



**II CAIM 2010**  
**Segundo Congreso Argentino**  
**de Ingeniería Mecánica**  
**San Juan - Noviembre 2010**

## **Controlador de tres ejes para manipulador**

Cirera Eduardo A. , Bravo, Gerardo A.

**Departamento de Electricidad y Electrónica- Facultad de Ingeniería**  
**Universidad Nacional del Nordeste**

**Av. Las Heras 727 Resistencia – Chaco – Argentina**

Tel/Fax: +54-03722420076 - E-mail: [ecirera@ing.unne.edu.ar](mailto:ecirera@ing.unne.edu.ar) ; [gbravo@ing.unne.edu.ar](mailto:gbravo@ing.unne.edu.ar)

### **RESUMEN**

El trabajo surge del Proyecto de Investigación PI 0076/07 de la Facultad de Ingeniería de la UNNE en el cual se desarrolla el control de un Brazo robótico de tres articulaciones conducido mediante red tendonal.

En el mismo se presenta el diseño y la construcción de un Controlador de 3 ejes para su implementación en la conducción de motores paso a paso de gran potencia destinados al posicionamiento de un robot antropomorfo de tres articulaciones. En esta primera etapa del trabajo se logró construir el prototipo del controlador con sus periféricos y drivers de control. En una segunda etapa se trabajará sobre el algoritmo referente a las leyes de control que se implementarán para el funcionamiento del robot. Las características generales del mismo son:

El controlador fue diseñado con tecnología de control embebido mediante el uso de microcontroladores para la adquisición y procesamiento de datos. Para el almacenamiento de los mismos se cuenta con memorias de lectura escritura EEPROM.

Cuenta con un puerto de comunicaciones estándar RS232C, un display de visualización de parámetros y teclado 4x4 para su programación.

La interacción con el operador se realiza mediante menús de programación con la posibilidad de variación de parámetros como posición, velocidad, aceleración y distintos modos de trabajo.

La codificación fue hecha en lenguaje C, lo que da portabilidad al diseño, puesto que se prevé la migración hacia microcontroladores de gama más alta.

Lo positivo de este trabajo es que se ha logrado un diseño robusto en cuanto a los drivers de periféricos y a la electrónica adoptada, además buena operabilidad por medio de sus puertos y teclado, y la versatilidad que brinda el empleo de código propio en los microcontroladores, lo que lo hace actualizable y abierto a nuevas ideas.

**Palabras Claves:** Robótica, microcontroladores, Mecatronica.

## 1. INTRODUCCIÓN

Dentro de las tareas de investigación realizadas en la facultad de Ingeniería, se ha diseñado y construido un robot antropomorfo espacial de tres articulaciones. El mismo es un desarrollo con fines didácticos y se pretende estudiar con el mismo sus leyes dinámicas y de control. Para su movimiento, dispone de tres motores paso a paso de potencia con sus respectivos drivers de control. En esta etapa, el principal interés está dado en el controlador que gobernará los drivers de control y estos a su vez a los motores. Si bien el robot cuenta con encoders rotativos del tipo digital, en un primer momento, se implementó un control a lazo abierto del mismo, dejando para su posterior desarrollo, un dispositivo de lectura dedicado a los mismos, que actúe como esclavo de la unidad controladora. Se requiere entonces, como primera medida, la actuación sobre los drivers de los motores. Los mismos cuentan para su control tres señales digitales (3 bits) correspondientes al Avance de un paso, Dirección y Habilitación de los motores. Por ello es necesario un dispositivo capaz de proporcionar estas señales a los tres drivers, visualizar en una pantalla LCD las variables básicas del control, controlar localmente los motores mediante un teclado y gestionar una comunicación con un PC de escritorio con el que se interactúa.

Existen en la actualidad dispositivos industriales tales como los controladores programables PLC, capaces de controlar entradas y salidas discretas, con lenguajes de alto nivel y de gran robustez que si bien cubren las necesidades de este desarrollo, se prefiere para el control, un dispositivo programable más versátil, con un lenguaje de programación estándar como es el ANSI C. Esto conduce a la implementación de un dispositivo programable con tecnología de control embebido con un conjunto de periféricos diseñados a medida de la aplicación. La ventaja principal de la codificación en lenguaje C, es que da portabilidad al diseño, puesto que se prevé la migración hacia microcontroladores de gama más alta manteniendo la base de desarrollo.

