

UNIVERSIDAD DE SALAMANCA

FACULTAD DE CIENCIAS



INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Resumen de la Memoria del trabajo práctica realizado para la superación
de la asignatura Proyecto Fin de Carrera

TÍTULO

**RESUMEN DEL PROYECTO DESARROLLO DE HARDWARE DE
COMUNICACIONES Y SOFTWARE DE CONTROL PARA UN BRAZO
ROBÓTICO MA2000**

AUTOR

RAÚL LÓPEZ CHICHES

TUTORES

DRA. BELÉN CURTO DIEGO
DR. VIDAL MORENO RODILLA
DR. ANDRÉS VICENTE LOBER

Septiembre 2012

Índice

1. Introducción.....	5
2. MA2000 USAL Interface Circuit.....	7
2.1 PIC18F2680.....	8
2.2 MAX232N.....	9
2.3 MA2000 USAL Interface Circuit.....	10
3. Driver del PIC.....	12
4. Biblioteca PIC.....	14
5. Conclusiones.....	15

Índice de ilustraciones

Ilustración 1: Robots Industriales.....	5
Ilustración 2: Sistema robótico MA2000.....	6
Ilustración 4: 18F2680.....	8
Ilustración 5: MAX232N.....	9
Ilustración 6: MAX232 Tablas de Función y Diagrama Lógico.....	10
Ilustración 7: Conexiones MA2000 USAL Interface Circuit.....	10
Ilustración 8: Reguladores de Voltaje Positivo.....	12
Ilustración 9: L7805CV.....	12

Índice de tablas

Tabla 1: Formato Mensajes Enviados.....	14
Tabla 2: Formato Mensajes Recibidos.....	14
Tabla 3: Correspondencia Caracteres-Funciones.....	15

1. Introducción

Según la Asociación de Industrias Robóticas (RIA) un **robot industrial** es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.



Ilustración 1: Robots Industriales

La ISO (Organización Internacional de Estándares) lo define como un manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas.

La gran mayoría de las tareas en las que se incluyen robots industriales pertenecen efectivamente al sector secundario, y tradicionalmente en empresas de gran tamaño que pueden hacer frente a los costes de la robotización. Las tareas más típicas son:

- Manipulación: carga y descarga de máquinas, paletizado y transporte y embalado.
- Ensamblado de piezas.
- Procesado: mecanizado como el taladro pulido y desbarbado de objetos, así como tareas de pintura, soldadura (por puntos o continua) sellado y encolado. Por ejemplo en la industria

automovilística o la fabricación de paneles de madera.

Para que la aplicación de estos robots resulte rentable deben ser flexibles y poder utilizarse para varias tareas, esto solo es factible si existe la posibilidad de cambiar su programación y no se requieren cambios permanentes en sus materiales o estructura. Una forma habitual de trabajo es realizar el control de los movimientos del robot desde un ordenador, como es el caso de los brazos articulados MA2000 de TecQuipment presentes en el laboratorio y sobre los que se desarrolla este proyecto.



Ilustración 2: Sistema robótico MA2000

El control por ordenador aporta una enorme flexibilidad ya no solo por la facilidad de cambiar de un programa a otro sino por el rango de posibilidades que aporta; como por ejemplo la teleoperación o su inclusión en un sistema mucho más complejo y que funcione conjuntamente para realizar una determinada tarea como ya se ha hecho en proyectos anteriores.

Para que el control por ordenador sea posible necesitamos obviamente que exista una comunicación entre el robot y el ordenador, esta comunicación se realiza en los robots de laboratorio utilizando un hardware específico que TecQuipment proporcionaba junto con los robots. El problema es que ese hardware ya no es compatible con los equipos actuales porque se conecta a la placa base utilizando un bus ISA lo que limita el trabajo y los desarrollos con los brazos articulados.

La solución a este problema es cambiar ese hardware de comunicación para que siga siendo compatible con el sistema robótico MA2000 y pueda ser utilizado en equipos más actuales. Para desarrollar este nuevo hardware se ha optado por utilizar un microcontrolador PIC, su versatilidad, la facilidad de trabajo con sus puertos y la facilidad en su programación hacen que sea una muy buena opción para este tipo de desarrollos; además de tener una buena relación rendimiento/precio.

