la ventana de *part view*. Utilice el botón derecho del mouse en la ventana de resultados y seleccione *tolerance*. Seleccione *perpendicularity* del sub-menú. Ahora se mostrará el cuadro de dialogo de la tolerancia.

Ingrese una zona deseada (para el ejemplo son 0.005 pulgadas). Para llevar a cabo la tolerancia de perpendicularidad, el programa QC-5000 construye una línea perpendicular teórica que pasa exactamente por el centro y a partir de esa línea se hacen las mediciones, en base a las comparaciones de las distancias entre esta línea y el cilindro. Si la prueba excede la zona de tolerancia la prueba falla, si es menor o igual esta prueba de tolerancia es exitosa y significa que pieza se encuentra dentro del rango de aceptación.

Utilice el plano 1 como característica de referencia. Ingrese 0 y 1 como el mínimo y máximo respectivamente para la *projected zone*. La zona proyectada es el área por abajo o por debajo de la proyección del plano de referencia donde la medición de tolerancia debió ser exitosa. Presione **OK** en el cuadro de dialogo de la tolerancia. Ahora se mostrará la ventana de resultados.

Paso 9

Construya el punto de perforación cilíndrica en el eje y plano respectivos. Seleccione el cilindro y el plano en la ventana de *part view*. Presione el botón de *point* en

la barra de herramientas de measure. Presione **OK** en el cuadro de dialogo, al hacer esto la perforación cilíndrica es construida.

Paso 10

Medida de dos círculos pequeños. Presione el botón de **circle** en la tabla de measure. Sondee al menos tres puntos para cada círculo. Agregue los clearance moves que sean necesarios para continuar con el programa. Presione **OK** en el cuadro de dialogo. Con esto aparecerán las características de los círculos en la ventana de **part view**.

Utilice la característica de **auto re-measure** para sondear puntos adicionales en el segundo círculo. Seleccione el segundo círculo en la lista de características. Seleccione **change feature** desde el menú de edit. Accese el número total de puntos adicionales y seleccione la característica de **remeasure feature** antes de continuar.

Paso 11

Elabore a una tolerancia de posición real para el primer círculo. Seleccione el primer círculo en la lista de características. Utilice el botón derecho del mouse en la ventana de resultados y seleccione **tolerance**. Seleccione **true position** en el submenú.

Ingrese los valores nominales en X y Y de la posición del centro del círculo (la posición Z no es requerida). Ingrese la zona tolerancia deseada en la cual el círculo debe de existir.

Ingrese el diámetro nominal en el cuadro de texto de **size**. Presione el botón de la derecha para cambiar de diámetro a radio y viceversa. Ingrese la tolerancia deseada en los cuadros **nom –** y **nom +**.

Utilice la función de **named tolerance** para guardar valores que pueden aplicar a más de una característica. Por ejemplo, dos pequeños círculos, que utilicen los mismos valores para la tolerancia. Almacene estos datos como una **named tolerance** para aplicarlos en otras mediciones. Presione el botón de **save** después de que estos datos hayan sido accesados.

Ingrese un nombre para esta tolerancia y presione **OK** en el cuadro de dialogo de **named tolerance**. Presione **OK** en el siguiente cuadro de dialogo para finalizar, con esto se mostrará la ventana de resultados, presione **OK** continuar.

Ahora elabore la tolerancia de posición real para el segundo círculo utilizando el archivo creado (named tolerance). Seleccione el círculo en la lista de características y repita

los pasos anteriores. Ingrese los datos en X y Y y luego selecciona el nombre del archivo previamente guardado en ***named tolerance***. (Para este caso se utiliza el archivo que se creó para el primer círculo).

Presione **OK** hasta que el programa continúe.

Paso 12

Medida de distancia entre dos objetos (en este caso círculos) y elaboración de la tolerancia de anchura. Seleccione los dos objetos deseados en la lista de características (por ejemplo los dos círculos que se midieron en el paso anterior). Presione el botón de ***distance*** en la barra de herramientas de measure.

Ahora seleccione la distancia resultante en el listado de características. Use el botón derecho del mouse, y seleccione ***tolerance***. Después seleccione ***width*** desde el submenú.

Ingrese la longitud nominal y/o la distancia a recorrer en X, Y y Z. Una vez proporcionados estos datos, se mostrará la ventana de los resultados. Presione **OK** para continuar.

Paso 13

Medida de un cono. Posicione la sonda sobre el cono a medir. Agregue los clearance moves que sean necesarios, para mover la sonda libremente. Presione el botón de ***cone*** en la barra de herramientas de measure.

Sondee al menos seis puntos (tres en la parte superior y tres del fondo). Presione **OK** en el cuadro de diálogo. Las características del cono se mostrarán en la ventana de ***part view***. Agregue clearance moves como sean necesarios.

Paso 14

Cambio de posición de la sonda. Mueva la sonda a una posición segura para evitar colisiones antes de que la sonda cambie de posición. Utilice el botón de ***goto here***. Seleccione ***contact probes*** del menú de probes. Ahora seleccione ***probe selector*** del submenú. Seleccione la posición de la sonda A15_B90.

Paso 15

Construcción de una angularidad y elaboración una tolerancia de angularidad. Medición del plano angulado. Utilice el botón de ***plane*** en la barra de herramientas de

measure. Sondee al menos tres puntos en el plano. Ahora presione **OK** para terminar con la medición. Ahora utilice el plano primario y el plano que acaba de medir para crear un ángulo. Seleccione ambos planos en la ventana de **part view**. Utilice el botón de **angle** en la barra de herramientas de measure. Presione **OK** en el cuadro de dialogo para terminar con la construcción del dibujo. Agregue clearance moves como sean necesarios.

Seleccione el ángulo resultante en la ventana de part view. Utilice el botón derecho del mouse en la ventana de resultados. Seleccione **tolerance** del menú y **angularity** en el submenú. Ingrese una zona de tolerancia deseada. Ingrese “0” y “1” como mínimo y máximo para la **projected zone**. Presione **OK** en el cuadro de dialogo. Con esto la ventana de resultados aparecerá. Presione **OK** para continuar.

Paso 16

Creación de un segundo marco de referencia. Seleccione el plano que se midió anteriormente en la ventana de part view. Seleccione el botón de **primary plane** en la barra de herramientas de datum y presione OK en el cuadro de dialogo de measurement. Al realizar esta acción este plano angulado queda definido como plano principal en la ventana de part view.

Utilice la opción de **part following** para desplazar con el joystick la sonda hacia el nuevo plano primario (principal). Seleccione **joystick** desde el menú de tools y seleccione **part following** del submenú. Si aparece palomeada esta opción significa que ya ha sido activada, de no ser así habrá que activarla.

Utilice el botón de **secondary line** en la barra de herramientas de datum. Sondee al menos dos puntos, para medir la línea o la distancia antes del hueco de la pieza, en la parte angulada. Presione **OK** en el cuadro de dialogo. Con esto aparecerán los resultados. Agregue clearance moves como sean necesarios.

Ahora utilice la medición de otra línea secundaria para medir la parte lateral de la pieza y con esta medición seleccione la primera línea medida y la segunda para crear un punto de unión. Para seleccionar estas dos líneas presione la tecla Ctrl. Agregue clearance moves como sean necesarios.

Seleccione el punto resultante de la unión desde la ventana de part view. Presione el botón de **zero** de la barra de herramientas de datum. Seleccione **auto-zero** en el cuadro de dialogo y presione **OK**.

Ahora guarde los cambios del cuadro de referencia con la opción de *save reference frame* en la barra de herramientas de datum. Asígnele un nombre y presione **OK**. Con esto el segundo cuadro de referencia ha sido almacenado.

Paso 17

Medición de un segundo cilindro (cilindro 2). El cilindro a medir esta a un ángulo de 15 grados y esta separado por una apertura de 0.25 pulgadas. Para este caso será necesario sondear el cilindro manualmente con la opción de *part following* activada.

Sondee al menos 6 puntos (3 en la parte superior y 3 en el fondo). Presione **OK** en el cuadro de dialogo. Agregue clearance moves como sean necesarios.

3.6.3 Asignación de salida a los resultados del programa

Utilice la opción de *report template* para asignar un formato a la salida del programa. Con esto se asigna un formato imprimible para los resultados obtenidos del programa.

Para crear un reporte nada más basta con seleccionar los datos deseados en la ventana de resultados y arrastrar los campos deseados al *report template*.

Si se desea exportar resultados en tiempo real mientras un programa este corriendo, es necesario utilizar la opción de DDE output y una vez estando ahí se selecciona el formato deseado para dar una salida a los resultados obtenidos.

3.6.4 Instrucciones de control de flujo

Ciclos. Se utilizan para repetir un paso o una serie de pasos. Los ciclos son útiles cuando se utilizan variables y etiquetas.

- **Para insertar un paso ciclado**

Paso 1

Seleccione los pasos deseados y asegúrese de que se este grabando un programa usando el botón de *record/edit*.

Paso 2

Presione el botón derecho del mouse y seleccione *flow control steps* en el menú. Seleccione *loop* en el sub menú.

Paso 3

Utilice el cuadro de diálogo de *loop statement*. Escriba un nombre para el ciclo y en el siguiente campo ingrese las veces en que desea que este ciclo se repita. Presione **OK**.

Variables. Cuando un programa es ejecutado el usuario responde a cuatro cuadros de diálogo que definen los siguiente parámetros: X offset, X ítems, Y offset y Y ítems. En pocas palabras se pueden definir desplazamientos entre pieza y pieza por medio de variables (una vez que se encuentren paletizadas) y el número de veces que se repite un ciclo.

Para agregar una variable a un programa es necesario que se encuentre el botón *record/edit* activado. Usando el botón derecho del mouse sobre el paso actual, seleccione *special steps* en el menú. Seleccione *set variable* en el sub menú.

En el cuadro de diálogo se definen el nombre de la variable, el tipo de medida; si la variable corresponde a un número que indica algún desplazamiento se activa la opción de inch/mm y si es un número que indica las veces en que se va a repetir un ciclo se utiliza fixed que representa a una medida adimensional. Y por último se define la pregunta o oración que se va a desplegar cuando se requiera que se de acceso a la variable.

Condiciones de prueba. Este tipo de condiciones generalmente llevan un razonamiento aritmético asociado a la medición de una cierta característica. Por ejemplo si un círculo tiene un diámetro mayor que el medido, entonces ejecuta una determinada acción.

Dentro de estas condiciones se encuentran las instrucciones if-goto, labels (etiquetas) goto label, mensajes de usuario, entre otras.

Una instrucción *if-goto* se utiliza cuando se realiza una prueba y dependiendo del resultado que arroje esta, el programa hace un salto de una etiqueta a otra. La instrucción *goto label* realiza un salto incondicional a una etiqueta deseada. Y por último los *user messages* son mensajes que el usuario define después de que se haya realizado una prueba en específico.

- **Para realizar un programa con condiciones de prueba**

Paso 1

Agregue la instrucción deseada. Haga click en el botón derecho del mouse y seleccione ***flow control steps*** del menú. Seleccione la instrucción deseada desde el sub menú.

Escriba la fórmula en el cuadro de diálogo, generalmente esta se escribe entre corchetes, seguida de un punto y la condición a medir, después va la condición deseada. Por ejemplo [Circle 6].A>.02, esto quiere decir si el área del círculo 6 es mayor que 0.02 realizará un salto, otra medición e inclusive prueba, dependiendo de que instrucción se trate.

Si se trata de una instrucción ***if-goto***, el siguiente paso es escribir el nombre de la etiqueta a la que se pretende realizar el salto.

Paso 2

Agregar una instrucción ***goto label***. Use el botón derecho del mouse y seleccione ***flow control steps*** del menú. Ahora seleccione ***goto label*** en el submenú. Tecleé el nombre de la etiqueta deseada.

Paso 3

Se procede ha agregar una etiqueta de usuario (***user label***), utilizando el botón derecho del mouse y seleccione ***flow control steps*** del menú. Seleccione ***label*** en el sub menú y agregue el nombre deseado.

Paso 4

Agregar un mensaje de usuario (***user message***) utilizando el botón derecho del mouse y seleccione ***special steps*** del menú. Seleccione ***user message*** en el sub menú. El texto puede ser “pieza mala” o la “prueba falló”.

Paso 5

Agregue otra etiqueta. Utilizando el botón derecho del mouse y seleccione ***flow control steps*** del menú. Seleccione ***label*** en el sub menú y agregue el nombre deseado. Esta otra etiqueta será la que indique la prueba fue exitosa, y el programa tome una acción diferente a la anterior.

Capítulo 4

Protocolos de Comunicaciones Industriales

4.1 Introducción a los Protocolos Industriales

Un importante número de empresas en nuestro país presentan la existencia de celdas de trabajo automatizadas, siendo en estos casos las redes y los protocolos de comunicación industrial indispensables para realizar un enlace entre las distintas etapas que conforman el proceso.

La irrupción de los microprocesadores en la industria ha posibilitado su integración a redes de comunicación con importantes ventajas, entre las cuales figuran: mayor precisión derivada de la integración de tecnología digital en las mediciones, mayor y mejor disponibilidad de información de los dispositivos de campo, diagnóstico remoto de componentes.

