

**DESARROLLO E IMPLEMENTACION DE UN AUTOMATA
MONTACARGAS**

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades
Departamento de Ingeniería Electrónica



DESARROLLO E IMPLEMENTACION DE UN AUTOMATA
MONTACARGAS

FRANCISCO JAVIER RODRIGUEZ MAZARIEGOS

Trabajo de graduación presentado para optar
al grado académico de Ingeniería Electrónica



Guatemala
1998

Vo. Bo. :

(f) Manuel A. López V
Dr.-Ing. Manuel López
Asesor

Tribunal :

(f) Manuel A. López V
Dr.-Ing. Manuel López

(f) Gonzalo Palarea
Ingeniero Gonzalo Palarea

(f) José Luis Alvarado
Ingeniero José Luis Alvarado

Fecha de aprobación: 30 de noviembre de 1998

A mis padres y hermanos.
A todos aquellos que me ayudaron
a realizar este sueño.

RESUMEN

El objetivo principal del presente trabajo es diseñar e implementar un autómata montacargas que permita a los estudiantes poner en práctica los conocimientos aprendidos en los cursos de Sistemas de Control. Para realizar este trabajo se diseñó e implementó una solución con los recursos disponibles por la Universidad del Valle de Guatemala, o con aquellos que fueran de fácil localización en Guatemala.

La funcionalidad del autómata estará enfocada en el movimiento de sus diferentes ejes, de forma que éstos permitan el movimiento de tarimas entre estanterías y estaciones de trabajo.

El presente trabajo cubre cada uno de los aspectos del funcionamiento del autómata, el proceso de diseño e implementación.

CONTENIDO

	Páginas
Resumen	VII
I. Introducción	1
II. Objetivos	3
III. Antecedentes	5
IV. Método de trabajo	7
A. Estructura	9
B. Fuente de voltaje	15
C. Circuito de control de la potencia	27
D. Motores	47
E. Sistema de comunicación	49
F. Sistema de control	61
G. Controlador lógico programable con PIC 17C44	71
H. Programación	75
V. Resultados	83
VI. Recomendaciones	85
VII. Bibliografía	87

LISTA DE GRAFICAS

Gráfica	Páginas
4.1 Perfil	9
4.2 Elevación	10
4.3 Planta	10
4.4 Sistema de rieles	11
4.5 Mecanismo para cambio de vías	11
4.6 Aislamiento	12
4.7 Posición de sensores verticales	13
4.8 Posición de los sensores	14
4.9 Posición de los elementos electricos	14
4.10 Circuito de la fuente	16
4.11 Configuraciones transformador 120:12	17
4.12 Circuito transformador 120:12	17
4.13 Circuito de la fuente de 16.5 VDC	18
4.14 Operación de la fuente de 16.5 VDC	18
4.15 Circuito impreso de la fuente de 16.5 VDC	20
4.16 Circuito 120 VAC / UPS	21
4.17 Circuito de la fuente de 5 VDC y 24 VDC	22
4.18 Conexión a la fuente d 5 VDC y 24 VDC	23
4.19 Circuito UPS DC	24
4.20 Circuito impreso UPS DC	24
4.21 Unidad UPS DC	25
4.22 Disposición elementos de la fuente	26

4.23	Unidad UPS DC	25
4.24	Disposición elementos de la fuente	26
4.25	Relación potencia / ancho de pulso	28
4.26	Circuito de potencia	29
4.27	Aislamiento eléctrico	30
4.28	Función ECG3086	31
4.29	Etapa de potencia	31
4.30	Etapa de potencia encendida	32
4.31	Etapa de potencia encendida (dirección inversa)	33
4.32	Etapa de control	33
4.33	Funcionamiento de la etapa de control	34
4.34	Circuito impreso	36
4.35	Etapa de potencia no utilizada: Etapa DC con convertidor D/A	37
4.36	Etapa de potencia no utilizada: Etapa DC con fuente única	39
4.37	Etapa de potencia no utilizada: Etapa PWM sin circuito de control	40
4.38	Simulador: Señal de entrada	41
4.39	Simulador: Terminal uno	41
4.40	Simulador: Terminal dos	41
4.41	Simulador: Voltaje en colector del ECG3086	42
4.42	Funcionamiento de la etapa de potencia: motor apagado	43
4.43	Operación destructiva de la etapa de potencia	44
4.44	Elementos del sistema de comunicación	49
4.45	Ventana principal programa Puerto01	52
4.46	Radio modem 3Jtech	54

4.47	Fuente 5 VDC para el radio modem	55
4.48	Algoritmo de recepción	59
4.49	Elementos del sistema de control	62
4.50	Funcionamiento unidad Sharp GP1R23R	63
4.51	Relación tren de pulsos / velocidad angular	63
4.52	Relación tren de pulsos / dirección	64
4.53	Adaptador de voltaje para las señales	64
4.54	Funcionamiento del contador HC I	66
4.55	Elementos del sistema de actuación	67
4.56	Circuito compuertas AND	69
4.57	PLC con PIC 17C44 (OTP)	72
4.58	Control PLC principal sobre alimentación de tarjetas	75
4.59	Programación contador y salida PLC principal	76

LISTA DE TABLAS

Tabla		Páginas
4.1	Especificaciones	15
4.2	Especificaciones	27
4.3	Respuestas de la tarjeta	27
4.4	Especificaciones motor horizontal	47
4.5	Especificaciones motor vertical	48
4.6	Protocolo de comunicación	49
4.7	Especificaciones computadora	50
4.8	Elementos de programa	53
4.9	Especificaciones de los radio modems	54
4.10	Conexiones en el convertidor	56
4.11	Byte y bits para configurar el contador HC1	65
4.12	Funcionamiento PWM	68
4.13	Características PIC17C44	71
4.14	Direcciones PLCs	75
4.15	Instrucciones PLC 1	77
4.16	Instrucciones PLC 2 o 3	78
4.17	Respuesta PLC 2 o 3 al comando 11	80
4.18	Respuesta PLC 2 o 3 al comando 21	81

I. INTRODUCCION

Actualmente los cursos de Sistemas de Control no cuentan en el laboratorio con un equipo que permita poner en práctica y estudiar lo aprendido en la clase teórica, limitándose únicamente a la realización de maquetas y modelos en los cuales no son patentes los problemas funcionales que se presentan en las aplicaciones reales.

La ausencia de mano de obra calificada y con experiencia en el desarrollo de sistemas automáticos en el mercado Guatemalteco genera desconocimiento de estas aplicaciones y limita su utilización a aquellas compañías que pueden importar la tecnología y la mano de obra desde el extranjero.

Ante esto, el funcionamiento de este automático y su futura utilización por parte de los alumnos desarrollará ingenieros con la disposición y el conocimiento necesario para implementar comercialmente aplicaciones automatizadas. Al mismo tiempo demuestra a la industria nacional que es posible y funcional desarrollar diseños e implementar soluciones automatizadas, con la mano de obra local y con los recursos disponibles.

Ante esto, se diseñaron e implementaron las diferentes partes que permiten la funcionalidad de un automático montacargas:

- La estructura necesaria para manejar las tarimas y que permita los diferentes movimientos de automático.
- La fuente de alimentación que provee los voltajes necesarios para que cada uno de los elementos de automático funcione.
- La transformación de potencia con la cual la señal de los PLC puedan encender y apagar los motores en la dirección y con la velocidad deseada.
- Los motores y las modificaciones que fueron necesarios para que funcionaran adecuadamente.
- El sistema de comunicación implementado, que mantiene un enlace permanente con el automático no importando donde se encuentre, dentro del área de trabajo.

- El sistema de Programmable Logic Controllers (PLCs), los diferentes sensores y los actuadores que permiten determinar y modificar la posición y la velocidad del autómeta en cada uno de los ejes.
- Finalmente la programación actual del autómeta.

El lector encontrará en este trabajo todos los aspectos funcionales de cada una de las partes que componen el autómeta, así como los detalles de los modelos que no funcionaron durante el desarrollo.

Se busca así que este documento se convierta en un manual de referencia para todos aquellos alumnos que utilizarán este dispositivo en el laboratorio de Sistemas de Control II, y para aquellos otros que quieran continuar desarrollando este autómeta.

II. OBJETIVOS

El objetivo de este trabajo es desarrollar e implementar un autómata montacargas de 3 ejes, utilizando recursos nacionales. Este autómata debe trabajar ejemplarmente una bodega con estanterías profundas o de alta densidad.

Los objetivos específicos del trabajo son:

1. Desarrollar un autómata que funcione como laboratorio del curso Sistemas de Control II, que en él se puedan medir parámetros de control, modelar filtros digitales y probar la respuesta del sistema a estos filtros.
2. Desarrollar la estructura mecánica del autómata, la cual debe moverse en 3 ejes y permitir recoger y entregar tarimas en la bodega.
3. Desarrollar un sistema de alimentación, que permita el funcionamiento de forma automática del montacargas. Para ello debe proveer la potencia necesaria para el funcionamiento, no importando la posición donde se encuentre.
4. Desarrollar un sistema de comunicación mediante radio módems que permita un enlace permanente entre el autómata y una computadora personal. Esta comunicación inalámbrica debe permitir al autómata transmitir y recibir datos sin importar su posición o la velocidad con la que se mueva.
5. Desarrollar un sistema que transforme las señales de control en una señal de potencia, con la cual se pueda prender, modificar la velocidad, y/o apagar los motores.
6. Desarrollar un sistema de control para el autómata: controladores programables (PLC), sensores, actuadores y algoritmos. Este sistema deberá permitir un movimiento controlado en cada eje, así como la capacidad de alcanzar una posición específica.
7. Desarrollar los controladores programables utilizando microprocesadores PIC.
8. Garantizar la seguridad de los datos del sistema de control. Todos los datos del sistema deben estar asegurados y siempre deben ser confiables.

III. ANTECEDENTES

Los antecedentes de este proyecto se remontan a tres años atrás. 1996, cuando en el laboratorio de Sistemas de Control II, los alumnos tuvieron como objetivo implementar un autómeta. El autor de este trabajo fue parte de este grupo.

En esta ocasión se desarrolló una estructura funcional cuyos detalles fueron:

- Movimiento en tres dimensiones: horizontal, vertical y profundidad.
- Motores de reciclados (limpiabrisas de automóvil) en cada uno de los ejes.
- Alimentación de voltaje positivo (+12 V) y negativo (-12V).
- Control exterior a través de un PLC Siemens colocado fuera del autómeta.
- Sensores de posición externos al autómeta

El desarrollo de este autómeta nunca se terminó y todos los detalles del funcionamiento se perdieron; lo único que existía al iniciar este trabajo eran la estructura y los motores.

Luego en 1997 se formó un grupo integrado por Minor Joaquín, Marco Antonio Hernández, Oscar Melchor y nuevamente el autor de este trabajo. Se trabajó en modificar la estructura del autómeta para hacerla operativa. Nunca se obtuvo una estructura funcional.

Adicionalmente, el mismo año, los alumnos del Laboratorio de Sistemas de Control II desarrollaron aplicaciones con microprocesadores PIC, cuya función era controlar la posición y la velocidad del autómeta en los ejes. Los resultados obtenidos de estos trabajos se detallan en el capítulo 4 sección F.

Las soluciones desarrolladas por los alumnos no se utilizaron pues, los microprocesadores no tenían la capacidad matemática para mejorar los diseños.

A partir de enero de 1998 el autor de este trabajo inicia el desarrollo y la implementación del autómeta, con todos los conocimientos adquiridos durante 1996 y 1997. El trabajo se inició desde cero y no se utilizaron elementos o diseños de 1996 y 1997, con excepción de un motor.

IV. METODO DE TRABAJO

El método de trabajo utilizado durante este trabajo consistió en diseñar, implementar y probar los diferentes elementos que se necesitaban para hacer funcionar el autómata. Conforme se realizaban las pruebas, se hacían modificaciones en los diseños y nuevamente se probaban estas modificaciones.

A continuación aparecen el capítulo 4 secciones de la A a la G donde se detalla cada parte que compone el autómata. Estos capítulos están organizados de la siguiente forma:

- Parte 1, descripción del funcionamiento.
- Parte 2, especificaciones técnicas.
- Parte 3, circuito eléctrico y su funcionamiento.
- Parte 4, circuitos anteriores realizados durante el desarrollo.

Hay capítulos en los que aparecen partes adicionales, las cuales son detalles importantes que deben ser considerados por aquellos lectores que quieran utilizar el autómata.

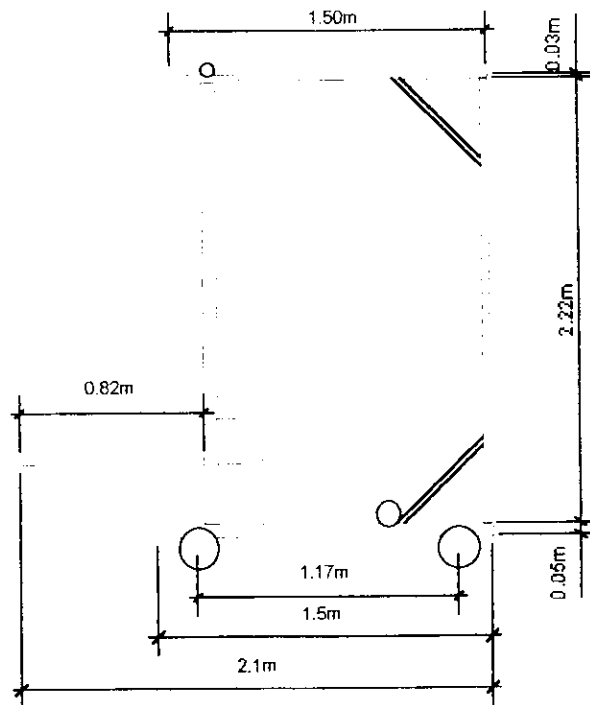
A. ESTRUCTURA

El fundamento de la operación del autómata es su estructura; como la función del autómata es ser un montacargas, su estructura es lo más parecido a un montacargas estándar.

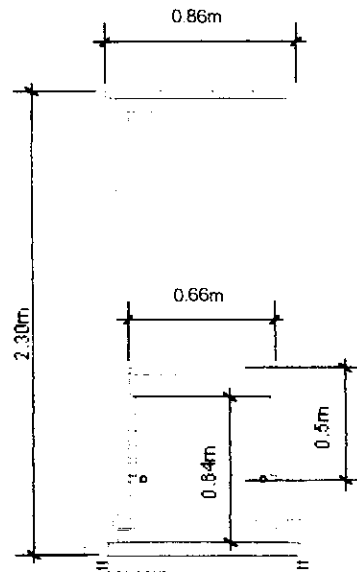
Actualmente el autómata cuenta con tres ejes de movimiento: horizontal, vertical y rotación. Esto permite intercambiar tarimas entre dos posiciones: recogiendo una tarima de una zona de carga y descarga, y entregándola en una estantería.

Además, el diseño del autómata permite aumentar sus ejes hasta cinco: agregando movimiento de profundidad en su tenedor, y desarrollando un sistema de cambio de vías. Estos ejes aunque no están desarrollados, sí pueden ser incorporados, tanto desde el punto de vista mecánico como del electrónico.

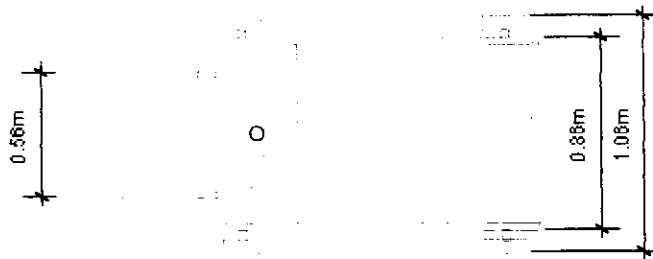
Las dimensiones del autómata son: alto 230 cm; ancho 106 cm; largo 210 cm (desde la punta de la tenaza) ó 150 (desde el inicio de la base). Su diseño es el siguiente:



Gráfica 4.1 "Perfil"



Gráfica 4.2 "Elevación"



Gráfica 4.3 "Planta"

A.1 Espacio y capacidad de carga

El espacio de carga es la parte fundamental del autómata montacargas, la razón final de su funcionamiento. En este espacio de carga se lleva el elemento transportado.

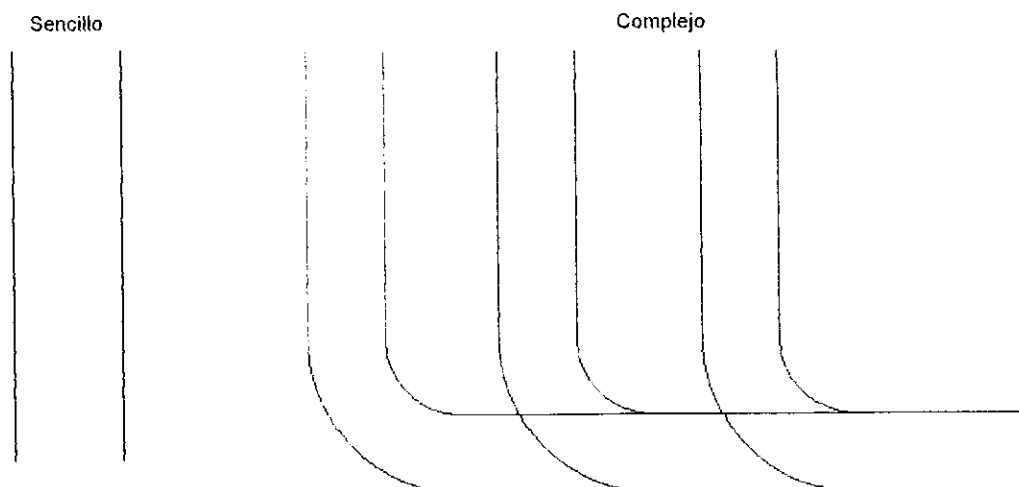
Actualmente la mayoría de la carga se almacena en las bodegas colocada en tarimas, las cuales tienen unas dimensiones estándar. Fue una premisa de diseño que el autómata pueda manejar estas tarimas, cuyas dimensiones son: alto 0.06 m; ancho 1.20 m; y profundidad 1 m.

Para manejar las tarimas el autómata cuenta con un tenedor compuesto por dos tubos de 87 cm de largo y $\frac{3}{4}$ de pulgada de diámetro, los cuales están separados 54 cm.

La capacidad máxima de carga del autómata es de 500 Kg. Todos los elementos del eje vertical pueden soportar este peso. Sin embargo el motor actual no tiene la potencia necesaria para elevar más de 22 Kg. Es la recomendación del autor, que únicamente se carguen tarimas y cajas vacías, aproximadamente 10 Kg. Si se desea aumentar la capacidad de carga se deberá sustituir el motor actual.

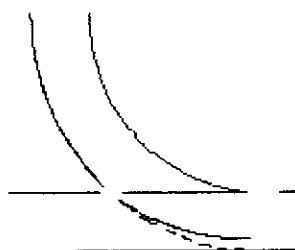
A.2 Eje horizontal

El eje horizontal del autómata lo constituyen los rieles y sus ruedas. Actualmente está formado de un par de rieles rectos que mide 6 m. Sin embargo la estructura del autómata permite trabajar sistemas de rieles más complejos.



Gráfica 4.4 "Sistema de rieles"

Utilizando un mecanismo de cambio de vías:



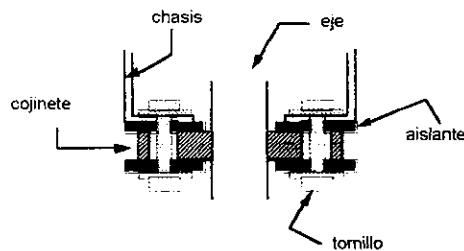
Gráfica 4.5 "Mecanismo para cambio de vías"

Y un eje delantero que cuenta con movimiento para adaptarse a las curvas.

Como se verá en detalle más adelante, la alimentación se realiza a través de los rieles. Por ellos se conduce el voltaje, y existe una diferencia de potencial de 12 VAC entre ambos.

Es muy importante que no haya conducción eléctrica entre las dos ruedas de un mismo eje, pues se produciría un corto circuito. Esto se evita eliminando los ejes corridos entre cada par de ruedas. Se utilizan un eje corto y un cojinete para cada rueda.

El conjunto rueda, eje y cojinete está aislado eléctricamente del chasis, mediante el uso de tornillos y aislamiento de plástico. En la siguiente gráfica se puede apreciar mejor la configuración:



Gráfica 4.6 "Aislamiento"

Para eliminar las interrupciones en la alimentación, las cuales se pueden producir cuando la rueda no está en contacto con el riel, se conectan ambas ruedas del mismo riel con un cable AWG10.

Otra función que se realiza en cada eje es medir la posición. Tanto el diseño del eje como de la medición están muy relacionadas.

Para medir la posición se cuenta con sensores de desplazamiento, el detalle de su funcionamiento aparece en el capítulo 4.6. En cada eje existen dos sensores, uno que indica una posición específica y otro que cuenta el desplazamiento desde y hacia esa posición.

Para el eje horizontal, el sensor que mide el desplazamiento va contando el número de vueltas que da una rueda. Este sensor se coloca en una rueda que no es impulsada por el motor, porque si la fuerza del motor hace resbalar la rueda sobre el eje, se registrarán mediciones incorrectas de desplazamiento y velocidad. Se

ha observado cómo se resbala la rueda al momento de encender el motor, apagarlo o encontrar algún obstáculo en la vía.

El sensor de la posición específica es un micro switch que se presiona cuando el autómata llega a una posición. Normalmente esta posición se conoce como posición inicial, ya que cuando el autómata se enciende y desconoce dónde está ubicado puede ir a esta posición y desde allí iniciar todo el trabajo.

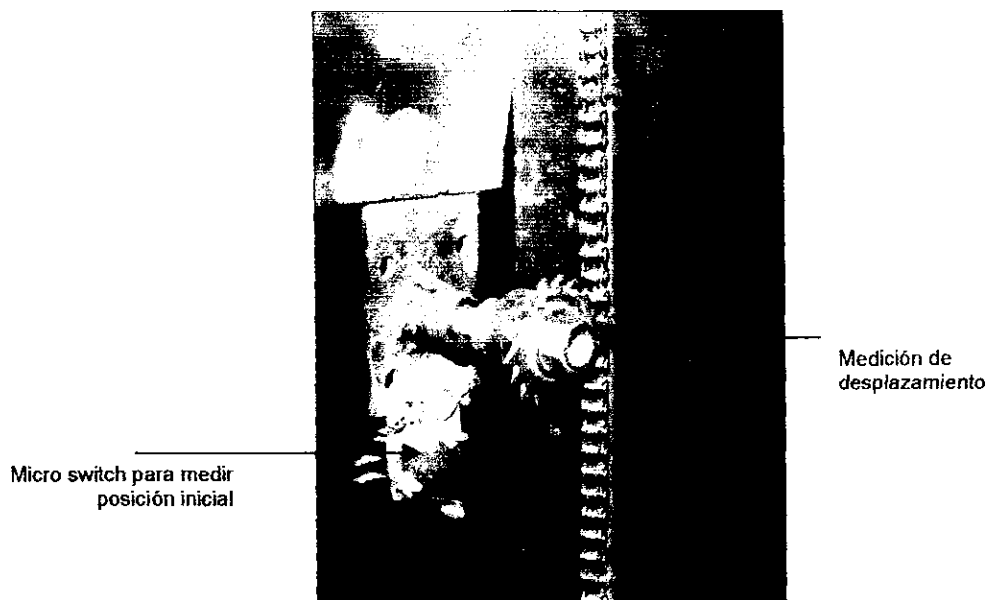
Para el caso del eje horizontal, la posición inicial es el principio de la vía, donde está ubicada una placa que acciona un micro switch colocado en la parte de abajo del autómata.

A.3 Eje vertical

El eje vertical del autómata se desplaza 120 cm. Está constituido por una pieza que gira completa como parte del eje de rotación.

Los sensores de este eje nuevamente están colocados de tal forma que miden el desplazamiento del eje y no el movimiento del motor. Utilizando una cadena y una rueda dentada, el sensor del movimiento registra únicamente los desplazamientos reales. Y la posición inicial se mide con un micro switch, que se acciona al llegar a la posición cero del eje vertical.

La ubicación de los sensores se observa en la siguiente gráfica:



Gráfica 4.7 "Posición sensores verticales"

A.4 Eje de rotación

El eje de rotación hace girar el tenedor del autómatas 180°.

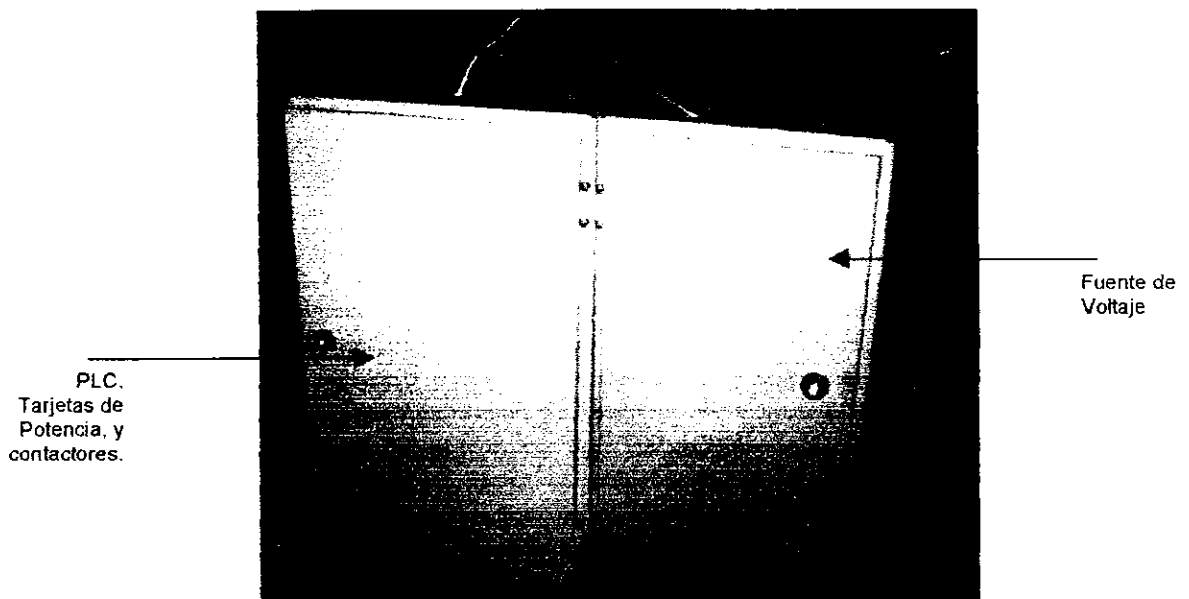
Al igual que en los dos ejes anteriores, se cuenta con un medidor de desplazamiento que está colocado en la parte superior de la estructura, acoplado directamente al eje de rotación, así como el micro switch que indica la posición cero:



Gráfica 4.8 "Posición de los sensores"

A.5 Posición de los elementos de control

Todos los elementos electrónicos están guardados y protegidos en dos cajas que se encuentran en la parte de atrás del autómatas. La ubicación de los elementos es la siguiente:



Gráfica 4.9 "Posición de los elementos electrónicos"

B. FUENTE DE VOLTAJE

La función del circuito de la fuente de voltaje es proveer al autómeta con los voltajes y la corriente necesarios para que cada uno de sus componentes funcione.

La fuente alimenta al autómeta a través de los rieles del mismo, y obtiene los diferentes voltajes mediante un grupo de transformadores. En los rieles está presente una diferencia de voltaje de 12 V AC. Este voltaje es tan bajo que permite una operación segura del autómeta, ya que cualquier persona puede tomar ambos rieles con las manos y no sufrir un golpe eléctrico.

Sin embargo, como ambos rieles están descubiertos existe la posibilidad de que se produzca un corto circuito si alguna pieza metálica cae sobre ellos.

La fuente debe operar bajo las siguientes premisas:

- Debe ser seguro para las personas que utilizan el robot.
- Debe proveer los voltajes y las corrientes necesarias para cada elemento del autómeta.
- Debe ser capaz de entregar esta energía al autómeta no importando en qué posición se encuentre.
- Debe reaccionar ante los cortos circuitos.

B.1. Especificaciones técnicas del proyecto para la Fuente de Voltaje

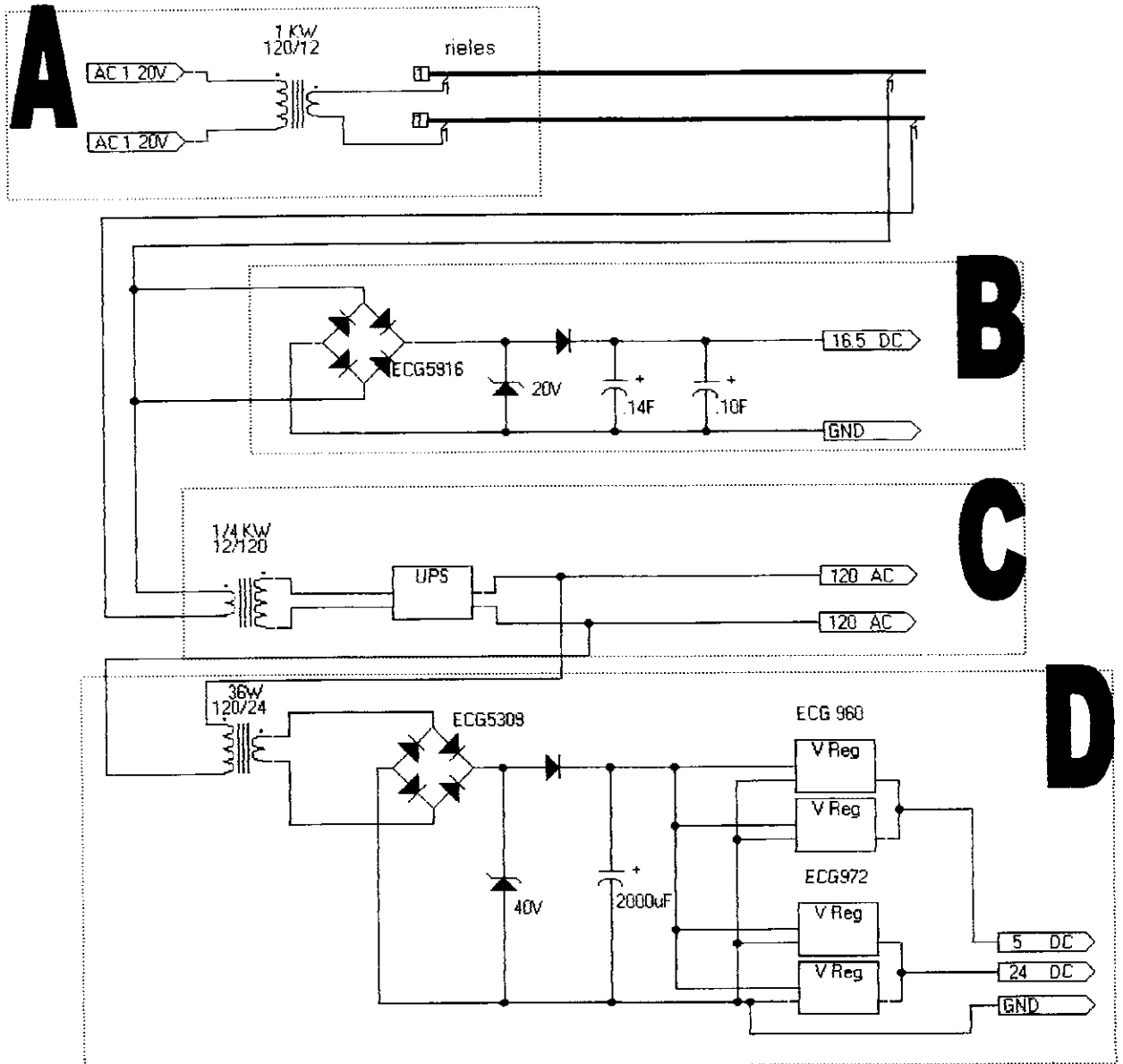
Alimentación		Voltaje	120 VAC
		Amperaje	20 A.
Salidas	#1	Voltaje	120 VAC
	Alimentación PLC	Amperaje	0.1 A.
	#2	Voltaje	24 VDC
	Alimentación PLC/Circuitos	Amperaje	2 A.
	#3	Voltaje	5 VDC
	Alimentación Circuitos	Amperaje	2 A.
	#4	Voltaje	16.5 VDC
	Alimentación Motores	Amperaje	20 A.
Señales de Entrada		Ninguna	
Señales de Salida		Ninguna	

Tabla 4.1 "Especificaciones"

B.2. Circuito de la Fuente

El circuito de la fuente se divide en las siguientes partes:

- A. Transformación de 120 a 12 V AC
- B. Fuente de 16.5 V DC
- C. Recuperación de 120 V AC
- D. Fuente de 24 y 5 V.