Concretamente este proyecto se centra en el desarrollo de un nuevo componente hardware que permita la conexión entre la computadora y el robot además de en el desarrollo del software que permita la utilización de este nuevo hardware. Pese a estos cambios los programas realizados con anterioridad y que funcionaban con la arquitectura antigua podrán continuar funcionando sin realizar cambios significativos, esto es posible gracias a que el API de las funciones de la biblioteca que realiza la comunicación con el robot se ha mantenido intacto sirviendo así de capa de abstracción entre el software de alto nivel disponible y el brazo robótico MA2000.

En el método de trabajo antiguo se utiliza la tarjeta MA2000 IBM Interface Circuit y el puerto ISA de la placa base, el nuevo esquema general solo varía en la tarjeta de interfaz y en la forma de conectarse con el computador, de este modo aislamos del cambio tanto al controlador/interfaz del robot MA2000 como al software que se ejecuta en el ordenador, pues este seguirá utilizando el mismo API de funciones que utilizaba para comunicarse con el controlador/interfaz. De este modo conseguimos abstraer al software de los posibles cambios sobre el hardware de comunicación.

2. MA2000 USAL Interface Circuit

La nueva placa MA2000 USAL Interface Circuit presenta unas grandes diferencias con respecto a la tarjeta MA2000 IBM Interface Circuit. Para empezar esta nueva interfaz no necesita “pincharse” en la placa madre del ordenador por lo que podrá ser usada por cualquier equipo que disponga de un puerto serie COM librándose de las restricciones que presenta la presencia o ausencia de cierto tipo de slots o ranuras en la placa madre. Enumeraremos a continuación las diferencias más significativas:

1. La placa MA2000 USAL Interface Circuit **no se trata de un PPI** como en el caso de la tarjeta antigua al ser su circuito principal un PIC no funciona solamente como un circuito de puertos sino que tiene su propio procesador lo que le permite ejecutar sus propios programas, escritos previamente en la memoria del PIC, y no solo responder a peticiones de lectura o escritura. Esta característica es fundamental como veremos más adelante.
2. No es necesario conectarla directamente en la placa madre, se conecta a un puerto serie COM del ordenador.
3. El número de circuitos utilizados y por lo tanto la complejidad de sus conexiones es mucho menor, esto es posible gracias a que el PIC18F2680 no necesita circuitos adicionales para aislar los buffers.

La capacidad de procesamiento del PIC18F2680 nos permite que la placa MA2000 USAL Interface Circuit no sea un mero intermediario sino que sea la propia placa la que ejecuta las operaciones completas de lectura o escritura, es decir si se le pide modificar el valor de uno de los latches el PIC recibirá la orden y los valores necesarios, pero no será el ordenador quien configure los puertos o modifique las líneas de control, sino el propio PIC que realizará todo el proceso de una forma completamente transparente al ordenador.

2.1 PIC18F2680

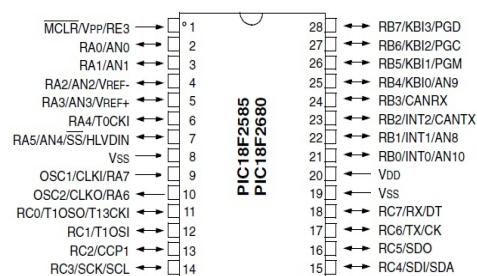


Ilustración 3: 18F2680

El circuito principal de la MA2000 USAL Interface Circuit es un PIC 18F2680; que se encuentra dentro del grupo PIC18F2585/2680/4585/4680. Los **PIC** son una familia de microcontroladores tipo RISC (en inglés *reduced instruction set computer*; Computación de Set de Instrucciones Reducidas) fabricados por Microchip Technology Inc. y derivados del PIC1650.

Que un microprocesador sea considerado RISC (Computación de Set de Instrucciones Reducidas) en la arquitectura de computación significa que tendrá unas instrucciones de tamaño fijo en un reducido número de formatos y que solo las operaciones de carga y almacenamiento acceden a la memoria de datos. Lo que se busca con un diseño RISC es que las instrucciones sean pequeñas y simples porque requieren menor tiempo para ejecutarse, se intenta también que los accesos a memoria sean los menos posibles.