La integración de las mencionadas islas automatizadas suele hacerse dividiendo las tareas entre grupos de procesadores jerárquicamente anidados. Esto da lugar a una estructura de redes industriales, las cuales es posible agrupar en tres categorías:

- Buses de campo.
- Redes LAN.
- Redes LAN-WAN.

En esta sección se hará referencia a los protocolos de comunicación más usados en la industria. Los buses de datos que permiten la integración de equipos para la medición y control de variables de proceso, reciben la denominación genérica de buses de campo.

Un bus de campo es un sistema de transmisión de información (datos) que simplifica enormemente la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción.

El objetivo de un bus de campo es sustituir las conexiones punto a punto entre los elementos de campo y el equipo de control a través del tradicional lazo de corriente de 4 - 20mA o 0 a 10V DC, según corresponda. Generalmente son redes digitales, bidireccionales, multipunto, montadas sobre un bus serie, que conectan dispositivos de campo como PLC's, transductores, actuadores, sensores y equipos de supervisión.

Varios grupos han intentado generar e imponer una norma que permita la integración de equipos de distintos proveedores. Sin embargo, hasta la fecha no existe un bus de campo universal.

Los buses de campo con mayor presencia en el área de control y automatización de procesos son:

- HART.
- Profibus.
- Foundation Fieldbus.

Otros protocolos ampliamente usados aunque de menor alcance son:

- Modbus.
- Device Net.

Cabe mencionar que este trabajo se enfoca a las redes Device Net, por tal motivo se hablará con más detalle de este tipo de redes, así como sus beneficios y ventajas sobre los demás tipos de redes. En el capítulo 6 se hablará de uso de este tipo de redes y su integración a un módulo que forma parte de un sistema de manufactura flexible.

4.2 Protocolo Hart

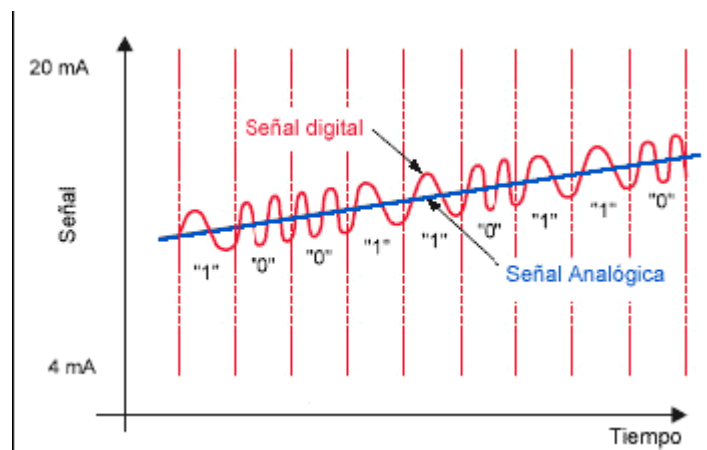


Figura 4.1. Señal de transmisión con Protocolo Hart.

El protocolo HART (High way-Addressable-Remote-Transducer) agrupa la información digital sobre la señal analógica típica de 4 a 20 mA DC. La señal digital usa dos frecuencias individuales de 1200 y 2200 Hz, que representan los dígitos 1 y 0 respectivamente y que en conjunto forman una onda senoidal que se superpone al lazo de corriente de 4-20 mA.

Como la señal promedio de una onda senoidal es cero, no se añade ninguna componente DC a la señal analógica de 4-20 mA., lo que permite continuar utilizando la variación analógica para el control del proceso.

4.3 Protocolo PROFIBUS

(Process Field Bus) Norma internacional de bus de campo de alta velocidad para control de procesos normalizada en Europa por EN 50170.

Existen tres perfiles:

- Profibus DP (Decentralized Periphery). Orientado a sensores/actuadores enlazados a procesadores (PLCs) o terminales.
- Profibus PA (Process Automation). Para control de proceso, cumple normas especiales de seguridad para la industria química (IEC 1 1 15 8-2, seguridad intrínseca).
- Profibus FMS (Fieldbus Message Specification). Para comunicación entre células de proceso o equipos de automatización.

4.4 Foundation Fieldbus

Foundation Fieldbus (FF) es un protocolo de comunicación digital para redes industriales, específicamente utilizado en aplicaciones de control distribuido. Puede comunicar grandes volúmenes de información, ideal para aplicaciones con varios lazos complejos de control de procesos y automatización.

Está orientado principalmente a la interconexión de dispositivos en industrias de proceso continuo. Los dispositivos de campo son alimentados a través del bus Fieldbus cuando la potencia requerida para el funcionamiento lo permite.

4.5 Modbus

Modbus es un protocolo de transmisión para sistemas de control y supervisión de procesos (SCADA) con control centralizado, puede comunicarse con una o varias Estaciones Remotas (RTU) con la finalidad de obtener datos de campo para la supervisión y control de un proceso. Las interfaces de capa física puede estar configuradas en: RS-232, RS-422, RS-485.

En Modbus los datos pueden intercambiarse en dos modos de transmisión:

- Modo RTU
- Modo ASCII

4.6 Device Net

Una red Device Net consiste en una rama o bus principal de hasta 500 metros con múltiples derivaciones de hasta 6 metros cada una, donde se conectan los diferentes dispositivos de la red. En cada red Device Net se pueden conectar hasta 64 nodos y cada uno puede soportar un número infinito de entradas y salidas aunque lo normal son 8, 16 ó 32, es decir un 2ⁿ entradas y salidas.

Una red Device Net provee conexiones entre dispositivos industriales simples, primarios, discretos y también controladores. Las conexiones pueden existir sin la necesidad de módulos de entrada y salida o bloques de los mismos.

La red Device Net consta de múltiples elementos, pero en general se compone de los siguientes: módulos de entrada y salida (o blocks de entrada y salida), sensores (de diferentes tipos, los cuales pueden ser resistivos, inductivos, capacitivos, entre otros), diferentes tipos de cables, escáner y un software apropiado. En la figura 4.2 se muestra un ejemplo genérico de una red Device Net.

Vease tabla 4.1, para comparar los tipos de redes industriales.

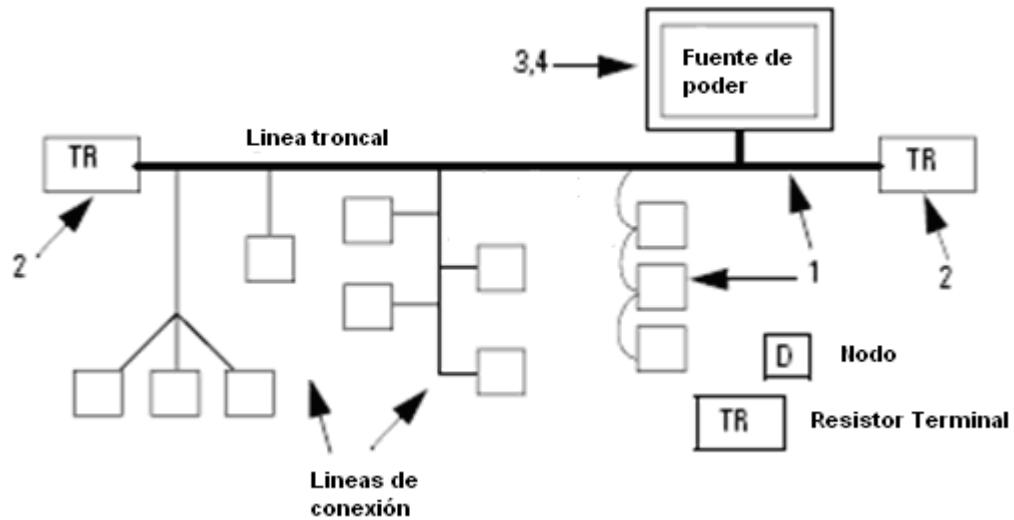


Figura 4.2. Vista generalizada de las conexiones de red

Tabla 4.1. Comparación entre algunos buses y protocolos.

Nombre	Topología	Soporte	Max. dispositivos	Vel. de trans. bps	Distancia max. en Km	Comunicación
Profibus DP	Línea, estrella y anillo.	Par trenzado, fibra óptica	127/ segm	Hasta 1.5 M y 12 M	0.1 segm 24 fibra	Maestro/Esclavo punto a punto
Profibus PA	Línea, estrella y anillo.	Par trenzado, fibra óptica	14400/segm	31.5 K	0.1 segm 24 fibra	Maestro/Esclavo punto a punto
Profibus FMS		Par trenzado, fibra óptica	127/segm	500 K		Maestro/Esclavo punto a punto
Foundation Fieldbus HSE	Estrella	Par trenzado, fibra óptica	240 p/segm 32.768 sist	100 M	0.1 par 2 fibra	Sencillo/Multi-maestro
Foundation Fieldbus H1	Estrella o bus	Par trenzado, fibra óptica	240 p/segm 32.768 sist	31.25 K	1.9 cable	Sencillo/Multi-maestro
LonWorks	Bus, anillo, lazo, estrella	Par trenzado, fibra óptica, coaxial, radio	32768/dom	500 K	2	Maestro/Esclavo punto a punto
Interbus-S	Segmentado	Par trenzado, fibra óptica	256 nodos	500 K	400/segm 12.8 total	Maestro/Esclavo
DeviceNet	Troncal/puntual c/bifurcación	Par trenzado, fibra óptica	2048 nodos	500 K	0.5 6 c/repetidores	Maestro/Esclavo, Multi-maestro, punto a punto
AS-I	Bus, anillo,	Par trenzado	31 p/red	167 K	0.1, 0.3	Maestro/Esclavo

	árbol, estrella				c/repetidores	
Modbus RTU	Línea, estrella, árbol, red con segmentos	Par trenzado, coaxial, radio	250 p/segm	1.2 a 115.2 K	0.35	Maestro/Esclavo
Ethernet Industrial	Bus, estrella, malla-cadena	Coaxial, par trenzado, fibra óptica	400 p/segm	10, 100 M	0.1, 100 mono c/switch	Maestro/Esclavo punto a punto
Hart		Par trenzado	15 p/segm	1.2 K		Maestro/Esclavo

4.6.1 Beneficios de una red Device Net

Device Net proporciona una red flexible y de conexión sencilla que ofrece entre sus beneficios más inmediatos, un control descentralizado y permite la conexión de dispositivos de diferentes marcas gracias a la interoperatividad y su carácter abierto y estándar. Con su instalación se obtiene una reducción drástica del cableado, del tiempo de puesta en marcha y del coste de la instalación porque elimina la necesidad de recorrer largas distancias de cable.

Las paradas de producción se minimizan ya que proporciona información de diagnóstico muy valiosa que permite llevar a cabo acciones preventivas y eficaces soluciones de los problemas, agilizando las tareas de mantenimiento y reparación. Además, en caso de avería de algún módulo esclavo, es posible sustituirlo en "caliente" sin necesidad de quitar la alimentación y en consecuencia sin tener que detener las comunicaciones o la instalación en sí. Por otro lado, cabe destacar su eficiencia en las comunicaciones ya que permite que la información de planta esté disponible en tiempo real al proporcionar un procesamiento de datos a alta velocidad, mayor seguridad de datos, un chequeo de errores eficiente y gran flexibilidad.

Los mercados y aplicaciones más significativas para este tipo de tecnología son: líneas de ensamblaje de automóviles; líneas de alimentación y bebidas, líneas de fabricación semiconductores, manipulado de material y empaquetado, papeleras, cementeras y canteras, clasificación y tratamiento de R.S.U. (Residuos Sólidos Urbanos), y líneas de productos de consumo, entre otros.

4.6.2 Kit de Iniciación de Device Net

Con el propósito de introducir la Tecnología Device Net, se armó y se realizaron pruebas de comunicación con el Kit de Iniciación de Device Net. Actualmente en el laboratorio se cuentan con dos Kit de Iniciación (Starter Kit) el Starter Kit 1 y el Starter Kit 7.

La diferencia entre estos kits es que el Starter Kit 1 se puede conectar a un PLC que cuenta con scanner Allen Bradley (es un módulo del PLC) que se conecta a la computadora por medio de una interfaz RS-232. El Starter Kit 7 no ocupa de esta interfaz y este cuenta con una scanner que se conecta directamente a la computadora sin necesidad de usar un PLC.

El scanner es el elemento principal de la red que nos permite reconocer otros elementos que se conecten a dicha red, es el encargado de establecer la comunicación entre la red y la computadora (en caso del Starter Kit 7) o al PLC (en caso del Starter Kit 1).

La conexión entre cada uno de los elementos de la red Device Net se hace a través de un cable plano que contiene 5 cables de diferente color, por lo que cabe mencionar que la red lleva las señales de alimentación para cada elemento (+24 V) y además incluye los cables de la señales que producen a la salida cada elemento. Por ejemplo si hablamos de un sensor inductivo de Device Net, a través de la red circular 24 V que son la alimentación del mismo y la señal que produce como señal de salida de este sensor que viene siendo una señal analógica que variará de acuerdo si existe o no un objeto metálico en su cercanía.

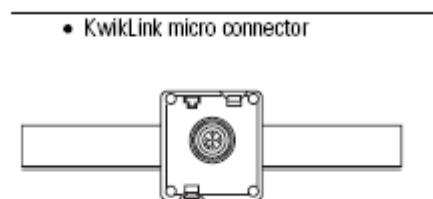


Figura 4.3. Cable plano flexible (5 hilos).