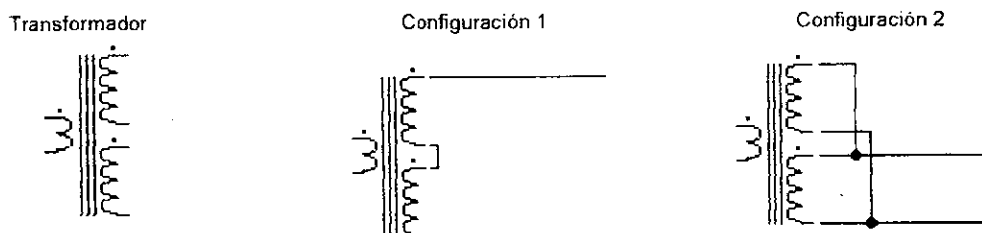


Gráfica 4.10 "Circuito de la fuente"

El funcionamiento de cada una de las partes es el siguiente:

A. Transformación de 120 a 12 V AC:

Para transformar el voltaje de 120 V a 12 V AC se utiliza un transformador de 1 KW, este transformador tiene un embobinado sencillo en la línea de 120 V y dos embobinados de 12 V. Estos dos embobinados se pueden utilizar en serie o en paralelo, dependiendo del voltaje que se desee obtener en la línea secundaria, 24 ó 12 VAC respectivamente:



Gráfica 4.11 "Configuraciones transformador 120/12"

La corriente varía con la configuración y estará dada por la siguiente ecuación de potencia ($I = P/V$):

- Configuración 1 (24 V) 42 A
- Configuración 2 (12 V) 84 A

Este transformador se utiliza en la configuración de 12 V y los 84 amperios que puede proveer es más de lo que el autómata consume.

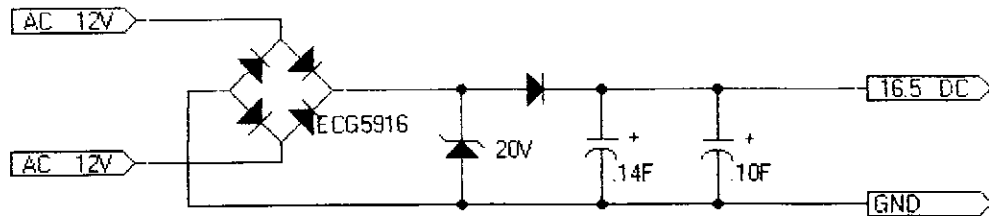
Como los rieles están descubiertos, el transformador está en peligro de arruinarse si ocurre un corto circuito. Por ello se colocaron protecciones en cada una de las salidas del transformador, dos brackers Siemens de 40 A.:



Gráfica 4.12 "Circuito transformador 120/12"

B Fuente de 16.5 V DC:

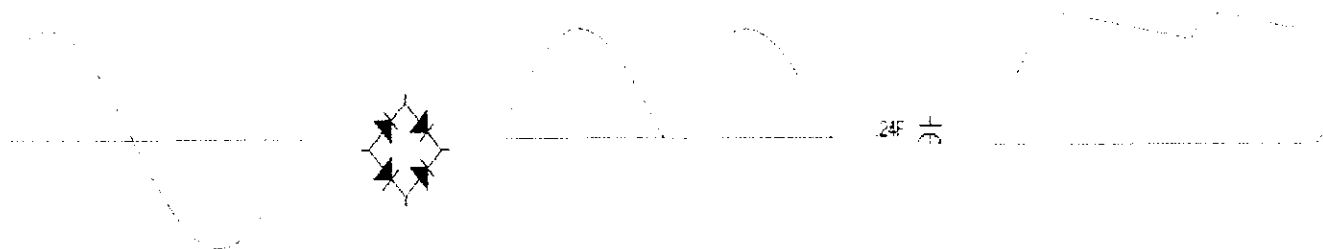
La fuente de 16.5 V es la fuente que alimenta los motores. Está constituida por los siguientes elementos:



Gráfica 4.13 "Circuito de la fuente de 16.5 VDC"

Cuatro diodos industriales ECG5916 y dos capacitores de 0.1 F y 0.14 F. Durante la operación de esta fuente, el voltaje alterno se convierte en una señal pulsante en el puente de diodos.

Esta señal no alternante se filtra y se mantiene a un voltaje muy cercano a 16.5 V mediante los dos capacitores que están en paralelo y funcionan como si fueran un único capacitor de 0.24 F. Gráficamente el funcionamiento de la fuente es el siguiente:



Gráfica 4.14 "Operación fuente de 16.5 VDC"

La fuente de 16.5 V no está regulada y funciona muy bien para alimentar los motores, ya que provee una corriente de 20 A, al mismo tiempo que tiene una componente capacitiva muy fuerte para eliminar el ruido generado por los motores DC (alto componente inductivo).

El límite máximo del amperaje de la fuente está limitado por la corriente máxima de los diodos industriales, en este caso 20 amperios. También existe un límite por la caída de voltaje en los capacitores cuando provee determinada corriente.

Para analizar esta caída, modelamos el circuito de carga / descarga de un capacitor. Se supone que inicialmente se aplica al capacitor un voltaje de 16.5 V y luego se descarga. El voltaje en las terminales estará dado por:

$$V(t) = V_{\max} (1 - e^{-t/RC}) \text{ durante un procedimiento de carga}$$

$$V(t) = V_{\max} (e^{-t/RC}) \text{ durante un procedimiento de descarga}$$

si es aceptable el 90% de valor nominal del voltaje, entonces el término $e^{-t/RC} = 0.9$

la relación $t/RC = 0.1$

$t = 8.33 \text{ E-3 segundos}$, que es el tiempo entre dos picos máximos

en la señal senoidal no alternante, $f = 120 \text{ Hz}$.

la resistencia se calcula en función del amperaje y el voltaje máximo. Para obtener 20 A, la resistencia

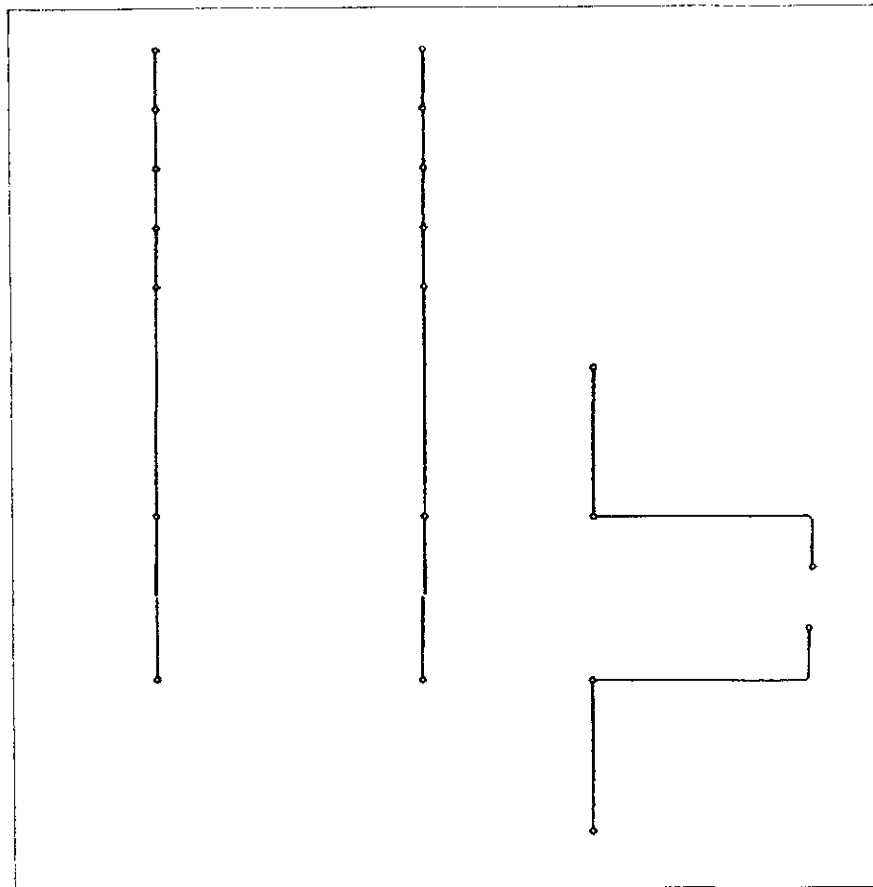
$$\text{debe ser } R = 16.5 \text{ V} / 20 \text{ A} = 0.825 \text{ } \Omega$$

por lo tanto $C = 0.1 \text{ F}$

El capacitor utilizado, 0.24 F sobrepasa las necesidades del sistema. Mas aún, no se consideró en este análisis que el tiempo durante el cual ocurre la descarga es menor, porque la señal es senoidal, y no cuadrada, como se supone en el modelo.

Para implementar el circuito impreso de la fuente, se utilizó el programa PCROUTE, cuyo autor Douglas E. Ehlers lo puso a disposición gratuita en la dirección de Internet: www.shareware.com

El diseño resultante fue el siguiente:



Gráfica 4.15 "Circuito impreso de la fuente de 16.5 VDC"

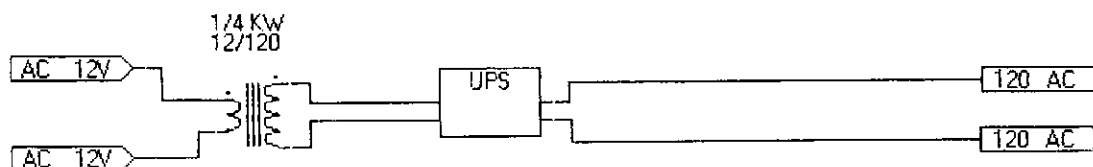
Únicamente se modificó el ancho del trazado, a 1 pulgada, para que resistiera los 20 amperios.

C. Recuperación de 120 VAC

Recuperar el voltaje de 120 VAC conlleva un aumento significativo de riesgo. 120 VAC aplicados a cualquier parte del cuerpo generan una sensación desagradable y peligrosa, no por un peligro de muerte por electrocutamiento sino porque la reacción al voltaje puede causar un accidente. Es por ello que todas las terminales y los cables que conducen este voltaje están debidamente señalizados y protegidos.

Se utiliza este voltaje a pesar del riesgo, porque los PLC Siemens necesitan 120 VAC.

El circuito utilizado es el siguiente:



Gráfica 4.16 "Circuito 120 VAC / UPS"

El transformador de 250 W recupera la señal original, al conectar su embobinado secundario a 12VAC.

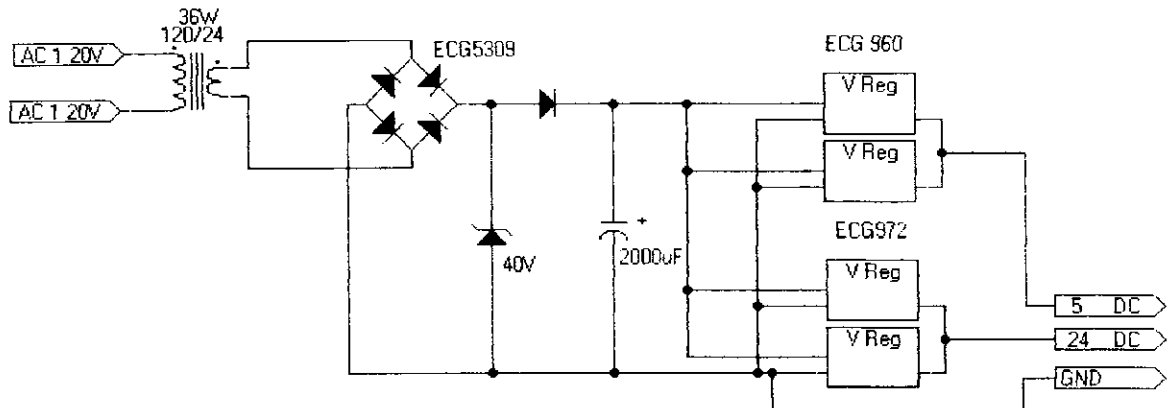
Como el lector se habrá dado cuenta, en el autómata la transmisión de la potencia se realiza a un voltaje muy bajo, con el fin de aumentar la seguridad por los rieles descubiertos. Esta configuración nunca se usa comercialmente por la pérdida de potencia que involucra. Sin embargo para la aplicación esta pérdida de potencia no es significativa.

El UPS conectado después del transformador se utiliza como una protección para la información de los PLCs, especialmente porque al estar moviéndose el autómata por los rieles existe la probabilidad de que en algún momento no haya contacto entre las ruedas y los rieles, y se pierda la alimentación. Si la información no está protegida, el autómata perdería los datos de su ubicación actual, dirección de destino y/o la velocidad.

Para evitar esto, se está empleando un UPS Powerware Personal 600 de Exide Electronics. El mismo necesita como alimentación una señal de 120 VAC, 6.0 A y 50/60 Hz, y tiene una salida de 117 VAC, 5.1 A, 600VA, 0.65pF y 50/60 Hz.

D. Fuente de 24 y 5 V:

Este circuito es utilizado para las proveer 24 VDC y 5 VDC a los elementos restantes: PLC auxiliar, sensores, adaptadores y otros circuitos TTL. Este circuito repite las partes A y B descritas con anterioridad. Prácticamente es la misma fuente de 16.5 VDC. La diferencia está en los reguladores de voltaje, con los que se obtiene una salida más confiable:



Gráfica 4.17 "Circuito de la fuente de 5 VDC y 24 VDC"

Los elementos son los mismos que las partes A y B, funcionan de la misma forma, sólo que las capacidades son más pequeñas y la conversión es 120/24 VAC. Esto genera un voltaje de 36 VDC no regulado en el capacitor de 200 µF.

Las otras diferencias con las partes A y B son:

- El transformador es 120/24 V AC, es decir, en su salida hay una señal de 24 VAC RMS.
- El puente de diodos está integrado en una sola pieza y no por cuatro diodos independientes. La corriente máxima que circula por el puente de diodos es de 2 A.
- El capacitor de 2000µF nuevamente tiene la capacidad de almacenar más carga de la necesaria, sólo que su cálculo se realizó con 2 A.

A partir de ese punto hay diferencias en el diseño, específicamente están presentes 4 reguladores de voltaje integrados, los cuales generan voltajes de 5 V y 24 V. Las especificaciones de estos dos tipos de reguladores se encuentran en un CD adjunto que contiene los archivos PDF y su lector.

Para organizar la conexión de todos los elementos que se necesitan de estos voltajes, se utilizaron como conectores: terminales de fuentes y tornillos. La configuración de los voltajes está determinada por el siguiente código de colores:

- Negro 0 V
- Rojo 5 V
- Verde 24V

Y la forma en que están ubicados se puede ver en la siguiente gráfica:



Gráfica 4.18 "Conexión a la fuente de 5 VDC y 24 VDC"

B.3. Desarrollo de la fuente

Para desarrollar la fuente se trabajaron varios modelos, especialmente porque los requerimientos fueron cambiando cuando cambiaban los elementos del autómata.

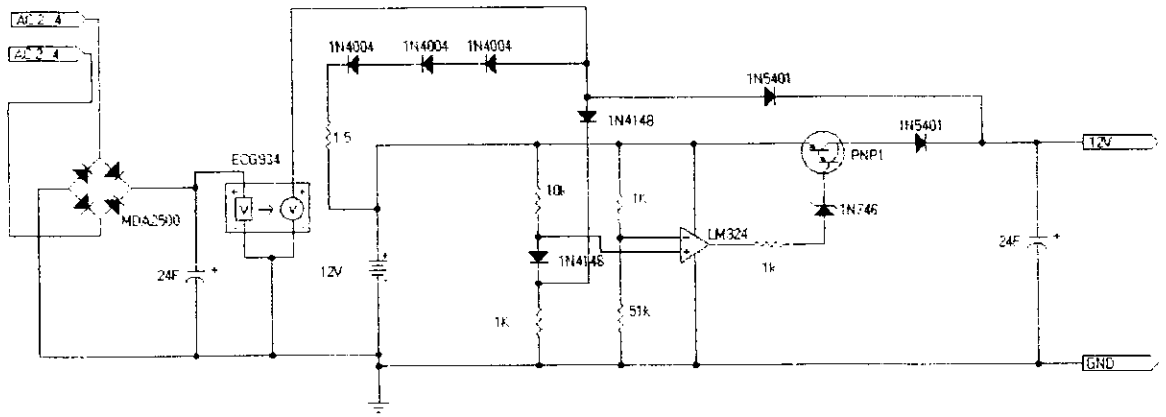
A. Fuente de voltaje con baterías:

La primera fuente de voltaje utilizada empleaba baterías, dos baterías de carro para generar los voltajes +12, 0 y -12 V. Se descartó esta fuente porque las baterías había que recargarlas constantemente y los tiempos que el autómata podía estar funcionando eran muy cortos.

B. Fuente actual con UPS DC:

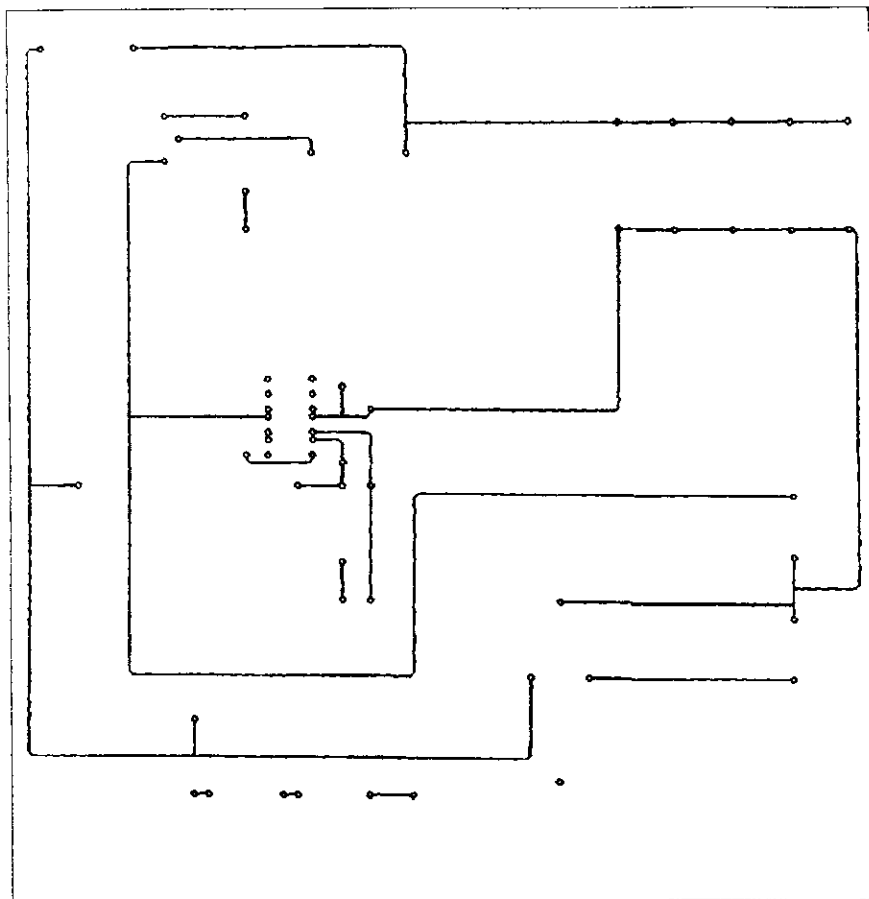
Después de desarrollar las partes A y B de la fuente actual, se desarrolló un sistema UPS para alimentar los PLCs. En ese momento los PLCs estaban formados por microprocesadores PIC los cuales necesitan ser alimentados solamente por 5 V.

El circuito de esta etapa de potencia, incluyendo el UPS DC es el siguiente:



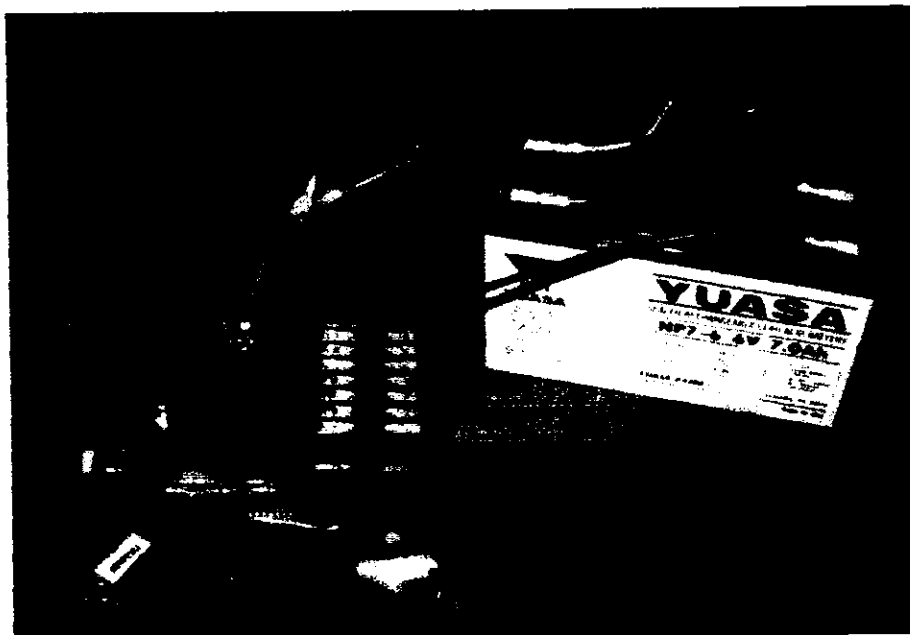
Gráfica 4.19 "Circuito UPS DC"

Este se implementó con el siguiente circuito impreso:



Gráfica 4.20 "Circuito impreso UPS DC"

Resulta el siguiente componente:



Gráfica 4.21 "Unidad UPS DC"

En este UPS DC el regulador de voltaje variable ECG934 produce un voltaje de 14.1 V, y está conectado a la salida con un diodo 1N5401 (5 A).

En todo momento y alimentado por las baterías un OpAmp LM324 revisa cuál es el voltaje que el regulador ECG934 tiene en su salida. Comparando el voltaje de su pata no inversora, con el de la pata inversora.

En la pata no inversora y a través de un divisor de voltaje siempre está presente el 98% del voltaje de la batería. Por el otro lado en su pata no inversora está presente siempre un voltaje de 14.1 V. Cuando falla la alimentación este voltaje cambia a 1.07 V, ya que no hay voltaje de alimentación y sólo queda el divisor de voltaje entre las resistencias de 10 K y 1 K.

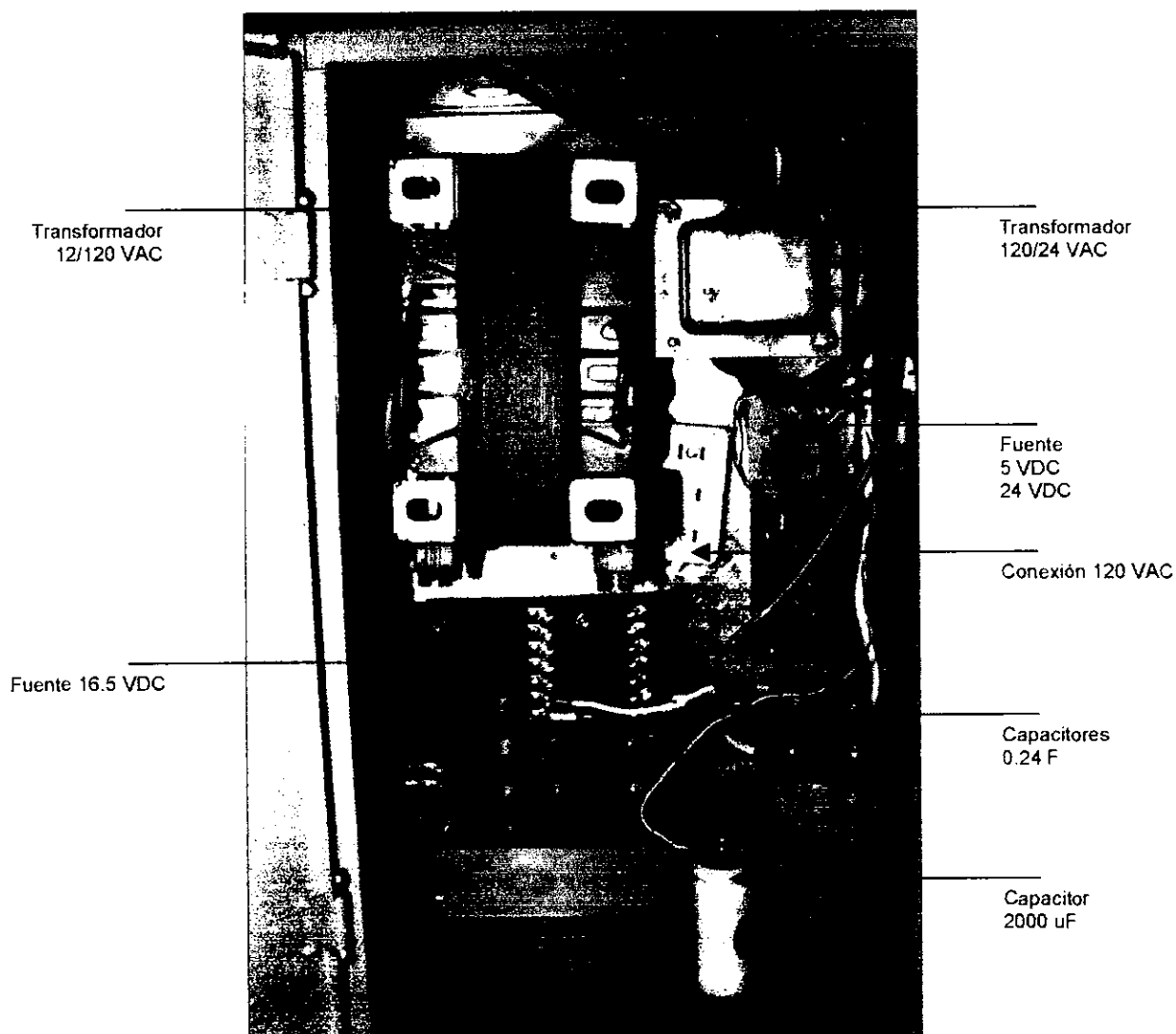
Este cambio en sus entradas, modifica la salida del OpAmp de 24 V a 0 V. Al cambiar este voltaje el transistor darlington TIP 115 conduce y las baterías alimentan el circuito.

Cuando las baterías se descargan, son cargadas a través de la resistencia de 1.5 Ω y los tres diodos 1N4004.

Lamentablemente, cuando se utilizaron los PLC Siemens 214 con alimentación de 120 VAC este UPS no se utilizó y se descartó del proyecto.

B.4. Organización física de la fuente

La fuente está organizada como un elemento físico diferente e independiente en el autómata, que se puede separar del mismo ya que está contenida toda en una caja.



Gráfica 4.22 "Disposición elementos fuente"

C. CIRCUITO PARA EL CONTROL DE LA POTENCIA

La función del circuito para el control de la potencia es permitir al microcontrolador encender los motores en la dirección y con la velocidad determinada por su algoritmo. Para ello debe transformar la salida del microcontrolador, una señal de baja potencia 0.1 W., en una señal de alta potencia 300 W.

Este circuito de potencia, para ser más eficiente, debe operar bajo las siguientes premisas:

- La señal de entrada y la de salida deben ser lo más parecido posible.
- La señal de entrada y la de salida deben ocurrir en forma simultánea.
- La potencia que se consume en el circuito debe ser mínima.

C.1. Especificaciones técnicas del proyecto para la etapa de potencia

Alimentación	Voltaje	15 VDC
	Amperaje	20 A.
Señales de entrada	Número	Dos
	Voltaje	5 V.
	Amperaje	10 mA.
	Frecuencia	En el rango de los 500 Hz., la frecuencia específica de cada etapa está determinada por las características del motor.
	Modulación	Modulación de ancho del pulso (PWM)
Señales de salida	Número	Uno.
	Voltaje	Variable entre -16.5 y 16.5 V.
	Amperaje	20 A. Máximo
	Frecuencia	En el rango de los 500 Hz., la frecuencia específica de cada etapa está determinada por las características del motor.
	Modulación	Modulación de ancho del pulso (PWM)

Tabla 4.2 "Especificaciones"

C.2. Operación de la etapa de potencia

El circuito de potencia recibe las señales Entrada 1 y Entrada 2, y dependiendo de sus valores puede dar las siguientes respuestas:











Entrada 1	Entrada 2	Salida
0	0	0
1	0	1
0	1	-1
1	1	NA

Tabla 4.3 "Respuestas de la tarjeta"

Es decir, si no está presente ninguna de las dos entradas el motor se encuentra apagado, si está encendida una entrada el motor gira en una dirección y si está encendida la otra, entonces gira en la dirección contraria. Finalmente si ambas entradas están prendidas el circuito no tiene respuesta, este caso específico se debe evitar porque produce la destrucción de la etapa de potencia, como se detalla más adelante.