## 2. ELECTRONICA

En base a los requerimientos se trabajó con un software de diseño electrónico denominado ISIS del paquete de desarrollo PROTEUS con el que se realizó primeramente el esquema circuital del controlador. Se simuló el circuito y luego, en base a este, se realizó el circuito impreso con el software ARES del mismo paquete. El controlador fue diseñado en dos partes separadas físicamente, en la primera, denominada *placa de control*, que contiene el microcontrolador y sus periféricos. La segunda, denominada *placa de Entradas-Salidas*, es una expansión de la primera y se logra mediante un bus de datos que las comunica.

### 2.1. Circuito de control

#### 2.1.1 Fuente de alimentación

La misma entrega una tensión regulada al circuito del controlador y se alimenta de una batería o fuente de corriente continua externa de 12 a 24 Voltios. Dispone de un regulador 7805 y capacitores de filtrado, además, un fusible de protección de 200mA. Ver Figura 1.

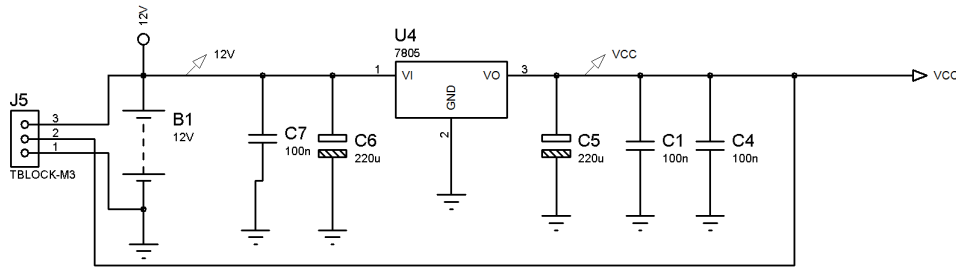


Figura 1 Fuente de alimentación.

### 2.1.2 Microcontrolador

El microcontrolador utilizado es el 16f877A de la firma Microchip, elegido en base a sus características de I/O (33 pines), frecuencia de operación (hasta 20Mhz), 4 temporizadores, entradas analógicas de 10 bits de resolución, EEPROM de datos de 256 bytes, RAM de datos de 368 bytes y memoria de programa de 14.3 kbytes. Se ha previsto el encapsulado de 40 pines, el cual contiene la misma configuración de pines que la línea 18FXXX correspondiente a la gama alta de microcontroladores de 8 bits, previendo ampliar las capacidades del controlador a futuro. En la Figura 2 se pueden ver las asignaciones de pines en el microcontrolador.

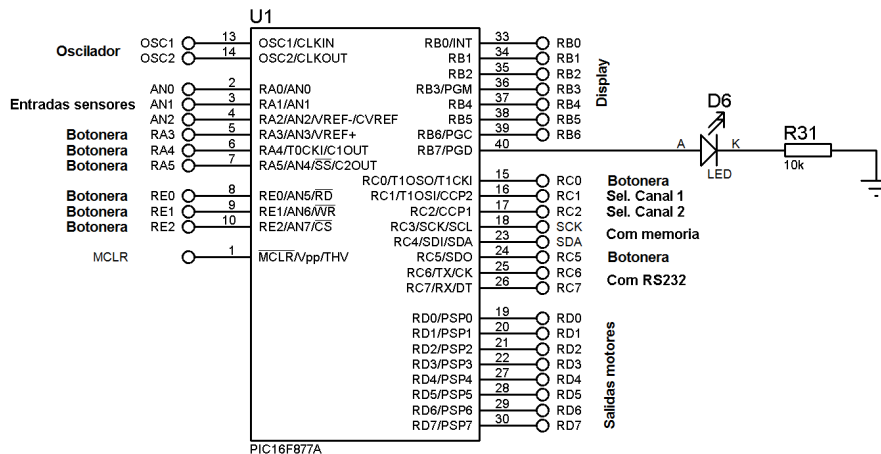


Figura 2 Asignación de pines.

### 2.1.3 Circuito oscilador y de reset

El circuito oscilador y de reset se realizó en base a la hoja de datos del microcontrolador, cuenta con un cristal de 20MHz y un push button para el reset del sistema. Ver Figura 3.