La ventaja de un sensor Device Net en comparación con un sensor convencional es que el primero es “inteligente” ya que es capaz de enviar a la red información de su deterioro con respecto al tiempo y es capaz de decirnos si se encuentra activo o inactivo. Además un sensor Device Net se puede conectar directo a la red sin necesidad de intermediarios. Un sensor convencional se conecta a través del Armor Block, que un módulo especial de Device Net.

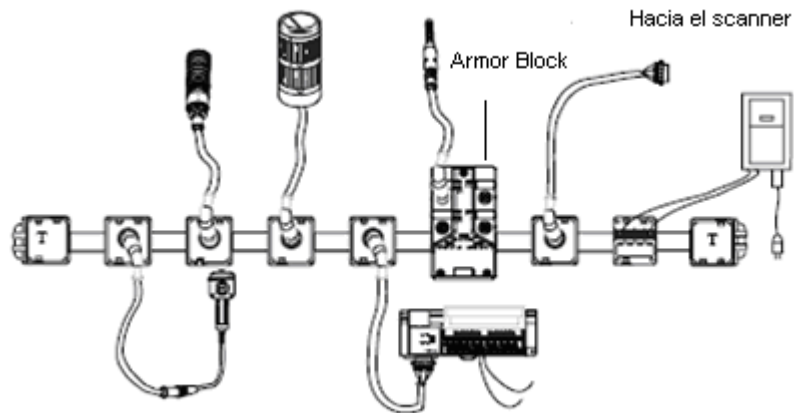


Figura 4.4. Vista del Kit de Iniciación 7 Device Net.

Para dar de alta a la red se ocupan de tres programas básicos que son el Rs-Linx, el Rs-Networx y el Soft Logics 500. El Rs-Linx se utiliza para dar de alta los drivers y cada elemento en la red. Para lograr lo anterior, el primer paso es dar de alta los drivers, indicando en este paso el tipo de scanner que se va a utilizar.

El Rs-Networx se utiliza para administrar la red y en general para comisionar los nodos de la cada elemento. Con esto se le asigna un lugar en el mapa de memoria y a su vez se le da la prioridad a cada elemento. Eso depende de las características individuales de cada uno y del uso que se les vaya a dar.

En el momento de realizar la comisión de nodos, esta se debe de hacer en dos partes: una es en cada elemento y la otra es en el programa Rs-Networx, por default el scanner queda comisionado en el nodo 00.

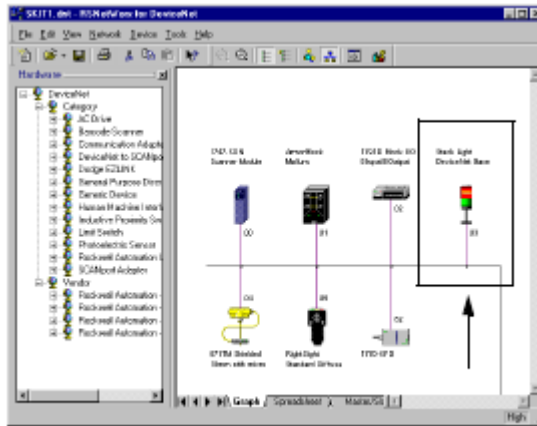


Figura 4.5. Red Device Net configurada en Rs-Networkx.

Aquí cabe mencionar que las direcciones posibles que se pueden asignar van desde las 00 hasta la 63, y debido a que se trata de un Kit de Iniciación nada más se pueden conectar solo un determinado número de elementos a la red.

Por último el Soft Logics 500 se utiliza para hacer la programación de la red, así como establecer el control con cada elemento.

Cabe mencionar que este último se utilizó para el manejo de entradas y salidas del Robot GE-Fanuc-LR Mate 200i para llevar a cabo las primeras pruebas de comunicación de la red y el robot.

Capítulo 5

El Microcontrolador

5.1 Antecedentes de los Microcontroladores Genéricos

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso.

En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo pero completo computador contenido en el corazón (chip) de un circuito integrado.

Un microcontrolador dispone normalmente de los siguientes componentes: procesador o UCP (Unidad Central de Proceso), memoria RAM para Contener los datos, memoria para el programa tipo ROM/PROM/EPROM, líneas de E/S para comunicarse con el exterior y diversos módulos para el control de periféricos (temporizadores, puertos serie y paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).

Los productos que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas: aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo, aumento de la fiabilidad: al reemplazar un elevado número de elementos por el microcontrolador disminuye el riesgo de averías y se precisan menos ajustes, reducción del tamaño en el producto acabado: la

integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los almacenes.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado (embedded controller).⁶

5.2 Modos de Operación y Mapas de Memoria

Existen básicamente 4 modos de operación del microcontrolador, los cuales son: el modo sencillo, el modo expandido, modo trampa de arranque y el modo prueba especial.

En modo sencillo el microcontrolador es un dispositivo monolítico sin direcciones externas, ni buses de datos.

En modo expandido el microcontrolador puede acceder a un direccionamiento de memoria hasta 64 kbytes. El espacio de memoria incluye los mismo que en modo sencillo más la memoria externa que se le agregue al dispositivo. Los buses de expansión son los puertos B y C.

La siguiente figura explica lo anterior:

⁶ <http://www.monografias.com/trabajos34/microcontroladores-genericos/microcontroladores-genericos.shtml>

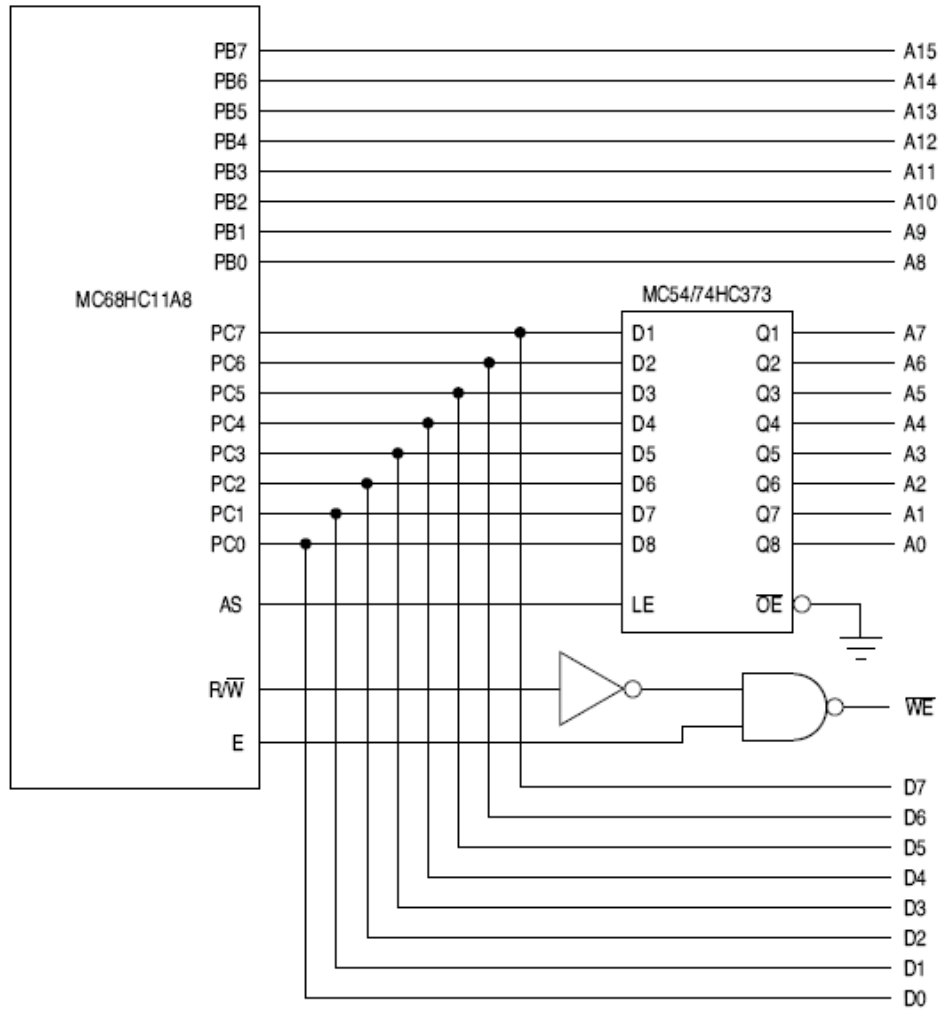


Figura 5.1. Demultiplexación de Datos/Direcciones.

El modo de trampa de arranque permite que ciertos programas ingresen a la memoria RAM interna. El cargador del programa utiliza la interfase serial (SCI) para leer un programa de 256 bytes en la memoria RAM de la microcontrolador, utilizando para ello el direccionamiento \$0000 hasta \$00FF, después de esto el programa continuará en la dirección \$0000 de la EEPROM.

Por último el modo de prueba especial se usa únicamente para pruebas del fabricante.

5.2.1 Mapas de memoria

Las posiciones de memoria son los mismos para modo expandido y modo sencillo. La memoria RAM de 256 bytes esta localizada en la dirección \$0000. Los 64 bytes de un block registros se encuentran a partir de la dirección \$1000. La EEPROM de 512 bytes esta ubicada en la dirección \$B600 a \$B7FF. Por último la memoria ROM de 8 kbytes se encuentra a partir de la dirección \$E000 hasta la \$FFFF. Véase la figura 5.2 para mayor información.

La prioridad del hardware se establece en el mapeo de memoria. Los registros tienen prioridad sobre la memoria RAM y la memoria RAM tiene prioridad sobre la memoria ROM.

En modo de trampa de arranque, la memoria ROM de arranque se ubica en la dirección \$BF40 hasta la dirección \$BFFF. En modo de prueba especial y en modo de trampa de arranque los vectores de interrupción y de reset se encuentran ubicados a partir de la dirección \$BFC0 hasta la dirección \$BFFF.

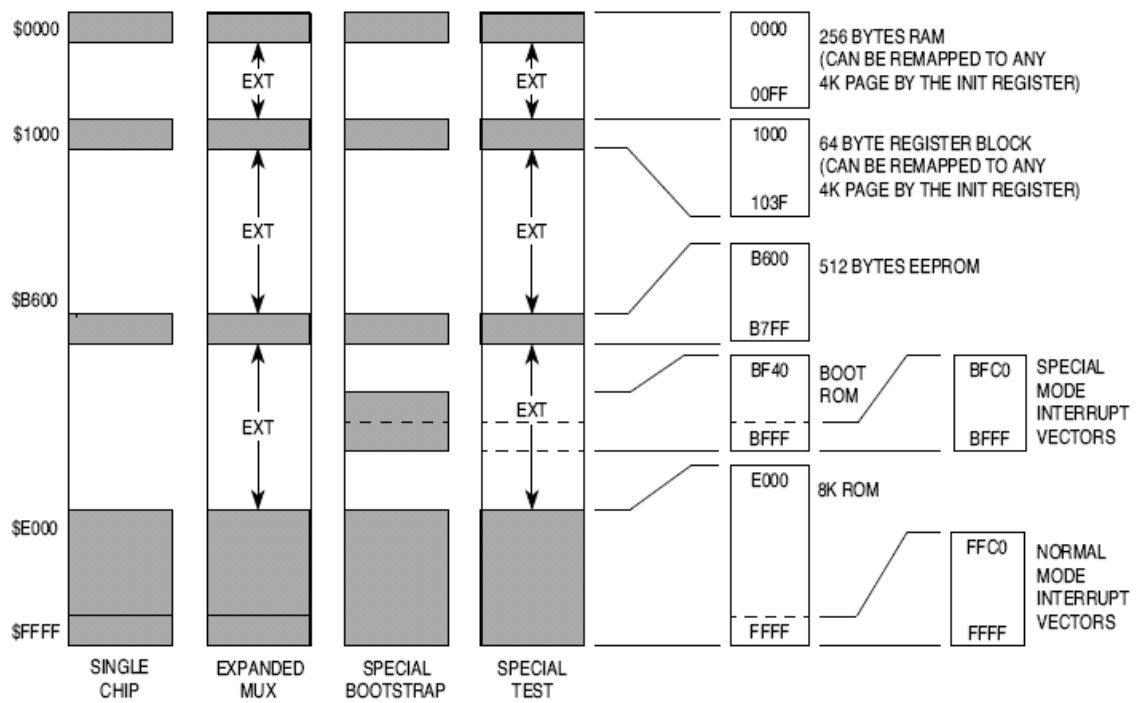


Figura 5.2. Mapa de memoria.

5.3 Hardware del Microcontrolador 68HC11

En los sistemas de control profesionales hay que evitar cualquier inestabilidad ya que en su caso se puede producir una catástrofe. Sistemas de este tipo son los utilizados en la industria del automóvil, aeronáutica, sistemas de seguridad, etc.

Por esta razón fue necesaria la inclusión de sofisticados sistemas COP, que tiene por objetivo velar por la fiabilidad del chip. Las acciones que realiza son las de verificar que la ejecución de las instrucciones se mantiene en sincronía con el reloj, que este último mantiene la frecuencia correcta sin demasiadas variaciones.