La velocidad también se controla con el circuito de potencia. Para ello en la señal de salida está presente la misma información de modulación del ancho del pulso que tiene la señal de entrada.

La modulación del ancho del pulso funciona modificando la potencia de una señal mediante la variación de su ciclo de trabajo, es decir, al modificar el tiempo en que una señal está prendida o apagada:

Forma de la señal	Potencia
On  Off 	0%
On  Off 	25%
On  Off 	50%
On  Off 	75%
On  Off 	100%

Gráfica 4.25 "Relación potencia/ancho de pulso"

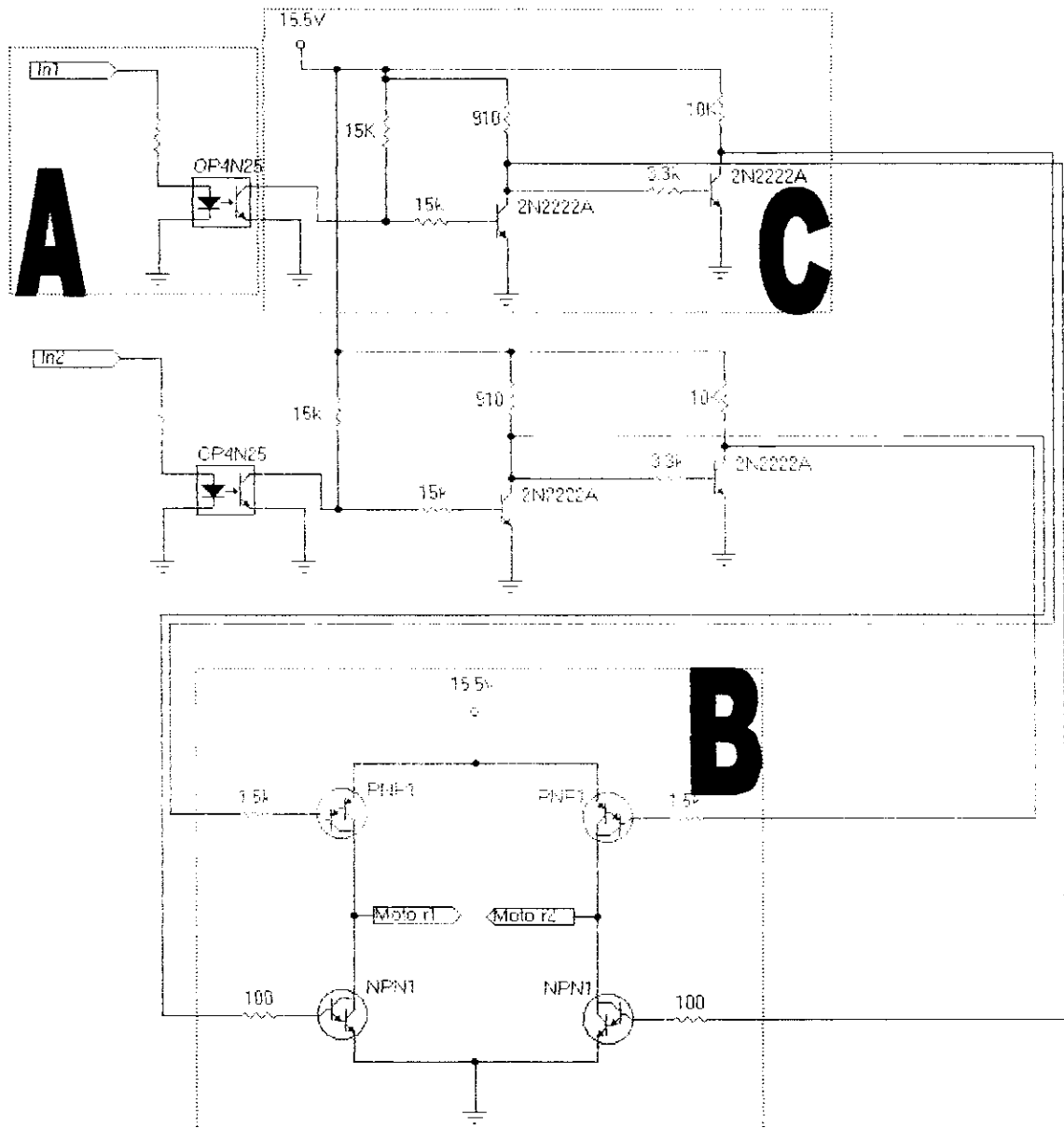
Una característica de los motores que se utilizan en este proyecto, motores DC de imán permanente con escobillas, es que la velocidad es proporcional a la potencia que se aplica, por lo tanto se puede controlar la velocidad utilizando una señal modulada en el ancho del pulso.

Aunque la relación no es lineal y depende de las características eléctricas y mecánicas del motor, para los fines de este proyecto se supondrá que la relación entre potencia y velocidad es lineal.

C.3 Circuito de la etapa de potencia

El circuito de la etapa de potencia se puede dividir en tres partes diferentes.

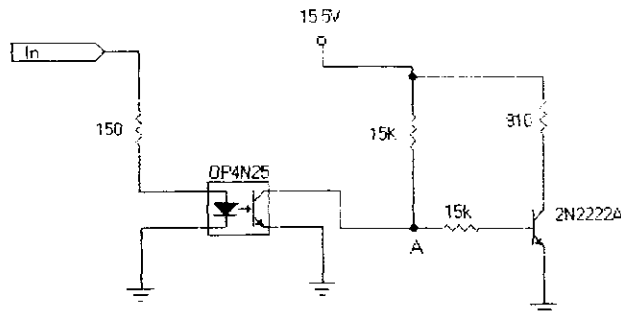
- A. Aislamiento eléctrico.
- B. Etapa de potencia
- C. Etapa de control



Gráfica 4.26 "Circuito de Potencia"

El funcionamiento de cada una de las partes es el siguiente:

A. Aislamiento eléctrico:

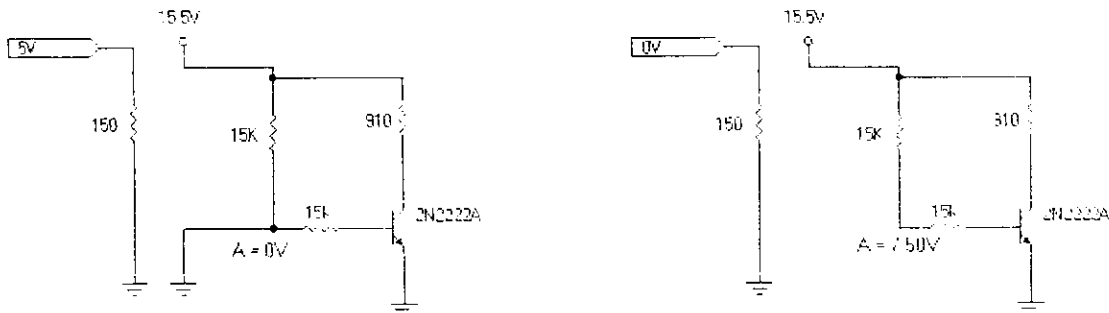


Gráfica 4.27 "Aislamiento eléctrico"

La función del aislamiento eléctrico es evitar que la interferencia que producen los motores en las líneas de alimentación afecten a los dispositivos de control. Esto es especialmente crítico cuando se están controlando los motores a través de una señal PWM y el ruido generado en los motores puede modificar el ancho del pulso.

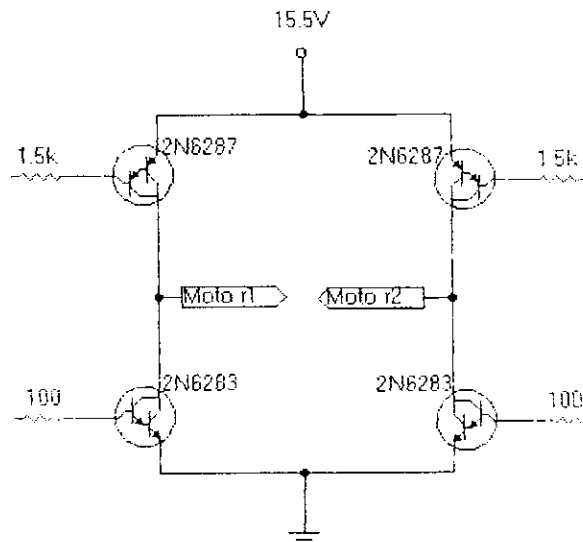
El dispositivo de aislamiento eléctrico consiste en un fotodiodo y un fototransistor acoplados en la misma unidad, OP4N25 ó ECG3086. Cuando circula corriente en el diodo del dispositivo, éste activa el transistor mediante la emisión de fotones, los cuales operan en el transistor como normalmente lo hace la corriente de base.

Para activar el diodo se utilizan 5 V y una resistencia de 150Ω . Fluyen 30 mA al mismo. El transistor del OP4N25 está en una configuración de inversor, por lo que el voltaje en su colector, nodo A, siempre está invertido respecto a la señal de entrada "IN". Es decir cuando en la entrada están presentes 5V, el transistor del OPN25 se satura y prácticamente su colector conduce a tierra (0.2 V). Cuando no se aplica más de 0.7 V en la entrada entonces el voltaje en el nodo A estará determinado por el divisor de voltaje de las dos resistencias de $15K\Omega$ y será de 7.5V.



Gráfica 4.28 "Función ECG3086"

B. Etapa de potencia:

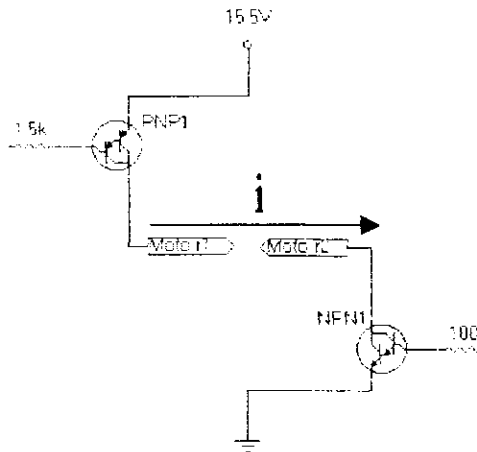


Gráfica 4.29 "Etapa de potencia"

La función de la etapa de potencia es conectar y aislar las terminales del motor, de las terminales de la fuente de voltaje. Para que el motor esté apagado, gira en una dirección o en la dirección contraria. La etapa de potencia está constituida por cuatro transistores Darlington de potencia, 2 transistores NPN: 2N6283 ó ECG251 y 2 transistores PNP: 2N6287 ó ECG252.

En el caso de que el motor deba estar apagado, el voltaje aplicado a las bases de los transistores PNP es V_{cc} (15.5 V) y a la base de los NPN es tierra (0.2 V). De esta forma no existe corriente en ninguna de las bases. Al no fluir corriente por ninguno de los transistores el motor está aislado y en sus terminales no existe diferencia de voltaje.

Cuando el motor se tiene que encender en una dirección, se activan dos transistores:

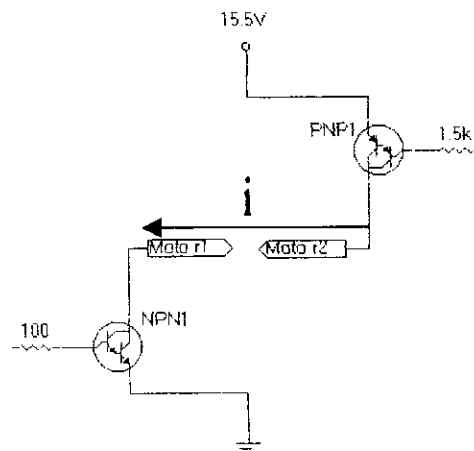


Gráfica 4.30 "Etapa de potencia encendida"

Estos dos transistores son llevados a saturación al aplicar por lo menos 5 mA de corriente a la base. La saturación se logra gracias a que tienen una ganancia de corriente (h_{FE}) mínima de 2,400. La corriente de saturación se obtiene de aplicar 0.2 V. a través de una resistencia de 1.5 K Ω en la base del transistor PNP, y al aplicar 4 V. con una resistencia de 100 Ω en el NPN.

Al estar saturados los transistores pueden hacer fluir una corriente máxima en el emisor de 20 Amperios. El valor de la misma queda determinado por el motor y la carga que se le aplique.

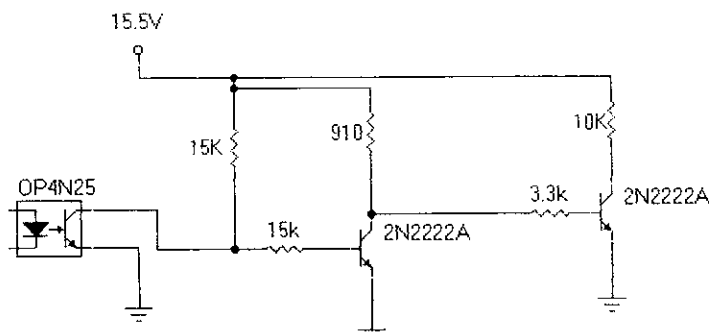
Si el motor debe girar en la dirección contraria, entonces se prenden los otros dos transistores. El cambio en la dirección de la corriente implica un cambio en la polaridad aplicada a las terminales. El cual se logra saturando los otros dos transistores, nuevamente con corrientes de base superiores a 5 mA.



Gráfica 4.31 "Etapa de potencia encendida (dirección inversa)"

El funcionamiento de la etapa de potencia permite variar el voltaje aplicado a los motores entre -13.2 V y 13.2 V con solamente una fuente de 15.5 V .

C. Etapa de control:



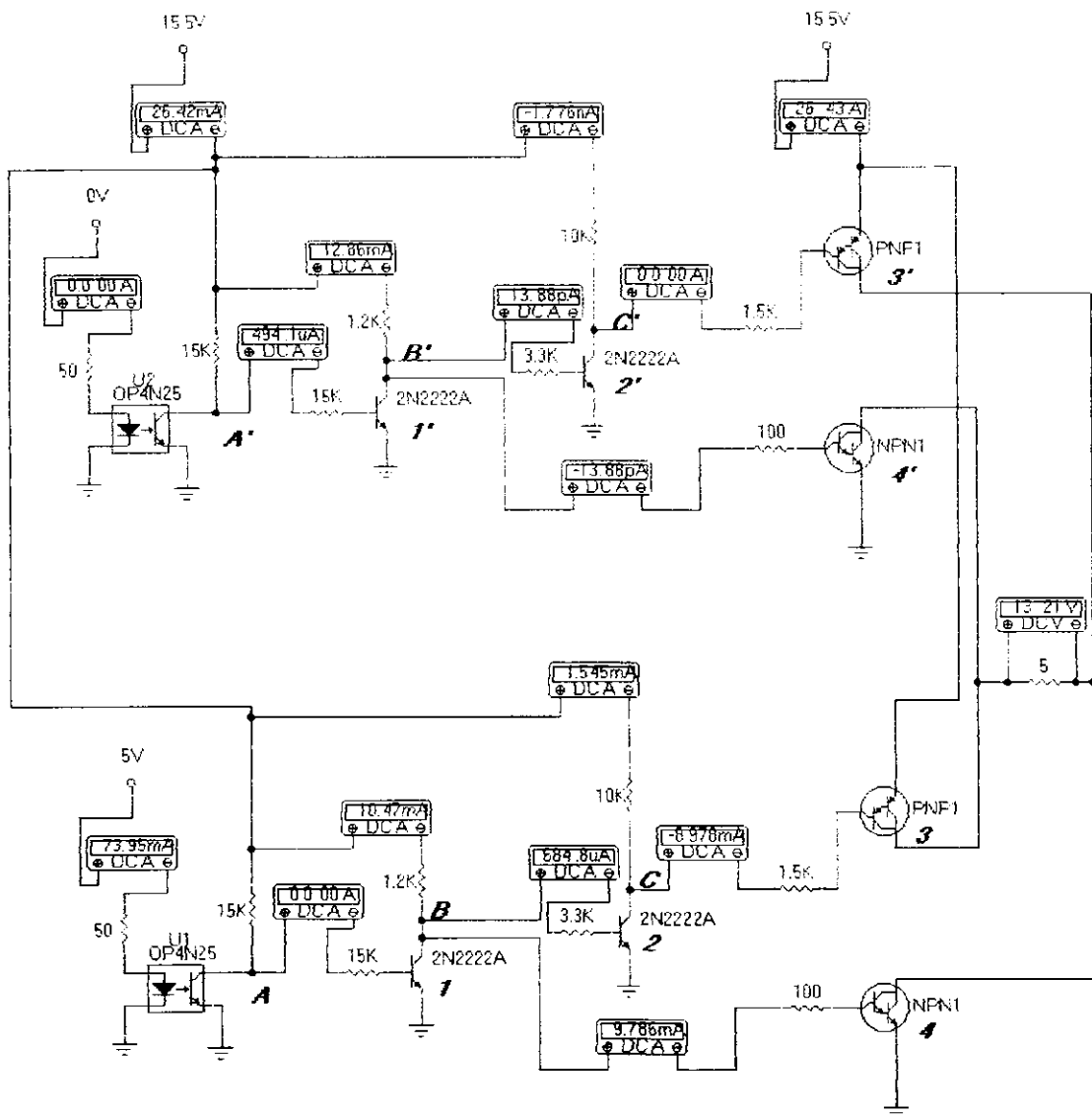
Gráfica 4.32 "Etapa control"

La función de la etapa de control es proveer las señales necesarias para el control de los transistores de potencia que se encuentran en la Parte B del circuito. Estas señales son 15.5 V (OFF) y 0.2 V (ON) en el caso de los transistores PNP; y 0.2 V (OFF) y 4 V (ON) para los transistores NPN.

El funcionamiento de la etapa de control es independiente para cada dirección, en ambos casos funciona igual y ambas tienen un circuito idéntico.

El funcionamiento es sencillo y únicamente está constituido por dos transistores que operan como inversores, el objetivo es proveer con al menos 5 mA a cada transistor de potencia. En la operación real, los transistores PNP reciben 8.99 mA y lo NPN 9.79 mA.

La corriente que recibe cada transistor y las que están presentes en cada resistencia, cuando el circuito de potencia está encendido en una dirección, son:



Gráfica 4.33 "Funcionamiento de la etapa de control"

En la gráfica 4.33 se aprecia cómo está activada la dirección que se encuentra en la parte de abajo del circuito. Esto se logra aplicando 5 V en una resistencia de 50Ω que alimenta el fotodiodo. Los 74 mA presentes en el fotodiodo saturan el fototransistor, y por lo tanto en el colector del mismo, nodo A, están presentes 0.14 V. Como en el nodo A no existe voltaje suficiente para polarizar la base del transistor 1, no fluye corriente por la resistencia de $15 \text{ K}\Omega$ y el transistor 1 está apagado. Al estar apagado el transistor, no interviene en el voltaje del nodo B, el cual queda determinado por el divisor de voltaje que constituyen las resistencias de $1.2 \text{ K}\Omega$, $3.3 \text{ K}\Omega$ y 100Ω . Este valor medido en el nodo B es de 2.9 V.

Este voltaje aplicado al transistor 4 genera 9.78 mA suficiente corriente para saturar el transistor 4. En el transistor 2 este voltaje genera la saturación, con 0.7 mA en la base. Al estar saturado este transistor, su colector está a 0.05 V (nodo C), lo que genera una corriente en el transistor 3 de 8.99 mA.

Al estar saturados los transistores 3 y 4, las terminales del motor quedan conectadas a 15.5 V y a tierra. Se perderán 2.3 V en ambos transistores Darlington.

La operación del circuito de la otra dirección es inversa. El fototransistor está apagado y no interviene en el voltaje del nodo A', cuyos 8.1 V, quedan determinados por el divisor de voltaje de las dos resistencias de $15 \text{ K}\Omega$. De esta forma, la base del transistor 1' queda alimentada directamente de la fuente a través de una resistencia de $30 \text{ K}\Omega$, saturando al transistor.

Al estar el transistor 1 saturado, en su colector, nodo B', se miden 0.07 V. La presencia de un voltaje tan bajo impide polarizar la base de los transistores 2' y 4'. Al no estar polarizado el transistor 4', no fluye corriente entre su base y su colector. Y la terminal del motor está aislada de tierra.

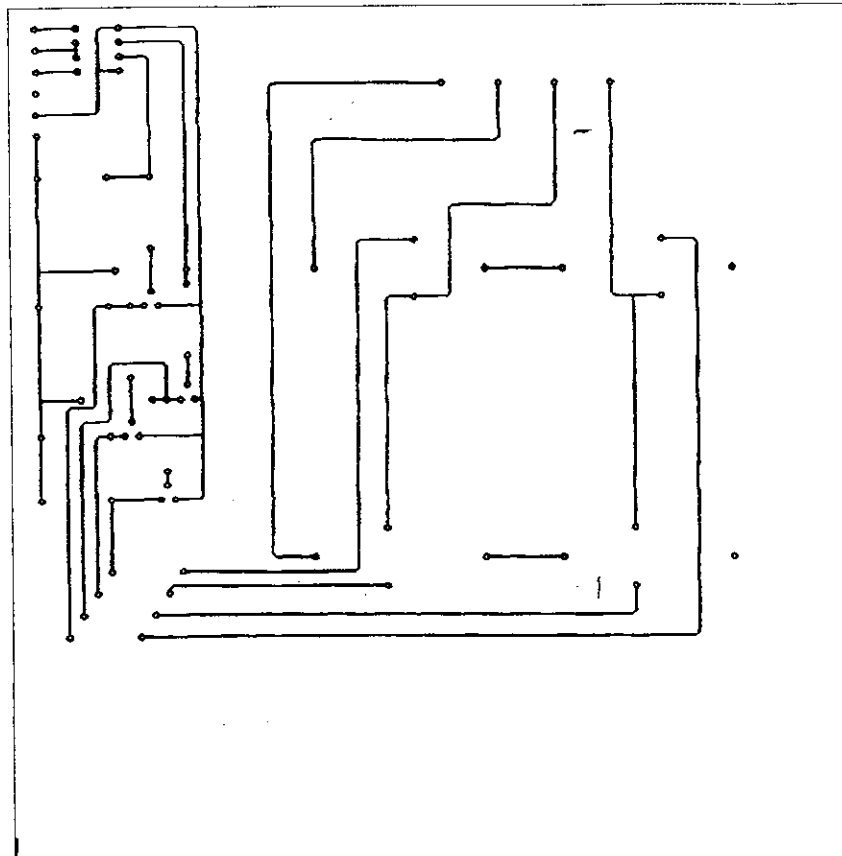
Al mismo tiempo, como el transistor 2' no está polarizado entonces el voltaje en el nodo C' es 15.5 V. La presencia de este voltaje impide polarizar la base del transistor 3'. Al no estar polarizado el transistor, no fluye corriente entre su base y su colector. Y la terminal del motor está aislada de 15.5 V.

En el caso contrario, cuando se quiere encender la otra dirección, se aplican 5V en el otro fotodiodo, la operación será igual a la que se describió anteriormente. Para realizar el análisis, se deberán intercambiar los nodos y transistores prima con los no prima.

C.4. Circuito Impreso

El circuito impreso nuevamente se realizó con el programa PCROUTE, pero se debieron realizar algunas modificaciones sobre el resultado, porque el dibujo generado era tan complicado, que prácticamente no se podía fabricar a mano la tarjeta impresa. También, el programa no considera las corrientes que circulan por el circuito y las líneas no tenían el ancho necesario para 20 A.

Por lo tanto se utilizó la opción del programa que permite definir algunas rutas que no pueden ser variadas. Estas rutas fueron aquellas con corrientes de 20 A. A partir de ese punto se optimizó el trazado restante y la solución fue la siguiente:



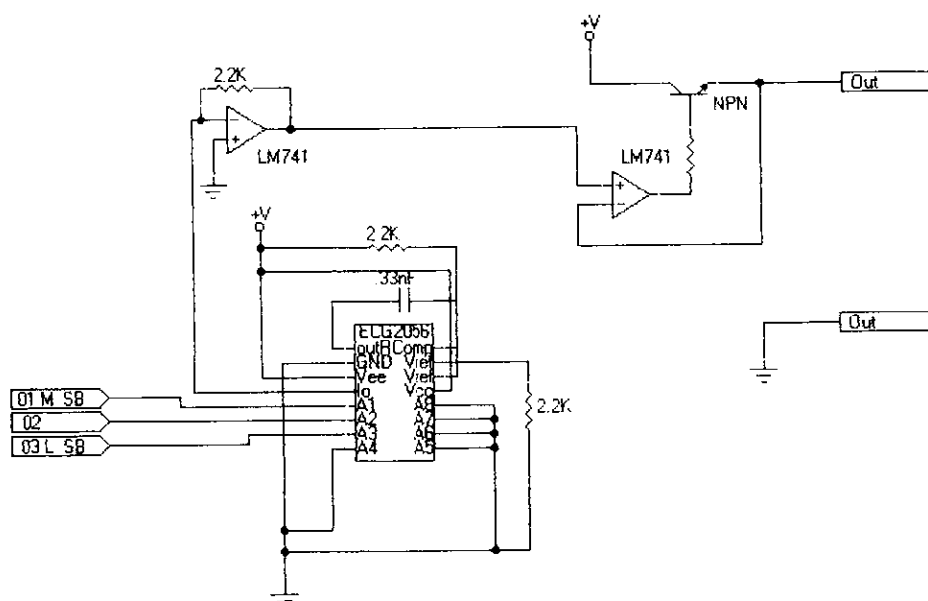
Gráfica 4.34 "Circuito Impreso"

C.5. Desarrollo del Circuito de Potencia

Para desarrollar el circuito de potencia se trabajaron varios modelos, los cuales se fueron descartando por problemas de implementación o por consideraciones de rendimiento. Estos circuitos fueron:

A. Etapa de potencia DC con convertidor D/A:

Originalmente se quería determinar la velocidad del motor a través de una señal de mando que tuviera 8 o 16 niveles diferentes. Para ello el microprocesador generaba una señal con 3 ó 4 bits. Estos bits eran convertidos en la etapa de potencia de digital a analógico y se obtenían las diferentes velocidades. El circuito es el siguiente:



Gráfica 4.35 "Etapa de potencia no utilizada: Etapa de potencia DC con convertidor D/A"

Esta solución presentó los siguientes inconvenientes:

- Se contaba con muy pocas velocidades, solamente 8 o 16 para ambas direcciones. Es decir, 4 u 8 velocidades en cada dirección.

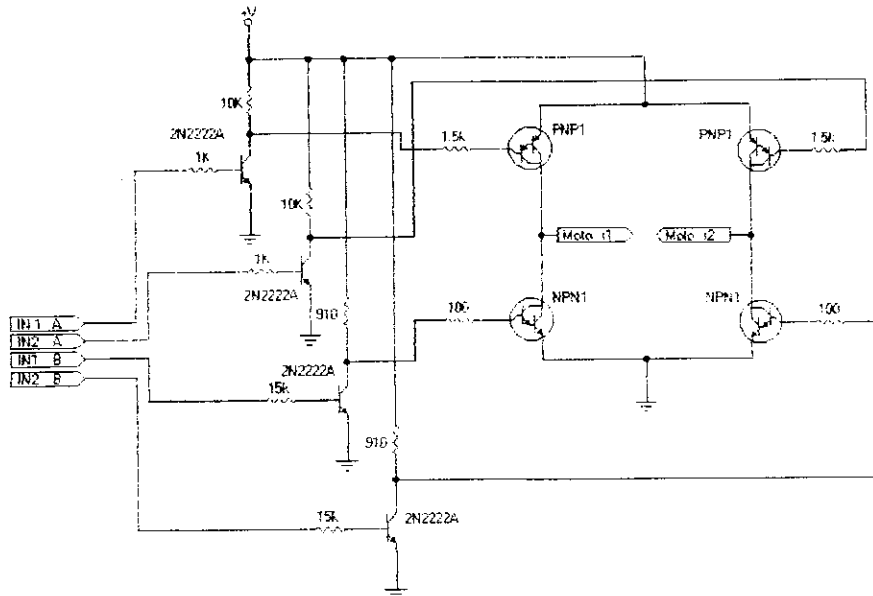
El número de velocidades puede aumentar conforme se utilicen más bits del convertidor digital/análogo hasta un máximo de 156 direcciones en cada dirección, cuando se utilizan 8 bits. Pero limita la utilización de bits de salida del microprocesador y aumenta significativamente el número de alambres que deben estar presentes en la interface entre el microprocesador y el circuito de potencia.

- La necesidad de utilizar fuentes de alimentación $+15.5\text{ V}$ y -15.5 V . Aunque es factible el manejo de $+15.5\text{ V}$ y -15.5 V involucra un cambio en la alimentación del autómata, específicamente se necesitarían 3 rieles. Adicionalmente, se debe colocar un circuito espejo que funcione para los voltajes negativos.

El mayor problema asociado a este circuito es el aumento del número de elementos que acarrea, ya que hay que usar: 2 convertidores D/A, 4 OpAmp, 2 Transistores de potencia y 3 rieles.

Si no se utiliza un circuito gemelo para proveer los voltajes negativos entonces se puede sustituir tierra con -15.5 V . Esta sustitución parece rápida y factible, el problema que conlleva es la necesidad de graduar con mucha exactitud el voltaje de 0 V , el cual estaría representado por 1000 ó 0111 si se trabaja con 4 bits. Cualquier cambio en los voltaje de alimentación puede modificar la posición del cero y podría iniciar el movimiento del motor cuando el microcontralador se supone que está apagado.

Finalmente, el problema más importante de esta implementación está relacionada con la potencia consumida. En todo momento y no importando cuál es el voltaje aplicado en el motor, siempre se está consumiendo toda la potencia, 300W. La única diferencia es que la potencia no se convierte en trabajo mecánico sino que se disipa como calor en el transistor de potencia.



Gráfica 4.37 "Etapa de potencia no utilizada: Etapa de potencia PWM sin circuito de control"

Este circuito soluciona los problemas de resolución y de potencia. Sin embargo necesita 4 señales PWM para funcionar. Estas señales deben estar sincronizadas por pares, para activar correctamente los transistores.

Aunque muchos microcontroladores tienen salidas PWM es muy difícil encontrar alguno que tenga 4 salidas de este tipo, por lo tanto la solución es generar dos de las señales invirtiendo las otras dos.