2.2 MAX232N

El MAX232 es un emisor/receptor dual que incluye un generador de voltaje capacitivo para suministrar niveles de voltaje TIA/EIA-232-F utilizando un suministro de 5 Voltios. La norma TIA/EIA-232 describe los aspectos eléctricos (niveles de las señales), lógicos (formato de los datos) y mecánicos (conectores) de una conexión serie entre un DTE (Data Terminal Equipment) y un DCE (Data Circuit-terminating Equipment). Cada receptor convierte las entradas TIA/EIA-232-F en niveles de 5-V TTL/CMOS (TransistorTransistorLogic/ComplementaryMetalOxideSemiconductor). El MAX232N está realizando la función de intermediario entre la información que llega por el conector RS232 y el PIC18F2680, el voltaje que llega por el conector serie y el que utiliza el PIC son diferentes por lo tanto es necesaria una transformación. El pin Rx y Tx del RS232 están conectados al Tx y Rx del PIC respectivamente pero pasando previamente por el MAX232 que realiza la conversión de voltajes.

MAX232 ... D, DW, N, OR NS PACKAGE
MAX2321 ... D, DW, OR N PACKAGE
(TOP VIEW)

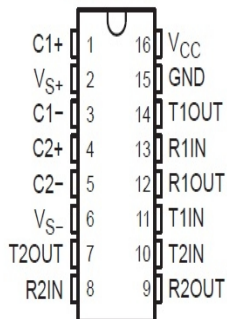


Ilustración 5: MAX232N

Function Tables

EACH DRIVER	
INPUT TIN	OUTPUT TOUT
L	H
H	L

H = high level, L = low level

EACH RECEIVER	
INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)