5.3.1 Pastillaje del MC68HC11

El MC68HC11 (MCU), dispone de 98 funciones de entrada y/o salida, las cuales se ven representadas por 52 pines en el caso de un encapsulado del tipo PLCC, o bien 48 para el del tipo DIP. Además, dependiendo del modelo MCU, se cuenta con diferentes tipos y tamaños de memorias (ROM, RAM, EPROM, EEPROM).

Debido al diseño de los circuitos internos del micro, muchas de las señales de salida son de **colector abierto**. El fabricante recomienda como resistencia de pull-up un valor de 4.7 k Ω .

Cuando se monta un sistema digital basado en microcontrolador, existe siempre el peligro de que un mal diseño provoque no solo un mal funcionamiento sino un daño irreparable de los circuitos. Esto se debe a que a diferencia de la lógica digital habitual, los microcontroladores, en general, trabajan con varios tipos de señales, conversores A/D, salidas PWM, líneas de transmisión y un largo etcétera, lo que provoca que un mal conexionado pueda tener graves consecuencias.

Un mecanismo de protección frente a este tipo de conflictos es el adoptado por el 68HC11 donde varios de sus pines se encuentran dotados de circuitos internos de protección. Igualmente este tipo de soluciones tienen sus propias limitaciones por lo que nunca se debe bajar la guardia. Para saber más sobre estas protecciones acudir al manual de referencia técnica de motorola.

Para facilitar la comprensión se clasifican todos los pines del microcontrolador en grupos de acuerdo a las funciones de los mismos, siendo estas agrupaciones las siguientes:

1. Alimentación: VDD, VSS.
2. Reloj: EXTAL, XTAL, E.
3. Reset: RESET.
4. Transmisión serie asíncrona: TxD, RxD.
5. Petición de interrupciones hardware: IRQ, XIRQ, IC1-3, PAI, STRA.
6. Modos de arranque: MODA, MODB.
7. Comparadores: OC1-5.
8. Capturadores: IC1-3, PAI.
9. Transmisión serie síncrona: SCK, MISO, MOSI, SS.
10. Puertos: PA0-7, PB0-7, PC0-7, PD0-3, PE0-3.
11. Conversores: AN0-7.
12. Buses: AD0-7, A8-15, AS, R/W.

Vease figura 5.3 para observar los detalles.

Figura 5.3. Pastillaje del 68HC11.

5.3.2 Pines de entrada/salida

La integración y la potencia de este microcontrolador obliga a un alto grado de multiplexación de las funciones que soporta, por tanto, si bien la cantidad de puertos existentes en el chip es grande y variada, estos se ven a veces desactivados ya que son utilizados para otros fines como comparadores, capturadores, control de transmisiones asíncronas, etc. de tal manera que el número de puertos netos, puede hasta resultar escaso para algunas aplicaciones.

Esto, que siempre es dependiente del sistema en cuestión, tiene una sencilla solución ya que el 68HC11 está preparado para soportar diferentes tipos de expansiones.

Tabla 5.1. Estructura de los puertos.

Puerto	A	B	C	D	E
Bit 0	entrada	salida	bidireccional	entrada	entrada
Bit 1	entrada	salida	bidireccional	salida	entrada
Bit 2	entrada	salida	bidireccional	bidireccional	entrada
Bit 3	salida	salida	bidireccional	bidireccional	entrada
Bit 4	salida	salida	bidireccional	bidireccional	entrada
Bit 5	salida	salida	bidireccional	bidireccional	entrada
Bit 6	salida	salida	bidireccional	–	entrada
Bit 7	bidireccional	salida	bidireccional	–	entrada

5.4 Programación del Microcontrolador 68HC11

5.4.1 Registros del CPU

La CPU del MCU dispone de 2 registros acumuladores de 8 bits, que se unen para formar el registro D de 16 bits, siendo el acumulador A la parte alta y el acumulador B la parte baja. Además dispone de 2 registros para direccionamiento indexado X, Y ambos de 16 bits. El puntero de pila y el contador de programa son también de 16 bits, lo que permite que la longitud máxima de un programa sea de 64Kbytes, que es el espacio máximo direccionable por el MCU. El registro CCR es el llamado registro de estado, que contiene unos bits de especial importancia que reflejan el estado de la CPU.

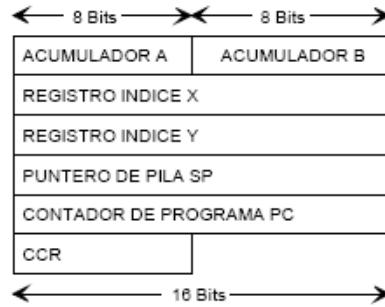


Figura 5.4. Registros del CPU del Microcontrolador 68HC11.

El **puntero de pila** debe ser inicializado por el usuario. La pila "crece" desde direcciones altas hacia direcciones bajas, por lo que al introducir un elemento en la pila, SP se decrementa en 1 ó 2 bytes dependiendo del tamaño del dato metido en la pila. Al sacar un elemento de la pila, SP se incrementa.

El contador de programa **PC** se va incrementado según se van ejecutando las instrucciones. Por tanto, los programas se ejecutan desde direcciones bajas a altas y la pila crece de direcciones altas a bajas. Es importante dar a SP un valor "seguro" de tal manera que la pila no se solape con el código, si es que el código se encuentra en RAM.

El registro **CCR** es de 8 bits. Cada bit tiene una letra asignada y representa una situación diferente del estado de la CPU.

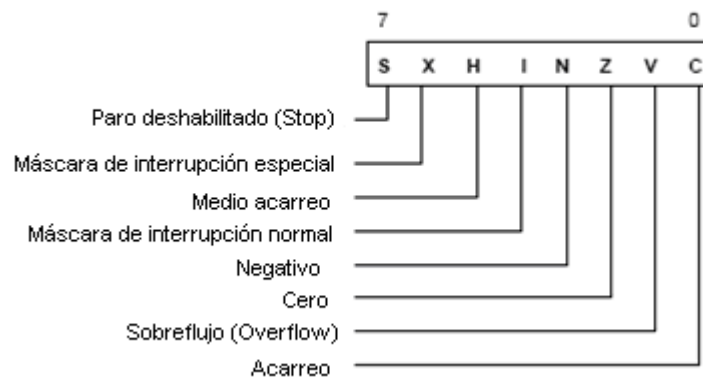


Figura 5.5. Registro de banderas CCR.

5.4.2 Modos de direccionamiento

Existen 6 modos de direccionamiento distintos. Los modos de direccionamientos son distintas formas que tiene la CPU de acceder a los datos que están en memoria.

- **Direccionamiento inmediato**

El dato al que se hace referencia se encuentra "dentro" de la instrucción, no es necesario acceder a memoria. El dato puede ser de 1 ó 2 bytes. Este modo de direccionamiento se indica mediante el signo #.

Ejemplo: LDAA #08 Esta instrucción carga el valor decimal 8 en el acumulador A. El valor 08 se encuentra a continuación del código de instrucción de LDAA. La instrucción LDAA tiene el código \$86, por tanto, en memoria esta instrucción queda representada mediante los valores: (Se supone que esta instrucción comienza en la dirección \$0000).

- **Direccionamiento Extendido**

El dato se encuentra en la dirección de memoria especificada. El dato puede estar en cualquier posición de la memoria dentro del límite de las 64Kb, por lo que la dirección ocupa 2 bytes.

Ejemplo: LDAA \$FC00 Esta instrucción carga en el acumulador el contenido de la dirección \$FC00. La dirección del dato se almacena después del código de la instrucción y ocupa 2 bytes. Las instrucciones con este modo de direccionamiento ocupan 3 bytes (1 byte para el código de la instrucción y 2 bytes para la dirección).

- **Direccionamiento Directo**

Este modo de direccionamiento es exactamente igual que el anterior con la excepción de que sólo actúa con direcciones comprendidas entre \$00-\$FF (256 primeros bytes de la memoria). La utilidad de este modo es que sólo necesita 1 byte para especificar la dirección del dato, con lo que se ahorra espacio y tiempo.

Ejemplo: (1) LDAB \$FC00

(2) LDAB \$05

Ambas instrucciones cargan en el acumulador B el contenido de la dirección especificada. En el caso de la instrucción 1, se utiliza direccionamiento extendido y la instrucción ocupa 3 bytes. En el caso de la instrucción 2, como la dirección es menor que \$FF se utiliza direccionamiento directo y la instrucción ocupa 2 bytes.

- **Direccionamiento Indexado**

Este modo de direccionamiento se utiliza para acceder a tablas (arreglos, cadenas, etc.) en la memoria. El dato se busca de la siguiente forma: Se toma la dirección del registro índice (X ó Y), se le suma un desplazamiento (offset) de 8 bits y el contenido de esa dirección es el dato buscado.

Este modo de direccionamiento se especifica colocando como argumentos en la instrucción un offset, una coma y el registro índice.

Ejemplo: LDAB 5,X Esta instrucción carga en el acumulador B el contenido de la dirección especificada por X más un offset de 5. Se simboliza de la siguiente manera: $dir = (X) + 5$, siendo dir la dirección que contiene el dato.

Para lograr lo del ejemplo se ocupa cargar al registro X con una dirección específica, por ejemplo: **LDX \$1000**. En este caso se asigna la dirección de los puertos de E/S. Siendo esta la asignación mas común que se utiliza en programación del microcontrolador.

- **Direccionamiento Relativo**

Este modo de direccionamiento sólo se utiliza con las instrucciones de **bifurcación**. Estas indican a la CPU que realice un salto de tantos bytes hacia adelante o hacia atrás. El desplazamiento **tiene signo** y es de un byte por lo que las bifurcaciones sólo se pueden hacer de 128 bytes hacia atrás ó 127 bytes hacia adelante:

Ejemplo: ciclo:

....

BNE ciclo

- **Direccionamiento Inherente**

Los operandos se encuentran en registros de la CPU. Por el código de la instrucción la CPU sabe de qué registros se trata.

Ejemplo:

	CÓDIGO	MNEMÓNICO	
	\$1B	ABA	; A:=A+B
	\$5C	INCB	; B:=B+1
	\$08	INX	; X:=X+1

La primera instrucción, de código \$1B, suma el contenido del acumulador A y B y el resultado lo introduce en el acumulador A. La segunda incrementa el acumulador B y la tercera el registro de índice X. Estas instrucciones sólo ocupan un byte y por ello conviene abusar de ellas.

5.4.3 Conjunto de instrucciones

Las instrucciones se dividen en distintos grupos. Todas las instrucciones tienen dos campos: uno es el mnemónico y el otro es el dato o la dirección a la que hace referencia la instrucción.

- **Instrucciones de carga**

Estas instrucciones permiten introducir un nuevo valor en los registros, leer una posición de memoria, un puerto etc. Nota: el punto y coma (;) se utiliza en el lenguaje ensamblador para agregar comentarios.

LDAA: Introduce un dato de 8 bits en el acumulador A.

Ej. LDAA #30 ; A:=30 (Direccionamiento inmediato)

LDAA \$1000 ; Introduce en A el contenido de la dirección \$1000.

LDAB: Introduce un dato de 8bits en el acumulador B

LDD: Introduce un dato de 8 ó 16 bits en el doble acumulador D (Formado por A y B yuxtapuestos).

Ej. LDD #\$FFCC ; D:=\$FFFF --> A:=\$FF; B:=\$CC

LDD #\$10 ; D:=\$0010 --> A:=\$00; B:=\$10

LDX: Introduce un dato de 16 bits en el registro índice X.

Ej. LDX #\$1000 ; X:=\$1000

LDX 3,Y ; Mete en X el contenido de la dirección Y+3

LDX 5,X ; Mete en X el contenido de la dirección X+3.

LDY: Introduce un dato de 16 bits en el registro de índice Y.

LDS: Introduce un dato de 16 bits en el SP (puntero de pila). Esta instrucción hay que utilizarla al menos una vez en nuestros programas para inicializar la pila.

Ej. LDS #\$FC00 ; Inicializa la pila a partir de la dirección \$FC00 hacia abajo

CLRA: Borra el contenido del acumulador A. Esta instrucción hace lo mismo que LDAA #0, con la diferencia de que el direccionamiento es inherente y sólo ocupa 1 byte, mientras que LDAA #0 ocupa 2 bytes.

CLRB: Borra el contenido del acumulador B.

- **Instrucciones de almacenamiento**

Estas instrucciones permiten alterar una posición de memoria, un puerto, registros internos, etc.

STAA: Almacena el acumulador A en una dirección de memoria.

Ej. STAA \$1000 ; Manda el acumulador por el puerto A

STAB: Almacena el acumulador B.

STD: Almacena el doble acumulador D (16 bits).

STX: Almacena el registro de índice X (16 bits).

STY: Almacena el registro de índice Y (16 bits).

STS: Almacena el puntero de pila SP.

CLR: Pone a cero el contenido de una dirección de memoria.

Ej. CLR \$1000 ; Manda un 0 por el puerto A.

Como nota adicional este tipo de instrucciones se utilizan en conjunto con las instrucciones de carga. Esto se debe a que primero se carga un dato en algún registro y luego se almacena en el registro que generalmente apunta a un puerto o una dirección de memoria (variable local, alguna cadena o arreglo).

- **Instrucciones de transferencia**

Permiten transferir datos de registro a registro y de memoria a memoria. El direccionamiento es inherente por lo que no es necesario especificar dirección. Con el mnemónico es más que suficiente.