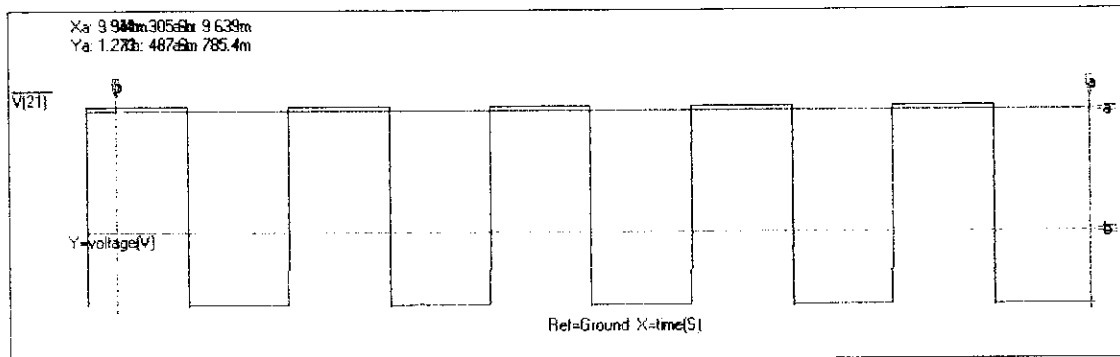
C.5. Resultados en el simulador

Todo el desarrollo de la etapa de potencia se hizo simulando las características de cada uno de los elementos en el programa Circuit Maker. En este programa se probaron originalmente los circuitos anteriormente descritos.

A continuación figura el análisis del circuito finalmente utilizado. Comparándolo contra cada una de las premisas que definieron el diseño del mismo:

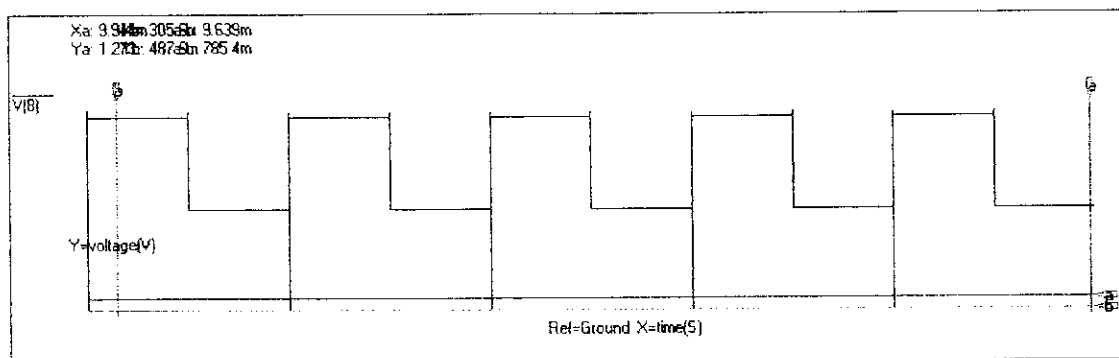
- La señal de entrada y la de salida deben ser lo más parecidas posible.

Señal de entrada:

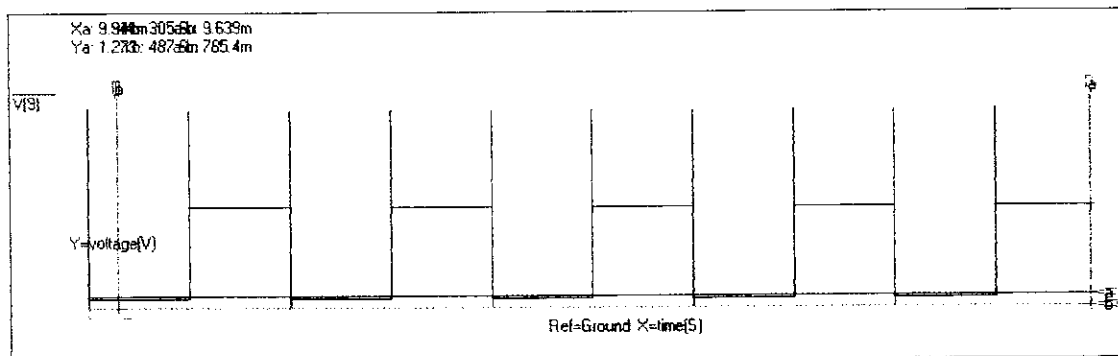


Gráfica 4.38 "Simulador: señal de entrada"

Señales de salida:



Gráfica 4.39 "Simulador: terminal uno"



Gráfica 4.40 "Simulador: terminal dos"

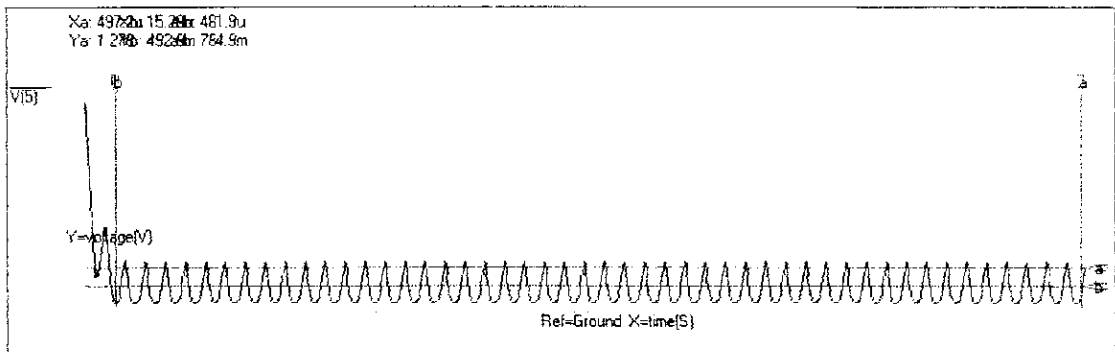
En esta simulación, la salida se obtiene de restar las dos señales de entrada, la forma de las señales son muy parecidas. La diferencia la constituyen los picos que están presentes en cada uno de los bordes de la señal, los cuales son debidos a los retrasos que se presentan en los transistores del circuito de control.

- La señal de entrada y la de salida deben ocurrir en forma simultánea.

En las dos gráficas anteriores se puede apreciar cómo los cambios suceden prácticamente de forma instantánea, los retrasos son debido a los transistores que están en cascada y al aislador óptico.

Además, es el aislador óptico la limitante para trabajar estas tarjetas en frecuencias más altas. No puede operar a más de 5 KHz. Esto se debe a que el transistor es activado y desactivado por una emisión de fotones, los cuales se producen en función de la cantidad de corriente que pasa por el fotodiodo, con un atraso en la emisión respecto a la señal aplicada

Este atraso degenera la señal conforme aumenta la frecuencia:



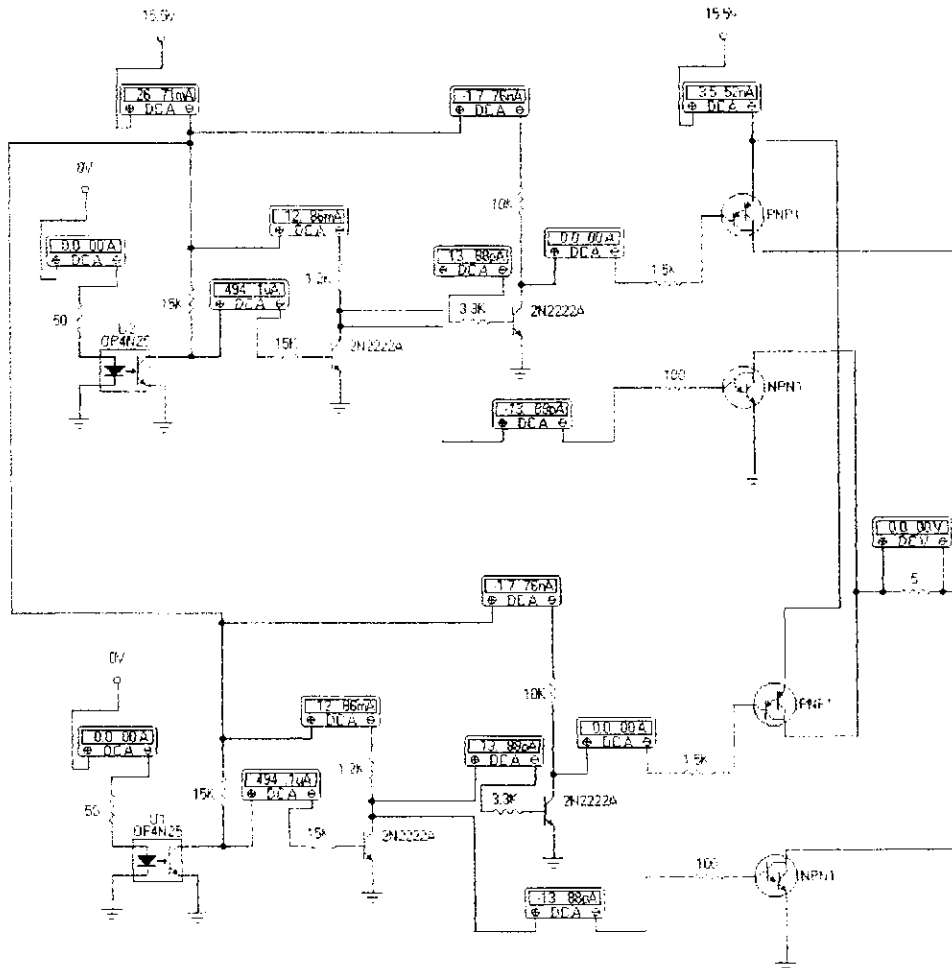
Gráfica 4.41 "Simulador: voltaje en el colector del ECG3086"

Lo que se observa en la Gráfica 4.41 es como la señal cuadrada que alimenta al fotodiodo del ECG3086 prácticamente activa permanentemente la emisión de fotones en el diodo y este flujo constante de fotones mantiene saturado al fototransistor, por lo tanto el voltaje en el colector siempre será 0.14V.

- La potencia que se consume en el circuito de potencia debe ser mínima.

Finalmente para analizar el consumo de potencia, se simulan los casos en que el motor está encendido o apagado.

Motor apagado:



Gráfica 4.42 "Funcionamiento de la etapa de potencia: motor apagado"

Para este caso la etapa consume 0.41 W ($5.5 \text{ V} \times 26.71 \text{ mA}$). Lo que representa una eficiencia del 0% ya que toda la potencia consumida no se aplica en el motor.

Motor Encendido:

El caso contrario, donde la etapa de potencia alcanza su mayor eficiencia es cuando el motor está encendido y con una velocidad máxima (PWM = 100%). Este caso lo podemos observar en la Gráfica 4.33.

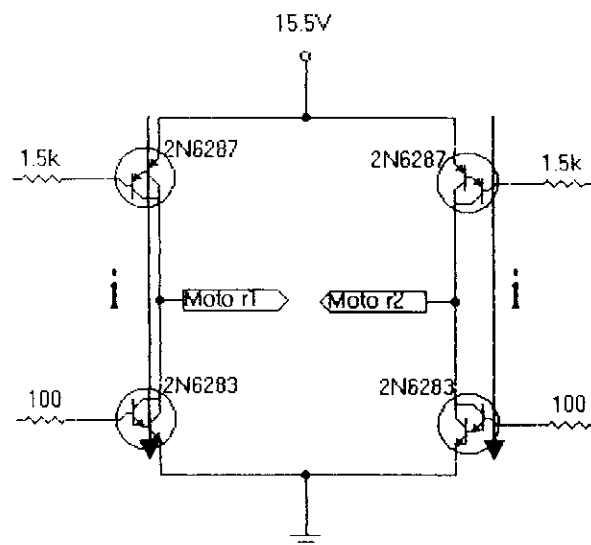
En ella se observa cómo la etapa de potencia está recibiendo 414 W ($15.5 \text{ V} \times 26.72 \text{ A}$) y está entregando en el motor 349W ($13.21 \text{ V} \times 26.43 \text{ A}$). Esto representa una eficiencia del 85%.

C.6. Inconvenientes prácticos

En su operación las etapas de potencia han respondido en función de todo el análisis antes descrito, aunque nunca se ha operado ninguna hasta el límite máximo.

Sin embargo en la operación práctica si ha surgido un inconveniente, generado por la configuración de los 4 transistores de potencia, la cual permite que éstos se quemen si conducen los 4 juntos o por parejas verticales. Esto sucede cuando las señales de control no son las adecuadas o se solicita ir en ambas direcciones.

Este inconveniente estaba previsto en el análisis y contemplado en el diseño de las señales de control.



Gráfica 4.43 "Operación destructiva de la etapa de potencia"

Lo que no estaba previsto y destruyó varias etapas de control es que durante el tiempo en que se enciende la etapa de control, las milésimas de segundo, que tardan los transistores de control en alcanzar su voltaje de operación normal (estando apagadas las señales de control provenientes del PLC), los transistores de potencia no reciben las señales de control adecuadas. Y durante un momento se saturan y conducen, quemándose tan rápidamente que cuando los transistores de control alcanzan su voltaje de operación, la etapa ya no sirve.

La solución a este problema es alimentar separadamente la etapa de control y los transistores de potencia. Conectando primero la etapa de control y después los transistores de potencia. Los detalles de cómo se implementó esta solución con un relay se encuentran en el capítulo 4 sección F.

D. MOTORES

En el autómatas se utilizan tres motores, uno para cada eje. De los tres motores dos han sido reciclados y un tercero fue comprado específicamente para la aplicación.

Los tres motores son DC de imán permanente, cuya velocidad es proporcional al voltaje que se les aplica en las terminales. Es proporcionalidad prácticamente lineal, pero se ve afectada por la resistencia del embobinado, el aislamiento del alambre en el embobinado, el flujo electromagnético del imán, la resistencia entre las escobillas y el rotor, y otros aspectos mecánicos y electromagnéticos.

Desde el punto de vista práctico y para efectos de esta aplicación la relación proporcional se supone lineal. Las características de cada uno de los motores son.

D.1 Motor de desplazamiento horizontal

El motor y su motorreductor están contruidos de una aleación de zinc. Ambos mecanismos están empacados en una sola unidad

El motorreductor esta compuesto por un eje helicoidal y sucesivos pasos de engranajes dentados, los cuales están fabricados de hierro galvanizado y hierro colado respectivamente. Las salidas del motorreductor están compuestas por anillos-O para evitar el goteo del aceite. El motorreductor está planificado para una operación continua (10 horas/día), y un factor de mantenimiento de 1.0.

El motor es un motor de 12 VDC de imán permanente, con conexión de bolas, puede operar a un máximo de 40°C, y las escobillas se pueden reemplazar externamente.

Sus especificaciones son las siguientes:

Modelo	1L474
F/L Rpm	6
F/L Torque In-Lbs	500
Input Hp	1/15
F/L Amps	8.5
Gear Ratio	267:1
Length Overall	10 1/32

Tabla 4.4 "Especificaciones motor horizontal"

C.2 Motor de desplazamiento vertical

El motor de desplazamiento vertical fue reciclado de una unidad de cinta magnética, las características de su construcción y su funcionamiento no se conocen con exactitud. Solamente están al alcance las especificaciones técnicas que aparecen en su placa:

Voltaje	24 VDC
Amperaje	0.5 A.
Velocidad	375 RPM
Operación	Continua

Tabla 4.5 "Especificaciones motor vertical"

Este motor no tiene la fuerza necesaria para elevar mucha carga. Por ello se construyó un motorreductor que le permitiera levantar por los menos 100 lbs. Actualmente la capacidad de carga del eje vertical son 50 lbs, porque la estructura del tenedor pesa 50 lbs.

Este motoreductor tiene una reducción de 300:1 así como un eje helicoidal y un segundo paso de engranaje dentado. Esta configuración permite que el desplazamiento en el eje se realice únicamente cuando el motor está funcionando. No importa cuál es la carga aplicada sobre el motorreductor, la configuración helicoidal/dentada evita su desplazamiento.

C.3. Motor del giro

Este motor fue reciclado del limpiabrisas de un automóvil, a diferencia de los otros motores, de éste no se tiene especificación alguna, ya que no cuenta ni con una placa de especificaciones. Su utilización se realizó sobre los resultados prácticos obtenidos en experimentos.

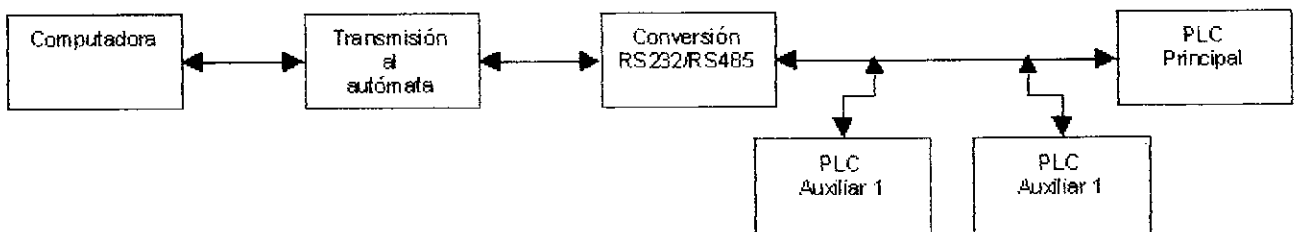
Este motor cuenta con un motorreductor y puede funcionar a dos velocidades. De éstas se emplean la más lenta. También cuenta con una serie de embobinados secundarios, por lo menos 2 embobinados más, los cuales nunca se logró que funcionaran.

E. SISTEMA DE COMUNICACION

La función del sistema de comunicación es mantener un canal abierto entre el autómata y un operario externo.

Las funciones específicas de este enlace dependen del operario y la programación realizada. Su función principal es transmitir la orden que el autómata debe realizar y recibir el mensaje cuando éste responda que ya la realizó. También se puede utilizar para transmitir parámetros, velocidades, órdenes intermedias y otras funciones.

Las partes que componen el sistema de comunicación son:



Gráfica 4.44 "Elementos del sistema de comunicación"

Todo el sistema de comunicación se fundamenta en la transmisión y la recepción de un grupo de 10 bytes que contiene la siguiente información:

# Byte	Descripción	Valor Nominal
1	Bandera Inicio	128
2	Dirección Destino	
3	Dirección Origen	
4	Comando 1	
5	Comando 2	
6	Dato 1	
7	Dato 2	
8	Dato 3	
9	Dato 4	
10	Bandera Final	128

Tabla 4.6 "Protocolo de comunicación"

Estos se transmiten a través de todo el sistema, a una velocidad de 9,600 baudios; con 8 bit por byte; un bit de inicio y uno de final; y sin bit de paridad.

Las características de cada uno de los elementos de este sistema son:

E.1. Computadora

Se está utilizando una computadora IBM Personal System/2 Modelo 55 SX, cuyas características técnicas son las siguientes:

Microprocesador	Intel 386 SX
Memoria RAM	20 Mbytes
Unidades de Disco	Duro: 30 Mbytes Flexible: 1.44 Mbytes
Puertos de Comunicación	Paralelo: LPT1 Serial: COM1

Tabla 4.7 "Especificaciones computadora"

En esta computadora la comunicación se realiza a través del puerto Serial COM1. Mediante un programa desarrollado específicamente para esta aplicación, llamado Puerto01.

E.2. Programa de Comunicación

Se desarrolló el programa Puerto01 en Visual Basic para ambiente Windows. Las principales rutinas para el funcionamiento del mismo son:

Inicialización:

```
Sub Inicio_puerto()
  MSComm1.CommPort = puerto
  ' 9600 baud, no parity, 8 data, and 1 stop bit.
  MSComm1.Settings = "9600,N,8,1"
  ' Tell the control to read entire buffer when Input is used.
  MSComm1.InputLen = 1
  ' Open the port.
  MSComm1.PortOpen = True
End Sub
```

En esta rutina se definen los parámetros de la comunicación: Puerto Comm1; Velocidad 9600, Bit de Inicio; Bit de final; Sin paridad; y 8 bits por Byte.

Transmisión:

```
Private Sub Command1_Click()
  If Not comx Then
    Option3.Value = True
  End If

  If Option1 Then
    MSComm1.Output = Chr(CByte(Text1))
    MSComm1.Output = Chr(CByte(Text2))
  End If
End Sub
```

```

MSComm1.Output = Chr(CByte(Text3))
MSComm1.Output = Chr(CByte(Text4))
MSComm1.Output = Chr(CByte(Text5))
MSComm1.Output = Chr(CByte(Text6))
MSComm1.Output = Chr(CByte(Text7))
MSComm1.Output = Chr(CByte(Text8))
MSComm1.Output = Chr(CByte(Text9))
MSComm1.Output = Chr(CByte(Text10))
Else
MSComm1.Output = Chr(CByte("&H" + Text1))
MSComm1.Output = Chr(CByte("&H" + Text2))
MSComm1.Output = Chr(CByte("&H" + Text3))
MSComm1.Output = Chr(CByte("&H" + Text4))
MSComm1.Output = Chr(CByte("&H" + Text5))
MSComm1.Output = Chr(CByte("&H" + Text6))
MSComm1.Output = Chr(CByte("&H" + Text7))
MSComm1.Output = Chr(CByte("&H" + Text8))
MSComm1.Output = Chr(CByte("&H" + Text9))
MSComm1.Output = Chr(CByte("&H" + Text10))
End If
End Sub

```

En esta rutina, primero se revisa si el puerto ha sido activado o no, para luego transmitir la información dependiendo del formato en el cual están ingresados los caracteres, decimal o hexadecimal.

Recepción:

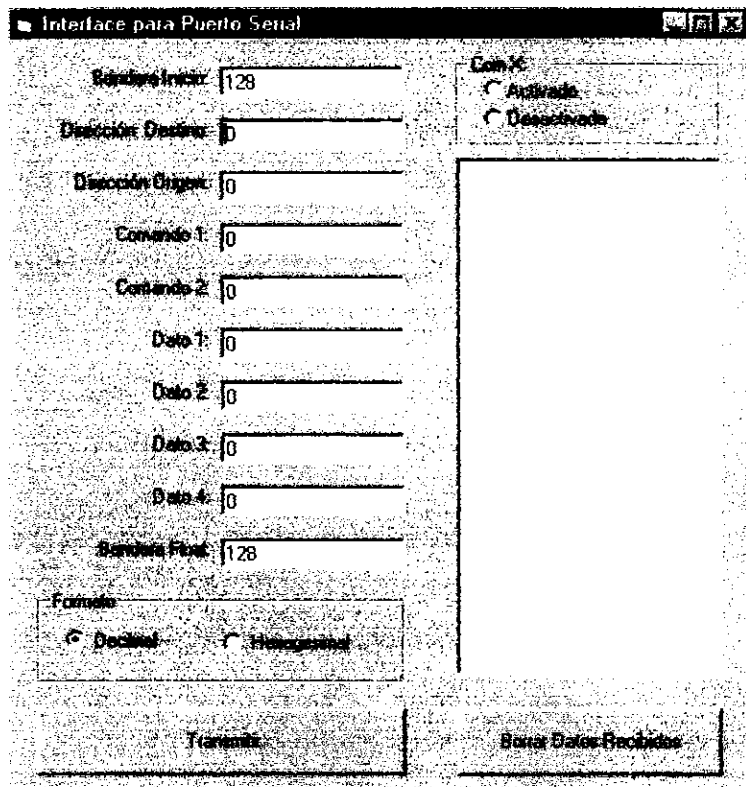
```

If MSComm1.InBufferCount Then
    MSComm1.InputLen = 1
    List1.AddItem CStr(CByte(Asc(MSComm1.Input)))
End If

```

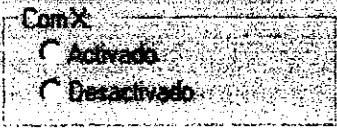
En esta rutina, se analiza si hay datos recibidos y estos datos son colocados en la ventana donde se acumulan los datos recibidos.

La operación de este programa es sencilla, ya está diseñada para un ambiente Windows. El usuario únicamente debe familiarizarse con los elementos del programa:



Gráfica 4.45 "Ventana principal programa Puerto01"

Las funciones son:

	<p>Activado: Al presionar sobre el botón «activado», el puerto de comunicación es seleccionado para el uso exclusivo de este programa. Al mismo tiempo aparece junto a "ComX:" el número del puerto seleccionado (en caso de que la computadora tenga varios).</p> <p>Desactivado: Al presionar sobre este botón el puerto es desactivado y cualquier otro programa puede utilizarlo.</p>
---	---


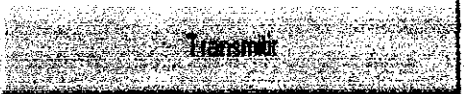

	<p>Decimal: Al seleccionar el botón «Decimal» se le indica al programa que todos los datos ingresados tienen <i>formato decimal</i>. El formato decimal es la selección natural. Si se selecciona decimal y antes estaba seleccionado Hexadecimal, todos los datos presentes serán convertidos.</p> <p>Hexadecimal: Al seleccionar el botón «Hexadecimal» se le indica al programa que todos los datos ingresados tienen formato Hexadecimal. Al seleccionar este botón todos los datos presentes serán convertidos.</p>
	<p>Al presionar el botón de «Transmitir» los 10 bytes que han sido colocados en las casillas de banderas, comandos y datos son enviados por el puerto de comunicación. Si se presiona el botón «Transmitir» antes de activar la transmisión, este paso se realizará automáticamente.</p>
	<p>Todos los datos recibidos por el puerto se van almacenando en la ventana blanca que está al lado derecho de la pantalla. No importa la dirección de origen y destino. Una vez los datos sobrepasan el tamaño de la ventana, a ésta le aparece una barra de desplazamiento en el lado derecho, para que todos los datos siempre se puedan revisar. Si se desea borrar los datos recibidos y dejar la ventana nuevamente en blanco se deberá presionar el botón «Borrar Datos Recibidos».</p>

Tabla 4.8 "Elementos del programa"

E.3. Transmisión al autómata

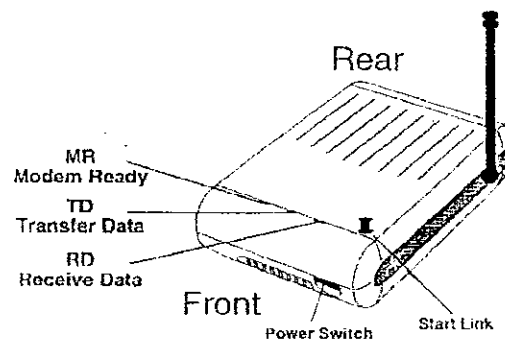
La transmisión de información tiene que ser continua no importando la posición o el movimiento del autómata.

Para lograr una comunicación permanente se utilizan dos Wireless radio modems 3JTECH, los cuales funcionan como si existiera un cable serial físico entre la computadora y el autómata. Las especificaciones técnicas de los radio modems son:

Dimensión:	105 x 70 x 25 mm.
Ambiente de Operación: Rango de temperatura Rango de humedad	0 – 50 °C 10 – 90 %
Ambiente de Almacenamiento: Rango de temperatura Rango de humedad	-20 – 70 °C 5 – 95 %
Requerimiento de Potencia:	DC 5V Provenientes de la conexión del teclado (laptop o desktop) usando adaptadores especiales. La corriente consumida es menor a 250 mA.
Velocidad de Transmisión	14,400 bps Full Duplex ó menores
Potencia de Transmisión	100 mW (Aproximadamente 300 m de espacio libre)
Frecuencia de Operación	902 – 928 MHz SST 10 canales, máximo 10 conexiones punto-punto
Selección de Canal	Auto Scan
Modulación	Digital Speed Spectrum 900 MHz
Link Layer Protocol	X.25 LAPB / HDLC

Tabla 4.9 "Especificaciones de los radio modems"

La operación del radio modem es la siguiente:



Gráfica 4.46 "radio modem 3JTech"

Una vez está conectado el cable de voltaje y el cable serial, se encienden los dos radio modems presionando el botón "Power Switch". Cuando los dos radio modems están encendidos, se presiona en cualquiera de los dos el botón "Start Link". Al presionar este botón se establece el enlace. Este enlace permanece hasta que algún radio modem es apagado.

En el radio modem están presentes tres luces indicadoras, que significan:

- MR (modem Ready): Hay un enlace existente entre los radio modems.
- TD (Transfer Data): Este radio modem está enviando información al otro.
- RD (Recibe Data): Este radio modem esta recibiendo información.

Los otros detalles técnicos, todas las funciones que realizar y los otros pasos para la instalación, aparecen en el Apéndice G.

Para la selección de estos radio modems, se investigó un buen número de opciones, entre ellas:

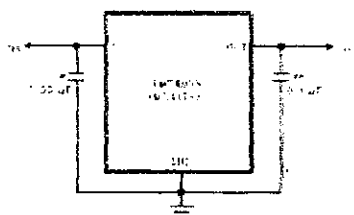
- ESTeem 72-73 MHz, 9,600 bps.
- SAI D8050/RFX 4,800 bps.
- Pacific Crest RFM96 9,600 bps
- Megadata SA9,600

Estas se descartaron porque necesitaban elementos secundarios como antenas, fuentes de poder, programadores, etc., o porque el precio era muy alto.

E.3 Conexión del radio modem en el Autómata

Para conectar el radio modem con el autómata es necesario tener dos elementos específicos. Una fuente de 5 VDC y una conexión RS232.

Se preparó una fuente adicional de 5 VDC, con una mayor protección a las fluctuaciones de voltaje:



Gráfica 4.47 "Fuente 5 VDC para el radio modem"

Al mismo tiempo que un conector para adaptar la salida del radio modem con la salida de convertidor RS232/RS485. Este adaptador funciona como un convertidor DB9 Macho / DB9 Macho.

El funcionamiento de este convertidor es intercambiar de las señales RX/TX del protocolo RS232:

Conector #1	Conector #2
Pin # 2 (Rx)	Pin # 3 (Tx)
Pin # 3 (Tx)	Pin # 2 (Rx)
Pin # 5 (GND)	Pin # 5 (GND)

Tabla 4.10 "Conexiones en el convertidor"

C.4 Convertidor RS232/RS485

Para convertir la información del formato RS232, formato utilizado en la computadora y los radio modems, al formato RS485, formato del PLC. Se utiliza el convertidor Siemens PC/PPI Cable.

Este convertidor trabaja con señales entre 1.2 Kbps a 38.6 Kbps. Se selecciona la velocidad por medio de un DIP switch, en el caso de los 9.6 Kbps el valor en el DIP switch es 0100.

C.5. Bus RS485 con tres PLCs conectados

El bus RS485 consiste de un par de alambres en los cuales la información se transmite mediante la inversión de la polaridad del voltaje. Dos o más dispositivos pueden estar conectados al mismo bus.

Es importante que cuando hay varios elementos en un bus, exista un protocolo definido para las direcciones, y otro para evitar las colisiones, es decir, que dos unidades no transmitan al mismo tiempo.

El primero de estos problemas se resuelve con las dos direcciones de cada mensaje. Todos los dispositivos reciben y transmiten en este formato.