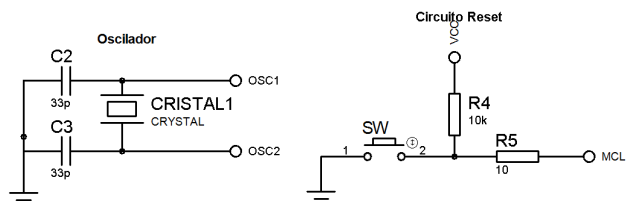


Figura 3 Circuito oscilador y de reset.

### 2.1.4 Módulo LCD

Para la visualización se dispuso de un módulo display de 2 líneas y 16 caracteres con backlight con la posibilidad de ajuste de contraste mediante un potenciómetro. La alimentación del mismo es de 5 voltios suministrados por la fuente. Se previó tanto la escritura como lectura de memoria del mismo. La comunicación con el microcontrolador se da en el modo de 4 líneas de datos y tres líneas de control correspondientes a *Selección de Registros, Lectura-Escritura y Habilitación*. Ver Figura 4.

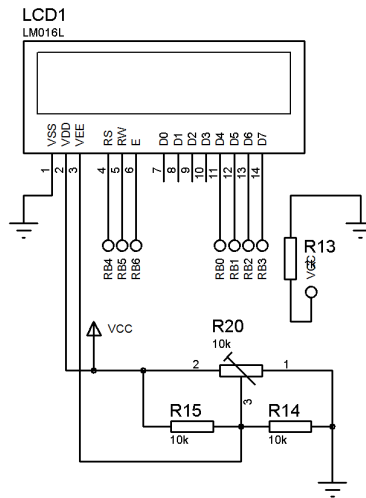


Figura 4 Display de visualización.

### 2.1.5 Módulo de comunicaciones

Para comunicar el controlador a una PC u otro dispositivo de control, se uso uno de los módulos UART que dispone el microcontrolador y mediante el integrado MAX232 se adecuan los niveles de señales TTL provenientes del microcontrolador a los  $\pm 15$  voltios que requiere la interfaz RS232C. El modo elegido fue Half Duplex y las características de comunicación son: 9600 baudios, 1 bit de parada, sin paridad y sin control de flujo. En la Figura 5 se aprecian las señales provenientes del microcontrolador (RC6 y RC7) y las que van hacia el conector DB9 macho. El integrado MAX232 contiene un oscilador interno que convierte la alimentación de 5 voltios a  $\pm 15$  voltios, esto lo logra gracias a los capacitores de 0.1uF conectados al mismo.

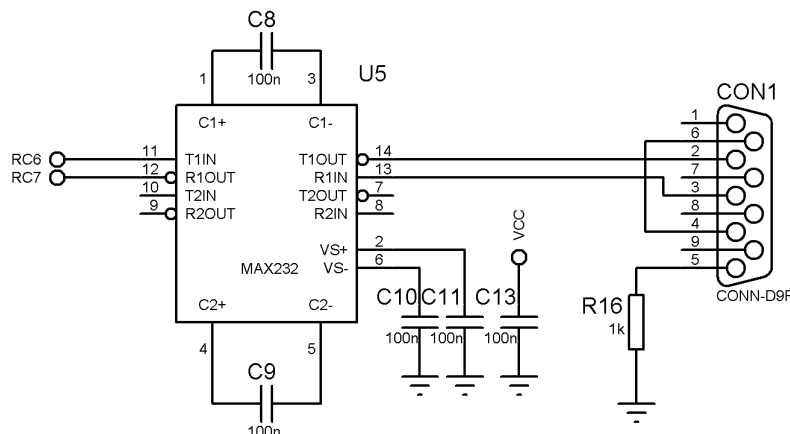


Figura 5 Módulo de comunicaciones Half Duplex.

### 2.1.6 Teclado

Para la carga de parámetros y control de movimiento local se ha previsto un teclado de 16 teclas, con 4 filas y 4 columnas. La lectura del teclado se realiza mediante 8 líneas. En la Figura 6, se aprecian las resistencias de Pull Up necesarias para su funcionamiento. Se han dispuesto dos tipos de conectores diferentes para mayor flexibilidad en el diseño.