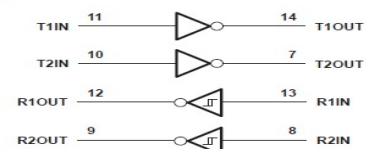


Ilustración 4: MAX232 Tablas de Función y Diagrama Lógico

2.3 MA2000 USAL Interface Circuit

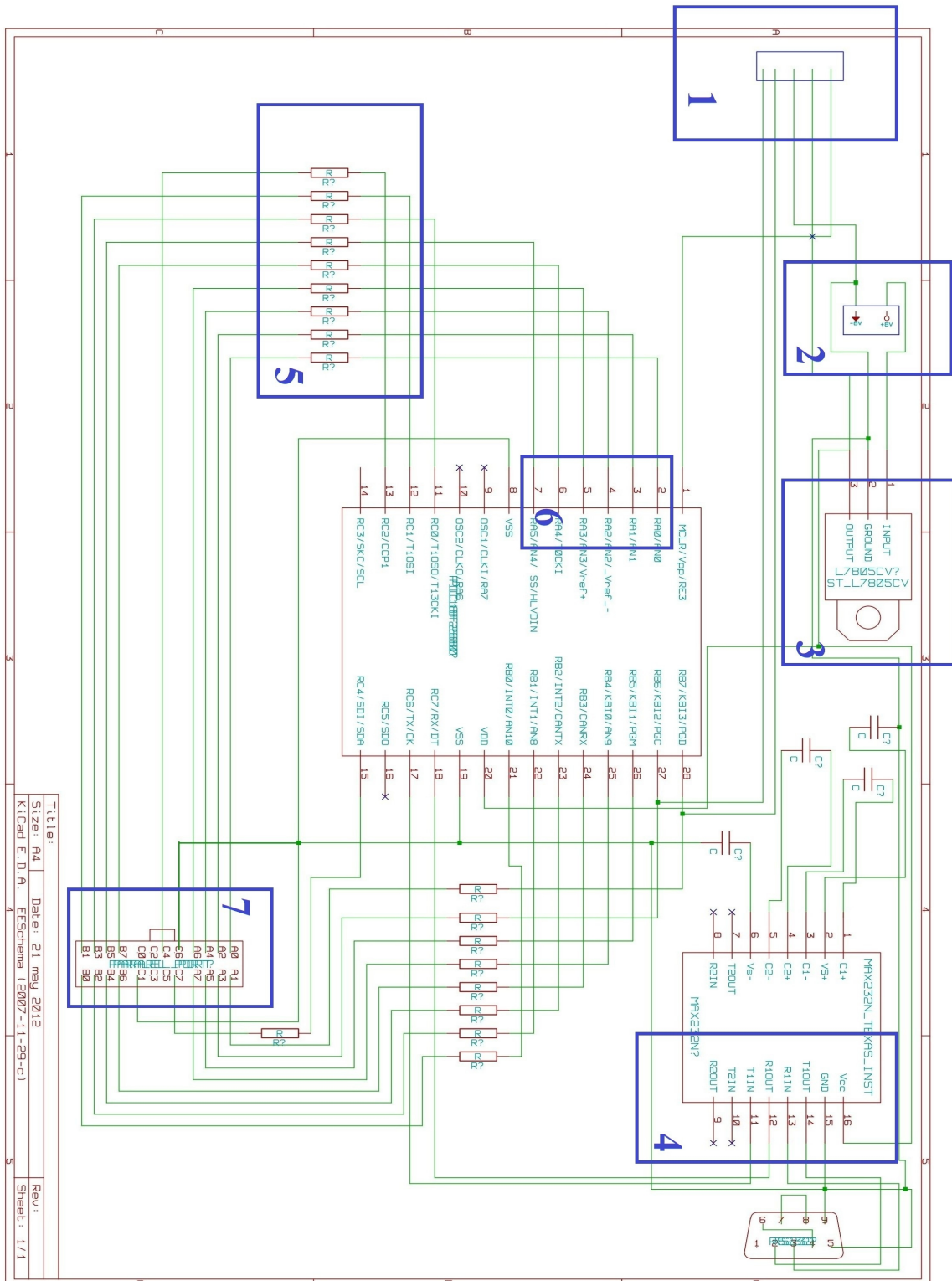


Ilustración 6: Conexiones MA2000 USAL Interface Circuit

Procederemos a realizar una descripción de las conexiones más importantes de la placa MA2000 USAL Interface Circuit y de la función que desempeñan sus componentes.

Se han marcado en azul y numerado los puntos más críticos o que más dificultad pueden representar a la hora de interpretar el diseño del circuito (Ilustración 6):

1. Estos cinco pines se utilizan para conectar el programador PICkit cuando se desea cambiar el programa que ejecuta el PIC; también pueden usarse para alimentar el circuito.
2. Estos dos conectores se utilizan para alimentar la placa, podemos observar como la entrada positiva no va directamente a los circuitos sino que pasa por el L7805CV que se encarga de que la entrada sea lo más próxima a 5V posible ya que es el voltaje al que funciona el PIC18F2680 y utilizar un voltaje superior o inferior podría estropear el circuito. La otra entrada se conecta al pin Ground de L7805CV sirviéndole de referencia, al igual que al resto de circuitos.
3. **L7805CV** se trata de un regulador positivo de tres terminales, se puede usar para una gran cantidad de aplicaciones diferentes, en nuestro caso limita o transforma el voltaje que se nos proporciona por la entrada de alimentación para evitar daños en el circuito.
4. Como ya se ha detallado al describir el circuito **MAX232N** el pin Rx y Tx del RS232 están conectados al Tx y Rx del PIC respectivamente pero pasando previamente por el MAX232 que realiza la conversión de voltajes.

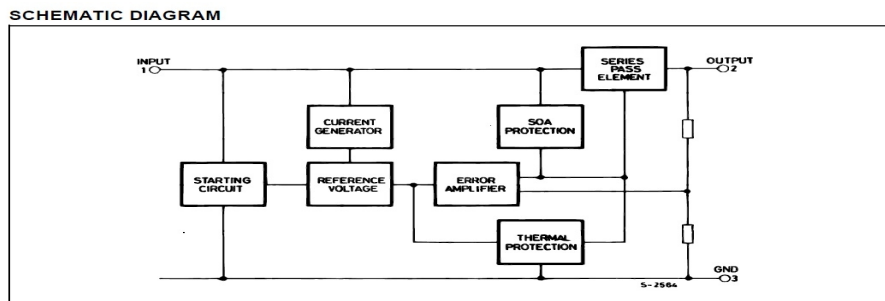


Ilustración 7: L7805CV

5. Las **resistencias** situadas entre los pines del PIC y los pines de los puertos A, B y C del conector paralelo cumplen una función de seguridad.