En lo que se refiere a las colisiones, la configuración Principal / Auxiliares evita este problema. El PLC principal tiene la estafeta de la transmisión y solicita a los PLCs auxiliares desarrollar determinada función. Este PLC queda a la espera de escuchar una respuesta del auxiliar antes de transmitir de nuevo.

C.6. Recepción de información en los PLCs

Para recibir la información se utiliza el algoritmo que aparece en el Apéndice D. Sus principales rutinas son:

Programa principal:

```

NETWORK 1
LD    SM0.1
CALL  0

NETWORK 2 // Activa el freeport
LD    SM0.7
=     SM30.0

NETWORK 3 // Llama subrutina byte recibido
LD    M0.0
CALL  1

NETWORK 4 // Llama subrutina mensaje recibido
LD    M0.1
CALL  2

NETWORK 5
MEND

```

En el programa principal se activa el modo free port con el cual los PLCs pueden usar el puerto RS485 para comunicación.

Además aparecen las llamadas a las subrutinas 1 y 2. Estas revisarán si el mensaje recibido tiene la dirección del PLC, para luego descifrarlo. Las banderas con las que se llaman a estas subrutinas son M0.0 y M0.1 respectivamente.

Interrupción de Byte recibido:

```

NETWORK 25
LD    SM0.0
MOVB  VB111, VB110 // Movimiento de todos los bytes en el buffer de
                    // información recibida
MOVB  VB112, VB111
MOVB  VB113, VB112
MOVB  VB114, VB113
MOVB  VB115, VB114
MOVB  VB116, VB115
MOVB  VB117, VB116
MOVB  VB118, VB117
MOVB  VB119, VB118
MOVB  SMB2, VB119 // Incorporación del byte nuevo al buffer
S      M0.0, 1     // Activación bandera M0.0

```

En esta interrupción se efectúa un desplazamiento en el buffer de los bytes recibidos, corriéndolos un byte hacia adelante y colocando el nuevo byte en la última posición.

También se activa la bandera que indica un byte recibido M0.0.

Subrutina 1:

```

NETWORK 10          // Borrar bandera de entrada
LD      SM0.0
R       M0.0, 1

NETWORK 11          // Comparación banderas y direcciones
LDB=    VB110, VB101
AB=     VB119, VB101
AB=     VB111, VB100
MOVB   VB112, VB122 // Si coinciden se memoriza la información recibida
MOVB   VB113, VB123
MOVB   VB114, VB124
MOVB   VB115, VB125
MOVB   VB116, VB126
MOVB   VB117, VB127
MOVB   VB118, VB128
S       M0.1, 1      // Activa bandera M0.1

```

En la primera parte de esta subrutina se borra la bandera que permitió entrar a la misma.

Después se comparan los bytes 1 y 10 del buffer para ver si corresponden a la bandera de inicio y final (posición de memoria VB100), y se revisa si la dirección del mensaje corresponde al PLC (posición VB101).

Cuando coinciden estos tres valores la información se coloca en memoria: VB122-VB128. También se enciende la bandera de mensaje recibido M0.1.

Subrutina 2:

```

LD      SM0.0
R       M0.1, 1

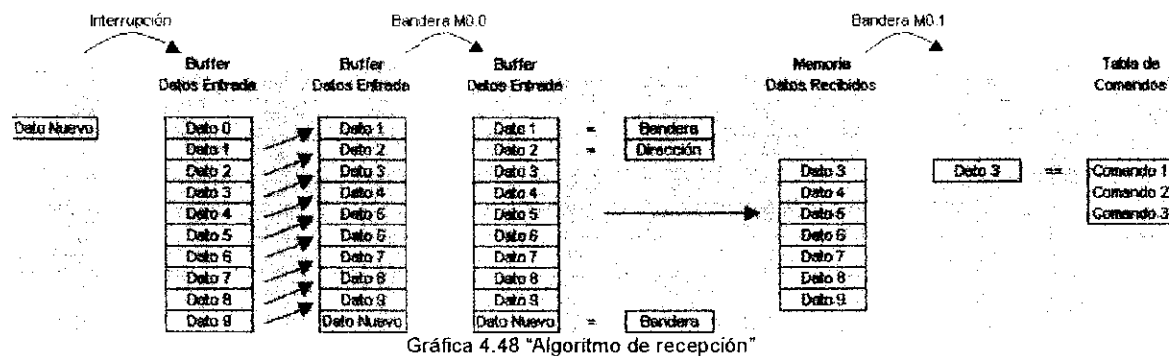
NETWORK 15
LDB=    VB123, 16#1
A       Q0.2
R       Q0.2, 1
R       Q0.0, 1
CRET

```

En esta subrutina primero se borra la bandera que permite la entrada.

Luego se comienza a revisar cuál es el valor que tiene la variable VB123, y dependiendo del mismo se levanta una bandera para realizar determinada función.

Gráficamente el funcionamiento de la rutina de Rx es el siguiente:



Gráfica 4.48 "Algoritmo de recepción"

En lo que se refiere a la transmisión de datos desde el PLC, ésta es muy sencilla y las rutinas ya están definidas.

Primeramente, se coloca en memoria una lista de bytes conteniendo en la primera posición el número de bytes a enviar y en las direcciones sucesivas los bytes a ser transmitidos.

```

NETWORK 12 // Inicialización Transmisión
LD      SMD.0
MOVB   9, SMB30           // Define el modo del puerto
ATCH   0, 8              // Activa interrupcion de Rx
MOVD   &VB130, VD102     // Activa puntero memoria de Tx
MOVB   10, VB130         // Inicializa memoria de Tx
MOVB   128, VB131
MOVB   0, VB132
MOVB   VB100, VB133
MOVB   0, VB134
MOVB   0, VB135
MOVB   0, VB136
MOVB   0, VB137
MOVB   0, VB138
MOVB   0, VB139
MOVB   128, VB140

```

Colocando las banderas de inicio, la de final y la dirección del PLC en su posición.

Para realizar la transmisión se ejecuta la función XMT:

```

MOVB   VB150, VB132
MOVB   VB151, VB134
MOVB   VB152, VB135
MOVD   +0, VD136
XMT    *VD102, 0

```

F. SISTEMA DE CONTROL

La función del sistema de control es recibir la información proveniente de los sensores y determinar cómo se deben realizar las funciones solicitadas.

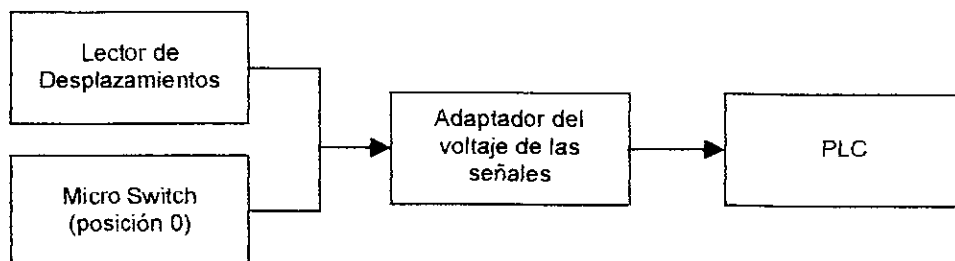
Para realizar el control se están utilizando los PLC Simens Simatic S-7 PLC 214, los cuales tienen las siguientes características técnicas:

- 2,948 palabras de memoria programable.
- 2,048 palabras de memoria de datos.
- Cartucho de memoria opcional.
- 14 entradas discretas y 10 salidas discretas.
- Soporte para 7 módulos adicionales de expansión I/O
- Capacidad de manejar 64 salidas o entradas discretas
- 128 temporizadores (cuatro de 1 ms, dieciséis de 10 ms, y ciento veintiocho de 100 ms)
- 128 contadores (96 contadores ascendentes y 32 contadores ascendentes/descendentes).
- 256 bits de memoria interna.
- 688 bits de memoria para fines especiales
- Matemática de cuatro funciones
- Capacidad de manejar interrupciones
- Interrupción de transmisión y recepción para protocolos definidos por el usuario
- 4 interrupciones en la transición ascendente o descendente de las señales de entrada
- 2 interrupciones temporizadas
- 7 interrupciones para contadores de alta velocidad
- 2 interrupciones para trenes de pulsos
- 1 contador de alta velocidad con entrada de 2 KHz.
- 2 contador de alta velocidad con entrada de 7 KHz:

- Soporta entradas en cuadratura en modo 1x hasta una velocidad de 7KHz
- Soporta entradas en cuadratura en modo 4x hasta una velocidad de 28KHz
- 2 pulsos de salida que pueden ser usados como Pulse Train Outputs (PTO) o salidas Pulse Width Modulation (PWM)
- 2 ajustes analógicos incorporados.
- Ejecución booleana rápida (0.8 microsegundos por instrucción)
- Capacidad de mantener la memoria hasta por 190 horas, gracias a un super capacitor.
- 3 niveles de protección con contraseñas.
- Reloj de tiempo real.

Hay dos PLCs encargados del funcionamiento del autómatas, cada uno de ellos controla dos ejes. Estos dos ejes funcionan dentro de cada PLC de forma independiente.

En cada PLC se controla el movimiento del eje determinando una posición inicial, el desplazamiento, y la velocidad. Estas funciones se realizan con sensores y adaptadores configurados de la siguiente forma:



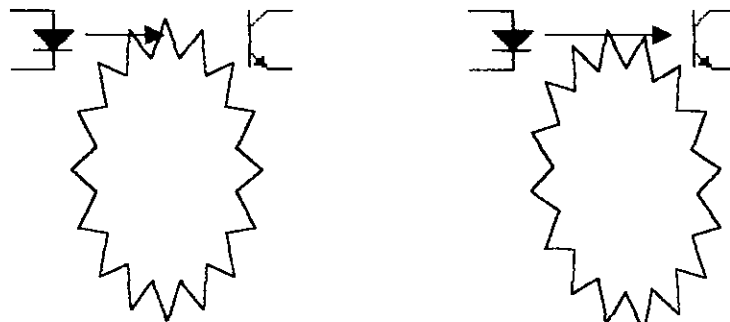
Gráfica 4.49 "Elementos sistema de control"

Los detalles de cada uno de los elementos son los siguientes:

F.1. Lector de desplazamientos

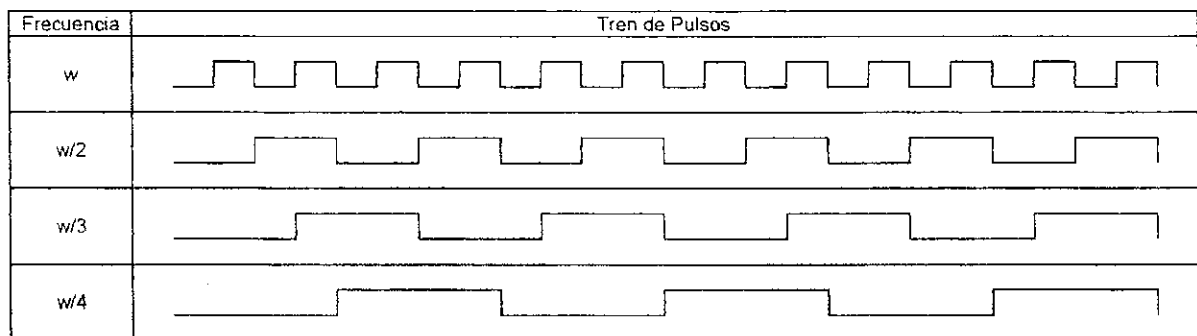
El lector de desplazamientos es una unidad Sharp Part No GP1R23R (Xerox No 146P00038). Esta unidad es un codificador que consiste de una rueda dentada con 360 surcos y dos optocoplas que detectan el movimiento de la rueda.

La optocopla está constituida por un fotodiodo y un fototransistor, en este caso el fotodiodo se encuentra conectado permanentemente y siempre emite fotones. La rueda dentada interrumpe el haz de fotones y produce un cambio en la salida.



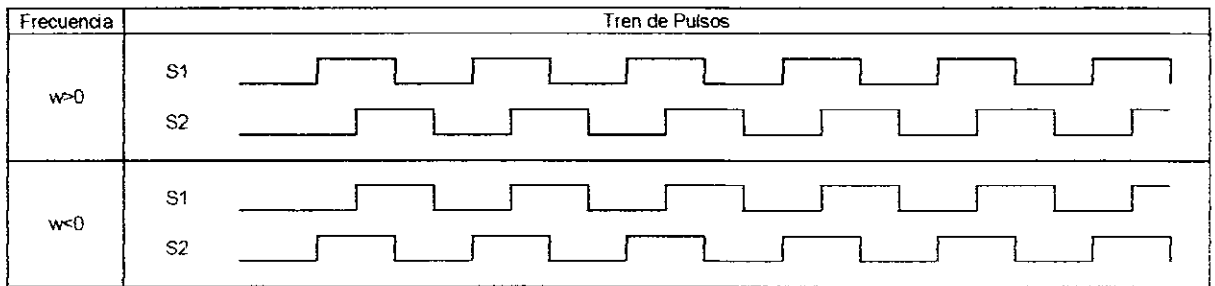
Gráfica 4.50 "Funcionamiento Unidad Sharp GP1R23R"

La salida de cada optocopla es un tren de pulsos cuya frecuencia es proporcionalmente lineal a la velocidad a la que gira la rueda:



Gráfica 4.51 "Relación tren de pulsos / velocidad angular"

La unidad Sharp cuenta con dos optocoplas, las cuales están desfasadas por $\frac{1}{4}$ de grado. Esto permite medir desplazamientos de $\frac{1}{4}$ de grado, así como determinar la dirección del giro:



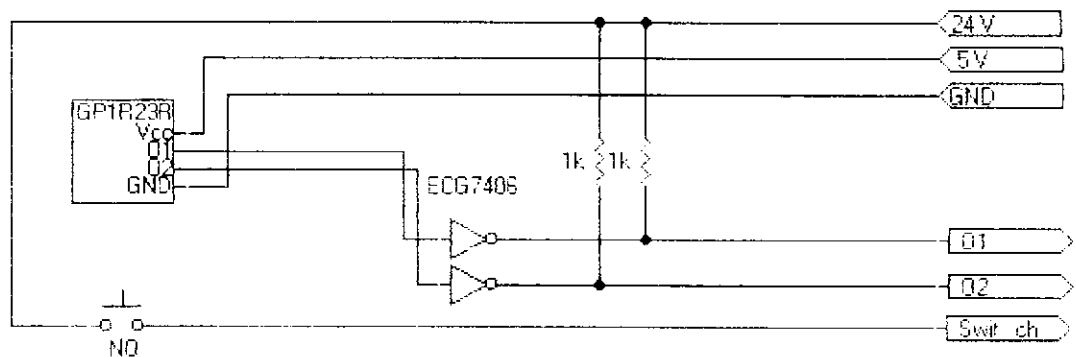
Gráfica 4.52 "Relación tren de pulsos / dirección"

F.2 Micro switch

El micro switch no es más que un botón normalmente abierto, el cual se cierra ante el contacto físico. Se utilizan en el autómatas para definir cual será la posición inicial, en cada uno de los ejes.

F.3 Adaptador del voltaje de las señales

Las señales de los sensores no tienen el voltaje usarse en el PLC. Para que los datos puedan ser leídos, se utiliza el siguiente circuito:



Gráfica 4.53 "Relación tren de pulsos / velocidad angular"

El micro Switch es alimentado con un voltaje de 24 V. Y la unidad Sharp GP1R23R es alimentada con un voltaje de 5 V. Sus salidas son conectadas a la compuerta ECG7406. Como la salida de la compuerta ECG7406 es Open Colector High Voltaje, puede manejar 24 V.

La unidad Sharp es una pieza reciclada, fue encontrada entre los desechos de fotocopiadoras y a partir de ese punto se realizaron. Los alumnos del laboratorio de Sistemas de Control II comenzaron a emplear este dispositivo. Amplificando la señal de salida de 0.1 V a 5 V a través de OpAmps.

El autor de este trabajo en experimentos posteriores encontró que la pieza es compatible con TTL.

F.4 controlador Lógico Programable (PLC)

Los algoritmos desarrollados para el control en los PLC Simens S-7 se encuentran en el Apéndice E. Todo el control de posición en los PLCs se realiza con los contadores de alta velocidad. Existen 3 contadores de alta velocidad en cada PLC, pero solamente dos de ellos pueden trabajar hasta 28 KHz.

El funcionamiento detallado de cada uno de los contadores aparece en el manual STEP 7-Microprogramming Reference Manual, sección 6.1. Sin embargo es importante definir la programación utilizada.

Primeramente, hay que revisar el funcionamiento de los diferentes bytes y bits donde se almacena la información y la programa el contador. Los dos contadores usados HC1 y HC2 tienen el mismo juego de bytes/bits de control, se describe solamente la información del contador HC1:

Byte / Bit	Descripción
I0.6	Fase A del reloj
I0.7	Fase B del reloj
I1.0	Reset
I1.1	Start
SM47	Configuración del contador 1:
SM47.7	Habilitar 1 / Deshabilitar 0
SM47.6	Actualizar valor actual
SM47.5	Actualizar valor predefinido
SM47.4	Actualizar dirección
SM47.3	Definir dirección ascendente / descendente
SM47.2	Determina el modo x1 ó x4
SM47.1	Determina el pulso Reset en ON ó en OFF
SM47.0	Determina el pulso Start en ON ó en OFF
SM48 – SM51	Palabra doble utilizada para actualizar el valor predefinido
SM52 – SM55	Palabra doble utilizada para actualizar el valor predefinido
SM46	Byte de status del contador
HC1	Valor actual del contador

Tabla 4.11 "Bytes y bits para configurar el contador HC1"

El contador se puede operar en 11 modos diferentes. Se utilizó el modo 10: contador x4 ascendente ó descendente con reloj externo de fase doble (dos entradas) y Reset Externo

En el algoritmo de control para programar el contador HC1 es:

```

NETWORK 4 //      Inicializa posición y velocidad 1
LD      SMO.0
MOVD   -0, VD0    // Inicializar variable 1 posición actual
MOVD   +0, VD4    // Inicializar variable 1 posición pasada
MOVD   +0, VD8    // Inicializar variable 1 velocidad
MOVD   &VB1000, VD12 // Inicializa puntero de memoria
MOVD   &VB2000, VD16 // Define ancho de memoria
MOVVB  16#F8, SMB47 // Definición del contador 1 x4 con Reset
HDEF   1, 10
MOVD   +0, SMD48
MOVD   +0, SMD52
HSC    1
MOVVB  100, SMB34 // Definición de la velocidad de muestreo
ATCH   0, 10

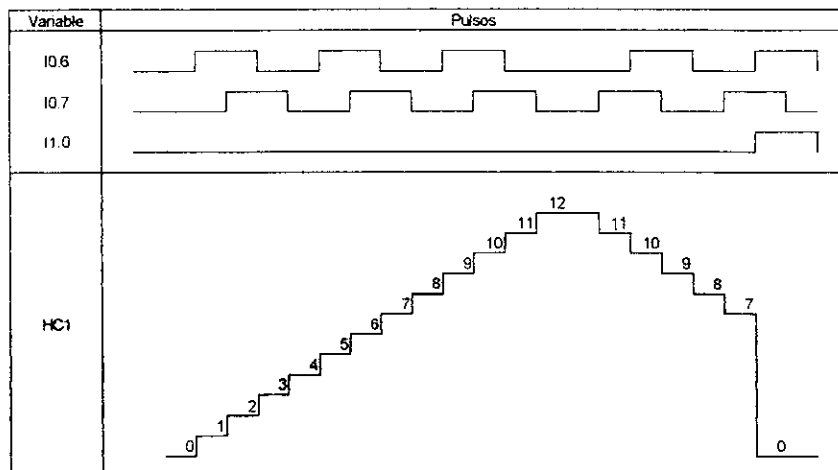
```

Donde los pasos seguidos han sido:

- Inicializar las variables de posición con cero.
- Almacenar el valor 11111000 en el byte de control de contador.
- Definir el modo de trabajo de contador como modo 10.
- Inicializar los valores actuales y predefinidos del contador con cero
- Programar el contador.

El CPU revisa los cambios en la señales de la unidad sharp, suma o resta en el contador dependiendo de cual entrada cambia primero. Y cuando el micro switch activa la entrada 11.0 el valor del contador es puesto en cero. De esta forma se registran los cambios de posición.

El valor de la posición siempre está almacenada en HC1. Gráficamente la operación del contador es la siguiente:



Gráfica 4.54 "Funcionamiento del contador HC1"

Para determinar la velocidad se utilizan las interrupciones temporizadas del PLC, nuevamente hay dos interrupciones temporizadas que trabajan independientemente, por lo cual, las velocidades de cada eje son independientes.

Para determinar la velocidad lo que se hace es restar dos posiciones, las cuales se miden con un tiempo de diferencia. En el caso particular del autómatas este periodo de tiempo es de 100 ms ó 0.1 s. La programación de esta interrupción se realiza con el siguiente algoritmo:

```

MOVW 100, SMB34 // Definición de la velocidad de muestreo
ATCH 0, 10

```

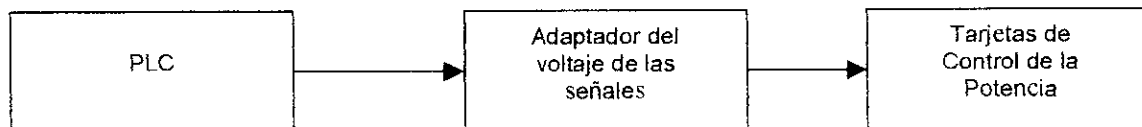
Donde se indica que cada vez que pasen 100 ms deberá ser llamada la interrupción cero:

```

LD SMO.0
MOVD VD0, VD4 // Mueve el valor anterior a su posición
MOVD HC1, VD8 // Mueve el valor actual a la variable de velocidad
MOVD VD8, VD0 // Mueve el valor actual a su posición
-D VD8, VD4 // Resta el valor actual del anterior y obtiene la
                velocidad

```

En esta interrupción se guarda la posición actual para ser utilizada en la siguiente medición, y se compara la medición anterior con la actual. Esa consulta siempre se realiza con el mismo periodo de tiempo. El valor de la velocidad se encuentra en todo momento en la variable VD4.



Gráfica 4.55 "Elementos del sistema de actuación"

Para controlar las tarjetas, el PLC utiliza sus salidas PWM. Nuevamente hay dos salidas en cada PLC, las cuales no alcanzan para manejar los dos ejes. La solución de este problema se describe más adelante.

La función detallada de cada PWM aparece en el manual STEP 7-Microprogramming Reference Manual, sección 6.3. Nuevamente, para describir el funcionamiento del autómatas se debe comprender la programación:

Byte / Bit	Descripción
Q0.0	Salida del PWM
SM67	Configuración del PWM 1:
SM67.7	Habilitar 1 / Deshabilitar 0
SM67.6	Seleccionar PWM 1 / PTO 0
SM67.5	No usado
SM67.4	No usado
SM67.3	Base del tiempo: 1us 0 / 1ms 1
SM67.2	Actualizar valor actual
SM67.1	Actualizar ancho del pulso
SM67.0	Actualizar frecuencia
SM68 – SM69	Palabra utilizada para actualizar el valor de la frecuencia
SM70 – SM71	Palabra utilizada para actualizar el valor del ancho del pulso
SM72 – SM75	Valor del PWM

Tabla 4.12 "Funcionamiento del PWM"

Para programar el PWM1 se utiliza el siguiente algoritmo:

```

MOVB 16#C3, SMB77
MOVW +5000, SMW78 // Modifica el valor enviado en Com2 como % del
PWM
MOVD +0, VD50
MOVB VB124, VB53
MUL +50, VD50 // Guarda el valor del ancho del pulso
MOVW VW52, SMW80 // Activa el PWM
PLS 1

```

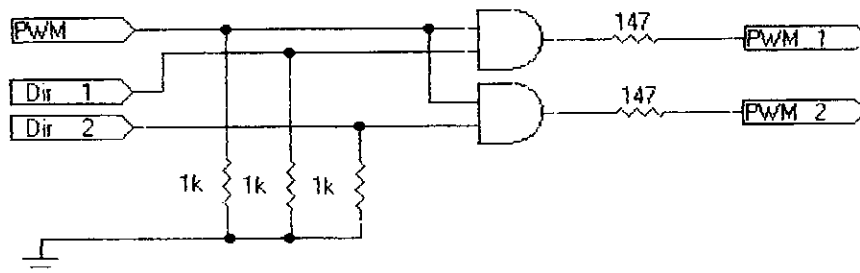
Donde los pasos seguidos han sido:

- Definir las características del PWM: 11000011.
- Definir la frecuencia del PWM: 200 Hz.
- Determina el valor del ancho del pulso a ser utilizado, multiplicando el dato recibido en Comando 2 por 500.
- Inicializa el PWM.

Para controlar cada una de las tarjetas, se necesitan dos señales PWM, sin embargo cada PLC sólo cuenta con una salida PWM para cada tarjeta, entonces se utilizan dos bits de control de dirección por cada eje.

Es decir, adicional al PWM que indica la velocidad, hay dos bits que indican la dirección. Estos bits nunca pueden estar encendidos simultáneamente para no dañar la tarjeta de potencia.

Las tres señales son combinadas en una compuerta AND para dos pulsos PWM. El circuito de esta compuerta es:



Gráfica 4.56 "Circuito compuertas AND"

Donde las resistencias de $1\text{ K}\Omega$ en la entrada garantizan que aunque la conexión al PLC se interrumpa, las salidas de las compuertas AND no serán 5V. Las compuertas AND tienen que recibir como entrada 5V, para ello se modifica la alimentación de las salidas de los PLC.

Finalmente la compuerta está conectada a las tarjetas de potencia mediante una resistencia de $147\ \Omega$

F.5 Desarrollo del Sistema de Control

Para desarrollar el circuito de potencia se trabajaron varios modelos, los cuales se fueron descartando en función de los nuevos elementos disponibles, así como por la capacidad de programación, puertos de salida y otros elementos.

Los diferentes sistemas desarrollados fueron:

A. Implementación de un PLC S7 214 con microprocesadores PIC 17C44:

Los detalles de cómo se implementó un PLC con microprocesadores PIC 17C44 y porque no se utilizaron aparecen en el capítulo 4.7

B. Implementación de funciones de control con microprocesadores PIC 16F84:

El primer intento por desarrollar un control con microprocesadores PIC fue llevado a cabo por los alumnos del laboratorio de Sistemas de Control II. Se desarrollaron aplicaciones señalizadas donde se controlaba la velocidad o la posición mediante rutinas sin control digital. Esto se debía a que los microprocesadores no tenían capacidad matemática para desarrollar aplicaciones de control.

Como resultado de estos trabajos se tuvo:

- Un dispositivo que alcanzaba una de las 6 posiciones indicadas por 3 bits de control.
- Un dispositivo cuya salida era linealmente proporcional a la diferencia entre la velocidad actual y la velocidad deseada (3 bits de control).

G. CONTROLADOR LOGICO PROGRAMABLE CON PIC 17C44

Uno de los objetivos de este trabajo fue implementar los PLC que necesitaba el autómata con microprocesadores PIC. De las familias de microprocesadores 16XX y 17XX se escogió el PIC 17C44 que tiene las siguientes características:

Memoria Programable	8192 palabras
Memoria RAM	232 bytes
Velocidad Máxima	33 MHz
Puertos entrada/salida	33
Puerto serial	USART
PWM	2
Contadores	Cuatro + WDT
Otros	20 mA en cada salida 35 mA en cada entrada 2 entradas de 60mA Capacidad de manejar memoria externa Multiplicación 8x8 de un ciclo.

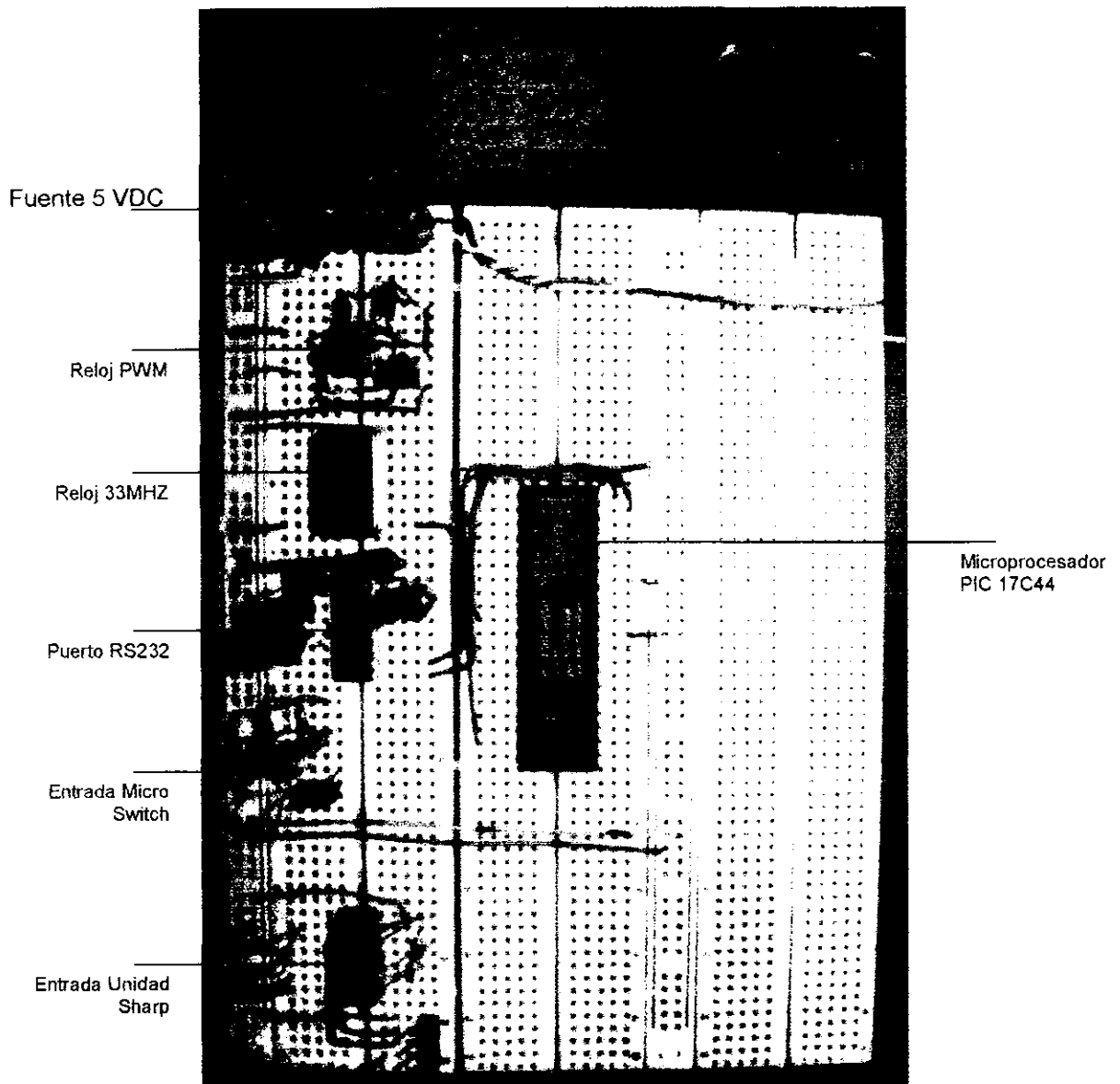
Tabla 4.13 "Características PIC17C44"

Desarrollar un PLC con las siguientes características:

- 8,192 palabras de memoria programable EPROM (solamente 10 programaciones permitidas).
- 232 bytes de memoria de datos.
- Puerto RS485 con capacidad de manejar paquetes de 10 bytes (2 banderas, 2 direcciones, 2 comandos y 4 datos)
- 2 salidas PWM cuya frecuencia queda determinada por un reloj externo.
- Lectura de la posición decodificando la posición de la unidad Sharp GR1R123R 14. Resolución de la posición de 1 grado.
- Determinación de la velocidad mediante la medición del ancho del pulso de la unidad Sharp GR1R123R.