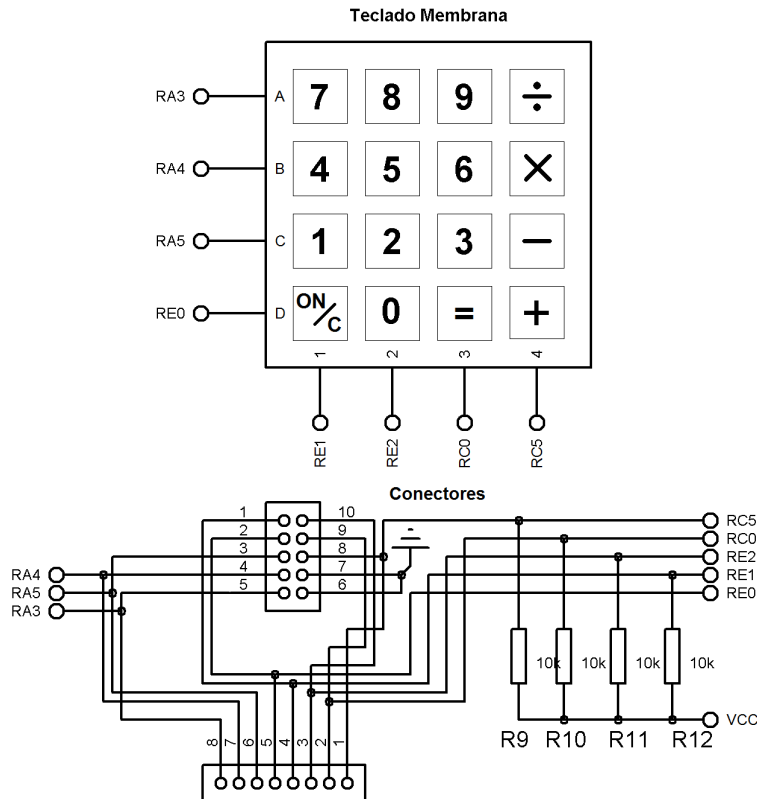


Figura 6 Teclado.

### 2.1.7 Entradas analógicas

Se han previsto 3 entradas analógicas-digitales previstas para la lectura de parámetros analógicos provenientes de una celda de carga o sensores de temperatura. Por ello, en cada bornera se entrega una tensión de referencia por medio de un regulador LM336 para generar una tensión estable e invariante con la temperatura en caso que se requiera. Previendo la posibilidad de una entrada externa diferente a la suministrada por el regulador se protegieron las entradas del microcontrolador mediante diodos zener de 5.1 voltios con sus respectivas resistencias reguladoras que impiden que la tensión supere los 5 Voltios y se dañe algún componente del controlador. En la Figura 7 se aprecian las borneras y su conexionado interno, conteniendo Alimentación de referencia(+), Alimentación Negativa(-) y señal. En la aplicación actual, se conectaron dichas entradas a sensores ópticos digitales con los que se referencia el robot en el momento de su utilización, quedando previsto el uso de las entradas analógicas en una aplicación a futuro.

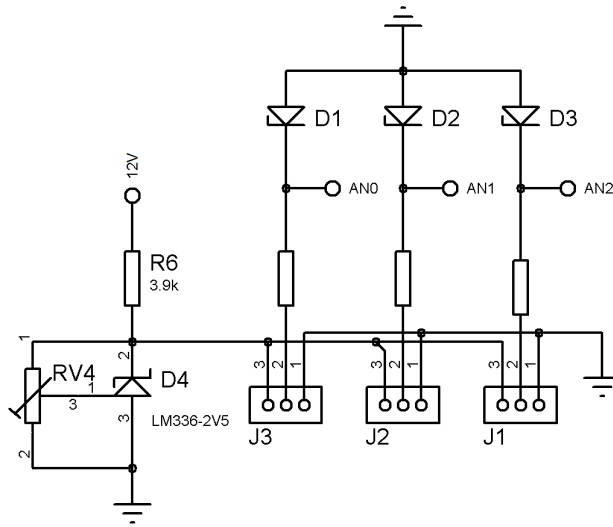


Figura 7 Entradas analógicas.

### 2.1.8 Memorias externas

Debido a la necesidad de almacenar trayectorias complejas en memoria para su posterior ejecución, se han incorporado 2 memorias externas del tipo serial cuya denominación es EEPROM 24LC256. Si bien el microcontrolador cuenta con 256 bytes (registros) de memoria EEPROM, esta resulta insuficiente en la mayoría de los casos. La comunicación del microcontrolador con tales memorias se realiza mediante el protocolo estándar I2C, puesto que se el microcontrolador dispone un módulo de hardware para tal fin. En la Figura 8 se puede ver el conexionado típico. La comunicación se da mediante las líneas SCK y SDA con sus respectivas resistencias de Pull Up. Las direcciones de cada una de ellas están configuradas mediante los pines A0, A1 y A2 y permiten el direccionamiento de hasta 8 memorias de este tipo. La capacidad que se logra con las dos memorias implementadas es de 64000 registros de 8bits.