6. Observando el circuito con detenimiento y sobretodo fijándonos en las conexiones entre el PIC y el conector paralelos vemos que no hay una correspondencia de puertos entre los puertos A, B y C del PIC con los puertos A, B y C del conector, esto es posible gracias a que los pines del PIC18F2680 se pueden configurar en entrada o salida de forma independiente.
7. Siguiendo con la cuestión del punto 6, ¿Cómo podemos enviar los datos correctamente a su destino, si los puertos A, B y C no se corresponden? Lo que sea ha echo ha sido “virtualizar” los puertos y que sea una parte muy concreta del software del PIC quien determine que pines son los que forman el puerto A, B y C según los entiende el resto del programa y el Controlador/Interfaz. De este modo nosotros pediremos enviar algo por el puerto B, pero será el software del PIC quien determine por qué pines saldrá el dato (que será una mezcla del puerto A B y C del PIC) de esta forma nosotros seguimos utilizando el mismo código que funcionaba con la otra placa; ya que lo que estamos haciendo con el PIC es simular el comportamiento del 8255.

3. Driver del PIC

El microcontrolador 18F2680 no se limita a ser un intermediario entre el programa que se ejecuta en el ordenador y el Controlador/Interfaz del robot MA2000; las funciones de comunicación con el robot implican el intercambio de varios valores de configuración y parámetros de control, de modo que si se hubiera optado porque el PIC se limitara a enviar o recibir por el puerto correspondiente una información, generaríamos un flujo de datos entre el ordenador y el circuito que ralentizaría la comunicación aparte de ser prescindible ya que el microcontrolador permite programarlo para que realice esas tareas y que solo se comunique con el ordenador solo para recibir el tipo de tarea o función y los parámetros correspondientes reduciendo drásticamente el número de bytes que deben circular por el puerto serie. Podremos dividir el código desarrollado para el microcontrolador en tres partes:

- El **bucle principal del programa**, que se encarga de leer las órdenes y los parámetros necesarios que el ordenador le envía por el puerto serie y de ejecutar la función correspondiente a la orden recibida. En caso de ser una

orden de lectura de un determinado valor también se encargará de enviar dicho valor al puerto serie del ordenador.

- Las funciones que conformaban la biblioteca **IO8255** que ahora se ejecutan en el microcontrolador para disminuir el volumen de datos que se deben leer y escribir en el puerto serie. Estas funciones son las encargadas de escribir y leer de la memoria compartida para: cambiar o leer posiciones, modificar o leer parámetros de control como el PID y leer el valor de los latches. Cada una de estas funciones se corresponde con una orden, una orden no es más que una letra que identifica la función y los parámetros que necesite en caso de tenerlos; de modo que cuando el programa que se ejecuta en el ordenador necesita realizar una determinada tarea de lectura o escritura envía la letra correspondiente a la función que desea realizar sobre el controlador/interfaz junto con sus parámetros por el puerto serie, el microcontrolador lo recibirá y ejecutará la función que corresponda. Esto en líneas generales es como funciona el protocolo de comunicación entre el MA2000 USAL Interface Circuit y el ordenador. Este código de ordenes que identifica a las antiguas funciones de la IO8255 se encuentra en el programa que se ejecuta en el ordenador en la biblioteca PIC.h de esta manera el cambio realizado y el protocolo que se utiliza es transparente al resto del programa.
- El código de las funciones que formaban la IO8255 no se ha alterado dentro del microcontrolador, de modo que para conseguir que se ejecute correctamente se ha **simulado el comportamiento del 8255** y de las funciones **outportb** e **inportb** que utilizaba la biblioteca, de esta manera cuando una función de escritura realiza un outportb realmente ejecuta otra función también presente en el PIC que se encargará de conseguir que a través de una serie de funciones que simulan el funcionamiento del 8255 se ponga el valor indicado en las líneas correctas pero utilizando las instrucciones y registros del PIC. En definitiva esta tercera parte del código del microcontrolador lo que consigue es **simular el**

comportamiento de la tarjeta MA2000 IBM Interface Circuit.