El algoritmo que se programó para realizar estas funciones se encuentra en el Apéndice F. El desarrollo de este algoritmo, así como su simulación y posterior implementación se realizó utilizando el programa MPLAB de Microchip. El cual se puede obtener gratuitamente en la dirección de Internet www.microchip.com

El resultado de este trabajo fue una unidad compuesta por las siguientes partes:



Gráfica 4.57 "PLC con PIC 17C44 (OTP)"

Sin embargo, aunque se obtuvo una unidad funcional en el protoboard, no se implementó esta solución en el autómatas por dos razones:

- No se cuenta en la Universidad con el equipo adecuado para preparar un circuito impreso para esta unidad. Al no tener un circuito impreso es muy difícil ubicarla en alguna parte del robot y garantizar su funcionamiento.

Se hicieron intentos por trabajar la unidad con la técnica "Wire Rap" pero no se logró hacer funcionar, se concluyó que la frecuencia de oscilación introduce mucha interferencia. Finalmente, con la técnica de circuitos impresos realizados con ácido, no se obtuvo un resultado funcional.

- Los PLC Siemens S-200 tienen sus contadores HC1 y HC2 desarrollados mediante hardware, esto quiere decir que si el PLC está realizando cualquier instrucción, interrupción u otra función siempre contará cualquier desplazamiento que envíe el detector de desplazamientos.

Mientras que los microprocesadores PIC, la función de contar los desplazamientos está realizada mediante software, y cuando se inhabilitan las interrupciones porque se está desarrollando otra interrupción o algunas funciones, el microprocesador no puede contar los desplazamientos que el sensor está leyendo. Esto constituye un error muy grave porque las posiciones se determinan sumando y restando desplazamientos, si se pierde alguno de nada sirve sumar o restar los restantes.

Por estas dos razones, pero principalmente por la segunda se descartó el uso de los PLC desarrollados con estos microprocesadores. Adicionalmente, se contó en ese momento con los PLC Siemens adicionales que permitían el control completo del autómatas.

H. PROGRAMACION

La programación del autómeta no está completamente desarrollada, corresponde a trabajos posteriores ir refinando e implementando soluciones que respondan mejor.

La programación que existe en este momento es únicamente la función más elemental como lo es el control de los motores, inicializar los ejes, alcanzar una posición específica y tener control sobre los motores para prenderlos y apagarlos

Como se ha visto en las secciones de comunicación y control, el sistema del autómeta está compuesto por tres PLCs. Los cuales funcionan uno como principal, y dos como auxiliares

Las direcciones de cada PLC son:

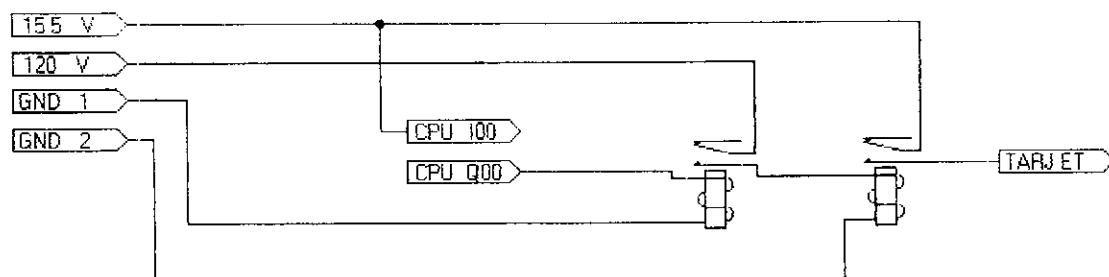
PLC	Dirección
Principal	1
Auxiliar 1	2
Auxiliar 2	3

Tabla 4.14 "Direcciones PLCs"

H.1 Programación del PLC principal

Actualmente la programación del PLC principal es muy sencilla, solamente controla la conexión de las tarjetas de control de potencia. La programación completa de este PLC se encuentra en el Apéndice C

Como se vio en el capítulo 4.3, es necesario que los transistores de potencia sean conectados después de los transistores de control y para ello se usan dos contactores.

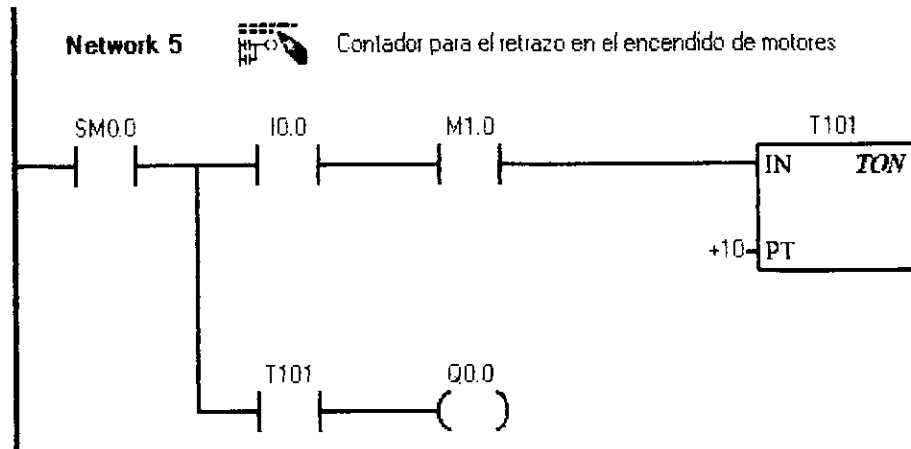


Gráfica 4.58 "Control PLC principal sobre alimentación de tarjetas"

Con este circuito el PLC revisa si existe alimentación a través de la entrada I0.0. La lectura de este valor del voltaje se hace a través de un cable que está soldado al cable de las tarjetas, por lo que en ningún momento se recibirá una señal equivocada.

Cuando el PLC determina que existe alimentación, utiliza uno de sus contadores rápidos para contar un segundo y luego encender la salida Q0.0.

Esto se hace con el contador TON 101. El funcionamiento de la función es el siguiente, si la entrada del contador TON 101 permanece encendida por más de 1 segundo entonces el valor de T101 es uno, si la salida de TON 101 se apaga, entonces inmediatamente el valor de T101 se convierte en cero. El algoritmo en el que se utiliza esta función es el siguiente:



Gráfica 4.59 "Programación contador y salida PLC principal"

Una vez transcurrido el segundo de espera, la salida Q0.0 activa el primer contactor que tiene bobina de 24 VDC, y ésta a su vez un segundo contactor con bobina de 120VAC. Este realiza la conexión de 16.5 V y tierra, para los transistores de potencia de las tarjetas

Adicionalmente, el PLC principal puede responder a dos instrucciones de la computadora para activar o desactivar estos contactores por control remoto.

Las instrucciones son:

Instrucción		Acción
Bandera:	128	Apaga los contactores y suspende la alimentación a los transistores de potencia de las tarjetas.
Destino:	1	
Origen:	X	
Comando1:	0	
Comando2:	X	
Dato1:	X	
Dato2:	X	
Dato3:	X	
Dato4:	X	
Bandera:	128	
Bandera:	128	Enciende los contactores y activa la alimentación a los transistores de potencia de las tarjetas.
Destino:	1	
Origen:	X	
Comando1:	1	
Comando2:	X	
Dato1:	X	
Dato2:	X	
Dato3:	X	
Dato4:	X	
Bandera:	128	

Tabla 4.15 "Instrucciones PLC 1"

Con estas dos instrucciones se activa y desactiva el bit M1.0, el cual es necesario para que el contador TON 100 permanezca activado, ya que constituye una de sus entradas.

H.2. Programación de los PLCs auxiliares

Los dos PLCs auxiliares tienen la misma programación, la única diferencia es la dirección de cada uno de los PLCs. Hay cuatro instrucciones implementadas en cada PLC:

Instrucción		Acción
Bandera:	128	Inicializa el eje 1 del PLC, activando el motor del eje con una velocidad determinada por el Comando 2 (0% - 100%). En el momento en que el PLC llega al origen responde.
Destino:	2/3	
Origen:	X	
Comando1:	10	
Comando2:	0/100	
Dato1:	X	
Dato2:	X	
Dato3:	X	
Dato4:	X	
Bandera:	128	

Instrucción	Acción
Bandera: 128 Destino: 2/3 Origen: X Comando1: 11 Comando2: 0/100 Dato1: Destino MSByte Dato2: Destino Dato3: Destino Dato4: Destino LSByte Bandera: 128	Pide al autómata colocarse en el eje 1 en la posición de destino (doble word), para ello debe encender el motor del eje con una velocidad determinada por el Comando 2 (0% - 100%). En el momento en que el PLC llega al destino responde.
Bandera: 128 Destino: 2/3 Origen: X Comando1: 20 Comando2: 0/100 Dato1: X Dato2: X Dato3: X Dato4: X Bandera: 128	Inicializa el eje 2 del PLC, activando el motor del eje con una velocidad determinada por el Comando 2 (0% - 100%). En el momento en que el PLC llega al origen responde.
Bandera: 128 Destino: 2/3 Origen: - X Comando1: 21 Comando2: 0/100 Dato1: Destino MSByte Dato2: Destino Dato3: Destino Dato4: Destino LSByte Bandera: 128	Pide al autómata colocarse en el eje 2 en la posición de destino (doble word), para ello debe encender el motor del eje con una velocidad determinada por el Comando 2 (0% - 100%). En el momento en que el PLC llega al destino responde.

Tabla 4.16 "Instrucciones PLC 2 ó 3"

Para realizar estas instrucciones primero se decodifican el mensaje recibido, activando los bits de memoria M10.0, M11.0, M20 ó M21 dependiendo de la instrucción recibida:

```

NETWORK 22 // Borra la bandera de entrada a esta subrutina
LD  SM0.0
R    MD.1, 1

```

```

NETWORK 23 // Compara el comando1 con "10" que significa inicializar 1
LDB= VB123, 10
S    M10.0, 1
S    M10.1, 1

```

```

NETWORK 24 // Compara el comando1 con "11" que significa ir a 1
LDB= VB123, 11
S    M11.0, 1
S    M11.1, 1

```

```

NETWORK 25 // Compara el comando1 con "20" que significa inicializar 2
LDB= VB123, 20
S    M20.0, 1
S    M20.1, 1

```

```

NETWORK 26 // Compara el comando1 con "21" que significa ir a 2
LDB= VB123, 21

```

```

S      M21.0, 1
S      M21.1, 1

```

En el caso de que la instrucción recibida sea inicializar el eje entonces se encenderán dos banderas M10.0 y M10.1 (M20.0 y M20.1 para el eje 2 , pero en lo sucesivo se analizará solo el eje 1). Estas banderas permiten que desde el programa principal se llame a la subrutina 3:

```

NETWORK 5 // Llama subrutina inicializar eje 1
LD      M10.0
CALL    3

```

Esta subrutina se realizará en cada ciclo del PLC hasta que el autómatas haya alcanzado la posición de inicio. La primera vez que se llama a la subrutina se enciende el PWM así como el bit que determina la dirección decreciente en el eje:

```

NETWORK 29
LD      M10.1
MOVW   16#C3, SMB67 // Activa el PWM1
MOVW   +5000, SMW68
MOVD   +0, VD40 // Modifica el valor enviado en Com2 como % del PWM

MOVW   VB124, VB43
MUL    +50, VD40
MOVW   VW42, SMW70 // Guarda el valor del ancho del pulso
PLS    0 // Activa el PWM
MOVW   16#C2, SMB67
S      Q0.2, 1 // Activa la salida de dirección
R      M10.1, 1 // Elimina bandera para ejecutar esta rutina de nuevo
MOVW   VB122, VB150 // Archiva los datos que enviará cuando termine la rutina
MOVW   VB123, VB151
MOVW   VB124, VB152

```

En lo sucesivo esta subrutina solamente revisará el valor presente en el micro switch, cuando éste se encienda apagará el PWM y transmitirá de vuelta al PLC:

```

NETWORK 30
LD      I1.0 // Verifica la entrada de posición cero
MOVW   16#42, SMB67
PLS    0
R      Q0.2, 1 // Borra la salida de dirección
R      M10.0, 1 // Elimina la bandera para entrar a la subrutina 3
MOVW   VB150, VB132 // Mueve los datos al buffer de salida
MOVW   VB151, VB134
MOVW   VB152, VB135
MOVD   +0, VD136
XMT    *VD102, 0, // Transmite

```

El mensaje transmitido será el siguiente:

Bandera	128
Destino	La dirección de origen del mensaje original
Origen	2/3
Comando1	11
Comando2	En valor 0/100 recibido originalmente.
Dato1	0
Dato2	0
Dato3	0
Dato4	0
Bandera	128

Tabla 4.17 "Respuesta PLC 2 6 3 a comando 11"

Cuando la instrucción recibida es ir a una posición determinada, entonces se encenderán dos banderas M11.0 y M11.1 (M21.0 y M21.1 para el eje 2 , pero nuevamente se analizará solo el eje 1). La dirección a la cual se desea llegar estará determinada por los cuatro bytes de Dato 1 – Dato 4, siendo Dato 1 el byte más significativo y Dato 4 el menos significativo, la información de este dato es en formato de entero con signo.

El signo negativo representa posiciones anteriores al micro switch, por lo que se debe evitar enviar datos de este tipo.

Esta bandera permiten que desde el programa principal se llame a la subrutina 4:

```
NETWORK 6 // Llama subrutina ir a esta posición eje 1
LD M11.0
CALL 4
```

Esta subrutina se realizará en cada ciclo del PLC hasta que el autómatas haya alcanzado la posición deseada. La primera vez que se llama a la subrutina se enciende el PWM y se calcula que bit de dirección debe encenderse dependiendo de la posición actual y la posición que se busca:

```
LD M11.1
LPS // Activa el PWM1
MOVB 16#C3, SMB67
MOVW +5000, SMW68 // Modifica el valor enviado en Com2 como % del PWM
MOVD +0, VD40
MOVVB VB124, VB43
MUL +50, VD40 // Guarda el valor del ancho del pulso
MOVW VW42, SMW70 // Activa el PWM
PLS 0
MOVB 16#C2, SMB67
R Q0.3, 2 // Borra las dos salidas de dirección
MOVD VD125, VD44 // Realiza la resta para determina la dirección
-D HC1, VD44
```

```

A      SM1.2
S      Q0.2, 1          // Dependiendo del resultado activa una u otra
                        // dirección
LRD
AN     Q0.2
S      Q0.3, 1
LRD
R      M11.1, 1        // Borra la bandera para no hacer el cálculo
                        // nuevamente
LPP
MOVD   VD125, VD44     // Archiva los datos que enviará cuando termine la
                        // rutina

MOVEB  VB122, VB150
MOVEB  VB123, VB151
MOVEB  VB124, VB152

```

Siempre se revisa antes de encener un bit de dirección cuál es el estado de la dirección contraria. Una vez activo el PWM, se continuará realizando esta subrutina, para determinar si se alcanzó la posición deseada. Una vez se llega se desactiva el PWM, los bits de dirección y transmite un mensaje:

```

NETWORK 34          // Subrutina de posición decreciente
LDD<=  HC1, VD44   // Verifica si llegó a la posición
A      Q0.2
MOVEB  16#42, SMB67
PLS    0
R      Q0.2, 1     // Borra la salida de dirección
R      M11.0, 1   // Elimina la bandera para entrar a la subrutina 3
MOVEB  VB150, VB132 // Mueve los datos al buffer de salida
MOVEB  VB151, VB134
MOVEB  VB152, VB135
MOVD   HC1, VD136
XMT    *VD102, 0   // Transmite

NETWORK 35          // Subrutina de posición creciente
LDD>=  HC1, VD44   // Verifica si llegó a la posición
A      Q0.3
MOVEB  16#42, SMB67
PLS    0
R      Q0.3, 1     // Borra la salida de dirección
R      M11.0, 1   // Elimina la bandera para entrar a la subrutina 3
MOVEB  VB150, VB132 // Mueve los datos al buffer de salida
MOVEB  VB151, VB134
MOVEB  VB152, VB135
MOVD   HC1, VD136
XMT    *VD102, 0   // Transmite

```

El mensaje enviado es el siguiente:

Bandera	128
Destino	La dirección de origen del mensaje original
Origen	2/3
Comando1	21
Comando2	En valor 0/100 recibido originalmente.
Dato1 – Dato 4	Posición actual
Bandera	128

Tabla 4.18 "Respuesta PLC 2 ó 3 a comando 21"

V. RESULTADOS

Se desarrolló e implementó un autómata montacargas. La mayor parte del trabajo se realizó con componentes que se pueden conseguir en el mercado guatemalteco. La mayoría de las piezas fueron compradas y/o desarrolladas localmente, otras fueron donadas por compañías como Siemens y Procter & Gamble, otras fueron recicladas de la basura y finalmente algunas piezas se importaron del exterior.

Los tres ejes con los que cuenta el autómata permiten mover tarimas entre una estación de trabajo y una estantería profunda.

Los resultados específicos del trabajo son:

1. Se desarrolló e implementó un autómata que funcionará como laboratorio de Sistemas de Control II. En él se pueden medir parámetros de control como la posición, velocidad y tiempo transcurrido entre eventos; al modificar el algoritmo de control de velocidad, se pueden implementar filtros digitales; y medir los cambios en los parámetros anteriores.
2. Se desarrolló la estructura mecánica del autómata. Esta estructura cuenta con tres ejes de rotación: horizontal, vertical y profundidad.

Esta disposición le permite recoger y entregar tarimas entre una estación de trabajo y una estantería profunda. La estación de trabajo es atendida gracias al movimiento de rotación, y la estantería profunda se atiende de frente. Para recoger o entregar las tarimas se cambia de nivel en el eje vertical.

La estructura del autómata permite aumentar el número de ejes a cinco, agregando profundidad y cambios de vías.

3. Se desarrolló un sistema de alimentación, que permita el funcionamiento de forma automática del montacargas. La alimentación se realiza a través de los rieles y los voltajes se obtienen mediante un conjunto de transformadores. En el autómata se cuenta con fuentes de +5VDC, +16.5VDC, +24VDC y 120VAC.

- 4 Se desarrolló un sistema de comunicación mediante radio modems que permita un enlace permanente entre el autómata (RS485) y una computadora personal (RS232). Esta comunicación inalámbrica permite transmitir y recibir cualquier tipo de datos dentro del formato específico. La transmisión se realiza sin importar la posición o la velocidad del autómata.
- 5 Se desarrolló un sistema que transforma las señales de control del PLC en señales de potencia, hasta 450W. Para ello se cuenta con unas tarjetas que transforman la señal del PLC, en estas tarjetas se utiliza una solución muy ingeniosa, que permite aplicar en los motores un voltaje que varía entre $-13.5V$ y $+13.5V$ contando solamente con $+16.5V$ de alimentación.
- 6 Se desarrolló un sistema de control que consiste de PLC, sensores, actuadores y algoritmos. Este sistema permite el movimiento controlado, es decir, se sabe cual es la posición de cada uno de los ejes, y se puede modificar.

La posición se determina mediante sensores que tiene una resolución de $\frac{1}{4}$ de grado en el eje de rotación; 0.02 cm en el eje vertical y 0.07 cm en el eje horizontal. Los cambios en cualquier eje se registran en los PLC no importando qué parte del programa, qué interrupción o qué función esté ejecutando.

Para modificar los valores de posición se activa el movimiento en cualquier dirección con alguna de las 100 velocidades disponibles, se puede determinar con qué velocidad se está moviendo el eje y modificar el voltaje aplicado al motor para alcanzar la velocidad deseada.

- 7 La seguridad de los datos de todo el sistema de control está garantizada por el UPS. Una vez se realiza la rutina de inicialización se conoce la posición exacta en cada uno de los ejes.

VI. RECOMENDACIONES

Observando el trabajo realizado, los resultados obtenidos, los fondos invertidos y todo el potencial latente en este trabajo, la más importante recomendación es que se establezca un fondo para la investigación y el desarrollo de soluciones automátatas en la Universidad del Valle de Guatemala

Muchas empresas están dispuestas a contribuir, pero la Universidad debe dar el primer paso

Con este trabajo queda marcado el camino a seguir, pero serán muchos los que por motivos económicos no puedan contribuir a él. Por eso es tan importante que no se desperdicie ese talento y que sea la Universidad quien promueva el desarrollo interno de este campo

En segundo lugar, y desde un punto de vista puramente funcional, los trabajos que se deben realizar en el autómatata son:

1. La fuente de voltaje se debe mejorar en lo que se refiere a los casos de picos de voltaje producidos al apagar los motores y el caso de cortocircuitos en alguno de los elementos de control
2. La programación para controlar el movimiento en cada eje se debe mejorar. Deben realizarse mediciones para determinar las características de cada eje y preparar filtros que permitan un funcionamiento más controlado
3. La programación en el PLC Principal se debe estructurar para que sea éste quien dé las órdenes a los PLCs Auxiliares. Permite de esta forma que el automata solamente reciba una instrucción y que todas las instrucciones internas se generen desde este PLC.

Es decir, que se le indique al autómatata recoger una tarima y entregarla en determinada posición. Y que el PLC Principal conozca dónde quedan las posiciones y qué instrucciones debe darle a sus Auxiliares.

4. Implementar el sistema de cambio de vías, mediante una modificación en los rieles y siguiendo el diagrama del capítulo 4.1. El cambio de vías permitirá aumentar el tamaño de la bodega a utilizar, así como simplificar el procedimiento de carga y descarga. Una vez implementado el cambio de vías el autómata está preparado para funcionar en cualquier bodega.
5. Implementar el eje de profundidad, con este eje el autómata podrá atender prácticamente cualquier tipo de bodega o estructura donde se almacenen tarimas.

VII. BIBLIOGRAFIA

- Sedra, A. 1989. Microelectronic Circuits. 3º Edición. Saunders College Publishing. Estados Unidos.
- Phillips ECG. 1987. ECG Semiconductors Master Replacement Guide. 16º Edición ECG Canda INC Estados Unidos.
- Hubert, C. 1990. Circuitos Eléctricos CA/CC Enfoque Integrado. 1º Edición. Mc Graw Hill. México.
- Millman, J. 1993. Microelectronica. 6º Edición. Editorial Hispano Europea. España.

DESARROLLO E IMPLEMENTACION DE UN AUTOMATA

MONTACARGAS

Libro de Apéndices

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades
Departamento de Ingeniería Electrónica



DESARROLLO E IMPLEMENTACION DE UN AUTOMATA
MONTACARGAS

Libro de Apéndices

FRANCISCO JAVIER RODRIGUEZ MAZARIEGOS

Trabajo de graduación presentado para optar
al grado académico de Ingeniería Electrónica

Guatemala
1998

APENDICES

		Páginas
A	Programa puerto 01	A-1
B	Programa CPU Auxiliar	B-1
C	Programa CPU Principal	C-1
D	Programa CPU Tx/Rx de datos	D-1
E	Programa CPU Control entradas y salidas	E-1
F	Programa PIC 17C44	F-1
G	Especificaciones RadioMódems	G-1

Apéndice A

Programación de la computadora "Puerto01"

VERSION 4.00

```
Begin VB.Form Form1
    Caption           = "Interface para Puerto Serial"
    ClientHeight     = 6852
    ClientLeft       = 2412
    ClientTop        = 1404
    ClientWidth      = 6756
    Height           = 7172
    Left             = 2364
    LinkTopic        = "Form1"
    ScaleHeight      = 6852
    ScaleWidth       = 6756
    Top              = 1128
    Width            = 6852
Begin VB.Frame Frame2
    Caption           = "Com K: "
    Height           = 632
    Left            = 4080
    TabIndex        = 27
    Top             = 120
    Width           = 2412
Begin VB.OptionButton Option4
    Caption           = "Desactivado"
    Height           = 252
    Left            = 240
    TabIndex        = 30
    Top             = 480
    Width           = 1932
End
Begin VB.OptionButton Option3
    Caption           = "Activado"
    Height           = 192
    Left            = 240
    TabIndex        = 29
    Top             = 240
    Width           = 1572
End
Begin VB.TextBox Text11
    Appearance       = 0 'Flat
    BackColor        = &H00C9C0C6&
    BorderStyle      = 0 'None
    Height           = 288
    Left            = 720
    TabIndex        = 28
    Top             = 0
    Width           = 132
End
End
Begin VB.TextBox Text1
    Height           = 288
    Left            = 1920
    TabIndex        = 26
    Text            = "128"
```

```

    Top           = 240
    Width        = 1692
End
Begin VB.CommandButton Command2
    Caption       = "Borrar Datos Recibidos"
    Height       = 612
    Left        = 4080
    TabIndex    = 25
    Top         = 6120
    Width       = 2412
End
Begin VB.CommandButton Command1
    Caption       = "Transmitir"
    Height       = 612
    Left        = 240
    TabIndex    = 11
    Top         = 6120
    Width       = 3372
End
Begin VB.ListBox List1
    Height       = 4848
    ItemData    = "PUERTO01.frx":0000
    Left        = 4080
    List        = "PUERTO01.frx":0002
    TabIndex    = 23
    TabStop     = 0 'False
    Top         = 1080
    Width       = 2412
End
Begin VB.Frame Frame1
    Caption      = "Formato"
    Height      = 852
    Left       = 240
    TabIndex   = 22
    Top       = 5040
    Width     = 3372
    Begin VB.OptionButton Option2
        Caption      = "Hexagesimal"
        Height      = 252
        Left       = 1680
        TabIndex   = 10
        Top       = 360
        Width     = 1452
    End
    Begin VB.OptionButton Option1
        Caption      = "Decimal"
        Height      = 192
        Left       = 240
        TabIndex   = 9
        Top       = 360
        Value      = -1 'True
        Width     = 1212
    End
End
End
Begin VB.TextBox Text10
    Height      = 288
    Left       = 1920

```

```

    TabIndex      = 8
    Text          = "128"
    Top           = 4560
    Width        = 1692
End
Begin VB.TextBox Text9
    Height        = 288
    Left          = 1920
    TabIndex      = 7
    Text          = "0"
    Top           = 4080
    Width        = 1692
End
Begin VB.TextBox Text8
    Height        = 288
    Left          = 1920
    TabIndex      = 6
    Text          = "0"
    Top           = 3600
    Width        = 1692
End
Begin VB.TextBox Text7
    Height        = 288
    Left          = 1920
    TabIndex      = 5
    Text          = "0"
    Top           = 3120
    Width        = 1692
End
Begin VB.TextBox Text6
    Height        = 288
    Left          = 1920
    TabIndex      = 4
    Text          = "0"
    Top           = 2640
    Width        = 1692
End
Begin VB.TextBox Text5
    Height        = 288
    Left          = 1920
    TabIndex      = 3
    Text          = "0"
    Top           = 2160
    Width        = 1692
End
Begin VB.TextBox Text4
    Height        = 288
    Left          = 1920
    TabIndex      = 2
    Text          = "0"
    Top           = 1680
    Width        = 1692
End
Begin VB.TextBox Text3
    Height        = 288
    Left          = 1920
    TabIndex      = 1

```

```

    Text           =    "0"
    Top            =    1200
    Width          =    1692
End
Begin VB.TextBox Text2
    Height         =    288
    Left          =    1920
    TabIndex      =    6
    Text          =    "0"
    Top           =    720
    Width         =    1692
End
Begin MSCommLib.MSComm MSComm1
    Left          =    3720
    Top           =    5760
    _version     =    65536
    _extentx     =    677
    _extenty     =    677
    _stockprops  =    0
    cdtimeout    =    0
    commport     =    1
    ctsttimeout  =    0
    dsrtimeout   =    0
    dtrenable    =    0    'False
    handshaking  =    0
    inbuffersize =    1024
    inputlen     =    0
    interval     =    1000
    nulldiscard  =    0    'False
    outbuffersize =    512
    parityreplace =    "?"
    rthreshold   =    0
    rtsenable    =    0    'False
    settings     =    "9600,n,8,1"
    sthreshold   =    0
End
Begin VB.Label Label12
    Alignment     =    1    'Right Justify
    Caption      =    "ComX: "
    Height       =    252
    Left        =    3720
    TabIndex    =    24
    Top         =    3720
    Width       =    2052
End
Begin VB.Label Label10
    Alignment     =    1    'Right Justify
    Caption      =    "Bandera Final:"
    Height       =    252
    Left        =    240
    TabIndex    =    21
    Top         =    4560
    Width       =    1572
End
Begin VB.Label Label19
    Alignment     =    1    'Right Justify
    Caption      =    "Dato 4:"

```

```

    Height      = 252
    Left        = 240
    TabIndex    = 20
    Top         = 4080
    Width       = 1572
End
Begin VB.Label Label8
    Alignment   = 1 'Right Justify
    Caption     = "Dato 3:"
    Height      = 252
    Left        = 240
    TabIndex    = 19
    Top         = 3600
    Width       = 1572
End
Begin VB.Label Label7
    Alignment   = 1 'Right Justify
    Caption     = "Dato 2:"
    Height      = 252
    Left        = 240
    TabIndex    = 18
    Top         = 3120
    Width       = 1572
End
Begin VB.Label Label6
    Alignment   = 1 'Right Justify
    Caption     = "Dato 1:"
    Height      = 252
    Left        = 240
    TabIndex    = 17
    Top         = 2640
    Width       = 1572
End
Begin VB.Label Label5
    Alignment   = 1 'Right Justify
    Caption     = "Comando 2:"
    Height      = 252
    Left        = 240
    TabIndex    = 16
    Top         = 2160
    Width       = 1572
End
Begin VB.Label Label4
    Alignment   = 1 'Right Justify
    Caption     = "Comando 1:"
    Height      = 252
    Left        = 240
    TabIndex    = 15
    Top         = 1680
    Width       = 1572
End
Begin VB.Label Label3
    Alignment   = 1 'Right Justify
    Caption     = "Dirección Origen:"
    Height      = 252
    Left        = 240
    TabIndex    = 14