PSHA: Introduce el acumulador A en la pila. Se introduce A en la dirección especificada por SP. SP se decrementa en 1

PSHB: Introduce el acumulador B en la pila

PSHX: Introduce el registro de índice X en la pila. Se introduce X en la pila. SP se decrementa en 2 unidades puesto que X es de 16 bits.

PSHY: Introducir el registro de índice Y en la pila.

PULA: Saca A de la pila. Se decrementa SP en una unidad y se introduce en A el dato contenido en la posición apuntada por SP.

PULB: Saca B de la pila.

PULX: Saca X de la pila. SP se incrementa en 2 unidades ya que X es de 16 bits.

PULY: Saca Y de la pila.

TAB: Transfiere el contenido de A en B.

TBA: Transfiere el valor de B en A.

TSX: Transfiere el valor de SP en X.

TSY: Transfiere el valor de SP en Y.

TXS: Transfiere el valor de X en SP.

TYS: Transfiere el valor de Y en SP.

XGDY: Intercambia el registro D con el X.

XGDX: Intercambia el registro D con el Y.

- **Instrucciones aritméticas**

Suma.

ADDA: Añade un dato al acumulador A.

Ej. ADDA #5 ; Suma 5 al acumulador. ADDA \$C000 ; Sumar el contenido de la dirección \$C000 al acumulador.

ADDB: Añade un dato al acumulador B.

ADDD: Añade un dato al doble acumulador D.

ADCA: Añade al acumulador A un dato y el contenido del acarreo.

ADCB: Añade al acumulador B un dato y el contenido del acarreo.

ABA: Suma el acumulador A y B y pone el resultado en A.

ABX: Suma B y X y pone resultado en X.

ABY: Suma B y Y y pone resultado en Y.

INCA: Incrementa el acumulador A.

INCB: Incrementa el acumulador B

INC: Incrementa el contenido de una dirección de memoria.

Ej. INC 2,Y ; Suma una unidad al byte que se encuentra en la dirección Y+2.

INX: Incrementa registro X.

INY: Incrementa registro Y.

INCS: Incrementa puntero de pila SP.

Resta

SUBA: Resta un dato al acumulador A.

Ej. SUBA #\$2C ; Resta \$2C al acumulador.

SUBB: Resta un dato al acumulador B.

SUBD: Resta un dato al doble acumulador D.

SBCA: Resta al acumulador A un dato y el contenido del acarreo.

SBCB: Resta al acumulador B un dato y el contenido del acarreo.

DECA: Decrementa acumulador A.

DECB: Decrementa acumulador B.

DEC: Decrementa byte de una dirección de memoria.

DEX: Decrementa registro X.

DEY: Decrementa registro Y.

DES: Decrementa puntero de pila SP.

Comparaciones

CMPA: Compara acumulador A con un dato. Se activan los bits correspondientes del registro de estado. Los bits que se activan son el Z (Cero) y el N (negativo).

Ej. CMPA #10 ; Compara Acumulador A con 10.

CMPB: Compara Acumulador B con un dato.

CPD: Compara doble acumulador D con un dato.

CPX: Compara registro X con un dato.

CPY: Compara registro Y con un dato.

CBA: Compara A con B.

Complemento a dos

Estas instrucciones permiten obtener números negativos en formato de complemento a dos.

NEG: Obtiene el complemento a dos un byte de la memoria.

NEGA: Obtiene el complemento a dos el acumulador A.

NEGB: Obtiene el complemento a dos el acumulador B.

Multiplicaciones y divisiones

MUL: Se multiplican A y B y el resultado se introduce en el doble acumulador D.

IDIV: Se divide D entre X y el resultado se guarda en X. El resto se guarda en D.

- **Operaciones lógicas y de manipulación de bits**

ANDA: Se realiza una operación AND lógica entre el registro A y la memoria. El resultado se almacena en el acumulador A.

ANDB: Se realiza una operación AND lógica entre el registro B y la memoria. El resultado se almacena en el acumulador B.

ORAA: Se realiza una operación OR lógica entre el acumulador A y la memoria. El resultado se almacena en el acumulador A.

ORAB: Se realiza una operación OR lógica entre el acumulador B y la memoria. El resultado se almacena en el acumulador B.

EORA: Realiza una operación OR Exclusiva (XOR) entre el registro A y memoria y almacena resultado en acumulador A.

EORB: Realiza una operación OR Exclusiva (XOR) entre el registro B y memoria y almacena resultado en acumulador B.

COMA: Se realiza el complemento a uno de A.

COMB: Se realiza el complemento a uno de B.

BITA: Esta instrucción sirve para comprobar si determinados bits de una posición de memoria están activados o no. Se realiza un AND lógico entre el acumulador y la posición de memoria pero no se altera ninguna de las dos. El resultado queda reflejado en el bit Z del registro CCR.

Ejemplo: Queremos comprobar si los bits 0 y 1 del puerto A están ambos activados:

LDAA #\$03 ; Meter el valor \$03 (00000011 en binario) en A

BITA PuertoA ; Comprobar bits 0 y 1.

BEQ subrutina; Saltar si ambos bits son cero.

BITB: Hace lo mismo que BITA pero con el acumulador B.

BCLR: Pone a cero los bits especificados de una posición de memoria. La sintaxis es: BCLR operando máscara. El operando es una posición de memoria a la que se puede acceder mediante cualquiera de los modos de direccionamiento. Máscara es un byte, cuyos bits a uno se corresponden con los bits del operando que se quieren poner a cero. Por

ejemplo, se quiere poner a cero los bits 0 y 1 del puerto A: **BCLR PORTA,X \$03**. Esta instrucción solo permite direccionamiento indexado.

BSET: Lo mismo que BCLR pero los bits en vez de ponerse a cero se ponen a 1.

BRCLR: Esta instrucción es muy útil y un poco diferente del resto porque tiene 3 parámetros. Se bifurca a la dirección especificada si unos bits determinados están a cero. La sintaxis es: **BRCLR operando máscara dirección**. Se realiza un AND lógico entre el operando y la máscara y se bifurca si la operación da como resultado cero, es decir, si todos los bits indicados estaban a cero. Un ejemplo muy típico es un bucle de espera hasta que se active un bit de una posición de memoria:

LDX #\$1000

wait BRCLR 0,X \$80 wait

El ciclo se repite mientras el bit 7 del puerto A sea distinto de cero. En cuanto se ponga a cero se sale del ciclo. Esta instrucción sólo permite direccionamiento indexado.

BRSET: Igual que BRCLR pero se salta cuando los bits indicados se ponen a 1.

- **Desplazamientos y rotaciones**

Desplazamientos aritméticos

ASL: Desplazamiento aritmético a la izquierda de un operando en memoria.

ASLA: Desplazamiento aritmético a la izquierda del acumulador A.

ASLB: Desplazamiento aritmético a la izquierda del acumulador B.

ASLD: Desplazamiento aritmético a la izquierda del acumulador D.

ASR: Desplazamiento aritmético a la derecha de un operando en memoria.

ASRA: Desplazamiento aritmético a la derecha del acumulador A.

ASRB: Desplazamiento aritmético a la derecha del acumulador B.

Desplazamientos lógicos

LSR: Desplazamiento lógico a la derecha de un operando en memoria.

LSRA: Desplazamiento lógico a la derecha del acumulador A.

LSRB: Desplazamiento lógico a la derecha del acumulador B.

LSRD: Desplazamiento lógico a la derecha del acumulador D.

Rotaciones

ROL: Rotación a la izquierda de un operando en memoria.

ROLA: Rotación a la izquierda del acumulador A.

ROLB: Rotación a la izquierda del acumulador B.

ROR: Rotación a la derecha de un operando en memoria.

RORA: Rotación a la derecha del acumulador A.

RORB: Rotación a la izquierda del acumulador B.

- **Bifurcaciones y saltos**

Bifurcaciones

Las bifurcaciones (instrucciones **BRANCH**) se diferencian de los saltos en que se realizan mediante direccionamiento relativo por lo que sólo se pueden utilizar para saltar 128 bytes hacia atrás o 127 bytes adelante. Las bifurcaciones condicionales bifurcan a la dirección especificada cuando se da una determinada condición en el registro de estado **CCR**.

BCC: Bifurcación si acarreo está a cero.

BCS: Bifurcación si acarreo está a uno.

BEQ: Bifurca si el resultado a sido cero ($Z=1$).

BGE: Bifurca si mayor o igual (Signo).

BGT: Bifurca si mayor que (Signo).

BHI: Bifurca si mayor que (Sin signo).

BHS: Bifurca si mayor o igual (Sin signo).

BLE: Bifurca si menor o igual (Signo).

BLO: Bifurca si menor (Sin Signo).
BLS: Bifurca si menor o igual (Sin signo).
BLT: Bifurca si menor (Signo).
BMI: Bifurca si negativo (N=1).
BNE: Bifurca si no igual (Z=0).
BPL: Bifurca si positivo (N=0).
BVC: Bifurca si overflow está a cero (V=0).
BVS: Bifurcar si overflow está a uno (V=1).
BRA: Bifurcar (Salto incondicional).
BSR: Llamar a una subrutina (incondicional).

Salto

Los saltos se pueden realizar a cualquier dirección de memoria.

JMP: Salto incondicional.
JSR: Salto incondicional a una subrutina.

- **Instrucciones de modificación del CCR**

Estas instrucciones alteran los bits del registro de estado CCR.

CLC: Pone a cero el bit de acarreo.
SEC: Pone el bit de acarreo a uno.
CLI: Pone el bit de interrupciones a cero. Las interrupciones se permiten.
SEI: Pone el bit de interrupciones a uno. Las interrupciones se inhiben.
CLV: Pone el bit de overflow a cero.
SEV: Pone el bit de overflow a uno.
TAP: Mueve el Acumulador A al registro CCR.
TPA: Mueve el CCR al acumulador A.

- **Otras instrucciones**

RTS: Regresa de una subrutina.

RTI: Regresa de una interrupción.

SWI: Interrupción por Software.

WAI: Espera hasta que ocurra una interrupción.

NOP: No operación. No hace nada salvo consumir un ciclo de reloj.

STOP: Para el reloj.

5.5 Comunicación Serial

El microcontrolador 68HC11 dispone de una unidad de comunicaciones serie (SCI) que permite realizar comunicaciones asíncronas a distintas velocidades y con paquetes de 8 y 9 bits. (8 ó 9 bits de datos).

Los paquetes mandados contienen un bit de comienzo en la cabeza (Bit de start), 8 ó 9 bits de datos y un bit de stop al final.

Para configurar los parámetros de la comunicación, enviar y recibir datos y comprobar el estado de la transmisión, el SCI dispone de 5 registros mapeados en memoria.

Además el SCI dispone del modo especial de funcionamiento WAKE-UP, usado en sistemas multireceptores pero de uso poco frecuente. En este capítulo se describen las funciones básicas de los bits que actúan sobre este modo.

5.5.1 Unidad de transmisión y recepción

El SCI está formado por una unidad de transmisión y una unidad de recepción que son totalmente independientes, lo que permite que las comunicaciones sean bidireccionales, es decir, se puede transmitir y recibir a la vez (Modo Full-duplex).

La unidad de transmisión está formada por un registro de desplazamiento con carga en paralelo, llamado **registro de transmisión**. Al introducir un valor en este registro, comienza a desplazarse hacia la derecha el contenido del registro, enviando los bits por la línea serie, a una velocidad configurable por el usuario.

De la misma manera la unidad de recepción dispone de otro registro, **registro de recepción** que recibe los bits en serie y los va desplazando hasta obtener un dato en paralelo que puede ser leído.

Tanto el registro de transmisión como el de recepción están mapeados en la misma dirección de memoria. Al escribir en esa dirección de memoria, el dato se cargará en el registro de transmisión. Al efectuar una lectura, el dato se leerá del registro de recepción. Ambos registros comparten la misma dirección física de memoria pero se trata de dos registros diferentes. Puesto que ambos registros comparten dirección física, se les ha asignado un único nombre: **registro de datos (SCDR)**, y está situado en la dirección \$102F.

Una vez configurado el SCI adecuadamente, enviar y recibir datos es una tarea muy sencilla: basta con leer o escribir en el registro de datos.

Ejemplo.

Para enviar datos:

LDX #\$1000 ; Utilizar X para acceder a memoria

LDAA #dato ; Cargar en el acumulador A el dato a enviar (Dato de 8 bits)

STAA \$2F,X ; Enviar dato por el puerto serie.

Para recibir datos:

LDX #\$1000

LDAA \$2F,X ; Cargar en acumulador A el dato recibido.

El siguiente código podría no funcionar correctamente:

LDX #\$1000

LDAA #dato1 ; Cargar dato1 en el acumulador A

STAA \$2F,X ; Mandar dato1

LDAA #dato2 ; Cargar dato2 en el acumulador A

STAA \$2F,X ; Mandar dato2

El problema está en que el programa se ejecuta más rápidamente que lo que tarda el SCI en enviar el dato. El dato1 se empieza a mandar. El MCU continúa ejecutando las siguientes instrucciones mientras el dato se sigue enviando. Cuando se escribe dato2 en el registro de datos, todavía no se ha terminado de enviar el dato1, por lo que el primero es traslapado.

Debido a esto, existe un bit en un registro del SCI que se pone a 1 cuando el registro de transmisión está vacío e indica que el dato que se había escrito antes ya se ha enviado.