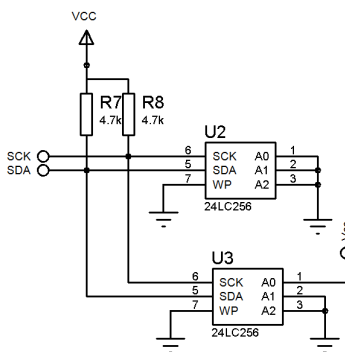


Figura 8 Memorias EEPROM externas.

### 2.2. Circuito de Entradas-Salidas

El circuito de Entradas-Salidas cuenta con 16 salidas digitales de potencia y 8 entradas digitales. El bus que se controla es de 8 bits de datos y 2 bits de control. Para ello hemos utilizamos la técnica de multiplexado  
Cirera, Eduardo A; Bravo, Gerardo A. / Controlador de tres ejes para manipulador.

de señales mediante circuitos integrados específicos. Para las salidas digitales, se utilizó el integrado 74HC573 y para las entradas digitales el 74LS245. Para la selección de estos integrados se realizó una lógica combinacional mediante los integrados 74HC08 y 74HC04. La misma fue diseñada en base a la tabla de verdad de la Tabla 1, teniendo como entradas RC1 y RC2 del microcontrolador y como salidas LE<sub>1</sub>, LE<sub>2</sub> y CE que son los pines de habilitación de los integrados conectados al bus de datos. Cabe aclarar que la selección de los integrados 74HC573 se realiza poniendo en alto los pines de habilitación LE<sub>1</sub> y LE<sub>2</sub>, mientras que en el 74LS245 se logra poniendo en bajo el Pin CE debido a su lógica inversa. Obsérvese que se activa cuando tiene un 0 lógico.

Tabla1 Tabla de Verdad.

RC2	RC1	LE <sub>1</sub>	LE <sub>2</sub>	CE
0	0	0	0	1
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