```

```

        Top           = 1200
        Width        = 1572
    End
    Begin VB.Label Label2
        Alignment     = 1 'Right Justify
        Caption       = "Dirección Destino:"
        Height        = 252
        Left          = 240
        TabIndex      = 13
        Top           = 720
        Width         = 1572
    End
    Begin VB.Label Label1
        Alignment     = 1 'Right Justify
        Caption       = "Bandera Inicio:"
        Height        = 252
        Left          = 240
        TabIndex      = 12
        Top           = 240
        Width         = 1572
    End
End
Attribute VB_Name = "Form1"
Attribute VB_Creatable = False
Attribute VB_Exposed = False
Dim puerto As Integer
Dim comx As Integer

Sub Inicio_puerto()
    MSComm1.CommPort = puerto
    ' 9600 baud, no parity, 8 data, and 1 stop bit.
    MSComm1.Settings = "9600,N,8,1"
    ' Tell the control to read entire buffer when Input is used.
    MSComm1.InputLen = 1
    ' Open the port.
    MSComm1.PortOpen = True
End Sub

Private Sub Command1_Click()
    If Not comx Then
        Option3.Value = True
    End If

    If Option1 Then
        MSComm1.Output = Chr(CByte(Text1))
        MSComm1.Output = Chr(CByte(Text2))
        MSComm1.Output = Chr(CByte(Text3))
        MSComm1.Output = Chr(CByte(Text4))
        MSComm1.Output = Chr(CByte(Text5))
        MSComm1.Output = Chr(CByte(Text6))
        MSComm1.Output = Chr(CByte(Text7))
        MSComm1.Output = Chr(CByte(Text8))
        MSComm1.Output = Chr(CByte(Text9))
    End If
End Sub

```

```

        MSComm1.Output = Chr(CByte(Text10))
    Else
        MSComm1.Output = Chr(CByte("&H" + Text1))
        MSComm1.Output = Chr(CByte("&H" + Text2))
        MSComm1.Output = Chr(CByte("&H" + Text3))
        MSComm1.Output = Chr(CByte("&H" + Text4))
        MSComm1.Output = Chr(CByte("&H" + Text5))
        MSComm1.Output = Chr(CByte("&H" + Text6))
        MSComm1.Output = Chr(CByte("&H" + Text7))
        MSComm1.Output = Chr(CByte("&H" + Text8))
        MSComm1.Output = Chr(CByte("&H" + Text9))
        MSComm1.Output = Chr(CByte("&H" + Text10))
    End If
End Sub

```

```

Private Sub Command2_Click()
    List1.Clear
End Sub

```

```

Private Sub Option1_Click()
    Text1.Text = CStr(CByte("&H" + Text1))
    Text2.Text = CStr(CByte("&H" + Text2))
    Text3.Text = CStr(CByte("&H" + Text3))
    Text4.Text = CStr(CByte("&H" + Text4))
    Text5.Text = CStr(CByte("&H" + Text5))
    Text6.Text = CStr(CByte("&H" + Text6))
    Text7.Text = CStr(CByte("&H" + Text7))
    Text8.Text = CStr(CByte("&H" + Text8))
    Text9.Text = CStr(CByte("&H" + Text9))
    Text10.Text = CStr(CByte("&H" + Text10))
End Sub

```

```

Private Sub Option2_Click()
    Text1.Text = CStr(Hex(CByte(Text1)))
    Text2.Text = CStr(Hex(CByte(Text2)))
    Text3.Text = CStr(Hex(CByte(Text3)))
    Text4.Text = CStr(Hex(CByte(Text4)))
    Text5.Text = CStr(Hex(CByte(Text5)))
    Text6.Text = CStr(Hex(CByte(Text6)))
    Text7.Text = CStr(Hex(CByte(Text7)))
    Text8.Text = CStr(Hex(CByte(Text8)))
    Text9.Text = CStr(Hex(CByte(Text9)))
    Text10.Text = CStr(Hex(CByte(Text10)))

    End Sub

```

```

Private Sub Option3_Click()
    On Error GoTo ErrorHandler
    puerto = 1
    Call Inicio_puerto
    comx = 1

```

Principal:

```
If MSComm1.InBufferCount Then
    MSComm1.InputLen = 1
    List1.AddItem CStr(CByte(Asc(MSComm1.Input)))
End If
DoEvents
If comx Then
    GoTo Principal
End If
Exit Sub
```

```
ErrorHandler: ' Rutina que determina el primer puerto libre.
    puerto = puerto + 1
    Call Inicio_puerto
    Text11.Text = CStr(puerto)
```

```
End Sub
```

```
Private Sub Option4_Click()
    comx = 0
    MSComm1.PortOpen = False
End Sub
```

Apendice B: CPU auxiliar

```
1 NETWORK 1
2 LD SM0.1
3 CALL 0
4
5 NETWORK 2 // Activa el freeport
6 LD SM0.7
7 = SM30.0
8
9 NETWORK 3 // Ejecuta rutina para revisar dirección de la transmisión
10 LD M0.0
11 CALL 1
12
13 NETWORK 4 // Ejecuta rutina para revisar la información recibida
14 LD M0.1
15 CALL 2
16
17 NETWORK 5 // Llama subrutina inicializar eje 1
18 LD M10.0
19 CALL 3
20
21 NETWORK 6 // Llama subrutina ir a esta posición eje 1
22 LD M11.0
23 CALL 4
24
25 NETWORK 7 // Llama subrutina inicializar eje 2
26 LD M20.0
27 CALL 5
28
29 NETWORK 8 // Llama subrutina ir a esta posición eje 2
30 LD M21.0
31 CALL 6
32
33 NETWORK 9 // Fin Programa principal
34 MEND
35
36 NETWORK 10 // Subrutina inicialización
37 SBR 0
38
39 NETWORK 11 // inicialización Recepción
40 LD SM0.0
41 MOVB 2, VB100 // Define la dirección del CPU
42 MOVB 128, VB101 // Define la bandera inicio final
43 MOVB 0, VB110 // Inicializa buffer de entrada
44 MOVB 0, VB111
45 MOVB 0, VB112
46 MOVB 0, VB113
47 MOVB 0, VB114
48 MOVB 0, VB115
49 MOVB 0, VB116
50 MOVB 0, VB117
51 MOVB 0, VB118
52 MOVB 0, VB119
53
54 NETWORK 12 // inicialización transmisión
55 LD SM0.0
56 MOVB 9, SMB30 // Define el modo del puerto
57 ATCH 0, 8
58 MOVB &VB130, VD102 // Activa puntero memoria de Tx
// Activa interrupción de Rx
59 MOVB 10, VB130 // Inicializa memoria de Tx
60 MOVB 128, VB131
61 MOVB 0, VB132
62 MOVB VB100, VB133
63 MOVB 0, VB134
64 MOVB 0, VB135
65 MOVB 0, VB136
```

Apendice B: CPU auxiliar

```
66     MOVB    0, VB137
67     MOVB    0, VB138
68     MOVB    0, VB139
69     MOVB    128, VB140
70
71     NETWORK 13      // Inicializa posición y velocidad 1
72     LD      SM0.0
73     MOVD    +0, VD0          // Inicializar variable 1 posición actual
74     MOVD    +0, VD4          // Inicializar variable 1 posición pasada
75     MOVD    +0, VD8          // Inicializar variable 1 velocidad
76     MOVB    16#F8, SMB47     // Definición del contador 1 x4 con reset
77     HDEF    1, 10
78     MOVD    +0, SMD48
79     MOVD    +0, SMD50
90     HSC     1
81     MOVB    100, SMB34      // definición del la velocidad de muestreo
82     ATCH    1, 10
83
84     NETWORK 14      // Inicializa posición y velocidad 2
85     LD      SM0.0
86     MOVD    +0, VB20        // Inicializar variable 2 posición actual
87     MOVD    +0, VD24        // Inicializar variable 2 posición pasada
88     MOVD    +0, VD28        // Inicializar variable 2 velocidad
89     MOVB    16#F8, SMB57     // definición del contador 2 x4 con reset
90     HDEF    2, 10
91     MOVD    +0, SMD58
92     MOVD    +0, SMD62
93     HSC     2
94     MOVB    100, SMB35      // definición del la velocidad de muestreo
95     ATCH    2, 11
96     ENI
97
98     NETWORK 15      // inicialización variables
99     MOVD    +0, VD40        // Variable division PWM1
100    MOVD    +0, VD44        // Variable resta posición 1
101    MOVB    0, VB150        // Dator RX eje 1
102    MOVB    0, VB151        // Dator RX eje 1
103    MOVB    0, VB152        // Dator RX eje 1
104    MOVD    +0, VD50        // Variable division PWM2
105    MOVD    +0, VD54        // Variable resta posicon 2
106    MOVB    0, VB160        // Dator RX eje 2
107    MOVB    0, VB161        // Dator RX eje 2
108    MOVB    0, VB162        // Dator RX eje 2
109
110    NETWORK 16      // Retorno subrutina inicio
111    RET
112
113    NETWORK 17      // Subrutina para revisar dirección del mensaje recibido
114    SBR     1
115
116    NETWORK 18      // Borra la bandera para entrar a esta subrutina
117    LD      SM0.0
118    R       M0.0, 1
119
120    NETWORK 19      // Compara las banderas y las direcciones para guardar
información
121    LDB=    VB110, VB101
122    AB=     VB119, VB101
123    AB=     VB111, VB100
124    MOVB    VB112, VB122
125    MOVB    VB113, VB123
126    MOVB    VB114, VB124
127    MOVB    VB115, VB125
128    MOVB    VB116, VB126
129    MOVB    VB117, VB127
130    MOVB    VB118, VB128
```

Apendice B: CPU auxiliar

```

131 S M0.1, 1 // Activa bandera de mensaje recibido dirección
correcta
132
133 NETWORK 20
134 RET // Retorno interrupción
135
136 NETWORK 21 // Subrutina para descifrar mensaje recibido
137 SBR 2
138
139 NETWORK 22 // Borra la bandera de entrada a esta subrutina
140 LD SM0.0
141 R M0.1, 1
142
143 NETWORK 23 // Compara el comando con "10" que significa inicializar 1
144 LDB= VB123, 10
145 S M10.0, 1 // Borra la bandera de motores encendidos
146 S M10.1, 1
147
148 NETWORK 24 // Compara el comando con "11" que significa ir a 1
149 LDB= VB123, 11
150 S M11.0, 1 // Borra la bandera de motores encendidos
151 S M11.1, 1
152
153 NETWORK 25 // Compara el comando con "20" que significa inicializar 2
154 LDB= VB123, 20
155 S M20.0, 1 // Borra la bandera de motores encendidos
156 S M20.1, 1
157
158 NETWORK 26 // Compara el comando con "21" que significa ir a 2
159 LDB= VB123, 21
160 S M21.0, 1 // Borra la bandera de motores encendidos
161 S M21.1, 1
162
163 NETWORK 27
164 RET // Fin de la subrutina
165
166 NETWORK 28 // Rutina inicialización dirección 1
167 SBR 3
168
169 NETWORK 29
170 LD M10.1
171 MOVB 16#C3, SMB67 // Activa el PWM1
172 MOVW +5000, SMW68
173 MOVD +0, VD40 // Modifica el valor enviado en Com2 como * del
PWM
174 MOVB VB124, VB43
175 MUL +50, VD40
176 MOVW VW42, SMW70 // Guarda el valor del ancho del pulso
177 PLS 0 // Activa el PWM
178 MOVB 16#C2, SMB67
179 S Q0.2, 1 // Activa la salida de dirección
180 R M10.1, 1 // Elimina bandera para ejecutar esta rutina de
nuevo
181 MOVB VB122, VB150
182 MOVB VB123, VB151
183 MOVB VB124, VB152
184
185 NETWORK 30
186 LD I1.0 // Verifica la entrada de posición cero
187 MOVB 16#42, SMB67
188 PLS 0
189 R Q0.2, 1 // Borra la salida de dirección
190 R M10.0, 1 // Elimina la bandera para entrar a la subrutina
3
191 MOVB VB150, VB132
192 MOVB VB151, VB134

```

Apéndice B: CPU auxiliar

```
193   MOVB   VB152, VB135
194   MOVD   +0, VD136
195   XMT    *VD102, 0
196
197   NETWORK 31
198   RET
199
200   NETWORK 32           // Rutina ir a posición eje 1
201   SBR    4
202
203   NETWORK 33
204   LD     M11.1
205   LPS                                // Activa el PWM1
206   MOVB   16#C3, SMB67
207   MOVW   +5000, SMW68                // Modifica el valor enviado en Com2 como
   del PWM
208   MOVD   +0, VD40
209   MOVB   VB124, VB43
210   MUL    +50, VD40                    // Guarda el valor del ancho del pulso
211   MOVW   VW42, SMW70                // Activa el PWM
212   PLS    0
213   MOVB   16#C2, SMB67
214   R      Q0.3, 2
215   MOVD   VD125, VD44
216   -D     HC1, VD44                    // Elimina bandera para ejecutar esta
   rutina de nuevo
217   A      SM1.2
218   S      Q0.2, 1
219   LRD
220   AN     Q0.2
221   S      Q0.3, 1
222   LRD
223   R      M11.1, 1
224   LPP
225   MOVD   VD125, VD44
226   MOVB   VB122, VB150
227   MOVB   VB123, VB151
228   MOVB   VB124, VB152
229
230   NETWORK 34
231   LDD<=  HC1, VD44                    // Verifica si llego a la posición
232   A      Q0.2
233   MOVB   16#42, SMB67
234   PLS    0
235   R      Q0.2, 1                      // Borra la salida de dirección
236   R      M11.0, 1                    // Elimina la bandera para entrar a la subrutina
   3
237   MOVB   VB150, VB132
238   MOVB   VB151, VB134
239   MOVB   VB152, VB135
240   MOVD   HC1, VD136
241   XMT    *VD102, 0
242
243   NETWORK 35
244   LDD>=  HC1, VD44                    // Verifica si llego a la posición
245   A      Q0.3
246   MOVB   16#42, SMB67
247   PLS    0
248   R      Q0.3, 1                      // Borra la salida de dirección
249   R      M11.0, 1                    // Elimina la bandera para entrar a la subrutina
   3
250   MOVB   VB150, VB132
251   MOVB   VB151, VB134
252   MOVB   VB152, VB135
253   MOVD   HC1, VD136
254   XMT    *VD102, 0
```

Apendice B: CPU auxiliar

```
255
256 NETWORK 36
257 RET
258
259 NETWORK 37 // Rutina inicialización dirección 2
260 SBR 5
261
262 NETWORK 38
263 LD M20.1
264 MOVB 16#C3, SMB77 // Activa el PWM2
265 MOVW +5000, SMW78
266 MOVD +0, VD50 // Modifica el valor enviado en Com2 como del
PWM
267 MOVB VB124, VB53
268 MUL +50, VD50
269 MOVW VW52, SMW80 // Guarda el valor del ancho del pulso
270 PLS 1 // Activa el PWM
271 MOVB 16#C2, SMB77
272 S Q0.4, 1 // Activa la salida de dirección
273 R M20.1, 1 // Elimina bandera para ejecutar esta rutina de
nuevo
274 MOVB VB122, VB160
275 MOVB VB123, VB161
276 MOVB VB124, VB162
277
278 NETWORK 39
279 LD I1.4 // Verifica la entrada de posición cero
280 MOVB 16#42, SMB77
281 PLS 1
282 R Q0.4, 1 // Borra la salida de dirección
283 R M20.0, 1 // Elimina la bandera para entrar a la subrutina
3
284 MOVB VB160, VB132
285 MOVB VB161, VB134
286 MOVB VB162, VB135
287 MOVD +0, VD136
288 XMT *VB102, 0
289
290 NETWORK 40
291 RET
292
293 NETWORK 41 // Rutina ir a posición eje 2
294 SBR 6
295
296 NETWORK 42
297 LD M21.1
298 LPS // Activa el PWM2
299 MOVB 16#C3, SMB77
300 MOVW +5000, SMW78 // Modifica el valor enviado en Com2 como
del PWM
301 MOVD +0, VD50
302 MOVB VB124, VB53
303 MUL +50, VD50 // Guarda el valor del ancho del pulso
304 MOVW VW52, SMW80 // Activa el PWM
305 PLS 1
306 MOVB 16#C2, SMB77
307 R Q0.5, 2
308 MOVD VD125, VD54
309 -D HC2, VD54 // Elimina bandera para ejecutar ésta
rutina de nuevo
310 A SM1.2
311 S Q0.4, 1
312 LRD
313 AN Q0.4
314 S Q0.5, 1
315 LPP
```

Apendice B: CPU auxiliar

```
316 R      M21.1, 1
317 LPP
318 MOVD   VD125, VD54
319 MOVB   VB122, VB160
320 MOVB   VB123, VB161
321 MOVB   VB124, VB162
322
323 NETWORK 43
324 LDD<=  HC2, VD54           // Verifica si llego a la posición
325 A      Q0.4
326 MOVB   16#42, SMB77
327 PLS    1
328 R      Q0.4, 1           // Borra la salida de dirección
329 R      M21.0, 1         // Elimina la bandera para entrar a la subrutina
    5
330 MOVB   VB160, VB132
331 MOVB   VB161, VB134
332 MOVB   VB162, VB135
333 MOVD   HC2, VD136
334 XMT    *VD102, 0
335
336 NETWORK 44
337 LDD>=  HC2, VD54           // Verifica si llego a la posición
338 A      Q0.5
339 MOVB   16#42, SMB77
340 PLS    1
341 R      Q0.5, 1           // Borra la salida de dirección
342 R      M21.0, 1         // Elimina la bandera para entrar a la subrutina
    5
343 MOVB   VB160, VB132
344 MOVB   VB161, VB134
345 MOVB   VB162, VB135
346 MOVD   HC2, VD136
347 XMT    *VD102, 0
348
349 NETWORK 45
350 RET
351
352 NETWORK 46
353 INT     0           // Interrupción por mensaje recibido
354
355 NETWORK 47           // Lee el dato y corre la información en el buffer de
    entrada
356 LD      SM0.0
357 MOVB   VB111, VB110
358 MOVB   VB112, VB111
359 MOVB   VB113, VB112
360 MOVB   VB114, VB113
361 MOVB   VB115, VB114
362 MOVB   VB116, VB115
363 MOVB   VB117, VB116
364 MOVB   VB118, VB117
365 MOVB   VB119, VB118
366 MOVB   SMB2, VB119
367 S      M0.0, 1           // Activa la bandera para indicar un nuevo byte
    recibido
368
369 NETWORK 48
370 RETI           // Fin de la interrupción
371
372 NETWORK 49           // interrupción velocidad 1
373 INT     1
374
375 NETWORK 50
376 LD      SM0.0
377 MOVD   VD0, VD4           // Mueve el valor anterior a su posición
```

Apendice B: CPU auxiliar

```
378  MOVD  HC1, VD8           // Mueve el valor actual a la variable de
    velocidad
379  MOVD  VD8, VD0           // Mueve el valor actual a su posición
380  -D    VD8, VD4           // Resta el valor actual del anterior y obtiene
    la velocidad
381
382  NETWORK 51
383  RETI
384
385  NETWORK 52           // interrupción velocidad 2
386  INT    2
387
388  NETWORK 53
389  LD     SM0.0
390  MOVD  VD20, VD24         // Mueve el valor anterior a su posición
391  MOVD  HC2, VD28         // Mueve el valor actual a la variable de
    velocidad
392  MOVD  VD28, VE20         // Mueve el valor actual a su posición
393  -D    VD28, VD24         // Resta el valor actual del anterior y
    obtiene la velocidad
394
395  NETWORK 54
396  RETI
```

Apendice C: Programa CPU principal

```
1 NETWORK 1
2 LD SM0.1
3 CALL 0
4
5 NETWORK 2 // Activa el freeport
6 LD SM0.7
7 = SM30.0
8
9 NETWORK 3 // Ejecuta rutina para revisar dirección de la transmisión
10 LD M0.0
11 CALL 1
12
13 NETWORK 4 // Ejecuta rutina para revisar la información recibida
14 LD M0.1
15 CALL 2
16
17 NETWORK 5 // Contador para el retraso en el encendido de motores
18 LD SM0.0
19 LPS
20 A I0.0
21 A M1.0
22 TON T101, +1S // Retraso de 1 segundo al momento de
recibir la señal
23 LPP
24 A T101
25 = Q0.0
26
27 NETWORK 6 // Fin Programa principal
28 MEND
29
30 NETWORK 7 // Subrutina inicialización
31 SBR 0
32
33 NETWORK 8
34 LD SM0.0
35 MOVB 1, VB100 // Define la dirección del CPU
36 MOVB 128, VB101 // Define la bandera inicio final
37 MOVB 0, VB110 // Inicializa buffer de entrada
38 MOVB 0, VB111
39 MOVB 0, VB112
40 MOVB 0, VB113
41 MOVB 0, VB114
42 MOVB 0, VB115
43 MOVB 0, VB116
44 MOVB 0, VB117
45 MOVB 0, VB118
46 MOVB 0, VB119
47 MOVB 9, SMB30 // Define el modo del puerto
48 ATCH 0, 8 // Activa interrupción de Rx
49 ENI
50 R M1.0, 1 // Activa bandera de salida motores
51
52 NETWORK 9 // Retorno subrutina inicio
53 RET
54
55 NETWORK 10 // Subrutina para revisar dirección del mensaje recibido
56 SBR 1
57
58 NETWORK 11 // Borra la bandera para entrar a esta subrutina
59 LD SM0.0
60 R M0.0, 1
61
62 NETWORK 12 // Compara las banderas y las direcciones para guardar
información
63 LDB= VB110, VB101
64 AB= VB119, VB101
```

Apendice C: Programa CPU principal

```
65 AB= VB111, VB100
66 MOVB VB112, VB122
67 MOVB VB113, VB123
68 MOVB VB114, VB124
69 MOVB VB115, VB125
70 MOVB VB116, VB126
71 MOVB VB117, VB127
72 MOVB VB118, VB128
73 S M0.1, 1 // Aciva bandera de mensaje recibido dirección
correcta
74
75 NETWORK 13
76 RET // Retorno interrupción
77
78 NETWORK 14 // Subrutina para descifrar mensaje recibido
79 SBR 2
80
81 NETWORK 15 // Borra la bandera de entrada a esta subrutina
82 LD SM0.0
83 R M0.1, 1
84
85 NETWORK 16 // Compara el comando con "0" que significa parar
86 LDB= VB123, 16#0
87 R M1.0, 1 // Borra la bandera de motores encendidos
88 CRET // Retorno condicional
89
90 NETWORK 17 // Compara el comando con "1" que significa encender
91 LDB= VB123, 16#1
92 S M1.0, 1 // Activa la bandera motores encendidos
93
94 NETWORK 18
95 RET // Fin de la subrutina
96
97 NETWORK 19
98 INT 0 // Interrupción por mensaje recibido
99
100 NETWORK 20 // Lee el dato y corre la información en el buffer de
entrada
101 LD SM0.0
102 MOVB VB111, VB110
103 MOVB VB112, VB111
104 MOVB VB113, VB112
105 MOVB VB114, VB113
106 MOVB VB115, VB114
107 MOVB VB116, VB115
108 MOVB VB117, VB116
109 MOVB VB118, VB117
110 MOVB VB119, VB118
111 MOVB SMB2, VB119
112 S M0.0, 1 // Activa la bandera para indicar un nuevo byte
recibido
113
114 NETWORK 21
115 RETI // Fin de la interrupción
```

Apéndice D

Programación de CPS S7-214 "Transmisión y Recepción de Datos"

```
NETWORK 1
LD    SM0.1
CALL  0

NETWORK 2 // Activa el freeport
LD    SM0.7
=     SM30.0

NETWORK 3
LD    M0.0
CALL  1

NETWORK 4
LD    M0.1
CALL  2

NETWORK 5
MEND

NETWORK 6
SBR   0

NETWORK 7
LD    SM0.0
MOVB  1, VB100 // Define la direccion del CPU
MOVB  128, VB101 // Define la bandera inicio final
MOVB  0, VB110 // Inicializa buffer de entrada
MOVB  0, VB111
MOVB  0, VB112
MOVB  0, VB113
MOVB  0, VB114
MOVB  0, VB115
MOVB  0, VB116
MOVB  0, VB117
MOVB  0, VB118
MOVB  0, VB119
MOVB  9, SMB30 // Define el modo del puerto
ATCH  1, 8
ENI

NETWORK 8
RET

NETWORK 9
SBR   1

NETWORK 10
LD    SM0.0
R     M0.0, 1

NETWORK 11
LDB=  VB110, VB101
AB=   VB119, VB101
```

```
AB=    VB111, VB100
MOVB   VB112, VB122
MOVB   VB113, VB123
MOVB   VB114, VB124
MOVB   VB115, VB125
MOVB   VB116, VB126
MOVB   VB117, VB127
MOVB   VB118, VB128
S      M0.1, 1
```

```
NETWORK 12
RET
```

```
NETWORK 13
SBR    2
```

```
NETWORK 14
LD     SM0.0
R      M0.1, 1
```

```
NETWORK 15
LDB=   VB123, 16#1
A      Q0.2
R      Q0.2, 1
R      Q0.0, 1
CRET
```

```
NETWORK 16
LDB=   VB123, 16#1
AN     Q0.2
R      Q0.3, 1
S      Q0.2, 1
S      Q0.0, 1
```

```
NETWORK 17
LDB=   VB123, 16#2
A      Q0.3
R      Q0.3, 1
R      Q0.0, 1
CRET
```

```
NETWORK 18
LDB=   VB123, 16#2
AN     Q0.3
R      Q0.2, 1
S      Q0.3, 1
S      Q0.0, 1
```

```
NETWORK 19
LDB=   VB123, 16#3
A      Q0.4
R      Q0.4, 1
R      Q0.1, 1
CRET
```

```
NETWORK 20
LDB=   VB123, 16#3
```

```
AN    Q0.4
R     Q0.5, 1
S     Q0.4, 1
S     Q0.1, 1
```

```
NETWORK 21
LDB=  VB123, 16#4
A     Q0.5
R     Q0.5, 1
R     Q0.1, 1
CRET
```

```
NETWORK 22
LDR=  VB123, 16#4
AN    Q0.5
R     Q0.4, 1
S     Q0.5, 1
S     Q0.1, 1
```

```
NETWORK 23
RET
```

```
NETWORK 24
INT   1
```

```
NETWORK 25
LD    SM0.0
MOVB  VB111, VB110
MOVB  VB112, VB111
MOVB  VB113, VB112
MOVB  VB114, VB113
MOVB  VB115, VB114
MOVB  VB116, VB115
MOVB  VB117, VB116
MOVB  VB118, VB117
MOVB  VB119, VB118
MOVB  SMB2, VB119
S     M0.0, 1
```

```
NETWORK 26
RETI
```

Apéndice E

Programación de CPS S7-214 "Lectura de Censores"

```
NETWORK 1 // Llamada subrutina de inicialización
LD SM0.1
CALL 0

NETWORK 2 // Fin del programa principal
MEND

NETWORK 3 // Subrutina de inicialización
SBR 0

NETWORK 4 // Inicializa posición y velocidad 1
LD SM0.0
MOVD +0, VD0 // Inicializar variable 1 posición actual
MOVD +0, VD4 // Inicializar variable 1 posición pasada
MOVD +0, VD8 // Inicializar variable 1 velocidad
MOVD &VB1000, VD12 // Inicializa puntero de memoria
MOVD &VB2000, VD16 // Define ancho de memoria
MOVB 16#F8, SMB47 // Definición del contador 1 x4 con
reset
HDEF 1, 10
MOVD +0, SMD43
MOVD +0, SMD52
HSC 1
MOVB 100, SMB34 // Definición del la velocidad de muestreo
ATCH 0, 10

NETWORK 5 // Inicializa posición y velocidad 2
LD SM0.0
MOVD +0, VD20 // Inicializar variable 2 posición actual
MOVD +0, VD24 // Inicializar variable 2 posición pasada
MOVD +0, VD28 // Inicializar variable 2 velocidad
MOVD &VB2000, VD32 // Inicializa puntero de memoria
MOVD &VB3000, VD36 // Define ancho de memoria
MOVB 16#F8, SMB57 // Definición del contador 2 x4 con
reset
HDEF 2, 10
MOVD +0, SMD58
MOVD +0, SMD62
HSC 2
MOVB 100, SMB35 // Definición del la velocidad de muestreo
ATCH 1, 11
ENI

NETWORK 6 // Fin subrutina inicialización
RET

NETWORK 7 // Interrupción velocidad 1
INT 0

NETWORK 8
LD SM0.0
MOVD VD0, VD4 // Mueve el valor anterior a su posición
```

```
MOVD  HC1, VD8           // Mueve el valor actual a la variable de
velocidad
MOVD  VD8, VD0           // Mueve el valor actual a su posición
-D    VD8, VD4           // Resta el valor actual del anterior y
obtiene la velocidad
```

```
NETWORK 9
RETI
```

```
NETWORK 10 // Interrupción velocidad 2
INT      1
```

```
NETWORK 11
LD     SM0.0
MOVD  VD20, VD24         // Mueve el valor anterior a su posición
MOVD  HC2, VD28         // Mueve el valor actual a la variable de
velocidad
MOVD  VD28, VD20         // Mueve el valor actual a su posición
-D    VD28, VD24         // Resta el valor actual del anterior y
obtiene la velocidad
```

```
NETWORK 12
RETI
```

Apéndice F

Programación de PIC 17C44

```
*****
**
; ROBOT05.ASM
;
; Funciones del Programa
;
; - Se controlarán los puertos PWM del PIC17C44 para que trabajen en dos
; señales de salida analógicas que controlen cada una de las direcciones
; del motor.
;   Las salidas de los PWM son PWM1: RB2 y PWM2: RB3. La salida del PWM
; tiene una frecuencia 127 veces menor a la entrada RB4.
;
; - Leer los dos pulsos de entrada del sensor de posición y en función
del
; desface determinar la dirección de desplazamiento y modificar el file
de
; posición (RB7 y RB6).
;
; - Leer el pulso de inicio de posición y borra la posición actual (RB5).
;
; - Mediante la utilización de un contador que es independiente de
hardware
; se determina la velocidad actual del dispositivo, originalmente
; se realizan 10 muestreos por segundo, pero este dato puede ser variado
a
; voluntad. Un factor importante es no tener velocidades mayores de 6
bits
; (formato de complementos 2's).
;
; -La operación de un bus RS485 a través de un puerto EUSART. El
funcionamiento
; del bus estará dado de la siguiente forma:
;   * Todos los dispositivos pic17c44 funcionarán como unidades
esclavas
;   y con una dirección cada una.
;   * La trama de transición estará compuesta por 10 bytes, de la
siguiente
;   forma:
;       1. Byte Inicio de TX: 11110000
;       2. Byte de Destino.
;       3. Byte de Fuente.
;       4. Byte de Comando I.
;       5. Byte de Comando II.
;       6. Byte de Dato I.
;       7. Byte de Dato II.
;       8. Byte de Dato III.
;       9. Byte de Dato IV.
;       10. Byte Fin de TX: 10000000
;   * Todas las transmisiones se realizarán mediante un procedimiento
de Eco en el
;   cual si el PIC recibe un mensaje destinado para él entonces
devuelve el mismo
;   mensaje al origen para revisar errores.
```

```

;      * La Transmisión será asincrona a 9,600 bps con un start bit y un
stop bit, no se utilizará
;      bit de paridad.
;
; -Adicionalmente a través de este puerto se pueden enviar los siguientes
comandos:
;
;*****
**