4. Biblioteca PIC

La biblioteca pic es utilizada por el software que se ejecuta en el ordenador, y se ha desarrollado como sustituto de la io8255; pese a haberse cambiado su nombre se ha mantenido su API de funciones, por lo tanto si se desea utilizar el software ya existente desarrollado en proyectos anteriores bastará con sustituir el nombre de io8255 por pic en las cabeceras del código que lo precisen.

Esta característica es muy importante porque conseguimos aislar al software de los cambios en el hardware de comunicación introduciendo una capa de abstracción entre ambos; cambios en la placa, en el protocolo de comunicación o en el puerto serie que la conectan con el ordenador no afectarían al software que gestiona los movimientos del robot.

Esta biblioteca incluye una serie de funciones con un API idéntico al que contenía la io8255 más una serie de funciones para gestionar el puerto serie. Las funciones que sustituyen a las presentes en la io8255 se limitan a enviar órdenes al PIC a través del puerto serie para que ejecute la función solicitada y en caso de ser operaciones de lectura leer del puerto serie la respuesta. La comunicación se realiza siguiendo un determinado **protocolo**, cada función tendrá asociado un carácter para que el programa del PIC pueda diferenciarlas. Se le envían también los parámetros de la función si los tuviera, siguiendo el mismo orden que tienen en el prototipo de la función. Por último se envía el carácter 13 para indicar el fin del mensaje.

Formato Mensajes Enviados		
Letra que identifica la función	Parámetros si la función los necesita.	13 como caracter de fin de mensaje

Tabla 1: Formato Mensajes Enviados

Formato Mensajes Recibidos					
Éxito	N Bytes de respuesta, cuando se trate de una operación de lectura.	O	K	10	13
Error	No se retorna ningún Byte con la respuesta en caso de error.	K	O	10	13

Tabla 2: Formato Mensajes Recibidos

Correspondencia Caracteres/Funciones	
A	RPutLatchA
a	RGetLatchA
B	RPutLatchB
bb	RGetLatchB
I	RInit8255
O	RPutOutputs
rr	RReadNSM
R	RWriteNSM
W	RW
*Podemos observar que en esta tabla no aparecen todas las funciones de la io8255 esto se debe a que algunas de ellas solo eran utilizadas por las propias funciones de la io8255 y se encuentran implementadas en el PIC y otras no se utilizaban realmente.	

Tabla 3: Correspondencia Caracteres-Funciones

5. Conclusiones

Este proyecto se planteaba en el ámbito del desarrollo del hardware y software necesario para sustituir la tarjeta MA2000 IBM Interface Circuit, incompatible con los equipos actuales por utilizar un conector ISA, por un hardware de comunicación que nos diera más independencia de la arquitectura del computador y nos permitiera modernizar el trabajo con los robots MA2000. A continuación se muestran los principales resultados de este Proyecto de Fin de Carrera donde se comprueba el grado de cumplimiento de los objetivos planteados:



Ilustración 8: Conexión Sistema Completo

- Desarrollo de un componente hardware que sustituya a la tarjeta MA2000 IBM Interface Circuit. Con la ayuda del profesor Andrés Vicente Lober se llevó a cabo el desarrollo de la placa MA2000 USAL Interface Circuit y su software asociado que permite sustituir el hardware de comunicación si que los demás elementos que intervienen en el sistema se vean realmente afectados.
- Se ha probado el funcionamiento de la placa MA2000 USAL Interface Circuit y pese a haber encontrado ciertas limitaciones asociadas al PIC utilizado y a la gestión del puerto serie por parte del sistema operativo, se ha demostrado que el problema que planteaba el uso de la tarjeta MA2000 IBM Interface Circuit por su conector ISA se puede solucionar de este modo.
- Se ha conseguido mantener el software que gestionaba los movimientos del robot desde el computador, el cambio del hardware de comunicación ha sido completamente transparente a este software; gracias a que se mantuvo el API de las funciones de lectura y escritura sobre la memoria del robot.

Podemos extraer como principal conclusión que el objetivo principal de este proyecto se ha cumplido, conseguir la comunicación entre el computador y el robot, cambiando el hardware de comunicación y eliminando la limitación que suponía el uso de una tarjeta con un conector ISA; manteniendo además el software de alto nivel existente.