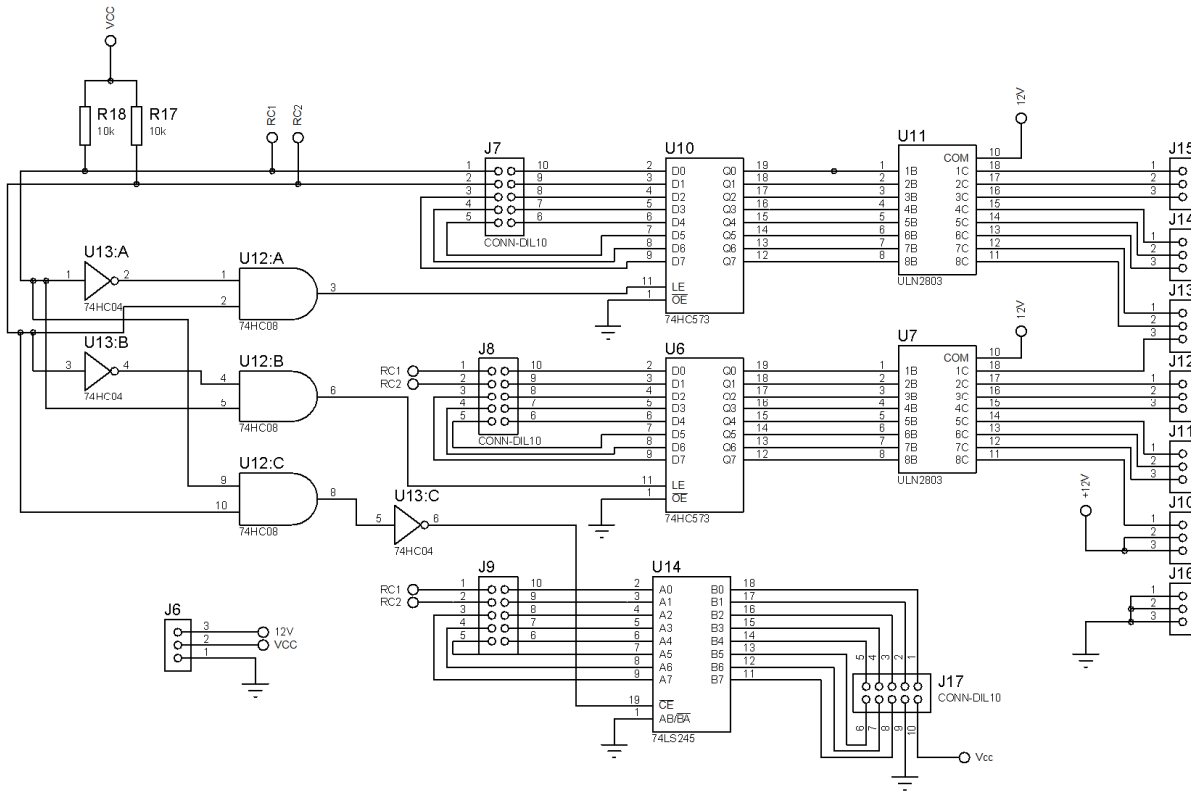


Figura 9 Circuito de Entradas-Salidas.

Para la amplificación de la capacidad de corriente y tensión de las salidas digitales, se utilizó un array darlington cuya denominación es ULN2803 y permite controlar cargas de hasta 500mA y 35 voltios. De las salidas disponibles, se utilizaron tres para el control de paso en los drivers, otras tres salidas para el control de dirección y una para el control de una electroválvula de una pinza neumática. Fue conveniente además, la deshabilitación por software de los drivers mediante otras tres salidas. Por otro lado, las entradas digitales se utilizaron para la conexión de un dispositivo esclavo capaz de administrar tres encoders digitales. Ver figura 9.

### 2.3. Circuito Impreso

El circuito impreso fue realizado en dos placas virgen de pertinax de 15x15cm. En una placa se realizó todo el circuito de control dejando en la otra, el circuito de Entradas-Salidas, comunicándose ambas mediante un cable plano de 10 hilos que hace de bus de datos. En la Figura 10 se observa el controlador terminado.

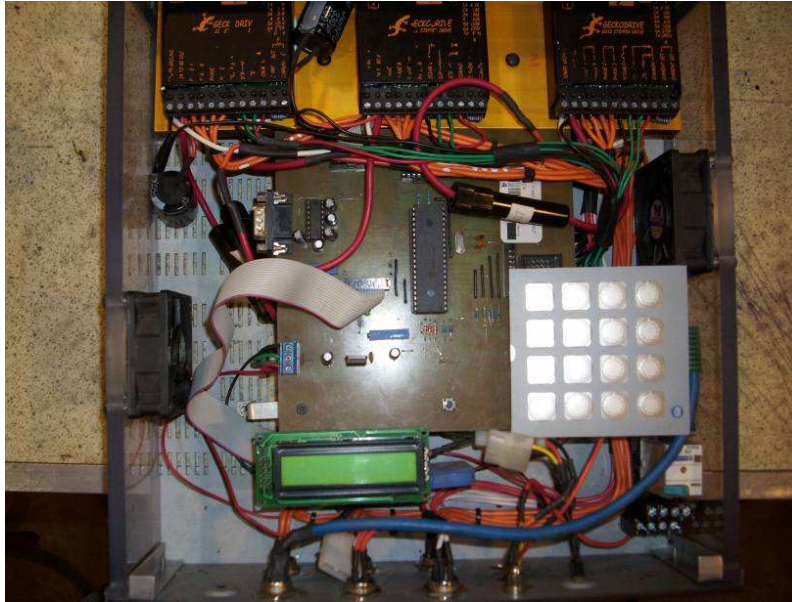


Figura 10a Circuito de control.