```

```

LIST P=I7C44, R=DEC
INCLUDE C:\JAVIER\MPLAB\I7C44.INC      ; I7C44 include file

```

```

;-----
--
; VARIABLES
;-----
--

```

```

POSERR      EQU    H'50'
TEMP01      EQU    H'51'
TEMP02      EQU    H'52'
TEMP03      EQU    H'53'
VELERR      EQU    H'54'
POSA01      EQU    H'55'
POSA02      EQU    H'56'
POSA03      EQU    H'57'
TEMP04      EQU    H'58'
DIRR        EQU    H'59'
RXIN01      EQU    H'5A'
RXIN02      EQU    H'5B'
RXIN03      EQU    H'5C'
RXIN04      EQU    H'5D'
RXIN05      EQU    H'5E'
RXIN06      EQU    H'5F'
RXIN07      EQU    H'6A'
RXIN08      EQU    H'6B'
RXIN09      EQU    H'6C'
RXIN10      EQU    H'6D'
RXERROR     EQU    H'6E'
TEMP05      EQU    H'6F'

```

```

;-----
--
; Variables de Operación
;-----
--

```

```

RXDR01      EQU    H'80'
RXCM01      EQU    H'81'
RXCM02      EQU    H'82'
RXDT01      EQU    H'83'
RXDT02      EQU    H'84'
RXDT03      EQU    H'85'
RXDT04      EQU    H'86'
TXDR01      EQU    H'87'
TXCM01      EQU    H'88'

```

```

TXCM02          EQU   H'89'
TXDT01          EQU   H'8A'
TXDT02          EQU   H'8B'
TXDT03          EQU   H'8C'
TXDT04          EQU   H'8D'
RXECO           EQU   H'8E'
PWMOUT01        EQU   H'8F'
PWMOUT02        EQU   H'90'
POS01           EQU   H'91'
POS02           EQU   H'92'
POS03           EQU   H'93'
VEL01           EQU   H'94'
VEL02           EQU   H'95'
VEL03           EQU   H'96'
VELCONF01       EQU   H'97'
VELCONF02       EQU   H'98'

```

```

;-----
; CODIGO DEL PROGRAMA (Direcciones especiales)
;-----

```

```

          ORG     0
          GOTO    StartUp
          ORG     H'8'
          GOTO    INTERRUPCIONES
          ORG     H'10'
          GOTO    INTERRUPCIONES
          ORG     H'18'
          GOTO    INTERRUPCIONES
          ORG     H'20'
          GOTO    INTERRUPCIONES

```

```

;-----
; PROCESO DE INICIALIZACION
;-----

```

```

          ORG     H'100'

```

StartUp

```

;Determinación de la dirección del PIC en el BUS RS485

```

```

          MOVLW 1
          MOVWF DIRR

```

```

;Inicialización del puerto B (Entrada de los pulsos y el reloj):

```

```

          MOVLB 0
          CLRF  PORTB
          MOVLW B'11100000' ;Inicialización con los Bytes 6 y 7 como
entradas

```

```

MOVWF DDRB

MOVLB 1
BSF  PIE,7      ; Activación de la interrupción de la
entrada.

CLRF  TEMP03    ; Borra la variable donde contiene el
último estado del puerto

;Inicialización del PWM

MOVLB 2
MOVLW 123      ;Selecciona la frecuencia del PWM
MOVWF PR1
CLRF  TMR1     ;Limpia el contador

MOVLB 3
CLRF  PW1DCL   ;Borra todos los registros de salida
del PWM

CLRF  PW2DCL
CLRF  PW1DCH
CLRF  PW2DCH

BSF  TCON1,0   ;Selección del contador con reloj
externo

BSF  TCON2,0   ;Activa el contador y los PWM
BSF  TCON2,4
BSF  TCON2,5

MOVLB 1       ; Activación de la interrupción del
contador.

BSF  PIE,4

; Activación del contador 3 y las interrupciones asociadas

CLRF  TEMP04   ; Limpieza del contador auxiliar
CLRF  POSA01
CLRF  POSA02
CLRF  POSA03

MOVLW H'F7'   ; La configuración inicial hace muestreos 10
veces por segundo
MOVWF VELCONF01
MOVLW H'05'
MOVWF VELCONF02

MOVLB 1
BSF  PIE,6    ; Activación de la interrupción de la
entrada.

CALL  CONF_VEL

;Inicializacion del puerto auxiliar:

```

```

MOVLB 1          ; Borra los datos del puerto
CLRF PORTC
MOVLW 0          ; Selecciona el puerto como salida
MOVWF DDRC
datos BCF PORTC,1          ; Activa una configuración de RX de
BCF PORTC,0

```

;Inicialización del USART

```

MOVLB 0
MOVLW 25          ; Selección de velocidad a 9600 bps
MOVWF SPBRG
CLRF TXSTA          ; Inicialización del puerto
MOVLW B'10010000' ; Inicialización del puerto
MOVWF RCSTA

MOVLB 1
BSF PIE,0          ; Activación de la interrupción de RX.

```

; Inicialización de las variables de operación

```

CLRF RXERROR
CLRF RXIN01
CLRF RXIN02
CLRF RXIN03
CLRF RXIN04
CLRF RXIN05
CLRF RXIN06
CLRF RXIN07
CLRF RXIN08
CLRF RXIN09
CLRF RXIN10
CLRF RXDR01
CLRF RXCM01
CLRF RXCM02
CLRF RXDT01
CLRF RXDT02
CLRF RXDT03
CLRF RXDT04
CLRF PWMOUT01
CLRF PWMOUT02
CLRF POS01
CLRF POS02
CLRF POS03
CLRF VEL01
CLRF VEL02
CLRF VEL03
BSF RXECO,0

```

; Activación de las interrupciones

```

pero bueno) BCF CPUSTA,4          ; Activación de las interrupciones (Repetido
BSF INTSTA,3          ; Activación de las interrupciones de
perifericos (Repetido)

```

```
        MOVLB 1
        CLRF PIR          ; Limpia el registro de cualquier
interupción durante la configuración
```

```
        GOTO Start
```

```
-----
;
; PROCEDIMIENTO DE INTERRUPCIONES
;
-----
```

INTERRUPCIONES

```
        MOVFP BSR,TEMP05
        MOVLB 1
        BTFSC PIR,6
        CALL VELOCIDAD
        MOVLB 1
        BTFSC PIR,7
        CALL POSICION
        MOVLB 1
        BTFSC PIR,4
        CALL PWM_VALORES
        MOVLB 1
        BTFSC PIR,0
        CALL RECEPCION
        MOVFP TEMP05,BSR
        RETFIE
```

```
-----
;
; RUTINAS Y PROCEDIMIENTOS
;
-----
```

```
-----
; Los datos del PWM son actualizados con los últimos datos que estén
contenidos en el PWM
;
-----
```

PWM_VALORES

```
        MOVLB 1
        BCF PIR,4

        MOVLB 3

        MOVFP PWMOUT01,WREG
        MOVWF PW1DCH

        MOVFP PWMOUT02,WREG
        MOVWF PW2DCH

        RETURN
```

```

;-----
;
; El dato de la posición se aumenta o se disminuye en función de las
; variables
; de entrada
;-----
;

```

```

POS_FIN
puerto      MOVFP TEMP01,WREG ; Se almacena el TEMP03 el último dato del
              MOVWF TEMP03
              RETURN

```

```

POS_INC
desplazamiento subiguiente
              MOVLW 1          ;Se incrementa pos01 y luego se suma el
              ADDWF POS01,1
              MOVLW 0
              ADDWFC POS02,1
              ADDWFC POS03,1

operando     BCF POSERR,1      ; Limpieza del bit de error
              BTFSC TEMP02,7   ; Se asume la resta y se revisa el signo del
              GOTO POS_FIN
              BTFSS POS03,7     ; Se revisa la segunda condición
              GOTO POS_FIN

              BSF POSERR,1      ; Si hay un error marca el bit 0 de POSERR
              MOVLW H'FF'
              MOVWF POS01
              MOVWF POS02
              MOVLW H'7F'
              MOVWF POS03
              GOTO POS_FIN

```

```

POS_DEC
desplazamiento del carry
              MOVLW 1
              SUBWF POS01,1     ;decrementa la posición y realiza el
              MOVLW 0
              SUBWFB POS02,1
              SUBWFB POS03,1

operando     BCF POSERR,0      ; Limpieza del bit de error
              BTFSS TEMP02,7   ; Se asume la resta y se revisa el signo del
              GOTO POS_FIN
              BTFSC POS03,7     ; Se revisa la segunda condición
              GOTO POS_FIN

              BSF POSERR,0      ; Si hay un error marca el bit 0 de POSERR
              CLRF POS01
              CLRF POS02
              MOVLW H'80'
              MOVWF POS03

```

```

        GOTO POS_FIN

POS_BORRAR

POSICION
        MOVLB 1          ; Limpieza del indicador de interrupción
        BCF  PIR,7

        MOVFP POS03,WREG ; Lectura de datos para un posterior chequeo
de demasia aritmetica
        MOVWF TEMP02          ; TEMP02 = dato del POS03 antes del
incremento/decremento

        MOVLB 0
        MOVFP PORTB,WREG ; Recepción de datos y lectura de los dos
pines en Temp01
        ANDLW B'11100000' ; TEMP01 = estado actual del puerto
        MOVWF TEMP01

        MOVFP TEMP01,WREG ; Revisión de entrada de un pulso en RB5,
por lo que se borran los datos
        BTFSC TEMP01,5
        CALL POS_RESET

        MOVFP TEMP01,WREG ; Eliminación del dato de RB5 para poder
continuar con la revisión de RB6 y RB7
        ANDLW B'11000000' ; TEMP01 = estado actual del puerto
        MOVWF TEMP01

        MOVFP TEMP03,WREG ; Revisa el último estado del puerto para
evitar un pulso falso en descenso
        XORLW B'11000000' ; (11000000), TEMP03 = último estado del
puerto
        BTFSC ALUSTA,2
        GOTO POS_FIN

        MOVFP TEMP01,WREG ; Revisa el estado actual del puerto para
evitar un pulso falso en descenso
        XORLW B'00000000' ; (00000000)
        BTFSC ALUSTA,2
        GOTO POS_FIN

        MOVFP TEMP01,WREG ; Revisa el estado actual del puerto para
incremento en la entrada del segundo pulso
        XORLW B'11000000' ; (11000000)
        BTFSC ALUSTA,2
        GOTO POS_FIN

        BTFSS TEMP01,7 ; Dependiendo del pulso aceptado se
incrementa o decrementa
        GOTO POS_INC
        GOTO POS_DEC

```

```

;-----
;
; Configuración del Timer 3 para determinar la frecuencia de muestreo de
la
; velocidad
;-----

```

CONF_VEL

```

    MOVLB 3
    BCF   TCON2,2
    MOVLB 2
    MOVFP VELCONF01,WREG
    MOVWF PR3L
    MOVFP VELCONF02,WREG
    MOVWF PR3H
    CLRF  TMR3L
    CLRF  TMR3H
    MOVLB 3
    BCF   TCON1,    2
    BSF   TCON2,    2

    RETURN

```

```

;-----
;
; Determinación de la velocidad del motor, la velocidad debe tener a lo
; sumo 8 bits o de lo contrario se genera una señal de error
;-----

```

VEL_ERR

```

    BSF   VELERR,0
    RETURN

```

VEL_ERR_NEG

```

    MOVLW H'FF'
    XORWF VEL03
    BTFSS ALUSTA,2
    GOTO  VEL_ERR
    XORWF VEL02
    BTFSS ALUSTA,2
    GOTO  VEL_ERR
    RETURN

```

VEL_ERR_POS

```

    MOVLW H'00'
    XORWF VEL03
    BTFSS ALUSTA,2
    GOTO  VEL_ERR
    XORWF VEL02
    BTFSS ALUSTA,2
    GOTO  VEL_ERR
    RETURN

```

VELOCIDAD

```

    MOVLB 1           ;Borra el bit de interrupción

```

```

        BCF   PIR, 6

        INCF  TEMP04           ;Utilización del contador auxiliar,
divide el reloj en 256
        BTFSS ALUSTA, 2
        RETURN

        MOVFP POSA01, WREG ;resta de las posiciones actuales y
anteriores
        SUBWF POS01, 0
        MOVWF VEL01
        MOVFP POSA02, WREG
        SUBWFB      POS02, 0
        MOVWF VEL02
        MOVFP POSA03, WREG
        SUBWFB      POS03, 0
        MOVWF VEL03

        MOVFP POS01, WREG ;Archiva la posición actual el siguiente
cálculo
        MOVWF POSA01
        MOVFP POS02, WREG
        MOVWF POSA02
        MOVFP POS03, WREG
        MOVWF POSA03

        BCF   VELERR, 0      ;Borra el error de la velocidad y revisa si
actualmente hay error
        BTFSS VEL03, 7
        GOTO  VEL_ERR_POS
        GOTO  VEL_ERR_NEG

;-----
;
; Rutina de TX de datos
;-----
;

ESP01
        MOVLB 0
        BTFSS TXSTA, 1
        GOTO  ESP01
        RETURN

TRANSMICION
        MOVLB 0           ; Desactiva la recepción del USART
        BCF   RCSTA, 4
        MOVLB 1
        BSF   PORTC, 0      ; Activa una configuración de TX de
datos
        BSF   PORTC, 1
        MOVLB 0
        BSF   TXSTA, 5      ; Activa la transmisión del USART

        MOVLW B'10000000' ; Se envía la palabra de Inicio
        MOVWF TXREG

```

```

CALL ESP01

MOVFP TXDR01,WREG ; Se envía la dirección del destino
MOVWF TXREG
CALL ESP01

MOVFP DIRR,WREG ; Se envía la dirección de la fuente
MOVWF TXREG
CALL ESP01

MOVFP TXCM01,WREG
MOVWF TXREG
CALL ESP01

MOVFP TXCM02,WREG
MOVWF TXREG
CALL ESP01

MOVFP TXDT01,WREG
MOVWF TXREG
CALL ESP01

MOVFP TXDT02,WREG
MOVWF TXREG
CALL ESP01

MOVFP TXDT03,WREG
MOVWF TXREG
CALL ESP01

MOVFP TXDT04,WREG
MOVWF TXREG
CALL ESP01

MOVLW B'10000000' ; Se envía la palabra de Final
MOVWF TXREG
CALL ESP01

MOVLB 0 ; Desactiva la transmisión del USART
BCF TXSTA,5
MOVLB 1 ; Coloca en el puerto C LSR el cierre del
Driver
datos BCF PORTC,1 ; Activa una configuración de TX de
BCF PORTC,0
MOVLB 0 ; Activa la recepción del USART
BSF RCSTA,4

RETURN

```

```

;-----
;
; Rutina de RX de datos
;-----

```

```

ERROR01
    BSF    RXERROR,0    ;Marcación de error
    MOVLB 0
    BCF    RCSTA,4      ;Reseteo del puerto para eliminar el
error
    BSF    RCSTA,4
    RETURN

RXGRAB                                ;Grabación de los archivos RX en los
archivos de datos
    MOVFP  RXIN03,WREG
    MOVWF  RXDR01
    MOVFP  RXIN04,WREG
    MOVWF  RXCM01
    MOVFP  RXIN05,WREG
    MOVWF  RXCM02
    MOVFP  RXIN06,WREG
    MOVWF  RXDT01
    MOVFP  RXIN07,WREG
    MOVWF  RXDT02
    MOVFP  RXIN08,WREG
    MOVWF  RXDT03
    MOVFP  RXIN09,WREG
    MOVWF  RXDT04

    BSF    RXERROR,1    ;Indica que se realizó una recepción
    BTFSC  RXECO,0      ;Si el eco está activado se genera una
trasmisión
    CALL  RX_TX_ECO
    RETURN

RECEPCION
    MOVLB 0
    BCF    RXERROR,0
    BTFSC  RCSTA,2      ; Revisa un error de "Framing"
    GOTO  ERROR01
    BTFSC  RCSTA,1      ; Revisa un error de "Overrun"
    GOTO  ERROR01

    MOVFP  RXIN02,WREG ;Movimiento de los archivos en el FIFO
    MOVWF  RXIN01
    MOVFP  RXIN03,WREG
    MOVWF  RXIN02
    MOVFP  RXIN04,WREG
    MOVWF  RXIN03
    MOVFP  RXIN05,WREG
    MOVWF  RXIN04
    MOVFP  RXIN06,WREG
    MOVWF  RXIN05
    MOVFP  RXIN07,WREG
    MOVWF  RXIN06
    MOVFP  RXIN08,WREG
    MOVWF  RXIN07
    MOVFP  RXIN09,WREG
    MOVWF  RXIN08

```

```

MOVFP RXIN10,WREG
MOVWF RXIN09
MOVLB 0
MOVFP RCREG,WREG
MOVWF RXIN10

MOVLW B'10000000' ;Revisión del byte inicial y final
XORWF RXIN01,0
BTSS ALUSTA,2
RETURN
MOVLW B'10000000'
XORWF RXIN10,0
BTSS ALUSTA,2
RETURN
MOVFP DIRR,WREG ;Revisión de la dirección de recepción
XORWF RXIN02,0
BTSS ALUSTA,2
CALL RXGRAB

CLRF RXIN01 ;Borrado de los archivos RX
CLRF RXIN02
CLRF RXIN03
CLRF RXIN04
CLRF RXIN05
CLRF RXIN06
CLRF RXIN07
CLRF RXIN08
CLRF RXIN09
CLRF RXIN10

RETURN

```

```

;-----
;
; Rutina de Eco: Toda la información RX se vuelve a TX
;-----
;

```

RX_TX_ECO

```

MOVFP RXDR01,WREG
MOVWF TXDR01
MOVFP RXCM01,WREG
MOVWF TXCM01
MOVFP RXCM02,WREG
MOVWF TXCM02
MOVFP RXDT01,WREG
MOVWF TXDT01
MOVFP RXDT02,WREG
MOVWF TXDT02
MOVFP RXDT03,WREG
MOVWF TXDT03
MOVFP RXDT04,WREG
MOVWF TXDT04
CALL TRANSMICION
RETURN

```

```
;-  
;-  
; PROGRAMA PRINCIPAL  
;-  
;-
```

```
;-  
;-  
; Envio del dato recibido a las posiciones del PWM  
;-  
;-
```

```
PWM_CAMBIO  
    MOVFP RXDT01,WREG  
    MOVWF PWMOUT01  
    MOVFP RXDT02,WREG  
    MOVWF PWMOUT02  
  
    RETURN
```

```
;-  
;-  
; Determina la segunda parte del comando de posición y responde en  
; funcion de éste  
;-  
;-
```

```
POS_RESET  
    MOVLW 0  
    MOVWF POS01  
    MOVWF POS02  
    MOVWF POS03  
    MOVWF POSA01  
    MOVWF POSA02  
    MOVWF POSA03  
    RETURN
```

```
POS_TX  
  
    MOVFP RXDR01,WREG  
    MOVWF TXDR01  
    MOVFP RXCM01,WREG  
    MOVWF TXCM01  
    MOVFP RXCM02,WREG  
    MOVWF TXCM02  
    MOVFP POS01,WREG  
    MOVWF TXDT01  
    MOVFP POS02,WREG  
    MOVWF TXDT02  
    MOVFP POS03,WREG  
    MOVWF TXDT03  
    MOVLW 0  
    MOVWF TXDT04  
    CALL TRANSMICION  
    RETURN
```

POS_CAMBIO

```
    MOVLW 82           ;Revisión del reset ("R")
    XORWF RXCM02,0
    BTFSC ALUSTA,2
    GOTO  POS_RESET

    MOVLW 79           ;Revisión del output ("O")
    XORWF RXCM02,0
    BTFSC ALUSTA,2
    GOTO  POS_TX

    RETURN
```

```
-----
;
; Determina la segunda parte del comando de velocidad y responde en
; función de éste
;-----
--
```

VEL_SET

```
    MOVFP RXDT01,WREG
    MOVWF VELCONF02

    MOVFP RXDT02,WREG
    MOVWF VELCONF01
    CALL  CONF_VEL

    RETURN
```

VEL_TX

```
    MOVFP RXDR01,WREG
    MOVWF TXDR01
    MOVFP RXCM01,WREG
    MOVWF TXCM01
    MOVFP RXCM02,WREG
    MOVWF TXCM02
    MOVFP VEL01,WREG
    MOVWF TXDT01
    MOVLW 0
    MOVWF TXDT02
    MOVWF TXDT03
    MOVWF TXDT04
    CALL  TRANSMICION

    RETURN
```

VEL_CAMBIO

```
    MOVLW 83           ;Revisión del set ("S")
    XORWF RXCM02,0
    BTFSC ALUSTA,2
    GOTO  VEL_SET

    MOVLW 82           ;Revisión del output ("O")
    XORWF RXCM02,0
```

```

BTFSO ALUSTA,2
GOTO VEL_TR

RETURN

```

```

-----
--
; Activa / desactiva el eco en la RX
;-----
--

```

```

ECO_CAMBIO
    MOVLW 44          ;Revisión del prender ("O")
    XORWF RXCM02,0
    BTFSO ALUSTA,2
    BCF  RXECO,0
    BSF  RXECO,0

    RETURN

```

```

-----
--
; Rutina de lectura de información entrante
;-----
--

```

```

COMANDO01
    BCF  RXERROR,1   ;Limpia la bandera de RX

    MOVLW 79          ;Revisión del cambio del PWM ("O")
    XORWF RXCM01,0
    BTFSO ALUSTA,2
    GOTO PWM_CAMBIO  ;Condición si hay coincidencia

    MOVLW 60          ;Revisión de los comandos de posición ("P")
    XORWF RXCM01,0
    BTFSO ALUSTA,2
    GOTO POS_CAMBIO  ;Condición si hay coincidencia

    MOVLW 80          ;Revisión de los comandos de velocidad ("V")
    XORWF RXCM01,0
    BTFSO ALUSTA,2
    GOTO VEL_CAMBIO  ;Condición si hay coincidencia

    MOVLW 69          ;Revisión de los comandos de velocidad ("E")
    XORWF RXCM01,0
    BTFSO ALUSTA,2
    GOTO ECO_CAMBIO  ;Condición si hay coincidencia

    RETURN

```

```

-----
--
; Rutina recursiva

```

--
Start

```
CLRWDI  
BTFS RXERROR,1  
GOTO Start  
CALL COMAND001  
GOTO Start  
END
```

Apéndice G
Características técnicas RadioModems 3JTech

3JTECH RadioModem

Multiple Point-to-Point Wireless Modem



Replace your serial cable by two of our RadioModem which runs at 14,400 bps and above. The serial cable could be connected between PCs, PC and printer, PC and Modem. It could also offer modem and printer sharing capability for multiple PC's. Besides, it can also provide an arbitrary radio link between any pair of PCs. It could also be configured as either a DCE for connecting to PC, or as a DTE for connecting to external modem or printer. It uses the spread spectrum technology working in ISM 902 - 928 MHz band to provide data security and anti-interference capabilities

There is no restrict number of terminals. However, one terminal could establish only one link with any other one. This is so called connection-oriented or point-to-point link. Maximum of 10 links could be established at one time. Firmware modification or software developm could be accomodated for special link like lan application.

Function Description

1. Use spread spectrum technology to operate 902-928 Mhz ISM band, in compliance wi US FCC 15.247 rule.
2. Can be configured as a DCE for connecting to PC or a DTE for connecting to exter modem or printer.
3. Regardless of the number of wl-modems, at one instant, one RadioModem can establ only one radio link with any another one which results in a virtually transparen RS232C link.
4. During the incoming call operation, the RadioModem connecting to external modem create the link with any RadioModem according to the address setting.
5. The synchronous half-duplex packeted data rate is 85.33Kbps.
6. Use the half-duplex data mode of SST IC as the physical layer protocol to imple the virtually full-duplex connection between terminals.
7. Adopt the X.25-LAPB/HDLC-based data link layer protocol in the synchronous seria port to provide rate adaptation, link maintenance, error control, flow control, addressing capabilities.
8. The packet size of frames is dynamically adjusted according to the data flow rat error rate. Hence, the propagation delay varies continuously.
9. The RF module provides 10 RF channels to be used, and the output power is 20 dBm
10. Provide selectable PN code sets.
11. In the slave mode, RadioModem should automatically scan each of the 10 RF chann and average the RSSI (received signal strength indicator) values of each channe This is the so called Auto Scan function.
12. In the beginning of master mode, RadioModem should choose one clearest RF chann based on the average RSSI values to create the radio link with the remote slave In other words, it should avoid to choose the channel in which there is certain

- interference present. This is called the Automatic Interference Avoidance function.
13. During the time of data transmission, if certain interference appears which results in the abortion of the link, the master should be able to detect the abortion and automatically change to the channel that is interference free. This is called the Automatic Channel Change function. If the interference does not cause the link to be interrupted, RadioModem will sustain the link continuously. This is the so-called Immunity to Interference.
 14. The configuration of RadioModem will be designed to be similar to the method of external modem used. Besides, we may need some Windows drivers to facilitate the usage of this product from the viewpoint of customers.
 15. The hardware extensibility of voice capability will be implemented in this version for future extension.

The minor features are described as follows:

- Provide the indicators of DCD (data carrier detect).
 - Provide the indicator of command mode or data transmission mode.
 - Provide the indicator of power on.
-

In addition to the above functions, this product could be upgraded to achieve the following extensible functions in the future.

1. Based on the minor modification of RF module, the synchronous data rate of SST I can be upgraded up to half-duplex 160Kbps, which means the overall radio link's throughput could be increased. However, in contrast, the usable number of RF channels will be decreased.
2. If the development of the RF module mentioned above is finished, with some upgrade this product could be used to connect to an ISDN Terminal Adaptor.
3. Due to the developing RF module of high power version, in the future, the RadioModem could offer longer range.

Specification

- **Dimension:** 105 x 70 x 25 mm
 - **Operation environment:** temperature range: 0 - 50 °C relative humidity range: 10 - 90%.
 - **Storage environment:** temperature range: 20 - 70 °C relative humidity range : 5 - 95%.
 - **Power requirement:** DC 5V from the keyboard connector of desktop or notebook PC. Maximum consumption is less than 250mA.
 - 14,400 full duplex
 - 100MW RF transmission
 - 902 ~ 928 MHz SST
 - Auto scan for optimum channel
-

3JTECH

RadioModem Installation Guide

Introduction

The RadioModem you just purchased uses the most advanced Spread Spectrum Technology (SST) to increase signal to noise ratio for the wireless transmission and receiving. It uses the frequency range from 902 to 928 mega hertz. Two RadioModems could replace any RS232 cable up to a certain distance depending on the working environment and the strength of the RF module. Our RadioModem is built-in with a 100 mw RF module. The transceiving distance is about 300 meters in the open space. In order to increase the distance you would need a RF booster and/or a directional antenna. Please check with your government regulation regarding using the RF booster. For the frequency that this RadioModem uses, it is known to be legal for usage up to one watt transceiving power in certain countries.

You will need one RadioModem at each end of the RS232 communication. Set the dip switch to be the same for each pair. Push the button on one of the Modem until "modem ready" light is turn on. You will then be all set. The two modems will let the system look as there is a cable in between the devices for up to three hundred (300) meters in open space. The transceiving will be about three concrete stories inside a building. No specific software nor driver is needed for the RadioModem to work. For Mac user, there is a special driver supplied by 3JTech (a3jtech@ms4.hinet.net) for accessing Apple local network via Apple Remote Access. The word *modem* is used for representing modulation and de-modulation. The RadioModem could not replace the regular data/fax modem. However, it could be used in series with regular data/fax modems.

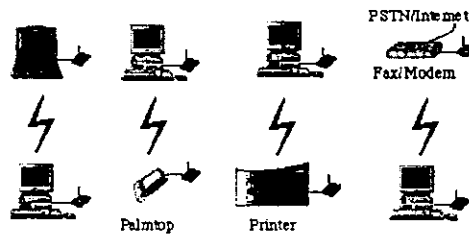
For more information please visit:

<http://www.a3j.com.tw/English/RadioModem.html>

You could use our RadioModem for maximum ten point to point serial (RS232) communications. Set each pair of the RadioModems to be the same ID by setting them to be the same dip switch configuration. There are five dip switches on each RadioModem. Total number of logical ID is therefore thirty-two. However, the RadioModem uses ten physical channels for transceiving and scans the optimum channel when setting up the connection and during transceiving. The results of over ten point to point communications will slow down the communication speed significantly.

Installation

- 1) Set the dip switch configuration of each pair of RadioModem to be the same,
- 2) Connect the serial cable for each modem,
- 3) Connect the 5VDC power adapter, or connect the keyboard, PSII mouse adapter to obtain the 5VDC power from keyboard port or PSII mouse ports;
- 4) Press the start link button of either RadioModem until the *MR* light is turn on,



- Special Drivers are available for Apple Remote Access:



**Possible configurations of system connections
(Maximum 10 point-to-point users)**

*If you have any questions or suggestions regarding our World Wide Webs
please send them to
[✉ a3jtech@ms4.hinet.net](mailto:a3jtech@ms4.hinet.net)*

**9698 Telstar Ave. #305 El Monte, CA, 91731 U.S.A
Tel: (626)575-8899; Fax:(626)575-8800**

**342 Fushing N. Rd., 2F, Taipei, Taiwan, Tel: 886-2-2500 6916;
Fax: 886-2-25069793; 2516-1305**

BACK

- 5) Use any application software (e.g. Laplink, Telix, etc.) or system utility (e.g.: serial printer) as there is a RS232 cable connected,
- 6) Set the DTE speed to be 14,400 bps inside your application software or system utility.

** RadioModem Cable Connection for MAC

** RadioModem Cable Connection for PC

*If you have any questions or suggestions regarding our World Wide Website,
please send them to*

a3jtech@ms4.hinet.net

**709 Brea Canyon Rd., Suite #4, Walnut, CA 91789, USA;
Tel: (909) 869-9305; Fax:(909)869-9315**

**342 Fushing N. Rd., 2F, Taipei, Taiwan, Tel: 886-2-500 6916;
Fax: 886-2-5069793**