5.5.2 Registros del SCI

El SCI dispone de 5 registros mapeados en memoria. Estos registros se pueden ver en la tabla 5.2.

Tabla 5.2 Registros del SCI.

REGISTRO	DIRECCIÓN	DESCRIPCIÓN
BAUD	\$102B	Registro de velocidad
SCCR1	\$102C	Registro de control 1
SCCR2	\$102D	Registro de control 2
SCSR	\$102E	Registro de estado
SCDR	\$102F	Registro de datos

- **Registro de velocidad (BAUD, \$102B)**

Tabla 5.3. Registro de velocidad de transmisión.

7	6	5	4	3	2	1	0
TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0

El registro de velocidad permite configurar la velocidad (en baudios) de la comunicación. Véase tabla 5.3. Los bits 4 y 5 (SCP0 y SCP1) determinan la máxima velocidad en baudios. Esta velocidad depende del cristal que se haya conectado al microcontrolador. Lo habitual es colocar un cristal de 8 Mhz.

Tabla 5.4. Selección de velocidades máximas de transmisión y recepción.

SCP1	SCP0	Baudios
0	0	125000
0	1	41667
1	0	31250
1	1	9600

Una vez determinada la velocidad máxima en baudios, con los bits 0, 1 y 2 (SCR0, SCR1 y SCR2) se divide la velocidad máxima por un valor:

Tabla 5.5. Relaciones entre las velocidades de transmisión y recepción.

SCR2	SCR1	SCR0	Dividir vel. máx. entre
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

De esta manera, si por ejemplo se selecciona 9600 baudios como velocidad máxima (SCP1=1 ; SCP2=1), al dividir entre 8 se obtiene una velocidad de 1200 baudios.

- **Registro de control 1 (SCCR1 \$102C)**

Tabla 5.6. Registro de control 1.

7	6	5	4	3	2	1	0
RS	TS	0	M	WAKE	0	0	0

El bit 4 (M) permite configurar el SCI para utilizar 8 ó 9 bits de datos: (Véase tabla 5.6 donde se muestran los bits del registro de control 1)

M=1 --> 9 bits de datos.

M=0 --> 8 bits de datos.

Lo normal es utilizar 8 bits de datos. Para comunicarse con un PC el bit se deberá poner a cero puesto que el PC no permite más de 8 bits de datos. Por defecto está a cero. Si se quieren transmitir 9 bits de datos, el noveno bit se escribe en el bit 6 del SCCR1 (T8) y los 8 bits menos significativos se escriben en el registro de datos.

Análogamente, si se quiere recibir un dato de 9 bits, el bit más significativo (el noveno) se sitúa en el bit 7 del registro SCCR1 (R8) y los 8 bits restantes en el registro de datos.

El SCI tiene un modo de funcionamiento especial (modo WAKEUP) para aumentar la eficiencia en sistemas multireceptores. Es el modo en el cual el receptor se queda con las interrupciones inhibidas esperando un evento hardware externo, (asociado a la línea de recepción), que le devuelva al estado activo con interrupciones. El evento externo puede ser de dos tipos y se selecciona con el bit 3 (WAKE). Si WAKE=1 entonces se espera hasta detectar una marca de dirección, si por el contrario WAKE=0 se espera hasta detectar que la línea de recepción esta vacía.

El modo WAKE-UP se selecciona mediante software activando el bit 1 del registro de control 2 (RWU). Los bits 0,1,2 y 5 no se usan y permanecen siempre a cero.

- **Registro de control 2 (SCCR2 \$102D)**

Tabla 5.7. Registro de Control 2.

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

Este registro de control 2 es el principal, con él se configura la salida al exterior del circuito de comunicaciones serie. Esta salida se corresponde con los pines emparejados a los bits 0 y 1 del puerto D. Por tanto, esos pines tienen dos funciones: ser utilizados como bits 0 y 1 del puerto D o como señales Tx y Rx para comunicaciones serie. Mediante el bit 3 del registro de control 2 (TE) se activa o desactiva el transmisor del pin correspondiente. Si TE=1 (Transmitter enabled) el transmisor está activo y el bit 1 del puerto D se desactiva. No se puede utilizar el bit PD1 del puerto D. Si TE=0 el transmisor se desconecta y no se mandan datos serie. El pin funciona como bit 1 del puerto D. Vease tabla 5.7 donde se observan los bits del registro de control 2.

Análogamente, con el bit 2 del registro de control 2 (RE) se activa o desactiva el receptor, que comparte pin con el bit 0 del puerto D. Si RE=1 (Receive Enable) el pin funciona para recibir datos serie del exterior. El bit 0 del puerto D deja de funcionar. Si RE=0 se desconecta el receptor y el pin funciona como bit 0 del puerto D.

En resumen, siempre que se quieran transmitir y recibir datos es imprescindible activar los bits RE y TE del registro de control 2. Una vez activados estos bits, cualquier lectura/escritura sobre el puerto D no se reflejará en los pines 0 y 1 del puerto D, como si no existiesen.

Los bits 4, 5, 6 y 7 (ILIE, RIE, TCIE, TIE) son máscaras que permiten que ciertas interrupciones ocurran o no. TCIE y TIE corresponden a dos interrupciones del transmisor y ILIE y RIE a dos interrupciones del receptor. En el transmisor, cada vez que el registro de transmisión se vacía, es decir, cada vez que se termina de mandar un dato y por tanto se puede mandar un dato nuevo, se genera una interrupción. Esta interrupción se enmascara con el bit TIE. Si TIE=1 la interrupción de transmisión está permitida y para TIE=0 está desactivada. El bit de estado asociado es TDRE, cuando TDRE=1 indica que se ha recibido un dato y si el bit TIE está a 1 entonces se produce la interrupción. También hay otra interrupción que indica cuando se ha quedado vacía la línea de transmisión después de mandar el dato, esta se puede enmascarar con el bit TCIE. Si TCIE=1 la interrupción queda permitida, si TCIE =0 está desactivada. El bit de estado asociado es TC y el funcionamiento es análogo al anterior.

En el receptor también existe una interrupción que aparece cada vez que se ha recibido un dato, esta se enmascara con el bit RIE del registro de control 2. Con RIE=1 se permite y con RIE=0 no. El bit de estado asociado es RDRF. La otra interrupción se activa cuando la línea de recepción está vacía, el bit ILIE se encarga de enmascararla. Igual que antes, ILIE=1 la permite y ILIE=0 la desactiva. El bit de estado asociado es IDLE.

Cuando se activa el bit 1, (SBK=1), se mandan señales de BREAK indefinidamente hasta que el bit se desactive. Las señales de BREAK se caracterizan porque se envía todo ceros por la línea serie, no sólo son cero los bits de datos, sino que también se hace cero el bit de stop que siempre vale 1.

Si el bit 1 (RWU) es igual a 1 significa que el modo especial WAKE-UP esta activo. En este modo el receptor esta aletargado con las interrupciones inhibidas, esperando una condición hardware que lo despierte. Dicha condición depende del valor que tenga el bit 3 (WAKE) del registro de control 1 (SCCR1). Generalmente es el programa quien pone este bit a 1 y la CPU quien lo desactiva.

- **Registro de estado (SCSR \$102E)**

Tabla 5.8. Registro de estado (SCSR \$102E).

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	IDLE	OR	NF	FE	0

El registro de estado es de sólo lectura y permite comprobar el estado del SCI. El bit 7 (TDRE) se pone a 1 cada vez que se ha terminado de enviar un carácter y por tanto el registro de transmisión está vacío y listo para enviar el siguiente carácter. Siempre que se vaya a transmitir un dato hay que asegurarse que este bit está a 1, porque de lo contrario traslapa el dato que se está enviando. (Vease tabla 5.8 para mayor información).

El bit 6 (TC) se pone a 1 cada vez que se ha enviado un carácter y la línea de transmisión se ha quedado vacía 'IDLE'. Este bit te ofrece mayores garantías cuando se quiere saber si la transmisión ha terminado completamente.

El bit 5 (RDRF) se pone a 1 cuando se ha recibido un dato nuevo en el registro de recepción. Por tanto, cuando se reciben datos hay que esperar a que este bit se ponga a 1.

El bit 4 (IDLE) se pone a 1 cuando detecta que la línea de recepción se ha quedado vacía (IDLE). Si el bit RWU del registro de control 2 esta activo entonces este bit estará inhibido.

El bit 3 (OR) se pone a 1 cuando se ha recibido un carácter por el puerto serie y el anterior dato recibido todavía no se ha leído. Cuando ocurre este error el dato que se pierde es el que se acaba de recibir.

El bit 2 (NF=Noise Flag) se activa cuando se ha detectado un error en el dato que se acaba de recibir. La activación de este bit no produce interrupción, será el software quien se preocupe de examinar este bit después de recibir un dato para saber si es válido o no.

El bit 1 (FE) se activa cuando se ha detectado un error en la trama enviada. Al recibir un dato se comprueba que el bit de stop este a nivel alto. Si esto no se cumple se pone este bit (FE) a nivel alto (1). El bit 0 no se utiliza.

Todos estos bits se ponen a cero automáticamente cuando se lee el registro SCSR y a continuación se lee del registro de datos (SCDR).

- **Registro de datos (SCDR \$102F)**

Este registro tiene una doble función. Si se escribe en él, el dato se manda al registro de transmisión en la unidad de transmisión del SCI para ser enviado con la velocidad y configuración establecida. Al leerlo se obtiene el valor que tiene el registro de recepción interno de la unidad de recepción del SCI.

Ejemplo de SCI.

; Este programa se debe cargar en la RAM interna del 6811.

; Se configura el SCI a 9600 baudios y 8 bits de datos. Se envían los caracteres ; 'A' y 'B' por el puerto serie.

```

; +-----+
; Registros del SCI
BAUD EQU $2B
SCCR1 EQU $2C
SCCR2 EQU $2D
SCSR EQU $2E
SCDR EQU $2F
; ----- CONFIGURACIÓN DEL SCI -----
LDX #$1000 ; Para acceder a registros del SCI
LDAA #$30
STAA BAUD,X ; Velocidad transmisión: 9600 baudios
LDAA #$00
STAA SCCR1,X ; 8 bits de datos
LDAA #$0C
STAA SCCR2,X ; Inhibir interrupciones SCI.
; Activar transmisor y receptor del SCI
; ---- CICLO PRINCIPAL ----
ciclo:
LDAA #'A'
BSR enviar
BSR pausa
LDAA #'B'
BSR enviar

```



```
BSR pausa
BRA ciclo
pausa:
LDY #$FFFF
wait DEY
CPY #0
BNE wait
RTS
```

```
;+-----+
```

```
;| Enviar un carácter por el puerto serie (SCI) |
;| ENTRADAS: El acumulador A contiene el carácter a enviar |
;| SALIDAS: Ninguna. |
```

```
;+-----+
```

```
enviar: BRCLR SCSR,X $80 enviar
STAA SCDR,X
RTS
```

Capítulo 6

Integración del Módulo de Calidad

En este capítulo se hablará de una manera más detallada del diseño de la parte práctica del trabajo o diseño del experimento, como lo manejan varios autores. Se pretende describir paso por paso cada una de las partes que lo conforman.

Como se mencionó anteriormente el problema general es la integración de un Sistema Flexible de Manufactura, pero el problema de este trabajo se enfoca a la integración del Módulo de Calidad.

El objetivo principal es de crear un Módulo de Calidad capaz de recibir un lote de piezas maquinadas previamente en el Módulo de Manufactura, dichas piezas son paletizadas por el Módulo de Transferencia. Con esto ya elaborado el Módulo de Calidad debe ser capaz de tomar pieza por pieza y colocarlas a través de un método en una máquina de medición por coordenadas.

Posteriormente el Módulo de Calidad debe retirar la pieza analizada (dimensionalmente) y ser capaz de tomar decisiones si la pieza pasó la prueba o no, y luego hacer la disposición de producto final. De aquí cabe mencionar que si el resultado del análisis dimensional no paso la prueba, el sistema debe reconocer si la pieza manufacturada se pasa por arriba del límite permitido o si esta por debajo de este.

Con lo anterior es posible identificar que pieza puede ser retrabajada (si pasa por encima del límite permitido) o ser totalmente rechazada (cuando esta se encuentre por debajo del límite permitido).

6.1 Caracterización del Robot GE-Fanuc-LR Mate 200i

Para llevar a cabo lo anterior y hacer la integración de un Módulo de Calidad se usó el Robot GE-Fanuc-LR Mate 200i, la Máquina de Medición por Coordenadas (CMM) y la red industrial Device Net.

El robot tiene la capacidad de reconocer piezas que se encuentran paletizadas y ponerlas una por una en la CMM. Todo esto supervisado y controlado por la red Device Net.

El robot tiene muchas aplicaciones, así como grandes posibilidades de comunicación. Este cuenta con módulos que son capaces de conectarlo directamente a red Ethernet, Device Net, entre otras, también el robot tiene entradas y salidas digitales por lo que lo hace flexible a comunicación por PLC o ser interconectado en forma directa a sensores, con posibilidades de automatización.

Para lograr la funcionalidad del robot se tuvo que crear un programa de movimiento, esto se logro después de estudiar detenidamente el manual de funcionamiento del robot GE-Fanuc-LR Mate 200i.