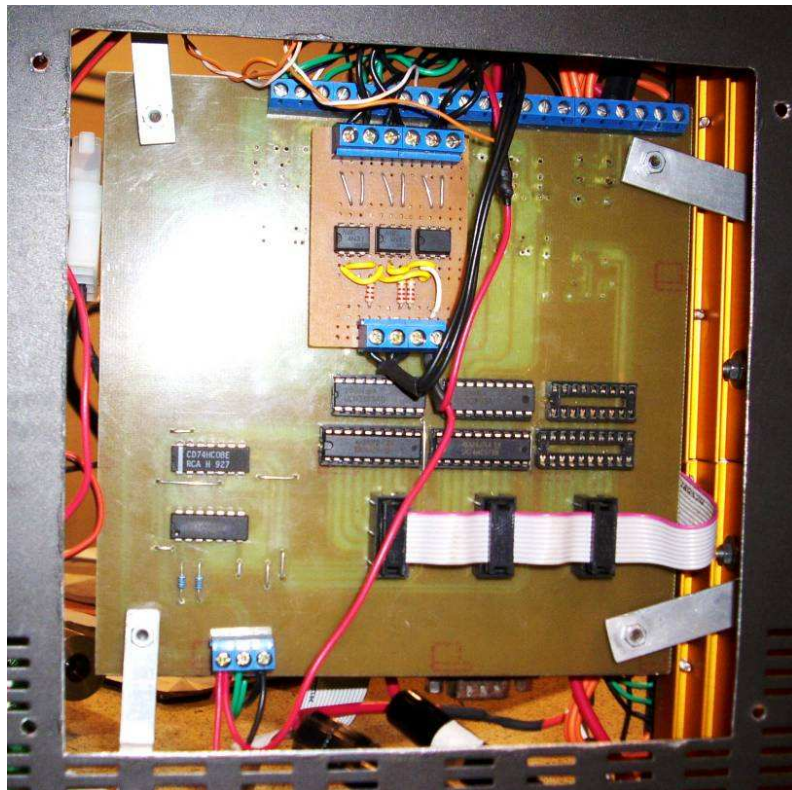


Figura 10b Circuito de Entradas-Salidas.

### 3. APLICACION

Cuando el controlador es iniciado, la aplicación diseñada cuenta con dos posibles modos de trabajo que se escogen mediante una tecla de *selección*. Independientemente del modo de trabajo, el controlador debe ejecutar una orden de movimiento con cuatro parámetros bien definidos:

*Motor*: Parámetro entero que indica el motor que se ha de mover (1, 2 o 3).

*Dirección*: Parámetro lógico que indica la dirección en que ha de realizarse el movimiento (0 o 1 lógico).

*Tiempo entre pasos*: Parámetro entero, correspondiente al tiempo entre pasos que indirectamente indica la velocidad articular del motor. Por otro lado, si hacemos que esta variable cambie en el tiempo, modificamos su aceleración articular. (100 a 5000microsegundos).

*Nº pasos*: Parámetro entero que indica el número de pasos que ha de moverse el motor seleccionado (0 a 5000 pasos).

#### 3.1 Modo Manual

En base a la selección manual de los parámetros mencionados se ejecuta el movimiento del motor seleccionado.

El Modo Manual permite el accionamiento de los motores individualmente, es decir, sin sincronismo entre ellos y mediante teclas de avance y retroceso, se ejecutan funciones en el controlador con los parámetros definidos previamente y cargados desde el teclado. Estas funciones se encargan de generar un tren de pulsos hacia los drivers de los motores y con ello lograr su movimiento.

Para la carga de parámetros tales como Tiempo entre pasos y Nº pasos, se disponen de teclas para su incremento y decremento.

#### 3.2 Modo Automático

El controlador hace de esclavo en una comunicación RS232 y recibe órdenes de un PC, el cual envía secuencialmente los parámetros Motor, Dirección, Tiempo entre Pasos y Nº pasos. Luego, el controlador ejecuta la petición si esta es válida.

Debido a las características inerciales del robot y al movimiento discreto de los motores empleados, el movimiento es mayormente rígido y se generan oscilaciones cuando se pasa del motor detenido a velocidad máxima y viceversa. Esto es producto de las elevadas aceleraciones en los transitorios y disminuye considerablemente la capacidad de manejo de carga del robot, puesto que la mayor parte corresponde a esfuerzos inerciales. La solución de este problema fue la incorporación de una rampa de aceleración que alcance la velocidad prefijada en el movimiento en un determinado número de pasos. Con esto se ha logrado un movimiento suave, disminuir considerablemente las vibraciones y aumentar la capacidad de carga de los motores.

En cuanto al modo automático, se ha implementado un programa en Matlab y se ha incluido en este la cinemática inversa del robot. Así, realizamos un preproceso donde es posible la interpolación del extremo del robot entre dos puntos definidos  $(x_0, y_0, z_0)$  y  $(x_1, y_1, z_1)$ . Luego, esta información es convertida en una serie de órdenes a enviar al controlador para conseguir el sincronismo de los motores y que el extremo del robot

se mueva por la función escogida entre dichos puntos. Hasta el momento se ha experimentando con funciones del tipo lineal y cuadrática.

Por otro lado, se ha implementado en el controlador el cálculo de la cinemática directa, permitiéndonos mostrar en un display la posición actual del extremo del robot manipulador. Esto se logra referenciando inicialmente el robot y luego, llevar una cuenta de los avances y retrocesos que han sufrido los motores.

En el diagrama de la Figura 11 se muestra en forma sintética la aplicación del controlador, en donde se puede ver como se ejecutan los modos Manual y Automático y la carga de parámetros.

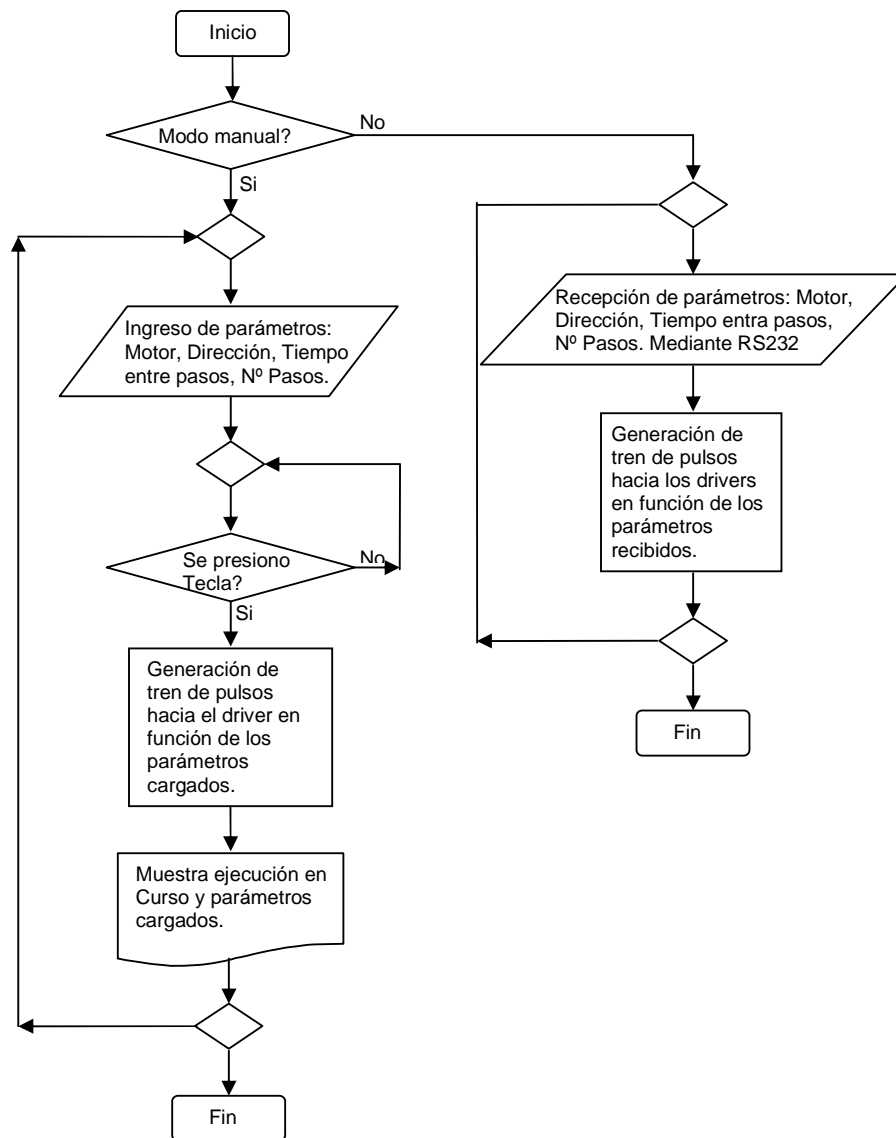


Figura 11 Diagrama de flujo.

#### 4. CONCLUSIONES

Se ha logrado un diseño robusto en cuanto a la electrónica adoptada y el software desarrollado, así como buena operabilidad por medio de sus puertos y teclado. Además, el hecho de que pueda ser expandido permite la modificación de la placa de potencia, manteniendo el circuito de control. La aplicación puede ser actualizada cuanto uno quiera, permitiendo utilizar el sistema con fines didácticos. Por otro lado, los

resultados experimentales obtenidos hasta el momento han sido muy satisfactorios, lo cual abre nuevas alternativas de control y diseño.

## 5. REFERENCIAS

- [1] Microchip, *Embedded Control Handbook vol. 1*, Microchip, USA , 1997.
- [2] Palacios E. y otros, *Microcontrolador PIC16F84*, Ra-ma, Mexico, 2006.