- **Programa del Robot Fanuc.**

1. LBL[1]
2. LBL[Lectura del sensor de pieza palletizada]
3. IF DI[3]=ON, JMP LBL[2]
4. JMP LBL[1]
5. LBL[2]
6. LBL[Robot se mueve a recoger la pieza]
7. J P[1] 100 % FINE
8. J P[2] 25 % FINE
9. LBL[Robot abre la pinza]
10. RO[1]=ON
11. WAIT 3.00(sec)

12. J P[3] 20 % FINE
13. LBL[Robot cierra la pinza]
14. RO[1]=OFF
15. WAIT 3.00(sec)
16. LBL[Robot se mueve a poner la pieza en la CMM]
17. J P[4] 100 % FINE
18. J P[5] 100 % FINE
19. J P[6] 100 % FINE
20. J P[7] 30 % FINE
21. RO[1]=ON
22. WAIT 3.00(sec)
23. J P[9] 100 % FINE
24. LBL[4]
25. LBL[Una vez que la CMM termina el robot recoge la pieza]
26. IF DI[1]=ON, JMP LBL[6]
27. IF DI[2]=ON, JMP LBL[6]
28. JMP LBL[4]
29. LBL[6]
30. J P[8] 100 % FINE
31. RO[1]=ON
32. WAIT 3.00(sec)
33. J P[10] 100 % FINE
34. RO[1]=OFF
35. LBL[7]
36. LBL[Robot lee si la pieza es buena o mala]
37. IF DI[1]=ON, JMP LBL[8]
38. IF DI[1]=ON, JMP LBL[9]
39. JMP LBL[7]
40. LBL[8]
41. LBL[Pieza Mala]
42. J P[11] 100 % FINE
43. J P[12] 100 % FINE
44. RO[1]=ON
45. WAIT 3.00(sec)
46. J P[13] 100 % FINE
47. JMP LBL[10]
48. LBL[9]
49. LBL[Pieza Buena]
50. J P[14] 100 % FINE
51. J P[15] 100 % FINE
52. RO[1]=ON
53. WAIT 3.00(sec)
54. J P[16] 100 % FINE
55. LBL[10]
56. END

El programa del robot FANUC, maneja instrucciones muy claras y abreviadas. En el programa del robot listado arriba se utilizó para leer una de las entradas digitales que provienen de los sensores Device Net que se colocaron de manera estratégica en la CMM. Entonces dependiendo del resultado de la medición el robot pondrá la pieza en lugar o en otro.

En primera instancia, el programa lee una entrada digital, que proviene del módulo de transporte indicándole que una pieza nueva ha llegado. Cuando esto sucede, el robot procederá a cargar la pieza en la CMM, para que esta realice las operaciones de medición correspondientes. Posteriormente, el robot pondrá en lugar las piezas “buenas” y en otro lugar las piezas “malas”.

6.2 Caracterización de la Máquina de Medición de Coordenadas (CMM).

La CMM tiene muchos usos, en lo que respecta a análisis dimensional, es capaz de realizar mediciones de piezas y llevar a cabo pruebas con varios límites de tolerancias. Pero sus capacidades de comunicación son muy limitadas. La CMM solo puede comunicarse a través del puerto serie RS-232. Es decir este sistema no cuenta con entradas y salidas digitales.

Con lo antes descrito, fue como se planteo la principal problemática de la integración del Módulo de Calidad. Más adelante se describirá como se resolvió este problema en particular y otros que fueron saliendo en cada una de las fases de integración.

Para llevar a cabo la integración se tuvo que estudiar por separado los manuales de operación del robot Fanuc, de la CMM y de la red Device Net. Lo que dio como resultado de generar un programa de inspección de la CMM, una rutina de movimiento en el robot Fanuc y un diagrama de escalera que realiza las funciones de control de la red.

Pero para realizar la comunicación de la CMM con el robot Fanuc, fue necesario crear una interfaz con un microcontrolador MC68HC11A1FN de Motorola. Para lo que se

ocupo recordar los métodos de programación en lenguaje ensamblador (lenguaje para programar un microcontrolador) y dio como resultado la generación de un código capaz de manipular datos del RS-232, entradas y salidas digitales.

En la siguiente sección se lista un ejemplo de un programa básico en el QC-5000 que realiza la medición de dos diámetros de una pieza previamente manufacturada en el Sistema de Manufactura Flexible.

Programa para medir una pieza.

```
Program Properties...
+ Initial settings...
/ Programa de Inspeccion...
Set full cnc mode...
Go to position (215.4821,109.1547,-325.5009)
- ● P ⊕ Measure "Small Circle"
    Wait for point(s) at (257.4900,91.3872
    Go to position (257.2339,85.0126,-357.
    Wait for point(s) at (274.7144,102.810
    Go to position (287.5436,106.3172,-357
    Wait for point(s) at (257.3840,128.728
    Go to position (252.2009,140.4431,-357
    Wait for point(s) at (238.9478,114.118
    Finish measurement
    If [Small Circle].Overallpass==1 Goto sigue TRUE (1.0000)
    Goto label "mala"...
: Label "siguiente"...
Go to position (209.8978,112.7037,-354.0735)
- ● P ⊕ Measure "Big Circle"
    Wait for point(s) at (257.2294,83.8997
    Go to position (254.0742,69.9468,-363.
    Wait for point(s) at (285.2576,107.980
    Go to position (287.9081,115.9435,-356
    Save part
    Wait for point(s) at (256.9505,137.328
    Go to position (247.4045,146.9855,-367
```

```

        Wait for point(s) at (231.9014,114.634
        Finish measurement
Go to position (209.4490,114.5335,-327.7042)
If [Big Circle].Overallpass==1 Goto buena... TRUE (1.0000)
Save part
:   Label "mala"...
    Go to position (567.1962,477.3706,-21.7565) .
    Show the message "Pieza Mala"
    Goto label "fin"...
:   Label "buena"...
    Go to position (22.9897,477.3343,-21.7604)..
    Show the message "Pieza Buena"
:   Label "fin"...
    Save part

```

Básicamente un programa de la Máquina de Medición de Coordenadas (CMM) se divide en cuatro columnas: la primera solo se utiliza para minimizar o expandir ciertos pasos de la medición, que contienen a su vez otros pasos, ejemplo de esto, es la medición de un círculo, donde en este en forma original aparecen solo instrucciones de espera hasta encontrar el punto a medir.

La medición de un círculo requiere al menos medir 3 puntos alrededor de una circunferencia (para diámetros exteriores) y la medición de puntos interiores cuando se trate de medir perforaciones circulares.

Para medir un diámetro exterior se vio en la necesidad de agregar instrucciones adicionales, en este caso se agregaron clearance moves, entre punto y punto, con el objetivo de que la medición se lleve a cabo de forma exitosa. De otra manera, aparecerían colisiones entre cada punto de la medición por lo que conlleva a un paro total de la CMM.

En la segunda columna aparece, un dibujo pequeño representativo de la medición a efectuar, en este caso aparece un círculo, puesto lo que se midió fueron dos diámetros exteriores de una pieza. Dentro de la misma columna aparece una P en color verde si la prueba fue exitosa (paso los límites de la tolerancia) y aparece un F en color rojo, en el caso contrario, es decir cuando la prueba falló.

En la tercera columna que es el cuerpo del programa, es en donde aparecen las instrucciones del programa y en la cuarta columna aparecen los datos, por ejemplo si se lee

un caracter en el puerto serial, este se mostrará en esta columna (se verá un ejemplo de esto mas adelante) y también puede aparecer un cierto o falso, dependiendo si agregó algún condicionante que arroje algún resultado.

El programa como antes se había mencionado, mide dos diámetros exteriores de una pieza, y si esta falla en algún paso, (en caso de que exceda los límites de la tolerancia) el programa hará que la sonda se mueva a un lugar o a otro. Después de esto el programa mostrará un mensaje en la pantalla que dirá “Pieza Buena” o “Pieza Mala”, dependiendo del caso.

Para este programa se le especificaron dos posiciones resultantes, que son detectadas por sensores Device Net para establecer a alguna opción de control posterior. Para hacer misma acción control también existe la posibilidad de usar el puerto serial de la computadora, donde el programa QC-5000 utiliza como opciones de entrada – salida.

6.3 Caracterización de la Red Industrial Device Net

Por medio de la red industrial Device Net se integró el Módulo de Calidad antes mencionado. Con el uso del módulo sink de entradas y salidas de la red Device Net, se llevo a cabo la comunicación entre la CMM y el robot.

Lo anterior se llevo a cabo por medio de la creación de un diagrama de escalera elaborado en Soft Logix 500, lo que permitió utilizar el módulo sink de Device Net como un PLC convencional.

Pero como nota a lo anterior la comunicación entre el robot y la CMM, seguía siendo el problema, ya que el la CMM no puede conectarse directamente al módulo sink, por lo que se optó por crear una interfaz con un microcontrolador para lograr el objetivo. En la siguiente sección se explica como fue que se llevo a cabo.

En la figura 6.1 se observa el diagrama de flujo de la red industrial, y nos da la idea generalizada de cómo es que funciona todo el sistema en general ya integrado.

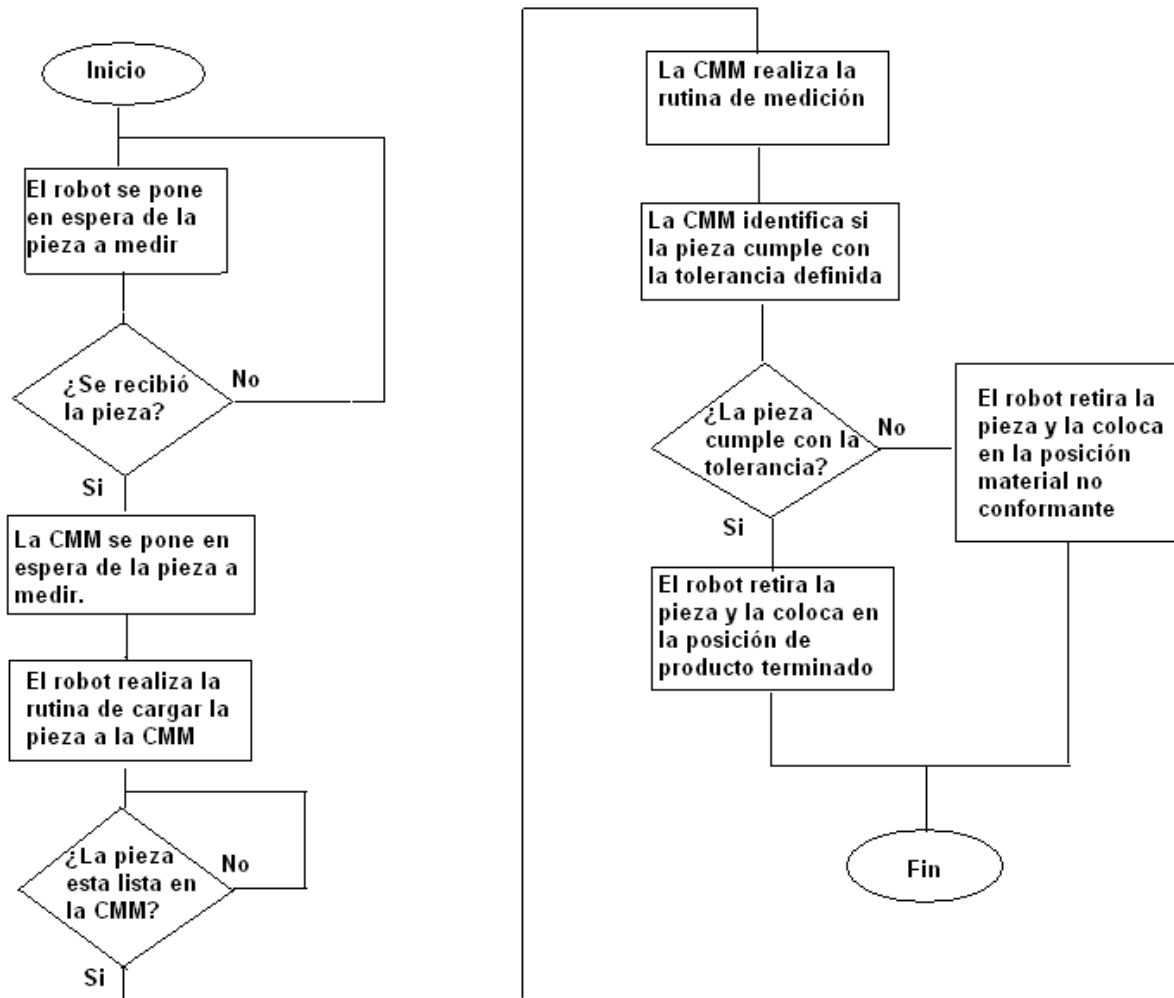


Figura 6.1. Algoritmo de la Red Device Net.

6.4 Interfaz de Comunicaciones CMM y Robot

Esta interfaz se creó para crear la comunicación entre el Robot Fanuc y la Máquina de Medición de Coordenadas, por medio de la red Device Net. Dicha interfaz se llevó a cabo por medio de un microcontrolador Motorola 68HC11.

En un intento anterior por establecer comunicación en los dos elementos antes mencionados (Robot y CMM) se elaboró un programa que en base a ciertas condiciones de

prueba la CMM se mueve a una posición u a otra. Donde posteriormente en esa posición se colocaron sensores Device Net lográndose así una comunicación parcial.

La opción anterior solo puede generar una salida dependiendo del resultado de la medición, pero la limitante es que no se puede generar una entrada digital que haga que la CMM inicie su secuencia de medición en forma automatizada.

Debido a lo anterior, ahora se utilizo el puerto serie RS-232 de la CMM. Cabe mencionar que esta opción es la única forma en que se pueden dar una entrada y una salida a una rutina de medición del QC-5000.

Con lo anterior se puede lograr que en una rutina de medición el programa espere hasta que se haya recibido una cadena de caracteres por medio del RS-232 y mandar otra cadena de caracteres cuando dicha rutina haya terminado. Esto representa una forma de crear una entrada y una salida de la CMM, pero debido a la complejidad del puerto RS-232 y que este elemento no se encuentra disponible en el Robot Fanuc.

Para que el puerto RS-232 se pueda utilizar en el Robot Fanuc es necesario comprar una interfaz capaz de crear una comunicación entre el PLC y el Robot creando con esto entradas y salidas digitales que resolverían los problemas de comunicación de una manera muy sencilla y dinámica. Pero dicha interfaz no se encuentra disponible en el laboratorio, y ocuparía de mucho tiempo y dinero para poder utilizarla.

Por esta razón se decidió crear una interfaz por medio de un microcontrolador Motorola 68HC11, que permite mandar una cadena de caracteres a través del puerto RS-232 cuando cierta entrada digital se haya activado. Y en su contraparte leer una cadena de caracteres y activar una salida digital.

Al realizar esto es posible ahora crear entradas y salidas digitales que se comuniquen de manera directa con el Robot Fanuc.

- **Código utilizado en el Microcontrolador MC68HC11FNA1**

```

baud equ $2B
sccr1 equ $2C
sccr2 equ $2D
scsr equ $2E
scdr equ $2F
puertoC equ $3
puertoB equ $4
DDRC EQU $7
TCNT EQU $0E
PACTL EQU $26
TFLG2 EQU $25
TMSK2 EQU $24
OPTION EQU $39
HPRIO EQU $3C
CONFIG EQU $3F

```

```

org $0          ;Variables locales

```

```

buf_size rmb 10    ;Variable para recepción de datos a través de RS-232
dato rmb 5         ;Variable de control
veces rmb 1        ;Variables para la interrupción de tiempo real
segundos rmb 1

```

```

org $B600

```

```

inicio: LDS #$C0    ;Asignación de dirección a la pila
      LDX #$1000    ;Apuntar X hacia los puertos
      LDAA #$F0     ;Configuración del Puerto C
      STAA DDRC,X

```

```

      CLRA
      STAA OPTION,X
      LDAA #$05
      STAA HPRIO,X

```

```

      LDY #dato
      CLRA
      STAA puertoC,X
      STAA puertoB,X
      STAA $0,Y
      INY           ;Limpieza de variables a utilizar
      STAA $0,Y
      LDY #buf_size

```

```

    LDAA #$10
clean: CLRB
    STAB $0,Y
    INY
    DECA
    BNE clean

    LDAA #$01    ;Habilitación de la interrupción de tiempo real
    STAA TMSK2,X

read: LDAB puertoC,X    ;Lectura de una entrada en el Puerto C
    CMPB #$01
    BEQ msg
    BRA read

msg:  NOP        ;Rutina para eliminar rebotes de contacto
    CMPB #$01
    BNE read
send: LDY #msg0    ;Transmite un cadena por RS-232
    JSR show
    LDY #buf_size  ;Recibe un caracter por RS-232
    JSR rx_cadena
    LDY #buf_size
    LDAB $0,Y      ;Se compara con la letra "r"
    CMPB #'r'
    BNE send
    INY
    LDAB $0,Y
    CMPB #$20
    BNE send

recibir: LDAA #$40
    STAA puertoC,X
    LDY #buf_size  ;Recibe una cadena por RS-232
    JSR rx_cadena
    CLRA
    STAA puertoC,X
    LDY #buf_size
    JSR revisar

mandar2: LDY #dato
    LDAB $0,Y
    CMPB #'F'
    BNE mandar
    LDAA #$40      ;Activa salidas del Puerto B

```

```
STAA puertoB,X
JSR retardo
BSET CONFIG,X $10
JMP inicio
```

```
mandar: LDY #dato
        LDAB $0,Y
        CMPB #'N'
        BNE saltar
        LDAA #20
        STAA puertoB,X
        JSR retardo
        BSET CONFIG,X $10
        JMP inicio
saltar: LDAB #80
        STAB puertoC,X
cicle:  NOP
        BRA cicle
```

Rutina para recibir una cadena de caracteres por medio de RS-232

```
show:  PSHY
        PSHB
        PSHA
        LDAA #30 ;9600 baud
        STAA baud,X
        LDAA #0 ;8N1
        STAA scsr1,X
        LDAA #08 ;Habilitar Tx
        STAA scsr2,X
show2: LDAB $0,Y
        CMPB #'$' ;Compara con fin de cadena
        BEQ show1
        JSR tx_caracter
        INY
        BRA show2
show1: PULB
        PULY
        PULA
        RTS
```

Rutina para enviar un caracter por medio de RS-232

```
tx_caracter: STAB scdr,X
state:      BRCLR scsr,X $80 state
```

RTS

Rutina para recibir un caracter por RS-232

```
rx_caracter: BRCLR scsr,X $20 rx_caracter
              LDAB scdr,X
              RTS
```

Rutina para recibir una cadena de caracteres por RS-232

```
rx_cadena: PSHY
           PSHA
           PSHB
           LDAA #$30
           STAA baud,X
           LDAA #0
           STAA scsr1,X
           LDAA #$04
           STAA scsr2,X
seguir_rx: JSR rx_caracter
           CMPB #$24
           BEQ fin_rx
           STAB $0,Y
           INY
           BRA seguir_rx
fin_rx:   TAB
           PULB
           PULA
           PULY
           RTS
```

Rutina de revisión de caracteres recibidos

```
revisar: PSHY
         PSHB
leer:   LDAB $0,Y
         CMPB #'m'
         BEQ sigue3
         CMPB #'b'
         BEQ fin
         BRA fin2
sigue3: INY
         LDAB $0,Y
         CMPB #$20
         BEQ salida
         BRA leer
salida: LDY #dato
```

```

LDAB #'F'
STAB $0,Y
INY
LDAB #24
STAB $0,Y
JMP mandar2
fin: LDY #dato
LDAB #'N'
STAB $0,Y
INY
LDAB #24
STAB $0,Y
JMP mandar
fin2: PULB
PULY
RTS

```

Rutina de retardo

```

retardo: PSHA
PSHY
LDAA #16
ret1: LDY #FFFF
ret2: DEY
BNE ret2
DECA
BNE ret1
PULY
PULA
RTS

```

interrupción de tiempo real a través de contador del sistema

```

timer: LDD TCNT,X
ANDA #80
CMPA #80
BNE timer
INC veces
LDAB veces
CMPB #31
BNE timer
INC segundos
LDAB segundos
CMPB #30
BNE timer
CLR veces
CLR segundos

```

RTS

```
***Mensajes para enviar***
```

```
msg0 FCC " Inicio $"
```

```
msg1 FCC " recibido $"
```

```
***Redireccionamiento para microcontroladores de 512 bytes de EEPROM***
```

```
org $B7FB
```

```
FCB #$7E
```

```
FCB #$B6
```

```
FCB #$00
```

```
FCB #$A5
```

```
FCB #$5A
```

Con este último elemento (la interfaz de comunicaciones) fue posible lograr la comunicación de la CMM y el robot, ahora se mostrará como es se hizo la conexión de todos los elementos con un diagrama a bloques.

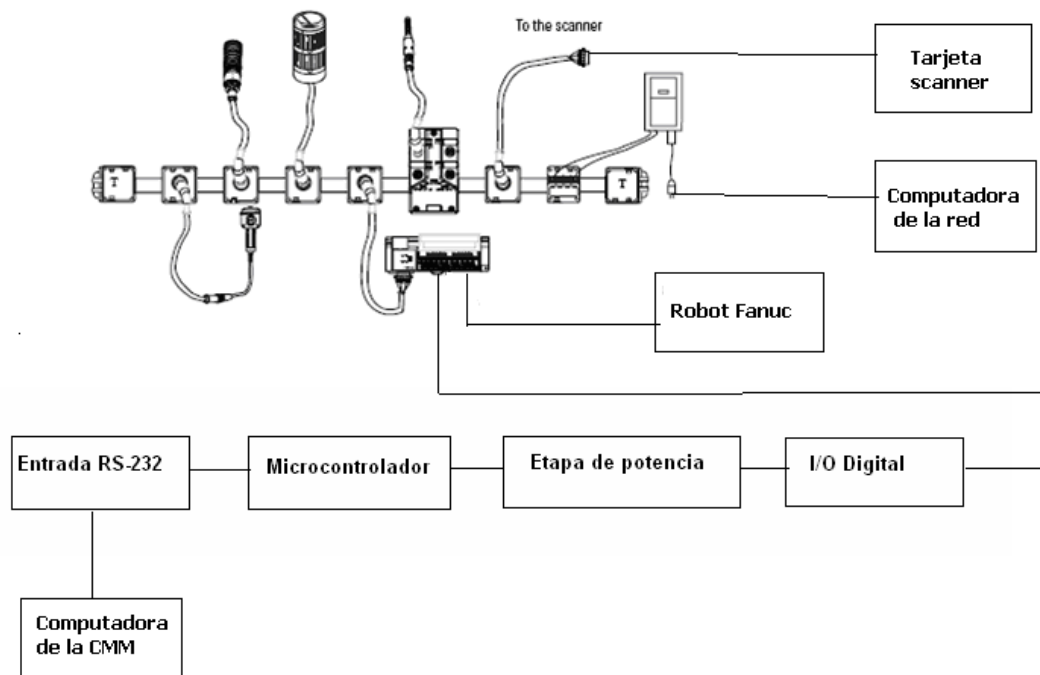


Figura 6.2. Diagrama a bloques de la conexión a red Device Net.

Conclusiones.

Se analizaron varias formas de comunicar la CMM con el Robot Fanuc de las cuales destacan el sensado externo cuya desventaja era que no existía forma de dar una entrada digital al proceso, el método del microcontrolador que resolvió en mayor parte de problema de comunicación y por último el método de la comunicación por medio del Rs Logix 500, que aun se encuentra en etapa experimental.

Se optó por usar una red Device Net debido a sus múltiples ventajas y por ser una red que es muy comercial y ampliamente utilizada.

Por último en este trabajo solo se mostró lo referente al módulo de calidad, los otros módulos se planea conectarse a la red Device Net usando sensores, actuadores de potencia y otros elementos, cabe mencionar que esta labor aun esta pendiente.

De lo anterior se concluye que el módulo de calidad esta listo para integrarse al Sistema Flexible de Manufactura, antes ya mencionado. El módulo de transporte se pretende conectar a través de arrancadores y mediante el uso de variadores de frecuencia, es posible regular la velocidad de la banda. Por otra parte el módulo de manufactura, es cuestión de acondicionar cada una de las señales y acoplarlas al módulo sink de la red Device Net.

Al final solo se hace un diagrama de escalera que gobierne todo el sistema, es posible crear subrutinas que contenga a cada módulo y mandarse llamar en un programa mas genérico. Ya con esto elaborado, se pueden generar enlaces con programas mas detallados, como algún programa administrativo, con el objetivo de que un operador sea capaz de manipular el Sistema Flexible de Manufactura completo.

Retomando el tema del módulo de calidad, dentro de sus aplicaciones o trabajos a futuro, es posible aplicar control estadístico de procesos dándole salida a los datos que se obtienen de la CMM que pueden ser exportados a una hoja de cálculo en Excel.

Una vez en Excel y con el uso de macros, es posible generar hojas de cálculo dinámicas, que nos indiquen en tiempo real los resultados de las mediciones efectuadas por la CMM. Con esto ya elaborado, se puede llevar a cabo un control estadístico de procesos, donde será posible determinar cuando un producto sobrepasa los límites de control establecidos y manejar correcciones dentro del sistema.

Referencias.

- [1] **Administración y Dirección de Operaciones**, Chase Aquilano. Mc Graw Hill.
- [2] **DeviceNet Starter Kit**, Rockwell Automation Allen-Bradley.
- [3] **Planning and Installation Manual**, Allen Bradley.
- [4] **Quadra-Chek 5000 User's Guide**, Metronics, Bedford New Hampshire USA.
- [5] [http://icaro.eii.us.es/descargas/RI%202005%20transparencias%20tema%201\(4%20por%20pag\).pdf](http://icaro.eii.us.es/descargas/RI%202005%20transparencias%20tema%201(4%20por%20pag).pdf)
- [6] <http://www.quizma.cl/servicios/robotica/robot.htm>
- [7] <http://www.intelmax.com/ensanluis/images/capitulo23.ppt>
- [8] <http://www.comflyer.com/strg/docs/j19ho55dcnq.pdf>
- [9] <http://www.monografias.com/trabajos34/microcontroladores-genericos/microcontroladores-genericos.shtml>